



*Université catholique
de Louvain*

Symbolic Modeling of Electromechanical Multibody Systems

Laurent Sass
Electromechanical engineer

Ph.D. Thesis

Members of the jury:

Pr. J.D. Legat (UCL), president
Pr. P. Fisette (UCL), supervisor
Pr. D. Grenier (UCL), supervisor
Pr. J.C. Samin (UCL)
Pr. F. Labrique (UCL)
Pr. J. McPhee (University of Waterloo, Ontario)
Pr. J. Swevers (KULeuven)

20 January 2004

Thanks to

*my supervisors, Paul Fisette and Damien Grenier
for their time and the fruitful advices they gave to me,*

*the members of the jury: Francis Labrique, Jean-Didier Legat,
Jean-Claude Samin, Jan Swevers and especially John McPhee for
his warm welcome during my stay at the University of Waterloo
and also for coming to Belgium for my private defense,*

my parents and my family for their trust and encouragements,

*my colleagues for the interesting professional
and non professional discussions*

*and a very special thanks to
the the one who will recognize herself
for the love she gives me and for her support.*

Abstract

In this thesis, we are interested in automatically generating models for multibody systems (MBS) involving a large number of degrees of freedom (d.o.f.), which tightly interacts with the electrical subsystems by means of electromechanical transducers. These systems will be referred to as electromechanical multibody systems.

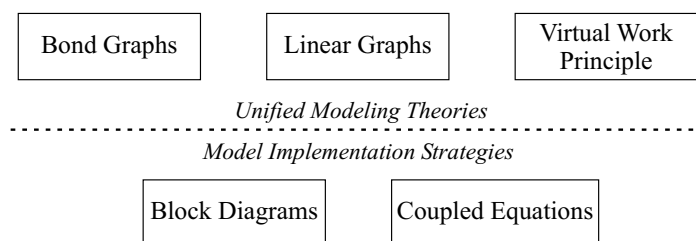


Figure 1: Modeling theories v.s. Modeling implementation strategies

An important distinction has to be done between the theories used to build models and the strategies, or techniques, used to implement multi-domain models. Apart from dedicated theories used in each domain of physics, *unified theories*, applicable to several physical domains, were developed. Graph theories, such as Bond graph and Linear graph approaches, and equational theories, such as the Virtual Work Principle, are examples of unified theories, which are presented and compared in chapter 3. Figure 1 mentions three important unified theories and the two main implementation strategies:

1. A model is constructed for each physical domain independently, and the *submodels are assembled at a numerical level*. This results, for instance, in a *block diagram representation* of the system as shown in figure 2. Co-simulation, which involves several simulation tools in the same process, is also part of this category of modeling techniques.

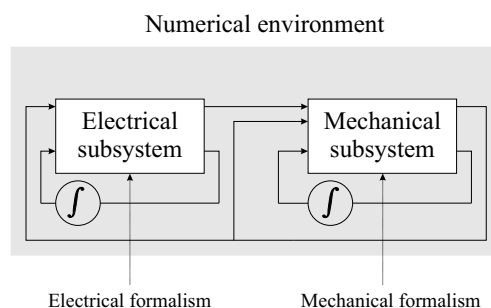


Figure 2: Block diagram assembly of submodels in a numerical environment

2. One global model, consisting of *one self-sufficient system of coupled equations*, is builded for the whole system and provided to the numerical tool. For instance, this model can be constructed using unified theories, as illustrated in figure 3. Strategies based on modeling languages also lead to one global model, but does not rely exclusively on unified theories.

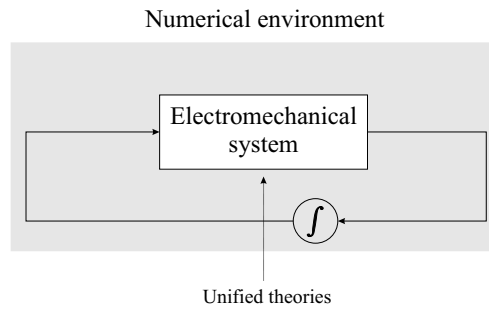


Figure 3: Unified theories provides a global model to the numerical environment

When developing control strategies for a system, block-diagrams are very often used, but for building models, the numerical problems associated with the use of block-diagrams or co-simulation, and especially the non-synchronized evaluation of the submodels, make this technique non-reliable for modeling large MBS with electromechanical interaction.

From a modeling point of view, strategies based on unified theories or modeling languages are more rigorous, but the efficiency of the models obtained with the latter can be called into question for the applications aimed in this work.

In chapter 3, we present and compare different unified theories: Bond graph, Linear graph and Virtual Work Principle. With this in depth confrontation on the basis of different “non-academic” electromechanical systems, we pioneered [60] in the world of modeling and simulation. Indeed, most of the time, researchers focus on their approach and literature about comparison of modeling theories is very poor.

The main conclusion from this comparison is that, at the present stage, none of the existing unified theories is fully satisfactory for the applications aimed in this work:

- Bond graphs are interesting for modeling 1D systems, described by scalar variables (e.g. electrical circuits, hydraulic or pneumatic systems, etc.), but they suffer from important drawbacks, when considering 3D multibody systems.
- Linear graphs offer the possibility to model multidimensional systems and are therefore effective for building electromechanical multibody models. Nevertheless, when large multibody structures are considered, it is less efficient than other approaches, such as the recursive formulations, unless the latter can be extracted from Linear graph theory.
- The Virtual Work Principle, based on an energy conservation principle, leads to direct manipulations of the equations. Several properties of the mathemat-

ical entities and the structure of the equations can be proven. For instance, it is shown in section 3.3.3 that the electromechanical interaction does not affect the structure of the electrical and mechanical equations, which can thus be generated separately. However, because of the partial derivatives and multiple summations involved in the computations, the Virtual Work Principle becomes inefficient in the present context.

Hence, neither block diagrams nor strategies based on modeling languages or unified theories are fully satisfactory in the present context. Consequently, in the second part of this thesis, a new strategy is proposed for building models of large electromechanical MBS. Despite the problems mentioned before, we propose to generate the submodels for the electrical and mechanical parts separately, but under a symbolic form. As illustrated in figure 4, the coupling of the resulting equations is achieved at a symbolic level and does not take place in the numerical environment. In this work, we chose to use dedicated formalisms, presented in chapter 4, to obtain efficient electrical and mechanical equations:

- The well-known Newton-Euler Recursive formalism is used for mechanical multibody systems, together with the coordinate partitioning method [75] for the elimination of the constraints.
- A circuit based formalism is described for electrical circuits which can involve mutual inductive and capacitive influences, and permanent magnets. Extension of the mechanical coordinate partitioning [75] is proposed for constrained electrical circuits.

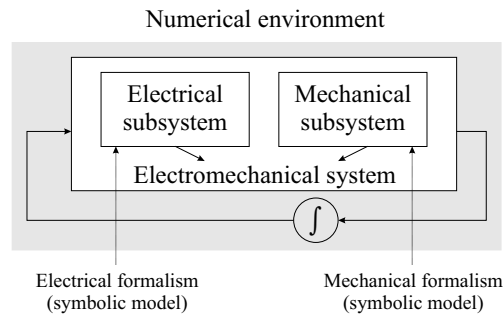


Figure 4: Symbolic submodels assembled in a global symbolic model

At the end of chapter 4, the symbolic submodels are rigorously coupled according to specific flowcharts, especially when constraints exist between the generalized variables of the system. One global model is provided to the numerical tool.

Chapter 5 deals with the symbolic implementation. Based on its own symbolic engine, Robotran [20] is a program which generates symbolic dynamic and kinematic equations for multibody systems. Its development started several years ago in our division and it has already been used in many studies in multibody dynamics. During

this research, on the basis of the same symbolic engine, we developed Electran which generates symbolic equations for electrical circuits and electromechanical converters, according to the formalism of chapter 4. As indicated in figure 5, these two softwares are used independently and the obtained submodels are coupled together into a global symbolic model, according to the flowcharts presented in chapter 4.

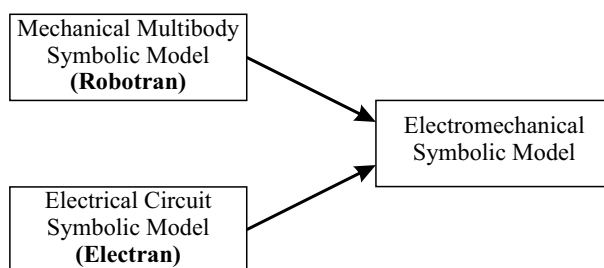


Figure 5: Two softwares are used to obtain an electromechanical model

Finally, in the third part of this thesis, we present some applications, starting with validation examples to assert the results obtained with Electran. In chapter 6, comparisons with other softwares and also with experimental measurements are shown. Afterwards, in chapter 7, two *industrial*¹ systems are considered, which are more representative of the applications aimed in this research:

1. a parking gate system (see figure 6), consisting of a flexible barrier mounted on a six-bar mechanism driven by an asynchronous three phase inductive motor, and
2. a railway bogie, with articulated chassis, driven by three phase inductive motors (see figure 7).

In both applications, electromechanical coupling is highlighted, which reinforces the interest for multidomain modeling, possible for such complex applications thanks to the use of dedicated formalisms to obtain efficient symbolic submodels, which are rigorously coupled according to the flowcharts presented in chapter 4.

¹This refers to applications encountered in and proposed by industries, in opposition to *academic* applications (e.g. double-pendulum, pendulum on a cart, etc.) classically considered in teaching and research activities.

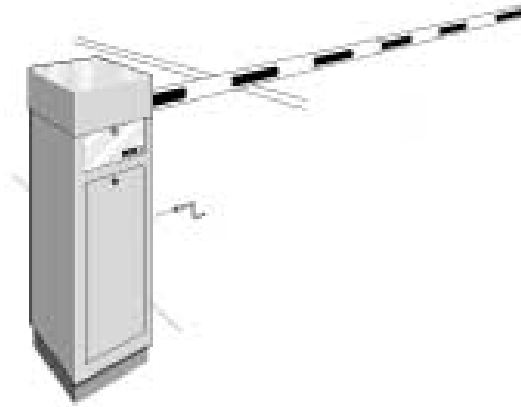


Figure 6: Parking gate

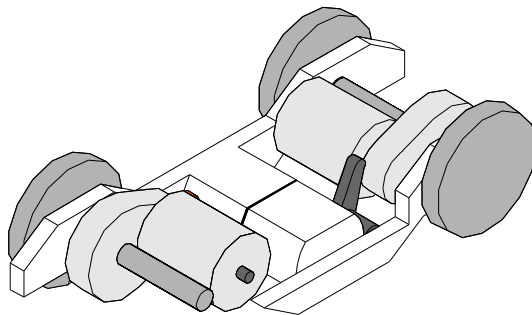


Figure 7: Railway bogie

List of symbols

Operators

\dot{a}	: first time derivative of a
\ddot{a}	: second time derivative of a
\tilde{a}	: tilde matrix associated with a (3 x 1) array a
$\tilde{\mathbf{a}}$: tilde tensor associated with vector \mathbf{a}
δa	: virtual change (variation) in a

Suffixes

m	: refers to mechanical variables
e	: refers to electrical variables
em	: refers to electromechanical variables

Indexes

u	: refers to independent variables
v	: refers to dependent variables
r	: refers to reduced equations

Mechanical Symbols

N^{body}	: number of bodies
N^{joint}	: number of joints
s^m	: mechanical generalized joint variables
$\{\hat{\mathbf{I}}\}$: inertial reference frame
$[\hat{\mathbf{I}}] = \begin{bmatrix} \hat{\mathbf{I}}_1 \\ \hat{\mathbf{I}}_2 \\ \hat{\mathbf{I}}_3 \end{bmatrix}$: array of the unit vectors $\hat{\mathbf{I}}_j$ associated with frame $\{\hat{\mathbf{I}}\}$
$\{\hat{\mathbf{X}}^i\}$: reference frame attached to body i
$[\hat{\mathbf{X}}^i] = \begin{bmatrix} \hat{\mathbf{X}}_1^i \\ \hat{\mathbf{X}}_2^i \\ \hat{\mathbf{X}}_3^i \end{bmatrix}$: array of the unit vectors $\hat{\mathbf{X}}_j^i$ associated with frame $\{\hat{\mathbf{X}}^i\}$
R^{ih}	: relative rotation matrix between frames $\{\hat{\mathbf{X}}^i\}$ and $\{\hat{\mathbf{X}}^h\}$
R^i	: absolute rotation matrix of frame $\{\hat{\mathbf{X}}^i\}$ with respect to frame $\{\hat{\mathbf{I}}\}$
Ω^i	: relative angular velocity vector of body i with respect to its parent h
ω^i	: absolute angular velocity vector of body i

O^i, O^i	: attach point of joint i on body i and its parent h , respectively
\mathbf{z}^i	: relative position vector $\overrightarrow{O^i O^i}$
\mathbf{d}^{ik}	: position vector of O^k with respect to O^i
\mathbf{d}_z^{ik}	: augmented position vector of O^k with respect to O^i
G^i	: center of mass of body i
\mathbf{d}^{ii}	: position vector of G^i with respect to O^i
\mathbf{d}_z^{ii}	: augmented position vector of G^i with respect to O^i
\mathbf{x}^i	: absolute position vector of G^i
m^i	: mass of body i
I^i	: inertia matrix of body i
\mathbf{I}^i	: inertia tensor of body i
\mathbf{g}	: vector of gravity
\mathbf{F}^i	: vector of the forces applied to body i by its parent h through joint i
\mathbf{F}_{ext}^i	: vector of the resultant of the external forces applied to body i
\mathbf{F}_{tot}^i	: vector of the resultant of all the forces applied to body i
\mathbf{L}^i	: vector of the torques applied to body i by its parent h through joint i
\mathbf{L}_{ext}^i	: vector of the resultant of the external torques applied to body i
\mathbf{L}_{tot}^i	: vector of the resultant of all the torques applied to body i
\mathbf{N}^i	: linear momentum of body i
\mathbf{H}^i	: angular momentum of body i
$\delta\boldsymbol{\pi}^i$: pseudo-vector of virtual infinitesimal rotation
M	: generalized mass matrix
Q^{tot}	: total generalized forces
Q	: generalized joint forces
Q^c	: generalized forces associated with conservative forces
Q^{nc}	: generalized forces associated with non-conservative forces
T	: kinetic energy
U	: potential energy
L^m	: mechanical Lagrangian
J^m	: Jacobian of the mechanical constraints
λ^m	: Lagrange multipliers associated with mechanical constraints

Electrical Symbols

N^{el}	: number of elements
N^{node}	: number of nodes
s^e	: electrical generalized variables
w	: electrical state variables
y	: electrical algebraic variables
z	: electrical sources values
s	: number of sources
su	: number of voltage sources
si	: number of current sources
r	: number of resistors
l	: number of inductors
c	: number of capacitors
p	: number of permanent magnets
i_k	: current flowing through element k
u_{kk}	: voltage drop across element k
i	: array of currents through the elements of a circuit
u	: array of voltage drops across the elements of a circuit
$q_k = \int_{-\infty}^t i_k dt$: charges accumulated in element k
$\varphi_k = \int_{-\infty}^t u_{kk} dt$: fluxes through element k
φ_{jk}^l	: flux through inductor j due to inductor k
φ_{jk}^p	: flux through inductor j due to permanent magnet k
$\varphi_j^p = \sum_{k=1}^p \varphi_{jk}^p$: flux through inductor j due to the permanent magnets
R_j	: resistance of resistor j
L_{jj}	: self inductance of inductor j
L_{jk}	: mutual inductance between inductors j and k
C_{jj}	: self capacitance of capacitor j
C_{jk}	: mutual capacitance between capacitors j and k
W_m^*	: magnetic co-energy
W_m	: magnetic energy
W_e^*	: electrical co-energy
W_e	: electrical energy
L^e	: electrical Lagrangian
Q^e	: generalized voltages
J^e	: Jacobian of the electrical constraints
λ^e	: Lagrange multipliers associated with electrical constraints
M^e	: electrical mass matrix

Electrical suffixes

<i>su</i>	: refers to voltage sources
<i>si</i>	: refers to current sources
<i>s</i>	: refers to sources
<i>r</i>	: refers to resistors
<i>l</i>	: refers to inductors
<i>c</i>	: refers to capacitors
<i>p</i>	: refers to permanent magnets

List of abbreviations

MBS	: MultiBody Systems
ODE	: Ordinary Differential Equations
DAE	: Differential Algebraic Equations
d.o.f.	: Degrees Of Freedom
NER	: Newton/Euler Recursive
MTF	: Modulated TransFormer
MGY	: Modulated Gyrator

Contents

Abstract	v
List of symbols	xi
List of abbreviations	xv
Table of contents	xvii
Part I General Context and Existing Modeling Approaches	1
1 Introduction	3
1.1 General context: Multidomain modeling	3
1.2 Existing multi-domain modeling techniques	8
1.2.1 Block-diagrams and coupled simulations	10
1.2.2 Strategy based on unified theories	13
1.2.3 Strategy based on modeling languages	14
1.3 Objectives	16
2 Definitions, concepts and notations	19
2.1 Multibody systems	20
2.1.1 Fundamental concepts	20
2.1.2 Topology representation	21
2.1.3 Frames and vectors	22
2.1.4 Joint modeling hypotheses	25
2.1.5 Dynamic quantities	26
2.1.6 Newton/Euler equations of motion	29
2.2 Electrical circuit	29
2.2.1 Fundamental concepts	29
2.2.2 Variables and notations	30
2.2.3 Constitutive equations and dynamic entities	31
2.2.4 Kirchoff's Equations	33

3	Unified theories confrontation	37
3.1	Bond Graph Method	38
3.1.1	Bond Graph and Electrical Systems	42
3.1.2	Bond Graph and 1D Electromechanical Systems	42
3.1.3	Bond Graph and Multibody Systems	44
3.1.4	Discussion	46
3.2	Linear Graph Method	46
3.2.1	Linear Graph and Electrical Systems	49
3.2.2	Linear Graph and 1D Electromechanical Systems	50
3.2.3	Linear Graph and Multibody Systems	52
3.2.4	Discussion	55
3.3	Virtual Work Principle	55
3.3.1	Virtual Work Formalism for Multibody Systems	56
3.3.2	Virtual Work for Electrical Systems	69
3.3.3	Virtual Work for Electromechanical Systems	83
3.4	Conclusions	86
Part II	Proposed Symbolic Modeling Approach	87
4	Formalisms for Electromechanical Systems	89
4.1	MBS - Newton-Euler Recursive Formalism	91
4.1.1	Forward kinematics	92
4.1.2	Backward dynamics	94
4.1.3	Coordinate partitioning for MBS systems	97
4.2	Electrical Circuit Theory based Formalism	99
4.2.1	Unconstrained Circuit	99
4.2.2	Constrained circuits	103
4.2.3	Electrical coordinate partitioning	106
4.2.4	Discussion	111
4.3	Electromechanical Coupling	112
4.3.1	Unconstrained electromechanical systems	112
4.3.2	Constrained electromechanical systems	114
5	Symbolic Implementation	117
5.1	Concept of symbolic generation	118
5.2	Robotran's Symbolic Engine	120
5.3	Modeling Multibody Systems with Robotran	122
5.3.1	Robotran's Conventions	122
5.3.2	Robotran's input	123
5.3.3	Robotran's output	124
5.3.4	Electromechanical interaction	127
5.4	Modeling Electrical Systems with Electran	127
5.4.1	Circuit representation	127
5.4.2	Algorithm for Generation of Kirchoff's Equations	129

5.4.3	Electran's Symbolic Implementation	132
5.4.4	Electran's Conventions	133
5.4.5	Electran's input	134
5.4.6	Electran's Global Algorithm	136
5.4.7	Electran's output	138
 Part III Applications		141
6	Validation Applications	143
6.1	Electrical Circuit Examples	145
6.1.1	Example 1 - RLC 2nd order filter	145
6.1.2	Example 2: circuit with one constraint	148
6.1.3	Example 3: triangle circuit	150
6.1.4	Example 4 - Class-E Amplifier	151
6.2	Scara Robot Driven by DC Motors	155
6.3	Three phase induction motor	157
6.3.1	Description of the system	157
6.3.2	Electrical model	159
6.3.3	Parameter identification	162
6.3.4	Comparison between simulation and experimental results . . .	163
6.4	Permanent magnets applications	168
6.4.1	Example 1: Electromagnetic speaker	168
6.4.2	Example 2: Three-phase step motor	170
7	Industrial applications	173
7.1	Parking Gate System	173
7.1.1	Description of the system	174
7.1.2	Parking Gate Model	175
7.1.3	Simulation results	178
7.2	Railway Bogie Driven by Inductive Motors	185
7.2.1	Description of the system	185
7.2.2	Bogie Model	187
7.2.3	Simulation results	195
8	Conclusions	201
A	Comparison Paper	207
B	Example 4: Electrical Circuit Benchmark	239
B.1	Benchmark Description	239
B.2	Proposed Solutions	242

C Scara Robot Benchmark	247
C.1 Benchmark Description	247
C.2 Proposed Solutions	254
D Three phase induction motor	257
D.1 Identification procedure	257
D.2 Electrical input file	261
E Electromagnetic Speaker Model	265
E.1 Electrical Parameters	265
E.2 Electrical input file	267
E.3 Electrical output file	268

Part I

General Context and Existing Modeling Approaches

Introduction

1.1 General context: Multidomain modeling

Scientists and engineers have always been interested in understanding, predicting and controlling the behavior of the systems they are studying. In this text, we consider a *system* as an entity separable from the rest of the universe by means of physical boundaries. The rest of the universe is the *environment* of the system. A system and its environment interact and exchange energy.

In order to achieve their goal, engineers construct and study a simplified and abstracted representation of a system, called the *system model*. Following this definition, the model will never be an exact representation of the system and the predicted behavior will always be approximate. Nevertheless, very precise models can be built.

As soon as computers were invented, engineers exploited the high computational potential offered by these super calculators. From the physical theories, they could obtain the equations describing the behavior of a system. Using computers they were able to numerically solve these equations in order to predict how the system will react to stimuli.

We can already distinguish two important steps in this process: first, the engineers write the equations; second, the latter are solved numerically. The first step is what we consider as the *modeling* phase, while the numerical step will be referred to as the *simulation* process. Initially, only the latter was implemented on computers while the equations were still written by hand. A wide range of solvers were developed

as independent software units allowing engineers to focus on the formulation of the equations.

In the last three decades, numerous tools have been developed to assist engineers in performing simulations, during both the modeling and the numerical process. Some of these are general purpose simulation programs such as Acsl or Matlab/Simulink. Others were developed for simulations in specific fields of engineering such as electrical circuits (e.g. Spice) or mechanical systems (Adams, Simpack, Dads, Robotran, ...). General purpose softwares allow to build models of any kind but the equations have to be formulated in agreement with the tool conventions. On the other hand, dedicated softwares use algorithms that are optimized for specific types of systems.

Engineers have recourse to models for several purposes:

- First, and this corresponds to the main use of models, they can be used for simulations of systems (e.g. time integration, steady state equilibrium, modal analysis, parametric study, etc.). When applied to non existing systems, simulation helps predicting the behavior of new concepts, or of an existing system subjected to modifications. Existing systems are also simulated to get information on unmeasurable quantities. For example, the dimensioning of mechanical parts requires knowledge of all the efforts acting on them. Time-simulation of benchmarks also plays an important role when developing modeling tools because they allow for validation of the implemented algorithms on the basis of comparisons with experimental measurements, or even with other simulation softwares.
- Secondly, when designing control algorithm, it is necessary to have a certain representation of the system to be controlled: the more accurate the model, the better the design of the controller. In this context, linearized models, inverse models, modal analysis and frequency-responses are commonly used.
- Thirdly, optimization procedures have become more and more used these last years. They allow for cost reduction and better characteristics of the optimized system compared to the original one. Because of the numerous evaluations of the cost functions, simplified representations of the systems were at first used, but nowadays the cost functions involved in the optimization process can be far more complex and may require a dynamic analysis step; for instance, making it necessary to run a full time simulation of the system.

Thanks to the constantly increasing performances of modern computers, modeling has taken on a larger place in the design process as a *pre-prototyping* stage, allowing for a cheaper and faster first analysis of a system. In parallel, for several years, the requirements for technological systems have pushed the limits of the discrete design approach, in which each physical part is designed independently from the others. In order to meet today's design requirements (precision, size reduction, low costs, etc.), an integrated design approach is necessary, making it necessary to take into account all the aspects of the systems at the same time. For the case of controlled electromechanical systems, this refers to the discipline called *mechatronics*. System modeling

followed the same evolution, going from models of specific elements in the systems - involving only one physical domain - to multi-domain modeling of the whole system. For example, in the past, when studying mechanisms, no attention was paid on the actuators, which were considered as *ideal* force or torque sources. Similarly, when analyzing electrical actuators, engineers did consider very simple mechanical load (e.g. constant inertia and viscous friction), not always representative of the actual mechanism actuated by this motor. Nowadays, the integration at the design level is spreading out over the modeling phase leading to multi-domain models, also called *multi-physics models*, a major field of research over the last years, that led to several programs in the field of mechatronic systems (20-sim¹, Dymola², Dynast³, DynaFlex⁴, etc.). Force and torque sources are replaced by actuator models and complex mechanical models take the place of simple inertial loads.

This integration at the design and modeling levels is of importance especially when the time constants of the different subsystems have the same order of magnitude, and also when considering systems with tight integration. Here are just a few examples illustrating the wide range of applications aimed by integrated approaches:

- mechatronic microsystems;
- spatial systems, for which large and light flexible structures are to be designed and controlled;
- medical surgery robots, characterized by higher precision standards, have pushed engineers to design more and more efficient controller requiring more and more accurate models;
- Active control of vehicle suspensions and braking systems is another present application in which modeling techniques are pushed as far as possible.

In this thesis, we are interested in building models of electromechanical systems with complex structure and tight interaction between electrical and mechanical parts. Our goal is to develop tools for automatically generating, in the most “compact” form, the equations describing the dynamics of these systems.

By *electromechanical systems*, we mean systems involving electrical and mechanical energies and conversion from one type to the other. We will consider systems with complex structure for which writing the equations by hand would be tedious and endless, making the use of computer programs unavoidable during the modeling process. A program generating the equations describing the dynamics of a system will be referred to as a *model generator* program.

¹<http://www.20sim.com>

²<http://www.dynasim.se>

³<http://icosym-nt.cvut.cz/dynast>

⁴<http://real.uwaterloo.ca/dynaflex>

Most of the industrial electromechanical applications may be considered as a mechanical *multibody systems* interacting with electromechanical converters. The conversion from electrical to mechanical energy is achieved by *actuators*, also called *motors*, while *sensors* transforms mechanical information into electrical signals.

Note: The *industrial* character of an application was used to refer to systems encountered in industries (e.g. cars, parallel robots, parking gate, railway bogies, etc.), in opposition with *academic* applications (e.g. double pendulum, pendulum on a cart, etc.) usually considered during teaching activities and fundamental research.

A *multibody system* (figure 1.1) is a set of bodies⁵ interconnected by *joints* (e.g. a spherical joint, a cylindrical joint, etc.) in which relative motion takes place. More concepts related to multibody systems will be defined later on (see chapter 2).

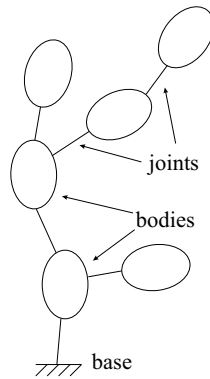


Figure 1.1: Multibody System Representation

The electromechanical converter is assumed to be represented by an equivalent circuit involving only elementary electrical dipoles (sources, resistors, inductors and capacitors). Mutual inductive and capacitive influences as well as permanent magnets will be considered. This still allows to cover most of the electromechanical converters involved in industrial applications (induction and permanent magnet motors, capacitive sensors, etc.), as can be observed in [25].

The constant evolution of computer technology has opened new perspectives for the use of models in every task where dynamic analysis is required or can bring new insights on the system.

Depending on his needs, an engineer will use different types of models, notably:

- *Continuous models* which deal with representations of the system on the basis of distributed parameters inside the matter. They consider that energy is distributed within the system (e.g. fluids, thermic models, etc.). From a mathematical point of view, the dynamic behavior of a continuous system is determined by partial

⁵This text only considers rigid bodies but multibody systems in general may involve flexible bodies.

derivative equations and boundary conditions. The finite-element based methods are very common for building continuous models in every field of engineering.

Continuous models give a prediction of the behavior of the matter but they require important computational resources and thus long simulation time.

- *Lumped parameter models* which consider the system as an assembly of components exchanging energy at specific locations called *poles*, or *ports*⁶. Components with two poles are called *dipoles*. For example, electrical resistors, sources, capacitors and inductors are electrical dipoles. The behavior of each component is then determined by algebraic and/or differential equations called *the constitutive equations*. These equations are often expressed in terms of several *parameters* characterizing the component. Besides the constitutive equations, interconnection equations expressing the energetic exchanges will be necessary to build a model.

Opposite to continuous models, lumped parameter models predict the macroscopic behavior of systems. They require fewer computations than continuous modeling and are much faster, but do not provide as much information.

Both continuous and lumped parameter models are very common, and the desired information and/or result will dictate the required level of modeling. For example, engineers interested in controlling an articulated robot will resort to a model of the involved joints and bodies in order to relate the control variables to the position of the tool: a lumped parameter model is usually sufficient since bodies can be assumed to be rigid. The same model will give information on the forces inside the joints of the robot. If interested in designing a specific mechanical part of the robot, the engineer may use a continuous model to analyze the deformations or vibrations of this part when the joint forces are applied.

In this research, we focus on lumped parameter modeling and the dynamic behavior of a system is determined by, at least, a set of ordinary differential equations⁷ (ODEs) in terms of the *dynamic variables* s and time t :

$$\dot{s} = f(s, t) \quad (1.1)$$

Note: Sometimes, the differential equations are of higher order but it is always possible to define a set of dynamic variables s to fit the generic form (1.1).

The physics of the system may impose *constraints* between the dynamic variables s and time t , that take the form of purely algebraic relations⁸

$$g(s, t) = 0, \forall t \quad (1.2)$$

⁶The concept of ports is inherited from the bond graph theory, see [35].

⁷The dynamic equations are given here in explicit form. They may also be formulated under an implicit form $f(\dot{s}, s, t) = 0$.

⁸Unilateral constraints (e.g. $g(s, t) \leq 0$) also arise in some situations, e.g. for variable configuration systems, but will not be considered in this work.

Because of these *constraints*, the dynamic variables s are not independent, and the full set of equations to be considered is a set of differential-algebraic equations (DAEs):

$$\begin{aligned} \dot{s} &= f(s, t) \\ g(s, t) &= 0 \end{aligned} \quad (1.3)$$

Numerous numerical solvers exist for ODEs as well as for DAEs. Nevertheless, for some highly non-linear applications encountered in multibody dynamics (for instance, in railway dynamics), solving DAEs may lead to numerical problems such as divergence and lack of stability [18]. The probability for such issues when using ODE solvers is much lower thanks to the existence of robust algorithms for these systems of equations. Therefore, for the applications aimed at in this work, DAEs will be avoided as far as possible, and transformed into ODEs by constraint reduction techniques, in which constraints 1.2 are rigorously solved (at *algebraic level*), analytically or by means of a Newton-Raphson type method, at each time step.

|| In this thesis, we enforced myself in always trying to obtain ODEs instead of DAEs. Where applicable, reduction procedures will be implemented in order to eliminate the constraints, once solved, and transform system 1.3 into system 1.1.

1.2 Existing multi-domain modeling techniques

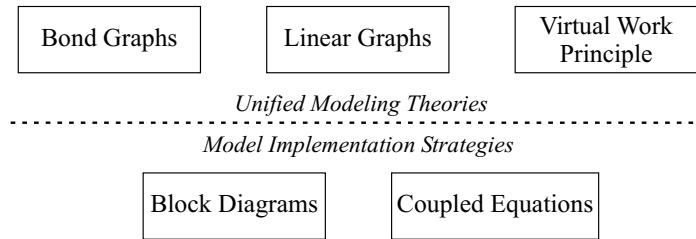


Figure 1.2: Modeling theories v.s. Modeling implementation strategies

At this stage, an important distinction has to be done between the theories used to build models and the strategies, or techniques, used to implement multi-domain models. Apart from dedicated theories used in each domain of physics, *unified theories*, applicable to several physical domains, were developed. Graph theories, such as Bond graph and Linear graph approaches, and equational theories, such as the Virtual Work Principle, are examples of unified theories, which are presented and compared in chapter 3. Figure 1.2 mentions three important unified theories and the two main implementation strategies:

1. A model is constructed for each physical domain independently, and the *sub-models are assembled at a numerical level*. This results, for instance, in a *block diagram representation* of the system as shown in figure 1.3. Co-simulation, which involves several simulation tools in the same process, is also part of this category of modeling techniques.

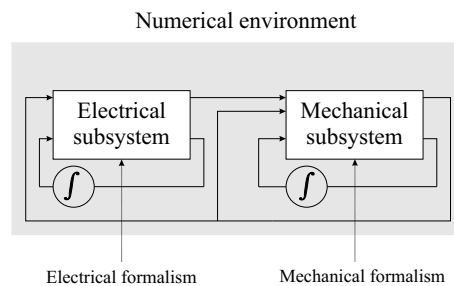


Figure 1.3: Block diagram assembly of submodels in a numerical environment

2. One global model, consisting of *one self-sufficient system of coupled equations*, is built for the whole system and provided to the numerical tool. For instance, this model can be constructed using *unified theories*, such as graph theories or the Virtual Work Principle. This is illustrated in figure 1.4. These theories are based on analogies between the different fields of physics and are used to obtain one global model of the whole system.

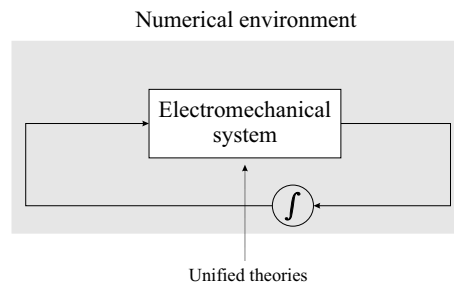


Figure 1.4: One model is provided to the numerical environment

Strategies based on *modeling languages* also lead to one global model, but does not rely exclusively on unified theories. Using general concepts of equations, objects and connections, a model is constructed for the different parts of the system. For instance, the program Dymola is based on the modeling language Modelica [72], which uses Bond graph theory, but whom multibody models are obtained with another approach.

In the following sections, in addition to the block diagram technique, strategies based on unified theories and modeling languages will be described independently because of the important differences appearing in the implementation of the corresponding tools.

1.2.1 Block-diagrams and coupled simulations

This strategy is illustrated in figure 1.3, which shows how the submodels are connected inside the numerical environment. The different submodels are usually generated using formalisms and programs optimized for their particular field of application, which leads to efficient submodels.

Block diagrams offer a good visualization of the system as the assembly of subsystems, but the simulation results *strongly depend on how the numerical program deals with their interconnection*. Algebraic loops, numerical instability and synchronization between the evaluation of the blocks are the main issues encountered with this modeling technique. Most of the time, the blocks are evaluated one after the other and their dependencies are not necessarily taken into account at the same value of time t .

For example, let us consider an electromechanical system involving an actuator driving a mechanism. Both subsystems contain differential equations and an electromechanical model is built by simply interconnecting the corresponding blocks, as shown in figure 1.5.a. The electrical machine model needs the mechanical positions and velocities, respectively denoted θ and $\frac{d\theta}{dt}$, while the mechanical system is directly influenced by the electromechanical force F_{elme} . Once the electrical block has been solved, the force F_{elme} is transmitted to the mechanical model that can be solved and the mechanical position is computed and transmitted to the electrical block. This process leads to a non-synchronized evaluation of both models and the electromechanical force F_{elme} is evaluated with the mechanical positions and velocities of the previous time. This non synchronized computation may be acceptable if the time difference is sufficiently reduced by imposing small time steps, at the costs of longer computation time; nevertheless, this is neither efficient nor rigorous.

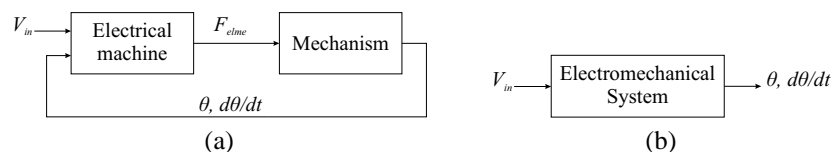


Figure 1.5: “Two-blocks” (a) and “one-block” (b) model of an electromechanical system

Even with smaller time steps, for systems with strong dynamics (e.g. highly non-linear, large frequency range, etc.), the validity of the results can be called into question. This was observed in practice with the model of a mechanism driven by an electrical motor. We compared the results obtained from a “two-blocks” model and a “one-block” model as in figure 1.5.b. In the “one-block” model, the electrical and

mechanical equations are properly coupled and evaluated together at the same instant of time. Figure 1.6 shows the rotor angular position (figure 1.6.a and a zoom in figure 1.6.b) and the electromechanical torque (figure 1.6.c and a zoom in figure 1.6.d)⁹. The obtained positions are almost identical for both models while the differences between the two torques are more important, especially at the start of the simulation, when the system behavior is more dynamic.

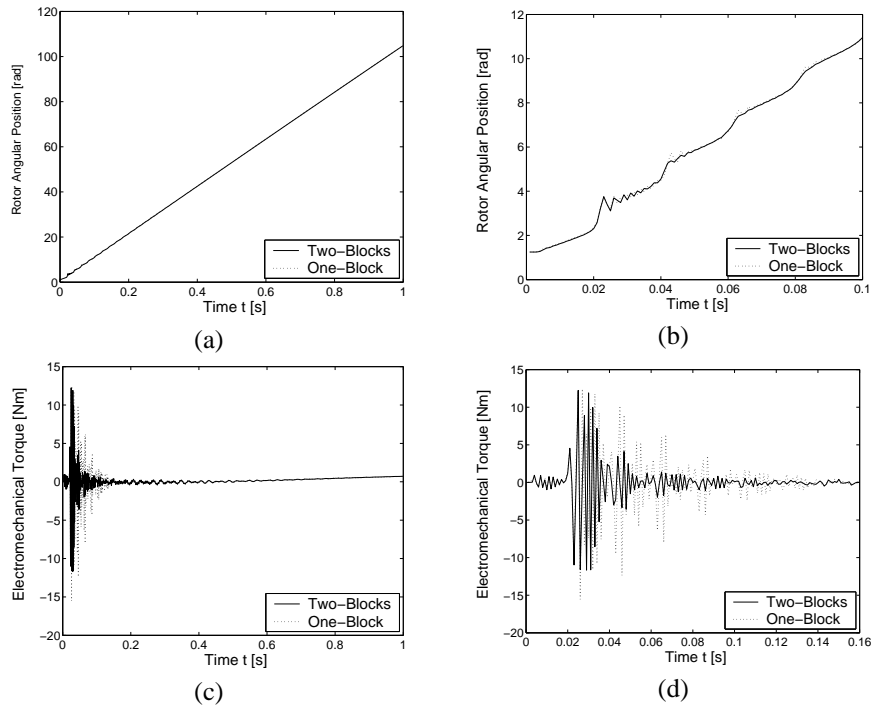


Figure 1.6: “Two-blocks” (a) and “one-block” (b) model simulation results

Looking at the numerical values in the result file (figure 1.7), we can observe that the differences are due to the non-synchronized evaluation of the electromechanical torque. The same torque is computed but applied one time step after when considering two blocks for the model¹⁰. From physical point of view, we are inclined to trust more the “one-block” model because of the synchronized evaluation of both systems of equations.

From this short discussion, it appears that, unless the interaction between the blocks is taken into account rigorously, gathering the submodels inside one global model is a better alternative to avoid numerical uncertainties about the results. We therefore recommend to avoid the use of block diagrams to build models, what does

⁹The dotted line represents the results obtained with a synchronized evaluation of both submodels.

¹⁰Both simulations used the same fixed time step integrator (Matlab’s integrator *ode5*, based on Dormand-Price integration procedure, with a 10^{-6} s time step).

not mean that block diagrams cannot be used when designing control applications, for which they are very well suited.

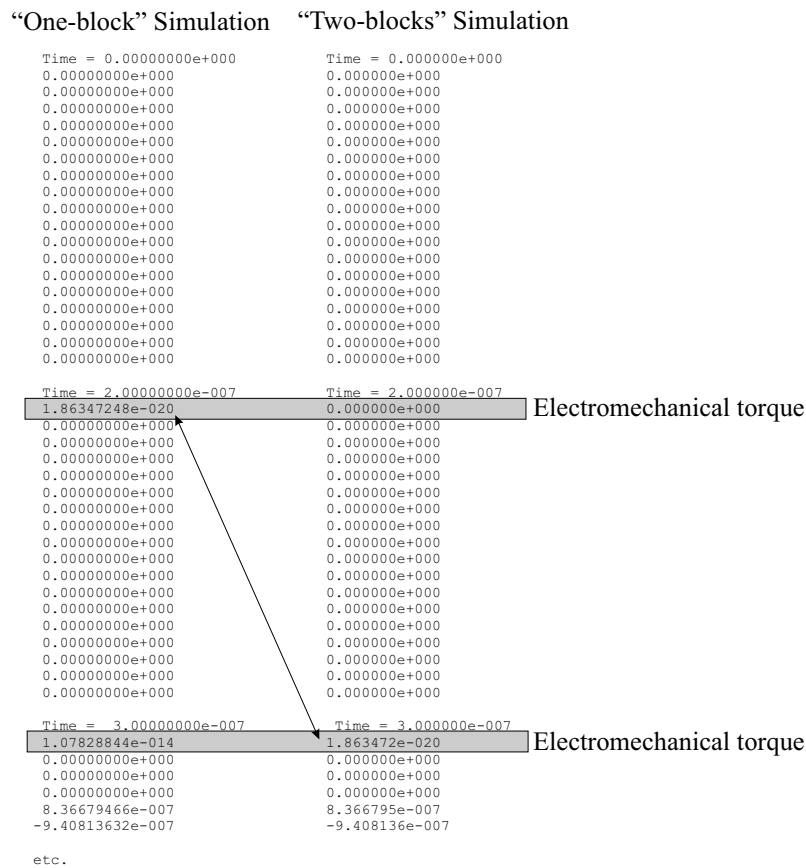


Figure 1.7: Results obtained with the “one-block” and “two-blocks” models

Coupled simulation is a similar approach that uses separate simulation programs integrated in the same numerical process. Once again, dedicated tools and formalisms may be used for each particular domain, but problems may arise during the numerical process [38].

1.2.2 Strategy based on unified theories

The strategy based on unified theories is illustrated in figure 1.4. It has the advantage that all the different parts of the system are represented in the same way and one global model can be constructed. All the equations from the different fields of physics are coupled and solved at the same time step, which protects us from some numerical problems.

There are two main types of unified theories:

- On one hand, based on electrical analogies between the physical domains, *graph theories*, also called *system theories* represent the system as an assembly of interconnected components [35, 53].
- On the other hand, *equational unified theories* directly starts from the fundamental laws describing the dynamics of the system and are based on energy conservation principles.

In [60], we confronted three of these theories: Bond graph, Linear graph and Virtual Work Principle, on the basis of simple and more complex electromechanical applications. In this work, we pioneered an in depth comparison of different modeling strategies, allowing us to have a precise view of the existing techniques. So far in literature, we could find description of different techniques but very rarely an objective and detailed confrontation as the one we conducted in [60].

Chapter 3 will expose these unified theories, among which Virtual Work Principle is an equational unified approach, compared to Bond graph and Linear graph which are based on a unified graphical representation of the different physical subsystems. Basic concepts and principles will be illustrated on a simple electromechanical system and discussions about the use of these methods in the present context are presented. Virtual Work Principle will be developed in detail, since it was the starting point of our research for an unified method able to deal with industrial applications. Moreover, it is an approach that is not very often considered as deeply as we do in this text, which was necessary in order to highlight the parallelism existing between electrical and mechanical models. Several properties of the mathematical entities involved in the equations can also be demonstrated thanks to the developments of the Virtual Work Principle.

Unified theories can be used to obtain the DAEs determining the behavior of the system. If necessary, reduction procedures can then be applied in order to transform these DAEs into ODEs, whatever the theory that was used. Nevertheless, our opinion is that, since reduction methods directly manipulates the equations, equational unified theories, such as the Virtual Work Principle, are more naturally followed by these techniques.

1.2.3 Strategy based on modeling languages

In 1978, Hilding Elmqvist proposed a new approach to model physical systems by designing and implementing the Dymola modeling language [11]. The basic idea behind Dymola was to use general equations, objects and connections to allow model developers to look at the modeling from a physical perspective instead of a mathematical one [72]. For the implementation, graph theory (based on bond graphs) was used and symbolic algorithms were introduced in order to transform the model into an appropriate form for numerical solvers. Several simulation tools based on the same approach were developed: NMF [57], ObjectMath [22], SIDOPS [9], etc., and in 1996, Hilding Elmqvist initiated an effort to unify those modeling languages into one unique modeling language called Modelica. A detailed history about Modelica can be found in [72], which has become well-developed since then.

The *causality*¹¹ assigned to an equation tells us whether the right member is affected to the left one, or reverse. Sometimes, the same set of equations can be solved with different causality assignment, leading to different results. Modelica is a modeling language based on the assumption that the behavior of any system can be represented by an acausal model, which can be used to obtain different results (e.g. direct or inverse dynamics). On the basis of Bond graph theory, Modelica combines *constitutive equations*, that dictate the behavior of each component of the model, with *conservation equations* [72]. Afterwards, causality analysis helps finding out how the equations can be used to obtain the desired results. It also allows to detect the presence of constraints. Nevertheless, constraint reduction is not considered in Modelica, so that DAE solvers are necessary.

From an algorithmic point of view, modeling languages are, most of the time, based on an object-oriented philosophy and benefit from several features such as inheritance and hierarchical modeling, and the possibility to build libraries of models. Already many libraries exist for the Modelica language helping the user in constructing multi-domain models.

Looking at the structure of a classical model using Modelica language, we find several levels of modeling going from a high one where the model is actually an interconnection of smaller units which are themselves the interconnection of additional subunits, etc. to a very basic level where the model is a set of DAEs describing the behavior of the elementary element. This is illustrated in figure 1.8 where the model of a controlled “parallel platform” was built. The complete model involves several submodels: one mechanical structure, denoted “wrist”, and four DC-motors with their controllers.

From an implementation point of view, the symbolic approach used in Modelica allows for providing a global model to the numerical solver, and also to simplify the equations. However, no reduction procedure is implemented and DAE solvers are used.

¹¹more details about this concept are given in section 3.1, when presenting the Bond graph theory.

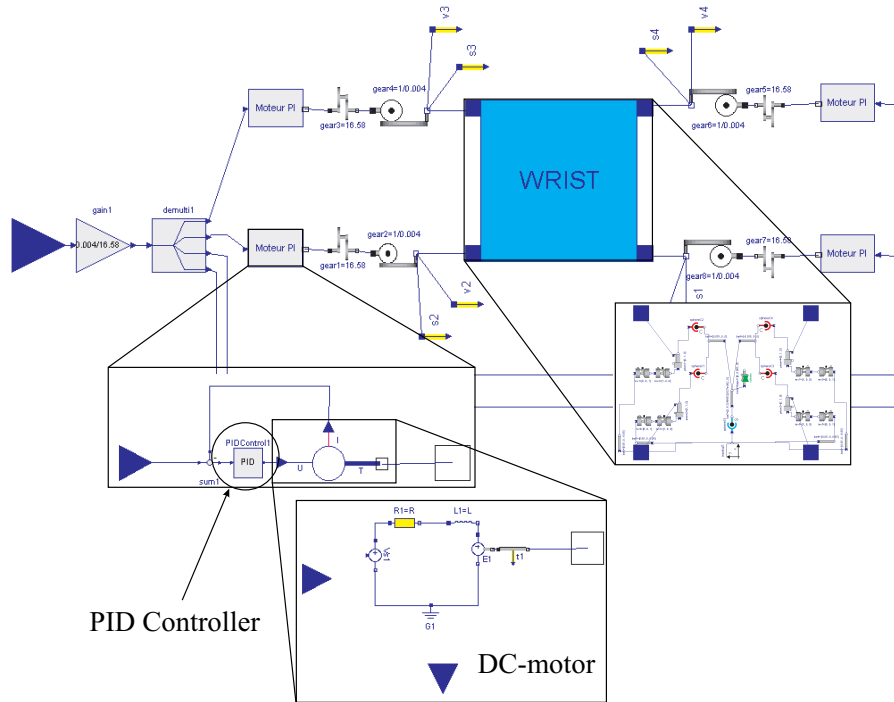


Figure 1.8: Parallel platform model using Modelica

The idea behind this modeling language is interesting but has several disadvantages:

- Since a model is assembled from several units already made up of subunits, the number of variables and equations to be considered explodes and many of these variables are equal. Indeed, at each level, new interaction variables are defined that correspond to variables of the subunits. This becomes critical when large and complex systems are considered. Fortunately, a symbolic approach is used and simplifications can be achieved but nevertheless, from our experience, the number of variables remains problematic.

In the example of figure 1.8, each DC-motor involves a connecting port through which the torque is applied to the mechanism and from which the mechanical velocity is retrieved as input to the electrical model of the motor. Associated to this port are two variables coming from another subunit, but also corresponding to variables of the complete model as well as the motor model. The same physical variable is duplicated several times throughout the model.

- From a numerical point of view, the use of DAEs solvers may lead to several numerical problems for highly nonlinear applications.

Modelica can be considered as a good base for building multidomain models. It has developed well over the last years and the provided libraries enrich its use and reduce the development work to be done. Of course, using these libraries forces us to accept several assumptions and implementation conventions decided by others, which is sometimes frustrating.

For example, while building the model of the parallel platform, we needed to retrieve some electrical variables to input them into our controller models. The controller and the DC-motor were modeled using different libraries which could not directly be interfaced and we were forced to redefine a conversion element; what requires to look deeply into the implementation of both libraries.

In this research, we are more interested in the formalisms and their implementation and we have chosen another platform to unify our models. Since *portability* was one of our objectives, we decided to give the possibility to generate models in different well-known languages such as Matlab, C, Java, etc. Modelica could be one of them.

1.3 Objectives

As already mentioned, we are interested in building models of electromechanical systems with *large* and *complex* multibody mechanical structure and tight interaction between electrical and mechanical parts. A multibody structure is considered to be *large* when it involves numerous (more than 20) degrees of freedom (d.o.f.). Our goal is to develop tools for automatically generating the equations describing the dynamic behavior of these systems.

For the reasons discussed above, block diagrams are rejected and strategies based on modeling languages will not be considered. Hence, we explored the possibility to use unified theories to represent the different parts of the system under the same formalism. In that sense, in [60], we presented and compared graph and equational approaches. This work pioneered an *in depth confrontation of different modeling strategies*, on the basis of complex electromechanical examples. Chapter 3 introduces the basic concepts of the considered theories and discuss their application on electrical circuits, unidimensional electromechanical and multibody systems.

Equational approaches and more precisely the Virtual Work Principle received, from a theoretical point of view, our full attention for several reasons:

- First, it offers an interesting framework for electrical, mechanical and electromechanical systems, especially when Lagrange equations are obtained.
- Second, constraints are rigorously taken into account by means of Lagrange multipliers, leading to reduction techniques of primal importance to avoid the use of DAE's solvers.

Note: These reduction procedures can also be applied to equations obtained with other unified theories. Nevertheless, we believe that they are more naturally applied when starting from equational approaches, such as the Virtual Work Principle, rather than from graph approaches.

- Third, the demonstration of several properties of the involved mathematical entities naturally follows the theoretical developments.

The Virtual Work Principle is exposed in detail in chapter 3, in order to highlight the original formulation that we propose for multibody mechanical and electromechanical systems conjointly:

- For mechanical systems, we start from Newton-Euler equations of rigid bodies as in [55] and [78], what differs from most of the recent approaches that start from the d'Alembert principle at a local level [76, 67, 31]. Lagrange equations for multibody systems are then derived.
- For electrical systems, our formulation is inspired from Hadwich and Pfeiffer [26] and Lagrange equations are obtained for electrical circuits.
- The electromechanical formulation is based on the definition of an electromechanical Lagrangian. This was already proposed by Hadwich and Pfeiffer [26] but they considered the electromechanical interaction by means of constraints, which is artificial and unnecessary from our point of view.

Despite these positive features associated with *theoretical aspects*, the Virtual Work Principle suffers from several disadvantages from the *implementation point of view*, as soon as complex applications are considered, since correct but inefficient models are generated. Moreover, physical insights are lost when considering electrical systems, as shown in section 3.3.2.

For all the reasons stated before and because we are interested in applications involving large and complex multibody structures and tight electromechanical interactions, none of the present techniques (block diagrams and strategies based on unified theories or modeling languages) is fully satisfactory.

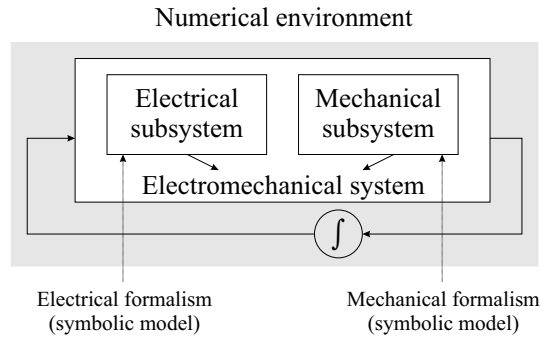


Figure 1.9: Symbolic submodels assembled in a global symbolic model

In the second part of this thesis, we propose a new strategy, illustrated in figure 1.9, for modeling large and complex multibody systems coupled with electromechanical converters. In this strategy, dedicated formalisms are used to generate the symbolic

models of the multibody and electrical subsystems. These submodels are then assembled into a global symbolic electromechanical model. The possibility to use dedicated formalisms was an advantage of the block-diagram strategy but, in the proposed approach, the assembly of the obtained submodels is achieved at a numerical level and does not take place in the numerical environment. On the contrary, we decided to provide a global model to the numerical tool.

In chapter 4, we present in details the specific formalisms that we decided to implement:

- The well-known Newton-Euler Recursive formalism is used for mechanical multibody systems. It has the advantage to provide equations identical to the Virtual Work Principle and thus benefits from the same advantages, among which the use of Lagrange multipliers technique for constrained systems, such as closed-loop systems.
- For electrical circuits, a circuit based formalism is described for circuits with mutual inductive and capacitive influences, and permanent magnets. Extension of the mechanical coordinate partitioning [75] is proposed for constrained electrical circuits.

In chapter 5, an original tree representation (obtained by opening the loops) is proposed for electrical circuits and topological concepts, inspired from those used in multibody dynamics, are defined. Symbolic implementation of the dedicated formalisms is realized. After generation of the electrical and mechanical submodels, the latter are assembled *symbolically* in a global *electromechanical model*.

In the third part of this text, we first present, in chapter 6, the validation of our implementation on the basis of simple electrical circuits and electromechanical systems. In chapter 7, we demonstrate our ability to model two industrial applications, requiring an effective coupling between electrical and mechanical equations:

1. a parking gate system, consisting of a flexible barrier mounted on a six-bar mechanism driven by a three phase asynchronous actuator, and
2. a railway bogie with three phase asynchronous motors.

In both applications, electromechanical coupling is highlighted, which reinforces the interest for multidomain modeling, possible for such complex applications thanks to the use of dedicated formalisms to obtain efficient symbolic submodels, which are rigorously coupled according to the flowcharts presented in chapter 4.

This text ends with some conclusions and prospects for future work.

Definitions, concepts and notations

Abstract

The general context of this research has been described at the beginning of this text. Let us remind that we will consider multibody systems coupled with electromechanical converters. This chapter focus on the definition of classical concepts in the fields of multibody dynamics and electrical circuit analysis. The different notations used throughout this thesis are also defined. Some symbols serve in both mechanical and electrical fields, while representing different things. So as not to confuse the reader, we tried, as far as possible, to avoid redundant symbols, at the cost of new notations replacing the traditional ones.

2.1 Multibody systems

2.1.1 Fundamental concepts

As defined before, a *multibody system* (see figure 2.1.a) is a set of bodies interconnected by joints (e.g. spherical joint, hinge joint, sliding joint, etc.). A *joint* is always assumed to be connected with two bodies at *connecting points*, also called *attach points*. At least one joint is connected with an inertial reference body called the *base* [56].

As regards the system topology, we will consider *tree-like* and *closed-loop* multibody structures, respectively illustrated in figures 2.1.a and 2.1.b. As will be explained later on, a closed-loop structure is modeled by first opening the loops and then writing geometrical constraints expressing their closure. Hence, a multibody system may always be represented by a tree with the bodies as *nodes* and the joints as *branches*, the base of the multibody system being the root of the tree. For tree-like structure, denoting N^{body} the number of bodies and N^{joint} the number of joints, we have $N^{body} = N^{joint}$. Moreover, there is only one *path* (sequence of branches) to go from the base to any other body.

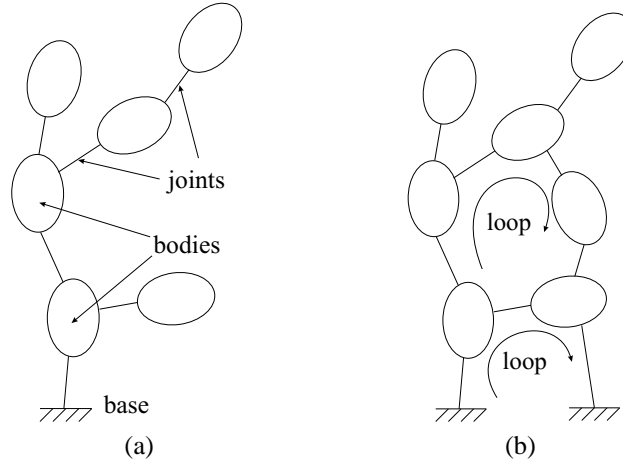


Figure 2.1: Tree-like structure (a) and closed-loop structure (b)

A *kinematic chain* is defined as an ordered collection of interconnected bodies that does not form a loop, starting at a reference body to end at a terminal body¹.

In line with the previous definition, it is possible to use some filiations concepts inherited from the family tree representation:

- *ancestor*: body i is ancestor of body j if the kinematic chain going from the root to body j (excluded) contains body i .

¹To be precise, the reference body can be any body and the terminal bodies are not necessarily “leaf” bodies.

- *direct ancestor* or *parent*: among the ancestors, the direct ancestor - or parent - of body i is the one directly connected to body i (because of the tree structure of multibody systems, there is only one parent).
- *descendant*: body i is descendant of body j if the kinematic chain going from the root to body i (excluded) contains body j .
- *direct descendant* or *child*: among the descendants, a direct descendant - or *child* - of body i is a descendant directly connected to body i (one body may have several children).
- *leaf*: a leaf is a terminal body that does not have any descendant.

These concepts can be illustrated with figure 2.2. In the latter, we can see that:

- body 2 is ancestor of bodies 3, 4, 5, 6, 7 and 8
- body 6 is parent of bodies 7 and 8
- body 3 is the child of body 2
- bodies 4, 7 and 8 are leaf bodies

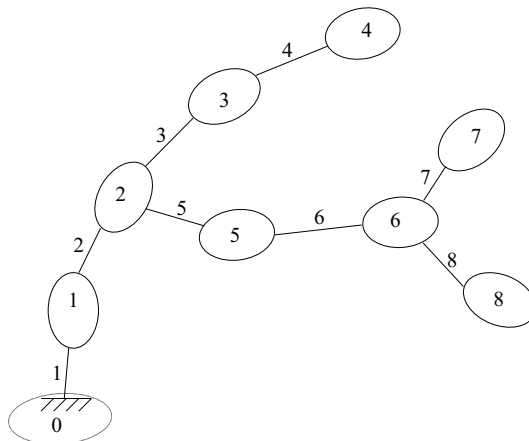


Figure 2.2: Illustration of filiation concepts

2.1.2 Topology representation

From a computational point of view, it is easier to refer to bodies and joints by means of indexes. The bodies are therefore numbered with rising numbers from the base (index 0) to the leaf bodies. The joints receive the same index as the body that follows directly in the tree structure. This is also illustrated in figure 2.2.

A *inbody* vector is used [56] to refer to the parent of each element: $inbody(i) = h$ if body h is the parent of body i . For the example of figure 2.2, $inbody = [0\ 1\ 2\ 3\ 2\ 5\ 6\ 6]$.

2.1.3 Frames and vectors

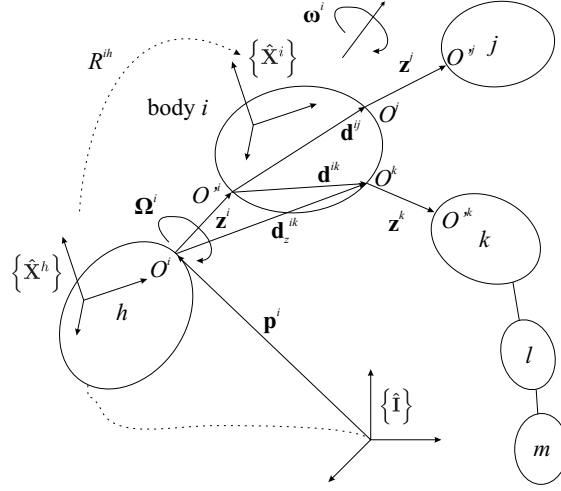


Figure 2.3: Frames and vectors definitions

Figure 2.3 shows several important frames and vectors defined below:

- O^i, O^i are the attach points of joint i on body i and its parent h , respectively;
- \mathbf{z}^i is the relative position vector $\overrightarrow{O^i O^i}$ within joint i ;
- $\mathbf{d}^{ik} \equiv \mathbf{d}^{il}$ is the position vector of the attach point O^k of joint k with respect to the attach point O^i of joint i , body k being the child of body i ;
- \mathbf{p}^i is the absolute position vector of the attach point O^i ;

It is also convenient to define the augmented position vector of joint k (figure 2.3),

$$\mathbf{d}_z^{ik} \triangleq \mathbf{z}^i + \mathbf{d}^{ik} \quad (2.1)$$

The orientation of the various bodies is described by means of the following quantities:

- $\{\hat{\mathbf{I}}\}$ is the inertial reference frame attached to the base (body 0). It is composed of three base vectors $\{\hat{\mathbf{I}}\} = \{\hat{\mathbf{I}}_1, \hat{\mathbf{I}}_2, \hat{\mathbf{I}}_3\}$,
- $\{\hat{\mathbf{X}}^i\}$ is the body-fixed frame attached to body i and located at its center of mass G^i ,
- R^{ih} is the rotation matrix such that $[\hat{\mathbf{X}}^i] = R^{ih} [\hat{\mathbf{X}}^h]$,
- R^i is the absolute rotation matrix such that $[\hat{\mathbf{X}}^i] = R^i [\hat{\mathbf{I}}]$,
- Ω^i is the relative angular velocity vector of body i with respect to its parent h .

In the previous definition, $[\hat{\mathbf{X}}^i]$ refers to a column array containing the unit vectors of the frame

$$[\hat{\mathbf{X}}^i] = \begin{bmatrix} \hat{\mathbf{X}}_1^i \\ \hat{\mathbf{X}}_2^i \\ \hat{\mathbf{X}}_3^i \end{bmatrix} \quad (2.2)$$

and a vector \mathbf{v} can be expressed in the body-fixed frame as²

$$\mathbf{v} = [\hat{\mathbf{X}}^i]^T v = [\hat{\mathbf{X}}^i]^T \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \quad (2.3)$$

$$= v_1 \hat{\mathbf{X}}_1^i + v_2 \hat{\mathbf{X}}_2^i + v_3 \hat{\mathbf{X}}_3^i \quad (2.4)$$

The absolute angular velocity vector $\boldsymbol{\omega}^i = [\hat{\mathbf{X}}^i]^T \boldsymbol{\omega}^i$, is defined on the basis of the rotation matrix R^i and the so-called *tilde matrix* $\tilde{\boldsymbol{\omega}}^i$ [56]:

$$\tilde{\boldsymbol{\omega}}^i \triangleq \begin{pmatrix} 0 & -\omega_3^i & \omega_2^i \\ \omega_3^i & 0 & -\omega_1^i \\ -\omega_2^i & \omega_1^i & 0 \end{pmatrix} \triangleq R^i \dot{R}^{iT} \quad (2.5)$$

and is of practical interest when differentiating with respect to time a position vector \mathbf{v}^P expressed in a mobile frame:

$$\mathbf{v}^P = [\hat{\mathbf{X}}^i]^T v \quad (2.6)$$

$$\dot{\mathbf{v}}^P \triangleq \frac{d\mathbf{v}^P}{dt} \quad (2.7)$$

$$= [\hat{\mathbf{I}}]^T \frac{d(R^{iT} v)}{dt} \quad (2.8)$$

$$= [\hat{\mathbf{I}}]^T (R^{iT} \dot{v} + \dot{R}^{iT} v) \quad (2.9)$$

$$= [\hat{\mathbf{X}}^i]^T \dot{v} + [\hat{\mathbf{X}}^i]^T R^i \dot{R}^{iT} v \quad (2.10)$$

One of the properties of the rotation matrix R^i is to be orthogonal, meaning that $R^i R^{iT} = E$, where E is the unit matrix. Taking the first time-derivative of this expression, one gets

$$\dot{R}^i R^{iT} + R^i \dot{R}^{iT} = 0 \quad (2.11)$$

and thus

$$R^i \dot{R}^{iT} = -\dot{R}^i R^{iT} \quad (2.12)$$

²where superscript T refers to the transposed matrix,

The latter indicates that $R^i \dot{R}^{iT}$ is antisymmetric and, there must exist a vector $\omega^i = [\hat{\mathbf{X}}^i]^T \tilde{\omega}^i$ whose tilde matrix is such that

$$\tilde{\omega}^i = R^i \dot{R}^{iT} \quad (2.13)$$

Replacing $\tilde{\omega}^i$ in 2.10, we get

$$\dot{\mathbf{v}}^P = [\hat{\mathbf{X}}^i]^T \dot{v} + [\hat{\mathbf{X}}^i]^T \tilde{\omega}^i v \quad (2.14)$$

$$= \overset{\circ}{\mathbf{v}}^P + \omega^i \times \mathbf{v}^P \quad (2.15)$$

where we used the vector cross product $\mathbf{v} \times \mathbf{w} = [\hat{\mathbf{X}}^i]^T \tilde{v}w = -[\hat{\mathbf{X}}^i]^T \tilde{w}v$.

Physically speaking, the first term of 2.15, $\overset{\circ}{\mathbf{v}}^P$, represents the *relative velocity* of point P with respect to frame $\{\hat{\mathbf{X}}^i\}$. The second term $\omega^i \times \mathbf{v}^P$ takes into account the rotation of frame $\{\hat{\mathbf{X}}^i\}$ with respect to the inertial frame $\{\hat{\mathbf{I}}\}$.

By additivity of the *relative angular velocity vectors* Ω^h , the absolute angular velocity vector ω^i of the body-fixed frame $\{\hat{\mathbf{X}}^i\}$ is equal to

$$\omega^i = \sum_{h \leq i} \Omega^h \quad (2.16)$$

where $\sum_{h \leq i}$ represents the summation over body i and its ancestors.

Note: Although the inequality sign is used here, it does not refer explicitly to an ordering in the mathematical sense: $i < j$ means that i is ancestor of j .

Taking the first time derivative, we obtain the absolute angular accelerations as

$$\dot{\omega}^i = \sum_{h \leq i} \dot{\Omega}^h = \sum_{h \leq i} \left(\overset{\circ}{\Omega}^h + \tilde{\omega}^h \cdot \Omega^h \right) \quad (2.17)$$

where $\tilde{\omega}^h \cdot \Omega^h = \omega^h \times \Omega^h$, and $\tilde{\omega}^h$ is the tilde tensor associated with ω^h .

In this study, we consider *rigid* bodies and the relative position between two material points of the same body is constant. Hence, knowing the position of one reference point Q attached to each body and the orientation of the latter is sufficient to determine the configuration of the system. The position vector of any material point P can thus be written:

$$\mathbf{v}^P = \mathbf{v}^Q + \overrightarrow{QP} \quad (2.18)$$

where \mathbf{v}^Q is the position vector of the reference point Q and \overrightarrow{QP} represents the relative position vector of P with respect to Q , see figure 2.4.

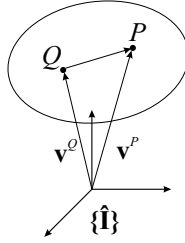


Figure 2.4: Position vectors for rigid bodies

2.1.4 Joint modeling hypotheses

The relative motions allowed by the joint define its number of *relative degree of freedom* (relative d.o.f.). As in [56] we have decided to limit the set of possible joints to two elementary cases: the *prismatic* and *revolute* single-degree-of-freedom joints, as illustrated in figure 2.5. This is not restrictive because any physical joint can be represented by a succession of elementary joints. For example, it is well-known that any rotation can always be replaced by three successive rotations about orthogonal axis. More details on this will be given in chapter 5.

One *generalized coordinate* s_i^m will be associated³ with each joint i . It represents:

- for a prismatic joint i , the *relative displacement* ξ^i of O^i with respect to O^i , measured along the unit vector \hat{e}^i (see figure 2.5.a),
- the *relative rotation angle* θ^i of body i with respect to its parent h around \hat{e}^i , if joint i is revolute.

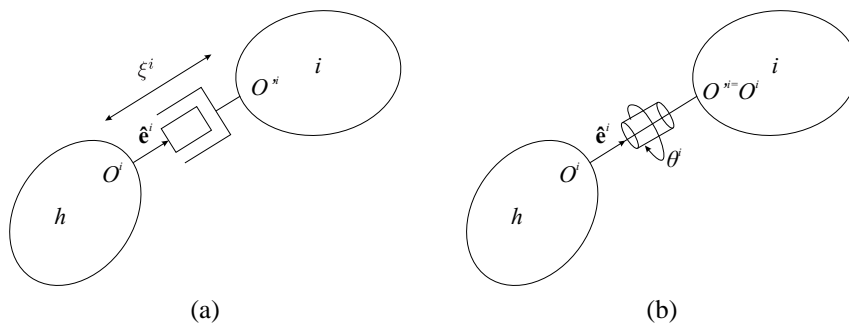


Figure 2.5: Prismatic joint (a) and revolute joint (b)

³Here, the superscript m is introduced to characterize the mechanical quantities. Later on, we will define the electrical generalized variables s^e .

For a tree-like multibody system with N^{body} bodies and joints, the system configuration can be fully determined by $N^{joint} = N^{body}$ generalized coordinates s^m representing the relative motion in the joints.

With the above definitions, the relative motion of body i with respect to its parent h can be characterized by the following joint vectors:

$$\begin{aligned} \mathbf{z}^i &= s_i^m \boldsymbol{\psi}^i \\ \boldsymbol{\Omega}^i &= \dot{s}_i^m \boldsymbol{\varphi}^i \end{aligned} \quad (2.19)$$

where vectors $\boldsymbol{\psi}^i$ and $\boldsymbol{\varphi}^i$ are defined as:

$$\begin{aligned} \boldsymbol{\psi}^i \triangleq \hat{\mathbf{e}}^i \quad \text{and} \quad \boldsymbol{\varphi}^i \triangleq 0 & \quad \text{if joint } i \text{ is prismatic} \\ \boldsymbol{\psi}^i \triangleq 0 \quad \text{and} \quad \boldsymbol{\varphi}^i \triangleq \hat{\mathbf{e}}^i & \quad \text{if joint } i \text{ is revolutive} \end{aligned} \quad (2.20)$$

The corresponding relative velocity and relative acceleration joint vectors are:

$$\begin{aligned} {}^o \mathbf{z}^i &= \dot{s}_i^m \boldsymbol{\psi}^i \\ {}^{oo} \mathbf{z}^i &= \ddot{s}_i^m \boldsymbol{\psi}^i \\ {}^o \boldsymbol{\Omega}^i &= \ddot{s}_i^m \boldsymbol{\varphi}^i \end{aligned} \quad (2.21)$$

where *generalized velocities* $\dot{s}_i^m \triangleq \frac{ds_i^m}{dt}$ and *generalized accelerations* $\ddot{s}_i^m \triangleq \frac{d^2 s_i^m}{dt^2}$ have been introduced.

2.1.5 Dynamic quantities

Here are the definitions of the main quantities, illustrated in figure 2.6, that will be necessary to characterize the dynamic behavior of the multibody system:

- m^i and $\mathbf{I}^i = [\hat{\mathbf{X}}^i]^T I^i [\hat{\mathbf{X}}^i]$ are respectively, the mass of body i and its inertia tensor with respect to the center of mass G^i .
- $\mathbf{d}^{ii} = [\hat{\mathbf{X}}^i]^T d^{ii}$ is the position vector of the center of mass G^i with respect to O^i .
- $\mathbf{x}^i = [\hat{\mathbf{I}}]^T x^i$ is the absolute position vector of the center of mass G^i .
- \mathbf{g} is the vector of gravity.
- \mathbf{F}^i and \mathbf{L}^i are respectively the internal resultant force and torque applied to body i by its parent h through the joint i . According to Newton's third law, reaction forces $-\mathbf{F}^i$ and $-\mathbf{L}^i$ are applied on body h . Similarly, in figure 2.6, body i undergoes the reactions $-\mathbf{F}^k$, $-\mathbf{L}^k$, $-\mathbf{F}^j$ and $-\mathbf{L}^j$ due to its children joints j and k .

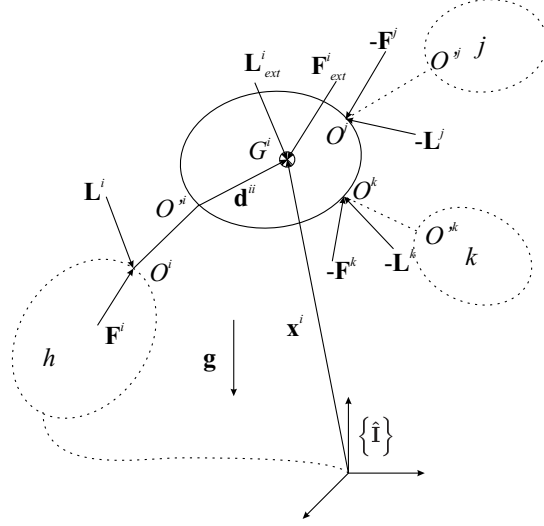


Figure 2.6: Dynamic quantities

- \mathbf{F}_{ext}^i and \mathbf{L}_{ext}^i are the resultant loads acting on body i from the multi-body system environment (excluding gravity) under the form of an equivalent resultant force \mathbf{F}_{ext}^i applied to its center of mass G^i and a resultant torque \mathbf{L}_{ext}^i with respect to the same point.

It is also convenient to define an augmented vector \mathbf{d}_z^{ii} to locate the center of mass:

$$\mathbf{d}_z^{ii} \triangleq \mathbf{z}^i + \mathbf{d}^{ii} \quad (2.22)$$

The total resultant force \mathbf{F}_{tot}^i and torque \mathbf{L}_{tot}^i applied on body i are given by:

$$\mathbf{F}_{tot}^i = \mathbf{F}^i - \sum_{h \in \bar{i}} \mathbf{F}^h + \mathbf{F}_{ext}^i + m^i \mathbf{g} \quad (2.23)$$

$$\mathbf{L}_{tot}^i = \mathbf{L}^i - \sum_{h \in \bar{i}} \left(\mathbf{L}^h + (\tilde{\mathbf{d}}^{ih} - \tilde{\mathbf{d}}^{ii}) \cdot \mathbf{F}^h \right) + \mathbf{L}_{ext}^i + \tilde{\mathbf{d}}_z^{ii} \cdot \mathbf{F}^i \quad (2.24)$$

where $\sum_{h \in \bar{i}}$ refers to all the children of body i .

According to the previous definitions, the absolute position vector of the center of mass G^i of body i can be obtained from the relative displacement vectors as

$$\mathbf{x}^i = \sum_{h \leq i} (\mathbf{z}^h + \mathbf{d}^{hi}) = \sum_{h \leq i} \mathbf{d}_z^{hi} \quad (2.25)$$

Taking the time derivatives, we find the absolute velocity vector of the center of mass G^i to be

$$\dot{\mathbf{x}}^i = \sum_{h \leq i} \dot{\mathbf{d}}_z^{hi} = \sum_{h \leq i} \left(\overset{o}{\mathbf{z}}^h + \tilde{\omega}^h \mathbf{d}_z^{hi} \right) \quad (2.26)$$

and its absolute acceleration vector as

$$\ddot{\mathbf{x}}^i = \sum_{h \leq i} \dot{\mathbf{d}}_z^{hi} = \sum_{h \leq i} \left(\overset{\omega^h}{\mathbf{z}} + 2\tilde{\omega}^h \mathbf{z}^h + \tilde{\omega}^h \mathbf{d}_z^{hi} + \tilde{\omega}^h \tilde{\omega}^h \mathbf{d}_z^{hi} \right) \quad (2.27)$$

As stated before, since bodies are rigid, knowing the position of one reference material point and the orientation of the body is sufficient for determining the position of any point of the body. If we take the center of mass G^i as reference point, the position vector of any point P on the body is

$$\mathbf{v}^P = \mathbf{x}^i + \mathbf{r}^P \quad (2.28)$$

where \mathbf{r}^P represents the relative position vector of P with respect to G^i . This vector is constant in the body fixed frame $\{\hat{\mathbf{X}}\}$ and $\mathbf{r}^P = 0$.

The velocity vector of point P is, taking into account 2.15,

$$\dot{\mathbf{v}}^P = \dot{\mathbf{x}}^i + \dot{\mathbf{r}}^P = \dot{\mathbf{x}}^i + \boldsymbol{\omega}^i \times \mathbf{r}^P = \dot{\mathbf{x}}^i + \tilde{\omega}^i \cdot \mathbf{r}^P \quad (2.29)$$

where $\tilde{\omega}^i$ is the tilde tensor associated with $\boldsymbol{\omega}^i$.

The *linear momentum* \mathbf{N}^i of body i is defined as:

$$\mathbf{N}^i \triangleq m^i \dot{\mathbf{x}}^i \quad (2.30)$$

The *angular momentum* of body i with respect to its center of mass is defined as:

$$\begin{aligned} \mathbf{H}^i &\triangleq \int_i (\mathbf{r} \times \dot{\mathbf{r}}) dm \\ &= \int_i (\mathbf{r} \times (\boldsymbol{\omega}^i \times \mathbf{r})) dm \\ &= - \int_i (\mathbf{r} \times (\mathbf{r} \times \boldsymbol{\omega}^i)) dm \\ &= [\hat{\mathbf{X}}^i]^T \left(- \int_i \tilde{r} \tilde{r} dm \right) \boldsymbol{\omega}^i \\ \mathbf{H}^i &= [\hat{\mathbf{X}}^i]^T I^i \boldsymbol{\omega}^i = \mathbf{I}^i \cdot \boldsymbol{\omega}^i \end{aligned} \quad (2.31)$$

where $I^i \triangleq - \int_i \tilde{r} \tilde{r} dm$ is the *inertia matrix* with respect to G^i .

In the above expressions, dm denotes the mass characterizing the material points constituting body i . Accordingly, we obtain the total mass of body i by summing these masses over the whole body [56]:

$$m^i = \int_i dm \quad (2.32)$$

2.1.6 Newton/Euler equations of motion

According to Newton's second law, the motion of the center of mass of body i subjected to forces \mathbf{F}_{tot}^i whatever their origins is given by

$$m^i \ddot{\mathbf{x}}^i = \mathbf{F}_{tot}^i \quad (2.33)$$

where we have used the definition of the linear momentum 2.30.

The equivalent to Newton's second law for the rotation motion of a body is the well-known Euler equation, that writes:

$$\dot{\mathbf{H}}^i = \mathbf{L}_{tot}^i \quad (2.34)$$

Using the definition of the angular momentum 2.31, equation 2.34 transforms into,

$$\mathbf{I}^i \cdot \dot{\boldsymbol{\omega}}^i + \tilde{\boldsymbol{\omega}}^i \cdot \mathbf{I}^i \cdot \boldsymbol{\omega}^i = \mathbf{L}_{tot}^i \quad (2.35)$$

2.2 Electrical circuit

2.2.1 Fundamental concepts

As already mentioned in the introduction, we limit our study to electrical devices for which an equivalent circuit can be defined to represent its behavior. The equivalent circuit contains elementary components (or elements): su voltage and si current sources respectively denoted Su and Si , r resistors, c capacitors and l inductors respectively denoted R , C and L . Mutual inductive or capacitive influences will also be considered in this text, as well as permanent magnets. This still covers a wide range of electromechanical converters [25]: induction, electrostatic and permanent magnet motors, as well as capacitive or inductive sensors.

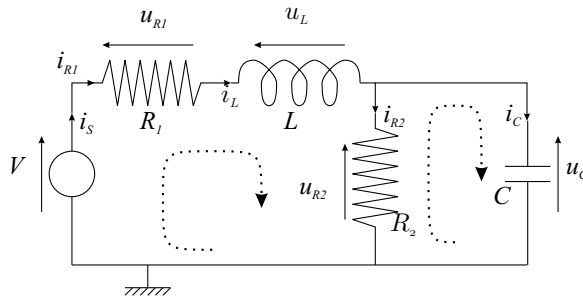


Figure 2.7: Classical representation of electrical circuits

By nature, an electrical circuit is a closed system that is classically represented, as in figure 2.7, by a set of oriented *edges*, corresponding to the elements of the circuit, interconnected at different *nodes*. The orientation of each edge refers to a positive

sign convention assigned to the current flowing through the element from the *starting* node to the *end* node.

Let us consider an electrical circuit with $(N^{node} + 1)$ nodes⁴ and N^{el} elements, with $N^{el} = su + si + r + c + l$.

2.2.2 Variables and notations

The description of an electrical circuit involves many quantities that are defined here:

- i_k is the current flowing through element k ;
- u_k is the absolute potential (with respect to the reference node) at node k ;
- u_{kj} is the voltage drop between node k and j : $u_{kj} = u_k - u_j$;
- u_{kk} is the voltage drop across element k : $u_{kk} = u_k - u_h$, where we assumed that nodes k and h are the *end* and *starting* nodes of element k , respectively;
- q_k is defined as⁵ $q_k \triangleq \int_{-\infty}^t i_k dt$. For a capacitor, it corresponds to the electrical charge accumulated inside the capacitor;
- φ_k is defined as⁶ $\varphi_k \triangleq \int_{-\infty}^t u_{kk} dt$. For an inductor, it corresponds to the magnetic flux passing through the inductor;

In particular, for inductors, we define

- φ_{jk} as the flux flowing through inductor j due to inductor k , when mutual inductive effects exist;
- φ_{jk}^p as the flux flowing through inductor j due permanent magnet k ; $\varphi_j^p = \sum_{k=1}^p \varphi_{jk}^p$ then represents the total flux passing through inductor j , due to the p permanent magnets;

With the previous definitions, the total flux passing through inductor j is

$$\varphi_j = \sum_{k=1}^l \varphi_{jk} + \sum_{k=1}^p \varphi_{jk}^p = \sum_{k=1}^l \varphi_{jk} + \varphi_j^p \quad (2.36)$$

The above electrical variables can be reorganized by sets of the same kind according to the element type. Thus, we can define:

- i^{su} , i^{si} and i^s as the arrays ($su \times 1$, $si \times 1$ and $s \times 1$, respectively) of currents through the voltage sources, the current sources and all the sources ($i^s = \{i^{su}, i^{si}\}$), respectively;

⁴ N^{node} represents the number of nodes excluding the reference one.

⁵assuming that $q_k = 0$ at time $t = -\infty$

⁶assuming that $\varphi_k = 0$ at time $t = -\infty$

- i^r as the array ($r \times 1$) of currents through the r resistors;
- i^l as the array ($l \times 1$) of currents through the l inductors;
- i^c as the array ($c \times 1$) of currents through the c capacitors;
- u^{su} , u^{si} and u^s as the arrays ($su \times 1$, $si \times 1$ and $s \times 1$, respectively) of voltage drops across the su voltage sources, the si current sources and all the sources ($u^s = \{u^{su}, u^{si}\}$), respectively;
- u^r as the array ($r \times 1$) of voltage drops across the r resistors;
- u^l as the array ($l \times 1$) of voltage drops across the l inductors;
- u^c as the array ($c \times 1$) of voltage drops across the c capacitors;
- $i = \{i^s, i^r, i^l, i^c\}$ as the array ($N^{el} \times 1$) of currents flowing through the N^{el} elements of the circuit;
- $u = \{u^s, u^r, u^l, u^c\}$ as the array ($N^{el} \times 1$) of voltage drops across the N^{el} elements of the circuit;

Similarly, for charges and fluxes, the following arrays can be defined:

$$q = \{q^s, q^r, q^l, q^c\} \text{ and } \varphi = \{\varphi^s, \varphi^r, \varphi^l, \varphi^c\}$$

with $q^s = \{q^{su}, q^{si}\}$ and $\varphi^s = \{\varphi^{su}, \varphi^{si}\}$.

Knowing the evolution of the currents i_k through each element and the voltage drop u_{kk} across each element is sufficient to describe the behavior of the circuit.

2.2.3 Constitutive equations and dynamic entities

Experiments allowed physicians to describe the behavior of electrical components with the following constitutive equations:

- | | | | |
|--------------------------|---|--|---|
| - for resistor j | : | $f(u_{jj}^r, i_j^r) = 0$ | the voltage drop across resistor j is directly related to the current flowing through it; |
| - for inductor j | : | $f(\varphi_j^l, i^l, \varphi_j^p) = 0$ | the flux flowing through inductor j is influenced by the currents i^l through all the inductors and the flux φ_j^p produced by the permanent magnets; |
| - for capacitor j | : | $f(q_j^c, u^c) = 0$ | the charge accumulated inside capacitor j is influenced by the voltage drops u^c across all capacitors; |
| - for voltage source j | : | $f(u_j^{su}, t) = 0$ | |
| - for current source j | : | $f(i_j^{si}, t) = 0$ | |

Assuming linear constitutive equations, we write the constitutive equations as:

- for resistor j : $u_{jj}^r = R_j i_j^r$,
- for inductor j : $\varphi_j^l = \sum_{k=1}^l L_{jk} i_k^l + \varphi_j^p$
- for capacitor j : $q_j^c = \sum_{k=1}^c C_{jk} u_{kk}^c$

The parameters associated with the different electrical elements considered here are defined as follows:

- R_j is the resistance associated with resistor j
- L_{jj} is the self inductance associated with inductor j
- $L_{jk} = L_{kj}$ is the mutual inductance associated with inductors j and k
- C_{jj} is the self capacitance associated with capacitor j
- $C_{jk} = C_{kj}$ is the mutual capacitance associated with capacitors j and k

We can then summarize the linear constitutive equations in matrix form as:

- for resistor j :

$$u^r = R i^r \quad (2.37)$$

- for inductor j :

$$\varphi^l = L i^l + \varphi^p \quad \text{or} \quad u^l = \frac{d}{dt} (L i^l + \varphi^p) \quad (2.38)$$

- for capacitor j :

$$q^c = C u^c \quad \text{or} \quad i^c = \frac{d}{dt} (C u^c) \quad (2.39)$$

where φ^p is the array of the total fluxes flowing through the inductors:

$$\varphi^p = (\varphi_1^p, \varphi_2^p, \dots, \varphi_l^p)^T$$

The following matrices were introduced in the matrix form of the constitutive equations 2.37, 2.38 and 2.39:

- $R = \begin{pmatrix} \ddots & 0 & 0 \\ 0 & R_j & 0 \\ 0 & 0 & \ddots \end{pmatrix}$ is the diagonal matrix of resistance
- $L = \begin{pmatrix} \ddots & \dots & L_{jk} \\ \vdots & L_{jj} & \vdots \\ L_{kj} & \dots & \ddots \end{pmatrix}$ is the symmetric matrix of inductance
- $C = \begin{pmatrix} \ddots & \dots & C_{jk} \\ \vdots & C_{jj} & \vdots \\ C_{kj} & \dots & \ddots \end{pmatrix}$ is the symmetric matrix of capacitance

In this research, we will restrict our formalism and applications to electrical elements with linear constitutive equations, although most of the developed theories apply also when non-linearities are taken into account (e.g. Linear Graph, Bond Graph, Virtual Work Principle). Linear elements lead to particular structure of the equations, allowing for analytical constraints reduction⁷. Moreover, we do not impose the matrices R , L and C and vector φ^p to be constant. In fact, we will allow these parameters to be function of mechanical or electrical variables, what is the case with electromechanical converters, which have parameters that vary with respect to the mechanical configuration.

2.2.4 Kirchoff's Equations

Kirchoff's laws are well-known equations relating the voltage drops u_{kk} and currents i_{kk} associated with different elements in the circuit:

- Kirchoff's current law states that for each node j of the circuit, the oriented sum of the currents through the elements connected to node j must be equal to zero:

$$\sum_{k=1}^{N^{el}} d_{jk} i_k = 0 \quad \text{for } j = 1 \dots (N^{node} + 1) \quad (2.40)$$

where

$$\begin{cases} d_{jk} = 0 & \text{if element } k \text{ is not connected to node } j \\ d_{jk} = 1 & \text{if element } k \text{ is connected and oriented towards node } j \\ d_{jk} = -1 & \text{if element } k \text{ is connected and oriented away from node } j \end{cases}$$

This is illustrated in figure 2.8.

- Kirchoff's voltage law states that for each loop j , a direction of circulation along the loop can be chosen and the oriented sum of the voltage drops across the elements involved in the loop must be zero:

$$\sum_{k=1}^{N^{el}} d_{jk} v_k = 0 \quad \text{for } j = 1 \dots N^{loop} \quad (2.41)$$

where

$$\begin{cases} d_{jk} = 0 & \text{if element } k \text{ is not involved in loop } j \\ d_{jk} = 1 & \text{if element } k \text{ is involved and oriented with loop } j \\ d_{jk} = -1 & \text{if element } k \text{ is involved and oriented opposite to loop } j \end{cases}$$

This is illustrated in figure 2.9.

⁷When non-linear elements are involved, constraint reduction requires to solve nonlinear equations, what is not always possible analytically.

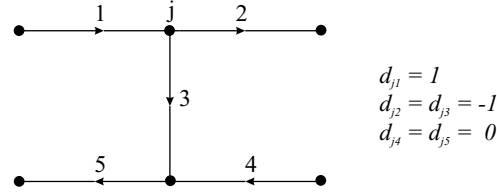


Figure 2.8: Kirchoff's current law

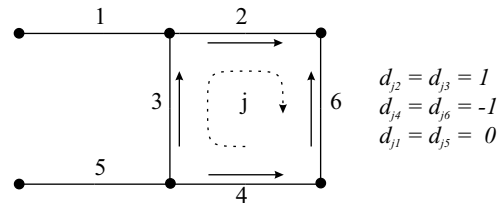


Figure 2.9: Kirchoff's voltage law

Kirchoff's theory also states that only N^{node} equations of type 2.40 and $N^{el} - N^{node}$ equations of type 2.41 are independent and the latter correspond to independent loops, which are such that their closures satisfy the closure of all the loops in the circuit. Consequently, a set of N^{el} independent algebraic relations can be obtained from Kirchoff's laws, which, together with the N^{el} constitutive equations described in section 2.2, makes a set of $2N^{el}$ equations sufficient to determine the dynamics of the circuit. Indeed, $2N^{el}$ variables, the currents and voltage drops associated with the N^{el} elements, are to be known in order to determine the full state of the circuit.

Kirchoff's current laws 2.40 can be time integrated to obtain relations in terms of charges:

$$\sum_{k=1}^{N^{el}} d_{jk} q_k = q_0 \quad (2.42)$$

where q_0 corresponds to the integration constant that can be obtained from initial conditions.

Similarly, Kirchoff's voltage laws 2.41 can be time integrated to obtain relations in terms of fluxes:

$$\sum_{k=1}^n d_{jk} \varphi_k = \varphi_0 \quad (2.43)$$

where φ_0 corresponds to the integration constant that can be obtained from initial conditions.

The initial conditions are usually equal to zero except if q_0 charges are initially stored inside the capacitors or if there are permanent magnets generating flux φ_0 .

Example For the RLC filter of figure 2.7, we have ($N^{el} = 5$) elements and ($N^{node} + 1 = 4$) nodes. Five independent Kirchoff laws, ($N^{node} = 3$) current laws and ($N^{el} - N^{node} = 2$) voltage laws, can be written:

$$\begin{aligned}i_S &= i_{R1} \\i_{R1} &= i_L \\i_L &= i_{R2} + i_C \\V &= u_{R1} + u_L + u_{R2} \\u_{R2} &= u_C\end{aligned}\tag{2.44}$$

Unified theories confrontation

Abstract

This chapter will present the three main unified theories : Bond graphs, Linear graphs and Virtual Work Principle. On the basis of a simple electromechanical systems, basic concepts are introduced and it is shown how these theories can be used to model simple unidimensional (1D) electromechanical systems, which can be described by scalar variables. Their use for multibody applications is also discussed. The Virtual Work Principle is presented in more detail since it was the starting point of our research for a unified theory, and parallelism between electrical and mechanical systems is drawn. This comparison is extracted from [60], in which we confronted unified theories on the basis of simple and more complex applications. A copy of this paper is provided in Appendix A and will be referred to during this section. With this in depth confrontation of existing theories to deal with complex applications, we pioneered in the field of modeling, in which most of the researchers focus on their approach. This sort of comparison has a growing interest in the scientific community [45].

Despite the interesting features of these unified theories, they suffer from several disadvantages making them either unapplicable (e.g. Bond graph theory) or inefficient, although effective, when dealing with systems involving large and complex multibody structures (Linear Graph and Virtual Work theories). These disadvantages are pointed out and discussed throughout this chapter.

The most common unified approaches, that is Bond Graph, Linear Graph and Virtual Work Principle, will be illustrated and compared on the basis of a common example: the condensator speaker shown in figure 3.1, excerpted from [26]. It consists of a capacitor C_2 connected in series with an inductor L_3 , a resistor R_1 and a voltage source E_4 . The upper plate of the capacitor, of mass m_5 , is allowed to move vertically and is connected to the roof by a spring-damper suspension k_6 and d_6 .

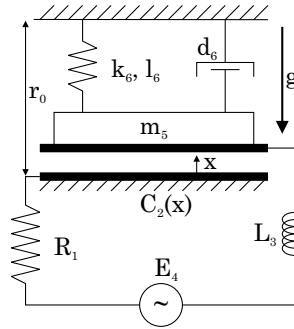


Figure 3.1: Schematic of the condensator speaker

3.1 Bond Graph Method

Bond graphs were invented by H.M. Paynter in 1959 and published by its inventor in 1961 [48]. One of the first books presenting this theory was written by Karnopp and Rosenberg in 1968. It has been republished in the year 2000 [35], from which the basic concepts presented here are extracted.

Bond graphs consider that exchanges of energy between a system and its environment only occur through interacting *ports*. For characterizing the power exchanges at the different ports, scalar *power variables* are defined at each port: the *effort* variable e and the *flow* variable f , the product of which is the power flowing through the port. Two other energy variables are also considered in bond graphs: the *momentum* $p = \int e dt$ and the *displacement* $q = \int f dt$.

Based on analogies between different physical domains, a bond graph is obtained by the interconnection of *junctions* and *elements*, by means of *bonds* [35]:

- energy *storage* elements (capacitor C and inertia I), energy *dissipative* elements (resistor R) and energy *sources* (flow source SF and effort source SE) are generic elements that can be found in every physical domain. Each of them is characterized by a constitutive equation relating the power variables and/or their derivatives.

Table 3.1 indicates the analogies existing between the different fields of physics. For electrical systems, currents are flow variables and voltages are effort variables. In mechanics, velocities are flow variables and forces are effort variables.

	Power variables		Storage Elements		Dissipation Elements
	Effort Variable	Flow Variable	Capacitor	Inertia	
Mechanics	Force	Velocity	Spring	Mass (inertia)	Damper
Electricity	Voltage drop	Current	Capacitor	Inductor	Resistor
Hydraulics	Pressure	Flow Rate	Capacitance	Inertance	Resistance
Thermics	Temperature	Heat Flow	Capacitance		Resistance

Table 3.1: Analogy used in bond graph theory

This is sometimes referred to as the “force-effort” analogy. Less popular is the “force-flow” analogy where forces are flow variables and velocities are effort variables.

- junctions are used to interconnect the elements according to the topology of the system: *0-junction* and *1-junction* connect elements having the same effort and flow variables, respectively. For instance, parallel and serial connections of electrical dipoles in a circuit are represented by 0-junction and 1-junction, respectively.

Figure 3.2 shows the bond graph that can be drawn for the condensator speaker of figure 3.1.

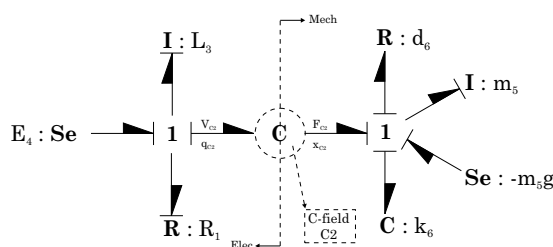


Figure 3.2: Bond graph of the electrostatic microphone

Let us already point out that the bond graph does not bear much resemblance to the physical system. Bond graphs actually indicate the structure through which energy is exchanged. Although this can be of interest to visualize the power flows, it is more intuitive to have a direct visual correspondence between the graph and the system.

The elements and junctions used to construct bond graphs are represented in Table 3.2. Besides the *one-port* elements (sources, resistors, capacitors and inductors), transducers such as transformers (*TF*) and gyrators (*GY*) are used for converting the variables from one energy domain to another. *C*- and *I*-fields give an extension of the *C* and *I* elements and are very useful for modeling more complex systems, like multidimensional mechanical systems or electromechanical converters. *C*-fields are multiport elements characterized by the following constitutive matrix equation: $e = e(q) = K q$ (in the linear case), where K is a square symmetric matrix. Similarly, the constitutive equation of an *I*-field is: $f = f(p) = K p$ (in the linear case). Mixed










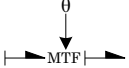
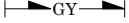

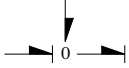
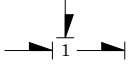
Basic 1-Port Elements		
Flow Source (SF)		$e = e(t)$
Effort Source (SE)		$f = f(t)$
Resistor (R)		$e = \phi_R(f)$
		$f = \phi_R^{-1}(e)$
Capacitor (C)		$q = \phi_C(e)$
		$e = \phi_C^{-1}(q)$
Inertia (I)		$f = \phi_I^{-1}(p)$
		$p = \phi_I(f)$
Basic 2-Port Elements		
Transformer (TF)		$e_1 = m \cdot e_2$ $f_1 = m \cdot f_2$
Modulated transformer (MTF)		$e_1 = m(\theta) \cdot e_2$ $f_1 = m(\theta) \cdot f_2$
Gyrator (GY)		$e_1 = r \cdot f_2$ $e_2 = r \cdot f_1$
Modulated gyrator (MGY)		$e_1 = r(\theta) \cdot f_2$ $e_2 = r(\theta) \cdot f_1$
Junctions		
0-junction		$e_1 = e_2 = e_3$ $f_1 = f_2 - f_3$
1-junction		$f_1 = f_2 = f_3$ $e_1 = e_2 - e_3$

Table 3.2: Basic bond graph elements and junctions with their constitutive equations

IC-fields are also used. With the exception of sources and resistors, all elements are power conservative.

In figure 3.2, a C-field element is used to model the capacitor. From an electrical point of view, it is a classical capacitor. From a mechanical point of view, it can be considered as an additional constitutive force element, just like a spring or a damper.

A sign convention for positive power flows is indicated on the bond graph by a half arrow, as shown in table 3.2, which gives a list of all the elements, and the associated constitutive equations, considered in bond graph theory. A positive power is given to the port pointed at by the bond. This convention is established by the modeler for each system. Usually, one tries to predict positive power flows from the energy sources to the loads. Thus, the arrow is directed away from sources, that deliver power, and towards R-elements, that dissipate energy.

Topological equations can be deduced from the junctions constitutive equations: for all the components connected to a 0-junction (1-junction), the efforts (flows) are equal and the flows (efforts) must sum to zero, taking into account the direction assigned to the bonds (see table 3.2. For electrical circuits, these equations correspond to Kirchoff's laws.

Solving equations always requires to define which of the right and left terms affects the other. The process for deciding in which direction each equation has to be solved is the *causality assignment*. In bond graph, causality is indicated by a bar, called the *causal stroke* crossing one end of each bond: the effort variable is imposed to the corresponding element [35]. Thus, the causal stroke for an effort (flow) source Se (Sf) is on the right (left) of the bond in table 3.2. The causal stroke indicated in table 3.2 for I-element implies either an *integral causality* or a *derivative causality* whether the effort is, respectively, the input or the output of the element. Indeed, when the effort is the input to an I-element, solving the constitutive equation $e = \phi_I \left(\frac{df}{dt}, f \right)$ requires an integration, while the other causality assignment lead to a derivation. Similarly, derivative or integral causality can be defined for C-elements [35].

Bond graphs are by nature acausal, which means that the structure of the graph is not influenced by the causality assignment. Hence, the same graph may be used to get different sets of equations depending on the needs (direct or inverse models, for example). Moreover, causality analysis helps detecting algebraic loops and constraints between state-variables. Indeed, by choosing integral causality for a bond graph, the governing equations take the form of ODE's (ordinary differential equations) expressed in terms of the primary variables, generally chosen as the flows and efforts associated with I- and C-elements, respectively. If derivative causality has to be assigned, it highlights topological particularities in the model [35], leading to algebraic relations between the primary variables.

After assigning sign conventions and causality to the graph, topological equations from the junctions and constitutive equations of the elements are combined and form a necessary and sufficient set of equations to determine the behavior of the modeled system.

3.1.1 Bond Graph and Electrical Systems

Electrical circuits or networks being unidimensional¹ (1D), it is straightforward to model them using bond graphs [35]. The main disadvantage is, once again, that bond graphs, have a structure different from that of the circuit, making the graph hard to understand.

Another challenge facing Bond Graph theoreticians is the presence of constraints between state-variables. Usually, the state variables are chosen as the flow (the current) of I-elements (inductor) and the effort (the voltage) of C-elements. When cutsets of inductors or loops of capacitors can be found in a circuit, constraints exist and the state variables are not independent. Causality analysis on the graph will indicate this and will allow for selecting a minimal set of independent variables [35]. Another solution consists in using some transformations to reduce the set of state variables to a set of independent state variables. For example, the Park transformation [39] can be used for modeling a star connected three-phase induction motor, where the sum of phase currents must equal zero, by an equivalent two-phase one, where all the phase currents are independent.

3.1.2 Bond Graph and 1D Electromechanical Systems

Bond graphs consider the same generic elements for every physical domain. This makes easy the modeling of multidomain systems, including electromechanical ones. The electromechanical interaction is generally modeled using the transformer (TF) and gyrator (GY) elements, as can be seen in appendix A with the DC-motor bond graph. When the electromechanical interactions occur within energy storage elements, C-, I- or mixed IC-fields may also be used, as shown in the condensator speaker example (figure 3.1) detailed below.

Example The complete bond graph for the electrostatic microphone described previously is given in Figure 3.2. A C-field is used as a result of the dependance of the capacitor value on the mechanical position: $C_2 = C_2(x)$. The constitutive equations for this C-field are determined as follows.

The energy in the C-field is given by, according to the sign convention shown in figure 3.2:

$$E(q_C, x) = \int (\dot{q}_C \cdot u_C - \dot{x} \cdot F_C) dt = \int u_C \cdot dq_C - \int F_C \cdot dx \quad (3.1)$$

It follows that the constitutive equations for the C-field are related to the energy by:

$$u_C = \frac{\partial E(q_C, x)}{\partial q_C} \quad \text{and} \quad F_C = -\frac{\partial E(q_C, x)}{\partial x} \quad (3.2)$$

¹Unidimensional systems can be described by scalar variables and are encountered in hydraulics and pneumatics, as well.

Experiments with capacitances tell us that the following relations can be established in the electrical field:

$$u_C = \frac{q_C}{C_2(x)} \Leftrightarrow i_C = C_2(x) \frac{du_C}{dt} \quad (3.3)$$

Assuming that at a constant x , we charge the capacitor from 0 to $q_C C b$, the energy stored in the capacitance will be:

$$E(q_C, x) = \int_0^{q_C} \frac{q_C}{C_2(x)} dq_C = \frac{1}{2} \frac{q_C^2}{C_2(x)} \quad (3.4)$$

The expression of the electrostatic force acting between the two plates is then obtained as:

$$F_C = \frac{1}{2} \frac{q_C^2}{C_2(x)^2} \frac{dC_2}{dx} = \frac{1}{2} u_C^2 \frac{dC_2}{dx} \quad (3.5)$$

Equations 3.3 and 3.5 are the constitutive equations of the C-field shown in Figure 3.2.

Deriving the dynamic equations for the system is easily done. We start by first writing down the 1-junctions equations:

$$\begin{aligned} \text{for the effort variables} & : \begin{cases} e_{E4} - e_{R1} - e_{L3} - e_{C2}^{Elec} = 0 \\ -e_{C2}^{Mech} + e_{d6} + e_{m5} + e_{k6} - e_g = 0 \end{cases} \\ \text{for the flow variables} & : \begin{cases} f_{E4} = f_{R1} = f_{L3} = f_{C2}^{Elec} = i_L \\ f_{C2}^{Mech} = f_{d6} = f_{m5} = f_{k6} = f_g = \dot{x} \end{cases} \end{aligned}$$

Using the constitutive equations of the different elements, we can derive the dynamic equations:

$$\begin{aligned} m_5 \ddot{x} &= \frac{1}{2} \frac{q_C^2}{C_2(x)^2} \frac{dC_2}{dx} - k_6 (r_0 + x - l_6) - d_6 \dot{x} - m_5 g \\ L_3 \frac{di_L}{dt} &= E_4(t) - R_1 \cdot i_L - u_C \\ i_L &= i_C \Leftrightarrow i_L = C_2 \frac{du_C}{dt} \end{aligned}$$

that reduces to

$$\begin{aligned} \frac{d}{dt} (m_5 \dot{x}) + k_6 x + d_6 \dot{x} - \frac{1}{2} \frac{dC_2(x)}{dx} (E_4(t) - R_1 i_L - \frac{d}{dt} (L_3 i_L))^2 &= -m_5 g \\ i_L = \frac{dC_2}{dx} \frac{dx}{dt} (-L_3 \frac{di_L}{dt} - R_1 i_L + E_4(t)) + C_2(x) \frac{d}{dt} (-L_3 \frac{di_L}{dt} - R_1 i_L + E_4(t)) & \quad (3.6) \end{aligned}$$

3.1.3 Bond Graph and Multibody Systems

A well-furnished history on the use of bond graphs in mechanics and multibody systems can be found in Wilfrid Favre's thesis [12]. Let us point out some of the most interesting facts.

Karnopp and Rosenberg [32, 35, 52] first proposed an analytical approach based on the following procedure: after writing of the kinematic laws by hand, the corresponding junction structure is drawn (using MTF's, 0- and 1-junctions, defined in table 3.2) and one-port elements are added. The *orthogonality principle* [35] then guarantees that the dynamic equations can be derived from the obtained graph. Like all bond graph techniques for multibody systems, this method uses *absolute velocities as power variables* and is based on *scalar variables*. The obtained bond graph does not bear much resemblance to the multidimensional mechanical system, as can be seen in figure 3.3 from [60] (see Appendix A), which shows the bond graph of a 2-link robot.

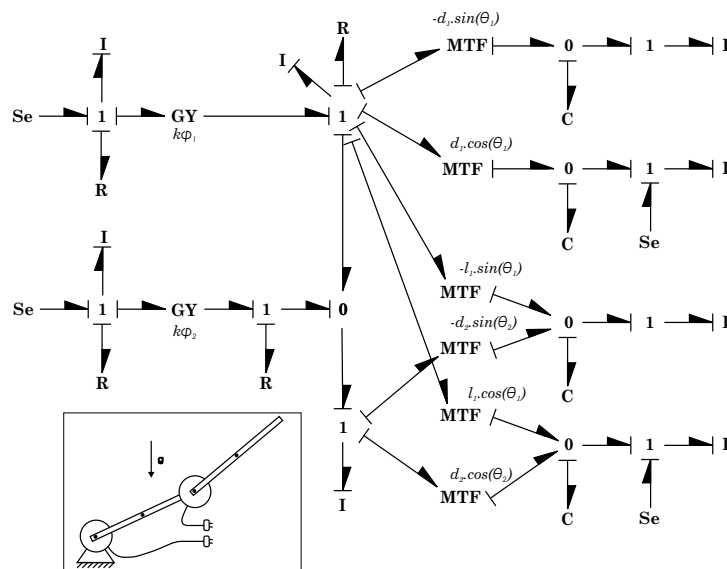


Figure 3.3: Bond graph of the 2-link robot manipulator

Kinematic constraints are common in multibody dynamics and lead to non-linear DAEs. They are indicated by the presence of *derivative causality*. In order to avoid these non-linear algebraic relations, Karnopp introduces compliant elements (springs and dampers) in the connections, at the cost of bringing high frequencies into the system [33]. This is illustrated in figure 3.3, where C-elements (mechanical springs) have been introduced in the model to avoid derivative causalities of the I-elements (masses and inertias). Other methods based on the Lagrange equations were developed by Karnopp to eliminate the constraints [34]. Actually, the constraints are included man-

ually in the equations before drawing the corresponding graph. Allen and Dubowsky [1] or Brown [10] developed some methods based on analytical computations but they still manipulate scalar variables, while dealing with multidimensional systems.

Different propositions [4, 28] have been made to represent multidimensional systems using vector bond graphs or *multibond graphs*. Breedveld established a standard representation for multibond graphs [6, 7]. Tiernego and Bos [5] use multibond graph to analyze open-loop multibody systems and, by applying well-known velocity transformations from absolute speeds to joint speeds, they are able to generate a minimal set of ODE's. Favre [13] has proposed an extension of Tiernego's work to closed-loop systems, leading to the diamond-shape bond graph of a body. His approach was used in [60] for the modeling of a swinging flexible barrier. The resulting Bond graph is shown in figure 3.4 and illustrates the complexity reached when Bond graph theory is applied to multibody dynamics.

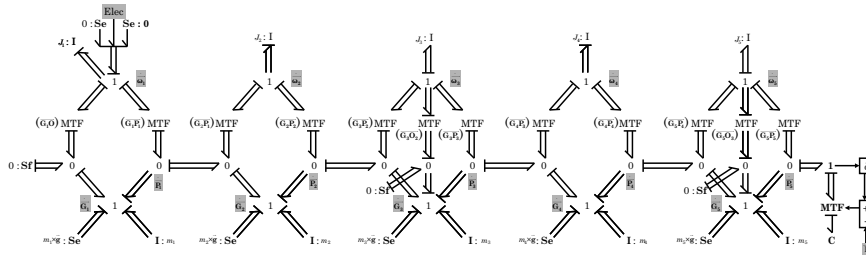


Figure 3.4: Bond graph of the mechanism driving a flexible barrier

Multibonds being arrays of three scalar quantities, they simplify the graph, which is still far from the physical intuitive structure of the system. Nevertheless, they are not frame independent tensors, which we use in classical mechanics, and the reference frame transformations are explicitly introduced in the graph by means of modulated transformers (MTF). This can be observed in figure 3.4.

Unfortunately, causality analysis of multibond graphs is not an easy task. Favre [12] has proposed an approach in which he comes back to a scalar bond graph after drawing the vector bond graph. Other approaches generalizing the Karnopp one-dimensional approaches (compliant elements, use of IC-fields for Lagrange equations,...) may also be developed for avoiding derivative causality but are not systematic. Transformation of the dependent I elements through transformers is also proposed in [5]. Some work for defining bond graph blocks describing basic multibody components has been done and leads to a more systematic definition of the bond graph [79].

As mentioned, all these approaches deal with absolute velocities and the constraints are thus written at velocity level. They have to be time integrated to be satisfied at position level, what forces the modeler to be particularly careful when choosing the initial guess (closed configuration) and also during the integration process, to avoid any drift of the constraints at position level, which is a very delicate problem in multibody dynamics.

3.1.4 Discussion

The previous developments on bond graphs aims at showing that they are well-suited to 1D multidomain systems, including electrical and electromechanical converters, but not to multidimensional applications and especially not to multibody systems. Several disadvantages have been pointed out: the use of absolute generalized velocities, the use of frame dependent vectors, the generation of constraints at velocity level, the causality assignment procedure hard to implement with multibond graphs, etc. The same issues are commented by McPhee in [45].

The current state of bond graphs is best summarized by Karnopp et al. [35]: “It is true that low-order, linear systems can be simulated with virtually no effort from the user. But complex nonlinear systems do require significant user input”.

This explains why we decided to reject bond graph theory in our context.

3.2 Linear Graph Method

Linear graph theory is a branch of mathematics devoted to the study of systems topology. It was invented by Leonhard Euler in the 1700s to study problems of connectivity [2], and was extended in the 1900s [36] to the modeling of physical systems. In this extension, linear graph theory is combined with the characteristics of physical components to obtain a unified systems theory; the term *graph-theoretic modeling* (GTM) is often used to denote this systems theory. In a nutshell, a system model is obtained by combining topological relationships from linear graph theory with the constitutive equations for individual components.

To model a physical system, individual components are identified and their constitutive equations are determined. In general, these constitutive relationships are obtained from experimental measurements of the component’s *through* and *across variables*; *through variables* are measured by an instrument in series with the component, while *across variables* are obtained from an instrument in parallel. Note that through and across variables may be tensors of any order, including scalars or vectors.

Once the constitutive equations are determined, the component models are combined in the topology defined by the structure of the physical system. A linear graph, consisting of lines (edges or branches) and circles (nodes or vertices), is used to represent the system topology, see figure 3.5. The edges represent the individual components, whereas nodes represent the points of their interconnection. From this graph, linear topological equations are systematically obtained in terms of the through and across variables for all components. The system model is simply the combination of these topological equations with the individual constitutive equations.

Similarly to what is done with bond graphs, scalar power variables can be defined as across and through variables and analogies can be found between the different fields of physics, as shown in Table 3.3. This allows to model simple unidimensional multidomain systems as done in [53]. “Force-flow” analogy results from the definition of the across and through variables that are respectively velocities and forces. Let us point out that unlike with bond graphs, scalar and vector positions are valid across

	Power variables		Storage Elements		Dissipation Elements
	Across Variable	Through Variable	A-type	T-type	
Mechanics	Velocity	Force	Mass (inertia)	Spring	Damper
Electricity	Voltage drop	Current	Capacitor	Inductor	Resistor
Hydraulics	Pressure	Flow Rate	Capacitance	Inertance	Resistance
Thermics	Temperature	Heat Flow	Capacitance		Resistance

Table 3.3: Analogy used in linear graph theory

variables in linear graph theory, without being power variables.

To illustrate these concepts, refer to figure 3.5 for the linear graph representation of the condensator speaker shown in figure 3.1. Edges R_1 , C_2 , L_3 and E_4 represent the resistor, capacitor, inductor, and voltage source, respectively. Note that, unlike the bond graph of figure 3.2, the linear graph bears a striking resemblance to the physical system, which is an advantage when it comes to modeling using this approach. Directions are assigned to each edge to establish a positive convention for measuring the through and across variables, similar to setting a polarity on a measuring instrument. The constitutive equations for electrical components are expressed in terms of the scalar variables, current i and voltage u . For the purpose of this example, we assume standard linear relationships for these components, e.g. $u_1 = R_1 i_1$ and $u_3 = L_3 \frac{di_3}{dt}$.

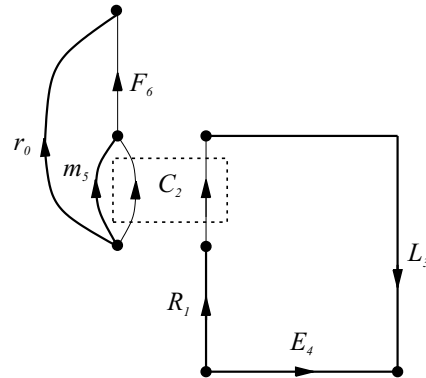


Figure 3.5: Linear Graph of the condensator speaker

Also shown in Figure 3.5 is the linear graph of the mechanical part of the condensator speaker, which only allows relative translation in a vertical direction. The mechanical subsystem is thus unidimensional and scalar variables are sufficient to characterize it, although vector quantities could be considered, as in [60] (see Appendix A). The edge m_5 represents the inertia and weight of the moving mass; the edge begins at a ground-fixed (inertial) reference node and terminates at the center of mass. Its constitutive equation is given by the combination of gravity with the Newton's Second Law: $F_5 = -m_5 \ddot{x}_5 - m_5 g$, where the force F_5 depends on gravity g and \ddot{x}_5 , the vertical acceleration of the mass m_5 . The edge F_6 represents the combined effects of the spring and damper components (these could easily be split

into separate edges for the spring and damper, if desired). Its constitutive equation is $F_6 = -k_6(r_6 - l_6) - d_6 v_6$, where l_6 is the undeformed spring length, k_6 and d_6 are the stiffness and damping coefficient, and v_6 is the relative velocity of the endpoints. Finally, the edge r_0 locates the point where the spring-damper is attached to the roof: $r_0 = l_0$.

For each of the two parts of the linear graph, mechanical and electrical, we can generate sets of topological equations that relate the through and across variables. They are based on the Vertex and Circuit postulates, also called compatibility and continuity laws in [53]:

- The Vertex Postulate states that for each node of the graph, the oriented sum of the through variables associated with the edges connected to the node must be zero.
- The Circuit Postulate states that for each closed loop (*circuit*) of the graph, the oriented sum of the across variables associated with the edges involved in the loop must be zero.

These two postulates correspond to the Kirchoff's laws (see equations 2.40 and 2.41) for electrical circuits. For mechanical systems, the Vertex Postulate gives equations for dynamic equilibrium.

The topological equations can be obtained manually by inspection of the graph, or by applying matrix operations to an *incidence matrix* that encapsulates the topology of the physical system. For a linear graph with e edges and v vertices, entry I_{jk} of the $e \times v$ incidence matrix I is $[0, +1, -1]$ if edge k is [not incident upon, incident and away from, incident and towards] the vertex j .

The Vertex Postulate [36] then allows us to write:

$$I \tau = 0 \quad (3.7)$$

where τ is a column matrix of all the through variables. Starting from the Vertex Postulate, two very useful sets of topological equations, the *cutset* and *circuit* equations, can be systematically derived by selecting a tree and applying elementary matrix operations to I . For electrical networks, the circuit equations correspond to Kirchoff's Voltage law around a closed circuit, while the cutset equations are linear combinations of the vertex equations for all the nodes in a given subgraph.

A *tree* is a set of $v - 1$ edges ("branches") that connects all of the nodes but does not contain any closed loop. The *cotree* is the set of edges that are not in the tree. A very attractive feature of linear graph theory is that *by selecting a tree, one can control the primary variables appearing in the final system*: they are the across variables α_b for branch elements, and the through variables τ_c for cotree elements (*chords*). This is accomplished by re-writing the cutset equations as the *chord transformations*

$$\tau_b = -A_c \tau_c \quad (3.8)$$

where τ_b are the branch through variables and A_c is obtained from elementary row operations on I , and by re-writing the circuit equations as the *branch transformations*

$$\alpha_c = -B_b \alpha_b \quad (3.9)$$

where α_c are the cotree across variables. The *Principle of Orthogonality* [36], which represents a generalized energy conservation principle, guarantees that $B_b = -A_c^T$.

Note: One cutset equation is associated with each branch b . It is obtained by separating the graph into two parts with a cut involving only branch b and the chords. Similarly, one circuit equations is associated with each chord c and is obtained by constructing the circuit involving only chord c and branches.

By selecting edges $R_1, L_3,$ and E_4 into the tree for the electrical sub-graph in figure 3.5, one gets the chord transformations:

$$\begin{pmatrix} i_1 \\ i_3 \\ i_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} i_2 \quad (3.10)$$

and the single branch transformation:

$$u_2 = - \begin{pmatrix} 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_3 \\ u_4 \end{pmatrix} \quad (3.11)$$

Assuming that there is one constitutive equation for each of the v elements, substituting the branch and chord transformations into these constitutive equations will result in v system equations in terms of the v primary variables. This procedure will be illustrated later on with the condensator speaker model.

3.2.1 Linear Graph and Electrical Systems

As mentioned previously, linear graphs are very well suited to the modeling of 1D physical systems and this of course concerns electrical circuits. The basic principles for modeling electrical networks using linear graph theory were established decades ago by Koenig et al [36], Roe [51], and others. Once the constitutive equations for each element are supplemented with the cutset and circuit equations, one has a necessary and sufficient set of v equations to obtain the v primary variables.

However, it is possible to reduce the equations to a smaller set by exploiting the nature of the constitutive equations. One approach is to generate one equation for each capacitor and inductor, and to use the remaining constitutive equations and branch/chord transformations to express all other variables in terms of the capacitor voltages and inductor currents. This approach was successfully implemented by Muegge [47]. For the electrical portion of the linear graph shown in Figure 3.5, one would get two first-order ordinary differential equations (ODEs) in terms of u_2 and i_3 .

Another approach is to express all variables in terms of the currents associated with chords, or the voltages associated with branches. The former is called the current formulation, while the latter is named the voltage formulation; both were implemented by

Scherrer and McPhee [63]. By selecting the tree appropriately, one can significantly reduce the final number of system equations.

For the example shown in Figure 3.5 with the capacitor C_2 selected into the cotree, the current formulation will give a single second-order ODE in terms of the corresponding current i_2 .

This is accomplished by substituting the chord transformations 3.10 into the constitutive equations for the branches, giving:

$$\begin{aligned} u_1 &= R_1 i_2 \\ u_3 &= L_3 \frac{di_2}{dt} \\ u_4 &= E_4(t) \end{aligned}$$

where $E_4(t)$ is the prescribed voltage source. Substituting these constitutive equations into the branch transformation 3.11 gives:

$$u_2 = -R_1 i_2 - L_3 \frac{di_2}{dt} + E_4(t) \quad (3.12)$$

which expresses the capacitor voltage in terms of its current. This equation will be combined with the constitutive equation for the moving-plate capacitor, defined in the next section, to obtain the final system equation.

3.2.2 Linear Graph and 1D Electromechanical Systems

An electromechanical system consists of electrical networks and mechanical systems that are coupled by electromechanical transducers, represented in figure 3.5 by a dotted rectangle. The moving-plate capacitor shown in Figure 3.1 is an example of such a transducer. The electrical characteristics of this capacitor depend upon the distance x between the plates:

$$i_2 = \frac{d(C_2(x)u_2)}{dt} = C_2(x) \frac{du_2}{dt} + \frac{dC_2(x)}{dx} \frac{dx}{dt} u_2 \quad (3.13)$$

where the capacitance C_2 is a function of x , and the second term is the motion-induced current in the component. Due to the electrical attraction of the plates, a force arises that depends upon the voltage across the capacitor and the relative distance x :

$$F_2 = \frac{1}{2} \frac{dC_2(x)}{dx} u_2^2 \quad (3.14)$$

which was already used before in equation 3.5.

Thus, the moving-plate capacitor is characterized by two constitutive equations, one associated with the electrical domain and the other with the mechanical domain. This is a common characteristic of transducer elements, which transform electrical energy into mechanical energy, and vice-versa.

Since the capacitor affects the physics of both domains, there is an edge for the capacitor in the linear graphs for the electrical and mechanical subsystems shown in

Figure 3.5. Equation 3.13 is associated with the electrical edge, while the mechanical edge is characterized by equation 3.14. It is through these constitutive equations that the two domains are coupled. Nevertheless, let us point out that both graphs remain separate, what may be disturbing when considering complex multi-domain systems.

Example As described previously, selecting the capacitor into the cotree of the linear graph and using the current formulation gives the capacitor voltage shown in equation 3.12. Substituting this equation into the electrical constitutive equation 3.13, the single ODE for the electrical domain is obtained:

$$\begin{aligned} \frac{dC_2}{dx} \frac{dx}{dt} \left(-R_1 i_2 - L_3 \frac{di_2}{dt} + E_4(t) \right) \\ + C_2(x) \frac{d}{dt} \left(-R_1 i_2 - L_3 \frac{di_2}{dt} + E_4(t) \right) - i_2 = 0 \end{aligned} \quad (3.15)$$

where the primary variables are the cotree current i_2 and the mechanical displacement x .

For the mechanical domain (see Figure 3.5), the body-fixed vector r_0 and mass m_5 are selected into the tree, resulting in the single branch coordinate $r_5 = x$. This is an independent coordinate for the 1-dof system, so no constraint equations are generated (there are no joints in the cotree). The single dynamic equation results from the cutset equation for the mass:

$$F_5 + F_2 - F_6 = 0 \quad (3.16)$$

Substituting the mechanical constitutive equations into this expression, and rearranging,

$$-m_5 \ddot{x} - m_5 g + \frac{1}{2} \frac{dC_2(x)}{dx} u_2^2 + d_6 \dot{x}_6 + k_6 (x_6 - l_6) = 0 \quad (3.17)$$

Assuming that the spring is unstretched at $x = 0$, which implies that $l_6 = r_0$, one gets the branch transformations:

$$\begin{aligned} x_2 &= x \\ x_6 &= r_0 - x \end{aligned}$$

which shows that the spring-damper shortens as x increases. Substituting these branch transformations and the capacitor voltage (3.12) into equation (3.17), one gets the single ODE for the mechanical domain:

$$m_5 \ddot{x} + d_6 \dot{x} + k_6 x - \frac{1}{2} \frac{dC_2}{dx} \left(-R_1 i_2 - L_3 \frac{di_2}{dt} + E_4(t) \right)^2 = 0 \quad (3.18)$$

Together, equations 3.15 and 3.18 can be solved for the primary variables $i_2(t)$ and $x(t)$; they are equivalent to equations 3.6 obtained with the bond graph approach.

3.2.3 Linear Graph and Multibody Systems

So far, only 1D systems have been considered, the behavior of which is directed by scalar variables. Nevertheless, no restrictions were made on the through and across variables that may be tensors of any order, including vectors, what is preferable when considering 3D multibody systems.

The same basic concepts apply when one models a multidimensional multibody mechanical system using linear graph theory: the system model is obtained by combining the constitutive equations for individual components with the linear cutset and circuit equations resulting from their connectivity. Again, the selection of a tree determines the primary variables appearing in the system equations. The cutset and circuit equations retain a simple form because linear graph theory allows the use of vector modeling variables. However, the constitutive equations for some components will be nonlinear due to the finite rotations of bodies in the system. Furthermore, the physical interpretation of nodes and edges must be generalized.

Each node in the linear graph represents the position and orientation of a body-fixed reference frame, while the edges represent transformations between frames corresponding to physical components: either a joint or a body or a force/torque element (including springs and dampers) or a position vector, also called “rigid-arm” [44]. For each element, there are now two sets of through and across variables: translational and rotational. Thus, there will be two sets of cutset and circuit equations, since these variables cannot be added together. Although the incidence matrix is the same for each, selecting different trees can be used to create different cutset and circuit equations for translation and rotation. This can be used to reduce the system equations to a set that is smaller in number than those generated by conventional multibody formalisms [44].

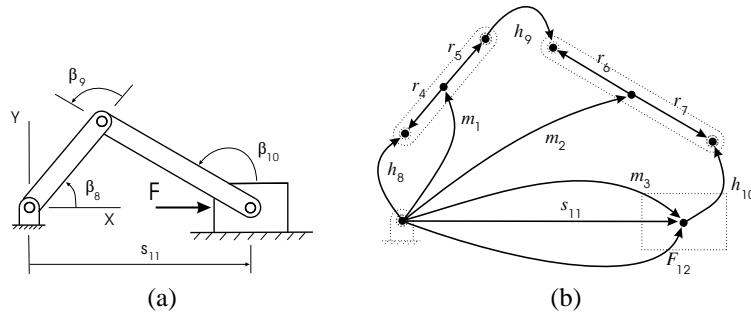


Figure 3.6: Slider-crank mechanism (a) and the corresponding linear graph (b)

To illustrate, consider the planar slider-crank mechanism shown in figure 3.6, along with its linear graph representation. The edges m_1 , m_2 , and m_3 represent both the translational and rotational inertia of the three rigid bodies (crank, connecting rod, and slider). These bodies are connected by revolute joints h_8 , h_9 , and h_{10} and by the prismatic joint s_{11} . The “rigid arm” elements $r_4 - r_7$ define the position and orientation, relative to the center of mass frames on the bodies, of the body-fixed frames

that define the connection points of these joints. Finally, the external force on the slider is modeled by the force element F_{12} , originating at the inertial frame (node) and terminating at the slider.

The constitutive equations for the multi-dimensional translation of rigid bodies and spring-dampers are the same as that shown in the previous section, but in terms of vectors. However, a second equation relates the torques on the body to its rotational inertia. This equation corresponds to Euler's equations for rotational motion, see section 2.1.6.

In the example from figure 3.6, for the rigid-arm elements, e.g. r_4 , the tip node does not rotate relative to the tail (center of mass) node; hence, the angular velocity ω_4 is zero. However, the translational velocity of the rigid-arm is a nonlinear function of the angular velocity of the body on which it resides, e.g. $\mathbf{v}_4 = \omega_1 \times \mathbf{r}_4$, which is a well-known result from rigid body kinematics². For the ideal joints, one always finds that the motion allowed by a joint, e.g. $\mathbf{r}_{11} = s_{11} \hat{\mathbf{i}}$ where \mathbf{r}_{11} is the translational displacement of the slider along X , is orthogonal to the reaction forces and torques that arise in the joint, e.g. $\mathbf{F}_{11} = F_{11} \hat{\mathbf{j}}$ and $\mathbf{T}_{11} = T_{11} \hat{\mathbf{k}}$ where $\hat{\mathbf{j}}$ and $\hat{\mathbf{k}}$ are unit vectors parallel to Y and Z directions, respectively. This is a result of the fact that ideal joints do no work, and it can be used to eliminate joint reactions in the system dynamic equations. This will also be observed later on when presenting the Virtual Work Principle in details, see section 3.3.1.

Because of the large rotations of the bodies, non-linearities arise in the constitutive equations, but, fortunately, the topological equations, 3.8 and 3.9, remain linear. Furthermore, the selection of trees can again be used to define the primary variables s^m and λ^m in the final system equations. The "branch coordinates" s^m are the unknown across variables for elements (branches) in the tree. By selecting components with known across variables (e.g. $r_4 - r_7$) into the tree, the number n of branch coordinates (and system equations) is reduced. If the tree is completed by $m_1 - m_3$, then the final equations are in terms of the absolute coordinates for the three bodies. If joints $h_8 - h_{10}$ are selected in place of $m_1 - m_3$, then one obtains equations in the joint coordinates $\beta_8 - \beta_{10}$. Thus, linear graph theory provides a unification of traditional absolute and relative (or joint) coordinate formulations.

Any joints left in the cotree, e.g. s_{11} , will provide the reaction loads (F_{11} and T_{11}) appearing in the Lagrange multipliers λ^m . Furthermore, these cotree joints will also provide one kinematic constraint equation for each reaction load. These p constraint equations express the relationships between the branch coordinates, which will not be independent if there are joints in the cotree. For the slider-crank example, the three joint coordinates $\beta_8 - \beta_{10}$ must be related by two constraint equations because the system has only 1 degree of freedom (d.o.f.). These constraint equations are always found by projecting the circuit equations for the cotree joints onto their reaction spaces.

²see section 2.1 for more details.

To illustrate, the translational and rotational circuit equations for cotree joint s_{11} are:

$$\mathbf{r}_8 - \mathbf{r}_4 + \mathbf{r}_5 + \mathbf{r}_9 - \mathbf{r}_6 + \mathbf{r}_7 - \mathbf{r}_{10} - \mathbf{r}_{11} = 0 \quad (3.19)$$

$$\beta_8 - \beta_4 + \beta_5 + \beta_9 - \beta_6 + \beta_7 - \beta_{10} - \beta_{11} = 0 \quad (3.20)$$

where $\mathbf{r}_8 = \mathbf{r}_9 = \mathbf{r}_{10} = 0$ from the constitutive equations for revolute joints, and all terms in the rotational equation are constant (again, from the constitutive equations) except for the three branch coordinates $\beta_8 - \beta_{10}$. Note that the circuit equations represent the zero summation of displacement vectors around a closed kinematic chain. The reaction space for s_{11} is spanned by unit vectors $\hat{\mathbf{j}}$ and $\hat{\mathbf{k}}$, as mentioned previously. Projecting the circuit equations onto these two unit vectors, and substituting all constitutive equations, results in the kinematic constraint equations:

$$h(s^m) = 0 \quad (3.21)$$

where the column matrix h is a nonlinear algebraic function of $s^m = \begin{pmatrix} \beta_8 \\ \beta_9 \\ \beta_{10} \end{pmatrix}$.

To obtain the n dynamic equations of the system, the cutset equations for each branch are projected onto the motion space for that branch. For example, for a revolute joint, the motion space is defined by the vector parallel to the joint axis, while the reaction space is the plane orthogonal to that same vector.

As an example, the motion space for branch joint h_9 is spanned by $\hat{\mathbf{k}}$, the unit vector parallel to the joint axis. The rotational cutset equation for h_9 is:

$$\mathbf{T}_9 + \mathbf{T}_2 + \mathbf{T}_3 + \mathbf{T}_{11} + \mathbf{T}_{12} = 0 \quad (3.22)$$

which represents the rotational dynamic equilibrium for bodies m_2 and m_3 that are isolated by the cutset. Projecting this equation onto $\hat{\mathbf{k}}$, and using the constitutive equations $\mathbf{T}_9 = \mathbf{T}_{12} = 0$, one obtains a dynamic equation in terms of the inertia of m_2 and m_3 , and the cotree joint reaction torque \mathbf{T}_{11} . Generating the two dynamic equations for the other two joints in the tree, and assembling in matrix form, gives the system dynamic equations:

$$M^m \ddot{s}^m + Q^m = J^{m^T} \lambda^m \quad (3.23)$$

where M^m is the $n \times n$ mass matrix, J^m is the Jacobian matrix of the kinematic constraint equations, and Q^m contains external loads and quadratic velocity terms. Together, the kinematic and dynamic equations constitutes a set of $n + p$ nonlinear differential-algebraic equations (DAEs), the solution of which will give the branch coordinates $s^m(t)$ and the Lagrange multipliers $\lambda^m(t)$.

Note that one can also generate the dynamic equations by combining linear graph theory with analytical mechanics, e.g. the Virtual Work Principle. For instance, this approach is very useful for incorporating flexible bodies into the multibody model [68], but it requires to abandon the idea of a fully unified theory.

3.2.4 Discussion

The linear graph theory presented here is of course perfectly suited to electrical circuits and it was explained that it allows to generate models of multibody systems, with interesting features such as the possibility to mix absolute and relative coordinates, or the automatic constraint formulation by means of chord transformations. It therefore provides an interesting platform for the modeling of electromechanical systems.

Nevertheless, we can point out some drawbacks of this technique:

- Constraints deduced from the topology are naturally taken into account but any additional constraint (ex.: a wheel on a railway track, a “user constraint”, etc.) requires definition of new specific element.
- Efficiency of the generated dynamic equations for MBS can be called into question, when compared to recursive techniques³.

These points are usually not really problematic but they might become critical when considering applications with large and complex multibody structures and tight electromechanical interaction, for which simulation times and model efficiency are essential.

3.3 Virtual Work Principle

As already observed with Bond Graph and Linear Graph theories, as soon as multi-domain systems are considered, energy and power are essential concepts.

The Virtual Work Principle [31, 67, 76, 77] was first developed in mechanics and several researchers already extended this technique to electrical systems [46, 62, 73]. It offers a rigorous unified theoretical framework for modeling of electromechanical systems, and leads to a minimal set of equations, especially when expressed under the Lagrange form.

Lagrange equations have been proposed by several authors as an unified formulation for electromechanical systems:

- Maisser [42] proposes to model multibody systems coupled with electromechanical drives using Lagrange equations. He limits himself to electromagnetic coupling and do not consider constraints.
- Scherpen [62] uses the Lagrange multiplier technique to model switching electrical circuits and also mutual inductances.
- Hadwich and Pfeiffer [26] propose to consider the electromechanical coupling as constraints between electrical and mechanical variables.

³Although Linear graph theory does not preclude it, as far as we know, current implementations do not consider recursive formulation.

All these formalisms consider either charges or flux as generalized variables, which are not conventional variables in circuit theory, which classically consider currents and voltages.

Schlacher [64] developed another formulation for electromechanical systems where he considers Lagrange equations for the mechanical subsystems and original equations, based on power functions, in terms of currents and voltages but not under Lagrange form, for electrical circuits.

Although virtual work formalisms are well-known, detailed developments are presented in order to show the parallels that may be drawn between electrical and mechanical systems:

- The multibody Virtual Work Principle proposed here starts from the Newton-Euler equations, what is different from classical formulations starting from the d'Alembert principle at local level [31, 67, 76, 77]. The proposed formulation is similar to the work by Wittenburg [78] and Samin [55].
- The electrical virtual work formulation proposed here is inspired from Hadwigh and Pfeiffer [26].
- On the basis of Lagrange equations, an interesting unification of the electrical and mechanical equations is proposed. More general than Maisser's proposition [42], it differs from Hadwigh and Pfeiffer's proposition [26] since no constraint is considered between the electrical and mechanical variables.

3.3.1 Virtual Work Formalism for Multibody Systems

Methods based on Newton-Euler equations with absolute generalized coordinates (with opposition to relative joint coordinates) can be the starting point of a multibody modeling tool. For a system composed of N^{body} bodies, in a three dimensional space, $6N^{body}$ differential equations⁴ must be written, according to 2.33 and 2.35. When two bodies are interconnected by a joint, several relative motions are prevented, what results in *reaction forces* in the joint reaction space constraining the displacements to take only place in the motion space.

Manipulating these $6N^{body}$ equations leads to the desired equations of motion but also to all the reaction forces in the joints. For most applications, the number of degrees of freedom (d.o.f.) is rather low compared to $6N^{body}$ and thus many "unnecessary" quantities are determined if we are only interested in the motion behavior of the system. For example, for a 4-bar planar mechanism, 9 equations can be written, which reduce to 1 equation of motion corresponding to the single d.o.f. This reduction procedure consists in combining the Newton-Euler equations in order to eliminate the 8 reaction forces. This makes the use of Newton-Euler equations with absolute coordinates demanding from a numerical point of view.

The general idea lying behind the Virtual Work Principle is to select in advance the correct linear combinations between the Newton-Euler equations in order to eliminate

⁴ $3N^{body}$ for planar systems.

the reaction forces in the joints. From Newton's second law 2.33 and Euler equation 2.34, and using the dynamic quantities defined in section 2.1.5 for multibody systems, we can write:

$$\sum_{i=1}^{N^{body}} (m^i \ddot{\mathbf{x}}^i - \mathbf{F}_{tot}^i) \cdot \boldsymbol{\alpha}^i + \sum_{i=1}^{N^{body}} (\dot{\mathbf{H}}^i - \mathbf{L}_{tot}^i) \cdot \boldsymbol{\beta}^i = 0 \quad (3.24)$$

Since the expressions into brackets are always equal to zero for all the bodies, the coefficients $\boldsymbol{\alpha}^i$ and $\boldsymbol{\beta}^i$ can be chosen arbitrarily. However, at first, when adding quantities, they must have the same dimensions. This imposes specific units for the components of vectors $\boldsymbol{\alpha}^i$ and $\boldsymbol{\beta}^i$. Different possibilities exist and lead to different approaches:

- $\boldsymbol{\alpha}^i$ and $\boldsymbol{\beta}^i$ are linear and angular virtual velocities respectively. This leads to the *Virtual Power Principle* since the products will have the dimension of mechanical power.
- $\boldsymbol{\alpha}^i$ and $\boldsymbol{\beta}^i$ are linear and angular virtual infinitesimal displacements respectively. This leads to the *Virtual Work Principle* since the products will have the dimension of mechanical work.

In this research, we will focus on the Virtual Work Principle⁵ and the coefficient arrays $\boldsymbol{\alpha}^i$ and $\boldsymbol{\beta}^i$, appearing in 3.24, will be called *virtual displacements* and denoted:

$$\boldsymbol{\alpha}^i = \delta \mathbf{x}^i \quad \boldsymbol{\beta}^i = \delta \boldsymbol{\pi}^i \quad (3.25)$$

The Virtual Work Principle 3.24 can be written:

$$\sum_{i=1}^{N^{body}} (m^i \ddot{\mathbf{x}}^i - \mathbf{F}_{tot}^i) \cdot \delta \mathbf{x}^i + \sum_{i=1}^{N^{body}} (\dot{\mathbf{H}}^i - \mathbf{L}_{tot}^i) \cdot \delta \boldsymbol{\pi}^i = 0 \quad (3.26)$$

Starting from d'Alembert's principle, the Virtual Work Principle can be obtained locally under integral form, but this local formulation of the motion is not necessary for rigid bodies, and, in this text, we remain at the body level. This requires to explicitly take the Euler equations of rotation into account, while local formulations consider only translation equations of infinitesimal mass elements [31, 67, 77].

When presenting the Virtual Work Principle, Wittenburg [78] and Samin [55] also start from 3.26.

⁵Detailed discussion of the Virtual Power Principle can be found in [56]. Both formalisms lead to the identical equations.

3.3.1.1 Virtual displacement

As defined in [76], for any position vector \mathbf{x}^i , a *virtual displacement* - denoted $\delta\mathbf{x}^i$ - corresponds to an arbitrary infinitesimal⁶ change in the generalized coordinates s^m , this change occurring at a fixed instant determined by time t [76].

If a position vector \mathbf{x}^i is expressed as a function of the generalized coordinates⁷ s^m and time t ,

$$\mathbf{x}^i = \mathbf{x}^i(s^m, t) \quad (3.27)$$

and the infinitesimal virtual displacement $\delta\mathbf{x}^i$ is equal to:

$$\delta\mathbf{x}^i = \sum_j \frac{\partial \mathbf{x}^i}{\partial s_j^m} \delta s_j^m = \frac{\partial \mathbf{x}^i}{\partial s^{mT}} \delta s^m \quad (3.28)$$

where δs_j^m is an infinitesimal change in variable s_j^m (at time t). In 3.28 the following notation has been used: $\frac{\partial \mathbf{x}^i}{\partial s^{mT}} \triangleq \left(\frac{\partial \mathbf{x}^i}{\partial s_1^m} \quad \dots \quad \frac{\partial \mathbf{x}^i}{\partial s_n^m} \right)$.

If the position vector \mathbf{x}^i is expressed in the inertial frame $\{\hat{\mathbf{I}}\}$ as

$$\mathbf{x}^i = [\hat{\mathbf{I}}]^T V(s^m, t) \quad (3.29)$$

then, we write the virtual displacement as

$$\delta\mathbf{x}^i = [\hat{\mathbf{I}}]^T \delta V \quad (3.30)$$

with

$$\delta V = \frac{\partial V}{\partial s^{mT}} \delta s^m \quad (3.31)$$

If the position vector \mathbf{x}^i is expressed in a mobile frame $\{\hat{\mathbf{X}}^i\}$ as

$$\mathbf{x}^i = [\hat{\mathbf{X}}^i]^T x^i(s^m, t) \quad (3.32)$$

then, we write the virtual displacement as

$$\delta\mathbf{x}^i = [\hat{\mathbf{I}}]^T \delta \left(R^{iT} x^i \right) \quad (3.33)$$

with R^i such that $[\hat{\mathbf{X}}^i] = R^i [\hat{\mathbf{I}}]$.

⁶The infinitesimal character assumed here is not necessary when using the virtual power principle in which velocities are considered; defining an instantaneous velocity at a given position and time instant is always possible.

⁷For instance, the kinematic relations obtained in section 2.1.4 and 2.1.5 allow us to rewrite the position vector \mathbf{x}^i and velocity vector $\dot{\mathbf{x}}^i$ as function of the joint variables s^m and their derivatives \dot{s}^m

Equation 3.33 can be rewritten as

$$\delta \mathbf{x}^i = [\hat{\mathbf{I}}]^T \left((\delta R^{iT}) x^i + R^{iT} \delta x^i \right) \quad (3.34)$$

Expressing the virtual displacement $\delta \mathbf{x}^i$ in the mobile frame, we get

$$\delta \mathbf{x}^i = [\hat{\mathbf{X}}^i]^T \left((R^i \delta R^{iT}) x^i + \delta x^i \right) \quad (3.35)$$

with $\delta x^i = \frac{\partial x^i}{\partial s^m} \delta s^m$

In order to determine $R^i \delta R^{iT}$, we will start from the fact that the rotation matrix R^i is orthogonal and thus,

$$R^i R^{iT} = E \quad (3.36)$$

where E is the unit matrix. Thus,

$$\delta (R^i R^{iT}) = 0 = (\delta R^i) R^{iT} + R^i \delta R^{iT} \quad (3.37)$$

Equation 3.37 proves that matrix $R^i \delta R^{iT}$ is antisymmetric and thus, we can define a vector $\delta \boldsymbol{\pi}^i$ such that:

$$R^i \delta R^{iT} = \begin{pmatrix} 0 & -\delta \pi_3^i & \delta \pi_2^i \\ \delta \pi_3^i & 0 & -\delta \pi_1^i \\ -\delta \pi_2^i & \delta \pi_1^i & 0 \end{pmatrix} = \delta \tilde{\boldsymbol{\pi}}^i \quad (3.38)$$

where $\delta \pi_1^i, \delta \pi_2^i$ and $\delta \pi_3^i$ are the components of $\delta \boldsymbol{\pi}^i$ in frame $\{\hat{\mathbf{X}}^i\}$: $\delta \boldsymbol{\pi}^i = [\hat{\mathbf{X}}^i]^T \delta \boldsymbol{\pi}^i$ and $\delta \boldsymbol{\pi}^i = (\delta \pi_1^i \quad \delta \pi_2^i \quad \delta \pi_3^i)^T$.

Vector $\delta \boldsymbol{\pi}^i$ is only valid for infinitesimal variations and will be called *the pseudo-vector of infinitesimal rotation* [76]. It will be used in the Virtual Work Principle 3.26 for each body. This vector has been defined in a way that is very similar to the absolute angular velocity $\boldsymbol{\omega}$ and benefits from the same properties (see equations 2.5 to 2.15). In particular, equation 3.35 may be rewritten as

$$\delta \mathbf{x}^i = [\hat{\mathbf{X}}^i]^T \delta x^i + \delta \boldsymbol{\pi}^i \times \mathbf{x}^i \quad (3.39)$$

From 3.38, it is possible to rewrite $\delta \boldsymbol{\pi}^i$ as a linear combination of the δs_j^m . Indeed, assuming that the rotation matrix can be expressed as a function of the generalized coordinates s^m and time t ,

$$R^i = R^i(s^m, t), \quad (3.40)$$

we can write

$$\delta R^i = \sum_{j=1}^n \frac{\partial R^i}{\partial s_j^m} \delta s_j^m \quad (3.41)$$

and thus

$$\widetilde{\delta\pi}^i = \sum_{j=1}^n R^i \left(\frac{\partial R^i}{\partial s_j^m} \right)^T \delta s_j^m = \sum_{j=1}^n \widetilde{b}_j^i \delta s_j^m \quad (3.42)$$

where the tilde matrix $\widetilde{b}_j^i \triangleq R^i \frac{\partial R^i}{\partial s_j^m}{}^T$ has been introduced. Defining vector $\mathbf{b}_j^i \triangleq [\widehat{\mathbf{X}}^i]^T b_j^i$ as the associated vector, we can write

$$\delta\pi^i = \sum_{j=1}^n \mathbf{b}_j^i \delta s_j^m \quad (3.43)$$

Recalling the definition of the angular velocity vector $\boldsymbol{\omega}^i$ (2.5)

$$\widetilde{\boldsymbol{\omega}}^i = R^i \left(\dot{R}^i \right)^T \quad (3.44)$$

and using

$$\dot{R}^i = \sum_{j=1}^n \frac{\partial R^i}{\partial s_j^m} \dot{s}_j^m + \frac{\partial R^i}{\partial t} \quad (3.45)$$

we get,

$$\widetilde{\boldsymbol{\omega}}^i = \sum_{j=1}^n R^i \left(\left(\frac{\partial R^i}{\partial s_j^m} \right)^T \dot{s}_j^m + \frac{\partial R^i}{\partial t} \right) \quad (3.46)$$

By comparing 3.46 and 3.42, we obtain

$$\frac{\partial \widetilde{\boldsymbol{\omega}}^i}{\partial \dot{s}_j^m} = R^i \left(\frac{\partial R^i}{\partial s_j^m} \right)^T = \widetilde{b}_j^i \quad (3.47)$$

and thus,

$$\mathbf{b}_j^i = \frac{\partial \boldsymbol{\omega}^i}{\partial \dot{s}_j^m} \quad (3.48)$$

This property will be used later on when developing the Virtual Work Principle.

3.3.1.2 Tree-like multibody systems

In this section, we will assume that n generalized coordinates

$$s^m = (s_1^m \quad \dots \quad s_n^m)^T$$

were defined⁸ and that \mathbf{x}^i and R^i do not exhibit explicit dependence with respect to time⁹ t :

$$\mathbf{x}^i = \mathbf{x}^i(s^m) \quad (3.49)$$

$$\dot{\mathbf{x}}^i = \sum_{j=1}^n \frac{\partial \mathbf{x}^i}{\partial s_j^m} \dot{s}_j^m \quad (3.50)$$

$$R^i = R^i(s^m) \quad (3.51)$$

From 3.28 and 3.43, the Virtual Work Principle 3.26 may be rewritten as

$$\sum_{i=1}^{N^{body}} (m^i \ddot{\mathbf{x}}^i - \mathbf{F}_{tot}^i) \cdot \sum_{j=1}^n \frac{\partial \mathbf{x}^i}{\partial s_j^m} \delta s_j^m + \sum_{i=1}^{N^{body}} (\dot{\mathbf{H}}^i - \mathbf{L}_{tot}^i) \cdot \sum_{j=1}^n \mathbf{b}_j^i \delta s_j^m = 0 \quad (3.52)$$

in which the terms can be recombined as

$$\underbrace{\sum_{i=1}^{N^{body}} \sum_{j=1}^n m^i \ddot{\mathbf{x}}^i \cdot \frac{\partial \mathbf{x}^i}{\partial s_j^m} \delta s_j^m}_{\boxed{1}} + \underbrace{\sum_{i=1}^{N^{body}} \sum_{j=1}^n \dot{\mathbf{H}}^i \cdot \mathbf{b}_j^i \delta s_j^m}_{\boxed{2}} - \underbrace{\sum_{j=1}^n Q_j^{tot} \delta s_j^m}_{\boxed{3}} = 0 \quad (3.53)$$

where Q_j^{tot} is defined as the total generalized force associated with s_j^m ,

$$Q_j^{tot} = \sum_{i=1}^{N^{body}} \left(\mathbf{F}_{tot}^i \cdot \frac{\partial \mathbf{x}^i}{\partial s_j^m} + \mathbf{L}_{tot}^i \cdot \mathbf{b}_j^i \right) \quad (3.54)$$

Let us now derive the Lagrange form of equation 3.53.

The first term $\boxed{1}$ of equation 3.53 can be written as

$$\sum_{i=1}^{N^{body}} \sum_{j=1}^n m^i \frac{d}{dt} \left(\dot{\mathbf{x}}^i \cdot \frac{\partial \mathbf{x}^i}{\partial s_j^m} \right) \delta s_j^m - \sum_{i=1}^{N^{body}} \sum_{j=1}^n m^i \dot{\mathbf{x}}^i \cdot \frac{d}{dt} \left(\frac{\partial \mathbf{x}^i}{\partial s_j^m} \right) \delta s_j^m \quad (3.55)$$

Taking the first time derivative of $\frac{\partial \mathbf{x}^i}{\partial s_j^m}$, we get

$$\frac{d}{dt} \left(\frac{\partial \mathbf{x}^i}{\partial s_j^m} \right) = \sum_{k=1}^n \frac{\partial^2 \mathbf{x}^i}{\partial s_j^m \partial s_k^m} \dot{s}_k^m \quad (3.56)$$

⁸For instance the joint variables s^m defined in section 2.1.4. In this case, $n = N^{body}$.

⁹If the motion of a joint is imposed as a function of time t , it will be considered as a constraint $s_j^m = f(t)$.

Taking the partial derivative of $\dot{\mathbf{x}}^i$ (equation 3.50) with respect to s_j^m , one get

$$\frac{\partial \dot{\mathbf{x}}^i}{\partial s_j^m} = \sum_{k=1}^n \frac{\partial^2 \mathbf{x}^i}{\partial s_j^m \partial s_k^m} \dot{s}_k^m \quad (3.57)$$

Comparing equation 3.56 with equation 3.57, one concludes that

$$\frac{d}{dt} \left(\frac{\partial \dot{\mathbf{x}}^i}{\partial s_j^m} \right) = \frac{\partial \dot{\mathbf{x}}^i}{\partial s_j^m} \quad (3.58)$$

From equation 3.50, $\dot{\mathbf{x}}^i = \sum_{j=1}^n \frac{\partial \dot{\mathbf{x}}^i}{\partial s_j^m} \dot{s}_j^m$, we easily deduce that

$$\frac{\partial \dot{\mathbf{x}}^i}{\partial \dot{s}_j^m} = \frac{\partial \mathbf{x}^i}{\partial s_j^m} \quad (3.59)$$

Replacing 3.58 and 3.59 into equation 3.55, term $\boxed{1}$ of 3.53 becomes

$$\sum_{i=1}^{N^{body}} \sum_{j=1}^n m^i \frac{d}{dt} \left(\dot{\mathbf{x}}^i \cdot \frac{\partial \dot{\mathbf{x}}^i}{\partial \dot{s}_j^m} \right) \delta s_j^m - \sum_{i=1}^{N^{body}} \sum_{j=1}^n m^i \dot{\mathbf{x}}^i \cdot \frac{\partial \dot{\mathbf{x}}^i}{\partial s_j^m} \delta s_j^m \quad (3.60)$$

If we define for body i the *kinetic energy of the translation motion* as

$$T^{ti} = \frac{1}{2} m^i \dot{\mathbf{x}}^i \cdot \dot{\mathbf{x}}^i \quad (3.61)$$

we can write:

$$\frac{\partial T^{ti}}{\partial \dot{s}_j^m} = m^i \dot{\mathbf{x}}^i \frac{\partial \dot{\mathbf{x}}^i}{\partial \dot{s}_j^m} \quad (3.62)$$

$$\frac{\partial T^{ti}}{\partial s_j^m} = m^i \dot{\mathbf{x}}^i \frac{\partial \dot{\mathbf{x}}^i}{\partial s_j^m} \quad (3.63)$$

and thus, from 3.60, we can rewrite term $\boxed{1}$ of 3.53 as

$$\sum_{i=1}^{N^{body}} \frac{d}{dt} \left(\frac{\partial T^{ti}}{\partial \dot{s}_j^m} \right)^T \delta s_j^m - \sum_{i=1}^{N^{body}} \left(\frac{\partial T^{ti}}{\partial s_j^m} \right)^T \delta s_j^m \quad (3.64)$$

where matrix notation was used: $\left(\frac{\partial T^{ti}}{\partial \dot{s}_j^m} \right)^T \delta s_j^m = \sum_{j=1}^n \frac{\partial T^{ti}}{\partial \dot{s}_j^m} \delta s_j^m$.

Similarly, according to 2.31, $\mathbf{H}^i = \mathbf{I}^i \cdot \boldsymbol{\omega}^i$ and the second term $\boxed{2}$ of 3.53 can be transformed as,

$$\sum_{i=1}^{N^{body}} \sum_{j=1}^n \left(\mathbf{I}^i \cdot \dot{\boldsymbol{\omega}}^i + \tilde{\boldsymbol{\omega}}^i \cdot \mathbf{I}^i \cdot \boldsymbol{\omega}^i \right) \cdot \mathbf{b}_j^i \delta s_j^m \quad (3.65)$$

Expressing all the tensors and vectors in the body-fixed frames $\{\hat{\mathbf{X}}^i\}$, the latter writes

$$\sum_{i=1}^{N^{body}} \sum_{j=1}^n b_j^{iT} (I^i \dot{\omega}^i + \tilde{\omega}^i I^i \omega^i) \delta s_j^m \quad (3.66)$$

where I^i is the inertia matrix i with respect to its center of mass. The latter is a constant when rigid bodies are considered.

The following property is satisfied by b_j^i and ω^i [54, 76]:

$$\frac{db_j^{iT}}{dt} - \frac{\partial \omega^{iT}}{\partial s_j^m} = b_j^{iT} \tilde{\omega}^i \quad (3.67)$$

Replacing the latter inside 3.66, we get

$$\sum_{i=1}^{N^{body}} \sum_{j=1}^n \left(b_j^{iT} I^i \dot{\omega}^i + \left(\frac{db_j^{iT}}{dt} - \frac{\partial \omega^{iT}}{\partial s_j^m} \right) I^i \omega^i \right) \delta s_j^m \quad (3.68)$$

Defining the *kinetic energy of the rotation* motion as

$$T^{ri} \triangleq \frac{1}{2} \omega^{iT} \cdot I^i \cdot \omega^i = \frac{1}{2} \boldsymbol{\omega}^i \cdot \mathbf{H}^i = \frac{1}{2} \mathbf{H}^i \cdot \boldsymbol{\omega}^i. \quad (3.69)$$

we will show that

$$\frac{d}{dt} \left(\frac{\partial T^{ri}}{\partial \dot{s}_j^m} \right) - \frac{\partial T^{ri}}{\partial s_j^m} = b_j^{iT} I^i \dot{\omega}^i + \left(\frac{db_j^{iT}}{dt} - \frac{\partial \omega^{iT}}{\partial s_j^m} \right) I^i \omega^i \quad (3.70)$$

Indeed, taking the partial derivative $\frac{\partial T^{ri}}{\partial \dot{s}_j^m}$ and using 3.48 we get

$$\frac{\partial T^{ri}}{\partial \dot{s}_j^m} = \frac{\partial \omega^{iT}}{\partial \dot{s}_j^m} I^i \omega^i = b_j^{iT} I^i \omega^i \quad (3.71)$$

the first time derivative of which writes:

$$\frac{d}{dt} \left(\frac{\partial T^{ri}}{\partial \dot{s}_j^m} \right) = \frac{db_j^{iT}}{dt} I^i \omega^i + b_j^{iT} I^i \dot{\omega}^i \quad (3.72)$$

On the other hand, the partial derivative $\frac{\partial T^{ri}}{\partial s_j^m}$ gives us

$$\frac{\partial T^{ri}}{\partial s_j^m} = \frac{\partial \omega^{iT}}{\partial s_j^m} I^i \omega^i \quad (3.73)$$

Finally, using 3.70, the second term $\boxed{2}$ of 3.53 transforms into,

$$\sum_{i=1}^{N^{body}} \sum_{j=1}^n \left(\frac{d}{dt} \left(\frac{\partial T^{ri}}{\partial \dot{s}_j^m} \right) - \frac{\partial T^{ri}}{\partial s_j^m} \right) \delta s_j^m \quad (3.74)$$

which can be written as, with matrix notation (as in 3.64),

$$\sum_{i=1}^{N^{body}} \frac{d}{dt} \left(\frac{\partial T^{ri}}{\partial \dot{s}^m} \right)^T \delta s^m - \sum_{i=1}^{N^{body}} \left(\frac{\partial T^{ri}}{\partial s^m} \right)^T \delta s^m \quad (3.75)$$

The total kinetic energy of a multibody system is defined¹⁰ as

$$T \triangleq \sum_{i=1}^{N^{body}} (T^{ti} + T^{ri}) \quad (3.76)$$

Replacing 3.64 and 3.75 into 3.53, the final equation of motion can be written as:

$$\left(\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{s}^m} \right) - \frac{\partial T}{\partial s^m} - Q^{tot} \right)^T \delta s^m = 0 \quad (3.77)$$

which has the form

$$\Phi^{mT} \delta s^m = 0 \quad (3.78)$$

Since equation 3.78 is valid for any independent joint's virtual displacements, we must have $\Phi^m = 0$, and thus

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{s}^m} \right) - \frac{\partial T}{\partial s^m} - Q^{tot} = 0 \quad (3.79)$$

For multibody systems, the total generalized forces Q^{tot} have been defined in equation 3.54. Using 3.48, the latter becomes:

$$Q^{totT} = \sum_{i=1}^{N^{body}} \mathbf{F}_{tot}^i \cdot \frac{\partial \mathbf{x}^i}{\partial \dot{s}^{mT}} + \sum_{i=1}^{N^{body}} \mathbf{L}_{tot}^i \cdot \frac{\partial \boldsymbol{\omega}^i}{\partial \dot{s}^{mT}} \quad (3.80)$$

where \mathbf{F}_{tot}^i and \mathbf{L}_{tot}^i respectively represent the resultant of all the forces and torques acting on body i .

Selecting virtual displacements δs^m compatible with the joint kinematics, it is a well-known fact that the reaction forces (also called constraint forces) of the joints disappear from the equation of motion, since their contribution to the virtual work vanishes. Imposing compatible virtual displacements δs^m , we satisfy the *implicit constraints* associated with the joint kinematics.

¹⁰The kinetic energy is often defined as $T = \int (\dot{\mathbf{r}} \cdot \dot{\mathbf{r}}) dm$. For multibody systems with rigid bodies, this is equivalent to the definition given in this text.

At this stage, we can deduce another form of 3.79. Starting from the kinetic energy

$$T = \sum_{i=1}^{N^{body}} \left(\frac{1}{2} m^i \dot{\mathbf{x}}^i \cdot \dot{\mathbf{x}}^i + \frac{1}{2} \boldsymbol{\omega}^i \cdot \mathbf{H}^i \right) \quad (3.81)$$

$$= \sum_{i=1}^{N^{body}} \left(\frac{1}{2} m^i \dot{x}^i \cdot \dot{x}^i + \frac{1}{2} \omega^i \cdot I^i \omega^i \right) \quad (3.82)$$

where $\dot{\mathbf{x}}^i = [\hat{\mathbf{I}}]^T \dot{x}^i$, $\boldsymbol{\omega}^i = [\hat{\mathbf{X}}^i]^T \omega^i$ and $\mathbf{H}^i = [\hat{\mathbf{X}}^i]^T I^i [\hat{\mathbf{X}}^i]$.

Let us rewrite equations 3.50 and 3.46 as

$$\dot{x}^i = \frac{\partial x^i}{\partial s^{mT}} \dot{s}^m \quad (3.83)$$

$$\omega^i = \frac{\partial \omega^i}{\partial \dot{s}^{mT}} \dot{s}^m \quad (3.84)$$

Replacing the latter in the kinetic energy, expression 3.82 becomes

$$\begin{aligned} T &= \frac{1}{2} \dot{s}^{mT} \sum_{i=1}^{N^{body}} \left(m^i \left(\frac{\partial x^i}{\partial s^{mT}} \right)^T \left(\frac{\partial x^i}{\partial s^{mT}} \right) \right) \dot{s}^m \\ &\quad + \frac{1}{2} \dot{s}^{mT} \sum_{i=1}^{N^{body}} \left(\left(\frac{\partial \omega^i}{\partial \dot{s}^{mT}} \right)^T I^i \left(\frac{\partial \omega^i}{\partial \dot{s}^{mT}} \right) \right) \dot{s}^m \\ &= \frac{1}{2} \dot{s}^{mT} M \dot{s}^m \end{aligned} \quad (3.85)$$

in which $\frac{\partial \omega^i}{\partial \dot{s}^{mT}}$ is only function of s^m , according to 3.46.

Introducing 3.85 in the equation of motion 3.79, the latter can be rewritten in the form:

$$M(s^m) \ddot{s}^m + c(s^m, \dot{s}^m, F_{ext}, L_{ext}, g) = Q(s^m, \dot{s}^m) \quad (3.86)$$

where the generalized accelerations \ddot{s}^m appear *linearly*. The matrix M is called the *generalized mass matrix*, which only depends on the generalized variables s^m . From 3.85, we can conclude that M is symmetric.

The kinetic energy is always positive and, from linear algebra theory [70], a symmetric matrix M such that $\dot{s}^{mT} M \dot{s}^m$ is positive for any $\dot{s}^m \neq 0$ is *symmetric positive definite*.

In expression 3.86, we have explicitly separated the generalized forces associated with the external forces F_{ext} and torques L_{ext} , including gravity, from the joint generalized forces Q .

Lagrange equations Among the forces acting on a system, we can always distinguish energy *conservative forces*. These kind of forces are characterized by a *potential function* $U(s^m)$ such that:

$$Q^c = -\frac{\partial U}{\partial s^m} \quad (3.87)$$

where Q^c refers to the generalized forces associated with conservative forces.

The virtual work expression 3.79 then becomes:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{s}^m} \right) - \frac{\partial T}{\partial s^m} + \frac{\partial U}{\partial s^m} - Q^{nc} = 0 \quad (3.88)$$

where Q^{nc} represents the generalized forces associated with non-conservative forces.

Defining the *Lagrangian* L^m of the mechanical system by

$$L^m \triangleq T - U, \quad (3.89)$$

the equations of motion can be written as:

$$\frac{d}{dt} \left(\frac{\partial L^m}{\partial \dot{s}^m} \right) - \frac{\partial L^m}{\partial s^m} - Q^{nc} = 0 \quad (3.90)$$

Equations 3.90 are the *Lagrange equations of motion* of a mechanical multibody system.

3.3.1.3 Constrained multibody systems: Lagrange multipliers technique

In the previous section, we have assumed a tree-like structure of the multibody system. The chosen generalized coordinates, the joint variables s^m , were independent and we could derive the equations of motion under the Lagrange form 3.90. Tree-like systems can be found for instance in some robotic applications.

However, in many practical cases, physical systems are not tree-like because closed-loops exist, as illustrated in figure 2.1, and the joint variables s^m are not independent anymore.

The previously described formalism is still valid if we define a *spanning tree*, that is a tree covering all the bodies of the system. This spanning tree can be obtained by *cutting* the loops, either by cutting bodies or by disregarding some joints (see figure 3.7).

The number of necessary cuts can be found since, for closed-loop systems, the number of joints N^{joint} is higher than the number of bodies N^{body} , whereas in a tree-like structure $N^{joint} = N^{body}$. The difference $N^{cut} \triangleq N^{joint} - N^{body}$ defines the *cyclomatic* number corresponding to the number of independent cuts which are necessary to form a spanning tree. Independent cuts are such that their closure automatically closes all the loops in the system.

Of course, for each independent loop, closure conditions have to be imposed to the spanning tree variables s^m and the number of d.o.f. will be directly related to the number of conditions. The latter depends on how we cut the loops [56].

To illustrate these concepts, let us consider the example of figure 3.7. In this example, three loops exist but two cuts are sufficient. Closing cut 1 and cut 2 automatically closes all the loops, but closing cut 1 does not close loop 2 and vice versa: cuts 1 and 2 are independent.

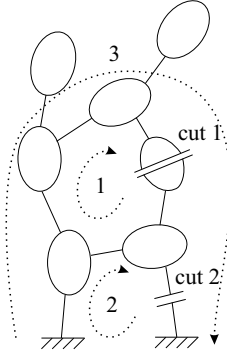


Figure 3.7: Independent loops and cuts

The loop closure conditions will lead to *explicit constraints* (in contrast with implicit constraint arising from the nature of the joints), between the generalized coordinates s^m corresponding to the spanning tree joint variables. These constraints are qualified as *holonomic constraints* because they relate the generalized variables at position level. Generally non-linear, they cannot be solved analytically.

The explicit holonomic constraints, discussed above, take the general form

$$h(s^m, t) = 0 \quad (3.91)$$

Driven joints, for which the motion is imposed, are also such that the corresponding joint variable is constrained to be equal to a given function of time t , what can be written under the form 3.91.

The first time-derivative of the constraints can be written

$$\dot{h}(\dot{s}^m, s^m, t) = J^m(s^m, t)\dot{s}^m + \frac{\partial h(s^m, t)}{\partial t} = 0 \quad (3.92)$$

where $J^m = \left(\frac{\partial h}{\partial s^{mT}} \right)$ is the *Jacobian* matrix associated with the constraints.

Sometimes, other constraints may exist at velocity level only, because they are not integrable, under the form 3.92. These constraints are called *non-holonomic* constraints and are linear¹¹ in the system generalized velocities \dot{s}^m . A typical example is the pure rolling contact of a disc on a plane. To summarize, we can say that equation 3.92 gathers the non-holonomic constraints and the first time derivative of holonomic constraints.

¹¹So far, non-holonomic constraints non-linear in terms of velocities have not been encountered. They are not considered here.

The constraints can be written at acceleration level as

$$\ddot{h}(\dot{s}^m, \dot{s}^m, s^m, t) = J^m(s^m, t)\ddot{s}^m + \left(\dot{J}^m(\dot{s}^m, s^m, t) + \frac{\partial^2 h}{\partial s^{mT} \partial t} \right) \dot{s}^m + \frac{\partial^2 h}{\partial t^2} = 0 \quad (3.93)$$

In the present case, because of these explicit constraints, the generalized coordinates s^m are not independent.

Let us choose virtual changes δs^m in the joint variables, which do not satisfy the explicit constraints. The associated constraint forces produce a virtual work that has to be taken into account in the virtual work expressions. Their effect is taken into account by a supplementary term Q' in the equation of motion as

$$M(s^m)\ddot{s}^m + c(s^m, \dot{s}^m, F_{ext}, L_{ext}, g) = Q + Q' \quad (3.94)$$

where Q' represents the generalized forces associated with the explicit constraints.

On the other hand, if the virtual displacements δs^m are compatible with the explicit constraints, $J^{mT} \delta s^m = 0$, and the generalized constraint forces Q' produce no work. Mathematically, this writes:

$$Q'^T \delta s^m = 0 \text{ for any } \delta s^m \text{ that satisfies: } J^{mT} \delta s^m = 0 \quad (3.95)$$

From elementary algebra this can only be true if the generalized forces are in the subspace defined by the rows of the Jacobian matrix and therefore, Q' must be a linear combination of the rows of the Jacobian matrix¹²,

$$Q' = J^{mT} \lambda^m \quad (3.96)$$

where λ^m are the Lagrange multipliers and J^m is the Jacobian of the constraints, and the new set of equations to be considered is:

$$\begin{aligned} M(s^m)\ddot{s}^m + c(s^m, \dot{s}^m, F_{ext}, L_{ext}, g) &= Q + J^{mT} \lambda^m \\ h(s^m, t) &= 0 \end{aligned} \quad (3.97)$$

Elimination of the constraints and the Lagrange multipliers from equation 3.97 can be done to obtain a reduced set of purely differential equations of the form:

$$M_r(s_u^m, t)\ddot{s}_u^m + c_r(s_u^m, \dot{s}_u^m, F_{ext}, L_{ext}, g, t) = Q_r \quad (3.98)$$

Like with graph theories, reduction procedures are out of the scope of the unified Virtual Work Principle theory. Nevertheless, from our point of view, it follows more naturally equational approaches, as the one presented in this section.

The generalized coordinate partitioning method [75] is a popular method to achieve the reduction from equation 3.97 to 3.98 and it will be discussed in details in section 4.1.3.

¹²A purely mathematical proof for relation 3.96 is proposed in [76].

3.3.1.4 Discussion

The main advantage in using Virtual Work Principle is the direct elimination of the reaction forces and torques associated with implicit and explicit constraints, by means of a good choice of the virtual changes δs^m and of the Lagrange multipliers technique. Additionally, it is helpful to demonstrate the symmetric positive definite character of the generalized mass matrix M .

Nevertheless, in the equations obtained from the Virtual Work Principle, multiple summations and partial derivatives appear. They result in a rapidly growing number of operations when the number of d.o.f. rises, which often becomes excessive for complex applications with numerous d.o.f. This makes the Virtual Work Principle inefficient when considering large and complex multibody systems [56], except if the summations and derivatives can be computed in a recursive way.

Unnecessary terms are computed when starting directly from Lagrange equations 3.90. Indeed, looking at the term $\frac{d}{dt} \frac{\partial T}{\partial \dot{s}^m}$ of 3.79, its practical computation will lead to terms cancelling those computed in $\frac{\partial T}{\partial s^m}$. For example, looking at the translational kinematic energy¹³ T^{ti} , we can write from 3.62

$$\frac{d}{dt} \frac{\partial T^{ti}}{\partial \dot{s}_j^m} = m^i \ddot{\mathbf{x}}^i \frac{\partial \dot{\mathbf{x}}^i}{\partial \dot{s}_j^m} + m^i \dot{\mathbf{x}}^i \frac{d}{dt} \frac{\partial \dot{\mathbf{x}}^i}{\partial \dot{s}_j^m} \quad (3.99)$$

$$= m^i \ddot{\mathbf{x}}^i \frac{\partial \dot{\mathbf{x}}^i}{\partial \dot{s}_j^m} + m^i \dot{\mathbf{x}}^i \frac{d}{dt} \frac{\partial \dot{\mathbf{x}}^i}{\partial \dot{s}_j^m} \quad (3.100)$$

$$= m^i \ddot{\mathbf{x}}^i \frac{\partial \dot{\mathbf{x}}^i}{\partial \dot{s}_j^m} + m^i \dot{\mathbf{x}}^i \frac{\partial \dot{\mathbf{x}}^i}{\partial s_j^m} \quad (3.101)$$

where 3.58 and 3.59 have been used successively.

Using 3.63, the second term of 3.101 will be cancelled when subtracting $\frac{\partial T^{ti}}{\partial s_j^m}$ in the Lagrange equation.

3.3.2 Virtual Work for Electrical Systems

The Virtual Work Principle, and more specifically the Lagrange equations, originally formulated for mechanical systems, have been extended to other physical domains and in particular to electrical systems by several authors, among which Meisel [46].

Scherpen [62] bases her work on Van Der Schaft's formulation of Lagrange equations for electrical circuit [73]. She considers classical circuits but also switching electrical circuits used in power electronics. Ramirez and Ortega [69] also based their work on a Lagrange formulation of electrical equations.

The formulation proposed here is inspired from Hadwich and Pfeiffer's proposition [26] and will be detailed in order to highlight the parallelism with the mechanical formulation.

¹³Similar cancellation exist when considering the rotational kinetic energy T^{ri} .

Energy concepts and definitions The *energy of a circuit* is defined as the time integral of the instantaneous power:

$$E = \int_{-\infty}^t u^T i \, dt = \int_{-\infty}^t i^T u \, dt, \quad (3.102)$$

in which u and i are the array of voltage drops across each element and the array of currents¹⁴ flowing through these elements, respectively (see section 2.2).

For a voltage or current source, assuming a generator (active) sign convention for sources, E corresponds to the energy delivered by the source, if positive, and absorbed by the source, if negative. Figure 3.8 illustrates the generator sign convention and shows the directions associated with positive values of voltage drops or currents.



Figure 3.8: Generator (Active) Sign Conventions for Voltage (a) and Current (b) Sources

For a resistor, assuming a receptor (passive) sign convention (see figure 3.9 where the indicated directions correspond to positive values) and linear constitutive equation 2.37, this energy becomes

$$E = \int_{-\infty}^t u_{kk} i_k \, dt = \int_{-\infty}^t R_k i_k^2 \, dt = \int_{-\infty}^t \frac{1}{R_k} u_{kk}^2 \, dt \quad (3.103)$$

It will always be positive and corresponds to dissipated energy, converted into heat by the resistor (Joule losses).

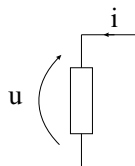


Figure 3.9: Receptor (Passive) Sign Convention

¹⁴We assume zero energy in the circuit at time $t = -\infty$.

For a capacitor, assuming a passive sign convention and linear constitutive equation 2.39, the electrical energy becomes

$$\begin{aligned}
 E &= \int_{-\infty}^t u_{kk} i_k dt \\
 &= \int_{-\infty}^t \frac{1}{C_k} q_k i_k dt \\
 &= \int_0^{q_k(t)} \frac{1}{C_k} q_k dq_k = \frac{1}{2C_k} q_k^2 = \frac{1}{2} C_k u_k^2
 \end{aligned} \tag{3.104}$$

Energy can thus be accumulated inside the capacitor and takes the form of accumulated charges q_k previously defined as $q_k = \int_{-\infty}^t i_k dt$.

For an isolated inductor (no mutual inductance), assuming a passive sign convention and linear constitutive equation 2.38, the electrical energy becomes

$$\begin{aligned}
 E &= \int_{-\infty}^t u_{kk} i_k dt \\
 &= \int_{-\infty}^t \frac{1}{L_{kk}} \varphi_k u_{kk} dt \\
 &= \int_0^{\varphi_k(t)} \frac{1}{L_{kk}} \varphi_k d\varphi_k = \frac{1}{2L_{kk}} \varphi_k^2 = \frac{1}{2} L_k i_k^2
 \end{aligned} \tag{3.105}$$

Energy can thus be accumulated inside the inductor and takes the form of a magnetic flux φ_k , previously defined as $\varphi_k = \int_{-\infty}^t u_{kk} dt$.

The concepts of charges q_k and fluxes φ_k defined here for capacitors and inductors are extended to any electrical components. This extension is purely mathematical since no physical insights can be found for charges accumulated inside an inductor or a resistor, for instance. This allows us to rewrite general expressions 3.102 as

$$E = \int_0^q u^T dq = \int_0^\varphi i^T d\varphi \tag{3.106}$$

We also define the *magnetic energy* W_m , associated with a network of inductors with mutual influences and permanent magnets, as

$$W_m(\varphi^l) \triangleq \int_0^{\varphi^l} i^{lT} d\varphi^l \tag{3.107}$$

The *magnetic co-energy* W_m^* is then defined as

$$W_m^*(i^l) \triangleq i^{lT} \varphi^l - \int_0^{\varphi^l} i^{lT} d\varphi^l = \int_0^{i^l} \varphi^{lT} di^l \tag{3.108}$$

Similarly for capacitors network, we define the *electrical energy* W_e as

$$W_e(q^c) \triangleq \int_0^{q^c} u^{cT} dq^c \tag{3.109}$$

The *electrical co-energy* W_e^* is then defined as

$$W_e^*(u^c) \triangleq u^{cT} q^c - \int_0^{q^c} u^{cT} dq^c = \int_0^{u^c} q^{cT} du^c \quad (3.110)$$

Note: In the linear case (see constitutive equations 2.38 and 2.39), expressions 3.107 to 3.110 become,

$$\begin{aligned} W_m(\varphi^l) &= \frac{1}{2} \varphi^{lT} L^{-1} \varphi^l - \varphi^{pT} L^{-1} \varphi^l \\ W_m^*(i^l) &= \frac{1}{2} i^{lT} L i^l + \varphi^{pT} i^l \\ W_e(q^c) &= \frac{1}{2} q^{cT} C^{-1} q^c \\ W_e^*(u^c) &= \frac{1}{2} u^{cT} C u^c \end{aligned} \quad (3.111)$$

Telegen's Theorem Telegen's theorem is a direct consequence of the conservation of charge applicable in electric circuit theory. It states that [3]:

$$u^T i = i^T u = \sum_{k=1}^n u_{kk} i_k = 0 \quad (3.112)$$

for any compatible set of voltage drops u and currents i . A compatible set is defined as a set of voltage drops u and currents i satisfying Kirchoff's laws.

Telegen's theorem 3.112 can be interpreted as a power conservation principle¹⁵ (first principle of thermodynamics): the instantaneous sum of the powers, produced or absorbed by each element, must be zero if Kirchoff's laws are respected. In other words, the energy of the circuit is constant: all the energy produced by the sources is either accumulated inside the capacitors and inductors or dissipated by the resistors.

3.3.2.1 Virtual Work for electrical circuits

Let us consider a set of voltage drops u and currents i compatible with Kirchoff's equations, Telegen's principle is satisfied at each time. If we fix the time t and apply virtual changes in the charges δq compatible with Kirchoff's laws, according to Telegen's principle, the energy of the circuit must be conserved and thus the variation in energy must be zero:

$$\delta E = u^T \cdot \delta q = 0 \quad (3.113)$$

Similarly, we can imagine that we apply virtual changes in fluxes $\delta \varphi$ compatible with Kirchoff's law and the corresponding virtual change in energy must be zero:

$$\delta E = i^T \cdot \delta \varphi = 0 \quad (3.114)$$

¹⁵more generally expressed in terms of across and through variables, defined in the linear graph theory, and thus applicable to the different fields of physics.

Expressions 3.113 and 3.114 show two possible formulations of the Virtual Work Principle with variations either in terms of charges δq or fluxes $\delta\varphi$.

Both formulations lead to the same equations expressed in terms of different variables. We will present in details the formulation with virtual changes in charges, which is the most common one, while the flux formulation will be shortly introduced.

3.3.2.2 Charges Formulation

Starting from the formulation 3.113, we can separate different terms:

$$u^T \delta q = u^{l^T} \delta q^l + u^{c^T} \delta q^c + u^{r^T} \delta q^r + u^{s^T} \delta q^s = 0 \quad (3.115)$$

The first term from 3.115 can be rewritten as follows:

$$\begin{aligned} u^{l^T} \delta q^l &= \left(\frac{d\varphi^l}{dt} \right)^T \delta q^l \\ &= \frac{d}{dt} (\varphi^{l^T} \delta q^l) - \varphi^{l^T} \frac{d}{dt} (\delta q^l) \end{aligned} \quad (3.116)$$

As mentioned in [26] the operator δ can be interchanged with operator $\frac{d}{dt}$ only when no constraints exist between the variables to which they apply. At this stage, we assume that the charges q^l inside the inductors are independent, and we can write that

$$\frac{d}{dt} (\delta q^l) = \delta \dot{q}^l \quad (3.117)$$

According to the definition of the magnetic co-energy 3.108,

$$\varphi^l = \frac{\partial W_m^*}{\partial \dot{q}^l} \quad (3.118)$$

Replacing 3.117 and 3.118 into 3.116, we get

$$u^{l^T} \delta q^l = \frac{d}{dt} (\varphi^l \delta q^l) - \left(\frac{\partial W_m^*}{\partial \dot{q}^l} \right)^T \delta \dot{q}^l \quad (3.119)$$

Using the electrical energy definition 3.109, the second term from 3.115 can be rewritten as

$$u^{c^T} \delta q^c = \left(\frac{\partial W_e}{\partial q^c} \right)^T \delta q^c \quad (3.120)$$

Introducing 3.119 and 3.120 in 3.115, the Virtual Work Principle in terms of variations in charges becomes

$$\underbrace{\frac{d}{dt} (\varphi^{l^T} \delta q^l) - \left(\frac{\partial W_m^*}{\partial \dot{q}^l} \right)^T \delta \dot{q}^l}_{\boxed{1}} + \underbrace{\left(\frac{\partial W_e}{\partial q^c} \right)^T \delta q^c}_{\boxed{2}} + \underbrace{u^{r^T} \delta q^r + u^{s^T} \delta q^s}_{\boxed{3}} = 0 \quad (3.121)$$

3.3.2.3 Generalized coordinates for charge formulation

Assuming that among the N^{el} elements of the circuit, we have s sources, r resistors, c capacitors and l inductors, Kirchoff's equations and resistors and sources constitutive equations form a set of $N^{el} + r + s$ algebraic relations, which allow us to rewrite all the electrical variables q and u in terms of the charges q^l and q^c , that were chosen as *electrical generalized variables*:

$$s^e = \begin{pmatrix} q^l \\ q^c \end{pmatrix}. \quad (3.122)$$

When these charges q^l and q^c are independent, $l + c$ differential equations can be written and, by analogy with mechanics, this defines the *number of degree of freedom of an electrical circuit* (d.o.f.):

$$\#d.o.f. = l + c \quad (3.123)$$

At this stage, we assume that the generalized variable s^e are independent.

3.3.2.4 Final equations for charges formulation

Because the constitutive equations of the inductors and capacitors are already involved in the process by means of the magnetic co-energy and electrical energy definition, we may use Kirchoff's laws and the constitutive equations of the resistors and sources to rewrite the electrical variables in terms of the generalized variables defined above¹⁶:

$$u^r = u^r(\dot{s}^e, t) \quad (3.124)$$

$$u^{si} = u^{si}(\dot{s}^e, t) \quad (3.125)$$

$$q = q(s^e, t) \quad (3.126)$$

$$\dot{q} = \frac{\partial q}{\partial s^{eT}} \dot{s}^e + \frac{\partial q}{\partial t} \quad (3.127)$$

where $\frac{\partial q}{\partial s^{eT}} = \left(\frac{\partial q}{\partial s_1^e} \ \dots \ \frac{\partial q}{\partial s_{l+c}^e} \right)$. Explicit dependance with time comes from the voltage and current sources. The following developments will help us obtaining Lagrange equations for electrical circuits.

From the fourth expression in 3.127, we can write that

$$\frac{\partial q}{\partial s^{eT}} = \frac{\partial \dot{q}}{\partial \dot{s}^{eT}}. \quad (3.128)$$

Let us consider the first term 1 of 3.121. It may be transformed as follows,

$$\frac{d}{dt} \left(\varphi^{lT} \delta q^l \right) - \left(\frac{\partial W_m^*}{\partial \dot{q}^l} \right)^T \delta \dot{q}^l = \frac{d}{dt} \left(\left(\frac{\partial W_m^*}{\partial \dot{q}^l} \right)^T \frac{\partial q^l}{\partial s^{eT}} \delta s^e \right) - \left(\frac{\partial W_m^*}{\partial \dot{q}^l} \right)^T \delta \dot{q}^l \quad (3.129)$$

¹⁶these relations correspond to the branch and chord transformations from linear graph theory presented in section 3.2.

then, using 3.128, the previous expression becomes

$$\frac{d}{dt} (\varphi^{lT} \delta q^l) - \left(\frac{\partial W_m^*}{\partial \dot{q}^l} \right)^T \delta \dot{q}^l = \frac{d}{dt} \left(\left(\frac{\partial W_m^*}{\partial \dot{q}^l} \right)^T \frac{\partial \dot{q}^l}{\partial \dot{s}^{eT}} \delta s^e \right) - \left(\frac{\partial W_m^*}{\partial \dot{q}^l} \right)^T \delta \dot{q}^l \quad (3.130)$$

Using 3.122 and assuming¹⁷ that $\frac{\partial W_m^*}{\partial \dot{q}^c} = 0$, we can reduce 3.130 as follows,

$$\begin{aligned} \frac{d}{dt} (\varphi^{lT} \delta q^l) - \left(\frac{\partial W_m^*}{\partial \dot{q}^l} \right)^T \delta \dot{q}^l &= \frac{d}{dt} \left(\left(\frac{\partial W_m^*}{\partial \dot{s}^e} \right)^T \delta s^e \right) - \left(\frac{\partial W_m^*}{\partial \dot{s}^e} \right)^T \delta \dot{s}^e \\ &= \frac{d}{dt} \left(\frac{\partial W_m^*}{\partial \dot{s}^e} \right)^T \delta s^e + \left(\frac{\partial W_m^*}{\partial \dot{s}^e} \right)^T \frac{d}{dt} (\delta s^e) \\ &\quad - \left(\frac{\partial W_m^*}{\partial \dot{s}^e} \right)^T \delta \dot{s}^e \\ &= \frac{d}{dt} \left(\frac{\partial W_m^*}{\partial \dot{s}^e} \right)^T \delta s^e + \left(\frac{\partial W_m^*}{\partial \dot{s}^e} \right)^T \delta \dot{s}^e \\ &\quad - \left(\frac{\partial W_m^*}{\partial \dot{s}^e} \right)^T \delta \dot{s}^e \\ &= \frac{d}{dt} \left(\frac{\partial W_m^*}{\partial \dot{s}^e} \right)^T \delta s^e \end{aligned} \quad (3.131)$$

Once again, we have interchanged operator $\frac{d}{dt}$ and δ , when passing from $\frac{d}{dt} (\delta s^e)$ to $\delta \dot{s}^e$. This is valid since we are still under the assumption that the generalized variables s^e are independent. Similar manipulations were accomplished in mechanics with expressions 3.58 and 3.67.

The second term $\boxed{2}$ of 3.121 can also be transformed as, according to $\frac{\partial W_e}{\partial q^l} = 0$,

$$\begin{aligned} u^{cT} \delta q^c &= \left(\frac{\partial W_e}{\partial q^c} \right)^T \delta q^c \\ &= \left(\frac{\partial W_e}{\partial q^c} \right)^T \frac{\partial q^c}{\partial s^{eT}} \delta s^e \\ &= \left(\frac{\partial W_e}{\partial s^e} \right)^T \delta s^e \end{aligned} \quad (3.132)$$

Eventually, the remaining term $\boxed{3}$ of 3.121 becomes

$$u^{rT} \delta q^r + u^{sT} \delta q^s = u^{rT} \frac{\partial q^r}{\partial s^{eT}} \delta s^e + u^{sT} \frac{\partial q^s}{\partial s^{eT}} \delta s^e \quad (3.133)$$

$$= u^{rT} \frac{\partial \dot{q}^r}{\partial \dot{s}^{eT}} \delta s^e + u^{sT} \frac{\partial \dot{q}^s}{\partial \dot{s}^{eT}} \delta s^e \quad (3.134)$$

¹⁷This assumption is not restrictive since the magnetic behavior of inductor can be assumed independent from the electrostatic behavior of capacitors. Magnetic and electrostatic fields are assumed to be independent.

Replacing 3.131, 3.132 and 3.134 into 3.121, we get

$$\frac{d}{dt} \left(\frac{\partial W_m^*}{\partial \dot{s}^e} \right)^T \delta s^e + \left(\frac{\partial W_e}{\partial s^e} \right)^T \delta s^e + u^{rT} \frac{\partial \dot{q}^r}{\partial \dot{s}^{eT}} \delta s^e + u^{sT} \frac{\partial \dot{q}^s}{\partial \dot{s}^{eT}} \delta s^e = 0 \quad (3.135)$$

that takes the generic form

$$\begin{aligned} \Phi^{eT} \delta s^e &= 0 & (3.136) \\ \text{with } \Phi^{eT} &= \frac{d}{dt} \left(\frac{\partial W_m^*}{\partial \dot{s}^e} \right)^T + \left(\frac{\partial W_e}{\partial s^e} \right)^T + \underbrace{u^{rT} \frac{\partial \dot{q}^r}{\partial \dot{s}^{eT}} + u^{sT} \frac{\partial \dot{q}^s}{\partial \dot{s}^{eT}}}_{\triangleq -Q^{eT}(\dot{s}^e, u^{su})} \end{aligned}$$

where Q^e has been introduced. By analogy with the generalized mechanical forces Q^{nc} associated with non-conservative forces, we will call Q^e the *generalized voltages*. Like the generalized forces Q^{nc} in mechanics, it corresponds to non-conservative elements: the resistors and sources.

Equations 3.136 is valid for any virtual charge variations δs^e compatible with Kirchoff's laws. Hence, we must have $\Phi^e = 0$, and thus

$$\frac{d}{dt} \left(\frac{\partial W_m^*}{\partial \dot{s}^e} \right) + \left(\frac{\partial W_e}{\partial s^e} \right) - Q^e(\dot{s}^e, u^{su}) = 0 \quad (3.137)$$

Note: In the linear case, the energies take the form given in 3.111 and the dynamic equations 3.137 become

$$\left(\frac{d}{dt} (L\dot{q}^i + \varphi^p) \right) - Q^e(\dot{s}^e, u^{su}) = 0 \quad (3.138)$$

Lagrange equations Defining L^e as

$$L^e(s^e, \dot{s}^e) = W_m^*(\dot{s}^e) - W_e(s^e) \quad (3.139)$$

equation 3.137 becomes

$$\frac{d}{dt} \left(\frac{\partial L^e}{\partial \dot{s}^e} \right) - \frac{\partial L^e}{\partial s^e} - Q^e = 0 \quad (3.140)$$

Equation 3.140 has the form of Lagrange equations defined in mechanics (see 3.90). They are the *Lagrange equations of electrical circuits*, and L^e can be called the *Lagrangian* for an electrical circuit.

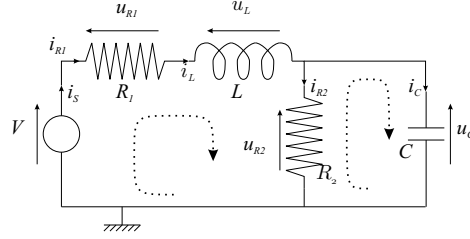


Figure 3.10: Example circuit: RLC filter

Example For the RLC filter of figure 3.10, we have two d.o.f. (one inductor and one capacitor). The generalized coordinates are chosen as:

$$s^e = \begin{pmatrix} q_L \\ q_C \end{pmatrix} \quad (3.141)$$

Kirchoff's equations write:

$$\begin{aligned} \dot{q}_S &= \dot{q}_{R1} \\ \dot{q}_{R1} &= \dot{q}_L \\ \dot{q}_L &= \dot{q}_{R2} + \dot{q}_C \\ V &= u_{R1} + u_L + u_{R2} \\ u_{R2} &= u_C \end{aligned} \quad (3.142)$$

The magnetic co-energy and electric energy can be computed from 3.111:

$$\begin{aligned} W_m^* &= \frac{1}{2} L i_L^2 \\ W_e &= \frac{1}{2} \frac{q_C^2}{C} \end{aligned} \quad (3.143)$$

According to 3.137 and Kirchoff's laws 3.142, the dynamic equations can be written:

$$\begin{aligned} \frac{d}{dt} (L \dot{q}_L) + 0 + u_{R1} + u_{R2} + u_S &= 0 \\ 0 + \frac{q_C}{C} + 0 - u_{R2} + 0 &= 0 \end{aligned} \quad (3.144)$$

Using Kirchoff's voltage law to express these equations in terms of the generalized variables, we get

$$\begin{aligned} L \ddot{q}_L + R_1 \dot{q}_L + \frac{q_C}{C} - V &= 0 \\ \frac{q_C}{C} - R_2 (\dot{q}_L - \dot{q}_C) &= 0 \end{aligned} \quad (3.145)$$

Note: The minus sign appearing in front of the voltage source value V comes from the fact that in the Virtual Work Principle, no distinction is done between active and passive elements concerning the sign convention. All the elements have the same sign convention. Thus, u_S , which represent the source value, is equal to $-V$ because, as seen in figure 3.10, when writing Kirchoff's equations, a generator sign convention is used for the source, while receptor sign convention is used for the other elements.

The dynamic equations of electrical circuit are classically expressed in terms of currents and voltages. For the considered circuit example, using $\dot{q}_L = i_L$ and $\frac{q_C}{C} = u_C$, we get in matrix form

$$\begin{pmatrix} L & 0 \\ 0 & R_2 C \end{pmatrix} \begin{pmatrix} \frac{di_L}{dt} \\ \frac{du_C}{dt} \end{pmatrix} + \begin{pmatrix} R_1 & 1 \\ -R_2 & 1 \end{pmatrix} \begin{pmatrix} i_L \\ u_C \end{pmatrix} + \begin{pmatrix} -V \\ 0 \end{pmatrix} = 0 \quad (3.146)$$

which is the classical dynamic model of the RLC filter.

3.3.2.5 Constraints Consideration for Charges Formulation

Some circuits have a structure such that purely algebraic equations exist between the chosen generalized variables s^e . These constraints come from the Kirchoff's equations and arise in two situations:

- Loops of capacitors and voltage sources

$$K_{1c}^T u^c + K_{1su}^T u^{su} = K_{1c}^T u^c(q^c) + K_{1su}^T u^{su} = 0 \quad (3.147)$$

- Cutsets of inductors, capacitors and current sources

$$K_{2l}^T \dot{q}^l + K_{2c}^T \dot{q}^c + K_{2si}^T \dot{q}^{si} = 0 \quad (3.148)$$

Here, a parallel can be done with the mechanical concepts of *implicit* and *explicit* constraints. Indeed, we have chosen q^l and q^c as generalized variables. This choice led us to the Lagrange equations for electrical circuits, by assuming that a set of algebraic relations, coming from Kirchoff's equations and the constitutive equations for resistors and sources, was satisfied. This is a set of *implicit* constraints between the electrical variables, allowing us to express them in terms of the generalized variables. This is similar to the implicit mechanical constraints relating the absolute positions and orientations to the generalized joint variables s^m .

The new set of algebraic relations 3.147 and 3.148 comes from the same Kirchoff's equations and constitutive equations but only appears with specific structure of the electrical circuit. They can be considered as *explicit constraints* on the generalized variables. Once again, they are similar to the explicit kinematic constraints existing for closed-loop multibody systems.

Because of these explicit constraints, the generalized variables s^e are not independent anymore and the number of d.o.f. is lower than $l + c$.

Deriving 3.147 with respect to time, the constraints may be rewritten as:

$$J^e \dot{s}^e = f(t) \quad (3.149)$$

where J^e is the Jacobian of the constraints and $f(t)$ explicitly takes into account the variation of sources value with respect to time t .

Similarly to what happens in mechanics, a supplementary term has to be considered in equation 3.137, which becomes:

$$\frac{d}{dt} \left(\frac{\partial W_m^*}{\partial \dot{s}^e} \right) + \left(\frac{\partial W_e}{\partial s^e} \right) - Q^e = J^{eT} \lambda^e \quad (3.150)$$

This can be proved from a purely mathematical reasoning identical to the one proposed by Willems [24, 76] for multibody systems. Indeed, when deriving the equations of motion 3.137, the virtual variations in charges δq have been assumed compatible with the implicit constraints. Assuming now that the generalized virtual charges δs^e are compatible with the explicit constraints, we can write:

$$J^e \delta s^e = 0 \quad (3.151)$$

Separating the variables into dependent s_v^e and independent ones s_u^e , we can apply the same mathematical reasoning proposed by Willems [76] for mechanical systems.

From a physical point of view, the $J^{eT} \lambda^e$ have the dimension of a voltage drop. Physical insight of the λ^e will be discussed in the following example.

Example Figure 3.11 shows a simple circuit which obviously involves a constraint since two inductors are serially connected.

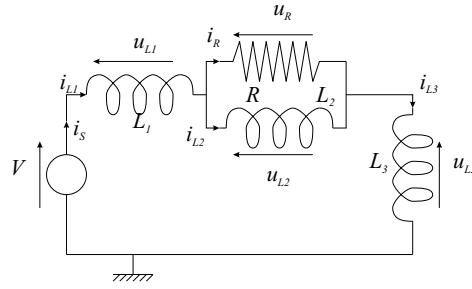


Figure 3.11: Example of circuit with constraint

The generalized variables are

$$s^e = \begin{pmatrix} q_{L1} \\ q_{L2} \\ q_{L3} \end{pmatrix} \quad (3.152)$$

From 3.111, the magnetic co-energy W_m^* can be easily computed as

$$W_m^* = \frac{1}{2} L_1 \dot{q}_{L1}^2 + \frac{1}{2} L_2 \dot{q}_{L2}^2 + \frac{1}{2} L_3 \dot{q}_{L3}^2 \quad (3.153)$$

and the electrical energy W_e equals 0 since the system does not contain capacitors.

Kirchoff's equations are easily obtained for such a system:

$$\begin{aligned} \dot{q}_{L1} &= \dot{q}_S \\ \dot{q}_{L1} &= \dot{q}_R + \dot{q}_{L2} \\ \dot{q}_{L3} &= \dot{q}_{L1} \\ V &= u_{L1} + u_{L2} + u_{L3} \\ u_{L2} &= u_R \end{aligned} \quad (3.154)$$

The third current equation corresponds to a constraint between the generalized variables

$$\dot{q}_{L3} - \dot{q}_{L1} = 0 \quad (3.155)$$

The corresponding Jacobian is

$$J^e = \begin{pmatrix} -1 & 0 & 1 \end{pmatrix} \quad (3.156)$$

The equations of motion 3.150 becomes for this circuit

$$\begin{aligned} L_1 \ddot{q}_{L1} - V + R(\dot{q}_{L1} - \dot{q}_{L2}) &= -\lambda^e \\ L_2 \ddot{q}_{L2} - R(\dot{q}_{L1} - \dot{q}_{L2}) &= 0 \\ L_3 \ddot{q}_{L3} &= \lambda^e \end{aligned} \quad (3.157)$$

Together with the constraint 3.155, equations 3.157 constitute the set of equations to be solved for simulating the behavior of the circuit.

The Lagrange multiplier λ^e can be eliminated and the constraint replaced and one get the final equations of motion

$$\begin{aligned} (L_1 + L_3) \ddot{q}_{L3} - V + R(\dot{q}_{L3} - \dot{q}_{L2}) &= 0 \\ L_2 \ddot{q}_{L2} - R(\dot{q}_{L3} - \dot{q}_{L2}) &= 0 \end{aligned} \quad (3.158)$$

if q_{L2} and q_{L3} are chosen as independent generalized variables, which can be rewritten as

$$\begin{aligned} (L_1 + L_3) \frac{di_{L3}}{dt} - V + R(i_{L3} - i_{L2}) &= 0 \\ L_2 \frac{di_{L2}}{dt} - R(i_{L3} - i_{L2}) &= 0 \end{aligned} \quad (3.159)$$

if we use the classical electrical variables i_{L2} and i_{L3} .

In this example, the Lagrange multiplier $\lambda^e = L_3 \ddot{q}_{L3}$ represents the voltage drop across inductor L_3 . Replacing L_3 by a voltage source of value λ^e transform the initial system into a non-constrained electrical circuit.

Note: This interpretation of the Lagrange multipliers is possible for constraints arising from Kirchoff's current laws. When considering a loop of capacitors and voltage sources, the associated Lagrange multiplier will have the dimensions of a current and must be equal to 0. Similar situations exist in mechanics when considering the contact between two rigid surfaces [15].

3.3.2.6 Flux Formulation

This is the dual case of the formulation in terms of virtual variations in charge.

Starting from the formulation with variations in terms of fluxes 3.114, we can separate different terms:

$$i^T \delta\varphi = i^l{}^T \delta\varphi^l + i^c{}^T \delta\varphi^c + i^r{}^T \delta\varphi^r + i^s{}^T \delta\varphi^s \quad (3.160)$$

Following a similar strategy as for the charges formulation, the two first terms of 3.160 may be rewritten as:

$$\begin{aligned} i^{cT} \delta\varphi^c &= \frac{d}{dt} \left(q^{cT} \delta\varphi^c \right) - \left(\frac{\partial W_e^*}{\partial u^c} \right)^T \delta u^c \\ i^{lT} \delta\varphi^l &= \left(\frac{\partial W_m}{\partial \varphi^l} \right)^T \delta\varphi^l \end{aligned} \quad (3.161)$$

The generalized coordinates that we chose are the flux associated with inductors φ^l and capacitors φ^c :

$$s^e = \begin{pmatrix} \varphi^l \\ \varphi^c \end{pmatrix} \quad (3.162)$$

Applying manipulations similar to those applied for the charges formulation, we obtain the final dynamic equation as:

$$\frac{d}{dt} \left(\frac{\partial W_e^*}{\partial \dot{s}^e} \right) + \left(\frac{\partial W_m}{\partial s^e} \right) + \underbrace{\left(\frac{\partial \dot{\varphi}^r}{\partial \dot{s}^{eT}} \right)^T i^r + \left(\frac{\partial \dot{\varphi}^s}{\partial \dot{s}^{eT}} \right)^T i^s}_{\triangleq -I^e(\dot{s}^e, i^{si})} = 0 \quad (3.163)$$

where I^e is defined as the *generalized currents* by analogy with the generalized voltages Q^e .

Note: In the linear case, equation 3.163 becomes,

$$\left(\frac{d}{dt} (C\dot{\varphi}^c) \right) + I^e \left(\dot{s}^e, i^{si} \right) = 0 \quad (3.164)$$

Defining $L_{bis}^e \triangleq W_e^* - W_m$, we get the Lagrange equations for electrical circuits in terms of flux variations:

$$\frac{d}{dt} \left(\frac{\partial L_{bis}^e}{\partial \dot{s}^e} \right) - \frac{\partial L_{bis}^e}{\partial s^e} - I^e(\dot{s}^e, i^{si}) = 0 \quad (3.165)$$

Constraints can be considered similarly to what was done for the charges formulation. The Lagrange multipliers λ^e , associated with the flux formulation, have the dimensions of a current.

Example Applied to the RLC filter from figure 3.10, this flux formulation leads to

$$\begin{array}{ccccccc} \frac{d}{dt} (C\dot{\varphi}_C) & + & 0 & - & i_{R1} & + & i_{R2} & + & 0 & = & 0 \\ 0 & & \frac{\varphi_L}{L} & - & i_{R1} & + & 0 & + & 0 & = & 0 \end{array} \quad (3.166)$$

Using Kirchoff's current law to express these equations in terms of the generalized variables, we get

$$\begin{aligned} C\ddot{\varphi}_C + \left(\frac{1}{R_1} + \frac{1}{R_2} \right) \dot{\varphi}_C + \frac{\dot{\varphi}_L}{R_1} - \frac{V}{R_1} &= 0 \\ \frac{\varphi_L}{L} + \frac{\dot{\varphi}_C}{R_1} + \frac{\dot{\varphi}_L}{R_1} - \frac{V}{R_1} &= 0 \end{aligned} \quad (3.167)$$

Using $\dot{\varphi}_C = u_C$ and $\frac{\varphi_L}{L} = i_L$, we get in matrix form

$$\begin{pmatrix} \frac{L}{R_1} & 0 \\ \frac{L}{R_1} & C \end{pmatrix} \begin{pmatrix} \frac{di_L}{dt} \\ \frac{du_C}{dt} \end{pmatrix} + \begin{pmatrix} 1 & \frac{1}{R_1} \\ 0 & \frac{1}{R_1} + \frac{1}{R_2} \end{pmatrix} \begin{pmatrix} i_L \\ u_C \end{pmatrix} + \begin{pmatrix} \frac{-V}{R_1} \\ \frac{-V}{R_1} \end{pmatrix} = 0 \quad (3.168)$$

which is the classical dynamic model of the RLC circuit (equivalent to equations 3.146 obtained with the charge formulation).

3.3.2.7 Discussion

We have explicitly developed the Virtual Work Principle on electrical systems in order to draw a parallel with the mechanical virtual work developments at different stages of the computations:

- Similar concepts of *implicit* and *explicit* constraints exist in mechanics as well as when considering electrical circuits
- From a computation point of view, the manipulations are very similar and based on the same methods. This can be observed when obtaining the generalized terms associated with the Lagrange multipliers λ^e and the Jacobian J^e .
- Generalized forces Q^{nc} are equivalent to generalized voltages Q^e and generalized currents I^e .
- The final equations can be written under Lagrange form in both cases, allowing for an automatic unification of the equations when considering electromechanical systems, as will be seen in the following section.

Nevertheless, the Virtual Work Principle, applied to electrical circuits, suffers from several disadvantages:

- First, physical insights is lost when analytically defining charges q_k for every element in the circuit.
- Second, in the case of electrical systems, the virtual work equations are expressed in terms of non measurable quantities (charges or fluxes). It is much more easier to consider voltages and currents when working with electrical circuits, and we showed with the RLC example that both Virtual Work Principles, in terms of charges and fluxes, lead to the same equations in terms of voltage drops across capacitors u^c and currents through inductors i^l ; which are classically used in circuit theory.
- Third, this technique involves multiple summations and partial derivatives. Consequently, like in mechanics, its efficiency can be called into question when considering circuits with many elements.
- A last practical disadvantage of the Virtual Work Principle is related to the order of the generated equations. For the charge formulation presented in details here,

the capacitor charges q^c only appear with their first derivative while the inductor charges q^l appear with their second time derivative (see equation 3.138). For the flux formulation, the dual case exist with the fluxes of inductors φ^l appearing with their first derivative and the fluxes of the capacitors φ^c with their second derivative. It is of course possible to predict, by examination of the system, the order of the final equations and the latter can be systematically rewritten as first order differential equations in terms of more classical variables u^c and i^l , but this requires supplementary manipulations of constitutive equations and Kirchoff's laws, which were already exploited. The same equations are used twice.

Let us mention that other authors, like Schlacher [64], formulate the electrical equations on the basis of power functions an obtain an interesting formulation in terms of classical variables u^c and i^l but they do not get the Lagrange form of the equations and electromechanical interaction is not so obvious.

3.3.3 Virtual Work for Electromechanical Systems

Virtual work theory applied to multibody systems and electrical circuits is summarized in table 3.4, under the form of Lagrange equations. A direct analogy exists between both fields of application.

	Multibody systems	Electrical Circuits
Generalized coordinates	s^m (e.g. joint coordinates)	$s^e = \begin{pmatrix} q^l \\ q^c \end{pmatrix}$
Unconstrained systems	$\frac{d}{dt} \left(\frac{\partial L^m}{\partial \dot{s}^m} \right) - \frac{\partial L^m}{\partial s^m} - Q^{nc} = 0$ with $L^m = T - U$	$\frac{d}{dt} \left(\frac{\partial L^e}{\partial \dot{s}^e} \right) - \left(\frac{\partial L^e}{\partial s^e} \right) - Q^e = 0$ with $L^e = W_m^* - W_e$
Constrained systems	$\frac{d}{dt} \left(\frac{\partial L^m}{\partial \dot{s}^m} \right) - \frac{\partial L^m}{\partial s^m} - Q^{nc} = J^{mT} \lambda^m$	$\frac{d}{dt} \left(\frac{\partial L^e}{\partial \dot{s}^e} \right) - \left(\frac{\partial L^e}{\partial s^e} \right) - Q^e = J^{eT} \lambda^e$

Table 3.4: Summary of the mechanical and electrical Lagrange equations

When electromechanical systems are considered, we assemble both formalisms by first defining the generalized variables as:

$$s = \begin{pmatrix} s^m \\ s^e \end{pmatrix} \quad (3.169)$$

Using developments similar to those presented before, we get the following equations:

$$\frac{d}{dt} \left(\frac{\partial L^{em}}{\partial \dot{s}} \right) - \left(\frac{\partial L^{em}}{\partial s} \right) - \Gamma = 0 \quad (3.170)$$

with the *electromechanical Lagrangian*: $L^{em} = L^m + L^e = T + W_m^* - U - W_e$ and the generalized terms as $\Gamma = Q^{nc} + Q^e$.

The Lagrange equations of electromechanical systems can be separated into two subsets of equations:

$$\frac{d}{dt} \left(\frac{\partial L^{em}}{\partial \dot{s}^m} \right) - \left(\frac{\partial L^{em}}{\partial s^m} \right) - \Gamma_m = 0 \quad (\text{mechanical equations}) \quad (3.171)$$

$$\frac{d}{dt} \left(\frac{\partial L^{em}}{\partial \dot{s}^e} \right) - \left(\frac{\partial L^{em}}{\partial s^e} \right) - \Gamma_e = 0 \quad (\text{electrical equations}) \quad (3.172)$$

Looking at the electrical equations and at the definition of the mechanical Lagrangian $L^m = T - U$ and of the kinetic and potential energies, it is obvious that the Lagrangian L^m will never be functions of the electrical generalized variables s^e , and the above electrical equations reduce to 3.137.

In the case of linear constitutive equations, equations 3.138 apply with the electrical parameters function of the mechanical positions and velocities: $u^{su}(t, s^m, \dot{s}^m)$, $i^{si}(t, s^m, \dot{s}^m)$, $R(s^m)$, $L(s^m)$ and $C(s^m)$. For example, the first time derivatives of the inductance matrix L is non zero and can be related to the mechanical velocities \dot{s}^m :

$$\frac{dL}{dt} = \left(\frac{\partial L}{\partial s^m} \right)^T \dot{s}^m \quad (3.173)$$

Looking at the mechanical equations, assuming that these dependencies are purely at position level: $W_m^* = W_m^*(s^m, t)$ and $W_e = W_e(s^m, t)$, supplementary terms in the mechanical equations take the form of electromechanical generalized forces Q^{em}

$$Q^{em} = \frac{\partial W_m^*}{\partial s^m} - \frac{\partial W_e}{\partial s^m} \quad (\text{Charge formulation}) \quad (3.174)$$

From a mechanical point of view, the electromechanical converter can be considered as an additional force source, similar to a spring or a damper, characterized by a more complex constitutive equation.

Note: Associated with the flux formulations are the derivative of the capacitance matrix $\frac{dC}{dt}$ and the generalized electromechanical forces

$$Q^{em} = \frac{\partial W_e^*}{\partial s^m} - \frac{\partial W_m}{\partial s^m} \quad (\text{Flux formulation}) \quad (3.175)$$

where $s^e = \begin{pmatrix} \varphi^l \\ \varphi^c \end{pmatrix}$.

This shows us that the electromechanical interaction taking place between the mechanical and the electrical subsystems does not affect the structure of the equations describing their behavior. The latter can thus be written independently and properly coupled afterwards. This will be discussed in more details in the second part of this text.

In [26], Hadwich and Pfeiffer did propose to consider the electromechanical interaction as a supplementary constraint between the mechanical and electrical domain. In their formalism, the electrical parameters, for example C , are considered as supplementary generalized variables and their dependency with respect to the mechanical

variables is expressed as a constraint, for example $C = C(s^m)$ where the capacitor C is part of the generalized variables. From our point of view, considering electrical parameters as function of other variables is not a problem and the technique proposed by Hadwich and Pfeiffer seems artificial.

Example: Condensator speaker The Virtual Work Principle can be applied to obtain the dynamic equations for the condensator speaker of figure 3.1.

Looking at the mechanical subsystem, the kinematic energy T and the potential energy V are:

$$\begin{aligned} T &= \frac{m_5 \dot{x}^2}{2} \\ V &= \frac{k_6 (r_0 + x - l_6)^2}{2} \end{aligned} \quad (3.176)$$

Assuming that the spring is unstretched when $x = 0$, the potential energy becomes:

$$V = \frac{k_6 x^2}{2} \quad (3.177)$$

The mechanical generalized forces are given by:

$$Q_m = d_6 \dot{x} + m_5 g \quad (3.178)$$

Looking at the electrical subsystem, the magnetic co-energy W_m^* and the electrical energy W_e are:

$$\begin{aligned} W_m^* &= \frac{L_3 \dot{q}_L}{2} \\ W_e &= \frac{q_C^2}{2C_2(x)} \end{aligned} \quad (3.179)$$

The electrical generalized voltages are given by:

$$Q_e = -u_{R1} + E_4(t) = -R_1 i_L + E_4(t) \quad (3.180)$$

We have a constraint between the electrical variables because of the serial connection of the inductor and the capacitor:

$$\dot{q}_L = \dot{q}_C \quad (3.181)$$

Combining all these terms and considering the generalized variables

$$s^T = (x \quad q_L \quad q_C) \quad (3.182)$$

the equations of motion are:

$$\begin{cases} \frac{d}{dt} (m_5 \dot{x}) + k_6 x - \frac{q_C^2}{2C_2^2(x)} \frac{dC_2(x)}{dx} = -d_6 \dot{x} - m_5 g \\ \frac{d}{dt} (L_3 i_L) = -R_1 i_L + E_4(t) + \lambda \\ \frac{q_C}{C_2} = 0 - \lambda \\ \dot{q}_L = \dot{q}_C \end{cases} \quad (3.183)$$

$$\Leftrightarrow \begin{cases} \frac{d}{dt} (m_5 \dot{x}) + k_6 x - \frac{q_C^2}{2C_2^2(x)} \frac{dC_2(x)}{dx} = -d_6 \dot{x} - m_5 g \\ \frac{d}{dt} (L_3 i_L) + \frac{q_C}{C_2} = -R_1 i_L + E_4(t) \\ \dot{q}_L = \dot{q}_C \end{cases} \quad (3.184)$$

One can get the final equations of motion for the global system as:

$$\begin{cases} \frac{d}{dt} (m_5 \dot{x}) + k_6 x + d_6 \dot{x} - \frac{1}{2} \frac{dC_2(x)}{dx} (E_4(t) - R_{L1} \cdot i_L - \frac{d}{dt} (L_3 i_L))^2 = -m_5 g \\ i_L = \frac{dC_2}{dx} \frac{dx}{dt} \left(-\frac{d}{dt} (L_3 i_L) - R_{L1} i_L + E_4(t) \right) \\ \quad + C_2(x) \frac{d}{dt} \left(-\frac{d}{dt} (L_3 i_L) - R_{L1} i_L + E_4(t) \right) \end{cases} \quad (3.185)$$

These equations are equivalent to 3.6 and 3.18 respectively obtained with the bond graph and linear graph approaches.

3.4 Conclusions

Several unified theories have been proposed and discussed on the basis of a simple example, for which all approaches are adequate. Nevertheless, as soon as large and complex systems are considered, the unified theories suffer from the concept of one formulation for all the fields of physics and are either not applicable, or inefficient:

- Bond graphs would require much more developments in order to be applied to 3D multibody dynamics: constraints at velocity level, absolute coordinates, frame dependent vectors, complex structure, causality assignment, etc. are the main issues.
- Linear graph have shown interesting features both for electrical and mechanical multibody systems, but when considering large systems, its efficiency can be called into question, with respect to intrinsically recursive approaches¹⁸.
- Virtual Work Principle provides an interesting framework for constraint consideration and reduction but still remains inefficient because of the multiple summations and partial derivatives involve in the computations. This is especially true when modeling large MBS as in this work. Moreover, physical insight is lost when considering electrical circuits.

Let us point out that although reduction procedures from DAEs to ODEs can be applied to the equations obtained with the different unified approaches, we believe that equational approaches, such as the Virtual Work Principle, are more naturally followed by these additional manipulations of the equations.

The in depth confrontation presented here lead us to a dead-end since neither block diagrams nor strategies based on modeling languages or unified theories are fully satisfactory for building multidomain models for the applications aimed in this research. Therefore, in the second part of this thesis, we will propose a new strategy applicable to large and complex electromechanical multibody systems.

It was nevertheless a good learning experience and interesting to deeply investigate different approaches. This type of work is currently rare in literature.

¹⁸Although Linear graph theory does not preclude it, as far as we know, current implementations do not consider recursive formulation.

Part II

Proposed Symbolic Modeling Approach

Abstract

In the first part of this thesis, we have presented and discussed most of the existing theories and strategies for modeling electromechanical systems and we highlighted the disadvantages of these techniques when dealing with large and complex multibody systems coupled with electromechanical converters. From our point of view, today's techniques are not fully satisfactory for modeling such systems efficiently.

In the second part of this thesis, we therefore propose a new approach in which the models for the different parts of the system are generated separately and symbolically by means of optimal dedicated formalism, and coupled afterwards in a global symbolic model.

4

Formalisms for Electromechanical Systems

Abstract

In section 3.3.3, we have shown that the electromechanical interaction does not affect the structure of the dynamic equations describing the behavior of the electrical and mechanical submodels, which may thus be generated separately. Nevertheless, their coupling have to be achieved rigorously and evaluation of the equations has to be done according to a specific sequence.

In this chapter, we will develop the dedicated formalism that were chosen for their efficiency when implemented symbolically:

- For mechanical multibody systems, we use the well-known Newton-Euler Recursive formalism [40]. This formalism is based on the Newton and Euler equations, but the use of joint coordinates and recursive procedures makes it more efficient than classical techniques based on absolute coordinates, when large systems (more than 5 to 7 d.o.f.) are considered. Moreover it is well suited to symbolic implementation [16]. The mechanical coordinate partitioning technique [75] will also be presented.
- For electrical circuits, we propose a formalism based on circuit theory, allowing for automatic analytical detection of the constraints, which can be automatically eliminated, using an “electrical coordinate partitioning” method inspired from the mechanical coordinate partitioning [75].

The coupling of the models obtained with these formalisms is discussed in the last section of this chapter.

4.1 MBS - Newton-Euler Recursive Formalism

The Newton-Euler recursive (NER) formalism [40] is one of the most efficient techniques to generate the equations of motion of multibody systems (MBS), when considering applications with many degrees of freedom (d.o.f.). Together with a recursive formulation, the use of relative joint coordinates makes it more efficient than Newton-Euler formalisms expressed in terms of absolute coordinates, although the equations are more complex [16].

The equations resulting from the Newton-Euler Recursive formalism are identical to 3.86 obtained from the Virtual Work Principle for unconstrained system [16], meaning that the Lagrange multiplier technique can still be used afterwards for expressing the constraints generalized forces and thus for constraints elimination.

The Newton-Euler recursive formalism can be used to obtain the *semi-explicit* form of the equations for tree-like multibody systems [56]:

$$M(s^m)\ddot{s}^m + c(s^m, \dot{s}^m, F_{ext}, L_{ext}, g) = Q \quad (4.1)$$

which is strictly equivalent to equation 3.86, but whose computer generation of the left-hand term is far more straightforward, thanks to the use of recursive computations and joint coordinates. It uses forward and backward recursive computations, as illustrated in figure 4.1:

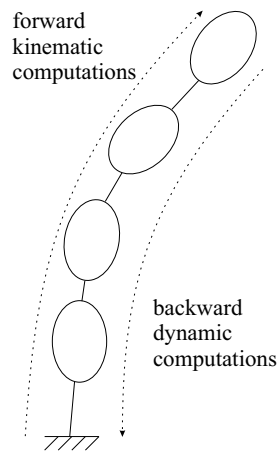


Figure 4.1: Forward kinematics and backward dynamics

- a forward kinematics computation of the position, velocity and acceleration vectors, is conducted from the root of the tree to the leaf bodies,
- a backward dynamics computation of the forces and torques on each body, is conducted from the leaf bodies to the root.

A particular procedure must be achieved to extract the system mass matrix recursively [16].

4.1.1 Forward kinematics

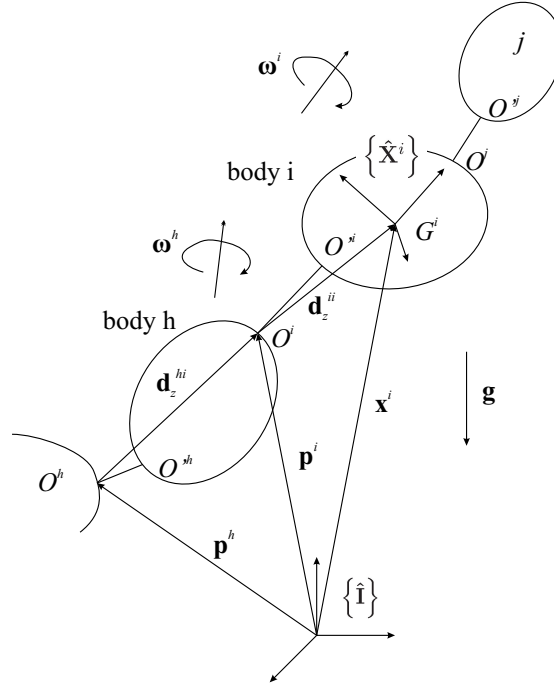


Figure 4.2: Forward kinematics

Let us consider¹, in figure 4.2, a rigid body i carried by parent body h via joint i . For body i , we can write:

- Absolute position vectors:

$$\begin{aligned} \mathbf{p}^i &= \mathbf{p}^h + \mathbf{d}_z^{hi}, \text{ for the attach point } O^i \text{ on body } h, \\ \mathbf{x}^i &= \mathbf{p}^i + \mathbf{d}_z^{ii}, \text{ for the center of mass of body } i, \end{aligned} \quad (4.2)$$

- Absolute velocities vector:

- angular

$$\boldsymbol{\omega}^i = \boldsymbol{\omega}^h + \boldsymbol{\Omega}^i = \boldsymbol{\omega}^h + \boldsymbol{\varphi}^i \dot{s}_i^m \quad (4.3)$$

- linear

$$\begin{aligned} \dot{\mathbf{p}}^i &= \dot{\mathbf{p}}^h + \tilde{\boldsymbol{\omega}}^h \cdot \mathbf{d}_z^{hi} + \boldsymbol{\psi}^h \dot{s}_h^m \\ \dot{\mathbf{x}}^i &= \dot{\mathbf{p}}^i + \tilde{\boldsymbol{\omega}}^i \cdot \mathbf{d}_z^{ii} + \boldsymbol{\psi}^i \dot{s}_i^m \end{aligned} \quad (4.4)$$

¹All the quantities represented in this figure have been defined in chapter 2.

- Absolute accelerations vector:

- angular

$$\dot{\omega}^i = \dot{\omega}^h + \tilde{\omega}^i \cdot \varphi^i \dot{s}_i^m + \varphi^i \ddot{s}_i^m \quad (4.5)$$

- linear

$$\begin{aligned} \ddot{\mathbf{p}}^i &= \ddot{\mathbf{p}}^h + \left(\tilde{\omega}^h + \tilde{\omega}^h \cdot \tilde{\omega}^h \right) \cdot \mathbf{d}_z^{hi} + 2\tilde{\omega}^h \cdot \boldsymbol{\psi}^h \dot{s}_h^m + \boldsymbol{\psi}^h \ddot{s}_h^m \\ \ddot{\mathbf{x}}^i &= \ddot{\mathbf{p}}^i + \left(\tilde{\omega}^i + \tilde{\omega}^i \cdot \tilde{\omega}^i \right) \cdot \mathbf{d}_z^{ii} + 2\tilde{\omega}^i \cdot \boldsymbol{\psi}^i \dot{s}_i^m + \boldsymbol{\psi}^i \ddot{s}_i^m \end{aligned} \quad (4.6)$$

In order to obtain the dynamic equations in a compact form, it is convenient to define the following quantities:

$$\begin{aligned} \boldsymbol{\beta}^i &\triangleq \tilde{\omega}^i + \tilde{\omega}^i \cdot \tilde{\omega}^i \\ \boldsymbol{\alpha}^i &\triangleq \ddot{\mathbf{p}}^i + 2\tilde{\omega}^i \cdot \boldsymbol{\psi}^i \dot{s}_i^m + \boldsymbol{\psi}^i \ddot{s}_i^m - \mathbf{g} \end{aligned} \quad (4.7)$$

where \mathbf{g} denotes the gravity vector.

Using 4.6, we can write the second relation of 4.7 in a recursive manner:

$$\boldsymbol{\alpha}^i = \boldsymbol{\alpha}^h + \boldsymbol{\beta}^h \cdot \mathbf{d}_z^{hi} + 2\tilde{\omega}^i \cdot \boldsymbol{\psi}^i \dot{s}_i^m + \boldsymbol{\psi}^i \ddot{s}_i^m \quad (4.8)$$

Note that, from 2.1 and 2.19, \mathbf{d}_z^{hi} is related to the joint coordinate s_i^m as follows

$$\mathbf{d}_z^{hi} = \mathbf{d}^{hi} + s_i^m \boldsymbol{\psi}^i \quad (4.9)$$

In order to express the mass matrix in the semi-explicit form 4.1, we need to isolate the generalized accelerations \ddot{s}^m in the recursive equations 4.5, 4.7 and 4.8 by splitting up the $\dot{\omega}^i$, $\boldsymbol{\beta}^i$ and $\boldsymbol{\alpha}^i$ quantities as follows, since we know from 4.5 and 4.6 that the accelerations \ddot{s}^m will appear linearly in $\dot{\omega}^i$ and $\boldsymbol{\alpha}^i$:

$$\begin{aligned} \dot{\omega}^i &= \sum_{k \leq i} \mathbf{O}_M^{ik} \dot{s}_k^m + \dot{\omega}_c^i \\ \boldsymbol{\beta}^i &= \sum_{k \leq i} \mathbf{B}_M^{ik} \dot{s}_k^m + \boldsymbol{\beta}_c^i \\ \boldsymbol{\alpha}^i &= \sum_{k \leq i} \mathbf{A}_M^{ik} \dot{s}_k^m + \boldsymbol{\alpha}_c^i \end{aligned} \quad (4.10)$$

where $\sum_{k \leq i}$ represents the summation over body i and its ancestors².

The recursive computation of the previous equations can then be performed, starting at the root of the tree, using the following algorithms, where we transformed all the expressions into their matrix forms by expressing all vectors and tensors in their appropriate frame, usually the body-fixed frame³:

²Although inequality signs is used here, it does not refer explicitly to an ordering in the mathematical sense ($i < j$ means that i is ancestor of j).

³For example, $\boldsymbol{\psi}^i = [\hat{\mathbf{X}}^i]^T \boldsymbol{\psi}^i$, $\mathbf{z}^h = [\hat{\mathbf{X}}^h]^T \mathbf{z}^h, \dots$

Initialization

$$\alpha_c^0 = -g; \omega^0 = 0; \dot{\omega}_c^0 = 0; O_M^{ik} = 0; A_M^{ik} = 0 \quad (4.11)$$

($\forall i = 0 : N^{body}, \forall k = 0 : i$)

*Recursion*For $i = 1 : N^{body}$

$$\begin{aligned} h &= \text{inbody}(i) \\ \omega^i &= R^{ih} \omega^h + \varphi^i \dot{s}_i^m \\ \dot{\omega}_c^i &= R^{ih} \dot{\omega}_c^h + \tilde{\omega}^i \varphi^i \dot{s}_i^m \\ \beta_c^i &= \tilde{\omega}_c^i + \tilde{\omega}^i \tilde{\omega}^i \\ \alpha_c^i &= R^{ih} (\alpha_c^h + \beta_c^h d_z^{hi}) + 2\tilde{\omega}^i \psi^i \dot{s}_i^m \end{aligned} \quad (4.12)$$

For $k = 1 : i$

$$\begin{aligned} O_M^{ik} &= R^{ih} O_M^{hk} + \delta^{ki} \varphi^i \\ (B_M^{ik} &= \tilde{O}_M^{ik}) \\ A_M^{ik} &= R^{ih} (A_M^{hk} + \tilde{O}_M^{hk} d_z^{hi}) + \delta^{ki} \psi^i \end{aligned} \quad (4.13)$$

with $\delta^{ki} = 1$ if $k = i$, and 0 otherwise

end

end.

4.1.2 Backward dynamics

According to the Newton equation of motion 2.33 and using the definition 2.30 of the linear momentum \mathbf{N}^i , the translational vector equation of motion of body i is

$$\mathbf{F}^i - \sum_{j \in \bar{i}} \mathbf{F}^j + \mathbf{F}_{ext}^i + m^i \mathbf{g} = m^i \ddot{\mathbf{x}}^i \quad (4.14)$$

where \mathbf{F}^i , defined in section 2.1.5, represents the resultant force acting on body i through joint i , evaluated at the connection point O^i (see figure 4.3), \mathbf{F}_{ext}^i is the external resultant force applied to the center of mass G^i of body i (except the gravity \mathbf{g}), and $\sum_{j \in \bar{i}}$ refers to all the children of body i (j and k in figure 4.3).

Using equation 4.6 and the definitions 4.7, we can write the Newton equation of motion as

$$\mathbf{F}^i = \sum_{j \in \bar{i}} \mathbf{F}^j - \mathbf{F}_{ext}^i + m^i (\boldsymbol{\alpha}^i + \boldsymbol{\beta}^i \cdot \mathbf{d}_z^{ii}) \quad (4.15)$$

or equivalently

$$\begin{aligned} \mathbf{F}^i &= \sum_{j \in \bar{i}} \mathbf{F}^j + \mathbf{W}^i \\ &\text{with } \mathbf{W}^i \triangleq m^i (\boldsymbol{\alpha}^i + \boldsymbol{\beta}^i \cdot \mathbf{d}_z^{ii}) - \mathbf{F}_{ext}^i \end{aligned} \quad (4.16)$$

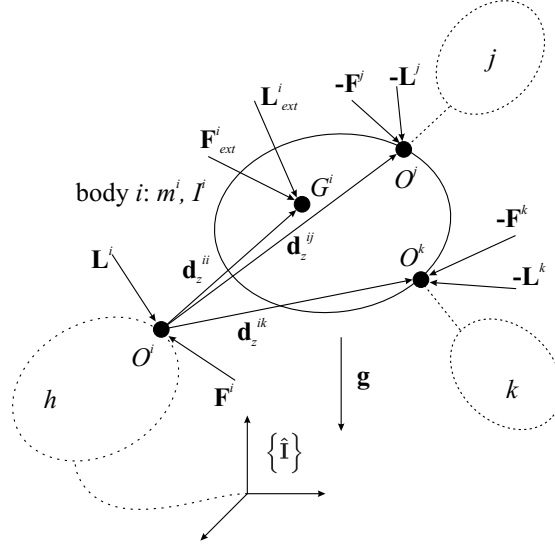


Figure 4.3: Backward Dynamics

Similarly, the Euler rotation equation 2.35 of motion of body i with respect to its center of mass G^i can be written as

$$\mathbf{L}^i = \sum_{j \in \bar{i}} \left(\mathbf{L}^j + \tilde{\mathbf{d}}_z^{ij} \cdot \mathbf{F}^j \right) + \tilde{\mathbf{d}}_z^{ii} \cdot \mathbf{W}^i - \mathbf{L}_{ext}^i + \mathbf{I}^i \cdot \dot{\boldsymbol{\omega}}^i + \tilde{\boldsymbol{\omega}}^i \cdot \mathbf{I}^i \cdot \boldsymbol{\omega}^i \quad (4.17)$$

where, according to 2.22 and 2.19, $\mathbf{d}_z^{ii} = \mathbf{d}^{ii} + s_i^m \boldsymbol{\psi}^i$.

Equations 4.16 and 4.17 can be recursively computed from the leaf bodies to the base (backward computation). In order to get the mass matrix M , we have to isolate the contribution of each generalized acceleration \ddot{s}_k^m :

$$\begin{aligned} \mathbf{F}^i &= \sum_k \mathbf{F}_M^{ik} \ddot{s}_k^m + \mathbf{F}_c^i & (a) \\ \mathbf{W}^i &= \sum_k \mathbf{W}_M^{ik} \ddot{s}_k^m + \mathbf{W}_c^i & (b) \\ \mathbf{L}^i &= \sum_k \mathbf{L}_M^{ik} \ddot{s}_k^m + \mathbf{L}_c^i & (c) \end{aligned} \quad (4.18)$$

By transforming all the equations into matrix form, we get the backward recursive algorithm detailed below. Note that the external forces \mathbf{F}_{ext}^i and torques \mathbf{L}_{ext}^i are assumed to be given by their components in the body-fixed frame $\{\hat{\mathbf{X}}^i\}$.

Recursion

For $i = N^{body} : 1$

$$\begin{aligned} W_c^i &= m^i (\alpha_c^i + \beta_c^i d_z^{ii}) - F_{ext}^i \\ F_c^i &= \sum_{j \in \bar{i}} R^{ij} F_c^j + W_c^i \\ L_c^i &= \sum_{j \in \bar{i}} \left(R^{ij} L_c^j + \widetilde{d}_z^{ij} R^{ij} F_c^j \right) + \widetilde{d}_z^{ii} W_c^i - L_{ext}^i + I^i \dot{\omega}_c^i + \widetilde{\omega}^i I^i \omega^i \end{aligned} \quad (4.19)$$

For $k = 1 : i$

$$\begin{aligned} W_M^{ik} &= m^i (A_M^{ik} + O_M^{ik} d_z^{ii}) \\ F_M^{ik} &= \sum_{j \in \bar{i}} R^{ij} F_M^{jk} + W_M^{ik} \\ L_M^{ik} &= \sum_{j \in \bar{i}} \left(R^{ij} L_M^{jk} + \widetilde{d}_z^{ij} R^{ij} F_M^{jk} \right) + \widetilde{d}_z^{ii} W_M^{ik} + I^i O_M^{ik} \end{aligned} \quad (4.20)$$

end

end.

The i th joint equation is then obtained by projecting the vector equations of motion 4.18 (a) and (c) in the motion space, leading to

$$Q^i = (\psi^i)^T F^i + (\varphi^i)^T L^i \quad (4.21)$$

for the forces and torques, together with

$$\begin{aligned} c^i &= (\psi^i)^T F_c^i + (\varphi^i)^T L_c^i \\ M^{ij} &= (\psi^i)^T F_M^{ij} + (\varphi^i)^T L_M^{ij} \end{aligned} \quad (4.22)$$

for the acceleration terms.

Finally, the i th equation of motion may be written in matrix form as

$$\sum_j M^{ij} \ddot{s}_j^m + c^i = Q^i, \text{ for } i = 1 : N^{body} \quad (4.23)$$

These equations of motion are fully equivalent to those obtained from the Virtual Work Principle 3.86 [16]. Because of this equivalence, we can assert that the generalized mass matrix M is symmetric and positive definite, although the recursive nature of the previous scheme makes it difficult to show. The same equivalence can also be used to justify the use of Lagrange multipliers technique for the system 4.23, when considering constrained systems.

The Newton-Euler recursive formalism was implemented in the Robotran symbolic software as explained in chapter 5.

4.1.3 Coordinate partitioning for MBS systems

When the mechanical subsystem is constrained, according to the Lagrange multiplier theory, the motion equation may be rewritten as 3.97:

$$\begin{aligned} M(s^m)\ddot{s}^m + c(s^m, \dot{s}^m, F_{ext}, L_{ext}, g) &= Q + J^{mT} \lambda^m \\ h(s^m, t) &= 0 \end{aligned} \quad (4.24)$$

where h represents the holonomic constraints while J^m is the Jacobian associated with these constraints. For simplicity of the presentation, we ignore the non-holonomic constraints, although they can easily be introduced, in addition to the holonomic constraint at velocity (\dot{h}) and acceleration (\ddot{h}) levels.

The reduction from 4.24 to a purely ODE system can be obtained by means of the coordinate partitioning method [75, 56]. Assuming that we have n generalized variables s^m and m holonomic constraints, the array of generalized variables s^m can be partitioned⁴ as

$$s^m = \begin{pmatrix} s_u^m \\ s_v^m \end{pmatrix} \quad (4.25)$$

where s_u^m represents the $n - m$ independent variables, while s_v^m represents the m dependent variables.

The Jacobian matrix J^m is also partitioned into

$$J^m = \begin{pmatrix} J_u^m & J_v^m \end{pmatrix} \quad (4.26)$$

where J_v^m is a square m by m matrix. In this work, we assume that the constraints are independent and the matrix J_v^m is non singular⁵.

The m Lagrange multipliers λ^m can be eliminated from the equations 4.24, leaving a system of $n - m$ differential equations in the $n - m$ unknowns s_u^m and their derivatives. For this purpose, the generalized mass matrix M and the vector c are also partitioned:

$$\begin{pmatrix} M_{uu} & M_{uv} \\ M_{vu} & M_{vv} \end{pmatrix} \begin{pmatrix} \ddot{s}_u^m \\ \ddot{s}_v^m \end{pmatrix} + \begin{pmatrix} c_u \\ c_v \end{pmatrix} = \begin{pmatrix} Q_u \\ Q_v \end{pmatrix} + \begin{pmatrix} J_u^{mT} \\ J_v^{mT} \end{pmatrix} \lambda^m \quad (4.27)$$

Since the matrix J_v^m is non singular, one obtains by eliminating the unknowns λ^m from the second set of equations

$$\begin{aligned} \begin{pmatrix} M_{uu} & M_{uv} \end{pmatrix} \begin{pmatrix} \ddot{s}_u^m \\ \ddot{s}_v^m \end{pmatrix} + B_{vu}^T \begin{pmatrix} M_{vu} & M_{vv} \end{pmatrix} \begin{pmatrix} \ddot{s}_u^m \\ \ddot{s}_v^m \end{pmatrix} + c_u + B_{vu}^T c_v \\ = Q_u + B_{vu}^T Q_v \end{aligned} \quad (4.28)$$

where

$$B_{vu} \triangleq -(J_v^m)^{-1} J_u^m \quad (4.29)$$

⁴with possible reordering.

⁵This is the case most of the time, although *singular* configurations may exist for particular structure of the multibody system. In such situations, new sets of dependent and independent variables have to be considered.

Then partitioning the first and second derivatives of the constraints $h(s^m, t) = 0$:

$$\begin{aligned} & \begin{pmatrix} J_u^m & J_v^m \end{pmatrix} \begin{pmatrix} \dot{s}_u^m \\ \dot{s}_v^m \end{pmatrix} + \frac{\partial h}{\partial t} = 0 \\ & \begin{pmatrix} J_u^m & J_v^m \end{pmatrix} \begin{pmatrix} \ddot{s}_u^m \\ \ddot{s}_v^m \end{pmatrix} + \left(\dot{J} + \frac{\partial^2 h}{\partial t \partial s^{mT}} \right) \dot{s}^m + \frac{\partial^2 h}{\partial t^2} = 0 \end{aligned} \quad (4.30)$$

we obtain the generalized velocities and accelerations \dot{s}_v^m and \ddot{s}_v^m respectively expressed in terms of the independent velocities \dot{s}_u^m and accelerations \ddot{s}_u^m :

$$\begin{aligned} \dot{s}_v^m &= B_{vu} \dot{s}_u^m - (J_v^m)^{-1} \frac{\partial h}{\partial t} \\ \ddot{s}_v^m &= B_{vu} \ddot{s}_u^m - (J_v^m)^{-1} b \\ &\text{with } b = \left(\dot{J} + \frac{\partial^2 h}{\partial t \partial s^{mT}} \right) \dot{s}^m + \frac{\partial^2 h}{\partial t^2} \end{aligned} \quad (4.31)$$

The latter can be eliminated from the differential equations of motion, producing the *reduced* system

$$M_{red}(s^m, t) \ddot{s}_u^m + c_{red}(s^m, \dot{s}_u^m, F_{ext}, L_{ext}, g, t) = Q_{red} \quad (4.32)$$

where

$$\begin{aligned} M_{red} &= M_{uu} + M_{uv} B_{vu} + B_{vu}^T M_{vu} + B_{vu}^T M_{vv} B_{vu} \\ c_{red} &= c_u + B_{vu}^T c_v - (M_{uv} + B_{vu}^T M_{vv}) (J_v^{-1}) b \\ Q_{red} &= Q_u + B_{vu}^T Q_v \end{aligned} \quad (4.33)$$

The algebraic constraints still have to be solved in order to eliminate the dependent variables s_v^m from 4.32. Because of the non-linearity of these relations, this cannot be done analytically and a numerical procedure is required. For example the Newton-Raphson algorithm can be used, which requires the Jacobian matrix J^m that can be obtained from recursive computations.

After this numerical elimination, we obtain the set of purely algebraic differential equations as was announced by 3.98:

$$M_r(s_u^m, t) \ddot{s}_u^m + c_r(s_u^m, \dot{s}_u^m, F_{ext}, L_{ext}, g, t) = Q_r \quad (4.34)$$

The flowchart of a numerical integration scheme of system 4.34 is shown in figure 4.4.

In opposition with the classical Newton/Euler procedure, the proposed method does not compute the constraint forces. In some cases, we may be interested in these forces. They can be computed afterwards, knowing the motion of the system, by means of the Lagrange multipliers since they replace the constraint forces in our dynamical model. From the previous elimination scheme, we can isolate the λ^m as

$$\lambda^m = \left((J_v^m)^T \right)^{-1} \left((M_{vu} + M_{vv} B_{vu}) \ddot{s}_u^m + c_v - Q_v - M_{vv} (J_v^m)^{-1} b \right) \quad (4.35)$$

which allows us to compute the constraint forces [56].

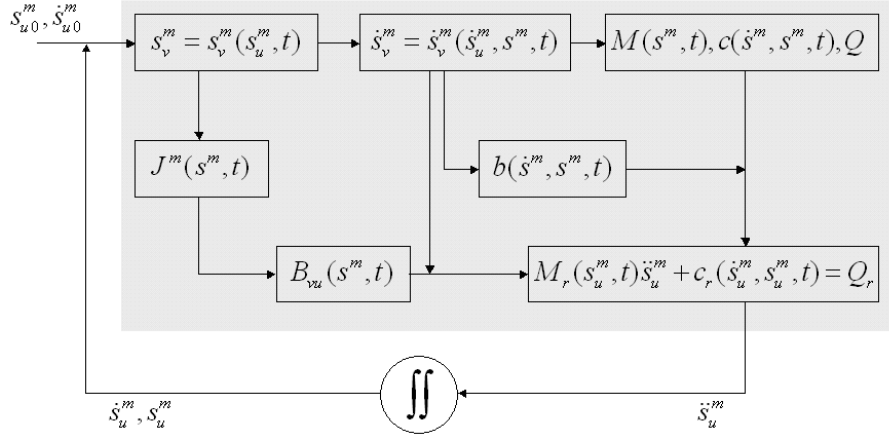


Figure 4.4: Mechanical Coordinate Partitioning and Integration Scheme

This coordinate partitioning procedure was also implemented in Robotran. Although solving the constraints requires a numerical procedure, the latter can be “embedded” within the symbolic model and a “quasi” fully-symbolic⁶ generation of the equations is thus possible [49].

4.2 Electrical Circuit Theory based Formalism

Based on the same starting equations as for the Virtual Work Principle, Kirchoff’s laws and constitutive equations, this circuit theory based formalism will lead to similar equations expressed in terms of voltages and currents, instead of charges or fluxes that suffer from physical insights, as discussed in section 3.3.2.

4.2.1 Unconstrained Circuit

Each element k of the circuit is characterized by the current i_k flowing through it, the voltage drop u_{kk} between its starting and end nodes and a constitutive equation relating these two variables. For a network containing N^{el} elements, $2N^{el}$ variables have to be determined at each step of computation. Part of these variables, more precisely the currents through the inductors i^l and the voltage drops across the capacitors u^c , will appear with their first time-derivative in the constitutive equations of the inductors and capacitors. They will be called the *state variables*⁷, denoted w :

$$w^T = \begin{pmatrix} i^{lT} & u^{cT} \end{pmatrix} \quad (4.36)$$

⁶The Newton-Raphson is a numerical iterative procedure, but the required quantities (Jacobian, constraints, etc.) are computed with symbolic equations and benefit of all the corresponding simplifications.

⁷in the sense of state-space representation of systems

The other variables are the *algebraic variables*, appearing only in purely algebraic relations, and denoted y :

$$y^T = \left(i^{s^T} \quad i^{r^T} \quad u^{s^T} \quad u^{r^T} \quad u^{l^T} \quad i^{c^T} \right) \quad (4.37)$$

For a circuit with s sources, r resistors, l inductors and c capacitors, we will have $N^{el} + s + r$ algebraic variables and $l + c$ state variables.

Considering linear dipoles, the constitutive equations of sources and resistors, given in section 2.2 are algebraic relations of the general form:

$$a_k i_k + b_k u_k = z_k \quad (4.38)$$

$$\text{where, } \begin{cases} a_k = 0, b_k = 1 \text{ for a voltage source with value } z_k \\ a_k = 1, b_k = 0 \text{ for a current source with value } z_k \\ a_k \neq 0, b_k \neq 0 \text{ and } z_k = 0 \text{ for resistor} \end{cases}$$

In this last case, when b_k is set to -1, a_k is equal to the electrical resistance R_k of the resistor.

Together with Kirchoff's laws 2.40 and 2.41, resistances and sources constitutive equations 4.38 form a set of $N^{el} + s + r$ pure algebraic relations between the currents and voltages. They will form the *algebraic system* of equations, expressing the algebraic variables in terms of the state variables:

$$Ay = Bw + Cz \quad (4.39)$$

The linear constitutive equation of inductors is a differential equation, given in section 2.2:

$$u^l = \frac{d(Li^l)}{dt} + \frac{d\varphi^p}{dt} \quad (4.40)$$

Similarly, the linear constitutive equation of the capacitors was given in section 2.2 as:

$$i^c = \frac{d(Cu^c)}{dt} \quad (4.41)$$

Combining equations 4.40 and 4.41 we can write a system of $l + c$ differential equations, called the *dynamic system* of equations:

$$\frac{d(M^e w)}{dt} = w^d + \frac{d}{dt} \begin{pmatrix} \varphi^p \\ 0 \end{pmatrix} \quad (4.42)$$

In equation 4.42, $M^e = \begin{pmatrix} L & 0 \\ 0 & C \end{pmatrix}$. To keep the parallelism with the mechanical formalisms, M^e is denoted the *electrical mass matrix*, which is symmetric since the inductance and capacitance matrices L and C are symmetric. The array w_d contains the subset of algebraic variables appearing in 4.40 and 4.41:

$$w^d = \begin{pmatrix} u^l \\ i^c \end{pmatrix} = Dy \quad (4.43)$$

System 4.42 can be rewritten as:

$$M^e \dot{w} = Ew + Dy + \Upsilon \quad (4.44)$$

with $E = -\frac{dM^e}{dt}$ and $\Upsilon = \frac{d}{dt} \begin{pmatrix} \varphi^p \\ 0 \end{pmatrix}$. For pure electrical circuits, the electrical mass matrix M^e can be considered as constant and E vanishes. In this work, we are interested in modeling electromechanical converters and matrix M^e becomes function of mechanical variables varying with time; accordingly, E has to be taken into account in 4.44.

Note: Equations 4.39 and 4.44 are also used by Dan Telteu [71] who works on the modeling of power electronic circuits involving switches. In his research, the electrical parameters are constant and matrix E vanishes. However, according to the switches configuration, the topology of the circuit varies, while being considered as fixed in the present text.

When matrix A is full rank, it is possible to solve the algebraic system 4.39 with respect to y . The corresponding solution

$$y = A^{-1} (Bw + Cz) \quad (4.45)$$

can be replaced inside 4.44, and the final set of ODEs is obtained as

$$M^e \dot{w} = (E + DA^{-1}B)w + DA^{-1}Cz + \Upsilon \quad (4.46)$$

At this stage, we can deduce some properties of the ‘‘electrical mass matrix’’. Looking at the energy stored in capacitors W_e^* ($= W_e$ since we consider linear elements) and inductors W_m^* ($= W_m$), we can write:

$$W_m^* + W_e^* = \int_0^{i^l} \varphi^{l^T} di^l + \int_0^{u^c} q^{c^T} du^c \quad (4.47)$$

$$= \frac{1}{2} i^{l^T} L i^l + \varphi^{p^T} i^l + \frac{1}{2} u^{c^T} C u^c \quad (4.48)$$

which can be rewritten

$$W_m^* + W_e^* = \frac{1}{2} w^T M^e w + \varphi^{p^T} i^l \quad (4.49)$$

Suppose that the magnets are not present in the circuit, expression 4.49 reduces to

$$W_m^* + W_e^* = \frac{1}{2} w^T M^e w \quad (4.50)$$

and since $W_m^* + W_e^* \geq 0$, we must have

$$\frac{1}{2} w^T M^e w \geq 0 \quad (4.51)$$

where the equality is only satisfied for $w = 0$. The latter guarantees us that M^e is *symmetric positive definite*, like the mass matrix M of mechanical MBS. Introducing magnets does not influence the M^e matrix, which always satisfies 4.51.

Example The described procedure will be illustrated with the RLC filter of figure 4.5. In this example, we will assume constant electrical parameters and E vanishes.

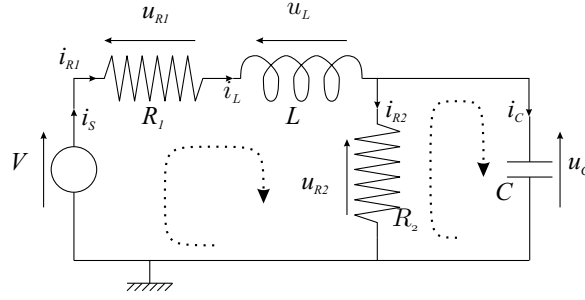


Figure 4.5: Example circuit: RLC filter

In this case, the arrays of variables are

$$w^T = (i_L \quad u_C) \quad (4.52)$$

$$y^T = (i_s \quad i_{R1} \quad i_{R2} \quad u_s \quad u_{R1} \quad u_{R2} \quad u_L \quad i_C) \quad (4.53)$$

$$z = V \quad (4.54)$$

On the basis of Kirchoff's equations

$$\begin{aligned} i_S &= i_{R1} \\ i_{R1} &= i_L \\ i_L &= i_{R2} + i_C \\ V &= u_{R1} + u_L + u_{R2} \\ u_{R2} &= u_C \end{aligned} \quad (4.55)$$

and the resistors and sources constitutive equations, the matrices A , B and C appearing in the algebraic system of equations 4.39 are constructed,

$$A = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & R_1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & R_2 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \quad C = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Matrix M^e is given by $M^e = \begin{pmatrix} L & 0 \\ 0 & C \end{pmatrix}$, while matrix

$$D = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Solving the algebraic system for y and replacing the solution in the dynamic system of equations, the final dynamic equations for this simple example are:

$$\begin{pmatrix} L & 0 \\ 0 & C \end{pmatrix} \begin{pmatrix} \frac{di^L}{dt} \\ \frac{du^C}{dt} \end{pmatrix} + \begin{pmatrix} R_1 & 1 \\ -1 & \frac{1}{R_2} \end{pmatrix} \begin{pmatrix} i^L \\ u^C \end{pmatrix} + \begin{pmatrix} -V \\ 0 \end{pmatrix} = 0 \quad (4.56)$$

which are equivalent to equations 3.146, previously obtained with the Virtual Work Principle.

4.2.2 Constrained circuits

Constraints expressions In the previous developments, we have assumed that the algebraic system 4.39 could be solved. When the structure of the circuit is such that Kirchoff's equations lead to constraints on the state variables under the form of algebraic relationship between the state variables and the sources values only, lines of zeros appear in matrix A . The latter is not full rank and equations 4.39 cannot be solved. This happens in two cases:

1. When a loop involves only capacitors and voltage sources, the corresponding Kirchoff voltage law induces a relation of the form:

$$K_{1c}^T u^c + K_{1su}^T u(t) = 0 \quad (4.57)$$

2. When a cutset involves only inductors and current sources, the corresponding Kirchoff current law induces a relation of the form:

$$K_{2l}^T i^l + K_{2si}^T i(t) = 0 \quad (4.58)$$

These constraints are slightly different from 3.147 and 3.148 considered in the virtual work formalism. This can be explained by the choice of dynamic variables that is different in both approaches: here, we have chosen i^l and u^c as state variables and thus, Kirchoff's current laws relating i^l and i^c are not considered as constraints, as it was with the Virtual Work Principle, that uses q^l and q^c as generalized variables.

To illustrate this difference, let us consider the simple RLC circuit presented in figure 4.6.

In this circuit, using the Virtual Work Principle, the circuit is considered as a constrained circuit. Indeed, we have two generalized variables q_L and q_C and one constraint written as $\dot{q}_L = \dot{q}_C$. We would obtain one second order equation:

$$L\ddot{q}_L + R\dot{q}_L + \frac{q_L}{C} = V \quad (4.59)$$

Using the circuit based formalism, this circuit is not constrained (two generalized variables i_L and u_C but no constraints) and two first order dynamic equations are obtained:

$$\begin{aligned} L \frac{di_L}{dt} &= V - u_C - Ri_L \\ C \frac{du_C}{dt} &= i_L \end{aligned} \quad (4.60)$$

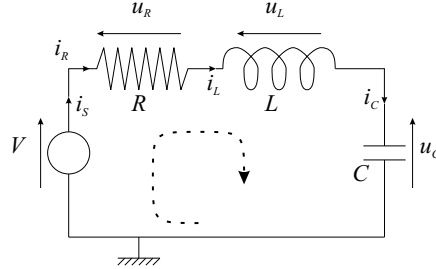


Figure 4.6: Simple RLC circuit

Both systems are completely equivalent, but in order to see it, we have to add the relation $i_L = \frac{dq_L}{dt}$ and replace it inside the second order virtual work equation 4.59 to obtain the set of first order equations:

$$\begin{aligned} \frac{dq_L}{dt} &= i_L \\ L \frac{di_L}{dt} &= V - Ri_L - \frac{q_L}{C} \end{aligned} \quad (4.61)$$

which is equivalent to the system obtained with the circuit based formalism 4.60. Indeed, if we replace

$$i_L = i_C = C \frac{du_C}{dt} \quad (4.62)$$

$$\frac{q_L}{C} = u_C \quad (4.63)$$

in 4.61, we get 4.60.

Degrees of freedom for constrained electrical circuits The discussion on the number of constraints reveals an ambiguity when trying to define the *number of degree of freedom* (d.o.f.) associated with electrical circuits, by analogy with the mechanical concept.

First, we could use the definition derived from the mechanical concept: the number of degree of freedom is the minimal number of independent variables that have to be specified to know the configuration of the multibody systems. This corresponds to the difference between the number of generalized coordinates and the number of independent constraints. Applying this definition to electrical circuits, we would have one d.o.f. for the RLC example of figure 4.6 with the Virtual Work Principle, and two d.o.f. with the circuit based approach. The ambiguity comes from the order of the equations that differs in both approaches.

Second we could define the number of d.o.f. as the minimal number of equations of the same order that are necessary to describe the behavior of the system. In mechanics, since the equations of motion are always of second order, the number of d.o.f. remains the same, but for electrical circuits, we would get the number of d.o.f.

associated with the circuit based approach since the latter guarantees us to count first order equations only, while the Virtual Work Principle may lead to first or second order equations. For the RLC circuit of figure 4.6, this would give two d.o.f. This last definition is the one we consider in this text, and the number of degrees of freedom for electrical circuits is equal to:

$$\#dof = l + c - m \quad (4.64)$$

where m is the number of constraints of types 4.57 and 4.58.

Let us mention that the chosen definition actually corresponds to the *order of the electrical circuit*, what is the number of independent first order differential equations describing the dynamics of the system. The order of the system is unique, whatever the order of the equations describing its behavior. In mechanics, the order is always equal to twice the #d.o.f.

Automatic derivation of the constraints In the present formalism, the constraint equations take the forms 4.57 or 4.58, which correspond to lines of zero in matrix A from the algebraic equations 4.39.

When writing the N^{el} Kirchoff's equations, only independent loops are considered⁸ and the constraint equations may not appear explicitly but exist and the lines of A are not independent. By applying a triangularization procedure to A , it is possible to detect the presence of dependencies between its lines, and to get the expression of the constraints.

Let us first rewrite the algebraic system of equations 4.39 as:

$$Tw' = 0 \quad (4.65)$$

with $T = \begin{pmatrix} A & -B & -C \end{pmatrix}$ and $w'^T = (y^T w^T z^T)$.

Gauss triangularisation, with pivoting on the columns of A and on all the lines, may be applied and leads to an equivalent system of equation:

$$T'w' = 0 \quad (4.66)$$

If matrix A is not full rank, T' has the form shown in figure 4.7.

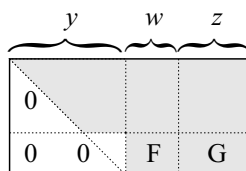


Figure 4.7: Structure of the triangularized matrix T'

⁸see section 2.2.4

The last equations of system 4.66 involve only state variables and sources values. They correspond to the m constraints 4.57 and 4.58 and can be written as:

$$Fw + Gz = 0 \quad (4.67)$$

Because of these constraints, the state variables w are not independent anymore. The set of equations to be solved is a mix of algebraic equations 4.39, differential equations 4.44 and algebraic constraints⁹ 4.67:

$$\begin{aligned} Ay &= Bw + Cz \\ M^e \dot{w} &= Ew + Dy + \Upsilon \\ Fw + Gz &= 0 \end{aligned} \quad (4.68)$$

It is of course possible to use DAEs solvers to solve this full set of equations but, we decided to avoid as much as possible DAEs, especially when an analytical elimination of the constraints is possible. In the following section, we will apply the coordinate partitioning technique to the electrical equations 4.68 and variables w , in order to get pure ODEs.

4.2.3 Electrical coordinate partitioning

Using a coordinate partitioning technique, as for MBS (see section 4.1.3), will make it possible to reduce the set of DAEs 4.68 to purely ordinary differential equations. Constraints 4.67 being linear, a full analytical elimination is possible.

The coordinate partitioning procedure requires to separate the state variables into m dependent and $l + c - m$ independent state variables, respectively denoted w_v and w_u :

$$w = \begin{pmatrix} w_u \\ w_v \end{pmatrix} \quad (4.69)$$

According to this subdivision of the state variables, the constraints matrix F may also be subdivided, and the constraints 4.67 become:

$$\begin{pmatrix} F_u & F_v \end{pmatrix} \begin{pmatrix} w_u \\ w_v \end{pmatrix} + Gz = 0 \quad (4.70)$$

which can be solved with respect to the dependent state variables w_v :

$$w_v = -F_v^{-1} F_u w_u - F_v^{-1} Gz \quad (4.71)$$

⁹which are redundant with some of the equations from 4.39.

Note: Here, we have explicitly assumed that F_v can be inverted. This assumption is valid for the cases we consider but not for every possible circuit. For example, when a loop of voltage sources exist in a circuit, a constraint exist on the voltage sources only and F contains a line of zero. These situations do not occur with equivalent circuits of electromechanical converters, but, when considering power electronic circuits with numerous switches, it may arise for certain configurations of the switches. This is out of the scope of this research but is considered by Dan Telteu who works on variable topology circuits. We shared a journal paper on the subject [71]. In his work, the electrical mass matrix M^e is considered as constant and permanent magnets are not considered.

Matrices M^e , E , D and Υ are also partitioned and the dynamic system of equations 4.44 can be transformed into

$$\begin{pmatrix} M_{uu}^e & M_{uv}^e \\ M_{vu}^e & M_{vv}^e \end{pmatrix} \begin{pmatrix} \dot{w}_u \\ \dot{w}_v \end{pmatrix} = \begin{pmatrix} E_{uu} & E_{uv} \\ E_{vu} & E_{vv} \end{pmatrix} \begin{pmatrix} w_u \\ w_v \end{pmatrix} + \begin{pmatrix} D_u \\ D_v \end{pmatrix} y + \begin{pmatrix} \Upsilon_u \\ \Upsilon_v \end{pmatrix} \quad (4.72)$$

The second set of equations from 4.72 can be solved with respect to \dot{w}_v , which is replaced into the first set of equations. Finally, using 4.71, the dependent variables are eliminated from the obtained equations, and we get the reduced set of ODEs:

$$M_r^e \dot{w}_u = E_r w_u + D_r y + \Upsilon_r + G_r z \quad (4.73)$$

where,

$$\begin{aligned} M_r^e &= M_{uu}^e - M_{uv}^e M_{vv}^{e-1} M_{vu}^e; \\ E_r &= -M_{uv}^e M_{vv}^{e-1} E_{vu} + E_{uu} - (M_{uv}^e M_{vv}^{e-1} E_{vv} - E_{uv}) (-F_v^{-1} F_u); \\ D_r &= -(M_{uv}^e M_{vv}^{e-1} D_v - D_u); \\ \Upsilon_r &= -(M_{uv}^e M_{vv}^{e-1} \Upsilon_v - \Upsilon_u); \\ G_r &= (M_{uv}^e M_{vv}^{e-1} E_{vv} - E_{uv}) (-F_v^{-1} G) \end{aligned} \quad (4.74)$$

Note: We proved on page 101 that the electrical mass matrix M^e is symmetric and positive definite. This guarantees that M_{vv}^e can be inverted.

Similarly, the algebraic system of equations 4.39 is transformed into

$$Ay = \begin{pmatrix} B_u & B_v \end{pmatrix} \begin{pmatrix} w_u \\ w_v \end{pmatrix} + Cz \quad (4.75)$$

The dependent state variables w_v may now be considered as algebraic variables and the algebraic system 4.75 may be rewritten as

$$\begin{pmatrix} A & -B_v \end{pmatrix} y_r = B_u w_u + Cz \quad (4.76)$$

in which the new set of algebraic variables $y_r \triangleq \begin{pmatrix} y \\ w_v \end{pmatrix}$ was introduced.

In 4.76, m additional algebraic variables w_v have been introduced and $(A - B_v)$ is not a square matrix. In order to determine the set of variables y_r , m new algebraic relations have to be added. They can be obtained by differentiating the constraints 4.67:

$$F\dot{w} + G\dot{z} = 0 \quad (4.77)$$

Using 4.44, the latter becomes

$$FM^{e-1}Ew + FM^{e-1}Dy + FM^{e-1}\Upsilon = -G\dot{z} \quad (4.78)$$

These m algebraic relations will supplement 4.76, and the resulting algebraic system of equations can be written as

$$A_r y_r = B_r w_u + \begin{pmatrix} C \\ 0 \end{pmatrix} z + \begin{pmatrix} 0 \\ -G \end{pmatrix} \dot{z} + \begin{pmatrix} 0 \\ -FM^{e-1} \end{pmatrix} \Upsilon \quad (4.79)$$

where

$$A_r = \begin{pmatrix} A & -B_v \\ FM^{e-1}D & FM^{e-1}E_v \end{pmatrix} \text{ and } B_r = \begin{pmatrix} B_u \\ -FM^{e-1}E_u \end{pmatrix}$$

have been introduced, with

$$E = \begin{pmatrix} E_u & E_v \end{pmatrix}$$

Note: Here, we assume that matrix A_r can be inverted and thus the augmented algebraic system of equations 4.79 can be solved. For circuit with variable topology, A_r might not be full-ranked, but this is out of the scope of the present work. As already mentioned, Dan Telteu works on this kind of circuit and deals with such particular situations.

Using this coordinate partitioning procedure, the algebraic system of equations is supplemented while the differential system of equations is reduced and expressed in terms of the independent variables w_u . Solving 4.79 and replacing the solution y_r into equations 4.73, we obtain pure ODEs, which can be solved using classical ODE solvers.

The flowchart of the numerical integration scheme is shown in figure 4.8, where z are the source values and \dot{z} their first derivative. They are input to the computation and may vary with respect to time t .

Let us point out that the constraints detection method (see section 4.2.2) and the coordinate partitioning procedure will always lead to the minimal number of first-order differential equations.

In multibody dynamics, the Jacobian matrix of the constraints $J^m(s^m)$ is nonlinear and varies with the configuration of the systems, which can be such that J_v^m is singular. This makes the choice of dependent variables critical. In this research, the topology of the electrical circuit is fixed and matrix F_v only contains zeros and/or

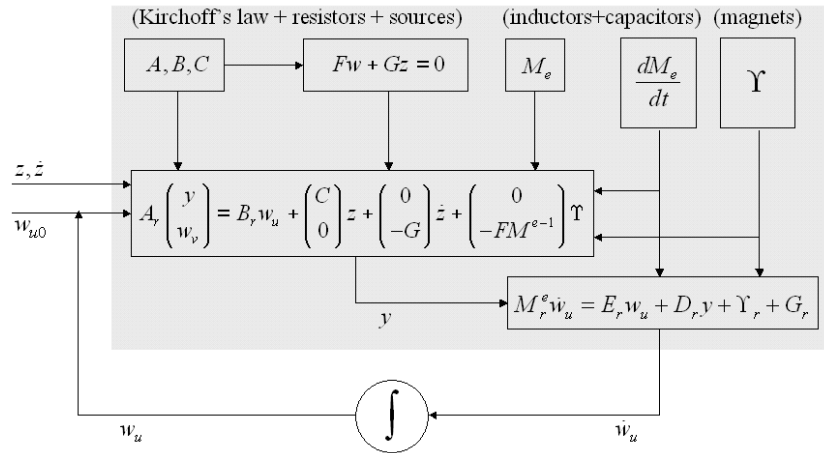


Figure 4.8: Electrical Coordinate Partitioning and Integration Scheme

resistance values. It is well conditioned from a numerical point of view and all the choices of dependent variables are acceptable¹⁰.

This circuit based formalism and the coordinate partitioning were implemented in the Electran software, the symbolic companion program of Robotran (see chapter 5).

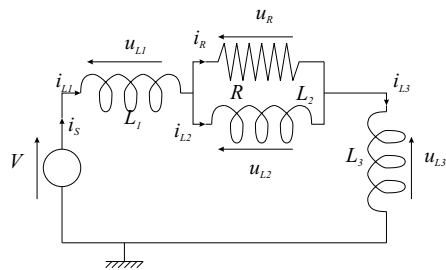


Figure 4.9: Example of circuit with constraint

Example Considering the example circuit of figure 4.9, that was already considered when presenting the Virtual Work Principle, the different set of variables are:

$$w^T = (i_{L1} \quad i_{L2} \quad i_{L3}) \tag{4.80}$$

$$y^T = (i_s \quad i_R \quad u_s \quad u_R \quad u_{L1} \quad u_{L2} \quad u_{L3}) \tag{4.81}$$

$$z = V \tag{4.82}$$

¹⁰This is always the case for circuits involving resistors with constant resistance. For circuits with variable parameters, zero resistances are problematic but the topology of the corresponding circuit is different from the one initially considered. Circuits with variable topology were not considered in this work.

There exist a constraint between the chosen state variables w . This constraint can be written as

$$i_{L3} - i_{L1} = 0 \quad (4.83)$$

We will show here that the above procedure automatically gives the expression of the constraints.

First, we have to construct the algebraic and dynamic systems of equations as follows

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & R & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad C = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (4.84)$$

Matrix M^e is given by $M^e = \begin{pmatrix} L_1 & 0 & 0 \\ 0 & L_2 & 0 \\ 0 & 0 & L_3 \end{pmatrix}$, while

$$D = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Matrix T' can be constructed and triangularized by executing the following transformations sequently:

- Lines 7 and 3 are interchanged
- Line 4 is replaced by the sum of lines 4 and 3
- Line 6 is replaced by the sum of lines 6 and 5
- Lines 4 and 5 are interchanged
- Eventually, line 6 is replaced by line 6 minus line 2 multiplied by R

This procedure can be fully automatized using linear algebra techniques such as LU factorization or Gauss triangularization¹¹.

The obtained T' matrix is

$$T' = \left(\begin{array}{cccccccc|ccc|c} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & +1 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -R & R & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \end{array} \right) \quad (4.85)$$

¹¹which is used in our implementation described in chapter 5.

As expected, the last line of T' has the structure of figure 4.7 and the constraint matrices F and G are

$$F = \begin{pmatrix} -1 & 0 & 1 \end{pmatrix} \text{ and } G = 0 \quad (4.86)$$

We can now separate the state variables into dependent $w_v = i_{L1}$ and independent ones $w_u = (i_{L2} \ i_{L3})^T$. Applying the procedure described from equations 4.69 to 4.79, we obtain the new algebraic system of equations in terms of

$$w_u^T = \begin{pmatrix} i_{L2} & i_{L3} \end{pmatrix} \quad (4.87)$$

$$y_r^T = \begin{pmatrix} i_s & i_R & u_s & u_R & u_{L1} & u_{L2} & u_{L3} & i_{L1} \end{pmatrix} \quad (4.88)$$

$$z = V \quad (4.89)$$

as

$$A_r = \left(\begin{array}{ccccccc|c} 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & R & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & -\frac{1}{L1} & 0 & +\frac{1}{L3} & 0 \end{array} \right) \quad B_r = \begin{pmatrix} 0 & 0 \\ -1 & 0 \\ 0 & -1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \quad (4.90)$$

The reduced electrical mass matrix M_r^e is given by

$$M_r^e = \begin{pmatrix} L_2 & 0 \\ 0 & L_3 \end{pmatrix} \quad (4.91)$$

Solving the transformed algebraic system of equations and replacing in the dynamic system of equations, one gets the reduced system of dynamic equations as

$$\begin{pmatrix} L_2 & 0 \\ 0 & L_3 \end{pmatrix} \dot{w}_u = \begin{pmatrix} -R & R \\ R\frac{L_3}{L_1+L_3} & -R\frac{L_3}{L_1+L_3} \end{pmatrix} w_u + \begin{pmatrix} 0 \\ V\frac{L_3}{L_1+L_3} \end{pmatrix} \quad (4.92)$$

which is equivalent to the equations 3.159 obtained using the Virtual Work Principle.

4.2.4 Discussion

When presenting the Virtual Work Principle, we did point out that this approach intrinsically takes into account the implicit constraints for both mechanical and electrical systems. Moreover, considering virtual variations δs^m or δs^e compatible with the constraints automatically lead to reduced equations.

In mechanics, classical approaches based on Newton-Euler equations with absolute coordinates require to write a large set of equations taking into account all the

forces and torques. Using recursive computations and joint coordinates makes the Newton/Euler Recursive formalism one of the most efficient technique for large MBS, especially when implemented symbolically, while leading to the same equations as the Virtual Work Principle [16, 56]. This allowed us to use the Lagrange multiplier technique and the coordinate partitioning method [75].

For electrical circuits, the formalism presented here requires to write and solve all the algebraic and differential equations. Nevertheless, since A is a sparse matrix, when implemented symbolically, this formalism becomes efficient. The reduced equations 4.73 are also equivalent to those obtained with the Virtual Work Principle, but expressed in terms of different variables¹². However, the coordinate partitioning proposed for electrical circuits is not based on the definition of the Lagrange multipliers, which was the case in multibody dynamics (see section 4.1.3). Let us point out that the electrical coordinate partitioning proposed here is even more interesting when considering electrical circuits with variable topology, as shown in the work by Dan Telteu [71].

Fully symbolic implementation is possible for constraints detection and reduction since linear expressions are involved. Nevertheless, we suspect that when considering non-linear constitutive equations techniques similar to the one proposed in [49], where the resolution of non-linear equations is “embedded” into the symbolic model, should be applicable.

4.3 Electromechanical Coupling

We will now introduce the electromechanical interaction and indicate how a global electromechanical model can be obtained. In section 3.3.3, we have shown that when an electrical circuit interacts with a MBS, its dynamic and algebraic equations are not changed, except for some parameters that become functions of the mechanical configuration. From a mechanical point of view, the interaction results in supplementary forces acting on the MBS.

The latter explains why we could use completely independent formalism to generate the models. Nevertheless, the resolution of the obtained equations has to be done at the same instant of time t and also in a specific order, as discussed in this section.

4.3.1 Unconstrained electromechanical systems

When considering unconstrained mechanical and electrical systems, equations 4.1, 4.39 and 4.44 apply:

- Mechanical equations

$$M(s^m)\ddot{s}^m + c(s^m, \dot{s}^m, F_{ext}, L_{ext}, g) = Q \quad (4.93)$$

¹²This last statement can be proven analytically when considering linear constitutive equations for the electrical components. It requires several computations on the basis of the algebraic equations, as shown at the end of Appendix A. When nonlinear constitutive equations are considered, the same reasoning applies, but requires resolution of nonlinear equations, what is not always possible analytically.

- Electrical equations

$$\begin{aligned} Ay &= Bw + Cs \\ M^e \dot{w} &= Ew + Dy + \Upsilon \end{aligned} \quad (4.94)$$

Let us, assume that only mechanical positions s^m influence the electrical parameters, except for sources that may also be influenced by velocities \dot{s}^m :

$$\begin{aligned} A(s^m) y &= Bw + Cz(s^m, \dot{s}^m) \\ M^e(s^m) \dot{w} &= E(s^m, \dot{s}^m) w + Dy + \Upsilon(s^m, \dot{s}^m) \end{aligned} \quad (4.95)$$

and matrix E can be written as:

$$E = -\frac{dM^e}{dt} = -\frac{\partial M^e}{\partial s^{mT}} \dot{s}^m \quad (4.96)$$

In the above expressions, B and D only concern Kirchoff's equations and are not influenced by mechanical variables. It was also assumed that M^e does not exhibit explicit dependence with time t .

In the mechanical equations, supplementary forces and/or torques of electromechanical origin will appear in the equations:

$$Q^{em} = \frac{\partial W_m^*}{\partial s^m} - \frac{\partial W_e}{\partial s^m} \quad (4.97)$$

It can be introduced as a supplementary generalized joint force or external force and the equations of motion become:

$$M(s^m) \ddot{s}^m = Q(w, s^m) - c(s^m, \dot{s}^m, w, F_{ext}, L_{ext}, g) \quad (4.98)$$

Figure 4.10 indicates schematically how the two submodels interact and the sequence to solve the equations at time t : first we solve the electrical equations to compute the electromechanical force Q_{em} , which requires the mechanical configuration s^m and \dot{s}^m and, second, we solve the mechanical equations with respect to \ddot{s}^m .

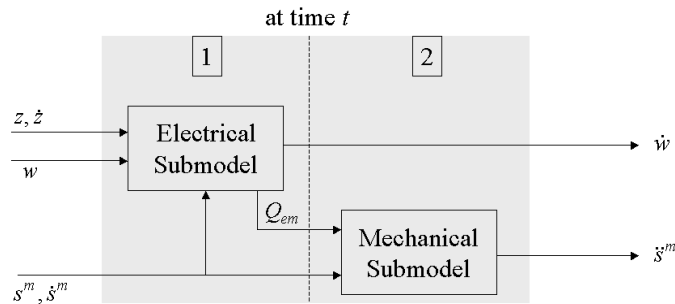


Figure 4.10: Computation sequence for unconstrained electromechanical systems

Solving these equations using classical numerical integrators requires to transform the mechanical equations into first-order equations. This can be done by defining the array of mechanical state variables as:

$$w^m \triangleq \begin{pmatrix} s^m \\ \dot{s}^m \end{pmatrix} \quad (4.99)$$

The detailed flowchart of the numerical integration scheme is shown in figure 4.11.

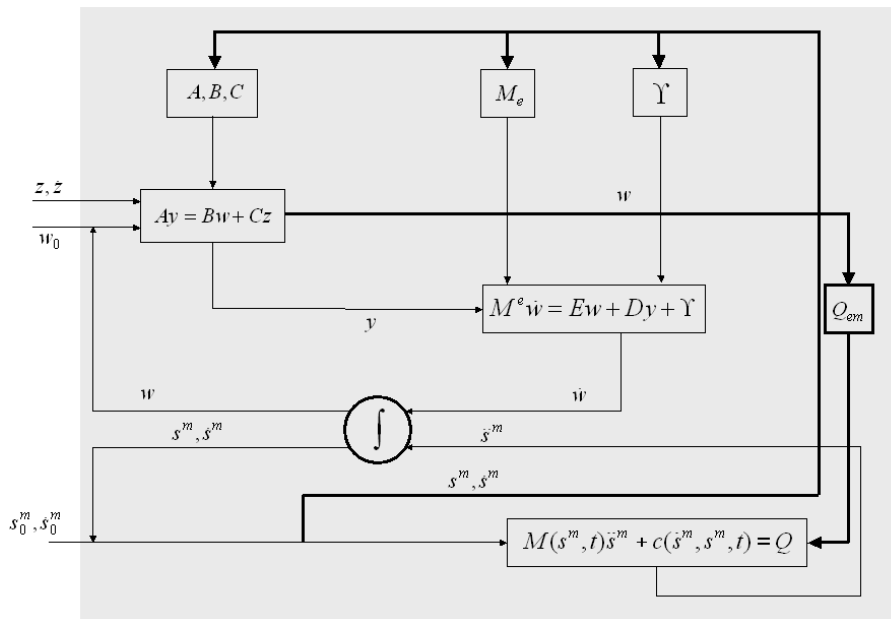


Figure 4.11: Electromechanical flowchart for unconstrained systems

4.3.2 Constrained electromechanical systems

Here, we consider electromechanical systems characterized by constrained subsystems. The mechanical or the electrical or both subsystems may be constrained. Coordinate partitioning can be applied, according to sections 4.1.3 and 4.2.3, and reduced systems of equations are obtained.

As shown in figure 4.12, the electrical equations can generally not be solved until the mechanical constraints have been considered at position and velocity level. Indeed, the electrical parameters are not necessarily influenced only by the independent joint variables s_u^m and \dot{s}_u^m .

Constrained electromechanical systems involve much more intricate calculations that have to be achieved carefully. The electrical equations have to be solved after the

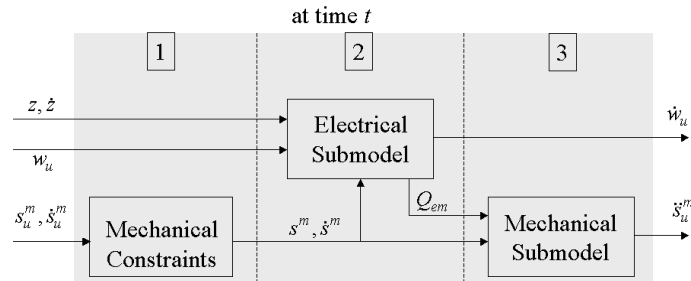


Figure 4.12: Computation sequence for constrained electromechanical systems

resolution of mechanical constraints but before the computation of the independent mechanical accelerations \ddot{s}_u^m .

Figure 4.13 shows the detailed flowchart for an electromechanical constrained system.

At the beginning of the dynamic calculations at time t , we now only have independent mechanical positions s_u^m and velocities \dot{s}_u^m and independent electrical variables w_u . Like for unconstrained systems, the electromechanical forces have to be calculated before the mechanical accelerations \ddot{s}_u^m . This requires to first solve the reduced electrical algebraic system of equations, at time t , which is influenced by the mechanical variables at position and velocity levels, s^m and \dot{s}^m . The latter can only be obtained after resolution of the mechanical constraints, at time t .

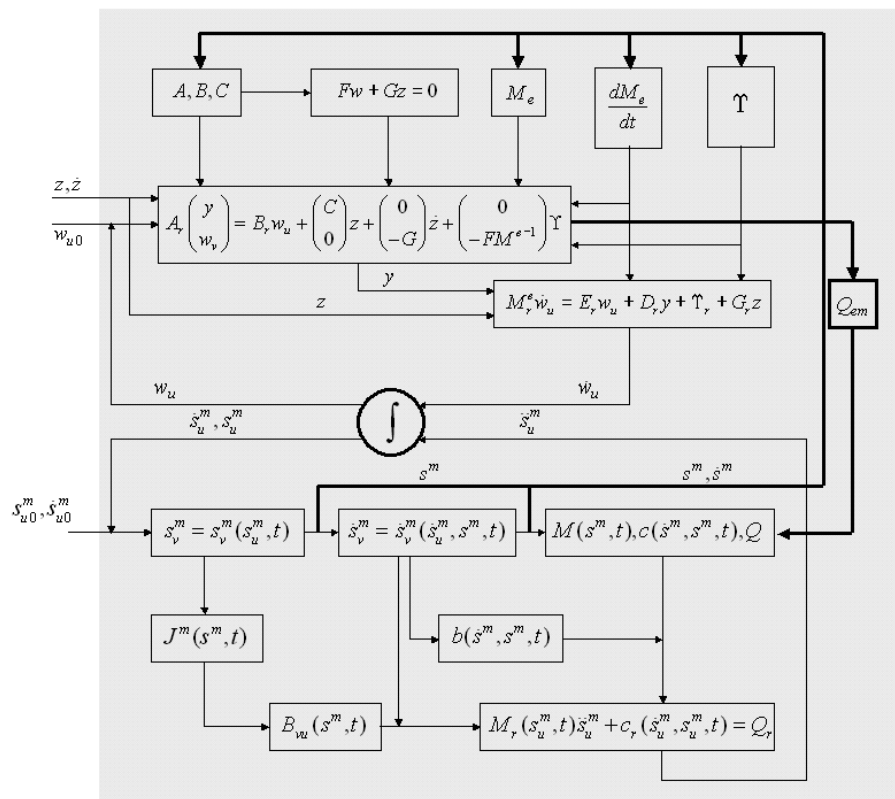


Figure 4.13: Electromechanical Integration Scheme

Symbolic Implementation

Abstract

Symbolic implementation is proposed and described in this chapter, where we start by briefly introducing the concept of symbolic implementation. The main advantages for such approach are then discussed.

Our symbolic kernel, presented in this chapter, was used to implement the formalisms described in chapter 4, what results into two programs:

- Robotran, the model generator for multibody systems developed at the PRM division of the Université Catholique de Louvain and whose latest developments permit a fully-symbolic generation of the direct dynamic reduced equations [49].
- Electran, the model generator for electrical circuits and electromechanical converters developed during this research.

A new representation, different from classical graph representations, is considered for electrical circuits. It is based on topological concepts inspired from multibody system representation and allows to deduce Kirchoff's laws necessary for the generation of the equations.

As pointed out before, the electrical and mechanical equations have to be solved in a certain sequence described in section 4.3, but the generation of the electrical and mechanical submodels can be achieved independently.

5.1 Concept of symbolic generation

Computer implementation is an unavoidable step when working in the field of modeling and simulation. Nowadays, “model generator” programs are the only way to obtain models for realistic industrial applications. The dynamics of the latter is determined by numerous equations (DAEs) and writing them by hand would be tedious and without interest from a scientific point of view.

The most usual way to simulate a system consists in using a numerical package (e.g.: Simpack, Adams, PSpice, Mecano, etc.). With such tool, a sequence of instructions (usually calls to other subprograms), using the instantaneous state of the system, is written once and allows to compute (and solve) the system of dynamic equations. This process of constructing the model via a set of subprograms offers the possibility to deal with many different applications, but in the same time, the CPU performance of such general purpose tools is affected and is clearly lower [16] than what could be done with dedicated programs.

The proposed implementation is based on another concept where the generation of the equations is separated from the rest of the process. The whole set of equations is written with alphanumerical symbols inside a subroutine (symbolic generation) that can be called by a numerical program (numerical analysis). The routine containing all these symbols and equations is the *symbolic model*.

Many advantages can be found when resorting to symbolic generation, since it is possible to:

- avoid non necessary arithmetic operations such as $0 \cdot a$ or $0 + a$, ..., a being a simple or complex expression
- simplify some mathematical expressions (factorization, trigonometric simplifications, divisions,...). For instance, $\frac{a \times b}{b}$ will be replaced by a , $a + b - a$ by b , etc., a and b being simple or complex expressions.
- avoid multiple computations of the same terms by defining auxiliary variables,
- avoid the computation of some unnecessary equations (typically for multibody recursive models, such as the Newton/Euler recursive formalism of section 4.1),
- generate models in different languages (Fortran, C, Matlab, Java,...) and also for different environments for control, optimization,...

Note: Let us also point out that a symbolic generation gives a good readability to the equations. This can be exploited for better comprehension of the system, and in particular for teaching.

From experience, it is well-known that the use of commercial symbolic packages (Maple, Mathematica,...) often leads to memory problem when the number of equations to manipulate increases. On the other hand, when the “symbolic kernel” of the program is specifically designed for a dedicated domain (e.g.: multibody dynamics

for the software Robotran), it is possible to deal with these memory problems, even when complex applications are considered (e.g.: Robotran is able to deal with more than 100 joint variables).

Different aspects have to be considered when going from theoretical formalisms to their practical computer implementation:

- First of all, one has to define what are the main objectives of the program, which can be either dedicated to a specific family of systems or designed to model any kind of systems. These are two extreme situations representative of two tendencies in modeling softwares. In the first family, the programs deal with specific systems and can be optimized for them leading to very efficient models, while in the other family, they are able to consider many kinds of systems but cannot be optimal for all of them. Softwares such as 20-sim, Matlab/Simulink, Modelica,... are part of the first family. Tools like PSpice, Adams or Robotran are specialized programs, optimized for their own field.

In this work, we aim at considering complex applications, for which efficiency becomes crucial. For this purpose, we use dedicated optimized formalisms for each part of the system, and we gather the symbolic submodels afterwards in a rigorous way.

- Secondly, one has to choose the way the models are generated. In this thesis, the implementation is *fully-symbolic*, which means that the whole model is generated using alphanumeric expressions, for which drastic simplifications may be achieved to reduce the number of floating-point operations during the subsequent computation.

The fully-symbolic character of our method indicates that, from the description of the system, the generated equations directly lead to the mechanical independent accelerations \ddot{s}_u^m [49] and the derivative of the independent electrical state variables \dot{w}_u . This means that the model can be provided to any numerical integrator for ODEs without further computations.

- Thirdly, it is important to specify the level of knowledge necessary for the user. What should he or she know to use the tool and to build models?
 - For the mechanical subsystems, concepts such as body, joint, loop, link, constraint, etc. are to be mastered in addition to basic mechanical concepts such as mass, inertia, stiffness or damping. A tree representing the bodies and their joints is the main input of a multibody program like Robotran.
 - For the electrical subsystems, the user has to be able to provide an equivalent circuit of the converter and its supply, what requires a good knowledge of its structure and working principles. This equivalent circuit is the main input to the electrical program Electran.

During this research, we opted for a strategy in which the symbolic submodels of the mechanical and electrical components of the electromechanical system are ob-

tained from different independent formalisms, presented in chapter 4 and rigorously interfaced afterwards, as discussed in section 4.3.

Two programs are thus necessary to obtain the submodels:

- Robotran which is a symbolic model generator for multibody systems. It is based on its own symbolic engine dedicated to multibody dynamics [20] and is able to deal with large systems involving more than 100 joint variables.
- Electran which is a symbolic model generator for electrical circuits and electromechanical converters. It was developed in the framework of this thesis and is based on the Robotran philosophy: symbolic manipulations, use of auxiliary variables, reduction of the equations, etc. Robotran's symbolic engine, which was complemented for this purpose, is used for the generation of electrical models [58], on basis of the formalism described in chapter 4.

Concerning the constraints, we can either treat them together with the dynamic equations by using DAE solvers, or solve them symbolically in order to reduce the equations to pure ODEs, what is even better. Although non-linear mechanical constraints require specific numerical algorithms to be solved, a fully-symbolic implementation is possible, as shown in [49], who proposes to use the Newton-Raphson procedure to solve the constraints at algebraic level and integrates this resolution inside the symbolic model. This technique was implemented in Robotran. For electrical systems, when considering linear constitutive equations for resistors, constraints are linear and can be automatically detected and reduced symbolically by Electran¹.

5.2 Robotran's Symbolic Engine

Robotran was developed to generate direct/inverse kinematic/dynamic symbolic models of multibody systems. Its kernel is a symbolic engine dedicated to the expressions and symbols involved in multibody dynamics [16, 41].

Starting from symbols and basic operators (+, -, *, =, sin, cos), Robotran uses a tree structure to represent the expressions. The operators are the nodes of the tree, while the symbols are the leaves of the tree, as we can see in figure 5.1 showing the tree of the expression $2 * a - g * \cos(s_1^m + s_2^m)$.

Robotran's expressions are "hierarchised" according to well-established arithmetic priority rules. The most general expression required is an addition/subtraction of terms being multiplication/division of factors, being a constant, a variable, a trigonometric relation, etc. or a full expression. This makes the symbolic manipulations intrinsically recursive, from an algorithmic point of view.

The C language has been chosen to implement Robotran. Using the "struct" type, to represent an expression (constant, operator with 1 or 2 arguments, etc.), and pointers, the tree structure illustrated in figure 5.1 can be built. For example the "struct" type expression representing the right operator * (see figure 5.1) contains two arguments that are pointers towards two other "struct" type expressions, one being a

¹Introducing non-linearities in the same way as done in [49] should be possible.

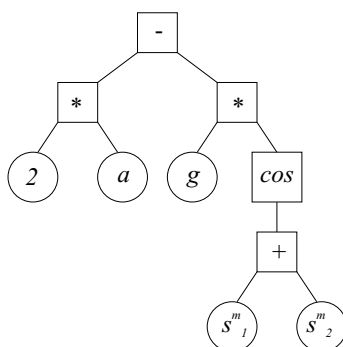


Figure 5.1: Tree representation of the expression $2 * a - g * \cos(s_1^m + s_2^m)$

constant g , the other being a cosine expression whose arguments point towards the addition of two variables s_1^m and s_2^m .

Above this basic expression level, a library of mathematical manipulations was developed in order to help building equations. It consists of elementary functions such as “add”, “sub” or “mult”. Each time such a function is called, the operands are observed in order to detect possible simplifications. For example, when subtracting expression b from expression a using $sub(a, b)$, we check whether $a = b$, in which case the symbol “0” is returned. By resorting to a lexicographical² organization of symbols (e.g. $0 < 8 < a < z < B < Z \dots$), each expression is systematically reorganized by Robotran in order to detect every single simplification. For example, expression $a + C + z + d - a$, constructed as the subtraction of $a + C - z + d$ and a , will be reorganized as $a - a + d - z + C$ so that the simplification $a - a$ explicitly appears.

Due to body rotations, trigonometric expressions are very common in multibody dynamics and many work has been done in this direction when developing Robotran. All the classical trigonometric simplifications are implemented (for example: $2\sin(a)\cos(a) = \sin(2a)$). Since many of these expressions are necessary several times during the computation process, auxiliary variables are automatically generated in order to avoid multiple evaluations of the same entity. For example, $\sin(a)$ might be used 10 times and would lead to 10 “expensive” calls to function \sin . We thus reduce the number of calls to 1 by defining the auxiliary variable $sa = \sin(a)$ and by using the symbol sa in the rest of the computations.

When printing out the final equations in a file, because the interdependency of the expressions is completely known by Robotran, it can automatically detect whether a full equation is necessary for the required result and thus whether it should be printed or not. This is of importance when dealing with recursive formalism that generates many expressions not always necessary for the desired result. Indeed, the formalism intrinsically compute unnecessary expressions since the equations of motion are obtained by projecting the vectors equations onto the joint space. Up to 30% of the ex-

²based on the ASCII table

pressions may be not necessary for the final result (e.g. the acceleration \ddot{s}_m for direct dynamics): this is not negligible. Similarly when considering constrained multibody systems, the dependent accelerations \ddot{s}_v^m are not required to run a time simulation. Of course, afterwards, we will need the full set of accelerations \ddot{s}^m for the computation of the joint reaction forces but this is already a post process of the main simulation. This illustrates that depending on the needs, some generated expressions do not need to be printed out in the model.

5.3 Modeling Multibody Systems with Robotran

This section will give a short description of the software capabilities. It will also illustrate the level of expertise required from the user.

5.3.1 Robotran's Conventions

Robotran only considers two types of single-degree-of-freedom (d.o.f.) joints: a revolute joint (type R) or a prismatic joint (type T). In this implementation, we will also assume that the joints are always aligned with one of the body-fixed frame axes: $\{\hat{\mathbf{X}}_1^i, \hat{\mathbf{X}}_2^i, \hat{\mathbf{X}}_3^i\}$, and six possible joints are considered: three prismatic joints denoted T1, T2 and T3, and three revolute joints denoted R1, R2 and R3. This is not restrictive because any physical joint can be represented by a succession of elementary joints. For example, it is well-known that any rotation can always be replaced by three successive rotations around orthogonal axes³.

Each joint connects two bodies (a parent and a child, see section 2.1). Accordingly, only one d.o.f. can be inserted between two bodies. In many practical cases, this is of course not sufficient because more d.o.f. exists between bodies (e.g. ball joint, hinge joint, etc.). We define a *fictitious body* as a dimensionless, massless body. By introducing one or more fictitious bodies between two bodies, we can insert the necessary d.o.f. without affecting the dynamics of the system⁴, as illustrated in figure 5.2 where a ball joint is modeled by means of two fictitious bodies.

The concept of *locked joint* is also introduced. A *locked joint* is defined as a joint whose generalized variable s^m is maintained constant. For example, it can be inserted between two bodies (fictitious or normal) rigidly attached but whose frames are not aligned, and consequently allows the introduction of joints that are not aligned with the body-fixed frame. More generally *driven joints* are such that their displacement is an imposed function of time: $s^m = f(t)$, $\dot{s}^m = \frac{df(t)}{dt}$ and $\ddot{s}^m = \frac{d^2f(t)}{dt^2}$.

Robotran also allows for inserting *links* between bodies. When the dynamic properties (mass and inertia) of a mechanical component are negligible but not its interaction with the bodies, a *link* is used and the component is replaced by two external

³We are aware of possible singularities that can occur but this is out of the scope of this research.

⁴and without making the equations more complex since zeros are automatically eliminated in the symbolic approach.

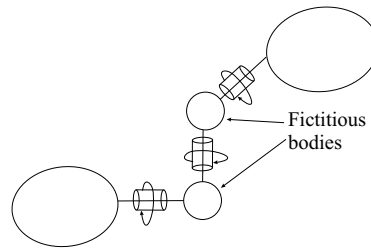


Figure 5.2: The use of fictitious bodies

forces, see figure 5.3, between two attach points in the system. These forces are function of the relative distance z and velocity \dot{z} of the points. Typical use of the links is the modeling of springs and dampers in vehicle suspensions, what will be shown in section 7.2 where the model of a railway vehicle is presented.

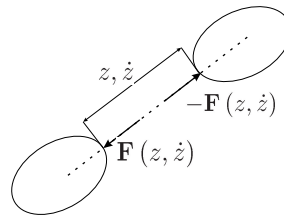


Figure 5.3: Link in Robotran

5.3.2 Robotran's input

In order to generate the equations of motion of the multibody system, Robotran requires a description of the bodies, joints and forces involved as well as the description of the system's topology. The input file summarizing all this information is a translation of the schematic tree representation of the system defined in section 2.1. An example of such a file and some explanations are given in figure 5.5, for a tree-like system. The corresponding tree representation is given in figure 5.4, in the so-called "reference" configuration.

All the information provided in this input file consists in symbols that will be used to build the equations. The "0" symbol should be used in the input file each time the value of the corresponding parameter is zero in the real system. This helps in noticeably reducing the size of the final generated equations. For example, lots of vectors have zero x , y or z components. Similarly, the fictitious bodies have zero mass, inertia,... According to the input file of figure 5.5, corresponding to the system of figure 5.4, body 4 is the only one to be affected by external forces and torques.

Closed-loop systems are described by first opening the loops using specific *cuts*.

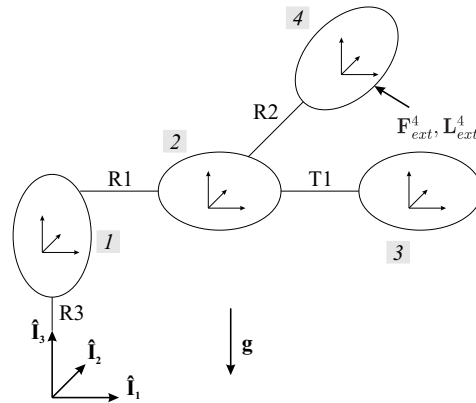


Figure 5.4: Example of multibody system representation

The corresponding tree is given as just explained, while the description of the cuts requires an additional symbolic input of Robotran (type and location of the cuts). More details on the different types of cuts, as well as other symbolic input like sensors, can be found in [56].

5.3.3 Robotran's output

Robotran generates symbolic equations for kinematics or dynamics analysis under different formats and using different formalisms. Among the possible model that Robotran can generate from the previous input file, let us mention:

- direct dynamics (open or closed-loop system) with the following formalisms
 - “NER”: the Newton-Euler Recursive formalism used in this work
 - “PPP”: the virtual power principle
 - “ODN”: Order-N formalism [65] (only for tree-like systems)
- inverse dynamics (computation of joint forces for a given system configuration and motion)
 - “NER”
 - “PPP”
- direct kinematics (including constraints, constraints Jacobian, position, velocity and acceleration of a specified points, etc.)
- inverse kinematics
- reaction dynamics (used for example for the computation of bedplate force of a robot)

Robot	System's name
4	Number of bodies (fictitious included)
0 1 2 2	Array of parent indexes (<i>inbody-array</i>)
-- Joints --	
R3 sm1 smd1 smdd1	Type of joint and symbols that will be used in the equations for
R1 sm2 smd2 smdd2	the generalized coordinates at position (<i>sm</i>), velocity (<i>smd</i>)
T1 sm3 0 0	and acceleration (<i>smdd</i>) level
R2 sm4 smd4 smdd4	
-- \mathbf{d}^i vectors --	
0 0 0	Components of the position vector \mathbf{d}^i for each joint with
D21 0 0	respect to the previous one
D31 D32 D33	
D41 0 D43	
-- \mathbf{d}^i vectors --	
0 0 0	Components of the position vector \mathbf{d}^i for each center of mass
L11 0 L23	with respect to the previous joint
L31 L32 0	
L41 L42 0	
-- Mass --	
0	Mass of each body m^i
m2	
m3	
m4	
-- Inertia --	
0 0 0	
0 0	
0	
I211 0 0	Inertia matrix of each body I^i
I222 0	
I233	
I311 0 0	
I322 0	
I333	
I411 I412 I413	
I422 I423	
I433	
-- External Forces --	
0 0 0	Components of the external resultant force vector acting on
0 0 0	each body \mathbf{F}_{ext}^i
0 0 0	
F41 F42 F43	
-- External Torques --	
0 0 0	
0 0 0	
0 0 0	
C41 C42 C43	Components of the external resultant torque vector acting on
-- Gravity --	each body \mathbf{L}_{ext}^i
0 0 g	

Figure 5.5: Example of the symbolic input file for Robotran

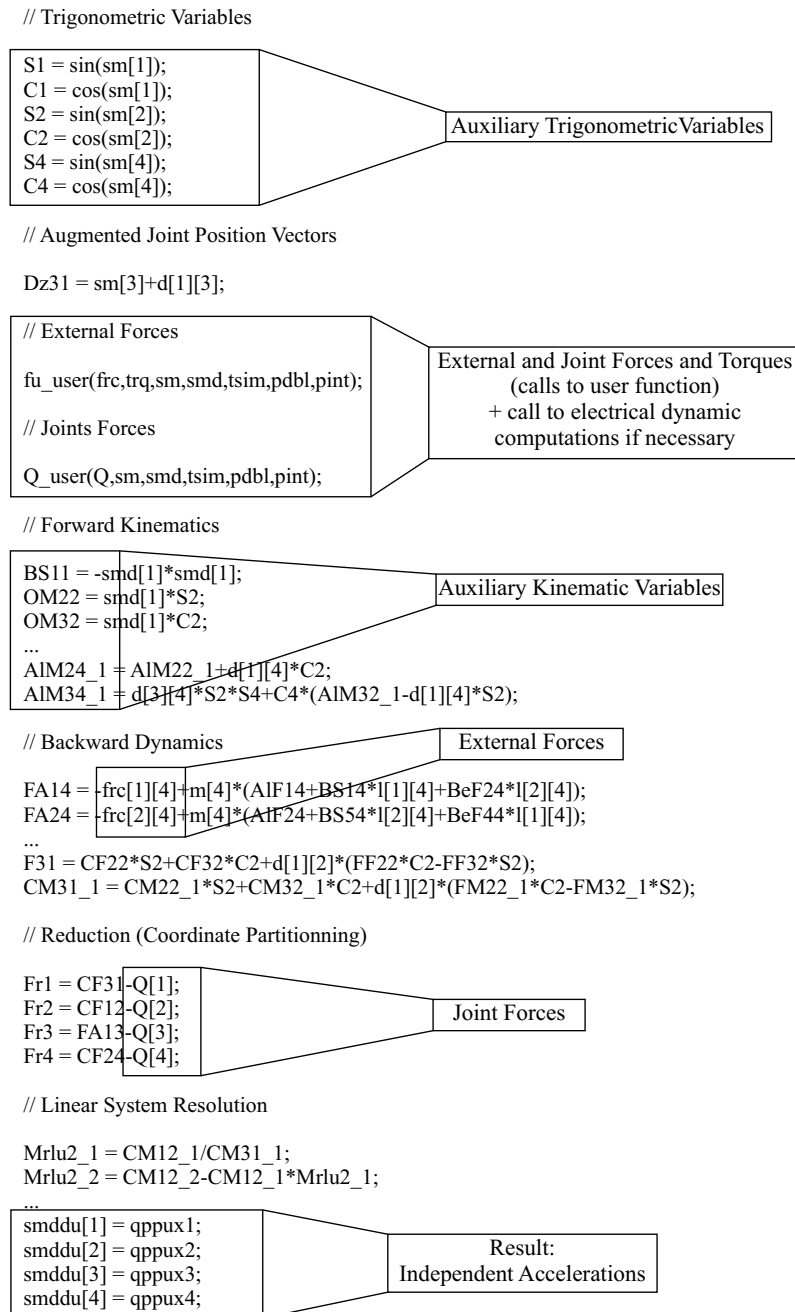


Figure 5.6: Example of Robotran output file

Once generated, the equations are printed out as functions that can be called within a numerical environment to be solved and/or time-integrated.

As an example, the output equations (direct dynamics with NER formalism) of the system defined in figure 5.5 are given in Matlab language in figure 5.6.

5.3.4 Electromechanical interaction

When interacting with electromechanical converters, the MBS is subjected to additional forces Q_{em} . The latter may be introduced either as joint or external forces acting on the MBS and a call to the electrical dynamics computation would appear in the generated code of figure 5.6 at the same location as the user function calls for external and joint forces, that is after the resolution of the constraints at position and velocity level, but before the computation of the independent acceleration \ddot{s}_u^m .

5.4 Modeling Electrical Systems with ElectrAn

5.4.1 Circuit representation

The electrical circuit is represented as a tree-like structure, as shown in figure 5.7 for a simple circuit. The branches of the tree correspond to the elements of the circuit and the nodes of the tree correspond to the physical nodes of the circuit. The *root* of the tree (node 0) is the reference node⁵. This tree representation was inspired from the MBS representation and is different from the classical graph representation, shown in figure 5.7.a.

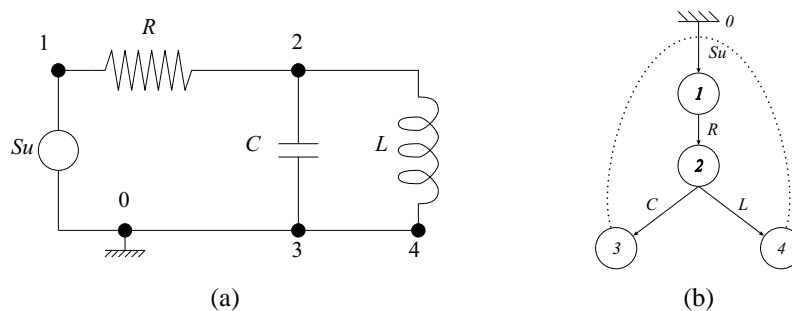


Figure 5.7: Example of a simple circuit and the corresponding tree-graph

In an electrical circuit, a *loop* is defined as a sequence of elements connecting a physical node to itself. In our tree representation, we have opened the loops by “cutting” the circuit at some physical nodes. By cutting a loop, we mean that we artificially split a physical node into two nodes in the graph. Hence, each loop is characterized by two elements having terminal nodes that correspond to the same physical node in the

⁵This reference node generally corresponds to the zero voltage node, but this is not mandatory.

circuit. For example, in figure 5.7.b, nodes 0, 3 and 4 correspond to the same physical node and the 3 loops of the circuit have been opened: loop 3-0, loop 4-0 and loop 4-3. Note that in this example specifying loops 3-0 and 4-0 automatically defines loop 4-3, what indicates that only two of three loops are independent.

Note: This “cutting” of the loops in an electrical circuit is analogous to what is done in [56] when opening loops in a MBS by splitting a body into two distinct bodies.

In figure 5.7.b, each branch is oriented from a *starting node* to an *end node*, which has the same index as the element. For example, in figure 5.7.b, element 2, the resistor R , has node 2 as terminal node and node 1 as starting node. The orientation of the branches will further indicate the sign convention by indicating the positive direction of the current flowing through the corresponding element.

We define a *path* from node i to node j as a sequence of branches ending with element j . For example, in figure 5.7.b, Su and R represent a path from node 0 to node 2.

As in MBS dynamics, the following concepts are defined for the electrical graph:

- *ascendant*: element i is ascendant of element j if the path (excluding element j) going from the root to node j contains element i .
- *direct ascendant or parent*: among the ascendants, the direct ascendant or parent of element i is the one directly connected to node k where k is the starting node for element i .
- *descendant*: element i is descendant of element j if the path (excluding element i) going from the root to node i contains element j .
- *direct descendant or child*: among the descendants, a direct descendant or child of element i is one that has node i as starting node.
- *leaf*: a leaf is an element that has no child. Its end node is called a *leaf node*.

Note: The tree structure is such that each element has only one parent, but may have several children.

For example, in figure 5.7.b:

- element 1 is ascendant of elements 2, 3 and 4
- element 2 is parent of element 3 and 4
- element 2 is child of element 1
- elements 3 and 4 are leaf elements

5.4.2 Algorithm for Generation of Kirchoff's Equations

To automatically write Kirchoff's equations, we take profit of the tree graph structure which is used to represent the electrical circuit.

Let us define different arrays of indexes that will be necessary to describe the algorithm:

- the *parent* array will be used to refer to the parent of each element. Because of the tree structure, each element has only one parent: *parent*[*j*] refers to the parent of element *j*, denoted *element*[*j*].
- the *c_element* array refers to the primary element involved in a loop: *c_element*[*k*] refers to the primary element (leaf of the tree) involved in loop *k*.
- the *c_parent* array refers to the secondary element involved in a loop: *c_parent*[*k*] refers to the secondary element (leaf or not) involved in loop *k*

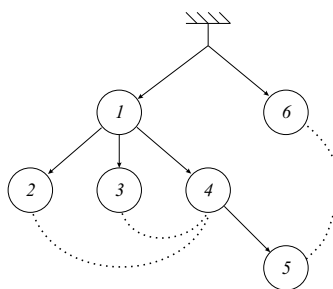


Figure 5.8: Circuit example

For instance, the arrays associated with the circuit represented in figure 5.8 are:

$$\mathit{element} = [1\ 2\ 3\ 4\ 5\ 6] \quad (5.1)$$

$$\mathit{parent} = [0\ 1\ 1\ 1\ 4\ 0] \quad (5.2)$$

$$\mathit{c_element} = [2\ 3\ 5] \quad (5.3)$$

$$\mathit{c_parent} = [4\ 4\ 6] \quad (5.4)$$

As explained in section 4.2, Kirchoff's equations are used to form the algebraic system of equations. Hence, writing Kirchoff's equations corresponds to filling in matrices *A* and *B* from system 4.39.

5.4.2.1 Kirchoff's current equations

For writing Kirchoff's current equation for node *k*, we have to find all the elements connected to the corresponding physical node, including element *k*. The following procedure can then be applied with each node *k* of the circuit:

1. From *parent*, we get all the children of element k , connected to node k .
2. From *c_parent* we obtain the loops involving element k . For each loop j involving element k , *c_element*[j] represents the element associated with k , let's call it the *partner* of element k for loop j , which is also connected to the physical node associated with node k .
3. The Kirchoff's current equation for node k is then written by associating $+1$ to the currents through element k and its partners and -1 to currents through the children of element k .
4. If element k is neither in *parent* nor *c_parent*, we go to the next element.

Figure 5.9 illustrates this procedure for the circuit example of figure 5.8.

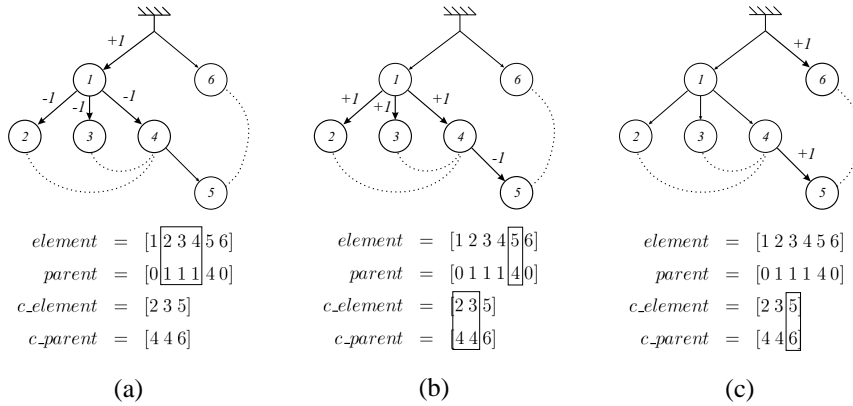


Figure 5.9: Illustration of the Kirchoff's current algorithm

Starting the procedure with element and node 1, we see from the arrays 5.1 to 5.4, that element 1 is parent of elements 2, 3 and 4 but is not in *c_parent*. Thus, the $+1$ and -1 symbols can be associated as in figure 5.9.a.

Elements 2 and 3 being neither in *parent* nor in *c_parent*, we keep going with element 4, which is in *c_parent*, associated with elements 2 and 3 from *c_element*. Element 4 is also the parent of element 5. Thus, elements 2, 3 and 4 receive a $+1$ symbol while element 5 receive a -1 symbol, as shown in figure 5.9.b.

Eventually, element 5 is skipped because not in *parent* nor in *c_parent*, and elements 6 and 5 receive a $+1$ symbol, as shown in figure 5.9.c.

A total of three independent equations were generated for a circuit with 4 nodes, what is in agreement with Kirchoff's theory.

The corresponding current equations are:

$$\begin{aligned}
 i_1 - i_2 - i_3 - i_4 &= 0 \\
 i_4 + i_3 + i_2 - i_5 &= 0 \\
 i_5 + i_6 &= 0
 \end{aligned}$$

5.4.2.2 Kirchoff's voltage equations

For writing Kirchoff's loop equations, we have to find the elements involved in a loop and to see whether they have the same orientation as the circulation associated with the considered loop. Let us remind that each loop j is characterized by two elements k_1 and k_2 , that initiate two separate branches of the loop. These branches get together at the common ascendant for element k_1 and element k_2 . The following procedure is applied in order to detect all the elements involved in each branch of the loop:

1. Branch 1 and branch 2 respectively receive element k_1 and k_2 as first element
2. From *parent* we get the parents k'_1 and k'_2 of elements k_1 and k_2 , respectively
3. While the parents are different, we add element k'_1 in branch 1 and element k'_2 in branch 2 and we look for their parents, elements k''_1 and k''_2 , and so on
4. When the same parent is found for both elements, it corresponds to the common ascendant and the loop is closed. We can associate $+1$ to the voltage drops u_{kk} of elements in one branch and -1 for the other branch.

Obviously, when the root of the tree is reached by one branch, the procedure stops for this branch.

For instance, in figure 5.10.a, a first loop is characterized by nodes 5 and 6 and their successive ascendants: [4 1 0] and 0 respectively. Figure 5.10.b shows a second loop characterized by nodes 2 and 4 whose common ascendant is node 1, as for the third loop, in figure 5.10.c.

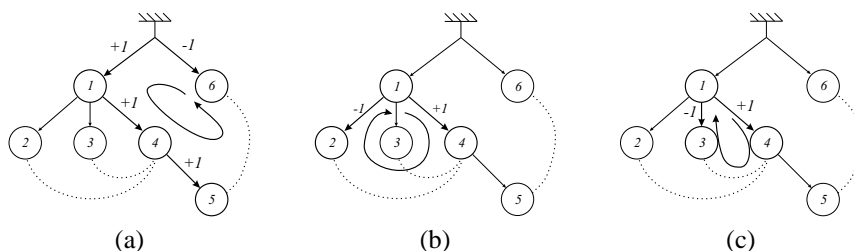


Figure 5.10: Illustration of the Kirchoff's voltage algorithm

The three voltage equations are:

$$\begin{aligned} u_1 + u_4 + u_5 - u_6 &= 0 \\ u_4 - u_2 &= 0 \\ u_4 - u_3 &= 0 \end{aligned}$$

Note: In this first implementation, we assume that the user provides a description of independent loops only. Checking for independence of these loops is straightforward, via a triangularization procedure similar to the one used for constraint detection (see section 4.2.2), and will be implemented in future versions of ElectrAn.

5.4.3 Electran's Symbolic Implementation

Electran [59] software was developed using the same philosophy as Robotran and uses the same symbolic kernel. Nevertheless, some additional features were required by the electrical formalism:

- First, further simplifications, such as common denominator reduction, related to the division were implemented. Indeed, this operator is used very often in the electrical formalism since two systems of equations have to be solved and matrices are to be explicitly inverted (see chapter 4 for details about the implemented formalism). The symbolic triangularization also extensively uses the division operator.
- Second, a new triangularization routine was implemented (with line and column permutation) in order to *symbolically* detect the presence of constraints⁶. This has to be done very carefully since permutations of columns correspond to variables permutations, what is not an easy task in symbolic programming.

As mentioned in section 4.2.3, when applying the coordinate partitioning approach to electrical circuits, the choice of dependent variables is not critical and can be automated; presently, the dependent variables are chosen by selecting the first variable appearing in each constraint. The complete process always leads to a minimal set of first-order ODEs, and “fully-symbolic” generation of the latter is possible, what leads to several advantages:

- The symbolic approach offers the opportunity to select the expressions and to print only the necessary expressions. In the circuit formalism detailed in section 4.2, the resolution of the full algebraic system 4.39, given the algebraic variables y and the sources values z , is necessary before solving the differential system of equation 4.42, although only a few variables ($w^d = Dy$) need to be known. When time simulation is conducted, unnecessary computations can be putted aside and used in a post-process to compute all the electrical variables. A similar gain in the computations was already observed with mechanical formalisms, when projecting the vector equations onto the joint space. The gain in mechanics is around 30% [16], while for electrical circuits, the gain is approximately of 20%. This gain strongly depends on the elements involved in the circuits and on its structure.
- The symbolic implementation also offers the possibility to modify parameters values during the simulation. This feature is very interesting as soon as parameters can change with respect to time or any other variable of the system, what is the case with electromechanical converters: the mechanical position influences the electrical parameters. We decided to let the user mastering the evolution of the parameters values by calling user's function updating the set of parameters before dynamic calculations. This is time consuming and the implementation could of course be optimized by only updating the non constant parameters.

⁶The column permutations only concern the matrix A , as mentioned in section 4.2.2

5.4.4 ElectrAN's Conventions

The classical circuit theory conventions apply here. Let us nevertheless insist on the sign convention chosen for our implementation. The positive sign is assigned for the currents through each element on the basis of the tree representation: *the currents flow from the root (zero voltage node) to the leaves*, as shown in figure 5.7. The passive sign convention is then used to assign the voltage drop positive direction, except for the sources that use an active sign convention. These active and passive conventions are illustrated in figure 5.11, where positive values correspond to the indicated directions.

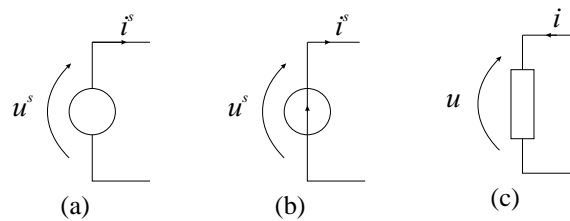


Figure 5.11: ElectrAN's sign conventions: generator (active) signs for voltage sources (a) and current sources (b) and receptor (passive) signs for resistance, capacitor and inductor (c)

Permanent magnets can be, from a physical point of view, considered as flux sources generating fluxes through inductors at their neighborhood. In order to simplify their implementation inside ElectrAN, a permanent magnet k is considered as a current source of value $i_s = 1A$ in series with an inductor. The inductor has mutual influences L_{jk} with each inductor j of the circuit. This is shown in figure 5.12.

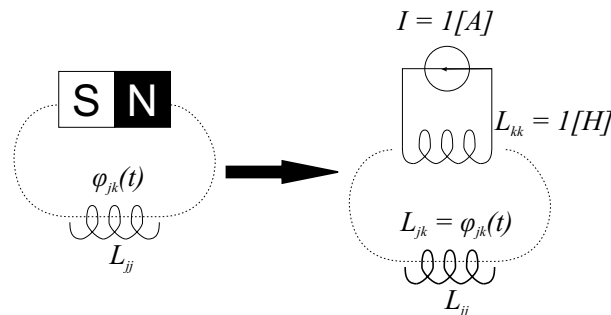


Figure 5.12: Modeling of a permanent magnet

The flux produced by the magnet through inductor j is directly related to the mutual inductance L_{jk} :

$$\varphi_{jk} = L_{jk} i_k = L_{jk} \cdot 1$$

This way of modeling permanent magnets was possible thanks to the constraints reduction procedure that we implemented. Indeed, connecting an inductor introduces a supplementary state variable that is constrained by the current source in series, and thus no additional dynamics is introduced by this supplementary circuit. This model of permanent magnets was also motivated by the possibility to reuse standard Electran elements.

5.4.5 Electran's input

In order to generate the dynamic equations of an electrical system, Electran requires a full description of the circuits involved. Several circuits may be involved in the same model, especially when electromechanical converters are modeled. For example, for an electrical motor, an equivalent circuit exists for the stator and the rotor. These circuits are physically separated but interact through mutual influences. The input file will contain the description of each circuit and the possible mutual influences.

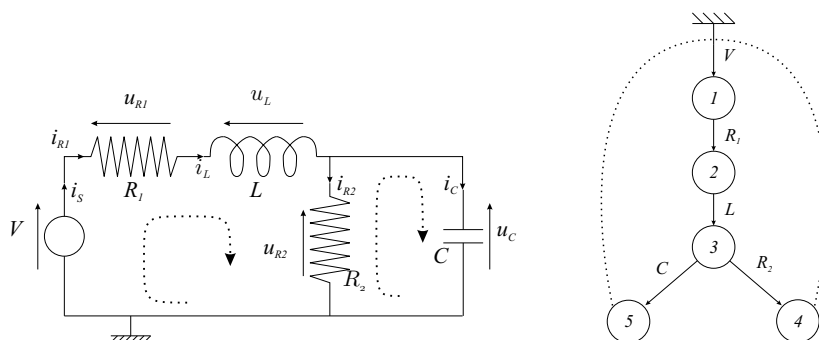


Figure 5.13: Example of a simple circuit and the corresponding tree-graph

Figure 5.13 shows a very simple circuit and the corresponding tree that will be used to write the input file. The latter is detailed here for this example and starts with a title section and three numbers α , $nbcircuit$ and $nbmagnet$ respectively determining:

- whether an electromechanical interaction takes place or not. The electrical system interacts with a mechanical system if α is non zero and the latter then indicates the index of the joint variable influencing the electrical parameters.

In this implementation, we assumed that only one joint variable s_α^m influences the electrical parameters. This is the case most of the time when considering electromechanical converters which have only one degree of freedom. Considering more mechanical variables influences could nevertheless be useful and easily implemented.

- the number of circuits
- the number of permanent magnets

```

% Description of the system
%-----
% Electromechanical interaction
0 ( $\alpha$ )
%Number of circuit
1 (nbcircuit)
%Number of permanent magnets
0 (nbmagnet)

```

For each circuit k , the input file details the number of elements $nelem(k)$ and for each element, the index of its parent and its type: R, L, C, U or I for respectively a resistor, an inductor, a capacitor, a voltage source and a current source.

```

%-----
% Description of circuit  $k$ 
%-----
%Number of element
5 ( $nelem(k)$ )
%Element 1
0 (parent index)
U (type of element)
%Element 2
1
R
%Element 3
2
L
%Element 4
3
R
%Element 5
3
C

```

After describing the elements of circuit k , the loops are described by first giving their number $nloops(k)$ and then by describing each loop with the two nodes involved

```

%Number of loops
2 ( $nloops(k)$ )
%Loop 1
4
0
%Loop 2
5
0

```

Eventually, the internal mutual influences are described with firstly their number $nmutint$, secondly their type (L or C) and finally the index of the elements influ-

enced by each other. In the considered example there is no mutual influence and the description of the circuit ends with the following lines:

```
% Internal mutual influences inside circuit 1
%Number of influences
0
```

When several circuits are involved, they are described after each other and possible external mutual influences between circuits are given afterwards.

Examples of input files are given for a three phase induction motor in Appendix D.2, and for an electromagnetic speaker in Appendix E.2.

5.4.6 Electran's Global Algorithm

The global algorithm, corresponding to the formalisms discussed in chapter 4, is shown in figure 5.14, and the different steps are detailed below:

- Electran starts by reading the input file to obtain general information on the electrical circuits
- Electran generates the algebraic and differential system of equations (respectively 4.39 and 4.44) by filling matrices A , B , C , D , M^e and E for the entire system. Therefore, the program performs the following steps for each circuit involved in the system:
 - Kirchoff's equations are written, that is 0, 1 and -1 symbols are placed inside matrices A_i and B_i .
 - Constitutive equations for resistor and sources are written, that is -1 and R symbols are placed inside matrices A_i , B_i and C_i
 - Constitutive equations for inductors and capacitors are written, that is L and C symbols are placed inside matrix M_i^e , and DL and DC symbols, respectively representing the derivative of the L and C matrices with respect to the mechanical variable s_α^m participating in the electromechanical conversion, are placed inside matrix E_i . Matrix D_i is also filled in by 1 and 0 symbols.
 - Possible internal mutual influences are considered and matrices M_i^e and E_i are filled in accordingly.

The A_i matrices are then gathered inside matrices $A = \begin{pmatrix} \ddots & 0 & 0 \\ 0 & A_i & 0 \\ 0 & 0 & \ddots \end{pmatrix}$,

and similarly B , C , D , M^e and E for the all system are constructed.

- Possible external mutual influences between the circuits are considered and matrices M^e and E are filled in accordingly.

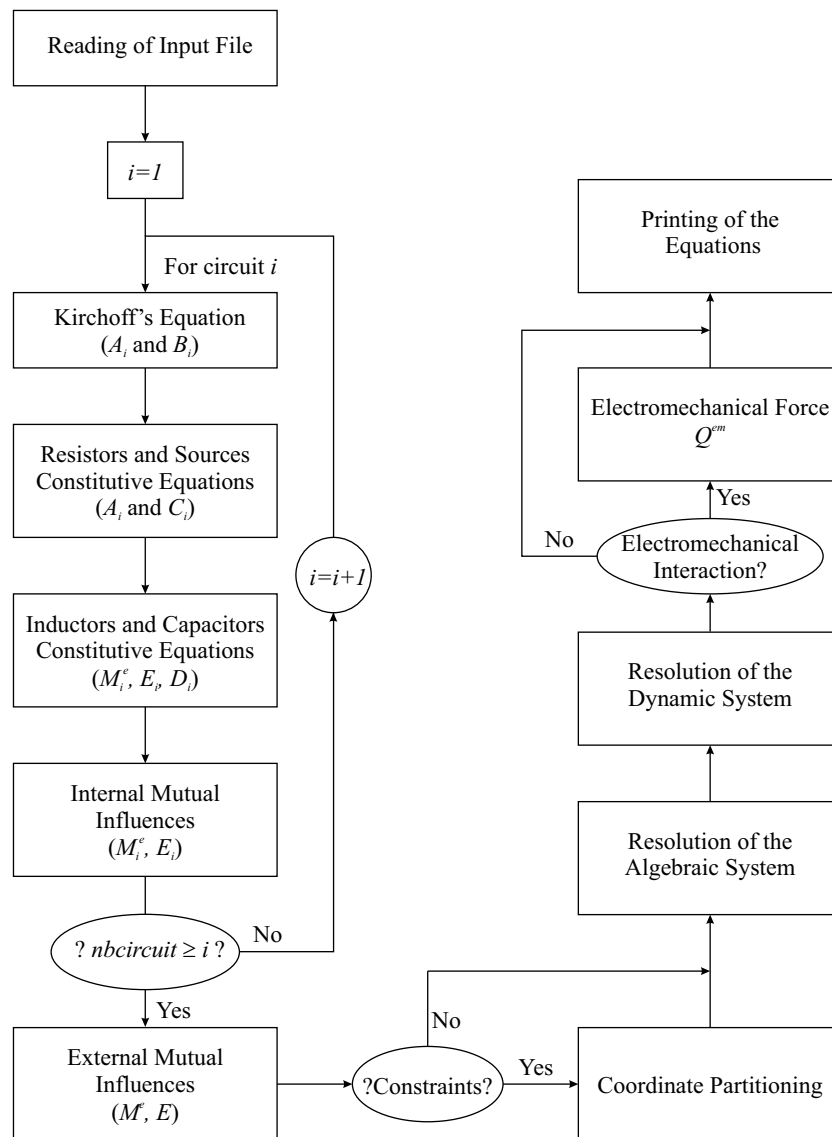


Figure 5.14: Global algorithm for ElectrAN

- Electran then checks whether there exist constraints or not by performing the triangularization procedure discussed in section 4.2.2.
- If constraints are detected, the systems of equations are updated according to the electrical coordinate partitioning procedure explained earlier (see section 4.2.3).
- Electran solves the algebraic system of equations before the dynamic one.
- Before printing the equations, if some electromechanical interaction exists, the mechanical torque/force is computed from expression 4.97.

5.4.7 Electran's output

So far, Electran was designed to produce the minimal set of first order differential equations to be solved in order to simulate the behavior of the circuit or the machine. Hence, Electran produces one symbolic function, called "circuit_dyn", that takes the state variables w (or w_u) and computes the derivative of the state variables $\dot{w} = \frac{dw}{dt}$ (or \dot{w}_u), using the circuit based formalism and the electrical coordinate partitioning presented in chapter 4. The other inputs to "circuit_dyn" are the time t and the mechanical joint variables s^m and \dot{s}^m .

Before doing any computation, "circuit_dyn" calls several user functions to update the parameters values, which can be function of time and/or mechanical variables. These user functions are also part of Electran's output and are called:

- "projet_Su.*", which updates the voltage sources values;
- "projet_Si.*", which updates the current sources values;
- "projet_R.*", which updates the resistances R_k ;
- "projet_L.*", which updates the self-inductances L_{kk} ;
- "projet_DL.*", which updates the first derivative $\frac{\partial L_{kk}}{\partial s_\alpha^m}$ of the self inductances with respect to the mechanical variable s_α^m when electromechanical conversion takes place;
- "projet_LM.*", which updates the mutual-inductances L_{jk} ;
- "projet_DLM.*", which updates the first derivative $\frac{dL_{jk}}{ds_\alpha^m}$ of the mutual-inductances with respect to the mechanical variable s_α^m when electromechanical conversion takes place;
- "projet_C.*", which updates the self-capacitances C_{kk} ;
- "projet_DC.*", which updates the first derivative $\frac{dC_{kk}}{ds_\alpha^m}$ of the self-capacitances with respect to the mechanical variable s_α^m when electromechanical conversion takes place;

- “*projet_Fl.**”, which updates the fluxes produced by the permanent magnets through the inductors φ_{jk} ;
- “*projet_DFl.**”, which updates the first derivative $\frac{d\varphi_{jk}}{ds_\alpha^m}$ of the fluxes produced by the permanent magnets through the inductors with respect to the mechanical variable s_α^m when electromechanical conversion takes place;

Note: Depending on the circuit, only part of these user functions are necessary.

For the example shown in figure 5.13, the output file generated by ElectrAn in C-code is shown in figure 5.15. The chosen state variables are $w = (i_L \ u_C)^T$.

As can be observed, the user has to enter the function defining the value of the inductances and capacitances but also their derivative with respect to the only mechanical variable involved in the electromechanical conversion.

After these parameter updates, the algebraic and dynamic systems are solved, what leads to the first-time derivative of the electrical state variables \dot{w} .

As for Robotran, the output format can be chosen between several possibilities such as Matlab or C syntax.

When an electromechanical interaction exists, “circuit_dyn” also computes the electromechanical force Q^{em} after the resolution of the dynamic system. This can be observed in the function generated for the electromagnetic speaker in Appendix E.3.

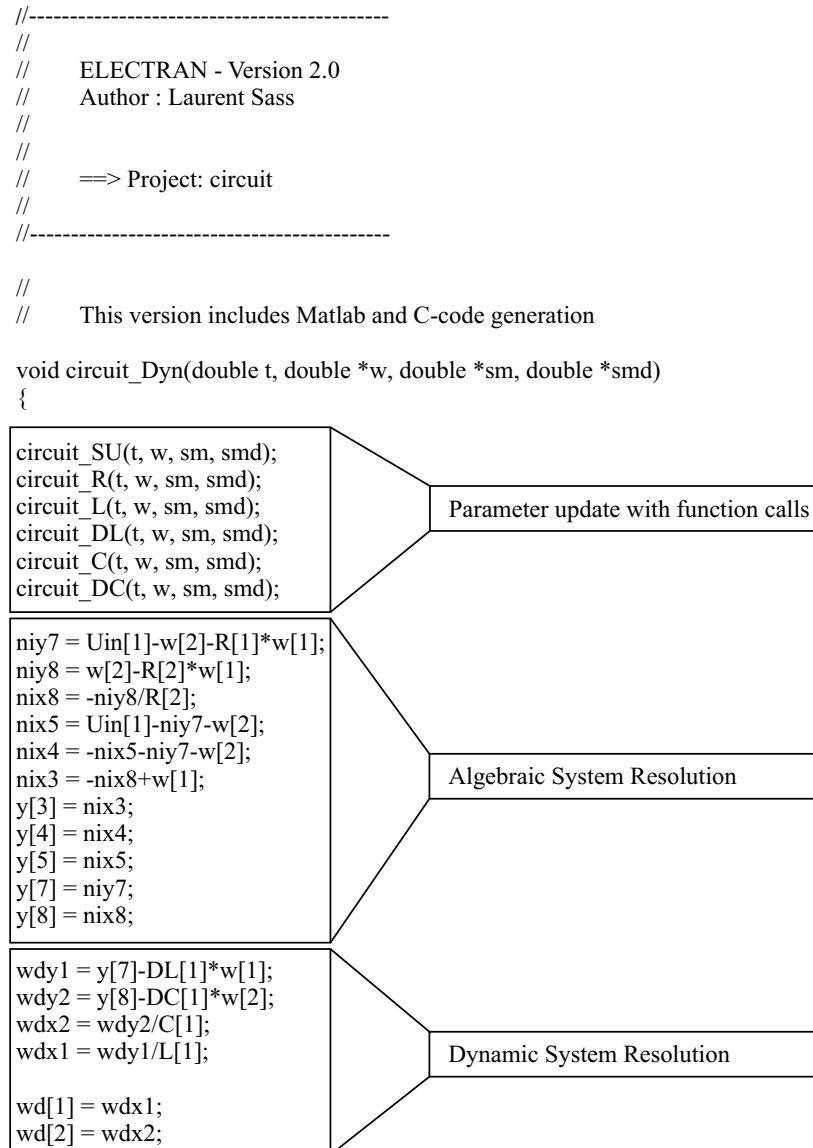


Figure 5.15: Electran output file for a simple circuit model

Part III

Applications

Abstract

In the third part of this thesis, we will demonstrate our ability to deal with complex multibody systems coupled with electromechanical converters, what was the main objective of this thesis, as explained in the introduction (chapter 1).

Very simple applications are firstly considered, in chapter 6 in order to validate the implementation, especially for the electrical software, and the assembling technique for electromechanical models. Four electrical circuits and two electromechanical systems are considered in order to compare our program with other modeling and simulation tools. Among these applications, a three phase actuator model is compared with experimental measurements on a motor to ensure that our models are correct. We end this validation by considering simple permanent magnets applications.

In chapter 7, two complex industrial applications are considered, which are much more representative of the systems aimed by this work. The first one consists of a flexible barrier used for access control to parking lots. The flexibility of the barrier introduces several mechanical natural frequencies among which one is close to the frequency of the source controlling the electrical three phase induction motor driving the barrier. In such situations, a tight electromechanical interaction takes place and multidomain refined models are necessary to simulate the system. The second application is a railway bogie also driven by three phase actuators. Railway bogies are typical applications in multibody dynamics but research in this area almost never considers the electrical motors coupled with the wheelsets. This was understandable as long as most of them were driven by DC-motors which have a smooth behavior. Nowadays, DC motors are sometimes replaced by more robust three phase induction converters inducing much more torque oscillations during starting and braking phases. It must be checked, by simulation, whether these oscillations can generate vibrations in the structure, what could lead to fatigue problems.

We do not want to claim the superiority of our approach but we observed in [60] that applications such as the flexible barrier or the railway bogie considered here cannot always be modeled with existing tools.

6

Validation Applications

Contrary to Robotran which has already been validated via many applications and benchmarks [16, 37], Electran is a new software and a validation of this tool had to be performed. The simultaneous use of both tools and their interfacing according to chapter 4 was also verified.

In this research, the Matlab/Simulink environment has been chosen to solve and time integrate the generated equations and to analyze the results. It provides robust and efficient solvers for differential equations and user-friendly post-processing tools (plots, frequency analysis, etc.). Although the Matlab environment was used, most of the models were generated in C-Code and compiled as cmex S-functions within Simulink¹. It is important to remind that, according to our strategy, we generate a global symbolic model outside of any numerical environment and Matlab/Simulink only serves as a numerical integration tools. It has the advantage of providing robust and efficient integration routines for ODE systems.

Figure 6.1 shows the structure of our Simulink projects, that involve *only one* dynamic block containing the full symbolic model. The *initialization block* simply set the initial values and retrieve information on the system (number of bodies, number of electrical elements,...). The input to the dynamical blocks are:

- Q_m that denotes the joint forces applied on the system, in addition to those of electromechanical origin (e.g. controller forces);

¹The Matlab environment is quite slow and compiling the models drastically reduces the computation time.

- *smuc* that denotes the time evolution of possible driven mechanical joints (e.g. crank constant velocity rotation);
- *GenInput* that offers a scalar input that can be used for general purpose;
- *ElecInput* that represents the electrical sources time evolution.

The Simulink dynamic block returns the electrical variables (*ElecVar*), which are the voltage drop and current associated with each element, and the mechanical variables (joint coordinates) at position (*sm*), velocity (*smd*) and acceleration (*smdd*) level.

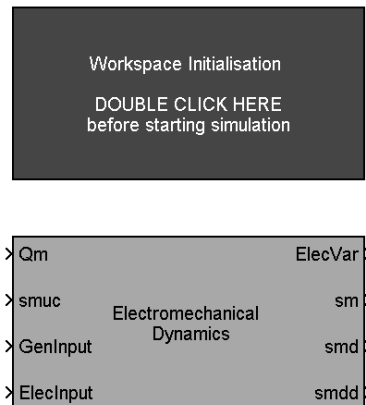


Figure 6.1: Simulink project for electromechanical MBS

To validate the electrical model generator Electran, different pure electrical circuits have been tested. The simulation results have been compared to those obtained with the well-known electrical circuit simulation tool: PSpice software ². The Power System Blockset ³ developed by Mathworks, Inc. for modeling electrical devices with Simulink was also used.

Two electromechanical systems are used for testing the electromechanical interconnection. The second one involves comparison with an experimental test bench.

Let us mention that when comparing the results from different modeling tools, we always use the same numerical values for the different models. Hence, the results should perfectly match (at the numerical integration precision). When possible, this was verified but the plots given below will always present the curves obtained with the different tools, sometimes superimposed to show that no differences exist.

²<http://www.orcad.com>

³<http://www.mathworks.com>

6.1 Electrical Circuit Examples

We start with a very simple RLC filter to illustrate how to use Electran and how the generated equations can be integrated afterwards using Matlab/Simulink. We then continue with two circuits involving constraints. We end this validation on purely electrical systems by considering a circuit with variable resistance, to illustrate the benefits of the symbolic approach.

6.1.1 Example 1 - RLC 2nd order filter

Description The considered circuit is represented in figure 6.2. It was already used throughout the description of the computer implementation in chapter 5.

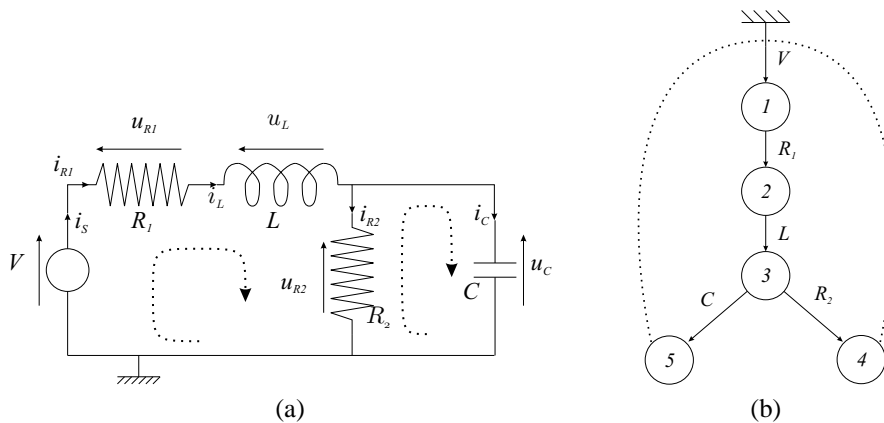


Figure 6.2: Example 1: RLC filter and the corresponding tree-graph

The values for the different parameters are:

$$\begin{aligned}
 R_1 &= 5 \quad \Omega \\
 R_2 &= 7 \quad \Omega \\
 L &= 2 \quad mH \\
 C &= 3 \quad \mu F \\
 U &= 12 \quad V
 \end{aligned}$$

This circuit contains two loops and involves two state variables $w = \begin{pmatrix} i_L \\ u_C \end{pmatrix}$, which are independent since there exists no loop of capacitors and no cutset of inductors.

Model The dynamic equations of this simple system can be easily written by hand. Starting from the five Kirchoff laws:

$$\begin{aligned} V &= u_{R1} + u_L + u_{R2} \\ u_{R2} &= u_C \\ i_s &= i_{R1} \\ i_{R1} &= i_L \\ i_L &= i_{R2} + i_C \end{aligned}$$

and the constitutive equations:

$$\begin{aligned} u_{R1} = R_1 i_{R1} & \ ; \ \frac{d(Li_L)}{dt} = u_L \\ u_{R2} = R_2 i_{R2} & \ ; \ \frac{d(Cu_C)}{dt} = i_C \end{aligned}$$

we obtain

$$\begin{aligned} \frac{d(Li_L)}{dt} &= V - R_1 i_L - u_C \\ \frac{d(Cu_C)}{dt} &= i_L - \frac{u_C}{R_2} \end{aligned}$$

These equations can be rewritten in terms of the state variable vector w :

$$\begin{aligned} \frac{dw(1)}{dt} &= \frac{1}{L} \left(V - R_1 w(1) - w(2) - \frac{dL}{dt} w(1) \right) \\ \frac{dw(2)}{dt} &= \frac{1}{C} \left(w(1) - \frac{w(2)}{R_2} - \frac{dC}{dt} w(2) \right) \end{aligned}$$

Electran was used to generate the dynamic equations that are shown in figure 5.15. They can be rewritten as:

$$\begin{aligned} wd[1] &= (v_s[1] - w[2] - R[1] * w[1] - DL[1] * w[1])/L[1]; \\ wd[2] &= ((-w[2] - R[2] * w[1])/R[2]) - DC[1] * w[2]/C[1]; \end{aligned}$$

which are identical to those previously obtained by hand.

Results We simulated the transient response when switching on the voltage source. The initial conditions are:

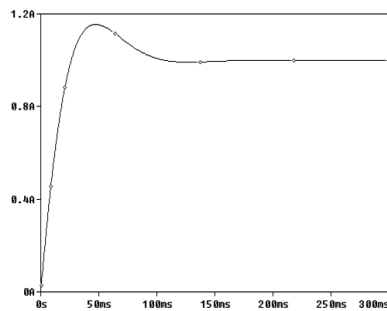
$$w_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (6.1)$$

In steady state, inductor and capacitor respectively correspond to short-circuit and open circuit. The steady state circuit is thus characterized by

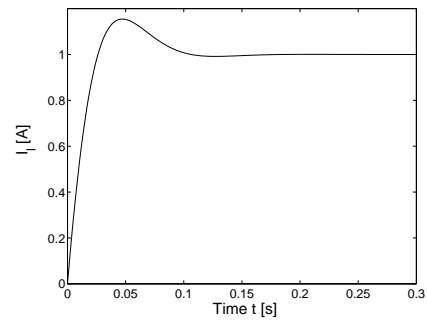
$$\begin{aligned} i_s &= 1A; & V &= 12V \\ i_{R1} &= 1A; & u_{R1} &= 5V \\ i_{R2} &= 1A; & u_{R2} &= 7V \\ i_L &= 1A; & u_L &= 0V \\ i_C &= 0A; & u_C &= 7V \end{aligned}$$

Figure 6.3 shows the currents flowing through the inductor i_L (plots a and b) and the capacitor i_C (plots c and d). Plots a and c were obtained using Pspice while plots b and d were obtained using Electran and Matlab ode45 function as time-integrator (based on the Runge-Kutta method).

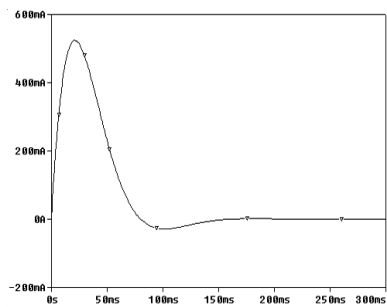
The results match perfectly and the simulation time is around 0.24 seconds with our approach.



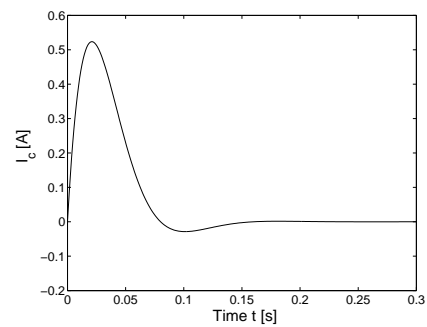
(a) PSPICE



(b) Electran



(c) PSPICE



(d) Electran

Figure 6.3: Example 1 - Comparison of simulation results: i_L (plots a and b), i_C (plots c and d)

6.1.2 Example 2: circuit with one constraint

Description The considered circuit is represented in figure 6.4.

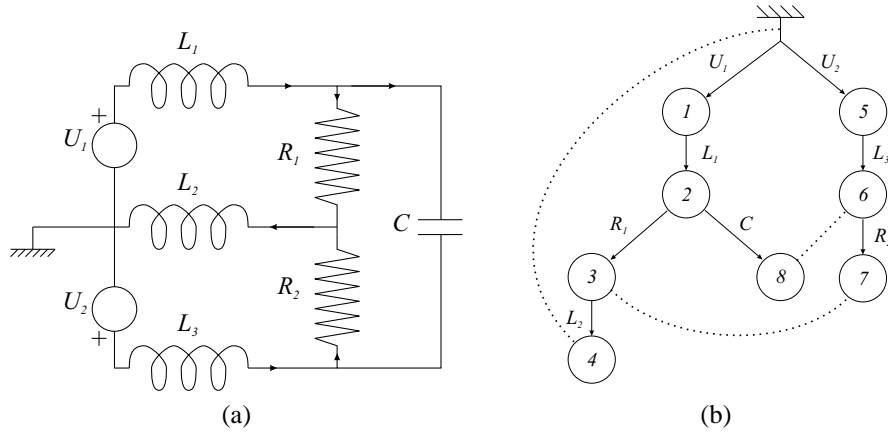


Figure 6.4: Example 2 - Simple circuit with one constraint (a) and the corresponding tree (b)

The values for the different parameters are:

$$\begin{array}{llll}
 U_1 & = & 12 & V \\
 U_2 & = & 10 & V \\
 R_1 & = & 5 & \Omega \\
 R_2 & = & 10 & \Omega \\
 L_1 & = & 10 & mH \\
 L_2 & = & 30 & mH \\
 L_3 & = & 50 & mH \\
 C & = & 8 & mF
 \end{array}$$

Model The tree representation of the circuit of figure 6.4.a is given in figure 6.4.b. According to our formalism (see section 4.2), there are four state variables for this circuit: i_{L1} , i_{L2} , i_{L3} and u_C . The structure of the circuit is such that a constraint exists between these state variables. Electran automatically detects the presence of this constraint (triangularization procedure, see section 4.2.2), whose expression is automatically derived as:

$$i_{L1} - i_{L2} + i_{L3} = 0 \quad (6.2)$$

This expression can also be obtained by cutting the circuit through the three inductors and writing the corresponding cutset equation, as done in linear graph approach.

Since we decided to avoid the mix of differential and algebraic equations because of numerical problems that can appear with time integration of DAE systems, the constraint is automatically eliminated by selecting i_{L1} as dependent variable: there remains only three differential equations. This automatic reduction of the system of equations is achieved symbolically, on the basis of the coordinate partitioning technique presented in section 4.2.3, what reduces the number of numerical operations and results in shorter computation times.

Let us point out that the Power Block Set developed for Matlab/Simulink does not accept to model this kind of electrical circuits, because of the constraint.

Results The transient response of the circuit has been simulated with PSpice and Electran. The results are presented in figure 6.5 where we decided to plot the current through inductors L_2 and L_3 . Once again, our results match with those obtained using PSpice and the simulation time is around 0.8 seconds with our model.

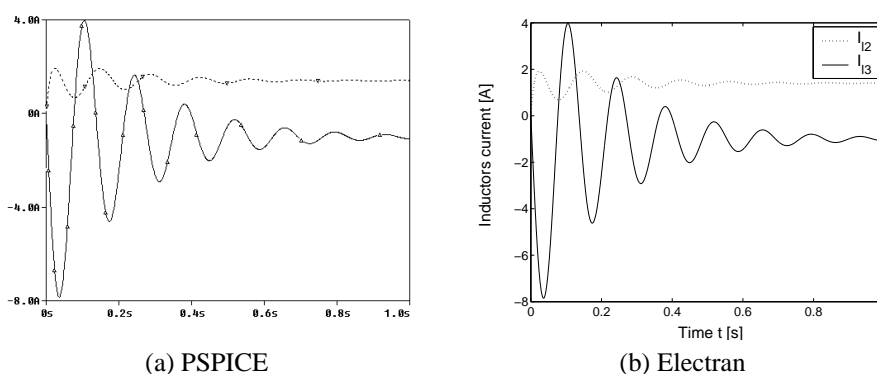


Figure 6.5: Example 2 - i_{L_2} and i_{L_3} time history

The constraint has been plotted in figure 6.6. We should obtain 0 during the simulation, what is the case for Electran (at computer double precision accuracy $\approx 10^{-15} A$) but not exactly with Pspice ($\approx 10^{-7} A$). No information could be found about the integrator precision used by Pspice, so that no further explanations can be given.

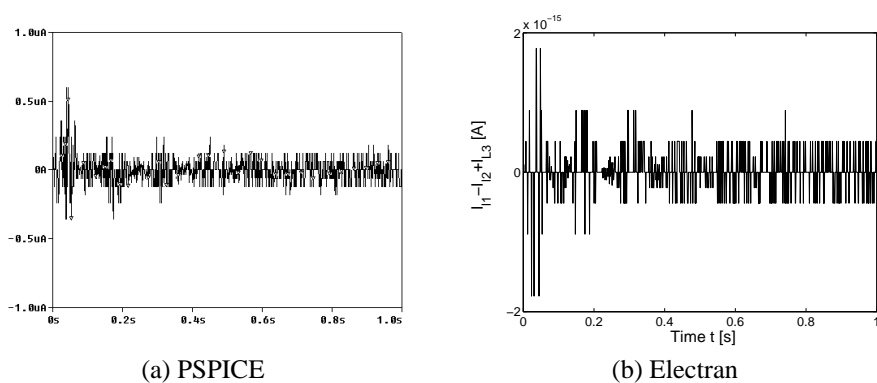


Figure 6.6: Example 2 - Constraint violation

6.1.3 Example 3: triangle circuit

Description The considered circuit is represented in figure 6.7.

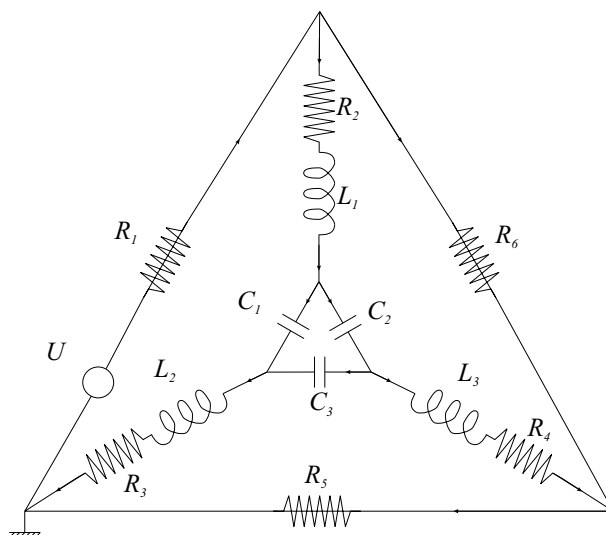


Figure 6.7: Example 3

The values for the different parameters are:

$$\begin{array}{llll}
 R_1 & = & 2 & \Omega & L_1 & = & 0.1 & mH \\
 R_2 & = & 3 & \Omega & L_2 & = & 0.7 & mH \\
 R_3 & = & 15 & \Omega & L_3 & = & 0.9 & mH \\
 R_4 & = & 12 & \Omega & C_1 & = & 0.02 & \Omega \\
 R_5 & = & 7 & \Omega & C_2 & = & 0.03 & \Omega \\
 R_6 & = & 8 & \Omega & C_3 & = & 0.05 & mF
 \end{array}$$

Model Despite the presence of 6 dynamic elements (3 inductors and 3 capacitors), only 4 independent differential equations will be obtained thanks to our automatic detection and reduction of the constraints. The latter are:

$$\begin{array}{rcl}
 i_{L1} + i_{L2} + i_{L3} & = & 0 \\
 u_{C1} - u_{C2} - u_{C3} & = & 0
 \end{array} \tag{6.3}$$

The procedure, based on the coordinate partitioning presented in section 4.2.3, being fully automatized, we must only specify the elements and their interconnection. Electran then generates the reduced set of ODEs in terms of the independent variables. This procedure always leads to the minimal number of ODEs.

Results The source U will be switched on: $U = 12V$, at time $t = 0$, and the transient response of the circuit is simulated using PSpice and Electran. Results are identical, as shown in figure 6.8 where we plotted the currents through inductors L_2 and L_3 .

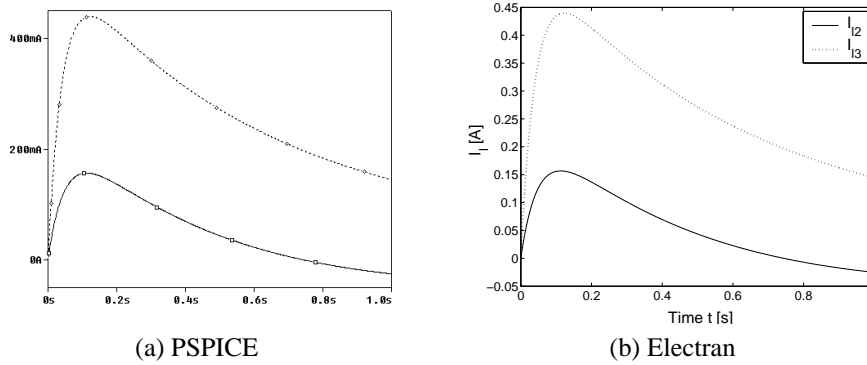


Figure 6.8: Example 3 - i_{L2} and i_{L3} time history

The voltage and current constraints have also been computed. In the present case, they are perfectly satisfied (=0) either with our approach or with PSpice.

6.1.4 Example 4 - Class-E Amplifier

The benchmark presented in this section was proposed by the Vienna University of Technology, more precisely by the Simulation Department Simtech, but is extracted from [43]. A detailed description of the benchmark is available on their web page that also provides many different solutions proposed by researchers around the world⁴. A copy of the benchmark description and of three solutions proposed by Breitenacker [8], Viertl [74] and Rada [50] are provided in Appendix B.

Description The considered circuit is represented in figure 6.9. It contains a time dependent resistor $R(t)$. The equations being generated symbolically, time-dependent parameters can be easily updated at each computation via calls to user functions at the beginning of each model evaluation.

The values for the different parameters are:

$$\begin{aligned}
 L_1 &= 79.9 \quad \mu H \\
 L_3 &= 232 \quad \mu H \\
 C_2 &= 17.9 \quad nF \\
 C_4 &= 9.66 \quad nF \\
 R_L &= 52.4 \quad \Omega \\
 U_1 &= 5 \quad V
 \end{aligned}$$

⁴<http://euosim.tuwien.ac.at/comparisons/c3/definition/c3def.html>

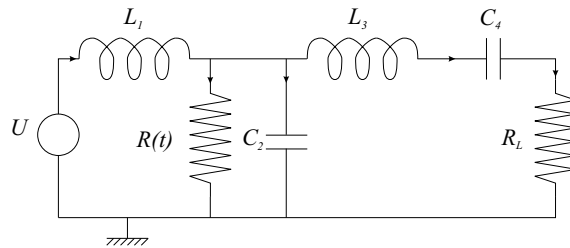


Figure 6.9: Example 4 - Class-E Amplifier

The time dependent resistor $R(t)$ models the active device acting as a switch with an ON-resistance of 0.05Ω and an OFF-resistance of $5M\Omega$. The evolution of $R(t)$ is shown in figure 6.10, where TRF denotes the rise/fall time.

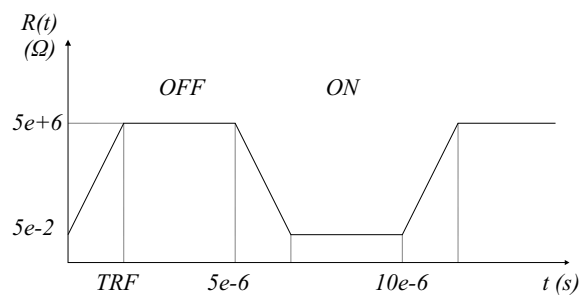


Figure 6.10: Example 4 - Evolution of the time-dependent resistance

Model The model was easily generated with Electran in which resistor R was modeled as a time-varying element, with the following user function (in C code):

```
void circuit_R(double t, double *w, double *sm, double *smd){
double tsim = t;
double trf = 1e-15;

while(tsim >= 10e-6) tsim = tsim - 10e-6;

if (tsim <= trf) R[1] = (5.0e6 - 5.0e-2) / trf * tsim + 5.0e-2;
else if (tsim <= 5e-6) R[1] = 5.0e6;
else if (tsim <= (5e-6 + trf)) R[1] = -(5.0e6 - 5.0e-2) / trf * tsim + 5.0e6;
else if (tsim <= 10e-6) R[1] = 5.0e-2;

R[2] = 52.4;
}
```

Time-varying elements are necessary as soon as electromechanical actuators are considered. The three phase actuator considered later on is another example where parameters vary with time and more precisely depend on the mechanical configuration.

Because of the very fast commutations between the ON and the OFF states, this system is also very demanding from a numerical point of view: robust and efficient integrators are necessary.

Results The benchmark proposes to first analyze the transient response. Figures 6.11 and 6.12 represent the evolution of the current through the time dependent resistor $R(t)$ and the voltage drops across the capacitors, respectively. Plots a were obtained by Breitenacker [8] (see appendix B) and are very similar to plots b, where our results, obtained with Electran, are shown. The latter are also similar to those proposed by Viertl [74] and Rada [50] (see Appendix B).

Our computation time for the transient simulation (figures 6.11 and 6.12) was around 2 seconds (with a processor PIII at 600 MHz). Breitenacker did not provide information about his simulation time but Viertl [74] and Rada [50] respectively announce 20 seconds and 35 seconds with a little bit slower computers (Celeron 500 MHz and PII 400 MHz) (see appendix B). As mentioned by Rada, calling a function at each time step to update the resistance value is time consuming. In our method, thanks to our symbolic approach, we can generate C-code that can be compiled, even within the Matlab environment, making our models much faster. This illustrates the intrinsic portability of the symbolic approach.

Steady-state simulations have also been conducted and led to figure 6.13, which represents the phase plane curves of $\frac{di_{l3}}{dt}$ as a function of i_{l3} for different values of the rise/fall time TRF : $10^{-7}s$, $10^{-9}s$, $10^{-11}s$ and $10^{-15}s$. This was requested by the benchmark's authors in order to highlight the difference appearing in the steady-state phase plane curve for $TRF = 10^{-7}s$. Once again the results from Electran and Breitenacker's solution are very similar.

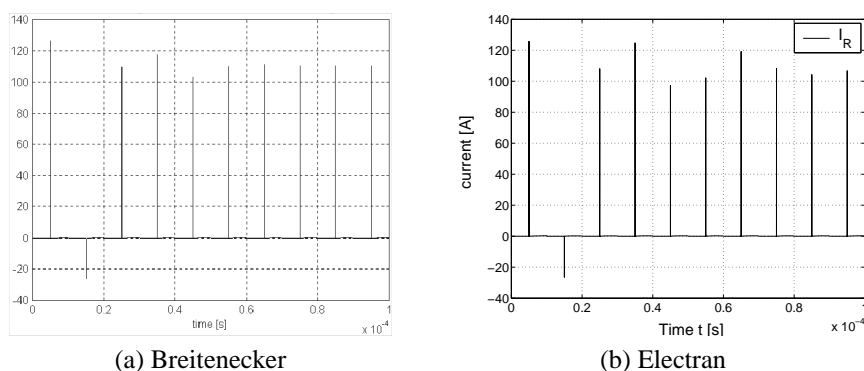
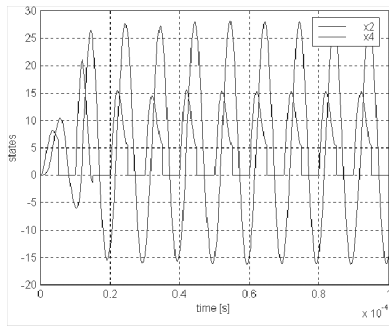
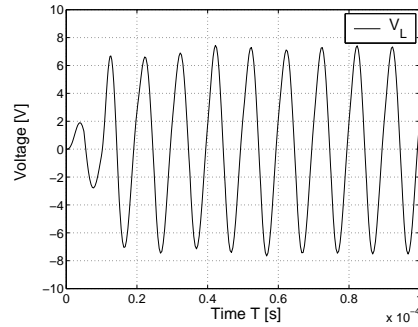


Figure 6.11: Example 4 - Evolution of the current i_R through the time-varying resistor

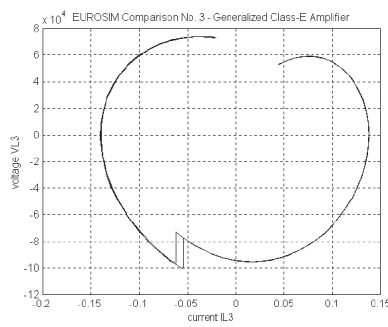


(a) Breitenecker

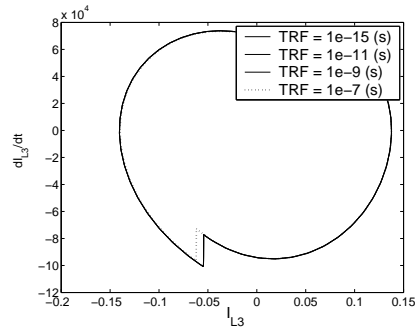


(b) Electran

Figure 6.12: Example 4 - Evolution of the voltage drop across the capacitors u_{C1} and u_{C2}



(a) Breitenecker



(b) Electran

Figure 6.13: Example 4 - Full cycle in steady state

6.2 Scara Robot Driven by DC Motors

Description This benchmark was also proposed by the Vienna University of Technology. It consists of a Scara type robot (see figure 6.14), whose joints are driven by DC-motors⁵. The two first joints are revolute, with parallel axes aligned with the Z-axis. The third joint is prismatic along the Z-axis. A full description of the robot is given in appendix C.

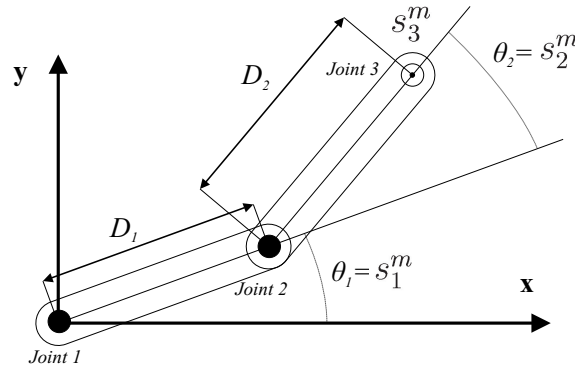


Figure 6.14: Electromechanical Example - A 3 d.o.f. Scara robot

Model The three DC-motors driving the robot were modeled on the basis of the equivalent circuit represented in figure 6.15. The emf_i voltage source represents the electromotive voltage, function of the joint velocity \dot{s}_i^m , where $\dot{s}_i^m = \dot{\theta}_i$ for $i = 1, 2$. It was modeled as a velocity dependent voltage source: $emf_i = k_{T_i} * \dot{s}_i^m$. Similarly to what was done in example 4, a user function is used to define the dependency of the voltage source. The electromechanical torques $Q_j = k_{T_j} \dot{i}_{L_j}$ are applied as joint torques on the mechanical multibody subsystem, consisting of 3 interconnected bodies: the two arms and the tool of the robot, as indicated in figure 6.15.

Electran generates the following dynamic equation for each of the motor:

$$L \frac{di_{L_i}}{dt} = U_i - emf_i - R_i i_{L_i} \text{ where } i = 1, 2 \text{ or } 3 \quad (6.4)$$

The benchmark imposes extreme values for the electromechanical torque, proportional to the current through the motor, and thus the current has to be limited to extreme values i_{max} and $-i_{max}$ (see the benchmark definition in appendix C). In order to take into account the effects of this limitation, we have to impose $\frac{di_{L_i}}{dt} = 0$ either when the current i_{L_i} reaches i_{max} and wants to keep raising ($\frac{di_{L_i}}{dt} > 0$) or when it reaches $-i_{max}$ and wants to keep lowering ($\frac{di_{L_i}}{dt} < 0$). The computation of $\frac{di_{L_i}}{dt}$ was

⁵<http://eurosim.tuwien.ac.at/comparisons/c11/definition/c11def.html>

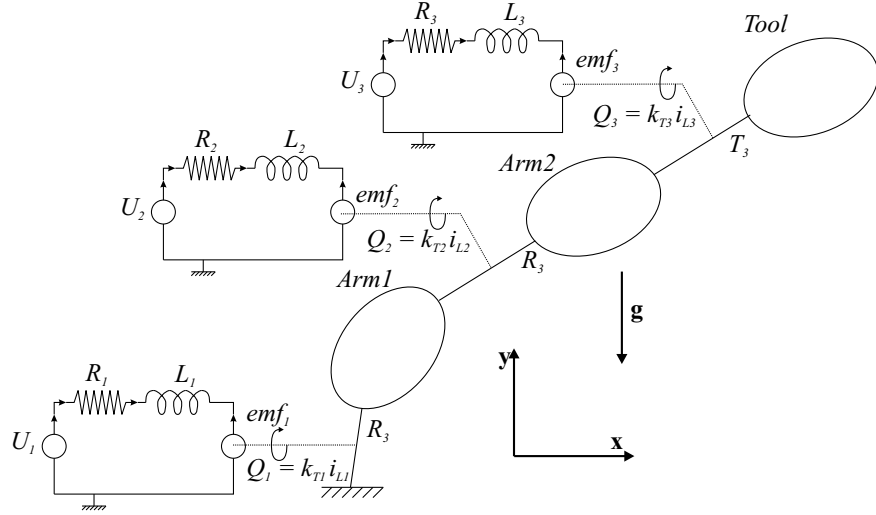


Figure 6.15: Scara robot - DC-motor equivalent circuits and MBS structure

thus complemented as follows:

$$\begin{aligned} \frac{di_{L_i}}{dt} &= \frac{1}{L_i} U_i + emf_i - R_i i_{L_i} \quad (\text{generated by Electran}) \\ \text{if } (|i_{L_i}| > i_{max} \ \& \ (\frac{|i_{L_i}|}{i_{max}} \cdot \frac{di_{L_i}}{dt}) > 0) \\ \frac{di_{L_i}}{dt} &= 0; \end{aligned} \quad (6.5)$$

A simple PD controller is coupled with each motor in order to control the motion of the robot. This controller is modeled directly in the voltage source constitutive equation and is implemented in the user function updating the source value:

$$U_i = P_i (s_{di}^m - s_i^m) - D_i \dot{s}_i^m \quad (6.6)$$

where P_i and D_i are the proportional and derivative gains, respectively, and s_{di}^m is the desired position. The output of the controller, which is the input voltage to the motor, is limited by a maximum value U_{imax} (see the benchmark definition in appendix C).

Note: The controller has been modeled within the constitutive equation of the voltage source. Nevertheless, control engineer usually consider the controller as a distinct unit. This would be possible with our approach, but the risk of non-synchronized evaluations of the model and the controller output exist.

Results According to the benchmark definition, a point-to-point motion was simulated, starting from $s_1^m = s_2^m = s_3^m = 0$ to $s_1^m = s_2^m = 2 \text{ rad}$, $s_3^m = 0.3 \text{ m}$.

Figure 6.16 shows the time evolution of the position computed with Electran's model and the approach based on block diagrams, proposed by Scheickl and Lingl [61].

Figure 6.17 shows the evolution of the currents flowing through the actuators. Our results perfectly match those from Scheikl and Lingl. Forsthuber and Ecker proposed a solution which uses Modelica [21]. Their results are identical to those shown here and are given in Appendix C, together with Scheikl and Lingl's solution.

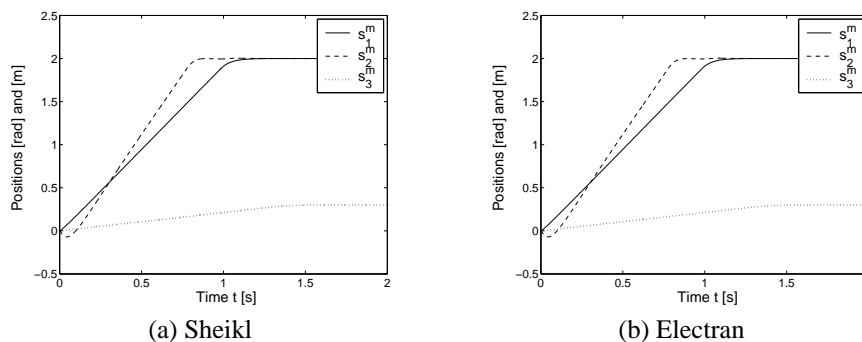


Figure 6.16: Scara Robot - Evolution of the joint positions

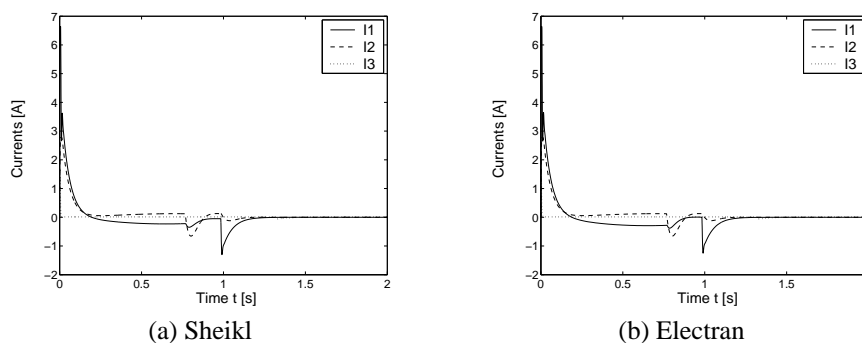


Figure 6.17: Scara Robot - Evolution of the currents through the motors

6.3 Three phase induction motor

6.3.1 Description of the system

For this example, we decided to compare our simulation results with measurements on an experimental bench. The latter simply consists of a three phase asynchronous motor fed by the three phase network through an industrial control unit:

- The motor is a 370W three phase induction motor with squirrel cage rotor. It is characterized by a very high statoric resistance as we shall see later on.

- The control unit is a Microverter D2.5, builded by AEG. It consists of a three phase inverter connected to the network. This inverter produces Pulse Width Modulation (PWM) signals for a U/F control, which is an equilibrated set of three phase voltages characterized by a frequency rising from f_{min} to f_{max} over T_r seconds, as shown in figure 6.18. In the same time, the voltage amplitude rises from 0 to V_{max} .

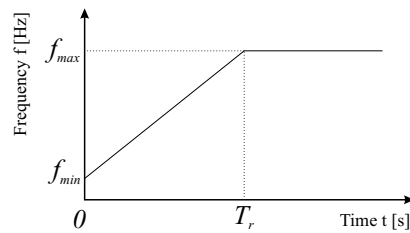


Figure 6.18: U/F control strategy: frequency ramp

The rising time T_r and the maximum voltage amplitude V_{max} can be adjusted by means of five parameters: the maximal and minimal frequencies f_{max} and f_{min} , the absolute maximal frequency f_{abs} , the frequency at which the maximum value is reached f_v and the absolute rising time T_{abs} . The relations relating these parameters are:

$$T_r = \frac{f_{max} - f_{min}}{f_{abs} - f_{min}} T_{abs}$$

$$V_{max} = \frac{f_{max} - f_{min}}{f_v - f_{min}} * 380V$$

As shown in figure 6.19, a DC-motor was coupled to the induction motor in order to emulate an additional load (inertia and friction).

Note: A flexible coupling was introduced between both motors in order to correct possible misalignment of their axes. It was not considered in our model of this bench.

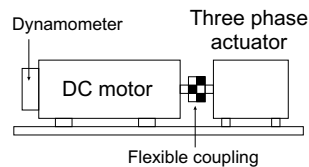


Figure 6.19: Our experimental bench

Several sensors are implemented on this bench. First, the motor is linked to a dynamometer to measure the axle angular velocity. Secondly, two current pinches are used to measure the current through two phases of the motor.

6.3.2 Electrical model

The model of the three phase induction motor is based on a well-known equivalent circuit representation, shown in figure 6.20. In this equivalent circuit a star-star connection is assumed at the stator and at the rotor. For modeling the squirrel cage effect, the rotor phases are short-circuited.

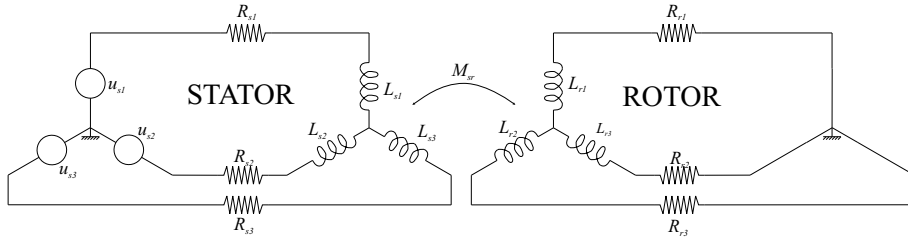


Figure 6.20: Stator and rotor equivalent circuit of a three-phase actuator

Both circuits are characterized by self and mutual induction effects and mutual induction influences take place between the two circuits. A full matrix L of inductances relates the currents and the fluxes through the inductors:

$$\varphi^l = \begin{pmatrix} \varphi_{s1}^l \\ \varphi_{s2}^l \\ \varphi_{s3}^l \\ \varphi_{r1}^l \\ \varphi_{r2}^l \\ \varphi_{r3}^l \end{pmatrix} = L i^l = \begin{pmatrix} L_s & M_s & M_s & & & \\ M_s & L_s & M_s & & & \\ M_s & M_s & L_s & & & \\ & & & L_r & M_r & M_r \\ & M_{sr}^T & & M_r & L_r & M_r \\ & & & M_r & M_r & L_r \end{pmatrix} \begin{pmatrix} i_{s1}^l \\ i_{s2}^l \\ i_{s3}^l \\ i_{r1}^l \\ i_{r2}^l \\ i_{r3}^l \end{pmatrix} \quad (6.7)$$

The mutual induction effects are responsible for the electromechanical interaction taking place in this machine and the mutual inductances M_{sr} between the rotor and the stator is classically assumed to vary sinusoidally with respect to the rotor electrical position $\theta_{em} = p\theta_m$, where $p = 2$ is the number of pairs of poles and θ_m is the rotor angular position:

$$M_{sr} = \begin{pmatrix} M_{sr} \sin(\theta_{em}) & M_{sr} \sin(\theta_{em} + \frac{2\pi}{3}) & M_{sr} \sin(\theta_{em} + \frac{4\pi}{3}) \\ M_{sr} \sin(\theta_{em} + \frac{4\pi}{3}) & M_{sr} \sin(\theta_{em}) & M_{sr} \sin(\theta_{em} + \frac{2\pi}{3}) \\ M_{sr} \sin(\theta_{em} + \frac{2\pi}{3}) & M_{sr} \sin(\theta_{em} + \frac{4\pi}{3}) & M_{sr} \sin(\theta_{em}) \end{pmatrix} \quad (6.8)$$

Let us already point out that in our implementation, the sinusoidal behavior of the motor is not mandatory and matrix M_{sr} might be different from 6.8, depending on the construction of the motor.

Another simplified equivalent circuit for three phase induction electromechanical converters is shown in figure 6.21, where L_μ is called the *magnetization inductance*. It is based on the assumption that the motor is characterized by perfectly sinusoidal mutual inductances M_{sr} , as given in 6.8. In steady-state, the currents and voltages in the different phases are sinusoidal and phasers notation may be used as in figure 6.21, in which \bar{u}_∞ , \bar{i}_s and $\bar{i}'_r = k\bar{i}_r$ respectively represent the network voltage phaser, the statoric current phaser and the rotoric current observed from the stator. The latter is proportional to the rotoric current with gain k determined by the windings at the stator and the rotor. In figure 6.21, $\gamma = \frac{\omega_\infty - \dot{\theta}_{em}}{\omega_\infty}$ denotes the *sliding*, which is the relative difference between the rotation speed of the rotor $\dot{\theta}_{em}$ and the *synchronous speed* ω_∞ . Resistance R_p represents the *magnetic losses*⁶.

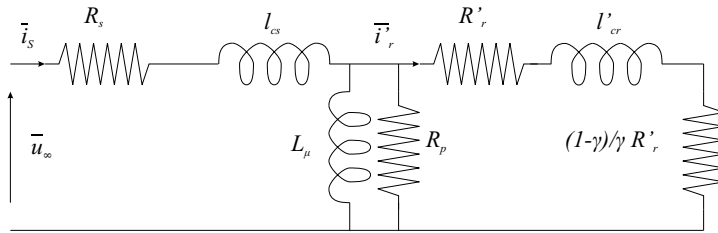


Figure 6.21: Simplified equivalent circuit of the three phase actuator

In the case of squirrel cage motor, it is impossible to make measurements of equivalent rotoric quantities and several common assumptions may be stated:

- It is common to assume the ratio k equal to 1.
- It is experimentally impossible to distinguish between the linkage inductances l_{cs} and l_{cr} and we assume that linkages are equally shared out between the rotor and the stator and $l_{cs} = l_{cr}$.

For classical design of a three phased machine, it is possible to show that [25], if the flux distribution is sinusoidal, $M_s = -\frac{1}{2}L_s$ and $M_r = -\frac{1}{2}L_r$.

The electrical parameters appearing in both equivalent circuits (figure 6.21 and

⁶also referred to as *iron losses*. It englobes losses due to hysteresis in the magnetic materials and Foucault currents.

figure 6.20) are correlated by the following relationship [25]:

$$\begin{aligned}
 l'_{cr} &= l_{cr} \\
 R'_r &= R_r \\
 X_f &= \omega_\infty (l_{cr} + l_{cs}) \\
 L_\mu &= \frac{3}{2} M_{sr} \\
 L_{cr} &= L_\mu + l_{cr} \\
 L_{cs} &= L_\mu + l_{cs} \\
 L_{cr} &= L_r - M_r = \frac{3}{2} L_r \\
 L_{cs} &= L_s - M_s = \frac{3}{2} L_s
 \end{aligned} \tag{6.9}$$

in which ω_∞ represents the network phase equal to $\omega_\infty = 2\pi f_\infty$. All these relations make it possible to pass from the equivalent circuit of figure 6.20 to the simplified phaser equivalent circuit of figure 6.21, and vice-versa.

Comparing the equivalent circuits of figures 6.20 and 6.21, we observe that the second one does not involve constraints, while the first one introduces 2 constraints between the state variables $w = (i_{s1}^l \ i_{s2}^l \ i_{s3}^l \ i_{r1}^l \ i_{r2}^l \ i_{r3}^l)^T$. For this reason, most of the tools simulating electromechanical three phase converters use the phaser representation of figure 6.21, but are restricted to motors with sinusoidal flux distribution. Other approaches for modeling three phase machines without constraints use the Concordia and Park transformations [25]. As for the phasers representation, these are based on the assumption that the device has an ideal sinusoidal behavior, what is not always the case in today's designs. The Power System Blockset developed by Mathworks for Simulink is based on these last transformations.

In our modeling approach, thanks to our constraints reduction procedure, we are not bothered by the star-star connection creating these constraints, and we can start from the equivalent circuit of figure 6.20. This means that we are not forced to assume a sinusoidal behavior of the machine to simplify or reduce the equivalent circuit.

Note: In this validation, in order to compare our results with the Power System Blockset from Mathworks, we assumed a sinusoidal distribution of the flux.

The tree graph given as input for Electran is represented in figure 6.22.

As pointed out before, because of the star-star connection, there exist two constraints between the state variables:

$$i_{s1}^l + i_{s2}^l + i_{s3}^l = 0 \tag{6.10}$$

$$i_{r1}^l + i_{r2}^l + i_{r3}^l = 0 \tag{6.11}$$

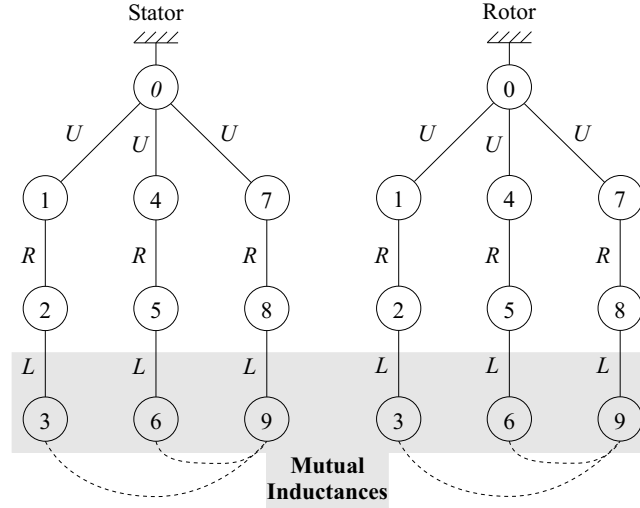


Figure 6.22: Electran's tree graph representation of the three phase motor

The independent state variables w_u automatically selected by Electran are:

$$w_u = \begin{pmatrix} i_{s2}^l \\ i_{s3}^l \\ i_{r2}^l \\ i_{r3}^l \end{pmatrix} \quad (6.12)$$

6.3.3 Parameter identification

In order to compare the simulation results with the experimental measurements, it is necessary to identify all the parameters of the model, especially the motor parameters: R_s , R_r , L_s , L_r , M_s , M_r and M_{sr} .

Classical identification procedures for three phase induction motors give estimation of the parameters for the phasers equivalent circuit of figure 6.21: L_μ , R_p , L_{cr} , L_{cs} , R'_r and R_s . Therefore, two experiments are conducted [66] after measuring the statoric resistance $R_s = 68.8 \Omega$ directly between the connections:

- the open shaft experiment ($\gamma \simeq 0$) to identify L_μ and R_p
- the locked rotor experiment ($\gamma = 1$) to measure R'_r , l_{cs} , and l_{cr} .

The identified values for the phaser equivalent circuit are:

$$\begin{aligned} L_\mu &= 3.0636 & (H) \\ R_p &= 1.194 \cdot 10^4 & (\Omega) \\ R'_r &= 8.18 & (\Omega) \\ l_{cs} &= 0.1229 & (H) \\ l_{cr} &= 0.1229 & (H) \end{aligned} \quad (6.13)$$

Note: More details about this classical identification method [66] can be found in appendix D.

Replacing the identified values 6.13 in relations 6.9, we obtain an estimation of the parameters for our model, see figure 6.20:

$$\begin{aligned}
 R_s &= 68.8 & (\Omega) \\
 R_r &= 8.18 & (\Omega) \\
 L_s &= 2.204 & (H) \\
 L_r &= 2.204 & (H) \\
 M_s &= -1.102 & (H) \\
 M_r &= -1.102 & (H) \\
 M_{sr} &= 2.04 & (H)
 \end{aligned} \tag{6.14}$$

The mechanical load of the motor simply consists in a constant inertia and friction. The equivalent inertia of the rotating pieces was identified: $I = 0.0012 \text{ Kg m}^2$. Friction has been introduced in the model by means of an equivalent viscous torque. The equivalent viscous friction coefficient $d = 0.002 \text{ N s/m}$ was identified on the basis of a first set of measurements as explained in the following section.

6.3.4 Comparison between simulation and experimental results

Validation of the identified values The identification procedure described above led us to estimated values of the parameters appearing in our model. In order to validate these values, we compared our simulation results and experimental measurements with the following settings for the unit control:

$$\begin{aligned}
 T_{abs} &= 1 \text{ s} \\
 f_{abs} &= 120 \text{ Hz} \\
 f_v &= 50 \text{ Hz} \\
 f_{min} &= 0 \text{ Hz} \\
 f_{max} &= 50 \text{ Hz}
 \end{aligned}$$

With these settings, the frequency linearly rises from 0 Hz to 50 Hz over $\frac{50}{120} \text{ s}$. Over the same time, the source amplitude rises from 0 V to 380 V .

Comparing the measurements with the simulation results, we could adjust the electrical parameters and the viscous friction coefficient d for the model. The latter was determined by comparing the steady-state currents magnitudes, directly related to the steady-state torque, which is equal to the friction torque.

The final set of value is given below:

$$\begin{aligned}
 R_s &= 68.8 & (\Omega) \\
 R_r &= 20.45 & (\Omega) \\
 L_s &= 2.414 & (H) \\
 L_r &= 2.414 & (H) \\
 M_s &= -1.162 & (H) \\
 M_r &= -1.162 & (H) \\
 M_{sr} &= 2.05 & (H) \\
 d &= 0.002 & (Ns/m) \\
 I &= 0.0012 & (Kg\,m^2)
 \end{aligned}
 \tag{6.15}$$

Except for the rotoric resistance R_r , the electrical parameters were slightly changed (less than 10% of variation with respect to the identified values 6.14). Since we assumed that $k = 1$, the rotoric resistance R_r is equal to R'_r . As explained in Appendix D, the latter is estimated as the difference between two measured entities, which are quite larger than R'_r . Its estimation is consequently far from being precise and large changes are not surprising.

Figure 6.23.a shows the comparison between the measured currents (continuous curve) and the simulated ones (dotted curve). As one can see, the curves are very similar. The simulated curve indicates a higher level of current before the measurements. This can be explained by a non-ideal frequency ramp at the output of the unit control. This is confirmed by figure 6.23.b which shows the rotor velocity. One can see that the measured velocity does not rise as fast as the simulated one. It was not possible to directly measure the output voltage of the control unit (it is a pulse width modulated signal with average value equal to the output voltage), but it seems that the actual frequency ramp starts more smoothly than that of the model.

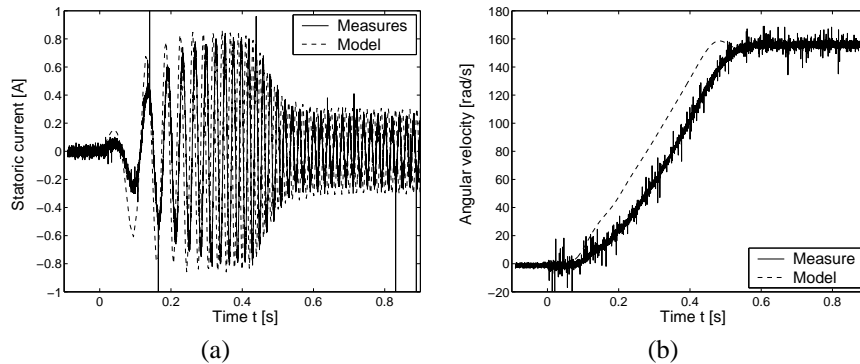


Figure 6.23: Experimental comparison with $f_{max} = 50Hz$ and $T_{obs} = 1s$

The same motor was also simulated using the Power System Blockset from Mathworks, Inc., and the corresponding simulation results are almost identical to those obtained with Electran. This is shown in figure 6.24 that compares the currents and velocities from both tools.

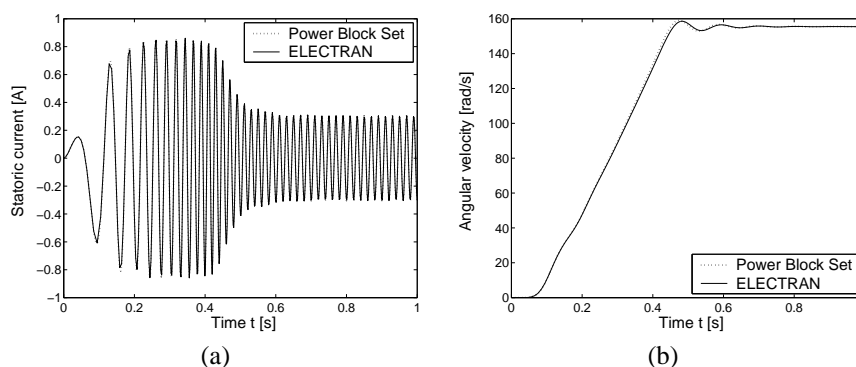


Figure 6.24: Comparison between Electran and Power System Blockset ($f_{max} = 50Hz$ and $T_{abs} = 1s$)

Comparison between the model and the measurements Now that our parameters have been adjusted (see values 6.15), we can compare simulation results and experimental measurements for different starting phase with different settings of the control unit. Figures 6.25, 6.26 and 6.27 show a good concordance between the model and the measurements.

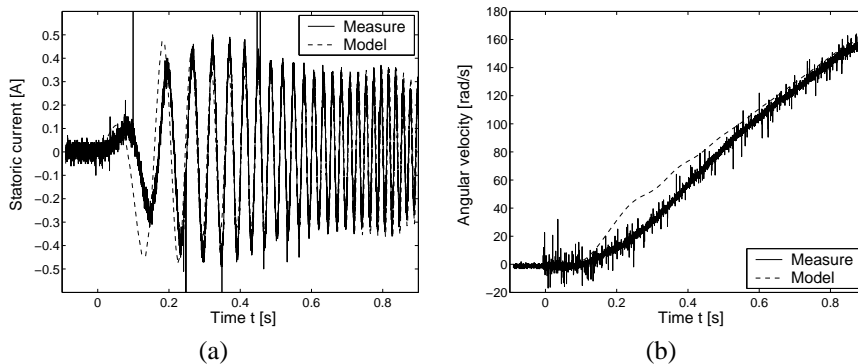
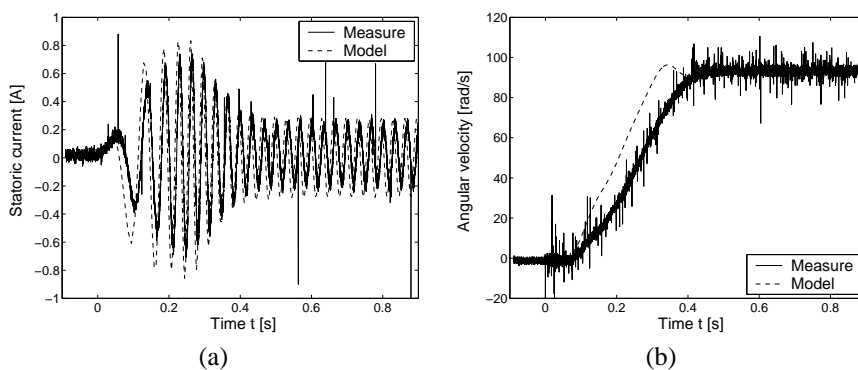
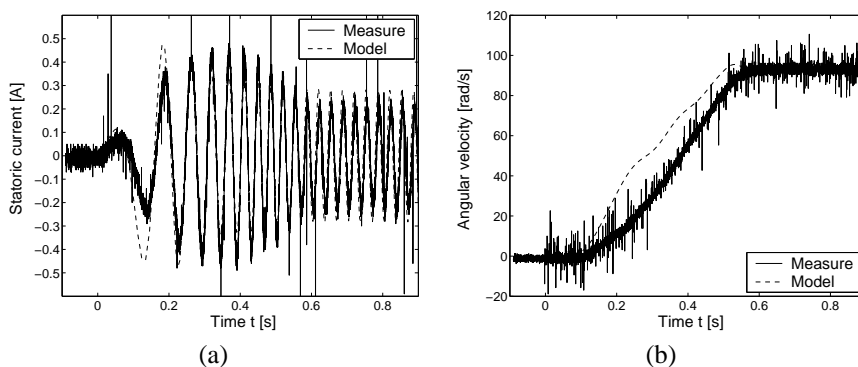
Figure 6.28 indicates conditions for which the simulation results and experimental measurements do not correspond. Nevertheless, when using the Power System Blockset, simulation results are not better, as shown in figures 6.29 and 6.30. The latter shows the difference between the statoric currents obtained with both tools, which remains below the numerical integrator precision (10^{-7}).

The differences appearing with the measurements may be due to several non-ideal effects not taken into account in the model:

- non-sinusoidal flux distribution
- saturation of the magnetic material
- magnetic losses associated with hysteresis phenomenon and Foucault current
- skin-effect

Magnetic losses and skin-effect are usually negligible and have to be considered only in very specific cases. We doubt that saturation is the explanation of the observed differences since the starting phase shown in figures 6.28 and 6.29 is smoother than the previous ones. A non-sinusoidal flux distribution might explain the differences. Indeed, differences are observed for a much slower rotation, at which the flux distribution has more influence.

Assuming that a non-sinusoidal behavior has to be modeled, tools such as the Power System Blockset, become less suitable, because based on the assumption of sinusoidal flux distribution, while our approach allows for considering any flux distribution.

Figure 6.25: Comparison with $f_{max} = 50Hz$ and $T_{abs} = 2s$ Figure 6.26: Comparison with $f_{max} = 30Hz$ and $T_{abs} = 1s$ Figure 6.27: Comparison with $f_{max} = 30Hz$ and $T_{abs} = 2s$

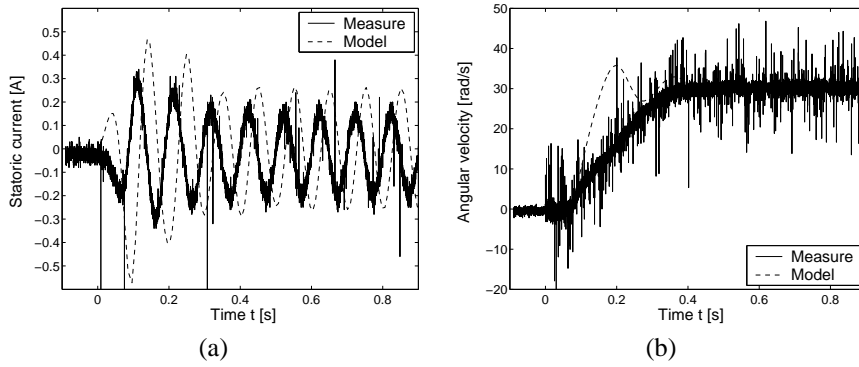


Figure 6.28: Comparison with $f_{max} = 10Hz$ and $T_{abs} = 1s$

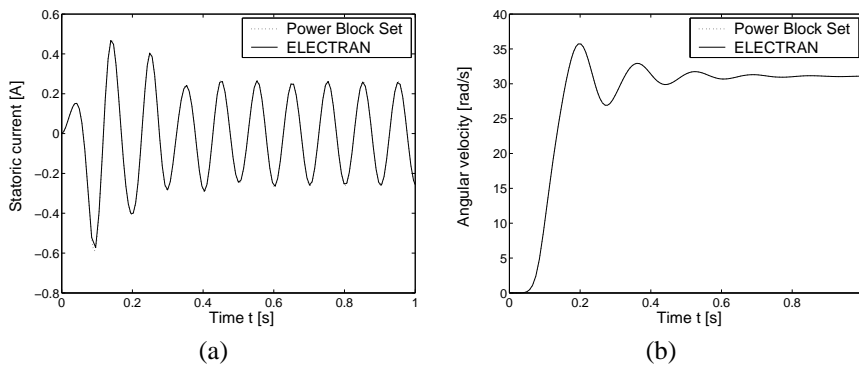


Figure 6.29: Comparison between Electran and Power System Blockset ($f_{max} = 10Hz$ and $T_{abs} = 1s$)

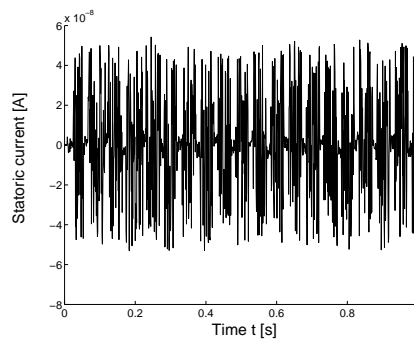


Figure 6.30: Difference between Electran and Power System Blockset ($f_{max} = 10Hz$ and $T_{abs} = 1s$)

6.4 Permanent magnets applications

As explained in section 5.4.4, when permanent magnets are involved in electromechanical converters, they induce a position dependent flux through the windings of the coils involved in the electrical subsystem.

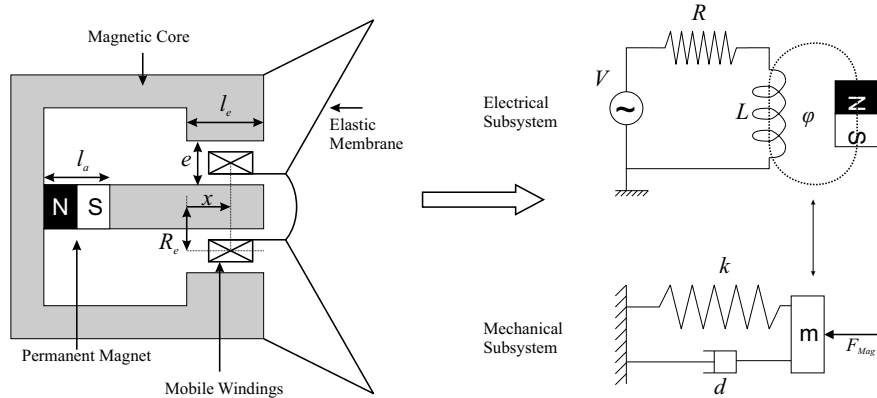


Figure 6.31: Electromagnetic speaker and the corresponding subsystem models

6.4.1 Example 1: Electromagnetic speaker

As a first example, let us consider the electromagnetic speaker⁷ shown in figure 6.31. A coil is attached to an elastic membrane and is allowed to slide along a magnetic core. The flux induced by the permanent magnet will vary with the position x of the coil along the core.

The electrical subsystem consists of an equivalent circuit of the windings, with a resistance R serially connected to an inductance L , coupled with the permanent magnet. The inductance L and the flux φ due to the magnet through the windings vary with the relative position x as follows:

$$\begin{aligned} L &= \alpha (\beta + \gamma x) \\ \varphi(x) &= \phi_0 (l_e - x) \end{aligned} \quad (6.16)$$

where l_e is defined in figure 6.31 as the axial length of the air-gap, and α , β , γ and ϕ_0 are constants taking into account the material and geometrical properties of the speaker. Detailed expressions are given in Appendix E, where the input file to Electran and the generated equations are also given.

The mechanical subsystem consists of a simple damped spring-mass system with damping coming from viscous friction. The mass is equal to the total mass of the mobile parts while the stiffness is the equivalent stiffness of the elastic membrane.

⁷http://www.lei.ucl.ac.be/multimedia/Convertisseurs/chapitre_2/labos/HautParleur/enonce.htm

More complex mechanical model could be considered but our goal is to show that we are able to model permanent magnets and not to precisely model the behavior of the membrane.

The spring-mass system is excited by an electromagnetic force F_{mag} whose expression can be found as (see equation 3.174):

$$F_{mag} = \frac{\partial W_m^*}{\partial x} = \frac{1}{2} \frac{\partial L(x)}{\partial x} (i_L)^2 + \frac{\partial \varphi(x)}{\partial x} i_L \quad (6.17)$$

The numerical parameters for this application can be found in Appendix E.1.

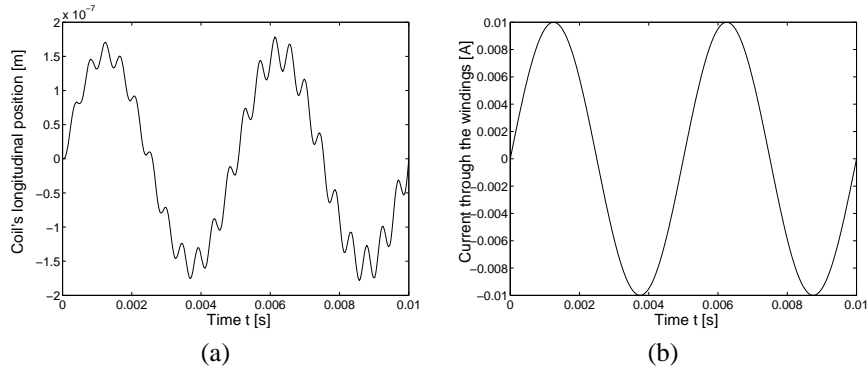


Figure 6.32: Electromagnetic speaker - Simulation results with Electran

Figure 6.32.a shows the position of the coil when connected to a 200 Hz voltage source. As one can see, the membrane vibrates at 200 Hz , with an additional oscillation at a frequency corresponding to $\frac{1}{2\pi} \sqrt{\frac{k}{m}} \cong 2.25 \cdot 10^3 \text{ Hz}$. Figure 6.32.b shows the current flowing through the windings under the same conditions. The mechanical oscillation at $2.25 \cdot 10^3 \text{ Hz}$ is not transmitted to the electrical circuit for two main reasons:

1. the electrical resistor has an impedance which is much higher than the one of the inductor;
2. the mechanical oscillation has a very small magnitude and does not produce significant flux variations.

6.4.2 Example 2: Three-phase step motor

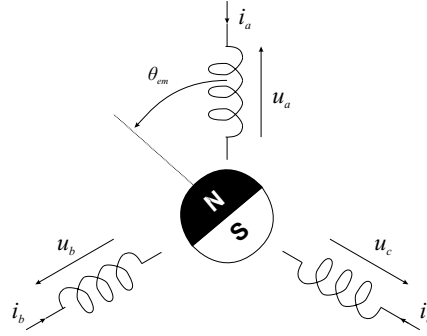


Figure 6.33: Schematic of the permanent magnet synchronous motor

Figure 6.33 shows a schematic representation of the considered motor. The stator is composed of windings identical to those of the three-phase induction motor studied before, and the rotor simply consists of a permanent magnet. The equivalent circuit is simply obtained by replacing the rotor in figure 6.20 by a permanent magnet.

The working principle of this kind of machine is based on the fact that the magnet prefers to be in line with the active windings. Feeding sequentially the different windings forces the magnet to move from one position to the other.

The fluxes φ flowing through the windings is given by:

$$\varphi = \begin{pmatrix} \varphi_a \\ \varphi_b \\ \varphi_c \end{pmatrix} = \varphi^0 + L i^l \quad (6.18)$$

where φ^0 represents the flux due to the permanent magnet, L is the self and mutual inductance matrix and $i^l = \begin{pmatrix} i_a^l \\ i_b^l \\ i_c^l \end{pmatrix}$ is the array of currents through the inductors.

From [25], we get the expression of φ^0 and L as function of the electrical angular position $\theta_{em} = p \theta_m$ where p is the number of pairs of poles and θ_m is the angular position of the rotor,

$$\varphi^0 = \phi^0 \begin{pmatrix} \cos(\theta_{em}) \\ \cos\left(\theta_{em} - \frac{2\pi}{3}\right) \\ \cos\left(\theta_{em} - \frac{4\pi}{3}\right) \end{pmatrix} \quad (6.19)$$

$$L = \begin{pmatrix} L_0 & M_0 & M_0 \\ M_0 & L_0 & M_0 \\ M_0 & M_0 & L_0 \end{pmatrix} + L_2 \begin{pmatrix} \cos(2\theta_{em}) & \cos(2\theta_{em} - \frac{2\pi}{3}) & \cos(2\theta_{em} - \frac{4\pi}{3}) \\ \cos(2\theta_{em} - \frac{2\pi}{3}) & \cos(2\theta_{em} - \frac{4\pi}{3}) & \cos(2\theta_{em}) \\ \cos(2\theta_{em} - \frac{4\pi}{3}) & \cos(2\theta_{em}) & \cos(2\theta_{em} - \frac{2\pi}{3}) \end{pmatrix}$$

These expressions of L and φ^0 have to be introduced by the user within the user functions generated by Electran.

One possible control for such an actuator imposes the three phases to be active after each other. This control is shown in figure 6.34.a and the resulting rotor position is shown in figure 6.34.b. The mechanical load is simply a constant inertia with viscous friction.

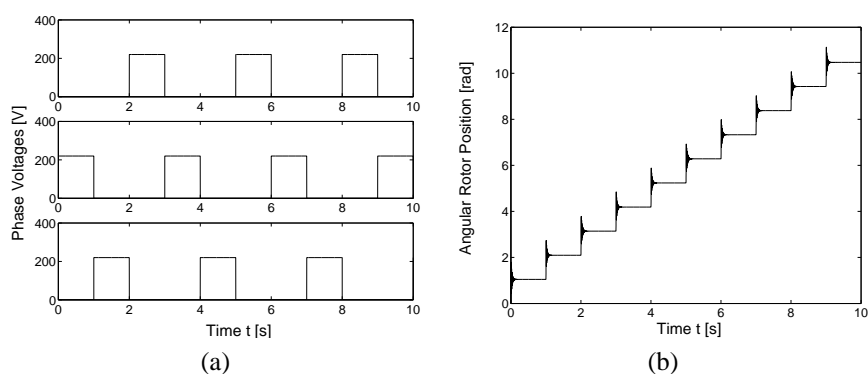


Figure 6.34: Synchronous permanent magnet motor - Simulation results with Electran

Industrial applications

7.1 Parking Gate System

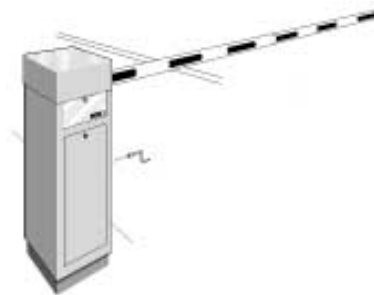


Figure 7.1: Picture of the mechanism driving the barrier

As shown on figure 7.1, we propose to model and simulate the behavior of a parking gate system [58].

During a visit of the installations of Automatic Systems S.A, a company that, among other things, build parking gates, it appeared that the design simply consists in finding the best combination of motor and gearbox for a given barrier, depending on the length and material of the barrier but also on the on-site conditions, notably the

opening-closing times. A trial-error process is performed until compromise is reached between the vibrations inside the barrier and the opening and closing times. Replacing this process by a faster model based design was our first motivation.

This study was also interesting for the tight electromechanical interaction taking place. Indeed, the flexibility of the barrier is not negligible because of its lanky shape and one of its natural frequencies is close to the frequency of the electrical network to which the motor is plugged in.

7.1.1 Description of the system

As shown in figure 7.2, the parking gate involves a barrier, consisting of a flexible beam, actuated by an asynchronous three-phase induction actuator through a six-bar mechanism and a gearbox. A spring is used to counteract the weight of the barrier.

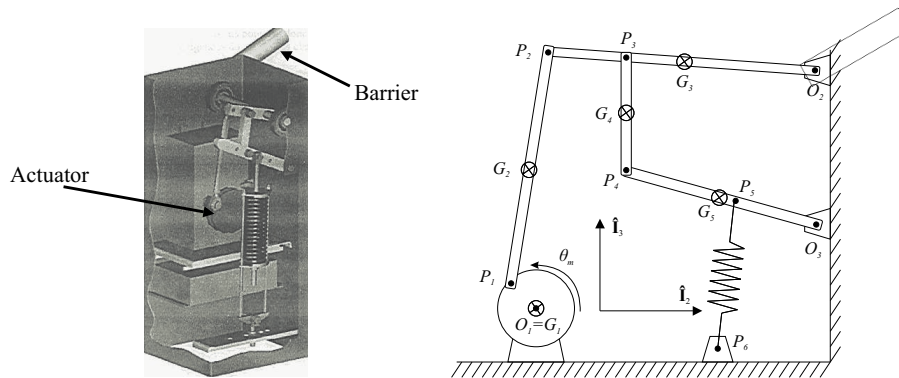


Figure 7.2: Components and structure of the parking gate system

The barrier is a 3m long aluminum beam with the cross section shown in figure 7.3.

The dynamic properties of the barrier are:

$$\begin{aligned}
 \text{Young modulus: } E &= 72 \cdot 10^9 \text{ N/m}^2; \\
 \text{Density: } \rho &= 2700 \text{ kg/m}^3; \\
 \text{Cross sectional moment of inertia: } I_{yy} &= 0.12315 \cdot 10^{-6} \text{ m}^4; \\
 \text{Mass: } m &= 2.682 \text{ kg};
 \end{aligned} \tag{7.1}$$

The spring is characterized by a stiffness $K = 13000 \text{ N/m}$ and a natural length $l_0 = 0.39 \text{ m}$.

The six-bar mechanism was designed in such a way that the barrier is horizontal when point P_1 is at the upper position and vertical when P_1 is at the lower position (see figure 7.2). By this way, one turn of the gearbox output rotor axle corresponds to one opening-closing cycle.

The behavior of the barrier will be analyzed for different gearbox ratios: $\frac{1}{53}$, $\frac{1}{16}$ and $\frac{1}{1}$.

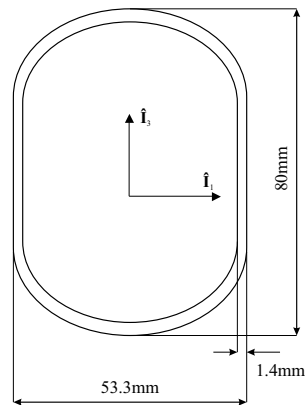


Figure 7.3: Cross section of the barrier

The actuator is a three phase induction motor similar to the one used in the previous experimental bench of section 6.3. A star-star configuration, as shown in figure 6.20, is assumed with a short-circuited rotor, and the matrices of inductance are identical to those given in 6.7 and 6.8.

7.1.2 Parking Gate Model

7.1.2.1 Flexible barrier model

According to the finite segment approach proposed by Huston [27], the beam is modeled as a series of identical segments connected together by equivalent revolute springs, as illustrated in figure 7.4. In [27], Huston proposes the following formula to calculate the equivalent stiffness K_{eq} of a beam subjected to bending:

$$K_{eq} = \frac{E \cdot I_{yy}}{l} \quad (7.2)$$

where l is the length of the segments.

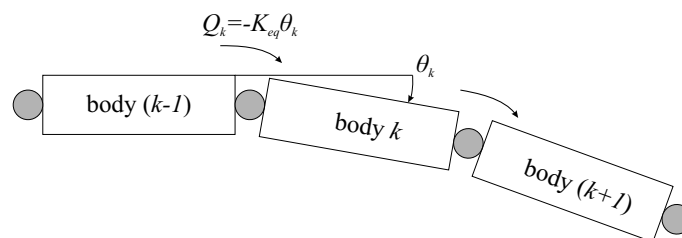


Figure 7.4: Finite segment model of a flexible beam

This lumped approach allows to use rigid multibody formalisms to deal with flexible beam, with a quite good accuracy in case of simple deformations. More advanced techniques could have been used (e.g. Finite element method or global shape function techniques) but this is out of the scope of the present work. However, in this application, only the first modes of vibration are of importance.

With the finite segment technique, the shorter the segments, the better the results, but also the larger the model. A good balance has to be reached between the required precision and the size of the model. In our case, 30 segments of 0.1 m were used. This was sufficient to reasonably approximate at least the two first eigenmodes of vibration of the beam, as shown in table 7.1.

Modes	Theoretical	Approximated
1	8.53 Hz	8.83 Hz
2	53.47 Hz	50.41 Hz
3	149.73 Hz	125.18 Hz

Table 7.1: Theoretical and approximated eigenfrequencies of the flexible barrier

The complete multibody system is represented in figure 7.5, in which bodies are represented by grey rectangles, while the lines represent the joints. In this planar system, all the joints, among which joint 1 is actuated, are revolute around the $\hat{\mathbf{I}}_1$ -axis, what is indicated by the $R1$ symbol. Two loops are present, and were opened¹ by means of two spherical cuts, denoted $C2$ [56]. A spherical cut corresponds in suppressing a revolute joint and imposing to the two corresponding attach points to have the same position all the time.

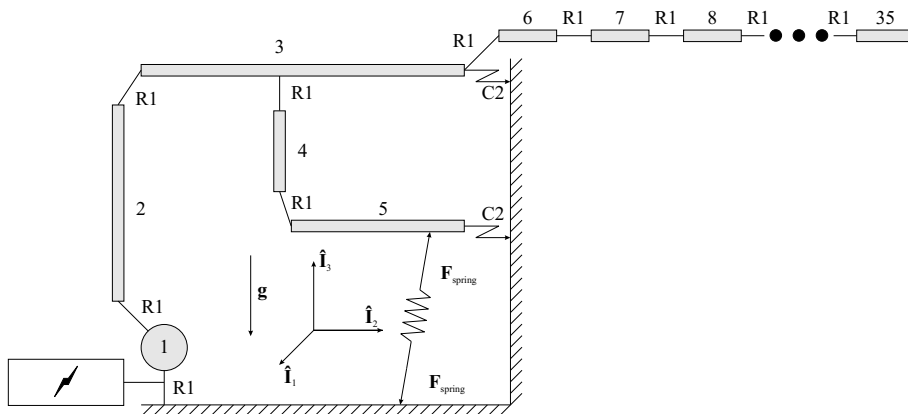


Figure 7.5: Multibody schematic of the flexible barrier and its mechanism

The spring has been modeled as a *link*. This concept was introduced in section 5.3.1 and can be used when the dynamic properties (mass, inertia,...) of a mechanical

¹closed-loop MBS are first transformed into tree-like structures by cutting the loops.

component can be neglected, while its effect cannot: typically, a spring has very low mass and inertia but produces important forces on the bodies to which it is attached.

Joint 6 connecting the first segment of the barrier with the six-bar mechanism is a *locked joint*², which maintains the angle between the barrier and the mechanism at a constant value such that when P_1 is at the upper position ($\theta_m = 0$), the barrier is horizontal.

7.1.2.2 The electrical actuator

The actuator model is the same as for the experimental bench except for the parameters, which were deduced from the datasheets:

$$\begin{aligned}
 p &= 3 \\
 R_s &= 27.35 \quad (\Omega) \\
 R_r &= 2.735 \quad (\Omega) \\
 L_s &= 1.058 \quad (H) \\
 L_r &= 1.058 \quad (H) \\
 M_s &= -0.525 \quad (H) \\
 M_r &= -0.525 \quad (H) \\
 M_{sr} &= 0.987 \quad (H)
 \end{aligned} \tag{7.3}$$

As a reminder, figure 7.6 shows the equivalent circuit.

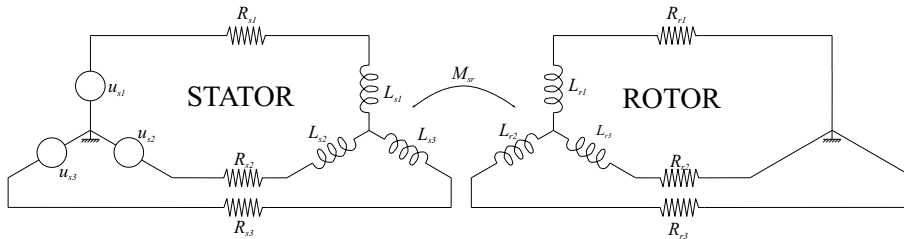


Figure 7.6: Stator and rotor equivalent circuit of a three-phase actuator

7.1.2.3 Interfacing of the models

As mentioned in section 4.3.2, when constraints exist in the electrical and mechanical subsystems, the evaluation of the models, at the same instant of time t , has to follow a certain sequence, according to the flowchart of figure 4.13. Knowing the mechanical independent positions s_u^m and velocities \dot{s}_u^m and the independent electrical state variables w_u ,

- first, the mechanical constraints have to be solved in order to obtain the positions and velocities of all the joints, s^m and \dot{s}^m respectively;

²defined in section 5.3.1

- second, the electrical model is then evaluated, resulting in the derivative of the independent electrical state variables \dot{w}_u and the electromechanical force Q^{em} ;
- third, the independent accelerations \ddot{s}_u^m can be computed

7.1.3 Simulation results

In our simulations, no specific control is used for the barrier. The motor is directly connected to the usual european three phase network (380 V, 50 Hz) and switched on instantaneously. According to the star-star connections, this corresponds to the following values for our statoric input voltages:

$$\begin{aligned} u_{s1} &= \sqrt{2} 220 \sin(2\pi 50t) \\ u_{s2} &= \sqrt{2} 220 \sin\left(2\pi 50t + \frac{2\pi}{3}\right) \\ u_{s3} &= \sqrt{2} 220 \sin\left(2\pi 50t + \frac{4\pi}{3}\right) \end{aligned}$$

The initial conditions for the system are:

- barrier in the horizontal position (P_1 at the upper position): $\theta_m = s_1^m = 0$ (see figure 7.2)
- zero voltage and current in the motor: $w_{u0} = (0 \ 0 \ 0 \ 0)^T$

We simulated the behavior of the barrier for three different gearbox ratios: $G_r = \frac{1}{53}$, $\frac{1}{16}$ and $\frac{1}{1}$, the first ratio $\frac{1}{53}$ being the value used on the actual system for the considered barrier.

7.1.3.1 Rigid or flexible model

We will begin the analysis of the simulation results by comparing the results obtained with a $3m$ rigid barrier and a $3m$ flexible barrier.

Figure 7.7, 7.8 and 7.9 show the results obtained with gearbox ratio $\frac{1}{53}$, $\frac{1}{16}$ and $\frac{1}{1}$ respectively. The different plots show the angular position of the gearbox output axle (a), the electromechanical torque (b), the statoric current (c) and the rotor current (d).

As can be seen on these figures, the higher the gearbox reduction, the less the influence of the flexibility. This can be understood by the fact that the barrier opens at different speeds and deforms accordingly:

- In figure 7.7, the highest ratio of $\frac{1}{53}$ was used³. The barrier opens slowly and deformations due to flexibility are insignificant. Flexible and rigid barrier have thus the same behavior. The barrier is completely opened after approximately 1.5s, corresponding to a half turn of the gearbox output axle, the angular position of which is plotted in figure 7.7.a.

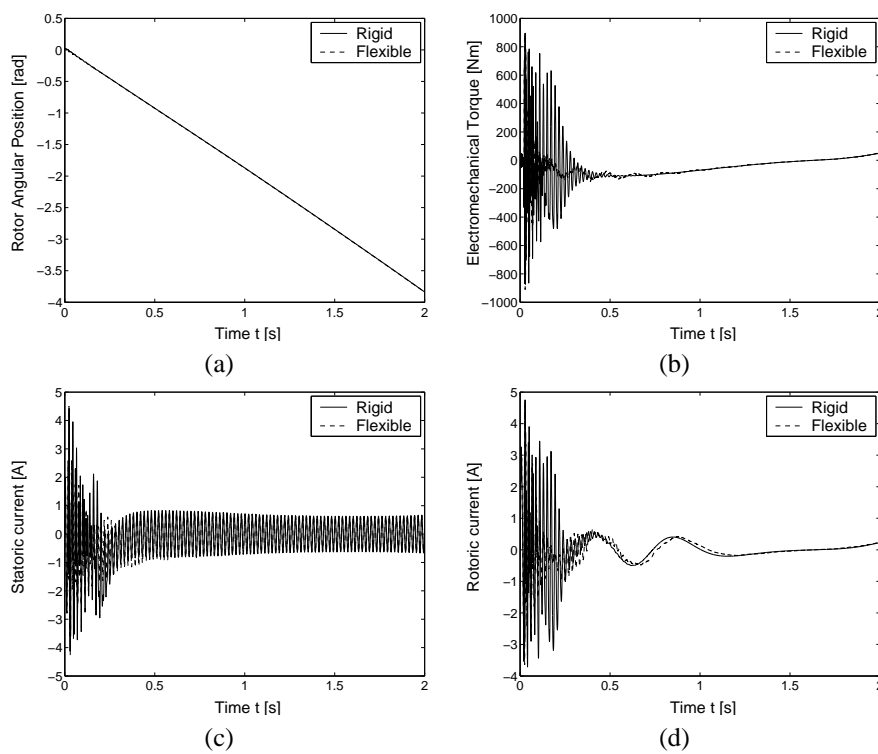


Figure 7.7: Comparison between the flexible and rigid models with actuator and $\frac{1}{53}$ gearbox ratio.

³This value is used on the actual barrier.

- For a $\frac{1}{16}$ gearbox ratio, the accelerations transmitted to the barrier are higher and the barrier deforms more, what influences the behavior of the whole system. This can be observed in figure 7.8 indicating important differences in the torque (plot b) and currents (plots c and d) of the motor when coupled to rigid or flexible barrier.

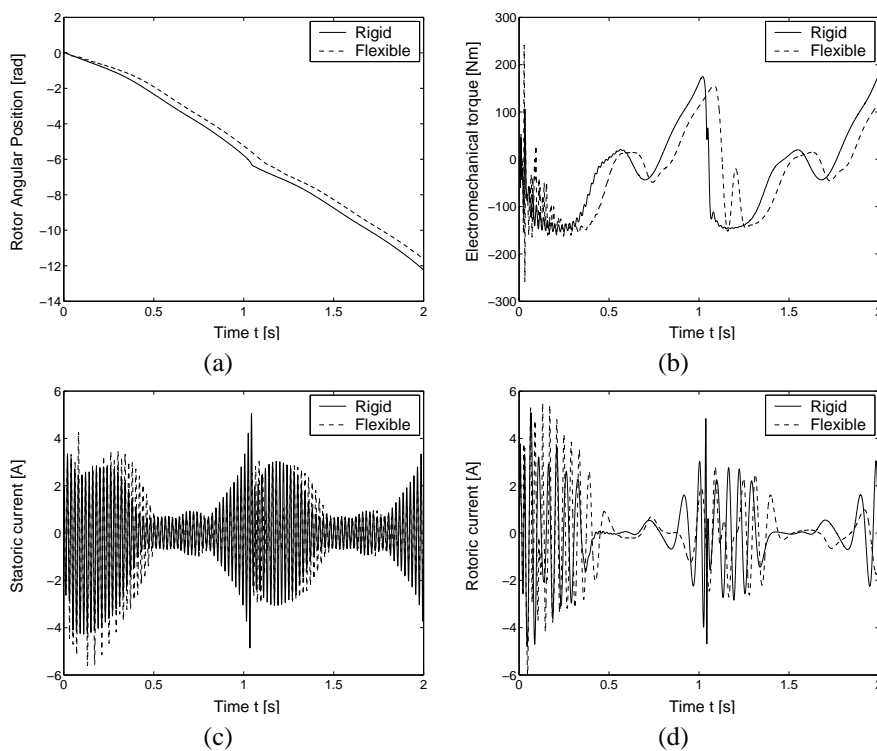


Figure 7.8: Comparison between the flexible and rigid models with actuator and $\frac{1}{16}$ gearbox ratio.

In figure 7.8.a, the kink appearing after 1 s corresponds to the instant when the barrier has achieved a full opening-closing cycle (one turn of the gearbox output axle) and starts opening again. Because of inertial effects, this requires more torque (figure 7.8.b) and the statoric and rotoric currents are higher, as shown in figure 7.8.c and d.

- When the gearbox ratio is too low, e.g. $\frac{1}{1}$, the motor is not strong enough to rise the barrier that oscillates, but never gets open. This generates low frequency deformations inside the barrier, which does not influence the electrical subsystem, but the behavior of the flexible barrier is completely different from that of the rigid one, as can be seen in figure 7.9.

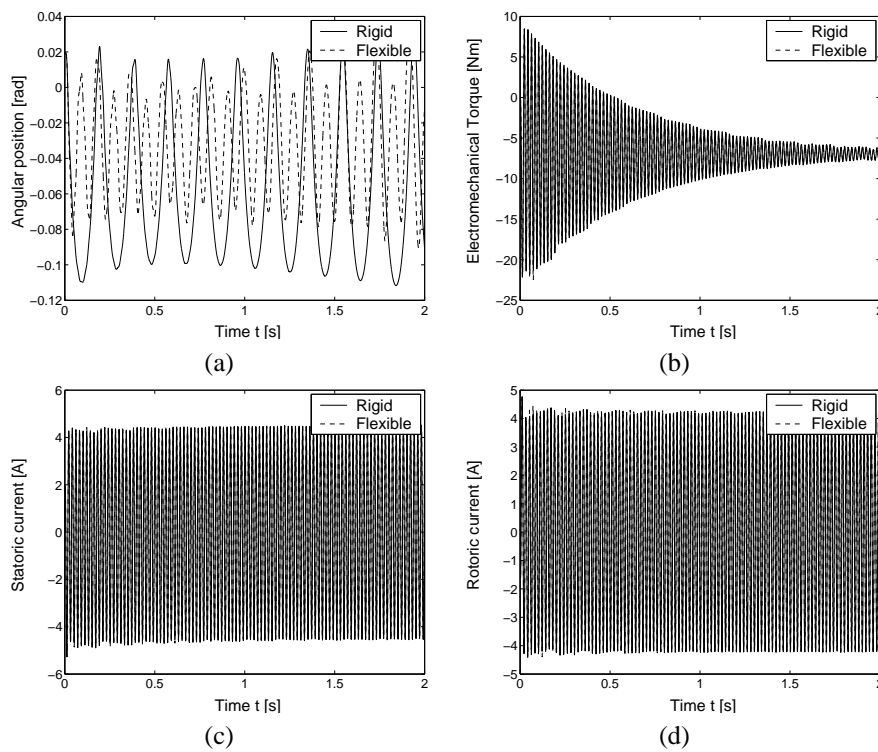


Figure 7.9: Comparison between the flexible and rigid models with actuator and $\frac{1}{1}$ gearbox ratio.

7.1.3.2 Electromechanical interaction analysis

We already pointed out some interesting conclusion concerning the electromechanical interaction, especially about the influence of the barrier flexibility and the corresponding flexible model.

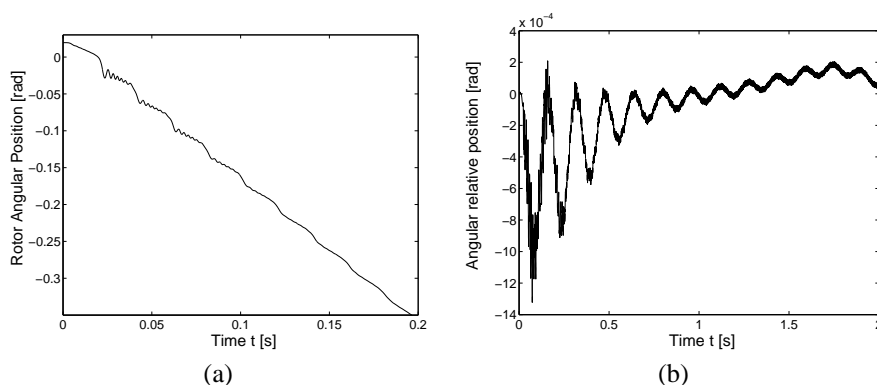


Figure 7.10: Mechanical damping through electrical resistors

Another important phenomenon confirming the actual interaction existing between the mechanical and electrical subsystems, has to be mentioned. Figure 7.10.a shows a detail (from $t = 0\text{ s}$ to $t = 0.2\text{ s}$) of the angular position of the gearbox output axle for a $\frac{1}{53}$ gear ratio. As one can see, oscillations due to the vibrations in the barrier are damped and disappear. In our flexible model, only equivalent stiffness was introduced but no damping, meaning that the mechanical model cannot dissipate energy. The electrical resistors are the only dissipative element in the electromechanical model. We can thus conclude that the mechanical oscillations and vibrations are damped by dissipation of energy through the electrical resistors. This is confirmed by figure 7.10.b showing the damped vibrations between two segments, what illustrates the tight electromechanical interaction that exists in this case.

Note: This damping was observed for different precision settings of the ODE integrator and also for different integrators. This aims at concluding that it does not correspond to an artificial numerical damping.

7.1.3.3 Comparison with a linear graph model

With the help of Chad Schmitke and Pr. John McPhee (University of Waterloo, Ontario, Canada), we compared our results with a linear graph model of the barrier⁴. This comparison is presented in [60], a copy of which is given in appendix A.

⁴The linear graph theory has been presented in chapter 3

Using Dynaflex⁵, they simulated the same parking gate system, whose flexible model is based on a shape function approach to approximate the beam deformation. Details about their models can be found in [60] (see appendix A) and the obtained linear graph is shown in figure 7.11.

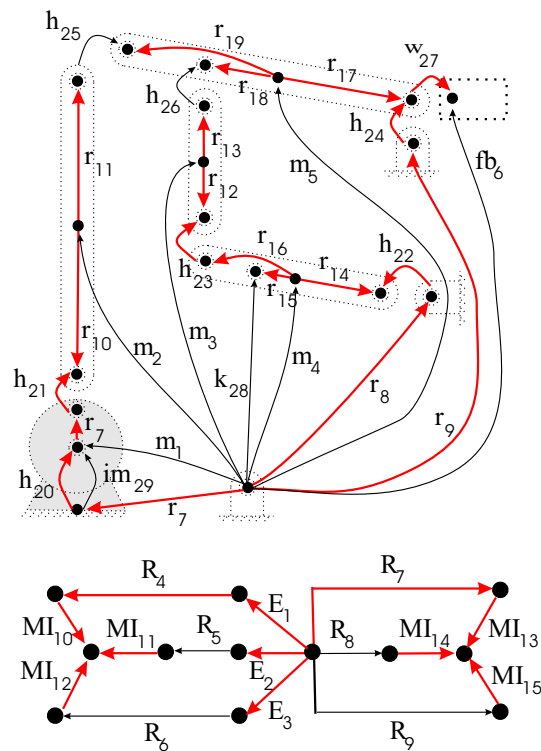


Figure 7.11: Linear graph model of the flexible barrier and its actuator

The results are compared in figures 7.12, 7.13 and 7.14.

As said before, flexibility has more influence when a lower gearbox ratio is used. Consequently, with a gear ratio of $\frac{1}{53}$, as shown in figure 7.12, both models give almost identical results, since the deformations of the barrier are insignificant. When lower gearbox ratios are used, figures 7.13 and 7.14, differences appear between our model and Dynaflex's model, mainly due to the use of different approaches to model the flexibility of the barrier.

⁵Dynaflex is a symbolic model generator for multibody systems, recently extended to electromechanical system, based on linear graph theory and developed at the University of Waterloo, Ontario, Canada

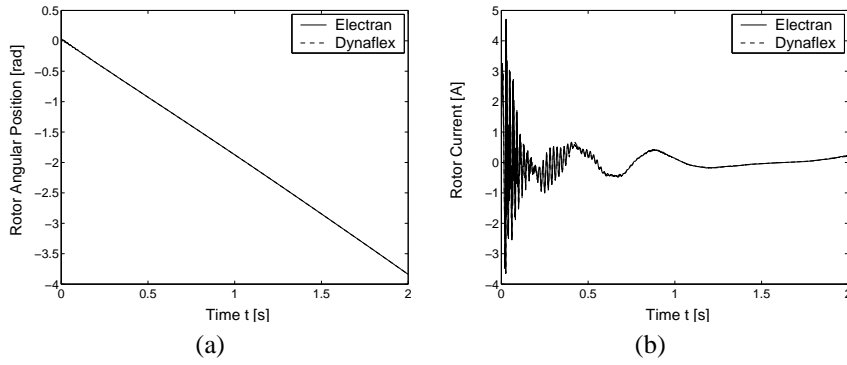


Figure 7.12: Simulation results comparison between Electran and Dynaflex models with gearbox ratio of $\frac{1}{53}$

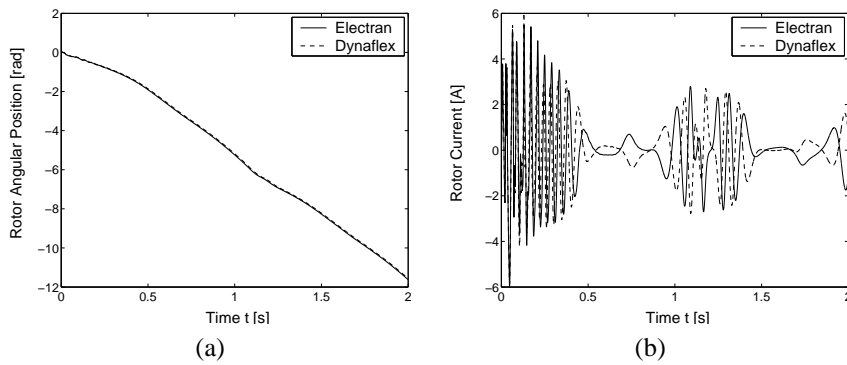


Figure 7.13: Simulation results comparison between Electran and Dynaflex models with gearbox ratio of $\frac{1}{16}$

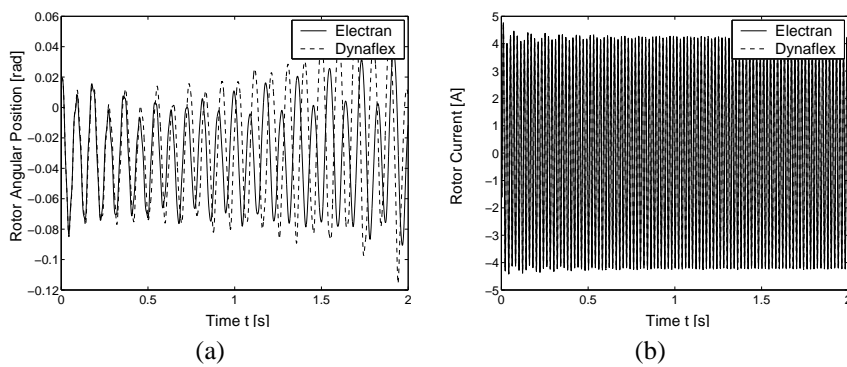


Figure 7.14: Simulation results comparison between Electran and Dynaflex models with gearbox ratio of $\frac{1}{1}$

7.2 Railway Bogie Driven by Inductive Motors

7.2.1 Description of the system

The application considered here is a subway bogie designed by Bombardier Transport for the city of Caracas (Venezuela).

It consists of a chassis, two axles and 4 wheels, as illustrated in Figure 7.15. Unlike a classical bogie, this one has an *articulated chassis* separated into two longitudinal parts assembled by a central joint allowing for left/right relative pitch⁶. Actually, this joint is made out of rubber and six relative degrees of freedom (d.o.f.) exist between the two parts of the chassis. The front wheels are rigidly connected to a common axle. This assembly is typically called a *rigid wheelset* and is also considered for the rear wheels.

Two three phase induction motors are coupled with the axles by means of reducers. These motors are supported by the chassis at two specific locations (see figure 7.15):

- at the back of the motor, a pin comes out of the motor and is introduced in a bushing (rubber joint) crimped in the chassis.
- at the front of the motor, a rod connects the latter with the chassis by means of another rubber bushing.

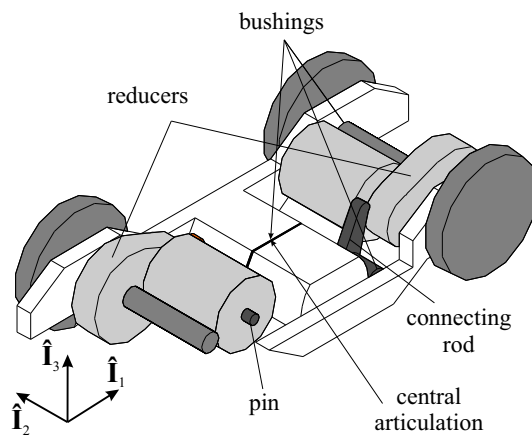


Figure 7.15: Bogie structure

The bushing elements between the wheelsets and the chassis are compliant elements and constitute the *primary suspension* of the bogie, ensuring -among others - a good lateral stability. Other bushings are present at the connections between the motors and the chassis.

⁶In railway vehicle dynamics, *pitch* refers to a rotation around a transversal axis, the longitudinal axis being aligned with the main motion direction. The *roll* motion is the rotation around the longitudinal axis, while the *yaw* motion is the rotation around a vertical axis

The behavior of a bogie is strongly influenced by the contact existing between the wheels and the rails. Their profiles, as well as the materials, determine this contact and will have to be taken into account in the wheel/rail contact model.

A typical simulation in railway dynamics, to assert for bogie stability, considers a system involving a carbody carried by a front bogie fully modeled and an ideal rear bogie, which perfectly follows the track [17]. Figure 7.16 illustrates this situation: the front of the carbody is supported by a bogie, through the *secondary suspension*, and the rear can only move vertically and longitudinally with respect to the track center. Pitch (around lateral axis) and yaw (around vertical axis) motions of the carbody are also allowed.

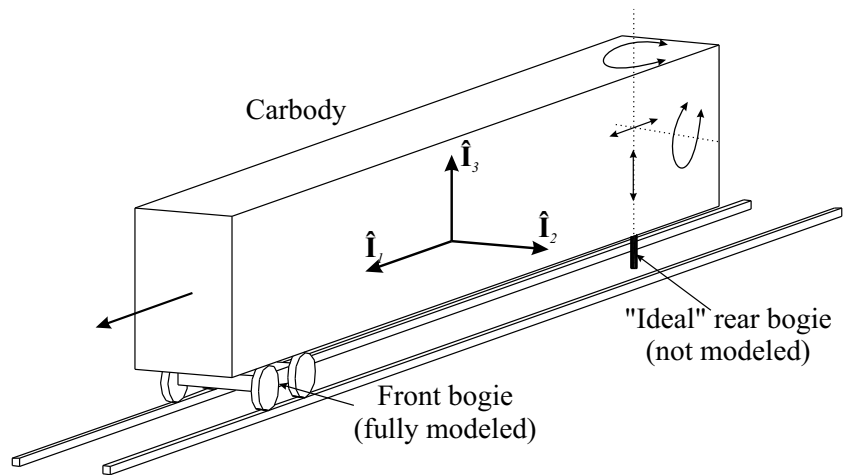


Figure 7.16: Complete system structure

The secondary suspension, represented in figure 7.17, consists of a *supporting bolster* resting on air cushions on both chassis. The carbody is connected to the bolster and only free relative yaw⁷ is allowed between these two bodies. The traction of the carbody is ensured by two longitudinal *traction rods* between the bolster and both parts of the chassis.

It is well-known that three phase actuators generate high torque oscillations during starting and the goal of this study is to investigate the effects of these oscillations on the mechanical structure of the bogie. We are therefore interested in obtaining time history of the efforts transmitted from the motors to the chassis through the bushings and connecting rods.

⁷relative rotation around a vertical axis.

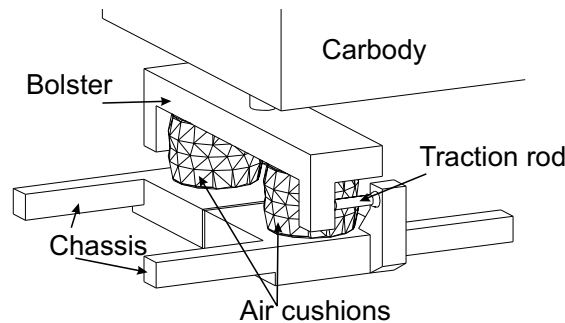


Figure 7.17: Secondary suspension

7.2.2 Bogie Model

7.2.2.1 The mechanical subsystem

Fourteen bodies can be defined to represent the system:

- one body for the carbody,
- one body for the bolster of the secondary suspension,
- two bodies for the articulated chassis of the front bogie,
- two bodies for the wheel axles,
- four bodies for the wheels (although rigid wheelsets are considered, for practical reasons concerning the contact model, distinct bodies are used for the wheels. The corresponding joints will of course be locked during the simulation),
- two bodies for each motor: the stator and the rotor.

From a modeling point of view, bushings are particular because together with the 6 relative d.o.f. between the interconnected bodies, they introduce high linear and angular stiffnesses in the three directions. According to Robotran's conventions (see section 5.3.1), they have been modeled by a sequence of fictitious bodies, as can be seen in figure 7.18, which represents the multibody model of the bogie. The latter consists of 57 bodies (14 "normal" and 43 "fictitious")⁸ and 6 kinematic loops opened by spherical cuts C^2 (= 18 independent constraints)⁹.

According to figure 7.18,

- Bodies 1 to 6 give the 6 d.o.f. of the right chassis with respect to the ground.
- Similarly, bodies 13 to 18 will introduce 6 relative d.o.f. between the left and right parts of the chassis.

⁸Fictitious bodies are defined in section 5.3.1.

⁹Spherical cuts were already used in the parking gate model, see section 7.1.

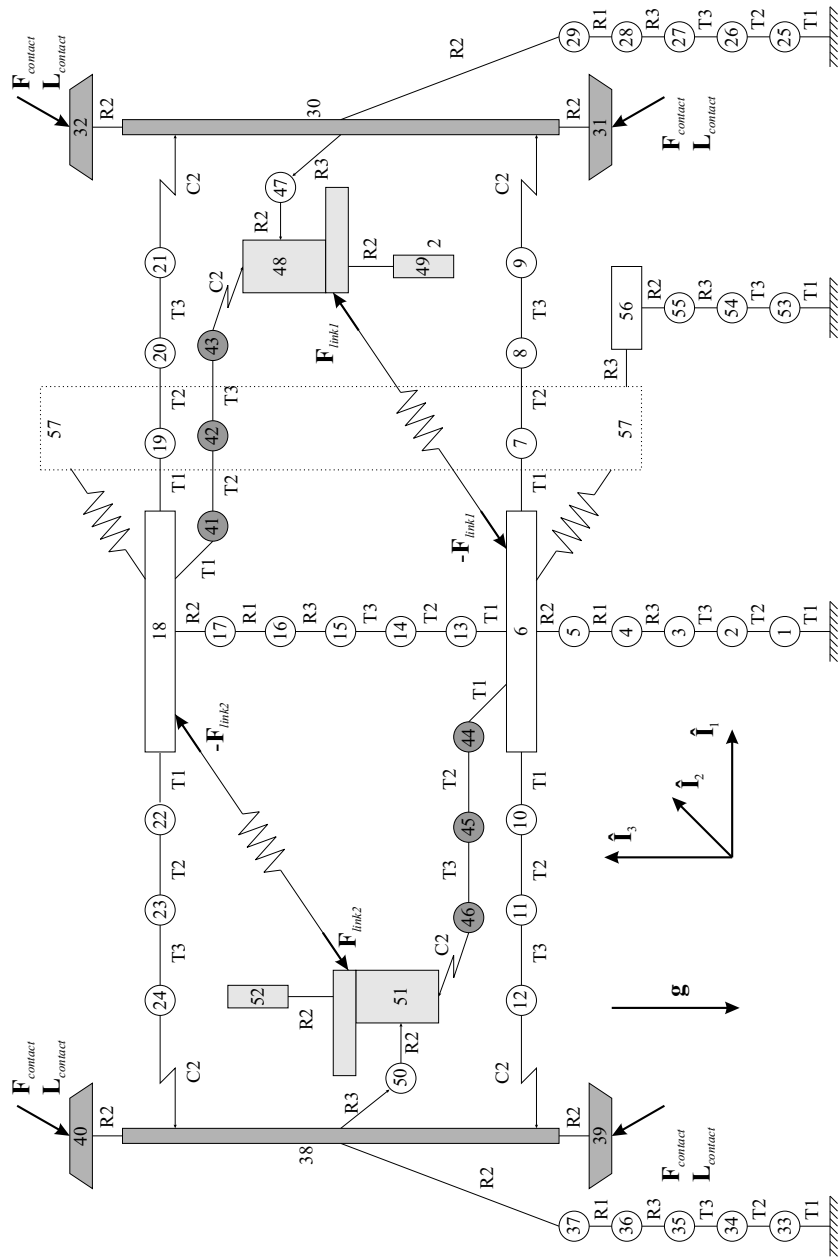


Figure 7.18: Multibody model of the bogie (Robotran's Conventions)

- The wheelsets are attached on the chassis by means of spherical bearings allowing free rotations in the three directions. Hence, the corresponding bushings work only in translation and are modeled by sequences of three fictitious bodies 7 to 9, 10 to 12, 19 to 21 and 22 to 24.
- Bodies 25 to 30 do not bring additional d.o.f. to the system but define the absolute position and orientation of the front wheelset, which will be necessary for the contact model. Of course, they could have been obtained from kinematic computations through the chassis but the length of the kinematic chains involved is such that large computation would be necessary. Introducing bodies 25 to 30 then reduces these kinematic computations while generating a new loop, which, in turn, requires additional kinematic computations.

All-in-one the computations are similar but adding bodies 25 to 30 allows to define a wheel/rail contact model based on joint variables and completely independent from the bogie structure.

Similarly, bodies 33 to 38 define the configuration of the rear wheelset.

- The front three phase actuator is logically modeled as a stator and a rotor with relative rotation (joint 49). The stator (body 48) is connected to the front wheelset by means of two revolute joints (joints 47 and 48). The joint coordinates s_{48}^m and s_{49}^m are related by the reduction ratio of the reducer: $s_{49}^m = k_{red} s_{48}^m$. Similarly, for the rear transmission, $s_{52}^m = k_{red} s_{51}^m$. The reduction ratio value is $k_{red} = \frac{97}{16}$.
- The stators are supported by both parts of the chassis by pins and bushings (bodies 41 to 43 and 44 to 46) on one side and by means of connecting rods and bushings on the other side. The latter is attached on the chassis and on the stator by means of spherical bearings allowing for free rotations and bushings with longitudinal elasticity. From a modeling point of view, the masses of the connecting rods and bushings can be neglected and they are replaced by equivalent springs acting on the stator and the chassis, as indicated in figure 7.18.
- Joints 53 to 56 give the degrees of freedom of the carbody (body 56), according to the ideal rear bogie.
- Body 57 represents the supporting bolster, which only has relative yaw (joint 57) with respect to the carbody. The secondary suspension between the bolster and the chassis is modeled by means of vertical and lateral springs representing the air cushions, while the *traction rods* are considered as very stiff springs.

Contact model When modeling railway systems, the wheel/rail contact must be precisely and carefully modeled because of its strong influence on the vehicle behavior.

In this research, our model is inspired from the general one described by Fiset and Samin in [15]. This model has been slightly simplified to meet our requirements and considers, for the geometric computations, conical profile for the wheel tread rolling on a “knife-edge” flat rail, as shown in figure 7.19. This is acceptable since we are interested in the time-simulation of a longitudinal motion along a straight track, what guarantees us small lateral displacements.

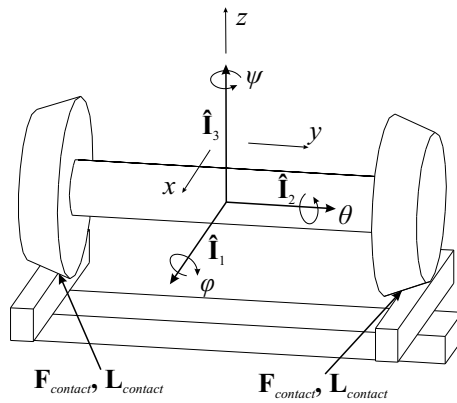


Figure 7.19: Single wheelset

Considering the single wheelset of figure 7.19, variables $x, y, z, \varphi, \psi, \theta$ define its configuration. According to the contact model described in [15], the wheels do not penetrate the rail what leads to one normal contact constraint for each wheel, and thus two constraints for the wheelset. Consequently, only four variables are independent: we chose x, y, ψ and θ .

In order to express the constraints, we have to find out which point on each wheel is in contact with the rail and we must impose that its vertical position corresponds to the height of the rail. The wheel point of contact can be determined from the position of the wheelset and its orientation. In this application, the bogie moves on a straight track, the yaw of the wheelset can thus be neglected for the geometric problem. Figure 7.20 represents a wheelset and defines all the quantities that will appear in the constraints.

Like in [15], we define the points O , some reference point fixed on the track, G , the wheel center of mass, Q , the contact point on the wheel and P , the contact point on the rail. Several frames are defined in [15] to locate the wheel point of contact Q . With the assumptions stated above, the following frames are sufficient in our case and are represented in figure 7.20, in which indexes R and L respectively refer to the right and left wheel:

- $\{\hat{\mathbf{I}}\}$, the inertial frame
- $[\hat{\mathbf{X}}] = A^G [\hat{\mathbf{I}}]$, the “geometrical-wheel” attached frame

- $[\hat{\mathbf{Y}}] = A^M [\hat{\mathbf{I}}]$, the “material-wheel” attached frame (note represented in figure 7.20)

Note: One should note the difference between the latter two frames. The second one $\{\hat{\mathbf{Y}}\}$ is attached to the physical wheel and rotates around its axis of symmetry according to the speed of the vehicle. The first one $\{\hat{\mathbf{X}}\}$ does not take this rotational speed into account and is attached to a so-called “frozen” wheel which would be rigidly attached to its bearing. In our case, wheelsets are considered and frame $\{\hat{\mathbf{X}}\}$ will be the same for both wheels. It corresponds to the frames attached to bodies 29 and 37 (see figure 7.18) for the front and rear wheelsets of our bogie, respectively.

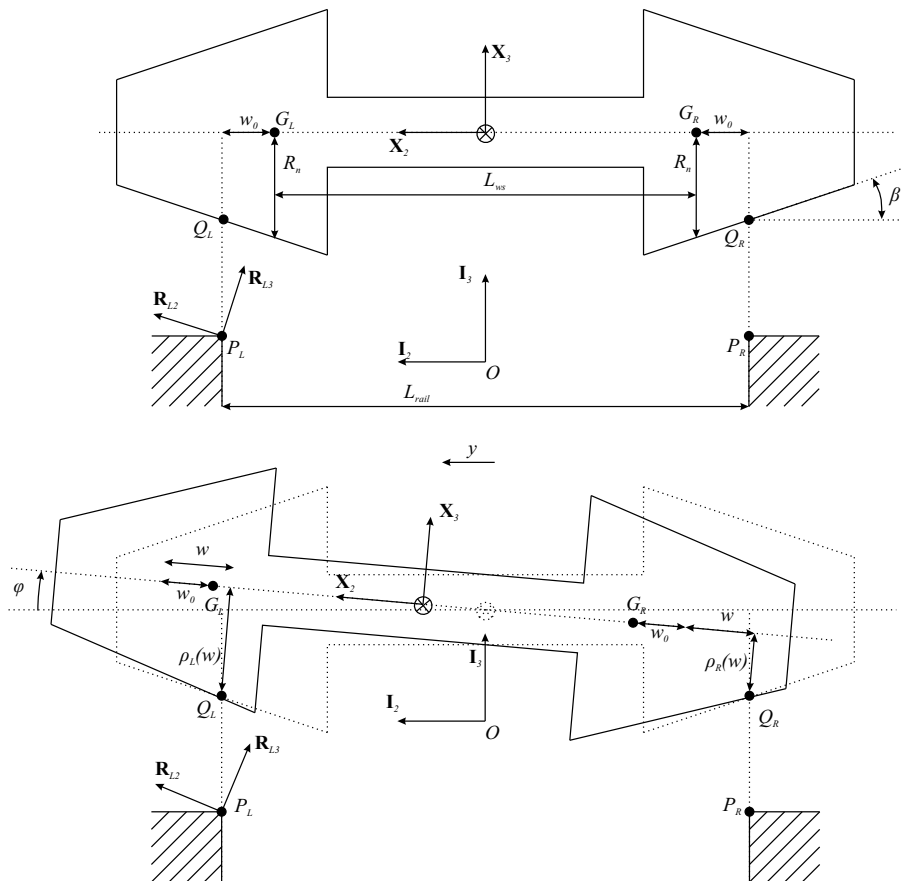


Figure 7.20: Contact model geometry

We may now express the position vectors of the centers of mass G_L and G_R and of the wheel contact points $\overrightarrow{G_L Q_L}$ and $\overrightarrow{G_R Q_R}$ as

$$\begin{aligned} \mathbf{x}_L = \overrightarrow{OG_L} \quad ; \quad \mathbf{w}_L = \overrightarrow{G_L Q_L} &= [\hat{\mathbf{X}}]^T \begin{pmatrix} 0 \\ w_0 - w \\ -\rho_L(w) \\ 0 \end{pmatrix} \\ \mathbf{x}_R = \overrightarrow{OG_R} \quad ; \quad \mathbf{w}_R = \overrightarrow{G_R Q_R} &= [\hat{\mathbf{X}}]^T \begin{pmatrix} 0 \\ -w_0 - w \\ -\rho_R(w) \\ 0 \end{pmatrix} \end{aligned} \quad (7.4)$$

where $\rho_L(w)$ and $\rho_R(w)$ are the wheel profiles defined by the following functions:

$$\rho_L(w) = R_n - \lambda(w_0 - w) \quad (7.5)$$

$$\rho_R(w) = R_n - \lambda(w_0 + w) \quad (7.6)$$

in which $\lambda = \tan(\beta)$ represents the slope, also called *equivalent conicity*, of the linear wheels profile.

The wheelset being part of the MBS, the position vectors x_L and x_R and the orientation of frame $\{\hat{\mathbf{X}}\}$ depend only on the generalized coordinates

$$s^m = \{x, y, z, \varphi, \psi, \theta\}$$

. On the other hand, we have introduced the auxiliary variable w to locate the points of contact. The constant w_0 is used to locate the nominal radius of the wheel R_n .

It is possible to relate w to the lateral displacement y and the roll angle φ of the wheelset (for a neglected yaw motion ψ):

$$w = \frac{y}{\cos(\varphi)} \quad (7.7)$$

We may now express the absolute position of the wheel contact points as:

$$\mathbf{u}_L = \mathbf{x}_L + \mathbf{w}_L = [\hat{\mathbf{I}}]^T \left(x_L(s^m) + A^G(s^m) \begin{pmatrix} 0 \\ w_0 - w \\ \rho_L(w) \end{pmatrix} \right) \quad (7.8)$$

$$\mathbf{u}_R = \mathbf{x}_R + \mathbf{w}_R = [\hat{\mathbf{I}}]^T \left(x_R(s^m) + A^G(s^m) \begin{pmatrix} 0 \\ -w_0 - w \\ \rho_R(w) \end{pmatrix} \right) \quad (7.9)$$

The wheels may not penetrate the rail and thus, the following kinematic contact constraints may be written:

$$\mathbf{u}_L \cdot \hat{\mathbf{I}}_3 = \mu_L(\mathbf{u}_L \cdot \hat{\mathbf{I}}_2) \quad (7.10)$$

$$\mathbf{u}_R \cdot \hat{\mathbf{I}}_3 = \mu_R(\mathbf{u}_R \cdot \hat{\mathbf{I}}_2) \quad (7.11)$$

where μ_L and μ_R represent the rail profile expressed in the $\{\hat{\mathbf{I}}_2, \hat{\mathbf{I}}_3\}$ plane. For this application, the rails correspond to a straight track on a horizontal plane and thus $\mu_L = \mu_R = 0$.

In order to obtain the Jacobian¹⁰ of these two contact constraints, we have to take their time derivative, what leads to the following relations at velocity level, according to [15]:

$$(\dot{\mathbf{x}}_L + \boldsymbol{\omega}^x \times \mathbf{w}_L) \cdot \hat{\mathbf{R}}_{L3} = 0 \quad (7.12)$$

$$(\dot{\mathbf{x}}_R + \boldsymbol{\omega}^x \times \mathbf{w}_R) \cdot \hat{\mathbf{R}}_{R3} = 0 \quad (7.13)$$

where vector $\boldsymbol{\omega}^x$ is the absolute angular velocity of frame $\{\hat{\mathbf{X}}\}$. In these last expressions, frame $\{\hat{\mathbf{R}}_L\}$ has been introduced and is defined as follows:

$$[\hat{\mathbf{R}}_L] = T_L^{\alpha_L} [\hat{\mathbf{I}}], \text{ with } T_L^{\alpha_L} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_L) & \sin(\alpha_L) \\ 0 & -\sin(\alpha_L) & \cos(\alpha_L) \end{pmatrix} \quad (7.14)$$

It is such that the unit vectors $\hat{\mathbf{R}}_{L1}$ and $\hat{\mathbf{R}}_{L2}$ belong to the tangent plane to the wheel at the rail contact point P_L , with $\hat{\mathbf{R}}_{L1} = \hat{\mathbf{I}}_1$. We thus have:

$$\alpha_L = \varphi + \beta \quad (7.15)$$

Frame $\{\hat{\mathbf{R}}_R\}$ is defined similarly and

$$\alpha_R = \varphi - \beta \quad (7.16)$$

Finally, the constraints must be expressed at acceleration level, to achieve the coordinate partitioning described in section 4.1.3 (see equations 4.31). The corresponding expressions can be found in [15].

Let us point out that using DAE solvers with systems involving this type of contact constraint, especially when considering the full contact model described in [15], can really not be envisaged. Obtaining the contact point position requires the use of a Newton/Raphson procedure when considering more complex nonlinear profiles for the wheel and the rails, and DAE solvers would not handle it [18].

Lateral and longitudinal tangent contact forces $\mathbf{F}_{contact}$, as well as the normal contact torques $\mathbf{L}_{contact}$ will be computed on the basis of the theory of Pr. Kalker [29, 30], which express them in frames $\{\hat{\mathbf{R}}_L\}$ and $\{\hat{\mathbf{R}}_R\}$, for the left and right wheel respectively. They will be transformed into the $\{\hat{\mathbf{Y}}_L\}$ and $\{\hat{\mathbf{Y}}_R\}$ wheel frames to comply with Robotran's conventions. In our bogie model, these forces are modeled as external forces on the leaf bodies representing the wheels, as shown in figure 7.18.

¹⁰necessary for the resolution and elimination of the constraints (see chapter 4)

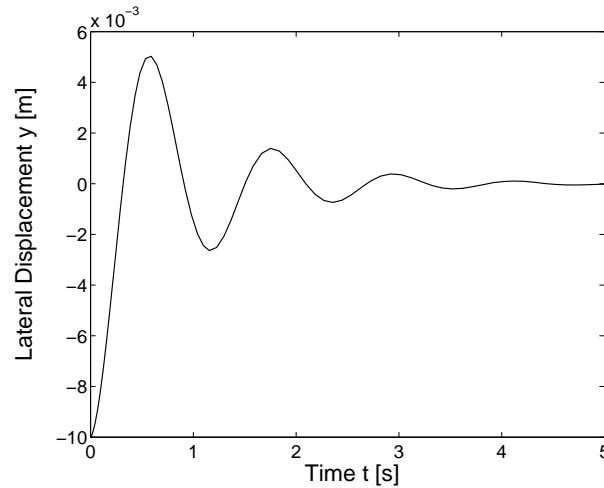


Figure 7.21: Stabilization of a single wheelset

The typical stable behavior of a single wheelset with the implemented contact model was obtained and is illustrated in figure 7.21. This result was confirmed by comparison with the FERRODYN software, which was developed on the basis of the 3D contact model [15], and was used for study of railway vehicles [14].

Degrees of freedom of the bogie To summarize, the MBS model of the bogie involves:

- 57 joint coordinates
- 18 loop closure constraints
- 4 constraints introduced by the contact model
- 2 additional constraints to model the reducers
- 4 locked joints (31, 32, 39 and 40) for the wheelsets

Consequently, our bogie model has 29 d.o.f. that correspond to the 6 d.o.f. of the right chassis (joints 1 to 6), the 6 relative d.o.f. of the left chassis (joints 13 to 18), the 2×4 d.o.f. of the wheelsets (joints 25, 26, 28, 30, 33, 34, 36 and 38), the actuators d.o.f. (joints 47, 49, 50 and 52), the carbody and bolster d.o.f. (joints 53 to 57).

7.2.2.2 Electrical subsystem

The actuators being three phase induction motors, we will reuse the model from section 6.3. The manufacturer provided us with the following parameters for their

motors:

$$\begin{aligned}
 p &= 2 & L_s &= 7 & (mH) \\
 R_s &= 0.03 & (\Omega) & L_r &= 7 & (mH) \\
 R_r &= 0.025 & (\Omega) & M_s &= -3.4 & (mH) \\
 & & & M_r &= -3.4 & (mH) \\
 & & & M_{sr} &= 6.65 & (mH)
 \end{aligned} \tag{7.17}$$

Note: In the actual system, the motors are fed through power inverters. In this work, we do not consider variable topology circuits and we assume the inverter to provide equilibrated sinusoidal phase voltages to the actuator.

7.2.3 Simulation results

7.2.3.1 Mechanical model calibration

The mechanical MBS model being quite complex, it was constructed in two steps:

- firstly, we modeled the whole system for a simplified articulated bogie, from which the motors have been removed. The corresponding model is obtained by suppressing bodies 41 to 52 in the MBS represented in figure 7.18.
- secondly, the actuators have been introduced and the whole system shown in figure 7.18 was considered.

Both bogies have been time simulated with initial longitudinal velocity of $10m/s$ and initial lateral offset of $2cm$. As shown in figure 7.22, both systems behave similarly and stabilize well after $6s$, what is acceptable for a good comfort of the passenger. Introducing the actuator bodies and the corresponding bushings (softer than the primary suspension bushings) only slightly changes the global “yaw stiffness” of the bogie, which, together with the wheel equivalent conicity λ , is one of the most influential parameters for lateral stability. This explains why both systems behave almost identically.

The vertical forces in the front and rear secondary suspensions of the complete bogie are plotted in figure 7.23. Both have a mean value of approximately $60000 N$ corresponding to half of the weight of the carbody. This confirms a good front/rear weight repartition of our system. Nevertheless, persistent vibrations are introduced by the additional bodies corresponding to the actuators.

7.2.3.2 Starting phase of the system with actuators

We assumed¹¹ a U/F control, similar to the one presented in section 6.3, from 0 to $50 Hz$ (0 to $290 V$) over 10 seconds for both motors. That corresponds to a typical subway acceleration of approximately $1.1 m/s^2$. The bogie starts in the equilibrium configuration.

¹¹We did not obtain further information on the control unit of the motors.

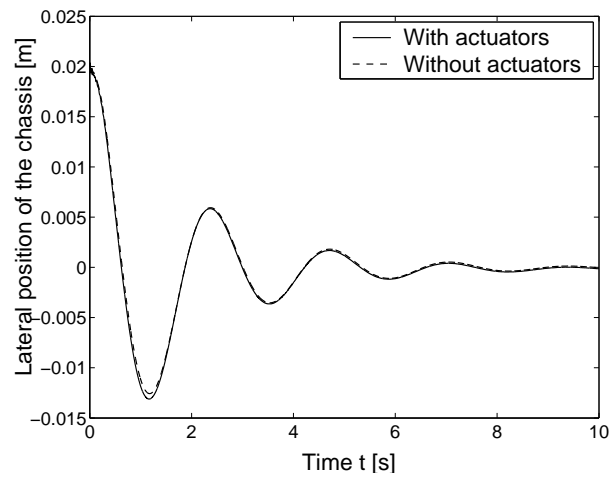


Figure 7.22: Stabilization of the articulated bogie with and without the actuators

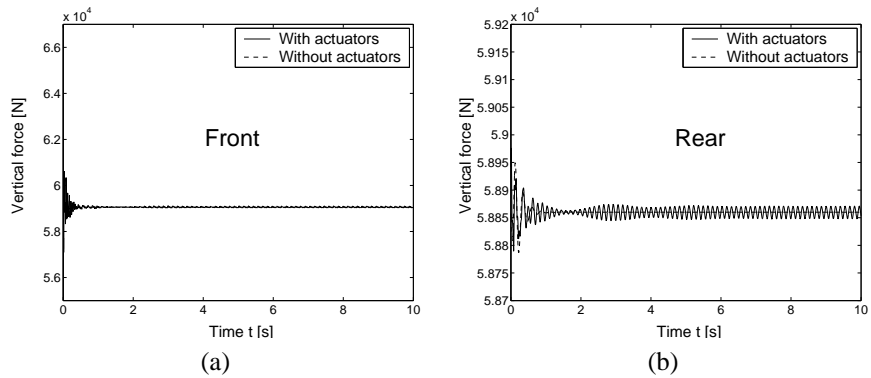


Figure 7.23: Front (a) and rear (b) vertical forces in the secondary suspension

Figure 7.24.a shows the global longitudinal velocity of the bogie. After 10s, the bogie reaches a velocity of 11.25m/s ($= 40.5\text{km/h}$). When torques are applied on the wheelsets by the motors, the articulated chassis deforms and left/right asymmetry appear, leading to a lateral offset of the bogie on the track (see figure 7.24.b).

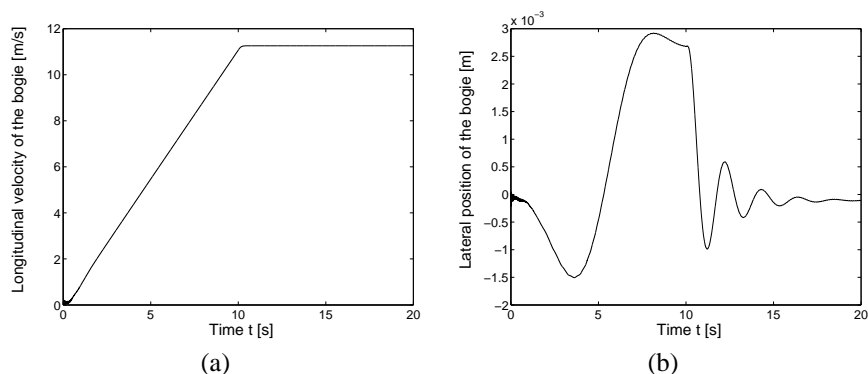


Figure 7.24: Longitudinal velocity (a) and lateral position (b) of the chassis

The time history of the efforts inside the central articulation between both chassis is plotted in figure 7.25. It confirms that, under the action of the motors, deformations take place in the central articulation of the chassis. Only longitudinal (figure 7.25.a), vertical (figure 7.25.c) and pitch (figure 7.25.f) deformations are significant.

The longitudinal (plot a), lateral (plot b) and vertical (plot c) forces inside the bushing supporting the pin of the front stator (joints 49 to 51) are represented in figure 7.26, where the electromechanical torque of the front actuator is also plotted (figure 7.26.d). Its absolute value is always below the maximal authorized torque of 1215Nm , extracted from the motor datasheets. The negative value of the electromechanical torque is logical since, because of the transmission, the motor and the wheelset have inverse directions of rotation.

The vertical force inside the bushing (figure 7.26.c) is higher during acceleration, what is logical, since the motor tends to lower itself when applying its torque. This is also shown in figure 7.27, where the forces inside the springs representing the bushings of the connecting rod, are plotted. As one can see, the front bushing is compressed (positive force) when the front motor applies a torque on the front wheelset. Compared to the front assembly, the rear motor is at the opposite side of the rear wheelset and it will tend to higher itself. This result in an extension (negative force) of the bushing associated with the connecting rod: forces in figures 7.27.a and b have opposite signs during the acceleration. When the desired value is reached, the torques vanish and the motors come back to an equilibrium position and both connecting rods are compressed and sustain the motors.

Eventually, one statoric current through the front actuator has been plotted in figure 7.28, which also shows one rotoric current. The frequency of the statoric current rises from 0 to 50Hz, according to the voltage inputs frequency imposed by the U/F control.

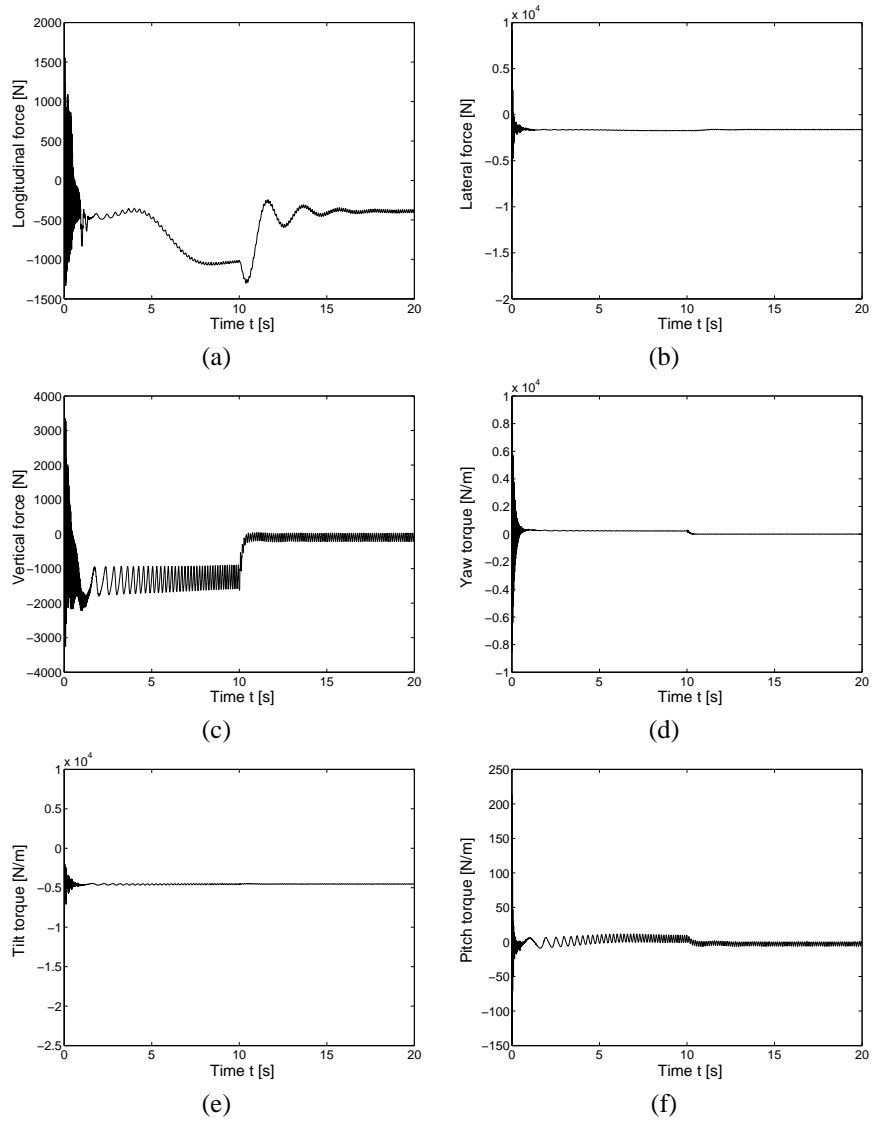


Figure 7.25: Chassis Central Articulation Efforts

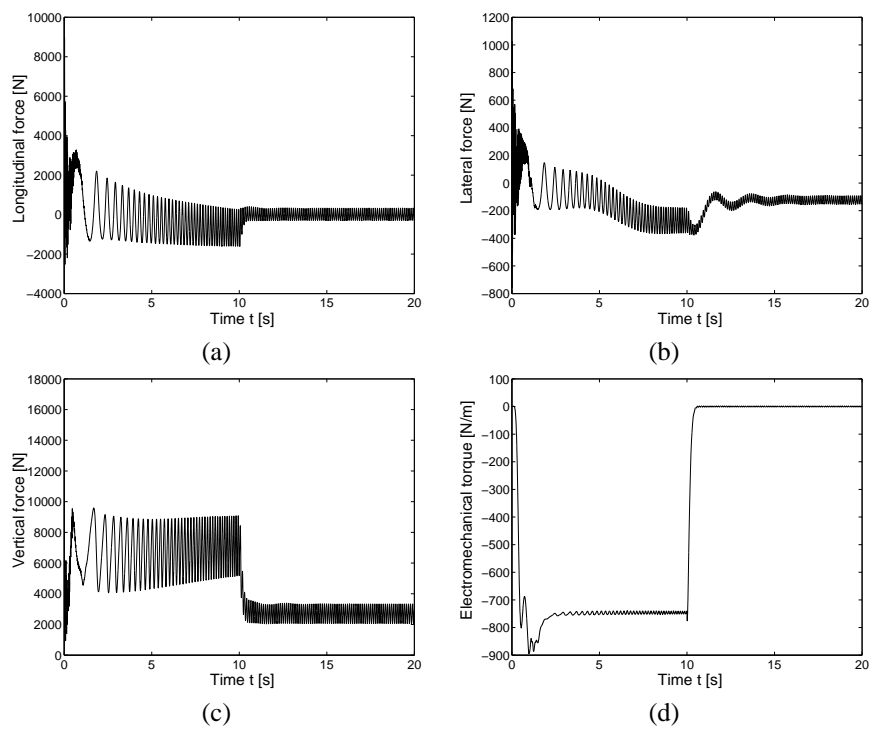


Figure 7.26: Front motor - Forces inside the pin bushing and electromechanical torque

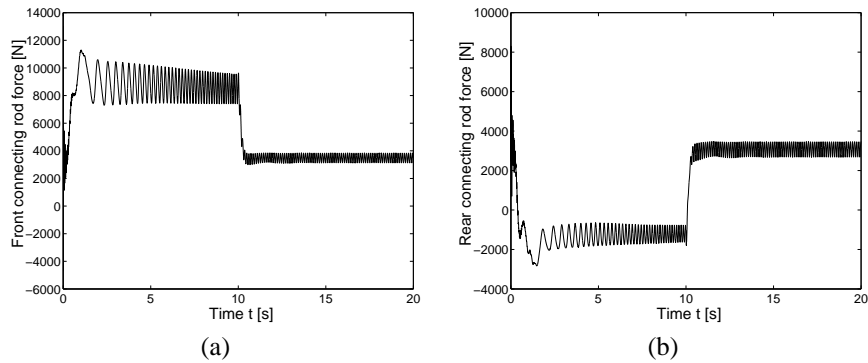


Figure 7.27: Forces inside the connecting rod for the front (a) and rear (b) motors

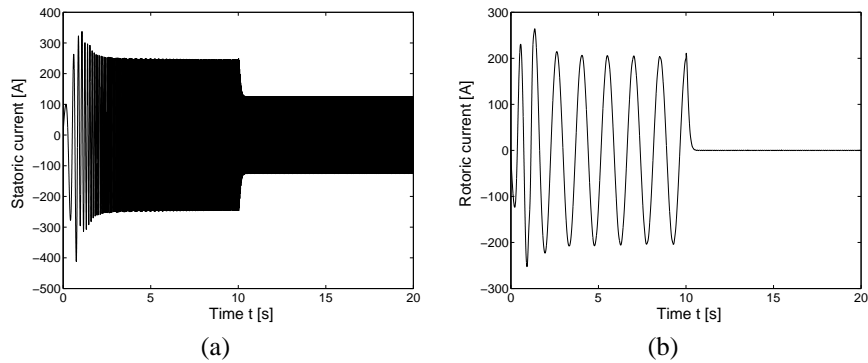


Figure 7.28: Front motor - Statoric (a) and rotoric (b) currents

Bombardier subway bogies are commonly driven by DC-motors, but their motor manufacturer proposed to replace them by three phase inductive motors. The latter are characterized by important torque oscillations when starting and Bombardier's engineers were worried about possible additional vibrations of the chassis, due to these torque oscillations. Indeed, additional vibrations could lead to fatigue issues, which should be avoided. The main objective in studying this bogie was thus to analyze the influence of the torque oscillations during starting of the bogie. After comparing the results shown here with those obtained with classical DC-motors¹², we could not conclude for significant additional vibrations induced on the chassis of the bogie, when introducing three phase actuators.

¹²These results are not shown here.

8

Conclusions

The main objective of this thesis was to provide a tool for generating automatically the models of complex electromechanical applications. We focused on large mechanical multibody systems (MBS) coupled with electromechanical converters, in which tight interactions take place.

In the first part of this thesis, classical strategies for building multidomain models have been considered. Because of the numerical uncertainties associated with a block-diagram strategy, we rejected the latter. Strategies based on modeling languages, although they offer an interesting alternative, only provide a framework for implementing formalism and existing libraries did not satisfy us in the present context. Moreover, they usually resort to differential algebraic (DAEs) solvers, which we decided to avoid as far as possible because of the numerical instability and drifts that might appear when considering highly nonlinear systems as in multibody dynamics. A strategy based on unified theories was the last alternative that we investigated.

Three unified theories: Bond graph, Linear graph and Virtual Work Principle, have been confronted in chapter 3. In [60] and [45] (for Bond Graph and Linear Graph theories), we pioneered by comparing the use of these approaches for modeling electromechanical multibody systems and their use for practical applications.

In chapter 3, the Virtual Work Principle was presented in detail in order to highlight the parallelism existing between mechanical and electrical models. For mechanical systems, unlike classical formulations at local level for mechanical systems [76, 67, 31], we deal with infinitesimal displacements at macroscopic level, by starting from the Newton translation equations and the Euler rotation equations. This formu-

lation was inspired from the Virtual Power Principle presented in [56], and from the work done by Wittenburg [77] and Samin [55]. For electrical circuits, the Virtual Work Principle was inspired from [26].

The main conclusions of this confrontation are the following:

- Bond graphs offer a rigorous interfacing between mechanical and electrical systems when considering 1D systems but when applied to 3D MBS, they suffer from several disadvantages: constraints at velocity level, frame dependent entities, graph's structure different from system's structure, causality assignments, etc.
- Linear graphs were at first designed for 1D systems but their use on 3D multi-body system is quite straightforward [44], eventhough not always as efficient as dedicated procedure for large MBS. They offer an interesting framework for modeling MBS coupled with electromechanical converters. Nevertheless, the obtained equations are less efficient than recursive formalisms when considering large MBS. Although Linear graph theory does not preclude it, as far as we know, current implementations do not consider recursive formulation.
- Virtual Work Principle allows to generate the reduced system of equations and helps demonstrating some interesting properties of the mathematical entities appearing in the dynamical equations. Lagrange equations deduced from the Virtual Work Principle allow for rigorous interfacing of electrical and mechanical subsystems. Nevertheless, because they involve partial derivatives and multiple summations, they are inefficient when considering large d.o.f. MBS. Moreover, physical insight is lost for electrical circuits, for which the equations are expressed in terms of unmeasurable quantities.

These discussions on modeling strategies led us to the conclusion that none of the existing strategies for building multidomain models (block diagrams or strategies based on modeling languages or unified theories) is fully satisfactory when considering complex electromechanical systems.

In the second part of this research, we thus propose a new approach, which takes advantage of the symbolic programming technique. Dedicated formalisms are used to generate the symbolic models of the multibody and electrical subsystems, which are then assembled into a global symbolic electromechanical model. The possibility to use dedicated formalisms was an advantage of the block-diagram strategy but, in our approach, the assembly of the obtained submodels is not achieved at numerical level. On the contrary, we decided to provide a global model to the numerical tool.

In chapter 4, the dedicated formalisms which were implemented are presented in details:

- for mechanical multibody systems, the Newton-Euler Recursive formalism was chosen. Thanks to the use of generalized joint coordinates and recursive computations, it is more efficient than most of the other techniques when considering large multibody systems [56]. Moreover when constrained MBS are considered,

Lagrange multipliers technique applies and constraint reduction is then straightforward and naturally follows the generation of the equations. The coordinate partitioning [75] approach was also presented, since ODEs are preferred.

- for electrical circuits, a new circuit based formalism was developed. On the basis of an original representation of electrical circuits, using topological concepts similar to those from MBS theory, the dynamic equations of electrical circuits are obtained and reduced. This reduction uses an “electrical coordinate partitioning” technique, inspired by the mechanical one.

Once generated, the mechanical and electrical submodels are coupled and a unified symbolic electromechanical model is built. As shown in section 3.3.3, the electromechanical interaction is introduced by means of an additional force acting on the multi-body subsystems. From a mechanical point of view, the electromechanical converter and its dynamics can be interpreted as a “constitutive force” element, like a spring or a damper. From an electrical point of view, some electrical parameters become functions of the mechanical variables and complex mechanical loads can be coupled with the actuators models.

The submodels can be generated separately by different tools but they have to be coupled rigorously, according to the flowcharts of figures 4.11 and 4.13.

Symbolic implementation was discussed in chapter 5 and the two model generators were presented:

- Robotran is used for generating in a recursive manner the MBS model, in which the interaction with the electrical model must be introduced via a force “element” (joint force or external force).
- Electran was developed during this research and is the extension of Robotran to electrical circuits and electromechanical converters. Based on Robotran’s symbolic kernel, it generates symbolic models of electromechanical converters involving mutual inductive and electrostatic effects. Permanent magnets are also considered.

In the third part of this thesis, we show our ability to model complex electromechanical systems. After validation of the softwares, two industrial applications have been considered:

- First, the flexible barrier which is a typical system where tight electromechanical interaction takes place: mechanical natural frequencies of the barrier and the electrical variables are of the same order of magnitude. The comparison that we conducted with the different gear ratios highlighted the changes in the mutual influence of the electrical and mechanical subsystems. The comparison with the linear graph approach confirms the validity of our results but also points out that there is never only one way to solve a problem, and that most of the time, using one or other approach depends on user’s background and habits.

The damping of mechanical vibrations through dissipation in the electrical resistors illustrates the tight electromechanical interaction.

- Second, a subway bogie driven by three phase induction motors illustrated that our approach is applicable to a very complex structure, involving many d.o.f., constraints and wheel/rail contact model. From a numerical point of view, because of the very high stiffness and the algebraic constraints, this model is very demanding. Thanks to the systematic symbolic reduction of the equations, we were able to provide ODE's to the numerical integrator, protecting us from several numerical issues and reducing the simulation times. This last example also perfectly illustrates the type of applications aimed by this work. We do not fear to say that only few tools, if any, are able to deal properly with such applications and with reasonable simulation times.

Future work and prospects At this stage of the research, several orientations can be envisaged for future investigations:

- From an implementation point of view, the present version of Electran suffers from several weaknesses which should be addressed:
 - First, the current implementation only considers the possibility to have one mechanical variable influencing the electrical parameters. This should be generalized to take into account several mechanical influences, present in multi-d.o.f. actuators, for instance.
 - Secondly, checking for the independence of the Kirchoff's equations should be implemented in order to allow the user to input the electrical circuits without taking care of independency of the described loops.

Although the presented applications (see Part III) only consider electromechanical actuation, sensors which can be represented as a multibody system, could also be modeled using our approach. Sensors are usually designed so that their dynamics has a very slight influence on the main system, and modeling them might be superfluous. However, for instance when considering micro-mechatronic applications, the sensor and the system might have similar dimensions and a model of the whole system could be of interest.

- From a modeling point of view, the proposed approach should be adapted to take into account possible nonlinearities of the electrical constitutive equations. This could, for example, help for modeling magnetic saturation effects. This would require the resolution of nonlinear systems of equations and applying the technique proposed in [49] might be a good solution.

It should be possible to use the circuit based formalism for generating “inverse” dynamic model of electromechanical converter, allowing for computation of the electrical sources inputs corresponding to a given trajectory of the mechanical load. Inverse dynamics is used by control engineer when developing feedforward controllers.

It would be interesting to couple our models with those generated by Dan Teltu [71] for power electronic circuits (with variable topology). This way, the

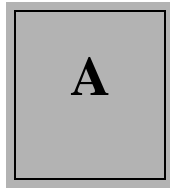
complete system could be modeled, as was already done in [71] on a specific application.

Finally, application of the proposed modeling strategy to other fields, like hydraulic or pneumatic systems, should be considered. Indeed, the flowchart presented in figure 4.12 can be generalized to multibody system involving additional constitutive forces of hydraulic or pneumatic origin, which can only be computed, if the complete mechanical configuration is known. However, variable topology considerations would have to be included to deal with valves, just like with power electronic circuits involving switches.

- From a numerical point of view, the idea of using, adapting or developing numerical ODE solvers able to deal with multi-time scale systems should be investigated. Indeed, even when the dynamics of the system involves very different frequencies, it might be of interest to consider the full dynamics, instead of filtering high frequencies by using stiff numerical integrators, for example. This happens when considering flexible bodies, among rigid bodies, introducing high frequencies, as in [18] and [23]. In the present context, the electrical dynamics also introduces high frequencies which might be of interest for electrical engineers.

When important differences exist between the frequency ranges, considering multi time-scale numerical integrators could reduce the computation times.

Acknowledgments This work has been supported by the Fonds National de la Recherche Scientifique (FNRS) and the Belgian Program on Interuniversity Attraction Poles initiated by the Belgian State, Prime Minister's office, Science Policy Programme (IAP-IV/24 and AMS IAP V/06). The scientific responsibility is assumed by its authors.



Comparison Paper

A Comparison of Different Methods for Modelling Multibody Systems with Electrical Drives

L. Sass*, J. McPhee*, C. Schmitke*, P. Fisette, D. Grenier

March 28, 2003

*Center for Research in Mechatronics
Université Catholique de Louvain
Bâtiment Stevin
Place du Levant, 2
1348 Louvain-La-Neuve (Belgium)*

**Systems Design Engineering
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1*

Abstract

Procedures for modelling multibody systems are well-known and many formulations and tools are available for these types of systems. For several years, emphasis has been placed on the modelling of electromechanical systems, particularly multibody systems that are driven by electrical actuators, like robots. This paper presents and compares three different modelling approaches based on different theories: the Principle of Virtual Work, Linear Graph Theory and Bond Graph theory. Three examples, including non-academic applications illustrate this comparison.

Keywords: multibody dynamics, electromechanical systems, modelling, linear graph, bond graph, Virtual Work, comparison

1 Introduction

For several years the requirements for technological systems has pushed the limits of the discrete design approach. In this approach each physical part is designed independently from the others. In order to meet today's design requirements (precision, size reduction, low costs, etc.), an integrative design approach is necessary. At the same time, modelling has taken on larger place in the design process as a "pre-prototyping" stage allowing for a cheaper and faster first analysis of a system. Integration and modelling naturally lead to multidomain modelling, a major field of research over the last few years.

A particular field of applications concerns electromechanical systems where multibody mechanical sub-systems interact with electrical circuits through magnetic or electrostatic fields. This involves systems such as electrically actuated multibody systems or electromechanical sensors.

From a theoretical background, there are two main approaches for modelling multidomain systems: graph theory and mathematical approach. On one side, based on energetical analogies between the different physical fields, an approach using graph theory represents the system as a connection of generic elements present in every system: energy storage elements, energy dissipative elements and energy sources. Bond Graph and Linear Graph Theory are two principal graph theories. On the other side, mathematical approaches directly

*aspirant FNRS

manipulate equations of the various fields and combine them to formulate a unique set of final dynamic equations. The unification of the equations is based on energy considerations such as the Virtual Work Principle, Virtual Power, Energy Conservation,... To be both rigorous and efficient, this unification should be achieved before any numerical processing. Therefore, symbolical formulations are good candidate for obtaining the global system equations.

This paper proposes a comparison between three formalisms respectively based on the Virtual Work Principle, Linear Graph Theory and Bond Graph Theory. In Section 2, a theoretical description of each approach is presented using the condensator speaker [1], a classical electromechanical system. Section 3 proposes two different applications for comparing the approaches. The first application is a two-link robot driven by DC-motors, while the second one is a barrier, modelled as a flexible beam, driven by a three-phase asynchronous actuator. It is used for controlling access to parking lots or highways. Section 4 concludes the paper and points out the important features of the different approaches.

2 Modelling Theories

2.1 Introduction

This section briefly outlines the three theoretical approaches that are compared. For each approach, a description of the basic concept is offered, followed by discussions about the use of these concepts in modelling multibody systems, electrical systems, and electromechanical systems.

A common example, the condensator speaker [1], will help the reader understand the basics of these three approaches. Figure 1 shows a schematic description of the system. It consists of a capacitor connected in series with an inductor, a resistor and a voltage source. The upper plate of the capacitor is allowed to move vertically and is connected to the ground by a spring-damper suspension. More complex systems are considered in Section 3.

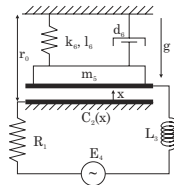


Figure 1: Schematic of the condensator speaker

2.2 Virtual Work Principle

2.2.1 Basics

Mathematical formulations are always based on energetical considerations: Virtual Work, Virtual Power, Energy Conservation Principle,... The Principle of Virtual Work applied to electromechanical system is described in three steps, starting from the established mechanical formulation, continuing with the electrical formulation and finishing with electromechanical formulation. At the end of this section the Principle of Virtual Work is used to obtain the governing equations of a condensator speaker.

2.2.2 Virtual Work and Multibody Systems [2]

A multibody system consists of rigid bodies interconnected by joints (e.g. revolute or prismatic joints). The absolute position vector \mathbf{x} of any material point of a body can be written as:

$$\mathbf{x} = \mathbf{r} + \mathbf{X}(\boldsymbol{\psi}) \quad (1)$$

where \mathbf{r} is the absolute position vector of the center of mass of the rigid body and $\mathbf{X}(\boldsymbol{\psi})$ is the relative position vector of the material point with respect to the center of mass. $\mathbf{X}(\boldsymbol{\psi})$ has constant components in a body-fixed frame whose absolute orientation can be described by, at least, three parameters $\boldsymbol{\psi}$, the Euler angles for instance. Hence, the rigid body configuration is fully determined by 6 variables: the three components of \mathbf{r} and the three parameters $\boldsymbol{\psi}$. This means $6N$ variables for a multibody systems consisting of N bodies.

For most systems, a reduced set of generalized variables \mathbf{q}_m is sufficient, depending on the system's topology. In this case:

$$\mathbf{x}(\mathbf{q}_m) = \mathbf{r}(\mathbf{q}_m) + \mathbf{X}(\mathbf{q}_m) \quad (2)$$

Applied to a multibody system of N bodies, the Principle of Virtual Work, also known as d'Alembert's Principle, states that for any virtual displacements defined by vectors $\delta\mathbf{r}$ and $\delta\boldsymbol{\psi}$ ¹,

$$\sum_{i=1}^N \left(\delta\mathbf{r}_i \cdot (m_i \ddot{\mathbf{r}}_i - \mathbf{F}_i) + \delta\boldsymbol{\psi}_i \cdot (\dot{\mathbf{H}}_i - \mathbf{L}_i) \right) = 0 \quad (3)$$

in which,

- $\ddot{\mathbf{r}}_i$ is the absolute acceleration of the center of mass of body i
- $\dot{\mathbf{H}}_i$ is the time derivative of the angular momentum of body i with respect to its center of mass
- \mathbf{F}_i and \mathbf{L}_i are respectively the resultant force applied to body i and the resultant torque with respect to its center of mass

From equation (2), the virtual quantities $\delta\mathbf{r}_i$ and $\delta\boldsymbol{\psi}_i$ can be expressed in terms of generalized virtual displacements $\delta\mathbf{q}_m$. Equation (3) can thus be rewritten in terms of the generalized variables \mathbf{q}_m , $\dot{\mathbf{q}}_m$, $\ddot{\mathbf{q}}_m$ and generalized virtual displacements $\delta\mathbf{q}_m$, leading to:

$$\boldsymbol{\Phi}_m^T(\mathbf{q}_m, \dot{\mathbf{q}}_m, \ddot{\mathbf{q}}_m) \cdot \delta\mathbf{q}_m = 0 \quad (4)$$

It can easily be shown [2] that:

$$\boldsymbol{\Phi}_m(\mathbf{q}_m, \dot{\mathbf{q}}_m, \ddot{\mathbf{q}}_m) = \mathbf{M}_m(\mathbf{q}_m) \ddot{\mathbf{q}}_m + \mathbf{Q}_m(\mathbf{q}_m, \dot{\mathbf{q}}_m) \quad (5)$$

where $\mathbf{M}_m(\mathbf{q}_m)$ is the symmetric positively defined mass matrix of the system and $\mathbf{Q}_m(\mathbf{q}_m, \dot{\mathbf{q}}_m)$ is the vector containing the generalized forces (joint forces \mathbf{Q} and external forces) as well as the Coriolis, gyroscopic and centrifugal terms.

For unconstrained multibody systems, equation (4) is valid for any choice of virtual displacements $\delta\mathbf{q}_m$, and thus:

$$\begin{aligned} \boldsymbol{\Phi}_m &= 0 \\ \Leftrightarrow \mathbf{M}_m(\mathbf{q}_m) \ddot{\mathbf{q}}_m + \mathbf{Q}_m(\mathbf{q}_m, \dot{\mathbf{q}}_m) &= 0 \end{aligned} \quad (6)$$

Note that Lagrange's Equations can be derived from d'Alembert's Principle [2], leading to:

$$\boldsymbol{\Phi}_m(\mathbf{q}_m, \dot{\mathbf{q}}_m, \ddot{\mathbf{q}}_m) = \frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\mathbf{q}}_m} \right) - \frac{\partial T}{\partial \mathbf{q}_m} + \frac{\partial V}{\partial \mathbf{q}_m} + \mathbf{Q}'^T = 0 \quad (7)$$

¹As shown in [2], vector $\delta\boldsymbol{\psi}$ can only be defined for infinitesimal motions. It then behaves like the angular velocity vector.

where T is the kinematical energy, V is the potential energy, \mathbf{Q}^* denotes the non-potential generalized forces vector.

Equation (6) is the standard form of the equations of motion for unconstrained multibody system. The principle of Virtual Work can be seen as a linear combination of the Newton-Euler equations. Choosing an adequate set of virtual displacements $\delta \mathbf{r}_i$ and $\delta \psi_i$ in terms of the generalized coordinates $\delta \mathbf{q}_m$ makes the constraints forces (e.g. the reaction forces in a joint) disappear from (3), leading directly to the equations of motion in terms of the generalized variables \mathbf{q}_m , $\dot{\mathbf{q}}_m$ and $\ddot{\mathbf{q}}_m$. Using classical Newton-Euler approaches involves the constraint equations and requires much more computation.

For constrained systems, it is not always desirable to choose a minimal set of independent generalized coordinates because of the complexity of the resulting expressions. From a practical point of view, a non-minimal set of generalized coordinates is often used. In these cases the variables are not independent because of the explicit constraints relating them.

Most of the time, these constraints are holonomic, relating the generalized coordinates at the position level:

$$\mathbf{h}(\mathbf{q}_m) = 0 \quad (8)$$

The latter must be satisfied at any time and thus, these constraints must also be satisfied at velocity and acceleration levels:

$$\begin{aligned} \dot{\mathbf{h}}(\mathbf{q}_m, \dot{\mathbf{q}}_m) &= \mathbf{J}_m(\mathbf{q}_m) \dot{\mathbf{q}}_m = 0 \\ \ddot{\mathbf{h}}(\mathbf{q}_m, \dot{\mathbf{q}}_m, \ddot{\mathbf{q}}_m) &= \mathbf{J}_m(\mathbf{q}_m) \ddot{\mathbf{q}}_m + \dot{\mathbf{J}}_m \dot{\mathbf{q}}_m = 0 \end{aligned} \quad (9)$$

where \mathbf{J}_m is the Jacobian of the constraint, defined as $\mathbf{J}_m(i, j) = \left(\frac{\partial h_i}{\partial q_j} \right)$.

Note that non-holonomic constraints, expressed at velocity level, can also be considered in the present formalism.

Using the Lagrange multipliers technique, it can be shown that the equations of motion can be written as (see [2]):

$$\mathbf{M}_m(\mathbf{q}_m) \ddot{\mathbf{q}}_m + \mathbf{Q}_m(\mathbf{q}_m, \dot{\mathbf{q}}_m) = \mathbf{J}_m^T \boldsymbol{\lambda}_m \quad (10)$$

where $\boldsymbol{\lambda}_m$ is a column vector containing the p Lagrange multipliers corresponding to the p constraints (8).

Elimination of the Lagrange multipliers, using a coordinate partitioning method [4], leads to a reduced set of $n - p = \#dof^2$ equations with a reduced mass matrix \mathbf{M}_{mr} and generalized forces \mathbf{Q}_{mr} :

$$\mathbf{M}_{mr}(\mathbf{q}_m, \dot{\mathbf{q}}_m) \ddot{\mathbf{u}}_m + \mathbf{Q}_{mr}(\mathbf{q}_m, \dot{\mathbf{q}}_m, t) = 0 \quad (11)$$

in which \mathbf{u}_m denotes a set of $n - p$ independent generalized coordinates.

From a computer implementation point of view, recursive formalisms [3] are an efficient way to obtain equations (6) or (8), (9), (10) and (11) when large multibody systems are considered. This formalism has been implemented in the software ROBOTRAN^{®3} [34], which is capable of symbolically generating the equations of motion for any multibody system. The main advantages of the symbolic formulation are the opportunity for drastic simplifications and the portability of the obtained equations towards various computer environments (simulation, optimization and control). These simplifications lead to fewer computations during simulation.

2.2.3 Virtual Work and Electrical Systems

The Principle of Virtual Work, originally formulated for mechanical systems, has been extended to other physical domains, in particular to electrical systems [1, 6, 7, 8].

The electrical energy of an electrical circuit is defined by $E = \int \mathbf{u}^T \mathbf{i} dt = \int \mathbf{u}^T d\mathbf{q} = \int \mathbf{i}^T d\boldsymbol{\varphi}$, in which \mathbf{u} and \mathbf{i} are the vectors of voltages across the elements of the circuit and the currents flowing through them. $\mathbf{q} = \int \mathbf{i} dt$ represents the amount of charge accumulated in the elements constituting the circuits while

²#dof = number of degrees of freedom

³www.prm.ucl.ac.be/robotran

$\varphi = \int \mathbf{u} dt$ represents the fluxes passing through the same elements. These two ways of calculating the electrical energy comprise the two main formulations for the Principle of Virtual Work:

- Formulation in terms of charge variations leading to charges as generalized variables.

$$\mathbf{u}^T \delta \mathbf{q} = 0 \quad (12)$$

- Formulation in terms of flux variations leading to flux as generalized variables.

$$\dot{\mathbf{i}}^T \delta \phi = 0 \quad (13)$$

Starting from the charge formulation (12) of the Principle of Virtual Work, it can be shown [1] that the dynamic equations for an electrical system, the "electrical equations of motion", can be written as:

$$\frac{d}{dt} \left(\frac{\partial W_m^*}{\partial \dot{\mathbf{y}}} \right) + \frac{\partial W_e}{\partial \mathbf{y}} + \underbrace{\sum_{r,s} u_r \frac{\partial \dot{q}_s}{\partial \dot{\mathbf{y}}}}_{\mathbf{U}} = \mathbf{0} \quad (14)$$

in terms of the generalized variables $\mathbf{y} = \begin{pmatrix} \mathbf{q}_i \\ \mathbf{q}_c \end{pmatrix}$. \mathbf{q}_i and \mathbf{q}_c respectively represent the charges associated to the inductors and capacitors of the circuit.

The magnetic co-energy W_m^* is defined by

$$W_m^*(\mathbf{i}_l) = \int \varphi_l^T d\mathbf{i}_l = \frac{1}{2} \dot{\mathbf{i}}_l^T \mathbf{L} \dot{\mathbf{i}}_l + \dot{\mathbf{i}}_l^T \varphi_0 \quad (\text{linear case}) \quad (15)$$

The electrical energy is defined by

$$W_e(\mathbf{q}_c) = \int \mathbf{u}_c^T d\mathbf{q}_c = \frac{1}{2} \mathbf{q}_c^T \mathbf{C}^{-1} \mathbf{q}_c \quad (\text{linear case}) \quad (16)$$

In these expressions, $\varphi_l = \int \mathbf{u}_l dt$, $\mathbf{i}_l = \dot{\mathbf{q}}_l$ and \mathbf{u}_c respectively represent the fluxes through the inductors, the currents through the inductors and the voltage drops across the capacitors. In the magnetic co-energy expression, φ_0 represents the fluxes generated by permanent magnets. The generalized voltages \mathbf{U} can be expressed in terms of the generalized variables and their derivative, using the Kirchoff current laws and the constitutive equations for resistor and sources.

Similar to mechanics, Lagrange multipliers can be used for constraint considerations, see e.g. [7, 9]. Constraints comes from Kirchoff's equations and are algebraic relations involving generalized variables and source values. Two types of constraints are considered:

- Loops of capacitors and voltage sources

$$\mathbf{K}_{1c}^T \mathbf{u}_c + \mathbf{K}_{1u}^T \mathbf{u}_{su} = \mathbf{K}_{1c}^T \mathbf{C}^{-1} \mathbf{q}_c + \mathbf{K}_{1u}^T \mathbf{u}(t) = 0 \quad (17)$$

- Cutsets of inductors, capacitors and current sources

$$\mathbf{K}_{2i}^T \dot{\mathbf{i}}_l + \mathbf{K}_{2c}^T \dot{\mathbf{q}}_c + \mathbf{K}_{2s}^T \dot{\mathbf{i}}_{si} = \mathbf{K}_{2i}^T \dot{\mathbf{q}}_l + \mathbf{K}_{2c}^T \dot{\mathbf{q}}_c + \mathbf{K}_{2s}^T \dot{\mathbf{i}}(t) = 0 \quad (18)$$

The second type of constraints is fully integrable. This ensure that the system is holonomic. Integrating the second type of constraints, we get all the constraints in terms of the generalized variables as:

$$\mathbf{J}_c \mathbf{y} = \mathbf{f}(t) \quad (19)$$

where \mathbf{J}_c is the Jacobian matrix of the constraints and $\mathbf{f}(t)$ represents the time dependent term due to the sources and the integration constants. The latter vanish if starting with zero charges as initial condition.

Using a Virtual Work approach similar to the one used in mechanics, it can be shown [9] that the "electrical motion equations" (14) becomes:

$$\frac{d}{dt} \left(\frac{\partial W_m^*}{\partial \dot{\mathbf{y}}} \right) + \frac{\partial W_e}{\partial \mathbf{y}} + \mathbf{U} = \mathbf{J}_e^T \boldsymbol{\lambda}_e \quad (20)$$

A coordinate partitioning approach similar to that used in mechanical system may also be used [35] to eliminate the Lagrange multipliers and obtain a reduced set of differential equations for the electrical model.

Commonly, it is more convenient to write the electrical equations in terms of the current through the inductors \mathbf{i}_l and the voltage drop across capacitors \mathbf{u}_c . Doing this, the differential equations will always be first-order ODEs. Appendix A shows how equations (14) and (20) can be rewritten in terms of those variables when linear constitutive equations are assumed.

ELECTRAN [35] is the electrical counterpart of ROBOTRAN[®] and has been recently developed using similar concepts and the same principle for symbolically generating the minimal set of equations in a very compact form.

2.2.4 Virtual Work and Electromechanical Systems

It is possible to formulate the Principle of Virtual Work for electromechanical systems by simply adding (3) and (12):

$$\sum_{i=1}^N \left(\delta \mathbf{r}_i \cdot (m_i \ddot{\mathbf{r}}_i - \mathbf{F}_i) + \delta \psi_i \cdot (\dot{\mathbf{H}}_i - \mathbf{L}_i) \right) + \mathbf{u}^T \delta \mathbf{q} = 0 \quad (21)$$

The electromechanical generalized variables can be chosen as:

$$\mathbf{s} = \begin{pmatrix} \mathbf{q}_m \\ \mathbf{y} \end{pmatrix} \quad (22)$$

and equation (21) can be written as:

$$\boldsymbol{\Phi}^T \cdot \delta \mathbf{s} = 0 \quad (23)$$

The coupling is expressed as supplementary forces/torque in the mechanical equations, coming from the dependance of the magnetic co-energy and potential energy with respect to mechanical variables. For instance, when a joint is driven by an electrical actuator, the generic expression for the joint force is given as:

$$Q_{em} = - \frac{\partial W_m^*}{\partial q_{act}} + \frac{\partial W_e}{\partial q_{act}} = \frac{1}{2} \mathbf{y}^T \frac{\partial \mathbf{M}}{\partial q_{act}} \mathbf{y} + \mathbf{l}_l^T \frac{\partial \boldsymbol{\varphi}_c}{\partial q_{act}} + \frac{\partial W_{mag0}}{\partial q_{act}} \quad (24)$$

where q_{act} corresponds to the actuated joint. The magnetic and electrical effects will usually be completely independent and most of the time, the torque will only come from one of these two effects.

No supplementary terms appear in the electrical equations but the electrical mass matrix \mathbf{M}_e is influenced by the mechanical variables and its derivatives.

Combined uses of ROBOTRAN[®] and ELECTRAN easily permits the modelling of complex electromechanical multibody systems, as shown with the applications discussed in Section 3.

2.2.5 Example

Looking at the mechanical subsystem of the example in Figure 1, the kinematic energy T and the potential energy V are:

$$\begin{aligned} T &= \frac{m_5 \dot{x}^2}{2} \\ V &= \frac{k_6 (r_0 + x - l_6)^2}{2} \end{aligned} \quad (25)$$

Assuming that the spring is unstretched when $x = 0$, the potential energy becomes:

$$V = \frac{k_6 x^2}{2} \quad (26)$$

The mechanical generalized forces are given by:

$$Q_m = d_6 \dot{x} + m_5 g \quad (27)$$

Looking at the electrical subsystem, the magnetic co-energy W_m^* and the electrical energy W_e are:

$$\begin{aligned} W_m^* &= \frac{L_3 \dot{q}_L}{2} \\ W_e &= \frac{q_C^2}{2C_2(x)} \end{aligned} \quad (28)$$

The electrical generalized voltages are given by:

$$Q_e = -u_{R1} + E_4(t) = -R_1 \dot{i}_L + E_4(t) \quad (29)$$

We have a constraint between the electrical variables because of the serial connection of the inductor and the capacitor:

$$\dot{q}_L = \dot{q}_C \quad (30)$$

Combining all these terms and considering the generalized variables $s = \begin{pmatrix} x \\ q_L \\ q_C \end{pmatrix}$ the equations of motion are:

$$\begin{cases} \frac{d}{dt}(m_5 \dot{x}) + k_6 x - \frac{q_C^2}{2C_2^2(x)} \frac{dC_2(x)}{dx} = -d_6 \dot{x} - m_5 g \\ \frac{d}{dt}(L_3 \dot{i}_L) = -R_1 \dot{i}_L + E_4(t) + \lambda \\ \frac{q_C}{C_2} = 0 - \lambda \\ \dot{q}_L = \dot{q}_C \end{cases} \quad (31)$$

$$\Leftrightarrow \begin{cases} \frac{d}{dt}(m_5 \dot{x}) + k_6 x - \frac{q_C^2}{2C_2^2(x)} \frac{dC_2(x)}{dx} = -d_6 \dot{x} - m_5 g \\ \frac{d}{dt}(L_3 \dot{i}_L) + \frac{q_C}{C_2} = -R_1 \dot{i}_L + E_4(t) \\ \dot{q}_L = \dot{q}_C \end{cases} \quad (32)$$

One can get the final equations of motion for the global system as:

$$\begin{cases} \frac{d}{dt}(m_5 \dot{x}) + k_6 x + d_6 \dot{x} - \frac{1}{2} \frac{dC_2(x)}{dx} \cdot (E_4(t) - R_1 \cdot i_L - \frac{d}{dt}(L_3 \dot{i}_L))^2 = -m_5 g \\ \dot{i}_L = \frac{dC_2}{dx} \frac{d}{dt} \left(-\frac{d}{dt}(L_3 \dot{i}_L) - R_1 \dot{i}_L + E_4(t) \right) + C_2(x) \frac{d}{dt} \left(-\frac{d}{dt}(L_3 \dot{i}_L) - R_1 \dot{i}_L + E_4(t) \right) \end{cases} \quad (33)$$

2.3 Linear Graph Theory

2.3.1 Basics

Linear graph theory is a branch of mathematics devoted to the study of systems topology. It was invented by Leonhard Euler in the 1700s to study problems of connectivity [10], and was extended in the 1900s [11] to the mathematical modelling of physical systems. In this extension, linear graph theory is combined with the characteristics of physical components to obtain a unified systems theory; the term ‘‘graph-theoretic modelling’’ (GTM) is often used to denote this systems theory. In a nutshell, a system model is obtained by combining topological relationships from linear graph theory with the constitutive equations for individual components. This systems theory is very methodical and well-suited to computer implementation.

To model a physical system, individual components are identified and their constitutive equations are determined. In general, these constitutive relationships are obtained from experimental measurements of the component’s ‘‘through’’ and ‘‘across’’ variables; through variables are measured by an instrument in series with the component, while across variables are obtained from an instrument in parallel. For electrical systems, the through and across variables are current and voltage, respectively. For mechanical systems, force and displacement (or its derivatives) play the role of through and across variables respectively. Note that through and across variables may be tensors of any order, including scalars and vectors.

Once the constitutive equations are determined, the component models are combined in the topology defined by the structure of the physical system. A linear graph, consisting of lines (edges) and circles (nodes or vertices), is used to represent the system topology. The edges represent the individual components, whereas nodes represent the points of their interconnection. From this graph, linear topological equations

are systematically obtained in terms of the through and across variables for all components. The system model is simply the combination of these topological equations with the individual constitutive equations.

To illustrate this, refer to Figure 2 for the linear graph representation of the condensator speaker shown in Figure 1. Edges R_1, C_2, L_3 and E_4 represent the resistor, capacitor, inductor, and voltage source, respectively. Note that the linear graph bears a striking resemblance to the physical system, which is an advantage when it comes to modelling using this approach. Directions are assigned to each edge to establish a positive convention for measuring the through and across variables, similar to setting a polarity on a measuring instrument. The constitutive equations for electrical components are expressed in terms of the scalar variables, current (i) and voltage (v). For the purpose of this example, we assume standard linear relationships for these components, e.g. $v_1 = R_1 i_1$ and $v_3 = L_3 \frac{di_3}{dt}$.

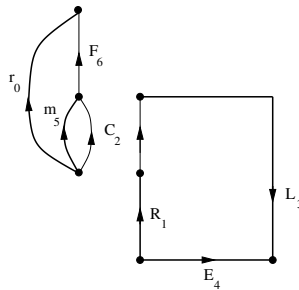


Figure 2: Linear Graph of the condensator speaker

Also shown in Figure 2 is the linear graph of the mechanical part of the condensator speaker. The edge m_5 represents the inertia and weight of the moving mass; the edge begins at a ground-fixed (inertial) reference node and terminates at the center of mass. Its constitutive equation is given by the combination of gravity with the d'Alembert form of Newton's Second Law: $\mathbf{F}_5 = -m_5 \ddot{\mathbf{x}}_5 - m_5 \mathbf{g}$, where the vector force \mathbf{F}_5 depends on gravity \mathbf{g} , the vector acceleration $\ddot{\mathbf{x}}_5 = \ddot{x} \hat{\mathbf{i}}$, and the upwards unit vector $\hat{\mathbf{i}}$ (parallel to x). The edge F_6 represents the combined effects of the spring and damper components (these could easily be split into separate edges for the spring and damper, if desired). Its constitutive equation is $\mathbf{F}_6 = -k_6(r_6 - l_6)\hat{\mathbf{r}}_6 - d_6(\mathbf{v}_6 \cdot \hat{\mathbf{r}}_6)\hat{\mathbf{r}}_6$, where l_6 is the undeformed spring length, k_6 and d_6 are the stiffness and damping coefficients, \mathbf{v}_6 is the relative velocity of the endpoints, and $\hat{\mathbf{r}}_6 = \mathbf{r}_6 / |\mathbf{r}_6|$ is the unit vector parallel to the component. Finally, the edge r_0 locates the point where the spring-damper is attached to the ground: $\mathbf{r}_0 = l_0 \hat{\mathbf{i}}$.

For each of the two parts of the linear graph, mechanical and electrical, we can generate sets of topological equations that relate the through and across variables. This can be done manually by inspection of the graph, or by applying matrix operations to an "incidence matrix" that encapsulates the topology of the physical system. For a linear graph with e edges and v vertices, entry I_{jk} of the $e \times v$ incidence matrix \mathbf{I} is $[0, +1, \text{ or } -1]$ if edge k is [not incident upon, incident and away from, or incident and towards the vertex v].

The Vertex Postulate [11] then allows us to write:

$$\mathbf{I}\tau = 0 \quad (34)$$

where τ is a column matrix of all the through variables. For electrical systems, the Vertex Postulate corresponds to Kirchoff's Current law at every node. For mechanical systems, the Vertex Postulate gives v equations for dynamic equilibrium. Starting from the Vertex Postulate, two very useful sets of topological

equations, the “cutset” and “circuit” equations, can be systematically derived by selecting a tree and applying elementary matrix operations to \mathbf{I} . For electrical networks, the circuit equations correspond to Kirchoff’s Voltage law around a closed circuit, while the cutset equations are linear combinations of the vertex equations for all the nodes in a given subgraph.

A tree is a set of $v - 1$ edges (“branches”) that connects all of the vertices but does not contain any closed loops. A very attractive feature of linear graph theory is that *by selecting a tree, one can control the primary variables appearing in the final system*: they are the across variables α_b for branch elements, and the through variables τ_c for cotree elements (“chords”). This is accomplished by re-writing the cutset equations as the chord transformations $\tau_b = -\mathbf{A}_c \tau_c$, where τ_b are the branch through variables and \mathbf{A}_c is obtained from elementary row operations on \mathbf{I} , and by re-writing the circuit equations as the branch transformations $\alpha_c = -\mathbf{B}_b \alpha_b$, where α_c are the cotree across variables. The Principle of Orthogonality, which represents a very generalized energy conservation principle, guarantees that $\mathbf{B}_b = -\mathbf{A}_c^T$.

By selecting edges R_1, L_3 , and E_4 into the tree for the electrical sub-graph in Figure 2, one gets the chord transformations:

$$\begin{Bmatrix} i_1 \\ i_3 \\ i_4 \end{Bmatrix} = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} i_2 \quad (35)$$

and the single branch transformation:

$$v_2 = - \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}^T \begin{Bmatrix} v_1 \\ v_3 \\ v_4 \end{Bmatrix} \quad (36)$$

Assuming that there is one constitutive equation for each of the v elements, substituting the branch and chord transformations into these constitutive equations will result in v system equations in terms of the v primary variables.

2.3.2 Linear Graphs and Multibody Systems

The same basic concepts apply when one models a multidimensional mechanical (“multibody”) system using linear graph theory: the system model is obtained by combining the constitutive equations for individual components with the linear cutset and circuit equations resulting from their connectivity. Again, the selection of a tree determines the primary variables appearing in the system equations. The cutset and circuit equations retain a simple form because linear graph theory allows the use of vector modelling variables. However, the constitutive equations for some components will be nonlinear due to the finite rotations of bodies in the system. Furthermore, the physical interpretation of nodes and edges must be generalized.

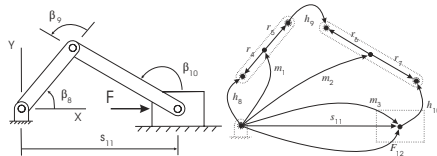


Figure 3: Slider-crank mechanism and the corresponding linear graph

To illustrate, consider the planar slider-crank mechanism shown in Figure 3, along with its linear graph representation.

Each node in the linear graph represents the position and orientation of a body-fixed reference frame, while the edges represent transformations between frames corresponding to physical components. For each

element, there are now two sets of through and across variables: translational and rotational. Thus, there will be two sets of cutset and circuit equations, since these variables cannot be added together. Although the incidence matrix is the same for each, selecting different trees can be used to create different cutset and circuit equations for translation and rotation. This can be used to reduce the system equations to a set that is smaller in number than those generated by conventional multibody formalisms [12].

In figure 3, the edges m_1 , m_2 , and m_3 represent both the translational and rotational inertia of the three rigid bodies (crank, connecting rod, and slider). These bodies are connected by revolute joints h_8 , h_9 , and h_{10} and by the prismatic joint s_{11} . The “rigid arm” elements $r_4 - r_7$ define the position and orientation, relative to the center of mass frames on the bodies, of the body-fixed frames that define the connection points of these joints. Finally, the external force on the slider is modelled by the force element F_{12} , originating at the inertial frame (node) and terminating at the slider.

The constitutive equations for the multi-dimensional translation of rigid bodies and spring-dampers are the same as that shown in the previous section. However, a second equation relates the d’Alembert torque on the body to its rotational inertia. This equation corresponds to Euler’s equations for rotational motion.

For the rigid-arm elements, e.g. r_4 , the tip node does not rotate relative to the tail (center of mass) node; hence, the angular velocity ω_4 is zero. However, the translational velocity of the rigid-arm is a nonlinear function of the angular velocity of the body on which it resides, e.g. $\mathbf{v}_4 = \omega_1 \times \mathbf{r}_4$, which is a well-known result from rigid body kinematics. For the ideal joints, one always finds that the motion allowed by a joint, e.g. $\mathbf{r}_{11} = s_{11} \hat{\mathbf{i}}$ where \mathbf{r}_{11} is the translational displacement of the slider along X , is orthogonal to the reaction forces and torques that arise in the joint, e.g. $\mathbf{F}_{11} = F_{11} \hat{\mathbf{j}}$ and $\mathbf{T}_{11} = T_{11} \hat{\mathbf{k}}$ where $\hat{\mathbf{j}}$ and $\hat{\mathbf{k}}$ are unit vectors parallel to Y and Z directions, respectively. This is a result of the fact that ideal joints do no work, and can be used to eliminate joint reactions in the system dynamic equations.

Fortunately, the topological equations remain linear regardless of the nonlinearities in the constitutive equations. Furthermore, the selection of trees can again be used to define the primary variables \mathbf{q} and λ in the final system equations. The “branch coordinates” \mathbf{q} are the unknown across variables for elements (branches) in the tree. Linear graph theory provides a unification of traditional absolute and relative (or joint) coordinate formulations.

By selecting components with known across variables (e.g. $r_4 - r_7$) into the tree, the number n of branch coordinates (and system equations) is reduced. If the tree is completed by $m_1 - m_3$, then the final equations are in terms of the absolute coordinates for the three bodies. If joints $h_8 - h_{10}$ are selected in place of $m_1 - m_3$, then one obtains equations in the joint coordinates $\beta_8 - \beta_{10}$. Thus, linear graph theory provides a unification of traditional absolute and joint coordinate formulations.

Any joints left in the cotree will provide the reaction loads appearing in the Lagrange multipliers λ_m . Furthermore, these cotree joints will also provide one kinematic constraint equation for each reaction load. These p constraint equations express the relationships between the branch coordinates, which will not be independent if there are joints in the cotree. This is always the case for a system with closed kinematic chains. These constraint equations are always found by projecting the circuit equations for the cotree joints onto their reaction spaces.

For the slider-crank example, the three joint coordinates $\beta_8 - \beta_{10}$ must be related by two constraint equations because the system has only 1 degree of freedom (dof). These constraint equations are always found by projecting the circuit equations for the cotree joints onto their reaction spaces.

To illustrate, the translational and rotational circuit equations for cotree joint s_{11} are:

$$\mathbf{r}_8 - \mathbf{r}_4 + \mathbf{r}_5 + \mathbf{r}_9 - \mathbf{r}_6 + \mathbf{r}_7 - \mathbf{r}_{10} - \mathbf{r}_{11} = 0 \quad (37)$$

$$\beta_8 - \beta_4 + \beta_5 + \beta_9 - \beta_6 + \beta_7 - \beta_{10} - \beta_{11} = 0 \quad (38)$$

where $\mathbf{r}_8 = \mathbf{r}_9 = \mathbf{r}_{10} = 0$ from the constitutive equations for revolute joints, and all terms in the rotational equation are constant (again, from the constitutive equations) except for the three branch coordinates $\beta_8 - \beta_{10}$. Note that the circuit equations represent the zero summation of displacement vectors around a closed kinematic chain. The reaction space for s_{11} is spanned by unit vectors $\hat{\mathbf{j}}$ and $\hat{\mathbf{k}}$, as mentioned

previously. Projecting the circuit equations onto these two unit vectors, and substituting all constitutive equations, results in the kinematic constraint equations:

$$\mathbf{h}(\mathbf{q}) = 0 \quad (39)$$

where the column matrix \mathbf{h} is a nonlinear algebraic function of $\mathbf{q} = \begin{pmatrix} \beta_8 \\ \beta_9 \\ \beta_{10} \end{pmatrix}$.

To obtain the dynamic equations of the system, the cutset equations for each branch are projected onto the motion space for that branch. For example, for a revolute joint, the motion space is defined by the vector parallel to the joint axis, while the reaction space is the plane orthogonal to that same vector.

As an example, the motion space for branch joint h_9 is spanned by $\hat{\mathbf{k}}$, the unit vector parallel to the joint axis. The rotational cutset equation for h_9 is:

$$\mathbf{T}_9 + \mathbf{T}_2 + \mathbf{T}_3 + \mathbf{T}_{11} + \mathbf{T}_{12} = 0 \quad (40)$$

which represents the rotational dynamic equilibrium for bodies m_2 and m_3 that are isolated by the cutset. Projecting this equation onto $\hat{\mathbf{k}}$, and using the constitutive equations $\mathbf{T}_9 = \mathbf{T}_{12} = 0$, one obtains a dynamic equation in terms of the inertia of m_2 and m_3 , and the cotree joint reaction torque \mathbf{T}_{11} . Generating the two dynamic equations for the other two joints in the tree, and assembling in matrix form, gives the system dynamic equations:

$$\mathbf{M}_m \ddot{\mathbf{q}} + \mathbf{Q}_m = \mathbf{J}_m^T \boldsymbol{\lambda}_m \quad (41)$$

where \mathbf{M}_m is the $n \times n$ mass matrix, \mathbf{J}_m is the Jacobian matrix of the kinematic constraint equations, and \mathbf{Q}_m contains external loads and quadratic velocity terms. Together, the kinematic and dynamic equations constitute a set of $n + p$ nonlinear differential-algebraic equations (DAEs), the solution of which will give the branch coordinates $\mathbf{q}(t)$ and Lagrange multipliers $\boldsymbol{\lambda}_m(t)$.

Note that one can also generate the dynamic equations by combining linear graph theory with analytical mechanics, e.g. the Principle of Virtual Work. This approach is very useful for incorporating flexible bodies into the multibody model [13]. It has been implemented using the Maple symbolic programming language into a multibody dynamics program called DynaFlex⁴.

2.3.3 Linear Graph and Electrical Systems

As mentioned previously, the basic principles of modelling electrical networks using linear graph theory were established decades ago by Koenig et al [11], Roe [14], and others. Once the constitutive equations for each element are supplemented with the cutset and circuit equations, one has a necessary and sufficient set of v equations to obtain the v primary variables.

However, it is possible to reduce the equations to a smaller set by exploiting the nature of the constitutive equations. One approach is to generate one equation for each capacitor and inductor, and to use the remaining constitutive equations and branch/chord transformations to express all other variables in terms of the capacitor voltages and inductor currents. This approach was successfully implemented by Muegge [15]. For the electrical portion of the linear graph shown in Figure 2, one would get two first-order ordinary differential equations (ODEs) in terms of v_2 and i_3 .

Another approach is to express all variables in terms of the currents associated with chords, or the voltages associated with branches. The former is called the current formulation, while the latter is named the voltage formulation; both were implemented in Maple by Scherrer and McPhee [16]. By selecting the tree appropriately, one can significantly reduce the final number of system equations.

For the example shown in Figure 2 with the capacitor C_2 selected into the cotree, the current formulation will give a single second-order ODE in terms of the corresponding current i_2 .

⁴<http://real.uwaterloo.ca/~dynaflex/>

This is accomplished by substituting the chord transformations in (35) into the constitutive equations for the branches, giving:

$$\begin{aligned} v_1 &= R_1 i_2 \\ v_3 &= L_3 \frac{di_2}{dt} \\ v_4 &= E_4(t) \end{aligned}$$

where $E_4(t)$ is the prescribed voltage source. Substituting these constitutive equations into the branch transformation (36) gives:

$$v_2 = -R_1 i_2 - L_3 \frac{di_2}{dt} + E_4(t) \quad (42)$$

which expresses the capacitor voltage in terms of its current. This equation will be combined with the constitutive equation for the moving-plate capacitor, defined in the next section, to obtain the final system equation.

2.3.4 Linear Graph and Electromechanical Systems

An electromechanical system consists of electrical networks and multibody systems that are coupled by electromechanical transducers. The moving-plate capacitor shown in Figure 1 is an example of such a transducer. The electrical characteristics of this capacitor depend upon the mechanical separation x between the plates:

$$i_2 = C(x) \frac{dv_2}{dt} + \frac{dC(x)}{dx} \frac{dx}{dt} v_2 \quad (43)$$

where the capacitance C is a function of x , and the second term is the motion-induced current in the component. Due to the electrical attraction of the plates, a force arises that depends upon the voltage across the capacitor:

$$F_2 = \frac{1}{2} \frac{dC(x)}{dx} v_2^2 \quad (44)$$

Thus, the moving-plate capacitor is characterized by two constitutive equations, one associated with the electrical domain and the other with the mechanical domain. This is a common characteristic of transducer elements, which transform electrical energy into mechanical energy, and vice-versa.

Since the capacitor affects the physics of both domains, there is an edge for the capacitor in the linear graphs for the electrical and mechanical subsystems shown in Figure 2. Equation (43) is associated with the electrical edge, while the mechanical edge is characterized by equation (44). It is through these constitutive equations that the two domains are coupled.

A number of transducer models, including those for moving-plate capacitors, moving-coil inductors, DC motors, and induction motors, have been incorporated into the DynaFlex symbolic modelling package, along with the previous current and voltage formulations for the electrical sub-systems. From a single, unified, linear graph representation of the electromechanical system, the governing equations are automatically generated in symbolic form. As a result of the coupling due to transducer constitutive equations, electrical terms will appear in the mechanical equations and vice-versa. Using the branch and chord transformations, all equations can be expressed in terms of the primary variables, which are determined by the user through the tree selection.

2.3.5 Example

DynaFlex was used to generate the governing equations for the condenser speaker. As described previously, selecting the capacitor into the cotree of the linear graph and using the current formulation gives the capacitor voltage shown in equation (42). Substituting this equation into the electrical constitutive equation (43), the single ODE for the electrical domain is obtained:

$$\frac{dC_2}{dx} \frac{dx}{dt} \left(-R_1 i_2 - L_3 \frac{di_2}{dt} + E_4(t) \right) + C_2(x) \frac{d}{dt} \left(-R_1 i_2 - L_3 \frac{di_2}{dt} + E_4(t) \right) - i_2 = 0 \quad (45)$$

where the primary variables are the cotree current i_2 and the mechanical displacement x .

For the mechanical domain (see Figure ??), the body-fixed vector r_0 and mass m_5 are selected into the tree, resulting in the single branch coordinate $r_5 = x$. This is an independent coordinate for the 1-dof system, so no constraint equations are generated (there are no joints in the cotree). The single dynamic equation results from the outset equation for the mass, projected onto its motion space defined by $\hat{\mathbf{t}}$:

$$F_5 + F_2 - F_6 = 0 \quad (46)$$

Substituting the mechanical constitutive equations into this expression, and re-arranging,

$$-m_5\ddot{x} - m_5g + \frac{1}{2}\frac{dC_2(x)}{dx}v_2^2 + d_6\dot{x}_6 + k_6(x_6 - l_6) = 0 \quad (47)$$

Assuming that the spring is unstretched at $x = 0$, which implies that $l_6 = r_0$, one gets the branch transformations:

$$\begin{aligned} x_2 &= x \\ x_6 &= r_0 - x \end{aligned}$$

which shows that the spring-damper shortens as x increases. Substituting these branch transformations and the capacitor voltage (42) into equation (47), one gets the single ODE for the mechanical domain:

$$m_5\ddot{x} + d_6\dot{x} + k_6x - \frac{1}{2}\frac{dC_2}{dx}\left(-R_1i_2 - L_3\frac{di_2}{dt} + E_4(t)\right)^2 = 0 \quad (48)$$

Together, equations (45) and (48) can be solved for the primary variables $i_2(t)$ and $x(t)$. Equations (45) and (48) are equivalent to equations (33) obtained with the mathematical approach.

2.4 Bond Graphs

2.4.1 Basics

Bond Graphs were invented by H.M. Paynter in 1959 and published by its inventor in 1961 [17]. One of the first books presenting the language was written by Karnopp and Rosenberg in 1968. It has been reedited in the year 2000 [18]. Like Linear Graphs, Bond Graphs are based on analogies between different physical domains. In every field of physics, energy sources, energy storage elements and dissipative elements can be found. Bond graphs consider a system as isolated from its environment such that exchanges of energy with its environment only occur through interacting ports. For characterizing the power exchanges at the different ports, power variables are defined at each port: the effort (e) and the flow (f) variables. The product of these two variables is the power flowing through the port. Two other energy variables are also considered in bond graphs: the momentum $p = \int e \cdot dt$ and the displacement $q = \int f \cdot dt$. A sign convention for positive power is indicated on the bond graph by a half arrow, as shown in Table 1. A positive power is given to the port pointed at by the bond. The basic components considered by Bond Graph theorists are given in Table 1. Transducers such as transformers and gyrators are used for converting the variables from one energy domain to another. With the exception of sources and resistors, all elements are power conservative.

Each component is characterized by a particular constitutive equations relating the effort and flow variables and/or their derivatives. Using these equations requires the specification of the sequence and direction in which the equations must be solved. This is done by assigning causality to the bonds of the graph. Causality is indicated by a little bar crossing the bond at one of its end: this bar indicates that the effort is imposed to the element connected at that end of the bond. Causality assignment is restricted by the nature of the components associated with the bond, making causality analysis an important step while analyzing the system. Causality analysis helps detecting algebraic loops and constraints between state-variables, similar to the tree selection procedure in Linear Graph Theory.

Table 1 indicates the basic elements with sign convention and causality assignments, see [18] for details.

Basic 1-Port Elements		
Flow Source		$e = e(t)$
Effort Source		$f = f(t)$
Resistor		$e = \phi_R(f)$
		$f = \phi_R^{-1}(e)$
Capacitor		$q = \phi_C(e)$
		$e = \phi_C^{-1}(q)$
Inertia		$f = \phi_I(p)$
		$p = \phi_I(f)$
Basic 2-Port Elements		
Transformer		$e_1 = m \cdot e_2$ $f_1 = m \cdot f_2$
Modulated transformer		$e_1 = m(\theta) \cdot e_2$ $f_1 = m(\theta) \cdot f_2$
Gyrator		$e_1 = r \cdot f_2$ $e_2 = r \cdot f_1$
Modulated gyrator		$e_1 = r(\theta) \cdot f_2$ $e_2 = r(\theta) \cdot f_1$
Junctions		
0-junction		$e_1 = e_2 = e_3$ $f_1 = f_2 - f_3$
1-junction		$f_1 = f_2 = f_3$ $e_1 = e_2 - e_3$

Table 1: Basic bond graph elements and their constitutive equations

C- and I-fields give an extension of the C- and I-elements and are very useful for modelling more complex systems, like multidimensional mechanical systems or electromagnetic converters. C-fields are multiport elements characterized by the following constitutive matrix equation: $\mathbf{e} = \mathbf{e}(\mathbf{q}) = \mathbf{K} \cdot \mathbf{q}$ (in the linear case), where \mathbf{K} is a square symmetric matrix. Similarly, an I-field's constitutive equation is: $\mathbf{f} = \mathbf{f}(\mathbf{p}) = \mathbf{K} \cdot \mathbf{p}$ (in the linear case). Mixed IC-fields are also used.

2.4.2 Bond Graph and Multibody Systems

A well-furnished history on the use of bond graphs in mechanics and multibody systems can be found in Wilfrid Favre's thesis [19]. Let us point out some of the most interesting facts.

A first statement can already be made about the choice of variables: all the methods developed on the basis of bond graphs use absolute coordinates, which is not necessarily the best choice in terms of compactness of the equations (and thus in terms of computational efficiency).

Concerning the methods for deriving the bond graph of a multibody system, Karnopp and Rosenberg [21, 18, 22] first proposed an analytical approach based on the writing of the kinematic laws, the drawing of the corresponding junction structure (using MTF's, 0- and 1-junctions) and the addition of 1-ports elements. The orthogonality principle guarantees that the dynamic equations can be derived from the obtained graph.

Kinematic constraints due to loops of bodies and ideal joints between the bodies is an important issue in multibody dynamics. These constraints lead to non-linear algebraic relations along with the dynamical equations. This kind of differential algebraic set of equations (DAE's) require specific integration algorithms. In order to avoid these non-linear algebraic relations, Karnopp introduces compliant elements (springs and dampers) in the connections, at the cost of bringing high frequencies into the system [23]. Other methods based on the Lagrange equations were developed by Karnopp to eliminate constraints [24]. Actually, the constraints are included manually in the equations before drawing the corresponding graph. Other methods based on analytical computations were developed by Allen and Dubowsky [25] or Brown [26]. All these methods use absolute coordinates and require the writing of at least the kinematic equations before drawing of the bond graph. It is thus very tedious to derive a systematic procedure for modelling 3D mechanical multibody systems.

All the previous methods are based on one-dimensional bond graphs also called scalar bond graphs. Different propositions [27, 28] have been made to represent multidimensional system using vector bond graphs or multibond graphs. Breedveld established a standard representation for multibond graphs [29], [30]. Using multibond graph leads to much easier and a much more systematic method for drawing the graph, almost without writing any equations [31, 19]. The structure of the graph is simpler but still very different from the physical structure of the system. Most of these methods also get into trouble as the equations still have to be written down. As with scalar bond graphs, vectorial derivative causality due to the constraints generates problems in deriving the equations. Favre [19] has proposed an approach in which he comes back to a scalar bond graph after drawing the vector bond graph. Other approaches generalizing the Karnopp one-dimensional approaches (compliant elements, use of IC-fields for Lagrange equations,...) may also be developed for avoiding derivative causality but are not systematic. Transformation of the dependent I elements through transformers is also proposed in [31]. Some work for defining bond graph blocks describing basic multibody components has been done and leads to a more systematic definition of the bond graph [32].

Another important statement regarding bond graphs is that they essentially manipulate energy variables, velocities and forces in mechanics. This means that no position can be introduced directly into the graph. Because of that, the kinematic constraints can only be derived at a velocity level. This forces the modeller to be particularly careful when choosing the initial guess (closed configuration) and also during the integration process, to make sure that the constraints are satisfied at the position level, which is a very delicate problem in multibody dynamics. Of course, positions can be extracted from the graph and used as signals through block diagrams, which is of absolute necessity since most of the transformers in the graph are position-modulated as will be illustrated later on. Modelling elements like springs are also position-modulated, but this is out of the bond graph representation.

Bond graphs have the main advantage of being acausal, meaning that the graph structure does not change because of a change in causality. Different causality assignments of the graph lead to different sets of equations depending on the required analysis. For example, the same graph may be used for obtaining a direct or inverse model.

2.4.3 Bond Graph and Electrical Systems

Electrical networks, being one-dimensional by definition, are much more suited to the bond graph approach. Modelling electrical circuits or networks is straightforward using bond graphs [18]. The main disadvantage is, once again, that bond graphs, unlike linear graphs, have a structure different from that of the circuit, making the graph hard to understand.

Another challenge facing Bond Graph theorists is the presence of constraints between state-variables. Usually, the state variables are chosen as the flow (the current) in the inertia (inductor) elements and the effort (the voltage) across the capacitors. When cutsets of inductors or loops of capacitors can be found in a circuit, constraints exist and the state variables are not independent. Causality analysis on the graph will indicate this and will allow for selecting a minimal set of independent variables [18]. Another solution consists in using some transformations to reduce the set of state variables to a set of independent state variables. For example, the Park transformation [33] can be used for modelling a star connected three-phase induction motor, where the sum of phase currents must equal zero, by an equivalent two-phase one, where all the phase currents are independent.

2.4.4 Bond Graph and Electromechanical Interaction

Bond graphs consider the same generic elements for every physical domain. This facilitates the modelling of multidomain systems, including electromechanical ones. The electromechanical interaction is generally modelled using the transformer (TF) and gyrator (GY) elements (shown in the DC-motor model in Section 3). When the interactions occur with energy storage elements, C-, I- or mixed IC-fields may also be very useful, as shown in the following example.

The software package 20-Sim⁵ has been used for simulating the bond graph models of the application discussed in Section 3.

2.4.5 Example

The complete bond graph for the electrostatic microphone described previously is given in Figure 4. A C-field is used as a result of the dependence of the capacitor value on the mechanical position: $C_2 = C_2(x)$. The constitutive equations for this C-field are determined as follows.

The energy in the C-field is given by:

$$E(q, x) = \int (\dot{q} \cdot V_c + \dot{x} \cdot F_c) dt = \int V_c \cdot dq + \int F_c \cdot dx \quad (49)$$

It follows that the constitutive equations for the C-field are related to the energy by:

$$V_c = \frac{\partial E(q, x)}{\partial q} \quad \text{and} \quad F_c = \frac{\partial E(q, x)}{\partial x} \quad (50)$$

Experiments on the capacitance tells us that the following relations can be established in the electrical field:

$$V_c = \frac{q}{C_2(x)} \quad (51)$$

Assuming that at a constant x , we charge the capacitor from 0 to q Coulomb, the energy stored in the capacitance will be:

⁵<http://www.20sim.com>

$$E(q, x) = \int_0^q \frac{q}{C_2(x)} dq = \frac{1}{2} \frac{q^2}{C_2(x)} \quad (52)$$

The expression of the electrostatic force acting between the two plates is then obtained as:

$$F_c = -\frac{1}{2} \frac{q^2}{C_2(x)^2} \frac{dC_2}{dx} \quad (53)$$

Equations (51) and (53) are the constitutive equations of the C-field shown in Figure 4.

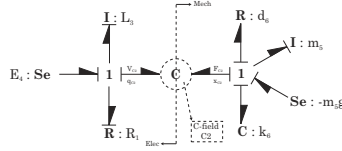


Figure 4: Bond graph of the electrostatic microphone

Deriving the dynamic equations for the system is easily done. We start by first writing down the 1-port equations:

$$\begin{cases} e_{E4} - e_{R1} - e_{L3} - e_{C2}^{Elec} = 0 \\ e_{C2}^{Mech} + e_{d6} + e_{m5} + e_{k6} - e_g = 0 \end{cases} \quad \begin{cases} f_{E4} = f_{R1} = f_{L3} = f_{C2}^{Elec} = i_L \\ f_{C2}^{Mech} = f_{d6} = f_{m5} = f_{k6} = f_g = \dot{x} \end{cases} \quad (54)$$

Using the constitutive equations of the different elements, we can derive the dynamic equations:

$$\begin{cases} m_5 \cdot \ddot{x} = \frac{1}{2} \frac{q^2}{C_2(x)^2} \frac{dC_2}{dx} - k_6 \cdot (r_0 + x - l_6) - d_6 \cdot \dot{x} - m_5 g \\ L_3 \cdot \frac{di_L}{dt} = E_4(t) - R_1 \cdot i_L - V_c \\ i_L = i_C \Leftrightarrow i_L = C_2 \frac{dV_c}{dt} \end{cases} \quad (55)$$

This reduces to

$$\begin{cases} \frac{d}{dt} (m_5 \dot{x}) + k_6 x + d_6 \dot{x} - \frac{1}{2} \frac{dC_2(x)}{dx} \cdot (E_4(t) - R_1 \cdot i_L - \frac{d}{dt} (L_3 i_L))^2 = -m_5 g \\ i_L = \frac{dC_2}{dx} \frac{dV_c}{dt} \left(-\frac{d}{dt} (L_3 i_L) - R_1 i_L + E_4(t) \right) + C_2(x) \frac{d}{dt} \left(-\frac{d}{dt} (L_3 i_L) - R_1 i_L + E_4(t) \right) \end{cases} \quad (56)$$

One more time, equations (56) is identical to equations (33) and (48), previously obtained with Virtual Work and Linear Graph approaches.

3 Examples and discussion

3.1 2-Link Robot Driven By DC-motors

3.1.1 Description of the system

In this first example, a two-link robot manipulator is modelled. It consists of two rigid arms articulated around two horizontal parallel axes. Two DC-motors are used to drive the two articulations. The actuators are controlled through a basic PD controller and are each required to follow a desired trajectory of 90° rotation in 4 seconds. A schematic representation of the system is shown in Figure 5.

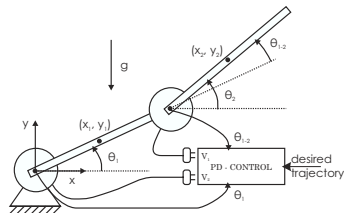
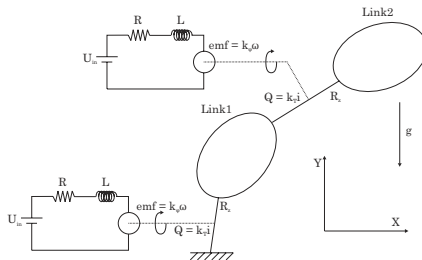


Figure 5: Schematic of the 2-link robot manipulator

3.1.2 Virtual work approach

Following ELECTTRAN/ROBOTRAN[®] conventions, input data files can be written for the DC motor and the multibody system. They are based on the system's representation shown in Figure 6. The DC motor is modelled as a serial connection of a voltage source, a resistor, an inductor and a driven voltage source for the electromotive force ($emf = k_p \omega$). The corresponding torque is imposed to the mechanical subsystem through the term Q_m appearing in equation (6). It is actually introduced as a joint force.

Equations were generated using the method previously described (see Section 2.2) and are expressed in terms of relative joint angles θ_1 and θ_{1-2} , defined in figure 5.

Figure 6: 2-link robot representation for ROBOTRAN[®] and ELECTTRAN

3.1.3 Linear graph approach

A unique linear graph involving electrical and mechanical elements can be drawn for the system. It is shown in Figure 7. The mechanical graph has been obtained by applying a systematic method:

- add one node for each body-fixed reference frame, center of mass, joint connection point, force application point,....;
- connect those nodes using the different types of edges (rigid-arm r , body m , revolute joint h , inertia J).

The electrical graph is directly obtained from the electrical circuit and the transducer elements (M elements) have been added for connecting both subgraphs.

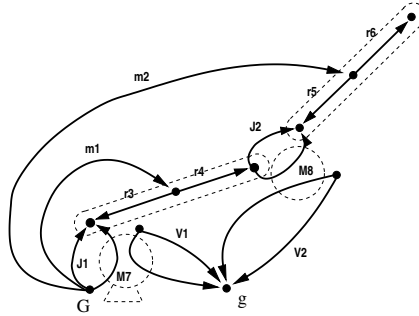


Figure 7: Linear graph of the 2-link robot manipulator

The terminal equations for the transducers are given below.

$$\begin{cases} K_\phi \omega + R i + L \frac{di}{dt} - u = 0 \\ T - b \omega - J \frac{d\omega}{dt} - K_\phi i = 0 \end{cases} \quad (57)$$

Equations were generated using the method previously described (see Section 2.3) and are expressed in terms of relative joint angles θ_1 and θ_{1-2} and the motor currents.

3.1.4 Bond graph approach

One of bond graph's advantages is that it allows the modeller to draw the bond graph for each subsystem and then connect the subsystems. Using that principle, we started by drawing the bond graph of the DC-motor. Figure 8 shows the DC-motor graph. By means of a 1-junction, the different elements are connected in series.

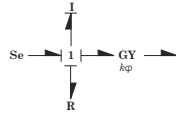


Figure 8: Bond graph of a DC-motor

Concerning the mechanical part, Karnopp's procedure has been used for obtaining the bond graph. First, we define the key variables as:

- State variables : θ_1 and θ_2 .
- Cartesian variables : x_1, x_2, y_1 and y_2 .

Note that all the variables correspond to absolute coordinates.

Then, we manually calculate the velocity transformations:

$$\begin{aligned} x_1 &= d_1 \cdot \cos(\theta_1) & x_2 &= l_1 \cos(\theta_1) + d_2 \cdot \cos(\theta_2) \\ y_1 &= d_1 \cdot \sin(\theta_1) & y_2 &= l_1 \sin(\theta_1) + d_2 \cdot \sin(\theta_2) \end{aligned} \quad (58)$$

$$\begin{aligned} \dot{x}_1 &= -d_1 \cdot \sin(\theta_1) \cdot \dot{\theta}_1 & \dot{x}_2 &= -l_1 \sin(\theta_1) \cdot \dot{\theta}_1 - d_2 \cdot \sin(\theta_2) \cdot \dot{\theta}_2 \\ \dot{y}_1 &= d_1 \cdot \cos(\theta_1) \cdot \dot{\theta}_1 & \dot{y}_2 &= l_1 \cos(\theta_1) \cdot \dot{\theta}_1 + d_2 \cdot \cos(\theta_2) \cdot \dot{\theta}_2 \end{aligned} \quad (59)$$

In these equations, l_1 and l_2 are the lengths of the two arms and d_1 and d_2 specify the position of the center of mass for both arms.

Eventually, we draw the graph and we add the inertia elements and other 1-ports if necessary. We connect the DC-motor graph to get the final bond graph shown in Figure 9. The 0-junction at which the second actuator is connected was necessary for obtaining a relative displacement between the two bodies. Because of the rigid connections between the arms, the translative I-elements would have derivative causalities. Since bond graphs formulation use absolute coordinates, four kinematic constraints, given by equations (59), exist between the six variables, leading to differential algebraic equations (DAEs). This was not the case with the previous approaches, which are able to end up with ordinary differential equations (ODEs) in terms of relative joint angles θ_1 and θ_{1-2} . Compliant elements have been introduced for avoiding DAEs, at the cost of high frequency vibrations.

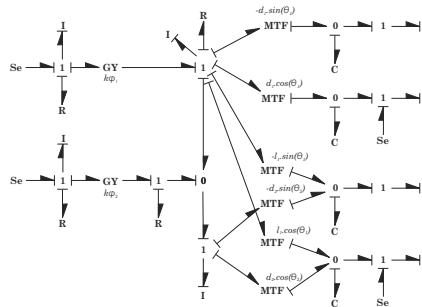


Figure 9: Bond graph of the 2-link robot manipulator

3.1.5 Comparison of the Simulation Results

The results obtained with the different tools are plotted in figures 10, 11 and 12. The obtained results are very similar, especially for the linear graph and equational approach. As expected, high frequencies occur in the results obtained with 20-Sim because of the compliant elements that we introduced. The latter have been introduced to avoid DAEs requiring specific solvers⁶.

3.2 Flexible Barrier Driven By a Three-Phase Induction Actuator

3.2.1 Description of the system

This second system is more complex but is more representative of the kind of systems we want to be able to deal with. It consists of a 3m flexible barrier, used for access control to parking lots. It is driven by an asynchronous three-phase inductive motor through a gearbox and a 6-bar mechanism (2 kinematic loops), as shown on figure 13. A spring is used in the mechanism to counteract the weight of the barrier. The actuator is in a star-star connection configuration with a short-circuited rotor. Mutual inductance effects exist inside

⁶Recent versions of 20-sim can deal with DAEs through a BDF solver

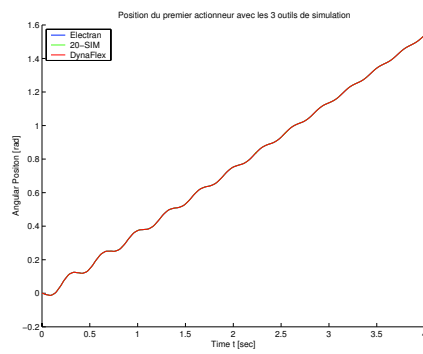


Figure 10: Position of the first actuator shaft

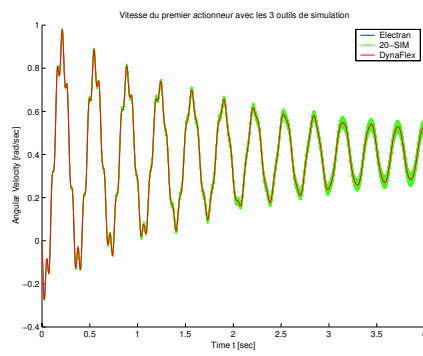


Figure 11: Velocity of the first actuator shaft

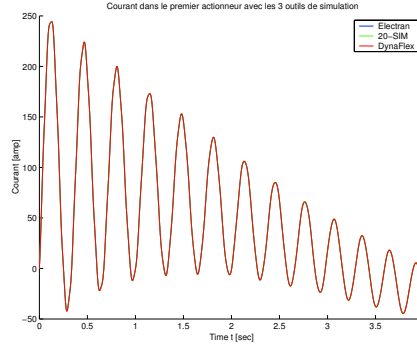


Figure 12: Current flowing into the first actuator

each circuit but also between the stator and rotor components. The position of the rotor influences the magnitude of these mutual effects. The inductance matrices are:

$$\mathbf{L}_s = \begin{pmatrix} L_s & M_s & M_s \\ M_s & L_s & M_s \\ M_s & M_s & L_s \end{pmatrix} \quad \mathbf{L}_r = \begin{pmatrix} L_r & M_r & M_r \\ M_r & L_r & M_r \\ M_r & M_r & L_r \end{pmatrix}$$

$$\mathbf{M}_{sr} = M_{sr} \begin{pmatrix} \cos(\theta_{em}) & \cos(\theta_{em} + \frac{2\pi}{3}) & \cos(\theta_{em} + \frac{4\pi}{3}) \\ \cos(\theta_{em} + \frac{4\pi}{3}) & \cos(\theta_{em}) & \cos(\theta_{em} + \frac{2\pi}{3}) \\ \cos(\theta_{em} + \frac{2\pi}{3}) & \cos(\theta_{em} + \frac{4\pi}{3}) & \cos(\theta_{em}) \end{pmatrix}$$

where θ_{em} is the electromechanical position : $\theta_{em} = p \cdot \theta$, p is the number of pairs of poles ($p = 3$) and θ is the angular position of the rotor. L_s , L_r , M_s , M_r and M_{sr} are the statoric and rotoric self-inductances, the internal statoric and rotoric mutual inductances and the external mutual inductance between the stator and the rotor, respectively. These mutual inductances are functions of the relative angular position between the rotor and the stator.

The actuator mechanism and equivalent circuit are shown in Figures 13 and 14. More details about this application can be found in [35].

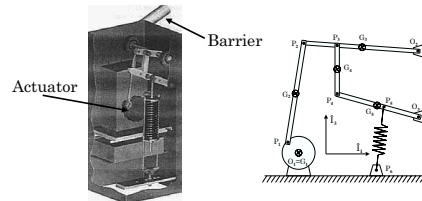


Figure 13: Picture of the mechanism driving the barrier

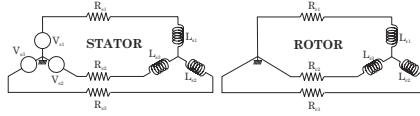


Figure 14: Stator and rotor equivalent circuit of a three-phase actuator

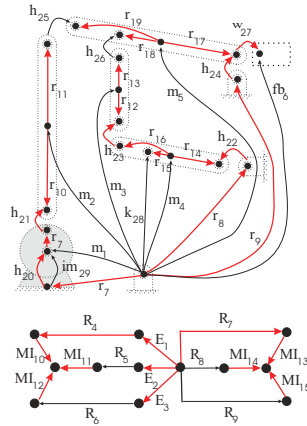


Figure 15: Linear Graph of Flexible Barrier

The simulation was started with zero currents and the barrier in the horizontal position. Sinusoidal input voltages (220V, 50Hz) are feeded to the actuator.

3.2.2 Virtual work approach

Even though ROBOTRAN[®] is capable of modelling flexible beams, we decided to model the flexible barrier using a finite-segment approach [36]. We divided the barrier into 30 segments and introduced an equivalent stiffness between each segment.

Concerning the electrical subsystem, we modelled it using the equivalent circuit given in figure 14 and the matrices given above. Because of the star-star connection, 2 constraints will be present between the 6 inductor currents. The reduced equations are generated and only 4 variables are integrated.

Because of the starting torque oscillations, most of the modes of the barrier are excited and high frequency vibrations occur in the barrier. At the same time, these vibrations have an influence on the load torque transmitted to the barrier, influencing the electrical actuator. This two-direction influence is typical in electromechanical systems.

3.2.3 Linear graph approach

Figure 15 depicts the linear graph representation of the flexible barrier system. The multibody portion of the model consists almost entirely of components discussed in Section 2.3. Three new components are included:

a weld joint (w), a flexible body (fb), and an induction motor (im). The weld joint, as its name suggests, simply locks two reference frames together.

The flexible body represents the flexible barrier. Details regarding the terminal equation for the flexible body can be found in [13]. To summarise, the beam uses a shape function to approximate the beam deformation. The beam is treated as an Euler-Bernoulli type beam (or Rayleigh type, to be precise).

For the flexible barrier, 5 variables were used to model the bending deflection of the barrier. This value was obtained by progressively adding more deformation variables until the simulation results converged. It was also found that including axial deformations in the flexible model did not affect the simulation results.

The terminal equation of the induction motor components is as follows:

$$T_{im} - \frac{1}{2p} \sum_{s=\{10,11,12\}} \sum_{r=\{13,14,15\}} \frac{\partial}{\partial \theta} (Msr \cdot i_s \cdot i_r) = 0 \quad (60)$$

The electrical graph is almost identical to that shown in Figure 14. The only exception is that the inductors have been replaced by mutual inductance components. The terminal equation for a MI edge is shown below:

$$v_{edge} - \sum_{m=1}^p L_{edge,m} \frac{di_m}{dt} - \sum_{n=1}^p \frac{d}{dt} (Msr_{edge,n} \cdot i_n) = 0 \quad (61)$$

Here the first summation represents the induction between MI components that are fixed with respect to the given edge (including self inductance). The second summation represents the mutual inductance between MI components that are related by the angle between the stator and the rotor. For descriptions of p , L and Msr see Section 3.2.1.

By selecting an appropriate tree (bolded in Figure 15) and using a current formulation, DynaFlex generates 14 equations: 5 for the rigid multibody system in terms of $\beta_{20} - \beta_{24}$, 5 for the deformation variables of the beam, and 4 for the electrical system in terms of i_5 , i_6 , i_8 and i_9 .

3.2.4 Bond graph approach

The system considered here reaches the limits of bond graph theory for modelling complex multi-dimensional systems, especially since manually writing the equations becomes very tedious for systems involving numerous bodies. As shown in the next paragraphs, it is possible to systematically draw a vector bond graph (also called a multibond graph), however the causality assignment and derivation of the equations is not straightforward and is still a source of problems.

Figure 16 shows the bond graph for the electrical subsystems. The inductive effects and the electromechanical transformation is accomplished in a mixed IC-field containing the inductances matrices \mathbf{L}_s , \mathbf{L}_r and \mathbf{M}_{sr} . The star-star connection is imposed by the two 0-junctions forcing the sum of the three currents to be zero. The star-star connections generate two constraints that can be seen in the bond graph as two derivative causalities for the I part of the IC-field.

The bond graph of the mechanism has been obtained using the method described by Favre [19]. Since we have a planar mechanism, we assume that every vector is expressed with respect to the inertial frame and the center of mass is the reference point for each body. Thus, the general bond graph proposed by Favre reduces to the diamond shape, as can be seen in Figure 17. The bond graph shown in Figure 17 contains 5 bodies (for simplicity, the beam is not shown in the bond graph) and the loop constraints are imposed by means of the zero-velocity source ($S_f = 0$). This way of opening and closing the kinematic loops is valid only because we have a revolute joint with the ground. Closing elsewhere would require us to get the relative velocities between the two points that are supposed to coincide all the time.

Even though bond graphs could be obtained for the system, their complexity and their nature prevent us from easily generating the equations. Also, the lack of tools for dealing with such complex bond graphs prevented us from getting simulation results for this approach.

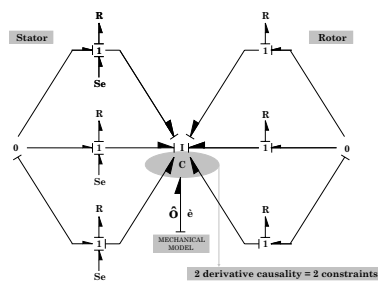


Figure 16: Bond graph of the three phase actuator

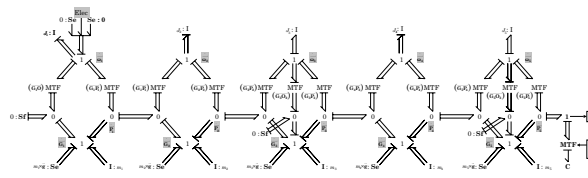


Figure 17: Bond graph of the mechanism driving the barrier

3.2.5 Comparison of the Simulation Results

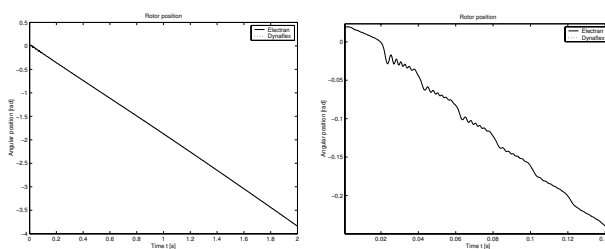


Figure 18: Position of the rotor of the actuator

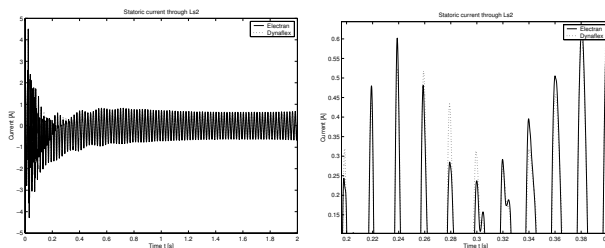


Figure 19: Current flowing through L_{s2}

The results are only compared between Virtual Work and Linear Graph approaches. Figures 18, 19 and 20 show the position of the rotor, the current flowing through statoric and rotoric inductances (right plot is a zoom to show the slight differences that are not visible on the left plot). These results have been obtained for a 1/53 gear ratio between the motor and the mechanism. They show almost identical results with slight differences in currents. These can be attributed to the slightly different models for the flexible beam.

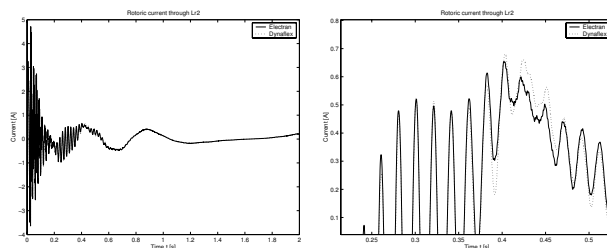
Note that the rotor oscillations shown in figure 18 are damped. The only dissipative elements in this model are the electrical resistance and changing the resistance value has an effect on the damping of the oscillations. This illustrates the tight interaction between electrical and mechanical subsystems.

Other simulations have been conducted with other gear ratios (1/16 and 1/1) and lead to similar conclusions. The differences appearing between the two approaches are accentuated when lower gear ratios are used. This is due to a tighter electromechanical interaction.

Changing the gear ratio also changes the behavior of the barrier. For example, the barrier cannot be opened when a 1/1 gear ratio is used and the magnitude of the vibrations raises.

4 Conclusions

Three different approaches have been presented and compared. The goal of these comparisons has been to present some commonly used formalisms and to show their advantages and limitations. Our aim is not to

Figure 20: Current flowing through L_{r2}

conclude which is “the best” approach, as the suitability of the approach is dependant upon the application being analysed. Each of the presented approaches has interesting features and even though some limitations are very strong, the choice for a formalism depends on many aspects: what system has to be modelled (1D, 2D, 3D, with or without constraints), what analysis has to be performed (time integration, optimization, inverse dynamics,...), what numerical tools will be used in what language (matlab, C-code, user’s tool,...). Moreover, depending on the background of the analyst, he will find himself more comfortable with one or another method.

However, some general comments about the different approaches can be stated. Mathematical approaches (for example, directly derived from Virtual Work Principle) have the advantage of leading to a minimal set of equations in every case, allowing for more efficient code. Also, constraints are easier to deal with when manipulating mathematical algebra. Reduction and constraint considerations can also be applied afterwards to the equations generated by the graph approaches but, this is out of the scope of the methods and means coming back to mathematical manipulations.

Generation of symbolic equations makes the generated models very portable, and providing a reduced set of equations helps to noticeably speed up simulation.

Linear Graph Theory offers the ability to select variables that will appear in the final equations. This allows different models to be constructed from the same graph. The approach is also suited to generating symbolic equations for complex system, as was shown in the paper.

Bond graphs are interesting for showing energy interactions between the different parts of the model. Nevertheless, they suffer from obvious limitations as soon as complex 3-dimensional (electro)mechanical systems are considered.

5 Acknowledgments

This work has been supported by the Belgian Program on Interuniversity Attraction Poles initiated by the Belgian State, Prime Minister’s office, Science Policy Programme (AMS IUAP 5/06). The scientific responsibility is assumed by its authors.

References

- [1] V. Hadwich, F. Pfeiffer, “The principle of virtual work in mechanical and electromechanical systems”, *Archive of Applied Mechanics* 65 (1995) 390-400 Springer-Verlag 1995
- [2] P.-Y. Willems, “Introduction à la Mécanique”, Masson Ed., Paris, 1979

- [3] Fiset P., Samin J.C. (1996), "Symbolic Generation of Large Multibody System Dynamic Equations using a New Semi-Explicit Newton/Euler Recursive Scheme", *Archive of Applied Mechanics*, vol.66, pp 187-199
- [4] Wehage R.A., Haug E.J. (1982), "Generalized Coordinate Partitioning for Dimension Reduction in Analysis of Constrained Dynamic Systems" *J. Mech. Design*, vol.134, pp 247-255
- [5] J. Meisel, "Principles of Electromechanical-energy conversion", McGraw-Hill, Inc. 1966
- [6] P. Maisser, O. Enge, H. Freudenberg, G. Kielo, "Electromechanical Interactions in Multibody Systems Containing Electromechanical Drives", *Journal of Multibody System Dynamics*, Vol. 1, n° 3, 1997, pp. 281-302
- [7] J.M.A. Sherpen, B. Klassens, L. Ballini, "Lagrangian Modelling and Control of Switching Networks with integrated coupled magnetics", *Proc. 39th IEEE Conf. Des. Contr.*, Sydney, Australia, December 2000, pp. 4054-4059
- [8] K. Schlacher, A. Kugi, R. Scheidl, "Tensor Analysis Based Symbolic Computation for Mechatronic Systems", *Mathematics and Computers in Simulation* 46, 1998, pp. 517-525
- [9] A.J. van der Schaft, " L_2 -gain and passivity techniques in non-linear control", Springer-Verlag, London, 2000
- [10] N. Biggs, E. Lloyd, R. Wilson, "Graph Theory: 1736-1936", Oxford University Press, 1976
- [11] H. Koenig, Y. Tokad, H. Kesavan, "Analysis of Discrete Physical Systems", Mc Graw-Hill, 1967
- [12] J. McPhee, "Automatic Generation of Motion Equations for Planar Mechanical Systems Using the New Set of "Branch Coordinates"", *Mech. Mach. Theory*, 33, 1998, pp. 805-823
- [13] P. Shi, J. McPhee, "Dynamic of Flexible Multibody Systems using Virtual Work and Linear Graph Theory", *Multibody System Dynamics*, Vol. 4, 2000, pp. 355-381
- [14] P. Roe, "Network and systems", Addison-Wesley, 1966
- [15] B. Muegge, "Graph-Theoretic Modelling and Simulation of Planar Mechatronic Systems", Master's Thesis, University of Waterloo, Canada, 1996
- [16] M. Scherrer, J. McPhee, "Dynamic Modelling of Electromechanical Multibody Systems", *Multibody System Dynamics*, Vol. 9, 2003, pp. 87-115
- [17] H.M. Paynter, "Analysis and design of engineering systems", The MIT Press, Cambridge, Mass, 1961
- [18] Dean C. Karnopp, Donald L. Margolis, Ronald C. Rosenberg, "System Dynamics : Modeling and Simulation of Mechatronic Systems (Third Edition)", John Wiley & Sons Inc., 2000
- [19] W. Favre, "Contribution à la Représentation Bond Graph des Systèmes Mécaniques Multicorps", PhD. Thesis, INSA de Lyon, 1997
- [20] W. Favre, S. Scavarda, "Bond Graph Representation of Multibody Systems with Kinematic Loops", *Journal of the Franklin Institute*, Vol. 335B-4, 1998, pp. 643-660, Ecole Doctorale des Sciences pour l'Ingénieur de Lyon, 1997
- [21] D. Karnopp, "Power-Conserving Transformations : "Physical Interpretations and Applications using Bond Graphs", *Journal of the Franklin Institute*, Vol. 288-3, 1969, pp. 175-201
- [22] R.C. Rosenberg, "Multiport Models in Mechanics", *Journal of Dynamic Systems, Measurement, and Control*, September 1972, pp. 206-212

- [23] D. Karnopp, "The Energetic Structure of Multibody Dynamic Systems", Journal of the Franklin Institute, Vol. 306-2, 1978, pp. 165-181
- [24] D. Karnopp, "An Approach to Derivative Causality in Bond Graph Models of Mechanical Systems", Journal of the Franklin Institute, Vol. 329-1, 1992, pp. 65-75
- [25] R.R. Allen, S. Dubowsky, "Mechanisms as components of dynamic systems: a bond graph approach", Journal of Engineering for Industry, Vol. 99-1, 1977, pp. 104-111
- [26] F.T. Brown, "Energy-Based Modeling and Quasi Coordinates", Journal of Dynamic Systems, Measurement, and Control, March 1981, Vol. 103, pp. 5-13
- [27] L. Bonderson, "Vector Bond Graphs Applied to One-Dimensional Distributed Systems", Journal of Dynamic Systems, Measurement, and Control, March 1975, pp.75-82
- [28] M.E. Ingram, G.Y. Masada, "The Extended Bond Graph Notation", Journal of Dynamic Systems, Measurement, and Control, March 1991, Vol. 113, pp. 113-117
- [29] P.C. Breedveld, "Proposition for an Unambiguous Vector Bond Graph Notation", Journal of Dynamic Systems, Measurement, and Control, September 1982, Vol. 104, pp. 267-270
- [30] P.C. Breedveld, "Multibond Graph Elements in Physical Systems Theory", Journal of the Franklin Institute, Vol. 319-1/2, 1985, pp. 1-36
- [31] A.M. Bos, J.L. Tiernego, "Formula Manipulation in the Bond Graph Modelling and Simulation Of Large Mechanical Systems", Journal of the Franklin Institute, Vol. 319-1/2, 1985, pp. 51-65
- [32] A. Zeid, C-H. Chung, "Bond Graph Modeling of Multibody Systems : a Library of Three-dimensional Joints", Journal of the Franklin Institute, Vol.329-4, 1992, pp. 605-636
- [33] W. Leonard, "Control of Electrical Drives", Springer-Verlag, 1985
- [34] P. Fiset, T. Postiau, L. Sass and J.C. Samin., "Fully Symbolic Generation of Complex Multibody Models", Mechanics of Structure and Machines, 30(1), 31-82. Marcel Dekker Inc., 2002
- [35] L.Sass, D. Telteu, D. Grenier, P. Fiset, "Unified symbolic modeling of mechatronic systems - Implementation and applications", Electrimacs 2002, Montreal, Quebec(Canada)
- [36] R.L. Huston, Y. Wang, M. Pereira, J. Ambrosio, "Flexibility Effects in Multibody Systems in Computer-Aided Analysis of Rigid and Flexible Mechanical Systems", J. (eds.), NATO ASI Series, Applied Sciences, Kluwer Academic Publishers, Dordrecht, 1993, pp. 351-376

A Electrical equations manipulations

The Virtual Work Principle leads to:

$$\frac{d}{dt} \left(\frac{\partial W_m^*}{\partial \dot{\mathbf{y}}} \right) - \frac{\partial W_m^*}{\partial \mathbf{y}} + \frac{\partial W_e}{\partial \mathbf{y}} + \underbrace{\sum_{r,s} u_r \frac{\partial q_i}{\partial \dot{\mathbf{y}}}}_{\mathbf{u}} = \mathbf{0} \quad (62)$$

in terms of the generalized variables $\mathbf{y} = \begin{pmatrix} \mathbf{q}_l \\ \mathbf{q}_c \end{pmatrix}$.

Commonly, the dynamic equations of an electrical circuit are expressed in terms of $\mathbf{w} = \begin{pmatrix} \mathbf{i}_l = \dot{\mathbf{q}}_l \\ \mathbf{u}_c = \mathbf{C}^{-1} \mathbf{q}_c \end{pmatrix}$.

This appendix show how, one can rewrite equation (62) in terms of *mathbf{w}* when linear constitutive equations are assumed.

In these developments, subscripts r , l , c , su and si respectively refer to resistances, inductances, capacitors, voltage sources and current sources.

First of all, using Kirchoff's current laws, we can rewrite the generalized voltages \mathbf{U} as:

$$\mathbf{U} = \left(\frac{\partial \hat{\mathbf{q}}_{r,s}}{\partial \hat{\mathbf{y}}} \right)^T \mathbf{u}_{r,s} = \mathbf{K}_r^T \mathbf{u}_r + \mathbf{K}_{su}^T \mathbf{u}_{su} \quad (63)$$

Assuming linear constitutive equations for resistors, equation (63) becomes:

$$\mathbf{U} = \mathbf{K}_r^T \mathbf{R} \mathbf{i}_r + \mathbf{K}_{su}^T \mathbf{u}_{su} \quad (64)$$

$$= \mathbf{K}_r^T \mathbf{R} \hat{\mathbf{q}}_l \begin{pmatrix} \dot{\mathbf{q}}_l \\ \dot{\mathbf{q}}_c \end{pmatrix} + \mathbf{K}_{su}^T \mathbf{u}_{su} \quad (65)$$

$$= \underbrace{\mathbf{K}_r^T \mathbf{R} \mathbf{K}_r}_{\mathbf{B}_r} \begin{pmatrix} \dot{\mathbf{q}}_l \\ \dot{\mathbf{q}}_c \end{pmatrix} + \underbrace{\mathbf{K}_r^T \mathbf{R} \mathbf{K}_{si}}_{\mathbf{B}_i} \mathbf{i}_{si} + \underbrace{\mathbf{K}_{su}^T}_{\mathbf{B}_u} \mathbf{u}_{su} \quad (66)$$

with \mathbf{R} being the diagonal matrix of resistance and \mathbf{K}_r , \mathbf{K}_{si} and \mathbf{K}_{su} being transformation matrices expressing resistor and voltage source currents in terms of the generalized variables. These matrices can be obtained by manipulations of Kirchoff's current equations.

Assuming linear constitutive equations for inductors and capacitors, we finally rewrite (14) as:

$$\left(\begin{array}{c} \frac{d}{dt} (\mathbf{L} \dot{\mathbf{q}}_l + \boldsymbol{\varphi}_0) \\ \mathbf{0} \end{array} \right) + \left(\begin{array}{c} \mathbf{0} \\ \mathbf{C}^{-1} \mathbf{q}_c \end{array} \right) + \mathbf{U} = \mathbf{0} \quad (67)$$

Equation (67) may be rewritten in terms of $\mathbf{w} = \left(\begin{array}{c} \mathbf{i}_l = \dot{\mathbf{q}}_l \\ \mathbf{u}_c = \mathbf{C}^{-1} \mathbf{q}_c \end{array} \right)$. This will give first-order differential equations of the generic form:

$$\frac{d}{dt} (\mathbf{M}_e \mathbf{w}) = \mathbf{D} \mathbf{w} + \mathbf{E} \begin{pmatrix} \mathbf{u}_{su} \\ \mathbf{i}_{si} \end{pmatrix} - \frac{d}{dt} \boldsymbol{\varphi}_0 \quad (68)$$

where $\mathbf{M}_e = \left(\begin{array}{cc} \mathbf{L} & \mathbf{0} \\ \mathbf{0} & \mathbf{C} \end{array} \right)$ is the "electrical mass matrix". The other matrices appearing in equation (68) are defined as:

$$\mathbf{B}_r = \begin{pmatrix} \mathbf{B}_{rl} & \mathbf{B}_{rlc} \\ \mathbf{B}_{rcl} & \mathbf{B}_{rcc} \end{pmatrix} \quad \mathbf{B}_i = \begin{pmatrix} \mathbf{B}_{il} \\ \mathbf{B}_{ic} \end{pmatrix} \quad \mathbf{B}_u = \begin{pmatrix} \mathbf{B}_{ul} \\ \mathbf{B}_{uc} \end{pmatrix} \quad (69)$$

$$\mathbf{D} = \begin{pmatrix} -\mathbf{B}_{rl} + \mathbf{B}_{rlc} \mathbf{B}_{rcc}^{-1} \mathbf{B}_{rcl} & \mathbf{B}_{rlc} \mathbf{B}_{rcc}^{-1} \\ -\mathbf{B}_{rcc}^{-1} \mathbf{B}_{rcl} & -\mathbf{B}_{rcc}^{-1} \end{pmatrix} \quad (70)$$

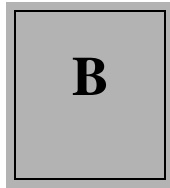
$$\mathbf{E} = \begin{pmatrix} -\mathbf{B}_{ul} + \mathbf{B}_{rlc} \mathbf{B}_{rcc}^{-1} \mathbf{B}_{uc} & -\mathbf{B}_{il} + \mathbf{B}_{rlc} \mathbf{B}_{rcc}^{-1} \mathbf{B}_{ic} \\ -\mathbf{B}_{rcc}^{-1} \mathbf{B}_{uc} & -\mathbf{B}_{rcc}^{-1} \mathbf{B}_{ic} \end{pmatrix} \quad (71)$$

Using the same manipulations, we can express the formulation when constraints are involved:

$$\frac{d}{dt} (\mathbf{M}_e \mathbf{w}) = \mathbf{D} \mathbf{x} + \mathbf{E} \begin{pmatrix} \mathbf{u}_{su} \\ \mathbf{i}_{si} \end{pmatrix} + \mathbf{F} \frac{d}{dt} \boldsymbol{\varphi}_0 + \mathbf{J}'^T \boldsymbol{\lambda}_c \quad (72)$$

where \mathbf{J}' is given by

$$\mathbf{J}' = \begin{pmatrix} \mathbf{J}'_{el} & -\mathbf{B}_{rlc} \mathbf{B}_{rcc}^{-1} \mathbf{J}'_{ec} \\ \mathbf{B}_{rcc}^{-1} \mathbf{J}'_{ec} \end{pmatrix} \quad (73)$$



Example 4: Electrical Circuit Benchmark

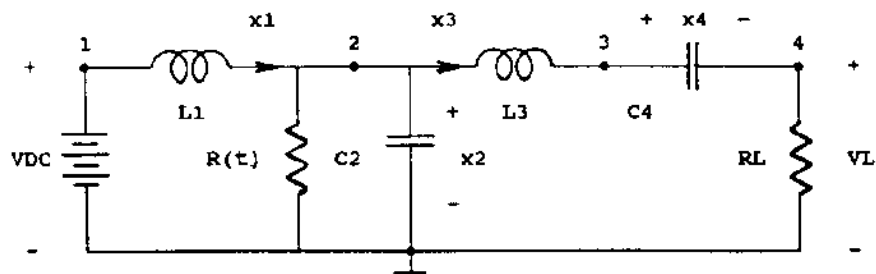
B.1 Benchmark Description

Software Comparison 3

Comparison 3 - Definition

This example is taken from the electrical engineering world.

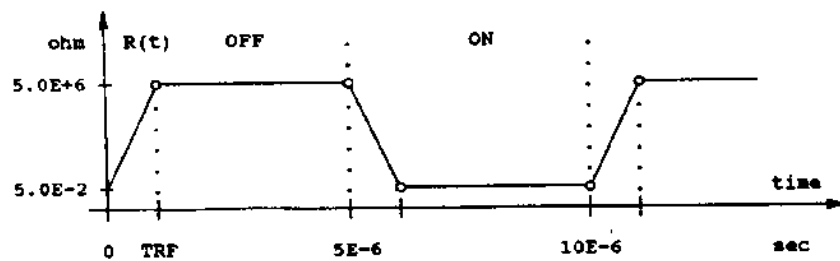
The basic class-E power amplifier was introduced by N.O. Sokal and A.D. Sokal in their classic paper from 1975 [1]. It is a switching-mode amplifier that operates with zero voltage and zero slope across the switch at switch turn-off. The actual numerical example is taken from J.C. Mandojana, K.J. Herman and R.E. Zulinski [2]. They use the following equivalent circuit of a generalized class-E amplifier as a test example for a procedure to evaluate steady state boundary conditions by means of MATLAB.



The component values are:

$VDC = 5$ volt, $L1 = 79.9E-6$ henry, $C2 = 17.9E-9$ farad, $L3 = 232.0E-6$ henry, $C4 = 9.66E-9$ farad and $RL = 52.4$ ohm.

The time dependent resistor $R(t)$ models the active device acting as a switch with an ON-resistance of 0.05 ohm and an OFF-resistance of $5.0E+6$ ohm. An extreme ON-resistance of value zero ohm will of course result in a pathological system i.e. the old story of what happens when an ideal capacitor with a certain charge is suddenly short circuited. Furthermore the DC voltage source will be short circuited through the ideal coil $L1$. As a function of time $R(t)$ is given in the following graph:



The duty ratio is 50%. The period is $10E-6$ seconds (frequency 100 kHz). The rise/fall time is $TRF = 1E-15$ seconds.

The equations describing the circuit may be the state-equations where inductor currents and capacitor voltages are chosen as system variables. By using the Kirchhoff voltage and current laws we get the following differential equations:

file:///E:/These/softs/electran/Projets/Comp3_e...sim%20-%20Argesim%20Homepage_fichiers/c3def.htm (1 sur 2) [27/11/2003 19:39:14]

Software Comparison 3

$$\begin{aligned}L1 \cdot dx1/dt &= -x2 + VDC \\C2 \cdot dx2/dt &= +x1 - x2/R(t) - x3 \\L3 \cdot dx3/dt &= +x2 - RL \cdot x3 - x4 \\C4 \cdot dx4/dt &= +x3\end{aligned}$$

where the variables are as follows: $x1 = IL1$ (the current of $L1$), $x2 = VC2$ (the voltage of $C2$), $x3 = IL3$ (the current of $L3$) and $x4 = VC4$ (the voltage of $C4$). Note that normally the setup of state equations demands a topological analysis of the circuit excluding some inductor currents and capacitor voltages as candidates for system variables (e.g if there is a loop of N capacitors then only $N-1$ of these may be given an arbitrary initial charge).

The following tasks should be performed:

- Calculation of the eigenvalues of the system in the ON-period: $R(t)=0.05$ ohm and in the OFF-period: $R(t)=5E+6$ ohm.
- Simulation of the system over the time interval $[0, 100E-6]$ sec with the zero-solution as initial state. Time curves of the state variables, the current in the switch resistor $IR(t) = x2/R(t)$ and the output voltage $VL = x3 \cdot RL$ are wanted.
- A parameter variation study over the time interval $[0, 9E-6]$ sec with initial solution equal to the final solution at $100E-6$ sec from task (b). The rise/fall time TRF should be varied through the values: $1E-15, 1E-11, 1E-9, 1E-7$ sec. The phase plane curves of $dx3/dt = VL3$ as a function of $x3 = IL3$ i.e the voltage difference $V2-V3$ as a function of the current $IL3$ are wanted. Time curves of the current in the switch resistor $IR(t) = x2/R(t)$ and the output voltage $VL = x3 \cdot RL$ are wanted.

References

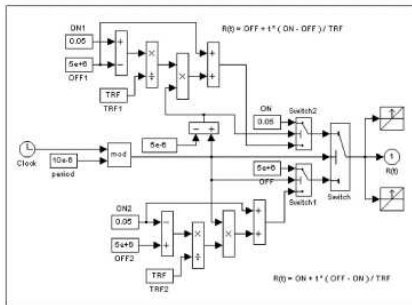
- Nathan O. Sokal and Alan D. Sokal, Class E - A New Class of High-Efficiency Tuned Single-Ended Switching Power Amplifiers, IEEE Journal of Solid-State Circuits, Vol. SC-10, No. 3, June 1975, pp. 168-176.
- Julio C. Mandojana, Kelly J. Herman and Robert E. Zulinski, A Discrete/Continuous Time-Domain Analysis of a Generalized Class E Amplifier, IEEE Transactions on Circuits and Systems, Vol. 37, No. 8, August 1990, pp. 1057-1060

Erik Lindberg, Institut of Circuit Theory and Telecommunication, 343 Technical University of Denmark, DK - 2800, Lyngby.

B.2 Proposed Solutions

**Comparison 3 - MATLAB/SIMULINK
Dynamic Function Approach**

MATLAB is a widely used software tool based on numerical vector and matrix manipulation, SIMULINK is MATLAB's extension for graphical modelling and numerical simulation of dynamic systems.



Model Description: The model for the amplifier (four states) is built up from the **Linear library** of SIMULINK. For the time-dependent resistor $R(t)$ first a table function (**From Workspace**) was used. But this method is not accurate enough, because the evaluation is done by linear interpolation, not synchronised with the time instants for rising and falling. Therefore a "dynamic" implementation as SIMULINK sub-model was chosen (see fig. above), which switches between the constant values for the ON- and OFF-period, and between the rising or falling slope. Two **Hit Crossing** blocks synchronise the switching times with the integration algorithm (state event finder).

Task a. The `linmod` command allows to linearise the system $\dot{x} = f(\bar{x}, \bar{u}, t)$ by calculating the matrices A and B for $\dot{x} = A(\bar{x} - \bar{x}_s) + B(\bar{u} - \bar{u}_s)$ around a linearisation point (\bar{x}_s, \bar{u}_s) , using numerical perturbation. As the model is linear in the ON- and in the OFF-period, it is not necessary to simulate until the linearisation point, and an arbitrary linearisation point (zero point) can be chosen. Only the time t (which is "hidden" in the 2nd algorithm parameter `para`) has to be set properly.

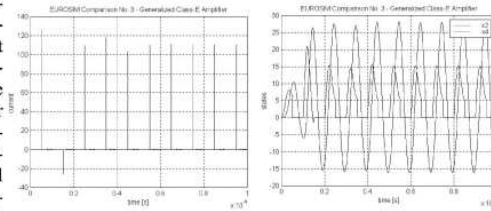
In order to validate the results of linearisation also the "generic" linear model is set up in MATLAB. In the following the `m`-file and the results are given (only in the OFF-period the results differ slightly):

```
» disp('linmod-linearisation: ON');
» xs=zeros(1,4);us=[];para=[1,e-5,0.7e-5];
» eigenvalues=eig(linmod('c3b',xs,us,para))
» disp('linmod-linearisation: OFF');
» para=[1,e-5,0.3e-5];
```

```
» eigenvalues=eig(linmod('c3b',xs,us,para))
» disp('Generic linear model: ON');
» A(1,:)=0 -1 0 0/L1;
» A(2,:)=1 -1/0.05 -3 0/C2;
» A(3,:)=0 1 -RL -1/L3;A(4,:)=0 0 1 0/C4;
» eigenvalues=eig(A)
» disp('Generic linear model: OFF');
» A(2,2)=(-1/5e-6)/C2;eigenvalues=eig(A)
```

```
linmod-linearisation: ON
eigenvalues = 1.0e+009 *
 -1.11731759441059
 -0.00000062578277
 -0.00011303881490 + 0.000658352220471i
 -0.00011303881490 - 0.000658352220471i
linmod-linearisation: OFF
eigenvalues = 1.0e+006 *
 -0.05470820246681 + 1.040797197845001i
 -0.05470820246681 - 1.040797197845001i
 -0.05822841860988 + 0.532750192401451i
 -0.05822841860988 - 0.532750192401451i
Generic linear model: ON
eigenvalues ..same as before
Generic linear model: OFF
eigenvalues = 1.0e+006 *
 -0.05470820246506 + 1.040797197854831i
 -0.05470820246506 - 1.040797197854831i
 -0.05822841860988 + 0.532750192405971i
 -0.05822841860988 - 0.532750192405971i
```

Task b. As the time instants for switching are synchronised with the integration, the results are sufficiently accurate, also for the impulse-like current $I_R(t) = x_2/R(t)$ and the nearly discontinuous states - see next two figures, using the **stiff Mod. Rosenbrock** method with initial stepsize **TRF**.



Task c. The following commands perform a simulation run over $[0, 9E-6]$ with storing the terminal values of the states, reinitialize the states with these values, and do the parameter study for varying rise/fall time.

Zooming into the phase plot shows, that only the simulation for **TRF** value $1.e-7$ is different from the smaller values.

```
» clear all;TRF = 1e-15; sim('c3c');
» x1=x(2); x2=x(3); x3=x(4); x4=x(5);
» trf=[1e-15 1e-11 1e-9 1e-7];
» for n=1:4
» TRF=trf(n); sim('c3c');
» ei(n)=erg1; e2(n)=erg2;
» vl(n)=VL3; ll(n)=LL3; end
```

F. Breitenecker, M. Lingl, Dept. Simulation Technique, TU Vienna, Wiedner Hauptstr. 8-10, A-1040 Vienna, email: Felix.Breitenecker@tuwien.ac.at

Figure B.1: Breitenecker's Proposition [8]

Comparisons of Simulation Tools and Simulation Technique

Comparison 3 – Maple Closed Model / Numerical Simulation

Maple V is a comprehensive computer system for mathematics, mainly used for symbolic computation, but also including numerical facilities. In this solution these numerical facilities of Maple are, even if the problem would allow to take advantage of a symbolic solution because of the linear periods of the system. Maple's `linalg` and `detools` libraries are needed, so they have to be called at the beginning of the Maple work sheet.

Model description: The differential equations describing the model were implemented in matrix form $dx/dt = A * x$, $x \in \mathbb{R}^3$ and $A \in \mathbb{R}^{3 \times 3}$, with the time dependent matrix $A = A(R(t))$ and the resistivity $R(t)$. The resistivity $R(t)$, an almost rectangular function, with raise-fall time t_r (TRF) of magnitude $1e-15$ seconds and constant periods of magnitude $1e-6$ seconds, was implemented as a continuous function using Maple's `piecewise` and `trunc` commands:

```
t_red:=(1e-5)*trunc(t/(1e-5));
R:=piecewise((0<=t_red and t_red<TRF),R1
(TRF<t_red and t_red<5e-6),R2
(5e-6<=t_red and t_red<5e-6+TRF),R3
(5e-6+TRF<=t_red),R4)
```

Task a: The Maple command `eigenvalues(.)` was used to calculate the eigenvalues of $A(t)$ at two different time points, once during the on-period and once during the off-period of the resistivity $R(t)$. Linearization was not necessary because of the implementation of the system in matrix form $dx/dt = A * x$:

```
eigenvalues(subs(t=0,A(t)));
eigenvalues(subs(t=TRF,A(t)));
```

Eigenvalues on-period	Eigenvalues off-period
-1.1173e+009	-5.4708e+004 +1.0408e+006i
-6.2578e+002	-5.4708e+004 -1.0408e+006i
-1.1304e+005 +6.5835e+005i	-5.8228e+004 +5.3275e+005i
1.1304e+005 -6.5835e+005i	-5.8228e+004 -5.3275e+005i

Figure 1: Table of eigenvalues

Task b: To simulate this stiff system Maple's function `dsolve` was used. Invoking the `dsolve` function with options `type=numeric` and `method=lsode` causes a numerical solution to be found using the Livermore Stiff ODE solver. The final values of the solution were stored for later use. For the plots Maple's `odeplot` command was called with the `numpoints=500` option to be sure the plotting-procedure does not ignore the current peaks (without this option – forcing a "high-frequency" output – the plots do not show the

peaks exactly and seem to be wrong, although the ODE solver works correctly with step sizes small enough for the rise and fall slope).

The figure after the Maple commands shows the current $IR(t)$ [solid] in Ampere and the voltage $VL(t)$ [dotted] in 0.1 Volts.

```
x_sol:=dsolve({deg,init},{x1(t),x2(t),
x3(t),x4(t)},type=numeric,
method=lsode[backdiag], 'itask'=3,
output=procedurelist);
odeplot(x_sol,[t,x2(t)/R],0..1e-4,
numpoints=500);
```

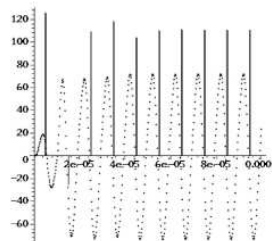


Figure 2: State variables IR and VL as functions of time

Task c: With the new initial condition stored in task b) the simulation using Maple's functions `dsolve` and `odeplot` was repeated four times in a loop, varying the rise/fall time TRF and storing the results to compare them. The same commands were used as in task b). A plot in the phase plain shows the small difference between the three coinciding curves at $TRF=1e-15$ seconds, $TRF=1e-11$ seconds, $TRF=1e-9$ seconds and the one slightly differing at $TRF=1e-9$ seconds.

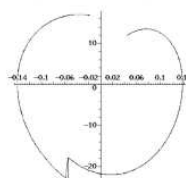




Figure 3: Phase plane curves $VL3=dx/dt$ as a function of $IL3=x_1$

All calculations were performed on a Celeron 500 under Linux 2.x and Maple V Release 5.1. The executing of the whole Maple worksheet took about 20 seconds and shows the numerical possibilities of Maple V Release 5.1.

N. Viertl, Vienna University of Technology,
viertl@gmx.at


SIMULATION NEWS EUROPE


C3 Generalized Class-E Amplifier – MATLAB

Overall - model approach

Simulator. Matlab is a widely used software tool based on vector and matrix manipulations for complex high speed numerical calculations and simulations. SIMULINK, an extension to MATLAB offers a graphical modelling environment enabling the user to operate on complex systems on a high abstraction level.

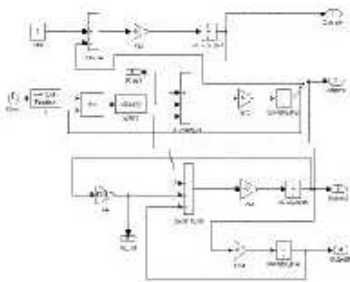


Figure 1: SIMULINK Model

Model: The approach followed in this solution is based on SIMULINK. The graphical model description is given in figure 1 where one can point out that not the whole model is implemented in SIMULINK. The resistance R(t) is realized with a MATLAB-function R(t):

```

function R_out=R(t)
global TRF;
TRF=1e-15; b=(15e+5)-(15e-2)/(TRF);
t_ced=mod(t,(10e-5));
if (t_ced < t_ced/2) R_out=(15e-2)+b*t_ced;
else R_out=(15e-5)+b*(t_ced-(15e-5));
end
    
```

Task a: Calculation of Eigenvalues. This task is implemented in MATLAB as a m-function. Therefore the given ordinary differential equations is realized in the form $dx/dt=Ax$, where x is a vector of R^2 and A a time dependent matrix of $R^{2 \times 2}$. The eigenvalues computed with the MATLAB eig() command are presented in the table below:

Eigenvalues OFF-Period	Eigenvalues ON-Period
-5.4708E+4 + 1.0408E+6 i	-1.1173E+9
-5.4708E+4 - 1.0408E+6 i	-6.2578E+2
-5.8228E+4 + 5.3273E+5 i	-1.1304E+5 + 6.5835E+5 i
-5.8228E+4 + 5.3273E+5 i	-1.1304E+5 - 6.5835E+5 i

Because of these eigenvalues it is clear, that the system is very stiff.

Task b: Simulation of Class E-Amplifier. To simulate this system, the MATLAB ode23-solver, a Rosenbrock – type method for stiff differential equations is used. Under the given initial state, the solution is given in figure 2 where the current $iR=x_2/R$ [in Ampere] and $V_L=x_3*RL$ [in 0.1 Volt] is visualized over time. Figure 3 shows $x_2(t)$ and $x_3(t)$.

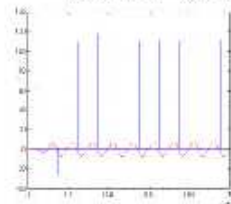


Figure 2:
iL and VR over time t

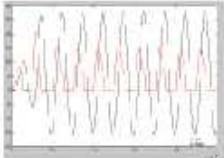


Figure 3:
x2 and x3 over time t

Task c: Parameter Variation Study
The parameter varied is the rise/fall time TRF. For values between 1e-15 and 1e-7 the phase portrait for x_3 , $V2V3$, $IL3$ is given in the figure 4.

Only for TRF=1e-7 the results differ. This TRF time means a slow switching process (time period of R(t) is 10e-6). All calculations were performed on a PII 400MHz with 256MB RAM on a LINUX platform.




Figure 4: Phase Portraits

The calculation for task b) are about 35 sec, which is slow. The reason is the definition of the resistance as external MATLAB function. This way of implementation is one the one side very flexible and easy, but on the other hand the evaluation is very slow because of the interpretative manner.

Bombosi Rada, Michael Wübner
Dept. Simulation, Vienna University of Technology
rada@surfeu.at, mwubner@smat.ttu.tuwien.ac.at

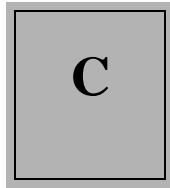
COMPARISONS

Issue 29/30

29

December 2000

Figure B.3: Rada's Proposition [50]



Scara Robot Benchmark

C.1 Benchmark Description

Comparison 11 - Definition

Comparison 11 (SCARA Robot) - Definition

C11 - SCARA Robot, is the 11th comparison on simulation software and modelling techniques. It is the 6th comparison of continuous type and deals with the handling of implicit systems.

Background: Mechanical and mechatronic systems often result in an implicit second order model description of the type

$$M(\vec{q})\ddot{\vec{q}} = \vec{g}(\vec{q}, \dot{\vec{q}}, \vec{u}, t)$$

with a state-dependent mass matrix M , an acceleration vector $\ddot{\vec{q}}$ and a generalised force function \vec{g} .

Simulators often impose restrictions for this type of model descriptions. Only a few simulators accept the description as given above, some allow a description as an implicit first order system

$$A(\vec{z})\dot{\vec{z}} = \vec{h}(\vec{z}, \vec{u}, t)$$

and some require the explicit description given by

$$\dot{\vec{z}} = \vec{f}(\vec{z}, \vec{u}) = A(\vec{z})^{-1}\vec{h}(\vec{z}, \vec{u}, t).$$

The symbolic derivation of the explicit form is only possible with reasonable effort for very small systems or systems with a simple-structured mass matrix. Therefore it is common practice to carry out the inversion of the mass matrix numerically.

Another interesting question is, whether a simulator that permits implicit descriptions breaks the implicit loop before integrating the states or uses an implicit integration scheme to solve the system directly. Few simulators offer so-called DAE solvers for the second method, sometimes with restrictions with respect to other features like linearisation, event handling etc. In general, advanced features like implicit description, DAE solvers, algebraic loop solvers etc. result in higher computation times and in some computational overhead. Therefore it has to be checked whether it is worth to use such a tool or to work "conventionally" by setting up an explicit system description. In order to investigate this class of problems, a model for a SCARA robot (*Selective Compliance Assembly Robot Arm*) as shown on the title page of this SNE issue was chosen.

Comparison 11 - Definition

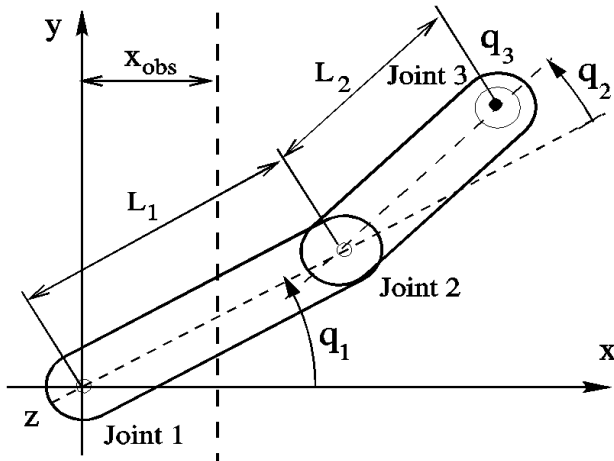


Figure 1

Mechanical System (Task a)

A three-axis SCARA robot as indicated in Fig.1 is investigated. This robot type has two vertical revolute joints and one vertical prismatic joint. The axes of all three joints are vertical (parallel to the z-axis in Fig. 1). The joint vector \vec{q} consists of the joint angles q_1 and q_2 and the joint distance q_3 .

$$\vec{q} = (q_1, q_2, q_3)^T, \quad \dot{\vec{q}} = \frac{d\vec{q}}{dt}, \quad \ddot{\vec{q}} = \frac{d\dot{\vec{q}}}{dt}$$

The equations of motion of can be written in the following compact form

$$M \ddot{\vec{q}} = \vec{b}$$

The mass matrix M is block-diagonal and can be easily inverted symbolically.

$$M = \begin{bmatrix} ma_{11} & ma_{12} & 0 \\ ma_{21} & ma_{22} & 0 \\ 0 & 0 & ma_{33} \end{bmatrix}$$



Comparison 11 - Definition

Several elements of M depend on the joint variable q_2

$$\begin{aligned} ma_{11} &= \Theta_1 + 2\Theta_2 \cos(q_2) + \Theta_3, \\ ma_{12} &= \Theta_2 \cos(q_2) + \Theta_3, \\ ma_{21} &= ma_{12}, \quad ma_{22} = \Theta_3, \\ ma_{33} &= m_{3L} + \Theta_{3\text{mot}} u_3^2. \end{aligned}$$

The calculation of the moments of inertia Θ_i is based on the assumption that the two physical links are rods of mass m_1, m_2 with homogeneous mass distribution along the length L_1, L_2 . The stator mass of the vertical drive motor is m_{3A} , the moment of inertia of the rotating parts is $\Theta_{3\text{mot}}$ and the mass of the load is m_{3L} .

$$\begin{aligned} \Theta_1 &= \left(\frac{m_1}{3} + m_2 + m_3\right) L_1^2, \quad \Theta_2 = \left(\frac{m_2}{2} + m_3\right) L_1 L_2, \\ \Theta_3 &= \left(\frac{m_2}{3} + m_3\right) L_2^2, \quad m_3 = m_{3A} + m_{3L} \end{aligned}$$

The right-hand side of the dynamic equations is

$$\begin{aligned} \vec{b} &= (b_1, b_2, b_3)^T, \\ b_1 &= T_1 + \Theta_2 (2\dot{q}_1 \dot{q}_2 + \dot{q}_2^2) \sin(q_2), \\ b_2 &= T_2 - \Theta_2 \dot{q}_1^2 \sin(q_2), \quad b_3 = T_3 - m_{3L} g \end{aligned}$$

with the joint torques $T_1(t), T_2(t)$ and the joint force $T_3(t)$. Numerical data for the geometric and mass parameters of the SCARA robot are given below:

$$\begin{aligned} m_1 &= 8\text{kg}, \quad L_1 = 0.4\text{m}, \quad g = 9.81\text{m/s}^2, \\ m_2 &= 6\text{kg}, \quad L_2 = 0.3\text{m}, \quad u_3 = 1047\text{m}^{-1}, \\ m_{3A} &= 2.5\text{kg}, \quad m_{3L} = 0.5\text{kg}, \quad \Theta_{3\text{mot}} = 9.1 \cdot 10^{-6} \text{kgm}^2 \end{aligned}$$

Servo Motor and PD-Control (Task b)

The electrical relationship of the armature of a robot servo motor is given by a first order differential equation

Comparison 11 - Definition

$$\dot{I}_i = \frac{(U_{ai} - k_{Ti} u_i \dot{q}_i - R_{ai} I_{ai})}{L_{ai}}, \quad i = 1, 2, 3$$

$$I_{ai} = [-I_i^{\max} \leq I_i \leq I_i^{\max}], \quad i = 1, 2, 3$$

where $U_{ai}(t)$ is the applied armature voltage. The resulting armature current I_i is limited to maximum value I_i^{\max} that can be calculated from the maximum permitted torque T_i^{\max}

$$I_i^{\max} = T_i^{\max} \left(\frac{\sqrt{3}}{2} k_{Ti} \right)^{-1}, \quad i = 1, 2, 3$$

The joint torque (force) T_i of a motor is proportional to the armature current I_{ai} and given by

$$T_i = u_i \frac{\sqrt{3}}{2} k_{Ti} I_{ai}, \quad i = 1, 2, 3$$

Numerical values for the motor constant k_{Ti} , the gear ratio u_i , the resistance R_i and the inductance L_i for each motor are given below. Note that u_3 includes the transformation from the rotational to the linear motion and is not dimensionless.

$$\begin{aligned} k_{T1} &= 0.4 \text{ Vs}, & k_{T2} &= 0.25 \text{ Vs}, & k_{T3} &= 0.4 \text{ Vs}, \\ R_{a1} &= 3.9 \text{ Ohm}, & R_{a2} &= 50 \text{ Ohm}, & R_{a3} &= 40 \text{ Ohm}, \\ L_{a1} &= 7.3 \text{ mH}, & L_{a2} &= 25 \text{ mH}, & L_{a3} &= 25 \text{ mH}, \\ u_1 &= 130, & u_2 &= 100, & u_3 &= 1047 \text{ m}^{-1}, \\ T_1^{\max} &= 2.3 \text{ Nm}, & T_2^{\max} &= 0.6 \text{ Nm}, & T_3^{\max} &= 0.6 \text{ Nm} \end{aligned}$$

In order to control the point-to-point motion of the robot a rather primitive single-axis PD-control is employed. For a given target joint position vector \hat{q} position errors $(\hat{q}_i - q_i)$ can be calculated. From the position errors and the joint velocities \dot{q}_i the control voltage U_{ai} is determined by

$$\begin{aligned} U_i &= P_i (\hat{q}_i - q_i) - D_i \dot{q}_i, \quad i = 1, 2, 3 \\ U_{ai} &= [-U_i^{\max} \leq U_i \leq U_i^{\max}], \quad i = 1, 2, 3. \end{aligned}$$

Proportional gains P_i and derivative gains D_i are given for each controller. In regular operation mode the armature voltage shall be limited by $U_{i\text{reg}}^{\max}$. However, in an emergency situation $U_{i\text{max}}^{\max}$ may be used (see task c).

Comparison 11 - Definition

$$\begin{aligned}
 P_1 &= 1000 \text{ V}, \quad P_2 = 1000 \text{ V}, \quad P_3 = 5000 \text{ V}, \\
 D_1 &= 10 \text{ Vs}, \quad D_2 = 25 \text{ Vs}, \quad D_3 = 10 \text{ Vs}, \\
 U_{1\text{reg}}^{\text{max}} &= 100 \text{ V}, \quad U_{2\text{reg}}^{\text{max}} = 75 \text{ V}, \quad U_{3\text{reg}}^{\text{max}} = 90 \text{ V}, \\
 U_{1\text{max}}^{\text{max}} &= 230 \text{ V}, \quad U_{2\text{max}}^{\text{max}} = 230 \text{ V}, \quad U_{3\text{max}}^{\text{max}} = 230 \text{ V}
 \end{aligned}$$

Obstacle definition and collision avoidance manoeuvre (Task c)

An elevation profile for the x - y plane is given by

$$\begin{aligned}
 h &= h_{\text{obs}} \quad \forall x \leq x_{\text{obs}}, \quad h = 0 \quad \forall x > x_{\text{obs}} \\
 h_{\text{obs}} &= 0.2 \text{ m}, \quad x_{\text{obs}} = 0.25 \text{ m}
 \end{aligned}$$

with a straight borderline at x_{obs} , separating the elevated area h_{obs} from the area with zero elevation. The border represents an obstacle for the tool-tip of the robot arm. Contact has to be avoided when the robot tool-tip moves from the starting point to a target position in the elevated area. Possible contact must be detected during robot motion and control of the rotational drives must be changed until the tool-tip has cleared the obstacle height. Maximum voltage $U_{i\text{max}}^{\text{max}}$ may be used in this situation for motors 1 and 2 to obtain maximum deceleration. An obstacle sensor shall measure the distance from the robot tool-tip to the borderline and shall trigger an emergency manoeuvre if the distance d falls below the critical distance $d_{\text{crit}} = 0.1 \text{ m}$.

If $(x_{\text{tip}} - x_{\text{obs}}) \leq d_{\text{crit}}$ and $q_3 < h_{\text{obs}}$ then
decelerate \dot{q}_1, \dot{q}_2 until $q_3 > h_{\text{obs}}$.

The x -position of the tool-tip can be calculated from

$$x_{\text{tip}} = x_3 = L_1 \cos(q_1) + L_2 \cos(q_1 + q_2).$$

The following tasks should be performed:

Task a) Modelling method. There are several ways to formulate and implement the model, depending on the simulator's features, e.g.

- i) "manual" symbolic manipulations for setting up explicit model equations, implementation of the explicit model description,
- ii) derivation of explicit equations using software for symbolic calculations, implementation of the explicit model description,

Comparison 11 - Definition

- iii) using special features of the simulator for deriving and simulating the equations (mechatronic modules, etc.),
- iv) implementation of the implicit equations, using algebraic loop breaking features of the simulator,
- v) implementation of the implicit equations, using an implicit solver of the simulator, etc.

The simulator's features for this type of models should be sketched briefly by giving (parts of) the model description of at least one (but preferably of some) of the above given methods. In case of alternative modelling approaches the effectiveness should be compared, taking into account preparation time, necessary knowledge for certain alternatives, etc.

Task b) Simulation of a point-to-point motion, controlled by a single axis PD-control shall be performed. No obstacle is present for this task.

Initial values at $t = 0$:

$$q_1 = q_2 = q_3 = 0, \quad \dot{q}_1 = \dot{q}_2 = \dot{q}_3 = 0$$

Target (terminal) values at \hat{t} :

$$\hat{q}_1 = \hat{q}_2 = 2, \quad \hat{q}_3 = 0.3 \text{ m}, \quad \dot{\hat{q}}_1 = \dot{\hat{q}}_2 = \dot{\hat{q}}_3 = 0$$

As results graphs of the joint positions should be plotted. In case of alternative model descriptions simulation times are to be compared.

Task c) Collision avoidance may cause difficulties in the models descriptions. Based on the point-to-point control of task b), now an obstacle has to be avoided (see problem definition). Extend the model description by a collision avoidance feature of the proposed type, using for instance state-dependent control, state event mechanism, etc.

For documentation the program extensions are to be outlined and a plot of $x_{ip}(t)$, $(q_3(t) - h_{obs})$ and x_{obs} over t is to be given.

References: R.J.Schilling, Fundamentals of Robotics, Prentice-Hall, 1990

Acknowledgement: The authors thank Dr. G. Kronreif (TU Vienna) for providing realistic robot data.

Horst Ecker, Institute for Machine Dynamics and Measurement, email: hecker@email.tuwien.ac.at, Tel: +43-1-58801 5567, and Felix Breitenecker, SIMTECH, Vienna University of Technology, Wiedner Hauptstr. 8-10, A-1040 Vienna/Austria.

C.2 Proposed Solutions

Comparison 11 – MATLAB/SIMULINK Hybrid Modelling Approach – Model Level

MATLAB is a widely used software tool based on numerical vector and matrix manipulation, SIMULINK is MATLAB's extension for graphical modelling and numerical simulation of dynamic systems.

Model Description (Task a): The model was implemented in two ways, using MATLAB 5.2, SIMULINK 1.3. First SIMULINK's Algebraic Constraint block was used and the implicit equation $b(q, \dot{q}) - M(q_2) \cdot \dot{q} = 0$ directly implemented:

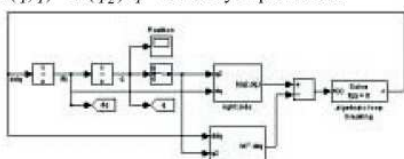


Fig.1: Implicit Equation, Algebraic Loop Breaking

For every integration step SIMULINK's Algebraic Constraint block searches for a solution of the implicit equation. This procedure is comfortable and does also work in the presence of a Hit Crossing block (which was needed for task c)! For the second solution the systems mass matrix M was inverted symbolically outside MATLAB and the explicit equation $\ddot{q} = M(q_2)^{-1} \cdot b(q, \dot{q})$ implemented in SIMULINK.

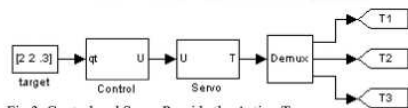
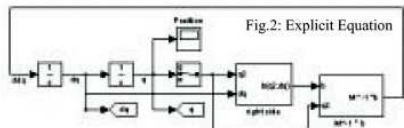


Fig.3: Control and Servo Provide the Acting Torque

Point to Point Control (Task b): For point to point movement both solutions use the same controller. A target vector is the input for Submodel Control which contains the PD-Controller (q and $\dot{d}q$ are provided by `Get0` blocks). The output u (the applied voltage) is fed into submodel Servo which models the servo drives of the three axes. Finally the resulting torque T is provided for the calculation of the right hand side b . For implementation of the boundaries of the voltages and currents, SIMULINK offers a very comfortable way: Voltage u is bounded by a Saturation block inside the submodel

Control and the resulting armature current is limited by the corresponding Integrator block itself (inside submodel Servo). Figure 4 shows the graph of the joint positions for the demanded movement. Of course execution of the explicit model takes more time than for the implicit one. The processing times were measured from MATLAB using the commands `tic` and `toc` (average of four runs):

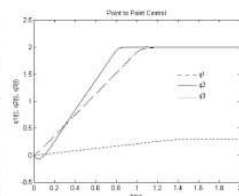


Fig.4: Joint Positions

Of course execution of the explicit model takes more time than for the implicit one. The processing times were measured from MATLAB using the commands `tic` and `toc` (average of four runs):

Model description	Norm. CPU-time
Explicit - inverted matrix	1 (4.57s at P150)
Implicit - algebraic loop breaking	3.28

Obstacle Avoidance (Task c): For collision avoidance submodel Control was extended. The distance between the obstacle and the tool tip is permanently checked. If it gets smaller than the critical distance (event `too near`) the target positions for the state-variables are changed to the current position and the emergency maximum voltages are allowed. The robot arms 1 and 2 slow down and return to the position where the danger has been detected. Just after the tool tip of the robot has reached an admissible height (event `clear`) the original target position is reactivated and the arms 1 and 2 start to move again.

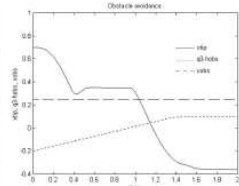


Fig. 5: Obstacle Avoidance

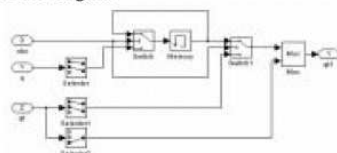


Fig. 6: Changed Target Position in Case of Obstacle

Figure 6 shows how the target position is changed in case of the event `too near`. As soon as the signal `obs` equals one, the current position q is stored in a memory block. This position is used as target for axes 1 and 2 until the occurrence of event `clear`: `obs` turns zero, target q_t is accepted again and the voltages are limited to U_{reg} .

J. Scheikl, M. Lingl, SIMTECH / ARGESIM, TU Vienna, Wiedner Hauptstr. 8-10, A-1040 Vienna, email: joxg@osiris.tuwien.ac.at

Figure C.1: Scheikl's Proposition [61]

Comparison 11 – DYMOLA

Classical mechanical approach
Automatic-symbolical and numerical inversion

DYMOLA (Dynamic Modelling Laboratory) is an object-oriented simulation environment for the modelling, simulation and visualisation of continuous processes. Besides the classical textual model definition, Dymola provides an editor for graphical model editing together with comfortable possibilities to reuse objects by means of (graphical) libraries. Model details can be given by ODE's and DAE's in the Dymola's object-oriented modelling language; for simulation either Dymola's simulator Dymosim can be used, or other commercial simulators (e.g. ACSL, SIMULINK).

Model Description (Task a): One of the main characteristics of Dymola is the law-oriented model description allowing the formulation of DAE models. This description can be manipulated symbolically depending on certain options. In this solution the textual mode of Dymola is used, defining the equations of motion in DAE form in a Dymola class, instantiated once:

```
model class components
  parameter L1=0.4, L2=0.3, L3=0.3, ...
  constant m1=8, m2=6, m3A=2.5, ...
  local M(3,3), b(3), q(3), dq(3), ...
  cut torq1(T1) torq2(T2) torq3(T3) ...
  M=[th1+2*th2*cos(q(2))+th3, ...
    b=[T1+th2*(2*dq(1)*dq(2)+dq(2)**2)* ...
    q=[q1;q2;q3]; dq=[dq1;dq2;dq3]
  dq=der(q)
  M*der(dq)=b
end
```

Dymola is able to transform these equations to explicit form by symbolically inverting the mass matrix, resulting in an explicit system. Another method is provided by a numerical inversion of the mass matrix by means of an iterative algorithm, resulting in a pseudo-explicit system. Depending on options, Dymola translates the system into both forms (as a third form, a pure implicit description is possible).

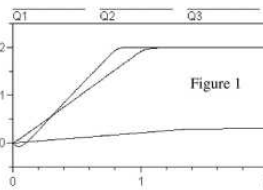
Point to Point Control (Task b): As three motors and three controls are required, two classes are defined: one describing the equations to specify a motor, one defining a control. Then the three instances of each class are connected with the instance of the components class, e.g.:

```
model class drive
  ...
  cut torque (T)
end
model scara
  submodel(drive) d1 d2 d3
  submodel(components) robot
  connect d1:torque at robot:torque1
  connect d2:torque at robot:torque2
  ...
end
```

The bounds for the voltages and the currents are considered by **if**-statements:

```
U=if abs(P*(qdach-q)-D*qd-Umax)0 then
  Umaxreg else P*(qdach-q)-D*qd-Umax
```

For simulation Dymosim is used. Dymola translates each **if** and **when** statement into a state event. Therefore the bounds for the current are formulated with state events in each target simulator. Dymosim handles state events by means of built-in features of the integration algorithms DASSL (used here) and LSODER. Unfortunately, Dymosim does not use the DASSL algorithm for direct integration of implicit equations (third method).



Of course numerical inversion of the implicit system takes more time than the integration of the explicit system; the relation is shown in the table below.

Figure 1 shows the graphs of the three joint positions.

Model description	Norm. CPU-time
Task b) explicit – symbolic inversion	1 (1.32s at P120)
Task b) pseudo-explicit – numerical inversion	1.69
Task c) explicit – symbolic inversion	1.43
Task c) pseudo-explicit – numerical inversion	2.27

Obstacle avoidance (Task c): For the collision avoidance a new class that observes and controls each state variable is implemented. The two instances (one for each state) check the distance between the obstacle and the tool-tip of the robot, and apply either the emergency maximum voltages and set the target positions for the state-variables to the current position, or reset the target positions to the original values bounding the voltages to their regular-mode-interval. Below a part of this description is shown, which is translated into a state event in Dymosim:

```
Xcrit=if abs(Xtip-Xobs)Xobs+Dcrit) then
true else false
Umax=if Xcrit and (q3-Hobs) then
Umaxmax else Umaxreg
qldach=if Xcrit and (q3-Hobs) then
q1 else q1target
```

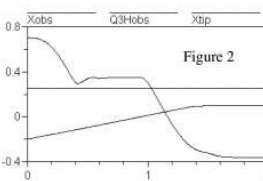
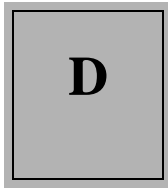


Figure 2 illustrates the behaviour of the tool-tip in this situation: it does not cross the critical region as long as the end-effector has not cleared the obstacle's height. This implementation of

obstacle avoidance increases simulation time by a factor 1.43 (see table).

E. Forsthuber, Technologie-Zentrum Steyr, A-4400 Steyr, email: forsthuber@titania.tuwien.ac.at

Figure C.2: Forsthuber's Proposition [21]



Three phase induction motor

D.1 Identification procedure

Before to identify our motor, we observed that it saturates above voltage supply of 250V, even with open shaft. This led us to envisage a star connection of the three statoric phases, so that a classical three phase 380V voltage supply can be used and only 220V are imposed by the phases of the motor.

The classical identification procedure for three phase induction motor is presented here [66]. After measuring the statoric resistance $R_s = 68.8\Omega$, two experiments are conducted in order to determine the parameters of the phaser equivalent circuit of the motor represented in figure D.1.

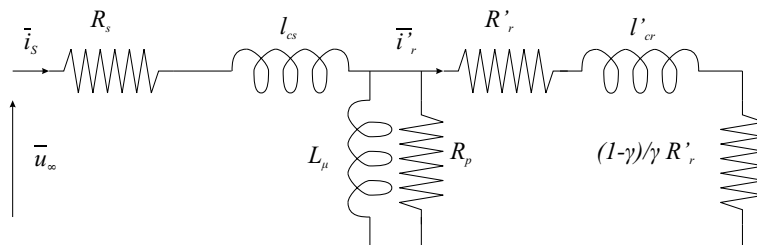


Figure D.1: Simplified equivalent circuit

Open shaft experiment The first experiment consists in feeding the motor with nominal voltages without mechanical load. In such situation, the motor almost reaches its synchronous speed and $\gamma \cong 0$. Assuming that l_{cs} is negligible with regards to L_μ , the equivalent circuit of figure D.1 can be simplified accordingly and only involves R_s , X_μ and R_p , as shown in figure D.2.

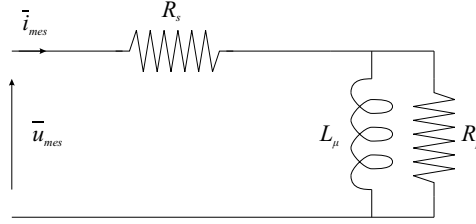


Figure D.2: Simplified equivalent circuit: open shaft experiment

We measure the power P_{mes} and the current i_{mes} absorbed by the motor as well as the supply voltage u_{mes} . The measured quantities are related by:

$$P_{mes} = 3 R_s i_{mes}^2 + P_{losses} + P_{mag} + P_{meca} \quad (D.1)$$

$$P_{losses} = \sqrt{3} (u_{mes} - R_s i_{mes}) i_{mes} \cos(\varphi) \quad (D.2)$$

$$X_\mu = \frac{u_{mes}/\sqrt{3}}{i_{mes} \sin(\varphi)} \quad (D.3)$$

$$R_p = \frac{u_{mes}/\sqrt{3}}{i_{mes} \cos(\varphi)} \quad (D.4)$$

In this first experiment, the mechanical power P_{meca} may be neglected. It is most of the time impossible to separate the magnetic losses P_{losses} and the magnetization power P_{mag} and we assume them to be equal, what leads to:

$$P_{losses} = \frac{1}{2} (P_{mes} - 3 R_s i_{mes}^2) \quad (D.5)$$

From all these relations and from our measurements, we obtain an estimation of $X_\mu = \omega_\infty L_\mu = 9.615 \cdot 10^2$ and $R_p = 1.197 \cdot 10^4$. This confirms that R_p can be neglected. Since the network frequency is of 50Hz, $\omega_\infty = 2\pi \cdot 50$, and the identified value for L_μ is:

$$L_\mu = 3.0636H \quad (D.6)$$

Locked rotor experiment In this second experiment, the rotor is locked and thus $\gamma = 1$. In this case, L_{mu} and R_p can be neglected and the equivalent circuit of figure D.1 can be simplified as in figure D.3, where $X_e = (l_{cs} + l'_{cr}) \omega_\infty$ and $R_e = R_s + R'_r$.

We measure the power P_{mes} and the current i_{mes} absorbed by the motor and the input voltage u_{mes} .

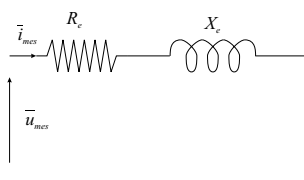


Figure D.3: Simplified equivalent circuit: locked rotor experiment

We can write the following relations:

$$P_{mes} = \sqrt{3} u_{mes} i_{mes} \cos(\varphi) \tag{D.7}$$

$$X_e = \frac{u_{mes}/\sqrt{3}}{i_{mes}} \sin(\varphi) \tag{D.8}$$

$$R_e = \frac{u_{mes}/\sqrt{3}}{i_{mes}} \cos(\varphi) \tag{D.9}$$

Our measurements led us to $X_e = 1.545 \cdot 10^2 \Omega$ and $R_e = 76.98 \Omega$, and we get $l_{cs} + l'_{cr} = 0.2458$ and $R'_r = 8.18 \Omega$. This confirms us that l_{cs} could be neglected in front of L_μ when doing the open shaft experiment. Let us point out that R'_r was obtained from the difference between R_e and R_s , that are both much larger than R'_r , the estimation of which is thus very approximate.

Parameters for our equivalent circuit In our modeling approach, the equivalent circuit of figure D.4 is used.

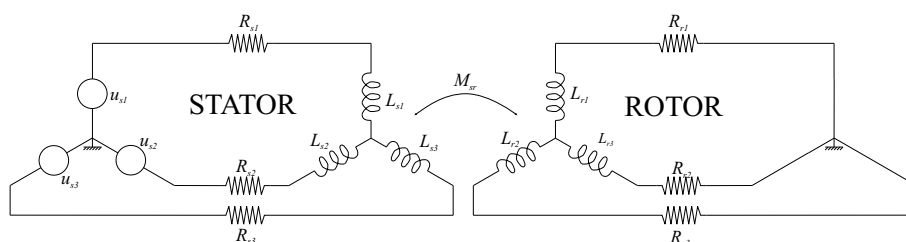


Figure D.4: Stator and rotor equivalent circuit of a three-phase actuator

As mentioned in section 6.3, under certain assumptions [25], it is possible to relate

the parameters appearing in figure D.4 to those from figure D.1:

$$\begin{aligned}
 l'_{cr} &= l_{cr} \\
 R'_r &= R_r \\
 X_f &= \omega_\infty (l_{cr} + l_{cs}) \\
 L_\mu &= \frac{3}{2} M_{sr} \\
 L_{cr} &= L_\mu + l_{cr} \\
 L_{cs} &= L_\mu + l_{cs} \\
 L_{cr} &= L_r - M_r = \frac{3}{2} L_r \\
 L_{cs} &= L_s - M_s = \frac{3}{2} L_s
 \end{aligned} \tag{D.10}$$

We thus obtain the following identified values for the model generated by Electran as:

$$\begin{aligned}
 R_s &= 68.8 \quad (\Omega) \\
 R_r &= 8.18 \quad (\Omega) \\
 L_s &= 2.204 \quad (H) \\
 L_r &= 2.204 \quad (H) \\
 M_s &= -1.102 \quad (H) \\
 M_r &= -1.102 \quad (H) \\
 M_{sr} &= 2.04 \quad (H)
 \end{aligned} \tag{D.11}$$

D.2 Electrical input file

```
% Description of the system
% -----
% Electromechanical Interaction
% Number of circuit
2
% Number of permanent magnets
0
% -----
% Circuit 1
% -----
% Number of elements =
9
% Element 1
0
U
% Element 2
1
R
% Element 3
2
L
% Element 4
0
U
% Element 5
4
R
% Element 6
5
L
% Element 7
0
U
% Element 8
7
R
% Element 9
8
L
%
% Number of loops
2
% Loop 1
3
9
```

```

% Loop 2
6
9
% Internal Mutual Influence
% Number
3
% Influence 1
L
3
6
% Influence 2
L
3
9
% Influence 3
L
6
9
% _____
% Circuit 2
% _____
% Number of elements =
9
% Element 1
0
U
% Element 2
1
R
% Element 3
2
L
% Element 4
0
U
% Element 5
4
R
% Element 6
5
L
% Element 7
0
U
% Element 8
7
R

```

```
% Element 9
8
L
% % Number of loops
2
% Loop 1
3
9
% Loop 2
6
9
% Internal Mutual Influence
% Number
3
% Influence 1
L
3
6
% Influence 2
L
3
9
% Influence 3
L
6
9
% _____
% External Mutual Influence
% _____
9
% Influence 1
L
3
3
% Influence 2
L
6
6
% Influence 3
L
9
9
% Influence 4
L
3
6
```

% Influence 5

L

3

9

% Influence 6

L

6

3

% Influence 7

L

6

9

% Influence 8

L

9

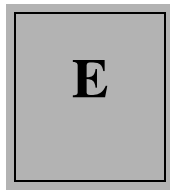
3

% Influence 9

L

9

6



Electromagnetic Speaker Model

E.1 Electrical Parameters

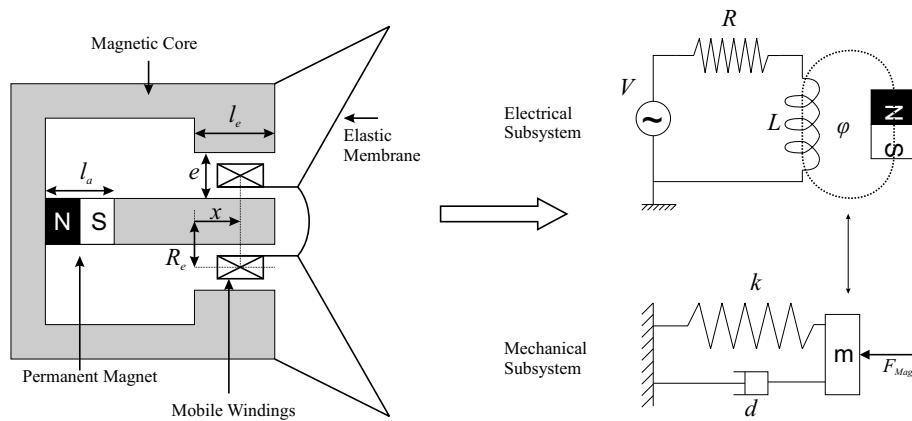


Figure E.1: Electromagnetic speaker and the corresponding subsystem models

As shown in figure E.1, the equivalent circuit of the windings involves an inductance L coupled with a permanent magnet and a resistor $R = 1000\Omega$. It can be shown that L and the flux φ through the inductance due to the permanent magnet are given

by:

$$L = \mu_0 n^2 2\pi R_e (l_e - x) \frac{\left(S_a \frac{e}{l_a} + 2\pi R_e x\right)}{\left(S_a \frac{e}{l_a} + 2\pi R_e l_e\right)} \quad (\text{E.1})$$

$$\varphi = -\frac{n B_{a0} S_a}{l_e} \left(1 - \frac{\mu_a}{\mu_a + \mu_0 \frac{l_a}{e} \frac{2\pi R_e l_e}{S_a}}\right) (l_e - x) \quad (\text{E.2})$$

where, x represents the position of the coil along the magnetic core, as shown in figure E.1. The parameters appearing in these expressions are:

- $l_a = 0.05m$, the length of the magnet
- $R_e = 0.06m$, the mean radius of the air-gap
- $e = 0.005m$, the thickness of the air-gap
- $l_e = 0.05m$, the axial length of the air-gap
- $S_a = \pi * \left(R_e - \frac{e}{2}\right)^2$, the section of the permanent magnet
- $B_{a0} = 0.8$, the magnetic field produced by the permanent magnet
- $\mu_0 = 4 \cdot 10^{-7}$, the permittivity of the air-gap
- $\mu_a = 1.5 * \mu_0$, the permittivity of the magnetic core
- $n = 100$, the number of loops in the windings

The mechanical submodel simply consists in a spring-mass system with mass $m = 0.005kg$, an equivalent stiffness $k = 10^6 N/m$ and a viscous friction coefficient $d = 0.01 Ns/m$.

E.2 Electrical input file

```
% Description of the system
% -----
% Electromechanical Interaction
1
% Number of circuit
1
% Number of permanent magnets
1
% -----
% Circuit 1
% -----
% Number of elements
3
% Element 1
0
U
% Element 2
1
R
% Element 3
2
L
%
% Number of loops
1
% Loop 1
3
0
% Internal Mutual Influence
% Number
0
```

E.3 Electrical output file

```

void Speaker_Dyn(double t, double *w, double *sm, double *smd) {
teta = sm[1];
tetap = smd[1];
Speaker_SU(t, w, sm, smd);
Speaker_SI(t, w, sm, smd);
Speaker_R(t, w, sm, smd);
Speaker_L(t, w, sm, smd);
Speaker_DL(t, w, sm, smd);
Speaker_Fl(t, w, sm, smd);
Speaker_DFl(t, w, sm, smd);
inv_AMc[2][2] = 1.000/L[2];
niy5 = Uin[1]-R[1]*w[1];
niy8 = w[1]*(DLM[1]*inv_AMc[2][2]*tetap)/inv_AMc[2][2];
niy9 = -Iin[1]/inv_AMc[2][2];
nix9 = -inv_AMc[2][2]*niy9;
nix8 = niy8+nix9*(DL[2]*inv_AMc[2][2]*tetap)/inv_AMc[2][2];
nix4 = Uin[1]-niy5;
nix3 = -nix4-niy5;
y[3] = nix3;
y[4] = nix4;
y[5] = niy5;
y[6] = nix9;
y[7] = -nix8;
y[8] = nix8;
y[9] = nix9;
wdy1 = y[5]-DL[1]*tetap*w[1];
wdx1 = wdy1/L[1];
wd[1] = wdx1;
Cmot = 0.500*(ni[9]*(DLM[1]*w[1]+DL[2]*ni[9])+w[1]*(DLM[1]*ni[9]+DL[1]*w[1]));
}

```


Bibliography

- [1] R.R. Allen, S. Dubowsky, "Mechanisms as components of dynamic systems: a bond graph approach", *Journal of Engineering for Industry*, Vol. 99-1, 1977, pp. 104-111
- [2] N. Biggs, E. Lloyd, R. Wilson, "Graph Theory: 1736-1936", Oxford University Press, 1976
- [3] R. Boite, J. Neiryneck "Théorie des réseaux de Kirchhoff", Vol 4, *Traité d'électricité Lausanne: Presses polytechniques romandes*, 1989
- [4] L. Bonderson, "Vector Bond Graphs Applied to One-Dimensional Distributed Systems", *Journal of Dynamic Systems, Measurement, and Control*, March 1975, pp.75-82
- [5] A.M. Bos, J.L. Tiernego, "Formula Manipulation in the Bond Graph Modelling and Simulation Of Large Mechanical Systems", *Journal of the Franklin Institute*, Vol. 319-1/2, 1985, pp. 51-65
- [6] P.C. Breedveld, "Proposition for an Unambiguous Vector Bond Graph Notation", *Journal of Dynamic Systems, Measurement, and Control*, September 1982, Vol. 104, pp. 267-270
- [7] P.C. Breedveld, "Multibond Graph Elements in Physical Systems Theory", *Journal of the Franklin Institute*, Vol. 319-1/2, 1985, pp. 1-36
- [8] F. Breitenecker, M. Lingl, "Comparison 3 - Matlab/Simulink Dynamic Function Approach", *EUROSIM - Simulation News Europe*, Number 22, March 1998
- [9] A.P.J. Breunese and J.F. Broenink. "Modeling mechatronic systems using SIDOPS+ language." *Proceedings of ICBGM'97, 3rd International Conference on Bond Graph Modeling and Simulation*, Phoenix, Arizona, vol. 19 of Simulation Series, pp 301-306, 1997
- [10] F.T. Brown, "Energy-Based Modeling and Quasi Coordinates", *Journal of Dynamic Systems, Measurement, and Control*, March 1981, Vol. 103, pp. 5-13

- [11] H. Elmqvist, "A structured Model language for Large Continuous Systems". PhD thesis, Lund Institute of Technology, Sweden, Department of Automatic Control, 1978
- [12] W. Favre, "Contribution à la Représentation Bond Graph des Systèmes Mécaniques Multicorps", PhD. Thesis, INSA de Lyon, 1997
- [13] W. Favre, S. Scavarda, "Bond Graph Representation of Multibody Systems with Kinematic Loops", *Journal of the Franklin Institute*, Vol. 335B-4, 1998, pp. 643-660, Ecole Doctorale des Sciences pour l'Ingénieur de Lyon, 1997
- [14] P. Fiset, J.C. Samin, "Lateral Dynamics of a Light Railway Vehicle with Independent Wheels." *Supplement to vehicle System Dynamics*, 20, pp. 157-171, 1992
- [15] P. Fiset and J.C. Samin, "A new wheel/rail contact model for independent wheels", *Archive of Applied Mechanics*, Springer-Verlag, vol. 64, pp. 180-191, 1994
- [16] P. Fiset, "Génération symbolique des équations du mouvement de systèmes multicorps et application dans le domaine ferroviaire", PhD Thesis, Université catholique de Louvain, 1994
- [17] P. Fiset, K. Lipinski, J.C. Samin, "Dynamic Behaviour Comparison Between Bogies: Rigid or Articulated Frame, Wheelset or Independent Wheels", *Supplement to Vehicle System Dynamics*, 25, pp. 152-174, 1995
- [18] P. Fiset, B. Vaneghem, "Numerical integration of multibody system dynamic equations using the coordinate partitioning method in an implicit Newmark scheme", *Comput. Methods Appl. Mech. Engrg.* 135, pp. 85-105, 1996
- [19] P. Fiset, J.C. Samin, "Symbolic Generation of Large Multibody System Dynamic Equations using a New Semi-Explicit Newton/Euler Recursive Scheme", *Archive of Applied Mechanics* 66, pp.187-199, 1996
- [20] P. Fiset, T. Postiau, L. Sass and J.C. Samin, "Fully Symbolic Generation of Complex Multibody Models", *Mechanics of Structure and Machines*, 30(1), 31-82. Marcel Dekker Inc., 2002
- [21] E. Forsthuber, "Comparison 11 - Dymola: Classical mechanical approach, Automatic-symbolical and numerical inversion", *SimulationNews Europe*, Number 23, July 1998
- [22] P. Fritzson, L. Viklund, D. Fritzson and J. Herber, "High-level mathematical modeling and programming." *IEEE Software*, 1995
- [23] M. Geradin, A. Cardona, "Numerical integration of second-order differential-algebraic systems in flexible mechanism dynamics", in: M. Pereira and J. Ambrosio, eds., *Computer-Aided Analysis of Rigid and Flexible Mechanical Systems*, NATO ASI Series, Applied Sciences, Vol. 268, Kluwer, pp. 233-284, 1993

- [24] H. Goldstein, "Classical Mechanics", Addison-Wesley Publishing Company, 1950, 1980 (ISBN 0-201-02918-9)
- [25] D. Grenier, F. Labrique, H. Buyse, E. Matagne, "Électromécanique: Convertisseurs d'énergie et actionneurs", Dunod, Paris, 2001
- [26] V. Hadwich, F. Pfeiffer, "The principle of virtual work in mechanical and electromechanical systems", *Archive of Applied Mechanics* 65 (1995) 390-400 Springer-Verlag 1995
- [27] R.L. Huston, Y. Wang, M. Pereira, J. Ambrosio, "Flexibility Effects in Multibody Systems in Computer-Aided Analysis of Rigid and Flexible Mechanical Systems", J. (eds.), NATO ASI Series, Applied Sciences, Kluwer Academic Publishers, Dordrecht, 1993, pp. 351-376
- [28] M.E. Ingram, G.Y. Masada, "The Extended Bond Graph Notation", *Journal of Dynamic Systems, Measurement, and Control*, March 1991, Vol. 113, pp. 113-117
- [29] J. J. Kalker, "A strip theory for rolling with slip and spin", *Proceedings, Kon. Ned. Akad. Wet., Amsterdam, B70*, pp. 10-62
- [30] J. J. Kalker, "On the rolling contact of two elastic bodies in the presence of dry friction", *Dotoral Thesis, Technical University of Delft, The Netherlands*, 1967
- [31] T. R. Kane and D. A. Levinson, "Dynamics: Theory and Applications", Mc Graw-Hill, 1985
- [32] D. Karnopp, "Power-Conserving Transformations: Physical Interpretations and Applications using Bond Graphs", *Journal of the Franklin Institute*, Vol. 288-3, 1969, pp. 175-201
- [33] D. Karnopp, "The Energetic Structure of Multibody Dynamic Systems", *Journal of the Franklin Institute*, Vol. 306-2, 1978, pp. 165-181
- [34] D. Karnopp, "An Approach to Derivative Causality in Bond Graph Models of Mechanical Systems", *Journal of the Franklin Institute*, Vol. 329-1, 1992, pp. 65-75
- [35] D. Karnopp, Donald L. Margolis, Ronald C. Rosenberg, "System Dynamics: Modeling and Simulation of Mechatronic Systems (Third Edition)", John Wiley & Sons Inc., 2000
- [36] H. Koenig, Y. Tokad, H. Kesavan, "Analysis of Discrete Physical Systems", Mc Graw-Hill, 1967
- [37] W. Kortum, R.S. Sharp (eds), "Multibody Computer Codes to Vehicle System Dynamics", *Supplement to Vehicle System Dynamics*, vol. 22, Swets & Zeitlinger, 1993.

- [38] R. Kubler, W. Schielen, "Modular Simulation in Multibody System Dynamics", *Multibody System Dynamics*, vol. 4, pp. 107-127, 2000
- [39] W. Leonard, "Control of Electrical Drives", Springer-Verlag, 1985
- [40] J.-Y.-S. Luh, N.-W. Walker, and R.-P.-C. Paul, "On-line computational scheme for mechanical manipulators", *Journal of Dynamics Systems, Measurements and Control* 102, 6976, 1980.
- [41] P. Maes, J.C. Samin, P.Y. Willems, "Autodyn - Multibody System Handbook", W. Schielen Ed., Springer-Verlag, pp. 225-245, 1989
- [42] P. Maisser, O. Enge, H. Freudenberg, G. Kielo, "Electromechanical Interactions in Multibody Systems Containing Electromechanical Drives", *Journal of Multibody System Dynamics*, Vol. 1, n° 3, 1997, pp. 281-302
- [43] J.C. Mandojana, K. J. Herman and R.E. Zulinski, "A Discrete/Continuous Time-Domain Analysis of a Generalized Class E Amplifier", *IEEE Transactions on Circuits and Systems*, Vol. 37, No.8, August 1990, pp. 1057-1060
- [44] J. McPhee, "Automatic Generation of Motion Equations for Planar Mechanical Systems Using the New Set of "Branch Coordinates"", *Mech. Mach. Theory*, 33, 1998, pp. 805-823
- [45] J. McPhee, "Multibody dynamics 2003", in press, Springer-Verlag, 2003
- [46] J. Meisel, "Principles of Electromechanical-energy conversion", McGraw-Hill Inc., 1966
- [47] B. Muegge, "Graph-Theoretic Modelling and Simulation of Planar Mechatronic Systems", Master's Thesis, University of Waterloo, Canada, 1996
- [48] H.M. Paynter, "Analysis and design of engineering systems", The MIT Press, Cambridge, Mass, 1961
- [49] T. Postiau, L. Sass, P. Fiset, J.C. Samin, "High-Performance Multibody Models of Road Vehicles: Fully Symbolic Implementation and Parallel Computation", *Suppl. To Vehicle System Dynamics*, 35 (2001), pp. 57-84
- [50] B. Rada, M. Wibmer, "C3 Generalized Class-E Amplifier - Matlab Overall-model approach", *Simulation News Europe*, Number 29-30, December 2000
- [51] P. Roe, "Network and systems", Addison-Wesley, 1966
- [52] R.C. Rosenberg, "Multiport Models in Mechanics", *Journal of Dynamic Systems, Measurement, and Control*, September 1972, pp. 206-212
- [53] D. Rowell and D.N. Wormley, "System Dynamics: An introduction", Prentice Hall, New Jersey, 1994

- [54] J.C. Samin, "Dynamique des corps déformables en rotation", PhD Thesis, Université Catholique de Louvain, 1974
- [55] J.C. Samin, P.Y. Willems, "Multibody Formalism Applied to Non-Conventional Railway Systems", Springer-Verlag, IUTAM/IFTToMM Symposium, Italy, 1985
- [56] J.C. Samin, P. Fiset, "Symbolic Modeling of Multibody Systems", Kluwer Academic Publishers, 2003
- [57] P. Sahlin, A. Bring and E.F. Sowell, "The Neutral Model Format for Building Simulation, version 3.02." Technical Report, The royal Institute of TEchnology, Stockholm, Sweden, 1996
- [58] L. Sass, P. Fiset, D. Grenier, "Modeling of mechatronic systems - Study of the actuation of a swinging barrier", Mechatronics 2002, The 8th Mechatronics Forum International Conference, University of Twente, Enschede, Netherlands, 24th - 26th June 2002
- [59] L.Sass, D. Telteu, D. Grenier, P. Fiset, "Unified symbolic modeling of mechatronic systems - Implementation and applications", Electrimacs 2002, Montréal, Quebec, Canada.
- [60] L. Sass, J. McPhee, C. Schmitke, P. Fiset and D. Grenier, "A comparison of different methods for modeling multibody systems with electrical drives", submitted to Multibody System Dynamics, 2003
- [61] J. Scheidl, "Comaprison 11 - Matlab/Simulink: Numerical Inversion/Hybrid Approach", Simulation NEws Europe, Number 25, March 1999
- [62] J.M.A. Sherpen, B. Klassens, L. Ballini, "Lagrangian Modelling and Control of Switching Networks with integrated coupled magnetics", Proc. 39th IEEE Conf. Des. Contr., Sydney, Australia, December 2000, pp. 4054-4059
- [63] M. Scherrer, J. McPhee, "Dynamic Modelling of Electromechanical Multibody Systems", Multibody System Dynamics, Vol. 9, 2003, pp. 87-115
- [64] K. Schlacher, A. Kugi, R. Scheidl, "Tensor Analysis Based Symbolic Computation for Mechatronic Systems", Mathematics and Computers in Simulation 46, 1998, pp. 517-525
- [65] R. Schwertassek and W. Rulka, "Aspects of Efficient and Reliable Multibody Systems Simulation", Real-Time Integration Methods for Mechanical System Simualtions, NATO ASI Series, Series F, E.-J. Haug and R.-C. Deyo (Eds.), Berlin, 1989, Vol. 69, pp. 55-96
- [66] G. Segulier, F. Notelet, "Electrotechnique Industrielle", Technique et Documentation, Ed. Lavoisier 1994
- [67] A. A. Shabana, "Dynamics of Multibody Systems", John Wiley & Sons, 1989

- [68] P. Shi, J. McPhee, "Dynamic of Flexible Multibody Systems using Virtual Work and Linear Graph Theory", *Multibody System Dynamics*, Vol. 4, 2000, pp. 355-381
- [69] H. Sira-Ramirez, R.A. Perez-Moreno, R. Ortega and M. Garcia-Esteban, "Passivity-Based Controllers for the Stabilization of DC-to-DC Power Converters", *Automatica*, vol. 33-4, pp. 499-513, Pergamon, 1997
- [70] G. Strang, "Linear Algebra and its Applications (third edition)". Fort Worth, Hacourt Brace Jovanovich Int. Edition, 1986.
- [71] D. Telteu, L. Sass, D. Grenier, P. Fiset, F. Labrique, "Modélisation et simulation des convertisseurs électroniques de puissance par analogie avec les systèmes mécaniques", *Revue Internationale de Génie Electrique*, vol. 6 - n3-4, pp. 429-456, 2003
- [72] M. Tiller, "Introduction to Physical Modeling with Modelica". Kluwer Academic Publishers, 2001
- [73] A.J. Van der Schaft, " L_2 -gain and passivity techniques in non-linear control", Springer-Verlag, London, 2000
- [74] N. Viertl, "Comparison 3 - Maple, Closed Model/Numerical Simulation", *Simulation News Europe*, Number 28, April 2000
- [75] R.A. Wehage, E.J. Haug (1982), "Generalized Coordinate Partitioning for Dimension Reduction in Analysis of Constrained Dynamic Systems" *J. Mech. Design*, vol.134, pp 247-255
- [76] P.-Y. Willems, "Introduction à la Mécanique", Masson Ed., Paris, 1979
- [77] J. Wittenburg, "Dynamics of Systems of Rigid Bodies", Teubner Ed., Stuttgart, 1977
- [78] J. Wittenburg, "Nonlinear Equations of Motion for Arbitrary Systems of Interconnected Rigid Bodies", Springer-Verlag, IUTAM Symposium, Germany, 1977
- [79] A. Zeid, C-H. Chung, "Bond Graph Modeling of Multibody Systems: a Library of Three-dimensional Joints", *Journal of the Franklin Institute*, Vol.329-4, 1992, pp. 605-636