Wolfgang Borutzky *Editor*

Bond Graphs for Modelling, Control and Fault Diagnosis of Engineering Systems

Second Edition

Springer

Bond Graphs for Modelling, Control and Fault Diagnosis of Engineering Systems

Wolfgang Borutzky
Editor

Bond Graphs for Modelling, Control and Fault Diagnosis of Engineering Systems

Second Edition

Springer

Editor
Wolfgang Borutzky
Bonn-Rhein-Sieg University of Applied Sciences
Department of Computer Science
Sankt Augustin, Germany

ISBN 978-3-319-47433-5 ISBN 978-3-319-47434-2 (eBook)
DOI 10.1007/978-3-319-47434-2

Library of Congress Control Number: 2016955827

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Foreword

Prof. Borutzky has asked me to write a foreword to this second edition of the book *Bond Graph Modelling of Engineering Systems*. This put me in mind of the time when Ronald C. Rosenberg and I, as Assistant Professors at MIT, used to stop at a bar in Harvard Square to wait for the traffic to subside on our way home from work. (Ron's doctoral thesis resulted in the ENPORT program, the first bond graph processor, which, although restricted to linear systems, would solve the problems associated with differential causality in simulation.) We often discussed what technical areas we should work on in the future and we came to the conclusion that bond graphs, recently invented by our advisor, Prof. Henry M. Paynter, were worthy of being better known to the general world rather than just to MIT graduate students.

The result was our first book intended for self-study of bond graph methods, *Analysis and Simulation of Multiport Systems: The Bond Graph Approach to Physical System Dynamics*, The MIT Press, 1968. When the book was finished, Prof. Paynter asked us if he could provide a Foreword in the form of a Historical Note. We naturally agreed and the result is a fascinating two page history of Paynter's thought starting, he says, in high school. His personal references go back to the 1950s, but his references to chemical structure graphs go back another 100 years.

His final paragraph of his Historical Note is as follows:

> Thus were devised on April 29, 1959, the two ideal 3-port energy junctions (0,1) to render the system of bond graphs a complete and formal discipline. The complete system was then first published in 1961.

As you can see, Prof. Paynter was thinking of his reputation in the future.

The publication mentioned was the MIT press book *Analysis and Design of Engineering Systems*, 1961, was in fact a paperback compilation of class notes taken by a graduate student and, as Paynter admitted, the glued together book tended to fall apart as its pages were turned. (This was a great disadvantage to slow learners.) Thus, bond graphs remained for a while only known to those who had studied at MIT.

Professor Paynter would certainly be gratified to see the interest in bond graphs that has developed in the past 60 years. The present volume is a testimony to large number of researchers who are presently contributing to extensions of bond graph techniques and to new application areas. He also might be surprised to learn that his "complete and formal discipline" has been extended in directions he never would have thought possible.

There is, of course, some danger in extending bond graph techniques to include idiosyncratic notation and manipulations that apply to ever more special cases. In the early years, those of us interested in promoting bond graph methods had to fight those who liked the force-current analogy better than the force-voltage analogy since they produced bond graphs with 0's and 1's and I's and C's interchanged, which provided a barrier to those newcomers interested in learning the benefits of bond graph methods. This volume is an example of the correct approach in which the authors take pains to make clear why special notation or special manipulations on a bond graph are necessary to provide insight into problems which don't yield to standard bond graph techniques.

When I contemplate Paynter's contribution to physical system dynamics, and of those of present day engineers who continue to advance bond graph techniques as evidenced by the present volume, I think of the well-known quote from Goethe's Faust:

> Was du ererbt von deinen Vätern hast, erwirb es, um es zu besitzen.

(That which you inherit from your fathers, you must earn in order to possess.)

The authors of the present volume have clearly taken this advice to heart.

Davis, CA, USA Dean Karnopp
July 2016

Preface

By the beginning of 2011, Springer published a compilation text of a number of researchers from all over the world on Bond Graph Modelling of Engineering Systems. The book covers theoretical and methodological topics as well as some applications and the use of software for bond graph modelling and simulation. The aim was to present the state of the art in bond graph modelling by addressing latest developments and integrating various works in a unified manner. In summer 2014, the editor was invited to prepare a second edition of the 2011 compilation text. Due to the active ongoing research of my colleagues, their commitment, their expertise, and their contributions the outcome of this new book project, in my view, has become more than a revision of the first edition. This new book again covers some theoretical issues, applications and software support as the first edition. However, the focus of the second edition is on latest topics as well as on subjects that haven't been covered in the first edition. Moreover, new young excellent researchers have joined the team together with co-authors who have been using bond graph in their research and in teaching for a long time. Therefore, the result of this project may be considered a novel book that reflects the active research in the bond graph modelling community and latest achievements over a rather short period of the past five years. A number of the contributed chapters have arisen from recently finished PhD theses.

Bond graphs were devised by Professor H. Paynter some 60 years back at the Massachusetts Institute of Technology (MIT) in Cambridge, Massachusetts, USA, and were developed into a methodology by his former PhD students Professor D. Karnopp and Professor D. Margolis (University of California at Davis) and Professor R. Rosenberg (Michigan State University, East Lansing, Michigan). Since then the bond graph approach to physical system modelling and simulation has been adopted by many researchers. The contributions to this book well demonstrate that today, 60+ years later, the bond graph methodology has spread from MIT all over the world and is successfully used in various engineering fields.

The book is organised into five parts. Each of them starts with an introduction that motivates the subjects and establishes relations between the chapters of a part.

The contributed chapters show that bond graphs are more than just a methodology for modelling and simulation of multidisciplinary systems described by continuous time models. Beyond that application area, bond graphs can also be used to represent *hybrid* models with continuous time and discrete state variables. Furthermore, bond graph modelling has proven useful in control and as one possible starting point for model-based fault diagnosis and fault prognosis in engineering systems. These areas become more and more important with an increase of the complexity of today's mechatronic systems and a trend to at least partly autonomous systems. Representing hybrid models in a bond graph framework and bond graph model-based fault diagnosis and fault prognosis are topics in the focus of this new book. Accordingly, the title of the first edition has been adapted.

Another topic that is not new but of still ongoing interest is the question of how open thermodynamic subsystems with compressible fluid flow and mechanical subsystems described by a distributed parameter model can be integrated in a lumped parameter model of a complex overall system. Part III presents an extension of common bond graphs for modelling thermodynamic subsystems introduced by Professor Brown that can be combined with conventional bond graphs for other parts of an engineering system and is supported by a mature modelling and simulation software environment newly developed over the past years. As to distributed parameter subsystems, an approach based on finite segments and an activity metric is presented that aims at a lumped parameter model of reduced order sufficiently accurate.

Part IV illustrates the remarkable wide range of applications in which bond graphs have been used for modelling mechatronic systems of current interest by contributed chapters ranging from a wheelchair, to robots for in vivo surgery, or walking machines that can be used for operations in hazardous fields, to wind turbines in the area of renewable energy sources.

Professor Henry Paynter, the inventor of bond graphs, envisaged applications of bond graph modelling to chemistry, electrochemistry and biochemistry already back in 1992. Since then not too many researchers have used bond graphs for modelling chemical reactions. One chapter in Part IV extends bond graph modelling even to biomolecular systems of living organisms. As the latter ones are open systems that are not at thermodynamic equilibrium, bond graph representations of their biomolecular systems are not evident. The chapter introduces appropriate bond graph variables. Molecular species are represented by a nonlinear C element and reactions by a nonlinear two-port R element. As a result, bond graph modelling as a graphical approach gives more intuitive insight than purely numerical approaches.

As to the results of simulation runs, it has become common that some software provides a postprocessing module that enables a visualisation of mechanical motion in 3D under different angles of views. The last chapter of this compilation text presents a novel approach that links a bond graph modelling and simulation program to a program for the geometrical design of complex multibody systems such as robots with several degrees of freedom so that the 3D motion of a robot is visualised simultaneously with some little delay to its numerical computation.

Although this contributed book aims to reflect the current state of the art in bond graph modelling by presenting and discussing advanced recent topics, all chapters have been written in such a way that newcomers to the methodology with some knowledge of the basics may easily get into the vast fascinating and open field of advanced bond graph modelling. Readers who may want to have a closer look at bond graph fundamentals may find references to latest monographs and textbooks. Furthermore, each chapter provides many references to conference papers, journal articles and PhD theses on advanced topics. This multiple authors book well complements latest monographs and textbooks on bond graph modelling and may serve as a guide for further self-study and as a reference.

Bond Graphs for Modelling, Control and Fault Diagnosis of Engineering Systems continues the presentation of its predecessor titled *Bond Graph Modelling of Engineering Systems* and like the latter one addresses readers in academia as well as practising engineers in industry and invites experts in related fields to consider the potential and the state of the art of bond graph modelling. It is hoped that the bond graph methodology may contribute to more insight into physical processes and to the development of useful models in their engineering field.

Acknowledgements

This volume would not have been possible without the commitment, the expertise and the efforts of my colleagues. I would like to take this opportunity to thank all my co-authors for their contributed chapters. Likewise, I wish to thank all colleagues who kindly took the time to read the chapters. Their valuable comments and suggestions are gratefully acknowledged.

My special thanks go to Professor Dean Karnopp, University of California at Davis, one of the leading pioneers of the bond graph methodology, who kindly honoured all authors of this compilation text by writing the foreword with some personal notes.

Also, I appreciate the support this multiple author book project received from Professors Yuri Merkuryev, Riga Technical University, Latvia; David Murray-Smith, University of Glasgow, Scotland, UK; Ronald Rosenberg, Michigan State University, East Lansing, MI, USA and Gregorio Romero, Universidad Politécnica de Madrid, Spain.

Last but not least, I wish to thank my Editor with Springer at New York, NY, USA, Mary James, for her invitation to this book project, for her kind constant support during this book project and Brinda Megasyamalan, Project Coordinator, Sasi Reka, Project Manager, for the cooperation throughout the book production.

Sankt Augustin, Germany Wolfgang Borutzky
July 2016

Contents

Part I Bond Graph Theory and Methodology

Part II Bond Graph Modelling for Fault Diagnosis, Fault Tolerant Control, and Prognosis

Part III Thermodynamic Systems and Distributed Parameter Systems

Part IV Applications

Part V Software

Contributors

Jorge Luis Baliño Escola Politécnica, Universidade de São Paulo, São Paulo-SP, Brazil

W. Borutzky Bonn-Rhein-Sieg University of Applied Sciences, Sankt Augustin, Germany

Benjamin Boudon Université Aix-Marseille, Marseille, France

Peter C. Breedveld Twente University, Enschede, The Netherlands

Forbes T. Brown Lehigh University, Bethlehem, PA, USA

Jean-Claude Carmona Ecole Nationale Supérieure d'Arts et Métiers, Centre d'Aix-en-Provence, Aix-en-Provence, France

Majda Čohodar University of Sarajevo, Sarajevo, Bosnia and Herzegovina

Vjekoslav Damić University of Dubrovnik, Dubrovnik, Croatia

G. Dauphin-Tanguy Ecole Centrale de Lille, Villeneuve d'Ascq, France

Abdennasser Fakri Université Paris-Est, Marne-la-Vallée, France

Jesus Felez Universidad Politécnica de Madrid (UMP), Madrid, Spain

Peter J. Gawthrop University of Melbourne, Parkville, VIC, Australia

M.M. Gor G.H. Patel College of Engineering & Technology, Gujarat, India

Mayank S. Jha Institut National des Sciences Appliquées de Toulouse (INSA), Toulouse, France

Sergio J. Junco Universidad Nacional de Rosario, Rosario, Argentina

S.W. Kwak Keimyung University, Daegu, South Korea

L.S. Louca University of Cyprus, Nicosia, Cyprus

François Malburet Ecole Nationale Supérieure d'Arts et Métiers, Centre d'Aix-en-Provence, Aix-en-Provence, France

Rebecca Margetts University of Lincoln, Lincoln, UK

Matías A. Nacusse Universidad Nacional de Rosario, Rosario, Argentina

Roger F. Ngwompo University of Bath, Bath, UK

B. Ould-Bouamama Ecole Polytechnique de Lille, Villeneuve d'Ascq, France

Pushpraj Mani Pathak Indian Institute of Technology, Roorkee, Uttarakhand, India

Om Prakash Indian Institute of Technology, Kharagpur, India

A. Medina Rios Universidad Michoacana de San Nicolás de Hidalgo, Morelia, Mexico

A.K. Samantaray Indian Institute of Technology, Kharagpur, India

R.Tapia Sánchez Universidad Michoacana de San Nicolás de Hidalgo, Morelia, Mexico

Japie Petrus Vilakazi Tshwane University of Technology, Pretoria, Republic of South Africa

J.M. Yang Kyungpook National University, Daegu, South Korea

Abbreviations

ANN	Artificial Neural Network
ARR	Analytical Redundancy Relation
AVP	Angular Velocity Propagation
BD	Block Diagram
BDF	Backward Differentiation Formula
BEM	Blade Element Momentum
BG	Bond Graph
BGD	Bond graph in preferred derivative causality
BG-LFT	Bond Graph in Linear Fractional Transformation
CBM	Condition Based Maintenance
CFD	Computational Fluid Dynamics
CG	Centre of Gravity
CJ	Controlled Junction
CKC	Closed Kinematic Chain
CTF	Coordinate Transformation Block
DAE	Differential-Algebraic Equation
DBG	Diagnostic Bond Graph
DC	Direct Current
DFIG	Doubly Fed Induction Generator
DFIM	Doubly Fed Induction Motor
DHBG	Diagnostic Hybrid Bond Graph
DM	Degradation Model
DOF	Degree of Freedom
DPP	Degradation Progression Parameter
DSCAP	Dynamic Sequential Causality Assignment Procedure
EJS	Euler Junction Structure
EOL	End of Life
EPPW	Electric Power Propelled Wheelchair
FDI	Fault Detection and Isolation
FDM	Finite Difference Method
FEM	Finite Element Method

FROI	Frequency Range of Interest
FSA	Finite State Automaton
FSM	Finite Signature Matrix
FTC	Fault Tolerant Control
FVM	Finite Volume Method
GARR	Global Analytical Redundancy Relation
GFSM	Global Fault Signature Matrix
GFSSM	Global Fault Sensitivity Signature Matrix
GSJ	Generalized Switched Junction
GYS	Gyristor
HBG	Hybrid Bond Graph
HJSM	Hybrid Junction Structure Matrix
HyPA	Hybrid Process Algebra
JSM	Junction Structure Matrix
LCP	Linear Complementarity Problem
LCS	Linear Complementarity System
LFT	Linear Fractional Transformation Form
LL	Leg Lift
LTI	Linear Time Invariant system
LWR	Lighthill, Whitham and Richards
MBDP	Model-Based Diagnosis and Prognosis
MBG	Multibond Graph
MBS	Multibody System
MCSM	Mode Change Signature Matrix
MCSSM	Mode Change Sensitivity Signature Matrix
MDM	Mechatronic Drive Modular
MDMW	Modular Driven Manual Wheelchair
MFC	Microsoft Foundation Class
MGB	Main Gear Box
MIMO	Multiple Input Multiple Output system
MODA	Model Order Reduction Algorithm
MORA	Model Order Deduction Algorithm
MPW	Manual Propelled Wheelchair
MSC	Machine Side Converter
NSC	Network Side Converter
ODE	Ordinary Differential Equation
PD	Proportional Derivative
PDE	Partial Differential Equation
PDF	Probability Density Function
PF	Particle Filter
PI	Proportional Integral
PID	Proportional, Integral and Derivative
PL	Prismatic Link
RA	Relative Accuracy
RANS	Reynolds-averaged Navier-Stokes

RMSE	Root Mean Square Error
RP	Revolute-prismatic
RUL	Remaining Useful Life
SARIB	Système anti-résonateur intégré à barres
SBG	Sensitivity Bond Graph
SCAP	Sequential Causality Assignment Procedure
SCR	Silicon Controlled Rectifier
SIR	Sampling Importance Resampling
SIS	Sequential importance sampling
SISO	Single Input Single Output system
SL	Step Length
SOH	State of Health
SPJ	Switched Power Junction
SS-Bond	Switchable Structure Bond
SSF	Set of Suspected Fault
SwBG	Switched Bond Graph
TCP	Tool Centre Point
TTF	Time to Failure
TVP	Translational Velocity Propagation
UIO	Unknown Input Observer
VTK	Visualization Toolkit
ZC	Zero Compliance
ZCP	Zero-order Causal Path
ZE	Zero Effort
ZF	Zero Flow

Part I
Bond Graph Theory and Methodology

In the first part of this multi-author book, Chap. 1 discusses fundamentals of bond graph modelling of multi-domain physical systems by starting from the consideration of symmetries, first principles and key concepts. It is shown that the common nine basic elements of bond graph modelling result from fundamental principles of physics, that a bond graph originates from a multiport storage, effort sources, and a power continuous multiport, and that multiports with linear constitutive relations can be decomposed in terms of the nine basic elements. The presentation in Chap. 1 clearly demonstrates that bond graph modelling is based on a sound theory.

Once the system components have been represented by ideal bond graph (BG) elements and the energy exchange between power ports has been captured by bonds, various procedures are available to extend an *acausal* bond graph into a *causal* one from which a mathematical model can be derived in a systematic manner. To that end, the well-known Sequential Causality Assignment Procedure (SCAP), introduced by Karnopp and Rosenberg, is widely used. Depending on the resulting causal pattern the mathematical model can take a state-space form or more generally the form of a set of Differential-Algebraic Equations (DAEs). Other known approaches try to assign derivative causalities to all storage elements (all-derivative procedure), use relaxed causalities at junctions, or derive Lagrange equations. If the inspection of causal paths in a BG reveals that a state space model can be derived, then it is well known that also transfer functions can be directly derived from the BG. In case causal paths indicate that the mathematical model is of the form of a DAE system, known approaches are to either account for small storage effects which turn the DAE system into a stiff ODE system and increase the number of equations, or to introduce Lagrange multipliers, or to determine tearing variables that break algebraic loops. In the area of modelling multibody systems, the joint coordinate method is used which provides a minimal set of ODEs equal to the number of degrees of freedom (DOFs) by means of a velocity transformation.

Chapter 2 addresses the derivation of a *minimal* set of equations derived from a BG representing a continuous time model. A minimal set of equations to be solved again and again in the simulation loop is important for real-time simulation. To that end, this chapter considers small examples of causal BGs with different types of so-called Zero-Order-Causal Paths (ZCPs). In the case of a causal path between a storage port in integral causality and a port of another storage element in derivative causality, a procedure equivalent to the joint coordinate method is used. For BGs with a ZCP of another class or with a combination of several ZCPs of different type it is shown that ZCPs or at least some of them disappear if the preferred integral causality of storage elements in appropriate places is changed into derivative causality. As a result, either a set of ODEs for the independent storage elements can be read out or a DAE system with a reduced number of break variables.

The next two chapters of Part I address BG representations of *hybrid* system models and the generation of a hybrid DAE system from a BG. Over the past decades, various bond graph representations of *hybrid* models have been proposed in the literature. All of them have their pros and cons and it appears that none of them has become a generally accepted standard in bond graph modelling so far.

Like some other approaches, the one presented in Chap. 3 aims at a single BG representation of which the structure and causalities remain fixed for all system modes. To that end, Chap. 3 introduces two new concepts that capture structural model changes. One of them is inspired by the *switched bond* introduced by Broenink and Wijbrans and is called *switchable structure bond* (SS-bond). The other one, called *generalised switched junction* (GSJ) *structure* is an extension of the standard 0- and 1-junctions and is inspired by the *Switched Power Junctions* (SPJs) introduced by Umarikar. It is shown that the switchable structure bond as well as the generalised switched junction structure can be implemented by means of standard BG elements and residual sinks. The latter ones in combination with a boolean modulated transformer were proposed by the Editor of this book. In the new BG structures proposed in Chap. 3 they ensure correct boundary conditions at power ports and avoid so-called dangling junctions. Chapter 3 demonstrates that the switched bond as well as the ideal switch introduced by Söderman can be considered special cases of the switched structure bond while the generalised switched junction encompasses the switched power junction as well as the controlled junction introduced by Mosterman and Biswas. Moreover, non-ideal switching can be captured if residual sinks are replaced by parasitic storage elements in combination with a resistor.

Chapter 4 adopts controlled junctions opposed to switches used in Chap. 5 for a bond graph representation of hybrid models. A local automaton associated with a controlled junction decides when it is on and when it is switched off, i.e. when a junction connects or disconnects model parts. That is, causality at the ports of a controlled junction depends on its discrete state. Causality changes at controlled junctions may affect the causality at storage or resistive ports. That is, preferred integral causality at a storage port may change into derivative causality. Accordingly, the authors of Chap. 4 distinguish between *static* and *dynamic* causality and visualise the latter one by assigning a causal stroke at one end of a bond and a dashed causal stroke at the other end. Storage elements with dynamic causality are assigned two pairs of effort and flow variables, i.e. there is a state variable if the storage element is in integral causality and a so-called pseudo-state variable if the element is in derivative causality. For the descriptor vector of state and pseudo-state variables, an implicit hybrid DAE can be derived from a BG that holds for all system modes of operation. The descriptor vector is pre-multiplied by a matrix with coefficients that are functions of the discrete states of controlled junctions. The BG representation of hybrid models proposed in Chap. 4 may be used for structural analysis as well as for simulation. The chapter is based on a PhD thesis of the first author that was supervised by the second author.

Chapter 1
Decomposition of Multiports

Peter C. Breedveld

1.1 Introduction

This chapter expects from its reader some basic port-based modeling knowledge and the ability to express these concepts in terms of bond graphs. This means that the reader is supposed to be familiar with the basics of the bond graph notation as explained, for instance, in [13, 16], in particular the nine basic node types C, I, Se, Sf, R, TF, GY, 0, and 1 (cf. Tables 1.1 and 1.2 show the block diagram expansions of their causal instantiations) and their multiport realizations.

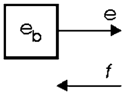
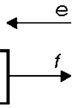
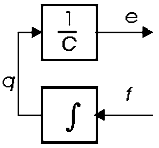
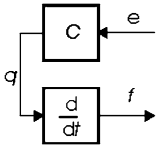
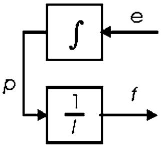
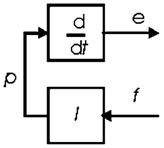
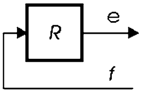
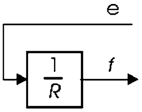
As a bond graph is a labeled di-graph, its structure is purely topological. This means that any configuration information of a model has to be added separately. This is the reason that bond graphs are often used for mechanical systems in linear motion and/or with fixed axis rotation, i.e., where the spatial configuration part is fixed, such that scalar variables are sufficient to describe the dynamic behavior. If planar or spatial effects need to be described, then the information about the chosen coordinate frames of the power variables has to be added to the model separately. In that case bond graphs still give an advantage due to the fact that velocity relations found from position relations can be used to find the relations between the conjugate forces without the need to use free-body diagrams, thus reducing the chance of sign errors.

Given that bonds in a bond graph represent bilateral relations between ports that describe specific concepts, there is thus no need for these concepts to be spatially separated. In other words: the interconnection structure of concepts in a model can itself be a mere conceptual structure without any meaning for the actual configuration. From this perspective we will show in the following by means of multiport decomposition [3, 4] how the nine basic concepts that are used in

P.C. Breedveld (✉)
University of Twente, Enschede, Netherlands
e-mail: p.c.breedveld@utwente.nl

W. Borutzky (ed.), *Bond Graphs for Modelling, Control and Fault Diagnosis of Engineering Systems*, DOI 10.1007/978-3-319-47434-2_1

Table 1.1 Basic one-port elements, linear block diagram implementation examples

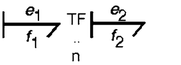
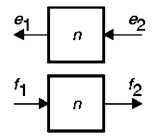
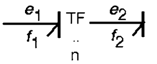
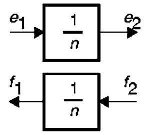
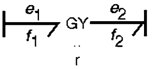
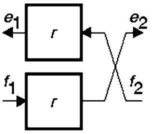
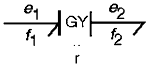
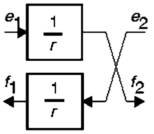
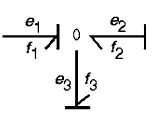
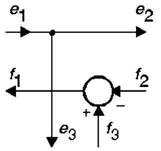
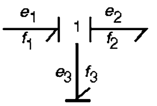
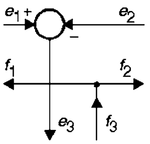
basic elements	block diagram expansion	basic elements	block diagram expansion
$e_b : S_e$ (e, f) $e = e_b$ 1		$f_b : S_f$ (e, f) $f = f_b$ 2	
integral causality $C : C$ (e, f) $e = \frac{1}{C}q$ $q = \int f dt + q(0)$ 3a		differential causality $C : C$ (e, f) $q = Ce$ $f = \frac{dq}{dt}$ 3b	
integral causality $I : I$ (e, f) $f = \frac{1}{I}p$ $p = \int e dt + p(0)$ 4a		differential causality $I : I$ (e, f) $p = If$ $e = \frac{dp}{dt}$ 4b	
$R : R$ (e, f) $e = Rf$ 5a		$R : R$ (e, f) $f = \frac{1}{R}\epsilon$ 5b	

a common bond graph follow from the temporal properties of a physical model, as well as the need to represent the influence of the environment, conceptual interconnection, and basic concepts like reversible and irreversible transduction (link between domains).

A decomposition of a multiport may have multiple purposes:

(1) Getting a better insight in the dynamic properties of a model, as loop gains of causal paths can be easily identified
(2) Getting a better insight in potential simplifications of a multiport due to resulting constitutive parameters that are negligible
(3) Being able to recognize how bond graph fragments can be composed into multiports or other types of basic elements, for instance, to eliminate causal loops
(4) Being able to use techniques for direct linear analysis on a bond graph, like Mason's rule, etc.
(5) Conversion of a bond graph containing multiports into a block diagram or a set of differential equations in state space form in a straightforward manner
(6) Being able to use the graphical input of a simulation package with a bond graph editor that does not support multiport elements

Table 1.2 Basic two- and multiport elements, with linear block diagram implementation examples

	basic elements	block diagram expansion		basic elements	block diagram expansion
6a	 $f_2 = nf_1 \quad e_1 = ne_2$		6b	 $f_1 = f_2/n \quad e_2 = e_1/n$	
7a	 $e_2 = rf_1 \quad e_1 = rf_2$		7b	 $f_2 = e_1/r \quad f_1 = e_2/r$	
8	 $e_2 = e_1$ $e_3 = e_1$ $f_1 = f_3 - f_2$		9	 $f_1 = f_3$ $f_2 = f_3$ $e_3 = e_1 - e_2$	

First the fundaments of a dynamic model are discussed: energy storage and positive entropy production. Multiport storage is analyzed on the basis of the properties of energy. Next it is discussed that interaction between stored quantities requires the concept of a power continuous multiport and how the relation to the environment can be represented. This approach is able to resolve the analogy paradox and after this is done, irreversible transduction is discussed and its relation to the paradoxal concept of "energy dissipation." Then the power continuous multiport is decomposed and it is motivated that a nonlinear, internally modulated transducer that is able to represent irreversible transduction deserves to be treated as a separate basic concept. Finally the canonical decompositions into basic elements are discussed.

1.2 Dynamic System Models of Multidomain Physical Systems: The Key Role of Time

The key concepts used in modeling the dynamic behavior of physical systems are the concepts of *state* and (rate of) *change* (of state). State and change are dialectic concepts: it is impossible to perceive what a state is if it can never change and at the same time it is impossible to understand change without the concept of state, i.e., some property that may change from one state into the other. In fact, all measurements of time are based on a repetitive change of some state, resulting in a time base.

A dynamic model is not only depending on time, but it should also be "transferable" in time, in other words: it should not lose its competence to describe relevant aspects of reality the moment it is made. This means that some property of the model of a system has to remain constant over time or the variations of this property should be completely determined by a change in the environmental influences on that system, commonly called its boundary conditions. This property that remains constant is called energy. This "transferability" in time of a model is called the *time translation symmetry* requirement and Noether [15] proved that all symmetries result in some form of a conservation principle that should hold for such a model, in this case *energy* conservation, also known as the first principle (or first "law") of thermodynamics. Thermodynamics is often wrongly considered as a mere theory of heat, but its approach of multiport storage of energy [9] can be considered the first systematic approach to multidomain physics, even if no heat or entropy is taken into account. This is why these two principles should not be limited to thermodynamics, but should be the basis of all models of physical systems: without at least one conserved quantity any model would lose its time translation symmetry and which harms its predictive or explaining value.

It is human experience that most (not all!) dynamic processes that we try to describe are characterized by "the arrow of time," i.e., if we would play a movie of the process backwards, it would appear unrealistic. This *time-reflection asymmetry* corresponds to the violation of a conservation principle, viz., that of entropy conservation and results in the *positive entropy production principle*, also called the second principle (or second "law") of thermodynamics. However, in order to properly describe the behavior of entropy, we also need to assign a state to it and consequently the concept of reversible storage. This also corresponds to the remark that not all processes seem irreversible: in some cases friction and other losses can disregarded and the entropy production can be considered to be approximately zero.

1.3 Multiport Storage

All sorts of symmetry principles lead to other conserved quantities, like momentum, angular momentum, electric charge, magnetic flux, and matter (in the incompressible case expressed in its volume or in the flexible case by a strain or displacement, like that of a spring) that can all serve to describe the state of a physical system in which these quantities play a role. When these conserved quantities, represented by a state vector $\mathbf{q}$, can be used to characterize the complete state of a system, there will always be an energy related to them: $E(\mathbf{q})$, which should be a first-degree homogenous function when the extensive states that can also serve as a boundary criterion, viz., amount of matter (moles) and/or available volume, even if they remain constant, are considered as states. This is why all energy density functions, i.e., energy functions in which the extent defining variable (volume or amount of matter) is set to unity and not considered as a state, are second-degree homogeneous and mostly quadratic. Kinetic energy, for example, is a first-degree homogenous function of momentum p and amount of moles N

$$E\left(\alpha p, \alpha N\right) = \frac{\alpha^2 p^2}{2M\alpha N} = \alpha^1 \frac{p^2}{2MN} = \alpha E\left(p, N\right) \tag{1.1}$$

where M stands for molar mass and α is an arbitrary parameter to check the degree of the homogenous function. However, the kinetic energy density (per unit mole) is

$$E\left(p, 1\right) = \varepsilon(p) \tag{1.2}$$

which is second-degree homogenous (in this case quadratic), because

$$\varepsilon\left(\alpha p\right) = \alpha^2 \frac{p^2}{2M} = \alpha^2 \varepsilon(p) \tag{1.3}$$

Obviously, if the amount of moles is assumed to remain constant and not considered a state, then the kinetic energy also appears to be second-degree homogeneous, even quadratic and, as explained later, this leads to a linear constitutive relation that is first-degree homogenous.

If a system is observed in which all these conserved quantities remain constant, i.e., $\mathrm{d}\mathbf{q}/\mathrm{d}t = 0$ and thus $\mathrm{d}\underline{E}/\mathrm{d}t = 0$, the system is said to be in equilibrium (or in static state or in stationary state) and thus not representing any dynamic behavior. In dynamic processes that are not influenced by their environment we commonly observe that they reach an equilibrium state after a certain period of time. The simplest situation of non-equilibrium is if the system can be thought to consist of two parts that are internally in equilibrium such that an energy can be assigned to each of them, while these part are not in equilibrium with respect to each other. After reaching the equilibrium state the conserved states variables of these two parts, including their energies, do not have to be equal in value. A criterion for equilibrium that will be explained later in more detail is that the rates of change of the energy with respect to the conserved state under consideration q_i have to be equal:

$$\frac{\partial E\left(\mathbf{q}\right)}{\partial q_i^1} = \frac{\partial E\left(\mathbf{q}\right)}{\partial q_i^2} \tag{1.4}$$

for all states q_i. Obviously, these partial derivatives are related to the total change of the energy in time, the power $P(t)$:

$$P(t) = \frac{\mathrm{d}E\left(\mathbf{q}(t)\right)}{\mathrm{d}t} = \sum_i \frac{\partial E\left(\mathbf{q}\right)}{\partial q_i}\frac{\mathrm{d}q_i}{\mathrm{d}t} = \sum_i e_i\left(\mathbf{q}(t)\right) f_i(t) = \mathbf{e}^{\mathrm{t}}\mathbf{f} \tag{1.5}$$

where $\frac{\partial E(\mathbf{q})}{\partial q_i} = e_i(\mathbf{q})$ is defined as a generalized effort, $\frac{\mathrm{d}q_i}{\mathrm{d}t} = f_i$ as a generalized flow and $\mathbf{e}$ and $\mathbf{f}$ are the column matrices of these efforts and flows, respectively, such that $\mathbf{e}^{\mathbf{t}}\mathbf{f}$ represents an inner product.

These generalized efforts are homogenous functions of all states themselves and 1° lower than the degree of the homogeneous energy function:

$$\frac{\partial E(\alpha\mathbf{q})}{\partial \alpha q_i} = \frac{\alpha^n \partial E(\mathbf{q})}{\alpha \partial q_i} = \alpha^{n-1} e_i(\mathbf{q}) \tag{1.6}$$

and form the constitutive relations of a multiport storage element, in a bond graph represented by a multiport C. Some authors use the terminology "C-field" instead of multiport C. As a "field" in physics also has a completely different meaning, it is not used herein to prevent confusion. As the stored energy in a multiport C is a conserved quantity, its mixed second derivatives should be equal

$$\frac{\partial^2 E(\mathbf{q})}{\partial q_i \partial q_j} = \frac{\partial^2 E(\mathbf{q})}{\partial q_j \partial q_i} \tag{1.7}$$

or

$$\frac{\partial e_i(\mathbf{q})}{\partial q_j} = \frac{\partial e_j(\mathbf{q})}{\partial q_i} \tag{1.8}$$

In other words: the Jacobian of $\mathbf{e}(\mathbf{q}(t))$ has to be symmetric according to the principle of energy conservation. This is called Maxwell reciprocity or Maxwell symmetry.

1.4 Resolution of an Analogy Paradox

Most (bond graph) modelers will immediately make the objection that the above definitions of effort and flow are not in line with what they consider to be an effort and a flow: they, together with many other engineers and scientists, consider a force, for example, to be an effort, while a force can indeed be not only the partial derivative of a potential energy with respect to a displacement (deformation of matter or distance in a gravitational field), but also the rate of change of a conserved quantity, viz., the momentum. Others have used the latter argument to consider a force as a flow, leading to an everlasting analogy discussion that highly complicates insight and education [10, 19]. This is why the attribute "generalized" is used: in the generalized bond graphs introduced in [3–5] a force is the generalized effort of the potential domain and the generalized flow of the kinetic domain, while the coupling between these domains is made explicit. In order to come up with one choice for the combination of the kinetic and the potential domains, the so-called mechanical domain, the role of effort and flow in one of these domains has to be inverted, in other words: this domain has to be dualized, i.e., the roles of effort and flow are interchanged. Obviously, either one of the two domains can be dualized, which causes the just described analogy paradox if the dualization remains implicit. A unit gyrator can be considered an explicit “dualizer” as it equates the flow of each port to the effort of the other port. However, the coupling between the kinetic and the potential domain can only under a certain condition be represented by a dualizing coupling, a unit gyrator called symplectic gyrator [1], viz., the condition that the system is described with respect to an inertial frame, because only then Newton’s second law holds. Likewise, a voltage is considered the generalized effort of the electric domain, i.e., the partial derivative of the energy with respect to the electric charge, and the generalized flow of the magnetic domain, i.e., the rate of change of magnetic flux linkage to a number of windings. In this case the additional assumptions that are required to be able to link the electric and the magnetic domains in this way are even more obvious: the rotation operations in Maxwell’s equations are only reduced to a simple gyrating relation when the system is assumed quasi-stationary, in other words: when an electric circuit may be assumed not to exchange energy by electromagnetic radiation which would require a third port of this coupling with an irreversible nature. In retrospect, we can also see the quasi-stationary requirement as an additional assumption for the coupling between the potential and the kinetic domain: such a mechanical system also has to be in a quasi-stationary state as it should not radiate acoustic energy. In both cases the radiated power would violate the power continuity of the symplectic gyrator that is assumed to be the interdomain coupling.

1.5 Properties of an Irreversible Transducer

Returning to the system that transcends from non-equilibrium to equilibrium (Fig. 1.1) shows why the generalized effort can be called an *equilibrium-determining variable*: the generalized efforts of the two parts (subsystems) have to be equal when they are in equilibrium. Likewise, the generalized flow can be characterized as an *equilibrium-establishing variable*: there will be a generalized flow between the two subsystems until this flow goes to zero in equilibrium. It is important to note that there is only one flow due to the flow continuity required by the conservation principle that corresponds to a conserved state. Note that this reasoning does not exclude cross-coupling: if the effort difference at one R-type

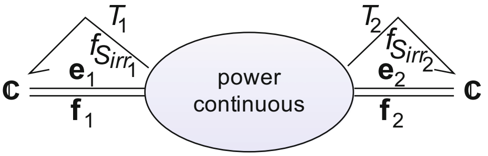


Fig. 1.1 Power continuity of an irreversible transducer due to the energy conservation principle (note that the orientation of the multibonds is not defined yet!)

port of a multiport irreversible transducer is zero, its flow may still be non-zero due to a non-zero effort difference at another R-type port. However, if all effort difference are zero, all flows are zero.

We thus see that the non-equilibrium situation of each of the domains is characterized by one generalized flow, out of one subsystem and into the other, so corresponding with rates of change of the respective conserved states that have an opposite sign (Fig. 1.1), while two generalized efforts with different values are converging to the same value. In other words: there has to be a relation between these efforts and the flow between the subsystems. We also know that during such a process entropy has to be generated in principle and that this entropy is also one of the states. The production can thus be represented by a generalized flow and its conjugate effort is the partial derivative of the energy with respect to the entropy which is known as an absolute temperature $\frac{\partial E}{\partial S} = T$ [9]. Due to the global energy conservation principle and the fact that the concept describing this irreversible process cannot store energy as this would be a "contradictio in terminis," the concept describing the transition from a non-equilibrium into an equilibrium state should be power continuous. If we just consider an irreversible transducer with one entropy producing port which will be labeled as an S-type port, because it act as an entropy source and one R-type port for reasons of simplicity:

$$-e_1 f_1 - e_2 f_2 - T f_{S_{irr}} = 0 \tag{1.9}$$

or, as $f_1 = -f_2$ due to flow continuity:

$$(e_1 - e_2) f_2 = T f_{S_{irr}} \geq 0 \tag{1.10}$$

As the absolute temperature T is positive and the entropy production is zero or positive we can conclude that the generalized effort difference $e_1 - e_2 = \Delta e$ has to have the same sign as the generalized flow f_2, in other words: the generalized flow is always directed from a higher generalized effort to a lower generalized effort, except for cross-coupling effects. At this point it is important to recognize that some efforts, like voltage, have a relative reference, while others, like absolute temperature, pressure, force, or chemical potential, have an absolute reference. In case of a relative reference the flow can only depend on the effort difference and this results in a property called "nodicity" [17]. In non-nodic models like chemical reaction networks the flow can be a function in which the individual values of the efforts are required: this means that such a relation has to be described by an element with at least 2 R-type ports with a nonlinear constitutive relation for which an additional flow continuity constraint holds.

The above conclusion that the generalized effort difference $e_1 - e_2 = \Delta e$ has to have the same sign as the generalized flow f_2 can also be formulated as follows: the relation between this generalized flow and this generalized effort difference has to lie in the first and third quadrant and thus has to pass through the origin. This explains why the equality of the generalized efforts (the partial derivatives of the energy with respect to their states) is an equilibrium criterion as mentioned before. This means that the slope of this relation, which is commonly called a resistance, is positive in the origin, but may have any value outside of the origin, which explains why the concept of a negative differential resistance does not violate the principle of positive entropy production.

Relation (1.10) can also be written for all n states:

$$\sum_{i=1}^{n}(e_{1_i}-e_{2_i})f_{2_i}=\sum_{i=1}^{n}\Delta e_i f_{2_i}=\Delta\mathbf{e}^t\mathbf{f}=Tf_{S_{irr}}\geq 0 \tag{1.11}$$

We can also conclude that the power port that represents the irreversibly produced thermal power cannot have a linear constitutive relation, even when all other ports are linear and can be characterized by a resistance matrix R, because

$$f_{S_{irr}}=\frac{\Delta\mathbf{e}^t\mathbf{f}}{T}=\frac{\mathbf{f}^t\mathbf{R}^t\mathbf{f}}{T}\geq 0 \tag{1.12}$$

As only the symmetric part of the resistance matrix contributes to this quadratic form:

$$f_{S_{irr}}=\frac{\mathbf{f}^t\mathbf{R}_s\mathbf{f}}{T}\geq 0 \tag{1.13}$$

and

$$\mathbf{f}^t\mathbf{R}_a\mathbf{f}=0 \tag{1.14}$$

where $\mathbf{R}_s=\frac{\mathbf{R}+\mathbf{R}^t}{2}$ and $\mathbf{R}_a=\frac{\mathbf{R}-\mathbf{R}^t}{2}$, such that

$$\mathbf{R}=\frac{\mathbf{R}+\mathbf{R}^t}{2}+\frac{\mathbf{R}-\mathbf{R}^t}{2}=\mathbf{R}_s+\mathbf{R}_a \tag{1.15}$$

The symmetry of the part of the relations that contributes to the entropy production is called Onsager symmetry. This symmetry relates the Peltier effect to the Seebeck effect, for example [9]. The other part can be identified as a multiport gyrator that is power continuous as a result of (1.14) and of which we have seen that it contains the dualizing interdomain couplings described earlier.

1.6 Influence of the Environment

It was not discussed yet that energy can also be exchanged with as well as stored in the environment of a system. Typical for the concept of an environment of a system is that there is no information available about its extent: the influence of conserved properties that are exchanged with the environment on the behavior of the system is neglected because no information about these extensive quantities is available: the state of the environment can only be characterized by an intensive state, i.e., a generalized effort that can be considered independent of its conjugate generalized flow. If this is not a valid assumption, the system boundary has to be changed such that it is valid. The environment can thus be seen as a storage element that is so large with respect to the storage in the system, that it can be considered infinitely large, such that the rate of change of extensive state does not influence the intensive state in a relevant manner. Such an element that represents a boundary condition is a generalized effort source Se. In contrast to a storage element, for which modulation would mean a violation of the energy conservation principle, an effort source can be modulated (MSe). When modulated storage elements are used after all, this implies the implicit assumption that the energy exchange due to modulation is at all times negligible compared to the energy exchange (power) through each of its ports.

If the energy storage of a system is described by a (multiport) storage element and the influence of the environment by effort sources, then all other elements have to be power continuous as a result of the energy conservation principle. The common concept of dissipation, which seems to violate this statement, does not refer to energy, but to one of its Legendre transforms [9, 18], viz., the free energy $F\left(\tilde{\mathbf{q}}, T\right) = E\left(\tilde{\mathbf{q}}, S\right) - TS$, where $\tilde{\mathbf{q}}$ is the state vector of all states $\mathbf{q}$ from which the entropy S has been eliminated. Furthermore,

$$\mathrm{d}F\left(\tilde{\mathbf{q}}, T\right) = \sum_i \frac{\partial E\left(\tilde{\mathbf{q}}, S\right)}{\partial q_i} \mathrm{d}q_i - S\mathrm{d}T = \sum_i e_i\left(\tilde{\mathbf{q}}, T\right) \mathrm{d}q_i - S\mathrm{d}T \tag{1.16}$$

which means that the contribution from the thermal domain to a change in free energy is zero when the temperature is constant. If free energy is assumed to be dissipated in a (multiport) resistor, we thus intrinsically assume that interaction with the thermal domain does not lead to additional dynamic properties of the model, even though temperature variations may influence the system via an activated bond, i.e., modulation by the temperature. In those cases the thermal port is not included in the model. This is the only way to obtain linear models, as the model that describes the entropy production port necessarily has a nonlinear constitutive equation, as we will confirm later. The solvability of linear models explains the "popularity" of models in which the above assumption about the thermal domain is made and where all energy is in fact free energy that can be dissipated.

1.7 Co-Energy

The quadratic representation of the kinetic energy of a constant amount of mass discussed earlier is *equal in value* to the sign inverse of its Legendre transform with respect to the momentum, i.e., $-L[E(p)]_p$, which is a so-called co-energy $E^*(v)$, a function in which the momentum is replaced by the velocity as function argument:

$$-L[E(p)]_p = -\left(\frac{p^2}{2m} - \frac{\mathrm{d}E}{\mathrm{d}p}p\right) = vp - \frac{p^2}{2m} = mv^2 - \frac{1}{2}mv^2 = \frac{1}{2}mv^2 = E^*(v) \tag{1.17}$$

Although the values of kinetic energy and kinetic co-energy are the same, their arguments are not: the kinetic energy is a function of a conserved quantity, the momentum, an extensive variable, while the kinetic co-energy is a function of an equilibrium determining variable (generalized effort), the velocity. This distinction is important when energy functions are used as generating function of constitutive relations of multiports. For example, if we return to the first-degree homogeneous form

$$E^*(v,N) = vp - \frac{p^2}{2MN} = MNv^2 - \frac{1}{2}MNv^2 = \frac{1}{2}MNv^2 = E^*(v,N) \tag{1.18}$$

If this co-energy is used as an energy-like generating function, one obtains for the total material potential

$$\frac{\partial E^*(v,N)}{\partial N} = \frac{\partial}{\partial N}\frac{1}{2}MNv^2 = \frac{1}{2}Mv^2 \neq \mu \tag{1.19}$$

because

$$\mu = \frac{\partial E(p,N)}{\partial N} = \frac{\partial}{\partial N}\frac{p^2}{2MN} = -\frac{p^2}{2MN^2} = -\frac{1}{2}Mv^2 \tag{1.20}$$

In other words: the wrong sign is obtained.

This is often not seen, due to the fact that for variable mass with constant density the available volume is used as a state instead of the amount of moles

$$E^*(v,V) = vp - \frac{p^2}{2\rho V} = \rho Vv^2 - \frac{1}{2}\rho Vv^2 = \frac{1}{2}\rho Vv^2 = E^*(v,V) \tag{1.21}$$

$$\frac{\partial E^*(v,V)}{\partial V} = \frac{\partial}{\partial V}\frac{1}{2}\rho Vv^2 = \frac{1}{2}\rho v^2 = p_{\text{dynamic}} \tag{1.22}$$

However, for a pressure, also for a dynamic pressure, an increase in energy corresponds to a decrease in volume, so:

$$p_{\text{dynamic}} = \frac{\partial E(p,V)}{-\partial V} = -\frac{\partial}{\partial V}\frac{p^2}{2\rho V} = \frac{p^2}{2\rho V^2} = \frac{1}{2}\rho v^2 \tag{1.23}$$

In this case the sign error due to the mix up of energy and co-energy is compensated by a second sign error in the partial derivative.

We return to a generic system for which true energy conservation holds and note that all other concepts that are used to describe relations between storage ports have to be multiports that satisfy energy conservation without storage, which means that they have to be power continuous. If we consider an arbitrary power continuous multiport, it may have mixed causality, in other words its constitutive relation may have the form:

$$\begin{bmatrix} \mathbf{e}_1 \\ \mathbf{f}_2 \end{bmatrix} = \Psi\left(\begin{bmatrix} \mathbf{f}_1 \\ \mathbf{e}_2 \end{bmatrix}\right) \tag{1.24}$$

where stands for Ψ an arbitrary function. Hogan and Fasse [11] demonstrated by means of scattering variables that such a power continuous constitutive relation can only be of a multiplicative form, i.e.,

$$\begin{bmatrix} \mathbf{e}_1 \\ \mathbf{f}_2 \end{bmatrix} = \mathbf{J}(.)\begin{bmatrix} \mathbf{f}_1 \\ \mathbf{e}_2 \end{bmatrix} \tag{1.25}$$

where the matrix **J** can still depend on any variable.

As a result of power continuity

$$P = \begin{bmatrix} \mathbf{e}_1^t & \mathbf{f}_2^t \end{bmatrix}\begin{bmatrix} \mathbf{f}_1 \\ \mathbf{e}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1^t & \mathbf{e}_2^t \end{bmatrix}\mathbf{J}^t\begin{bmatrix} \mathbf{f}_1 \\ \mathbf{e}_2 \end{bmatrix} = 0 \tag{1.26}$$

We obtain

$$\mathbf{J} = -\mathbf{J}^{\mathbf{t}} \tag{1.27}$$

At this point we see that we have identified sufficient bond graph concepts to describe dynamic behavior of a system that also interacts with its environment: multiport storages that can interact via a power continuous structure with each other and with the environment that is represented by effort sources (infinite storage) as shown in Fig. 1.2, i.e., three basic concepts. However, given the choice of positive orientation (towards the multiports except for the sources), it turns out that some sign changes would have to be made on the fly that require additional concepts (Fig. 1.3). An experienced bond graph modeler knows that the concept of a junction is needed for this and this is why decomposition of the power continuous connection structure described by (1.25) is discussed next. The need for this additional junction is what is not recognized in the so-called port-Hamiltonian approach, where all

Fig. 1.2 Power continuous interaction between environment and system

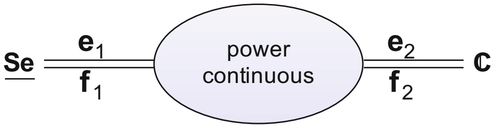
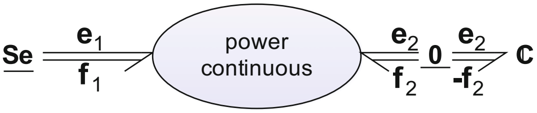


Fig. 1.3 Positive orientation conventions require a 0-junction



orientations to a power continuous structure are chosen positive inward, all positive orientations of multiport storage and dissipation are chosen outward (resulting in an uncommon minus sign in the constitutive relations, but when junction structures have to be combined this results in an anomaly, because junctions have not been defined separately, which may result in error prone sign manipulation at the equation level.

1.8 Decomposition of the Power Continuous Multiport

A quadratic form of a matrix is only zero if it is skew symmetric, cf. (1.27), so it can be decomposed as follows:

$$\begin{bmatrix} \mathbf{e_1} \\ \mathbf{f_2} \end{bmatrix} = \mathbf{J} \begin{bmatrix} \mathbf{f_1} \\ \mathbf{e_2} \end{bmatrix} = \begin{bmatrix} \mathbf{G_e} & \mathbf{T^t} \\ -\mathbf{T} & \mathbf{G_f} \end{bmatrix} \begin{bmatrix} \mathbf{f_1} \\ \mathbf{e_2} \end{bmatrix} \tag{1.28}$$

where

$$\begin{aligned} \mathbf{G_e} &= -\mathbf{G_e^t} \\ \mathbf{G_f} &= -\mathbf{G_f^t} \end{aligned} \tag{1.29}$$

Hogan and Fasse [11] also showed that if we add the property of port-symmetry to a power continuous multiport, there are only two solutions. These solutions appear to be linear without a parameter for all domains where energy is a useful concept: the 0-junction and the 1-junction. Given that these two types of multiports are ideal connectors, these are the only elements that do not need a fixed positive orientation of their ports to guarantee unambiguous use, as a change of orientation is merely a sign change in the summation of either the flows or the efforts, respectively. Using these junctions (in array form [4]) it can be seen (cf. Fig. 1.4) in a straightforward manner that an arbitrary power continuous multiport can be decomposed into two (modulated) multiport gyrators, one with effort-out causality and characterized by the matrix $\mathbf{G_e}$ and one with flow-out causality and characterized by the matrix $\mathbf{G_f}$, as well as a (modulated) multiport transformer characterized by the matrix $\mathbf{T}$. In turn, it can be seen by inspection of the operations of a matrix multiplication, that these multiports can be decomposed into junctions and 2-port (modulated) gyrators and transformers: the multiport gyrator characterized by $\mathbf{G_e}$ by connecting a 1-junction

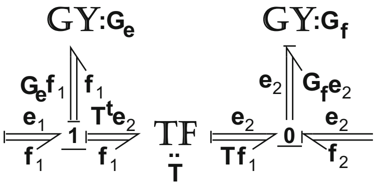


Fig. 1.4 Generic decomposition of a power continuous multiport (**TF** and **GY** may be modulated)

to each port and next connecting each of these ports to all other ports by 2-port gyrators, the multiport gyrator characterized by the matrix $\mathbf{G_f}$ by connecting a 0-junction to each port and next connecting each of these ports to all other ports by 2-port gyrators and finally the multiport transformer characterized by the matrix $\mathbf{T}$ by connecting a 1-junction to each port with effort-out causality and a 0-junction to each port with flow-out causality and next connecting each 1-junction to all 0-junctions by means of 2-port transformers. In this way each 2-port gyrator or 2-port transformer has one of the independent matrix elements of the matrix $\mathbf{J}$ as ratio. As discussed in more detail in [3, 4], these are canonical, immediate decompositions.

Note that a gyrators and transformers, except for unit transformers that are in fact equal to bonds, cannot exist inside one domain, as they would violate generalized flow continuity: generalized flows are the rates of change of the conserved quantity that characterizes the domain and neither a transformer nor a gyrator are flow continuous. In other words: they are interdomain couplings. When the objection is made that transformers in mechanical systems seem to violate this reasoning, one should realize that each coordinate in a mechanical model should be considered a separate domain, because a bond graph cannot represent configuration properties: the coordinate frames have to be represented separately and are linked to the scalar values that are used in the topological representation that a bond graph is.

In [3, 4] it was shown that a linear multiport storage element can be decomposed in many ways using a multiport transformer by making use of the congruence properties of its constitutive matrix with a diagonal matrix. However, there is only one congruence decomposition that is canonical, i.e., in which the number of independent parameters of the decomposition is equal to the number of independent parameters of the multiport storage element: the transformation matrix is triangular, such that less 2-port transformers are required. Just like the immediate decompositions of the multiport gyrator and transformer that were just discussed (Fig. 1.4) and that appeared to be canonical, a linear multiport storage element also has such an immediate decomposition: in preferred integral causality all ports are connected to a 1-junction and next each 1-junction is connected to all others via a 0-junction. Finally each junction is connected to a 1-port C. In differential causality the junction types have to be interchanged.

Just like a multiport C, a multiport gyrator also has a congruence decomposition as any skew symmetric matrix is congruent with a block-diagonal matrix with 2×2 symplectic matrices as blocks.

So far we have identified that all models in which energy is a relevant variable can be constructed from 1-port storage elements (C), 1-port effort sources (Se), 2-port (modulated) transformers and gyrators, and 0- and 1-junctions, i.e., 6 basic elements. Given that a unit gyrator can be combined with a 1-port element, dual storage elements (I) and dual sources (Sf) can also be defined, but one should be aware that the asymmetry is then lost between effort and flow as equilibrium-determining and equilibrium-establishing variable, respectively. We then have identified eight basic elements and bond graph modelers will notice that the resistor is still missing: an R-element contradicts energy conservation and can only be used if the influence of the thermal domain on the dynamic properties of a system can be disregarded.

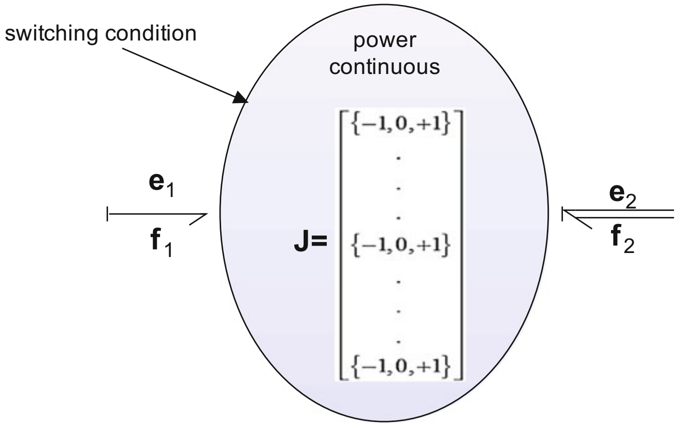

switching condition
power
continuous
e_1
f_1
J=
$\begin{bmatrix}\{-1,0,+1\}\\ \vdots \\ \{-1,0,+1\}\\ \vdots \\ \{-1,0,+1\}\end{bmatrix}$
e_2
f_2

Fig. 1.5 X0-junction as special case of a power continuous multiport (transformer)

Before resolving this paradox two additional basic concepts are added that are in principle not needed, but of such a high conceptual value that they deserve a special representation: the switched junctions. Switched junctions (X0 and X1) represent that the possibility of energy exchange may depend on the state of a system, the easiest example being a bouncing object [7]. As a junction can be considered a power continuous multiport (that is even non-mixing, so in fact a multiport transformer) with a special form of the transformation matrix **J** in (1.25), viz., a column matrix with elements that can only take the values -1 or $+1$, the energy exchange can be switched off by setting all the matrix elements to zero, i.e., by modulating it by a Boolean as a function of some condition that commonly depends on the state of a system (Fig. 1.5). However, since the possibility of modulation of elements is not considered to increase the number of basic elements, we will stick to the common number of nine basic elements.

1.9 Representation of Irreversibility

The elements we have identified so far as needed for a consistent description of energetic behavior (C, I, (M)Se, (M)Sf, (M)TF, (M)GY, (X)0, and (X)1) are all reversible, such that irreversibility due to the second principle of thermodynamics does not seem present, although we have seen that it may be part of a nonlinear power continuous junction structure. This implies the need for a 2-port with a resistive port and an entropy producing port that is power continuous. Such an RS element has indeed been defined, but causes a paradox with Hogan's and Fasse's [11] result that all power continuous 2-ports are either modulated gyrators or modulated transformers. This paradox is resolved by rewriting the constitutive relations of an RS in such a way that they represent a port-modulated transformer or gyrator, respectively, depending on the causality:

$$\text{RS}:\begin{cases} e = e(f) \\ f_{S_{irr}} = \frac{ef}{T} \end{cases} \tag{1.30}$$

$$\text{MTF}:\begin{cases} f_{S_{irr}} = \left(\frac{e(f)}{T}\right) f \\ e = \left(\frac{e(f)}{T}\right) T \end{cases} \tag{1.31}$$

$$\text{RS}:\begin{cases} f = f(e) \\ f_{S_{irr}} = \frac{ef}{T} \end{cases} \tag{1.32}$$

$$\text{MGY}:\begin{cases} f_{S_{irr}} = \left(\frac{f(e)}{T}\right) e \\ f = \left(\frac{f(e)}{T}\right) T \end{cases} \tag{1.33}$$

In principle, an effort-out causality of the entropy producing port is also possible, but not shown, because this is a rare possibility. As a consequence, a separate RS is in principle not needed, but such a representation would mean that a key property of dynamic system, viz., irreversibility, is not represented by a separate conceptual element symbol, but remains more or less hidden in a specific way to modulate an element that is reversible in its unmodulated form. Therefore, the RS is added as the ninth element and this fundamentally nonlinear element can only be linearized by reducing the influence of the thermal domain to an imposed temperature, such that no dynamic interaction with the thermal domain takes place and entropy does not need to be modeled. Such a change of causality for the multiport storage corresponds to a Legendre transform of its stored energy with respect to the entropy. As we have seen, this Legendre transform $F = E–TS$ is called the free energy, which can be considered to be dissipated in a resistor (R). By doing this the entropy-maximum principle for nonlinear models is transformed into the more common (free!) energy-minimum principle that also applies to linear models.

Obviously, a 1-port R can be generalized into a multiport resistor characterized by a constitutive matrix or a set of interdependent nonlinear constitutive relations with a constitutive Jacobian. In order to decompose it we will have to consider its properties first.

1.10 Decomposition into Basic Elements

The above has made clear that when multiports are assumed to have linear constitutive relations, decomposition in terms of junctions and the remaining seven basic one- and two-elements (cf. Tables 1.1 and 1.2) is possible. As there are many possibilities to do this, the concept of a canonical decomposition was introduced, which means that the decomposition does require the same number of independent constitutive parameters as the original multiport [3, 4]. Two types of canonical decomposition are possible, the so-called congruence canonical decompositions that are based on the congruence properties of the constitutive matrix and immediate canonical decompositions that follow from direct inspection of the constitutive relations. For 2-ports it has been shown that these two types are exhaustive which strongly suggests that this is also the case for arbitrary multiports [6]. In [3, 4] it was also shown that decomposition of most elements with nonlinear constitutive relations can also be realized by means of internal modulation, with the exception of nonlinear multiport storage elements, as this would introduce multiport artifacts that need to compensate the gyristors that would be caused by the modulation, which would destroy the simplifying nature of a decomposition.

Multiport transformers only have immediate decompositions: their two conjugate constitutive relations consist of a multiplication of a flow vector by a matrix resulting in another flow vector and a multiplication of an effort vector by the transpose of that matrix resulting in another effort vector. A matrix multiplication can be decomposed into multiplications and summations: the multiplications are represented by 2-port transformers with the multiplying matrix element as transformation ratio and the summations are represented by 0-junctions and 1-junctions, respectively, as shown in (Fig. 1.6).

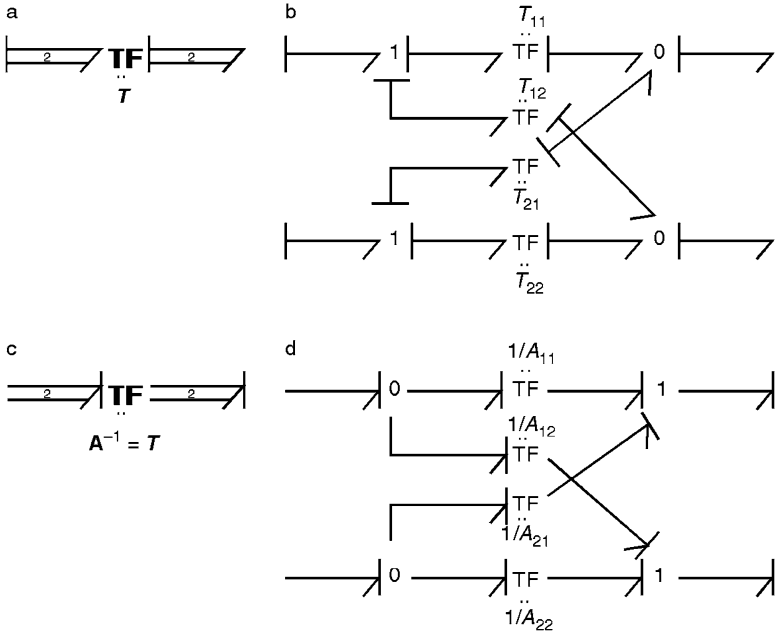


Fig. 1.6 Immediate decomposition of a multiport transformer

A linear multiport resistor or linear multiport storage element is characterized by a symmetric matrix. If such a multiport is connected to a multiport transformer, it can be combined into new multiport element characterized by a matrix $\tilde{\mathbf{C}}$ that equals the inverse original matrix **C** pre-multiplied by the transpose of the transformation matrix and post-multiplied by the transformation matrix, resulting again in a symmetric matrix:

$$\begin{aligned}
&\mathbf{T}^{\mathbf{t}}\mathbf{C}^{-1}\mathbf{T} = \tilde{\mathbf{C}}^{-1}\\
&\tilde{\mathbf{C}} = \left(\mathbf{T}^{\mathbf{t}}\mathbf{C}^{-1}\mathbf{T}\right)^{-1} = \mathbf{T}^{-1}\mathbf{C}\mathbf{T}^{-\mathbf{t}}\\
&\tilde{\mathbf{C}}^{t} = \left(\mathbf{T}^{-1}\mathbf{C}\mathbf{T}^{-\mathbf{t}}\right)^{\mathbf{t}} = \mathbf{T}^{-1}\mathbf{C}^{-1}\mathbf{T}^{-\mathbf{t}} = \mathbf{T}^{-1}\mathbf{C}\mathbf{T}^{-\mathbf{t}} = \tilde{\mathbf{C}}
\end{aligned} \qquad (1.34)$$

When we combine this result with the property of a symmetric matrix that it is congruent with a diagonal matrix, it is obvious that a combination of a multiport transformers and an array of uncoupled one-ports storage elements or resistive elements can represent a linear multiport storage element or a linear multiport resistive element, respectively (Fig. 1.7).

However, such a linear multiport has $n + \left(n^2 - n\right)/2 = \left(n^2 + n\right)/2$ independent parameters, while a combination of a multiport transformers and an array of one-ports has $n^2 + n$ parameters, such that $\left(n^2 + n\right)/2$ of these parameters are dependent. However, congruence of a symmetric matrix with a unit matrix can be realized by a triangular matrix, which has $\left(n^2 - n\right)/2$ zero elements and all n diagonal elements of a unit matrix are unity, such that $\left(n^2 + n\right)/2$ independent parameters remain.

If the diagonal elements of such a triangular matrix are set to unity and the unit matrix is replaced by an arbitrary diagonal matrix, a combination of n one-port elements with a multiport transformer with $\left(n^2 + n\right)/2$ independent parameters is obtained, which is decomposable by immediate decomposition (Fig. 1.7), hence a

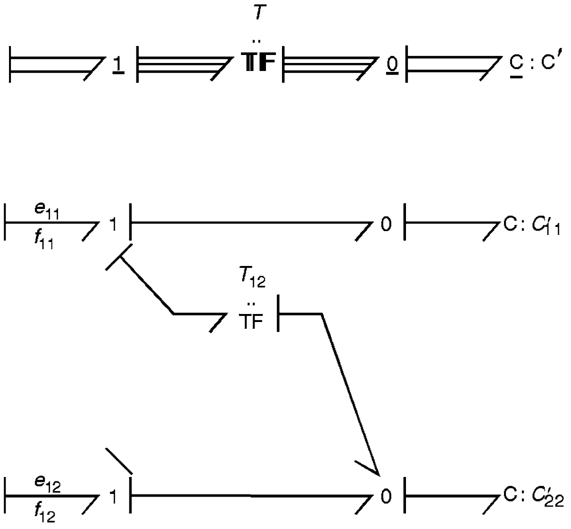


Fig. 1.7 (De)composition of a multiport, C respectively, into a multiport C (2-port shown for simplicity)

canonical decomposition. However, one should keep in mind that the decomposition of a multiport transformer depends on its causality: causal inversion of the decomposed form leads to algebraic loops.

Immediate decompositions of multiport storage or resistive elements are also possible by rewriting the matrix relations. For example, for two-ports this results in the so-called Π- and T-type decompositions, depending on causality. For storage elements the disadvantage of this canonical decomposition is that the number of stored quantities and initial conditions (three in case of a two-port) becomes larger than the number of independent states and initial conditions of the multiport (two in case of a two-port). More details can be found in [3, 4], but what is important here is that all linear multiports are decomposable into the basic 1-ports and 2-ports in a junction structure of bonds and 0- and 1-junctions.

1.11 Conclusion

The nine common basic elements of bond graphs were shown to result from the fundamental principles of physics and many implicit assumptions that are often not considered. It will improve the modeling process, when a modeler is more aware that these assumptions may not hold. The use of the bond graph notation already helps a modeler when the grammar rules and semantics of this graphical language are respected. The additional awareness that a bond graph stems from multiport storage, effort sources to represent the environment and a power continuous multiport that may also contain irreversible transduction, may improve the modeling process even further. For example, using the concept of the generalized domain in a generalized bond graph notation makes clear that the required flow continuity only allows the use of C, (M)Se, (X)0, (X)1, and thermal (M)RS elements inside the domain, while (M)TF, M(GY), and other (M)RS elements are interdomain couplings. If the thermal domain is excluded and power is considered a flow of free energy, then C, (M)Se, (X)0, (X)1, and R are intradomain elements. Obviously, multiport C and (M)R can serve as interdomain couplings too.

A central role plays the resistive multiport: if seen as a generalization of a simple 1-port R we have seen that the generic relation between an effort and a flow vector also generates a part that does not contribute to the entropy production, viz., the skew-symmetric part of its Jacobian, which can be represented by a multiport gyrator. By contrast, if we start from the power continuity that also should hold for irreversible transducers, we saw that an RS can in principle be replaced by a modulated transformer or gyrator. One could say that the first approach is kind of bottom up from the common approach that uses free energy that can be dissipated, while the second approach starts from the energy conservation principle. The first approach has the advantage that we can recognize that close to equilibrium of domains with an absolute reference no gyrator can exist, such that processes only have the nature of relaxation, while for from equilibrium the Jacobian of nonlinear

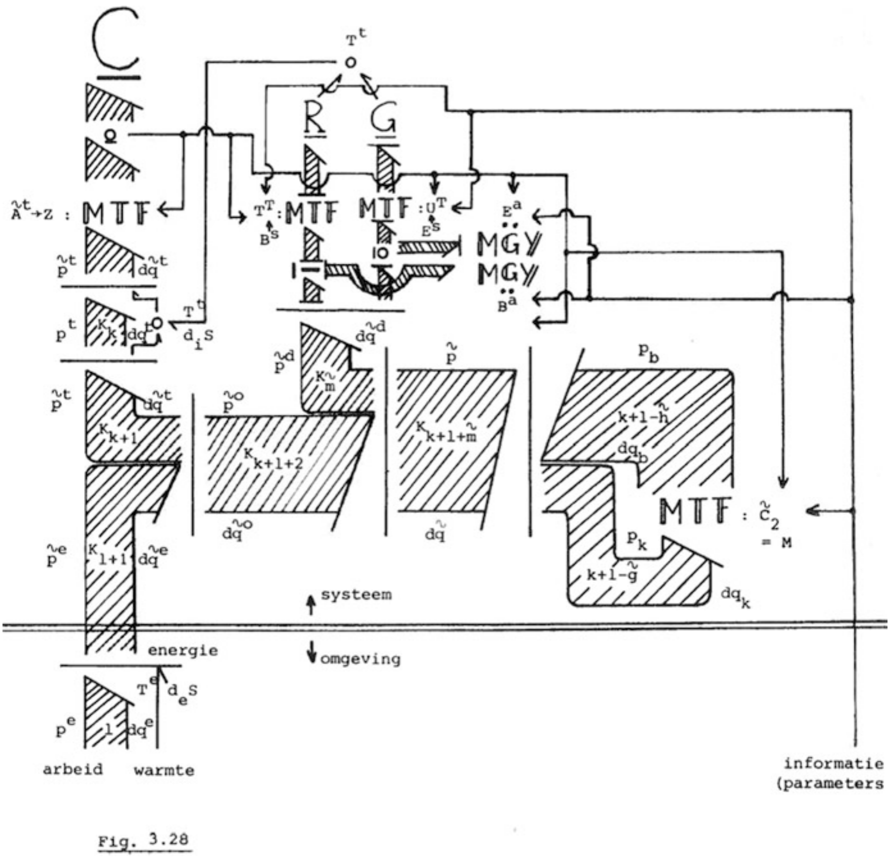


Fig. 1.8 Historic [1] multibond decomposition (except MTF and MGY)

relation may result in gyrating couplings that, for instance, can explain chemical oscillations or the use of a transistor to create an oscillator while biased into a non-equilibrium operating point.

Finally, a picture from the author's master thesis [1] (Fig. 1.8: ignore the Dutch and the uncommon choice of some symbols, like the G referring to a resistive port in conductive causality, no addition of an S to the symbol of an irreversible transducer and a direct sum definition that allowed a sign change) is shown to demonstrate that a multibond representation of a generic decomposition (except for the MTF's and MGY's of which immediate decomposition is straightforward) has been available for over 35 years. The rather cryptic symbols for effort and flow variables, bond dimensions, and constitutive matrices are the result of the fact that this multibond graph tried to represent and extend an analysis by [12] of the power continuous conceptual structure in thermodynamic systems. Obviously, it was not recognized at that time that the RS is just a part of the power continuous interconnection structure and, as such, a nonlinear "constraint."

References

1. Breedveld, P. C. (1979). *Irreversibele Thermodynamica en Bondgrafen: EEN synthese met enige praktische toepassingen (in Dutch)*. Master thesis, 1241.2149, University of Twente, Enschede, Netherlands.
2. Breedveld, P. C. (1982). Thermodynamic bond graphs and the problem of thermal inertance. *Journal of Franklin Institute, 314*(1), 15–40. doi:10.1016/0016-0032(82)90050-3.
3. Breedveld, P. C. (1984). *Physical systems theory in terms of bond graphs*. Ph.D. thesis, University of Twente, Faculty of Electrical engineering, ISBN 90-9000599-4.
4. Breedveld, P. C. (1984). Decomposition of multiport elements in a revised multibond graph notation. *Journal of the Franklin Institute, 318*(4), 253–273. doi:10.1016/0016-0032(84)90014-0. ISSN 0016-0032.
5. Breedveld, P. C. (1985). Multibond graph elements in physical systems theory. *Journal of the Franklin Institute, 319*(1–2), 1–36. doi:10.1016/0016-0032(85)90062-6. ISSN 0016-0032.
6. Breedveld, P. C. (1995) Exhaustive decompositions of linear two-ports. In: In F. E. Cellier & J. J. Granda (Eds.), *Proceedings of the SCS 1995 International Conference on Bond Graph Modeling and Simulation (ICBGM'95), SCS Simulation Series*, *27*(1), 11–16, Las Vegas, 15–18 January, ISBN 1-56555-037-4.
7. Breedveld, P. C. (1996). The context-dependent trade-off between conceptual and computational complexity illustrated by the modeling and simulation of colliding objects. In P. Borne et al. (Eds.), *Proceedings of the Computational Engineering in Systems Applications '96 IMACS/IEEE-SMC Multiconf.*, 48–54, Lille, France, 9–12 July 1996, Late Papers.
8. Breedveld, P. C. (1999). On state-event constructs in physical system dynamics modeling. *Simulation Practice and Theory, 7*(5–6), 463–480. doi:10.1016/S0928-4869(99)00017-8. ISSN 0928-4869.
9. Callen, H. B. (1985). *Thermodynamics and an introduction to thermostatistics*. New York, NY: Wiley.
10. Firestone, F. A. (1933). A new analogy between mechanical and electrical systems. *The Journal of the Acoustical Society of America, 4*, 249–267.
11. Hogan, N., & Fasse, E. D. (1988). Conservation principles and bond-graph junction structures. In L. R. C. Rosenberg & R. Redfield (Eds.), *Automated modeling for design* (pp. 9–13). New York, NY: ASME.
12. Jongschaap, R. J. J. (1978). About the role of constraints in the linear relaxational behaviour of thermodynamic systems. *Physica A: Statistical Mechanics and Its Applications, 94*(3–4), 521–544. doi:10.1016/0378-4371(78)90085-7.
13. Karnopp, D. C., Margolis, D. L., & Rosenberg, R. C. (1990). *System dynamics: A unified approach*. New York, NY: Wiley.
14. Maschke, B. M., van der Schaft, A. J., & Breedveld, P. C. (1992). An intrinsic hamiltonian formulation of network dynamics: Non-standard poisson structure and gyrators. *Journal of the Franklin Institute, 329*(5), 923–966. doi:10.1016/S0016-0032(92)90049-M. ISSN 0016-0032.
15. Noether, E. (1918). *Invariante varlationsprobleme*. Nachr. D. König, Gesellsch. D. Wiss, Zu Göttingen, Math-phys. Klasse, pp. 235–257. English translation Travel M. A. (1971) *Transport Theory and Statistical Physics 1*(3), 183–207.
16. Paynter, H. M. (1961). *Analysis and design of engineering systems*. Cambridge, MA: MIT.
17. Paynter, H. M. (1972). The dynamics and control of eulerian turbomachines. *J Dynamic Systems Measurement and Control, 94*(3), 198–205.
18. Strang, G. (1986). *Introduction to applied mathematics*. Wellesley, MA: Wellesley-Cambridge.
19. Trent, H. M. (1954). Isomorphisms between oriented linear graphs and lumped physical systems. *The Journal of the Acoustical Society of America, 27*, 500–527.

Chapter 2
A Method for Minimizing the Set of Equations in Bond Graph Systems with Causal Loops

Jesus Felez

Abbreviations

BDF	Backward differential formulae
DAE	Differential algebraic equations
EJS	Eulerian junction structure
GYS	Gyristor
MBG	Multi-bond graph
MBS	Multibody systems
MGY	Modulated gyrator
MTF	Modulated transformer
ODE	Ordinary differential equations
ZC	Zero compliance
ZCP	Zero-order causal path

2.1 Introduction

The objective of this paper is to establish an improved formulation able to be used in real time simulations. Simulation requires integrating the dynamic equations. The size, complexity, and type of these equations are fundamental in order to reduce the simulation time if real time is to be achieved. This problem is very important in the real time simulation of mechanical systems, such as in a complex multibody model of a vehicle used in a driving simulator, where a great number of differential equations and constraint equations are used to build the model in a systematic way.

J. Felez (✉)
Mechanical Engineering Department, Universidad Politécnica de Madrid, Madrid, Spain
e-mail: jesus.felez@upm.es

W. Borutzky (ed.), *Bond Graphs for Modelling, Control and Fault Diagnosis of Engineering Systems*, DOI 10.1007/978-3-319-47434-2_2

In recent years, the generation of dynamic equations of systems modeled with bond graphs has been the topic of considerable research. These equations have been presented in a variety of forms. Classical formulations express these equations in terms of a large number of flows associated with inertances and displacements associated with compliances. In order to obtain these equations it is necessary to establish the causality in the model. Several procedures have been developed to achieve this. Sequential Causality Augmentation Procedure (SCAP) [10] has been traditionally the way to implement causality in bond graph models. Causal cycles are a consequence of the problems raised from this implementation. They appear when the SCAP procedure drives to indeterminacy in the application of causality. The following step is the appearance either of a causal cycle or a causal violation [17, 18]. How to deal with these causal problems has been an important aim in the past [8]. Karnopp and Margolis [11] contributed with the stiff compliance approach, where high stiffness compliances are introduced in the model to eliminate the causal loops between integral and derivative causal storage ports. In this case, a set of differential equations is obtained, including as variables the flows associated with the previous inertances, displacements associated with the previous compliances, and the corresponding variables associated with the new stiff elements introduced, seriously increasing the number of differential equations to be solved. Usually, this approach needs the use of special numerical solvers for stiff differential equations, requiring very small integration step times. For these reasons, this procedure is not appropriate for simulation in real time.

Another approach was the introduction of Lagrange multipliers in the model [2, 5, 20]. Some other authors [7] introduce residual sinks and sources in a similar solution to the Lagrange approach. When Lagrange multipliers are used to break causal paths, a set of differential algebraic equations is obtained, composed of a number of differential equations equal to the number of inertances plus compliances, and a number of algebraic (or constraint) equations equal to the number of Lagrange multipliers. The number of equations is the same as with the previous method, it also being inappropriate for real time. Nevertheless, this approach presents important formulation advantages that must be considered.

Previous methods prevent the existence of causal loops, but it is necessary to pre-analyze the model and to modify the model in a subsequent operation. Another approach to solve the problem [3, 4] is the use of break variables to open the causal loops. Causal loops always present an algebraic character. Algebraic loops relate their internal variables by means of algebraic relationships. It means that these loops do not involve integration operations. This fact leads to the definition of Zero-order Causal Paths (ZCPs) [4, 17]. The mathematical model obtained from bond graphs with ZCPs and opened with break variables is also a differential algebraic equation set (DAE).

This last approach reduces the number of equations to a number equal to the number of inertances plus compliances plus break variables.

The use of Lagrange multipliers or the introduction of break variables implies the necessity of new variables and the appearance of constraint equations, considerably increasing the number of involved variables and the number of equations to be solved. These procedures are very useful to obtain the system equations in a systematic way, but they have to be improved or modified for their use in real time simulation.

Another approach is to obtain the dynamic equations in terms of a minimum set of variables, related with the degrees of freedom of the system. There are several developed procedures, mainly in multibody analysis. The most popular method is Kane's formulation [9], which uses a minimum number of generalized velocities. The numerical integration of these equations is much more efficient than the previous one because it needs only a reduced number of equations. However, simplicity in the formulation and the ease of manipulation of dynamic equations are lost with this method.

The method presented in this paper is based on a velocity transformation process used in multibody dynamics [12, 13, 19], extending the methodology to the bond graph specificities. This method has the advantages of the simplicity in its formulation and the computational efficiency of Kane's method. In this procedure state variables can be established as in the classical bond graph formulation. Next, according to the system causality and the ZCPs class 1 that appear in the system, a set of independent state variables are chosen obtaining the relationships between the dependent and the independent ones. The dynamic equations will then be transformed into a reduced set. The resultant equations are equal in number to the number of degrees of freedom of the system.

This paper discusses the procedure applied to each type of ZCP and the way to manipulate the causal assignments in order to obtain the minimal set of differential equations. In this way, no algebraic equations appear in the set of equations obtained and only a small number of differential equations appear (ODEs corresponding to the system's degrees of freedom).

The way to solve ZCPs class 2, 3, and 4 is based on the idea to assign derivative causality to the storage elements to eliminate the causal loops [15]. Nevertheless in class 2, 3, and 4, except class with unity gain, the break variables can be eliminated and the equations reduced and written without any break variable and constraint equations, but this elimination must be done by hand and no automatic procedures can be used. For this reason the procedures shown in this paper have been developed in order to systematize the way to obtain the system equations in an optimized form, with the aim of implementing them in a computer program. In Ref. [14] algorithms are shown along with the procedures for symbolically obtaining the transformations needed for the different cases put forward.

Sueur and Dauphin-Tanguy [16] describe the set of conditions for the method when applying a derivative causality for linear systems.

An interesting reference is presented in [1] where Borutzky considers an extension of the bond graph methodology showing its application for multibody systems, presenting also an interesting formulation of a reduced set of equations of motion.

The paper is based on the previous work of Felez et al. [6]. Some examples are presented to illustrate the different cases. These examples are very simple and their only aim is to show the procedure for each ZCP. For each example, a comparison between the use of break variables and the reduced formulation proposed has been done, showing the differences between both cases.

2.2 Zero-Order Causal Paths

A causal path is a sequence of causal bonds between two vertices of the bond graph. The causal strokes must be attached to each bond at the "same relative position."

A ZCP is a causal path with topological loops whose variables are related to themselves by means of algebraic assignments [17].

The classification of ZCPs originates from the research work carried out by Van Dijk and Breedveld [17] and was completed by Cacho et al. [4]. The specific procedure to break each ZPC type is shown in detail in [4].

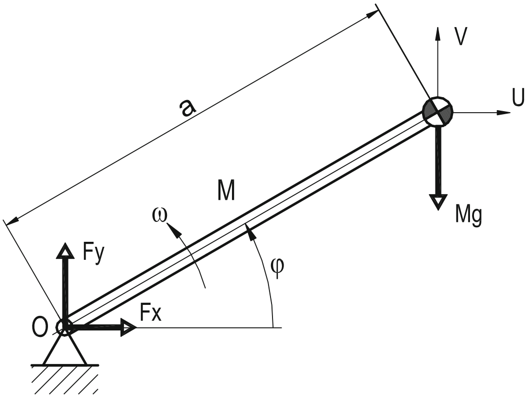
2.3 ZCP Class 1

ZCP class 1 is set between storage ports with integral causality and storage ports with differential causality.

The proposed approach is based on a velocity transformation and needs the use of Lagrange multipliers in order to break the causal loops. Once the Lagrange multipliers have been introduced, the equations obtained are simplified and reduced by means of the velocity transformation.

This procedure is shown in the following example (Fig. 2.1). Figure 2.1 represents a single pendulum composed of a bar of inertia *J* with respect to the center of mass and, additionally, a concentrated mass *M* at the same point. Its movement is defined by three displacements (x, y, φ) and their corresponding velocities (u, v, ω).

Fig. 2.1 A single pendulum



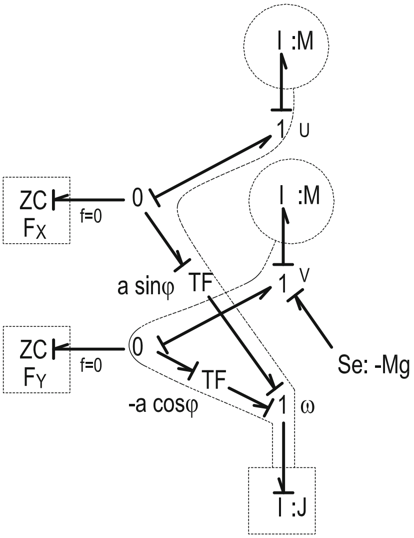


Fig. 2.2 Bond graph of a system with a ZCP class 1

Figure 2.2 represents the bond graph corresponding to the single pendulum. Elements ZC [5] represent the introduction of two Lagrange multipliers in the constrained displacements of the hinge point *O*. In order to obtain the relationships between dependent velocities and the independent one, in the bonds associated with ZC elements flow causality is established, with the meaning that the associated velocity in this case is equal to zero. In this bond graph, two ZCPs class 1 appear, relating the independent velocity ω with the dependent ones u and v.

Following the previous described procedure, it is possible to express the velocities according to the only independent one:

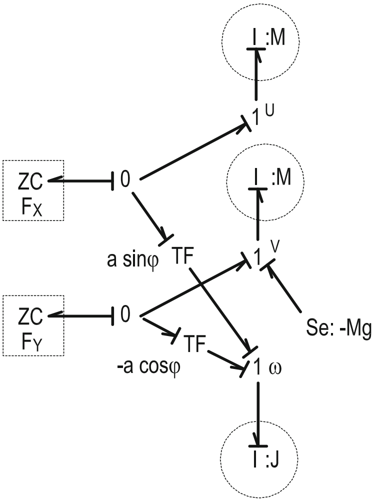
$$\omega = \dot{\varphi} \tag{2.1a}$$

$$\begin{bmatrix} \omega \\ u \\ v \end{bmatrix} = S\,[\omega] = \begin{bmatrix} 1 \\ -a\ \sin\varphi \\ a\ \cos\varphi \end{bmatrix} [\omega] \tag{2.1b}$$

$$\begin{bmatrix} \dot{\omega} \\ \dot{u} \\ \dot{v} \end{bmatrix} = S\left[\dot{\omega}\right] + \dot{S}\left[\omega^2\right] = \begin{bmatrix} 1 \\ -a\ \sin\varphi \\ a\ \cos\varphi \end{bmatrix} \left[\dot{\omega}\right] + \begin{bmatrix} 0 \\ -a\ \cos\varphi \\ -a\ \sin\varphi \end{bmatrix} \left[\omega^2\right] \tag{2.1c}$$

Figure 2.3 presents the causality assignment established in the bond graph in order to obtain its dynamic equations. Due to the fact that elements ZC represent the introduction of two Lagrange multipliers, with the meaning of constraint forces, two new variables, F_x and F_y, are introduced by these two elements.

Fig. 2.3 Introduction of Lagrange multipliers



With this causality assignment, ZCPs class 1 disappears and the dynamic equations result in three differential and two algebraic equations:

$$\begin{bmatrix} J & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & M \end{bmatrix} \begin{bmatrix} \dot{\omega} \\ \dot{u} \\ \dot{v} \end{bmatrix} + \begin{bmatrix} -a\ \sin\varphi & a\cos\varphi \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} F_X \\ F_Y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\text{Mg} \end{bmatrix} \tag{2.2a}$$

$$\begin{aligned} u + a\omega\ \sin\varphi &= 0 \\ v - a\omega\ \cos\varphi &= 0 \end{aligned} \tag{2.2b}$$

But these equations can be reduced to only one using the previously described method. Then pre-multiplying (2.2a) by $\left[\,1,\,-a\ \sin\varphi,\,a\ \cos\varphi\,\right]$:

$$\left[J,\,-\text{Ma}\sin\varphi\ \text{Ma}\cos\varphi\right] \begin{bmatrix} \dot{\omega} \\ \dot{u} \\ \dot{v} \end{bmatrix} + \left[0\ 0\right] \begin{bmatrix} F_X \\ F_Y \end{bmatrix} = -\text{Mga}\cos\varphi \tag{2.3}$$

Substituting in (2.3) the acceleration vector by its expression (2.1c):

$$\begin{aligned} &\left[J\ -\text{Ma}\sin\varphi\ \text{Ma}\cos\varphi\right] \begin{bmatrix} 1 \\ -a\sin\varphi \\ a\cos\varphi \end{bmatrix} \left[\dot{\omega}\right] + \\ &+\left[J\ -\text{Ma}\sin\varphi\ \text{Ma}\cos\varphi\right] \begin{bmatrix} 0 \\ -a\cos\varphi \\ -a\sin\varphi \end{bmatrix} \left[\omega^2\right] = -\text{Mga}\cos\varphi \end{aligned} \tag{2.4}$$

And finally:

$$\begin{cases} \left[J + \mathrm{Ma}^2\right]\left[\dot{\omega}\right] = -\mathrm{Mga}\cos\varphi \\ \omega = \dot{\varphi} \end{cases} \tag{2.5}$$

2.4 ZCP Class 2

The causal path is set between elements whose constitutive relations are algebraic (resistors are the most typical case). There are also several ways to obtain the set of equations. The first one is the introduction of Lagrange multipliers in order to change the causality in the R elements, introducing then a new variable that can be used as break variable. The second one is the use of a variable (flow or effort, depending on the case) in the R element as break variable (Fig. 2.4). In both cases, the system is increased with a number of variables (multipliers in the first one and break variables in the second one) and a number of algebraic equations equal to the number of ZCPs class 2 presented. The new proposed procedure consists in the change of the causality of an energetic element (C or I) in order to invert the causality in one R element and to eliminate the causal path (Fig. 2.5). The following example illustrates the procedure.

If we use break variables, if integral causality is established in the inertance, a ZCP class 2 appears between resistances (Fig. 2.4), it then being necessary to establish as break variable effort F_1 for the resistance R_1.

The dynamic equation and the constraint equation established are

$$\begin{bmatrix} m & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \dot{v} \\ \dot{F}_1 \end{bmatrix} + \begin{bmatrix} 0 & -1 \\ 1 & \dfrac{1}{R_1} + \dfrac{1}{R_2} \end{bmatrix}\begin{bmatrix} v \\ F_1 \end{bmatrix} = \begin{bmatrix} \mathrm{Se} \\ \mathrm{Sf} \end{bmatrix} \tag{2.6}$$

On the other hand, if differential causality is established in the inertance (Fig. 2.5), the causality of this inertia eliminates the causal path between resistances.

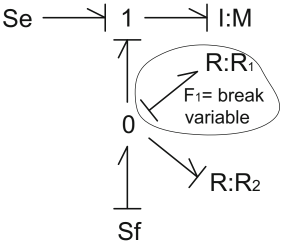


Fig. 2.4 System with a ZCP class 2. Use of a break variable

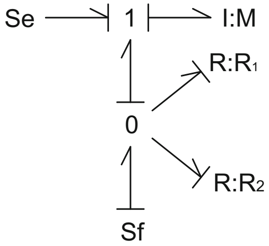


Fig. 2.5 System with a ZCP class 2 and inertance with derivative causality

Then, a single equation is obtained and the break variable and the algebraic equation disappear:

$$v = -m\left(\frac{1}{R_1} + \frac{1}{R_2}\right)\dot{v} + \mathrm{Sf} + \mathrm{Se}\left(\frac{1}{R_1} + \frac{1}{R_2}\right) \tag{2.7}$$

This can be reorganized as follows:

$$m\dot{v} + \frac{R_1 R_2}{R_1 + R_2}v = \mathrm{Se} + \frac{R_1 R_2}{R_1 + R_2}\mathrm{Sf} \tag{2.8}$$

For nonlinear systems, the proposed method is not applicable when the R elements among which the ZCP is produced do not let the flow variable be explicitly obtained from the effort variable using an equation of the type:

$$f = e/R \tag{2.9}$$

where the R coefficient may be non-constant, but never a function of the state variables.

2.5 ZCP Class 3

ZCP class 3 is a causal cycle whose topological loops are open (only one of the two variables, effort or flow, associated with each bond is used). The causal path starts and ends in the same port. Each ZCP class 3 needs one break variable and introduces one algebraic equation.

The way to solve both ZCP classes 3 and 4 is similar. It is also possible to add the new variables (multipliers or break variables), both being equivalent. If a Lagrange multiplier is introduced, this implies that a new variable is introduced in the system, and this new variable can be considered as break variable of the topological loop because the Lagrange multiplier does not change the causality in the loop. The proposed new procedure implies the establishment of derivative causality in inertance or compliance ports connected to the path, working the corresponding co-energetic variable as break variable.

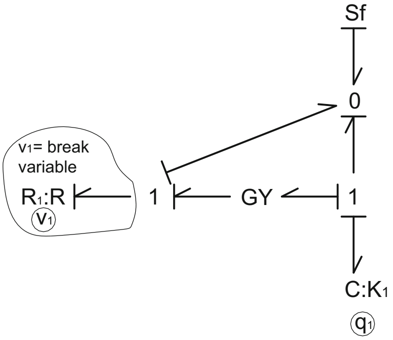


Fig. 2.6 System with a ZCP class 3 and a break variable

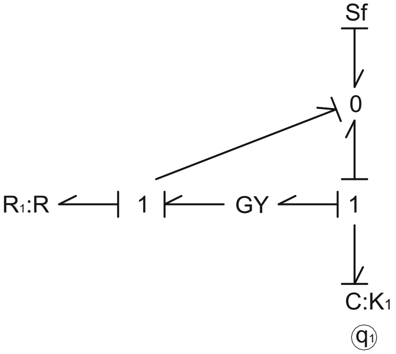


Fig. 2.7 System with a ZCP class 3 and a compliance with derivative causality

In Fig. 2.6, a break variable is established. If integral causality is considered in the compliance element, a causal path appears. In order to break this, flow v_1 in resistance R_1 is introduced as break variable.

The dynamic equation and the constraint equation established are

$$\begin{cases} \dot{q}_1 = -v_1 - \mathrm{Sf} \\ R_1 v_1 = -\mathrm{GSf} + K_1 q_1 \end{cases} \tag{2.10}$$

If differential causality is imposed on the compliance (Fig. 2.7), the causal path disappears since flow q_1, which imposes compliance, is used as break variable.

The following expression is obtained:

$$K_1 q_1 = -R_1 \dot{q}_1 - R_1 \mathrm{Sf} + \mathrm{GSf} \tag{2.11}$$

That can be reorganized as:

$$\dot{q}_1 + \frac{K_1}{R_1} q_1 = \frac{\mathrm{GSf}}{R_1} - \mathrm{Sf} \tag{2.12}$$

For nonlinear systems, as in the ZCP class 2, the proposed method is not applicable when the elements among which the ZCP is produced do not let the derivative of the state variable to be explicitly obtained.

2.6 ZCP Class 4

ZCP class 4 is a causal cycle whose topological loops are closed. Each ZCP class 4 needs two break variables (usually a flow and an effort), and introduces two algebraic equations.

The following example illustrates the case of ZCP class 4. Two break variables are introduced (the flow v_1 and the effort e_1) (Fig. 2.8).

The dynamic equations and the constraint equations established, with break variables, are

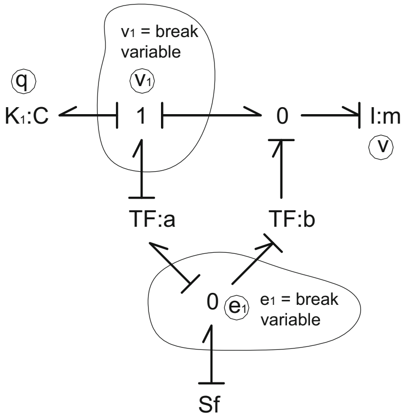
$$\begin{cases} m\dot{v} = \dfrac{e_1}{b} \\ \dot{q} = v_1 \\ \dfrac{v_1}{a} = Sf - \dfrac{(v - v_1)}{b} \\ \dfrac{e_1}{a} = K_1 q + \dfrac{e_1}{b} \end{cases} \tag{2.13}$$

If differential causality is established on the inertance and the compliance (Fig. 2.9), the two causal paths disappear and no additional break variables are needed, since flow $\dot{q}$ and $m\dot{v}$ effort associated with the compliance and the inertance, respectively, act as break variables.

The equations are then as follows:

$$\begin{cases} v = \left(1 - {}^{b}\!/_{a}\right) \dot{q} + b\mathrm{Sf} \\ K_1 q = -\left(1 - {}^{b}\!/_{a}\right) m\dot{v} \end{cases} \tag{2.14}$$

Fig. 2.8 System with a ZCP class 4 and two break variables



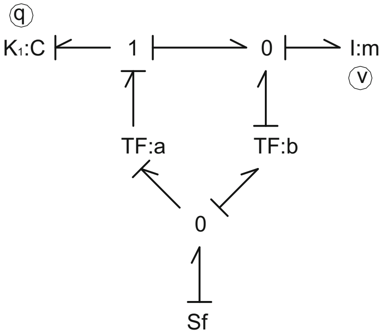


Fig. 2.9 System with a ZCP class 4

Then:

$$\begin{bmatrix} m\left(1-{}^{b}/_{a}\right) & 0 \\ 0 & -\left(1-{}^{b}/_{a}\right) \end{bmatrix}\begin{bmatrix} \dot{v} \\ \dot{q} \end{bmatrix}+\begin{bmatrix} 0 & K_1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} v \\ q \end{bmatrix}=\begin{bmatrix} 0 \\ b\mathrm{Sf} \end{bmatrix} \tag{2.15}$$

Although the example showed corresponds to a linear system, the proposed method is applicable when the TF or GY variables involved in the ZCP are nonlinear (*a* and *b* in the case of example are the constitutive relations of the transformer), with the only possible constraint that in the final equations a singular position can appear (like a division by 0 as example), and in this case, a change of the causality direction in all the closed loop could be necessary.

The procedure is valid when constitutive relations of TF and GY are functions of the state variables, for instance, but is not valid if they are a function of the derivatives of the state variables.

2.7 ZCP with a Loop-Gain Equal to 1

Figure 2.10 represents a ZCP class 4 with a loop-gain equal to 1. To open the loop, two break variables *e* and *f* are needed as can be seen in Fig. 2.10.

The equations obtained are as follows:

$$\begin{cases} \dot{q}_1 = f \\ \dot{q}_2 = f - v_1 \\ m_1\dot{v}_1 = K_1q_1 + K_2q_2 + e - \mathrm{Se} \\ m_2\dot{v}_2 = -K_1q_1 - e + \mathrm{Se} \\ K_1q_1 + K_2q_2 - \mathrm{Se} = 0 \\ v_1 - v_2 = 0 \end{cases} \tag{2.16}$$

Fig. 2.10 System with a ZCP class 4 and loop-gain equal to 1 with two break variables

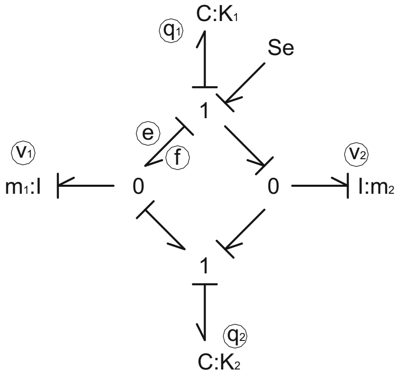
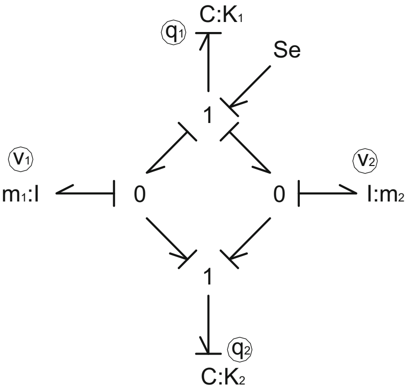


Fig. 2.11 System with a ZCP class 4 and loop-gain equal to 1



The problem with these equations is that they are differential algebraic equations of order 2 that cannot be solved numerically if they are not derived and the nilpotency index reduced.

However, if the proposed method of inverting the causality of elements *C* and *I* is used by assigning them differential causality, the ZCP class 4 disappears (Fig. 2.11).

The following equations are obtained:

$$\begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & m_1 & m_2 \\ 0 & 0 & m_1 & m_2 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{v}_1 \\ \dot{v}_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ K_1 & 0 & 0 & 0 \\ 0 & -K_2 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \text{Se} \\ 0 \end{bmatrix} \tag{2.17}$$

In this case, these equations can be simplified because if the first two rows are summed, it is obtained that:

$$v_1 = v_2 \tag{2.18}$$

and (2.17) can be reduced to:

$$\begin{bmatrix} 2m_1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{v}_2 \\ \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & -K_2 \\ -1 & 0 & 0 \\ 0 & K_1 & K_2 \end{bmatrix} \begin{bmatrix} v_2 \\ q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \mathrm{Se} \end{bmatrix} \tag{2.19}$$

The above Eq. (2.19) are differential algebraic equations and do not present any additional break variables.

2.8 Combinations of Several ZCPs

The objective of this section is to show the application of the proposed method when several ZCPs simultaneously appear. The idea is quite simple. First step is to apply a procedure to assign causality like the shown in [1, 4] developed by Felez et al. to detect the different ZCPs that appear in the model. In this step, when ZCPs are identified, causality is assigned and creak variables are used to solve and break the different identified ZCPs.

Because the aim of this work is to show the way of applying the proposed method, very simple and academic examples are proposed. The focus is not the physical interpretation of the bond graph models. The main focus is to show how, with a bond graph structure that presents combination of ZCPs, the proposed method can obtain a more reduced set of equations by eliminating break variables when it is possible.

Let's analyze two different cases. In both cases, several ZCPs simultaneously appear, several storage elements, *I* and *C*, appear and for each case, equations obtained by using break variables and preferred causality for *I* and *C* elements are showed in comparison with the proposed method that inverts the causality of *I* and *C* in order to reduce the number of equations.

Equations are obtained with the assumption of linear behavior, but the procedure can be applied in the same way for nonlinear components behavior, taking into account that in this case several causality constraints may be taken into account.

2.8.1 Combination Between Two ZCPs Class 3

Figure 2.12 shows an example that includes one *C* element (energy storage element) and two *R* elements. If causality assignment procedure is applied, *C* element is assigned with preferred causality and two break variables, f_1 and f_2, are needed in order to solve the two ZCPs class 3 that appear.

Expression (2.20) shows the obtained equations. They are composed by a set of three equations. This first one is a differential one, related with the *C* element, and the last two ones are algebraic equations related with the two *R* elements and the two class 3 ZCPs. Variables in the set of equations are q, f_1, and f_2.

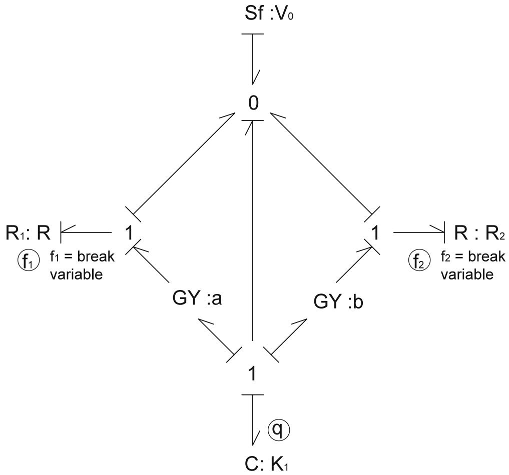


Fig. 2.12 System with combination of class 2 and 3 ZCPs

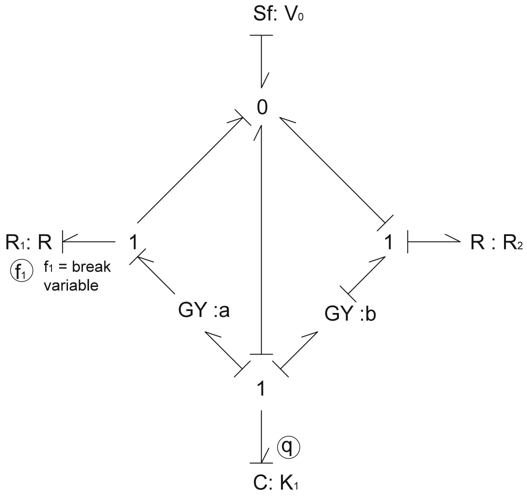


Fig. 2.13 Opening procedure for a system with combination of class 2 and 3 ZCPs

$$\begin{cases} \dot{q} = \mathrm{Sf} = f_1 - f_2 \\ R_1 f_1 = a\mathrm{Sf} + K_1 q + (b-a) f_2 \\ R_2 f_2 = b\mathrm{Sf} + K_1 q + (a-b) f_1 \end{cases} \tag{2.20}$$

Figure 2.13 shows the application of the proposed method for this case. If causality in the C element is inverted, the C element imposes its associated effort $K_1 q$. But, because of the presence of only one energy storage element, only one of the two original ZCPs can be broken, and one break variable is needed to solve the causality assignment. This break variable is f_1.

Expression (2.21) shows the obtained equations. I this case they are composed by a set of two equations. This first one is an algebraic equations, related with the C element where the effort $K_1 q$ is imposed, and the other equation is an n equation

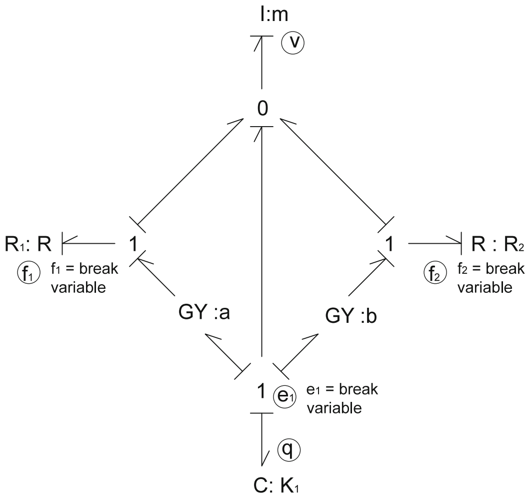


Fig. 2.14 System with class 2 and 3 ZCPs

obtained when imposing the break variable, but that is differential due to the flow associated with the *C* element. Variables in the set of equations are q and f_1. In conclusion, originally one *C* element and two break variables (three equations), and with the application of the proposed method, the causality inversion in the *C* element eliminates one break variable, so only two equations are obtained.

$$\begin{cases} K_1 q = (B - a) f_1 - b\mathrm{Sf} \\ R_1 f_1 - (a - b)\,\dot{q} + R_2\,(\mathrm{Sf} - f_1) \end{cases} \tag{2.21}$$

Let's show other example, similar to the previous one, but with two energy storage elements, one *C* and one *I*. Figure 2.14 shows the example. If causality assignment is applied, *C* and *I* elements are assigned with preferred causality and two break variables, f_1 and f_2, are needed in order to solve the two ZCPs class 3 that also appear.

Expression (2.22) shows the obtained equations. They are composed by a set of four equations. Two are differential equations, related with the *C* and *I* elements, and the other two are algebraic equations related with the *R* elements and the two class 3 ZCPs. Variables in the set of equations are q, v, f_1, and f_2.

$$\begin{cases} \dot{q} = v - f_1 - f_2 \\ m\dot{v} = -kq - af_1 - bf_2 \\ R_1 f_1 = av + Kq + (b - a) f_2 \\ R_2 f_2 = bv + Kq + (a - b) f_1 \end{cases} \tag{2.22}$$

If causality in the *C* and *I* elements are inverted, the *C* element imposes its associated effort $K_1 q$ and the *I* element imposes its associated flow v (Fig. 2.15).

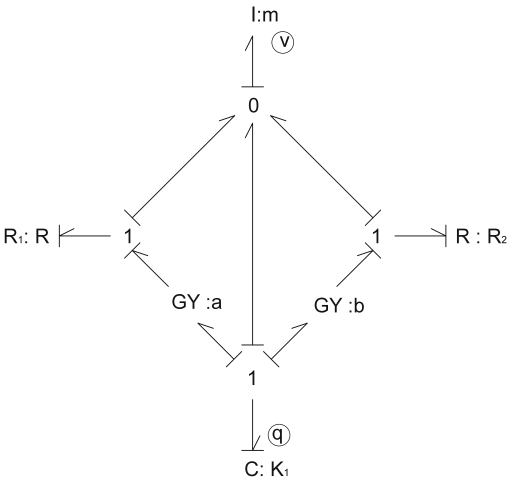


Fig. 2.15 Causal paths opening procedure

The presence of two energy storage elements produces the elimination of the two previous ZCPs, obtaining in consequence only two differential equations, and no one algebraic Eq. (2.23). So it can be concluded that the causality inversion of one energy storage element produces the elimination of one ZCP. In this particular case, two energy storage elements eliminate two ZCPs.

$$\begin{cases} Kq = -\left(m - {}^{a}\!/_{R_1} - {}^{b}\!/_{R_2}\right)\dot{v} - \left(a^2 - b^2\right)\dot{q} \\ v = \left(1 + {}^{a}\!/_{R_1} + {}^{b}\!/_{R_2}\right)\dot{q} - \left({}^{1}\!/_{R_1} + {}^{1}\!/_{R_2}\right)m\dot{v} \end{cases} \tag{2.23}$$

2.8.2 Combination Between ZCPs Class 2 and 4

Let's show other examples where combination of different ZCPs also appears. In this case, combination between class 2 and class 4 is presented, as shown in Fig. 2.16. The program presents one *I* element and two resistances. If preferred causality is applied to the *I* element, two break variables are needed. The first one e_2 is associated with the class 2 causal path, and the second one, e_1, is associated with the closed loop presented in the right hand of the bong graph in Fig. 2.16.

Expression (2.24) shows the obtained equations. They are composed by a set of three equations. This first one is a differential one, related with the *I* element, and the last two ones are algebraic equations related with the ZCPs. Variables in the set of equations are *v*, e_1, and e_2.

$$\begin{cases} m\dot{v} = e_1 + \mathrm{Se} \\ \dfrac{e_2}{R_2} = (e_2 - e_1)\left(\dfrac{1}{R_1} + \dfrac{1}{R_3}\right) + \dfrac{e_1 + \mathrm{Se}}{R_3} \\ v = \mathrm{Sf} + \dfrac{e_2 - e_1}{R_1} \end{cases} \tag{2.24}$$

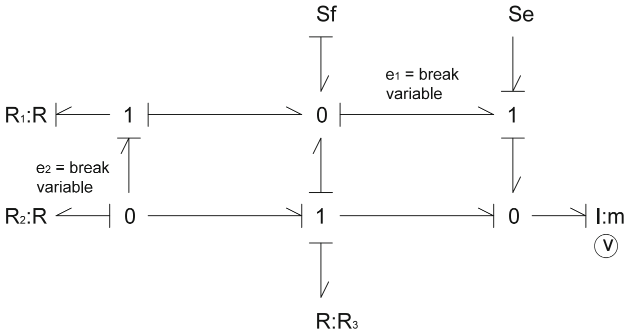


Fig. 2.16 System with class 3 and 4 ZCPs

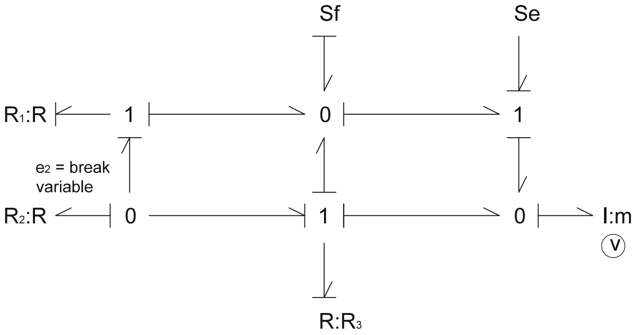


Fig. 2.17 Opening procedure for class 3 and 4 ZCPs

Figure 2.17 shows the application of the proposed method for this case. If causality in the *I* element is inverted, the *I* element imposes its associated effort. But, because of the presence of only one energy storage element, only one of the two original ZCPs can be broken, and one break variable is needed to solve the causality assignment. This break variable is e_2.

Expression (2.25) shows the obtained equations. *I* this case they are composed by a set of two equations. There is a set of implicit differential equations. Variables in the set of equations are v and e_2. In conclusion, originally one *I* element and two break variables (three equations), and with the application of the proposed method, the causality inversion in the *I* element eliminates one break variable, so only two equations are obtained.

$$\begin{cases} v = -\dfrac{e_2 + \mathrm{Se} + m\dot{v}}{R_1} - \mathrm{Sf} \\ \dfrac{e_2}{R_2} = -\left(e_2 + \mathrm{Se} + m\dot{v}\right)\left(\dfrac{1}{R_1} + \dfrac{1}{R_3}\right) - \dfrac{m\dot{v}}{R_3} \end{cases} \tag{2.25}$$

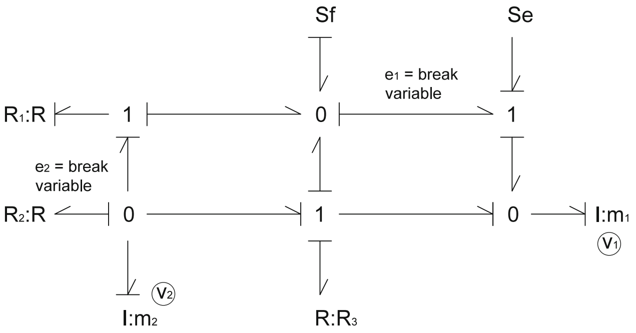


Fig. 2.18 System with class 3 and 4 ZCPs

Now, other similar example is presented, but with two *I* elements. Figure 2.18 shows the example. If causality assignment is applied, both *I* elements are assigned with preferred causality and two break variables, e_1 and e_2, are needed in order to solve the two ZCPs that also appear.

Expression (2.26) shows the obtained equations. They are composed by a set of four equations. Two are differential equations, related with the *I* elements, and the other two are algebraic equations related with the *R* element and ZCP class 4. Variables in the set of equations are v_1, v_2, e_1, and e_2.

$$\begin{cases} m_1\dot{v}_1 = e_1 + \text{Se} \\ m_2\dot{v}_2 = e_2 \\ \dfrac{e_2}{R_2} = -v_2 - (e_2 - e_1)\left(\dfrac{1}{R_1} + \dfrac{1}{R_3}\right) + \dfrac{e_1 + \text{Se}}{R_3} \\ \text{Sf} + (e_2 - e_1)\left(\dfrac{1}{R_1} + \dfrac{1}{R_3}\right) - \dfrac{e_1 + \text{Se}}{R_3} = v_1 - \dfrac{e_2 - e_1}{R_3} + \dfrac{e_1 + \text{Se}}{R_3} \end{cases} \tag{2.26}$$

And finally, Fig. 2.19 shows the application of the proposed method for this case. If causality in the *I* elements is inverted, the *I* element imposes its associated effort and both ZCPs disappear.

Expression (2.27) shows the obtained equations. They are composed by a set of two differential equations.

$$\begin{cases} v_1 = \text{Sf} + \dfrac{m_2\dot{v}_2 - m_1\dot{v}_1 + \text{Se}}{R_1} \\ v_2 = -\dfrac{m_2\dot{v}_2}{R_2} - (m_2\dot{v}_2 - m_1\dot{v}_1 + \text{Se})\left(\dfrac{1}{R_1} + \dfrac{1}{R_3}\right) + \dfrac{m_1\dot{v}_1 - \text{Se}}{R_3} \end{cases} \tag{2.27}$$

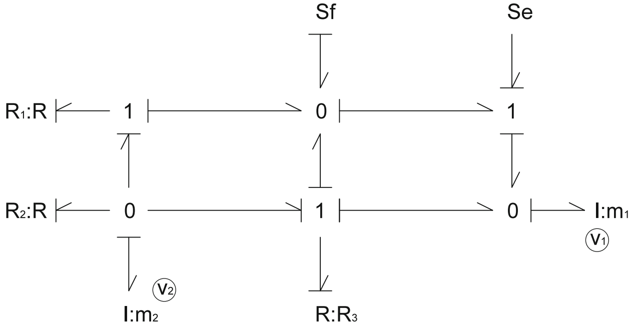


Fig. 2.19 Opening procedure

2.9 Conclusions

In this work, a method to eliminate the additional break variables needed when ZCPs appear in a bond graph is presented. This method has been developed with the aim of obtaining a minimal set of the dynamic equations of a system modeled with bond graphs. The method is based on an assignment of causality consisting in applying differential causality of I and C elements when ZCPs appear. With this causality assignment, ZCPs disappear and no break variables are needed.

The developed procedure allows the reduction of the number of equations in the dynamic system, obtaining the equations only in terms of the number of independent inertance and compliance elements of the system. A systematic analysis of each ZCP class has shown the way to apply the procedure. Some examples have been presented in order to show the application of the proposed method. The final equations are in most of the case purely ODEs, but in several particular cases, DAE equations remain.

Several examples are also provided in order to demonstrate the applicability of the proposed method when combination of different ZCOs appear simultaneously in the model.

The results obtained with this formulation are clearly superior, from the point of view of computational time, to those obtained with the classical bond graph formulation with break variables.

References

1. Borutzky, W. (2010). *Bond graph methodology: Development and analysis of multidisciplinary dynamic system models*. London: Springer. ISBN 978-1-84882-881-0.
2. Bos, A. M. (1986). *Modeling multibody systems in terms of multibond graphs*. Ph.D. thesis, Twente University, Enschede, The Netherlands.
3. Cacho, R., Félez, J., & Vera, C. (1997). Deriving simulation models from bond graphs with any combination of topological loop classes. *International Conference on Bond Graph Modeling and Simulation ICBGM'97* (Vol. 29, No. 1, pp. 85–93). Phoenix, AZ: SCS Publishing, Simulation Series.

4. Cacho, R., Félez, J., & Vera, C. (2000). Deriving simulation models from bond graphs with algebraic loops. The extension to multibond graph systems. *Journal of the Franklin Institute, 337*, 579–600.
5. Félez, J., Vera, C., San José, I., & Cacho, R. (1990). BONDYN: A Bond Graph based simulation program. *Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control, 112*, 717–727.
6. Felez, J., Romero, G., Maroto, J., & Martinez, M. L. (2011). Simulation of multi-body systems using multi-bond graphs. In *Bond graph modelling of engineering systems* (pp. 323–354). New York: Springer. doi:10.1007/978-1-4419-9368-7_9. Print ISBN: 978-1-4419-9367-0, Online ISBN: 978-1-4419-9368-7.
7. Gawthrop, P. J., & Smith, L. S. (1992). Causal augmentation of Bond Graphs with algebraic loops. *Journal of the Franklin Institute, 329*(2), 291–303.
8. Granda, J. (1984). Bond Graph modeling solutions of algebraic loops and differential causality in mechanical and electrical systems. *Proceedings IASTED Applied Simulation and Modeling Conference*, San Fancisco, pp. 188–193.
9. Kane, T. R., & Levinson, D. A. (1985). *Dynamics theory and applications*. New York, NY: McGraw-Hill.
10. Karnopp, D. C. (1983). Alternative bond graph causal patterns and equation formulation for dynamic systems. *Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control, 105*, 58–63.
11. Karnopp, D. C., & Margolis, D. L. (1979). Analysis and simulation of planar mechanism systems using Bond Graph. *Journal of Mechanical Design, 101*(2), 187–191.
12. Kim, S. S., & Vanderploeg, M. J. (1986). A general and efficient method for dynamic analysis of mechanical systems using velocity transformations. *ASME Journal of Mechanism, Transmissions, and Automation in Design, 108*, 176.
13. Nikravesh, P. E. (1990). Systematic reduction of multibody equations of motion to a minimal set. *Journal of Non-Linear Mechanics, 25*(2–3), 143–151.
14. Romero, G., Felez, J., & Vera, C. (2005). Optimised procedures for obtaining the symbolic equations of a dynamic system by using the bond graph technique. *International Conference on Bond Graph Modeling and Simulation ICBGM'05* (Vol. 37, No. 1, pp. 51–58). New Orleans, Phoenix, AZ: SCS Publishing, Simulation Series.
15. Rosenberg, R. C., & Andry, A. N. (1979). Solvability of bond graph junction structures with loops. *IEEE Transactions on Circuits and Systems, 26*, 130–137.
16. Sueur, C., & Dauphin-Tanguy, G. (1991). Bond-graph approach for structural analysis of MIMO linear systems. *Journal of Franklin Institute, 328*(1), 55–70.
17. Van Dijk, J., & Breedveld, P. (1991). Simulation of system models containing Zero-order Causal Paths-I. Classification of Zero-order Causal Paths. *Journal of the Franklin Institute, 328*(5/6), 959–979.
18. Van Dijk, J., & Breedveld, P. C. (1991). Simulation of system models containing Zero-order Causal Paths—II. Numerical implications of class 1 Zero-order Causal Paths. *Journal of the Franklin Institute, 328*(5/6), 959–979.
19. Wittenburg, J. (1977). *Dynamics of systems of rigid bodies*. Stuttgart: B.G. Teubner.
20. Zeid, A., & Overholt, J. (1995). Singularly perturbed Bond Graph models for simulation of multibody systems. *Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control, 117*(3), 401–410.

Chapter 3
Controlled Switched Structures for Bond-Graph Modelling and Simulation of Hybrid Systems

Matías A. Nacusse and Sergio J. Junco

3.1 Introduction

Hybrid dynamical systems are *mathematical abstractions* involving continuous- and/or discrete-time and discrete-event models [37]. More precisely, a hybrid system can be viewed as a finite set of subsystems called modes, each of them described by a differential equation (implying continuous-valued variables on a continuous-time domain) or a difference equation (involving continuous-valued variables on a discrete-time domain), and a discrete-event system acting as a switching supervisor. These modes evolve or are active once at a time and succeed each other in response to a switching sequence of discrete events happening at any time instant according to the rules defined by the supervisor. The discrete-event sequence is restricted to a finite number of them in any finite time interval [38].

More technically, and taking the case of switching among continuous modes described by state equations to fix ideas affine to the subject of this chapter, a hybrid system is usually presented as a collection of a finite number of vector fields $\{f_i,\ i = 1, \ldots, m\}$, and a rule dictating the switching among the dynamic modes $\dot{x}(t) = f_i\,(x(t),\ u(t))$, with the restriction mentioned above. Correspondingly, one can notice that just the simple happening of abrupt parameter changes in a system suffices to define a hybrid system. But in engineering, switching is broadly understood as involving structural or topological changes in a technical system, an occurrence that can cause more than just parametric changes: even if in many cases this technical switching can be captured by a collection of state equations like the above, in the more general situation DAE-systems, implicit differential equations

M.A. Nacusse (✉) • S.J. Junco
Laboratorio de Automatización y Control, Departamento de Control, Facultad de Ciencias Exactas, Ingeniería y Agrimensura, Universidad Nacional de Rosario, Ríobamba 245 Bis, S2000EKE Rosario, Argentina
e-mail: nacusse@fceia.unr.edu.ar

W. Borutzky (ed.), *Bond Graphs for Modelling, Control and Fault Diagnosis of Engineering Systems*, DOI 10.1007/978-3-319-47434-2_3

and even different sets of state variables are necessary to capture the admitted phenomena. See [10] for a comprehensive mathematical classification of hybrid and switched systems (put forward in the context of addressing tools for the analysis of their stability).

Yet another situation could arise due to the occurrence of abrupt changes in the interconnection structure of a system: discontinuities or instantaneous jumps in the state variables of its model could be induced by the switching [23]. Even if this is not allowed in physics, in engineering models this is frequently admitted, mainly due to the fact that the phenomena in which engineers are interested evolve over a time scale much bigger than the time scale of the change, and that the details inside the time window of the change are not relevant for the problem under study. Therefore, ignoring the dynamics driving the abrupt changes or commutations, and just taking into account their initial and final states, brings significant simplifications for the model builder and system analyst. Thus, modelling tools for switched systems should be capable to capture abrupt topological changes in the interconnection structure and be endowed with the ability to manage state jumps occurrences.

This chapter discusses in detail two simulation-oriented bond-graph modelling devices for hybrid systems satisfying these requirements. They are the Switchable-Structure Bond and the Generalized Switched Junction or, for short, SS-Bond [25] and GSJ [26]. The SS-Bond allows to represent all commutation modes between two power ports with a precise definition of their associated power variables. The GSJ, a controlled switched bond-graph structure, comes in two dual versions and can represent all the interconnections enforced by commutations involving bond-graph elements around otherwise standard 0- or 1-junctions. Besides illustrating their properties with simple examples, some technical applications are addressed. Related results previously presented in the bond-graph literature either as hybrid-, switched or variable-topology systems are reviewed and shown to be representable by these two new formalisms, so that the formers can be considered as special of particular cases of the latter.

In the bond-graph domain, the critical topics summarized above as implied by modelling hybrid systems manifest themselves mainly as changes in the power interconnection structure, the constituent elements and/or their constitutive relationships, and the causality assignment. In this sense, dealing with these questions, both switched structures, SS-Bonds as well as GSJs, have been conceived in order to yield a single bond graph, with fixed causality and a unique interconnection structure, simultaneously representing or capturing all the commutation modes. This single all-mode bond graph, that depending on the problem could contain both types of switched structures, is called *Switched* Bond Graph, or SwBG, for short. It should be noted that this name is kept despite the fact that switched systems are mostly understood as the subclass of hybrid systems where state jumps or discontinuities are *not* allowed [34]. It is so done because this is the most commonly used name in the bond-graph literature addressing commutations and, also, because it evokes the established usage of the word switching in engineering to refer to technical systems subjects to structural commutations.

Fixed causality is strived to as it is an important feature from a simulation perspective. Indeed, some modelling and simulation (M&S) software do not allow changes in the causality of the model during the simulation. Considering structural commutations happening instantaneously is the source of causal constraints that generate causality changes at some components. A very common approach to break these causal constraints is the adroit addition of parasitic elements in the model, see [19, 30, 31] therein. In the sequel, this approach to obtain models with fixed causality will be called *non-ideal* or *soft-switching* modelling, in opposition to the former approach, that will be designed as *ideal* or *hard-switching* modelling. Both formalisms discussed here, SS-Bonds and GSJs, can be set for modelling hard as well as soft-switching phenomena. The key tool allowing to achieve the all-mode single SwBG with fixed causality and unique power interconnection structure when modelling commutations as hard-switching is the integration of residual sinks into the internal structure of the formalisms. In the case of modelling with the soft-switching approach, the same effect is achieved incorporating parasitic components in place of the residual sinks.

It should also be noticed that, unlike many other tools, both devices can be employed in situations where the set of state variables is not the same in all of the switching modes as well.

With the sequence of modes being properly defined by a set of control variables in the switching supervisor, a SwBG featuring the above-mentioned properties is ready for a one-shot simulation, without need of interrupting and restarting it at any switching instant. This is possible even in the presence of state jumps, as a prescription is provided to calculate the necessary re-initialization of the discontinuous state variables, either offline, previous to the simulation, and to the integration of the results into the model, or at run-time. The key concept supporting this is the principle of charge and impulse conservation.

The next section briefly summarizes the most relevant devices introduced to deal with commutation phenomena in the bond-graph domain and some auxiliary tools used in the constitution of both SS-Bonds and GSJs. These two formalisms are addressed in detail, including application examples and some simulation results, in Sects. 3.3 and 3.4, respectively. In both cases the devices are introduced in their ideal- or hard-switching version. The non-ideal or soft-switching form is discussed in Sect. 3.5. Finally, the conclusions are presented in Sect. 3.6.

3.2 Summary Review of Some Bond-Graph Devices for Modelling Switching Systems and Auxiliary Tools

Originally, bond graphs have been introduced assuming the continuity and smoothness properties of classic physics. In order to cope with abrupt changes in the structure of physical systems, many bond-graph devices have been proposed that allow to face some of the challenges presented by hybrid systems including switching among the continuous modes.

Table 3.1 Properties of Switching components

		Fixed/Variable causality	Hard/Soft-switching	Well-defined boundary conditions	Single/ Multiple BG
Single port	Switch	V	H	Yes	M
	Ideal switch	V	H	Yes	M
	ISw and CSw	F	S	Yes	S
	MTFs + *r*S	F	H	Yes	S
	MTFs + *R*	F	S	Yes	S
Two port	MTFs	F	H	No	S
	Switchable bonds	V	H	No	M
	SS-Bonds	F	H/S	Yes	S
Multiport	Controlled junctions	V	H	Yes	S
	SPJ	F	H	Yes	S
	GSJ	F	H/S	Yes	S

These switching devices with their main features are classified and briefly discussed in §2.1 below according to the number of power ports that they connect (or disconnect): single-port, two-port and multiport switched structures. Relevant questions of importance concerning the possibilities of automatic simulation are: Does the use of a given formalism produce a single BG representing all the switching modes or does it produce multiple models? Is the causality fixed for all modes or is it variable? Are the boundary conditions on the bonds attached to the switching device well defined or not? Are there dangling junctions in the resulting model? Are the devices bivalent in the sense of admitting hard- and soft-switching modelling or just one of both? An overview of all of this is given in Table 3.1.

Another feature of practical relevance is the possibility of performing one-shot simulation runs of switched systems, i.e., running the simulations without interruption. Satisfying this requirement demands the provision of solutions based on physical principles and the concurrency of a simulation software tool able to implement the corresponding calculations, as discussed in §2.2 below.

3.2.1 *Bond-Graph Switching Devices*

In the *single-port category* fall the following elements, all of them with mode-dependent causality (see their BG representations in Table 3.2).

The *ideal switch* Sw, introduced by [33], encapsulates two ideal sources of zero value each. Indeed, it transitions between two modes assigning either zero flow (ZF) or zero effort (ZE) to its adjacent bond. It produces a single BG for all switching modes but the causality is not fixed, it must be re-assigned for each combination of modes of the ideal switches. The *switch* defined in [14], being an element

Table 3.2 Iconic representation of properties of single-port switched structures

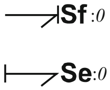
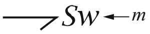
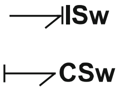
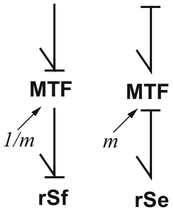
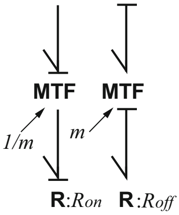
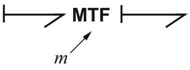
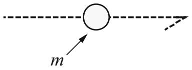
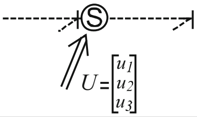
	Switch	*Ideal Switch*	*ISw and CSw*	*MTFs+rS*	*MTFs+R*
Iconic representation					

Table 3.3 Iconic representation of properties of two-port switched structures

	MTF	*Switchable bonds*	*SS-Bonds*
Iconic representation			

commutating between a zero-valued current source and a zero-valued voltage source (its introduction is restricted to electric circuits), is just an explicit version of the previous tool. Both are *hard-switching* devices.

The ISW and CSW, introduced in [16], belong to the category of *dynamic* soft-switching approximations of hard-switching, introduced with the aim of producing a single BG with fixed causality in all modes. The external port of the ISW is internally connected through a 1− junction to an *I*− element in integral causality and an ideal Sw, so that it is an effort-out element, behaving like an *I* when the switch is ON and like a ZF-source when the switch is OFF. Featuring a *C*− element instead of an *I*, its complementary device, the CSW, is explained dualizing all the previous terms.

Other one-port elements with fixed causality are the combination of an *MTF* (modulated by a gain taking the values 0 or 1) with a *residual sink* [3], and the combination of an *MTF* with a *R*-element [2, 9, 15], which is commonly used in the literature to represent a non-ideal switch. Both are static switching devices, the former modelling hard- and the latter soft-switching.

The simplest structure in *the two-port category* is the MTF modulated with a gain being either 0 (switch open) or 1 (switch closed) [2, 4, 20]. This use of the MTF results in a *fixed causality* representation of the switch element. In some cases when the switch is open, the boundary conditions may not be well defined, since this representation imposes zero flow or zero effort to the attached power ports and this condition may fail to disconnect the subsystems [33].

The switchable bond, introduced in [11] proposes conditionally present bonds. The boundary conditions on its adjacencies are not always explicitly defined in all the switching modes, which results in the dangling junction problem [32], see their BG representations in Table 3.3.

Table 3.4 Iconic representation of properties of multiport switched structures

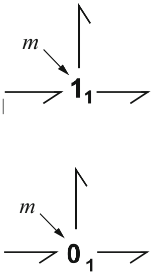
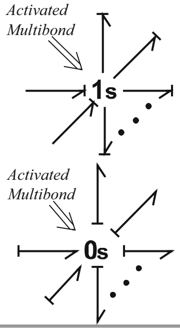
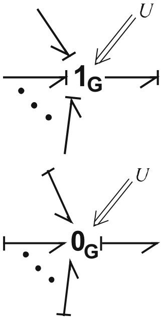
	Controlled Junctions	Switched Power Junctions	Generalized Switched Junction Structures
Iconic representation			

Table 3.5 SS-Bond modes and control variables

u_1	u_2	u_3	Mode
0	1	0	Σ_a and Σ_b connected
0	0	1	$\Sigma_a$and Σ_b in ZF
1	0	0	$\Sigma_a$and Σ_b in ZE
1	0	1	Σ_a in ZE and Σ_b in ZF
0	0	0	Σ_a in ZF and Σ_b in ZE

Finally, the switched structures comprising *the multiport category* have been conceived to extend the properties of the standard BG power conserving multiport structures, i.e., 0- and 1-junctions.

The controlled junction [22, 23], a *variable causality* device commanded by one control signal, has two operation modes, the *on-* and the *off-mode*. In the *on-mode*, it behaves like a standard junction, while in the *off-mode* it imposes zero value to the power variable common to all its adjacent bonds.

The Switched Power Junction [35], or SPJ for short, is a *fixed causality* formalism in which more than one bond can impose effort (0-SPJ) or flow (1-SPJ) as the common variable at the junction. The SPJs are commanded by multiple control signals, as among the many bonds that can impose the common junction variable, only one is allowed to do it at any given time instant; the value of the associated control signal is one (it selects the bond as imposing the effort or flow), the rest control signals are zero (they "disconnect" the corresponding bonds, see their BG representations in Table 3.4).

Most of the above-mentioned formalisms have been extensively discussed in the literature, see [6, 7], for instance. The prior brief summary reveals that none of these formalisms features all the desirable properties of providing a unique bond graph for all modes, fixed causality, ability to switch any number of power ports, well-defined state of all the components and suitability for modelling with the hard- and the soft-switching approaches.

The two formalisms being discussed in detail ahead in this chapter, the SS-Bond, first presented in [25], and the GSJ, introduced in [26], belong to the two- and multi-port categories, respectively. They have been conceived in order to overcome the shortcomings of previous devices of their classes: both produce a single BG with fixed causality in all the switching modes, the boundary conditions in their adjacency are well defined, there are no dangling bonds left, both are able to model with the hard- as well as the soft-switching approach, and can be conveniently supplemented to perform once-through simulation runs.

3.2.2 Auxiliary Tools

3.2.2.1 Residual Sinks

Instrumental to realize some of the previous features of both SS-Bonds and GSJ is the residual sink concept and device [5]. Indeed, residual sinks are used to break the causality constraints produced by switching—and so to achieve the fixed causality assignment—and to explicitly define the boundary conditions of each mode, what also precludes the presence of dangling bonds. There are two kinds of *residual sinks*: effort residual sinks *r*Se and flow residual sinks *r*Sf. The residual sink injects the necessary effort, or flow, in order to make vanish the power conjugated variable into the sink, see Fig. 3.1.

3.2.2.2 Conservation of Charge and Impulse and Integrator Resetting

The automatic change of models at the switching instant is determined by the variables controlling the SS-Bonds and the GSJs. When a state jump occurs, or when a new state variable enters the scene, the accomplishment of an additional task has to be assured: the values of these state variables at the inception of the new mode should be calculated and set in the corresponding integrator. In the case of state jumps, this calculation is performed using the conservation principles of generalized charge and momentum, see [23, 36] for a generalization of this principle to electrical and mechanical switched systems in the framework of port-Hamiltonian systems.

As for the integrator resetting, the possibility of executing a one-shot simulation, i.e., to perform the whole simulation run without interruption, demands a tool able to reset the integrators on the fly. This feature is offered, for instance, by 20Sim [12] through the sentence *resint*, and by OpenModelica [28] via the command *reinit*.

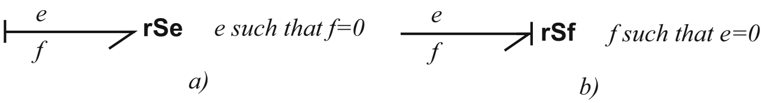


Fig. 3.1 (**a**) Effort residual sink. (**b**) Flow residual sink

3.3 Switchable-Structure Bond

The SS-Bond, depicted in Fig. 3.2, allows to model all commutation modes between two power ports, labelled "*a*" and "*b*" in the figure. When both ports are connected, the SS-Bond behaves like a standard power bond, thus performing full power transmission. When the power connection is absent, the SS-Bond annihilates the power transmission between the ports and fully captures four possible states of the adjacent power ports avoiding the problem of dangling junctions. As SS-Bonds are originally defined to model ideal switching, the aforementioned four possible states of the disconnected power ports are zero-flow (ZF) or zero-effort (ZE) each. These four situations, together with the normally connected state, define five possible switching modes for an SS-Bond, fact which calls for the set of the three binary control variables collected in the vector U in Fig. 3.2. In each specific problem these variables are to be determined by the switching supervisor commanding the SS-Bond.

3.3.1 Formal Definition of SS-Bond

The formal mathematical definition of this device should fully specify the relationships between the four power variables, $e_{a,b}$ and $f_{a,b}$, at the ports of subsystems Σ_a and Σ_b, see Fig. 3.2.

The causality strokes are defined *as being fixed for all the* SS-Bond *modes*. Specifying that the effort e_a is imposed by Σ_a and the flow f_b is calculated by Σ_b, it reads $e_b = e_a$ and $f_a = f_b$ in the normally connected state (from now on, the *ground connection mode*), in which the SS-Bond enforces full power transmission between the two ports. In addition to that, also the mathematical rules of the zero power transmission modes must be specified, where each power port can independently reach two modes, the zero flow and the zero effort mode. This behaviour is enforced by the action of the control signals u_i (with i from 1 to 3). Each of these signals can take only two values, either 0 or 1.

When Σ_b is in the ZE mode, the SS-Bond disregards the value of e_a and imposes zero effort to Σ_b. When Σ_b is in the ZF mode, the SS-Bond must enforce $f_b = 0$. As the causality at the power port of Σ_b indicates that it receives effort and delivers flow, the SS-Bond performs its task adjusting the value of e_b in order to

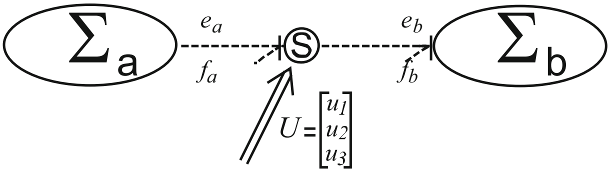


Fig. 3.2 SS-Bond interconnecting two power ports

achieve $f_b = 0$. This behaviour, which calls for an internal residual sink in the SS-Bond realization (to be presented in the sequel), allows to keep fixed the causality assignment.

When Σ_a is in the ZF mode the SS-Bond, disregarding the value of f_b, imposes zero flow to Σ_a. On the contrary, when Σ_a is in the ZE mode ($e_a = 0$) the SS-Bond enforces the value of f_a as needed to satisfy the algebraic restriction $e_a = 0$; here appears again a residual sink.

In Eqs. (3.1) and (3.2) the values of e_b and f_a for each mode are expressed. There, e_r and f_r are calculated by the residual sinks through the corresponding algebraic constraints $f_b = 0$ and $e_a = 0$.

$$e_b = \begin{cases} e_a \text{ when } \Sigma_a \text{ and } \Sigma_b \text{ are connected} \\ e_r \text{ when } \Sigma_b \text{ is in ZF mode} \\ 0 \text{ when } \Sigma_b \text{ is in ZE mode} \end{cases} \tag{3.1}$$

$$f_a = \begin{cases} f_b \text{ when } \Sigma_a \text{ and } \Sigma_b \text{ are connected} \\ f_r \text{ when } \Sigma_b \text{ is in ZE mode} \\ 0 \text{ when } \Sigma_b \text{ is in ZF mode} \end{cases} \tag{3.2}$$

3.3.2 SS-Bond Internal Realization

Figure 3.3 shows the internal realization of the SS-Bond, using components from the BG standard set. *MTFs* are used to model the mode switching, and residual sinks to solve the algebraic constraints of each mode which enforce the power preserving property of this structure.

Using the rules expressed in Eqs. (3.1) and (3.2) and the standard equation-reading procedure on bond graphs, the equations that govern the SS-Bond behaviour can be obtained as detailed in (3.3). Table 3.5 shows the values of the control signals necessary to determine each mode.

$$\begin{aligned} e_b &= (1 - u_1)\ u_2\, (1 - u_3)\, e_a + u_3\, (1 - u_2)\, e_r \\ f_a &= (1 - u_1)\ u_2\, (1 - u_3) f_b + u_1\, (1 - u_2) f_r \end{aligned} \tag{3.3}$$

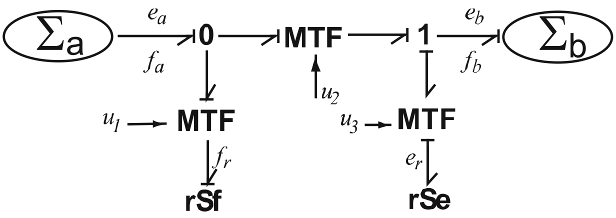


Fig. 3.3 Internal SS-Bond realization

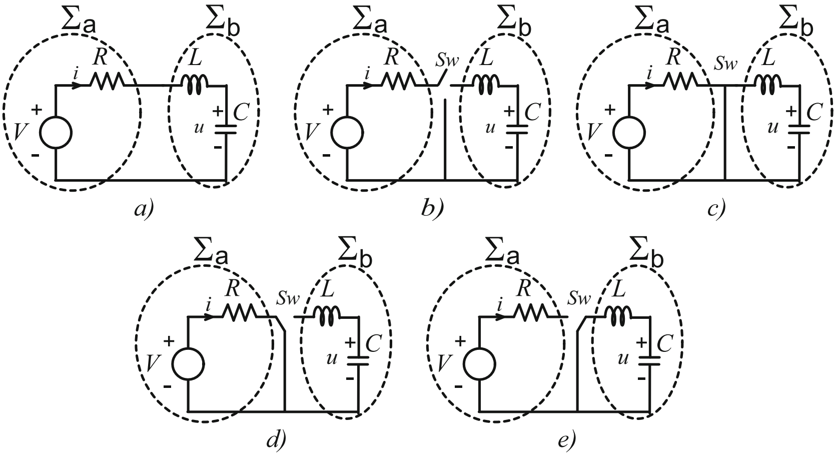


Fig. 3.4 Series RLC circuit (**a**) ground mode. (**b**) $\Sigma_a$and Σ_b in ZF mode. (**c**) $\Sigma_a$and Σ_b in ZE mode. (**d**) Σ_a in ZE mode and Σ_b in ZF mode. (**e**) Σ_a in ZF mode and Σ_b in ZE mode

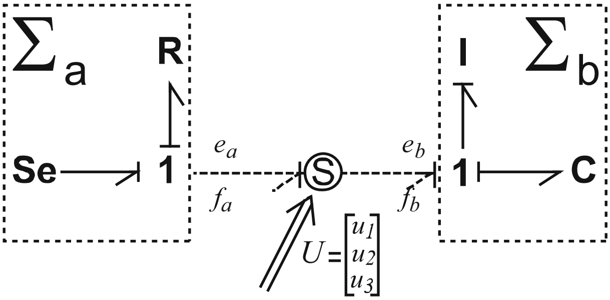


Fig. 3.5 SwBG model of a faulty RLC circuit

3.3.3 Modelling Example: Series RLC Circuit

Consider the series RLC circuit depicted in its ground configuration in Fig. 3.4a, and suppose that the two arbitrarily defined subsystems Σ_a and Σ_b can couple and decouple as indicated in the remaining figures. This possible behaviour covers all the five switching modes at their power ports. In a practical example, the different operational modes could be produced not only by a hypothetical intentional switching but also by abrupt faults in the electrical circuit. Figure 3.4b–e shows the different circuit switching modes. The SwBG of Fig. 3.5 fully models this switching behaviour employing an SS-Bond.

As already said, keeping fixed causality calls for the residual sinks to solve algebraic constraints in some modes. The following are the calculations of e_r and f_r for the different modes of this example.

When switching to the operation modes represented in Fig. 3.4c, d, the calculation of f_r is trivial: it is equal to $f_r = V/R$. But, when switching at a commutation instant t_C from any of the configurations of Fig. 3.4a, c, e into those of Fig. 3.4b, d, the inductor current is forced to jump instantaneously from its previous value (reached at ${t_C}^-$) to zero (to restart its evolution at ${t_C}^+$). *Integrator resetting*. To achieve this in these cases, the effort on Σ_b should be computed as $e_r = e_c + W\,\delta\,(t - t_C)$, where e_c is the effort on the capacitor and $W\,\delta\,(t - t_C)$ is the weighted Dirac impulse, with W the necessary effort value in order to bring the inductance current to zero because of the switching (circuit opening) at time $t = t_C$. Instead of doing this, as in a numerical simulation the Dirac impulse cannot be implemented, to force to zero the flow in the 1-junction, the integrator of the I element is simply reset to zero. In these particular examples, determining the new value of the state variable is trivial, it is evident from the new circuit configuration, so that it demands no special calculation. In another, arbitrary situation, appealing to integrator resetting to circumvent the impossible task of computing a delta-function will indeed demand a calculation of the new state. To this aim the principles of charge and momentum conservation will be called for, as shown in the stick–slip friction example further ahead.

3.3.4 Modelling Other Switched Formalisms Using SS-Bonds

In this subsection the SS-Bond is used to model two switched devices. First the switchable bond [11] is discussed and its shortcomings exposed through an electrical example. Then, using SS-Bonds, the same electrical example is modelled defining explicitly all the boundary conditions of each operation mode.

The second formalism is the ideal switch element [33], which is represented with fixed causality using SS-Bonds.

3.3.4.1 Modelling the Switchable Bond Using SS-Bond

The *switchable bond* presented in [11] is a controlled bond commanded by a control signal m that, taking the values 1 or 0, indicates the presence or absence of the bond in the BG model, i.e., indicates full or zero power transmission, respectively. The dashed power line suggests that this bond is only conditionally present (Fig. 3.6).

This approach has some problems, caused by the fact that the boundary conditions on the adjacency of the switchable bonds are not always explicitly defined

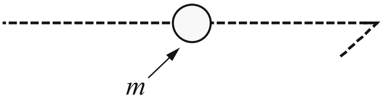


Fig. 3.6 Switchable bond representation

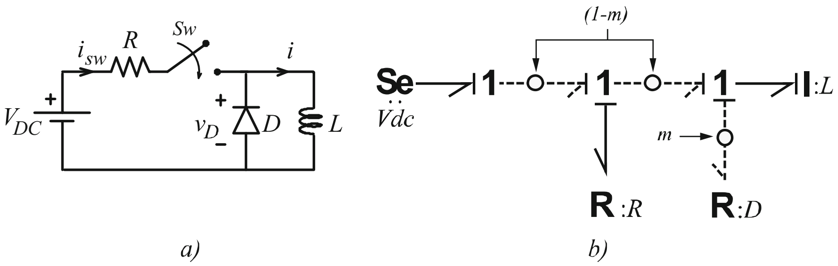


Fig. 3.7 (**a**) Switched electrical circuit. (**b**) SwBG model with switchable bonds

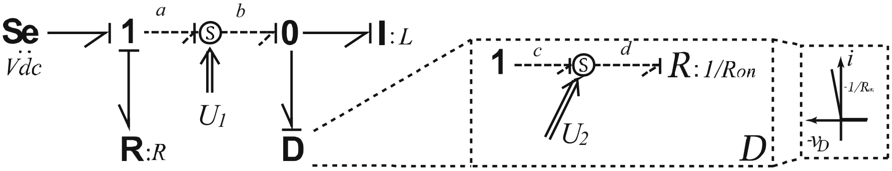


Fig. 3.8 SwBG model of the switched electric circuit with SS-Bonds, $U_i = [u_{i1}, u_{i2}, u_{i3}]$

in all the switching modes [32]. This fact, known as the problem of the *dangling junctions*, is illustrated with the help of the SwBG in Fig. 3.7b, where the ideal switch Sw in the electrical circuit of Fig. 3.7a is modelled using two switchable bonds. Also the diode's model, labelled *D*, uses a switchable bond, complemented with the resistor *D*, modelling its conduction state. The problem arises in the circuit configuration Sw = OFF, corresponding to $m = 1$, where the switchable bonds commanded by $(1 - m)$ are disconnected: the source **Se** and the resistor **R** receive each an undefined flow information, each one from an otherwise disconnected 1-junction. There is no problem with the switchable bond commanded by *m*, which connects the **I** and the **R**(*D*). The other circuit configuration, Sw = ON or $m = 0$, is properly defined.

Figure 3.8 shows the SwBG model of the switched electrical circuit of Fig. 3.7a using SS-Bonds for both, the ideal switch and the internal realization of the non-standard, newly defined diode model *D*. The diode always receives the voltage produced by the electrical series of the source **Se** and the linear resistor **R** and delivers current (flow-out fixed causality).

Both commutation modes of the circuit are explained next.

3.3.4.2 Mode 1: Electrical Switch ON and Diode OFF

In this mode, the SS-Bond commanded by U_1 is in its *ground connection mode*, while both ports of the SS-Bond commanded by U_2 are in the ZF state. Table 3.6 shows the values of the control signals associated with each *mode*.

Table 3.6 SS-Bond modes and control variables for Mode 1 of electrical circuit

Mode 1	SS-Bond mode	U_i	u_{i1}	u_{i2}	u_{i3}
Electrical switch ON	Σ_a and Σ_b connected	U_1	0	1	0
Diode OFF	Σ_c and Σ_d in ZF	U_2	0	0	1

Table 3.7 SS-Bond modes and control variables for Mode 2 of electrical circuit

Mode 2	SS-Bond mode	U_i	u_{i1}	u_{i2}	u_{i3}
Electrical switch OFF	$\Sigma_a$and Σ_b in ZF	U_1	0	0	1
Diode ON	Σ_c and Σ_d connected	U_2	0	1	0

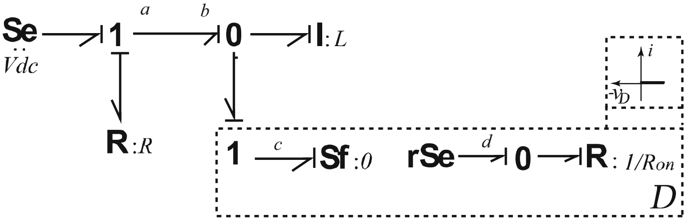


Fig. 3.9 BG model of the switched electrical circuit in mode 1

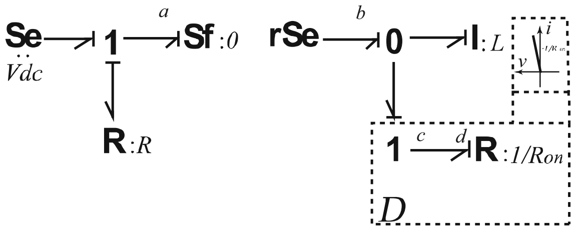
The power variables of the SS-Bond modelling the electrical switch are $e_b = e_a$ and $f_a = f_b$, while the power variables of the SS-Bond in the diode are $e_d = e_r$ and $f_c = 0$. As the conduction state of the diode is modelled as a (negative-slope) resistance, then $e_r = 0$ to enforce $f_d = 0$.

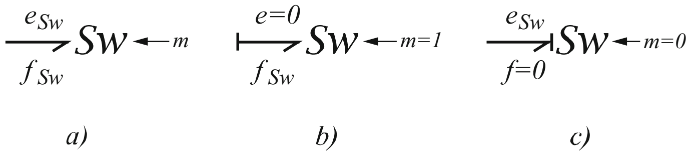
3.3.4.3 Mode 2: Electrical Switch OFF, Diode ON

In this mode, as the electric parallel of inductance *I* and diode *D* always receive the voltage from the power port *b* of the SS-Bond commanded by U_1, the power variables of the SS-Bond of the electrical switch are $e_b = e_r$ (the power port *b* is connected via an effort residual sink) and $f_a = 0$, while the power variables of the diode's SS-Bond are $e_d = e_c$ and $f_c = f_d$. Table 3.7 shows the values of the control signals associated with each mode.

To better understand how the SwBG model of Fig. 3.8 commutates between modes, the SS-Bond expanded model of Fig. 3.3 can be used: Figs. 3.9 and 3.10 show the equivalent BGs enforced on the single BG of Fig. 3.8 by the (hypothetical, not shown) external switching supervisor imposing the values 0 and 1 on *m* for each of the modes.

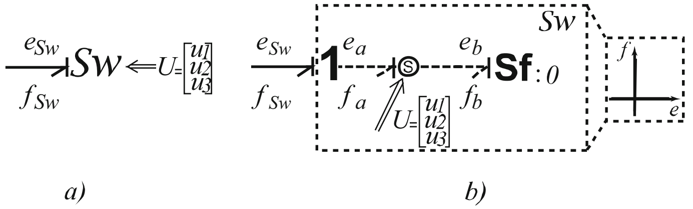
Fig. 3.10 BG model of the switched electrical circuit in mode 2





a) b) c)

Fig. 3.11 (**a**) Ideal switch element. (**b**) ZE mode. (**c**) ZF mode



a) b)

Fig. 3.12 (**a**) Controlled flow-out Sw element. (**b**) Internal representation with SS-Bond

3.3.4.4 Modelling the Ideal Switch Element Using SS-Bond

The ideal switch element presented in [33] and noted Sw is a controlled switched BG component with variable causality. The two operation modes of this element are commanded by its control variable *m:* setting $m = 1$ (ON-mode) enforces ZE on the system, while setting $m = 0$ (OFF-mode) imposes ZF on it. Figure 3.11 shows the iconic representation and the available modes of the Sw element.

Figures 3.12 and 3.13 show the flow-out and effort-out fixed causality versions of the Sw, as well as their internal realization using SS-Bonds. Very conveniently for simulation purposes, the user can choose the one more adapted to the problem at hands (for instance, to assure the integral-causality representation of a storage affected by the commutation in the BG model). Only two of the five available modes of each SS-Bond are used in this case, the *ground connection mode* and the ZE or ZF mode, the two latter in accordance with the effort-out or flow-out version, respectively; See Tables 3.8 and 3.9 for the corresponding values of the control signals associated with each mode. In the dashed squares of each figure the constitutive relationship of each element is represented; when the effort–flow relationship is indeterminate (in Fig. 3.12, Sw in ZE mode and in Fig. 3.13, Sw in ZF mode) the SS-Bonds connect to residual sinks that calculate the flow or effort, respectively.

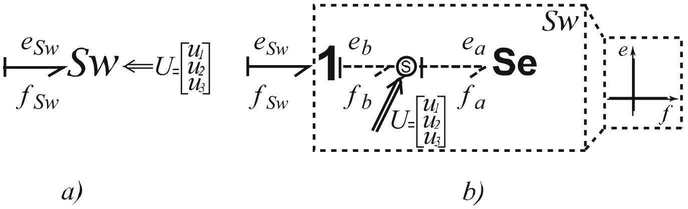

e_{Sw}
f_{Sw}
Sw
$U=\begin{bmatrix}u_1\\u_2\\u_3\end{bmatrix}$
a)
e_{Sw}
f_{Sw}
Sw
1
e_b
f_b
S
e_a
f_a
Se
$U=\begin{bmatrix}u_1\\u_2\\u_3\end{bmatrix}$
e
f
b)

Fig. 3.13 (**a**) Controlled effort-out Sw element. (**b**) Internal representation with SS-Bond

Table 3.8 Sw and SS-Bond modes control variables for flow-out causality

Sw mode	SS-Bond mode	u_1	u_2	u_3
OFF	Σ_a and Σ_b connected	0	1	0
ON	Σ_a and Σ_b in ZE	1	0	0

Table 3.9 Sw and SS-Bond modes control variables for effort-out causality

Sw mode	SS-Bond mode	u_1	u_2	u_3
ON	Σ_a and Σ_b connected	0	1	0
OFF	Σ_a and Σ_b in ZF	0	0	1

3.3.5 Application Example: Modelling Stick–Slip Friction with SS-Bonds

Stick–slip behaviour is a well-known discontinuous phenomenon due to dry friction resisting the relative lateral motion of two bodies at their contact surface. According to Coulomb's model, the friction force F_f is independent of the relative speed of the surfaces. At the static friction, or stiction, or stuck regime (no relative motion), this force equals any applied external force up to a threshold or maximum value above which motion would start. At the dynamic, or kinetic, or sliding regime, it takes a fixed value independent of the relative speed of the surfaces [1]. Both characteristic values are given by the formula $F_f = \mu N$, where μ is the friction coefficient and N is the force normal to the surfaces, with μ taking different values, μ_S in the stuck mode and μ_k in the sliding mode, the latter being usually less than the first. This discontinuous behaviour, which forces zero velocity in the stick mode and imposes a force in the slip mode, gives place to a switched system in any mechanical model where it is considered. Many models are available in the literature, among them those proposed in [18, 21] in the BG domain, with fixed causality in both cases. The first one introduces a trick which consists in the addition of a nonlinear relationship between the velocity (v) and the momentum (p) making it zero when the system is in stick mode, while the second one introduces parasitic elements to break the constraints at the origin of the causality changes.

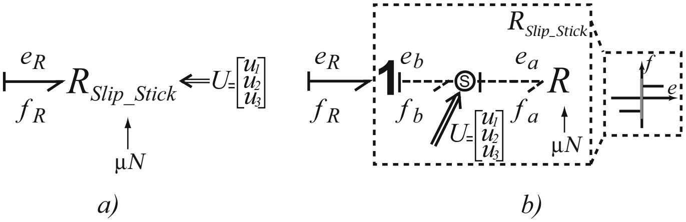


Fig. 3.14 $R_{\text{Slip\_Stick}}$ element. (**a**) Iconic representation. (**b**) Internal representation with SS-Bonds

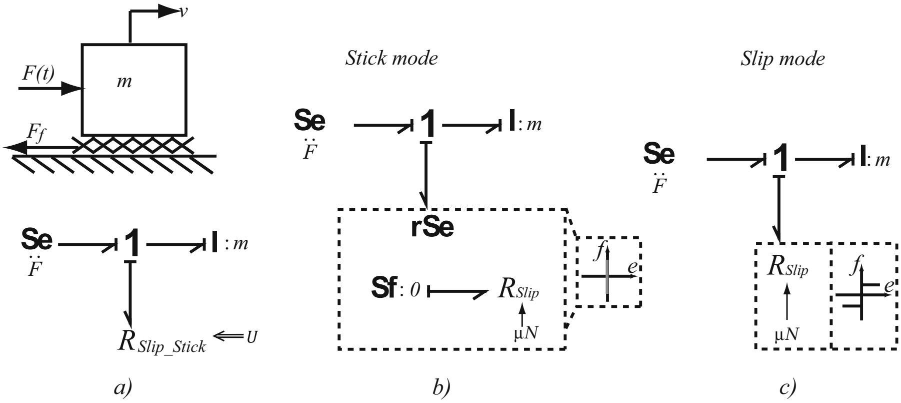


Fig. 3.15 (**a**) Mechanical system and SwBG with internal representation of $R_{\text{Slip\_Stick}}$, (**b**) Stick mode, (**c**) Slip mode

A new controlled switched BG-element is introduced next, labelled as $R_{\text{Slip\_Stick}}$. It features effort-out fixed causality, as shown in Fig. 3.14a and a SS-Bond in its internal realization, as depicted in Fig. 3.14b, and models the following behaviour, which is better explained with the help of the simple mechanical example of Fig. 3.15.

Stick mode: the velocity v (more generally, the relative velocity between the sliding surfaces, f_R in Fig. 3.14) is equal to zero and remains so until the applied force F becomes greater than the threshold force F_{th} and the commutation condition C_{Slip} into the Slip mode is reached $\left(C_{\text{Slip}} : |F| > F_{\text{th}}\right)$. While in this mode, the friction force F_f developed by the new device (e_R in Fig. 3.14) is $F_f = F$. Even with both power ports of the SS-Bond in ZF mode, the effort e_r must be calculated through the algebraic restriction $v = 0$ because the constitutive relation is undefined at this speed. The net effect of the SS-Bond in this mode is shown in Fig. 3.15b.

Slip mode: here the velocity v is different from zero and the value of the friction force is $F_f = \mu_k N \text{sign}(v)$. The net effect of the SS-Bond in this mode is shown in Fig. 3.15c. The system remains in this mode until the commutation condition into the stick mode is reached, which is $C_{\text{Stick}} : |v| \leq v_{\min} \wedge |F| \leq F_{\text{th}}$, where the symbol $\wedge$ stands for the logical operation "and". The system leaves this mode when

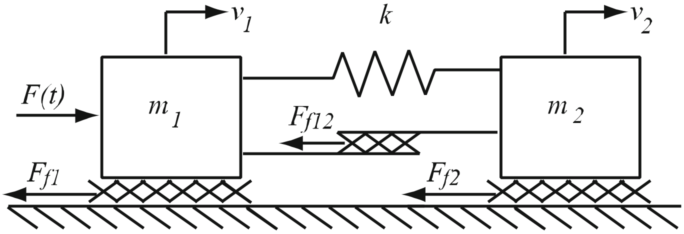


Fig. 3.16 Sketch of mechanical system

the speed $|v|$ diminishes until a minimum value $|v| = v_{\min}$ and instantaneously enters the stick mode, where the speed is zero. Instead of injecting the necessary effort impulse to enforce this finite speed jump, the storage element I is reinitialized to zero, similarly as it was done in the electric circuit example of §3.3. Equation (3.4) summarizes the computations of the friction force in both modes.

$$F_f = \begin{cases} \mu_k N \text{sign}(v) & \text{slip mode} \\ e_r \ \text{ Such that } f = 0 & \text{stick mode} \end{cases} \tag{3.4}$$

3.3.5.1 Application Example

The following example, developed in [21], is revisited here solving the M&S task with two different approaches, in both cases using SS-Bonds. The first approach, developed in full detail (including simulations in the next section), makes the assumption of strictly ideal commutations or hard-switching. The second one, using the soft-switching approach, in line with [21] employs parasitic *R*-*C* components to avoid causality changes; it is discussed further ahead, in § 3.5.1.

The mechanical sketch of Fig. 3.16 shows the system configuration featuring two masses, m_1 and m_2, a spring k coupling both masses, and dry friction occurring at the contact surface of both masses with the ground and at the relative motion between them. The friction forces are noted F_{f1}, F_{f2} and F_{f12}. An external force $F(t)$ acts on the system.

Figure 3.17 shows the associated SwBG model with three $R_{\text{Slip\_Stick}}$ elements. As each $R_{\text{Slip\_Stick}}$ has two operation modes, the mechanical system can have up to $2^3 = 8$ available modes, but only five of them are possible, because the three combinations of two $R_{\text{Slip\_Stick}}$ elements in stick mode are equivalent to all the three of them in stick mode. In Table 3.10 the different switched modes of the SwBG are presented along with the associated modes of each of the $R_{\text{Slip\_Stick}}$ components, as well as the order of the state vector (n_{SV}) of the particular BG of each mode (not of the mechanical system). Despite the fact that Fig. 3.17 shows all the three storages in integral causality, the SwBG order changes with the switching and can take all the integer values from 0 (static friction everywhere, full static behaviour of the system) to 3 (dynamic friction everywhere), over 1 (static friction only in the

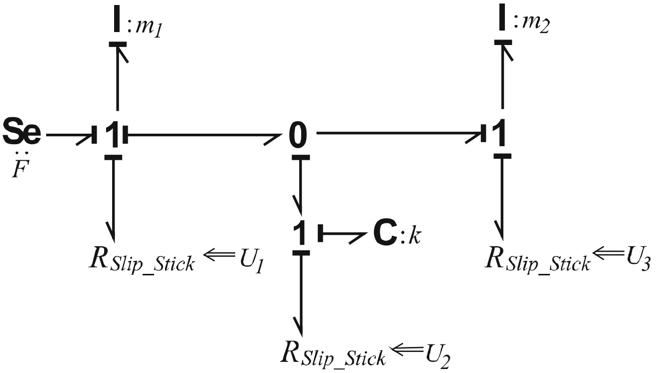

I:m_1
I:m_2
Se
$\ddot{F}$
1
0
1
1
C:k
$R_{Slip\_Stick} \Leftarrow U_1$
$R_{Slip\_Stick} \Leftarrow U_3$
$R_{Slip\_Stick} \Leftarrow U_2$

Fig. 3.17 SwBG model of mechanical system

Table 3.10 SwBG modes, modes of the $R_{\text{slip\_stick}}$ elements, and n_{SV}

	$R_{\text{slip\_stick1}}$		$R_{\text{slip\_stick12}}$		$R_{\text{slip\_stick2}}$		
Modes	Mode	F_{f1}	Mode	F_{f12}	Mode	F_{f2}	n_{SV}
Mode 1	Stick	$F - kq_k - F_{f12}$	Stick	0	Stick	$kq_k + F_{f12}$	0
Mode 2	Slip	$\mu_1 N_1 \text{sign}(v_1)$	Slip	$\mu_{12} N_{12} \text{sign}(v_{12})$	Stick	$kq_k + F_{f12}$	2
Mode 3	Stick	$F - kq_k - F_{f12}$	Slip	$\mu_{12} N_{12} \text{sign}(v_{12})$	Slip	$\mu_2 N_2 \text{sign}(v_2)$	2
Mode 4	Slip	$\mu_1 N_1 \text{sign}(v_1)$	Slip	$\mu_{12} N_{12} \text{sign}(v_{12})$	Slip	$\mu_2 N_2 \text{sign}(v_2)$	3
Mode 5	Slip	$\mu_1 N_1 \text{sign}(v_1)$	Stick	$\frac{F - F_{f1} - \left(1 + \frac{m_1}{m_2}\right) kq_k + \frac{m_1}{m_2} F_{f2}}{1 + \frac{m_1}{m_2}}$	Slip	$\mu_2 N_2 \text{sign}(v_2)$	1

middle) and 2 (static friction only between one of the masses and ground). Due to the switching the overall system is a hybrid system, with discrete states and transitions between them, which can be captured by a finite state machine (FSM) as shown in Fig. 3.18. This FSM is, in this example, the switching supervisor mentioned in the Introduction, responsible for generating the commutation signals driving the hybrid system. To each discrete state corresponds the continuous dynamics of each of the switching modes. Notice that in this case there is a feedback or bidirectional interaction between the discrete-event supervisor and the continuous dynamics of each mode, as the switching decisions of the former depend on the state variables of the latter. In Fig. 3.18, the symbol $\vee$ stands for the logic operator "or", and C_{slip1}, for instance, stands for "the transition condition from stick to slip of $R_{\text{Slip\_Stick}}$ commanded by U_1".

As stated in [21], when the three $R_{\text{Slip\_Stick}}$ elements are in stick mode, one of the three friction forces is mathematically undetermined, but using the residual sink concept, if $R_{\text{Slip\_Stick1}}$ is in stick mode ($v_1 = 0$) and $R_{\text{Slip\_Stick2}}$ is in stick mode ($v_2 = 0$), then the necessary effort F_{f12} to maintain the algebraic restriction $v_1 - v_2 = 0$ is equal to zero.

Storage Resetting and Impulse Conservation When the system enters Mode 5 the *I*-storages must be reinitialized in such a way that the total momentum immediately

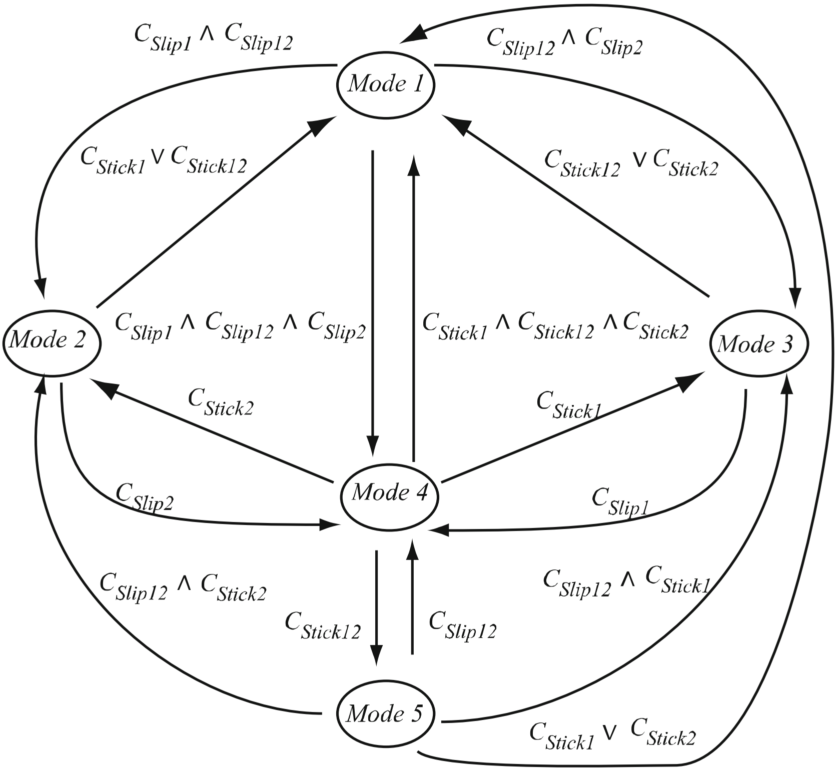

Mode 1
Mode 2
Mode 3
Mode 4
Mode 5
$C_{Slip1} \wedge C_{Slip12}$
$C_{Slip12} \wedge C_{Slip2}$
$C_{Stick1} \vee C_{Stick12}$
$C_{Stick12} \vee C_{Stick2}$
$C_{Slip1} \wedge C_{Slip12} \wedge C_{Slip2}$
$C_{Stick1} \wedge C_{Stick12} \wedge C_{Stick2}$
C_{Stick2}
C_{Stick1}
C_{Slip2}
C_{Slip1}
$C_{Slip12} \wedge C_{Stick2}$
$C_{Slip12} \wedge C_{Stick1}$
$C_{Stick12}$
C_{Slip12}
$C_{Stick1} \vee C_{Stick2}$

Fig. 3.18 Finite state machine of the mechanical system, the switching supervisor of the hybrid system

before the mode switching is conserved after it. Let the mode switching happen at time instant t_0 and t_0^- and t_0^+ be their left and right limits, then, the previous condition translates into:

$$p_1\left(t_0^-\right) + p_2\left(t_0^-\right) = p_1\left(t_0^+\right) + p_2\left(t_0^+\right) \tag{3.5}$$

When the system enters Mode 5 the algebraic restriction $v_1(t) - v_2(t) = 0$ must be accomplished, which is equivalent to $p_2(t) = \frac{m_2}{m_1} p_1(t)$ ($\forall\ t > t_0$ in Mode 5), so the storage elements must be reinitialized after the switching as follows:

$$p_1\left(t_0^+\right) = \frac{m_1}{m_1 + m_2}\left[p_1\left(t_0^-\right) + p_2\left(t_0^-\right)\right] \tag{3.6}$$

$$p_2\left(t_0^+\right) = \frac{m_2}{m_1 + m_2}\left[p_1\left(t_0^-\right) + p_2\left(t_0^-\right)\right] \tag{3.7}$$

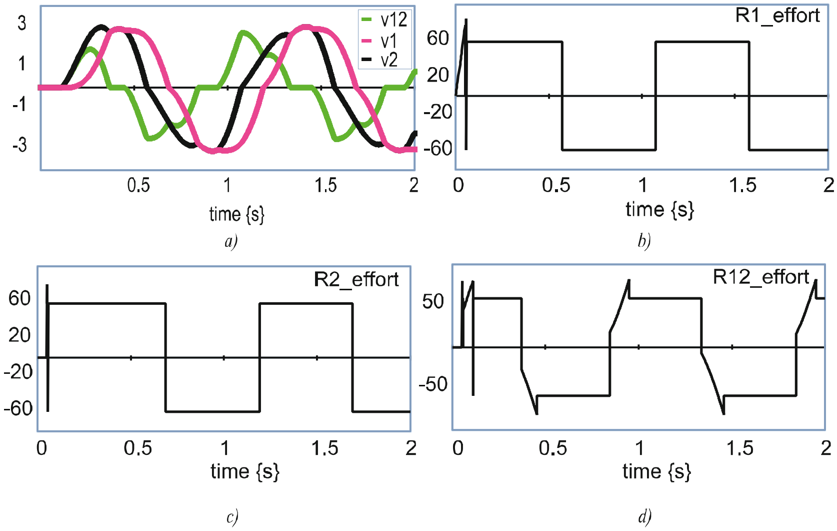


Fig. 3.19 (**a**) Velocity responses, in [***m/s***]. (**b**) Friction force on mass $\boldsymbol{m}_1$ in [*N*]. (**c**) Friction force on $\boldsymbol{m}_2$ in [*N*]. (**d**) Friction force on masses m_1 and m_2 in [*N*]

3.3.5.2 Simulation Results

The parameters used in the simulations are [21]: $m_1 = 5$ kg, $m_2 = 5$ kg, $k = (m_1 + m_2)(2\pi)^2$, $\mu_k = 0.6$, $F_{th} = 1.4$ $\mu_k m_i g$, with $i = 1, 2$ $v_{\min} = 0.001$ m/s, $F(t) = 200 \sin\ (\Omega t)$ and input frequency equal to 1 Hz, where F_{th} is the threshold force.

With the system initially at rest, the force $F(t)$ is applied at $T = 0$ s to the mass m_1. The simulation responses are shown in Fig. 3.19, where the time evolutions of the velocities and those of the friction forces of the $R_{\text{Slip\_Stick}}$ elements are depicted.

Remark In the example, the added algebraic constraints were solved offline, which means that the simulator solves an explicit set of differential equations. In case where the added algebraic constraints cannot be solved offline, they must be numerically solved at each integration step, through an implicit calculation, so the simulator must solve a DAE system, with the consequent increment of the computational cost.

3.4 Generalized Switched Junction Structures

In this section two multiport controlled switched structures are introduced. These structures, first presented in [26], called generalized switched junction structures 1-GSJ and 0-GSJ, or GSJ for short, can represent all the interconnections modes enforced by commutations involving BG-elements around the standard 0- and 1-junctions. They will be graphically represented as 0_G and 1_G.

The main attributes of the GSJ are presented in three different subsections for the hard-switching case. First, their behaviours are detailed, exposing their constitutive relationships and showing their principal features using a simple electrical example. Second, the internal realizations of the GSJs using BG components from the standard set are presented. Finally, it is shown how other multiport switched structures, as CJ and SPJ, can be represented as particular configurations of the GSJs.

3.4.1 Formal Definition of the GSJ

To better understand the behaviour of the GSJs consider that they have a ground configuration where they behave like standard BG-junctions. This reference configuration is just one of their possible switching modes. In any of the other switching modes, the junction behaves as in the ground configuration but only for a subset of all the adjacent bonds, while the remaining bonds get disconnected from the junction but conserving the structure. Thus, in a 1-GSJ (0-GSJ) these bonds do not contribute any effort (flow) to the junction, while their flows (efforts) are determined by the structural condition which their own efforts (flows) must satisfy. The configuration of a set of control variables decides which is the subset of bonds sticking to the ground junction configuration (selected bonds) and which is the subset disconnected (not selected bonds).

Some terminology taken from [27] is used here to clarify the behaviour of the GSJ. In the standard set of BG components in a 1-junction (0-junction) the bond called "weak causality bond" imposes effort (flow) to the junction, on the contrary the "strong causality bond" imposes the shared flow (effort) to the junction.

The iconic representation of the GSJ with causality assignment is shown in Fig. 3.20, where $U = [u_1, u_2, \ldots, u_n]$ is the vector of control signals. The bonds numbered from 1 to $(n-1)$ are "weak causality bonds" while the nth is the "strong causality bond" of the junction. Each control signal u_i $(i = 1, 2 \ldots n)$ can only take the value 1 or 0 and commands the ith bond (notice that, on the contrary as it happens in the SPJ, in the GSJ the control variables can take their values, 1 or 0,

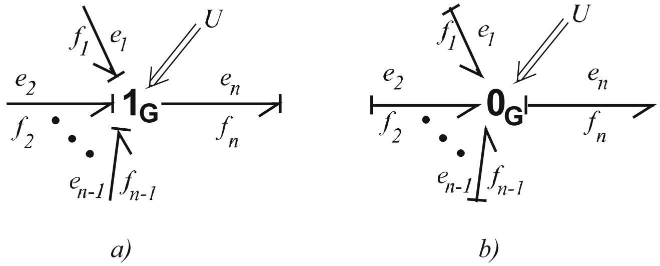


Fig. 3.20 (**a**) 1-GSJ representation. (**b**) 0-GSJ representation

independently from each other). This situation defines two operation modes for each of the bonds adjacent to the GSJ, which are called zero effort (ZE) and zero flow (ZF) mode, as seen from the GSJ standpoint. This terminology will become clear with the following explanation.

For *weak causality bonds*, when $u_i = 0$ (with $i = 1,\ 2\ldots,\ n-1$), the *i*th bond is in ZE mode for a 1-GSJ or in ZF for a 0-GSJ. This does not necessarily means that the value of the associated variable is zero, but that this bond does not contribute its associated power variable to the junction and its co-variable of power is calculated through an algebraic restriction. The use of these algebraic restrictions, which in the BG domain are implemented with residual sinks, furnishes the GSJ with the ability of explicitly defining the boundary conditions, i.e., the efforts and flows values, for all the adjacent bonds in all switching modes. As seen before, this feature is shared by the SPJ and the CJ, but not by the switchable bonds, which is the cause of the dangling junction problem of this formalism [32].

When $u_n = 0$, then the *strong causality bond* (*n*th bond) imposes zero to its associated power variable (which is transmitted by the junction only to the selected bonds) and its co-variable of power is obtained from an algebraic restriction. Equations (3.8) and (3.9) specify precisely the relationships among all the variables in the 1-GSJ and the 0-GSJ, respectively, where assigning the sign plus or minus depends on whether the power flow comes in or out the junction.

$$\begin{aligned} e_n &= u_n \sum_{i=1}^{n-1} u_i\,(\pm e_i) \pm (1-u_n)\,e_r \\ f_i &= u_i u_n f_n + (1-u_i) f_r \ \forall\ i = 1 \text{ to n} \end{aligned} \tag{3.8}$$

$$\begin{aligned} f_n &= u_n \sum_{i=1}^{n-1} u_i (\pm f_i)_i \pm (1-u_n) f_r \\ e_i &= u_i u_n e_n + (1-u_i)\,e_r \ \forall\ i = 1 \text{ to n} \end{aligned} \tag{3.9}$$

In (3.8) the variables e_r and f_r, which are associated to restrictions, are calculated according to the following rules:

e_r is calculated through the algebraic restriction $f_n = 0$ when the *strong causality bond* of the 1-GSJ is in ZF mode. This uses the embedded residual sink of the GSJ in order to enforce the value of e_r as to satisfy the restriction $f_n = 0$.

f_r is calculated through the algebraic restriction $\sum_{j=1}^{j=m} \left(1 - u_{\mu j}\right)\ e_{\mu j} = 0$, where $m \le n-1$ is the number of *weak causality bonds* in ZE mode, i.e., $u_{\mu j} = 0$, and $\mu_j \in \{1,\ 2,\ldots,n-1\}$ (i.e., μ_j is the index of the not selected bonds).

Analogue algebraic restrictions are used to obtain f_r and e_r for the *strong causality bond* in ZE and the *weak causality bonds* in ZF mode of (3.9).

3.4.2 GSJ Internal Realization

Similarly as done with the SS-Bond in the previous section, the GSJs can be represented by standard BG components, cf. Figs. 3.21 and 3.22. The control signal enters in the BG multiplying the power variables through *MTFs*; the algebraic operations between power variables are carried out by the standard junctions of the BG formalism and the algebraic constraints are added using residual sinks.

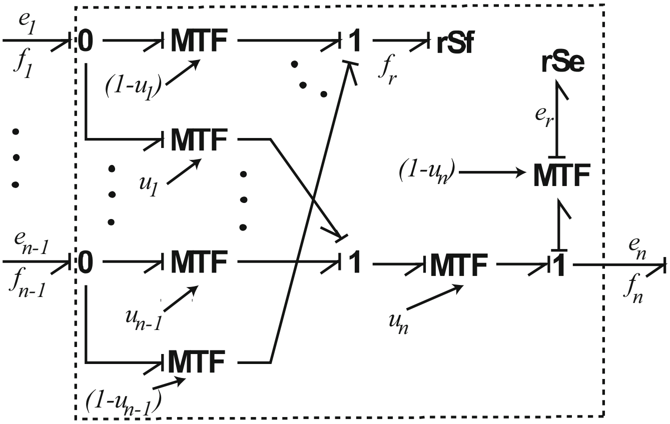


Fig. 3.21 1-GSJ represented with elementary BG elements

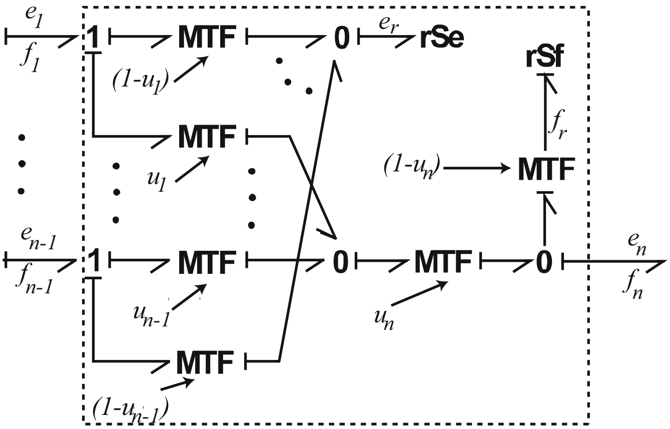


Fig. 3.22 0-GSJ represented with elementary BG elements

Fig. 3.23 1-GSJ model of switching series circuit

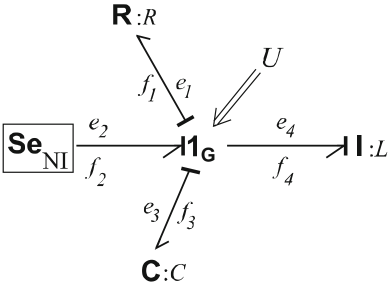


Table 3.11 Switching modes of a 1-GSJ

Figures	u_4	u_3	u_2	u_1	Modes
24a	0	0	0	0	$\mathrm{Se_{NI}}$ RC in ZE and I in ZF
24b	0	0	0	1	$\mathrm{Se_{NI}}$ C in ZE and RI in ZF
24c	0	0	1	0	RC in ZE and $\mathrm{Se_{NI}}I$ in ZF
24d	0	0	1	1	C in ZE and $\mathrm{Se_{NI}}$RI in ZF
24e	0	1	0	0	$\mathrm{Se_{NI}}R$ in ZE and CI in ZF
24f	0	1	0	1	$\mathrm{Se_{NI}}$ in ZE and RIC in ZF
24g	0	1	1	0	R in ZE and $\mathrm{Se_{NI}}$CI in ZF
24h	0	1	1	1	$\mathrm{Se_{NI}}$RIC in ZF
25a	1	0	0	0	$\mathrm{Se_{NI}}$RIC in ZE
25b	1	0	0	1	$\mathrm{Se_{NI}}C$ in ZE and RI in ZE
25c	1	0	1	0	RC in ZE and $\mathrm{Se_{NI}}I$ in ZE
25d	1	0	1	1	C in ZE and $\mathrm{Se_{NI}}$RI in ZE
25e	1	1	0	0	$\mathrm{Se_{NI}}R$ in ZE and CI in ZE
25f	1	1	0	1	$\mathrm{Se_{NI}}$in ZE and RIC in ZE
25g	1	1	1	0	R in ZE and $\mathrm{Se_{NI}}$CI in ZE
25h	1	1	1	1	Standard 1 − junction

3.4.3 Modelling Example: Series RLC Circuit

As an example, consider the series circuit of Fig. 3.25h and assume the possible occurrence of the ample spectrum of configurations depicted in Figs. 3.24 and 3.25. All of them can be captured by the BG of Fig. 3.23, with the control vector U defined in Table 3.11 (the non-standard BG element named $\mathbf{Se_{NI}}$ stands for a non-ideal effort source, represented by the rectangular block identified with the voltage V in Fig. 3.24h). There are 16 configurations in Figs. 3.24 and 3.25, the reference configuration of Fig. 3.25h plus 15 switching modes, so that a control vector with four variables is needed: $U = [u_1,\ u_2, u_3, u_4]$.

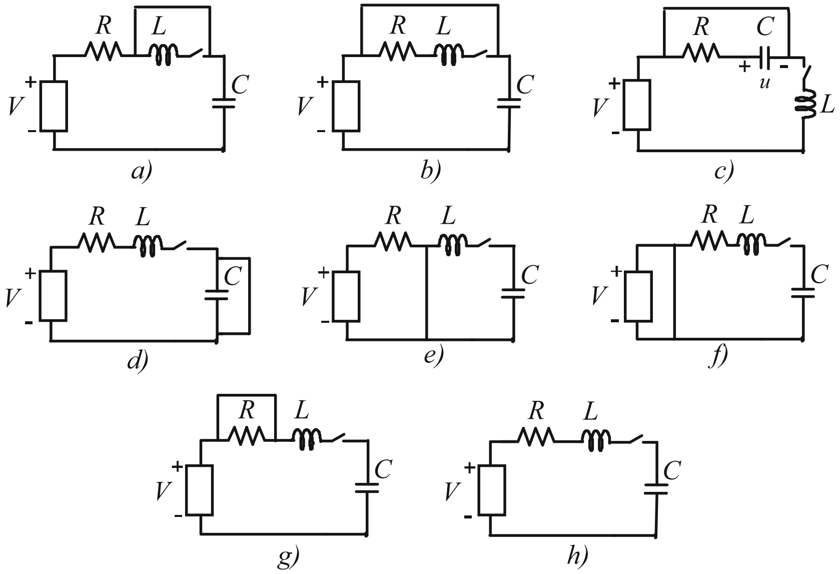


Fig. 3.24 ZF modes of the series electrical circuit

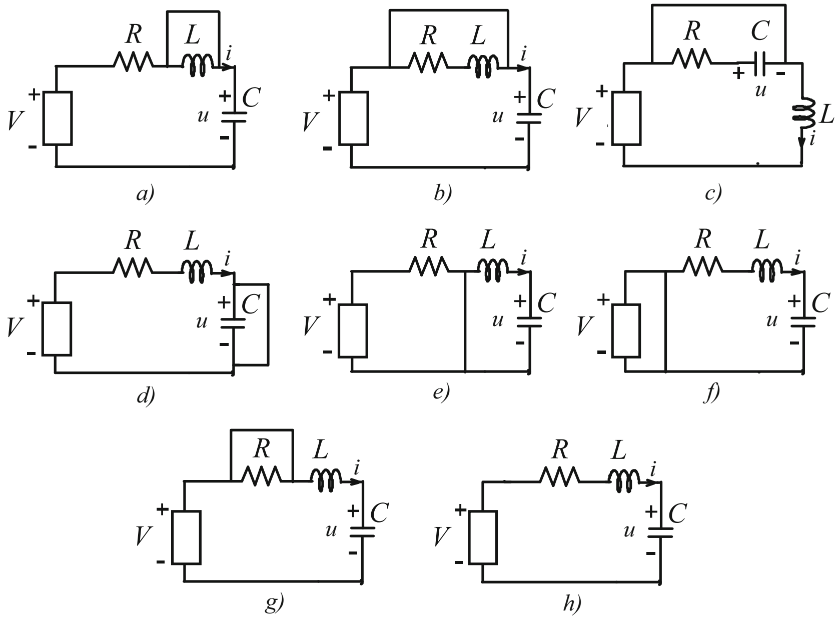


Fig. 3.25 ZE modes of the series electrical circuit

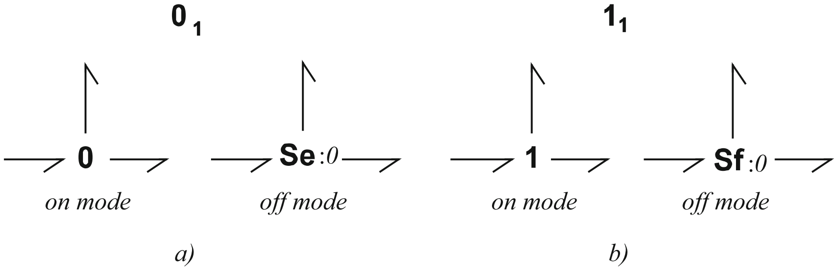


Fig. 3.26 Controlled junctions. (**a**) 0_1-junction. (**b**) 1_1-junction

Table 3.12 Pairing between GSJ and CJ

GSJ	u_n	u_i	GSJ modes	CJ modes	CJ
1_G	1	1	Standard 1-junction	On-mode	1_1
	0	1	All bonds in ZF mode	Off-mode	
0_G	1	1	Standard 0-junction	On-mode	0_1
	0	1	All bonds in ZF mode	Off-mode	

3.4.4 Modelling Other Switched Formalisms Using GSJ

In this subsection the CJ and the SPJ are represented as particular configurations of GSJ to expose the flexibility of the latter to represent different switched structures.

3.4.4.1 Representing CJ with Fixed Causality Using GSJ

The Controlled Junctions (CJ) presented in [23] are power conserving junction structures with variable causality conceived to model structural changes in switched systems in the BG domain. There are two kinds of them, noted as 0_1 and 1_1, each one has two operation modes, "on mode" and "off mode". In the *on mode* the CJ behaves like a standard junction, while in the *off mode* it imposes zero effort or zero flow as shared power variable. Figure 3.26 shows the iconic representation of the CJ and the two operation modes. Notice that, **Se** and **Sf** are to be interpreted as multibond ideal sources with zero value at all their output ports.

Representing the CJ using GSJ is a trivial task, since the CJ in the on mode behave like standard junctions and in the *off mode* they can be represented by a GSJ imposing ZE (for 0_1-junctions) or ZF (for 1_1-junctions) mode to all their bonds. In Table 3.12 the different values of the control signals to represent CJ using **GSJ** are shown.

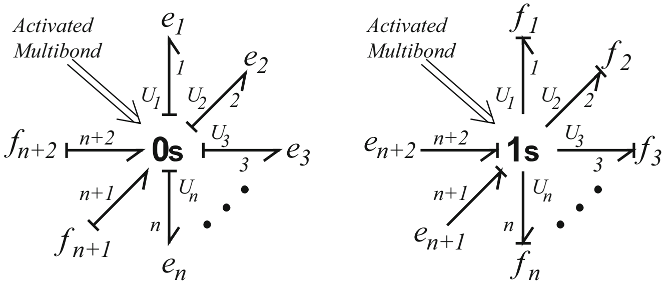


Fig. 3.27 Switched Power Junctions with causality assignment

3.4.4.2 Representing Switched Power Junction using GSJ

The Switched Power Junctions (SPJs) are extensions of the standard 0- and 1-junctions [35]. They are represented as receiving the effort (0_s) or flow (1_s) information from *more than one bond*. To prevent the causal conflicts this would otherwise imply, control signals, taking values over the set {1, 0}, are added to the new elements. Only one of these signals is allowed to have the value 1 at a given time instant, the remaining being zero. In this way, only one of the bonds imposing effort (0_s) or flow (1_s) is selected (i.e., becomes operative) and the value zero is imposed to the power co-variables of the remaining bonds, which results in their disconnection. Figure 3.27 shows the SPJs with causality assignment and Eqs. (3.10) and (3.11) express the mathematical relationships—for the 0_s and the 1_s, respectively—among the power variables and the control signals U_i injected to select the appropriate bond.

$$\begin{aligned} \text{Effort} &= U_1 e_1 + U_2 e_2 + \cdots + U_n e_n \\ f_i &= U_i \left(f_{n+1} + f_{n+2}\right);\ i = 1, \ldots, n \end{aligned} \tag{3.10}$$

$$\begin{aligned} \text{Flow} &= U_1 f_1 + U_2 f_2 + \cdots + U_n f_n \\ e_i &= U_i \left(e_{n+1} + e_{n+2}\right);\ i = 1, \ldots, n \end{aligned} \tag{3.11}$$

Figure 3.28 shows the 0_S realization using GSJ where each 0_G is commanded by the vector control signal $U_i = [u_{i1}\ u_{i2}]$ (with i from 1 to n) and the 1_S is commanded by the vector control signal $U' = \left[u'_1\ u'_2 \ldots u'_{n+1}\right]$. This realization requires the definition of $\left(2^2 n\right)\left(2^{n+1}\right) - 2n$ control signals to represent the behaviour of the 0_S-junction. Figure 3.28 can be simplified, to obtain the expanded version of the SPJ using BG components of the standard set as depicted in Fig. 3.29, as presented in [17]. To see a step-by-step derivation of the latter BG starting from that of Fig. 3.28 refer to [26].

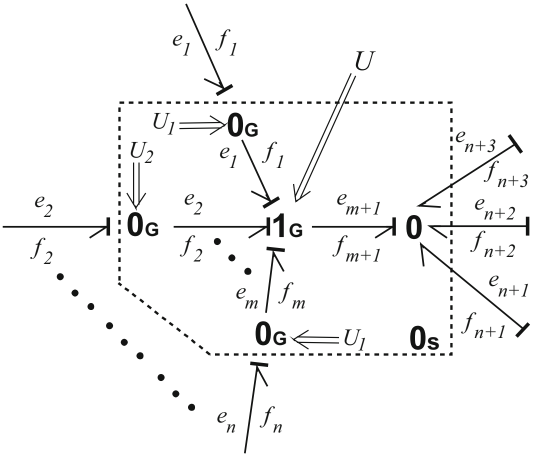


Fig. 3.28 0-SPJ internal realization using GSJ

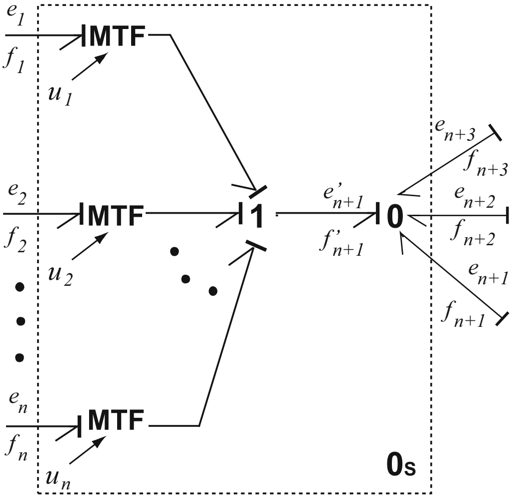


Fig. 3.29 0-SPJ using BG basic components

3.4.5 Application Example: Z-Source DC Breaker

The ability of the GSJs to represent multiple switches configuration in a practical situation is illustrated next via the modelling of an electronic circuit. The *Z*-Source DC circuit breaker [13]—a switched circuit capable of isolating a short circuit fault on the terminals of a load of a DC source—is modelled and simulated in this section using GSJs.

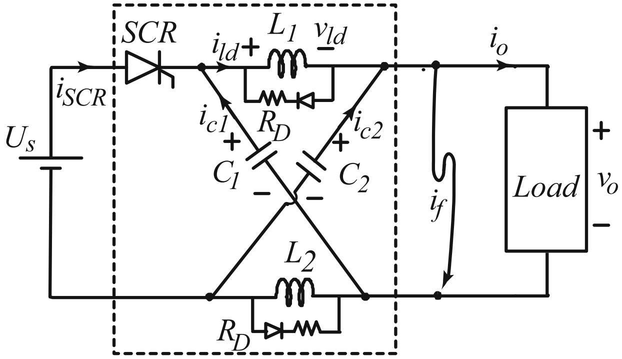


Fig. 3.30 *Z*-Source DC circuit breaker

3.4.5.1 *Z*-Source DC Circuit Breaker

The *Z*-Source DC circuit breaker was introduced as a mean to protect the power source from short circuit faults in DC buses. This circuit, inspired in the *Z*-Source inverter presented by [29], consists in an SCR in series with a LC network placed between the voltage source and the load as it is shown in Fig. 3.30. The main feature of the inverter application inspiring this circuit breaker is that in the former the *Z*-Source network allows the simultaneous closing of the two transistors on the same leg, which would be impossible in a classic inverter circuit. This quality was used in [13] to handle fault in medium voltage DC power systems. In the BG domain, a SwBG model of a monophasic *Z*-Source inverter was presented in [24] where SPJ and residual sinks were used to model the power switches and enforced integral causality in all the storage elements, respectively.

The behaviour of the *Z*-Source DC breaker will be analysed with the help of Fig. 3.31 where the four commutation modes are depicted. In steady state and normal operation, i.e., in the absence of faults or *fault free mode*, the SCR is in ON-state and, in steady state, each of the capacitors is charged at the source voltage (see Fig. 3.31a). A short circuit fault occurring at the load terminals (see Fig. 3.30) would try to equalize the sum of the two capacitor voltages, satisfying $U_{c1} + U_{c2} = 2\ U_s$ immediately before the fault, to U_s, the voltage at the source terminals. As this is obviously a situation inconsistent with Kirchhoff's second law, that imposes $U_{c1} + U_{c2} = U_s$, this constraint would attempt to produce a current impulse from the capacitors to the voltage source, to obey the principle of charge conservation. This phenomenon does not happen with the SCR in the circuit; instead, this attempt of negative current provokes the SCR to switch into the OFF-state isolating the voltage source from the fault. The transition from the ON-state to the OFF-state of the SCR can be analysed at the light of both, the hard- and soft-switching approaches:

In a hard-switching approach, this transition will occur instantaneously, so that the circuit switches from the *fault free mode* to the *fault mode 1* without entering any *transient mode*. In a soft-switching approach, i.e., when neither the SCR nor

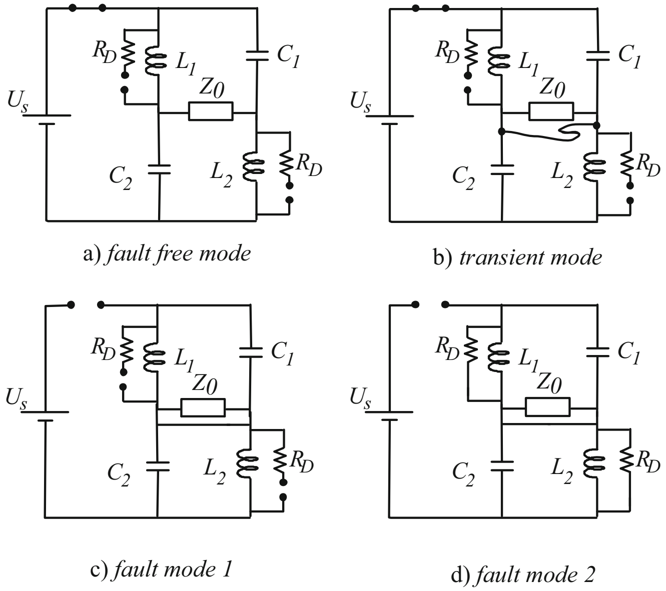


Fig. 3.31 *Z*-Source DC circuit breaker modes. (**a**) *Fault free mode*. (**b**) *Transient mode*. (**c**) *Fault mode 1*. (**d**) *Fault mode 2* Z_0 represents the load impedance

the short circuits are considered to be ideal, but to offer a very small resistance, the system enters into a mode that will be called *transient mode*. This mode, depicted in Fig. 3.31b, considers a short transition time between the *fault free mode* and the *fault mode 1*, where the current flowing through the SCR goes gradually from its steady state value to zero. In this subsection only the hard-switching models are presented, the presentation of the soft-switching construct being deferred to the next section.

Once the SCR is in the OFF-state, the remaining circuit will oscillate until the voltage at the inductors become negative, in that moment, the free wheel diodes will switch to the ON-state and the stored energy will be dissipated at the resistors, this is *fault mode 2*. For a detailed report of the *Z*-Source DC breaker, refer to [13].

The construction of the SwBG model of the *Z*-Source DC breaker is an easy task using GSJ since all the commutations involve efforts, which are modelled using 0-GSJ as shown in Fig. 3.32.

Figure 3.33 shows the *fault free mode* of the *Z*-Source DC breaker with its associated control signal in Table 3.13, where the SCR is in the ON-state and the diodes are in the OFF-state. The two effort residual sinks model the open circuit modes of the diodes, injecting the necessary effort to make the flow zero, which in this case, are each equal to zero.

When a short circuit fault occurs at the load the capacitors try to balance their charge, by injecting a current impulse into the voltage source, but the SCR limit the current passing to off-state when the current attempts to be negative. This situation, in an ideal switching approach, occurs at the same time instant. Notice that the

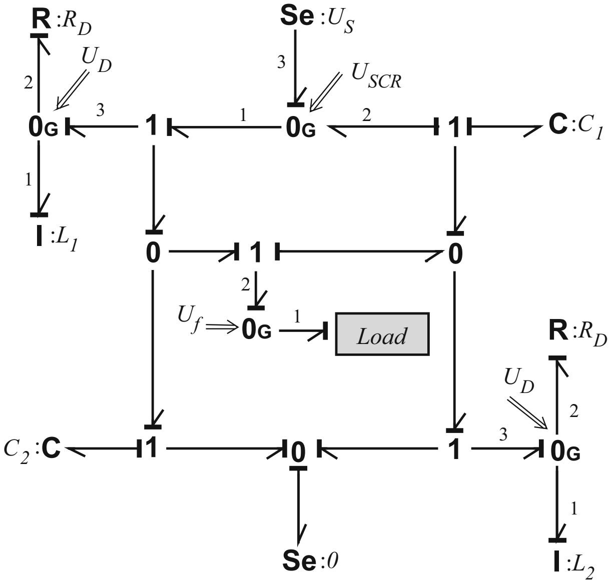


Fig. 3.32 SwBG model of the *Z*-Source DC circuit breaker

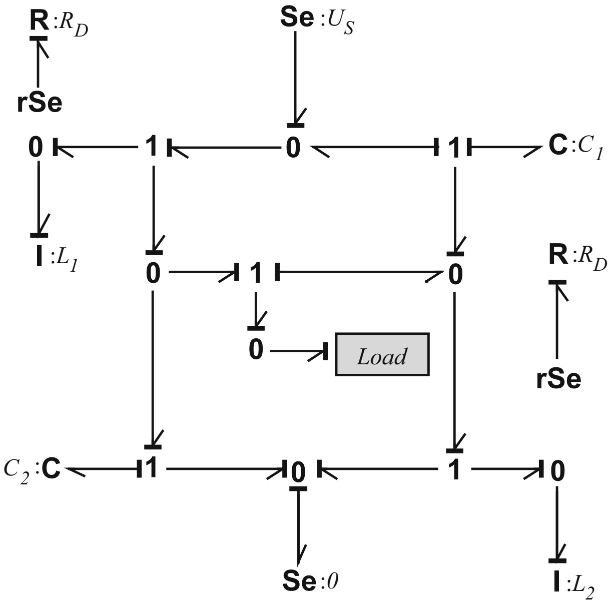


Fig. 3.33 *Fault free mode* of the *Z*-Source DC circuit breaker

Table 3.13 *Z*-Source DC circuit breaker in fault free mode

	u_1	u_2	u_3	Modes
U_{SCR}	1	1	1	SCR in ON-state
U_f	1	1	–	Fault free
U_D	1	0	1	Diodes in OFF-state

Table 3.14 *Z*-Source DC circuit breaker in fault mode 1

	u_1	u_2	u_3	Modes
U_{SCR}	0	0	1	SCR in OFF-state
U_f	1	0	–	Short circuit fault
U_D	1	0	1	Diode in OFF-state

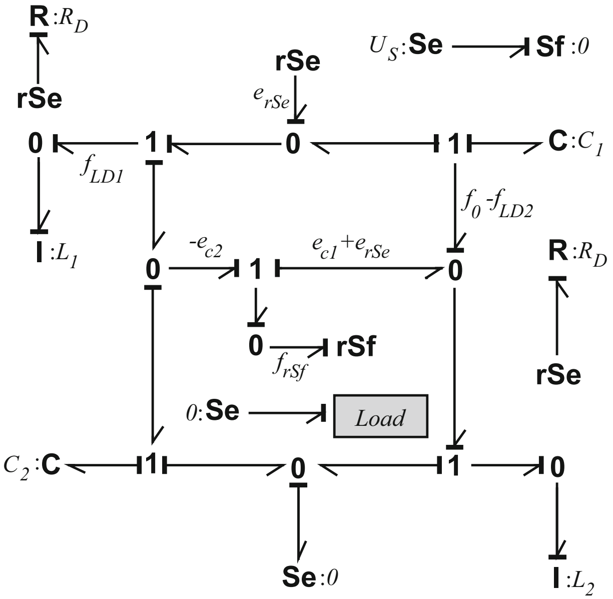


Fig. 3.34 *Fault mode 1* of the *Z*-Source DC circuit breaker

fault injection is simply made by changing the control signal of U_f as shown in Table 3.14. Figure 3.34 shows the resulting BG for *fault mode 1*, where the residual sinks associated with the fault and the SCR OFF-state are calculated reading directly from the BG. In particular, the **rSe** and **rSf** must inject the necessary effort or flow to make $-f_{\text{LD1}} - f_{\text{LD2}} - f_{r\text{Sf}} = 0$ and $-e_{C2} - e_{C2} - e_{r\text{Se}} = 0$, respectively. This two restrictions must be accomplished at the same time which implies $f_{r\text{Sf}} = f_{\text{LD1}} + f_{\text{LD2}}$ and $e_{r\text{Se}} = -e_{C2} - e_{C2}$.

The system remains in this mode until the freewheeling diodes, which are connected in parallel with the inductors, switch to the conduction state and allow the resistors R_D to dissipate the stored power on the LC network yielding the BG of Fig. 3.35 with control inputs shown in Table 3.15.

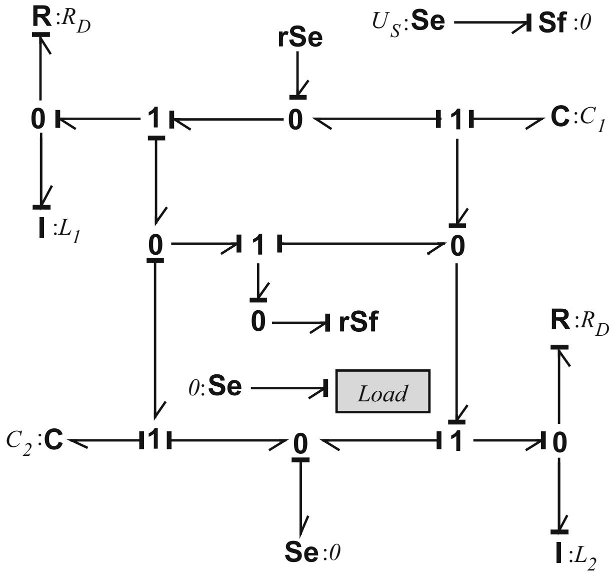


Fig. 3.35 *Fault mode 2* of the *Z*-Source DC circuit breaker

Table 3.15 *Z*-Source DC circuit breaker in fault mode 2

	u_1	u_2	u_3	Modes
U_{SCR}	0	0	1	SCR in OFF-state
U_f	1	0	–	Short circuit fault
U_D	1	1	1	Diode in ON-state

Table 3.16 Parameters of *Z*-Source DC circuit breaker

Parameter	Value	Parameter	Value
$C_1 = C_2$	125 μF	R_D	0.1Ω
$L_1 = L_2$	200 μH	R_L	6Ω
U_s	6000 V	C_L	1 mF

3.4.5.2 Simulation Results

In this subsection some simulation results of the *Z*-Source DC breaker are presented to show the correct behaviour of the GSJs in simulation and their versatility to fit both the ideal- and non-ideal switching approaches. The second one takes into account a more realistic model of a short circuit fault and the SCR. The simulation parameters shown in Table 3.16 were obtained from [13] where a parallel RC-load was used to test the circuit. The simulation scenario is the same for both modelling approaches. The system is in steady state operation when a short circuit fault occurs at time $t_f = 0.2$ s.

The SwBG model of Fig. 3.32 was implemented in 20Sim [12]. Figure 3.36a shows the time evolution of the currents, while Fig. 3.36b shows the voltages,

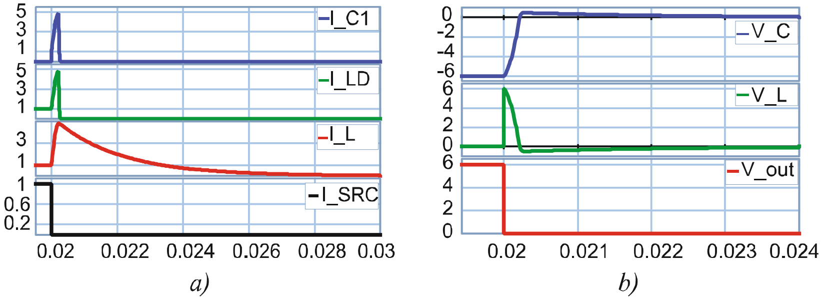


Fig. 3.36 Time evolution. (**a**) Current waveforms, in amperes [*KA*], (**b**) Voltage waveforms, in volts [*KV*]

where the null time span of the fault can be appreciated in the V_{out} voltage. The time interval of Fig. 3.36b has been chosen shorter to clearly show the transients in the two first voltage plots; it should be noticed that both figures correspond to the same simulation run.

3.5 Non-Instantaneous or Soft-Switching Modelling

The residual sink can be interpreted as the limit case of a storage element with its internal parameter tending to zero [8]. Thus, a non-ideal implementation of a residual sink would be reached assigning a very low value to this parameter. As it would cause a response with high frequency contents, the addition of a R component of convenient value is suggested to quickly damp the fast dynamics. So, replacing with parasitic elements the residual sinks employed in the implementation of the switching BG devices would make them suitable for the *non-ideal or soft-switching* modelling approach.

The representations of Fig. 3.3 for the SS-Bond, Fig. 3.21 for the 1-GSJ and Fig. 3.22 for the 0-GSJ are directly useful if willing to resign the hard-switching method in favour of the soft-switching approach: it suffices to replace the residual sinks with the parasitic components. Another valid alternative is to replace the residual sinks with MTFs plus dissipators, as done in [6, 15]. For instance, in the 0-GSJ of Fig. 3.22 the residual sinks would be replaced each with one of the options shown in Fig. 3.37.

The main benefit of this approach is that no algebraic restrictions are added to the system equations. However, and detrimentally, besides the well-known increment in the model order and the numerical stiffness it produces, the added parasitic elements are difficult to parameterize and, usually, their values depend on the simulation tests to be performed. Guidelines to perform this parameterization can be borrowed from the literature on mechanical systems, where this practice is widely extended as a method to eliminate derivative causality in multibody system models due to

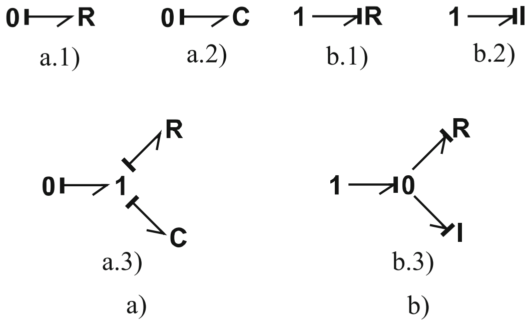


Fig. 3.37 (**a**) *r*Se replacing options. (**b**) *r*Sf replacing options

constraints introduced by mechanism joints. A parameter selection method based on the energetic activity of the parasitic components can be found in [30] as well as an account of other techniques previously contributed within the bond-graph community.

3.5.1 Soft-Switching Approach in the Stick–Slip Friction Example

Similar results to those in [21] can be obtained by replacing the effort residual sinks of the SS-Bonds internal representation by an *R-C* pair of BG components coupled by a 1-junction. This is shown in the BG of Fig. 3.38, where R_{Slip} imposes the effort in slip mode ($F_{fi} = \mu_i N_i \text{sign}\,(v_i)$), while in stick mode the friction force is calculated trough the resultant effort from the *R-C* pair, which yields $F_{fi} = k_s q_{si} + b_s \dot{q}_{si}$, where k_s is the parameter of a very rigid spring, q_{si} is the displacement associated with the added *C* and b_s is the damping associated with the *R* element.

3.5.2 Soft-Switching Approach in the Z-Source DC Breaker

In this example, the short circuit fault is modelled with an admittance evolving as a ramp from zero to $1/R_f = 50\ \ 1/\Omega$, during a time interval of $\Delta t_{\text{ramp}} = 0.1$ ms, and the SCR OFF-resistance with $R_{\text{SCR-off}} = 10\ \text{K}\Omega$. Figure 3.39 shows the new SwBG model with the added fault resistance R_f. It should be noticed that the 0_G modelling the SCR is now internally equipped with the $R_{\text{SCR-off}}$ resistance. Now the vector control signal changes from the fault free case $U_f = [1, 0, 1]$ to the faulty case $U_f = [1, 1, 1]$. Figure 3.40a shows the time evolution of the currents, while Fig. 3.40b shows a zoomed time window of the voltages, where—contrasting with the evolutions shown in Fig. 3.36, corresponding to the hard-switching M&S approach—a non-instantaneous transient occurring in the V_{out}—voltage can be appreciated.

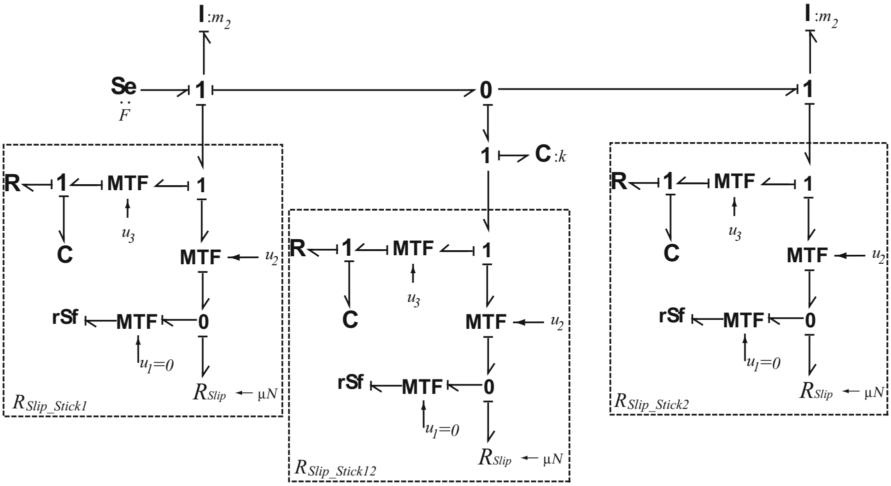


Fig. 3.38 SwBG with a series *R*-*C* replacing the effort residual sink in the internal realization of the SS-Bond

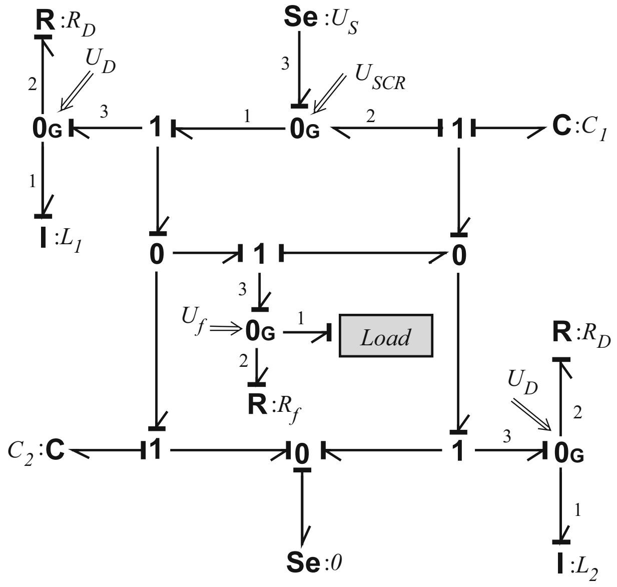


Fig. 3.39 SwBG model of the *Z*-Source DC breaker with fault resistance R_f (shown) and SCR OFF-resistance (hidden in 0G)

The GSJ has been used to model and simulate switched systems with multiples switches. Moreover the task of modelling realistic phenomena, as the short circuit admittance, is quite simple since it can be done by merely adding some extra components. This feature provides to the user an important tool to construct models of different complexity.

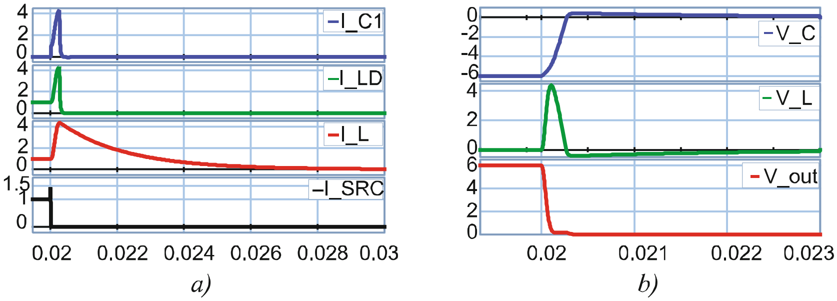


Fig. 3.40 Time evolution for the non-ideal switching approach. (**a**) Current waveforms, in amperes [*KA*], (**b**) Voltage waveforms, in volts [*KV*]

3.6 Conclusions

This chapter discussed two controlled formalisms allowing to represent all the commutation modes of a switched system in a single BG model with fixed causality and well-defined conditions on all its components in all switching modes. Both formalisms can be instantiated to model commutations with both, the hard-switching and the soft-switching approach. They are also suitable for modelling situations where the state variables—including their number, i.e., the system order—are not the same in different switching modes. State jumps can be taken into account by using the principles of (generalized) charge and impulse conservation, either with explicit offline computations previous to the simulation (in the case where the associated restrictions are explicitly solvable) or on-line during the simulation using implicit numerical methods. These features allow for uninterrupted or one-shot simulation runs of the switched system under any scenario.

The first formalism presented, the SS-bond, allowing to handle all commutation modes between two power ports, can represent pre-existing one- and two-port switched structures like the Ideal Switch or Sw-element, and the switchable bond, solving the shortcomings of the latter.

The two dual versions of the GSJ, the second formalism, which are controlled switched bond-graph structures, can represent all the interconnections enforced by commutations involving bond-graph elements around otherwise standard 0- or 1-junctions. These structures can not only represent these junctions in their standard form, but also pre-existing multiport switched bond-graph formalisms like the Controlled Junctions, CJ, and the Switched Power Junctions, SPJ, as particular configurations.

Both, SS-Bond and GSJ, can be implemented in a computer simulation tool directly programming the mathematics of their respective formal definitions or by assembling their internal structures with elements out of the standard BG-library. These are alternatives that can be advantageously used according to the specific modelling and simulation problem at hands.

The advantage of hard-switching yielding a simulation with fixed causality comes with a drawback: each residual sink component adds an algebraic constraint which implies the dynamics being described by a DAE system. These constraints must be calculated numerically at each integration step. When it can be solved analytically offline prior to the simulation, only an explicit computation has to be performed at each step. Otherwise, it must be done implicitly, with the subsequent increment of the computational cost. On the other hand, soft-switching modelling achieved with the inclusion of *R*, *C* and/or *I* elements, i.e., adding parasitic dissipation and/or dynamics, results in numerically stiff models with higher dynamic order that will increase the simulation time. Moreover, in many cases the parasitic components are not related to any physical attribute of the system, what complicates the task of parameterizing them. Thus, the choice between implementing these tools according to the ideal or hard-switching paradigm or the non-ideal or soft-switching approach, even co-existing in a same model, is left to the user's best convenience. It would be the result of a trade-off among the overall computational costs associated with each of the implementations.

References

1. Armstrong-Hélouvry, B., Dupont, P., & Wit, C. C. D. (1994). A survey of models, analysis tools and compensation methods for the control of machines with friction. *Automatica, 30*(7), 1083–1138.
2. Asher, G. (1993). The robust modeling of variable topology circuits using bond graphs. *Proceedings ICBGM'93, the International Conference on Bond Graph Modeling and Simulation*, San Diego, CA.
3. Borutzky, W. (1995). Representing discontinuities by means of sinks of fixed causality. *1995 International Conference on Bond Graph Modeling, ICBGM'95, Proceedings of the 1995 Western Simulation Multiconference*.
4. Borutzky, W. (1995). Discontinuities in a bond graph framework. *Journal of the Franklin Institute, 332*(2), 141–154.
5. Borutzky, W. (2004). *Bond graphs—A methodology for modelling multidisciplinary dynamic systems* (Vols. %1 de %2FS-14). Erlangen, San Diego, CA: SCS Publishing House.
6. Borutzky, W. (2010). *Bond graph methodology—Development and analysis of multidisciplinay dynamic system models* (pp. XXII, 662). London: Springer-Verlag.
7. Borutzky, W. (2015). *Bond graph model-based diagnosis of hybrid systems*. Switzerland: Springer.
8. Borutzky, W., & Cellier, F. E. (1996). Tearing algebraic loops in bond graphs. *Transactions of the Society for Computer Simulation International, 13*(2), 102–115.
9. Borutzky, W., Dauphin-Tanguy, G., & Thoma, J. U. (1995). Advances in bond graph modelling: Theory, software, applications. *Mathematics and Computers in Simulation, 39*(5), 465–475.
10. Branicky, M. S. (1998). Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Transactions on Automatic Control, 43*(4), 475–482.
11. Broenink, J., & Wijbrans, K. (1993). Describing discontinuities in bond graphs. *Proceedings ICBGM'93, the International Conference on Bond Graph Modeling and Simulation*, San Diego, CA.
12. Controllab Products, B. V. (2013). 20-sim. [En línea]. Available from http://www.20sim.com.

13. Corzine, K. A., & Ashton, R. W. (2010). A new z-source dc circuit breaker. *Industrial Electronics (ISIE), 2010 IEEE International Symposium on*.
14. Demír, Y., Poyraz, M., & Kösal, M. (1997). Derivation of state and output equations for systems containing switches and a novel definition of a switch using the bond graph model. *Journal of the Franklin Institute, 334*(2), 191–197.
15. Garcia, J., Dauphin-Tanguy, G., & Rombaut, C. (1997). A bond graph approach for modeling switching losses of power semiconductor devices. *International Conference on Bond Graph Modeling and Simulation (ICBGM'97)*, Phoenix.
16. Gawthrop, P. J. (1997). *Hybrid bond graphs using switched I and C components*. Technical Report CSC-97005. Glasgow: Faculty of Engineering, University of Glasgow.
17. Junco, S., Diéguez, G., & Ramírez, F. (2007). On commutation modeling in Bond Graphs. *Proceedings of the International Conference on Bond Graph Modeling and Simulation*, San Diego, CA.
18. Karnopp, D. (1985). Computer simulation of stick–slip friction in mechanical dynamic systems. *Journal of Dynamic Systems, Measurement, and Control, 107*(1), 100–103.
19. Karnopp, D., & Margolis, D. (1979). Analysis and simulation of planar mechanism systems using bond graphs. *ASME Journal of Mechanical Design, 2*(101), 187–191.
20. Karnopp, D., Margolis, D. L., & Rosenberg, R. C. (2006). *System dynamics: Modeling and simulation of mechatronic systems* (4th ed.). New York, Chichester: Wiley.
21. Margolis, D. (2005). Fixed causality slip–stick friction models for use in simulation of non-linear systems. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 219*(3), 199–206.
22. Mosterman, P., & Biswas, G. (1995). Behaviour generation using model switching. A hybrid bond graph modeling technique. *Proceedings ICBGM'95, the International Conference on Bond Graph Modeling and Simulation*, Las Vegas, NV.
23. Mosterman, P., & Biswas, G. (1998). A theory of discontinuities in physical system models. *Journal of the Franklin Institute, 335*(3), 401–439.
24. Nacusse, M., & Junco, S. (2011). Simplifying switched bond graphs using residual sinks to enforce causality: Application to modeling the z-source inverter. *Mecánica Computacional Industrial Applications (B), 30*(33), 2533–2548.
25. Nacusse, M., & Junco, S. (2013). Switchable structured bond: A bond graph device for modeling power coupling/decoupling of physical systems. *Journal of Computational Science, 5*(3), 450–462.
26. Nacusse, M. A., & Junco, S. J. (2015). Generalized controlled switched bond graph junctions. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 229*(9), 851–866.
27. Ort, J. R., & Martens, H. R. (1973). The properties of bond graph junction structure matrices. *Journal of Dynamic Systems, Measurement, and Control, 1*, 362–367.
28. OSMC (2016). The open source modelica consortium. https://www.openmodelica.org/ [En línea]. Available from https://www.openmodelica.org/.
29. Peng, F. Z. (2003). Z-source inverter. *IEEE Transactions on Industry Applications, 39*(2), 504–510.
30. Rideout, G., & Stein, J. L. (2003). An energy-based approach to parameterizing parasitic elements for eliminating derivative causality. *Proceedings of the ICBGM'03, the International Conference on Bond Graph Modeling and Simulation*, Orlando, FL.
31. Rosenberg, R. C., & McCalla, J. (1991). Power to the user in formulating model equations. *American Society of Mechanical Engineers, Dynamic Systems and Control Division (Publication) DSC, 34*, 25–35, Publ by ASME.
32. Strömberg, J. (1994). *A mode switching modelling philosophy*. Ph.D. thesis, Linköping University, Sweden.
33. Strömberg, J., Top, J. & Söderman, U. (1993). Variable causality in bond graph caused by discrete effects. *Proceedings of the ICBGM'93, the International Conference on Bond Graph Modeling and Simulation*, San Diego, CA.

34. Sun, Z., & Ge, S. S. (2011). *Stability theory of switched dynamical systems*. London: Springer. S. L. Limited, Ed.
35. Umarikar, A. C., & Umanand, L. (2005). Modelling of switching systems in bond graphs using the concept of switched power junctions. *Journal of the Franklin Institute, 342*(2), 131–147.
36. van der Schaft, A. J., & Camlibel, M. K. (2009). A state transfer principle for switching port-Hamiltonian systems. *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*.
37. van der Schaft, A. J., & Schumacher, J. M. (2000). *Introduction to hybrid dynamical systems*. London: Springer. Springer, Ed.
38. Xu, X., & Zhai, G. (2005). Practical stability and stabilization of hybrid and switched systems. *IEEE Transactions on Automatic Control, 50*(11), 1897–1903.

Chapter 4
Dynamic Causality in Hybrid Bond Graphs

Rebecca Margetts and Roger F. Ngwompo

4.1 Introduction

In constructing dynamic systems models, it is frequently desirable to abstract rapidly changing, highly nonlinear behaviour to a discontinuity. When rapidly changing behaviour is described by a continuous differential equation, it must be integrated using very small time steps in order to achieve any level of accuracy. Abstracting rapidly changing behaviour to a discontinuous equation can therefore aid solvability and improve computer simulation times. In addition, a user may find it intuitive to think of certain elements (like an electrical switch or hydraulic valve) or phenomena (such as contact, dry friction or breakage) as discontinuous.

Hybrid models are those containing any continuous and any discontinuous behaviour. They can be visualised as continuous modes on areas of state space linked by a discontinuous state mapping [42], and described as a hybrid automaton, i.e. one that contains both finite and continuous state spaces [58]. The dynamics consist of discrete transitions plus an evolution of the continuous part in each location.

Definition 4.1. Hybrid Model: any model which describes both continuous and discontinuous behaviour.

The terms 'hybrid' and 'switched' system are used almost interchangeably in the bond graph literature, but there is a subtle difference. Switching models 'comprise a family of dynamical subsystems together with a switching signal determining the active system at a current time' [61]. They are a subset of hybrid systems,

R. Margetts (✉)
School of Engineering, University of Lincoln, Brayford Pool, Lincoln LN6 7TS, UK
e-mail: rmargetts@lincoln.ac.uk

R.F. Ngwompo
Department of Mechanical Engineering, University of Bath, Claverton Down, Bath BA2 7AY, UK
e-mail: r.f.ngwompo@bath.ac.uk

W. Borutzky (ed.), *Bond Graphs for Modelling, Control and Fault Diagnosis of Engineering Systems*, DOI 10.1007/978-3-319-47434-2_4

where there is some discontinuous behaviour modelled by an on/off switch or other binary signal. Switched models can be used to describe multimodal systems and variable-structure systems. The hybrid bond graph therefore usually yields a switched system, and some variants are referred to as switching bond graphs.

Definition 4.2. Switching Model: a subset of hybrid model, which contains continuous equations and binary switching devices. The 'switches' select the active continuous equation(s) or behaviours at a given time.

Branicky et al. [7] categorise hybrid models into Switching and Impulse models, which can be Controlled or Autonomous. Switching models are defined as those where the vector field changes discontinuously when the state hits a boundary. Impulse models are those where the continuous state changes impulsively on hitting prescribed regions of state space, i.e. there is a 'jump' between the continuous equations in state space. The classic example is Newton's Collision law, where the state of a body changes from positive to negative velocity on impact, and any dissipative effects are accounted for by a coefficient of restitution. These types of models have been created by some hybrid bond graph practitioners [36, 65].

Definition 4.3. Impulse Model: a subset of hybrid model where the state changes impulsively, i.e. there is an impulse loss on commutation.

Variable topology systems are those where the size of the state equation matrices changes, such as contact (Fig. 4.1). They are frequently represented by impulse models [7] such as Newton's Collision Law with restitution. This type of model exhibits an impulsive 'jump' in state space on commutation, which violates the conservation of energy fundamental to the bond graph. The state variables are unknown after commutation, necessitating the use of state reinitialisation [35, 36] and state estimation techniques [45].

A number of hybrid and switching bond graphs have been proposed to model discontinuities in the bond graph framework. From early on in the development of bond graphs, there was a need to model discontinuities in the form of elements like switches and valves. Borutzky gives an overview in his text [6], and Margetts presents a survey and discussion [29]. The different methods appear to reflect the different backgrounds and motivations of the users, with switched sources and controlled/switched junctions falling into more common usage than the other methods. One of the main differences between methods is the treatment of dynamic causality, and its consequences for analysis and computation.

4.1.1 Dynamic Causality

Bond graphs are constructed as acausal models, and an ideal computational causality is assigned to the model once complete. However, in a hybrid system the ideal causal assignment can change with commutation. For example, two rigid bodies might

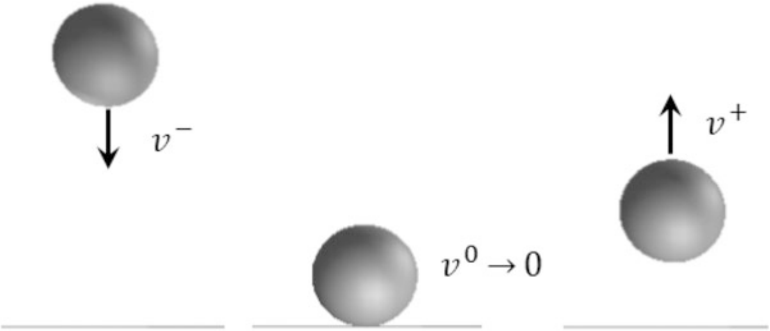


Fig. 4.1 The three 'Modes' of a bouncing ball: falling, contact and rebound

be in integral causality, but when they make contact one body will be forced into derivative causality (reflecting a genuine kinematic constraint: the two rigid bodies now act as one).

Dynamic causality is a feature of all ideal switching. Variable topology is an example of this, where the model changes significantly with commutation, e.g. a contact problem where the state equations change size.

Dynamic causality can be minimised and controlled using 'Causality Resistance' as originally proposed by Asher [2], to give a causally static model. This technique has been used successfully in the commercial bond graph package 20-Sim [8, 9, 59]. Dynamic causality can also be minimised by revising the causality assignment procedure, i.e. the use of Hybrid-SCAP [28].

The debate regarding static versus dynamic causality is ongoing at the time of writing. The body of work using switched sources and controlled junctions generally accepts dynamic causality, but many practitioners use methods that give static or near-static causality for ease of representation and computation. However, this chapter argues the case for embracing dynamic causality.

One of the many strengths of the bond graph method is its physical relevance. It is an Idealised Physical Modelling method, where an engineer uses the physical model to build an acausal representation, from which a mathematical model is derived. It aligns with the principles of behavioural modelling and object orientation, as opposed to the 'input-output thinking' used in block diagrams and signal flow graphs which results in the 'control/physics barrier' [62]. From early in the bond graph's inception, structural analysis and causality exploitation were used to provide information about the system [34, 47]. 'Structural Analysis' refers to information that can be obtained from the bond graph structure, either by investigating the graph or the junction structure matrix. This information includes the form of the state equations, solvability, etc., and is analogous to the structural analysis of state matrices in control engineering. An analyst can easily see derivative causality and algebraic loops in a bond graph's causal assignment, and causality can guide the engineer in assembling submodels and suggesting suitable test conditions. There is also the potential to look at interaction between nonlinear constitutive equations and uniqueness of system response [47].

It is therefore logical that a hybrid bond graph should continue to uphold these values and be suitable for qualitative analysis. Although causally dynamic models may at first appear difficult to simulate, they do provide valuable information to the user. Constraining causality is a move away from the real physics of the system. Adding parasitic elements or causality resistance can create an overly complex model with undesirable high-frequency dynamics [11]. The use of switched or modulating resistance and transformer components wrongly implies that switching is dissipative [38]. And there is a danger that compliances and resistances may be added purely to aid computation with no consideration of the physical system: a modeller can unwittingly create physically meaningless or computationally inefficient models. In contrast, a causally dynamic model offers engineering insight [14].

4.1.2 Categorising Discontinuities

A discontinuity is an abstraction made in order to simplify a model. It is possible to model any system using continuous functions. Discontinuities are therefore artificial, and made at the discretion of the user. Their purpose is to simplify the equations used to describe a system's behaviour; where a system's behaviour changes rapidly with time, describing that change as a discontinuity can improve simulation time, and aid engineering insight and analysis. They usually describe highly nonlinear behaviour which would be difficult to describe and time-consuming to compute using continuous functions. They can also describe variable topology problems, which are where the equations used for each mode of operation change significantly, with varying numbers of states and boundary conditions (for example, contact).

The term 'discontinuity' is fairly vague, and so a classification is made to aid application to engineering problems. Discontinuities are often classified as ideal (no losses) or non-ideal (associated with an energy loss) [46], switching or impulse [7] or according to whether they are autonomous or externally controlled. If they are assumed to be controlled by some form of automaton, they can be classified according to whether the controlling automata are time-scale dependant or parameter dependent [40]. Here, an additional distinction is made between structural and parametric discontinuities [29]. This distinction is necessary to describe where in the model (and underlying equations) the discontinuity should occur: between elements or internal to an element.

Definition 4.4. Structural Discontinuities: discontinuities which occur when parts of the model are connected or disconnected, interrupting power flow between components. These discontinuities often give rise to variable topology models.

Engineering examples of this type of discontinuity are the hydraulic valve, mechanical clutch, ideal electrical switch or contact between bodies.

Definition 4.5. Parametric Discontinuities: discontinuities which occur when an element has a highly nonlinear constitutive equation, and the user has abstracted this to a piecewise-continuous function. The structure of the model is unchanged, it is the equation describing the behaviour of an element which changes.

Common examples of parametric discontinuities are dry friction, tyre forces, a nonlinear damper 'breaking out' or saturation of an electrical capacitor or hydraulic accumulator.

These two types of discontinuity can be represented differently in a hybrid bond graph: a controlled junction with dynamic causality for structural switching and a controlled element for parametric switching.

In many cases—particularly the mechanical domain—the distinction between structural and parametric switching is clear. However, there are cases where it is less so. An electrical switch is physically an element which the user inserts into a circuit, and is often visualised in control theory as a discontinuous input, hence the use of switching sources and elements in the literature. Consequently, there is a case for treating it as parametric switching. Here the dynamic causal assignment is key: disconnecting a voltage or current source can force electrical storage elements to discharge, which is consistent with them switching to derivative causality. The controlled junction proposed for structural switching clearly shows where structure is disconnected and ideal causality assignment changes with commutation.

4.2 Structural Discontinuities

Structural switching activates or deactivates part of a system, and a controlled junction can be used to (dis)connect or (de)activate part of the model accordingly. Controlled junctions as described by Mosterman and Biswas [39] are selected to represent structural switching because they clearly show where structure connects and disconnects, and breaks the path of power flow. They are preferable to the use of switched sources which imply the switch is an energy-processing element when it is, in fact, a control element [6, 37]. This is not only important from the point of view of engineering insight, but also the controlled junction lends itself to being represented in the junction structure matrix and hence developing hybrid system equations.

4.2.1 The Controlled Junction

A controlled junction behaves as a normal 1- or 0-junction when ON and a source of zero flow or effort (respectively) when OFF. A controlled 1-junction is therefore used to break or inhibit flow (for example, an electrical switch which breaks the flow of current) and a controlled 0-junction is used to inhibit effort (for example, a clutch or other physical non-contact in a mechanical system). This always gives

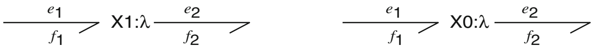


Fig. 4.2 Bond graph representation of switched junctions X1 and X0

rise to dynamic causality on one of the attached bonds. The commonly accepted notation for controlled junctions is X1 and X0, which will be used in this paper.

Based on the above description, controlled junctions X1 and X0 can be formally defined as elements with associated Boolean parameters. They are initially defined as 2-ports for clarity, and the definition can easily be extended to more than 2 ports. The bond graph representations of controlled junctions X1 and X0 are as shown in Fig. 4.2, and their defining relationships are given by Eqs. (4.1) and (4.2), respectively.

$$\begin{cases} \lambda f_1 = \lambda f_2, \\ \lambda(e_1 - e_2) = 0, \\ \bar{\lambda} f_1 = 0, \\ \bar{\lambda} f_2 = 0 \end{cases} \tag{4.1}$$

$$\begin{cases} \lambda e_1 = \lambda e_2, \\ \lambda(f_1 - f_2) = 0, \\ \bar{\lambda} e_1 = 0, \\ \bar{\lambda} e_2 = 0 \end{cases} \tag{4.2}$$

The Boolean parameter λ selects the set of equations that are valid given the state of the switch: 1 when the switch is ON and 0 when the switch is OFF. For each controlled junction, the defining Eqs. (4.1) and (4.2) lead to three possible causal configurations:

- Two causal configurations when the switch is ON, i.e. (first two equations equivalent to a normal 1 or 0 junction)
- a unique causal configuration when the switch is OFF, i.e. (last two equations equivalent to null sources of flow or null sources of effort imposed by the element to both power ports with conjugate variables externally imposed to the element).

In switching between ON and OFF states, the causal assignment around a controlled junction must change. In the ON state, where it behaves as a regular junction, there must be a causal input (i.e. a bond with a causal stroke defining the common effort for a 0-junction or common flow for a 1-junction). In the OFF state, where the controlled junction becomes a null source on each incident bond, there is no causal input and the causal assignment on that one bond changes. This is known as dynamic causality, and using this definition of a controlled junction it is unavoidable.

4.2.2 Simplification of the Hybrid Bond Graph

If a bond graph is constructed from the schematic diagram of a system, there is often potential to simplify the bond graph model. Since a controlled junction only behaves as a junction in one state, it cannot be eliminated and the rules for simplifying bond graphs must be augmented as follows:

Rule 1: 2-Port Controlled Junctions
A controlled junction with only two ports cannot be removed and replaced with a single bond (whereas a regular junction could).

A regular junction with 2-ports can be replaced by a single bond, since the efforts and flows on the two incident bonds are equal. A controlled junction with 2-ports connects and disconnects its incident bonds with commutation: in principle like a switching bond or Boolean-modulated transformer (with dynamic causality). Removing the controlled junction would result in the surrounding structure being connected at all times.

Rule 2: Neighbouring Junctions: Controlled and Regular Junctions
When a regular junction and controlled junction of the same type are neighbouring, they can be merged into a single controlled junction.
That is, when a 1-junction and X1-junction are next to each other, they can be merged into a single X1-junction.
Likewise, a 0-junction and X0-junction next to each other can be merged into a single X0-junction.

When two like regular junctions are next to each other, they can be merged into a single junction. When one of those junctions is controlled, the commutating behaviour must be retained. This simplification results in elements being disconnected with commutation, whereas they would have remained connected to some substructure without the simplification (shown in Fig. 4.3).

Rule 3: Neighbouring Junctions: Multiple Controlled Junctions
When neighbouring controlled junctions have two ports only, they can be combined into a single controlled junction. This controlled junction is ON only when the states of both the constituent controlled junctions are ON.
When neighbouring controlled junctions have more than two ports, they cannot be combined. This is because the power to the incident elements or subsystems depends on the state of the individual controlled junction.

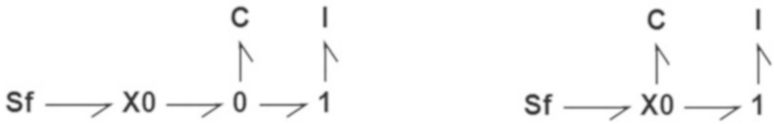


Fig. 4.3 An example subsystem with neighbouring regular and controlled junctions: unsimplified (*left*) and simplified (*right*)

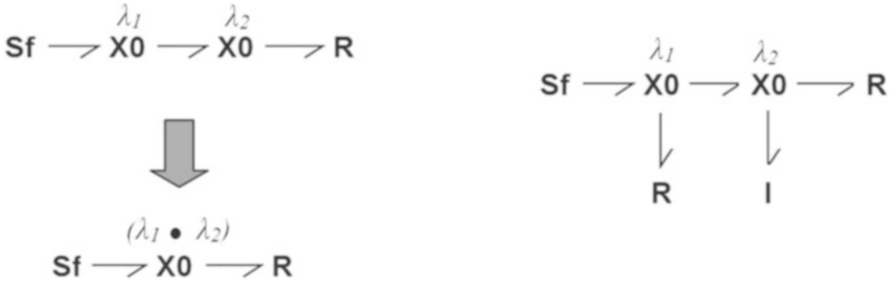


Fig. 4.4 An example subsystem with neighbouring controlled junctions: 2-port junctions, simplified to a single junction which is ON when λ_1 AND λ_2 are true (*left*) and additional elements giving 3-port junctions (*right*)

Figure 4.4 demonstrates how two neighbouring 2-port controlled junctions can be combined: the power source is only connected to the resistor when controlled junction 1 AND controlled junction 2 are ON. The causal conflict arising between the junctions when both are OFF may be ignored.

However, when the controlled junctions have additional elements attached, it is no longer appropriate to combine them. There is power flow across each junction when it is ON and the other is OFF. The two controlled junctions cannot be combined in any manner which would reflect this behaviour.

Structure which adds nothing to the model can frequently be removed. This often happens in the case of electrical and hydraulics circuits where there is a return line to a zero ground or open tank. Ground parts are source elements which also act as a sink. For example, a mechanical ground is represented by a Sf-element (which has zero velocity and is a sink for force), and grounds in other domains are represented by Se-elements (e.g. an electrical ground, which is a source of 0V and a sink for current). They are usually null sources, but can have nonzero values (such as a pressurised hydraulic tank or undulating mechanical ground).

Rule 4: Removal of Ground Parts

When a controlled junction is positioned between a dissimilar ground and the main structure, it is not appropriate to remove the ground. That is, a null source of flow connected via an X0-junction cannot be removed. Likewise, a null source of effort connected via an X1-junction cannot be removed.

When the ground or tank is a null source, and it is not a causal input to the structure of interest, it can be deleted. An example is given in Fig. 4.5: the ground is a source of zero effort, and it adds nothing to the 1-junctions it is attached to (about which efforts are summed).

When a controlled junction exists between the ground and the main structure, it may not be appropriate to remove the ground. For example, in the system in Fig. 4.6, an electrical switch (represented by a X1-junction) could be inserted so that the resistance is now a non-ideal switch, shown in Fig. 4.6. In real terms, this breaks the circuit and changes its behaviour. In bond graph terms, this means that

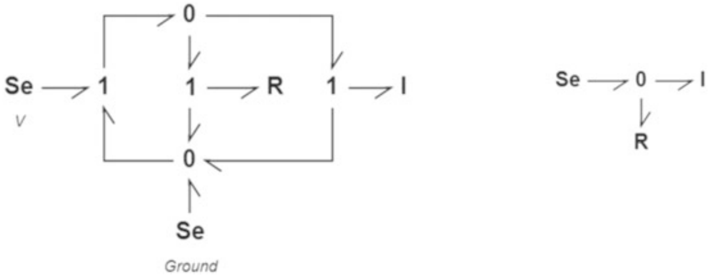
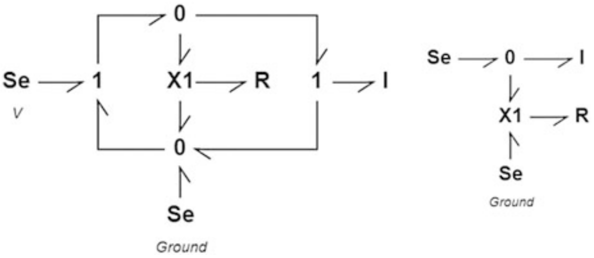


Fig. 4.5 An example system with a ground: full model (*left*) and simplified (*right*)

Fig. 4.6 An example system with a ground and a controlled junction: full model (*left*) and simplified (*right*)



incident structure has a zero flow imposed on it when the X1-junction is OFF, rather than a zero effort (from the ground) which will have implications for the causal assignment. In simplifying the bond graph, the controlled junction and null source group must therefore be kept, because they can have a significant effect on the system.

4.2.3 *The Dynamic Causality Assignment Procedure*

Causality in a bond graph is typically assigned using the Sequential Causality Assignment Procedure (SCAP) [22]. Using controlled junctions, dynamic causality is unavoidable (N.b. it can be minimised using Hybrid-SCAP and a variant of the controlled junction which is deleted when ‘OFF’ [28] but this can result in hanging junctions [29]).

The causality assignment procedure for the hybrid bond graph proposed in this paper starts with a reference mode of operation. This is defined with a maximum number of elements in integral causality, and controlled junctions preferably ON. This is the mode which should be easiest to simulate. Deviations from this reference due to dynamic causality are marked as dashed causal strokes. This enables the user to see the effects of commutation on causality, and aids in equation generation. The Dynamic Sequential Causality Assignment Procedure (DSCAP) to represent all modes of a hybrid bond graph model can be summarised in the following procedure:

Procedure 1: Dynamic Sequential Causality Assignment Procedure (DSCAP) for hybrid bond graph

Step 1. Assign causality according to SCAP with preferred integral causality, stopping when a controlled junction is reached. That is, start by assigning causality to a source element, and propagate causality throughout the bond graph as far as any controlled junctions. Repeat for other source elements, and then for any storage elements which have not yet been assigned causality. If causal conflict occurs in this stage, the model should be changed.

The causal assignment from step 1 may dictate whether some switches are ON or OFF.

Step 2. Choose a controlled junction which does not have its causality fully assigned. Assign causality around the controlled junctions assuming the switch to be ON and propagate as far as possible. Repeat this stage until all controlled junctions have their causality fully assigned.

Step 3. Finish propagating causality throughout the bond graph to any resistance elements or remaining bonds and propagate as far as possible.

Step 4. Taking each controlled junction in turn, consider the causality assignment when it is in the other state to the reference configuration. Mark this causality assignment with a dashed causal stroke, and propagate throughout the bond graph. If causal conflict occurs in this stage, then the other state of the controlled junction is not allowed.

Remark. Causal propagations in step 2 and step 4 of the algorithm above may dictate the state (ON or OFF) of some controlled junctions as a result of the assigned state of others. This reveals some constraints in the state of switches indicating the allowed configurations or physically feasible modes of operation.

Figure 4.7 shows a simple example of the effect of the causality assignment around a controlled junction when ON and OFF.

4.2.4 Dynamic Causality Notation and the Reference Configuration

The dynamic causality notation differs from that typically used [5] as it is designed to aid structural analysis and equation generation by hand. A reference configuration is used which indicates the mode of operation where most storage elements are in integral causality (giving information about model order and solvability). The dashed causal stroke notation aids the user in identifying regions of dynamic

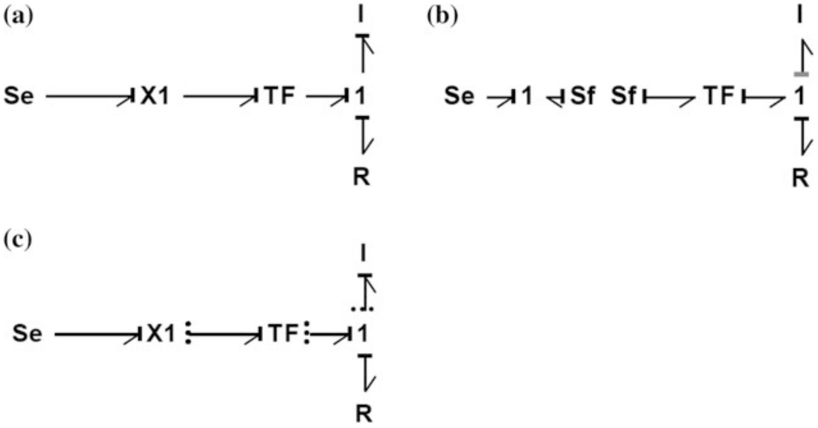


Fig. 4.7 An example of causality assignments and their effect around a controlled junction. (**a**) The junction in the ON (reference) mode. (**b**) The junction shown by null sources when OFF. (**c**) Dynamic causality notation

causality. Rather than assuming 2^n modes of operation (where n is the number of switches), the user can inspect individual regions of the chart and generate a truth table or similar for a relatively small subsystem.

Note that the parameter λ of a switch indicates the absolute state of the switch, i.e. $\lambda = 1$ when the switch is ON and $\lambda = 0$ when the switch is OFF.

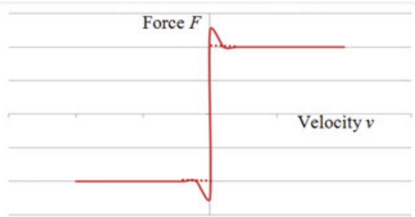
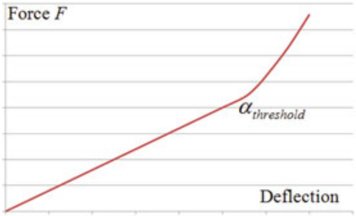
4.3 Parametric Discontinuities

Parametric switching has been defined as the case where an element has a piecewise-continuous constitutive equation. These may be hard nonlinearities, where the behaviour of an element changes so quickly that it can be considered instantaneous. Alternatively, they can occur where some relationship (gained via empirical data or a high-order function) is best described using a piecewise-continuous function. Classic examples are friction and tyre lateral stiffness, shown in Table 4.1.

The controlled element is recommended for the modelling of parametric switching. It should not be confused with the existing switched element, which has an on/off behaviour [20].

Parametric switching can be considered as mode switching, i.e. a collection of continuous modes of operation. These are controlled by an automaton, petri-net or similar, which allows the system to switch between modes of operation. Mode switching is historically modelled by a ‘tree’ of ideal switches and elements. Each element gives the constitutive equation for a specific mode of operation, and the ideal switches (de)activate it as required. Naturally, only one ideal switch can be ON at any time during a simulation. Strömberg [52] formulates mode switching

Table 4.1 Examples of piecewise-continuous equations

'Hard' Friction	Generic Tyre Static Load/Deflection Curves
	
Where F_N is normal force and μ is friction coefficient	Where α is slip angle, related to linear deflection

trees of switched sources, and Mosterman and Biswas [39] present a multi-bond controlled junction selecting a continuous bond graph element from a number of possibilities.

Mode switching has a conceptual advantage in that it aids the development of finite state automata for simulation. However, the 'tree' notation means a model can rapidly grow to a vast size with multiple inputs and outputs for all possible modes of operation. This makes it less ideal for structural analysis and equation generation purposes. The multi-bond notation suggested by Mosterman and Biswas goes some way to controlling this, but it is a little confusing because multi-bond notation is typically used for multiple degrees of freedom in a model. Their idea is used as a basis for the controlled element defined here.

4.3.1 The Controlled Element

Consider an element with a piecewise-continuous constitutive function. A mode-switching tree can be constructed using the controlled junctions with associated Boolean terms (as used for structural switching), as shown in Fig. 4.8. Note that a resistance element is shown, but the principle holds true for inertia and compliance elements.

In this tree, controlled junctions (de)activate the modes of operation, which are given by resistance elements on each branch. These 'branches' are then connected by a regular junction which sums the output values.

- In Fig. 4.8 (left) efforts are summed about a 1-junction: these efforts are the effort exerted by the resistance when a junction is ON plus the zero efforts exerted by the X0-junctions when they are OFF.

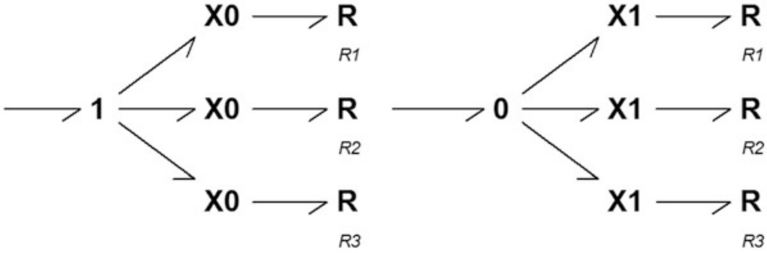


Fig. 4.8 Bond graph 'Trees' for a piecewise linear resistance element, assuming three modes of operation: a 'Tree' of X0-junctions (*left*) and a 'Tree' of X1-junctions (*right*)

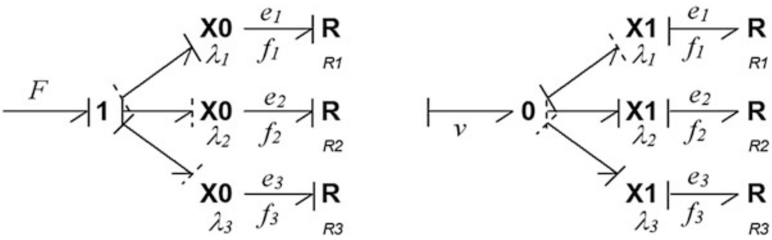


Fig. 4.9 The piecewise linear resistance element subsystem, showing quantities used in equation generation

- In Fig. 4.8 (right), it is flows which are summed around a zero junction: these flows are the flow exerted by the resistance when a junction is ON plus the zero flows exerted by the X1-junctions when they are OFF.

In a bond graph tree it is important to note that the controlled junctions are constrained so that only one may be ON at any time.

In order to condense the 'tree' into a single controlled element, consider the underlying equations. Quantities are shown in Fig. 4.9, which also includes some source elements in order to obtain the equations. A reference configuration of $\lambda_1 = 1$, $\lambda_2 = 0$, $\lambda_3 = 0$ is arbitrarily assumed. Note that dynamic causality is internal to the tree: there is static causality on the resistance elements and the input bond.

This controlled element has the general constitutive function:

$$\text{output} = \sum_{n=1}^{i} \lambda_n \Phi_n(\text{input}) \tag{4.3}$$

where n is the number of branches to the tree, λ_n is the Boolean term associated with nth controlled junction and Φ_n is the constitutive function of the nth element.

The controlled element may be in dynamic causality (i.e. the output is effort in some modes and flow in others) it can be treated in the same way as a standard element in dynamic causality, i.e. having two input/output pairs for the two causal assignments [32]. For example, a hydraulic accumulator is a compliance element with an effort output in normal operation, but when it saturates it becomes a source of zero flow.

4.4 Deriving the Mixed-Boolean State Equation

Generating a state equation from a bond graph is well established. Here, a similar process is used to show that a causally dynamic hybrid bond graph can be used to generate a state equation which is mixed-Boolean.

4.4.1 Pseudo-States and Dynamic Causality

For a regular (causally static) bond graph, the inputs and outputs to the system from the various elements are used in generating equations. Specifically, the inputs to the system from the storage fields (i.e. the outputs of the compliance and inertia elements in integral causality) are usually taken as the time-derivatives of the state variables. The state variables are consequently displacement (for compliance elements) and momentum (for inertia elements).

When elements are in derivative causality, the state equations are no longer independent: there are dependent states associated with the elements in derivative causality which yield algebraic equations. Pseudo-state variables (associated with each element in derivative causality) have therefore been used to generate an implicit mathematical model containing the relevant algebraic constraints [12, 53].

When causality is dynamic, storage elements may switch from integral to derivative causality, and the inputs and outputs of the resistance elements may consequently reverse. The resulting state space matrices can change size depending on the mode. Storage elements in dynamic causality are therefore described using a variable for each of the two possible causal assignments: a state variable for the integral causality case and a pseudo-state variable for derivative causality. The model then describes all possible modes of operation by including input and output variables for both possible states of an element in dynamic causality. These are (de)activated in the appropriate modes of operation.

4.4.2 The General Hybrid Bond Graph

A causal bond graph model can be represented in matrix format, as a Junction Structure Matrix (JSM) consisting of ones and zeros which relate the system inputs and outputs. The JSM based on the Paynter Junction Structure is used here, since it has reached common use in bond graph structural analysis. The coefficients in

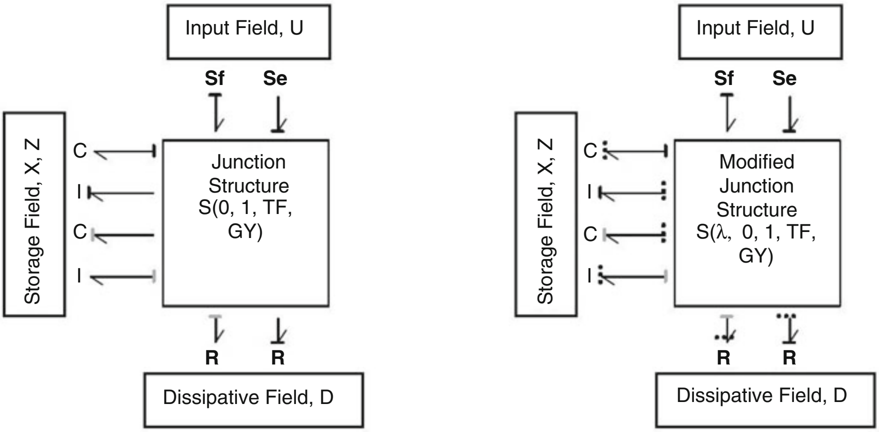


Fig. 4.10 The junction structure matrix and generalised bond graph: general junction structure (*left*) and hybrid junction structure incorporating switching coefficients and dynamic causality (*right*)

the transformer field (representing any transformer or gyrator elements, sometimes expressed outside the junction structure matrix) are brought inside the JSM to give terms other than one and zero.

The General Bond Graph structure is shown in Fig. 4.10, with a modified ‘hybrid’ version to capture structural switching behaviour and the induced dynamic causality. There are two significant differences:

1. Using the Dynamic Sequential Causality Assignment Procedure (DSCAP), the causal hybrid bond graph displays some elements with static causality and some with dynamic causality (represented by dashed causal strokes, as shown in Fig. 4.10). The Hybrid Junction Structure Matrix (relating all possible system inputs and outputs) and state equation can be derived from this representation.
2. The Hybrid Junction Structure Matrix **S** contains Boolean parameters λ indicating the state of controlled junctions (used to describe structural switching). These Boolean ‘switching terms’ in the submatrices of **S** will therefore be carried through into the state equations derived from it.

Note that the Boolean terms λ appearing in the Junction Structure Matrix reflect controlled junctions between elements, and indicate where casual paths are severed or connected with commutation. There may be additional Boolean terms in the storage and resistance fields where parametric discontinuities exist within controlled elements.

Figure 4.11 shows the key variables used in the causally dynamic hybrid bond graph, which are defined as follows. Note that the inputs to the elements are the outputs from the junction structure, and vice versa.

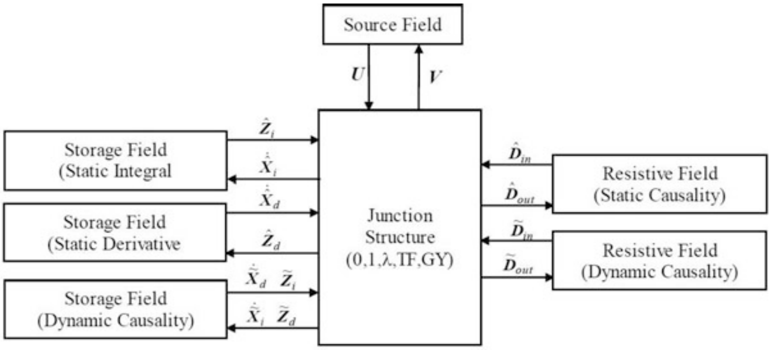

Source Field
$\boldsymbol{U}$
$\boldsymbol{V}$
Storage Field
(Static Integral
Storage Field
(Static Derivative
Storage Field
(Dynamic Causality)
$\hat{\boldsymbol{Z}}_i$
$\dot{\hat{\boldsymbol{X}}}_i$
$\dot{\hat{\boldsymbol{X}}}_d$
$\hat{\boldsymbol{Z}}_d$
$\dot{\tilde{\boldsymbol{X}}}_d$ $\tilde{\boldsymbol{Z}}_i$
$\dot{\tilde{\boldsymbol{X}}}_i$ $\tilde{\boldsymbol{Z}}_d$
Junction
Structure
(0,1,λ,TF,GY)
$\hat{\boldsymbol{D}}_{in}$
$\hat{\boldsymbol{D}}_{out}$
$\tilde{\boldsymbol{D}}_{in}$
$\tilde{\boldsymbol{D}}_{out}$
Resistive Field
(Static Causality)
Resistive Field
(Dynamic Causality)

Fig. 4.11 Quantities used in hybrid junction structure matrix and subsequent development

1. Elements with static causality have the usually defined variables:
   - Input vectors denoted $\dot{\hat{\boldsymbol{X}}}_i$ composed of $\dot{p}$ and $\dot{q}$ on I and C elements in integral causality, $\hat{\boldsymbol{Z}}_d$ composed of f and e on I and C elements in derivative causality, and $\hat{\boldsymbol{D}}_{\text{out}}$ composed of effort or flow variables into dissipative elements (N.b. the output from the JSM is the input to the element).
   - Output vectors denoted $\hat{\boldsymbol{Z}}_i$ and $\dot{\hat{\boldsymbol{X}}}_d$ for storage elements, and $\hat{\boldsymbol{D}}_{\text{in}}$ for dissipative elements
2. However, dynamic causality is captured in the block diagram by specifying two sets of inputs and output variables:
   - Two input vectors. For storage elements in dynamic causality, these inputs are $\dot{\tilde{\boldsymbol{X}}}_i$ for the integral causality case and $\tilde{\boldsymbol{Z}}_d$ for the derivative causality case, and are composed of $\dot{p}$, $\dot{q}$, f and e. For dissipative elements in dynamic causality, there is an effort input $\tilde{\boldsymbol{D}}_{e(\text{out})}$ and flow output $\tilde{\boldsymbol{D}}_{f(\text{out})}$. In any single mode of operation, one input is active and the other is redundant.
   - Two output vectors which are the complements of the inputs above. For storage elements in dynamic causality, these are $\dot{\tilde{\boldsymbol{X}}}_d$ and $\tilde{\boldsymbol{Z}}_i$ composed of $\dot{p}$, $\dot{q}$, f and e. For dissipative elements in dynamic causality, there is an effort output $\tilde{\boldsymbol{D}}_{e(\text{in})}$ and flow output $\tilde{\boldsymbol{D}}_{f(\text{in})}$. Again, in any single mode of operation, one output is active and the other is redundant.

   It is worth noting that an element can only have two modes of operation (flow input/effort output and effort input/flow output), although a model can have several modes of operation overall if it contains multiple controlled junctions.

Controlled junctions in the bond graph are assigned Boolean variables λ in the Junction Structure. λ has a value of 1 when the junction is ON and 0 when OFF, signifying that there is a connection between two quantities when the junction is ON. A single bond graph therefore represents all possible modes of operation and causal assignments. Vectors $\boldsymbol{X}_i = [\hat{\boldsymbol{X}}_i\ \tilde{\boldsymbol{X}}_i]^T$ and $\boldsymbol{X}_d = [\hat{\boldsymbol{X}}_d\ \tilde{\boldsymbol{X}}_d]^T$ are the state and pseudo-state of the storage fields in integral and derivative causality, respectively. $\boldsymbol{Z}_i = [\hat{\boldsymbol{Z}}_i\ \tilde{\boldsymbol{Z}}_i]^T$ and $\boldsymbol{Z}_d = [\hat{\boldsymbol{Z}}_d\ \tilde{\boldsymbol{Z}}_d]^T$ are the complementary vectors of these states (shown in Fig. 4.10), related by $\boldsymbol{Z}_i = \boldsymbol{F}_i\boldsymbol{X}_i$ and $\boldsymbol{Z}_d = \boldsymbol{F}_d\boldsymbol{X}_d$. Resistive field also have inputs and outputs $\boldsymbol{D}_{\text{in}} = [\hat{\boldsymbol{D}}_{\text{in}}\,\tilde{\boldsymbol{D}}_{\text{in}}]^T$ and $\boldsymbol{D}_{\text{out}} = [\hat{\boldsymbol{D}}_{\text{out}}\,\tilde{\boldsymbol{D}}_{\text{out}}]^T$ related by $\boldsymbol{D}_{\text{in}} = \boldsymbol{L}\boldsymbol{D}_{\text{out}}$.

There are two inputs and two outputs for each 1-port element in dynamic causality. The input/output sets are exclusive of each other, and the Boolean terms in the Hybrid Junction Structure Matrix (HJSM) will activate one of these for each mode of operation.

In order to establish which outputs of the junction structure are active, the vector of outputs must be multiplied by a diagonal matrix of Boolean expressions $\Lambda(\boldsymbol{\lambda})$. In any one mode of operation, some rows of the matrices will be set to zeros and others will give the Junction Structure for that mode. Therefore, outputs which are in static causality are assigned a 1 in the diagonal of the matrix $\Lambda(\boldsymbol{\lambda})$ because they are fixed outputs, while variables associated with elements in dynamic causality are assigned a Boolean function $f(\lambda)$ determined by the combination of the switch parameters λ that dictate the output status of the variable. For each Boolean term $f(\lambda)$, there will always be a NOT term $\overline{f(\lambda)}$ present in the matrix $\Lambda(\boldsymbol{\lambda})$ which activates another row to describe the dynamic element's behaviour in its other state.

In order to construct the matrix $\Lambda(\boldsymbol{\lambda})$, consider each 1-port element in Dynamic Causality in turn, determine any causal path between this elements and the controlled junctions and report the state of the switch and the output variable in a truth table. A truth table (e.g. Table 4.2) can therefore be used to construct the combination of states, and hence function of Boolean variables, that result in each causal change. For example, if a storage element is in integral causality only when two switches are ON, this could be expressed by assigning the state variable a term in $\Lambda(\boldsymbol{\lambda})$ of $(\lambda_1 \cdot \lambda_2)$, i.e. switch 1 AND switch 2 are true or ON (Table 4.2). The pseudo-state complementary variable Z_d would therefore be assigned $(\overline{\lambda_1 \cdot \lambda_2})$ because the element is in derivative causality when switch 1 AND switch 2 is NOT true, i.e. OFF.

Often a controlled junction simply creates a path of Dynamic Causality between it and a nearby element, and the term in $\Lambda(\boldsymbol{\lambda})$ can be quickly and easily assessed. There is the potential to reduce the amount of work required to obtain the equations by modularising and reusing submodels for larger systems.

The hybrid JSM is thus represented mathematically as:

$$\begin{bmatrix} \boldsymbol{\Lambda}_{11} & 0 & 0 \\ 0 & \boldsymbol{\Lambda}_{22} & 0 \\ 0 & 0 & \boldsymbol{\Lambda}_{33} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{X}}_i \\ \boldsymbol{Z}_d \\ \boldsymbol{D}_{\text{out}} \end{bmatrix} = \begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} & \mathbf{S}_{13} & \mathbf{S}_{14} \\ -\mathbf{S}_{12}^T & 0 & 0 & \mathbf{S}_{24} \\ -\mathbf{S}_{13}^T & 0 & \mathbf{S}_{33} & \mathbf{S}_{34} \end{bmatrix} \begin{bmatrix} \boldsymbol{Z}_i \\ \dot{\boldsymbol{X}}_d \\ \boldsymbol{D}_{\text{in}} \\ \boldsymbol{U} \end{bmatrix} \tag{4.4}$$

Table 4.2 Example truth table for two switches

Switch 1	Switch 2	Causality	Output	Term in $\Lambda(\boldsymbol{\lambda})$
0	0	Derivative		
0	1	Derivative	Z_d	$(\overline{\lambda_1 \cdot \lambda_2})$
1	0	Derivative		
1	1	Integral	$\dot{X}_i$	$(\lambda_1 \cdot \lambda_2)$

4.4.3 Notation and the Reference Configuration

A reference configuration has been used to aid the construction of the causally dynamic bond graph, and to act as a basis for the proposed dynamic causality notation. However, the Junction Structure matrix encapsulates all possible modes of operation and it is therefore of little consequence which mode is selected for the reference. The parameter λ of a switch indicates the absolute state of the switch, i.e. $\lambda = 1$ when the switch is ON and $\lambda = 0$ when the switch is OFF.

4.4.4 The Unique Hybrid Implicit Equation

The state equations express the time-derivatives of the states and (where there is derivative causality) the pseudo-states—$\boldsymbol{X}_i$ and $\boldsymbol{X}_d$—in terms of their derivatives and the system inputs $\boldsymbol{U}$. An implicit form can be derived from the junction structure matrix which, like the JSM, is mixed-Boolean. The LTI form remains valid because the switching behaviour is not necessarily a function of time: the equations capture the model at all time points.

$$\mathbf{E}(\boldsymbol{\Lambda})\dot{\boldsymbol{X}} = \mathbf{A}(\boldsymbol{\lambda})\boldsymbol{X} + \mathbf{B}(\boldsymbol{\lambda})\boldsymbol{U} \tag{4.5}$$

$$\begin{bmatrix} \Lambda_{11} & -\mathbf{S}_{12} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{X}}_i \\ \dot{\boldsymbol{X}}_d \end{bmatrix} = \begin{bmatrix} \mathbf{KF}_i & \mathbf{KF} \\ -\mathbf{S}_{12}^T\mathbf{F}_i - \boldsymbol{\Lambda}_{22}\mathbf{F}^T & -\mathbf{S}_{12}^T\mathbf{F} - \boldsymbol{\Lambda}_{22}\mathbf{F}_d \end{bmatrix} \begin{bmatrix} \boldsymbol{X}_i \\ \boldsymbol{X}_d \end{bmatrix} + \begin{bmatrix} \mathbf{S}_{14} + \mathbf{S}_{13}\mathbf{H}\mathbf{S}_{34} \\ \mathbf{S}_{24} \end{bmatrix} \begin{bmatrix} \boldsymbol{U} \end{bmatrix} \tag{4.6}$$

Again, linear coefficients are assumed in order to derive the LTI form. These matrices could contain any number of functions and off-diagonal terms (including terms in $\mathbf{F}$) in a nonlinear model, and a similar derivation process can be followed to obtain nonlinear system equations which would take the general form:

$$\boldsymbol{\Lambda}\dot{\boldsymbol{X}} = f_1(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{U}, \boldsymbol{\lambda}, t) \tag{4.7}$$

$$\mathbf{0} = f_2(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{U}, \boldsymbol{\lambda}, t) \tag{4.8}$$

To obtain the implicit system equation (4.6) the following procedure is proposed:

Procedure 2: A Procedure for finding the implicit system equations of a hybrid bond graph

1. Construct the diagonal matrix Λ
   - Consider each 1-port element in dynamic causality in turn, and determine all paths of dynamic causality between these elements and the controlled junctions.
   - Use a truth table to construct the combination of states, and hence function of Boolean variables, that result in each causal change.
2. Construct the Hybrid Junction Structure Matrix (HJSM) in form of Eq. (4.13).
   - The HJSM relates system inputs to outputs. For elements in static causality, there will be one input and one output. For elements in dynamic causality, there are two inputs (effort and flow) and two outputs.
   - For static causality, the HJSM is constructed by using 1's and 0's to denote whether quantities are related or not.
   - Where a path between two elements crosses a TF or GY element, a variable or function other than one may appear in the Hybrid Junction Structure Matrix.
   - Where a path between two elements crosses a controlled junction, a λ (or function of λ) is used to show that the relationship only occurs when the junction is ON (or OFF).
   - Where an element is in dynamic causality (shown by a dotted causal stroke in addition to the solid one) each variable will only be an input to the system in certain modes of operation. Referring to the truth table constructed in step 1, assign a function of λ which denotes the modes in which the variable is an input.
   - Recall that the matrix should be skew-symmetric, and submatrices $\mathbf{S}_{22}$, $\mathbf{S}_{23}$ and $\mathbf{S}_{32}$ should be zeros.
3. Derive the LTI Implicit form.
   - Find matrices **L** and **F** from the (linear) relationships in the 1-port elements.
   - Take the submatrices of **S** and Λ from the Junction Structure Matrix equation, and insert them into the general implicit LTI form in Eq. (4.6).
   - Simplify this equation to give the state equations plus some additional equations relating to the pseudo-states.

4.5 Solving the Hybrid Mathematical Model

Solving hybrid models by computer is not a new concept: there has been a tremendous amount of work in this area. However, the mixed-Boolean state model generated from the causally dynamic hybrid bond graph is particularly simple to implement.

4.5.1 *Background*

There is a body of work on simulation of hybrid bond graphs using causally static models (e.g. the use of causality resistance in 20-Sim), transforming the models to other environments such as the acausal modelling language Modelica [44], its commercial GUI Dymola [50], and formalising the transformation of bond graph models into block diagrams for simulation in SIMULINK [3, 4, 13, 21]. Hybrid Process Algebra (HyPA) has been applied to bond graphs [15, 16]. The potential for cosimulation of bond graphs and block diagrams has been explored [54], as has deriving input/output port-Hamiltonian models which allow the bond graph model to be embedded in or cosimulated with other environments [55–57]. These moves make sense from a computational point of view (and much of the research here was conducted by computer scientists) but lose the graphical advantages and relation to the physical system which are important to the systems engineer. This work in other software packages typically using controlled junctions also yielded results in the field of FDI [18, 43] where the focus was on efficient simulation rather than exploiting the bond graph [23, 45, 49].

4.5.2 *Simulating the Mixed-Boolean State Model*

At the time of writing, there are no commercial environments for simulating causally dynamic hybrid bond graphs (although there are controlled junctions in the latest versions of 20-Sim). The user therefore has two options:

- insert an equational subsystem into a Bond Graph environment (e.g. 20-Sim)
- transfer the mixed-Boolean state model to a programming environment (e.g. Modelica, Matlab).

Matlab has been used successfully to simulate the mixed-Boolean State Model [30]. Conditional statements and loops can be used to assign values to the Boolean parameters at any given time step, and the system can then be solved using regular solvers. Where the model is implicit in some modes of operation, an implicit solver such as BDF or ode15i (in Matlab) must be used. An example Matlab script is provided in the appendix.

Immediately after a discontinuity, the state (or pseudo-state) variable is assigned the same value that the corresponding state (or pseudo-state) had immediately before the discontinuity. This is because the model is inherently a switching model, and the causally dynamic hybrid bond graph described here guides the user towards developing a switching model rather than an impulsive one. There are no unknown energy losses, and hence no need for state reinitialisation. This is discussed in more detail in Sect. 4.6.6.

There is a body of literature on the simulation of hybrid and nonsmooth dynamical systems which has developed in parallel with the hybrid bond graph. A number of issues merit consideration [1]:

- Switching must always occur at the end of a time step, in order to be captured. This usually motivates an event-driven method. However, this can be impractical where there is a large number of switching instants or it is not known where they occur.
- Chattering may occur where a sliding mode cannot be reached due to numerical approximation.
- A procedure for accurately finding the location of events may be required, along with some method for reinitialising states after the event.
- Where there are a number of events, there may be a finite accumulation point past which the event-driven method cannot progress.
- There may be an impulsive term on commutation giving a Dirac or Steltjes measure. For example, the differential measure of velocity which manifests on impact between bodies.

Acary and Brogliato [1] suggest the use of discrete-time Moreau's second-order sweeping process for solving problems such as the bouncing ball discussed here. There are no detection times, and hence no accumulation point. Nonsmooth measures are treated rigorously with no 'jump' in acceleration, and hence no impulse losses or need to reinitialise states.

4.5.3 Developing an Impulse Model

Where a mode of operation is very short (e.g. the 'in contact with the ground' mode of a bouncing ball) it can be computationally inefficient to simulate it. This is why these kinds of problems are often abstracted to impulsive models. The problem with impulsive models is that the impulse loss on commutation is unknown, and the values of state variables after the discontinuity are consequently unknown. The impulse must be explicitly modelled (as in Impulse Bond Graphs [65]) or the states must be reinitialised using some algorithm [36, 45].

The [switching] hybrid bond graph can be used to generate an impulsive model, with the benefit of allowing the impulse loss to be calculated rather than simply estimated. For example, in the bouncing ball problem, the model can be simplified to give Newton's Collision Law (an impulsive model) with a coefficient of restitution calculated from the model [31].

4.6 Structural Analysis in Causally Dynamic HBGs

Here the term ‘structural analysis’ refers to the qualitative analyses that can be carried out on the bond graph prior to simulation. They can guide the user in revisiting their modelling assumptions or even the design of the system. They will also provide some information about computation, such as whether the model is explicit or implicit. The structural analyses mirror those carried out on state equations, because the bond graph and state model are closely linked.

4.6.1 State and Implicit Models

Control properties are usually found from the state space representation of the model with causality assigned (i.e. the input/output model). An overview is given by the most standard control textbooks such as Sontag [51]. Although there have been significant developments in control theory since that time, these basic parameters are still widely taught and used today, often in the context of the explicit linear time-invariant (LTI) state space equation. Authors typically strive to obtain the explicit regular state space system for ease of both computation and analysis. Since state equations are easily derived from the bond graph, it follows that control properties normally found from the state equations are reflected in the bond graph’s structure and causal assignment. A body of work by Sueur, Dauphin-Tanguy and others brings the notions of structural analysis and exploiting causality closer together. These properties (controllability, stability, (in)finite structure, etc.) aid with instrumentation for experimentation and fault diagnosis and identification (FDI).

However, implicit systems in various forms appear to arise naturally when looking at interconnected systems, and Lewis [25] argues that they are more suitable for signal processing and modelling tasks than explicit state space models. Specific implicit forms which have been investigated in detail are singular systems, semistate systems and descriptor systems. Yip and Sincovec [64] establish properties of the descriptor system, and Verghese et al. [60] develop a generalised theory for singular systems: both essentially present control properties for implicit systems which mirror those established for explicit ones. Lewis [24–26] gives a useful review of implicit systems and techniques for analysing them, and Dai [17] looks specifically at matrix-rank criteria for singular systems. Their results will be extended to the equations generated by the causally dynamic hybrid bond graph here.

The main considerations for a singular system are the presence and treatment of impulsive modes, and of causality. Causality in this sense refers to whether a value can be calculated from past values (causal) or depends on both past and forward values (noncausal): implicit systems are by nature noncausal. This allows them to be manipulated into a state space form and perhaps more significantly they can be considered as behavioural models. Willems [62, 63] demonstrates that control can be studied from a behavioural point of view without introducing inputs and outputs. Lewis and Ozcaldiran [27] therefore investigate the geometric properties of implicit models referencing Willems, arguing that they give increased engineering insight.

4.6.2 Observations on the Dynamic Causality Assignment

The dynamic causality notation was designed to give some insight into the model and be more usable than existing notation [5]. This section presents a series of observations on dynamic causality manifesting in the hybrid bond graph.

An immediate observation is that paths of dynamic causality between controlled junctions and elements can be identified, and these clearly show the elements affected by commutation of a controlled junction. This means that compiling a truth table for the model and constructing the Λ-matrix for any subsequent equation derivation is greatly simplified. Rather than constructing a truth table for the whole model with 2^i possible modes of operation (where i is the number of controlled junctions), a series of smaller truth tables can be constructed for each segment of the model in dynamic causality.

Additional observations on the paths of dynamic causality can be made in line with those already made for causally static bond graphs, such as Margolis and Rosenberg's work on exploiting causality [34, 47] and Rosenberg and Andry's work on solving causal loops [48].

The number of storage elements in dynamic causality is a measure of the variation in model size. Recall that elements in integral causality provide the state variables in deriving the state equations. Therefore, when all of storage elements are in integral causality, the maximum number of states is active. In a well-constructed model, there are no elements in static derivative causality and this mode (the reference mode) gives a fully explicit state space model. Likewise, the mode of operation where most storage elements are in derivative causality gives the minimum number of state variables (and the maximum number of pseudo-states which yield additional algebraic equations). Dynamic causality on resistance elements does not affect the model size.

If dynamic causality is not controlled, it can be exploited in much the same way as static causality. Causal paths between elements (in the reference or other modes) can be traced, and signify various types of algebraic or kinematic constraint. In addition to identifying these paths in the case of dynamic causality, it is possible to further classify them.

The classic variable topology problem—ideal 'hard' contact resulting in coalescence—is visible via a controlled junction which is OFF in the reference mode and results in a kinematic constraint between rigid bodies (i.e. a causal path between two I-elements, one of which will be in derivative causality) when it is ON. This usually manifests as a path of dynamic causality between a controlled junction and I-element, shown in Fig. 4.12. It is also possible for compliance elements to become kinematically constrained. A user may choose to break the kinematic constraint by revising modelling assumptions: the classic approach is

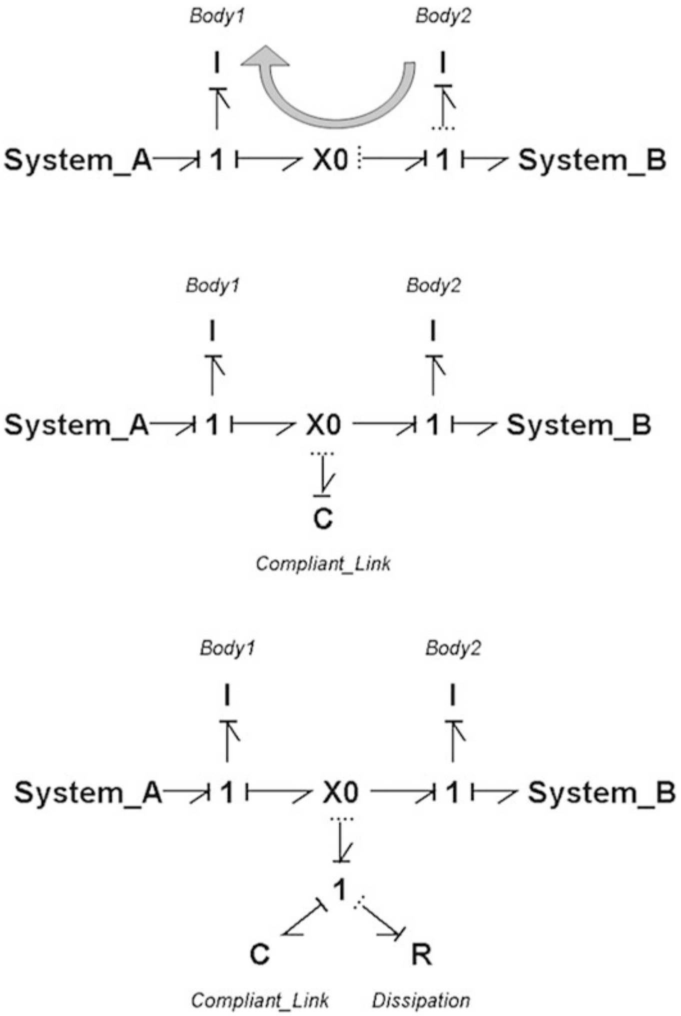


Fig. 4.12 Example of a type 1 discontinuity: 'Hard' contact resulting in a kinematic constraint between bodies 1 and 2 (*top*), compliant contact (*middle*) and compliant contact with resistance (*lower*)

to redefine a hard contact problem as stiff contact, by adding a stiff compliance between the inertia elements. This may not always be appropriate, especially if a 'proper model' or model devoid of high-frequency stiff dynamics is required. If the mode(s) of operation for which the constraint exists were to be considered in isolation, the constrained storage elements could be lumped together: this may be laborious by hand, but a computer programme could feature an algorithm for lumping constrained elements or using relaxed causalities [22] in the appropriate modes of operation.

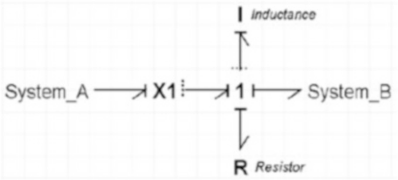


Fig. 4.13 Example of a type 2 discontinuity

Where there is compliant contact, the inertia elements will remain in integral causality and there is no kinematic constraint. There will be a path of dynamic causality between the junction and a compliance element instead. There is typically some dissipation associated with this type of contact, and a resistance element may be added which will act as causality resistance: in this case the path of dynamic causality will be between the controlled junction and resistance element.

Definition 4.6. Type 1 Structural Discontinuity: The controlled junction is OFF in the reference configuration. When it is ON, two subsystems are joined and a kinematic constraint may result.

The alternative situation is a controlled junction which is ON in the reference configuration, and divides the model into subsystems when it is OFF, as shown in Fig. 4.13. Disconnecting a power source—for example, in an electrical or hydraulic circuit—can result in storage elements discharging, and this is reflected by them switching to derivative causality. As with the type 1 structural discontinuities, parasitic elements (additional compliance, causality resistance) can be used to control this dynamic causality but must be used with caution.

Definition 4.7. Type 2 Structural Discontinuity: The controlled junction is ON in the reference configuration. When it is OFF, the system is divided into subsystems, and storage elements may discharge to compensate for a lack of power source in a subsystem.

As dynamic causality occurs, various other paths and loops may be created or broken with commutation. These are not a feature of the type of discontinuity.

Some observations may be made regarding the control properties of the system. Assume a system is [structurally] controllable and observable in the reference mode. If the system contains type 1 discontinuities, the system's finite dynamics remain controllable and observable after commutation. If it contains type 2 discontinuities, the model may be subdivided into uncontrollable and unobservable subsystems with commutation.

In assigning the causality around a controlled junction, the user may make an arbitrary decision regarding which element to place in dynamic causality. Consider the 'hard contact' in Fig. 4.12: dynamic causality was assigned to Body 2 but could just as easily have been assigned to Body 1. The basic effects on the system are the same:

- When the controlled junction is OFF the I-elements
  - are both in integral causality and
  - both yield a state variable and equation.

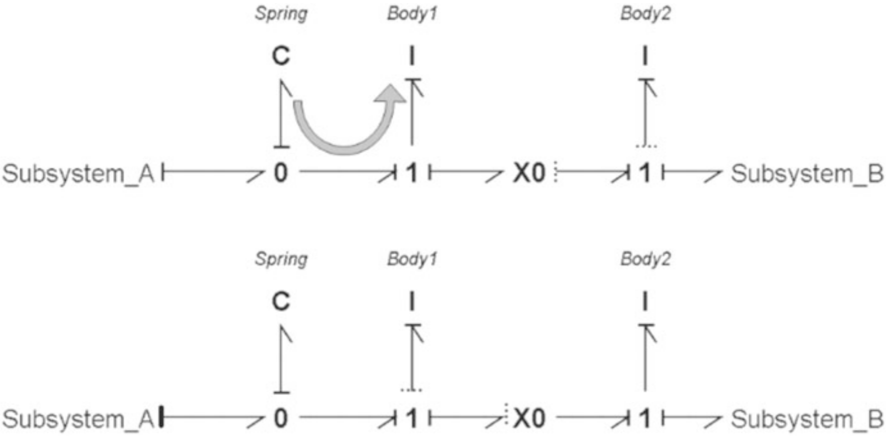


Fig. 4.14 Hard contact with a causal path (indicated by *arrow*): Body 2 in dynamic causality (*top*) and Body 1 in dynamic causality (*lower*)

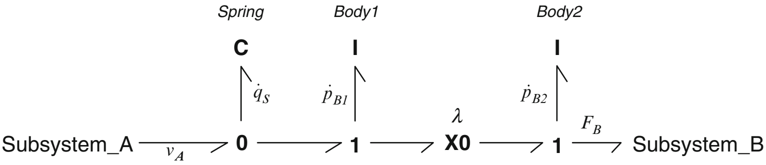


Fig. 4.15 Notation used in equation derivation for the hard contact example

- When the controlled junction is ON,
  - one of the I-elements switches to derivative causality,
  - the order of the model is reduced by one and
  - the elements yield one state variable and equation, and an additional pseudo-state providing an algebraic constraint between the two bodies.

The only time that the choice of element to place in derivative causality may be significant is when there is an existing causal path between an element and some other structure. This will still be captured via the new causal paths, but may be less immediately obvious to the user and computationally inefficient. An example is shown in Fig. 4.14.

The causality assignment in Fig. 4.14 (top) results in a static causal path between Body 1 and the spring. The equally legitimate causality assignment in Fig. 4.14 (lower) does not have this static causal path. Instead, there is a causal path between the spring and Body 1 only when the controlled junction is OFF. When the controlled junction is ON, this path no longer exists but another causal path appears between the spring and Body 2. Inspecting the structure and causality of the systems (using the notation in Fig. 4.15) yields the equations in Table 4.3. It can be seen that the first set of equations is more elegant and concise. The second set still relates

Table 4.3 Junction Structure & Implicit State Equations for the Hybrid Bond Graph in Figure 4.14

Body 2 in Dynamic Causality	Body 1 in Dynamic Causality
$\mathbf{\Lambda}\begin{bmatrix}\dot{\hat{q}}_S\\ \dot{\hat{p}}_{B1}\\ \dot{\tilde{p}}_{B2i}\\ \tilde{f}_{B2d}\end{bmatrix}=\begin{bmatrix}0 & -1 & 0 & 0 & 1 & 0\\ 1 & 0 & 0 & -\lambda & 0 & \lambda\\ 0 & 0 & 0 & 0 & 0 & -\overline{\lambda}\\ 0 & \lambda & 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\hat{e}_S\\ \hat{f}_{B1}\\ \tilde{f}_{B2i}\\ \dot{\tilde{p}}_{B2d}\\ v_A\\ F_B\end{bmatrix}$ $\mathbf{\Lambda}=diag\begin{bmatrix}1 & 1 & \overline{\lambda} & \lambda\end{bmatrix}$	$\mathbf{\Lambda}\begin{bmatrix}\dot{\hat{q}}_S\\ \dot{\tilde{p}}_{B1i}\\ \dot{\hat{p}}_{B2}\\ \dot{\tilde{p}}_{B1d}\end{bmatrix}=\begin{bmatrix}0 & -\overline{\lambda} & -\lambda & 0 & 1 & 0\\ \overline{\lambda} & 0 & 0 & 0 & 0 & \lambda\\ \lambda & 0 & 0 & \lambda & 0 & -1\\ 0 & 0 & \lambda & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\hat{e}_S\\ \tilde{f}_{B1i}\\ \hat{f}_{B2}\\ \dot{\tilde{p}}_{B1d}\\ v_A\\ F_B\end{bmatrix}$ $\mathbf{\Lambda}=diag\begin{bmatrix}1 & \overline{\lambda} & 1 & \lambda\end{bmatrix}$
$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & \lambda\\ 0 & 0 & \overline{\lambda} & 0\\ 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\dot{\hat{q}}_S\\ \dot{\hat{p}}_{B1}\\ \dot{\tilde{p}}_{B2i}\\ \dot{\tilde{p}}_{B2d}\end{bmatrix}=\begin{bmatrix}0 & -1/L_{B1} & 0 & 0\\ -1/C_S & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & \lambda/L_{B1} & 0 & -\lambda/L_{B2}\end{bmatrix}\begin{bmatrix}\hat{q}_S\\ \hat{p}_{B1}\\ \tilde{p}_{B2i}\\ \tilde{p}_{B2d}\end{bmatrix}+\begin{bmatrix}1 & 0\\ 0 & \lambda\\ 0 & -\overline{\lambda}\\ 0 & 0\end{bmatrix}\begin{bmatrix}v_A\\ F_B\end{bmatrix}$	$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & \overline{\lambda} & 0 & 0\\ 0 & 0 & 1 & -\lambda\\ 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\dot{\hat{q}}_S\\ \dot{\tilde{p}}_{B1i}\\ \dot{\hat{p}}_{B2}\\ \dot{\tilde{p}}_{B1d}\end{bmatrix}=\begin{bmatrix}0 & \overline{\lambda}/L_{B1} & -\lambda/L_{B2} & 0\\ -\overline{\lambda}/C_S & 0 & 0 & 0\\ \lambda/C_S & 0 & 0 & 0\\ 0 & 0 & -\lambda/L_{B2} & -\lambda/L_{B1}\end{bmatrix}\begin{bmatrix}\hat{q}_S\\ \tilde{p}_{B1i}\\ \hat{p}_{B2}\\ \tilde{p}_{B1d}\end{bmatrix}+\begin{bmatrix}1 & 0\\ 0 & 0\\ 0 & -\lambda\\ 0 & 0\end{bmatrix}\begin{bmatrix}v_A\\ F_B\end{bmatrix}$

the spring to Body 1, but when the controlled junction is ON this is done via the algebraic constraint between the bodies.

Where a causal conflict occurs in the dynamic causal assignment, this indicates a conflict in a specific mode of operation, and the mode is a ‘forbidden mode’. Forbidden modes may be a consequence of the modelling assumptions, or reflect a real case such as a short-circuit.

A mode in which there is a causal conflict is ‘forbidden’ in the sense that causality cannot be assigned, and hence the mathematical model cannot be constructed for that mode, and the model cannot be simulated. An interesting property of the hybrid bond graph is that causal conflicts reflect modes that would be undesirable or impossible in reality. This is because the method was developed to reflect the physics of the system.

4.6.3 Transfer Function Using Shannon-Mason Loop Rule

It is well-documented that causal paths in a bond graph are equivalent to signal loops, and a transfer function can therefore be found directly from the causal bond graph using Shannon-Mason loop rule [10]. In the hybrid bond graph proposed here, commutation clearly affects the causal paths in the model (where they cross a controlled junction), and commutation will therefore also clearly manifest in the transfer function. By looking at the causal paths present in the reference configuration, and then each path of dynamic causality, a transfer function for all possible modes of operation can be obtained. Where the paths cross a controlled junction, or are induced by a certain combination of operations, the relevant Boolean term can be inserted into the expression for gain in the transfer function.

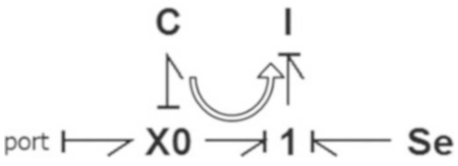


Fig. 4.16 A causal path crossing a controlled junction

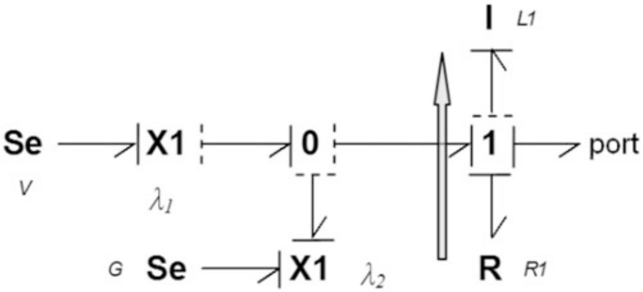


Fig. 4.17 A causal path in dynamic causality

In constructing the transfer function, causal paths between elements and sources are used to generate gain terms in the determinant. Where that path crosses a controlled junction, it only exists in the ON state. Multiplying the gain term by the Boolean factor ensures it is sent to zero when OFF. An example of this situation is shown in Fig. 4.16, where the causal path indicated by the arrow has a gain of $\frac{-\lambda}{s^2 \mathrm{IC}}$.

Where a causal path is dynamic or partially dynamic, the gain is factored by the Boolean function (term in Λ) which activates the path to give the signal loop under consideration.

Where a path between two elements is in dynamic causality, the complete path only exists in a certain mode of operation. For the example in Fig. 4.17, the causal path between the I- and R-elements (shown by the arrow) is partially in dynamic causality. This means that there is a path between the two elements in some modes of operation only (those where the I-element is in integral causality). Hence, the Boolean function associated with activating integral causality of the I-element must be used as a factor in the gain, giving $-\frac{(\lambda_1 \oplus \lambda_2)R}{sI}$. When the I-element is in derivative causality, there is no causal path and the gain term is sent to zero.

The general form of the transfer function is therefore identical to that of the standard bond graph:

$$h_{ij} = \frac{1}{\Delta} \sum_k G_k \Delta_k \tag{4.9}$$

where k denotes the kth path between input and output, and the graph determinant is

$$\Delta = 1 - \sum_i G_i + \sum_{i,j} G_i G_j - \sum_{i,j,k} G_i G_j G_k + \cdots \tag{4.10}$$

However, for the hybrid bond graph G is not only a function of any resistance, inertia or compliance coefficients relating to elements in the loop, but can also contain Boolean terms.

Where a controlled junction breaks/joins the causal paths in a model, the dynamic behaviour of the system is affected with commutation. The following points follow logically:

- If a controlled junction is in a path, then a Boolean term will be present in the graph determinant and hence the denominator of the transfer function. The roots of the denominator are the eigenvalues of the system, so it follows that the system will lose/gain one or more poles with commutation.
- If a controlled junction is in a path which does not 'touch' the input–output path (i.e. the paths do not share any nodes), then the 'reduced determinant' and hence the numerator of the transfer function will also contain a Boolean term. It follows that the system will lose/gain zeros with commutation.
- If a controlled junction is on the input–output path, the gain of the system will be affected. The numerator of the transfer function will again contain a Boolean and the system will lose/gain zeros with commutation.
- Where elements are in derivative causality, a path will only be present in some modes of operation and a Boolean function will denote this. Some paths will never touch because they are in exclusive modes of operation, and this must be taken into account when calculating the determinants.

4.6.4 The LTI Full Descriptor System

The states of the bond graph model can be used to generate an implicit system equation from the junction structure. For models where all storage elements are in integral causality, $\mathbf{E} = \mathbf{I}$ (i.e. $\mathbf{E}$ is non-singular) and the well-known explicit or regular state space equation is generated. For models where some storage elements are in derivative or dynamic causality, $\mathbf{E}$ is singular and the system is implicit [26]. Hybrid bond graphs with storage elements in dynamic causality will always generate implicit equations, because the derivative causality cases generate algebraic equations with no differential term (and hence a zero term in $\mathbf{E}$) and the constraints they represent give off-diagonal coupling terms in $\mathbf{E}$.

The LTI implicit equation forms one part of a descriptor system (4.11), the other part being given by the output equation (4.12).

$$\mathbf{E}(\boldsymbol{\Lambda})\dot{\boldsymbol{X}} = \mathbf{A}(\boldsymbol{\lambda})\boldsymbol{X} + \mathbf{B}(\boldsymbol{\lambda})\boldsymbol{U} \tag{4.11}$$

$$\boldsymbol{Y} = \mathbf{C}(\boldsymbol{\lambda})\boldsymbol{X} + \mathbf{D}(\boldsymbol{\lambda})\boldsymbol{U} \tag{4.12}$$

The $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$ and $\mathbf{E}$ matrices of these equations are used in defining control parameters such as controllability and observability, usually using matrix-rank criteria [17].

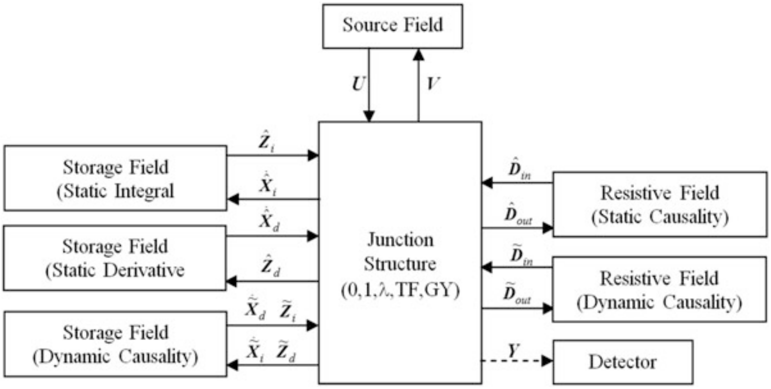

Source Field
U
V
Storage Field (Static Integral
Storage Field (Static Derivative
Storage Field (Dynamic Causality)
$\hat{Z}_i$
$\dot{\hat{X}}_i$
$\dot{\hat{X}}_d$
$\hat{Z}_d$
$\dot{\tilde{X}}_d$ $\tilde{Z}_i$
$\dot{\tilde{X}}_i$ $\tilde{Z}_d$
Junction Structure (0,1,λ,TF,GY)
$\hat{D}_{in}$
$\hat{D}_{out}$
$\tilde{D}_{in}$
$\tilde{D}_{out}$
Y
Resistive Field (Static Causality)
Resistive Field (Dynamic Causality)
Detector

Fig. 4.18 The general hybrid bond graph with signal detectors giving outputs

There is no standard output element in the bond graph framework. Some authors take an output as being the complementary variable of an input, which is logical in systems where source elements are also used as sinks. A fairly common notation is that of detector elements (De- or Df-elements) which are essentially null sources added to junctions. These act in precisely the same manner as a source/sink, and simply have a different notation for clarity. However, output is not a property of a system, and the use of detector elements with a power flow suggests that sensors are energy-processing.

Signal detectors similar to those in the commercial package 20-Sim are used here to indicate output. The use of detector elements (De- or Df-elements)—which are essentially null sources added to junctions—was avoided as they imply that output is a system property and that the sensors are energy-processing. Signal detectors are not energy-processing, and can output quantities other than effort and flow.

Outputs can be added to the junction structure, shown graphically in Fig. 4.18, and hence expressed in terms of the junction structure and system inputs, shown in Eq. (4.13):

$$\begin{bmatrix} \boldsymbol{A}_{11} & 0 & 0 & 0 \\ 0 & \boldsymbol{A}_{22} & 0 & 0 \\ 0 & 0 & \boldsymbol{A}_{33} & 0 \\ 0 & 0 & 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{X}}_i \\ \boldsymbol{Z}_d \\ \boldsymbol{D}_{\text{out}} \\ \boldsymbol{Y} \end{bmatrix} = \begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} & \mathbf{S}_{13} & \mathbf{S}_{14} \\ -\mathbf{S}_{12}^T & 0 & 0 & \mathbf{S}_{24} \\ -\mathbf{S}_{13}^T & 0 & \mathbf{S}_{33} & \mathbf{S}_{34} \\ \mathbf{S}_{41} & \mathbf{S}_{42} & \mathbf{S}_{43} & \mathbf{S}_{44} \end{bmatrix} \begin{bmatrix} \boldsymbol{Z}_i \\ \dot{\boldsymbol{X}}_d \\ \boldsymbol{D}_{\text{in}} \\ \boldsymbol{U} \end{bmatrix} \tag{4.13}$$

This can be used to derive an output equation:

$$\boldsymbol{Y} = \mathbf{JF}_i\boldsymbol{X}_i + \mathbf{JF}\boldsymbol{X}_d + \mathbf{S}_{42}\dot{\boldsymbol{X}}_d + (\mathbf{S}_{43}\mathbf{HS}_{34} + \mathbf{S}_{44})\boldsymbol{U} \tag{4.14}$$

where $\mathbf{J} = \mathbf{S}_{41} - \mathbf{S}_{43}\mathbf{HS}_{13}^T$. This is a rather unorthodox form of the output equation, since it is a function of $\dot{\boldsymbol{X}}_d$. When all storage elements are in static causality, this is of little consequence, since $\dot{\boldsymbol{X}}_d = 0$. The expected descriptor system can then be found, and standard matrix-rank criteria can be used to analyse the hybrid bond graph.

The output equation takes the usual LTI Descriptor System form if there is static causality on all of the storage elements. However, in a hybrid bond graph, storage elements can take dynamic causality and the $\dot{\boldsymbol{X}}_d$ term must be considered.

The hybrid bond graph can therefore be used to generate the standard LTI Descriptor System form. This allows comparison with control properties using established techniques such as matrix-rank criteria.

$$\begin{bmatrix} \Lambda_{11} & -\mathbf{S}_{12} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{X}}_i \\ \dot{\boldsymbol{X}}_d \end{bmatrix} = \begin{bmatrix} \mathbf{K}\mathbf{F}_i & \mathbf{K}\mathbf{F} \\ -\mathbf{S}_{12}^T\mathbf{F}_i - \boldsymbol{\Lambda}_{22}\mathbf{F}^T & -\mathbf{S}_{12}^T\mathbf{F} - \boldsymbol{\Lambda}_{22}\mathbf{F}_d \end{bmatrix} \begin{bmatrix} \boldsymbol{X}_i \\ \boldsymbol{X}_d \end{bmatrix} + \begin{bmatrix} \mathbf{S}_{14} + \mathbf{S}_{13}\mathbf{H}\mathbf{S}_{34} \\ \mathbf{S}_{24} \end{bmatrix} \begin{bmatrix} \boldsymbol{U} \end{bmatrix} \tag{4.15}$$

$$\begin{bmatrix} \boldsymbol{Y} \end{bmatrix} = \begin{bmatrix} (\mathbf{S}_{41} - \mathbf{S}_{43}\mathbf{H}\mathbf{S}_{13}^T)\mathbf{F}_i & (\mathbf{S}_{41} - \mathbf{S}_{43}\mathbf{H}\mathbf{S}_{13}^T)\mathbf{F} \end{bmatrix} \begin{bmatrix} \boldsymbol{X}_i \\ \boldsymbol{X}_d \end{bmatrix} + \begin{bmatrix} \mathbf{S}_{43}\mathbf{H}\mathbf{S}_{34} + \mathbf{S}_{44} \end{bmatrix} \begin{bmatrix} \boldsymbol{U} \end{bmatrix} \tag{4.16}$$

4.6.5 Impulse Losses

A key concept in variable topology systems is that of impulsive losses. The term ‘impulse’ is used in conjunction with two subtly different issues which must be distinguished here. The first is the impulse modes associated with the causally dynamic model and the initial value of the pseudo-state: i.e. the mathematical treatment of ideal switching. The second is the abstraction of the modes of operation themselves: much of the existing work on the subject investigates collisions using restitution, where the contact phase itself (and consequent dissipation) is so short as to be abstracted to a discontinuity.

Considerable work has been dedicated to the question of impulse losses on commutation, including Mosterman’s work on implicit modelling [35] and Zimmer and Cellier’s proposal for an Impulse Bond Graph [65]. Mosterman’s work in particular gave rise to a body of work where state variables are reinitialised after each discontinuous event. These authors use the classical case study of Newton’s Collision Law with restitution. Collision is an example of a subset of hybrid model called the Impulse model [7]. Here, the continuous state changes impulsively on hitting prescribed regions of state space. The ‘jumps’ in state space are not energy conserving: and hence an impulse loss must be accounted for. In the case of collision, the continuous state changes from positive to negative velocity with any energy loss accounted for via the coefficient of restitution.

The use of a Boolean controlled junction in the hybrid bond graph dictates the way a discontinuity is abstracted: a switch must be ON or OFF, contact TRUE or FALSE, etc. In the case of a collision, this means that the short ‘in contact’ phase is modelled (whereas Newton’s collision law neglects this). Any energy dissipation is modelled on the bond graph during this phase: there is no restitution. The hybrid bond graph with structural switching is always a ‘Switching model’ as defined by Branicky et al. [7], i.e. the vector field changes discontinuously when the state hits a boundary, but these changes are not impulsive.

4.6.6 Impulse Modes

Impulse modes (also referred to as infinite frequency modes) are a feature of hybrid systems. They occur where a storage element switches from integral to derivative causality, giving a step change in the value of the state. When all elements are in integral causality, **E** is an identity matrix and the model is in the explicit state space form. When a storage element changes to derivative causality with commutation, an algebraic constraint is typically set up and a non-diagonal term manifests in **E**. This term, which changes instantaneously from zero to a finite factor of a state variable on commutation, is what gives the impulse mode [60]. Recall the implicit equation (4.15), and compare it to the standard implicit equation. There is a time-varying term $\dot{\boldsymbol{X}}_d$ in the algebraic equations yielded by the pseudo-states of the storage elements in derivative causality. This is multiplied by zero (by the lower portion of **E**) to give an algebraic constraint. However, where storage elements in derivative causality are coupled to the states (i.e. $\mathbf{S}_{12}$, and therefore $\mathbf{E}_{12}$ in the implicit equation, are nonzero), a $\dot{\boldsymbol{X}}_d$ term is present and the pseudo-state is differentiated across the commutation. The pseudo-state is nominally assumed to have a zero initial value, and take a nonzero value on commutation. This means that there is a step increase in the pseudo-state between the initial condition (incrementally before commutation) and the finite value it holds at time t (incrementally after commutation). The first row of (4.15) after commutation therefore gives

$$\boldsymbol{\Lambda}_{11}\dot{\boldsymbol{X}}_i - \mathbf{S}_{12}\dot{\boldsymbol{X}}_d = \mathbf{KF}_i\boldsymbol{X}_i + \mathbf{KF}\boldsymbol{X}_d + (\mathbf{S}_{14} + \mathbf{S}_{13}\mathbf{HS}_{34})\boldsymbol{U} \tag{4.17}$$

where:

$$\dot{\boldsymbol{X}}_d = \frac{(\boldsymbol{X}_d)_t - (\boldsymbol{X}_d)_0}{dt} \tag{4.18}$$

Differentiating this step change over zero time yields a mode of infinite frequency: this is the impulse mode.

$$dt \to 0 \quad \therefore \quad \dot{\boldsymbol{X}}_d \to \infty \tag{4.19}$$

Commutation does not always result in a step change and subsequent impulse mode. Recall that the states and pseudo-states are intimately connected: each energy-storage element in dynamic causality has both a state variable (active when the element is in integral causality) and a pseudo-state (active when it is in derivative causality). By considering the type of discontinuity, some observations can be made on the relationship between states and pseudo-states.

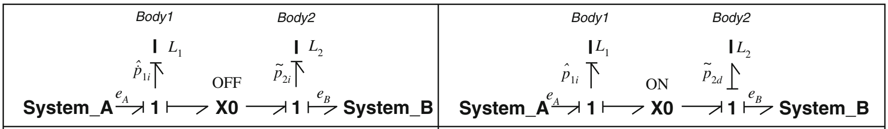
A type 1 structural discontinuity yields an impulse on initial commutation as two subsystems with different dynamic properties become joined and constrained (setting a storage element to derivative causality). When the system returns to its original state on subsequent commutation, there is no impulsive mode.

For a type 1 structural discontinuity, where bodies are disconnected (OFF) in the reference mode and the commutation connects them, the initial value of the pseudo-state may indeed be zero if that body was at rest. Alternatively, it may have another value if it is controlled by another source or subsystem. There is typically an energy loss as the body changes its behaviour suddenly (for example, a falling rigid body hitting the ground, or a truck clutch being engaged: both of which give an audible loss). In real life—which is continuous—this is a measurable dissipation occurring over a finite time (albeit a small one). The abstraction to a discontinuity with no resistance is responsible for the impulse loss. The equations for an example system are shown in Fig. 4.19.

Assume that a commutation occurs at time t during a simulation. Prior to commutation (at time $t-1$) the system is in the reference mode and both storage elements are in integral causality. After commutation, Body 2 is in derivative causality. The second row of the implicit equation gives an algebraic term for $\tilde{p}_{2d}$, but there is also a differential $\dot{\tilde{p}}_{2d}$ term in the first row. Looking at the differentiation across the commutation:

$$\dot{\tilde{p}}_{2d} \approx \frac{\tilde{p}_{2d_{(t)}} - \tilde{p}_{2i_{(t-1)}}}{\Delta t} \tag{4.20}$$

where $\tilde{p}_{2i_{(t-1)}}$ is known from the previous time-step's calculation, and $\tilde{p}_{2d_{(t)}} = \hat{p}_{1i_{(t)}}$ because the two inertia elements are now rigidly constrained. Note that all values are known, so there is no need to reinitialise the variables using canonical forms.



$$\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\dot{\hat{p}}_{1i}\\ \dot{\tilde{p}}_{2i}\end{bmatrix} = \begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\hat{p}_{1i}\\ \tilde{p}_{2i}\end{bmatrix} + \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}e_A\\ e_B\end{bmatrix}$$

$$\begin{bmatrix}1 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}\dot{\hat{p}}_{1i}\\ \dot{\tilde{p}}_{2d}\end{bmatrix} = \begin{bmatrix}0 & 0\\ L_1 & L_2\end{bmatrix}\begin{bmatrix}\hat{p}_{1i}\\ \tilde{p}_{2d}\end{bmatrix} + \begin{bmatrix}1 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}e_A\\ e_B\end{bmatrix}$$

Fig. 4.19 Equations for modes in isolation, type 1 discontinuity: reference mode (*left*) and after commutation of the X1-junction (*right*)

The time step is also nonzero and known, and governed by the integrator parameters used for computation. A large step change (compared to the time step) may cause computational difficulties at this point as the left-hand side of the state equation can become large. In this case it may be advantageous to set $\tilde{\dot{p}}_{2d_{(t)}} = 0, \tilde{p}_{2d_{(t)}} = \hat{p}_{1i_{(t)}}$ and neglect the impulse. This naturally violates conservation of energy but would ease computation considerably.

If the reverse situation is true, i.e. there is a type 2 structural discontinuity where the bodies are connected in the reference mode and the commutation disconnects them, the initial value of the pseudo-state is not zero.

A type 2 structural discontinuity does not yield an impulse on initial commutation. In this case, the state of the storage element which switches to derivative causality is identical to its value immediately before commutation. Hence there is no step change in state variable and no energy loss.

When the system returns to its original state on subsequent commutation, there is a step change in state variable. However, this does not manifest as an impulsive off-diagonal term in the $\mathbf{E}$ matrix: it is simply the newly activated $\dot{\mathbf{X}}_i$ term. This term is the output calculated from the state variables and inputs at the current time, and there has been no differentiation over a zero time step. Hence there is no impulse.

In this case, the initial value of the pseudo-state is equal to the [usually finite] value of the corresponding state variable immediately before commutation. Furthermore, after commutation the pseudo-state is not sent to zero. The behaviour of the element may be controlled by some other system, or may tend to zero over time (for example, a clutch disconnecting a load which free-wheels until it finally reaches rest). In this case there is no step change in variable and no impulse. However, when commutation occurs again and the disconnected body is reconnected to the system (going from zero to a finite value), a step change in state variable may then occur. Consider a system with one element in dynamic causality, which has just commutated back to the mode in which it is in integral causality. Row 1 of the implicit equation (which is now explicit) gives

$$\mathbf{I}\dot{\boldsymbol{X}}_i = \mathbf{KF}_i\boldsymbol{X}_i + (\mathbf{S}_{14} + \mathbf{S}_{13}\mathbf{HS}_{34})\boldsymbol{U} \qquad (4.21)$$

Although a step change may have occurred, there is no impulsive term in the equation.

Hence each term in $\mathbf{E}_{12}$ potentially represents an impulse mode, but in reality there is only an impulse loss where there are type 1 discontinuities. Any algorithm for computing impulsive modes must take the variety of possible cases into account.

The use of pseudo-states means that the state variable never needs to be reinitialised. In type 1 systems, the pseudo-state variable arises because there is a kinematic constraint between two elements, and the pseudo-state of one is equal to the state of the other. In type 2 systems, there is no step change on the initial commutation, but a subsequent commutation may result in a step change: this does not result in an impulsive term in the equations because the differential term is explicit.

A mathematical impulse has infinite magnitude and zero width, hence there is no actual energy loss. In reality, there can be dissipation in the form of an audible noise, spark or heat loss. This can be represented on the bond graph via a resistance element, or neglected in which case conservation of energy does not hold true. Mosterman [41] suggests that conservation of state is the important concept in system modelling, and the impulse energy dissipated during discontinuous events is 'free energy'.

Each mode change is a sliding mode change and the implicit equation can always be used because one side of it is always known. The use of impulse models like Newton's collision law (i.e. restitution) are a different case. In impulse models there is a jump in state space between modes of operation and an associated energy loss. This type of impulsive loss should not be confused with the impulse modes described above.

4.6.7 Control Properties

The term 'control properties' refers to those properties obtained from the model which can benefit the control engineer. These are typically stability, controllability/observability, solvability and related properties which can aid the engineer in assessing a design and defining instrumentation and stabilising controllers.

Analysis of the state and implicit equations using matrix-rank criteria—with or without transforming the model to various canonical forms—is well established and perhaps the most common form of system analysis. The use of matrix-rank criteria necessitates an input–output model, i.e. they must be applied after causality has been assigned. The validity of this approach has been called into question. A more promising approach, in keeping with the ideals of physical and behavioural modelling, might be a geometric one as proposed by Lewis [26] and Willems [63]. This is recommended as a topic for further study.

It is well established that a controllability matrix can be constructed for an LTI descriptor system, which has rank equal to model order when the system is controllable. This controllability matrix is a function of the **A** and **B** matrices, which are in turn comprised from the submatrices of **S**. Controllability can hence be seen on the bond graph by analysing the causal paths, or established from the junction structure matrix **S**. Previous work on hybrid bond graphs (using switched sources) has investigated R-controllability and Impulse-controllability , relating to the finite and impulse modes, respectively. The same distinction is made here.

Recall that the order of the model, and hence the number of finite modes of the system, is given by the number of storage elements in integral causality. It therefore follows that the maximum number of finite modes occurs when most storage elements are in integral causality, i.e. the reference mode. The maximum number of finite modes is given by the order of the model in the reference mode. This is the maximum number of storage elements in integral causality: dim[$\mathbf{X}_i$]. R-controllability of these finite modes is an intuitive concept, fundamentally the same as the structural controllability defined for a continuous LTI system. The model is R-controllable when independent relationships in the equations—corresponding to causal paths on the bond graph which exist both when preferred integral and preferred derivative causality are applied—are present between each storage element and a source element.

When there is a controlled junction in a causal path, that path is severed or ceased to exist when the junction is OFF. It is intuitive that when a controlled junction occurs between the storage element in integral causality and a source element:

- a Boolean term occurs in the underlying equations which sets the relationship to zero when the junction is OFF, and
- the dynamic causality around the junction clearly shows that the causal path is broken when the junction is OFF.

There are two possible outcomes when this happens. The first is that the storage element remains in integral causality, in which case it may be controlled by another source, or it may be uncontrolled, i.e. there is an uncontrolled finite mode. The other outcome is that the storage element changes its causal assignment and the finite mode ceases to exist: in this case the impulse controllability must be assessed.

There may also be instances where a nearby controlled junction(s) results in dynamic causality on a causal path between the storage element in integral causality and a source element. The controlled junction(s) does not physically sever the causal path, but still clearly affects controllability because the relationship between the source and storage elements is nonexistent in some modes of operation.

A hybrid bond graph is structurally R-controllable iff:

1. There is a causal path between each storage element in integral causality and a source element in all modes of operation, i.e.:
   (a) the causal path does not cross a controlled junction or,
   (b) there is another path between it and a storage element crossing a controlled junction which operates in a mutually exclusive manner with the first, or,
   (c) the causal path crosses a controlled junction which forces the storage element into derivative causality when it is OFF.
2. In the reference mode, the rank of the controllability matrix is equal to the model order: i.e. the number of storage elements in integral causality when preferred integral causality is applied is equal to the number of storage elements in derivative causality when preferred derivative causality is applied (allowing dualisation of source elements).

This is an extension of R-controllability for a static bond graph, acknowledging the observations made for structural switching and dynamic causality, i.e. that relationships in the model can be dependent on commutation.

Impulse controllability has been established by hybrid bond graph practitioners such as Rahmani and Dauphin-Tanguy [46] by inspecting the switching sources, which is clearly not applicable here as there are none. An equivalent criteria of establishing causal paths between controlled junctions and storage elements could be stated. However, the impulse modes in the underlying equations no longer relate to the switching laws present in the hybrid bond graph with switching sources. In this hybrid bond graph, the impulse modes relate solely to storage elements in dynamic causality. Impulse controllability, in the classical sense, is whether these impulse modes can be controlled by a non-impulsive input, verified by the algebraic tests which involve inspecting the ranks of the **E**, **A** and **B** matrices. In the discussion of impulse modes (Sect. 4.6.6) it was noted that an impulse does not always occur in the model: it depends on the type of discontinuity and commutation.

Recall that impulse [infinite] modes occur when a storage element is in dynamic causality, i.e. it switches between integral and derivative causality with commutation. In a well-constructed model there would not normally be any elements in static derivative causality.

The maximum number of impulse modes is given by the number of storage elements in dynamic causality: dim $[\tilde{\mathbf{X}}_d]$.

This is an extension of the property for switched bond graphs (using switched sources). The number of impulse modes in any single mode of operation is given by the number of storage elements in derivative causality. It therefore follows that the maximum possible number of impulse modes is given when all possible storage elements are in derivative causality, and this is in turn given by the storage elements in dynamic causality. Note that this is the maximum for the overall model: when some modes are mutually exclusive, there may not be a single mode of operation where all impulse modes occur.

Since the impulsive modes only exist when the respective storage element is in derivative causality, impulse controllability could be established by a causal path (and algebraic relation) between the element and a source (either directly or via another element which is controlled) in that mode of operation. Looking at the hybrid bond graph, since impulse modes relate to storage elements in dynamic causality, this manifests as a causal path (at least part of which will be dashed, i.e. dynamic) between a storage element in derivative [dynamic] causality and a source element. Hence, Rahmani et al.'s criterion for impulse controllability [46] is adapted and reused here.

The model is impulse controllable iff there exists a causal path between an input source and a controlled junction passing through a storage element in derivative causality.

An uncontrolled impulse mode would indicate an element which has been severed from the sources in the system and switched to derivative causality, such as an accumulator or capacitor discharging when the pump/battery is disconnected.

Observability is assessed in much the same way as controllability, as it is the dual property. The difference is that the observability matrix is a function of the **C** matrix relating the outputs to the model states (as opposed to the **B** matrix relating the inputs to the model states).

Impulsive terms can occur when storage elements take derivative causality with commutation. Impulse observability is assumed to be the dual property of impulse controllability. Consequently R-observability for finite modes also exists and is assumed to be the dual of R-controllability. The following properties are therefore presented without proof.

A hybrid bond graph is structurally R-observable iff:

1. There is a causal path between each storage element in integral causality and a detector element in all modes of operation, i.e.
   (a) the causal path does not cross a controlled junction or,
   (b) there is another path between it and a detector element crossing a controlled junction which operates in a mutually exclusive manner with the first, or,
   (c) the causal path crosses a controlled junction which forces the storage element into derivative causality when it is OFF.
2. In the reference mode, the rank of the observability matrix is equal to the model order: i.e. the number of storage elements in integral casuality when preferred integral causality is applied is equal to the number of storage elements in derivative causality when preferred derivative causality is applied (allowing dualisation of detector elements).

The model is impulse observable iff there exists a causal path between an detector and a controlled junction passing through a storage element in derivative causality.

Asymptotic stability is typically established by finding the solutions of the characteristic polynomial. The roots for all possible modes of operation can be obtained and plotted: roots with positive real parts indicate unstable behaviour. This is a numeric approach rather than a structural one, and outside the scope of structural analysis.

Where asymptotic stability does not exist, it indicates the presence of ‘zero modes’ (eigenvectors with vanishing eigenvalues). Recall that ‘structurally null modes’ (i.e. eigenvalues which are zero, or the poles at the origin) are given by the storage elements which are in integral causality when preferred derivative causality is assigned [19].

For the hybrid bond graph this philosophy can be extended to the number of I- and C-elements which have to stay in [static] integral causality plus the number of I- and C-elements in dynamic causality when a preferred derivative causality is assigned to the bond graph model.

The maximum possible number of structurally null modes is given by the number of storage elements which can take integral causality when preferred derivative causality is assigned to the bond graph.

This is the number of storage elements in [static] integral causality, plus the number of storage elements in dynamic causality in the BGD (Bond graph in preferred derivative causality).

$$D_0 = (\dim[\hat{\mathbf{X}}_i] + \dim[\tilde{\mathbf{X}}_i])_{\text{BGD}} \tag{4.22}$$

This property is a logical extension of the procedure for finding structurally null modes in a static bond graph. Recall that placing the bond graph in preferred derivative causality yields a mathematical model in an alternative form including the inverse of the system matrices. When storage elements remain in integral causality, it means that $\mathbf{A}$ is singular and there are no unique solutions to the system of equations. In particular, inspection of the characteristic equation reveals that there

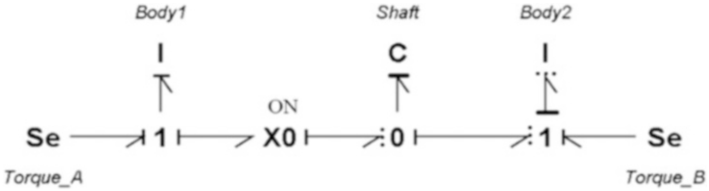


Fig. 4.20 Example system in preferred derivative causality, with ideal clutch

are k structurally null modes relating to the rows of **A** in which the causal constraints (in the BGD) exist, and these in turn relate to the storage elements that remain in integral causality in the BGD. The characteristic polynomial for the hybrid system:

$$P(s) = |s\mathbf{E}(\Lambda) - \mathbf{A}(\Lambda)| \tag{4.23}$$

$$P(s) = s^k(s^q + a_{q-1}s^{q-1} + \cdots + a_1 s^1 + a_0) \tag{4.24}$$

The k structurally null modes may not be obvious from the hybrid bond graph in integral causality. Consider a simple example of two rotating bodies connected by a shaft with a clutch fitted. The hybrid bond graph in preferred derivative causality is shown in Fig. 4.20.

In preferred derivative causality, the reference mode (the mode with most elements in derivative causality) occurs with the clutch engaged (ON), in which case body 1 is still in integral causality. When the clutch is OFF, body 2 is also sent to integral causality. Hence there are two possible structurally null modes, with one of them dependent on commutation of the clutch.

4.7 Conclusion

A causally dynamic hybrid bond graph has been proposed which provides a wealth of engineering insight (through structural analysis and exploiting causal assignment) and is suitable for simulation.

In proposing this method, the authors felt it important to retain the graphical advantages of bond graph modelling and the principles of idealised physical, acausal model construction. This has been achieved by classifying discontinuities so that they can be represented in the most intuitive way, developing a causally dynamic general hybrid bond graph (with a new dynamic causality notation and novel application of pseudo-state variables), and deriving a unique mixed-Boolean implicit system equation which describes the model in all possible modes of operation.

Classifying discontinuities as structural or parametric immediately shows whether the hybrid assumptions will affect the structural analysis of the model or not, and the use of controlled junctions and elements (respectively) is often intuitive.

The use of the controlled junction, with its inherent dynamic causal assignment, clearly shows how structural commutations (dis)connect submodels and form algebraic constraints. This is reflected in the junction structure matrix by the use of Boolean variables, which denote connections between elements dependent on the state of a controlled junction.

The dynamic causality notation is designed to facilitate analysis of the model's structure, as well as aiding in equation generation (which describes all modes of operation). Elements in dynamic causality reverse inputs and outputs depending on the mode of operation (i.e. they have an effort input and flow output in some modes, and a flow input and effort output in others). Hence elements in dynamic causality are assigned two inputs and two outputs during equation generation. During any single mode of operation, one input/output pair is active and the other is redundant.

If a storage element is in dynamic causality, it only yields a state output in some modes of operation (when it is in integral causality). In other modes it is in derivative causality, in which case it can be assigned a 'pseudo-state' variable which generates an algebraic equation.

A model describing all possible modes of operation is generated. The dynamic causality offers engineering insight, for example, when two rigid bodies contact and a kinematic constraint exists indicating that they behave as a single rigid body. This can be preferable to constraining causality through the addition of parasitic elements, which not only overcomplicates the model but can also even mask an issue or give incorrect behaviour.

The dynamic causality notation offers a unique new way of visualising variable causality and variable topology problems: the latter is no longer strictly variable topology because all system elements are described; they are just not necessarily connected. Dynamic causality can be exploited in the same way as static causality notation, showing how system properties vary with commutation and enabling graph-based structural analysis.

Variable topology problems—notably impact—have received much attention with regard to hybrid bond graphs, due to concerns over whether losses are treated correctly. Using the hybrid bond graph developed here, conservation of momentum holds true because the impact is abstracted to capture the short contact phase, and, in addition, the parameters for Newton's Collision Law can be derived and the model simplified to a classical collision problem.

Likewise, state reinitialisation has received some attention in the context of hybrid bond graphs. The way that discontinuities are abstracted using this method allows the initial value of each state to be logically deduced.

The unique model describing all possible modes of operation is suitable for both equation-based structural analysis (which is well-established for implicit descriptor systems) and simulation. Simulation often requires the use of solvers developed for implicit descriptor systems, or the use of some model simplification or symbolic manipulation to produce an explicit equation for each mode of operation. A simple example is provided in the Appendix.

The derivations and analyses presented in this thesis largely assume a linear time-invariant model, in order to facilitate comparison with existing control literature. However, the same derivations can be followed with nonlinear functions in place of the linear coefficients during equation generation.

Appendix

In this case study, a simple power converter connected to a DC Motor (an example used repeatedly in the literature) [30, 33] was populated with generic data and used to investigate a discontinuous event. A constant input was used, and the simulation was halted at the event time at which point the load was disconnected and then run again. The states are not reinitialised at the event time: they are simply taken to be the same as at the last time step (which is reasonable since they have reached steady state). N.b. it is possible to simplify this code by specifying event times as an integrator option.

```
% Run simulation until event
tspan1 = [0:1e-3:(5-(1e-3))];
y01 = [20 0 0]’;
yp01 = zeros(1,3)’;
[t1, y1] = ode15i(@hbg, tspan1, y01, yp01);

% Run simulation from event
tspan2 = [5:1e-3:10];
y02 = y1((length(t1)),:)’;
yp02 = ((y1((length(t1)),:) - y1((length(t1)-1),:))/1e-5)’;
[t2, y2] = ode15i(@hbg, tspan2, y02, yp02);

% Concatenate results vectors
t = [t1; t2];
y = [y1; y2];
```

where the functions are

```
function  out = hbg(t, y, yp)

if t == [0:0.2],[0.4:0.6];
% Change this statement depending on the problem under investigation
l1=0;
l2=1;
l3=1;
hsm = hss(l1, l2, l3);
else
l1=1;
l2=0;
l3=1;
hsm = hss(l1, l2, l3);
end
```

```
d = length(hsm(:,1));
w = length(u(:,1));
A = hsm(:,1:d);
B = hsm(:,d+1:d+w);
E = hsm(:,d+w+1:d+w+d);
u = [12;0];

out = A*y + B*u - E*yp;

end

---------------------------------
function hsm = hss(l1, l2, l3)

% Define coefficients
L1 = 10e-3;
L2 = 2.25e-7 ;
L3 = 1e-5 ;
R1 = 1.73 ;
R2 = 2e-5  ;
a = 0.00902;

% Populate matrices for this mode
A = [-1*xor(l1,l2)/(L1*R1) -a*xor(l1,l2)/L2
0  0 0
a*xor(l1,l2)/L1 -l3/(L2*R2) 0 0 0
0 0 ~l3/(L3*R2) 0 0
0 0 0 ~xor(l1,l2)/L1 0
0 l3/L2 0 0 -l3/L3];
B = [l1 l2
0 0
0 0
0 0
0 0];
E = [xor(l1,l2) 0 0 0 0
0 1 0 0 l3
0 0 ~l3 0 0
0 0 0 0 0
0 0 0 0 0];

% Remove null lines/columns
order = length(A(1,:));
z = zeros(1,order);
i = 1;
for n = (1:order)
if sum(A(n,:)) == 0 % Delete rows of zeros
else
Ared1(i,:) = A(n,:);
Bred1(i,:) = B(n,:);
Ered1(i,:) = E(n,:);
i = i+1;
end;
end;
```

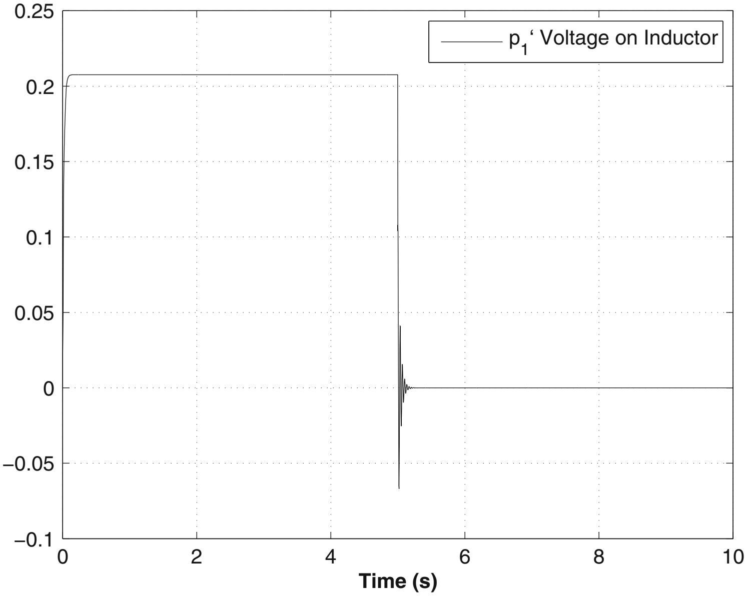


Fig. 4.21 Voltage on example power converter: constant input, load disconnected during operation

```
i = 1;
for n = (1:order) % Delete Columns of zeros
if sum(Ared1(:,n)) == 0
else
Ared2(:,i) = Ared1(:,n);
Bred2 = Bred1;
Ered2(:,i) = Ered1(:,n);
i = i+1;
end;
end;

hsm = [Ared2 Bred2 Ered2];
end
```

Some results are shown in Figs. 4.21 and 4.22 for illustration. The load is disconnected 5 s into the simulation. The model runs quickly, and shows a ‘spike’ in voltage immediately after the event. The torque on the motor is increased after the event, consistent with the absence of the load.

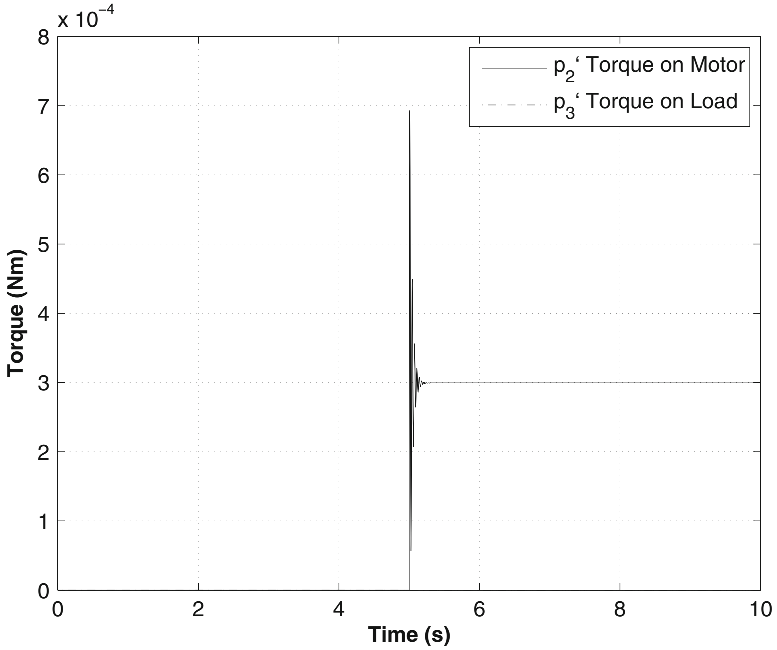


Fig. 4.22 Torque on example power converter: constant input, load disconnected during operation

References

1. Acary, V., & Brogliato, B. (2008). *Numerical methods for nonsmooth dynamical systems — Applications in mechanics and electronics*. Berlin: Springer.
2. Asher, G. M. (1993). The robust modelling of variable topology circuits using bond graphs. In *Proceedings of the 1993 Western Simulation Multiconference – International Conference on Bond Graph Modeling ICBGM'93, La Jolla, CA* (pp. 126–131).
3. Beers, C. (2005). Efficient Simulation Model for Hybrid Bond Graph [Poster]. Vanderbilt University. Available http://fountain.isis.vanderbilt.edu/publications/BondITRReviewPoster.ppt. Cited Aug. 2010.
4. Beers, C. D., Manders, E., Biswas, G., & Mosterman, P. J. (2006). Building efficient simulations from hybrid bond graph models. In *2nd IFAC Conference on Analysis and Design of Hybrid Systems* (Vol. 2, pp. 71–76). Alghero, Italy.
5. Bidard, C., Favret, F., Goldztejn, S., & Lariviere, E. (1993). Bond graph and variable causality. In *Proceedings of International Conference on Systems, Man and Cybernetics: Systems Engineering In the Service of Humans* (Vol. 1, pp. 270–275). Le Touquet, France.
6. Borutzky, W. (2010). *Bond graph methodology – Development and analysis of multi-disciplinary dynamic system models*. London: Springer.
7. Branicky, M. S., Borkar, V. S., & Mitter, S. K. (1998). A unified framework for hybrid control: Model and optimal control theory. *IEEE Transactions on Automatic Control, 43*, 31–45. http://doi.org/10.1109/9.654885.

8. Breedveld, P. C. (2000). An alternative model for static and dynamic friction in dynamic system simulation. In *1st IFAC Conference on Mechatronic Systems, Darmstadt, Germany* (pp. 717–722).
9. Breedveld, P. C. (2002). Modelling and simulation of bouncing objects: Newton's Cradle revisited. In *Proceedings of the Mechatronics*.
10. Brown, F. T. (1972). Direct application of the loop rule to bond graphs. *Transactions ASME Journal of Dynamic Systems, Measurement and Control, 94*(3), 253–261.
11. Buisson, J. (1993). Analysis and characterization of hybrid systems with bond-graphs. In *Proceedings of the International Conference on Systems, Man and Cybernetics: Systems Engineering in the Service of Humans, Le Touquet, France* (pp. 264–269).
12. Buisson, J., Cormerais, H., & Richard, P. Y. (2002). Analysis of the bond graph model of hybrid physical systems with ideal switches. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 216*, 47–63.
13. Calvo, J. A., Boada, M. J., Roman, J. L. S., & Olmeda, E. (2010). BONDSYM: Simulink based educational software for analysis of dynamic system. *Computer Applications in Engineering Education, 18*, 238–251. http://doi.org/10.1002/cae.20246.
14. Cellier, F. E., Otter, M., & Elmqvist, H. (1994). Bond graph modeling of variable structure systems. *Simulation Series, 27*, 49.
15. Cuijpers, P. J. L., & Reniers, M. A. (2005). Hybrid process algebra. *Journal of Logic and Algebraic Programming, 62*(2), 191–245.
16. Cuijpers, P. J. L., Broenink, J. F., & Mosterman, P. J. (2008). Constitutive hybrid processes: A process-algebraic semantics for hybrid bond graphs. *Simulation: Transactions of the Society for Modeling and Simulation International, 84*, 339–358. http://doi.org/10.1177/0037549708097215.
17. Dai, L. (1989). Impulsive modes and causality in singular systems. *International Journal of Control, 50*(4), 1267–1281.
18. Daigle, M. J., et al. (2009). A qualitative event-based approach to continuous systems diagnosis. *IEEE Transactions on Control Systems Technology, 17*(4), 780–793.
19. Dauphin-Tanguy, G., & Sueur, C. (2002). Bond graph for modelling, analysis, control design, fault diagnosis [Presentation Slides]. Available www.fceia.unr.edu.ar/~kofman/seminario/Argentine-nov02.ppt. Cited Dec. 2010.
20. Gawthrop, P. J. (1997). Hybrid Bond Graphs Using Switched I and C Components (CSC Report 97005). Centre for Systems and Control, University of Glasgow.
21. Geitner, G. H. (2006). Power flow diagrams using a bond graph library under SIMULINK. In *32nd Annual Conference on IEEE Industrial Electronics IECON 2006, Paris, France* (pp. 1359–1365).
22. Karnopp, D. C., Margolis, D. L., & Rosenberg, R. C. (2006). *System dynamics modeling and simulation of mechatronic systems*. New York, NY: Wiley.
23. Lattmann, Z. (2010). *A Multi-domain Functional Dependency Modeling Tool Based on Extended Hybrid Bond Graphs*. Dept. Electrical Engineering, Vanderbilt University.
24. Lewis, F. L. (1986). Survey of linear singular systems. *Circuits, Systems and Signal Processing, 5*(1), 3–36.
25. Lewis, F. L. (1992). Review of 2-D implicit systems. *Automatica, 28*(2), 345–354.
26. Lewis, F. L. (1992). A tutorial on the geometric analysis of linear time-invariant implicit systems. *Automatica, 28*(1) 119–137.
27. Lewis, F. L., & Ozcaldiran, K. (1989). Geometric structure and feedback in singular systems. *IEEE Transactions on Automatic Control, 34*(4), 450–455.
28. Low, C. B., Wang, D., Arogeti, S., & Zhang, J. B. (2010). Causality assignment and model approximation for hybrid bond graph: Fault diagnosis perspectives. *IEEE Transactions on Automation Science and Engineering, 7*, 570–580.
29. Margetts, R. (2013). Modelling and analysis of hybrid dynamic systems using a bond graph approach. University of Bath Dept. Mechanical Engineering. Available via OPUS. http://opus.bath.ac.uk/48041/. Cited 3 June 2016.

30. Margetts, R., Boudon, B., & Ngwompo, R. F. (2017). Simulation of causally dynamic hybrid bond graphs, with application to a power converter. In *The 36th IASTED International Conference on Modelling, Identification and Control (MIC 2017)*. Innsbruck: IASTED (accepted).
31. Margetts, R., & Ngwompo, R. F. (2014) Hybrid bond graphs for contact, using controlled junctions and dynamic causality. In *Proceedings of the Summer Simulation Multiconference, Monterey, CA*.
32. Margetts, R., & Ngwompo, R. F. (2015). Mode switching in causally dynamic hybrid bond graphs. *Mechatronics*. doi:10.1016/j.mechatronics.2015.04.013.
33. Margetts, R., Ngwompo, R. F., & da Cruz, M. F. (2013). Modelling a drop test of a landing gear using a hybrid bond graph. In E. P. Klement, W. Borutzky, T. Fahringer, M. H. Hamza, & V. Uskov (Eds.), *Proceedings of the IASTED International Conference on Modelling, Identification and Control* (pp. 312–317). Innsbruck: ACTA Press. http://doi.org/10.2316/P.2013.794-072.
34. Margolis, D. L. (1987). Exploiting causality for structured models using bond graphs. In *1987 American Control Conference, Minneapolis, MN* (pp. 1457–1461).
35. Mosterman, P. J. (2000). Implicit modeling and simulation of discontinuities in physical system models. In *4th International Conference on Automation of Mixed Processes: Hybrid Dynamic Systems ADPM2000, Dortmund, Germany* (pp. 35–40).
36. Mosterman, P. J. (2002). HyBrSim — A modelling and simulation environment for hybrid bond graphs. *Proceedings of the IMechE Part I: Journal of Systems and Control Engineering, 216*(11), 35–46.
37. Mosterman, P. J., & Biswas, G. (1995). Modeling discontinuous behavior with hybrid bond graphs. In *9th International Workshop on Qualitative Reasoning about Physical Systems, Amsterdam, Netherlands* (pp. 139–147).
38. Mosterman, P. J., & Biswas, G. (1996). A theory of discontinuities in physical system models. *Journal of The Franklin Institute, 335B*(3), 401–439.
39. Mosterman, P. J., & Biswas, G. (1997). Formal specifications for hybrid dynamical systems. In *Proceedings of the 15th International Joint Conf. Artificial Intelligence IJCAI-97, Nagoya, Japan* (pp. 568–573).
40. Mosterman, P. J., & Biswas, G. (1998). Hybrid automata for modeling discrete transitions in complex dynamic systems. In *7th IFAC Symposium on Artificial Intelligence in Real-Time Control, Grand Canyon National Park, AR* (pp. 43–48).
41. Mosterman, P. J., & Biswas, G. (2000). A comprehensive methodology for building hybrid models of physical systems. *Artificial Intelligence, 121*(1–2), 171–209.
42. Mosterman, P. J., Zhao, F., Biswas, G., & Amer Assoc Artificial, I. (1998). An ontology for transitions in physical dynamic systems. In *15th National Conference on Artificial Intelligence (AAAI 98)* (pp. 219–224). Madison, WI: American Association Artificial Intelligence.
43. Narasimhan, S. (2002). *Model-Based Diagnosis of Hybrid Systems*. Dept. Computer Science, Vanderbilt University.
44. Otter, M., Remehle, M., Engell, S., & Mosterman, P. (2000). Hybrid models of physical systems and discrete controllers. *AT - Automatisierungstechnik, 48*, 35–40.
45. Podgursky, B., Biswas, G., & Koutsoukos, X. D. (2010). Efficient tracking of behavior in complex hybrid systems via hybrid bond graphs. In *Annual Conference of the Prognostics and Health Management Society*. Portland.
46. Rahmani, A., & Dauphin-Tanguy, G. (2006). Structural analysis of switching systems modelled by bond graph. *Mathematical and Computer Modelling of Dynamical Systems, 12*(2–3), 235–247.
47. Rosenberg, R. C. (1987). Exploiting bond graph causality in physical system models. *Transactions on ASME Journal of Dynamic Systems, Measurement and Control, 109*(4), 378–383.
48. Rosenberg, R. C., & Andry, A. N., Jr. (1979). Solvability of bond graph junction structures with loops. *IEEE Transactions on Circuits and Systems, 26*(2), 130–137.
49. Roychoudhury, I., Daigle, M. J., Biswas, G., & Koutsoukos, X. D. (2011). Efficient simulation of hybrid systems: A hybrid bond graph approach. *Simulation, 87*, 467–498.

50. Shiva, A. (2004). *Modeling Switching Networks Using Bond Graph Technique*. Dept. Aerospace and Mechanical Engineering, University of Arizona.
51. Sontag, E. D. (1998). *Mathematical control theory — Deterministic finite dimensional systems. Textbooks in Applied Mathematics* (Vol. 6). New York: Springer.
52. Strömberg, J. E. (1994). *A Mode Switching Philosophy* (Dissertation No. 353). Linköping University.
53. Sueur, C., & Dauphin-Tanguy, G. (1989). Structural controllability/observability of linear systems represented by bond graphs. *Journal of The Franklin Institute, 326*(6), 869–883.
54. Tudoret, S., Nadjm-Tehrani, S., Benveniste, A., & Stromberg, J. E. (2000). Co-simulation of hybrid systems: Signal-simulink. In M. Joseph (Ed.), *Proceedings of the 6th International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems FTRTFT 2000* (Vol. 1926, pp. 134–151). Pune, India.
55. Van Der Schaft, A. (2005). Theory of Port-Hamiltonian Systems Chapter 1: Port-Hamiltonian formulation of network models; the lumped-parameter case. Dutch Institute of Systems and Control (DISC). Available http://www.math.rug.nl/~arjan/teaching.html. Cited May 2010.
56. Van Der Schaft, A. (2006). Port-Hamiltonian systems — An introductory survey. In *Proceedings of the International Congress of Mathematicians, Madrid, Spain* (pp. 1339–1365).
57. Van Der Schaft, A. (2009). Port-Hamiltonian systems: From geometric network modeling to control [Lecture notes for the EECI Graduate Course, LSS-Supelec, Gif-sur-Yvette]. Available http://www.math.rug.nl/~arjan/teaching.html. Cited May 2010.
58. Van Der Schaft, A. J., & Schumacher, J. M. (1999). *An introduction to hybrid dynamical systems. Lecture notes in control and information sciences* (Vol. 251). London: Springer.
59. Van Kampen, D. (2003). Paper path modeling case in 20-SIM (Report 2003CE025). Department of Electrical Engineering, University of Twente, Enschede, The Netherlands.
60. Verghese, G. C., Levy, B. C., & Kailath, T. (1981). A generalized state-space for singular systems. *IEEE Transactions on Automatic Control, 26*, 811–831. http://doi.org/10.1109/tac.1981.1102763.
61. Vu, L., & Liberzon, D. (2008). Invertibility of switched linear systems. *Automatica, 144*, 949–958.
62. Willems, J. C. (2007). The behavioural approach to open and interconnected systems. *IEEE Control Systems Magazine, 27*(6), 46–99. http://doi.org/10.1109/MCS.2007.906923.
63. Willems, J. C. (2008). Modeling interconnected systems. In *2008 3rd International Symposium on Communications, Control and Signal Processing ISCCSP2008, St. Julians, Malta* (pp. 421–424).
64. Yip, E. L., & Sincovec, R. F. (1981). Solvability, controllability and observability of continuous descriptor systems. *IEEE Transactions on Automatic Control, 26*(3), 702–707.
65. Zimmer, D., & Cellier, F. E. (2007). Impulse-bond graphs. In *Proceedings of the 2007 International Conference on Bond Graph Modeling (ICBGM 2007), San Diego, CA* (pp. 3–11).

Part II
Bond Graph Modelling for Fault Diagnosis, Fault Tolerant Control, and Prognosis

One of the major topics of this compilation text besides BG representations of hybrid models and the generation of equations is the integration of bond graph-based fault diagnosis with fault accommodation and with fault prognosis, respectively. In the past decade, quite a number of publications have demonstrated that the BG methodology can also support model-based Fault Detection and Isolation (FDI). Analytical Redundancy Relations (ARRs) can be derived off-line from a BG of a physical model in a systematic manner and ARR residual, their numerical evaluation in real-time, can serve as fault indicators. At first, BG model-based FDI focused on systems described by a continuous time model but has meanwhile extended to systems described by a hybrid model.

The first chapter in Part II combines fault diagnosis based on ARRs derived from a BG representing a mode switching linear time-invariant model with the reconstruction of a system input through inverse simulation that accommodates for a severe fault. The chapter starts with an approach to a BG representation of hybrid system models and the derivation of a hybrid DAE system, briefly addresses a possible alternative derivation of a linear complementarity system formulation from the BG of a mode switching LTI model, discusses ARR-based as well as observer-based FDI, and presents an inverse simulation based approach to fault accommodation that uses the residuals of ARRs established for FDI.

A forward model with nominal parameters derived from a BG of the healthy system is considered as reference model. If it is coupled to the real faulty system or to its replacement by a behavioural BG model through residual sinks, then the outputs of these coupling sinks are inputs into the reference model in addition to the control input and force it to behave as the faulty system model. They can serve as fault indicators. Given the residual sink outputs and the desired system output, the forward model of the adapted reference model with nominal parameters can be considered a hybrid DAE that determines the system input to be reconstructed. The forward model of the healthy system and the input reconstruction module constitute a feedforward controller in addition to the nominal feedback controller of a closed loop system. Depending on its type, the latter one can compensate minor parametric faults in the plant and in actuators so that they are not detected by monitoring system outputs. Slightly faulty monitored values of a controlled variable due to a small sensor offset are also not detected. Input reconstruction is thus important for accommodation of severe faults. An advantage of the presented approach is that the forward model of the healthy system, the evaluation of ARRs, and the inverse model can be computed in parallel, that fault isolation and estimation is not needed, and that neither ARRs nor the reconstructed input is needed in analytical form.

Another topic of current relevance that BG modelling has been recently addressing is fault prognosis. In the context of Condition Based Management (CBM) of complex engineering systems it is important to constantly monitor a system's behaviour and to detect and isolate progressive faults. If the magnitude of an incipient fault is still below an alarm threshold, the question then is how fast a degradation of the system behaviour progresses due to that fault, i.e. when the fault magnitude will reach its alarm level so that maintenance becomes necessary because a critical failure value of a parameter, a failure threshold is reached to avoid a component or system failure.

Previous chapters demonstrate that BGs are suited to represent hybrid models of multi-energy domain systems and can support model-based FDI. As to fault prognosis, so far, a rather small number of works using a BG approach have been reported in the literature. They assume that a single component parameter value continuously drifts with time according to a known degradation model while all other parameter values remain within given uncertainty limits. Estimation of the component's Remaining Useful Life (RUL) is performed by incorporating the degradation model into the ARRs. However, systems represented by a hybrid model operate in various modes. Accordingly, ARRs and the RUL of a component are mode dependent.

The subject of Chap. 6 is an approach to an *integrated* model-based fault diagnosis and fault prognosis for systems represented by a hybrid model. If a hybrid model is in use, discrete faulty mode transitions may occur in addition to parametric faults that cause a degradation of a component's behaviour over time. The approach presented in Chap. 6 can detect and isolate discrete mode faults as well as parametric faults. If the latter ones are progressive, the component's RUL is predicted.

To overcome the problem that a fault may affect a subsequently occurring fault, the model is updated after each fault identification. For RUL estimation not one single degradation model is used but a sequence of mode dependent degradation models obtained by identification.

Finally, the integrated approach presented in Chap. 6 makes use of extensions of the Fault Signature Matrix (FSM) and the Mode Change Signature Matrix (MCSM) to reduce the set of possible fault candidates after a fault has been detected.

The chapter is based on a recently defended PhD thesis of the first author that was supervised by the second author.

Chapter 7 combines FDI based on ARRs with uncertain parameters obtained from a BG of a continuous time model with Bayesian estimation techniques for diagnosis and prognosis of incipients parametric faults in an *integrated* real-time health monitoring approach that exploits model-based and data-driven techniques and enables a prediction of the RUL of a faulty system component or of a system given various uncertainties such as parameter uncertainties or sensor noise.

BGs in Linear Transformation Form (LTF) are used for modelling a system with uncertain parameters, while Particle Filters (PFs) based on a sequential Monte Carlo method are used for estimation of the state of health of a real closed-loop feedback system. The chapter assumes that a single system parameter is a candidate for progressive degradation, that this parameter has been isolated and that a model for the degradation process is known a priori. Robust fault detection determines the beginning of the degradation process as a parametric fault, while measurement noise and noise in the degradation process are taken into account in the estimation of the state of health and in RUL prediction by means of particle filters.

Chapter 7 is also based on a recently defended PhD thesis of the first author that was supervised by the second and third author.

Chapter 5
Integrating Bond Graph-Based Fault Diagnosis and Fault Accommodation Through Inverse Simulation

W. Borutzky

5.1 Introduction

Today's closed-loop engineering systems are more and more equipped with communicating smart sensors and actuators, embedded systems and software so that they can operate autonomously to some extent. However, their increase in complexity makes them more vulnerable to malfunctions and faults that can lead to a component or a system failure if not properly dealt with in a timely manner. Safety and reliability of complex systems require that they are able to react autonomously to some faults. This can be achieved either in a passive approach by designing a controller that ensures stability and control objectives in the presence of faults from a certain class of anticipated faults, or by an active, reconfigurable fault-tolerant control approach in which only the control law is changed known as fault accommodation, or even the control loop is restructured because some sensors or actuators have become faulty so that the system remains operational and an acceptable performance in the presence of a persisting fault can be maintained. A prerequisite for fault tolerant control (FTC) is that a system is constantly monitored and faults are detected and isolated.

Approaches to fault detection and isolation (FDI) can be roughly categorised into data-driven and model-based methods. Among the latter ones are techniques based on bond graphs that derive analytical redundancy relations (ARRs) from a bond graph (BG) and check their residuals against thresholds in order to decide whether a fault has occurred [13, 26, 63, 69, 75]. An elaborated presentation of fault tolerant control may be found in [10, 45]. An excellent survey is given in [4, Chap. 2].

Bond graph model-based approaches to passive as well as to active FTC have been presented in [2, 42, 50, 63]. In active FTC, changing the controller law after

W. Borutzky (✉)
Department of Computer Science, Bonn-Rhein-Sieg University of Applied Sciences, 53754 Sankt Augustin, Germany
e-mail: wolfgang.borutzky@h-brs.de

W. Borutzky (ed.), *Bond Graphs for Modelling, Control and Fault Diagnosis of Engineering Systems*, DOI 10.1007/978-3-319-47434-2_5

a fault has occurred requires system inversion, i.e. to find an input so that the faulty system produces a desired output. One way to decide whether a model is invertible and to determine an input required to produce a desired output is to assign bicausalities to a BG [33, 42, 52, 53].

The subject of this chapter is an integrated bond graph model-based approach to FDI and to FTC. Fast switching electronic devices, hydraulic check valves or fast stops (hard limits) in mechanical motion suggest to adopt the abstraction of instantaneous, discrete state transitions, especially in the modelling of complex mechatronic systems. Therefore, hybrid models and their bond graph representation are considered. For illustration, simple switched power electronic systems are used throughout the chapter. The presented bond graph approaches may, however, be applied to other systems represented by a hybrid model as well. Power electronic devices are subject to fast switching. Over current surge, voltage overswings as well as short-circuit or open-circuit faults in transistors may happen. Therefore, FDI in power electronic circuits is important for FTC and for protection of complex mechatronic systems, especially when used in safety critical systems.

This chapter is organised into three sections. The next section addresses the modelling abstraction of instantaneous state transitions, its consequences with regard to a combined symbolic-numerical solution of a hybrid model and a bond graph representation of hybrid models.

Subsequently, a simple pragmatic approach to a causal bond graph representation of hybrid models is proposed that can use either ideal or non-ideal switches and exploits the features of the declarative modelling language Modelica®[1] [47]. The burden of a symbolic preprocessing and a numerical solution of a set of hybrid Differential Algebraic Equations (DAEs) is left to a software such as the sophisticated open-source modelling and simulation software OpenModelica [56]. The language allows for *acausal* mathematical equations and the software is able to reformulate equations as necessary and to sort the equations of a DAE system by means of the Tarjan algorithm and by using tearing [18, 32]. Therefore, the standard Sequential Causality Assignment Procedure (SCAP) is applied to a bond graph of a hybrid model. Causal strokes at switch ports result from causality assignment at storage ports and ports of resistors with a non-invertible characteristic in previous steps of the procedure and just reflect switch states in one specific system mode of operation when a maximum number of storage ports has got integral causality. Accordingly, *causal* equations are derived from a BG except for switches. The switches in the BG of a hybrid model are described by an *implicit* mathematical equation relating their two conjugate port variables. The Modelica implementation of switches accounts for their operating mode. If there are causal paths between resistors in a BG and/or storage elements in derivative causality then it is well known that they also contribute *implicit* equations. If the purpose of a study is to simulate the dynamic behaviour and if equations are formulated in Modelica, actually, one

[1] Modelica® is a registered trademark of the Modelica Association.

could even start from an *acausal* BG and read out *all* equations in *implicit* form as the OpenModelica software can determine the computational causalities of all equations and can automatically choose state variables.

If the implicit equation of a switch is not part of a subset of m algebraic equations with m unknowns, i.e. the switch does not belong to an algebraic loop, if the symbolic preprocessing of all model equations or if the numerical solution of the symbolically preprocessed equations by means of a BDF solver fails with a singular matrix in one mode of operation, then a small modification of the model is required. This problem occurs when the closing of an ideal switch turns the hybrid DAE system of the model into a higher index system. Modelica assumes that a hybrid DAE representation is *structurally time invariant*, i.e. that the set of variables and the set of equations remain fixed during simulation. Conditional equations, however, can turn some equations off or on so that the number of *active* equations can vary at run-time [32]. Physical insight into the problem combined with an account of the purpose of the model can help to add an auxiliary element of appropriate parameter value in the right place or to turn an ideal switch into a non-ideal one. An advantage of this approach is that causalities can be assigned once independent of switch states. Dynamic reassignment of causalities during simulation is taken into account automatically by the switch models implemented in Modelica. The approach is illustrated by application to a thyristor controlled half-wave rectifier circuit as a simple example of a hybrid system but is not limited to power electronic systems.

Section 5.3 discusses fault detection and isolation (FDI). It is assumed that a hybrid model of a system is given. Effective, correct and fast FDI is a prerequisite for fault accommodation which is the subject of the last section in this chapter. A contribution of bond graph modelling to FDI is that Analytical Redundancy Relations (ARRs) serving as fault indicators can be derived from a BG based on physical insight into a system in a systematic manner. It is shown that in some simple cases, ARR residuals can even be used for isolation and estimation of parametric faults. The chapter also presents an offline simulation of a fault scenario in which a short-circuit fault is deliberately introduced into an example circuit.

The last section in this chapter presents a bond graph-based fault accommodation approach that uses inverse simulation for reconstructing the input required for fault recovery.[2] In this approach, a DAE system is derived from a bond graph of the faulty system. Replacing its output by the one of the healthy system, the DAE system determines the input into the faulty system required for fault accommodation. In other words, a forward model deduced from a BG is considered a DAE system of the inverse model. The output of the healthy system model is considered the desired output for which an input must be reconstructed after a fault has happened.

Moreover, it is shown that ARR residuals can not only be used for FDI but also for input reconstruction. Advantages of an input reconstruction by inverse simulation are that there is no need for an analytical determination of the required

[2] First ideas have been presented in a paper at the 8th International Conference on Integrated Modelling and Analysis in Applied Control and Automation (IMAACA 2015) [14].

input in closed form and that existing software for the combined symbolic-numerical solution of DAE systems can be used. Disadvantages are that, in general, the solution of a DAE system of the inverse model is not guaranteed and that the inverse model is not guaranteed to be stable. One possible remedy is to slightly modify a plant model before it is inverted and to use an approximate inverse model in the controller.

The approach is applied to a buck-converter driven DC-motor. For this application to a simple often used example of a power electronic system, the hybrid model of the converter is replaced by a model with variables averaged over the switching time period. As a fault scenario, an abrupt increase of the armature resistance in the DC-motor and the recovery from this fault is studied.

5.2 Hybrid Models

Depending on a system or process to be modelled and the objectives of a study it may be appropriate to abstract fast state transitions into instantaneous ones, i.e. some state variables are assumed to change *discontinuously* at some time instances. For instance, if the purpose is to analyse the electronic behaviour of a small circuit with some few transistors in detail, then dynamic continuous time state transitions are of interest and a single set of Ordinary Differential Equations (ODEs) may be solved by using a variant of the well-known circuit simulator Spice. For larger circuits with many fast switching electronic devices, for power electronic systems or when an electronic circuit with fast dynamics is part of a mechatronic systems, it may be more appropriate to model fast switching devices as ideal, or non-ideal on-off switches. In the mechanical part of a mechatronic system, clutches or phenomena such as stick-slip friction may be represented by switches. As a result of this modelling abstraction, a model captures discrete events as well as the continuous time behaviour of a system in various modes of operation. The discrete events are either controlled by local automata or take place autonomously and cause system mode changes. Such models are usually called *hybrid* system models. In general, they can be described by a set of Differential Algebraic Equations (DAEs) that hold for the time interval between two consecutive discrete events and depend on the discrete states $m_k^{(j)}(t)$, $k = 1, \dots, n$, of all n switches which are constant during the time interval I^j when the system is in mode j, $m_k^{(j)} \in \{0, 1\}$ $\forall$ $k = 1, \dots, n$. Every feasible combination of discrete switch states represents an operational system mode j.

Let $n_f \leq 2^n$ denote the number of all physically feasible system modes and $\boldsymbol{m}^{(j)} := [m_1^{(j)}, \dots, m_n^{(j)}]^T$, let $\boldsymbol{\Theta}$ be the vector of all parameters, then a hybrid model can be described by a set of DAEs

$$\boldsymbol{F}_j(\boldsymbol{x}(t), \dot{\boldsymbol{x}}(t), \boldsymbol{u}(t), \boldsymbol{y}(t), \boldsymbol{\Theta}, \boldsymbol{m}^{(j)}) = \boldsymbol{0}, \quad j = 1, \dots, n_f \tag{5.1}$$

together with a set of conditions

$$j = C(t, \boldsymbol{x}(t), i) \tag{5.2}$$

where $\boldsymbol{F}$ denotes a vector-valued function, t the time, $\boldsymbol{x}$ the vector of unknown variables composed of state variables and algebraic variables, $\boldsymbol{u}$ the vector of known inputs and $\boldsymbol{y}$ the vector of outputs. In condition (5.2), j denotes the current system mode and i the previous one. Evaluation of the function C at the advent of a discrete event determines which DAE system is to be used to describe the dynamic behaviour for the duration of the next system mode, i.e. an instantaneous state transition from system mode i to mode j happens when $\boldsymbol{x}(t)$ reaches the surface $S_{ij} := \{x | C(\boldsymbol{x}(t), i) = j\}$. A system (5.1)–(5.2) is often termed a system of *hybrid* DAEs (see also [37, 43]).

A special case of hybrid system models is given by mode switching linear time invariant (LTI) systems that describe the dynamic continuous time behaviour between two consecutive discrete events i and j by means of a LTI system.

$$\dot{\boldsymbol{x}}(t) = \mathbf{A}(\boldsymbol{\Theta}, \boldsymbol{m}^{(j)})\boldsymbol{x}(t) + \mathbf{B}(\boldsymbol{\Theta}, \boldsymbol{m}^{(j)})\boldsymbol{u}(t), \quad j = 1, \ldots, n_f \tag{5.3}$$

where the coefficients of the matrices $\mathbf{A}, \mathbf{B}$ are constant, but their values depend on the discrete switch states $\boldsymbol{m}^{(j)}$ in system mode j.

A formal definition of hybrid models may be found in [60, 74].

5.2.1 Solution of Hybrid Models

The abstraction of instantaneous state changes entails a number of consequences with regard to the symbolic-numerical solution of a hybrid model.

- As switches connect and disconnect parts of a hybrid model, the latter one is of *variable* structure. The connection of model parts by means of an ideal switch may lead to structural singularities and as result to a higher-index problem.
- The number of state variables, the index of a DAE system and the equations for fault indicators may be mode-dependent.
- Re-initialisation of the numerical integration is necessary at the advent of discontinuous *state* changes.
- Ideal switches entail *variable* computational causalities and may result in model equations that cannot be numerically computed by means of the widely used DASSL solver so that appropriate small modifications of the model may become necessary that require physical insight.
- Non-ideal switches replace instantaneous discontinuous state changes by fast continuous state transitions and thus lead to stiff model equations in general.
- Pantelides algorithm [59] for the determination of a set of consistent initial conditions cannot be used for a reduction of the index of a hybrid DAE system

as it requires that the index does not change [19]. Index reduction could be applied to a DAE system valid for the time span between two discrete events. This, however, would mean not only to re-initialise numerical integration at the advent of a discrete event but also to interrupt numerical integration, to apply index reduction to the DAE system for the following time span, to generate new code and to resume simulation with a new modified DAE system of index ≤ 1 which is inefficient.

Another possible approach is to generate a unique hybrid DAE system valid for all modes of operation from a BG, to identify all physically feasible switch combinations, to establish DAE systems for all admissible modes of operation and to apply index reduction if the index of one of the DAE systems is greater than one. The result then is a set of DAE systems of index ≤ 1 obtained *offline*. Simulation of the dynamic behaviour then switches from one preprocessed DAE system to another when the mode of operation changes according to a finite state automaton (FSA) capturing all system modes and the discrete transitions between them. For mode switching LTI models with ideal switches, Cormerais et al. developed a program that determines all physically feasible modes of operation and establishes an implicit state-space system in symbolic form for each mode of operation starting from a reference configuration by using Mathematica® [23, 76].

In the case of a LTI model with ideal switches, yet another two options could be to express the port variables of the switches by means of state variables and time derivatives of state variables and to substitute these expressions into the implicit equations of the switches. The result then is a DAE system of index $\leq$ 1 depending on the mode of operation. Or finally, one could solve the hybrid variable index DAE system for all modes of operation by a program that detects discrete events during the evaluation of the current continuous time model, re-initialises the numerical integration and computes the DAE system for the time span between two consecutive discrete events by means of the implicit Runge–Kutta code RADAU5 [35] that can solve linear implicit DAE systems $\mathbf{M}\dot{\boldsymbol{x}}(t) = \mathbf{f}(t, \boldsymbol{x}(t))$ of index $1, 2$ or 3 where $\mathbf{M}$ is a square constant coefficient.

A declarative modelling language that is based on the synchronisation of DAEs and discrete equations [30, 57] and that offers constructs suited for the description of hybrid models is Modelica® [47]. It is supported, e.g., by the sophisticated open source modelling and simulation software OpenModelica [56]. The software accepts *acausal* equations and their combined symbolic and numerically solution features include

- symbolic preprocessing of equations,
- discrete event detection during the evaluation of a continuous time model for the current system mode,
- initialisation at start time and re-initialisation of the numerical integration at discrete state events.

The language features and its software support enable the following approach.

- The Standard Causality Assignment Procedure (SCAP) is applied to a bond graph of a hybrid model. The port of a switch obtains its causality either as possible or as required by other elements independent of the state of the switch.
- Switches may be either ideal or non-ideal. The Modelica implementation of their static behaviour accounts for their two discrete states.
- For models of medium complexity, equations may be derived directly from a causal bond graph in a systematic manner. In this process, unknown variables are eliminated by following causal paths except port variables of switches.
- For switches, an *implicit* algebraic equation relating their port variables by means of a discrete state variable is written. Based on the value of the discrete state of a switch its Modelica implementation determines the output variable of the switch.

If an ideal switch causes a problem because of an inertia element in series with the switch or because two storage elements become dependent when the switch is closed, a possible remedy is to turn it into a non-ideal switch. Another option in the first case is to replace the differential equation of the inertia by a numerical integration formula. In circuit simulation, the result is known as companion model of the storage element [15]. This is also supported in Modelica® and is called inline integration [19, 31]. The problem of dependent storage elements which Cellier denotes as a structural singularity means that the DAE system is of higher index which may be resolved in OpenModelica by symbolic index reduction if the index is time-independent. As a result, in some cases, the symbolic-numerical solution of a hybrid model in OpenModelica needs some help from the modeller. Recently, because of the limitations of handling variable structure models in the Modelica framework, Zimmer has extended Modelica concepts and has developed an experimental dynamic DAE processor that can handle changes in the set of equations at run time and is thus more suited for the solution of variable structure models [78, 79].

Modelica® and OpenModelica have been used for the subsequent examples in this chapter. For medium scale bond graph models, equations can be immediately read out from a hierarchical causal bond graph eliminating intermediate variables by following causal paths and can be directly formulated in Modelica. An equation-based description of hybrid models allows to directly exploit the features of an object-oriented acausal modelling language such as Modelica and to make use of functional blocks available from the free Modelica Standard Library. The result can be a concise efficient model description with a reduced set of equations to be solved repeatedly in a simulation loop. On the other hand, bond graph elements and the interconnection of their ports can be described in Modelica [11, Chap. 11.5]. Accordingly, the freely available library *BondLib* developed by Cellier [20] may be used for a graphical creation of bond graph models in OpenModelica.[3] The library is suited for the development of bond graphs for hybrid models as it contains models for the ideal and the non-ideal switch as well as for diodes. Models created either graphically or directly formulated in Modelica undergo a symbolic preprocessing, i.e. a flattening of the model hierarchy, a sorting and optimisation of equations before C++ code is generated.

Furthermore, as to the numerical computation of hybrid models described in Modelica, ongoing research work is reported in the literature [8] that allows to apply Quantised State System (QSS) Solvers developed by Kofman [40] without prior manual modification of the Modelica description. It appears that numerical computation of hybrid models based on state quantisation offers advantages over traditional solvers that use time discretisation such as DASSL and performs better than the latter ones.

5.2.2 *Bond Graphs of Hybrid Models*

Various approaches to a bond graph representation of hybrid models have been proposed in the literature. A comprehensive survey may be found in [13]. Two main different routes can be distinguished. The dynamic behaviour of fast switching devices, fast state transitions may be roughly approximated by means of either ideal or non-ideal switches. Both approaches do have their pros and cons.

In [13], a non-ideal switch representation introduced by Ducreux et al. [29] back in 1993 has been used. The current, i_D, versus voltage drop, u_D, characteristic of a diode is approximated by a linear one in the ON- as well as in the OFF-mode. In the latter mode, the conductance is neglected (cf. Fig. 5.1).

This bond graph representation has a number of advantages for modelling and simulation of fault scenarios.

- There is a clear one-to-one mapping between a switching device in an initial system schematic and its representation in a bond graph.
- There is one single bond graph that holds for all system modes. The current system mode is determined by the Boolean values of the MTFs of all switches.

[3] The library, BondLib v2.3, is encoded in Modelica 2.2.1 while version 1.9.3 of the OpenModelica software supports Modelica 3.2.1. To use BondLib in the OpenModelica editor OMEdit it is necessary to have the library converted to Modelica 3.x.

A feature of the library somewhat uncommon to bond graph modellers is that bonds are to be picked and placed like BG elements. Bonds and BG elements have got connectors that are displayed by little grey dots and are to be connected in a rubber band manner. In bond graph modelling environments such as 20sim® [22], bonds with proper attributes are created by connecting power ports. Connecting connectors may lead to errors that show up not before the model hierarchy has been flattened so that error messages may not indicate the cause for a problem. Therefore, a modification of the connectors used in BondLib has been recently proposed [25].

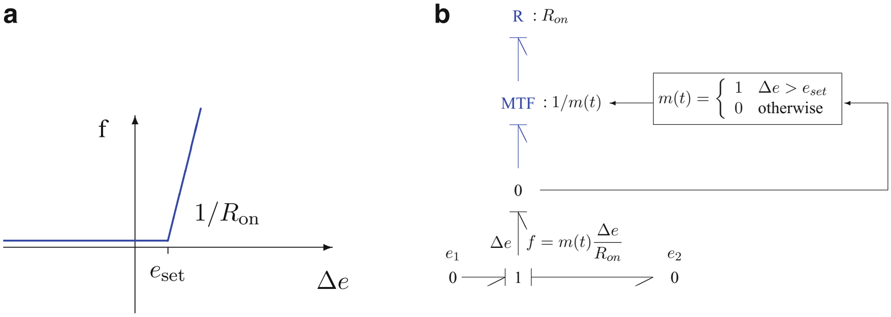

a
f
$1/R_{\mathrm{on}}$
e_{set}
Δe
b
R : R_{on}
MTF : $1/m(t)$
$m(t) = \begin{cases} 1 & \Delta e > e_{set} \\ 0 & \text{otherwise} \end{cases}$
0
Δe
$f = m(t)\dfrac{\Delta e}{R_{on}}$
e_1
e_2
0
1
0

Fig. 5.1 Bond graph representation of a non-ideal switch [13]. (**a**) Piecewise linear approximation of the non-linear characteristic of a diode. (**b**) Bond graph model of a switching device in mode-independent conductance causality

- The mode-independent fixed conductance causality of the switches and the application of the SCAP result in causalities that do not change with changes of the system mode.
- A single set of model equations as well as equations for fault indicators valid for all system modes can be (automatically) derived from the bond graph. The equations contain the discrete moduli $m_i(t) \in \{0, 1\}$, $i = 1, \dots, n_s$, of the switching MTFs.

where n_s denotes the number of switches in a bond graph model.

Causal conflicts may result that can be removed by adding an auxiliary storage element or a resistor with appropriate parameter value. Some of these parameters may be set to zero after symbolic reformulation of equations so that small time constants in linearised models can be avoided. For some switches, even their ON-resistance R_{on} can be neglected turning them into ideal switches. Elements with a small parameter value cannot always be avoided and it may be justified from a physical point of view to account for them.

The use of a non-ideal and of an ideal switch model shall be illustrated by means of the simple half-wave rectifier circuit depicted in Fig. 5.2.

Illustrative Example

Figure 5.2 shows the circuit schematic created by means of the OpenModelica editor OMEdit.

The sinusoidal signal provided by the voltage source is rectified by means of the thyristor. Its gate is controlled by a trigger signal. Transformation of the circuit schematic into a bond graph is straightforward (Fig. 5.3).

For convenience, the thyristor is represented by a non-standard generic Sw-element which may stand for the MTF-R:R$_{\mathrm{on}}$ pair in Fig. 5.1 or an ideal switch. The sensors of the currents through the inductances, i_{L_1}, i_{L_2} and of the voltage u_C across the capacitor are taken into account by the detector elements Df and De

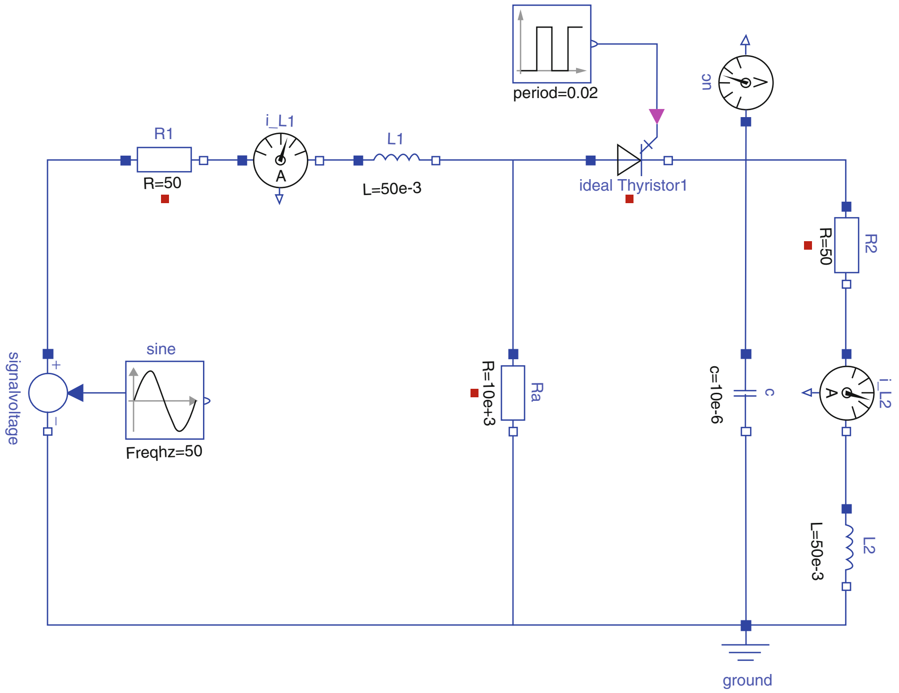


Fig. 5.2 Schematic of a thyristor-controlled half-wave rectifier circuit

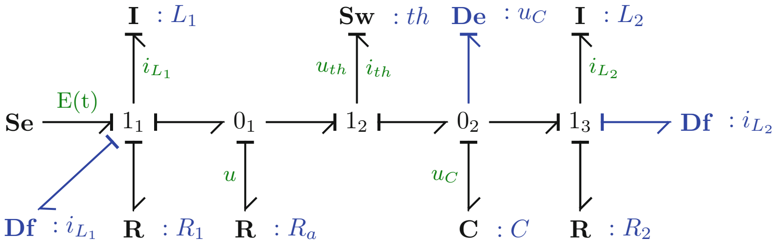


Fig. 5.3 Bond graph of the half-wave rectifier circuit in Fig. 5.2

highlighted in blue colour. Causality strokes have been obtained by application of the Standard Causality Assignment Procedure (SCAP). The resistor $R : R_a$ is an auxiliary element with a high resistance R_a that is needed to make the model solvable in the case the thyristor is considered an ideal switch. This can be seen by considering the equations derived from the causal bond graph.

$$u = R_a(i_{L_1} - i_{th}) \tag{5.4}$$

$$0 = m \cdot i_{L_1} + (1 - m) \cdot u_{th} \tag{5.5}$$

$$u = u_{th} + u_C \tag{5.6}$$

$$L_1 \frac{d}{dt} i_{L_1} = E - R_1 i_{L_1} - u \tag{5.7}$$

$$C\dot{u}_C = i_{th} - i_{L_2} \tag{5.8}$$

$$L_2 \frac{d}{dt} i_{L_2} = u_C - R_2 i_{L_2} \tag{5.9}$$

Implicit Eq. (5.5) expresses that for an ideal switch either the current i_{th} or the voltage drop u_{th} across the element is equal to zero depending on the value of the discrete switch state m. On the right-hand side of (5.8), variable i_{th} is not a state. It is determined by the set of algebraic equations (5.4)–(5.6). The latter one has a unique solution that is valid for each of the two switch states.

$$i_{th} = \frac{m-1}{m-(1-m)R_a}(R_a i_{L_1} - u_C) \tag{5.10}$$

or

$$i_{th} = \begin{cases} i_{L_1} - \dfrac{u_C}{R_a} & m = 0 \quad \text{(switch closed)} \\ 0 & m = 1 \quad \text{(switch open)} \end{cases} \tag{5.11}$$

as to be expected.

5.2.3 *Derivation of a Hybrid DAE System from the Bond Graph of a Hybrid Model*

In the illustrative example, the behaviour of the ideal switch is described by *implicit* Eq. (5.5) independent of the assigned causality. Therefore, the causal path from the Sw-element to the auxiliary resistor R : R_a could also be reversed. Only (5.4) would have to be reformulated and the algebraic system would still be solvable. This observation can be exploited when using a software such as OpenModelica that accepts *acausal* equations. That is, switching devices may be modelled as ideal switches. Causality is assigned to a bond graph following the SCAP. Derivation of equations from the BG accounts for assigned causalities except for the ideal switches. The latter ones are given by an *implicit mode-dependent* equation relating their two port variables. It is then left to the software to reduce the system of equations symbolically if possible and to solve the resulting set of equations numerically.

However, this is not always feasible. If $R_a \rightarrow \infty$, i.e. if the auxiliary resistor R : R_a is missing, then the equations derived from the bond graph read

$$0 = i_{L_1} - i_{th} \tag{5.12}$$

$$0 = m \cdot i_{th} + (1-m) \cdot u_{th} \tag{5.13}$$

$$u = u_{th} + u_C \tag{5.14}$$

$$L_1 \frac{d}{dt} i_{L_1} = E - R_1 i_{L_1} - u \tag{5.15}$$

$$C\dot{u}_C = i_{th} - i_{L_2} \tag{5.16}$$

$$L_2 \frac{d}{dt} i_{L_2} = u_C - R_2 i_{L_2} \tag{5.17}$$

In order to express voltage u on the right-hand side of (5.15) by state variables, the algebraic system (5.12)–(5.14) must be solved for u which fails for $m = 1$.

$$0 = m \cdot i_{L_1} + (1 - m) \cdot u - (1 - m) \cdot u_C \tag{5.18}$$

If an auxiliary resistor R : R_a is used, then for reasons of accuracy, a finite high value of resistance R_a should be chosen.

Another option is to replace the ideal switch by a non-ideal one with a resistance that either takes the value R_{on} or R_{off} depending on the discrete state of the switch. That is, the causality stroke at the port of the non-ideal switch is assigned once by following the SCAP. The value of its resistance $R_{sw}(m)$, however, is mode-dependent.

$$R_{sw}(m) = \frac{m + (1 - m) \cdot R_{\text{on}}}{m \dfrac{1}{R_{\text{off}}} + 1 - m} \tag{5.19}$$

Assuming a non-ideal switch model and omitting the auxiliary resistor R : R_a, the following equations can be derived from the bond graph in Fig. 5.4.

$$u_{th} = R_{sw}(m)\, i_{L_1} \tag{5.20}$$

$$L_1 \frac{d}{dt} i_{L_1} = E - R_1 i_{L_1} - u_{th} - u_C \tag{5.21}$$

$$L_2 \frac{d}{dt} i_{L_2} = u_C - R_2 i_{L_2} \tag{5.22}$$

Due to the mode-dependent switch resistance $R_{sw}(m)$, this model holds for both switch modes and can be solved.

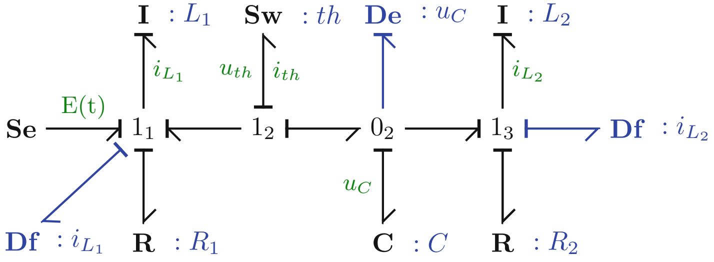


Fig. 5.4 Bond graph of the half-wave rectifier circuit in Fig. 5.2 without the auxiliary resistor R_a

```
record parameters
  // Half-wave rectfier circuit
  //
  constant  Real pi = Modelica.Constants.pi;
  // Input Voltage
  parameter Real omega = 2 * pi * f;
  parameter Real A = 30 "Amplitude [V]";
  parameter Real f = 50 "Frequency [Hz]";
  // thyristor gate signal
  parameter Real period  = 1 / f "Times for one period [s]" ;
  constant  Real  startTime = 0.002 "[s]";
  parameter Real width   = 10 "Widths of pulses in % of periods";
  parameter Real T_width = period * width / 100;
  //
  // Thyristor (non-ideal switch):
  parameter Real Roff = 1e-5 "[Ohm]";
  parameter Real Gon  = 1e-5 "[Ohm]";
  // Auxiliary resistor:
  parameter Real Ra    = 10e+3 "[Ohm]";
  //
  // circuit parameters
  parameter Real R1 = 10 "[Ohm]";
  parameter Real R2 = 50 "[Ohm]";
  parameter Real C  = 10e-6 "[F]";
  parameter Real L1 = 50e-3 "[H]"; // 10e-10 "[H]";
  parameter Real L2 = 50e-3 "[H]";
end parameters;
```

Fig. 5.5 Parameters of the half-wave rectifier circuit

Simulation runs using the parameters in the Modelica record in Fig. 5.5 give the same results in both cases, i.e. if the switch is assumed to be ideal and if an auxiliary resistor is used to make the model solvable or if the auxiliary resistor is omitted and the ideal switch is replaced by a non-ideal one (Fig. 5.6).

In general, a disadvantage of a non-ideal switch model accounting for an ON-resistance R_{on} and an OFF-resistance R_{off}, is that it causes small time constants in a linearised model, i.e. the model becomes a set of stiff equations.

In this simple example, the inductors in series with the thyristor cause the problem described above. The closed ideal switch connects the inductor $\mathrm{I} : L_2$ and the capacitor $\mathrm{C} : C$ with the inductor $\mathrm{I} : L_1$. In that mode, the order of the model is three. If the switch is open and if there is no auxiliary resistor, inductor $\mathrm{I} : L_1$ is disconnected and the model order reduces to two.

The problem can be overcome by deriving from the BG expressions for the port variables of the ideal switch in terms of state variables *and if necessary time derivatives of states* and by substituting them into the implicit equation of the ideal switch. If the switch is considered ideal then the following equations are obtained from the BG in Fig. 5.4.

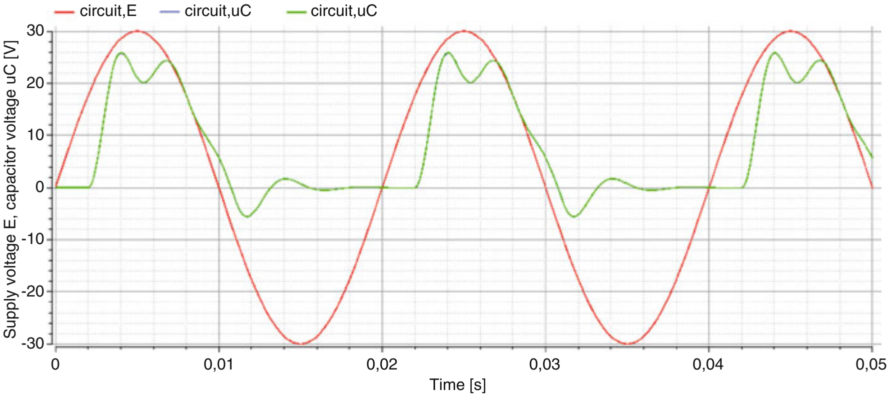


Fig. 5.6 Time evolutions of the input voltage E and the voltage u_C across the capacitor

$$i_{th} = i_{L_1} \tag{5.23}$$

$$u_{th} = E - R_1\, i_{th} - L_1 \frac{d}{dt} i_{L_1} - u_C \tag{5.24}$$

$$0 = m\, i_{th} + (1-m)\, u_{th} \tag{5.25}$$

Elimination of i_{th}, u_{th} in (5.25) gives

$$(1-m)\, L_1 \frac{d}{dt} i_{L_1} = [m - (1-m)R_1] i_{L_1} + (1-m)(E - u_C) \tag{5.26}$$

Equation (5.26) together with the state equations of the other two storage elements $\mathrm{C} : C$ and $\mathrm{I} : L_2$ is a DAE system that holds for all modes of operation. For $m = 0$ (closed switch), the constitutive equations of all storage elements constitute an explicit ODE system of order 3. For $m = 1$ (open switch), ODE (5.26) correctly turns into the algebraic equation $i_{L_1} = 0$ and accordingly, (5.15) gives $u_{th} = E - u_C$ while the state equations of the other two storage elements build an explicit ODE system of order 2.

The same result would be obtained by an approach that takes ideal structural switching into account by *controlled* junctions and uses the concept of *dynamic* causalities as has been proposed in [46] and presented in the previous chapter.

In the case that there are no inductors, modelling the thyristor as an ideal switch causes no problems. Figure 5.7 shows the simplified bond graph.

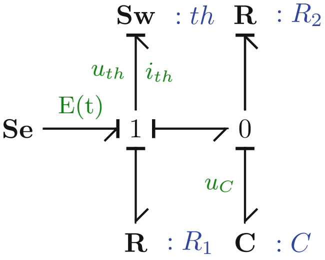
Using both port variables of the ideal switch, i.e. by not eliminating but keeping them, the following equations can be derived.

$$u_{th} = E - R_1\, i_{th} - u_C \tag{5.27}$$

$$0 = m\, i_{th} + (1-m)\, u_{th} \tag{5.28}$$

$$C\, \dot{u}_C = i_{th} - \frac{u_C}{R_2} \tag{5.29}$$

Fig. 5.7 Bond graph of the rectifier with no inductances



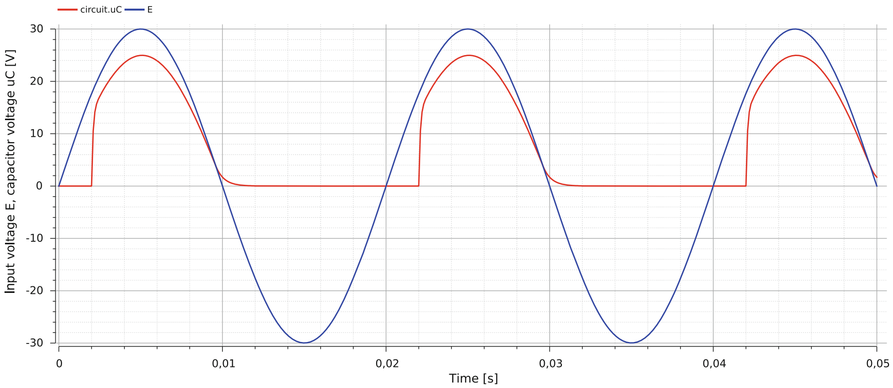


Fig. 5.8 Time evolutions of voltages E and u_C for a rectifier with no inductances

The algebraic equations (5.27)–(5.28) can be solved for i_{th}.

$$i_{th} = \frac{m-1}{m-(1-m)R_1}(E-u_c) \tag{5.30}$$

This expression is valid for both switch states. Figure 5.8 shows the time evolutions of voltages E and u_C for a rectifier with no inductances.

Finally, the partial listing in Fig. 5.9 shows a widely used Modelica description of a thyristor considered as a non-ideal switch [18, 58, 67]. The variable s denotes an independent curve parameter of the current versus voltage drop characteristic of a diode indicating whether the diode is blocking or conducting. In contrast to a diode, a thyristor is only conducting if the voltage drop is positive and if the trigger signal at its gate is positive. The Boolean variable `fire` is determined by the trigger signal. The Boolean variable `Off` indicates whether the discrete state of the switch is off, `pre(off)` refers to the previous state.

The Modelica description of a thyristor can be easily turned into that of an ideal switch by setting $G_{\text{on}} = R_{\text{off}} = 0.0$. The equations in lines 12 and 13 of Fig. 5.9 then read

```
  ...
  // Thyristor
  Real uTh, iTh, s;
  Boolean off(start = true );
  Boolean fire(start = true );
  //
  ...
equation
  ...
  // Thyristor:
  off = s < 0 or pre(off) and not fire;
  uTh = s * (if off then 1 else Roff);
  iTh = s * (if off then Gon else 1);
  u = uTh + uC;
  ...
```

Fig. 5.9 Modelica description of the thyristor

$$u_{Th} = s \cdot \underbrace{(\text{if } \texttt{off} \text{ then } 1 \text{ else } 0)}_{=:\, 1-b} \tag{5.31}$$

$$i_{Th} = s \cdot \underbrace{(\text{if } \texttt{off} \text{ then } 0 \text{ else } 1)}_{=:\, b} \tag{5.32}$$

Elimination of s gives

$$0 = b\, u_{Th} - (1-b)\, i_{Th} \tag{5.33}$$

Note that `off` does not only account for the region of operation but also for the trigger signal at the gate of the thyristor. Singularities that may occur with the use of an ideal switch can be avoided by a slight modification of the discrete switch state b.

$$b = \text{if } \texttt{off} \text{ then low else high} \tag{5.34}$$

where low := ϵ and high := $1-\epsilon$ and $0 < \epsilon \ll 1$. This turns the ideal switch into a non-ideal one with $R_{\text{on}} = 1/R_{\text{off}} = \epsilon/1-\epsilon$.

The description can also be used for a diode. Only the condition (`and not fire`) has to be removed. Figure 5.10 displays the finite state automaton (FSA) of a thyristor. Only the requirement of a positive gate voltage v_G has to be removed in the condition for the transition from the ON-state into the OFF-state in order to obtain the FSA of a diode.

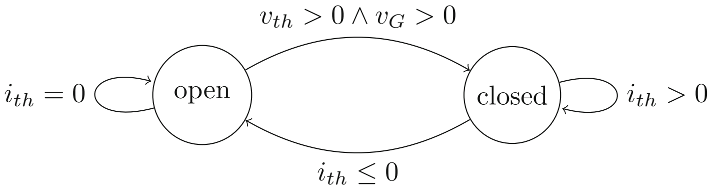


Fig. 5.10 Finite state automaton of a thyristor

5.2.4 Derivation of a Linear Complementary System from a Bond Graph of a Mode Switching LTI Model

Throughout this chapter, equations derived from a bond graph of a hybrid model are in the form of a hybrid DAE system and the DASSL solver integrated into OpenModelica is used for its numerical solution. Alternatively, equations in the form of a *linear complementarity system* (LCS) may be derived from a bond graph of a mode switching LTI system.

Let the system be in mode $\boldsymbol{m}$ where $\boldsymbol{m}$ denotes the vector of all n present discrete switch states. Then for the port of the ith ideal switch two complementarity variables z_i, w_i are chosen that fulfill the so-called *complementarity conditions*

$$z_i(t) \geq 0 \quad w_i(t) \geq 0 \quad z_i(t)w_i(t) = 0 \qquad \forall\, i = 1, \ldots, n \qquad \forall\, t \geq 0 \tag{5.35}$$

often briefly expressed as

$$\mathbf{0} \leq \boldsymbol{z} \perp \boldsymbol{w} \geq \mathbf{0} \tag{5.36}$$

and understood componentwise.

For instance, let i_{D} be the current through a diode modelled as an ideal electronic switch and v_{D} the voltage drop across the diode then its complementarity variables are $z = i_{\mathrm{D}}$ and $w = -v_{\mathrm{D}}$.

If there are no dependent storage elements and no causal paths between resistive ports in a BG of a mode switching LTI model then a state space model for system mode $\boldsymbol{m}$ can be derived from the BG in the form of a linear complementarity system

$$\dot{\boldsymbol{x}}(t) = \mathbf{A}\boldsymbol{x}(t) + \mathbf{B}\boldsymbol{z}(t) + \mathbf{E}\boldsymbol{u}(t) \tag{5.37a}$$

$$\boldsymbol{w}(t) = \mathbf{C}\boldsymbol{x}(t) + \mathbf{D}\boldsymbol{z}(t) + \mathbf{F}\boldsymbol{u}(t) \tag{5.37b}$$

where $\boldsymbol{x}(t)$ denotes the state vector and $\boldsymbol{u}(t)$ the vector of all inputs into the system, i.e. the outputs of all independent sources Se and Sf. The matrices in (5.37) are of appropriate dimensions. They can be obtained from a BG by collecting all switches, all sources, all storage elements and all resistors in separate fields and by partitioning the junction structure matrix accordingly.

Discretising the time derivative of the state $\boldsymbol{x}(t)$ by means of the numerical backward Euler integration method and substituting (5.37a) into (5.37b) yields the linear complementarity problem (LCP)

$$\boldsymbol{w}_{k+1} = \mathbf{M}\boldsymbol{z}_{k+1} + \boldsymbol{q}_k \ , \quad \boldsymbol{w}_{k+1} \geq \mathbf{0} \ \wedge \ \boldsymbol{z}_{k+1} \geq \mathbf{0} \ , \quad \boldsymbol{w}_{k+1}^T \boldsymbol{z}_{k+1} = 0 \tag{5.38}$$

to be solved for each discrete time t_k, $k \in \mathbb{N}$. That is, given matrix $\mathbf{M}$ and vector $\boldsymbol{q}$, find vectors $\boldsymbol{z}$ and $\boldsymbol{w}$ such that (5.38) is fulfilled.

The study of the linear complementarity problem and the description of hybrid models in the complementarity formalism has produced a body of theoretical results (see for instance [7, 17, 70]) and computational methods reported in the literature. One of the results is that the LCP (5.38) has a unique solution if and only if all minors of the matrix $\mathbf{M}$ are positive. For a numerical solution of the LCP, the software Siconos, for instance, may be used [1].

5.3 Fault Diagnosis

According to a list of definitions compiled by the Safeprocess Technical Committee of IFAC, the International Federation of Automation and Control [10, 39], fault diagnosis means to detect and to isolate faults and to analyse their type and their magnitude. Methods for fault diagnosis may be classified into those that exploit process history, or make use of qualitative models or are based on quantitative models [71–73]. A brief survey of fault detection methods may be found in [13]. With regard to their occurrence, faults can be categorised into abrupt, incipient, progressive or intermittent parametric faults and into discrete mode faults.

In the following, bond graph-based fault detection in hybrid models focuses on faults that change the model structure such as short circuit or open circuit faults, i.e. discrete event faults that cause a failure of a switching element and on faults that can be mapped onto parameter changes.

5.3.1 ARR-Based Fault Detection

Fault indicators can be obtained by evaluation of analytical constraints between known input signals into a system and measured output signals called Analytical Redundancy Relations (ARRs) in the literature. The result of an ARR evaluation is termed an ARR residual. Let $\boldsymbol{u}(t), \boldsymbol{y}(t)$ be the input and the output signal respectively, $\boldsymbol{m}^{(j)}(t)$ the vector of all discrete switch states at time t when the system is in mode j, $\boldsymbol{\Theta}$ the vector of parameters, and f_k a real-valued function. The discrete switch states are constant while the system is in mode j and take values $m_\sigma^{(j)} \in \{0, 1\}$, $\sigma = 1, \ldots, n_s$, where n_s denotes the number of switches. Then, in system mode j, the kth residual takes the form

$$r_k(t) = f_k(\boldsymbol{u}(t), \boldsymbol{y}(t), \boldsymbol{\Theta}, \boldsymbol{m}^{(j)}) \tag{5.39}$$

Some components of the parameter vector $\boldsymbol{\Theta}$ may take constant nominal values, others may be uncertain or slowly time varying in comparison to the time history of the state variables, or a parameter value, i.e. may abruptly jump to another constant value in the case of an abrupt parametric fault.

If these ARR residuals are close to zero for the time interval I^j in which the system is in operating mode j, then the system is in a healthy state for that time interval.

$$|r_k(t)| < \epsilon_k^j \qquad \forall\ k \quad \forall\ t \in I^j \tag{5.40}$$

where ϵ_k^j denotes an admissible small error bound for $r_k(t)$ in system mode j that accounts for e.g. numerical inaccuracies.

If there is at least one ARR residual r_i that exceeds a given fault threshold thr_i^j in system mode j, where thr_i^j depends on user defined specifications accounting for modelling and parameter uncertainties, measurement uncertainties and noise, then this indicates that at least one single fault has happened. In hybrid system models, parameters as well as the structure of the model equations and thus the structure of ARRs may change when a fault occurs due to discrete switch states included in the equations. If a switch fails then a part of an ARR may be permanently enabled or disabled which may lead to residuals exceeding their fault thresholds.

$$\exists\ i \quad \exists\ I_i^j \subset \mathbb{R}^+ \quad \text{such that} \quad |r_i(t)| \geq \mathrm{thr}_i^j \qquad \forall\ t \in I_i^j \tag{5.41}$$

Moreover, parameters can be uncertain, i.e. values may deviate from their nominal values. In order to avoid false alarms, fault indicators should be insensitive to parameter variations but sensitive to true faults. For fault detection robust with regard to uncertain system parameters, adaptive mode-dependent fault detection thresholds $\mathrm{thr}_k^j(t)$ can be determined [12, Chap. 5], [27]. If an ARR residual has reached such a fault detection threshold due to a *time varying* parameter, $\Theta(t)$, a supervisory system may decide to still tolerate the progressive fault, but *failure prognosis* will start to determine the remaining useful life (RUL) of the component and to decide when maintenance actions are to be taken.

One of the contributions of bond graph modelling to fault detection and isolation (FDI) is that ARRs for fault diagnosis can be derived from a bond graph in a systematic way. The bond graph used for FDI has been termed *diagnostic* bond graph (DBG) in the literature [62]. First, a bond graph with detectors accounting for the available sensors is set up. Computational causalities are then assigned in such a way that detectors are in inverted causalities and storage elements are in derivative causality. The reason for this modification of the SCAP is that outputs of sensors attached to a real system are measured quantities and thus can be considered *known* inputs into the DBG model. Furthermore, if the BG model is connected to a real system, it is difficult to reconstruct initial values for ODEs if storage elements are in integral causality. A drawback is that measured inputs are to be differentiated. As measurements are overloaded with noise, they should be properly filtered and differentiation is to be performed in discrete time.

ARRs are obtained from a DBG by summing power variables at those junctions to which a detector is attached. If constitutive element equations permit to eliminate unknown variables by following causal paths then ARRs in explicit form can be obtained. An ARR could be established by summing the power variables at a junction with no detector attached as it can be algebraically obtained from ARRs of detector junctions. In any case, the number of ARRs equals the number of available sensors. Generation of ARRs from a BG has been implemented in the commercial software Symbols® [38].

The information which component parameter contributes to which ARR in some system mode can be expressed in a structural Fault Signature Matrix (FSM) that is called All-mode FSM [13] or Global FSM [41] for hybrid system models with rows for the component parameters or components such as sources and sensors and columns for the ARRs. A parameter fault can be detected if the parameter contributes to an ARR in some system mode. Detectability of a fault is usually indicated in an additional column. As it is likely that there are more parameters than available sensors, i.e. the FSM is not square, there are component fault signatures that are not unique in some system modes. As a result, parameter faults cannot be isolated by just inspecting the structure of a FSM. However, information can be improved if entries in a structural FSM are replaced by parameter sensitivities of ARR residuals turning the matrix into a sensitivity fault signature matrix, and if those parameters that contribute only weakly to an ARR are set to zero.

ARRs can be established off-line. Clearly, their evaluation giving residuals has to be performed in real-time if measured inputs into a DBG model are obtained from an engineering system. Moreover, in order to avoid that false alarms are sent to a supervisory system, ARR residuals as fault indicators should be insensitive to parameter variations but should clearly indicate a fault that has happened. ARR based fault detection robust with regard to uncertain parameters can be supported by bond graphs in linear fractional transformation (LFT) form [12, Chap. 3] or by incremental bond graphs [13].

Example. Consider the bond graph of the thyristor-controlled half-wave rectifier in Fig. 5.2. The thyristor is modelled as an ideal switch. Summation of power variables at junctions with a detector attached yields the following ARRs r_1, r_2, r_3.

$$i_{th} = \frac{1-m}{(1-m)\,R_a - m}\,(R_a i_{L_1} - u_C) \tag{5.42}$$

$$u = R_a\,(i_{L_1} - i_{th}) \tag{5.43}$$

$$r_1 = E - R_1 i_{L_1} - L_1 \frac{d}{dt} i_{L_1} - u \tag{5.44}$$

$$r_2 = i_{th} - C\dot{u}_C - i_{L_2} \tag{5.45}$$

$$r_3 = u_C - R_2 i_{L_2} - L_2 \frac{d}{dt} i_{L_2} \tag{5.46}$$

Table 5.1 FSM for the thyristor controlled half-wave rectifier circuit in Fig. 5.2

Component	Parameter/output	r_1	r_2	r_3	D_b	I_b
R : R_1	R_1	1	0	0	1	0
L : L_1	L_1	1	0	0	1	0
R : R_a	R_a	1	1	0	1	0
Sw : Thy	m	1	1	0	1	0
C : C	C	0	0	1	1	1
R : R_2	R_2	0	0	1	1	0
L : L_2	L_2	0	0	1	1	0
Se	$E(t)$	1	0	0	1	0
Df : i_{L_1}	$i_{L_1}(t)$	1	1	0	1	0
De : u_C	$u_C(t)$	0	1	0	1	0
Df : i_{L_2}	$i_{L_2}(t)$	0	0	1	1	0

A structural fault signature matrix in Table 5.1 accordingly indicates which component parameter contributes to which ARR.

In Table 5.1, the last four rows can be omitted if sensors can be considered faultless and if measurement noise can be neglected. The last but one column on the right side with the heading D_b indicates that all possible faults can be detected. The most right column with the heading I_b indicates that none of them can be isolated with the given set of sensors except a fault in the capacitance C. This is not surprising as there is only a sensor for the current through resistor R : R_1, inductor I : L_1, and the thyristor connected in series. The voltage sensor across the capacitor and the current sensor in the series connection of elements R : R_2, inductor I : L_2 in parallel to the capacitor are not sufficient to isolate a fault in these three elements.

Off-Line Simulation of a Single Fault Scenario

For an off-line simulation of a fault scenario, the real world rectifier circuit is replaced by a behavioural bond graph model. The latter one is coupled to a diagnostic bond graph as displayed in Fig. 5.11. The detector elements in the couplings of the two models denote *virtual* sensors of the ARR residuals. To distinguish them from sensors of real measurements, they are marked with a star ($*$). The behavioural model provides substitutes for real measurements that are fed into a diagnostic bond graph model. The latter one provides the residuals serving as fault indicators (cf. Fig. 5.12).

The fault scenario assumes that the trigger signal at the gate of the thyristor is faultless but that the thyristor is short-circuited as of $t_e = 0.055$ s.

Figure 5.13 displays the time evolution of voltages E and u_C and of residuals r_1, r_2, r_3. As can be seen from Fig. 5.13a, there is no rectification of the sinusoidal input voltage $E(t)$ any more as of $t_e = 0.055$.

Figure 5.13b–d indicates that residuals r_1, r_2 are sensitive to a short-circuit in the thyristor while residual r_3 is not which is in accordance with the FSM in Table 5.1.

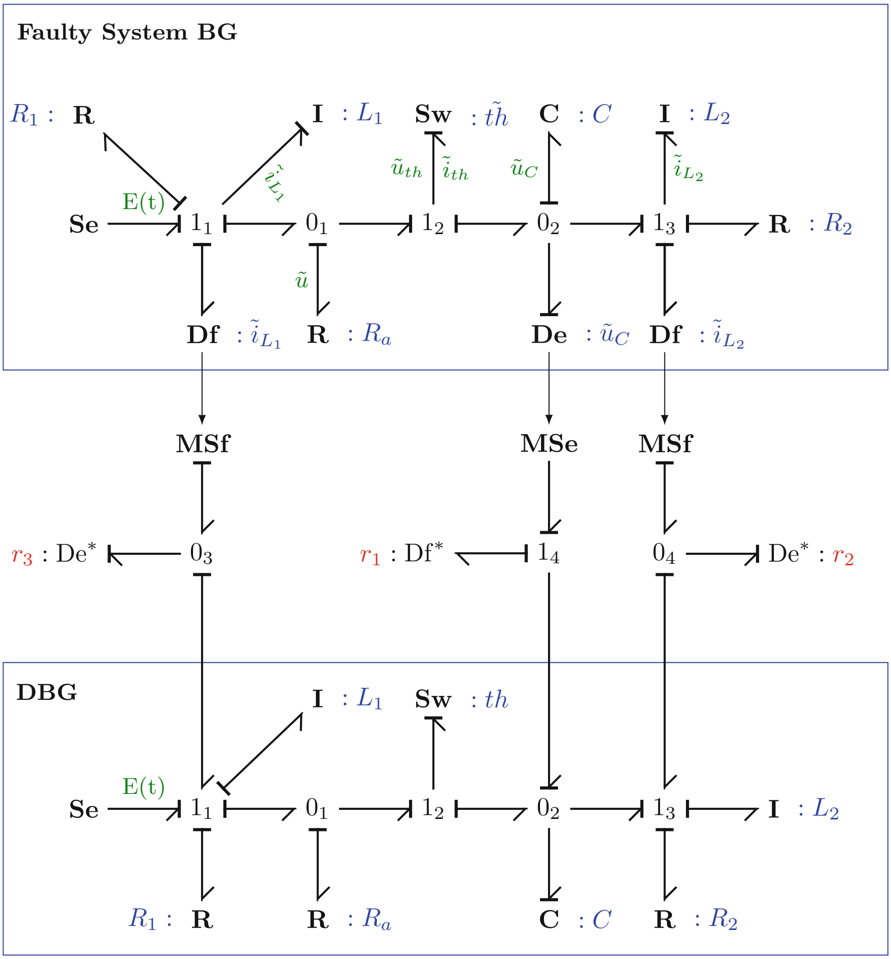


Fig. 5.11 Coupling of a faulty system BG and a diagnostic BG for the half-wave rectifier in Fig. 5.2

Finally, Fig. 5.14 shows a Modelica listing of the bond graph of the thyristor controlled faulty rectifier circuit.

In a system with switching devices such as power electronic circuits, the detection of faults changing the circuit topology are of particular interest. The effect of parametric faults can also be easily studied in an off-line simulation by simply changing parameter values in the behavioural model of the system. Moreover, as ARR residuals should be sensitive to true faults but insensitive to admissible parameter variation, the possible effect of parameter uncertainties on fault alarm thresholds can be studied. For a fault detection robust with regard to parameter variations, bond graph elements can be split into a part with nominal parameters

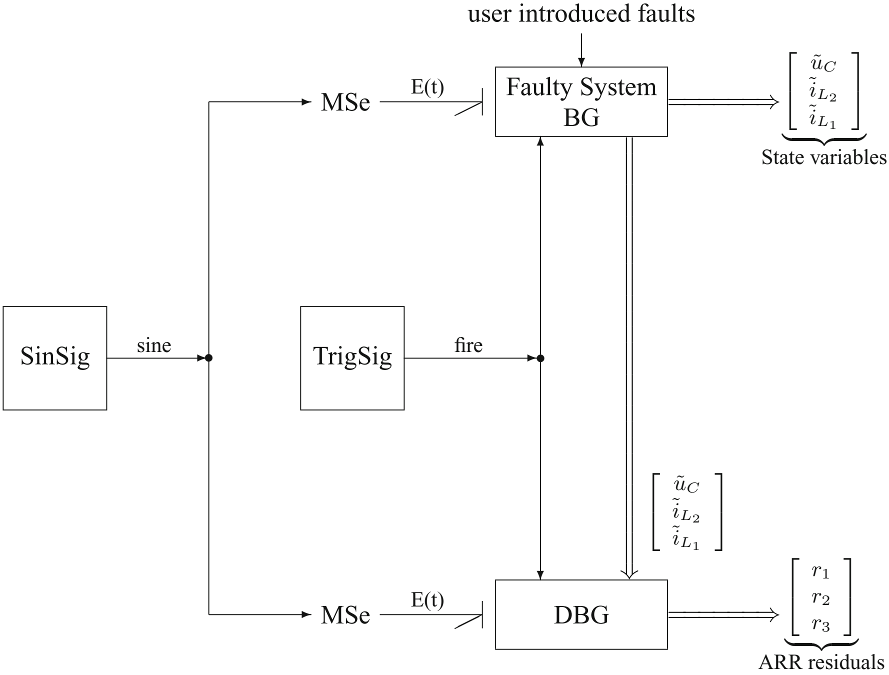

user introduced faults
Faulty System BG
MSe
E(t)
$\begin{bmatrix}\tilde{u}_C\\ \tilde{i}_{L_2}\\ \tilde{i}_{L_1}\end{bmatrix}$
State variables
SinSig
sine
TrigSig
fire
$\begin{bmatrix}\tilde{u}_C\\ \tilde{i}_{L_2}\\ \tilde{i}_{L_1}\end{bmatrix}$
MSe
E(t)
DBG
$\begin{bmatrix}r_1\\ r_2\\ r_3\end{bmatrix}$
ARR residuals

Fig. 5.12 Block diagram of the coupling of the bond graph models in Fig. 5.11

and another part that accounts for possible parameter uncertainties [24, 28], or the latter ones can be represented separately in an incremental bond graph [12, Chap. 4]. Furthermore, bounded measurement uncertainties assumed to be additive can be taken into account by adding modulated sources to detector outputs (cf. Fig. 5.16, see also [12, 68]).

5.3.2 ARR-Based Isolation and Estimation of Parametric Faults

In the example circuit considered in the previous section, a single structural fault has been introduced by assuming that there is a short-circuit in the thyristor as of some time instant t_e. As a result, time evolutions of residuals r_1, r_2 distinctly differ from values close to zero for $t > t_e$. As this fault signature is not unique, it is not clear that its cause is the short-circuit fault in the thyristor. Parametric faults can only be isolated by inspection of a structural FSM if a sufficient number of sensors is available and component fault signatures are unique. In the case of hybrid models, it must be taken into account that ARRs are mode-dependent as they include discrete

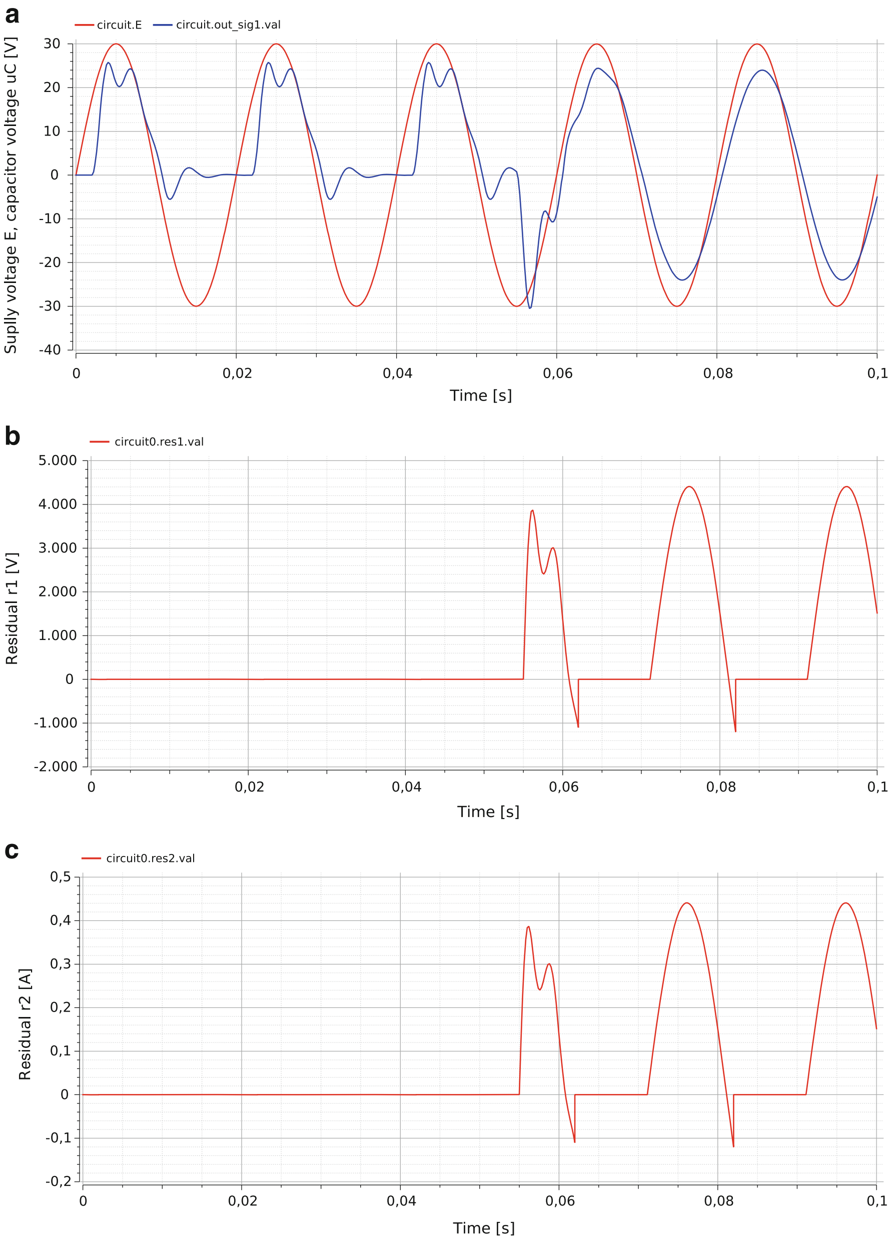


Fig. 5.13 Time evolution of voltages E and u_C and of residuals r_1, r_2, r_3 in the case the switch is short-circuited as of $t_e = 0.055$ s. (**a**) Supply voltage $E(t)$, capacitor voltage $u_C(t)$. (**b**) Residual $r_1(t)$. (**c**) Residual $r_2(t)$. (**d**) Residual $r_3(t)$

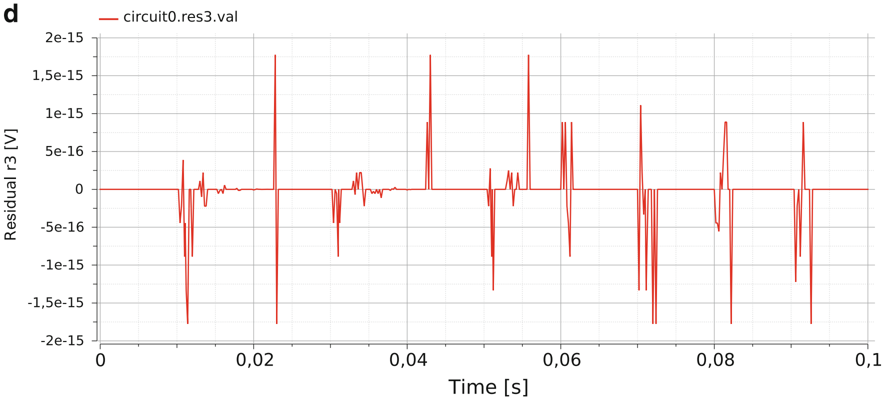


Fig. 5.13 (continued)

switch states. That is, ARRs must use current discrete switch state values in order to avoid that the evaluation of ARRs gives false residuals. That is, when monitoring a real system, the challenge is to identify the current system mode and to detect and isolate possible faults in real-time [5, 41]. In other words, when a change into another valid system mode or a discrete mode fault such as a persistent open circuit or short circuit fault in a transistor has been identified, discrete states in the ARRs have to be updated accordingly for detection and isolation of subsequent faults.

Furthermore, in the case of multiple faults one fault may diminish or even compensate the effect of another fault. If it can be assumed that fault events are uncorrelated, the probability that multiple faults simultaneously occur is small. That is, it may be assumed that there are at least small time delays between multiple fault events, but their effects may superimpose each other and one parametric fault that has happened in the current system mode may not be detectable as long as discrete switch states disable the parameter in the ARRs.

Once a parametric fault has occurred, it depends on its severity whether a nominal feedback controller can compensate the effects of that fault or whether fault accommodation is required. In any case, the system is no longer the original one. In case a progressive parametric fault has been detected, isolated and estimated, FDI of further future faults can be facilitated by replacing the nominal value of that parameter by the estimated one in the DBG model, i.e. the faulty system is treated as a new normally operating system. However, when a progressive parametric fault has started, some degradation of the parameter takes place with time. For the detection and isolation of further future faults this means that the faulty parameter, $\Theta(t)$, has to be estimated again from time to time and to be updated in the DBG model. Moreover, a progressive fault may eventually turn into a component failure. This requires to predict the so-called remaining useful life (RUL) of the faulty component in order to schedule maintenance measures. Clearly, correct FDI is a prerequisite for fault prognosis (see Chaps. 6 and 7 in this volume and e.g. [75]).

```
block FaultySystemBG4b
  //
  extends parameters;
  input Signal in_sig1;
  input BooleanSignal in_sig2;
  output Signal out_sig1, out_sig2, out_sig3;
  Real E;
  // Thyristor
  Real uTh, iTh, s;
  Boolean off(start = true);
  Boolean fire(start = true);
  Real te = 0.055;
  //
  Real uC(start = 0.0);
  Real iL1(start = 0.0);
  Real uL1, u;
  Real iL2(start = 0.0);
equation
  E = in_sig1.val;
  fire = in_sig2.val;
  // 1_1 junction:
  R1 * iL1 = E - uL1 - u;
  L1 * der(iL1) = uL1;
  // 0_1 junction:
  u = Ra * (iL1 - iTh);
  // 1_2 junction:
  // Thyristor:
  off = s < 0 or pre(off) and not fire;
  uTh = s * (if off and time < te then 1 else 0);
  iTh = s * (if off then Gon else 1);
  u = uTh + uC;
  // 0_2 junction:
  C * der(uC) = iTh - iL2;
  // 1_3 junction:
  L2 * der(iL2) = uC - R2 * iL2;
  //
  out_sig1.val = uC;
  out_sig2.val = iL2;
  out_sig3.val = iL1;
  //
end FaultySystemBG4b;
```

Fig. 5.14 Modelica listing of the bond graph of the rectifier circuit with a short-circuited thyristor as of $t_e = 0.055$

Given that a single fault hypothesis can be assumed, one way to isolate parametric faults in real-time simulation is to use *multiple* (diagnostic) bond graph models [34]. In each of them, the nominal value of one single parameter is replaced by an estimated value obtained from known inputs and measured outputs of the real world system while for all other parameters their nominal values are retained. If the estimated parameter used in one of the diagnostic bond graph models matches the unknown faulty parameter in the real system, then the latter one and its bond graph model should produce outputs and their difference should be close to zero. In other words, ARR residuals as outputs of the DBG should be close to zero. A drawback of this approach is that the more complex a system is the more models are needed although the latter ones can be evaluated in parallel on a multiprocessor system.

In the case of multiple parametric faults, parameter estimation can be used for fault isolation. A comparison of estimated parameter values with their nominal ones isolates the faulty parameters [13, 63, 75]. However, if discrete switch states are considered as parameters then parameter estimation would result in meaningless real values for the discrete switch states. Therefore, in [3], it is proposed to perform a parameter estimation for each feasible combination of switch state combinations. The switch combination that results in the lowest of all optimal values for the cost function then identifies the actual current system mode and the minimisation of the functional provides estimated values for the continuous parameters.

Estimation of all parameters can also be used to isolate a single switch state fault. If all estimated parameter values match their nominal ones then it must be the switch state that is faulty.

In the following, it is shown that ARRs that have been derived offline from a DBG and that are evaluated in real-time can not only be used to detect parametric faults but also be used for estimation of their magnitude in some simple cases.

In the first case it is assumed that

- a single fault hypothesis can be adopted,
- the fault can be mapped onto a faulty parameter,
- the parameter has a unique structural fault signature in the FSM and thus can be isolated.

As an example, consider the fragment of a DBG in Fig. 5.15 and assume that the resistor R : R_n in the real system has become faulty. As of some time instant, its parameter value is $\tilde{R} = R_n + \Delta R$. Accordingly, an effort $\tilde{e}(t)$ is measured and is input into the DBG.

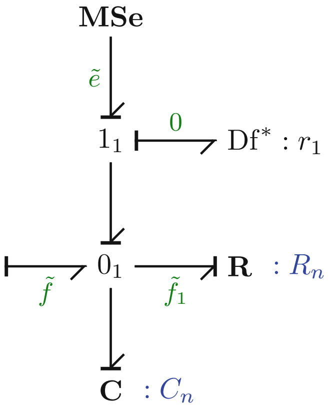


Fig. 5.15 Part of a diagnostic bond graph with the measured input $\tilde{e}$

Reformulation of the sum of flows at the junction 0_1 yields

$$0 \neq r_1 = R_n\tilde{f} - R_n C_n \dot{\tilde{e}} - \tilde{e} \tag{5.47}$$

The sum equals zero if the nominal resistance R_n is replaced by the single faulty value $\tilde{R} = R_n + \varDelta R$.

$$\begin{aligned} 0 &= \tilde{R}\tilde{f} - \tilde{R}C_n\dot{\tilde{e}} - \tilde{e} \\ &= \underbrace{R_n\tilde{f} - R_n C_n\dot{\tilde{e}}}_{r_1} + \underbrace{(\tilde{f} - C_n\dot{\tilde{e}})}_{\tilde{f}_1} \varDelta R \end{aligned} \tag{5.48}$$

Thus,

$$\varDelta R = -\frac{r_1}{\tilde{f}_1} \tag{5.49}$$

As a result, an estimate of the size of a single isolatable parametric fault can be obtained by means of the residual of an ARR to which the faulty parameter contributes.

In the second case it is assumed that the single fault hypothesis still holds. However, two components with parameters $\varTheta_1$ and $\varTheta_2$ respectively have got the same structural component fault signature in the FSM. That is, the parametric fault cannot be isolated just by inspecting the structural FSM. Let

$$0 = g_i(\tilde{\varTheta}_1, \tilde{\varTheta}_2, \tilde{\boldsymbol{e}}, \tilde{\boldsymbol{f}}), \quad i = 1, 2 \tag{5.50}$$

be the sums of power variables at junctions in the BG of the faulty system to which parameters $\tilde{\varTheta}_1, \tilde{\varTheta}_2$ contribute. Then, for $i = 1, 2$

$$0 = \underbrace{g_i(\varTheta_1^n, \varTheta_2^n, \tilde{\boldsymbol{e}}, \tilde{\boldsymbol{f}})}_{=:\, r_i \neq 0} + \left.\frac{\partial g_i}{\partial \varTheta_1}\right|_n \varDelta\varTheta_1 + \left.\frac{\partial g_i}{\partial \varTheta_2}\right|_n \varDelta\varTheta_2 \tag{5.51}$$

The two linear equations (5.51) can be solved for $\varDelta\varTheta_1$, $\varDelta\varTheta_1$. In the case that $\varTheta_1$ is faulty but $\varTheta_2$ is not, then $\varDelta\varTheta_1 \neq 0 \;\wedge\; \varDelta\varTheta_2 = 0$ is equivalent to the two conditions

$$-\varDelta\varTheta_1 = \underbrace{r_1 \left(\left.\frac{\partial g_1}{\partial \varTheta_1}\right|_n \right)^{-1}}_{\varPsi_1(r_1)} = \underbrace{r_2 \left(\left.\frac{\partial g_2}{\partial \varTheta_1}\right|_n \right)^{-1}}_{\varPsi_2(r_2)} \tag{5.52}$$

$$\underbrace{r_1 \left(\left.\frac{\partial g_1}{\partial \varTheta_2}\right|_n \right)^{-1}}_{\varPsi_3(r_1)} \neq \underbrace{r_2 \left(\left.\frac{\partial g_2}{\partial \varTheta_2}\right|_n \right)^{-1}}_{\varPsi_4(r_2)} \tag{5.53}$$

That is, if the above two conditions hold, the absolute fault $\varDelta\varTheta_1$ can be isolated and its size can be estimated. Functions $\varPsi_j()$, $j = 1, \ldots, 4$, can be obtained from the DBG by setting up two ARRs and by considering one parameter constant while the other one deviates from its nominal value. The other case, i.e. $\varTheta_2$ is faulty but $\varTheta_1$ is not, can be checked likewise.

5.3.3 ARR-Based Isolation and Estimation of Additive Sensor and Actuator Faults

Clearly, even if a plant to be controlled is in a healthy state, sensor and actuator faults may degrade the closed-loop control and thus the performance of the system. Given a single fault hypothesis, sensor and actuator faults assumed to be additive can be isolated and estimated by means of an ARR-based approach. They can be modelled by additional sources. The magnitude of these faults can be expressed by means of an ARR residual.

As an example, consider the effort sensor in Fig. 5.16 delivering a faulty effort $\tilde{e}$ into the diagnostic bond graph. The sensor fault taken into account by the modulated effort source, MSe, can be obtained directly from the ARR established for junction 0_1.

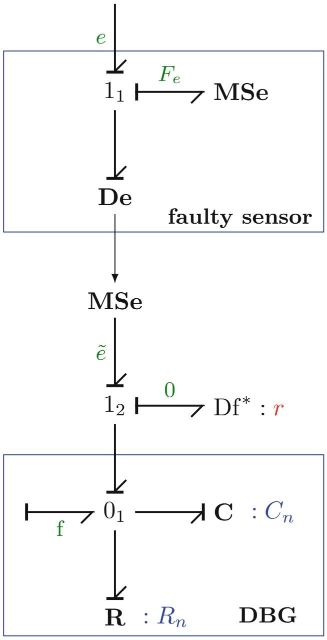


Fig. 5.16 An effort sensor providing a faulty effort $\tilde{e}$ into the diagnostic bond graph

In case the effort sensor is not faulty, the sum of flows at junction 0_1 equals zero.

$$0 = -f + C_n \dot{e} + \frac{e}{R_n} \tag{5.54}$$

If the sensor effort e is replaced by the faulty reading $\tilde{e} = e + F_e$, where F_e denotes the sensor fault, the sum of flows is no longer equal to zero but results in a residual $r \neq 0$.

$$\begin{aligned} 0 \neq r &= -f + C_n \dot{\tilde{e}} + \frac{\tilde{e}}{R_n} \\ &= -f + C_n(\dot{e} + \dot{F}_e) + \frac{1}{R_n}(e + F_e) \\ &= \underbrace{-f + C_n \dot{e} + \frac{e}{R_n}}_{=0} + C_n \dot{F}_e + \frac{F_e}{R_n} \end{aligned} \tag{5.55}$$

Hence, the sensor fault is given by the ordinary differential equation (ODE)

$$\dot{F}_e + \frac{1}{R_n C_n} F_e = \frac{1}{C_n} r \tag{5.56}$$

5.3.4 Observer-Based Isolation and Estimation of Faults

Instead of an ARR-based approach also Unknown Input Observers (UIO) can be used for fault isolation and estimation. In [13, Chap. 7], Ghoshal and Samantaray consider a number of possible fault candidates in a two-tank example system represented by a hybrid model and develop an UIO for each faulty system model. For an UIO, the state estimation error approaches zero asymptotically regardless of the presence of any disturbances. Therefore, the UIO from a bank of observers with an output residual close to zero isolates the fault.

If there are additive unknown disturbances on the system in addition to parametric faults in the plant, they can be decoupled from the state estimation error so that the output residual only depends on the faults [36]. This is briefly recalled in the following. Consider a healthy system that is subject to additive unknown disturbances $\boldsymbol{d}(t)$ described by the following LTI state space model

$$\dot{\boldsymbol{x}}(t) = \mathbf{A}\boldsymbol{x}(t) + \mathbf{B}\boldsymbol{u}(t) + \mathbf{E}\boldsymbol{d}(t) \tag{5.57}$$

$$\boldsymbol{y}(t) = \mathbf{C}\boldsymbol{x}(t) \tag{5.58}$$

with known constant coefficient matrices of appropriate dimensions. Without loss of generality, matrix $\mathbf{E}$ is assumed to have full column rank. A full order UIO is then given by the equations

$$\dot{z}(t) = \mathbf{F}z(t) + \mathbf{TB}\boldsymbol{u}(t) + \mathbf{K}\boldsymbol{y}(t) \tag{5.59}$$

$$\hat{\boldsymbol{x}}(t) = z(t) + \mathbf{H}\boldsymbol{y}(t) \tag{5.60}$$

where z denotes the state vector of the UIO and $\hat{\boldsymbol{x}}$ the estimate of $\boldsymbol{x}$. If the matrices in (5.59)–(5.60) fulfill the conditions

$$\mathbf{K} = \mathbf{K}_1 + \mathbf{K}_2 \tag{5.61a}$$

$$\mathbf{0} = (\mathbf{HC} - \mathbf{I})\mathbf{E} \tag{5.61b}$$

$$\mathbf{T} = \mathbf{I} - \mathbf{HC} \tag{5.61c}$$

$$\mathbf{F} = \mathbf{TA} - \mathbf{K}_1\mathbf{C} \tag{5.61d}$$

$$\mathbf{K}_2 = \mathbf{FH} \tag{5.61e}$$

then the disturbances $\boldsymbol{d}(t)$ can be decoupled and $\boldsymbol{e}(t) := \boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t) \rightarrow \mathbf{0}$ for $t \rightarrow \infty$ if all eigenvalues of $\mathbf{F}$ are stable.

Now consider a system with parametric plant faults that is subject to unknown inputs.

$$\dot{\boldsymbol{x}}(t) = (\mathbf{A}_n + \mathit{\Delta}\mathbf{A})\boldsymbol{x}(t) + \mathbf{B}\boldsymbol{u}(t) + \mathbf{E}\boldsymbol{d}(t) \tag{5.62}$$

$$\boldsymbol{y}(t) = \mathbf{C}\boldsymbol{x}(t) \tag{5.63}$$

It is assumed that the parametric faults only affect the system matrix $\mathbf{A}$ which is decomposed into a matrix $\mathbf{A}_n$ with coefficients being functions of the nominal parameters and a matrix $\mathit{\Delta}\mathbf{A}$ accounting for the parametric faults.

Assume that the UIO (5.59)–(5.60) exists for the healthy system (5.57)–(5.58). If the UIO is used for the faulty system (5.62)–(5.63), then one obtains for the state estimation error

$$\begin{aligned}\dot{\boldsymbol{e}}(t) &= \dot{\boldsymbol{x}}(t) - \dot{\hat{\boldsymbol{x}}}(t) \\ &= \mathbf{F}\boldsymbol{e}(t) + (\mathbf{I} - \mathbf{HC})(\mathit{\Delta}\mathbf{A})\boldsymbol{x}(t)\end{aligned} \tag{5.64}$$

That is, the state estimation error has been decoupled from the disturbance $\mathbf{d}(t)$ but the error $\mathbf{e}(t)$ and thus the output residual $\mathbf{r}(t) := \mathbf{y}(t) - \mathbf{C}\hat{\boldsymbol{x}}(t) = \mathbf{y}(t) - \hat{\boldsymbol{y}}(t) = \mathbf{C}\boldsymbol{e}(t)$ does not approach to zero for $t \rightarrow \infty$ due to the parametric faults taken into account by the matrix $\mathit{\Delta}\mathbf{A}$. As has been pointed out in [36], the existence of an UIO requires that the number of measured outputs, i.e. the number of sensors exceeds the number of disturbances.

In [21], Chen et al. use an UIO to decouple unknown disturbances from the output residual for an LTI system with possible additive sensor and actuator faults.

5.4 Fault Accommodation Based on Inverse Simulation

Fault accommodation aims at an automatic controller reconfiguration in order to prevent that a severe fault that has been detected by monitoring and fault diagnosis leads to a system or a component failure and to make sure that the dynamic behaviour of the closed-loop system remains close to a required reference dynamic behaviour despite of a persistent fault. Fault accommodation is an active Fault Tolerant Control (FTC) technique that changes the controller law after a fault has occurred and relies on a faultless operation of the sensors and the actuators [10, Appendix 5], [9, 64]. That is, a system that has become faulty as of some time instant is not repaired immediately and remains in operation due to a modified control signal. The control objective is still achieved, the performance of the faulty system may, however, show some degradation. The conceptual fault accommodation scheme in Fig. 5.17 displays the interplay of fault diagnosis and controller redesign.

If some sensors fail, the system becomes partially unobservable and information needed by the controller gets lost. On the other hand, some actuator faults may make the system partially uncontrollable. As a result, a real-time re-adjustment of the controller law linking the same controller inputs and outputs in use before the fault event is not possible any more. Some sensor information into the controller as well as some actuators may not be available any more, that is, some control loops are broken so that both a change of the control-loop structure and a new control law are required that account for the changed configuration. This more general active FTC strategy is termed *control reconfiguration* in the literature. Reconfigurable FTC requires a fast fault diagnosis and that the system remains observable and controllable with the sensors and actuators still in operation. Nevertheless, it always takes some time to detect and to isolate a fault. Accordingly, there is a time span,

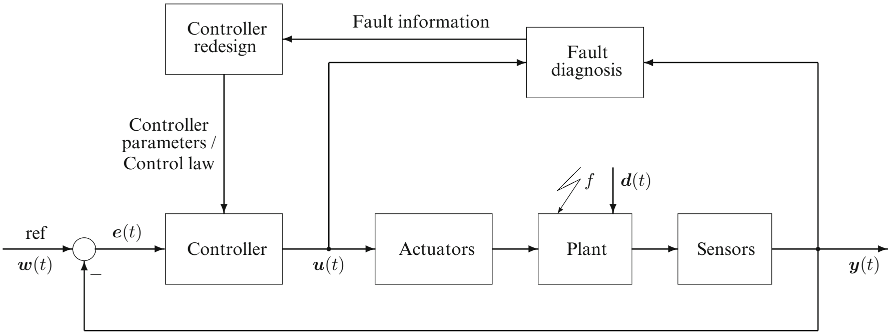


Fig. 5.17 Conceptual fault accommodation scheme [13, 45]

the time to reconfiguration, in which the system behaviour is faulty but the input required to compensate the fault hasn't been reconstructed yet. Depending on a fault, its severity and its location, it may happen that the control objective can be achieved neither by fault accommodation nor by control reconfiguration. What remains in such a case is to change the control objective. A discussion of reconfigurable FTC techniques may be found in [44, 45]. In this chapter it is assumed that sensors and actuators do not fail. Faults may only occur in the plant (Fig. 5.17).

5.4.1 Input Reconstruction Using Identified Faulty Parameters

Assume that an abnormality in the dynamic behaviour of a multiple input, multiple output (MIMO) system has been observed at time instant t_1 and that it is due to a single parametric fault. Once the fault has been isolated and its magnitude has been estimated, it can be accommodated by designing an input into the faulty system so that the fault is compensated and the system produces an output close to a desired output behaviour despite the fault. In other words, the considered fault recovery is based on a preceding successful fault diagnosis.

For closed loop systems with a nominal feedback controller, FDI (robust with regard to uncertain parameters) is not an easy task as its objectives are opposite to those of the controller so that there is a trade-off between good closed loop performance and good fault detection. The problem of FDI in closed loop systems has been addressed for instance in [6, 54].

Small multiplicative faults in the plant and additive faults in its actuators as well as small disturbances on the system are compensated by the feedback controller and will thus not be detected by monitoring the system output and the control error as long as the latter one is forced to values close to zero with time. Slightly faulty monitored values of a controlled variable due to a small sensor offset are also not detected in case there are no redundant sensors so that a voting among multiple sensors of the same signal is not possible. The feedback controller will make a controlled variable provided by a faulty sensor equal to the reference.

Passive fault tolerant control uses a controller that has been designed so that it ensures stability and the control objectives in the presence of a number of anticipated faults from a certain class. Large faults either in the plant or in its sensors and actuators may exceed the robustness of the feedback controller and can lead to a degradation of the normal dynamic closed-loop behaviour and can impair safety or reliability so that an active FTC is required for a fast recovery from a severe fault and for maintaining stability and accuracy. Once a large fault has occurred, the nominal closed-loop controller aims at compensating the fault which takes some time. During this time, a fault indicator will exceed a fault threshold so that there is time to detect the fault. However, it is important that fault detection, isolation and input reconstruction are fast. If there are disturbances on the system in addition to parametric faults in the plant, an unknown input observer (UIO) can be used to decouple them from the faults so that the observer generated residual only depends on the faults as has been outlined in Sect. 5.3.4.

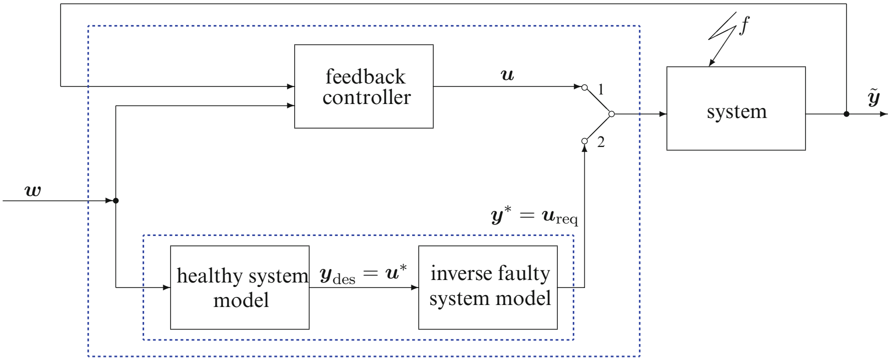


Fig. 5.18 Fault accommodation using an inverse model

Now, suppose that a large fault has happened, that its effect on the dynamic behaviour of the closed-loop system has been detected and that the fault has been isolated. The input required for its accommodation then can be determined by means of an inverse model that uses the identified faulty parameter and the desired system output as input. The block diagram in Fig. 5.18 displays the scheme of a fault accommodation using an inverse model.

As long as there is no fault, switch 2 is open and the output of the real system $\tilde{\boldsymbol{y}}$ equals the output $\boldsymbol{y}_{\mathrm{des}}(t)$ of the healthy system model $\tilde{\boldsymbol{y}}(t) = \boldsymbol{y}_{\mathrm{des}}(t)$ which serves as a reference. (The output of the inverse model equals the command variable.) When an abrupt fault, f, has happened in the system at time instant t_1, it takes some time Δt to detect and to isolate it. Meanwhile, the dynamic output behaviour of the faulty system, $\tilde{\boldsymbol{y}}(t)$, deviates increasingly from the desired output behaviour $\boldsymbol{y}_{\mathrm{des}}(t)$ and the nominal feedback controller aims at compensating the error. Once the fault has been detected and isolated, the faulty parameter, $\tilde{p}$, is fed into the inverse system model so that the latter one becomes an inverse faulty system model. With the information about the fault and the output of the healthy system model as input, $\boldsymbol{y}_{\mathrm{des}} = \boldsymbol{u}^*$, the inverse faulty system model provides an output signal $\boldsymbol{y}^* = \boldsymbol{u}_{\mathrm{req}}$. By closing switch 2, this signal is added to the output of the nominal feedback controller at time instant $t_2 > t_1$ (cf. [55]). At this time instant, the error has already been somewhat reduced by the nominal controller. The sum of both inputs into the faulty system drives the error to zero and forces the faulty system to behave as the healthy system despite the fault. The healthy system model and the inverse faulty system model constitute an additional feedforward controller which becomes active in the case of a fault and provides $\boldsymbol{u}_{\mathrm{req}}$ as an input into the faulty system. The fault compensated system then can be treated as a normally operating system. In case there is a small disturbance on the faulty system after the fault has been accommodated, it is compensated by the feedback controller.

There are various approaches to system inversion. The input to be reconstructed can be obtained by

- designing a feedback system (proper inversion) [16, 49],
- numerical solution of a DAE system [66],
- application of bicausality to a bond graph [33, 42, 51]

A review of inverse simulation methods may be found in [48, Chap. 4]. All of them have their pros and cons.

This chapter presents an approach to fault accommodation based on the numerical solution of a hybrid DAE system.

5.4.1.1 Fault Accommodation in Open-Loop Systems

In the following, first, fault accommodation in open-loop is considered. Assume that the dynamic behaviour of a healthy system may be captured by a mode switching linear time-invariant multiple input, multiple output (MIMO) forward model derived from a BG.

$$\dot{\boldsymbol{x}}(t) = \mathbf{A}\boldsymbol{x}(t) + \mathbf{B}\boldsymbol{u}(t) \tag{5.65a}$$

$$\boldsymbol{y}(t) = \mathbf{C}\boldsymbol{x}(t) \tag{5.65b}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ denotes the vector of state variables, $\boldsymbol{u}(t) \in \mathbb{R}^m$ the known input vector and $\boldsymbol{y}(t) \in \mathbb{R}^p$ the output vector. The matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are of appropriate dimensions. Their coefficients are constant for the time intervals between two discrete events, i.e. for the duration of a system mode. It is assumed that the pencil $(s\mathbf{I} - \mathbf{A})$ is regular, i.e. that the determinant $\det(s\mathbf{I} - \mathbf{A})$ is not identical zero for $s \in \mathbb{C}$.

Suppose that there are no disturbances on the system and that a single parametric fault that has occurred at t_1 has been isolated and identified as of $t_2 > t_1$. Then for $t > t_2$ an input $\boldsymbol{u}_{\text{req}}(t)$ to be determined is required that forces the faulty system to produce the output of the healthy system as the desired output despite the presence of the fault, i.e. $\boldsymbol{y}_{\text{des}}(t) = \boldsymbol{y}(t)$. The equations of the faulty system then read

$$\dot{\tilde{\boldsymbol{x}}}(t) = \tilde{\mathbf{A}}\tilde{\boldsymbol{x}}(t) + \tilde{\mathbf{B}}\boldsymbol{u}_{\text{req}}(t) \tag{5.66a}$$

$$\boldsymbol{y}_{\text{des}}(t) = \tilde{\mathbf{C}}\tilde{\boldsymbol{x}}(t) \tag{5.66b}$$

or

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\tilde{\boldsymbol{x}}} \\ \dot{\boldsymbol{u}}_{\text{req}} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{A}} & \tilde{\mathbf{B}} \\ \tilde{\mathbf{C}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{x}} \\ \tilde{\boldsymbol{u}}_{\text{req}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ -\boldsymbol{y}_{\text{des}} \end{bmatrix} \tag{5.67}$$

where $\tilde{\boldsymbol{x}}(t)$ denotes the unknown state of the faulty system and $\boldsymbol{u}_{\text{req}}(t)$ the input to be determined. As the faulty parameter may affect the coefficients of all matrices, the latter ones are distinguished from the ones of the healthy system by a tilde.

The forward model (5.65) is said to be invertible if the determinant of the transfer matrix $\mathbf{T}(s) := \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$ does not vanish identical for $s \in \mathbb{C}$.[4] Then an inverse of (5.65) is

$$\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\dot{z} = \begin{bmatrix}\mathbf{A}-\mathbf{BC} & \mathbf{B}\\ -\mathbf{C} & \mathbf{0}\end{bmatrix} z + \begin{bmatrix}\mathbf{B}\\ \mathbf{I}\end{bmatrix}\boldsymbol{y} \tag{5.68a}$$

$$\boldsymbol{u} = \begin{bmatrix}-\mathbf{C} & \mathbf{I}\end{bmatrix} z + \boldsymbol{y} \tag{5.68b}$$

where $z^T = [\boldsymbol{x}^T\ \boldsymbol{u}^T]^T$ [65]. Tan and Vandewalle show in [65] that the solvability of the DAE of the inverse is equivalent to the invertibility of the forward model and that the transfer matrix $\hat{\mathbf{T}}(s)$ of system (5.68a)–(5.68b) is $\hat{\mathbf{T}}(s) = \mathbf{T}^{-1}(s)$.

Differentiation of the algebraic constraint (5.66b) with respect to time and substitution of (5.66a) into the result gives the DAE system

$$\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\dot{\tilde{\boldsymbol{x}}}\\ \dot{\boldsymbol{u}}_{\mathrm{req}}\end{bmatrix} = \begin{bmatrix}\tilde{\mathbf{A}} & \tilde{\mathbf{B}}\\ \tilde{\mathbf{C}}\tilde{\mathbf{A}} & \tilde{\mathbf{C}}\tilde{\mathbf{B}}\end{bmatrix}\begin{bmatrix}\tilde{\boldsymbol{x}}\\ \boldsymbol{u}_{\mathrm{req}}\end{bmatrix} + \begin{bmatrix}\mathbf{0}\\ -\dot{\boldsymbol{y}}\end{bmatrix} \tag{5.69}$$

DAE system (5.69) is of index 1 if $(\tilde{\mathbf{C}}\tilde{\mathbf{B}})^{-1}$ exists and can be numerically computed by a BDF-based solver. In that case, the algebraic constraint in (5.69) can be solved for $\boldsymbol{u}_{\mathrm{req}}$ and substituted into the ODE for $\tilde{\boldsymbol{x}}$.

$$\dot{\tilde{\boldsymbol{x}}}(t) + \tilde{\mathbf{M}}\tilde{\mathbf{A}}\tilde{\boldsymbol{x}}(t) = \tilde{\mathbf{B}}(\tilde{\mathbf{C}}\tilde{\mathbf{B}})^{-1}\dot{\boldsymbol{y}}(t) \tag{5.70}$$

where $\tilde{\mathbf{M}} := \tilde{\mathbf{B}}(\tilde{\mathbf{C}}\tilde{\mathbf{B}})^{-1}\tilde{\mathbf{C}} - \mathbf{I}$.

With the solution $\tilde{\boldsymbol{x}}(t)$ of (5.70) the required input $\boldsymbol{u}_{\mathrm{req}}(t)$ reads

$$\boldsymbol{u}_{\mathrm{req}}(t) = (\tilde{\mathbf{C}}\tilde{\mathbf{A}})^{-1}(\dot{\boldsymbol{y}}(t) - \tilde{\mathbf{C}}\tilde{\mathbf{A}}\tilde{\boldsymbol{x}}(t)) \tag{5.71}$$

Given a specific mode switching forward model (5.65) with constant coefficients for a time interval between two consecutive discrete events, the question is how often which equations of inverse DAE system (5.67) determining $\boldsymbol{u}_{\mathrm{req}}(t)$ need to be differentiated so that (5.67) can be transformed into an explicit ODE for that time interval. Pantelides algorithm initially developed for the determination of a set of

[4] For systems with a regular pencil this is equivalent to the requirement that the determinant of the Rosenbrock matrix

$$\mathbf{P}(s) := \left[\begin{array}{c|c} s\mathbf{I} - \mathbf{A} & -\mathbf{B}\\ \hline \mathbf{C} & \mathbf{0}\end{array}\right]$$

does not vanish identically for $s \in \mathbb{C}$ as

$$\det\mathbf{P}(s) \;=\; \det(s\mathbf{I} - \mathbf{A})\det[\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}] \;=\; \det(s\mathbf{I} - \mathbf{A})\det\mathbf{T}(s).$$

consistent initial conditions can be used to identify a minimal set of equations to be differentiated.[5]

It may happen that some states of the faulty system can be expressed as a function of the remaining states, the input $\boldsymbol{u}_{\text{req}}$ to be determined, the desired output $\boldsymbol{y}_{\text{des}}$ and its time derivatives. In the following example, all unknown states of the faulty system can be eliminated. As a result, the required input $\boldsymbol{u}_{\text{req}}$ can be expressed by the desired output $\boldsymbol{y}_{\text{des}}$ and its derivatives.

Example: Buck Converter Driven DC Motor

Figure 5.19 displays a circuit schematic of a buck-converter operating in continuous conduction mode (CCM) and driving a DC motor. The transistor Q and the diode have been modelled as a non-ideal switch with an ON-resistance R_{on}, i.e. by a series connection of an MFT controlled by a Boolean variable and a resistor (cf. Fig. 5.3). As the transistor Q and the diode D switch oppositely, there are only two system modes which can be captured by a single MTF if it is assumed that the ON resistance is the same for the transistor and the diode.

Figure 5.20 shows a simplified forward BG model where R_L denotes the resistance of the inductor, R_a and L_a the resistance and the inductance of the motor's armature winding, b a friction parameter and τ_{load} an external load torque, $m := 1 - d$, where d denotes the duty cycle of the signal $u(t)$ controlling the transistor Q. Variables in this BG model have been averaged over the switching time period of the signal $m(t)$.

[5] $(\tilde{\mathbf{C}}\tilde{\mathbf{B}})$ being non-singular means that the matrix is square, i.e. the number of inputs equals the number of outputs, $m = p$, and $(\tilde{\mathbf{C}}\tilde{\mathbf{B}}) \neq \mathbf{0}$. $(\tilde{\mathbf{C}}\tilde{\mathbf{B}}) \neq \mathbf{0}$ means that a step change in $\mathbf{u}(t)$ causes an immediate response in the derivative of $\mathbf{y}(t)$. That is, the transfer functions in the transfer function matrix relating $\mathbf{u}(t)$ and $\mathbf{y}(t)$ have got one more pole than zeros. Let $N(s)$ be the nominator polynomial and $\Delta(s)$ the denominator polynomial of a transfer function. Let deg denote the degree of a polynomial. Then, $\varrho := \deg(\Delta(s)) - \deg(N(s)) = 1$. (A transfer function is said to be *strictly proper* if the relative degree $\varrho \geq 1$.) Laplace-transform of the differentiated output Eq. (5.66b) and the state equations yields

$$\begin{aligned} s\mathbf{Y}(s) &= [(\tilde{\mathbf{C}}\tilde{\mathbf{B}})(s\mathbf{I} - \tilde{\mathbf{A}})^{-1}\tilde{\mathbf{B}} + (\tilde{\mathbf{C}}\tilde{\mathbf{B}})]\mathbf{U}(s) \\ &= [(\tilde{\mathbf{C}}\tilde{\mathbf{B}})\operatorname{adj}(s\mathbf{I} - \tilde{\mathbf{A}})\tilde{\mathbf{B}} + (\tilde{\mathbf{C}}\tilde{\mathbf{B}})\Delta(s)]\frac{1}{\Delta(s)}\mathbf{U}(s) = \underbrace{\frac{\mathbf{N}(s)}{\Delta(s)}}_{=:\,\mathbf{H}_1(s)}\mathbf{U}(s) \end{aligned} \tag{5.72}$$

where $\Delta(s) := \det(s\mathbf{I} - \tilde{\mathbf{A}})$ and $\mathbf{N}(s)$ is a $p \times m$ square matrix. Each entry in $\mathbf{H}_1(s)$ is a transfer function of relative degree zero. The transfer function matrix relating $\mathbf{Y}(s)$ and $\mathbf{U}(s)$, however, is of relative degree one.

Given a causal bond graph, the relative degree of a transfer function between an input u_i and an output y_i is equal to the length L_i of the shortest causal path between the detector of y_i and the source providing u_i. If preferred integral causality has been assigned to the bond graph, then L_i equals the number of storage elements in integral causality on that path [61].

Finally, matrix $(\tilde{\mathbf{C}}\tilde{\mathbf{B}})$ is related to sensor/actuator placement. If $\tilde{\mathbf{C}}\tilde{\mathbf{B}}$ turns out to be singular, practically speaking, it could be fixed by adding or moving sensors.

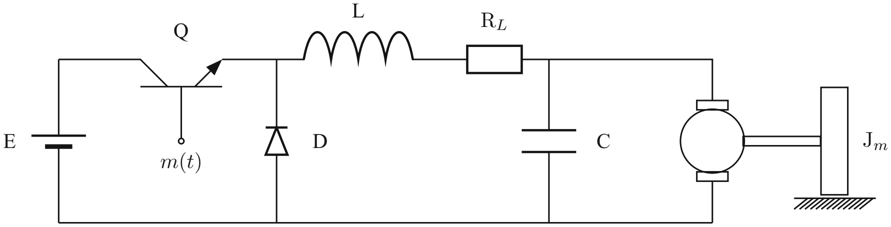


Fig. 5.19 Schematic of a buck-converter-driven DC motor

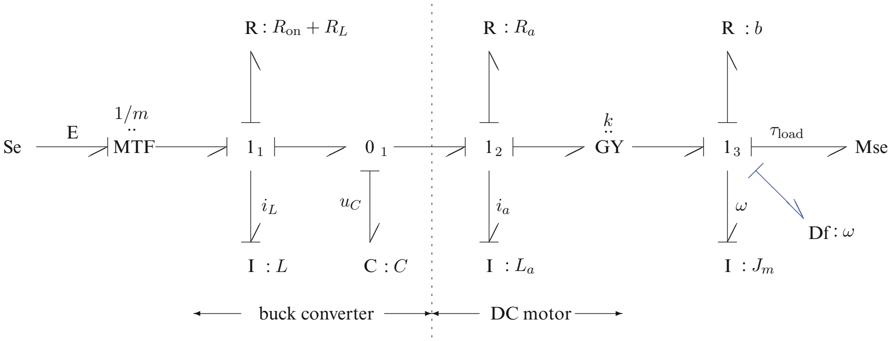


Fig. 5.20 Averaged forward BG model of the healthy buck-converter-driven DC motor

From the BG in Fig. 5.20, the following equations of the healthy system can be derived.

$$1_1 : 0 = mE - (R_{\text{on}} + R_L)i_L - L\frac{di_L}{dt} - u_C \tag{5.73}$$

$$0_1 : 0 = i_L - C\frac{du_C}{dt} - i_a \tag{5.74}$$

$$1_2 : 0 = u_C - R_a i_a - L_a\frac{di_a}{dt} - k\,\omega \tag{5.75}$$

$$1_3 : 0 = k\,i_a - b\,\omega - J_m\frac{d\omega}{dt} - \tau_{\text{load}} \tag{5.76}$$

$$y = \omega \tag{5.77}$$

Assume that there are no disturbances and let the identified parametric fault be the resistance of the motor inductance, $\tilde{R}_a := R_a + \Delta R_a$. In case the load torque is different from zero it must be measured or estimated. Accordingly, a load torque estimator would be required in a practical implementation. Furthermore, let $y_{\text{des}} = \omega_{\text{des}} = y = \omega$, $u := mE$, $u_{\text{req}} = \tilde{m}E = (1 - \tilde{d}(t))E$, $\tilde{x}_1 := \tilde{i}_L$, $\tilde{x}_2 := \tilde{u}_C$, $\tilde{x}_3 := \tilde{i}_a$, $\tilde{x}_4 := \tilde{\omega} = \omega$ and $R := R_{\text{on}} + R_L$. Then the DAE system of the inverse model determining u_{req} reads

$$0 = u_{\text{req}} - R\tilde{x}_1 - L\dot{\tilde{x}}_1 - \tilde{x}_2 \tag{5.78}$$

$$0 = \tilde{x}_1 - C\dot{\tilde{x}}_2 - \tilde{x}_3 \tag{5.79}$$

$$0 = \tilde{x}_2 - \tilde{R}_a\tilde{x}_3 - L_a\dot{\tilde{x}}_3 - k\,\tilde{x}_4 \tag{5.80}$$

$$0 = k\,\tilde{x}_3 - b\,\tilde{x}_4 - J_m\dot{\tilde{x}}_4 - \tau_{\text{load}} \tag{5.81}$$

$$y = \tilde{x}_4 \tag{5.82}$$

Substituting (5.82) into (5.81) gives

$$k\tilde{x}_3 = by + J_m\dot{y} + \tau_{\text{load}} \tag{5.83}$$

Comparison with (5.76) yields $\tilde{x}_3 = x_3$. Accordingly,

$$\begin{aligned} \tilde{x}_2 &= \tilde{R}_a x_3 + L_a\dot{x}_3 + ky \\ &= (R_a + \Delta R_a)x_3 + L_a\dot{x}_3 + ky \\ &= R_a x_3 + L_a\dot{x}_3 + ky + (\Delta R_a)x_3 \\ &= x_2 + (\Delta R_a)x_3 \end{aligned} \tag{5.84}$$

Furthermore,

$$\begin{aligned} \tilde{x}_1 &= C\dot{\tilde{x}}_2 + \tilde{x}_3 \\ &= C\dot{x}_2 + C(\Delta R_a)\dot{x}_3 + x_3 \\ &= x_1 + C(\Delta R_a)x_3 \end{aligned} \tag{5.85}$$

Finally,

$$\begin{aligned} u_{\text{req}} &= R(x_1 + C(\Delta R_a)\dot{x}_3) + L(\dot{x}_1 + C(\Delta R_a)\ddot{x}_3) + x_2 + (\Delta R_a)x_3 \\ &= \underbrace{Rx_1 + L\dot{x}_1 + x_2}_{u} + (x_3 + RC\dot{x}_3 + LC\ddot{x}_3)(\Delta R_a) \\ &= u + \frac{1}{k}[(b\,y + J_m\dot{y}) + RC(b\,\dot{y} + J_m\ddot{y}) + LC(b\ddot{y} + J_m y^{(3)})]\Delta R_a \\ &\quad + \frac{1}{k}[\tau_{\text{load}} + (RC)\dot{\tau}_{\text{load}} + (LC)\ddot{\tau}_{\text{load}}]\Delta R_a \end{aligned} \tag{5.86}$$

As a result, in this example, all unknown states of the faulty system, $\tilde{\boldsymbol{x}}(t)$, can be eliminated. The required input $u_{\text{req}}(t)$ is a function of the initial input $u(t)$, the load torque τ_{load} and its derivatives, and of the desired output y and its derivatives. The derivatives of y in (5.86) can be expressed by the output y, the state variables of the healthy system model, the load torque τ_{load} and its time derivatives. If there is no analytical function for the time history of the load torque, $\tau_{\text{load}}(t)$, but measured values carrying noise then a state variable filter (SVF) can provide filtered derivatives of $\tau_{\text{load}}(t)$ [10, 77].

$$\dot{y} = \frac{1}{J_m}(ki_a - by - \tau_{\text{load}}) \tag{5.87}$$

$$\ddot{y} = \frac{k}{J_m L_a}(u_c - R_a i_a - ky) - \frac{b}{J_m^2}(ki_a - by - \tau_{\text{load}}) - \frac{1}{J_m}\dot{\tau}_{\text{load}} \tag{5.88}$$

$$\begin{aligned} y^{(3)} &= \frac{k}{J_m L_a}\left[\frac{1}{C}(i_L - i_a) - \frac{R_a}{L_a}(u_C - R_a i_a - ky) - k\dot{y}\right] \\ &\quad -\frac{b}{J_m^2}\left[\frac{k}{L_a}(u_C - R_a i_a - ky) - b\dot{y} - \dot{\tau}_{\text{load}}\right] - \frac{1}{J_m}\ddot{\tau}_{\text{load}} \end{aligned} \tag{5.89}$$

In the case of no fault $\varDelta R_a$, the required input equals the initial input.

5.4.1.2 Fault Accommodation in Closed-Loop Systems

Let $\mathbf{K}(\,)$ denote the function of the feedback controller and let $\boldsymbol{u}_K(t) = \mathbf{K}(\boldsymbol{w}(t) - \tilde{\boldsymbol{y}}(t))$ be its output. The equations of the faulty closed loop system then read

$$\dot{\tilde{\boldsymbol{x}}}(t) = \tilde{\mathbf{A}}\tilde{\boldsymbol{x}}(t) + \tilde{\mathbf{B}}[\boldsymbol{u}_{\text{req}}(t) + \boldsymbol{u}_K(t)] \tag{5.90a}$$

$$\tilde{\boldsymbol{y}}(t) = \tilde{\mathbf{C}}\tilde{\boldsymbol{x}}(t) \tag{5.90b}$$

Differentiation of (5.90b) with respect to time yields for $\boldsymbol{u}_{\text{req}}(t)$

$$(\tilde{\mathbf{C}}\tilde{\mathbf{B}})\,\boldsymbol{u}_{\text{req}}(t) = \dot{\tilde{\boldsymbol{y}}}(t) - \tilde{\mathbf{C}}\tilde{\mathbf{A}}\tilde{\boldsymbol{x}}(t) + \tilde{\mathbf{C}}\tilde{\mathbf{A}}\boldsymbol{u}_K(t) \tag{5.91}$$

Again, if $\tilde{\mathbf{C}}\tilde{\mathbf{B}}$ is non-singular, then substituting $\boldsymbol{u}_{\text{req}}(t)$ into (5.90a) yields for the state of the faulty system the ODE

$$\dot{\tilde{\boldsymbol{x}}}(t) + \tilde{\mathbf{M}}\tilde{\mathbf{A}}\tilde{\boldsymbol{x}}(t) = \tilde{\mathbf{B}}(\tilde{\mathbf{C}}\tilde{\mathbf{B}})^{-1}\dot{\tilde{\boldsymbol{y}}}(t) - \tilde{\mathbf{M}}\mathbf{B}\boldsymbol{u}_K \tag{5.92}$$

These results coincide with Eqs. (5.70), (5.71) for an open loop system if the controller output $\boldsymbol{u}_K(t)$ is identical equal to zero, i.e. if the system is in open-loop.

5.4.2 Inverse Simulation-Based Input Reconstruction Using ARRs

So far, ARRs have been derived from a DBG model of a real system for the purpose of fault detection and isolation. Once, ARR residuals are available, they can also be used for an input reconstruction in the case of a fault. The input reconstruction approach in this section using ARR residuals does not require to isolate and to estimate the fault. As to the previously considered example system, it is sufficient to know ARR residual r_3. A different active FTC approach that also doesn't need fault isolation and estimation for recovery from a fault has recently been reported in [2]. It uses a Luenberger observer and inserts output residuals into the control loop.

In [11], a behavioural BG model of the faulty system has been coupled to a DBG of a reference model with nominal parameters by the so-called *residual sinks*. Their input is the difference between a 'measured' output from the behavioural model and its corresponding output from the reference model. The outputs of the residual sinks are inputs into the DBG and force the reference model to adapt to the faulty system so that differences between 'measured' outputs from the behavioural model and their corresponding outputs from the reference model vanish. If there is no difference, the output of a residual sink is equal to zero. Thus, the outputs of the coupling residual sinks can serve as fault indicators.

Let

$$\dot{\tilde{\boldsymbol{x}}}(t) = \tilde{\mathbf{A}}\tilde{\boldsymbol{x}}(t) + \tilde{\mathbf{B}}\boldsymbol{u}(t) \tag{5.93a}$$

$$\tilde{\boldsymbol{y}}(t) = \tilde{\mathbf{C}}\tilde{\boldsymbol{x}}(t) \tag{5.93b}$$

be the state space model of the faulty system. (Variables and matrices carry a tilde.) Then the state space model of the adapted reference model with nominal parameters reads

$$\dot{\tilde{\boldsymbol{x}}}(t) = \mathbf{A}\tilde{\boldsymbol{x}}(t) + \mathbf{B}\boldsymbol{u}(t) + \mathbf{B}_1\boldsymbol{r}(t) \tag{5.94a}$$

$$\tilde{\boldsymbol{y}}(t) = \mathbf{C}\tilde{\boldsymbol{x}}(t) + \mathbf{D}\boldsymbol{r}(t) \tag{5.94b}$$

The task now is to find an input $\boldsymbol{u}_{\text{req}}(t)$ such that the output of the reference model receiving the outputs of the residual sinks is not $\tilde{\boldsymbol{y}}(t)$ but the desired output $\boldsymbol{y}(t) = \boldsymbol{y}_{\text{des}}(t)$. With the reconstructed input $\boldsymbol{u}_{\text{req}}(t)$ and the input $\boldsymbol{r}(t)$ from the residual sinks the state of the reference model becomes $\dot{\hat{\boldsymbol{x}}}(t)$.

$$\dot{\hat{\boldsymbol{x}}}(t) = \mathbf{A}\hat{\boldsymbol{x}}(t) + \mathbf{B}\boldsymbol{u}_{\text{req}}(t) + \mathbf{B}_1\boldsymbol{r}(t) \tag{5.95a}$$

$$\boldsymbol{y}(t) = \mathbf{C}\hat{\boldsymbol{x}}(t) + \mathbf{D}\boldsymbol{r}(t) \tag{5.95b}$$

Equations (5.95a)–(5.95b) constitute a DAE system for the unknowns $\hat{\boldsymbol{x}}$ and $\boldsymbol{u}_{\text{req}}$. Differentiating the algebraic equation yields for $\boldsymbol{u}_{\text{req}}$

$$(\mathbf{CB})\,\boldsymbol{u}_{\text{req}}(t) = -\mathbf{CA}\hat{\boldsymbol{x}}(t) - \mathbf{CB}_1\boldsymbol{r}(t) - \mathbf{D}\dot{\boldsymbol{r}}(t) + \dot{\boldsymbol{y}}(t) \tag{5.96}$$

In case the matrix $\mathbf{CB}$ is nonsingular, the state $\hat{\boldsymbol{x}}(t)$ is determined by the ODE

$$\dot{\hat{\boldsymbol{x}}}(t) + \mathbf{MA}\hat{\boldsymbol{x}}(t) = \mathbf{B}(\mathbf{CB})^{-1}\dot{\boldsymbol{y}}(t) - \mathbf{MB}_1\boldsymbol{r}(t) - \mathbf{B}(\mathbf{CB})^{-1}\mathbf{D}\dot{\boldsymbol{r}}(t) \tag{5.97}$$

where $\mathbf{M} := \mathbf{B}(\mathbf{CB})^{-1}\mathbf{C} - \mathbf{I}$.

Case Study

For illustration of the input reconstruction using ARR residuals, the example of the open-loop buck converter driven DC motor in Sect. 5.4.1 is considered once again.

Figure 5.21 displays a DBG of the buck converter-DC motor system with four sensors from which the following ARRs are obtained.

$$1_1 : \text{ARR}_1 : \quad 0 = mE - (R_{\text{on}} + R_L)i_L - L\frac{di_L}{dt} - u_C \tag{5.98}$$

$$0_1 : \text{ARR}_2 : \quad 0 = i_L - C\frac{du_C}{dt} - i_a \tag{5.99}$$

$$1_2 : \text{ARR}_3 : \quad 0 = u_C - R_a i_a - L_a\frac{di_a}{dt} - k\,\omega \tag{5.100}$$

$$1_3 : \text{ARR}_4 : \quad 0 = k\,i_a - b\,\omega - J_m\frac{d\omega}{dt} - \tau_{\text{load}} \tag{5.101}$$

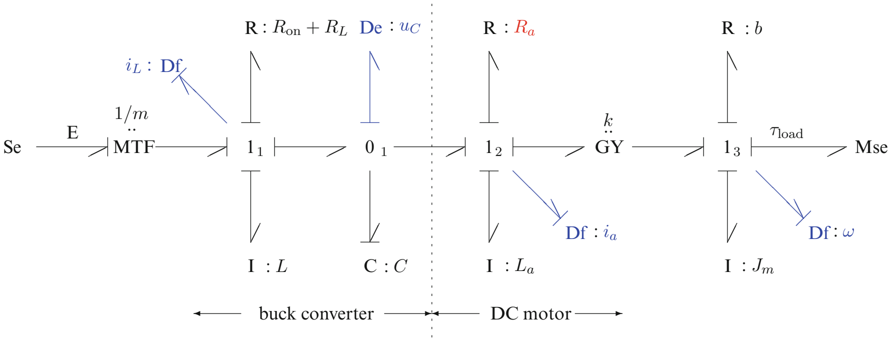


Fig. 5.21 Averaged DBG model of the healthy buck-converter-driven DC motor with four sensors

Table 5.2 Structural fault signature matrix of the BG model in Fig. 5.21 with sensors Df : i_L and De : u_C, Df : i_a, Df : ω

Component	ARR$_1$	ARR$_2$	ARR$_3$	ARR$_4$	D$_b$	I$_b$
Sw : R_{on}	1	0	0	0	1	0
R : R_L	1	0	0	0	1	0
I : L	1	0	0	0	1	0
C : C	0	①	0	0	1	①
R : R_a	0	0	1	0	1	0
I : L_a	0	0	1	0	1	0
GY : k	0	0	①	①	1	①
R : b	0	0	0	1	1	0
I : J_m	0	0	0	1	1	0
τ_{load}	0	0	0	1	1	0

In the case of a non-faulty voltage supply and non-faulty sensors, the structural dependencies of the ARRs from the component parameters are captured by the FSM in Table 5.2.

Assume that the DC-motor armature resistance R_a abruptly jumps to a value $\tilde{R}_a = 2R_a$ at time instant $t_1 = 1.0\,\text{s}$ and stays at that increased constant value. Then this fault cannot be simply identified by inspection of the FSM (Table 5.2). If the absolute value of residual r_3 of ARR$_3$ exceeds a given threshold thr$_3$, i.e. $|r_3| > \text{thr}_3$, then a look at the FSM in Table 5.2 indicates that component parameters R_a, L_a and the motor constant k have to be considered as potential fault candidates. If residual r_4 of ARR$_4$ is within admissible bounds, i.e. $|r_4| < \text{thr}_4$, then k can be excluded as a fault candidate leaving R_a and L_a as fault candidates. If a single fault hypothesis can be adopted, only one of the two parameters will be faulty.

The healthy motor driven by a healthy buck converter is considered as the reference system. The desired angular velocity $\omega_{\text{des}}(t)$ is the one of the healthy motor. An abrupt change in the armature resistance R_a or in the inductance L_a at $t = t_s$ leads to a different angular velocity $\tilde{\omega}(t)$. ARR$_3$ then reads

$$\tilde{u}_C = R_a\tilde{i}_a + L_a\frac{d}{dt}\tilde{i}_a + k\tilde{\omega} + r_3 \tag{5.102}$$

with a residual r_3 exceeding a threshold thr$_3$: $|r_3| > \text{thr}_3$.

In order to maintain the desired steady state angular velocity ω_{des}, the output voltage of the buck converter driving the motor must be adapted for $t > t_s$. The required voltage $\tilde{u}'_C = V_{\text{req}}$ can be obtained from the ARRs.

$$V_{\text{req}}(t) = R_a\tilde{i}'_a(t) + L_a\frac{d}{dt}\tilde{i}'_a(t) + k\,\omega_{\text{des}}(t) + r_3 \tag{5.103}$$

$$k\tilde{i}'_a = J_m\dot{\omega}_{\text{des}} + b\omega_{\text{des}} + \tau_{\text{load}} \tag{5.104}$$

The right-hand side of (5.104) is determined by ω_{des} and τ_{load}. If the load torque has not changed, $\tilde{i}'_a$ equals the current i_a in the non-faulty system obtained from a model of the non-faulty system. Hence, for $t > t_s$

$$\begin{aligned} V_{\text{req}} &= R_a i_a + L_a \frac{di_a}{dt} + k\omega_{\text{des}} + r_3 \\ &= u_c + r_{31} \end{aligned} \tag{5.105}$$

and

$$u_{\text{req}} \;=\; \tilde{m}E = (R_{\text{on}} + R_L)\tilde{i}_L + L\frac{d\tilde{i}_L}{dt} + V_{\text{req}} \tag{5.106}$$

$$\begin{aligned} \tilde{i}_L &= C\dot{V}_{\text{req}} + \tilde{i}'_a \;=\; C\dot{V}_{\text{req}} + i_a \\ &= C\dot{u}_C + C\dot{r}_3 + i_a \;=\; i_L + C\dot{r}_3 \end{aligned} \tag{5.107}$$

Substituting (5.107) and (5.105) into (5.106) yields

$$\begin{aligned} u_{\text{req}} \;&=\; \tilde{m}E = (m + \Delta m)E \\ &= mE + (r_3 + RC\dot{r}_3 + LC\ddot{r}_3) \end{aligned} \tag{5.108}$$

where $R := R_{\text{on}} + R_L$.

As a result, the required input, u_{req}, depends on ARR residual r_3 and its derivatives with respect to time. If the inputs into an ARR are not obtained by offline simulation of the faulty system behaviour but are sampled measured outputs of the real system then derivatives are to be computed in discrete time, or by use of interpolation polynomials, or by a state variable filter [10, 77] which is a nth order low-pass filter that provides a filtered signal and its derivatives up to the nth order.

In steady state, (5.108) becomes

$$(1 - \tilde{d})E = (1 - d)E + r_3 \tag{5.109}$$

and thus

$$\tilde{d} = d - \frac{r_3}{E} \tag{5.110}$$

In case it is known that the fault is due to a faulty armature resistance $\tilde{R}_a = R_a + \Delta R_a$, residual r_3 can be analytically determined in this example.

$$r_3 = (\Delta R_a) i_a \tag{5.111}$$

In this case, (5.108) then reads

$$\tilde{m}E = (m + \Delta m)E \tag{5.112}$$

$$= mE + \left(i_a + RC\frac{di_a}{dt} + LC\frac{d^2 i_a}{dt^2} \right) \Delta R_a \tag{5.113}$$

In the case of a faulty inductance L_a, analytical computation of the residual yields

$$r_3 = (\Delta L)\frac{di_a}{dt} \tag{5.114}$$

In steady state, ARRs (5.98)–(5.101) yield for ω_{des}

$$\left(1 + \frac{k^2}{b}\frac{1}{R_{\text{on}} + R_L + R_a}\right)\omega_{\text{des}} = \frac{k}{b}\frac{1}{R_{\text{on}} + R_L + R_a}(1-d)E - \frac{1}{b}\tau_{\text{load}} \tag{5.115}$$

Steady State Values

Given the parameter values in Table 5.3, the following steady state values of variables are obtained analytically. The reference steady state value of $\omega_{\text{des}}(t)$ becomes 41.9 rad/s. Substituting this value into (5.104) yields 1.851 A for the reference steady state value of $i_a(t)$. According to (5.100), the steady state value of the capacitor voltage in the healthy system reads $u_C = 5.63$ V. Finally, replacing R_a by $\tilde{R}_a = 2R_a$ gives for the faulty time evolution $\tilde{\omega}(t)$ the steady state value 11.7 rad/s.

Table 5.3 Component parameters of the buck converter driven DC-motor

Parameter	Value	Units	Meaning
E	12.0	V	Voltage supply
L	20	mH	Inductance
R_L	0.1	Ω	Resistance of the coil
R_{on}	0.1	Ω	ON resistance (switch, diode)
d	0.5	–	Duty ratio
C	40	µF	Capacitance
L_a	2.6	mH	Armature inductance
R_a	2.0	Ω	Armature resistance
ΔR_a	2.0	Ω	
k	0.046	V s/rad	Motor constant
J_m	$7.0 \cdot 10^{-4}$	kg m^2	Moment of inertia
b	$8.4 \cdot 10^{-4}$	N m s/rad	Friction coefficient
T_{load}	0.05	N m	Load moment
T_i	1.0	s	Integral time constant,
K_p	0.4	–	Proportional gain of the PI controller

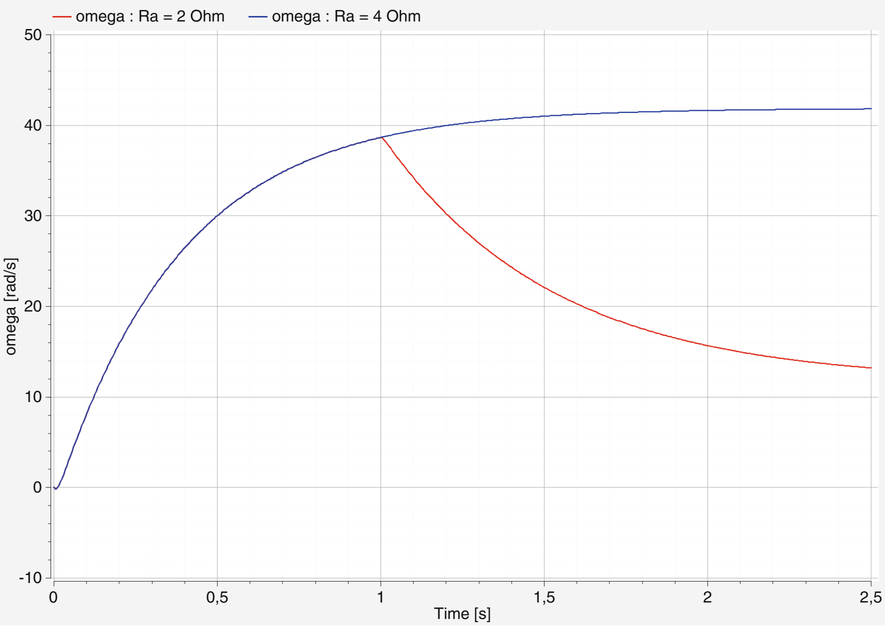


Fig. 5.22 Drop of the angular velocity $\omega(t)$ in the case of an increase of armature resistance R_a

5.4.2.1 Simulation of a Fault Scenario

In an offline simulation, the equations of the real system model are solved numerically and sensor outputs from the model are used in the evaluation of ARRs. As of time instant $t_1 = 1.0\,\text{s}$ the system is assumed to become faulty and the reconstructed signal $\tilde{m}(t)$ (5.108) instead of $m(t)$ is used to control the switching of the transistor Q so that the faulty system is forced to resume the angular velocity ω_{des} of the healthy system.

Figure 5.22 indicates that the steady state value of $\omega(t)$ of the open-loop system significantly drops from 42 to 12 rad/s in the case R_a increases from $2\,\Omega$ to $4\,\Omega$ in accordance with analytical results above.

If the angular velocity of the motor shall be ω_{des} despite a faulty armature resistance $\tilde{R}_a$, then the duty cycle d must be changed. In this case, (5.115) reads

$$\left[(R_{\text{on}} + R_L + \tilde{R}_a)b + k^2\right]\omega_{\text{des}} = k(1-\tilde{d})R - (R_{\text{on}} + R_L + \tilde{R}_a)\tau_{\text{load}} \quad (5.116)$$

The steady state value $\tilde{d} = d - i_a \Delta R_a / E$ obtained from (5.110) and (5.111) satisfies (5.116).

In the considered fault scenario, it has been assumed that resistance R_a has become faulty as of $t_1 = 1.0\,\text{s}$. As the presented input reconstruction uses ARR residuals an isolation of the fault by means of parameter estimation is not necessary.

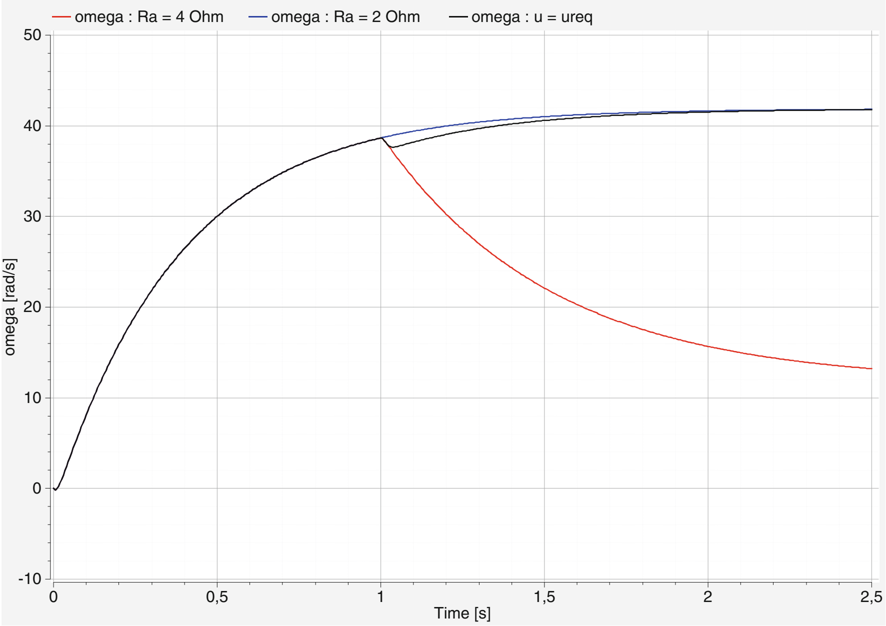


Fig. 5.23 Angular velocity $\omega(t)$ of the faulty motor with fault compensation

The evaluation of ARR residuals for some further time steps concurrently to the performance of an engineering system, however, takes some time and causes a time delay especially if ARRs cannot be established in closed form and an entire DBG model has to be re-evaluated instead in order to compute the ARR residuals. In addition, fault detection requires residual signals to be checked against their admissible fault thresholds. Therefore, accounting for some time delay in FTC due to fault detection and input reconstruction, the correction term $(\Delta m)E$ that compensates the fault in R_a is assumed to become effective as of $t_2 = 1.02$ s, i.e. 20 ms after the fault has happened. The result is depicted in Fig. 5.23.

It depends on the time needed for fault detection and input reconstruction how much the faulty angular velocity $\tilde{\omega}(t)$ will progressively deviate from the desired time evolution $\omega_{\text{des}}(t)$. In other words, the state of the system drifts away from the current desired state to a faulty state the longer it takes to provide the new system input that compensates for the parametric fault. When the new system input becomes effective it depends on the transient behaviour of the faulty system, i.e. on its time constants, how fast the system will recover from the faulty state and how fast its output will reach the desired trajectory. With the assumed delay in FTC of 20 ms and the given parameter values of the example system it takes about 1 s for the angular velocity to reach again its desired time history. If the input into the faulty open loop system is not corrected, $\tilde{\omega}(t = 2\,\text{s})$ would deviate about 25.9 rad/s or about 62 % from its desired value 1 s after the occurrence of the fault.

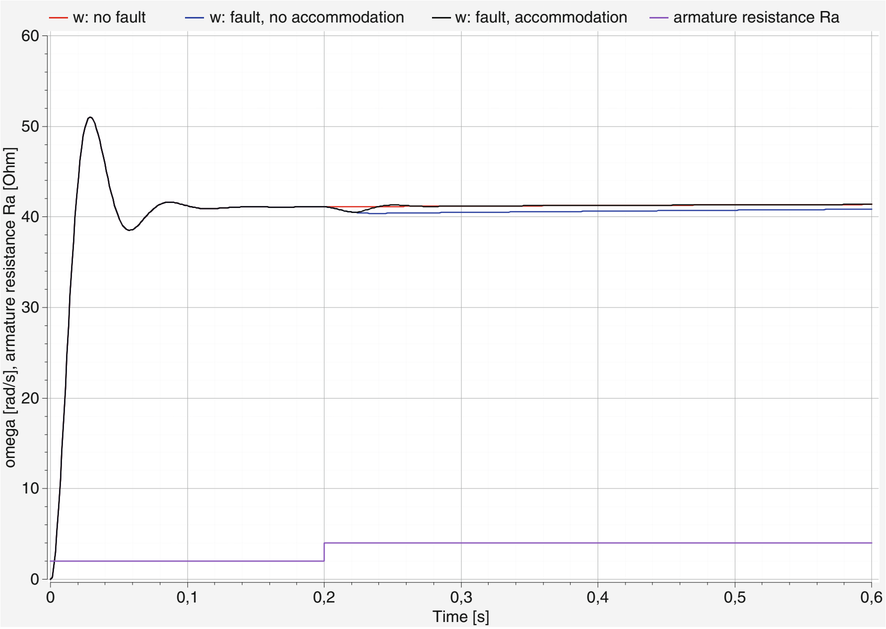


Fig. 5.24 Angular velocity $\omega(t)$ of the closed-loop faulty motor with and without fault accommodation

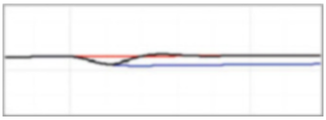


Fig. 5.25 Waveforms of angular velocity $\omega(t)$ in the time interval around $t = 0.2\,\text{s}$

Figure 5.24 displays the time evolution of angular velocity $\omega(t)$ of the buck-converter driven DC motor in closed-loop with a PI feedback controller with and without fault accommodation.

The step response of the closed-loop healthy system is depicted as a reference in red colour. In comparison to the open-loop system, the PI controller makes sure that the angular velocity of the motor quickly reaches its steady state value of about 41.9 rad/s. At time instant $t = 0.2\,\text{s}$ the value of the armature resistance R_a abruptly doubles in this fault scenario as depicted in the bottom of Fig. 5.24. As a result, the angular velocity drops (blue waveform). The PI controller can compensate for that persistent fault, but it takes some time for $\omega(t)$ to converge to the steady state value. If fault accommodation becomes effective with some delay as of $t = 0.22\,\text{s}$, $\omega(t)$ returns to its steady state value very quickly (black waveform). Figure 5.25 displays an enlargement of the time histories of $\omega(t)$ in the time interval around the event at $t = 0.2\,\text{s}$.

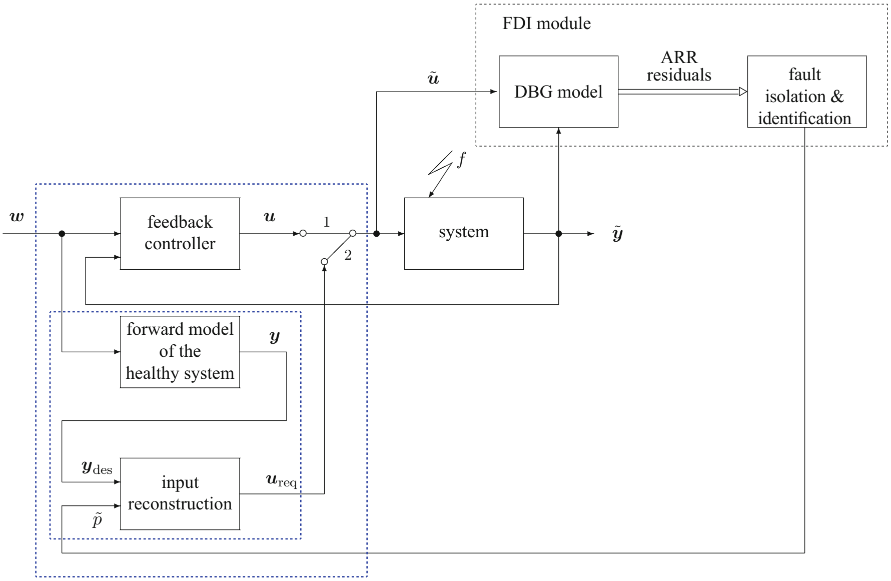


Fig. 5.26 Scheme of the inverse simulation based FTC strategy using identified faulty parameters

5.4.3 Scheme of the Inverse Simulation-Based FTC Strategy

Figure 5.26 displays the scheme of the active FTC strategy based on inverse simulation by solving a DAE system. Blue coloured dashed bounding boxes indicate that the nominal feedback controller is assisted by a feedforward controller of which the forward model of the healthy system and the inverse model of the faulty system are core components.

The engineering system may be subject to a fault as of some time instant t_1. Its input $\tilde{\boldsymbol{u}}(t)$ and measured outputs $\tilde{\boldsymbol{y}}(t)$ are fed into a DBG model that generates ARR residuals. On the basis of these ARR residuals a diagnosis module decides whether a fault has happened or not. If a parametric fault has occurred, it is isolated and quantified. At time $t_2 > t_1$, the identified faulty parameter $\tilde{p}$ is provided into an input reconstruction module that takes the output $\boldsymbol{y}(t)$ of a forward model of the healthy system as the desired output $\boldsymbol{y}_{\text{des}}(t)$ and determines an input $\boldsymbol{u}_{\text{req}}(t)$ into the faulty system by numerically solving a DAE system so that the output of the faulty system recovers from the fault and matches the one of the healthy system.

Alternatively, the ARR residual generator may feed its outputs directly into the input reconstruction module as has been illustrated in Sect. 5.4.2 so that an isolation and estimation of the fault is not necessary. If a small disturbance on the faulty system occurs after recovery from the fault, it is compensated by the feedback controller.

The nominal feedback controller can be designed independently from the FDI and FTC problem. The control law may have been designed so that the controller is robust against some modelling uncertainties and against anticipated faults from a certain class of faults. The forward model is a BG that can be developed offline on the basis of a physical understanding of the system. The DAE system derived from that BG may be considered as a DAE system of the inverse model. Symbolic processing features such as reformulation of equations, symbolic differentiation, index reduction (given the DAE index is not mode-dependent) and code generation of a Modelica modelling and simulation environment such as OpenModelica can be exploited for an automatic offline construction of the inverse model [66]. A requirement for the use of an inverse plant model in a controller is that its DAE system has a unique solution and that the inverse model is stable. However, even if the DAE system for the inverse model can be solved numerically, it is not guaranteed that the inverse model is stable in a region around all possible operating points. If an inverse model is unstable, or if equations cannot be inverted, or if there are actuator limits, the forward model may be modified before inversion. For instance, if the system behaviour can be described by a single-input, single-output (SISO) system, unstable zeros in its transfer function could be omitted. In any case, an appropriate modification of a forward model that captures the dynamic system behaviour sufficiently accurate results in an approximate inverse model. Its output, however, will not fully compensate a fault in the system.

5.5 Conclusion

A contribution of bond graph modelling to FDI is that ARRs serving as fault indicators can be derived from a DBG of a system in a systematic manner. This chapter shows that beyond FDI these ARRs can also be used for an input reconstruction through inverse simulation. The presented integrated approach has the following advantages.

- The offline generated DBG model, a forward model of the healthy system and the DAE system of the inverse model can be computed numerically in parallel on a multiprocessor system concurrently to the operation of the engineering system.
- Once a fault has occurred, values of the reconstructed input into the faulty system are available with little time delay which enables a fast recovery from a fault.
- Neither ARRs nor the reconstructed input are needed in closed analytical form.
- Existing sophisticated modelling and simulation software such as OpenModelica can be used for the construction of the inverse model in case the DAE index is not mode-dependent.

In this chapter it is assumed that the dynamic behaviour of the engineering system under consideration can be described sufficiently accurate by a mode switching LTI multiple-input, multiple-output model. The latter one is a special case of a hybrid model that is of interest in various applications. The presented approach is not limited to this subset of hybrid models. In the more general case, nonlinear hybrid DAE systems of the DBG model, the forward and the inverse models are to be solved numerically. However, the existence of a nonlinear inverse model, i.e. a unique solution of the DAE system describing the inverse model is not guaranteed in general. Moreover, to be used in a controller, the inverse model must be stable. If the inverse model does not exist or is not stable, modifications of the forward model before inversion may be a remedy. For mode switching LTI models, the numerical integration based on the BDF solver may fail due to structural singularities caused by switches. In such cases small modifications of a hybrid model based on physical understanding in consideration of the purpose of a simulation become necessary. In the case of a complex hierarchical model the problem is to start from a program's error message in mathematical terms and to identify the part of the model that has caused the computational problem.

An engineering system may be subject to modelling uncertainties, parameter variations and disturbances besides faults. Sensor signals carry measurement noise. Therefore, the latter ones should be appropriately filtered before used in the evaluation of ARRs. To avoid false alarms caused by parameter variations, adaptive thresholds for ARR residuals can be obtained from a DBG in LFT form or from an incremental BG. As long as ARR residuals are within their time-dependent adaptive bounds, no adaption of system input values is necessary. Zero values are fed into the constantly active reconstruction module. As a result, the input into the system is not changed.

For systems operating in various modes, their dynamic behaviour in one mode may be quite different from that in another mode. Accordingly, given a hybrid model, active FTC using ARRs requires for system mode identification in order to use the correct values for the discrete switch state variables in the ARRs or in the equations derived from the DBG. That is, input reconstruction actually is mode dependent.

Acknowledgements The author would like to thank Peter Gawthrop from Melbourne School of Engineering for reading of the draft of this chapter and for his valuable feedback that helped to improve some aspects.

References

1. Acary, V., Bremond, M., Huber, O., Perignon, F., & Pissard-Gibollet, R. (2016). *Siconos*. http://siconos.gforge.inria.fr/users_guide/lcp_solvers.html
2. Allous, M., & Zanzouri, N. (2014). Active fault tolerant control based on bond graph approach. *Advances in Electrical Engineering*. Article ID 216153. http://dx.doi.org/10.1155/2014/216153
3. Alvari, M., Luo, M., Wang, D., & Zhang, D. (2011). Fault diagnosis for power electronic inverters: A model-based approach. In *Proceedings of 2011 IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives (SDEMPED)* (pp. 221–228). Bologna, Italy: IEEE. doi:10.1109/DEMPED.2011.6063627.
4. Alwi, H., Edwards, C., & Pin Tan, C. (2011). *Fault detection and fault-tolerant control using sliding mode*. New York: Springer. http://www.springer.com/978-0-85729-649-8

5. Arogeti, S. A., Wang, D., & Low, C. B. (2010). Mode identification of hybrid systems in the presence of fault. *IEEE Transactions on Industrial Electronics, 57*(4), 1452–1467.
6. Baïkeche, H., Marx, B., Maquin, D., & Ragot, J. (2006). On parametric and nonparametric fault detection in linear closed-loop systems. In *Workshop on Advanced Control and Diagnosis, ACD'2006, Nancy, France, CDROM*.
7. Battle, C., Fossas, E., Merrillas, I., & Miralles, A. (2005). Generalized discontinuous conductionmodes in the complementarity formalism. *IEEE Transactions on Circuits and Systems, 52*(8), 447–451.
8. Bergero, F., Floros, X., Fernández, J., Kofman, E., & Cellier, F. E. (2012). Simulating modelica models with a stand-alone quantized state systems solver. In *Proceedings of 9th International Modelica Conference, Munich, Germany* (pp. 237–246).
9. Blanke, M., Frei, C., Kraus, F., Patton, R., & Staroswiecki, M. (2000). What is fault-tolerant control? Aalborg University, Department of Control Engineering. http://www.iau.dtu.dk/secretary/pdf/safeprocess_02h.pdf
10. Blanke, M., Kinnaert, M., Lunze, J., & Staroswiecki, M. (2006). *Diagnosis and fault-tolerant control*. Berlin: Springer.
11. Borutzky, W. (2010). Bond graph methodology – Development and analysis of multidisciplinary dynamic system models. London, UK: Springer. ISBN: 978-1-84882-881-0 (print).
12. Borutzky, W. (Ed.). (2011). Bond graph modelling of engineering systems – Theory, applications and software support. New York, NY, USA: Springer.
13. Borutzky, W. (2015). *Bond graph model-based fault diagnosis of hybrid systems*. Switzerland: Springer International Publishing.
14. Borutzky, W. (2015). Fault accommodation by inverse simulation through solving a differential algebraic system obtained from a bond graph. In A. Bruzzone, G. Dauphin-Tanguy, S. Junco, & F. Longo (Eds.), *Proceedings of the 8th International Conference on Integrated Modelling and Analysis in Applied Control and Automation (IMAACA 2015), part of the I3M Multiconference, Bergeggi, Italy* (pp. 29–38).
15. Borutzky, W., & Cellier, F. E. (1996). Tearing algebraic loops in bond graphs. *Transactions of the SCS, 13*(2), 102–115.
16. Buchholz, J. J., & von Grünhagen, W. (2008). *Inversion impossible?* Tech. rep., Bremen University of Applied Sciences and DLR Braunschweig, Germany.
17. Camlibel, K., Iannelli, L., & Vasca, F. (2004). Modelling switching power converters as complementarity systems. In *IEEE Conference on Decision and Control* (pp. 2328–2333). Piscataway: IEEE.
18. Cellier, F. E., & Kofman, E. (2006). *Continuous system simulation*. New York: Springer.
19. Cellier, F. E., & Krebs, M. (2007). Analysis and simulation of variable structure systems using bond graphs and inline integration. In *Proceedings of ICBGM 07, 8th SCS International Conference on Bond Graph Modeling and Simulation, San Diego, CA, USA* (pp. 29–34).
20. Cellier, F. E., & Nebot, A. (2005). The Modelica bond graph library. In *Proceedings of* 4*th International Modelica Conference, Hamburg, Germany* (Vol. 1, pp. 57–65).
21. Chen, J., Patton, R., & Zhang, H. (1996). Design of unknown input observers and robust fault detection filters. *Journal of Control, 63*(1), 85–105. doi:10.1080/00207179608921833
22. Controllab Products. (n.d.). *20-sim the power in modeling*. http://www.20sim.com
23. Cormerais, H., Buisson, J., Leirens, S., & Richard, P. Y. (2002). Calcul symbolique de l'ensemble des équations d'état pour les graphs en commutation. In *Personnel of CIFA 2002, Nantes, France*.
24. Dauphin-Tanguy, G., & Kam, C. (1999). How to model parameter uncertainties in a bond graph framework. In G. Horton, D. Möller, & U. Rüde (Eds.), *Simulation in Industry, 11th European Simulation Symposium, ESS'99, Erlangen, Germany* (pp. 121–125).
25. de la Calle, A., Cellier, F., Yebra, L., & Dormido, S. (2013). Improvements in bondlib, the modelica bond graph library. In *Proceedings of the 8th EUROSIM Congress on Modelling and Simulation* (pp. 282–287). Cardiff: IEEE.
26. Djeziri, M., Merzouki, R., Ould Bouamama, B., & Dauphin-Tanguy, G. (2007). Bond graph model based for robust fault diagnosis. In *Proceedings of the 2007 American Control Conference* (pp. 3017–3022). New York City, USA: IEEE.

27. Djeziri, M. A. (2007). Diagnostic des Systèmes Incrementains par l'Approche Bond Graph. Ph.D. thesis, Ecole Centrale de Lille, France. http://hal.archives-ouvertes.fr/docs/00/20/00/30/PDF/These-Djeziri-07-12-2007.pdf
28. Djeziri, M. A., Ould Bouamama, B., & Merzouki, R. (2009). Modelling and robust FDI of steam generator using uncertain bond graph model. *Journal of Process Control, 19*, 149–162.
29. Ducreux, J., Dauphin-Tanguy, G., & Rombaut, C. (1993). Bond graph modelling of commutation phenomena in power electronic circuits. In J. Granda & F. Cellier (Eds.), *International Conference on Bond Graph Modeling, ICBGM'93, Proceedings of the 1993 Western Simulation Multiconference. Simulation Series* (Vol. 25, No. 2, pp. 132–136). San Diego, CA, USA: SCS Publishing. ISBN: 1-56555-019-6.
30. Elmqvist, H., Cellier, F. E., & Otter, M. (1993). Object-oriented modeling of hybrid systems. In *Proceedings of the 1993 European Simulation Symposium, Ghent* (pp. xxxi–xli).
31. Elmqvist, H., Otter, M., & Cellier, F. E. (1995). A new mixed symbolic/numeric approach for solving differential-algebraic equations systems. In *Proceedings of European Simulation Multiconference, ESM'95, Prag* (pp. xxiii–xxxiv).
32. Fritzson, P. (2004). *Principles of object-oriented modeling and simulation with Modelica 2.1*. Hoboken: Wiley. ISBN: 0-471-47163-1.
33. Gawthrop, P. (1995). Bicausal bond graphs. In F. Cellier & J. Granda (Eds.), *ICBGM'95, International Conference on Bond Graph Modeling and Simulation: Vol. 27(1). Simulation series* (pp. 83–88). San Diego: SCS Publishing.
34. Ghoshal, S., Samantaray, A., & Mukherjee, A. (2005). Improvements to single fault isolation using estimated parameters. In J. Granda, & F. Cellier (Eds.) *Proceeding of the International Conference on Bond Graph Modeling, ICBGM'05, SCS Publishing, New Orleans, LA, USA: Vol. 37. Simulation series* (No. 1, pp. 301–306). ISBN: 1-56555-287-3.
35. Hairer, E., & Wanner, G. (1996). *Solving ordinary differential equations II, stiff and differential-algebraic problems* (2nd ed.). Berlin: Springer-Verlag.
36. Hao-Yuan Chou, H. (2000). Fault diagnosis of a heat exchanger system using unknown input observers. Master's thesis, University of Toronto, Graduate Department of Electrical and Cornputer Engineering.
37. Heemels, W. P. M. H., Lehmann, D., Lunze, J., & De Schutter, B. (2009). Chapter: Introduction to hybrid systems. In *Handbook of hybrid systems control: Theory, tools, applications*. Cambridge: Cambridge University Press (pp. 4–30).
38. HighTec Consultants. (n.d.). SYMBOLS Sonata™ . http://www.htcinfo.com/
39. Isermann, R., & Ballé, P. (1997). Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice, 5*(5), 709–719.
40. Kofman, E. (2004). Discrete event simulation of hybrid systems. *SIAM Journal on Scientific Computing, 25*(5), 1771–1797. doi:10.1137/S1064827502418379. http://dx.doi.org/10.1137/S1064827502418379
41. Levy, R., Arogeti, S., & Wang, D. (2013). Improving mode-change and fault isolation of hybrid systems using instantaneous sensitivity matrices. In *Proceeding of the 1st International and 16th National Conference on Machines and Mechanisms (iNaCoMM2013)* (pp. 116–123).
42. Loureiro, R. (2012). Bond graph model based on structural diagnosability and recoverability analysis: Application to intelligent autonomous vehicles. Ph.D. thesis, L' Université Lille 1.
43. Lundvall, H., Fritzson, P., & Bachmann, B. (2008). *Event handling in the OpenModelica compiler and runtime system*. Tech. rep., Linköping University, Department of Computer and Information Science, PELAB - Programming Environment Laboratory, Series: Technical reports in Computer and Information Science. ISSN 1654-7233; 2.
44. Lunze, J. (2004). Control reconfiguration. In *Control systems, robotics and automation* (Vol. XVI, pp. 320–333). Oxford, UK: Eolss Publishers [Encyclopedia of Life Support Systems (EOLSS) edn.].
45. Lunze, J., & Richter, H. (2008). Reconfigurable fault-tolerant control: A tutorial introduction. *European Journal of Control, 5*, 359–386.
46. Margetts, R. (2013). Modelling and analysis of hybrid dynamic systems using a bond graph approach. Ph.D. thesis, University of Bath, UK.

47. Modelica Association. (n.d.). *Modelica and the modelica association*. http://www.modelica.org
48. Murray-Smith, D. (2012). *Modelling and simulation of integrated systems in engineering – Issues of methodology, quality, test and application*. Cambridge, UK: Woodhead Publishing.
49. Murray-Smith, D. J. (2011). Feedback methods for inverse simulation of dynamic models for engineering systems applications. *Mathematical and Computer Modelling of Dynamic Systems, 17*(5), 515–541.
50. Nacusse, M., & Junco, S. J. (2011). Passive fault tolerant control: A bond graph approach. In: A. Bruzzone, G. Dauphin-Tanguy, S. Junco, & M. A. Piera (Eds.), *Proceedings of the 5th International Conference on Integrated Modelling and Analysis in Applied Control and Automation (IMAACA 2011), DIPTEM Università di Genova, Rome, Italy* (pp. 75–82).
51. Ngwompo, R., Scavarda, S., & Thomasset, D. (1996). Inversion of linear time-invariant SISO systems modelled by bond graph. *Journal of the Franklin Institute, 333*(B)(2), 157–174.
52. Ngwompo, R., Scavarda, S., & Thomasset, D. (2001). Physical model-based inversion in control systems design using bond graph representation part 1: Theory. *Proceedings of the IMechE Part I Journal of Systems and Control Engineering, 215*(2), 95–103.
53. Ngwompo, R., Scavarda, S., & Thomasset, D. (2001). Physical model-based inversion in control systems design using bond graph representation part 2: Applications. *Proceedings of the IMechE Part I Journal of Systems and Control Engineering, 215*(2), 105–112.
54. Niemann, H., & Stoustrup, J. (1997). Robust fault detection in open loop vs. closed loop. In *Proceedings of the 36th IEEE Conference on Decision and Control* (Vol. 5, pp. 4496–4497).
55. Noura, H., Sauter, D., Hamelin, F., & Theilliol, D. (2000). Fault-tolerant control in dynamic systems: Application to a winding machine. *Control Systems 20*(1), 33–49. doi:10.1109/37.823226.
56. OpenModelica Consortium. (n.d.). *OpenModelica*. https://www.openmodelica.org/
57. Otter, M., Elmqvist, H., & Mattson, S. E. (1999). Hybrid modeling in modelica based on the synchronous data flow principle. In *Proceedings of the 1999 IEEE Symposium on Computer-Aided Control System Design, CACSD'99, IEEE Control Systems Society, HI, USA*
58. Otter, M., Elmqvist, H., & Mattsson, S. E. (1999). *Modeling of mixed continuous/discrete systems in modelica*. Tech. rep.
59. Pantelides, C. (1988). The consistent initialization of differential-algebraic systems. *SIAM, Journal of Scientific and Statistical Computation, 9*, 213–231.
60. Provan, G. (2009). Model abstractions for diagnosing hybrid systems. In *Proceedings of the 20th International Workshop on Principles of Diagnosis, DX-09, Stockholm, Sweden* (pp. 321–328). http://www.cs.ucc.ie/ccsl/GP-papers/2009/Provan_DX_2009_2.pdf
61. Rahmani, A., Sueur, C., & Dauphin-Tanguy, G. (1992). Formal determination of controllability/observability matrices for multivariable systems modelled by bond graph. In *Proceeding of IMACS/SICE International Symposium of Robotics, Mechatronics and Manufacturing, System'92* (pp. 573–580)
62. Samantaray, A., Medjaher, K., Ould Bouamama, B., Staroswiecki, M., & Dauphin-Tanguy, G. (2006). Diagnostic bond graphs for online fault detection and isolation. *Simulation Modelling Practice and Theory, 14*(3), 237–262.
63. Samantaray, A., & Ould Bouamama, B. (2008). *Model-based process supervision – A bond graph approach. Advances in industrial control*. London: Springer.
64. Staroswiecki, M. (n.d.). Fault tolerant systems. In *Control systems, robotics and automation* (Vol. XVI). Oxford, UK: Eolss Publishers [Encyclopedia of Life Support Systems (EOLSS) edn.].
65. Tan, S., & Vandewalle, J. (1988). Inversion of singular systems. *IEEE Transactions on Circuits and Systems, 35*(5), 583–587.
66. Thümmel, M., Looye, G., Kurze, M., Otter, M., & Bals, J. (2005). Nonlinear inverse models for control. In *Proceedings of the 4th International Modelica Conference, Hamburg, Germany* (pp. 267–279).
67. Tiller, M. (2001). *Introduction to physical modeling with modelica*. Boston, MA: Kluwer Academic Publishers. ISBN: 0-7923-7367-7.

68. Touati, Y., Merzouki, R., & Ould Bouamama, B. (2011). Fault detection and isolation in presence of input and output uncertainties using bond graph approach. In A. Bruzzone, G. Dauphin-Tanguy, S. Junco, & M. Piera (Eds.), *Proceedings of the 5th International Conference on Integrated Modeling and Analysis in Applied Control and Automation (IMAACA 2011), DIPTEM University of Genoa* (pp. 221–227).
69. Touati, Y., Merzouki, R., Ould Bouamama, B., & Loureiro, R. (2012). Detectability and isolability conditions in presence of measurement and parameter uncertainties using bond graph approach. In *8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS), Mexico City, Mexico* (pp. 958–963).
70. van der Schaft, A. J., & Schuhmacher, M. (1998). Complementarity modeling of hybrid systems. *IEEE Transactions on Automatic Control, 43*(4), 483–490.
71. Venkatasubramanian, V., Rengaswamy, R., Yin, K., & Kavuri, S. (2003). A review of process fault detection and diagnosis, part I: Quantitative model-based methods. *Computers and Chemical Engineering, 27*, 293–311.
72. Venkatasubramanian, V., Rengaswamy, R., Yin, K., & Kavuri, S. (2003). A review of process fault detection and diagnosis, part II: Qualitative models and search strategies. *Computers and Chemical Engineering, 27*, 313–326.
73. Venkatasubramanian, V., Rengaswamy, R., Yin, K., & Kavuri, S. (2003). A review of process fault detection and diagnosis, part III: Process history based methods. *Computers and Chemical Engineering, 27*, 327–346.
74. Wainer, G. A. (2009). *Discrete-event modeling and simulation – A practitioner's approach*. Boca Raton, FL, USA: CRC Press, Taylor and Francis Group.
75. Wang, D., Yu, M., Low, C., & Arogeti, S. (2013). *Model-based health monitoring of hybrid systems*. New York: Springer.
76. Wolfram, S. (2003). *The mathematica book* (5th ed.). Champaign: Wolfram Media, Inc.
77. Young, P. C. (1981). Parameter estimation for continuous-time models – A survey. *Automatica, 17*(1), 23–39.
78. Zimmer, D. (2010). Equation-based modeling of variable-structure systems. Ph.D. dissertation no. 18924, ETH Zürich.
79. Zimmer, D. (2013). A new framework for the simulation of equation-based models with variable structure. *Simulation, 89*(8), 935–963.

Chapter 6
Model-Based Diagnosis and Prognosis of Hybrid Dynamical Systems with Dynamically Updated Parameters

Om Prakash and A.K. Samantaray

6.1 Introduction

Fault detection and isolation (FDI) and prognosis for large complex process engineering systems are important research areas of industrial importance in order to improve the safety, reliability and availability of critical machineries/processes. In condition-based maintenance (CBM), FDI is intended for prompt detection, isolation and classification of any fault in a system and to quantify the severity of fault; whereas prognosis is intended to predict the remaining useful life (RUL) of faulty component or subsystem based on the current health status of the system and its past degradation profile or trend provided by diagnosis. Precise prediction of RUL assists the plant technicians to plan the future maintenance activities. Since diagnosis and prognosis both are concerned with the health monitoring of the industrial system, subsystems or components; it is logical to integrate them in a common framework for process supervision. There are generally two types of fault situations, namely anticipated or unanticipated types. An anticipated fault situation is generally known in advance based upon the history of system behaviour and past experience; but, the unanticipated or unexpected fault situation is generally not known in advance and that must be detected during process monitoring to maintain the safety and reliability of the system. Nowadays, many modern integrated systems or processes such as chemical plants, automobiles and airplanes use embedded system architecture where electronics and communication systems play important roles. These systems contain various dynamical components or subsystems which exhibit both continuous and discrete dynamics and are hence called hybrid systems. In a hybrid dynamical system, faulty discrete events may occur in addition to parametric faults and occurrence of these may be unknown in advance. Most of the

O. Prakash • A.K. Samantaray (✉)
Systems, Dynamics and Control Laboratory, Department of Mechanical Engineering,
Indian Institute of Technology-Kharagpur, 721302 Kharagpur, India
e-mail: omimech21@gmail.com; samantaray@mech.iitkgp.ernet.in; samantaray@lycos.com

W. Borutzky (ed.), *Bond Graphs for Modelling, Control and Fault Diagnosis of Engineering Systems*, DOI 10.1007/978-3-319-47434-2_6

model-based diagnosis and prognosis methods available in literature are intended for continuous systems and these approaches cannot be easily applied to hybrid dynamical systems as the supervision of such systems need tracking of continuous as well as discrete state variables.

Generally two types of fault sources may occur in a hybrid dynamical system. The first one is a parametric fault related to some component degradation and the second one is due to some unexpected transition of nominal mode of the system, i.e., discrete fault (e.g., valve stuck on or stuck off fault, controller transition command failure, etc.). Hybrid dynamics contain two types of discrete transitions: supervisory controlled discrete transition and autonomous mode discrete transition. Usually, the supervisory controlled discrete input to the plant is implemented in software and it is possible to measure the discrete input signals issued by the controller at the interface between the supervisory controller and the plant. So, we assume that the supervisory controlled discrete input signals to the plant are directly observed and known to us. Another way to determine controlled mode transitions is to use the model of the supervisory controller to predict such transitions. Even when the supervisory controlled mode change information is known to us; a discrete fault may be possible like valve stuck on or stuck off fault and pump stuck on or pump stuck off fault, etc., and such faults should also be detected and isolated. In contrast to supervised controlled discrete transition, autonomous mode discrete transitions are usually not known and may not be directly measurable. However, the conditions for autonomous mode transitions are known either in terms of measured plant output variables or in terms of state variables. So, the autonomous mode transitions can be known to diagnosis module based on the measurement of the outputs.

Most of the existing diagnosis and prognosis approaches are based on single fault hypothesis. These assume that the system or subsystem is immediately repaired once a fault is detected and isolated. However, a simple fault can lead to a sequence of other catastrophic faults and it may not be possible to repair each fault within available time. Moreover, some faults may be tolerated and the process operation may be continued in the presence of one or more known faults and RUL of such faults should be known to plant technician so that maintenance activities can be scheduled accordingly. Detection of mode transition and any subsequent fault (which may be serious) after a few known minor faults should also be possible. FDI method based on single fault hypothesis fails to predict the actual fault candidates when next fault occurs after the first fault because effects of one fault may be concealed or compensated by the effects of another fault. One solution to this problem is to use a lot of sensors to decouple fault effects. However, this is a costly approach and each process variable may not be measurable. Other approaches rely on building observers or a bank of observers (including unknown input observers) of the system and tapping measurements/inconsistencies from the observer. However, observer-based approaches cannot be easily applied to hybrid and often non-linear dynamical systems.

For better planning and scheduling of maintenance activities, a good supervision system should detect and isolate small faults and should predict the RUL of faulty/degraded components. Without isolation of correct faults and their types, RUL estimation is not possible. Hence, it is needed to develop a method having ability to correctly detect and isolate the actual fault of unknown type and unknown degradation behaviour and at the same time, it should provide some information about the severity of the fault and predict the RUL of the faulty components keeping the system level performance constraints in the view.

In a hybrid dynamical system, various components or subsystems operate in different modes or environmental conditions. This results in varying degradation rate of the components throughout the system's life cycle. In fact, the prognosis of hybrid systems is challenging due to the fact that the same component can exhibit different degradation behaviours in different operating modes. Traditionally, RUL estimation is performed by utilizing a single degradation model which assumes that degradation rate parameters used in the degradation model are constants. Although a single degradation model may be enough for a particular degradation pattern, it does not suffice when components or subsystems have different operational profiles. Utilization of multiple degradation models which include operational modes as additional control parameter and evolve through degradation model identification is suggested in this chapter. Note that identifying appropriate multiple degradation models is a challenging task when components have dynamic degradation patterns.

This chapter precisely deals with the afore-mentioned problem faced in diagnosis and prognosis of hybrid systems. The premise of the solution proposed in this chapter is a basic assumption that there is a very rare chance of occurrence of simultaneous faults. Even apparent simultaneous faults are separated by a small time interval and we assume that time interval is large enough to carry out the necessary parametric fault or mode identification, degradation pattern identification and model updating steps. We assume that infinite mode transitions in a finite time do not occur and only partial parametric faults occur in a system. For RUL estimation, it is also assumed that mode of operation of each hybrid component is known in advance.

Different approaches for model-based diagnosis and prognosis have been developed depending on the kind of knowledge used to describe the process model. Usually, a specific methodology is applied for a specific process. Diagnosis methods may be broadly classified into two types: model-based methods and data-driven-based methods. Likewise, prognosis methods intended for RUL estimation can also be classified into three types: model-based prognosis, data-driven prognosis and experience or probability-based prognosis [1]. Every method has its own advantages and disadvantages. In the present chapter, bond graph model-based diagnosis and prognosis (MBDP) scheme is proposed.

For model-based process supervision, a precise and reliable mathematical model of the actual plant behaviour is required. A unified multi-energy domain Bond Graph (BG) [2, 3] and its extended form Hybrid Bond Graph (HBG) [4–6] are well-suited for modelling of continuous and hybrid dynamical behaviours, respectively. BG tool is also useful in the design and development of model-based FDI for both continuous and hybrid dynamical systems [6, 7]. A considerable amount of literature can be found related to model-based FDI for hybrid dynamical systems [8–11]; but, very few works are reported on integration of both diagnosis and prognosis of hybrid dynamical system in a common framework. Also, very few literatures are available for prognosis of hybrid dynamical systems. In model-based diagnosis, the BG model is used to derive a set of consistency rules called analytical redundancy relations (ARRs). ARRs are constraints expressed in terms of measurable process variables and nominal parameters of plant [7]. These constraints remain valid until a system operates according to its normal operation model. A fault is detected by monitoring the trend of the residuals. For consistency checking, residual must be tested with predetermined fixed threshold value or an adaptive threshold value [6–8]. Diagnostic Bond Graph (DBG) method has been introduced in [12], so that residuals are directly obtained by making use of the current and few past measured sample data even when ARRs cannot be obtained in explicit symbolic form. DBG has been extended to Diagnostic Hybrid Bond Graph (DHBG) and adapted for FDI of hybrid systems [6, 8]. Few works related to the prognosis, which are based on BG approach, can be found in [13–16]. In [13, 14], it is assumed that degradation models of the faulty components are known beforehand and RUL estimation is performed by incorporating the degradation model into constrained ARRs equation. Further, it is assumed that a single component's parameter value continuously drifts with the evolution of time and all other components behave normally. In [15, 16], a BG framework is utilized for FDI but the RUL estimation is performed by using the Monte Carlo framework (particle filter technique). The existing model-based prognosis methods are intended for continuous systems and those cannot be easily applied to hybrid dynamical systems.

Components may degrade due to both internal stresses (load, torque, speed, etc.) and external stresses (wind, temperature, humidity, etc.). In order to accurately predict the RUL, it is necessary to take into account how and where the components will be used and what will be the mode of operation. According to domain knowledge about the considered system and its components, and known environmental and operational conditions, degradation model of the deteriorating components can be identified by understanding the physics of degradation in a model-based prognosis framework. A degradation model of a component may be obtained by accelerated life tests method and then that degradation model may be used to track the degradation of component once an incipient fault is detected by FDI module; whereupon only the coefficients of model need to be estimated and RUL can be predicted [14]. Thus, identification of precise dynamic degradation model and specification of a well-defined failure threshold for RUL estimation are the main challenges in model-based prognosis of hybrid system. In this regard, this chapter makes the following contributions:

- A unified sequential multiple fault diagnosis and prognosis method based on DHBG approach is developed for hybrid dynamical systems by introducing the concept of model updating after each fault identification. The proposed method is able to diagnose faults whose effect may be masked due to previously existing faults and also predicts the RUL of the faulty component if the isolated fault is of progressive type.
- RUL estimation uses the common framework, i.e., BG modelling approach, that has been used for system modelling, virtual prototyping, fault diagnosis rule

development, and parameter and system identification. Utilization of multiple degradation models for RUL estimation is suggested which include operational modes as additional control parameter and evolve through degradation model identification. Same DHBG model of the system as used in FDI module is used in a modified form to identify the degradation pattern of the components after detection of parametric fault. Models are continually evolved with time by adapting to the new information of the state of degradation of the monitored system to provide accurate RUL with bounded uncertainty value.

- Instantaneous fault sensitivity signature for both parametric and discrete faults is used for minimizing the set of suspected faults (SSF), which also provides the expected directions of corresponding parameter deviations. Accordingly, constrained parameter estimation is proposed for improving the diagnosis and prognosis tasks of hybrid system.
- The proposed approach can detect and isolate both parametric and discrete faults and can diagnose different types of fault like abrupt, incipient and progressive faults. However, the method may also detect and isolate the intermittent faults if there is a sufficient time window for parameter estimation. The proposed approach can also track the discrete mode transitions even in presence of one or more faults in a system.

6.2 Model-Based Diagnosis and Prognosis for Hybrid Systems

The performance of model-based diagnosis and prognosis (MBDP) approaches depend on the accuracy or quality of the model of the considered system. Models serve as knowledge representation of a large amount of structural, functional and behavioural information and their relationship. This knowledge representation is capitalized to create complex cause-effect reasoning leading to construction of powerful and robust automatic process supervision tools. The model development for a large complex system is a challenging task, especially for a hybrid dynamical system whose dynamical behaviour changes with the change of operating mode of the system. Assumptions taken during modelling, exclusion of minor dynamics and inclusion of major dynamics and the used modelling technique always affect the accuracy or quality of diagnosis and prognosis outcomes. If an appropriate model of a system which provides the expected behaviour of the real system in normal healthy condition is developed, then it can be utilized for the process supervision of the system. However, there is always some mismatch between the outputs of the behavioural model with the real system measurements even if there is no fault present in the real system. This generally happens because of modelling uncertainties, parameter uncertainties, unknown disturbances, measurement uncertainties, etc. For robust supervision, these factors must be taken into account. BG modelling is a good approach to deal with a multi-energy domain system which may have continuous and discrete dynamical behaviours with different uncertainties. Also, BG modelling can be used as a common framework for system modelling, virtual prototyping, fault diagnosis rule development, parameter and system identification, and RUL estimation.

Generally, BG model-based diagnosis is broadly classified into qualitative and quantitative approaches [7]. In qualitative model-based method, model considers the cause and effect relationships, and the functional relationships between the outputs and inputs of a system are represented in terms of qualitative functions. In quantitative model-based method, dynamic behaviour of a system is generally obtained from the first principles and the functional relationships between the outputs and inputs of a system are represented in terms of mathematical equations. However, for MBDP, quantitative method is preferred as it provides the common framework for both diagnosis and prognosis module development. Further, this quantitative method may be classified as observer-based, parity relation-based, parameter estimation-based and ARRs-based [8]. These methods can be used to generate the residuals which are the primary step in the process supervision of a system. Among these methods, ARR-based methods are more popular for the development of process supervision. ARRs are constraints expressed in terms of measurable process variables and nominal parameters of plant [7]. These constraints remain valid until a system operates according to its normal operation model. A fault is detected by monitoring the trend of the residuals. Quantitative ARRs-based methods can be further classified as symbolic and numerical methods. In symbolic methods, symbolic ARRs are obtained from BG model to evaluate the residuals; whereas, in numerical methods, residuals can be numerically evaluated using the DBG/DHBG model approach.

Quantitative BG model-based fault diagnosis method consists of two main steps: generation of residual or a DBG/DHBG model and evaluation of residual. The generation of residual is a technique for constructing ARRs using the BG model of the system. In the residual evaluation step, the trends of the residuals are interpreted to check any inconsistency. The inconsistency which indicates presence of one or more faults is detected by testing whether each residual is enveloped by a prescribed adaptive threshold, which in turn is defined based on the known uncertainties in parameter estimation, operating mode information and statistical parameters of measurement noise and unknown disturbances. Once any inconsistency in the residuals is found, the fault identification module is triggered to determine the severity of the fault and its nature/type. After information of severity of fault is obtained, the system may be reconfigured or fault may be accommodated. In case of incipient fault, RUL must be estimated to assists the plant technicians for planning the future maintenance activities. In the following sections, basic concepts in model-based diagnosis and prognosis are introduced.

6.2.1 Basic Framework of Fault Diagnosis

The global ARR, GARR($\boldsymbol{MD}$,$\boldsymbol{\theta}$, $\boldsymbol{U}$, $\boldsymbol{Y}$), of an uncertain hybrid system may be written as

$$\mathrm{GARR}_n \pm \lambda = 0 \tag{6.1}$$

where $\boldsymbol{MD}$ represents the controlled junction mode vector, $\boldsymbol{\theta}$ represents a known parameter vector, $\boldsymbol{U}$ represents known control or input vector and $\boldsymbol{Y}$ represents sensor output vector. Online evaluation of each nominal part, GARR$_n$, and uncertain part, λ, using $\boldsymbol{MD}$, $\boldsymbol{\theta}$, $\boldsymbol{U}$ and $\boldsymbol{Y}$ provides residual r and adaptive threshold $\varepsilon = |\lambda|$, respectively. The usual approach to detect fault induced inconsistencies is to test whether the numerical value of GARR$_n$ at corresponding mode remains bounded between $\pm\varepsilon$. Residual (r) and adaptive threshold (ε) may be evaluated directly from DBG [12] or DHBG model in linear fractional transformation (LFT) form [6–8]. A binary coherence vector ($\boldsymbol{C}$) is used to represent the signature for a fault; whose standard form is $\boldsymbol{C} = [c_1, c_2, \ldots, c_n]$ where c_i ($i = 1, 2, \ldots, n$) are obtained from a decision procedure, Θ, which is used to generate the alarms, i.e., $\boldsymbol{C} = [\Theta(r_1), \Theta(r_2), \ldots, \Theta(r_n)]$. For robust FDI, each residual r_i(t) is checked against the time varying adaptive threshold ε_i(t) as follows:

$$c_i = \Theta\left(r_i\right) = \begin{cases} 0, \text{ if} - \varepsilon_i(t) \le r_i(t) \le \varepsilon_i(t) \\ 1, \text{ otherwise} \end{cases} \tag{6.2}$$

During online monitoring, the coherence vector ($\boldsymbol{C}$) is obtained at each and every sampled time for consistency checking. An alarm is raised if one or more than one elements of the coherence vector show nonzero value, i.e., $\boldsymbol{C} \neq [0, 0 \ldots 0]$. After detection of fault, the coherence vector is matched with the fault signature matrix (FSM) at corresponding mode for isolation of actual fault candidate [7]. A fault in a component is detectable/monitorable, if at least one of the residual is sensitive to this fault, i.e., its monitorability index represented by $M_b = 1$. A fault in a component can be isolated only when it is monitorable and its fault signature (a corresponding row in FSM) is unique; which is represented by isolability index $I_b = 1$.

6.2.1.1 FSM, GFSM, GFSSM and MCSM, MCSSM

An FSM, $\boldsymbol{S}$, represents the relation between a set of parametric faults and their assumed signatures. It is used to detect and isolate the actual faults. The elements of FSM are either 1 or 0 as determined from

$$S_{ji} = \begin{cases} 1, \text{ if } r_i \text{ is a function of } P_j \\ 0, \text{ otherwise} \end{cases} \tag{6.3}$$

where r_i is the residual of the ith column, P_j is the parameter of the jth row in FSM, and $i = (1, 2 \ldots n)$, $j = (1, 2 \ldots p)$, n is number of residuals and p is number of parameters.

Hybrid system dynamics contains both continuous and discrete modes. So, FSMs for such system are mode dependent and need to be separately derived for each mode. Global fault signature matrix (GFSM), ***GS***, is derived in a global form [6, 8], whose elements are obtained from

$$GS_{ji} = \begin{cases} f(a_1, \ldots, a_m), & \text{if } r_i \text{ is a function of } P_j \text{ depending on the values } a_1, \ldots, a_m, \\ 1, & \text{if } r_i \text{ is a function of } P_j \text{ at all modes,} \\ 0, & \text{otherwise} \end{cases} \tag{6.4}$$

where f represents a logical function of controlled junction in a bond graph model and $a_1, \ldots, a_m$, represent the controlled junction state variables, and m is the number of controlled junctions.

Global fault sensitivity signature matrix (GFSSM) is an extension of GFSM, which has capability to differentiate between increasing (P_j ↑) and decreasing (P_j ↓) parametric fault. Its elements are updated at each and every instant by using the instantaneous sign of each residual sensitivity with respect to the component parameters [9]. The elements of the GFSSM, ***GSS***, are determined from

$$GSS_{ji} = \begin{cases} -\text{sign}(\partial r_i/\partial P_j), & \text{if } r_i \text{ is a function of } P_j \text{ and } P_j \text{ is expected to} \\ & \text{increase due to fault,} \\ +\text{sign}(\partial r_i/\partial P_j), & \text{if } r_i \text{ is a function of } P_j \text{ and } P_j \text{ is expected to} \\ & \text{decrease due to fault,} \\ 0, & \text{otherwise} \end{cases} \tag{6.5}$$

This new kind of signature is termed here sensitivity signature as it can be analysed by sensitivity theory and this residual sensitivity can be derived numerically by using Sensitivity Bond Graph (SBG) approach as in [17–19], where GARRs in closed symbolic form may or may not be derivable.

Residuals are also sensitive to discrete mode fault in a hybrid system and any inconsistency in actual mode may be identified by using mode change signature matrix (MCSM). The elements of the MCSM are determined from

$$\text{MCSM}_{ki} = \begin{cases} 1, \text{ if } r_i \text{ is a function of } a_k \\ 0, \text{ otherwise} \end{cases} \tag{6.6}$$

Mode change sensitivity signature matrix (MCSSM) is an extension of MCSM with capability to differentiate between of an increasing (a_k ↑) and decreasing (a_k ↓) mode fault. Its elements are updated at each and every instant by using the instantaneous sign of each residual sensitivity with respect to the mode [9]. The elements of the MCSSM are determined from

$$\text{MCSSM}_{ki} = \begin{cases} -\text{sign}(\partial r_i/\partial a_k), & \text{if } r_i \text{ is a function of } a_k \text{ and } a_k \text{ is expected} \\ & \text{to change from 0 to 1,} \\ +\text{sign}(\partial r_i/\partial a_k), & \text{if } r_i \text{ is a function of } a_k \text{ and } a_k \text{ is expected} \\ & \text{to change from 1 to 0,} \\ 0, & \text{otherwise} \end{cases} \tag{6.7}$$

where a_k represents the controlled junction state variables of the kth row of MCSM or MCSSM, and $a_k \in (0,1)$, $k \in (1, 2, \ldots m)$.

6.2.1.2 Adaptive Thresholds for Robust FDI

Adaptive thresholds are used to achieve robustness in FDI by accounting for the process and measurement uncertainties and also the mode transitions so that supervision system can minimize false alarms and misdetections. An uncertainty on a particular parameter value θ_j can be introduced as

$$\theta_j = \theta_{jn}(1 + \delta_{\theta_j}) \tag{6.8}$$

$$\text{or } \theta_j = \theta_{jn} + \varDelta\theta_j \tag{6.9}$$

where $\theta_j \in (I, C, R, TF, GY)$ corresponds to parameters associated with the model, and $\delta_{\theta_j} = (\varDelta\theta_j/\theta_{jn})$ and $\varDelta\theta_j$ are the relative and the absolute deviations of nominal parameter value θ_{jn}.

Adaptive thresholds using BG-LFT method [7, 8] can be used, in which system uncertainties in parameters are detached from their nominal parameters model and modelled as feedback loops of internal variables. For instance, when the real parameter value of a capacitance C is not accurately identified, it can be expressed as $C_n \pm \varDelta C = C_n(1 \pm \delta_C)$, where C_n is represented as nominal parameter value and $\pm\varDelta C = \pm\delta_C C_n$ is the uncertainty part of the parameter. If the C element is modelled in derivative causality, then its constitutive relation is given as

$$f = \frac{1}{C_n \pm \varDelta C}\dot{e} = \frac{1}{C_n}\left(1 \mp \delta_{1/C}\right)\dot{e} = \frac{\dot{e}}{C_n} \mp w_{1/C} \tag{6.10}$$

where $(\mp\delta_{1/C}/C_n)\dot{e} = \mp\, w_{1/C}$ is the extra contribution of flow because of uncertain part of parameter and may be treated as a disturbance. Note that $\delta_{1/C}$ is the uncertainty in estimating the value of $1/C$. The C element in derivative causality and parameter uncertainty can be modelled in BG-LFT form as given in Fig. 6.1.

Similarly, for the non-linear resistive R element modelled in conductive causality for representing the flow through a non-linear valve, the constitutive relation including uncertainty is written as

$$\begin{aligned} f &= \frac{1}{R_n \pm \varDelta R}\sqrt{e} = \frac{1}{R_n}\left(1 \mp \delta_{1/R}\right)\sqrt{e} \\ &= (C_d \mp \varDelta C_d)\sqrt{e} = C_d(1 \mp \delta_{1/R})\sqrt{e} = C_d\sqrt{e} \mp w_{1/R} \end{aligned} \tag{6.11}$$

Fig. 6.1 Capacitance (C) element in derivative causality modeled in LFT form

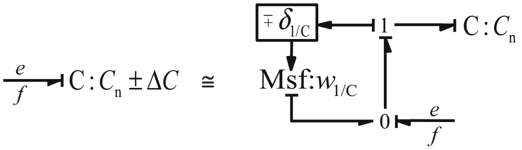
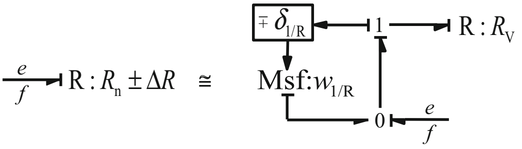


Fig. 6.2 Resistive (R) element in conductive causality modeled in LFT form



where $\delta_{1/R} = \Delta C_d / C_d$, e is the difference of pressure across the non-linear valve, C_d is the nominal value of discharge coefficient of valve, ΔC_d is the uncertainty part of the parameter C_d and $\mp w_{1/R} = \mp \delta_{1/R}\ C_d \sqrt{e}$ is the extra contribution of flow because of uncertain part of parameter and may be treated as a disturbance. The R element in conductive causality and parameter uncertainty can be modelled in BG-LFT form as shown in Fig. 6.2. Likewise, other parameters (I-element, and TF and GY two-ports) with uncertainties can be modelled [20].

6.2.2 Basic Framework of Prognosis

The term prognosis is often used in medical domain to describe the prediction of poor health of a patient by considering the actual diagnosis of one or more symptoms and their evolution compared with other similar observed cases. In industrial domain, the same reasoning of prognosis can be transposed to machines and components to answer the question about the RUL of a machine or a component once an impending failure condition is detected, isolated and identified by diagnosis module.

RUL, also called time to failure (TTF), is the time left before observing a failure of a component or subsystem given the current health status of system and its past degradation profile. Once the degradation trend of component's parameter is obtained, then that can be extrapolated with some set value of failure threshold to predict the RUL of faulty component [6].

$$\mathrm{RUL}(t, z) = t_{\mathrm{fl}} - t_0 |\ t_{\mathrm{fl}} > t_0,\ D(t) \tag{6.12}$$

where t_{fl} indicates the random variable of TTF, t_0 represents the current age of the component or system, $D(t)$ represents the past degradation profile up to the current time and z is the operating mode of the system.

Component's degradation may be captured by a continuous drift of its parameter value in its life cycle as reported in most of the existing literature on prognosis. However, in a hybrid dynamical system, various components or subsystems operate at different modes or environmental conditions which result in varying degradation rate of the components throughout the system's life cycle. Utilization of multiple degradation models which include operational modes as additional control parameter and evolve through degradation model identification is suggested in this chapter. Models are continually evolved with time by adapting to the new information of the state of degradation of the monitored system to provide accurate RUL with bounded uncertainty value. This overcomes the drawback of other similar models, where the parameters of the model are estimated only once and then the estimated degradation pattern is kept fixed irrespective of subsequent new available observations. RUL estimation and scheduling maintenance activities of component/subsystem based on single pre-identified degraded state are not an optimal solution, especially for hybrid systems that operate under variable modes. Intelligent prognosis must adapt according to change of the state of degradation of the constantly monitored system.

The proposed approach for RUL prediction requires simultaneous monitoring of both degradation and operating modes of the system. Let $\theta_j(t, z) \in \boldsymbol{\theta}$ be the parameter associated with jth component of a system which has started to degrade detectably at time t_0, as determined (detected and isolated) by FDI module. Figure 6.3 shows the different known operating modes (z) of the system and the jth component's (θ_j) degradation behaviour evolution in response to the operating mode changes. It is shown that the change points, denoted by $t^{(i)}$, (i = 1, 2,…)

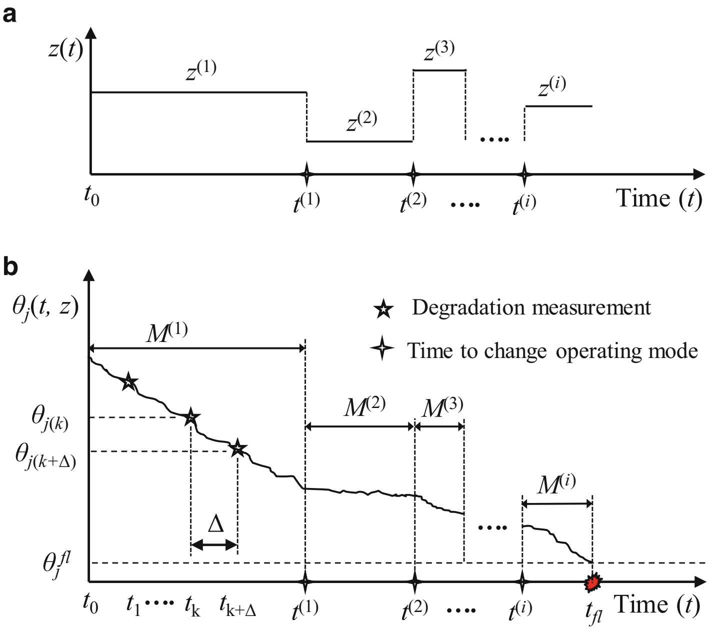


Fig. 6.3 (**a**) Different operating mode (**b**) degradation behaviour evolution at various operating mode of the component (θ_j)

of the operating mode are known and the operating mode is shifted to new mode $z^{(i)} \in z$ after $t^{(i)}$. After the shift, the state of degradation of the component changes at new mode $(z^{(i)})$ and degradation model is identified as $M^{(i)}$. Finally, the degradation curve of the component (θ_j) crosses a predefined failure threshold θ_j^{fl} at time t_{fl}. Thus the mode dependent RUL of the component (θ_j) as per defined performance (θ_j^{fl}) is estimated as

$$\mathrm{RUL}(t, z) = t_{\mathrm{fl}} - t_0 \tag{6.13}$$

Such a case of transition of operating condition from one mode to another is very common in hybrid dynamical systems. For instance, performance of an on–off valve which is generally used in processing plants may degrade because of fouling, which decreases the coefficient of discharge (C_d) of water flow. But the coefficient of discharge (C_d) does not continuously decrease at all times because of discrete operating nature of valve. When the valve is in open condition, fluid flow through the unit results in increased fouling such as due to sediment/lime scale deposition; but when the valve is in closed condition there is no such fouling. So, this operating condition pattern should be considered for RUL prediction.

6.2.2.1 An Integrated Framework of Diagnosis and Prognosis for Multiple Faults in Hybrid Dynamical System

Since diagnosis and prognosis both are concerned with the health monitoring of the industrial system, subsystems or components; it is reasonable to integrate them in a common framework for process supervision. Moreover, hybrid system contains both discrete and continuous dynamics; thus, discrete mode faults may occur in addition to parametric fault (abrupt or progressive type) and the occurrence of these faults are generally unknown in advance. In case of abrupt parametric fault or discrete fault, there is a step-like deviation in the corresponding component's parameter and it generally persists with the evolution of time. However in case of incipient or progressive fault, there is a slow change in the component's parameter with some dynamic degradation pattern which may be unknown beforehand. In abrupt or discrete fault, it is essential that the diagnosis scheme detects the faults quickly to avoid catastrophic consequences. In such cases, prompt fault detection and fault accommodation are the main aim of fault diagnosis. On the other hand, incipient faults are more significant in maintenance activities where it is necessary that slowly evolving faults are detected early enough to avoid more severe consequences. Once the nature of degradation pattern of incipient fault is obtained, RUL can be predicted by using the degradation model. In a large complex hybrid system, the occurrence of sequential multiple faults are much more likely, while the occurrence of simultaneous faults may be very rare and this is taken as a key assumption in the developments presented in this chapter. In this section, an integrated approach to BG-MBDP in a hybrid dynamical system for sequential multiple fault of unknown nature or type is proposed. The main goal of this section is to show how the same

DHBG model that is used for FDI of hybrid system can be further used to identify the degradation pattern of the faulty components and to estimate their RUL.

For estimation of component's degradation pattern in hybrid dynamical system, the fault magnitude of degrading parameter $\theta_j(t, z)$ is estimated with a fixed window of sample data collected at different time instants at various operating modes (z). As a result, a set of estimates of parameter values of degrading component is obtained at different time instances at various operating modes. Then degradation model, $M_{\theta_j}^{(i)}(\mathrm{z})$, which is best fit equation of parameter value evolution at any operating mode, is identified through curve fitting tool. Consequently, the obtained degradation models corresponding to various operating modes can be used for RUL estimation based on well-defined failure thresholds and known future operating modes of the component.

A complete flow chart of the proposed integrated MBDP for hybrid dynamical system is represented in Fig. 6.4. In a hybrid dynamical system, GARRs are used to detect any inconsistency in the nominal behaviour of the monitoring system at any mode. The general form of GARR which is obtained from DHBG model can be written as

$$\mathrm{GARR}_i = \mathrm{GARR}_i(\boldsymbol{MD}, \boldsymbol{\theta}, \boldsymbol{U}, \boldsymbol{Y}) \tag{6.14}$$

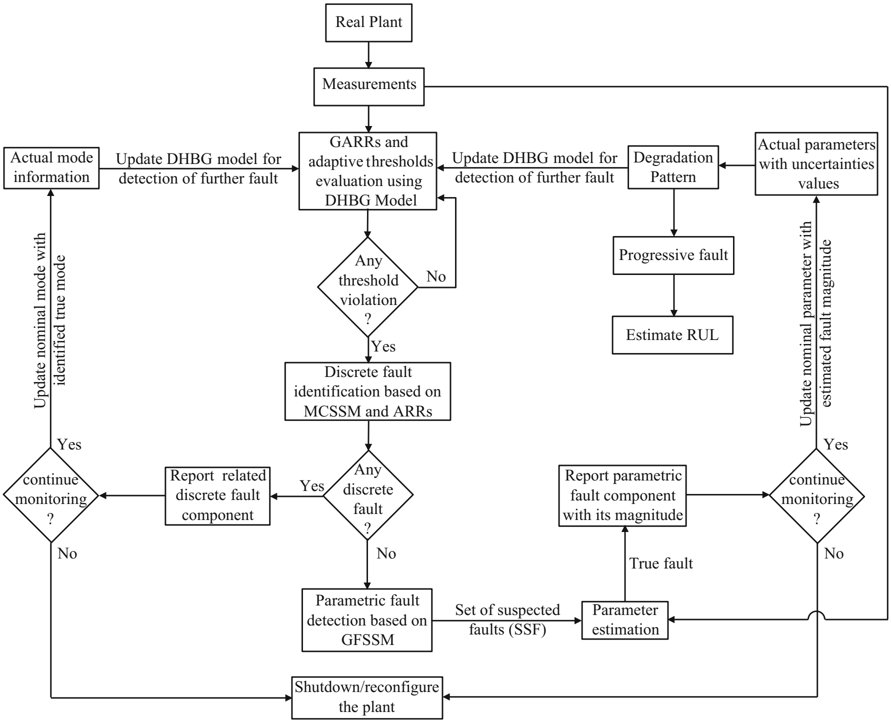

Real Plant
Measurements
Actual mode information
Update DHBG model for detection of further fault
GARRs and adaptive thresholds evaluation using DHBG Model
Update DHBG model for detection of further fault
Degradation Pattern
Actual parameters with uncertainties values
Any threshold violation ?
No
Yes
Progressive fault
Estimate RUL
Update nominal mode with identified true mode
Update nominal parameter with estimated fault magnitude
Discrete fault identification based on MCSSM and ARRs
Yes
continue monitoring ?
Report related discrete fault component
Yes
Any discrete fault ?
Report parametric fault component with its magnitude
continue monitoring ?
Yes
No
No
No
True fault
Parametric fault detection based on GFSSM
Set of suspected faults (SSF)
Parameter estimation
Shutdown/reconfigure the plant

Fig. 6.4 Flow chart of proposed integrated diagnosis and prognosis method

where $\boldsymbol{MD} = [a_1, a_2, \ldots, a_k, \ldots, a_m]^T$ represents the controlled junction mode vector, $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_j, \ldots, \theta_p]^T$ represents a known parameter vector which includes p number of component's nominal parameters, **U** represents known input vector and is formed by $\boldsymbol{U} \in (\boldsymbol{Sf}, \boldsymbol{Msf}, \boldsymbol{Se}, \boldsymbol{Mse})$ where $\boldsymbol{Sf}$ or $\boldsymbol{Msf}$ represents flow or modulated flow input vector and $\boldsymbol{Se}$ or $\boldsymbol{Mse}$ represents effort or modulated effort input vector, $\boldsymbol{Y}$ denotes sensor output vector and is formed by $\boldsymbol{Y} \in (\boldsymbol{Df}, \boldsymbol{De})$ where $\boldsymbol{Df}$ and $\boldsymbol{De}$ represent flow and effort sensor output vector, respectively, and $i \in (1, 2, \ldots n)$ is the number of residuals of the system.

Online evaluation of $\mathrm{GARR}_i = \mathrm{GARR}_i(\boldsymbol{MD}, \boldsymbol{\theta}, \boldsymbol{U}, \boldsymbol{Y})$ using $\boldsymbol{MD}$, $\boldsymbol{\theta}$, $\boldsymbol{U}$ and $\boldsymbol{Y}$ in normal operation may be written as

$$\mathrm{GARR}_{ni} \pm \lambda_i = 0 \tag{6.15}$$

where GARR_{ni} is nominal part and λ_i is the uncertain part of GARR_i, respectively, whose evaluations provide nominal residual r_{ni} and adaptive threshold $\varepsilon_i = |\lambda_i|$, respectively:

$$r_{ni}(t) = \mathrm{Eval}\,\{\mathrm{GARR}_{ni}(\boldsymbol{MD}, \boldsymbol{\theta}, \boldsymbol{U}, \boldsymbol{Y})\} \text{ and } \varepsilon_i(t) = \pm\mathrm{Eval}\,\{\lambda_i\} \tag{6.16}$$

During normal operation of the system, $\varepsilon_i(t) \geq r_i(t) \geq -\varepsilon_i(t)$ is satisfied. The residuals (r_{ni}) are sensitive to both parametric and discrete faults. Some of the residuals which are sensitive to a particular discrete or parametric fault in a system cross either upper or lower threshold when any type of the fault occurs. If any one of the nominal parameter (say θ_j) of θ changes (more than uncertainty value) or any one of the nominal mode of $\boldsymbol{MD}$ (say a_k) is inconsistent, then only a set of particular residuals which are sensitive to the change of this parameter θ_j or a_k cross the threshold in due time. When any threshold violation occurs, we initially hypothesize that this inconsistency is due to a discrete fault. If the sensitivity signature obtained corresponding to threshold violations (+1 for crossing upper threshold, −1 for crossing lower threshold and 0 for lying within thresholds) has unique match in MCSSM, then the component related to a_k is declared as a faulty one. If this discrete fault (a_k) cannot be isolated because more than one components share the same signature in MCSSM, then the ARR-based mode tracking [6] is followed in which all inconsistent ARRs are evaluated with each hypothesized mode fault and the actual faulty mode, if any, is identified as the one that gives consistent residuals. Then, the current discrete fault information of a_k is fed into the DHBG model to update the GARRs and adaptive thresholds for the isolation of subsequent faults. Once discrete fault is identified, we assume that it persists indefinitely thereafter if the plant is not allowed to shut down. The ARR-based mode tracking can also be used to track the initial mode of the system if it is unknown [6].

If the inconsistency in the residuals is not due to a particular discrete fault, i.e., the initial hypothesis could not be validated, then it is hypothesized that the inconsistency is due to a parametric fault only. If the sensitivity signature obtained corresponding to threshold violations has unique match in GFSSM corresponding to component (θ_j), then the FDI module detects the degradation of component (θ_j) (say at time t_0) and component (θ_j) is isolated as degrading component at mode $z = z^{(i)}$ (say). On the other hand, if the degrading component (θ_j) cannot be isolated because of more than one components share the same signature in GFSSM, then a set of suspected fault (SSF) is hypothesized for further course of action.

After postulation of faults candidates in SSF, a targeted parameter estimation technique is triggered for identification of true fault and its degradation pattern. This module tries to quickly estimate only a few possible parameters from which the true fault candidate (θ_j) can be isolated [7, 8, 17, 18, 21]. Now the parameter θ_j is finally declared as a degrading faulty parameter and is represented by $\theta_j(t, z^{(i)})$ at mode $z^{(i)}$. The estimated fault magnitude of degrading parameter $\theta_j(t, z^{(i)})$ at kth instant of time is represented by $\theta_j^f(k, z^{(i)})$. If the nominal part of each GARR$_{ni}$ is evaluated again with the estimated fault magnitude $\theta_j^f(k, z^{(i)})$ (real value of degrading parameter of plant/system at that instance), and adaptive thresholds are also updated, then the corresponding evaluated residuals do not cross the corresponding residual thresholds [8]. So, the original vector $\boldsymbol{\theta}$ is updated by replacing the nominal θ_j by the estimated fault magnitude, $\theta_j^f(k, z^{(i)})$ of the real plant. Now updated $\boldsymbol{\theta}$ at kth instant is assumed to be the new nominal parameter vector that is used to update DHBG model in LFT form to predict further degradation pattern of faulty component $\theta_j(t, z^{(i)})$. Note that a large deviation in a parameter value is considered as an abrupt fault. Likewise, any discrete (mode) fault is also treated as abrupt fault.

If the detected parametric fault (θ_j) is of progressive type, then the component's parameter $\theta_j(t, z^{(i)})$ is degrading slowly according to operating mode $z^{(i)}$. Thus, the same set of residuals which are sensitive to the change of this parameter $\theta_j(t, z^{(i)})$ would cross the adaptive threshold again after some more time. Again, parameter (θ_j) is estimated and its estimated value at $(k + \Delta)$th instant of time is represented by $\theta_j^f(k + \Delta, z^{(i)})$. Since up to the current time only two data points of fault magnitude of degrading parameter are known, a linear degradation model may be assumed for the initial estimation of RUL which alerts the maintenance engineer for scheduling the maintenance activities or other tasks. This initial linear degradation model is further adapted with modified model when more parameter estimates are obtained during monitoring. For accurate estimation of degradation pattern, sufficient number of data points should be obtained at particular mode $z^{(i)}$ and then the degradation model is identified by using the curve fitting tools at corresponding mode $z^{(i)}$. If the jth operating change point occurs at time instant $t^{(j)}$ corresponding to new mode $z^{(j)}$ during data collection for estimating the parameter value in previous operating mode $z^{(i)}$, then the fresh data of a fixed window size corresponding to new mode $z^{(j)}$ are collected and fault magnitude is estimated in this new mode $z^{(j)}$. This process is repeated until the true degradation model, $M_{\theta_j}^{(i)}(\mathrm{z})$, is obtained for parameter $\theta_j(t, z)$ at different operating modes. Then the obtained true model $M_{\theta_j}^{(i)}(\mathrm{z})$ is further used in prognosis module to predict RUL of faulty component (θ_j) with the future known operating mode by extrapolating the model $M_{\theta_j}^{(i)}(\mathrm{z})$. Also, the finally obtained true degradation model $M_{\theta_j}^{(i)}(z)$ of the component (θ_j) is fed into the DHBG model, so that the subsequent faults can be detected and isolated. In the case of sequential multiple progressive faults, RUL is estimated for each progressive fault component and the RUL of the component which has least value is the significant for the maintenance engineer.

6.3 Application to a Two-Tank Benchmark System

6.3.1 Description of a Two-Tank Hybrid System

The benchmark hybrid two-tank system is adapted from [11]. Its process and instrumentation diagram and hybrid bond graph (HBG) model are shown in Figs. 6.5 and 6.6, respectively. This system consists of two tanks (T_1 and T_2) that are connected with pipes and valves (V_1 and V_2). C_{dvi} is the coefficient of discharge of non-linear valve V_i, $i = 1, 2$. The liquid level in tank T_1 is regulated by a hydraulic pump which is controlled by a PI-controller installed in this system and tries to maintain water level of 0.5 m. The flow of pump (Q_P) is proportional to the output of the PI-controller (U_{PI}). Two drainage pipes (L_1 and L_2) with coefficients of discharge C_{dL1} and C_{dL2}, respectively, showing linear behaviour are also used. This system shows the hybrid dynamics and includes both autonomous and supervisory controller transition modes. Valve V_1 is switched to on and then off state according to command input given by the supervisory controller. When water level in tank T_1 exceeds level H_{L1} then the water starts flowing from tank T_1 to tank T_2 through drainage pipe L_1 (autonomous mode, a_1). Similarly, when water level in tank T_2 exceeds level H_{L2} then the water starts flowing from tank T_2 to atmosphere through

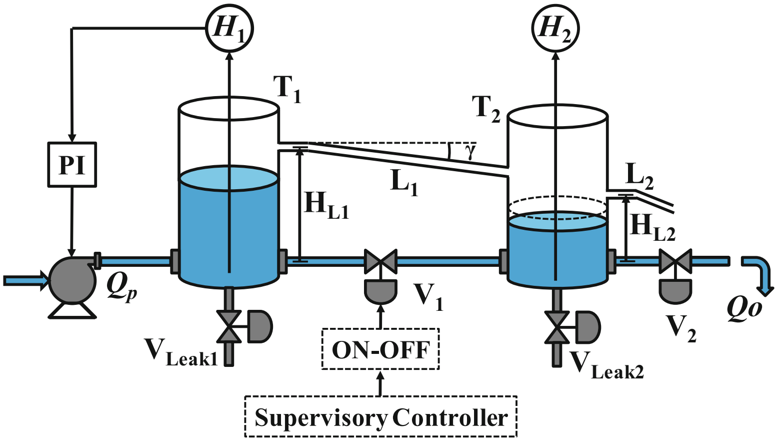


Fig. 6.5 Schematic diagram of a two-tank hybrid system

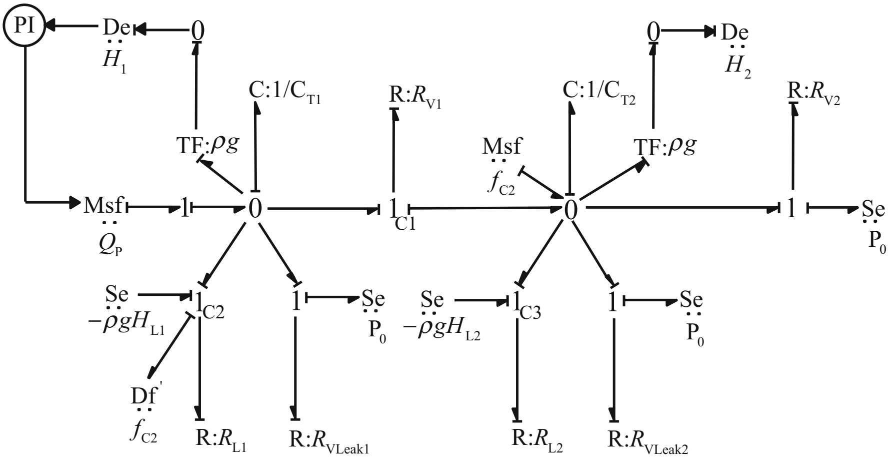


Fig. 6.6 HBG model of a two-tank hybrid system

drainage pipe L$_2$ (autonomous mode, a_2). Leakage fault can be introduced in tank T$_i$ using imaginary valve V$_{\text{Leak}i}$ having coefficient of discharge $C_{\text{dLeak}i}$, $i = 1, 2$.

Two level sensors H_1 and H_2 and one flow sensor Q_P are installed in the system for measuring the water levels in tanks T$_1$ and T$_2$ and water input flow by pump, respectively. The atmospheric pressure is assumed to be the reference pressure. The small angle γ, volume of water in the drainage pipe and inertia effect of water flow are neglected in this system; with the former two being considered as part of the uncertainties in tank capacities. This way of simplified model building by neglecting minor dynamics improves the speed of diagnosis without unduly complicating the process of development of the supervision system.

Pump saturation characteristic (Q_P) and PI-controller output law (Φ_{PI}) are, respectively, given as

$$Q_P = \begin{cases} U_{PI}, & 0 \le U_{PI} \le f_{max} \\ 0, & U_{PI} \le 0 \\ f_{max}, & U_{PI} \ge f_{max} \end{cases} = \Phi_P(U_{PI}) \tag{6.17}$$

$$\begin{aligned} U_{PI} &= K_P(S_{pt} - \rho \cdot g \cdot H_1(t)) + K_I \int (S_{pt} - \rho \cdot g \cdot H_1(t))dt \\ &= \Phi_{PI}(H_1(t)) \end{aligned} \tag{6.18}$$

where f_{max} is the maximum flow from the pump, S_{pt} is a pressure (or level) set point and K_P and K_I are the proportional and integral gains, respectively. In this work, we do not consider the actuator, controller and sensor faults; they may be diagnosed with additional hardware/sensor redundancies and hence are not relevant for this study.

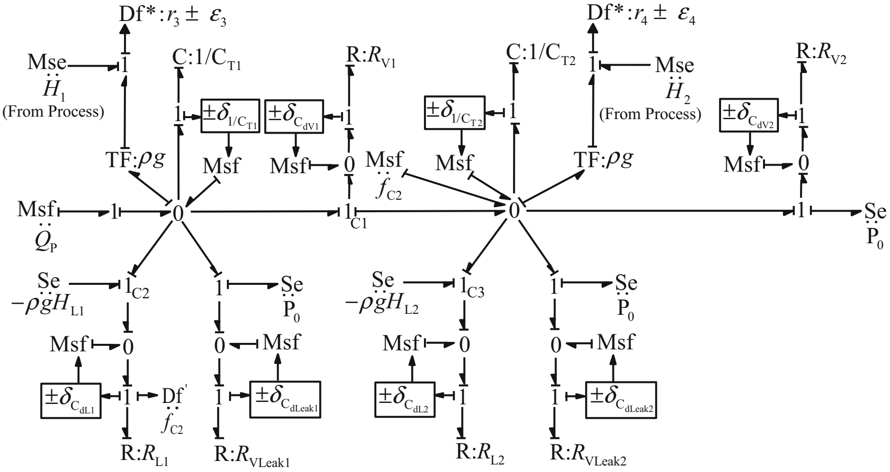


Fig. 6.7 DHBG-LFT model of a two-tank hybrid system

6.3.2 DHBG Model and GARRs Generation

Diagnostic hybrid bond graph (DHBG) model is obtained from HBG model of a system, in which all controlled junctions and storage elements are assigned with appropriate causalities so that all active BG elements remain active at all operating modes. This permits consistent causal description and generation of GARRs for a hybrid dynamical system from its DHBG model. The DHBG of the considered two-tank hybrid system in LFT form is shown in Fig. 6.7. Two virtual flow sensors (*Df**) are used to derive two constraints GARR$_3$ and GARR$_4$.

The ARRs for the actuators and the controllers (which will not be used in this study) are simply obtained from comparisons of input and output relationships as

$$\mathrm{ARR}_1 : Q_\mathrm{P} - \Phi_\mathrm{P}\left(U_\mathrm{PI}\right) = 0 \tag{6.19}$$

$$\mathrm{ARR}_2 : U_\mathrm{PI} - \Phi_\mathrm{PI}\left(H_1(t)\right) = 0 \tag{6.20}$$

$$\begin{aligned}\mathrm{GARR}_3 : Q_\mathrm{P} - C_\mathrm{T1}\cdot\frac{d}{dt}(\rho\cdot g\cdot H_1(t)) - a_\mathrm{v1}\cdot C_\mathrm{dv1}\cdot\sqrt{|\rho\cdot g\cdot(H_1(t)-H_2(t))|}\\ \cdot\mathrm{sign}(\rho\cdot g\cdot(H_1(t)-H_2(t))) - a_1\cdot C_\mathrm{dL1}\cdot\rho\cdot g\cdot(H_1(t)-H_\mathrm{L1})\\ -C_\mathrm{dLeak1}\cdot\sqrt{|\rho\cdot g\cdot H_1(t)|}\pm\lambda_3 = 0\end{aligned} \tag{6.21}$$

$$\begin{aligned}\mathrm{GARR}_4 : a_\mathrm{v1}\cdot C_\mathrm{dv1}\cdot\sqrt{|\rho\cdot g\cdot(H_1(t)-H_2(t))|}\cdot\mathrm{sign}(\rho\cdot g\cdot(H_1(t)-H_2(t))\\ +a_1\cdot C_\mathrm{dL1}\cdot\rho\cdot g\cdot(H_1(t)-H_\mathrm{L1}) - a_2\cdot C_\mathrm{dL2}\cdot\rho\cdot g\cdot(H_2(t)-H_\mathrm{L2})\\ -C_\mathrm{T2}\cdot\frac{d}{dt}(\rho\cdot g\cdot H_2(t)) - C_\mathrm{dv2}\cdot\sqrt{|\rho\cdot g\cdot H_2(t)|} - C_\mathrm{dLeak2}\cdot\\ \sqrt{|\rho\cdot g\cdot H_2(t)|}\pm\lambda_4 = 0\end{aligned} \tag{6.22}$$

where $C_{T1} = A_1/g$, $C_{T2} = A_2/g$, $a_1 = \begin{cases} 0, & H_1(t) \le H_{L1} \\ 1, & H_1(t) > H_{L1} \end{cases}$ and $a_2 = \begin{cases} 0, & H_2(t) \le H_{L2} \\ 1, & H_2(t) > H_{L2} \end{cases}$

GARR$_3$ and GARR$_4$, as presented in (6.21) and (6.22), respectively, contain the uncertain parts λ_3 and λ_4. The effects of uncertainties in various parameters on a GARR are un-correlated with possibility of cancelling out each other. Hence, absolute values of the individual effects are considered for adaptive threshold evaluation as follows:

$$\begin{aligned}\lambda_3 = {} & |\delta_{C1} \cdot C_{T1} \cdot \frac{d}{dt}(\rho \cdot g \cdot H_1(t))| + |a_{v1} \cdot \delta_{C_{dv1}} \cdot C_{dv1} \cdot \sqrt{|\rho \cdot g \cdot (H_1(t) - H_2(t))|}| \\ & + |a_1 \cdot \delta_{C_{dL1}} \cdot C_{dL1} \cdot \rho \cdot g \cdot (H_1(t) - H_{L1})| + |\delta_{C_{dLeak1}} \cdot C_{dLeak1} \cdot \sqrt{\rho \cdot g \cdot H_1(t)}|\end{aligned} \tag{6.23}$$

$$\begin{aligned}\lambda_4 = {} & |\delta_{C2} \cdot C_{T2} \cdot \frac{d}{dt}(\rho \cdot g \cdot H_2(t))| + |a_{v1} \cdot \delta_{C_{dv1}} \cdot C_{dv1} \cdot \sqrt{|\rho \cdot g \cdot (H_1(t) - H_2(t))|}| \\ & + |a_1 \cdot \delta_{C_{dL1}} \cdot C_{dL1} \cdot \rho \cdot g \cdot (H_1(t) - H_{L1})| + |a_2 \cdot \delta_{C_{dL2}} \cdot C_{dL2} \cdot \rho \cdot g \cdot (H_2(t) - H_{L2})| \\ & + |\delta_{C_{dv2}} \cdot C_{dv2} \cdot \sqrt{\rho \cdot g \cdot H_2(t)}| + |\delta_{C_{dLeak2}} \cdot C_{dLeak2} \cdot \sqrt{\rho \cdot g \cdot H_2(t)}|\end{aligned} \tag{6.24}$$

Using (6.4) and (6.6) on (6.21) and (6.22), the GFSM and MCSM for two-tank hybrid system are obtained as shown in Tables 6.1 and 6.2, respectively; whereas using (6.5) and (6.7) on (6.21) and (6.22), the GFSSM and MCSSM are found as shown in Tables 6.3 and 6.4, respectively. The parameters related to leakage fault in tank T$_1$ and T$_2$, i.e., C_{dLeak1} and C_{dLeak2}, respectively, have only increasing possibility (i.e., leakage), while for other parameters (i.e., C_{dv1}, C_{dv2}, C_{dL1}, C_{dL2}), both increasing (i.e., leakage) and decreasing faults (i.e., blockage) are possible. Likewise, discrete stuck on and off faults for a_{v1} of valve V$_1$ are also possible.

Table 6.1 GFSM (***GS***) for the two-tank hybrid system

Parameter	GARR$_3$ (r_3)	GARR$_4$ (r_4)	M_b	I_b Single fault
C_{dv1}	a_{v1}	a_{v1}	1	$\bar{a}_{v1} = (1 - a_1)a_{v1}$
C_{dv2}	0	1	1	0
C_{dL1}	a_1	a_1	a_1	$\bar{a}_1 = (1 - a_{v1})a_1$
C_{dL2}	0	a_2	a_2	0
C_{dLeak1}	1	0	1	1
C_{dLeak2}	0	1	1	0

Table 6.2 MCSM for the two-tank hybrid system

Parameter	GARR$_3$ (r_3)	GARR$_4$ (r_4)	M_b	I_b Single fault
a_{v1}	1	1	1	1

Table 6.3 GFSSM (***GSS***) for the two-tank hybrid system

Parameter	GARR$_3$ (r_3)	GARR$_4$ (r_4)
$C_{\mathrm{dv1}} \uparrow$	$a_{\mathrm{v1}}\mathrm{sign}(H_1(t) - H_2(t))$	$-a_{\mathrm{v1}}\mathrm{sign}(H_1(t) - H_2(t))$
$C_{\mathrm{dv1}} \downarrow$	$-a_{\mathrm{v1}}\mathrm{sign}(H_1(t) - H_2(t))$	$a_{\mathrm{v1}}\mathrm{sign}(H_1(t) - H_2(t))$
$C_{\mathrm{dv2}} \uparrow$	0	$+1$
$C_{\mathrm{dv2}} \downarrow$	0	-1
$C_{\mathrm{dL1}} \uparrow$	a_1	$-a_1$
$C_{\mathrm{dL1}} \downarrow$	$-a_1$	a_1
$C_{\mathrm{dL2}} \uparrow$	0	a_2
$C_{\mathrm{dL2}} \downarrow$	0	$-a_2$
$C_{\mathrm{dLeak1}} \uparrow$	$+1$	0
$C_{\mathrm{dLeak2}} \downarrow$	0	$+1$

Table 6.4 MCSSM for the two-tank hybrid system

Parameter	GARR$_3$ (r_3)	GARR$_4$ (r_4)
$a_{\mathrm{v1}} \uparrow$	$\mathrm{sign}(H_1(t) - H_2(t))$	$-\mathrm{sign}(H_1(t) - H_2(t))$
$a_{\mathrm{v1}} \downarrow$	$-\mathrm{sign}(H_1(t) - H_2(t))$	$\mathrm{sign}(H_1(t) - H_2(t))$

These possibilities, called technological specifications which are derived from deep knowledge of the system, are considered in Tables 6.3 and 6.4, respectively.

Note that the sensitivity signatures presented in GFSSM and MCSSM [9] are simplified expressions derived from GARRs. These can be numerically obtained from SBG model if GARRs cannot be symbolically derived. For example, if we consider the sensitivity signature element $GSS_{1,1}^{C_{\mathrm{dv1}}\uparrow}$ due to the fault $C_{\mathrm{dv1}} \uparrow$, then the corresponding element in the GFSSM is calculated as

$$\begin{aligned} GSS_{1,1}^{C_{\mathrm{dv1}}\uparrow} &= -\mathrm{sign}(\partial \mathrm{GARR}_3/\partial C_{\mathrm{dv1}}) \\ &= -\mathrm{sign}\big(-a_{\mathrm{v1}} C_{\mathrm{dv1}}\sqrt{|\rho g(H_1(t) - H_2(t))|}\,\big) \\ &= a_{\mathrm{v1}}\mathrm{sign}(H_1(t) - H_2(t)) \end{aligned} \tag{6.25}$$

6.3.3 Simulation Study and Results

In this section, an integrated MBDP method for sequential multiple faults of unknown nature/type in a hybrid dynamical system is tested through simulation. Also, model-based process supervision scheme using the most recent existing approach [9] which considers single fault hypothesis and the newly proposed approach which considers sequential multiple faults hypotheses, along with comparison between these two approaches are presented. This section also shows that the methods which are based on single fault hypothesis fail to predict the actual fault candidates in case of sequential multiple faults and without correct isolation of fault

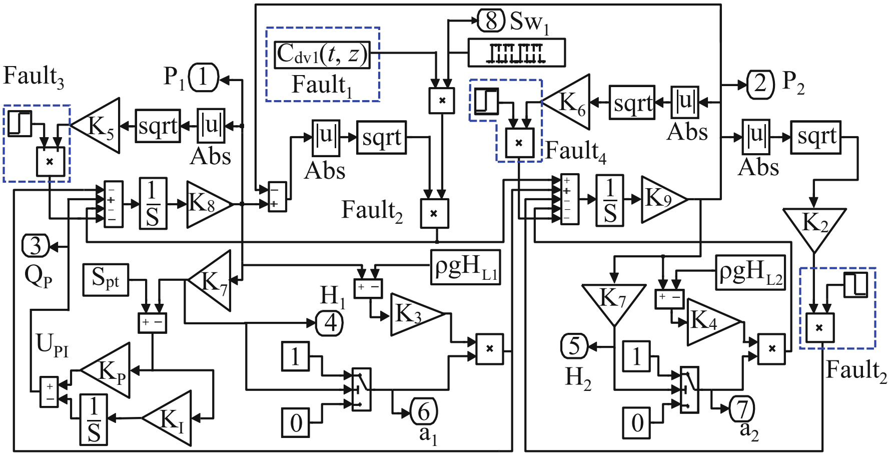


Fig. 6.8 Simulink model of two-tank system with provisions to introduce faults

candidates, prognosis is irrelevant. It is also shown that the newly proposed method gives improved fault isolation capabilities in both single and multiple fault scenarios and also improves the RUL estimation if the identified fault is of progressive type.

The HBG model of the two-tank hybrid system (shown in Fig. 6.6) is converted into MATLAB-SIMULINK model (with provisions of faults introduction in model as shown in Fig. 6.8) to generate the process data through simulation for the validation of the proposed method. In Fig. 6.8, $C_{\text{dv1}}(t,z)$ block represents the progressive fault function block for valve V$_1$ and $K_2 = C_{\text{dv2}}, K_3 = C_{\text{dL1}}, K_4 = C_{\text{dL2}}$, $K_5 = C_{\text{dLeak1}}$, $K_6 = C_{\text{dLeak2}}$, $K_7 = 1/\rho g$, $K_8 = g/A_1$, $K_9 = g/A_2$ are the corresponding gains blocks. Similarly, the DHBG-LFT model (shown in Fig. 6.7) can be converted into MATLAB-SIMULINK model for residuals and adaptive thresholds evaluation [11]. Here, the residuals and thresholds evaluation are done in a simple MATLAB program that evaluates, respectively, the nominal GARR$_3$ and GARR$_4$ in (6.21) and (6.22), and the uncertain parts λ_3 and λ_4 in (6.23) and (6.23). These, correspondingly, provide the residuals r_3 and r_4, and adaptive thresholds $\varepsilon_3 = \pm\lambda_3$ and $\varepsilon_4 = \pm\lambda_4$. Sensor's measurements data, parameter's nominal values and parameter's uncertainty values are the inputs to the program [7, 11]. The overall threshold may be evaluated to account for the sensor's noise as, $\varepsilon_i = \pm(\lambda_i + k_i)$, $i = 3, 4$, where k_i is the static threshold which is chosen based on the sensor characteristics or with the model of additive sensor biases presented in [8]. In the simulations, we have assumed $k_i = 0$.

The two-tank hybrid system is simulated for duration of 1800 s using a fixed step size of 0.02 s by initializing all state variables to zero. The nominal value of system's parameters used in the model are given in Table 6.5. Two types of faults are introduced in the simulation. The first one is a progressive fault in valve V$_1$ (see Table 6.6). The parameter C_{dv1} of on–off valve V$_1$ (nominal value = $1.593 \times 10^{-2}\,\text{kg}^{1/2}\,\text{m}^{1/2}$ in on state) is considered to drift slowly with time according to

Table 6.5 Nominal parameters of the hybrid two-tank

Symbol	Description	Nominal value
K_{P}	Proportional gain of controller	1 ms
K_{I}	Integral gain of controller	5×10^{-2} m
S_{pt}	Set point of the PI-controller	0.5 m
$f_{\max}$	Maximum outflow from pump	1 kg/s
A_{i}	Cross-sectional area of tank $\mathrm{T}_i (i = 1, 2)$	$1.472 \times 10^{-2}\,\mathrm{m}^2$
$C_{\mathrm{dv}i}$	Discharge coefficient of valve V_i including connected pipe $(i = 1, 2)$	$1.593 \times 10^{-2}\,\mathrm{kg}^{1/2}\,\mathrm{m}^{1/2}$
$C_{\mathrm{dL}i}$	Discharge coefficient of drainage pipe L_{i} $(i = 1, 2)$	1×10^{-3} ms
$C_{\mathrm{dLeak}i}$	Discharge coefficient of $\mathrm{V}_{\mathrm{Leak}i} (i = 1, 2)$	$0\,\mathrm{kg}^{1/2}\,\mathrm{m}^{1/2}$
H_{L1}	Height of the drainage pipe L_1 of tank T_1 from datum	0.58 m
H_{L2}	Height of the drainage pipe L_2 of tank T_2 from datum	0.40 m
P_0	Atmospheric pressure	$0\,\mathrm{N/m}^2$
ρ	Density of water	$1000\,\mathrm{kg/m}^3$
g	Acceleration due to gravity	$9.81\,\mathrm{m/s}^2$

Table 6.6 Simulated faults in the model

Parameter	Description	Degradation nature	Start time, t_{fi} (s)	End time
C_{dv1}	Valve blockage	Progressive type as per (6.26)	225	1800
C_{dv2}	Valve blockage	Abrupt type as per (6.27)	1475	1800

operating mode ($z = z^{(i)} = a_{\mathrm{v1}}$) as shown in Fig. 6.9b. Second one is an abrupt fault in valve V_2. In the simulation, we have introduced the fault in parameters C_{dv1} and C_{dv2} at a time instants $t_{\mathrm{f1}} = 225$ s and $t_{\mathrm{f2}} = 1475$ s, respectively, as defined through the functions given in (6.26) and (6.27), respectively.

$$C_{\mathrm{dv1}}(t, z) = \begin{cases} C_{\mathrm{dv1n}}(z) \cdot a_{\mathrm{v1}}, & t < t_{\mathrm{f1}} \\ C_{\mathrm{dv1n}}(z) \left(e^{-r(z) \cdot t_{\mathrm{on}}}\right) \cdot a_{\mathrm{v1}}, & t \geq t_{\mathrm{f1}} \end{cases} \tag{6.26}$$

$$C_{\mathrm{dv2}}(t, z) = \begin{cases} C_{\mathrm{dv2n}}(z), & t < t_{\mathrm{f2}} \\ 0.9 C_{\mathrm{dv2n}}(z), & t \geq t_{\mathrm{f2}} \end{cases} \tag{6.27}$$

where $C_{\mathrm{dv}i\mathrm{n}}(z)$ is the nominal parameter value of valve $\mathrm{V}_i (i = 1, 2)$ at corresponding operating mode $(z), r(z) = 1.0 \times 10^{-4}\,\mathrm{s}^{-1}$ at $z = z^{(1)} = a_{\mathrm{v1}} = 1$, with each $z^{(1)}$ for 80 s and $r(z) = 0\,\mathrm{s}^{-1}$ at $z = z^{(2)} = a_{\mathrm{v1}} = 0$, with each $z^{(2)}$ for 30 s, $t_{\mathrm{on}} = \int_{t_{\mathrm{f1}}}^{t} a_{\mathrm{v1}} \cdot dt$, and t_{f1} and t_{f2} are the time instances when the first and second faults start, respectively.

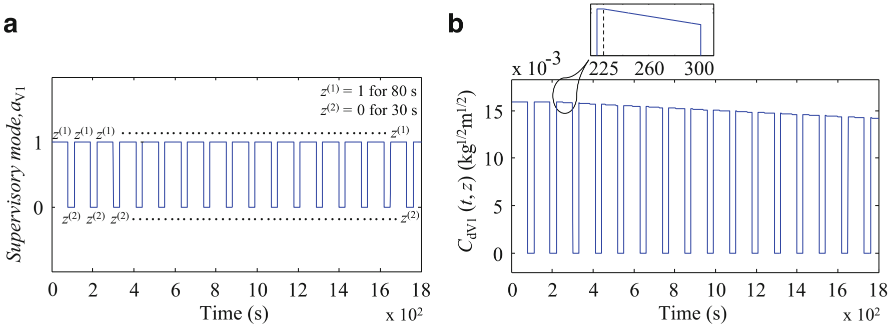


Fig. 6.9 (**a**) Operating mode (**b**) introduced degradation of on–off valve V_1

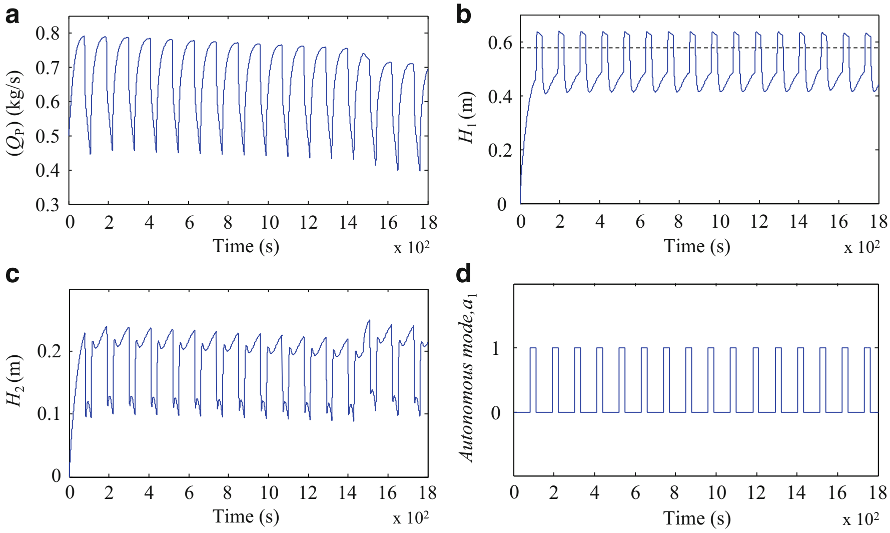


Fig. 6.10 Time responses of measurements (**a**) Pump flow (**b**) Water level in tank T_1 (**c**) Water level in tank T_2 and (**d**) Autonomous mode of drainage pipe L_1

Note that parameter C_{dv1} is considered to drift slowly only in on state of the valve, while there is no drift of parameter C_{dv1} during off state as the flow is zero in that state. The drifting of parameter C_{dv1} just after $t_{f1} = 225$ s is clearly shown in enlarged view of a portion of Fig. 6.9b. The failure threshold (C^{fl}_{dv1}) is considered as half of the nominal value in on state.

The measurements (i.e., Q_P, H_1 and H_2) from the simulated model at a sampling rate of 0.02 s are fed to the residuals and thresholds evaluation program. The time responses of measured input (Q_P), outputs (H_1, H_2) and obtained autonomous mode (a_1) are shown in Fig. 6.10. The transition of mode a_1 from 0 to 1 subjected to given conditions (when $H_1(t)$ exceeds $H_{L1} = 0.58$ m) corresponding to drainage pipe L_1 is clearly noticeable. However, no autonomous mode change is found for drainage pipe L_2; hence $a_2 = 0$ throughout the simulation period.

Table 6.7 GFSSM (***GSS***) for the two-tank hybrid system

Parameter	GARR$_3$ (r_3)	GARR$_4$ (r_4)	M_b	I_b Single fault
C_{dv1} ↑	a_{v1}	$-a_{v1}$	a_{v1}	$\bar{a}_{v1} = (1-a_1)a_{v1}$
C_{dv1} ↓	$-a_{v1}$	a_{v1}	a_{v1}	$\bar{a}_{v1} = (1-a_1)a_{v1}$
C_{dv2} ↑	0	+1	1	0
C_{dv2} ↓	0	−1	1	$\bar{a}_2 = (1-a_2)$
C_{dL1} ↑	a_1	$-a_1$	a_1	$\bar{a}_1 = (1-a_{v1})a_1$
C_{dL1} ↓	$-a_1$	a_1	a_1	$\bar{a}_1 = (1-a_{v1})a_1$
C_{dL2} ↑	0	a_2	a_2	0
C_{dL2} ↓	0	$-a_2$	a_2	0
C_{dLeak1} ↑	+1	0	1	1
C_{dLeak2} ↑	0	+1	1	0

Table 6.8 MCSSM for the two-tank hybrid system

Parameter	GARR$_3$ (r_3)	GARR$_4$ (r_4)	M_b	I_b Single fault
a_{v1} ↑	+1	−1	1	1
a_{v1} ↓	−1	+1	1	1

6.3.3.1 Implementation of Integrated Diagnosis and Prognosis Approach

In practice, the elements of GFSSM and MCSSM have to be updated at each and every instant by using the instantaneous measurement data. It is observed from Fig. 6.10 that the measurement H_1 is always greater than H_2, so the obtained GFSSM and MCSSM during the duration of observation (0–1800 s) for the considered two-tank hybrid system can be simplified to the forms given in Tables 6.7 and 6.8, respectively. The response of residuals (r_3 and r_4) and adaptive thresholds obtained from DHBG-LFT model using previously existing method [9] and the newly proposed method with dynamically updated parameter and updated adaptive thresholds are shown in Figs. 6.11 and 6.12, respectively (solid lines indicate residuals and dashed/dotted lines indicate adaptive thresholds). The response of residuals (r_3 and r_4) and adaptive thresholds during normal operation (up to $t = 225$ s) and during identification of degradation behaviour of valves V$_1$ and V$_2$ using proposed method (after $t = 225$ s) is clearly shown in Fig. 6.12. Note that the first two columns of the coherence vector related to actuator and controller faults, which are irrelevant in this study, have been dropped from the analysis.

A discussion on fault isolation capabilities using existing FDI methods considering single fault hypothesis is presented here. From the simulated faults listed in Table 6.6, both valves V$_1$ and V$_2$ are faulty between 1475 and 1800 s as the fault in valve V$_1$ is not repaired. The observed coherence vector (***C***) just after 225 s (after initiation of blockage fault in V$_1$) is obtained from Fig. 6.11a, b as $\boldsymbol{C} = [1\ 1]$ (if not considering residual sensitivity signature) or $\boldsymbol{C} = [-1 + 1]$ (if considering the residual sensitivity signature). According to coherence vector $\boldsymbol{C} = [1\ 1]$ or $\boldsymbol{C} = [-1+1]$, the set of suspected faults (SSF) is C_{dv1} and a_{v1} using GFSM/GFSSM

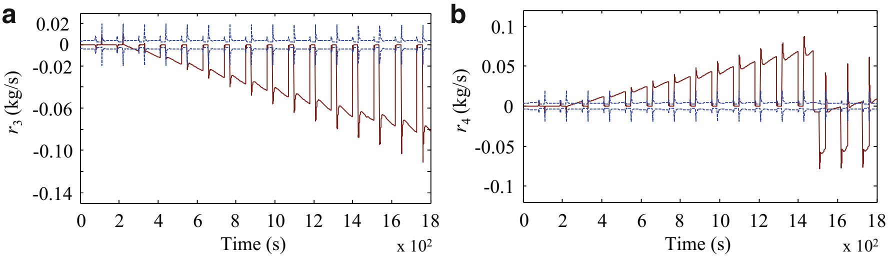


Fig. 6.11 Response of residuals (**a**) r$_3$ and (**b**) r$_4$ using previously existing method [9]

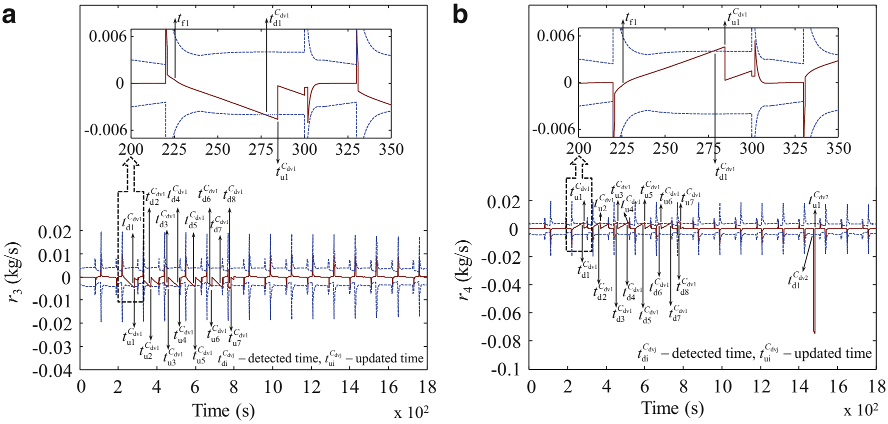


Fig. 6.12 Response of residuals (**a**) r$_3$ and (**b**) r$_4$ using proposed method with dynamically updated parameter and adaptive threshold

and MCSM/MCSSM at mode $a_1 = 0$; and if mode $a_1 = 1$, SSF is C_{dv1}, C_{dL1} and a_{v1} (see Tables 6.1, 6.2 and Tables 6.7, 6.8). In both modes, i.e., at mode $a_1 = 1$ and $a_1 = 0$, fault is non-isolatable as signature of C_{dv1}, C_{dL1} and a_{v1} are same. As the dynamics of valve V$_1$ is mode dependent, the residuals lie within the thresholds even after fault for $a_{\mathrm{v0}} = 0$ (See Fig. 6.11) and under such situation, fault cannot be detected and isolated until the system moves into a different mode ($a_{\mathrm{v1}} = 1$).

When the next fault in valve V$_2$ is introduced at 1475 s, the coherence vector is observed as $\boldsymbol{C} = [1\ 0]$ (if not considering residual sensitivity signature) and $\boldsymbol{C} = [-1\ 0]$ (if considering the residual sensitivity signature) Fig. 6.11a, b. According to coherence vector $\boldsymbol{C} = [1\ 0]$, possible SSF may be C_{dLeak1} at mode $a_{\mathrm{v1}} = 1$ and this lead to a wrong fault isolation. On the other hand, $\boldsymbol{C} = [-1\ 0]$ does not have any match in GFSSM and MCSSM and the fault is not isolated. While the fault is detected through both approaches (GFSM/MCSM and GFSSM/MCSSM) in this case of sequential multiple faults, the misdiagnosis is natural because the residuals are not diagonal or structured and thus, not suitable for multiple fault diagnosis [7]. Here, one fault effect hides the other fault effect that results in wrong fault signatures.

To identify the actual fault candidates in such cases, it is needed to estimate all the parameters simultaneously by using the non-linear parameter estimation techniques [7, 8, 17, 18, 21] and tracking the real mode using ARR-based mode identification technique [6]. However, this is a very tough and computationally complex task for a large complex system; especially considering the fact that prompt detection and true faults isolation are the main objectives in diagnosis task. Under the assumptions stated beforehand for sequential multiple faults, we need to estimate only a few postulated fault candidates or possible conflicts [17]. The parameter estimation can be done with least squares response matching or minimization of residuals (see [7, 8, 17, 18, 21] for details). Here, we will use least squares response matching approach for parameter estimation. This approach gives best performance when there are few parameters to estimate (as detected from GFSSM) and there are constraints on parameter values (expressed in terms of penalty functions added to the objective function to be minimized). For example, to estimate value of C_{dv1}, the parameter value is constrained between specified minimum (in this case 0) and maximum values. For the considered fault scenario (Table 6.6), GFSSM indicates a possible decrease in value of C_{dv1} and thus its value may be constrained between 0 and the earlier known value (nominal value). This way, the search zone is reduced. In addition, parameter estimation requires initial guess values of parameters. For discharge coefficient of valve V_1, it may be assigned as αC_{dv1}, where $0 < \alpha < 1$. In fact, value of α can be approximately obtained from the rate of change of residuals. The reduced search zone and closer guess values improve the convergence of the optimization process. The estimation procedure requires transient data after the fault. Therefore, a small delay amounting to a chosen window length is present between fault detection and its isolation. We assume that no more faults occur within this chosen window length after detection of a fault event.

A discussion on fault isolation capabilities using the new proposed integrated MBDP method, considering sequential multiple faults hypotheses using GFSSM and MCSSM, is presented here. When the simulated fault (progressive type) for valve V_1 is introduced at 225 s as shown in Fig. 6.9b, the coherence vector (for a short time just after 225 s) is obtained from Fig. 6.12a, b as $\boldsymbol{C} = [-1+1]$. According to coherence vector $\boldsymbol{C} = [-1 + 1]$, the possible SSF can be $C_{\mathrm{dv1}} \downarrow$ and $a_{\mathrm{v1}} \downarrow$ using GFSSM and MCSSM at mode $a_1 = 0$; otherwise, the possible SSF can be $C_{\mathrm{dv1}} \downarrow$, $C_{\mathrm{dL1}} \downarrow$ and $a_{\mathrm{v1}} \downarrow$ if mode $a_1 = 1$ (see Tables 6.7 and 6.8). In both cases, i.e., at mode $a_1 = 0$ and $a_1 = 1$, fault is non-isolatable as signature of $C_{\mathrm{dv1}} \downarrow$, $C_{\mathrm{dL1}} \downarrow$ and $a_{\mathrm{v1}} \downarrow$ are same. Since we assume the discrete mode fault occurs first, the ARR-based mode identification algorithm [6] is triggered. Whether the discrete mode fault $a_{\mathrm{v1}} \downarrow$ (valve V_1 stuck off) occurs or not is checked by evaluating all sensitive ARRs at current mode information and it is found that mode $a_{\mathrm{v1}} = 1$ is consistent. This indicates that the inconsistency is due to parametric faults and elements of SSF are refined as $C_{\mathrm{dv1}} \downarrow$ only at $a_1 = 0$, otherwise elements of SSF are $C_{\mathrm{dv1}} \downarrow$ and $C_{\mathrm{dL1}} \downarrow$ at $a_1 = 1$. Then the parameter estimation technique is triggered and $C_{\mathrm{dv1}} \downarrow$ is isolated as a true fault in both modes, i.e., $a_1 = 0$ and $a_1 = 1$. This blockage fault in valve $\mathrm{V}_1(C_{\mathrm{dv1}} \downarrow)$ is detected at time instant $t_{\mathrm{d1}}^{C_{\mathrm{dv1}}} = 277.68$ s at its on state $(z = z^{(1)} = a_{\mathrm{v1}} = 1$ and $a_1 = 0)$. There is some time delay between fault detection and its occurrence because of slow evolution of fault and the uncertainties included in residual threshold evaluation.

Since the direction of parameter variation of C_{dv1} is known by GFSSM (decrease in value of C_{dv1}), this information is used in constrained parameter estimation technique for degradation model identification. This makes degradation model identification fast, which is one of the main requirements in CBM. For the estimation of fault magnitude of C_{dv1}, a small time window of 7 s transient data after the fault has been selected. Now the fault magnitude of degrading parameter C_{dv1} is estimated after detecting the fault at time instant $t_{\mathrm{d1}}^{C_{\mathrm{dv1}}} = 277.68\,\mathrm{s}$. The estimated parameter value is known at time instant $t_{\mathrm{u1}}^{C_{\mathrm{dv1}}} = 277.68\,\mathrm{s} + 7\,\mathrm{s} = 284.68\,\mathrm{s}$ and is represented by $C_{\mathrm{dv1}}^{f}(t_{\mathrm{u1}}^{C_{\mathrm{dv1}}}, z^{(1)})$, whose value is $C_{\mathrm{dv1}}^{f}(t_{\mathrm{u1}}^{C_{\mathrm{dv1}}}, z^{(1)}) = 1.584 \times 10^{-2}\,\mathrm{kg}^{1/2}\,\mathrm{m}^{1/2}$. Now, the DHBG model is updated by replacing the nominal value $C_{\mathrm{dv1}} = 1.593 \times 10^{-2}\,\mathrm{kg}^{1/2}\,\mathrm{m}^{1/2}$ by the estimated fault magnitude $C_{\mathrm{dv1}}^{f}(t_{\mathrm{u1}}^{C_{\mathrm{dv1}}}, z^{(1)}) = 1.584 \times 10^{-2}\,\mathrm{kg}^{1/2}\,\mathrm{m}^{1/2}$ of the real plant. Now the residuals and adaptive thresholds evaluation obtained through updated nominal part of GARRs and updated uncertainties parts obtained from the updated DHBG model in LFT form force the residuals to lie within the updated adaptive thresholds (see Fig. 6.12 where this updated time instant is denoted as $t_{\mathrm{u1}}^{C_{\mathrm{dv1}}}$). Since, the parameter C_{dv1} is degrading progressively at mode $z = z^{(1)} = a_{\mathrm{v1}} = 1$, again a set of residuals (r_3 and r_4) which are sensitive to the change of this parameter, C_{dv1}, cross the adaptive threshold with the evolution of time (marked as $t_{\mathrm{d2}}^{C_{\mathrm{dv1}}} = 364.02\,\mathrm{s}$ in Fig. 6.12) and again the new fault magnitude of degrading parameter C_{dv1} is estimated and known at new time instant $t_{\mathrm{u2}}^{C_{\mathrm{dv1}}} = 364.02\,\mathrm{s} + 7\,\mathrm{s} = 371.02\,\mathrm{s}$ and represented by $C_{\mathrm{dv1}}^{f}(t_{\mathrm{u2}}^{C_{\mathrm{dv1}}}, z^{(1)})$, whose value is $C_{\mathrm{dv1}}^{f}(t_{\mathrm{u2}}^{C_{\mathrm{dv1}}}, z^{(1)}) = 1.575 \times 10^{-2}\,\mathrm{kg}^{1/2}\,\mathrm{m}^{1/2}$. Since only two data points are known up to the current time a linear degradation model is assumed for the initial estimation of RUL to alert the maintenance engineer. The obtained data points at different time instances ($t_{\mathrm{u1}}^{C_{\mathrm{dv1}}}$ and $t_{\mathrm{u2}}^{C_{\mathrm{dv1}}}$) corresponding to mode $z = z^{(1)} = a_{\mathrm{v1}} = 1$ are used to find the linear degradation mathematical model for the parameter C_{dv1} using the curve fitting tool in Matlab-Simulink and is shown in Fig. 6.13a. For curve fitting, t_{on} is used as a time reference in abscissa. Initial estimated value of RUL using the linear degradation model at on state of valve V_1 and no degradation at off state based on defined failure threshold ($C_{\mathrm{dv1}}^{\mathrm{fl}} = 0.5 C_{\mathrm{dv1}}$) and known future operating modes of valve V_1 is found as 6822.19 s (see Fig. 6.14a). This initial degradation model is further adapted with modified model when more new information of data points are obtained during monitoring. In Fig. 6.14, $t_{\mathrm{u1}}^{C_{\mathrm{dv1}}}$ is considered as a zero time reference for RUL estimation.

Note that at different time instances, the valve V_1 is switched to off state as per the command input given by the controller. In this duration, there is no flow through the valve V_1. Although valve V_1 is isolated as a faulty element, residuals r_3 and r_4 become zero (see Fig. 6.12) at the corresponding off state ($z = z^{(2)} = a_{\mathrm{v1}} = 0$) which provides $C_{\mathrm{dv1}}(t, z^{(2)}) = 0\,\mathrm{kg}^{1/2}\,\mathrm{m}^{1/2}$.

For accurate estimation of degradation pattern, sufficient data points should be used. As the monitoring is continued, newer information of degradation data is obtained. Estimated magnitude of degrading parameter C_{dv1} at the same mode

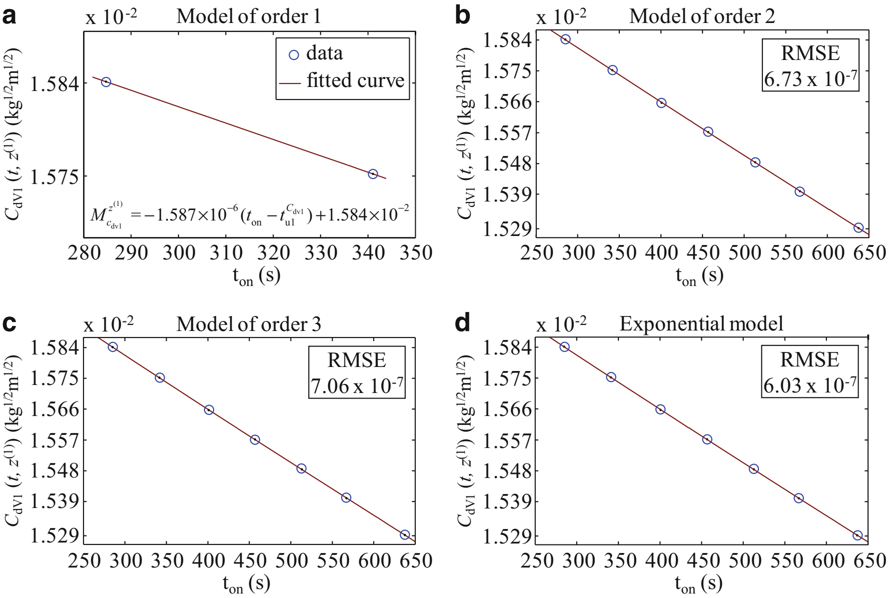


Fig. 6.13 (**a**) Linear degradation model at mode $a_{v1} = 1$ as two data points are known (**b**)–(**d**) Identification of degradation model at mode $a_{v1} = 1$ with new information of data points

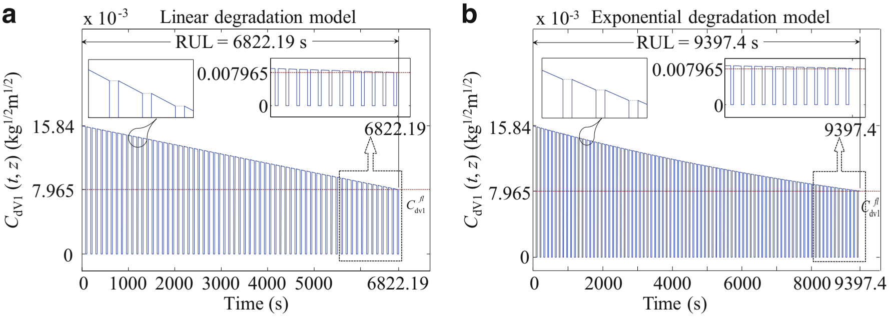


Fig. 6.14 Estimated RUL of parameter C_{dv1} (**a**) Initially assumed linear degradation model when only two data points are known (**b**) Finally confirmed exponential degradation model with the new information of degradation state at on state of the valve V$_1$ and no degradation at off state

$z = z^{(1)} = a_{v1} = 1$ at different time instances $t^{C_{dv1}}_{ui}$ $(i = 3, 4, \ldots, 7)$ are shown in Table 6.9. The new obtained data points at different time instances with previous known data points are used to refine the degradation behaviour of parameter C_{dv1}. Various degradation models of second order, third order polynomial fit and exponential fit, etc., are tried and shown in Fig. 6.13b–d along with root mean square error (RMSE) values. Exponential fit model is found as a best degradation model

Table 6.9 Estimated data points at different time instances

	Detected time (s)	Updated time (s)		Estimated magnitude
Parameter	$(t_{\mathrm{d}i}^{C_{\mathrm{dv}j}})$	$(t_{\mathrm{u}i}^{C_{\mathrm{dv}j}})$	t_{on} (s)	$C^f_{\mathrm{dv}j}(t_{\mathrm{u}i}^{C_{\mathrm{dv}j}}, z)$ ($\mathrm{kg}^{1/2}\,\mathrm{m}^{1/2}$)
$C_{\mathrm{dv1}} \downarrow$	277.68	284.68	284.68	1.584×10^{-2}
	364.02	371.02	341.02	1.575×10^{-2}
	453.56	460.56	400.56	1.566×10^{-2}
	509.50	516.50	456.50	1.557×10^{-2}
	595.78	602.78	512.78	1.548×10^{-2}
	679.76	686.76	566.76	1.539×10^{-2}
	780.04	787.04	637.04	1.529×10^{-2}
$C_{\mathrm{dv2}} \downarrow$	1475.02	1482.02	–	1.434×10^{-2}

at on state of the valve $\mathrm{V}_1(a_{\mathrm{v1}} = 1)$ according to its goodness of fit (RMSE $=$ 6.03×10^{-7}) which is the actual degradation pattern as per simulated fault. The obtained exponential model having coefficient a and b with 95 % confidence bound is represented as

$$M^{z^{(1)}}_{C_{\mathrm{dv1}}} = a \cdot \exp(b(t_{\mathrm{on}} - t_{\mathrm{u1}}^{C_{\mathrm{dv1}}})) \tag{6.28}$$

where; $a = 1.583 \times 10^{-2}$ and $b = -1.001 \times 10^{-4}$

Using the above degradation model RUL is estimated as 9397.4 s (see Fig. 6.14b) according to set failure threshold ($C^{\mathrm{fl}}_{\mathrm{dv1}} = 0.5C_{\mathrm{dv1}}$). Note that in the simulation, we have inserted a fast rate of progressive degradation, but in real situation the degradation rate may be very slow.

When the abrupt fault in valve V_2 (blockage) at 1475 s is inserted, the coherence vector ($\boldsymbol{C}$) just after 1475 s, i.e., after blockage fault initiation in V_2, is obtained from Fig. 6.12a, b as $\boldsymbol{C} = [0 - 1]$. This gives the possible SSF as $C_{\mathrm{dv2}} \downarrow$ (i.e., blockage fault in V_2) using GFSSM since $a_2 = 0$ at all times for the considered system. If the mode $a_2 = 1$ for some other system configuration, then the SSF would be obtained from GFSSM as $C_{\mathrm{dv2}} \downarrow$ and $C_{\mathrm{dL2}} \downarrow$; and parameter estimation technique can be used for the suspected parameters C_{dv2} and C_{dL2} to isolate the actual fault ($C_{\mathrm{dv2}} \downarrow$) and its magnitude. In the current configuration, the fault $C_{\mathrm{dv2}} \downarrow$ is directly isolatable at $a_2 =$ 0. Still we need to estimate the fault magnitude for updating the DHBG model for diagnosis of subsequent faults. In this case, $C_{\mathrm{dv2}} \downarrow$ is found as actual fault parameter with estimated fault magnitude of $C^f_{\mathrm{dv2}} \approx 0.9C_{\mathrm{dv2}}$ and this estimated parameter value is then onwards considered as the new nominal parameter value. Now the residuals and adaptive thresholds evaluation obtained through updated nominal part of GARRs and updated uncertainties parts obtained from the updated DHBG model in LFT form (i.e., C_{dv2} is replaced by C^f_{dv2}) force the residuals to lie within the updated adaptive thresholds (see Fig. 6.12b after time $t_{\mathrm{u1}}^{C_{\mathrm{dv2}}} = 1482.02$ s). Now the residuals (r_3 and r_4) remain inside the adaptive threshold until the next fault occurs. This way, subsequent parametric faults and their degradation pattern can be isolated and RUL can be estimated for progressive faults.

6.4 Equivalent Hybrid Electrical System

A great range of equivalent systems can be derived from a given system by using the equivalences of various BG elements and also transformer (*TF*) and gyrator (*GY*) equivalences of BG theory. A linear hydraulic system can be modeled by its analogous linear electrical system because of the similarities in their dynamics or governing differential equations. In the present work, an equivalent electrical system is modelled for the considered two-tank system and is further scaled down for easy and low cost experimental implementation. Note that while the equivalent electrical model is linear, the hybrid nature (mode dependent changes) are retained in it.

6.4.1 Circuit Layout

An equivalent electrical circuit of the two-tank hybrid system is shown in Fig. 6.15, whose HBG and DHBG models are similar to the two-tank hybrid system as shown in Figs. 6.6 and 6.7, respectively. In the electrical domain, power variables current and voltage, respectively, are equivalent to the flow rate and pressure in hydraulic domain. In the circuit, two electrical capacitors C_1 and C_2 are considered instead of two tanks T_1 and T_2, respectively. Resistors R_1, R_2 corresponding to valves V_1, V_2, and $R_{\mathrm{d}1}$, $R_{\mathrm{d}2}$ corresponding to drainage pipes L_1, L_2 are considered, respectively. Diodes D_1 and D_2 are considered as switches to permit the current in equivalent drainage pipe resistors $R_{\mathrm{d}1}$ and $R_{\mathrm{d}2}$ corresponding to set threshold voltages $V_{\mathrm{set}1}$ and $V_{\mathrm{set}2}$, respectively. $I_{\mathrm{d}1}$ is considered as a drainage current source flowing through resistor $R_{\mathrm{d}1}$ and it charges the capacitor C_2 only if the corresponding set condition is reached. A modulated current source I_{in} is used in place of modulated pump

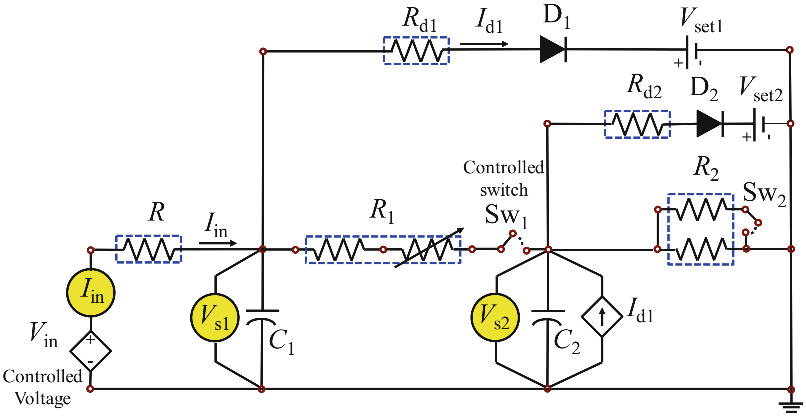


Fig. 6.15 Schematic diagram of equivalent electrical circuit of two-tank

flow Q_P. The modulated current source I_{in} is achieved by controlling the input voltage V_{in} which is the PI-controller's output and by using a known resistor R. The PI-controller, whose output is the modulated voltage V_{in}, is constructed by using various operational amplifiers (op-amps), resistors and capacitor combinations. The objective of PI-controller is to maintain a constant set voltage ($V_{set} = 5\,\mathrm{V}$) across the capacitor C_1 which is equivalent to level or pressure set point in tank T_1. A microcontroller (Arduino Uno) is used to open and close the relay switch Sw_1, which sets resistor R_1 at on and off state as per the command given by the controller. Switch Sw_2 is used to introduce the abrupt fault in resistors R_2 by using parallel connection of resistors where due to parallel connection, the resistance is low when the switch is on (nominal state) and high when it is off (simulated faulty state). Also, a series connection of a fixed resistor and a variable resistor is used to introduce the progressive fault in equivalent resistor R_1 at a particular instance according to its operating mode (see Table 6.13 for different time instances of introduced faults and their nature). A controlled servomotor is used to vary the resistance value of resistor R_1 at a particular rate at on state of resistor R_1. For controlling the servomotor, another microcontroller board (Arduino Uno) is used. The flow sensor (Q_P) and level sensors (H_1, H_2) in the hydraulic system are replaced by current sensor (I_{in}) and voltage sensors (V_{s1}, V_{s2}), respectively, in the electrical domain.

The PI-controller law as used for the modulated voltage source, V_{in}, and the modulated current source I_{in} are given as

$$V_{in}(t) = K_P(V_{set} - V_{s1}(t)) + K_I \int (V_{set} - V_{s1}(t))dt \tag{6.29}$$

$$I_{in}(t) = \frac{V_{in}(t) - V_{s1}(t)}{R} \tag{6.30}$$

The values of pump input flow Q_P and the pressures P_1, P_2 of linear hydraulic two-tank system at steady state were measured by doing a simulation in Matlab-Simulink and obtained as follow:

$$\begin{aligned} Q_P &= 0.8\,\mathrm{kg/s} \\ P_1 &= \rho \cdot g \cdot H_1 = 1000 \times 9.81 \times 0.5 = 4905\,\mathrm{N/m^2} \\ P_2 &= \rho \cdot g \cdot H_2 = 1000 \times 9.81 \times 0.25 = 2452.5\,\mathrm{N/m^2} \end{aligned} \tag{6.31}$$

A hydraulic-electrical analogy technique is used to determine the nominal parameters of the equivalent electrical circuit. For example, linear resistances R_1 and R_2 corresponding to the two valves V_1 and V_2 are obtained as

$$\begin{aligned} R_1 &= \tfrac{P_1 - P_2}{Q_P} = \tfrac{4905 - 2452.5}{0.8} = 3065.625\,\Omega \\ R_2 &= \tfrac{P_2}{Q_P} = \tfrac{2452.5}{0.8} = 3065.625\,\Omega \end{aligned} \tag{6.32}$$

In the similar manner, the rounded off values of all parameters of the equivalent electrical system are obtained as follows: $R = R_1 = R_2 = 3066\,\Omega$, $R_{\mathrm{d1}} = R_{\mathrm{d2}} = 1000\,\Omega$, $C_1 = C_2 = 1500\,\mu\mathrm{F}$.

The op-amps, capacitances, resistances and other electronic/electrical components purchased from the market come with uncertainties. Thus, the rounding-off errors may be considered as uncertainties in corresponding parameters.

6.4.2 Model Scaling

For conducting the experiment, the voltage and capacitance values were found to be out of range. For instance, as per (6.31), we need to operate the circuit around 5 kV electrical voltage range. Thus, a dual time and amplitude scaling was performed. For linear systems, the response $x(t)$ of the original system may be scaled as

$$x_s(t_s) = \alpha x(\beta t) \tag{6.33}$$

where α and β are the constants for amplitude and time scaling, respectively. We have considered $\alpha = 1/981$ which means the input current is reduced or equivalently, the voltage set point for PI-controller is set at $4905/981 = 5$ V. The hydraulic system has a slow response. We reduced the response time by $\beta = 10$ times. For that, the time constant ($\tau = RC$) was adjusted by retaining the value of R and reducing the value of C by ten times. The PI-controller's gains are adjusted to match with time constant of the modified system. The final scaled parameter values given in Table 6.10 are considered for developing an experimental setup.

Table 6.10 Nominal parameters of the scaled equivalent electrical circuit

Symbol	Description	Nominal value
K_{P}	Proportional gain of controller	0.3066
K_{I}	Integral gain of controller	1
V_{set}	Set point of the PI-controller	5 V
$I_{\mathrm{in(max)}}$	Maximum input electrical current	5 mA
C_i	Electrical capacitance value of ith capacitor ($i = 1, 2$)	150 μF
R_i	Electrical resistance corresponding to valve $R_i (i = 1, 2)$	3066 Ω
$R_{\mathrm{d}i}$	Electrical resistance corresponding to drainage pipe $R_{\mathrm{d}i} (i = 1, 2)$	1000 Ω
V_{set1}	Switch threshold voltage for R_{d1}	5.1 V
V_{set2}	Switch threshold voltage for R_{d2}	2.8 V

6.5 Experimental Study

An experimental setup of equivalent electrical hybrid system is shown in Fig. 6.16a. The circuit in the breadboard is the equivalent electrical hybrid system of the simulated two-tank hybrid system in reduced scale whose dynamics is nearly similar to the two-tank hybrid system dynamics. The schematic diagram of equivalent electrical circuit of two-tank hybrid system is already shown in Fig. 6.15. The various components of the circuit including PI-controller and the microcontroller are marked in Fig. 6.16b.

The experimental data were collected at a fixed sampling rate of 0.002 s from the current sensor I_{in} and voltage sensors $V_{\text{s}1}$ and $V_{\text{s}2}$ using a data-acquisition card (NI-USB6211). For computation of residuals, thresholds, mode identification and parameter estimation, LabVIEW-Matlab interface was used. The experimentally collected data were fed into the DHBG-LFT model of the electrical hybrid system for evaluation of residuals and adaptive thresholds. The GFSSM and MCSSM of the equivalent electrical system presented in Tables 6.11 and 6.12, respectively, are nearly same as that of the hydraulic two-tank system; only the corresponding parameter's nomenclature is changed. The coefficient of discharge parameter $C_{\text{d}i} \downarrow$ is considered equivalent to $1/R_i \downarrow$, where R_i is the linear resistance in electrical domain, i.e., the signature of $C_{\text{d}i} \downarrow$ is replaced by $R_i \uparrow$. Similarly, the signature of $C_{\text{d}i} \uparrow$ is replaced by $R_i \downarrow$.

As in the simulation, two types of faults were introduced in the experimental model. First one is a progressive fault in resistor R_1 as per Eq. (6.34) in its on state. The second one is an abrupt fault in resistor R_1 as per Eq. (6.35) (see Table 6.13). The failure threshold (R_1^{fl}) of resistor R_1 is considered as twice of its nominal value at on state for RUL estimation.

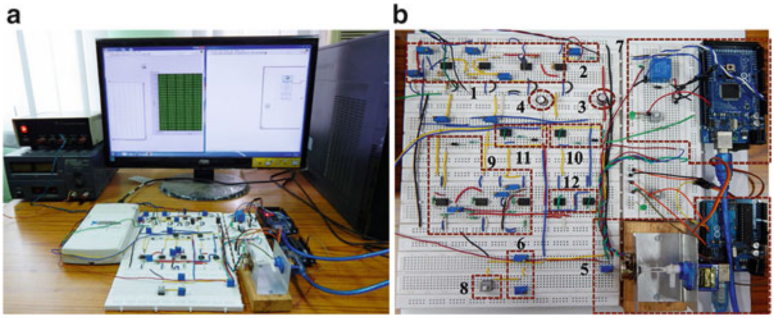


Fig. 6.16 (**a**) An experimental setup of electrical hybrid system (**b**) Enlarge view of breadboard showing different components within *dotted boxes*: (*1*) PI-controller, (*2*) R, (*3*) C_1, (*4*) C_2, (*5*) Variable resistor (R_1) with microcontroller, (*6*) R_2, (*7*) Relay switch (Sw_1) with microcontroller, (*8*) Sw_2, (*9*) $I_{\text{d}1}$, (*10*) D_1, $R_{\text{d}1}$ and $V_{\text{set}1}$, (*11*) D_2, $R_{\text{d}2}$ and $V_{\text{set}2}$, (*12*) Buffers

Table 6.11 GFSSM (***GSS***) for the electrical hybrid system

Parameter	GARR$_3$ (r_3)	GARR$_4$ (r_4)	M_b	I_b Single fault
$R_1 \downarrow$	a_{R1}	$-a_{\text{R1}}$	a_{R1}	$\bar{a}_{\text{R1}} = (1 - a_1)a_{\text{R1}}$
$R_1 \uparrow$	$-a_{\text{R1}}$	a_{R1}	a_{R1}	$\bar{a}_{\text{R1}} = (1 - a_1)a_{\text{R1}}$
$R_2 \downarrow$	0	$+1$	1	0
$R_2 \uparrow$	0	-1	1	$\bar{a}_2 = (1 - a_2)$
$R_{\text{d1}} \downarrow$	a_1	$-a_1$	a_1	$\bar{a}_1 = (1 - a_{\text{R1}})a_1$
$R_{\text{d1}} \uparrow$	$-a_1$	a_1	a_1	$\bar{a}_1 = (1 - a_{\text{R1}})a_1$
$R_{\text{d2}} \downarrow$	0	a_2	a_2	0
$R_{\text{d2}} \uparrow$	0	$-a_2$	a_2	0
$R_{\text{Leak1}} \downarrow$	$+1$	0	1	1
$R_{\text{Leak2}} \downarrow$	0	$+1$	1	0

Table 6.12 MCSSM for the electrical hybrid system

Parameter	GARR$_3$ (r_3)	GARR$_4$ (r_4)	M_b	I_b Single fault
$a_{\text{R1}} \uparrow$	+1	-1	1	1
$a_{\text{R1}} \downarrow$	-1	+1	1	1

Table 6.13 Introduced faults in the experimental model

Parameter	Description	Degradation nature	Start time, t_{fi}, (s)
R_1	Resistance increase	Progressive type as per (6.34)	323.5
R_2	Resistance increase	Abrupt type as per (6.35)	1249.7

$$R_1(t,z) = \begin{cases} R_{1\text{n}}(z) \cdot (1/a_{\text{R1}}), & t < t_{\text{f1}} \\ (R_{1\text{n}}(z) + k(z)t_{\text{on}}) \cdot (1/a_{\text{R1}}), & t \geq t_{\text{f1}} \end{cases} \tag{6.34}$$

$$R_2(t,z) = \begin{cases} R_{2\text{n}}(z), & t < t_{\text{f2}} \\ 1.2R_{2\text{n}}(z), & t \geq t_{\text{f2}} \end{cases} \tag{6.35}$$

where $R_{\text{in}}(z)$ is the nominal parameter value of resistor $R_i (i = 1, 2)$ at corresponding operating mode (z), $k(z) = 2.35\,\Omega/\text{s}$ if $z = z^{(1)} = a_{\text{R1}} = 1$, each $z^{(1)}$ is for 8 s and $k(z) = 0\,\Omega/\text{s}$ if $z = z^{(2)} = a_{\text{R1}} = 0$, each $z^{(2)}$ is for 3 s, $t_{\text{on}} = \int_{t_{\text{f1}}}^{t} a_{\text{R1}} \cdot dt$, and t_{f1} and t_{f2} are the time instances when the first and second faults start, respectively.

In a similar way as done in simulation (Sect. 6.3.3.1), sequential multiple faults are detected and isolated by the new proposed method using the experimental data of the electrical hybrid system and RUL is predicted for progressive faults. The evaluated residuals and adaptive thresholds with the real measurement data collected from the experimental setup are shown in Fig. 6.17. Note that due to discrete derivatives, residuals show spikes that appear during mode transitions. The observed coherence vector just after 323.5 s is $\boldsymbol{C} = [-1 + 1]$ which gives $R_1 \uparrow$ and $a_{\text{R1}} \downarrow$ as SSF elements at $a_1 = 0$; otherwise $R_1 \uparrow$, $R_{\text{d1}} \uparrow$ and $a_{\text{R1}} \downarrow$ are

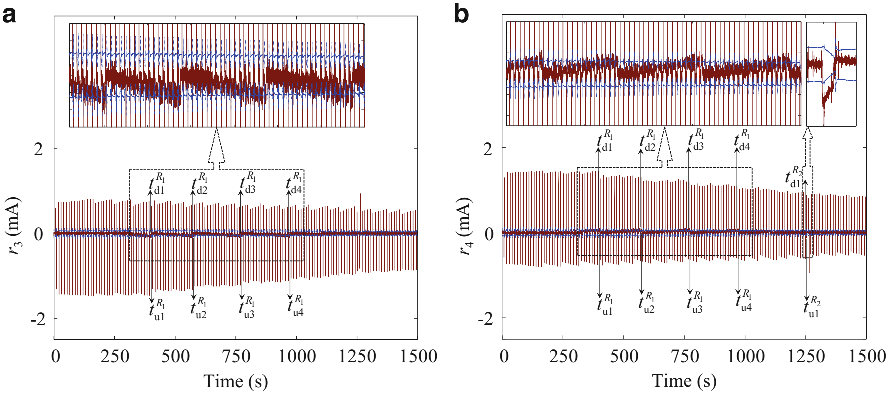


Fig. 6.17 Response of residuals (**a**) r$_3$ and (**b**) r$_4$ using proposed method with dynamically updated parameter and adaptive threshold

Table 6.14 Estimated data points at different time instances

Parameter	Detected time (s) ($t_{ui}^{R_j}$)	Updated time (s) ($t_{ui}^{R_j}$)	t_{on} (s)	Estimated magnitude $R_j^f(t_{ui}^{R_j}, z)$ (Ω)
R_1 ↑	399.768	402.768	402.768	3326.53
	565.572	568.572	523.572	3605.01
	749.380	752.380	656.380	3917.57
	938.692	941.692	794.692	4250.02
R_2 ↑	1249.702	1252.702	–	3740.5

SSF elements at $a_1 = 1$. Consistency in mode $a_{\mathrm{R1}} = 1$ is found by evaluating the all sensitive ARRs at current mode information which indicates the observed inconsistency is due to parametric fault. Upon parameter estimation and successive updation of DHBG model at various time instances, it is confirmed that the fault R_1 ↑ is of progressive nature (see Fig. 6.17 where residuals r_3 and r_4 are updated at various time instances during degradation pattern identification of R_1). The estimated magnitude of degrading resistance R_1 at the mode $z = z^{(1)} = a_{\mathrm{R1}} = 1$ at different time instances $t_{ui}^{R_1}$ $(i = 1, 2, 3, 4)$ is shown in Table 6.14.

Initially, the obtained data points at time instances $(t_{u1}^{R_1})$ and $(t_{u2}^{R_1})$ corresponding to mode $z = z^{(1)} = a_{\mathrm{R1}} = 1$ are used to find the linear degradation model for the resistor R_1 as presented in Fig. 6.18a. Initial estimated value of RUL using the linear degradation model at on state of resistor R_1 and infinite resistance value at off state is found as 1672.92 s (see Fig. 6.19a). This initial degradation model is further updated with modified model when newer information of parameter estimates are obtained during monitoring.

As the monitoring is continued, the new information of degradation data of resistor $R_1(t, z)$ is obtained via parameter estimation at time instances $t_{ui}^{R_1}$, $(i = 3, 4)$ at the same mode $z = z^{(1)} = a_{\mathrm{R1}} = 1$, which are presented in Table 6.14. Various

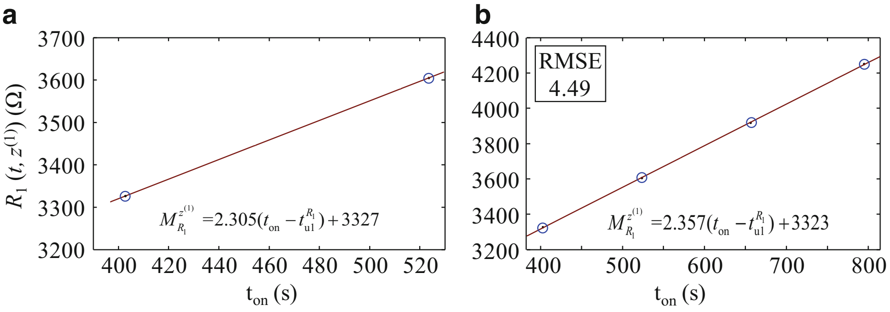


Fig. 6.18 (**a**) Linear degradation model at mode $a_{R1} = 1$ as two data points are known, (**b**) finally identified degradation model at mode $a_{R1} = 1$ with new information of data points

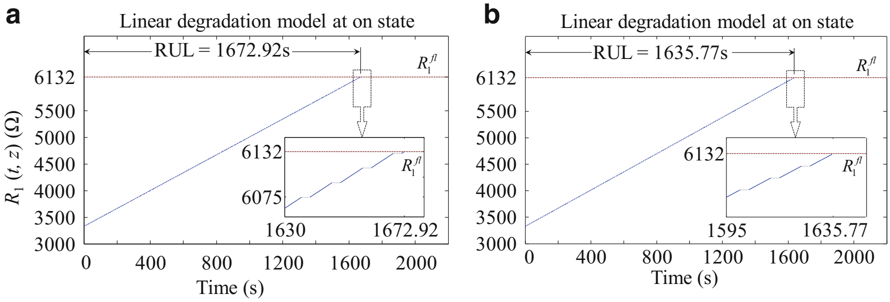


Fig. 6.19 Estimated RUL of parameter R_1 (**a**) With initially assumed linear degradation model when only two data points are known (**b**) with finally confirmed linear degradation model with the new information of degradation state at on state of resistor R_1, and considering infinite resistance in off state

degradation models are tried and linear degradation model was found as a best fit model for on state of the resistor $R_1(a_{\mathrm{R}1} = 1)$ according to goodness of fit (RMSE = 4.49). Indeed, this is the actual degradation introduced in the experiment. The final linear degradation model having coefficient P_1 and P_2 with 95 % confidence bound is given as

$$M_{R_1}^{z^{(1)}} = P_1(t_{\mathrm{on}} - t_{\mathrm{ul}}^{R_1}) + P_2 \tag{6.36}$$

where $P_1 = 2.357$ and $P_2 = 3323$.

Using the degradation model (6.36), the RUL is estimated as 1635.77 s (see Fig. 6.19b) according to set failure threshold ($R_1^{\mathrm{fl}} = 2R_1$).

When the abrupt fault in resistor R_2 at 1249.7 s is inserted, the coherence vector ($\boldsymbol{C}$) just after 1249.7 s, i.e., after fault initiation in resistor R_2, is obtained from Fig. 6.17a, b as $\boldsymbol{C} = [0-1]$. This gives the possible SSF as $R_2 \uparrow$ (i.e., increasing fault in resistor R_2) using GFSSM since $a_2 = 0$. Even though the fault $R_2 \uparrow$ is directly isolable at $a_2 = 0$, we still need to estimate the fault magnitude for updating the

DHBG model for diagnosis of subsequent faults. In this case, $R_2 \uparrow$ is found as actual fault parameter with estimated fault magnitude of $R_2^f \approx 1.2R_2$ at $(t_{\mathrm{u1}}^{R_2}) = 1252.702\,\mathrm{s}$ and this estimated parameter value is then onwards considered as the new nominal parameter value.

6.6 Conclusions

This chapter proposes a strategy for fault diagnosis and prognosis of hybrid dynamical system by using bond graph modelling as a common framework for system modelling, virtual prototyping, fault diagnosis rule development, parameter and system identification, and RUL estimation. The proposed approach detects and isolates sequential multiple faults of different types, i.e., discrete mode faults, abrupt and progressive parametric faults; and also predicts the RUL if the detected fault is due to progressive parameter drift. The developed method is first applied to a simulated benchmark problem and then it is experimentally validated on a scaled equivalent electrical circuit model.

The response time (time constant) of the system and the time taken for degradation model are the two critical parameters that govern the applicability of this diagnosis and prognosis scheme. For processes or systems with slow response time, the time spent for parameter estimation is not significant. However, for fast systems like the equivalent electrical system considered in this study, the parameter estimation needs to be faster. This is achieved by narrowing down the number of suspected faults and using sensitivity signature for information regarding fault direction. Global fault sensitivity signatures of the hybrid dynamical system are used to identify the possible fault directions from the residual responses. The information of operating mode, possible fault candidates and directions of corresponding parameter deviations are used in a constrained least square error minimization-based estimation of fault parameters. It was found that richer information gleaned from sensitivity signatures allows quicker and reliable identification of faults and degradation model for RUL estimation.

For RUL estimation, use of multiple degradation models which include operational modes as additional control parameter and evolve through degradation model identification is considered in this study for hybrid dynamical system. Models are continually evolved with time by adapting to the new information of the state of degradation of the monitored system to provide accurate RUL with bounded uncertainty value. This overcomes the drawbacks of various existing methods.

References

1. Vachtsevanos, G., Lewis, F. L., Roemer, M., Hess, A., & Wu, B. (2006). *Intelligent fault diagnosis and prognosis for engineering systems*. Hoboken: Wiley.
2. Mukherjee, A., Karmakar, R., & Samantaray, A. K. (2006). *Bond graph in modelling, simulation and fault identification*. New Delhi: I. K. International Pvt. Ltd.

3. Borutzky, W. (2010). *Bond graph methodology: Development and analysis of multidisciplinary dynamic system models*. London: Springer.
4. Mosterman, P. J., & Biswas, G. (1995). Behavior generation using model switching: A hybrid bond graph modelling technique. *Transactions of The Society for Computer Simulation, 27*(1), 177–182.
5. Roychoudhury, I., Daigle, M. J., Biswas, G., & Koutsoukos, X. (2010). Efficient simulation of hybrid systems: A hybrid bond graph approach. *Simulation: Transactions of the Society for Modeling and Simulation International, 87*(6), 467–498.
6. Wang, D., Yu, M., Low, C. B., & Arogeti, S. (2013). *Model-based health monitoring of hybrid systems*. New York: Springer.
7. Samantaray, A.K., & Bouamama, B. O. (2008). *Model-based process supervision: A bond graph approach*. London: Springer.
8. Borutzky, W. (2015). *Bond graph model-based fault diagnosis of hybrid systems*. Switzerland: Springer.
9. Levy, R., Arogeti, S., Wang, D., & Fivel, O. (2015). Improved diagnosis of hybrid systems using instantaneous sensitivity matrices. *Mechanism and Machine Theory, 91*, 240–257.
10. Narasimhan, S., & Biswas, G. (2007). Model-based diagnosis of hybrid systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, 37*(3), 348–361.
11. Ghoshal, S. K., Samanta, S., & Samantaray, A. K. (2012). Robust fault detection and isolation of hybrid systems with uncertain parameters. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 226*(8), 1013–1028.
12. Samantaray, A. K., Medjaher, K., Bouamama, B. O., Staroswiecki, M., & Dauphin-Tanguy, G. (2006). Diagnostic bond graphs for online fault detection and isolation. *Simulation Modelling Practice and Theory, 14*, 237–262.
13. Medjaher, K., & Zerhouni, N. (2009). Residual-based failure prognostic in dynamic systems. In *The 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes Held in Barcelona, Spain* (pp. 716–721).
14. Medjaher, K., & Zerhouni, N. (2013). Hybrid prognostic method applied to mechatronic systems. *International Journal of Advanced Manufacturing Technology, 69*(1–4), 823–834.
15. Ming, Y., Wang, D., & Luo, M. (2015). An integrated approach to prognosis of hybrid systems with unknown mode changes. *IEEE Transactions on Industrial Electronics, 62*(1), 503–515.
16. Jha, M. S., Dauphin-Tanguy, G., & Bouamama, B. O. (2016). Particle filter based hybrid prognostics for health monitoring of uncertain systems in bond graph framework. *Mechanical Systems and Signal Processing, 75*, 301–329.
17. Samantaray, A. K., & Ghoshal, S. K. (2007). Sensitivity bond graph approach to multiple fault isolation through parameter estimation. *Proceedings of the Institution of Mechanical Engineers. Part-I: The Journal of Systems and Control Engineering, 221*(4), 577–587.
18. Gawthrop, P. J. (2000). Sensitivity bond graphs. *Journal of the Franklin Institute, 337*(7), 907–922.
19. Borutzky, W. (2011). Incremental bond graphs. In *Bond graph modelling of engineering systems* (pp. 135–176). New York: Springer.
20. Merzouki, R., Samantaray, A. K., Pathak, P. M., & Bouamama, B. O. (2012). *Intelligent mechatronic systems: Modeling, control and diagnosis*. New York: Springer Science and Business Media.
21. Low, C. B., Wang, D., Arogeti, S., & Luo, M. (2009). Fault parameter estimation for hybrid systems using hybrid bond graph. In *Proceedings of the 3rd IEEE Multi-Conference on Systems and Control, Saint Petersburg, Russia*, July 8–10 (pp. 133–134).

Chapter 7
Particle Filter Based Integrated Health Monitoring in Bond Graph Framework

Mayank S. Jha, G. Dauphin-Tanguy, and B. Ould-Bouamama

7.1 Introduction

Besides the abrupt faults that have been considered in the previous chapters, incipient system faults and degradations of the system parameters pose significant hurdles in efficient maintenance of the system. For example, fatigue enabled wear in turbine blades, incipient leakage in valves of process engineering systems, friction induced jamming of rod in aircraft actuators, etc., pose great threat to system reliability and safety. Such problems are efficiently resolved when addressed under the realm of the so-called *condition based maintenance* (CBM) and *prognostics and health management* (PHM) [34]. The latter represent a *predictive* maintenance philosophy that has emerged only recently on contrary to the traditional strategies based upon *preventive* and *corrective* maintenance.

The main feature of CBM is the consideration of the "actual" condition of system component for designing maintenance actions rather than on an elapsed time or running hours' basis. Thus, CBM primarily depends upon current assessment of system health or state and involves real time data monitoring and processing. The two basic aspects of CBM are *diagnostics* and *prognostics.* As seen in the previous chapters, *Diagnostics* involves detection of fault and thereby, identification and quantification of the root cause of a problem. *Prognostics* involves prediction of the future health of the equipment either before or after a problem occurred [34,

M.S. Jha (✉)
Ecole Centrale de Lille, Toulouse, France
e-mail: mayank-shekhar.jha@centralelille.fr; jha.mayank.jha@gmail.com

G. Dauphin-Tanguy
Ecole Centrale de Lille, Villeneuve-d'Ascq, France

B. Ould-Bouamama
Ecole Polytech de Lille, Villeneuve d Ascq cedex, France

W. Borutzky (ed.), *Bond Graphs for Modelling, Control and Fault Diagnosis of Engineering Systems*, DOI 10.1007/978-3-319-47434-2_7

60]. As stated in [33], *prognostics* is "*estimation of time to failure and risk for one or more existing and future failure modes*."

The *Remaining Useful Life* (RUL) becomes a reliable estimate of the time to failure; it denotes how long system can function safely/reliably and within the prescribed limits of system functionalities. Thus, assessment of RUL involves predictions in future. In this context, the major motivation remains in providing sufficient lead-time between detection of a fault (diagnostic step) and occurrence of the system/component failure so that *pro-active* maintenance actions can be strategized in advance [62].

RUL prediction is not a trivial task as it involves future predictions which not only require precise information of current health, but also remain sensitive to various types of uncertainties to a large degree. These uncertainties involve stochastic evolution of incipient degradations, failure modes, varying operational conditions, measurement noise, etc. In face of all such uncertainties, the prognostic procedure must be able to accurately assess the rapidity of system degradation till failure and novel events that may significantly influence the assumed/learnt degradation trend. Due to inherent stochastic phenomena and uncertainty involved, evaluation of confidence on RUL predictions is given a significant weightage. In fact, several business decisions are based upon confidence limits associated with RUL predictions rather than the specific value of RUL itself [59]. In essence, determination of accurate and precise RUL estimate forms the core objective of any prognostics procedure.

On the other hand, the term PHM describes the systems that implement a CBM philosophy [62]. However, in the context of PHM, *prognostics* gains a wider meaning encompassing the tasks of fault detection, fault-identification, current health assessments, performance monitoring, and RUL predictions [34]. Thus, diagnostics and prognostics form building blocks of any CBM enabled PHM architecture. When these two essential tasks are achieved in an integrated manner, such a common paradigm may be given the designation of integrated health monitoring framework [9, 35].

In BG framework, diagnostics and prognostics task can be achieved in an integrated way by exploiting the properties of Analytical Redundancy Relations (ARRs) and their numerical evaluations or residuals. In this context, due to deterministic nature of ARRs, most of the existing works have neglected the inherent randomness in damage progression [20, 23, 48, 49], which in turn has led to RUL predictions that do not incorporate associated uncertainties and inherent stochasticity.

This chapter details ARR based integrated health monitoring methodology where the benefits of *BG in Linear Fractional Transformations* (BG-LFTs) have been integrated with advantages of Bayesian inference techniques to obtain accurate and precise estimate of parametric health in probabilistic domain. The inherent randomness in degradation progression is effectively managed by using sequential Monte Carlo based particle filters (PF) for estimation of state of a system parameter and subsequent RUL prediction in probabilistic domain.

After this introduction, Sect. 7.2 details various approaches of prognostics, BG-LFT method, and non-linear Bayesian inference technique using PFs. Section 7.3

discusses degradation models (DM). The method of prognostics is described in the next section. Sections 7.4 and 7.5 discuss the integrated health monitoring strategy and evaluation metrics, respectively. Section 7.6 details the application of methodology on a mechatronic system in real system. Section 7.7 draws conclusions.

7.2 Background and Techniques

This section discusses different techniques of prognostics. Moreover, BG-LFT technique of modelling uncertain systems and associated fault detection technique is discussed briefly. The latter is employed for detection of degradation initiation for the integrated health monitoring purposes. Additionally, non-linear Bayesian filtering using particle filters (PF) is described as it plays a significant role in the prognostics method presented in this chapter.

7.2.1 Approaches of Prognostics

Last decade has witnessed an extensive surge in development of various prognostics techniques and its application in diverse technical domains. Due to the inherent versatility, approaches of prognostics have been attempted to be classified in different ways [33, 34, 43, 45, 62], etc. Here, the authors have preferred to adapt the classification presented in [60].

Probabilistic Life-Usage Models These approaches depend upon the statistical information collected to assess the historical failure rate of the components and develop life-usage models [6, 30, 53]. Various functions can be applied to model statistical failure data such as exponential, normal, lognormal, and Weibull functions [39]. Moreover, the RUL is described as a *probability density function* (PDF) [60, 62]. Accurate assessment of RUL demands huge sets of failure database and extensive testing.

Data-Driven Prognostics The data associated with system functionality, degradation patterns, etc., are exploited using machine learning techniques to extract system signals and features which can be used to obtain behavior of damage progression, health index, etc. Broadly, two major strategies can be identified as discussed below.

Degradation Trend Extrapolation and Time Series Predictions In broad terms, the signals that indicate the state of the system are mapped as function of time and extrapolated in future using various techniques until a prefixed failure threshold is reached/crossed [27]. Mainly time series forecasting techniques are borrowed for this purpose such as: linear/non-linear regression techniques, autoregressive models [63], exponential smoothing techniques [10], autoregressive moving average

(ARMA), and autoregressive integrated moving average (ARIMA) [8]. The ARMA models and associated variants prove efficient for short-term predictions. Due to noise and inefficient uncertainty management, they prove less reliable for long term predictions.

Learning Damage Progression The degradation trends, failure patterns, etc., are learnt for training mathematical models. The latter in turn is used to model the relationship between damage progression and RUL. Employment of artificial neural networks (ANNs) and their numerous variants fall under this category. Feed-forward ANNs are extensively employed to estimate the current degradation index (state) by using system features (extracted signals, feature pattern, etc.) as inputs. Then, one step ahead prediction is generated by using previous state of degradation values (degradation index). The next iteration uses this prediction to produce long term predictions [31]. Major drawback in this context is that the efficiency of predictions remain limited in face of variable degradation trends, novel failure modes, etc. As such, accurate RUL predictions are not obtained on individual component unit to unit basis, but rather over large sets of component population. A comprehensive updated review of data-driven techniques can be found in [3, 61].

Model Based Prognostics Under this category, physics-of-failure models or degradation models (DM) are typically used to assess the damage progression and *state of health* (SOH). These DMs are derived from the first principles of physics. As such, they possess the capability of attaining maximum accuracy and versatility (scope of adaptation under varying degradation trend). There is a clear understanding of the underlying degradation process. There exists vast literature such as fatigue models for modelling initiation and propagation of cracks in structural components [65], electrolytic overstress aging [12], Arrhenius equation for prediction of resistance drift [41], physics-inspired power model [47] or log-linear model for degradation of current drain [46], and physics-inspired exponential degradation model for aluminum electrolytic capacitors [42].

Given the behavioral model of damage progression, the current SOH is popularly obtained in probabilistic domain with the help of Bayesian estimation techniques. Based upon the current SOH estimate, prediction of RUL is done. Such a probabilistic framework involving recursive Bayesian techniques efficiently addresses the main issues related to SOH under variable degradation; efficient management of uncertainty, environmental noise, future loading conditions, and associated confidence limits for RUL predictions [15, 16, 18, 54]. Filter for estimation and prediction process is chosen based upon the modelling hypothesis and desired performances [19]. Well-known Kalman filter, an optimal estimator for linear systems, has been used for prognostics in [12]. Extended Kalman filter (EKF) or unscented Kalman filter may also be used for joint state-parameter estimation as presented in [13, 52], respectively. However, they remain restricted to additive

Gaussian noise. Additionally, EKF being sub-optimal diverges quickly if the initial estimate of state is different from the reality by big measure or the model considered for estimation is not correct [57].

Set in Monte Carlo framework, PFs form a suitable filter choice in this context [4, 25]. PF can be applied to non-linear systems corrupted with non-Gaussian noises, for which optimal solutions may be unavailable or intractable. Comprehensive comparison of filters for prognostic purposes is found in [3, 19, 57]. Recently, PFs have been extensively for prognostic purposes [50]. Significant works include prediction of end of life (EOL) in lithium-ion batteries [55], battery health monitoring [56], prediction of battery grid corrosion [1], estimation and prediction of crack growth [11], fuel cell prognostics [37], application to damage prognostics in pneumatic valve [15, 17], estimation–prediction of wear as concurrent damage problem in centrifugal pumps with a variance control algorithm [18], employment in distributed prognosis [54], and uncertainty management for prognostics [5]. Particle filters attract considerable attention [2], owing to the ever growing efforts being made for betterment in performances and computational efficiency, such as the use of correction loops [51], fixed–lag filters [14], and kernel smoothing method [32].

The major issue in this type of approach is the accurate and reliable modelling of underlying degradation progression. Often, such accurate degradation models are not available.

Hybrid Prognostics The problem of non-availability of highly accurate degradation models is alleviated by fusing the advantages of model based and data-driven techniques. This way, there is significant amelioration in the overall prognostic approach [36, 37]. The basic philosophy remains in capturing the damage progression using DMs that can be: (1) based upon physics of failure, first principles of behavioral physics (2) derived using machine learning techniques, and (3) obtained statistically by finding a mathematical model that best fits a given set of degradation data such as linear model $D(t) = at + b$, logarithmic model $D(t) = a\ln(t) + b$, power model $D(t) = bt^a$, exponential model $D(t) = b \times e^{at}$ with $D(t)$ as an index representing the degradation (change, percentage change, etc.), and a and b as the model parameters. In this context, significant works are obtaining capacitance loss DM using non-linear least square regression [12], relevance vector machine regression performed over aging tests data [57], DM approximated by a linear part and logarithmic/exponential part [37], and residual based statistical DM [36]. Once the DM has been obtained with acceptable accuracy, recursive Bayesian techniques as discussed previously can be employed to estimate SOH and obtain subsequent RUL predictions. This way, benefits of Bayesian estimators are integrated with data-driven approaches to learn the DM as the current information arrives sequentially.

7.2.2 *Prognostics in BG Framework*

Almost all of the existing attempts in BG framework for prognostics have been ARR based and deterministic in nature. Moreover, DMs are considered deterministic so that the SOH and subsequent RUL predictions are obtained deterministically [20, 23, 35, 48, 49, 64]. Being restricted in deterministic domain, the randomness associated with variable damage progression, novel events, noises, etc., are simply ignored. As such, this leads to an inefficient management of the uncertainty in prognostication process and renders the RUL predictions without confidence limits. Recently, Jha et al. [36] proposed a methodology of hybrid prognostics where the benefits of Bayesian filtering techniques and BG enabled ARRs are integrated for efficient prognostics in probabilistic domain. In fact, this chapter is inspired by the work detailed in [36].

7.2.3 *Bond Graph in Linear Fractional Transformations*

BG-LFT is an efficient and systematic way of representing parametric uncertainty over nominal models. An uncertainty on a parameter value θ can be introduced under either an additive form or a multiplicative one, as shown in (7.1) and (7.2), respectively.

$$\theta = \theta_n \pm \varDelta\theta; \qquad \varDelta\theta \geq 0 \tag{7.1}$$

$$\theta = \theta_n\,(\,1 \pm \delta_\theta\,)\,; \quad \delta_\theta = \frac{\varDelta\theta}{\theta_n} \tag{7.2}$$

where $\varDelta\theta$ and δ_θ are, respectively, the absolute and relative deviations around the nominal parametric value θ_n. When the element characteristic law is written in terms of $\frac{1}{\theta}$, (7.2) becomes:

$$\frac{1}{\theta} = \frac{1}{\theta_n} \times \left(1 + \delta_{1/\theta}\right);\ \ \delta_{1/\theta} = \frac{-\varDelta\theta}{\theta_n + \varDelta\theta} \tag{7.3}$$

7.2.3.1 Representation on BG

The representation technique is illustrated briefly by taking a pedagogical example of R-element in *resistance* causality. The characteristic law corresponding to R-element in the linear case (see Fig. 7.1) is given as,

$$e_R = R \times f_R \tag{7.4}$$

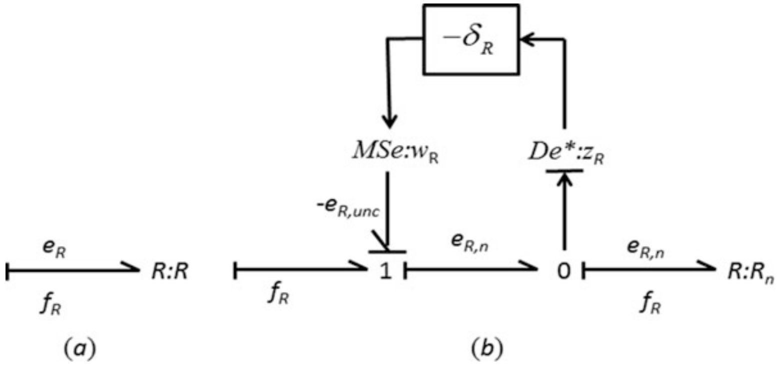


Fig. 7.1 (**a**) R-element in resistance causality. (**b**) Uncertain R-element in resistance causality in LFT form

In case of uncertainty on R, (7.4) becomes

$$e_R = R_n\,(1 + \delta_R) \times f_R = R_n \times f_R + \delta_R \times R_n \times f_R = e_{Rn} + e_{Runc} \tag{7.5}$$

Constitutive Eq. (7.5) can be represented as uncertain R-element as shown in Fig. 7.1b, wherein a modulated source MSe is introduced. The latter is associated with auxiliary input w_R and a virtual effort sensor which is associated with auxiliary output $z_{R.}$ It must be noted that negative (−) sign appears in the BG-LFT representation (see Fig. 7.1) due to the convention of power conservation. Moreover, the symbol De * represent virtual detectors. The virtual detectors are used to represent the information exchange/transfer.

Similarly, parametric uncertainty on the other passive elements can be represented. The technique remains similar for various other BG elements.

7.2.3.2 BG-LFT Based Robust Fault Detection

Fault diagnosis in BG-LFT framework is mainly dependent upon ARR generation [22]. ARRs are constraint relationships involving only known variables. In the context of BG modelling, an ARR : $f\big(\mathbf{SSe}(t), \mathbf{SSf}(t), \mathbf{Se}(t), \mathbf{Sf}(t), \boldsymbol{\theta}\big) = 0$, where $\boldsymbol{\theta}$ is vector of system parameters.

Generation of Uncertain ARRs The generation of robust analytical redundancy relations from an observable bond graph model is explained by the following steps:

First Step: Preferred derivative causality is assigned to the nominal model and detectors De (Df) are dualized to SSe (SSf); wherever possible. The BG-LFT model is constructed.

Second Step: The candidate ARRs are generated from “1” or “0” junction, where power conservation equation dictates that sum of efforts or flows, respectively, is equal to zero, as:

- For 0-junction:

$$\sum s_i \times f_{i,n} + \sum \mathrm{Sf} + \sum s_i w_i = 0 \tag{7.6}$$

- For 1-junction:

$$\sum s_i \times e_{i,n} + \sum \mathrm{Se} + \sum s_i w_i = 0 \tag{7.7}$$

with s being the sign rendered to the *bond* due to energy convention, w_i is the uncertain effort (flow) brought by the multiplicative parametric uncertainty δ_{θ_i} associated with ith system parameter θ_i, at 1(0) junction.

Third Step: The unknown effort or flow variables are eliminated using covering causal paths from unknown variables to known (measured) variables (dualized detectors), to obtain the ARRs which are sensitive to known variables as,

$$R = \Phi \left\{ \sum \mathrm{Se}, \sum \mathrm{Sf}, \mathrm{SSe}, \mathrm{SSf}, R_n, C_n, I_n, \mathrm{TF}_n, \mathrm{GY}_n, \mathrm{RS}_n, \sum w_i \right\} \tag{7.8}$$

where subscript n represents the nominal value of the corresponding BG element.

Generation of Adaptive Thresholds The ARR derived in (7.8) consists of two perfectly separable parts due to the properties of the BG-LFT model: a nominal part noted r shown in (7.9) and an uncertain part noted $b = \sum w_i$ shown in (7.10).

$$r = \Phi \{\mathrm{Se}, \mathrm{Sf}, \mathrm{SSe}, \mathrm{SSf}, R_n, C_n, I_n, \mathrm{TF}_n, \mathrm{GY}_n, \mathrm{RS}_n\} \tag{7.9}$$

$$\begin{aligned} &b = \sum w_i \\ &w_i = \Phi \{\mathrm{Se}, \mathrm{Sf}, \mathrm{SSe}, \mathrm{SSf}, R_n, C_n, I_n, \mathrm{TF}_n, \mathrm{GY}_n, \mathrm{RS}_n, \delta_R, \delta_I, \delta_C, \delta_{\mathrm{TF}}, \delta_{\mathrm{GY}}, \delta_{\mathrm{RS}}\} \end{aligned} \tag{7.10}$$

The uncertain part generates the adaptive threshold over the nominal part. From (7.8), (7.9), and (7.10), following may be obtained:

$$\begin{aligned} r + b &= 0 \\ r = -b &= -\sum w_i \end{aligned} \tag{7.11}$$

The thresholds are formed in form of envelop as:

$$-a < r < a \tag{7.12}$$

where

$$a = \sum |w_i| \tag{7.13}$$

The use of absolute values to generate the thresholds of normal operation ensures the robustness of this algorithm to false alarms. BG-LFT technique is well developed and detailed in literature. Readers are referred to [21, 22] for details.

7.2.4 Non-Linear Bayesian Inference Using Particle Filters

Consider a dynamic system whose state at time step t_k is represented by the vector $\mathbf{x}_k$. The evolution of the system state is described by a state space model,

$$\mathbf{x}_k = \boldsymbol{f}_k\left(\mathbf{x}_{k-1}, \boldsymbol{v}_{k-1}\right) \tag{7.14}$$

$$\mathbf{y}_k = \boldsymbol{h}_k\left(\boldsymbol{x}_k, \mathbf{w}_k\right) \tag{7.15}$$

where

- $\mathbf{f}_k : \mathbb{R}^{N_x} \times \mathbb{R}^{N_v} \rightarrow \mathbb{R}^{N_x}$ is a non-linear state transition function.
- $\boldsymbol{h}_k : \mathbb{R}^{N_x} \times \mathbb{R}^{N_w} \rightarrow \mathbb{R}^{N_y}$ is observation function describing the sequence of measurements $\mathbf{y}_k$, obtained sequentially at successive time steps t_k.
- $\boldsymbol{v}_k \in \mathbb{R}^{N_v}$ is the process noise sequence of known distribution assumed *independent and identically distributed* (i.i.d).
- $\boldsymbol{w}_k \in \mathbb{R}^{N_w}$ is i.i.d measurement noise sequence of known distribution.

Equations (7.14) and (7.15) can be equivalently represented as,

$$\mathbf{x}_k = \boldsymbol{f}_k\left(\mathbf{x}_{k-1}, \boldsymbol{v}_{k-1}\right) \leftrightarrow p\left(\mathbf{x}_k \middle| \mathbf{x}_{k-1}\right) \tag{7.16}$$

$$\mathbf{y}_k = \boldsymbol{h}_k\left(\mathbf{x}_k, \boldsymbol{w}_k\right) \leftrightarrow p\left(\mathbf{y}_k \middle| \mathbf{x}_{k-1}\right) \tag{7.17}$$

where $p\left(\boldsymbol{x}_k \middle| \boldsymbol{x}_{k-1}\right)$ represents the state transition probability, $p\left(\mathbf{y}_k \middle| \mathbf{x}_{k-1}\right)$ is the likelihood function which signifies the probability of the observation of $\mathbf{y}_k$, given the current estimate of $\boldsymbol{x}_k$.

Objective of filtering procedure is to obtain estimates of $\mathbf{x}_k$, based upon all of the available measurement sequences $\mathbf{y}_{1:k} = \{\mathbf{y}_k, k = 1, 2, \ldots .k\}$. From the perspectives of Bayesian inference, the objective remains in recursive calculation of state distribution $\mathbf{x}_k$, given the set of observations $\mathbf{y}_{1:k}$ up to time t_k, with some degree of belief. Construction of PDF $p\left(\mathbf{x}_k \middle| \mathbf{y}_{1:k}\right)$, known as the filtered *posterior state* PDF, provides all the information about $\mathbf{x}_k$, inferred from the measurements $\mathbf{y}_{1:k}$ and the initial state PDF $p(\mathbf{x}_0)$. The latter $p(\mathbf{x}_0)$ is assumed to be known. Given

$p\left(\mathbf{x}_{k-1}\middle|\mathbf{y}_{1\ \ k-1}\right)$ at time t_{k-1}, theoretically, the posterior state can be estimated in a recursive way via two sequential steps: *prediction* and *update*.

Prediction Application of Chapman–Kolmogorov equation over $p\left(\mathbf{x}_{k-1}\middle|\mathbf{y}_{1\ :\ k-1}\right)$ at time k–1 gives the estimation of *prior* state PDF $p\left(\mathbf{x}_{k}\middle|\mathbf{y}_{1\,:\ k-1}\right)$ at time t_k as,

$$\begin{aligned} p\left(\mathbf{x}_k\middle|\mathbf{y}_{1\ :\ k-1}\right) &= \int p\left(\mathbf{x}_k\middle|\mathbf{x}_{k-1}, \mathbf{y}_{1\ :\ k-1}\right) p\left(\mathbf{x}_{k-1}\middle|\mathbf{y}_{1\ :\ k-1}\right) \\ &= \int p\left(\mathbf{x}_k\middle|\mathbf{x}_{k-1}\right) p\left(\mathbf{x}_{k-1}\middle|\mathbf{y}_{1\ :\ k-1}\right) d\mathbf{x}_{k-1} \end{aligned} \tag{7.18}$$

Here, $p\left(\mathbf{x}_k\middle|\mathbf{x}_{k-1}\right)$ is obtained from (7.16), where the system is assumed to follow first order Markov dynamics.

Update Bayes rule is used to *update* the *prior* as the new measurement $\mathbf{y}_k$ arrives, to obtain the *posterior* distribution of $\mathbf{x}_k$ as,

$$p\left(\mathbf{x}_k\middle|\mathbf{y}_{1\ :\ k}\right) = \frac{p\left(\mathbf{x}_k\middle|\mathbf{y}_{1\ :\ k-1}\right) p\left(\mathbf{y}_k\middle|\mathbf{x}_k\right)}{p\left(\mathbf{y}_k\middle|\mathbf{y}_{1\ :\ k-1}\right)} \tag{7.19}$$

with the normalizing constant being,

$$p\left(\mathbf{y}_k\middle|\mathbf{y}_{1\ :\ k-1}\right) = \int p\left(\mathbf{x}_k\middle|\mathbf{y}_{1\ :\ k-1}\right) p\left(\mathbf{y}_k\middle|\mathbf{x}_k\right) d\mathbf{x}_k \tag{7.20}$$

This step incorporates the latest measurement into a priori state PDF $p\left(\mathbf{x}_k\middle|\mathbf{y}_{1\ :\ k-1}\right)$ to estimate the *posterior* state PDF $p\left(\mathbf{x}_k\middle|\mathbf{y}_{1\ :\ k}\right)$. The exact Bayesian solution obtained from recurrence relations (7.18) and (7.19) forms the basis of optimal Bayesian inference. This procedure remains tractable and produces best results for ideal systems such as linear Gaussian state space models. For the latter, it leads to the formation of classical Kalman filter. In general, optimal and closed form solutions for non-linear systems with non-Gaussian noises cannot be analytically determined. For non-linear state space models with additive Gaussian noises, sub-optimal Extended Kalman filter (EKF) has been developed. To obtain optimal solutions for non-linear systems, one resorts to Monte Carlo Methods. One such popular method is described below.

Particle filter (PF) is a type of Sequential Monte Carlo method [25], used for obtaining recursive Bayesian inferences via Monte Carlo simulations. Basic philosophy rests in representing the *posterior* state PDF by a set of random samples or "particles" where each of the particles has an associated weight based upon which the stateestimates are computed [26]. *Sequential importance sampling* (SIS) PF is one of the most popular PFs in which posterior state PDF $p\left(\mathbf{x}_{0:k}\middle|\mathbf{y}_{1:k}\right)$ by a set of N number of weighted particles [4],

$$\left\{\left(\mathbf{x}_{0:k}^{i}\right), \mathrm{w}_{k}^{i}\right\}_{i=1}^{N} \tag{7.21}$$

where $\left\{\mathbf{x}_{0:k}^{i}, i=1,\ldots N\right\}$ is the set of particles representing the state value with corresponding associated importance weights as $\left\{w_{k}^{i}, i=1,\ldots N\right\}$. Moreover, $\mathbf{x}_{0\ :\ k}=\left\{\mathbf{x}_{j}, j=0,\ldots.,k\right\}$ is the set of all states up to time k. It should be noted that these weights are the approximations of relative posterior probabilities of the particles normalized such that,

$$\sum_{i} w^{i}{}_{k}=1 \tag{7.22}$$

The posterior PDF is approximated as,

$$p\left(\mathbf{x}_{0:k}\middle|\mathbf{y}_{1\ :\ k}\right)\approx\sum_{i=1}^{N}\mathrm{w}_{k}^{i}\times\delta\left(\mathbf{x}_{0:k}-\mathbf{x}_{0:k}^{i}\right) \tag{7.23}$$

where δ denotes the Dirac delta function. This gives discrete weighted approximation to the true posterior state distribution $p\left(\mathbf{x}_{0:k}\middle|\mathbf{y}_{1\ :\ k}\right)$. As N tends to large numbers, the Monte Carlo approximation becomes an equivalent representation to the posterior state PDF.

7.2.5 Importance Sampling

Obtaining the particle weight(s) is not a trivial task. It becomes virtually impossible to sample from a posterior state $p\left(\mathbf{x}_{0:k}\middle|\mathbf{y}_{1\ :\ k}\right)$ without a closed form distribution. To resolve this issue, principle of *importance sampling* is used [4]. Here, a proposal distribution $q(x)$, known as *importance density*, is chosen such that $p(x)\propto q(x)$ and $q(x)$ is a PDF from which samples can be easily drawn. For example, if a set of samples $x^{i}\sim q(x),\ i=1,\ldots,N$ is generated from the proposal distribution $q(x)$, then the weighted approximation of the density $p(x)$ is given as,

$$p(x)\approx\sum_{i=1}^{N}\mathrm{w}^{i}\times\delta\left(x-x^{i}\right) \tag{7.24}$$

where normalized weight can be obtained as,

$$\mathrm{w}^i \approx \frac{p\left(x^i\right)}{q\left(\mathrm{x}^i\right)} \tag{7.25}$$

For a set of samples $\left\{\mathbf{x}^i_{0:k}, i = 1, \ldots N\right\}$, this leads to weights being defined as,

$$\mathrm{w}^i_k \propto \frac{p\left(\mathbf{x}_{0:k}\middle|\mathbf{y}_{1:k}\right)}{q\left(\mathbf{x}_{0:k}\middle|\mathbf{y}_{1:k}\right)} \tag{7.26}$$

For online implementation, a recursive estimation procedure is sought. In other words, distribution $p\left(\mathbf{x}_{0:k}\middle|\mathbf{y}_{1:k}\right)$ at time t_k must be estimated from $p\left(\mathbf{x}_{0:k-1}\middle|\mathbf{y}_{1:k-1}\right)$ at time t_{k-1}, in a sequential manner. To this end, a constraint on *importance density* is placed so that it is factorable as,

$$q\left(\mathbf{x}_{0:k}\middle|\mathbf{y}_{1\ :\ k}\right) = q\left(\mathbf{x}_k\middle|\mathbf{x}_{0:k-1}, \mathbf{y}_{1\ :\ k}\right) q\left(\mathbf{x}_{0:k-1}, \mathbf{y}_{1\ :\ k-1}\right) \tag{7.27}$$

Then, the new state $\mathbf{x}^i_{0:k} \sim q\left(\mathbf{x}_k\middle|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k}\right)$ can be appended with existing samples $\mathbf{x}^i_{0:k-1} \sim q\left(\mathbf{x}_{0:k-1}\middle|\mathbf{y}_{1:k-1}\right)$ to obtain new sets of samples $\mathbf{x}^i_{0:k} \sim q\left(\mathbf{x}_{0:k}\middle|\mathbf{y}_{1:k}\right)$. This is followed by update of particle weights. The posterior state PDF is expressed as,

$$p\left(\mathbf{x}_{0:k}\middle|\mathbf{y}_{1\ :\ k}\right) = p\left(\mathbf{x}_{0:k-1}\middle|\mathbf{y}_{0\ :\ k-1}\right) \frac{p\left(\mathbf{y}_k\middle|\mathbf{x}_k\right) p\left(\mathbf{x}_k\middle|\mathbf{x}_{k-1}\right)}{p\left(\mathbf{y}_k, \mathbf{y}_{1\ :\ k-1}\right)} \tag{7.28}$$

Then, using (7.26), (7.27), and (7.28), particles are updated recursively as,

$$\begin{aligned}
\mathrm{w}^i_k &\propto \frac{p\left(\mathbf{x}_{0:k}\middle|\mathbf{y}_{1:k}\right)}{q\left(\mathbf{x}_{0:k}\middle|\mathbf{y}_{1:k}\right)} \\
&\propto \frac{p\left(\mathbf{x}_{0:k-1}\middle|\mathbf{y}_{0\ :\ k-1}\right) p\left(\mathbf{y}_k\middle|\mathbf{x}_k\right) p\left(\mathbf{x}_k\middle|\mathbf{x}_{k-1}\right)}{q\left(\mathbf{x}_k\middle|\mathbf{x}_{0:k-1}, \mathbf{y}_{1\ :\ k}\right) q\left(\mathbf{x}_{0:k-1}, \mathbf{y}_{1\ :\ k-1}\right)} \\
&\propto \mathrm{w}^i_{k-1} \frac{p\left(\mathbf{y}_k\middle|\mathbf{x}_k\right) p\left(\mathbf{x}_k\middle|\mathbf{x}_{k-1}\right)}{q\left(\mathbf{x}_k\middle|\mathbf{x}_{0:k-1}, \mathbf{y}_{1\ :\ k}\right)}
\end{aligned} \tag{7.29}$$

In SIS PF, the *importance density* is set equal to a priori PDF of state, i.e., $q\left(\mathbf{x}_{0\ :\ k}\middle|\mathbf{x}_{0\ :\ k-1}\right) = p\left(\mathbf{x}_k\middle|\mathbf{x}_{k-1}\right) = f_k\left(\mathbf{x}_k\middle|\mathbf{x}_{k-1}\right)$. This translates to the fact that new particles can be generated from the previous set of particle by simulating the state transition function $f_k\left(\mathbf{x}_k\middle|\mathbf{x}_{k-1}\right)$. Moreover, assumption of Markov dynamics implies that $q\left(\mathbf{x}_k^i\middle|\mathbf{x}_{0:k-1}^i, \mathbf{y}_{1\,:\,k}\right) = q\left(\mathbf{x}_k^i\middle|\mathbf{x}_{k-1}^i, \mathbf{y}_k\right)$. This renders the whole procedure suitable for online implementation as only the filtered estimate $p\left(\mathbf{x}_k\middle|\mathbf{y}_{1:k}\right)$ is required at each step. Thus, only $\mathbf{x}_k^i$ and $\mathbf{y}_{1\,:\,k}$ should be stored and the previous state path up to $\mathbf{x}_{0:k-1}^i$ can be neglected. Weight update step (7.29) can be modified as,

$$\begin{aligned} \mathrm{w}_k^i &\propto \mathrm{w}_{k-1}^i \frac{p\left(\mathbf{y}_k\middle|\mathbf{x}_k^i\right) p\left(\mathbf{x}_k^i\middle|\mathbf{x}_{k-1}^i\right)}{q\left(\mathbf{x}_k^i\middle|\mathbf{x}_{0:k-1}^i, \mathbf{y}_{1\,:\,k}\right)} \\ &\propto \mathrm{w}_{k-1}^i \; p\left(\mathbf{y}_k\middle|\mathbf{x}_k^i\right) \end{aligned} \tag{7.30}$$

Then, the posterior filtered PDF $p\left(\mathbf{x}_k\middle|\mathbf{y}_{1:k}\right)$ is approximated as,

$$p\left(\mathbf{x}_k\middle|\mathbf{y}_{1\,:\,k}\right) \approx \sum_{i=1}^{N} \mathrm{w}_k^i \times \delta\left(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^i\right) \tag{7.31}$$

This simplified algorithm can be used for recursive estimation of state as the observations arrive sequentially. The likelihood functions of the new observations $p\left(\mathbf{y}_k\middle|\mathbf{x}_k^i\right)$ result in evaluation of weights of particles constituting the next state estimate.

7.2.6 *Particle Degeneracy and Resampling*

During the propagation steps, the approximation density is adjusted through reweighting of the particles. Previous steps lead to an inevitable situation where due to increase in weight variance, the importance weights become increasingly skewed. After few iterations, all but one particle have negligible weights (particle *degeneracy*) [26]. To avoid the latter, a new swarm of particles are resampled from the approximate *posterior* distribution obtained previously in the *update* stage, constructed upon the weighted particles [44]. The probability for a particle to be sampled remains proportional to its weight. This way, particles with smaller weights (signifying less contribution to estimation process) are discarded and particles with large weights are used for resampling. To resolve this issue, the standard SIS is accompanied by a *resampling* step (referred to as *Sampling-Importance resampling*) (SIR) PF [4]. The different ways of *resampling* can be referred in [24]. In this work, SIR PF is employed for estimation of SOH and RUL predictions. In general, the particles are forced in the region of high likelihood by multiplying high weighted particles and abandoning low weighted particles. In other words, *resampling* step involves elimination of those particles that have small weights so that focus shifts on the particles with large weight. This step results in generation of a new set of particles $\left\{\left(\mathbf{x}_{0:k}^{i*}\right), \mathrm{w}_k^i\right\}_{i=1}^N$ by resampling N times without *replacement* from the discrete approximation of $p\left(\mathbf{x}_k \middle| \mathbf{y}_{1\ :\ k}\right)$ as,

$$p\left(\mathbf{x}_k \middle| \mathbf{y}_{1\ :\ k}\right) \approx \sum_{i=1}^{N} \mathrm{w}_k^i \times \delta_{(\mathbf{x}_{0:k})}\left(d\mathbf{x}_{0\ :\ k}\right) \tag{7.32}$$

such that $Pr\left(\mathbf{x}_k^{i*} = \mathbf{x}_k^i\right) = w_k^i$. The new set of particles represents i.i.d from (7.32) and thus, the particle weights are reset again as $w_k^i = 1/N$.

7.3 Degradation Models

DMs capture the underlying degradation of a given component/subsystem with time, environmental and operational conditions, etc. DMs can be obtained based upon physics of degradation or statistical approaches [28, 29]. Given a prognostic candidate (system parameter) θ^d, the associated DM can be expressed as,

$$\theta^d(t) = g^d\left(\boldsymbol{\gamma}^d(t), \boldsymbol{v}^{\theta^d}(t)\right) \quad ; \qquad \theta^d\left(t = 0\right) = \theta_n^d \tag{7.33}$$

where $g^d(.)$ denotes the linear/non-linear *degradation progression function* (DPF) obtained from the corresponding DM. It models the degradation progression of $\theta^d(t)$. Moreover, $\boldsymbol{\gamma}^d(t) \in \mathbb{R}^{N_{\gamma^d}}$ presents the vector of *degradation progression parameters* (DPP), $\boldsymbol{v}^{\theta^d}(t) \in \mathbb{R}^{N_{\theta^d}}$ is the associated process noise vector and θ_n^d denotes nominal value of θ^d.

7.3.1 *Obtaining Degradation Model in BG Framework*

In BG framework, the DM of a system parameter $\theta^d \in \boldsymbol{\theta}$, $\boldsymbol{\theta} \in \mathbb{R}^{N_\theta}$ can be obtained from the time evolution profile of the respective ARR to which it is sensitive, assuming that the rest of the system parameters sensitive to the same ARR do not undergo any kind of progressive fault or degradation [7, 49]. Here, consider the point valued part of the dth I-ARR, $r^d(t)$ such that with $\boldsymbol{\theta}' = \boldsymbol{\theta} \backslash \theta^d(t)$, $t > 0$, $r^d(t) \neq 0$,

$$r^d(t) = \Psi_1^d\left(\theta^d(t), \boldsymbol{\theta}_n', \mathbf{SSe}(t), \mathbf{SSf}(t), \boldsymbol{Se}(t), \boldsymbol{Sf}(t)\right) \tag{7.34}$$

where subscript n denotes nominal value. The computed values of $r^d(t)$ at time sample points gives an implicit relation of the degradation profile of $\theta^d(t)$ in time.

Assuming that *implicit function theorem* is satisfied [40], (7.34) gives a real valued function ψ_d such that,

$$\theta^d(t) = \psi_d\left(r^d(t), \boldsymbol{\theta}'_n, \mathbf{SSe}(t), \mathbf{SSf}(t), \boldsymbol{Se}(t), \boldsymbol{Sf}(t)\right) \tag{7.35}$$

Equation (7.35) is a function of system measurements inputs (known variables), signal derivative(s), etc., it is always corrupted with noise. It should be noted that residual based DM should be obtained prior to prognostics. This routine can be performed offline, i.e., prior to the phase when system's health monitoring is of interest.

7.3.2 *Methodology of Hybrid Prognostics*

In this section, the methodology for prognostics is described. Following assumptions are made:

- Only system parameters are considered uncertain. Sensors are considered non-faulty.
- A single system parameter (prognostics candidate) is assumed to be under progressive degradation. In fact, it is assumed that single mode of degradation affects the system parameter.
- The system parameter (prognostics candidate) that undergoes degradation is assumed to be known a priori. The issue of isolation or isolability of the prognostic (faulty) candidate is assumed resolved. Let $\theta^d(t) \in \boldsymbol{\theta}$ be such prognostic candidate.
- Degradation model (DM) of $\theta^d(t) \in \boldsymbol{\theta}$ is assumed to be known a priori.
- For an ARR derived, only one system parameter sensitive to it (known a priori) varies with time.
- Noise associated with measurements (residuals) is assumed normally distributed Gaussian in nature.

  Objectives are
- Reliable estimation of prognostic candidate's SOH and state of hidden degradation parameters that accelerate or vary the degradation progression.
- Reliable prediction of the RUL of the prognostic candidate.

7.3.3 *Robust Detection of Degradation Initiation*

The problem of detecting the degradation beginning is treated as robust fault detection problem. The BG-LFT enabled fault detection method presented in Sect. 7.2.3 is exploited in the form of an efficient diagnostic module. To this end, following steps are taken.

Table 7.1 Detection of degradation

```
Algorithm 1: Detection of degradation initiation
Input: $r^d(k)$, $\sum |w_i(t)|$
Output: degradation detection
if $r^d(k) \geq -\sum |w_i(t)|$ and $r^d(k) \leq \sum |w_i(t)|$
    degradation detection ← false
    else
    degradation detection ← true
end if
```

Step 1: Preferred derivative causality is assigned to nominal model and sensors are dualized.

Step 2: BG-LFT model of the nominal system is obtained.

Step 3: ARR sensitive to θ^d is derived. Let the ARR be $R(t)$ and the associated residual (numerical evaluation of ARR) be $r^d(t)$.

Step 4: Robust thresholds are derived as explained in Sect. 7.2.3. Degradation initiation is detected when the residual goes out of the BG-LFT thresholds. The corresponding pseudo algorithm is given in Table 7.1.

7.3.4 Fault Model Construction

This section describes the fault model constructed for estimating the state of the prognostic candidate which denotes the state of health of the parameter.

7.3.4.1 State Equation

The parameter under degradation $\theta^d(t)$ is included as a tuple $(\theta^d, \boldsymbol{\gamma}^d, g^d)$ to model the damage progression in state space form. Here, $\boldsymbol{\gamma}^d(t) \in \mathbb{R}^{N_{\gamma^d}}$ is the vector of hidden parameters (DPP) that influence the speed of degradation significantly. The *fault model* for is constructed in state space form by considering the parameter θ^d as the state variable augmented with the DPP vector as,

$$\dot{\boldsymbol{x}}^d(t) = \boldsymbol{f}^d\left(\boldsymbol{x}^d(t), \boldsymbol{v}^{x_d}(t)\right) \tag{7.36}$$

where $\boldsymbol{x}^d(t) = \left[\theta^d(t), \ \boldsymbol{\gamma}^d(t)\right]^T$ is the augmented state vector, $\mathtt{f}^d$ is state transition function following the Markov dynamics, and $\boldsymbol{v}^{\boldsymbol{x}_d} \in \mathbb{R}^{N_{v^d}}$ is the process noise vector.

7.3.4.2 ARR Based Observation Equation

The nominal residual used for detection of degradation initiation can be further exploited used for SOH estimation if the corresponding ARR expression is altered to obtain the observation equation. To this end, following theorem is enunciated:

Theorem *Under the single degradation hypothesis, assuming that the nominal part $r_n^d(t)$ of an ARR derived from the BG-LFT model can be expressed as a linear combination of non-linear functions of degradation candidate parameter $\theta^d(t)$, the measurement of the $\theta^d(t)$ can be obtained from $r_n^d(t)$.*

Proof Let $\theta^d(t)$ be the degradation candidate and $\boldsymbol{\theta}' = \boldsymbol{\theta} \backslash \theta^d(t)$. Assuming the nominal part $r_n^d(t)$ can be expressed as,

$$r_n^d(t) = \varXi\left(\boldsymbol{\theta}'_n, \boldsymbol{SSe}(t), \boldsymbol{SSf}(t), \boldsymbol{Se}(t), \boldsymbol{Sf}(t)\right) + A^T \boldsymbol{\varphi}\left(\theta_n^d\right) \tag{7.37}$$

where $\forall i \big| i = 1, 2 \ldots m, A^{m\times 1} = [a_1 \;\; a_2 \ldots a_m]^T$ is a vector of known (measured system variables) with $a_i = \phi_i\left(\boldsymbol{\theta}_n{}', \mathrm{SSe}(t), \boldsymbol{SSf}(t), \boldsymbol{Se}(t), \boldsymbol{Sf}(t)\right)$ and $\boldsymbol{\varphi}^{m\times 1}\left(\theta^d(t)\right) = \left[\varphi_1\left(\theta^d(t)\right), \varphi_2\left(\theta^d(t)\right), \ldots. \varphi_m\left(\theta^d(t)\right)\right]^T$ is the vector of non-linear functions of $\theta^d(t)$. Then, $\forall t \geq 0$ power conservation at the BG *junction* where the corresponding ARR is derived, gives

$$\mathrm{ARR} : r^d(t) = \Xi\left(\boldsymbol{\theta}'_n, SSe(t), \boldsymbol{SSf}(t), \boldsymbol{Se}(t), \boldsymbol{Sf}(t)\right) + A^T \boldsymbol{\varphi}\left(\theta^d(t)\right) = 0 \tag{7.38}$$

or,

$$\begin{aligned} r^d(t) = \Xi & \left(\boldsymbol{\theta}'_n, \mathbf{SSe}(t), \mathbf{SSf}(t), \sum Se, \sum Sf, \right) \\ & + A^T \boldsymbol{\varphi}\left(\theta_n^d\right) + \left(A^T \boldsymbol{\varphi}\left(\theta^d(t)\right) - A^T \boldsymbol{\varphi}\left(\theta_n^d\right)\right) = 0 \\ r^d(t) = & \, r_n^d(t) + A^T\left(\boldsymbol{\varphi}\left(\theta^d(t)\right) - \boldsymbol{\varphi}\left(\theta_n^d\right)\right) = 0 \\ r_n^d(t) = & - \; A^T\left(\boldsymbol{\varphi}\left(\theta^d(t)\right) - \boldsymbol{\varphi}\left(\theta_n^d\right)\right) \end{aligned} \tag{7.39}$$

Thus, state of $\theta^d(t)$ can be linked implicitly with measurements obtained by the nominal part $r_n^d(t)$.

Corollary *When* $\boldsymbol{\varphi}\left(\theta_n^d\right) = \varphi\left(\theta_n^d\right) = \theta_n^d$, *the vector* $A = a_1$, $a_1 = \phi_1\left(\boldsymbol{\theta}'_n, \mathbf{SSe}(t), \mathbf{SSf}(t), \sum \mathrm{Se}, \sum \mathrm{Sf}\right)$, *can be understood as a coefficient function linking the fault value to the residual. It can be found as,*

$$a_1 = \frac{\partial\left(r_n^d(t)\right)}{\partial\left(\theta^d(t)\right)} \tag{7.40}$$

Thus, observation equation can be formed as,

$$y^d(t) = r_n^d(t) = -A^T\left(\boldsymbol{\varphi}\left(\theta^d(t)\right) - \boldsymbol{\varphi}\left(\theta_n^d\right)\right) \tag{7.41}$$

In this work noise is considered additive, i.i.d., drawn from a zero mean normal distribution and is assumed uncorrelated to $\boldsymbol{x}^d(t)$. Observation equation is formed from (7.41) as,

$$y^d(t) = h^d\left(\boldsymbol{x}^d(t)\right) + w^d(t) \tag{7.42}$$

where $h_d\left(\cdot\right)$ is a non-linear observation function obtained from (7.41) and $w^d(t) \sim \mathcal{N}\left(0, \sigma_{w^d}^2\right)$. Moreover, the standard deviation σ_{w^d} is approximated from residual measurements during the degradation tests.

Thus, the nominal residual can provide information of damage and SOH of the prognostic candidate.

7.3.5 State of Health Estimation

In discrete time step $k \in \mathbb{N}$, the fault model can be described as,

$$\boldsymbol{x}_k^d = \boldsymbol{f}_k^d\left(\boldsymbol{x}_{k-1}^d, \boldsymbol{v}_{k-1}^{\boldsymbol{x}_d}\right) \tag{7.43}$$

$$y_k^d = \boldsymbol{h}_k^d\left(\boldsymbol{x}_k^d\right) + w_k^d \tag{7.44}$$

The initial state PDF $p\left(\theta_{k-1}^d, \boldsymbol{\gamma}_{k-1}^d \middle| y_{k-1}^d\right)$ is assumed to be known a priori. Estimations of θ_k^d, $\boldsymbol{\gamma}_k^d$ are obtained Bayesian framework as explained in Sect. 7.2.4. The latter is obtained as PDF $p\left(\theta_k^d, \boldsymbol{\gamma}_k^d \middle| y^d{}_{0:\ k}\right)$, at discrete time k, based upon the history of measurements till time k, $y_{0:\ k}^d$.The arriving measurement y_k^d is assumed conditionally independent of the state process. The likelihood function becomes as,

$$p\left(y_k^d \middle| \theta_k^d,\ \boldsymbol{\gamma}_k^d\right) = \frac{1}{\sigma_{w_k^d}\sqrt{2\pi}} \exp\left(-\left(y_k^d - h^d\left(x_k^d\right)\right)^2 / 2\sigma_{w_k^d}^2\right) \tag{7.45}$$

Estimation procedure using PF (see Sect. 7.2.4) is carried out such that the state PDF is approximated by set of discrete weighted samples or particles, $\left\{\left(\theta_k^{d,i},\ \boldsymbol{\gamma}_k^{d,i}\right), \mathrm{w}_k^i\right\}_{i=1}^N$, where N is the total number of particles. For ith particle at time k, $\theta_k^{d,i}\ \boldsymbol{\gamma}_k^{d,i}$ are the joint estimate of the state. In PF, the posterior density at any time step k is approximated as,

$$p\left(\theta_k^d, \boldsymbol{\gamma}_k^d \middle| y_{0\ :\ k}^d\right) \approx \sum_{i=1}^N \mathrm{w}_k^i \times \delta_{\left(\theta_k^d, \boldsymbol{\gamma}_k^d\right)}\left(d\theta_k^d\ \ d\boldsymbol{\gamma}_k^d\right) \tag{7.46}$$

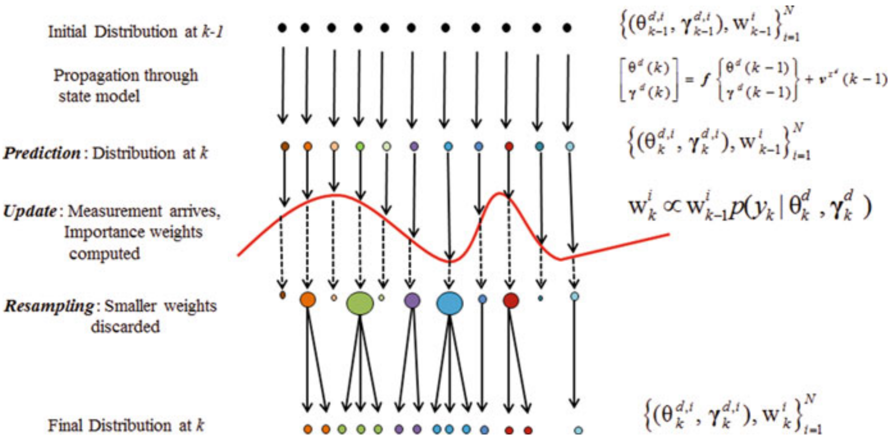

Initial Distribution at *k-1*
Propagation through state model
Prediction: Distribution at *k*
Update: Measurement arrives, Importance weights computed
Resampling: Smaller weights discarded
Final Distribution at *k*
$\{(\theta^{d,i}_{k-1}, \gamma^{d,i}_{k-1}), w^{i}_{k-1}\}_{i=1}^{N}$
$\begin{bmatrix}\theta^{d}(k)\\ \gamma^{d}(k)\end{bmatrix} = f\begin{Bmatrix}\theta^{d}(k-1)\\ \gamma^{d}(k-1)\end{Bmatrix} + v^{x^d}(k-1)$
$\{(\theta^{d,i}_{k}, \gamma^{d,i}_{k}), w^{i}_{k-1}\}_{i=1}^{N}$
$w^{i}_{k} \propto w^{i}_{k-1} p(y_k \mid \theta^{d}_{k}, \gamma^{d}_{k})$
$\{(\theta^{d,i}_{k}, \gamma^{d,i}_{k}), w^{i}_{k}\}_{i=1}^{N}$

Fig. 7.2 Illustration of estimation process in particle filters

where $\delta_{(\theta^d_k, \boldsymbol{\gamma}^d_k)}\left(d\theta^d_k \;\; d\boldsymbol{\gamma}^d_k\right)$ denotes the Dirac delta function located at $(\theta^d_k, \boldsymbol{\gamma}^d_k)$ and sum of the weights $\sum_{i=1}^{N} w^i_k = 1$. In this work, SIR PF is employed, owing to the easiness of importance weight evaluation [4]. Firstly, it is assumed that the set of random samples (particles) $\left\{\left(\theta^{d,i}_{k-1}, \;\; \boldsymbol{\gamma}^{d,i}_{k-1}\right), \mathrm{w}^i_{k-1}\right\}_{i=1}^{N}$ are available as the realizations of posterior probability $p\left(\theta^d_{k-1}, \boldsymbol{\gamma}^d_{k-1} \middle| y^d_{0\;:\;k-1}\right)$ at time $k-1$. Then, three significant steps are followed as illustrated in Fig. 7.2.

Prediction The particles are propagated through system model by: sampling from the system noise $\boldsymbol{v}^{\boldsymbol{x}_d}_{k-1}$ and simulation of system dynamics shown in (7.43).

This leads to new set of particles which are nothing but the realizations of prediction distribution $p\left(\theta^d_k, \boldsymbol{\gamma}^d_k \middle| y^d_{0\;:\;k-1}\right)$.

Update As the new measurement y^d_k arrives, a weight w^i_k is associated with each of the particles based on the likelihood of observation y^d_k made at time k as,

$$w^i_k = p\left(y^d_k \middle| \theta^{d,i}_k, \boldsymbol{\gamma}^{d,i}_k\right) / \sum_{j=1}^{N} p\left(y^d_k \middle| \theta^{d,j}_k, \boldsymbol{\gamma}^{d,j}_k\right) \tag{7.47}$$

Resampling There exist many types of resampling techniques [24]. In this work, *systematic resampling* is preferred owing to its simplicity in implementation, $O(N)$ computational time, and modular nature. The *resampling* method is well detailed in literature and thus, not described here.

The *prediction*, *update*, and *resample* procedures form a single iteration step; they are applied at each time step k.

Table 7.2 SIR particle filter for SOH estimation

```
Algorithm 2: estimation using SIR filter
Inputs: $\left\{\left(\theta_{k-1}^{d\ ,i},\ \boldsymbol{\gamma}_{k-1}^{d\ ,i}\right), w_{k-1}^{i}\right\}_{i=1}^{N}, y_k^d$
Output: $\left\{\left(\theta_k^{d,i},\ \boldsymbol{\gamma}_k^{d,i}\right), w_k^{i}\right\}_{i=1}^{N}$
for $i = 1$ to N do
    $\boldsymbol{\gamma}_k^{d,i} \sim p\left(\boldsymbol{\gamma}_k^{d,i} \middle| \boldsymbol{\gamma}_{k-1}^{d,i}\right)$
    $\theta_k^{d,i} \sim p\left(\theta_k^{d,i} \middle| \theta_{k-1}^{d,i}, \boldsymbol{\gamma}_{k-1}^{d,i}\right)$
    $w_k^i \sim p\left(y_k^d \middle| \theta_k^{d,i}, \boldsymbol{\gamma}_k^{d,i}\right)$
end for
$W \leftarrow \sum_{i=1}^{N} w_k^i$
for $i = 1$ to N do
    $w_k^i \leftarrow w_k^i / W$
end for
$\left\{\left(\theta_k^{d,i},\ \boldsymbol{\gamma}_k^{d,i}\right), w_k^{i}\right\}_{i=1}^{N} \leftarrow$ RESAMPLE$\left\{\left(\theta_k^{d,i},\ \boldsymbol{\gamma}_k^{d,i}\right), w_k^{i}\right\}_{i=1}^{N}$
```

The pseudo algorithm is provided in Table 7.2.

7.3.6 RUL Prediction

The critical/failure value θ_{fail}^d of $\theta^d(t)$ must be fixed beforehand. Once the *posterior* PDF $p\left(\theta_k^d, \boldsymbol{\gamma}_k^d \middle| y_{0\ :\ k}^d\right)$ has been estimated at time step k, it should be projected in future in such a way that information about EOL at time step k, EOL_k, is obtained depending upon the actual SOH. Then, RUL at time k can be obtained as,

$$\text{RUL}_k = \text{EOL}_k - k \tag{7.48}$$

Obviously, such a projection of degradation trajectory in future has to be done in absence of measurements. Thus, this process remains outside the domain of traditional Bayesian filtering techniques. In practice, one of the efficient ways to achieve such a projection is to propagate the *posterior* PDF $p\left(\theta_k^d, \boldsymbol{\gamma}_k^d \middle| y_{0\ :\ k}^d\right)$ using the DM inspired state model (7.43) until the failure horizon θ_{fail}^d is reached. The latter may take l^d time steps so that $\theta^d = \theta_{\text{fail}}^d$ at a time $t + l^d$. This calls for computation of the predicted degradation state $p\left(\theta_{k+l^d}^d, \boldsymbol{\gamma}_{k+l^d}^d \middle| y_{0\ :\ k}^d\right)$ as [25],

$$
\begin{aligned}
&p\left(\theta^d_{k+l^d}, \boldsymbol{\gamma}^d_{k+l^d} \middle| y^d_{0\,:\,k}\right) \\
&= \int \cdots \int \prod_{j=k+1}^{k+l^d} p\left(\left(\theta^d_j, \boldsymbol{\gamma}^d_j\right) \middle| \left(\theta^d_{j-1}, \boldsymbol{\gamma}^d_{j-1}\right)\right) \, p\left(\theta^d_k, \boldsymbol{\gamma}^d_k \middle| y^d_{0\,:\,k}\right) \prod_{j=k}^{k+l^d-1} d\left(\theta^d_j, \boldsymbol{\gamma}^d_j\right)
\end{aligned}
\tag{7.49}
$$

Obtaining the numerical value of this integral is computationally very expensive.

PFs can be employed for optimal estimation of such integrals under certain assumptions [51] reviews various methods for computation of (7.49). In [25], it is proposed that weights of the particles from time step k until $k+\ l^d$ can be kept constant for l^d step ahead computation. This is based on the assumption that error generated/accumulated by keeping the weights same is negligible compared to other error sources, such as settings of process noise, measurement noise, random walk variance, and model inaccuracy [50].

In our context, as illustrated in Fig. 7.3, RUL predictions can be achieved by projecting the current SOH estimation into future [15, 16, 18, 37]. Once the particles $\left\{\left(\theta^{d,i}_k,\ \boldsymbol{\gamma}^{d,i}_k\right), w^i_k\right\}^N_{i=1}$, constituting the realizations of the current joint state-parameter estimate $p\left(\theta^d_k, \boldsymbol{\gamma}^d_k \middle| y^d_{0\,:\,k}\right)$ are obtained, each of the particles is propagated into future to obtain a l^d-step ahead state distribution with $l^d = 1, \ldots T^d - k$, where T^d is the time until SOH remains less than failure value, i.e., time until $\theta^d_{k+l^d} \geq \theta^d_{\text{fail}}$. For l^d-step ahead state distribution, each of the particles is propagated using the state equation of the *fault model*. Here, for the ith particle, the corresponding weight during the $l^{d,i}$-step propagation is kept equal to weight w^i_k at time of prediction k. Then, for ith particle, $\text{RUL}^i_k = k + l^{d,i} - k = l^{d,i}$ and the corresponding PDF is obtained as,

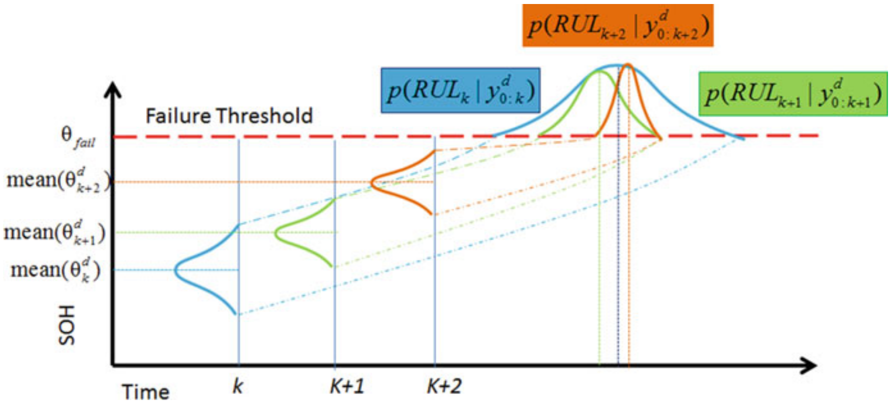


Fig. 7.3 Schematic illustration of RUL prediction process

Table 7.3 RUL prediction

```
Algorithm 3: RUL prediction
Inputs: $\left\{\left(\theta_k^{d,i},\ \boldsymbol{\gamma}_k^{d,i}\right), w_k^i\right\}_{i=1}^N$
Variable: $l$
Outputs: $\left\{\mathrm{RUL}_k^i, w_k^i\right\}_{i=1}^N$
  for $i = 1$ to $N$ do
    $l = 0$
    while $\theta_{k+l}^{d,i} \leq \theta_{\mathrm{fail}}^d$ do
      $\boldsymbol{\gamma}_{k+1}^{d,i} \sim p\left(\boldsymbol{\gamma}_{k+1}^{d,i} \middle| \boldsymbol{\gamma}_k^{d,i}\right)$
      $\theta_{k+1}^{d,i} \sim p\left(\theta_{k+1}^{d,i} \middle| \theta_k^{d,i}, \boldsymbol{\gamma}_k^{d,i}\right)$   $l \leftarrow l + 1$
    end while
    $\mathrm{RUL}_k^i \leftarrow l$
  end for
  $\left\{\mathrm{RUL}_k^i, w_k^i\right\}_{i=1}^N \sim p\left(\mathrm{RUL}_k \middle| y_{0\,:\,k}^d\right)$
```

$$p\left(\mathrm{RUL}_k \middle| y_{0\,:\,k}^d\right) \approx \sum_{i=1}^{N} w_k^i\ \delta_{\left(\mathrm{RUL}_k^i\right)}\left(d\mathrm{RUL}_k^i\right) \tag{7.50}$$

The associated pseudo algorithm is provided in Table 7.3.

7.4 Integrated Health Monitoring

The degradation initiation is detected by BG-LFT based robust fault detection technique, as discussed in Sect. 7.3.3. The initial value of SOH of prognostic candidate is set as:

$$\theta_{t=t_d}^d \sim U\left(\theta_n^d - \Delta\theta_l, \theta_n^d + \Delta\theta_u\right) \quad ; t = t_d \tag{7.51}$$

where t_d is the time when degradation is detected as fault. The associated uncertainty interval limits $[-\Delta\theta_l, \Delta\theta_u]$ decide the bounds of the uniform distribution.

The complete algorithm is shown in Table 7.4. Figure 7.4 shows the schematic description of the methodology presented in this chapter.

7.5 Evaluation Metrics

In this section, evaluation metrics are provided to assess prognostic performance. For details, readers referred to [18, 59].

Table 7.4 Integrated health monitoring of prognostic candidate

Algorithm 4: Health monitoring of θ_0^d

```
while system is running do
Detect the beginning of degradation using Algorithm 1
if fault detection = true then
//set initial conditions
 $\theta_0^d \sim U\left(\theta_n^d - \Delta\theta_l, \theta_n^d + \Delta\theta_u\right)$
  $\boldsymbol{\gamma}_0^d = 0$
 $y_0^d = r_n^d(k)$
do SOH Estimation using Algorithm 2
do RUL prediction using Algorithm 3
end if
end while
```

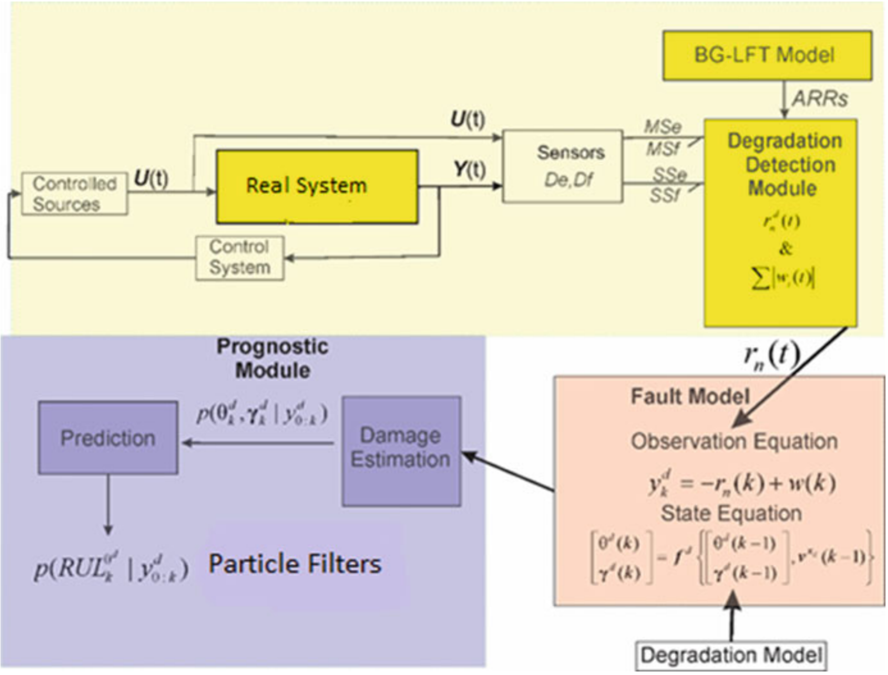


Fig. 7.4 Schematic description of the Health Monitoring Methodology

Root mean square error (RMSE) metric expresses the relative estimation accuracy as:

$$\mathrm{RMSE}_X = \sqrt{\mathrm{Mean}_k\left[\left(\frac{\mathrm{mean}(X) - X^*}{X^*}\right)^2\right]} \tag{7.52}$$

where for a specie X, $X*$ denotes its corresponding true value. Mean_k denotes the mean over all values of k. This metric is useful in assessing the estimation performance.

On the other hand, assessment of RUL predictions is possible if the actual RUL or RUL ground truth is known. The terms RUL ground truth and true RUL are used interchangeably in this chapter. A fairly good idea of true RUL can be obtained beforehand from the corresponding DM, under the assumption that degradation proceeds with uniform speed. Obviously, the hidden DPPs influence the actual speed and SOH. As such, in reality, true RUL can only be estimated with certain degree of belief. In this chapter, it is assumed that degradation progresses with uniform speed. As such, for evaluation purposes, true RUL is assessed from DM. A detailed discussion on this subject and RUL evaluation metrics can be found in [58, 59].

Alpha–Lambda ($\alpha - \lambda$) Metric [59] An accuracy cone is formed by choosing α such that $\alpha \in [0, 1]$, followed by generation of accuracy cone (envelope) over true RUL at time instant k, RUL_k^*, as $\left[(1-\alpha)\,\text{RUL}_k^*, (1+\alpha)\,\text{RUL}_k^*\right]$. Clearly, value of α signifies the degree of uncertainty associated with RUL_k^*, allowed for assessment of RUL predictions. Figure 7.5 shows ground truth RUL line and α cone that envelopes it. The estimated RUL PDFs must have significant amount of probability mass within the α-cone, to be accepted as "true" predictions. Then, accuracy of RUL predictions can be efficiently assessed by *relative accuracy* (RA) metric. The latter is explained by first recalling the fact that RUL predictions are obtained as PDFs (see Fig. 7.3). In this work, RUL PDFs are represented using box plot representation. As shown in Fig. 7.5, the box plot representation is capable of denoting the PDF's mean, median, 5th and 95th percentiles of distribution data, and the associated outliers. At a particular prediction instant k, the RUL prediction accuracy for θ^d is evaluated by relative accuracy (RA) metric as,

$$\text{RA}_k = \left(1 - \frac{\left|\text{RUL}_k^* - \text{Median}\,p\left(\text{RUL}_k\right)\right|}{\text{RUL}_k^*}\right) \tag{7.53}$$

$$\overline{\text{RA}} = \text{Mean}_k p\left(\text{RA}_k\right) \tag{7.54}$$

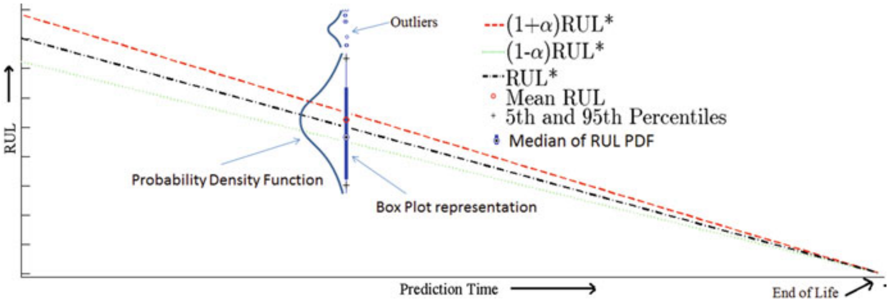


Fig. 7.5 Illustration of box plot representation and $\alpha - \lambda$ accuracy cone

where RUL_k^* denotes the true RUL at time k for θ^d. The overall accuracy is determined by $\overline{\mathrm{RA}}$ as shown in (7.53), where RA_k is averaged over all the prediction points.

7.6 Application on Mechatronic System in Real Time

This section describes the application of the method over a mechatronic system [36] shown in Fig. 7.6. Real time implementation is achieved through 20 SIM 4C 2.1. The SOH estimation and RUL prediction algorithms are written in *Matlab Function Block* in Simulink. The embedded code is generated through Simulink Coder in Matlab2013a®.

7.6.1 Nominal System

The functional schematic model of the mechatronic system [38] is shown in Fig. 7.6. The designation of system variables and associated values are listed in Table 7.5. The system consists of the Maxon® servo motor that provides the controlled actuation (rotation) to disks (Fig. 7.7). The high stiffness transmission belt provides torque the transmission ratio of k_{belt} to the motor disk. The motor disk is connected to load disk through a flexible shaft that constitutes the drive train. The shaft is modelled as spring-damper element. The friction in the bearings of the motor disk and load disk are modelled as viscous friction. Friction arising due to belt is lumped with viscous friction coefficient at motor disk b_{Md}. The setup is equipped with motor encoder and load encoder that measure angular position of motor shaft and load disk (2000 pulses per revolution), respectively (Fig. 7.8). Angular position motor disk is obtained by dividing the motor encoder counts by belt ratio. The BG model of the

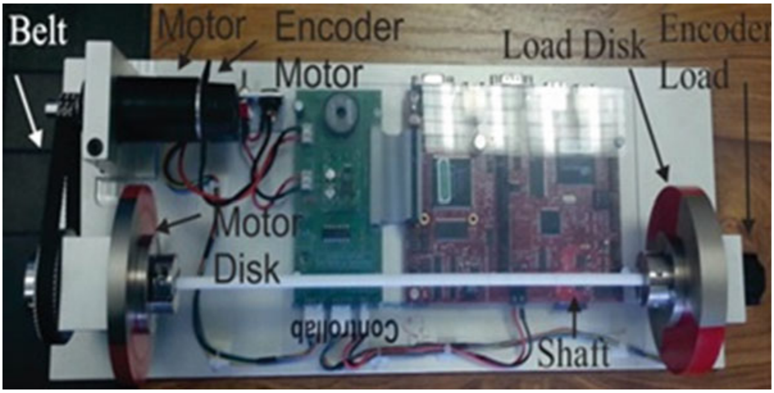


Fig. 7.6 Mechatronic torsion bar 1.0 system

Table 7.5 Details of system variables

Parameter θ	Designation	Nominal value θ_n	Multiplicative uncertainty δ_θ
k_s	Spring constant of the shaft	1.786 N m/rad	10 %
b_s	Damping coefficient of shaft	5.11×10^{-4} N m/rad	10 %
k_m	Torque constant	3.89×10^{-4} N m/A	–
k_{belt}	Teeth ratio (motor disk and motor shaft)	3.75	–
La	Rotor inductance	1.34×10^{-3} H	–
Ra	Rotor resistance	1.23Ω	–
Jm	Rotor inertia	6.76×10^{-6}kg m^2/rad	20 %
f_m	Motor friction coefficient	2×10^{-6} N m s/rad	20 %
J_{Md}	Motor disk rotational inertia	9.07×10^{-4}kg m^2/rad	10 %
b_{Md}	Viscous friction in motor disk	5.025×10^{-3} N m s/rad	20 %
J_{Ld}	Load disk rotational inertia	1.37×10^{-3}kg m^2/rad	20 %
b_{Ld}	Viscous friction in load disk	2.5×10^{-5}N m s/rad	20 %
$\text{SSf}_1 : \omega_m$	Motor velocity measurement	–	–
$\text{SSf}_2 : \omega_{\text{Ld}}$	Load disk velocity measurement	–	–
μ	Friction coefficient	0.27	10 %

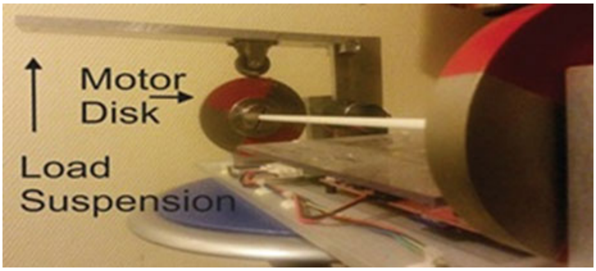
nominal system in integral causality is given in Fig. 7.9. Only the monitorable part is used for analysis. The system is considered operating in feedback closed loop with Proportional-Integral (PI) controlled input voltage. The control input from PI controller (controlled variable: motor speed ω_m) modulates the input voltage MSe*:* U_{PI}.

For experiments, a mechanical lever type arrangement is fabricated as shown in Fig. 7.7 which introduces frictional torque τ_{Mech} over the motor disk by suspension of load in form of sand. The associated frictional torque is due to Coulomb friction existing between the surfaces (μ being friction coefficient). It is modulated by the suspended load M as,

$$\begin{aligned} \tau_{\text{Mech}} &= f_{\text{mech}} \times r_{\text{Md}} \\ f_{\text{mech}} &= \mu \ \ \text{Mg} \ \ (\omega_{\text{Md}} / \left|\omega_{\text{Md}}\right|) \end{aligned} \tag{7.55}$$

with r_{Md} as the radius of the motor disk. In the BG model, it is incorporated as non-linear resistance element R: b_{Md}. The corresponding characteristic equation becomes as,

Fig. 7.7 Fabricated mechanical lever type arrangement for load (mass) suspension



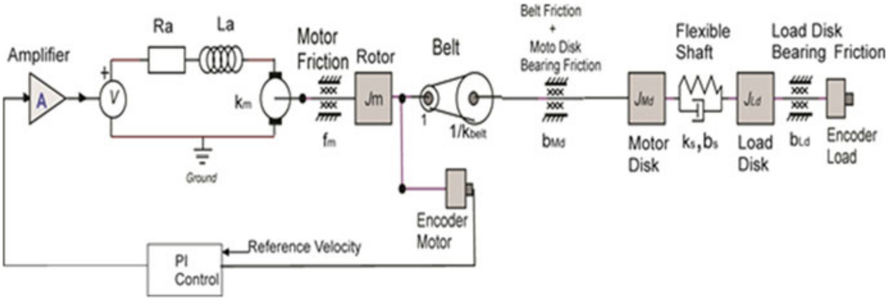


Fig. 7.8 Schematic model of the mechatronic system

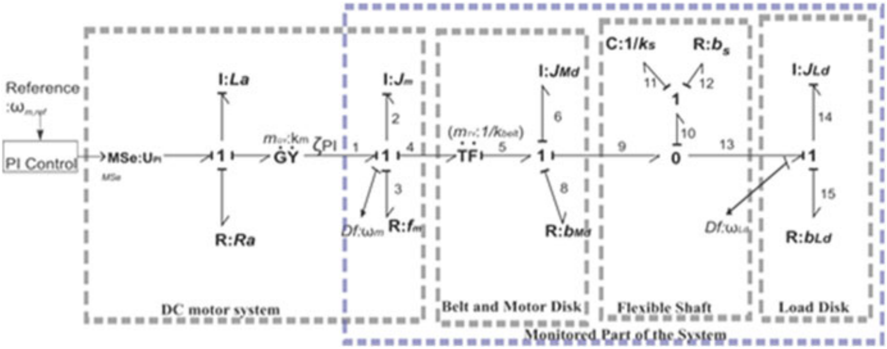


Fig. 7.9 BG model (preferred integral causality) of the nominal system

$$R = b_{\mathrm{Md}} + \mu \times M(t) \times r_{\mathrm{Md}} g / |\omega| \tag{7.56}$$

$$e_8 = R(f_8) = b_{\mathrm{Md}} \omega_{\mathrm{Md}} + \mu \times M(t) \times r_{\mathrm{Md}} g \times (\omega_{\mathrm{Md}} / |\omega_{\mathrm{Md}}|) \tag{7.57}$$

Involving only non-destructive experiments, μ is assumed undergoing no wear.

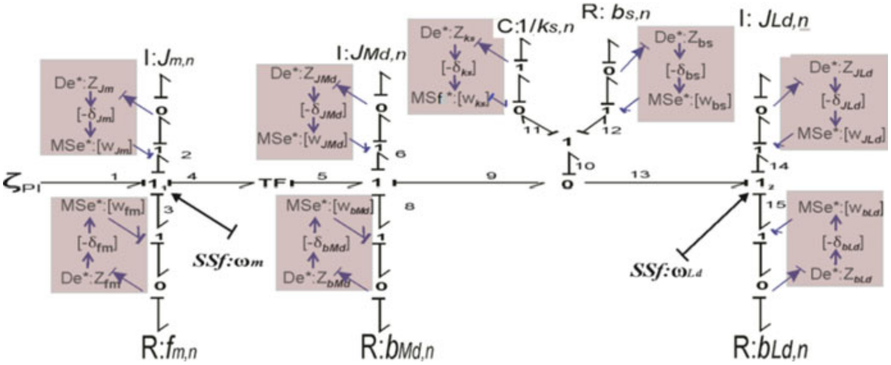


Fig. 7.10 BG-LFT model of monitorable part of the system

7.6.2 BG-LFT Model and ARR Generation

The BG-LFT model constructed in preferred derivative causality is shown in Fig. 7.10. Both the sensors are dualized and impose corresponding flows as $\boldsymbol{Y}(t) = [\mathrm{SSf}_1 : \omega_m, \mathrm{SSf}_2 : \omega_{\mathrm{Ld}}]^T$. C element remains in integral causality with the initial condition given by the flow at respective **0**-junction, provided by encoder readings as $f_{10} = f_9 - f_{13} = (\omega_m/k_{\mathrm{belt}}) - \omega_{\mathrm{Ld}}$. Moreover, electrical torque $\mathrm{MSe} : \tau_{\mathrm{PI}}$ is the PI controlled input to the monitorable part of the system and is given as:

$$\mathrm{MSe} : \tau_{\mathrm{PI}} = k_m \times i_m = k_m \times \frac{(U_{\mathrm{PI}} - k_m \times \omega_m)}{\mathrm{Ra}} \left(1 - e^{-(\mathrm{Ra}/\mathrm{La}) \times t}\right) \tag{7.58}$$

where U_{PI} is the PI controlled voltage input and i_m is the motor stator current.

Following the steps described in Sect. 7.2.3, an ARR can be generated from the detectable junction $\mathbf{1_1}$ of Fig. 7.10 as,

$$R_1 = r_1(t) + \sum w_i \tag{7.59}$$

where

$$\begin{aligned} r_1(t) = {} & \tau_{\mathrm{in}} - J_{m,n}\dot{\omega}_m - f_{m,n}\omega_m \\ & - \frac{1}{k_{\mathrm{belt}}} \begin{pmatrix} J_{\mathrm{Md},n}\frac{\dot{\omega}_m}{k_{\mathrm{belt}}} + b_{\mathrm{Md},n}\frac{\omega_m}{k_{\mathrm{belt}}} + \mu_n M_n g\ \ r_{\mathrm{Md}}\,\mathrm{sgn}\,(\omega_m/k_{\mathrm{belt}}) \\ + k_{s,n}\int \left(\dfrac{\omega_m}{k_{\mathrm{belt}}} - \omega_{\mathrm{Ld}}\right) dt + b_{s,n}\left(\dfrac{\omega_m}{k_{\mathrm{belt}}} - \omega_{\mathrm{Ld}}\right) \end{pmatrix} \end{aligned} \tag{7.60}$$

$$\begin{aligned}
\sum w_i &= w_{Jm} + w_{fm} + w_{J_{\mathrm{Md}}} + w_{b_{\mathrm{Md}}} + w_{ks} + w_{b_s} \\
w_{Jm} &= -\delta_{J_m} J_{m,n} \dot{\omega}_m ; w_{fm} = -\delta_{f_m} f_{m,n} . \omega_m ; \\
w_{J_{\mathrm{Md}}} &= -\tfrac{1}{k_{\mathrm{belt}}} \delta_{J_{Md}} J_{\mathrm{Md},n} \tfrac{\dot{\omega}_m}{k_{\mathrm{belt}}} ; \\
w_{b_{\mathrm{Md}}} &= -\tfrac{1}{k_{\mathrm{belt}}} \delta_{b_{\mathrm{Md},}} b_{\mathrm{Md},n} \tfrac{\omega_m}{k_{\mathrm{belt}}} + \delta_\mu \mu_n M_n g r_{\mathrm{Md}} sgn\left(\omega_m / k_{\mathrm{belt}}\right) \\
w_{ks} &= -\tfrac{1}{k_{\mathrm{belt}}} \delta_{k_s} k_{s,n} \int \left(\frac{\omega_m}{k_{\mathrm{belt}}} - \omega_{\mathrm{Ld}} \right) dt ; \\
w_{b_s} &= -\tfrac{1}{k_{\mathrm{belt}}} \delta_{b_s} b_{s,n} \left(\tfrac{\omega_m}{k_{\mathrm{belt}}} - \omega_{\mathrm{Ld}} \right)
\end{aligned} \tag{7.61}$$

Robust thresholds over the residual can be formed as (see Sect. 7.2.3),

$$-a_1 < r_1(t) < a_1 \tag{7.62}$$

where

$$a_1 = \sum |w_i| \tag{7.63}$$

Remark Only one I-ARR has been derived in this work. This serves the purpose of demonstration. Following similar steps, another independent ARR(s) can be derived.

Figure 7.11 shows the residual profile under nominal conditions, wherein the residual is well within the envelope formed by thresholds. Figure 7.12 shows the effect of adding load (or frictional toque) in a discrete way on the system. ω_{Md} is controlled at 30 rad/s. Addition of load leads increase in frictional torque and degradation in speed. Due to action of PI controller, the motor disk speed is maintained at set reference value of 30 rad/s. However, the residual $r_1(t)$ is sensitive to the variation in PI enabled input voltage U_{PI}. As such, the residual captures the variation of disk speed due to load suspension. Saturation limit U_{PI} is reached around $t = 65$ s when the total load suspended is 1.6 Kg. Thereafter, controller is unable to compensate the change in ω_{Md}. With addition of more load thereafter ($t > 65$ s), motor disk speed decreases rapidly and stops at around $t = 70$ s. For safety reasons, the disk is stopped momentarily, after which the suspended load is removed.

7.6.3 Degradation Model: Offline Phase

The experiments performed are non-destructive in nature. Here, load in form of sand of M Kg is suspended in a uniform manner until a prefixed limit of M_{fail} is reached. In this context, $M(t)$ is treated as a system parameter under degradation, the prognostic candidate.

The experiments were conducted in two distinct phases:

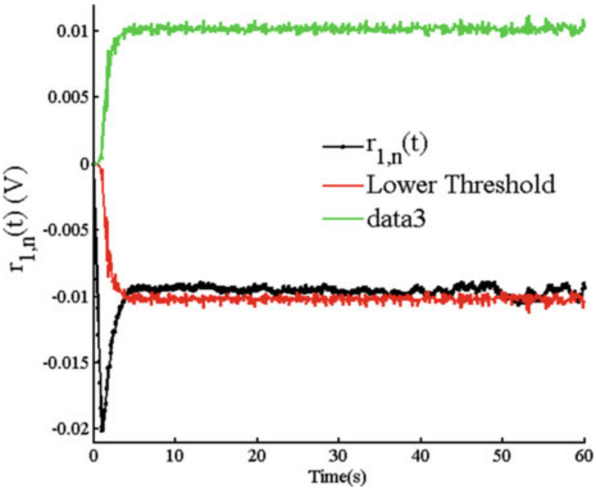


Fig. 7.11 Residual $r_1(t)$ under nominal conditions

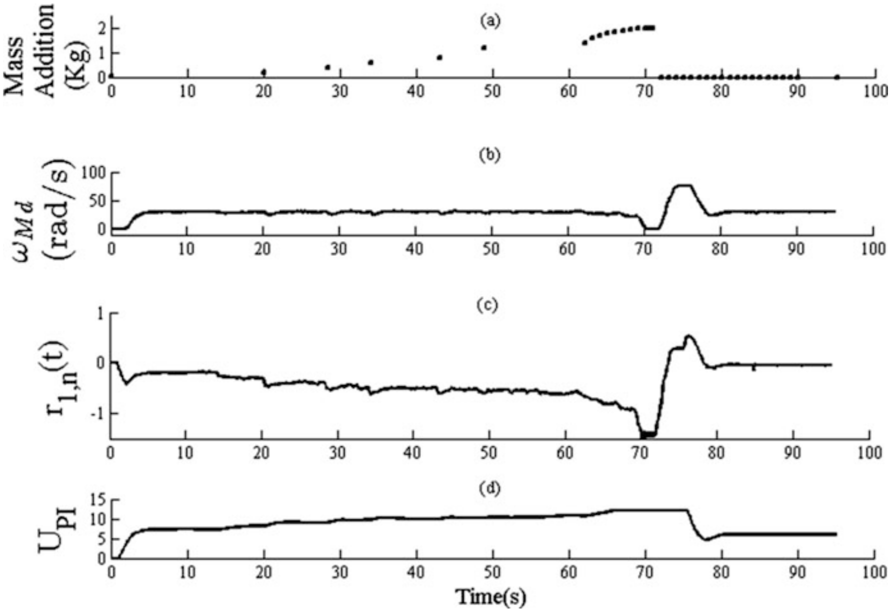


Fig. 7.12 (**a**) Addition of load (**b**). Motor disk speed (**c**) Nominal residual $r_1(t)$ (**d**) Input voltage (PI controlled)

- Offline Phase: Mass is suspended uniformly. As explained in Sect. 7.3.1, variations of $M(t)$ are obtained from the evolution of $r_1(t)$. Then, statistical techniques such as (curve fitting) are used to obtain DM of $M(t)$.
- Online health monitoring: In real time, load is added in a similar manner under similar environmental conditions as offline phase, until the prefixed failure value

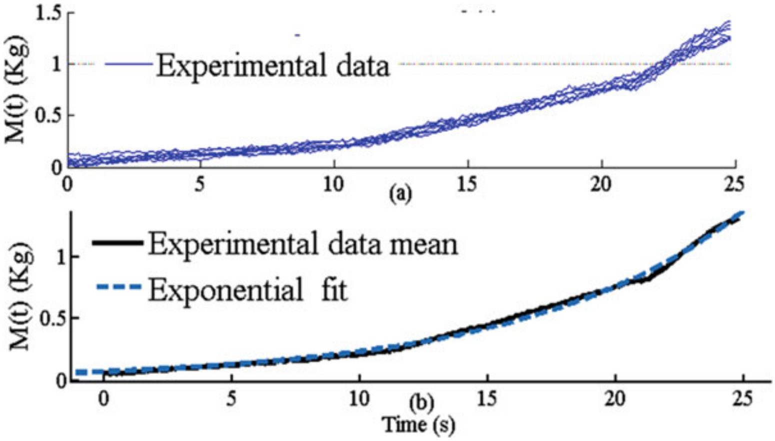


Fig. 7.13 Exponential variation of mass. (**a**) Experimental data. (**b**) Exponential fit over experimental data mean

$M(t)$ is reached. In real time, estimation of $M(t)$ and associated DPPs, and subsequent RUL predictions are obtained.

7.6.3.1 Exponential Variation of Load

Load is varied uniformly in an exponential manner. Eight experiments are carried out in total. Figure 7.13a shows the experimental data and Fig. 7.13b shows the exponential fit over the experimental data mean. This way, an exponential DM is obtained as,

$$\begin{aligned} b_{\mathrm{Md}}(t) &= g_1\left(M, \gamma_1\right) + v_{M1} \\ &= M_n \;\; e^{\gamma_1(t)} + v_{M1} \end{aligned} \tag{7.64}$$

where $g_1(.)$ is the DM, $\theta^d = M(t)$, DPP vector $\boldsymbol{\gamma}^d = \left[\gamma^d\right] = \gamma_1$, and normally distributed process noise $v_{M1}(t) \sim \mathcal{N}\left(0, \sigma_{M1}^2\right)$.

The DM provides an approximate true value of DPP, $\gamma_1^* = 0.05$ Kg/s. Regression residuals provide standard deviation of the process noise v_{M1}, $\sigma_{M1} = 8 \times 10^{-4}$ Kg.

7.6.4 *Health Monitoring: Online Phase*

In the online phase, environmental conditions are kept unaltered. It is recalled that $M(t)$ is treated as a system parameter under degradation, the prognostic candidate.

Failure value is prefixed at $M_{\text{fail}} = 1.8$ Kg. Load is varied in the similar manner until M_{fail} is reached.

7.6.5 Fault Model

In discrete time step k, the tuple $(g_1, M(t), \gamma_1)$ is formulated in state space as,

$$\begin{aligned} M_k &= M_{k-1} \times e^{\gamma_{1,k-1}\Delta t} + v_{M1,k-1} \\ \gamma_{1,k} &= \gamma_{1,k-1} + \xi_{1,k-1} \end{aligned} \tag{7.65}$$

where $\xi_{1,k} \sim \mathcal{N}\left(0, \sigma^2_{\xi_1}\right)$ is a normally distributed artificial random walk noise added to the DPP $\gamma_{1,k}$ for a suitable convergence. The magnitude of this noise should be sufficiently large for a desirable convergence of estimations and small enough for a good estimation accuracy. Usually, this noise is tuned with the help of simulations or multiple offline testing. Readers are referred to [36] for a simplified variance adaption scheme proposed in this context.

Observation equation is constructed from the ARR derived (Sect. 7.6.2) using the Theorem given in Sect. 7.3.4. The ARR : R_1 can be decomposed as,

$$R_1 : r_1(t) = r_{1,n}(t) + (M(t) - M_n) \times \frac{\partial\left(r_{1,n}(t)\right)}{\partial(M)} = 0 \tag{7.66}$$

Then, observation equation can be constructed as,

$$y_{1,k} = r_{1,n,k} + w_{1,k}(t) = (M_k - M_n) \left(\frac{\mu_n g \;\; r_{\text{Md}} \operatorname{sgn}\left(\omega_{\text{Md},k}\right)}{k_{\text{belt}}}\right) + w_{1,k} \tag{7.67}$$

so that the nominal part of the ARR $r_{1,n}(t)$ can be used to obtain the measurement of the state variables. Here, $w_{1,k} \sim \mathcal{N}\left(0, \sigma^2_{w1}\right)$ models noise manifesting in the residual measurements. Approximate value of σ_{w1} is determined from $r_1(t)$ values during degradation tests.

7.6.5.1 State of Health Estimation

Figure 7.14 shows the profile of residual under exponential degradation. The degradation initiation is detected when the residual goes outside the threshold envelope at around $t = 22$ s, after which prognostic module is triggered.

Estimation of SOH The estimation of suspended load $\widehat{M}$ is shown in Fig. 7.15. The estimation of SOH is performed with number of particles $N = 50$, sample time $\Delta t = 0.1$ s, initial random walk variance noise $\sigma^2_{\xi_1,k=0} = 4 \times 10^{-6}$, and standard deviation $\sigma_{w1} = 5 \times 10^{-3}$ V. For estimation of SOH, particle filter assumes

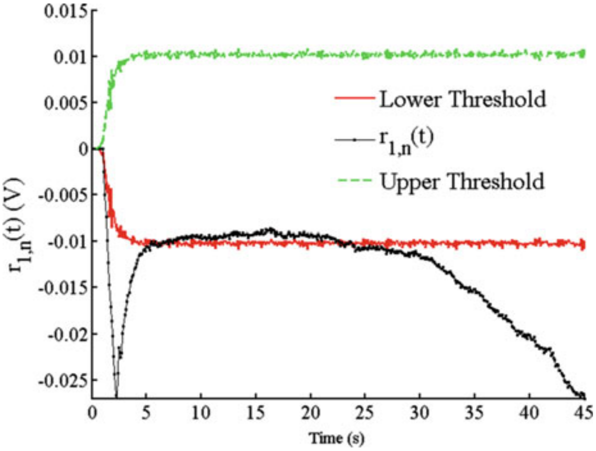


Fig. 7.14 Nominal residual $-r_{2,n}(t)$ while system is under degradation (exponential case)

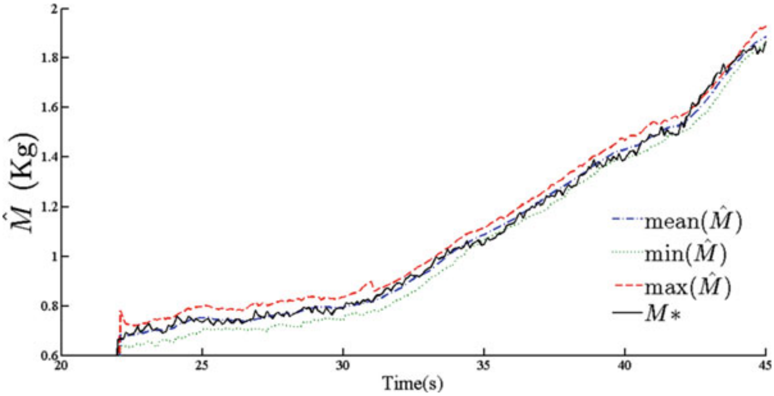


Fig. 7.15 Estimation of SOH of prognostic candidate

measurement noise variance nine times that of measurement variance σ_{w1}^2. This is done to counter sample *impoverishment* problem during the estimation process [17, 24]. The estimation is achieved with $\text{RMSE}_M = 3.78\%$. This indicates a high accuracy in estimation performance.

Figure 7.16 shows the estimation of DPP γ_1, achieved with $\text{RMSE}_{\gamma 1} = 7.6\ \%$. The convergence is achieved very quickly with large initial estimation spread. This is due to a high artificial noise variance set for the desirable quick convergence. It should be noted that RMSE_γ obtained via experiments is higher than those obtained via simulations, as the true speed of degradation γ_1^* does not remain perfectly constant in reality. Also, lesser number of particles are employed here used so that RUL predictions may be achieved in real time without significant data loss. With higher number of particles, greater accuracy may be achieved.

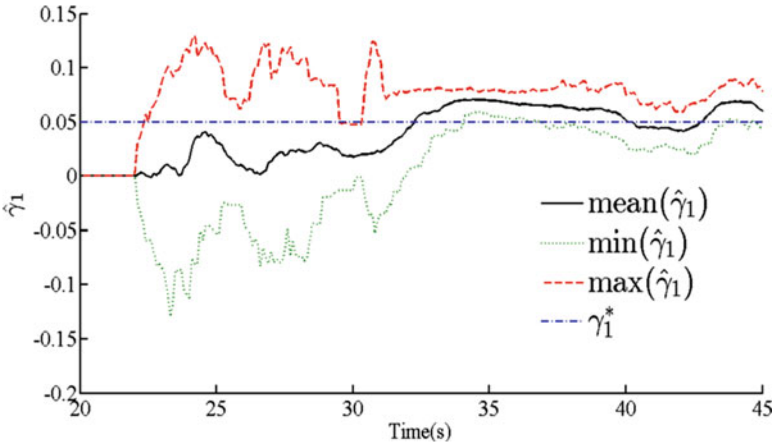


Fig. 7.16 Estimation of DPP

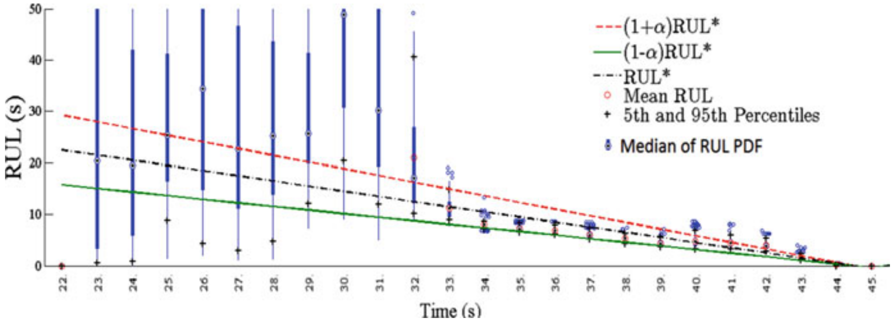


Fig. 7.17 RUL predictions

Figure 7.17 shows the RUL prediction with $\alpha = 0.2$.The RUL distributions obtained until $t = 32$ s are not good predictions and suffer with large variance spread due to a large corresponding spread in $\widehat{\gamma}_1$ (see Fig. 7.16). This makes their utility virtually null. However, after $t = 32$ s, with significant improvement in estimation of DPP, the RUL distributions are well within accuracy cone such that more than 50 % of RUL probability mass lies within accuracy cone. Ignoring the initial period of convergence, the overall prediction performance is obtained with $\overline{\text{RA}} = 97.02\%$.

7.7 Conclusions

Prognostics is the science of assessing the end of life of a system/component and prediction of the remaining useful life of the same. Due to various kinds of uncertainties that manifest in form of parametric uncertainties, environmental conditions, sensor noises, uncertain future conditions, etc., RUL prediction becomes a very challenging issue. In this work, benefits of BG-LFT modelling based robust fault detection are integrated with the advantages of Bayesian estimation method for efficient prognostics. In particular, particle filters are exploited for optimal estimations of actual state of the prognostic candidate and subsequent RUL predictions. In this chapter, a single system parameter is chosen as the prognostic candidate and RULs are obtained with respect to that parameter. As such, the work presented here paves the future path for development of efficient system level prognostics in BG framework.

The ARR based BG-LFT technique is employed for robust detection of degradation beginning. The same ARR is then exploited for prognostic purposes. Being sensitive to the control inputs, nominal residual is able to capture the parametric degradation profile even while the system outputs remain in feedback closed loop regime. This aspect renders the approach appropriate for system level health management. Approximation of noise distribution present in residuals can be difficult or impossible, due to presence of derivative or integral terms in the ARR function arguments. As such, particle filter algorithms form the best choice in this regard as they are not restricted by non-Gaussian noises. Moreover, degradation of non-linear nature can be efficiently estimated using particle filters. Additionally, this method also demonstrates that fusion of BG-LFT framework and Monte Carlo framework leads to efficient management of various types of uncertainties. While parametric uncertainties are modelled and managed by using BG-LFT for efficient detection of degradation initiation; degradation process noise, measurement (residual) noise, etc., are efficiently accounted for, by PF for estimation of SOH and RUL predictions.

References

1. Abbas, M., Ferri, A. A., Orchard, M. E., & Vachtsevanos, G. J. (2007). An intelligent diagnostic/prognostic framework for automotive electrical systems. *Intelligent Vehicles Symposium, 2007 IEEE*. IEEE, pp. 352–357.
2. An, D., Choi, J.-H., & Kim, N. H. (2013). Prognostics 101: A tutorial for particle filter-based prognostics algorithm using Matlab. *Reliability Engineering and System Safety, 115*, 161–169.
3. An, D., Kim, N. H., & Choi, J.-H. (2015). Practical options for selecting data-driven or physics-based prognostics algorithms with reviews. *Reliability Engineering and System Safety, 133*, 223–236.
4. Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing, 50*(2), 174–188.
5. Baraldi, P., Mangili, F., & Zio, E. (2013). Investigation of uncertainty treatment capability of model-based and data-driven prognostic methods using simulated data. *Reliability Engineering and System Safety, 112*, 94–108.
6. Blischke, W. R., & Murthy, D. P. (2011). *Reliability: Modeling, prediction, and optimization* (Vol. 767). New York, NY: John Wiley & Sons.
7. Borutzky, W. (2015). Failure prognosis for hybrid systems based on ARR residuals. In W. Borutzky (Ed.), *Bond graph model-based fault diagnosis of hybrid systems* (pp. 221–233). New York: Springer.

8. Box, G. E., Jenkins, G. M., & Reinsel, G. C. (2011). *Time series analysis: Forecasting and control* (Vol. 734). Hoboken, NJ: John Wiley & Sons.
9. Bregon, A., Daigle, M., & Roychoudhury, I. (2012). An integrated framework for model-based distributed diagnosis and prognosis. *DTIC Document*.
10. Byington, C. S., & Roemer, M. J. (2002). Prognostic enhancements to diagnostic systems for improved condition-based maintenance [military aircraft]. *Aerospace Conference Proceedings, 2002. IEEE*. IEEE, vol. 2816, pp. 6-2815-2816-2824.
11. Cadini, F., Zio, E., & Avram, D. (2009). Monte Carlo-based filtering for fatigue crack growth estimation. *Probabilistic Engineering Mechanics, 24*(3), 367–373.
12. Celaya, J., Kulkarni, C., Biswas, G., Saha, S., & Goebel, K. (2011). A model-based prognostics methodology for electrolytic capacitors based on electrical overstress accelerated aging. In *Annual Conference of the Prognostics and Health Management Society (PHM),* Montreal.
13. Daigle, M. J., Bregon, A., & Roychoudhury, I. (2014). Distributed prognostics based on structural model decomposition. *IEEE Transactions on Reliability, 2*(63), 495–510.
14. Daigle, M., & Goebel, K. (2009). Model-based prognostics with fixed-lag particle filters. *Conference of the PHM Society*.
15. Daigle, M., & Goebel, K. (2010). Model-based prognostics under limited sensing. *Aerospace Conference, 2010 IEEE*. IEEE, pp. 1–12.
16. Daigle, M. J., & Goebel, K. (2011). A model-based prognostics approach applied to pneumatic valves. *International Journal of Prognostics and Health Management, 2*(2), 1–16.
17. Daigle, M. J., & Goebel, K. (2011). A model-based prognostics approach applied to pneumatic valves. *International Journal of Prognostics and Health Management, 2*(color), 84.
18. Daigle, M. J., & Goebel, K. (2013). Model-based prognostics with concurrent damage progression processes. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 43*(3), 535–546.
19. Daigle, M., Saha, B., & Goebel, K. A. (2012). Comparison of filter-based approaches for model-based prognostics. *Aerospace Conference, 2012 IEEE*. IEEE, pp. 1–10.
20. Djeziri, M., Ananou, B., & Ouladsine, M. (2013). Data driven and model based fault prognosis applied to a mechatronic system. *Power Engineering, Energy and Electrical Drives (POWERENG), 2013 Fourth International Conference on.* IEEE, pp. 534–539.
21. Djeziri, M. A., Bouamama, B. O., Dauphin-Tanguy, G., & Merzouki, R. (2011). LFT bond graph model-based robust fault detection and isolation. In W. Borutzky (Ed.), *Bond graph modelling of engineering systems* (pp. 105–133). New York: Springer.
22. Djeziri, M. A., Merzouki, R., Bouamama, B. O., & Dauphin-Tanguy, G. (2007). Robust fault diagnosis by using bond graph approach. *IEEE/ASME Transactions on Mechatronics, 12*(6), 599–611.
23. Djeziri, M., Toubakh, H., & Ouladsine, M. (2013) Fault prognosis based on fault reconstruction: Application to a mechatronic system. *Systems and Control (ICSC), 2013 3rd International Conference on*. IEEE, pp. 383–388.
24. Douc, R., & Cappé, O. (2005). Comparison of resampling schemes for particle filtering. *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*. IEEE, pp. 64–69.
25. Doucet, A., De Freitas, N., & Gordon, N. (2001). *An introduction to sequential Monte Carlo methods*. New York, NY: Springer.
26. Doucet, A., & Johansen, A. M. (2009). A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering, 12*, 656–704.
27. Engel, S. J., Gilmartin, B. J., Bongort, K., & Hess, A. (2000). Prognostics, the real issues involved with predicting life remaining. *Aerospace Conference Proceedings, 2000 IEEE.* IEEE, pp. 457–469.
28. Gebraeel, N., & Pan, J. (2008). Prognostic degradation models for computing and updating residual life distributions in a time-varying environment. *IEEE Transactions on Reliability, 57*(4), 539–550.
29. Guo, H., & Liao, H. (2015). Practical approaches for reliability evaluation using degradation data. *Annual Reliability and Maintainability Symposium*.

30. Helton, J. C. (1993). Uncertainty and sensitivity analysis techniques for use in performance assessment for radioactive waste disposal. *Reliability Engineering and System Safety, 42*(2), 327–367.
31. Herzog, M. A., Marwala, T., & Heyns, P. S. (2009). Machine and component residual life estimation through the application of neural networks. *Reliability Engineering and System Safety, 94*(2), 479–489.
32. Hu, Y., Baraldi, P., Di Maio, F., & Zio, E. (2015). A particle filtering and kernel smoothing-based approach for new design component prognostics. *Reliability Engineering and System Safety, 134*, 19–31.
33. Hitchcock, L. (2006). ISO standards for condition monitoring. In *Engineering asset management* (pp. 606–613). London: Springer.
34. Jardine, A. K., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing, 20*(7), 1483–1510.
35. Jha, M., Dauphin-Tanguy, G., & Ould Bouamama, B. (2014). Integrated diagnosis and prognosis of uncertain systems: A bond graph approach. *Second European Conference of the PHM Society 2014-Nantes France, 2014. European Conference of the PHM Society 2014 Proceedings*, pp. 391–400.
36. Jha, M. S., Dauphin-Tanguy, G., & Ould-Bouamama, B. (2016). Particle filter based hybrid prognostics for health monitoring of uncertain systems in bond graph framework. *Mechanical Systems and Signal Processing, 75*, 301–329.
37. Jouin, M., Gouriveau, R., Hissel, D., Péra, M.-C., & Zerhouni, N. (2014). Prognostics of PEM fuel cell in a particle filtering framework. *International Journal of Hydrogen Energy, 39*(1), 481–494.
38. Kleijn, C. (2011). *20 SIM 4C 2.1 Reference Manual*. Enschede, Controllab Products B.V. ISBN 978-90-79499-14-4.
39. Klutke, G.-A., Kiessler, P. C., & Wortman, M. (2003). A critical look at the bathtub curve. *IEEE Transactions on Reliability, 52*(1), 125–129.
40. Krantz, S. G., & Parks, H. R. (2012). *The implicit function theorem: History, theory, and applications*. New York, NY: Springer.
41. Kuehl, R. W. (2010). Using the Arrhenius equation to predict drift in thin film resistors. *Proceedings CARTS Europe*, pp. 121–133.
42. Kulkarni, C. S., Celaya, J., Biswas, G., & Goebel, K. (2012). Physics based electrolytic capacitor degradation models for prognostic studies under thermal overstress. In *First European Conference of the Prognostics and Health Management Society* (Vol. 3, p. 9). Dresden.
43. Lee, J., Ni, J., Djurdjanovic, D., Qiu, H., & Liao, H. (2006). Intelligent prognostics tools and e-maintenance. *Computers in Industry, 57*(6), 476–489.
44. Li, T., Sun, S., Sattar, T. P., & Corchado, J. M. (2014). Fight sample degeneracy and impoverishment in particle filters: A review of intelligent approaches. *Expert Systems with applications, 41*(8), 3944–3954.
45. Liao, S.-H. (2005). Expert system methodologies and applications—a decade review from 1995 to 2004. *Expert Systems with Applications, 28*(1), 93–103.
46. Lu, J.-C., Park, J., & Yang, Q. (1997). Statistical inference of a time-to-failure distribution derived from linear degradation data. *Technometrics, 39*(4), 391–400.
47. Maricau, E., & Gielen, G. (2009). A methodology for measuring transistor ageing effects towards accurate reliability simulation. *On-Line Testing Symposium, 2009. IOLTS 2009. 15th IEEE International*. IEEE, pp. 21–26.
48. Medjaher, K., & Zerhouni, N. (2009). Residual-based failure prognostic in dynamic systems. *7th IFAC International Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFE PROCESS'09, vol sur CD ROM. IFAC*, p. 6.
49. Medjaher, K., & Zerhouni, N. (2013). Hybrid prognostic method applied to mechatronic systems. *The International Journal of Advanced Manufacturing Technology, 69*(1–4), 823–834.

50. Orchard, M. E. (2007). A particle filtering-based framework for on-line fault diagnosis and failure prognosis. *Georgia Institute of Technology*.
51. Orchard, M., Kacprzynski, G., Goebel, K., Saha, B., & Vachtsevanos, G. (2008). Advances in uncertainty representation and management for particle filtering applied to prognostics. *Prognostics and health management, 2008. phm 2008. International conference on*. IEEE, pp. 1–6.
52. Plett, G. L. (2004). Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 3. State and parameter estimation. *Journal of Power sources, 134*(2), 277–292.
53. Rausand, M., & Høyland, A. (2004). *System reliability theory: Models, statistical methods, and applications* (Vol. 396). Hoboken, NJ: John Wiley & Sons.
54. Roychoudhury, I., & Daigle, M. (2011). An integrated model-based diagnostic and prognostic framework. *Proceedings of the 22nd International Workshop on Principle of Diagnosis (DX'11)*, Murnau, Germany.
55. Saha, B., & Goebel, K. (2009). Modeling Li-ion battery capacity depletion in a particle filtering framework. *Proceedings of the annual conference of the prognostics and health management society*, pp. 2909–2924.
56. Saha, B., Goebel, K., Poll, S., & Christophersen, J. (2009). Prognostics methods for battery health monitoring using a Bayesian framework. *IEEE Transactions on Instrumentation and Measurement, 58*(2), 291–296.
57. Saha, B., Goebel, K., & Christophersen, J. (2009). Comparison of prognostic algorithms for estimating remaining useful life of batteries. *Transactions of the Institute of Measurement and Control, 31*(3/4), 293–308.
58. Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2009). Evaluating algorithm performance metrics tailored for prognostics. In *2009 IEEE Aerospace Conference* (pp. 1–13). IEEE.
59. Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2010). Metrics for offline evaluation of prognostic performance. *International Journal of Prognostics and Health Management, 1*(1), 21.
60. Sikorska, J., Hodkiewicz, M., & Ma, L. (2011). Prognostic modelling options for remaining useful life estimation by industry. *Mechanical Systems and Signal Processing, 25*(5), 1803–1836.
61. Tsui, K. L., Chen, N., Zhou, Q., Hai, Y., & Wang, W. (2015). Prognostics and health management: A review on data driven approaches. *Mathematical Problems in Engineering, 2015*, 79316.
62. Vachtsevanos, G., Lewis, F., Roemer, M., Hess, A., & Wu, B. (2007). *Intelligent fault diagnosis and prognosis for engineering systems*. Hoboken, NJ: John Wiley & Sons.
63. Wu, W., Hu, J., & Zhang, J. (2007) Prognostics of machine health condition using an improved arima-based prediction method. *Industrial Electronics and Applications, 2007. ICIEA 2007. 2nd IEEE Conference on*. IEEE, pp. 1062–1067.
64. Yu, M., Wang, D., Luo, M., & Huang, L. (2011). Prognosis of hybrid systems with multiple incipient faults: Augmented global analytical redundancy relations approach. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 41*(3), 540–551.
65. Zio, E., & Peloni, G. (2011). Particle filtering prognostic estimation of the remaining useful life of nonlinear components. *Reliability Engineering and System Safety, 96*(3), 403–409.

Part III
Thermodynamic Systems and Distributed Parameter Systems

Bond graph modelling of thermodynamic systems as part of an overall system and the modelling of mechanical subsystems to be considered as distributed parameter systems have been addressed in many publications in the past. Nevertheless, both topics are still subject of ongoing research.

Modelling open thermodynamic systems, especially those including compressible fluid and phase changes, has often been considered difficult. Accordingly, various approaches in the bond graph framework are known. To be closer to familiar notions of temperature, heat flow and relations used in practice, one approach dismisses the fundamental principle of energy exchange between ports and adopts temperature and heat flow as the covariables of a bond. Besides an easier understanding another advantage is that much of the bond graph methodology can be retained with regard to the systematic construction of so-called pseudo-bond graphs, the assignment of causalities and the derivation of equations. A disadvantage is that pseudo BGs cannot easily be coupled with standard BGs of other parts of an overall system.

Alternatively, since the beginning of bond graph modelling, bond graphs often termed true bond graphs have been developed for thermodynamic problems that use temperature and the time derivative of entropy as effort and flow variables for heat transfer and comply with physical conservation laws.

Instead of accounting for the different energy components carried by a flow between two ports by four conventional bonds, a third approach introduced by Professor Brown quite some time back in the past and recalled in the first chapter of this part uses so-called *convection* bonds with *two* efforts and one flow. By ignoring heat conduction through a port, the three remaining components of energy propagation in a fluid flow are described by three instead of six variables in a convection BG which leads to less complex bond graph representations of thermodynamic systems. It appears that convection BGs haven't been much used in the past because there was no software supporting them so that they can be used in conjunction with conventional bond graphs for other parts of an overall system. Ultimately, Professor Brown's efforts have resulted in such a simulation software called THERMOSIM® and a user-friendly graphical interface BondGrapher® that allows for a *compatible* mix of conventional BGs and convection BGs, thus offering a novel convenient and promising mean to include thermodynamic subsystems with compressible fluid flow into the bond graph modelling of an overall system. Chapter 8 presents the theoretical basis as well as the newly developed supporting software.

Chapter 9 develops a *true multibond graph* for a single-phase, single-component, multidimensional compressible fluid flow that accounts for viscosity and thermal effects. The rigorous theoretical step-by-step development starts from the *total* energy per volume as a function of independent variables. Its time derivative is formulated as a sum of products of generalised potentials and the time derivatives of the independent variables. Balance equations are derived from conservation laws that are formulated as Partial Differential Equations (PDEs). A space discretisation is obtained by application of the Finite Element Method (FEM), i.e. by introducing time-dependent nodal values of the independent variables and position-dependent interpolation functions (shape functions) and by introducing nodal state variables and nodal potential. The resulting equations can be represented by a multibond graph that expresses the conservation equations for the nodal values of linear momentum, mass and entropy. The presentation in Chap. 9 demonstrates that it is not necessary to escape to a pseudo-BG representation of a thermofluid subsystem, that a true BG can be developed and that BG modelling can be combined with widely used Computational Fluid Dynamics (CFD) techniques.

Many mechanical systems incorporate parts that are to be considered as *distributed parameter systems*, e.g. the arm of a robot, the blades of a wind turbine or of a helicopter, or the outrigger of a crane. The dynamic behaviour of such a subsystem is mathematically given by a boundary value problem. Often a cantilever beam is considered and either the Euler-Bernoulli or the Timomoshenko theory is applied. Two approaches, namely the modal analysis and the finite element discretisation are well known. Results of both approaches lead to a lumped parameter approximation that can be well represented by a bond graph and thus can be used in a bond graph model of an overall system. For both approaches the model complexity necessary to ensure a sufficient accuracy is an important question. As to the modal analysis, the user defined frequency range of interest (FROI) determines the frequencies that are important for a specific scenario, i.e. modes with frequencies within that range are retained in a model of *reduced order* and modes outside this range are eliminated. For a finite element approach, the model accuracy improves with an increase of the number of segments. However, it is difficult to predict the number of segments required to achieve a required accuracy.

An approach aiming at a lumped parameter model of *reduced order* developed in the past and promoted by the author of the third chapter in Part III and other researchers is to consider all energetic elements in a lumped parameter model, to quantify their contribution by a so-called *activity metric*, and to eliminate elements that do not contribute significantly to the dynamic behaviour. The approach is based on a model reduction algorithm that is supported by software. As to distributed parameter systems, the activity metric-based model reduction algorithm can be used for a finite segment approximation and can be used to determine when and in which segment shear and rotary inertial effects of the Timoshenko theory are to be included in the model in order to ensure a sufficient accurate prediction of the dynamic behaviour.

The use of an energy-based activity metric in order to decide which energetic elements are to be kept in a lumped parameter model suggests to use a bond graph representation from which needed power variables can be easily computed.

Chapter 8
Convection Bond Graphs for Thermodynamic Systems

Forbes T. Brown

8.1 Introduction

In his development of bond graphs, Henry Paynter focused on systems most familiar to engineers who dealt with the dynamics of mechanical and electrical systems, in effect postponing consideration of thermodynamic systems which are inherently more difficult. Nevertheless, he envisioned these graphs to express the dynamics of general physical systems in a graphical language based on the principles of energy. Thermodynamics is the science of energy, so the author believes that inclusion of thermodynamic systems into the scope of bond graphs is necessary for these graphs to assume the status of a universal language for the dynamics of physical systems. The challenge is that conventional bonds with their single effort and single flow variables are very awkward when dealing with the flows of highly compressible fluids.

One result of the enormous complexity of thermodynamic systems has been emphasis on steady-state behavior. Many engineers and scientists devote their careers to expanding knowledge of heat transfer or the thermodynamic and transport properties of the substances employed in thermodynamic systems. Since bond graphs are used almost exclusively for dynamic behavior, this means that few people active in the field are conversant with thermodynamics. In order to add the additional layer of dynamics to thermodynamic analysis effectively requires a willingness to forgo a degree of accuracy in the treatment of the steady-state behavior, which the author long ago accepted as a necessary compromise. It is now time for the bond graph community to roll up its sleeves and dig in to the field of thermodynamic systems, with the aim of expediting the approximate modeling and simulation of their dynamic behavior, which can be crucial for design success.

F.T. Brown (✉)
Lehigh University, Bethlehem, PA, USA
e-mail: ftbmhb@aol.com

W. Borutzky (ed.), *Bond Graphs for Modelling, Control and Fault Diagnosis of Engineering Systems*, DOI 10.1007/978-3-319-47434-2_8

The physical property of entropy and its corollary exergy—sometimes called “availability”—is essential to thermodynamic systems, but not to the mechanical and electrical systems originally addressed by Paynter. Attempts to shoehorn thermodynamic models with entropy into conventional bond graphs have resulted, in the authors opinion, in discouraging and unnecessary complexity. The author’s approach is to extend the bond graphs themselves by allowing a new kind of bond called a convection bond [1, 5], and new elements which employ these bonds, often in conjunction with the traditional—“simple”—bonds. The result can be called general bond graphs; bond graphs with no convection bonds become a subset of these, which could be called “traditional” bond graphs. The presentation below starts with an introduction to convection bonds and elements that employ them, and proceeds to the determination of the properties of thermodynamic substances before addressing practical systems.

THERMOSIM® [8], which employs a graphical interface called BondGrapher, is available without charge on the author’s website [9], also on which a radically revised Users’ Manual should become available before this book is published.

8.2 General Bond Graphs

The conventional or simple bond was conceived to represent the interconnection between two physical entities so as to represent the “effort” and “flow” variables which when multiplied together give the instantaneous power being transferred. Examples include force and velocity, electric voltage and current, and pressure and—for incompressible fluids—the volume flow rate. Four such pairs are needed to completely represent the flow of a compressible pure substance through a port, three of which are significant: pressure and the volume flow rate for the “flow power,” internal energy and the mass flow rate for the transport of internal energy, and kinetic energy and the mass flow rate for the transport of kinetic energy. The fourth pair is absolute temperature and entropy flux for heat conduction through the port. This fourth mode is so small, due to a small temperature gradient, it is almost universally ignored in the examples given in thermodynamics textbooks, and the author also ignores it. Heat transfer through conduction from the fluid to its surroundings, transverse to the flow, is a totally different issue, however; it is central to the operation of nearly all thermodynamic systems.

What the author has done is combine the three significant modes of energy propagation for flow through a port, with its three bonds, into a single bond with *two* independent efforts and one flow, replacing six bond variables with only three. A great simplification results. The flow is the mass flow rate, $\dot{m}$. Ignoring for the moment the sometimes small mode for transport of kinetic energy, the power factor effort is the enthalpy of the fluid, h. Enthalpy is the sum of the internal energy, u, and the product of the absolute pressure, P, and the specific volume, v, or inverse of density. The first mode gives the transport of internal energy and the second gives the pressure-flow power. According to the state postulate of thermodynamics, the

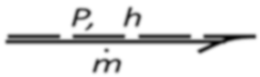


Fig. 8.1 The convection bond

state of a pure substance can be expressed as a function of any pair of independent intensive properties; enthalpy is used here as one of them. The other one most commonly used is the pressure or the stagnation pressure, because it is the causal effort. Other pairs of intensive properties also substitute for particular purposes; many are routinely computed. Entropy, s, is one of them; it plays a particularly important role in characterizing how much of the power is available as work—the "exergy." The transport of kinetic energy is added to the other two modes simply by replacing the nominal enthalpy with the stagnation enthalpy. The difference between the two can be computed readily using in part the bond variable of mass flow rate.

The resulting "convection bond" is distinguished from a simple bond by the use of a second bond line, drawn dashed as shown in Fig. 8.1. The use of convection bonds implies the introduction of a small number of new bond graph elements, most of which have both conventional and convection ports. From the perspective of thermodynamics, conventional bonds could be considered to be a special case of convection bonds in which only the power factor effort variable is used and the flow variable is allowed to be other than a mass flow rate.

20 bond graph elements are employed. The conventional elements include the effort source (Se), the flow source (Sf), the resistance (R), the compliance (C), the inertance (I), the transformer (T or TF), the gyrator (G or GY), the common-effort junction (0), and the common-flow junction (1). Causal stokes are employed in the same manner as with simple bonds. Linearized analysis is not attempted in view of the highly nonlinear behavior of the primary applications. Elements that may include convection ports are discussed in the following section, along with others that have simple ports involving heat conduction. Relevant information not cited elsewhere herein is also available [2, 7], including a PowerPoint survey [6] that predates BondGrapher.

For lack of space this chapter does not discuss the multiport compliance Cf element [7], which in THERMOSIM has a unique ability to deduce differential equations when the only information provided by the user is relations for stored energy. This is most relevant to electromechanical systems, which are not the focus here.

8.2.1 The CS Element

The compliance element C in non-thermodynamic systems represents a storage of energy as a function of the time integral of the flow on the attached bond, the state variable called its displacement. A compliance element for a chamber containing a potential two-phase mixture of a liquid and a vapor has energy that is a function

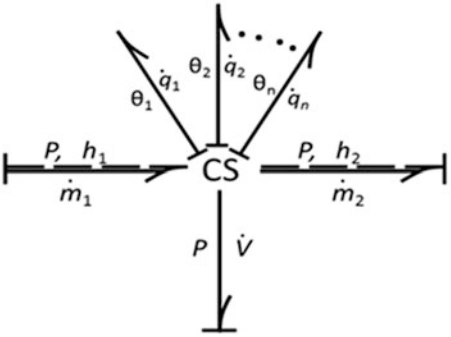


Fig. 8.2 The CS element for a thermodynamic system comprising a compressible fluid in a control volume or chamber

of two state variables, for practical reasons the specific volume of the fluid and its absolute temperature. This chamber is represented by the CS element, shown in Fig. 8.2. It is the workhorse of thermodynamic system models.

In the THERMOSIM® simulation package, zero, one, or two convection bonds—or fluid channels—may be attached to the element. The causal stokes on these bonds are always at the remote ends of their bonds; the mass flows are causal inputs and the pressure is a causal output, but not the only causal output. In addition, if the volume of the chamber is changeable, as by the motion of a piston, a simple bond is drawn with the effort P and flow $\dot{V}$, which is the rate of change of the volume. Here the pressure is again the causal output—the only one. Finally, any number of heat conduction bonds may be attached. These are also represented by simple bonds, but are distinguished from the volume change bond by their causal stokes which are always adjacent to the element. The efforts on these bonds are the external temperatures which when compared with the fluid temperature drives the heat conduction. The thermal "resistance" to these heat flows are part of the CS element itself; the element is technically a macro element that contains both reversible elements and irreversible elements—and is so represented in the author's textbook—but the user does not need to know this. In keeping with the traditional use of power-factoring variables for simple bonds, the flow on a heat conduction bonds is the flux of entropy that it represents. Since most people prefer to deal with heat rates directly, however, the author employs the popular notation of quasi-bond graphs, labeling the flow as the heat transfer itself.

There are three state variables associated with the CS element: the mass of the fluid therein, the volume of the chamber, and the absolute temperature of the fluid. For purposes of this chapter they are called m_i, V_i, and T_i, respectively. Note that the specific volume of the fluid, v_i, equals the ratio V_i/m. These are used because most available analytical relations for thermodynamic states are functions of v_i and T_i.

The constitutive equations for these three variables are as follows:

$$\frac{dm_i}{dt} = \dot{m}_{\text{in}i} - \dot{m}_{\text{out}i} \tag{8.1}$$

Table 8.1 Meanings of symbols in Eqs. (8.1), (8.2), and (8.3)

$\dot{m}_i$	Mass flow rate (with various subscripts), kg/s
t	Time, seconds
V_i	Volume of the *i*th CS element, m^3
m_i	Mass of the fluid in the *i*th CS element, kg
T_i	Temperature of the fluid in the *i*th CS element, K
P_i	Pressure of the fluid in the *i*th CS element (either static or dynamic), Pa
QQ$_i$	Net heat flow into the *i*th CS element, J/s
v_i	Specific volume of the fluid in the ith CS element, equal to V_i/m_i, m^3/kg
h_i	Enthalpy of the fluid in the *i*th CS element, J/kg
$dudT_i$	Gradient of internal energy with respect to temperature, J/kg/K
$dudv_i$	Gradient of the internal energy with respect to specific volume, J/m^3

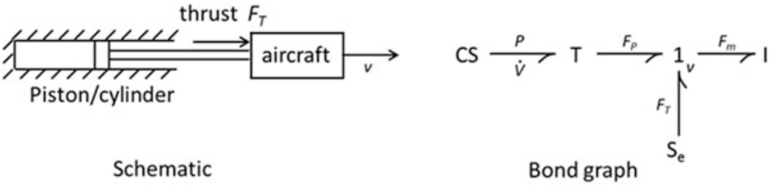


Fig. 8.3 Model of a steam catapult for an aircraft on a carrier, including engine thrust

$$\frac{dV_i}{dt} = fb_i \qquad \text{(the externally causal flow)} \tag{8.2}$$

$$\begin{aligned}\frac{dT_i}{dt} &= \left[\mathrm{QQ}_i + (P_i + \mathrm{dudv}_i)\left(v_i + \frac{dm_i}{dt} - Vd_i\right)\right] / \left(m_i\,\mathrm{dudT}_i\right)\\ &\quad + \left[(h_{\mathrm{in}i} - h_{\mathrm{out}i})\,\dot{m}_{\mathrm{in}i} - (h_{\mathrm{out}i} - h_i)\,\dot{m}_{\mathrm{out}i}\right] / \left(m_i\,\mathrm{dudT}_i\right)\end{aligned} \tag{8.3}$$

The symbols as used here are defined in Table 8.1. Implementing them properly in THERMOSIM with the relations to all the other elements in a system requires a couple 1000 lines of code and a more sophisticated nomenclature in the principal simulation program *convec2.m*, a very different story from conventional bond graphs and bond graph elements. A major reason for the complexity is the frequent need to employ analytical Runge–Kutta iterations when the two known independent thermodynamic variables are a different pair, such as pressure and enthalpy, or entropy and pressure.

A very simple example of the use of a CS element is for a model of a steam catapult such as used on an aircraft carrier, shown in Fig. 8.3. Since the fluid system which contains the steam has no active fluid ports or significant heat conduction during the takeoff run, the only bond to the CS element is the one for the expansion of the chamber. The CS element is the only non-traditional part of the model. Convection bonds and elements can be mixed with simple bonds and elements—of course following certain rules.

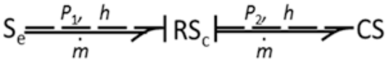


Fig. 8.4 A tank being charged with a fluid from a source through a restriction

8.2.2 The RSc Element

A model for an independent source of fluid—in effect a chamber of infinite size—connected to a fluid chamber through a restriction such as valve or an orifice is shown in Fig. 8.4. The source is called the convection Se element, sometimes written as Sec. The restriction is called the convection RSc element, which is highly irreversible despite the fact that the power flows on the two bonds are identical and equal to $h\dot{m}$. In THERMOSIM the RSc element is treated as an equivalent orifice with upstream flow that is reversible but an expansion that is not. The fluid upstream and/or downstream could be a saturated mixture of vapor and liquid, which introduces a complication because the effect of gravity on whichever chamber is upstream—has the higher pressure—may separate the liquid from the vapor, depending on the elevation of the physical port. One also does not know in general from which end the fluid is emitted; it may even change in the middle of a run. Therefore, the user must choose one of the following nine possibilities:

(1) No gravity separation for liquid or vapor. (This is the default choice.)
(2) Only vapor passes into the input side (for forward flow); there is no restriction for flow passing into the output side (for reverse flow).
(3) Only liquid passes into the input side (for forward flow); there is no restriction for flow passing into the output side (for reverse flow).
(4) Only vapor passes into the output side (for reverse flow); there is no restriction for flow passing into the input side (for forward flow)
(5) Only liquid passes into the output side (for reverse flow); there is no restriction for flow passing into the input side (for forward flow).
(6) Only vapor passes in both directions.
(7) Only liquid passes into the input side (for forward flow); only vapor passes into the output side (for reverse flow).
(8) Only vapor passes into the input side (for forward flow); only liquid passes into the output side (for reverse flow).
(9) Only liquid passes in both directions.

THERMOSIM employs a special function file *convecRS2.m* to compute the mass flow rate and enthalpy of the flow through a general compressible fluid through a vena contracta, which the user characterizes by an equivalent orifice area. The flow upstream of the vena contracta is assumed to be adiabatic and isentropic (reversible with constant entropy). The expanding flow downstream of the vena contracta is assumed to have the downstream pressure, and therefore is highly irreversible. The flow at the throat can be subsonic or choked. The function file has as independent

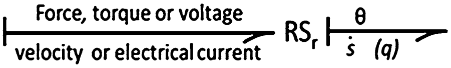


Fig. 8.5 The work-to-heat element RSr

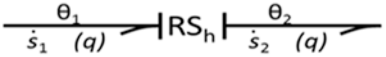


Fig. 8.6 The RSh element for heat conduction

arguments, in addition to the area and the gravity separation index, the upstream and downstream pressures and 14 different thermodynamic properties of the upstream fluid. Over 200 lines of code are required, plus calls to ancillary files, to compute the local properties of the substance within the element.

8.2.3 The RSr Element

The traditional use of bond graphs assumes all elements except the one-port “resistance” *R* conserve energy. The *R* element “dissipates” energy, making the thermal energy that it creates unavailable to the rest of the system. Several authors have remedied this limitation by replacing the *R* element with the two-port work-to-heat element RS element, or as given in THERMOSIM, the RSr element. In Fig. 8.5, the mechanical or electrical power on the left-hand bond that is converted to heat emerges on the right-hand bond as heat. As above, the effort variable on the heat conduction bond is absolute temperature, and the true flow variable is the entropy flux. Nevertheless, to satisfy the common practice, the flow on the heat conduction bond is treated in THERMOSIM as heat flux itself—given as *q*, the symbol commonly used to heat flux in thermodynamics, which is not to be confused the use of the same symbol in conventional bond graphs to represent generalized displacement—consistent with the use of quasi-bond graphs. This is not only more familiar to most engineers and scientists, but also more likely corresponds to the use of linear constitutive relations with approximate constant parameters.

8.2.4 The RSh Element

A version of the RS element, called in THERMOSIM the RSh element, is also used to represent the temperature and entropy changes that occur when heat is conducted through a thermal “resistance.” This element is pictured in Fig. 8.6. Note that all the RS elements create entropy without changing the power. Note also that the causal strokes shown are mandatory, and therefore automatically applied by BondGrapher.

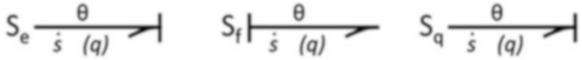


Fig. 8.7 Source elements for heat conduction

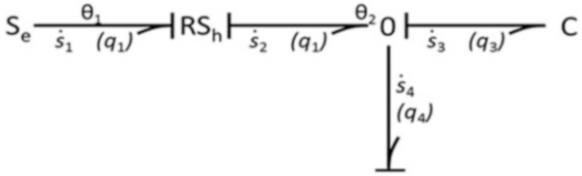


Fig. 8.8 A common combination of thermal conduction bonds and elements represented by the Ct macro element

8.2.5 The Heat Source Elements: Se, Sf, and Sq

The conventional source elements Se and Sf can be used for thermal sources, with the variables shown in Fig. 8.7. In THERMOSIM a variation of the thermal Sf source, called the Sq source, also shown, is sometimes employed to solve a very special problem.

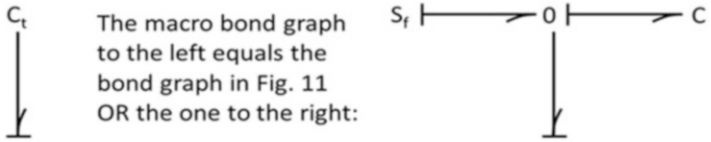
Its difference with Sf is merely the placement of the causal stroke, which in fact is a false placement; the heat flow is still the actual causal variable. The sole purpose for using this element is to allow a heat source element to be bonded to a CS element without the bond being confused with a pressure-velocity bond.

8.2.6 The Ct Element

A particular combination of heat conduction bonds and elements occurs so often that it is designated in THERMOSIM by a special shorthand element. This is the Ct macro element which represents heat being passed between the fluid in a CS element—or from an IRS element, described below—and the walls of a chamber or a fluid channel, and also heat being passed between these walls and the environment. In the first of two implementations, shown in Fig. 8.8, the environment is given a particular temperature, and the thermal resistance thereto is represented by an RSh element. In the second implementation, shown in Fig. 8.9, the environment comprises a heat source, such as from an electrical heater. In both cases the vertical bonds are connected to the CS or IRS element, and the thermal compliance of the walls is represented by the *C* element. Both combinations are represented by the Ct element shown at the left in Fig. 8.9.

8.2.7 The 0S Element

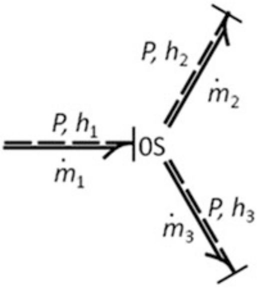
The 0S element shown in Fig. 8.10 represents a fluid junction, or pipe tee. In THERMOSIM the junction is restricted to three channels or convection bonds. This



The macro bond graph to the left equals the bond graph in Fig. 11 OR the one to the right:

Fig. 8.9 The second combination of element designated by the Ct element, and the Ct notation itself

Fig. 8.10 The OS junction for three joined fluid channels

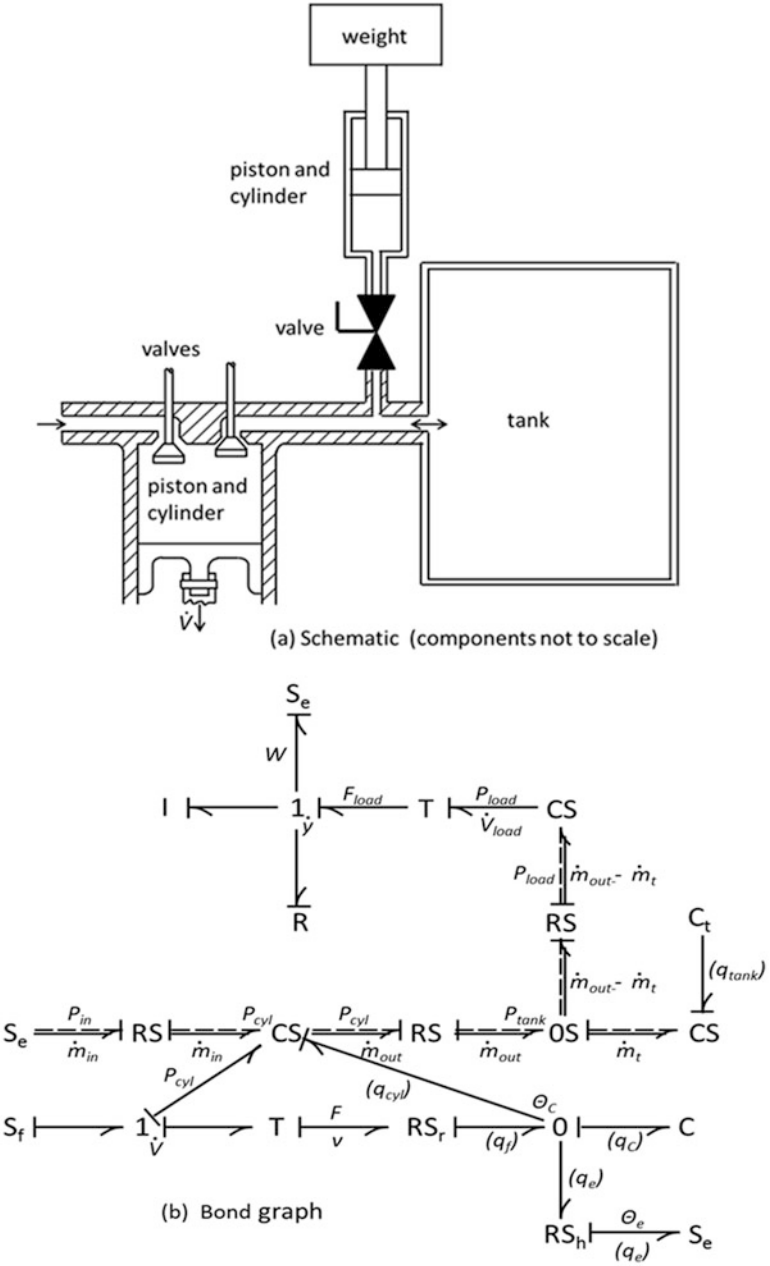


element is like a conventional 0-junction in that the total input mass flow equals the total output mass flow, and the three pressures are equal. If the flow is diverging, the effort variables from the single input channel are conveyed to the output channels, and the junction is thermodynamically reversible. If, rather, the flows from two channels are merging, the two input flows may have different temperatures, and irreversible mixing occurs, which means a loss in available energy (exergy). The flux of enthalpy which emerges is set equal to the sum of the two input enthalpy fluxes.

When there are more than one 0S-junctions in a system model, complex issues of their relative status require about 300 lines of code to sort out, but all the user needs to know is that some restrictions apply to allowed bond graph structures.

8.2.8 Example System

An example that illustrates the use of the various elements presented above is given in Fig. 8.11. This is a piston-cylinder air compressor with a tank and a mechanical load. The piston is driven from the mechanical source Sf, and the air enters from its atmospheric source Se. The cylinder is represented by the left-most CS element; the RS (RSc) elements on its two sides represent the input and output valves, which can be programmed to operate as a function of the position of the piston or in response to the various pressures so as to keep the flow passing from left to right. The flow bifurcates at the fluid tee represented by the 0S element, to pass to the large tank as given by the right-hand CS element, or toward the load as programmed by the user who sets the behavior of the load valve given by the right-most RS (RSc) element.



(a) Schematic (components not to scale)

(b) Bond graph

Fig. 8.11 Piston-cylinder air compressor with tank and load

The air in the load tank gets hot. Heat is conducted to the walls of the tank, where it is stored, and from there to the environment; these features are all represented in the macro Ct element. The output piston and cylinder is represented by the CS element toward the upper-right. The attached T element transforms the volume change rate to velocity. The weight of the load is represented by the uppermost Se element, its mass or inertia by the I element, and an effective dashpot by the R element. The transformer element T near the bottom of the bond graph transforms the rate of change of the piston volume to velocity, and the RSr element there uses this velocity, v, and the temperature of the 0-junction, θ_c, to compute the friction force on the piston, F, and the heat generated thereby, q_f. Some of this heat goes to the environment Se through the thermal resistance RSh, some goes to heat the walls of the cylinder—which gets quite hot—and some passes back into the air in the cylinder.

8.2.9 The 1- and 1S-Junction Elements with Convection Bonds

The only work interaction between convection and non-convection parts of a system presented thus far is through the volume change bond of the CS element. The 1S-junction and the generalized 1-junction perform the other work interactions.

The traditional 1-junction is a mainstay of non-convection bond graphs; it can have any number of bonds which are forced to share a common flow. In THERMOSIM it is restricted to three bonds; additional junctions are used when needed. Precisely two of the bonds are allowed to be convection bonds, leaving the third as a simple bond. Since the flow on convection bonds is always a mass flow, this means that all the flow on all three bonds of any such 1-junction must be the same mass flow. Further, *the junction is defined so as to represent thermodynamically reversible behavior*, so the entropies on its two convection bonds must equal one another. Since energy is also conserved, as with the traditional 1-junction, the effort on the simple or mechanical bond equals the difference between the enthalpies of the two convection bonds—assuming their power convention arrows are aligned the same way.

Sometimes one desires to have a common-flow junction with two convection bonds having equal pressures rather than equal entropies. This is the definition of 1S-junction, which thereby becomes thermodynamically irreversible. Use of the 1- and 1S-junctions is illustrated by the model for the heart of a positive-displacement compressor shown in Fig. 8.12.

Part (a) of the figure shows an ideal or lossless compressor (100 % volumetric and adiabatic efficiencies) with torque M and angular velocity $\dot{\phi}$. The modulus of the transformer T_m is the ratio of the mass flow to the angular velocity, which is the product of the density of the entering fluid times the volumetric displacement per radian or rotation. Part (b) shows how to represent compressors with less than 100 % volumetric and adiabatic efficiencies. The modulus of the transformer T can

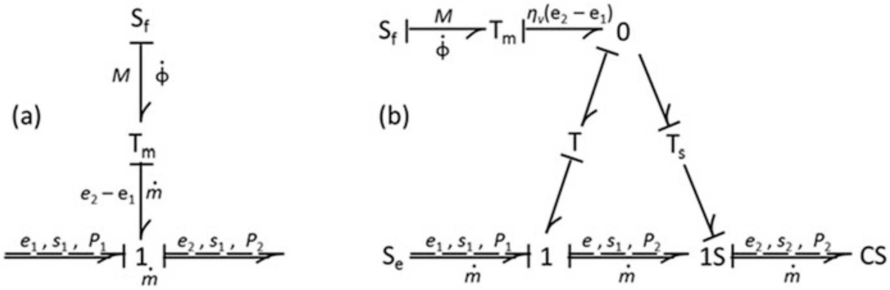


Fig. 8.12 Bond graphs (**a**) and (**b**) for a positive-displacement compressor or pump

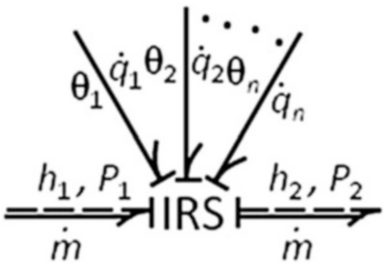


Fig. 8.13 The IRS element

be shown to be $T = \eta_v / \eta_{\text{ad}}$, where η_v is the volumetric efficiency and η_{ad} is the adiabatic efficiency. The modulus of the transformer T_s can be shown to be $T_s = \eta_v / (1 - \eta_{\text{ad}})$.

Note that elements in THERMOSIM containing the letter *S* in their names are defined to be at least potentially irreversible.

8.2.10 The IRS Element

The final element with convection ports is the IRS element, first introduced in 2003 [4] with a major update in 2016 [10]. This represents a fluid channel with fluid friction, inertia and heat transfer to its surroundings—normally the walls. Such channels are the principal way engineers achieve critical heat transfer in the systems they design; the most common are round tubes. There are two versions of this element in THERMOSIM, one for nominal liquids such as water or a hydraulic fluid, and a second for general compressible substances. Both are represented by the bond graph notation shown in Fig. 8.13, and employ the momentum of the stream as its single state flow variable. The multiple heat conduction bonds shown are for the general case. Often, however, none or only one such bond is used.

The principal parameters are the length of the channel and its area, which is assumed to be uniform. In keeping with the philosophy that lumped bond graph elements should use approximate constitutive relations in order to achieve a reasonable compromise between accuracy and computational efficiency, the wall friction models assume an equivalent steady or equilibrium velocity profile across the channel. Further, a round cross-section—that is a tube—or its equivalent using the concept of a hydraulic diameter is assumed. Implementing this requires computation of the effective Reynolds number of the flow—or for two-phase flow, separate Reynolds numbers for the liquid and vapor phases. This is turn requires knowledge of the viscosity or viscosities of the fluid. For purposes of relating mass flow to the momentum, a slug-flow approximation is employed, so that the mass flow is taken as the ratio of the momentum to the length. The rate of change of the momentum is proportional to the pressure difference between the two ends of the element minus the pressure drop due to wall shear, with a correction for the difference between the pressures associated with the momentum fluxes at the two ends.

The compliance of the fluid in the channel is not included within the IRS element, but rather is modeled by CS elements that may be placed at its ends. This means that when compressibility effects are especially important—often they are not—such as when wave propagation effects are of interest, multiple IRS elements interspersed with CS elements should be used. Normally, the compliance of the fluid in the tube is split between the two adjacent CS elements by assigning half of its volume to each.

The simple bonds in Fig. 8.13 are for the heat conduction to the surroundings, as with the CS element. It may be attached to a Ct macro element or a more primitive combination of heat conduction bonds and elements. Many scientists and engineers devote their careers to modeling such heat transfer. The author [10] has selected what he hopes are the most useful but reasonably simple set of heat-transfer correlations available in the literature, but the user may prefer to substitute others.

The IRS model for liquid media is simpler than that for general compressible substances, and is vastly simpler when there is no heat conduction bond or only one. The hydraulic system pictured schematically in Fig. 8.14 with the corresponding bond graph in Fig. 8.15 illustrates its use when there is one heat conduction bond. BondGrapher, which is introduced in Sect. 8.4, created this figure in which the selected element CS22 is highlighted with color.

Note the incorporation of the model in Fig. 8.12 for the pump, allowing for adiabatic and volumetric efficiencies less than 100 %. The two fluid tees are represented by the two 0S elements. The elements CS18 and CS36 represent the hydraulic sides of the low-pressure and high-pressure accumulators, respectively, while CS22 and CS31 represent the nitrogen sides. Se23 and Se33 represent the temperature of the environment with which the accumulators interact, while R21 and R27 represent resistances associated with the motion of the accumulator interfaces, which are necessary for computational causality purposes although they are too small to have a significant effect. CS6 and CS9 represent small fixed volumes along the fluid lines to accommodate the small effects of compressibility and to satisfy computational causality. The friction and inertia effects of the fluid lines are represented by IRS7, IRS10, and IRS16; the latter two are quite long—4 m—and also exchange heat with their surroundings through the use of elements Ct34

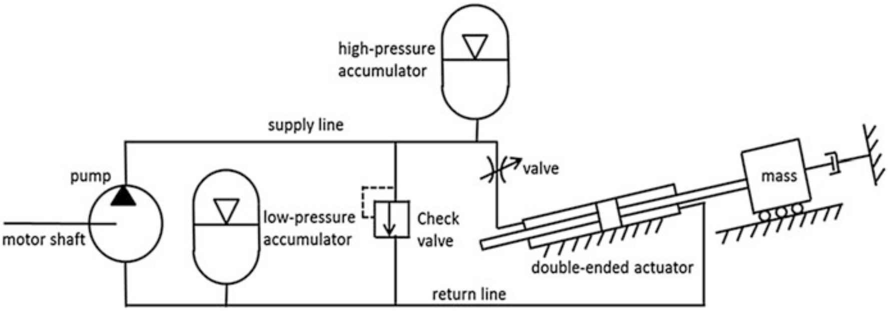
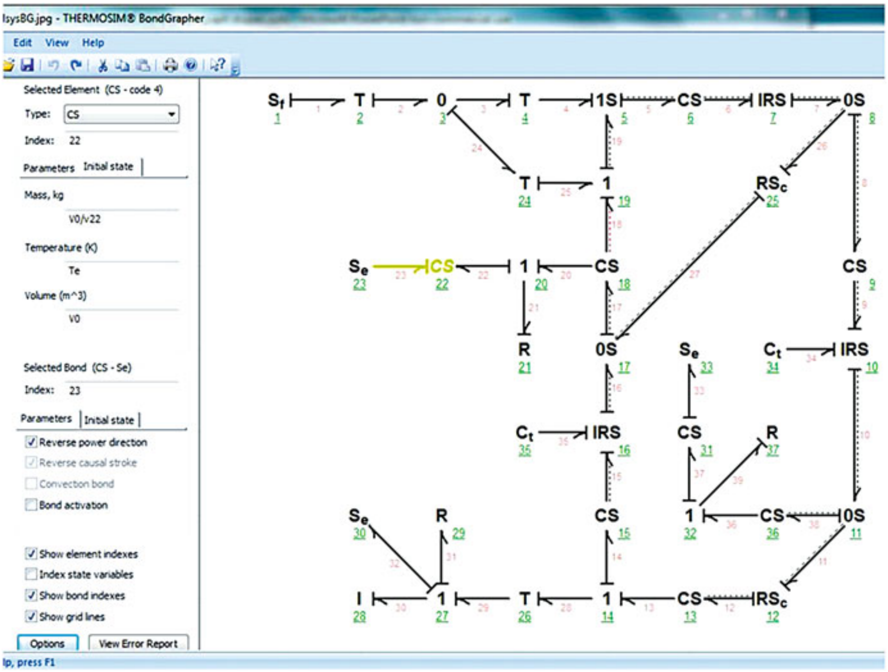


Fig. 8.14 Schematic for the hydraulic system (using fluid power symbols)


IsysBG.jpg - THERMOSIM® BondGrapher

Selected Element (CS - code 4)

Type: CS

Index: 22

Parameters | Initial state

Mass, kg: V0/v22

Temperature (K): Te

Volume (m^3): V0

Selected Bond (CS - Se)

Index: 23

Parameters | Initial state

☑ Reverse power direction
☑ Reverse causal stroke
☐ Convection bond
☐ Bond activation

☑ Show element indexes
☐ Index state variables
☑ Show bond indexes
☑ Show grid lines

Fig. 8.15 Bond graph for the hydraulic system

and Ct35. RS25 models a spring-loaded check valve that partially opens when the upstream, pressure exceeds 20 MPa, preventing overpressure. RS12 is a valve that opens when the user wishes to move the double-sided piston-cylinder actuator, which operates with a hydraulic fluid. Further discussion including simulation results is given in Sect. 8.4.3.

For long channels the properties of the fluid and the wall temperatures may vary considerably from one end to the other. This is particularly true when the fluid is a highly compressible substance, but also applies to pure liquids with significant

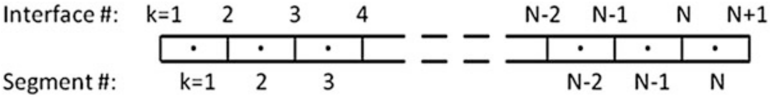


Fig. 8.16 Segmentation of the IRS element

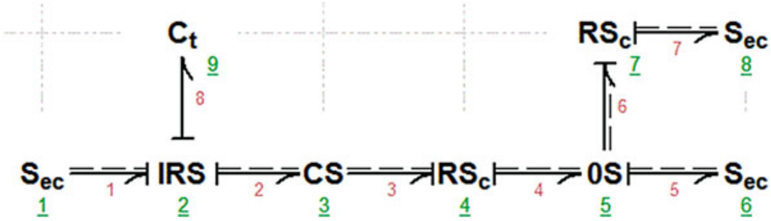


Fig. 8.17 Bond graph for the hot water shower system

transport time from one end to the other and significant thermal interaction with the channel walls. The author models such situations by segmenting the channel, as shown in Fig. 8.16. Each such segment has one of the heat conduction bonds shown in Fig. 8.13 which is assumed to be attached to a Ct element. Each Ct element also has its thermal compliance and state variable for the average temperature of the wall at that location. The users draws only one Ct element with the BondGrapher interface, however; the rest are inferred by a designation of the number of the segments *N(I)*, where *I* is the index number of the IRS element, avoiding what otherwise could be a very messy bond graph. The user assigns the overall thermal compliance of the tube to this element, represented here as *C*; the program automatically divides *C* by *N* to give the compliance of the individual segments.

Along with the *N* state variables for the wall temperatures of the segments there are an additional *N* state variables for the average fluid temperatures within the segments. Thus the IRS element has a total of $2N + 1$ state variables, the 1 being for the momentum or mass flow rate. The wall shear is also computed separately for each segment, but this requires no further state variables. Temperatures of the wall and the fluid are computed for both the segments and at the interfaces.

A person wants to take a quick shower; how long does he have to wait for the hot water emerging from a hot water heater 10 m away to reach him, at location Sec6, and of necessity also warm the walls of the copper pipe? A bond graph model for this system is shown in Fig. 8.17. The hot water emerges from the element Sec1. Cold water also emerges from Sec8 to mix with the hot water at pipe tee 0S5. The valves from the two sources, RSc4 and RSc7, are abruptly opened at time zero. The element IRS2 is the long pipe modeled with 20 segments, and CS3 is a small volume representing the mixing of new water with old and the slight compressibility of the water in the pipe, and satisfying causal requirements. This example is continued, with simulation results, in Sect. 8.4.4.

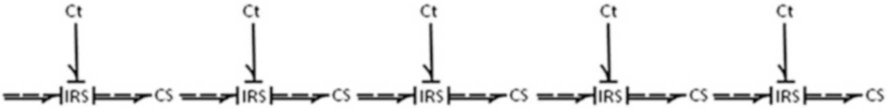


Fig. 8.18 A model for a fluid channel containing a compressible fluid using unsegmented IRS elements

When a channel contains a compressible fluid, the state of that fluid can change radically along its length, potentially being a liquid at one end, a saturated mixture in the middle, and a pure vapor at the other end. A proper lumped model of this channel contains a long cascade of alternating lumped IRS and CS elements, part of which is shown in Fig. 8.18; the CS elements represent the fluid compliance and the IRS elements represent the wall friction and inertia and thermal interaction with the walls of the channel via the attached single Ct elements. An especially difficult situation arises when the segment of fluid line represented by the element is partly a compressed liquid and partly a saturated mixture, since the properties of the fluid in these two phases are radically different. The length of the cascade would have to be exceedingly long to accommodate this occurrence, imposing a huge computational burden. A radical structural simplification is proposed to reduce this complexity without overly compromising accuracy.

Experience has shown that the mass flow rates approach equilibrium vastly quicker than the temperatures of the fluid and the channel wall. This is associated with the fact that the effect of the mechanical compliance—the compressibility of the fluid—is effectively over an order of magnitude smaller than the effect of the thermal compliance—heat capacity—of the walls. This suggests approximating a channel with at least a modest span of fluid states by a single IRS segmented element with its uniform mass flow rate, allowing the heat transfer to the walls and the state of the fluid to vary along its length. The resulting number of IRS elements needed to represent a channel adequately should be reduced radically, sometimes to only one.

The first-generation IRS element required the user to specify the thermal conductance between the wall and the environment, and was implemented only for refrigerant R12. Further, the temperature of the tube wall was not allowed to vary quite properly along its length in response to varying heat-transfer rates. A final difficult problem was the danger that the phase of the fluid would change in the middle of a segment to or from liquid or to or from vapor—the other state being a saturated mixture—producing gross errors in key properties that determine the heat transfer for that segment. To combat this frequent occurrence, the transition was anticipated and the segment sizes adjusted so that the transition would occur very near the end of a segment. This caused great complication and on some occasions even caused the program to abort.

The new segmentation also has a key advantage of allowing efficient treatment of the problem of sudden changes in phase in the middle of a segment. It also allows a sudden change from boiling to condensation to be treated efficiently.

The calculations for the IRS element are coded in the 1000-plus-line script file *IRS4.m*, which also calls the function file *viscosity.m* for the thermal conductivities as well as the viscosity, and the function file *boiling.m*. The file *IRS4.m* is called by the main simulation program for potentially multiphase fluids, *convec2.m*. The differential equation for the state (or temperature) for the wall segment k is

$$\frac{dx_k}{dt} = \left(T_e - T_{\text{seg}}\,(I,k)\right) * H\,(I,k) - Q_{Is}\,(I,k) \tag{8.4}$$

in which $H(I,k)$ is the thermal conductance of the "thermal resistance" of the Ct element for the kth segment. This conductance is based on the properties of the environmental—or "alternative" fluid—and certain properties. It is computed in a new relatively short function file *CtH.m* also called by *IRS4.m*, which can also be called to compute the heat transfer associated with CS elements.

The differential equation for the temperature of the fluid in a segment is

$$\frac{dx_{N+k}}{dt} = ((T_e - x_{k+N}\,(I,k))\,\dot{m} + Q_{I2}\,(I,k)\,/dh/dt)/(m/N) \tag{8.5}$$

in which I is the index of the IRS element, k is the index of the segment, Q_{I2} is the heat flux from the wall to the fluid for the segment, *dh/dt* is the specific heat of the fluid, and *m/N* is the mass of the fluid in one segment. This version of the equation has been specialized for pure liquids to speed up execution with negligible error.

CS elements are nominally considered to represent physical volumes with large enough dimensions that the interior velocities of the fluid are small enough not to produce dynamic heads of significance. For example, a CS element bonded to an 0S-junction is assumed to give that junction a static pressure common to all of its bonds. The hydraulic system above is an example. IRS elements, by contrast, nominally have such a small cross-sectional area that the dynamic head is indeed significant. What does it mean, then, when a CS element is bonded to an IRS element? For the nominal case, including the hydraulic system, it means that the interior area of the CS element through which fluid is passing is presumed to be significantly larger than that of the attached IRS element. Clearly, however, that is not the case for the cascade of IRS and CS elements depicted in Fig. 8.18. In effect, a different kind of CS element is needed there, which produces a total dynamic pressure on the joining bond rather than a static pressure. This is called a "total-pressure CS element," for which the dynamic head on the connecting bond is simply added to the static pressure to give the proper total pressure. It is necessary for the user to designate that this is what he wants, however; BondGrapher has been modified to ask the modeler's choice, with the default being the "static pressure CS element."

8.3 Thermodynamic and Physical Properties of Common Substances

Starting in 1999 [3] and continuing until the present [3, 5, 8, 9] the author has developed practical code for modeling the properties of gaseous, liquid, and saturated mixtures thereof of 30 common pure substances that are identified as potentially being in the vapor state or saturated mixture state—ammonia, argon, butane, carbon dioxide, cesium, ethane, ethylene, helium, heptane, hexane, hydrogen, isobutene, isopentane, mercury, methane, neon, nitrogen, octane, oxygen, propane, propyl alcohol, propylene, the refrigerants 12, 13, 134a, 22, 23, 502, and 503, rubidium, and water. Prominent amongst his sources is the work of Reynolds [12]. Gaseous air is treated as a mixture of nitrogen, oxygen, and argon. He has also focused on four additional substances identified as liquids—liquid water and the hydraulic fluids MIL H 5606, MIL PRF 27601, and MIL PRF 83282.

In the spirit of a compromise between accuracy and complexity he has avoided use of the most accurate models available for some substances, particularly for water where lifetimes of effort have produced very accurate but extremely complex models that would drastically retard the speed of simulation, favoring instead a model that was once considered generally excellent. He has also forgone some available accuracy in the region of the critical point; should this especially complex region become critical, making an improvement ought to be considered.

Any pair of independent properties are necessary and sufficient to specify the state of a pure substance. For the pure liquids, the most practical pair is temperature and pressure. For the general substances, as is pointed out in Sect. 2.1, the pair that is effectively mandated by the available data is temperature and specific volume—the inverse of mass density. This use of specific volume can be problematical in the compressed liquid state, however, since liquids have nearly constant specific volume. Depending on the substance, approximate values may not be computed unless one specifies a value for the bulk modulus, which is an option.

Analytic models taken from the literature are used to compute the dependent thermodynamic properties, which include enthalpy, internal energy, specific heat, entropy, the quality of saturated mixtures—the author extends the definition of quality to make it useful also for the compressed liquid and vapor states—specific volume, enthalpy and entropy of a saturated liquid at the given temperature, and these same properties for a saturated vapor at the given temperature. The list is of course expanded by absolute pressure for the general substances and mass density for the pure liquids. Further, simulation programs used by THERMOSIM employ several Runge–Kutta iterations that are expedited if the gradients of the non-saturation dependent properties with respect to the two independent properties are also available in analytic form. The author therefore derived analytical derivatives of the available empirical relations for pressure, internal energy, enthalpy and entropy, plus the gradient of quality with respect to temperature, adding nine more dependent variables to the list for the general substance. When the phase changes between pure liquid and saturated vapor, these gradients change abruptly by orders of magnitude, requiring special treatment. Abrupt changes in heat-transfer and wall friction domains also are cause for special smoothing.

Separate simulation programs are employed for systems in which the principal fluid is a liquid or is a general substance.

The transport properties of viscosity and thermal conductivity are also needed if one is to compute heat transfer and fluid friction. Critically important dimensionless groups including Reynolds and Nusselt numbers depend on these. Analytical expressions for viscosity and thermal conductivity as functions of temperature and specific volume are difficult to find or deduce, unfortunately, so at this writing the author has only limited results. He managed to convert into the needed form some raw data for the four liquids listed above. For general substances he has results for water, nitrogen, oxygen, argon, air as a vapor, and an approximate form for the refrigerant R12. It is not easy to find appropriate correlations in the literature; much more work remains. See also Sect. 8.5 below.

8.4 The Graphical Interface BondGrapher and Examples

Each system model is coded in a MATLAB® system file that, when called by the user, directs the desired dynamic simulation. In practice, this system file is most conveniently produced with the help of a graphical interface called BondGrapher®, which was written by the author's son, Scott A. Brown. All the files needed to carry out the simulation except for BondGrapher, assuming one has access to MATLAB, are freely downloadable from the author's website [9]. The prospective user should also download the extensive Users' Manual (make sure it is Version 4), which provides the details and many examples of the procedure. BondGrapher includes help files, but only the ones that pertain to drawing bond graphs are useful as of this writing. To acquire BondGrapher, which also is free for the foreseeable future, one should email the author and include a commitment not to transmit the program to a third party without permission.

The structure of the computational part of the THERMOSIM package is suggested in Fig. 8.19.

8.4.1 BondGrapher and the Steam Catapult System

BondGrapher has a main screen and a screen called *Global Options*. The main screen with the bond graph for the aircraft catapult of Fig. 8.3 drawn is shown in Fig. 8.20. The graph may have been entered directly, or downloaded from an existing system file created and saved earlier using a name of the user's choice. The procedure to draw a new bond graph involves placing symbols for elements over node locations, and placing bonds between nodes by clicking on the element at one end of the bond and then double-clicking the node at the other end. The

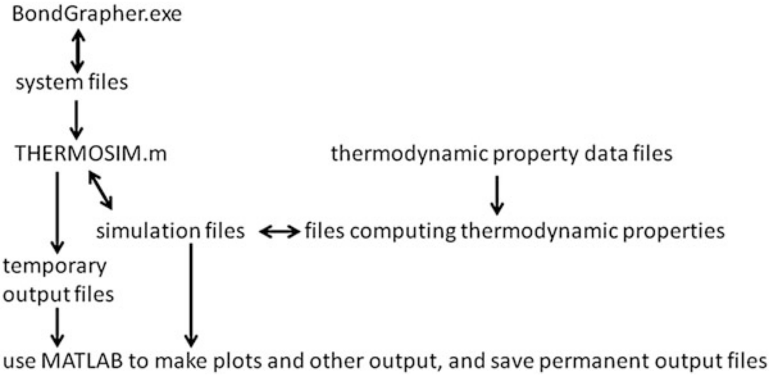
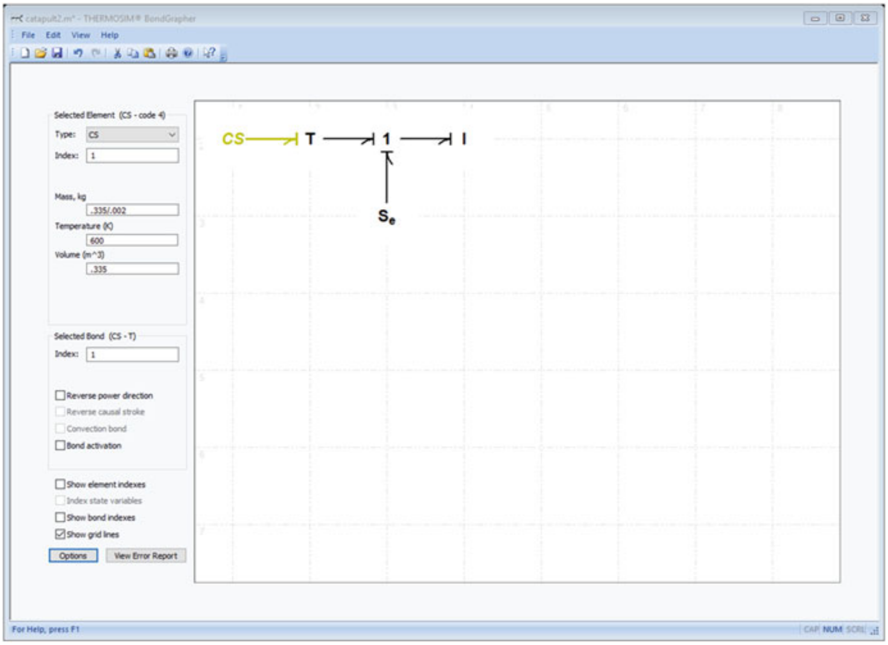

BondGrapher.exe
system files
THERMOSIM.m
thermodynamic property data files
simulation files
files computing thermodynamic properties
temporary output files
use MATLAB to make plots and other output, and save permanent output files

Fig. 8.19 Structure of the THERMOSIM package



Selected Element (CS - code 4)

Type: CS

Index: 1

Mass, kg: .335/.002

Temperature (K): 600

Volume (m^3): .335

Selected Bond (CS - T)

Index: 1

☐ Reverse power direction
☐ Reverse causal stroke
☐ Convection bond
☐ Bond activation

☐ Show element indexes
☐ Index state variables
☐ Show bond indexes
☑ Show grid lines

Options | View Error Report

Fig. 8.20 The main screen of BondGrapher showing the aircraft catapult of Fig. 8.3

symbols (Fig. 8.20). The main screen of BondGrapher showing the aircraft catapult of Fig. 8.3 may be chosen by typing their first letter, and hitting the space bar if more than one symbol begins with that letter. Alternatively, one can check the desired element on a list that is available on the left side of the screen. The program knows from the choice of elements whether a bond should be a simple one or a convection one. The program almost always knows where to place the causal strokes. The power-convention half arrow is placed according to the order of the chosen elements.

The user can make changes by clicking where indicated at the left side of the screen. Also, bonds can be activated with another click. With a little practice this procedure is very fast. It is suggested that one first sketch the desired graph on a sheet of paper, to make sure it does not have confusing crossing bonds and otherwise is easy to read. One can always move the elements around with a simple click-and-drag operation. The elements and bonds are numbered, separately, in the order they have been selected; the numbers can be changed. The bond graph image can be moved by clicking and dragging on any location not too near a node. Also, the size of the graph may be changed; the author's preferred method is to use the mouse wheel. The Instructors' Manual gives more detail and many examples. One may print out the bond graph or the entire screen at any time.

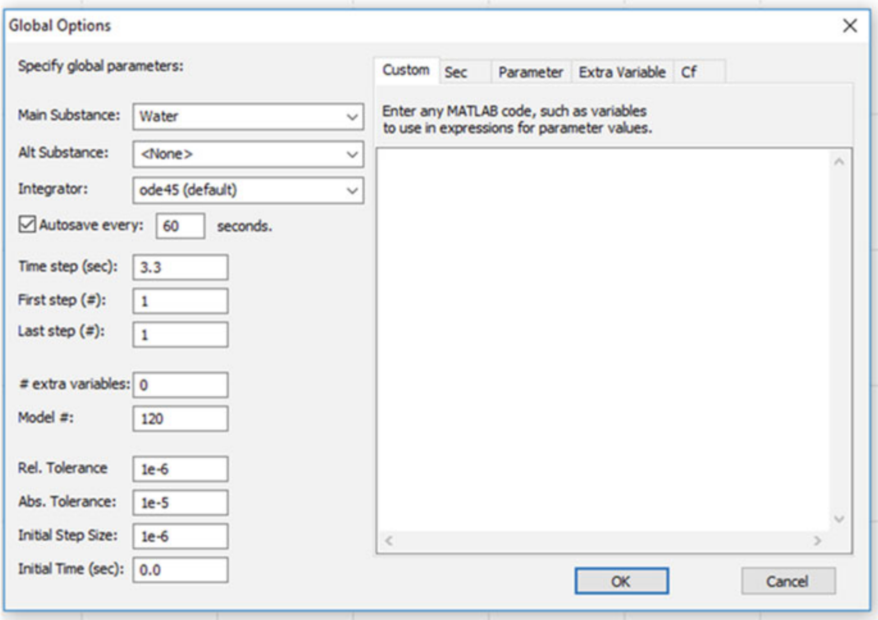
Constant parameter values for elements may be inserted by clicking on the element and then clicking on the *Parameter* tab on the left side of the screen. The CS element with this tab is chosen in the example. If the element requires parameters, one or two corresponding boxes appear into which one may place either direct values or symbols for the indicated parameters, to be given numerical values later. One similarly indicates the values of any initial conditions by clicking of the *initial state* tab. When no value is entered, the default value of zero is assumed. The constitutive relations entered this way for R, C, I, and RS_r elements may be either linear or satisfy the power-law relation

$$\text{Output variable} = \text{parameter} \cdot \text{sign}\,(\text{input variable}) \cdot \text{abs}(\text{input variable})^{\text{exponent}}$$

An exponent set to one directs a linear relation. The R and RS_r elements give coulomb friction when the exponent is set equal to 0. Elements other than these three allow only linear relations unless information is inserted elsewhere, as described below.

The macro element Ct and the element IRS are special cases in which three parameters must be chosen. The third parameter is entered where a second initial condition for the bond would normally be placed. There are two versions of the Ct element, as described above. The version with the Se reservoir is chosen by placing the heat conduction coefficient of its RSh element, with units of W/K, in the first parameter position. The version with the Sf reservoir is chosen by placing 0 there. In both cases the second position is used for the thermal compliance, in J/K. The third parameter is the temperature of the Se, in K, or the heat rate of the Sf, in W. The first initial state is the actual heat in the compliance, which is the product of its compliance and its temperature, with units of Joules. The third parameter for the IRS element is the number of segments, N, into which the channel is divided.

The simulation program setup by BondGrapher saves the values for the temperature and the specific volume—or density for pure liquids—for any bonds with fluid flow, after each time step. These vectors, called Tc and vc, help start iterations for the subsequent time step. Trial values for these are needed for the very first step, but usually any values not orders of magnitude incorrect will work. Therefore, default



Global Options

Specify global parameters:

Main Substance: Water

Alt Substance: <None>

Integrator: ode45 (default)

☑ Autosave every: 60 seconds.

Time step (sec): 3.3

First step (#): 1

Last step (#): 1

extra variables: 0

Model #: 120

Rel. Tolerance 1e-6

Abs. Tolerance: 1e-5

Initial Step Size: 1e-6

Initial Time (sec): 0.0

Enter any MATLAB code, such as variables to use in expressions for parameter values.

Fig. 8.21 The Global Options page for the example of the airplane steam catapult

values are given. Should these fail, one can change them by clicking on one of the relevant bonds, and click on *initial state* in the *selected bond* group on the left side of the screen. The user is asked if he would like to use that value for all of the relevant bonds; “yes” is usually the best answer.

At any point one may click of the *view error report* box located toward the lower left of the screen. This report points out any violations of the rules or essential omissions you need to correct before running a simulation. It will also observe issues over which only the user knows the correctness of, such as whether parameters or initial conditions still set at zero are what is intended.

The *Global Options* screen is displayed by clicking on the *Options* button toward the lower-right corner of the main screen. The example for the aircraft catapult is shown in Fig. 8.21. If the system employs one or two fluids, the choice is made on the left side of the screen: the catapult uses water (in the wet steam region). In general, a second “alternate” fluid is also allowed for regions that cannot mix physically with the “main” substance, such as for the gas in an accumulator used in a hydraulic system, or for the thermal reservoir, for example, included within a Ct macro element that is attached to a CS element or an IRS element. One also chooses the integrator to be employed by MATLAB; the integrator ode45 is most commonly chosen for non-thermodynamic systems, and ode23 often appears to be more robust for thermodynamic systems.

Another choice selected on the left side of the *Global Options* screen is how long each time segment in a run should be, and how many time segments are to be run. If the numbers of the "first step" and the "last step" are both 1, as with the catapult, there is only one time segment and the "time step" is its duration in seconds. The simulation stores the history of the states through the most recent time step in a file *TSIMdata.mat*, which the MATLAB command *load TSIMdata* allows the user to access. If the "first step" is chosen to be 1 and the "last step" is a larger than the "first step," the states at the ends of each time segment are stored in a file *dataX.mat*, which can be accessed by the command *load dataX*. This provides a data compression feature that may be attractive if a run would otherwise produce data files that are longer than one would prefer. In practice, however, the most useful feature of the data compression mode is the ability to interrupt a simulation at any point, look at the results up to that point, and then resume the simulation. This is also helpful if the simulation is very slow, or if one wishes to change one or more parameters in the middle of an overall run. Following an interruption in a simulation and commanding *load dataX*, one can determine the number of the last time segment completed with the command *size(time)* or *size(state)*. To resume the simulation, one enters this number as the "first step" of the continuation.

Finally, the bottom of the left side of the Global Options page allows the user to choose the relative and absolute tolerance of the simulation, and the initial step size. See MATLAB *help* for detailed information.

The Steam Catapult system uses only one of the six tabs on the right side of the Global Options page, that called *Custom*. This tab is used to provide the values of any constants indicated elsewhere only by symbols. The only such symbol for this system is the temperaureT00, used in the definition of Tc. Other uses of the right side of the Global Options page are discussed in the context of other systems below.

After running the system file, you may enter MATLAB commands to plot, print, or save whatever results you wish. Selected results from the simulation of the steam catapult system are shown in Fig. 8.22.

8.4.2 The Piston-Cylinder air Compressor and Programming Nonlinearities and Special Functions

The main page of the BondGrapher that produced the MATLAB system file *aircompsys2.m* is shown in Fig. 8.23, with the element *C* and the parameter tab clicked. On the screen the element is highlighted in color. The exponent of the constitutive relation for this compliance is given as 1, indicating a linear relation, of several either directly given or computed in the *Custom tab* on the right side of the Global Options page, as shown in Fig. 8.24, using standard MATLAB coding.

There are four more tabs to the right of the Custom tab. The most important of these are the *Parameter* tab and the *Extra variable* tab, which are discussed presently. First, however, it is necessary to know something about a matrix called *el(i, j)*.

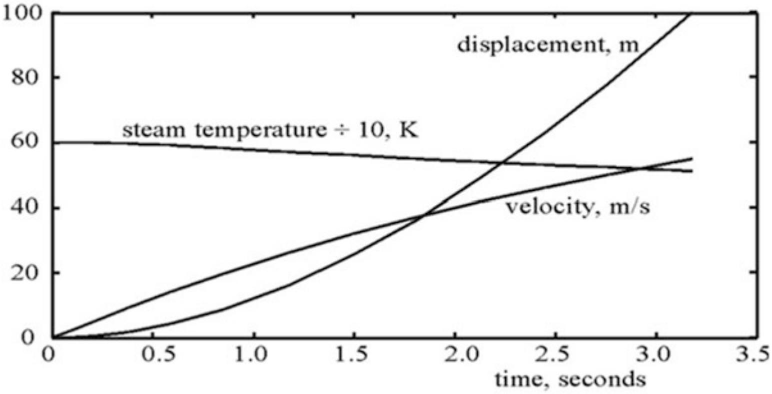
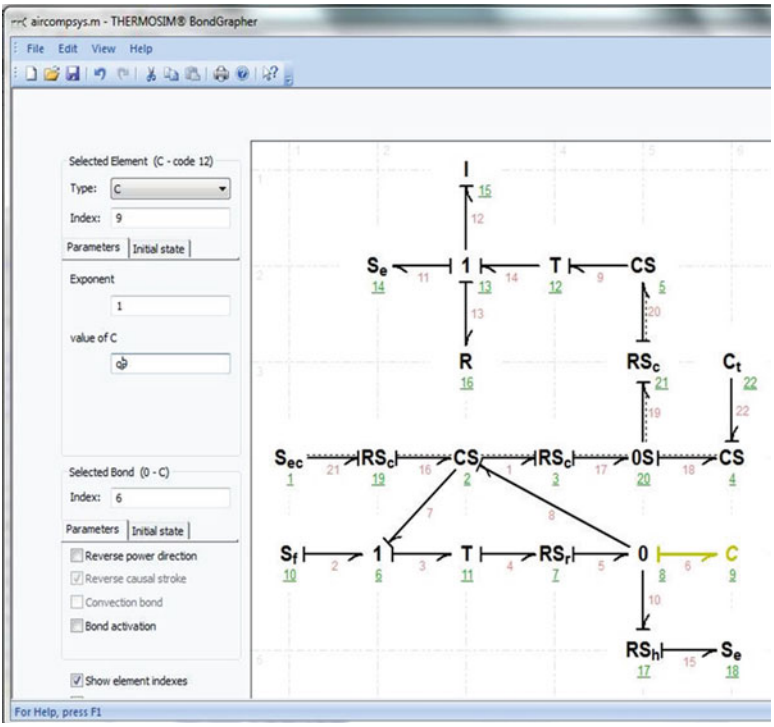


Fig. 8.22 Selected results of the steam catapult system



Selected Element (C - code 12)

Type: C

Index: 9

Exponent

1

value of C

Selected Bond (0 - C)

Index: 6

□ Reverse power direction

☑ Reverse causal stroke

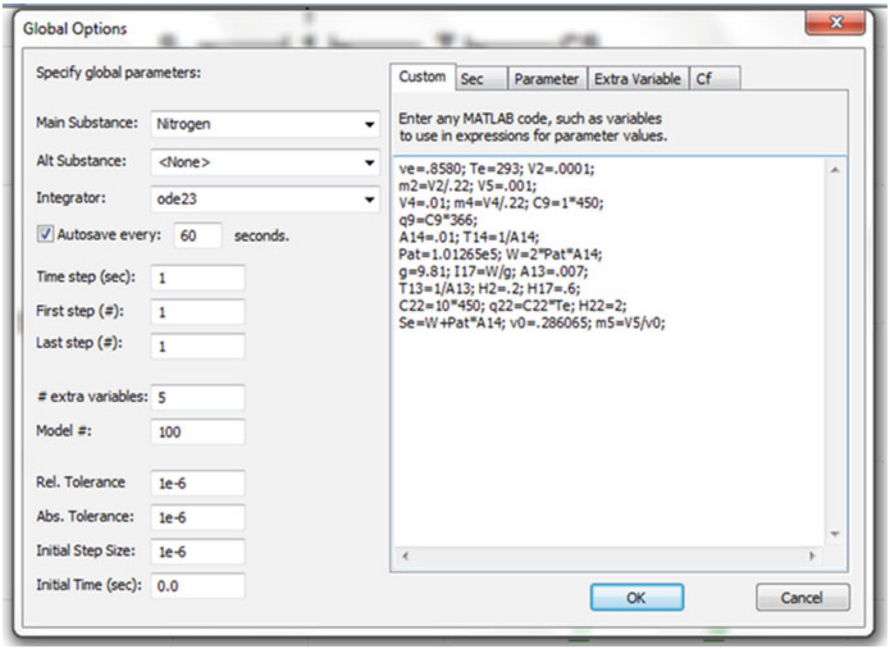
□ Convection bond

□ Bond activation

☑ Show element indexes

For Help, press F1

Fig. 8.23 Main page of BondGrapher for the piston-cylinder air compressor



Specify global parameters:

Main Substance: Nitrogen

Alt Substance: <None>

Integrator: ode23

☑ Autosave every: 60 seconds.

Time step (sec): 1

First step (#): 1

Last step (#): 1

extra variables: 5

Model #: 100

Rel. Tolerance 1e-6

Abs. Tolerance: 1e-6

Initial Step Size: 1e-6

Initial Time (sec): 0.0

Enter any MATLAB code, such as variables to use in expressions for parameter values.

```
ve=.8580; Te=293; V2=.0001;
m2=V2/.22; V5=.001;
V4=.01; m4=V4/.22; C9=1*450;
q9=C9*366;
A14=.01; T14=1/A14;
Pat=1.01265e5; W=2*Pat*A14;
g=9.81; I17=W/g; A13=.007;
T13=1/A13; H2=.2; H17=.6;
C22=10*450; q22=C22*Te; H22=2;
Se=W+Pat*A14; v0=.286065; m5=V5/v0;
```

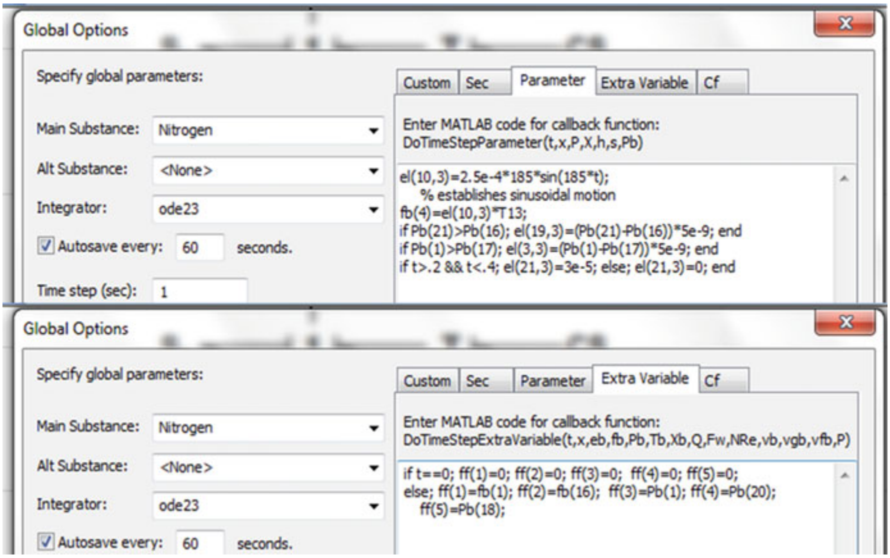
Fig. 8.24 Global Options page of BondGrapher for the air compressor system

The matrix *el* is the repository of the order and content of all the elements in a system model. It is a prominent part of the system file created by BondGrapher. It has one row for each element, in the order of their element numbers. Each row has three members. The first member is a code number designating which of the 20 types the element is. The other two members are both zero for the four junctions. Their content for the remaining 16 element types is given in a table in Sect. 1.6 of the Users' Manual (version 4) which is too large to reproduce here.

In the *Parameter* tab the parameters are defined by the user in one of the following forms, in which *X* refers to the quality of the fluid and *Pb* refers to the pressure on bonds:

$$
\begin{aligned}
el\,(i,2) &= \text{function}\,(t,x,P,X,h,s,Pb)\ \text{(for multiphase substancess)}\\
&\quad \text{or } el\,(i,2) = \text{function}\,(t,x,Pb,\ h,s)\ \text{(for liquids)}\\
el\,(i,3) &= \text{function}\,(t,x,P,X,h,s,Pb)\ \text{(for multiphase substances)}\\
&\quad \text{or } el\,(i,3) = \text{function}\,(t,x,Pb,h,s)\ \text{(for liquids)}
\end{aligned}
$$

Here *i* is the number of the bond and the 2 or 3 refers to the parameter as identified in the table. Look at the *Parameter* tab for the air compressor system given in Fig. 8.25. Its first line specifies the sinusoidally varying angular velocity of the input shaft; the table says that the value of the flow for Sf elements is given in the third position,



Specify global parameters:

Main Substance: Nitrogen

Alt Substance: <None>

Integrator: ode23

☑ Autosave every: 60 seconds.

Time step (sec): 1

Enter MATLAB code for callback function:
DoTimeStepParameter(t,x,P,X,h,s,Pb)

```
el(10,3)=2.5e-4*185*sin(185*t);
    % establishes sinusoidal motion
fb(4)=el(10,3)*T13;
if Pb(21)>Pb(16); el(19,3)=(Pb(21)-Pb(16))*5e-9; end
if Pb(1)>Pb(17); el(3,3)=(Pb(1)-Pb(17))*5e-9; end
if t>.2 && t<.4; el(21,3)=3e-5; else; el(21,3)=0; end
```

Specify global parameters:

Main Substance: Nitrogen

Alt Substance: <None>

Integrator: ode23

☑ Autosave every: 60 seconds.

Enter MATLAB code for callback function:
DoTimeStepExtraVariable(t,x,eb,fb,Pb,Tb,Xb,Q,Fw,NRe,vb,vgb,vfb,P)

```
if t==0; ff(1)=0; ff(2)=0; ff(3)=0;  ff(4)=0; ff(5)=0;
else; ff(1)=fb(1); ff(2)=fb(16);  ff(3)=Pb(1); ff(4)=Pb(20);
   ff(5)=Pb(18);
```

Fig. 8.25 The *Parameter* and *Extra Variable* tabs for the air compressor system

explaining the number 3. Its fourth and fifth lines specify how the openings and closings of the inlet and outlet spring-loaded check valves for the cylinder depend on the pressures on either side of the valves; the table says that the area of RSc elements is also given in the third position. The last line specifies the times at which the valve to the load (another RSc element) is opened and closed (to achieve a desired motion of the load). The third line satisfies a requirement for certain cases of RSr elements dictated by a singular peculiarity of THERMOSIM; the causal input velocity to the element is specified explicitly. This is required when the velocity is not a direct function of a state variable but rather comes from a transformer or a gyrator, since the relations for T and G elements are not invoked before the input velocity is needed. The number 10 there is the number of the Sf element in the bond graph.

"Extra variables" are most commonly used to allow non-state variables to be drawn or otherwise displayed or recorded. For example, one may wish to plot a *pressure* on a *bond* that is not available as a state variable of some CS element. A second use is to compute the integral of a variable that itself is not already a state variable, for example, the *displacement* associated with an inertance that is not already a state variable because no associated compliance element is assigned, or the total mass that passes through a convection bond.

The *time derivatives* of these variables are coded by the user in the *Extra variable* tab in the form

$$ff(i) = \text{function of } t,\ x,\ \text{eb},\ \text{fb},\ \text{Pb},\ \text{Tb},\ \text{Xb},\ Q,\ \text{PW},\ \text{NRe},\ \text{vb},\ \text{fgb}, \text{vfb},\ P,\ X$$

where i ranges from 1 to the total number of such variables. The arguments pressure, Pb, quality, Xb, effort, eb, flow fb, specific volumes, vb, vgb, and vfb, and temperature, Tb, refer to bonds, and use their index numbers. The heat flux Q is associated with IRS and CS elements, the friction pressure drop PW and Reynolds number NRe are associated with the IRS elements, and the pressure P and quality X refer to the CS elements. (When the fluid is air, Xb refers to the water component only.) The needed indices are given on the bond graph drawn by BondGrapher, assuming the proper boxes are checked. Indices are also needed for the components of any elements of the state vector, $\boldsymbol{x}$, which may be used.

In addition to entering the actual code for the extra variables, the *number* of them must be entered into the indicated box on the left of the *Options* screen.

For the air compressor system it was desired to plot five variables as functions of time that cannot be directly represented as functions of the state extra variables: the flows on the bonds 16 and 1, and the pressures on bonds 1, 5, and 18. The resulting *Extra variable* tab is also shown in Fig. 8.25.

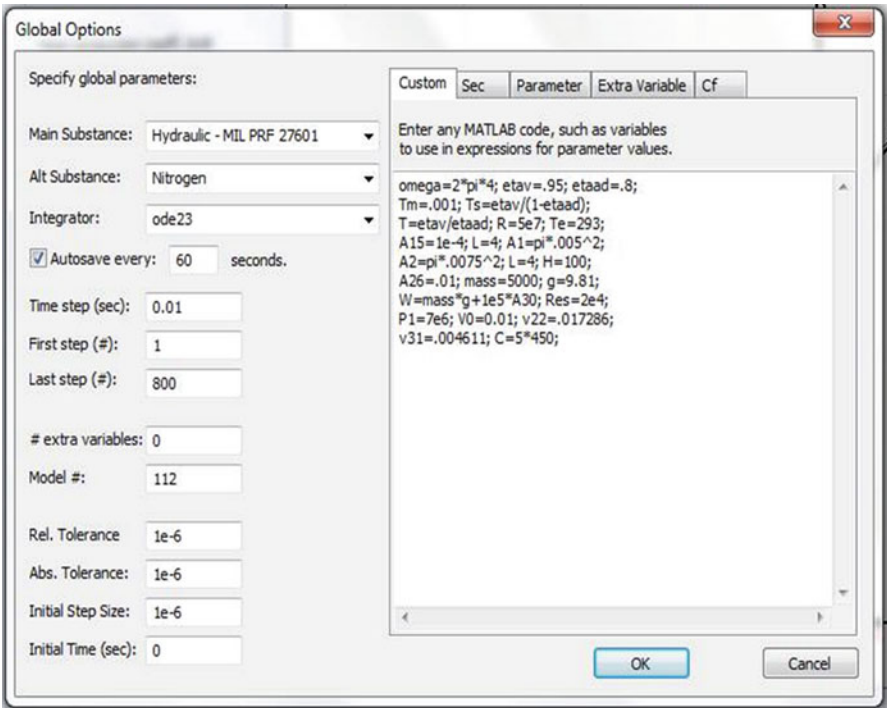
Direct plotting of an "extra" variable satisfies the "second use" mentioned above. To plot or recover a variable for the primary use, its *time derivative* must be computed using the differences between successive values. This is done using the MATLAB "diff" function, as described the Manual.

8.4.3 BondGrapher and the Hydraulic System

The main page of BondGrapher for the hydraulic system, given above in Fig. 8.15, shows that element CS22 has been clicked—it is given in color—and that the initial state tab is also clicked. This element represents the gas side of the low-pressure accumulator. Symbols are entered to represent the initial volume of the gas—nitrogen, since it is not flammable—and its initial temperature and mass. The chamber has no convection bond attached to it, so the mass of the gas is constant. Its volume decreases at the same rate as the volume of the liquid side of the accumulator increases (and vice versa), as mandated by the volume-change bond. Its temperature changes due to heat conduction with the environment through the heat conduction bond.

The *Global Options* page with the *Custom* tab clicked is shown in Fig. 8.26. The left side of this page includes the choices of the two fluids, the MATLAB integrator, and the size of the time steps and their number for the compressed data feature that was used. The values of the constant parameters are entered into the *Custom* tab.

The values of the non-constant parameters are entered into the *Parameter* tab of the *Global Options* page, as shown in Fig. 8.27. There, the relief valve as represented by RS25 is shut—has a zero orifice area—when the hydraulic pressure is less than 20 MPa. It is opened for times less than 3 s and more than 6 s into the run, as set by the user. Between these two times this valve opens by an amount proportional to the pressure difference across it, similar to the relief valve.



Specify global parameters:

Main Substance: Hydraulic - MIL PRF 27601

Alt Substance: Nitrogen

Integrator: ode23

☑ Autosave every: 60 seconds.

Time step (sec): 0.01

First step (#): 1

Last step (#): 800

extra variables: 0

Model #: 112

Rel. Tolerance 1e-6

Abs. Tolerance: 1e-6

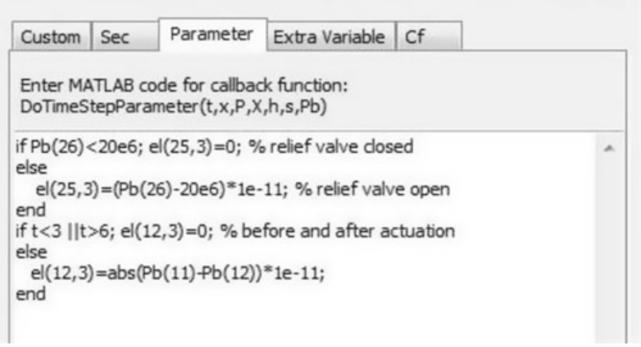
Initial Step Size: 1e-6

Initial Time (sec): 0

Enter any MATLAB code, such as variables
to use in expressions for parameter values.

```
omega=2*pi*4; etav=.95; etaad=.8;
Tm=.001; Ts=etav/(1-etaad);
T=etav/etaad; R=5e7; Te=293;
A15=1e-4; L=4; A1=pi*.005^2;
A2=pi*.0075^2; L=4; H=100;
A26=.01; mass=5000; g=9.81;
W=mass*g+1e5*A30; Res=2e4;
P1=7e6; V0=0.01; v22=.017286;
v31=.004611; C=5*450;
```

Fig. 8.26 The *Global Options* page with the *Custom* tab for the hydraulic system



Enter MATLAB code for callback function:
DoTimeStepParameter(t,x,P,X,h,s,Pb)

```
if Pb(26)<20e6; el(25,3)=0; % relief valve closed
else
   el(25,3)=(Pb(26)-20e6)*1e-11; % relief valve open
end
if t<3 ||t>6; el(12,3)=0; % before and after actuation
else
  el(12,3)=abs(Pb(11)-Pb(12))*1e-11;
end
```

Fig. 8.27 The *Parameter* tab for the hydraulic system

Saving the BondGrapher file with the name *hydraulicsys2.m* produces a MATLAB file which, when run, produces results that can be plotted using conventional MATLAB commands. Selected results are shown in Fig. 8.28.

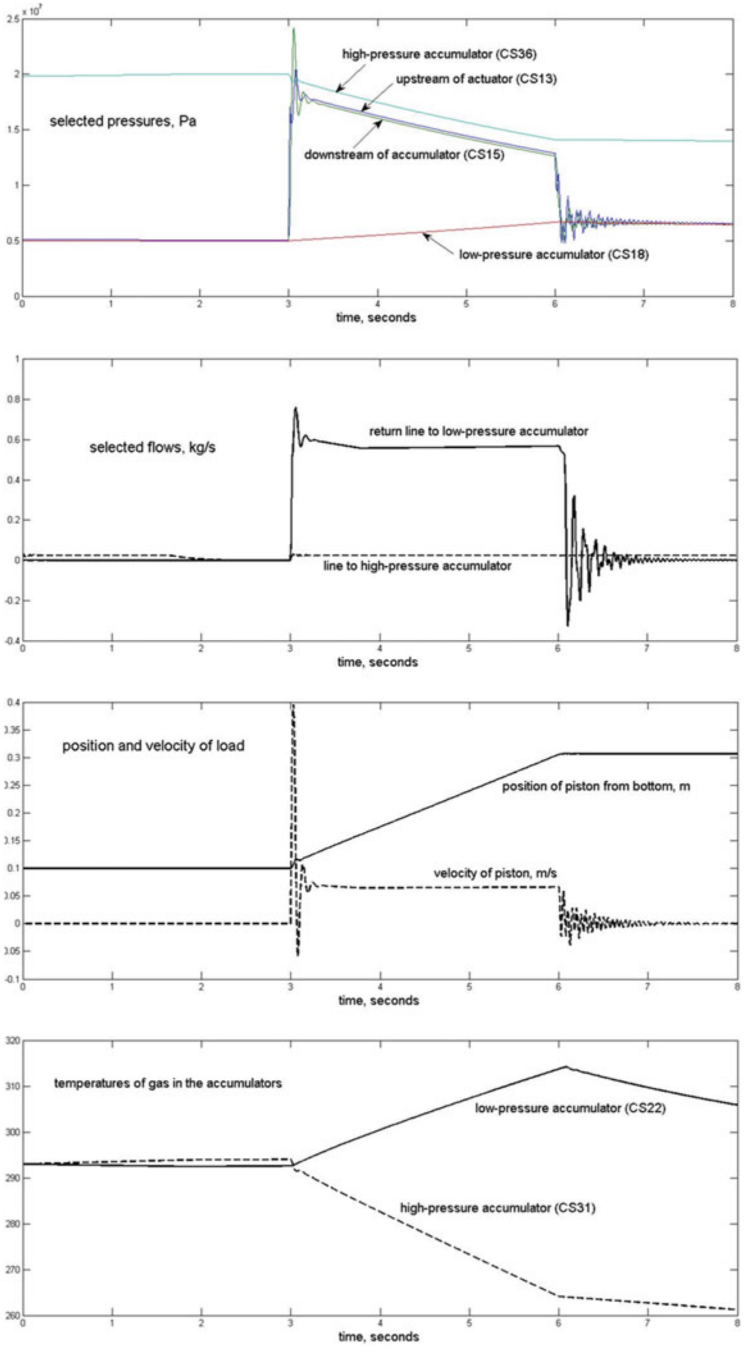
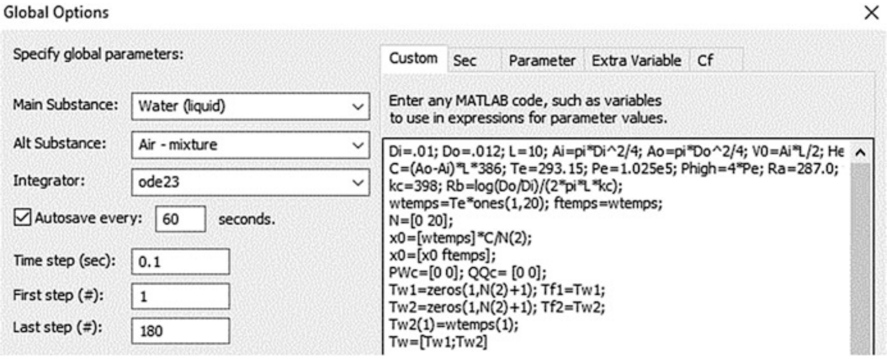


Fig. 8.28 Selected results from the simulation of the hydraulic actuator system



Specify global parameters:

Main Substance: Water (liquid)

Alt Substance: Air - mixture

Integrator: ode23

☑ Autosave every: 60 seconds.

Time step (sec): 0.1

First step (#): 1

Last step (#): 180

Enter any MATLAB code, such as variables
to use in expressions for parameter values.

```
Di=.01; Do=.012; L=10; Ai=pi*Di^2/4; Ao=pi*Do^2/4; V0=Ai*L/2; He
C=(Ao-Ai)*L*386; Te=293.15; Pe=1.025e5; Phigh=4*Pe; Ra=287.0;
kc=398; Rb=log(Do/Di)/(2*pi*L*kc);
wtemps=Te*ones(1,20); ftemps=wtemps;
N=[0 20];
x0=[wtemps]*C/N(2);
x0=[x0 ftemps];
PWc=[0 0]; QQc= [0 0];
Tw1=zeros(1,N(2)+1); Tf1=Tw1;
Tw2=zeros(1,N(2)+1); Tf2=Tw2;
Tw2(1)=wtemps(1);
Tw=[Tw1;Tw2]
```

Fig. 8.29 The Global Options screen with the Custom tab for the shower system

8.4.4 BondGrapher and Segmented IRS Elements

For segmented IRS elements for liquids the user must provide the initial temperatures of the *N* wall temperatures and *N* fluid temperatures. This is done in the *Custom* tab of the *Global Options* screen. This screen is shown in Fig. 8.29 for the shower problem introduced in Sect. 8.2.10 and Fig. 8.17. Note that all these initial temperatures are set equal to the environmental temperature (Te). The only higher initial temperature is that of the water heater, which is set 44 K higher in the parameter tab for Sec1 on the main screen of BondGrapher. The Custom tab also establishes the values of all the parameters. The inner diameter of the copper pipe is 0.01 m, its outer diameter is 0.012 m, and its length is 10 m. The effective orifice area of the two valves is one-fifth that of the inside of the pipe.

The main simulation program for pure liquids is *convecliq4.m*, which calls the script file *IRSliq4.m* to handle whatever IRS elements are present. This latter program also calls the function file *CtH.m* to treat the implicit Ct elements. For general highly compressible substances the respective files are *convec4.m, IRS4.m* and (again) *CtH.m.*

Selected results of a simulation of the start-up transient are shown in Fig. 8.30. For the specified valve settings, the mass flow rate reaches its equilibrium value of about 0.13 kg/s in under 0.2 s, giving a fluid velocity of about 1.7 m/s. Thus it takes about 6 s for a particle of water to traverse the pipe. It takes longer for the output stream to reach its equilibrium value, partly since the walls of the pipe absorb heat and partly because of the mixing in the pipe as represented by the CS volume which is set to equal half of the total volume of the pipe. The output temperature felt by the person showering was computed using the extra variable option; the temperature is that of bond 5, or Tb5. The computed rises in the temperatures of the water and the tube are slower than an exact analysis would give, and demonstrate some oscillation, due to the finite sizes of the segments, but the filtering effect of the fluid compliance is greater. The final temperature of the shower is lukewarm, despite the two valves

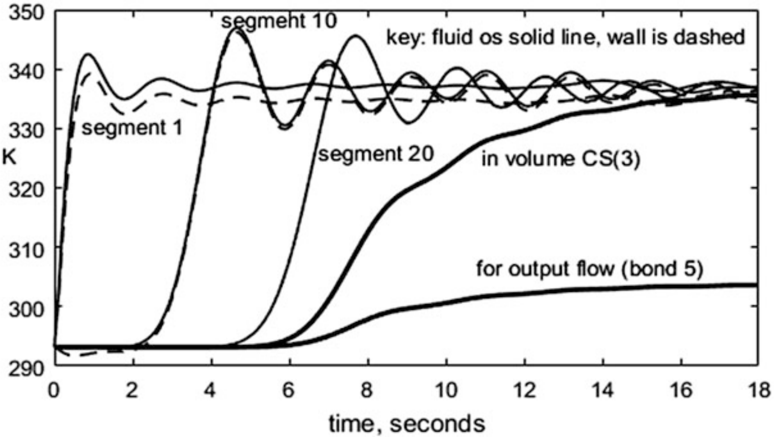


Fig. 8.30 Temperature results for the hot water shower system

having the same area, because of the large pressure drop through the pipe from turbulent viscous friction. The cold-water valve should have its area reduced.

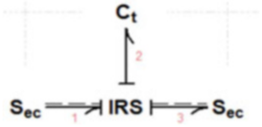
The temperatures of the tube wall segments are close to those of the water because the heat capacity of the walls is considerably smaller than that of the water, and the conduction rates are strong. The situation is very different, and therefore of greater importance, for highly compressible substances.

As of this writing, the author is updating the simulation files for IRS elements with highly compressible fluids that have produced preliminary results [10]. He expects to post corrected results on his website [9] before this book is published. The tentative results strongly suggest, for highly compressible fluids and especially those spanning the saturated mixture region, that fluid inertia plays a small role compared to that from fluid compliance coupled with heat transfer and wall friction. The inertia also increases execution time. One conclusion is that a new RS element should be developed that eliminates the inertia while retaining the other phenomena. Such an element does not appear to be particularly useful for pure liquids, however, because of their higher density and vastly smaller compressibility.

The analysis for highly compressible general substances employs the Models A and B shown in Figs. 8.31 and 8.32, with cryogenic nitrogen in the liquid, saturated mixture, and pure vapor phases. The two models each have a total of 48 segments, which for Model B, means 12 for each of its four IRS elements. The more elaborate Model B is used to illustrate the effects of the fluid compliance in a channel. The volumes of each of its three CS chambers are taken at one-quarter the volume of the entire tube. These are specified to be "total pressure" CS elements, as per the discussion at the end of Sect. 8.2 above.

The author presented models for a heat pump in 1999 [3] and in his textbook [5]; the latter is reproduced in Fig. 8.33. The analysis there employed the now

Fig. 8.31 Model A for a fluid channel with compressible flow



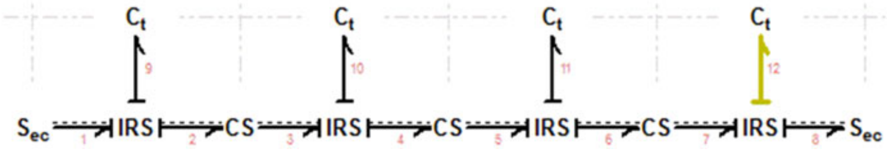


Fig. 8.32 Model B of the same channel

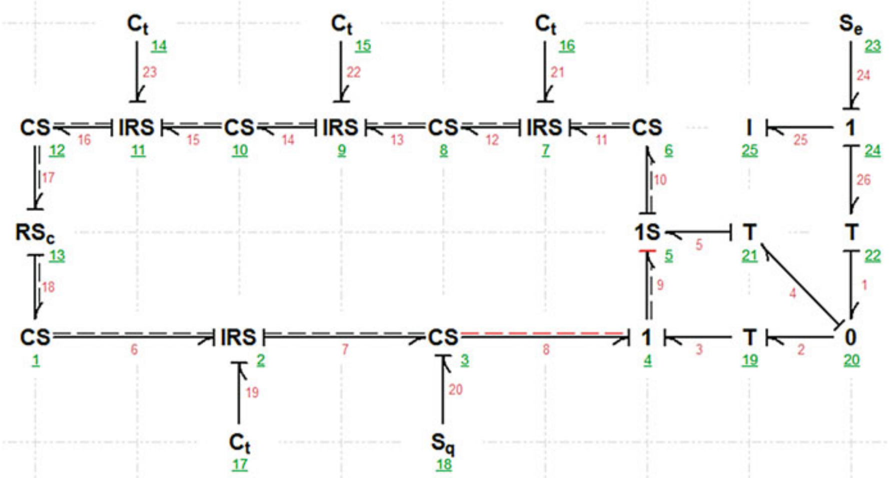


Fig. 8.33 Bond graph model of a heat pump system using IRS elements

discontinued first generation of the IRS element for compressible fluid flow, and is on the author's agenda for updating using the new IRS element.

The right side of this bond graph shows a compressor driven by an induction motor represented by Se23 and rotational inertia represented by I25 and transfer and 1- and 1S-junction elements that impart less than 100 % volumetric and adiabatic efficiencies, as with other systems above. The critical heat transfers take place on the inside and outside of tubes, represented by the four IRS elements and their attached Ct elements. The part of the evaporator that conducts heat—and exhibits fluid inertia and critical wall shear friction—is represented by CS1, IRS2, Ct12 which has the cold side thermal reservoir, and CS3. An electrical heater—used to insure that vapor enters the compressor—is modeled by the element Sq18. The thermal energy storage in the fluid there is represented by the elements CS1 and CS2, which split the volume of the tube equally. The evaporator is represented by the elements CS6, IRS7, Ct16, CS8, IRS9, Ct15, CS10, IRS11, Ct14, and CS12; the Ct elements there have the hot-side thermal reservoir. The expansion device is a simple orifice-type restriction, represented by the element RSc13.

Assuming the compressed-data option is used, THERMOSIM records time in the *time* vector, the momentum of the flow (product of length and mass flow rate), the state of the CS element, the temperatures and the thermal displacements of the wall (product of thermal compliance and temperatures) for each segment in the *state* matrix, the history of the wall temperatures for the interfaces in the *historyTwj* array, the history of the fluid temperatures for the segments in the *historyTfj* array, the history of the overall pressure drop due to wall shear in the *historyPWc* matrix, and the history of the overall heat-transfer rate between the fluid and the wall in the *historyQQc* matrix. Any number of other variables can be computed using the “extra variables” option.

8.5 Correlations for Viscosity, Thermal Conductivity, and Heat Transfer

The program *viscosity.m* addresses the thermal conductivity and viscosity of water, argon, nitrogen, oxygen, gaseous air, and somewhat restricted results for refrigerant R12. Three hydraulic fluids and liquid water are addressed separately. As of this writing the viscosity and thermal conductivity of many other fluids remain to be inserted into THERMOSIM; it is not easy to find appropriate correlations in the literature.

Heat transfer is an enormously complicated subject for which many people have devoted entire careers. For purposes of THERMOSIM, only a few commonly used approximations for equilibrium conditions have been incorporated for general use; the user can introduce others. Relations for heat conduction for simple shapes such as cylinders and tubes have been coded for several systems, using the Custom or Parameter tabs. The Parameter tab has also been used for convection correlations. Correlations for twelve different convection modes are incorporated into the program *IRS4.m* for internal and external convective heat transfer in tubes, which can also be borrowed for use with CS elements. Which applies depends on whether the fluid is a pure liquid, a mixture of liquid and vapor or a pure vapor, the Reynolds number of the flow and sometimes on the possible two phases of the flow, and whether there is heating or cooling. The correlations and their sources are listed [10]; the user may wish to substitute different correlations. Perhaps the most complex cases is boiling, for which the program has been adapted from a widely usable approximation given by Kandlikar [11]. Application is limited so far to only a few substances. A short function file *CtH.m* expedites approximating the external convective heat transfer between a tubular structure and the environment.

8.6 Conclusions

Convection bonds are non-traditional bonds having two efforts rather than one in order to represent simply and effectively the flow of a compressible fluid, particularly a highly compressible fluid. Their use is compatible with simple bonds, and introduces new bond graph elements that typically employ both types of bonds. THERMOSIM is a simulation package for the dynamic simulation of physical systems based on bond graphs with or without convection bonds. Its use is expedited by an exceptional graphical interface called BondGrapher, which the author himself could not have written.

THERMOSIM efficiently computes fairly accurately the thermodynamic properties of 35 different substances, using where necessary iterations based on analytic expressions for the gradients of properties derived by the author from available models of those substances. Nevertheless, for some practical models the necessary complexity of the software can make execution rather slow. The program also computes transport properties of several substances necessary for the calculation of fluid friction and heat transfer, but the list is far from complete.

Most systems that can be treated with other bond graph modeling and simulation software can also be treated by THERMOSIM, including models with activated bonds. Analysis of linear or linearized models such as the production of linear transfer functions or frequency-response analysis not included, however. The objective is simply the dynamic simulation of models, since the systems of primary interest are so highly nonlinear that linearized models are of little utility.

As of this writing the IRS element has been fully developed only for use with pure liquids; complete implementation for general substances should become available before this book is published. The results of the preliminary study of IRS elements for general substances nevertheless suggest that a new convection-bond element should be developed, similar to the IRS element but without the inertia. This conclusion does not apply to liquids.

One unique feature of THERMOSIM for non-thermodynamic systems which is of particular value to modeling and simulating electromechanical systems is its ability to deduce automatically the constitutive equations for nonlinear multiport compliances defined merely by expressions for their stored energy [7]. The Symbolic Math Toolbox of MATLAB is used.

The author is nearing the end of his career, and would welcome the interest of anyone who would like to investigate the possibility of contributing to THERMOSIM or eventually succeeding as its manager. One needn't be a thermodynamicist—the author is not—but have some interest in the subject.

References

1. Brown, F. T. (1991). Convection bond graphs. *Journal of the Franklin Institute, 328*(5/6), 871–886.
2. Brown, F. T. (1999). Applications of convection bonds. In: J. J. Granda & F. E. Cellier (Eds.), *1999 International Conference on Bond Graph Modeling and Simulation* (pp. 24–29). San Francisco, CA: The Society for Computer Simulation International (SCS).
3. Brown, F. T. (1999). Simulating thermodynamic systems. In: J. J. Granda & F. E. Cellier (Eds.), *1999 International Conference on Bond Graph Modeling and Simulation* (Vol. 31, no. 1, pp. 196–201). San Francisco, CA: The Society for Computer Simulation International (SCS).
4. Brown, F. T. (2003). Kinetic energy in convection bond graphs. In: F. E. Cellier & J. J. Granda (Eds.), *2003 International Conference on Bond Graph Modeling and Simulation* (pp. 2,191–197). Orlando, FL: The Society for Computer Simulation International (SCS).
5. Brown, F. T. (2007). *Engineering system dynamics: A graph-centered approach* (2nd ed.). Boca Raton, FL: CRC Press, Taylor and Francis.
6. Brown, F. T. (2007). *Bond graph modeling and simulation of thermodynamic systems*. PowerPoint presentation. Available from: www.lehigh.edu/~inmem.
7. Brown, F. T. (2012). Extension of simulation software for thermodynamic and other systems including energy-based modeling. *Procedings, 2012 International Conference on Bond Graph Modeling and Simulation,* The Society for Computer Simulation International (SCS), Monterey, CA.
8. Brown, F. T. (2014). Improved simulation of thermodynamic and other systems using a new graphical interface. In: J. J. Granda & D. C. Karnopp (Eds.), *2014 International Conference on Bond Graph Modeling and Simulation* (pp. 157–168). Monterey, CA: The Society for Computer Simulation International (SCS).
9. Brown, F. T. (2014, to be updated 2016). Website: www.lehigh.edu/~inmem click on *faculty, emeritus*, and *Forbes T. Brown*.
10. Brown, F. T. (2016). A new convection bond element for multiphase flow through a channel. In: J. J. Granda & D. C. Karnopp (Eds.), 2016 *International Conference on Bond Graph Modeling and Simulation,* Montreal, Canada: The Society for Computer Simulation International (SCS).
11. Kandlikar, S. G. (1990). A general correlation for saturated two-phase boiling heat transfer inside horizontal and vertical tubes. *Journal of Heat Transfer, 112*, 219–228.
12. Reynolds, W. C. (1979). *Thermodynamic properties in SI*. Stanford, CA: Department of Mechanical Engineering, Stanford University.

Chapter 9
Finite Element Formulation for Computational Fluid Dynamics Framed Within the Bond Graph Theory

Jorge Luis Baliño

9.1 Introduction

9.1.1 The Bond Graph Formalism

Since the invention of the Bond Graph (BG) formalism by Henry Paynter in 1959, this technique has become a powerful tool for modeling and simulating dynamic systems. Used in the beginning in the fields of Electrical and Mechanical Engineering, applications in different areas such as Thermodynamics and Electrodynamics have been successful [13–16, 18, 22, 29, 32, 39, 43].

Once the BG representation of a system has been obtained, there is a systematic procedure for: (a) determining a set of state variables; (b) defining the input variables by means of generalized effort and flow sources; (c) assigning causality, which assures the mathematical well-posedness of the state equations; and (d) obtaining the state equations and output variables.

In this way, the BG formalism can be regarded as a filter through which mathematical inconsistencies can be detected in the modeling process from the beginning.

9.1.2 Numerical Methods for Computational Fluid Dynamics

An important fraction of the problems in Fluid Mechanics falls within what is called Computational Fluid Dynamics (CFD). This branch of the human knowledge has

J.L. Baliño (✉)
Departamento de Engenharia Mecânica, Escola Politécnica, Universidade de São Paulo,
Av. Prof. Mello Moraes, 2231, Cidade Universitária, CEP 05508-900 São Paulo-SP, Brazil
e-mail: jlbalino@usp.br

W. Borutzky (ed.), *Bond Graphs for Modelling, Control and Fault Diagnosis of Engineering Systems*, DOI 10.1007/978-3-319-47434-2_9

nurtured the development of various numerical approaches, being the most popular ones the Finite Element Method (FEM) [23, 34, 47], the Finite Volume Method (FVM) [35], and the Finite Difference Method (FDM) [42].

Although a separate comprehensive introduction to these numerical approaches is not attempted, it can be said that all these methods try to solve the problem by discretizing the continua, that is, by replacing the continuous variables by a combination of a finite set of nodal values and interpolating functions. The result is a (generally nonlinear) algebraic problem, instead of the original differential or integro-differential one. No matter what kind of numerical method is selected, they all end up with an algebraic system of a large number of unknowns.

The members of this family of methods, called CFD solvers, have all advantages and disadvantages with respect to each other. Some of them are easy to understand and usually also easy to implement, but may lack of generality, like the FDM. On the other hand, some methods may lead to more general solvers but are in turn theoretically more involved, like the FEM or the FVM.

9.1.3 Motivation

In the field of Fluid Dynamics, the potential benefits of BG have not been yet fully exploited. The applications to fluid dynamic systems were not oriented to a systematic spatial discretization of flow fields, typical of CFD problems. Fluid Dynamics is a challenging area for bondgraphers, because these systems are rigorously described by nonlinear partial differential equations (PDEs) with important spatial effects and exhibit couplings between different energy domains.

In [16] the concept of *convection bond* was introduced, as a way of dealing with the coupling between mechanical and thermal energy domains in a compressible flow. A convection bond has two independent variables to describe the effort (pressure and stagnation enthalpy) and a flow variable (mass flow). The true effort variable is the stagnation enthalpy, being pressure an auxiliary variable. Full upwind (upstream value) is used for the evaluation of the stagnation enthalpy.

A pioneer contribution linking BG to CFD appeared in [26], although the formulation was restricted to prescribed shape functions and nodalization. Besides, heat conduction (which leads to advection-diffusion problems) was not modeled. A systematic procedure, within the frame of the BG theory, for integrating in volume the power conservation equations in order to get the state equations would be desirable, in order to bridge the gap between the continuous (distributed parameter) formulation of the conservation equations and a discretized (lumped parameter) numerical scheme.

It is well known that the BG representation depicts in a very elegant way conservation of energy in the various forms in which it may appear in a given dynamic, lumped-parameter system. The definition of suitable generalized effort and flow variables, based on the system total energy, allows to obtain the state equations in an orderly fashion.

Another characteristic of the BG approach is the natural representation of dynamic systems with mixed energy domains. This feature is interesting when dealing with CFD problems because in a multidimensional, viscous, compressible flow with heat transfer, the right understanding of the transformation of mechanical energy into thermal energy, as well as the generation of irreversibility, is essential to assure mathematical well-posedness and realistic results.

CFD solvers share many well-known nodalization-related difficulties [35]. Advection dominated flows may give rise to numerical instabilities that require upwinding or other stabilization techniques. Checkerboard pressure fields are also a common problem, requiring the use of staggered grids or equal-order methods. It would be interesting to find out whether a BG approach suffers from these difficulties (as well as many others) and if it is so, how they are handled and what tools are rendered to solve them.

The methodology presented in this chapter has the following characteristics, briefly described as follows:

1. The methodology is based on the representation of the total energy per unit volume as a function of a set of independent variables. The time derivative of the total energy per unit volume is represented at the continuum level by a summation of products of generalized potentials (constitutive relations) and time derivatives of the independent variables. This representation satisfies the Maxwell relations [17].
2. The balance equations, corresponding to each one of the terms appearing in the time derivative of the total energy rate per unit volume, are derived based on the PDEs representing the conservation laws; in this way, all physical effects can naturally be taken into account. The balance equations represent the power structure of the system at the continuum level, showing three type of terms: (a) Divergence terms, related to the boundary conditions; (b) Source terms, and (c) Coupling terms, related to the power transfer between the different energy domains, appearing as terms with opposite signs in pairs of balance equations.
3. The discretization is made in terms of time-dependent nodal values of the independent variables and position-dependent interpolation functions, characteristic of the FEM. Nodal state variables and associated nodal potentials are defined in a straightforward manner. In this way, all the properties are kept for the energy representation at the discretized level, resulting generalized effort and flow variables characteristic of true BG.
4. The state equations are obtained by establishing weak formulations of the continuous problem, as done in the FEM. The chosen test functions are the shape functions for the momentum conservation equation and weight functions for the mass and/or entropy conservation equations, resulting respectively a Galerkin and a Petrov–Galerkin method. The weight functions are suitable for introducing upwind or other numerical schemes, depending on the mathematical nature of the problem.

Consequently, the formulation links two areas, namely the BG methodology and CFD. It is interesting to see that, although temperature and entropy rate are the natural BG variables in thermal problems, bondgraphers resort to pseudo-BG or other non-BG elements when modeling thermofluid flow problems [32, 43], as if there were some intrinsic difficulty in incorporating naturally all the relevant physical effects.

As the BG approach allows the easy interconnection of systems of different energy domains, it is foreseeable that the findings of this contribution could be used for problems such as fluid-structure interaction, Magnetohydrodynamics, etc.

9.1.4 Organization

The organization of this contribution is outlined as follows. In Sect. 9.2 the methodology is applied to a single-phase, single-component, multidimensional compressible flow with viscosity and thermal effects. In Sect. 9.3 the methodology is applied to a flow in which the incompressibility constraint is set, showing the differences with the general case. In Sect. 9.4 numerical applications obtained with the methodology outlined above are shown, as well as the extension of the BG methodology to other flow problems. Finally, Sect. 9.5 shows the main conclusions and perspectives that can be drawn from the work done so far.

9.1.5 Notation

In the following, bold letters will be used to define vectors ($\boldsymbol{V}, \boldsymbol{p}_{v}$, etc.). With regard to matrix notation, column vectors associated with nodal values will be denoted by single underscored plain or bold type ($\underline{m}$, $\underline{S}$, $\underline{\boldsymbol{V}}$, $\underline{\varphi_{\rho}}$, etc.) while multidimensional matrices will be identified by double underscored type ($\underline{\underline{M}}$, $\underline{\underline{\Omega_{\rho}}}$, etc.). Second order tensors will be denoted by double underscored type ($\underline{\underline{\tau}}$, $\underline{\underline{I}}$, etc.). In some cases, it will be convenient to express an entity either as a column vector or as a matrix (for instance, $\underline{\Theta}$ and $\underline{\underline{\Theta}}$). The superscript T will be used to denote the transpose. Einstein convention of summation over repeated indices is *not* used. All the variables will be defined as they appear in the contribution.

9.2 Compressible Flows

Compressible flow is a branch of Fluid Mechanics that studies flows in which there are significant changes in the fluid density. When the fluid velocity is comparable to the speed of sound, compressibility effects become important [40].

An important parameter in compressible flows is the Mach number Ma, which is the ratio of the particle speed to the local speed of sound; as a rule of thumb, compressibility effects must be taken into account when $Ma > 0.3$. Mass, momentum, and energy conservation equations are coupled and an equation of state is needed to close the problem. As this condition is frequently encountered in gas systems (usually the fluid is considered as an ideal gas), the object of study is known as gas dynamics.

Basic studies of compressible pipe flows assume that viscous and heat transfer effects are negligible, leading to isentropic flows; performance of converging and converging-diverging nozzles are typical examples studied with this approximation. More sophisticated studies take into account wall shear stresses (Fanno flows) and heating/cooling (Rayleigh flows). Distinctive effects in compressible flows are the limitation of the mass flow rate when the local velocity equals the sound of speed (choking) and existence of very thin discontinuities in the flow properties, associated with irreversibilities (shock waves). The study of compressible flow is relevant to high-speed aircraft, jet engines, rocket motors, gas pipelines, and many other fields.

In this section a single-phase, single-component, multidimensional compressible flow with viscosity and thermal effects is considered for the application of the BG methodology [9, 10].

9.2.1 Continuum Formulation

In this section the representation of the power structure and physical model is presented at the continuum level, that is, for a control volume of differential size. The dynamic equations are presented in a form that is suitable to be framed, after integration in the system volume, within the BG theory.

9.2.1.1 Energy and Power Representation

The total energy per unit volume e_v ($e_v = \rho\,\hat{e}$, where ρ is the density and $\hat{e}$ is the total energy per unit mass) is defined as the sum of the internal energy per unit volume u_v ($u_v = \rho\,\hat{u}$, where $\hat{u}$ is the internal energy per unit mass) and the kinetic coenergy per unit volume t_v^*:

$$e_v = u_v\left(\rho,\, s_v\right) + t_v^* \tag{9.1}$$

It is assumed that the internal energy per unit volume is a function of density and entropy per unit volume s_v ($s_v = \rho\,\hat{s}$, where $\hat{s}$ is the entropy per unit mass). The kinetic coenergy per unit volume is defined as:

$$t_v^* = \frac{1}{2}\,\rho\,\boldsymbol{V}^2 \tag{9.2}$$

where $\boldsymbol{V}$ is the velocity. The following potentials are defined [17]:

$$\boldsymbol{p}_v = \left(\frac{\partial t_v^*}{\partial \boldsymbol{V}}\right)_\rho = \rho\,\boldsymbol{V} \tag{9.3}$$

$$\kappa = \left(\frac{\partial t_v^*}{\partial \rho}\right)_{\boldsymbol{V}} = \frac{1}{2}\,\boldsymbol{V}^2 \tag{9.4}$$

$$\theta = \left(\frac{\partial u_v}{\partial s_v}\right)_\rho \tag{9.5}$$

$$\psi = \left(\frac{\partial u_v}{\partial \rho}\right)_{s_v} = \frac{1}{\rho}\,(u_v + P - \theta\, s_v) \tag{9.6}$$

where $\boldsymbol{p}_v$, κ, θ, ψ, and P are respectively the linear momentum per unit volume, the kinetic coenergy per unit mass, the absolute temperature, the Gibbs free energy per unit mass, and the absolute pressure. The time derivative of the total energy per unit volume can be written as:

$$\frac{\partial e_v}{\partial t} = (\psi + \kappa)\,\frac{\partial \rho}{\partial t} + \boldsymbol{p}_v \cdot \frac{\partial \boldsymbol{V}}{\partial t} + \theta\,\frac{\partial s_v}{\partial t} \tag{9.7}$$

Notice that the energy and power representation is made independently of the approximations and fluid constitutive laws chosen for the conservation equations; the same representation could be used, for instance, for a newtonian or non-newtonian fluid. Besides, the right-hand side of Eq. (9.7) determines the power terms that must be defined from the physical flow model.

An alternative formulation can be derived by taking $\boldsymbol{p}_v$ instead of $\boldsymbol{V}$ as independent variable. In this case, the formulation would be symmetric, in a sense that the volume integrals of the independent variables would result in the system mass, linear momentum, and entropy. Nevertheless, velocity is chosen because it is more popular as discretized variable and because the resulting expressions are easier to calculate.

A comment should be made concerning the existence of the potentials for incompressible flows. For this particular case, as density is a constant, the internal energy per unit volume is a function of the entropy per unit volume only, being the pressure an external function which must adjust to satisfy the incompressibility condition. Thus, pressure is no longer a thermophysical property and Eq. (9.6) is no longer valid. Incompressible flows deserve a special treatment, presented in Sect. 9.3.

9.2.1.2 Constitutive and Maxwell Relations

The resulting constitutive relations come from the first derivatives of the total energy per unit volume:

$$\psi + \kappa = \psi\,(\rho,\, s_v) + \kappa\,(\boldsymbol{V}) \tag{9.8}$$

$$\boldsymbol{p}_v = \boldsymbol{p}_v\,(\rho,\,\boldsymbol{V}) = \rho\,\boldsymbol{V} \tag{9.9}$$

$$\theta = \theta\,(\rho,\,s_v) \tag{9.10}$$

For the total energy be a single-valued function of the independent variables ρ, $\boldsymbol{V}$ and s_v, the constitutive relations must satisfy the Maxwell relations, which arise from the equality of the mixed partial derivatives. These derivatives can be written as a function of the independent variables, through the constitutive relations and three independent derivatives [17], which can be chosen as:

$$\alpha = -\frac{1}{\rho}\left(\frac{\partial \rho}{\partial \theta}\right)_P \tag{9.11}$$

$$c_v = \frac{\theta}{\rho}\left(\frac{\partial s_v}{\partial \theta}\right)_\rho \tag{9.12}$$

$$\kappa_\theta = \frac{1}{\rho}\left(\frac{\partial \rho}{\partial P}\right)_\theta \tag{9.13}$$

where α, c_v and κ_θ are respectively the coefficient of thermal expansion, the constant volume specific heat, and the coefficient of isothermal compressibility. In this way, the Maxwell relations can be written as:

$$\left(\frac{\partial \theta}{\partial \rho}\right)_{s_v} = \left[\frac{\partial}{\partial s_v}\,(\psi + \kappa)\right]_\rho = \left(\frac{\partial \psi}{\partial s_v}\right)_\rho = \frac{\theta}{\rho\,c_v}\left(\frac{\alpha}{\rho\,\kappa_\theta} - \frac{s_v}{\rho}\right) \tag{9.14}$$

$$\left(\frac{\partial \theta}{\partial \boldsymbol{V}}\right)_{s_v} = \left(\frac{\partial \boldsymbol{p}_v}{\partial s_v}\right)_{\boldsymbol{V}} = \boldsymbol{0} \tag{9.15}$$

$$\left[\frac{\partial}{\partial \boldsymbol{V}}\,(\psi + \kappa)\right]_\rho = \left(\frac{\partial \kappa}{\partial \boldsymbol{V}}\right)_\rho = \left(\frac{\partial \boldsymbol{p}_v}{\partial \rho}\right)_{\boldsymbol{V}} = \boldsymbol{V} \tag{9.16}$$

For the particular case of an ideal gas, the internal energy per unit volume and thermodynamic potentials result:

$$u_v = \rho\,c_v\,\theta \tag{9.17}$$

$$P = \rho\,c_v\,(\gamma - 1)\,\theta \tag{9.18}$$

$$\psi = \left(\gamma\,c_v - \frac{s_v}{\rho}\right)\theta \tag{9.19}$$

$$\theta = \theta_R\left(\frac{\rho}{\rho_R}\right)^{\gamma-1}\exp\left(\frac{s_v}{\rho\,c_v}\right) \tag{9.20}$$

where γ is the heat capacity ratio and ρ_R and θ_R are respectively a reference density and temperature for which the entropy per unit volume is zero.

9.2.1.3 Conservation Equations

The mass, linear momentum, and thermal energy conservation equations for a single-phase, single-component fluid are [44]:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{V}) = 0 \tag{9.21}$$

$$\rho \frac{\partial \boldsymbol{V}}{\partial t} = -\rho \nabla \kappa + \rho \boldsymbol{V} \times (\nabla \times \boldsymbol{V}) - \nabla P + \rho \boldsymbol{G} + \nabla \cdot \underline{\underline{\tau}} \tag{9.22}$$

$$\frac{\partial u_v}{\partial t} = -\nabla \cdot (u_v \boldsymbol{V}) - \nabla \cdot \boldsymbol{q} - P \nabla \cdot \boldsymbol{V} + \nabla \boldsymbol{V} : \underline{\underline{\tau}} + \rho \Phi \tag{9.23}$$

where t is the time, $\boldsymbol{G}$ is the force per unit mass, $\underline{\underline{\tau}}$ is the viscous stress tensor, $\boldsymbol{q}$ is the heat flux, and Φ is the heat source per unit mass. If the reference coordinate system used to describe the flow is not inertial, the force per unit mass must include the non-inertial forces. Notice that all physical effects (compressibility, heat transfer, viscous dissipation) are included in the analysis.

9.2.1.4 Balance Equations

The conservation equations have different units and are normally the starting equations to be discretized in the numerical methods. The balance equations, which can be derived starting from the conservation equations and the constitutive relations, are power equations (per unit volume) corresponding to each one of the terms that contributes to the time derivative of the total energy per unit volume, namely Eq. (9.7). Multiplying Eq. (9.21) by $\psi + \kappa$ it results:

$$(\psi + \kappa) \frac{\partial \rho}{\partial t} = -\nabla \cdot [\rho (\psi + \kappa) \boldsymbol{V}] + \rho \boldsymbol{V} \cdot \nabla \psi + \rho \boldsymbol{V} \cdot \nabla \kappa \tag{9.24}$$

Making the dot product of Eq. (9.22) and the velocity and taking into account the following identities:

$$[\boldsymbol{V} \times (\nabla \times \boldsymbol{V})] \cdot \boldsymbol{V} = 0 \tag{9.25}$$

$$\left(\nabla \cdot \underline{\underline{\tau}}\right) \cdot \boldsymbol{V} = \nabla \cdot \left(\underline{\underline{\tau}} \cdot \boldsymbol{V}\right) - \nabla \boldsymbol{V} : \underline{\underline{\tau}} \tag{9.26}$$

it can be obtained:

$$\boldsymbol{p}_v \cdot \frac{\partial \boldsymbol{V}}{\partial t} = \nabla \cdot \left(\underline{\underline{\tau}} \cdot \boldsymbol{V}\right) + \rho \boldsymbol{G} \cdot \boldsymbol{V} - \boldsymbol{V} \cdot \nabla P - \nabla \boldsymbol{V} : \underline{\underline{\tau}} - \rho \boldsymbol{V} \cdot \nabla \kappa \tag{9.27}$$

Finally, from Eqs. (9.5), (9.6), (9.21), and (9.23) it results:

$$\theta\,\frac{\partial s_v}{\partial t} = -\nabla\cdot\boldsymbol{q} - \nabla\cdot(\theta\,s_v\,\boldsymbol{V}) + \rho\,\Phi - \rho\,\boldsymbol{V}\cdot\nabla\psi + \boldsymbol{V}\cdot\nabla P + \nabla\boldsymbol{V} : \underline{\underline{\tau}} \tag{9.28}$$

As stated before, one of the key issues in modeling fluid dynamic systems with viscous, compressible, and thermal effects is the right understanding of the transformation of mechanical to thermal energy, and the generation of irreversibility. The balance equations show one of the advantages of this methodology, that is, the representation of the power structure of the system. In the balance equations there can be identified three type of terms: divergence, source, and coupling terms. The divergence terms take into account the power introduced in the system through the boundary conditions. The source terms constitute the different power sources, external to the system. Finally, the coupling terms represent power transfer between the velocity, mass, and entropy balance equations; these coupling terms appear, with opposite signs, in pairs of balance equations.

The power transfer between the velocity and entropy balance equations is represented by the irreversible term $\nabla\boldsymbol{V} : \underline{\underline{\tau}} \geq 0$ (also known as *viscous dissipation function*) and the reversible term $\boldsymbol{V}\cdot\nabla\,P$. The power transfer between the velocity and mass balance equations is represented by the term $\rho\,\boldsymbol{V}\cdot\nabla\kappa$, while the power transfer between the mass and entropy balance equations is represented by the term $\rho\,\boldsymbol{V}\cdot\nabla\psi$.

Taking into account Eq. (9.7), it can be verified that coupling terms cancel out when the balance equations are added, resulting the conservation of total energy:

$$\frac{\partial e_v}{\partial t} + \nabla\cdot(e_v\,\boldsymbol{V}) = \nabla\cdot\left[\left(-P\,\underline{\underline{I}} + \underline{\underline{\tau}}\right)\cdot\boldsymbol{V}\right] - \nabla\cdot\boldsymbol{q} + \rho\,\boldsymbol{G}\cdot\boldsymbol{V} + \rho\,\Phi \tag{9.29}$$

The cancelation of the coupling terms means that they influence the power distribution among the different energy domains but not the total power in the system.

The pressure and Gibbs free energy gradients can be written as a function of the independent variables as:

$$\begin{aligned}\nabla P &= \left(\frac{\partial P}{\partial s_v}\right)_\rho \nabla s_v + \left(\frac{\partial P}{\partial \rho}\right)_{s_v} \nabla\rho \\ &= \frac{\theta}{\rho\,c_v}\,\frac{\alpha}{\kappa_\theta}\left[\nabla s_v + \left(\frac{c_v}{\alpha\,\theta} + \frac{\alpha}{\rho\,\kappa_\theta} - \frac{s_v}{\rho}\right)\nabla\rho\right]\end{aligned} \tag{9.30}$$

$$\begin{aligned}\nabla\psi &= \left(\frac{\partial \psi}{\partial s_v}\right)_\rho \nabla s_v + \left(\frac{\partial \psi}{\partial \rho}\right)_{s_v} \nabla\rho \\ &= \frac{\theta}{\rho\,c_v}\left(\frac{\alpha}{\rho\,\kappa_\theta} - \frac{s_v}{\rho}\right)\nabla s_v + \left[\frac{1}{\rho^2\,\kappa_\theta} + \frac{\theta}{\rho\,c_v}\left(\frac{\alpha}{\rho\,\kappa_\theta} - \frac{s_v}{\rho}\right)^2\right]\nabla\rho\end{aligned} \tag{9.31}$$

For the rest of the unknown variables in the balance equations, a definition of additional fluid constitutive relations is needed. If the fluid obeys Fourier's law, the heat flux can be calculated as:

$$
\begin{aligned}
\boldsymbol{q} = -\lambda\,\nabla\theta &= -\lambda\left[\left(\frac{\partial\theta}{\partial s_v}\right)_{\rho}\nabla s_v + \left(\frac{\partial\theta}{\partial\rho}\right)_{s_v}\nabla\rho\right] \\
&= -\frac{\lambda\,\theta}{\rho\,c_v}\left[\nabla s_v + \left(\frac{\alpha}{\rho\,\kappa_\theta} - \frac{s_v}{\rho}\right)\nabla\rho\right]
\end{aligned}
\tag{9.32}
$$

where λ is the thermal conductivity.

Finally, the viscous stress tensor $\underline{\underline{\tau}}$ can be expressed for a newtonian fluid, considering Stoke's hypothesis, as:

$$
\underline{\underline{\tau}} = \mu\left(\nabla\boldsymbol{V} + \nabla\boldsymbol{V}^T\right) - \frac{2}{3}\,\mu\,(\nabla\cdot\boldsymbol{V})\,\underline{\underline{I}}
\tag{9.33}
$$

where μ is the fluid viscosity and $\nabla\boldsymbol{V}$ and $\underline{\underline{I}}$ are respectively the velocity gradient tensor and the identity tensor.

9.2.2 Discrete Formulation

9.2.2.1 Description of the Flow Fields

In order to formulate the discrete model of the fluid continuum in the domain Ω, as shown in Fig. 9.1, it is necessary to specify the description of the flow fields corresponding to the independent variables. As it is done in the FEM [47], this description is made in terms of a finite set of nodal values and interpolation functions:

$$
\rho\,(\boldsymbol{r},\,t) = \sum_{k=1}^{n_\rho}\rho_k\,(t)\;\varphi_{\rho\,k}\,(\boldsymbol{r}) = \underline{\rho}^T\cdot\underline{\varphi_\rho}
\tag{9.34}
$$

$$
\boldsymbol{V}\,(\boldsymbol{r},\,t) = \sum_{m=1}^{n_V}\boldsymbol{V}_m\,(t)\;\varphi_{V\,m}\,(\boldsymbol{r}) = \underline{\boldsymbol{V}}^T\cdot\underline{\varphi_V}
\tag{9.35}
$$

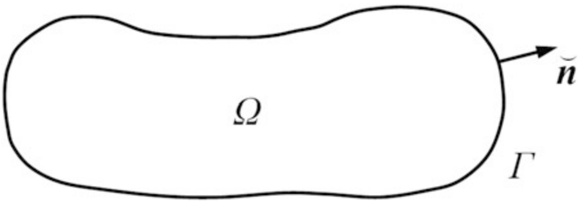


Fig. 9.1 Domain Ω with boundary Γ and unit outer normal $\breve{\boldsymbol{n}}$

$$s_v(\boldsymbol{r}, t) = \sum_{l=1}^{n_S} s_{v\,l}(t)\ \varphi_{S\,l}(\boldsymbol{r}) = \underline{s_v}^T \cdot \underline{\varphi_S} \tag{9.36}$$

where $\underline{\rho}$ (size n_ρ), $\underline{\boldsymbol{V}}$ (size n_V) and $\underline{s_v}$ (size n_S) are time-dependent nodal vectors, while $\underline{\varphi_\rho}$, $\underline{\varphi_V}$ and $\underline{\varphi_S}$ are the corresponding position-dependent nodal interpolation or shape functions. The interpolation functions have the following properties:

$$\sum_{k=1}^{n_\rho} \varphi_{\rho\,k}(\boldsymbol{r}) = 1 \qquad \forall\, \boldsymbol{r} \in \Omega \tag{9.37}$$

$$\sum_{m=1}^{n_V} \varphi_{V\,m}(\boldsymbol{r}) = 1 \qquad \forall\, \boldsymbol{r} \in \Omega \tag{9.38}$$

$$\sum_{l=1}^{n_S} \varphi_{S\,l}(\boldsymbol{r}) = 1 \qquad \forall\, \boldsymbol{r} \in \Omega \tag{9.39}$$

For simplicity in the treatment of the boundary conditions, it is also required for the interpolation functions to be equal to one at the reference node, and to be equal to zero at the rest of the nodes, that is:

$$\varphi_{\rho\,k}(\boldsymbol{r}_n) = \delta_{k\,n}, \text{ for a density node located at position } \boldsymbol{r}_n \tag{9.40}$$

$$\varphi_{V\,m}(\boldsymbol{r}_n) = \delta_{m\,n}, \text{ for a velocity node located at position } \boldsymbol{r}_n \tag{9.41}$$

$$\varphi_{S\,l}(\boldsymbol{r}_n) = \delta_{l\,n}, \text{ for an entropy node located at position } \boldsymbol{r}_n \tag{9.42}$$

In Eqs. (9.40)–(9.42), $\delta_{k\,n}$ is the Kronecker's delta ($\delta_{k\,n} = 1$ if $k = n$, $\delta_{k\,n} = 0$ otherwise).

The representation of the flow fields in terms of nodal values and interpolation functions allows to define the corresponding values at any position within Ω, so it is possible to calculate univocally all the integrals corresponding to the state equations; this is not evident for other methodologies like FDM or FVM, where only nodal values are defined and additional considerations must be made in order to integrate the differential equations. Besides, the chosen representation can make use of the considerable amount of computational tools already available for the popular FEM.

9.2.2.2 Integrated Variables

Nodal vectors of integrated values are defined, related to the discretized ones as:

$$\underline{m} = \underline{\underline{\Omega_\rho}} \cdot \underline{\rho} \tag{9.43}$$

$$\underline{S} = \underline{\underline{\Omega_S}} \cdot \underline{s_v} \tag{9.44}$$

The diagonal volume matrices $\underline{\underline{\mathit{\Omega}_{\rho}}}$ and $\underline{\underline{\mathit{\Omega}_{S}}}$, respectively associated with the density and entropy per unit volume, are defined as:

$$\left\{\mathit{\Omega}_{\rho}\right\}_{kn} = \mathit{\Omega}_{\rho k}\, \delta_{kn} \tag{9.45}$$

$$\left\{\mathit{\Omega}_{S}\right\}_{ln} = \mathit{\Omega}_{S l}\, \delta_{ln} \tag{9.46}$$

where:

$$\mathit{\Omega}_{\rho k} = \int_{\mathit{\Omega}} \varphi_{\rho k}\, d\mathit{\Omega} \tag{9.47}$$

$$\mathit{\Omega}_{S l} = \int_{\mathit{\Omega}} \varphi_{S l}\, d\mathit{\Omega} \tag{9.48}$$

The system mass m and entropy S are related to the integrated variables as follows:

$$m = \int_{\mathit{\Omega}} \rho\, d\mathit{\Omega} = \sum_{k=1}^{n_{\rho}} m_k \tag{9.49}$$

$$S = \int_{\mathit{\Omega}} s_v\, d\mathit{\Omega} = \sum_{l=1}^{n_S} S_l \tag{9.50}$$

9.2.2.3 System Total Energy

The system total energy E is defined as the sum of the system internal energy U and the system kinetic coenergy T^*:

$$E = U\left(\underline{m},\, \underline{S}\right) + T^*\left(\underline{m},\, \underline{\boldsymbol{V}}\right) \tag{9.51}$$

where:

$$E = \int_{\mathit{\Omega}} e_v\, d\mathit{\Omega} \tag{9.52}$$

$$U = \int_{\mathit{\Omega}} u_v\, d\mathit{\Omega} \tag{9.53}$$

$$T^* = \int_{\mathit{\Omega}} t_v^*\, d\mathit{\Omega} \tag{9.54}$$

From Eqs. (9.35) and (9.54), it can be easily shown that the system kinetic coenergy can be expressed as the following bilinear form:

$$T^* = \frac{1}{2}\, \underline{\boldsymbol{V}}^T \cdot \underline{\underline{\boldsymbol{M}}} \cdot \underline{\boldsymbol{V}} \tag{9.55}$$

where $\underline{\underline{M}}$ is the system inertia matrix (size n_V, symmetric and regular):

$$\{M\}_{m\,n} = \int_{\Omega} \rho\, \varphi_{V\,m}\, \varphi_{V\,n}\, d\Omega \tag{9.56}$$

The following potentials are defined:

$$\underline{\boldsymbol{p}}\,(\underline{m},\, \underline{V}) = \left(\frac{\partial T^{*}}{\partial \underline{V}}\right)_{\underline{m}} = \underline{\underline{M}} \cdot \underline{V} = \int_{\Omega} \boldsymbol{p}_v\, \underline{\varphi_V}\, d\Omega \tag{9.57}$$

$$\underline{K}\,(\underline{V}) = \left(\frac{\partial T^{*}}{\partial \underline{m}}\right)_{\underline{V}} = \underline{\underline{\Omega_\rho}}^{-1} \cdot \left(\int_{\Omega} \kappa\, \underline{\varphi_\rho}\, d\Omega\right) \tag{9.58}$$

$$\underline{\Theta}\,(\underline{m},\, \underline{S}) = \left(\frac{\partial U}{\partial \underline{S}}\right)_{\underline{m}} = \underline{\underline{\Omega_S}}^{-1} \cdot \left(\int_{\Omega} \theta\, \underline{\varphi_S}\, d\Omega\right) \tag{9.59}$$

$$\underline{\Psi}\,(\underline{m},\, \underline{S}) = \left(\frac{\partial U}{\partial \underline{m}}\right)_{\underline{S}} = \underline{\underline{\Omega_\rho}}^{-1} \cdot \left(\int_{\Omega} \psi\, \underline{\varphi_\rho}\, d\Omega\right) \tag{9.60}$$

where $\underline{\boldsymbol{p}}$, $\underline{K}$, $\underline{\Theta}$ and $\underline{\Psi}$ are respectively nodal vectors of linear momentum, kinetic coenergy per unit mass, temperature, and Gibbs free energy per unit mass.

It is important to notice that Eq. (9.57) defines, in the BG terminology, a multibond modulated transformer *MTF* relating the nodal vectors of velocity and linear momentum, as shown in Fig. 9.2. According to the power conservation across the *MTF*, the generalized effort $\underline{\boldsymbol{F}}$ is given by:

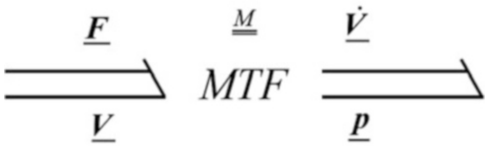
$$\underline{\boldsymbol{F}} = \underline{\underline{M}} \cdot \underline{\dot{V}} \tag{9.61}$$

Regarding the convention used in multibonds it can be observed that, in Fig. 9.2, the generalized efforts and flows are nodal vectors whose elements are vector variables. For a three-dimensional problem, it means that these type of multibonds are equivalent to $3\,n$ single bonds, being n the size of the nodal vectors involved (in this case, n_V), as shown in Fig. 9.3.

According to Eq. (9.57), the nodal vector of linear momentum can be regarded as a system volume integral of the local values weighted by the velocity interpolation function. It can be easily shown that the system linear momentum $\boldsymbol{p}$ can be obtained as:

$$\boldsymbol{p} = \int_{\Omega} \boldsymbol{p}_v\, d\Omega = \sum_{m=1}^{n_V} \boldsymbol{p}_m \tag{9.62}$$

Fig. 9.2 Modulated transformer connected to the inertial *IC*-port


$\underline{\boldsymbol{F}}$
$\underline{\underline{M}}$
$\underline{\dot{V}}$
MTF
$\underline{V}$
$\underline{\boldsymbol{p}}$

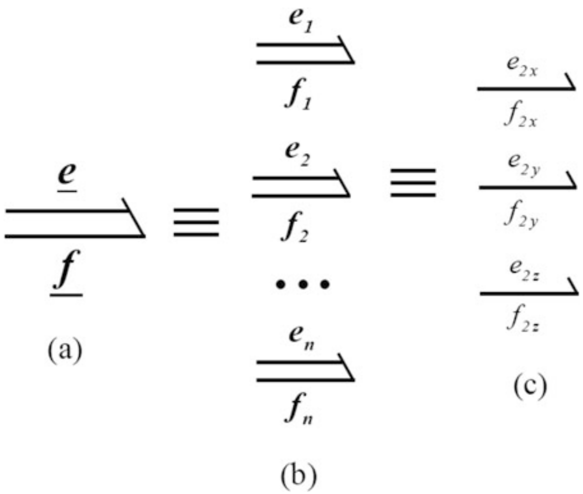


Fig. 9.3 Multibond with nodal vector of vector variables for a three-dimensional problem (**a**), equivalent to *n* multibonds of vector variables (**b**), each one of these equivalent to three single bonds (**c**)

According to Eqs. (9.58)–(9.60), the nodal vectors $\underline{K}$, $\underline{\Theta}$ and $\underline{\Psi}$ can be regarded as system volume averages of the corresponding local values, weighted by the interpolation functions. Therefore, it is important to realize that the values of the nodal vectors may be different from the corresponding values calculated with the local variables at the nodal positions.

The time derivative of the system total energy can be written as:

$$\dot{E} = (\underline{\Psi} + \underline{K})^T \cdot \dot{\underline{m}} + \underline{\boldsymbol{p}}^T \cdot \dot{\underline{V}} + \underline{\Theta}^T \cdot \dot{\underline{S}} \tag{9.63}$$

It can also be shown that the volume integrals of the left side terms of Eqs. (9.24), (9.27), and (9.28) can be calculated as:

$$\int_{\Omega} (\kappa + \psi) \frac{\partial \rho}{\partial t} \, d\Omega = (\underline{K} + \underline{\Psi})^T \cdot \dot{\underline{m}} \tag{9.64}$$

$$\int_{\Omega} \boldsymbol{p}_v \cdot \frac{\partial \boldsymbol{V}}{\partial t} \, d\Omega = \underline{\boldsymbol{p}}^T \cdot \dot{\underline{V}} \tag{9.65}$$

$$\int_{\Omega} \theta \, \frac{\partial s_v}{\partial t} \, d\Omega = \underline{\Theta}^T \cdot \dot{\underline{S}} \tag{9.66}$$

9.2.2.4 System Constitutive and Maxwell Relations

The system constitutive relations are:

$$\underline{\Psi} + \underline{K} = \underline{\Psi}\,(\underline{m}, \underline{S}) + \underline{K}\,(\underline{V}) \tag{9.67}$$

$$\underline{\boldsymbol{p}} = \underline{\boldsymbol{p}}\,(\underline{m}, \underline{V}) = \underline{\underline{M}} \cdot \underline{V} \tag{9.68}$$

$$\underline{\Theta} = \underline{\Theta}\,(\underline{m}, \underline{S}) \tag{9.69}$$

The Maxwell relations corresponding to the system total energy arise from the equality of the mixed partial derivatives of the system total energy expressed as a function of the independent variables $\underline{m}$, $\underline{\boldsymbol{V}}$, and $\underline{S}$, regarded as the state variables for the BG formalism:

$$\left(\frac{\partial \underline{\Theta}}{\partial \underline{m}}\right)_{\underline{S}} = \left[\frac{\partial}{\partial \underline{S}}\left(\underline{\Psi}+\underline{K}\right)\right]_{\underline{m}}^{T} = \left(\frac{\partial \underline{\Psi}}{\partial \underline{S}}\right)_{\underline{m}}^{T} = \underline{\underline{\Omega_{S}}}^{-1}\cdot\underline{\underline{M_{S\rho}}}\cdot\underline{\underline{\Omega_{\rho}}}^{-1} \tag{9.70}$$

$$\left(\frac{\partial \underline{\Theta}}{\partial \underline{\boldsymbol{V}}}\right)_{\underline{S}} = \left(\frac{\partial \underline{\boldsymbol{p}}}{\partial \underline{S}}\right)_{\underline{\boldsymbol{V}}}^{T} = \underline{\underline{\mathbf{0}}} \tag{9.71}$$

$$\left(\frac{\partial \underline{\boldsymbol{p}}}{\partial \underline{m}}\right)_{\underline{\boldsymbol{V}}} = \left[\frac{\partial}{\partial \underline{\boldsymbol{V}}}\left(\underline{\psi}+\underline{K}\right)\right]_{\underline{m}}^{T} = \left(\frac{\partial \underline{K}}{\partial \underline{\boldsymbol{V}}}\right)_{\underline{m}}^{T} = \underline{\underline{\boldsymbol{M}_{V\rho}}}\cdot\underline{\underline{\Omega_{\rho}}}^{-1} \tag{9.72}$$

where the matrices $\underline{\underline{M_{S\rho}}}$ (n_S rows and n_ρ columns) and $\underline{\underline{\boldsymbol{M}_{V\rho}}}$ (n_V rows and n_ρ columns) result:

$$\left\{M_{S\rho}\right\}_{lk} = \int_{\Omega}\frac{\theta}{\rho\, c_v}\left(\frac{\alpha}{\rho\,\kappa_\theta}-\frac{s_v}{\rho}\right)\varphi_{S\,l}\,\varphi_{\rho\,k}\,d\Omega \tag{9.73}$$

$$\left\{\boldsymbol{M}_{V\rho}\right\}_{mk} = \int_{\Omega}\boldsymbol{V}\,\varphi_{V\,m}\,\varphi_{\rho\,k}\,d\Omega \tag{9.74}$$

The constitutive relations (9.67)–(9.69) and the Maxwell relations (9.70)–(9.72) define a multibond IC-field associated with the system total energy, as shown in Fig. 9.4. This field has an inertial port (the velocity port) and two capacitive ports (the entropy and the mass ports). The generalized effort variables associated with these ports are respectively $\underline{\dot{\boldsymbol{V}}}$, $\underline{\Theta}$ and $(\underline{\Psi}+\underline{K})$, while the generalized flow variables are respectively $\underline{\boldsymbol{p}}$, $\underline{\dot{S}}$ and $\underline{\dot{m}}$.

In Fig. 9.4, the generalized efforts and flows of the multibonds connected to the capacitive ports are nodal vectors whose elements are scalar variables; these type of multibonds are equivalent to n (in this case, n_ρ or n_S) single bonds, as shown in Fig. 9.5.

Fig. 9.4 System IC-field representing energy storage

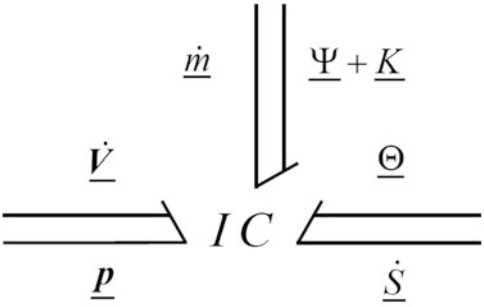
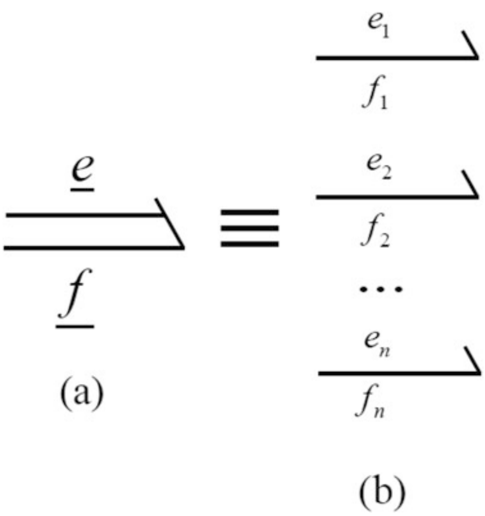

$\underline{\dot{m}}$
$\underline{\Psi}+\underline{K}$
$\underline{\dot{\boldsymbol{V}}}$
$\underline{\Theta}$
IC
$\underline{\boldsymbol{p}}$
$\underline{\dot{S}}$

Fig. 9.5 Multibond with nodal vector of scalar variables (**a**), equivalent to *n* single bonds (**b**)



9.2.3 State Equations

9.2.3.1 Mass Port

Nodal density weight functions $w_{\rho k}\,(\boldsymbol{r},\,t)$ are introduced, with the following properties:

$$\sum_{k=1}^{n_\rho} w_{\rho k}\,(\boldsymbol{r},\,t) = 1 \qquad \forall\,\boldsymbol{r}\,\in\,\Omega,\,\forall\,t \tag{9.75}$$

$$w_{\rho k}\,(\boldsymbol{r}_n,\,t) = \delta_{kn}, \text{ for a density node located at position } \boldsymbol{r}_n \tag{9.76}$$

The nodal density weight functions are introduced to satisfy the power interchanged by the system through the boundary conditions, as well as to share the importance of different power terms among neighboring nodes. These functions can be used to introduce numerical schemes, for instance controlled upwind, in compressible CFD problems.

As it is done in the Petrov–Galerkin method [47], each term of the mass balance equation (9.24) is multiplied by the test function $w_{\rho k}$; then, the resulting terms are integrated over the domain Ω and Green's theorem is applied whenever necessary, obtaining:

$$\underline{\dot{m}} = \underline{\dot{m}_W^{(\Gamma)}} + \underline{\dot{m}_W} + \underline{\dot{m}_U} + \underline{\dot{m}_K} \tag{9.77}$$

where the different nodal vectors of mass rates are:

$$\underline{\dot{m}_W^{(\Gamma)}} = -\left(\underline{\underline{\Psi}} + \underline{\underline{K}}\right)^{-1} \cdot \left[\int_\Gamma \underline{w_\rho}\, \rho\, (\psi + \kappa)\, \boldsymbol{V} \cdot \check{\boldsymbol{n}}\, d\Gamma\right] \tag{9.78}$$

$$\underline{\dot{m}_W} = \left(\underline{\underline{\Psi}} + \underline{\underline{K}}\right)^{-1} \cdot \left[\int_\Omega \rho\, (\psi + \kappa)\, \boldsymbol{V} \cdot \underline{\nabla w_\rho}\, d\Omega\right] \tag{9.79}$$

$$\underline{\dot{m}_U} = \left(\underline{\underline{\Psi}} + \underline{\underline{K}}\right)^{-1} \cdot \left(\int_\Omega \underline{w_\rho}\, \rho\, \boldsymbol{V} \cdot \nabla\psi\, d\Omega\right) \tag{9.80}$$

$$\underline{\dot{m}_K} = \left(\underline{\underline{\Psi}} + \underline{\underline{K}}\right)^{-1} \cdot \left(\int_\Omega \underline{w_\rho}\, \rho\, \boldsymbol{V} \cdot \nabla\kappa\, d\Omega\right) \tag{9.81}$$

The square matrices $\underline{\underline{\Psi}}$ and $\underline{\underline{K}}$ (size n_ρ) are defined as:

$$\{\Psi\}_{kj} = \frac{1}{\Omega_{\rho j}} \int_\Omega \psi\, w_{\rho k}\, \varphi_{\rho j}\, d\Omega \tag{9.82}$$

$$\{K\}_{kj} = \frac{1}{\Omega_{\rho j}} \int_\Omega \kappa\, w_{\rho k}\, \varphi_{\rho j}\, d\Omega \tag{9.83}$$

The nodal vectors $\underline{\Psi}$ and $\underline{K}$ are the sum of the corresponding matrix columns:

$$\Psi_j = \textstyle\sum_{k=1}^{n_\rho} \{\Psi\}_{kj} \tag{9.84}$$

$$K_j = \textstyle\sum_{k=1}^{n_\rho} \{K\}_{kj} \tag{9.85}$$

Taking into account Eqs. (9.84) and (9.85) it can be verified that the product $(\underline{\Psi} + \underline{K})^T \cdot \underline{\dot{m}_X}$, where $\underline{\dot{m}_X}$ is any nodal vector of mass rate, recovers the corresponding power term integrated in the system. So, the product $(\underline{\Psi} + \underline{K})^T \cdot \underline{\dot{m}_W^{(\Gamma)}}$ recovers the power due to the flux of Gibbs free energy plus kinetic coenergy through the system boundary Γ, while $(\underline{\Psi} + \underline{K})^T \cdot \underline{\dot{m}_W}$ is a power term that vanishes, because of Eq. (9.75). Notice that $\dot{m}_{Wk}^{(\Gamma)}$ may be nonzero only for the nodes located at the system boundary. Making the product of $\underline{\Psi} + \underline{K}$ times Eq. (9.77), it can be easily shown that the integral mass balance equation is satisfied, that is:

$$\int_\Omega (\psi + \kappa)\, \frac{\partial\rho}{\partial t}\, d\Omega = -\int_\Gamma \rho\, (\psi + \kappa)\, \boldsymbol{V} \cdot \check{\boldsymbol{n}}\, d\Gamma + \int_\Omega \rho\, \boldsymbol{V} \cdot \nabla\psi\, d\Omega + \int_\Omega \rho\, \boldsymbol{V} \cdot \nabla\kappa\, d\Omega \tag{9.86}$$

9.2.3.2 Velocity Port

As it is done in the Galerkin method [47], the momentum conservation equation is multiplied by the test function φ_{Vm} and integrated over the domain Ω. Applying Green's theorem whenever necessary, it can be obtained:

$$\underline{\dot{\boldsymbol{V}}} = \underline{\underline{M}}^{-1} \cdot \left(\underline{\boldsymbol{F}_V^{(\Gamma)}} + \underline{\boldsymbol{F}_G} + \underline{\boldsymbol{F}_R} - \underline{\boldsymbol{F}_P} - \underline{\boldsymbol{F}_V} - \underline{\boldsymbol{F}_K} \right) \tag{9.87}$$

where the different nodal vectors of forces are:

$$\underline{\boldsymbol{F}_V^{(\Gamma)}} = \int_\Gamma \left(\underline{\underline{\tau}} \cdot \check{\boldsymbol{n}} \right) \underline{\varphi_V} \, d\Gamma \tag{9.88}$$

$$\underline{\boldsymbol{F}_G} = \int_\Omega \rho \, \boldsymbol{G} \, \underline{\varphi_V} \, d\Omega \tag{9.89}$$

$$\underline{\boldsymbol{F}_R} = \int_\Omega \rho \, \boldsymbol{V} \times (\nabla \times \boldsymbol{V}) \, \underline{\varphi_V} \, d\Omega \tag{9.90}$$

$$\underline{\boldsymbol{F}_P} = \int_\Omega \nabla P \, \underline{\varphi_V} \, d\Omega \tag{9.91}$$

$$\underline{\boldsymbol{F}_V} = \int_\Omega \underline{\underline{\tau}} \cdot \underline{\nabla \varphi_V} \, d\Omega \tag{9.92}$$

$$\underline{\boldsymbol{F}_K} = \int_\Omega \rho \, \nabla \kappa \, \underline{\varphi_V} \, d\Omega \tag{9.93}$$

Adding the nodal components of Eq. (9.87) it can be easily shown that the integral momentum equation is satisfied, that is:

$$\int_\Omega \rho \, \frac{D\boldsymbol{V}}{Dt} \, d\Omega = \int_\Gamma \left(-P \, \underline{\underline{I}} + \underline{\underline{\tau}} \right) \cdot \check{\boldsymbol{n}} \, d\Gamma + \int_\Omega \rho \, \boldsymbol{G} \, d\Omega \tag{9.94}$$

The reason why the Galerkin method was chosen is that the dot product $\underline{\boldsymbol{F}_X}^T \cdot \underline{\boldsymbol{V}}$, where $\underline{\boldsymbol{F}_X}$ is any nodal vector of force, recovers the corresponding power term integrated in the system. It can be observed that the dot product $\underline{\boldsymbol{F}_V^{(\Gamma)}}^T \cdot \underline{\boldsymbol{V}}$ recovers the power due to the flux of the viscous stress through the system boundary, while the dot product $\underline{\boldsymbol{F}_V}^T \cdot \underline{\boldsymbol{V}}$ recovers the power dissipation. Notice that $\boldsymbol{F}_{V\,m}^{(\Gamma)}$ may be nonzero only for the nodes located at the system boundary.

Making the dot product of Eq. (9.87) and $\underline{\boldsymbol{V}}$ and taking into account Eq. (9.25), it can be easily shown that the integral velocity balance equation is satisfied, that is:

$$\begin{aligned} \int_\Omega \boldsymbol{p}_v \cdot \frac{\partial \boldsymbol{V}}{\partial t} \, d\Omega = {} & \int_\Gamma \left(\underline{\underline{\tau}} \cdot \boldsymbol{V} \right) \cdot \check{\boldsymbol{n}} \, d\Gamma + \int_\Omega \rho \, \boldsymbol{G} \cdot \boldsymbol{V} \, d\Omega - \int_\Omega \boldsymbol{V} \cdot \nabla P \, d\Omega \\ & - \int_\Omega \left(\nabla \boldsymbol{V} : \underline{\underline{\tau}} \right) d\Omega - \int_\Omega \rho \, \boldsymbol{V} \cdot \nabla \kappa \, d\Omega \end{aligned} \tag{9.95}$$

9.2.3.3 Entropy Port

Nodal entropy weight functions $w_{S\,l}(\boldsymbol{r},\, t)$ are also introduced, with the following properties:

$$\sum_{l=1}^{n_S} w_{S\,l}(\boldsymbol{r},\, t) = 1 \qquad\qquad \forall\, \boldsymbol{r} \in \mathit{\Omega},\, \forall\, t \tag{9.96}$$

$$w_{S\,l}(\boldsymbol{r}_n,\, t) = \delta_{l\,n}, \text{ for an entropy node located at position } \boldsymbol{r}_n \tag{9.97}$$

The nodal entropy weight functions are also introduced to satisfy the power interchanged by the system through the boundary conditions, as well as to share the importance of different power terms among neighboring nodes. These functions can be used, for instance, to introduce controlled upwind schemes in advection-diffusion problems.

As it is done in the Petrov–Galerkin method [47], each term of the entropy balance equation (9.28) is multiplied by the test function $w_{S\,l}$; then, the resulting terms are integrated over the domain $\mathit{\Omega}$ and Gauss' theorem is applied whenever necessary, obtaining:

$$\underline{\dot{S}} = \underline{\dot{S}_Q^{(\Gamma)}} + \underline{\dot{S}_C^{(\Gamma)}} + \underline{\dot{S}_Q} + \underline{\dot{S}_C} + \underline{\dot{S}_F} - \underline{\dot{S}_U} + \underline{\dot{S}_P} + \underline{\dot{S}_V} \tag{9.98}$$

where the different nodal vectors of entropy rates are:

$$\underline{\dot{S}_Q^{(\Gamma)}} = -\underline{\underline{\mathit{\Theta}}}^{-1} \cdot \left(\int_{\Gamma} \underline{w_S}\, \boldsymbol{q} \cdot \check{\boldsymbol{n}}\, d\Gamma \right) \tag{9.99}$$

$$\underline{\dot{S}_C^{(\Gamma)}} = -\underline{\underline{\mathit{\Theta}}}^{-1} \cdot \left(\int_{\Gamma} \underline{w_S}\, \theta\, s_v\, \boldsymbol{V} \cdot \check{\boldsymbol{n}}\, d\Gamma \right) \tag{9.100}$$

$$\underline{\dot{S}_Q} = \underline{\underline{\mathit{\Theta}}}^{-1} \cdot \left(\int_{\mathit{\Omega}} \boldsymbol{q} \cdot \underline{\nabla w_S}\, d\mathit{\Omega} \right) \tag{9.101}$$

$$\underline{\dot{S}_C} = \underline{\underline{\mathit{\Theta}}}^{-1} \cdot \left(\int_{\mathit{\Omega}} \theta\, s_v\, \boldsymbol{V} \cdot \underline{\nabla w_S}\, d\mathit{\Omega} \right) \tag{9.102}$$

$$\underline{\dot{S}_F} = \underline{\underline{\mathit{\Theta}}}^{-1} \cdot \left(\int_{\mathit{\Omega}} \underline{w_S}\, \rho\, \Phi\, d\mathit{\Omega} \right) \tag{9.103}$$

$$\underline{\dot{S}_U} = \underline{\underline{\mathit{\Theta}}}^{-1} \cdot \left(\int_{\mathit{\Omega}} \underline{w_S}\, \rho\, \boldsymbol{V} \cdot \nabla \psi\, d\mathit{\Omega} \right) \tag{9.104}$$

$$\underline{\dot{S}_P} = \underline{\underline{\mathit{\Theta}}}^{-1} \cdot \left(\int_{\mathit{\Omega}} \underline{w_S}\, \boldsymbol{V} \cdot \nabla P\, d\mathit{\Omega} \right) \tag{9.105}$$

$$\underline{\dot{S}_V} = \underline{\underline{\mathit{\Theta}}}^{-1} \cdot \left[\int_{\mathit{\Omega}} \underline{w_S} \left(\nabla \boldsymbol{V} : \underline{\underline{\tau}} \right) d\mathit{\Omega} \right] \tag{9.106}$$

In Eqs. (9.99)–(9.106) the temperature matrix $\underline{\underline{\Theta}}$ results:

$$\{\Theta\}_{lj} = \frac{1}{\Omega_{Sj}} \int_{\Omega} \theta \, w_{S\,l} \, \varphi_{Sj} \, d\Omega \tag{9.107}$$

The nodal vector of temperature is the sum of the temperature matrix columns:

$$\Theta_j = \sum_{l=1}^{ns} \{\Theta\}_{lj} \tag{9.108}$$

Taking into account Eq. (9.108) it can be verified that the product $\underline{\Theta}^T \cdot \underline{\dot{S}_X}$, where $\underline{\dot{S}_X}$ is any nodal vector of entropy rate, recovers the corresponding power integrated in the system. Thus, the products $\underline{\Theta}^T \cdot \underline{\dot{S}_Q^{(\Gamma)}}$ and $\underline{\Theta}^T \cdot \underline{\dot{S}_C^{(\Gamma)}}$ recovers respectively the power due to heat flux and entropy advection, while $\underline{\Theta}^T \cdot \underline{\dot{S}_Q}$ and $\underline{\Theta}^T \cdot \underline{\dot{S}_C}$ are power terms that vanish, because of Eq. (9.96). Notice that $\dot{S}_{Q\,l}^{(\Gamma)}$ and $\dot{S}_{C\,l}^{(\Gamma)}$ may be nonzero only for the nodes located at the system boundary. Making the product of Eq. (9.98) times $\underline{\Theta}$, it can be easily shown that the integral entropy balance equation is satisfied, that is:

$$\begin{aligned}\int_{\Omega} \theta \, \frac{\partial s_v}{\partial t} \, d\Omega = &- \int_{\Gamma} (\boldsymbol{q} + \theta \, s_v \, \boldsymbol{V}) \cdot \check{\boldsymbol{n}} \, d\Gamma + \int_{\Omega} \rho \, \Phi \, d\Omega - \int_{\Omega} \rho \, \boldsymbol{V} \cdot \nabla \psi \, d\Omega \\ &+ \int_{\Omega} \boldsymbol{V} \cdot \nabla P \, d\Omega + \int_{\Omega} \left(\nabla \boldsymbol{V} : \underline{\underline{\tau}} \right) d\Omega \end{aligned} \tag{9.109}$$

9.2.4 Coupling Matrices

Once defined the generalized effort and flow variables, it is necessary to represent the power coupling (appearing in the balance equations per unit volume shown in Sect. 9.2.1.4) to a discretized level. This representation is performed through the coupling matrices, which relate generalized variables whose product gives rise to power terms appearing in pairs of multiports. Depending on the variables being related, it will be seen that these matrices define modulated transformers *MTF*, a modulated resistance-entropy source field *MRS* or a modulated gyrator *MGY*. As the nodal vectors may have different size, the coupling matrices are rectangular, thus setting a restriction in the allowable causalities.

9.2.4.1 Coupling Between the Velocity and Mass Ports

From Eqs. (9.81) and (9.93):

$$\underline{\dot{m}_K} = \left[\left(\underline{\underline{\Psi}} + \underline{\underline{K}} \right)^{-1} \cdot \underline{\underline{\boldsymbol{M}_K}} \right] \cdot \underline{\boldsymbol{V}} \tag{9.110}$$

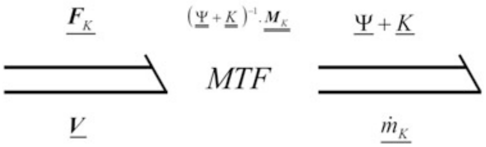

$\underline{F_K}$
$(\underline{\Psi}+\underline{K})^{-1}.\underline{M_K}$
$\underline{\Psi}+\underline{K}$
MTF
$\underline{V}$
$\underline{\dot{m}_K}$

Fig. 9.6 Modulated transformer coupling the velocity and mass ports

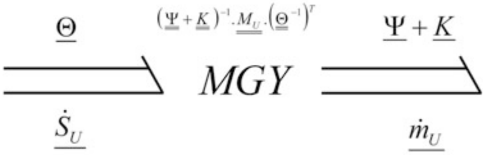

$\underline{\Theta}$
$(\underline{\Psi}+\underline{K})^{-1}.\underline{M_U}.(\underline{\Theta}^{-1})^T$
$\underline{\Psi}+\underline{K}$
MGY
$\underline{\dot{S}_U}$
$\underline{\dot{m}_U}$

Fig. 9.7 Modulated gyrator coupling the entropy and mass ports

$$\underline{\boldsymbol{F}_K} = \left[\left(\underline{\underline{\Psi}} + \underline{\underline{K}} \right)^{-1} \cdot \underline{\underline{\boldsymbol{M}_K}} \right]^T \cdot (\underline{\Psi} + \underline{K}) \tag{9.111}$$

where $\underline{\underline{\boldsymbol{M}_K}}$ is a rectangular matrix (n_ρ rows and n_V columns) defined as:

$$\{\boldsymbol{M}_K\}_{k\,m} = \int_\Omega w_{\rho\,k}\,\varphi_{V\,m}\,\rho\,\nabla\kappa\,d\Omega \tag{9.112}$$

Equations (9.110) and (9.111) define a multibond *MTF* modulated by the state variables, as shown in Fig. 9.6.

9.2.4.2 Coupling Between the Entropy and Mass Ports

From Eqs. (9.80) and (9.104):

$$\underline{\dot{m}_U} = \left[\left(\underline{\underline{\Psi}} + \underline{\underline{K}} \right)^{-1} \cdot \underline{\underline{M_U}} \cdot \left(\underline{\underline{\Theta}}^{-1} \right)^T \right] \cdot \underline{\Theta} \tag{9.113}$$

$$\underline{\dot{S}_U} = \left[\left(\underline{\underline{\Psi}} + \underline{\underline{K}} \right)^{-1} \cdot \underline{\underline{M_U}} \cdot \left(\underline{\underline{\Theta}}^{-1} \right)^T \right]^T \cdot (\underline{\Psi} + \underline{K}) \tag{9.114}$$

where $\underline{\underline{M_U}}$ is a rectangular matrix (n_ρ rows and n_S columns) defined as:

$$\{M_U\}_{k\,l} = \int_\Omega w_{\rho\,k}\,w_{S\,l}\,\rho\,\boldsymbol{V}\cdot\nabla\psi\,d\Omega \tag{9.115}$$

Equations (9.113) and (9.114) define a multibond *MGY* modulated by the state variables, as shown in Fig. 9.7.

9.2.4.3 Coupling Between the Velocity and Entropy Ports

From Eqs. (9.91) and (9.105):

$$\underline{\boldsymbol{F}_P} = \left(\underline{\underline{\Theta}}^{-1} \cdot \underline{\underline{\boldsymbol{M}_P}}\right)^T \cdot \underline{\Theta} \tag{9.116}$$

$$\underline{\dot{S}_P} = \left(\underline{\underline{\Theta}}^{-1} \cdot \underline{\underline{\boldsymbol{M}_P}}\right) \cdot \underline{\boldsymbol{V}} \tag{9.117}$$

where $\underline{\underline{\boldsymbol{M}_P}}$ is a rectangular matrices (n_V rows and n_S columns) defined as:

$$\{\boldsymbol{M}_P\}_{m\,l} = \int_\Omega \varphi_{V\,m}\, w_{S\,l}\, \nabla P\, d\Omega \tag{9.118}$$

Equations (9.116) and (9.117) define a multibond *MTF* modulated by the state variables, as shown in Fig. 9.8.

From Eqs. (9.92) and (9.106):

$$\underline{\boldsymbol{F}_V} = \left(\underline{\underline{\Theta}}^{-1} \cdot \underline{\underline{\boldsymbol{M}_V}}\right)^T \cdot \underline{\Theta} \tag{9.119}$$

$$\underline{\dot{S}_V} = \left(\underline{\underline{\Theta}}^{-1} \cdot \underline{\underline{\boldsymbol{M}_V}}\right) \cdot \underline{\boldsymbol{V}} \tag{9.120}$$

where $\underline{\underline{\boldsymbol{M}_V}}$ is a rectangular matrices (n_V rows and n_S columns) defined as:

$$\{\boldsymbol{M}_V\}_{m\,l} = \int_\Omega \left(\underline{\underline{\tau}} \cdot \nabla\varphi_{V\,m}\right) w_{S\,l}\, d\Omega \tag{9.121}$$

Equations (9.119) and (9.120) define a multibond *MRS* modulated by the state variables, as shown in Fig. 9.9.

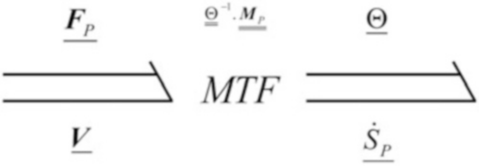


Fig. 9.8 Modulated transformer coupling the velocity and entropy ports

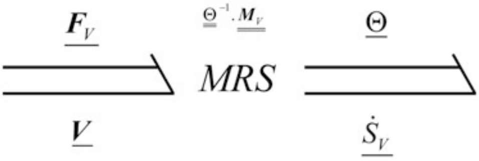


Fig. 9.9 Modulated resistance-entropy source field coupling the velocity and entropy ports

9.2.5 System BG

The system BG is shown in Fig. 9.10. Energy storing (kinetic and potential) is represented by an IC-field. A MTF with the inertia matrix $\underline{\underline{M}}$ is connected to the inertial port of the IC-field, in order to bring the nodal velocities as generalized flow variables.

At the 1-junction with common $\underline{\boldsymbol{V}}$ all the nodal vector forces are added; in this way, the effort balance represents the linear momentum conservation equation for the nodal velocity values. At the 0-junction with common $(\underline{\Psi} + \underline{K})$ all the nodal mass rates are added; in this way, the flow balance represents the mass conservation equations for the nodal mass values. At the 0-junction with common $\underline{\Theta}$ all the nodal entropy rates are added; in this way, the flow balance represents the entropy conservation equation for the nodal entropy values.

The MTFs, the MRS, and the MGY between the junction elements connect power terms that appear in the balance equations corresponding to pairs of multiports. As seen in Sect. 9.2.4, their coupling matrices are rectangular, setting a restriction in the allowable causalities.

The acausal source elements $\boldsymbol{S}$ connected to the bonds with $\underline{\dot{m}_W^{(\Gamma)}}$, $\underline{\boldsymbol{F}_T^{(\Gamma)}}$, $\underline{\dot{S}_Q^{(\Gamma)}}$ and $\underline{\dot{S}_C^{(\Gamma)}}$ represent different source terms; in each single port these sources behave as effort or flow sources, depending on the boundary conditions. The rest of the sources, effort $\boldsymbol{S}_e$ or flow $\boldsymbol{S}_f$ (the ones connected to the bonds with $\underline{\dot{m}_W}$, $\underline{\boldsymbol{F}_R} + \underline{\boldsymbol{F}_G}$ and $\underline{\dot{S}_Q} + \underline{\dot{S}_C} + \underline{\dot{S}_F}$) represent volumetric power terms; the determination of causality for these sources and for other bonds shown in the graph results from the standard causality extension procedure described in [32].

The net power input (sum over the bonds) corresponding to the multibond with the rotational force $\underline{\boldsymbol{F}_R}$ is zero, because of Eq. (9.25). As seen before, the contributions of $\underline{\dot{S}_Q}$ and $\underline{\dot{S}_C}$ to the net power input are also zero.

9.2.6 Initial and Boundary Conditions

9.2.6.1 Initial Conditions

In order to solve the state equations, it is needed to set initial and boundary conditions. The nodal initial values may be readily specified as:

$$\underline{m}\,(t = 0) = \underline{m_0} \tag{9.122}$$

$$\underline{\boldsymbol{V}}\,(t = 0) = \underline{\boldsymbol{V}_0} \tag{9.123}$$

$$\underline{S}\,(t = 0) = \underline{S_0} \tag{9.124}$$

Alternatively, if spatial functions $\rho_0\,(\boldsymbol{r})$, $\boldsymbol{V}_0\,(\boldsymbol{r})$, and $s_{v\,0}\,(\boldsymbol{r})$ are specified for the initial time respectively for density, velocity, and entropy per unit volume, the nodal values must be determined in order to conserve the initial system mass, linear momentum, and entropy. In this case, it can be easily shown that:

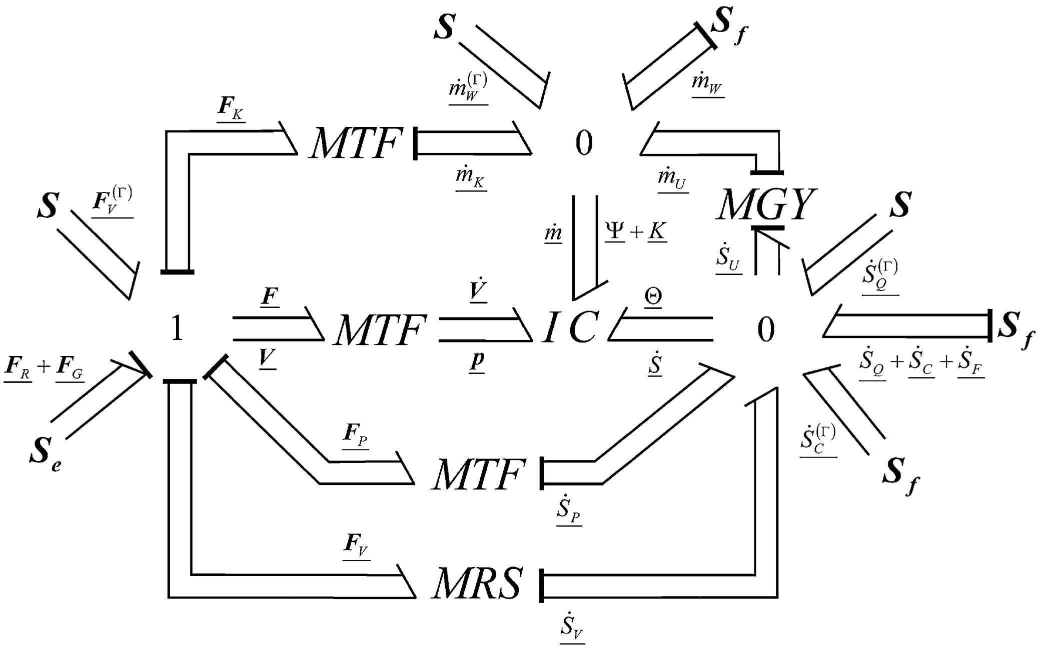


Fig. 9.10 System BG for a compressible flow

$$m_{0k} = \int_{\Omega} \rho_0\left(\boldsymbol{r}\right) \varphi_{\rho k}\, d\Omega \tag{9.125}$$

$$\boldsymbol{V}_{0m} = \frac{\int_{\Omega} \rho_0\left(\boldsymbol{r}\right) \boldsymbol{V}_0\left(\boldsymbol{r}\right) \varphi_{Vm}\, d\Omega}{\int_{\Omega} \rho_0\left(\boldsymbol{r}\right) \varphi_{Vm}\, d\Omega} \tag{9.126}$$

$$S_{0l} = \int_{\Omega} s_{v\,0}\left(\boldsymbol{r}\right) \varphi_{Sl}\, d\Omega \tag{9.127}$$

9.2.6.2 Boundary Conditions

The boundary conditions establish relationships among the variables at the boundary Γ. At a discretized level, the boundary conditions can be regarded, in the BG methodology, as the input variables. It is necessary, for the model being mathematically well defined, that the boundary conditions allow to determine the causality for all the bonds in the resulting BG. The boundary conditions are introduced through the bonds corresponding to the superficial source terms $\underline{\dot{m}_W^{(\Gamma)}}$, $\underline{\boldsymbol{F}_T^{(\Gamma)}}$, $\underline{\dot{S}_Q^{(\Gamma)}}$, and $\underline{\dot{S}_C^{(\Gamma)}}$. All boundary conditions can be described in terms of generalized effort or flow sources.

In a general boundary value problem, boundary conditions can be classified in essential, natural, and mixed.

In an essential boundary condition (often called Dirichlet boundary condition), independent variables are specified at a part of the boundary. Typical examples in CFD can be problems in which the flow conditions (velocity and/or thermophysical properties) are specified at the inlet or outlet of a flow passage section or when normal and tangential components are specified at a wall satisfying the no-normal or no-slip condition. For these cases, a derivative causality will result for some boundary bonds belonging to the *IC*-field.

Natural boundary conditions (often called Neumann boundary conditions) are those that are automatically satisfied after solution of the problem. These boundary conditions appear when gradients of the independent variables are specified at a part of the boundary. In a CFD problem, natural boundary conditions are associated with imposed components of the viscous stress tensor or heat flux at the boundary. For these cases, integral causality will result for the boundary nodes belonging to the *IC*-field.

Finally, mixed boundary conditions are those that involve gradients as well as values of the independent variables at a part of the boundary. For these cases, integral causality will also result for the boundary nodes belonging to the *IC*-field. Newton's cooling law, relating the normal component of the heat flux to the difference between the boundary and a reference temperature, is a typical example.

9.2.6.3 Considerations for Pure Advective Flows

If heat conduction effects are neglected in the conservation equation (9.23), the multibonds corresponding to $\underline{\dot{S}_Q^{(\Gamma)}}$ and $\underline{\dot{S}_Q}$ disappear. This approximation makes the entropy balance equation first order and has consequences in the causality assignment. In particular, for pure advective flows only thermophysical properties at parts of the boundary can be specified. In this case, the entropy boundary conditions are introduced through the bonds corresponding to the superficial source term $\underline{\dot{S}_C^{(\Gamma)}}$.

9.2.6.4 Considerations for Inviscid Flows

If the viscous stress tensor is neglected in the conservation equation (9.22), the multibond corresponding to $\underline{\boldsymbol{F}_T^{(\Gamma)}}$ disappears. This approximation makes the momentum balance equation first order and has consequences in the causality assignment. For instance, only the normal component of the velocity field can be specified at a wall, while the other components are part of the problem solution.

9.2.6.5 Procedure for Causality Assignment

In [32] a sequential causal assignment procedure is described. Sources are chosen first, the required causality is assigned, and the causal implications are extended through the graph as far as possible, using the constraint elements (in this case 0-junctions, 1-junctions, *MTF*s, *MRS* and *MGY*). Then, the ports corresponding to the storage elements (in this case, the *IC*-field) are chosen, integral causality is assigned, and, again, the causal implications are extended through the graph as far as possible.

In a grid generated for a CFD problem usually there is a huge amount of nodes. The domain is divided in cells, the volume integrations are performed locally in each cell, and the results are assembled to build the system integrals. For doing this, it is necessary to know what is called the connectivity information, that is, the information needed to completely identify each cell and all of the neighbors of that cell in a computational grid.

As a result of the calculation procedure, the resulting matrices are sparse, that is, have a few nonzero elements. Regarding causality extension through the *MTF*s, *MRS*, and *MGY* elements, the constitutive laws are sets of linear relationships among the variables involved. Thus, causality can be extended for a bond with a variable only when the bonds corresponding to the rest of the variables in the linear relationship have assigned causalities.

It is worth noting that, as the interpolation functions are zero at the boundary Γ for inner nodes, causality is assigned by definition at these bonds. Thus, a zero-effort source is connected to an inner $\boldsymbol{F}_{V\,m}^{(\Gamma)}$, and respectively zero-flow sources are connected to an inner $\dot{m}_{W\,k}^{(\Gamma)}$, $\dot{S}_{Q\,l}^{(\Gamma)}$ and $\dot{S}_{C\,l}^{(\Gamma)}$.

As there are only energy storing and conserving elements, all bonds should have causality assigned after the procedure detailed above. The order of the system is the number of bonds connected to the energy storing elements, resulting with integral causality. This causality procedure can be implemented automatically, knowing the connectivity of the computational grid, as a way of checking the correctness of the boundary conditions in the problem.

As a consequence of the causality extension, the sources connected to the bonds with $\underline{\boldsymbol{F}_G} + \underline{\boldsymbol{F}_R}$ always behave as effort sources, and the sources connected to the bonds with $\underline{\dot{S}_Q} + \underline{\dot{S}_C} + \underline{\dot{S}_F}$, $\underline{\dot{S}_C^{(\Gamma)}}$, and $\underline{\dot{m}_W}$ always behave as flow sources. Besides, causality for the bonds connected to the coupling *MTF*s, *MRS*, and *MGY* is also defined. For the *MTF*s and the *MRS*, it always results $\underline{\boldsymbol{V}}$ the input and respectively $\underline{\dot{m}_K}$, $\underline{\dot{S}_P}$ and $\underline{\dot{S}_V}$ outputs. For the *MGY*, it always results $\underline{\Theta}$ the input and $\underline{\dot{m}_U}$ the output.

The resulting causality for the coupling *MTF*s and *MRS* indicate that fluid motion generates the entropy rate (both reversible and irreversible) and mass rates, in agreement with the Second Principle of Thermodynamics. Besides, $\underline{\Theta}$ and $\underline{\Psi} + \underline{K}$ always result inputs to the coupling *MTF*s and *MRS*, indicating that thermophysical properties influence the corresponding output forces.

9.3 Incompressible Flows

An interesting type of problems are those in which the fluid is incompressible, that is, density is constant.

When viscosity variations with temperature are small, the traditional incompressible form of the Navier–Stokes (N–S) equation is usually selected for the analysis. A set of equations (continuity, momentum, and thermal energy) results with three unknowns, for which usually velocity, pressure, and temperature are chosen; this is known as the *primitive-variable approach*. Other alternatives have a limitation to bi-dimensional flows (vorticity-stream function approach), or are less attractive for three-dimensional flows (vector potential approach); consequently, the N–S equations are often solved in their primitive variable form.

For constant viscosity, the energy equation can be uncoupled, so the temperature field can be obtained after the velocity field has been computed. As the nonlinearities are related to the convective term, the attention is focused on the solution of the continuity and momentum equations.

An important characteristic of the N–S equations is that no time derivative of pressure appears. Pressure is no longer a thermophysical property, but a function that must act in such a way that the resulting velocity field has divergence zero. In an incompressible flow pressure perturbations propagate at infinite speed, obeying an elliptic, Poisson's type partial differential equation, where the source term is a function of the velocity field.

A strategy often employed for the numerical solution of the incompressible N–S equations is the *pressure correction approach*, in which a derived equation is used to determine the pressure field. Typically, the momentum equations are solved for the velocity components using linearized expressions in which time-lagged values are used for the variables other than the unknown, including pressure. In this step, the obtained velocity field does not satisfy the continuity equation. Next, the solution is substituted in the discretized continuity equation and often a Poisson equation is developed for the pressure (or pressure changes), from which a new pressure field is obtained. This pressure field is used to calculate a new velocity field until a solution is produced that satisfies both the momentum and the continuity equations. The literature on numerical schemes using the pressure correction approach is extensive, differing the methods in the algorithms used to solve the component equations and the improved pressure field. A review of these methods can be found in [42].

In this section a single-phase, single-component, multidimensional incompressible flow with viscosity and thermal effects is considered for the application of the BG methodology [5, 7]. Main characteristics of the methodology explained in Sect. 9.2 will not be repeated here; only the new concepts will be stressed.

9.3.1 Continuum Formulation

9.3.1.1 Energy and Power Representation

For an incompressible fluid, the density ρ_0 is no longer a state variable. The total energy per unit volume e_v can be written as:

$$e_v = u_v\,(s_v) + t_v^*\,(\boldsymbol{V}) \tag{9.128}$$

For a single-phase, single-component fluid with constant specific heat c_v, the kinetic coenergy and internal energy per unit volume can be written as:

$$t_v^* = \frac{1}{2}\,\rho_0\,\boldsymbol{V}^2 = \rho_0\,\kappa \tag{9.129}$$

$$u_v = u_{v\,R} + \rho_0\,c_v\,\theta_R\left[\exp\left(\frac{s_v}{\rho_0\,c_v}\right) - 1\right] \tag{9.130}$$

where $u_{v\,R}$ and θ_R are respectively reference values of internal energy per unit volume and absolute temperature, for which the entropy per unit volume is zero. The linear momentum per unit volume and the absolute temperature are defined as:

$$\boldsymbol{p}_v = \frac{dt_v^*}{d\boldsymbol{V}} = \rho_0\,\boldsymbol{V} \tag{9.131}$$

$$\theta = \frac{du_v}{ds_v} = \theta_R\,\exp\left(\frac{s_v}{\rho_0\,c_v}\right) \tag{9.132}$$

The time derivative of Eq. (9.128) can be written as:

$$\frac{\partial e_v}{\partial t} = \boldsymbol{p}_v \cdot \frac{\partial \boldsymbol{V}}{\partial t} + \theta\,\frac{\partial s_v}{\partial t} \tag{9.133}$$

As the internal energy per unit volume is only a function of the entropy per unit volume and the kinetic coenergy per unit volume is only a function of the velocity, these two energy domains can be decoupled.

9.3.1.2 Conservation Equations

For an incompressible fluid the continuity, linear momentum, and thermal energy conservation equations are [44]:

$$\nabla \cdot \boldsymbol{V} = 0 \tag{9.134}$$

$$\rho_0 \, \frac{\partial \boldsymbol{V}}{\partial t} = -\rho_0 \, \nabla \kappa + \rho_0 \, \boldsymbol{V} \times (\nabla \times \boldsymbol{V}) - \nabla P + \rho_0 \, \boldsymbol{G} + \nabla \cdot \underline{\underline{\tau}} \tag{9.135}$$

$$\frac{\partial u_v}{\partial t} = -\nabla \cdot \boldsymbol{q} - \nabla u_v \cdot \boldsymbol{V} + \nabla \boldsymbol{V} : \underline{\underline{\tau}} + \rho_0 \, \Phi \tag{9.136}$$

For a newtonian, incompressible fluid and assuming Fourier's law, the viscous stress and the heat flux can be written as:

$$\underline{\underline{\tau}} = \mu \, \left(\nabla \boldsymbol{V} + \nabla \boldsymbol{V}^T \right) \tag{9.137}$$

$$\boldsymbol{q} = -\lambda \, \nabla \theta = -\frac{\lambda \, \theta}{\rho_0 \, c_v} \, \nabla s_v \tag{9.138}$$

9.3.1.3 Balance Equations

Transforming the conservation equations in the same fashion as it was made in Sect. 9.2.1.4, it can be obtained:

$$\boldsymbol{p}_v \cdot \frac{\partial \boldsymbol{V}}{\partial t} = \nabla \cdot \left(\underline{\underline{\tau}} \cdot \boldsymbol{V} \right) - \nabla \cdot (P \, \boldsymbol{V}) - \rho_0 \, \nabla \kappa \cdot \boldsymbol{V} + \rho_0 \, \boldsymbol{G} \cdot \boldsymbol{V} - \nabla \boldsymbol{V} : \underline{\underline{\tau}} \tag{9.139}$$

$$\theta \, \frac{\partial s_v}{\partial t} = -\nabla \cdot \boldsymbol{q} - \nabla \cdot (\theta \, s_v \, \boldsymbol{V}) + s_v \, \nabla \theta \cdot \boldsymbol{V} + \rho_0 \, \Phi + \nabla \boldsymbol{V} : \underline{\underline{\tau}} \tag{9.140}$$

The only coupling term $\nabla \boldsymbol{V} : \underline{\underline{\boldsymbol{\tau}}}$ represents the power transfer (mechanical energy dissipation) between the velocity and entropy equations; this coupling term appears, with opposite signs, in the balance equations. Adding the balance equations, it can be easily obtained the conservation of total energy:

$$\rho_0 \, \frac{D\hat{e}}{Dt} = \nabla \cdot \left[\left(-P \, \underline{\underline{I}} + \underline{\underline{\tau}} \right) \cdot \boldsymbol{V} - \boldsymbol{q} \right] + \rho_0 \, \boldsymbol{G} \cdot \boldsymbol{V} + \rho_0 \, \Phi \tag{9.141}$$

where $\hat{e} = \hat{u} + \frac{1}{2} \, \boldsymbol{V}^2$ is the total energy per unit mass and $\frac{D}{Dt}$ is the material derivative.

9.3.2 Discrete Formulation

9.3.2.1 Description of the Flow Fields

The description of the flow fields corresponding to the velocity and entropy per unit volume is made as in Sect. 9.2.2.1. The description of the pressure field is made as:

$$P(\boldsymbol{r}, t) = \sum_{k=1}^{n_P} P_k(t)\, \varphi_{P\,k}(\boldsymbol{r}) = \underline{P}^T \cdot \underline{\varphi_P} \tag{9.142}$$

where $\underline{P}$ (size n_P) is the time-dependent nodal pressure vector and $\underline{\varphi_P}$ is the corresponding nodal shape function.

9.3.2.2 System Total Energy

Following the same procedure as in Sect. 9.2.2.3, the system total energy E results:

$$E = U(\underline{S}) + T^*(\underline{V}) \tag{9.143}$$

The system kinetic coenergy can be expressed as the following bilinear form:

$$T^* = \frac{1}{2}\,\underline{\boldsymbol{V}}^T \cdot \underline{\underline{M}} \cdot \underline{\boldsymbol{V}} \tag{9.144}$$

where $\underline{\underline{M}}$ is the system inertia matrix:

$$\{M\}_{mn} = \rho_0 \int_\Omega \varphi_{V\,m}\, \varphi_{V\,n}\, d\Omega \tag{9.145}$$

Comparing Eq. (9.56) with Eq. (9.145), it is verified that the inertia matrix is constant for incompressible flows.

The following potentials are defined:

$$\underline{\boldsymbol{p}}(\underline{\boldsymbol{V}}) = \frac{dT^*}{d\underline{\boldsymbol{V}}} = \underline{\underline{M}} \cdot \underline{\boldsymbol{V}} = \int_\Omega \boldsymbol{p}_v\, \underline{\varphi_V}\, d\Omega = \rho_0 \int_\Omega \boldsymbol{V}\, \underline{\varphi_V}\, d\Omega \tag{9.146}$$

$$\underline{\Theta}(\underline{S}) = \frac{dU}{d\underline{S}} = \underline{\underline{\Omega_S}}^{-1} \cdot \left[\int_\Omega \theta\, \underline{\varphi_S}\, d\Omega\right] \tag{9.147}$$

where $\underline{\boldsymbol{p}}$ and $\underline{\Theta}$ are respectively nodal vectors of linear momentum and temperature. The potentials defined in Eqs. (9.146) and (9.147) allow to represent kinetic and internal energy storage respectively as an inertial I and a capacitive C multibond field, as shown in Fig. 9.11.

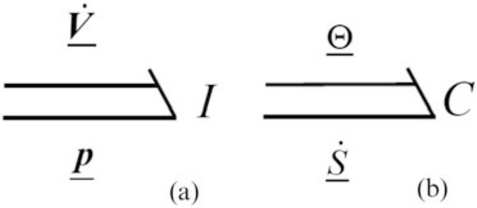


Fig. 9.11 Inertial (**a**) and capacitive (**b**) fields, representing kinetic and internal energy storage for an incompressible fluid

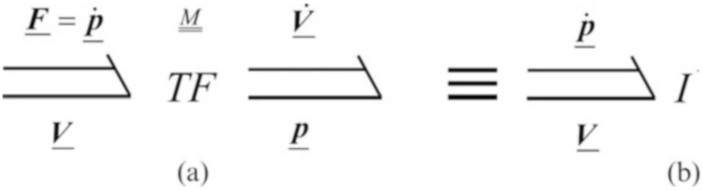


Fig. 9.12 Multibond transformer connected to the inertial port (**a**), and equivalent inertia field for incompressible flow (**b**)

As the inertia matrix is constant, Eq. (9.146) defines a multibond transformer relating the nodal vectors of velocity and linear momentum, as shown in Fig. 9.12a, with generalized effort given by:

$$\underline{F} = \underline{\underline{M}} \cdot \underline{\dot{V}} \tag{9.148}$$

From Eqs. (9.146) and (9.148), $\underline{F} = \underline{\dot{p}}$ for incompressible flows. In this case, the combination of the inertia field from Fig. 9.11a and the transformer of Fig. 9.12a are equivalent to an inertia field in which the generalized momentum is the nodal vector of linear momentum, as shown in Fig. 9.12b.

The time derivative of Eq. (9.143) can be written as:

$$\dot{E} = \underline{p}^T \cdot \underline{\dot{V}} + \underline{\Theta}^T \cdot \underline{\dot{S}} \tag{9.149}$$

9.3.3 State Equations

9.3.3.1 Velocity Port

Following the same procedure as in Sect. 9.2.3.2, the velocity state equation results:

$$\underline{\dot{V}} = \underline{\underline{M}}^{-1} \cdot \left(\underline{F_V^{(\Gamma)}} + \underline{F_P^{(\Gamma)}} + \underline{F_R} + \underline{F_P} + \underline{F_G} - \underline{F_V} - \underline{F_K} \right) \tag{9.150}$$

where $\underline{\boldsymbol{F}_V^{(\Gamma)}}$ and $\underline{\boldsymbol{F}_V}$ were already defined respectively in Eqs. (9.88) and (9.92), while the rest of the forces are defined as:

$$\underline{\boldsymbol{F}_P^{(\Gamma)}} = -\int_\Gamma P \,\underline{\varphi_V}\, \check{\boldsymbol{n}}\, d\Gamma \tag{9.151}$$

$$\underline{\boldsymbol{F}_R} = \rho_0 \int_\Omega \boldsymbol{V} \times (\nabla \times \boldsymbol{V})\, \underline{\varphi_V}\, d\Omega \tag{9.152}$$

$$\underline{\boldsymbol{F}_P} = \int_\Omega P\, \underline{\nabla \varphi_V}\, d\Omega \tag{9.153}$$

$$\underline{\boldsymbol{F}_G} = \rho_0 \int_\Omega \boldsymbol{G}\, \underline{\varphi_V}\, d\Omega \tag{9.154}$$

$$\underline{\boldsymbol{F}_K} = \rho_0 \int_\Omega \nabla \kappa\, \underline{\varphi_V}\, d\Omega \tag{9.155}$$

Adding the nodal components of Eq. (9.150) it can be easily shown that the integral momentum equation is satisfied:

$$\rho_0 \int_\Omega \frac{D\boldsymbol{V}}{Dt}\, d\Omega = \int_\Gamma \left(-P\, \underline{\underline{I}} + \underline{\underline{\tau}}\right) \cdot \check{\boldsymbol{n}}\, d\Gamma + \rho_0 \int_\Omega \boldsymbol{G}\, d\Omega \tag{9.156}$$

9.3.3.2 Entropy Port

Following the same procedure as in Sect. 9.2.3.3, the entropy state equation results:

$$\underline{\dot{S}} = \underline{\dot{S}_Q^{(\Gamma)}} + \underline{\dot{S}_C^{(\Gamma)}} + \underline{\dot{S}_Q} + \underline{\dot{S}_C} + \underline{\dot{S}_F} + \underline{\dot{S}_A} + \underline{\dot{S}_V} \tag{9.157}$$

where $\underline{\dot{S}_Q^{(\Gamma)}}$, $\underline{\dot{S}_C^{(\Gamma)}}$, $\underline{\dot{S}_Q}$, $\underline{\dot{S}_C}$ and $\underline{\dot{S}_V}$ were already defined respectively in Eqs. (9.99)–(9.102) and (9.106), while the rest of the nodal vectors of entropy rate are defined as:

$$\underline{\dot{S}_F} = \underline{\underline{\Theta}}^{-1} \cdot \left(\rho_0 \int_\Omega \underline{w_S}\, \Phi\, d\Omega\right) \tag{9.158}$$

$$\underline{\dot{S}_A} = \underline{\underline{\Theta}}^{-1} \cdot \left(\int_\Omega \underline{w_S}\, s_v\, \boldsymbol{V} \cdot \nabla\theta\, d\Omega\right) \tag{9.159}$$

Multiplying Eq. (9.157) by $\underline{\Theta}$, it can be easily shown that the entropy balance equation (9.140) integrated in volume is satisfied, that is:

$$\int_\Omega \theta\, \frac{\partial s_v}{\partial t}\, d\Omega = -\int_\Gamma (\boldsymbol{q} + \theta\, s_v\, \boldsymbol{V}) \cdot \check{\boldsymbol{n}}\, d\Gamma + \int_\Omega s_v\, \boldsymbol{V} \cdot \nabla\theta\, d\Gamma$$
$$+\rho_0 \int_\Omega \Phi\, d\Omega + \int_\Omega \left(\nabla \boldsymbol{V} : \underline{\underline{\tau}}\right) d\Omega \tag{9.160}$$

9.3.3.3 Pressure and Integral Incompressibility Constraint

Making the dot product of Eq. (9.150) and $\underline{\boldsymbol{V}}$ and taking into account Eq. (9.25), it can be obtained:

$$\int_{\Omega} \boldsymbol{p}_v \cdot \frac{\partial \boldsymbol{V}}{\partial t} \, d\Omega = \int_{\Gamma} \left(\underline{\underline{\tau}} \cdot \boldsymbol{V} \right) \cdot \check{\boldsymbol{n}} \, d\Gamma - \int_{\Gamma} \left(P \cdot \check{\boldsymbol{n}} \right) \, d\Gamma - \rho_0 \int_{\Omega} \nabla \kappa \cdot \boldsymbol{V} \, d\Omega$$
$$+\rho_0 \int_{\Omega} \boldsymbol{G} \cdot \boldsymbol{V} \, d\Omega - \int_{\Omega} \left(\nabla \boldsymbol{V} : \underline{\underline{\tau}} \right) \, d\Omega + \underline{\boldsymbol{F}_P}^T \cdot \underline{\boldsymbol{V}} \quad (9.161)$$

Comparing Eq. (9.139) integrated in volume with Eq. (9.161), the velocity balance equation (conservation of mechanical energy) integrated over the domain Ω is satisfied if $\underline{\boldsymbol{F}_P}^T \cdot \underline{\boldsymbol{V}} = 0$. This power term can be expressed as:

$$\underline{\boldsymbol{F}_P}^T \cdot \underline{\boldsymbol{V}} = \int_{\Omega} P \left(\nabla \cdot \boldsymbol{V} \right) d\Omega = \underline{P}^T \cdot \underline{Q} \quad (9.162)$$

where the nodal vector of volumetric flow $\underline{Q}$ results:

$$\underline{Q} = \int_{\Omega} \underline{\varphi_P} \left(\nabla \cdot \boldsymbol{V} \right) d\Omega \quad (9.163)$$

As a consequence, the integral incompressibility condition that must satisfy the discretized velocity field is:

$$\underline{Q} = \underline{0} \quad (9.164)$$

The system of equations (9.150) and (9.164) is coincident with the one obtained by the Galerkin method for the weak formulation of the problem in Finite Elements [47], in which general boundary conditions are possible at the bonds with the superficial forces $\underline{\boldsymbol{F}_V^{(\Gamma)}}$ and $\underline{\boldsymbol{F}_P^{(\Gamma)}}$.

Adding the components of vector $\underline{Q}$, it can be verified that the integral continuity equation is satisfied, that is:

$$\int_{\Omega} \left(\nabla \cdot \boldsymbol{V} \right) d\Omega = \int_{\Gamma} \boldsymbol{V} \cdot \check{\boldsymbol{n}} \, d\Gamma = 0 \quad (9.165)$$

The power conserving transformation between the force and pressure ports is represented by a multibond transformer, as shown in Fig. 9.13, with relations given by:

$$\underline{\boldsymbol{F}_P} = \underline{\underline{\boldsymbol{M}_{PV}}} \cdot \underline{P} \quad (9.166)$$

$$\underline{Q} = \underline{\underline{\boldsymbol{M}_{PV}}}^T \cdot \underline{\boldsymbol{V}} \quad (9.167)$$

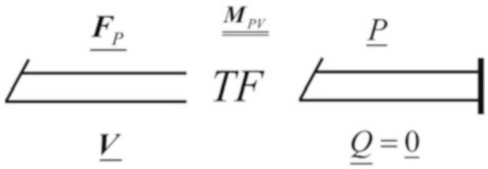

$\underline{\boldsymbol{F}_P}$
$\underline{\underline{\boldsymbol{M}_{PV}}}$
$\underline{P}$
TF
$\underline{\boldsymbol{V}}$
$\underline{Q} = \underline{0}$

Fig. 9.13 Multibond transformer representing the integral incompressibility constraint

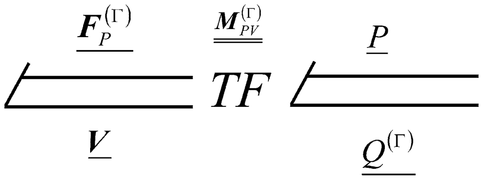

$\underline{\boldsymbol{F}_P^{(\Gamma)}}$
$\underline{\underline{\boldsymbol{M}_{PV}^{(\Gamma)}}}$
$\underline{P}$
TF
$\underline{\boldsymbol{V}}$
$\underline{Q^{(\Gamma)}}$

Fig. 9.14 Multibond transformer representing the superficial pressure force

where $\underline{\underline{\boldsymbol{M}_{PV}}}$ is a rectangular matrix, with n_V rows and n_P columns, defined as:

$$\{\boldsymbol{M}_{PV}\}_{mk} = \int_{\Omega} \nabla\varphi_{V\,m}\,\varphi_{P\,k}\,d\Omega \tag{9.168}$$

Concerning the bond corresponding to the superficial pressure force $\underline{\boldsymbol{F}_P^{(\Gamma)}}$, the power term can be written as:

$$\underline{\boldsymbol{F}_P^{(\Gamma)}} \cdot \underline{\boldsymbol{V}} = -\int_{\Gamma} P\,\boldsymbol{V} \cdot \check{\boldsymbol{n}}\,d\Gamma = \underline{P}^T \cdot \underline{Q^{(\Gamma)}} \tag{9.169}$$

where the nodal vector of superficial volumetric flow $\underline{Q^{(\Gamma)}}$ is defined as:

$$\underline{Q^{(\Gamma)}} = -\int_{\Gamma} \underline{\varphi_P}\,V \cdot \check{\boldsymbol{n}}\,d\Gamma \tag{9.170}$$

The power conserving transformation between the superficial force and pressure is represented by a multibond transformer, as shown in Fig. 9.14, with relations given by:

$$\underline{\boldsymbol{F}_P^{(\Gamma)}} = \underline{\underline{\boldsymbol{M}_{PV}^{(\Gamma)}}} \cdot \underline{P} \tag{9.171}$$

$$\underline{Q^{(\Gamma)}} = \underline{\underline{\boldsymbol{M}_{PV}^{(\Gamma)^T}}} \cdot \underline{\boldsymbol{V}} \tag{9.172}$$

where $\underline{\underline{\boldsymbol{M}_{PV}^{(\Gamma)}}}$ is a rectangular matrix, with n_V rows and n_P columns, defined as:

$$\left\{\boldsymbol{M}_{PV}^{(\Gamma)}\right\}_{mk} = -\int_{\Gamma} \varphi_{V\,m}\,\varphi_{P\,k}\,\check{\boldsymbol{n}}\,d\Gamma \tag{9.173}$$

9.3.4 System BG

The system BG is shown in Fig. 9.15. Energy storage (kinetic and internal) are represented respectively by an inertial *I* and a capacitive *C* field. At the 1-junction with common $\underline{V}$ all the nodal vector forces are added; in this way, the effort balance represents the linear momentum conservation equation for the nodal velocity values. At the 0-junction with common $\underline{\Theta}$ all the nodal entropy rates are added; in this way, the flow balance represents the thermal energy conservation equation for the nodal entropy values.

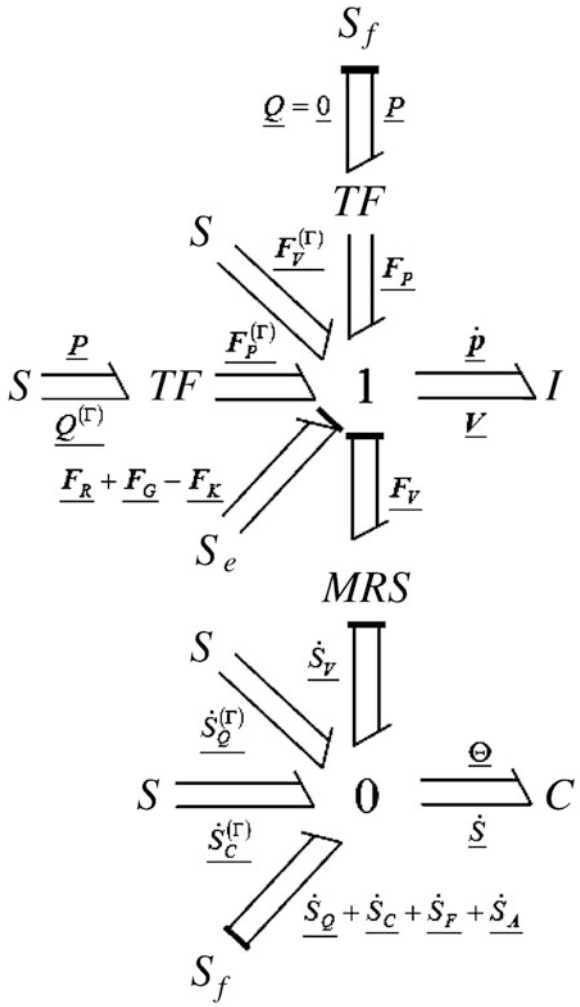

S_f
$\underline{Q} = \underline{0}$
$\underline{P}$
TF
S
$\underline{F}_V^{(\Gamma)}$
$\underline{F}_P$
$\underline{F}_P^{(\Gamma)}$
S
$\underline{P}$
TF
$\underline{Q}^{(\Gamma)}$
1
$\dot{\underline{p}}$
I
$\underline{V}$
$\underline{F}_R + \underline{F}_G - \underline{F}_K$
$\underline{F}_V$
S_e
MRS
S
$\dot{\underline{S}}_V$
$\dot{\underline{S}}_Q^{(\Gamma)}$
S
0
$\underline{\Theta}$
C
$\dot{\underline{S}}_C^{(\Gamma)}$
$\dot{\underline{S}}$
$\dot{\underline{S}}_Q + \dot{\underline{S}}_C + \dot{\underline{S}}_F + \dot{\underline{S}}_A$
S_f

Fig. 9.15 System BG for an incompressible flow

The *MRS* field connecting the 1 and 0 junctions represents the power transfer between the velocity and entropy ports, due to viscous dissipation.

The sources S (the ones connected to the bonds with $\underline{\boldsymbol{F}_V^{(\Gamma)}}$, $\underline{Q^{(\Gamma)}}$, $\underline{\dot{S}_Q^{(\Gamma)}}$, and $\underline{\dot{S}_C^{(\Gamma)}}$) represent different source terms related to the boundary conditions; as seen in Sect. 9.2.6.2, in each single port these sources behave as effort or flow sources, depending on the boundary conditions.

The rest of the sources, effort S_e or flow S_f (the ones connected to the bonds with $\underline{Q}$, $\underline{\boldsymbol{F}_R} + \underline{\boldsymbol{F}_G} - \underline{\boldsymbol{F}_K}$ and $\underline{\dot{S}_Q} + \underline{\dot{S}_C} + \underline{\dot{S}_F} + \underline{\dot{S}_A}$) represent volumetric power terms; the determination of causality for these sources and for the bonds connected to the *MRS* field results from the causality extension procedure detailed in Sect. 9.2.6.5. The power input in any bond corresponding to the multibond with $\underline{Q}$ is zero, according to the integral incompressibility constraint, Eq. (9.164); as a consequence the causality is such that, in any of these bonds, flow is imposed to the ports connected to the transformer and the modulated source becomes a flow source S_f. The net power input (sum over the bonds) corresponding to the multibond with the rotational force $\underline{\boldsymbol{F}_R}$ is zero, because of Eq. (9.25). As seen before, the net power input corresponding to the multibonds with the entropy rates $\underline{\dot{S}_Q}$ and $\underline{\dot{S}_C}$ is also zero.

9.4 Applications

9.4.1 *Numerical Benchmarks*

As it was stated before, the applications of this methodology to CFD problems were successful so far.

In [11, 27, 37], as an application example focused on the upwind problem, the formalism was applied to a simple one-dimensional ($0 \leq x \leq L$, where L is the length) advection-diffusion problem in which the thermophysical properties are constant, the velocity field is constant and thermal power source is zero. The resulting state equation involves only the entropy capacitive port of the system total energy. For each node, the shape functions were chosen to be piecewise constant and the weight functions to be piecewise linear, displaced by an upwind parameter β ($|\beta| \leq \frac{1}{2}$, as seen in Fig. 9.16a–c. The existing contributions coming from the discontinuities in the description of the flow fields could be successfully handled in the integration process by using distributional derivatives [31].

This kind of problem is interesting because destabilization of the numerical solution, related to the nodalization, may arise. The following thermal energy equation results:

$$\rho\, c_v \left(\frac{\partial \theta}{\partial t} + V\, \frac{\partial \theta}{\partial x} \right) = \lambda\, \frac{\partial^2 \theta}{\partial x^2} \tag{9.174}$$

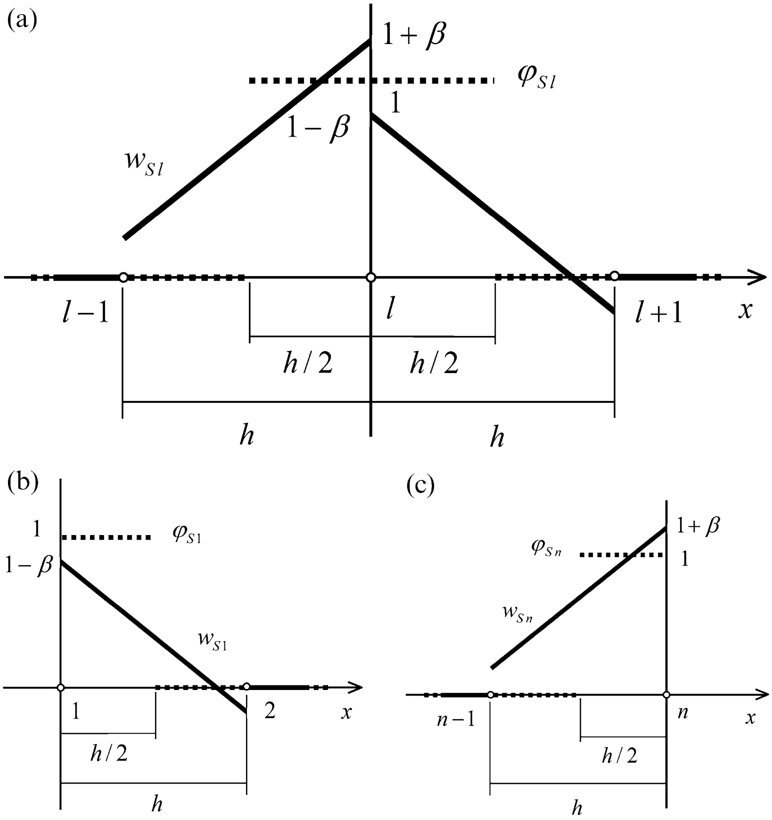


Fig. 9.16 Entropy test functions. In *dashed line*, the shape functions and, in *solid line*, the weight functions [11]. (**a**) Inner node. (**b**) First node. (**c**) Last node

As stated in Sect. 9.2.6.5, the resulting equation is second order and two space boundary conditions are needed. Considering the boundary conditions $\theta\,(0,\,t) = \theta_0$, $\theta\,(L,\,t) = \theta_L$ and $\theta\,(x,\,0) = \theta_0$, the nondimensional steady state solution θ^*_∞ is [35]:

$$\theta^*_\infty\left(x^*\right) = \frac{\theta_\infty - \theta_0}{\theta_L - \theta_0} = \frac{\exp\left(x^*\,Pe_L\right) - 1}{\exp\left(Pe_L\right) - 1} \tag{9.175}$$

$$Pe_L = \frac{\rho\,c_v\,V\,L}{\lambda} \tag{9.176}$$

where V is the velocity and Pe_L is the Peclet number. The transient nondimensional temperature

$$\theta^* = \frac{\theta - \theta_0}{\theta_L - \theta_0} \tag{9.177}$$

satisfies

$$\frac{\partial \theta^*}{\partial Fo} + Pe_L \, \frac{\partial \theta^*}{\partial x^*} = \frac{\partial^2 \theta^*}{\partial x^{*2}} \tag{9.178}$$

$$Fo = \frac{\lambda \, t}{\rho \, c_v \, L^2} \tag{9.179}$$

where *Fo* is the Fourier number. The solution of Eq. (9.178) is:

$$\begin{aligned}\theta^*_{ad}\left(x^*, \, Fo, \, Pe_L\right) = \frac{\exp\left(x^* \, Pe_L\right) - 1}{\exp\left(Pe_L\right) - 1} &+ \exp\left[-\left(1 - x^*\right) \frac{Pe_L}{2}\right] \\ &\times \sum_{m=1}^{\infty} \frac{8 \, (-1)^m \, m \, \pi}{\left(Pe_L^2 + 4 \, m^2 \, \pi^2\right)} \, \sin\left(m \, \pi \, x^*\right) \\ &\times \exp\left[-\frac{1}{4} \left(Pe_L^2 + 4 \, m^2 \, \pi^2\right) Fo\right]\end{aligned} \tag{9.180}$$

For diffusive problems, the corresponding entropy rate vectors $\underline{\dot{S}_Q}$ and $\underline{\dot{S}_Q^{(\Gamma)}}$ are nonzero. To satisfy the temperature boundary conditions, the sources $\dot{S}_{Q1}^{(\Gamma)}$ and $\dot{S}_{Qn}^{(\Gamma)}$ were adjusted in the iterative process in order to keep constant temperature values at $x = 0$ and $x = L$.

It was verified that, for centered schemes (no upwind, $\beta = 0$), the numerical solution is stable if the *grid* Peclet number $Pe_h = \frac{\rho \, c_v \, V \, h}{\lambda} < 2$, thus limiting the maximum grid spacing.

Many schemes for stabilizing the numerical solution for coarse meshes exist [35]. In the simulations, full upwind was chosen ($\beta = \frac{1}{2} \, \mathrm{sgn}\,(V)$); with full upwind, the solution is intrinsically stable, though it has more numerical diffusion. Figure 9.17 shows the comparison of the nondimensional numerical and analytical solutions for different nondimensional times (Fourier numbers) and full upwind. In order to obtain a fair simulation of the transient behavior, a sufficient number of time steps should be calculated before the times of interest. Here, a fixed time step was implemented such that there are ten steps until the first Fourier number plotted. A fair agreement can be observed.

In order to decrease the numerical diffusion for a stable numerical solution of a coarse mesh, different upwind schemes can be used. In particular, the following upwind scheme provides a numerical solution coincident with the exact one for one-dimensional advection-diffusion flow in steady state [27]:

$$\beta = \frac{\exp Pe_h + 1}{2 \, (\exp Pe_h - 1)} - \frac{1}{Pe_h} \tag{9.181}$$

Figure 9.18 shows the comparison of the nondimensional numerical and analytical solutions for different nondimensional times (Fourier numbers) using this controlled upwind scheme. An excellent agreement can be observed.

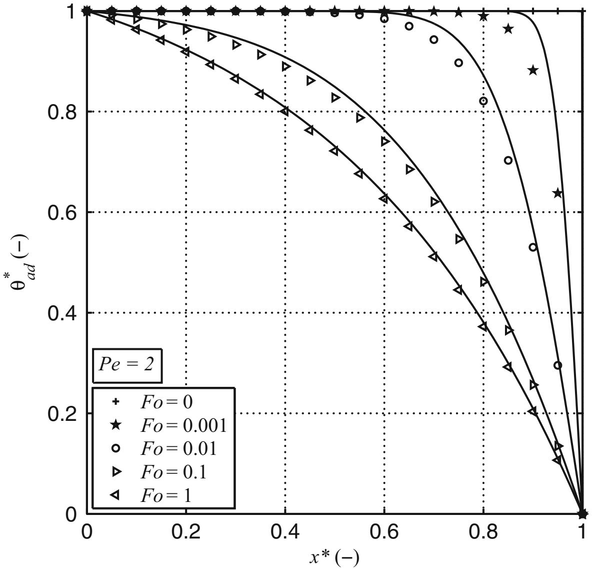


Fig. 9.17 Comparison of analytical (*solid line*, $Pe_L = 2$) and numerical nondimensional temperatures for combined advection and diffusion, with full upwind [11]

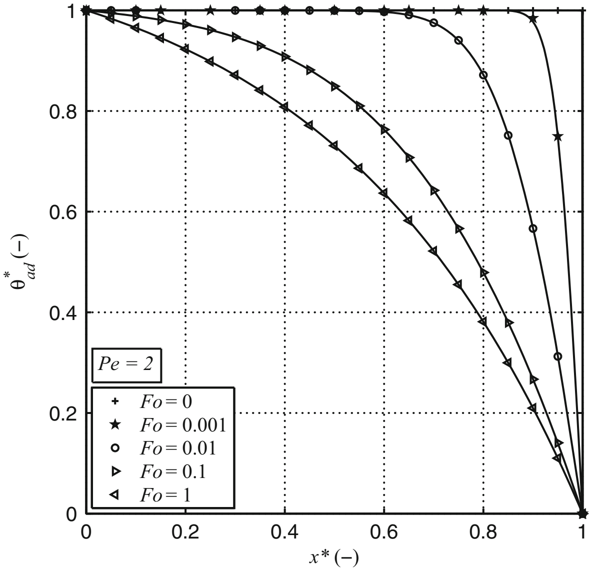


Fig. 9.18 Comparison of analytical (*solid line*, $Pe_L = 2$) and numerical nondimensional temperatures for combined advection and diffusion, with controlled upwind [11]

An important point in obtaining discrete solutions for problems of continuous nature is to ensure that the error (difference between the discrete and the analytical solutions) diminishes as the discretization is refined; this is verified with a mesh convergence test. Such analysis is shown in Fig. 9.19 for combined advection and diffusion flow, where increasing the number of nodes leads the steady state numerical solution closer to the analytical solution.

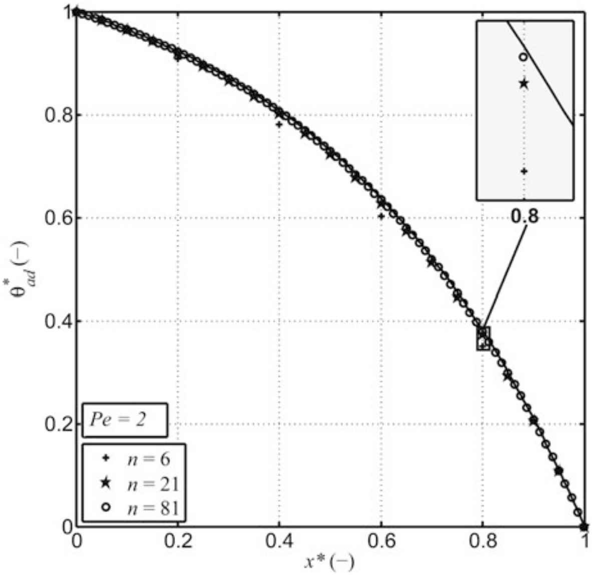


Fig. 9.19 Influence of number of nodes on steady state solution: mesh convergence test. Comparison of analytical (*solid line*, $Pe_L = 2$) and numerical solutions with full upwind for combined advection and diffusion [11]

A quantitative mesh convergence analysis is shown in Fig. 9.20, where the maximum normalized deviation between the analytical and numerical solution is plotted against the number of nodes. It can be seen that the rate of convergence is highly dependent on the upwind scheme used. Full upwind is intrinsically stable but has a slower convergence rate. It should be noted that zero upwind could have provided an unstable solution. The optimal upwind scheme for this specific case, given by Eq. (9.181), provides the best result. This is in accordance with standard results [35].

As seen before, the solution was easily stabilized by modifying the entropy weight functions in a very simple and automatic fashion, loosely related to the Petrov–Galerkin approach used in the FEM. As a consequence, advection-diffusion problems can be handled in the formalism through the right choice of automatically calculated upwind-biased weight functions. A full upwind scheme can be safely used.

In [1, 4], the resulting state equations were presented for a one-dimensional problem with constant piecewise shape functions and lumped forms of the temperature, Gibbs free energy, and kinetic coenergy per unit mass matrices. This nodalization and the choice of the shape functions allowed to perform a closed calculation of the state equations. As for this case the state variables correspond to the mass, velocity, and entropy in control volumes, it was possible to make a comparison with other numerical schemes. The existing contributions coming from the discontinuities in the description of the flow fields could be successfully handled in the integration process by using distributional derivatives [31]. Although viscous effects could not be modeled with a constant piecewise velocity profile, heat conduction could be taken into account with entropy weight functions with nonzero gradients at the

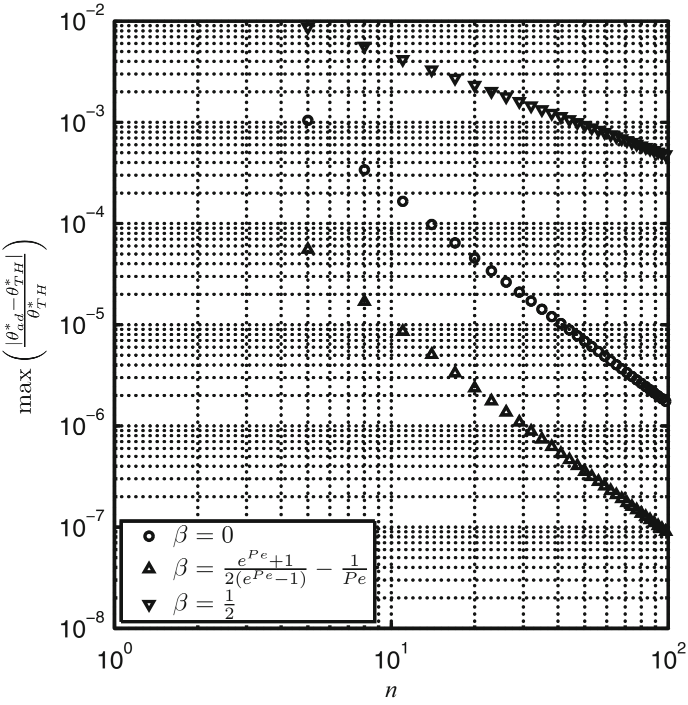


Fig. 9.20 Mesh convergence analysis for different upwind schemes. Relative deviation of numerical and analytical steady state solutions for combined advection and diffusion as a function of number of nodes [11]

discontinuities. The density and entropy weight functions, which are elements of this approach, were capable of taking into account the upwind nature of the fluid equations. Based on the linearized expressions of the state equations, a comparison was made with a finite-volume and with a finite-difference numerical scheme, obtaining an interpretation of the density and entropy weight functions appearing in the BG formalism. It was found that the density and entropy weight functions can be regarded as weight factors in the calculation of the corresponding fluxes within a control volume, while the gradient of the entropy weight function come out to be proportional to the weight factors in the calculation of the conductive entropy fluxes. Based on the Second Principle of Thermodynamics, it was also shown that the entropy weight functions must decrease as the distance to the corresponding node position increases.

In [28] the methodology was used to solve the so-called *shock tube* problem [45]. Concerning the shape functions, constant piecewise were adopted for density and entropy, while continuous linear were adopted for velocity, in order to be able to model viscous effects. As weight functions, linear piecewise were adopted for the entropy, while linear continuous were adopted for the density. Lumped forms of the temperature, Gibbs free energy, and kinetic coenergy per unit mass matrices were also chosen. No-flow and adiabatic boundary conditions were specified at both tube ends. Comparing the numerical and analytical solutions a reasonable agreement was found, being the numerical results a bit diffusive because of the artificial viscosity introduced. Although more work would be needed in the selection of weight and shape functions, the simple ones chosen in this work have shown to be adequate for dealing with a complex nonlinear problem involving all physical effects.

9.4.2 Extension to Other Flow Problems

9.4.2.1 Multicomponent Flows

In [2, 3] the methodology was extended to single-phase, multicomponent flows. A classical mixture, or solution, is a material in which the components are not physically distinct, that is, the mixing is at molecular level. In this case, when described using Continuum Theory, all the components of the solution are able to occupy the same region of space at the same time [25] and can be assumed to be in thermodynamic equilibrium. In a solution, each component has its own velocity, density, and internal energy. The balance principles for the components resemble those for a single component, except that they are allowed to interact with one another.

Two approximations were studied: the multivelocity model [2] and the diffusion model [3]. It was shown that, for the multivelocity model, the resulting independent variables are the densities and velocities of the components and the mixture entropy per unit volume. In the diffusion model, the dynamics of a multicomponent solution is described in terms of the average (center of mass) velocity of the mixture and the mass flux of each component relative to the average velocity, thus reducing the number of state variables. These relative mass fluxes are modeled using diffusion theory.

The diffusive mass fluxes consist of different contributions associated with the driving forces (mechanical or thermal) existing in the system [12]. In ordinary diffusion, the mass flux depends in a complicated way on the concentration gradients of the present components; in most of the problems, this is the most important contribution. The pressure diffusion indicates that there may be a differential net movement of a component in the mixture if there is a pressure gradient imposed to the system; this effect is important in centrifuge separation, in which tremendous pressure gradients are established. The forced diffusion appears when the components are under different external forces, as in the case of ionic systems in the presence of electric fields. Finally, the thermal diffusion describes the tendency for the components to separate under the influence of a temperature gradient. Although this effect is small, it can be enhanced by producing very steep temperature gradients.

The relative fluxes are assumed to be dependent on the entropy per unit volume and the component densities. This functional dependence allows to deal with ordinary (concentration driven) diffusion, pressure diffusion, forced diffusion (with steady forces), and thermal diffusion [12].

9.4.2.2 One-Dimensional Incompressible Pipe Flow

Incompressible pipe flow is a classic branch of Hydraulics which addresses liquid and low velocity gas flows within a closed conduit, without a free surface. The behavior of pipe flow is governed mainly by the Reynolds number, measuring the relative influence of inertial and viscous forces.

The one-dimensional analysis demands that suitable average values of all flow parameters are assumed to be functions of time and the coordinate along the length of the pipe. Small changes in pipe direction and cross-sectional area are allowed, as long as they do not create flow separation or secondary flows. In order to reintroduce the information lost in the averaging process, closure laws for wall momentum and heat transfer, as well as suitable profile correction factors, must be defined.

For incompressible flows, the volumetric flow is independent of position. The resulting effort balance at the inertial port results in the generalized Bernoulli equation, showing transient, convective, and frictional contributions to the piezometric pressure drop.

The BG methodology was applied to the PDEs corresponding to the one-dimensional approximations for incompressible pipe flows in [6]. In [11, 37] different transient problems were numerically implemented and simulated: advective heating of a fluid flow due to constant wall heat flux and due to constant wall temperature, advective heating of a fluid flow due to viscous dissipation and combined advection-diffusion. Viscous dissipation is of particular interest as it is not modeled in other BG formulations, which decouples the inertial and thermal ports. Also, combined advection-diffusion is important given that other BG formulations only take into account advection effects. For each simulation the complementary effects were suppressed for a correct comparison against analytical benchmark solutions. Finally, a unit cell BG representation was shown by enforcing a lumping approximation (summing over the rows or columns) in the matrices associated with the transient terms; comparisons were made for simulations run with this approximation, showing no deterioration of the obtained solutions.

9.4.2.3 Traffic Flow

Traffic flow was regarded, in the pioneer works [33, 38], as a compressible flow whose behavior is described by a density transport equation and a constitutive algebraic equilibrium law relating density and velocity. The so-called Lighthill, Whitham, and Richards (LWR) model has some limitations, being the most important the impossibility of describing problems in which boundary conditions are established at both ends of a traffic pathway (stop-and-go waves [41]).

To overcome the limitations of the LWR model, two-equation models were developed by adding a velocity transport equation [36]. As it was pointed out in [24], a vehicle can be regarded as a particle that responds to frontal stimuli, which means that the characteristic directions (eigenvalues) of the hyperbolic system of equations cannot be greater than the local velocity. Besides, the velocity cannot change sign (wrong-way-travel). Many models do not satisfy these requirements; as a consequence, they can lead to unrealistic results [24]. In [30] a continuum traffic flow model based on an improved car-following model was developed; the resulting system of equations is hyperbolic and doesn't suffer wrong-way-travel problems, making it suitable for the study of diverse nonlinear dynamical phenomena observed in freeway traffic. This model was extended to simulate the flow in two lanes [19].

In [20, 21] the BG methodology was used to frame traffic flow models, by making an analogy between traffic flow and compressible flow. Two-equation traffic flow models were framed within the BG methodology and some transient simulations corresponding to propagation of shock and rarefaction waves were successfully run.

9.4.2.4 One-Dimensional Compressible Pipe Flow

In [8] the BG methodology was used to model compressible one-dimensional pipe flows with rigid walls. All physical effects compatible with the one-dimensional approximation such as shear wall and normal stresses, wall and axial heat conduction, and flow passage area changes can be modeled naturally. The BG representation is similar to the one obtained in Sect. 9.2. Although only the theoretical aspects were presented, the model is suitable to study isentropic, Fanno or Rayleigh flows, as well as nozzle performance.

9.5 Conclusions and Perspectives

The present contribution addresses the theoretical development of a general true BG approach for CFD. The system state equations are obtained in terms of state variables.

For single-phase, single-component compressible flows the state variables result nodal values of mass, velocity, and entropy. The set of generalized effort and flow variables was derived based on energy considerations, while the state equations were obtained as a Galerkin formulation of the momentum conservation equation and Petrov–Galerkin formulations of the mass and entropy balance equations; as a consequence, the computational tools developed for the FEM, as well as for other numerical methods, can be used to solve the resulting state equations. It is interesting to notice that the matrices resulting for the mass and entropy state equations (respectively $\underline{\underline{\Psi}} + \underline{\underline{K}}$ and $\underline{\underline{\Theta}}$) are different to the ones found when other formulations for the mass and thermal energy conservation equations are discretized.

For single-phase, single-component incompressible flows the state variables result nodal values of pressure, velocity, and entropy. The resulting representation shows the role of pressure as external function acting to satisfy the incompressibility condition and the coupling between the inertial and thermal ports through the power dissipation term. The state equations for momentum and continuity equations are coincident with the one obtained by using the Galerkin method for the weak formulation of the problem in the FEM. The integral incompressibility constraint was derived based on the integral conservation of mechanical energy. The weak formulation for thermal energy equation was modeled with true BG elements, resulting a Petrov–Galerkin method. Again, the matrix resulting for entropy state equation $\underline{\underline{\Theta}}$ is different to the ones found when other formulations for the thermal energy conservation equation are discretized.

Regarding the power interaction between the system and the external environment, this is made through the boundary source terms $\underline{\dot{m}_W^{(\Gamma)}}$ and $\underline{\dot{S}_C^{(\Gamma)}}$ (related to fluid flow across the boundary), $\underline{\boldsymbol{F}_V^{(\Gamma)}}$ (related to viscous stresses applied at the boundary), and $\underline{\dot{S}_Q^{(\Gamma)}}$ (related to heat power at the boundary). Other (more subtle) interaction could be established through the force per unit mass $\boldsymbol{G}$, which may include non-inertial or electrodynamic forces, as in Magnetohydrodynamics.

As a result of a combination of BG concepts with elements of numerical methods, a new approach was developed, which is a foundation of a bridge between BGs and CFD. As the formulation is based on the definition of nodal discretized variables, different numerical schemes can be obtained by means of the appropriate choice of the interpolation and weight functions.

For the applications of the methodology made so far (single-phase, single or multicomponent systems) the modeling and determination of the state equations can be made based on Continuum Theory, conservation laws, Thermodynamics of equilibrium, and constitutive laws coming from the molecular theory. The difficulties in the resolution of a problem are limited, in these cases, to discretization and numerical aspects. Besides framing these flows within the BG theory, the contributions coming from the methodology in these problems are related to the establishment of acceptable boundary conditions satisfying causality.

For more sophisticated flows, like turbulent [46] or multiphase [25], the situation is different. Due to the closure problem in the Reynolds-averaged Navier–Stokes (RANS) equation approach for incompressible turbulence, for instance, it doesn't exist a theory for all turbulent flows, but models with restricted validity. These models differ in order (algebraic or with a variable number of transport equations), definition of independent variables and fundamentals, so there is no equivalence between them. The situation is far more complex for multiphase flows, where the existence of interfaces makes indispensable the definition of statistical averages related to the existence of a phase (component indicator function); new variables such as the void fraction or the interfacial volumetric area are vital in the description of multiphase systems.

It is interesting to notice that the state of the art in multiphase modeling is not sufficient to guarantee well-posed equations. For one-dimensional bubbly flow, there exist complex eigenvalues for a range of working parameters, which is physically unacceptable. These complex characteristics seem to arise from the unsuccessful modeling of the coupling terms between the momentum equations for each phase [25], being the systematic inclusion of terms soundly based on Physics the remedy to yield an appropriate working model. The concept of balance equations in the BG methodology, with coupling terms between ports, would render a new perspective to deal with this problem.

It becomes evident that the difficulties in the resolution of complex flows like turbulent or multiphase is related to modeling as well as to discretization. Thus, a methodology synthesizing properties of dynamic systems is very useful to frame different models, as well as to develop new ones. Obviously, BG theory does not replace Physics, but works as a structure that allows to discover mathematical inconsistencies from the beginning of the modeling process. A BG based methodology contributes to answer questions such as what the independent variables are, what equations must be satisfied in order to guarantee conservation of energy in the system, and how the boundary conditions are introduced.

This contribution shows that starting from the governing PDEs and using discretization techniques coming from CFD is the right strategy for producing general models, framed within the BG theory, for fluid dynamic systems. The author hopes that the findings of this contribution encourage other researchers to use this formalism in more complex problems.

References

1. Baliño, J. L. (2002). Bond-graph approach for computational fluid dynamics: A comparison with other numerical methods. In *Second IEEE International Conference on System, Man and Cybernetics (SMC 2002), Paper TA1B3*, 6 pp.
2. Baliño, J. L. (2003). BG-CFD methodology for multicomponent solutions. Part I: Multivelocity model. In *2003 International Conference on Bond Graph Modeling and Simulation (ICBGM 2003)* (pp. 41–46).
3. Baliño, J. L. (2003). BG-CFD methodology for multicomponent solutions. Part II: Diffusion model. In *2003 International Conference on Bond Graph Modeling and Simulation (ICBGM 2003)* (pp. 47–52).
4. Baliño, J. L. (2003). Bond-graph formulation of CFD problems with constant piecewise shape functions. *International Journal of Heat and Technology, 21*(1), 59–66.
5. Baliño, J. L. (2005). Bond graph modeling of incompressible flows. In *17th IMACS World Congress, Scientific Computation, Applied Mathematics and Simulation (IMACS 2005)*, 8 pp.
6. Baliño, J. L. (2006). Modeling one-dimensional incompressible duct flows. In *20th European Conference on Modelling and Simulation (ECMS 2006), Paper 76*, 6 pp.
7. Baliño, J. L. (2009). Galerkin finite element method for incompressible thermofluid flows framed within the bond graph theory. *Simulation Modelling Practice and Theory, 17*, 35–49.
8. Baliño, J. L. (2016). True bond graph formulation for compressible thermofluid duct flows. In *The 9th International Conference on Integrated Modeling and Analysis in Applied Control and Automation (IMAACA 2016) (pp. 24–31).*
9. Baliño, J. L., Larreteguy, A. E., & Gandolfo, E. F. (2001). A general bond graph approach for computational fluid dynamics. Part I: Theory. In *2001 International Conference on Bond Graph Modeling and Simulation (ICBGM 2001)* (pp. 41–46).

10. Baliño, J. L., Larreteguy, A. E., & Gandolfo, E. F. (2006). A general bond graph approach for computational fluid dynamics. *Simulation Modelling Practice and Theory, 14*, 884–908.
11. Baliño, J. L., & Pellegrini, S. P. (2016, in press). True bond graph formulation for one-dimensional incompressible pipe flows. Modeling and applications. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*.
12. Bird, R. B., Stewart, W. E., & Lightfoot, E. N. (1960). *Transport phenomena*. New York: Wiley.
13. Borutzky, W. (2004). *Bond graphs. A methodology for modelling multidisciplinary dynamic systems*. San Diego, CA: SCS Publishing House.
14. Borutzky, W. (2010). *Bond graph methodology. Development and analysis of multidisciplinary dynamic systems models*. New York: Springer.
15. Borutzky, W. (2011). *Bond graph modelling of engineering systems. Theory, applications and software support*. New York: Springer.
16. Brown, F. T. (2006). *Engineering system dynamics. A unified graph-centered approach*. Boca Raton, FL: CRC Press.
17. Callen, H. B. (1960). *Thermodynamics*. New York: Wiley.
18. Cellier, F. E. (1991). *Continuous system modeling*. New York: Springer.
19. Chang-Fu. T., Jiang. R., & Qing-Song, W. (2007). Extended speed gradient model for traffic flow on two-lane freeways. *Chinese Physics, 16*, 1570–1575.
20. Chera, C. M., Baliño, J. L., & Dauphin-Tanguy, G. (2010). Two-equation traffic flow models framed within the bond graph theory. In *2010 International Conference on Bond Graph Modeling and Simulation (ICBGM 2010)*, 8 pp.
21. Chera, C. M., Baliño, J. L., & Dauphin-Tanguy, G. (2013). Bond graph and computational fluid dynamics in traffic flow. In *2nd International Conference on Systems and Computer Science (ICSCS 2013)*, 6 pp.
22. Crandall, S. H., Karnopp, D. C., Kurtz, E. F., & Pridmore-Brown, D. C. (1968). *Dynamics of mechanical and electromechanical systems*. New York: MacGraw-Hill.
23. Cuvelier, C., Segal, A., & van Steenhoven, A. A. (1986). *Finite element methods and Navier-Stokes equations*. Dordrecht: D. Reidel Publishing Company.
24. Daganzo, C. F. (1995). Requiem for second-order fluid approximations of traffic flow. *Transportation Research Part B, 29B*, 277–286.
25. Drew, D. A., & Passman, S. L. (1999). *Theory of multicomponent fluids*. New York: Springer.
26. Fahrenthold, E. P., & Venkataraman, M. (1996). Eulerian bond graphs for fluid continuum dynamics modeling. *ASME Journal of Dynamic Systems, Measurement, and Control, 118*, 48–57.
27. Gandolfo Raso, E. F., Larreteguy, A. E., & Baliño, J. L. (2001). A general bond graph approach for computational fluid dynamics. Part II: Applications. In *2001 International Conference on Bond Graph Modeling and Simulation (ICBGM 2001)* (pp. 47–52).
28. Gandolfo Raso, E. F., Larreteguy, A. E., & Baliño, J. L. (2002). Modeling of 1-D compressible flows. In *Second IEEE International Conference on System, Man and Cybernetics (SMC 2002), Paper TA1B4*, 6 pp.
29. Gawthrop, P. J., & Smith, L. P. S. (1996). *Metamodelling: Bond graphs and dynamic systems*. London: Prentice-Hall.
30. Jiang, R., Wu, K. S., & Zhu, Z. J. (2001). A new continuum model for traffic flows and numerical tests. *Transportation Research Part B, 36*, 405–419.
31. Kanwal, R. P. (1998). *Generalized functions: Theory and technique*. Boston: Birkhëuser.
32. Karnopp, D. C., Margolis, D. L., & Rosenberg, R. C. (2012). *System dynamics: Modeling, simulation and control of mechatronic systems*. New York: Wiley.
33. Lighthill, M. J., & Whitham, G. B. (1955). On kinematic waves II. A theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London, Series A, 229*, 317–345.
34. Löhner, R. (2001). *Applied CFD techniques: An introduction based on finite element methods*. New York: Wiley.
35. Patankar, S. V. (1980). *Numerical heat transfer and fluid flow*. Washington: Hemisphere Publishing Corporation.

36. Payne, H. J. (1971). Models of freeway traffic and control. *Mathematical Models of Public System, 1*, 51–61.
37. Pellegrini, S. P., & Baliño, J. L. (2014). Application of a true bond graph formulation for incompressible thermofluid duct flows. In *2014 International Conference on Bond Graph Modeling and Simulation (ICBGM 2014)*, 10 pp.
38. Richards, P. I. (1956). Shock waves on the highway. *Operations Research, 4*, 42–51.
39. Rosenberg, R. C., & Karnopp, D. C. (1983). *Introduction to physical system dynamics*. New York: McGraw-Hill.
40. Schapiro, A. V. (1953). *The dynamics and thermodynamics of compressible fluid flow* (Vol. I). New York: The Ronald Press Company.
41. Schönhof, M., & Helbing, D. (2007). Empirical features of congested traffic state and their implications for traffic modeling. *Transportation Science, 41*, 135–166.
42. Tannehill, J. C., Anderson, D. A., & Pletcher, R. H. (1997). *Computational fluid mechanics and heat transfer*. Washington, DC: Taylor & Francis.
43. Thoma, J., & Ould Bouamama, B. (2000). *Modelling and simulation in thermal and chemical engineering. A bond graph approach*. Berlin: Springer.
44. Whitaker, S. (1977). *Fundamental principles of heat transfer*. New York: Pergamon Press.
45. Whitham, G. B. (1999). *Linear and nonlinear waves*. New York: Wiley.
46. Wilcox, D. C. (2000). *Turbulence modeling for CFD*. La Canada, CA: DCW Industries, Inc.
47. Zienkiewicz, O. C., Taylor, R. L., & Nithiarasu, P. (2005). *The finite element method for fluid dynamics*. Boston: Elsevier.

Chapter 10
Model Complexity of Distributed Parameter Systems: An Energy-Based Approach

L.S. Louca

10.1 Introduction

Modeling and simulation have yet to achieve wide utilization as commonplace engineering tools. One reason for this is that current modeling and simulation techniques are inadequate. Specifically, a major disadvantage is that they require sophisticated users who are often not domain experts and thus lack the ability to effectively utilize the model and simulation tools to uncover the important design trade-offs. Another drawback is that models are often large and complicated with many parameters, making the physical interpretation of the model outputs, even by domain experts, difficult. This is particularly true when "unnecessary" features are included in the model.

A variety of algorithms have been developed and implemented to help automate the production of proper models of dynamic systems. Wilson and Stein developed Model Order Deduction Algorithm (MODA) that deduces the required system model complexity from subsystem models of variable complexity using a frequency-based metric [25]. They also defined proper models as the models with physically meaningful states and parameters that are of necessary but sufficient complexity to meet the engineering and accuracy objectives. Additional work on deduction algorithms for generating proper models in an automated fashion has been reported by previous research [4, 5, 24]. The above algorithms have also been implemented in an automated modeling computer environment [22].

In an attempt to overcome the limitations of the frequency-based metrics, the author introduced a new model reduction technique that also generates proper models [16]. This approach uses an energy-based metric (element activity) that

L.S. Louca (✉)
Department of Mechanical and Manufacturing Engineering, University of Cyprus,
Nicosia 1678, Cyprus
e-mail: lslouca@ucy.ac.cy

W. Borutzky (ed.), *Bond Graphs for Modelling, Control and Fault Diagnosis of Engineering Systems*, DOI 10.1007/978-3-319-47434-2_10

in general can be applied to nonlinear systems, and considers the importance of all energetic elements (generalized inductance, capacitance, and resistance) [17]. The contribution of each energy element in the model is ranked according to the activity metric under specific excitation. Elements with small contributions are eliminated in order to produce a reduced model using a systematic methodology called Model Order Reduction Algorithm (MORA). The activity metric was also used as a basis for even further reduction, through partitioning the model into smaller and decoupled submodels [19].

Such modeling approaches should be able to handle real mechanical systems that typically include distributed parameter (continuous) components, e.g., rods, beams, plates, etc. Frequently, modeling objectives and assumptions allow the lumping of continuous component properties into ideal energy elements that lead to a dynamic model described by a set of ordinary differential equations. However, when property lumping is not acceptable, modeling of a continuous component requires a different approach since its inertial, compliance, and resistive properties are spatially distributed and cannot be lumped into single equivalent elements. The dynamic behavior of continuous components is thus described by partial differential equations with derivatives in both time and space. Another approach that is considered in this work is the modeling of a continuous component with finite segments that are spatially distributed. This is an approximation for which the accuracy is a function of the number of segments. The model accuracy improves as the number of segments increases. Model accuracy and the required number of segments can be addressed using a frequency-based metric [5].

Beyond the physical-based modeling, modal decomposition is also used to model and analyze continuous and discrete systems [18]. One of the advantages of modal decomposition is the ability to straightforwardly adjust (i.e., reduce) model complexity since all modes are orthogonal to each other. The reduction of such modal decomposition models is mostly based on frequency, and the user-defined Frequency Range Of Interest (FROI) determines the frequencies that are important for a specific scenario. In this case, modes with frequencies within the FROI are retained in the reduced model and modes outside this range are eliminated. As expected, mode truncation introduces error in the predictions that can be measured and adjusted based on the accuracy requirements [9, 10].

Element activity is another metric that has more flexibility than frequency-based metrics, which address the issue of model complexity by only adding compliant elements, leaving unaccounted the importance of inertial and resistive elements. In contrast, the activity metric considers the importance of all energetic elements, and therefore, the significance of all energy elements in the model can be quantified. It is the purpose of this work to develop a new methodology using the activity metric for addressing the model complexity of distributed parameter systems and specifically cantilever beams. The methodology is specifically developed using the finite segment approximation and the goal is to identify the physical phenomena to be included in each segment in order to accurately predict the dynamic behavior.

The chapter starts with providing background on the energy-based activity metric along with the reduction algorithm. Next, the equation formulation for a finite segment Timoshenko beam is presented along with the closed form expressions for steady state element activities. Then, the complexity of a cantilever beam is analyzed under various conditions using MORA. Finally, in the last section, discussion and conclusions are provided.

10.2 Background

The original work on the energy-based metric for model reduction is briefly described here for convenience. More details, extensions, and applications of this methodology can be found in previous publications [13–15, 17]. The main idea behind this model reduction technique is to evaluate the "element activity" of individual energy elements in a full system model under a stereotypical set of inputs and initial conditions. The activity of each energy element establishes a hierarchy of importance for all elements in a system. Those below a user-defined threshold of acceptable level of activity are eliminated from the model. A reduced model is then generated and a new set of governing differential equations is derived.

The activity metric has been formulated originally for systems with nonlinearities in both the element constitutive laws and kinematics. In this work, the activity metric is applied to linear systems for which analytical expressions for the activity can be derived, and therefore, avoid the use of numerical time integration that could be cumbersome. The analysis is further simplified if, in addition to the linearity assumption, the system is assumed to have a single sinusoidal excitation, and only the steady state response is studied. These assumptions are motivated from Fourier analysis where an arbitrary function can be decomposed into a series of harmonics. Using this frequency decomposition, the activity analysis can be performed as a function of frequency in order to study the frequency dependency of element activity in a dynamic system.

10.2.1 Element Activity for Linear Systems

A measure of the power response of a dynamic system, which has physical meaning and a simple definition, is used to develop the modeling metric, element activity (or simply "activity"). Element activity, A, is defined for each energy element as:

$$A = \int_0^\tau |\mathbb{P}(t)| \mathrm{d}t \tag{10.1}$$

where $\mathbb{P}(t)$ is the element power and τ is the time over which the model has to accurately predict the system behavior. The activity has units of energy, representing the amount of energy that flows in and out of the element over the given time τ. The energy that flows in and out of an element is a measure of how active this element is (how much energy passes through it), and consequently the quantity in (10.1) is termed activity. Activity can be defined independent of the energy domain, type of energy element, or nonlinearities.

The activity is calculated for each energy element based on the system response that is calculated from the system's state equations. In the case that the system is modeled using a bond graph formulation, the state equations are derived using the multi-port bond graph representation [2, 3, 7, 21]. In addition, when the system has linear junction structure and constitutive laws and a single input, the state equations are linear time invariant and have the following general form:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}u \tag{10.2}$$

where, $\mathbf{A} \in \mathbb{R}^{m \times m}, \mathbf{b} \in \mathbb{R}^{m}$ are the state space matrices, $\mathbf{x} \in \mathbb{R}^{m}$ is the state variable vector, $u \in \mathbb{R}$ is the input, and m is the number of independent states.

For the above system appropriate outputs are defined in order to effortlessly calculate the power of each energy element in the model using the constitutive law of each element. For convenience, the outputs are selected to be the generalized flow, effort, and flow for inertial, compliant, and resistive elements, respectively. The dual effort or flow variables needed for calculating the power are derived from the output variables and constitutive laws. The output vector for this set of variables has the form:

$$\mathbf{y} = \left\{ \begin{matrix} \mathbf{f}_I \\ \mathbf{e}_C \\ \mathbf{f}_R \end{matrix} \right\} \tag{10.3}$$

where $\mathbf{y} \in \mathbb{R}^{k}$ and $\mathbf{f}_I \in \mathbb{R}^{k_I}, \mathbf{e}_C \in \mathbb{R}^{k_C},$ and $\mathbf{f}_R \in \mathbb{R}^{k_R}$. The variables k_I, k_C, and k_R represent the number of inertial, compliant, and resistive elements, respectively. The total number of energy elements is $k = k_I + k_C + k_R$. Note that the output vector is defined such that the required variables of the inertial elements are first followed by the variables of compliant and then resistive elements.

Each output variable is a linear function of the state variables, and possibly input, given that they have linear constitutive laws. Using the output variables set in (10.3), the output equations are written as:

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{d}u \tag{10.4}$$

where $\mathbf{C} \in \mathbb{R}^{k \times m}, \mathbf{d} \in \mathbb{R}^{k}$ are the output state space matrices.

Given this set of output variables the missing efforts or flows, needed for calculating the element power, are computed from the linear constitutive laws of each type of energy element as shown below:

$$\begin{array}{l} \mathsf{I}: \; p_I = r_I f_I \iff e_I = \dot{p}_I = r_I \dot{f}_I \\ \mathsf{C}: \; q_C = r_C e_C \iff f_C = \dot{q}_C = r_C \dot{e}_C \\ \mathsf{R}: \; e_R = r_R f_R \end{array} \tag{10.5}$$

where r_I, r_C, r_R are known constants representing the linear constitutive law coefficients of inductance, compliance, and resistance, respectively. For deriving compact expressions in the analysis, a vector, $\mathbf{r} \in \mathbb{R}^k$, with all the linear constitutive law coefficients is introduced as shown below:

$$\mathbf{r} = \begin{Bmatrix} \mathbf{r}_I \\ \mathbf{r}_C \\ \mathbf{r}_R \end{Bmatrix} \tag{10.6}$$

where $\mathbf{r}_I \in \mathbb{R}^{k_I}$, $\mathbf{r}_C \in \mathbb{R}^{k_C}$, and $\mathbf{r}_R \in \mathbb{R}^{k_R}$ are the constant constitutive law coefficients.

Finally, the power needed for calculating the activity of each element, as defined in (10.1), is computed as the product of generalized effort and flow. By using (10.5) the following expressions for the power of each element type are derived:

$$\begin{aligned} \mathbf{I}:\ & \mathbb{P}_I = e_I f_I = r_I f_I \dot{f}_I \\ \mathbf{C}:\ & \mathbb{P}_C = e_C f_C = r_C e_C \dot{e}_C \\ \mathbf{R}:\ & \mathbb{P}_R = e_R f_R = r_R f_R f_R = r_R f_R^2 \end{aligned} \tag{10.7}$$

The expressions for element power in (10.7) are generalized with the use of the defined structure of the output vector in (10.3) and parameter vector in (10.6). Thus, the power for energy storage elements (inertial and compliant) is given by (10.8) and for energy dissipation elements (resistive) in (10.9).

$$\mathbb{P}_i = r_i y_i \dot{y}_i, \quad i = 1, \ldots, k_I + k_C \tag{10.8}$$

$$\mathbb{P}_i = r_i y_i^2, \quad i = k_I + k_C + 1, \ldots, k \tag{10.9}$$

The above element power is then used to calculate the element activity based on its definition in (10.1). Element parameters are assumed to be constant thus the activity for the energy storage elements is given in (10.10) and for energy dissipation element in (10.11).

$$A_i = \int_0^\tau |\mathbb{P}_i| = r_i \int_0^\tau \left| y_i \dot{y}_i \right| \mathrm{d}t, \quad i = 1, \ldots, k_I + k_C \tag{10.10}$$

$$A_i = \int_0^\tau |\mathbb{P}_i| = r_i \int_0^\tau \left| y_i^2 \right| \mathrm{d}t = r_i \int_0^\tau y_i^2 \mathrm{d}t, \quad i = k_I + k_C + 1, \ldots, k \tag{10.11}$$

10.2.2 Activity for Single Harmonic Excitation

The time response of the output vector, $\mathbf{y}(t)$, in (10.3) and (10.10) is required in order to complete the calculation of element power. For nonlinear systems, numerical integration is typically used to calculate the system response; however, in this case linear system analysis can be used to obtain closed form expressions. In addition, for the purposes of this work, the excitation is assumed to be a single harmonic given by:

$$u(t) = U \sin(\omega t) \tag{10.12}$$

where $U \in \mathbb{R}$ is the amplitude of the excitation and ω is the excitation frequency. The steady state response of the linear system in (10.2) and (10.4) under the harmonic excitation in (10.12) is calculated using linear system analysis theory. The response is given by the following closed form expression:

$$y_i(t,\omega) = UY_i(\omega) \cdot \sin(\omega t + \varphi_i(\omega)), \quad i = 1,\ldots,k \tag{10.13}$$

where $Y_i(\omega)$ and $\varphi_i(\omega)$ are the steady state amplitude and phase shift, respectively, that can be easily calculated using linear system analysis.

Within the context of this analysis, the output $y_i(t,\omega)$ in (10.13) is either an effort or a flow that is used for calculating the power of each element in (10.8) and (10.9). Finally, the activity can be calculated by (10.10) and (10.11), but first the upper bound, τ, of this integral must be specified. For this case, the steady state and periodicity features of the response are exploited. A periodic function repeats itself every T seconds, and therefore, a single period of this function contains the necessary information about the response. Thus, the upper bound of the integral is set to one period of the excitation, $\tau = T = 2\pi/\omega$. Therefore, the steady state activity for energy storage elements is given by:

$$\begin{aligned} A_i^{\text{ss}}(\omega) &= r_i \int_0^T |y_i \dot{y}_i| \mathrm{d}t \\ &= \frac{1}{2} r_i U^2 Y_i^2(\omega)\, \omega \int_0^T |\sin(2(\omega t + \varphi_i(\omega)))| \mathrm{d}t \\ &\Rightarrow A_i^{\text{ss}}(\omega) = 2 r_i U^2 Y_i^2(\omega) \end{aligned} \tag{10.14}$$

and for energy dissipation elements by:

$$\begin{aligned} A_i^{\text{ss}}(\omega) &= r_i \int_0^T y_i^2 \mathrm{d}t \\ &= r_i U^2 Y_i^2(\omega) \int_0^T \sin^2(\omega t + \varphi_i(\omega)) \mathrm{d}t \\ &\Rightarrow A_i^{\text{ss}}(\omega) = \frac{\pi r_i U^2 Y_i^2(\omega)}{\omega} \end{aligned} \tag{10.15}$$

The above simple closed form expressions can be used to calculate the activity of energy elements for a given single harmonic excitation. These expressions are proportional to the square of the amplitude; however, they have no dependency on the phase shift that is eliminated through the integration. The superscript “ss” in

(10.14) and (10.15) denotes the activity under a steady state harmonic response. Note that the activity for both energy storage and energy dissipation elements is a function of the excitation frequency but not the phase shift.

10.2.3 Activity Index and MORA

The activity as defined in (10.1) is a measure of the absolute importance of an element as it represents the amount of energy that flows through the element over a given time period. In order to obtain a relative measure of the importance, the element activity is compared to a quantity that represents the "overall activity" of the system. This "overall activity" is defined as the sum of all the element activities of the system, is termed total activity (A^{Total}), and is given by:

$$A^{\text{Total}}(\omega) = \sum_{i=1}^{k} A_i(\omega) \tag{10.16}$$

where A_i is the activity of the ith element given by (10.14) and (10.15). Thus a normalized measure of element importance, called element activity index or just activity index, is defined as:

$$\text{AI}_i^{\text{ss}}(\omega) = \frac{A_i^{\text{ss}}(\omega)}{A^{\text{Total}}(\omega)} = \frac{A_i^{\text{ss}}(\omega)}{\sum\limits_{i=1}^{k} A_i^{\text{ss}}(\omega)} \tag{10.17}$$

The activity index, $\text{AI}_i^{\text{ss}}(\omega)$, is calculated for each element in the model and it represents the portion of the total system energy that flows through a specific element. The input amplitude, U, does not appear in any of the element activity indices since all element activities are proportional to the square of the amplitude.

With the activity index defined as a relative metric for addressing element importance, the Model Order Reduction Algorithm (MORA) is constructed. The first step of MORA is to calculate the activity index for each element in the system as defined in (10.17). Next, the activity indices are sorted to identify the elements with high activity (most important) and low activity (least important). With the activity indices sorted, the model reduction proceeds given the desired engineering specifications. These specifications are defined by the modeler who then converts them into a threshold β of the total activity (e.g., 99 %) that he or she wants to include in the reduced model. This threshold defines the borderline between the eliminated and retained elements in the model. The elimination process is shown in Fig. 10.1 where the sorted activity indices are summed starting from the most important element until the specified threshold is reached. The element which, when included, increments the cumulative activity index above the threshold, is the last element to be included in the reduced model. The elements that are above this threshold are removed from the model, e.g., when using the bond graph formulation,

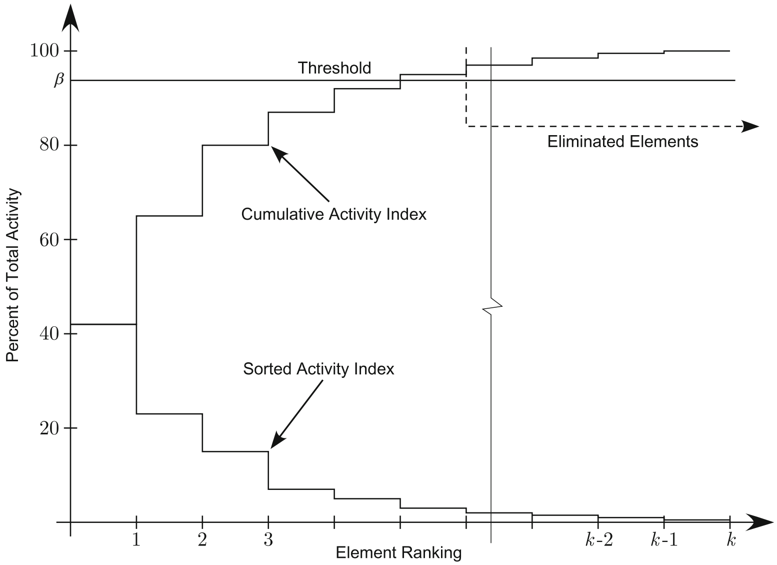


Fig. 10.1 Activity index sorting and elimination

delete the corresponding low activity energy element. The junction structure of the bond graph is retained in the reduced model, and therefore, the reduced model realization is the same as the full model, so its physical meaning and relation to the physical system are retained.

10.3 Cantilever Beam Model

The state space model used in the previous section assumes a lumped parameter system representation, where individual components exhibit only inertial, compliant, or resistive behavior. This means that the dynamic behavior of a component can be lumped and modeled as a single inertial, compliant, or resistive energy element. This can be a valid assumption for many components; however, real system components can possess all dynamic properties (inertial, compliant, and resistive) simultaneously. In addition, these properties may vary or be distributed spatially. In these cases, a lumped parameter modeling approach cannot be used since it will result in an incorrect model and produce inaccurate predictions. An example of such component is a beam and more specifically a cantilever beam that is widely used in engineering applications. Therefore, these components must be considered as distributed parameter or continuous, which require a different modeling approach.

Models of continuous systems are developed using solid mechanics theory, which instead of Ordinary Differential Equations, lead to Partial Differential

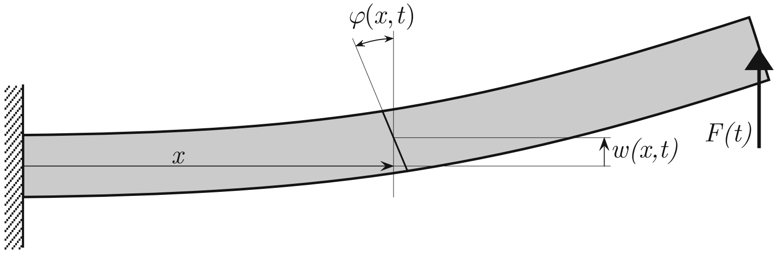


Fig. 10.2 Cantilever beam transverse vibration

Equations (PDE) with derivatives in both space and time [1, 6, 8, 23]. The continuous cantilever beam used in this work is shown in Fig. 10.2, where its transverse motion is considered when excited by a vertical load at its free end. The motion of a given gross section, $w(x,t)$ and $\varphi(x,t)$, from its undeformed state varies with time and location thus having a set of PDEs describing its motion. Note that due to the rotation φ, a cross section does not remain normal to the neutral axis according to the Timoshenko beam theory. One method for solving these PDEs is separation of variables, which produces a modal expansion solution [18]. This approach can also be combined with other lumped parameter elements in order to model a real system that consists of both lumped and distributed parameter components [7]. An analysis of the advantages and disadvantages of this approach is beyond the scope of this work, however, it is safe to say that the solution of PDEs is more cumbersome than the solution of ordinary differential equations that describe the behavior of lumped parameters system.

A different approach for modeling the transverse vibration of a cantilever beam is to divide it into segments of equal length. This approach is motivated by the procedure for deriving the PDEs describing the motion of a beam. Each of these segments has linear inertial and compliant properties that can be determined from solid mechanics theory. Shear effects and rotational inertial effects are also considered, which results in a more generic model that is valid for a larger range of geometric parameters. This is known as the Timoshenko beam model, which is usually used for non-slender beams in order to get accurate model predictions. The use of this more complex model using the Timoshenko beam theory is also mandated from the use of MORA in the process of determining the appropriate model complexity. In this approach the most complex model is first developed, and then MORA is used to identify what is actually needed in order to reach a reduced model with accurate predictions.

The ideal physical model of a cantilever beam under these assumptions is shown in Fig. 10.3 where the beam is divided into n equal segments. This model approaches the partial differential equations of the continuous system, as the number of segments approaches infinity. However, it is difficult to predict the number of segments required to achieve a given level of accuracy. It is well known that a large number of segments are required for accurately predicting low frequency dynamics.

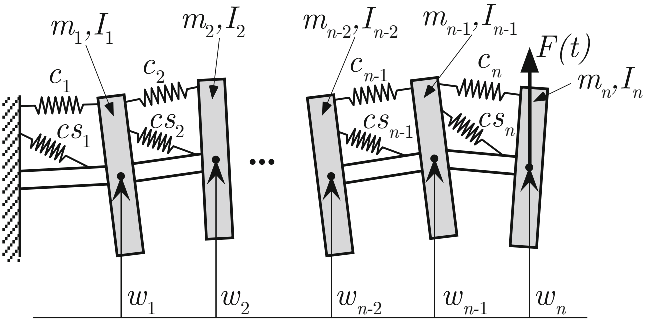


Fig. 10.3 Ideal physical model of a Timoshenko beam

For the purposes of this work the number of segments is chosen based on previous research, such that the model accurately predicts low frequency dynamics that are considered in this work [11, 12]. With the given number of segments, the physical phenomena to be included in each segment, for the model to accurately predict the dynamic behavior, will then be identified using the proposed methodology in this chapter.

For calculating the constitutive law parameters of the energy storage elements, the beam is assumed to have density ρ, Young's modulus E, shear modulus G, length L, cross sectional area A, and cross sectional moment of inertia I. Given these physical parameters of the beam, the element parameters in the above linear model are given by the expressions below:

$$\begin{aligned} m_i &= \rho A \Delta x, \quad i = 1, \ldots, n \\ I_i &= \rho I \Delta x, \\ c_i &= \tfrac{\Delta x}{EI} \\ cs_i &= \tfrac{\Delta x}{\kappa GA} \end{aligned} \tag{10.18}$$

where $\Delta x = L/n$ is the length of each segment, κ is a dimensionless constant that accounts for the non-uniform distribution of the shear stress and depends on the shape of the cross section. The inertial parameters m_i and I_i represent the linear and rotational inertia of each segment, respectively. The parameters c_i and cs_i represent the bending and shear compliance between two segments, respectively. The beam is assumed to have no energy losses therefore there are no damping elements in the model. These parameters are used to define the parameter vector as defined in (10.6).

For developing the dynamic equations, the bond graph formulation is used. Bond graphs provide the power topography of the system and it is a natural selection for implementing the power-based activity metric. The bond graph model of the ideal physical model as shown in Fig. 10.3 is developed and given in Fig. 10.4. The bond graph has $4n$ independent state variables since each segment is modeled by four independent energy storage elements and the state vector has the form $\mathbf{x} = \{p_1, \ldots, p_n, p_{I1}, \ldots, p_{In}, q_1, \ldots, q_n, q_{s1}, \ldots, q_{sn}\}^T$. The variable p represents the momentum of inertial elements and q the displacement of compliant elements. The velocity of the each mass, v_i, represents the transverse velocity at a given location of the continuous beam and (10.19) expresses the relation between the

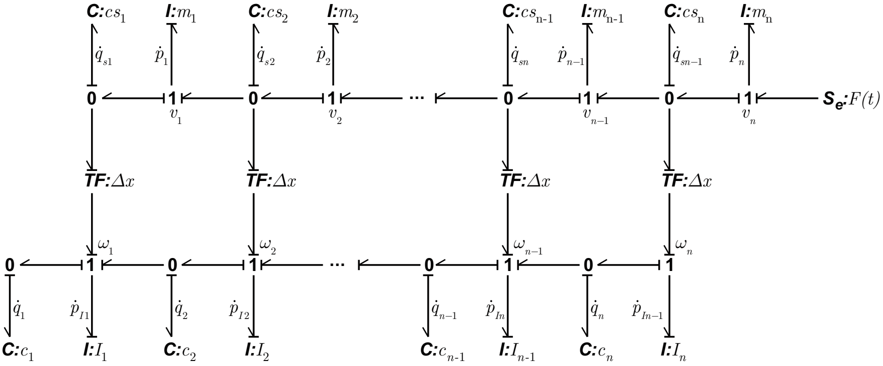


Fig. 10.4 Bond graph model of a Timoshenko beam

discrete and continuous variables. The other kinematic variable of the model, ω_i, is the rotation at a given location and its relation to the continuous variable is given in (10.20).

$$v_i(t) = \dot{w}_i = \dot{w}\left(i\Delta x, t\right) \tag{10.19}$$

$$\omega_i(t) = \dot{\varphi}_i = \dot{\varphi}\left(i\Delta x, t\right) \tag{10.20}$$

The state and output equations in matrix form are derived using the multi-port approach, which provides easy derivations of the state matrices [20]. According to this approach, most of the junction structure matrices are zero, and the state space and input matrices are given by:

$$\mathbf{A} = \mathbf{J}_{\text{SS}}\,\mathbf{S}, \quad \mathbf{b} = \mathbf{J}_{\text{SU}} \tag{10.21}$$

where $\mathbf{S} \in \mathbb{R}^{4n\times 4n}$ is a diagonal matrix with the inertial and compliant parameters of each element, and $\mathbf{J}_{\text{SS}} \in \mathbb{R}^{4n\times 4n}$, $\mathbf{J}_{\text{SU}} \in \mathbb{R}^{4n}$ are the junction structure matrices describing the interconnections between the energy elements, and they are given in the Appendix.

The output vector as define in the previous section in (10.3) becomes $\mathbf{y} = \{f_1, \ldots, f_{2n}, e_1, \ldots, e_{2n}\}^T$. Thus, the output matrices as defined in (10.4), which are required for calculating the power flow into the energy elements, are given by:

$$\mathbf{C} = \mathbf{S}, \quad \mathbf{d} = \mathbf{0} \tag{10.22}$$

The dimensions of the state space matrices as defined in the previous section are $m = 4n$ and $k = 4n$.

For the above model with n segments the steady state response is first calculated using (10.13) and based on the state space equations in (10.21) and (10.22). Then the element activity is calculated from (10.14), which gives the following expression for the energy storage elements of the model:

$$A_i^{\text{ss}}\left(\omega\right) = 2r_i U^2 Y_i^2\left(\omega\right), \quad i = 1, \ldots, 4n \tag{10.23}$$

The above analysis enables the calculation of the element activity for a given single harmonic excitation. The activity index that is used by MORA is independent of the excitation amplitude, as shown in (10.17), and therefore can be set to an arbitrary value, e.g., set to one (1) for simplicity. Model complexity and which physical phenomena need to be included, can be determined given the element activity in (10.23) and MORA. The complexity of the beam is investigated in the next section in order to identify the significant elements based on beam length and element location. A series of analysis is performed in order to get more insight into the important beam dynamics under different scenarios.

10.4 Beam Complexity Based on Activity

The activity metric and MORA is applied to a steel cantilever beam with parameters $\rho = 7860$ kg/m^3, $E = 210$ GPa, $G = 80$ GPa, $A = 3 \times 10^{-3}$ m^2, $I = 2.5 \times 10^{-6}$ m^4, $\kappa = 0.85$. The length of the beam is varied, $L = 0.2$–2.0 m, in order to study the variation of element significance. The methodology is easy and computationally inexpensive to implement due to the simple and closed form expressions used for calculating the state space matrices, frequency response, and activity.

First, the beam length is set to 2.0 m such that the beam is considered to be slender. The number of segments is set to $n = 30$ and therefore there are a total of 120 energy storage elements modeling the beam. In this case the modeling target is set to accurately predict static behavior of low frequency dynamics, thus, the excitation frequency is set to 95 % of the first natural frequency (122.7 rad/s).

The results of the activity analysis using (10.23) and under these assumptions are shown in Fig. 10.5 where the activity index of all 120 elements is shown. Element numbers 1–30 represent the activity index of the linear inertia (m_i) and 31–60 the activity index of the rotational inertia (I_i) of each segment. Next, element numbers 61–90 and 91–120 represent the activity index of the bending (c_i) and shear ($\mathrm{cs_i}$) compliance, respectively. For each range of elements the smallest numbers represent elements that are next to the fixed end of the beam. It is clear from the activity analysis that the most important elements are related to the linear inertia and the bending stiffness of the beam. On the contrary, the elements related with the rotary inertia and shear stiffness have very low activity and thus are insignificant under these conditions. This initial activity analysis agrees with common practice, in which a slender beam is modeled using the Euler–Bernoulli theory that neglects rotational inertia and shear stress effects.

Model complexity is systematically addressed using MORA as it is described in Sect. 10.2.3. Elements are ranked according to their activity index as shown in Fig. 10.6 where the sorted activity indices along with the cumulative activity index are plotted. According to the activity analysis, 40 of the 120 elements account for almost 99 % of the energy flows through the model. This is a significant result verifying that unnecessary complexity is included in the model; however, the figure does not directly depicts the elements that are insignificant and could be eliminated from the model.

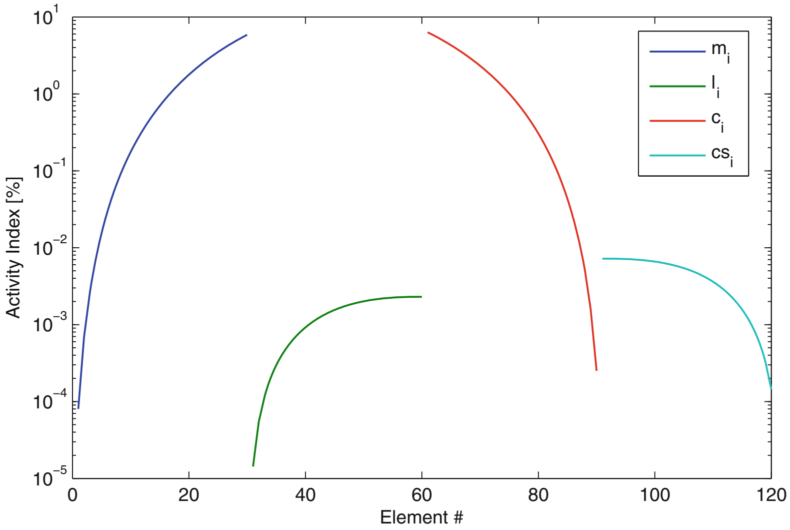


Fig. 10.5 Element activity indices for slender beam

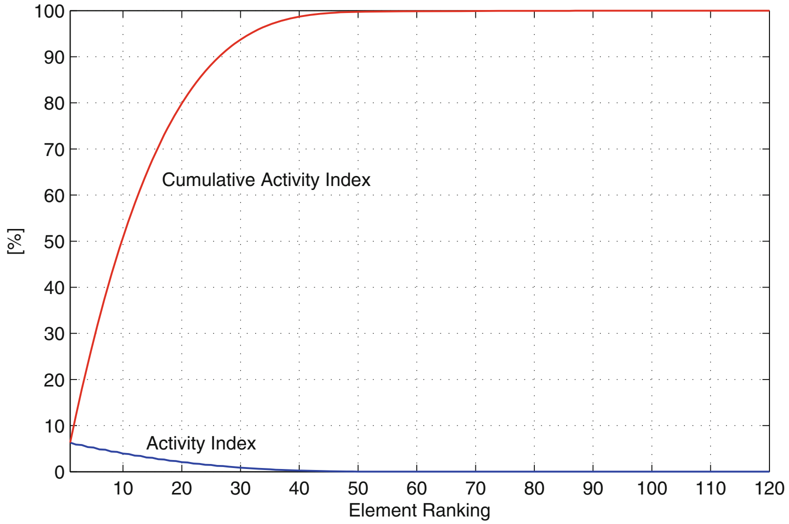


Fig. 10.6 Element ranking for slender beam

The important elements are next identified using MORA. Using a reduction threshold, $\beta = 99.5\ \%$, MORA identifies the elements that have a significant contribution to the system dynamic behavior. The results of this analysis are shown in Fig. 10.7 where both the activity and elimination/inclusion in the reduced model are depicted. The "+" symbol identifies the elements with significant contribution and must be included, where the "o" symbol identifies that an element

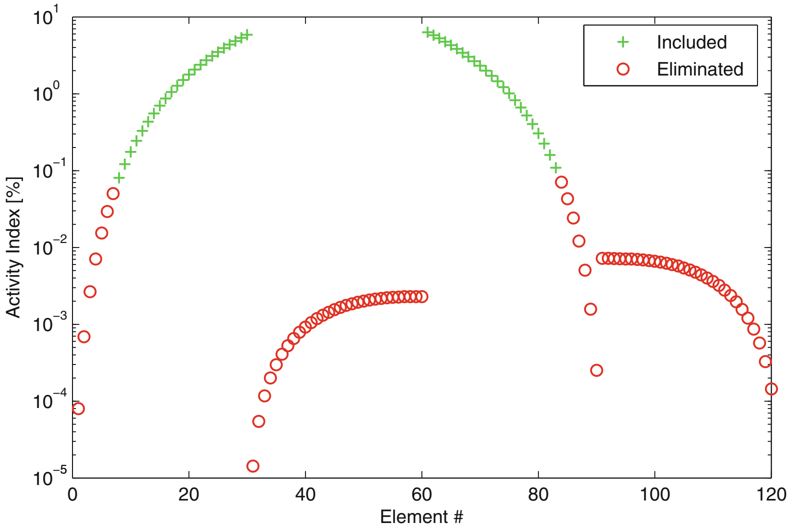


Fig. 10.7 Model reduction for slender beam, $L = 2$ m

is insignificant and must be eliminated from the full model in order to generate the reduced model. Out of the 120 elements only 46 are important and the remaining 74 can be eliminated. More specifically, MORA identifies that all rotational inertia and shear stiffness elements must be eliminated from the model. Linear inertia elements that are close to the support have low activity and can be eliminated from the model, where inertial elements towards the free end of the beam have high activity and must be retained. The reverse is true for the bending stiffness elements, where the elements towards the free end can be eliminated and the ones near the support must be retained. More specifically, 23 of the linear inertia and 23 of the bending stiffness elements have high activity and must be included in the reduced model.

The same reduction using MORA is performed with different beam lengths in order to study how element importance changes as the length is reduced. The reduction for a beam length of 0.7 m is shown in Fig. 10.8. The same trend is observed for the elimination of linear inertia and bending stiffness elements. The activity index of all rotational inertia elements (31–60) is higher than before ($L = 2$ m) but still very low, and therefore, they are eliminated from the model. The activity of shear stiffness (91–120) also increases and some of these elements become important. The shear stiffness elements that are close to the support have higher activity index and have to be included in the reduced model, while the ones towards the free end are eliminated. A total of 71 elements are included in the reduced model with 25 linear inertia, 25 bending stiffness, and 21 shear stiffness elements.

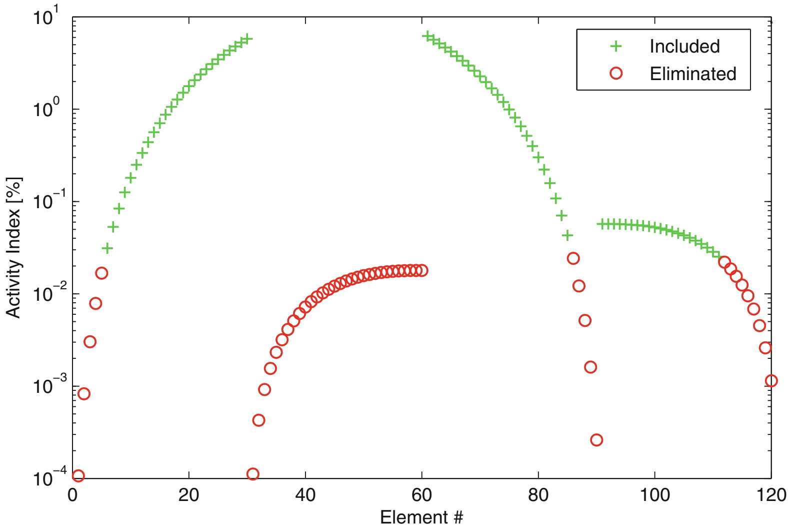


Fig. 10.8 Model reduction for $L = 0.7$ m

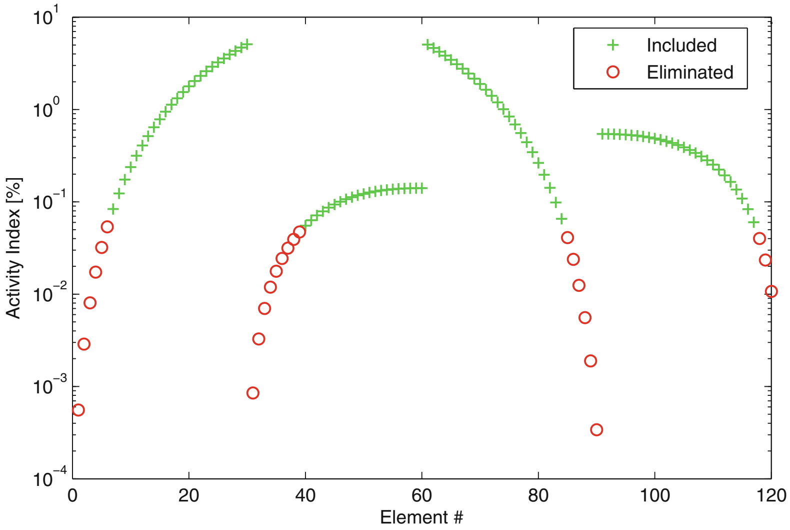


Fig. 10.9 Model reduction for $L = 0.2$ m

The beam length is further reduced to 0.2 m in order to examine if more elements become important. The activity index of the linear inertia and bending stiffness remains almost unchanged as shown in Fig. 10.9. However, the activity index of the rotational inertia and shear stiffness is further increased such that some of the rotational inertia elements also become important. More specifically the rotational

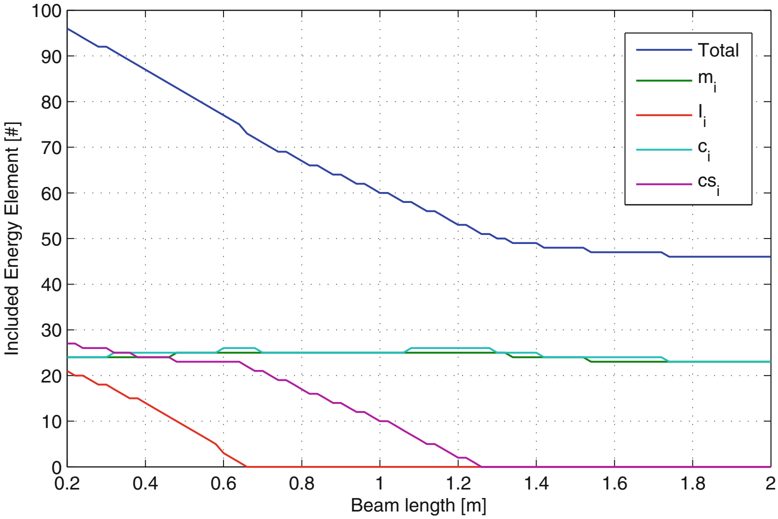


Fig. 10.10 Model reduction for length variation

inertia elements that are towards the free end are important and the ones near the fixed end are eliminated. A total of 96 elements out of 120 are included in the reduced model with 24 linear inertia, 21 rotational inertia, 24 bending stiffness, and 27 shear stiffness elements.

The variation of beam length showed that the total number of important elements increases as the beam length decreases. This variation is investigated in more detail by varying the beam length from 0.2 to 2 m with a step of 20 mm. The number of included linear and rotational inertia, and bending and shear stiffness is recorded along with the total number of elements. The results of this analysis are shown in Fig. 10.10. The total number of elements is monotonically increasing as the beam length is decreased. The number of linear inertia and bending stiffness remains almost constant as the length changes. On the contrary, the number of shear stiffness elements is zero until about 1.2 m where it becomes important and starts increasing. Further reduction in length results in a monotonic increase in the number of included shear stiffness element. A similar behavior is observed for the number of the rotational inertia elements; however, they become important at a lower beam length of about 0.6 m.

The validity of the generated reduced models is verified by analyzing the accuracy of the model. Specifically the steady state response amplitude for the velocity at the free end and the torque at the fixed end are calculated. The comparison is made with the corresponding response of the full model and over the range of beam lengths used before. The accuracy for both variables, as shown in Fig. 10.11, varies as the beam length is changed, with averages around 91.5 %. The discrete variation in accuracy is due to the change of model complexity as different elements are added or removed in the reduced model according to the activity metric.

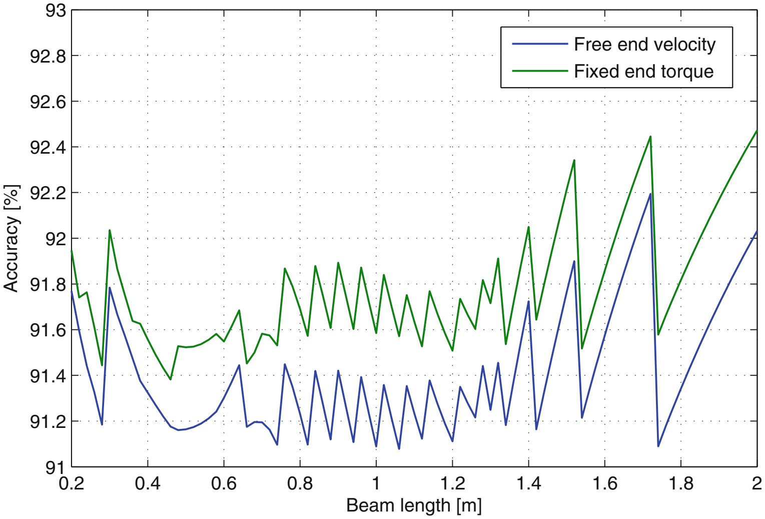


Fig. 10.11 Model accuracy

Next, a similar experiment is performed with the same conditions except the excitation frequency, which is increased to 95 % of the second natural frequency and more specifically at 722.8 rad/s. Given that the number of segments remains the same, it is expected that the number of important elements will increase. This is a known feature of finite segment models that require an increased number of segments in order to accurately predict higher frequency dynamics. Model reduction and identification of significant dynamics is performed for a slender beam ($L = 2.0$ m). The results of the activity analysis are shown in Fig. 10.12 and in general a higher number of elements are required in order to achieve a similar level of accuracy. Again, some elements at the ends of the beam can be eliminated due to their low activity. Another general observation is that low activity elements appear also within the span of the beam, in addition to the ones at the ends of the beam. The drop in activity appears around the nodes of the second mode where there is a stationary point. For example, for the linear inertia there is one very low activity element that coincides with the node of the second transverse vibration mode. Similar behavior is observed for the other energy elements as shown in Fig. 10.12. More specifically a total of 68 elements out of 120 are included in the reduced model with 27 linear inertia, two rotational inertia, 28 bending stiffness, and 11 shear stiffness elements. Similar reduction patterns, as with the excitation near the first mode, are observed for various beam lengths but not shown here for brevity.

The effect of beam length on element importance is studied next by varying the beam length ($L = 0.2 - 2.0$ m) and performing model reduction using MORA. The results of this analysis are shown in Fig. 10.13 where the number of each type of element, that needs to be included in the reduced model, is shown as a function of the beam length. The number of important linear inertia and bending stiffness elements is remains almost constant as the beam length is varied. On the contrary,

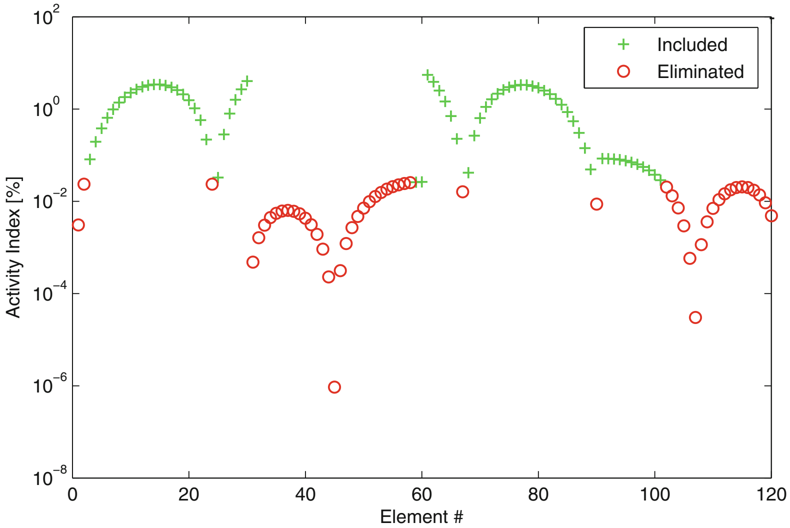


Fig. 10.12 Model reduction for slender beam, excitation near second mode, $L = 2.0$ m

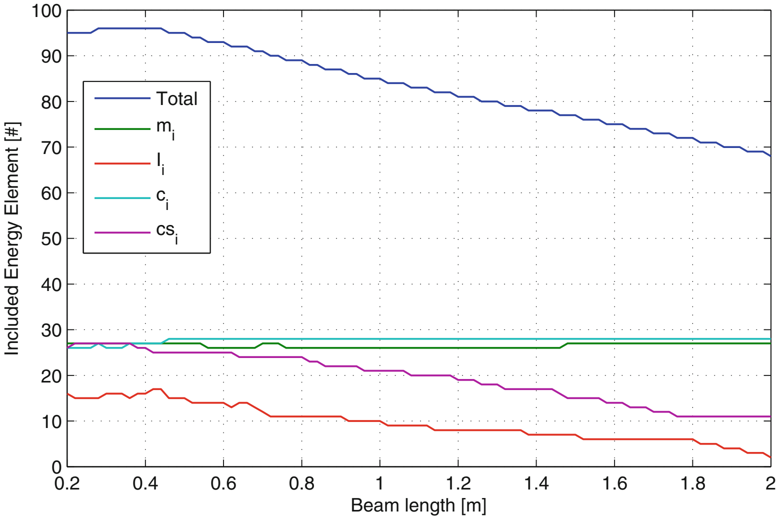


Fig. 10.13 Model reduction for length variation, excitation near second mode

the number of rotational inertia and shear stiffness elements reduces as the length of the beam increases, in accordance with the Timoshenko beam theory. However, rotational inertia and shear stiffness elements are important and have to be included in the reduced model even when the beam is slender, i.e., $L = 2.0$ m. The total number of important elements is overall higher, as compared to the total number of elements when the excitation is near the first natural frequency.

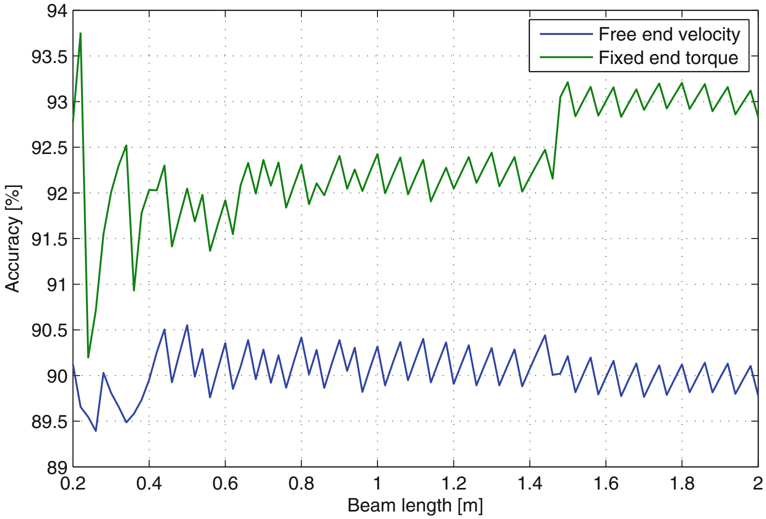


Fig. 10.14 Reduced model accuracy, excitation near second mode

The accuracy of the reduced model is also calculated for the linear velocity at the free end and the torque at the fixed end (Fig. 10.14). The comparison is made with the initial full model that includes all elements, i.e., 30 segments and 120 energy storage elements. The two variables have similar accuracy throughout the whole range of beam lengths. Also, the overall accuracy of there reduced models is similar to the accuracy of the reduced models when the excitation is near the first natural frequency.

10.5 Discussion and Conclusions

A new methodology is developed that reduces the complexity of a Timoshenko or Euler–Bernoulli beam model, by providing more insight into the beam dynamic behavior at the same time. The proposed methodology provides a systematic modeling procedure for cantilever beams that are modeled through the finite segment approach. The previously developed activity metric is used as the basis for determining the physical phenomena that need to be included in each segment in order for a reduced model to accurately predict the dynamic behavior of a beam. The procedure starts with the most complicated model, Timoshenko in this case, and then eliminates insignificant elements that do not contribute to the dynamic behavior according to the activity metric.

The results presented in this work are in agreement with the assumptions of beam theories, which suggest that the Timoshenko beam model must be used for shorter rather than slender beams. The proposed methodology can be used when modeling beams, in order to decide which of the two models to use, Timoshenko or Euler–Bernoulli. In addition, the activity metric can refine the modeling assumptions by identifying what physical phenomena need to be included in each segment, i.e., linear and rotational inertia, bending and shear stiffness. This results in a non-uniform reduced model where different physical phenomena are included along the length of the beam.

The number of segments is a significant parameter when it comes to modeling using the finite segment approach but it was considered constant in the analysis of the presented results. The methodology was also performed with various, lower and higher, number of segments; however, these results are not presented in this chapter for brevity. The reduced models for different number of segments are identical with the ones presented in this work as shown in Fig. 10.10. The only difference is the actual number of included elements; however, the ratio of included elements to the total number of elements remains the same.

The activity analysis is performed for two excitation frequencies that are around the first and second natural frequencies. These excitations are chosen by making the assumption that the model will be used with low frequency excitations. The analysis showed that a higher number of elements are needed as the excitation frequency increases; however, the overall accuracy of the reduced model remains the same. A similar analysis can be performed for even higher excitation frequencies or range of frequencies in order to account for more realistic excitations. However, this procedure has to be formalized and this remains as an item for future research.

Because this work uses an energy-based modeling metric, it is convenient to use a model representation and formulation approach from which energy can be easily extracted/calculated. The bond graph approach explicitly represents the power topography of a dynamic system, and therefore, it is used in this work for calculating the necessary variables required for the power calculations. To be clear, the use of this methodology is not limited to systems represented by bond graphs. It can also be applied when the continuous system is modeled using any other modeling methodology, e.g., Lagrange's equations, Newton's Law, etc. However, in this case the calculation of power that is required for the proposed methodology might not be as trivial as using the bond graph formulation.

The activity metric effectively addresses the model complexity of distributed parameter components and in addition provides physical insight into the model. The results of this chapter provide more insight into the nature of the reduced models produced by MORA, and therefore, demonstrate that MORA is an even more useful tool than previously realized for the production of proper models of nonlinear systems.

Appendix: Junction Structure Matrices

$$\mathbf{S} = \begin{bmatrix} m_i\,\mathbf{I}_{n\times n} & \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} \\ \mathbf{0}_{n\times n} & I_i\,\mathbf{I}_{n\times n} & \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} \\ \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} & c_i\,\mathbf{I}_{n\times n} & \mathbf{0}_{n\times n} \\ \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} & \mathrm{cs}_i\,\mathbf{I}_{n\times n} \end{bmatrix}^{-1}$$

$$\mathbf{J}_{SS} = \begin{bmatrix} \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} & \mathbf{J}_1 \\ \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} & \mathbf{J}_1 & \Delta x\,\mathbf{I}_{n\times n} \\ \mathbf{0}_{n\times n} & -\mathbf{J}_1^T & \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} \\ -\,\mathbf{J}_1^T & -\Delta x\,\mathbf{I}_{n\times n} & \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} \end{bmatrix}$$

$$\mathbf{J}_{SU} = \begin{Bmatrix} \mathbf{0}_{(n-1)\times 1} \\ 1 \\ \mathbf{0}_{3n\times 1} \end{Bmatrix}$$

$$\mathbf{J}_1 = \begin{bmatrix} -1 & 1 & 0 & 0 & \ldots & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 & -1 \end{bmatrix}$$

References

1. Bauchau, O. A., & Craig, J. I. (2009). *Structural analysis*. Berlin: Springer. ISBN 978-90-481-2515-9.
2. Borutzky, W. (2004). *Bond graph methodology: Development and analysis of multidisciplinary dynamic systems*. Berlin: Springer. ISBN 978–1848828810.
3. Brown, F. T. (2006). *Engineering system dynamics: A unified graph-centered approach* (2nd ed.). Boca Raton, FL: CRC Press. ISBN 9780849396489.
4. Ferris, J. B., & Stein, J. L. (1995). Development of proper models of hybrid systems: A bond graph formulation. *Proceedings of the 1995 International Conference on Bond Graph Modeling* (pp. 43–48), January, Las Vegas, NV. Published by SCS, ISBN 1-56555-037-4, San Diego, CA.

5. Ferris, J. B., Stein, J. L., & Bernitsas, M. M. (1998). Development of proper models of hybrid systems. *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control, 120*(3), 328–333.
6. Genta, G. (2009). *Vibration dynamics and control*. Dordrecht: Springer. ISBN 978-0-387-79579-9.
7. Karnopp, D. C., Margolis, D. L., & Rosenberg, R. C. (2006). *System dynamics: Modeling and simulation of mechatronic systems* (4th ed.). New York: Wiley. ISBN 978-0-471-70965-7.
8. Li, X. F. (2008). A unified approach for analyzing static and dynamic behaviors of functionally graded Timoshenko and Euler–Bernoulli beams. *Journal of Sound and Vibration, 318*(4–5), 1210–1229.
9. Li, D. F., & Gunter, E. J. (1981). Study of the modal truncation error in the component mode analysis of a dual-rotor. *Transactions of the ASME, Journal of Engineering for Gas Turbines and Power, 104*(3), 525–532.
10. Liu, D.-C., Chung, H.-L., & Chang, W.-M. (2000). Errors caused by modal truncation in structure dynamic analysis. *Proceedings of the International Modal Analysis Conference—IMAC*, v 2 (pp. 1455–1460). Bethel, CT: Published by Society for Experimental Mechanics Inc., ISSN 1046–6770.
11. Louca, L. S. (2014). Complexity of distributed parameter bond graph models. *Proceedings of the 11th International Conference on Bond Graph Modeling and Simulation—ICGBM'2012* (pp. 789–797), Monterey, CA, USA. San Diego, CA: Published by the Society for Computer Simulation International.
12. Louca, L. S. (2015). Finite segment model complexity of an Euler-Bernoulli beam. *Proceedings of the 8th International Conference on Mathematical Modeling* (pp. 334–340). Vienna, Austria: Published by the International Federation of Automatic Control-IFAC.
13. Louca, L. S., Rideout, D. G., Stein, J. L., & Hulbert, G. M. (2004). Generating proper dynamic models for truck mobility and handling. *International Journal of Heavy Vehicle Systems (Special Issue on Advances in Ground Vehicle Simulation), 11*(3/4), 209–236. Published by Inderscience Enterprises Ltd., ISSN 1744−232X, St. Helier, United Kingdom.
14. Louca, L. S., & Stein, J. L. (2002). Ideal physical element representation from reduced bond graphs. *Journal of Systems and Control Engineering, 216*(1), 73–83. Published by the Professional Engineering Publishing, ISSN 0959–6518, Suffolk, United Kingdom.
15. Louca, L. S., & Stein, J. L. (2009). Energy-based model reduction of linear systems. *Proceedings of the 6th International Symposium on Mathematical Modeling*, Vienna, Austria. Published in the series ARGESIM-Reports No. 35, ISBN 978-3-901608-35-3, Vienna, Austria.
16. Louca, L. S., Stein, J. L., Hulbert, G. M., & Sprague, J. K. (1997). Proper model generation: An energy-based methodology. *Proceedings of the 1997 International Conference on Bond Graph Modeling* (pp. 44–49), Phoenix, AZ. Published by SCS, ISBN 1-56555-103-6, San Diego, CA.
17. Louca, L. S., Stein, J. L., & Hulbert, G. M. (2010). Energy-based model reduction methodology for automated modeling. *Journal of Dynamic Systems Measurement and Control, 132*(6), 061202 (16 pages). Published by the American Society of Mechanical Engineers, ISSN Print 0022–0434, ISSN Online 1528–9028, New York, NY.
18. Meirovitch, L. (1967). *Analytical methods in vibrations*. New York, NY: Macmillan Publishing Inc.
19. Rideout, D. G., Stein, J. L., & Louca, L. S. (2007). Systematic identification of decoupling in dynamic system models. *Journal of Dynamic Systems Measurement and Control, 129*(4), 503–513. Published by the American Society of Mechanical Engineers, ISSN Print 0022–0434, ISSN Online 1528–9028, New York, NY.
20. Rosenberg, R. C. (1971). State-space formulation for bond graph models of multiport systems. *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control, 93*, 36–40.
21. Rosenberg, R. C., & Karnopp, D. C. (1983). *Introduction to physical system dynamics*. New York: McGraw-Hill. ISBN 0070539057.
22. Stein, J. L., & Louca, L. S. (1996). A template-based modeling approach for system design: Theory and implementation. *Transactions of the Society for Computer Simulation International*. San Diego, CA: Society for Computer Simulation International. Published by SCS, ISSN 0740-6797/96.

23. van Rensburg, N. F. J., & van der Merwe, A. J. (2006). Natural frequencies and modes of a Timoshenko beam. *Wave Motion, 44*(1), 58–69.
24. Walker, D. G., Stein, J. L., & Ulsoy, A. G. (1996). An input–output criterion for linear model deduction. *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control, 122*(3), 507–513.
25. Wilson, B. H., & Stein, J. L. (1995). An algorithm for obtaining proper models of distributed and discrete systems. *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control, 117*(4), 534–540.

Part IV
Applications

The chapters in Part IV demonstrate that bond graphs are well suited for modelling and analysis of a wide range of mechatronic systems that are of current interest and of importance in their field of application.

The first chapter in this part presents a multibond graph modelling and simulation case study of a vibration absorber system in a helicopter. The subject is interesting and important from a practical, industrial as well as from a bond graph modelling point of view. Vibrations transmitted from the rotor to the fuselage affect the flight, the comfort of the crew and in the long run cause fatigue of the mechanical parts. In the helicopter considered in this study, a reduction of the transmitted vibrations is achieved by a semi-active suspension system between the helicopter's main gear box and the fuselage that creates an anti-resonance effect.

From a modelling point of view, the considered vibration absorber system is of interest as it is a complex 3D-multibody system with 14 rigid bodies, 18 kinematic joints of different type and four *closed* kinematic chains. For practitioners in industrial projects it might be interesting to see that for a complex 3D-multibody system an overall BG model can be developed in a systematic and structured approach at a *graphical level* provided that BG library models have been developed for a rigid body and various standard types of joints. BG modelling and simulation software can automatically derive the equations of motion from a hierarchical BG of a complex mechatronic system and solve them numerically. It is well known that the way joints are modelled will have an effect on the form of the generated mathematical model and on the problems that may occur with its numerical computation. These aspects are also addressed in Chap. 11 which is based on a PhD thesis of the first author that was supervised by the second and third author.

In an aging population there is an increasing demand for various types of wheelchairs not only in hospitals and homes for the elderly but in general for mobility disabled people to allow for independence. The case study in Chap. 12 analyses how an auxiliary electric power system fitted to a manually operated wheelchair can improve its mobility with regard to a given trajectory and time. The occupant of a wheelchair as well as an aid walking behind the chair will benefit from a cost-effective design of a wheelchair equipped with an additional electric drive module. That is, the subject of this study is a clearly relevant one and bond graphs are well suited for modeling such a mechatronic system.

Chapter 12 presents the step-by-step development of a causal bond graph model of the motorised wheelchair from which state-space equations are derived and numerically computed by a block-diagram based simulation program. Subject of the simulation study is the accuracy of the motion along a prescribed trajectory and its speed.

The subject of Chap. 13 is also interesting and of great practical importance. The design of a miniaturised robot for in-vivo biopsy, its manufacturing and its test can be considered a significant contribution to an improvement of *minimal invasive* techniques in surgery that clearly offers a number of promising advantages such as a reduction of patients' discomfort and a facilitation of endoscopy tasks. The robot considered in this chapter is a four degree-of-freedom, miniaturised externally controlled wire actuated in-vivo robot equipped with articulated links, a light source and a camera which may provide improved visibility and better maneuverability. A force sensor and a force control in conjunction with a trajectory control can help avoid the damage of healthy tissue by limiting the interaction force between the robot's tip and the tissue.

The robot is a tree-like multibody systems. Its study comprises the analysis of its forward kinematic, the development of a scalar bond graph, an inverse kinematic analysis essential for placing the surgical robot tip at the desired point of interest with minimum error, the design of a trajectory and a force control, and experiments for the validation of simulation results. The chapter is based on a recently successfully defended PhD thesis of the first author supervised by the second author.

Chapter 14 studies a completely different robot for a different application, a biologically inspired walking machine with four compliant legs meant to operate autonomously off-road in remote, hazardous rough terrain that is inaccessible to wheeled mobile robots. As the studied robot has got legs, its locomotion requires a coordinated actuation of the legs to ensure a required gait pattern and to make sure that the robot moves along a given trajectory. Each of the four legs has got a hip joint and a knee joint that must be operated in a coordinated way by controlling the joint motors. The lower link of the legs is equipped with a spring to improve the robot's locomotion. The practical importance of this application is also evident and bond graphs are well suited for modelling the controlled robot.

The research presented in Chap. 14 comprises the development of a three dimensional dynamic bond graph model, an experimental setup, a robot control in workspace, a posture control, and strategies for sensor and actuator failures. Simulation experiments indicate that a robot with compliant legs performs more energy efficient in trot gait than a robot with rigid legs.

According to their task, mobile robots should be able to react to a failure in the presence of limited or no hardware redundancy. As the robot is a legged one, two types of essential component failures are considered, namely a locked joint restricting the workspace of a leg tip and a joint that does not react to the torque of its motor. Chapter 14 is based on the PhD thesis of the first author supervised by the second author.

Renewable energy sources is another application area that is highly relevant and important from an ecological as well as from an economic point of view. Wind turbines are complex controlled mechatronic systems composed of mechanical components such as the blades, a gear box, an induction motor, and a power electronic converter. The blades are to be considered distributed parameter systems with aerodynamic wind forces acting on them. Chapter 15 studies a variable speed wind turbine with torque control and power control. The blades are considered cantilever beams and are modelled by means of C fields and R fields. Simulations of the dynamic behaviour of the overall systems carried out by means of the 20sim® software consider a constant as well as a varying wind profile with real wind data. A major contribution of Chap. 14 is that the study of such a complex systems accounts for all its components.

The last chapter in Part IV introduces the bond graph modelling of biomolecular systems being part of living organisms. Some researchers have already used bond graphs in the past for the modelling of chemical reactions. Their application to biomolecular systems, however, is, to the Editors knowledge, still quite new. As living systems are neither at thermodynamic equilibrium nor closed, a bond graph approach to the modelling of biomolecular systems is not evident. Therefore, Chap. 16 first recalls some thermodynamic basics, then introduces appropriate bond graph covariables, a nonlinear C storage element for representing molecular species and a non-linear R element for reactions. The chapter considers a number of examples. Furthermore, causal analysis is used to examine the properties of the junction structure of the bond graphs set up for biomolecular systems. It is shown how causal properties of the BG junction structure are related to results of stoichiometric analysis of biomolecular systems. By this way, new insights are provided into the dynamics of biomolecular systems.

Chapter 11
Bond Graph Modeling and Simulation of a Vibration Absorber System in Helicopters

Benjamin Boudon, François Malburet, and Jean-Claude Carmona

Notation

Subscripts

i	Relative to the i loop ($i = 1 \dots 4$)
j	Relative to the body j ($j =$ MGB, SB$_i$, MB$_i$, F)
MGB	Relative to the body MGB
SB$_i$	Relative to the body SARIB bar i
MB$_i$	Relative to the body MGB bar i
F	Relative to the body fuselage

Mechanical Notation

Generality

$\overrightarrow{MN}^{h}$	Vector associated with the bipoint (MN) expressed in the R_h frame
$\overrightarrow{P}^{\,0}_{pes\rightarrow j}$	Weight vector of body j expressed in the inertial reference frame R_0

B. Boudon (✉)
Aix-Marseille Université, CNRS, ISM UMR 7287,
13288, Marseille Cedex 09, France
e-mail: benjamin.boudon@univ-amu.fr

F. Malburet • J.-C. Carmona
Arts-et-Métiers ParisTech, CNRS, LSIS, 2 Cours des Arts-et-Métiers,
13617, Aix-en-Provence, France

W. Borutzky (ed.), *Bond Graphs for Modelling, Control and Fault Diagnosis of Engineering Systems*, DOI 10.1007/978-3-319-47434-2_11

$\overrightarrow{\Omega}(j/0)^j$	Angular velocity vector of body *j* with regard to the inertial reference frame expressed in the frame R_j
$\overrightarrow{V}(M/R_0)^j$	Linear absolute velocity (in regard to the inertial frame) of point *M* expressed in the frame R_j
$\overrightarrow{V}(M \in j/k)^j$	Linear relative velocity vector of point *M* of body *j* with regard to the body k expressed in the frame R_j
P_0^j	Transformation matrix from the inertial reference frame R_0 to the frame R_j
$\tau\left(\text{ext} \rightarrow S_j\right)$	External mechanical wrench applied to body *j*

Model Minutiae

G_j	Body *j's* center of mass
O_{FF}	Origin of fixed frame
A_i	Point located on the revolute joint between the fuselage and the SARIB beater *i*
B_i	Center of the spherical joint between an SARIB beater *i* and a MGB bar *i*
C_i	Center of the spherical joint between an MGB bar *i* and the MGB
A_{MGB}	Point located on the MGB on the prismatic joint with the fixed frame

Reference Frames

$R_0 = \left(O_0, \overrightarrow{x}_0, \overrightarrow{y}_0, \overrightarrow{z}_0\right)$	Inertial reference frame (or Galilean frame)
$R_j = \left(O_j, \overrightarrow{x}_j, \overrightarrow{y}_j, \overrightarrow{z}_j\right)$	Local reference frame of body *j*
$R_{\text{FF}} = \left(O_{\text{FF}}, \overrightarrow{x}_{\text{FF}}, \overrightarrow{y}_{\text{FF}}, \overrightarrow{z}_{\text{FF}}\right)$	Local reference frame ($R_{\text{FF}} = R_0$) attached to the fixed frame
$R_{\text{MGB}} = \left(G_{\text{MGB}}, \overrightarrow{x}_{\text{MGB}}, \overrightarrow{y}_{\text{MGB}}, \overrightarrow{z}_{\text{MGB}}\right)$	Local reference frame attached to the MGB
$R_F = \left(G_F, \overrightarrow{x}_F, \overrightarrow{y}_F, \overrightarrow{z}_F\right)$	Local reference frame attached to the fuselage
$R_{F\ \text{int}_i} = \left(A_i, \overrightarrow{x}_{F\ \text{int}_i}, \overrightarrow{y}_{F\ \text{int}_i}, \overrightarrow{z}_{F\ \text{int}_i}\right)$	Intermediate local reference frame attached to fuselage so as to facilitate the definition of the axis of the revolute joints
$R_{\text{SB}} = \left(G_{\text{SB}}, \overrightarrow{x}_{\text{SB}}, \overrightarrow{y}_{\text{SB}}, \overrightarrow{z}_{\text{SB}}\right)$	Local reference frame attached to the SARIB Beaters
$R_{\text{MB}} = \left(G_{\text{MB}}, \overrightarrow{x}_{\text{MB}}, \overrightarrow{y}_{\text{MB}}, \overrightarrow{z}_{\text{MB}}\right)$	Local reference frame attached to the MGB Bar

Model Variables

x_j, y_j, z_j Positional parameters of body *j*'s center of mass (m)

$\alpha_j, \beta_j, \gamma_j$ Angular parameters of the body *j* with regard to the inertial reference frame (rad)

Model Parameters

Ω Rotational velocity of the main rotor (rad.s^{-1})

b Number of rotor's blades (adim)

m_j Body *j*'s mass (kg)

I_{Gj} Body *j*'s Inertia matrix (kg m^2)

11.1 Introduction

In the 1990s, thanks to the multibond graph formalism (an extension of bond graphs where the scalar power bonds become vectors bonds and the elements multiports), the bond graph was extended to the study of multibody systems with three dimensions [7, 29]. Nevertheless, few complex multibody systems with kinematic closed loops have been simulated. In the last 20 years, computer science and software dedicated to bond graphs such as 20-sim software have progressed considerably and contributed greatly to the development of bond graphs [1]. First, the graphical aspect of bond graphs can be fully exploited. Indeed, friendly environments enable the entering, modifying, or interpreting of bond graphs. Secondly, the automatic generation of equations from a bond graph enables engineers to avoid solving equations. Consequently, this step is less cumbersome and prone to errors. Moreover, the simulation software mentioned is now equipped with efficient numerical solvers.

This chapter presents a bond graph model and the associated simulations of a complex multibody system: a vibration absorber system in helicopters. The system model is a complex multibody system (according to the definitions given in [28]) because of the numerous bodies and joints and the presence of several kinematic loops.

The operation of a helicopter's rotor causes important vibration levels affecting, namely the flight handling, the fatigue of the mechanical parts, and the crew's comfort. The considered vibrations created by the aerodynamic and inertia forces acting on the rotor excite the main gearbox then the fuselage at a specific frequency $b\Omega$ where b is the number of blades of the main rotor and Ω the rotational velocity of the main rotor. Suspensions between the main gear box (MGB) and the fuselage help to filter these problematic vibrations. Different passive technical solutions exist for the completion of this joint. The role of the studied vibration absorber system is to reduce the efforts transmitted from the rotor to the fuselage at the $b\Omega$ frequency by creating an anti-resonance phenomenon. The principle of this system will be described in Sect. 11.4.

The design and the analysis of such complex systems are usually conducted with analytical methods based on physical equations or signal-flow methods based on transfer functions written with block diagrams form. Unfortunately, these two classical approaches may cause a loss of the physical sense and the visibility of the modeling assumptions [2, 33]. Moreover, reusing models and taking into account the increasing complexity can be cumbersome and prone to errors because of the need of manual transformations so as to build physics-based model libraries with block diagrams [33]. In this context, we present a bond graph (BG) approach [23] that permits a structural and modular approach of a complex mechatronic system. These well-known features [5, 27]: graphic, object-oriented, multiphysic, and acausal can be exploited for this class of multibody system with embedded electronics. Mainly with its graphical nature, the bond graph brings a more global view and comprehension of complex mechatronic systems which lead to more sustainable solutions. Based on oriented-object and acausal features, bond graphs also permit a modular approach which allows a better knowledge capitalization to better store and reuse your modeling works. Moreover, it would facilitate the automation of the modeling task.

The objective of this chapter is to model and simulate a helicopter's vibration absorber system with bond graphs. Particular attention will be given in showing the interest of the bond graph tool relative to more conventional tools.

This chapter is organized as follows. In Sect. 11.2, we review the modeling and simulation methods of multibody systems based on a bond graph approach. Section 11.3 is dedicated to the presentation of the vibration absorber system. Section 11.4 starts by describing the kinematic structure and the mechanical assumptions retained for the suspension model. After, the modeling and simulation framework is applied and the construction of the bond graph model of the vibration absorber system detailed. Simulation results will be presented in Sect. 11.5. Finally, in the last section, the conclusion will be given.

11.2 BG Modeling and Simulation of Multibody Systems

11.2.1 Brief Review on the BG Modeling of Multibody Systems

The aim of this section is to recall the main contributors concerning methodologies for modeling the dynamics of three-dimensional multibody systems (MBS). More detailed reviews specifying applications of BG modeling for MBS can be found in [5, 13, 27].

The first bond graph models of multibody systems have been proposed by D.C. Karnopp and R.C. Rosenberg [16, 26], thanks to an analytical approach based on an appropriate choice of generalized coordinates, the construction of junction structure for the formulation of the kinematic laws, and a Lagrangian formulation.

In the 1990s, thanks to the multibond graph (MBG) formalism [4, 9] (an extension of bond graphs where the scalar power bonds become vector bonds and the elements multiports), the application fields of the bond graph were extended to the study of multibody systems with three-dimensional movements.

The bond graph approach used for multibody systems was introduced by A. M. Bos [7, 29]. In his PhD, he developed bond graph models for three-dimensional multibody systems and discussed how to derive the equations of motion from the bond graph in several forms. At this time, although he managed to conduct simulations of a 3D motorcycle, the equations had been derived with a manual process.

Library models for a rigid body and for various types of joints have been provided in [34] so that bond graph models of rigid multibody systems can be assembled in a systematic manner.

Felez [14] developed a software that helps with modeling multibody systems using bond graphs. To handle derivative causalities with this software, he proposes a way to introduce Lagrange multipliers into the system so as to eliminate derivative causality.

In [30, 31], different methods for simulating BG models have been presented. These simulations have been conducted with a predecessor of 20-sim software and numerically compared between themselves mainly between the computing time and accuracy. Even if the possibility of using multibond graphs was evocated, the difficulty of implementing bond graphs with vector bonds was not mentioned. This point will be detailed in Sect. 11.2.4.

11.2.2 Approach Chosen: The Tiernego and Bos Method with a 3D Model and Vector Bonds

Bond graph construction based on multibody dynamic equations can be established either with the Newton–Euler equations or by using Lagrange equations. Depending on the starting point, several bond graph construction methods have been developed:

- the “Tiernego and Bos” method from the application of the Newton–Euler equations,
- the “Karnopp and Rosenberg” method from the application of the Lagrange equations.

In order to keep a modular approach, the more appropriated method for modeling multibody systems with bond graphs is the Bos and Tiernego method [29].

This method enables a multibody system to be built in bond graphs as an assembly of bodies and joints. The principle of this method is based on the use of absolute coordinates systems and Newton–Euler equations. Indeed, in such a way, the dynamic equations of a rigid body depend on, obviously, its mass/inertia parameters and also on geometric parameters that are defined only for the considered body. The kinematic joints impose a set of constraints on the effort and flow vectors of the articulation points involved in the assembly of two bodies so that the desired relative motion can be achieved. Consequently, the dynamic equations of the complete system consist of the dynamic equations of each body depending only on its own parameters and the constraint equations of each joint.

From the authors' point of view, a good approach, in the first preliminary design stage and still, in the frame of a modular approach, is to use in a more general way: a 3D model and vector bonds so as to obtain a synthetic model. The use of vector bond graphs (also called multibond graphs) requires multibond graph elements (multibond or vector bond, junctions, multiport energy storage elements, multiport transformers, and gyrators) which are supposed to be known and directly used in this paper. Readers can refer to [13] to find the details of the modeling of those elements.

11.2.3 Rigid Body Modeling

Let us remember the architecture of a rigid body multibond graph model based on [5, 7, 20, 22].

This bond graph architecture is based on the Newton–Euler equations (Eqs. (11.1) and (11.2)) with the inertia matrix (modeled with a multiport energy store element $\left[\mathrm{I}_{S_j,G_j}\right]_j$ in the upper part) associated with gyroscopic terms, respectively (modeled with a multiport gyrator element also called Eulerian Junction Structure about mass-center of body j expressed in its frame $\left[\mathrm{EJS}_{G_j}\right]_j$ and the mass matrix modeled with a multiport energy store element $[\mathrm{m}_j]_0$ in the lower part).

$$\sum_h \overrightarrow{F}_{h\to j}^{\;0} + \overrightarrow{P}_{\mathrm{pes}\to j}^{\;0} = m_j \frac{\mathrm{d}}{\mathrm{d}t}\left(\overrightarrow{V}\left(G_j/R_0\right)^0\right)_0 \tag{11.1}$$

$$\sum_h \overrightarrow{M}_G^{\;j} = [I_{G_j}]^j \left[\frac{\mathrm{d}}{\mathrm{d}t}\left(\overrightarrow{\Omega}(j/0)^j\right)_0\right]^j + \overrightarrow{\Omega}(j/0)^j \wedge \left([I_{G_j}]^j \overrightarrow{\Omega}(j/0)^j\right) \tag{11.2}$$

The upper part of the bond graph represents the rotational dynamic part expressed in the body frame while the lower part is for the translational dynamic part expressed in the inertial reference frame (or Galilean frame). Notice that the power bonds corresponding to the rotational quantities are in purple while the power bonds corresponding to the translational quantities are in green. The two corresponding 1-junctions arrays correspond, respectively, to the angular velocity vector of body *j* with regard to the inertial frame $\overrightarrow{\Omega}(j/0)^j$ and the translational velocity vector of body *j's* center of mass in regard to the inertial frame $\overrightarrow{V}\left(G_j/R_0\right)^0$ expressed in these two coordinate frames.

The central part of the MBG describes the kinematic relations (Eq. (11.3)) between the velocities of the two points of the body *j* ($\overrightarrow{V}(M_k/R_0)^i$ and $\overrightarrow{V}(M_l/R_0)^0$) and the velocity of the center of mass $\overrightarrow{V}\left(G_j/R_0\right)^j$ resulting from the formula of the rigid body.

$$\begin{array}{l}\overrightarrow{V}(M_k/R_0)^0 = \overrightarrow{V}\left(G_j/R_0\right)^0 + \overrightarrow{\Omega}\left(S_j/0\right)^0 \wedge \overrightarrow{G_jM_k}^0 \\ \overrightarrow{V}(M_l/R_0)^0 = \overrightarrow{V}\left(G_j/R_0\right)^0 + \overrightarrow{\Omega}\left(S_j/0\right)^0 \wedge \overrightarrow{G_jM_l}^0\end{array} \tag{11.3}$$

As the translational dynamic is expressed in the inertial reference frame, a modulated transformation element (MTF) between $\overrightarrow{V}\left(G_j/R_0\right)^j$ and $\overrightarrow{V}(G_i/R_0)^0$ permits the coordinate transformation (Eq. (11.4)) between the body frame R_j and the inertial frame R$_0$ (Fig. 11.1).

$$\overrightarrow{V\left(G_j/R_0\right)}^0 = P_0^j.\overrightarrow{V\left(G_j/R_0\right)}^j \tag{11.4}$$

The rotation matrix P_0^j can be calculated from Cardan angles. In this paper, the XYZ Cardan angles have been employed. The angular velocity components of the considered body expressed in the body frame (called the pseudo-velocities or Cardan angles rates of changes) are used to determine the body's orientation and the corresponding coordinate transformation matrix. This classical process used in the Cardan block is detailed in Fig. 11.2. It should be noted that the initial conditions used for the integration of time derivatives of the Cardan angles must be consistent with regard to the kinematic constraints.

11.2.4 Main Difficulties Linked to the Approach Chosen

11.2.4.1 About Causalities Imposed by Vector Bonds

The first step in carrying out a simulation is the assignment of causalities. Vector bonds create some causality constraints that will force the modeler to choose the right method. Indeed, as we can find in [3, 12], two causality constraints appear with vector bonds.

The first causality constraint (C1) is: *each component of a vector bond must have the same causality*. Consequently, it is not possible to constrain the motion using Sf elements in only one or two dimensions without introducing some parasitic elements into the remaining unconstrained dimensions(s).

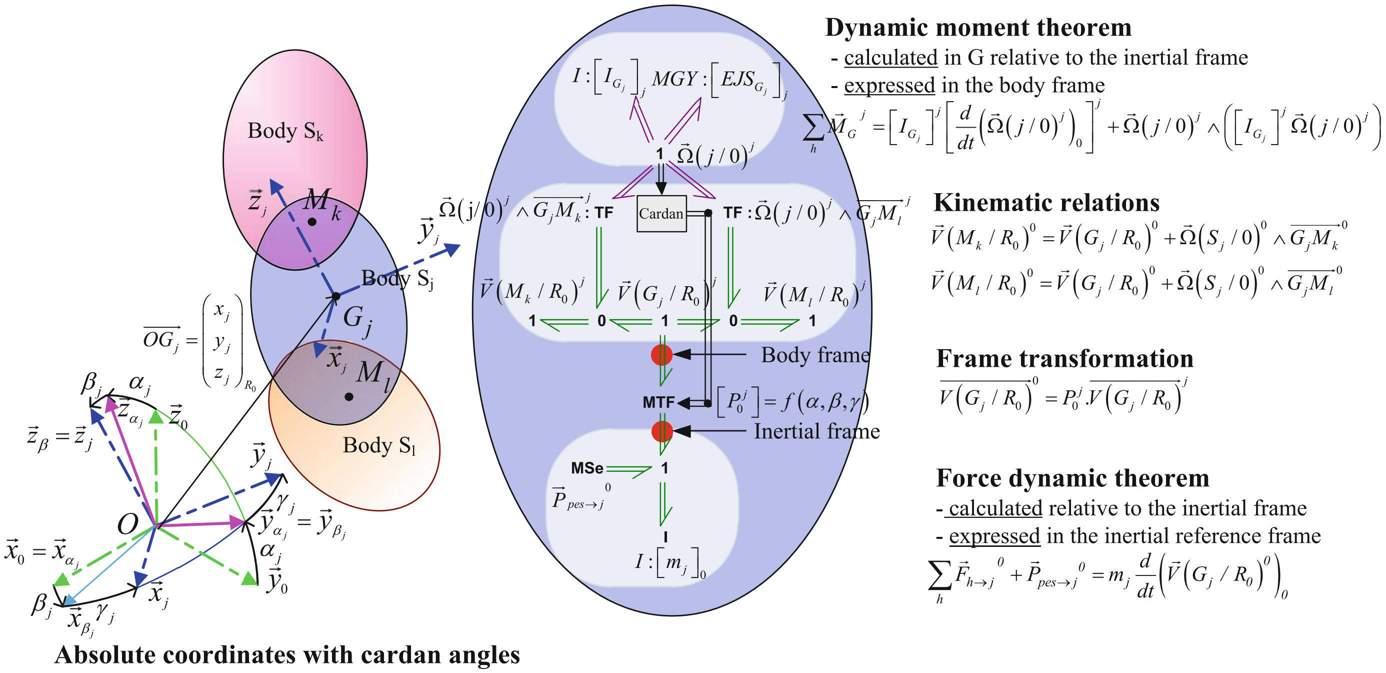
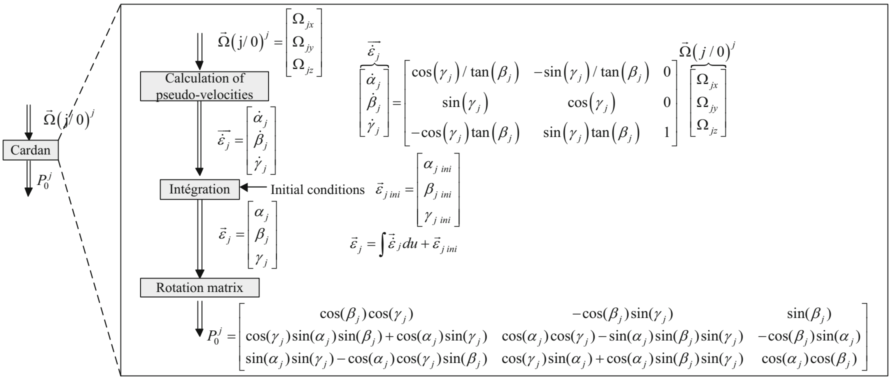

Dynamic moment theorem
- calculated in G relative to the inertial frame
- expressed in the body frame
$\sum_h \vec{M}_G{}^j = \left[I_{G_j}\right]^j \left[\frac{d}{dt}\left(\vec{\Omega}(j/0)^j\right)_0\right]^j + \vec{\Omega}(j/0)^j \wedge \left(\left[I_{G_j}\right]^j \vec{\Omega}(j/0)^j\right)$
Kinematic relations
$\vec{V}(M_k/R_0)^0 = \vec{V}(G_j/R_0)^0 + \vec{\Omega}(S_j/0)^0 \wedge \overrightarrow{G_jM_k}^0$
$\vec{V}(M_l/R_0)^0 = \vec{V}(G_j/R_0)^0 + \vec{\Omega}(S_j/0)^0 \wedge \overrightarrow{G_jM_l}^0$
Frame transformation
$\overrightarrow{V(G_j/R_0)}^0 = P_0^j . \overrightarrow{V(G_j/R_0)}^j$
Force dynamic theorem
- calculated relative to the inertial frame
- expressed in the inertial reference frame
$\sum_h \vec{F}_{h\to j}{}^0 + \vec{P}_{pes\to j}{}^0 = m_j \frac{d}{dt}\left(\vec{V}(G_j/R_0)^0\right)_0$
$I:\left[I_{G_j}\right]_j$
$MGY:\left[EJS_{G_j}\right]_j$
1
$\vec{\Omega}(j/0)^j$
$\vec{\Omega}(j/0)^j \wedge \overrightarrow{G_jM_k}^j$: TF
Cardan
TF : $\vec{\Omega}(j/0)^j \wedge \overrightarrow{G_jM_l}^j$
$\vec{V}(M_k/R_0)^j$
$\vec{V}(G_j/R_0)^j$
$\vec{V}(M_l/R_0)^j$
1
0
1
0
1
Body frame
MTF
$\left[P_0^j\right] = f(\alpha,\beta,\gamma)$
Inertial frame
MSe
$\vec{P}_{pes\to j}{}^0$
1
I
$I:\left[m_j\right]_0$
Body S_k
Body S_j
Body S_l
$\vec{z}_j$
M_k
$\vec{y}_j$
G_j
$\vec{x}_j$
M_l
$\overrightarrow{OG_j} = \begin{pmatrix} x_j \\ y_j \\ z_j \end{pmatrix}_{R_0}$
β_j
α_j
$\vec{z}_{\alpha_j}$
$\vec{z}_0$
$\vec{z}_\beta = \vec{z}_j$
$\vec{y}_j$
γ_j
$\vec{y}_{\alpha_j} = \vec{y}_{\beta_j}$
O
$\vec{x}_0 = \vec{x}_{\alpha_j}$
α_j
$\vec{y}_0$
$\vec{x}_j$
β_j
$\vec{x}_{\beta_j}$
γ_j
Absolute coordinates with cardan angles

Fig. 11.1 Bond graph model of the rigid body



$$\vec{\Omega}(j/0)^j = \begin{bmatrix}\Omega_{jx}\\ \Omega_{jy}\\ \Omega_{jz}\end{bmatrix}$$

$$\overbrace{\begin{bmatrix}\dot{\alpha}_j\\ \dot{\beta}_j\\ \dot{\gamma}_j\end{bmatrix}}^{\vec{\dot{\varepsilon}}_j} = \begin{bmatrix}\cos(\gamma_j)/\tan(\beta_j) & -\sin(\gamma_j)/\tan(\beta_j) & 0\\ \sin(\gamma_j) & \cos(\gamma_j) & 0\\ -\cos(\gamma_j)\tan(\beta_j) & \sin(\gamma_j)\tan(\beta_j) & 1\end{bmatrix}\overbrace{\begin{bmatrix}\Omega_{jx}\\ \Omega_{jy}\\ \Omega_{jz}\end{bmatrix}}^{\vec{\Omega}(j/0)^j}$$

$$\vec{\dot{\varepsilon}}_j = \begin{bmatrix}\dot{\alpha}_j\\ \dot{\beta}_j\\ \dot{\gamma}_j\end{bmatrix} \qquad \vec{\varepsilon}_{j\,ini} = \begin{bmatrix}\alpha_{j\,ini}\\ \beta_{j\,ini}\\ \gamma_{j\,ini}\end{bmatrix} \qquad \vec{\varepsilon}_j = \begin{bmatrix}\alpha_j\\ \beta_j\\ \gamma_j\end{bmatrix} \qquad \vec{\varepsilon}_j = \int \vec{\dot{\varepsilon}}_j du + \vec{\varepsilon}_{j\,ini}$$

$$P_0^j = \begin{bmatrix}\cos(\beta_j)\cos(\gamma_j) & -\cos(\beta_j)\sin(\gamma_j) & \sin(\beta_j)\\ \cos(\gamma_j)\sin(\alpha_j)\sin(\beta_j)+\cos(\alpha_j)\sin(\gamma_j) & \cos(\alpha_j)\cos(\gamma_j)-\sin(\alpha_j)\sin(\beta_j)\sin(\gamma_j) & -\cos(\beta_j)\sin(\alpha_j)\\ \sin(\alpha_j)\sin(\gamma_j)-\cos(\alpha_j)\cos(\gamma_j)\sin(\beta_j) & \cos(\gamma_j)\sin(\alpha_j)+\cos(\alpha_j)\sin(\beta_j)\sin(\gamma_j) & \cos(\alpha_j)\cos(\beta_j)\end{bmatrix}$$

Fig. 11.2 Calculation of the Cardan angles and rotation matrix from the angular velocity

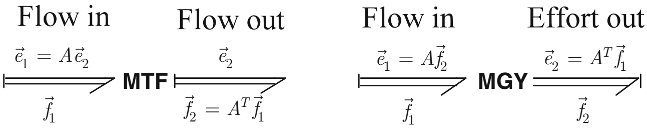


Fig. 11.3 Causalities imposed with TF and GY elements

The second causality constraint (C2) is: *the causality of transformers implied in cross products and the causality of gyrators in the bond graph model of the rigid body are intrinsically fixed, the transformers implied in cross products must be flow-in-flow-out causality, and the gyrators must be flow-in causality*. Indeed, contrary to scalar bond graphs, the ideal two-ports elements cannot propagate causality in both directions when the moduli are not invertible. In the frame studied (multibody system modeling with bond graphs and the Tiernego and Bos approach), these cases in which moduli (matrices associated to the elements) are not invertible are present for two elements: the transformer (TF) between the rotational and translational domain and the gyristor (GR) since both elements implement cross products. Consequently, transformers and gyrators have the mandatory fixed causality assignment specified above.

The transformers and gyrators with the acceptable causal forms mentioned are given in Fig. 11.3.

These imposed causalities on the transformers lead to some specific precautions when multibody systems with kinematic loops are considered. Figure 11.4 presents the Bond graph model of a rigid body with vector bonds and the imposed causalities mentioned above. Consequently, in the BG of a rigid body, attaching flow sources (Sf) to more than one hinge point or the center mass at the same time is not possible and leads to causality conflicts. This situation typically occurs when the multibody system is composed with kinematic loops. In [12], the author gives the example of an oscillating bar which, with its two joints, is a closed kinematic chain system with one body.

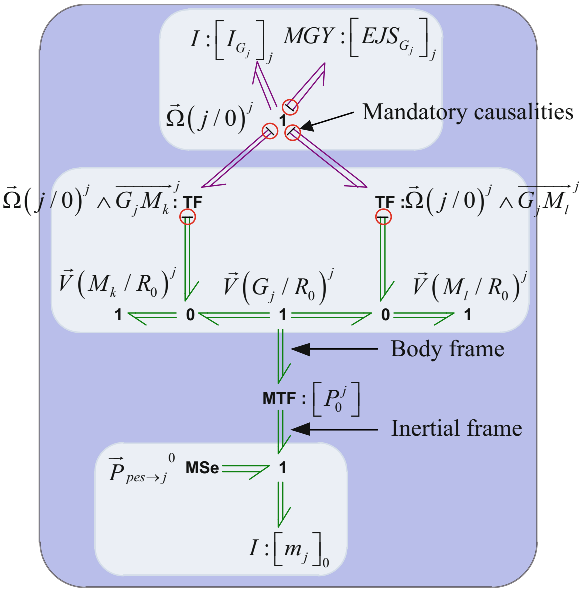

$I:\left[I_{G_j}\right]_j$
$MGY:\left[EJS_{G_j}\right]_j$
$\vec{\Omega}(j/0)^j$
1
Mandatory causalities
$\vec{\Omega}(j/0)^j \wedge \overrightarrow{G_jM_k}^j$: TF
TF : $\vec{\Omega}(j/0)^j \wedge \overrightarrow{G_jM_l}^j$
$\vec{V}(M_k/R_0)^j$
$\vec{V}(G_j/R_0)^j$
$\vec{V}(M_l/R_0)^j$
1
0
1
0
1
Body frame
MTF : $\left[P_0^j\right]$
Inertial frame
$\vec{P}_{pes\rightarrow j}^{\;0}$
MSe
1
$I:\left[m_j\right]_0$

Fig. 11.4 Bond graph model of rigid body with vector bonds

11.2.4.2 About DAE Formulation

In this chapter, in order to keep a modular approach as mentioned in the previous sections, the Tiernego and Bos method has been chosen and then absolute coordinates are selected. It is important to notice that both open chain (OC) and closed kinematic closed (CKC) systems lead to a DAE formulation.

Indeed, due to the kinematic constraints, derivative causality appears on the inertial elements and leads to differential-algebraic equations. In other words, the variables of these inertial elements with derivative causality depend on the variables of the inertial elements with integral causality through algebraic constraints. Consequently, one of the necessary priorities of the simulation method exposed in this paper will be either to handle DAEs using Lagrange multipliers method and specific solvers such BDF or to transform these DAEs it into ODE forms using the parasitic elements method.

11.2.5 Existing Simulation Methods

The methods presented in this section come from the references given in Sect. 11.3.1. The added value of this study is to give practical guidelines to modelers so that they can implement the adapted method in 20-sim software.

11.2.5.1 ZCP Method

General Aspects

Bond graphs are composed of different types of causal paths. The closed causal paths without integration operations are called zero-order causal paths (ZCP) [11, 13, 25, 30]. The mathematical models with ZCPs lead to DAEs. There is a direct link between the nature of the ZCPs in the bond graphs and the index of the DAE's index. The definitions of ZCPs used are those presented in [13] and are the following:

Class 1 ZCPs: The causal path is set between storage ports with integral causality and storage ports with differential causality. The associated topological loops are flat.

Class 2 ZCPs: The causal path is set between elements whose constitutive relations are algebraic (resistors are the most typical case). The topological loops are flat.

Class 3 ZCPs: It is a causal cycle whose topological loops are open. The causal path starts and ends in the same port of an element.

Class 4 ZCPs: The Class-4 ZCPs are defined as causal cycles whose topological loops are closed.

Only Class 4 ZCPs lead to DAEs with an index of 2. When the DAE's index is inferior to 2, a Backward Differentiation Formula (BDF) solver (such as the one in 20-sim) can handle these equations. In some cases (scalar bonds or vector bonds with no kinematic loops), the simulation can thus be conducted without the addition of specific elements.

Pros

The advantages of the ZCP method are as follows. First, this method enables simulations to be conducted directly from the model without requiring the addition of elements (R/C elements or controlled effort sources) at the correction location. The initial physical model is therefore not changed contrary to the R/C elements. Secondly, as we will see in Sect. 11.4, this method is faster than the two following methods (R/C elements and controlled effort sources).

Cons

When multibody systems with kinematic loops are modeled, the bond graph may contain Class-4 ZCPs which lead to a DAE formulation with an index superior to 1 and, then, the BDF solver often encounters difficulties in the numerical computation of the model. The ZCP opening method consists of opening the Class-4 ZCP, thanks to the use of break variables. This technique can be done at an equation level or at a graphical level. The classic technique is to add modulated sources (MSe) at 1-junctions of the Class-4 ZCPs so as to open them. Without a systematic approach to the detection of the Class-4 ZCPs, this method may be difficult to use when dealing with a case that involves complex multibody systems. This method was thus not used in this chapter in the case of multibody systems with kinematic loops.

11.2.5.2 R/C or Parasitic Elements

General Aspects

The R/C elements (or parasitic elements) method is based on the introduction of: compliances and resistances in the BG models of the joints. This method has been used for a long time. It appeared for the first time with the denomination “Stiff-compliance” approach in [17]. In the literature, other’s terminologies can be found for this method: virtual springs [32] or also singular perturbation [35].

In the framework of this chapter (focused on the simulation of multibody systems), this method can have two objectives: eliminating the kinematic constraints and also consequently eliminating the derivative causality or algebraic loops so that an explicit solver may be used. As we have previously mentioned (in Sect. 11.3.1.1), the nature of vector bond graphs imposes some supplementary causality constraints (C1 and C2). For the enforcement of these constraints, the flow sources (Sf) can lead to causal conflicts because of their flow-out causality. The enforcement of constraints with parasitic elements (R and C elements) can circumvent these constraints (C1 and C2) instead of flow sources (Sf), thanks to their effort-out causalities. Firstly, it allows the preservation of the same causality assignments for all of the bonds of a multibond (C1 respected). Secondly, it allows the suppression of the causality conflict which may appear because of the imposed causality of the transformers implied in the cross product and gyrators (C2 respected).

Moreover, the use of R/C elements into joint models decouples the energy store elements associated with the inertia of the rigid bodies. All these energy store elements are then kept with integral causalities.

The values of the compliant elements must be chosen carefully. To our knowledge, two methods for selecting these elements exist: (1) the eigenvalues decoupling between the parasitic frequency and the system frequency, (2) the use of activity metric [24]. These parameters can be chosen so as to model the joint compliances which exist in all mechanical joints. Thus, this point gives to this method a physical significance. The stiffnesses introduced should be high enough so as not to change the dynamic of the system but not too high so as to prevent the numerical difficulties of stiff problems (with high-frequency dynamics). This method leads to a necessary compromise between the accuracy of the results and the simulation time: the stiffer the system is, the more numerical errors are reduced but the more simulation time remains important. However, the increase of the simulation time can be balanced by parallel processing as the mass matrix in a block-diagonal form can be used to decouple the system as it is explained in [32]. As T. Rayman recommends, adding a damping element (R resistive element) in parallel with the stiff spring (C energy store element) enables the dampening of the high eigen frequency associated with the high stiffness. The exact influence of these parameters still remains a topic of research in which the authors are particularly interested in.

Pros

With the RC elements method, all the derivative causalities are removed. This leads to an explicit ODE model which can be solved with a fixed step solver such as Runge Kutta 4.

Contrary to the Lagrange multipliers methods, some slight imprecisions could be made on the initial conditions. Moreover, overconstrained (also called hyperstatic) systems can be simulated without any difficulties.

Cons

This method introduces new elements (R and C elements) in the initial bond graph whose parameters and initial conditions associated with the C store elements need to be specified. The values of the compliant elements must be chosen carefully so as to model the joint compliances which exist in all mechanical joints. The stiffnesses that are used should be high enough so as not to change the dynamics of the system and should be able to sufficiently approximate the constraints. However, too high stiffnesses introduce numerical difficulties of stiff problems (with high-frequency dynamics). Indeed, these kinds of problems force the solvers to take very small integration steps to meet the tolerance criteria and, consequently, the simulation is slowed. This method leads to a necessary compromise between the accuracy of the results and the simulation time: the stiffer the system is, the more numerical errors are reduced but the more the simulation time remains important.

Implementation

Even if explicit solvers can be used, the authors recommended the use of the implicit Modified Backward Differentiation Formula (MBDF) solver for two reasons. The first reason is that the BDF solver guarantees the numerical stability of the solution. The second reason is that, with the controlled effort sources, only the MBDF solver can be used. Also, using an MBDF solver with the R/C Elements is a good way to facilitate comparison between R/C elements and controlled effort sources methods.

11.2.5.3 The Lagrange Multipliers Method

General Aspects

In the “Lagrange-multiplier method,” the constraint forces are not modeled by parasitic elements but by controlled effort sources (MSe). The origin of this method comes from [7, 14] and has been implemented in 20-sim in [5, 6, 31]. The same concept was also implemented by [12] with pseudoflow source (PSf) elements.

The general idea of this method is similar to the parasitic elements that is to say: removing the flow sources (Sf) and enforcing the constraint by other elements (here the controlled effort sources MSe)) which don’t have flow causality and thus suppress the causality conflict.

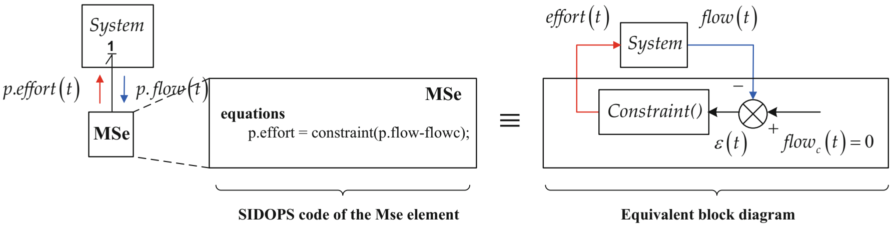
The controlled effort sources (MSe) are computed such that the difference between velocities for the constraints is zero. The principle of the method is to apply an effort equivalent to the one that the system would impose on a flow source. It would have a practical operation as a flow source but with effort-out causality instead of a flow-out causality.

The constitutive laws of the controlled effort sources are the following:

- the constitutive law of an effort source,
- the following constraints: effort = e such as flow(e) = f.

Figure 11.5 may help to understand the implementation of this constraint. This implementation uses a specific function provided by the 20-sim software: *constraint()* function. At every simulation step, this function induces an iterative procedure to find exactly the force that keeps the velocity offset at zero. This iterative procedure is only supported by the Modified Backward Differentiation Formulation method of the 20-sim software.

Due to the nature of the causality of the controlled effort sources (effort-out causality), all the inertial elements (I elements) receive integral causality. The dependent states are thus not visible in the form of derivative causality. However, a controlled effort source (MSe elements) establishes itself algebraic dependencies with the state variables of the inertia and thus indicates an implicit form of equations. As it is mentioned in [5], the mathematical model obtained (a semi-explicit state space model) can be solved by a BDF solver although the DAE system is index 2 due to the fact that the constraint forces do not appear in the algebraic constraints.



```
equations
    p.effort = constraint(p.flow-flowc);
```

Fig. 11.5 Implementation of the controlled effort source

Pros

Contrary to the parasitic elements method, the modulated sources (MSe) don't need additional tuning because no additional parameters are added to the system. In other words, the order of the system is not modified because no new states (with associated parameters and initial conditions) are introduced. Moreover, the absence of supplementary parameters makes a bond graph that describes the system ideally in the limit of the numerical tolerance on the constraints equations.

Even if, during the simulation, iterations are required at each time step to satisfy the constraint equations, the computational load is comparable or better in comparison to the parasitic elements method where the differential equations are truly explicit but very stiff.

Cons

The bond graph obtained with this approach leads to implicit differential equations, so explicit integration algorithms cannot be used. Moreover, during the simulation the constraints can only be met within some numerical tolerances.

Consequently, the implementation of this method requires special care, which is described hereafter.

Implementation

The Lagrange multipliers method is more difficult to implement than the use of RC elements because the kinematic constraints imposed by the joints may produce computational conflicts. This issue occurs on closed kinematic chains with overconstrained multibody systems. Due to the topology of the system, more than one joint could impose the same constraint on the system, in which case the simulation will no longer be possible. In other ways, more practically, the number of the controlled effort sources must not exceed the degrees of freedom that need to be eliminated. In order to solve this problem, the redundant constraints must be removed.

In addition, there may be the inconsistent initial conditions. Indeed, initial conditions of the system should be consistent with constraints. If a controlled effort source is attached to a 1-junction to keep the velocity at zero and a non-zero initial velocity to the I element connected to the same 1-junction is imposed, then a simulation cannot be conducted. We can notice at this point that 20-sim does not offer an automatic correction of inconsistent initial conditions but, a BG modeler can always code the calculation of consistent initial conditions to handle it automatically.

11.2.6 Kinematic Joints Modeling

The joint models permit the expression of the constraints that are introduced when rigid bodies are connected. As the bond graph model of the rigid body, the joint models have been built in a modular way in the sense that their modeling does not change when the whole model of the system is assembled. The idea of this section is to allow the BG practitioners to be able to use a library of all the common existing kinematic joint models.

11.2.6.1 Common Aspects

The modeling of kinematic joints determines the rotational or translation degree of freedom allowed by the joint.

Flow sources can be used to suppress the joint's degrees of freedom. However, in order to circumvent the causality constraints mentioned before, the joints models are presented with an L element. The element L stands either for an R/C element or a modulated effort source MSe depending on the choice of the simulation methods (R/C elements methods or the use of Lagrange multipliers).

For the mobilities let free, the choice of modeling assumptions can lead to add elements. If the joints are assumed to be perfect that is to say without any dissipating or energy storage phenomenon, the junction 1 can be let free of elements. If, however, dissipating or spring phenomenon are assumed in the joints, R or C elements can be added at these 1 junctions. One can notice that these R/C elements shouldn't be confused with the R/C elements modeling parasitic elements and will be called functional elements in Sect. 11.4.

The joint model is built from the following kinematic relationships:

$$\overrightarrow{\Omega}(1/0)^1 = \overrightarrow{\Omega}(1/2)^1 + \overrightarrow{\Omega}(2/0)^1 \tag{11.5}$$

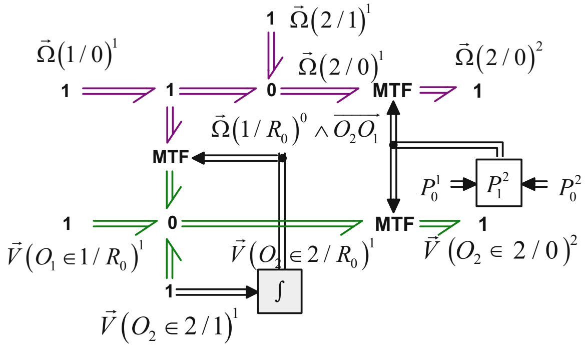
where

$\overrightarrow{\Omega}(1/0)^1$ is the absolute velocity of body 1 expressed in R_1 frame
$\overrightarrow{\Omega}(1/2)^1$ is the relative angular velocity of body 1 with regard to body 2 expressed in R_1 frame
$\overrightarrow{\Omega}(2/0)^1$ is the absolute velocity of body 2 expressed in R_1 frame
and

$$\begin{aligned}\overrightarrow{V}(O_2 \in 2/0)^1 &= \overrightarrow{V}(O_2 \in 2/1)^1 + \overrightarrow{V}(O_2 \in 1/0)^1\\ &= \overrightarrow{V}(O_2 \in 2/1)^1 + \left(\overrightarrow{V}(O_1 \in 1/0)^1 + \overrightarrow{\Omega}(1/0)^1 \wedge \overrightarrow{O_1O_2}^1\right)\end{aligned} \tag{11.6}$$

which can be also written as

Fig. 11.6 General kinematic joint model



$$\overrightarrow{V}(O_2/0)^1 = \overrightarrow{V}(O_2 \in 2/1)^1 + \left(\overrightarrow{V}(O_1/0)^1 + \overrightarrow{\Omega}(1/0)^1 \wedge \overrightarrow{O_1O_2}^1\right) \tag{11.7}$$

where

$\overrightarrow{V}(O_2/0)^1$ is the absolute velocity of point O_2
$\overrightarrow{V}(O_2/0)^1$ is the absolute velocity of point O_1
$\overrightarrow{V}(O_2 \in 2/1)^1$ is the relative velocity of point O_2 with regard to the frame R_1
$\overrightarrow{\Omega}(1/0)^1 \wedge \overrightarrow{O_1O_2}^1$ is the velocity component due to the rotation of frame R_1 with regard to the inertial reference frame

The general kinematic joint model is then detailed in Fig. 11.6.

In the general kinematic joint model, MTF elements modulated by coordinate transformation matrix are used so as to express the kinematic quantities in the appropriate frame associated with the body where they are connected to.

11.2.6.2 Joint Models

Based on the general kinematic joint model, the models of the common kinematics joints are given in Figs. 11.7, 11.8, and 11.9.

11.3 Presentation of the Vibration Absorber System

11.3.1 Context

The rotor of a helicopter can generate various vibration phenomena. Let us consider:

- forced vibrations,

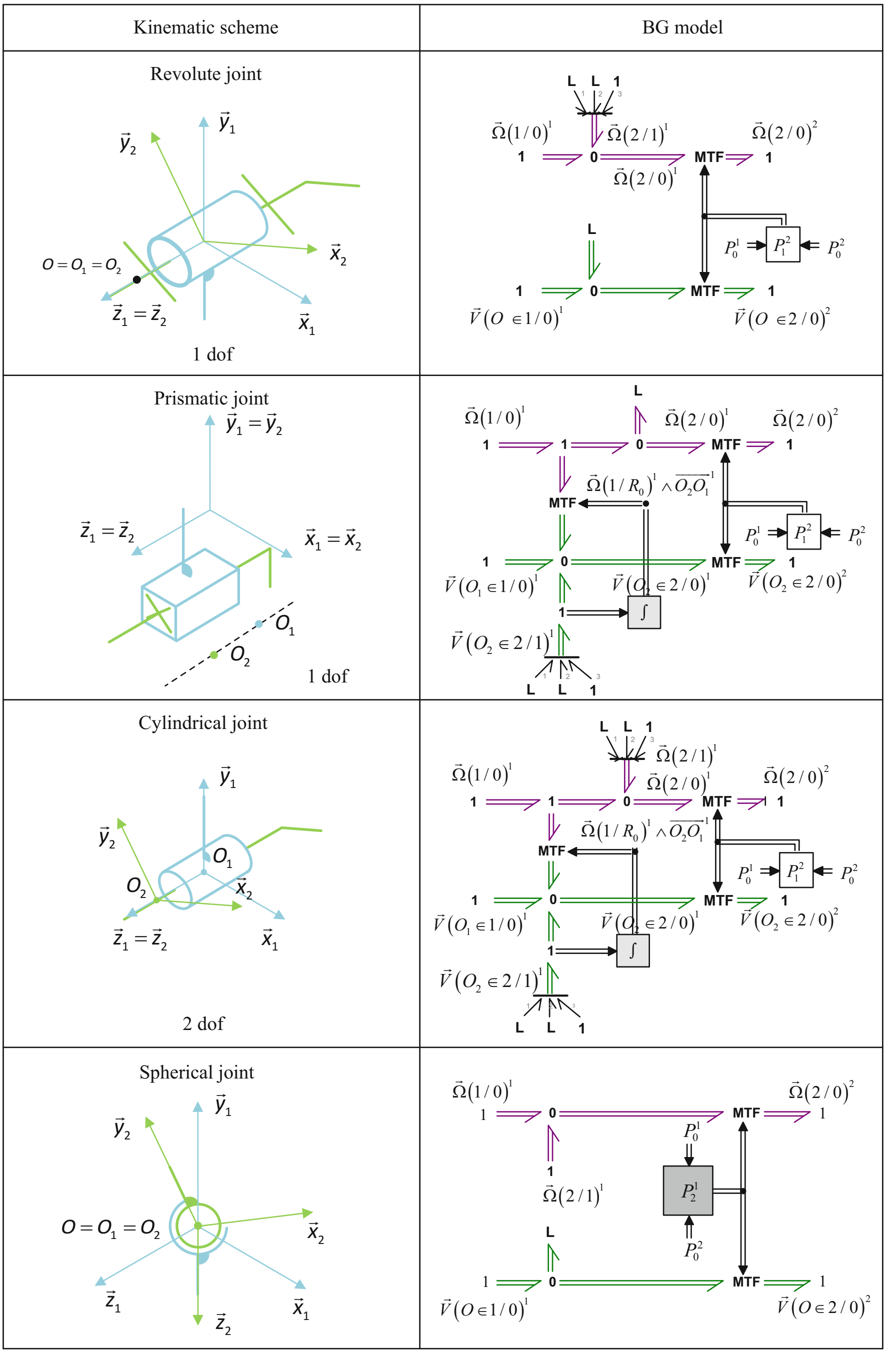


Fig. 11.7 Kinematic joint models—part 1

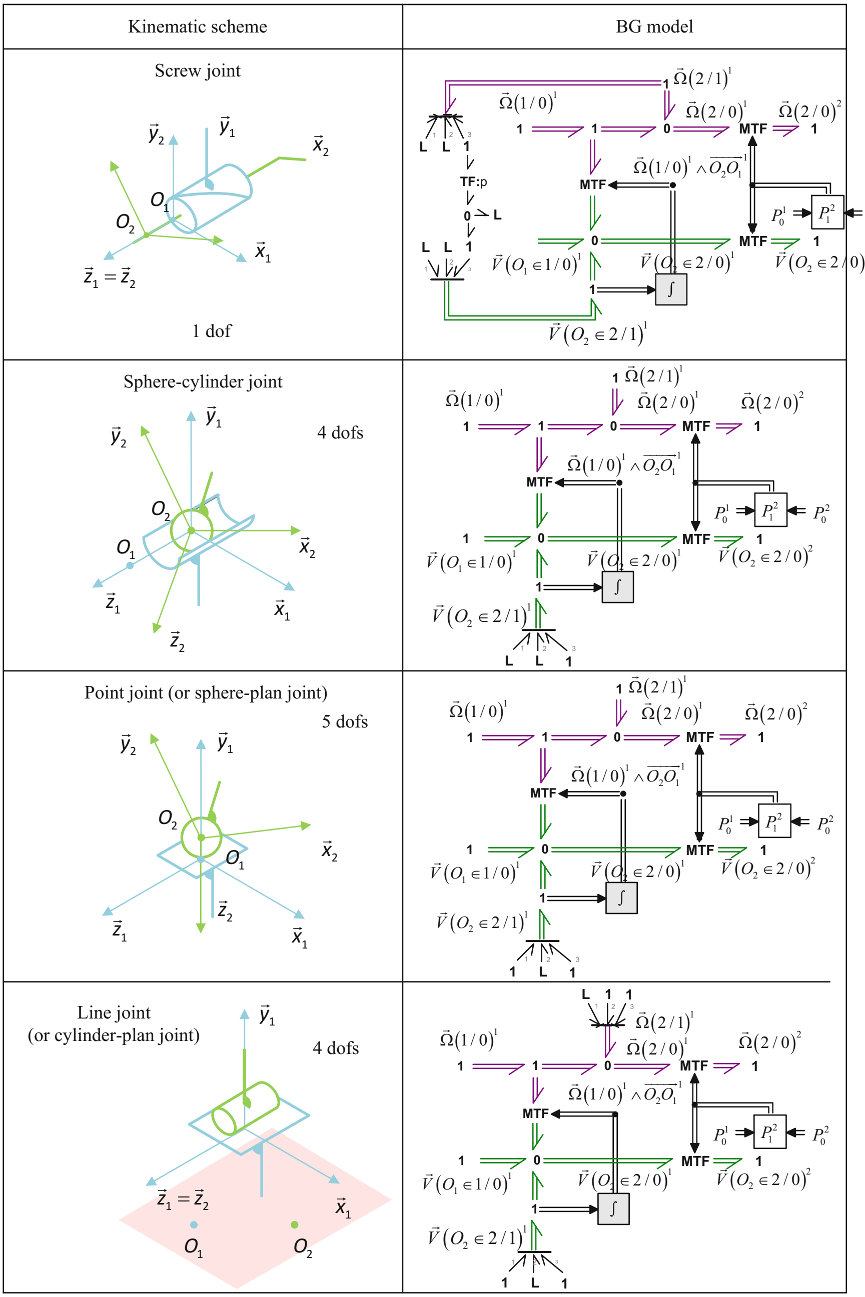


Fig. 11.8 Kinematic joint models—part 2

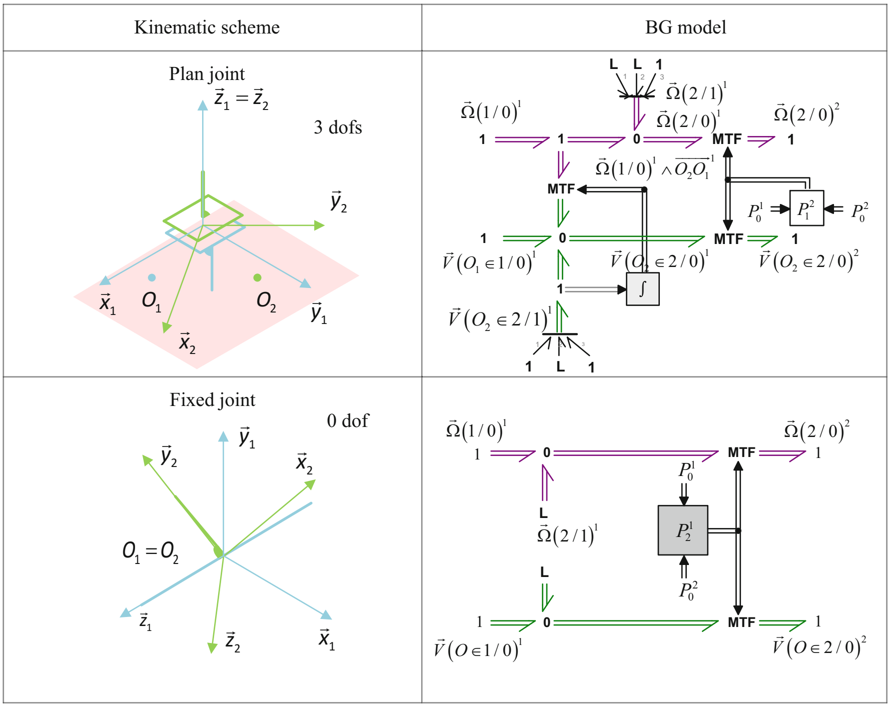


Fig. 11.9 Kinematic joint models—part 3

- resonances “ground and air,”
- dynamic problems of the power chain.

Blades undergo periodic and alternating inertia and aerodynamic forces whose fundamental frequency is the rotation frequency of the rotor as explained in [18]. These efforts on the blades cause forces and torques on the hub which then becomes a mechanical excitation of the fuselage. It is always shown in [18] that these forces and moments transmitted from the rotor to the fuselage have frequencies whose are frequency harmonics (kbΩ) with b the number of blades of the main rotor and Ω the rotation speed of the main rotor. Moreover, most of the time, the first harmonic bΩ is preponderant.

Therefore, its behavior depends on its dynamic characteristics and the filtering systems placed between the rotor and the fuselage (as shown in Fig. 11.10). In this sequel, we will focus on a vibration absorber system: the SARIB system (called Suspension Antivibratoire à Résonateur Intégré dans les Barres in French).

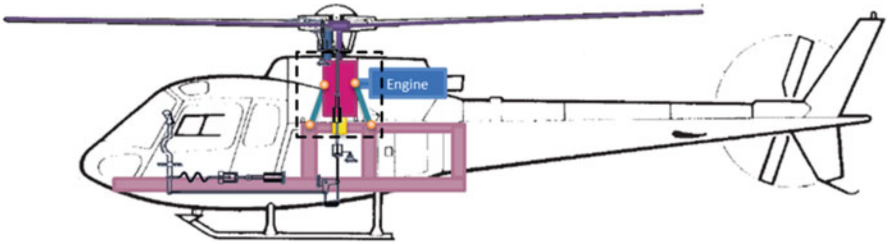


Fig. 11.10 Helicopter suspension between the MGB and the aircraft structure

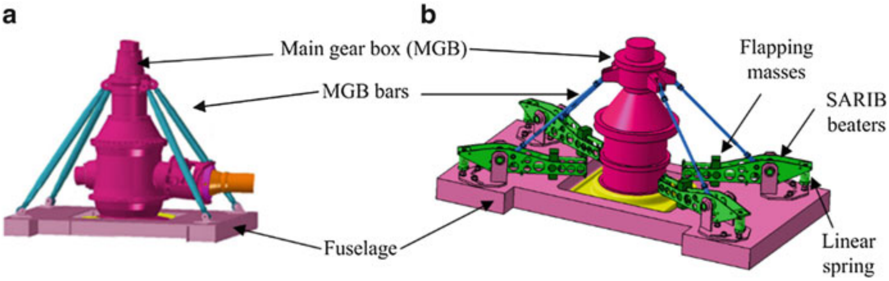


Fig. 11.11 (**a**) Flexible classical MGB-fuselage suspension, (**b**) MGB-fuselage suspension with SARIB system

11.3.2 Interest and Functions

The MGB-fuselage suspension must ensure several important functions. Firstly, the joint allows the transmission of the static force necessary to the sustentation of the helicopter with a limited static displacement. Moreover, the suspension helps to reduce the mechanical vibrations transmitted to the fuselage according to the force and displacement aspects. In this paper, we will focus on this latter function.

11.3.3 Operating Principle

The classical MGB-fuselage suspension is composed of four MGB bars and a main membrane as shown in Fig. 11.11a. The MGB bars can rigidly suspend the fuselage without flexibility to the rotor and thus transmit the lift from the rotor to the structure.

The membrane is a flexible suspension with the following observed behavior:

- a low stiffness for angular movements around the roll and pitch axes and the linear vertical pumping displacement,
- a very high stiffness for linear movements perpendicular to the vertical direction and for the yaw movement.

Thus, the membrane allows the angular movement of the MGB around the pitch and roll axes. The flexibility of the membrane around these axes allows a strong filtration of the dynamic moments around these axes. In addition, the membrane transmits the main rotor torque, thanks to its very high stiffness around the yaw axis. In conclusion, the conventional suspensions allow filtering pitch and roll dynamic moments without filtering the pumping dynamic efforts.

The suspension equipped with SARIB system is a tuned-mass vibration absorber. Its purpose is to render the filtering of these pumping dynamic efforts possible. On the SARIB solution, beaters with flapping masses are added between the MGB bars and the fuselage (Fig. 11.11b). They permit a vertical degree of freedom to the MGB with regard to the fuselage. The principle of the SARIB system is to tune, for an accurate defined frequency, these flapping masses on each SARIB bar so as to create inertial forces on the fuselage opposite to the force of the MGB bars on the beaters. Moreover, the specificity of the SARIB system among the class of tuned-mass absorbers comes from the fact that a rotating SARIB bar works as a mechanical gear ratio that makes the resonating flapping mass appear larger. More details on this operating principle are given in [15] through the equilibrium of an SARIB bar. In other words, there is an anti-resonance phenomenon on the transmissibility function between the excitation applied to the MGB. Hence, the forces transmitted to the fuselage can be strongly reduced at this anti-resonance frequency.

11.3.4 Use of the Experimental Setup

To analyze the simulation results, an experimental setup on a small scale has been used. The geometry and mass properties correspond to a light helicopter with a scale of approximately ½. The model proposed in this chapter has been simulated with the numerical data of this experimental setup detailed in [8].

The experimental setup has to represent the flight behavior of the helicopter in different operating phases: hover, up/down, or turn. In these operating phases, the MGB is subjected to pumping, pitch, and roll excitements. On the experimental setup, the rotor action, provided by the aerodynamic forces and the blades stiffness, applied to the MGB will be approximated by a sinusoidal excitation at the frequency bΩ obtained with a vibration equipment. The experimental setup was performed by maintaining the architecture of a real device (Fig. 11.12).

The correspondence between the different mechanical elements of a helicopter and the experimental setup will be detailed hereafter.

To reproduce the effects of the rotor lift, the set {BTP-link-fuselage} has been suspended from the fixed frame (Fig. 11.12). This joint has to lift up the entire weight suspended by analogy to the static force of lift, it was performed by the use of pneumatic components. They were designed so that the natural frequency induced by the overall stiffness is below 2 Hz.

The fuselage was replaced by a rigid mass called fuselage mass (Fig. 11.12).

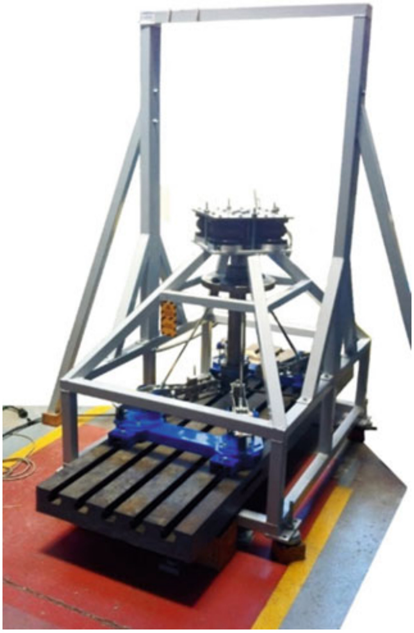


Fig. 11.12 Experimental setup of the semi-active suspension

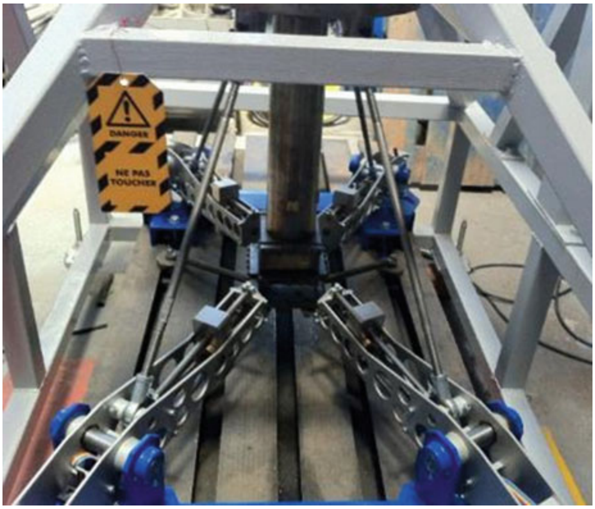


Fig. 11.13 Position of the fixtures to the fuselage

A single body has replaced the set {rotor shaft-MGB} and the rotation has been deleted. This single body is composed of a shaft equipped with a recessed tray (Fig. 11.13).

The overall joint between the fuselage mass and the system equivalent to the set {rotor shaft-MGB} has four bars, four SARIB beaters, and a system representing the usually existing membrane on the helicopter.

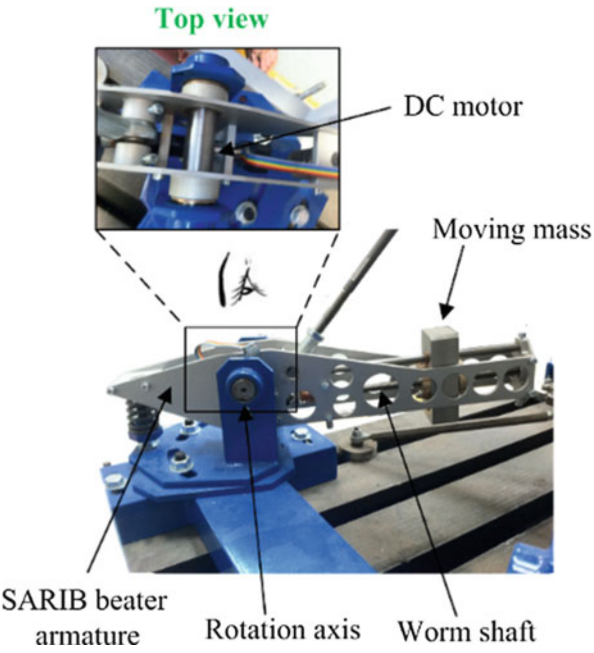


Fig. 11.14 Revolute joint between an SARIB beater and the fuselage

The MGB bars consist of threaded rods that have a spherical joint without backlash at each end.

Each SARIB beater is linked by a revolute joint to the fuselage, which are themselves clamped to the fuselage mass as shown in Fig. 11.13. Each SARIB beater is fitted with a moving mass (Fig. 11.14) connected with a prismatic joint along the SARIB beater. It should be noted that, for the passive suspension presented in this chapter, the moving masses are fixed on the SARIB beaters.

11.4 Modeling of the Vibration Absorber Suspension

The bond graph modeling steps described above will now be applied to model the MGB-fuselage suspension equipped with the SARIB device. First, the mechanical model of the MGB-fuselage suspension and the associated assumptions will be presented. Secondly, the construction of the bond graph model of the MGB-fuselage suspension will be detailed.

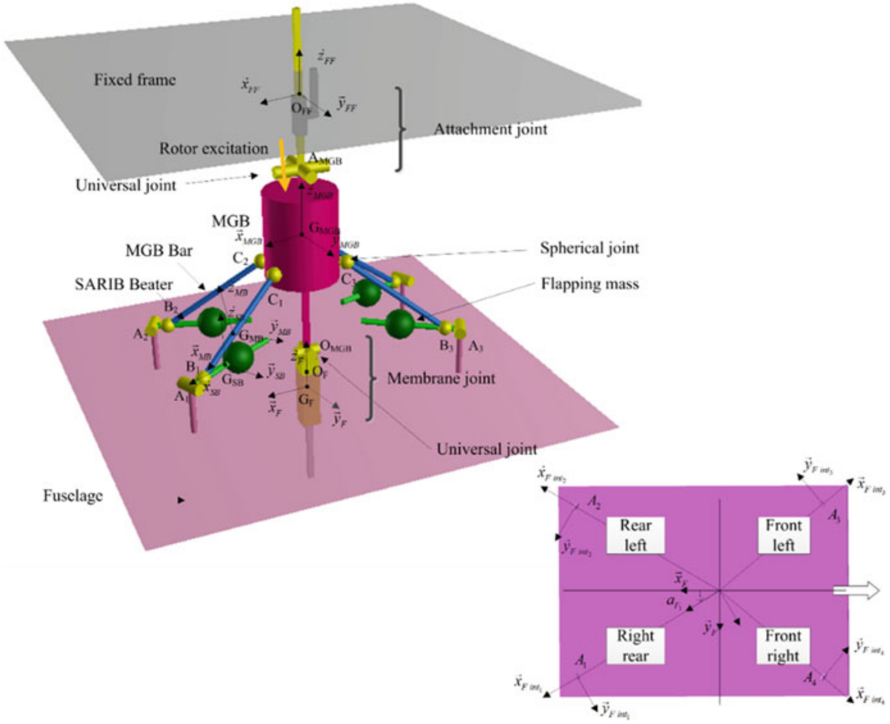


Fig. 11.15 Kinematic scheme of the 3D joint between the main gearbox and the fuselage

11.4.1 Modeling Assumptions of the Vibration Absorber Suspension

The 3D vibration absorber suspension is a set of 14 rigid bodies (the MGB, four MGB bars, four SARIB beaters, four intermediate bodies, and the fuselage) and 18 kinematic joints (four revolute joints (R), eight spherical joints (S), and two prismatic joints (P)). The kinematic scheme of the 3D MGB-fuselage suspension is shown in Fig. 11.15.

For the sake of clarity, the different kinematic joints are also described in the joints graph (Fig. 11.16).

The MGB is suspended by a joint called attachment to the fixed frame. The role of this joint is to support the weight of the system and consequently let the fuselage with six degrees of freedom in the air. This joint consists of a universal joint and a prismatic joint. The weight of the system (mainly the weight of the MGB and the fuselage) is taken up by a tension spring at the prismatic joint of this attach. The spring is dimensioned such that the natural frequency of the mass/spring system is small enough and distant from the excitation frequency applied to the MGB.

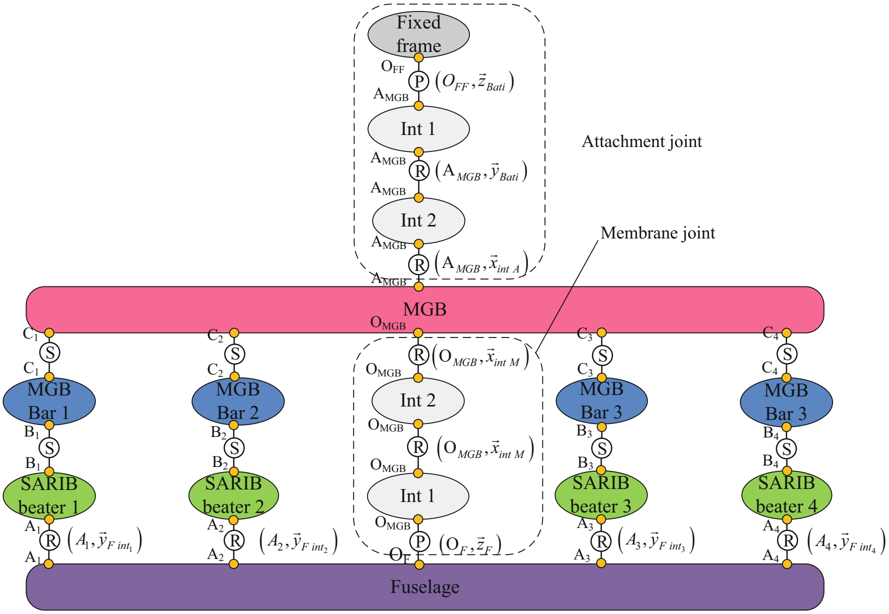


Fig. 11.16 Joints graph of the 3D vibration absorber suspension

The structure of the suspension consists of four identical legs and a central membrane. Each leg consists of SARIB beaters and an MGB bar connected by a spherical joint. The upper ends of these legs are connected to the MGB with spherical joints and the lower ends of these legs are connected to the fuselage through revolute joints. The revolute joints between SARIB beaters and the fuselage have a torsion spring dimensioned such that the anti-resonance frequency in the efforts transmitted to the fuselage is positioned on the rotor excitation frequency while taking up enough effort to lift the helicopter with a small displacement between the MGB and fuselage.

Given the physical phenomena observed, the main membrane between the MGB and the fuselage is modeled by two revolute joints with orthogonal axes (also called universal joint) and a prismatic joint in serial. The flexibilities of the membrane in pumping, roll, and pitch are modeled by the degrees of freedom of these joints combined with stiffness. A torsion spring with low stiffness at each revolute joint of the universal joint and a linear spring also with low stiffness along the z-axis will model the filtering carried out by the membrane. The rigidities of the membrane are modeled by rigid joints or, in other words, by a lack of degree of freedom around or along the movements considered.

These bodies are assumed to be rigid. The intermediate parts (called Int1 and Int2) are supposed with negligible masses. As previously mentioned, absolute coordinates have been chosen for modularity reasons. Some local moving reference frames are attached to these bodies:

$R_{\mathrm{FF}} = \left(O_{\mathrm{FF}}, \overrightarrow{x}_{\mathrm{FF}}, \overrightarrow{y}_{\mathrm{FF}}, \overrightarrow{z}_{\mathrm{FF}}\right)$ attached to the fixed frame,
$R_{\mathrm{MGB}} = \left(G_{\mathrm{MGB}}, \overrightarrow{x}_{\mathrm{MGB}}, \overrightarrow{y}_{\mathrm{MGB}}, \overrightarrow{z}_{\mathrm{MGB}}\right)$ attached to the MGB,
$R_{\mathrm{SB}_i} = \left(G_{\mathrm{SB}_i}, \overrightarrow{x}_{\mathrm{SB}_i}, \overrightarrow{y}_{\mathrm{SB}_i}, \overrightarrow{z}_{\mathrm{SB}_i}\right)$ attached to the SARIB Beaters,
$R_{\mathrm{MB}_i} = \left(G_{\mathrm{MB}_i}, \overrightarrow{x}_{\mathrm{MB}_i}, \overrightarrow{y}_{\mathrm{MB}_i}, \overrightarrow{z}_{\mathrm{MB}_i}\right)$ attached to the MGB Bar,
$R_F = \left(G_F, \overrightarrow{x}_F, \overrightarrow{y}_F, \overrightarrow{z}_F\right)$ attached to the fuselage and $R_{F\ \mathrm{int}_i} = \left(A_i, \overrightarrow{x}_{F\ \mathrm{int}_i}, \overrightarrow{y}_{F\ \mathrm{int}_i}, \overrightarrow{z}_{F\ \mathrm{int}_i}\right)$ also attached to fuselage so as to facilitate the definition of the axis of the revolute joints.

11.4.2 Modeling Assumptions of the Suspension Environment

As previously mentioned in Sect. 11.3, the action of the rotor on the MGB in the experimental setup is approximated by a sinusoidal excitation with a frequency bΩ.

The external mechanical actions applied to the MGB are the forces applied by the vibrating shakers. The description of the experimental setup shall be specified in Sect. 11.5.2 of this paper. These excitations will be applied along or around the joints of the “attach” joint between the fixed frame and the MGB. They allow the MGB to be subjected to pumping, pitch, and roll excitations. These mechanical actions will only consist of a dynamic component to analyze the vibration behavior of the suspension.

For the pumping excitation, the wrench applied by the actuator at the mobility of the prismatic joint is

$$\tau\,(\text{actuator along vertical axis} \rightarrow \text{prismatic joint}) = {}_{O_{\mathrm{FF}}}\left\{\begin{matrix} f(t)\overrightarrow{z}_{\mathrm{FF}} \\ \overrightarrow{0} \end{matrix}\right\} \quad \text{avec} \quad f(t) = F \times g(t) \tag{11.8}$$

For roll excitation, the wrench applied by the actuator at the mobility of the revolute joint around roll axis is

$$\tau\,(\text{Actuator around roll axis} \rightarrow \text{roll revolute joint}) = {}_{A_{\mathrm{MGB}}}\left\{\begin{matrix} \overrightarrow{0} \\ m_r(t)\overrightarrow{y}_{\mathrm{FF}} \end{matrix}\right\} \quad \text{avec} \quad m_r(t) = M \times g(t) \tag{11.9}$$

For pitch excitation, the wrench applied by the actuator at the mobility of the revolute joint around pitch axis is

$$\tau\,(\text{actuator around pitch axis} \rightarrow \text{pitch revolute joint}) = {}_{A_{\mathrm{MGB}}}\left\{\begin{matrix} \overrightarrow{0} \\ m_p(t)\overrightarrow{x}_{\mathrm{int}\ A} \end{matrix}\right\} \quad \text{avec} \quad m_p(t) = M \times g(t) \tag{11.10}$$

Different types of excitation are used by modulating the amplitude of each excitation by a dimensionless function g(t): constant, sinus type, or swept sine. The constant excitation permits to ensure that the equilibrium position of the system under its own weight is physically acceptable. The sine excitation permits to get more accurate frequencies of the anti-resonance frequencies. The swept sine excitation is used to analyze the frequency behavior of the system including resonances and anti-resonances frequencies.

For the swept sine excitation, a modulation function was constructed, as shown in four parts:

- within [0, t_1], a progressive increase to a sinusoidal excitation at a frequency of f_{start} de 5 Hz is carried out with a linear amplitude variation,

$$\begin{cases} \forall t \in \left[0, \frac{t_1}{2}\right], \quad g(t) = \left(4\frac{t^2}{{t_1}^2 2}\right)\sin\left(\omega_{\text{start}} t\right) \\ \forall t \in \left[\frac{t_1}{2}, t_1\right], \quad g(t) = \left(-\frac{2}{{t_1}^2}\left(t-\frac{t_1}{2}\right)^2 + \frac{2}{t_1}\left(t-\frac{t_1}{2}\right)+\frac{1}{2}\right)\sin\left(\omega_{\text{start}} t\right) \end{cases} \tag{11.11}$$

- within [t_1, t_2], the sine excited mode at f_{start} is preserved the necessary time so that the transient mode disappears and there is only permanent regime,

$$\forall t \in [t_1, t_2], \quad g(t) = \sin\left(\omega_{\text{start}} t\right) \tag{11.12}$$

- within [t_2, t_3], a linear swept sine is used with a sweep frequency from f_{start} (5 Hz) to f_{end} (25 Hz). The expression of the chirp signal is given in [19] and recalled below :

$$\forall t \in [t_2, t_3], g(t) = \sin\left(\left(\omega_{\text{start}} + \frac{\Delta\omega}{2\Delta T}(t-t_2)\right)\times t\right) \tag{11.13}$$

With $\Delta\omega = \omega_{\text{end}} - \omega_{\text{start}}$ and $\Delta T = t_3 - t_2$

- within [t_3,$+\infty$], the sine excited mode at $f_{\text{end}} = \frac{\omega_{\text{end}}}{2\pi}$ (25 Hz) is kept.

$$\forall t \in [t_3, +\infty], \quad g(t) = \sin\left(\omega_{\text{end}} t\right) \tag{11.14}$$

For the conducted simulations, the time intervals were defined with the numerical following values: $t_1 = 3$ s, $t_2 = 5$ s, and $t_3 = 25$ s. The chirp signal is built in a such manner that from t_2 to t_3 the excitation at the i s corresponds to an excitation at the i Hz (Fig. 11.17).

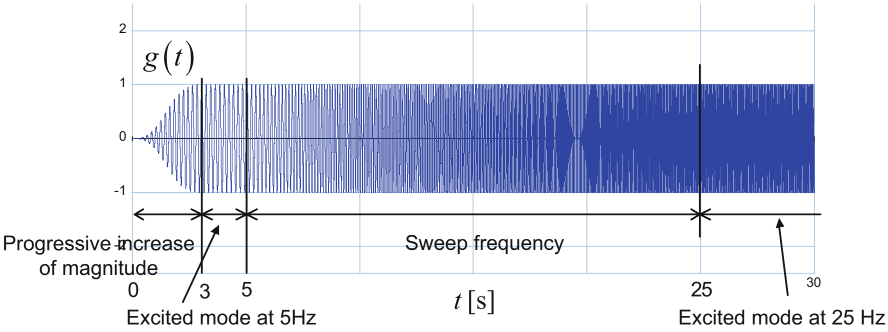


Fig. 11.17 Modulation function of the excitation $g(t)$

11.4.3 The Simulation Approach Chosen: The R/C or Parasitic Elements Methods

The bond graph simulation with the R/C or parasitic elements method is quite easy to implement compared to conventional techniques since there are no kinematic constraints to deal with. The R/C or parasitic elements method allows to tackle derivative causalities which can appear on some inertial elements and causalities constraints due to vector bonds. Moreover, this method, from our point of view, keeps a physical insight and permits the modeling with a modular approach.

11.4.4 Construction of the Bond Graph Model of the MGB-Fuselage Joint

11.4.4.1 Components Modeling

The components modeling (rigid bodies and kinematic joints) of the system will now be presented.

Rigid Bodies

The MGB

The MBG representation of the construction is given in Fig. 11.18. The bond graph representation shows as many branches in the structure junctions as kinematic joints for the determination of the velocities of corresponding points. From the linear velocity of the center of mass G_{MGB} and the instant rotation velocity $\overrightarrow{\Omega}$ (MGB/*Fixed frame*), the linear velocity of the point O_{MGB}, the linear velocity of

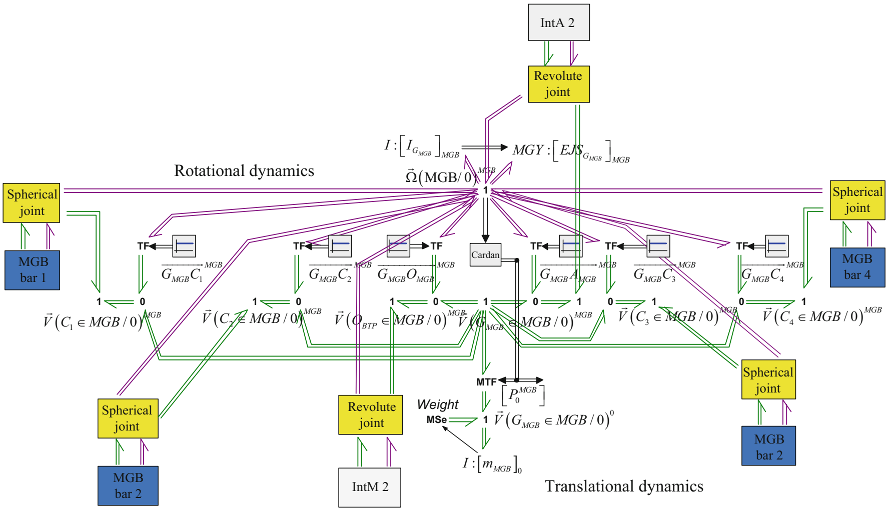


Fig. 11.18 Bond graph modeling of the MGB with its environment

A_{MGB} point of the MGB, and the linear velocities Ci point at the ball joints between BTP and BTP bars are calculated.

The Fuselage

With a similar approach, the bond graph modeling of the fuselage is given in Fig. 11.19.

The SARIB Beaters

The bond graph model of an SARIB beater i kinematically linked to the fuselage and the MGB bar i is given in Fig. 11.20.

The MGB Bars

The bond graph model of an MGB bar i kinematically linked to the MGB and an SARIB bar i is given in Fig. 11.21.

Kinematic Joints

The joint modeling will now be detailed. The singular perturbation method is here implemented by the addition of parasitic elements which were presented in the previous section.

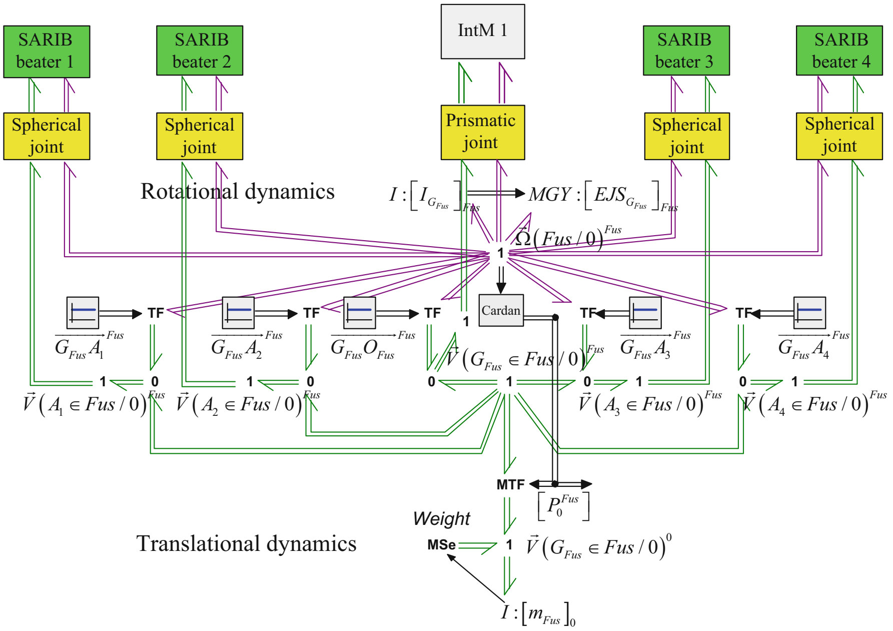

SARIB beater 1
SARIB beater 2
IntM 1
SARIB beater 3
SARIB beater 4
Spherical joint
Spherical joint
Prismatic joint
Spherical joint
Spherical joint
Rotational dynamics
$I:[I_{G_{Fus}}]_{Fus}$
$MGY:[EJS_{G_{Fus}}]_{Fus}$
$\vec{\Omega}(Fus/0)^{Fus}$
1
TF
TF
TF
Cardan
TF
TF
$\overrightarrow{G_{Fus}A_1}^{Fus}$
$\overrightarrow{G_{Fus}A_2}^{Fus}$
$\overrightarrow{G_{Fus}O_{Fus}}^{Fus}$
$\overrightarrow{G_{Fus}A_3}^{Fus}$
$\overrightarrow{G_{Fus}A_4}^{Fus}$
$\vec{V}(G_{Fus}\in Fus/0)^{Fus}$
1
0
1
0
0
1
0
1
0
1
$\vec{V}(A_1\in Fus/0)^{Fus}$
$\vec{V}(A_2\in Fus/0)^{Fus}$
$\vec{V}(A_3\in Fus/0)^{Fus}$
$\vec{V}(A_4\in Fus/0)^{Fus}$
MTF
$[P_0^{Fus}]$
Weight
MSe
1
$\vec{V}(G_{Fus}\in Fus/0)^0$
Translational dynamics
$I:[m_{Fus}]_0$

Fig. 11.19 Bond graph model of the fuselage with its environment

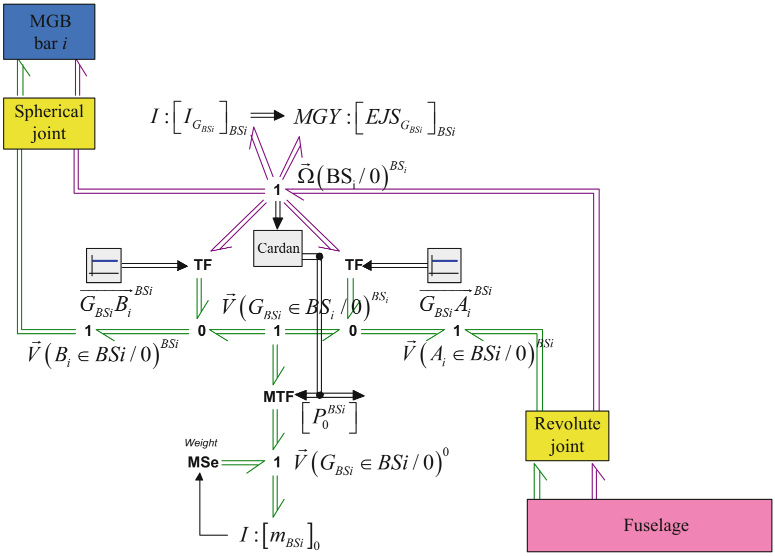

MGB bar i
Spherical joint
$I:[I_{G_{BSi}}]_{BSi}$
$MGY:[EJS_{G_{BSi}}]_{BSi}$
$\vec{\Omega}(\mathrm{BS_i}/0)^{BS_i}$
1
Cardan
TF
TF
$\overrightarrow{G_{BSi}B_i}^{BSi}$
$\overrightarrow{G_{BSi}A_i}^{BSi}$
$\vec{V}(G_{BSi}\in BS_i/0)^{BS_i}$
1
0
1
0
1
$\vec{V}(B_i\in BSi/0)^{BSi}$
$\vec{V}(A_i\in BSi/0)^{BSi}$
MTF
$[P_0^{BSi}]$
Weight
MSe
1
$\vec{V}(G_{BSi}\in BSi/0)^0$
$I:[m_{BSi}]_0$
Revolute joint
Fuselage

Fig. 11.20 Bond graph model of an SARIB beater *i* kinematically linked to its environment

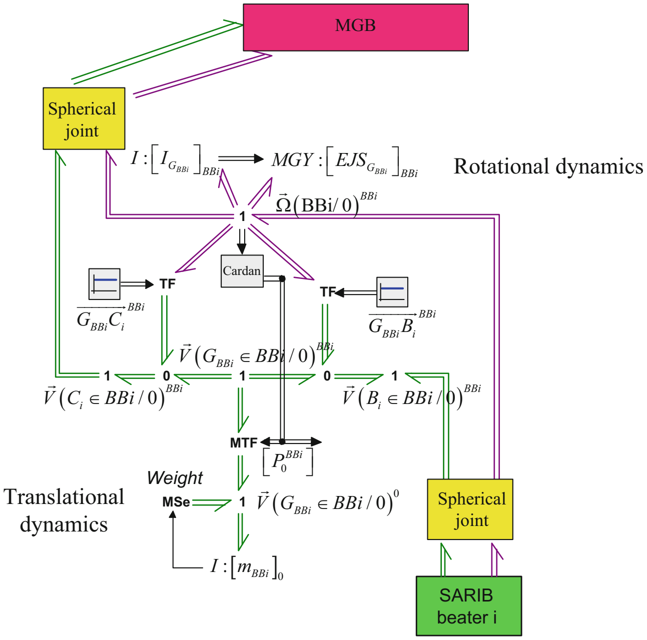

MGB
Spherical joint
$I:\left[I_{G_{BBi}}\right]_{BBi}$
$MGY:\left[EJS_{G_{BBi}}\right]_{BBi}$
Rotational dynamics
$\vec{\Omega}(\mathrm{BBi}/0)^{BBi}$
1
Cardan
TF
TF
$\overrightarrow{G_{BBi}C_i}^{BBi}$
$\overrightarrow{G_{BBi}B_i}^{BBi}$
$\vec{V}(G_{BBi}\in BBi/0)^{BBi}$
1
0
1
0
1
$\vec{V}(C_i\in BBi/0)^{BBi}$
$\vec{V}(B_i\in BBi/0)^{BBi}$
MTF
$\left[P_0^{BBi}\right]$
Weight
Translational dynamics
MSe
1
$\vec{V}(G_{BBi}\in BBi/0)^{0}$
$I:[m_{BBi}]_0$
Spherical joint
SARIB beater i

Fig. 11.21 Bond graph model of MGB bar *i* kinematically linked to its environment

The Revolute Joints

Eight revolute joints are present in the system: two for the universal joint in the attach joint between the fixed frame and the MGB, two for the universal joint in the membrane between the MGB and the fuselage, and four between each SARIB beaters and the fuselage. The bond graph modeling between the SARIB beaters and the fuselage is given in Fig. 11.22. The bond graph models of the other revolute joints are designed in a similar manner.

The upper part of this joint's model corresponds to the angular velocities. The lower part of this joint's model corresponds to the translational velocities. The R/C elements in this model are functionally different. On the one hand, the R/C elements corresponding to the stiffness and damping around the y axis (revolute joint's axis) model the behavior of the torsional spring. On the other hand, the R/C elements with high stiffness and some damping around the other axes model the virtual springs since there are no physical springs in these directions. They permit to block the degree of freedom along and around these axes and make the numerical simulation possible (as explained in Sect. 11.3).

The Spherical Joints

Eight spherical joints are present in the system: four between the SARIB beaters and the MGB bars and also four between the MGB bars and the MGB. The bond graph modeling of the spherical joint between the SARIB beater and an MGB Bar

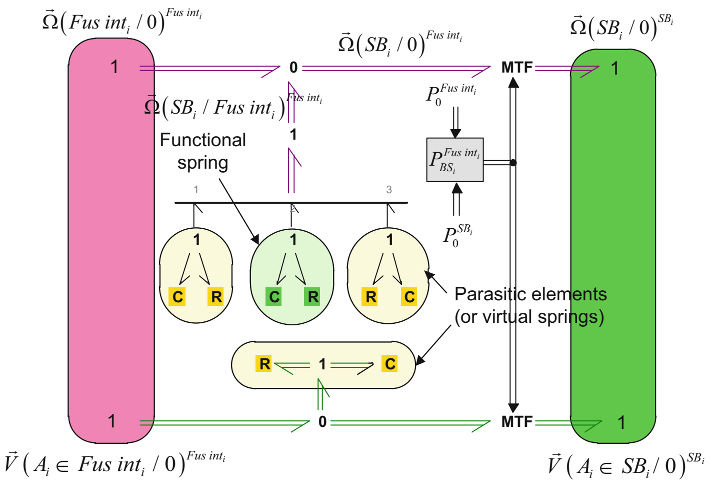

$\vec{\Omega}(Fus\ int_i / 0)^{Fus\ int_i}$
$\vec{\Omega}(SB_i / 0)^{Fus\ int_i}$
$\vec{\Omega}(SB_i / 0)^{SB_i}$
1
0
MTF
1
$\vec{\Omega}(SB_i / Fus\ int_i)^{Fus\ int_i}$
$P_0^{Fus\ int_i}$
Functional spring
1
$P_{BS_i}^{Fus\ int_i}$
1
3
$P_0^{SB_i}$
1
C
R
1
C
R
1
R
C
Parasitic elements (or virtual springs)
R
1
C
1
0
MTF
1
$\vec{V}(A_i \in Fus\ int_i / 0)^{Fus\ int_i}$
$\vec{V}(A_i \in SB_i / 0)^{SB_i}$

Fig. 11.22 Bond graph model of the revolute joint between the SARIB beaters and the fuselage

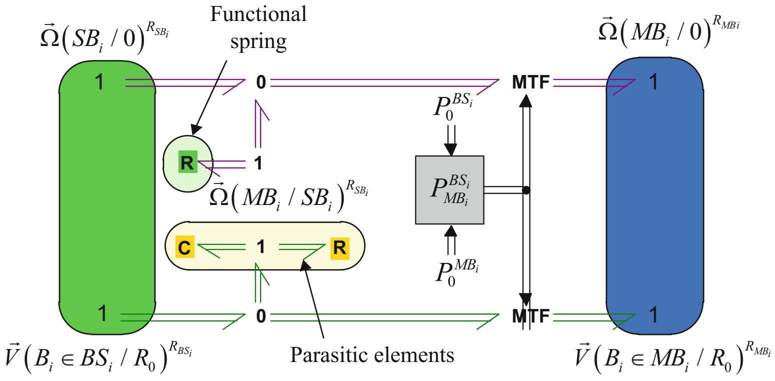

$\vec{\Omega}(SB_i / 0)^{R_{SB_i}}$
Functional spring
$\vec{\Omega}(MB_i / 0)^{R_{MBi}}$
1
0
MTF
1
$P_0^{BS_i}$
R
1
$\vec{\Omega}(MB_i / SB_i)^{R_{SB_i}}$
$P_{MB_i}^{BS_i}$
C
1
R
$P_0^{MB_i}$
1
0
MTF
1
$\vec{V}(B_i \in BS_i / R_0)^{R_{BS_i}}$
Parasitic elements
$\vec{V}(B_i \in MB_i / R_0)^{R_{MB_i}}$

Fig. 11.23 Bond graph model of the spherical joint between an MGB bar *i* and an SARIB beater *i*

is described in Fig. 11.23. The latter shows the blocking of the translational degree of freedom in which the relative translational velocities are close to zero, thanks to the virtual springs. The other spherical joints are built in the same manner.

The Prismatic Joints

The bond graph model of the prismatic joint between the MGB and an intermediate body is given in Fig. 11.24.

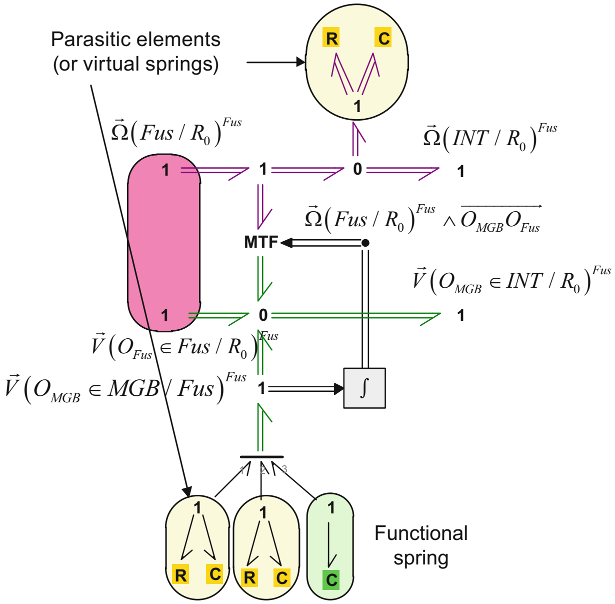

Parasitic elements
(or virtual springs)
R
C
1
$\vec{\Omega}(Fus/R_0)^{Fus}$
$\vec{\Omega}(INT/R_0)^{Fus}$
1
1
0
1
$\vec{\Omega}(Fus/R_0)^{Fus} \wedge \overrightarrow{O_{MGB}O_{Fus}}$
MTF
$\vec{V}(O_{MGB} \in INT/R_0)^{Fus}$
1
0
1
$\vec{V}(O_{Fus} \in Fus/R_0)^{Fus}$
$\vec{V}(O_{MGB} \in MGB/Fus)^{Fus}$
1
∫
1
R
C
1
R
C
1
C
Functional
spring

Fig. 11.24 Bond graph model of the prismatic joint between the MGB and an intermediate body

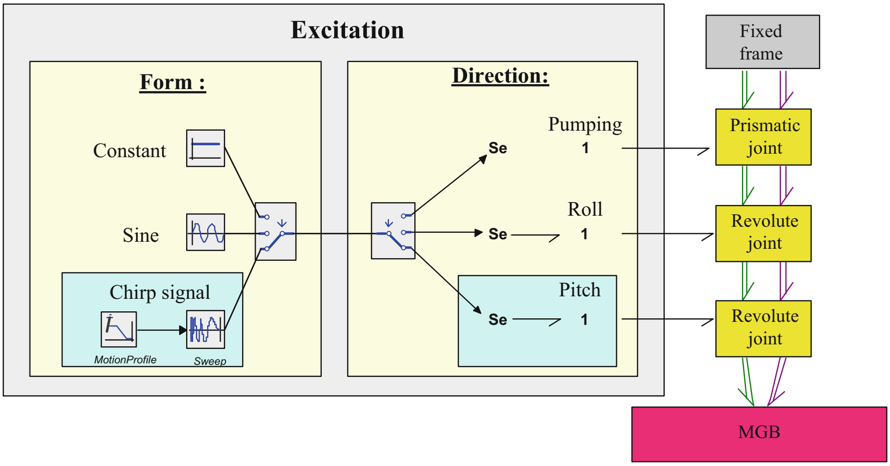


Fig. 11.25 Bond graph model of the actuators of the excitation

11.4.4.2 Excitation Models

The mechanical actions developed by the actuators are modeled by an effort source connected to the junctions **1** which are free in the attach joint. The example of the implementation of an excitation by a swept sine at the pitch joint is shown in Fig. 11.25.

Depending on the selected excitation direction (pumping, roll, or pitch), this source will be, respectively, connected either to the free junction of the prismatic joint, or the revolute joints around the roll or pitch axis. Three choices of excitation forms may also be selected: constant for a natural mode excitation (if this constant is zero) or a static study, sinusoidal, or swept sine for excited modes.

The practical realization of these actuators will be conducted using vibration shakers as we will detail in the experimental validation section.

11.4.4.3 Assembly

These rigid bodies and joints models, as described above, are then connected together according to the architecture defined in the kinematic diagram. The bond graph model of the 3D MGB-fuselage suspension is given below in Fig. 11.26. As expected, the structural aspect of the model is explicit in so far as the structure of the model is similar to the joints graph of the system.

11.4.4.4 Simulation Protocol

Getting Equations

The step of generating the mechanical equations is fully automated and transparent to the user using the 20-sim software. In this step, the solver reads data corresponding to the bond graph model and builds a mathematical problem: the equations of the system.

Resolution of the Equations

The used bond graph software (20-sim) then solves the equations. With the method of singular perturbation, this step resolution can be performed simply using classical integration schemes for solving differential equations ODE solver such as Runge Kutta 4 (RK4). However, the integration scheme that has been used is the Backward Differentiation Formula (BDF). It was preferred as it allows a simulation of a stiff problem which is much faster than explicit solvers (like RK4). It is Gear's backward differentiation formula which is an implicit multi-step variable-order method. The implementation of the integrator is based on the Dassl library which has been developed in [10]. This simulation algorithm has required two parameters: the absolute and relative integration which has been tuned on the default values (absolute integration error = e-5 and relative integration error = e-5). This method chooses the step and the order in function of the error. Moreover, the BDF solver guarantees the numerical stability of the solution. From the mathematical solution, the solver computes and communicates the results requested by the user. 20-sim

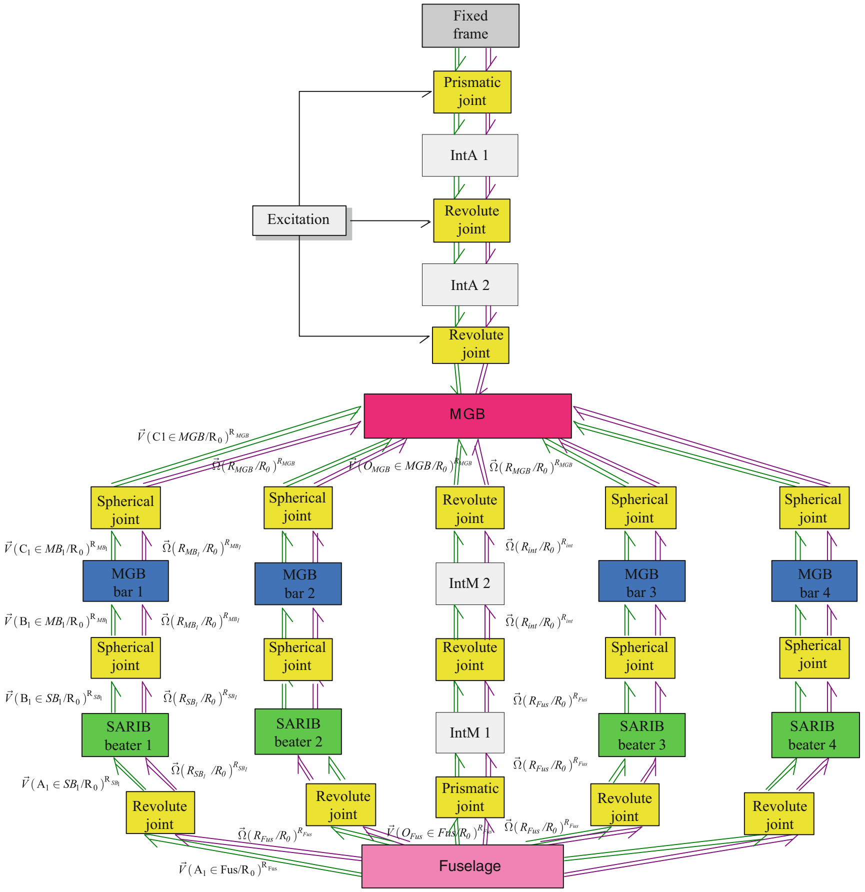

Fixed frame
Prismatic joint
IntA 1
Excitation
Revolute joint
IntA 2
Revolute joint
MGB
$\vec{V}(C1 \in MGB/R_0)^{R_{MGB}}$
$\vec{\Omega}(R_{MGB}/R_0)^{R_{MGB}}$
$\vec{V}(O_{MGB} \in MGB/R_0)^{R_{MGB}}$
$\vec{\Omega}(R_{MGB}/R_0)^{R_{MGB}}$
Spherical joint
Spherical joint
Revolute joint
Spherical joint
Spherical joint
$\vec{V}(C_1 \in MB_1/R_0)^{R_{MB_1}}$
$\vec{\Omega}(R_{MB_1}/R_0)^{R_{MB_1}}$
$\vec{\Omega}(R_{int}/R_0)^{R_{int}}$
MGB bar 1
MGB bar 2
IntM 2
MGB bar 3
MGB bar 4
$\vec{V}(B_1 \in MB_1/R_0)^{R_{MB_1}}$
$\vec{\Omega}(R_{MB_1}/R_0)^{R_{MB_1}}$
$\vec{\Omega}(R_{int}/R_0)^{R_{int}}$
Spherical joint
Spherical joint
Revolute joint
Spherical joint
Spherical joint
$\vec{V}(B_1 \in SB_1/R_0)^{R_{SB_1}}$
$\vec{\Omega}(R_{SB_1}/R_0)^{R_{SB_1}}$
$\vec{\Omega}(R_{Fus}/R_0)^{R_{Fus}}$
SARIB beater 1
SARIB beater 2
IntM 1
SARIB beater 3
SARIB beater 4
$\vec{\Omega}(R_{SB_1}/R_0)^{R_{SB_1}}$
$\vec{\Omega}(R_{Fus}/R_0)^{R_{Fus}}$
$\vec{V}(A_1 \in SB_1/R_0)^{R_{SB_1}}$
Revolute joint
Revolute joint
Prismatic joint
Revolute joint
Revolute joint
$\vec{\Omega}(R_{Fus}/R_0)^{R_{Fus}}$
$\vec{V}(O_{Fus} \in Fus/R_0)^{R_{Fus}}$
$\vec{\Omega}(R_{Fus}/R_0)^{R_{Fus}}$
Fuselage
$\vec{V}(A_1 \in Fus/R_0)^{R_{Fus}}$

Fig. 11.26 Bond graph model of the passive MGB-fuselage suspension

allows the user to evaluate and easily plot any of the physical quantities involved in the bond graph model of the system.

Post-Processing

From the mathematical solution, the solver computes and communicates the results requested by the user. 20-sim allows the user to evaluate and easily plot any of the physical quantities involved in the bond graph model of the system. In addition, graphs can be exported to various types of image-formats or data with .csv or .xls file. The data exportation allows for the comparison of results from different sources in the same graphic interface (for example, Matlab).

11.5 Simulation and Verification of the Vibration Absorber Suspension Model

11.5.1 Use of MapleSim for BG Model Verification

So as to verify the BG suspension model, simulation results conducted with MapleSim software (which contains a library dedicated to multibody systems) have been used. The simulation parameters used for both Bond graph conducted with 20-sim and MapleSim are summarized in Table 11.1.

11.5.2 Comparison of Simulation Numerical Results

The vibration analysis of the system is mainly evaluated by analyzing the acceleration of the fuselage points and forces transmitted from the MGB to the fuselage at the SARIB beater/ fuselage joint. Simulation results with different types of excitation (pumping, roll, and pitch) are the following and will permit to analyze the dynamic behavior of the suspension. The results presented were compared with a multibody software MapleSim.

11.5.2.1 Dynamic Behavior with Pumping Excitation

Acceleration of the Point O_F of the Fuselage

The point O_F is defined as the intersection of the MGB's axis and the upper plane of the fuselage. It is a central point on the fuselage that can be measured.

The vertical component of the acceleration of this point is given in Fig. 11.27

Table 11.1 Simulation comparison

Software	MapleSim	Bond graph conducted with 20-sim
Solver	Rosenbrock (stiff)	Backward differentiation formula (BDF)
Method	Linear graph theory	Singular perturbation
Number of coordinates	84 (14 bodies)	84 (14 bodies)
Number of constraints	74 (8 spherical joints, 8 revolute joints, 2 prismatic joints)	0
DoF	10 (6 + 4)	84

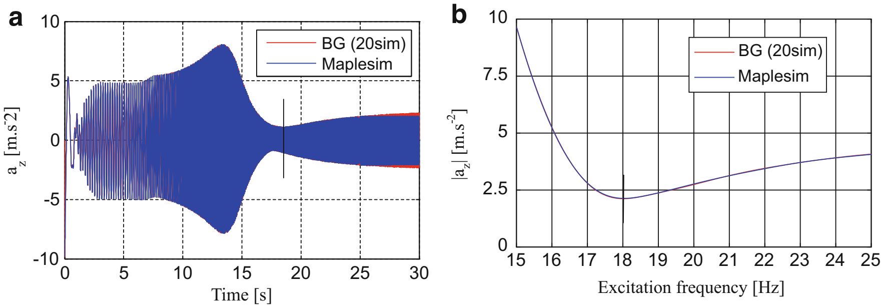


Fig. 11.27 Vertical component of the acceleration of the point O_F for a pumping excitation (**a**) with chirp signal, (**b**) with sine signal

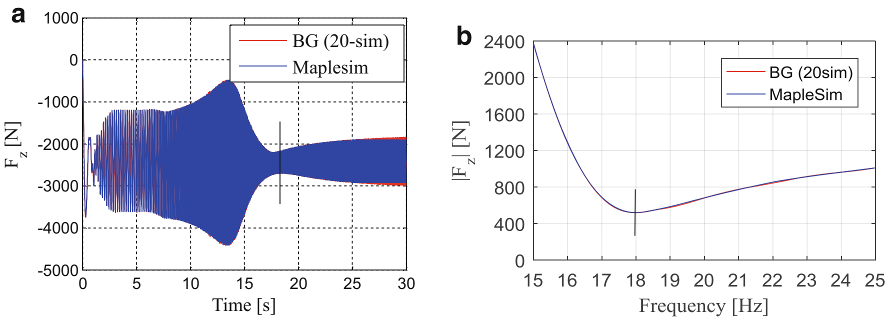


Fig. 11.28 Vertical force transmitted at the revolute joint A1 for a pumping excitation (**a**) with chirp signal, (**b**) with sine signal

An anti-resonance frequency is observed at 18 Hz. The curve obtained with MapleSim is very close to the curve obtained with the simulation of the bond graph model with 20-sim.

Forces Transmitted to the Fuselage

The forces transmitted to the fuselage at the revolute joints between the SARIB beaters and the fuselage expressed in the intermediate pins fuselage were determined as shown in Fig. 11.28.

As shown in the above figures, the curves of the vertical forces transmitted at the revolute joints are logically similar to the curve of the acceleration of the point O_F. Through the action of inertial forces, the transmission forces of the MGB to the fuselage are reduced at a certain frequency called anti-resonance frequency. They present an anti-resonance at about 18 Hz. SARIB beaters thus play their roles.

We can observe in Fig. 11.29 that the anti-resonance phenomenon observed in the force along x-axis in the intermediate fuselage frames does not occur at the same

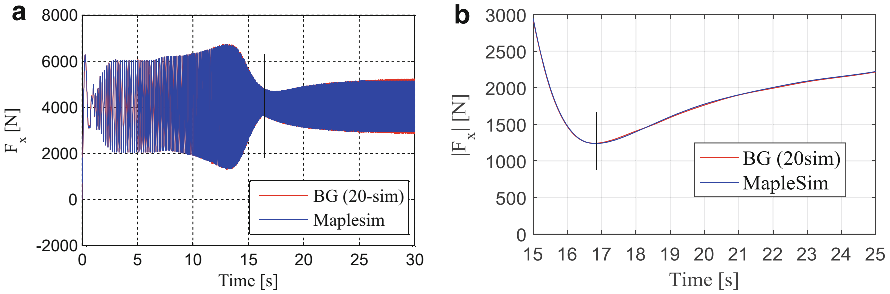


Fig. 11.29 Force along *x*-axis in the intermediate fuselage frame at the revolute joint A1 for a pumping excitation (**a**) with chirp signal, (**b**) with sine signal

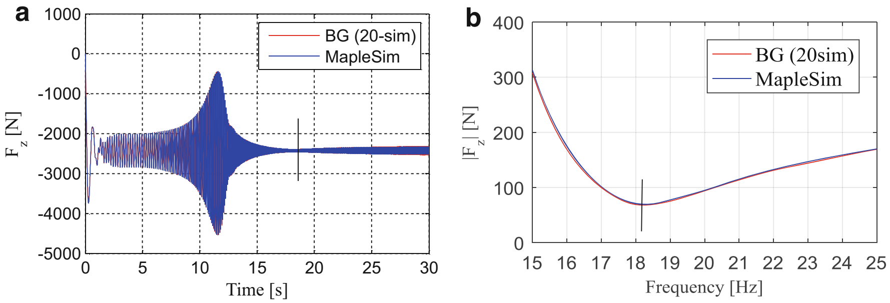


Fig. 11.30 Vertical force at the revolute joint A1 for a roll excitation (**a**) with chirp signal, (**b**) with sine signal

frequency as the one which had been observed for the transmission of vertical forces to the fuselage (for reminding around 18 Hz). The frequency of anti-resonance is around 16.8 Hz. The designer will have to monitor the amplitude of the component along *x*-axis of these forces which are not minimal at the anti-resonant frequency for the vertical forces transmitted to the fuselage.

11.5.2.2 Dynamic Behavior with Roll Excitation

The vertical force transmitted to the fuselage at the revolute joint A1 between the SARIB beaters and the fuselage expressed in the intermediate fuselage frames was determined as shown in Fig. 11.30. As previously, an anti-resonance phenomenon is observed at the frequency of 18.2 Hz.

In Fig. 11.31, we can notice the phase shift between the vertical force at the revolute joint A1 and the revolute joint A2 when the MGB is submitted to a roll excitation.

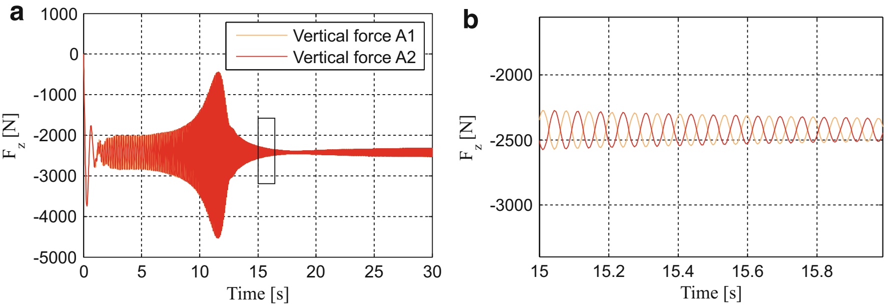


Fig. 11.31 (**a**) Vertical forces at the revolute joints A1 and A2 with a roll excitation, (**b**) zoom

11.5.2.3 Dynamic Behavior with Pitch Excitation

Similar results are observed for a pitch excitation but are not presented for the sake of concision.

11.5.3 About the Modeling and Simulation of a Semi-Active Suspension

In the frame of this research work, a semi-active suspension has also been modeled and simulated. The tuning of the anti-resonant frequency is then obtained by changing the position of the moving masses on SARIB beaters. The objective of the semi-active suspension is to improve the reduction of vibrations despite some fluctuations of the frequency of the rotor excitation by maintaining the positioning of the anti-resonant frequency of the suspension on the rotor excitation frequency. For this purpose, the position of the moving masses driven by electric DC motors and screw-nut systems has been optimized and PI controllers have permitted to position the moving masses to the setpoints generated by a control algorithm. Models and simulations on this semi-active suspension model have been conducted in [8] and are not here developed for the sake of concision. This study shows that multiphysic properties of the bond graph approach enable to add easily electrical actuators and a control command structure directly in a unique model.

11.6 Conclusion

A structural and modular approach with bond graphs has been presented in this chapter so as to model and simulate the dynamics of a complex mechatronic multibody system: a helicopter's vibration absorber system.

Three main methods (ZCP, R/C elements, and Lagrange Multipliers) of simulating BG models of multibody systems have been presented. Both the conditions of application and practical rules for applying these methods with graphical bond graph software, such as 20-sim software, have been given.

The proposed methodology consists of the assembly of modules (these components are rigid bodies and compliant kinematic joints) in a similar manner to the real structure of the physical system. This assembly can be easily conducted with the help of a well-structured library of components. Therefore, the constructed multibody dynamic model permits the representation of a complex system and the bond graph model highlights the topology of systems.

The parasitic compliant elements in the kinematic joints have been used to avoid dealing with kinematic constraint equations and consequently to have only ODE systems to solve instead of DAEs. Moreover, this method has suppressed the constraints linked to the causality imposed by the use of vector bonds.

The simulation results of the bond graph model of the studied suspension have been conducted with dedicated software for bond graphs such as 20-sim and have been verified with another multibody tool (MapleSim) for three excitations (pumping, roll, and yaw). These simulations have permitted the identification and quantification of the anti-resonance which were sought so as to filter the vibrations coming from the rotor to the fuselage.

It has been shown than the BG should permit a structural and modular modeling of complex 3D mechatronic systems. It is worthwhile to note that the most recent structural commercial modeling tools such as MapleSim and others could also be used for the modeling of the studied suspension. However, to take benefit from the unified power-based approach and exploit specific properties of BG, bond graphs have been chosen as the modeling language. Future works are being conducted so as to exploit two promising BG exploitations which are the analysis of energetic flow transiting among components and the control of multibody systems by means of inversion techniques permitted by BG structural analysis [21].

Acknowledgments This research work received support from the Chair "Dynamics of complex mechanical systems - EADS Corporate Foundation - Arts et Métiers ParisTech and Ecole Centrale de Marseille." Special thanks to Paul B.T. Weustink who works at Controllab Products for his help with the use of complementary tools from 20-sim software.

References

1. Abadie, V., Guillemard, F., & Rault, A. (2000). Apport du bond graph dans la démarche mécatronique appliquée à l'automobile. In G. Dauphin-Tanguy (Ed.), *Les bond graphs*. Paris: Hermès Sciences Europe Ltd. ISBN:2-7462-0158-5.
2. Amerongen, J. V., & Breedveld, P. (2003). Modelling of physical systems for the design and control of mechatronic systems. *Annual Reviews in Control, 27*(1), 87–117.
3. Behzadipour, S., & Khajepour, A. (2005). Causality in vector bond graphs and its application to modeling of multi-body dynamic systems. *Simulation Modelling Practice and Theory, 14*, 279–295.

4. Bonderson, L. S. (1975). Vector bond graphs applied to one-dimensional distributed systems. *Journal of Dynamic Systems, Measurement, and Control, 1*, 75–82.
5. Borutsky, W. (2010). *Bond graph methodology—Development and analysis of multidisciplinary dynamic system models* (Springer ed.th ed.). London: Springer.
6. Borutzky, W. (1995). Representing discontinuities by means of sinks of fixed causality. In: *Proceedings of the 1995 Western Simulation Multiconference*, Las Vegas, Nevada.
7. Bos, A. M. (1986). *Multibody systems in terms of multibond graphs with application to a motorcycle multibody system*. PhD thesis, University of Twente, Enschede, The Netherlands.
8. Boudon, B., Malburet, F., & Carmona, J. C. (2016). Simulation of a helicopter's main gearbox semiactive suspension with bond graphs. *Multibody System Dynamics,* 1–31.
9. Breedveld, P. C. (1985). Multibond graph elements in physical systems theory. *Journal of the Franklin Institute, 319*, 1–36.
10. Brenan, K. E., Campbell, S. L., & Petzold, L. R. (1996). *Numerical solution of initial-value problems in differential-algebraic equations.* SIAM. ISBN:978-0-89871-353-4.
11. Cacho, R., Felez, J., & Vera, C. (2000). Deriving simulation models from bond graphs with algebraic loops: The extension to multibond graph systems. *Journal of the Franklin Institute, 337*, 579–600.
12. Ersal, T., Stein, J. L., & Louca, L. S. (2004). A bond graph based modular modeling approach towards an automated modeling environment for reconfigurable machine tools. In *International conference on integrated modeling and analysis in applied control and automation* (2004).
13. Felez, J., Romero, G., Maroto, J., & Martinez, M. L. (2011). Simulation of multi-body systems using multi-bond graph. In W. Borutsky (Ed.), *Bond graph modeling of engineering systems.* Springer. ISBN:978-1-4419-9367-0.
14. Felez, J., Vera, C., San Jose, I., & Cacho, R. (1990). BONDYN: A bond graph based simulation program for multibody systems. *Journal of Dynamic Systems, Measurement, and Control, 112*, 717–727.
15. Henderson, J.-P. (2012). *Vibration isolation for rotorcraft using electrical actuation*. PhD thesis, Mechanical Engineering, University of Bath.
16. Karnopp, D. (1969). Power-conserving transformations: Physical interpretations and applications using bond graphs. *Journal of the Franklin Institute, 288*, 175–201.
17. Karnopp, D. C., & Margolis, D. L. (1979). Analysis and simulation of planar mechanisms systems using bond graphs. *Journal of Mechanical Design, Transactions of the ASME, 101*, 187–191.
18. Krysinski, T., & Malburet, F. (2006). *Mechanical vibrations: Active and passive control.* London: Wiley-ISTE.
19. Lalanne, C. (2014). *Mechanical vibration and shock analysis*. London: Wiley-ISTE.
20. Marquis-Favre, W. (1997). *Contribution à la représentation bond graph des systèmes mécaniques multicorps*. PhD thesis, INSA de Lyon.
21. Marquis-Favre, W., & Jardin, A. (2011). Bond graphs and inverse modeling for mechatronic system design. In: W. Borutsky (Ed.), *Bond graph modeling of engineering systems*. Berlin: Springer.
22. Marquis-Favre, W., & Scavarda, S. (1998). Bond graph representation of multibody systems with kinematic loops. *Journal of the Franklin Institute, 335B*, 643–660.
23. Paynter, H. M. (1961). *Analysis and design of engineering systems*. Cambridge: MIT Press.
24. Rideout, G. (2004). *System partitioning and physical domain proper modeling through assessment of power-conserving model structure*. PhD thesis, University of Michigan.
25. Romero, G., Felez, J., Maroto, J., & Cabanellas, J. M. (2007). A minimal set of dynamic equations in systems modelled with bond graphs. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 221*(1), 15–26.
26. Rosenberg, R. C. (1972). Multiport models in mechanics. *Journal of Dynamic Systems, Measurement, and Control, 94*, 206–212.

27. Sass, L., McPhee, J., Schmitke, C., Fisette, P., & Grenier, D. (2004). A comparison of different methods for modelling electromechanical multibody systems. *Multibody System Dynamics, 12*, 209–250.
28. Schiehlen, W. (2007). Research trends in multibody system dynamics. *Multibody System Dynamics, 18*, 3–13.
29. Tiernego, M. J. L., & Bos, A. M. (1985). Modelling the dynamics and kinematics of mechanical systems with multibond graphs. *Journal of the Franklin Institute, 319*, 37–50.
30. van Dijk, J., & Breedveld, P. C. (1991). Simulation of system models containing zero-order causal paths—I. Classification of zero-order causal paths. *Journal of the Franklin Institute, 328*, 959–979.
31. van Dijk, J., & Breedveld, P. C. (1991). Simulation of system models containing zero-order causal paths—II Numerical implications of class 1 zero-order causal paths. *Journal of the Franklin Institute, 328*, 981–1004.
32. Wang, J., Gosselin, C., & Cheng, L. (2001). Modeling and simulation of robotic systems with closed kinematic chains using the virtual spring approach. *Multibody System Dynamics, 7*, 145–170.
33. Xu, W., Liu, Y., Liang, B., Wang, X., & Xu, Y. (2010). Unified multi-domain modelling and simulation of space robot for capturing a moving target. *Multibody System Dynamics, 23*, 293–331.
34. Zeid, A., & Chung, C.-H. (1992). Bond graph modeling of multibody systems: a library of three-dimensional joints. *Journal of the Franklin Institute, 329*, 605–636.
35. Zeid, A., & Overholt, J. L. (1995). Singularly perturbed formulation: Explicit modeling of multibody systems. *Journal of the Franklin Institute, 332*.

Chapter 12
Manual Driven Wheelchair Bond Graph Modelling

Abdennasser Fakri and Japie Petrus Vilakazi

12.1 Introduction

The motivation for this work derives from attempts to employ few portable mechatronic drive modules (MDM) that can be coupled when necessary to a large scale of manual propelled wheelchair (MPW) systems with the aims of improving propulsion efficiency. The engagement of such a module on MPW is illustrated in Fig. 12.1. This yields good advantage to the system since the weight of the module is imposed on the module itself, thus prevents deformation of the structure. In the next, the modelling of such a system is presented in terms of Bond graphs.

Indeed, the investigation on improving specifically the ordinary MPW propulsion method [9] (wheelchairnet.org) has become increasingly imperative. This is due to the growing population of the manual wheelchair users and the requirements for efficient mobility to maintain a quality of live equivalent to the general population.

Currently, when creating a physical device, one of the major tasks is the implementation of the model that integrates the control system, sensors, and actuators' dynamics in order to allow the simulation software tools to be integrated in the modelling process. The Bond graph methodology is a very well-suited graphical tool for modular modelling approach based on energy transfer in multi-domain systems [8]. We assume that the reader has some knowledge on the BG modelling method employed; therefore a brief review is given below.

A. Fakri (✉)
ISYS Lab ESIEE Paris Département Systèmes Electroniques, Université Paris-Est,
Cite Descartes, Bd Blaise Pascal – BP 99, 93162 Noisy le Grand Cedex, France
e-mail: fakria@esiee.fr

J.P. Vilakazi
Tshwane University of Technology, 993 Khama Street, Standerton 2430, South Africa
e-mail: pvilakazi@sudorcoal.co.za

W. Borutzky (ed.), *Bond Graphs for Modelling, Control and Fault Diagnosis of Engineering Systems*, DOI 10.1007/978-3-319-47434-2_12

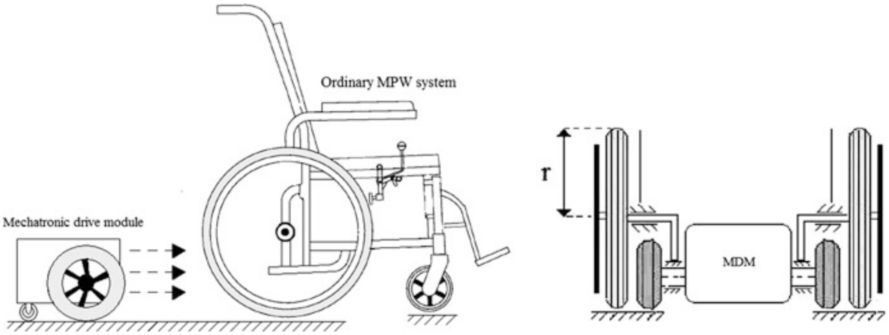


Fig. 12.1 Manual propelled wheelchair (MPW) system with the mechatronic module

BGs models developed represent the elementary transfer and storage of energy through the different dynamic component on the system. The 0 and 1 junctions are used to denote common effort and flow cases, thus permitting the extraction of balance junction dynamic equations which in turn enables derivation of mathematical models of the system. Each graphical bond carries power that is product of two variables: flow and effort. Power is transferred between different domains with the use of *gyrator* and *transformer* elements. For control purposes, sources of *flow* or *effort* are used as input to the system. Some part of the transferred energy is dissipated in resistive elements, Bond graphically denoted by symbol *(R)* or stored in kinetic or potential form in, respectively, inertial *(I)* or capacitive (*C*) elements. For more information, readers may refer to [2, 3, 8]

If properly applied, the Bond graph methodology enables one to develop graphical models that are consistent with the first principle of energy conservation without having the need to start with establishing and reformulating equations. In the next sections, the studied systems test bed structures are analyzed and model components elaborated. Mathematical models in terms of state space equations or balance junction differential equations are extracted. Some simulation results of these systems are also presented. First, we present the decomposition of the MPW, its BG model, state space model, and simulation results in an obstacle avoidance predetermined trajectory. Note that the simulation was carried out on Matlab Simulink using the BG equivalent block diagram constructed directly from the BG model. This method is presented in [4]. Then, we present the schematic of the MDM, its BG model as well as simulation results in a predetermined trajectory showing behavior of this system in open and close-loop control. Finally we present the whole system (MDM coupled on MPW) BG model and simulation in the same trajectory. The last section gives some conclusions on this work. The parameters used for the simulation are presented at the end of the chapter. This work has been the subject of two conference papers and a memory submitted for the degree MTech [5–7].

12.2 Manual Propelled Wheelchair

12.2.1 Description and Test Bed

MPW is required in hospitals and old age institutions to transport patients who are too unwell to walk [1]. The user can maneuver the chair by turning the hand push rims attached to the rear wheels, as shown in Fig. 12.1. For users who cannot operate the MPW, handles are made available for assistance from an attendant.

An illustration of an ordinary MPW test bed structure used in this modelling is depicted in Fig. 12.2. It essentially consists of two casters and manual rear wheels. An approach of a two wheel drives robotic system has been applied in this modelling with the caster wheels lumped together and assumed to be imposing resistive force to the systems flow. V_{CG} and ω_{CG} represent the center of mass velocity and inertial mass rotation, ω_1 and ω_r indicate the angular velocity of the left and right wheels, respectively, and rear manual wheel radius and wheelchair width are represented by (***r****)* and *(Lw)*, respectively. The below matrices (12.1) show the kinematic mathematical model of the system, where θ is the orientation angle, x and y show the systems geometric position, respectively:

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = r \begin{bmatrix} \frac{\cos\theta}{2} & \frac{\cos\theta}{2} \\ \frac{\sin\theta}{2} & \frac{\sin\theta}{2} \\ \frac{2}{Lw} & \frac{-2}{Lw} \end{bmatrix} \begin{bmatrix} \omega r \\ \omega l \end{bmatrix} \tag{12.1}$$

The components contributing to the dynamic behavior of this ideal MPW system have been identified from the point at which torque (Se) is applied and found to be:

- *Wheels rotational inertia, Inertial moment*
- *Wheel spokes stiffness:* this refers to the spring/damper imposed by the air inflated rear wheels. This exists from the center of the wheel to the ground.

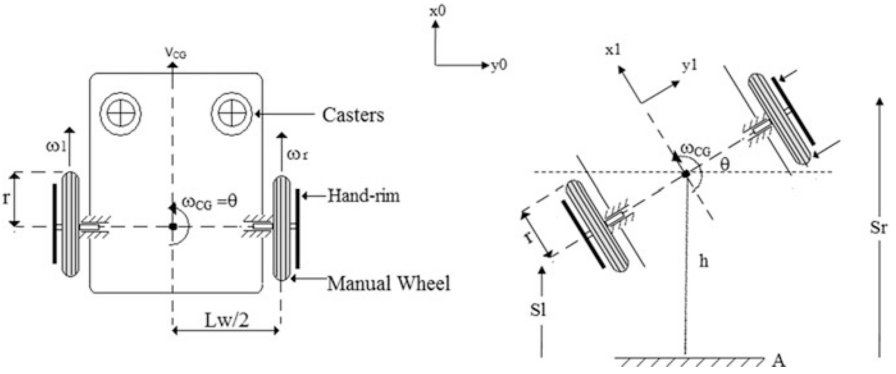


Fig. 12.2 Top view of manual propelled wheelchair (MPW) system

- *Frictional losses:* the frictional losses were considered and said to exist at the rear wheels circumference to ground. The resistive force (loss of energy) imposed by the front caster wheels during rotation was also considered.
- *Mass:* includes weight of the chair and the assumed mass of the user. This point is located at the center of gravity where the systems average mass is obtained. In reducing the complexity of the model, the center of gravity has been assumed to be occurring at the same x-axis with the rear propulsion wheels, thus permitting rotation of the system with reference to that pivot as indicated by ω_{cg} in Fig. 12.2. Point A was considered as the initial (x_0, y_0) and reference position of the wheelchair, and the system said to have moved h distance and rotated at the center of gravity as depicted in the same figure.

The left and right wheels' displacements with reference to point A are given by Eqs. (12.2) and (12.3), respectively.

$$S_1 = h - \frac{Lw}{2} \sin\ \theta \tag{12.2}$$

$$S_r = h + \frac{Lw}{2} \sin\ \theta \tag{12.3}$$

From the BG analogies, integration of the flow equations yields the displacement equations of the system. Therefore using the above equations, we obtained the required flow Eqs. (12.4) and (12.5) to construct the BG model of the wheelchair structure.

$$\omega_r = \dot{S}_r = \dot{h} - \dot{\theta}\frac{Lw}{2} \cos\ \theta \tag{12.4}$$

$$\omega_r = \dot{S}_r = \dot{h} + \dot{\theta}\frac{Lw}{2} \cos\ \theta \tag{12.5}$$

Using the constitutive laws of the one-port I-element we describe the motion in terms of the velocity of center of gravity (V_{CG}) and the angular velocity $\omega_{\mathrm{CG}} = \dot{\theta}$ as depicted in Fig. 12.2. The average force on the wheelchair is equal to the rate of change of linear momentum that is related to V_{CG} by the systems total mass in Eq. (12.6).

$$V_{\mathrm{CG}} = \frac{P_{\mathrm{CG}}}{m} \tag{12.6}$$

The average torque at the center of gravity is equal to the rate of change of angular momentum which is related to $\omega_{\mathrm{CG}} = \dot{\theta}$ by the systems moment of inertia in Eq. (12.7).

$$\omega_{CG} = \frac{P_\theta}{J_{\mathrm{chair}}} \tag{12.7}$$

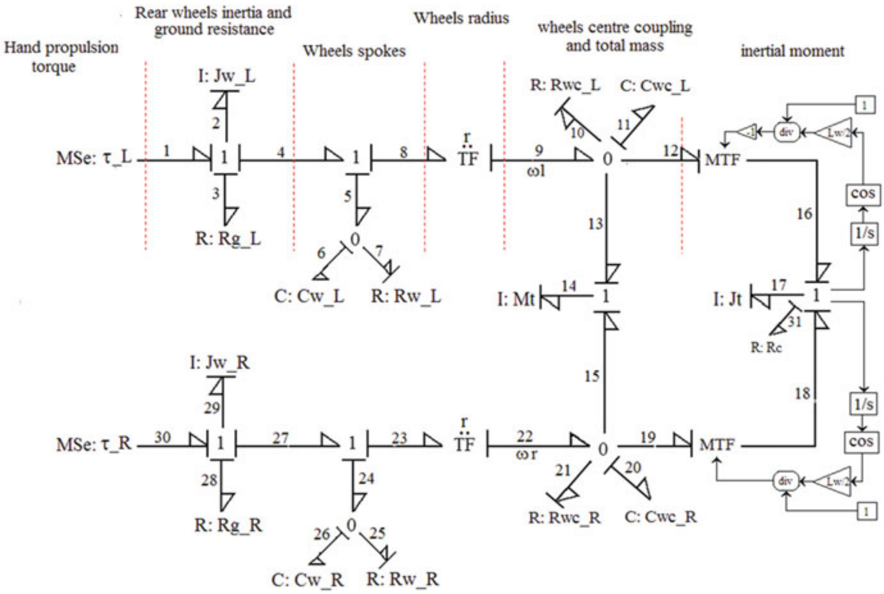


Fig. 12.3 A bond graph representation of the MPW system

The BG model of the MPW depicted in Fig. 12.3 is constructed from the above described components contributing to the dynamic behavior of the system. The causality of the MPW system bonds is adjusted according to the constrains on the body. The energy produced by the hand propulsion torque on rear wheel is transferred to the system structure as indicated by the bond half-arrows in Fig. 12.3. All the external torques and rotational inertia have been connected to the 1-junctions, L and R represent the left and right sides of the system. The constitutive equations of these junctions where the mass and inertial moment of the entire system (rigid body) have been connected to represent the Newton's second law. The ideal transducer TF indicates the transformation of energy through the radius of the rear wheels.

A remarkable feature of BGs is that an inspection of causal path can reveal information about structural control of the system behavior. The survey of BG capabilities shows how this modelling technique serves as a core model representation, from which different information can be derived depending on purpose of the study.

12.2.2 Bond Graph Model of MPW

The test bed shown in Fig. 12.2 is represented in an integral causalled BG shown in Fig. 12.3. Where: *MSE:τ* [Nm] shows the source of effort generated from the

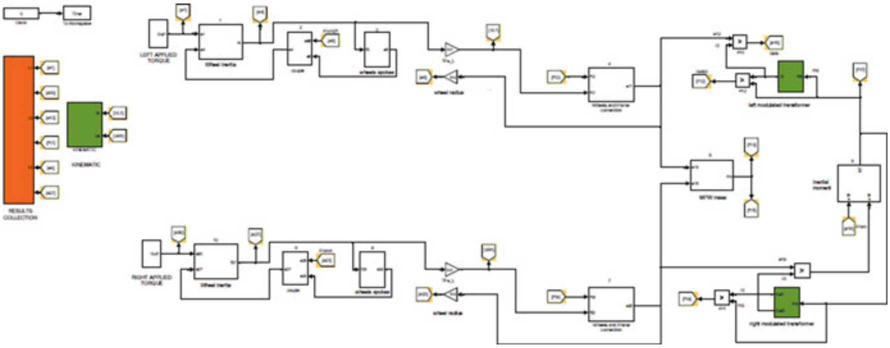


Fig. 12.4 Block diagram structure of the MPW model

tangential force applied by the user on the wheels, *L* [left] or *R* [right], *Jw_L* [kg m^2] is the rotational inertia of the rear wheels, *Rg* is the rear wheel to ground frictional constant, *Mt* [kg] is the total mass of the system, *Jt* [kg m^2] is the total inertial moment of the system, *Rc* is the losses on caster wheels during motion, *Rwc, Cwc* shows the wheel to structure coupling stiffness and elasticity, respectively, whilst *Cw and Rw* represent the rear wheels spring/damper spokes constants, respectively. The non-linear MTF represents the energy transferred to structure depending on the applied force.

A block diagram model is derived as depicted in Fig. 12.4, following the procedures in section directly from the BG model developed in [3].

12.2.3 Mathematical Models of the MPW

The state space equations used by control engineers for system analysis and simulations are extracted directly from the BG (Fig. 12.3). These can be used for controllability, observability, and stability studies of the system. Computer software like CAMP-G, 20sim, and Symbols can be used to extract these equations automatically. Equation (12.8) shows the state space representation of MPW where *x, u*, and *y* are the system state, input, and output, respectively. These Eqs. (12.8), (12.9), (12.10), (12.11), (12.12), (12.13), (12.14), and (12.15) are directly derived from the BG model in Fig. 12.3.

$$\dot{p}_2 = \tau l - \frac{p_2}{Jw\_L} Rg\_L - \frac{q_{11}}{Cwc\_L} k1 - \frac{q_6}{Cw\_L} \tag{12.8}$$

$$\dot{q}_6 = \frac{p_2}{Jw\_L} - \frac{q_6}{Cw\_LRw\_L} \tag{12.9}$$

$$\dot{q}_{11} = \frac{p2}{Jw\_L}k4 - \frac{q11}{Cwc\_L}R3 - \frac{p17}{Jtk2} - \frac{p14}{Mt} \tag{12.10}$$

$$\dot{p}14 = \frac{q11}{Cwc\_L} + \frac{q20}{Cwc\_R} \tag{12.11}$$

$$\dot{p}_{17} = \frac{q20}{k3Cw\_R} - \frac{q11}{Cw\_L}k3 \tag{12.12}$$

$$\dot{q}20 = \frac{p29}{Jw\_R}k4 - \frac{p14}{Mt} - \frac{p17}{Jtk4}k3 - \frac{q20}{R4Cw\_R} \tag{12.13}$$

$$\dot{q}25 = \frac{p29}{Jw\_R} - \frac{q25}{Rw\_RCw\_R} \tag{12.14}$$

$$\dot{p}_{29} = \tau R - \frac{p_{29}}{Jw\_R}Rg\_R - \frac{q_{20}}{Cwc\_R}k4 - \frac{q_{25}}{Cw\_R} \tag{12.15}$$

Equations (12.16) and (12.17) show the state space representation of MPW where *x, u*, and *y* are the system state, input, and output, respectively.

$$\dot{x} = [\mathrm{A}]\,x + [\mathrm{B}]\,u \tag{12.16}$$

$$\begin{bmatrix} e_2 \\ f_6 \\ f_{11} \\ e_{14} \\ e_{17} \\ f_{20} \\ f_{25} \\ e_{29} \end{bmatrix} = \begin{bmatrix} \dot{p}_2 \\ \dot{q}_6 \\ \dot{q}_{11} \\ \dot{p}_{14} \\ \dot{p}_{17} \\ \dot{q}_{20} \\ \dot{q}_{25} \\ \dot{p}_{29} \end{bmatrix} = \begin{bmatrix} \frac{-Rg\_L}{JW\_L} & \frac{-1}{Cw\_L} & \frac{-k1}{Cwc\_L} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{JW\_L} & \frac{-1}{Rw\_LCw\_L} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{k1}{JW\_L} & 0 & \frac{-Rwc\_L}{Cwc\_L} & \frac{-1}{Mt} & \frac{-1}{k2Jt} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{Cwc\_L} & 0 & 0 & \frac{1}{Cwc\_R} & 0 & 0 \\ 0 & 0 & \frac{-k2}{Cwc\_L} & 0 & 0 & \frac{1}{k3Cwc\_R} & 0 & 0 \\ 0 & 0 & 0 & \frac{-1}{Mt} & \frac{-1}{k3Jt} & \frac{-1}{Rwc\_RCwc\_R} & 0 & \frac{-k4}{Mt} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{Rw\_RCw\_R} & \frac{1}{JW\_R} \\ 0 & 0 & 0 & 0 & 0 & \frac{-k4}{Cwc\_R} & \frac{-1}{Cw\_R} & \frac{-Rg\_R}{JW\_R} \end{bmatrix}$$

$$\begin{bmatrix} p_2 \\ q_6 \\ q_{11} \\ p_{14} \\ p_{17} \\ q_{20} \\ q_{25} \\ p_{29} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tau L \\ \tau R \end{bmatrix}$$

$$y = [C]\,x \tag{12.17}$$

$$\begin{bmatrix} \omega_L \\ \omega_R \\ v_{CG} \\ \theta_{CG} \end{bmatrix} = \begin{bmatrix} f_4 \\ f_{27} \\ f_{14} \\ f_{17} \end{bmatrix} = \begin{bmatrix} \frac{1}{Jw\_L} & 0 & 0 & 0 \\ 0 & \frac{1}{Jw\_R} & 0 & 0 \\ 0 & 0 & \frac{1}{Mt} & 0 \\ 0 & 0 & 0 & \frac{1}{Jt} \end{bmatrix} \begin{bmatrix} p_2 \\ p_{29} \\ p_{14} \\ p_{17} \end{bmatrix}$$

The dimension of the system is given by the sum of C and I in integral causality. The causal model proposed MPW is an 8th order. This is also the dimension of the state vector **X**.

12.2.4 Simulation Results of the MPW

The general characteritics and values of the elements are presented Table 12.1. The forces applied to the hand rim during propulsion is sinusoidal and includes rapid rate of loading in the beginning of the push leading to an impact spike and followed by more gradual application and release of force. For the purpose of this study, the system was simulated with step input signals (representing the manual torque) to produce an obstacle avoidance predetermined trajectory.

The main objective was to illustrate the behavior of the MPW system without the MDM engaged on it. A combination of step signals in the range of 4–10 Nm was applied on both wheels as depicted in Fig. 12.5, position and orientation (with reference to the x-axis) were observed. The profile of input torque signals (assumed to be produced by the tangential force exerted by the user) within the range of 2–10 Nm was applied on the MPW rear wheels and depending on the desired lateral velocity and rotational direction, the signal magnitude alters in order to achieve the desired trajectory as depicted on the top of Fig. 12.5 with the dotted and solid lines denoting the torque on left and right wheels, respectively. The systems total mass is assumed to be 100 kg including the user. At $t = 2$ s is produced some movement on the system in a straight direction and resulted in the rear manual wheels rotating at a maximum velocity of 3.2 r/s as depicted.

12.3 Mechatronic Drive Module

12.3.1 Description and Schematic

To study the different components that contribute to the dynamic behavior of the system, a separation of the MDM from the structure was carried out and basic layout is shown in Fig. 12.6. A typical MDM is depicted in Fig. 12.6. It comprises of a direct current (DC) motor converting electric energy (from batteries in this case) into mechanical energy. Velocity reduction mechanical gears are used and the manner in which motorized wheels are coupled to the system is illustrated.

Table 12.1 System parameters

Parameter	Description	Value
Manual propelled wheelchair (MPW)		
MSe:τ	Rear wheel propulsion torque (Nm)	12
Jw	Rear wheel rotational inertia (kg m^2)	0.005
Mt	Systems mass (kg)	100
Jt	Systems inertial moment (kg m^2)	64
Rg	Wheel to ground resistance	0.006
Cw	Wheel spoke spring	0.0021
Rw	Wheel spoke damper	12
r	Rear wheel radius (m)	0.30226
Lw	Wheelchair width (m)	0.8
Mechatronic drive module (MDM)		
Ie	Electric motor inductance (*H*)	0.0033
Re	Electric motor resistance (Ω)	0.9
Jr	Rotor rotational inertia (kg m^2)	0.078
Rb	Motor bearing damper (Nm-s/rad)	0.008
Mt	Module mass (kg)	30
Jt	Module inertial moment (kg m^2)	7.5
Lw	Module width (m)	0.6
Cs	Motor to gear shaft torsion (Nm/rad)	0.00237
Rs	Motor to gear shaft damper	11
k_1, k_8	Motor torque constant (Nm/A)	0.288
k_2, k_7	Mechanical gear ration	0.18
k_3, k_6	Motorized wheels radius (m)	0.127
MSe:L	Control input voltage (V)	24
Modular driven manual wheelchair (MDMW)		
J	Manual wheels (kg m^2)	0.005
k_9, k_{10}	manual wheels radius (m)	0.30226
Mt	Combined systems total mass (kg)	130
Jt	Total inertial moment (kg m^2)	83.5

The small values of rotational dampers and torsion springs on the shafts have been lumped together and represented by *Rs* and *Cs*, respectively. The different components contributing to the dynamic behavior of the MDM are identified as:

- Electric inductance of the motor windings.
- Rotational inertia of the armature.
- Stiffness of the shaft coupling power motor to the mechanical gears.
- Rotational inertia of the motorized wheels.
- Mass of the MDM system, inertial moment, and the stiffness at the wheels connection to the body.

From this we hence construct the relevant BG model as depicted in Fig. 12.7. Notice that it only represents the components contributing to the dynamic behavior of the MDM system (depicted in Fig. 12.6b) extracted from the EPPW.

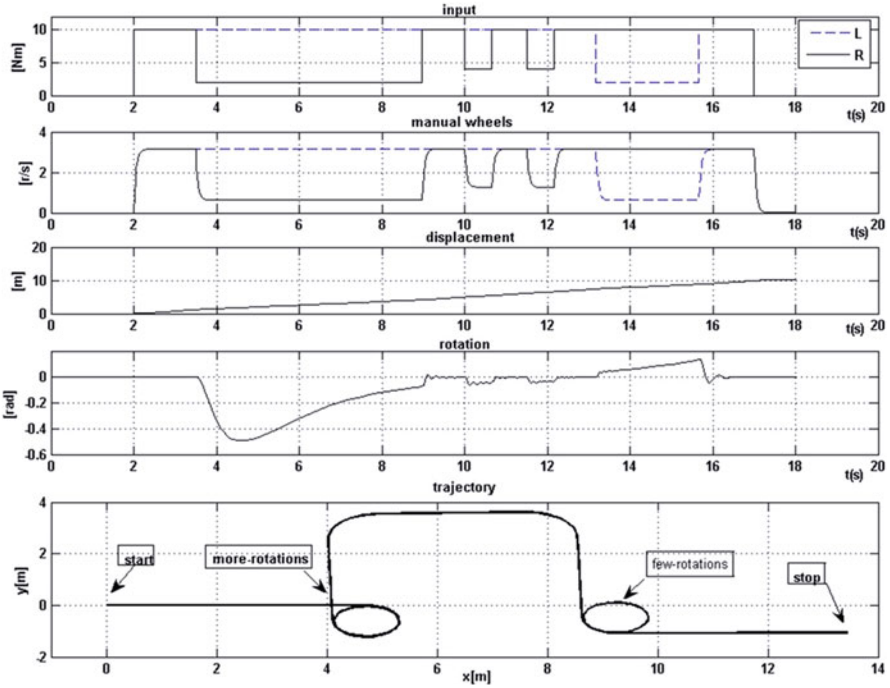


Fig. 12.5 Propelled wheelchair simulation results

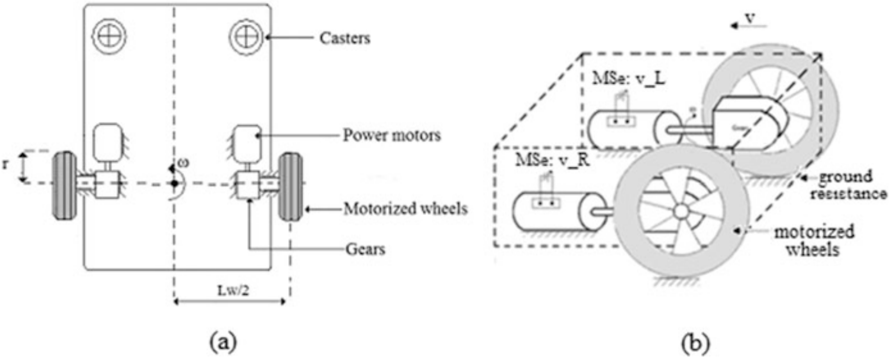


Fig. 12.6 Typical mechatronic drive module

12.3.2 Bond Graph Model of the MDM

Different components forming part of the MDM are represented by the Bond graph in Fig. 12.7. This BG model is decomposed into four main parts, the first two representing the electric motor and the transmission gear, while the second shows the wheels inertia, and the last showing the dynamics of the wheelchair structure built around the translational *Mt* and rotational *Jt* inertia. To study the dynamics of

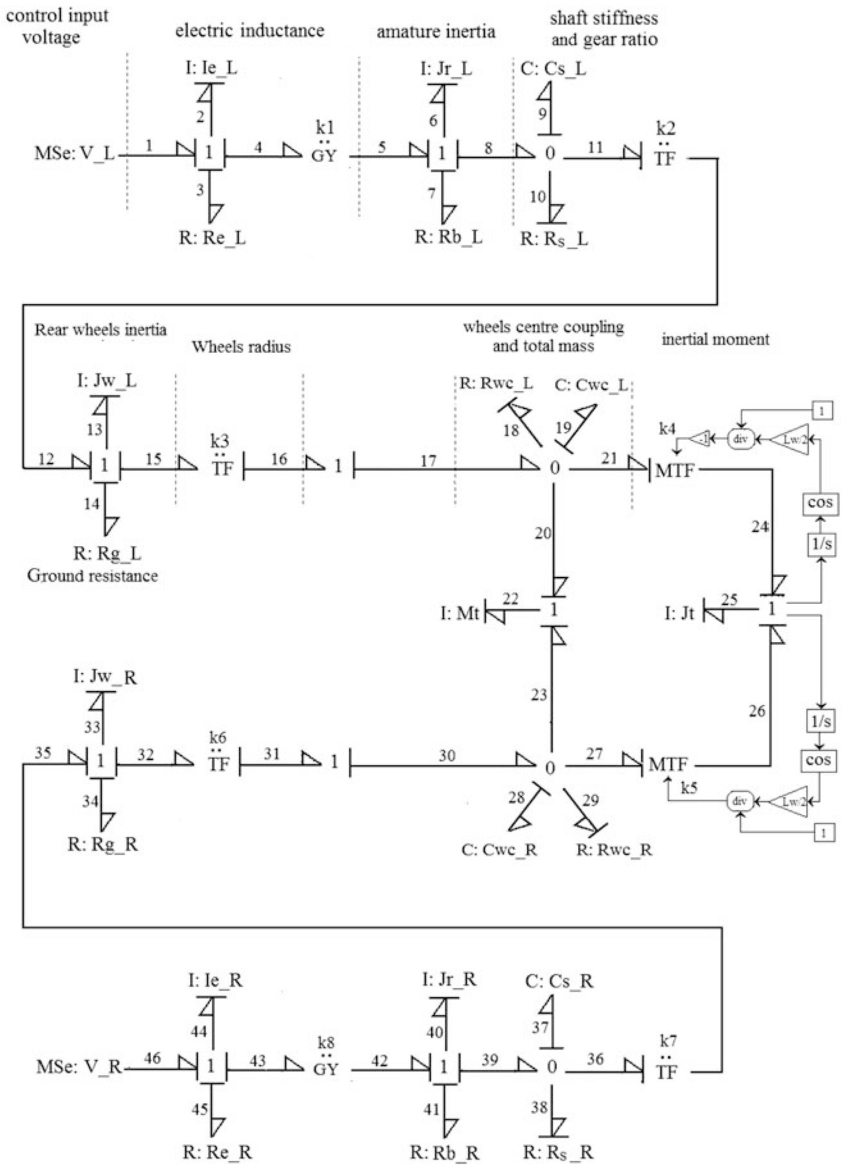


Fig. 12.7 Bond graph representation of MDM

the system, the imperative outputs to be observed are i_2, ω_4, ω_{13}, and v_{23} which denote the motor electric current, armature angular velocity, wheel angular velocity, and the module lateral velocity or center of mass, respectively. The given state space differential equations further denote all the outputs observed in this modelling.

12.3.3 State Space Representation of the MDM

Equations (12.4) and (12.5) represent the state space models of the MDM system. These equations can be used for simulation; however, in complex multi-domain systems, the order increases and the difficulty of manual extraction of state spaces increases in proportion. This clearly denotes the difference between the Bond graph and state space mathematical modelling. In this section, the equations of junctions are not reproduced, they are explicitly represented in the state representation below.

$$\dot{\mathrm{x}} = [\mathrm{A}]\,x + [\mathrm{B}]\,u \tag{12.18}$$

$$\begin{bmatrix} e_2 \\ e_6 \\ f_9 \\ e_{13} \\ f_{19} \\ e_{22} \\ e_{25} \\ f_{28} \\ e_{33} \\ f_{37} \\ e_{40} \\ e_{44} \end{bmatrix} = \begin{bmatrix} \dot{p}_2 \\ \dot{p}_6 \\ \dot{q}_9 \\ \dot{p}_{13} \\ \dot{q}_{19} \\ \dot{p}_{22} \\ \dot{p}_{25} \\ \dot{q}_{28} \\ \dot{p}_{33} \\ \dot{q}_{37} \\ \dot{p}_{40} \\ \dot{p}_{44} \end{bmatrix} = [\mathbf{A}] \begin{bmatrix} p_2 \\ p_6 \\ q_9 \\ p_{13} \\ q_{19} \\ p_{22} \\ p_{25} \\ q_{28} \\ p_{33} \\ q_{37} \\ p_{40} \\ p_{44} \end{bmatrix} + \begin{bmatrix} 1\ 0 \\ 0\ 0 \\ 0\ 0 \\ 0\ 0 \\ 0\ 0 \\ 0\ 0 \\ 0\ 0 \\ 0\ 0 \\ 0\ 0 \\ 0\ 0 \\ 0\ 0 \\ 0\ 1 \end{bmatrix} \begin{bmatrix} VL \\ VR \end{bmatrix}$$

where the dynamics evolution matrix **A** is

$$\mathbf{A} = \begin{bmatrix}
\frac{-Re\_L}{Ie\_L} & \frac{-k1}{Jr\_L} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{k1}{Ie\_L} & \frac{-Rb\_L}{Jr\_L} & \frac{-1}{Cs\_L} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{-1}{Jr\_L} & \frac{-1}{Rg\_LCs\_L} & \frac{-1}{k2Jw\_L} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{k2Cs\_L} & \frac{-Rg\_L}{Jw\_L} & \frac{-k3}{Cwc\_L} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{k3}{Jw\_L} & \frac{-1}{Rwc\_LCwc\_L} & \frac{-1}{Mt} & \frac{k4}{Jt} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{Cwc\_L} & 0 & 0 & 0 & \frac{1}{Cwc\_R} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{k4}{Cwc\_L} & 0 & 0 & \frac{1}{k5Cwc\_R} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{-1}{Mt} & \frac{-1}{k5Jt} & \frac{-1}{Rwc\_RCwc\_R} & \frac{k6}{Jw\_R} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-k26}{Cwc\_R} & \frac{-Rg\_R}{Jw\_R} & \frac{1}{k7Cs\_R} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{k7Jw\_R} & \frac{-1}{Rg\_RCs\_R} & \frac{-1}{Jr\_R} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{Cs\_R} & \frac{-Rb\_R}{Jr\_R} & \frac{k8}{Ie\_R} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-k8}{Jr\_R} & \frac{-Re\_R}{Ie\_R}
\end{bmatrix}$$

The outputs are given by:

$$y = [\mathrm{C}]\,x \tag{12.19}$$

$$\begin{bmatrix} i_L \\ \tau_L \\ \omega_{mL} \\ \omega_{wL} \\ V_{\mathrm{CG}} \\ \theta_{\mathrm{CG}} \\ \omega_{wR} \\ \omega_{mR} \\ \tau_R \\ i_R \end{bmatrix} = \begin{bmatrix} i_2 \\ e_6 \\ f_8 \\ f_{13} \\ f_{22} \\ f_{25} \\ f_{33} \\ f_{39} \\ e_{40} \\ i_{44} \end{bmatrix} = \begin{bmatrix} \frac{1}{Ie\_L} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{k1}{Jr\_L} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{Jr\_L} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{Jw\_L} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{Mt} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{Jt} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{Jw\_R} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{Jr\_R} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{k8}{Jr\_R} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{Ie\_R} \end{bmatrix} \begin{bmatrix} p_2 \\ p_6 \\ p_{13} \\ p_{22} \\ p_{25} \\ p_{33} \\ p_{40} \\ p_{44} \end{bmatrix}$$

12.3.4 Simulations of the MDM Behavior

The system model is simulated and responses observed during a motion in a space of 20 × 10 m^2 predetermined trajectory (Fig. 12.8).

The systems were simulated for a period of 45 s with a profile of step input signals applied on the electric motor terminals ranging from 0 to 24 V, with the maximum value resulting into a lateral velocity of 1.8 m/s, and eventually brought to

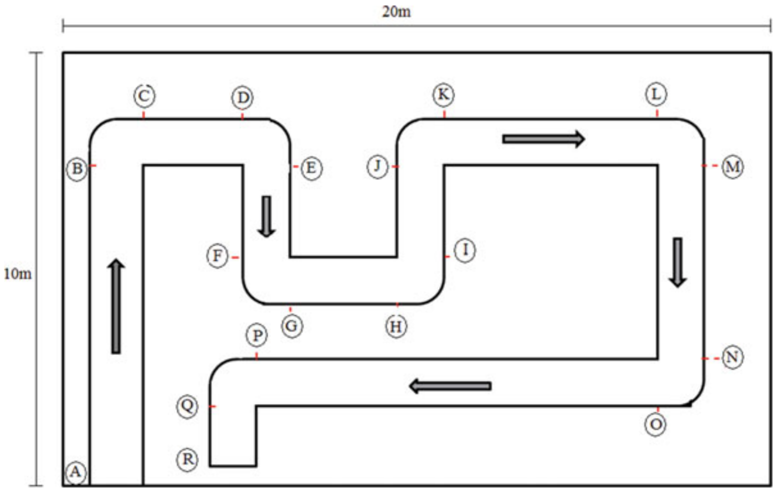


Fig. 12.8 Predetermined trajectory

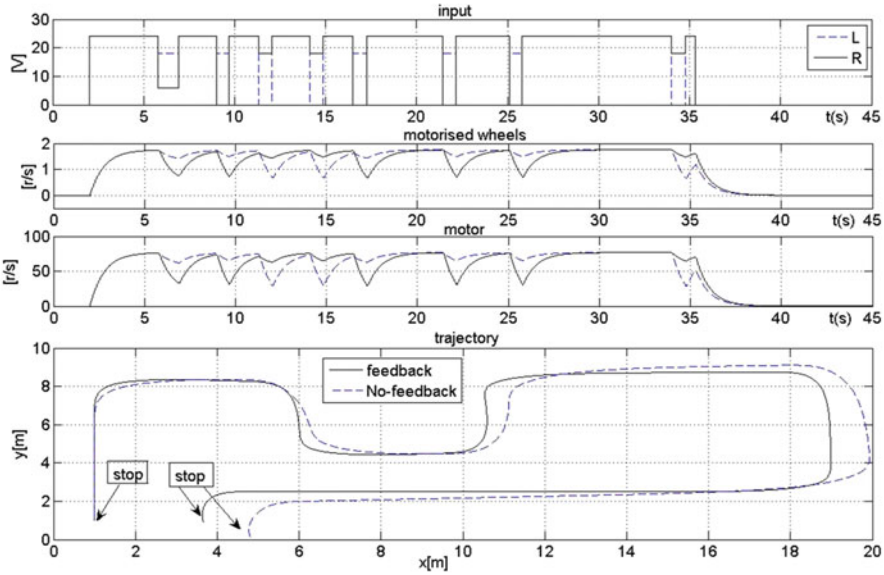


Fig. 12.9 Mechatronic drive module simulation results

stationary position at $t = 36$ s. Illustrated in Fig. 12.9 are some of the system outputs observed during motion in a space of 20×10 m^2. The system was simulated for 45 s with a profile of step input signals applied on the electric motor terminals ranging from 0 to 24 V, with maximum value resulting into a lateral velocity of 1.8 m/s, and eventually brought to stationary position at $t = 36$ s.

In all the graphs, the solid and dashed lines show the right and left side dynamics, respectively. We can visualize that angular speeds of both the motors and wheels follow the input profile. Rotation at the center of mass was observed by an altering "rotation" signal (not shown). The predetermined trajectory is depicted on the lower graph with the system operating in both open- and closed-loop control.

12.4 Modular Driven Manual Wheelchair

12.4.1 Test Bed Description

Ultimately, illustrated in Fig. 12.10 is the top view of an MDMW system model presented in Fig. 12.1. The module introduces two more wheels on the existing four wheels of MPW. The module is assumed to be coupled on the same axes with the rear manual wheels as shown below. *Lw*/2 represents the width of MPW from center of rotation to the center of rear wheel. The BG and some simulations are presented in the following.

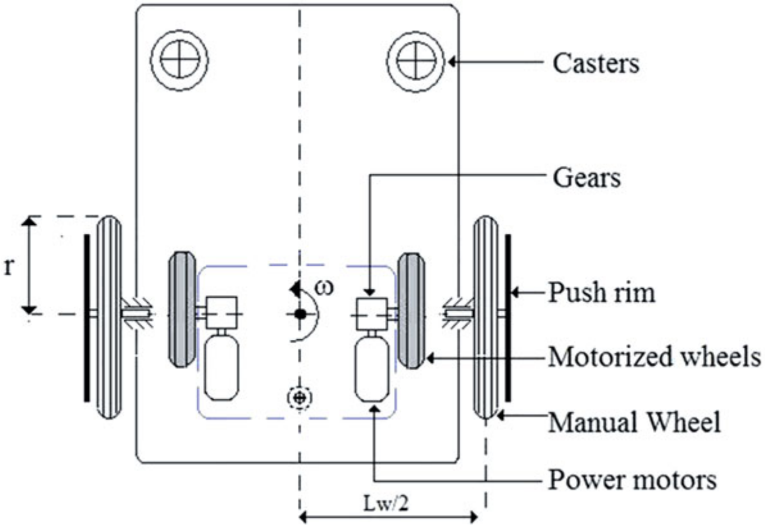


Fig. 12.10 MDMW system

12.4.2 Bond Graph Model

As stated in Fig. 12.1, the motivation of this work derives from attempts to employ few portable MDM which can be coupled when necessary to a large scale of MPW systems with the aims of improving propulsion efficiency. The proposed engagement of such a module (to be modelized and simulated) on MPW is as illustrated in Fig. 12.1. Bond graphs are based on energy exchange, and Fig. 12.11 gives an illustration on how energy is transferred and exchanged from the voltage inputs to the MPW structure. This allows observation of different dynamic information and the system behavior. The kinetic energy generated by the module is transmitted mechanically to the MPW system. In Fig. 12.10, the parameters *Mt and Jt* represent the translational and rotational inertia of the entire systems. With the MDM serving as source of kinetic energy to the wheelchair system, the manual wheels of the MPW are now considered as passive. This introduces the unpreferred *derivative causality* at the manual wheels rotational inertia. The motorized and manual wheels angular velocities are observed. The following section gives the results obtained.

12.4.3 Simulation Results for the Whole System

We use the same predetermined trajectory and profile of step input voltage signals used for the MDM. The whole model is simulated and responses observed during a motion in the same space of 20×10 m^2 presented in Fig. 12.8. The aim is to

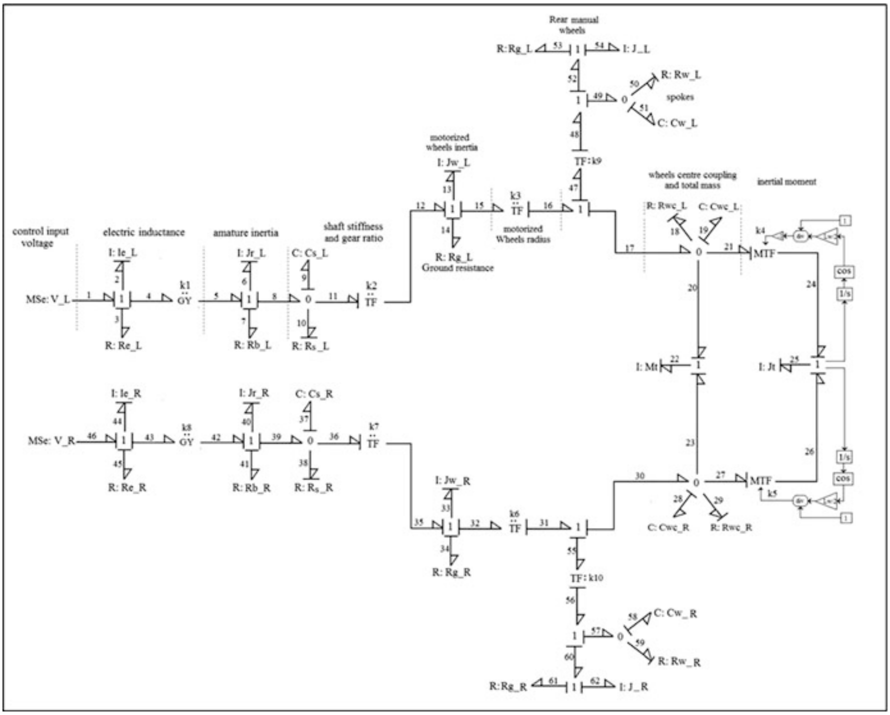


Fig. 12.11 MDM attached to the MPW system

observe and compare whether the MDM propels the MPW successfully. With more weight on the MDMW and operated in open loop control, the system displaced for 15 m in 45 s instead of 32 m. A local feedback from the motor’s angular velocity including the PI controller was employed. Coupling stiffness increased. The system reaching the predetermined trajectory in the exact time; however, more amperage consumption was noticed due to high propulsion torque required to withstand the increased translational inertia. Figure 12.12 depicts the motorized wheels speed.

Figure 12.13 shows the torque produced by the consumed amperage, it can be seen that the system requires a torque of approximately 12 Nm in order to move from the stationary position and accordingly the signal settles down to zero after an appropriate speed is attained. Torque is dependent on input amperage which rises with the sudden reduction on input voltage supply due to the fact that the actuators are induction.

A set of results is presented in Fig. 12.14. To be compared, the variables presented are the same that in Fig. 12.9.

As stated before, a profile of step voltage input signal ranging from 0 to 24 V (shown in Fig. 12.8) is applied on the actuators to move the system from point *A* to *R* with different turning points. The MDMW system moved from the stationary position at point A and due to the combined weight of the two systems (MPW

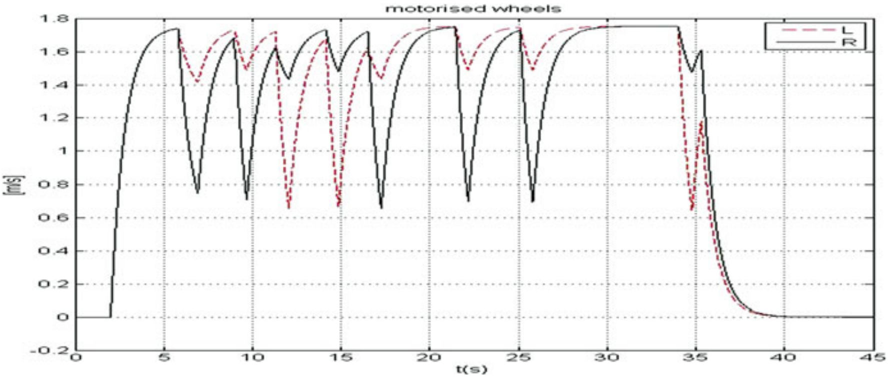


Fig. 12.12 Motorized wheels speed

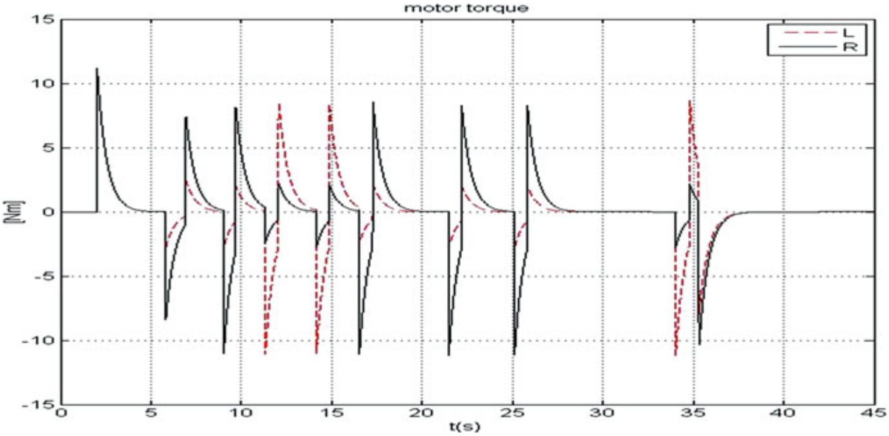


Fig. 12.13 Motor torque

and MDM) coupled together, we observe a high amount of amperage consumption. However the interesting fact is that after the required lateral velocity is attained, the amperage consumption settles to a reasonably low value. The angular velocity of the manual wheels shown in Fig. 12.9 depicts the motorized wheels' speed. From these two figures, it can be seen that with the high amperage consumption on the actuators, the motorized wheels still manage to attain a speed of 1.8 r/s at maximum power supply. The manual wheels rotate at a speed of 0.525 r/s due to the difference in radius size when compared to the motorized wheels. The MDMW system position depicted above, with the results presented in both open- and closed-loop controls using the same local motor speed feedback. From this figure it is observed that the system follows the predetermined trajectory although the open loop control becomes out of phase (takes longer to respond to the direction change signals).

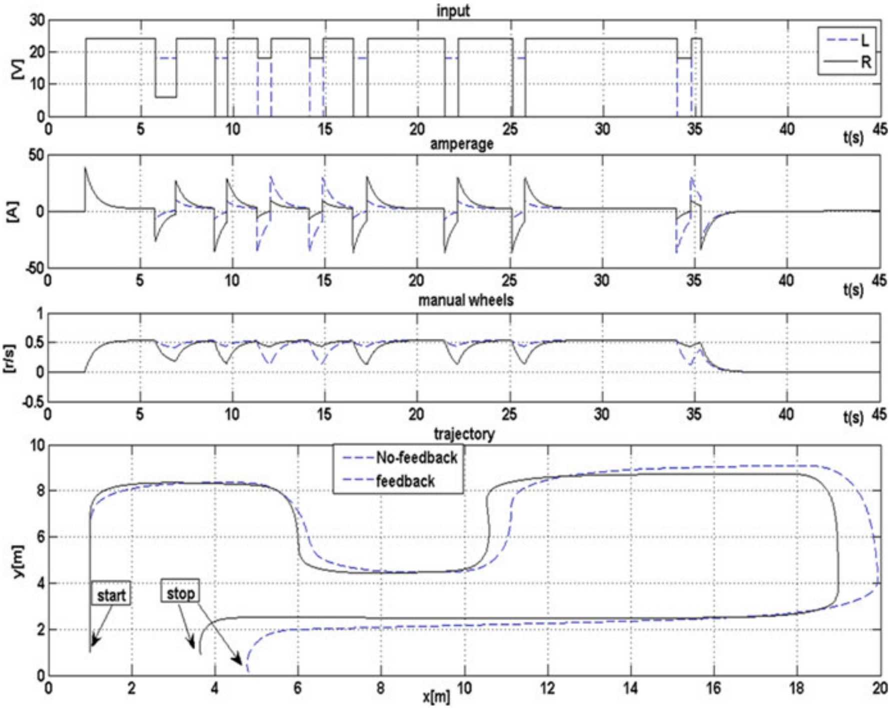


Fig. 12.14 Modular driven manual wheelchair simulation results

12.5 Conclusions

In this chapter we have presented the Bond graph modelling, analysis, and simulation work of a MDM and its coupling to the ordinary manual propelled wheelchair. The Bond graph method was employed to study the different substructures and the linked systems. The models were simulated on a realistic area and can be used as a tutorial for the Bond graph studies of movable systems.

These models will be utilized to design and implement real-time controller that can be implemented on digital control loops. We also plan to build a prototype that will be used to test the reliability functioning and to improve power efficiency management of this system.

Acknowledgments Sincere appreciation goes to Ile de France, ESIEE Paris, and Tshwane University of Technology and FSATI for all the financial assistance and for allowing us to utilize their facilities.

Glossary

BD	Block Diagram
BG	Bond Graph
EPPW	Electric Power Propelled Wheelchair
C	1-port Capacitive Element
I	1-port Inductive Element
GY	Gyrator
MSe	Modulated Source of Effort
MDM	Mechatronic Drive Modular
MDMW	Modular Driven Manual Wheelchair
MPW	Manual Propelled wheelchair
MSf	Modulated Source of Flow
MSe	Modulated Source of Effort
MTF	Modulated Transformer
$\boldsymbol{P}_{\mathbf{CG}}$	Linear Momentum
$\boldsymbol{P}_{\theta}$	Angular Momentum
R	1-port Resistive Element
TF	Transformer
$\boldsymbol{V}_{\mathbf{CG}}$	Center of mass velocity
$\omega_{\mathbf{CG}}$	Inertial mass velocity

References

1. Abel, E. N., & Frank, T.G. (1991). The design of an attendant propelled wheelchair. Prosthetic and orthotic international, Pubmed (pp. 38–45), Apr 1991.
2. Borutzky, W. (2010). *Bond graph methodology, development and analysis of multidisciplinary dynamic system models* (1st ed.). Berlin: Springer.
3. Dauphin-Tanguy, G. (2000). *Les bond graphs*. Paris: Hermes Edition.
4. Fakri, A., Rocaries, F., & Carriere, A. (1997). A simple method for the conversion of bond graph models in presentation by block diagrams. *International Conference on Bond graph Modelling and Simulation, ICBGM'97* (pp. 15–19), Phoenix Arizona.
5. Fakri, A., & Vilakazi, J. P. (2010). Modular driven wheelchair bond graph modelling. *The European Modeling & Simulation Symposium, I3M2010 MultiConference,* October 2010 Fes, Morocco.
6. Fakri, A., & Vilakazi, J. P. (2014). Wheelchair and electric drive add-on a whole bond graph modelling. *Proceedings of the 2014 11th International Conference on Bond graph Modeling and Simulation (ICBGM'14). SummerSim 2014 Multiconference July 6–10 2014*, Monterey, CA, USA. ISBN: 978-1-63266-700-7, Vol. 46 #8, Collection:Simulation Series, 206-210, 222 pp. 113.
7. Vilakazi, J. P. (2011). A study of a mechatronic drive module to be coupled on to an ordinary manual propelled wheelchair. *French South African Technical Institute in Electronics (F'SATIE)*, Tshwane University of Technologie.

8. Karnopp, D. C., Margolis, D. L., & Rosenberg, R. C. (2006). *System dynamics: A unified approach* (4th ed.). New York: Wiley.
9. Wheelchairnet.org (2010). *Discussion preparation for manual wheelchair propulsion*. Mid Atlantic Region: Rehabilitation Engineering Research Centre on Technology Transfer Federal Laboratory Consortium (April 15).

Chapter 13
Bond Graph Modelling and Control of Hyper-Redundant Miniature Robot for In-Vivo Biopsy

Mihir Kumar Sutar and Pushpraj Mani Pathak

DOF	Degrees of freedom
PI	Proportion-integral controller
CG	Centre of gravity
PD	Proportional derivative controller
EJS	Euler junction structure

13.1 Introduction

Endoscopy involves a procedure that uses a specialized tool to view, take biopsy sample, and operate on the abdominal cavity of human body when required. The procedure allows a surgeon to examine the abdominal cavity to ascertain the cause of discomfort reported by the patient, and for remedial action. It is performed with an endoscope, which is a flexible tube with all attached accessories required for biopsy and viewing. Gastroscopy is a type of endoscopy. It is a symptomatic endoscopic procedure that examines the gastrointestinal tract. It is considered a minimally invasive procedure as it does not require any incision into the body and does not require any significant recovery time after the procedure. Conventional endoscopy complicates the task of the surgeon due to factors like lack of dexterity, poor viewing

M.K. Sutar
Department of Mechanical Engineering, Veer Surendra Sai University of Technology, Burla, Odisha, 768018, India
e-mail: mihirsutar05@gmail.com

P.M. Pathak (✉)
Department of Mechanical and Industrial Engineering, Indian Institute of Technology Roorkee, Roorkee, 247667, India
e-mail: pushpfme@iitr.ac.in; pushppathak@gmail.com

W. Borutzky (ed.), *Bond Graphs for Modelling, Control and Fault Diagnosis of Engineering Systems*, DOI 10.1007/978-3-319-47434-2_13

angles, problems in manoeuvring the endoscope inside the abdominal cavity for exploration, and controlling the tip of the flexible endoscope tube to reach at the exact point of interest.

The adoption of in-vivo robots in conjunction with laparoscopic procedure involves the insertion of a robot through strategically located incision to place it properly at the operating location in the abdominal cavity [1]. Laparoscopic in-vivo procedure significantly reduces patient trauma from a long hospital stay and expensive medical bills as compared to the conventional open surgery. Additionally, it also eliminates the complications associated with surgeon's tremor during the traditional procedure. However, these robots pose problems of insertion, retraction, and controlling of the robot.

In order to address the above problems there is a growing awareness to combine the robot with the endoscope in a procedure that may be called robotized endoscopy. The main advantage of robotized endoscopy is that it is incision-free and hence bereft of the difficulties associated with insertion and withdrawal of robot encountered in laparoscopic procedures. However the challenge lies in developing a miniature robot that is small enough to be carried by one of the endoscope channels and yet has enough manoeuvrability for executing the desired surgical procedures. Research in robotized endoscopy is still in its nascent stage and researchers round the globe are working on issues of concern such as compromised dexterity, limited visibility for the surgeon when performing the task, fewer degrees of freedom (DOF), and limited manoeuvrability.

The size of the robot should be such that available tool channel in the endoscope can be used to place the robotized manipulator in it. The endoscope is stationed at the entrance to the abdominal cavity, the robot comes out and performs its desired task of taking biopsy at a particular location by means of an effective inverse kinematic solution and then comes back to its home position in the endoscope.

Several robotic systems have been developed over the years to increase the surgeon's dexterity by masterly manipulating laparoscopic tools [2] where surgical dexterity refers to the ability to steer devices such as needles, rigid laparoscopic tools, and robotic tools [3]. To counter the problem of limitation on dexterity and to obtain high flexibility and manoeuvrability hyper-redundant robots are preferred over conventional ones. These types of robots are termed so, because their number of actuatable DOF is much higher than the DOF of their intended workspace [4]. Robots akin to snake, tentacles, or elephant trunks are examples of hyper-redundant robots [3, 5].

The efficiency of robotized surgical manipulator also depends on how it moves inside the abdominal environment in a controlled manner. Conventional endoscopes rely on the expertise of the surgeon to move the forceps tip to the desired location by rotating the knob in the endoscope. It is a tedious process and the surgeon faces considerable difficulty in manoeuvring the endoscope during forceps tip positioning at the location of interest. For the motion of tip of the surgical robot from one desired position to another, a proper inverse kinematic analysis is essential. Therefore there is a progressive need to develop a simple but effective inverse kinematic solution to place the surgical robot tip at the desired point of interest with minimum error.

Controlling of the hyper-redundant manipulator is a challenging task and is one of the major concerns in robotized endoscopy. During biopsy it is required to move the manipulator in the abdominal cavity environment and to place its tip at the desired point of interest, without damaging the healthy tissues. For this a proper control scheme for trajectory and force control are required and various strategies including hybrid position and force control strategies have been addressed by researchers in this regard [6]. In robotized endoscopy, the forceps tool can be moved to follow a certain path, by employing a trajectory controller which should be competent enough to ensure that the tip of the robotic manipulator follows the desired trajectory inside abdominal cavity.

13.2 Forward Kinematic Analysis of the In-Vivo Robot

Redundant robots have high dexterity and manoeuvrability as compared to conventional one. However, motion planning and controlling of redundant manipulators have always been a challenging task for researchers. To carry out analysis of a robot, the first important step is the creation of its dynamic model. Dynamic models are important to have feed-forward information for controlling the manipulators actions, by controlling the related actuators through interactive controlling software. This helps in

- Quick and correct motion of the manipulator along a predefined or designed path.
- Effective compensation system for inertia, centrifugal, gravity, and Coriolis components during motion.

Dynamic modelling of robot is essential for its kinematic analysis to obtain the transformer moduli required to develop the bond graph model. Mostly three methodologies have been adopted for this purpose namely,

- Lagrange–Euler (L-E) dynamic formulation
- Iterative Newton–Euler (N-E) dynamic formulation
- Bond Graph approach.

The bond graph modelling is simpler, and the effort equations derived for the robot are in single scalar closed form for each joint and can be used for online computations. These equations are always faster and more efficient than the iterative and recursive form of equations. Above all the modelling is easier, modular in nature, and gives physical insight and intuitive feeling of the dynamics. Also bond graph modelling is a unified approach, well suited for modelling of multi-domain discipline (due to the presence of mechanical, electrical, thermal, mechatronics, etc.) systems. It allows precise and independent nature of all the domains, with exchange of power in the system in the form of storage, dissipation, and transformation. The modelling of system is flexible as it allows graphical addition of friction, stiffness-effect elements as and when required [7].

The first step in the kinematic analysis is to assign frames to the in-vivo robot model. The frame assignment to the in-vivo robot is shown in Fig. 13.1.

Fig. 13.1 Frame assignments to the in-vivo manipulator

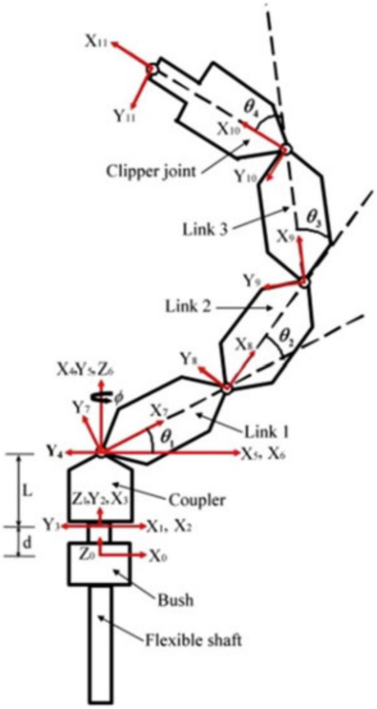


Table 13.1 D-H parameters of the in-vivo robotic manipulator

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	d	0
2	90°	0	0	0
3	0	0	0	90°
4	0	L	0	0
5	0	0	0	90°
6	−90°	0	0	ϕ
7	90°	0	0	θ_1
8	0	L_1	0	θ_2
9	0	L_2	0	θ_3
10	0	L_3	0	θ_4
11	0	L_4	0	0

A coordinate frame is attached to each link. The bush is considered as link 0 and inertial frame {0} is attached to the bush. The subsequent frames are attached to the base of the coupler, the links, and to the clipper joint, respectively. The joint between link i and $i+1$ terms as joint $i+1$, e.g., the frame i is attached to link i. Standard D-H conventions have been used [8]**,** for describing the frames. The notations are given in Table 13.1. The Denavit–Hartenberg notations basically depict the link and joint parameters. The overall homogeneous transformation matrix based on its general form [9] can be expressed as:

$$ {}_{11}{}^{0}T = \begin{bmatrix} c\phi c_{1234} & -c\phi s_{1234} & s\phi & L_1 c\phi c_1 + L_2 c\phi c_{12} + L_3 c\phi c_{123} + L_4 c\phi c_{1234} \\ s\phi c_{1234} & -s\phi s_{1234} & -c\phi & L_1 s\phi c_1 + L_2 s\phi c_{12} + L_3 s\phi c_{123} + L_4 s\phi c_{1234} \\ s_{1234} & c_{1234} & 0 & (L+d) + L_1 s_1 + L_2 s_{12} + L_3 s_{123} + L_4 s_{1234} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13.1} $$

From Eq. (13.1) the manipulator tip positions can be expressed as follows:

$$ X_{\text{tip}} = c\phi \left(L_1 c_1 + L_2 c_{12} + L_3 c_{123} + L_4 c_{1234} \right) \tag{13.2} $$

$$ Y_{\text{tip}} = s\phi \left(L_1 c_1 + L_2 c_{12} + L_3 c_{123} + L_4 c_{1234} \right) \tag{13.3} $$

$$ Z_{\text{tip}} = (L + d) + \left(L_1 s_1 + L_2 s_{12} + L_3 s_{123} + L_4 s_{1234} \right) \tag{13.4} $$

The manipulator is a connected chain of links, capable of motion relative to its neighbour link. The velocity of the link $i + 1$ will be equal to the velocity of link i and new velocity component added by joint $i + 1$. Since translational velocity is associated with a point, and angular velocity is associated with a body, the term “velocity of a link” signifies the translational velocity of the origin frame attached to that link, in addition to the rotational velocity of that link. It is evident that the rotational velocities may be added when both angular velocity vectors are written with respect to the same frame.

The angular velocity of the link $i + 1$ is the addition of the angular velocity of link i and a component caused by rotational velocity of joint $i + 1$, with respect to the inertial frame {0}. So, the angular velocity relation of link $i + 1$ with respect to inertial frame {0} and expressed in frame $i + 1$ coordinate, is given by,

$$ {}^{i+1}\left({}^{0}\omega_{i+1}\right) = {}_{i}^{i+1}R^{i}\left({}^{0}\omega_{i}\right) + \dot{\theta}_{i+1}\,{}^{i+1}\widehat{Z}_{i+1} \quad \text{(for rotational joint)} \tag{13.5} $$

$$ {}^{i+1}\left({}^{0}\omega_{i+1}\right) = {}_{i}^{i+1}R^{i}\left({}^{0}\omega_{i}\right) \quad \text{(for translational or prismatic joint)} \tag{13.6} $$

In the present analysis the bush of in-vivo robot is considered as the inertial frame {0} so, using Eqs. (13.5) and (13.6) angular velocity of the links can be derived by substituting for different values of i. Therefore substituting the values of i the angular velocities obtained are as follows:

$$ {}^{1}\left({}^{0}\omega_{1}\right) = {}^{2}\left({}^{0}\omega_{2}\right) = {}^{3}\left({}^{0}\omega_{3}\right) = {}^{4}\left({}^{0}\omega_{4}\right) = {}^{5}\left({}^{0}\omega_{5}\right) = \left[0\ 0\ 0\right]^{T} \tag{13.7} $$

$$ {}^{6}\left({}^{0}\omega_{6}\right) = \left[0\ 0\ \dot{\phi}\right]^{T} \tag{13.8} $$

$$ {}^{7}\left({}^{0}\omega_{7}\right) = \left[s_1\dot{\phi}\ c_1\dot{\phi}\ \dot{\theta}_1\right]^{T} \tag{13.9} $$

$$^{8}\left({}^{0}\omega_{8}\right) = \left[s_{12}\dot{\phi}\ c_{12}\dot{\phi}\ \dot{\theta}_{1} + \dot{\theta}_{2}\right]^{T} \tag{13.10}$$

$$^{9}\left({}^{0}\omega_{9}\right) = \left[s_{123}\dot{\phi}\ c_{123}\dot{\phi}\ \dot{\theta}_{1} + \dot{\theta}_{2} + \dot{\theta}_{3}\right]^{T} \tag{13.11}$$

$$^{10}\left({}^{0}\omega_{10}\right) = \left[s_{1234}\dot{\phi}\ c_{1234}\dot{\phi}\ \dot{\theta}_{1} + \dot{\theta}_{2} + \dot{\theta}_{3} + \dot{\theta}_{4}\right]^{T} \tag{13.12}$$

Similarly the linear velocity of the origin of the frame $\{i+1\}$ will be equal to the sum of the origin of frame $\{i\}$ and a new component caused by rotational velocity of the link i and is expressed as:

$$^{i+1}\left({}^{0}V_{i+1}\right) = {}_{i}^{i+1}R\left[{}^{i}\left({}^{0}V_{i}\right) + {}^{i}\left({}^{0}\omega_{i}\right) \times {}^{i}\left({}^{i}P_{i+1}\right)\right] \quad \text{(for rotational joint)} \tag{13.13}$$

$$^{i+1}\left({}^{0}V_{i+1}\right) = {}_{i}^{i+1}R\left[{}^{i}\left({}^{0}V_{i}\right) + {}^{i}\left({}^{0}\omega_{i}\right) \times {}^{i}\left({}^{i}P_{i+1}\right)\right] + \dot{d}_{i+1} \times {}^{i+1}\widehat{Z}_{i+1} \quad \text{(for prismatic joint)} \tag{13.14}$$

The position ${}^{i}\left({}^{i}P_{i+1}\right)$ represents the position of the origin of the frame $\{i+1\}$ with respect to the frame $\{i\}$ expressed in the ith frame, and the velocity ${}^{i}({}^{0}V_{i})$ represents the velocity of the origin of the frame $\{i\}$ with respect to the inertial frame $\{0\}$ and expressed in frame i coordinate, ${}_{i}^{i+1}R$ represents the orientation of frame $\{i\}$ with respect to the frame $\{i+1\}$. Translational velocities of each frame can be obtained by substituting different values of i in Eq. (13.13) or Eq. (13.14), depending upon whether the joint is rotary or prismatic, respectively. The translational velocities are obtained by substituting different values of i and they are expressed with respect to the inertial frame and are expressed as follows:

$$^{0}\left({}^{0}V_{1}\right) = {}^{0}\left({}^{0}V_{2}\right) = {}^{0}\left({}^{0}V_{3}\right) = {}^{0}\left({}^{0}V_{4}\right) = {}^{0}\left({}^{0}V_{5}\right) = {}^{0}\left({}^{0}V_{6}\right) = {}^{0}\left({}^{0}V_{7}\right) = \left[0\ 0\ \dot{d}\right]^{T} \tag{13.15}$$

$$^{0}\left({}^{0}V_{8}\right) = \begin{bmatrix} -L_{1}c\phi s_{1}\dot{\theta}_{1} - L_{1}s\phi c_{1}\dot{\phi} \\ -L_{1}s\phi s_{1}\dot{\theta}_{1} + L_{1}c\phi c_{1}\dot{\phi} \\ \dot{d} + L_{1}c_{1}\dot{\theta}_{1} \end{bmatrix} \tag{13.16}$$

$$^{0}\left({}^{0}V_{9}\right) = \begin{bmatrix} -L_{1}c\phi s_{1}\dot{\theta}_{1} - L_{2}c\phi s_{12}\left(\dot{\theta}_{1} + \dot{\theta}_{2}\right) - s\phi\left(L_{1}c_{1} + L_{2}c_{12}\right)\dot{\phi} \\ -L_{1}s\phi s_{1}\dot{\theta}_{1} - L_{2}s\phi s_{12}\left(\dot{\theta}_{1} + \dot{\theta}_{2}\right) + c\phi\left(L_{1}c_{1} + L_{2}c_{12}\right)\dot{\phi} \\ \dot{d} + L_{1}c_{1}\dot{\theta}_{1} + L_{2}c_{12}\left(\dot{\theta}_{1} + \dot{\theta}_{2}\right) \end{bmatrix} \tag{13.17}$$

$$^{0}\left({}^{0}V_{10}\right)=\begin{bmatrix} -L_1c\phi s_1\dot{\theta}_1-L_2c\phi s_{12}\left(\dot{\theta}_1+\dot{\theta}_2\right)-L_3c\phi s_{123}\left(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3\right) \\ -\,s\phi\left(L_1c_1+L_2c_{12}+L_3c_{123}\right)\dot{\phi} \\ -L_1s\phi s_1\dot{\theta}_1-L_2s\phi s_{12}\left(\dot{\theta}_1+\dot{\theta}_2\right)-L_3s\phi s_{123}\left(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3\right) \\ +\,c\phi\left(L_1c_1+L_2c_{12}+L_3c_{123}\right)\dot{\phi} \\ \dot{d}+L_1c_1\dot{\theta}_1+L_2c_{12}\left(\dot{\theta}_1+\dot{\theta}_2\right)+L_3c_{123}\left(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3\right) \end{bmatrix} \tag{13.18}$$

$$^{0}\left({}^{0}V_{11}\right)=\begin{bmatrix} -L_1c\phi s_1\dot{\theta}_1-L_2c\phi s_{12}\left(\dot{\theta}_1+\dot{\theta}_2\right)-L_3c\phi s_{123}\left(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3\right) \\ -\,L_4c\phi s_{1234}\left(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3+\dot{\theta}_4\right) \\ -s\phi\left(L_1c_1+L_2c_{12}+L_3c_{123}+L_4c_{1234}\right)\dot{\phi} \\ -L_1s\phi s_1\dot{\theta}_1-L_2s\phi s_{12}\left(\dot{\theta}_1+\dot{\theta}_2\right)-L_3s\phi s_{123}\left(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3\right) \\ -L_4s\phi s_{1234}\left(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3+\dot{\theta}_4\right) \\ +c\phi\left(L_1c_1+L_2c_{12}+L_3c_{123}+L_4c_{1234}\right)\dot{\phi} \\ \dot{d}+L_1c_1\dot{\theta}_1+L_2c_{12}\left(\dot{\theta}_1+\dot{\theta}_2\right)+L_3c_{123}\left(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3\right) \\ +L_4c_{1234}\left(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3+\dot{\theta}_4\right) \end{bmatrix} \tag{13.19}$$

For preparing the dynamic bond graph model of the in-vivo robot the velocities of the centre of mass of each link are also required therefore these velocities can be expressed as:

$$^{0}\left({}^{0}V_{G_7}\right)=\begin{bmatrix} -L_{G_1}c\phi s_1\dot{\theta}_1-L_{G_1}s\phi c_1\dot{\phi} \\ -\,L_{G_1}s\phi s_1\dot{\theta}_1+L_{G_1}c\phi c_1\dot{\phi} \\ \dot{d}+L_{G_1}c_1\dot{\theta}_1 \end{bmatrix} \tag{13.20}$$

$$^{0}\left({}^{0}V_{G_8}\right)=\begin{bmatrix} -L_1c\phi s_1\dot{\theta}_1-L_{G_2}c\phi s_{12}\left(\dot{\theta}_1+\dot{\theta}_2\right)-s\phi\left(L_1c_1+L_{G_2}c_{12}\right)\dot{\phi} \\ -\,L_1s\phi s_1\dot{\theta}_1-L_{G_2}s\phi s_{12}\left(\dot{\theta}_1+\dot{\theta}_2\right)+c\phi\left(L_1c_1+L_{G_2}c_{12}\right)\dot{\phi} \\ \dot{d}+L_1c_1\dot{\theta}_1+L_{G_2}c_{12}\left(\dot{\theta}_1+\dot{\theta}_2\right) \end{bmatrix} \tag{13.21}$$

$$^{0}\left({}^{0}V_{G_9}\right)=\begin{bmatrix} -L_1c\phi s_1\dot{\theta}_1-L_2c\phi s_{12}\left(\dot{\theta}_1+\dot{\theta}_2\right)-L_{G_3}c\phi s_{123}\left(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3\right) \\ -\,s\phi\left(L_1c_1+L_2c_{12}+L_{G_3}c_{123}\right)\dot{\phi} \\ \\ -L_1s\phi s_1\dot{\theta}_1-L_2s\phi s_{12}\left(\dot{\theta}_1+\dot{\theta}_2\right)-L_{G_3}s\phi s_{123}\left(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3\right) \\ +\,c\phi\left(L_1c_1+L_2c_{12}+L_{G_3}c_{123}\right)\dot{\phi} \\ \\ \dot{d}+L_1c_1\dot{\theta}_1+L_2c_{12}\left(\dot{\theta}_1+\dot{\theta}_2\right)+L_{G_3}c_{123}\left(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3\right) \end{bmatrix} \tag{13.22}$$

These equations are obtained by substituting $L_1 = L_{G_1}$ in Eq. (13.16), $L_2 = L_{G_2}$ in Eq. (13.17), $L_3 = L_{G_3}$ in Eq. (13.18), and $L_4 = L_{G_4}$ in Eq. (13.19), respectively.

The transformer moduli obtained from the kinematic analysis has been used for the bond graph modelling of the in-vivo robot.

$$
{}^{0}\left({}^{0}V_{G_{10}}\right) = \begin{bmatrix}
-L_1 c\phi s_1 \dot{\theta}_1 - L_2 c\phi s_{12}\left(\dot{\theta}_1 + \dot{\theta}_2\right) - L_3 c\phi s_{123}\left(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3\right) \\
- L_{G_4} c\phi s_{1234}\left(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4\right) \\
- s\phi\left(L_1 c_1 + L_2 c_{12} + L_3 c_{123} + L_{G_4} c_{1234}\right)\dot{\phi} \\
\\
-L_1 s\phi s_1 \dot{\theta}_1 - L_2 s\phi s_{12}\left(\dot{\theta}_1 + \dot{\theta}_2\right) - L_3 s\phi s_{123}\left(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3\right) \\
- L_{G_4} s\phi s_{1234}\left(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4\right) \\
+ c\phi\left(L_1 c_1 + L_2 c_{12} + L_3 c_{123} + L_{G_4} c_{1234}\right)\dot{\phi} \\
\\
\dot{d} + L_1 c_1 \dot{\theta}_1 + L_2 c_{12}\left(\dot{\theta}_1 + \dot{\theta}_2\right) + L_3 c_{123}\left(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3\right) \\
+ L_{G_4} c_{1234}\left(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4\right)
\end{bmatrix} \tag{13.23}
$$

13.3 Bond Graph Modelling of the In-Vivo Robot

Modulated transformer moduli in terms of the translational and angular velocities required to create the bond graph model are obtained from the kinematic analysis of the robot from the preceding section. Based on these transformer moduli the dynamic model of the in-vivo robot was prepared in bond graph [7, 10, 11, 12] which represents the power exchange portrait of the system. Complete bond graph of the in-vivo robot is shown in Fig. 13.2. In the present analysis, bond graph modelling and its simulation are performed using bond graph modelling software SYMBOLS Shakti®, a bond graph modelling software.

The rotational dynamics of different links of the in-vivo robot are represented by Euler Junction Structure EJS, shown on the right side of the model. Translational dynamics of the link are shown on the left side of the model. The centre of gravity CG velocity of links depends on the link inertia. Therefore in the bond graph model *I* element is attached at velocity junctions representing the CG velocity of links. Tip velocities are indicated on the top of the model. Integrators are used to integrate the velocity components to obtain the tip positions, i.e., $X_{\text{tip}}, Y_{\text{tip}}$ and Z_{tip}. The proficiency of the wire actuation is tested in the present modelling by actuating the joints independently and then by actuating the joints simultaneously. In case of the independent joint actuation individual efforts are provided to each joint as shown in Fig. 13.2. The efforts SE are assumed to be equivalent to the wire actuation effort.

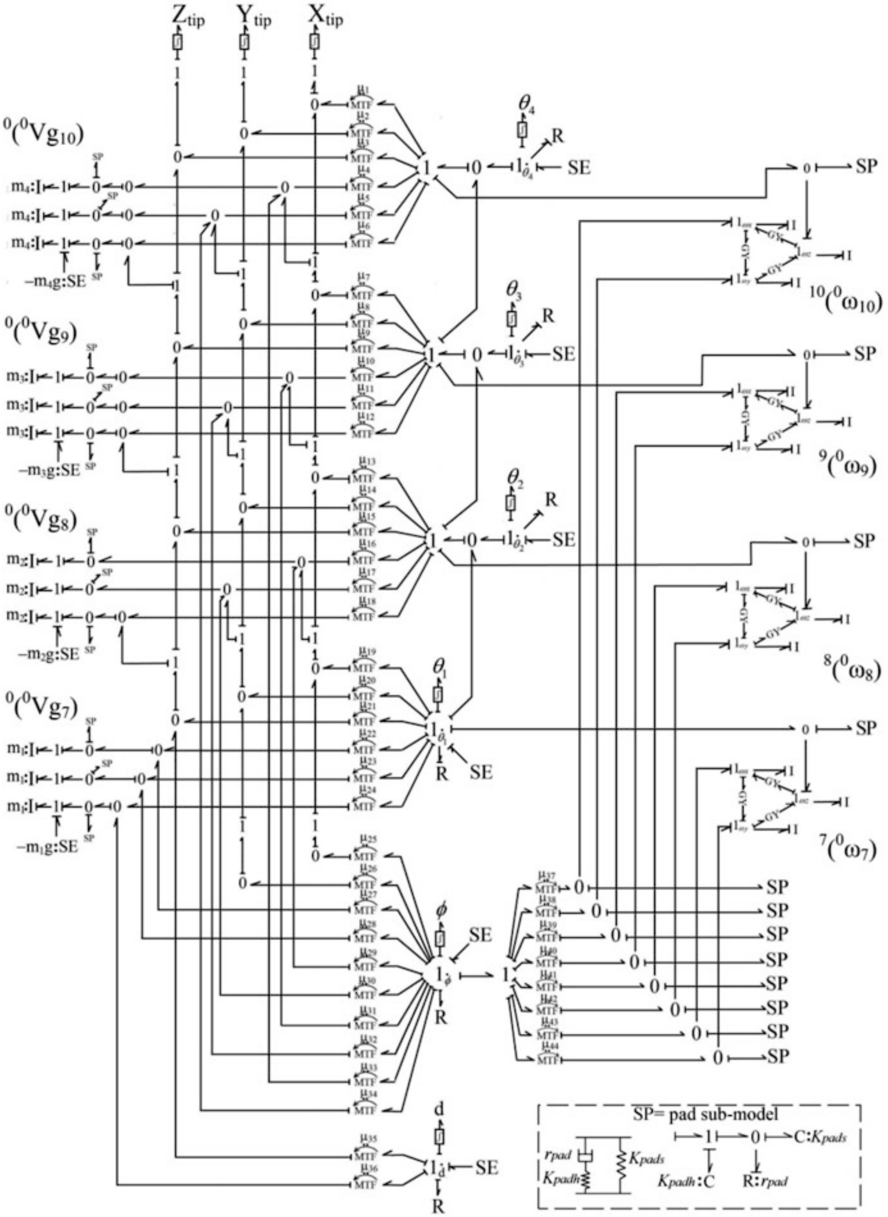


Fig. 13.2 Bond graph model of the in-vivo robot with individual joint actuation

For rotation of joint a very negligible joint resistance of 0.001 Nm/ (rad/s) is given and when it is required to lock the joint the joint resistance is set at 10,000 Nm/ (rad/s).

13.3.1 Independent Joint Actuation Through Wire Actuation

The initial validity of the bond graph model was tested without wire actuation with a four-scaled model. For the simulation various parameters used are as enlisted in Table 13.2. The initial configuration of the robot is shown in Fig. 13.3. Initially the validity of the bond graph was checked without wire actuation. A torque of 2 Nm was applied on 1st joint and it was rotated while all other joints were locked at zero degree initial value. As it can be seen a circle of 0.176 m radius (sum of four planar link lengths) was obtained in *X*-*Y* plane, as shown in Fig. 13.4. A torque of 2 Nm was applied on 1st joint and it was rotated while all other joints were locked at zero degree initial value.

Tip trajectory obtained is a circle of 0.176 m radius in *X*-*Z* plane as plotted in Fig. 13.5. Physically this configuration is possible when the manipulator is out of the tool channel and is inside the stomach model.

13.3.2 Simultaneous Joint Actuation Through Wire Actuation

For simultaneous joint wire actuation efforts are provided to each joint through the wire actuation as shown in Fig. 13.6. Wire actuation system is integrated in the bond graph model and shown on the top right-hand side of the figure. Wire flexibility in

Table 13.2 Simulation parameters for in-vivo robot without wire actuation

Parameter	Value
Link length	0.044 m
Link mass	0.001 kg
Torque applied	2 Nm
Joint resistance	R_1=0.001 Nm/ (rad/s)

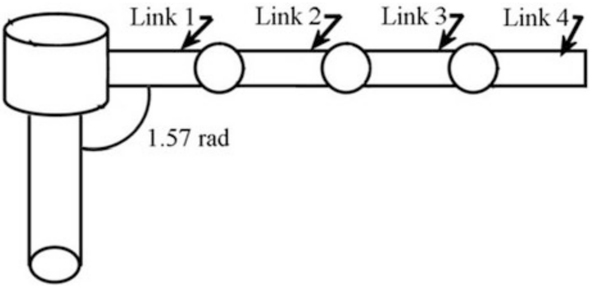


Fig. 13.3 Initial configuration of the in-vivo robot

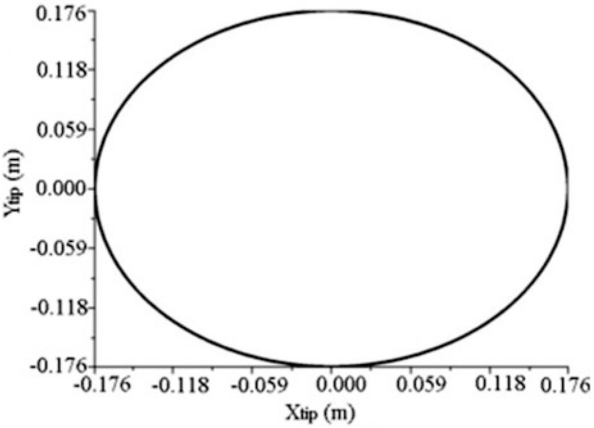


Fig. 13.4 Tip trajectory of in-vivo robot by rotating first joint with all other joints locked at zero degree initial value

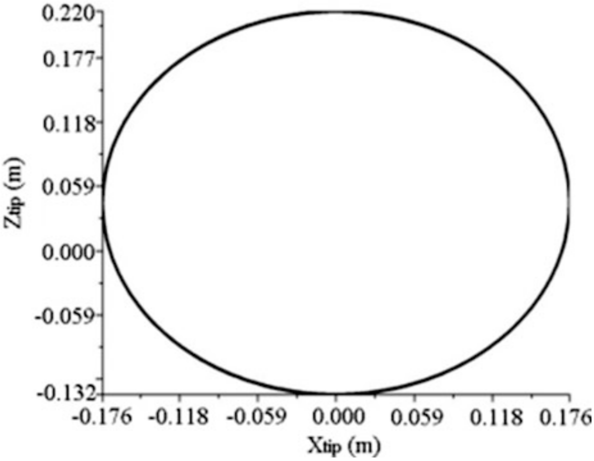


Fig. 13.5 Tip trajectory by rotating second joint with all other joints locked at zero degree initial value

the wire actuation system is represented by K_{wire} and R_{wire}. Modulated transformer moduli obtained from the kinematic analysis have been used to construct the bond graph model and are shown in Fig. 13.6. Translational dynamics is indicated on the left-hand side of the figure and rotational dynamics is indicated on the right-hand side of the figure. The modulated transformer moduli for the bond graph are shown in Table 13.3. For the simulation various parameters used are as enlisted in Table 13.4. The initial configuration was same as shown in Fig. 13.3.

The wire actuation motion for obtaining the 3rd DOF was simulated by activating the upward and downward wire actuation and the tip trajectory obtained from the simulation is shown in Figs. 13.7 and 13.8, where Fig. 13.7 indicates the upward wire actuation and Fig. 13.8 shows the downward wire actuation. During the actuation process all the joints were given an equal joint rotation.

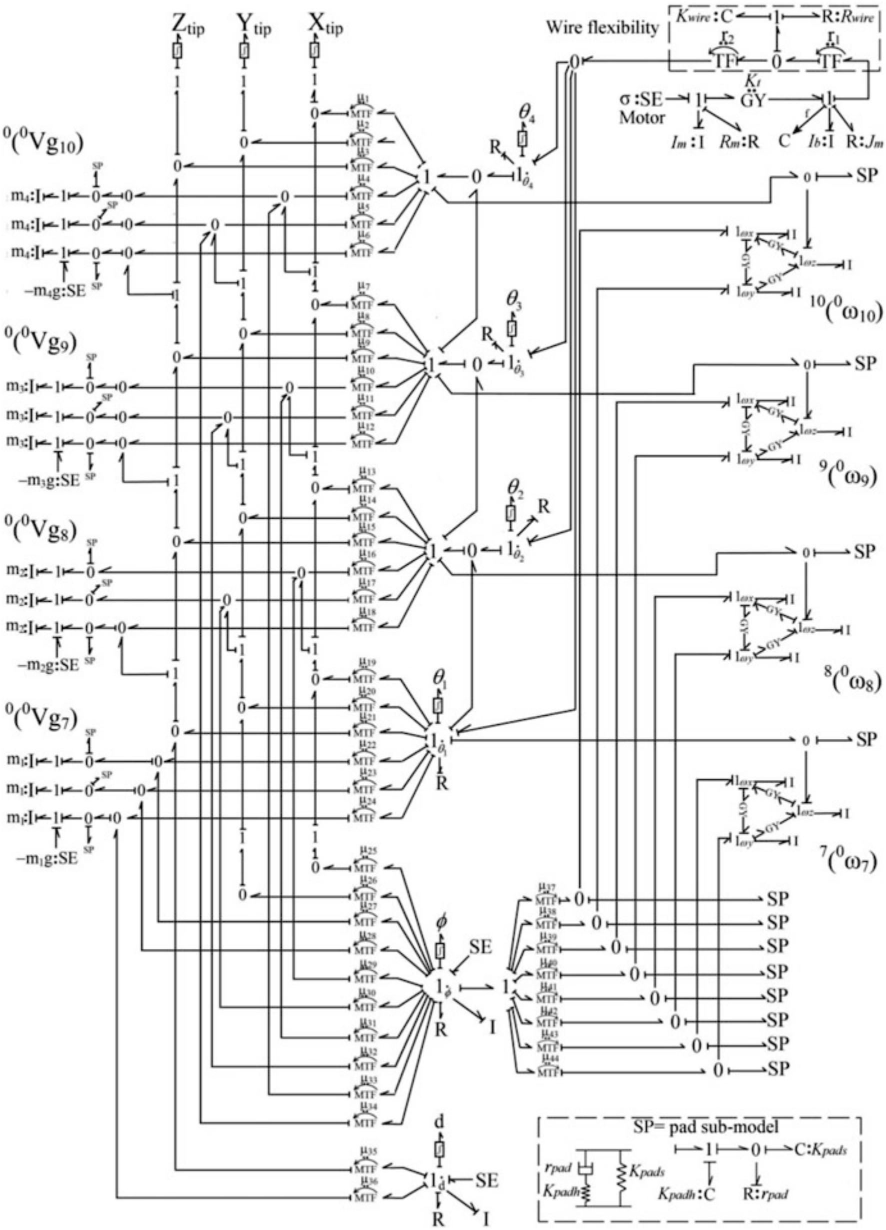


Fig. 13.6 Bond graph model of in-vivo robot with simultaneous joint actuation

Table 13.3 Simulation parameters for simultaneous joint actuation

Parameter	Value
Link length	0.044 m
Link mass	0.001 kg
Torque applied	2 Nm
Joint resistance	R_1=0.001 Nm/ (rad/s)
Motor inductance	0.0001 H
Motor resistance	0.001 Ω
Motor constant (K_t)	7.8 mNm/A
Wire resistance	0.001 Nm/ (rad/s)
Radius of pulley	0.008 m

Table 13.4 Transformer moduli for the bond graph model of the virtual link based controller

Transformer	Modulus	Transformer	Modulus
μ_1	$-L_v \sin(\theta_1 + \theta_2 + \zeta_1)$	μ_4	$L_1 \cos\theta_1$
μ_2	$L_v cos(\theta_1 + \theta_2 + \zeta_1)$	μ_5	$-0.5L_1 \sin\theta_1$
μ_3	$-L_1 \sin\theta_1$	μ_6	$0.5L_1 \cos\theta_1$

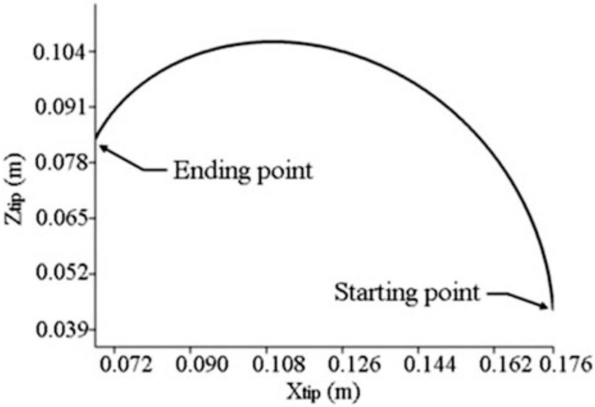


Fig. 13.7 Tip trajectory by actuating wire with tip moving in upward direction

13.4 Experimental Validation of Wire Actuation Simulation in Four-Scaled Model

Experimentation was carried out to validate the simulation results obtained for the tip positions during the upward and downward wire actuation motion. Different tip positions recorded during the upward wire actuation and four of them are shown in Fig. 13.9.

Initial tip position was recorded at (0.176, 0.044) m shown in Fig. 13.9a, 1st intermediate tip position was recorded at (0.162, 0.079) m shown in Fig. 13.9b, 2nd intermediate tip position was located at (0.108, 0.106) m shown in Fig. 13.9c, final

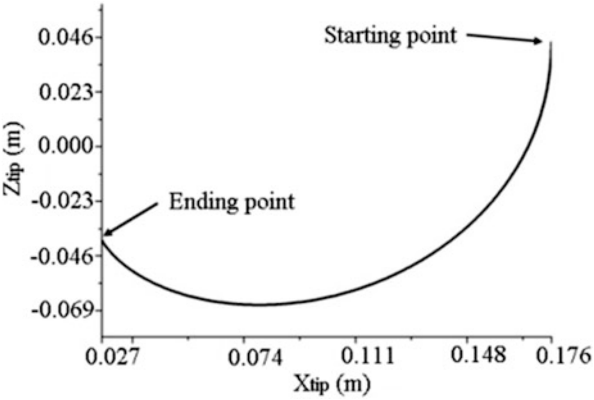


Fig. 13.8 Tip trajectory by actuating wire with tip moving in downward direction

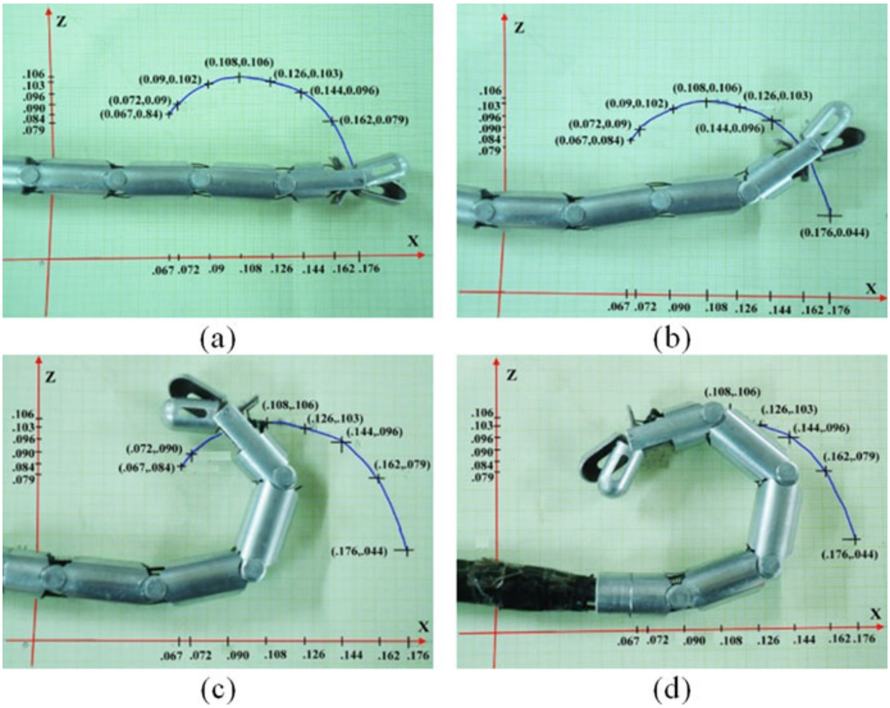


Fig. 13.9 Upward direction of motion of wire actuated four-scaled model (**a**) initial tip position, (**b**) 1st intermediate tip position, (**c**) 2nd intermediate tip position, and (**d**) final tip position

tip positions were recorded at (0.067, 0.084) m shown in Fig. 13.9d. It can be seen by comparing Figs. 13.7 and 13.9 that in experimentation the tip followed the same path in a very close approximation. Similarly the tip positions during the downward wire actuation have been shown in Fig. 13.10. Initial tip position was recorded at (0.176, 0.044) m as shown in Fig. 13.10a, 1st intermediate tip position is at (0.148,

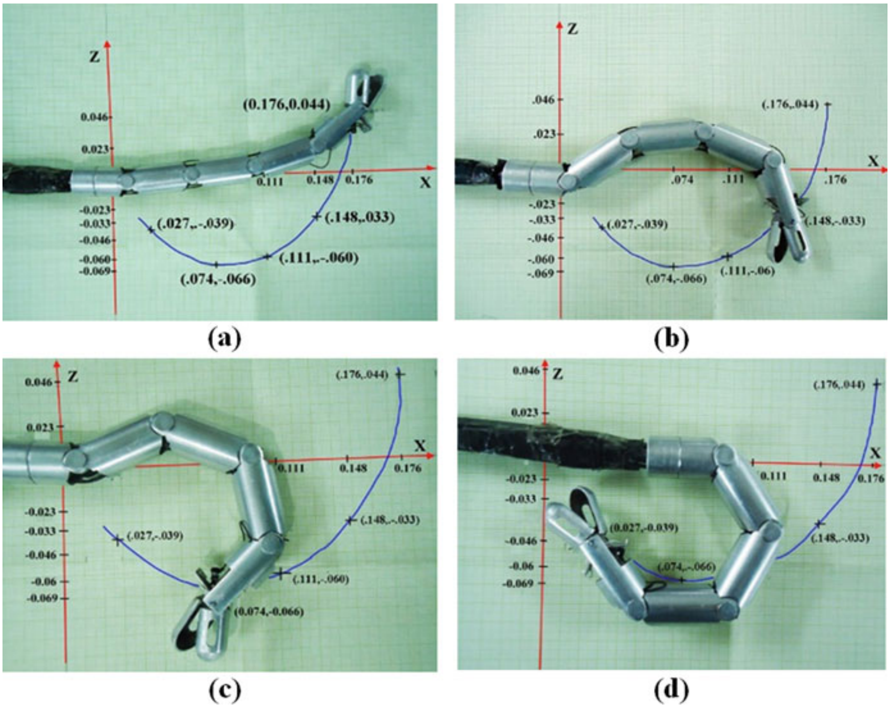


Fig. 13.10 Downward direction of motion of wire actuated four-scaled model (**a**) initial tip position, (**b**) 1st intermediate tip position, (**c**) 2nd intermediate tip position, and (**d**) final tip position

−0.033) m shown in Fig. 13.10b, 2nd intermediate tip position is obtained at (0.074, −0.066) m shown in Fig. 13.10c, final tip position is located and situated (0.027, −0.039) m shown in Fig. 13.10d.

13.5 Experimental Validation of Wire Actuation Simulation in Actual-Scaled Model

Same experimentation was also carried out on actual scale in-vivo robot to validate the simulation results obtained for the tip positions during the upward and downward wire actuation motion. Different tip positions were recorded during the upward wire actuation and four of them are shown in Fig. 13.11. The initial tip position was recorded at (0.044, 0.011) m shown in Fig. 13.11a, 1st intermediate tip position was recorded at (0.0405, 0.0197) m shown in Fig 13.11b, 2nd intermediate tip position was located at (0.0225, 0.0255) m shown in Fig. 13.11c, final tip positions were recorded at (0.0167, 0.021) m shown in Fig. 13.11d. It can be seen by comparing Figs. 13.9 and 13.11 that in experimentation the tip followed the same tip position in a scaled manner.

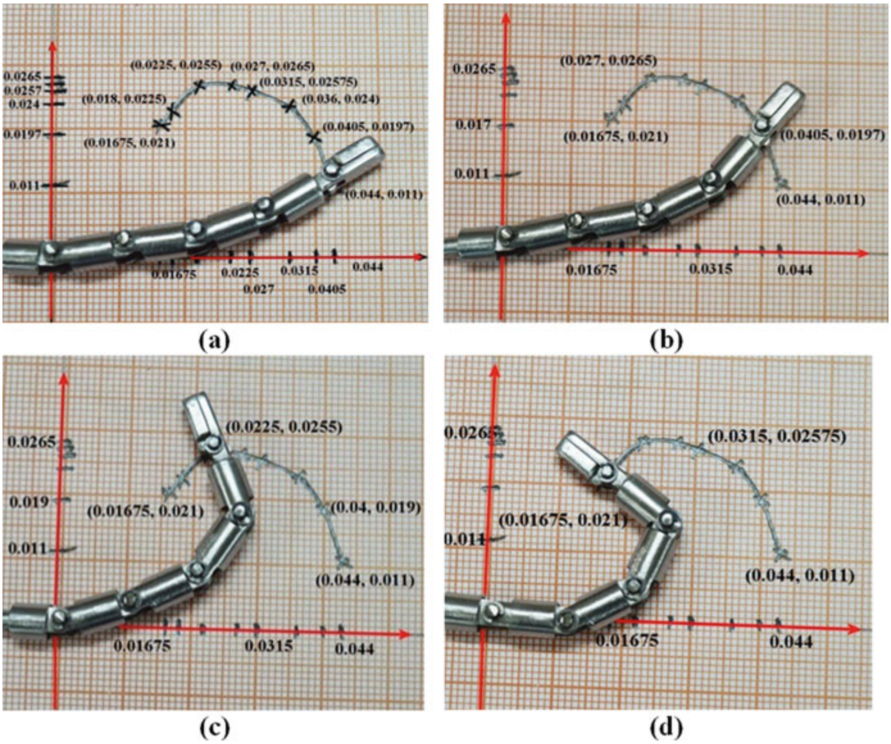


Fig. 13.11 Upward direction of motion of wire actuated actual scale model (**a**) initial tip position, (**b**) 1st intermediate tip position, (**c**) 2nd intermediate tip position, and (**d**) final tip position

Similarly the tip positions during the downward wire actuation have been shown in Fig. 13.12. Initial tip position was recorded at (0.044, 0.011) m as shown in Fig. 13.12a, 1st intermediate tip position is at (0.037, −0.008) m shown in Fig. 13.12b, 2nd intermediate tip position is obtained at (0.0277, −0.015) m shown in Fig. 13.12c, final tip position is located and situated at (0.0675, −0.0097) m shown in Fig. 13.12d.

13.6 Inverse Kinematic Analysis

In this section the inverse kinematic analysis of the proposed 4-DOF in-vivo robot has been explained. The tip Cartesian coordinates can be obtained from the kinematic analysis as expressed in Eqs. (13.2), (13.3), and (13.4). Now, assuming all the links of the manipulator to be equal, the link lengths can be expressed as,

$L_1 = L_2 = L_3 = L_4 = L_l$ and a planar case, i.e., $\phi = 0$ and with all joint angles being equal, i.e.,

$$\theta_1 = \theta_2 = \theta_3 = \theta_4 = \theta, \tag{13.24}$$

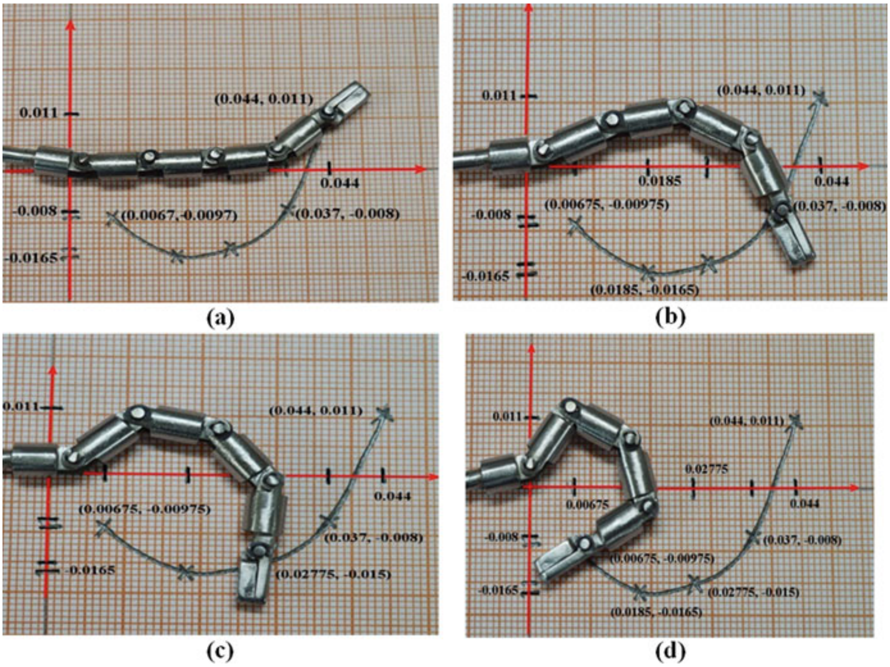


Fig. 13.12 Downward direction of motion of wire actuated actual scale model (**a**) initial tip position, (**b**) 1st intermediate tip position, (**c**) 2nd intermediate tip position, and (**d**) final tip position

Equations (13.2), (13.3), and (13.4) can be expressed as:

$$X_{\text{tip}} = L_l\,(\cos\theta + \cos 2\theta + \cos 3\theta + \cos 4\theta) \tag{13.25}$$

$$Y_{\text{tip}} = 0 \tag{13.26}$$

$$Z_{\text{tip}} = (L + d) + L_l\,(\sin\theta + \sin 2\theta + \sin 3\theta + \sin 4\theta) \tag{13.27}$$

Assuming there is no translational motion, i.e., $d = 0$ and $L_l = 0.044$ m (for four-scaled model), the joint angle θ can be expressed for a given value of X_{tip} and Z_{tip} as:

$$\theta = \arccos\left\{\frac{1}{3}\left(\frac{\sqrt[3]{3\sqrt{3}\times\sqrt{27A^2-4A}+27A-2}}{\sqrt[3]{2}} + \frac{\sqrt[3]{2}}{\sqrt[3]{3\sqrt{3}\times\sqrt{27A^2-4A}+27A-2}} - 1\right)\right\} \tag{13.28}$$

where

$$A = 64.566\left(X_{\text{tip}}^2 + Z_{\text{tip}}^2\right) - 5.682Z_{\text{tip}} + 0.125 \tag{13.29}$$

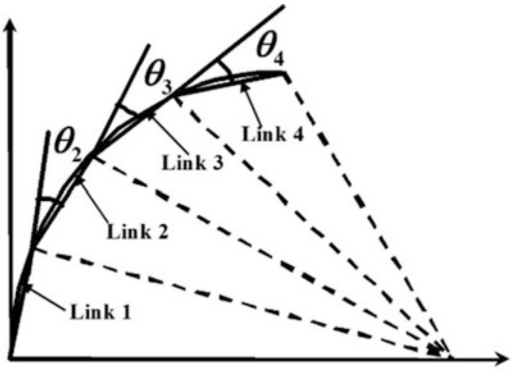


Fig. 13.13 Representation of wire actuated section of in-vivo robot arm

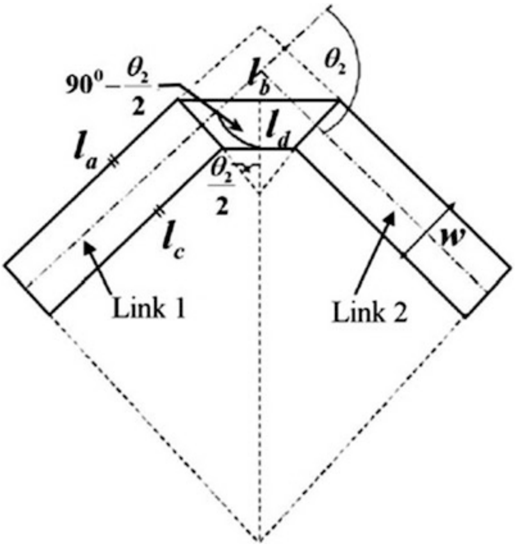


Fig. 13.14 Link segment between link 1 and link 2

13.6.1 Determination of Joint Angle in Terms of Wire Length

A relational approach has been established to express joint angle in terms of the length of outer and inner wire used in actuation. In the analysis it is assumed that the wire actuated portion comprising of four articulated links as shown in Fig. 13.13 is a perfect arc of a circle.

Now considering a two link segment of the above entire section, i.e., link 1 and link 2 of joint angle θ_2 shown in Fig. 13.14, and considering the wire actuation on the two link segment length of outer and inner wire length are as follows, respectively:

$$l_2 = l_a + l_b \text{ and,}$$

$$l_1 = l_c + l_d$$

The relation between the outer and inner wire length can be obtained by considering the whole wire actuated portion as shown in Fig. 13.14., and with a known relation $l_a = l_c$.

The outer wire length is

$$L_o = 4l_a + 3l_b \tag{13.30}$$

and the inner wire length is

$$L_o = 4l_a + 3l_b \tag{13.31}$$

From Fig. 13.14 the following geometrical relation can be deduced:

$$\sin\frac{\theta_2}{2} = \frac{l_b - l_d}{2D} \tag{13.32}$$

where D is the diameter of link. Substituting the values of l_b and l_d from Eqs. (13.30) and (13.31) in Eq. (13.32), we have

$$\sin\frac{\theta_2}{2} = \frac{\left(\frac{L_o - 4l_a}{3}\right) - \left(\frac{L_i - 4l_c}{3}\right)}{2D} \tag{13.33}$$

Further simplifying the difference in outer and inner wire length can be expressed as:

$$L_0 - L_i = 6D\sin\frac{\theta}{2} \tag{13.34}$$

If wire is wrapped on the same pulley, with direction of wrapping for outer wire to be opposite to that of inner wire, the length of wire wrapped on pulley is given by,

$$L_0 - L_i = r_p.\theta_p \tag{13.35}$$

where r_p is the radius of pulley and θ_p is the pulley rotation. Substituting the value of $L_o - L_i$ from Eq. (13.34) in Eq. (13.35) the pulley rotation can be expressed as:

$$\theta_p = \left(6D/r_p\right) sin\,(\theta/2) \tag{13.36}$$

Thus for a given X_{tip} and Z_{tip} one can find θ from Eq. (13.28) and for this θ, the value of pulley rotation θ_p in terms of tip positions can be obtained. A Proportional Derivative (PD) controller can be used to take the pulley to desired θ_p rotation, thus the robot can be moved to a desired position. To implement the inverse kinematics solution the motor voltage for a desired pulley rotation corresponding to desired tip location can be evaluated as:

$$V = K_p\left(\theta_p - \theta_a\right) + K_v\left(\dot{\theta}_p - \dot{\theta}_a\right) \tag{13.37}$$

where V is the voltage applied to the motor, K_p is proportional gain, K_V is derivative gain, θ_P is desired pulley rotation in radian, θ_a is the actual pulley rotation in radian, $\dot{\theta}_P$ is desired angular velocity of the pulley, and $\dot{\theta}_a$ is actual angular velocity of the pulley.

13.6.2 Implementation of the Control Scheme

Desired tip positions are given as input to the bond graph and the joint angles are evaluated from the relations. Then from the joint angle, difference in outer and inner wire length is evaluated and finally the pulley rotation required to cover the wire length difference is obtained. Then the voltage supplied to motor to rotate the pulley, to wrap and unwrap the wire on the opposite sides, for the tip of the robot to reach from its current position to the point of interest is calculated. This calculation is based on the PD control law. Voltage is given as input to the system. Evaluation of motor voltage for desired tip position of robot has been explained in the form of a flowchart shown in Fig. 13.15.

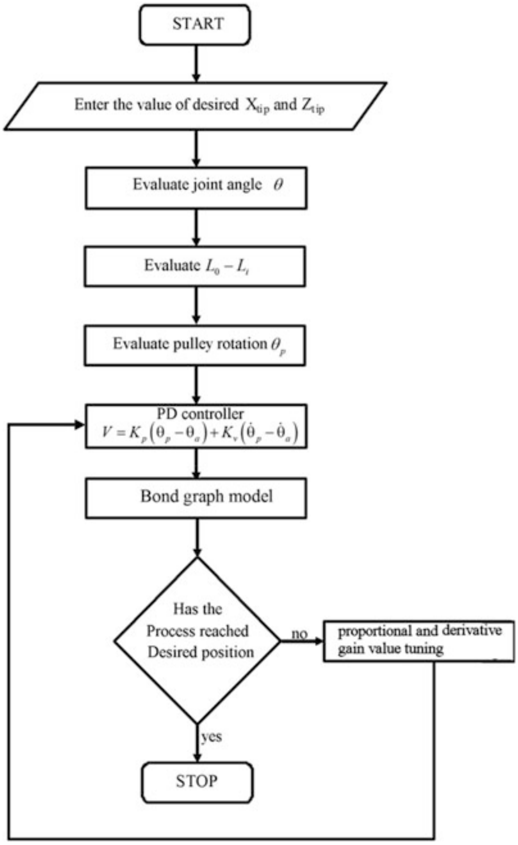
13.6.3 Simulation and Experimental Results

Simulation results obtained for three different tip trajectories in workspace are shown in Fig. 13.16. Simulations have been carried out to move the tip from an initial tip position of $X_{\mathrm{tip}} = 0.1565$ m and $Z_{\mathrm{tip}} = 0.1170$ m to $X_{\mathrm{tip}} = -0.022$ m and $Z_{\mathrm{tip}} = -0.012$ m, as shown by trajectory 1 in Fig. 13.16. Then the tip was moved from an initial position of $X_{\mathrm{tip}} = 0.1047$ m and $Z_{\mathrm{tip}} = 0.1688$ m to $X_{\mathrm{tip}} = -0.066$ m and $Z_{\mathrm{tip}} = 0.013$ m, as shown by trajectory 2 in Fig. 13.16. Further a tip movement from an initial tip position of $X_{\mathrm{tip}} = 0.0381$ m and $Z_{\mathrm{tip}} = 0.1862$ m to a final tip position of $X_{\mathrm{tip}} = 0.022$ m and $Z_{\mathrm{tip}} = 0.108$ m has been performed by trajectory 3 as shown in Fig. 13.16.

First trajectory was obtained after simulating the system for 6.5 s; second trajectory was obtained after a simulation of 8.25 s, and the third trajectory after a simulation of 3.4 s. The tip velocities with respect to time for the above mentioned trajectories are shown in Fig. 13.17. In the plot an initial drift was recorded which was minimized later with time.

Animations of the 4-DOF in-vivo robotic manipulator for the 1st, 2nd, and 3rd trajectory are shown in Fig. 13.18a–c, respectively. Different joint postures during the simulation from the initial position to the desired position are shown in these figures. Fig. 13.18a shows the animation for the first trajectory, Fig. 13.18b shows the animation for the second trajectory, and Fig. 13.18c shows the animation for the third trajectory.

Fig. 13.15 Flow chart for tip motion control


START
Enter the value of desired X_{tip} and Z_{tip}
Evaluate joint angle θ
Evaluate $L_0 - L_i$
Evaluate pulley rotation θ_p
PD controller
$V = K_p(\theta_p - \theta_a) + K_v(\dot{\theta}_p - \dot{\theta}_a)$
Bond graph model
Has the Process reached Desired position
no
proportional and derivative gain value tuning
yes
STOP

13.6.4 Experimental Validation of Desired Tip Positions

Experiments have been carried out on the actual scale model of in-vivo robot to validate the simulation results obtained for desired tip positions. The desired tip positions were obtained in the experimentation by rotating the pulley through wire actuation. Experimentations were performed to validate the three different trajectories obtained in simulation. For the actual scale model the initial position was recorded at (0.039, 0.0292) m shown in Fig. 13.19a and the final tip position was obtained at (–0.005, –0.003) m shown in Fig. 13.19d. Intermediate tip locations are at (0.0375, 0.01575) m and at (0.0325, 0.0077) m are shown in Fig. 13.19b, c, respectively. Figure 13.20 indicates the initial, two intermediate, and final tip positions during the second trajectory for the actual-scaled model. The initial tip position was at (0.02675, 0.042) m shown in Fig. 13.20a and the final tip position was measured to be (–0.0165, 0.003) m shown in Fig. 13.20d. The intermediate tip positions recorded during the experiments are shown in Fig. 13.20b, c respectively. Similarly the tip positions for the 3rd trajectory of actual scale model were recorded and are shown in Fig. 13.21.

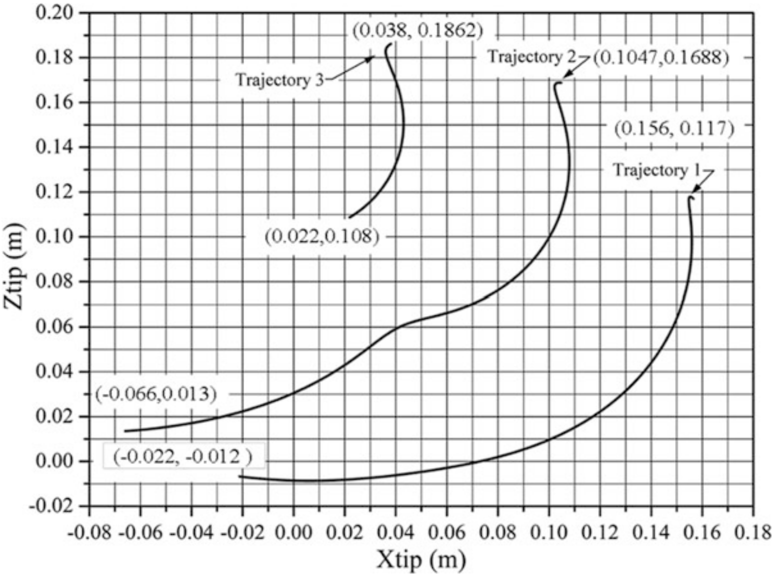


Fig. 13.16 Three different tip trajectories in workspace

The initial tip position was recorded at (0.09, 0.0465) m shown in Fig. 13.21a, 1st intermediate tip position was recorded at (0.105, 0.04) m shown in Fig. 13.21b, 2nd intermediate tip position was obtained at (0.0085, 0.03) m shown in Fig. 13.21c, and the final tip position was located at (0.005, 0.02675) m plotted in Fig. 13.21d.

Comparisons between the trajectories obtained in simulation and from experimental results for the three cases are shown in Figs. 13.22, 13.23, and 13.24. Comparison between the simulation and experimental result obtained for the first trajectory is shown in Fig. 13.22. Similarly the comparison between the simulation and experimental results for the second and third trajectories is shown in Figs. 13.23 and 13.24, respectively. Difference between *X* and *Z* coordinates gives the deviation between simulated and experimental results for the three cases. This can be found from numerical data of simulated and experimental values

The comparison was done between the simulation and experimental results for the four-scaled model and the actual scale for the three trajectories. From the comparison of only experimental results between the four-scaled and actual scale model it was evident that the actual scale model follows the same proportionate trajectory as that of the four-scaled model.

The deviation between simulated and experimental results of *X* and *Z* coordinates for tip position in three different cases for the four-scaled model are $\left(2.4 \times 10^{-6}, 4.1 \times 10^{-6}\right)$ *m* for the 1st trajectory, $\left(5.632 \times 10^{-4}, 4.25 \times 10^{-5}\right)$ m for the 2nd trajectory, and $\left(2.97 \times 10^{-5}, 2 \times 10^{-5}\right)$ m for the 3rd trajectory. In fact this is the actual condition faced by surgeons where they would be interested in taking the robot tip from one position to the desired position.

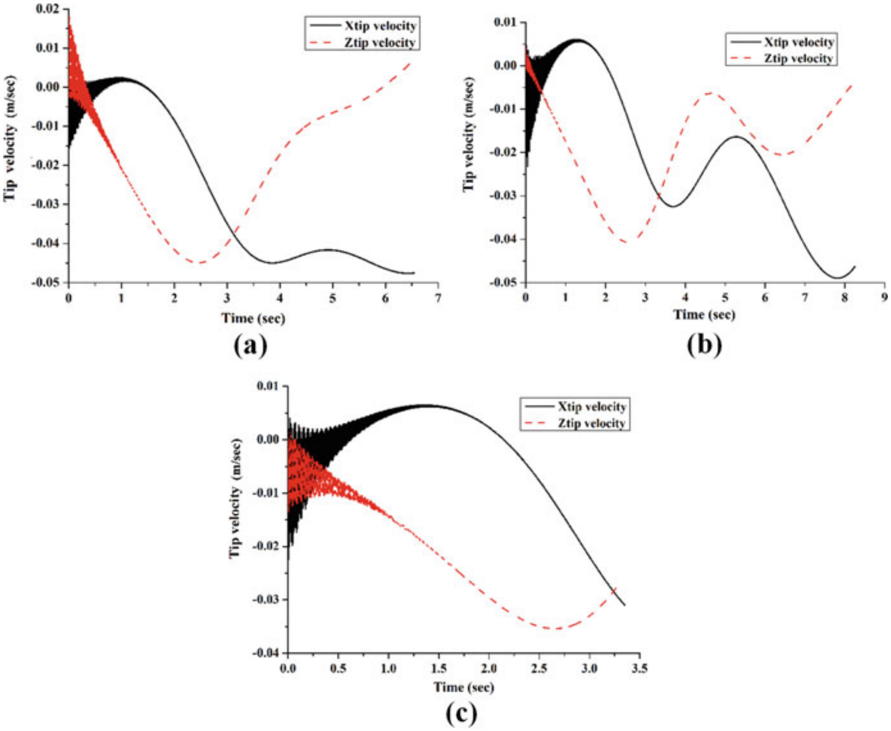


Fig. 13.17 X_{tip} and Z_{tip} trajectory versus time for (**a**) 1st trajectory, (**b**) 2nd trajectory, and (**c**) 3rd trajectory

13.7 Trajectory Control Analysis of In-Vivo Robot

Controlling hyper-redundant robotic manipulators as in-vivo manipulators is a major concern for researchers working in this particular field. During biopsy it is required to move the manipulator in the abdominal cavity environment and to place its tip at the desired point of interest. For this a proper trajectory control scheme is required.

A trajectory control scheme for a planar hyper-redundant manipulator has been proposed. The scheme is based on a virtual link based controller. An example of three link planar manipulator is considered to illustrate the methodology. The 3-link planar manipulator was modelled using bond graph, and a controller, using the virtual link concept has been formulated. Relations between the real and virtual links were established to design the controller and the virtual link based controller was used to send effort signals to all the joints of 3-link planar manipulator upon receiving a corrective signal from PI controller. Different desired trajectory shapes were used to check the efficiency of the controller. Simulations and animations have been performed to test the competence of the controller.

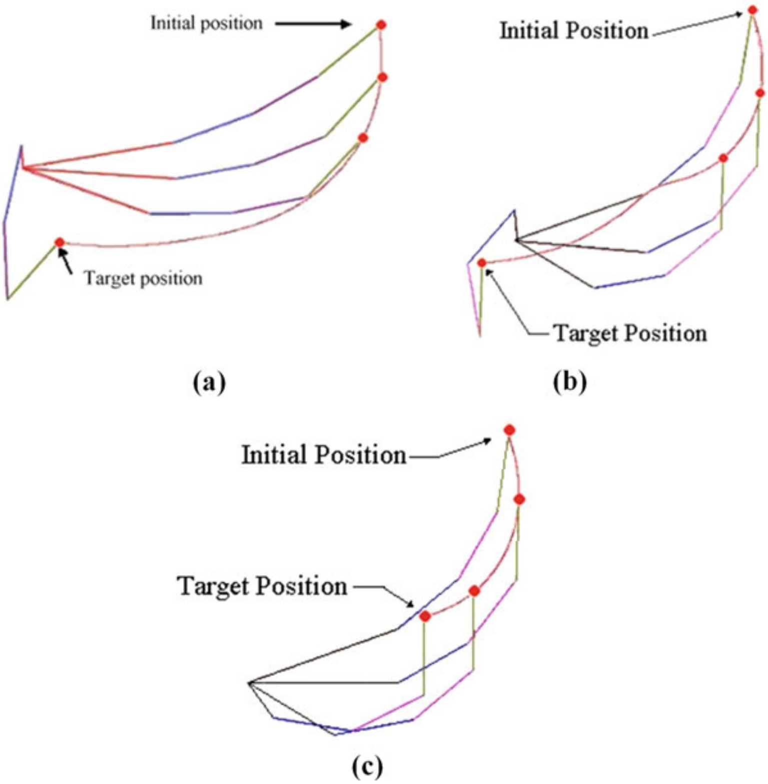


Fig. 13.18 Animation results of 4-DOF in-vivo robot for (**a**) 1st trajectory, (**b**) 2nd trajectory, and (**c**) 3rd trajectory

13.7.1 Virtual Link Based Controller for Trajectory Control of 3-Link Planar Redundant Manipulator

The trajectory control of in-vivo robot is based on the virtual link based controller [13]. In the present approach the numbers of real links present in the controller are reduced to the minimum number of fictitious links, called virtual links. This proposed control scheme is applicable to any number of planar hyper-redundant manipulator links. A generalized solution technique for any number of planar links (more than one) has been discussed before its implementation to the 3-link planar redundant manipulator. For simplicity a 5-link planar manipulator example shown in Fig. 13.25 has been considered. The derivation of the controller can be explained in the following steps:

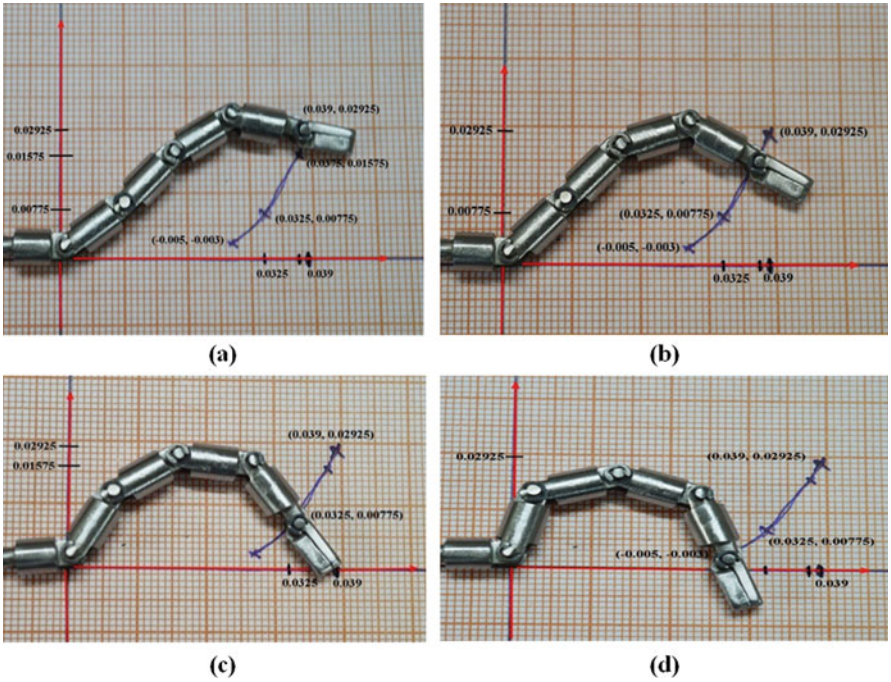


Fig. 13.19 Experimental results of tip position for 1st trajectory of actual scale model (**a**) initial, (**b**) 1st intermediate, (**c**) 2nd intermediate, and (**d**) final tip position

13.7.1.1 Step I

In the first step shown in Fig. 13.25a two real links L_5 and L_4 are replaced with a fictitious link of varying length L_{v_1}. The joint angle of the virtual link L_{v_1} can be expressed as $(\zeta_2 + \theta_4)$, where θ_4 is the angle of real link L_4 with respect to its previous link L_3, ζ_2 is the angle between L_4 and L_{v_1}. Also in this figure ζ_1 is the angle between L_5 and L_{v_1}. From the geometrical relations, the angles ζ_1 and ζ_2 can be deduced as,

$$\zeta_1 + \zeta_2 = \theta_5 \tag{13.38}$$

The virtual link L_{v_1} can be expressed as,

$$L_{v_1}^2 = L_4^2 + L_5^2 - 2L_4L_5\cos(\pi - \theta_5)$$

Simplifying the above equation we have,

$$L_{v_1} = \sqrt{L_4^2 + L_5^2 + 2L_4L_5\cos\theta_5} \tag{13.39}$$

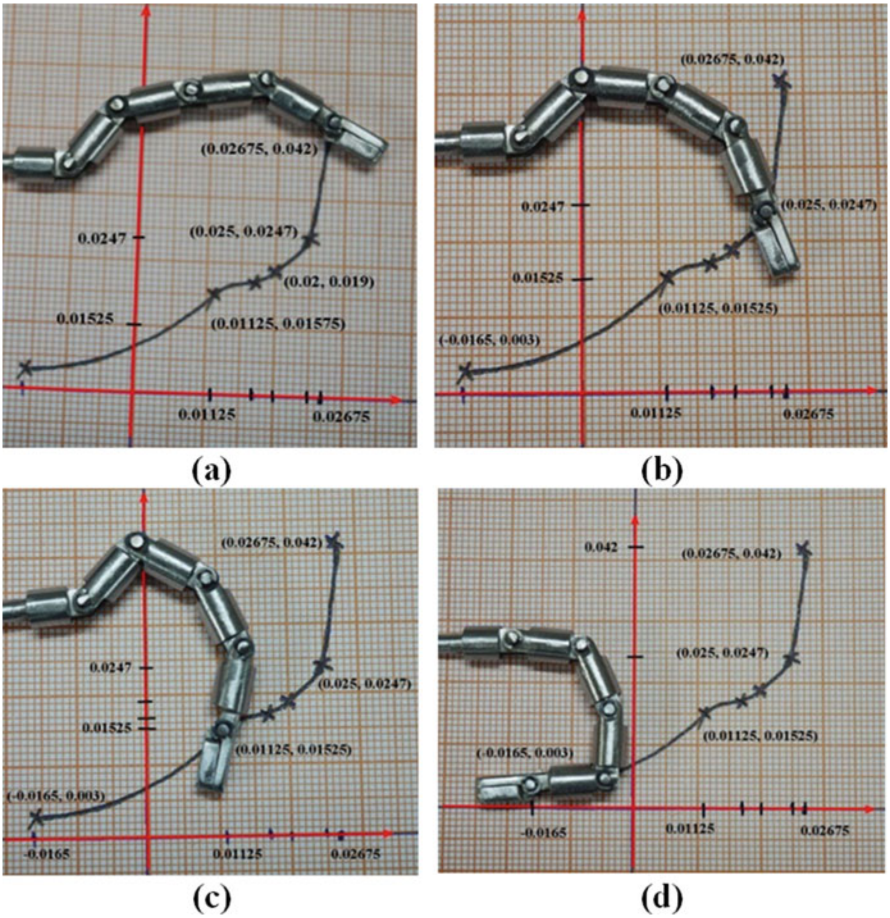


Fig. 13.20 Experimental results of tip position for 2nd trajectory of actual scale model (**a**) initial, (**b**) 1st intermediate, (**c**) 2nd intermediate, and (**d**) final tip position

13.7.1.2 Step II

In the second step shown in Fig. 13.25b, the fictitious link L_{v_1} and real link L_3 are replaced with an imaginary link L_{v_2}. In this figure ζ_3 is the angle between L_{v_1} and L_{v_2}, and ζ_4 is the angle between L_3 and L_{v_2}. The geometrical relation between the angles can be expressed as,

$$\zeta_3 + \zeta_4 = \zeta_2 + \theta_4 \tag{13.40}$$

Length of L_{v_2} can be expressed as,

$$L_{v_2}^2 = L_{v_1}^2 + L_3^2 - 2L_{v_1}L_3 \cos\{\pi - (\zeta_2 + \theta_4)\}$$

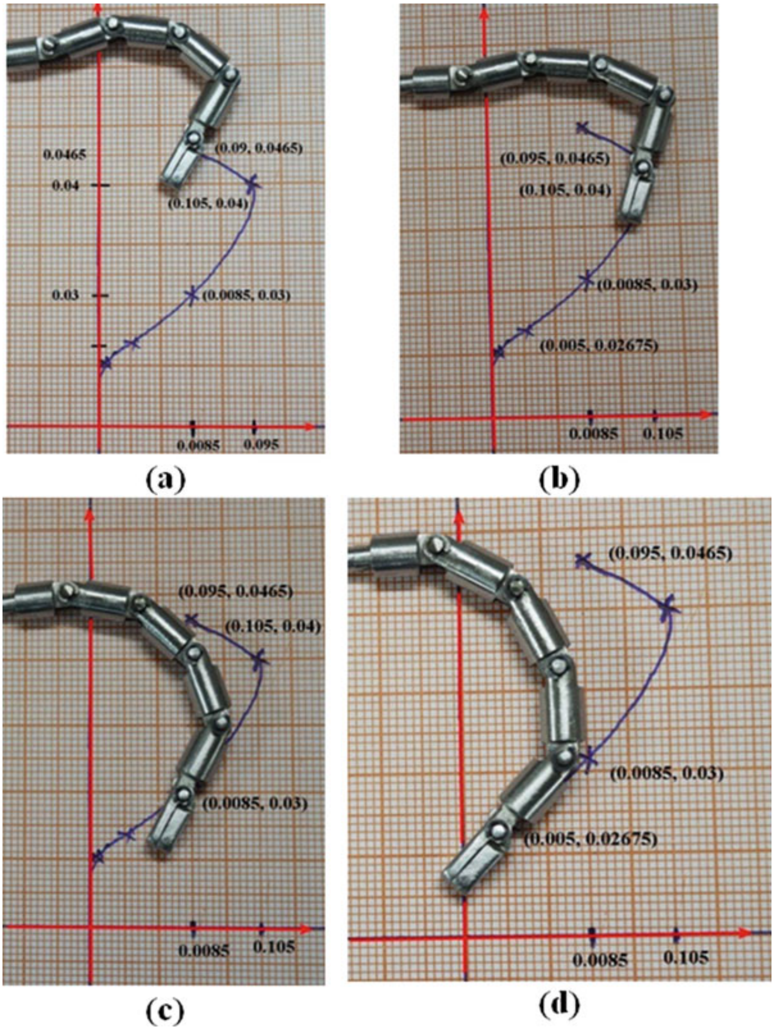


Fig. 13.21 Experimental results of tip position for 3rd trajectory of actual scale model (**a**) initial, (**b**) 1st intermediate, (**c**) 2nd intermediate, and (**d**) final tip position

Thus

$$L_{v_2} = \sqrt{L_{v_1}^2 + L_3^2 + 2L_{v_1}L_3\cos(\zeta_2 + \theta_4)} \tag{13.41}$$

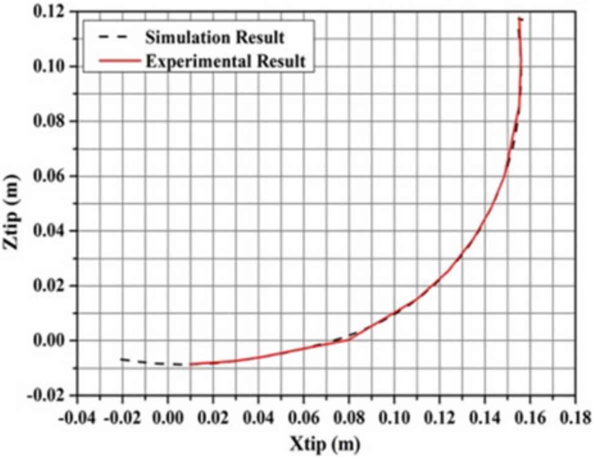


Fig. 13.22 Comparison between simulation and experimental result for 1st trajectory for four-scaled model

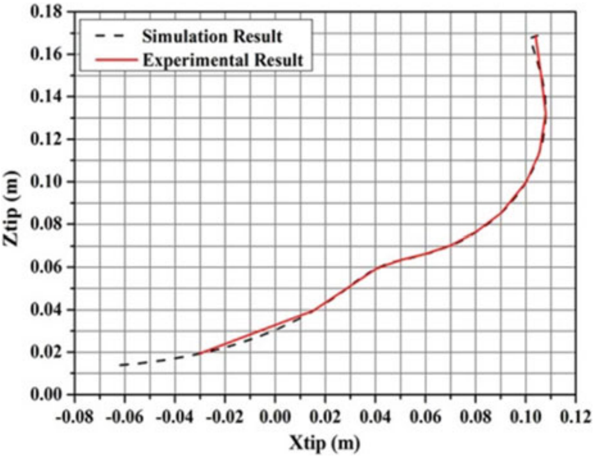


Fig. 13.23 Comparison between simulation and experimental result for 2nd trajectory for four-scaled model

13.7.1.3 Step III

In the third step shown in Fig. 13.25c, the fictitious link L_{v_2} and real link L_2 are replaced with an imaginary link L_{v_3}. In this figure ζ_5 is the angle between L_{v_3} and L_{v_2} whereas ζ_6 is the angle between L_2 and L_{v_3}. The geometrical relation between the angles can be expressed as,

$$\zeta_5 + \zeta_6 = \zeta_4 + \theta_3 \tag{13.42}$$

The fictitious link length L_{v_3} can be expressed as,

$$L_{v_3}^2 = L_{v_2}^2 + L_2^2 - 2L_{v_2}L_2 \cos\{\pi - (\zeta_4 + \theta_3)\},$$

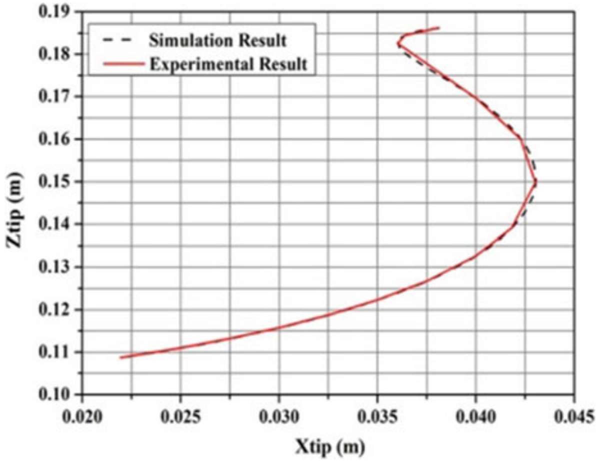


Fig. 13.24 Comparison between simulation and experimental result for 3rd trajectory for four-scaled model

or

$$L_{v_3} = \sqrt{L_{v_2}^2 + L_2^2 + 2L_{v_2}L_2\cos(\zeta_4 + \theta_3)} \tag{13.43}$$

13.7.1.4 Step IV

The real links of the 5-link manipulator are converted to one virtual and one real link, as shown in Fig. 13.25d. The length of the virtual link is expressed in Eq. (13.43), and the joint angle of this link is $(\zeta_6 + \theta_2)$. Using the above developed methodology a virtual link based controller to control the 3-link planar manipulator in a two-dimensional workspace has been designed. This controller will provide the necessary joint torque required to each joint for a given end-effector trajectory. The steps described above in Sects. 13.7.1.1, 13.7.1.2, 13.7.1.3, and 13.7.1.4 have been followed for the development of virtual link based controller for a 3-link planar manipulator shown in Fig. 13.26. The controller is composed of two links (one real and one imaginary) and two revolute joints. The joint angles $(\theta_1, \theta_2, \zeta_1)$ define the virtual link manipulator configuration. In the virtual link based controller an imaginary link of length L_v has been used instead of two actual links of length L_2 and L_3. The length of the virtual link from the geometrical relation can be deduced as,

$$L_v^2 = L_2^2 + L_3^2 - 2L_2L_3\cos(\pi - \theta_3) \tag{13.44}$$

Further simplifying Eq. (13.44) we have the following relation:

$$L_v = \sqrt{L_2^2 + L_3^2 + 2L_2L_3\cos\theta_3} \tag{13.45}$$

Also, $\zeta_1 + \zeta_2 + (\pi - \theta_3) = \pi$

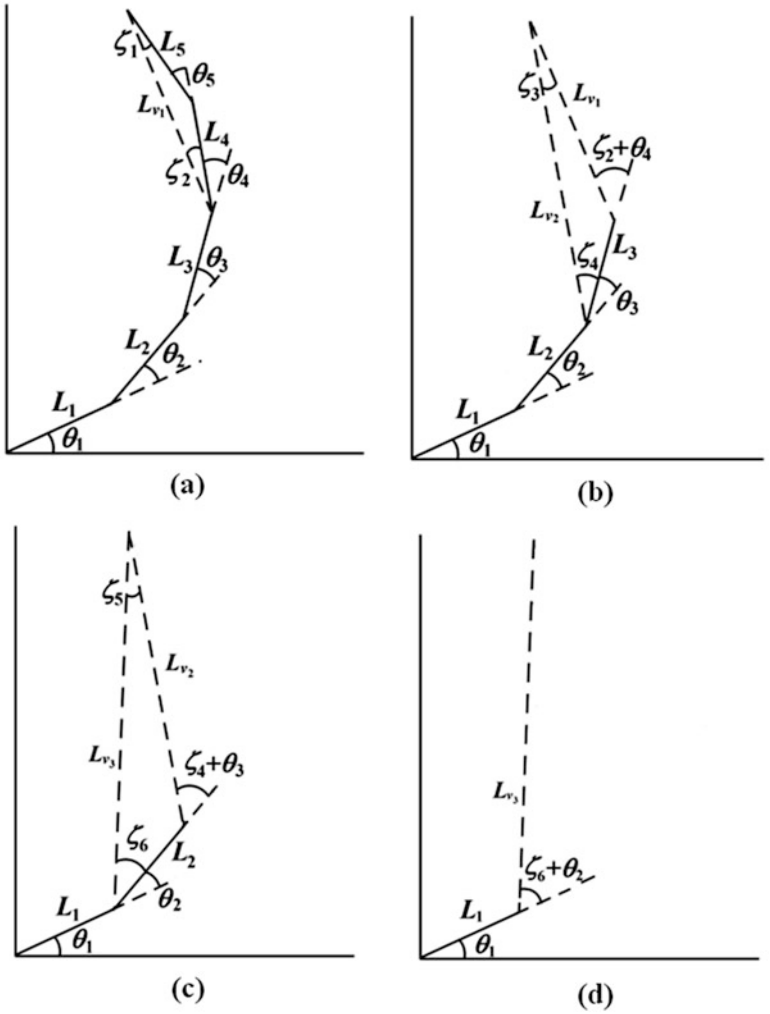


Fig. 13.25 Derivation of controller for a 5-DOF planar manipulator (**a**) Step I, (**b**) Step II, (**c**) Step III, and (**d**) Step IV

Thus, $\zeta_1 + \zeta_2 = \theta_3$

Assuming that the real links have equal lengths, we have $\zeta_1 = \zeta_2$. So,

$$\zeta_1 = \zeta_2 = \theta_3/2 \tag{13.46}$$

Thus the virtual manipulator link length can be expressed as given in Eq. (13.45) and its joint angle can be expressed as $(\zeta_1 + \theta_2)$ with respect to link of length L_1. The tip position of the virtual manipulator in X and Y directions can be expressed as,

$$X_{\text{tip\_vir}} = L_1 \cos\theta_1 + L_v \cos(\theta_1 + \theta_2 + \zeta_1) \tag{13.47}$$

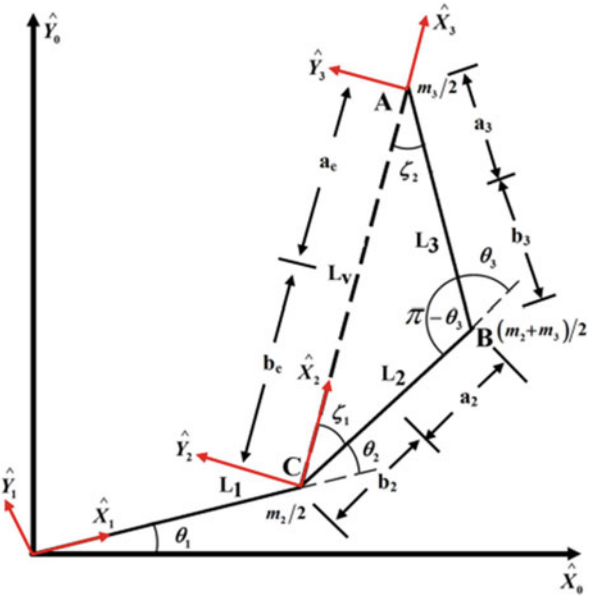


Fig. 13.26 Schematic diagram of virtual link based controller

$$Y_{\text{tip\_vir}} = L_1 \sin\theta_1 + L_v sin\,(\theta_1 + \theta_2 + \zeta_1) \tag{13.48}$$

The corresponding tip velocities can be obtained by differentiating the above equations and can be expressed as,

$$\dot{X}_{\text{tip\_vir}} = -L_1\dot{\theta}_1 \sin\theta_1 - L_v\left(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\zeta}_1\right) sin\,(\theta_1 + \theta_2 + \zeta_1) \tag{13.49}$$

$$\dot{Y}_{\text{tip\_vir}} = L_1\dot{\theta}_1 \cos\theta_1 + L_v\left(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\zeta}_1\right) cos\,(\theta_1 + \theta_2 + \zeta_1) \tag{13.50}$$

From Eqs. (13.49) and (13.50) different transformer moduli required for developing the controller based on bond graph for a 3-link planar manipulator can be derived as shown in Fig. 13.27. The transformer moduli obtained from the above mentioned equations are given in Table 13.4.

These transformer moduli have been used to construct the bond graph model of the virtual link based controller. The joints of virtual link based controller are represented by $1_{\dot{\zeta}_1}$, $1_{\dot{\zeta}_2}$ and $1_{\dot{\zeta}_1}$ junction. The difference in the reference and actual velocity of the manipulator is termed as the error. The error signal is sent to the proportional-integral (PI) controller and it minimizes the error and sends it to the virtual link based controller. The controller receives the signal from the PI controller and supplies the required amount of torque to the 3-link planar hyper-redundant manipulator to obtain the desired tip trajectory. The modulus value at the $1_{\dot{\zeta}_1}$ junction has been taken as 0.5 as per the relation obtained in Eq. (13.46).

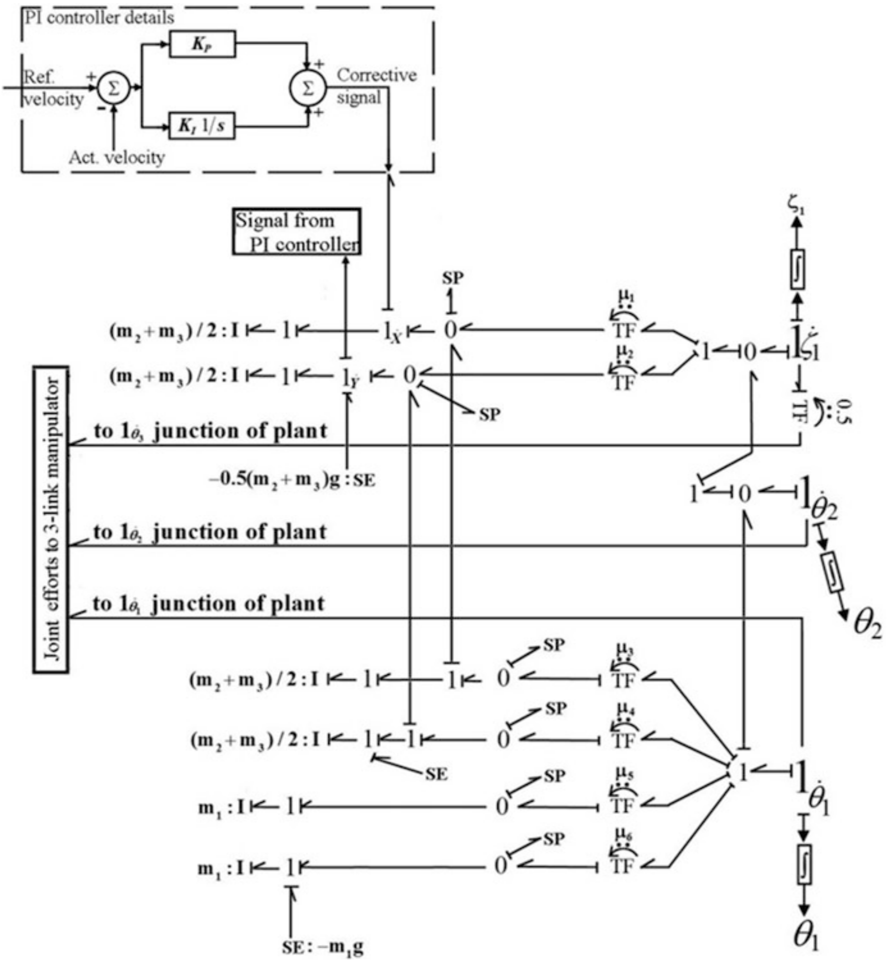


Fig. 13.27 Bond graph model of virtual link based controller for 3-link planar manipulator

The virtual link based controller is then modelled with dynamic equivalence along with the plant to simulate different trajectories. The dynamic equivalence is based on the following three conditions according to [14].

1. The total mass of the real links must be equal to that of the virtual link
2. The centre of gravity (CG) of equivalent system must coincide with that of the virtual link
3. Total moment of inertia about an axis through the CG must be equal to that of the virtual link.

The bond graph model of the dynamically equivalent virtual link based controller along with the three link planar manipulator is shown in Fig. 13.28 and the

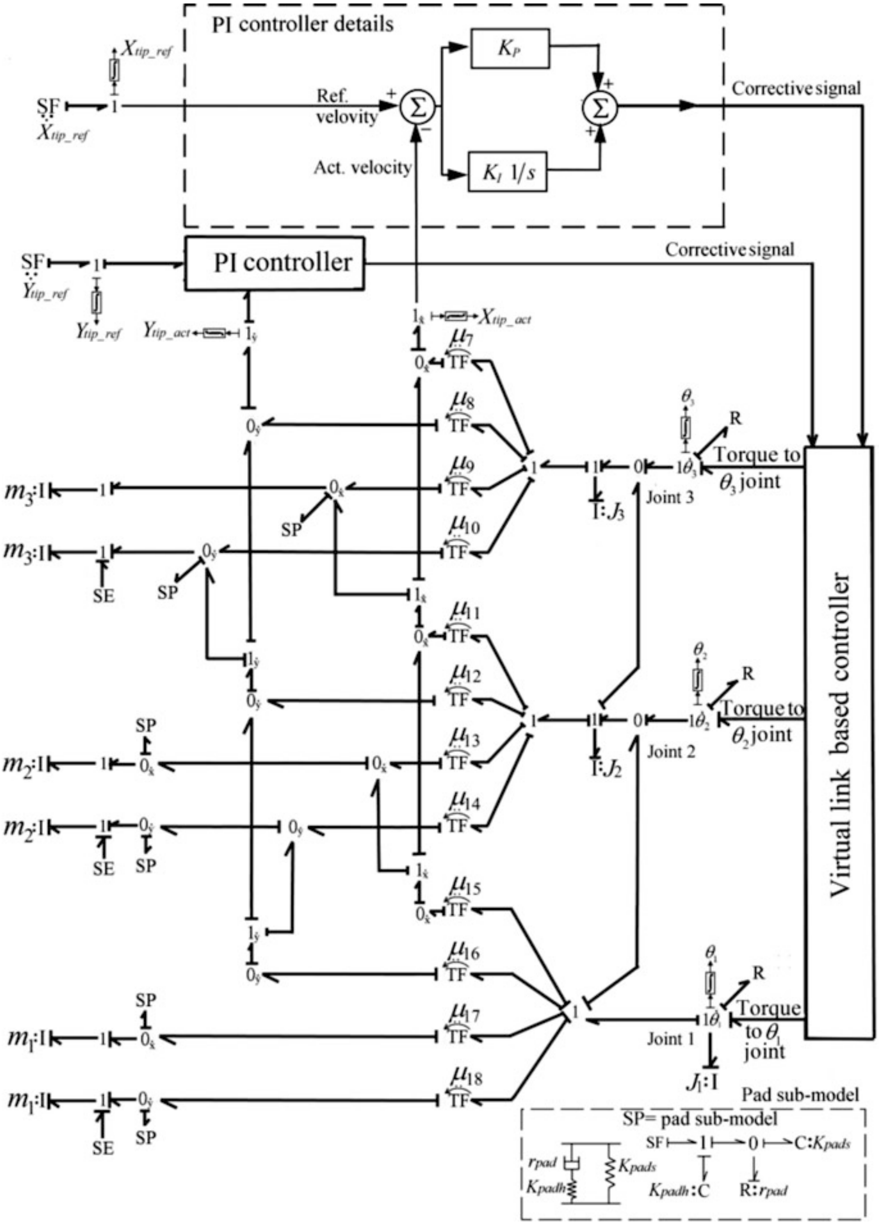


Fig. 13.28 Bond graph model of 3-link planar manipulator

translational dynamics of each link segment in *X* and *Y* direction is shown on the left-hand side of the model. Each link has the *CG* velocity which depends on the link inertia. Therefore in the bond graph model *I* elements are attached at each

Table 13.5 D-H parameters of 3-link planar manipulator

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	L_1	0	θ_2
3	0	L_2	0	θ_3
4	0	L_3	0	0

Table 13.6 Transformer moduli for the bond graph model of the 3-link planar manipulator

Transformer	Modulus
μ_7	$-L_1\sin\theta_1 - L_2\sin(\theta_1+\theta_2) - 0.5L_3\sin(\theta_1+\theta_2+\theta_3)$
μ_8	$-L_1\sin\theta_1 - L_2\sin(\theta_1+\theta_2) - L_3\sin(\theta_1+\theta_2+\theta_3)$
μ_9	$L_1\cos\theta_1 + L_2 cos(\theta_1+\theta_2) + L_3 cos(\theta_1+\theta_2+\theta_3)$
μ_{10}	$L_1\cos\theta_1 + L_2 cos(\theta_1+\theta_2) + 0.5L_3 cos(\theta_1+\theta_2+\theta_3)$
μ_{11}	$-L_1 sin\theta_1 - 0.5L_2 sin(\theta_1+\theta_2)$
μ_{12}	$-L_1 sin\theta_1 - L_2 sin(\theta_1+\theta_2)$
μ_{13}	$L_1\cos\theta_1 + L_2 cos(\theta_1+\theta_2)$
μ_{14}	$L_1\cos\theta_1 + 0.5L_2 cos(\theta_1+\theta_2)$
μ_{15}	$-0.5L_1\sin\theta_1$
μ_{16}	$-L_1\sin\theta_1$
μ_{17}	$L_1\cos\theta_1$
μ_{18}	$0.5L_1\cos\theta_1$

velocity junction representing the *CG* velocity of each link. The transformer moduli obtained from the kinematic analysis with the help of the D-H parameters shown in Table 13.5 have been used to draw the bond graph and they are listed in Table 13.6. Soft pads are artificial compliances which have been used in bond graph to avoid differential causality.

The tip velocities are indicated on the top of the bond graph model. Integrators are used to integrate the actual velocity components to obtain the tip positions in X and Y directions, i.e., $X_{\text{tip\_act}}$ and $Y_{\text{tip\_act}}$. The reference position of the end-effector tip is denoted by $X_{\text{tip\_ref}}$ and $Y_{\text{tip\_ref}}$. In Fig. 13.28, J_1, J_2, and J_3 represent the joint inertia, respectively, and the R element on each joint represents the joint resistance.

13.7.2 Simulation and Animation Results

Three types of end-effector trajectories have been considered to validate the control strategy. First trajectory is a circle, the second one is a knot-shaped trajectory based on Lissajous curve [15], and the third one is a cubic polynomial trajectory. The initial configuration of the 3-link planar manipulator is shown in Fig. 13.29. All the joints were kept at an initial position of 0.698 rad. Parameters used in the

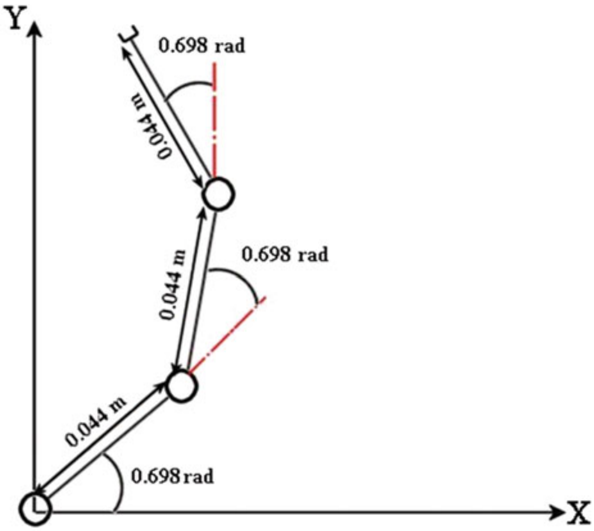


Fig. 13.29 Initial configuration of three link planar manipulator

Table 13.7 Simulation parameters for the three different trajectories

Parameters	Value	Parameters	Value
Joint resistance	$R_j = 0.001$ Nm/(rad/s)	Acceleration due to gravity	$g = 9.81$ m/s^2
Link length	$L = 0.044$ m	Mass of link	$m = 0.01$ kg
Joint inertia	$I_j = 0.001$ kg m^2	Proportional gain	$K_p = 0.5$
Amplitude	$A = 0.01$ m	Integral gain	$K_i = 0.01$
Frequency	$\omega = 10$ rad/s	Lissajous constants	$A = 1, B = 1,$ $\delta = \pi/2, a/b = 0.5$

simulation are enlisted in Table 13.7. The three different cases considered to check the efficiency of the proposed control scheme are discussed in this section.

13.7.2.1 Case I: Circular Reference Trajectory

Let us consider the reference trajectory for the 3-link planar manipulator tip in X and Y direction to be a circle of radius A. Then the tip coordinates are

$$X_{\text{tip\_ref}} = A\cos(\omega t) + X_0 \tag{13.51}$$

$$Y_{\text{tip\_ref}} = A\sin(\omega t) + Y_0 \tag{13.52}$$

where (X_0, Y_0) is the centre of the reference circular trajectory, A is the amplitude of circle, ω is the angular velocity, and t is the time in seconds. Hence the

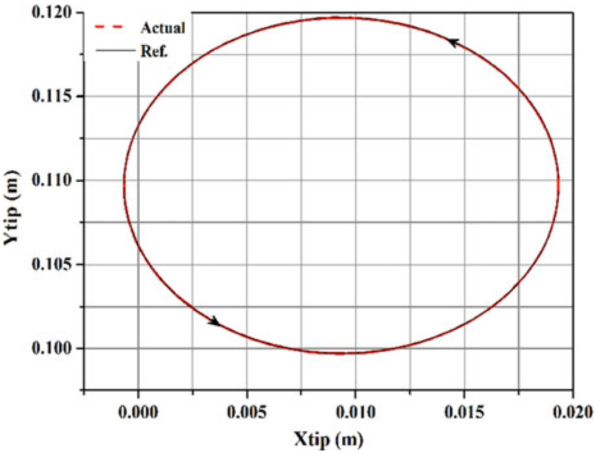


Fig. 13.30 Reference versus actual tip trajectory for a circular trajectory

corresponding reference velocity trajectory can be expressed as,

$$\dot{X}_{\text{tipref}} = -A\omega \sin(\omega t) \tag{13.53}$$

$$\dot{Y}_{\text{tip\_ref}} = A\omega \cos(\omega t) \tag{13.54}$$

Figure 13.30 shows a comparative trajectory plot between the reference and actual tip trajectory, and Fig. 13.31 indicates the X_{tip} and Y_{tip} positional error with respect to time for the circular trajectory for which simulation was run for 6 s.

The X_{tip} error was recorded between 0.0214 and –0.00912 mm, and the Y_{tip} error was between 0.043015 and –0.0048 mm. Figure 13.32 shows the animation result for the circular trajectory for the *case I*. The direction of rotation of the tip position is indicated in this figure.

13.7.2.2 Case II: Knot Shape Lissajous Curve as Reference Trajectory

Let us assume that the reference trajectory in X and Y directions is a knot-shape trajectory based on the Lissajous curve [15] and the tip coordinates are as follows:

$$X_{\text{tip\_ref}} = A\sin(at + \delta) \tag{13.55}$$

$$Y_{\text{tip\_ref}} = B\sin(bt) \tag{13.56}$$

where A and B are constants and *a/b* is a ratio to be maintained, i.e., $a/b = 0.5$ for this particular knot-shape, and the value of phase difference δ is taken as $\pi/2$. The corresponding reference velocity trajectory can be expressed as,

$$\dot{X}_{\text{tip\_ref}} = Aa\cos(at) \tag{13.57}$$

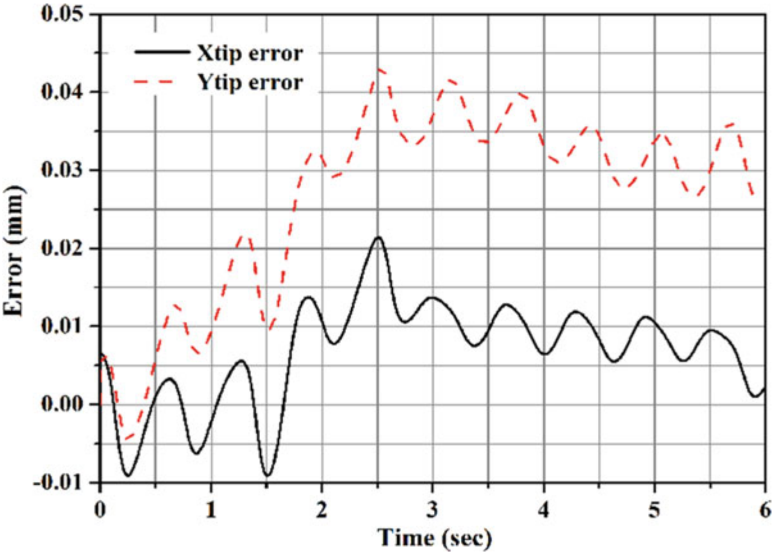
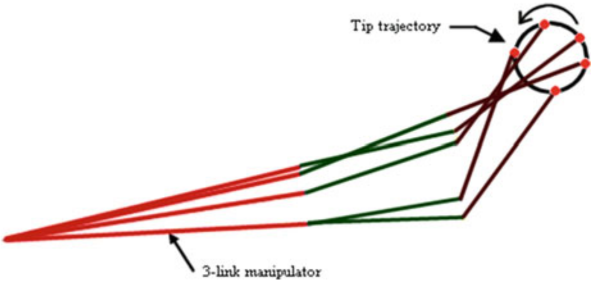


Fig. 13.31 X_{tip} and Y_{tip} error versus time for circular trajectory

Fig. 13.32 Animation result for circular trajectory



$$\dot{Y}_{\text{tip\_ref}} = Bb\cos(bt) \tag{13.58}$$

Figure 13.33 shows the comparative tip trajectory between the reference and actual tip trajectory. Figure 13.34 indicates X_{tip} and Y_{tip} positional error with respect to time, respectively. For the knot-shape Lissajous curve trajectory the simulation period was 6.4 s. The X_{tip} error was recorded between 0 and 0.027325 mm and the Y_{tip} error was between 0 and 0.058107 mm. The animation result for the knot-shape Lissajous curve trajectory along with the tip direction of motion is shown in Fig. 13.35.

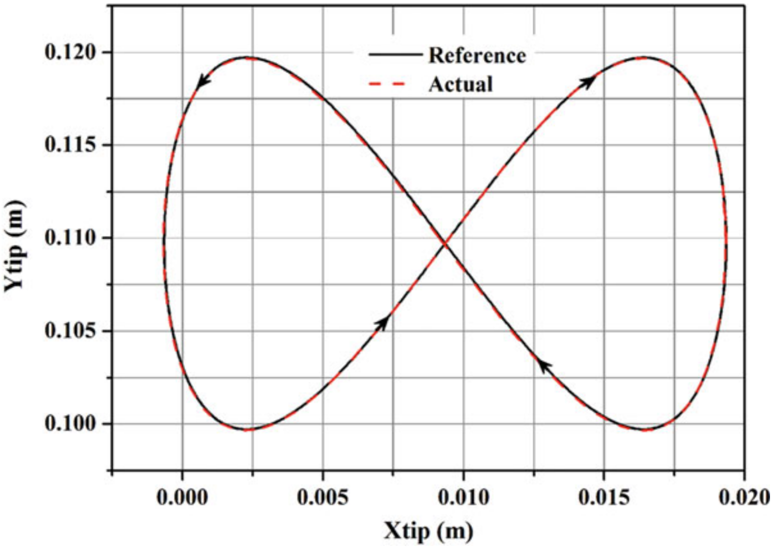


Fig. 13.33 Reference versus actual tip trajectory for the knot-shape Lissajous curve

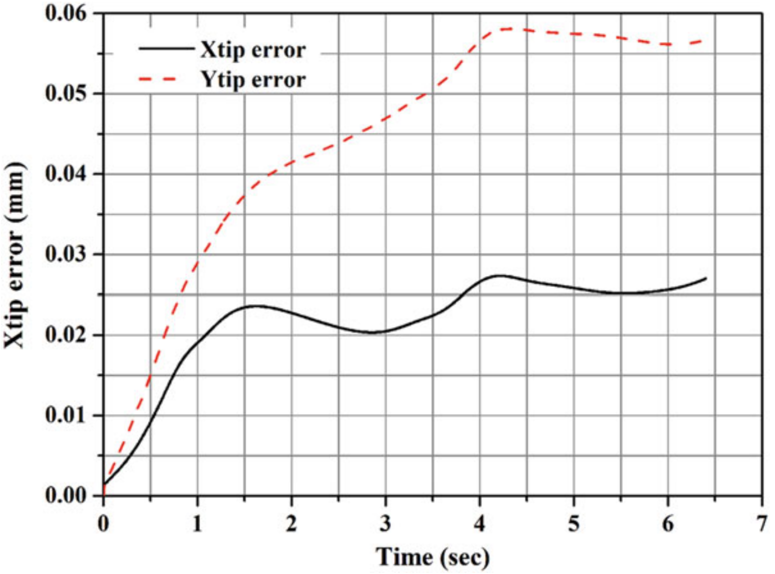


Fig. 13.34 X_{tip} and Y_{tip} error versus time for knot-shape Lissajous curve

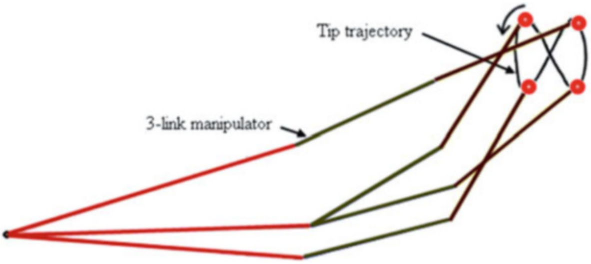


Fig. 13.35 Animation result for a knot-shaped Lissajous curve trajectory

13.7.2.3 Case III: Cubic Polynomial as Reference Trajectory

In the third case a cubic polynomial trajectory has been considered. Let us assume the reference trajectory in X and Y directions as follows:

$$X_{\text{tip\_ref}} = X_0 + 3\left(X_f - X_0\right)\left(t/t_f\right)^2 - 2\left(X_f - X_0\right)\left(t/t_f\right)^3 \tag{13.59}$$

$$Y_{\text{tip\_ref}} = Y_0 + 3\left(Y_f - Y_0\right)\left(t/t_f\right)^2 - 2\left(Y_f - Y_0\right)\left(t/t_f\right)^3 \tag{13.60}$$

where (X_f, X_0) are the final and initial tip coordinates in X direction and (Y_f, Y_0) are the final and initial tip coordinates in Y direction, t_f is the final time in sec and t is the simulation time in sec. The corresponding reference velocities can be represented as,

$$\dot{X}_{\text{tip\_ref}} = 6/t_f^2\left(X_f - X_0\right)\left(1 - t/t_f\right)t \tag{13.61}$$

$$\dot{Y}_{\text{tip\_ref}} = 6/t_f^2\left(Y_f - Y_0\right)\left(1 - t/t_f\right)t \tag{13.62}$$

Figure 13.36 shows the comparative tip trajectory between the reference and actual tip trajectory. Figure 13.37 indicates X_{tip} and Y_{tip} positional error with respect to time. For the cubic polynomial trajectory the simulation period was 1.5 s. The X_{tip} error was recorded between 0.00196 and –0.0069 mm and the Y_{tip} error was between 0.029 and –0.0002 mm. Figure 13.38 shows the animation result and the tip direction of motion for the polynomial trajectory.

The animation frames clearly indicate the positions taken by the manipulator links while tracking the desired trajectory. These animations have been performed from simulation data in Symbols Shakti [12], environment.

13.7.3 Virtual Link Based Controller for Developed In-Vivo Robot

The competence of the virtual link based controller was tested in three-dimensional Cartesian coordinate after getting satisfactory result from the planar case of a 3-link

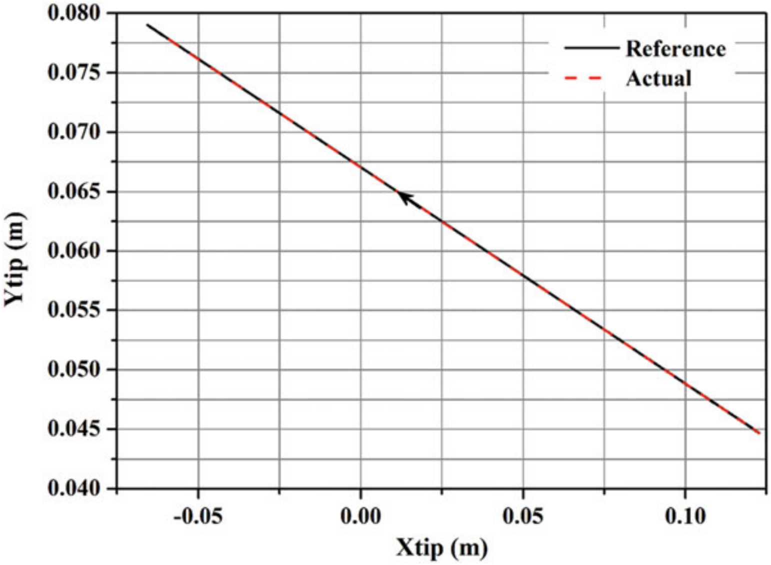


Fig. 13.36 Reference versus actual tip trajectory for cubic polynomial trajectory

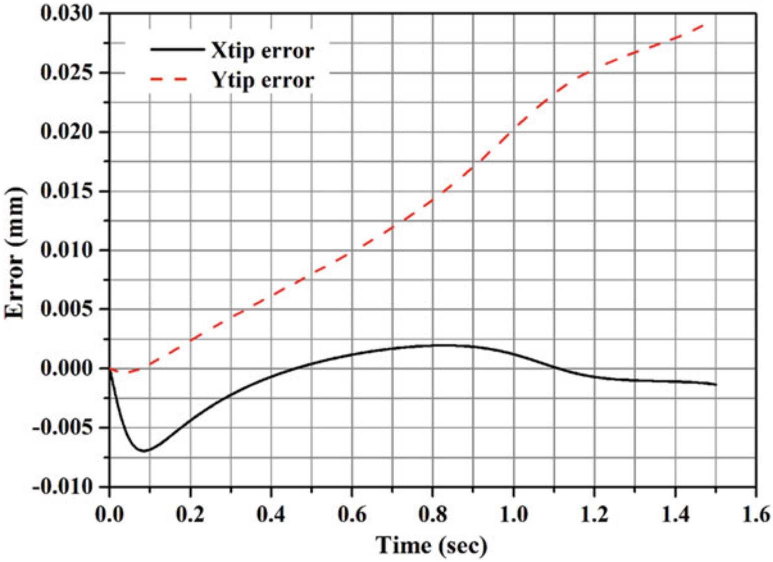


Fig. 13.37 X_{tip} and Y_{tip} error versus time for cubic polynomial trajectory

manipulator. As the bond graph model of the actual in-vivo robot has already explained earlier. In this section only the bond graph model of the virtual link based controller is explained. The bond graph model is shown in Fig. 13.39.

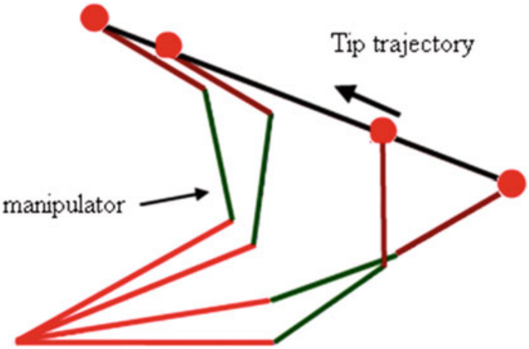


Fig. 13.38 Animation for a cubic polynomial trajectory

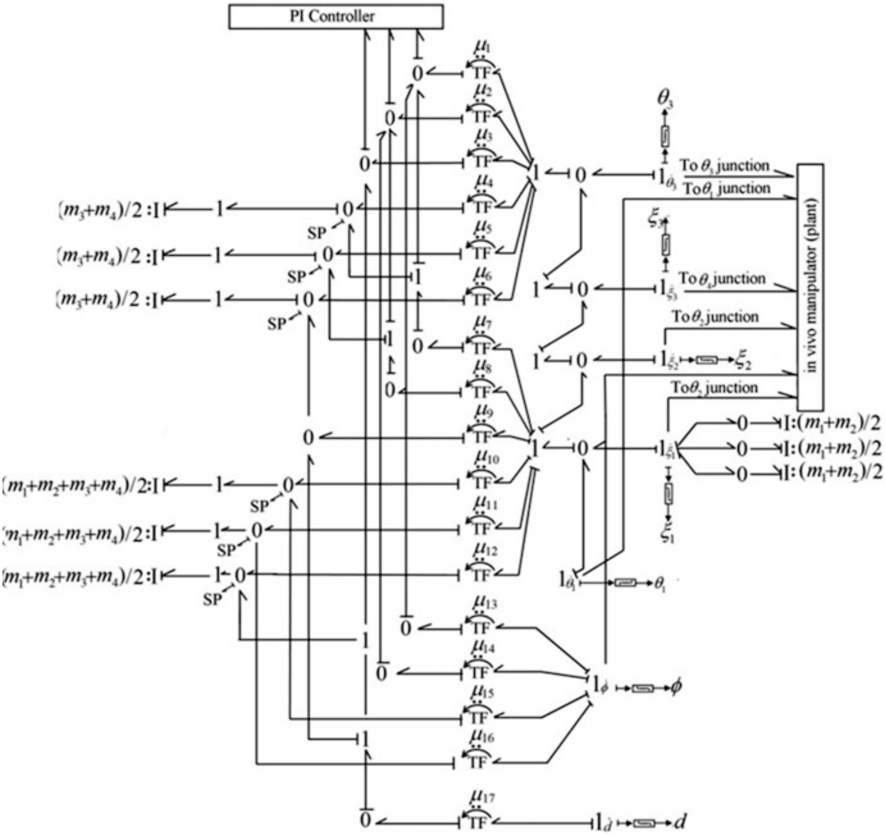


Fig. 13.39 Bond graph model of virtual link based controller of in-vivo robot for three-dimensional case

The virtual link lengths of the virtual link based controller used in case of three-dimensional trajectory control of in-vivo robot are derived following the same methodology adopted for the 3-link planar manipulator and can be expressed as:

$$L_{v_1} = \sqrt{L_1^2 + L_2^2 + (2L_1L_2\cos\theta_2)},\ \text{and}$$

$$L_{v_2} = \sqrt{L_3^2 + L_4^2 + (2L_3L_4\cos\theta_4)}$$

The transformer moduli are obtained using the same methodology that has been adopted in the planar case and they are enlisted in Table 13.8.

13.7.3.1 Simulation Results

The competence of the developed controller was further tested for three-dimensional workspace trajectory for which a cubic polynomial trajectory was chosen. Let us assume the reference trajectory in X and Y directions as follows:

$$X_{\text{tip\_ref}} = X_0 + 3\left(X_f - X_0\right)\left(t/t_f\right)^2 - 2\left(X_f - X_0\right)\left(t/t_f\right)^3 \tag{13.63}$$

$$Y_{\text{tip\_ref}} = Y_0 + 3\left(Y_f - Y_0\right)\left(t/t_f\right)^2 - 2\left(Y_f - Y_0\right)\left(t/t_f\right)^3 \tag{13.64}$$

Table 13.8 Transformer moduli for the bond graph model of virtual link based controller for three-dimensional case

Modulus	Expression
μ_1	$-L_{v_2}\cos\phi\sin(\theta_1 + \theta_3 + \zeta_1 + \zeta_2 + \zeta_3)$
μ_2	$-L_{v_2}\sin\phi\sin(\theta_1 + \theta_3 + \zeta_1 + \zeta_2 + \zeta_3)$
μ_3	$-L_{v_2}\cos(\theta_1 + \theta_3 + \zeta_1 + \zeta_2 + \zeta_3)$
μ_4	$-L_{v_2}\cos\phi\sin(\theta_1 + \theta_3 + \zeta_1 + \zeta_2 + \zeta_3)$
μ_5	$-L_{v_2}\sin\phi\sin(\theta_1 + \theta_3 + \zeta_1 + \zeta_2 + \zeta_3)$
μ_6	$-L_{v_2}\cos(\theta_1 + \theta_3 + \zeta_1 + \zeta_2 + \zeta_3)$
μ_7	$-L_{v_1}\cos\phi\sin\theta_1$
μ_8	$-L_{v_1}\sin\phi\sin\theta_1$
μ_9	$L_{v_1}\cos\theta_1$
μ_{10}	$-L_{v_1}\cos\phi\sin\theta_1$
μ_{11}	$-L_{v_1}\sin\phi\sin\theta_1$
μ_{12}	$L_{v_1}\cos\theta_1$
μ_{13}	$-\sin\phi\left\{L_{v_1}\cos(\theta_1 + \zeta_1) + L_{v_2}\cos(\theta_1 + \theta_3 + \zeta_1 + \zeta_2 + \zeta_3)\right\}$
μ_{14}	$\cos\phi\left\{L_{v_1}\cos(\theta_1 + \zeta_1) + L_{v_2}\cos(\theta_1 + \theta_3 + \zeta_1 + \zeta_2 + \zeta_3)\right\}$
μ_{15}	$-L_{v_1}\sin\phi\cos(\theta_1 + \zeta_1)$
μ_{16}	$L_{v_1}\cos\phi\cos(\theta_1 + \zeta_1)$

Table 13.9 Simulation parameters for three-dimensional case

Parameters	Value
Joint resistance	$R_j = 0.001$ Nm/(rad/s)
Link length	$L = 0.011$ m
Amplitude	$A = 0.01$ m
Frequency	$\omega = 10$ rad/s
Mass of link	$m = 0.001$ kg
Acceleration due to gravity	$g = 9.81$ m/s^2
Proportional gain	$K_p = 1000$
Integral gain	$K_i = 5000$
Initial position in X coordinate	0.0336 m
Final position in X coordinate	0.03275 m
Initial position in Y coordinate	0.02825 m
Final position in Y coordinate	0.02748 m
Initial position in Z coordinate	0.01291 m
Final position in Z coordinate	0.02047 m

$$Z_{\text{tip\_ref}} = Z_0 + 3\left(Z_f - Z_0\right)\left(t/t_f\right)^2 - 2\left(Z_f - Z_0\right)\left(t/t_f\right)^3 \tag{13.65}$$

where (X_f, X_0) are the final and initial tip coordinates in X direction and (Y_f, Y_0) are the final and initial tip coordinates in Y direction, (Z_f, Z_0) are the final and initial tip coordinates in Z direction, t_f is the final time in sec, and t is the simulation time in sec. The corresponding reference velocities can be represented as:

$$\dot{X}_{\text{tip\_ref}} = 6/t_f^2\left(X_f - X_0\right)\left(1 - t/t_f\right)t \tag{13.66}$$

$$\dot{Y}_{\text{tip\_ref}} = 6/t_f^2\left(Y_f - Y_0\right)\left(1 - t/t_f\right)t \tag{13.67}$$

$$\dot{Z}_{\text{tip\_ref}} = 6/t_f^2\left(Z_f - Z_0\right)\left(1 - t/t_f\right)t \tag{13.68}$$

The simulation parameters are expressed in Table 13.9.

The reference and actual tip trajectory is shown in Fig. 13.40 from which it can be seen that the robot tip precisely followed the reference path assigned to it from one known position to another in the three-dimensional Cartesian coordinate.

13.8 Concluding Remarks

In this chapter a forward kinematic analysis of the in-vivo robot manipulator was performed. The results obtained from the simulation validate the bond graph model. Further an inverse kinematic solution was found to determine the pulley rotation for different known position of the tip by taking three different trajectories. In fact this is

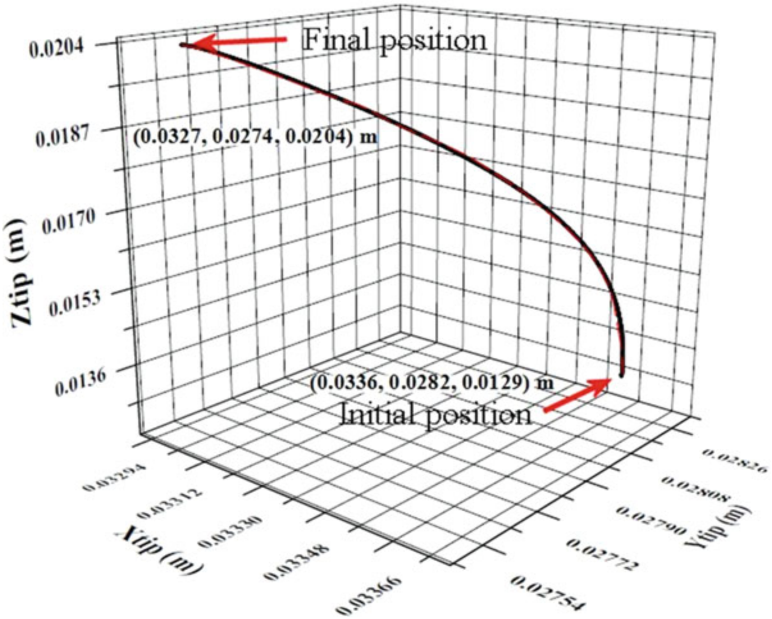


Fig. 13.40 Reference versus actual tip trajectory for three-dimensional cubic polynomial trajectory in workspace

the actual condition faced by surgeons where they would be interested in taking the robot tip from one position to the desired position. This inverse kinematics solution was validated using simulation results. Further experiments have been carried out to validate the results obtained in simulation and comparisons between simulated and experimental results are satisfactory. A very small deviation between simulated and experimental results of *X* and *Z* coordinates for tip position in three different cases were observed.

Acknowledgments The authors render their sincere gratitude to the Department of Science and Technology of Government of India to sponsor this research (Scheme Grant Code SR/S3/MERC/048/2009).

References

1. Rentschler, M. E., Farritor, S. M., & Iagnemma, K. (2007). Mechanical design of robotic in vivo wheeled mobility. *Journal of Mechanical Design, 129*, 1037–1045.
2. Kim, V. B., Chapman, W. H. H., Albrecht, R. J., Bailey, B. M., Young, J. A., Nifong, L. W., et al. (2002). Early experience with tele manipulative robot-assisted laparoscopic cholecystectomy using da Vinci. *Surgical Laparoscopy, Endoscopy & Percutaneous Techniques, 12*(1), 33–44.

3. Webster, R. J., III, & Jones, B. A. (2010). Design and kinematic modeling of constant curvature continuum robots: A review. *The International Journal of Robotics Research, 29*(13), 1661–1683.
4. Wang, J., Li, Y., & Zhao, X. (2010). Inverse kinematics and control of a 7-DOF redundant manipulator based on the closed-loop algorithm. *International Journal of Advanced Robotic Systems, 7*(4), 1–9.
5. Chirikjian, J. S., & Burdick, J. W. (1991). Kinematics of hyper-redundant robot locomotion with application to grasping. *Proceedings of the IEEE International Conference on Robotics and Automation* (Vol. 1, pp. 720–725), Sacramento, California, 9–11 April.
6. Raibert, M., & Craig, J. (1981). Hybrid position/force control of manipulators. *Transactions of ASME Journal of Dynamic Systems, Measurement and Control, 102*(2), 126–133.
7. Merzouki, R., Samantaray, A. K., Pathak, P. M., & Ould Boumama, B. (2013). *Intelligent mechatronic systems: Modeling, control and diagnosis*. London: Springer.
8. Hartenberg, R. S., & Denavit, J. (1955). A kinematic notation for lower pair mechanisms based on matrices. *Journal of Applied Mechanics, ASME Transactions, 23*, 215–221.
9. Craig, J. J. (2005). *Introduction to robotics mechanics and control*. Upper Saddle, NJ: Pearson Prentice Hall.
10. Borutzky, W. (2010). *Bond graph methodology—Development and analysis of multidisciplinary dynamic system models*. London: Springer.
11. Karnopp, D. C., Margolis, D. L., & Rosenberg, R. C. (2012). *System dynamics: Modelling and simulation of mechatronic systems* (4th ed.). Hoboken, NJ: Wiley.
12. Mukherjee, A., Karmakar, R., & Samantaray, A. K. (2006). *Bond graph in modeling, simulation and fault identification* (2nd ed.). New Delhi: I. K. International Publishing House Pvt. Ltd/CRC Press.
13. Sutar, M. K., Pathak, P. M., Sharma, A. K., Mehta, N. K., & Gupta, V. K. (2013). Forward kinematic analysis of in-vivo robot for stomach biopsy. *Journal of Robotic Surgery, 7*, 281–287.
14. Bevan, T. (2010). *The theory of machines*. New Delhi: Pearson Education.
15. French, A. P. (1971). *Vibrations and waves*. London: CRC Press.

Chapter 14
Bond graph Modeling and Control of Compliant Legged Quadruped Robot

M.M. Gor, P.M. Pathak, A.K. Samantaray, J.M. Yang, and S.W. Kwak

Abbreviations

AVP	Angular velocity propagation
CG	Center of gravity
CTF	Coordinate transformation block
DOF	Degree of freedom
EJS	Euler Junction Structure
LL	Leg lift
PL	Prismatic link
PI	Proportion-integral controller
PID	Proportional-integral-derivative controller

M.M. Gor
G H Patel College of Engineering and Technology, Vallabh Vidyanagar, Gujarat, India
e-mail: mehulmgor@gmail.com

P.M. Pathak (✉)
Indian Institute of Technology Roorkee, Roorkee, Uttarakhand, India
e-mail: pushpfme@iitr.ac.in; pushppathak@gmail.com

A.K. Samantaray
Indian Institute of Technology Kharagpur, Kharagpur, West Bengal, India
e-mail: samantaray@mech.iitkgp.ernet.in

J.M. Yang
Kyungpook National University, Daegu, South Korea
e-mail: jmyang@ee.knu.ac.kr

S.W. Kwak
Keimyung University, Daegu, South Korea
e-mail: ksw@kmu.ac.kr

W. Borutzky (ed.), *Bond Graphs for Modelling, Control and Fault Diagnosis of Engineering Systems*, DOI 10.1007/978-3-319-47434-2_14

RP	Revolute-prismatic
SL	Step length
TVP	Translational velocity propagation
ZMP	Zero moment point

14.1 Historical Survey of Modeling and Control of Quadruped Robot

Legged robot offers many advantages over wheeled robots including greater adaptability to terrain irregularities and superior off-road mobility [30, 41]. Legged systems require only a series of discrete footholds along the pathway for off-road locomotion. This property enables legged robots to traverse surfaces inaccessible to wheeled mobile robots. Compliance in the leg improves locomotion of legged robot [40]. Variable compliance in the legs [16] overcomes the size, weight, fragility, and efficiency problem. Basically, legged robots are discrete systems in which joints of each leg have to operate in particular fashion. So, dynamics plays an important role in the operation and control of a walking robot. Recently, there has been a noteworthy increase in the use of computational dynamics for design, analysis, simulation, and control of various robotic systems. This is due to availability of various multi-body dynamic analysis tools and faster computational resources. To this end, various researchers used different dynamic analysis methods for multi-body systems, such as the methods based on Lagrangian equation [23], Newton–Euler equation [3, 43], Kane's equation [2, 5, 33], variational methods [4], and bond graph method [26].

Benani and Giri [6] presented a dynamic model approach of quadruped considering open and/or closed kinematic chain mechanisms. It is based on Newton–Euler approach and the explicit formulation of kinematic holonomic constraints for the closed loop mechanism. Mahapatra and Roy [27] developed a dynamic model of six legged in CATIA solid modeler, SimDesigner, and ADAMS multi-body dynamic solver and kinematic and dynamic simulation is performed based on virtual prototyping technology. Krishnan et al. [26] presented a bond graph model of compliant legged quadruped robot in a sagittal plane. The sagittal plane dynamics have been tested through experimental set-up. Soyguder and Ali [39] solved the stance and flight phase dynamic structures in a sequential closed loop for quadruped and obtained the equation of motion for pronking gait. Shah et al. [36] presented a concept of kinematic modules for the development of the dynamic model of the four legged robots where each module is considered as a set of serially connected links. Module-level Decoupled Natural Orthogonal Complement (DeNOC) matrices were introduced which help to analyze the large number of links as a system with a smaller number of modules. Recursive kinematic relationships were obtained between two adjoining modules. Ganesh and Pathak [17] developed a dynamic model of four legged in a sagittal plane by formulating kinetic and potential energy equation of body and leg. These were used to derive Lagrangian function and then equation of motion. A locomotion control strategy for quadruped robot has been presented in [20]. Dynamic modeling and analysis of quadruped robot through bond graph technique has been presented in [19].

Many of the quadruped robots developed worldwide are biologically inspired. Process of natural selection governing evolution of species forces animals adapt to their specific physical features and environment by optimizing their locomotion. Some animals are better at doing certain things in comparison to others. Thus, keeping the required task or operation features in the view, more and more specialized biologically inspired quadrupeds are being developed now-a-days. Some of them are Baby Elephant [11], BigDog [31], Cheetah-cub [40], HyQ [35], LittleDog [10, 37], and Tekken [14, 25]. Hydraulically actuated Baby Elephant [11] was designed to work as mechanical carrier. It has 12 DOFs and compliant legs. Multi-body dynamic simulation was used in [11] for its design and the results were experimentally validated. BigDog [31] was developed with the goal to move in rough terrain without human assistance. It has 20 DOFs and about 50 sensors. Four joints of each leg are operated by hydraulic actuator. It uses a two-stroke internal combustion engine that delivers up to 15 hp power. Electrically actuated Cheetah-cub [40] was designed for high-speed locomotion. Cheetah-cub's legs are spring loaded and pantograph mechanism with multiple segments is used for shock absorption during running. This robot's self-stabilizing properties were demonstrated in hardware model and in simulation carried out in Webots software. HyQ [35] developed at IIT Genova was designed to perform highly dynamic tasks like jumping and running. It has 12 DOFs and both hydraulic and electrical actuation systems. During running and jumping, generated impact forces were absorbed by hydraulic actuation mounted on hip and knee joints in the flexion/extension plane of the leg. The hip abduction/adduction joint was actuated by brushless electric motor which provides constant output torque. LittleDog [10, 37] has 12 DOFs and each joint is operated by a high-gain servo motor. Sensors mounted on the robot measure body orientation, joint angles, and ground-foot contact. Sensing, communication, and actuators are controlled by onboard PC-level computer. Tekken [14, 25] is a light weight (4.3 kg) manually operated power autonomous compliant legged quadruped robot. It has 16 DOFs, three joints around pitch axis (ankle, knee, and hip), and one hip joint around yaw axis at each leg. At ETH Zurich, two quadrupeds have been developed having similar structure, size, and morphology, but different concept of actuation [32]. The first, ALoF, is a classically stiff actuated robot that is controlled kinematically; whereas the second, StarlETH, uses a soft actuation scheme based on highly compliant series elastic actuators.

In this chapter, three dimensional dynamic model of compliant legged quadruped robot using bond graph has been developed. A quadruped robot configuration used for analysis is two links legged robot in which upper link is rigid and a lower link is compliant. Lower link is considered similar to a prismatic link in which, piston and piston rod is sliding inside the cylinder and movement is restricted by the spring which generates compliance in the leg. The strategy for locomotion control in joint space as well as workspace is discussed. To validate the same, simulation and animation of the trot gait performed by the quadruped is carried out which is further verified by experiment results. Since trot gait is dynamically stable gait thus successful validation of model in trot gait ensures the model validity in other gaits also. To prove the versatility of the three dimensional model generated a turning motion of the robot is demonstrated by varying the leg speed in amble gait. Influence of compliance on quadruped locomotion and posture disturbance is studied. Performance analysis is carried out considering energy efficiency as deciding criteria. Performance analysis on rigid and compliant legged robots, static and dynamic gaits, and foot trajectory is carried out. Nomenclature used throughout this chapter is listed below.

Nomenclature

$\{A\}$	Inertial frame
$\{B\}$	Robot body frame
c_i, s_i	$\cos(\theta_i)$, $\sin(\theta_i)$
D	Width of a piston of a prismatic link
F	Force
F_{xc}, F_{yc}, F_{zc}	External force acting at the cylinder body fixed x, y, and z axes
H	Angular momentum
I	Link number, frame number
I_{rot}	Rotor inertia
I_{xb}, I_{yb}, I_{zb}	Moment of inertia of the robot body about x, y, and z axes
I_{xx1}, I_{yy1}, I_{zz1}	Moment of inertia of the upper link of robot leg about x, y, and z axes
I_{xxc}, I_{yyc}, I_{zzc}	Moment of inertia of the cylinder part of a prismatic link about x, y, and z axes
I_{xxp}, I_{yyp}, I_{zzp}	Moment of inertia of the piston part of a prismatic link about x, y, and z axes
k_b	Contact point stiffness at the piston cylinder of prismatic link
k_f	Flexibility due to hydraulic pressure inside the cylinder of prismatic link
K_{gx}, K_{gy}, K_{gz}	Ground contact stiffness in x, y, and z direction
K_P, $K_{I,}K_D$	Proportional, integral, and derivative gain of controller
K_{pb}, K_{ib}, K_{db}	Proportional, integral, and derivative gain of controller for body forward
l_c	Contemporary length of prismatic link
l_{cg}	Distance of cylinder CG from a cylinder end frame of prismatic link
l_i	Length of link i
l_p	Length of a piston and piston rod of prismatic link
l_{pg}	Distance of piston CG from a piston end frame of prismatic link
L_m	Motor inductance
m_b	Mass of the body
m_c	Mass of cylinder part of the prismatic link
m_{li}	Mass of link i
m_p	Mass of a piston and piston rod of a prismatic link
M_{xc}, M_{yc}, M_{zc}	External moment acting at the cylinder body fixed x, y, and z axes
n	Gear ratio
p	Translational momentum
${}_{B}^{A}R$	Transformation from body frame $\{B\}$ to inertial frame $\{A\}$

R_b	Contact point resistance at the piston cylinder of prismatic link
R_f	Damping between piston and cylinder of prismatic link
R_{gx}, R_{gy}, R_{gz}	Ground contact resistance in x, y, and z direction
r_{ix}, r_{iy}, r_{iz}	Position of the frame {0} of ith leg with respect to the body CG
R_m	Motor resistance
t	Time
v_x, v_y, v_z	Translational velocities of the body
θ_i	Angular displacement of frame i
τ	Torque
ψ, θ, ϕ	Euler angles representing a robot body rotation about x, y, z axis of the body fixed frame
ψ_c, θ_c, ϕ_c,	Cardan angles about x, y, z axis of the moving fixed frame
ω	Angular velocity
ω_{xc}, ω_{yc}, ω_{zc}	Angular velocities of the mass center of the cylinder in the body fixed frame

14.2 Modeling of a Quadruped Robot

Modeling of a quadruped robot consists of modeling of angular and translational dynamics of robot body and legs. Figure 14.1a shows physical model of quadruped robot, while Fig. 14.1b shows the schematic diagram of a quadruped robot model in which {A} is an inertial frame and {B} is the body frame attached to body center of gravity (CG).

Frame {0} is fixed at the hip joint of each leg which is fixed on the robot body. Each leg of the quadruped robot has two degree of freedom (DOF) with two revolute

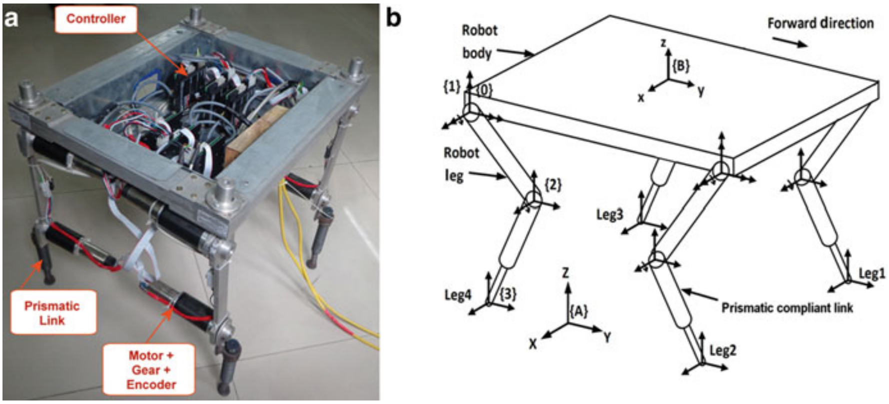


Fig. 14.1 (**a**) Physical model of quadruped robot [21]. (**b**) Schematic representation of quadruped robot with compliant legs [19, 21]

joints per leg. The joint between links i and $i+1$ is numbered as $i+1$. A coordinate frame $\{i+1\}$ is attached to $(i+1)$ joint. Frame $\{1\}$ is attached to joint 1 of each leg. Frame $\{0\}$ is coinciding with frame $\{1\}$. Frame $\{2\}$ is attached to joint $\{2\}$, while frame $\{3\}$ is attached to leg tip. The rotational inertias are defined about frames fixed at the CG of the link. The CG frame is fixed along the principal directions in the link or body. The surface on which the robot is walking is assumed as a hard surface.

14.2.1 Dynamics of a Robot Body

For a given instant, any rigid body has absolute translational velocity $\overrightarrow{v}$ and absolute angular velocity $\overrightarrow{\omega}$. The translational velocity and angular velocity vectors have been resolved into three mutually perpendicular components v_x, v_y, v_z and ω_x, ω_y, ω_z. The net force $\overrightarrow{F}$ acting on the body and the linear momentum $\overrightarrow{p}$ can be related with respect to rotating frame as [24],

$$\overrightarrow{F} = \left(\frac{\mathrm{d}\overrightarrow{p}}{\mathrm{d}t}\right)_{\mathrm{rel}} + \overrightarrow{\omega} \times \overrightarrow{p} \tag{14.1}$$

where $\left(\frac{\mathrm{d}\overrightarrow{p}}{\mathrm{d}t}\right)_{\mathrm{rel}}$ is the rate of change of momentum relative to the moving frame. Similarly, the relationship between the net torque $\overrightarrow{\tau}$ acting on the body and the angular momentum $\overrightarrow{h}$ can be written as,

$$\overrightarrow{\tau} = \left(\frac{\mathrm{d}\overrightarrow{h}}{\mathrm{d}t}\right)_{\mathrm{rel}} + \overrightarrow{\omega} \times \overrightarrow{h} \tag{14.2}$$

Using the right-hand rule for Eqs. (14.1) and (14.2), the component equations can be written as,

$$F_x = m_b\dot{v}_x + m_b\omega_y v_z - m_b\omega_z v_y \tag{14.3}$$

$$F_y = m_b\dot{v}_y + m_b\omega_z v_x - m_b\omega_x v_z \tag{14.4}$$

$$F_z = m_b\dot{v}_z + m_b\omega_x v_y - m_b\omega_y v_x \tag{14.5}$$

and

$$\tau_x = I_{xx}\dot{\omega}_x + \big(I_{zz} - I_{yy}\big)\,\omega_y\omega_z \tag{14.6}$$

$$\tau_y = I_{yy}\dot{\omega}_Y + (I_{xx} - I_{zz})\,\omega_z\omega_x \tag{14.7}$$

$$\tau_z = I_{zz}\dot{\omega}_z + \left(I_{yy} - I_{xx}\right)\omega_x\omega_y \tag{14.8}$$

These nonlinear differential equations are known as Newton–Euler‘s equations [28]. The cross product terms can be treated as forces in a set of Eqs. (14.3), (14.4), and (14.5) and as torques in a set of Eqs. (14.6), (14.7), and (14.8). The forces and torques can be added at the respective 1-junctions and using gyrator-ring structures bond graph is generated as presented in [28]. Generated structure is known as Euler Junction Structures (EJS).

14.2.2 Dynamics of an Upper Link of Leg

Translational velocity of frame {0} of each leg with reference to frame $\{A\}$ and expressed in term of frame $\{A\}$ is given by [12],

$${}^{A}\left({}^{A}\overrightarrow{V}_0\right) = {}^{A}\left({}^{A}\overrightarrow{V}_B\right) + {}_{B}{}^{A}R\left[-{}^{B}\left({}^{B}\overrightarrow{P}_0\right) \times {}^{B}\left({}^{A}\overrightarrow{\omega}_B\right)\right] \tag{14.9}$$

where ${}^{A}\left({}^{A}\overrightarrow{V}_B\right)$ represents the translational velocity of body frame $\{B\}$ with respect to an inertial frame $\{A\}$ and expressed in frame $\{A\}$; ${}^{B}\left({}^{A}\overrightarrow{\omega}_B\right)$ represents the angular velocity of body frame $\{B\}$ with respect to inertial frame $\{A\}$ and expressed in frame $\{B\}$; ${}^{B}\left({}^{B}\overrightarrow{P}_0\right)_i$ represents the position vector of frame {0} of ith leg with respect to the body CG frame $\{B\}$ and expressed in frame $\{B\}$. It can be expressed as ${}^{B}\left({}^{B}\overrightarrow{P}_0\right)_i = \left[\, r_{ix}\ r_{iy}\ r_{iz} \,\right]^T$ where i denotes legs 1–4. Here r denotes position of frame {0} with respect to body CG frame. In Eq. (14.9), ${}_{B}^{A}R$ represent the transformation from body frame $\{B\}$ to inertial frame $\{A\}$ and can be expressed as,

$${}_{B}^{A}R = \begin{pmatrix} c\theta\ c\phi & s\psi\ s\theta\ c\phi - c\psi\ s\phi & c\psi\ s\theta\ c\phi + s\psi\ s\phi \\ c\theta\ s\phi & s\psi\ s\theta\ s\phi + c\psi\ c\phi & c\psi\ s\theta\ s\phi - s\psi\ c\phi \\ -\ s\theta & s\psi\ c\theta & c\psi\ c\theta \end{pmatrix} \tag{14.10}$$

where $c\theta$ is shorthand for cosθ, $s\theta$ for sinθ, and so on. ϕ, θ and ψ are the Z-Y-X Euler angles. Governing equation for an angular velocity propagation (AVP) of links of a leg can be given as [12],

$${}^{i+1}\left({}^{A}\overrightarrow{\omega}_{i+1}\right) =_{i}^{i+1}R^{i}\left({}^{A}\overrightarrow{\omega}_i\right) + {}^{i+1}\left({}^{i}\overrightarrow{\omega}_{i+1}\right) \tag{14.11}$$

where ${}^{i+1}\left({}^{A}\overrightarrow{\omega}_{i+1}\right)$ is the angular velocity of $(i+1)$ link with respect to inertial frame $\{A\}$ and expressed in $(i+1)$th frame, ${}^{i}\left({}^{A}\overrightarrow{\omega}_i\right)$ is the angular velocity of the *ith* link with respect to the inertial frame $\{A\}$ and expressed in *ith* frame, and

${}^{i+1}\left({}^{i}\overrightarrow{\omega}_{i+1}\right)$ is the angular velocity of $(i+1)$ link as observed from ith link and expressed in $(i+1)$th frame. The term can be expressed for link 1 and 2, respectively as, ${}^{1}\left({}^{0}\overrightarrow{\omega}_{1}\right) = \left[\dot{\theta}_1\ 0\ 0\right]^T$, ${}^{2}\left({}^{1}\overrightarrow{\omega}_{2}\right) = \left[\dot{\theta}_2\ 0\ 0\right]^T$ where $\dot{\theta}_1$ represents angular velocity of frame $\{1\}$ with respect to frame $\{0\}$ expressed in frame $\{1\}$ and similarly $\dot{\theta}_2$ represents the angular velocity of frame $\{2\}$ with respect to frame $\{1\}$ expressed in frame $\{2\}$.

For translational velocity propagation (TVP), governing equation for the link tip velocity and link CG velocity are given as,

$$ {}^{A}\left({}^{A}\overrightarrow{V}_{i+1}\right) = {}^{A}\left({}^{A}\overrightarrow{V}_{i}\right) +{}_{i}^{A}R\left[{}^{i}\left({}^{A}\overrightarrow{\omega}_{i}\right) \times {}^{i}\left({}^{i}\overrightarrow{P}_{i+1}\right)\right] \tag{14.12} $$

where ${}^{A}\left({}^{A}\overrightarrow{V}_{i+1}\right)$ represents the translational velocity of body frame $\{i+1\}$ with respect to an inertial frame $\{A\}$ and expressed in frame $\{A\}$, ${}^{A}\left({}^{A}\overrightarrow{V}_{i}\right)$ represents the translational velocity of body frame $\{i\}$ with respect to an inertial frame $\{A\}$ and expressed in frame $\{A\}$, ${}^{i}\left({}^{i}\overrightarrow{P}_{i+1}\right)$ represents position of frame $\{i+1\}$ with respect to frame $\{i\}$ and expressed in frame $\{i\}$. Link lengths l_1 and l_2 are taken along the principal Y-axis of the links and hence represented in vector form as, ${}^{0}\overrightarrow{P}_1 = \left[0\ 0\ 0\right]^T$, ${}^{1}\overrightarrow{P}_2 = \left[0\ l_1\ 0\right]^T$, ${}^{2}\overrightarrow{P}_3 = \left[0\ l_2\ 0\right]^T$

Equation (14.12) can be simplified as,

$$ {}^{A}\left({}^{A}\overrightarrow{V}_{i+1}\right) = {}^{A}\left({}^{A}\overrightarrow{V}_{i}\right) +{}_{i}^{A}R\left[-{}^{i}\left({}^{i}\overrightarrow{P}_{i+1}\right)\times\right]\left[{}^{i}\left({}^{A}\overrightarrow{\omega_{i}}\right)\right] \tag{14.13} $$

For position of a link CG, ${}^{i}\left({}^{i}\overrightarrow{P}_{Gi}\right) = \left[0\ l_{Gi}\ 0\right]^T$

$$ {}^{A}\left({}^{A}\overrightarrow{V}_{Gi}\right) = {}^{A}\left({}^{A}\overrightarrow{V}_{i}\right) +{}_{i}^{A}R\left[-{}^{i}\left({}^{i}\overrightarrow{P}_{Gi}\right)\times\right]\left[{}^{i}\left({}^{A}\overrightarrow{\omega_{i}}\right)\right] \tag{14.14} $$

Equations (14.12), (14.13), and (14.14) represent the TVP of link in each leg of the robot. The CG velocity of links depends on link inertia. In the bond graph model "I" elements (representing mass of a link) are attached at flow junctions. They yield the CG velocities of links. The starting point of the current link is same as the previous link tip. Hence, the tip velocity of the previous link and the angular velocity of the current link are used to find the tip velocity and the CG velocity of the current link. ${}^{i}\left({}^{A}\overrightarrow{\omega_{i}}\right)$ in above equations can be obtained from the AVP for the current link.

The EJS to represent angular dynamics of the link can be constructed similarly as discussed in Sect. 14.2.1. In case of link, torque is provided in x direction only.

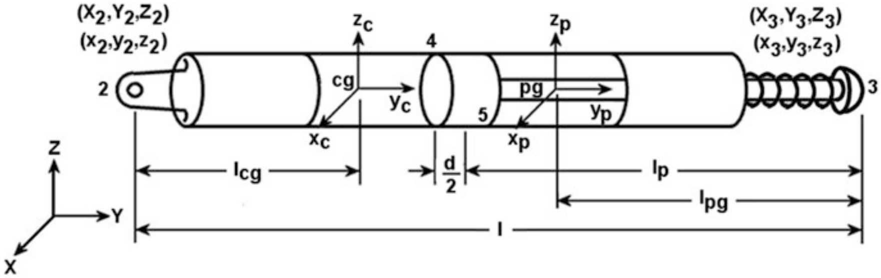


Fig. 14.2 Schematic diagram of prismatic link [7, 9]

14.2.3 Dynamics of the Prismatic Link

Lower link of the quadruped leg can be considered as a prismatic link, in which piston and piston rod is sliding inside cylinder. The movement is restricted by the internal hydraulic pressure of the cylinder and helical compression spring attached at the piston rod. This arrangement generates compliance in the leg. The sliding of piston is one of the most difficult multi-body components which give rise to nonlinear equations of motion. It is important to develop bond graph model of prismatic link with proper mass distribution. Incorrect modeling of entire prismatic link generates improper inertial forces. Thus, utmost care should be taken at the time of generating bond graph model of the three dimensional prismatic link. Bond graph modeling of prismatic link is developed from the concept presented in [7–9, 13, 28]. The schematic drawing of prismatic link is shown in Fig. 14.2.

Local coordinate frame is attached at the center of mass of piston (x_p, y_p, z_p) and cylinder (x_c, y_c, z_c) and they are assumed to be aligned with the inertial principal axes. Piston and cylinder motions are described with reference to this body fixed coordinate system which rotate and translate with the respective rigid bodies. The end of the cylinder part will be fixed with the link 1 of quadruped robot and piston end will touch the ground. X_2, Y_2, Z_2 and X_3, Y_3, Z_3 are the inertial coordinate system while x_2, y_2, z_2 and x_3, y_3, z_3 are body fixed or non-inertial coordinate system of the cylinder and piston end, respectively. The contemporary length l_c is the distance between the two end points. The center of gravity of the cylinder is located at a distance of l_{cg} from the fixed end. The combined center of gravity of the piston and the rod is located at a distance of l_{pg} from the rod end. The length of the piston is d. The center of the piston is located at a distance l_p from the rod end.

The velocity vector of the cylinder in the inertial frame is represented as $\overrightarrow{v}_{Xc,Yc,Zc}$ and in the body fixed frame as $\overrightarrow{v}_{xc,yc,zc}$. The angular velocity vector in the body fixed frame is $\overrightarrow{\omega}_{xc,yc,zc}$. Then, Euler equation for translatory motion of the cylinder can be given as,

$$F_{xc} = M_c\ddot{x}_c + M_c\left(\dot{z}_c\omega_{yc} - \dot{y}_c\omega_{zc}\right) \tag{14.15}$$

$$F_{yc} = M_c \ddot{y}_c + M_c \left(\dot{x}_c \omega_{zc} - \dot{z}_c \omega_{xc} \right) \tag{14.16}$$

$$F_{zc} = M_c \ddot{z}_c + M_c \left(\dot{y}_c \omega_{xc} - \dot{x}_c \omega_{yc} \right) \tag{14.17}$$

where F_{xc}, F_{yc}, F_{zc} are external forces acting in body fixed x_c, y_c, and z_c directions, respectively, ω_{xc}, ω_{yc}, and ω_{zc} are angular velocities of the mass center of the cylinder in the body fixed frame. $\dot{x}_c, \dot{y}_c$ and $\dot{z}_c$ are velocities of the mass center in the body fixed frame, $\ddot{x}_c, \ddot{y}_c$ and $\ddot{z}_c$ are accelerations of the mass center in the body fixed frame. Similarly, Euler equations for rotary motion of the cylinder can be given as,

$$M_{xc} = I_{xc} \dot{\omega}_{xc} - \left(I_{yc} - I_{zc} \right) \omega_{yc} \omega_{zc} \tag{14.18}$$

$$M_{yc} = I_{yc} \dot{\omega}_{yc} - \left(I_{zc} - I_{xc} \right) \omega_{zc} \omega_{xc} \tag{14.19}$$

$$M_{zc} = I_{zc} \dot{\omega}_{zc} - \left(I_{xc} - I_{yc} \right) \omega_{xc} \omega_{yc} \tag{14.20}$$

where I_{xc}, I_{yc}, and I_{zc} are second moment of inertia about the principal axes, M_{xc}, M_{yc}, and M_{zc} are components of resultant moment due to external forces and couples about the non-rotating coordinate frame whose axes are momentarily aligned with the principal axes of the body.

Above Euler equations can be represented by the double gyrator rings where the gyrators are modulated by the angular velocities in the body fixed frame. Similarly, Euler equations can be generated for the piston also and it can be represented by the double gyrator rings. Generated bond graph model of above equations are shown in Fig. 14.3 in which G1–G12 are the gyrator moduli taken from above equations.

The position of the fixed point in the body fixed frame is x_2, y_2, z_2. The velocity of the cylinder in the body fixed frame is

$$\dot{x}_2 = \dot{x}_c + z_2 \omega_{yc} - y_2 \omega_{zc} \tag{14.21}$$

$$\dot{y}_2 = \dot{y}_c + x_2 \omega_{zc} - z_2 \omega_{xc} \tag{14.22}$$

$$\dot{z}_2 = \dot{z}_c + y_2 \omega_{xc} - x_2 \omega_{yc} \tag{14.23}$$

Equations (14.21), (14.22), and (14.23) are body fixed velocities. So, it is necessary to convert it into the inertial frame by coordinate transformation block (CTF). CTF block is generated using successive multiplication of rotation matrices as follows:

$$\begin{Bmatrix} \dot{X}_2 \\ \dot{Y}_2 \\ \dot{Z}_2 \end{Bmatrix} = T_{\phi c, \theta c, \psi c} \begin{Bmatrix} \dot{x}_2 \\ \dot{y}_2 \\ \dot{z}_2 \end{Bmatrix} \tag{14.24}$$

where

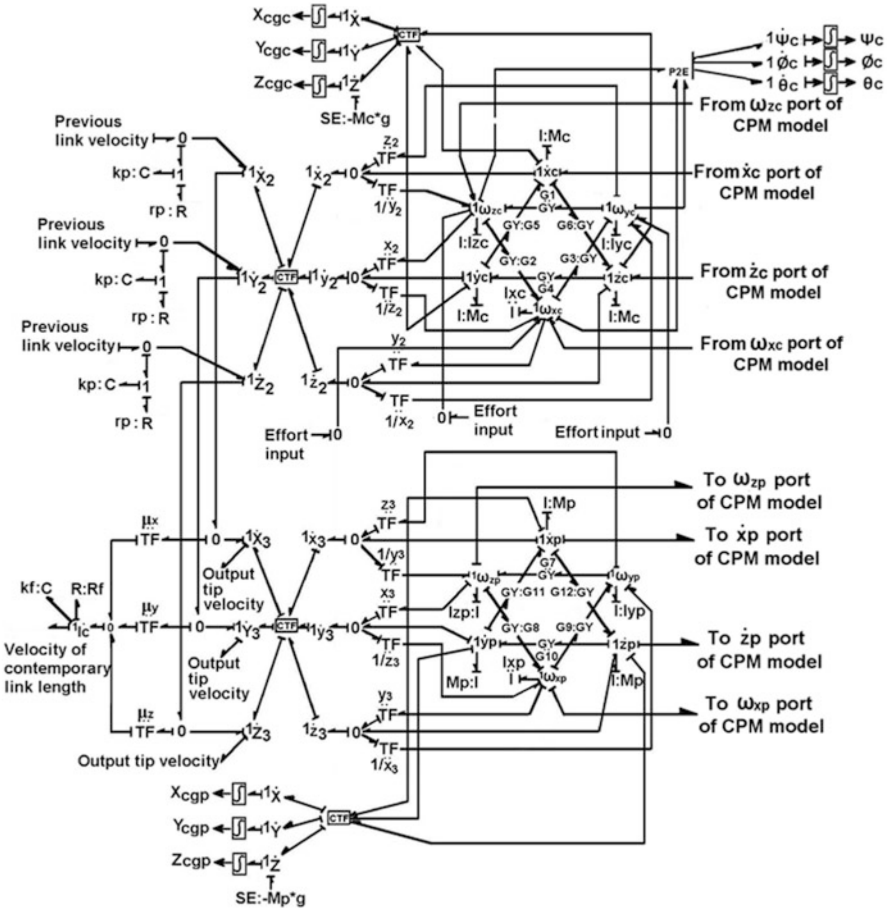


Fig. 14.3 Bond graph model sub-model of prismatic link (PL)

$$
T_{\phi c,\theta c,\psi c} = \begin{bmatrix} c\phi_c & -s\phi_c & 0 \\ s\phi_c & c\phi_c & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta_c & 0 & s\theta_c \\ 0 & 1 & 0 \\ -s\theta_c & 0 & c\theta_c \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\psi_c & -s\psi_c \\ 0 & s\psi_c & c\psi_c \end{bmatrix} \tag{14.25}
$$

$c\theta_c$ is shorthand for $\cos\theta_c$, $s\theta_c$ for $\sin\theta_c$, and so on and ϕ_c, θ_c and ψ_c are the *Z-Y-X* Cardan angles. Components of $T_{\phi c,\theta c,\psi c}$ are used to construct CTF block. Similarly, the velocity of the piston in the body fixed frame can be written and it can be converted into an inertial frame. It is to be noted that the required angle for CTF block is derived from the inverse transformation from body fixed angular velocities to Euler angle rates [28].

The normal fixed velocities at the contact point 4 and 5 on the cylinder and piston along x and z directions by assuming a thin but long piston can be given as [7]

$$\dot{x}_{4c} = \dot{x}_c + \left(l - l_p - l_{cg} - \frac{d}{2}\right)\omega_{zc} \tag{14.26}$$

$$\dot{z}_{4c} = \dot{z}_c + \left(l - l_p - l_{cg} - \frac{d}{2}\right)\omega_{xc} \tag{14.27}$$

$$\dot{x}_{4p} = \dot{x}_p - \left(l_p - l_{pg} + \frac{d}{2}\right)\omega_{zp} \tag{14.28}$$

$$\dot{z}_{4p} = \dot{z}_p - \left(l_p - l_{pg} + \frac{d}{2}\right)\omega_{xp} \tag{14.29}$$

$$\dot{x}_{5c} = \dot{x}_c + \left(l - l_p - l_{cg} + \frac{d}{2}\right)\omega_{zc} \tag{14.30}$$

$$\dot{z}_{5c} = \dot{z}_c + \left(l - l_p - l_{cg} + \frac{d}{2}\right)\omega_{xc} \tag{14.31}$$

$$\dot{x}_{5p} = \dot{x}_p - \left(l_p - l_{pg} - \frac{d}{2}\right)\omega_{zp} \tag{14.32}$$

$$\dot{z}_{5p} = \dot{z}_p - \left(l_p - l_{pg} - \frac{d}{2}\right)\omega_{xp} \tag{14.33}$$

where $\overrightarrow{\omega}$ indicates the body fixed angular velocity about the axis indicated in subscript. Subscript c has been used for cylinder while p has been used for the piston. Above discussed contact point mechanism is model as shown in Fig. 14.4.

The rate of change of contemporary length between two end points of prismatic link can be expressed as

$$\dot{l}_c = \mu_x\left(\dot{X}_2' - \dot{X}_3'\right) + \mu_y\left(\dot{Y}_2' - \dot{Y}_3'\right) + \mu_z\left(\dot{Z}_2' - \dot{Z}_3'\right) \tag{14.34}$$

where X_2', Y_2', Z_2' and X_3', Y_3', Z_3' are coordinates of cylinder end and piston end, respectively, in the inertial frame, $\mu_x = \frac{(X_2' - X_3')}{l_c}$, $\mu_y = \frac{(Y_2' - Y_3')}{l_c}$, and $\mu_z = \frac{(Z_2' - Z_3')}{l_c}$ are moduli used to derive the relative sliding velocity between piston and cylinder at "0" junction. Compliance in the link is modeled by "C" and "R" element. For contact point mechanics, to compute contact point velocities in body fixed frame moduli β_1–β_4 and β_5–β_8 are determined from kinematic analysis of the cylinder and piston, respectively. Through a set of transformer moduli μ_1–μ_{12} similar to an expanded form of CTF block, body fixed velocities are transformed into inertial velocities and then they are implicitly constrained. The relative normal velocity between the contact point on the cylinder and the normal velocity at the contact point on the piston is implicitly constrained by contact stiffness and damping parameters,

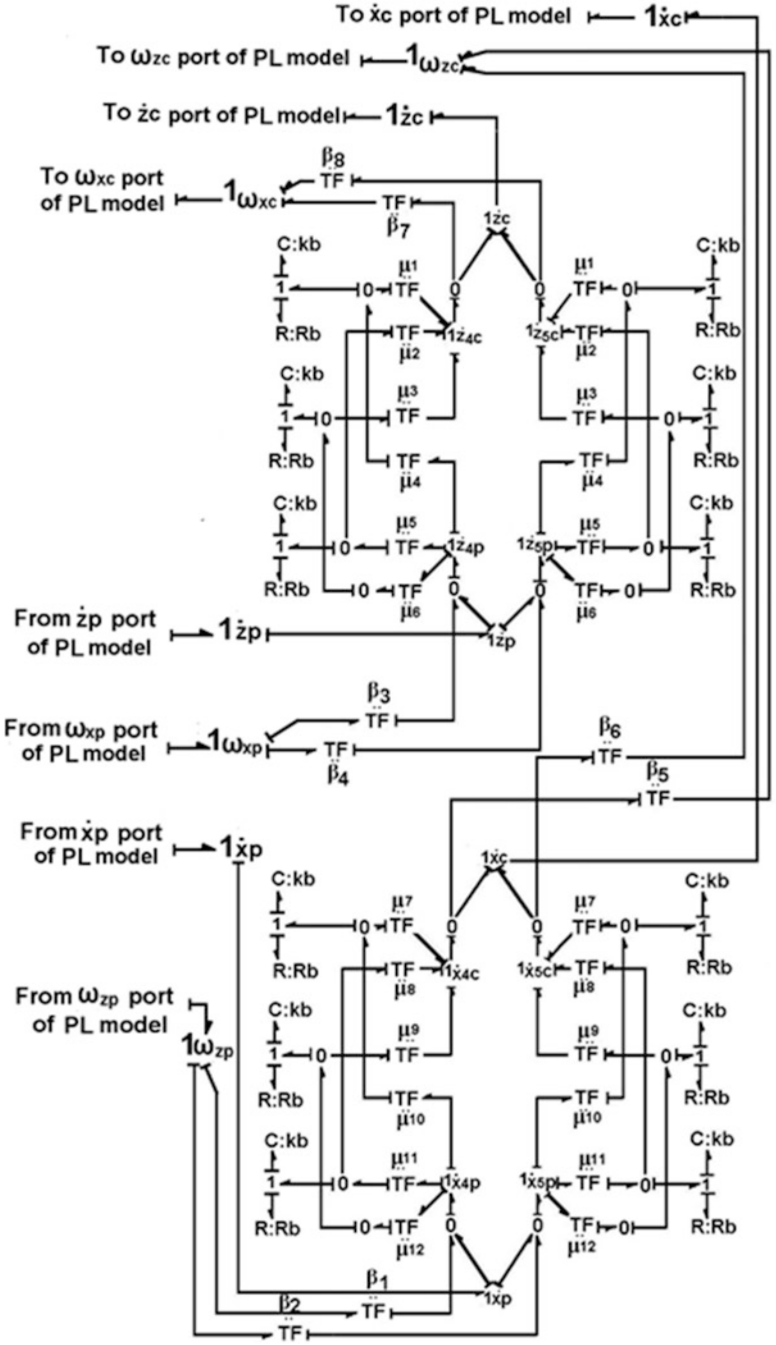


Fig. 14.4 Bond graph sub-model of contact of point mechanism (CPM) for prismatic link

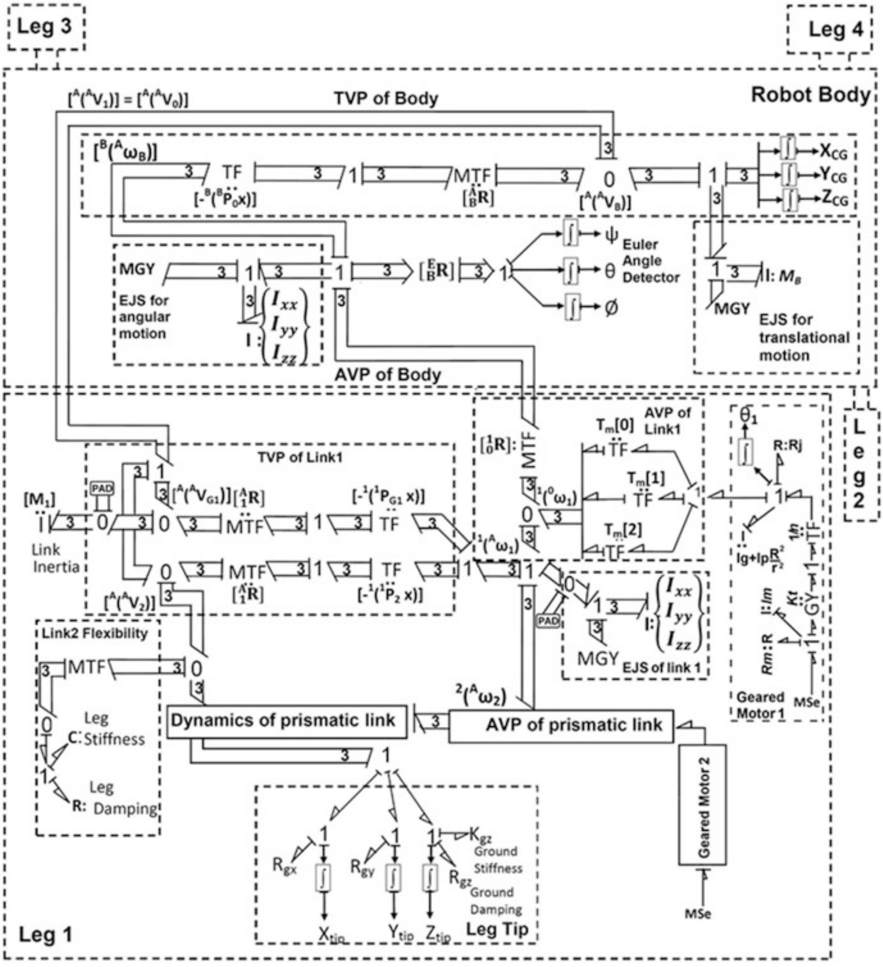


Fig. 14.5 Multibond graph of a quadruped robot

k_b and R_b, respectively. The three external effort inputs are from the motor. Since the motor applies torque about body fixed x-axis on the cylinder, two of these effort inputs are zero.

14.2.4 Dynamics of Combined Body and Leg Links

The bond graph model is developed using above discussed body and leg dynamics. A compact and a simple presentation of a bond graph model can be carried out in multibond graph form. Here also developed three dimensional model of quadruped robot is presented in multibond graph form as shown in Fig. 14.5. TVP sub-model of the "Body" part is shown in the multibond graph. It takes the angular velocity from a body ${}^{B}\left({}^{A}\overrightarrow{\omega}_{B}\right)$ (obtained from EJS) and translational velocity ${}^{A}\left({}^{A}\overrightarrow{V}_{B}\right)$ (decided by body mass) as input and gives out the velocity of {0} frame to the link 1 of each leg. Frames {0} and {1} are coincident for each leg. Hence, the velocity of frame {1} is same as the frame {0}. "Leg" sub-model in the multibond graph represents a two DOF leg. It takes the angular and linear velocity of body and joint torques about x-axis as input. It uses AVP and TVP sub-models of links 1 and 2 and gives out leg tip velocity as output. This sub-model furnishes complete dynamics of a two link leg. The various sub-models shown in Fig. 14.5 for leg 1 can also be used to model legs 2, 3, and 4. The leg tip sub-model in Fig. 14.5 represents the modeling of leg tip-ground interaction. An R element is appended to "1" junction of each leg in the X and Y direction, to model the frictional resistance offered by ground. Similarly, C and R elements are attached in Z direction to model the normal reaction force from the ground. Leg tip position sensors in each direction yield the leg tip position coordinates.

A systematic construction of bond graph model yields a dynamics expression that can be written in matrix form as

$$\overrightarrow{\tau}_i = \left[A(\theta)\,\ddot{\theta} + B\left(\theta, \dot{\theta}\right) + C(\theta)\right]_i - J_i^T \overrightarrow{F}_i \tag{14.35}$$

where $\overrightarrow{\tau}$ is the 2×1 matrix of joint torque and $\overrightarrow{F}$ is the 3×1 vector of the ground contact force of leg i, J is the Jacobian matrix, $A(\theta)$ is the 3×3 mass matrix, B is a 3×1 matrix of centrifugal and Coriolis terms, and $C(\theta)$ is a 3×1 matrix of gravity terms. A pad is used to avoid differential causality. Pads are artificial compliances/lumped flexibilities that can be used in bond graph [18, 29]. Bond graph model of above discussed body and leg dynamics is developed in SYMBOLS software [34].

Compliance in the link improves locomotion of quadruped robot. But over compliance reduces locomotion speed and also affects posture disturbance. So, most suitable value of compliance is must for specific robot configuration. This objective can be achieved by simulating bond graph model of quadruped. Number of simulations can be carried out by varying the compliance (which is discussed in coming section), and its values can be finalized for maximum locomotion speed of robot. This stiffness and load coming on each leg becomes the key parameters for designing the spring used in prismatic link. Number of turns of spring n can be decided as

$$n = \frac{\delta G d}{8 W C^3} \tag{14.36}$$

where W is load, δ is axial deflection derived from load(W)/stiffness(K), C is spring index derived from coil diameter(D)/wire diameter(d), and G is the Modulus of rigidity of the spring material. Here, W is the dynamic load estimated from simulations and the spring design is valid only when the inertial effects of the spring are neglected, i.e., when the rigid-body acceleration is not very large.

14.3 Locomotion Control of Quadruped Robot

A quadruped robot can be considered as a multi-arm industrial manipulator with a mobile base. As a consequence, its control is considerably difficult as compared to that of a terrestrial manipulator. Also, each leg has to pass through stand and motion phases depending upon the gait pattern. In such a situation, the locomotion control is a difficult problem which demands accurate kinematic and dynamic analyses. Walk can be classified as “static walk” and “dynamic walk.” In static walk stability is maintained by keeping at least three feet planted on the ground and maintaining the center of gravity within the support of polygon. In dynamic walk stability is maintained by continuously moving either the feet or the body to maintain balance. Alexander [1] shows various gait patterns followed by four legged animals. Here statically stable gait “Amble” and dynamically stable gait “Trot” are considered. In case of trot gait one pair of diagonal legs moves forward while other pair remains on the ground, which reverses in next phase of locomotion. In case of statically stable amble gait, at a time only one leg moves forward while remaining legs maintain contact with the ground. Here, legs are operated one by one in 1-4-2-3 sequence. Both the gaits are shown in Fig. 14.6, where dark line indicates corresponding leg contact with the ground.

Quadruped locomotion can be controlled either in joint space or in workspace. In joint space control commands are directly given for specific joint rotation, while in workspace control, based on the need of feet trajectory joints movements are derived and commands are given to actuators for the same.

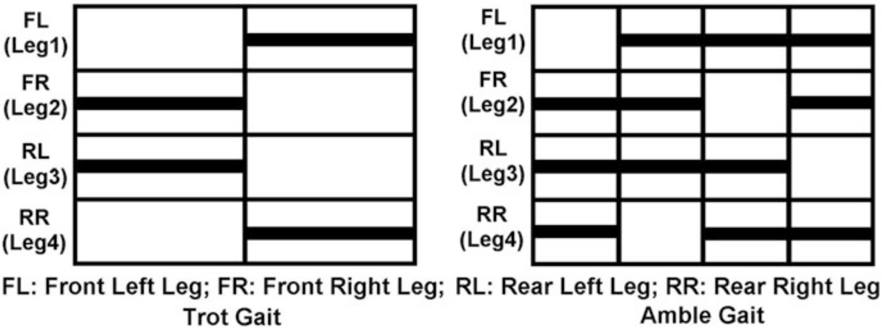


Fig. 14.6 Trot and amble gait

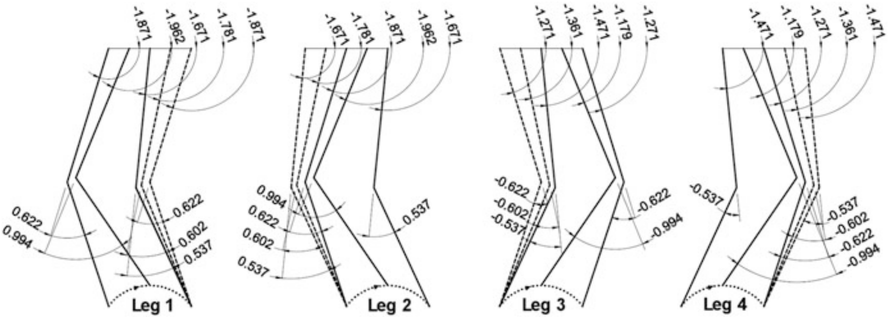


Fig. 14.7 Graphical representation of leg and body movement

Table 14.1 Joint position for trot gait

Joint position corresponding to time interval	Leg 1		Leg 2		Leg 3		Leg 4	
	Joint 1	Joint 2	Joint 1	Joint 2	Joint 1	Joint 2	Joint 1	Joint 2
Initial joint angle	−1.871	0.622	−1.671	0.537	−1.271	−0.622	−1.471	−0.537
$T_0 \leq t < T_1/2$	−1.962	0.944	−1.781	0.602	−1.361	−0.602	−1.179	−0.944
$T_1/2 \leq t < T_1$	−1.671	0.537	−1.871	0.622	−1.471	−0.537	−1.271	−0.622
$T_1 \leq t < T_2/2$	−1.781	0.602	−1.962	0.944	−1.179	−0.944	−1.361	−0.602
$T_2/2 \leq t < T_2$	−1.871	0.622	−1.671	0.537	−1.271	−0.622	−1.471	−0.537

14.3.1 Joint Space Control

In joint space control, commands are directly given to the joint based on the required gait pattern. To simulate the bond graph model in any of the gait, it is necessary to know rotation required at each joint so that a required voltage can be supplied to get the desired rotation of a joint. Thus, to determine a joint rotation, for leg forward and body forward movement, graphical analysis is carried out. Graphical analysis as shown in Fig. 14.7 gives an idea about a joint rotation required for the said movement.

From Fig. 14.7, joint position with reference to time is listed in Table 14.1 for trot gait. Here, cubic curve is fitted for smooth joint rotation. The required voltage for the said movement is supplied by actuator which is controlled by Proportional-Integral-Derivative (PID) controller and can be represented as,

$$V = K_P\left(\theta_d - \theta_a\right) + K_D\left(\dot{\theta}_d - \dot{\theta}_a\right) + K_I\int\left(\theta_d - \theta\right)\mathrm{d}t \tag{14.37}$$

where V is the input voltage supplied to the joint actuator of leg; K_P, K_D, and K_I are the proportional, derivative, and integral gains, respectively; θ_d is the desired position, θ_a is the actual measured angular position, $\dot{\theta}_d$ is the desired joint velocity, and $\dot{\theta}_a$ is the actual measured joint velocity.

14.3.2 Workspace Control

To control the quadruped locomotion in the workspace two distinct control steps are executed in sequence as per the selected gait pattern:

1. Leg forward motion control where the desired leg tip trajectory in the forward direction is obtained by fitting a cubic curve in the path to be followed and the leg vertical motion in Z direction is obtained by fitting two cubic curves. The trajectory error is fed to a PI controller whose output is then transformed through Jacobian to generate the corrective joint torques.
2. Body forward motion control where the leg tips remain fixed on the ground and the joints are actuated in order to align leg segments to their initial positions. Here, the required torque is directly supplied to the joints by the actuators, which are controlled by PID controller.

14.3.2.1 Reference Tip Velocity Generation

In this control strategy authors assumed joint motions such that leg tip has horizontal and vertical translation. The entire trajectory generation process consisting of various steps has been explained in Fig. 14.8.

If the robot joints are powered in the sequence as discussed in Sect. 14.3.1, the leg tip position for leg 1 in Y and Z directions as obtained from a dynamic model turns out to be as shown in Fig. 14.9. However, such actuation of joints is a joint space control and the problem it is that the complicated trajectories with constraints, such as obstacles, cannot be achieved easily due to nonlinear kinematics of the system. If similar kind of motion is required through workspace control, tip trajectory should be taken as input.

Here, cubic curve equation is used for fitting the leg tip movement in Y direction shown in Fig. 14.9.

$$y(t) = y_1 + \frac{3\,(y_2 - y_1)}{tf^2}t^2 + \frac{-2\,(y_2 - y_1)}{tf^3}t^3 \tag{14.38}$$

where y_1 is initial position of leg tip, y_2 is final position of leg tip, and t_f is trajectory end time. In generation of trajectory given by Eq. (14.38), it is assumed that at $t = 0; y(t) = y_1, \dot{y}(t) = 0$ and at $t = t_f; y(t) = y_2, \dot{y}(t) = 0$. For Z direction,

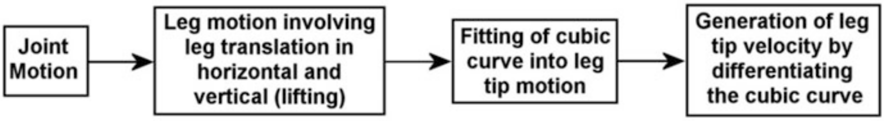

Joint Motion
Leg motion involving leg translation in horizontal and vertical (lifting)
Fitting of cubic curve into leg tip motion
Generation of leg tip velocity by differentiating the cubic curve

Fig. 14.8 Leg tip trajectory generation

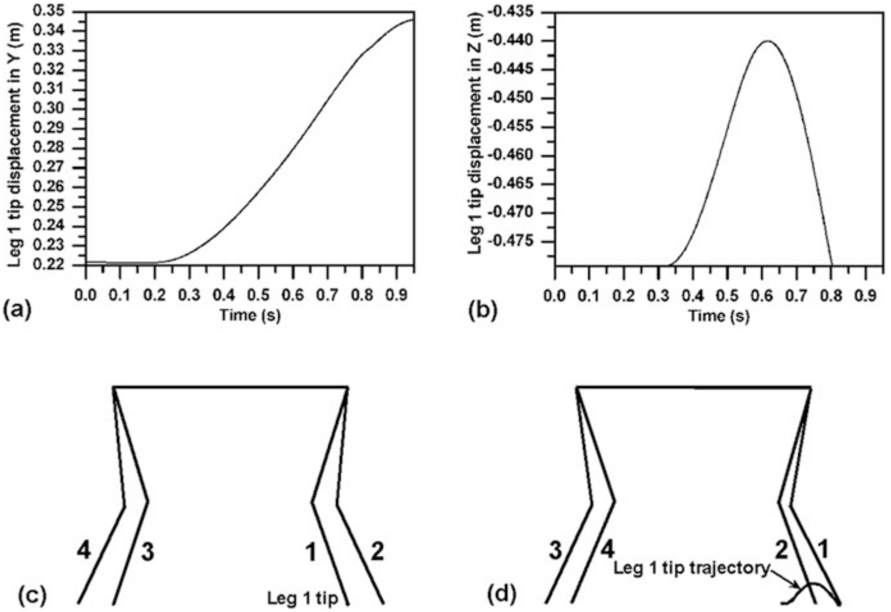


Fig. 14.9 Leg 1 tip displacement in (**a**) *Y* and (**b**) *Z* directions; (**c**) initial position of legs and (**d**) position of the legs at the end of first phase

the trajectory is divided into two segments. For the smooth motion of quadruped, the reference tip velocity and its derivative should be continuous. There are several ways in which desired velocities at via points can be specified [12]. For *Z* direction, cubic polynomials are fitted with constraints that the accelerations at via points have to be continuous. The first and second cubic interpolation polynomials are

$$z_1(t) = z_i + \frac{(12z_v - 3z_f - 9z_i)}{4{t_{f1}}^2}t^2 + \frac{(-8z_v + 3z_f + 5z_i)}{4{t_{f1}}^3}t^3 \tag{14.39}$$

$$z_2(t) = z_v + \frac{(3z_f - 3z_i)}{4t_{f2}}t + \frac{(-12z_v + 6z_f + 6z_i)}{4{t_{f2}}^2}t^2 + \frac{(8z_v - 5z_f - 3z_i)}{4{t_{f2}}^3}t^3 \tag{14.40}$$

where z_i is initial position, z_f is final position, z_v is via point, and $t = t_{f1}$ is trajectory end time for first segment and $t = t_{f2}$ is trajectory end time for second segment. In generation of trajectories given by (14.39) and (14.40), it is assumed that

$$\text{at } t = 0; z_1(t) = z_i, \dot{z}_1(t) = 0,$$

$$\text{at } t = t_{f1}; z_1(t) = z_v, z_2(t) = z_v, \dot{z}_1(t) = \dot{z}_2(t), \ddot{z}_1(t) = \ddot{z}_2(t),$$

$$\text{and at } t = t_{f2}; z_2(t) = z_f, \dot{z}_2(t) = 0.$$

$$\dot{y}(t) = \frac{6\left(y_2 - y_1\right)}{{t_f}^2}t + \frac{-6\left(y_2 - y_1\right)}{{t_f}^3}t^2 \tag{14.41}$$

$$\dot{z}_1(t) = \frac{2\left(12z_v - 3z_f - 9z_i\right)}{4{t_{f1}}^2}t + \frac{3\left(-8z_v + 3z_f + 5z_i\right)}{4{t_{f1}}^3}t^2 \tag{14.42}$$

$$\dot{z}_2(t) = \frac{\left(3z_f - 3z_i\right)}{4t_{f2}} + \frac{2\left(-12z_v + 6z_f + 6z_i\right)}{4{t_{f2}}^2}t + \frac{3\left(8z_v - 5z_f - 3z_i\right)}{4{t_{f2}}^3}t^2 \tag{14.43}$$

Use of Eqs. (14.38), (14.39), (14.40), (14.41), (14.42), and (14.43) with required gait patterns yields the reference profile for leg as shown in Figs. 14.10 and 14.11. Thus, Fig. 14.9a, b has been faithfully mathematically represented by Fig. 14.10a, b. Similarly, reference profile for remaining legs can also be obtained. These reference profiles are used to operate the model in workspace. The velocities are fed in the bond graph model by using SF elements.

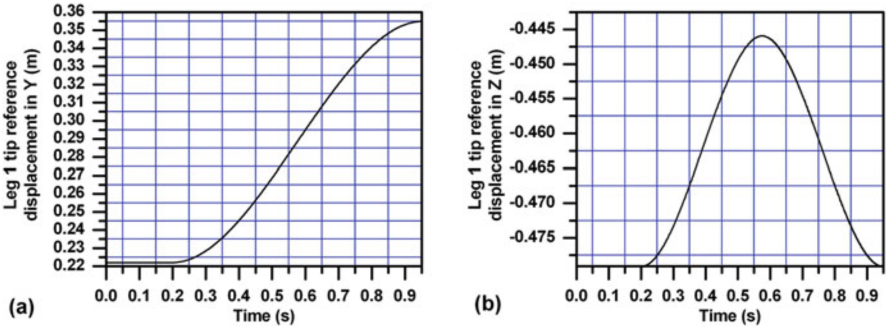


Fig. 14.10 Displacement profile through cubic polynomial: (**a**) displacement in *Y*, (b) displacement in *Z* directions

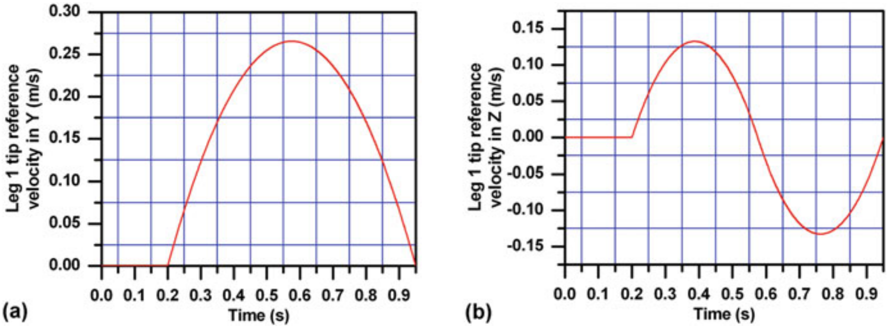


Fig. 14.11 Trajectory through cubic polynomial: (**a**) velocity in *Y* and (**b**) velocity in *Z* directions

14.3.2.2 Proportional Integral Controller

A PI controller is used here to achieve the leg forward movement according to a desired trajectory profile. Here, the joint input torque is controlled by calculating an error value as the difference between the measured and reference velocities at a leg tip. The PI controller gives a control signal gives a control signal

$$V_1 = K_P e + K_I \int e\mathrm{d}t \tag{14.44}$$

where V_1 is the output from the controller when there is an error e which is changing with time t, K_P is the proportionality constant, and K_I is the integral constant.

14.3.2.3 Jacobian

Jacobian is a mapping from velocities in joint space to Cartesian space. The inverse problem, where the joint velocities are to be determined for a given tip velocity, requires the inverse of the Jacobian. For quadruped robot working in workspace, one needs to evaluate the Jacobian for joint control through a controller. Considering compliance in the lower links of the leg and velocity propagation, the velocity of the leg tip ${}^{A}\left({}^{A}\overrightarrow{V}_3\right)$ represents the translational velocity of frame $\{3\}$ with respect to the inertial frame $\{A\}$ and expressed in frame $\{A\}$. ${}^{A}\left({}^{A}\overrightarrow{V}_3\right)$ can be written as,

$${}^{A}\left({}^{A}\overrightarrow{V}_3\right) = {}^{A}\left({}^{A}\overrightarrow{V}_2\right) + {}_{2}^{A}R\left[-{}^{2}\left({}^{2}\overrightarrow{P}_3\right)\times\right]\left[{}^{2}\left({}^{A}\overrightarrow{\omega}_2\right)\right] + {}_{2}^{A}R\dot{d}_3{}^{3}\widehat{Z}_3 \tag{14.45}$$

where ${}^{A}\left({}^{A}\overrightarrow{V}_2\right)$ is the translational velocity of frame $\{2\}$ with respect to inertial frame $\{A\}$ and expressed in frame $\{A\}$, ${}_{2}^{A}R$ is the rotation matrix which describes $\{2\}$ relative to $\{A\}$, ${}^{2}\left({}^{2}\overrightarrow{P}_3\right)$ is the position vector of frame $\{3\}$ with respect to frame $\{2\}$ and expressed in frame $\{2\}$, and ${}^{2}\left({}^{A}\overrightarrow{\omega}_2\right)$ is the angular velocity of link 2 with respect to the inertial frame $\{A\}$ and expressed in frame $\{2\}$.

Using Eq. (14.45), leg tip velocity of each leg can be derived; the expanded form of this equation is shown in (14.46).

$$
\begin{matrix}\dot{X}_{i\mathrm{tip}}\\ \dot{Y}_{i\mathrm{tip}}\\ \dot{Z}_{i\mathrm{tip}}\end{matrix} =
\begin{bmatrix}
\left\{\begin{array}{l}
\dot{X}_{CG} + \Bigl(\bigl(-s\psi s\theta c\varnothing s_{1i}l_1 + c\psi s\varnothing s_{1i}l_1 + c\psi s\theta c\varnothing c_{1i}l_1 + s\psi s\varnothing c_{1i}l_1 \\
-s\psi s\theta c\varnothing c_{1i}s_{2i}l_2 + c\psi s\varnothing c_{1i}s_{2i}l_2 - c\psi s\theta c\varnothing s_{1i}s_{2i}l_2 - s\psi s\varnothing s_{1i}s_{2i}l_2 \\
-s\psi s\theta c\varnothing s_{1i}c_{2i}l_2 + c\psi s\varnothing s_{1i}c_{2i}l_2 + c\psi s\theta c\varnothing c_{1i}c_{2i}l_2 + s\psi s\varnothing c_{1i}c_{2i}l_2\bigr)\dot{\theta}_1 \\
+\bigl(-s\psi s\theta c\varnothing c_{1i}s_{2i}l_2 + c\psi s\varnothing c_{1i}s_{2i}l_2 - c\psi s\theta c\varnothing s_{1i}s_{2i}l_2 \\
-s\psi s\varnothing s_{1i}s_{2i}l_2 - s\psi s\theta c\varnothing s_{1i}c_{2i}l_2 + c\psi s\varnothing s_{1i}c_{2i}l_2 \\
+c\psi s\theta c\varnothing c_{1i}c_{2i}l_2 + s\psi s\varnothing c_{1i}c_{2i}l_2\bigr)\dot{\theta}_2 \\
+\bigl(c\psi s\theta c\varnothing r_{yi} + s\psi s\varnothing r_{yi} - s\psi s\theta c\varnothing r_{zi} + c\psi s\varnothing r_{zi} - s\psi s\theta c\varnothing s_{1i}l_1 \\
+c\psi s\varnothing s_{1i}l_1 + c\psi s\theta c\varnothing c_{1i}l_1 + s\psi s\varnothing c_{1i}l_1 - s\psi s\theta c\varnothing c_{1i}s_{2i}l_2 + c\psi s\varnothing c_{1i}s_{2i}l_2 \\
-c\psi s\theta c\varnothing s_{1i}s_{2i}l_2 - s\psi s\varnothing s_{1i}s_{2i}l_2 - s\psi s\theta c\varnothing s_{1i}c_{2i}l_2 + c\psi s\varnothing s_{1i}c_{2i}l_2 \\
+c\psi s\theta c\varnothing c_{1i}c_{2i}l_2 + s\psi s\varnothing c_{1i}c_{2i}l_2\bigr)\omega_x \\
+\bigl(-c\psi s\theta c\varnothing r_{xi} - s\psi s\varnothing r_{xi} + c\theta c\varnothing r_{zi} + c\theta c\varnothing l_1 s_{1i} \\
+c\theta c\varnothing l_2 s_{2i}c_{1i} + c\theta c\varnothing l_2 c_{2i}s_{1i}\bigr)\omega_y \\
+\bigl(s\psi s\theta c\varnothing r_{xi} - c\psi s\varnothing r_{xi} - c\theta c\varnothing r_{yi} - c\theta c\varnothing l_1 c_{1i} \\
+c\theta c\varnothing l_2 s_{2i}s_{1i} - c\theta c\varnothing l_2 c_{2i}c_{1i}\bigr)\omega_z \\
+\bigl(((s\psi s\theta c\varnothing - c\psi s\varnothing)\,c_{1i} + (c\psi s\theta c\varnothing + s\psi s\varnothing)\,s_{1i})\,c_{2i} \\
+(-(s\psi s\theta c\varnothing - c\psi s\varnothing)\,s_{1i} + (c\psi s\theta c\varnothing - s\psi s\varnothing)\,c_{1i})\,s_{2i}\bigr)\dot{l}_{2i}\Bigr)
\end{array}\right\} \\
\left\{\begin{array}{l}
\dot{Y}_{CG} + \Bigl(\bigl(-s\psi s\theta s\varnothing s_{1i}l_1 - c\psi c\varnothing s_{1i}l_1 + c\psi s\theta s\varnothing c_{1i}l_1 \\
-s\psi c\varnothing c_{1i}l_1 - s\psi s\theta s\varnothing c_{1i}s_{2i}l_2 - c\psi c\varnothing c_{1i}s_{2i}l_2 - c\psi s\theta s\varnothing s_{1i}s_{2i}l_2 \\
+s\psi c\varnothing s_{1i}s_{2i}l_2 - s\psi s\theta s\varnothing s_{1i}c_{2i}l_2 - c\psi c\varnothing s_{1i}c_{2i}l_2 + c\psi s\theta s\varnothing c_{1i}c_{2i}l_2 - s\psi c\varnothing c_{1i}c_{2i}l_2\bigr)\dot{\theta}_1 \\
+\bigl(-s\psi s\theta s\varnothing c_{1i}s_{2i}l_2 - c\psi c\varnothing c_{1i}s_{2i}l_2 - c\psi s\theta s\varnothing s_{1i}s_{2i}l_2 \\
+s\psi c\varnothing s_{1i}s_{2i}l_2 - s\psi s\theta s\varnothing s_{1i}c_{2i}l_2 - c\psi c\varnothing s_{1i}c_{2i}l_2 \\
+c\psi s\theta s\varnothing c_{1i}c_{2i}l_2 - s\psi c\varnothing c_{1i}c_{2i}l_2\bigr)\dot{\theta}_2 \\
+\bigl(c\psi s\theta s\varnothing r_{yi} - s\psi c\varnothing r_{yi} - s\psi s\theta s\varnothing r_{zi} - c\psi c\varnothing r_{zi} - s\psi s\theta s\varnothing s_{1i}l_1 + c\psi c\varnothing s_{1i}l_1 \\
+c\psi s\theta s\varnothing c_{1i}l_1 - s\psi c\varnothing c_{1i}l_1 - s\psi s\theta s\varnothing c_{1i}s_{2i}l_2 - c\psi c\varnothing c_{1i}s_{2i}l_2 \\
-c\psi s\theta s\varnothing s_{1i}s_{2i}l_2 + s\psi c\varnothing s_{1i}s_{2i}l_2 - s\psi s\theta s\varnothing s_{1i}c_{2i}l_2 - c\psi c\varnothing s_{1i}c_{2i}l_2 \\
+c\psi s\theta s\varnothing c_{1i}c_{2i}l_2 - s\psi c\varnothing c_{1i}c_{2i}l_2\bigr)\omega_x \\
+\bigl(-c\psi s\theta s\varnothing r_{xi} + s\psi c\varnothing r_{xi} + c\theta s\varnothing r_{zi} + c\theta s\varnothing l_1 s_{1i} \\
+s_{2i}c_{1i}c\theta s\varnothing l_2 + c_{2i}s_{1i}c\theta s\varnothing l_2\bigr)\omega_y \\
+\bigl(s\psi s\theta s\varnothing r_{xi} + c\psi c\varnothing r_{xi} - c\theta s\varnothing r_{yi} - c\theta s\varnothing l_1 c_{1i} \\
+s_{2i}s_{1i}c\theta s\varnothing l_2 - c_{2i}c_{1i}c\theta s\varnothing l_2\bigr)\omega_z \\
+\bigl(((s\psi s\theta s\varnothing - c\psi c\varnothing)\,c_{1i} + (c\psi s\theta s\varnothing - s\psi c\varnothing)\,s_{1i})\,c_{2i} \\
+(-(s\psi s\theta s\varnothing + c\psi c\varnothing)\,s_{1i} + (c\psi s\theta s\varnothing - s\psi c\varnothing)\,c_{1i})\,s_{2i}\bigr)\dot{l}_{2i}\Bigr)
\end{array}\right\} \\
\left\{\begin{array}{l}
\dot{Z}_{CG} + \Bigl(\bigl(-s\psi c\theta s_{1i}l_1 + c\psi c\theta c_{1i}l_1 - s\psi c\theta c_{1i}s_{2i}l_2 \\
-c\psi c\theta s_{1i}s_{2i}l_1 - s\psi c\theta s_{1i}c_{2i}l_2 + c\psi c\theta c_{1i}c_{2i}l_2\bigr)\dot{\theta}_1 \\
+\bigl(-s\psi c\theta c_{1i}s_{2i}l_2 - c\psi c\theta s_{1i}s_{2i}l_2 - s\psi c\theta s_{1i}c_{2i}l_2 \\
+c\psi c\theta c_{1i}c_{2i}l_2\bigr)\dot{\theta}_2 \\
+\bigl(c\psi c\theta r_{yi} - s\psi c\theta r_{zi} - s\psi c\theta s_{1i}l_1 + c\psi c\theta c_{1i}l_1 - s\psi c\theta c_{1i}s_{2i}l_2 \\
-c\psi c\theta s_{1i}s_{2i}l_2 - s\psi c\theta s_{1i}c_{2i}l_2 + c\psi c\theta c_{1i}c_{2i}l_2\bigr)\omega_x \\
+(-c\psi c\theta r_{xi} - s\theta r_{zi} - s_{1i}s\theta l_1 - s_{2i}c_{1i}s\theta l_2 - c_{2i}s_{1i}s\theta l_2)\,\omega_y \\
+\bigl(s\theta r_{yi} + s\psi c\theta r_{xi} + c_{1i}s\theta l_1 - s_{2i}s_{1i}s\theta l_2 + c_{2i}c_{1i}s\theta l_2\bigr)\,\omega_z \\
+((s\psi c\theta c_{1i} + c\psi c\theta s_{1i})\,c_{2i} + (-s\psi c\theta s_{1i} + c\psi c\theta c_{1i})\,s_{2i})\,i_{2i}\Bigr)
\end{array}\right\}
\end{bmatrix}
\tag{14.46}
$$

Using these equations, a sub-model of the Jacobian is prepared in bond graph form as a conservative transformer junction structure. For the leg tip velocity control, reference velocity is compared with the actual leg tip velocity and error values are sent to PI controller. PI controller sends corrective signals to the Jacobian. Jacobian decides required efforts at the joints for leg forward movement, which is supplied to the motor in terms of voltage. The equations for the voltage supplied to the motor can be given as,

$$V_1 = s_X\alpha_{12} + s_Y\alpha_{11} + s_Z\alpha_{10} \tag{14.47}$$

$$V_2 = s_X\alpha_{15} + s_Y\alpha_{14} + s_Z\alpha_{13} \tag{14.48}$$

where s_X, s_Y, and s_Z are the corrective signals from PI controllers in X, Y, and Z directions, respectively. The detailed description of coefficients α_{ij} used in Eqs. (14.47) and (14.48) are obtained from leg tip velocity expressions. Figure 14.12 shows signal flow diagram of three dimensional model of Jacobian

14.3.2.4 Body Forward Motion Control

Body forward motion is achieved by actuating joints 1 and 2 of those legs which are in contact with the ground so that joint angles are restored to their initial positions. The voltage supplied to the joints through a PID controller is given as

$$V_2 = K_{pb}\left(\theta_d - \theta_a\right) + K_{db}\left(\dot{\theta}_d - \dot{\theta}_a\right) + K_{ib}\int\left(\theta_d - \theta_a\right)\mathrm{d}t \tag{14.49}$$

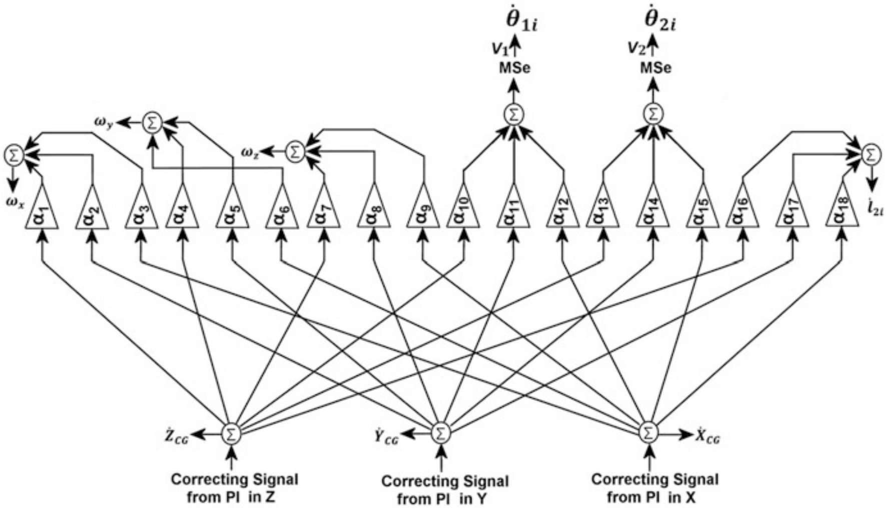


Fig. 14.12 Signal flow diagram of three dimensional model of Jacobian

where V_2 is the input voltage supplied to the joint actuator of leg; K_{pb}, K_{db}, and K_{ib} are the proportional, derivative, and integral gains, respectively; θ_d is the desired position, θ_a is the actual position, $\dot{\theta}_d$ is the desired joint velocity, and $\dot{\theta}_a$ is the actual joint velocity.

14.3.2.5 Combined Control Law

Combined form of the two control laws is entered in bond graph and since both the controllers do not work simultaneously for same joint, a switching is used to operate a particular control law. Combined control law can be presented as

$$V = a_1 V_1 + a_2 V_2 \tag{14.50}$$

where

$a_1 = 1$ during leg forward motion and 0 during body forward motion, and
$a_2 = 1$ during body forward motion and 0 during leg forward motion

14.3.2.6 Three Dimensional Model of Quadruped Robot for Workspace Control

Three dimensional model of quadruped robot presented in Sect. 14.2 is modified to control in workspace. In this model, joint actuation torque is evaluated as per the given reference tip trajectory through Jacobian and PI controller. The word bond graph of quadruped robot control in workspace is shown in Fig. 14.13. While multibond graph presentation of the model is shown in Fig. 14.14.

14.4 Results and Discussions

Above discussed bond graph model can be used for various research aspects pertaining to quadruped robot. Discussed joint space control and workspace control strategies are simulated using developed bond graph model, and its results are verified through animation and experimental results. Gait pattern considered for locomotion control is trot gait. In trot gait, diagonally opposite pairs of legs are actuated together to move forward. This is inspired from the way a horse moves. This two-beat diagonal gait minimizes the shift in body centroid and ensures good dynamic stability. Therefore, the quadruped can achieve higher locomotion speed with this gait. The usefulness of a dynamic model comes when dynamic forces are significant. Therefore, trot gait has been considered in this chapter for model validation. Influence of compliance is studied on locomotion parameter. The performance measure is evaluated based on energy efficiency.

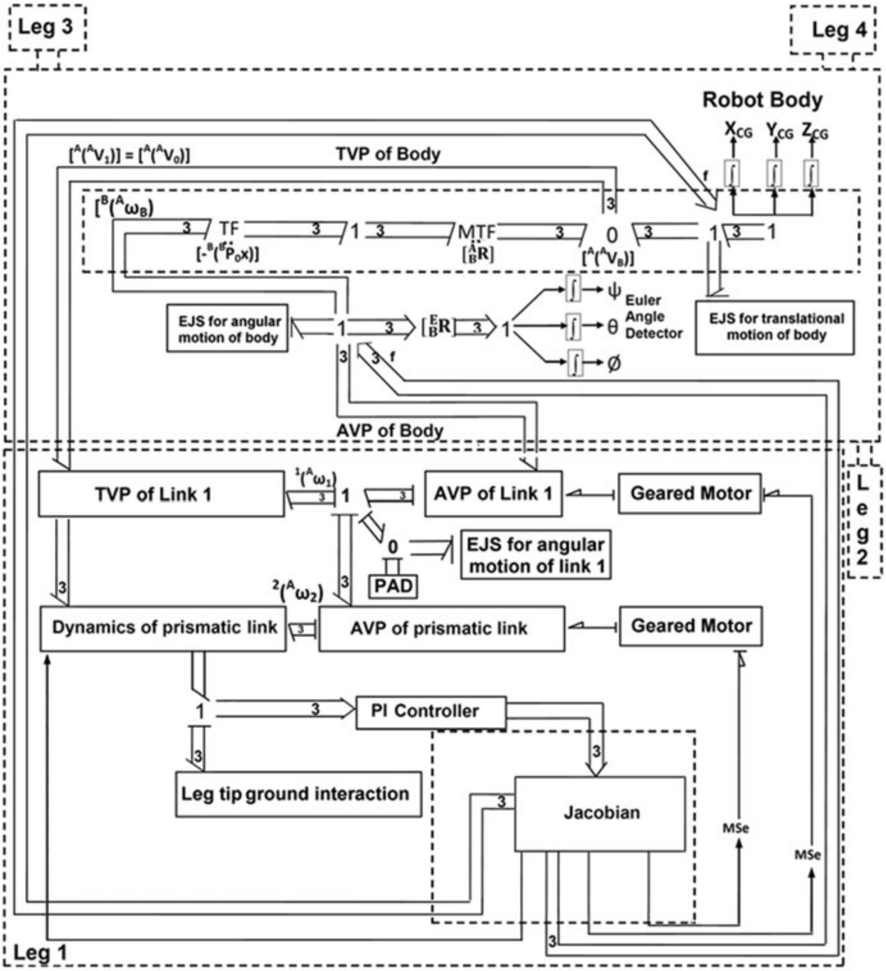


Fig. 14.13 Multibond graph of quadruped robot for locomotion control in workspace

14.4.1 Simulation, Animation, and Experiment Results of Locomotion Control in Joint Space

In this section, locomotion control strategy discussed in Sect. 14.3.1 is validated using simulation and animation of bond graph dynamic model and also through experiment result.

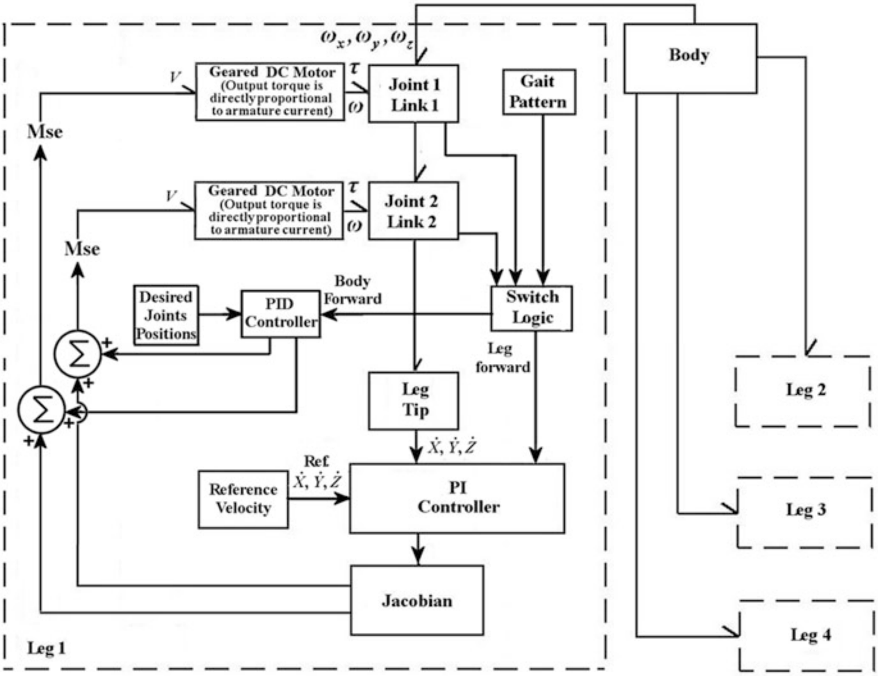

$\omega_x, \omega_y, \omega_z$
Body
V
Geared DC Motor (Output torque is directly proportional to armature current)
τ
ω
Joint 1 Link 1
Gait Pattern
Mse
V
Geared DC Motor (Output torque is directly proportional to armature current)
τ
ω
Joint 2 Link 2
Mse
Desired Joints Positions
PID Controller
Body Forward
Switch Logic
Leg forward
+
+
+
+
Leg Tip
Leg 2
$\dot{X}, \dot{Y}, \dot{Z}$
Ref $\dot{X}, \dot{Y}, \dot{Z}$
Reference Velocity
PI Controller
Leg 3
Jacobian
Leg 4
Leg 1

Fig. 14.14 Word bond graph of quadruped robot control in workspace

14.4.1.1 Simulation Results for Locomotion Control in Joint Space

Bond graph model is simulated for trot gait, for which controlled efforts are supplied to the actuators, to reach to required joint position as shown in Table 14.1. Input parameters considered for the simulation are listed in Table 14.2.

Positive *Y* direction is considered as forward direction. Time duration for each cycle is 1.7 s. Simulation is carried out for five cycles. Figure 14.15 shows leg tip displacement in *Y* direction, while Fig. 14.16 shows body CG movement in *X*, *Y*, and *Z* directions. It shows robot moves in a forward *Y* direction. Legs joint rotations for the said motion are shown in Fig. 14.17a, b for joint 1 and 2, respectively. Figure 14.18 shows leg tip displacement in *Z* direction. Similar way, simulation can be carried out for other gait also.

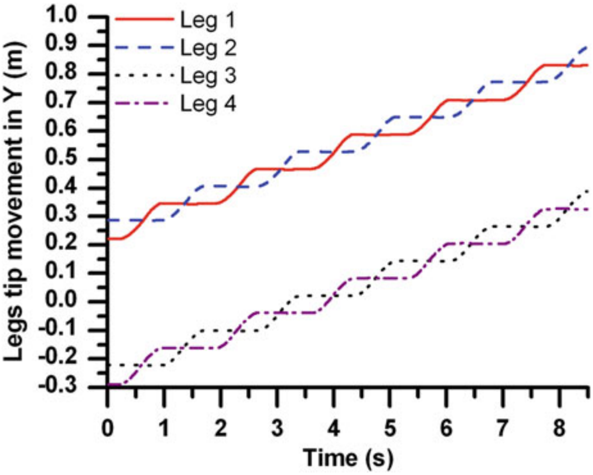
14.4.1.2 Animation Results for Locomotion Control in Joint Space

Animation of above discussed locomotion strategy of quadruped robot is carried out in SYMBOLS Shakti Animator [34]. The physical characteristics of certain selected aspects of bond graph model can be directly visualized in this animator, for a better

Table 14.2 Input parameters

Parameters	Value
Leg parameters	
First link length of leg (l_1)	0.225 m
Mass of first link (M_{l1})	1.11 kg
Mass of cylinder part of the prismatic link (M_c)	0.3 kg
Mass of piston part of the prismatic link (M_p)	0.2 kg
Inertia of link 1	
I_{xx1}	0.013346 kg m^2
I_{yy1}	0.0073965 kg m^2
I_{zz1}	0.011563 kg m^2
Inertia of cylinder part of prismatic link	
$I_{xxc} = I_{zzc}$	0.005144 kg m^2
I_{yyc}	0.0000487 kg m^2
Inertia of piston and piston rod of prismatic link	
$I_{xxp} = I_{zzp}$	0.00168 kg m^2
I_{yyp}	0.000025 kg m^2
Stiffness of spring in compliant link (k_f)	5000 N/m
Friction between the piston and cylinder of a prismatic link (R_f)	274 Ns/m
Contact point stiffness at the piston cylinder of prismatic link (k_b)	10^8 N/m
Contact point resistance at the piston and cylinder of prismatic link (R_b)	10^3 Ns/m
Length of piston and piston rod of prismatic link l_p	0.1 m
Distance of cylinder CG from a cylinder end frame of a prismatic link (l_{cg})	0.05 m
Distance of piston CG from a piston end frame of a prismatic link (l_{pg})	0.07 m
Mass of piston and piston rod of the prismatic link (m_p)	0.2 kg
Mass of cylinder part of the prismatic link (m_c)	0.3 kg
Position of the cylinder end point with respect to the body fixed frame at the mass center in meter (x_2, y_2, z_2)	(0.0, −0.05, 0.0)
Position of the piston end point with respect to the body fixed frame at the mass center in meter (x_3, y_3, z_3)	(0.0, 0.07, 0.0)
Common parameter	
Mass of body (Mb)	6.94 kg
Inertia of body	
I_{xb}	0.1470 kg m^2
I_{yb}	0.1045 kg m^2
I_{zb}	0.2466 kg m^2
Ground damping in x, y, z direction (R_{gx}, R_{gy}, R_{gz})	1000 Ns/m
Ground stiffness in z direction (K_{gz})	10^6 N/m
Controller parameter	
Proportional gain of controller (K_P)	100
Derivative gain of controller (K_D)	80
Integral gain of controller (K_I)	50
Actuator parameter	
Motor constant (K_m)	0.0276 Nm/A
Motor armature resistance (R_m)	0.386 Ω
Motor inductance (L_m)	0.001 H
Gear ratio (n)	230

Fig. 14.15 Legs tip movement in *Y* direction



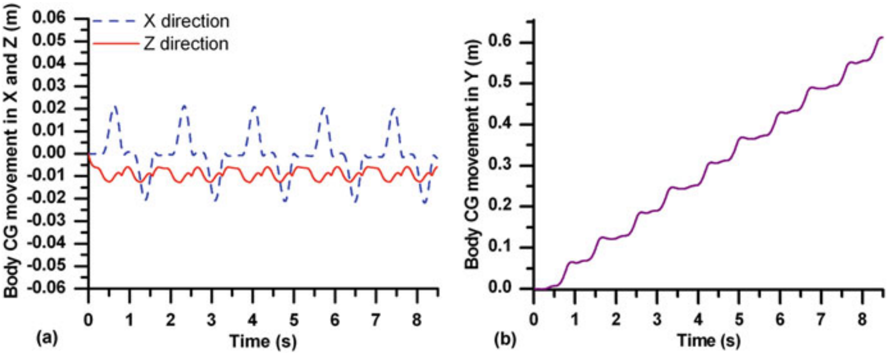


Fig. 14.16 Body CG movement; (**a**) *X* and *Z* direction, (**b**) *Y* direction

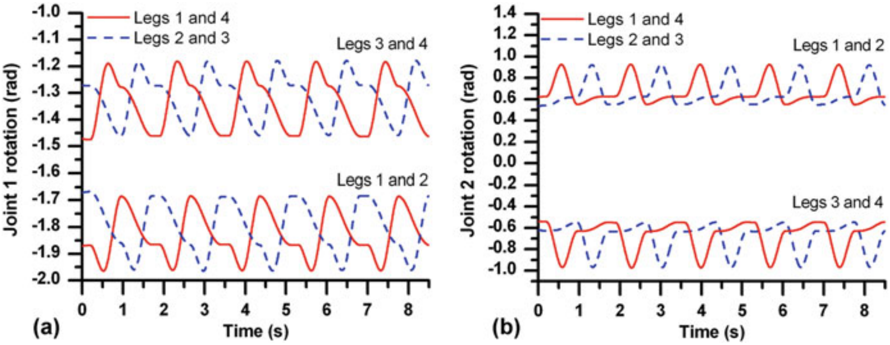


Fig. 14.17 Joints rotation: (**a**) Joint 1 and (**b**) Joint 2

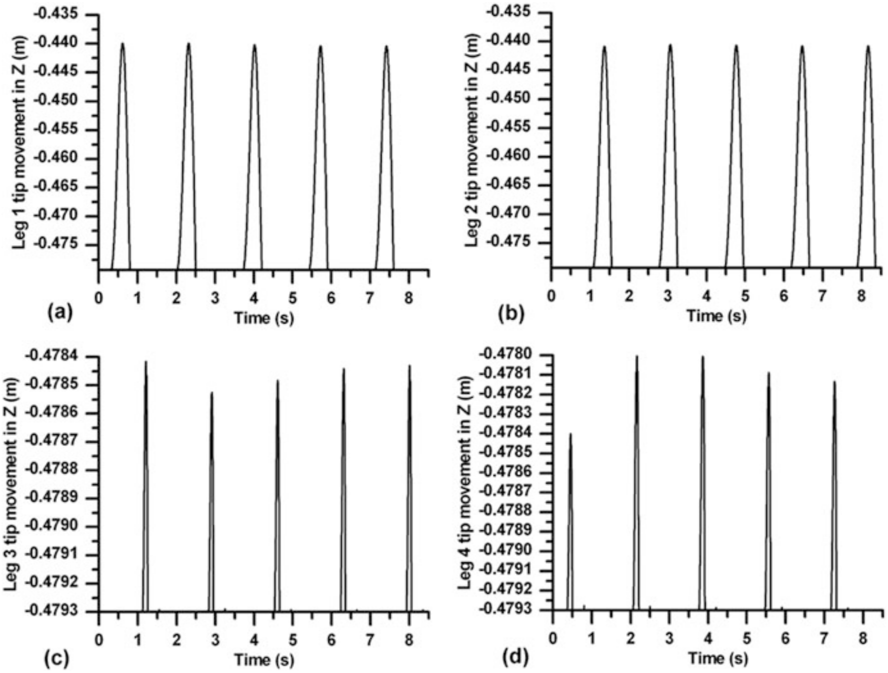


Fig. 14.18 Legs tip Z displacement; (**a**) leg 1, (**b**) leg 2, (**c**) leg 3, (**d**) leg 4

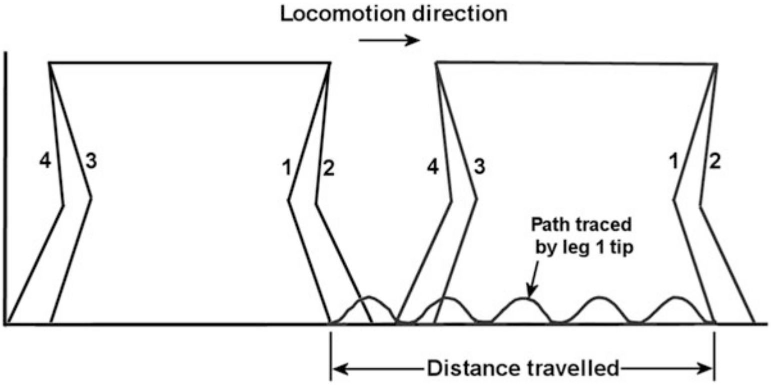


Fig. 14.19 Animation frames of quadruped robot locomotion control in joint space

understanding of intrinsic behavior of quadruped robot. Animation of quadruped robot is created from the simulation results. Figure 14.19 shows animation frames of quadruped robot walking.

To maintain neatness of the figure only two frames of animations are shown. First frame indicates starting position and second frame indicates final position of quadruped robot at the end of five cycles. Path traced by tip of leg 1 is also shown in this figure.

14.4.1.3 Experiment Result for Locomotion Control in Joint Space

Above presented simulation and animation results show that quadruped robot travels 0.61 m in five cycles with the help of the developed locomotion strategy. The same strategy is implemented in the physical model. As shown in Fig. 14.1a, the physical model of quadruped robot contains total eight number of Maxon brushless DC motors. To control these motors, eight Maxon (EPOS) controllers are used. One of those acts as master and the remaining seven act as slave controllers. This robot has approximately 15 kg weight and its body length and width are 0.5 m and 0.42 m, respectively. Its height is 0.479 m. Above discussed locomotion strategy is applied on physical model and it is observed that it travels 0.55 m in five cycles. Figure 14.19 shows few snaps taken during locomotion of quadruped robot. Figure 14.20a shows beginning of a cycle, Fig. 14.20b, c shows leg motion in intermediate stages, and Fig. 14.20d shows end of a cycle. For the movements detailed above, the joint rotations performed by all legs are shown in Fig. 14.21.

The pattern of joint rotations during the experiments is almost the same as simulation. There are always some assumptions made for numerical modeling of any system. Here, the assumptions for quadruped modeling are mass center of link is located at the mid of its length, center of gravity of top body is located at the center of body, top body and upper links are rigid, joint rotation allows rotation of link about one axis only, robot is walking on hard surface and on even terrain, and external force and moment effects are negligible. Condition of the surface, on which physical robot walks, affects robot locomotion. It seems from our simulation, animation, and experimental results that little deviations observed in experiments are because of the assumptions considered during modeling and uncertain surface conditions like its roughness, friction, and elevation. These experimental results support the correctness of the dynamic model generated in bond graph.

14.4.2 Simulation, Animation, and Experiment Results of Locomotion Control in Workspace

In this section, the control strategy developed in Sect. 14.3.2 is validated. Simulation error 5×10^{-6} and the step size 10 are considered for numerical integration in bond graph simulation environment. Input parameters used for simulation are same as shown in Table 14.2.

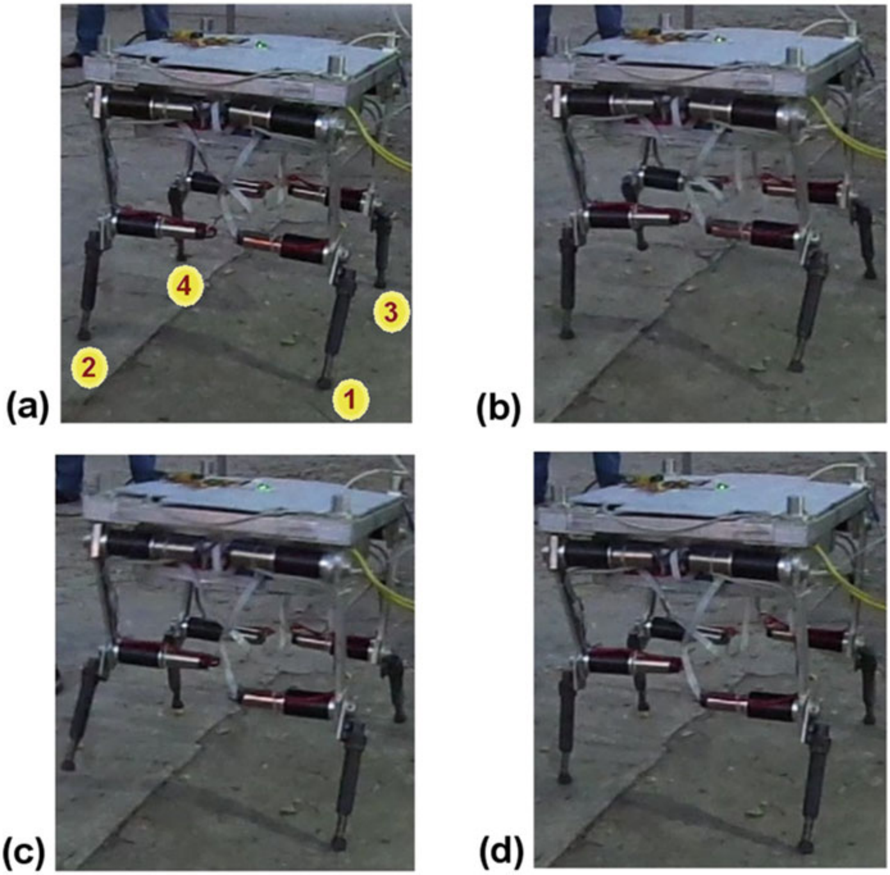


Fig. 14.20 Experiment on physical model, (**a**) initial position, (**b**) and (**c**) intermediate positions, (**d**) end of cycle

14.4.2.1 Simulation and Animation Results for Locomotion Control in Workspace

This dynamic model is simulated for trot gait. Simulations and animations have been carried out for six cycles. In case of trot gait, each cycle takes 1.7 s time. Figure 14.22 shows few animation frames of quadruped motion in trot gait. To maintain neatness of the figure only a few animation frames are shown from the entire simulation results. It shows that quadruped locomotion is achieved in the position *Y* direction.

Figure 14.23a shows the reference tip displacement (as per the prescribed reference velocity) and the actual tip displacement of legs 1 and 4, while Fig. 14.23b shows those of legs 2 and 3. These results show that the leg tips follow the corresponding reference trajectories. Error in legs displacement in *Y* direction of

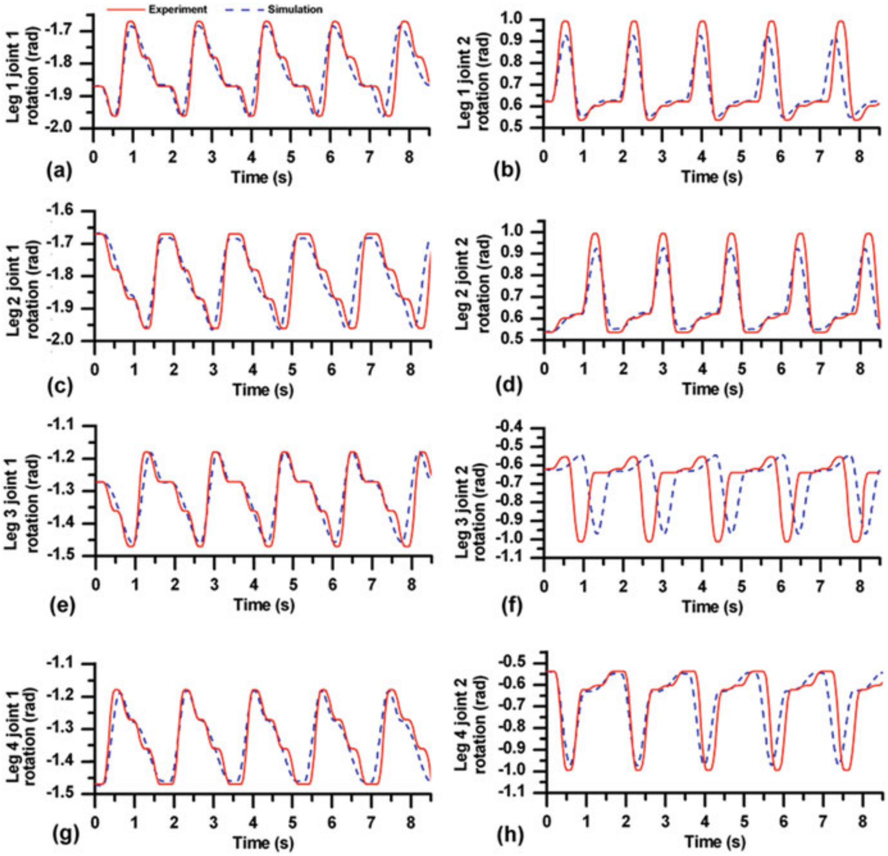


Fig. 14.21 Joints rotations during experiment and simulation, (**a**) leg 1 joint 1, (**b**) leg 1 joint 2, (**c**) leg 2 joint 1, (**d**) leg 2 joint 2, (**e**) leg 3 joint 1, (**f**) leg 3 joint 2, (**g**) leg 4 joint 1, and (**h**) leg 4 joint 2

legs 1–4 are shown in Fig. 14.24a–d, respectively. The robot body movement in *Y* direction is shown in Fig. 14.25a while that in *Z* direction is shown in Fig. 14.25b. For the executed locomotion, the joint rotations in joints 1 and 2 of all legs are shown in Fig. 14.26a, b, respectively.

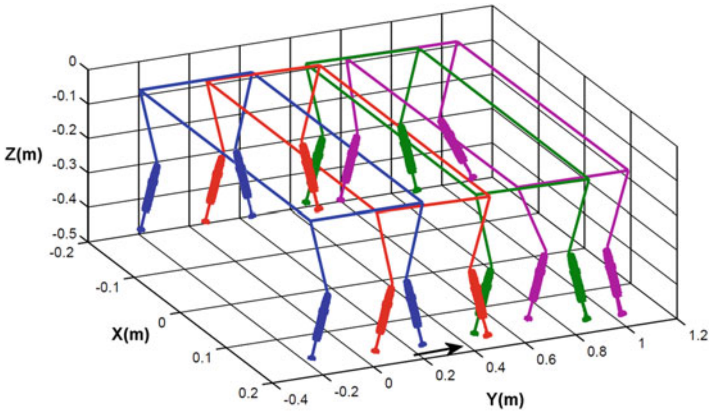


Fig. 14.22 Animation frames of quadruped locomotion control in workspace

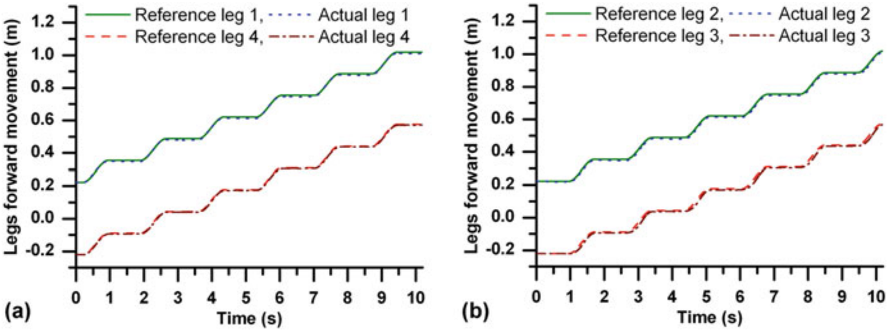


Fig. 14.23 Actual and reference leg tip displacements in forward direction for trot gait (**a**) legs 1 and 4, (**b**) legs 2 and 3

The trot gait in quadruped robot is a dynamic gait where the robot is supported by either of the diagonally opposite pairs of legs during which the body moves forward. In order to prevent robot falling down, the resultant of gravitational and inertial forces passes through a point on the line joining the tips of the supporting legs. This ensures dynamic stability. Here, Zero Moment Point (ZMP) introduced in [42] is evaluated to ensure the dynamic stability. By having the ZMP on the diagonal line which connects tips of two legs in contact with the ground, stability can be assured. The ZMP can be calculated as in [44]

$$X_{\text{zmp}} = \frac{\sum_{i=1}^{n} m_i \left(\ddot{Z}_i + g\right) X_i - \sum_{i=1}^{n} m_i \ddot{X}_i Z_i - \sum_{i=1}^{n} I_{iy} \ddot{\theta}_{iy}}{\sum_{i=1}^{n} m_i \left(\ddot{Z}_i + g\right)} \tag{14.51}$$

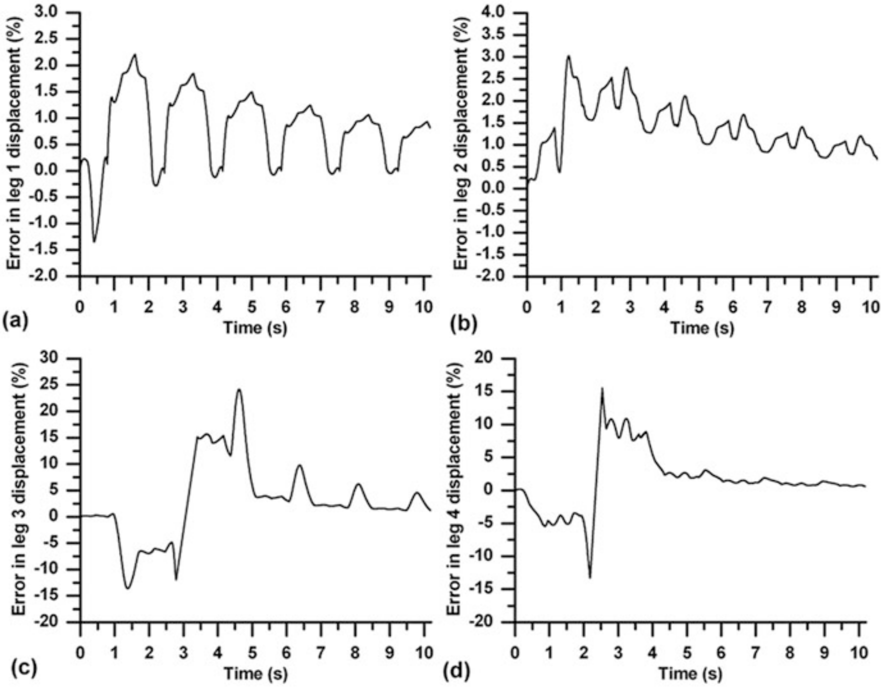


Fig. 14.24 Error in legs forward movement for trot gait: (**a**) leg 1, (**b**) leg 2, (**c**) leg 3 and (**d**) leg 4

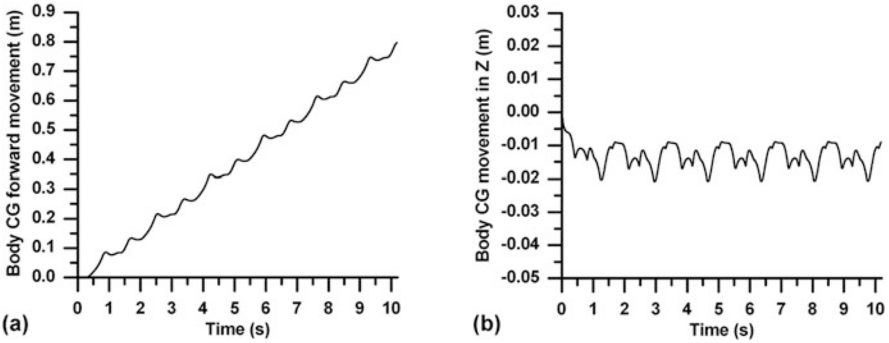


Fig. 14.25 Body CG displacement for trot gait: (**a**) *Y* and (**b**) *Z* direction

$$Y_{\text{zmp}} = \frac{\sum_{i=1}^{n} m_i\left(\ddot{Z}_i + g\right)Y_i - \sum_{i=1}^{n} m_i\ddot{Y}_i Z_i - \sum_{i=1}^{n} I_{ix}\ddot{\theta}_{ix}}{\sum_{i=1}^{n} m_i\left(\ddot{Z}_i + g\right)} \tag{14.52}$$

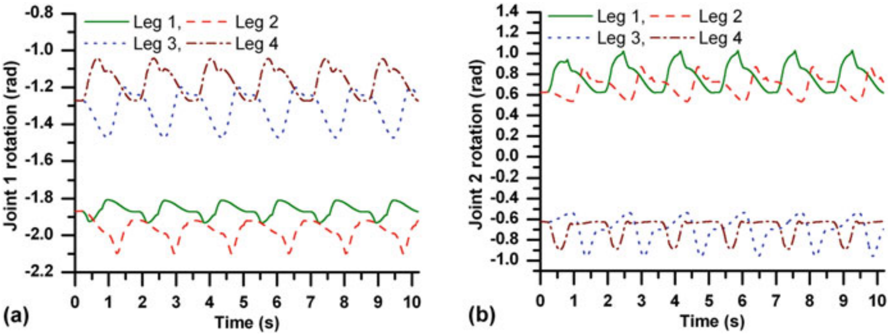


Fig. 14.26 Joints rotation for trot gait: (**a**) joint 1 and (**b**) joint 2

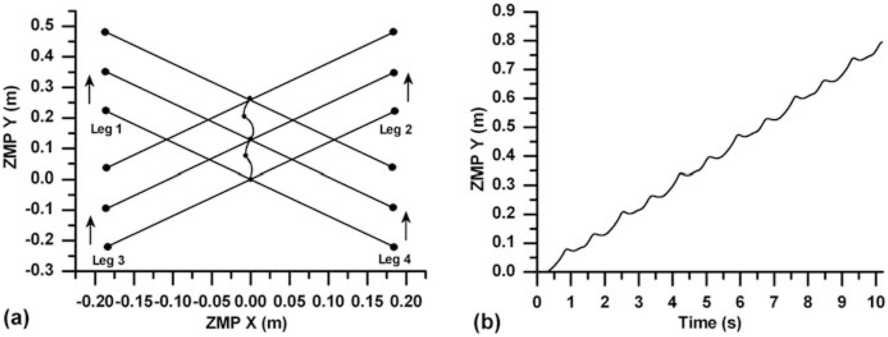


Fig. 14.27 ZMP movement for trot gait: (**a**) *X*-*Y* plane for two cycles and (**b**) in *Y* direction

where (X_{zmp}, Y_{zmp}) are coordinates of the ZMP, n is the number of links, i is an enumerator, m_i is mass of i-th link, g is gravitational acceleration, I_i is rotary inertia of i-th link, (X_i, Y_i, Z_i) are coordinates of the mass center of link i, and $\ddot{\theta}_i$ is the angular acceleration of link i. Figure 14.27a presents the change in the location of the ZMP as the locomotion is executed. For maintaining the clarity of presentation, only the steps followed by the legs during the first two cycles are shown. During the first cycle, legs 2 and 3 are in contact with the ground and the ZMP lies on the line connecting the tips of legs 2 and 3, while during the second cycle, legs 1 and 4 are in contact with the ground and the ZMP lies on the line connecting the tips of legs1 and 4. For six numbers of cycles, the shift of the ZMP in X direction is found to be in the range −0.020 m to +0.001 m. Figure 14.27b shows the ZMP drift in Y direction due to robot locomotion.

14.4.2.2 Experiment Results for Locomotion Control in Workspace

Input to the Physical Model

According to the proposed locomotion control strategy, reference leg tip velocity should be given as input which is continuously compared with actual velocity. The error should be used to send correcting signal to the actuators. To know leg tip velocity/position, sensors (motion trackers) are required. These sensors were not available with the author at the time of experiment. So, data obtained through simulation results are used to perform experiment. Here, joint positions obtained during simulation are fed to the physical model. Maxon controllers operate all the actuators as per the data fed.

Experiment Results

As per the simulation results when quadruped walk with trot gait (control in workspace), it travels 0.79 m distance. When same data are used to operate the physical model it travels around 0.75 m distance for six cycles. Few snaps are taken during these movements which are shown in Fig. 14.28. Figure 14.29 shows joints position of physical model while walking with trot gait along with simulation results which shows that both are matching with each other. To perform this experiment,

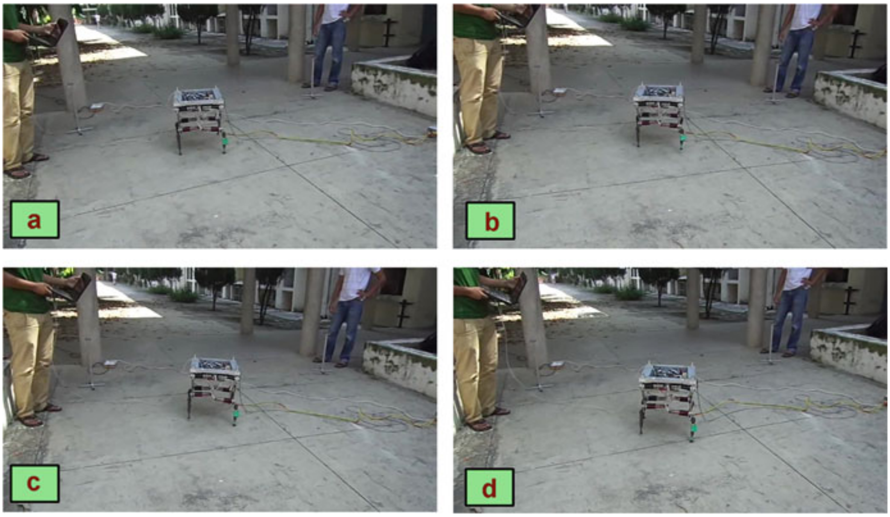


Fig. 14.28 Snaps taken during quadruped walk with trot gait control in workspace: (**a**) initial position, (**b**) and (**c**) intermediate position, (**d**) final position

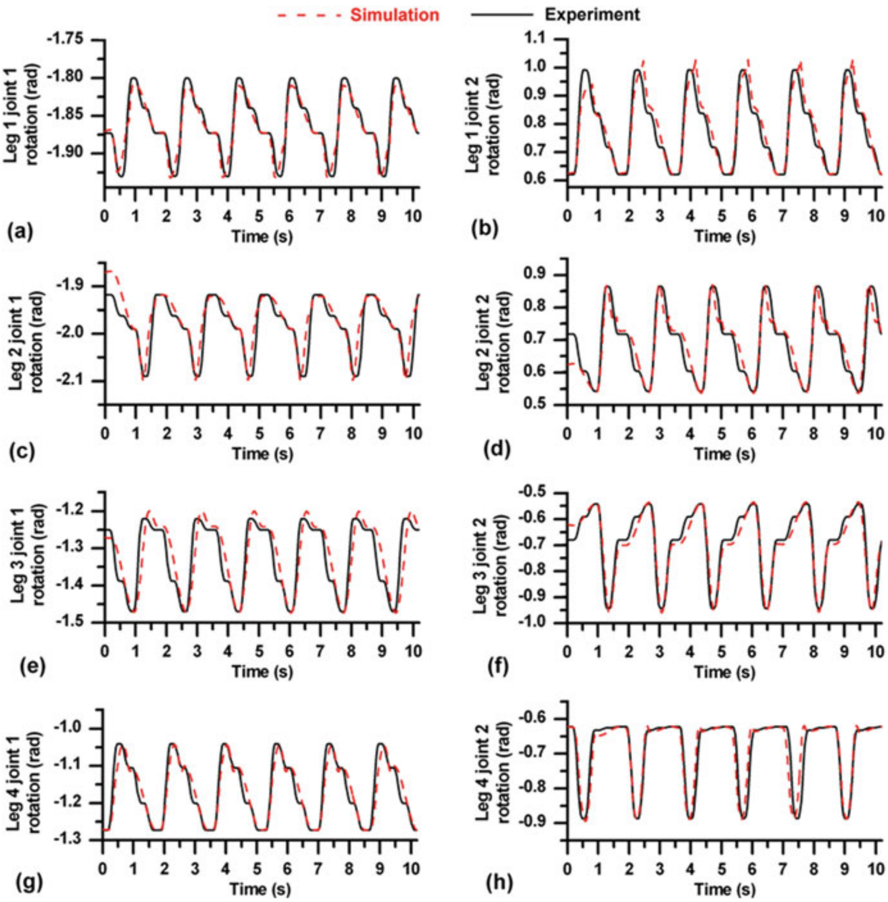


Fig. 14.29 Joints rotation of quadruped walk with trot gait control in workspace: (**a**) leg 1 joint 1, (**b**) leg 1 joint 2, (**c**) leg 2 joint 1, (**d**) leg 2 joint 2, (**e**) leg 3 joint 1, (**f**) leg 3 joint 2, (**g**) leg 4 joint 1, and (**h**) leg 4 joint 2

required code is prepared in OPEN-PCS using the joint angles obtained through simulation results. Then, it is fed to Maxon controller. While executing this code, it is found that physical model travels perfectly.

14.4.3 Simulation of Turning Motion

It is understood that if robot leg has more than one degree of freedom with joint rotation about different axis, then only other than straight line motion is possible. The discussed robot configuration has two DOF per leg but both axes of rotation are same, i.e., about *X*. Thus each leg tip travels same distance while in motion during

Fig. 14.30 Animation frames in top view of quadruped robot in turning motion

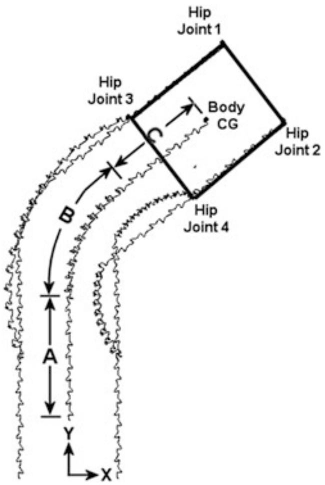


Table 14.3 Joint position of amble gait for turning motion

Joint position corresponding to time interval	Leg 1		Leg 2		Leg 3		Leg 4	
	Joint 1	Joint 2	Joint 1	Joint 2	Joint 1	Joint 2	Joint 1	Joint 2
Initial joint angle	−1.871	0.622	−1.747	0.586	−1.271	−0.622	−1.395	−0.586
$T_o \leq t < T_1/2$	−1.979	1.199	−1.781	0.602	−1.323	−0.615	−1.431	−0.564
$T_1/2 \leq t < T_1$	−1.417	0.265	−1.813	0.613	−1.381	−0.593	−1.471	−0.537
$T_1 \leq t < T_2/2$	−1.530	0.407	−1.843	0.620	−1.448	−0.554	−1.179	−0.994
$T_2/2 \leq t < T_2$	−1.618	0.494	−1.871	0.622	−1.523	−0.494	−1.271	−0.622
$T_2 \leq t < T_3/2$	−1.694	0.554	−1.962	0.944	−1.611	−0.407	−1.299	−0.620
$T_3/2 \leq t < T_3$	−1.760	0.593	−1.671	0.537	−1.725	−0.265	−1.328	−0.613
$T_3 \leq t < T_4/2$	−1.819	0.615	−1.710	0.564	−1.162	−1.199	−1.361	−0.602
$T_4/2 \leq t < T_5$	−1.871	0.622	−1.747	0.586	−1.271	−0.622	−1.395	−0.586

one cycle and robot has straight line motion. If one side of both the leg tips travels same distance but different from opposite side of leg tips (i.e., legs 1 and 3 tip travels same distance but they are different from legs 2 and 4 tip travels), then this motion gives a turn instead of moving in straight line. This concept has been demonstrated using amble gait. Simulation and animation is carried out for 20 cycles. Each cycle takes 0.9 s. Figure 14.30 shows animation frames of quadruped locomotion which moves straight for initial five cycles (A) then it takes turn for next ten cycles (B) and finally moves straight for remaining five cycles (C). In this figure path traced by hip joints and body CG are shown. Joint position corresponding to time interval is shown in Table 14.3.

Figure 14.31 shows body CG *X* motion while Figs. 14.32 and 14.33 show body CG *Y* and *Z* motions, respectively. Leg tip motion in *X* and *Y* directions are shown in Figs. 14.34 and 14.35, respectively. As discussed above, it is necessary to have different joint rotations for this kind of locomotion. Joints 1 and 2 rotations carried

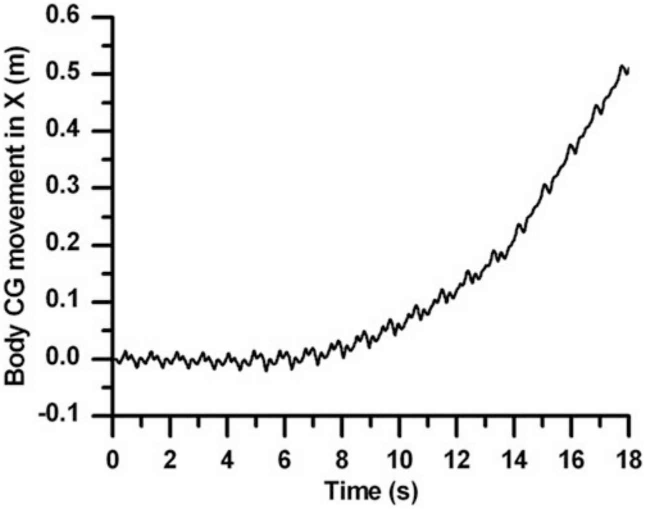


Fig. 14.31 Body CG movement in *X* direction

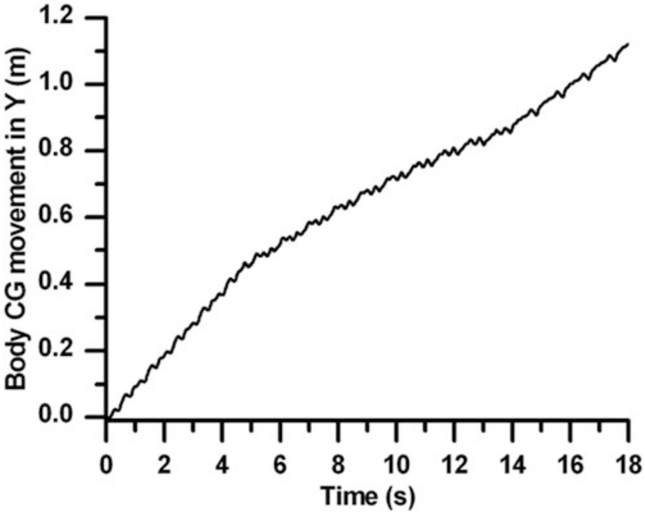


Fig. 14.32 Body CG movement in *Y* direction

out of all legs for the said motions are shown in Figs. 14.36 and 14.37, respectively. Thus, by having combination of different joint angles, motion other than straight line is possible.

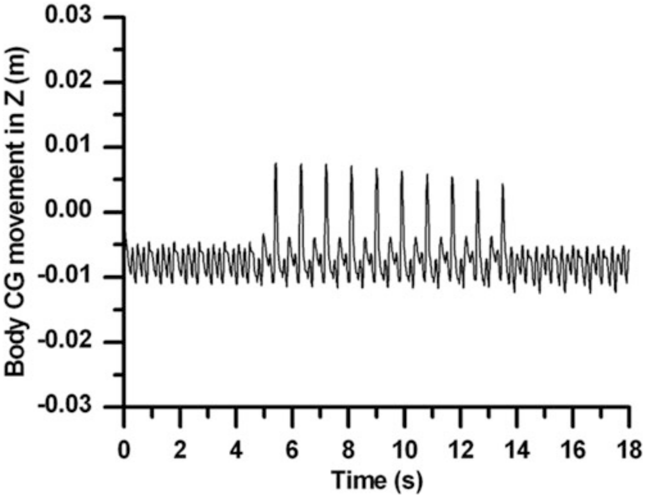


Fig. 14.33 Body CG movement in *Z* direction

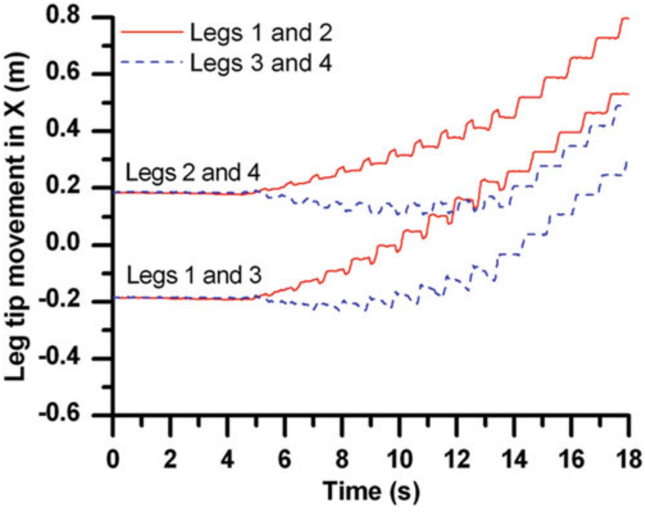


Fig. 14.34 Leg tip movement in *X* direction

14.4.4 Influence of Compliance on locomotion Parameter

Compliance affects locomotion speed and posture disturbance. In this section, attempt is made to study influence of compliance on both these locomotion parameters. Developed bond graph model is simulated for various compliance and damping value of the same damping ratio. The body CG travels (indicating motion

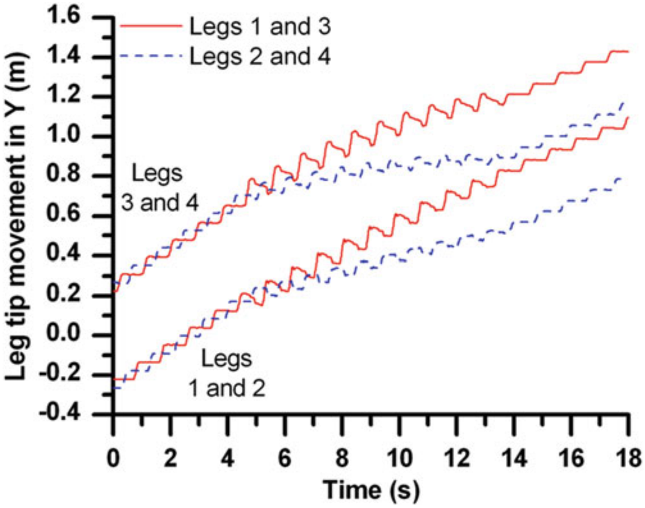


Fig. 14.35 Leg tip movement in *Y* direction

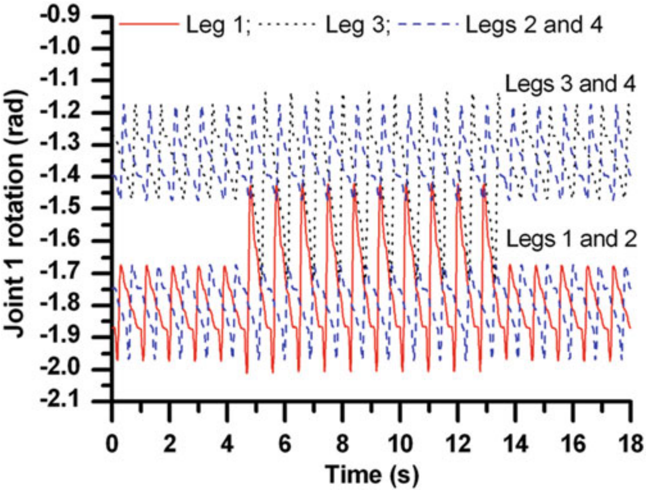


Fig. 14.36 Joint 1 rotation of legs

of robot in forward direction) and posture disturbance are analyzed. Figure 14.38a shows body CG travels by quadruped robot and Fig. 14.38b shows range of posture disturbance in roll and pitch during the locomotion.

It is concluded from the results that as the stiffness decreases, the body CG travel reduces. Body CG travel does not increase noticeably after increasing the stiffness further than about 7000 N/m. It is also seen from the results that the posture disturbance range at roll and pitch becomes higher at higher stiffness. It is always

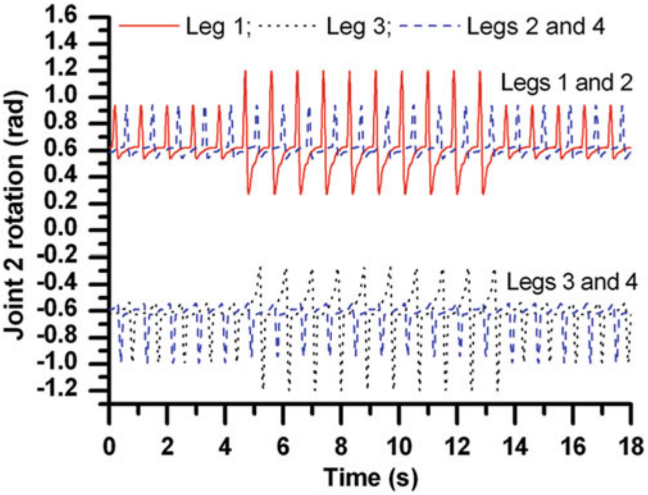
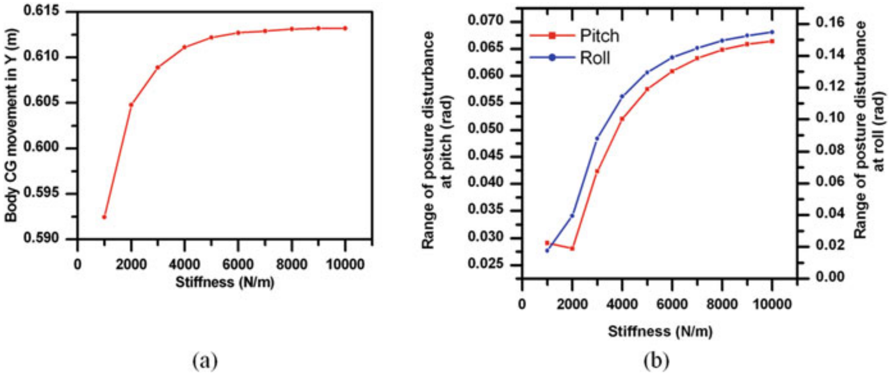


Fig. 14.37 Joint 2 rotation of legs



(a) (b)

Fig. 14.38 Influence of compliance on (**a**) body CG travels and (**b**) posture disturbance

desirable to have minimum posture disturbance with maximum body CG travel. Thus, for our robot configuration we have chosen 5000 N/m stiffness as a trade-off, which gives good locomotion with relatively low range of posture disturbances in roll and pitch.

14.4.5 Performance Measures Through Energy Efficiency

Robot walk may be evaluated by stability, maximum speed, and energy consumption. Parameters like a type of gait, stride length, duty factor, length of leg, joint angle, and rotation are deciding the above-mentioned performance criteria. To discuss advantage and disadvantage of different designs and control strategies, it is necessary to evaluate and compare robot performance and abilities with respect to common criteria. A common measure to evaluate and compare the energy efficiency of vehicles is the energy consumption per unit distance. To justify the comparison, mass moved, and velocity obtained should also be considered with the cost of locomotion. We are comparing here quadruped walk with specific resistance (ε). The specific resistance [15, 22] is a dimensionless number describing the energy efficiency of a mobile system. The specific resistance is defined as

$$\varepsilon = \frac{P(v)}{mgv} = \frac{E}{mgd} \tag{14.53}$$

where $P(v)$ is the power needed to move the body with velocity v, m is the mass of the system, g is the acceleration due to gravity, d is the distance travelled, and E is the energy spent. Since velocity is not constant in a cycle, energy spent to move unit weight by unit distance is preferred as the performance measure. Sufficient number of cycles of motion till steady-state is reached should be considered for evaluation of specific resistance. Lower specific resistance implies higher energy efficient.

14.4.5.1 Energy Efficient Structure of Quadruped Robot

In this section, energy efficiency of rigid legged and compliant legged robots are evaluated and compared. Locomotion gait is considered as trot for both the cases. The above developed bond graph model for compliant legged robot was modified to generate the model of rigid legged robot. This is done by locking the sliding motion of the piston within the prismatic link; thereby retaining the same mass parameters in both the models. The locking is performed through a high-stiffness virtual spring which is called a pad in bond graph terminology. Both rigid legged and compliant legged models are simulated for same speed which is fixed at 0.42 m distance travelled in 4.5 s., i.e., with an average locomotion speed of 0.093 m/s. The same speed requirement is satisfied by manually tuning the controller parameters and the cycle time durations. Figure 14.39a shows body CG travels by rigid legged quadruped robot in trot gait. Figure 14.39b shows body CG travels by compliant legged robot in trot gait. Sensors are placed at the actuators in bond graph models to extract the power consumption data. The specific resistance defined in Eq. (14.53) is evaluated for both these cases considering robot mass as 15 kg and gravitational acceleration as 9.81 m/s^2. It was found that the specific resistance for the compliance legged model is 0.925 while it is 1.10 for rigid legged model. It is known that smaller the specific resistance, higher the energy efficiency [15, 22, 38]. So from the above results, smaller specific resistance of compliant legged robot implies that it is energy efficient than rigid legged robot. Leg compliance reduces impact induced bounce and allows for better grip; thereby improving the locomotion efficiency. However, it may be noted that too low or too high-stiffness reduce grip and increase losses

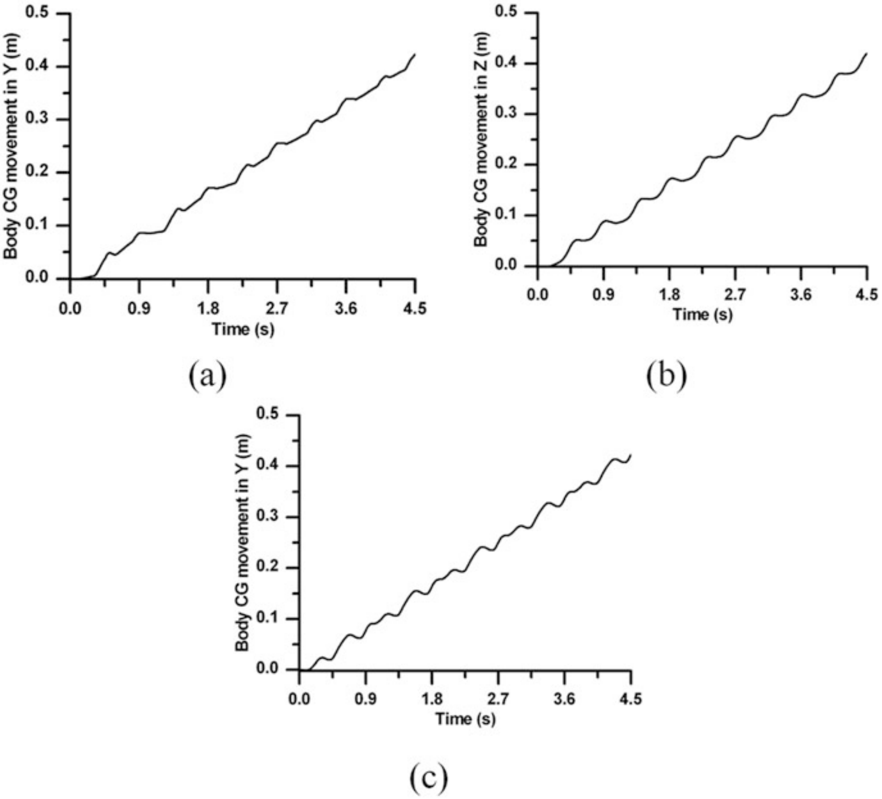


Fig. 14.39 Body CG Y displacement in various conditions: (**a**) rigid legged model with trot gait, (**b**) compliant legged model with trot gait, (**c**) compliant legged model with amble gait

due to sliding between the piston and cylinder of the prismatic pairs. Thus, there is an optimum value of compliance where lowest specific resistance is possible. That optimum value has not been obtained here. We have merely compared the performance of the compliant legged robot with design parameters chosen from Fig. 14.38 with a rigid legged robot.

14.4.5.2 Energy Efficient Locomotion Gait

In this section, energy efficient locomotion gait is evaluated from the static and dynamic gaits. The trot gait is considered as a dynamically stable gait, while the amble gait is considered as a statically stable gait. Specific resistance of trot gait of compliant legged structure is just evaluated. Thus, it is only need to evaluate specific resistance of quadruped walking with amble gait for comparison. Model is simulated for amble gait. Here also results are compared for the same speed which is obtained by tuning controller parameters and cycle duty times. Body CG displacement obtained in *Y* direction (along the longitudinal axis) for amble gait is shown in Fig. 14.39c. Results obtained are used to evaluate Eq. (14.53) which gives the specific resistance of the amble gait as 6.71. From this result it seems that specific resistance of the trot gait is smaller. Thus the trot gait is energy efficient than the amble gait.

14.4.5.3 Energy Efficient Foot Trajectory

Foot trajectory in case of walking robot has two main parameters: leg lift (LL) and step length (SL), which almost decides the foot trajectory of leg tip. However, the way leg tip reach to LL decides exact curve. Here, numbers of simulations are carried out to analyze the effect of LL and SL. For this analysis dynamic model should be controlled in workspace mode. Assumptions considered for the analysis are robot walks in straight line and on even terrain with trot gait.

First, simulations are carried out for different SL by keeping LL fixed as 0.033 m. Figure 14.40 shows given leg tip motion in *Y* and *Z* directions. Specific resistance for this run is plotted in Fig. 14.41 which indicates that for fixed LL as SL increases the specific energy decreases thus walk becomes more and more energy efficient.

Next, simulations are carried out for different LL by keeping SL fixed. Here, two sets of simulations are performed for SL as 0.099 and 0.132 m. Figure 14.42 shows given leg tip trajectory while Fig. 14.43 shows specific resistance. In both the cases it is observed that as LL increases specific resistance increases thus walk becomes less energy efficient. At the same time too much low LL also reports less energy efficient as its specific resistance goes high.

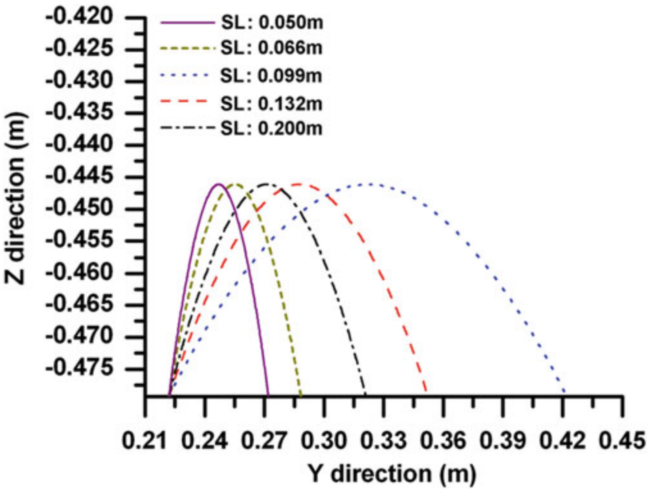


Fig. 14.40 Leg tip motion in *YZ* plane for different step length with same leg lift

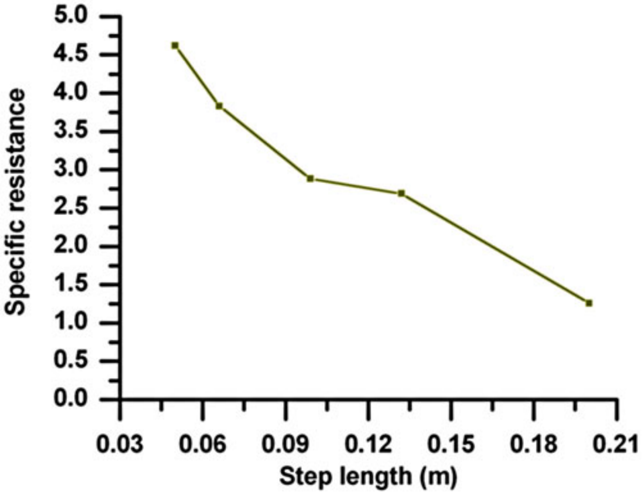


Fig. 14.41 Specific resistance for different step length with same leg lift

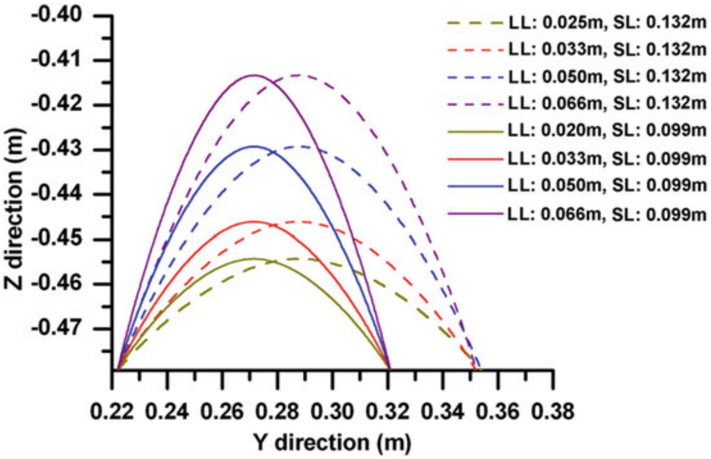


Fig. 14.42 Leg tip motion in *YZ* plane for different foot height with same step length

Thus from all these analysis it can be concluded that for energy efficient walk, SL should be as high as possible and LL should be low if it walks on even terrain. For 0.033 m LL, 0.200 m SL gives most energy efficient walks as its specific resistance is lowest. However, for different size of robot best combination will be differing.

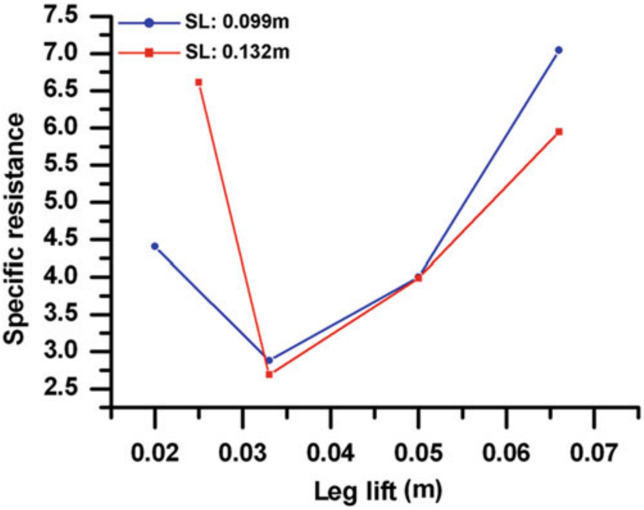


Fig. 14.43 Specific resistance for different leg lift with same step length

14.5 Conclusions

A three dimensional model of compliant legged quadruped robot using bond graph has been developed. The legs contain telescopic tubes with axial compliance and the model considers its detailed dynamics. The developed model is simulated for locomotion control in joint space as well as in workspace, and the obtained simulation and animation results are compared with the experimental results for validation. The developed model is useful for the various research activities pertaining to quadruped control. Its versatility is demonstrated by performance analysis, which shows that the compliant legged robot (specific resistance 0.925) is more energy efficient than rigid legged robot (specific resistance 1.10), a dynamically stable trot gait (specific resistance 0.925) is more energy efficient than a statically stable amble gait (specific resistance 6.71), and a foot trajectory consisting of maximum possible step length with minimum leg lift gives energy efficient performance. Influence of compliance on locomotion parameters is studied and best suitable compliance value, i.e., 5000 N/m for the robot is obtained. Turning motion of the robot is demonstrated by differential leg tip velocity of robot. In future, it is envisaged to use this model for workspace control, posture control, and fault accommodation of the quadruped robot.

Acknowledgment The work of M. M. Gor, P. M. Pathak, and A. K. Samantaray has been funded by DST, India under Indo–Korea Joint Research in Science and Technology vide Grant No. INT/Korea/P–13. The work of J.–M. Yang and S. W. Kwak was supported by the Korea government (MEST) vide the National Research Foundation of Korea Grant No. NRF–2011–0027705. M. M. Gor is thankful to G H Patel College of Engineering and Technology, Gujarat, India for allowing him to carry out research work at IIT Roorkee. Authors are thankful to B.Tech. students Divye, Kamran, Piyush, Param, and Rohan for their assistantship in developing the set-up and conducting the experiments.

References

1. Alexander, R. M. N. (1984). The gaits of bipedal and quadrupedal animals. *The International Journal of Robotics Research, 3*(2), 49–59.
2. Anderson, K. S. (1990). *Recursive derivation of explicit equations of motion for efficient dynamic/control simulation of large multibody systems*. Ph.D. dissertation, Stanford University, Stanford, CA.
3. Armstrong, W. W. (1979). Recursive solution of the equations of motion of an n-link manipulator. In *Proceedings of the Fifth World Congress on the Theory of Machines and Mechanisms* (Vol. 2, pp. 1342–1346). Montreal, Canada.
4. Bae, D. S., & Haug, E. J. (1987). A recursive formulation for constrained mechanical system dynamics: Part I. Open loop systems. *Mechanisms, Structures, and Machines, 15*(3), 359–382.
5. Banerjee, A. K. (1993). Block-diagonal equations for multibody elastodynamics with geometric stiffness and constraints. *Journal of Guidance, Control, and Dynamics, 16*(6), 1092–1100.
6. Bennani, M., & Giri, F. (1996). Dynamic modelling of a four legged robot. *Journal of Intelligent and Robotic Systems, 17*, 419–428.
7. Bera, T. K., Merzouki, R., Bouamama, B. O., & Samantaray, A. K. (2012). Force control in a parallel manipulator through virtual foundations". *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 226*(8), 1088–1106.
8. Bera, T. K., Samantaray, A. K., & Karmakar, R. (2009). Bond graph modeling of planar prismatic joint. In *Proceedings of the 14th National Conference Machines and Mechanisms (NaCoMM-09)* (pp. 22–26). Durgapur, India, December 17–18, 2009.
9. Bera, T. K., Samantaray, A. K., & Karmakar, R. (2012). Bond graph modeling of planar prismatic joints. *Mechanism and Machine Theory, 49*, 2–20.
10. Byl, K. (2008). *Metastable legged-robot locomotion*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, September.
11. Chen, X., Gao, F., Qi, C., & Zhao, X. (2013). Spring parameters design to increase the loading capability of a hydraulic quadruped robot. *Proceeding of the 2013 International conference on Advanced Mechatronics Systems* (pp. 535–540). Luoyang, China.
12. Craig, J. J. (2005). *Introduction to robotics mechanics and control* (3rd ed.). Upper Saddle River, NJ: Pearson Education.
13. Ersal, T., Fathy, H. K., & Stein, J. L. (2009). Structural simplification of modular bond-graph models based on junction inactivity. *Simulation Modelling Practice and Theory, 17*, 175–196.
14. Fukuoka, Y., & Kimura, H. (2009). Dynamic locomotion of a biomorphic quadruped 'Tekken' robot using various gaits: walk, trot, free-gait and bound. *Applied Bionics and Biomechanics, 6*, 63–71.
15. Gabrielli, G., & von Karman, T. (1950). What price speed? Specific power required for propulsion of vehicles. *Mechanical Engineering, 72*(10), 775–781.
16. Galloway, K. C., Clark, J. C., & Koditschek, D. E. (2013). Variable stiffness legs for robust, efficient, and stable dynamic running. *Journal of Mechanisms and Robotics*, *5*, 011009(1–11).
17. Ganesh, K., & Pathak, P. M. (2013). Dynamic modelling and simulation of a four legged jumping robot with compliant legs. *Robotics and Autonomous Systems, 61*, 221–228.
18. Ghosh, A. K., Mukherjee, A., & Faruqi, M. A. (1991). Computation of driving efforts for mechanisms and robots using bond graphs. *Transactions of the ASME Journal of Dynamic, Systems, Measurement, and Control, 113*(4), 744–748.
19. Gor, M. M., Pathak, P. M., Samantaray, A. K., Yang, J.-M., & Kwak, S. W. (2013). Dynamic modeling and simulation of compliant legged quadruped robot. In *Proceedings of the 1st International and 16th National Conference on Machines and Mechanisms (iNaCoMM2013)* (pp. 7–16). IIT Roorkee, India, 18–20 December 2013.
20. Gor, M. M., Pathak, P. M., Samantaray, A. K., Yang, J.-M., & Kwak, S. W. (2015). Control of compliant legged quadruped robots in the workspace. *Simulation: Transactions of the Society for Modeling and Simulation International, 91*(2), 103–125.

21. Gor, M. M., Pathak, P. M., Samantaray, A. K., Yang, J.-M., & Kwak, S. W. (2015). Control oriented model-based simulation and experimental studies on a compliant legged quadruped robot. *Robotics and Autonomous System, 72*, 217–234.
22. Gregorio, P., Ahmadi, M., & Buehler, M. (1997). Design, control, and energetics of an electrically actuated legged robot. *IEEE Transactions on Systems, Man and Cybernetics, 27*(4), 626–634.
23. Hollerbach, J. M. (1980). A recursive Lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity. *IEEE Transactions on Systems, Man, and Cybernetics, 10*(11), 730–736.
24. Karnopp, D. C., Margolis, D. L., & Rosenberg, R. C. (2006). *System dynamics: Modeling and simulation of mechatronic systems* (4th ed.). New York: Wiley.
25. Kimura, H., & Fukuoka, Y. (2004). Biologically inspired adaptive dynamic walking in outdoor environment using a self-contained quadruped robot: 'Tekken2'. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004, (IROS 2004)* (pp. 986–991).
26. Krishnan, V. L., Pathak, P. M., Sardana, L., & Jain, S. C. (2010). Simulation and experimental studies on walking robot with flexible legs. In *Proceeding of 4th International Conference on Integrated Modeling and Analysis in Applied Control and Automation (IMAACA 2010)*, Fes, Morocco (pp. 11–19).
27. Mahapatra, A., & Roy, S. S. (2009). Computer aided dynamic simulation of six legged robot. *International Journal of Recent Trends in Engineering, 2*(2), 146–151.
28. Merzouki, R., Samantaray, A. K., Pathak, P. M., & OuldBouamama, B. (2013). *Intelligent mechatronic systems—Modeling, control and diagnosis*. London: Springer.
29. Pathak, P. M., Mukherjee, A., & Dasgupta, A. (2005). Impedance control of space robots using passive degrees of freedom in controller domain. *Transactions of ASME, Journal of Dynamic Systems, Measurement, and Control, 127*(4), 564–578.
30. Raibert, M. H. (1986). *Legged robots that balance*. Cambridge, MA: MIT Press.
31. Raibert, M., Blankespoor, K., Nelson, G., & Playter, R. (2008). BigDog, the rough terrain quadruped robot. *Proceedings of the 17th IFAC World Congress, COEX* (pp. 10823–10825). South Korea.
32. Remy, C. D., Hutter, M., Hoepflinger, M., Bloesch, M., Gehring, C., & Siegwart, R. (2012). Quadrupedal robots with stiff and compliant actuation. *at-Automatisierungstechnik, 60*(11), 682–691.
33. Rosenthal, D. E. (1990). An order n formulation for robotic systems. *Journal of Astronautical Sciences, 38*(4), 511–529.
34. Samantaray, A. K., & Mukherjee, A. (2000). *User manual of SYMBOLS*. Kharagpur: High Tech Consultants, S.T.E.P., Indian Institute of Technology.
35. Semini, C., Tsagarakis, N. G., Guglielmino, E., Focchi, M., Cannella, F., & Caldwell, D. G. (2011). Design of HyQ—A hydraulically and electrically actuated quadruped robot. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 225*(6), 831–849.
36. Shah, S. V., Saha, S. K., & Dutt, J. K. (2012). Modular framework for dynamic modeling and analyses of legged robots. *Mechanism and Machine Theory, 49*, 234–255.
37. Shkolnik, A., Levashov, M., Manchester, I. R., & Tedrake, R. (2011). Bounding on rough terrain with the LittleDog robot. *The International Journal of Robotics Research, 30*, 192–215.
38. Siciliano, B., & Khatib, O. (Eds.). (2008). *Springer handbook of robotics*. Berlin: Springer.
39. Soyguder, S., & Ali, H. (2012). Computer simulation and dynamic modeling of a quadrupedal pronking gait robot with SLIP model. *Computers and Electrical Engineering, 28*, 161–174.
40. Sprowitz, A., Tuleu, A., Vespignani, M., Ajallooeian, M., Badri, E., & Ijspeert, A. J. (2013). Towards dynamic trot gait locomotion design control and experiments with cheetah-cub, a compliant quadruped robot. *The International Journal of Robotics Research, 32*(8), 932–950.
41. Todd, D. J. (1985). *Walking machines: An introduction to legged robots*. London: Koran Page.
42. Vukobratovic, M., & Jurwicic, D. (1969). Contribution to the synthesis of biped gait. *IEEE Transactions on Bio-Medical Engineering, 16*, 1–6.

43. Walker, M. W., & Orin, D. E. (1982). Efficient dynamic computer simulation of robotic mechanisms. *Journal of Dynamic Systems, Measurements, and Control, 104*, 205–211.
44. Zhang, S., Gao, J., Duan, X., Li, H., Yu, Z., Chen, X., et al. (2013). Trot pattern generation for quadruped robot based on the ZMP stability margin. *ICME International Conference on Complex Medical Engineering* (pp. 608–613). Beijing, China, May 25–28, 2013.

Chapter 15
Modeling and Control of a Wind Turbine

R. Tapia Sánchez and A. Medina Rios

Abbreviations

DFIG	Doubly fed induction generator
MSC	Machine side converter
NSC	Network side converter
DC	Direct current
Cp	Power coefficient
BEM	Blade element momentum
DFIM	Doubly fed induction motor

15.1 Introduction

Wind turbine is a complex system in which different technical areas are involved. Some publications deal with this topic, e.g., [22, 24, 25]. In [22] a parametric and nonparametric model is proposed using advantage algorithms, only the power curve is considered. Other contribution [25] reviews recent research of numerical simulation applied to wind energy and in [24] the modeling of a small-scale distributed power system containing the power demand, a wind turbine, the photovoltaic arrays, and the electrical connection is presented. These contributions highlight the importance of having a reliable model of the wind turbine, in order to conduct dynamic studies of such system. Nevertheless, the use of different techniques previously developed, and the complexity of a wind turbine make

R.T. Sánchez (✉) • A.M. Rios
División de Estudios de Posgrado, Facultad de Ingeniería Eléctrica, Universidad Michoacana de San Nicolás de Hidalgo, Morelia 58000, Mexico
e-mail: rtsanchez@dep.fie.umich.mx

W. Borutzky (ed.), *Bond Graphs for Modelling, Control and Fault Diagnosis of Engineering Systems*, DOI 10.1007/978-3-319-47434-2_15

possible to visualize even different approaches for their study and analysis. Under this context, in order to analyze the system in the same reference frame, the bond graph methodology [3, 19, 37] can represent the whole structure. This methodology presents some properties that can be applied directly to the model [7].

The bond graph wind turbine model has been addressed in several works, e.g., [1, 10, 46]. In the first publication, a detailed model of a blade is proposed. The aerodynamic structural forces are considered, and also real data of a wind turbine are used, in order to calculate the output torque. Besides, it is a general model that can be applied to any wind turbine blade. In [46] a six-mass drive train model is presented. This complete model is formulated and then simplified. The authors did not use real parameters of a wind turbine and the aerodynamics forces are not considered. In [10], a complete wind turbine based on parameters taken from a real turbine is proposed. The model presents all components of the wind turbine, but the aerodynamics are not considered in detail. The publication is centered on drive train effects.

The mechanical part of the wind turbine presented in this chapter is taken from [44] which consider the blade model presented in [1].

Nowadays, most of the installed wind turbines have kept the configuration with the use of a gearbox. Based on this context, a model of the gearbox is used in order to complete the whole system. In terms of a bond graph methodology, gears have been modeled for transmission applications [6, 8]. The model used here is based on the one reported in [21], in which a planetary gears is adopted. Also, in [17] a complete review of graphical tools for modeling gears is given. The authors conclude that bond graph methodology has major properties, when compared to others. It is important to mention that gearbox is the most important part of a wind turbine, because is at this stage where most of faults occur. It is estimated that a wind turbine has around 20 years of life span, but normally the gearbox needs to be replaced every 5 years [36].

In this chapter a complete bond graph model of a variable-speed wind turbine with doubly fed induction generator (DFIG) configuration is presented, which accounts real data and parameters of a real wind turbine of 750 kW [13]. This configuration has some important advantages, e.g., the possibility of controlling active and reactive power, the capability of reducing stresses of the mechanical structure, and acoustic noise [5]. Also, losses in the power electronics converter are reduced, as compared to a direct-driver synchronous/induction generator (variable-speed wind turbine with induction generator configuration). This is due to the fact that the converter placed between the grid and the double-fed induction machine rotor handles only a fraction of the turbine rate power [26, 47].

The control of a DFIG has been addressed in several works, e.g., [9, 15, 32, 35]. In [9] a sliding mode control is used, and in [35] the power control of a doubly fed induction machine via output feedback is presented. The behavior of such machines in large wind farms, along with the general active and reactive power control has been addressed in [15]. A novel simplified model of the DFIG appropriate for bulk power system studies is presented in [32]. In this chapter, the control law developed in [45] is taken because this control presents a different structure and the bond graph model of the DFIG is used in order to obtain the control law applying the bicausality concepts [11].

The outline of this chapter is as follows: the model description is given in Sect. 15.2. Then, the wind turbine model is detailed in Sect. 15.3. The DFIG control is described in Sect. 15.4. The whole system is simulated in Sect. 15.5. The global conclusions of the chapter are drawn in Sect. 15.6.

15.2 System Description

The wind turbine connected to the electrical network can be represented by Fig. 15.1. Every single part (blades, hub, bearings, brake, and hub) of the wind turbine scheme is joined by a bond. In most of the elements there is an associated mechanical power, but the last part (DFIG) transforms this mechanical power into a three phase electrical power, represented by three bonds. The stator is directly connected to the power network, while the rotor is connected through slip rings to a power electronic converter.

Different considerations can be made for the study of this system. Here, an analysis of the complete model is presented, by considering all the components as shown in Fig. 15.1.

The function of blades is to convert the wind velocity into a torque, transforming the wind into a force and then into a torque. In Fig. 15.1, the causality of the first bond shows that torque is provided by the blades.

A large torque is produced by blades, and is transferred to the gearbox, which transforms this big torque into a small one, to be applied into generator. A similar situation is valid for angular velocity, i.e., there is a high speed provided for generator, which is transformed by the gearbox into a low speed, presented in the rotor turbine (hub).

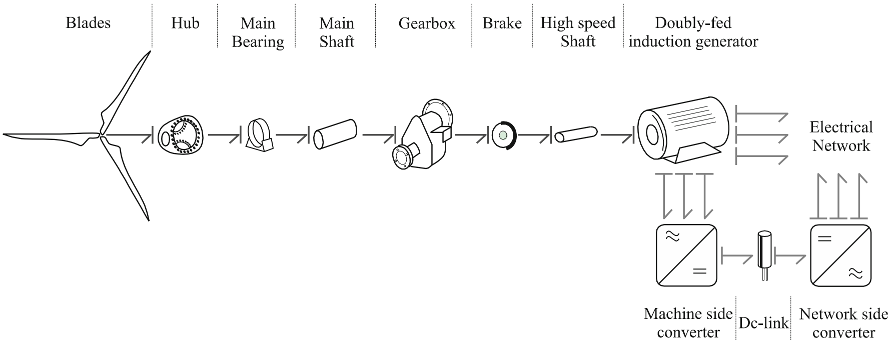


Fig. 15.1 Wind turbine presented in terms of word bond graph

From the generator point of view, a wind turbine has different configurations. Also, the wind turbine can operate with either under fixed-speed or variable-speed mode. This operation depends directly on the generator connection. It means that for fixed-speed wind turbines, the generator is directly connected to the power network, since the speed is closely tied to the grid frequency. Besides, for a variable-speed wind turbine, the generator is controlled through power electronics converters, which make possible to control the rotor speed. Therefore, as it was mentioned before, a variable-speed wind turbine with doubly fed generator configuration is used (Fig. 15.1).

The main advantage of this configuration is the fact that the power electronic converter has to handle only a fraction (30 %) of the total power [47]. Therefore, the losses in the power electronic converter can be reduced.

Two power electronics converters: machine side converter (MSC) and network side converter (NSC) are used in order to have a DC-link between them, thus allowing the power transfer. With the MSC, it is possible to control the torque or speed in the DFIG and the power factor at the stator terminals, while with the NSC functions the DC-link voltage is kept constant.

In the following sections the model of each elements is presented.

15.3 Model Components

In this section the individual bond graph models for the blades, the gearbox, the doubly fed induction generator, and the converters are presented.

15.3.1 Blades

As mentioned before, the blade model is taken from [1]. In that publication, the blade structure represents a general model; it means that wind turbines of different powers capacity can be simulated by changing the data. The model presented in [1] is divided into three sections; however, it can be divided into more sections. Here this model is recalled, since it represents the blade aerodynamic wind turbine. Nevertheless, the blade is divided into only two sections to allow the improvement of results, in terms of the coefficient of power Cp, as it will be later shown in this section.

There are problems in system modeling, where lumping of inertia and compliance, used in rigid body dynamics, fails to get the essential dynamics of the system. Situations like this often arise with systems consisting of long slender members, whose flexibility plays an important role in the dynamics of the system. For example, in wind turbine modeling, it will be inadequate to treat the blade as a rigid body. These members are essentially distributed system parameters, governed by partial differential equations, and are lumped in space by finite approximation.

The model is based on Rayleigh Beam Model [29] and the Blade Element Momentum (BEM) theory [16]; it is a dynamic model in which a modal analysis can be made. Then, the blade *is formed by* two main parts: the structural and the aerodynamic.

15.3.1.1 Model Structure

The structure of the blade is made using C-field and R-field elements, which represent the stiffness and the structural damping matrix between the center of gravity of adjacent elements, respectively. These matrices are calculated at the center of gravity of adjacent elements. Figure 15.2a shows a schematic of a cantilever beam with regulation and lumping of inertia, which is used in the bond graph formulation.

As it is shown, the whole blade structure is divided into two sections. The length of each section is the same, 11.7 m, thus allowing a blade of 23.4 m, which is the characteristic blades length for a 750 kW wind turbine.

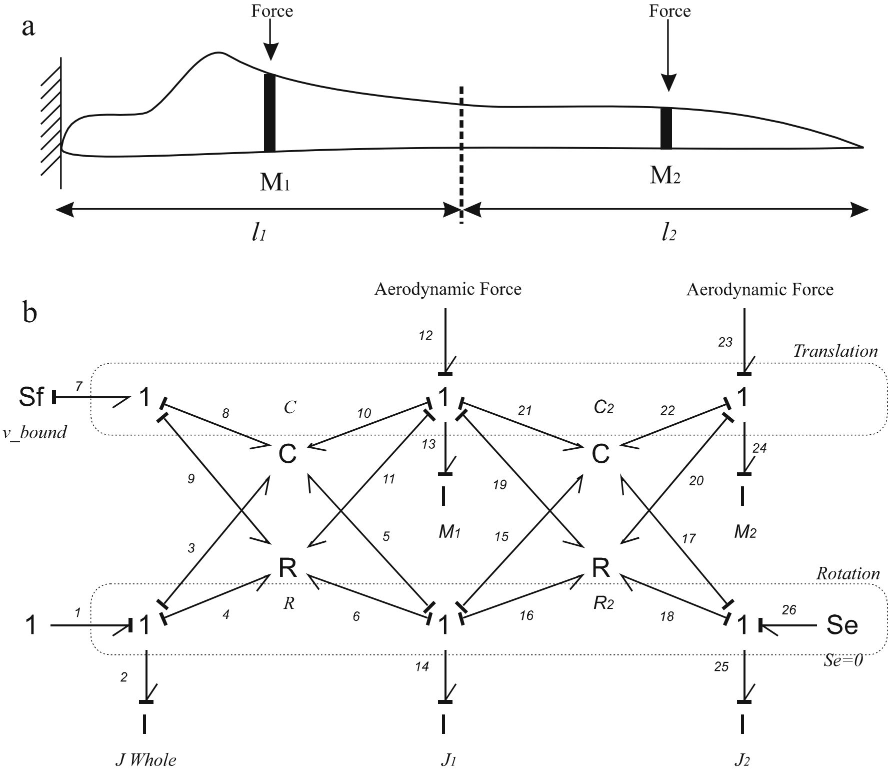


Fig. 15.2 (**a**) Cantilever beam of blade and (**b**) structural bond graph of blade

The bond graph model presented in Fig. 15.2b shows the translation and rotation motion at the top and bottom of the figure, respectively. Aerodynamic forces for each section are applied in the translation structure (bonds 12 and 23); this means that the effort is imposed.

The stiffness matrix is modeled as a 4–port C–field storing, due to the four generalized displacements shown in Fig. 15.2b. In terms of flexural rigidity (EI) and element length (l), the stiffness matrix is given by Eq. (15.1).

$$[K_i] = \frac{EI}{l^3}\begin{bmatrix} 12 & 6l & -12 & 6l \\ 6l & 4l^2 & -6l & 2l^2 \\ -12 & -6l & 12 & -6l \\ 6l & 2l^2 & -6l & 4l^2 \end{bmatrix} \tag{15.1}$$

where E is the Young module of the material, and I the second moment of area about the axis of deflection. The structural damping matrix is given by $[R_i] = \mu[K_i]$, where the μ factor represents the coefficient of structural damping.

The boundary condition of the model is represented by the Sf:v_bound and Se sources. The connection between the blade and the hub is assumed to be rigid, i.e., Sf:v_bound = 0, and the blade has only one degree of freedom Se = 0. Finally, the rotating inertia J_whole is the rigid body inertia of the whole blade.

Dynamic equations and also the natural frequencies of the blade can be directly obtained from the bond graph of Fig. 15.2b. In this case, the main dynamics around the blade structure take into account the parameters (given in Appendix (Table 15.6)) for an NACA 4415 blade [48].

Considering Fig. 15.2b the causal loops and causal paths are presented in Table 15.1. The causal loops consider the **C** and **R** matrix values, in order to calculate the frequency. Therefore, each causal loop involved with one I element and one element of the damping matrix (**R**) is considered. Besides, there are causal paths present in the model, between the I-elements and the stiffness matrix, and Table 15.1 shows also the numerical values of these causal paths. It is important to mention that the main frequencies are produced only by the causal loops.

As shown in Table 15.1, the loop involving the translational mass M_1 of section 1 has small values (bonds 11, 13, 19), compared to the mass M_2 (bonds 20, 24). This verified that high frequency vibrations are introduced by the last blade sections, and

Table 15.1 Causal loop and causal paths frequencies

Causal loop	2 → 4 → 4 → 2	14 → 6 → 6 → 14	14 → 16 → 16 → 14	25 → 18 → 18 → 25	13 → 11 → 11 → 13	13 → 19 → 19 → 13	24 → 20 → 20 → 24
Frequency (rad/s)	120.88	36632.9	54245	76827.7	8.77	12.99	32.22
Causal path	2 → 3	14 → 5	13 → 10	14 → 15	13 → 21	25 → 17	24 → 22
Frequency (rad/s)	690832.9	39685.3	224807.7	39685.33	224807.8	9873.15	482100.3

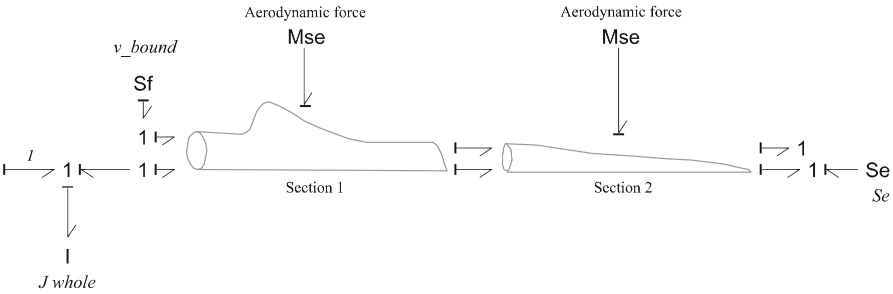


Fig. 15.3 Two sections (sub-model) bond graph blade

low frequencies are presented in the first blade section (it is the same for the inertias J_1 and J_2). A similar behavior is presented for the causal paths; the high frequency is introduced by the last blade sections. It is important to say that these are not the real natural frequencies, they are only an approximation. If an exactly value is required, then a matrix structure needs to be found, taking into account all the causal loops and path of the whole structure.

The model of the blade can be represented by sub-models, as shown in Fig. 15.3. The rotating inertia *J* whole presents a derivative causality because the angular velocity is given by bond 1. This means that the blade gives the torque as output and then it needs the angular velocity as input. Besides, bond 1 makes it possible to have a connection with the hub.

The two modulate sources of effort, which represent the aerodynamic force, need to be calculated from the input wind. This means that the wind speed needs to be converted into aerodynamical force. This process will be presented in the next subsection.

15.3.1.2 Aerodynamic Force

The aerodynamic loading is caused by the flow (wind) past the structure, in other words the blades and the tower. Accurate models of the aerodynamic aspects of wind turbines are essential to successfully design and analyze wind energy systems. Wind turbine aerodynamic models are used to relate wind inflow conditions to loads applied to the turbine.

The subsequent analysis develops the most common aerodynamics theory employed in the wind turbine design and analysis environment. It consists on the blade element momentum theory (BEM), the fundamental aerodynamic theory used by the bond graph model presented in this paper.

The whole BEM theory can be found in [27]. Here, the used mathematical expressions will be recalled, as well as the principal expressions taken from [1], which are necessary to convert the wind into an aerodynamic force.

As it was explained before, blade structure is divided into sections. For each section BEM theory is applied, in order to provide aerodynamic force to the blade structure. Equation (15.2) expresses the aerodynamic force F_i applied to the ith section.

$$F_i = \left(\frac{1}{2} \rho V_w \frac{(1-a_i)^2}{\sin^2 \varphi_i} \left(C_{li} \sin \varphi_i - C_{di} \cos \varphi_i \right) c_i l_i \right) V_w \tag{15.2}$$

where V_w represents the wind velocity, ρ the air density, φ_i the wind inflow angle (Eq. (15.3)), C_{li} and C_{di} are the lift and drag dimensionless coefficients function on the angle of attack α_i, defined as the angle between the incoming flow stream and the chord line of the airfoil in the ith section. Also, a_i represents the axial tangential induction factor and is calculated from Eqs. (15.4) and (15.5).

$$\varphi_i = \tan^{-1} \left(\frac{V_w (1-a_i)}{\Omega_r r_i (1+a_i')} \right) \tag{15.3}$$

$$a_i = \left(1 + \frac{4 \sin^2 \varphi_i}{\sigma_1' \left(C_{li} \cos \varphi_i + C_{di} \sin \varphi_i \right)} \right)^{-1} \tag{15.4}$$

$$a_i' = \left(-1 + \frac{4 \sin^2 \varphi_i}{\sigma_1' \left(C_{li} \cos \varphi_i - C_{di} \sin \varphi_i \right)} \right)^{-1} \tag{15.5}$$

Parameters involving in last expressions are graphically represented in Fig. 15.4.

A modulated gyrator MGY element is used to implement Eqs. (15.2), (15.3), (15.4), and (15.5), since wind is transformed into an aerodynamic force, as shown in Fig. 15.5.

Modulated inputs to MGY elements of Fig. 15.5 are the pitch angle and the angular velocity Ωr.

In order to simulate the blade model, Eqs. (15.2), (15.3), (15.4), and (15.5) are placed within each MGY element; it means that their traditionally constitutive relation is changed. These expressions represent an iterative process, where the axial tangential induction factor and the wind inflow angle are changed. The code

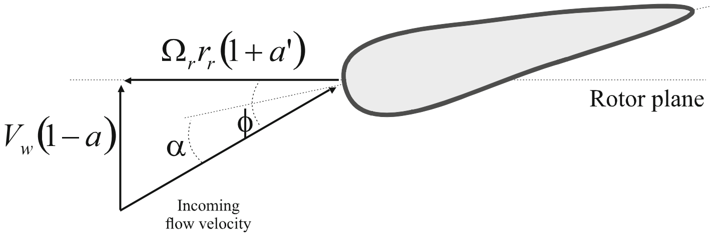
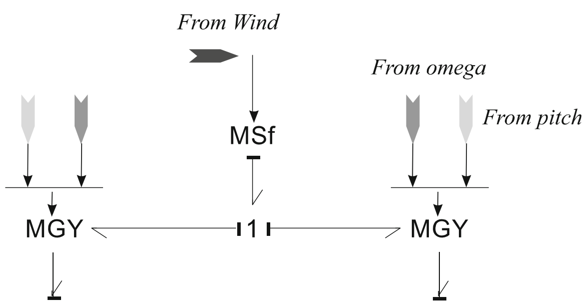


Fig. 15.4 Velocities at rotor blade

Fig. 15.5 Representation of aerodynamic force conversion



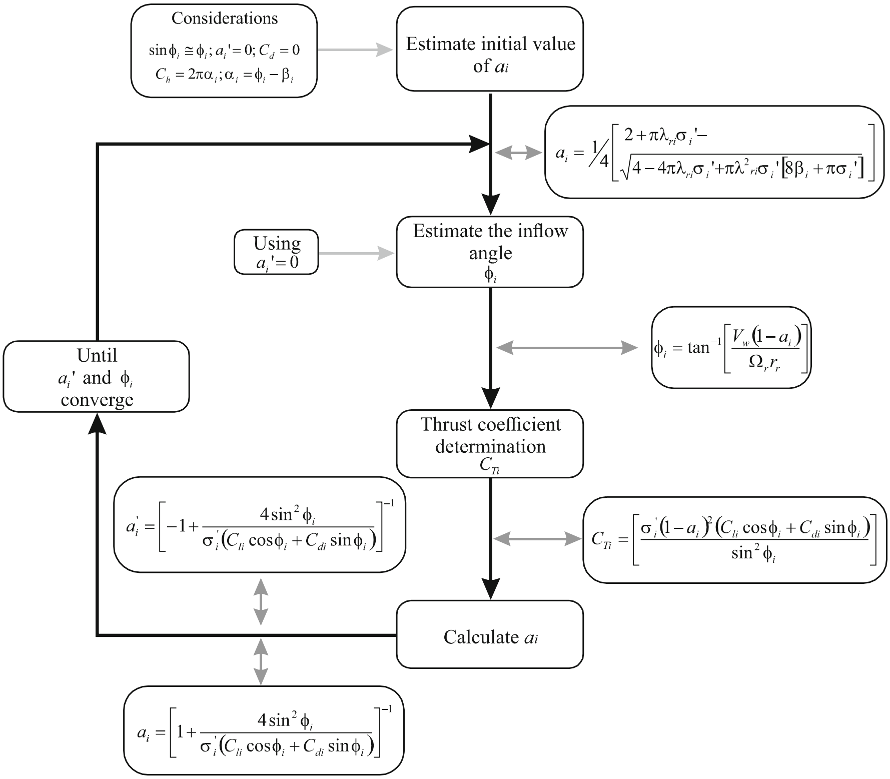

Considerations
$\sin\phi_i \cong \phi_i;\ a_i' = 0;\ C_d = 0$
$C_h = 2\pi\alpha_i;\ \alpha_i = \phi_i - \beta_i$
Estimate initial value of a_i
$a_i = \frac{1}{4}\left[2 + \pi\lambda_{ri}\sigma_i' - \sqrt{4 - 4\pi\lambda_{ri}\sigma_i' + \pi\lambda_{ri}^2\sigma_i'\left[8\beta_i + \pi\sigma_i'\right]}\right]$
Using $a_i' = 0$
Estimate the inflow angle ϕ_i
$\phi_i = \tan^{-1}\left[\frac{V_w(1-a_i)}{\Omega_r r_r}\right]$
Until a_i' and ϕ_i converge
Thrust coefficient determination C_{Ti}
$C_{Ti} = \left[\frac{\sigma_i'(1-a_i)^2(C_{li}\cos\phi_i + C_{di}\sin\phi_i)}{\sin^2\phi_i}\right]$
$a_i' = \left[-1 + \frac{4\sin^2\phi_i}{\sigma_i'(C_{li}\cos\phi_i + C_{di}\sin\phi_i)}\right]^{-1}$
Calculate a_i
$a_i = \left[1 + \frac{4\sin^2\phi_i}{\sigma_i'(C_{li}\cos\phi_i + C_{di}\sin\phi_i)}\right]^{-1}$

Fig. 15.6 Flow diagram code

flow diagram is shown in Fig. 15.6. In order to estimate the initial value of the axial induction factor, some conditions are assumed at the starting time; then, an estimation of the inflow angle is made. This allows to determinate the accurate coefficient [4]. After that, the calculation of the axial and tangential induction factor is performed and the new value is used for the next cycle. This iterative process is conducted until the tangential induction factor and wind inflow angle have converged to their final values.

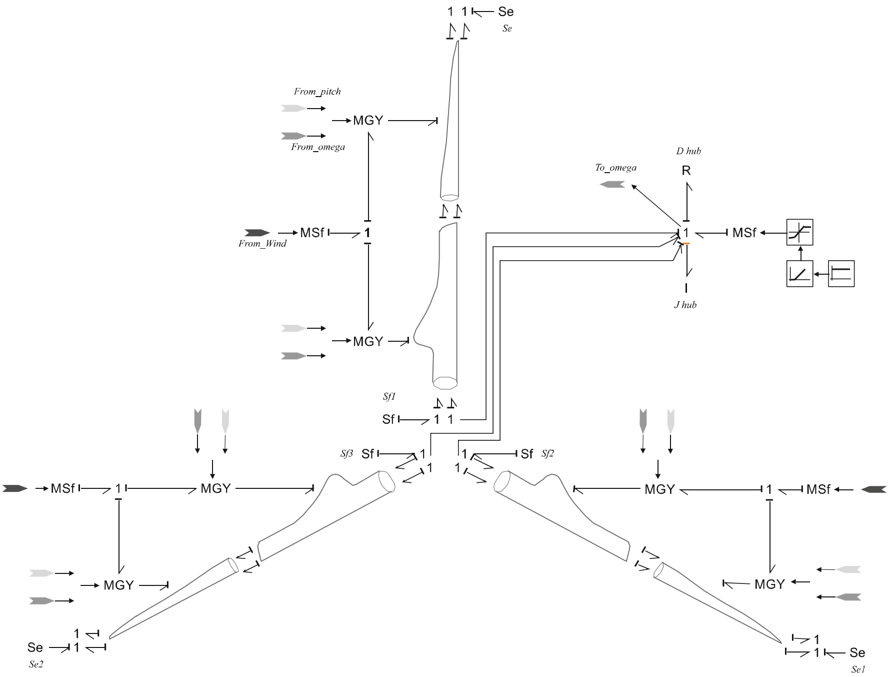


Fig. 15.7 Three blades and hub model

Figure 15.6 also shows the mathematical expressions involved in each stage.

It is important to observe that, due to the complexity involving in the aerodynamic force conversion, the use of the MGY element was necessary. Nevertheless, it can be possible to model this part in terms of bond graph allowing having a physical behavior of this conversion.

A simulation is carried out in order to verify the dynamic response of blades; this is made using the 20Sim software (www.20sim.com). Figure 15.7 shows the simulation scheme.

Three blades are used; each of them is divided into two sections with the forces from the wind in each section applied. The rotating inertia J_{hub} corresponds to the hub structure and represents the rotating inertia of each blade and the hub itself; it means that $J_{\text{hub}} = 3{*}J_{\text{whole}} + J_{\text{hub}}$ itself. Besides, the R element called D_{hub} represents the main bearing.

To observe the transient response in the modulated flow source MSf the angular velocity applied with a ramp profile is used. The traditional curve of power coefficient versus tip speed (Cp vs λ) is shown in Fig. 15.8.

In Fig. 15.8, the maximum Cp obtained is 0.4, having a tip speed of 7.4; the theoretical maximum value of Cp is 0.59 (due to Betz limit), at around a tip speed of 8. In practical designs, the maximum achievable Cp is below 0.5 for high speed, two-blade turbines, and between 0.2 and 0.4 for slow speed turbines with more

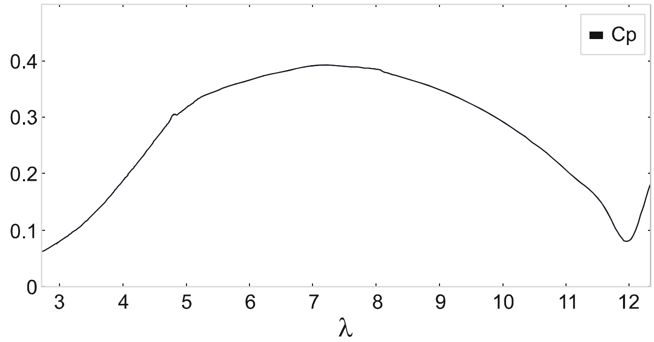


Fig. 15.8 Curve of Cp vs λ

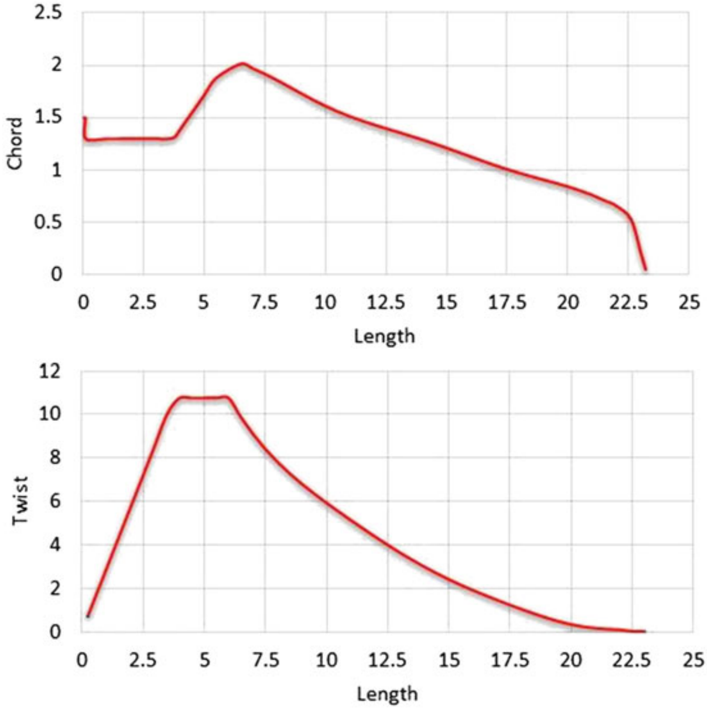


Fig. 15.9 Chord and twist along the blade

blades [33]. The approximation to standard values is acceptable and it is better that the value shown in [1] (around 0.33). The improvement results from taking two sections of the blade allow applying the aerodynamic force with maximum chore and twisting distribution (Fig. 15.9) along the blade length.

Figure 15.9 shows the chord and twist along the blade. By the action of twisting, the wind comes with a large angle of incidence as it approaches the bottom of blade.

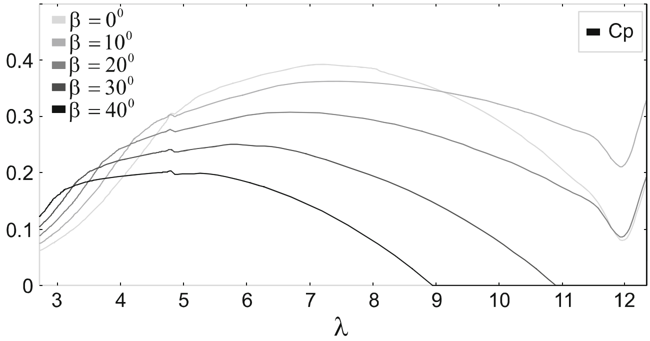


Fig. 15.10 Curve Cp vs λ for different pitch angles

Then, the chord is the distance between the leading edge and the trailing edge of the blade section. According to these two figures, the values of chore and twist taken for each section are presented in the Appendix.

It is known that pitch angle is another important parameter in wind turbines, since the power generated by the turbine needs to be regulated with this parameter. Figure 15.10 shows the Cp coefficient for different pitch angle values, and confirms that dynamical model of the blade + hub + main bearing adequately performs. This is because when the pitch angle increases, the mechanical power decreases.

15.3.2 Gearbox

A planetary stage and two more parallel stages (to increase the angular velocity) normally compose the gearbox of a wind turbine. Figure 15.11 shows the gearbox scheme.

Bond graph models of gears have been addressed in previous contributions [6, 8]. The simplest way to model the gearbox is as a whole structure; it means that conversion ratio (N_p/N_g, where N_p is the teeth number of pinion and N_g the teeth number of gears) can be introduced directly in a transformer element TF. However, as the gearbox is one of the most important parts of a wind turbine due to its involved dynamics; thus, a more precise model needs to be developed.

In [17, 21] a detailed gearbox bond graph model is proposed. In [17] the authors detail the technique, making a comparison of different methodologies in a gear analysis, and in [21] a planetary gear system is presented. The last publication is taken as a reference for the wind turbine gearbox model.

Then, in the gearbox model some considerations are taken into account, i.e.,

- Ring gear is a fixed stage.
- Only one parallel stage is considered (including the two parallel stage effects).
- Carrier is considered as input; however, sun is the output.

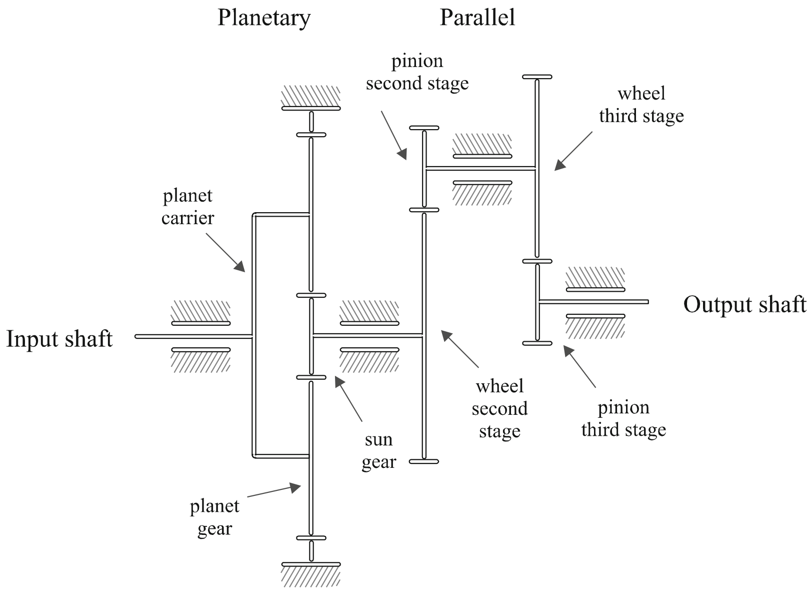


Fig. 15.11 Gearbox schema

- Mesh stiffness between planet-ring and planet-sun is considered.
- Bearing stiffness are not considered in the model.

15.3.2.1 Analysis of the Gearbox Model

The bond graph model formulation is made taking as a reference (Fig. 15.11), and the consideration above mentioned. Figure 15.12 shows the gearbox bond graph model.

The model of Fig. 15.12 shows the three planet gears represented by momentum of inertias J_{pj} ($j = 1, 2, 3$). These planets are bounded to sun and ring gears by the mesh stiffness K_{sp} and K_{rp}, respectively. Also, there is a relationship between mesh stiffness and carrier gear. As it is shown, the momentum of inertia J_r (ring gear) has a derivative causality; this is due to a fixed ring stage consideration; it means that a $Sf = 0$ source is bounded (at the 1_5 junction) in order to impose a non-angular velocity at this stage of planetary gearbox. In the model, Z_i ($i = \mathrm{p, s, r}$) represents the teeth number of each gear. The sun output is bonded to TF element, which represents the two parallel stages output. This model allows a design of the gearbox, using the parameter estimation concept developed in [30].

The flow junction $1_{1\to 3}$, 1_4, 1_5, 1_6 represents the angular velocity of planets, carrier, ring, and sun gear, respectively. They are all related to each other by TF elements.

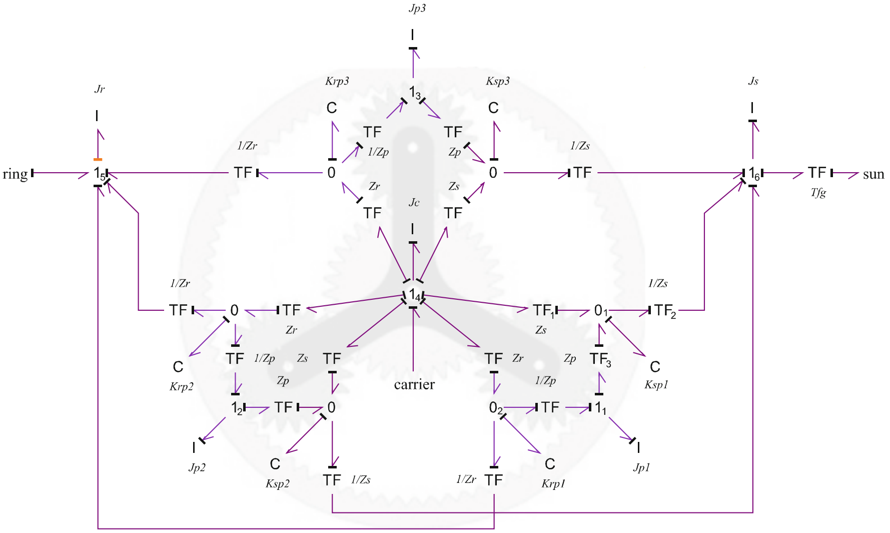


Fig. 15.12 Gearbox model

Considering the planet 1 (see Fig. 15.12), it is shown that there is a zero junction 0_1, between the transformer TF_1, TF_2, and TF_3, which represents (effort variable) that the planet moves in tangential direction. In this junction, the mesh stiffness between the planet and sun is joined. Transformers TF_1:Z_s, TF_2:1/Z_s, and TF_3:Z_p allow to obtain the rotating speed of the sun gear, the linear velocity of the planet revolution around the sun gear, and the planet autorotation. This structure is the same for the zero junction 0_2, but considering the relationship between the planet, the ring, and the carrier.

If a more precise model of a gearbox is required, the design can be done as it was made for the blade model. This means that the gears can be taken as a structure, considering their mass and the translational displacement. Also, bearing stiffness and torsional support stiffness can be added to the model.

The main dynamics representation can be directly achieved. Opposite to blade structure, gearbox of Fig. 15.12 does not have causal's loops, since nonresistive elements (bearings) are considered. Thus, only a path loop between the inertial gears and the mesh stiffness can be found. Parameters (see Appendix) for a 750 kW wind turbine gearbox are used and have been taken from [13].

15.3.2.2 Gearbox Simulation

One simulation is carried out in order to verify the gearbox model. Table 15.2 shows the numerical values used for the simulation. Torque and angular velocity for sun and carrier are shown in Fig. 15.13.

Table 15.2 Gearbox numerical values

Gearbox
$J_s = 3.2$ kg.m^2, $J_r = 144.2$ kg.m^2, $J_c = 59.1$ kg.m^2, $J_p = 3.2$ kg.m^2
$Z_p = 39$, $Z_r = 99$, $Z_s = 21$, $K_{sp} = 16.9e9$ N/m, $K_{rp} = 19.2e9$ N/m, $T_{fg} = 10.5$

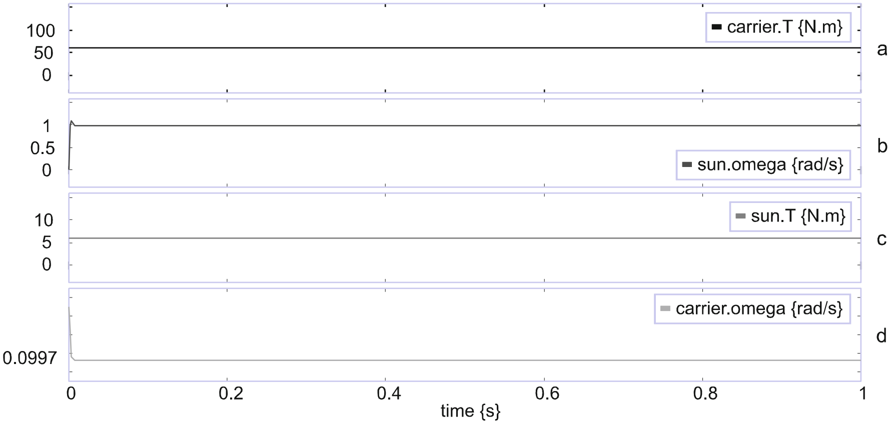


Fig. 15.13 Simulation gearbox (**a**) carrier torque, (**b**) sun torque, (**c**) sun angular velocity, and (**d**) carrier angular velocity

For the simulation, a constant carrier torque is applied (Fig. 15.13a). This torque is reduced at the sun output torque with a ratio of 60 (Fig. 15.13c). The same case is observed in the angular velocity curves (Fig. 15.13b, d). It is known that this type of system presents a fast vibration, due to the dynamics involved between the mesh stiffness and the momentum of inertias. In Fig. 15.13, the initial conditions (for the momentum of inertias) have been considered, in order to show the conversion ratio of the gearbox. This allows eliminating these vibrations when simulation starts. Figure 15.13 verifies that the complete gain of gearbox is 60. This is an acceptable value of the gearbox gain; since a value of 60 or 70 is expected for a wind turbine with these dimensions (750 kW).

15.3.3 Doubly Fed Induction Generator

Induction machines have been addressed in many publications. Models can be represented in two general frameworks: one using a Park reference frame [18, 38, 39] and the other one using the natural reference frame (three sinusoidal waveforms) [28].

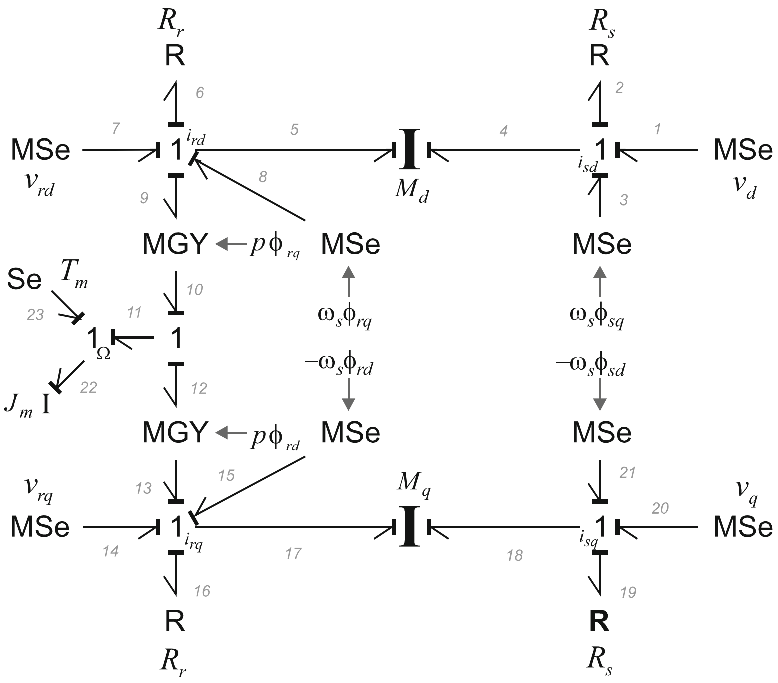


Fig. 15.14 DFIG 2-axis reference frame

15.3.3.1 Park Reference Frame Model

Figure 15.14 shows a bond graph model of the induction machine in the Park 2-axis *d-q* reference frame [38].

The induction machine model is based on the following assumptions:

- Magnetic hysteresis and magnetic saturation effects are neglected.
- The stator winding is sinusoidally distributed along the air-gap.
- The stator slots cause no appreciable variation of the rotor inductances with rotor position.

An arbitrary *dq-frame* rotating around the homopolar *0-axis* to the speed ω_s is chosen.

The equations that describe the bond graph model of Fig. 15.14 are given by Eq. (15.6).

$$\begin{aligned}
v_d &= R_s i_{sd} + \frac{\mathrm{d}\varphi_{sd}}{\mathrm{d}t} - \omega_s \varphi_{sq} \\
v_q &= R_s i_{sq} + \frac{\mathrm{d}\varphi_{sq}}{\mathrm{d}t} + \omega_s \varphi_{sd} \\
v_{rd} &= R_r i_{rd} + \frac{\mathrm{d}\varphi_{rd}}{\mathrm{d}t} - \omega_s \varphi_{rq} + p\Omega\varphi_{rq} \\
v_{rq} &= R_r i_{rq} + \frac{\mathrm{d}\varphi_{rq}}{\mathrm{d}t} + \omega_s \varphi_{rd} - p\Omega\varphi_{rd} \\
\tau_{11} &= p\left(\varphi_{rq} i_{rd} - \varphi_{rd} i_{rq}\right) \\
\tau_{11} &= J_m \frac{\mathrm{d}\Omega}{\mathrm{d}t} - T_m
\end{aligned} \tag{15.6}$$

The inductance matrices for I-field elements M_d and M_q allow coupling the stator and rotor fluxes by the matrix shown in Eq. (15.7).

$$\begin{pmatrix} \varphi_{sd/q} \\ \varphi_{rd/q} \end{pmatrix} = \begin{pmatrix} L_s & L_m \\ L_m & L_r \end{pmatrix} \begin{pmatrix} i_{sd/q} \\ i_{rd/q} \end{pmatrix} \tag{15.7}$$

where L_s, L_m, and L_r are the stator self-inductance, mutual inductance between the stator and rotor, and rotor self-inductance, respectively. I-field element J_m represents the shaft and rotor moment of inertia; R_s and R_r are stator and rotor resistances, p is the number of pole pairs, and ω corresponds to $2\pi f$, with f being the network frequency.

It is important to remark that the four modulated sources (dependent on ω_s) represent virtual sources (not physical sources), since their power sum is zero and are only a mathematical consequence of the model, as demonstrated in [18]. They are considered by the arbitrary framework assumption and can be removed if the stationary frame is chosen (setting *d-axis* with stator phase a) since $\omega_\text{s} = 0$. As the machine model is related to the rotor, the stator equations are not influenced by the rotor speed.

The model of the DFIG was developed by using the equivalent electric circuit of the induction machine shown in Fig. 15.15 [38].

15.3.3.2 Model Simplification

The methodology shown in [2] can be applied to the bond graph model of the induction machine, in order to reduce the order model from the energy metric point of view. However, since the application of the induction machine in the wind turbine (as a generator) does not involve a high rotor speed, it is assumed that the neglected mode cannot be excited.

In order to identify the dynamics that can be eliminated, the dynamics approximation made before with the blade structure is applied to the DFIG. Since the model has I-field multiports, the dynamics are obtained by using the methodology proposed in [20], which allows obtaining the causal loops and paths when a multiport is used.

The dynamic model can be simplified through an algebraic equivalent. This can be achieved if the dynamics are eliminated, the transient is cancelled and the order of the model is reduced. Thus, by setting $\dot{\varphi}_{sq}$ and $\dot{\varphi}_{sd}$ equal to zero, the model order is reduced from 5 to 3.

Regarding Fig. 15.14, the causality change is required in bonds 4 and 19. Figure 15.16 shows the stator *d/q-axis* simplified model of the induction machine.

Effort in bond 4 is zero, since the stator dynamics are eliminated. As the current i_sd is used to calculate the flux φ_sd (necessary for the MSe source along *q-axis*), the same structure is kept, and only the causality is changed. A similar process is made in bond 19.

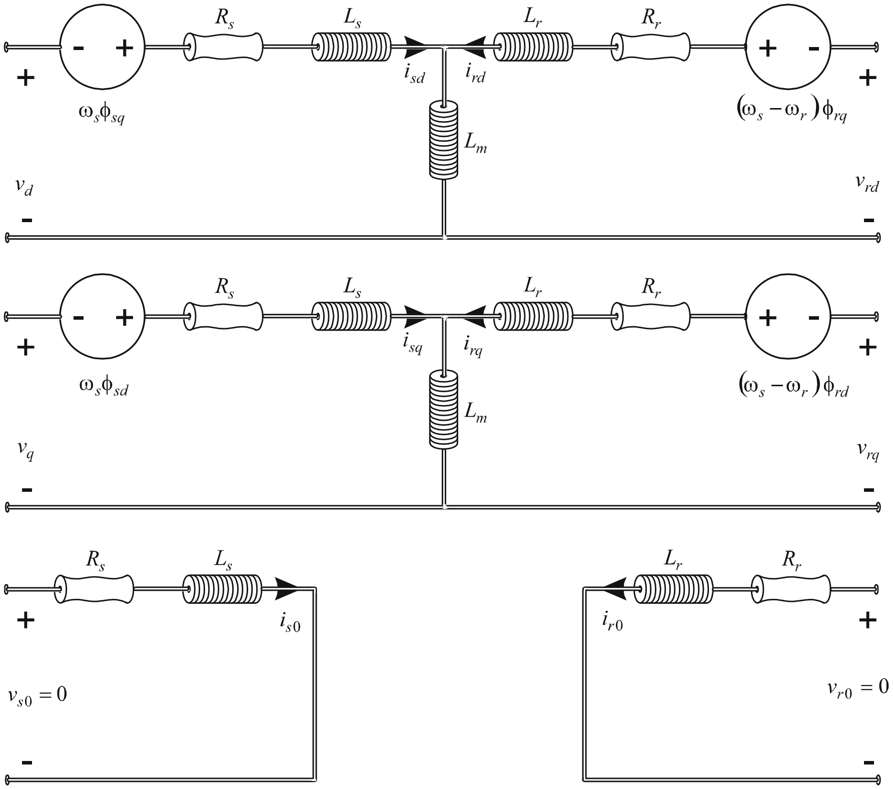


Fig. 15.15 Induction machine electrical circuit in dq_0 arbitrary frame

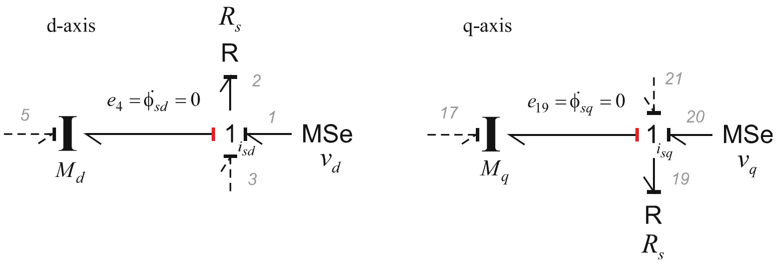


Fig. 15.16 Stator *d*, *q*-axis simplified

Table 15.3 Doubly fed induction machine parameters

Power	750 kW	Stator inductance	2.51 mH
Poles numbers	2	Rotor inductance	2.45 mH
Voltage	690 V	Mutual inductance	2.41 mH
Stator resistance	0.0022 Ω	Inertia	63.89 kg.m^2
Rotor resistance	0.0018 Ω	Frequency	60 Hz

Causality changes require a modification of (15.7). Matrix expressions in the I-field elements are changed to:

$$\begin{pmatrix} \varphi_{sd,q} \\ i_{rd,q} \end{pmatrix} = \begin{pmatrix} L_s - \frac{L_m^2}{L_r} & \frac{L_m}{L_r} \\ -\frac{L_m}{L_r} & \frac{1}{L_r} \end{pmatrix} \begin{pmatrix} i_{sd,q} \\ \varphi_{rd,q} \end{pmatrix} \tag{15.8}$$

Causality changes reduce from 5 to 3 the number of independent state variables in the model. The dynamic equation in the stator part is changed for a static one, and the effort is set equal to zero in these bonds.

15.3.3.3 DFIG Simulation

In order to compare the behavior of the simplified model (order 3) against the complete one (order 5), a simulation is carried out by considering the doubly fed induction machine as a motor (DFIM). Table 15.3 shows the numerical parameters for the machine.

The DFIM is started at instant $t = 0$ s, then at $t = 6.5$ s the nominal torque is applied ($T_{\text{nom}} = 3978.88$ Nm). After that, at $t = 8.0$ s a voltage reduction of 50 % during 200 ms is applied. Figure 15.17 shows the obtained results.

As observed from Fig. 15.17, the response obtained with both models is in close agreement. It is important to observe that oscillations are eliminated when the simplified model is used. For example, the speed in the complete model ("speed Comp," Fig. 15.17a) oscillates at the instant voltage falls; these are eliminated in the simplified model (speed Simp). A more pronounced difference between models is observed at the machine starting (Fig. 15.17b).

It is important to mention that the simulation of the DFIM is made considering the two sources input: one for the stator (v_{d}, v_{q}) and the other for the rotor (v_{rd}, v_{rq}).

If nominal values around the equilibrium point are assumed, pole location analysis can be performed. Table 15.4 shows the pole values for the two models. By using these values, it is verified again that the change of causality (Fig. 15.16) and dynamics elimination ($\dot{\varphi}_{sq} = \dot{\varphi}_{sd} = 0$) allow to obtain a model without oscillations.

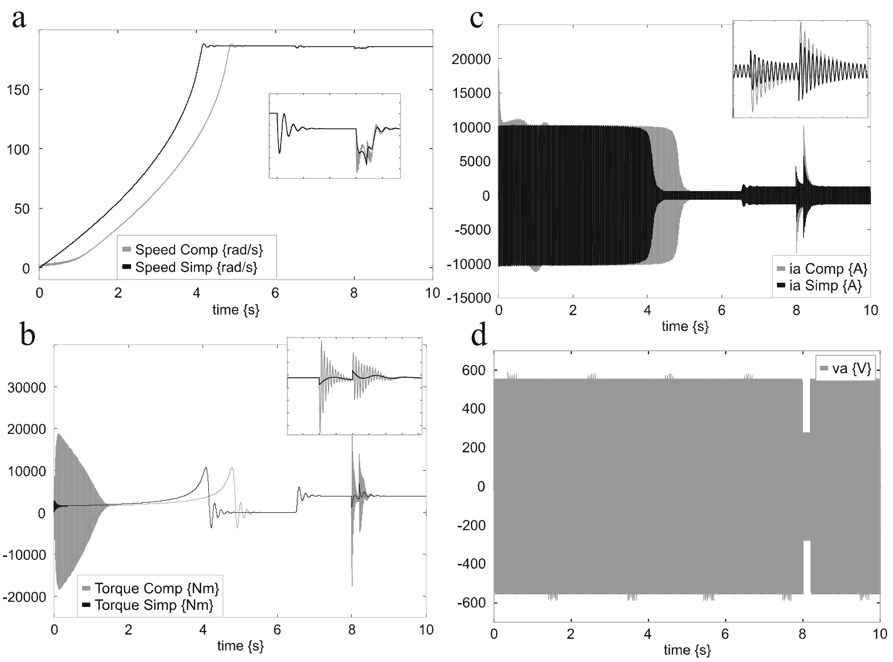


Fig. 15.17 Comparison responses of the complete (comp) and simplified (simp) DFIM, (**a**) speed, (**b**) torque, (**c**) current phase a, and (**d**) voltage input phase a

Table 15.4 Numerical pole values for complete and simplified model

Complete DFIM Model	Simplified DFIM Model
$P_1 = -6.8 + 35.9i$	$P_1 = -6.86 + 35.9i$
$P_2 = -6.8 - 35.9i$	$P_2 = -6.86 - 35.9i$
$P_3 = -12.4$	$P_3 = -12.39$
$P_4 = -15.69 + 376.5i$	
$P_5 = -15.69 - 376.5i$	

15.3.4 Power Electronic Converter

The three phase converter model has been addressed in many papers, such as [12, 23]. Each presented model has its own characteristic and is used for different applications.

15.3.4.1 Power Converter Model

As said before, two power electronics converters, the machine side converter (MSC) and network side converter (NSC), are used in order to have a DC-link between them allowing the power transfer.

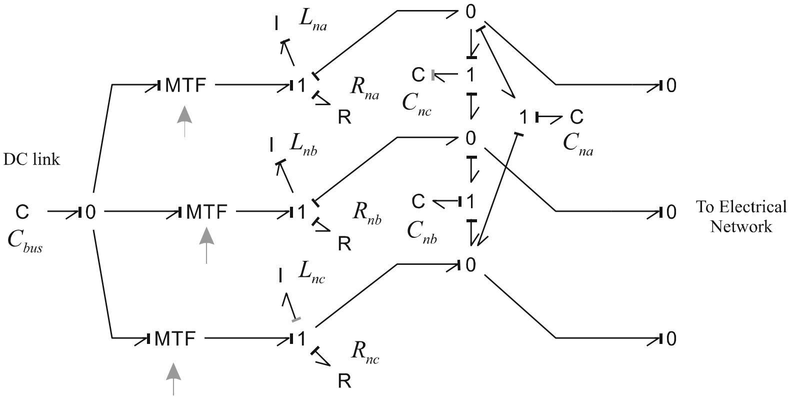


Fig. 15.18 Network side converter with a three phase LC filter

For the NSC, the model is taken from [42, 43]. This model corresponds to a three phase converter using an LC filter connection. The filter allows to give a voltage source behavior for the converter, and also to eliminate the harmonic distortion introduced by the converter. Figure 15.18 shows the NSC used.

The characteristics of the model are the DC-link is considered as a dynamical element (C_{bus}), and the LC filter is composed of three inductances with their associated resistances and three Delta connected capacitors.

The causality assignment gives to the power electronic converter associated with the LC filter a voltage source behavior.

In the next section it will be presented the control law necessary to transfer the active and reactive power to the network.

15.4 Control of the Doubly Fed Induction Generator

As previously shown, the induction machine is a coupled 5th or 3th order system (depending on the considered model) with four inputs. Actually, the simplification made in the previous section allows to intuitively knowing that stator dynamics are not directly involved in the control law. For that reason, the control inputs which are present in the doubly fed induction machine are those of rotor. The last assumption is commonly made when a generator control law is deduced, but here, the bond graph model supports this assumption.

15.4.1 Torque Control Formulation (Machine Side Converter)

When a wind turbine generator is used, the objective is to control the output torque. Also, the torque can be controlled indirectly by the speed. This selection will be dependent on the particular application.

It is important to remark that the proposed control does not consider the Sommerfeld effect [40, 41], since unbalance rotor conditions are not considered.

In order to control the output torque, it is necessary to control the *d-axis* rotor flux supplied by the DFIG. To this aim, a specific algorithm is designed, based on bicausal bond graph [11].

For the formulation of the inverse bond graph, it is necessary to change the effort detectors De (assumed ideal), which will be placed in bond 5 of Fig. 15.14 in the original bond graph, by a source named SS, (which impose zero flow but non-zero effort to the inverse model), then propagate bicausality (in only one line of power transfer) from this source (SS: $\dot{\varphi}_{\mathrm{rd}}$) to the input source of the original bond graph which becomes a detector (i.e., SS: v_{rd}) in the inverse bond graph [31].

The structure of the control in open loop is designed with the inverse bond graph. The decoupling actions are defined (inverse matrix and disturbance compensation). The open loop structure is then extended to a closed loop control by fixing the dynamics of errors.

Figure 15.19 shows the preferred derivative causality assigned in the inverse bond graph, which allows to deduce the open loop control laws. The rotor flux and torque sensors are simultaneously inverted via bicausal bonds; two disjoint bicausal paths are drawn to the two desired inverse model outputs, corresponding to the two control signals (MSe:v_{rd}, MSe:v_{rq}), which demonstrates that the model is invertible [31]. It is important to notice that the J_m inertia is not considered in the inverse bond graph of Fig. 15.19. This is because non-unbalanced rotor operation has been considered and because this inertia will represent the whole wind turbine inertia.

For the rotor flux control law, no change is presented in the I-field element M_{d}. Otherwise, the causality change in bond 4 is due to the simplified model consideration (this bond has not a dynamic behavior). Equation (15.9) is derived from the inverse bond graph of Fig. 15.19, and this relationship corresponds to the 1-junction placed between bonds 5, 6, 7, 8, and 9.

$$-e_6 + e_7 + e_8 - e_9 = e_5 \tag{15.9}$$

It is assumed that the numerical value of the elements (resistances and inductances) is the same for 2 axes. These values have been taken as estimated values, e.g., $\widehat{R}_r$ of the actual system parameters, in order to consider a slight error. By replacing these values in (15.4) it gives

$$v_{\mathrm{rd}} = \frac{\mathrm{d}}{\mathrm{d}t}\varphi_{\mathrm{rd}} + \widehat{R}_r i_{\mathrm{rd}} + p\varphi_{\mathrm{rq}}\Omega - \omega\varphi_{\mathrm{rq}} \tag{15.10}$$

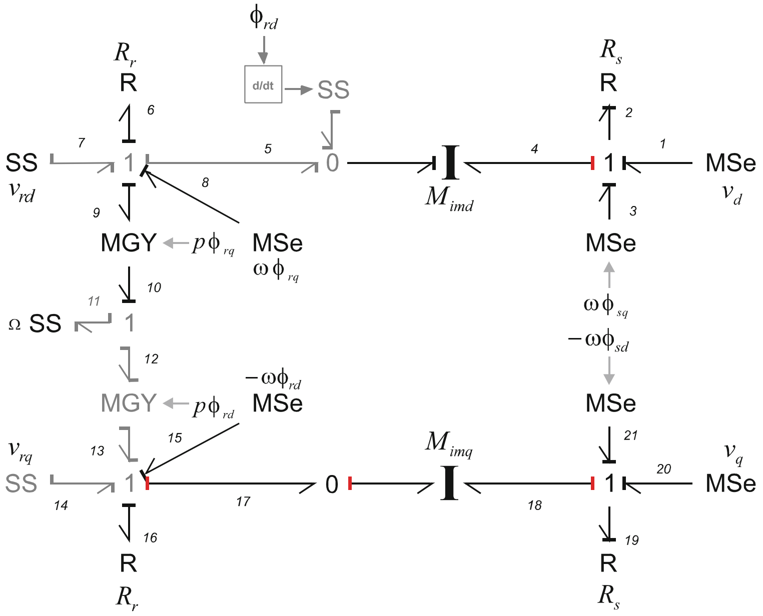


Fig. 15.19 Inverse DFIG for calculation of the controls laws

To establish the closed loop control law, the dynamics of the error ($\varepsilon = \varphi_{\text{rqref}} - \varphi_{\text{rq}}$) are set in (15.10), as $\dot{\varepsilon} + k_1\varepsilon = 0$, where k_1 represents the controller to be used. Expression (15.10) becomes (15.11) as:

$$v_{\text{rd}} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\varphi_{\text{rdref}} - \varepsilon\right) + \widehat{R}_r i_{\text{rd}} + p\varphi_{\text{rq}}\Omega - \omega\varphi_{\text{rq}} \tag{15.11}$$

Finally,

$$v_{\text{rd}} = \frac{\mathrm{d}}{\mathrm{d}t}\varphi_{\text{rdref}} + k_1\left(\varphi_{\text{rdref}} - \varphi_{\text{rd}}\right) + \widehat{R}_r i_{\text{rd}} + p\varphi_{\text{rq}}\Omega - \omega\varphi_{\text{rq}} \tag{15.12}$$

For the last expression, the rotor flux φ_{rq} is assumed to be zero. This is justified since φ_{rq}, compared to φ_{rd} has not an important impact on the voltage v_{rd}. Actually, this value is very small compared to the *d-axis* flux.

Taking the last consideration, (15.12) is represented in the Laplace domain as,

$$(v_{rd})_{(s)} = s(\varphi_{\text{rdref}})_{(s)} + k_1(\varphi_{\text{rdref}} - \varphi_{\text{rd}})_{(s)} + \widehat{R}_r(i_{\text{rd}})_{(s)} \tag{15.13}$$

In the same manner as for the determination of the rotor flux control, the torque control law is calculated.

Equation (15.14) is derived from the bond graph of Fig. 15.19 as,

$$e_{14} - e_{18} + e_{15} + e_{13} - e_{17} = 0 \tag{15.14}$$

For (15.14), the effort e_{17} needs to be calculated from (15.8), by assuming that $\varphi_{\mathrm{rq}} = \int e_{17}$. Making the appropriate substitutions in (15.14), it yields,

$$v_{\mathrm{rq}} = \widehat{L}_r \frac{\mathrm{d}}{\mathrm{d}t} i_{\mathrm{rq}} + \widehat{R}_r i_{\mathrm{rq}} - p\varphi_{\mathrm{rd}}\Omega + \omega\varphi_{\mathrm{rd}} \tag{15.15}$$

In (15.15), the current i_{sq} of (15.8) has been assumed zero, since the balanced condition of the Park transformation has been considered. However, the mutual inductance is not present in (15.15).

The voltage v_{rq} and torque relation are not explicitly shown in (15.15). Thus, in order to relate (15.15) with the torque, another expression is needed.

Two different procedures are visualized, i.e., the first is to replace the speed Ω for his equivalent relation with the torque, and the second one is to establish an internal control loop for the i_{rq} current, and cascade the torque relationship in the previously internal control loop.

The second solution is chosen, since it provides a control law structure similar to the traditional vector control [49] applied to this machine.

By using the estimated values and establishing the closed control loop (for i_{rq} variable) as previously described, (15.15) becomes,

$$v_{\mathrm{rq}} = \widehat{R}_r i_{\mathrm{rq}} - p\varphi_{\mathrm{rd}}\Omega + \omega\varphi_{\mathrm{rd}} + \widehat{L}_r \frac{\mathrm{d}}{\mathrm{d}t} i_{\mathrm{rqref}} + \widehat{L}_r k_2 \left(i_{\mathrm{rqref}} - i_{\mathrm{rq}}\right) \tag{15.16}$$

where k_2 represents the controller used for this loop. The other necessary expression is taken from the original model (Fig. 15.14); specifically, in bond 11 the torque is given by,

$$T_{e11} = p\left(\varphi_{\mathrm{rq}} i_{\mathrm{rd}} - \varphi_{\mathrm{rd}} i_{\mathrm{rq}}\right) \tag{15.17}$$

As $\varphi_{\mathrm{rq}} = 0$, expression (15.17) becomes $T_{e11} = -p\varphi_{\mathrm{rd}} i_{\mathrm{rq}}$, and can be expressed as $T_{\mathrm{ref}} = -p\varphi_{\mathrm{rd}} i_{\mathrm{rqref}}$. This expression is replaced in (15.16), to obtain the control law (15.18).

$$\left(v_{\mathrm{rq}}\right)_{(s)} = \widehat{R}_r \left(i_{\mathrm{rq}}\right)_{(s)} - p\varphi_{\mathrm{rd}}\Omega + \omega\varphi_{\mathrm{rd}} + \widehat{L}_r s \left(\frac{T_{\mathrm{ref}}}{-p\varphi_{\mathrm{rd}}}\right)_{(s)} + \widehat{L}_r k_2 \left(\frac{T_{\mathrm{ref}}}{-p\varphi_{\mathrm{rd}}} - i_{\mathrm{rq}}\right)_{(s)} \tag{15.18}$$

The controller block diagram (Eqs. (15.13) and (15.17)) is shown in Fig. 15.20. It is important to notice that the structure of the control law contains a feed-forward with a derivative action on the reference signals. The signals are constants, so that they can be removed, and by taking into consideration that the wind does not have a sudden change, they are kept in this proposed control law. In addition, the control law needs an estimation of φ_{rd}, then the expression ($L_{\mathrm{s}}{*}i_{\mathrm{rd}} + L_{\mathrm{m}}{*}i_{\mathrm{sd}}$) [49] is used.

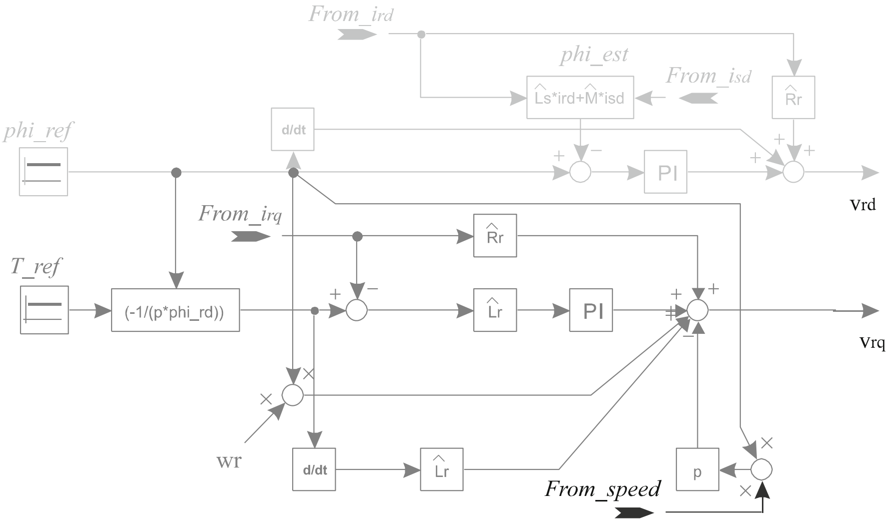


Fig. 15.20 Torque control law schema

15.4.2 *Torque Control Simulation*

In order to verify control law, a simulation is carried out by considering that the DFIG has a primary governor to emulate the torque provided by the turbine mechanical part. Also, as it is considered that the converter has not an important impact in the machine behavior, the machine side converter is not used. This allows to connect the v_{rd} and v_{rq} voltages directly via the MSe sources. Figure 15.21 shows the simulation scheme.

The stator machine power is regulated by the rotor. In order to simulate the DFIG, a three phase load is connected at machine terminals. Park's transformation is used in order to transform from the *dq* reference frame to the *abc* framework.

The complete DFIG model is used in the simulation. The results presented here have been verified against the simplified DFIG model, obtaining the same results; the only difference being that the oscillations due to the fast dynamics are present in the complete model.

A speed governor is used to maintain the nominal speed in the generator. As the objective is to test the behavior of the control law; this stage has been represented by using an ideal Governor (PI controller).

Results are shown in Fig. 15.22. Numerical values of Tables 15.3 and 15.5 are used for the simulation. The DFIG is started at instant $t = 0$ s applying the nominal torque as reference in the control ($T_{\mathrm{ref}} = 3978.88$ Nm), then at $t = 5$ s the nominal torque is reduce to $T_{\mathrm{ref}} = 2500$. This is made in order to observe that the torque in the machine is also reduced.

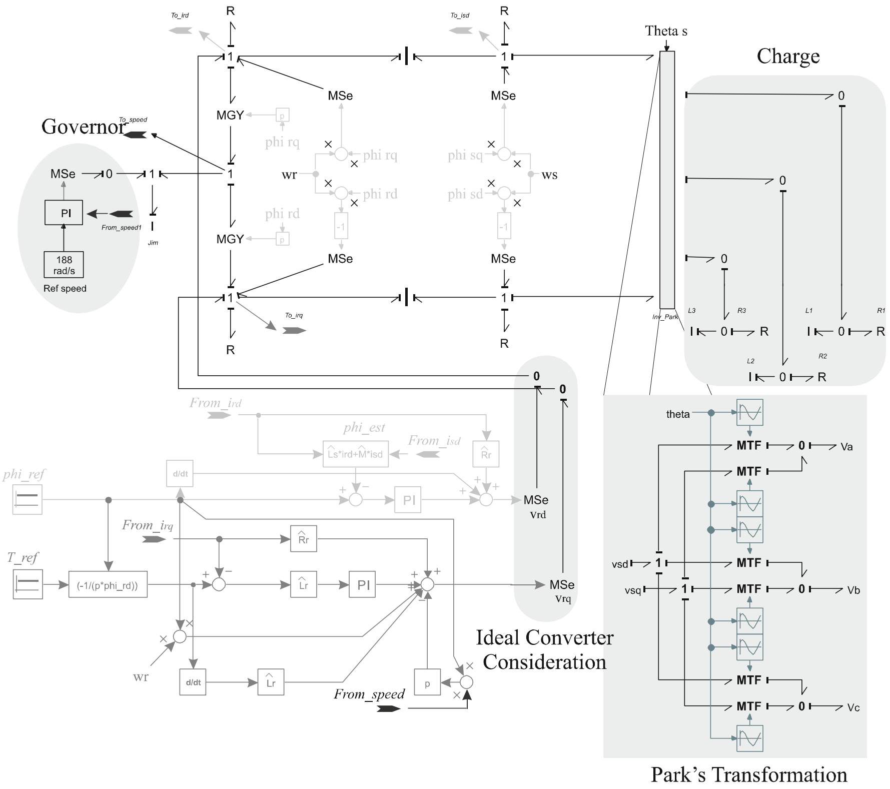


Fig. 15.21 Simulation schema for the DFIG—constant speed

The charge was selected to have a power factor of 0.8, and how is shown in Fig. 15.22d the total active power is supplied to the charge when the nominal torque is set in the reference control, and is reduced when the torque decreases.

15.4.3 *Robustness Test*

In order to verify the robustness of the proposed control law, the simulation schema presented before is used (Fig. 15.21).

The pole-plot can be taken directly from the simulator. Then, in order to verify the robustness of the control, different values of power factor in the load (0.5–1) are used. The nominal values of torque and the magnetic flux (T_{ref} and phi$_{\mathrm{ref}}$ in Fig. 15.21) have been considered by taking into account the parameters of Table 15.3 and Table 15.5. Figure 15.23 shows the 3D pole-plot evolution.

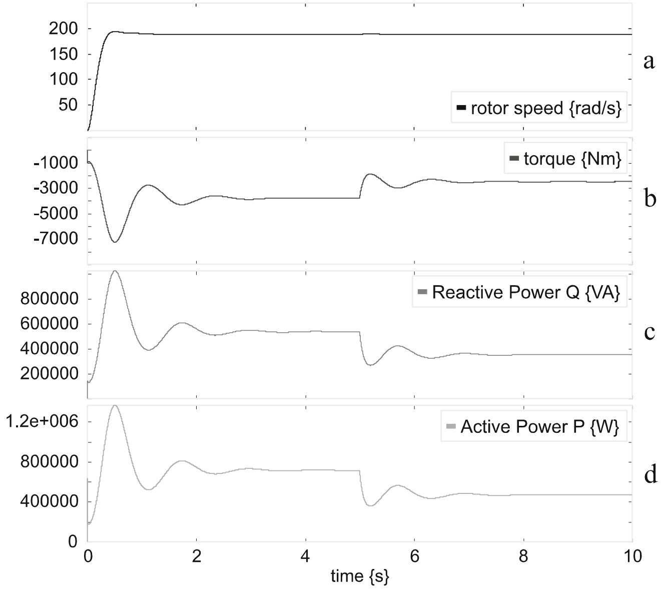


Fig. 15.22 DFIG responses, (**a**) Machine speed, (**b**) torque, (**c**) reactive power in the charge, and (**d**) active power in the charge

Table 15.5 Control and charge numerical values

Control	Charge	
PI v_{rd}	PI v_{rq}	
$K_{\mathrm{p}} = 1000$	$K_{\mathrm{p}} = 150$	$R_{1,2,3} = 0.5078\,\Omega$
$K_{\mathrm{i}} = 1$ m	$K_{\mathrm{i}} = 10$ m	$L_{1,2,3} = 1.01$ mH

Poles which have a significant movement correspond to the load poles. Unlike the generator and control poles, these do not have a significant movement when the power factor is varied. This shows the robustness of the proposed control in this system. It is important to mention that in order to take into account the uncertainties due to parameter estimation the simulation is carried out by assuming a slight error in the parameters, i.e., 10 % between the model and the controller parameters.

15.4.4 Network Side Converter

As said before, the three phase NSC uses an LC filter in order to impose a voltage behavior in the converter.

The structure of the converter was presented in Fig. 15.18. Then in this section, the active and reactive power control is presented.

The control law of this converter is also derived from the inverse bond graph (using the bicausality concept); the robustness and accurate performance of the

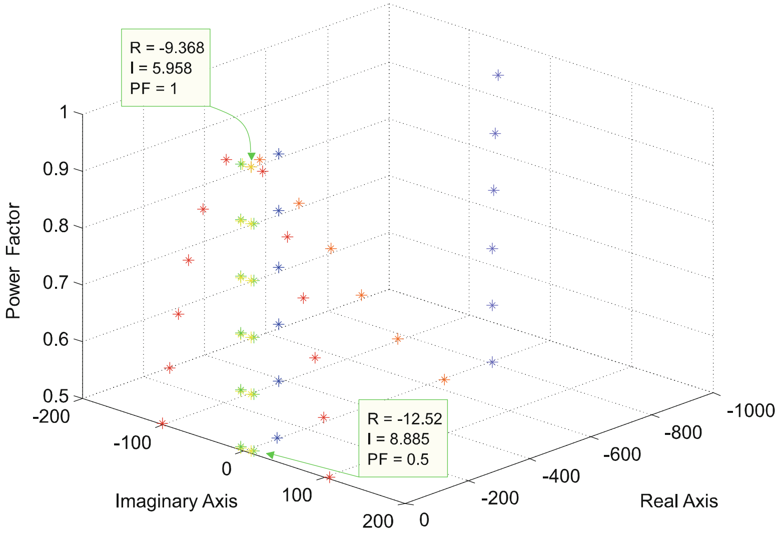


Fig. 15.23 3D pole-plot evolution

proposed control has been demonstrated for voltage regulation [42]. Here, the highlights of this control are recalled. They are as follows:

- The controller parameters are obtained considering the two loops in cascade, one for the current in the L_{ni} and another one for the voltage in C_{ni}-elements.
- Only two phase to phase voltages are considered in the control, the third one fixing the voltage reference.
- The current i_{nc} is linearly dependent on the two others (i_{na} and i_{nb}).
- The average behavior of the converter is considered.

The control is composed by three principal stages:

1. DC bus regulation. It is based on a traditional PI (proportional + integral) controller. It provides the active power reference.
2. Active and reactive power regulation. Inspired from the power flow concept between two sources, connected through a line impedance.
3. Voltage and current regulation. A resonant and proportional controller for voltage and current control, respectively.

Figure 15.24 resumes the control stages for the NSC.

From Fig. 15.24, the source V_{dcref} sets the DC-link regulation reference, the i_{bus} and v_{bus} are necessaries in order to have the total power available in the DC-link. Since this power provides the active power reference. In the right side of Fig. 15.24 it can see three I-elements connected in series with their respective R-elements. These elements represent the line impedance, and are necessaries in order to apply the power flow concept.

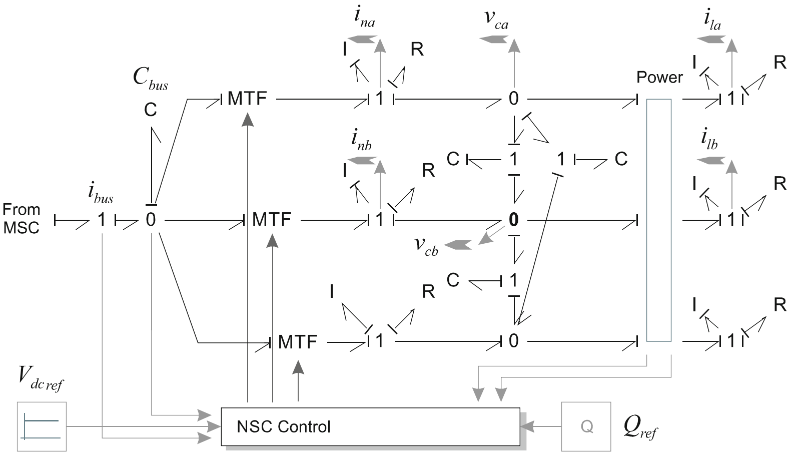


Fig. 15.24 Control schema simulation for the NSC

Finally in Fig. 15.24 it can be identified by a block connected between the delta connection capacitors (C-elements) and the line impedance. Inside this block is implemented Eqs. (15.19) and (15.20) which allows to get the active and reactive power measures.

$$P = v_{na}i_{na} + v_{nb}i_{nb} + v_{nc}i_{nc} \tag{15.19}$$

$$Q = \left(\left(v_{na} - v_{nb}\right) i_{nc} + \left(v_{nb} - v_{nc}\right) i_{na} + \left(v_{nc} - v_{na}\right) i_{nb}\right) / \sqrt{3} \tag{15.20}$$

15.4.4.1 Active and Reactive Power Control

In case of connection with an infinite voltage source, the power control loop is inspired from the power flow concept between two sources (v_{con} and v_{res}) connected by an impedance line $Z = R + jX$ [14]. Assuming that the transmission line has small resistance compared to the reactance ($R = 0$), the active power is given by,

$$\begin{array}{c} P = \frac{V_{con}V_{res}}{X}\delta_c \\ Q = \frac{V_{con}}{X}\left(V_{con} - V_{res}\right) \end{array} \tag{15.21}$$

where: V_{con} and V_{res} are, respectively, the RMS values for v_{con} and v_{res}, and δ_c is the phase of V_{con} versus V_{res}. Expression (15.21) shows that a small difference on the angle has a direct impact on the active power flow. The magnitudes difference has a direct impact on the reactive power. The active power control is defining the value for δ_c angle and the $\Delta V = V_{con} - V_{res}$ define the reactive power, as shown in the schematic diagram of Fig. 15.25.

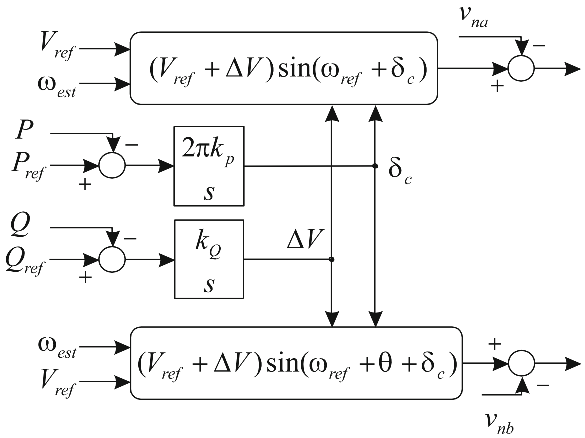


Fig. 15.25 Active and reactive power loop block diagram

In Fig. 15.25, k_P and k_Q are the gains in the controller, which have to be calculated taking into account the desired time response, ω_{est} is the grid pulsation, and V_{ref} is the network reference voltage.

15.5 Complete System

In this section the previously models are set up in order to simulate the complete system.

15.5.1 Simulation of a Variable-Speed Wind Turbine

The blades, gearbox, DFIG, and the power converters are associated in order to formulate the complete model of a variable-speed wind turbine as is shown in Fig. 15.26.

In Fig. 15.26, the block “Wind Turbine Model” corresponds to the blades and hub model of the turbine (Fig. 15.7). The gearbox model presented before (Fig. 15.12) is represented by the “GearBox” block. Also, Fig. 15.26 shows the whole control law involved in the variable-speed wind turbine. The block called “Wind control” corresponds to the wind turbine angular speed control. This control provides the reference torque, which is introduced in the torque control (Fig. 15.20) for the DFIG (Fig. 15.14). The active power is sensed in order to calculate the maximum power available in the mechanical part of the turbine. Then, this power is regulated via the pitch angle, given by the block “Pitch control.” The block “NSC Converter (+Control),” groups the continuous voltage (DC-link), power, voltage, and current controls for the converter (Fig. 15.24).

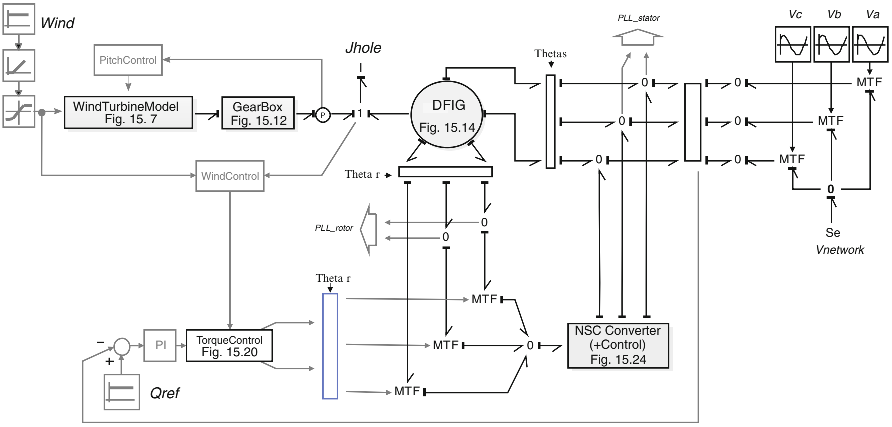


Fig. 15.26 Variable wind speed simulation model

It is important to mention that the control laws used for the angular speed and pitch angle of the blades have been taken from [34], and the NSC control from [42].

Additional elements such as power sensors, Park's transformation, and the phase lock-loop (PLL) are required to simulate the system. The turbine, the generator, and the power converter require sensors. For the turbine a power and angular speed sensors are used, while for the generator the rotor and stator currents sensors are needed to estimate the magnetic fluxes. These current sensors are also used in the power converter control.

In order to show the behavior of the complete model, two different simulations are conducted. First, a constant wind profile is considered, and then real wind profile data are used. As made before, these simulations have been performed using the 20Sim software.

15.5.2 Constant Wind Consideration

In this section the wind profile is considered constant with a ramp change in their speed.

The scenario for simulation is as follows: a constant wind of 5 m/s is applied at the simulation start; then, at $t = 10$ s, a wind ramp is applied. This wind ramp increases from 5 m/s, at $t = 10$ s to 15 m/s, at $t = 27.5$ s; this value is maintained until the simulation concludes. Figure 15.27a–c shows the simulation responses for the wind speed, DFIG speed, the tip speed, the power coefficient, the pitch angle, and the torque in the DFIG, respectively.

The speed in the DFIG increases in the same ratio as the wind input, i.e., up to the maximum value (226 rad/s) when the wind speed exceeds 12 m/s (see Fig. 15.27b). This is due to the fact that this value has been assumed to be the maximum set

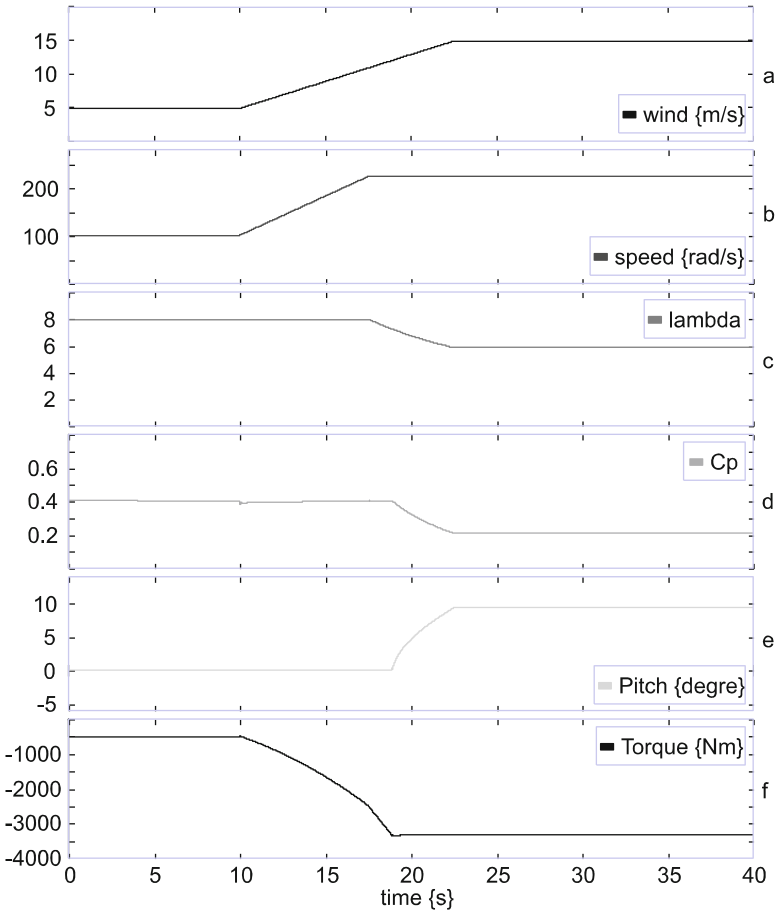


Fig. 15.27 Responses for a constant wind profile (**a**) wind, (**b**) DFIG speed, (**c**) λ, (**d**) power coefficient, (**e**) pitch angle, and (**f**) DFIG torque

point. The curve called "lambda" corresponds to the tip speed ratio, which presents its nominal value before the ramp profile starts (Fig. 15.27c). When machine speed increases to its maximum value, the tip speed decreases from 8 to 5.8. A similar case is shown for the power coefficient Cp (Fig. 15.27d). Figure 15.27e also shows the pitch angle applied to the blades, which starts at the instant when the wind speed exceeds 12 m/s. Finally, in Fig. 15.27f the actual torque in the generator is presented. There is a negative torque because the mechanical power is transferred to the power network, and their maximum value is −3315 Nm.

Active power needs to be provided by the wind turbine, and then distributed to the power network. Nevertheless, as a DFIM is used, the reactive power is present. Figure 15.28 shows the mechanical and electrical power in the wind turbine model. The power of the main shaft and the high speed shaft exactly match (750 kW); the

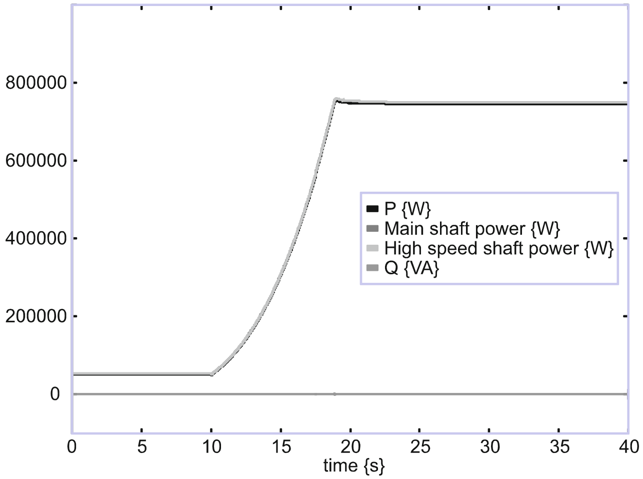


Fig. 15.28 Power curves for a constant wind profile

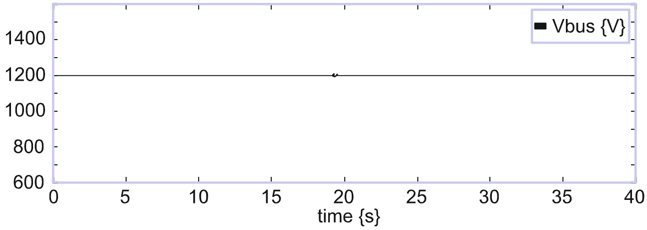


Fig. 15.29 Voltage in the DC-link for a constant wind profile

generator active power is closely to these ones (745 kW). This difference is due to the fact that estimated values for the generator have been calculated. Zero reactive power (Q) has been set as reference because not reactive power is absorbed by the network.

The regulation of the continuous bus is also an important parameter to take into consideration. Figure 15.29 shows the DC-link capacitor voltage, which follows the voltage reference (1200 V). It is important to mention that an initial condition is used in order to allow the simulation to be performed.

Figure 15.30 shows the three phase stator currents. A 460 V DFIG is used, with currents having larger values in order to meet active power demand.

The last simulation results allow the verification of theoretical concepts related to the wind turbine dynamic operation.

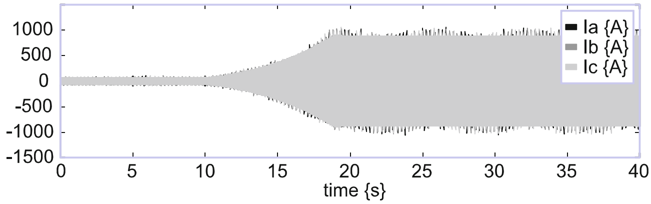


Fig. 15.30 Stator currents of DFIG—constant wind profile

15.5.3 Variable Wind Consideration

Practical wind behavior has a varying profile instead of being constant. The performance of the proposed control law can be confirmed if a real wind profile is used. Data are introduced in the simulator with a "data from file" block, which takes the numerical values from a table each time step. Figure 15.31 shows the simulation response for the selected wind turbine variables.

The wind changes (from 9 to 13 m/s) are reflected in the DFIG speed with the same ratio. When the wind crosses the limit of 12 m/s for small periods of time, i.e., from 10 to 60 s, the DFIG reaches its maximum value (226 rad/s). Also, this behavior is presented in the lambda (Fig. 15.31c), the Cp (Fig. 15.31d), and the pitch angle (Fig. 15.31e) responses.

In order to verify the generation of active power, Fig. 15.31e shows the actual torque. The torque arises to its maximum value when the wind speed increase (up to 12 m/s) or same, when the pitch angle increases.

Figure 15.32 shows the power curves for a real wind profile. Unlike the results presented with constant wind profile, in this simulation changes are observed in the power generated. As same as in the constant wind profile consideration, the main shaft and high speed shaft powers exactly match. Zero reactive power (Q) has also been set as reference.

The DC-link voltage is not presented, because is same as for the constant wind consideration (Fig. 15.29). Figure 15.33 shows the three phase currents supplied by the wind turbine to the network. Unlike the current results presented in the previous section, the changes in the magnitude are notorious.

The simulation results can verify that the proposed model and controls present a good performance, either for constant or variable wind speed. Nevertheless, some improvements can be suggested, as for example, the angular speed and pitch control needs to be improved. This is because they have been taken directly from [34], but need to be developed taking into account the model presented here.

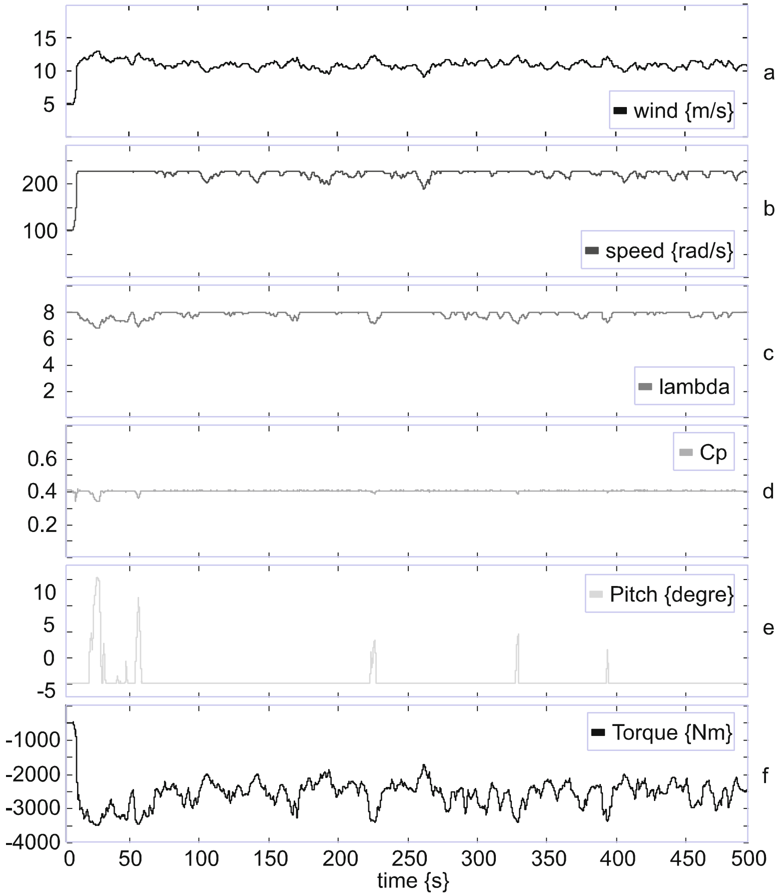


Fig. 15.31 Responses for a real wind profile (**a**) wind, (**b**) DFIG speed, (**c**) λ, (**d**) power coefficient, (**e**) pitch angle, and (**f**) DFIG torque

15.6 Conclusions

The complete model of a variable-speed wind turbine has been presented.

In order to apply the aerodynamic force, blade structure is considered to be a flexible body. BEM theory has been used for the wind aerodynamic force conversion. Values of torque and the coefficient of power in simulations have shown the validity of this model.

A gearbox model has been presented, where mesh stiffness between planet-sun and ring-planet are considered, and parallel stages are joined in only one stage. Power flows transferred by each stage has been presented in the simulation results.

For the DFIG, the causal loop concept applied to the model allowed to identify the dynamics (slow and fast) of the model approximation. Then, by choosing the

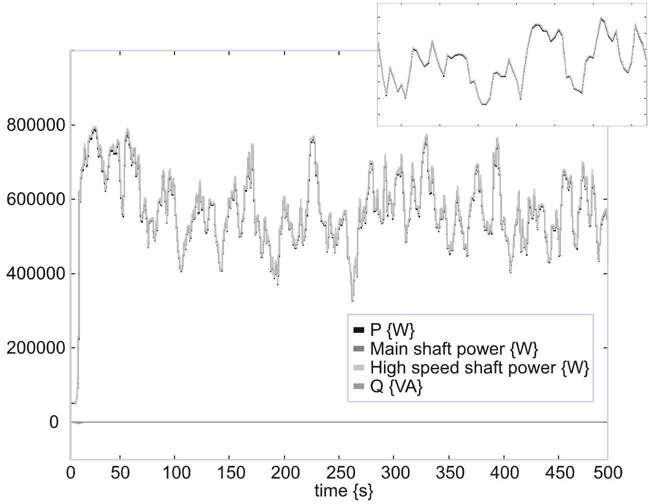


Fig. 15.32 Power curves for a real wind profile

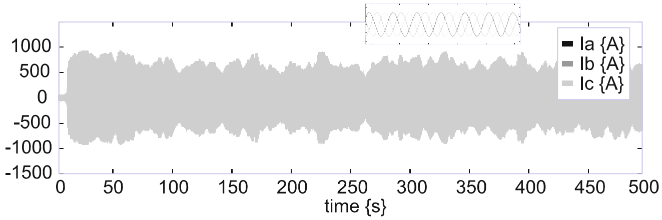


Fig. 15.33 Stator currents of DFIG—real wind profile

elimination of the fast dynamics, in order to reduce the order of the model, the simplified DFIG model has been used. By applying the bicausality concept, the simplified model has been used in order to formulate the inverse bond graph. The control laws were intuitively obtained by considering the simplified model.

As the proposed control law strongly depends on the estimated DFIM parameters, a traditional pole-place analysis has been conducted in order to verify the robustness of the control. The small error considered in the parameters did not vary the pole-place diagram.

The wind turbine behavior was simulated in order to verify the dynamic performance of the whole system. It has been demonstrated, via these simulations, that the proposed model yields good results; for real and for a constant wind consideration. Nevertheless, this model needs to be subjected to different tests, i.e., turbulent wind, gearbox faults, etc., in order to know the model performance.

It is suggested to test the proposed control law in a real-time simulator, in order to make a validation still closer to reality, before being tested in a real wind turbine.

Only a few publications involving all the stages of a wind turbine can be found in the open literature. In this context, the model presented here is an important starting point for the analysis of this complex system.

Appendix

Table 15.6 Wind turbine data

Hub and main shaft	
$J_{\mathrm{hub}} = 5000$ kg.m^2, $D_{\mathrm{hub}} = 1000$ N/m, $K_{\mathrm{ms}} = 3.67\mathrm{e}7$ N/m, $D_{\mathrm{ms}} = 200$ N/m	
High speed shaft	
$K_{\mathrm{hs}} = 10\mathrm{e}7$ N/m, $D_{\mathrm{hs}} = 1\mathrm{e}-3$ N/m	
Blade structure data	
Section 1	Section 2
$E = 1.7\mathrm{e}10$	$E = 1.7\mathrm{e}10$
$l = 11.7$ m	$l = 11.7$ m
$M = 120.8$ kg	$M_2 = 48.7$ kg
$J = 3.3$ kg.m^2	$J_2 = 2.33$ kg.m^2
$\mu = 0.01$	$\mu = 0.01$
$J_{\mathrm{whole}} = 1000$ kg.m^2	
Blade aerodynamic conversion	
$c_1 = 1.9$	$c_2 = 1$
$\beta_{\mathrm{t1}} = 11$ deg	$\beta_{\mathrm{t2}} = 1.7$ deg
$\rho_{\mathrm{air}} = 1.225$	$\rho_{\mathrm{air}} = 1.225$
$r_1 = 5.85$ m	$r_2 = 17.5$
$l_1 = 11.7$ m	$l_2 = 11.7$ m

References

1. Agarwal, S., Chalal, L., Dauphin-Tanguy, G., & Guillaud, X. (2012). Bond graph model of wind turbine blade. *IFAC Proceedings Volumes, 45*(2), 409–414.
2. Borutzy, W. (2011). *Bond graph modelling of engineering systems: Theory, applications and software support*, Chapter 2. New York: Springer Science Business Media, LLC.
3. Brown, F. (2001). *Engineering system dynamics*. New York: Marcel Dekker. ISBN 0-8247-0616-1.
4. Buhl, M. L. Jr. (2004). *A new empirical relationship between thrust coefficient and induction factor for the turbulent windmill state*. NREL/TP-500-36834. Golden, CO: National Renewable Energy Laboratory.
5. Burton, T., Sharpe, D., Jenkins, N., & Bossanyi, E. (2001). *Wind energy handbook* (p. 11). New York: Wiley.
6. Coudert, N., Dauphin-Tanguy, G., & Rault, A. (1993). Mechatronic design of an automatic gear box using bond graphs. In *IEEE International Conference on Systems, Man and Cybernetics, 'Systems Engineering in the Service of Humans'* (pp. 216–221).

7. Dauphin-Tanguy, G. (2000). *Les bond graphs*. Paris: Hermès Science Editor.
8. Deur, J., Ivanović, V., Assadian, F., Kuang, M., Tseng, E. H., & Hrovat, D. (2012). Bond graph modeling of automotive transmissions and drivelines. *Proceedings of the 7th Vienna International Conference on Mathematical Modelling (MATHMOD)*, Vienna, Austria.
9. Etienne-Vidal, P., Pietrzak-David, M., & Bonnet, F. (2008). Mixed control strategy of a doubly fed induction machine. *Electrical Engineering (Archiv fur Elektrotechnik), 90*(5), 337–346.
10. Gallo, C., Kasuba, R., Pintz, A., & Spring, J. (1986). *Design and dynamic simulation of a fixed pitch 56 kw wind turbine drive train with a continuously variable transmission*. U.S. Department of Energy Conservation and Renewable Energy Wind/Ocean Technology Division, NASA CR-179543.
11. Gawthrop, P. J. (1995). Bicausal bond graph. *Proceedings of the International Conference on Bond Graph Modeling and Simulation (ICBGM'95)* (Vol. 27, pp. 83–88).
12. Gonzáñez-Contreras, M., Rullán-Lara, J. L., Vela-Valdés, L. G., & Claudio, S. A. (2007). Modeling, simulation and fault diagnosis of the three-phase inverter using bond graph. In *IEEE International Symposium on Industrial Electronics* (pp. 130–135).
13. Guo, Y., Keller, J., & Parker, R. G. (2012). Dynamic analysis of wind turbine planetary gears using an extended harmonic balance approach. *International Conference on Noise and Vibration Engineering*, Leuven, Belgium, September 17–19, 2012.
14. Hadi, S. (1999). *Power system analysis* (pp. 26–28). New York: McGraw-Hill Editions.
15. Hansen, A. D., Sorensen, P., Iov, F., & Blaabjerg, F. (2005). Centralized power control of wind farm with doubly fed induction generators. *Renewable Energy, 31*(7), 935–951.
16. Hansen, M. O. L., Sorensen, J. N., Voutsinas, S., Sorensen, N., & Madsen, H. A. (2006). State of the art in wind turbine aerodynamics and aeroelasticity. *Progress in Aerospace Sciences, 42*, 285–330.
17. Józef, W., Jerzy, K., & Zawiśla, S. (2006). Gears and graphs. *Journal of Theoretical and Applied Mechanics, 44*(1), 139–162.
18. Karnopp, D. (1991). State functions and bond graph dynamic models for rotary, multi-winding electrical machines. *Journal of the Franklin Institute, 328*(1), 45–54.
19. Karnopp, D., Margolis, D., & Rosenberg, R. (1990). *System dynamics: A unified approach*. New York: Wiley.
20. Kubiak, P. (1996). *Analyse symbolique des systèmes physiques modélises par bond graph en comportant des éléments multiports*. Thèse Université des Sciences et Technologies de Lille, USTL. Lille France.
21. Luo, Y., & Tan, D. (2011). Dynamics modeling of planetary gear set considering meshing stiffness based on bond graph. *Procedia Engineering, 24*, 850–855.
22. Lydia, M., Immanuel Selvakumar, A., Suresh Kumar, S., & Edwin Prem Kumar, G. (2013). Advanced algorithms for wind turbine power curve modeling. *IEEE Transaction on Sustainable Energy, 4*(3), 827–835.
23. Mezghanni, D., Andoulsi, R., Mami, A., & Dauphin-Tanguy, G. (2007). Bond graph modeling of a photovoltaic system feeding an induction motor-pump. *Simulation Modeling Practice and Theory, 15*(10), 1224–1238.
24. Mikati, M., Santos, M., & Armenta, C. (2013). Electric grid dependence on the configuration of a small-scale wind and solar power hybrid system. *Renewable Energy, 57*, 587–593.
25. Miller, A., Chang, B., Issa, R., & Chen, G. (2013). Review of computer-aided numerical simulation in wind energy. *Renewable and Sustainable Energy Reviews, 25*, 122–134.
26. Morel, L., Godfroid, H., Mirzaian, A., & Kauffmann, J. (1998). Double-fed induction machine: Converter optimization and field oriented control without position sensor. *IEE Proceedings—Electric Power Applications, 145*(4), 360–368.
27. Moriarty, P. J., & Hansen, A. C. (2005). *AeroDyn theory manual*. National Renewable Energy Laboratory NREL/TP-500-36881.
28. Mukerjee, A., Karmakar, R., & Samantaray, A. K. (1999). Modeling of basic induction motors and source loading in rotor-motor systems with regenerative force field. *Simulation Practice Theory, 7*, 563–576.
29. Mukherjee, A., Karmakar, R., & Samantaray, A. K. (2000). *Modelling and simulation of engineering systems through bond graph*. New Delhi: Narosa Publishing House.

30. Ngwomo, R. F., & Scarvarda, S. (1999). Dimensioning problems in system design using bicausal bond graph. *Simulation Practice and Theory*, 577–587.
31. Ngwompo, R. F., Scavarda, S., & Thomasset, D. (1996). Inversion of linear time-invariant SISO systems modelled by bond graph. *Journal of the Franklin Institute, 333(B)*(2), 157–174.
32. Ontiveros, L. J., Mercado, P. E., & Suvire, G. O. (2010). A new model of the doubly-feed induction generator wind turbine. *IEEE/PES Transmission and Distribution Conference and Exposition: Latin America*.
33. Patel Mukund, R. (2000). *Wind and solar power systems*. Boca Raton, FL: CRC Press LLC.
34. Patiño, C., Tapia, R., Medina, A., & Fuerte, C. (2014). Wind turbine inverse control: A bond graph approach. *IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC2014)* (pp. 1–6). doi:10.1109/ROPEC.2014.7036354.
35. Peresada, S., Tilli, A., & Tonielli, A. (2004). Power control of a doubly fed induction machine via output feedback. *Control Engineering Practice, 12*(1), 41–57.
36. Ragheb, A., & Ragheb, M. (2010). Wind turbine gearbox technologies. *Proceedings of the 1st International Nuclear and Renewable Energy Conference (INREC10)*, Amman, Jordan, March 21–24, 2010.
37. Rosenberg, R. C., & Adry, A. N. (1979). Solvability of bond graph structures with loops. *IEEE Transactions on Circuits and Systems , CAS-26*(2), 130–137.
38. S. Junco. (1999). Real- and complex-power bond graph modeling of the induction motor. *Proceedings of the International Conference on Bond Graph Modeling and Simulation (ICBGM'99)*, San Francisco (Vol. 31, pp. 323–328).
39. Sahm, D. (1979). A two-axis, bond graph model of the dynamics of synchronous electrical machines. *Journal of the Franklin Institute, 308*(3), 205–218.
40. Samantaray, A. K., Dasgupta, S. S., & Bhattacharyya, R. (2010). Sommerfeld effect in rotationally symmetric planar dynamical systems. *International Journal of Engineering Science, 48*(1), 21–36.
41. Samantaray, A. K., Dasgupta, S. S., & Bhattacharyya, R. (2010). Bond graph modeling of an internally damped non-ideal flexible spinning shaft. *Journal of Dynamic Systems, Measurement, and Control*, *132*(6), 502–509.
42. Sanchez, R., Dauphin-Tanguy, G., Guillaud, X., & Colas, F. (2010). Bond graph based control of a three-phase inverter with LC filter connection to passive and active loads. *Simulation Modelling Practice and Theory, 18*, 1185–1198.
43. Sanchez, R., Guillaud, X., & Dauphin-Tanguy, G. (2012). Hybrid electrical power system modeling and management. *Simulation Modelling Practice and Theory, 25*, 190–205.
44. Sanchez, R., & Medina, A. (2014). Wind turbine model simulation: A bond graph approach. *Simulation Modelling Practice and Theory, 41*, 28–45.
45. Tapia, R., & Medina, A. (2015). Doubly-fed wind turbine generator control: A bond graph approach. *Simulation Modelling Practice and Theory, 53*, 149–166.
46. Tore, B., & Reza, K. (2011). Wind turbine modeling using the bond graph. *IEEE International Symposium on Computer-Aided Control System Design (CACSD)* (pp. 1208–1213).
47. Xu, L., & Wei, C. (1995). Torque and reactive power control of a doubly fed induction machine by position sensorless scheme. *IEEE Transactions on Industry Application, 31*(3), 636–642.
48. Younsy, R., El-Batanony, I., Tritsch, J. B., Naji, H., & Landjerit, B. (2001). Dynamic study of a wind turbine blade with horizontal axis. *European Journal of Mechanics, A. Solids, 20*, 241–252.
49. Zaimeddine, R., & Undeland, T. (2010). DTC control schemes for induction motor fed by three-level NPC-VSI using space vector modulation. *International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM 2010)*, Pisa, Italy, June 14–16, 2010.

Chapter 16
Bond-Graph Modelling and Causal Analysis of Biomolecular Systems

Peter J. Gawthrop

16.1 Introduction

Bond graphs were introduced by Paynter [32] and their engineering application is described in number of textbooks [15, 21, 25, 44] and a tutorial for control engineers [10]. Bond graphs were first used to model chemical reaction networks by Oster et al. [28] and a detailed account is given by Oster et al. [29]. Subsequent to this, the bond graph approach to chemical reactions has been extended by Cellier [7], Thoma and Mocellin [42] and Greifeneder and Cellier [16]. In 1993, the inventor of bond graphs, Henry Paynter, said [33][1]:

> Katchalsky's breakthroughs in extending bond graphs to biochemistry are very much on my own mind. 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. This will occur both in device form, say as chemfets, biochips, etc., as well as in the basic sciences of biology, genetics, etc.

Based on the work of Katchalsky and coworkers [28, 29], and the more recent work of Gawthrop and Crampin [11, 12] and Gawthrop et al. [13], this chapter presents an introduction to the bond graph modelling of the biomolecular systems of living organisms. In particular, the approach is based on the transduction of Gibbs energy and the corresponding chemical potential/molar flow covariables appropriate to isothermal and isobaric thermodynamic systems [3, 5]. Molecular species are represented by non-linear **C** components and reactions by non-linear two-port

[1] This was pointed out to me by the Editor of this volume, Wolfgang Borutzky.

P.J. Gawthrop (✉)
Systems Biology Laboratory, Melbourne School of Engineering, University of Melbourne, Melbourne, VIC 3010, Australia
e-mail: peter.gawthrop@unimelb.edu.au

W. Borutzky (ed.), *Bond Graphs for Modelling, Control and Fault Diagnosis of Engineering Systems*, DOI 10.1007/978-3-319-47434-2_16

R components. As living systems are neither at thermodynamic equilibrium nor closed, open and non-equilibrium systems are considered and illustrated using examples of biomolecular systems. The bond graph representation of biomolecular systems is complicated by the fact that the **R** and **C** components are fundamentally nonlinear; however the bond graph representation of biomolecular systems is simplified by the fact that there are no **I** or **GY** components.

In addition to their role in ensuring that models are energetically correct, bond graphs provide a powerful and natural way of representing and analysing *causality*. Causality is used in this chapter to examine the properties of the junction structures of biomolecular systems. Junction structures define the behaviour of the systems and thus the analysis of junction structures can reveal hidden information about a biomolecular system. Ort and Martens [26] and Perelson [34] give a basic analysis of junction structures and Sueur and Dauphin-Tanguy [38, 39, 40] show how basic system properties such as structural controllability can be derived from the bond graph junction structure by applying both integral and derivative causality.

For the purposes of simulating a biomolecular system, the **C** components are all in integral causality. As will be seen in this chapter, different causal patterns can be used to probe the fundamental properties of the junction structure. Stoichiometric analysis of biomolecular systems [18, 22, 30, 31] looks at the null spaces of the *stoichiometric matrix* to derive fundamental properties of the systems expressed as *conserved moieties*[2] and *flux paths*. This chapter shows how these results are related to the causal properties of the bond graph junction structure and provides new insights into the dynamics of biomolecular systems.

Section 16.2 introduces bond graph modelling of biomolecular systems and Sect. 16.3 describes structural analysis of biomolecular systems using the junction structure concept. Section 16.4 gives a bond graph model of a classical biomolecular cycle described by Hill [17] and provides a junction structure based analysis of the system. Section 16.5 gives a bond graph model of a simplified model of glycolysis, an important component of the metabolism of living systems. Section 16.6 gives similar modelling and analysis of a key component of living systems: the phosphorylation/dephosphorylation reaction system. Section 16.7 contains concluding remarks and suggestions for further work.

16.2 Bond Graph Modelling of Biomolecular Systems

Section 16.2.1 gives some background thermodynamics based on the textbook of Atkins and de Paula [3]. Following the exposition of Gawthrop and Crampin [11], Sect. 16.2.2 defines the pair of energy covariables used for biomolecular modelling: chemical potential and molar flow rate. Section 16.2.3 looks at the bond graph **C**

[2] In this context, a *moiety* is a part of a molecule; in some reactions, such moieties are *conserved* across different molecules.

component which, in this context, corresponds to a chemical species. The effort variable is chemical potential which is also discussed by Fuchs [9] and Job and Herrmann [19]. Section 16.2.4 looks at the two-port **R** component (given the special name **Re** here) introduced by Oster et al. [29] to model a chemical reaction. A one-port **R** with associated **1** junction is also used to model reactions as discussed by Karnopp [20]. The corresponding effort variable is *affinity*, the weighted sum of chemical potentials, and the flow variable is molar flow. As discussed by Oster et al. [28, 29] and Gawthrop and Crampin [11] it is possible to model biomolecular systems without the use of inertial **I** or gyrator **GY** components. Some fundamental ideas related to this point are discussed by Breedveld [6].

The discussion here is restricted to single-compartment systems and a simple illustrative example is given in Sect. 16.2.5.

16.2.1 Basic Thermodynamics

The following definitions and results are drawn from [3, Sect. F2]; the adjectives *isobaric* and *isothermal* mean constant pressure and temperature respectively.

Molar amount. The number of moles n_A of a substance A is the ratio of the number of entities N_A to the Avogadro constant N_{Avo}

$$n_A = \frac{N_A}{N_{\text{Avo}}} \text{ mol} \tag{16.1}$$

Mole fraction. In a mixture of substances the mole fraction χ_A[3] of substance A is the ratio of the molar amount n_A of A to the total molar amount of all substances in the mixture n_{total}

$$\chi_A = \frac{n_A}{n_{\text{total}}} \tag{16.2}$$

By definition: $0 \le \chi_A \le 1$.

Molar concentration. In a mixture of substances the molar concentration $[A]$ of substance A is the ratio of the molar amount n_A of A to the total volume V_{total} m^3

$$[A] = \frac{n_A}{V_{\text{total}}} \text{ mol m}^{-3} \tag{16.3}$$

Confusingly, the symbol $[A]$ is also used to represent the non-dimensional quantity

$$\frac{[A]}{c^{\varnothing}} = \frac{n_A}{Vc^{\varnothing}} = \frac{\chi_A n_{\text{total}}}{Vc^{\varnothing}} \tag{16.4}$$

where $c^{\varnothing}$ is the standard concentration of 1 mol dm^{-3}.

[3] The standard notation would be x_A but this clashes with our bond graph notation.

Molar volume. The molar volume V_A of substance A is the volume that it occupies per mole

$$V_A = \frac{V_{\text{total}}}{n_A}\ \text{m}^3\ \text{mol}^{-1} \tag{16.5}$$

Gibbs energy. The Gibbs energy G J (also called Gibbs free energy or just free energy) is defined as in terms of enthalpy H J, temperature T K and entropy S J K^{-1} as

$$G = H - TS\ \text{J} \tag{16.6}$$

Molar enthalpy. The molar enthalpy[4] H_A of substance A is the enthalpy per mole

$$H_A = \frac{H}{n_A}\ \text{J}\ \text{mol}^{-1} \tag{16.7}$$

Partial molar property. A partial molar property of a substance A forming part of a mixture is the contribution of that (extensive) property to the overall property of the mixture. The partial molar property depends not only on the substance but also on the other substances forming the mixture.

Chemical potential. The *partial* molar Gibbs energy of substance A is given the evocative title of *chemical potential* μ_A

$$\mu_A = G_{Am} = \frac{G_m}{n_A}\ \text{J}\ \text{mol}^{-1} \tag{16.8}$$

where G_m is the Gibbs energy of A within the current mixture. Alternatively, μ_A may be defined as $\frac{\partial G}{\partial n_A}$ where G is the total Gibbs energy of the mixture.[5]

Entropy exchange. When the *enthalpy* of an *isobaric system* changes by ΔH J, the *entropy* of the *environment* S_{env} J K^{-1} (assumed to be infinite and at a constant temperature T) changes by

$$\Delta S_{\text{env}} = -\frac{\Delta H}{T}\ \text{J}\ \text{K}^{-1} \tag{16.9}$$

Total entropy. If the system is isothermal (temperature T K) and isobaric, the total entropy change ΔS_{total} is given by

$$\Delta S_{\text{total}} = \Delta S - \frac{\Delta H}{T} \tag{16.10}$$

[4] Other extensive quantities such as internal energy, entropy, Gibbs energy also have molar versions.

[5] Michael Pan, private communication.

Change in Gibbs energy. For isothermal (temperature T K) and isobaric systems

$$\Delta G = \Delta H - T\Delta S = -T\Delta S_{\text{total}}\ \text{J} \tag{16.11}$$

Chemical potential of substance in an ideal solution. If Henry's law holds (equivalent to a dilute solution) and substance A is dilute

$$\mu_A = \mu_A^\star + RT \ln \chi_A\ \text{J mol}^{-1} \tag{16.12}$$

where μ_A is the chemical potential of A, χ_A is the mole fraction of A (16.2) and $\mu_A^\star$ is the value of μ_A when it is pure ($\chi_A = 1$). Defining x_A as the number of moles of substance A:

$$x_A = n_A\ \text{mol} \tag{16.13}$$

and x_{total} as the total number of moles of all substances in the mixture, Eq. (16.12) becomes

$$\mu_A = \mu_A^\star + RT \ln \frac{x_A}{x_{\text{total}}}\ \text{J mol}^{-1} \tag{16.14}$$

It is convenient to reformulate Eq. (16.14) as

$$\mu_A = RT \ln K_A x_A\ \text{J mol}^{-1} \tag{16.15}$$

$$\text{where } K_A = \frac{\exp \frac{\mu_A^\star}{RT}}{x_{\text{total}}}\ \text{mol}^{-1} \tag{16.16}$$

16.2.2 Bond Graph Covariables

This chapter considers isothermal isobaric systems. In this context, an appropriate pair of energy covariables for substance A is

Effort. Chemical potential μ as expressed in Eq. (16.15) with units of J mol^{-1}.
Flow. Molar flow rate v with units of mol s^{-1}.

The product of the effort and flow variables is $p = \mu v$ with units of J s^{-1} or W.

16.2.3 The Bond Graph **C** Component

The **C** component has one port with flow variable $v\ \text{mol s}^{-1}$ and associated effort variables the chemical potential $\mu\ \text{J mol}^{-1}$ given by Eq. (16.15). The component

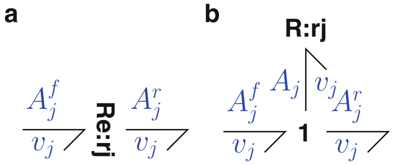


Fig. 16.1 Reaction component. (**a**) The **Re**:**rj** component relates the jth reaction flow rate v_j to the forward A_j^f and reverse A_j^r affinities. (**b**) The simplified version emphasises the common flow rate v_j by using the single port **R**:**rj** and the net affinity $A_j = A_j^f - A_j^r$

state $x_A = n_A$, the number of moles of substance A. Thus for substance A the corresponding **C** component has the equations

$$\dot{x}_A = v \qquad \text{mol}\,\text{s}^{-1} \tag{16.17}$$

$$\mu_A = RT \ln K_A x_A \; \text{J}\,\text{mol}^{-1} \tag{16.18}$$

The sole parameter is the *thermodynamic constant* K_A given by Eq. (16.16) with units of mol^{-1}; thus the argument of $K_A x_A$ of ln is dimensionless. The constant RT has units of J mol^{-1}.

16.2.4 The Bond Graph **Re** *Component*

The jth **Re** component of Fig. 16.1a has two ports with a common flow variable v_j and the associated effort variables are the forward and backward affinities A_j^f and A_j^r. As discussed by Gawthrop and Crampin [11] and based on the work of Van Rysselberghe [43] and Oster et al. [29, Sect. 5.1], the molar flow (or reaction rate) v_j is given by the *Marcelin–de Donder formula* in terms of A^f and A^r by

$$v_j = v_j^+ - v_j^- \tag{16.19}$$

$$\text{where } v_j^+ = \kappa_j e^{\frac{A_j^f}{RT}} \qquad \text{and} \qquad v_j^- = \kappa_j e^{\frac{A_j^r}{RT}} \tag{16.20}$$

where the *reaction constant* $\kappa_j \geq 0$. The exponential term in Eq. (16.19) of the **Re** component, coupled with the logarithmic term in Eq. (16.18) of the **C** component gives rise to sums of states being converted into products of states and gives rise to the *mass-action* form of the reaction flows.

An **Re** component with net affinity $A_j = A_j^f - A_j^r$ and molar flow v_j dissipates Gibbs energy at a rate $\dot{G}_j$ given by

$$\dot{G}_j = v_j \left(A_j^f - A_j^r \right) = \left(v_j^+ - v_j^- \right) \left(A_j^f - A_j^r \right) \tag{16.21}$$

Note that when $A_j = 0$ both $v_j = 0$ and $\dot{G}_j = 0$. Thus, to show that **Re** is dissipative, it is necessary to show that $\dot{G}_j$ is positive for all $v_j \neq 0$.

Using Eq. (16.20)

$$A_j^f = RT \ln \frac{v_j^+}{\kappa_j} \quad \text{and} \quad A_j^r = RT \ln \frac{v_j^-}{\kappa_j} \tag{16.22}$$

$$\text{and so } \dot{G}_j = RT \left(v_j^+ - v_j^- \right) \ln \frac{v_j^+}{v_j^-} \tag{16.23}$$

Equation (16.23) is given by Qian and Beard [36, Eq. (5)] and Polettini and Esposito [35].

From Eq. (16.20), both v_j^+ and v_j^- are positive and so it is possible to write $v_j^+ = \rho v_j^-$ where ρ is positive. Hence Eq. (16.23) can be rewritten as

$$\dot{G}_j = RT v_j^- (\rho - 1) \ln \rho \tag{16.24}$$

There are three possibilities: if $\rho > 1$ then both $(\rho - 1) > 0$ and $\ln \rho > 0$ and so, as $v_j^- > 0$, $\dot{G}_j > 0$; if $\rho < 1$ then both $(\rho - 1) < 0$ and $\ln \rho < 0$ and so, as $v_j^- > 0$, $\dot{G}_j > 0$; and, if $\rho = 1$, both $(\rho - 1) = 0$ and $\ln \rho = 0$ and $\dot{G}_j = 0$. Hence if $v_j \neq 0$, $\dot{G}_j > 0$ and if $v_j = 0$, $\dot{G}_j = 0$.

From Eqs. (16.10) and (16.11), the *enthalpy* dissipated by the resistor reappears as external *entropy* and the remainder of the Gibbs energy ($T \Delta S$) represents a change of internal entropy. Thus only the enthalpy portion of the Gibbs energy is dissipated as "heat of reaction"—more properly called enthalpy of reaction. This phenomenon has been called *entropy stripping* [41, 42].

The two ports of the reaction component **Re** of Fig. 16.1a are needed to separately convey the affinities A_j^f and A_j^r to Eq. (16.19). However, as pointed out by Karnopp [20], the one port **R** and **1** junction version of Fig. 16.1b more clearly shows that the molar flow v_j is common to both ports and thus mass is conserved. The **R** component can be suitably modulated to have the same effect as the **Re** component. The one port version of Fig. 16.1b will be used for the structural analysis of Sect. 16.3.

16.2.5 Example

Figure 16.2a corresponds to a biomolecular system with three species (A, B, C) and three reactions ($r1$, $r2$, $r3$) given by the chemical formulae

$$A \overset{1}{\rightleftharpoons} B \overset{2}{\rightleftharpoons} C \overset{3}{\rightleftharpoons} A \tag{16.25}$$

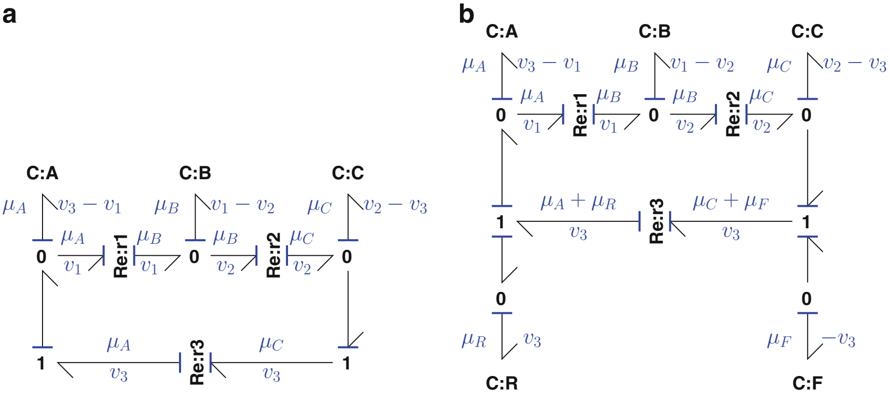


Fig. 16.2 Simple example. (**a**) $A \overset{1}{\rightleftharpoons} B \overset{2}{\rightleftharpoons} C \overset{3}{\rightleftharpoons} A$. The three **C** components represent the chemical species A, B and C respectively and the three **Re** components represent the corresponding reactions. These components are connected by bonds and **0** and **1** junctions which clearly show the cyclic nature of the reaction system. (**b**) Two additional species F and R are introduced and the third reaction of (**a**) is replaced by $C + F \overset{3}{\rightleftharpoons} A + R$. Causality has been completed with integral causality on the **C** components

Causality has been completed with integral causality on the **C** components using the sequential causality assignment procedure (SCAP) [21].[6]

The three **C** components correspond to the three species and are associated with states x_A, x_B and x_C and chemical potentials μ_A, μ_B and μ_C. The three reaction flows are v_1, v_2 and v_3 and are driven by the forward affinities $A_1^f = \mu_A$, $A_2^f = \mu_B$ and $A_3^f = \mu_C$ and by the reverse affinities $A_1^r = \mu_B$, $A_2^r = \mu_C$ and $A_3^r = \mu_A$. The state derivatives are

$$\dot{x}_A = v_3 - v_1 \tag{16.26}$$

$$\dot{x}_B = v_1 - v_2 \tag{16.27}$$

$$\dot{x}_C = v_2 - v_3 \tag{16.28}$$

and the chemical potentials are

$$\mu_A = RT \ln K_A x_A \tag{16.29}$$

$$\mu_B = RT \ln K_B x_B \tag{16.30}$$

$$\mu_C = RT \ln K_C x_C \tag{16.31}$$

[6] SCAP is used for all the examples in this chapter. Unless otherwise stated, all **C** components have integral causality.

and the reaction flows are

$$v_1 = \kappa_1 \left(e^{\frac{A_1^f}{RT}} - e^{\frac{A_1^r}{RT}} \right) = \kappa_1 \left(K_A x_A - K_B x_B \right) \tag{16.32}$$

$$v_2 = \kappa_2 \left(e^{\frac{A_2^f}{RT}} - e^{\frac{A_2^r}{RT}} \right) = \kappa_2 \left(K_B x_B - K_C x_C \right) \tag{16.33}$$

$$v_3 = \kappa_3 \left(e^{\frac{A_3^f}{RT}} - e^{\frac{A_3^r}{RT}} \right) = \kappa_3 \left(K_C x_C - K_A x_A \right) \tag{16.34}$$

These equations are *linear* because there is only one species on each side of each reaction.

Because of this linearity, the bond graph of Fig. 16.2a could have been represented using linear **C** components and replacing **Re** of Fig. 16.1a by the one port **R** and **1** junction version of Fig. 16.1b. Such an approach has two disadvantages: the resultant bond graph has no energy properties and would thus be a *pseudo bond graph* and such an approach is not scalable to more complex systems.

In particular, the extended system of Fig. 16.2b has the chemical equation of the third reaction replaced by

$$C + F \overset{3}{\rightleftharpoons} A + R \tag{16.35}$$

which has two species on each side of the reaction; and hence $A_3^f = \mu_C + \mu_f$ and $A_3^r = \mu_A + \mu_r$. Thus the reaction flow v_3 becomes

$$v_3 = \kappa_3 \left(e^{\frac{A_3^f}{RT}} - e^{\frac{A_3^r}{RT}} \right) = \kappa_3 \left(K_C K_F x_C x_F - K_A K_R x_A x_R \right) \tag{16.36}$$

This equation contains the product of states and is thus a *nonlinear* equation. This essential nonlinearity is a feature of biomolecular systems.

16.3 Structural Analysis

Although the properties of biomolecular systems arise from the non-linear behaviour of the **C** and **Re** components, the characteristics of each individual system depend on how these components are interconnected via the linear *junction structure* formed from bonds, **0** and **1** junctions.[7] Thus the analysis of such

[7] **TF** components representing reaction stoichiometry also occur—see example in Sect. 16.5.

junction structures reveals generic system properties which are independent of the parameters of the **C** and **Re** components.

Section 16.3.1 motivates the analysis using a hydraulic analogy of the simple biomolecular systems of Fig. 16.2. Section 16.3.2 looks at properties of *closed systems* where the junction structure transmits, but does not inject or consume energy. However, living systems are not *closed systems* but need to interact with the environment to replenish dissipated energy; hence Sect. 16.3.3 extends the analysis to *open systems* where such environmental interactions are modelled by *chemostats*. Section 16.3.4 shows how stoichiometric information can be deduced from the junction structure.

16.3.1 Motivation

Analogies are central to scientific thinking by providing a way for results and intuition form one physical domain to be transferred to another [24]. Bond graphs provide a formalism to discussing analogies. Thus, for example, the bond graph representing a network of biomolecular reactions has an electrical circuit representation [27]. Unfortunately, these circuits are quite complicated when more than one species is involved as substrate or product of a reaction—see Oster and Perelson [27, Fig. 6] for an example. However, for the purposes of motivation, the simple network of three reactions (16.25), modelled by the bond graph of Fig. 16.2a, is used as it does have simple analogies. Figure 16.3a gives an electrical circuit analogy and Fig. 16.3b gives a hydraulic analogy; the latter is used as a simple intuitive motivational example.

Consider the analogy of three open tanks of liquid Fig. 16.3b represented by the three **C** components connected by three narrow pipes represented by the three **Re**

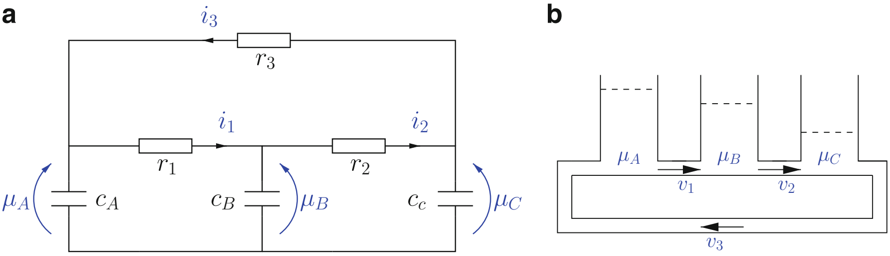


Fig. 16.3 Analogues of simple system. (**a**) An electric circuit analogue has three capacitors representing the three species and three resistors representing the three reactions; voltage is the analogue of chemical potential and current the analogue of molar flow rate. (**b**) A hydraulic analogy has three tanks of liquid representing the three species and three pipes representing the three reactions; pressure is the analogue of chemical potential and volumetric flow rate the analogue of molar flow rate. Linear analogies are only possible in simple cases such as this

components. μ_A, μ_B and μ_C then represent the pressures associated with the tanks and v_1, v_2 and v_3 the volumetric flow rates though the pipes.

In the linear case, the **C** components become linear and the **Re** components can be replaced by an **R** component and **1** junction as in Fig. 16.1 where in the case of **Re**:$\mathbf{r_1}$ the forward "affinity" or pressure $A_1^f = \mu_A$ and the reverse "affinity" or pressure $A_1^r = \mu_B$; the pressure drop across r_1 is $A_1 = A_1^f - A_1^r$. The following statements can be made about this particular system

1. The sum of the volumes of liquid in the three tanks (x_{tot}) is constant

$$x_A(t) + x_B(t) + x_C(t) = x_{\text{tot}} \tag{16.37}$$

   This is analogous to a *conserved moiety* in biochemistry.
2. As $A_1 + A_2 + A_3 = (\mu_A - \mu_B) + (\mu_B - \mu_C) + (\mu_C - \mu_A) = 0$, the sum of the three pressure drops is zero

$$A_1(t) + A_2(t) + A_3(t) = 0 \tag{16.38}$$

3. In the steady state of this closed system, there is no energy dissipation and so all flows are zero and thus the pressures are equal

$$\mu_A(\infty) = \mu_B(\infty) = \mu_C(\infty) \tag{16.39}$$

4. If the third pipe were replaced by a constant-flow pump with flow rate v_0, then in the steady state ($\dot{X} = 0$)

$$v_1(\infty) = v_2(\infty) = v_0 \tag{16.40}$$

These four statements are illustrated by the simulation of Fig. 16.4. Here all reaction constants and thermodynamics constants are unity ($\kappa_1 = \kappa_2 = \kappa_3 = K_A = K_B = K_C = 1$) and the initial states are $x_A = 3$, $x_B = 2.5$ and $x_C = 0.5$.

These four statements are obvious in the context of this simple system, the purpose of the next section is to show how these statements generalise to arbitrary biomolecular networks. The fourth statement involves the use of a pump. This requires an external source of energy and thus the resultant system is no longer closed. Thus the analysis is extended to include open as well as closed systems. The design of biomolecular pumps, and the source of their energy requirement, is key to understanding living systems. As will be discussed in Sect. 16.3.3, such pumps can be modelled using *chemostats* which are closely related to the bond graph effort source **Se**. Such pumps make use of the inherent non-linearity of biomolecular systems when multiple substrates and products of a reaction are present; in particular

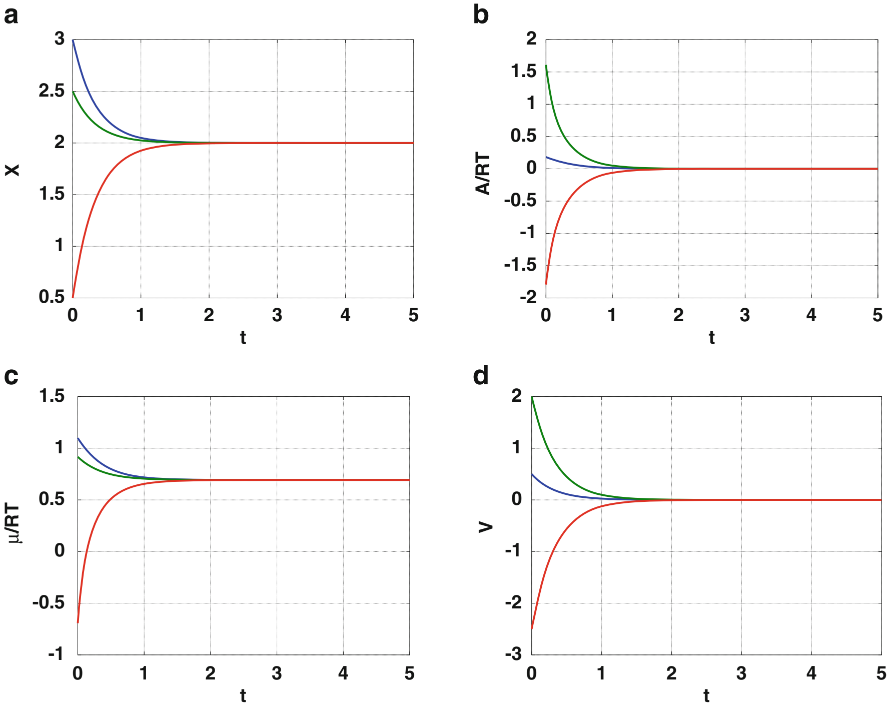


Fig. 16.4 Simulation of simple system. (**a**) The system states become constant in the steady state and their sum remains at the value 6 for all time: this corresponds to a *conserved moiety*. (**b**) The reaction affinities become zero in the steady state and their sum is 0 for all time. (**c**) The chemical potentials equalise in the steady state. (**d**) The reaction flows become zero in the steady state because this is a *closed* system

$ATP \rightleftharpoons ADP + P$[8] can be coupled multiplicatively to an underlying reaction to produce a pump; this is discussed further in Sect. 16.6.

16.3.2 Junction Structure of Closed Systems

This section is concerned with structural principles rather than details of component behaviour; hence, following Sect. 16.3.1, the **Re** component is replaced by a single-port **R** component and an associated **1** junction as in Fig. 16.1. The choice between

[8] *ATP* (Adenosine triphosphate) is the "fuel" which drives many biomolecular processes via its conversion to *ADP* (Adenosine diphosphate) and *P* inorganic phosphate [1].

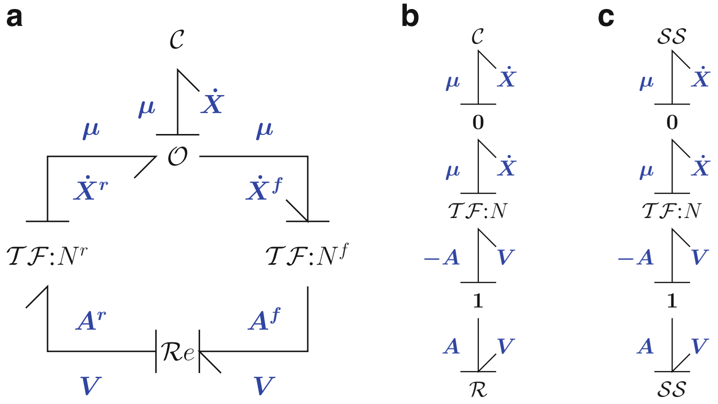


Fig. 16.5 General closed system with integral causality. (**a**) The general representation of a closed biomolecular system used by Gawthrop and Crampin [12]. (**b**) Simplified version with the two-port **Re** replaced by one-port **R** and **1** junction as in Fig. 16.1. (**c**) The **C** and **R** components are replaced by source-sensor (**SS**) components and *integral* causality is shown

the two representations is discussed by Karnopp [20]; in the context of this chapter, the **R** representation has the advantage that, via the **1** junction, it ensures continuity of flow.

The abstract representation of a closed system of Gawthrop et al. [13] and Gawthrop and Crampin [12] is given in Fig. 16.5a in terms of the **Re** reaction representation of Fig. 16.1a. The bond symbols $\rightharpoondown$ correspond to *vectors* of bonds; $\mathscr{C}$, $\mathscr{R}e$ and $\mathscr{O}$ correspond to arrays of **C**, **Re** and **0** components; the two $\mathscr{TF}$ components represent the intervening junction structure comprising bonds, **0** and **1** junctions and **TF** components. N^f and N^r are the forward and reverse stoichiometric matrices. Using the **R** reaction representation of Fig. 16.1b gives the simplified Fig. 16.5b where the stoichiometric matrix $N = N^r - N^f$.

The *multiport transformer* $\mathscr{TF}$:N has the n_X vector $\boldsymbol{\mu}$ as the effort variable paired with the n_X vector $\dot{\boldsymbol{X}}$ as flow variable on one port and the n_V vector $-\boldsymbol{A}$ as the effort variable paired with the n_V vector $\boldsymbol{V}$ as flow variable on the other port. It follows that

$$\dot{\boldsymbol{X}} = N\boldsymbol{V} \tag{16.41}$$

$$\boldsymbol{A} = -N^T\boldsymbol{\mu} \tag{16.42}$$

This multiport transformer abstracts the connections between the n_X **C** components and the n_V **R** components; it is independent of the properties of the **C** and **R** components. It represents the junction structure of the underlying bond graph. In this paper, the junction structure is revealed by replacing each **C** and **R** component by a *source-sensor* **SS** component [14, 15].[9] For example, the bond graph of Fig. 16.2a becomes the junction structure bond graph of Fig. 16.6a. The **SS** components have now preferred causality; in Fig. 16.6a the **SS** causality has been chosen to correspond to the preferred causality of Fig. 16.2a where the **C** components have integral causality (effort output) leading to flow output on each **R** component.

16.3.2.1 Structure Matrices

With the integral causality shown in Fig. 16.5c the junction structure *outputs* are the n_X **C** *flows* (rate of species change, $\dot{\boldsymbol{X}}$ mol s^{-1}) and the n_V **R** *efforts* (affinities, $\boldsymbol{A}$ J mol^{-1}). and the junction structure *inputs* are the n_X **C** *efforts* (chemical potentials, μ mol s^{-1}) and the n_V **R** *flows* (molar reaction flows, $\boldsymbol{V}$ mol s^{-1}). Combining the junction structure outputs into the vector $\boldsymbol{Y}$, and combining the junction structure inputs into the vector $\boldsymbol{U}$, Eqs. (16.41) and (16.42) can be rewritten in a more compact form as

$$\boldsymbol{Y} = S\boldsymbol{U} \tag{16.43}$$

$$\text{where } \boldsymbol{Y} = \left(\begin{array}{c}\dot{\boldsymbol{X}}\\ \hline \boldsymbol{A}\end{array}\right),\ \boldsymbol{U} = \left(\begin{array}{c}\boldsymbol{\mu}\\ \hline \boldsymbol{V}\end{array}\right) \tag{16.44}$$

$$\text{and } S = \left(\begin{array}{c|c}0_{n_X\times n_X} & N\\ \hline -N^T & 0_{n_V\times n_V}\end{array}\right) \tag{16.45}$$

The structure matrix S has two notable features:

1. As discussed, for example, by Karnopp et al. [21, Sect. 7.4] the *junction structure* matrix S is skew symmetric; this arises from the energy transmission properties of the multiport transformer.
2. The two block zero elements $0_{n_X\times n_X}$ and $0_{n_V\times n_V}$ arise because, in the junction structure, the effort and flow variables do not interact: biomolecular systems do *not* contain gyrators.

16.3.2.2 Derivative Causality

As was noted by Sueur and Dauphin-Tanguy [38, 39], investigating the junction structure with **C** and **I** components in *derivative causality* is the key to determining system properties. With this in mind, and noting that the only dynamic components are **C** components, the integral causality leading to S is replaced by maximising the

[9] The **SS** component is equivalent to the **EN** (environment) component introduced by Rosenberg and Andry [37].

number of **C** components in derivative causality. As noted by Sueur and Dauphin-Tanguy [39] with reference to their Fig. 2, the presence of causal loops requires a modification of this procedure; in particular, the causal loop must be broken by leaving a **C** component in integral causality.

This procedure will be referred to as *maximising derivative causality* in the sequel and leads to the concept of *derivative structure matrices*.

16.3.2.3 Derivative Structure Matrices

The vector $\boldsymbol{Y}_d$ contains the outputs of the network with maximum derivative causality and has four partitions:

1. $\boldsymbol{\mu}^D$ the chemical potential of the **SS** components corresponding to **C** in *derivative causality*.
2. $\dot{\boldsymbol{X}}^I$ the flow of the **SS** components corresponding to **C** remaining in *integral causality*.
3. $\boldsymbol{V}^D$ the flows of the **SS** components corresponding to **R** in *derivative causality*.
4. $\boldsymbol{A}^I$ the affinity of the **SS** components corresponding to **R** remaining in *integral causality*.

$\boldsymbol{Y}_d$ contains the same variables as $\boldsymbol{Y}$ but they may be in a different order. The vector $\boldsymbol{U}_d$ contains the covariables of $\boldsymbol{Y}_d$.

$\boldsymbol{Y}_d$ and $\boldsymbol{U}_d$ are related by the *derivative structure matrix* S_d where

$$\boldsymbol{Y}_d = S_d\boldsymbol{U}_d \tag{16.46}$$

$$\text{where } \boldsymbol{Y}_d = \begin{pmatrix} \boldsymbol{\mu}^D \\ \hline \dot{\boldsymbol{X}}^I \\ \hline \boldsymbol{V}^D \\ \hline \boldsymbol{A}^I \end{pmatrix}, \; \boldsymbol{U}_d = \begin{pmatrix} \dot{\boldsymbol{X}}^D \\ \hline \boldsymbol{\mu}^I \\ \hline \boldsymbol{A}^D \\ \hline \boldsymbol{V}^I \end{pmatrix} \tag{16.47}$$

$$\text{and } S_d = \left(\begin{array}{c|c|c|c} 0 & S_{\mu\mu} & S_{\mu A} & 0 \\ \hline S_{xx} & 0 & 0 & S_{xv} \\ \hline S_{vx} & 0 & 0 & S_{vv} \\ \hline 0 & S_{A\mu} & S_{AA} & 0 \end{array}\right) \tag{16.48}$$

As with S, S_d is skew symmetric because of energy considerations and the eight block zero elements arise as the effort and flow variables do not interact: biomolecular systems do *not* contain gyrators.

Example. In the case of the junction structure of Fig. 16.6a

$$N = \begin{pmatrix} -1 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix} \tag{16.49}$$

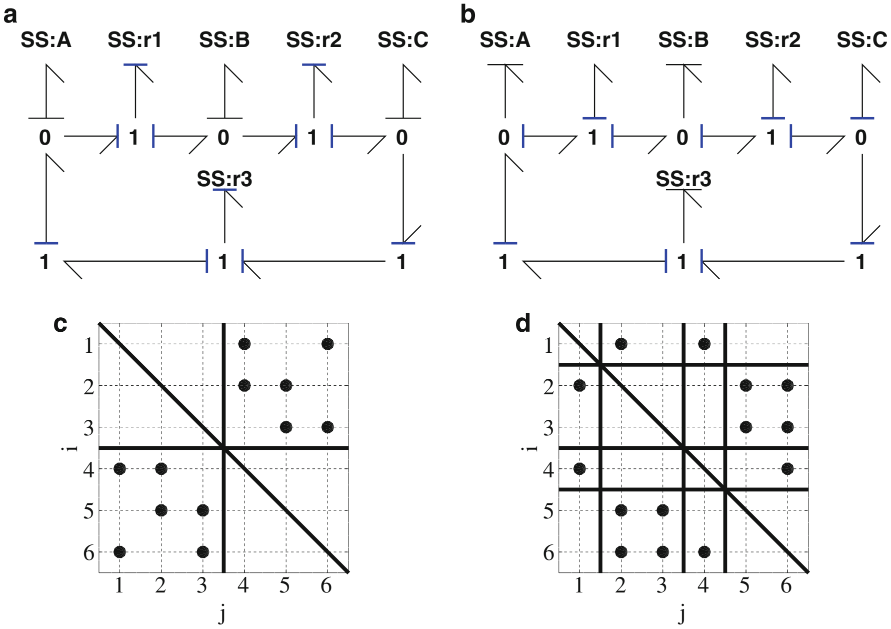


Fig. 16.6 Simple Closed System junction structure. (**a**) and (**b**) The junction structure corresponding to the bond graph of Fig. 16.2a with integral and maximum derivative causality respectively. (**c**) and (**d**) Alternative representation of the structure matrices S (16.43) and S_d (16.48) where *filled circle* indicates a non-zero entry and i and j the row and column indices respectively

and:

$$\boldsymbol{Y} = \begin{pmatrix} \dot{x}_A \\ \dot{x}_B \\ \dot{x}_C \\ \hline A_1 \\ A_2 \\ A_3 \end{pmatrix}, \; S = \left(\begin{array}{ccc|ccc} 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ \hline 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \end{array}\right), \; \boldsymbol{U} = \begin{pmatrix} \mu_A \\ \mu_B \\ \mu_C \\ \hline v_1 \\ v_2 \\ v_3 \end{pmatrix} \tag{16.50}$$

In the case of the junction structure of Fig. 16.6b with maximum derivative causality

$$\boldsymbol{Y}_d = \begin{pmatrix} \mu_A \\ \mu_B \\ \hline \dot{x}_C \\ \hline v_1 \\ v_2 \\ \hline A_3 \end{pmatrix}, \; S_d = \left(\begin{array}{cc|c|cc|c} 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ \hline -1 & -1 & 0 & 0 & 0 & 0 \\ \hline -1 & 0 & 0 & 0 & 0 & 1 \\ -1 & -1 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & -1 & -1 & 0 \end{array}\right), \; \boldsymbol{U}_d = \begin{pmatrix} \dot{x}_A \\ \dot{x}_B \\ \hline \mu_C \\ \hline A_1 \\ A_2 \\ \hline v_3 \end{pmatrix} \tag{16.51}$$

As an alternative to Eqs. (16.50) and (16.51), the structure matrices S and S_d can be visualised by the diagrams of Fig. 16.6c, d where the ij non-zero matrix elements are plotted as $\bullet$ in the ij location. The partition lines separating $\dot{\boldsymbol{X}}$ from $\boldsymbol{A}$ and $\boldsymbol{\mu}$ from $\boldsymbol{V}$ have been included for clarity along with the diagonal about which the matrix is skew symmetric. This representation is particularly useful for large systems when the matrix expression becomes unwieldy.

The four observations of Sect. 16.3.1 can now be explained in terms of the partitions of the S_d matrix (16.48) and (16.51).

1. From the second row partition

$$S_{xx} = \begin{pmatrix} -1 & -1 \end{pmatrix} \tag{16.52}$$

$$\text{hence } \dot{x}_C = -\dot{x}_A - \dot{x}_B \tag{16.53}$$

$$\text{and } \dot{x}_A + \dot{x}_B + \dot{x}_C = 0 \tag{16.54}$$

   Integrating (16.54) with respect to time gives Eq. (16.37).

2. From the fourth row partition

$$S_{AA} = \begin{pmatrix} -1 & -1 \end{pmatrix} \tag{16.55}$$

$$\text{hence } A_3 = -A_1 - A_2 \tag{16.56}$$

   Rearranging Eq. (16.56) gives Eq. (16.38).

3. From the first row partition

$$S_{\mu\mu} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \; S_{\mu A} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \tag{16.57}$$

$$\text{hence } \mu_A = A_1 + A_2 + \mu_C \tag{16.58}$$

$$\mu_B = A_2 + \mu_C \tag{16.59}$$

   In the steady state, the flows are zero $v_1 = v_2 = v_3 = 0$ and thus the affinities are also zero $A_1 = A_2 = A_3 = 0$; substituting into Eqs. (16.58) and (16.59) gives Eq. (16.39).

4. From the third row partition

$$S_{vv} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \; S_{vx} = \begin{pmatrix} -1 & 0 \\ -1 & -1 \end{pmatrix} \tag{16.60}$$

$$\text{hence } v_1 = -\dot{x}_A + v_3 \tag{16.61}$$

$$v_2 = -\dot{x}_A - \dot{x}_B + v_3 \tag{16.62}$$

Setting $\dot{x}_A = \dot{x}_B = 0$ gives Eq. (16.40). Of course, in this closed system, the three steady-state flows are actually zero; to obtain "pumping" and non-zero flows we now turn to consideration of *open* systems.

16.3.3 Junction Structure of Open Systems

All living systems are connected to their environment: mass and energy can enter and leave the system. Thus living systems are *open* systems and, for this reason, this section considers the junction structures of open systems. Following Gawthrop and Crampin [12], thermodynamically closed systems are converted to thermodynamically open systems using *chemostats*.[10]

Gawthrop and Crampin [12] also use *flowstats*.[11] But this idea is not pursued in this chapter.

As discussed by Gawthrop and Crampin [12], a chemostat generates a chemical potential μ which is not dependent on the flow covariable $\dot{x}$ and may be modelled as a **C** component with a fixed state. For the purposes of junction structure analysis it is convenient to represent the chemostat by a **C** component and a unit *effort amplifier* **AE** [15] component where the **AE** component has unit gain and infinite input impedance thus drawing zero flow from the **C** component; see Fig. 16.7a. The **AE** component has fixed causality as indicated in Fig. 16.7b; the input and output of the **AE** are explicitly indicated to avoid ambiguity. The **AE** component is *active* and thus represents the power source of an open system.

As in Sect. 16.3.2, the junction structure is obtained from the system bond graph by replacing each **C** and **R** component by a *source-sensor* **SS** component; this leaves the **AE** components associated with chemostats as part of the junction structure. This has a number of consequences:

1. The junction structure is no longer energy conserving; the chemostats insert energy into the system.
2. Equations (16.41) and (16.42) are replaced by:

$$\dot{\boldsymbol{X}} = N^{cd}\boldsymbol{V} \tag{16.63}$$

$$\boldsymbol{A} = -N^T\boldsymbol{\mu} \tag{16.64}$$

[10] The term *chemostat* was used by Polettini and Esposito [35] and is equivalent to the "concentration clamping" of Qian and Beard [36].

[11] The term *flowstat* is equivalent to "boundary flux injection" [36].

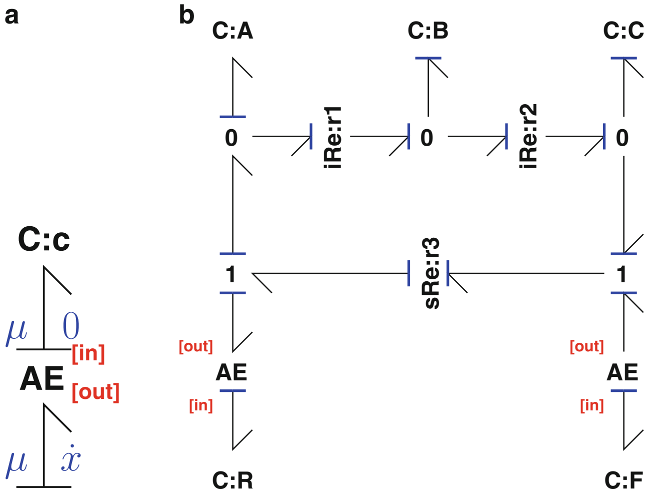


Fig. 16.7 Chemostats. (**a**) A *chemostat* is modelled by a **C** component and a **AE** effort amplifier; this means that the corresponding chemical potential μ is imposed on the rest of the system but the corresponding flow $\dot{x}$ does not effect the **C** component. (**b**) The simple example of Fig. 16.2b is modified with **AE** components to turn **C**:**F** and **C**:**R** into chemostats thus creating an open system from a closed system

where N^{cd} is N with rows corresponding to the chemostats removed [12, Sect. 3.4]. The structure matrix S is no longer of the skew symmetric form of Eq. (16.45) but is rather of the form

$$S = \left(\begin{array}{c|c} 0 & N^{cd} \\ \hline -N^T & 0 \end{array} \right) \tag{16.65}$$

3. When maximising derivative causality, the possible causal patterns of the junction structure are constrained by the fixed causality of the **AE** components.

16.3.3.1 Example (Chemostats)

The closed system of Fig. 16.6 is turned into the open system of Fig. 16.7b where the flow v_3 is "pumped" by the addition of a forward-driving chemostat **C**:**F** and a reverse-driving chemostat **C**:**R** arranged so that there is no net mass inflow to the system. This is the chemostatic version of the chemical equation (16.35) with the bond graph of Fig. 16.2b. Figure 16.8a shows the corresponding junction structure in integral causality and Fig. 16.8b with maximum derivative causality.

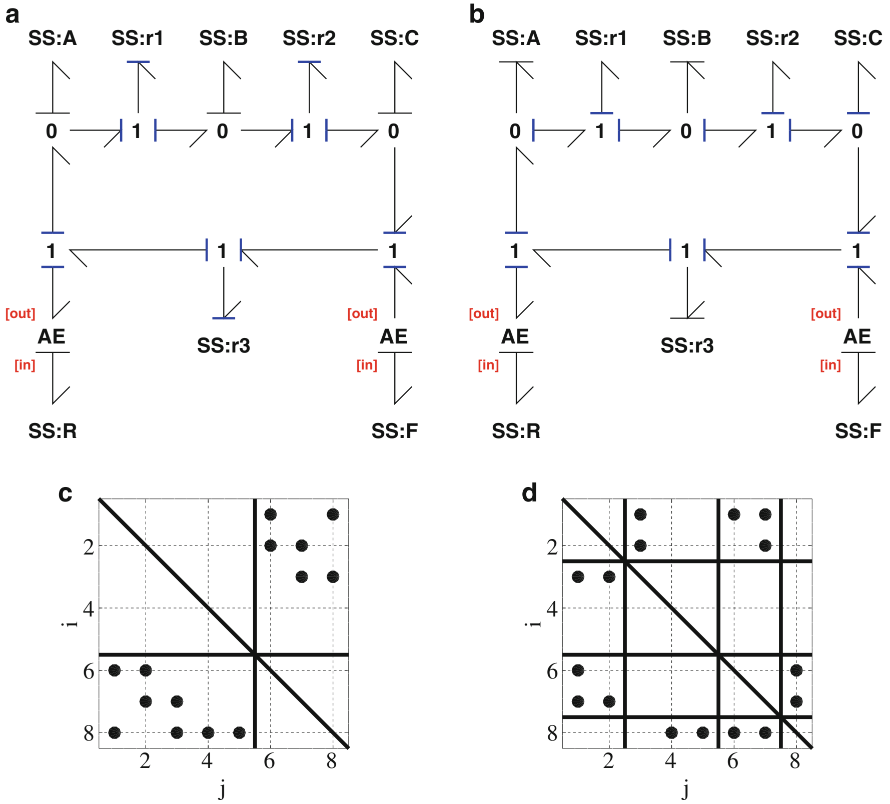


Fig. 16.8 Simple open system junction structure. (**a**) and (**b**) The junction structure corresponding to the bond graph of Fig. 16.7b with integral and maximum derivative causality respectively. (**c**) and (**d**) Alternative representation of the structure matrices S (16.43) and S_d (16.48) where *filled circle* indicates a non-zero entry and i and j the row and column indices respectively. Note that these matrices are no longer skew-symmetric due to the **AE** components and that the driving term $S_{A\mu}$ is no longer zero

The corresponding structure matrix S is

$$\boldsymbol{Y} = \begin{pmatrix} \dot{x}_A \\ \dot{x}_B \\ \dot{x}_C \\ \dot{x}_F \\ \dot{x}_R \\ \hline A_1 \\ A_2 \\ A_3 \end{pmatrix}, \; S = \left(\begin{array}{ccccc|ccc} 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & -1 & 0 & 0 & 0 \end{array}\right), \; \boldsymbol{U} = \begin{pmatrix} \mu_A \\ \mu_B \\ \mu_C \\ \mu_F \\ \mu_R \\ \hline v_1 \\ v_2 \\ v_3 \end{pmatrix} \qquad (16.66)$$

The corresponding derivative-causality structure matrix S_d is

$$\boldsymbol{Y}_d = \begin{pmatrix} \mu_A \\ \mu_B \\ \hline \dot{x}_C \\ \dot{x}_F \\ \dot{x}_R \\ \hline v_1 \\ v_2 \\ \hline A_3 \end{pmatrix}, \quad S_d = \left(\begin{array}{cc|ccc|cc|c} 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ \hline -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 1 & -1 & -1 & -1 & 0 \end{array}\right), \quad \boldsymbol{U}_d = \begin{pmatrix} \dot{x}_A \\ \dot{x}_B \\ \hline \mu_C \\ \mu_F \\ \mu_R \\ \hline A_1 \\ A_2 \\ \hline v_3 \end{pmatrix} \tag{16.67}$$

Again, S (16.66) and S_d (16.67) can be visualised using Fig. 16.8c, d.

S_d for the open system (16.67) differs from that from the closed system (16.51) in a number of ways.

1. The fourth row partition contains the driving term $\mu_f - \mu_r$ in the second column partition.
2. The two zero rows in the second row partition correspond to the two chemostats; in addition to the conserved moiety of Eq. (16.37), the two chemostats correspond to the two conserved moieties x_F and x_R.

Figure 16.9 gives a simulation of the simple system with chemostatic pump. Like the simulation shown in Fig. 16.4, all quantities tend to constant values; but, unlike the simulation shown in Fig. 16.4, the flows (v) and affinities (A) tend to non-zero constant values: the two chemostats **C**:**F** and **C**:**R** act as a pump.

16.3.4 Stoichiometric Information and Simulation

The *stoichiometric matrix* N of a biomolecular network is the matrix that relates the reaction flows described by the vector $\boldsymbol{V}$ to the rate of change of species described by the vector $\dot{\boldsymbol{X}}$ as in Eq. (16.41). As discussed in the text books of Palsson [30], Alon [2] and Klipp et al. [22], analysis of the stoichiometric matrix yields useful information about the properties of the underlying biomolecular network. In particular the *null space* or *kernel* space of both N and N^T are of interest and are defined by the corresponding kernel, or subspace, matrices. The null space of N relates to those linear combinations of reaction flows which can be non-zero when $\dot{\boldsymbol{X}}$ is zero; the null space of N^T relates to *conserved moieties*: those linear combinations of $\boldsymbol{X}$ which remain constant.

The structure matrix S_d (16.48), corresponding to maximum derivative causality, contains eight non-zero submatrices. This section provides interpretations of these matrices and how they relate to the kernel matrices of the stoichiometric matrix.

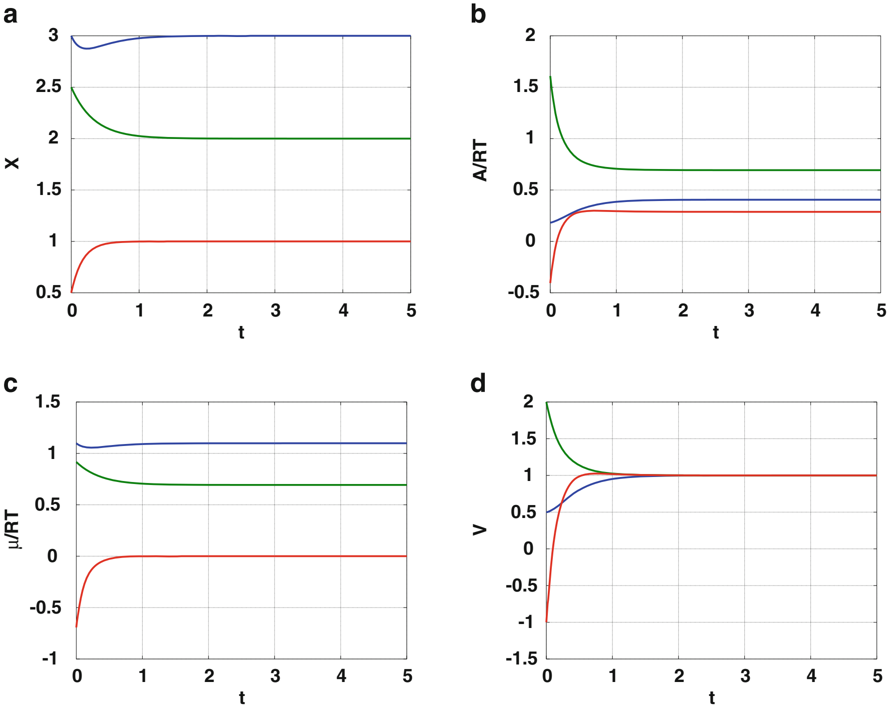


Fig. 16.9 Simulation of simple system with chemostatic pump. $K_F = K_R = 1$, $x_F = 4$ and $x_R = 1$ (**a**) As in the closed case, the system states become constant in the steady state and their is sum 6 for all time. (**b**) The reaction affinities become constant in the steady state and their sum is $\ln K_F x_F - \ln K_R x_R = 1.386$ for all time. (**c**) The chemical potentials no longer equalise in the steady state. (**d**) The reaction flows are not zero in the steady state as this is an open system with F and R acting as a pump

16.3.4.1 State Reconstruction

The vectors Y_d and U_d (16.47) partition the state vector derivative $\dot{\boldsymbol{X}}$ into $\dot{\boldsymbol{X}}^I$ and $\dot{\boldsymbol{X}}^D$ and the flow vector $\boldsymbol{V}$ into $\boldsymbol{V}^I$ and $\boldsymbol{V}^D$. This decomposition process can be written as

$$\dot{\boldsymbol{X}}^I = T_{xi}\dot{\boldsymbol{X}},\ \dot{\boldsymbol{X}}^D = T_{xd}\dot{\boldsymbol{X}} \tag{16.68}$$

$$\boldsymbol{V}^I = T_{vi}\boldsymbol{V},\ \boldsymbol{V}^D = T_{vd}\boldsymbol{V} \tag{16.69}$$

Equations (16.68) and (16.69) can be rewritten as:

$$\begin{pmatrix} \dot{\boldsymbol{X}}^I \\ \hline \dot{\boldsymbol{X}}^d \end{pmatrix} = T_x\dot{\boldsymbol{X}} \text{ where } T_x = \begin{pmatrix} T_{xi} \\ \hline T_{xd} \end{pmatrix} \tag{16.70}$$

$$\text{and } \left(\frac{\boldsymbol{V}^I}{\boldsymbol{V}^d}\right) = T_v \boldsymbol{V} \text{ where } T_v = \left(\frac{T_{vi}}{T_{vd}}\right) \tag{16.71}$$

Because T_x and T_v permute the elements of $\dot{\boldsymbol{X}}$ and $\boldsymbol{V}$, they are unitary matrices and therefore

$$T_x^{-1} = T_x^T = \left(T_{xi}^T \middle| T_{xd}^T\right) \tag{16.72}$$

$$\text{and } T_v^{-1} = T_v^T = \left(T_{vi}^T \middle| T_{vd}^T\right) \tag{16.73}$$

It follows that $\dot{\boldsymbol{X}}$ and $\boldsymbol{V}$ can be recomposed as

$$\dot{\boldsymbol{X}} = T_{xi}^T \dot{\boldsymbol{X}}^I + T_{xd}^T \dot{\boldsymbol{X}}^D \tag{16.74}$$

$$\boldsymbol{V} = T_{vi}^T \boldsymbol{V}^I + T_{vd}^T \boldsymbol{V}^D \tag{16.75}$$

Using (16.74) and the second partition of structure matrix S_d (16.48), $\dot{\boldsymbol{X}}$ can be expressed in terms of $\dot{\boldsymbol{X}}^D$ as

$$\begin{aligned}\dot{\boldsymbol{X}} &= T_{xi}^T \left(S_{xx}\dot{\boldsymbol{X}}^D + S_{xv} V^I\right) + T_{xd}^T \dot{\boldsymbol{X}}^D \\ &= L_{Xx}\dot{\boldsymbol{x}} + L_{Xv} V^I \end{aligned} \tag{16.76}$$

$$\text{where } \dot{\boldsymbol{x}} = \dot{\boldsymbol{X}}^D,\ L_{Xx} = \left(T_{xi}^T S_{xx} + T_{xd}^T\right),\ L_{Xv} = T_{xi}^T S_{xv} \tag{16.77}$$

When $L_{Xv} = 0$ (as in all the examples in this chapter)

$$\dot{\boldsymbol{X}} = L_{Xx}\dot{\boldsymbol{x}} \tag{16.78}$$

Integrating with respect to time gives

$$\boldsymbol{X} = L_{Xx}\boldsymbol{x} + G_X \boldsymbol{X}_0 \tag{16.79}$$

$$\text{where } G_X = I_{n_X \times n_X} - L_{Xx} T_{xd} \tag{16.80}$$

$$\text{and } \boldsymbol{X}_0 = \boldsymbol{X}(0) \tag{16.81}$$

As discussed by Gawthrop and Crampin [11, Sect. 4(c)], Eq. (16.79) is useful because it allows the system ODE to be solved for $\boldsymbol{x}$ which is of lower dimension than $\boldsymbol{X}$ and avoids issues with conserved moieties. For compatibility with previous work, and with reference to Eq. (16.68), it is convenient to define

$$L_{xX} = T_{xd} \tag{16.82}$$

$$\text{so that } \boldsymbol{x} = L_{xX}\boldsymbol{X} \tag{16.83}$$

16.3.4.2 Kernel Matrices

In the particular case that $S_{xv} = 0$, it follows from Eq. (16.47) and the second row partition of (16.48) that

$$\dot{\boldsymbol{X}}^I - S_{xx}\dot{\boldsymbol{X}}^D = 0 \tag{16.84}$$

Using the decomposition equations (16.68), (16.84) gives

$$G^{cd}\dot{\boldsymbol{X}} = 0 \tag{16.85}$$

$$\text{where } G^{cd} = T_{xi} - S_{xx}T_{xd} \tag{16.86}$$

Using Eq. (16.63), it follows from Eq. (16.85) that:

$$G^{cd}N^{cd}\boldsymbol{V} = 0 \tag{16.87}$$

As Eq. (16.87) must be true for all $\boldsymbol{V}$, it follows that

$$G^{cd}N^{cd} = 0 \tag{16.88}$$

Hence G^{cd} is the left kernel matrix of N^{cd}.

In the steady state, $\dot{\boldsymbol{X}}^D = 0$ and so, using the third row partition of (16.48)

$$\boldsymbol{V}^D = S_{vv}\boldsymbol{V}^I \tag{16.89}$$

hence, using the recomposition equation (16.75):

$$\boldsymbol{V} = K^{cd}\boldsymbol{V}^I \tag{16.90}$$

$$\text{where } K^{cd} = T_{vi}^T + T_{vd}^T S_{vv} \tag{16.91}$$

Using Eq. (16.90), the steady-state condition also implies that

$$\dot{\boldsymbol{X}} = N^{cd}\boldsymbol{V} = N^{cd}K^{cd}\boldsymbol{V}^I = 0 \tag{16.92}$$

As this must be true for all $\boldsymbol{V}^I$ it follows that

$$N^{cd}K^{cd} = 0 \tag{16.93}$$

and thus K^{cd} is the right kernel matrix of N^{cd}.

Kernel matrices can be found numerically from N^{cd} using Gaussian elimination. However, the approach here gives a clear physical derivation of the kernel matrices using causality arguments.

Example. The open system of Sect. 16.3.3 with the junction structure of Fig. 16.8 has structure matrices S and S_d given by Eqs. (16.66) and (16.67). In this case

$$T_{xi} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},\ T_{xd} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix},\ L_{xX} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{16.94}$$

$$L_{Xx} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix},\ G_X = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{16.95}$$

L_{xX} (16.94) reflects the fact that $\boldsymbol{x}$ contains the first two states x_A and x_B of $\boldsymbol{X}$ and L_{xX} (16.95) reflects the fact that the third state x_C is related to the first two via a conserved moiety as does the third row of G_X (16.95). The last two rows of G_X (16.95) correspond to the constant states of the two chemostats.

The two kernel matrices are

$$G^{cd} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{16.96}$$

$$K^{cd} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tag{16.97}$$

The first row of G^{cd} (16.96) again reflects the conserved moiety and the last two rows correspond to the two chemostats. The single column of K^{cd} (16.97) corresponds to the pathway though the three reaction components.

16.4 Example: Biomolecular Cycle

In his classic monograph, "Free energy transduction and biochemical cycle kinetics" Hill [17] discusses how the concentration difference of a species M existing both outside (M_o) and inside (M_i) a membrane can be used to pump another species L from inside (L_i) to outside (L_o) the membrane. This cycle uses a large protein molecule with two con formations E and E^*[12] the former allowing successive

[12] A protein molecule with a given chemical composition may have many different geometric "shapes" or conformations with different Gibbs energy—this is the basis of much cell biology [1].

binding to M_i and L_i and the latter to M_o and L_o. This is represented by seven reactions

$$M_i + E \overset{em}{\rightleftharpoons} EM \tag{16.98}$$

$$L_i + EM \overset{lem}{\rightleftharpoons} LEM \tag{16.99}$$

$$LEM \overset{lesm}{\rightleftharpoons} LE^\star M \tag{16.100}$$

$$LE^\star M \overset{esm}{\rightleftharpoons} L_o + E^\star M \tag{16.101}$$

$$E^\star M \overset{es}{\rightleftharpoons} M_o + E^\star \tag{16.102}$$

$$E^\star \overset{e}{\rightleftharpoons} E \tag{16.103}$$

$$EM \overset{\text{slip}}{\rightleftharpoons} E^\star M \tag{16.104}$$

where the last reaction is the so-called slip term. The corresponding bond graph appears in Fig. 16.10a where $E^\star$ is replaced by Es. The bond graph clearly shows the cyclic structure of the chemical reactions (16.98)–(16.104) and is topologically similar to the diagram of Hill [17, Fig. 1.2(a)]. As discussed by Hill [17], the four species M_o, M_i, L_o and L_i are assumed to have constant concentration: therefore they are modelled by four chemostats with the corresponding **AE** components.

The partitions of $\boldsymbol{Y}$ corresponding to Eq. (16.43) are

$$\dot{\boldsymbol{X}} = \left(\dot{x}_{EM}\ \dot{x}_{LEM}\ \dot{x}_{Es}\ \dot{x}_{EsM}\ \dot{x}_{LEsM}\ \dot{x}_{Li}\ \dot{x}_{Lo}\ \dot{x}_{Mi}\ \dot{x}_{Mo}\ \dot{x}_{E}\right)^T \tag{16.105}$$

$$\boldsymbol{A} = \left(A_{em}\ A_{lem}\ A_{es}\ A_{esm}\ A_{\text{slip}}\ A_e\ A_{lesm}\right)^T \tag{16.106}$$

and the partitions of $\boldsymbol{U}$ corresponding to Eq. (16.43) are

$$\boldsymbol{\mu} = \left(\mu_{EM}\ \mu_{LEM}\ \mu_{Es}\ \mu_{EsM}\ \mu_{LEsM}\ \mu_{Li}\ \mu_{Lo}\ \mu_{Mi}\ \mu_{Mo}\ \mu_{E}\right)^T \tag{16.107}$$

$$\boldsymbol{V} = \left(v_{em}\ v_{lem}\ v_{es}\ v_{esm}\ v_{\text{slip}}\ v_e\ v_{lesm}\right)^T \tag{16.108}$$

The partitions of $\boldsymbol{Y}_d$ corresponding to Eq. (16.47) are

$$\boldsymbol{\mu}^D = \left(\mu_{EM}\ \mu_{LEM}\ \mu_{Es}\ \mu_{EsM}\ \mu_{LEsM}\right)^T \tag{16.109}$$

$$\dot{\boldsymbol{X}}^I = \left(\dot{x}_{Li}\ \dot{x}_{Lo}\ \dot{x}_{Mi}\ \dot{x}_{Mo}\ \dot{x}_{E}\right)^T \tag{16.110}$$

$$\boldsymbol{V}^D = \left(v_{em}\ v_{lem}\ v_{es}\ v_{esm}\ v_{\text{slip}}\right)^T \tag{16.111}$$

$$\boldsymbol{A}^I = \left(A_e\ A_{lesm}\right)^T \tag{16.112}$$

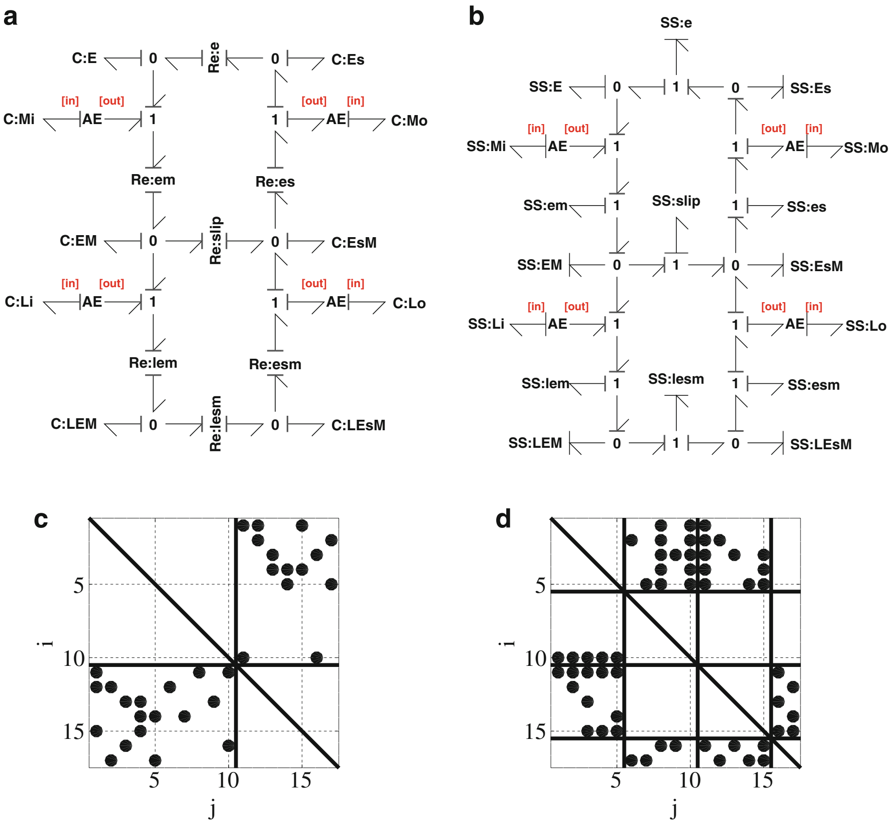


Fig. 16.10 Biomolecular Cycle. This example is taken from Hill [17] and shows how one species can pump another across a membrane. (**a**) System bond graph. (**b**) Junction structure with maximum derivative causality. (**c**) and (**d**) Alternative representation of the structure matrices S (16.43) and S_d (16.48) where *filled circle* indicates a non-zero entry and i and j the row and column indices respectively

and the partitions of $\boldsymbol{U}_d$ corresponding to Eq. (16.47) are

$$\dot{\boldsymbol{X}}^D = \left(\dot{x}_{EM}\ \dot{x}_{LEM}\ \dot{x}_{Es}\ \dot{x}_{EsM}\ \dot{x}_{LEsM}\right)^T \tag{16.113}$$

$$\boldsymbol{\mu}^I = \left(\mu_{Li}\ \mu_{Lo}\ \mu_{Mi}\ \mu_{Mo}\ \mu_E\right)^T \tag{16.114}$$

$$\boldsymbol{A}^D = \left(A_{em}\ A_{lem}\ A_{es}\ A_{esm}\ A_{\text{slip}}\right)^T \tag{16.115}$$

$$\boldsymbol{V}^I = \left(v_e\ v_{lesm}\right)^T \tag{16.116}$$

The 17 × 17 matrix S_d is summarised in Fig. 16.10d. The relevance of the submatrices S_{xx}, S_{AA}, $S_{A\mu}$ and S_{vv} are now examined in turn.

1. From the second row partition

$$S_{xx} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 \end{pmatrix} \tag{16.117}$$

the first four rows of S_{xx} correspond to the four chemostats *Li*, *Lo*, *Mi* and *Mo*.

$$\dot{x}_{Li} = \dot{x}_{Lo} = \dot{x}_{Mi} = \dot{x}_{Mo} = 0 \tag{16.118}$$

The fifth row of S_{xx} corresponds to the conserved moiety

$$\dot{x}_e + \dot{x}_{em} + \dot{x}_{lem} + \dot{x}_{es} + \dot{x}_{esm} + \dot{x}_{lesm} = 0 \tag{16.119}$$

2. From the fourth row partition

$$S_{AA} = \begin{pmatrix} -1 & 0 & -1 & 0 & -1 \\ 0 & -1 & 0 & -1 & 1 \end{pmatrix} \text{ and } S_{A\mu} = \begin{pmatrix} 0 & 0 & 1 & -1 & 0 \\ 1 & -1 & 0 & 0 & 0 \end{pmatrix} \tag{16.120}$$

the first row of S_{AA}, together with the first row of $S_{A\mu}$, corresponds to the sum of the affinities around the upper loop of the biomolecular cycle of Fig. 16.10a being equal to the weighted sum of the relevant chemostat potentials

$$A_e + A_{em} + A_{es} + A_{\text{slip}} = \mu_{Mi} - \mu_{Mo} \tag{16.121}$$

the second row of S_{AA}, together with the second row of $S_{A\mu}$, corresponds to the weighted sum of the affinities around the lower loop of the biomolecular cycle of Fig. 16.10a being equal to the weighted sum of the relevant chemostat potentials

$$A_{lesm} + A_{lem} + A_{esm} - A_{\text{slip}} = \mu_{li} - \mu_{lo} \tag{16.122}$$

3. From the third row partition

$$S_{vv} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & -1 \end{pmatrix} \tag{16.123}$$

In the steady state ($\dot{X} = 0$). In this case the first four rows of the first column of S_{vv} correspond to the three steady-state reaction flows in the upper loop being equal

$$v_{em} = v_{es} = v_e \tag{16.124}$$

Similarly, the first four rows of the second column of S_{vv} correspond to the three steady-state reaction flows in the lower loop being equal

$$v_{lem} = v_{esm} = v_{lesm} \tag{16.125}$$

and the fifth row of S_{vv} corresponds to the slip flow being the difference of the two loop flows

$$v_{\text{slip}} = v_e - v_{lesm} \tag{16.126}$$

16.5 Example: Glycolysis

Metabolism converts energy from the environment into the fuel that drives living processes. One example of metabolism is *glycolysis* which converts the chemical energy stored in glucose to the chemical energy stored in ATP (which, as discussed in Footnote 8, fuels biomolecular systems) and NADH.[13] Detailed mathematical models of glycolysis are given by Lambeth and Kushmerick [23] and Beard [4].

As bond graphs focus on energy transduction, it follows that bond graphs provide a natural method to model metabolism. Gawthrop et al. [13] develop a hierarchical bond graph model of metabolism based on that of Lambeth and Kushmerick [23] and highlight the advantages of the bond graph approach. This example looks at a simplified model of glycolysis used by Cloutier et al. [8] in the context of metabolism in the human brain and illustrates the junction structure approach to analysing the key properties of glycolysis.

The partitions of Y are

$$\dot{\boldsymbol{X}} = \left(\dot{x}_{G6P}\ \dot{x}_{F6P}\ \dot{x}_{GAP}\ \dot{x}_{PEP}\ \dot{x}_{ATP}\ \dot{x}_{ADP}\ \dot{x}_{NAD}\ \dot{x}_{NADH}\ \dot{x}_{GLC}\ \dot{x}_{PYR}\right)^T \tag{16.127}$$

$$\boldsymbol{A} = \left(A_{PGI}\ A_{PFK}\ A_{PGK}\ A_{PK}\ A_{HK}\right)^T \tag{16.128}$$

and the partitions of U are

$$\boldsymbol{\mu} = \left(\mu_{G6P}\ \mu_{F6P}\ \mu_{GAP}\ \mu_{PEP}\ \mu_{ATP}\ \mu_{ADP}\ \mu_{NAD}\ \mu_{NADH}\ \mu_{GLC}\ \mu_{PYR}\right)^T \tag{16.129}$$

$$\boldsymbol{V} = \left(v_{PGI}\ v_{PFK}\ v_{PGK}\ v_{PK}\ v_{HK}\right)^T \tag{16.130}$$

[13] NADH (reduced nicotinamide adenine dinucleotide) is an electron transporter within biomolecular processes [1].

The partitions of Y_d are

$$\boldsymbol{\mu}^D = \left(\mu_{G6P}\ \mu_{F6P}\ \mu_{GAP}\ \mu_{PEP}\right)^T \tag{16.131}$$

$$\dot{\boldsymbol{X}}^I = \left(\dot{x}_{ATP}\ \dot{x}_{ADP}\ \dot{x}_{NAD}\ \dot{x}_{NADH}\ \dot{x}_{GLC}\ \dot{x}_{PYR}\right)^T \tag{16.132}$$

$$\boldsymbol{V}^D = \left(v_{PGI}\ v_{PFK}\ v_{PGK}\ v_{PK}\right)^T \tag{16.133}$$

$$\boldsymbol{A}^I = \left(A_{HK}\right)^T \tag{16.134}$$

and the partitions of U_d are

$$\dot{\boldsymbol{X}}^D = \left(\dot{x}_{G6P}\ \dot{x}_{F6P}\ \dot{x}_{GAP}\ \dot{x}_{PEP}\right)^T \tag{16.135}$$

$$\boldsymbol{\mu}^I = \left(\mu_{ATP}\ \mu_{ADP}\ \mu_{NAD}\ \mu_{NADH}\ \mu_{GLC}\ \mu_{PYR}\right)^T \tag{16.136}$$

$$\boldsymbol{A}^D = \left(A_{PGI}\ A_{PFK}\ A_{PGK}\ A_{PK}\right)^T \tag{16.137}$$

$$\boldsymbol{V}^I = \left(v_{HK}\right)^T \tag{16.138}$$

The 15 × 15 matrix S_d summarised in Fig. 16.11d. The relevance of the submatrices S_{xx}, S_{AA}, $S_{A\mu}$ and S_{vv} are now examined in turn.

1. From the second row partition, S_{xx} has six zero-valued rows corresponding to the six chemostats *ATP*, *ADP*, *NAD*, *NADH*, *GLY* and *PYR*

$$\dot{x}_{ATP} = \dot{x}_{ADP} = \dot{x}_{NAD} = \dot{x}_{NADH} = \dot{x}_{GLY} = \dot{x}_{PYR} = 0 \tag{16.139}$$

   Unlike Example 16.4, there are no other conserved moieties in this open system.
2. From the fourth row partition

$$S_{AA} = \left(-1\ -1\ -2\ -2\right) \tag{16.140}$$

$$\text{and } S_{A\mu} = \left(-2\ 2\ 2\ -2\ 1\ -2\right) \tag{16.141}$$

   S_{AA}, together with $S_{A\mu}$, corresponds to the sum of the affinities along the glycolytic pathway being equal to the weighted sum of the relevant chemostat potentials

$$\begin{aligned} A_{PGI} + A_{PFK} + 2A_{PGK} + 2A_{PK} =& \mu_{GLC} - 2\mu_{PYR} \\ & - 2(\mu_{ATP} - \mu_{ADP}) - 2(\mu_{NADH} - \mu_{NAD}) \end{aligned} \tag{16.142}$$

   The right-hand side of this equation must be positive for glycolysis to proceed; thus the difference of the chemical potential of *GLC* (μ_{GLC}) and twice that of *PYR* ($2\mu_{PYR}$) must be sufficient to drive the creation of *ATP* from *ADP* and *NADH* from *NAD*.

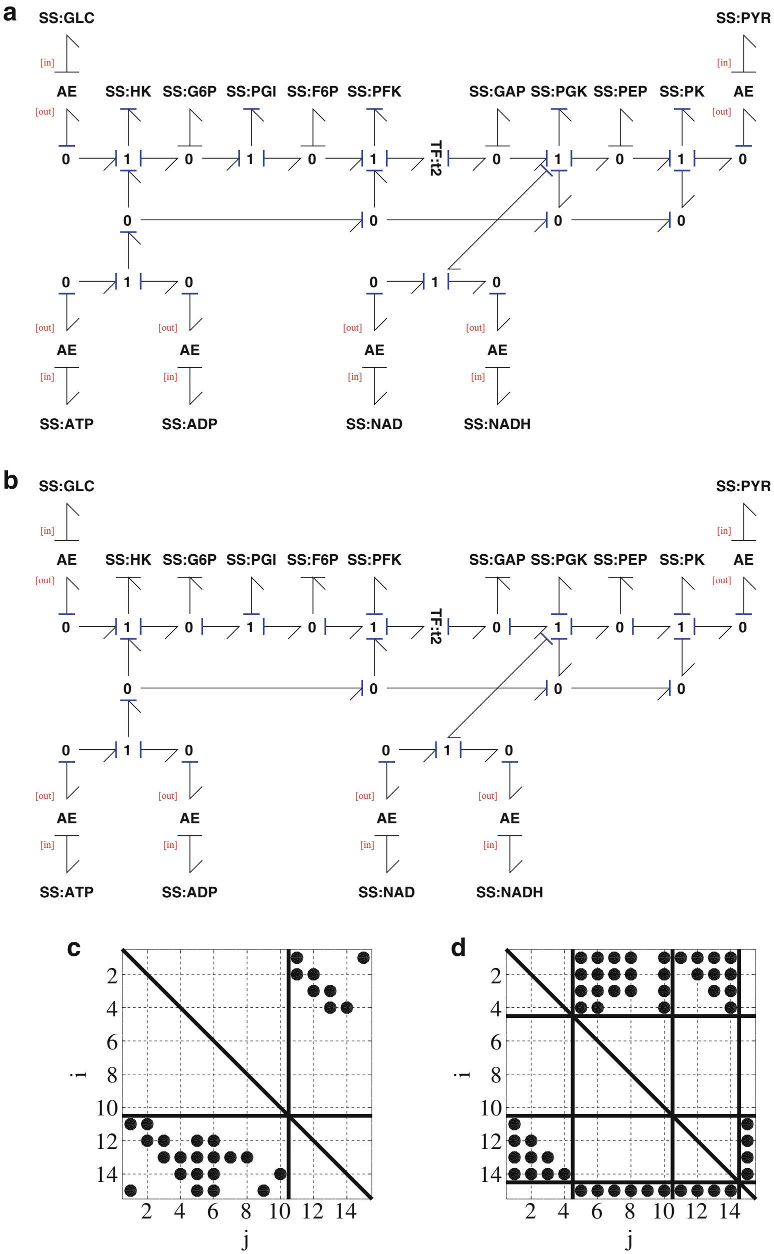


Fig. 16.11 Glycolysis. This simplified model of glycolysis corresponds to that of Cloutier et al. [8]. (**a**) System bond graph. (**b**) Junction structure with maximum derivative causality. (**c**) and (**d**) Alternative representation of the structure matrices S (16.43) and S_d (16.48) where *filled circle* indicates a non-zero entry and i and j the row and column indices respectively

3. From the third row partition

$$S_{vv} = \begin{pmatrix} 1 & 1 & 2 & 2 \end{pmatrix}^T \tag{16.143}$$

In the steady state ($\dot{\boldsymbol{X}} = 0$). In this case S_{vv} implies that, in the steady state, the reaction flows are related by

$$v_{PGI} = v_{PFK} = v_{HK} \text{ and } v_{PGK} = v_{PK} = 2v_{HK} \tag{16.144}$$

16.6 Example: Phosphorylation/Dephosphorylation

The phosphorylation/dephosphorylation reaction network is a standard building block of biomolecular systems. It is treated in detail by Beard and Qian [5, Sect. 5.1.1] and discussed by Gawthrop and Crampin [11, 12]. It comprises four reactions

$$\begin{aligned} M + ATP + E_1 &\overset{1}{\rightleftharpoons} C_1 & \qquad C_1 &\overset{2}{\rightleftharpoons} + ADP + MP + E_1 \\ MP + E_2 &\overset{3}{\rightleftharpoons} C_2 & \qquad C_2 &\overset{4}{\rightleftharpoons} M + P + E_2 \end{aligned} \tag{16.145}$$

In this example there are three chemostats corresponding to ATP, ADP and P; these act as the driver of the system.

The bond graph junction structure representation is given in Fig. 16.12b which is shown with derivative causality. The corresponding structure matrices S and S_d can be visualised using Fig. 16.12c, d. Note that the last three rows of N^{cd} correspond to the three chemostats and are zero.

The partitions of Y_d are

$$\boldsymbol{\mu}^D = \begin{pmatrix} \mu_{C1} & \mu_{C2} & \mu_{MP} \end{pmatrix}^T \tag{16.146}$$

$$\dot{\boldsymbol{X}}^I = \begin{pmatrix} \dot{x}_{E1} & \dot{x}_{E2} & \dot{x}_M & \dot{x}_{ATP} & \dot{x}_{ADP} & \dot{x}_P \end{pmatrix}^T \tag{16.147}$$

$$\boldsymbol{V}^D = \begin{pmatrix} v_1 & v_2 & v_3 \end{pmatrix}^T \tag{16.148}$$

$$\boldsymbol{A}^I = A_4 \tag{16.149}$$

and the partitions of U_d are

$$\dot{\boldsymbol{X}}^D = \begin{pmatrix} \dot{x}_{C1} & \dot{x}_{C2} & \dot{x}_{MP} \end{pmatrix}^T \tag{16.150}$$

$$\boldsymbol{\mu}^I = \begin{pmatrix} \mu_{E1} & \mu_{E2} & \mu_M & \mu_{ATP} & \mu_{ADP} & \mu_P \end{pmatrix}^T \tag{16.151}$$

$$\boldsymbol{A}^D = \begin{pmatrix} A_1 & A_2 & A_3 \end{pmatrix}^T \tag{16.152}$$

$$\boldsymbol{V}^I = v_4 \tag{16.153}$$

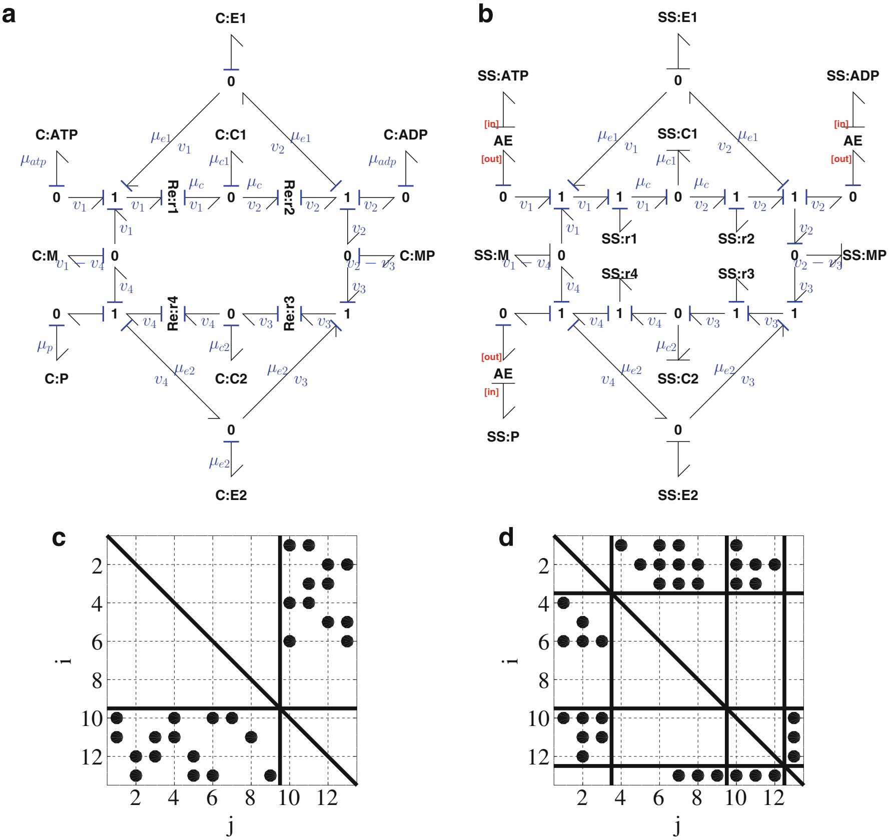


Fig. 16.12 Phosphorylation/dephosphorylation. This example is taken from Beard and Qian [5] and shows how driven by *ATP* hydrolysis, *M* is phosphorylated to *MP*, and *MP* dephosphorylated to *M*. (**a**) System bond graph. (**b**) Junction structure with maximum derivative causality. (**c**) and (**d**) Alternative representation of the structure matrices S (16.43) and S_d (16.48) where *filled circle* indicates a non-zero entry and *i* and *j* the row and column indices respectively

The 13×13 matrix S_d is summarised in Fig. 16.12d. The relevance of the submatrices S_{xx}, S_{AA}, $S_{A\mu}$ and S_{vv} are now examined in turn.

1. From the second row partition

$$S_{xx} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ -1 & -1 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{16.154}$$

the first and second rows of S_{xx} correspond to the conserved moieties $x_{e1} + x_{c1} = x_{\text{tot1}}$ and $x_{e2}+x_{c2} = x_{\text{tot2}}$. The third row of S_{xx} corresponds to the conserved moiety $x_{mp} + x_{c1} + x_{c2} + x_m = x_{\text{tot}}$. The fourth, fifth and sixth rows of S_{xx} correspond to the three constant chemostat states: x_{ATP}, x_{ADP} and x_P.

$$\dot{x}_{ATP} = \dot{x}_{ADP} = \dot{x}_P = 0 \tag{16.155}$$

2. From the fourth row partition

$$S_{AA} = \begin{pmatrix}-1 & -1 & -1\end{pmatrix} \qquad \text{and} \qquad S_{A\mu} = \begin{pmatrix}0 & 0 & 0 & 1 & -1 & -1\end{pmatrix} \tag{16.156}$$

the first row of S_{AA}, together with the first row of $S_{A\mu}$, corresponds to the sum of the affinities around the central loop being equal to the weighted sum of the chemostat potentials

$$A_1 + A_2 + A_3 + A_4 = \mu_{ATP} - \mu_{ADP} - \mu_P \tag{16.157}$$

Thus the flow around the loop comprising $r_1 \dots r_4$ is driven by the $ATP \rightleftharpoons ADP + P$ reaction.

3. From the third row partition

$$S_{vv} = \begin{pmatrix}1\\1\\1\end{pmatrix} \tag{16.158}$$

In the steady state ($\dot{X} = 0$). In this case S_{vv} corresponds to the four reaction flows being equal

$$v_1 = v_2 = v_3 = v_4 \tag{16.159}$$

16.7 Conclusion

The causal properties of the bond graph junctions structures representing biomolecular systems have been examined and shown to provide an alternative approach to generating the related stoichiometric matrices. As a graphical approach, the bond graph method gives more intuitive insight than purely numerical approaches.

The bond graph approach uses causality arguments and the sequential causality assignment procedure (SCAP); the numerical approach uses Gaussian elimination. It follows that the bond graph approach must, in some sense, be equivalent to the Gaussian elimination approach; this deserves further investigation.

As discussed in this chapter, it can be more convenient to represent reactions by a one-port **R** component and associated **1** junction rather than a two-port **Re** component. This raises the question as to whether it is always possible to model in this way. The key issue here is under what circumstances forward and reverse stoichiometric matrices N^f and N^r can be deduced from the stoichiometric matrix $N = N^r - N^f$.

This chapter focuses on turning closed biomolecular systems into open systems using the *chemostat* (constant chemical potential) concept previously used by Polettini and Esposito [35] and Gawthrop and Crampin [12]. It would be interesting to examine the use of the dual concept of *flowstats* (constant molar flow) [12] in the context of this chapter.

Gawthrop et al. [13] use *hierarchical* bond graph models to allow large models to be built out of submodels and use the notion of a *port* component to do this. It would be interesting to develop a hierarchical version of the methods presented in this chapter.

Acknowledgements Peter Gawthrop would like to thank the Melbourne School of Engineering for its support via a Professorial Fellowship. He would also like to thank Michael Pan and Joe Cursons for their close reading of the draft chapter.

References

1. Alberts, B., Johnson, A., Lewis, J., Morgan, D., Raff, M., Roberts, K., et al. (Eds.). (2015). *Molecular biology of the cell* (6th ed.). Abingdon: Garland Science.
2. Alon, U. (2007). *Introduction to systems biology: Design principles of biological networks*. Boca Raton: CRC Press.
3. Atkins, P., & de Paula, J. (2011). *Physical chemistry for the life sciences* (2nd ed.). Oxford: Oxford University Press.
4. Beard, D. A. (2012). *Biosimulation: Simulation of living systems*. Cambridge: Cambridge University Press. ISBN: 978-0-521-76823-8.
5. Beard, D. A., & Qian, H. (2010). *Chemical biophysics: Quantitative analysis of cellular systems*. Cambridge: Cambridge University Press.
6. Breedveld, P. C. (1982). Thermodynamic bond graphs and the problem of thermal inertance. *Journal of the Franklin Institute, 314*(1), 15–40. ISSN: 0016-0032. doi:10.1016/0016-0032(82)90050-3.
7. Cellier, F. E. (1991). *Continuous system modelling*. New York: Springer.
8. Cloutier, M., Bolger, F. B., Lowry J. P., & Wellstead P. (2009). An integrative dynamic model of brain energy metabolism using in vivo neurochemical measurements. *Journal of Computational Neuroscience, 27*(3), 391–414. ISSN: 0929-5313. doi:10.1007/s10827-009-0152-8.
9. Fuchs, H. U. (1996). *The dynamics of heat*. New York: Springer.
10. Gawthrop, P. J., & Bevan, G. P. (2007). Bond-graph modeling: A tutorial introduction for control engineers. *IEEE Control Systems Magazine, 27*(2), 24–45. doi:10.1109/MCS.2007.338279.
11. Gawthrop, P. J., & Crampin, E. J. (2014). Energy-based analysis of biochemical cycles using bond graphs. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science, 470*(2171), 1–25. doi:10.1098/rspa.2014.0459. Available at arXiv:1406.2447.
12. Gawthrop, P. J., & Crampin, E. J. (2016). Modular bond-graph modelling and analysis of biomolecular systems. *IET Systems Biology, 10*, 2016. ISSN: 1751-8849. doi:10.1049/iet-syb.2015.0083. Available at arXiv:1511.06482.

13. Gawthrop, P. J., Cursons, J., & Crampin, E. J. (2015). Hierarchical bond graph modelling of biochemical networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 471*(2184), 1–23. ISSN: 1364-5021. doi:10.1098/rspa.2015.0642. Available at arXiv:1503.01814.
14. Gawthrop, P. J., & Smith, L. (1992). Causal augmentation of bond graphs with algebraic loops. *Journal of the Franklin Institute, 329*(2), 291–303. doi:10.1016/0016-0032(92)90035-F.
15. Gawthrop, P. J., & Smith, L. P. S. (1996). *Metamodelling: Bond graphs and dynamic systems*. Hemel Hempstead: Prentice Hall. ISBN: 0-13-489824-9.
16. Greifeneder, J., & Cellier, F. E. (2012). Modeling chemical reactions using bond graphs. In *Proceedings ICBGM12, 10th SCS International Conference on Bond Graph Modeling and Simulation, Genoa, Italy* (pp. 110–121).
17. Hill, T. L. (1989). *Free energy transduction and biochemical cycle kinetics*. New York: Springer.
18. Jamshidi, N., & Palsson, B. (2010). Mass action stoichiometric simulation models: Incorporating kinetics and regulation into stoichiometric models. *Biophysical Journal, 98*(2), 175–185. ISSN: 0006-3495. doi:10.1016/j.bpj.2009.09.064.
19. Job, G., & Herrmann, F. (2006). Chemical potential – A quantity in search of recognition. *European Journal of Physics, 27*(2), 353–371 (2006). doi:10.1088/0143-0807/27/2/018.
20. Karnopp, D. (1990). Bond graph models for electrochemical energy storage: Electrical, chemical and thermal effects. *Journal of the Franklin Institute, 327*(6), 983–992. ISSN: 0016-0032. doi:10.1016/0016-0032(90)90073-R.
21. Karnopp, D. C., Margolis, D. L., & Rosenberg, R. C. (2012). *System dynamics: Modeling, simulation, and control of mechatronic systems* (5th ed.). New York: Wiley. ISBN: 978-0470889084.
22. Klipp, E., Liebermeister, W., Wierling, C., Kowald, A., Lehrach, H., & Herwig, R. (2011). *Systems biology*. Weinheim: Wiley.
23. Lambeth, M. J., & Kushmerick, M. J. (2002). A computational model for glycogenolysis in skeletal muscle. *Annals of Biomedical Engineering, 30*(6), 808–827. ISSN: 0090-6964. doi:10.1114/1.1492813.
24. Maxwell, J. C. (1871). Remarks on the mathematical classification of physical quantities. *Proceedings London Mathematical Society, 3*, 224–233.
25. Mukherjee, A., Karmaker, R., & Samantaray, A. K. (2006). *Bond graph in modeling, simulation and fault identification*. New Delhi: I.K. International.
26. Ort, J. R., & Martens, H. R. (1973). The properties of bond graph junction structure matrices. *Journal of Dynamic Systems, Measurement, and Control, 95*, 362–367. ISSN: 0022-0434. doi:10.1115/1.3426736.
27. Oster, G., & Perelson, A. (1974). Chemical reaction networks. *IEEE Transactions on Circuits and Systems, 21*(6), 709–721. ISSN: 0098-4094. doi:10.1109/TCS.1974.1083946.
28. Oster, G., Perelson, A., & Katchalsky, A. (1971). Network thermodynamics. *Nature, 234*, 393–399. doi:10.1038/234393a0.
29. Oster, G. F., Perelson, A. S., & Katchalsky, A. (1973). Network thermodynamics: Dynamic modelling of biophysical systems. *Quarterly Reviews of Biophysics, 6*(01), 1–134 doi:10.1017/S0033583500000081.
30. Palsson, B. (2006). *Systems biology: Properties of reconstructed networks*. Cambridge: Cambridge University Press. ISBN: 0521859034.
31. Palsson, B. (2011). *Systems biology: Simulation of dynamic network states*. Cambridge: Cambridge University Press.
32. Paynter, H. M. (1961). *Analysis and design of engineering systems*. Cambridge: MIT Press.
33. Paynter, H. M. (1993). Preface. In J. J. Granda & F. E. Cellier (Eds.), *Proceedings of the International Conference On Bond Graph Modeling (ICBGM'93)*. Simulation Series, La Jolla, CA, USA, January 1993 (Vol. 25). Society for Computer Simulation. ISBN: 1-56555-019-6.
34. Perelson, A. S. (1975). Bond graph junction structures. *Journal of Dynamic Systems, Measurement, and Control, 97*, 189–195. ISSN: 0022-0434. doi:10.1115/1.3426901.

35. Polettini, M., & Esposito, M. (2014). Irreversible thermodynamics of open chemical networks. I. Emergent cycles and broken conservation laws. *The Journal of Chemical Physics, 141*(2), 024117 doi:10.1063/1.4886396.
36. Qian, H., & Beard, D. A. (2005). Thermodynamics of stoichiometric biochemical networks in living systems far from equilibrium. *Biophysical Chemistry, 114*(2–3), 213–220. ISSN: 0301-4622. doi:10.1016/j.bpc.2004.12.001.
37. Rosenberg, R. C., & Andry, A. N. (1979). Solvability of bond graph junction structures with loops. *IEEE Transactions on Circuits and Systems, 26*(2), 130–137. ISSN: 0098-4094. doi:10.1109/TCS.1979.1084615.
38. Sueur, C., & Dauphin-Tanguy, G. (1989). Structural controllability/observability of linear systems represented by bond graphs. *Journal of the Franklin Institute, 326*, 869–883.
39. Sueur, C., & Dauphin-Tanguy, G. (1991). Bond-graph approach for structural analysis of MIMO linear systems. *Journal of the Franklin Institute, 328*, 55–70.
40. Sueur, C., & Dauphin-Tanguy, G. (1997). Controllability indices for structured systems. *Linear Algebra and its Applications, 250*, 275–287.
41. Thoma, J. U., & Atlan, H. (1977). Network thermodynamics with entropy stripping. *Journal of the Franklin Institute, 303*(4), 319–328. ISSN: 0016-0032. doi:10.1016/0016-0032(77)90114-4.
42. Thoma, J. U., & Mocellin, G. (2006). *Simulation with entropy thermodynamics: Understanding matter and systems with bondgraphs*. Heidelberg: Springer. ISBN: 978-3-540-32798-1.
43. Van Rysselberghe, P. (1958). Reaction rates and affinities. *The Journal of Chemical Physics, 29*(3), 640–642. doi:10.1063/1.1744552.
44. Wellstead, P. E. (1979). *Introduction to physical system modelling*. Academic Press: New York.

Part V
Software

There are a number of commercial integrated multi-formalism modelling and simulation software environments available that support bond graph modelling, e.g. 20sim®, Symbols®. Another integrated software environment of which a basic version is freely available is BondSim. A graphical bond graph user interface to the sophisticated open source modelling and simulation software OpenModelica is provided by the freely available BondLib library developed by Cellier.

As bond graphs can be systematically converted into block diagrams also programs such as Matlab®/Simulink® or the open source software Scilab/Xcos can be used for numerical computation of bond graph models. Furthermore, bond graph preprocessors such as CAMP-G can derive state space equations automatically from a causal bond graph so that Matlab®/Simulink® can be used to generate transfer functions in symbolic form or to perform a simulation.

For medium scale bond graphs, equations in the form of a DAE system can be manually derived in a systematic manner and can be directly formulated either in the script languages of Matlab® or Scilab or in the modelling language Modelica®. Scilab/Xcos even has a Modelica compiler that supports a subset of the language so that a submodel formulated in Modelica can be used as a signal block in the block diagram interface Xcos.

On computers with a single (multi-core) CPU, the motion of a virtual complex multibody system (MBS) in 3D space is often visualised for performance reasons after the simulation loop has finished. (For rather small class room examples, simulation results can be visualised while the simulation is running.) Data provided by the simulation engine can then be used by a postprocessing software module to replay the 3D-motion in real-time or at another speed that may be chosen for replay. On computers with a multi-core CPU and a graphical co-processor (GPU), one core may compute the dynamic behaviour while a program for the design of geometric and kinematic models of multibody systems running on another core can simultaneously visualise the 3D motion of a MBS by using the GPU. Some little delay is caused by the concurrent transfer of results from main memory to the graphical board.

The last chapter in this book presents an approach that uses the BondSim software environment for the development of a dynamic bond graph model and the simulation of the dynamic behaviour of a multibody system. Updated values of displacements of joints, i.e. revolute joint angles and prismatic joint displacements of the MBS are repeatedly sent via interprocess communication to a program BondSimVisual for the design of geometric and kinematic models of multibody systems that visualises the motion of the MBS by redrawing the virtual 3D scene. BondSim and BondSimVisual have been developed by Professor Damić. Only rather small amounts of data are to be exchanged between the two programs so that the overhead due to the interprocess communication is small. As a result, simulation of the dynamic behaviour of an MBS can be performed on one computer while the 3D motion of an MBS can be simultaneously visualised on another remote computer (with a GPU).

Chapter 17
Multibody System Modeling, Simulation, and 3D Visualization

Vjekoslav Damić and Majda Čohodar

Abbreviations

BDF	Backward differentiation formula
DAE	Differential-algebraic equations
EJS	Euler junction structure
MBS	Multibody system
MFC	Microsoft foundation class
RGB	Red–green–blue
TCP	Tool centre point
VTK	Visualization Toolkit

17.1 Introduction

There is an extremely large body of literature dealing with the modeling and simulation of MBS. They mostly deal with kinematics and dynamics of 3D mechanical objects and systems. To enhance understanding of real behavior of such systems more attention was directed to the visualization of their motion in the virtual 3D scenes. These are often two separate approaches. Many well-known modeling and simulation software systems, which are quite capable of solving the dynamical side of the problem, support also the visualization. On other side 3D visualization

V. Damić (✉)
University of Dubrovnik, Dubrovnik, Croatia
e-mail: vdamic@unidu.hr

M. Čohodar
University of Sarajevo, Sarajevo, Bosnia and Herzegovina
e-mail: cohodar@mef.unsa.ba

W. Borutzky (ed.), *Bond Graphs for Modelling, Control and Fault Diagnosis of Engineering Systems*, DOI 10.1007/978-3-319-47434-2_17

tools often support also MBS kinematics. In this chapter the modeling and analysis of MBS systems from two different points of view are analyzed—dynamic and geometric (visual).

We start with dynamical modeling of MBS based on acausal component models based technique described in more detail in the recent book [3]. It is shown how a model of a complex MBS can be systematically developed. It leads essentially to implicit Euler–Lagrange models. It is well known that such models are higher index DAEs models. However, using bond graphs the corresponding models are velocity based and thus of a lower index and could be readily solved using available technique [3]. Such models can be systematically developed using BondSim© modeling and simulation environment.

The next topic is geometrical modeling of complex MBS such as robots, or other mechanisms, moving in 3D space. The approach used is based on visualization pipeline techniques implemented in VTK C++ library of Will Schroeder et al. [12]. The models can be constructed by assembling simple primitive objects such as cubes, cylinders, and spheres, or the CAD system created geometric objects. The visualization model of the system can be described by a simple script. This script is read by a special designed BondSimVisual© application. It constructs in the computer memory the corresponding visualization pipeline objects representing the system in the question. These basically show the corresponding mechanisms as the static objects in a 3D virtual scene.

In order to interconnect these two models—dynamical bond graph models and geometrical objects in 3D virtual scene an inter-process communication technique is developed. The technique used is based on *named pipe* technology. It enables that the dynamical model during the simulation drive the corresponding visual model in 3D virtual scene, thus gives to a user 3D view on the problem. In this way the dynamical and visual model behaves as a complex multi-physics model of the problem (see, e.g., Damic and Cohodar [2]).

17.2 Motion of Constrained Rigid Bodies in Space

17.2.1 Basic Kinematics

To describe motion of a body in space we use two fundamental coordinate frames (Fig. 17.1)—a base frame $Oxyz$ and a body frame $Cx'y'z$. There can be a number of bodies and, hence, a number of body frames that are used for their description. On the other hand, there is a single base (inertial) frame. In robotics it is often convenient to introduce other frames, as well [13]. It is assumed that all frames are *3D Cartesian coordinate* frames.

We assume that the *position* of a body frame with respect to the base is defined by a position vector $\mathbf{r}_C$ of the origin C of the body frame. The point C is the reference

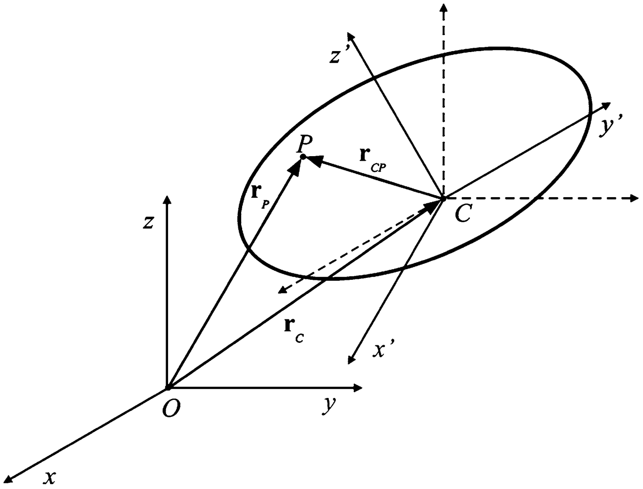


Fig. 17.1 The base and body frame

point for describing the body position—a body center (pole). It is usually the body mass center, but could also be some other point. The body *orientation* is defined by a rotation matrix **R** composed of the direction cosines of the body axes with respect to the base axes. The position of any point *P* fixed in the body with respect to the base is given by

$$\mathbf{r}_P = \mathbf{r}_C + \mathbf{R}\mathbf{r}'_{CP} \tag{17.1}$$

where $\mathbf{r}_{CP}'$ is the vector of its *material* coordinates, i.e., the coordinates with respect to the body frame.

This relation describes the position of rigid body in 3D space. If we assume that initially the both coordinate frames are coincident, then (17.1) show that a body can be moved into an arbitrary position in the space by rotation of the body frame about the common origin defined by rotation matrix **R** and translation of the body frame with respect to the base for vector $\mathbf{r}_C$. Because the addition in (17.1) is a commutative operation it is also possible to exchange the rotation and translation transformations, i.e., first apply the translations and then the rotation about the translated frame (Fig. 17.1).

The relationship (17.1) can be described also in terms of the homogenous transformation. Thus, we introduce the homogenous four-dimensional vector corresponding to the common 3D vector $\mathbf{r} = \begin{pmatrix} x\ y\ z \end{pmatrix}^T$ by a

$$\tilde{\mathbf{r}} = \begin{pmatrix} \mathbf{r} \\ \overline{1} \end{pmatrix} \tag{17.2}$$

The fourth component is a scale factor assumed here to be 1. We may introduce a matrix of homogenous transform defined by

$$\mathbf{T} = \left(\begin{array}{c|c} \mathbf{R} & \mathbf{r}_C \\ \hline \mathbf{0}^T & 1 \end{array}\right) \tag{17.3}$$

Now (17.1) can be written compactly as

$$\tilde{\mathbf{r}}_P = \mathbf{T}\tilde{\mathbf{r}}_{CP} \tag{17.4}$$

The homogenous transformations of position vectors are often used when dealing with geometric representation of body motion in 3D space (visualization). In dynamics we are more interested in the velocities.

During the motion, the positions of the body center $\mathbf{r}_C$ and its orientation—represented by matrix $\mathbf{R}$—change with time. The material coordinates of a point fixed in the body do not change. The velocity of a point P, as seen from the base frame, can be found by differentiating (17.1) with respect to time, i.e.,

$$\mathbf{v}_P = \mathbf{v}_C + \frac{d\mathbf{R}}{dt}\mathbf{r}'_{CP} \tag{17.5}$$

The relative position of a point P with respect to the base frame can be found by multiplying its body frame coordinates by the rotation matrix $\mathbf{R}$,

$$\mathbf{r}_{CP} = \mathbf{R}\mathbf{r}'_{CP} \tag{17.6}$$

Note that the rotation matrix is an orthogonal matrix, i.e., $\mathbf{R}^T\mathbf{R} = \mathbf{R}\mathbf{R}^T = \mathbf{I}$, where subscript T denotes the transposition operator and $\mathbf{I}$ is the identity matrix. Substituting (17.6) into (17.5) yields

$$\mathbf{v}_P = \mathbf{v}_C + \frac{d\mathbf{R}}{dt}\mathbf{R}^T\mathbf{r}_{CP} \tag{17.7}$$

In analogy with vector product notation we may introduce the symbol (Borri et al. [1])

$$\boldsymbol{\omega}\times = \frac{d\mathbf{R}}{dt}\mathbf{R}^T \tag{17.8}$$

Thus, (17.7) can be written as

$$\mathbf{v}_P = \mathbf{v}_C + \boldsymbol{\omega} \times \mathbf{r}_{CP} \tag{17.9}$$

This reminds us of common vector product operation between the vectors. The $\boldsymbol{\omega}$ is really a vector, as can be easily shown.

The matrix in (17.8) is a skew-symmetric matrix (tensor). It has only three independent elements, and thus can be written in a familiar form

$$\frac{d\mathbf{R}}{dt}\mathbf{R}^T = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\,\omega_y & \omega_x & 0 \end{pmatrix} \tag{17.10}$$

We may introduce now a vector composed of this components. This is the so-called *axial vector* corresponding to a skew-symmetric matrix

$$\boldsymbol{\omega} = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = axial\left(\frac{d\mathbf{R}}{dt}\mathbf{R}^T\right) \tag{17.11}$$

In the case of rotation matrix $\mathbf{R}$, the axial vector $\boldsymbol{\omega}$ is familiar the body angular velocity.

It could also be shown that the following well-known formula holds:

$$\boldsymbol{\omega}\times\mathbf{r} = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\,\omega_y & \omega_x & 0 \end{pmatrix}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -y\omega_z + z\omega_y \\ x\omega_z - z\omega_x \\ -\,x\omega_y + y\omega_x \end{pmatrix} = \begin{pmatrix} 0 & z & -y \\ -\,z & 0 & x \\ y & -x & 0 \end{pmatrix}\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = -\mathbf{r}\times\boldsymbol{\omega}$$

The relations similar to (17.8) and (17.9) hold in the body frame as well. The components of the global velocity $\mathbf{v}_P$ expressed in the body frame are given by

$$\mathbf{v}'_P = \mathbf{R}^T\mathbf{v}_P \tag{17.12}$$

Multiplying (17.7) by $\mathbf{R}^T$ yields

$$\mathbf{v}'_P = \mathbf{v}'_C + \mathbf{R}^T\frac{d\mathbf{R}}{dt}\mathbf{r}'_{CP} \tag{17.13}$$

Now we have

$$\mathbf{R}^T\frac{d\mathbf{R}}{dt} = \boldsymbol{\omega}'\times \tag{17.14}$$

where $\boldsymbol{\omega}'$ is the vector of the rectangular components of body angular velocity in the body frame. Thus, (17.13) can be written as

$$\mathbf{v}'_P = \mathbf{v}'_C + \boldsymbol{\omega}'\times\mathbf{r}'_{CP} \tag{17.15}$$

To represent these relationships compactly we introduce a generalized vector of *6D* component composed of the linear velocity vector of the body center and the body angular velocity. In the base frame the generalized velocity is given by

$$\mathbf{f}_C = \begin{pmatrix} \mathbf{v}_C \\ \boldsymbol{\omega} \end{pmatrix} \tag{17.16}$$

and similarly in the body frame as

$$\mathbf{f}'_C = \begin{pmatrix} \mathbf{v}'_C \\ \boldsymbol{\omega}' \end{pmatrix} \tag{17.17}$$

These quantities are denoted by **f** because, as will be shown later, these are simply the flows of the bond graph component ports.

In a similar way, a generalized velocity of any other point P fixed in the body can be represented in the base frame by

$$\mathbf{f}_P = \begin{pmatrix} \mathbf{v}_P \\ \boldsymbol{\omega} \end{pmatrix} \tag{17.18}$$

and in the body frame as

$$\mathbf{f}'_P = \begin{pmatrix} \mathbf{v}'_P \\ \boldsymbol{\omega}' \end{pmatrix} \tag{17.19}$$

Note that the following holds:

$$\mathbf{R}\left(\boldsymbol{\omega}' \times \mathbf{r}'_{CP}\right) = \boldsymbol{\omega} \times \mathbf{r}_{CP} = -\mathbf{r}_{CP} \times \boldsymbol{\omega} = \left(-\mathbf{r}_{CP} \times \mathbf{R}\right)\boldsymbol{\omega}'$$

Using (17.15) the corresponding transformation of flows between the body and base frames can be written compactly as

$$\mathbf{f}_P = \begin{pmatrix} \mathbf{R} & -\mathbf{r}_{CP} \times \mathbf{R} \\ \mathbf{0} & \mathbf{R} \end{pmatrix} \mathbf{f}'_C \tag{17.20}$$

The matrix

$$\mathbf{C} = \begin{pmatrix} \mathbf{R} & -\mathbf{r}_{CP} \times \mathbf{R} \\ \mathbf{0} & \mathbf{R} \end{pmatrix} \tag{17.21}$$

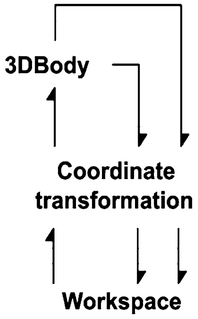
can be termed a representation of the body frame *configuration tensor* [13]. Hence, (17.20) can be written as

$$\mathbf{f}_P = \mathbf{C}\mathbf{f}'_C \tag{17.22}$$

The similar relationships hold in the body frame

$$\mathbf{f}'_P = \mathbf{C}'\mathbf{f}_C \tag{17.23}$$

Fig. 17.2 Representation of a body moving in space



where

$$\mathbf{C}' = \begin{pmatrix} \mathbf{R}^T & -\mathbf{r}'_{CP} \times \mathbf{R}^T \\ \mathbf{0} & \mathbf{R}^T \end{pmatrix} \tag{17.24}$$

The last equations imply also the flow relationships in the body frame:

$$\mathbf{f}'_P = \begin{pmatrix} \mathbf{I} & -\mathbf{r}'_{CP}\times \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \mathbf{f}'_C \tag{17.25}$$

17.2.2 Bond Graph Representation of a Body Moving in Space

To develop a fundamental bond graph representation of the bodies moving in space we consider a single body interacting with its environment. The environment consists of the bodies the analyzed body is connected to. We model the body and its environment by two components—3DBody and Workspace (Fig. 17.2).

We assume that the body is connected to the other bodies at two points. We also assume that a body in the environment acts on the analyzed body by a force and moment. Likewise *the body* acts also to another body in the environment. In this way there is a transfer of power from the environment to the body and from the body back to the environment. In order to represent these interactions we assume that the 3DBody component has *three* ports. Two of these, the bottom and the upper ones in Fig. 17.2, correspond to the power interactions just described. We added the third one (the side port) to correspond to the body center of mass. It is possible to develop a model without the explicit use of the mass center port, but it simplifies the description of the body dynamics.

As explained previously, velocities seen at a port can be represented by the 6D *flow* vectors of (17.18) and (17.19). Similarly we can use 6D *efforts* to represent the resultant force and moment at a port. Such efforts can be represented in the base frame by

$$\mathbf{e}_P = \begin{pmatrix} \mathbf{F}_P \\ \mathbf{M}_P \end{pmatrix} \tag{17.26}$$

and, similarly, in the body frame as

$$\mathbf{e}'_P = \begin{pmatrix} \mathbf{F}'_P \\ \mathbf{M}'_P \end{pmatrix} \tag{17.27}$$

Here $\mathbf{F}_P$ and $\mathbf{F}'_P$ are 3D representations of the resultant force at connection point P and $\mathbf{M}_P$ and $\mathbf{M}'_P$ of the resultant moment.

To represent these 6D vector quantities, the ports of the 3DBody and Workspace components are assumed to consist of two subports. The first one corresponds to the linear velocity and resultant force, and the other to the body angular velocity and resultant moment. This way the ports can be used to access efforts and flows at the port by two ordered 6D efforts and flows. At the mass center port, however, as will be seen later, a 3D representation is sufficient. The flow is the mass center velocity and the effort is the resultant of the forces acting on the body at other ports (interconnection points) and reduced to the mass center. The gravity force (the body weight) acting there is not accounted for and is taken into account when dealing with the body translation dynamics. The reason for this is that body weight can be easily described in the base frame. However, this is not so in a body frame, because it is generally rotated with respect to the first.

We represent effort and flow vectors at body ports as seen in the body, i.e., with respect to a body fixed frame. Similarly at the workspace ports the corresponding quantities are expressed with respect to the base frame. Transformations between these two representations are given in Fig. 17.2 by the coordinate transformation component. This component transforms effort and flow at a body port to the corresponding base frame representation and vice versa. The transformation between efforts is given by

$$\mathbf{e}_P = \begin{pmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{pmatrix} \mathbf{e}'_P \tag{17.28}$$

and similarly for the flows,

$$\mathbf{f}_P = \begin{pmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{pmatrix} \mathbf{f}'_P \tag{17.29}$$

where $\mathbf{R}$ is the rotation matrix of the body frame with respect to the base.

We use Cartesian 3D frames only and, thus, the transformation matrix is orthogonal. It can be easily shown that the following relationship holds:

$$\mathbf{f}'^T_P \mathbf{e}'_P = \mathbf{f}^T_P \mathbf{e}_P \tag{17.30}$$

and thus the transformation does not change the power transfer.

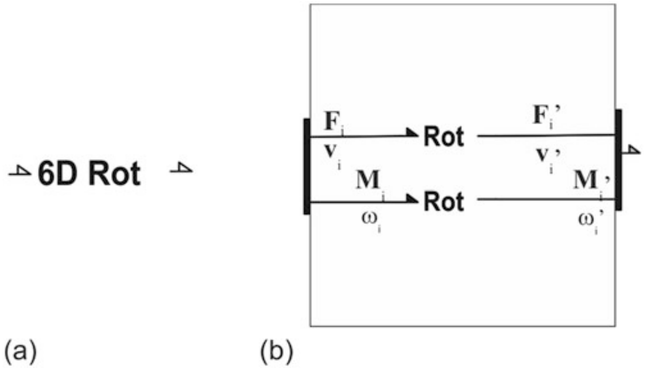


Fig. 17.3 6D transformation: (**a**) The component representation, (**b**) The structure

The transformation of the port quantities can be represented by the 6D transform component of Fig. 17.3a. This component transforms the effort-flow pairs of the corresponding ports, according to (17.28) and (17.29), as shown in Fig. 17.3b. The transformations are represented by the Rot components that describe the transformation of 3D effort and flow vectors by the rotation matrix **R**. The transformation can be represented using transformer elements TF.

The velocities at a port are related to the velocity of the body mass center by (17.23) and (17.24). To find the relation between forces and moments, we evaluate the power transfer between the ports. We obtain the following relation:

$$\mathbf{v}'^{T}_{P}\mathbf{F}'_{P} + \boldsymbol{\omega}'^{T}\mathbf{M}'_{P} = \mathbf{v}'^{T}_{C}\mathbf{F}'_{P} + \boldsymbol{\omega}'^{T}\left(\mathbf{r}'_{CP} \times \mathbf{F}'_{P} + \mathbf{M}'_{P}\right) \tag{17.31}$$

The last equation, jointly with (17.25), describes the basic velocity and force relationships for the rigid bodies. The structure of the 3D Body component following these relationships is given in Fig. 17.4.

Component 1 is an array of the effort junctions corresponding to the center of mass velocity. The component Rotation defines the body rotational dynamic discussed later. The components 0 represent the array of flow junctions that describe the port velocities relationship given by the first 3D row of (17.25). It gives the linear velocity at a port as the sum of the velocity of the body mass center and the relative velocity due to the rotation of the body about the mass center. The joint variable is the force at the port. The body angular velocity, however, is a property of the body and it is directly transferred to the Rotation component.

The LinRot components represent transformations between linear and angular quantities. These transformations are defined by the skew-symmetric tensor operations in (17.25) and (17.31) and can be represented by transformers, as in Fig. 17.5.

Every transformer in Fig. 17.5 corresponds to a nonzero entry of tensor $-\mathbf{r}_{CP}'\times$ or its transpose. The flow junctions on the left evaluate the relative velocities at the left port and the effort junctions on the right give the moment at the right

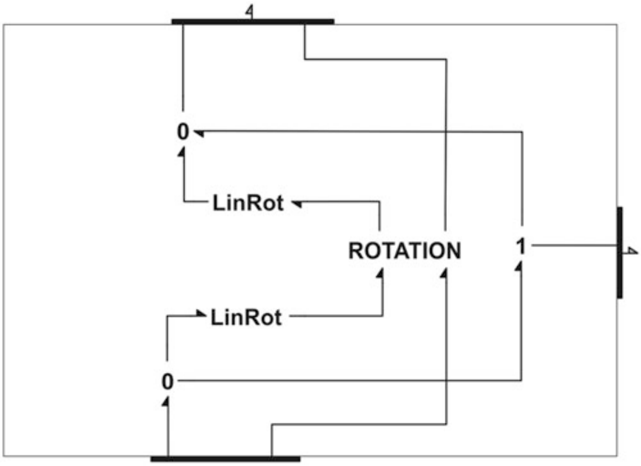


Fig. 17.4 The model of 3D body motion

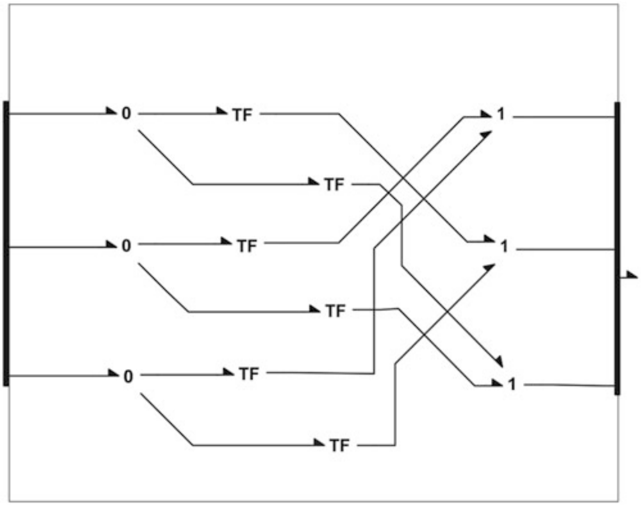


Fig. 17.5 The representation of LinRot transformation

port. The ratios of the transformers are the material coordinates of a body point corresponding to the port with respect to the mass center. Thus, they are parameters that depend on the geometry of the rigid body and don't change with its motion.

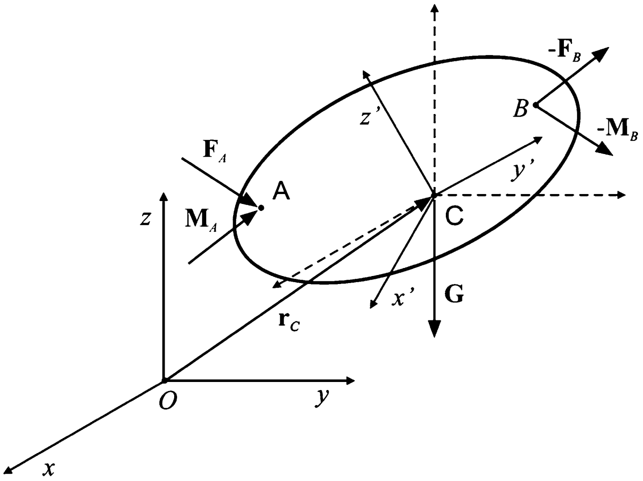


Fig. 17.6 Motion of the body in space

17.2.3 Dynamic of Rigid Bodies

To complete the model, we need a dynamic equation governing rigid body motion. The simplest form of such an equation is given with respect to axes translating with the body mass center (Fig. 17.6). We assume that the base frame is an inertial.

The translational part of the motion is described by

$$\left.\begin{aligned}\mathbf{p} &= m\mathbf{I}\mathbf{v}_C\\ \tfrac{d\mathbf{p}}{dt} &= \mathbf{F}\end{aligned}\right\} \tag{17.32}$$

Here, m is the body mass and $\mathbf{F}$ is the resultant of the forces reduced to the mass center. We represent the dynamics of body translation in the Workspace (Fig. 17.7a) by CM component that describes the motion of the body mass center. This last component (Fig. 17.7b) consists of effort junctions corresponding to the x, y, and z components of the velocity of the body mass center with respect to the base frame. These junctions are connected to the Workspace port to which the resultant of the forces acting on the body is transferred.

Note that the weight of the body is added here represented by the source efforts defining the weight components with respect to the base frame axes. The momentum law in (17.32) is represented by the inertial elements I with the body mass as a parameter. Because it is used both in the SE and I components, it can be defined just once, at the CM document level.

The rotational part of the body dynamics is commonly described in the frame translating with the mass center of the body and is parallel to the base frame. In this frame it reads

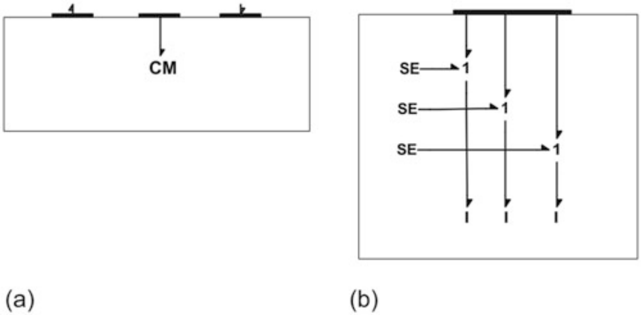


Fig. 17.7 Rigid body translation: (**a**) Representation in the Environment (**b**) The CM component

$$\left.\begin{array}{l}\mathbf{H} = \mathbf{J}\boldsymbol{\omega} \\ \frac{d\mathbf{H}}{dt} = \mathbf{M}\end{array}\right\} \tag{17.33}$$

where **J** is the mass inertia matrix and **M** is the resultant moment about the body mass center. Due to the body rotation the inertia matrix changes during the motion. Because of this the equations of the rotational dynamics are as a rule (at least for rigid bodies) represented with respect to a body fixed frame. It is not difficult to transform (17.33) to this form.

In the body frame the body moment of momentum is given by

$$\mathbf{H} = \mathbf{R}\mathbf{H}' \tag{17.34}$$

Substituting from (17.33) we obtain

$$\mathbf{H}' = \mathbf{R}^T\mathbf{H} = \mathbf{R}^T\mathbf{J}\boldsymbol{\omega} = \mathbf{R}^T\mathbf{J}\mathbf{R}\boldsymbol{\omega}' = \mathbf{J}'\boldsymbol{\omega}' \tag{17.35}$$

where

$$\mathbf{J}' = \mathbf{R}^T\mathbf{J}\mathbf{R} \tag{17.36}$$

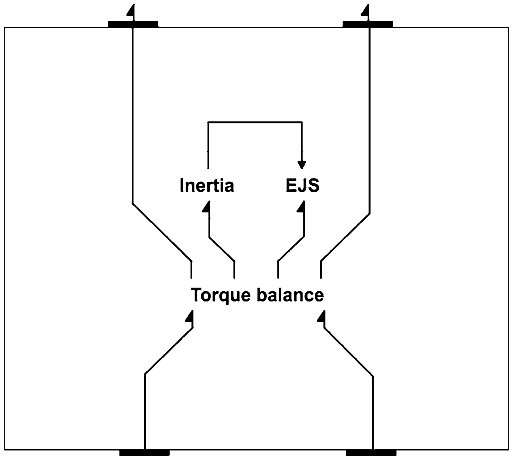
is the body mass inertia matrix with respect to the central body axes. Multiplying second (17.33) by $\mathbf{R}^T$ and substituting from (17.34), we obtain

$$\mathbf{R}^T\frac{d}{dt}\left(\mathbf{R}\mathbf{H}'\right) = \mathbf{M}' \tag{17.37}$$

or

$$\frac{d\mathbf{H}'}{dt} + \mathbf{R}^T\frac{d\mathbf{R}}{dt}\mathbf{H}' = \mathbf{M}' \tag{17.38}$$

Fig. 17.8 ROTATION component



Applying (17.14) the last equations can be written as

$$\frac{d\mathbf{H}'}{dt} + \boldsymbol{\omega}' \times \mathbf{H}' = \mathbf{M}' \tag{17.39}$$

These are the famous Euler equations of body rotation in which the rate of change of the moment of momentum is represented by its local change and a part generated by the body rotation. This other part has a very elegant representation in a bond graph setting. This is the celebrated *Euler Junction Structure* (*EJS*) [6].

Now we can describe the ROTATION component of Fig. 17.4, which represents the body rotation dynamic about the center of mass with respect to the body frame (Fig. 17.8). The Torque balance component in the middle consists simply of an array of three effort junctions that corresponds to the x, y, and z components of the body angular velocity with respect to the body frame. The left and right ports transfer the moments of the forces and the moments acting on the body. In the center it is connected to Inertia and EJS components. The first of these consists of an array of inertial elements that describes the local rate of change of the moment of momentum. There is a control-out port that serves for the transfer of information on the moment of momentum vector $\mathbf{H}'$ that the EJS component needs. This can be seen in (17.39). The EJS component is shown in Fig. 17.9.

It consists of three gyrators connected in a ring. Note that the power circulates inside the EJS, thus there is no net power generation or dissipation. The gyrators are modulated by the body moment of momentum.

This completes development of the component model of a body moving in 3D space. In BondSim© program the model is stored in the *Models Library*, under *Word Model Components*, from which it can be easily used by dragging it into the working window where the models are developed. Now we turn our attention to the modeling of interconnections between the bodies in space.

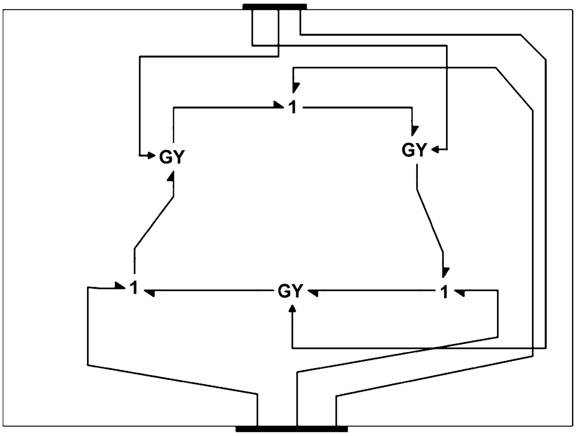


Fig. 17.9 Euler junction structure (EJS)

17.2.4 Modeling of Body Interconnection

Typical mechatronics systems, such as robots, consist of manipulators guided by controllers. The manipulators are multibody systems consisting of several members (links) interconnected by the suitable joints. They are powered by servo-actuators. In the previous sections we developed a fairly general component model of bodies that can be used for the representation of manipulator links. Now we develop models of the joints. Two types of joint are considered—revolute and prismatic. Based on these component models, a typical robotic manipulator can be represented in a way similar as how the real manipulators are assembled.

The approach is applicable to other multibody systems as well. Robot manipulators are chosen for several reasons. These are fascinating systems that have influenced development in many fields such as multibody mechanics, conventional and intelligent control, sensors and actuator technology, and have promoted mechatronics as design philosophy.

We consider here only the most common revolute joints. The analysis of the prismatic and the other joints can proceed in a similar way [3]. A typical revolute joint is shown in Fig. 17.10.

It consists of two bodies A and B, which can rotate about the common axis. The axis can have arbitrary position in space. The bodies could, for example, be two links of a robot manipulator joined by a revolute joint. We attach to body A the joint coordinate frame $O_A x_A y_A z_A$ with axis z directed along the rotation axis. We assume further that there is also frame $O_B x_B y_B z_B$ attached to the body B. The precise position of the frame is not prescribed and in a specific multibody system can be defined as is most convenient, e.g., using the Denavit–Hartenberg convention [2]. The frame $Oxyz$ is the base frame.

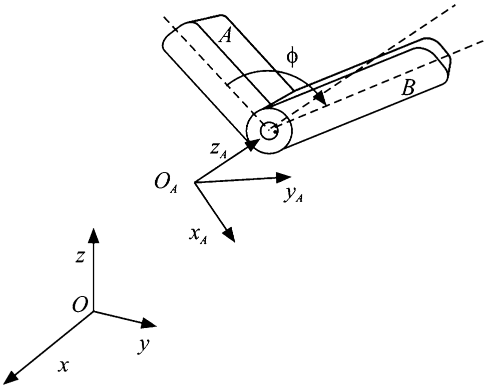


Fig. 17.10 Revolute joint in space

Let P be the center point of the revolute joint used as the reference connection point. The joint can be represented as a word model component with two ports corresponding to body A and B. We consider flows and efforts at these ports expressed in their respective body frames. At A (the power-in) these are

$$\mathbf{f}_P^A = \begin{pmatrix} \mathbf{v}_P^A \\ \boldsymbol{\omega}_A^A \end{pmatrix} \tag{17.40}$$

and

$$\mathbf{e}_P^A = \begin{pmatrix} \mathbf{F}_P^A \\ \mathbf{M}_P^A \end{pmatrix} \tag{17.41}$$

Similarly at B (the power-out) we have

$$\mathbf{f}_P^B = \begin{pmatrix} \mathbf{v}_P^B \\ \boldsymbol{\omega}_B^B \end{pmatrix} \tag{17.42}$$

and

$$\mathbf{e}_P^B = \begin{pmatrix} \mathbf{F}_P^B \\ \mathbf{M}_P^B \end{pmatrix} \tag{17.43}$$

The linear velocities of bodies A and B at the connection point are common to both of the bodies. Hence

$$\mathbf{v}_P^A = \mathbf{R}_B^A \mathbf{v}_P^B \tag{17.44}$$

where $\mathbf{R}_B^A$ is the rotation matrix of the body B frame with respect to the body A frame.

A similar relation holds for the forces. The linear velocities at the common point are equal, and the same is true for the power transferred across the joint during its translation, i.e.,

$$\left(\mathbf{v}_P^A\right)^T \mathbf{F}_P^A = \left(\mathbf{v}_P^B\right)^T \mathbf{F}_P^B \tag{17.45}$$

Substituting from (17.44) yields

$$\mathbf{F}_P^B = \mathbf{R}_A^B \mathbf{F}_P^A \tag{17.46}$$

Equations (17.44) and (17.46) describe the relationships between the linear parts of the flow-effort 3D vectors at the joint's ports. We develop next the relationships between the angular parts, i.e., the angular velocities and moments.

The rotation matrices of the bodies A and B frames with respect to the base frame are related by composition of the rotations, i.e.,

$$\mathbf{R}_B = \mathbf{R}_A \mathbf{R}_B^A \tag{17.47}$$

Differentiating it with respect to time we get

$$\frac{d\mathbf{R}_B}{dt} = \frac{d\mathbf{R}_A}{dt}\mathbf{R}_B^A + \mathbf{R}_A \frac{d\mathbf{R}_B^A}{dt} \tag{17.48}$$

From (17.8) and (17.14) we obtain

$$\boldsymbol{\omega}_B \times \mathbf{R}_B = \mathbf{R}_A \boldsymbol{\omega}_A^A \times \mathbf{R}_B^A + \mathbf{R}_A \boldsymbol{\omega}_{AB}^A \times \mathbf{R}_B^A \tag{17.49}$$

where

$$\boldsymbol{\omega}_{AB}^A = \text{axial}\left(\frac{d\mathbf{R}_B^A}{dt}\mathbf{R}_A^B\right) \tag{17.50}$$

is the relative angular velocity of body B with respect to body A expressed in body A frame. Simplifying (17.49) we get

$$(\mathbf{R}_A)^T \boldsymbol{\omega}_B \times \mathbf{R}_A = \boldsymbol{\omega}_A^A \times + \boldsymbol{\omega}_{AB}^A \times \tag{17.51}$$

Note that relative velocity of a point in body B with respect to the reference point P due to rotation of body B expressed in global frame is given by

$$\mathbf{v}_{PB} = \boldsymbol{\omega}_B \times \mathbf{r}_{PB}$$

The same velocity expressed in the body frame A defined by rotation matrix $\mathbf{R}_A$ can be found as

$$\mathbf{v}_{PB}^A = (\mathbf{R}_A)^T \mathbf{v}_{PB} = (\mathbf{R}_A)^T \boldsymbol{\omega}_B \times \mathbf{R}_A (\mathbf{R}_A)^T \mathbf{r}_{PB} = (\mathbf{R}_A)^T \boldsymbol{\omega}_B \times \mathbf{R}_A \mathbf{r}_{PB}^A$$

Note that $\mathbf{r}_{PB}^{A}$ is a relative vector in body B expressed in body A frame. Therefore, the last expression implies that the axial vector

$$\boldsymbol{\omega}_B^A = \operatorname{axial}\left((\mathbf{R}_A)^T \boldsymbol{\omega}_B \times \mathbf{R}_A\right) \tag{17.52}$$

represents the angular velocity of body B expressed in body frame A. From (17.51) and (17.52) follows the relation between the relevant angular velocities of the revolute joint

$$\boldsymbol{\omega}_B^A = \boldsymbol{\omega}_A^A + \boldsymbol{\omega}_{AB}^A \tag{17.53}$$

Note also that

$$\boldsymbol{\omega}_{AB}^A = \left(0\ 0\ \dot{\varphi}\right)^T \tag{17.54}$$

where φ is the joint's angle of the rotation.

We can now develop the bond graph component model of a revolute joint. Its word model is shown in Fig. 17.11. It has three power port and one control-out port. Two power point serves for connection to corresponding bodies, which rotate about the revolute joint (e.g., the robot links). The third one (shown on the left side) serves for actuating the joint. The signal port serves for extraction of information on the joint angle.

Its structure is shown in Fig. 17.12 and consists of two main parts: Revolute and Joint rotation. The Revolute represents the basic relations between the port variables in the frame of lower joint part (body A). The other component, Joint rotation, transforms the port variables from the frame of the upper body (body B) to the frame of the lower (body A).

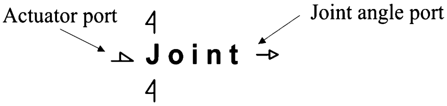


Fig. 17.11 Revolute joint component model

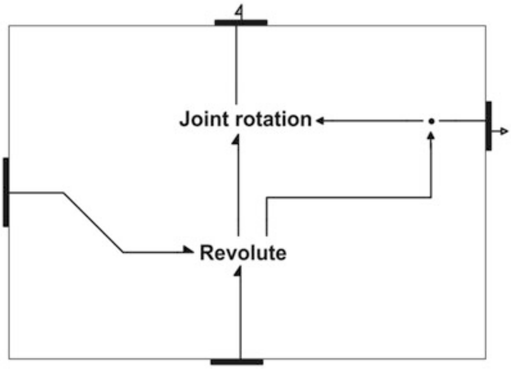


Fig. 17.12 Structure of the revolute joint component model

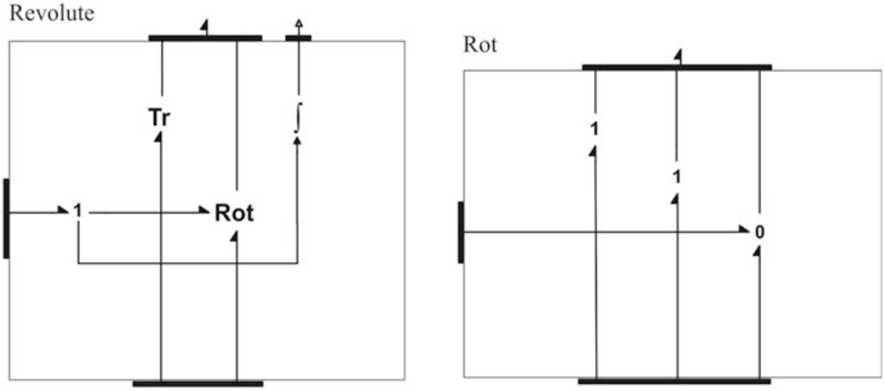


Fig. 17.13 The structure of: Revolute component, Rot subcomponent

The structure of the Revolute component is shown in Fig. 17.13 on the left. It consists of two components. The Tr component represents the translational part of the joint model and expresses the fact that the joint center point is common to the both bodies. Thus it consists solely of three effort junctions for the translation in the direction of x_A, y_A, and z_A axes. This ensures that the joined ends of the bodies move with the common velocity and that the corresponding forces are equal.

The other component, Rot, whose structure is shown in Fig. 17.13 on the right, represents the relationship between the angular velocities as given by (17.53) and (17.54). The model is very simple and consists of two effort junctions in the x_A- and y_A-directions in which the angular components are the same for both of the joined bodies. There is also a flow junction that corresponds to the relative rotation about the z_A-axis. The 1-junction on the left in Fig. 17.13 is inserted to extract the joint relative angular velocity and to obtain the joint rotation angle by integration. It is not only used in the Joint rotation component, but is also transmitted to the output port (Fig. 17.12).

An important function of the joint is the rotation transformation between the two link frames (Fig. 17.12). This is depicted in Fig. 17.14. The transformations are applied to the linear effort-flow parts, and separately to the angular. These are represented by components RAB. The components represent transformations, as given by (17.44) and (17.46) for linear variables. The same transformations hold for the angular variables.

We do not give here the general structure of the component because it depends on the structure of the kinematic links, which it is a part. Revolute joints play an important role in the design of robotic manipulators. They offer the simplest way to change the orientation of the robot links. The component model introduced here gives the main functionality of such joints. They are used later for the building of manipulator models.

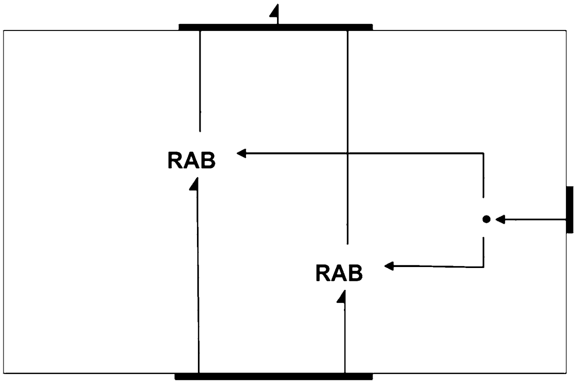


Fig. 17.14 Rotational transformations by the joint

17.3 Dynamics of an Anthropomorphic Robot Manipulator

In this section we apply modeling approach developed in the previous sections to modeling the dynamics of anthropomorphic robot manipulators. Many industrial manipulators are of this type. The first robot with anthropomorphic configuration was famous PUMA 560 released in 1978. As an example of such robots ABB IRB 1600 was chosen [5]. We concentrated on manipulator in open loop to show typical dynamic behavior of the robot. The closed model can be easily developed in a similar way as in Damic and Montgomery [3] for PUMA.

17.3.1 Kinematics Structure of ABB IRB 1600

The IRB 1600 is a six degree of freedom manipulator using revolute joints only. The robot coordinate frames are shown in Fig. 17.15. The links body frames are attached to the corresponding joint with one of the axes directed along the joint rotation axis. However, it is not required to choose for it the Z-axes, as in Denavit–Hartenberg scheme [9, 13].

The joints and frames are numbered from the base to the robot tip. They are generated simply by translations from base $X_0Y_0Z_0$. Thus, the first link frame $X_1Y_1Z_1$ in the zero angle pose coincides with the base frame. The second frame can be obtained from the first by translation along X-axis for L_2 and along Z-axis for L_1. Similarly hold for the third and the fourth joint frame. The rotation axes of the last three joints intersect in the common point and are orthogonal. Therefore, the last three frames coincide in the zero angle pose. As is well known in anthropomorphic robot configurations the human-like terms are used for joints and robot links, i.e., body, upper- and fore-arms, the shoulder, elbow, and wrist.

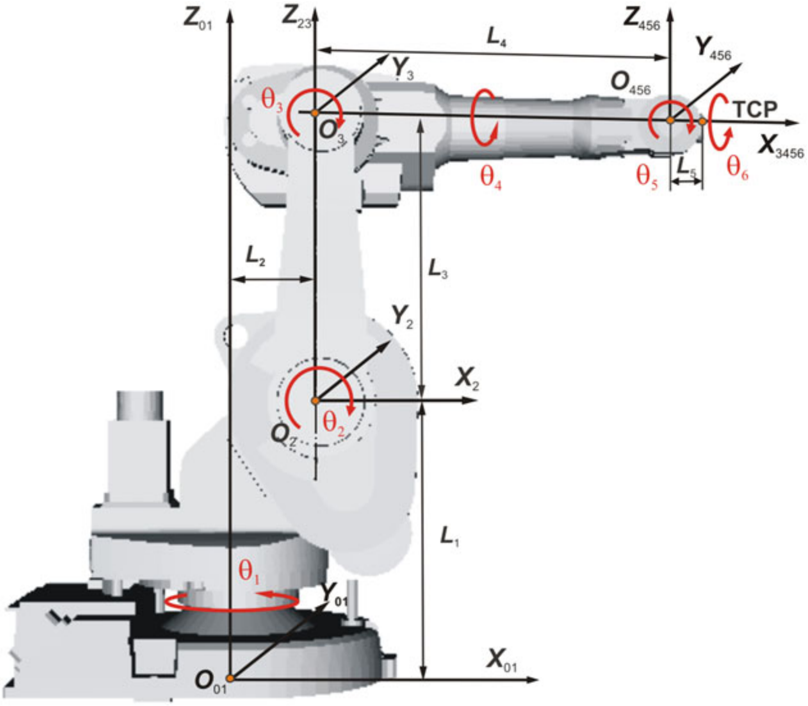


Fig. 17.15 Kinematic structure of IRB 1600 robot

The assumed values of the parameters in [m] are

- $L_1 = 0.4865, L_2 = 0.150, L_3 = 0.475, L_4 = 0.6, L_5 = 0.065$

The working range for IRB 1600-x/1.2 is [5]

- $-180° \le \theta_1 \le 180°$
- $-63° \le \theta_2 \le 136°$
- $-235° \le \theta_3 \le 55°$
- $-190° \le \theta_4 \le 190°$
- $-115° \le \theta_5 \le 115°$
- $-288° \le \theta_6 \le 288°$

The kinematics is relatively simple. There are only three different rotation matrices: $\mathbf{R}_x$ (joints 4 and 6), $\mathbf{R}_y$ (joints 2, 3, and 5), and $\mathbf{R}_z$ (joint 1). The corresponding matrices read:

$$\mathbf{R}_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \tag{17.55a}$$

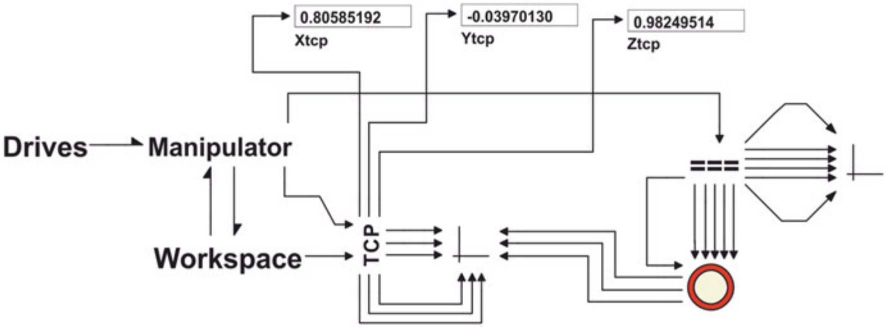


Fig. 17.16 System level model of IRB 1600 robot

$$\mathbf{R}_y = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \tag{17.55b}$$

$$\mathbf{R}_z = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{17.55c}$$

17.3.2 The Basic Structure of the Model

A brief description of Bond Graph model of the robot is given. The model has a hierarchical structure, which reminds of the structure of the real robot. The complete model can be downloaded from the web page[1] and analyzed using BondSim© program.

The system level model of robot ABB IRB1600 is shown in Fig. 17.16. It consists of three main components: Manipulator, Workspace, and Drives.

The first two represent the model of the robot, and Drives represents simple open-loop drives of the robot actuators. The other components serve to post-process the signals and display them in the display components, which show changes with time of different variables of the interest during the simulation.

[1] http://www.bondsimulation.com is BondSim program web page, from which the basic version BondSim2015 of the program can be freely downloaded and used. The corresponding ABB_IRB1600.ept file, which contains the complete model including the last simulation results, can be downloaded as well. It can be saved to a suitable place (hard disk, external disk, or memory stick), from which it can be imported into the BondSim2015 using the Import command.

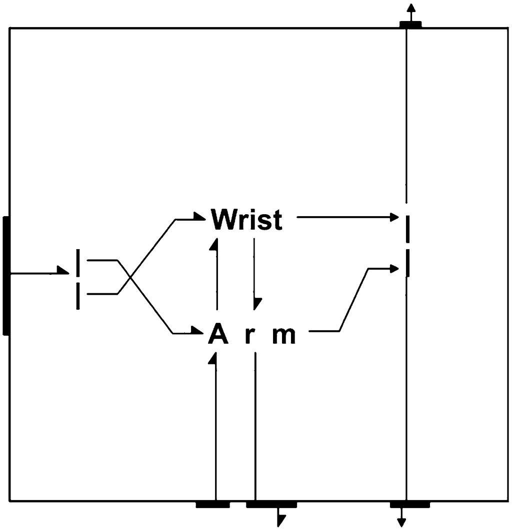


Fig. 17.17 The structure of the manipulator model

A special role in this model plays the component in the form of a ring (pipe). It is a component which enables inter-process communication between the robot dynamical model (within BondSim© program) and 3D geometric (visual) model of the same robot (within the corresponding BondSimVisual© application). It is really the client side of the communication (named) pipe. We will come to this topic later.

Model of the manipulator is shown in Fig. 17.17. It consists of Arm and Wrist components. On the left are two bonds, which deliver power to the arm and wrist actuators from the left multi-port connector. Similarly the connector on the right picks and further distributes the joint angles and angular velocities from these components.

The arm and wrist are interconnected by bond lines. It is assumed that power flows from the Workspace (Fig. 17.16) over Arm to Wrist. The wrist with a tool acts to the workspace. Therefore it delivers power to the workspace. However, the wrist and arm "feel" these interactions, and there is transfer of power from the Wrist (and tool) over Arm to Workspace. Thus, the model structure in Fig. 17.16 depicts what really happens in the robot and thus represents a pretty close physical model of it.

The models of Arm and Wrist are shown in Fig. 17.18. The Link 1 models the robot body, which can rotate about vertical the robot base axis (Fig. 17.15). Joint 2 is the shoulder, about which the upper arm (Link 2) can rotate. This arm is further connected by the elbow (Joint 3) to the forearm (Link 3), which carries the wrist. The wrist (Fig. 17.18 on right) represents a spherical joint (see [3] for details). It consists of Joint 4–6, and Link 4–6 that house the actuators driving the joints.

Consider the bonds connecting the Joints and Links. The left ones we already have discussed. More interesting are bonds going down from the last Link 6 to the base. We remind of the modeling approach we discussed in Sect. 17.2.2, Fig. 17.2. We use three-port component models to represent space body models. Thus, the

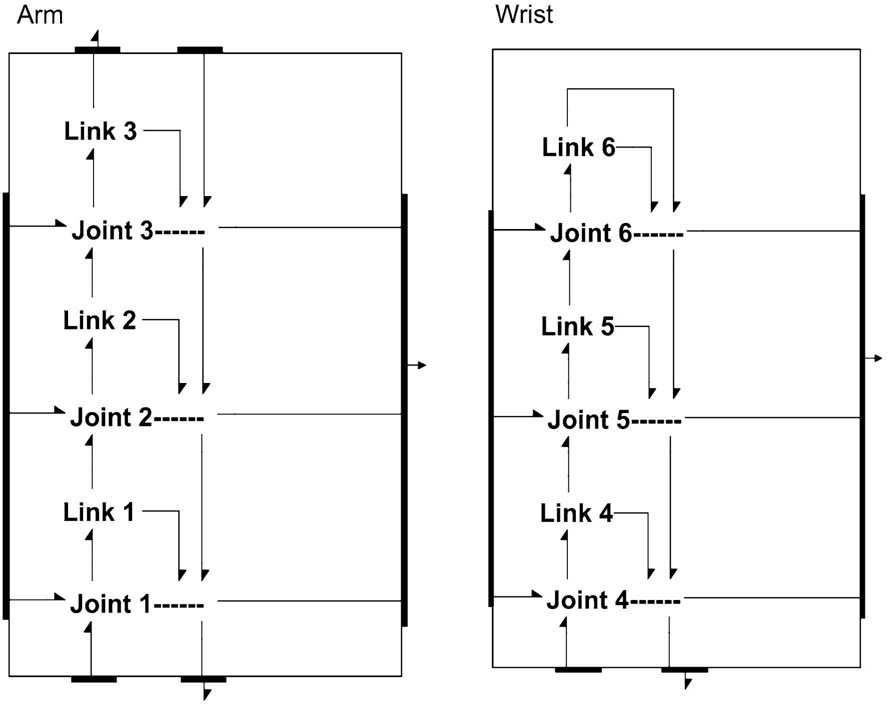


Fig. 17.18 Models of Arm (*left*) and Wrist (*right*)

bond coming out of the upper port of Link 6 transfers 3D efforts and flows of the robot wrist (holding a tool) action to the workspace. In addition at it's center of mass there are the resultant force and its velocity. All these quantities are described in the body (Joint 6) frame. There are different for an observer in the base. However, we do not know exact position and orientation of this frame in the base, and thus cannot transform them directly. Instead, we systematically transform these quantities from the joint to joint until the base. The same holds for the other links as well, only that there are smaller number of the quantities to transform as we approach the base.

Note how the bonds are gradually built. It is important because we need to unpack them later. At every joint to the already transformed quantities we add the corresponding link center of mass quantities and transform these new quantities further. This looks somewhat complicated, but if we apply this approach systematically there should be no problem. The similar effects appear in the reality. When we hold a heavy load in our hand, we feel it at every body parts to the foot.

To see these transformations in more detail we open, e.g., Joint 2 component (e.g., by double clicking). The model of Joint 2 on the left is similar to that of

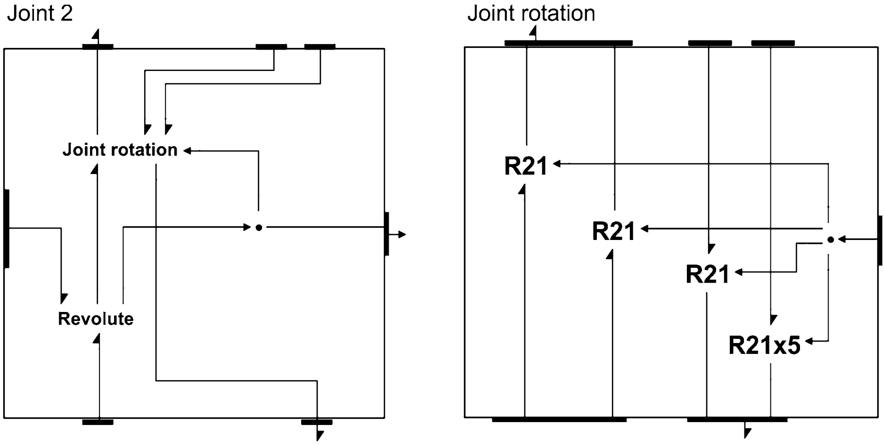


Fig. 17.19 Structure of Joint 2 model component

Fig. 17.20 Component R21: rotation about Y-axis

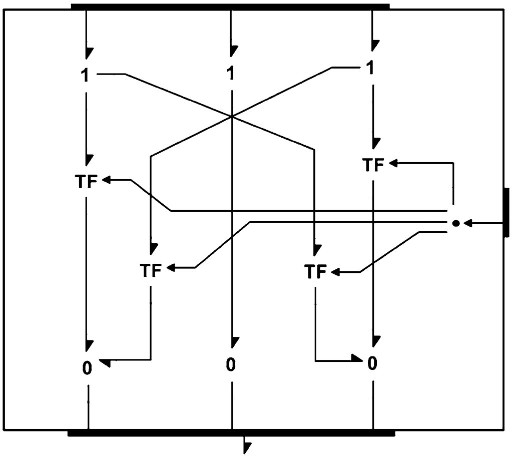


Fig. 17.12. There are, however, the additional two top right ports, which transfer the Link 2 center of mass quantities and the quantities from the previous joint, respectively (Fig. 17.18).

The Revolute component, which model the revolute joint, was already discussed in Sect. 17.2.4, Fig. 17.13. The Joint rotation model component is shown in Fig. 17.19 on the right. The **R21** component describes the transformation corresponding to rotation about axis Y through the joint angle θ_2, Fig. 17.15. The transformations are governed by matrix (17.55b) and are shown in Fig. 17.20.

We direct at the end our attention to Workspace component in Fig. 17.16. The model is shown in Fig. 17.21 left. We may first observe Robot base component. It is a simple component consisting of six source flows, which fix the base to the ground in its initial position, i.e., they imply zero linear and angular velocities of the base.

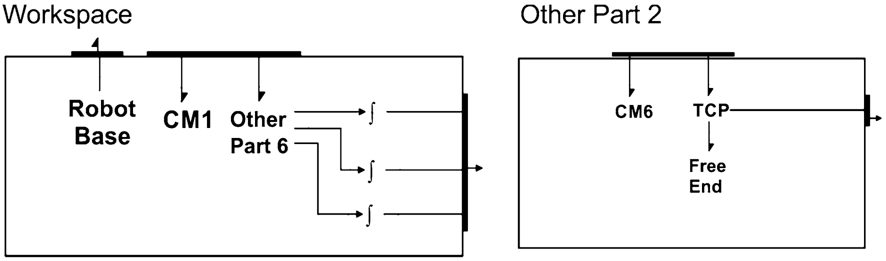


Fig. 17.21 The structure of model of the Workspace

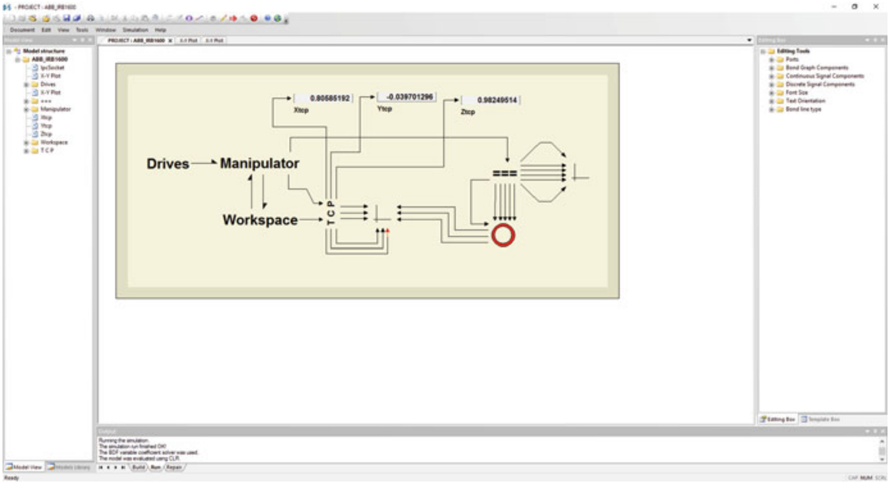
The next component CM1 describes the dynamics of the Link1 center of mass. We have discussed it earlier in Sect. 17.2.3 (see Fig. 17.7). To establish the correct connection to the manipulator (Fig. 17.18) we add also a component called Other Part 6, indicating that it must contains six other components—the dynamic of centers of masses of other five links, and the robot wrist end component representing the TCP (the tool center point). The structure of the last component Other Part 2 is shown in Fig. 17.21 right. It contains CM6 component describing the dynamics of the center of the mass of the last link and the TCP component. The last component consists simply of three 1-junctions connected to the components port. They represent the nodes of TCP velocity components in the basic frame. Free End component defines the conditions at the robot end. It could be simply zero source efforts if the robot end moves freely in the environment. The bond at the right of TIP extracts its velocity. These signals go out from Other Part 6 in the Fig. 17.21 left and are integrated to obtain information on TCP position in the base frame.

One of the strong points in dealing with multibody dynamics as described above is a systematical use of compounded ports, i.e., the ports that consist of other subports, these of the others, and so on to the arbitrary depth. This enables representing of the complex interaction appearing in the devices of such a complexity in a straight and readable manner. Otherwise we should deal with forest of the signals. Of course the corresponding mathematical models could be difficult or even impossible to obtain by hand. BondSim© model build algorithm knows how to do it automatically without intervention of the modeler.

17.3.3 Simulation of the Robot Motion

We will now conduct the simulation of developed model of ABB IRB 1600 robot. We consider behavior of the robot in open loop. Thus, we will drive robot by the source flows, which generates the angular velocities of the joins defined by simple sinusoidal laws

$$\omega_i = \alpha_i \omega_0 \cos(\omega_0 t), \quad (i = 1, \ldots, 6) \tag{17.56}$$



```
Running the simulation.
The simulation run finished OK!
The BDF variable coefficient solver was used.
The model was evaluated using CLR.
```

Fig. 17.22 BondSim© program screen with open ABB_IRB1600 project

This correspond to rotations of the links given by

$$\theta_i = \alpha_i \sin(\omega_0 t)\,, (i = 1, \ldots, 6) \tag{17.57}$$

The circular frequency of the sinusoidal function was select relatively slow, e.g., $\omega_0 = \pi/10$ rad/s, and the amplitudes are set to

- $\alpha_1 = \pi$, $\alpha_2 = \pi/3$, $\alpha_3 = 0.25\pi$, $\alpha_4 = \pi$, $\alpha_5 = 0.6\pi$, $\alpha_6 = \pi$

We run the robot joint angles for one period of the input sinusoid, i.e., the simulation time is $t = 20$ s. Figure 17.22 shows BondSim© program screen with open project ABB_IRB1600.

The system level model of the project is shown in the central part of the screen. On the left is a window, which displays hierarchical structure of the model in the form of an Explorer style tree.

The enlarged part of this tabbed window is shown in Fig. 17.23 left. It shows the hierarchical structure of the model. The model project is at the root, and then the components. The components with the “folder” symbol denote the word model components, which contains the other components. By clicking the leading “+”, the corresponding component in the main windows opens. The same could be achieved by double clicking the component title in the main window. The components denoted by page symbols are elementary Bond Graph components.

The next tab contains Model Library (Fig. 17.23 in the middle). It is used often during development of models. It contains the complete projects, and different components, which are stored in the library (see [3] for details).

The basic editing tools are contained in Editing Box at the right side of the BondSim© screen (Fig. 17.22). Its enlarged view is shown in Fig. 17.23 right. It

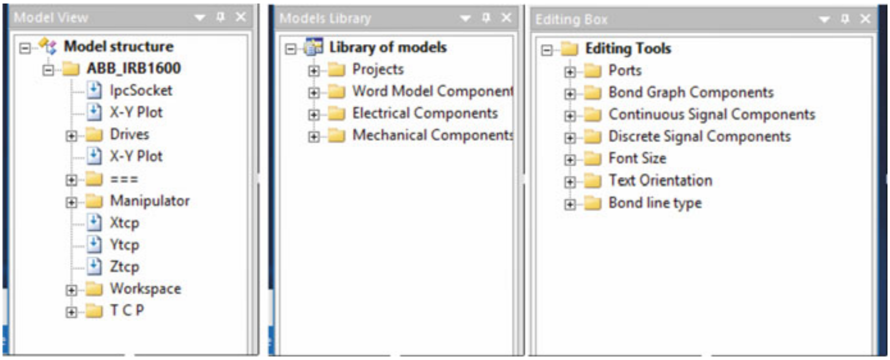


Fig. 17.23 Enlarged parts of the BondSim© screen

contains several tools for developing Bond Graphs, including the continuous and discrete time block diagrams. The development of the models is basically conducted by dragging the tools from Editing box and dropping them into the main windows (refer to [3] for details).

Before starting simulation we need to build the mathematical model of the system. This can be done by opening *Simulation* menu, and choosing *Build* command. The system starts reading all components the system consists of, checks the connections, generates the internal variables, and finally generates the mathematical model. The model is generated in symbolic form, which can be recompiled into a human readable form for inspection. We will not go in details here. Interested reader can consult [3] for details.

The current model consists of 959 implicit equations, and symbolically generated Jacobian matrix contains 2637 elements. Thus, it is very sparse, with on average less than 3 elements per row.

To start simulation, chose *Simulation* command *Run*. In the dialog that opens set the simulation interval to 20 s, the output interval, and maximum step size to 0.05 s. (The output delay interval is not important now.) For all other simulation parameters accept the default values (the method and error tolerances). Click *Restart* (if it is not already set) and select OK. Now simulation starts. System generates the messages in Output window at the bottom, which inform the user on simulation advance.

During a simulation run the model is repeatedly solved using a Backward Differentiation Formula (BDF) type solver. The solver successfully solved the model as the simulation statistics in Table 17.1 shows.

Figure 17.24 shows the motion of the TCP during the simulation, and Fig. 17.25 shows how changes the absolute error in TCP position.

These errors were calculated by comparing the values generating by the simulations and values calculated by the direct kinematics in component TCP. Diagram shows that these errors are of order of the simulation error tolerances in Table 17.1. Thus, the accuracy of the simulation is good.

Table 17.1 Simulation statistics

Name	Value
Simulation time	20 s
Absolute and relative error tolerances	1.0e-6
Number of the steps	414
Number of the function evaluation	851
Number of the partial derivative matrix evaluation	449
Number of matrix ordering and decomposition	24
Number of fixed order decomposition	418
Total calculation time	0.55 s

The simulation was conducted on a laptop with i7 processor, installed RAM 12 GB, and SSD with 224 GB

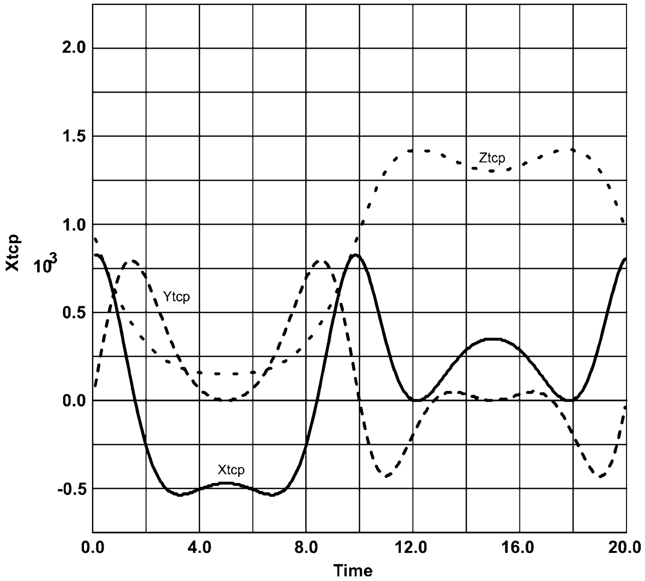


Fig. 17.24 Changes of coordinates of TCP with time

17.4 Visualization of Robots in 3D

In the previous section we dealt with the dynamics of a robot. It gives a deeper insight into its behavior. However, it is rather an esoteric topic. Perhaps visualization of its motion in a virtual 3D space is more informative. We wish to stress that we are not speaking of animation of a robot motion, but on visual representation of simulated robot motion in 3D scene on computer screen. Thus, there is one-to-one correlation between the simulated robot motion of the last section and its visual 3D representation.

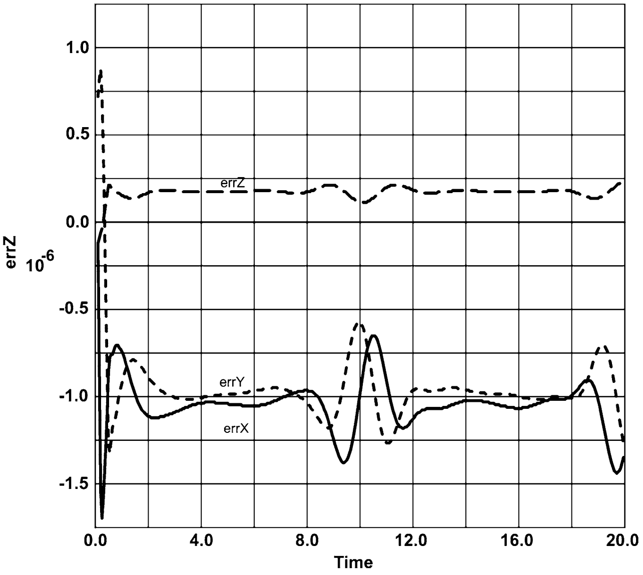


Fig. 17.25 Errors in TCP coordinates

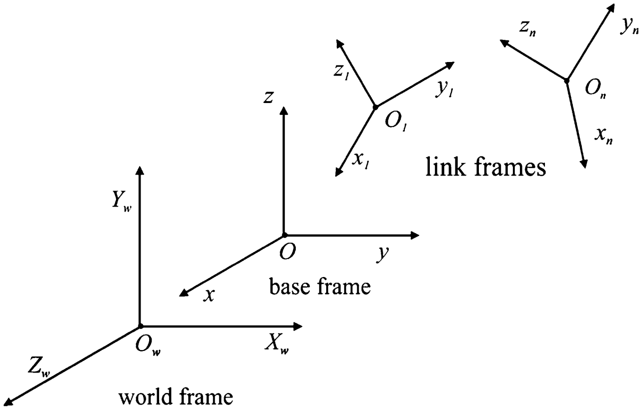


Fig. 17.26 The coordinate frames

Many robot manufacturers use 3D robot models for off-line programming, e.g., ABB [11], Fanuc [10], KUKA [7], BYG [4], and others. These software applications were designed with the specific goals in mind and are based on geometric and kinematical models.

The geometrical models driven kinematically are useful for solving some problems, such as off-line programming, but cannot simulate the real behavior of the robots. It is quite complicated, however, to have in one application both the geometry and dynamics. Thus, we developed a separate application *Bond-*

SimVisual©, which supports developing of 3D geometric models of robots and similar mechanical systems and their rendering on the computer screen in a 3D virtual scene. The simulated motion of a robot using *BondSim*© can be visualized in parallel in a separate window on the same or another computer by inter-process communications between these two. In this section we describe concept of visualization employed. Inter-process communication between the dynamical and virtual 3D space is discussed next.

17.4.1 Description of Robot System

The first step in generating 3D virtual scene is geometric description of the system that is the subject of the study. Usually it is the same system for which bond graph dynamical model was developed. The system is described in form of a relatively simple textual script written in ASCII code using corresponding commands. We will next describe the basic commands and show how a typical robot can be described. Note that in spite we are speaking of robots, the same is applicable to other mechanical objects like see-saws in children play grounds, the legged platforms, cranes, and similar 3D mechanisms.

Description of a robot or any other multibody mechanism can be done in two basic steps:

- Description of kinematic structure of the robot. This in essence defines all necessary coordinate frames and their relationships.
- Description of all main parts that robot consists of such as the base, links, tool, and its environment.

The description is given in form of a simple script, which is later used by the visualization program to create in the memory all objects that represent the 3D visual model of the robot. These objects draw itself on the computer screen as 3D picture of the robot. It is shown in a specified initial position.

17.4.1.1 Kinematic Structure of the Robot

The kinematic structure of a robot system defines relationships between different coordinate frames used to describe the geometry of a device. We use only the right Cartesian coordinate frames. There are three types of coordinate frames we use (Fig. 17.26). The first is the global World coordinate system $X_wY_wZ_w$. The origin of the system is in the center of the screen with X_w-axis directed horizontally to the right, Y_w-axis vertically upward, and Z_w-axis out of the screen. To every mechanism in the scene there is attached a *base coordinate frame*.

In addition to every link there is also attached a *link frame*. The origins of the frames are set at center point of the corresponding joint. The joints, and corresponding links and their frames, are ordered going from the base to the tip.

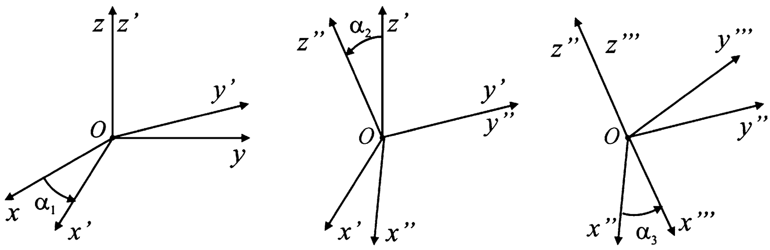


Fig. 17.27 Euler ZYZ angles

To define position between the coordinate frames the homogenous transforms are used (see Sect. 17.2.1). We assume that they are coincident at first. We rotate then first one frames with respect to the other about the common origin by a rotation matrix, and then translates it by a 3D vector. To define the rotation we use *ZYZ* Euler angles (Fig. 17.27).

The homogenous transform of one frame with respect to the other is defined in the form

```
(Shift x value1 y value2 z value3 Euler angle1  angle2  angle3)
```

Such a specification is used in different commands that we will encounter later.

The position of the base frame in the word coordinates and its kinematical structure is defined by the *Device* command, which has the form shown below. The specific command words are shown here in bold. They can be written in small, large, or mixed letters. The others are the parameters defined by the modeler. The terms in the command are separated by spaces. The command ends by semicolon. The same holds for the other commands as well.

```
Device Title (Shift x shx y shy z shz Euler eu1 eu2 eu3 LBox blen)
      Joint 1 (Shift x shx y shy z shy Euler eu1 eu2 eu3)
         Prismatic or Revolute x or y or z
         (Shift x shx y shy z shy Euler eu1 eu2 eu3)        . . .
      Joint n (Shift x shx y shy z shy Euler eu1 eu2 eu3)
         Prismatic or Revolute x or y or z
         (Shift x shx y shy z shy Euler eu1 eu2 eu3)
      Tcp (shift x shx y shy z shy)
      Extend joint_index
      Initial (value_1 value_2 . . . value_n);
```

The title of the device is specified first, then its position in the world frame, and the size of the bounding box. Next the first link frame is defined with respect to the base, then the second with respect to the first, and so on until the last link frame is defined.

Note that every joint is numbered going from 1 and up (the index 0 is reserved for the base). Every joint frame in the kinematic tree is defined first by specifying a pre-transform, then type of the joint, and finally a post-transform. Usually only pre-transform is enough, but sometimes we need both pre- and post-transforms. It assumes that joints are simple revolute or prismatic. Its rotation or sliding axis is aligned with one of the link coordinate axes.

The Tcp command defines position of the *tool central point*. The mechanisms often do not have the simple serial structure, but contains the parallelograms, or more than one arm. In that case subcommand Extend defines the index of the joint or base where the branching starts.

Finally there are subcommands that define the initial values of the joint variable (angle or displacement). The command defines only the initial position of the mechanism. It could be moved manually by a menu command or dynamically by another application (BondSim©).

17.4.1.2 Geometric Representation of the Robot Parts

The geometric model of the device base or its links can be constructed in two ways—using the primitive forms or 3D CAD generated part files. The predefined primitive forms are shown in Fig. 17.28. To define the mechanism structure we used the corresponding commands, which are based on corresponding *VTK* classes [12].

Cylinder

It serves to create a simple body in form of a cylinder (Fig. 17.28a). Its longitudinal axis is aligned with z axis. The parameters of the command are the dia[meter] of

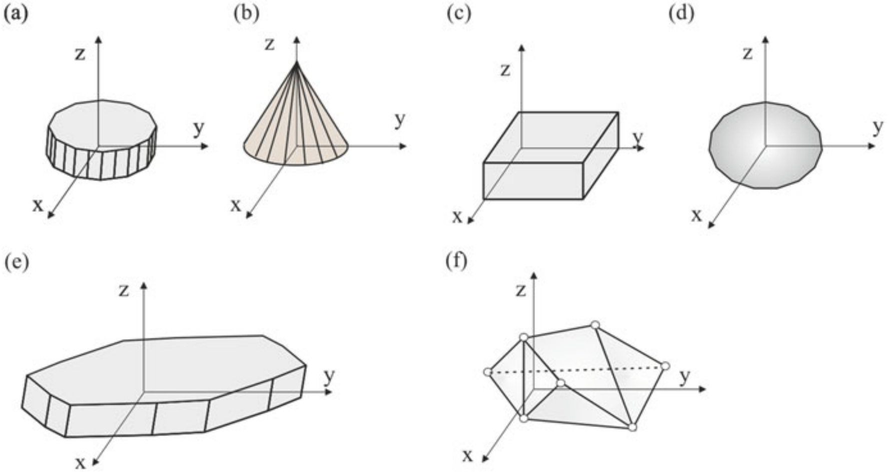


Fig. 17.28 Primitive forms: (**a**) Cylinder, (**b**) Cone, (**c**) Cube, (**d**) Sphere, (**e**) Polyprism, and (**f**) Module

its base and its len[gth] (height) in millimeters. The cylinder bases are generated as regular polygons. The parameter res[olution] defines the number of divisions of the base circles. The cylinder center is on *z*-axis at a half of its height (length). The syntax of cylinder command is

```
Cylinder Title dia cdia res cres len clen;
```

Cone

A cone is defined in similar way (Fig. 17.28b). The base of the cone is circular with its center at the origin of the coordinates, but can be set to any other point. Similarly, direction vector from the center toward the apex is aligned with *z*-axis, but can be set to different directions. The other parameters are dia[meter] of the base and its len[gth]. The circle of the base is generated as a regular polygon whose number of divisions is defined by the res[olution] parameter. The syntax of cone command is

```
Cone Title (Center x value y value z value Direction x value
y value z value) dia cdia res cres len cH;
```

Cube

It serves to create a simple body whose three orthogonal edges are aligned with *x*, *y*, and *z* axes (Fig. 17.28c). The parameters of the command are the lengths of the edges *xLen*, *yLen*, and *zLen*. The coordinates of the cube center are (0.5 *xLen*, 0.5 *yLen*, and 0.5 *zLen*). The syntax of cube command is

```
Cube Title xLen yLen zLen;
```

Sphere

It serves to create a simple body with its center in the origin of the coordinates and radius (Fig. 17.28d). It is generated as a polygonal object with a res[olution]. The syntax of the sphere command is

```
Sphere Title radius  sR res sres;
```

Polyprism

This command serves to create a prismatic body, which is generalization of a cylinder or cube (Fig. 17.28e). Its base is a closed polygon formed of the straight lines and circular arcs. The syntax of the command is

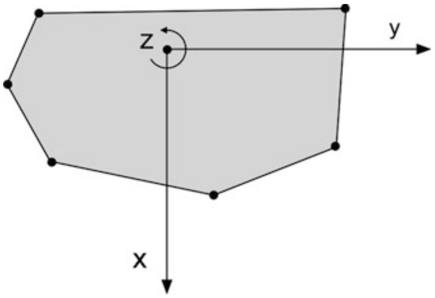


Fig. 17.29 Definition of the coordinates

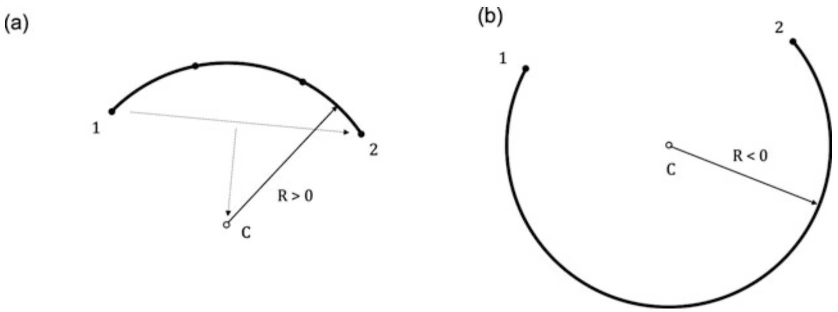


Fig. 17.30 The arcs of Polyprism: (**a**) Normal and (**b**) Large

```
Polyprism Title
Height value
Coord1 coord2
...
coord1 coord2
Arc to value1 value2 radius value res value
Arc Large to value1 value2 radius value res value
...
;
```

The Polyprism base lies in *xy* coordinate plane and its edges are orthogonal to the base and parallel to *z* axis. The command specifies the name of the polyprism and its height. The coordinates of the vertices are defined by its base-plane coordinates (Fig. 17.29).

The arcs are drawn between the point specified by the previous vertex and the point defined by the *x*–*y* coordinates after to subcommands. The first form defines a (normal) short arc between the end points (Fig. 17.30a), its radius, and the resolution. The other form is similar but defines a large arc (Fig. 17.30b). The positive value of the radius denotes that the arc is drawn in clockwise direction Fig. 17.30a. If it is negative it is drawn in the opposite direction.

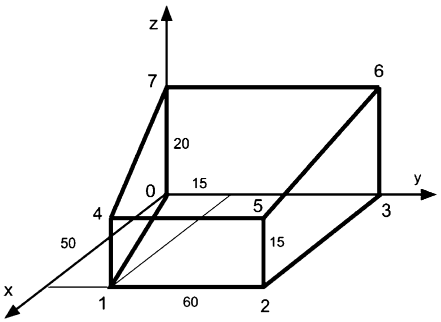


Fig. 17.31 Body in form of a wedge

Module

This command is used to define a general form of a body (Fig. 17.28f). It defines the name (title) of the body. The Points gives the list of vertices defined by their indices and the triplets of their x, y, and z coordinates in the local frame. The polygonal sides of the module are defined by Polygons, which consists of the list of indices of vertices enclosed in parentheses. To illustrate how such a form can be defined consider a body in form of a wedge in Fig. 17.31.

To define the body by the module command, we define first its name, and an ordered list of the coordinates of the vertices (points), and finally the polygons (sides). The vertices are numbered from 0 to 7 and are defined by specifying its coordinates in the local coordinate frame. The polygonal sides are defined by listing the indices of their vertices. The corresponding command reads:

```
Module Wedge
Points
0 0.0 0.0 0.0
1 50.0 15.0 0.0
2 50.0 75.0 0.0
3 0.0 75.0 0.0
4 50.0 15.0 15.0
5 50.0 75.0 15.0
6 0.0 75.0 20.0
7 0.0 0.0 20.0
Polygons
(0 1 2 3)
(1 2 5 4)
(2 3 6 5)
(3 6 7 0)
(7 4 1 0)
(4 5 6 7)
;
```

Parts

The bodies that make the robot can be generated using the part files generated by a Computer Aided Design (CAD) system such as Catia, Solid Works, ProEngeener, and the others. The robot manufacturers as a rule publish on their web sites the CAD files of the robot parts in the standard formats. BondSimVisual© supports the *stl* (stereo-lithography) format. This format is almost exclusively used for visualization and rapid prototyping. Even if original files are not in *stl* format but in some other public format, e.g., *STEP*, *DXF*, etc., they can be readily converted into it. Such files can be imported into BondSimVisual© program. To use parts defined in such way the Part command is used. It simply defines the filename of the imported part file (without extension *.stl*):

```
Part Filename;
```

Set

This command is used to create a complex body consisting of the simpler ones. We assign the name to the new assembly and add to it already defined bodies. These component bodies can be translated or rotated with respect to the local frame. This is described by the corresponding transforms enclosed in the parentheses. The syntax of the command is shown below.

The important application of the Set command is in creation of the device links. Because the links are associated with the joints, the same name is used for the both. In the command the link is defined by *devicet_title#joint_index*. Due to the fact that the link coordinate frame was already defined it is taken as the local frame of the assembly.

```
Set Title add comp_1 (Shift x shx y shy z shy Euler eu1 eu2 eu3)
              comp_2 (Shift x shx y shy z shy Euler eu1 eu2 eu3)
              .....
              comp_n
              ...
              ;
```

Therefore, the Set commands are commonly used to assign the body assembles to the joints and thus create the links.

Render

All the previous commands serve to define the structure of the mechanism. The Render command is really a visualization command. It asks the system to render the corresponding component and shows it on the screen. Its format is

```
Render Title color red green blue;
```

Table 17.2 Common colors defined by RGB values

Color	*RGB*
Black	0 0 0
White	1 1 1
Red	1 0 0
Green	0 1 0
Blue	0 0 1
Yellow	1 1 0
Cyan	0 1 1
Magenta	1 0 1
Sky blue	0.5 0.5 1.0

The *title* defines the component which is the target of the command. It can be the base of the device, specified by its title, the links specified by the device title and its joint index attached by “#” sign (similarly as in Set command), e.g., IRB1600#2. The command also defines the *RGB* color values of the rendered component by specifying *Red*, *Green*, and *Blue* values in the range 0.0–1.0. Some common RGB values are given in Table 17.2 [12]. Note the “colour” is acceptable as well.

Probe

This command is used to specify a point in the mechanism we are interested in. It can be a part of the mechanism, but can be a separate point. The program calculates its coordinates and outputs it. The syntax is:

```
Probe Title Frame_in (Shift x shx y shy z shz)Refer Frame_to;
```

The position of the probe is specified by Shift command (within the brackets) with respect to *Frame_in*, which can be the mechanism base, or any of the joint coordinate frames. They are specified as in Set and Render commands. The values of probe’s coordinates are calculated with respect to the same or a different frame. It is specified by Refer and *Frame_to*, which can be World global frame, the robot base, or some of the joint frames. This command can be useful for visualization of the robot motion during its simulation. The visualization program returns the coordinates of the probe, which can be used by the connected (client) program, e.g., BondSim©.

In order to include the comments in the scripts the text line with leading ‘!’ is used. For more text the command ‘$’ followed with text lines and ended ‘$’ may be used. The comments are skipped when processing the scripts.

17.5 The Program Script

As application of the commands given above we analyze the script in Box 17.1, which defines the visual model of ABB IRB 1600 robot of Fig. 17.15. Device command defines the kinematical structure of the robot by defining the frames and initial position of the links. Note the robot title IRB_1600 may be different from the program name under which it was developed and stored. It is used locally inside the script only.

The origin of the base frame coincides with the global frame, but the axes are rotated first for −90° about global Z-axis (out of the screen), which moves x-axis downwards, and y-axis to the right. Next it is rotated for −90° about the new y-axis, which rotates x-axis out of the screen, and z-axis vertically up. This is the position we need and thus the third rotation is zero. The last item in the Device command is the size of the bounding box that defines the extent of the virtual screen. It is used by the windows system to properly scale the screen and do not rescale it every time the robot moves.

Next the joint coordinate frames are defined (Fig. 17.15). Starting from the base frame we have just defined, we assume that the first joint frame in zero angle pose is coincident with it. The joint is the revolute with z rotation axis. The second joint frame we obtain by translating along x-axis for L_2 and for L_1 along z-axis. It is again a revolute joint with y as the rotation axis. Next we translate upward for L_3. The third joint is the revolute with y as the rotation axis. We proceed in the same way. Now we translate along x-axis for L_4. The fourth joint is revolute about x-axis. Note the last three joint makes a spherical joint, which rotation axes are orthogonal in zero angles pose. Therefore, we assume that the last two frames coincide with the fourth joint frame. The fifth joint is revolute with y-axis as rotation axis, and the last joint is revolute again but about x-axis.

Tcp (tool center point) is displaced by L_5 along x-axis. Finally the initial values of all joint angles are defined, which are simply equal to zeros.

The next six commands define the robot parts, the robot base, and six links. The Part names represent the filenames under which the *stl* parts are imported into the program from the manufacturer web page. This enormously simplifies building the visual model of the robot.

The first Set command attaches the Body0 part to the robot base frame and renders it in the gray color. Similarly the next Set command attaches the first link body to the first joint frame. In this way they define the robot body and its first link. The next commands did it with the other links. However, these commands incorporate the shifts, which we will explain.

The parts supplied by ABB are defined with respect to the base coordinate frame. Therefore to define them with respect to the local joint axes the parts must be first shifted back to the base frame and then set to the corresponding joint frame. Note also that all bodies are rendered gray. This follows the recent coloring scheme that ABB Robotics uses.

The final command is the Probe. It is shifted along x-axis of the last joint frame for L_5 and hence coincides with TCP. Its motion is calculated with respect to the base robot coordinate frame. In the visual model of the robot it appears as yellow point, as the robot carries a tiny lamp.

17.5.1 Visualization Based on VTK

Among different systems that can be used for the visualization in 3D space a very interesting and powerful one is the Visualization Toolkit (VTK) by Kitware [12]. It is a large C++ open library. We use only a part of its capabilities. BondSimVisual© is written in MS VisualC++ and uses also Microsoft Foundation Class (MFC) library. The visual models of the systems which we are interested in are described in the form of script document as we discussed in the previous subsection. The management of script document is supported by the program. Thus, it is possible to create a new one, open and modify an existing one, delete copy and export them, etc. Because the scripts are simple text files, as the editor Notepad is used, which is called from the program directly.

To create a 3D virtual scene the corresponding menu command is issued, which offers to the user to select a robot script for which the scene will be generated. The script is then read and translated into series of the byte codes. During this translation phase the script is checked for eventual errors, which are returned to the user. When the translation phase was successfully completed, starts the final phase in which the visualization objects in computer memory are created and corresponding 3D scene, including the lights and camera, is generated.

Visualization in BondSimVisual© is based on VTK concept of visualization pipeline [12]. Thus, following the script the C++ objects are created in the memory and are interconnected as illustrated in Fig. 17.32.

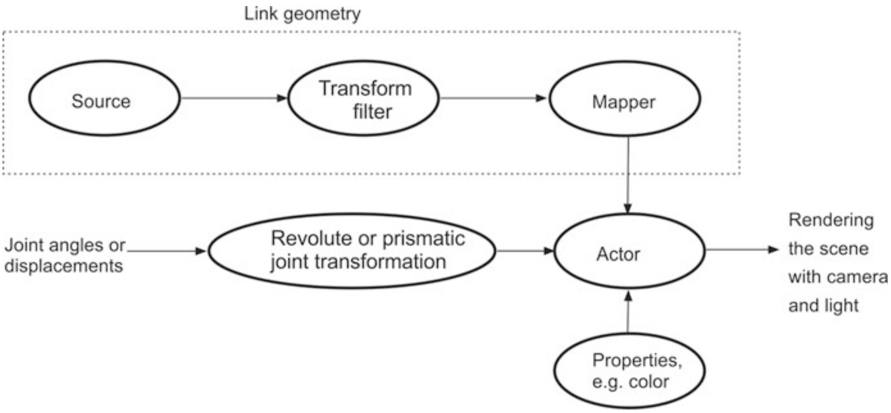

Link geometry
Source
Transform filter
Mapper
Joint angles or displacements
Revolute or prismatic joint transformation
Actor
Rendering the scene with camera and light
Properties, e.g. color

Fig. 17.32 Visualization pipeline

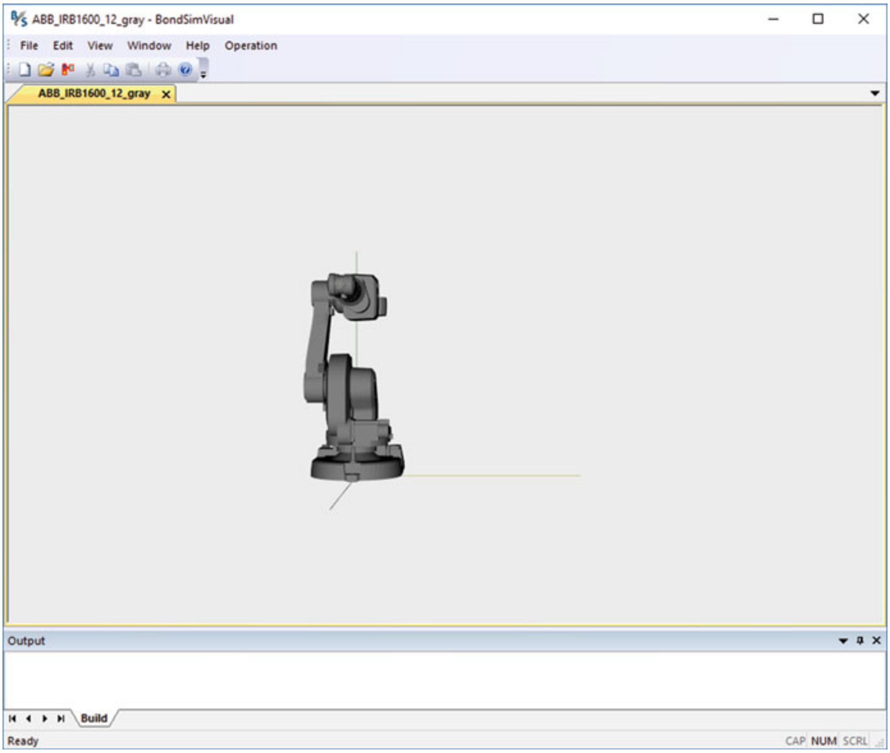

ABB_IRB1600_12_gray - BondSimVisual
File Edit View Window Help Operation
ABB_IRB1600_12_gray
Output
Build
Ready
CAP NUM SCRL

Fig. 17.33 3D scene with ABB IRB1600 robot in the initial pose

The components in the upper part of the figure define the geometry of a link. Typically they use the sources of cubes, cylinders, and others forms, which define the parts the link is composed of. They are transformed and connected to the mapper objects, which defines the geometry of the link. It is also possible to use 3D CAD models of the parts, e.g., in stl format, and using the corresponding reader to read them into the memory. In the similar manner the base is created.

The outputs of the Mappers are connected to the Actors, which are responsible for rendering of the objects. Its properties can be defined, such as color, by Render script commands. Its input port is connected to the transform object, which describes rotation or sliding of the link by the revolute or prismatic joints. If the joint's angles change, then all connected components are reevaluated and the robot changes its position on the screen.

Figure 17.33 shows BondSimVisual© screen showing ABB IRB1600 robot scene. It is shown in the initial zero angles pose.

Program BondSimVisual© is designed to work jointly with another program, which supplies the value of the joint variables, the revolute joint angles, or prismatic joint displacements. The primary dynamic modeling and simulation program is BondSim©, which supplies these data by the inter-process communication (IPC), Fig. 17.34.

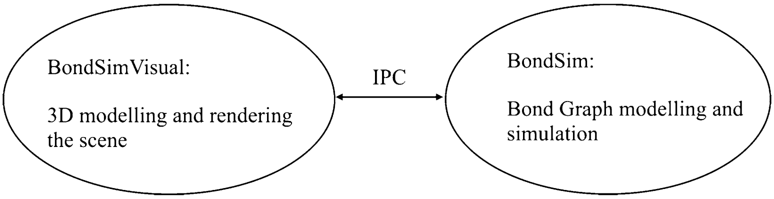

BondSimVisual:
3D modelling and rendering the scene
IPC
BondSim:
Bond Graph modelling and simulation

Fig. 17.34 Concept of 3D visual and dynamic modeling and IPC

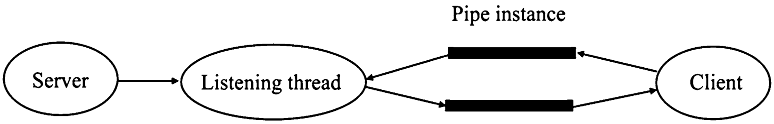


Fig. 17.35 The configuration of IPB using named pipe

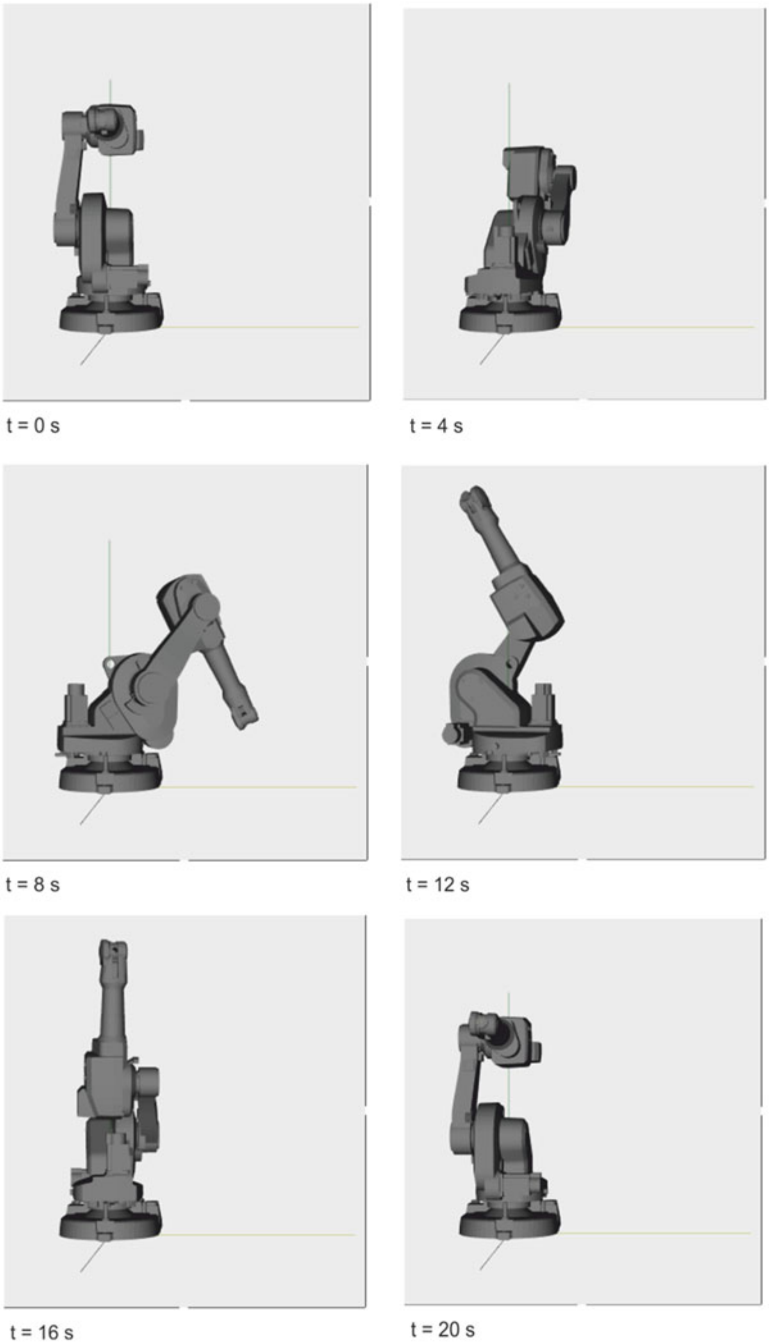
The BondSim© sends regularly, e.g., every 50 ms, the current values of joint variables to the BondSimVisual©, which processes them and redraws the virtual scene. In response it returns, e.g., the new position of the Probe to the BondSim©, which use it for its purposes. Currently, it plots it alongside with other variables. However, such information could be useful, e.g., for modeling and simulation of the crushes.

The interval for sending the data is defined at the start of the simulation. It is the *output delay* whose default value is 50 ms, which amounts to 1/0.050 = 20 frames/s. This is a common value, but could be set to a different value.

Inter-process communication is based on the named pipes [8] (Fig. 17.35). The server (BondSimVisual©) is responsible for creating the pipe. It also creates a special listening thread, which enables that program simultaneously with the other tasks listen to messages from the client. The server also sends a message to the potential clients to connect to. The client BondSim©, after building the simulation model of the corresponding modeling project, connects to server using a command in the Simulation menu.

When the two-way communication is established the simulation can start on the client side. As the simulation advances the robot starts moving its arm over the virtual scene. Figure 17.36 shows a sequence of IRB1600 postures during the simulation.

Figure 17.37 shows coordinates of TCP during the simulation. On the same plot are drawn the coordinates of the Probe returned by IPC from the server. Note that the probe was positioned exactly at the TCP (see Box 17.1). As can be seen the probe's broken curve closely follows TCP simulation curve, displaced along the time axis for double output delay (0.1 s).



t = 0 s

t = 4 s

t = 8 s

t = 12 s

t = 16 s

t = 20 s

Fig. 17.36 The sequence of motion of ABB IRB1600 robot during the simulation

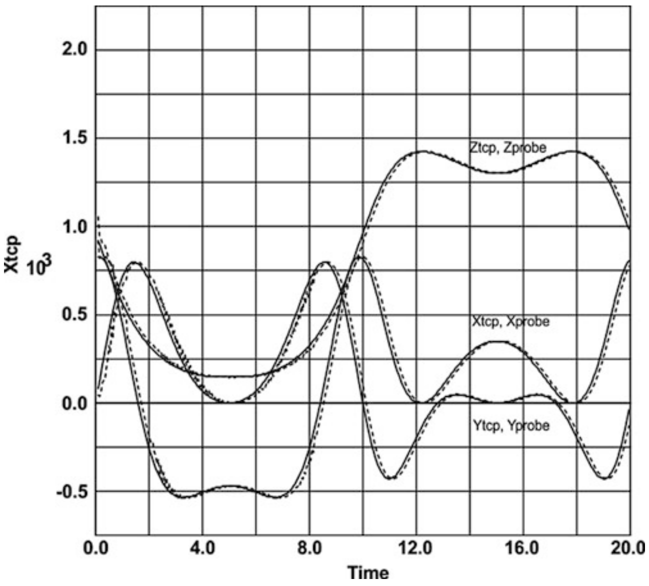


Fig. 17.37 Motion of TCP by: (**a**) Simulation (*thin line*) and (**b**) Visualization (probe, *broken line*)

Box 17.1 ABB IRB1600 Script

```
!-------- ABB IRB1600_12 -----------------------------------
! Characteristic lengths are L1 = 486.5 L2 = 150 L3 = 475
     L4 = 600 L5 = 65 mm
Device IRB1600 (euler -90.0 -90.0 0.0 lbox 1200)
     Joint 1   revolute Z
     Joint 2  (shift x 150 z 486.5) revolute y
     Joint 3  (shift z 475) revolute y
     Joint 4 (shift x 600) revolute x
     Joint 5  revolute y
     Joint 6  revolute x
     Tcp (shift x 65)
     initial (0.0 0.0 0.0 0.0 0.0 0.0);

Part IRB1600_X-120_m2004_rev0_01-1_Body0;
Part IRB1600_X-120_m2004_rev0_01-7_Body1;
Part IRB1600_X-120_m2004_rev0_01-4_Body2;
Part IRB1600_X-120_m2004_rev0_01-5_Body3;
Part IRB1600_X-120_m2004_rev0_01-3_Body4;
Part IRB1600_X-120_m2004_rev0_01-2_Body5;
Part IRB1600_X-120_m2004_rev0_01-6_Body6;

Set IRB1600 add IRB1600_X-120_m2004_rev0_01-1_Body0 ;
Render IRB1600 color 0.5 0.5 0.5;
Set IRB1600#1 add IRB1600_X-120_m2004_rev0_01-7_Body1 ;
```

(continued)

```
Render IRB1600#1 color 0.5 0.5 0.5 ;
Set IRB1600#2 add IRB1600_X-120_m2004_rev0_01-4_Body2 (shift
  x-150 z -486.5 );
Render IRB1600#2 color 0.5 0.5 0.5 ;
Set IRB1600#3 add IRB1600_X-120_m2004_rev0_01-5_Body3 (shift
  x-150 z -961.5 );
Render IRB1600#3 color 0.5 0.5 0.5 ;
Set IRB1600#4 add IRB1600_X-120_m2004_rev0_01-3_Body4 (shift
  x-750 z -961.5 );
Render IRB1600#4 color 0.5 0.5 0.5 ;
Set IRB1600#5 add IRB1600_X-120_m2004_rev0_01-2_Body5 (shift
  x-750 z -961.5 );
Render IRB1600_12#5 color 0.5 0.5 0.5 ;
Set IRB1600#6 add IRB1600_X-120_m2004_rev0_01-6_Body6 (shift
  x-750 z -961.5 );
Render IRB1600#6 color 0.5 0.5 0.5 ;
Probe Point1 IRB1600#6(shift x 65) refer IRB1600;
!------------- End -------------------------------------------
```

BondSimVisual© supports also creating *avi* files during the simulations. These files can be run by the most media players. More information can be found on BondSim© web site.

References

1. Borri, M., Trainelli, L., & Bottasso, C. L. (2000). On representation and parameterizations of motions. *Multibody Systems Dynamics, 4*, 129–193.
2. Damic, V., & Cohodar, M. (2015) Dynamic analysis and 3D visualization of multibody systems. In: Junco & Longo (Eds.), *Proceedings of the International Conference on Integrated Modeling and Analysis in Applied Control and Automation*, pp. 89–96, Bergeggi, Italy, 21–23 September 2015, ISBN 978-88-97999-63-8.
3. Damic, V., & Montgomery, J. (2016). *Mechatronics by bond graphs* (2nd ed.). Berlin Heidelberg: Springer.
4. Grasp 10. www.bygsimulations.com. Accessed 27 May 2015.
5. IRB 1600. *The highest performance 10 kg robot*. http://new.abb.com/products/robotics/industrial-robots/irb-1600. Accessed 20 May 2016.
6. Karnopp, D. C., Margolis, D. L., & Rosenberg, R. C. (2000). *System dynamics: Modeling and simulation of mechatronic systems* (3rd ed.). New York, NY: John Wiley.
7. KUKASim. http://www.kukarobotics.com/en/pressevents/productnews/NN_040630_KUKASim.htm Accessed 09 May 2015.
8. Named Pipes. https://msdn.microsoft.com/en-us/library/windows/desktop/aa365590(v=vs.85).aspx. Accessed 20 May 2015.
9. Peter Corke, R. (2011). *Vision and control: Fundamental algorithms in matlab*. Heidelberg: Springer.

10. RoboGuide. http://www.fanucamerica.com/products/vision-software/ROBOGUIDE-simulation-software.aspxccessed 09 May 2015.
11. RobotStudio. http://new.abb.com/products/robotics/robotstudio. Accessed 09 May 2015.
12. Schroeder, W., Martin, K., & Lorensen, B. (1998). *The Visualization Toolkit*, Prentice Hall PTR, Upper Saddle River, New Jersey. The version of the library used is VTK-6.3.0 of 1.9.2015.
13. Sciavicco, L., & Siciliano, B. (1996). *Modeling and control of robot manipulators*. New York, NY: McGraw-Hill.

Appendix A
Some Textbooks on Bond Graph Modelling

In the following, some textbooks on bond graph modelling are listed separately from the references at the end of each chapter for convenience [1–8]. They may introduce into the methodology, provide a survey, and serve as a knowledge source for questions regarding details of the bond graph methodology.

References

1. Borutzky, W. (2010). *Bond graph methodology – Development and analysis of multidisciplinary dynamic system models*. London: Springer. ISBN:978-1-84882-881-0 (print).
2. Brown, F. T. (2001). *Engineering system dynamics: A unified graph-centered approach*. New York, Basel: Marcel Dekker. ISBN:0-8247-0616-1.
3. Damić, V., & Montgomery, J. (2016). *Mechatronics by bond graphs* (2nd ed.). Berlin, Heidelberg: Springer.
4. Gawthrop, P., & Smith, L. (1996). *Metamodelling: Bond graphs and dynamic systems*. Hemel Hempstead: Prentice Hall International (UK) Limited. ISBN:0-13-489824-9.
5. Karnopp, D. C., Margolis, D. L., & Rosenberg, R. C. (2012). *System dynamics: Modeling, simulation, and control of mechatronic systems* (5th ed.). Hoboken, NJ: Wiley.
6. Merzouki, R., Samantaray, A., Pathak, P., & Ould Bouamama, B. (2013). *Intelligent mechatronic systems*. Berlin: Springer.
7. Mukherjee, A., Karmakar, R., & Samantaray, A. (2006). *Bond graph in modeling, simulation and fault identification*. New Delhi, India: I.K. International Publishing House. ISBN: 81-88237-96-5.
8. Samantaray, A., & Ould Bouamama, B. (2008). *Model-based process supervision – A bond graph approach*. Advances in industrial control. London: Springer.

W. Borutzky (ed.), *Bond Graphs for Modelling, Control and Fault Diagnosis of Engineering Systems*, DOI 10.1007/978-3-319-47434-2

Index

W. Borutzky (ed.), *Bond Graphs for Modelling, Control and Fault Diagnosis of Engineering Systems*, DOI 10.1007/978-3-319-47434-2