Graph-Theoretic Sensitivity Analysis of Dynamic Systems

by

Joydeep Banerjee

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2013

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The main focus of this research is to use graph-theoretic formulations to develop an automated algorithm for the generation of sensitivity equations. The idea is to combine the benefits of direct differentiation with that of graph-theoretic formulation. The primary deliverable of this work is the developed software module which can derive the system equations and the sensitivity equations directly from the linear graph of the system.

Sensitivity analysis refers to the study of changes in system behaviour brought about by the changes in model parameters. Due to the rapid increase in the sizes and complexities of the models being analyzed, it is important to extend the capabilities of the current tools of sensitivity analysis, and an automated, efficient, and accurate method for the generation of sensitivity equations is highly desirable.

In this work, a graph-theoretic algorithm is developed to generate the sensitivity equations. In the current implementation, the proposed algorithm uses direct differentiation to generate sensitivity equations at the component level and graph-theoretic methods to assemble the equation fragments to form the sensitivity equations.

This way certain amount of control can be established over the size and complexity of the generated sensitivity equations. The implementation of the algorithm is based on a commercial software package `MapleSim[Multibody]` and can generate governing and sensitivity equations for multibody models created in MapleSim.

In this thesis, the algorithm is tested on various mechanical, hydraulic, electro-chemical, multibody, and multi-domain systems. The generated sensitivity information are used to perform design optimization and parametric importance studies. The sensitivity results are validated using finite difference formulations.

The results demonstrate that graph-theoretic sensitivity analysis is an automated, accurate, algorithmic method of generation for sensitivity equations, which enables the user to have some control over the form and complexity of the generated equations. The results show that the graph-theoretic method is more efficient than the finite difference approach. It is also demonstrated that the efficiency of the generated equations are at par or better than the equation obtained by direct differentiation. The developed software module can be used to generate sensitivity equations for multibody systems which can subsequently be used for design optimization, parameter identification, model simplification, and other related applications.

Acknowledgements

I would like to express my
gratitude to my supervisor Professor John J. McPhee
for the unrelenting support and consideration, valuable guidance,
and the academic environment he provided during the course of this research.

I would like to thank
Dr. Chad Schmitke for his constant encouragements,
valuable guidance, and his help regarding the software implementation.
Without his support, this implementation would have been almost impossible.

I wish to thank
the other members of my Ph.D. examination committee,
Prof. Gordon Savage, Prof. Brian Ingalls, and Prof. William L. Cleghorn,
for their valuable time, attention, and efforts toward the improvement of this thesis.

I am grateful to
my colleagues at the Motion Research Group,
for their efforts that made working in the lab a wonderfully rewarding experience.

I would like to mention
the efforts from my parents throughout all these years,
without which I would not have been the person I am today.

I am grateful to
my uncle Prof. Subir Bandyopadhyay and aunt Rita,
cousins Dr. Avik Banerjee and Dr. Rupa Banerjee,
and their families for their help and support during the past five years.

Finally, I would like to acknowledge
the love, support, and care I got from
my wife Indira and our two baby sons Padmanavo and Ishaan,
that made me feel alive and kept me going through the stressful times.

Acknowledgements

Special mention

I would also like to thank the professors at the Department of Mechanical Engineering, Indian Institute of Technology, Kharagpur, for teaching me various aspects of dynamics and system modelling and enduring me and my countless queries for four years.

Prof. A. Dasgupta, Prof. K. Bhattacharyya, Prof. A. K. Samantaray, Prof. S. K. Pal, and Prof. G. Chakraborty

They taught me various aspects of dynamics of rigid bodies. It helped me extensively in the context of my research.

Prof. A. Mukherjee, Prof. R. Karmakar, and Prof. R. Bhattacharya

I learned Bond-graph because of their efforts. It helped me a lot while learning linear graph for system modelling.

Prof. S. K. Roy Chowdhury and Prof. B. Maiti

They showed me the values of experimental methods, which I found to be invaluable while working here.

Prof. Kumar Ray

He showed me the values of having priorities in life. It has helped me a lot while balancing my responsibilities as a young father of two and a research assistant in the last few years.

This thesis would not have happened if it was not for their efforts.

I would also like to express my gratitude to Mr. Amulyadhan Bar, headmaster (retired) of the Scottish Church Collegiate School - Kolkata, Dr. Nirjhar Bar, and their family members Mrs. Sucheta Bar and Dr. Ananya Bar for their constant moral support and unrelenting faith in my abilities.

Dedication

To the memory of my grandparents
Sushobhan Banerjee and Nandini Banerjee

Contents

List of Tables

List of Figures

ॐ यदक्षरन्ग परिभ्रष्ठन्ग मात्राहीनञ्च यद्भबेत ।
पुर्णन्ग भबतु तत् सर्ब्बन्ग तत्प्रसादाज्जनार्दन ॥

Translation
The inadequacies of this offering
may all be fulfilled by the blessings of Lord Vishnu.

Phonetic Transliteration
Om...Yadaksharang paribhrastang maatraahinancha yadbhabet,
Purnang bhabatu tat sarbbang tatprasaadaat janaardana.

Chapter 1

Introduction

Sensitivity analysis refers to the study of changes in system behaviour brought in by the changes in entities inherent to the system. Mathematically, it becomes a problem of finding the derivative of a function with respect to the system parameters. To illustrate the importance of sensitivity analysis, it is worthwhile to look into the range of engineering applications that use sensitivity analysis.

Design and optimization of physical systems: From the simple mechanism that closes the door to the complex spacecraft that sends man to the moon, proper design makes the difference between a complete failure and "a giant leap for mankind". Detailed system analysis and design computations are expensive in terms of human time and resources, but without these steps, for most practical systems of the modern age, it is not possible to meet the ever increasing demands of performance, economic and safety requirements. For these reasons, optimization processes are almost always required for most design tasks. Efficient optimization routines require gradient information, which are essentially sensitivity data.

Model simplification: Model simplification is an attempt to capture the important system behaviours by using a bare minimum of model entities and ignoring the rest of the features. It is important for analysis, control design and simulation of complex large scale systems. Essentially the model simplification problem can be described as the problem of finding out which feature is important for the intended purpose. Sensitivity analysis provides a systematic way to evaluate the effect of each entity on the desired behaviour.

Robust design: To achieve robust design for dynamic systems, one has to minimize the deviation of the system from its intended behaviour due to changes in the varying parameters. The problem essentially becomes a minimization problem where gradient information or sensitivity data greatly improves efficiency and accuracy.

Parameter identification: Parameter identification finds the parameters for a given system model to make it match a measured behaviour as closely as possible. Mathematically it is a problem of minimization of the difference between the model prediction and the measured data. Sensitivity data enables the implementation of efficient techniques to deal with this problem.

Process sensitivity: Efficient management of any process requires knowledge of the key issues that affect the output of the process. For processes where a mathematical model is available, sensitivity studies can provide invaluable insight about the critical parameters and optimization strategies.

Optimal control: Sensitivity analysis is also used while designing optimal controls for dynamic systems, to identify control parameters and to study the behaviour or the system with changes in control parameters.

In light of the above discussion, it is quite clear how important it is to have an efficient and accurate algorithm for sensitivity analysis. Unfortunately, for practical systems, the sensitivity study is a very complicated problem. To explain the challenges of sensitivity analysis, at this point it is necessary to discuss the process of sensitivity analysis of dynamic systems in a more mathematical form.

1.1 Background

Theoretical background relevant for the current research is presented in this section. To cover all the relevant aspects, this section is divided into three unique parts. In the first part, the graph-theoretic method of system modelling is presented. Next, different types of governing equations that are routinely encountered, are discussed. In the third and final subsection, the topic of sensitivity analysis of dynamic systems is reviewed.

1.1.1 Graph-theoretic modelling of dynamic systems

Graph-theoretic modelling method is known for its unified approach toward system modelling. It's a unique way of modelling systems that uses directed linear graphs to represent system topologies and constitutive equations to capture the individual characteristics of the system components.

To describe the process of graph-theoretic modelling of a system, an example of a simple resistive circuit is shown in figure 1.1. The system shown in figure 1.1 contains linear resistors and a time varying voltage source. These components are represented by the edges of the corresponding linear graph shown in the same figure. The nodes of the graph represent junction points for the electrical circuit. The arrows in this linear graph are used to designate a reference direction for measurement.

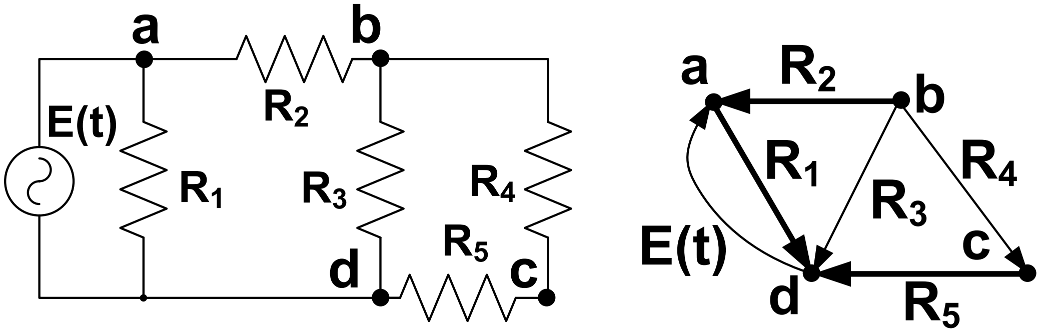


Figure 1.1: Resistive circuit and the corresponding linear graph

Every single edge of a linear graph is associated with a pair of complementary sets of measurements. These are known as the through variables (τ) and across variables (α). Through variables are quantities which are measured by instruments placed in series with the physical component represented by the edge, and across variables are quantities which are measured by instruments placed in parallel to the physical components associated with the edge. Different physical domains involve different through and across variables. For an electrical system, the appropriate through and across variables are the currents and voltage drops, respectively.

To explain the process by which the topology of the system is captured by the linear graph, a few key terms need to be defined. Assuming a linear graph G with e number of edges and n number of nodes, the following terms are defined.

A circuit is a sub-graph of G where every pair of nodes has exactly two distinct paths between them. There can be many circuits in a single linear graph.

A tree of a linear graph G is defined as a connected sub-graph of G that contains all the nodes of G and has only one unique path between any two nodes. Edges of the tree are known as branches. A tree in G has exactly $w = n - 1$ branches.

A cotree is the part of the graph G which remains after removing the tree. All the edges of a cotree are known as chords. In G there are $u = e - n + 1$ chords in the cotree.

A cutset of a linear graph G is a subset of edges that, when removed, divides G into two parts. Also, no subset of this cutset can do the same.

A fundamental cutset (f-cutset) is a cutset that has one branch from the linear graph G and a unique set of chords from G. There are w f-cutsets in G. Similarly, a fundamental circuit is a circuit with a single chord and a unique set of branches. There are u f-circuits in G.

From the linear graph of the system it is possible to derive three distinct types of equations. The first type of equation is known as the *cutset equations.* These equations captures the relationships between individual through variables. Unique *fundamental cutset equations* can be generated from these equations by selecting different tree branches.

For the system shown in figure 1.1, the edges corresponding to the resistors R$_1$, R$_2$, and R$_5$ are chosen as the tree and are shown in bold lines. These edges are used to derive the fundamental cutset equations. The resulting equations are expressed in matrix form as

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 1 & -1 \\ 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \end{array}\right] \left\{ \begin{array}{cccccc} i_{R_1} & i_{R_2} & i_{R_5} & i_{R_4} & i_{R_3} & i_E \end{array} \right\}^T = 0 \tag{1.1}$$

The second type of equations are called the circuit equations. These capture the relationships between the individual across variables. Unique sets of *fundamental circuit equations* can be generated from these equations for different choices of branches.
The fundamental circuit equations for the tree selection shown in figure 1.1 are

$$\begin{bmatrix} 1 & -1 & -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \left\{ \begin{array}{cccccc} v_{R_1} & v_{R_2} & v_{R_5} & v_{R_4} & v_{R_3} & v_E \end{array} \right\}^T = 0 \tag{1.2}$$

The third type of equations are used to capture the physical characteristics of the components that are represented by the edges. These are known as the terminal or constitutive equations. The equations are associated with each of the edges of the graph and describe the relationships between the through and across quantities. For the system shown in figure 1.1 there are six terminal equations as shown in equation (1.3).

$$\begin{array}{lll} v_{R_1} = R_1 i_{R_1} & v_{R_2} = R_2 i_{R_2} & v_{R_3} = R_3 i_{R_3} \\ v_{R_4} = R_4 i_{R_4} & v_{R_5} = R_5 i_{R_5} & v_E = E(t) \end{array} \tag{1.3}$$

The twelve scalar equations (1.2), (1.1), and (1.3) can be solved to obtain the six across quantities and six through quantities of the system. As such, these equations can be considered as the equations governing the response of the system shown in figure 1.1 generated from the corresponding linear graph.

It is also possible to reduce the number of governing equations using the graph-theoretic method, employing what is known as the branch-chord formulation. Using equation (1.2) and equation (1.1), it is possible to eliminate the branch through quantities and chord across quantities from the governing equations. The final equations obtained hereby are always expressed in terms of the cotree through variables and the tree across variables. This reduces the number of governing equations dramatically. For example, if branch-chord formulation is used for the system shown in figure 1.1, the governing equations will be generated in terms of three through variables from the chords and three across variables from the branches, which would bring down the total number of unknowns from twelve to six.

Depending on the nature of the component and the configuration of the system, it can also be possible to make a judicial choice of the tree to achieve further reduction in the number of equations.

Graph-theoretic modelling method offers flexibility in the process of equation generation. It allows users to generate symbolic governing equations using different coordinate systems and with varying levels of complexities while avoiding uncontrolled increase in the size of the symbolic expressions. Further details about graph-theoretic modelling methods is presented in the next chapter along with a brief description of available literature and research results on the subject.

1.1.2 Governing equations of multibody systems

The process of analytical sensitivity analysis starts at the set governing equations of the model. Depending on the nature of the system being investigated, the structure of the governing equations can be different and different approaches must be taken for subsequent analysis.

Kinematic systems

These systems are governed by algebraic equations and as a result are considerably easier to solve. They are often useful during inverse dynamic analysis and motion analysis for complex mechanisms. As an example we can consider the case of a driven slider-crank mechanism.

Figure 1.2 shows a kinematically driven slider-crank mechanism where the motion of the crank is specified as a function of time.

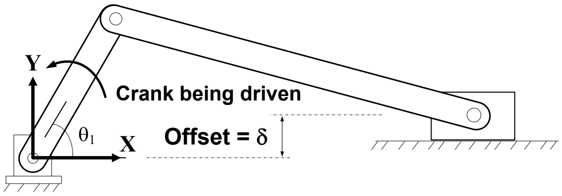


Figure 1.2: Kinematically driven slider-crank mechanism

The scope of kinematic analysis is to determine the position and orientation of each of the parts of the mechanism as a function of time. There are three rigid bodies in this mechanism, which are connected to each other through various joints as shown in figure 1.3 .

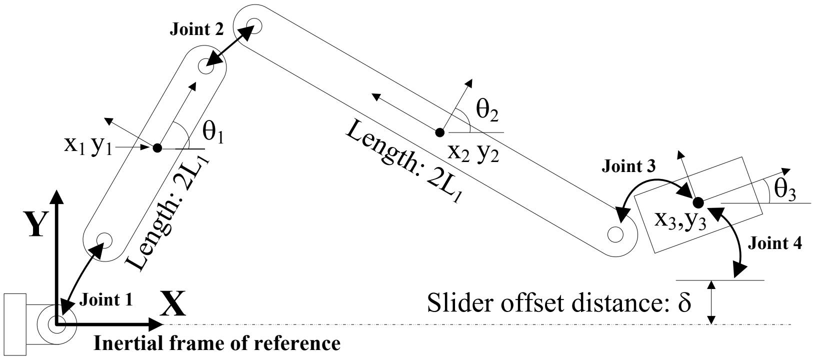


Figure 1.3: Selection of absolute coordinates

If we choose to use absolute coordinates for this system, we obtain the vector of generalized coordinates that describes the position and orientation of the system as

$$\mathbf{q} = [x_1, y_1, \theta_1, x_2, y_2, \theta_2, x_3, y_3, \theta_3]^{\mathrm{T}} \tag{1.4}$$

The joints in the system impose conditions on these generalized coordinates. These conditions give rise to algebraic constraint equations. Depending on the source of the constraint equations, they are called either kinematic constraints, which arise due to physical restrictions imposed by the joints, or driving constraints, which arise due to enforced motion on the system. For the slider-crank mechanism shown in figure 1.2 and figure 1.3, the driving constraint is simply the temporal variation of the angle of the crank.

$$\mathbf{\Phi}^{Driven}(q, t) = [\theta_1 - g_0(t)] = 0 \tag{1.5}$$

The kinematic constraint equations for this system can be derived by considering the physical restrictions the joints enforce on the system.

$$\mathbf{\Phi}^{Kinematic} = \begin{bmatrix} x_1 - L_1 \cos\theta_1 \\ y_1 - L_1 \sin\theta_1 \\ x_1 - x_2 + L_1 \cos\theta_1 + L_2 \sin\theta_2 \\ y_1 - y_2 + L_1\ \sin\theta_1 - L_2 \cos\theta_2 \\ x_2 - x_3 + L_2 \sin\theta_2 \\ y_2 - y_3 - L_2 \cos\theta_2 \\ y_3 - \delta \\ \theta_3 \end{bmatrix} = 0 \tag{1.6}$$

Together these constraints form a set of algebraic equations that governs the system behavior. It can be represented by the constraint vector $\mathbf{\Phi}$.

$$\mathbf{\Phi}\,(q,t) = \begin{bmatrix} \mathbf{\Phi}^{Kinematic}\,(q,t) \\ \mathbf{\Phi}^{Driven}\,(q,t) \end{bmatrix} = 0 \tag{1.7}$$

To evaluate the velocity and acceleration of the system, the relevant equations are obtained by differentiating equation (1.7) with respect to time as shown in equation (1.8).

$$\begin{aligned} \mathbf{\Phi}_{\mathbf{q}}\dot{\mathbf{q}} &= -\mathbf{\Phi}_t = \boldsymbol{\nu} \\ \mathbf{\Phi}_{\mathbf{q}}\ddot{\mathbf{q}} &= -(\mathbf{\Phi}_{\mathbf{q}}\dot{\mathbf{q}})_{\mathbf{q}}\dot{\mathbf{q}} - 2\mathbf{\Phi}_{\mathbf{q}t}\dot{\mathbf{q}} - \mathbf{\Phi}_t\, t = \boldsymbol{\gamma} \end{aligned} \tag{1.8}$$

In equation (1.8) the subscripts refer to differentiation operation. $\mathbf{\Phi}_{\mathbf{q}}$ is the Jacobian matrix of the constraint vector with respect to the generalized coordinates, also ν and γ are used to denote the right hand sides of the velocity and acceleration equations respectively. By solving equations (1.7) and (1.8) the complete kinematic analysis of the system can be performed.

Dynamic Systems

For a system with n generalized co-ordinates and m kinematic constraints, the degrees of freedom (DOF) can be calculated by DOF $= n - m$. In this scenario, if the external forces acting on the system are provided along with sufficient number of proper initial conditions, we can perform a dynamic analysis to determine the response of the system.

To perform a dynamic analysis, the equations of motion of the constrained systems of rigid bodies must be derived and solved. Depending on the method of formulation, the governing equations can take the shape of a set of ordinary differential equations (ODEs) or differential algebraic equations (DAEs).

Example of a dynamic problem (as a set of DAEs)

A freely suspended double pendulum can be considered as an example of a dynamic analysis. The system can be modelled using either absolute coordinates or joint coordinates $q = \{\alpha, \beta\}$, as shown in figure 1.4.

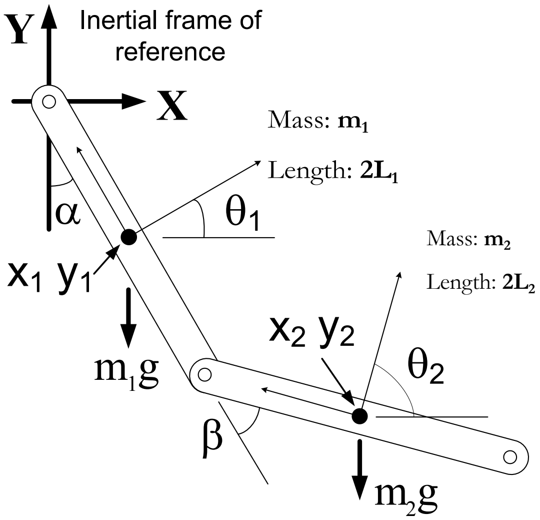


Figure 1.4: Double pendulum with absolute and joint coordinates

If absolute coordinates are chosen for the formulation, the system is described by six generalized coordinates. Since there are two revolute joints in the system, four kinematic constraints are obtained and from that we can calculate the degrees of freedom for this system as DOF $= 6 - 4 = 2$.

Compatibility of the dimensions of the matrices is an important cross check for the formulation process. For this system we expect the mass matrix and the Jacobian to be 6 × 6 and 4 × 6, respectively. The constraint vector can be expressed as shown in equation (1.9).

$$\mathbf{\Phi} = \begin{bmatrix} x_1 - L_1 \sin\theta_1 \\ y_1 + L_1 \cos\theta_1 \\ x_1 - x_2 + L_1 \sin\theta_1 + L_2 \sin\theta_2 \\ y_1 - y_2 - L_1 \cos\theta_1 - L_2 \cos\theta_2 \end{bmatrix} = 0 \tag{1.9}$$

The dynamic equations can be generated using Lagrange multipliers. A general form as shown in equation (1.10).

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{q}} + \mathbf{\Phi}_{\mathbf{q}}^{\mathbf{T}}\lambda =& \mathbf{Q}^{\mathbf{A}} \\ \mathbf{\Phi} = \mathbf{0} \end{aligned} \tag{1.10}$$

M: represents the mass matrix of the system

q: The vector of generalized coordinates, $[x_1, y_1, \theta_1, x_2, y_2, \theta_2]^T$

Q$^{\mathrm{A}}$**:** is the vector of generalized forces

λ**:** is the vector of Lagrange multipliers that enforces the constraints.

The equation below illustrates the governing dynamic equations for the double pendulum shown in figure 1.4 using absolute coordinate formulation.

$$\begin{bmatrix} m_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & m_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & m_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & J_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & J_1 \end{bmatrix} \begin{bmatrix} \ddot{x}_2 \\ \ddot{y}_2 \\ \ddot{x}_1 \\ \ddot{y}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_1 \end{bmatrix} + \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix} = \begin{bmatrix} 0 \\ -m_2 g \\ 0 \\ -m_1 g \\ 0 \\ 0 \end{bmatrix} \tag{1.11}$$

In equation (1.11), $J_i = m_i {L_i}^2/3$ and the associated Jacobian matrix is given by

$$\mathbf{\Phi}_{\mathbf{q}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & -L_1\cos\theta_1 \\ 0 & 0 & 0 & 1 & 0 & -L_1\sin\theta_1 \\ 1 & 0 & -1 & 0 & -L_2\cos\theta_2 & -L_1\cos\theta_1 \\ 0 & 1 & 0 & -1 & -L_2\sin\theta_2 & -L_1\sin\theta_1 \end{bmatrix} \tag{1.12}$$

Example of a dynamic problem (as a set of ODEs)

If joint coordinates are chosen for the derivation of equations, the governing equations take the form of a set of ODEs. In this system it is possible to model the double pendulum in terms of the two angles α and β. This results in two coupled differential equations as shown in (1.13).

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{Bmatrix} \ddot{\alpha} \\ \ddot{\beta} \end{Bmatrix} = [\mathbf{Q}]^{2\times 1} \tag{1.13}$$

where

$$\begin{aligned} M_{11} &= \frac{4}{3}m_1{L_1}^2 + \frac{4}{3}m_2{L_2}^2 + 4m_2L_1L_2\cos\beta + 4m_2{L_1}^2 \\ M_{12} &= M_{21} = \frac{2}{3}m_2L_2\left(2L_2 + 3L_1\cos\beta\right) \\ M_{22} &= \frac{4}{3}m_2{L_2}^2 \end{aligned} \tag{1.14}$$

And

$$\mathbf{Q} = \begin{bmatrix} \begin{pmatrix} -m_2gL_2\cos\alpha\sin\beta - m_2gL_2\sin\alpha\cos\beta \\ +4m_2L_1L_2\dot{\theta}_1\dot{\theta}_2\sin\beta + 2m_2L_1L_2\dot{\beta}^2\sin\beta \\ -m_1g\ L_1\sin\alpha - 2m_2g\ L_1\sin\alpha \end{pmatrix} \\ -m_2L_2\left(g\ \cos\alpha\sin\beta + g\ \sin\alpha\cos\beta + 2L_1(\dot{\alpha})^2\sin\beta\right) \end{bmatrix} \tag{1.15}$$

It can be clearly seen that .these equations contain various model parameters. For this system of double pendulum, the model parameters are the masses of the links m_1 and m_2, the lengths L_1 and L_2 and the acceleration due to gravity g. It is clear that the response of the system is dependent on the values of these parameters. This is precisely where sensitivity analysis comes into the picture. Using sensitivity analysis it is possible to understand the causal connection between the perturbation of the model parameter values and the response of the system.

1.1.3 Sensitivity analysis of dynamic systems

For multibody lumped parameter systems, the most general form of the governing equation is a set of DAEs as shown in equation (1.10). To perform sensitivity analysis on a system, a measure for the relevant performance or an objective function must be defined. For multibody systems governed by equation (1.10), a general objective function can be formulated as shown in equation (1.16).

$$\Psi = G\left(\left.(\mathbf{q}, \dot{\mathbf{q}})\right|_{t=T}, \mathbf{p}, T\right) + \int_{t_0}^{T} F\left(\mathbf{q}, \dot{\mathbf{q}}, \lambda, \mathbf{p}, t\right)\,dt \tag{1.16}$$

In the above equation, G and F represent arbitrary functions with sufficient smoothness, $\mathbf{p}$ represents the set of parameters, and $\mathbf{q}, \dot{\mathbf{q}}$ and λ are the generalized displacements, velocities and the Lagrange multipliers, respectively. The degree of smoothness required is dependent on the type of analysis being carried out. For first order sensitivity analysis the functions G and F are required to be C^1 continuous.

The term G is a function of the final time T, the model parameters $\mathbf{p}$ and the values of $\mathbf{q}$

and $\dot{\mathbf{q}}$ evaluated at $t = T$. The term F is an objective function in an integrated form where the integration is performed over a time span with the final time $T = T(\mathbf{p})$ being a function of the model parameters.

The sensitivity of the objective function Ψ can be obtained by differentiating the expression with respect to $\mathbf{p}$. Using the chain rule of differentiation on equation (1.16) we obtain an expression for the sensitivity S. For convenient illustration, we use subscripts to denote partial differentiation operation.

$$f_b = \frac{\partial f}{\partial b} \tag{1.17}$$

$$\begin{aligned} \Psi_p \;\; = \;\; & G_{\mathbf{p}} + G_T T_{\mathbf{p}} + G_{\mathbf{q}} \left(\mathbf{q}_{\mathbf{p}}|_{t=T} + \left(\dot{\mathbf{q}}|_{t=T} \right) T_{\mathbf{p}} \right) + G_{\dot{\mathbf{q}}} \left(\dot{\mathbf{q}}_{\mathbf{p}}|_{t=T} + \left(\ddot{\mathbf{q}}|_{t=T} \right) T_{\mathbf{p}} \right) \\ & + \int\limits_{t_0}^{T} \left(F_{\mathbf{q}} \mathbf{q}_{\mathbf{p}} + F_{\dot{\mathbf{q}}} \dot{\mathbf{q}}_{\mathbf{p}} + F_{\lambda} \lambda_{\mathbf{p}} + F_{\mathbf{p}} \right) dt + \left(F|_{t=T} \right) T_{\mathbf{p}} \end{aligned} \tag{1.18}$$

Thus, to evaluate $\Psi_{\mathbf{p}}$, one needs to evaluate $T_{\mathbf{p}}, \mathbf{q}_{\mathbf{p}}, \dot{\mathbf{q}}_{\mathbf{p}}$ and $\lambda_{\mathbf{p}}$, i.e., the derivatives of $T, \mathbf{q}, \dot{\mathbf{q}}$ and λ with respect to the parameters, respectively.

There are different numerical, analytical and hybrid methods that can evaluate these derivatives. The divided difference method, automatic differentiation techniques, direct differentiation and adjoint variable method have been used by many researchers to perform sensitivity analysis. Each of these methods has its own sets of advantages and disadvantages. For proper application of sensitivity analysis, it is necessary to address the issues encountered by these methods to improve their suitability in different engineering applications. In the subsequent chapters, details of some of the existing methods are discussed with examples and simulation results from kinematic and dynamic problems.

1.2 Challenges and motivation

Many researchers who work in this area often face questions from individuals, not acquainted with the challenges of sensitivity analysis, regarding the justification of research efforts towards better methods of sensitivity analysis, which after all is just a process that evaluates derivatives of certain quantities. To address this query, it is important to emphasize the ever-increasing complexity of the situation. With the rapid increase in the size and complexity of models being analyzed, the process of providing efficient and accurate sensitivity information suitable for diverse engineering applications is never "*just*" a differentiation. To illustrate further on the topic, it is necessary to compare the advantages and disadvantages of the various methods currently in use for sensitivity analysis.

The finite difference method is notorious for its numerical instability and susceptibility to parameter perturbation. Also, for a system that is described by n differential equations, m algebraic constraints and has p parameters, the finite difference method requires the solution of $3 \times p(n+m)$ equations. Because of these issues, the finite difference method is not an option for efficient and accurate sensitivity analysis.

In direct differentiation, the number of equations to be solved is $(p+1)(n+m)$. Although, it is far better than finite difference formulation in terms of numerical stability and accuracy, the conventional direct differentiation based formulation becomes intractable for systems with a large number of parameters, especially if the size of the system is also large. Special direct differentiation techniques like the Recursive Formulation claim to be better in performance, but recursive formulations are most effective for systems with long chain-like structures and under normal circumstances don't offer sizable advantages.

The total number of equations solved in the adjoint variable method is $(n+m+p)$. But the very nature of the adjoint variable method requires the storage of every bit of information of the forward dynamic problem to evaluate the sensitivity information. This makes it susceptible to memory and interpolation problems. The implementation of the adjoint variable method is complicated, and requires different formulations for different problem structures. For large systems with nonlinear constitutive equations, the adjoint system of equation becomes complicated to solve and as a result, the adjoint variable method becomes inefficient.

In light of the above discussion, it is clear that none of the methods, currently being used to perform sensitivity analysis for various applications, can deliver the performance that is expected. In this thesis an attempt is made to fill this void by combining direct differentiation with graph-theoretic modelling methods. It is expected to result in a method that would have the simplicity, stability and accuracy of direct differentiation method and clarity, flexibility and efficiency of the graph-theoretic approach.

Further details of the methods used for sensitivity analysis along with a review of available literature and research results are presented in the next chapter.

1.3 Applications

As explained in a previous section, sensitivity analysis is routinely used in many engineering applications. Generally, the entire process can be divided into three main stages. The first stage is the generation of the underlying model of the system. Better and efficient models lead to more efficient post-processing. For multibody systems, commercial softwares like MapleSim are available, that uses graph-theoretic formulations for efficient generation of simplified governing equations.

The second stage is where the sensitivity analysis is formulated and the equations governing the individual sensitivities are generated. In practical applications, this stage is often intertwined with the third stage where the equations are numerically solved to produce the desired results.

The proposed method of graph-theoretic sensitivity analysis is an attempt to extend the use of graph-theoretic formulation from the first stage to the second stage of this procedure. The expectation is to achieve the capability of controlling the size and the structure of the symbolic expressions of the sensitivity equations using graph-theoretic processing.

The proposed method is implemented using `Maple` and `MapleSim`. Although this method can be used for multi-domain systems, the current implementation in `MapleSim` is restricted to multibody systems only. This is because in `MapleSim`, only the multibody portion is derived using the graph-theoretic approach. At its present form, the `MapleSim` implementation allows an user to model a multibody system using `MapleSim`'s component library and then use the attached worksheet to generate the desired sensitivity equations. The worksheet also offers flexibility in terms of selection of different modelling features, which results in greater control over the structure and size of the resulting set of sensitivity equations.

1.4 Thesis organization

This section contains a brief description of the contents of this thesis. In Chapter 2, descriptions of available papers and research results are presented. The first half of Chapter 2 is focused on graph-theoretic modelling methods and contains an overview of the process of equation generation for three-dimensional multibody systems. The second half contains information about sensitivity analysis. Different methods of sensitivity analysis are presented along with descriptions of the associated advantages and disadvantages.

In chapter 3, a graph-theoretic approach is proposed for the sensitivity analysis of scalar systems. This chapter starts with a theoretical background on the subject and a discussion about the derivation of sensitivity equations using direct differentiation. Then the graph-theoretic method is used to generate the same equations to demonstrate the validity of the process. The new approach is first explained in theory and then in practice,using examples of a Ni-MH battery and a hydrodynamic torque converter model. The chapter also contains some examples about possible applications of sensitivity analysis.

In chapter 4, graph-theoretic sensitivity analysis is applied to three-dimensional multibody systems. Using an example problem, the generation of sensitivity equations is explained and demonstrated.

In chapter 5, the the graph-theoretic sensitivity analysis algorithm is applied to a few multibody systems to demonstrate the applicability of this method and to evaluate its performance for comparison. Also, to showcase its applicability toward multi-domain systems, an example of an electric motor driven multibody mechanism is also included in this chapter.

A summary of research contributions and a list of resulting publications are provided in chapter 6. It also contains recommendations for future work on the subject of efficient sensitivity analysis in general and graph-theoretic sensitivity analysis in particular.

Chapter 2

Literature review

This chapter presents available literature and existing research results, that are relevant to the current topic. The first part of this chapter is used to cover graph-theoretic modelling methods and the second part is used to discuss sensitivity analysis. In each section, different formulations are discussed along with their applicability and disadvantages.

2.1 Graph-theoretic modelling

Graph theory was first used to study interconnections between entities by the famous Swiss mathematician Leonhard Euler [1] in the year 1736. Over the centuries, it was developed by various mathematicians as a tool to study system topologies.

In the twentieth century however, engineers recognized its potential as a powerful tool for system modelling. Koenig et al. [2] , Trent [3], Seshu and Reed [4] and many others have extended graph theory to model physical systems by combining linear graphs with the set of constitutive equations representing the physical behaviour of system components. This particular method of system modelling or "*Graph-theoretic Modelling*" is known for its unified approach toward modelling multi-domain systems. In light of the definitions presented in the previous chapter, this section presents an overview of the mathematics behind graph-theoretic modelling and a brief discussion of the available literature on the subject.

As a compact mathematical representation of the connectivity of the linear graph representing a system, an incidence matrix **I** can be constructed from the graph itself. Each of the columns in this matrix corresponds to a particular edge of the graph, and the rows correspond to the nodes.The matrix is generated column by column, by putting +1 at the node at which the edge originates, −1 at the node at which the edge ends, and 0 at the nodes not connected by the edge.

The reduced incidence matrix **A** can be derived from the incidence matrix **I** by deleting the row corresponding to the node which is selected as a datum. The rows of this matrix **A** are linearly independent, which is essential for the formulation.

Performing Gauss-Jordan elimination on the reduced incidence matrix the fundamental cutset matrix $\mathbf{A_f}$ is obtained which has the following form.

$$\mathbf{A_f} = [[1][\mathbf{A_u}]] \tag{2.1}$$

The first sub-matrix of $\mathbf{A_f}$ is a $w \times w$ identity matrix. The sub-matrix $\mathbf{A_u}$ is a $w \times u$ matrix also known as the chord transformation matrix.

The generation of the topological equations are governed by the vertex and circuit postulates as presented below.

Vertex postulate: The sum of through variables at any node of a linear graph must equal zero when due account is taken of the orientation of edges incident upon that node, McPhee [5], Savage [6].

This postulate summarizes the balance of the through variables at every node of the complete graph G. The f-cutset equations can be obtained by considering the balance of the through variables in the sub-graphs of G, left out by the fundamental cutsets. Mathematically the fundamental cutset equations can be expressed as

$$\mathbf{A_f}\boldsymbol{\tau} = 0 \tag{2.2}$$

where $\mathbf{A_f}$ is the fundamental cutset matrix and $\boldsymbol{\tau}$ is the vector of through variables of the graph, arranged in the proper order.

Circuit postulate: The sum of across variables around any circuit of a graph must equal zero when due account is taken of the direction of edges in the circuit, McPhee [5], Savage [6].

By applying this postulate to the fundamental circuits it is possible to obtain u independent equations, which are known as the f-circuit equations. Mathematically, they can be expressed as

$$\mathbf{B_f}\boldsymbol{\alpha} = 0 \tag{2.3}$$

In equation (2.3), $\boldsymbol{\alpha}$ is the column vector of across variables of the graph arranged in proper order. The matrix $\mathbf{B_f}$ is known as the fundamental circuit matrix and it has a structure as shown in equation.

$$\mathbf{B_f} = [[\mathbf{B_w}][1]] \tag{2.4}$$

The second sub-matrix of the fundamental cuset matrix $\mathbf{B_f}$ is a $u \times u$ identity matrix. The first sub-matrix $\mathbf{B_w}$ is a $u \times w$ matrix known as the branch transformation matrix. It can be evaluated using the relationship shown in equation (2.5).

$$\mathbf{B_w} = -\mathbf{A_u}^{\mathrm{T}} \tag{2.5}$$

By combining the constitutive equations with the topological equations shown in equation (2.3) and (2.2) a complete set of governing equations can be generated.

The nature of the constitutive equations and the specific process of equation generation depend on the nature of the system. For hydraulic and electrical systems the state variables are scalar in nature and as a result the through and across variables also become scalars which considerably simplifies the process of equation generation. For this reason, sensitivity analysis of scalar systems are presented in a separate chapter of this thesis.

For three-dimensional multibody systems, the process of graph-theoretic modelling is a little more involved. For these systems, not only the through and across variables become vectors, also it becomes necessary to include two distinct domains, i.e., rotational and translational, in the modelling process. It is therefore worthwhile to present an outline of graph-theoretic formulation of multibody systems.

2.1.1 Graph-theoretic modelling of multibody systems

McPhee [7] has provided a fairly comprehensive summary of efforts from various researchers on multi-body systems modelling using graph-theoretic methods. Multi-body systems can be modelled using various coordinate schemes. Absolute coordinates are one of the simplest of choices. However, the capabilities and the efficiency of the graph-theoretic method are not demonstrated by the absolute coordinate formulation.

The real benefits of GTM become apparent when other coordinates are used. For example in a joint coordinate formulation, it is very important to keep track of the topology of the system. This is exactly where graph-theoretic models provide very convenient descriptions. Since absolute coordinate formulations are often inefficient because of the large number of equations they generate, graph-theoretic methods are in fact very important tools for modelling large systems.

One of the first applications of linear graphs to multi-body systems was done by Sheth and Uicker [8]. They derived supplementary loop closure equations for a multi-body system consisting of closed kinematic loops from a linear graph where the nodes represented the links and the edges represented the joints. As an extension, Wittenburg [9] used linear graphs to represent systems of rigid bodies with both closed and open kinematic loops. His representation generalized the edges to include springs, dampers and other complex connections with six degrees of freedom. In his approach graph theory was used to generate the kinematic loop closure equations. Hwang and Haug [10] have implemented a recursive linear-order formulation based on linear graph representation of systems of rigid bodies. Lai et al. [11] has used “extended graphs”, where nodes were used to represent reference frames. This particular representation is very similar to that used in the current work.

All of these efforts mentioned so far are examples of ad hoc applications of graph theory to dynamic systems modelling. One of the first applications of formal graph-theoretic methods to multi-body dynamics was done by Andrews and Kesavan [12] in their “Vector Network Graphs”.

In the current work, to model multi-body systems linear graphs are used, where the nodes represent body-fixed reference frames. The edges represent all the physical entities of the system (e.g., masses, forces, joints etc). This is entirely based on the work presented by McPhee [5].

The general motion of any rigid body can be represented by the combination of purely translational and purely rotational motions. When rigid bodies are connected through joints to form a multi-body system, the translational and rotational motions of one constituting body get intertwined with the translational and rotational motions of the other bodies. Therefore to represent the motion of a multi-body system properly, it is necessary to consider two sets of through and across variables associated with the edges of the graph representing the system. For instance, an edge which represents the centre of mass of a rigid body would have both torque and force as its through variables and both translational displacement/velocity/acceleration and rotational orientation/velocity/acceleration as its across variables. Consequently two sets of cutset and circuit equations are generated from the linear graph, i.e., one each for the translational and the rotational aspect of the motion. Since the through and across variables for a graph representing a multi-body system are represented by vectors, these cutset and circuit equations appear as vector equations.

The terminal equations for multi-body systems capture the physical characteristics of the component in question. Since there are two sets of through and across variables, every component is associated with two terminal equations. As an example, for an ideal revolute joint the equations are

$$\bar{r} = 0 \quad \text{and} \quad \bar{T}.\hat{u} = 0 \tag{2.6}$$

In the above equations $\bar{r}$ and $\bar{T}$ are the *translational across quantity* and the *rotational through quantity* associated with the edge representing the joint respectively and $\hat{u}$ is the unit vector along the revolute axis. More general discussions about terminal equations are presented by Li and Andrews [13] and Andrews et al. [14]. Using two sets of cutset, circuit and terminal equations it is possible to solve for the quantities describing the motion of the system. However, if a branch-chord formulation is used, then all secondary variables (i.e., through variables from the tree and across variables from the cotree) can be eliminated from the system equations. The resulting equations are expressed in terms of the across variables associated with the branches, and through variables associated with the chords. Collectively these quantities are known as the primary variables. Thus by using a branch-chord formulation, a decrease in the number of equations is possible.

The following steps are performed to generate the governing equations for a multibody system.

- Projection of cutset equations on the motion space
- Projection of circuit equations on the reaction space
- Elimination of secondary variables (in the specified order)
  1. Rigid arm forces
  2. Cotree translational across variables
  3. Cotree rotational across variables
- Assembly of motion equations

One great advantage of graph-theoretic formulation is that it offers the flexibility to select the set of generalized coordinates that best describe the system. By selecting different trees for the translational and rotational aspect of the motion, it is possible to ensure generation of governing equations in term of any reasonable set of generalized coordinates.

This method can be used to generate an optimized set of governing equations if the branches for the T-tree and the R-tree are selected in a manner such that the set of primary variables of the model has minimum number of unknown quantities. McPhee [5] and Leger [15] has presented a smart scheme for tree selection that ensures optimized coordinate formulation. Throughout this work, translational trees (T-trees) and rotational trees (R-trees) from a single graph are used, in order to generate the governing equations in terms of the generalized coordinates.

An example of a simple pendulum is provided here to illustrate the procedure of the generation of governing equations for multibody systems.

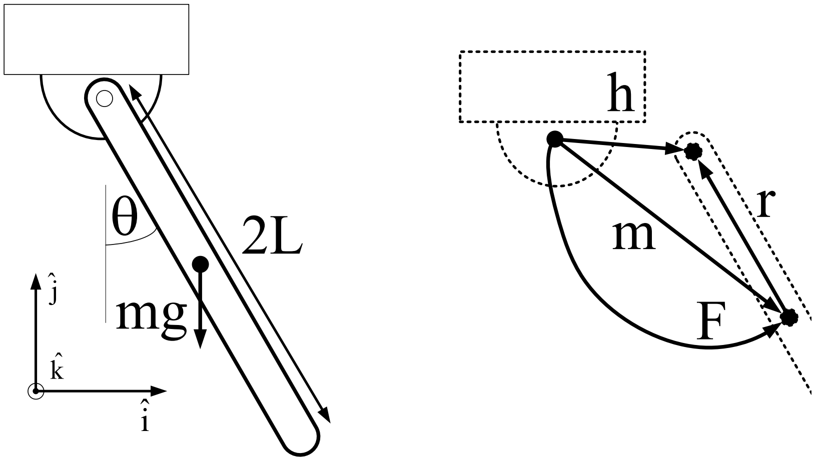


Figure 2.1: A simple pendulum and the corresponding linear graph

Figure 2.1 shows a simple pendulum of length $2L$ hinged about a point. It also shows the corresponding linear graph of the system. The nodes of the graph represents body fixed frames of references. For example, the node at the centre of the rod represents a frame of reference fixed to the rod at its centre of mass.

On the other hand, the edges of the linear graph represent sets of related measurements. The relationships between these measurements determine the distinct physical components the edges correspond to. The figures shows the labels of these edges in accordance with their functional identities. The edge h is the revolute joint, r is the rigid arm vector attached to the rod, m represents the mass and rotational inertia of the body and the edge F represents the force of gravity acting on the pendulum.

With these four edges, the first step is to identify the through and across variables. Since all of these edges have both translational and rotational significance, they are associated with through and across variables from both translational and rotational domains.

Translational through variables: $\bar{F}_m, \bar{F}_r, \bar{F}_h, \bar{F}_F$

Rotational through variables: $\bar{T}_m, \bar{T}_r, \bar{T}_h, \bar{T}_F$

Translational across variables: $\bar{r}_m, \bar{r}_r, \bar{r}_h, \bar{r}_F$

Rotational across variables: $\bar{\phi}_m, \bar{\phi}_r, \bar{\phi}_h, \bar{\phi}_F$

Step 1: Selection of a tree

The edges h, r for the T-tree and the edges r, m for the R-tree are chosen as tree, ensuring the generation of equations in terms of the coordinate $\mathbf{q} = \{\theta\}$.

Step 2: Projection of cutset equations onto the motion space

From figure 2.1 it can be seen that the motion space of the revolute joint is the direction given by the unit vector $\hat{k}$ and the reaction space is given by the unit vectors $\hat{\imath}$ and $\hat{\jmath}$. Since the cutset equations from the T-tree are in terms of quantities which are always orthogonal to the direction vector , the projected cutset equations from the T-tree reduce to zero. From the R-tree however, a single projected cutset equation is obtained.

$$\left(\bar{T}_m - \bar{T}_h = 0\right) \quad . \quad \hat{k} \tag{2.7}$$

Step 3: Projection of circuit equations onto the reaction space does not yield any equations, as expected for the 1-DOF system modelled by a single independent coordinate.

Step 4: Elimination of secondary variables

Substituting terminal equations into equation (2.7)

$$\left(-I_{cm}\ddot{\bar{\phi}}_m - \left(\bar{r}_r \times \bar{F}_r\right) = 0\right) \ . \ \hat{k} \tag{2.8}$$

For the rigid arm force

$$\bar{F}_r = \bar{F}_m + \bar{F}_F = -m\ddot{\bar{r}}_m - mg\,\hat{\jmath} \tag{2.9}$$

For the cotree across variable

$$\ddot{\bar{r}}_m = -\ddot{\bar{r}}_r - \ddot{\bar{r}}_h = -\ddot{\bar{\phi}}_m \times \bar{r}_r + \left(\left|\dot{\bar{\phi}}_m\right|\right)^2 \bar{r}_r \tag{2.10}$$

Definition of $\bar{r}_r$ and $\dot{\bar{\phi}}_m$

$$\bar{r}_r = -\left(L\sin\theta\right)\hat{\imath} + \left(L\cos\theta\right)\hat{\jmath} \quad , \quad \dot{\bar{\phi}}_m = \dot{\theta}\,\hat{k} \tag{2.11}$$

Step 5 : Assembly of governing equations

By substituting equations (2.9),(2.10), and (2.11) into equation (2.8) the following differential equation in terms of θ is obtained.

$$\left(I_{cm} + mL^2\right)\ \ddot{\theta} = -mgL\ \ \sin\theta \tag{2.12}$$

A similar process of equation generation is used throughout the course of this research.

2.1.2 Applications of graph-theoretic modelling

Graph-theoretic modelling methods have been used extensively in many diverse applications. Because of its generalized approach, graph-theoretic modelling methods can be readily extended from modelling complex mechanisms to multi-domain systems where different forms of energy interact with each other.

Schmitke et al. [16] have used graph-theoretical methods to create efficient models to study vehicle dynamics. Similar efforts have been used to model tire soil interactions.

Sass et al. [17] has provided a comparative study of different methods of modelling electro-mechanical multi-body systems. Scherrer and McPhee [18] have presented a unified method for modelling electro-mechanical systems with multi-body mechanical systems and analog electrical components. An extension of this paper was presented by McPhee et al. [19] to include flexible multi-body systems and the concept of assembly of subsystems. Linear graph models have also been used to model electro-chemical reactions and to simulate the operation of Ni-MH batteries, Dao and McPhee [20]. The main issue behind these multi-domain models is to model transducer components that have terminal equations in both domains. Moving plate capacitors, moving coil inductors and electrical prime movers are all examples of such components.

Using similar principles, graph-theoretic methods have also been used to model hydrodynamic machines. Banerjee and McPhee [21] have presented a graph-theoretic model of an automotive torque converter.

2.1.3 Software Implementation

The automated generation of symbolic governing equations is a desirable feature for efficient sensitivity analysis of dynamic systems. During the design phase of development, system models undergo numerous alterations. An automated generation of governing equations can greatly improve the effectiveness of this process. In this section, brief overviews of some of the existing software packages that are used for modelling multibody and/or multi-domain systems will be presented.

Maple: Maple is a general-purpose computer algebra system developed and marketed by Waterloo Maple Inc. It incorporates a dynamically typed programming language which resembles Pascal. There are provisions of interfacing with C, Fortran, Java and Matlab. The heart of Maple is a kernel written in **C**, which provides the Maple language. Most mathematical functionalities are provided by libraries. The user interface is written in Java.

The main aspect of Maple is the ability to manipulate symbolic equations and expressions. It contains a very large library of symbolic operations. Simplification and modification of symbolic expression are routinely done by Maple.

Apart from the symbolic capabilities, Maple can also be used for numerical simulations. Advanced numerical routines allow users to solve complicated large systems of ODEs and DAEs using a variety of different algorithms.

Because of its excellent symbolic and numeric capabilities, Maple can be used to simulate multibody systems effectively. Mathematically it becomes the process of solving a set of ODEs or DAEs depending on the nature of the system. Also it provides the platform for other modelling softwares.

DynaFlexPro: DynaFlexPro was developed by McPhee and Schmitke at the University of Waterloo, Canada. It is capable of generating symbolic governing equations for multi-domain engineering systems.

Their formulation is based on linear graph theory and the principle of orthogonality [22]. The input to the program is a topological description of the system using predefined blocks. DynaFlexPro is implemented using Maple. It extensively uses Maple's symbolic processing libraries to generate and simplify symbolic equations. It can also be used to generate very efficient simulation codes for different languages. The "CodeGeneration" package can produce optimized simulation codes for Maple, C, Matlab and Fortran.

MapleSim: MapleSim is a multi-domain modelling and simulation tool developed by Maplesoft Inc. It is capable of simulating electrical, electronic, mechanical, hydraulic and magnetic systems. The input to the software is the description of the system using the components from a central library that can be drag-dropped on to a worksheet.

MapleSim is built on Maple, which enables it to perform very effective symbolic simplifications of the generated equations. Also, using Maple's DAE and ODE solvers, it can simulate and present the output of the models in an interactive three-dimensional environment. It can also perform post-processing on the generated data using predefined templates.

The multibody package of MapleSim is based on the DynaFlexPro engine and uses a linear graph based formulation. Other components (e.g. electrical, magnetic, thermal, hydraulic etc) are based on Modelica codes.

Apart from the built-in library of standard components, MapleSim allows users to create custom components for user specific implementation. These custom components are based on the Maple language and can be readily included in models created using MapleSim's standard components. It is also possible to create custom components based on Modelica codes.

The proposed method of sensitivity analysis is implemented using MapleSim. Since Maplesim uses graph-theoretic method to model multibody systems only, the implementation is specifically applicable for multibody systems. However, theoretically it can be applied to any multi-domain system represented by a linear graph.

2.2 Sensitivity analysis of dynamic systems

Sensitivity analysis can be classified into two basic categories. Based on the domain in which their results stay valid, sensitivity analysis can be described as either a global or a local study.

The term "global sensitivity analysis" was introduced by noted econometrician Edward Leamer [23]. In a successful global sensitivity analysis, the conclusions remain valid for the entire range of values of the parameters [24]. Mathematically, global sensitivity analysis can be performed by variance based methods like high dimensional model representation [25] or sampling based methods like the elementary effect method and Monte Carlo filtering.

However, for practical multibody systems, these methods can be extremely inefficient. Due to the requirement of large number of simulations, sampling and statistical sensitivity analysis is often not suitable and much better performance can be obtained by employing what is known as local sensitivity analysis.

On the other hand, local sensitivity analysis gives results that are valid only in the neighborhood of the point at which they are evaluated. Since multibody systems are non-linear in nature, local sensitivity information are generally not valid for the entire range of possible parameter values. As a result, conclusions based on local sensitivity analysis need a specification of operating point to become meaningful.

Direct differentiation, the adjoint variable method, automatic differentiation and other analytical formulations can be used to evaluate local sensitivity information. The fact that these methods yield locally valid information makes these approaches suitable for optimization algorithms and have been used in a wide range of engineering applications.

Haug and Serban [26] have stated

> Dynamic design sensitivity analysis of multi-body systems represents the link between optimization tools and simulation tools.

2.2.1 Finite difference formulation

One of the easiest methods of local sensitivity analysis for multibody systems is to use a finite difference scheme to evaluate the gradient information of the objective function with respect to the parameters. This method is probably the simplest to implement and can easily be extended to evaluate derivatives of higher orders. This method can be explained by a simple Taylor series expansion.

For a system governed by a differential equation $f(q, \dot{q}, b, t) = 0$ with state variable q and model parameter b, a perturbation in the value of the parameter also affects the variation of q with time. Since q is a function of both b and t, its perturbed value can be written as $q(b + \delta b, t)$. Using Taylor expansion it can be written

$$q\left(b+\delta b, t\right) = q\left(b, t\right) + \frac{\partial q}{\partial b} \times \delta b s \tag{2.13}$$

Rearranging terms

$$\frac{\partial\, q}{\partial\, b} = \frac{q\left(b+\delta b, t\right) - q\left(b, t\right)}{\delta b} \tag{2.14}$$

This demonstrates the use of forward difference scheme to determine the first order sensitivity of the state variable q with respect to the model parameter b. Other difference schemes can also be used for this purpose. Also similar approaches can be applied to evaluate higher order sensitivities.

However it is often not worthwhile for practical implementation because of its dependence on the perturbation size. Furthermore, researchers [27, 28, 29] have shown that extra computational cost is necessary to determine optimal perturbations for the parameters for different scenarios.

2.2.2 Direct differentiation

The objective of local sensitivity analysis for a multibody system is to evaluate the derivative of the state variables representing the system. For models where symbolic equations are available, this can be accomplished by a method known as the direct differentiation.

To perform sensitivity analysis on these models, one approach is to find a stepping stone between the system equations governing the state variables and the sensitivity equations governing the sensitivities of the state variables. The simplest method to derive the set of sensitivity equations is to use the method of direct differentiation [26, 30, 31] and differentiate the governing equations symbolically.

In direct differentiation, a set of auxiliary equations known as the sensitivity equations are generated from the original system equations by differentiating them with respect to the model parameters. By solving these sensitivity equations, the corresponding sensitivity information is obtained as functions of time. To illustrate this method, we consider a system governed by a set of nonlinear equations as shown in equation (2.15), where $\mathbf{q}$ is a vector of state variables, $\mathbf{p}$ is the vector of model parameters and $\mathbf{f}$ is a vector of sufficiently smooth functions (C^1 continuous for 1^{st} order sensitivity analysis).

$$\mathbf{f}\left(\mathbf{q}, \mathbf{p}, t\right) = 0 \tag{2.15}$$

We also introduce a set of objective functions as given in equation (2.16).

$$\mathbf{g}\left(\mathbf{q}, \mathbf{p}, t\right) = 0 \tag{2.16}$$

In equations (2.15) and (2.16), the number of state variables and the number of model parameters are n_1 and n_2 respectively i.e., $\mathbf{q} \in \mathrm{R}^{n_1}$ and $\mathbf{p} \in \mathrm{R}^{n_2}$. The vectors $\mathbf{f}$ and $\mathbf{g}$ have n_1 and k functions respectively, $\mathbf{f} : \mathrm{R}^{n_2+n_1} \rightarrow \mathrm{R}^{n_1}$ and $\mathbf{g} : \mathrm{R}^{n_2+n_1} \rightarrow \mathrm{R}^{k}$.

The objective of sensitivity analysis is the evaluation of the derivative of $\mathbf{g}$ with respect to the model parameters $\mathbf{p}$. Using matrix notations, the expression for this derivative is shown in equation (2.17).

$$\mathbf{S} = \frac{d\mathbf{g}}{d\mathbf{p}} = \mathbf{g_q}\mathbf{q_p} + \mathbf{g_p} \tag{2.17}$$

The terms $\mathbf{g_q}$ and $\mathbf{g_p}$ in equation (2.17) can be derived from the structure of the vector $\mathbf{g}$. However, the $n_1 \times n_2$ matrix $\mathbf{q_p}$ is unknown and can only be evaluated from the original model. To calculate the matrix $\mathbf{q_p}$ we differentiate equation (2.15) with respect to the model parameter vector $\mathbf{p}$ and obtain the expression shown in equation (2.18).

$$\mathbf{f_q q_p} + \mathbf{f_p} = 0 \tag{2.18}$$

Equation (2.18) is matrix of $n_1 \times n_2$ equations that can be solved to evaluate the $n_1 \times n_2$ elements of the matrix $\mathbf{q_p}$. The matrix $\mathbf{q_p}$ can then be substituted in equation (2.17) to evaluate the required sensitivity information.

Direct differentiation is portable, easy to implement, stable, and produces results that are numerically exact. This method has been used to perform sensitivity analysis for systems governed by kinematic equations [32], ordinary differential equations [6, 33, 34] and also differential-algebraic equations [35]. Serban and Freeman [35] have performed simultaneous sensitivity analysis with respect to multiple parameters. It is widely used to perform parameter identification [33] and design optimization [36].

The drawback of direct differentiation is also apparent from the presented example. To evaluate a sensitivity matrix $\mathbf{S}$ of $k \times n_2$ elements, direct differentiation requires the solution of $n_1 \times n_2$ equations, (2.18). Usually, for practical scenarios, the number k is much smaller than n_1 or n_2. This means, using direct differentiation method, one needs to solve a much larger set of equations to get a smaller set of required information.

The size and the complexity of equation (2.18) depends on the nature of the governing equation $\mathbf{f}$, the size of the vector of state variables $\mathbf{q}$ and the number of model parameters $\mathbf{p}$. For large systems, especially while dealing with a large number of parameters, this makes direct differentiation unsuitable for implementation. To address this issue, an alternate method using adjoint variables was developed.

2.2.3 Adjoint variable method

The process of finding the gradient of a function involving the state variables of a system with respect to a system parameter can be described as the process of optimizing an objective function, where the quantities are constrained by the set of governing system equations. Lagrange's method of optimization can be used to convert the constrained optimization problem into an unconstrained optimization problem using a set of *adjoint variables*. This is known as the adjoint variable method.

Mathematically, the adjoint variable method can be demonstrated by considering the system governed by equation (2.15). To evaluate the sensitivity of the objective function vector given in equation (2.17), the first step in the adjoint variable method is to introduce the adjoint variables by multiplying equation (2.18) with the transpose of a $n_1 \times k$ matrix $\boldsymbol{\lambda}$.

$$\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{f_q q_p} + \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{f_p} = 0 \tag{2.19}$$

The adjoint equations are formed as shown in equation (2.20). The terms $\mathbf{f_q}$ and $\mathbf{g_q}$ are derived from the vectors $\mathbf{f}$ and $\mathbf{g}$ using symbolic differentiation.

$$\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{f_q} = \mathbf{g_q} \tag{2.20}$$

By multiplying equation (2.20) with $\mathbf{q_p}$ we obtain

$$\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{f_q}\,\mathbf{q_p} = \mathbf{g}_{\,\mathbf{q}}\,\mathbf{q_p} \tag{2.21}$$

Combining equations (2.21) and (2.19) we obtain

$$\mathbf{g_q}\,\mathbf{q_p} = -\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{f_p} \tag{2.22}$$

The required sensitivity matrix can be then written as

$$\mathbf{S} = \frac{d\mathbf{g}}{d\mathbf{p}} = -\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{f_p} + \mathbf{g_p} \tag{2.23}$$

With the values of $\boldsymbol{\lambda}$ evaluated by solving equation (2.20), the sensitivity information $\mathbf{S}$ can be easily evaluated by substitution.

Using adjoint variable method, the $k \times n_2$ elements of $\mathbf{S}$ are evaluated through a set of $n_1 \times k$ adjoint variables. The number of adjoint variables required is independent of the number of model parameters. This makes this approach suitable for scenarios where a large number of model parameters are under study.

Physically, the adjoint variable method uses the Lagrange multipliers as a stepping stone to avoid the problems of direct differentiation. The adjoint variables force the governing equations to hold. Thus the goal switches from finding the derivatives of the governing equations to finding the adjoint variables which would make the system satisfy the governing equations.

The implementation of adjoint variable method is more complicated than that of direct differentiation. In case of dynamic systems, i.e., systems that are governed by ODEs or DAEs the generated adjoint systems become sets of ODEs and DAEs themselves.

Often, for practical applications, the objective functions are defined in an integrated form. For dynamic systems, this leads to a situation where the adjoint system becomes a terminal value problem, instead of an initial value problem.

Theoretically, this can be solved by simulating the governing equations in the forward time direction and using the simulation data to solve the adjoint system in the backward direction. This poses a challenge for the practical implementation of this method.

Efficient numerical integrators use adaptive step size selection method. If adaptive step size solvers are used, the time points where the actual quadrature is evaluated, never match up for the forward and backward simulations. To address this problem, interpolation polynomials associated with every time point of the forward problem need to be stored along with the response of the system.

From this discussion, the drawbacks of the adjoint variable method can be summarized. Its complex implementation and requirement of data storage makes it unsuitable for large systems, and for longer simulations. Also, for complicated objective functions, the adjoint variable method generates more complicated equations that need to be solved.

The adjoint variable method has been extensively used for optimal control and optimal design by Haug [37] and Bestle et al. [28, 38]. For control applications, the model size is usually small, which makes adjoint variable method appropriate for these applications.

Sandu et al. [39], Hindmarsh et al. [40, 41] and Petzold et al. [42, 43, 44] have worked extensively on this subject and have developed numerical packages to perform sensitivity analysis using the adjoint variable method on systems governed by different types of DAEs and ODEs.

Ding et al. [45] have extended this method to perform second-order sensitivity studies on DAEs. Serban [46] has presented a parallel computational model based on the adjoint variable method.

2.2.4 Recursive formulation

Recursive formulations were originally used to generate efficient sets of governing equations for multibody systems [10]. To avoid the complicated and resource-hungry implementation of the adjoint variable method, Anderson and other researchers [29, 47, 48, 49, 50] have extended the formulation to generate sensitivity equations using recursive methods.

However, the implementation of the recursive approach is very complicated and shows its benefits mainly for systems with open kinematic chains. It is also suggested that there is a critical number of bodies that must be present in a serial chain to make the recursive algorithm effective. For most multibody systems, especially those encountered in mechanisms and vehicular systems, this critical number is higher than the maximum number of bodies connected in series.

2.2.5 Automatic differentiation

To simulate models of physical systems, numerical methods are often used to calculate the values of the state variables that describe the configuration of the system as functions of time. For most practical systems, the numerical methods involve some sort of an integrator to solve the different types of differential equations encountered in the models.

Automatic differentiation is a numerical tool which, when applied to any numerical method, can calculate the derivatives of the output of the numerical method with respect to any relevant quantity.

It is based on the fact that any numerical process, however complicated, can be broken down into smaller elemental operations. Therefore, using the chain rule of differentiation and a table of standard differentiation results, it is possible to construct the derivative of the original numerical process.

In case of sensitivity analysis, when automatic differentiation is applied to the integrator that simulates the system, it calculates the values of the state variables and their derivatives at different points of time. This method of differentiation is completely different from the finite difference approach where the entire numerical process is performed repeatedly with perturbed values of the parameters.

Over the years, many numerical integration schemes have been developed to simulate systems governed by kinematic, differential [51] and differential-algebraic equations [52, 53]. Automatic differentiation has been successfully used to produce the state variables and their derivatives [27, 54, 55, 56, 57].

To compare automatic differentiation with other methods of sensitivity analysis, one obvious choice is to compare the number of equations that are processed in the course of the evaluation of sensitivity information. However, that is not possible due to the inherent characteristics of automatic differentiation. It is to be noted that in automatic differentiation, the starting point is the governing set of system equations, and the process of the evaluation of the derivatives is a purely numerical one.

Essentially, automatic differentiation is an augmentation of the numerical simulation routine that generates the sensitivity information. Therefore it does not benefit from the advantages of the symbolic approach toward sensitivity analysis and cannot formally be classified as a symbolic method.

2.2.6 Bond-graph based sensitivity analysis

Bond graphs represent an unified approach toward modelling multi-domain physical systems. Since the bond graph method is capable of generating the governing equations in an automated fashion, sensitivity analysis based on a bond-graph formulation has received considerable attention. Cabanellas et al. [58] have formulated sensitivity analysis for bond graph models using the concept of Pseudo-bond graphs. Gawthrop [59] has developed the "Sensitivity Bond Graph" to perform sensitivity analysis. Ronco and Gawthrop [60] and Perry et al. [61] have used similar methods to perform parameter estimation, optimization and uncertainty analysis for multi-domain mechatronic systems. Unfortunately, the bond graph method is not very effective when it comes to three-dimensional multi-body systems [17]. Its application to sensitivity analysis of multibody systems is limited due to the complexities involved.

2.2.7 Sensitivity analysis based on graph theory

Savage [6] has presented a method to automatically generate sensitivity models of first and higher orders from a linear graph representing the system. Carr and Savage [34] have extended this method to include cases of nonlinear constitutive equations. These methods were restricted to steady-state problems and scalars were used to represent the variables.

From the discussion of the existing literature it is clear that as the size of the system increases, the established methods start to become more and more inefficient. The finite difference method becomes unstable for different perturbation sizes, direct differentiation becomes computationally expensive for large number of parameters, the adjoint variable method becomes inefficient for large systems, and automatic differentiation misses out on the benefits of symbolic simplification procedures. Furthermore, most of these methods require system-specific considerations.

In the subsequent chapters, a graph-theoretic approach for the generation of sensitivity equations will be presented which is expected to offer flexibility in terms of equation generation and control over the structure and size of the resulting equations.

Chapter 3

Graph-theoretic sensitivity analysis of scalar systems

3.1 Introduction

In scalar systems the through and across variables are represented by scalar quantities. In this chapter graph-theoretic sensitivity analysis will be applied to these systems to demonstrate the applicability of the proposed algorithm for these systems and to illustrate the practical applications of effective sensitivity analysis.

The theoretical background of graph-theoretic sensitivity analysis will be presented using a spring-mass-damper system as an example. In light of this discussion a couple of example problems will be presented. First, a graph-theoretic model of a hydrodynamic torque converter will be chosen where sensitivity analysis will be used to perform design optimization. Next, graph-theoretic sensitivity analysis will be applied to a Nickel-Metal Hydride (NiMH) battery model to perform parametric importance analysis. A brief overview of the software implementation for the example problems will also be presented in this chapter.

3.2 Formulation of the graph-theoretic framework

The key benefits of direct differentiation for the sensitivity analysis of dynamic systems are:

- Conceptual simplicity
- Direct evaluation of required quantities and numerical stability
- Numerically exact results and easy adaptability to different problem structures

But at the same time, direct differentiation generates a large number of equations and becomes intractable for large systems and especially for a large number of parameters. To overcome this problem a graph-theoretic approach is proposed hereby to combine the benefits of graph-theoretic modelling methods and the simplicity of direct differentiation. By combining the symbolic methodology of linear graph with direct differentiation based sensitivity analysis, the proposed algorithm will be shown to achieve computational efficiency while not sacrificing the benefits of a direct method. The goal is to achieve the following:

- Automatic generation of sensitivity equations
- Computational efficiency through branch-chord formulation
- Symbolic representation
- Integration with modelling software for the ease of the user

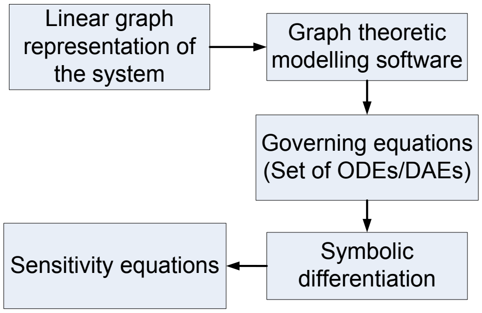

Linear graph representation of the system
Graph theoretic modelling software
Governing equations (Set of ODEs/DAEs)
Symbolic differentiation
Sensitivity equations

Figure 3.1: Generation of sensitivity equations using existing methods

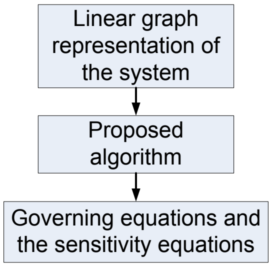

Linear graph representation of the system
Proposed algorithm
Governing equations and the sensitivity equations

Figure 3.2: Generation of sensitivity equations by the proposed method

The existing method for sensitivity analysis is essentially a serial process, where the algorithm for the generation of sensitivity equations is applied to the completely formed governing equations.

Because of the fact that the system equations are assembled before sensitivity equations are generated, this process does not leave much room for improvements. On the other hand, the proposed algorithm generates the sensitivity equations simultaneously with the system equations by approaching the sensitivity problem from a much lower level.

Figure 3.2 shows the schematics of the proposed method. It is expected that this approach would lead to simpler and controllable sensitivity equations, which is very important for efficient computation.

To explain the process of generating the sensitivity equations from a linear graph, one needs to understand the basic steps that generate the system equations in the first place.

Detailed description of this process is presented in chapter 2. In this section, the same process will be used to illustrate the generation of sensitivity equations.

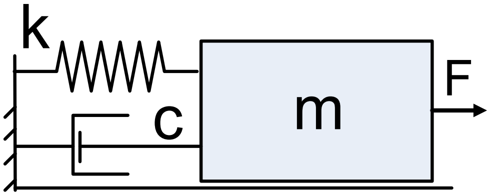


Figure 3.3: A spring-mass-damper system

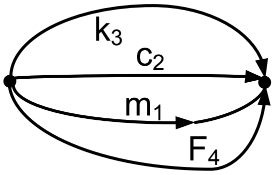


Figure 3.4: Linear graph G_1 for the system

Figure 3.3 shows a simple one-dimensional spring-mass-damper system. Using the formulation presented in chapter 2, the governing equation for this system can be generated from the linear graph G_1 that captures the topology of this system, as shown in figure 3.4.

The linear graph for any dynamic system generates three distinct types of equations.

- F-Cutset equations
- F-Circuit equations
- Terminal equations

Although generally these equations are simplified by substitution and other methods to generate a more compact form of system equations, it is mathematically justified to use these equations together and treat them as the mathematical model of the system.

To obtain the sensitivity equations using graph-theoretical modelling, the method proposes to formulate a second graph G_2 which will yield the sensitivity equations. To determine the structure of G_2 one needs to generate the sensitivity equations using direct differentiation and then formulate the graph G_2 so that the equations can be generated from it. At this point the structure of G_2 is unknown and the following discussion will be used to bridge the gap between the unknown structure of a linear graph and the known set of equations that it is expected to generate.

F-Cutset equations

The F-Cutset Equations are linear relationships involving the through variables associated with the edges of the graph. The general form of the f-cutset equation is

$$\mathbf{A_f}\boldsymbol{\tau} = 0 \tag{3.1}$$

The matrix $\mathbf{A_f}$ is the fundamental cutset matrix which is a constant, and $\boldsymbol{\tau}$ is the vector of through variables. When equation (3.1) is differentiated with respect to a single scalar parameter b, the following equation is obtained.

$$\mathbf{A_f}\frac{\partial\,\boldsymbol{\tau}}{\partial\,b} = \mathbf{A_f}\,\boldsymbol{\tau}_b = 0 \tag{3.2}$$

It is to be noted that equation (3.2) uses the same $\mathbf{A}_f$ matrix as equation (3.1). This means that the sensitivities of the through variables are through variables themselves and the structure of G_2 should be such that its f-cutset matrix would be identical to that of the graph G_1.

F-Circuit equations

Like the f-cutset equations, the f-circuit equations are linear relationships involving the across variables associated with the edges of the graph.

$$\mathbf{B_f}\boldsymbol{\alpha} = 0 \tag{3.3}$$

By direct differentiation, the relationship between the sensitivities of the across variables are determined to be given by the following equation.

$$\mathbf{B_f}\frac{\partial\,\boldsymbol{\alpha}}{\partial\,b} = \mathbf{B_f}\,\boldsymbol{\alpha}_b\!= 0 \tag{3.4}$$

Once again it can be seen that the sensitivities of the across variables are across variables themselves. Also, it is apparent that equation (3.4) requires the graph G_2 to have the same f-circuit matrix as the graph G_1.

The terminal equations

The terminal equations are connections between the through and across variables, usually associated with the same edge. For the sensitivity graph G_2 the terminal equations would be the relationship between $\boldsymbol{\tau}_b$ and $\boldsymbol{\alpha}_b$. These relationships can obtained by differentiating the terminal equations of the graph G_1.

From the above discussion, it is clear that the graph G_2 should have the same topology as the graph G_1, with the same number of nodes and edges. However the through and across variables associated with the edges of graph G_2 will obviously be different from that in G_1

with respect to the chosen model parameter. In fact they will be the corresponding sensitivities. Also the terminal equations of G_2 will be the derivatives of the terminal equations of G_1. Figure 3.5 below shows the sensitivity linear graph G_2 for the spring mass damper system shown in figure 3.3. In the graph shown in figure 3.5 the through and across variables associated with the edges are denoted by $\boldsymbol{\tau}$ and $\boldsymbol{\alpha}$. The subscripts refer to the parameter with respect to which sensitivity analysis is being carried out, and the superscripts refer to the components with which the variables are associated .

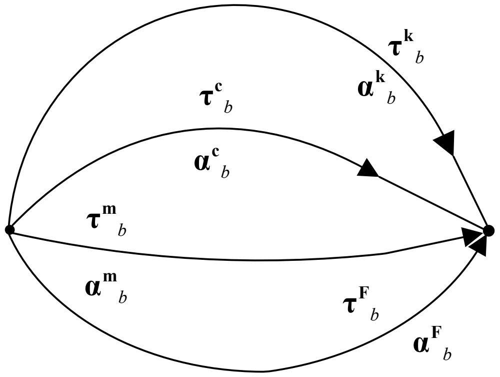

$\boldsymbol{\tau}^{\mathbf{k}}{}_b$
$\boldsymbol{\alpha}^{\mathbf{k}}{}_b$
$\boldsymbol{\tau}^{\mathbf{c}}{}_b$
$\boldsymbol{\alpha}^{\mathbf{c}}{}_b$
$\boldsymbol{\tau}^{\mathbf{m}}{}_b$
$\boldsymbol{\alpha}^{\mathbf{m}}{}_b$
$\boldsymbol{\tau}^{\mathbf{F}}{}_b$
$\boldsymbol{\alpha}^{\mathbf{F}}{}_b$

Figure 3.5: Sensitivity graph (G_2) for the spring-mass-damper system

The sensitivity equations can be generated from G_2 following the same procedure used to generate the system equations from G_1. Also, all the usual techniques of branch-chord formulation can be applied during the equation generation. The following paragraphs will further illustrate this process by using the current example.

$$m\ddot{x} + c\,\dot{x} + k\,(x - l_0) = F(t) \tag{3.5}$$

Equation (3.5) is the governing equation of motion for the system represented by the graph G_1. In this equation, m is the mass of the block, c is the damping coefficient, k is the spring constant, and $F(t)$ is the applied force on the block. This equation can be generated from G_1 using graph-theoretic methods and the following terminal equations.

$$\tau^{\,m} = -m\,\ddot{\alpha}^{\,m} \qquad \tau^{\,F} = F(t) \qquad \tau^{\,c} = -c\,\dot{\alpha}^{\,c} \qquad \tau^{\,k} = -k\left(\alpha^{\,k} - l_0\right) \tag{3.6}$$

The corresponding sensitivity terminal equations can be obtained by differentiating equation (3.6) with respect to the desired model parameter. For this example a general parameter b is chosen and the following sensitivity terminal equations are derived.

$$\begin{array}{rl} \tau_b^m = -\frac{\partial m}{\partial b}\ddot{\alpha}_m - m\ddot{\alpha}_b^m & \tau_b^F = \frac{\partial}{\partial b}F\left(t\right) \\ \tau_b^c = -\frac{\partial c}{\partial b}\dot{\alpha}^c - c\dot{\alpha}_b^c & \tau_b^k = -\frac{\partial k}{\partial b}\left(\alpha^k - l_0\right) - k\left(\alpha_b^k - \frac{\partial l_0}{\partial b}\right) \end{array} \tag{3.7}$$

For the specific case when $b = k$, the sensitivity terminal equations take the form shown in equation (3.8).

$$\begin{array}{ll} \tau_k^m = -m\,\ddot{\alpha}_k^m & \tau_k^m = 0 \\ \tau_k^c = -c\,\dot{\alpha}_k^c & \tau_k^k = -\left(\alpha^k - l_0\right) - k\alpha_k^k \end{array} \tag{3.8}$$

To generate the sensitivity equations using the same branch-chord formulation demonstrated in chapter 2, the same edge m is chosen as the tree for the graph G_2. This results in the following f-cutset and f-circuit equations, which are derived assuming a general model parameter b.

$$\begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \begin{Bmatrix} \tau_b^m \\ \tau_b^k \\ \tau_b^c \\ \tau_b^F \end{Bmatrix} = 0 \quad \textit{and} \quad \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \alpha_b^m \\ \alpha_b^k \\ \alpha_b^c \\ \alpha_b^F \end{Bmatrix} = 0 \tag{3.9}$$

To derive the final set of equations, we assume that the parameter in question is the spring constant k. By combining equations (3.9) and (3.8) and using the substitutions $b = k$, $\alpha^m = x$, and $\alpha_k^m = x_k$ the sensitivity equation for this system is generated.

$$\begin{aligned} &\tau_k^m + \tau_k^k + \tau_k^c + \tau_k^F = 0 \\ &\to -m\ddot{\alpha}_k^m - c\,\dot{\alpha}_k^c - \left(\alpha^k - l_0\right) - k\,\alpha_k^k = 0 \\ &\to -m\ddot{\alpha}_k^m - c\,\dot{\alpha}_k^m - \left(\alpha^m - l_0\right) - k\,\alpha_k^m = 0 \\ &\to m\ddot{x}_k + c\,\dot{x}_k + k\,x_k = \left(l_0 - x\right) \end{aligned} \tag{3.10}$$

The validity of this equation can be easily verified by differentiating equation (3.5) with respect to the model parameter k, which yields identical results.

The example shown here demonstrates the applicability of graph-theoretic sensitivity analysis on scalar systems. The important point to note is that *the algorithm follows the exact same path as that of generating the system equations and thus can be readily implemented using modelling software that uses graph-theoretic modelling techniques.*

It is to be noted that although the sensitivity equations were derived from the second graph G_2, it is not independent of the system graph G_1. This is clearly evident from the constitutive equations of the graph G_2 as shown in equation (3.7), where the through and across variables of the graph G_1 appears along with the through and across variables of the graph G_2.

In effect, the system response can be considered as the factor that drives the sensitivity equations. For the example of first order sensitivity analysis of the presented spring-mass-damper mechanism, this dependence is clearly illustrated by the final form of the sensitivity equation. As shown in (3.10), the position x, an across variable from the system graph, appears in the right hand side and act as what can be described as a *sensitivity driver*.

In the subsequent sections in this chapter, the graph-theoretic sensitivity analysis will be applied to different scalar systems to demonstrate its application and suitability. First, an automotive hydrodynamic torque converter model will be considered. Graph-theoretic sensitivity analysis will be applied to the model to perform design optimization and parametric importance analysis. Next, an electro-chemical model of a Ni-MH battery will be considered to demonstrate the application of sensitivity analysis for the identification of important parameters.

3.3 Example of a hydro-mechanical system: An automotive torque converter

From the assistive devices that help medical professionals move patients with disabilities, to the actuators that control the flying surfaces of a modern day jet, there are countless instances of hydro-mechanical systems encountered in the modern world. However different their respective applications can be, the common theme in these systems is the fact that they convert energy back and forth between the mechanical and the hydraulic domains. Efficient, accurate and effective modelling and analysis of hydro-mechanical systems is therefore essential for the design, development and manufacturing of many practical devices.

In this example, an automotive hydro-mechanical torque converter is chosen and the linear graph approach is used to model the system and perform sensitivity analysis.

A torque converter is an essential component for an automobile with automatic transmission. Its function is to transmit torque from the engine to the drive shaft while allowing discrepancies between the angular velocities on each side. For this aspect, it serves the function of a mechanical friction clutch. In addition to this, a torque converter makes it possible to achieve torque multiplication that results in better acceleration and it also provides some degree of vibration isolation between the engine and the transmission.

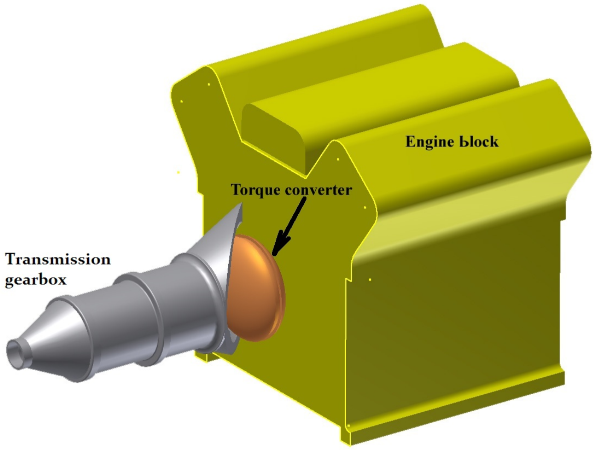


Figure 3.6: Location of a torque converter in a transmission system

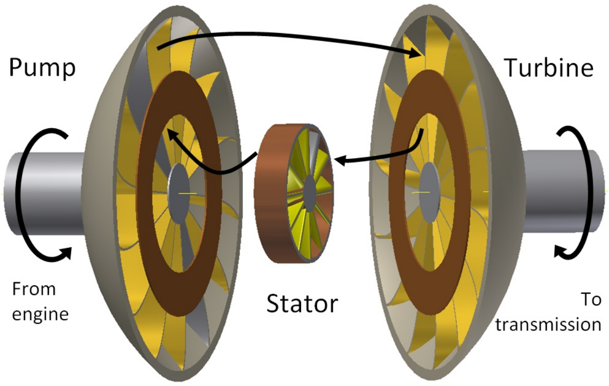


Figure 3.7: Schematic diagram of a hydro-mechanical torque converter

The hydro-mechanical torque converter is the most common design used for this purpose. It has three main components, known as the pump, the turbine, and the stator. The pump and the turbines have vanes built onto them, that can channel the flow of hydraulic fluid in the radial direction. The stator is a circular array of blades that deflect the flow of fluid from the turbine and align them with the vanes of the pump. The pump section is rigidly connected to the engine flywheel and the turbine section is connected to the drive shaft leading to the transmission gearbox. The stator is connected to the automobile chassis using a special revolute joint that allows motion in one direction only.

3.3.1 Graph-theoretic modelling of a torque converter

To model a torque converter, the first step is to understand its function and structure. A torque converter works between the engine and the rest of the transmission system. Figure 3.6 shows the location of a torque converter in an automotive transmission system.

The operation of a torque converter can be described as that of a hydraulic pump driving a hydraulic turbine. The torque from the engine drives the pump which imparts energy to the hydraulic fluid in the system; the hydraulic fluid flows through the vaned construction of the turbine and makes it rotate thereby transmitting the torque coming from the engine toward the rest of the transmission system. From the standpoint of energy transmission, the pump converts mechanical energy into hydraulic energy and the turbine converts it back to the mechanical domain. A linear graph can be used to represent the flow of power across different domains of the system and account for the storage and losses incurred in the process.

The ratio of the angular speeds of the turbine ω_t, and the pump ω_p is known as the speed ratio. It is denoted as ω_r as shown in equation (3.11).

$$\omega_r = (\omega_t/\omega_p) \tag{3.11}$$

Depending on the value of ω_r, a torque converter can act in two different modes of operation. When ω_r is less than unity, that is when the angular velocity of the pump is greater than that of the turbine, the pump drives the system and torque is transmitted from the pump to the turbine. This is known as the forward flow mode. This mode of operation corresponds to the acceleration of the vehicle.

On the other hand, when ω_r is greater than unity, the turbine drives the system and torque is transmitted from the transmission to the engine. This is known as the backward or reverse flow mode. This corresponds to engine braking or coasting operation of the vehicle.

Figure 3.7 shows the schematic diagram of a typical torque converter. The mechanical components and the path followed by the hydraulic fluid during the forward flow mode are shown in the diagram.

Conventional modelling methods require manual derivation of equations [62], which can be tedious and prone to errors. By employing linear graph theory to model the behaviour of the torque converter, it is possible to generate the governing equations in an automated fashion [63]. The linear graph-theoretic approach makes it possible to encapsulate the model as a subsystem, making it easier to reuse it for different modelling assignments [64].

Also, from the standpoint of system analysis, a linear graph model of a torque converter enables one to perform graph-theoretic sensitivity analysis of the torque converter.

Basic Physics

To model a dynamic system using linear graph theory, one must capture the underlying physics of the system into a set of constitutive equations. For that purpose, it is necessary to identify the through and across variables of the system.

In a torque converter, energy is transmitted through both mechanical and hydraulic domains. As a result, through and across variables from both domains are expected to appear in the governing equations. The through and across variables associated with the two domains are given in table 3.1.

Table 3.1: Through and across variables for the model

	Through Variable	Across Variable
Mechanical Domain	Torque (τ)	Angular velocity (ω)
Hydraulic Domain	Flow rate (Q)	Pressure drop (p)

A torque converter can be considered as an assembly of three impellers that are basically circular arrangements of vanes. It is to be noted that although the physical dimensions and properties of these impellers are different from each other, the underlying dynamics are the same.

In this section, the equations governing the state variables of a subsystem, which comprises one of these impellers, are derived. Once the equations are derived, they can be used to formulate the linear graph that would represent the impeller subsystem. The full model of the torque converter can then be formulated by combining three such subsystems using the method outlined by Schmitke and McPhee [64]. At this point, it is important to outline the assumptions made during the course of the derivation.

- The curved profile of the actual impeller is flattened in places to come up with a simplified impeller profile as outlined in figure 3.8.

- The cross-sectional net torus flow area is assumed to be constant along the radial line. This implies spatially constant magnitude of the radial flow velocity.
- The three-dimensional flow is approximated by the flow along a median streamline.
- The vanes have negligible thicknesses.
- Thermal effects are neglected.

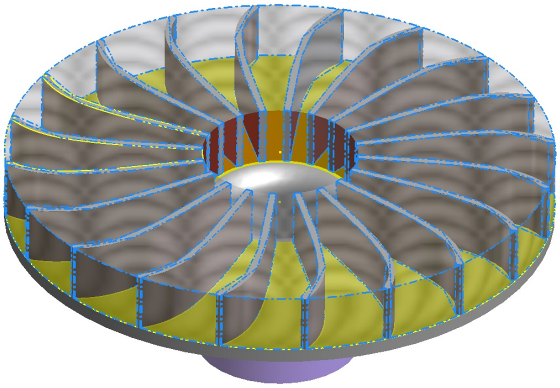


Figure 3.8: Simplified impeller profile

Assuming the impeller is completely submerged in fluid, a cylindrical control volume can be used to derive the dynamic equations of the system. The control volume is shown in figure 3.8 as the shaded volume.

Due to angular symmetry of the impeller, the governing equations can be generated by considering the flow path between two consecutive vanes. Figure 3.9 shows an example of such a section. The radial and tangential unit vectors, $\hat{i}_r$ and $\hat{i}_\varphi$ are shown at the centre, along with the mechanical through and across variables of the system, i.e., ω and τ. The inner and outer radii of the impeller are R_1 and R_2.

The flow is approximated by the flow along the median streamline CD as shown in figure 3.8. The flow path is characterized by the flow area A and the curvature of the vane. The curvature is given by the quantity α, which represents the angle between the radial line and the tangent to the curve CD at any radial location, $\alpha = \alpha\left(r\right)$.

Physically, the dynamics of the motion of the impeller can be described as follows:

Applied external torque τ forces the impeller to rotate with angular velocity ω. This forces the hydraulic fluid radially outward and starts a volumetric flow rate Q. The flow of the hydraulic fluid gets deflected by the vanes and as a result a pressure difference is created between the point C and D given by $(p_C - p_D)$. In return the hydraulic fluid produces a reactive torque on the impeller.

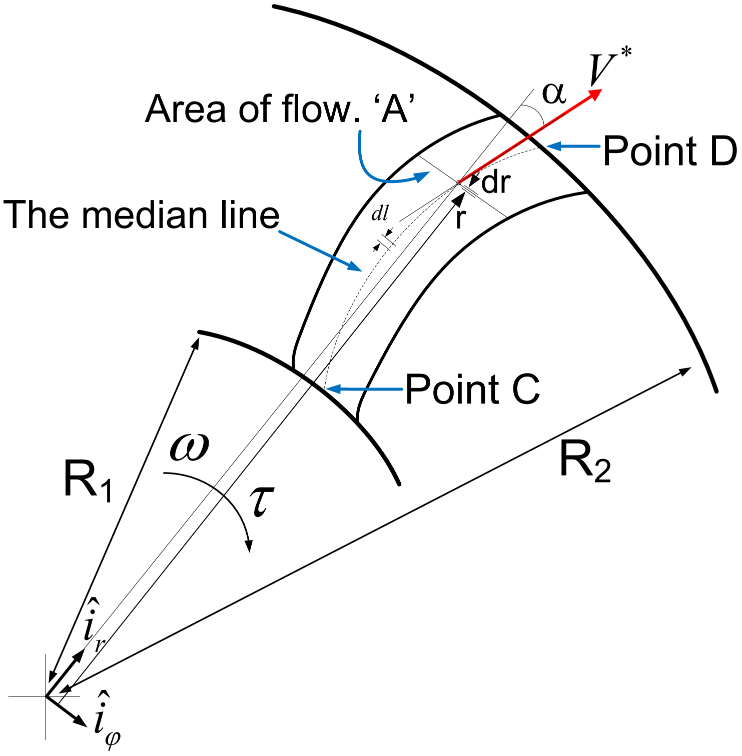


Figure 3.9: Flow path over the impeller

The net torque acting on the control volume is given by equation (3.12).

$$\tau = J_m\dot{\omega} + J_f\dot{\omega} + \psi Q + \rho\, S\, \dot{Q} \tag{3.12}$$

where

α_e: Vane angle at point D

α_e^{back}: Angle of the vane leading to point C

ω_{back}: Assuming a rotating vaned structure from which fluid enters the impeller at point C, the angular velocity of that preceding structure is denoted as ω_{back}.

J_f: Polar moment of inertia of the fluid mass, $J_f = \underset{\forall_{total}}{\iiint} \rho\, r^2 d\forall$

S: Vane profile parameter, $S = \int r \tan\alpha \; dl$

and

$$\psi = \rho \left(\omega R_2^2 + R_2 \frac{Q}{A} \tan \alpha_e - \omega_{back} R_1^2 - R_1 \frac{Q}{A} \tan \alpha_e^{back} \right) \tag{3.13}$$

The pressure difference between the points C and D is given by equation (3.14)

$$(p_C - p_D) = -\psi\,\omega + \frac{\rho L^f}{A}\dot{Q} + \rho\,S\,\dot{\omega} + \frac{P_{shock}}{Q} + \frac{P_{flow}}{Q} \tag{3.14}$$

where

L^f: Vane profile constant $L^f = \int\limits_{\text{Streamline}} \sec^2\alpha\,dl$

P_{shock}: Rate of shock losses (vide equation (B.28))

P_{flow}: Rate of flow losses (vide equation (B.26))

In equation (3.12), the through variable of the mechanical domain is expressed as a function of the through variable of the hydraulic domain and the across variable of the mechanical domain. Similarly in equation (3.14) the across variable of the hydraulic domain is expressed as a function of the through variable of the hydraulic domain and across variable of the mechanical domain.

These two equations form a set of constitutive equations that can be worked into a graph-theoretic framework, which would allow the impeller to be modelled as a standalone subsystem. By proper assembly of similar subsystems, a model of a torque converter can be constructed.

Graph-theoretic framework

To construct a graph-theoretic model of the impeller, it is necessary to identify the natures of the components of the model.

Since the system has two physical domains associated with it, the corresponding linear graph model must have two non-connected sub-graphs. By considering equation (3.12), the following components for the mechanical domain of the model can be clearly identified.

Inertia elements: The terms $-J_m\,\dot{\omega}$ and $-J_f\,\dot{\omega}$ are from the inertia effects of the impeller and the fluid contained therein. These are the inertia elements.

Transducer elements: The terms $-\psi Q$ and $\rho S\dot{Q}$ refer to transducer elements that connect through variables in mechanical domain (τ) to the through variables in the hydraulic domain (Q). As such there are two transducer elements in this domain.

For the hydraulic domain, the following components are identified from equation (3.14).

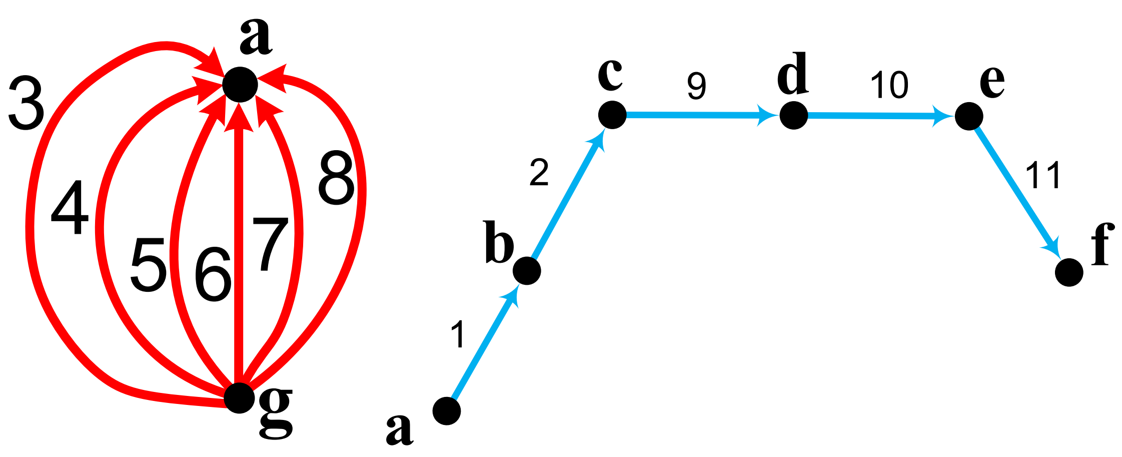
Inertia elements: The term $\frac{\rho L^f}{A}\dot{Q}$ corresponds to a fluid inertia element.

Loss elements 1: The terms $\frac{P_{shock}}{Q}$ and $\frac{P_{flow}}{Q}$ correspond to pressure loss elements.

Transducer elements: The terms $-\psi\,\omega$ and $\rho\,S\,\dot{\omega}$ are the transducer elements that connects the across variable of the hydraulic domain (p) to the across variable of the mechanical domain (ω).

From equations (3.12) and (3.14) it is also clear that, for the mechanical domain, all the elements share the same across variable, i.e., the angular velocity ω and for the hydraulic domain all the elements share the same through variable, i.e., the volumetric flow rate Q.

From this information, a linear graph representation for the impeller system can be constructed. Figures 3.10a and 3.10b show the linear graph for the hydro-mechanical impeller system.



(a) Mechanical Domain

(b) Hydraulic Domain

Figure 3.10: Linear graph model for the hydro-mechanical impeller

In the mechanical domain graph, node g is the ground node, i.e, the inertial reference frame. Node a represents the reference frame fixed to the center of the impeller. The edges correspond to the various components of the mechanical domain:

Edge 3: This edge corresponds to a transducer element. The constitutive equation for this edge is $\tau_3 = -\psi Q_1$.

Edge 4: This edge corresponds to the second transducer element. The constitutive equation for this edge is $\tau_4 = \rho S \dot{Q}_2$.

Edges 5, 7: Inertia elements, $\tau_5 = -J_m \, \dot{\omega}_5$ and $\tau_7 = -J_f \, \dot{\omega}_7$.

Edge 6: This edge corresponds to the revolute joint between the ground and the impeller. For ideal joints the constitutive equation is $\tau_6 = 0$.

Edge 8: This edge refers to the external applied torque on the impeller. The constitutive equation is $\tau_8 = \tau(t)$.

In the hydraulic domain graph, the nodes represent points in the fluid along the flow path, where pressure measurements are obtained. For this graph the following edges can be defined:

Edges 1, 2: These edges correspond to the transducer elements. The constitutive equations for these edges are $p_1 = -\psi \, \omega_3$ and $p_2 = \rho \, S \, \dot{\omega}_4$.

Edges 9, 10: These edges are associated with the loss elements. The constitutive equations are $p_9 = P_{shock}/Q_9$ and $p_{10} = P_{flow}/Q_{10}$

Edge 11: This edge corresponds to the fluid inertia element. The pressure difference is created by the inertia forces acting on the accelerating fluid mass. The equation is $p_{11} = \frac{\rho L^f}{A} \dot{Q}_{11}$.

The complete representation of the hydro-mechanical impeller is illustrated in figure 3.10. It captures the relevant dynamic effects and represents the topology of the system. Using the presented impeller subsystem to model the pump, the turbine, and the stator and assembling them in the proper order, a complete model of a torque converter is obtained.

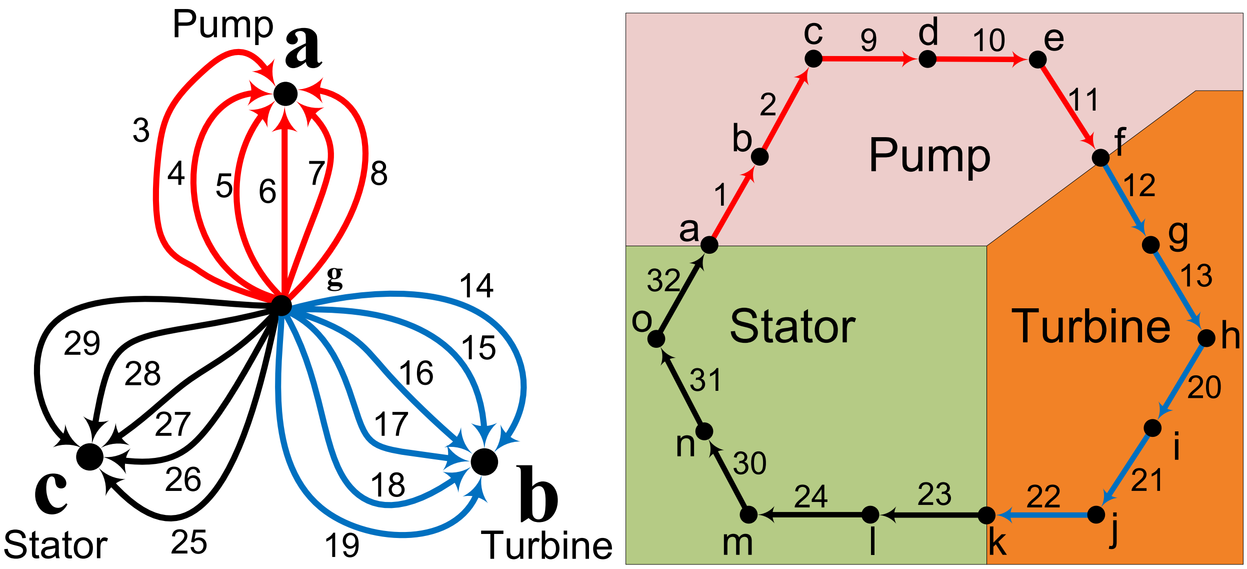
Figure 3.11 shows the linear graphs of the torque-converter for both mechanical and hydraulic domains. The sections of the graph that represent the three components of the torque converter are marked in the figure. They share identical topologies but have different constitutive equations, which reflects the differences in their physical properties.

In a torque converter, external torques are applied to the pump and the turbine. Edges 8 and 19 are included to account for $\tau_{pump}(t)$ and $\tau_{turbine}(t)$. The pump and the turbine are connected to the mounting through revolute joints. Edges 6 and 16 represent these revolute joints. In this example, ideal revolute joints were used for modelling and analysis.

Edge 28 represents the one-way revolute joint used to connect the stator to the casing. The corresponding constitutive equations are

$$\omega_{28} = \begin{cases} 0 & \tau_{28} > 0 \\ \omega_s\left(t\right) & \tau_{28} < 0 \end{cases} \quad \mathit{and} \quad \tau_{28} = \begin{cases} \tau_s\left(t\right) & \tau_{28} > 0 \\ 0 & \tau_{28} < 0 \end{cases} \tag{3.15}$$

The presence of ideal multi-domain transducers are clearly illustrated in the linear graph shown in figure 3.11.



(a) Mechanical Domain

(b) Hydraulic Domain

Figure 3.11: Linear graph model for the torque converter

Edges 1 and 3 constitute one such transducer. The terminal equations for these edges can be written in a compact matrix form.

$$\begin{bmatrix} \tau_3 \\ \omega_3 \end{bmatrix} = \begin{bmatrix} \psi^p & 0 \\ 0 & -\frac{1}{\psi^p} \end{bmatrix} \begin{bmatrix} Q_1 \\ p_1 \end{bmatrix} \tag{3.16}$$

It is clearly evident that for edges 1 and 3, the quantity $\tau_3\omega_3$ is equal in magnitude with the quantity $Q_1 p_1$. As a result, the net power loss between these edges remains zero, as expected in case of an ideal transducer.

In this example, the letters p, t, and s are used as subscripts to refer to the pump, the turbine, and the stator. For example, the symbol ψ^p refers to the modulated transducer coefficient ψ^p derived specifically for the pump element.

For operation under forward flow mode, the component that leads up to the pump is the stator. Hence the expression for ψ^p is derived from equation (B.22) by replacing R_2 by R_p, R_1 by R_s, ω by ω_5, ω_{back} by ω_{27}, α_e by α_p and α_e^{back} by α_s. Similar substitutions are used to derive ψ^t and ψ^s.

$$\psi^p = \rho\left[R_p^2\omega_5 + R_p\frac{Q_1}{A}\tan\alpha_p - R_s^2\omega_{27} - R_s\frac{Q_1}{A}\tan\alpha_s\right] \tag{3.17}$$

$$\psi^t = \rho\left[R_t^2\omega_{16} + R_t\frac{Q_{12}}{A}\tan\alpha_t - R_p^2\omega_5 - R_p\frac{Q_{12}}{A}\tan\alpha_p\right] \tag{3.18}$$

$$\psi^s = \rho\left[R_s^2\omega_{27} + R_s\frac{Q_{23}}{A}\tan\alpha_s - R_t^2\omega_{16} - R_t\frac{Q_{23}}{A}\tan\alpha_t\right] \tag{3.19}$$

where α_p, α_t, and α_s are the vane angles at the exit end of the pump, the turbine, and the stator, respectively. The exit end is defined with respect to the positive flow rate Q.

The other type of transducer used in this model is the ones shown between edges 2 and 4, 13 and 15, and 24 and 26. In these transducers, the through variable of the mechanical domain τ is expressed as a function of the derivative of the through variable of the hydraulic domain $\dot{Q}$ and the across variable of the hydraulic domain is expressed as a function of the derivative of the across variable of the mechanical domain $\dot{\omega}$.

The compact matrix notation of the constitutive equations for edges 2 and 4 are given below to illustrate this form.

$$\begin{bmatrix} \tau_4 \\ \dot{\omega}_4 \end{bmatrix} = \begin{bmatrix} \rho S_p & 0 \\ 0 & \frac{1}{\rho S_p} \end{bmatrix} \begin{bmatrix} \dot{Q}_2 \\ p_2 \end{bmatrix} \tag{3.20}$$

Once the graph-theoretic model is fully described in terms of its topology (figure 3.11) and the terminal equations, the graph-theoretic method [63] and principle of orthogonality [22] are used to derive the system equations for the torque converter.

Backward flow mode

The derivation shown so far assumes that the flow rate Q is positive, i.e., the pump is driving the turbine. The terms entry and exit vane angles are defined with respect to this positive direction of flow.

The schematic diagram shown in figure 3.12 describes the relevant entry angles $\beta_{p,t,s}$ and exit angles $\alpha_{p,t,s}$. When the flow direction is reversed, the torque converter operates in backward flow mode. Physically this mode corresponds to the operation of engine braking in a vehicle. During this mode, the turbine drives the pump and as a result, the entry and exit angles change places between themselves. The equations describing the dynamics of the torque converter must be changed accordingly.

For example, the modulated transducer coefficients $\psi^{p,t,s}$ become different for backward flow mode. During this mode, instead of the stator, the turbine is the component that leads up to the pump. Also the radii of the inlet and outlet surfaces get changed.

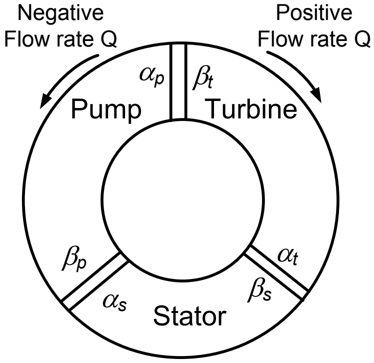


Figure 3.12: Schematic diagram of the vane angles

The modified expression for ψ^p is given in equation (3.21).

$$\psi^p = \rho \left[R_p^2 \omega_{16} + R_p \frac{Q_1}{A} \tan \beta_t - R_s^2 \omega_5 - R_s \frac{Q_1}{A} \tan \beta_p \right] \tag{3.21}$$

Modified expressions for ψ^t, ψ^s, P_{shock}, and P_{flow} can also be derived using similar substitutions.

Derivation of system equations

To derive the system equations a branch-chord formulation [5] is used. Since there are two graphs for this system, two separate trees are selected for equation generation.

Domain	**Branches (Edge No)**	**Chords (Edge No)**
Mechanical	5 , 16, 27	3, 4, 6 - 8, 14, 15, 17 - 19, 25, 26, 28, 29
Hydraulic	1, 2, 9 - 13, 20 - 24, 30, 31	32

Table 3.2: Selection scheme for the branch-chord formulation

From the linear graph shown in figure 3.11 and the tree selection scheme shown in table 3.2, the cutset and circuit equations of both mechanical and hydraulic domain of the torque converter system are obtained. Using the branch and chord transformation equations the system equations are expressed in terms of branch across variables and chord through variables only. For the particular choice of branch selection as illustrated in table 3.2, the equations are generated in terms of the three angular velocities ω_5, ω_{16}, ω_{27} , three torques τ_6,τ_{17}, τ_{28}, and the flow rate Q_{32}. The symbols representing these quantities are changed according to the chart shown in table 3.3.

Variable name	**Changed symbol**
ω_5	$\omega_p(t)$
ω_{16}	$\omega_t(t)$
ω_{27}	$\omega_s(t)$
τ_6	$\tau_p(t)$
τ_{17}	$\tau_t(t)$
τ_{28}	$\tau_s(t)$
Q_{32}	$Q(t)$

Table 3.3: State variables for the torque converter model

Mechanical cutset equations

$$\begin{aligned} \tau_3 + \tau_4 + \tau_5 + \tau_6 + \tau_7 + \tau_8 &= 0 \\ \tau_{14} + \tau_{15} + \tau_{16} + \tau_{17} + \tau_{18} + \tau_{19} &= 0 \\ \tau_{25} + \tau_{26} + \tau_{27} + \tau_{28} + \tau_{29} &= 0 \end{aligned} \tag{3.22}$$

Mechanical circuit equations

$$\begin{aligned} &\omega_3 = \omega_4 = \omega_5 = \omega_6 = \omega_7 = \omega_8 \\ &\omega_{14} = \omega_{15} = \omega_{16} = \omega_{17} = \omega_{18} = \omega_{19} \\ &\omega_{25} = \omega_{26} = \omega_{27} = \omega_{28} = \omega_{29} \end{aligned} \tag{3.23}$$

Hydraulic cutset equations

$$Q_{32} = Q_{[1,2,9,10,11,12,13,20,21,22,23,24,30,31]} \tag{3.24}$$

Hydraulic circuit equations

$$\begin{gathered} \sum p_i = 0 \\ i = 1, 2, 9, 10, 11, 12, 13, 20, 21, 22, 23, 24, 30, 31, 32 \end{gathered} \tag{3.25}$$

The torque converter system is fully determined by the set of equations formed by combining equations (3.22) to (3.25) and the constitutive equations described in the previous section. However, to generate equations in terms of a smaller set of state variables, branch and chord transformation equations are used.

The generated system equations are expressed in matrix format in equation 3.26.

$$\mathbf{M}\dot{\boldsymbol{q}} + \mathbf{G}\boldsymbol{q} + \boldsymbol{\xi} - \varsigma = 0$$

$$\tau_s = \begin{cases} \tau_s & \tau_s > 0 \\ 0 & \tau_s < 0 \end{cases} \quad \text{and} \quad \omega_s = \begin{cases} 0 & \tau_s > 0 \\ \omega_s & \tau_s < 0 \end{cases} \tag{3.26}$$

where the state vector $\boldsymbol{q}$ and the mass matrix $\mathbf{M}$ is given by

$$\boldsymbol{q} = \left\{ \begin{array}{cccc} \omega_p & \omega_t & \omega_s & Q \end{array} \right\}^T \tag{3.27}$$

$$\mathbf{M} = \begin{bmatrix} J_p & 0 & 0 & \rho S_p \\ 0 & J_t & 0 & \rho S_t \\ 0 & 0 & J_s & \rho S_s \\ \rho S_p & \rho S_t & \rho S_s & \frac{\rho}{A}(L_p^f + L_t^f + L_s^f) \end{bmatrix} \tag{3.28}$$

where $L_{p,t,s}^f$ are the equivalent fluid lengths evaluated using equation (B.32) and $J_{p,t,s}$ are the sum of J_m and J_f for the pump, the turbine and the stator respectively.

The matrix $\mathbf{G}$ is an antisymmetric coupling matrix. It is expressed in terms of the modulated transducer coefficients as illustrated in equation (3.29). The expressions for $\psi^{p,t,s}$ are different for forward and backward flow simulations.

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 & \psi^p \\ 0 & 0 & 0 & \psi^t \\ 0 & 0 & 0 & \psi^s \\ -\psi^p & -\psi^t & -\psi^s & 0 \end{bmatrix} \tag{3.29}$$

The vector $\boldsymbol{\xi}$ accounts for the losses in the system.

$$\boldsymbol{\xi} = \begin{bmatrix} 0 & 0 & 0 & P_L \end{bmatrix}^T \tag{3.30}$$

where P_L is the sum of pressure losses due to fluid friction and shock effects encountered around the flow path of the fluid. The expressions for P^i_{shock} and P^i_{flow} are derived using the equations (B.28) and (B.26).

$$P_L = \frac{1}{Q}\left[\sum_{i=p,t,s}\left(P^i_{shock} + P^i_{flow}\right)\right] \tag{3.31}$$

The external force vector $\boldsymbol{\varsigma}$ is given by

$$\boldsymbol{\varsigma} = \begin{bmatrix} \tau_p & \tau_t & \tau_s & 0 \end{bmatrix}^T \tag{3.32}$$

System response

To characterize and quantify the response of the torque converter, the following quantities are defined.

Speed ratio is the ratio of the turbine and pump speeds as shown in equation (3.11).

Efficiency is the fraction of the input power that is available as output power.

$$\eta = \frac{|\tau_t \omega_t|}{|\tau_p \omega_p|} \tag{3.33}$$

Torque ratio is the ratio of turbine torque and pump torque.

$$T_r = \frac{\tau_t}{\tau_p} \tag{3.34}$$

Capacity factor is the ability of the torque converter to absorb and transmit torque. For successful operation, a torque converter must be matched with a transmission system with comparable range of capacity factor [65].

$$K_p = \frac{\omega_p}{\sqrt{\tau_p}} \tag{3.35}$$

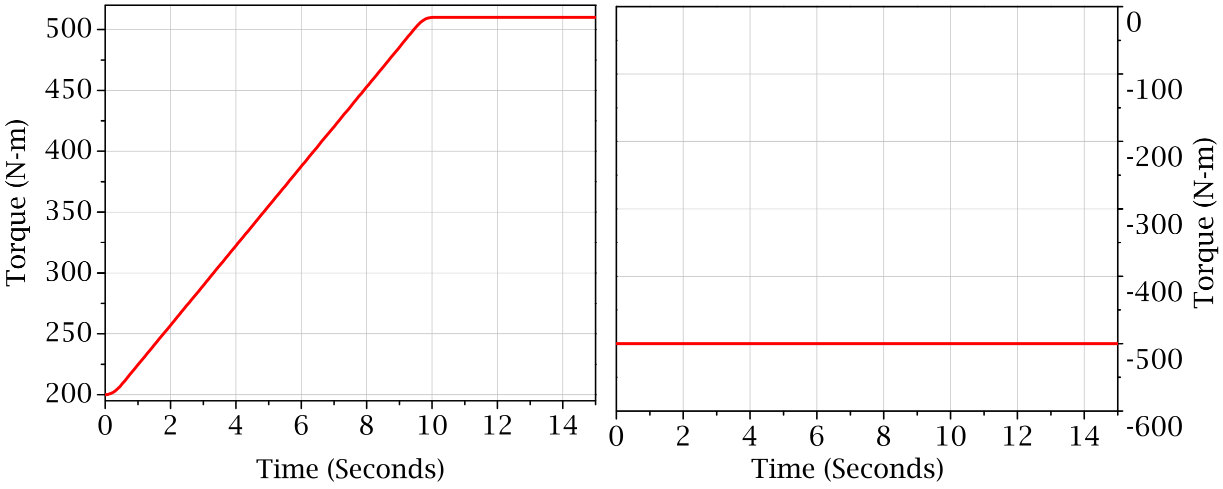
Equation (3.26) is a set of first-order differential equations. To ensure proper results, proper initial conditions of the state variables must be specified which can represent the initial configuration of the system. For this example, the system is initialized with the turbine and stator at rest and the pump rotating with non-zero angular speed. The initial values of the state variables are given in table 3.4. Numerical values for the model parameters of equation (3.26) are obtained from existing literature [66], [67].

A complete list of parameters are given in table A.1.

States	Initial Values	Unit
$Q(0)$	0.13	m^3/s
$\omega_s(0)$	0	rad/s
$\omega_p(0)$	280	rad/s
$\omega_t(0)$	0	rad/s

Table 3.4: Initial conditions used for simulation and analysis

The pump torque τ_p and the turbine torque τ_t are taken as inputs. The pump torque is increased gradually, where the turbine torque is kept constant. Figure 3.13 below shows the variation of τ_p and τ_t with time.



(a) Pump torque (τ_p)

(b) Turbine torque (τ_t)

Figure 3.13: τ_p and τ_t as functions of time

Using the inputs shown in figure 3.13, the equations shown in (3.26) are solved numerically using the `dsolve[numeric]` routine from Maple. Fehlberg fourth-fifth order Runge-Kutta formulation was used to solve the set of DAEs. The maximum function evaluation was set at $maxfun = 100000$ with default error tolerance at $\epsilon = 1 \times 10^{-6}$.

Figure 3.14 shows the variation of the pump speed ω_p and turbine speed ω_t as the speed ratio increases during the simulation. To obtain this plot, the simulation was allowed to run until the speed ratio reached $\omega_r = 0.8$. The difference between the pump speed and the turbine speed is also plotted on the same axes.

The figure shows that the pump speed increases from the initial 280 rad/s. Also it shows that as the pump speed increases, the turbine speeds up from its initial stationary state and eventually the system reaches the coupling point when the stator starts to rotate. These are expected behaviour of a torque converter in forward acceleration mode. As evident from figure 3.14, the increasing pump torque forces the pump to speed up, which forces the turbine to speed up due to the fluid coupling action. As the turbine speeds up, the speed ratio increases from an initial zero value. The variation of speed ratio with time is illustrated in figure 3.15.

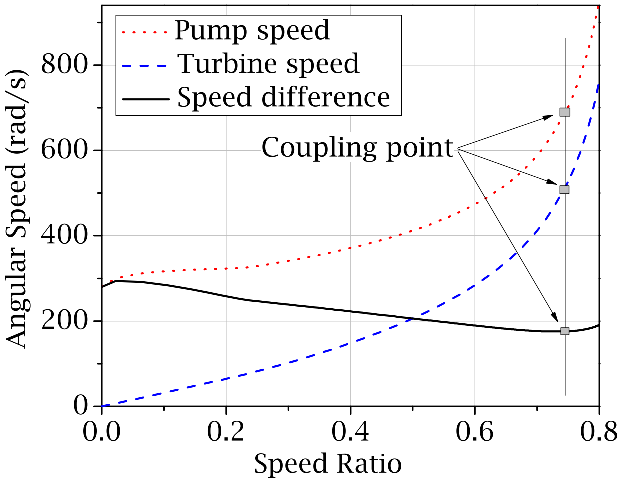


Figure 3.14: Variation of ω_p and ω_t with ω_r

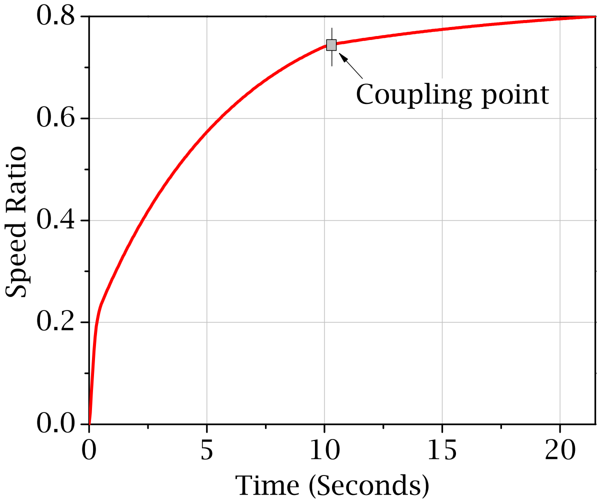


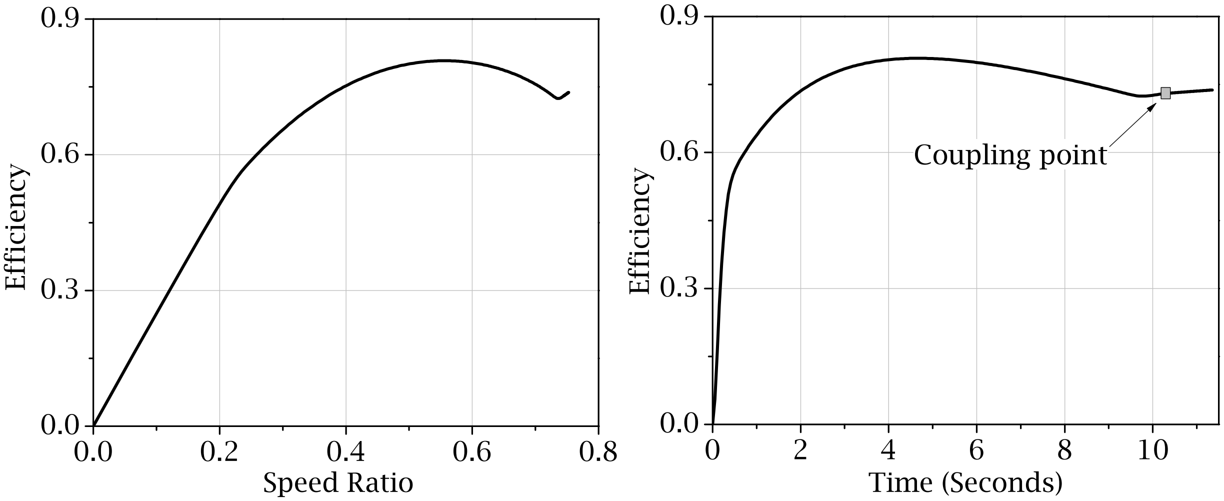
Figure 3.15: Variation of speed ratio

Due to the presence of losses, the fluid coupling cannot force ω_t to match up with ω_p completely. As a result, the difference between ω_p and ω_t reaches a minimum around the coupling point, which is defined as the point in time when the stator starts to rotate. If the simulation is continued past the coupling point, the individual magnitudes of ω_p and ω_t increase while the difference between them settles. As a result of this, the speed ratio of a torque converter approaches unity asymptotically. This is evident from figure 3.15 where the curve becomes increasingly flatter as time progresses.

Usually the speed ratio is used to get a qualitative description of the difference between the angular speed of the turbine and that of the pump. However, this notion can be misleading at times. The plot of the differences between the pump speed and the turbine speed shows some interesting features. From figure 3.14, it can be clearly noticed that before the coupling point, the difference between the turbine and pump speed decreases steadily. At this stage, increasing speed ratio is associated with the phenomenon of the turbine catching up with the pump. However, after the coupling point is reached, the difference between the pump speed and the turbine speed starts to increase. Contrary to the observed trend in the pre-coupling stage, a higher speed ratio at this stage actually represents the two speeds moving away from each other.

The increase in the difference between the pump speed and the turbine speed in the post-coupling phase can be explained by considering the dynamics of the system. Before the coupling point, the difference between the turbine speed and pump speed decrease due to fluid coupling. At this point the only components storing energy are the pump, the turbine and the hydraulic fluid. However after the coupling point is reached, the stator starts to rotate and stores some energy due to its angular motion. As the stator speeds up, more energy is channeled into the stator as its kinetic energy. This increases the difference between the pump speed and turbine speed.

With this observation, it is possible to justify the choice of a termination criterion for the simulation. Because of the presence of the hydraulic and mechanical losses, it is inefficient to operate the torque converter in the coupling mode past the coupling point.



(a) Pump torque (τ_p)

(b) Turbine torque (τ_t)

Figure 3.16: τ_p and τ_t as functions of time

This is why a lock-up clutch is often used to clamp the pump and the turbine together and eliminate the hydraulic coupling action and the associated losses. To perform the locking operation, it is desirable that the turbine speed and the pump speed are as close to each other as possible. Since the difference between these speeds increase in the post-coupling stage, it is therefore logical to activate the lock-up clutch right after the coupling occurs. For this study the simulations were terminated when the stator speed reached $\omega_s = 20\, rad/s$.

The efficiency of the process is an important issue for the design and operation of a torque converter. Figure 3.16b shows the variation of efficiency during the simulation. The plot shows that the efficiency increases from zero in the beginning to a maximum value around half way through the simulation, but right before the coupling point, it decreases a little. Right after the coupling point, the efficiency settles to a stable value. Since the speed ratio curve also becomes flat in this region, the efficiency versus speed ratio plot becomes somewhat linear, as shown in the plot in figure 3.16a.

Figure 3.17 shows the input and the output powers of the system. These quantities are obtained by multiplying the torque and the angular speed of the pump and the turbine respectively.

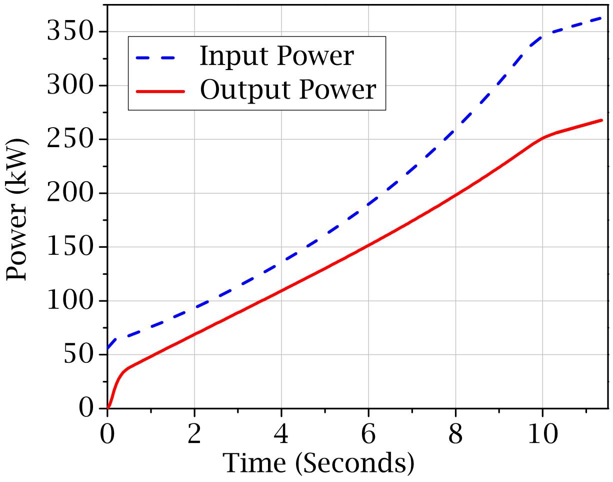


Figure 3.17: Input and output power

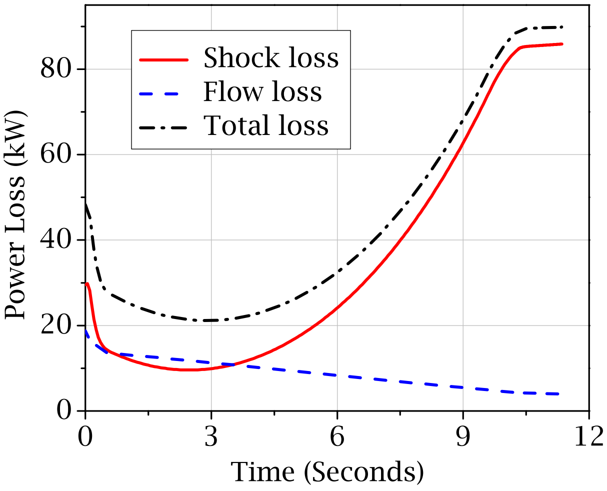


Figure 3.18: Shock and friction losses

The difference between the two curves, i.e., the difference between the input and output power, can be accounted for by adding the rate of change of energy stored in the torque converter elements and the losses incurred during the process.

Figure 3.18 shows the power lost due to shear stresses in the boundary layer (friction losses) and non-ideal flow conditions at the interfaces between the components (shock losses). The total power lost during the process is also plotted on the same axes. The plot identifies the shock losses as the predominant form of power loss for the system. Also, the plot clearly shows that the total lost power hits a minimum at around $t = 3$ seconds. As a result, the difference between the input power and the output power goes through a minimum at around the same point. Also, the efficiency stays high around this point, as shown in figure 3.16b, due to relatively low value for lost power.

By comparing figure 3.18 and figure 3.17, it can be explained why the efficiency decreases after going through a maximum. It is because, at that stage the losses increase, making the difference between the input and output power more pronounced.

Right before the coupling point and thereafter, the losses become more or less constant, as shown in figure 3.18. This makes the input and output power curves almost parallel to each other, vide figure 3.17. This results in a somewhat slow increase in the efficiency of the torque converter at this stage.

The simulation results convincingly establish the validity of the model of the torque converter. Since the topology of the model is captured by a linear graph, it is now possible to perform a graph-theoretic sensitivity analysis with this model. In the subsequent sections, the linear graph for this model will be used to generate the sensitivity equations of the system.

3.3.2 Graph-theoretic sensitivity analysis

As discussed in section 3.2, the linear graph model of the torque converter allows the application of graph-theoretic sensitivity analysis on the system. This section illustrates the application of graph-theoretic sensitivity analysis on the torque converter system. The relationship between the sensitivities of the through and across variables are derived using graph-theoretic modelling techniques [5, 63]. By assembling these equations, the sensitivity equations for the torque converter system are generated from the linear graph of the system shown in figure 3.11. The computer implementation of the process is also briefly described.

Sensitivity linear graph

A second graph G_2 is required to extract the sensitivity equations for the torque converter model. As explained in section 3.2, the second graph G_2 needs to be isomorphic with the linear graph shown in figure 3.11. The structure of the graph shown in figure 3.11 can be used to generate the sensitivity equations, since it not only captures the relationships between the individual through and across variables of the system, but also the relationships between their sensitivities.

At this point, proper symbols must be declared to represent the through and across variables of the sensitivity graph G_2. For the graph G_2, the mechanical domain variables are τ_{ib} and ω_{ib}, while the hydraulic domain variables are Q_{ib} and p_{ib}.

The subscripts of each of these variables have two parts. The first numerical part designates the identity of the associated edge. The second symbolic part represents the model parameter under study. The variables of graph G_2 can be derived from the variables of graph G_1 by differentiating them with respect to a scalar model parameter b.

$$\tau_{i\,b} = \frac{d}{db}\tau_i \tag{3.36}$$

Using these variables, the sensitivity cutset and circuit equations can be easily derived from the sensitivity linear graph G_2 as illustrated below.

Cutset, circuit and terminal equations for the sensitivity system

Sensitivity circuit equations (Hydraulic Domain):

$$\begin{gathered} \sum p_{i\,b} = 0 \\ i = 1,\ 2,\ 9,\ 10,\ 11,\ 12,\ 13,\ 20,\ 21,\ 22,\ 23,\ 24,\ 30,\ 31,\ 32 \end{gathered} \tag{3.37}$$

Sensitivity circuit equations (Mechanical Domain):

$$\begin{aligned} &\omega_{3\,b} = \omega_{4\,b} = \omega_{5\,b} = \omega_{6\,b} = \omega_{7\,b} = \omega_{8\,b} \\ &\omega_{14\,b} = \omega_{15\,b} = \omega_{16\,b} = \omega_{17\,b} = \omega_{18\,b} = \omega_{19\,b} \\ &\omega_{25\,b} = \omega_{26\,b} = \omega_{27\,b} = \omega_{28\,b} = \omega_{29\,b} \end{aligned} \tag{3.38}$$

Sensitivity cutset equations (Mechanical Domain):

$$\begin{aligned} &\tau_{3\,b} + \tau_{4\,b} + \tau_{5\,b} + \tau_{6\,b} + \tau_{7\,b} + \tau_{8\,b} = 0 \\ &\tau_{14\,b} + \tau_{15\,b} + \tau_{16\,b} + \tau_{17\,b} + \tau_{18\,b} + \tau_{19\,b} = 0 \\ &\tau_{25\,b} + \tau_{26\,b} + \tau_{27\,b} + \tau_{28\,b} + \tau_{29\,b} = 0 \end{aligned} \tag{3.39}$$

Sensitivity cutset equations (Hydraulic Domain):

$$Q_{32\,b} = Q_{[1,\,2,\,9,\,10,\,11,\,12,\,13,\,20,\,21,\,22,\,23,\,24,\,30,\,31]\,b} \tag{3.40}$$

The sensitivity terminal equations can be obtained by differentiating the system terminal equations. For the portion of G_2 that is isomorphic to the graphs shown in figure 3.10, the terminal equations are presented below. The two domains are considered separately and the sensitivity terminal equations are generated accordingly.

The terminal equations for the edge 28 are considered separately because of its unique nature. Since this edge represents the one way revolute joint the sensitivity terminal equation must be declared as a piecewise function which switches from one equation to the other depending on the torque on that edge or τ_{28}. The following equations can be derived from equation (3.15).

$$\omega_{28\,b} = \begin{cases} 0 & \tau_{28} > 0 \\ \omega_{s\,b}\,(t) & \tau_{28} < 0 \end{cases} \quad and \quad \tau_{28\,b} = \begin{cases} \tau_{s\,b}\,(t) & \tau_{28} > 0 \\ 0 & \tau_{28} < 0 \end{cases} \tag{3.41}$$

The following equations are used as the sensitivity terminal equations pertaining to the mechanical domain edges.

$$\begin{aligned} \tau_{3\,b} &= -\frac{d}{db}\,(\psi)\;Q_{\,1} - \psi\,Q_{\,1\,b} \\ \tau_{4\,b} &= \frac{d\rho}{db}S\dot{Q}_{\,2} + \rho\frac{dS}{db}\dot{Q}_{\,2} + \rho S\dot{Q}_{\,2\,b} \\ \tau_{5\,b} &= -\frac{d}{db}\,(J_m\,)\;\dot{\omega}_{\,5} - J_m\,\dot{\omega}_{\,5\,b} \\ \tau_{7\,b} &= -\frac{d}{db}\,(J_f\,)\;\dot{\omega}_{\,7} - J_f\,\dot{\omega}_{\,7\,b} \\ \tau_{6\,b} &= 0 \\ \tau_{8\,b} &= \frac{d}{db}\tau(t) \end{aligned} \tag{3.42}$$

The following equations are used as the sensitivity terminal equations pertaining to the hydraulic domain edges.

$$\begin{aligned} p_{\,1\,b} &= -\frac{d}{db}\,(\psi)\;\omega_{\,3} - \psi\,\omega_{\,3\,b} \\ p_{\,2\,b} &= \frac{d}{db}\,(\rho\,S)\;\dot{\omega}_{\,4} + \rho\,S\,\dot{\omega}_{\,4\,b} \\ p_{\,9\,b} &= \frac{\frac{d}{db}\,(P_{shock})\,Q_{\,9} - P_{shock}Q_{\,9\,b}}{\left(Q_{\,9}\right)^2} \\ p_{\,10\,b} &= \frac{\frac{d}{db}\,(P_{flow})\,Q_{\,10} - P_{flow}Q_{\,10\,b}}{\left(Q_{\,10}\right)^2} \\ p_{\,11\,b} &= \frac{d}{db}\left(\frac{\rho L^f}{A}\right)\dot{Q}_{\,11} + \frac{\rho L^f}{A}\dot{Q}_{\,11\,b} \end{aligned} \tag{3.43}$$

In the equations shown in (3.43), P_{shock} and P_{flow} refer to the expressions for the power lost due to shock and flow effects as illustrated in equations (B.28) and (B.26), respectively.

The terminal equations for the edges that correspond to the turbine and the stator can be similarly derived. For those equations, the structure of the terminal equations must remain the same but the coefficients like ψ, L^f etc. must be chosen properly.

Generation of the sensitivity equation

To generate the sensitivity equations the first step is to consider the linear graphs G_1, as shown in figure 3.11, and its identical twin G_2. Using a graph-theoretic formulation, both system and the corresponding sensitivity equations can be generated from these two graphs. For the purpose of convenience of the reader, the primary variable names are changed according to the table shown below.

Variable name	**Changed symbol**
$\omega_{5\,b}$	$\omega_{p\,b}(t)$
$\omega_{16\,b}$	$\omega_{t\,b}(t)$
$\omega_{27\,b}$	$\omega_{s\,b}(t)$
$\tau_{6\,b}$	$\tau_{p\,b}(t)$
$\tau_{17\,b}$	$\tau_{t\,b}(t)$
$\tau_{28\,b}$	$\tau_{s\,b}(t)$
$Q_{32\,b}$	$Q_{\,b}(t)$

Table 3.5: Sensitivity state variables for the torque converter model

Equation (3.44) shows the relationship between the changed variable names. Along with the convention shown in table 3.3, these variables are used to represent the sensitivity equations.

$$\omega_{p\,\rho} = \frac{d}{d\rho}\left(\omega_p\right) \tag{3.44}$$

The governing equations for this model can be expressed as a set of ordinary differential equations. Although the structure of the governing equation is already illustrated in equation (3.26), it is worthwhile to express it in a slightly different form to highlight the connection between the structure of the system equations and the sensitivity equations. The governing equation can be modified to the basic form of a set of ODEs which can be written in a matrix form as shown below.

$$\mathbf{M}\,\dot{\mathbf{q}} = \mathbf{F} \tag{3.45}$$

where the mass matrix $\mathbf{M}$ is given by equation (3.28), and the vector of generalized coordinates $\mathbf{q}$ is given by the equation (3.27). The vector $\mathbf{F}$ combines the applied torques on the system and the dynamic torques that are generated by the fluid coupling effect. From equation (3.26) the vector $\mathbf{F}$ can be constructed as shown below.

$$\mathbf{F} = \varsigma - \mathbf{G}\mathbf{q} - \xi \tag{3.46}$$

To demonstrate the process, the complete set of sensitivity equations are presented here by choosing the density of the hydraulic fluid ρ as the model parameter. The combined set of sensitivity and governing equations can be written in a matrix form as shown below.

$$\underbrace{\begin{bmatrix} \left(\mathbf{M}_b^{4\times4} + \sum\limits_{k=1}^{n} \frac{\partial \mathbf{M}_{ij}}{\partial \mathbf{q}_k} \mathbf{q}_{kb}\right) & \mathbf{M}^{4\times4} \\ \mathbf{M}^{4\times4} & 0^{4\times4} \end{bmatrix}}_{\mathbf{\Gamma}} \overbrace{\begin{Bmatrix} \dot{\mathbf{q}}^{4\times1} \\ \dot{\mathbf{q}}_b^{4\times1} \end{Bmatrix}}^{\dot{\boldsymbol{\vartheta}}} = \underbrace{\begin{Bmatrix} \mathbf{F}_b + \mathbf{F}_{\mathbf{q}}\mathbf{q}_b \\ \mathbf{F} \end{Bmatrix}^{8\times1}}_{\mathbf{\Omega}} \tag{3.47}$$

where $\mathbf{q}_k$ is the k th element of the vector $\mathbf{q}$, $\mathbf{M}_{ij}$ is a particular element of the mass matrix $\mathbf{M}$, and $\mathbf{q}_{kb}$ is the sensitivity of the k th element of the vector $\mathbf{q}$. The modified vector of state variables is given by $\boldsymbol{\vartheta}$ as shown below.

$$\boldsymbol{\vartheta} = \begin{Bmatrix} \omega_p & \omega_t & \omega_s & Q & \omega_{p\rho} & \omega_{t\rho} & \omega_{s\rho} & Q_\rho \end{Bmatrix}^T \tag{3.48}$$

The combined mass matrix is given by $\mathbf{\Gamma}$. The symbols $J_{p,t,s}$, $S_{p,t,s}$, and L^f are defined in equations (3.26) through (3.32).

$$\mathbf{\Gamma} = \begin{bmatrix} 0 & 0 & 0 & S_p & J_p & 0 & 0 & \rho S_p \\ 0 & 0 & 0 & S_t & 0 & J_t & 0 & \rho S_t \\ 0 & 0 & 0 & S_s & 0 & 0 & J_s & \rho S_s \\ S_p & S_t & S_s & \frac{L^f}{A} & \rho S_p & \rho S_t & \rho S_s & \frac{\rho L^f}{A} \\ J_p & 0 & 0 & \rho S_p & 0 & 0 & 0 & 0 \\ 0 & J_t & 0 & \rho S_t & 0 & 0 & 0 & 0 \\ 0 & 0 & J_s & \rho S_s & 0 & 0 & 0 & 0 \\ \rho S_p & \rho S_t & \rho S_s & \frac{\rho L^f}{A} & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.49}$$

The combined force vector $\mathbf{\Omega}$ is defined in equation (3.50).

$$\boldsymbol{\Omega} = \left\{ \begin{array}{c} -Q\sigma_1 - \rho\,\sigma_1\,Q_{\,\rho} - \rho\,Q\xi_1 + \tau_{p\,\rho} \\ -Q\sigma_2 - \rho\,\sigma_2\,Q_{\,\rho} - \rho\,Q\xi_2 + \tau_{t\,\rho} \\ -Q\sigma_3 - \rho\,\sigma_3\,Q_{\,\rho} - \rho\,Q\xi_3 + \tau_{s\,\rho} \\ \zeta_0 + \rho\,\zeta_1 + \frac{Q}{A}\zeta_2 + \rho\frac{Q_{\,\rho}}{A}\,\zeta_2 + \rho\frac{Q}{A}\zeta_3 - \frac{d}{d\rho}\left(P_L(t)\right) \\ -\rho\,Q\sigma_1 + \tau_p \\ -\rho\,Q\sigma_2 + \tau_t \\ -\rho\,Q\sigma_3 + \tau_s \\ \rho\,\zeta_0 + \rho\frac{Q}{A}\zeta_2 - P_L\left(t\right) \end{array} \right\} \tag{3.50}$$

where

$$\begin{aligned} \sigma_1 &= \left(\omega_p {R_p}^2 + \frac{Q}{A} R_p \tan\alpha_p - \omega_s {R_s}^2 - \frac{Q}{A} R_s \tan\alpha_s\right) \\ \sigma_2 &= \left(\omega_t {R_t}^2 + \frac{Q}{A} R_t \tan\alpha_t - \omega_p {R_p}^2 - \frac{Q}{A} R_p \tan\alpha_p\right) \\ \sigma_3 &= \left(\omega_s {R_s}^2 + \frac{Q}{A} R_s \tan\alpha_t - \omega_t {R_t}^2 - \frac{Q}{A} R_t \tan\alpha_t\right) \\ \xi_1 &= {R_p}^2 \omega_{p\,\rho} - {R_s}^2 \omega_{s\,\rho} + \frac{Q_{\,\rho}}{A} R_p \tan\alpha_p - \frac{Q_{\,\rho}}{A} R_s \tan\alpha_s \\ \xi_2 &= {R_t}^2 \omega_{t\,\rho} - {R_p}^2 \omega_{p\,\rho} + \frac{Q_{\,\rho}}{A} R_t \tan\alpha_t - \frac{Q_{\,\rho}}{A} R_p \tan\alpha_p \\ \xi_3 &= {R_s}^2 \omega_{s\,\rho} - {R_t}^2 \omega_{t\,\rho} + \frac{Q_{\,\rho}}{A} R_s \tan\alpha_s - \frac{Q_{\,\rho}}{A} R_t \tan\alpha_t \end{aligned} \tag{3.51}$$

$$\zeta_0 = \left({\omega_p}^2 {R_p}^2 + {\omega_t}^2 {R_t}^2 + {\omega_s}^2 {R_s}^2 - \omega_p\,\omega_t\,{R_p}^2 - \omega_p\,\omega_s\,{R_s}^2 - \omega_s\,\omega_t\,{R_t}^2\right) \tag{3.52}$$

$$\zeta_1 = \left(\begin{array}{l} 2{R_p}^2\,\omega_p\,\omega_{p\,\rho} + 2{R_t}^2\,\omega_t\,\omega_{t\,\rho} + 2{R_s}^2\,\omega_s\,\omega_{s\,\rho} \\ -{R_p}^2\,\omega_t\,\omega_{p\,\rho} - {R_p}^2\,\omega_p\,\omega_{t\,\rho} - {R_s}^2\,\omega_s\,\omega_{p\,\rho} \\ -{R_s}^2\,\omega_p\,\omega_{s\,\rho} - {R_t}^2\,\omega_t\,\omega_{s\,\rho} - {R_t}^2\,\omega_s\,\omega_{t\,\rho} \end{array} \right) \tag{3.53}$$

$$\begin{aligned} \zeta_2 &= \left(\begin{array}{c} \left(R_p \tan\alpha_p - R_s \tan\alpha_s\right)\omega_p + \left(R_t \tan\alpha_t - R_p \tan\alpha_p\right)\omega_t \\ + \left(R_s \tan\alpha_s - R_t \tan\alpha_t\right)\omega_s \end{array} \right) \\ \zeta_3 &= \left(\begin{array}{c} \left(R_p \tan\alpha_p - R_s \tan\alpha_s\right)\omega_{p\,\rho} + \left(R_t \tan\alpha_t - R_p \tan\alpha_p\right)\omega_{t\,\rho} \\ + \left(R_s \tan\alpha_s - R_t \tan\alpha_t\right)\omega_{s\,\rho} \end{array} \right) \end{aligned} \tag{3.54}$$

By solving this combined system of equations numerically, the state variables and the corresponding sensitivities can be evaluated as a function of time. The time varying sensitivity information obtained hereby can be used in many engineering applications. In a subsequent section some of the practical applications of sensitivity analysis will be presented.

3.3.3 Computer implementation

The model of the torque converter and the subsequent graph-theoretic sensitivity analysis thereof was implemented in *Maple 15*. Additionally, the graph-theoretic model of the hydrodynamic impeller was implemented in *MapleSim* and was used to assemble an acausal model of a torque converter. In this section, brief overview of these software implementations will be presented to outline the general methodology.

Maple implementation

Apart from the symbolic capabilities, as mentioned in chapter 2, Maple can also be used for numerical simulations. Advanced numerical routines allow users to solve complicated large systems of ODEs and DAEs using a variety of different algorithms. Because of its excellent symbolic and numeric capabilities, Maple can be used to simulate multibody systems effectively. Mathematically it becomes the process of solving a set of ODEs or DAEs depending on the nature of the system. For the implementation of the graph-theoretic model and the subsequent sensitivity analysis, Maple 15 was used for its excellent symbolic capabilities and its ability to solve ODEs involving piecewise functions effectively.

As a first step for graph-theoretic sensitivity analysis, the topology of the torque converter system, i.e., the linear graph shown in figure 3.11 is coded into Maple in terms of the incidence matrices for the two domains. Lists of through and across variables are also provided as inputs to the program. Using linear algebra, the fundamental cutset matrices and the fundamental circuit matrices are generated from the incidence matrices, which are then utilized to construct the cutset and circuit equations for both system and sensitivity equations pertain to the two domains present in the system.

The system terminal equations are provided to the program in the form of a list. The sensitivity terminal equations are then generated from the system terminal equations using symbolic differentiation. Using symbolic manipulation, these terminal equations and the generated cutset and circuit equations are combined to form the system and the sensitivity equations in a single unified process.

After the generation of the equations, the built in numeric ODE solvers are used to evaluate the state variables and the corresponding sensitivities as functions of time which are then used for different applications.

MapleSim implementation

MapleSim is a multi-domain modelling and simulation tool developed by *Maplesoft Inc.* It is capable of simulating electrical, electronic, mechanical, hydraulic and magnetic systems. The input to the software is the description of the system using the components from a central library that can be drag-dropped on to a worksheet. MapleSim is built on the Maple software, which enables it to perform very effective symbolic simplifications of the generated equations. By using `dsolve[numeric]`, the built-in ODE/DAE solver routine in Maple, MapleSim can simulate and present the output of the models in an interactive three-dimensional environment. It can also perform post-processing of the generated data using predefined templates.

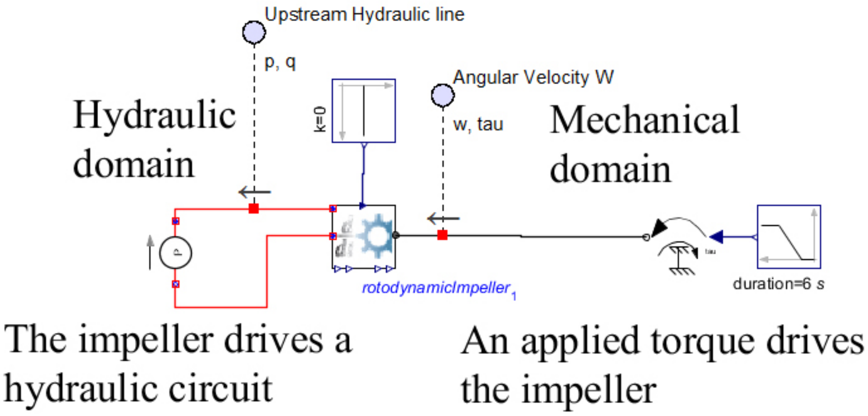


Figure 3.19: MapleSim model of a rotodynamic impeller

The multibody package of MapleSim uses a linear graph based formulation. Other components (e.g., electrical, magnetic, thermal, hydraulic etc.) are based on **Modelica**$^{\text{TM}}$ codes. Apart from the built-in library of standard components, MapleSim allows users to create custom components for user specific implementation. These custom components are based on the Maple language and can be readily included in models created using MapleSim's standard components.

MapleSim was chosen for the implementation of the torque converter model due to its ability to accept user defined modules. The key point for the implementation of a physics based model of a torque converter is the fact that all three components of the torque converter, i.e., the pump, the turbine, and the stator, are essentially the same mechanical component. Regardless of their appearances and functions, all of these components act as transducers between hydraulic and mechanical domains and either produce a torque when a flow is established with a pressure difference across their vanes or drive a flow rate against a pressure gradient when there is a torque applied to them. These components are known as rotodynamic impellers, which can be modelled easily in MapleSim.

The schematic model in figure 3.19 shows the implementation of one such impeller model in MapleSim. The component at the centre named "*rotodynamicImpeller$_1$*" is an user defined module which holds the governing equations for the impeller. These equations are derived from the partial linear graph shown in figure 3.10. The components to the right of this module belong to the mechanical domain. In this example, a torque driver is used to deliver a ramp torque as an input to the system. On the other side, the components to the left of the diagram belong to the hydraulic domain. In this example, there is a single pressure driver which sets up a pressure gradient against which the impeller is forced to work.

Mathematically speaking, the only differences between the elements of a torque converter is the values of their physical parameters. Thus by changing the model parameter values and repeating the same model of a rotodynamic impeller, it is possible to construct a complete torque converter model in MapleSim.

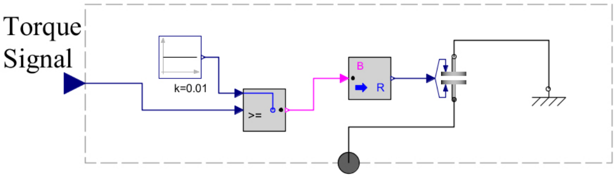


Figure 3.20: Model of a one way revolute flange

Another important component of a torque converter is the one-way joint that allows the stator to rotate in one direction only. This serves a very important function for the torque converter and it must be modelled properly for accurate simulation. In this example, the one way clutch was implemented using built-in components from MapleSim.

Figure 3.20 shows the subsystem that functions as a one-way clutch. The input rotating flange, which is connected to that of the stator, is connected to the ground through a friction clutch. The input signal, which carries the transmitted torque information, controls the activation of the clutch. When sufficiently high clutch constants are used, this assembly ensures that the stator rotates freely in one direction and locks up when pushed in the other direction.

Figure 3.21 shows the model of the torque converter implemented in MapleSim. Since it is an acausal model, it contains connections in both hydraulic and mechanical domains. The different components, i.e., the pump, the turbine, and the stator are shown as blocks. They differ in the values of their physical parameters but the underlying model is the same for all three of them. These rotating components are connected to the ground using revolute joints and in case of the stator element, through a one way clutch as illustrated in the figure.

The hydraulic losses are included as flow-dependent pressure drivers inserted in the hydraulic connections which form a closed loop to represent the path taken by the hydraulic fluid. Since the losses are also dependent on the angular velocity of the rotating elements, this information are fed into the relevant blocks using signal connections.

Both the pump and the turbine are connected to torque drivers. For the pump element, the torque driver is programmed to deliver a ramp torque to simulate torque supply from the engine, whereas for the turbine, the torque driver is programmed to deliver a constant negative torque to simulate a constant resistive load.

Both of these implementations are used to simulate and analyze the behaviour of a torque converter. The Maple 15 implementation in particular is used to gather sensitivity information and perform design optimization. In the next section, one such example will be presented to showcase the application of sensitivity analysis.

3.3.4 Importance analysis and design optimization

The model parameters, i.e., the dimensions and the geometric properties of the components and the physical and hydrodynamic properties of the hydraulic fluid, play a major role in determining the dynamics of a torque converter. In many cases, it is important to understand which of these parameters have more influence on the dynamics than others.

Importance analysis

When it is necessary to determine which of the parameters are most important for the model, using sensitivity analysis it is possible to develop a relative order of importance among the parameters. To perform importance analysis on a system, a suitable criterion must be specified to provide a basis for the comparison. The results of importance studies are very much dependent on the chosen criteria or what are known as the objective functions. A

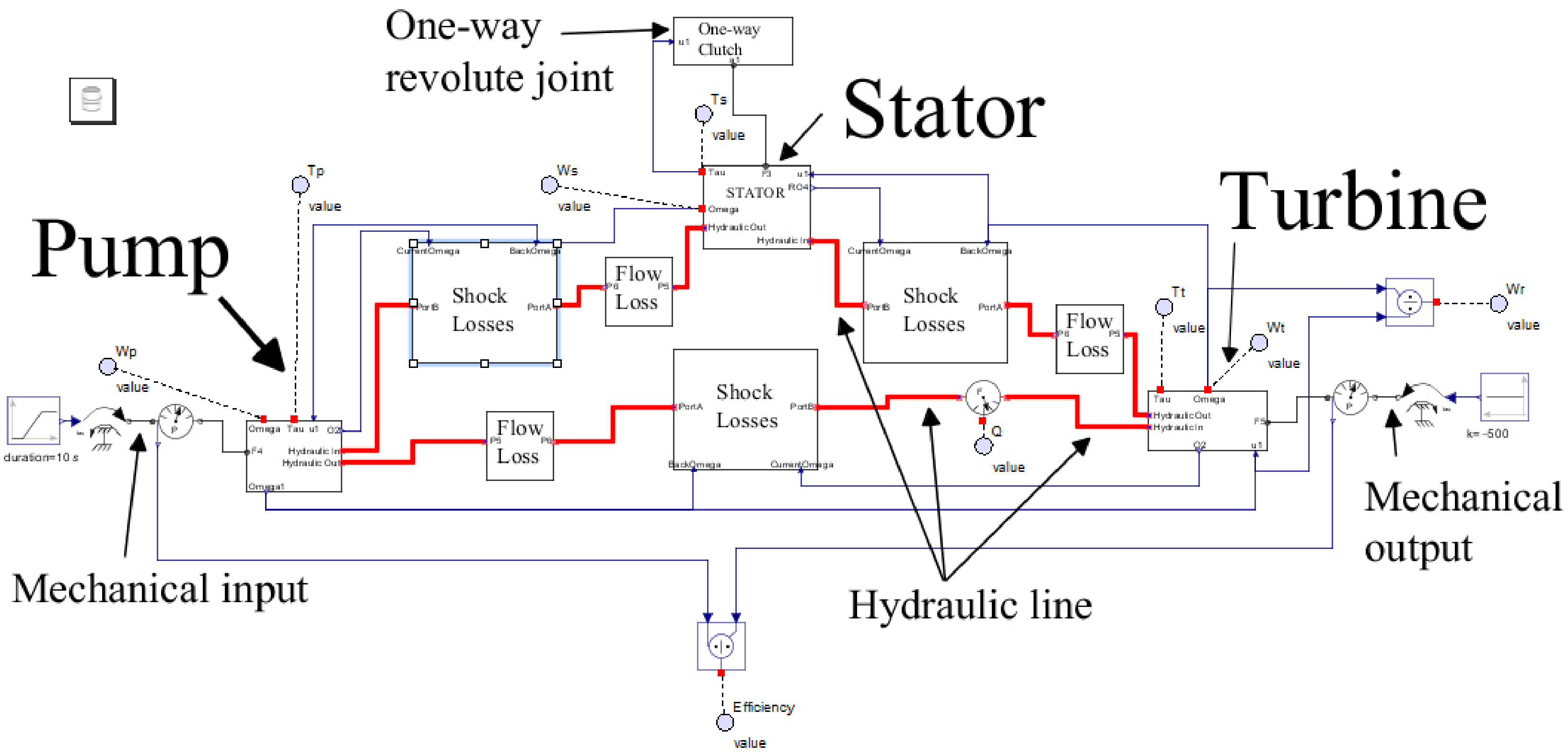


Figure 3.21: MapleSim implementation of the model of a hydrodynamic torque converter

general objective function can be expressed in terms of the set of model parameters **p** and the state variables **q** and can be written in the general form:

$$\lambda = \int_{T_1}^{T_2} f\left(\mathbf{q}, \mathbf{p}, t\right) dt \tag{3.55}$$

By differentiating equation (3.55) with respect to a single model parameter b, the following expression is obtained.

$$\lambda_b = \frac{\partial \lambda}{\partial b} = f\left(\mathbf{p}, T_2\right) \frac{\partial T_2}{\partial b} - f\left(\mathbf{p}, T_1\right) \frac{\partial T_1}{\partial b} + \int_{T_1}^{T_2} \left(\frac{\partial f}{\partial b} + \frac{\partial f}{\partial \mathbf{q}} \mathbf{q}_b \right) dt \tag{3.56}$$

In the above equation, λ_b is the sensitivity of the objective function with respect to an arbitrary model parameter b. It is also known as the absolute sensitivity of the objective function f with respect to the parameter. The quantity $\mathbf{q}_b$ is the sensitivity of the state variables with respect to the parameter.

The first two terms of equation (3.56) are required only if the limits of integration are dependent on the parameter in question. Since for most of the time, the limits are independent of the model parameters, these terms don't figure in the expressions. However, for certain situations these assumptions might not be practical and due consideration must be given to these terms to ensure accurate computation of the sensitivity information.

The first term inside the integral sign of equation (3.56) can be evaluated symbolically from the structure of the function f. However, the second term requires sensitivity information that can be obtained by solving the sensitivity equations presented in previous sections.

Depending on the effect that a particular parameter has on the objective function, the value of the absolute sensitivity would be different. This piece of information can be used to arrange the parameters in their order of importance based on the values of the corresponding sensitivity evaluations.

For this study, the total loss of energy is chosen as the objective function for the importance analysis. The energy lost during the process can be written in terms of the input torque τ_p, the angular velocity ω_p, and the efficiency of the torque converter η as

$$E_{loss} = \int_{0}^{T_{lockup}} Power\left(t\right) dt = \int_{0}^{T_{lockup}} \left(\tau_p\left(t\right) \omega_p\left(t\right) \left(1 - \eta\left(t\right)\right)\right) dt \tag{3.57}$$

The upper limit of the integration T_{lockup} is the time when the lock-up clutch joins the two sides of the torque converter into a single piece and eliminates the dynamic effects of the torque converter. Using (3.56) one can obtain the expression of the sensitivity of the quantity with respect to an arbitrary model parameter as

$$\begin{gathered} E_{loss_b} = \frac{\partial E_{loss}}{\partial b} = Power(T_{lockup}) \times \frac{\partial T_{lockup}}{\partial b} \\ + \int\limits_{0}^{T_{lockup}} \left(\frac{\partial \tau_p}{\partial b} \omega_p (1-\eta) + \tau_p \frac{\partial \omega_p}{\partial b} (1-\eta) - \tau_p \omega_p \frac{\partial \eta}{\partial b} \right) dt \end{gathered} \tag{3.58}$$

The first term of equation (3.58) is particularly troublesome due to the fact that, although the time of lock-up T_{lockup} is dependent on the model parameters, it is not easily expressed as an explicit function of any of them. It is dependent on the action of the lock-up clutch which activates when a particular angular speed is achieved. Thus, T_{lockup} can only be determined from the numerical simulations. Consequently, the only way to calculate the sensitivity of T_{lockup} is to approximate the derivative with a finite difference approach and simulate the model repeatedly with perturbed values of the parameters, something that may undermine the benefits of the direct differentiation approach.

However, numerical experiments and comparison has shown that the magnitude of this term is small compared to that of the second term and can be safely ignored for the purpose of analysis. Results confirming this statement are presented in a subsequent section.

Normalized sensitivity score

The absolute sensitivity of the objective function with respect to an arbitrary model parameter is a measure of the change in the objective function for a unit change in the parameter value, assuming a first order approximation. However this piece of information, in this form, is not very suitable to determine which parameter has more effect on the objective function. Since different parameters have different units, the significance of the perturbation of one unit is completely different for two separate parameters. To overcome this problem we normalize the absolute sensitivity by multiplying it with the nominal value of the parameter and dividing it by the nominal value of the objective function. This gives us what is known as the relative sensitivity of the objective function and is defined as [68]

$$\mu = \frac{\partial E_{loss}}{\partial b} \left| \frac{b^*}{E_{loss}(b^*)} \right| \tag{3.59}$$

In equation (3.59) the symbol b^* denotes the nominal value of the parameter. The quantity thus obtained is dimensionless and suitable for the purpose of comparison.

At this point, it is worthwhile to explain that the relative sensitivity defined in equation (3.59), represents the fractional *increase* in the objective function for a fractional *increase* in the model parameter. More explanation on this definition is provided in Appendix B.5.

Results

The sensitivity of the objective function as outlined in (3.58) poses a question of whether or not the derivative of the lock-up time with respect to the model parameter is significant compared to the other terms in the expression. In a mathematical form it translates to the question of whether or not

$$\chi = \int_{0}^{T_{lockup}} \left(\frac{\partial \tau_p}{\partial b}\, \omega_p \left(1-\eta\right) + \tau_p \frac{\partial \omega_p}{\partial b} \left(1-\eta\right) - \tau_p\, \omega_p \frac{\partial \eta}{\partial b} \right) dt \tag{3.60}$$

is a good approximation for the sensitivity of the objective function E_{loss_b} term, as shown in equation (3.58).

One possible way to resolve this issue is to compare the values obtained for equation 3.60 and the evaluation of the quantity E_{loss_b} using a finite difference formulation, which would require simulation of the system with appropriate perturbed values for the parameters.

To determine the correct perturbation factor for this system, a convergence test can be performed where a parameter is perturbed by a series of small factors δ and the resulting perturbed values i.e., $\tilde{b} = b^* \times (1 \pm \delta)$ are used to simulate the system and calculate the sensitivity of the objective function using a central difference formulation.

The results are expected to converge for an appropriate value of δ. Furthermore, if χ is a good approximation of the quantity, it is expected that the finite difference results would agree with the results obtained by direct differentiation.

These differences between the results from the direct differentiation application and that from finite difference formulation are also dependent on the values of the parameters. To study the effect of parameter values on the convergence of the finite difference formulation, comparisons between the direct differentiation and finite difference are made for different values of the model parameters. For an acceptable scenario, it is expected that the error bound, or the difference between the finite difference result and direct differentiation result, for the entire range of values of the model parameters, should be reasonable.

Figure 3.22 shows the result of one such study. The perturbation factor was varied from 10^{-2} to 10^{-8} and sensitivities were calculated using central difference formulation. The plot shows that the solution converges for perturbation factors between 10^{-3} and 10^{-6}. Also, it is to be noted that the convergence occurs close to the value calculated using direct differentiation, which is indicative of the accuracy of the approximation.

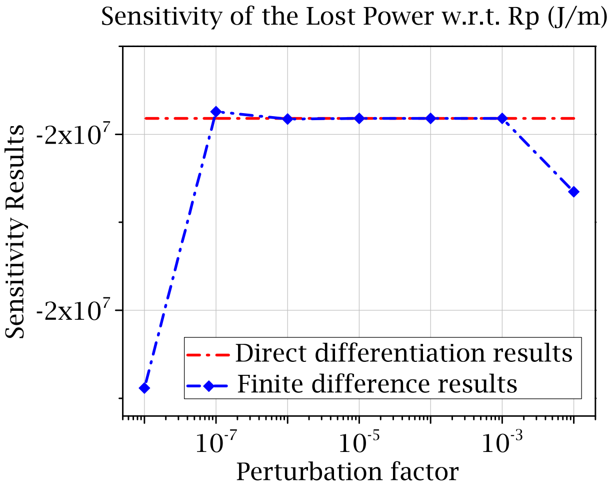


Figure 3.22: Convergence study to determine optimal perturbation factor

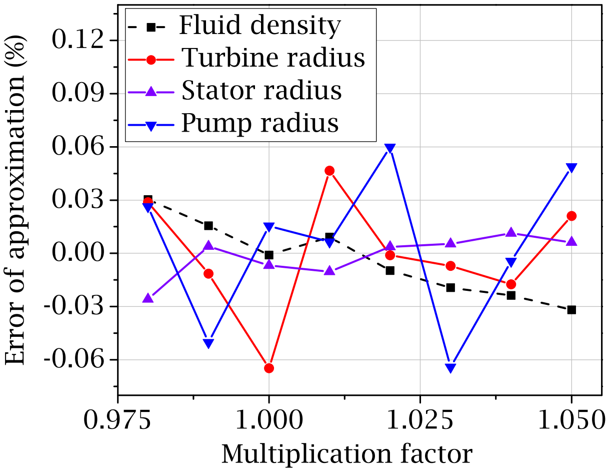


Figure 3.23: Error in approximation for different values of parameters

Figure 3.23 shows the error of approximation for different values of model parameter. In this plot, the percent error of approximation is plotted along the vertical axis and the normalized range of the parameter values are plotted on the horizontal axis. This way the horizontal axis represents the factor, which the nominal value of a particular parameter is multiplied with, to arrive at the new value of the parameter. The plot shows the trend for four different parameters. It can be clearly seen that the approximation error remains within reasonable levels for the chosen range of values of the model parameters.

From this discussion, it is quite clear that for the relevant ranges of parameter values, the quantity χ is indeed a good approximation of the sensitivity of the objective function E_{loss_b}. Therefore, one can safely neglect the leading term of equation (3.58), and use equation (3.60) to evaluate the sensitivity of the objective function, with respect to any arbitrary parameter, at least for the simulation considered.

Using equations (3.60) and (3.59) the normalized importance scores are generated for all the relevant model parameters. Figure 3.24 shows the list of model parameters in decreasing order of importance. It is to be noted that although the plot shows both positive and negative values, only the absolute values of the normalized sensitivity scores are used while arranging the parameters in this order.

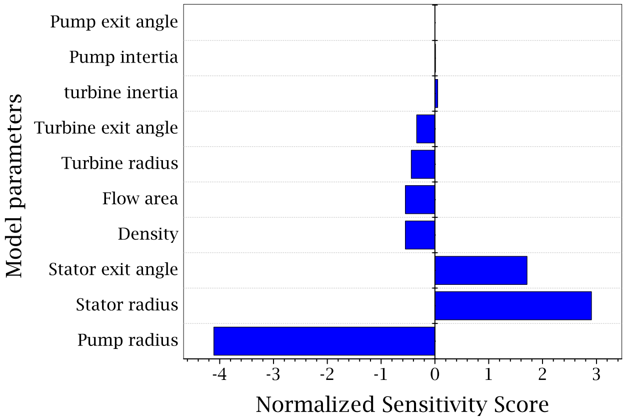


Figure 3.24: Ordered importance scores for the model parameters

Figure 3.24 clearly identifies the radii of the pump, stator and turbine, the density of the hydraulic fluid, and the stator exit angle to be the most important parameters of the model from the standpoint of the chosen objective function, i.e., total energy loss during the process.

Parameters	Absolute Sensitivity	Normalized Sensitivity score
R_p	-1.99×10^7	-4.11
R_s	1.85×10^7	2.91
α_s	1.47×10^4	1.71
ρ	-3.62×10^2	-0.55
A	-3.03×10^7	-0.55
R_t	-3.63×10^6	-0.44
α_t	-3.25×10^3	-0.34
I_t	9.11×10^4	0.05
I_p	8.68×10^4	0.01
α_p	-1.08×10^3	0

Table 3.6: Importance scores for the torque converter parameters

From this list, it can be inferred that for a design optimization problem, these parameters should be given priority, because they would have the greatest influence on the objective function. A complete list of the numerical values for the absolute and normalized importance scores for the model parameters is provided in table 3.6.

At this point, it should be clarified that, while the magnitudes of the sensitivity information tell us the amount of influence a particular parameter has on the objective function, its sign tells us the direction in which this influence acts. This piece of information is critical for design optimization problems, where the direction of steepest descent is required to arrive at the optimal point. Also it needs to be stated that this result reflects the behaviour of the system in a strictly local sense and is therefore restricted to be valid for small perturbations only.

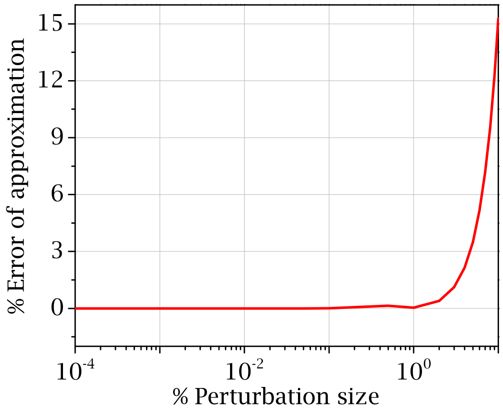


Figure 3.25: Accuracy of the sensitivity information

To validate the sensitivity information, obtained using the analysis shown above, and to illustrate the effect of the perturbation size on the accuracy of computation, the amount of lost energy for a perturbed system can be compared with that predicted by an approximation, based on first order sensitivity analysis of the original system. The approximation can be written as shown in equation (3.61),where ${}^{*}E_{loss}$ is the predicted loss of energy for the perturbed system, E_{loss} and E_{loss_b} correspond to the original system and δb denotes the perturbation of the parameter.

$$ {}^{*}E_{loss} = E_{loss} + E_{loss_b}\delta b \tag{3.61} $$

The difference between the quantities ${}^{*}E_{loss}$ and the actual loss calculated by simulating the perturbed system can be used to assess the accuracy of the sensitivity analysis for different values of perturbation sizes.

Figure 3.25 shows the difference between the predicted and actual energy losses for different perturbation sizes of the model parameter R_p or the pump radius. In the figure, the percentage perturbation of the value of the pump radius is plotted on the horizontal axis using a log scale. The percentage difference between the values of lost energy, calculated using equation (3.61) for the original system and equation (3.57) for the perturbed system, are plotted on the vertical axis. The plot clearly shows the dependence of the accuracy of sensitivity analysis on the perturbation sizes and it validates the sensitivity results for reasonably small perturbation sizes.

Design Optimization

Using the sensitivity information, a design optimization problem can be set up to tune the model parameters of the torque converter for a better design. To improve the design of the torque converter, the main criterion is to minimize the energy loss during the operation. Mathematically it can be achieved by minimizing the cost function shown below.

$$\lambda_1 = \frac{1}{\tau_p(0)\,\omega_p(0)} \left(\int_0^{T_{lockup}} \tau_p(t)\,\omega_p(t)\,(1-\eta(t))\,dt \right) \tag{3.62}$$

However, it is also a general observation that when efficiency is increased, the torque capacity of a torque converter usually goes down. As a result, a more efficient torque converter often becomes severely restricted in terms of usability. A cost function can be considered to include the effect of capacity factor as shown below.

$$\lambda_2 = \left(\frac{\sqrt{\tau_p(0)}}{\omega_p(0)} \right) \int_0^{T_{lockup}} K_c(t)\,dt \tag{3.63}$$

In this scenario, the use of a single criterion cost function is highly inadvisable since it can result in a practically useless configuration. Thus a proper optimization philosophy should focus on decreasing the area under the power lost curve and increasing the area under the torque capacity curve, at the same time. In other words, the cost function should combine the effects of both equations (3.62) and (3.63). In this example a single objective function is constructed by a linear combination of the two cost functions mentioned before.

$$\text{minimize } \lambda = a_1\,\lambda_1 - a_2\,\lambda_2 \tag{3.64}$$

In equation (3.64), λ_1 and λ_2 represent the two separate criteria that are being combined into a single objective function. The constants a_1 and a_2 are the weight coefficients that determine the relative weight or importance given to the individual criterion. It is to be noted that the outcome of the optimization is expected to be strongly dependent on the values of these weights.

$$\begin{aligned}
\frac{\partial\lambda}{\partial b} &\approx \frac{a_1}{T_p\,(0)\,\omega_p\,(0)} \int\limits_0^{T_{lockup}} \Gamma\; dt - a_2\, K_c\,(0) \int\limits_0^{T_{lockup}} \frac{\partial K_c}{\partial\, b}\; dt \\
\Gamma &= \left(\frac{\partial\,\tau_p}{\partial b}\,\omega_p\,(1-\eta) + \tau_p \frac{\partial\,\omega_p}{\partial b}\,(1-\eta) - \tau_p\,\omega_p \frac{\partial\,\eta}{\partial b} \right)
\end{aligned} \tag{3.65}$$

By using direct differentiation and assuming minimal dependence of the term T_{lockup} on the model parameters, the expression for the sensitivity of the objective function can be obtained as shown in equation (3.65). This expression is to be used to determine the gradient of the objective function during the optimization iterations.

In this example, only the parameters with greater importance are considered for optimization and the others are assumed to be constants. From the results of importance analysis presented earlier, four parameters are chosen for this purpose and a constraint declaration is constructed as shown below. It is to be noted, that the ranges specified here was taken from device specification sheets and do not correspond to a particular design optimization problem.

$$\begin{aligned}
&0.114\,\mathrm{m} \le R_p \le 0.13\,\mathrm{m} &\quad& 0.08\,\mathrm{m} \le R_s \le 0.09\,\mathrm{m} \\
&0.065\,\mathrm{m} \le R_t \le 0.15\,\mathrm{m} && 830\,Kg/m^3 \le \rho \le 880\,Kg/m^3
\end{aligned} \tag{3.66}$$

For successful optimization, apart from making the iterations follow the constraints imposed on the values of the model parameters, it is also required to decide on the relative weights or importances to be given to the individual objective functions λ_1 and λ_2. In other words, before the optimization can be carried out, one must choose appropriate values for the constants a_1 and a_2.

The expressions shown in (3.62) and (3.63) are evaluated at the extreme values of the parameters (equation (3.66)) to obtain the ranges of variation of λ_1 and λ_2. The quantities a_1 and a_2 are subsequently calculated to scale these ranges to be equal to each other. To ensure proper scaling, a_1 and a_2 are also constrained to satisfy the relationship $a_1 + a_2 = 1$, which leads to the following values that are used for this analysis.

$$a_{\,1} = 0.47 \quad \text{and} \quad a_{\,2} = 0.53 \tag{3.67}$$

Assuming that the lock-up mechanism is activated when stator speed reaches 20 rad/s, the optimization was carried out for this system, using gradient-based steepest decent method to adjust the parameters. The parameter values were restricted to be inside a particular range, shown in equation (3.66). Equation (3.64) was used to calculate the value of the objective function and equation (3.65) is used to calculate the gradient of the cost function. For this optimization problem, the simulation is continued past the coupling point until the stator speed reached $\omega_s = 20\, rad/s$. Equation (3.68) shows the optimal parameter values which results in $E_{loss} = 391.3\,\mathrm{kJ}$.

$$\begin{bmatrix} R_t = 0.065\,\mathrm{m} & R_p = 0.12\,\mathrm{m} & R_s = 0.08\,\mathrm{m} & \rho = 830\,\mathrm{Kg}/m^3 \end{bmatrix} \tag{3.68}$$

At this point, it is worthwhile to take note of the initial values of the parameters of the system before the optimization was performed and the corresponding amount of E_{loss}.

$$\begin{bmatrix} R_t = 0.068\,\mathrm{m} & R_p = 0.115\,\mathrm{m} & R_s = 0.088\,\mathrm{m} & \rho = 840\,\mathrm{Kg}/m^3 \end{bmatrix} : E_{loss} = 557.8\,\mathrm{kJ} \tag{3.69}$$

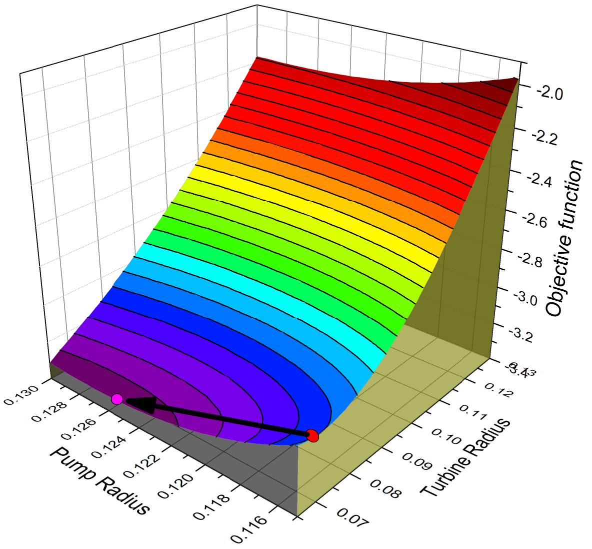


Figure 3.26: Variation of λ with R_p and R_t

The process of optimization can be partially visualized by plotting the variation of the objective function λ for various values of the model parameters. A surface plot that shows the values of the objective function for different values of R_p and R_t is shown in figure 3.26.

Figure 3.26 shows the trend in the optimization process. Due to the dependence of the objective function on the other model parameters, the optimal point shown in the figure above does not coincide with the optimal point illustrated by equation (3.68). However the direction of change for the model parameters are clearly evident in the plot.

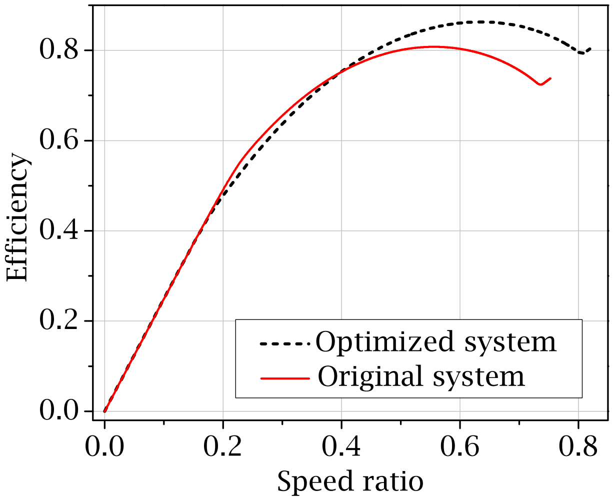


Figure 3.27: Efficiency vs. speed ratio for the torque converter

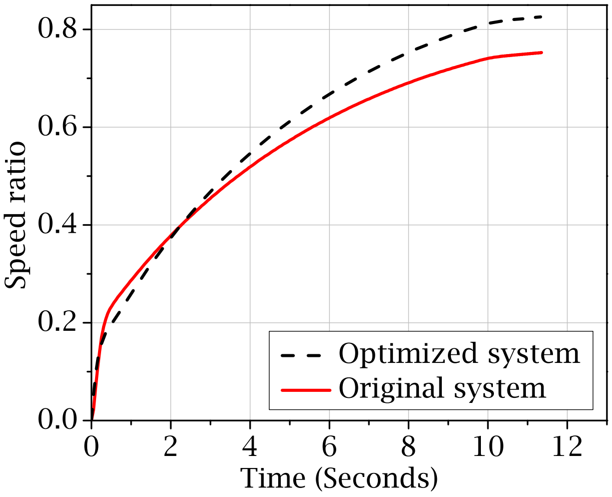


Figure 3.28: Speed ratio vs. time for the torque converter

The effects of the optimized model parameters are clearly evident in the behaviour of the system.

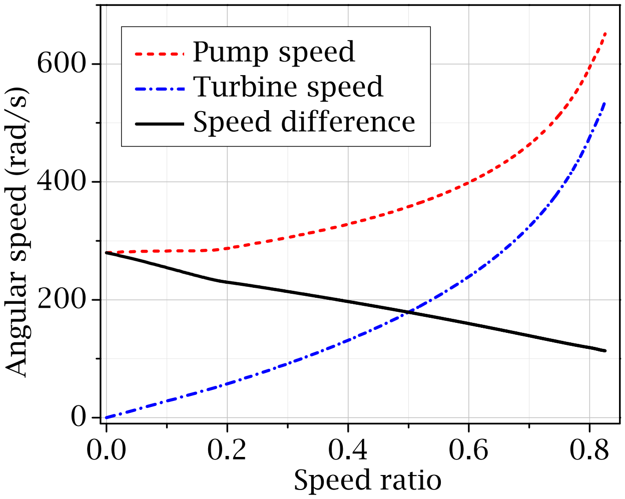


Figure 3.29: Angular speed of the components for the optimized system

Figure 3.27 plots the efficiency of the system as a function of the speed ratio for the optimized and the original system. It shows a clear increase in the efficiency of operation right before the coupling point, an area where the efficiency usually drops down. Also the efficiency at the coupling point is found be improved.

Apart from the improvement in the efficiency of the operation, the optimized values of model parameters also increase the speed ratio at which the lock up clutch is activated, vide figure 3.28.

Figure 3.29 shows a plot of the angular speeds of the rotating parts of the optimized system against time. The difference between the two speeds is also plotted on the same axes. It clearly demonstrates that in the optimized system, the slip speed at the point of coupling is around 125 rad/s, which is lower than that shown in figure 3.14. This directly results in an improved life for the lock-up clutch.

To illustrate the modality by which the optimized parameters minimize loss of energy, it is important to study different types of losses and the effect of the optimization on them. Shock losses and friction losses are the two types of losses included in this model. Shock losses are due to non-ideal flow velocity conditions inside the torque converter and the friction losses are due to fluid friction and pressure drag effects.

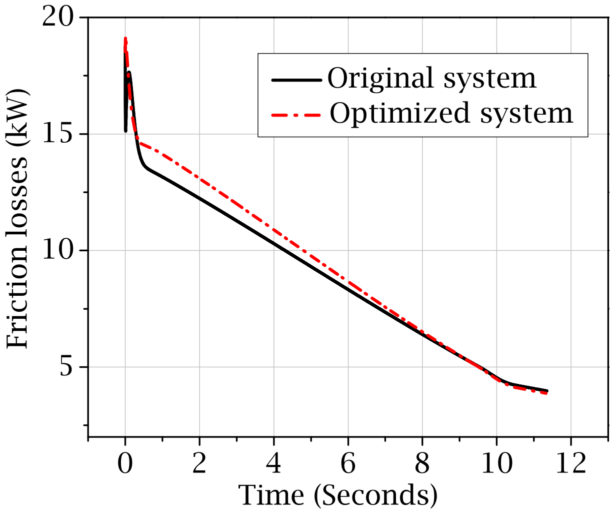


Figure 3.30: Friction loss vs. time

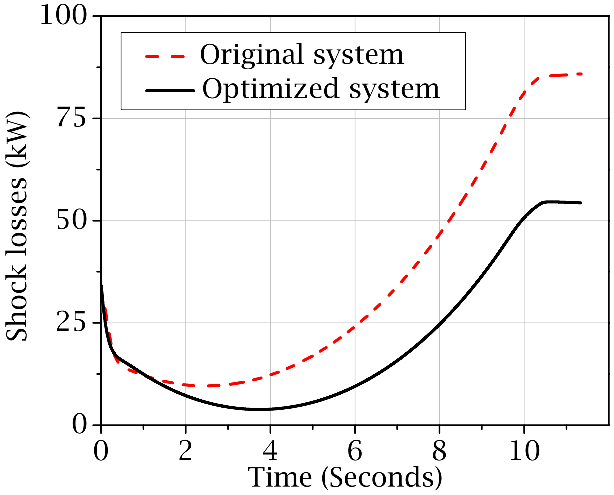


Figure 3.31: Shock loss vs. time

Figure 3.30 shows the variation of the friction losses with time for the optimized and original system. The plots once again identify the friction losses as the less dominant of loss terms and indicates that in the optimal system, the friction loss is actually a little higher than that in the original system.

On the other hand, figure 3.31 plots the shock loss terms. In this figure, the variation of the shock losses for the optimized and the original systems are plotted on the vertical axis and time in seconds is plotted on the horizontal axis. These plots clearly identify the shock losses as the dominant form of losses and also indicate that the shock loss term in the optimized system is lower than that in the original system.

The combined effect of the improved efficiency and decreased friction and shock loss terms can be clearly seen in figure 3.32. In this figure the lost power is plotted on the vertical axis and time is plotted on the horizontal axis. The plot clearly shows that in the optimized system, the power loss is lower than that in case of the original system, and the system reaches the lock-up state in a slightly less amount of time.

From these plots it can be stated that compared to the original system, the optimized system shows a lower shock loss term but a higher friction loss term. However, due to the fact that the shock losses are the predominant loss term and in the optimized system the shock losses are less than that in the original system, the total power loss in the optimized system is considerably lower than that in the original system, as demonstrated by figure 3.32.

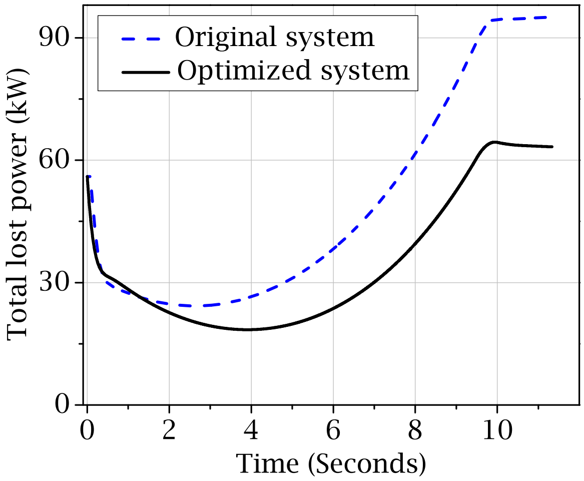


Figure 3.32: Total power loss vs. time for the torque converter

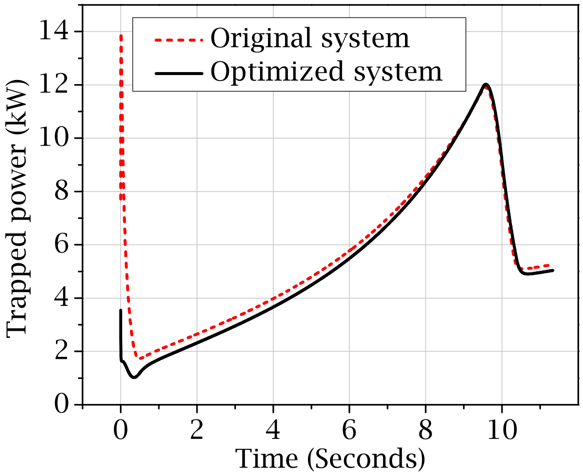


Figure 3.33: Trapped power vs. time for the torque converter

To explain why the optimized system has lower shock loss term, one needs to consider the angular velocity of the pump and the turbine, as shown in figure 3.29. The plot clearly demonstrates that, in the optimized system, the final angular velocity of the pump and the turbine, right before the activation of the lock-up clutch, is lower than that in case of the original system. Since the shock losses are dependent on the angular speeds of the components of the torque converter, lower values of ω_p and ω_t result in lower shock losses for the system.

The optimized system requires less time to reach the locked up state, which results in higher volumetric flow rate of the hydraulic fluid. Since the friction losses are dependent on the flow rate of the fluid, a higher flow rate results in higher friction losses.

Apart from the losses due to fluid friction and shock effects, a small portion of input power is left trapped inside the torque converter as the power associated with the whirling hydraulic fluid and later with the rotating stator. Figure 3.33 shows the variation of the trapped power with time. It also demonstrates that the optimized system lowers the amount of trapped power, which results in an overall increase in the available power from the torque converter.

Effect on the capacity factor

As mentioned in a previous section, the variation of the capacity factor of the torque converter limits its usability in many situations. It is therefore worthwhile to investigate the behavior of the capacity factor for the optimized system. Figure 3.34 shows the variation of the capacity factor, for both original and the optimized system. It clearly shows that the range of variations of the capacity factor for the original and the optimized torque converter are comparable to each other, although it is a little tighter for the optimized system.

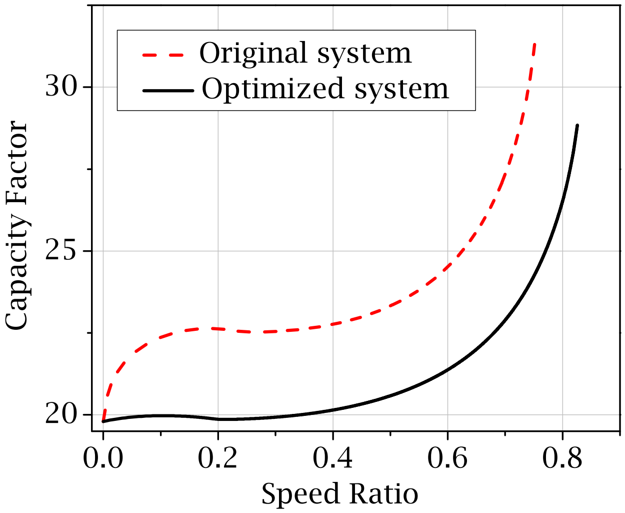


Figure 3.34: Capacity factor for the original and the optimized system

In the context of the above mentioned discussion, the merit of including the capacity factor into the objective function can be demonstrated by solving the optimization system with a different set of weights. To determine the optimal point, when the sole objective is to reduce the loss of energy, the optimization problem is solved using $a_1 = 1$ and $a_2 = 0$. This results in the following values of the model parameters as the optimal set.

$$\left[\ R_t = 0.088\,\mathrm{m} \quad R_p = 0.13\,\mathrm{m} \quad R_s = 0.08\,\mathrm{m} \quad \rho = 880\,\mathrm{Kg}/m^3\ \right] \tag{3.70}$$

When the values shown in equation (3.70) are used for the simulation of the torque converter, the total energy loss is determined to be lower than that found in the previous case. The quantity E_{loss} is determined to be

$$E_{loss} = 259.6\ \mathrm{kJ} \tag{3.71}$$

The simulation results of the torque converter using the values shown in equation (3.70) are shown in the figures below. The first plot, figure 3.35, reveals that the slip speed at the point of coupling is around $70\,rad/s$ which is even lower than that shown in figure 3.29.

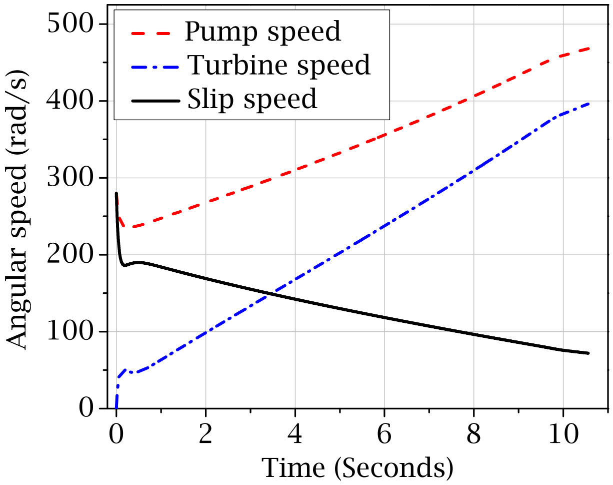


Figure 3.35: ω_p, ω_t, and slip speed (using values shown in (3.70))

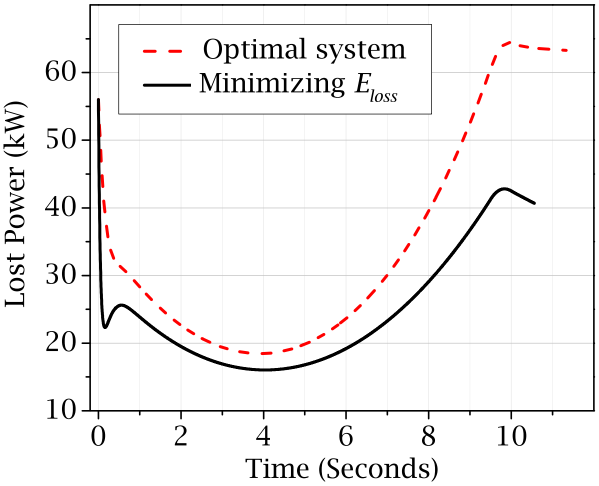


Figure 3.36: Power loss vs. time (using values shown in (3.70))

Figure 3.36 compares the lost power of the system that uses parameter values from equation (3.70) with the lost power of the system that uses parameter values from equation (3.68). It clearly shows that when the minimization of lost energy criterion is used as the sole objective function, the evaluated optimal point results in considerable reduction in E_{loss}.

Although all these results suggest that the parameter values presented in (3.70) results in a system which performs better than the optimal system presented in the previous sections, it is important to consider the usability of the system before accepting it as a superior design.

To determine the usability of the torque converter, one can consider the capacity factor which is a rough estimate of the amount of power that can be transmitted through the device. Figure 3.37 shows the variation of the capacity factor for the original system, i.e., system that uses parameters listed in table A.1, the optimized system that uses parameter values shown in (3.68), and also the system that uses parameter values shown in equation (3.70).

Figure 3.37 clearly demonstrates that the system which minimizes the lost power also has the minimum range of variation for the capacity factor. In other words, although the system defined by (3.70) manages to minimize the amount of power that is lost in the process, it is practically useless due to the limited range of its capacity factor.

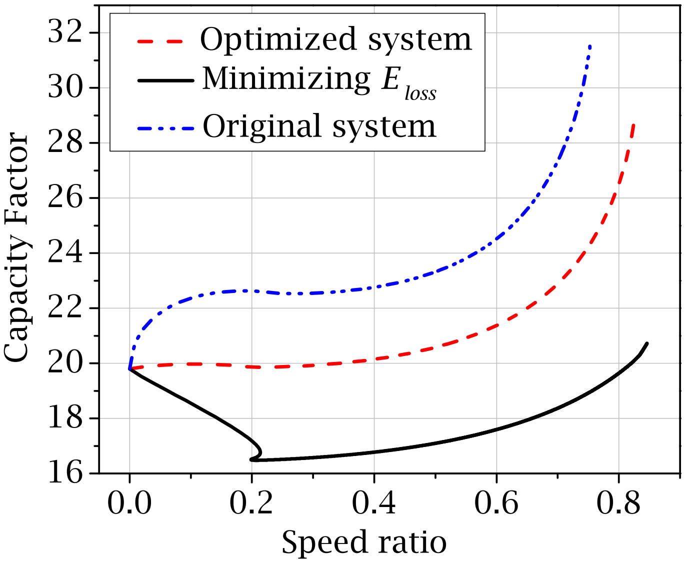


Figure 3.37: Capacity factor vs. speed ratio

From these results it can be inferred that, if the objective function is solely focused on minimization of power losses, it becomes equivalent to the scenario where the optimization routine modifies a torque converter for a truck and makes it more efficient by reducing it to a torque converter suitable for a car. Thus, to ensure that the approach toward optimization is practical and due consideration is given to avoid the factors that limit the usability of results, the only way is to include the capacity factor into the optimization procedure as shown in equation (3.64).

The capacity factor can be included in the optimization process by considering a non-zero value for the constant a_2. The relative value of a_2 with respect to a_1 determines the priority given to the capacity factor criterion. In the following section, the effect of varying priorities to these competing criteria will be illustrated.

Competing criteria

The competitive nature of the two components of the cost function λ_1 and λ_2 can be visualized by plotting the values of the two quantities for different values of model parameters on X and Y axes. Since the design philosophy is to reduce the lost power and counteract the associated decrease in the range of capacity factor, it is necessary to plot λ_1 on the X axis and inverse of λ_2 on the Y axis. λ_1 and λ_2 are defined by (3.62) and (3.63), respectively.

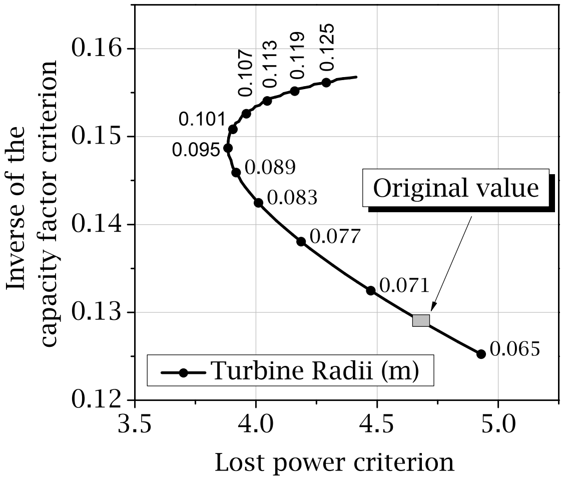


Figure 3.38: Competing criteria for different values of R_t

Figure 3.38 shows the plot of λ_1 on the X axis and $1/\lambda_2$ on the Y axis for different values of R_t. The original value of the model parameter, as shown in table A.1, is also denoted in the figure. It graphically demonstrates the trade off situation that is present in this system. If more priority is assigned to the lost power minimization criterion, then the operating point moves to the left on this plot which in turn results in higher value on the Y axis and an increasingly useless system. On the other hand, if one is particularly keen on maximizing λ_2, then the optimal point moves towards the right of the curve and results in a higher value for λ_1 which results in a system which wastes more energy.

Equation (3.70) refers to the solution where the sole purpose was to minimize λ_1. The resulting optimal value was determined to be $R_t = 0.088\,m$, which is located towards the left edge of the curve. On the other hand, when equal priorities are assigned to λ_1 and λ_2, the resulting optimal value is determined to be $R_t = 0.065\,m$ which is located to the right of the curve.

This analysis can be done for all the other parameters. Figure 3.39 shows the plots with respect to the fluid density ρ. It clearly shows that when all the priorities are assigned to minimize λ_1, the operating point shifts towards the left and settles at $\rho = 880\,Kg/m^3$ as shown in equation (3.70). However when the capacity factor criterion is included in the optimization process the optimal point moves to the right and settles at $\rho = 830\,Kg/m^3$ as shown in equation (3.68). Similar observations can be made for the parameters R_p and R_s as demonstrated in figure 3.40 and 3.41, respectively.

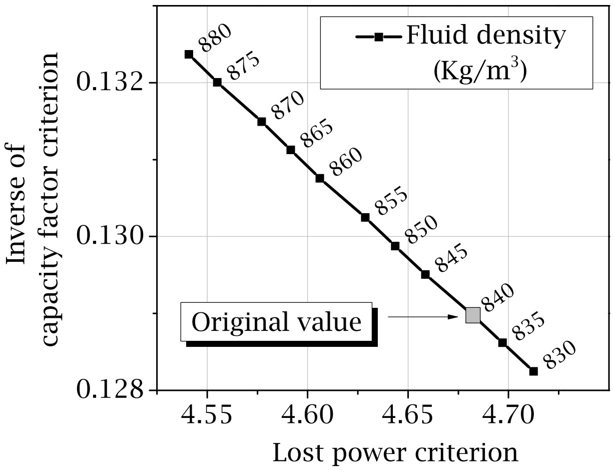


Figure 3.39: Two competing criteria for different values of ρ

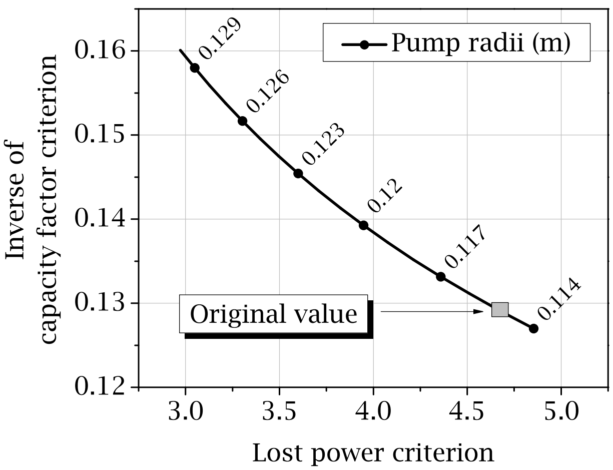


Figure 3.40: Two competing criteria for different values of R_p

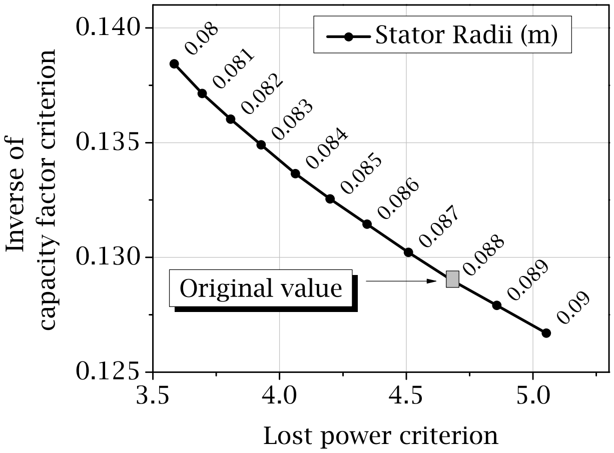


Figure 3.41: Two competing criteria for different values of R_s

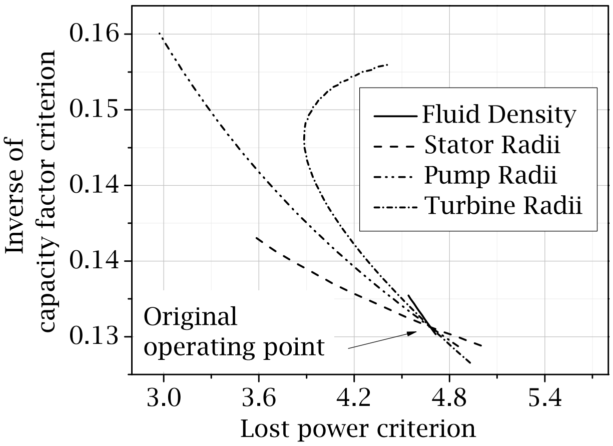


Figure 3.42: Competing criteria for different values of model parameters

The magnitudes of the effects of the perturbation of parameter values on the two competing criteria are different for the four parameters. This can be visualized by comparing the individual Pareto fronts for the four parameters on a single plot as shown in figure 3.42. The slope of the individual curves denote the sensitivity of the particular criterion with respect to the individual model parameters.

Also, it is worth mentioning that the four curves shown in figure 3.42 pass through a single point which represents the original operating point of the system.

Effect of the pump exit angle

The exit vane angle of the pump is one of the important design parameters of a torque converter. In preliminary studies, it was considered as one of the design variables. However, the results showed that the objective function is related to the pump exit angle in a linear fashion. During the optimization iterations, the parameter settled to the value used by the original system, which was the lower limit of acceptable values.

3.3.5 Section summary

A graph-theoretic model of an automotive torque converter was presented in this section. The graph-theoretic framework was developed by looking at the physical construction of a torque converter and identifying different components that store, dissipate, transmit, and transform energy in and between hydraulic and mechanical domains. Using the graph-theoretic model sensitivity equations were generated using the proposed algorithm. The resulting sensitivity information was used to perform parametric importance analysis and solve a design optimization problem.

3.4 Electro-chemical system: Ni-MH battery

Because of its widespread use in electric and hybrid electric vehicles, nickel metal hydride (Ni-MH) battery performance is very important for automotive researchers and manufacturers. The performance of a battery can be described as a direct consequence of various chemical and thermodynamic phenomena taking place inside the container. The current methods of modelling this behaviour can be classified into two broad categories.

The first category, also known as the circuit-based approach, tries to model the behaviour of the battery as an electric circuit, [69], [70], and [71], which results in a conceptually simple model. In this approach, the characteristics of a particular battery is captured by using empirical relationships in the form of look-up tables and the charge discharge behaviour is simulated using simple circuit analysis. However, the actual physical parameters stay hidden and there is no explicit relationship between the model parameters and the battery parameters. As a result, this approach offers little help when it comes to the design and optimization of battery systems. On the other hand, the second approach, or the chemistry based approach, models the actual chemical reactions and other electrochemical and/or thermodynamic processes inside the battery, vide [72], [20], and [73]. This method allows access to physical model parameters of the battery and is suitable for design optimization and other analytical applications.

Both of these approaches can be implemented using linear graph-theory. The circuit based approach can be modelled using the same method used to model electrical systems with modulated non-linear components. The chemistry-based approach requires modelling the sequence of chemical reactions taking place inside the battery. In this section, the graph-theoretic framework formulated by Dao et al. [20] is used to present an overview of graph-theoretic sensitivity analysis of a Ni-MH battery along with the results of the subsequent parametric importance analysis performed to identify key model parameters. An overview of this section is provided below.

At first a brief description of the graph-theoretic model is presented to outline the framework of the modelling approach. Using similar formulations presented in section 3.3, the process of the generating the sensitivity equations from the linear graph will be touched upon next. The general structure of the system and the sensitivity equations will also be presented. Finally, the application of sensitivity analysis will be showcased using the example of parametric importance analysis applied to the model of the NiMH battery.

3.4.1 Graph-theoretic framework

To capture the electro-chemical phenomenon inside the battery, one needs to start from the basic chemical reactions taking place at the individual electrodes. The following chemical equations represent the main reaction activities that happen inside a NiMH battery.

Main reaction on positive electrode:

$$NiOOH + H_2O + e^- \underset{charge}{\overset{discharge}{\rightleftharpoons}} Ni(OH)_2 + OH^- \tag{3.72}$$

Side reaction on positive electrode:

$$2OH^- \rightarrow \frac{1}{2}O_2 + H_2O + 2e^- \tag{3.73}$$

Main reaction on negative electrode:

$$MH + OH^- \underset{charge}{\overset{discharge}{\rightleftharpoons}} M + H_2O + e^- \tag{3.74}$$

Side reaction on negative electrode:

$$\frac{1}{2}O_2 + H_2O + 2e^- \rightarrow 2OH^- \tag{3.75}$$

For each of these reactions, there are two important quantities that define the generation of the electromotive force. The first quantity is known as the open circuit potential difference and is given by the Nernst equation. The general structure of the Nernst equation is shown in equation (3.76). The quantity ϕ denotes the thermodynamically predicted potential difference of the Redox reaction.

$$\phi\left(t\right) = U + \left(T\left(t\right) - T_{ref}\right)\frac{\partial U}{\partial T} - \frac{RT\left(t\right)}{\nu F}\ln\left(\prod_k c_k\left(t\right)\right) \tag{3.76}$$

In equation (3.76), U is the open circuit potential at standard conditions, $T(t)$ is the temperature of the cell, T_{ref} is the reference temperature, $\frac{\partial U}{\partial T}$ is the reversible heat constant for the reaction, R is the gas constant, F is the Faraday constant, ν is the number of electrons per ion, and $c_k(t)$ is the concentration of the reactant.

At this point, it is to be noted that the concentration $c_k(t)$ of the reactants can be related to the charges q_k carried by them using Faraday's law. The charge can then be related to the current density $j_k(t)$ using an appropriate constant ς.

$$c_k = \frac{q_k}{\nu\, F} = \frac{\varsigma}{\nu\, F} \int j_k(t)\, dt \tag{3.77}$$

The second important quantity is known as the overpotential. The electrochemical term *overpotential* refers to the potential difference between the thermodynamically predicted potential and actual observed potential of a reaction. This quantity is related to the current density of the cell by the Butler-Volmer equation. The general structure of the equation for the k^{th} reaction is shown below.

$$j_k\,(t) = 2 i_{0,k}\,(t) \sinh\left(\frac{F}{RT\,(t)} \alpha_k\, \eta_k\,(t)\right) \tag{3.78}$$

In equation (3.78), η_k is the overpotential for the k^{th} reaction, α_k is a dimensionless constant called the "*charge transfer coefficient*", and $i_{0,k}$ is the exchange current density for the reaction. By nature, the exchange current density is similar to the rate constant used in chemical kinetics and it can be related to the ambient temperature and reactant concentration using the following equation.

$$i_{0,k} = i_{0,k,\,ref}\, f\left(c_k\,(t)\right)\; e^{\left[\frac{E_{a,k}}{R}\left(\frac{1}{T(t)} \quad - \quad \frac{1}{T_{ref}}\right)\right]} \tag{3.79}$$

where $E_{a,k}$ is the activation energy, $f(c_k(t))$ is a function of the reactant concentrations, and $i_{\,0,k,\,ref}$ is the reference exchange current density at standard temperature and concentrations.

By looking at the equations governing the open circuit potential, it is possible to work backwards and construct a linear graph based representation of the system, where the individual edges correspond to these equations and can be physically analogous to standard electrical components.

For example, in equation (3.76), the potential difference is calculated by multiplying a constant with the time integral of the current density. This is exactly how a capacitor functions and hence equation (3.76) can be represented by an edge which corresponds to a modulated capacitor. Similarly in equation (3.78), it is observed that the across variable $\eta(t)$ can be expressed as an inverse hyperbolic function of the $j_k(t)$.

$$\eta_k\,(t) = \frac{RT\,(t)}{F\alpha_k} \sinh^{-1}\left(\frac{j_k}{2\, i_{0,k}}\right) \tag{3.80}$$

Equation (3.80) can be identified as the typical behaviour shown by a very nonlinear resistor and thus this equation can be represented by an edge corresponding to a modulated resistive element, whose terminal equation can be readily derived from (3.78).

The modality of the connection of these components are also apparent from the physical phenomena happening inside the cell. By definition, the observed open circuit potential of a reaction is defined as the combined potential difference of the thermodynamic open circuit potential and the overpotential. Since these two phenomena can be represented by a capacitor and a resistor in an analogous electrical circuit, these two components must be placed in series to obtain proper configuration.

The main reactions at the cathode and anode, i.e., equations (3.72) and (3.74), and the reaction denoted in equation (3.73) can be accurately represented as an analogous electrical circuit consisting of modulated capacitors and nonlinear resistors.

For the reduction of oxygen reaction shown in equation (3.75), a limiting current equation can be used for the rate of the reaction. For the electrical analogy, this corresponds to a modulated current source, where the terminal equation takes the form as shown below.

$$j\left(t\right) = -\frac{p_{\mathrm{O}_2}\left(t\right)}{p_{\mathrm{O}_2,\mathrm{ref}}}\left(i_{0,\mathrm{ref}}\right)e^{\left[E_a\,R^{-1}\left(\left(T(t)\right)^{-1}-\left(T_{ref}\right)^{-1}\right)\right]} \tag{3.81}$$

The presented approach is based on the work done by Dao and McPhee [20]. According to their research, all the electro-chemical reactions inside a NiMH battery cell can be represented using the analogous electrical circuit represented by a linear graph, where the terminal equations are based on the equations illustrated above.

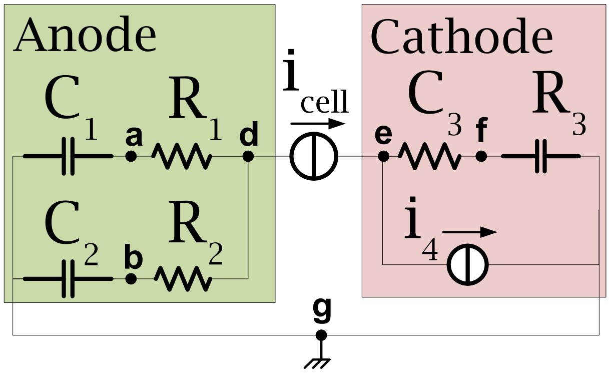


Figure 3.43: Equivalent electrical circuit

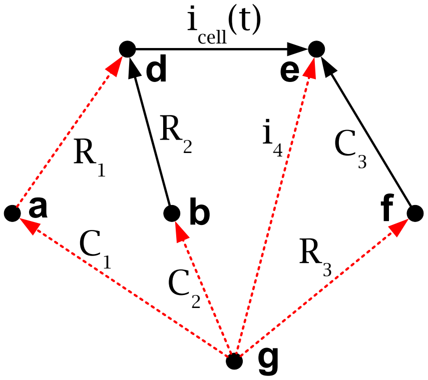


Figure 3.44: Corresponding linear graph

Since the topology of the system are be captured by the structure of the linear graph, the graph-theoretic method offers a more efficient way to model the behaviour of the battery. Furthermore, the use of the branch-chord formulation can further simplify the expressions for both governing and sensitivity equations.

Figure 3.43 shows the equivalent electrical circuit that represents the electrochemical reactions in a NiMH battery. For this example, the thermal aspect of the model will be ignored and the analysis will be done assuming isothermal conditions. Dao and McPhee have modelled the thermodynamic aspect of the Ni-MH battery using a linear graph, further details of the modelling method can be found in the original paper [20].

For this example, it is assumed that the battery is being driven by a constant current source and the internal resistance is negligible. This is reflected in the equivalent electrical circuit and the corresponding linear graph.

3.4.2 Generation of sensitivity equations

Using the formulation presented in 3.2, the sensitivity equations of the system can be generated from the linear graph shown in figure 3.44. For this analysis the edges marked C_1, C_2, R_1, R_2, R_3, and C_3 are selected as branches and the rest are classified as chords.

The across variables for the graph are the voltage differences across the components and the corresponding sensitivities. On the other hand, the through variables for the graph are the currents flowing through the various components. For an arbitrary model parameter b, the sensitivity variables are shown in equation (3.82).

$$\begin{aligned}
\alpha_{branches}^{sensitivity} &= \left[\begin{array}{ccccc} v_b^{C_1} & v_b^{C_2} & v_b^{C_3} & v_b^{R_2} & v_b^{R_3} \end{array} \right]^T \\
\alpha_{chords}^{sensitivity} &= \left[\begin{array}{ccc} v_b^{R_1} & v_b^{4} & v_b^{cell} \end{array} \right]^T \\
\tau_{branches}^{sensitivity} &= \left[\begin{array}{ccccc} i_b^{C_1} & i_b^{C_2} & i_b^{C_3} & i_b^{R_2} & i_b^{R_3} \end{array} \right]^T \\
\tau_{chords}^{sensitivity} &= \left[\begin{array}{ccc} i_b^{R_1} & i_b^{4} & i_b^{cell} \end{array} \right]^T
\end{aligned} \tag{3.82}$$

The concentration of various chemical species play a pivotal role in the behaviour of a electro-chemical reaction. Thus the terminal equations for the edges are often expressed in terms of these concentrations or the charges held by different ions. Since the rate of change of charge is a through variable, the charges held by the components can also be treated as through variables, as they frequently appear in terminal equations.

Using the formulation presented in 3.2, the sensitivity cutset and circuit equations can be generated from the linear graph shown in figure 3.44, which can then be used to generate the sensitivity branch and chord transformation equations that are required for the subsequent generation and simplification of the sensitivity equations. Equation (3.83) shows the chord transformation equations for the current though variables.

$$\begin{bmatrix} i_b^{C_1} \\ i_b^{C_2} \\ i_b^{C_3} \\ i_b^{R_2} \\ i_b^{R_3} \end{bmatrix} = - \begin{bmatrix} -1 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & 1 \\ 1 & 0 & -1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} i_b^{R_1} \\ i_b^{4} \\ i_b^{cell} \end{bmatrix} \tag{3.83}$$

Chord transformation equations are also required for the charge through variables. These equations are structurally similar to the equations shown above and can be derived from the same fundamental cutset matrix. However, for these equations the initial charges on the components must be considered to obtain a proper description of the topology.

$$\begin{bmatrix} q_b^{C_1} - q_b^{C_1}(0) \\ q_b^{C_2} - q_b^{C_2}(0) \\ q_b^{C_3} - q_b^{C_3}(0) \\ q_b^{R_2} - q_b^{R_2}(0) \\ q_b^{R_3} - q_b^{R_3}(0) \end{bmatrix} = - \begin{bmatrix} -1 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & 1 \\ 1 & 0 & -1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} q_b^{R_1} - q_b^{R_1}(0) \\ q_b^{4} - q_b^{4}(0) \\ q_b^{cell} - q_b^{cell}(0) \end{bmatrix} \tag{3.84}$$

The relationships between the across variables are defined by the branch transformation equations. For this system the branch transformation equations for the sensitivity across variables are shown below.

$$\begin{gathered} \begin{bmatrix} v_b^{R_1} \\ v_b^{4} \\ v_b^{cell} \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 \\ 0 & 1 & -1 & 1 & -1 \end{bmatrix} \{\boldsymbol{v}\} \\ \boldsymbol{v} = \begin{bmatrix} v_b^{C_1} & v_b^{C_2} & v_b^{C_3} & v_b^{R_2} & v_b^{R_3} \end{bmatrix}^T \end{gathered} \tag{3.85}$$

The sensitivity terminal equations for the NiMH battery systems can be obtained by symbolically differentiating the terminal equations of the original system. To explain the relationships between the terminal equations and model parameters and to describe the generation of the sensitivity terminal equations, at this point it is necessary to provide a brief description of the terminal equations used by Dao and McPhee.

As described in section 3.4.1, the basic terminal equations are expressed in terms of the concentrations of the chemical species and the current densities in the cell. These quantities are related to the physical properties of the cell and thus a connection can be established between the model parameters, the state variables that are easily measurable, and the terminal equations.

For instance, in equation (3.76), the concentration of the reactants can be expressed as a function of the charges carried by the species using model parameters which are directly related to the physical properties of the NiMH cell. For the chemical reaction shown in equation (3.72) that is represented by the edges C_1 and R_1, the quantity c_{H^+} can be expressed as a function of the model parameters and the charge carried by the component C_1 as shown below.

$$c_{H^+}(t) = \frac{\rho_{NiOH_2}}{F\,A_{pos}\,L_{NiOH_2}}\,q^{C_1}(t) \tag{3.86}$$

In equation (3.86), ρ_{NiOH_2} is the density of the active material, A_{pos} is the actual area of the positive electrode, and L_{NiOH_2} is the loading of the nickel active material. These quantities are assumed to be constants for the simulations presented in this section.

Since the linear graph relates the charges carried by the various components, this type of equation bridges the gap between the topological equations and the governing equations of the chemical reactions.

Current densities can be expressed in terms of the actual current through the components. For the resistor R_2, the relationship is shown below

$$j_2(t) = \frac{1}{A_{pos}\,a_{pos}\,l_{pos}}\,i^{R_2}(t) \tag{3.87}$$

The pressure of the generated oxygen can also be related to the quantities q^{C_2} and q^4 as shown below.

$$p^{O_2}(t) = \frac{RT}{FV_{gas}}\left(q^{C_2}(t) + q^4(t)\right) \tag{3.88}$$

In the above equations, V_{gas} is the volume of the entrapped gas inside the cell and a_{pos} is the specific area of the positive electrode which is a measure of the porosity of the electrode. When multiplied with the thickness of the electrode l_{pos} and the electrode area, the total reacting area is obtained for the calculation.

Using these expressions, the basic governing equations shown in (3.76) - (3.81) are transformed into the terminal equations for the various edges of the linear graph. These equations are expressed in terms of the physical properties of the battery (the model parameters) and can be used to derive the sensitivity terminal equations.

For example, the terminal equation for the component C_2 can be expressed as

$$v^{C_2}(t) = U_2 + (T - Tref)\,\frac{dU_2}{dT} + \frac{RT \ln \frac{\sqrt{\frac{RT\left(q^{C_2}(t) + q^4(t)\right)}{FV_{gas}}}}{{c_e}^2}}{2F} \tag{3.89}$$

By differentiating equation (3.89) with respect to any arbitrary model parameter one can obtain the relationship between the sensitivities of $v^{C_2}(t)$, q^{C_2}, and q^4.

$$\frac{d}{dV_{gas}} v^{C_2}(t) = \frac{V_{gas}\left(\frac{\left(q^{C_2}_{V_{gas}}(t) + q^4_{V_{gas}}(t)\right)RT}{FV_{gas}} - \frac{RT\left(q^{C_2}(t) + q^4(t)\right)}{F{V_{gas}}^2}\right)}{4\left(q^{C_2}(t) + q^4(t)\right)} \tag{3.90}$$

Equation (3.90) shows the sensitivity terminal equation for the edge C_2 with respect to the model parameter V_{gas}. The following notations are used to attain a compact form for the equation.

$$q^{C_2}_{V_{gas}} = \frac{d}{dV_{gas}} q^{C_2} \quad \text{and} \quad q^4_{V_{gas}} = \frac{d}{dV_{gas}} q^4 \tag{3.91}$$

Using the sensitivity cutset, circuit, and terminal equations the complete set of sensitivity equations for the NiMH battery can be derived using methods illustrated in section 3.2. As expected from the theory of graph-theoretic modelling, the equations are generated in terms of the across variables of the branches and the through variables of the chords. For the present selection of branches, the governing and sensitivity equations are formed in terms of the following state variables and the corresponding sensitivities.

$$\begin{aligned} \mathbf{q} &= \left\{i^4,\ i^{R_1},\ i^{cell},\ q^4,\ q^{R_1},\ q^{cell}\right\} \\ \mathbf{q}_b &= \left\{i^4_b,\ i^{R_1}_b,\ i^{cell}_b,\ q^4_b,\ q^{R_1}_b,\ q^{cell}_b\right\} \end{aligned} \tag{3.92}$$

The use of branch-chord formulation significantly reduces the number of equations to be solved. The graph-theoretic method generates 4 nonlinear algebraic equations and 6 equations that connect the electrical charges to the current variables.

Since the quantities i^{cell} and i^{cell}_b are known functions of time, this leaves a set of differential-algebraic equations with four algebraic constraint equations and four simple differential equations which can be readily transformed into a set of ODEs using substitution. This combined set of governing and sensitivity equations can be solved numerically to determine the sensitivities of all the state variables.

3.4.3 Numerical simulation and system response

To obtain the response of the system and the sensitivities of the state variables, the set of governing equations and sensitivity equations are generated and solved numerically using the `dsolve[numeric]` routine built into *Maple*. Employing the DAE extension of the *Rosenbrock* algorithm, the integration was carried out with relative and absolute error tolerances set to 1×10^{-6} and 1×10^{-7} respectively.

The thermal aspect of the cell behaviour and the internal resistance of the battery are neglected and a constant value of i^{cell} is assumed for the simulation. The values of the model parameters used are listed in Tables A.6 and A.7. The results shown in this section are obtained by performing sensitivity analysis with respect to the loading of nickel active material L_{NiOH_2}.

Table 3.7: Specified initial conditions for the NiMH battery

Variable	**Expression or value**
q^{C_1}	$\frac{F\,A_{pos}}{\rho_{NiOH_2}}\,c_H\,(0)$
q^{C_3}	$\frac{F\,A_{neg}\,L_{MH}}{\rho_{MH}}\,c_{MH}\,(0)$
q^4	$\frac{FV_{gas}}{RT}p_{O_2}\,(0)$
$c_H(0)$	$c_{H,\max} \times 10^{-2}$
$c_{MH}(0)$	$(1 - 10^{-2}) \times c_{MH,\max}$
q^{R_1}	0
q^{C_2}	0
q^{R_2}	0
q^{R_3}	0
q^{cell}	0
$p_{O_2}(0)$	1×10^{-6}

To ensure proper simulations, the initial conditions for the system were evaluated to be consistent with the governing equations of the system. Since the state variables of the models are not directly related to physically measurable quantities, their initial values are calculated using the initial values of the physical quantities they are related to.

For example, the quantity q^{C_1} is related to the concentration of the $NiOH_2$ in the electrode through the following equation, using which the value of $q^{C_1}(0)$ can be evaluated from a specified initial value for the quantity $c_H(t)$.

$$q^{C_1}(t) = \frac{F\, A_{pos}\, L_{NiOH_2}}{\rho_{NiOH_2}}\, c_H(t) \tag{3.93}$$

The other state variables are calculated using the constraint equations to ensure that all the initial conditions are consistent. Table 3.7 shows the list of specified initial conditions used for the simulation.

The initial values for the sensitivities of the state variables are calculated from these specified initial conditions. For example, for the quantity q^{C_1} the initial value of the corresponding sensitivity variable ${q^{C_1}}_{L_{NiOH_2}}$ can be determined by differentiating the expression shown in equation 3.93 with respect to the model parameter L_{NiOH_2}.

$$q^{C_1}_{L_{NiOH_2}}(0) = \frac{F\, A_{pos}}{\rho_{NiOH_2}}\, c_H(0) \tag{3.94}$$

Similar equations can be used to determine the initial values of q^{C_3}, q^4, and their sensitivities from the initial values of c_{MH}, the concentration of metal hydride and p_{O_2}, the pressure of oxygen inside the cell.

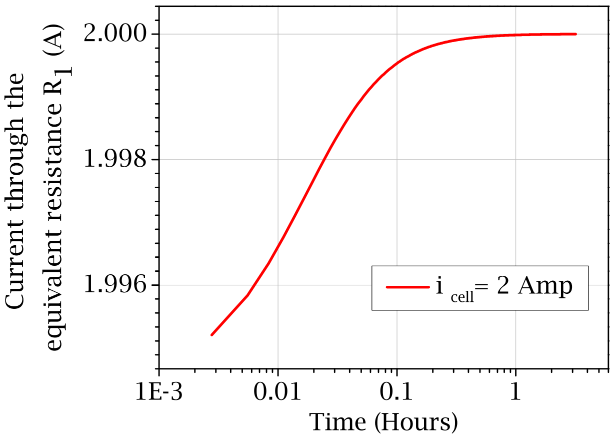


Figure 3.45: i^{R_1} vs. $time$

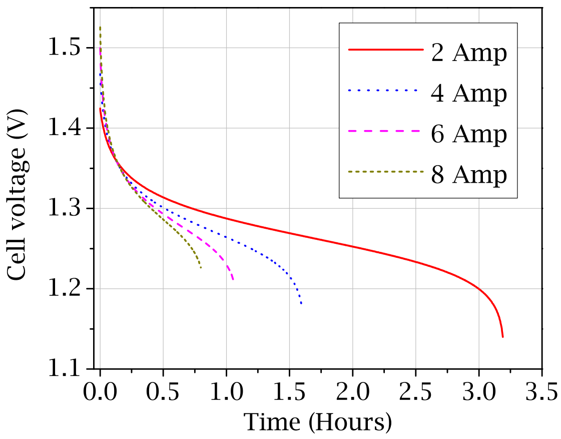


Figure 3.46: v^{cell} vs. $time$

The integration of the set of equations result in the time histories of the state variables shown in equation 3.92 and their sensitivities. These plots can provide physical insight into the electro chemical activities inside the cell. For example, the quantity i^{R_1} is related to the rate of the main reaction happening at the positive electrode. The plot shown in figure 3.45 shows the variation of $i^{R_1}(t)$ with time, which clearly shows the gradual increase of the rate of reaction at the positive electrode until the current matches that of i^{cell}.

The other relevant variables can be evaluated by using the branch transformation equations and the terminal equations. For example, the quantity v^{cell} can be expressed as

$$v^{cell} = -v^{R_3} + v^{C_2} - v^{C_3} + v^{R_2} \tag{3.95}$$

which can be used with the terminal equations to evaluate v^{cell}. The plot shown in figure 3.46 shows the variation of v^{cell} with time for different constant values of i^{cell}.

The sensitivity of v^{cell} can also be calculated from the branch transformation equations. For this example the following expression can be obtained for the sensitivity of cell voltage with respect to the model parameter.

$$v^{cell}_{L_{NiOH_2}} = -v^{R_3}_{L_{NiOH_2}} + v^{C_2}_{L_{NiOH_2}} - v^{C_3}_{L_{NiOH_2}} + v^{R_2}_{L_{NiOH_2}} \tag{3.96}$$

To compare the solutions, the same reactions of the NiMH battery are modelled using lumped parameter equations as illustrated by Banerjee et al [74]. The governing equations presented in this paper are a set of differential algebraic equations that captures the same chemical reactions modelled by the linear graph presented in this chapter.

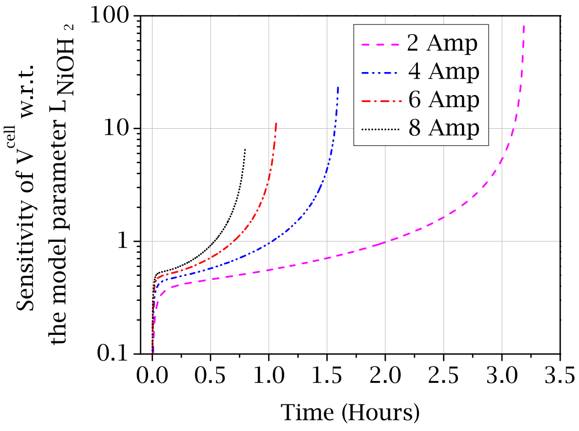


Figure 3.47: $v^{cell}_{L_{NiOH_2}}$ vs. *time*

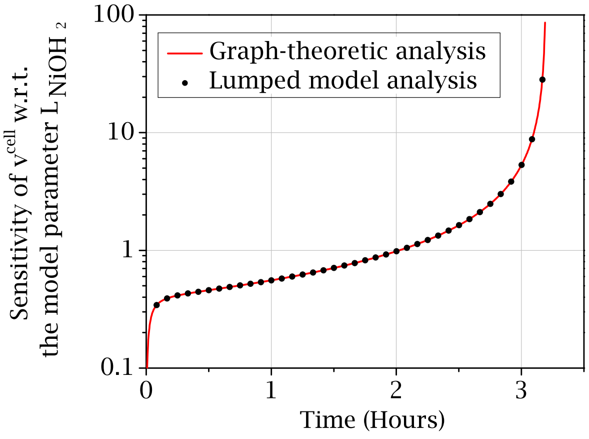


Figure 3.48: $v^{cell}_{L_{NiOH_2}}$ vs. *time*

Figure 3.48 shows the variation of $v^{cell}_{L_{NiOH_2}}$ obtained using direct differentiation onto the equations presented in the paper [74]. It also demonstrates the congruity of the solution with that obtained using the graph-theoretic method being presented in this chapter.

The variation of the quantity $v^{cell}_{L_{NiOH_2}}$ depends on the discharge current of the system. Figure 3.47 shows the variation of the sensitivity of v^{cell} with respect to the model parameter L_{NiOH_2} for different values of i^{cell}.

The benefit of using a graph-theoretic formulation is also reflected in the reduced amount of time that is required to integrate the resulting equations. To compare the complexity of the generated equations, the integration time is measured for the simulation of the graph-theoretic sensitivity formulation for the NiMH battery.

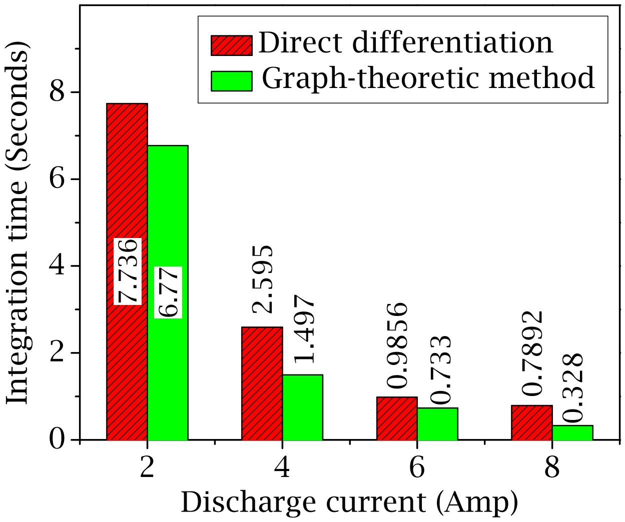


Figure 3.49: Comparison of integration time for two algorithms

It is then compared to the simulation time of sensitivity analysis of the model presented by Banerjee et al. using direct differentiation. Initial conditions are properly chosen to ensure that the two approaches simulate the same system undergoing the same physical process. The figure above shows the differences in the integration times while using the two competing methods for different values of i^{cell}. It clearly identifies the superior efficiency of the graph-theoretic method for sensitivity analysis.

Figure 3.49 clearly shows that the integration time decreases as the value of i^{cell} increases. This can be readily explained by the fact that when higher currents are used for the discharge, the reactions generally speed up resulting in a shorter discharge time, which in turn results in a decreased integration time.

Table 3.8 lists the integration times for the two methods along with their differences. The graph theoretic method shows clear improvement in the integration time for all values of discharge current. This indicates a more efficient set of governing equations. It is also observed that the relative decrease in the integration time increases with the value of i^{cell}.

Table 3.8: Integration times for the two competing methods

Discharge Current	**Direct Differentiation**	**Graph-theoretic Method**	**% Reduction**
2 A	7.736 s	6.770 s	12.5
4 A	2.595 s	2.122 s	18.2
6 A	0.986 s	0.733 s	25.6
8 A	0.789 s	0.328 s	58.4

3.4.4 Application: parametric importance analysis

To showcase the application of sensitivity information for the NiMH battery, in this section a parametric importance analysis will be presented to identify key model parameters.

Importance analysis is used to identify the parameters which have the greatest effect on a chosen performance measure. The results are dependent on the choice of the objective function and are often valid specifically for the context of the study. In this example, the efficiency of the battery is considered as the performance measure. Mathematically it can be defined as the ratio of the energy available during discharge and the energy spent while charging it. For constant charging and discharging currents, the equation reduces to the form illustrated in equation (3.97).

$$f\left(v^{cell}, b, t\right) = \frac{i^{cell\ ,\text{discharge}} \times {}_{\text{discharge}}\int v^{cell}\left(b, t\right)\ dt}{i^{cell\,,charge} \times {}_{charge}\int v^{cell}\left(b, t\right)\ dt} \tag{3.97}$$

To evaluate the effect of the perturbation of individual model parameters on this function, the sensitivity of the quantity $f\left(v^{cell}, b, t\right)$ needs to be determined. To do that, equation (3.97) is differentiated with respect to an arbitrary model parameter b to obtain the expression for the sensitivity of the performance measure.

$$f_b = \frac{\left(E_b^{\text{discharge}} \times E^{\text{charge}}\right) - \left(E^{\text{discharge}} \times E_b^{\text{charge}}\right)}{\left(E^{\text{charge}}\right)^2} \qquad E_b = \left(i^{cell} \times \int \frac{d}{db}\left(v^{cell}\right)\ dt\right) \tag{3.98}$$

Equation (3.98) demonstrates the dependence of the sensitivity of the objective function on the sensitivity of the state variable v^{cell}. Various methods can be used to evaluate the state variables and their sensitivities. For this study, equations generated using the proposed graph-theoretic formulation were used to generate the sensitivity information.

Normalized importance score

The sensitivity of the objective function given in equation (3.98) can be used to determine which model parameter has greater influence on the efficiency of the battery. This procedure is known as importance analysis. While equation (3.98) yields the absolute sensitivity of the objective function, to compare the influence of different parameters on the objective function, a normalized sensitivity scale must be defined as shown in equation (3.99).

$$\mu = \frac{\partial f}{\partial b} \ \times \ \left| \frac{b^*}{f\left(b^*\right)} \right| \tag{3.99}$$

In the above equation b^* and $f(b^*)$ are the nominal values of the parameter and the objective function respectively, and $(\partial f / \partial b)$ is the absolute sensitivity.

This normalization brings all the parameters to a common ground for comparison. A greater absolute value of the relative sensitivity indicates higher influence of the parameter on the efficiency, i.e., a more important parameter (Smith et al. [68]). In reality the change in the objective function f due to a parameter perturbation can be both negative and positive. However, to determine the most important parameter, the absolute values of the relative sensitivity scores would be presented in this study.

The choice of the model parameters for importance analysis depends on the intended application. For quality control studies, since it is not possible to enforce quality control on every parameter, it is not worthwhile to study the entire list. On the other hand, model reduction and parameter estimation would require extensive importance analysis of all the model parameters.

In the presented system, the material densities can be considered as constants. However the loading of the active material at the electrodes denoted by L_{MH} and L_{NiOH_2} are parameters which are affected by many factors. For this reason it is worthwhile to study sensitivities with respect to these parameters. Apart from these parameters, the thermal properties and the parameters related to the structure of the battery are expected to be critical for the performance.

To demonstrate the process, five parameters are chosen for the importance analysis.

$$\mathbf{p} = \left\{ \, c_e, \ \ L_{MH}, \ L_{Ni(OH)_2}, \ R_{\text{int}}, \ V_{gas} \right\}^T \tag{3.100}$$

Assuming a constant current of $i^{cell} = 0.5Amp$ for both charging and discharging processes, an importance analysis was performed with respect to the parameters listed in (3.100). By substituting the solution obtained by solving the sensitivity equations into the expression for sensitivity of the efficiency, i.e., equation (3.98), the absolute sensitivities of the efficiency of the battery with respect to its parameters are obtained.

Using this procedure, considering one parameter at a time, the relative importance scores are calculated for all the selected parameters listed in (3.100). To compare the importance of these parameters, the absolute sensitivity values are subsequently converted into relative sensitivity scores using equation (3.99). Figure 3.50 shows the order of importance for each of the five parameters.

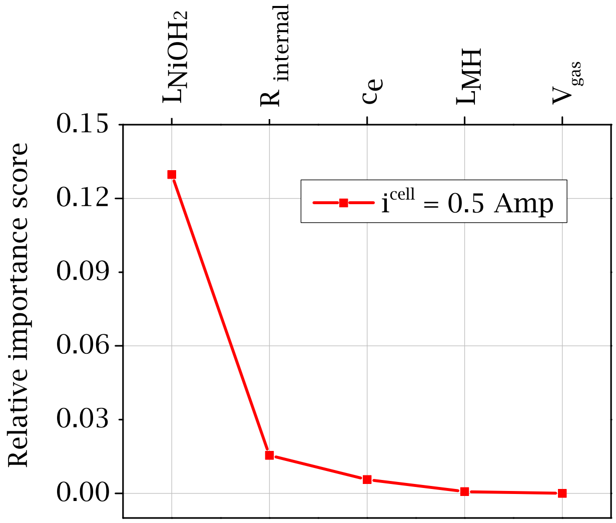


Figure 3.50: Importance order for the model parameters at $i^{cell} = 0.5Amp$

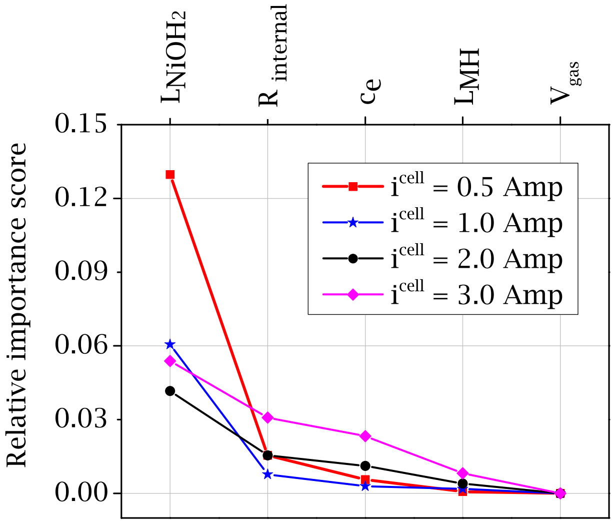


Figure 3.51: Importance scores for the model parameters at different i^{cell}

These results identify the loading of the nickel active electrode material to be the most important parameter in the context of the efficiency of the system. In comparison, the other parameters were found to have minimal effect on the performance of the battery. However it is to be noted that, the importance order obtained by this analysis is specific for the current operating condition. To determine the effect of different charging / discharging currents on the order of importance, importance analysis is performed using different values for i^{cell}. Figure 3.51 shows the variation of relative sensitivity scores of model parameters.

The figure demonstrates that the loading of nickel hydroxide has the highest influence on the efficiency of the battery for all currents. It also shows that, as the current through the cell is increased, the influences of the model parameters on the efficiency decrease at first and then start to increase. This is observed for all the parameters under study. However, the relative importance order of the model parameters is shown to remain constant for different cell currents.

The fact that the loading of nickel active material has the highest influence on the efficiency of the system is also observable indirectly through the general behaviour of the cell. The parameter L_{NiOH_2} refers to the surface density of an active electrode material. Its effect on the efficiency of the battery is the main reason behind the gradual degradation of the state of health of Ni-MH batteries.

It is a well known fact that although the cell reactions are reversible in theory, in practise the amount of available active material inside the cell gradually decreases, due to the formation of small amounts of non-reactive byproducts. This directly alters the efficiency of the battery in a negative fashion and reduces the amount of energy available from it. Though it has not been a part of this presentation it is worthwhile to mention that the formation of non-reactive by-products also increases the internal resistance, which further reduces the amount of available energy and eventually makes it unusable.

3.4.5 Example summary

In an attempt to illustrate the application of graph-theoretic sensitivity analysis to an electro-chemical system, a graph-theoretic model of nickel metal hydride battery [20] is analyzed in this section.

The used model is based on the actual electro-chemical process and it captures the charging discharging characteristics of a typical Ni-MH battery along with the effects of side reactions taking place at the electrodes. The chemistry-based approach ensures a direct correlation between the model parameters and the actual physical parameters of the battery. The graph-theoretic model enabled the application of graph-theoretic sensitivity analysis.

A brief outline of the modelling approach is illustrated in the context of the supporting graph-theoretic framework. In light of this discussion, the process of the application of graph-theoretic sensitivity analysis of the Ni-MH battery is explained. Next, simulation results for the system behaviour and the sensitivity information are presented. Finally a parametric importance analysis is performed to identify key model parameters and to showcase the application of sensitivity information.

The software *Maple* is used in this study, for the simulations and analysis. The cutset, circuit and terminal equations of the sensitivity domain are derived using symbolic operations in **Maple**. The final assembly of sensitivity equations are also carried out using Maple's symbolic engine. The generated sensitivity model is completely analytical and permits the study of physical battery parameters. The simulation is done by numerically integrating the governing set of DAEs.

The loading of the nickel active material is found to be the most important parameter as far as the efficiency of the battery is concerned. The gas volume of the cell is found to be the least influential parameter for the cell efficiency. This section demonstrates that the proposed method of graph-theoretic sensitivity analysis is equally applicable for complicated electro-chemical systems and can be used to generate efficient sets of sensitivity equations.

3.5 Chapter summary

The application of graph-theoretic sensitivity analysis to scalar systems was presented in this chapter. A brief discussion was included to provide a theoretical background about the subject. The formulation was presented using an example of a simple spring mass damper system.

A hydrodynamic torque converter model was used to demonstrate the application of the algorithm. A graph-theoretic model of the torque converter was developed and subsequently, graph-theoretic sensitivity analysis was used on the model to identify important model parameters. A design optimization problem was also presented to showcase the application of sensitivity analysis. The graph-theoretic model of the torque converter was presented at the 7^{th} *Vienna International Conference on Mathematical Modelling* in 2012 and has since been accepted for publication in the MathMOD proceedings at the International Federation of Automatic Control's online server [21].

The sensitivity analysis of the torque converter and the results of the design optimization process was presented at SAE World Congress 2012 and has been published in the *SAE International Journal of Passenger Cars* [75].

As a second example, the algorithm was applied to an electro-chemical model of a NiMH battery. An existing graph-theoretic model was used to generate the sensitivity equations and the resulting sensitivity information was used to identify critical model parameters important for the performance of the battery. The results of the importance analysis were presented at the SAE World Congress 2011 [74].

Chapter 4

Graph-theoretic sensitivity analysis of multibody systems

4.1 Introduction

This chapter will present the derivation and application of graph-theoretic sensitivity analysis on three-dimensional multibody systems. The implementation of the proposed graph-theoretic sensitivity analysis algorithm for three-dimensional multibody systems poses some unique challenges. This chapter will begin with a brief overview of these salient aspects.

Using the theoretical background, graph-theoretic sensitivity analysis for multibody systems will be formulated and presented. A simple pendulum model will be used to describe the formulation procedure and the subsequent results of sensitivity analysis. The flexibility of the proposed algorithm will also be demonstrated using the same model.

4.2 Theory

Compared to systems that involve electrical and hydraulic domains, multi-body systems are unique in quite a few aspects. For this reason, algorithms used for modelling and subsequent analyses must be properly adapted to obtain correct and efficient results. In this section, the theoretical derivation of the graph-theoretic sensitivity analysis for three-dimensional multibody systems will be presented. To illustrate the process of graph-theoretic sensitivity analysis, it is worthwhile to look at the key differences between multibody systems and systems involving other domains that makes the process unique.

4.2.1 Salient features of multibody systems

The main differences in the modelling and analyses methods used for multibody systems originate from the following factors.

Translational and rotational domain: As discussed in section 2.1.1, a proper description of multibody systems require the consideration of both translational and rotational domains. Since both translational and rotational motion variables are required to describe the motion of a rigid body, both domains must be included to derive the corresponding sensitivities. For this reason, during the software implementation of graph-theoretic sensitivity analysis of multibody systems, separate variables are required to include sensitivity variables from both domains.

Through and across variables: As mentioned in the Chapter 2, the through and across variables used to describe multibody systems are vectors. Hence, the corresponding sensitivities are also represented by vector quantities. Also, the cutset and circuit equations for both translational and rotational domains take the form of vector equations. This requires the use of different data structures and certain types of projection methods for proper derivation of system and sensitivity equations.

Rotational circuit equations: For multibody systems, the direction cosine matrices that define the three-dimensional orientation of the constitutive rigid bodies and joints are the rotational across variables. The rotational circuit equations capture the relationships between these variables and take the general form as shown below.

$$\mathbf{R}_1\mathbf{R}_2\cdots\mathbf{R}_n = \mathbf{U}^{3\times 3} \tag{4.1}$$

where $\mathbf{U}$ is the identity matrix, which is the product of all the rotation matrices around a closed kinematic loop.

Similarly, the relationship between the sensitivities of the rotational across variables are captured by a set of equations that will be referred to as the sensitivity circuit equations. These equations can be derived from the rotational circuit equations by direct differentiation. However, since the rotational circuit equations are expressed as products of rotation matrices, the chain rule of differentiation must be used to obtain the proper expression for the corresponding sensitivity equations. This step is specific for multibody systems. To illustrate the process, a sensitivity circuit equation is generated from the circuit equation (4.1), by differentiating the expression of equation (4.1) with respect to an arbitrary model parameter b as shown in the equation below.

$$\frac{\partial \mathbf{R}_1}{\partial b}\mathbf{R}_2....\mathbf{R}_n + \mathbf{R}_1\frac{\partial \mathbf{R}_2}{\partial b}....\mathbf{R}_n + ... + \mathbf{R}_1\mathbf{R}_2....\frac{\partial \mathbf{R}_n}{\partial b} = \mathbf{0}^{3\times 3} \tag{4.2}$$

The derivation of sensitivity equations for multibody systems requires the evaluation of the derivatives of rotation matrices. At this point, it is important to illustrate the physical implications of this process.

4.2.2 Derivatives of rotation matrices

Rotation matrices represent the three-dimensional orientation of a rigid body. For analyses of multibody dynamics, the rotation matrices are often written in terms of the time varying Euler angles, which are state variables. To illustrate the basic structure of a rotation matrix, the following example is considered. The matrix $\mathbf{R}$ corresponds to a uniaxial rotation of an angle $\alpha\left(t,b\right)$ about the X axis.

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\left(\alpha\left(t,b\right)\right) & \sin\left(\alpha\left(t,b\right)\right) \\ 0 & -\sin\left(\alpha\left(t,b\right)\right) & \cos\left(\alpha\left(t,b\right)\right) \end{bmatrix} \tag{4.3}$$

The general structure of a rotation matrix does not depend on model parameters explicitly. However, it can be clearly seen that the rotation matrices are implicitly dependent on the model parameters through the state variables. When a rotation matrix is differentiated, the derivative does not retain the properties of a rotation matrix, such as the property of orthogonality which requires the following relationship to be true.

$$\mathbf{R}\mathbf{R}^{\mathrm{T}} = \mathbf{R}^{\mathrm{T}}\mathbf{R} = \mathbf{U}^{3\times 3} \tag{4.4}$$

It can be easily verified that the derivative of the matrix $\mathbf{R}$ does not satisfy the relationship shown in equation (4.4). However, it can be shown that the matrix $\mathbf{R}_b$, i.e., the derivative of the rotation matrix $\mathbf{R}$, can be expressed as the product of a skew symmetric matrix $\mathbf{S}$ and the rotation matrix $\mathbf{R}$ itself:

$$\frac{d\mathbf{R}}{db} = \mathbf{R}_b = \mathbf{S}\mathbf{R} \tag{4.5}$$

This property is useful in deriving the sensitivity circuit equations for a system. The proof of this property is provided in Appendix B.2. For the matrix shown in equation (4.3), the matrix $\mathbf{S}$ can be evaluated as

$$\mathbf{S} = \mathbf{R}_b \mathbf{R}^T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \frac{d}{db}\alpha\left(t, b\right) \\ 0 & -\frac{d}{db}\alpha\left(t, b\right) & 0 \end{bmatrix} \tag{4.6}$$

From equation (4.6) some physical insights can be obtained about the nature of the matrix $\mathbf{S}$. It can be seen that physically, the matrix represents the sensitivity of the three-dimensional orientation of the body with respect to the model parameter in a skew symmetric form. In other words, the matrix $\mathbf{S}$ holds the changes in orientation of a rigid body about the three axes for an infinitesimal perturbation in the value of the parameter. Since $\mathbf{R}$ from equation (4.3) represents an uniaxial rotation, the corresponding $\mathbf{S}$ matrix holds uniaxial components only. However, for a general rotation matrix, the $\mathbf{S}$ matrix will contain the sensitivity terms about all axes. This is illustrated in the example given below.

Equation (4.7) shows a general time-varying rotation matrix written in terms of the three Euler angles $\zeta - \eta - \xi$ denoting a Euler intrinsic (body-fixed) rotation scheme X - Y - Z .

$$R = \begin{bmatrix} \mathrm{c}\,\xi\,\mathrm{c}\,\eta & \mathrm{c}\,\xi\,\mathrm{s}\,\eta\,\mathrm{s}\,\zeta + \mathrm{s}\,\xi\,\mathrm{c}\,\zeta & -\,\mathrm{c}\,\xi\,\mathrm{s}\,\eta\,\mathrm{c}\,\zeta + \mathrm{s}\,\xi\,\mathrm{s}\,\zeta \\ -\,\mathrm{s}\,\xi\,\mathrm{c}\,\eta & -\,\mathrm{s}\,\xi\,\mathrm{s}\,\eta\,\mathrm{s}\,\zeta + \mathrm{c}\,\xi\,\mathrm{c}\,\zeta & \mathrm{s}\,\xi\,\mathrm{s}\,\eta\,\mathrm{c}\,\zeta + \mathrm{c}\,\xi\,\mathrm{s}\,\zeta \\ \mathrm{s}\,\eta & -\,\mathrm{c}\,\eta\,\mathrm{s}\,\zeta & \mathrm{c}\,\eta\,\mathrm{c}\,\zeta \end{bmatrix}$$
$$\mathrm{c} \equiv \cos \ \text{and} \ \mathrm{s} \equiv \sin \tag{4.7}$$

The corresponding $\mathbf{S}$ matrix is shown in equation (4.8) .

$$\begin{aligned} S &= \begin{bmatrix} 0 & -\psi_3 & \psi_2 \\ \psi_3 & 0 & -\psi_1 \\ -\psi_2 & \psi_1 & 0 \end{bmatrix} \\ \psi_1 &= -\left(\frac{\partial \eta}{\partial b}\right)\sin\xi - \left(\frac{\partial \zeta}{\partial b}\right)\cos\eta\cos\xi \\ \psi_2 &= -\left(\frac{\partial \eta}{\partial b}\right)\cos\xi + \left(\frac{\partial \zeta}{\partial b}\right)\cos\eta\sin\xi \\ \psi_3 &= -\left(\frac{\partial \zeta}{\partial b}\right)\sin\eta - \left(\frac{\partial \xi}{\partial b}\right) \end{aligned} \tag{4.8}$$

The three elements ψ_1, ψ_2 and ψ_3 represent the infinitesimal changes in the orientation about the three axes for a differential change in the parameter b. They are dependant on not only the state variables ζ, η and ξ but also on the corresponding sensitivities ζ_b, η_b and ξ_b.

The concept of rotations about three orthogonal axes raises the question about the nature of the elements of the matrix $\mathbf{S}$. From the looks of it, it appears that these rotations ψ_1, ψ_2 and ψ_3 are acting like vectors. However, it is a well known fact that three-dimensional rotations are not vector quantities at all. To resolve this apparent contradiction, one must consider the fact that these elements refer to infinitesimal rotations only. Since the order of rotation does not matter for infinitesimal rotations, they can be treated as vector-like quantities.

4.2.3 Derivation of sensitivity equations

As discussed in the previous chapter, to derive the sensitivity equations from the linear graph of the system, three kinds of equations must be generated. In this section, the derivation of the sensitivity equations for a three-dimensional multibody system will be illustrated using the example of a simple one-body pendulum with a revolute joint. Figure 4.1 shows the linear graph of the system shown in figure 2.1. The labels identify the nature of the components and the identifiers used to denote their association with the through and across variables of the system.

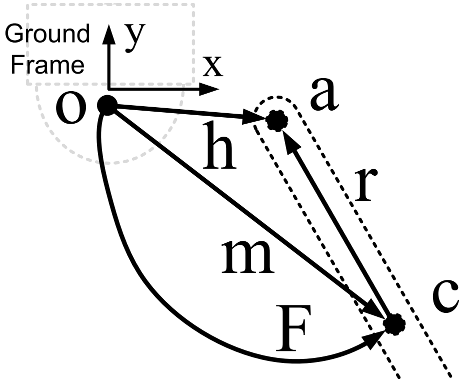


Figure 4.1: Linear Graph of a simple pendulum

During the course of this derivation, variables will be assumed to be three-dimensional in nature and consequently, they will be represented by vectors or matrices. Using the formulation presented in section 2.1.1 both the governing and the sensitivity equations will be generated from the linear graph. The following variables are used to denote the system:

Translational through variables : $\bar{F}^h, \bar{F}^r, \bar{F}^m, \bar{F}^F$

Rotational through variables : $\bar{T}^m, \bar{T}^r, \bar{T}^h, \bar{T}^F$

Translational across variables : $\bar{r}^m, \bar{r}^r, \bar{r}^h, \bar{r}^F$

Rotational across variables : $\mathbf{R}^m, \mathbf{R}^r, \mathbf{R}^h, \mathbf{R}^F$

Apart from the position-level variables, the time derivatives or the velocity and acceleration level variables are also used as across variables.

Velocity level translational across variable: $\bar{v}^m, \bar{v}^r, \bar{v}^h, \bar{v}^F$

Velocity level rotational across variables: $\bar{\omega}^m, \bar{\omega}^r, \bar{\omega}^h, \bar{\omega}^F$

Acceleration level translational across variables: $\bar{a}^m, \bar{a}^r, \bar{a}^h, \bar{a}^F$

Acceleration level rotational across variables: $\bar{\alpha}^m, \bar{\alpha}^r, \bar{\alpha}^h, \bar{\alpha}^F$

The subscript b refers to a scalar parameter with respect to which the sensitivity analysis is being carried out. For analyses with respect to more than one parameters, similar variables must be considered for each and every parameter under study.

To generate the sensitivity equations, the sensitivities of the aforementioned variables are also required. They key point here is the fact that the sensitivity of a through variable is a through variable itself and a similar statement can be made about the sensitivity of an across variable. The sensitivity equations are the governing equations that dictate the behaviours of these variables. As such, for the purpose of the generation of sensitivity equations, the sensitivities of the through and across variables will be referred to as the sensitivity through variables and sensitivity across variables respectively. For this system, the following sensitivity variables are defined.

Translational sensitivity through variables : $\bar{F}^h_b, \bar{F}^r_b, \bar{F}^m_b, \bar{F}^F_b$

Rotational sensitivity through variables : $\bar{T}^m_b, \bar{T}^r_b, \bar{T}^h_b, \bar{T}^F_b$

Translational sensitivity across variables : $\bar{r}^m_b, \bar{r}^r_b, \bar{r}^h_b, \bar{r}^F_b$

Velocity level variables: $\bar{v}^m_b, \bar{v}^r_b, \bar{v}^h_b, \bar{v}^F_b$

Acceleration level variables : $\bar{a}^m_b, \bar{a}^r_b, \bar{a}^h_b, \bar{a}^F_b$

Rotational sensitivity across variables : $\mathbf{R}^m_b, \mathbf{R}^r_b, \mathbf{R}^h_b, \mathbf{R}^F_b$

Velocity level variables: $\bar{\omega}^m_b, \bar{\omega}^r_b, \bar{\omega}^h_b, \bar{\omega}^F_b$

Acceleration level variables : $\bar{\alpha}^m_b, \bar{\alpha}^r_b, \bar{\alpha}^h_b, \bar{\alpha}^F_b$

At this point, it is worthwhile to note that while the position-level variables for the rotational domain are tensors or matrices, the velocity and acceleration-level variables are in fact vectors and should be treated accordingly.

Topological equations

For three-dimensional multibody systems, apart from the usual cutset and circuit equations, separate sets of equations are generated to describe the relationships between the sensitivity through variables or the sensitivity across variables. For the purpose of the generation of topological equations for both the original and the sensitivity variables, it is necessary to consider a separate graph for the sensitivity variable, one which is topologically identical to the original graph. Formulation methods like the selection of a tree is also necessary for the sensitivity variables. However, it is possible to select different trees for the system and the sensitivity graphs. For this example, the revolute joint h and the rigid arm r are chosen as the tree branches for both graphs and the following equations are generated. This particular choice of branches results in equations in terms of the joint angle θ^h and the corresponding sensitivity θ^h_b.

Translational cutset equations

Cutset equations are generated from the system graph and they contains the relationships between the through variables. For this system the translational cutset equations and the corresponding sensitivity cutset equations are shown in equations (4.9) and (4.10), respectively.

$$\begin{bmatrix} \bar{F}^h + \bar{F}^m + \bar{F}^F \\ \bar{F}^r - \bar{F}^m - \bar{F}^F \end{bmatrix} = 0 \tag{4.9}$$

$$\begin{bmatrix} \bar{F}^h_b + \bar{F}^m_b + \bar{F}^F_b \\ \bar{F}^r_b - \bar{F}^m_b - \bar{F}^F_b \end{bmatrix} = 0 \tag{4.10}$$

Equations (4.9) and (4.10) can also be written as a set of translational chord transformation equations. These equations express the translational through variables of the branches in terms of the translational through variables of the chords. The combined set of general and sensitivity chord transformation equations are given in equation (4.11).

$$\begin{bmatrix} \bar{F}^h \\ \bar{F}^h_b \\ \bar{F}^r \\ \bar{F}^r_b \end{bmatrix} = \begin{bmatrix} -\bar{F}^m - \bar{F}^F \\ -\bar{F}^m_b - \bar{F}^F_b \\ \bar{F}^m + \bar{F}^F \\ \bar{F}^m_b + \bar{F}^F_b \end{bmatrix} \tag{4.11}$$

Rotational cutset equations

For the rotational domain, the cutset equations are

$$\begin{bmatrix} \bar{T}^{h} + \bar{T}^{m} + \bar{T}^{F} \\ \bar{T}^{r} - \bar{T}^{m} - \bar{T}^{F} \end{bmatrix} = 0 \tag{4.12}$$

In physical terms, these equations denote the balance of torques acting through different components of the system. The corresponding sensitivity equations or the rotational sensitivity cutset equations can be generated from the sensitivity graph as well and are given in equation (4.13).

$$\begin{bmatrix} \bar{T}^{h}_{b} + \bar{T}^{m}_{b} + \bar{T}^{F}_{b} \\ \bar{T}^{r}_{b} - \bar{T}^{m}_{b} - \bar{T}^{F}_{b} \end{bmatrix} = 0 \tag{4.13}$$

These equations can also be written as chord transformation equations. The main purpose of the chord transformation equations is to replace the branch through variables using the chord through variables.

For this system, the set of rotational chord transformation equations for the system graph and the sensitivity graph are given in equation (4.14).

$$\begin{bmatrix} \bar{T}^{h} \\ \bar{T}^{h}_{b} \\ \bar{T}^{r} \\ \bar{T}^{r}_{b} \end{bmatrix} = \begin{bmatrix} -\bar{T}^{m} - \bar{T}^{F} \\ -\bar{T}^{m}_{b} - \bar{T}^{F}_{b} \\ \bar{T}^{m} + \bar{T}^{F} \\ \bar{T}^{m}_{b} + \bar{T}^{F}_{b} \end{bmatrix} \tag{4.14}$$

Translational circuit equations

For this system, the translational circuit equations and the translational sensitivity circuit equations can be directly evaluated from their respective graphs and are given in equations (4.15) and (4.16), respectively. The translational circuit equations correspond to the topological loop equations present in the system.

$$\begin{bmatrix} -\bar{r}^{h} + \bar{r}^{r} + \bar{r}^{m} \\ -\bar{r}^{h} + \bar{r}^{r} + \bar{r}^{F} \end{bmatrix} = 0 \tag{4.15}$$

$$\begin{bmatrix} -\bar{r}^{h}_{b} + \bar{r}^{r}_{b} + \bar{r}^{m}_{b} \\ -\bar{r}^{h}_{b} + \bar{r}^{r}_{b} + \bar{r}^{F}_{b} \end{bmatrix} = 0 \tag{4.16}$$

By rearranging the circuit equations, it is possible to write a set of branch transformation equations that can be used to replace the chord across variables with the branch across variables. These equations are essential for the branch-chord formulation. For this system, the combined set of branch transformation equations obtained from the system and the sensitivity graphs are given in equation (4.17).

$$\begin{bmatrix} \bar{r}^{m} \\ \bar{r}^{F} \\ \bar{r}^{m}_{b} \\ \bar{r}^{F}_{b} \end{bmatrix} = \begin{bmatrix} \bar{r}^{h} - \bar{r}^{r} \\ \bar{r}^{h} - \bar{r}^{r} \\ \bar{r}^{h}_{b} - \bar{r}^{r}_{b} \\ \bar{r}^{h}_{b} - \bar{r}^{r}_{b} \end{bmatrix} \tag{4.17}$$

The translational branch transformation equations can also be written in terms of the velocities or even accelerations of the system. These equations are obtained by differentiating equation (4.17) with respect to time. These equations are sometimes required to simplify generated equations.

$$\begin{bmatrix} \bar{v}^{m} \\ \bar{v}^{F} \\ \bar{v}^{m}_{b} \\ \bar{v}^{F}_{b} \end{bmatrix} = \begin{bmatrix} \bar{v}^{h} - \bar{v}^{r} \\ \bar{v}^{h} - \bar{v}^{r} \\ \bar{v}^{h}_{b} - \bar{v}^{r}_{b} \\ \bar{v}^{h}_{b} - \bar{v}^{r}_{b} \end{bmatrix} \tag{4.18}$$

Rotational circuit equations

As mentioned before, the rotational circuit equations are written in terms of the rotational across variables, which are in fact matrices. To capture the relationships between these rotational across variables, it is necessary to define the rotation matrix **R** and establish the direction of the transformation it represents.

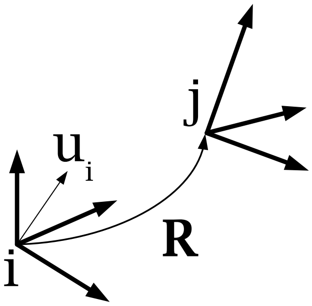


Figure 4.2: Rotation matrix from frame i to frame j

Figure 4.2 shows the schematic representation of two distinct frames of reference i and j, where **R** is the transformation matrix from frame i to frame j.

A vector in frame i can be represented in frame j using the following relationship.

$$\{\mathbf{u}_{\mathrm{j}}\} = \mathbf{R}_{\mathrm{i}\to\mathrm{j}}\,\{\mathbf{u}_{\mathrm{i}}\} \tag{4.19}$$

The rotational circuit equations for the pendulum are derived using the convention described in equation (4.19). The linear graph shown in figure 4.1 shows the presence of two kinematic loops. Hence, a total of two rotational circuit equations are obtained for the system, as shown in equation (4.20).

$$\begin{aligned} \left(\mathbf{R}^h\right)^T \mathbf{R}^r\,\mathbf{R}^m &= \mathbf{U}^{3\times 3} \\ \left(\mathbf{R}^h\right)^T \mathbf{R}^r\,\mathbf{R}^F &= \mathbf{U}^{3\times 3} \end{aligned} \tag{4.20}$$

As mentioned in the preceding section, since the rotation matrices deal with finite rotations, the order of multiplication of these matrices are important. Also, to obtain the correct equations for the rotational sensitivity across variables, one must use the chain rule of differentiation. By differentiating equation (4.20) with respect to the model parameter b, the rotational sensitivity circuit equations are obtained, as shown in equation (4.21).

$$\begin{aligned} \left(\mathbf{R}^h_b\right)^T \mathbf{R}^r\,\mathbf{R}^m + \left(\mathbf{R}^h\right)^T \mathbf{R}^r_b\,\mathbf{R}^m + \left(\mathbf{R}^h\right)^T \mathbf{R}^r\,\mathbf{R}^m_b &= 0 \\ \left(\mathbf{R}^h_b\right)^T \mathbf{R}^r\,\mathbf{R}^F + \left(\mathbf{R}^h\right)^T \mathbf{R}^r_b\,\mathbf{R}^F + \left(\mathbf{R}^h\right)^T \mathbf{R}^r\,\mathbf{R}^F_b &= 0 \end{aligned} \tag{4.21}$$

For the branch chord formulation, it is required to express the chord across variables, i.e., $\mathbf{R}^m, \mathbf{R}^F, \mathbf{R}^m_b$ and $\mathbf{R}^m_b$, in terms of the branch across variables, i.e., $\mathbf{R}^h, \mathbf{R}^r$ and their sensitivities. From the system graph or equation (4.20), the first two branch transformation equations can be obtained as

$$\begin{aligned} \mathbf{R}^m &= (\mathbf{R}^r)^T \mathbf{R}^h \\ \mathbf{R}^F &= (\mathbf{R}^r)^T \mathbf{R}^h \end{aligned} \tag{4.22}$$

From equation (4.21), one can derive the expressions for the rotational sensitivity branch transformation equations, i.e., the equations that express $\mathbf{R}^m_b$ and $\mathbf{R}^m_b$ in terms of the rotational branch across variables:

$$\begin{aligned} \mathbf{R}^m_b &= -(\mathbf{R}^r)^T \mathbf{R}^h \left(\left(\mathbf{R}^h_b\right)^T \mathbf{R}^r \mathbf{R}^m + \left(\mathbf{R}^h\right)^T \mathbf{R}^r_b \mathbf{R}^m \right) \\ \mathbf{R}^F_b &= -(\mathbf{R}^r)^T \mathbf{R}^h \left(\left(\mathbf{R}^h_b\right)^T \mathbf{R}^r \mathbf{R}^F + \left(\mathbf{R}^h\right)^T \mathbf{R}^r_b \mathbf{R}^F \right) \end{aligned} \tag{4.23}$$

Using properties of the rotation matrix, equation (4.23) can be simplified considerably.

The reduced equation is illustrated in equation (4.24). The formal proof of the simplification is given in Appendix B.3.

$$\begin{aligned}\mathbf{R}_b^m &= (\mathbf{R}_b^r)^T\mathbf{R}^h + (\mathbf{R}^r)^T\mathbf{R}_b^h \\ \mathbf{R}_b^F &= (\mathbf{R}_b^r)^T\mathbf{R}^h + (\mathbf{R}^r)^T\mathbf{R}_b^h\end{aligned} \tag{4.24}$$

Rotational circuit equations can also be written using the velocity and acceleration level variables. The key difference is the fact that the velocity and acceleration level rotational across variables are vectors and the corresponding equations are much simpler in form. For this system, the velocity level branch transformation equations are given in equation (4.25). Along with their translational counterparts, these equations are also used to simplify generated equations.

$$\begin{bmatrix} \bar{\omega}^m \\ \bar{\omega}^F \\ \bar{\omega}_b^m \\ \bar{\omega}_b^F \end{bmatrix} = \begin{bmatrix} \bar{\omega}^h - \bar{\omega}^r \\ \bar{\omega}^h - \bar{\omega}^r \\ \bar{\omega}_b^h - \bar{\omega}_b^r \\ \bar{\omega}_b^h - \bar{\omega}_b^r \end{bmatrix} \tag{4.25}$$

Terminal Equations

Terminal equations capture the nature of the component that is represented by a particular edge of a linear graph. For the generation of sensitivity equations, proper terminal equations must be specified for the edges from the system graph and the sensitivity graph simultaneously.

For the system graph, the terminal equations are specified to capture the physical nature of the components represented by the edges. On the other hand, the terminal equations for the sensitivity graph edges are derived by symbolically differentiating the terminal equations of the corresponding edges of the system graph. The terminal equations used for this system are explained below.

The rigid body element

For the edge denoted by m in figure 4.1, representing the rigid body, the terminal equations must be specified for both translational and rotational domains. For the translational domain, the quantity $\bar{r}^m$ is declared as a symbolic variable and the quantity $\bar{F}^m$ is specified as a function of the translational across variable as illustrated in equation (4.26).

$$\bar{r}^m = \left\{ \begin{array}{c} {}_x r^m(t) \\ {}_y r^m(t) \\ {}_z r^m(t) \end{array} \right\} \qquad \bar{F}^m = -m\frac{d^2}{dt^2}(\bar{r}^m) \tag{4.26}$$

The corresponding sensitivity equations are given in equation (4.27). These equations are obtained by symbolically differentiating the equations shown in equation (4.26) with respect to the model parameter b. The quantity $\bar{r}_b^m$ is a symbolic variable which is obtained by differentiating the vector $\bar{r}^m$ with respect to the model parameter b.

$$\bar{r}_b^m = \left\{ \begin{array}{c} {}_x r_b^m(t) \\ {}_y r_b^m(t) \\ {}_z r_b^m(t) \end{array} \right\} \qquad \bar{F}_b^m = -m\frac{d^2}{dt^2}\left(\bar{r}_b^m\right) - \left(\frac{\partial m}{\partial b}\right)\frac{d^2}{dt^2}\left(\bar{r}^m\right) \tag{4.27}$$

In equations (4.26) and (4.27) the subscripts to the right represent differentiation with respect to the symbols appearing as the subscript. The subscripts on the left represent the three components of the vector. Also, it is assumed that the vectors are expressed in the ground frame. The same convention is followed for all the subsequent equations illustrated in this chapter.

For the rotational domain the quantities that need to be defined are the rotation matrix $\mathbf{R}^m$, the torque $\bar{T}^m$ and their sensitivities, i.e., $\mathbf{R}_b^m$ and $\bar{T}_b^m$ respectively. For this example the $\mathbf{R}^m$ matrix is defined in terms of three Euler angles $\zeta - \eta - \xi$ assuming an intrinsic rotation scheme X - Y - Z. For this particular choice Euler angles, the expression for $\mathbf{R}^m$ is given by (4.7).

The dynamic torque $\bar{T}^m$, acting on a rigid body undergoing a general three-dimensional rotational motion, can be expressed in terms of the moment of inertia dyadic $\overline{\overline{J}}$, the time derivative of the angular velocity vector $\dot{\bar{\omega}}^m$, and the translational through and across variables $\bar{r}^r$ and $\bar{F}^r$, associated with the rigid arm elements connected to the rigid body. Equation (4.28) shows the general expression for the torque vector, where the symbols $\tilde{r}_r$ and $\tilde{\omega}^m$ refer to the skew symmetric forms of the vectors $\bar{r}_r$ and $\bar{\omega}^m$ respectively.

$$\bar{T}^m = -\overline{\overline{J}}\,\dot{\bar{\omega}}^m - \tilde{r}_r\bar{F}_r - \tilde{\omega}^m\,\overline{\overline{J}}\,\bar{\omega}^m \tag{4.28}$$

Since the motion of the pendulum is restricted to a single plane, the gyroscopic term can be safely neglected. When expressed in the frame fixed to the centre of mass of the rigid body, equation (4.28) can be rewritten in terms of the centroidal moment of inertia matrix $\mathbf{J}$ and the other vectors. Equation (4.29) shows the expression of vector $\bar{T}^m$.

$$\bar{T}^m = -\mathbf{J}\,\dot{\bar{\omega}}^m - \bar{r}_r \times \bar{F}_r \tag{4.29}$$

The centroidal moment of inertia matrix $\mathbf{J}$ can be written in terms of the mass of the rigid body M, the length L, and the diameter of cross-section D, as shown in equation (4.30).

$$\mathbf{J} = \begin{bmatrix} \frac{ML^2}{12} & 0 & 0 \\ 0 & \frac{MD^2}{4} & 0 \\ 0 & 0 & \frac{ML^2}{12} \end{bmatrix} \tag{4.30}$$

The sensitivity terminal equation can be obtained through direct differentiation of the specified terminal equation. The quantity $\bar{T}_b^m$ is given by the following expression.

$$\bar{T}_b^m = -\frac{d}{db}(\mathbf{J})\,\dot{\bar{\omega}}^m - \mathbf{J}\,\dot{\bar{\omega}}_b^m - \bar{r}_b^r \times \bar{F}^r - \bar{r}^r \times \bar{F}_b^r \tag{4.31}$$

In equation (4.31), the term $\bar{\omega}_b$ denotes the sensitivity of the term $\bar{\omega}$ with respect to model parameter b and the terms $\bar{r}^r$, $\bar{F}^r$ and their sensitivities $\bar{r}_b^r$ and $\bar{F}_b^r$ are defined as the terminal equations for the edge representing the rigid arm element of the system as described in the next section.

The sensitivity terminal equation for the rotational domain also includes the definition of the term $\mathbf{R}_b^m$. This expression is obtained by differentiating equation (4.7) with respect to model parameter b.

$$\mathbf{R}_b^m = \begin{bmatrix} \Psi_1 & \Psi_2 & \Psi_3 \\ \Psi_4 & \Psi_5 & \Psi_6 \\ \eta_b\,\mathrm{c}\,\eta & \eta_b\,\mathrm{s}\,\eta\,\mathrm{s}\,\zeta - \zeta_b\,\mathrm{c}\,\eta\,\mathrm{c}\,\zeta & -\eta_b\,\mathrm{s}\,\eta\,\mathrm{c}\,\zeta - \zeta_b\,\mathrm{c}\,\eta\,\mathrm{s}\,\zeta \end{bmatrix} \tag{4.32}$$
$$c \equiv \cos \quad \text{and} \quad s \equiv \sin$$

where

$$\begin{aligned}
\Psi_1 &= -\eta_b\,\mathrm{s}\,\xi\,\mathrm{c}\,\eta - \eta_b\,\mathrm{s}\,\eta\,\mathrm{c}\,\xi \\
\Psi_2 &= -\xi_b\,\mathrm{s}\,\xi\,\mathrm{s}\,\eta\,\mathrm{s}\,\zeta + \eta_b\,\mathrm{c}\,\xi\,\mathrm{c}\,\eta\,\mathrm{s}\,\zeta + \zeta_b\,\mathrm{c}\,\xi\,\mathrm{s}\,\eta\,\mathrm{c}\,\zeta + \xi_b\,\mathrm{c}\,\xi\,\mathrm{c}\,\zeta - \zeta_b\,\mathrm{s}\,\xi\,\mathrm{s}\,\zeta \\
\Psi_3 &= \xi_b\,\mathrm{s}\,\xi\,\mathrm{s}\,\eta\,\mathrm{c}\,\zeta - \eta_b\,\mathrm{c}\,\xi\,\mathrm{c}\,\eta\,\mathrm{c}\,\zeta + \zeta_b\,\mathrm{c}\,\xi\,\mathrm{s}\,\eta\,\mathrm{s}\,\zeta + \xi_b\,\mathrm{c}\,\xi\,\mathrm{s}\,\zeta + \zeta_b\,\mathrm{s}\,\xi\,\mathrm{c}\,\zeta \\
\Psi_4 &= -\xi_b\,\mathrm{c}\,\xi\,\mathrm{c}\,\eta + \eta_b\,\mathrm{s}\,\eta\,\mathrm{s}\,\xi \\
\Psi_5 &= -\xi_b\,\mathrm{c}\,\xi\,\mathrm{s}\,\eta\,\mathrm{s}\,\zeta - \eta_b\,\mathrm{s}\,\xi\,\mathrm{c}\,\eta\,\mathrm{s}\,\zeta - \zeta_b\,\mathrm{s}\,\xi\,\mathrm{s}\,\eta\,\mathrm{c}\,\zeta - \xi_b\,\mathrm{s}\,\xi\,\mathrm{c}\,\zeta - \zeta_b\,\mathrm{c}\,\xi\,\mathrm{s}\,\zeta \\
\Psi_6 &= \xi_b\,\mathrm{c}\,\xi\,\mathrm{s}\,\eta\,\mathrm{c}\,\zeta + \eta_b\,\mathrm{s}\,\xi\,\mathrm{c}\,\eta\,\mathrm{c}\,\zeta - \zeta_b\,\mathrm{s}\,\xi\,\mathrm{s}\,\eta\,\mathrm{s}\,\zeta - \xi_b\,\mathrm{s}\,\xi\,\mathrm{s}\,\zeta + \zeta_b\,\mathrm{c}\,\xi\,\mathrm{c}\,\zeta
\end{aligned} \tag{4.33}$$

The matrix $\mathbf{R}_b^m$ can not be considered as a rotation matrix by itself, since it is not an orthogonal matrix and its determinant is not equal to unity. However, it is required to construct the sensitivity terminal equations and sensitivity circuit equations for the system.

At this point, it is worthwhile to point out that although the expressions of these terminal equations are somewhat complicated, their forms are independent of the topology of the system. Thus, it is possible to hard code these equations into a software implementation where the case-specific equations can be generated on the fly during the simulation. At that point, the problem reduces to evaluating symbolic differentiation of some simple expressions.

The rigid arm element

The rigid arm element, i.e., the edge denoted with r, represents the geometric transformation from the frame fixed to the centre of mass of the body m to the frame fixed to the point on the body where the revolute joint is attached. As explained previously, the terminal equations for the rigid arm element must be specified for both rotational and translational domains. On top of this, for the generation of sensitivity equations, it is also required to specify the sensitivity terminal equations for these domains. For the translational domain, the terminal equations for the system equations can be declared as illustrated below.

For the edge representing the rigid arm element, the translational through variable $\bar{F}^r$ and the corresponding sensitivity through variable $\bar{F}^r_b$ are defined as symbolic variables which are vectors.

$$\bar{F}^r = \left\{ \begin{array}{c} {}_xF^r(t) \\ {}_yF^r(t) \\ {}_zF^r(t) \end{array} \right\} \quad \text{and} \quad \bar{F}^r_b = \left\{ \begin{array}{c} {}_xF^r_b(t) \\ {}_yF^r_b(t) \\ {}_zF^r_b(t) \end{array} \right\} \tag{4.34}$$

The simplest way to specify the rigid arm translational across variable $\bar{r}^r$ is to define the displacement vector in the frame fixed to the centre of mass of the body it is fixed with.

$$\bar{r}_0 = \{\bar{r}^r\}_{c.o.m} = \left\{ \begin{array}{ccc} 0 & L/2 & 0 \end{array} \right\}^T \tag{4.35}$$

To specify the constitutive equations, the body fixed vector $\bar{r}_0$ is expressed in the ground frame using a rotational transformation matrix $\mathbf{R}^m$. The matrix $\mathbf{R}^m$ corresponds to a rotational transformation from the frame fixed to the centre of mass of the rigid body to the ground frame and is shown in equation (4.7).

The resulting quantity $\bar{r}^r$ becomes a time varying vector and contains the Euler angles and their time derivatives to include the effects of the rotations undergone by the body.

$$\begin{gathered} \bar{r}^r = \mathbf{R}^m \bar{r}_0 \\ \dot{\bar{r}}^r = \bar{v}^r = \bar{\omega}^m \times \bar{r}^r \quad \text{and} \quad \ddot{\bar{r}}^r = \bar{a}^r = \dot{\bar{\omega}}^m \times \bar{r}^r - (|\bar{\omega}^m|)^2 \bar{r}^r \end{gathered} \tag{4.36}$$

The corresponding sensitivity terminal equations are

$$\begin{aligned} \bar{r}^{\,r}_b &= \mathbf{R}^m \frac{d}{db}\left(\bar{r}_0\right) + \mathbf{R}^m_b\,\bar{r}_0 \\ \dot{\bar{r}}^{\,r}_b &= \bar{\omega}^m_b \times \bar{r}^{\,r} + \bar{\omega}^m \times \bar{r}^{\,r}_b \\ \ddot{\bar{r}}^{\,r}_b &= \dot{\bar{\omega}}^m_b \times \bar{r}^{\,r} + \dot{\bar{\omega}}^m \times \bar{r}^{\,r}_b - 2\left(\bar{\omega}^m \cdot \bar{\omega}^m_b\right)\bar{r}^{\,r} - (|\bar{\omega}^m|)^2 \bar{r}^{\,r}_b \end{aligned} \tag{4.37}$$

For the rotational domain, the terminal equation for the matrix $\mathbf{R}^r$ captures the relative orientation of the frame of reference fixed to the hinge point and the frame of reference fixed to the centre of mass of the pendulum. For this system these frames are assumed to be aligned with each other, which makes the quantity $\mathbf{R}^r$ a constant identity matrix and the corresponding sensitivity $\mathbf{R}^r_b$ a zero matrix.

The revolute joint

The edge marked with h represents the revolute joint in this system. This joint allows rotational motion about the Z axis and the system and sensitivity terminal equations are defined to capture this functionality.

Translational system terminal equations

$$\bar{r}^{\,h} = \left\{\begin{matrix} 0 \\ 0 \\ 0 \end{matrix}\right\} \quad \text{and} \quad \bar{F}^h = \left\{\begin{matrix} {}_xF^h \\ {}_yF^h \\ {}_zF^h \end{matrix}\right\} \tag{4.38}$$

Translational sensitivity terminal equations

$$\bar{r}^{\,h}_b = \left\{\begin{matrix} 0 \\ 0 \\ 0 \end{matrix}\right\} \quad \text{and} \quad \bar{F}^h_b = \left\{\begin{matrix} {}_xF^h_b \\ {}_yF^h_b \\ {}_zF^h_b \end{matrix}\right\} \tag{4.39}$$

Rotational system terminal equations

$$\mathbf{R}^h = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \bar{\tau}^{\,h} = \left\{\begin{matrix} {}_x\tau^h \\ {}_y\tau^h \\ 0 \end{matrix}\right\} \tag{4.40}$$

It is also possible to express the angular velocity of the component in terms of the position level variable. For the revolute joint, the angular velocity variable is denoted as $\bar{\omega}^h$ and can be expressed in terms of $\mathbf{R}^h$ and α as shown in equation (4.41).

$$\bar{\omega}^h = \left\{ \begin{array}{ccc} 0 & 0 & \dot{\alpha}(t) \end{array} \right\}^T \tag{4.41}$$

Rotational sensitivity terminal equations

$$\mathbf{R}_b^h = \begin{bmatrix} -\alpha_b \, \mathrm{s}\, \alpha & \alpha_b \, \mathrm{c}\, \alpha & 0 \\ -\alpha_b \, \mathrm{c}\, \alpha & -\alpha_b \, \mathrm{s}\, \alpha & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \bar{\tau}_b^h = \left\{ \begin{array}{c} {}_x\tau_b^h \\ {}_y\tau_b^h \\ 0 \end{array} \right\} \tag{4.42}$$

The external force element

The edge marked with F represents the body force acting on the pendulum due to its mass. In terms of linear graph theory, this is known as a through driver. The terminal equations for this edge shown in equations (4.43) and (4.44).

For the translational domain

$$\bar{F}^F = \left\{ \begin{array}{c} 0 \\ -mg \\ 0 \end{array} \right\} \quad \text{and} \quad \bar{F}_b^F = \left\{ \begin{array}{c} 0 \\ -\frac{d}{db}(mg) \\ 0 \end{array} \right\} \tag{4.43}$$

For the rotational domain

$$\tau^F = \{0\}^{3\times 3} \quad \text{and} \quad \tau_b^F = \{0\}^{3\times 3} \tag{4.44}$$

It is very important to make sure that the vector quantities are expressed in the same frame of reference before any of the algebraic operations are carried out. For the vectors specified in specific frames, proper frame transformation must be ensured before they can be used in any of the other terminal equations specified.

Generation of system and sensitivity equations

To demonstrate the process of generation of the governing equations for the system and the sensitivity equation, we reiterate the choice of the tree branches as mentioned before. For this system we choose the edges corresponding to the revolute joint and the rigid arm as the tree branches as illustrated in figure 4.3 using solid lines. The edges corresponding to the rigid body m and body force F are classified as the cotree chords. Also the length of the pendulum L is chosen as the model parameter with respect to which the sensitivity analysis is being carried out.

As presented in chapter 2, the next step is the projection of cutset equations and the sensitivity cutset equations of all the rigid bodies, passive joints and through drivers on to their respective motion spaces. This ensures the elimination of the reaction forces and their sensitivities that correspond to the passive kinematic constraints included in the tree. Since this system involves both rotational and translational domains, cutset equations from both domains must be projected onto their respective motion spaces.

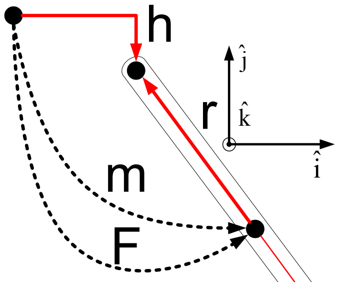


Figure 4.3: Tree selection for joint coordinate formulation

At this point, the concept of projection for the sensitivity equations must be clarified. Normally, the projection process involves the evaluation of the dot product of two vectors. However, for the sensitivity equations, this process must be modified to capture the chain rule of differentiation.

For a general system, the motion spaces of the joints can be implicit functions of the model parameters. In these cases, the form shown in equation (4.45) must be followed for accurate derivation of sensitivity equations. It is to be noted that in equation (4.45), all the quantities on the right hand side are vectors and care must be taken to express them in a common frame before the dot products are evaluated.

$$\begin{bmatrix}\text{Projected}\\ \text{equation}\end{bmatrix} = \begin{bmatrix}\text{Sensitivity}\\ \text{cutset equations}\end{bmatrix} \cdot \begin{bmatrix}\text{Motion}\\ \text{space}\end{bmatrix} + \begin{bmatrix}\text{Cutset}\\ \text{equations}\end{bmatrix} \cdot \begin{bmatrix}\text{Sensitivity of}\\ \text{motion space}\end{bmatrix} \tag{4.45}$$

For the translational domain the cutset equations of the only eligible component for the first step, i.e., the revolute joint are shown in equation (4.46).

$$\begin{aligned}\bar{F}^h + \bar{F}^m + \bar{F}^F &= 0\\ \bar{F}^h_L + \bar{F}^m_L + \bar{F}^F_L &= 0\end{aligned} \tag{4.46}$$

However, the translational motion space is null because the revolute joint does not allow movement in any direction. For this reason, none of these equations get projected and appear in the system and/or sensitivity equations.

For the rotational domain, the system and sensitivity cutset equations can be projected onto the motion space which is the axis of rotation of the revolute joint. For this system the motion space is the $\hat{k}$ axis as shown in figure 4.3.

Since the motion space for this particular joint is a constant, the second part of equation (4.45) can be neglected and the projected equations can be evaluated as shown in equation (4.47).

$$\begin{aligned} \left(\bar{T}^h + \bar{T}^m + \bar{T}^F\right) . \hat{k} = 0 \\ \left(\bar{T}^h_L + \bar{T}^m_L + \bar{T}^F_L\right) . \hat{k} = 0 \end{aligned} \tag{4.47}$$

Equation (4.47) shows the projected cutset equations. Substituting the terminal equations the following equations are obtained.

$$\begin{aligned} &\left(-\mathbf{J}\dot{\bar{\omega}}^m - \bar{r}_r \times \bar{F}_r\right) . \hat{k} = 0 \\ &\left(-\mathbf{J}_L\,\dot{\bar{\omega}}^m - \mathbf{J}\dot{\bar{\omega}}^m_L - \bar{r}^r_L \times \bar{F}^r - \bar{r}^r \times \bar{F}^r_L\right) . \hat{k} = 0 \end{aligned} \tag{4.48}$$

In equation 4.48 the term $\mathbf{J}_L$ represents the derivative of the inertia matrix with respect to the model parameter L and can be derived from equation 4.30.

$$\mathbf{J}_L = \begin{bmatrix} \frac{mL}{6} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{mL}{6} \end{bmatrix} \tag{4.49}$$

The next step is the elimination of secondary variables. As explained in chapter 2, it is done in a specific order. First the rigid arm forces are eliminated by substituting the translational branch transformation equations (4.11) into equation (4.48). After substitution the equations are formed in terms of the tree across variables, cotree through variables, and the cotree across variables.

$$\begin{aligned} \left[-\mathbf{J}\dot{\bar{\omega}}^m - \bar{r}^r \times \left(\bar{F}^m + \bar{F}^F\right)\right] . \hat{k} = 0 \\ \left[-\mathbf{J}_L\,\dot{\bar{\omega}}^m - \mathbf{J}\dot{\bar{\omega}}^m_L - \bar{r}^r_L \times \left(\bar{F}^m + \bar{F}^F\right) - \bar{r}^r \times \left(\bar{F}^m_L + \bar{F}^F_L\right)\right] . \hat{k} = 0 \end{aligned} \tag{4.50}$$

The next step is to replace the cotree translational across variables. First the terminal equations are substituted in (4.50) and then (4.17) is used to replace the translational across variables from the chords with the translational across variables of the branches. Equation (4.51) shows the first equation after the substitution.

$$\left(-\mathbf{J}\dot{\bar{\omega}}^{m} - \bar{r}^{r} \times \left(-m\left(\ddot{\bar{r}}^{h} - \ddot{\bar{r}}^{r}\right) + \bar{F}^{F}\right)\right) \cdot \hat{k} = 0 \tag{4.51}$$

Since, $\bar{r}^{h} = 0$, this equation can be simplified further as illustrated below.

$$\begin{aligned} \left(-\mathbf{J}\dot{\bar{\omega}}^{m} - \bar{r}^{r} \times \left(m\left(\dot{\bar{\omega}}^{m} \times \bar{r}^{r} - (|\bar{\omega}^{m}|)^{2}\bar{r}^{r}\right) + \bar{F}^{F}\right)\right) \cdot \hat{k} = 0 \\ \Rightarrow \left(-\mathbf{J}\dot{\bar{\omega}}^{m} - m\left(\bar{r}^{r} \times \dot{\bar{\omega}}^{m} \times \bar{r}^{r}\right) - \bar{r}^{r} \times \bar{F}^{F}\right) \cdot \hat{k} = 0 \\ \Rightarrow \left(-\mathbf{J}\dot{\bar{\omega}}^{m} - m\left(\mathbf{R}^{m}\bar{r}_{0} \times \dot{\bar{\omega}}^{m} \times \mathbf{R}^{m}\bar{r}_{0}\right) - \mathbf{R}^{m}\bar{r}_{0} \times \bar{F}^{F}\right) \cdot \hat{k} = 0 \end{aligned} \tag{4.52}$$

At this stage, the only undesired component in this equation is the cotree rotational across variables $\mathbf{R}^{m}$ and $\bar{\omega}^{m}$. They can be replaced by the rotational across variables of the branches using equation (4.22) and the terminal equations. By substituting the equations into equation (4.52) the following equation is obtained.

$$\left(-\mathbf{J}\dot{\bar{\omega}}^{h} - m\left(\mathbf{R}^{h}\bar{r}_{0} \times \dot{\bar{\omega}}^{h} \times \mathbf{R}^{h}\bar{r}_{0}\right) - \mathbf{R}^{h}\bar{r}_{0} \times \bar{F}^{F}\right) \cdot \hat{k} = 0 \tag{4.53}$$

Equation (4.53) can be simplified by using the expression for $\bar{r}_{0} = \{0, L/2, 0\}^{T}$ and the expression for $\mathbf{R}^{h}$ from equation (4.40), which gives rise to the following equation.

$$\mathbf{R}^{h}\bar{r}_{0} = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ L/2 \\ 0 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} L\sin\alpha \\ L\cos\alpha \\ 0 \end{bmatrix} \tag{4.54}$$

Also using equation (4.30) and (4.41) the following equation can be derived.

$$\mathbf{J}\dot{\bar{\omega}}^{h} = \frac{1}{12}\begin{Bmatrix} 0 \\ 0 \\ mL^{2}\ddot{\alpha} \end{Bmatrix} \tag{4.55}$$

Substituting equations (4.54) and (4.55) into (4.53), a much simpler form is obtained.

$$\left(-\frac{1}{12}\begin{bmatrix} 0 \\ 0 \\ mL^{2}\ddot{\alpha}\left(t\right) \end{bmatrix} - \frac{1}{4}\begin{bmatrix} 0 \\ 0 \\ mL^{2}\ddot{\alpha}\left(t\right) \end{bmatrix} - \mathbf{R}^{h}\bar{r}_{0} \times \bar{F}^{F}\right) \cdot \hat{k} = 0 \tag{4.56}$$

After the projection onto the $\hat{k}$ axis, equation (4.56) transforms into the regular expression for the governing equation of a simple pendulum.

$$\frac{mL^2}{3}\ddot{\alpha} + \frac{mgL}{2}\sin\alpha = 0 \tag{4.57}$$

Due to the added sensitivity terms, the derivation of the sensitivity equation is a little more involved than the example shown so far. However, it can be generated using the same structured algorithm. The process is illustrated by the example below.

After substituting the terminal equations into the second equation from (4.50) the following equation is obtained.

$$\left[-\mathbf{J}_L\dot{\bar{\omega}}^m - \mathbf{J}\dot{\bar{\omega}}^m_L + m\left(\bar{r}^r_L \times \ddot{\bar{r}}^m\right) - \bar{r}^r_L \times \bar{F}^F + m\left(\bar{r}^r \times \ddot{\bar{r}}^m_L\right)\right] \cdot \hat{k} = 0 \tag{4.58}$$

By using translational branch transformation equations (4.17) the translational chord across variables are replaced by translational branch across variables. In other words, $\bar{r}^m$ is replaced by $\bar{r}^h$ and $\bar{r}^r$. Using terminal equations for the newly formed variables, equation (4.58) can be modified to the equation shown below.

$$\left[-\mathbf{J}_L\dot{\bar{\omega}}^m - \mathbf{J}\dot{\bar{\omega}}^m_L - m\left(\bar{r}^r_L \times \ddot{\bar{r}}^r\right) - \bar{r}^r_L \times \bar{F}^F - m\left(\bar{r}^r \times \ddot{\bar{r}}^r_L\right)\right] \cdot \hat{k} = 0 \tag{4.59}$$

Using equations (4.36) and (4.37), the terms $\ddot{\bar{r}}^r$ and $\ddot{\bar{r}}^r_L$ can be replaced by rotational across variables from the chords. Rotational branch transformation equations (4.22) and (4.25) can then be used to replace the chord across variables with the branch across variables or the primary variables. Using this technique on equation (4.59) the following equation is obtained.

$$\left[\bar{\sigma}_0 - m\left(\bar{\sigma}_1 \times \bar{\sigma}_2\right) - \bar{\sigma}_4 \times \left(\dot{\bar{\omega}}^h \times \bar{\sigma}_1\right) - \bar{\sigma}_4 \times \bar{\sigma}_3 - \bar{\sigma}_1 \times \bar{F}^F\right] \cdot \hat{k} = 0 \tag{4.60}$$

where

$$\begin{gathered}\bar{\sigma}_0 = -\mathbf{J}_L\dot{\bar{\omega}}^h - \mathbf{J}\dot{\bar{\omega}}^h_L \quad \bar{\sigma}_1 = \left(\mathbf{R}^h(\bar{r}_0)_L + \mathbf{R}^h_L\bar{r}_0\right) \quad \bar{\sigma}_2 = \dot{\bar{\omega}}^h \times \mathbf{R}^h\bar{r}_0 \\ \bar{\sigma}_3 = \dot{\bar{\omega}}^h_L \times \mathbf{R}^h\bar{r}_0 \qquad\qquad \bar{\sigma}_4 = m\mathbf{R}^h\bar{r}_0\end{gathered} \tag{4.61}$$

At this point, the equations are expressed in terms of the primary variables only and thus can be considered as complete. To obtain the usual form of the sensitivity equation, the terminal equations must be substituted into equation (4.61) and the resulting equation must be projected onto the $\hat{k}$ axis as shown in the equation. The following expressions show the evaluation of the various terms of the equation (4.61).

$$\bar{\sigma}_0 = -\frac{1}{6}\mathbf{J}_L \begin{bmatrix} 0 \\ 0 \\ \ddot{\alpha} \end{bmatrix} - \mathbf{J} \begin{bmatrix} 0 \\ 0 \\ \ddot{\alpha}_L \end{bmatrix} = \frac{1}{12} \begin{bmatrix} 0 \\ 0 \\ -2mL\ddot{\alpha} - mL^2\ddot{\alpha}_L \end{bmatrix} \tag{4.62}$$

$$\bar{\sigma}_1 = \left(\mathbf{R}^h (\bar{r}_0)_L + \mathbf{R}^h_L \bar{r}_0\right) = \frac{1}{2} \begin{bmatrix} \sin\alpha + L\,\alpha_L \cos\alpha \\ \cos\alpha - L\,\alpha_L \sin\alpha \\ 0 \end{bmatrix} \tag{4.63}$$

$$\bar{\sigma}_2 = \dot{\bar{\omega}}^h \times \mathbf{R}^h\,\bar{r}_0 = \frac{1}{2} \begin{bmatrix} -L\,\ddot{\alpha}\cos\alpha \\ L\,\ddot{\alpha}\sin\alpha \\ 0 \end{bmatrix} \tag{4.64}$$

$$\bar{\sigma}_3 = \dot{\bar{\omega}}^h_L \times \mathbf{R}^h\,\bar{r}_0 = \frac{1}{2} \begin{bmatrix} -L\,\ddot{\alpha}_L\cos\alpha \\ L\,\ddot{\alpha}_L\sin\alpha \\ 0 \end{bmatrix} \tag{4.65}$$

$$\bar{\sigma}_4 = m\mathbf{R}^h\,\bar{r}_0 = \frac{1}{2} \begin{bmatrix} mL\sin\alpha \\ mL\cos\alpha \\ 0 \end{bmatrix} \tag{4.66}$$

By substituting equations (4.62) - (4.66) into equation (4.61) a much simpler form is obtained which can then be projected on to the $\hat{k}$ axis for the final equation.

$$\left(\frac{mL}{12}\begin{bmatrix} 0 \\ 0 \\ \Lambda \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \frac{mL(2\ddot{\alpha}+L\ddot{\alpha}_L)}{4} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \frac{mg(\mathrm{s}\,\alpha+\alpha_L L\,\mathrm{c}\,\alpha)}{2} \end{bmatrix}\right) \cdot \hat{k} = 0 \tag{4.67}$$
$$\Lambda = -2\ddot{\alpha} - L\ddot{\alpha}_L$$

After projecting this equation the final form of the sensitivity equation is obtained as shown in equation (4.68). It can be clearly demonstrated that the same equation can be obtained by differentiating equation (4.57) manually.

$$\frac{2mL}{3}\ddot{\alpha} + \frac{mL^2}{3}\ddot{\alpha}_L + \frac{mg}{2}\sin\alpha + \frac{mgL}{2}\alpha_L\cos\alpha = 0 \tag{4.68}$$

4.2.4 Effect of branch selection

As discussed in the preceding sections, the size, structure, and complexity of the generated equations depend on the selection of the branches for the branch-chord formulation. The proposed algorithm for the generation of sensitivity equations allows the user to select different trees for the governing and sensitivity equations. This offers superior control over the structure of the final set of equations, as explained here.

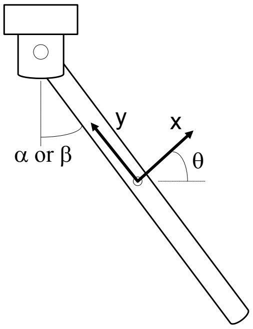


Figure 4.4: Absolute and joint coordinate system for a pendulum

To demonstrate the effect of branch selection on the generated sensitivity equations, the example of the simple pendulum is considered as shown in figure 4.4.

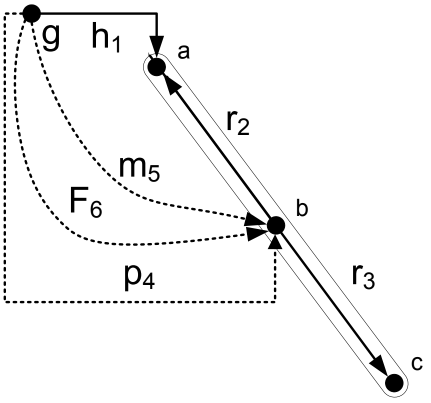


Figure 4.5: Tree 1

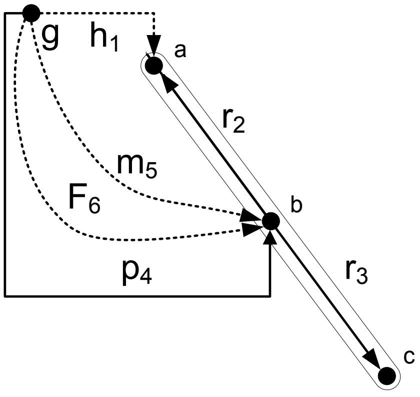


Figure 4.6: Tree 2

The pendulum shown in the figure 4.4 can be modelled in terms of the joint angle α, or in terms of the absolute coordinates x, y, and θ which are associated with the frame at the centroid of the pendulum. These separate equations can be generated by selecting different trees for the graph-theoretic formulation for the system. To demonstrate the process, the pendulum is represented using a linear graph as shown in the figures 4.5 and 4.6.

The relevant across variables are assigned in such a way so that if the revolute joint is selected in the tree, the equations are generated in terms of the quantity α, vide figure 4.4, and if the planar joint is selected as a branch, the equations are generated in terms of the quantities x, y, and θ, vide figure 4.4.

If the branch selection scheme shown in figure 4.5 is used to generate both system and sensitivity equations for the simple pendulum, the equations will be generated in terms of the angle α and the sensitivity of the same α_L. The derivation of these equations is presented in detail previously. Equation (4.69) presents these equations in a matrix form.

$$\begin{bmatrix} \frac{mL^2}{3} & 0 \\ \frac{2mL}{3} & \frac{mL^2}{3} \end{bmatrix} \begin{bmatrix} \ddot{\alpha} \\ \ddot{\alpha}_L \end{bmatrix} = \begin{bmatrix} -\frac{mgL}{2}\sin\alpha \\ -\frac{mg}{2}\sin\alpha - \frac{mgL}{2}\alpha_L\cos\alpha \end{bmatrix} \tag{4.69}$$

If the branch selection scheme shown in figure 4.6 is selected for both system and sensitivity equations, the final set of equations are formed in terms of x, y, θ, x_L, y_L, θ_L.

However, if the scheme shown in figure 4.6 is used for the system equations and the scheme shown in figure 4.5 is used for the sensitivity equations, the resulting set of equations are formed in terms of x, y, θ, and the rotational across variable of the revolute joint edge in the sensitivity graph, β_L. The formed equations are shown in equation (4.70):

$$\begin{bmatrix} m & 0 & 0 & 0 \\ 0 & m & 0 & 0 \\ 0 & 0 & \frac{mL^2}{12} & 0 \\ 0 & 0 & \frac{5mL}{12} & \frac{mL^2}{3} \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \\ \ddot{\beta}_L \end{bmatrix} + \mathbf{\Phi}_{\mathbf{q}}^T \begin{bmatrix} {}_xF^h \\ {}_yF^h \end{bmatrix} = \begin{bmatrix} 0 \\ -mg \\ 0 \\ \frac{mL^2\beta_L\dot{\theta}^2}{4} \end{bmatrix} \quad \text{and} \quad \mathbf{\Phi} = 0 \tag{4.70}$$

where ${}_xF^h$ and ${}_yF^h$ are the joint reaction forces in the global X and Y directions, and the matrix $\mathbf{\Phi}_{\mathbf{q}}^T$ is the transpose of the Jacobian matrix.

$$\mathbf{\Phi} = \begin{bmatrix} x - \frac{L\sin\theta}{2} \\ y + \frac{L\cos\theta}{2} \end{bmatrix} = 0 \quad \mathbf{\Phi}_{\mathbf{q}} = \begin{bmatrix} -1 & 0 & \frac{L\cos\theta}{2} & \frac{\cos\theta - L\beta_L\sin\theta}{2} \\ 0 & -1 & \frac{L\sin\theta}{2} & \frac{\sin\theta + L\beta_L\cos\theta}{2} \end{bmatrix} \tag{4.71}$$

Equation (4.70) is unique in the sense that it is a collection of equations which uses absolute coordinates for the governing equations, but joint coordinates for the sensitivity equations; generated using an automated and structured algorithm.

The solution of β and α are identical to each other as they represent the same physical angle. Consequently, the solution of α_L and β_L are also identical. To demonstrate that the equations (4.70) and (4.69) solve for the same sensitivity and yield same results, the equations are solved by using the following values for the model parameters.

$$m = 1 \quad g = 9.8 \quad L = 1 \tag{4.72}$$

Also the following initial conditions are used to ensure that the equations solve for the same physical systems with identical initial configurations.

$$\begin{bmatrix} \alpha(0) = 30^\circ & \dot{\alpha}(0) = 0 \\ \alpha_L(0) = 0 & \dot{\alpha}_L(0) = 0 \end{bmatrix} \text{ and } \begin{bmatrix} \theta(0) = 30^\circ & \dot{\theta}(0) = 0 \\ \beta_L(0) = 0 & \dot{\beta}_L(0) = 0 \end{bmatrix} \tag{4.73}$$

At this point, it is important to take note of the fact that the equations presented in (4.70) is a set of DAEs and to ensure successful simulation, consistent initial conditions must be specified for all the state variables.

Since equation (4.70) consists of two algebraic constraint equations and has four state variables, only two of the state variables can be specified arbitrarily. For this simulation, initial conditions are specified for θ , β_L and their derivatives to ensure that the initial configuration matches the one used by the other simulation.

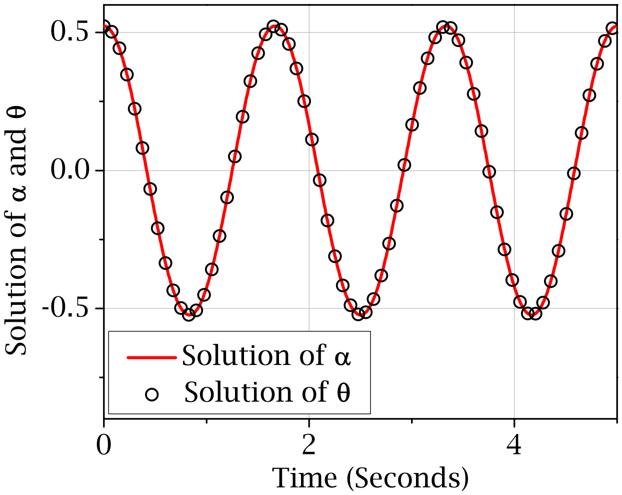


Figure 4.7: Solutions of α and θ

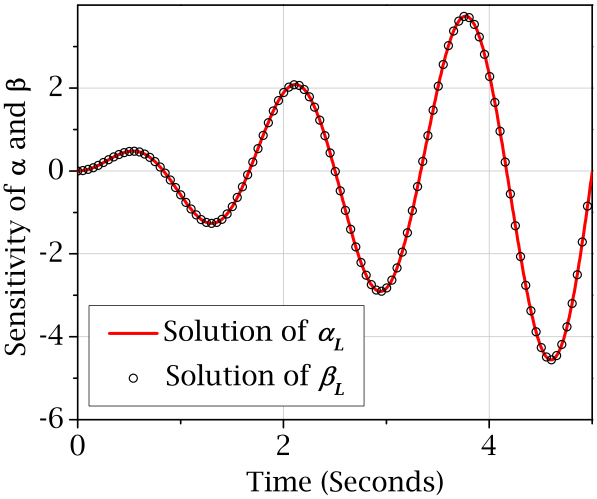


Figure 4.8: Solutions of α_L and β_L

Figure 4.7 demonstrates complete congruity between the quantities α and θ, confirming the statement that the two simulations represent the same physical system and its configuration. Figure 4.8 shows that the solution of β_L matches that of α_L and the two quantities are physically identical. This validates the ability of the proposed algorithm to generate sensitivity equations in terms of sensitivity quantities other than the sensitivities of the state variables of the governing equations.

This is an unique and very helpful feature of the proposed algorithm. In many practical cases, the most efficient models do not necessarily consist of the state variables of which sensitivity information is required.

Under those circumstances, the proposed algorithm can be used to generate a set of sensitivity equations that are dependent only on the required sensitivity quantities by selecting the corresponding edges as the branches of the sensitivity tree. This avoids the simulation of equations consisting of unnecessary sensitivity variables and makes the process more efficient.

At this point, it is necessary to mention that if the trees selected for the sensitivity and the system equations are different, the resulting equations are generated in terms of different sets of through and across variables. The relationships between the primary variables of the sensitivity tree and the primary variables of the system tree can be determined by using the chord and branch transformation equations.

For the example shown above, the primary variable for the sensitivity equation just happens to be equal to one of the primary variables used in the system equation. For a more general system, it is not reasonable to expect such equality among the two sets of variables. For those systems the generated branch and chord transformations must be used to bridge the gap between the two sets of variables.

4.3 Chapter Summary

The process of automated generation of sensitivity equations for multibody systems from its linear graph is demonstrated in this chapter. A brief theoretical description is used to demonstrate the salient aspects of graph-theoretic representation and subsequent analysis of multibody systems. In light of this description, an algorithm for graph-theoretic sensitivity analysis is formulated for multibody systems. An effort is made to interpret the physical implications of the derivative of rotation matrices which are used frequently to represent the three-dimensional orientation of rigid bodies.

Using a simple pendulum as an example, the process of generation of governing equations and sensitivity equations are demonstrated using full three-dimensional formalism. The flexibility of the proposed algorithm is also presented using the same example. In the next chapter, these theoretical formulations will be implemented as a part of *MapleSim*, a commercial software for multibody and multi-domain modelling.

The two-dimensional version of the proposed algorithm, i.e., the application of graph-theoretic sensitivity analysis to planar mechanisms has been presented at the Multibody Dynamics Conference 2011 at the Universit Catholiqu de Louvain [76].

The three-dimensional formulation of the proposed graph-theoretic sensitivity analysis algorithm has been submitted to the ASME Journal of Computational and Nonlinear Dynamics and is currently undergoing the review process.

Chapter 5

Implementation and Results

Automated graph-theoretic sensitivity analysis for three-dimensional multibody systems has been implemented through a modelling and simulation software package called MapleSim . In this chapter, details of the implementation will be presented along with the results of the application of the algorithm to three dynamic systems and one kinematic system.

The first section will present the software platform MapleSim, in which the algorithm has been implemented. In light of this discussion, the structural details of the implementation will be illustrated. Next, the user interface for the application of the algorithm will be explained to outline the software platform of the implementation.

The application of the algorithm will be presented in the subsequent sections. First, a simple sliding pendulum mechanism will be presented to illustrate the intermediate steps of equation generation. Next, a three-dimensional pendulum model will be considered to illustrate validity of the three-dimensional formulation of the proposed algorithm. A motor driven slider crank mechanism will also be analyzed to demonstrate the applicability of the algorithm to multi-domain systems. Finally a double-wishbone suspension system will be considered to illustrate the applicability of the algorithm for complicated multibody automotive systems.

5.1 Computer Implementation

MapleSim is a multi-domain modelling and simulation tool from MapleSoft Inc. To illustrate the software implementation of the proposed algorithm, it is necessary to provide a brief description of the structure and the function of the relevant modules used by MapleSim .

5.1.1 Structure and function of MapleSim

MapleSim offers the capability of simulating dynamic systems consisting of mechanical, electrical, hydraulic, thermal, and magnetic sub-systems in an integrated modelling environment. However, the underlying engines that generate the governing equations are different for the different domains. Figure 5.1 an extremely simplified structure of the software.

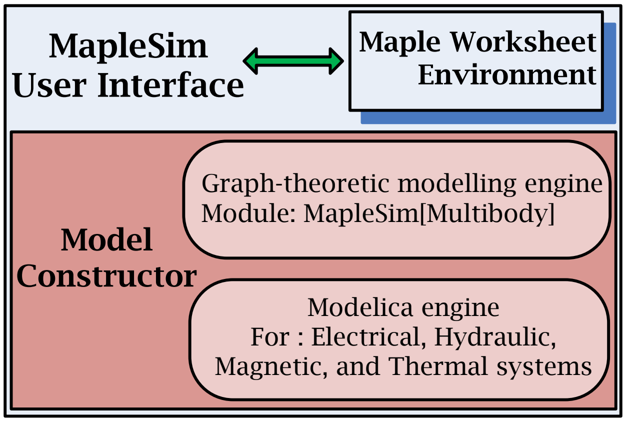

MapleSim
User Interface
Maple Worksheet
Environment
Model
Constructor
Graph-theoretic modelling engine
Module: MapleSim[Multibody]
Modelica engine
For : Electrical, Hydraulic,
Magnetic, and Thermal systems

Figure 5.1: Modules of MapleSim

The user interface of MapleSim offers a collection of drag and drop components from different domains, with which a model can be constructed by simply arranging and connecting them in proper order. The interface also interacts with the Maple worksheet environment, which can be used for downstream processing and numerical simulation. Figure 5.2 shows an example of an acausal model constructed in MapleSim .

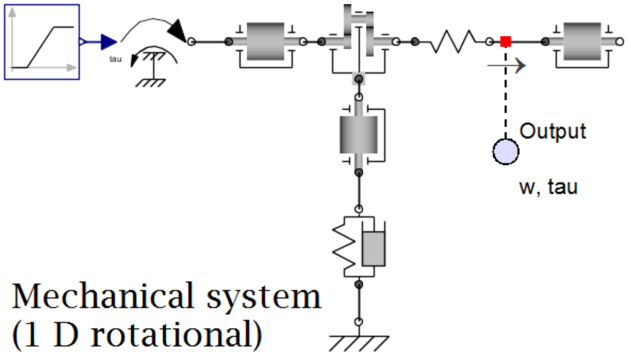


Figure 5.2: A typical model created in MapleSim

Once the model is defined, MapleSim calls the internal modules to generate the governing equations, which are then solved numerically for the generation of simulation results and visualization data. However, the modules used to generate governing equations are different for different domains. For multibody systems, the `MapleSim[Multibody]` module uses a graph-theoretic formulation; for all the other domains, Modelica-based formulations are used. A model constructor routine is used to combine the output of all the modules to generate the final set of governing equations.

To generate the governing equations for the multibody components, MapleSim uses a module called `MapleSim[Multibody]` that employs a graph-theoretic formulation as presented in chapter 2. It uses a number of sub-modules to process a graph-theoretic description of a multibody system and generate the governing equations. To explain how the implementation is done, it is important to illustrate the function of this module. Figure 5.3 below shows a schematic diagram of the function of `MapleSim[Multibody]` and its sub-modules.

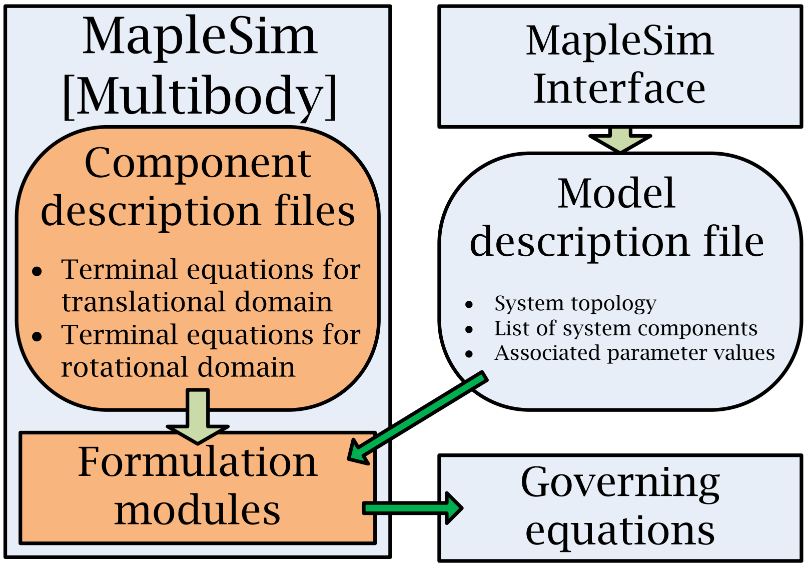
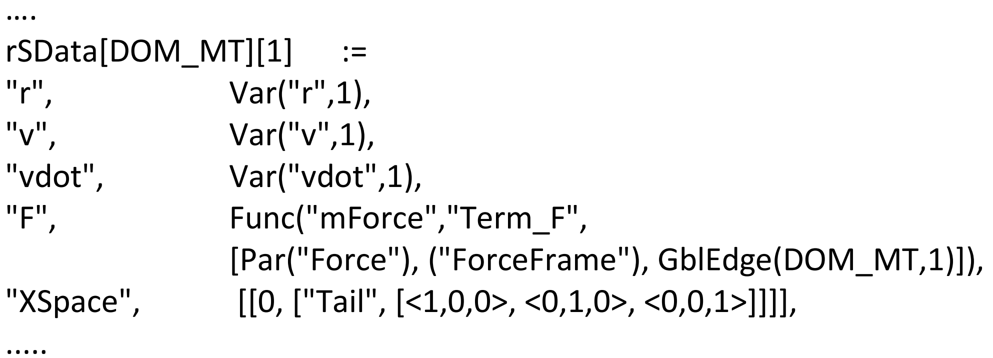

MapleSim [Multibody]
Component description files
Terminal equations for translational domain
Terminal equations for rotational domain
Formulation modules
MapleSim Interface
Model description file
System topology
List of system components
Associated parameter values
Governing equations

Figure 5.3: Structure and function of MapleSim[Multibody]

The module `MapleSim[Multibody]` uses many sub-modules that hold the definitions of the functions that are used to generate the governing equations using a graph-theoretic formulation. Apart from these "**Formulation Modules**", it also uses a list of "**Component description files**". These individual component description files hold the terminal equations for individual multibody components.

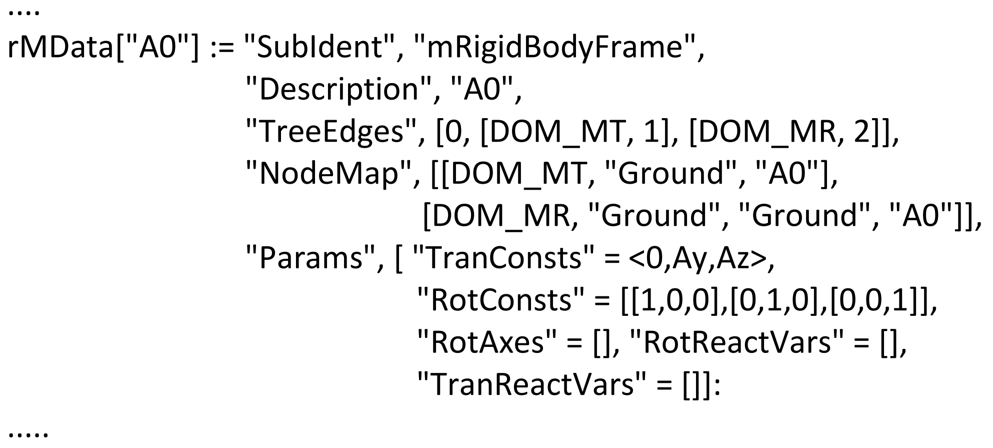
These files declare the relevant domains of the system and list the terminal equations for all through and across variables associated with those domains present in the system. For example, the terminal equation for the multibody component "**Force Driver**" contains information for the mechanical translational domain only, and the only specified quantity is the through variable for which the terminal equation must be specified. A portion of the actual description of this component is given below for reference.



```
....
rSData[DOM_MT][1]     :=
"r",                Var("r",1),
"v",                Var("v",1),
"vdot",             Var("vdot",1),
"F",                Func("mForce","Term_F",
                    [Par("Force"), ("ForceFrame"), GblEdge(DOM_MT,1)]),
"XSpace",            [[0, ["Tail", [<1,0,0>, <0,1,0>, <0,0,1>]]]],
.....
```

Figure 5.4: Sample from the component description file

The through and across variables for every domain have their own internal designation. For the translational domain, designated by `DOM_MT`,the symbols `r`, `v`, and `vdot` are used to refer to the position, velocity, and acceleration level across variables and the symbol `F` is used to refer to the through variable. Similarly, the symbols `Rp`, `w`, and `wdot` are used for across variables and the symbol `T` is used for through variables for the rotational domain designated by `DOM_MR`. The definitions of these quantities are specified in the component description files, which are different for different components.



```
....
rMData["A0"] := "SubIdent", "mRigidBodyFrame",
                "Description", "A0",
                "TreeEdges", [0, [DOM_MT, 1], [DOM_MR, 2]],
                "NodeMap", [[DOM_MT, "Ground", "A0"],
                            [DOM_MR, "Ground", "Ground", "A0"]],
                "Params", [ "TranConsts" = <0,Ay,Az>,
                            "RotConsts" = [[1,0,0],[0,1,0],[0,0,1]],
                            "RotAxes" = [], "RotReactVars" = [],
                            "TranReactVars" = []]:
.....
```

Figure 5.5: Sample from the model description file

When a model is created that includes multibody subsystems, the interface of MapleSim creates what is known as a model description file. A single entry of the model description file of a slider-crank mechanism is shown in figure 5.5. It corresponds to the edge that represents a rigid body frame.

The code shown above refers to a rigid body frame. Since a rigid body frame is necessarily a multibody component, the model specification file contains specifications relevant for both translational and rotational domains. The topology specification is done for both `DOM_MT` and `DOM_MR` as evident from the `NodeMap` specification. All the relevant parameter specifications are provided under the category of `Params`. Also, the information about whether or not a particular edge is placed in the tree is embedded into the model description file through the category `TreeEdges`.

For a complete model, the model description file contains a list of such entries for each component present in the model, which combine to form a text-based description of the system; the list of components used and the topology of the system described, i.e., information about the connectivity of these components. The file also holds the values of associated parameters and the structure of the user-selected tree.

When `MapleSim[Multibody]` is pressed into action, it loads up the formulation modules and the component description files. After reading these modules, it opens the model description file and creates the graph-theoretic model of the given system. The functions declared in the formulation module are then used to generate the governing equations.

5.1.2 Implementation of the proposed sensitivity algorithm

The main goal of the implementation is to obtain the sensitivity equations as the output of the module `MapleSim[Multibody]`. The plan here is to allocate different domains that correspond to individual model parameters, so that the equations produced from those domains become the sensitivity equations with respect to the associated parameters. Since the existing software structure retains the provision of declaring distinct constitutive equations for different domains, it is possible to include the sensitivity terminal equations as constitutive equations specific to the newly added sensitivity domains. The entire flow of operation is described below.

When a model is created, an user needs to specify a list of parameters with respect to which sensitivity analysis is to be carried out. Depending on this list, separate domains are declared to account for the sensitivity equations for a particular parameter. Since every multibody system needs two domains to be properly specified, for n parameters, a total of $2n$ sensitivity domains need to be declared.

For each of these newly-defined domains, the terminal equations must be specified to enable the generation of sensitivity equations. These sensitivity equations can be generated on the fly using the formulations presented in chapter 4. The component description files must be modified to include the definitions of these additional sensitivity terminal equations and that can be achieved using the Maple programming language.

The model description file must also be modified to supply the relevant information to the `MapleSim[Multibody]` package regarding the newly defined domains. This depends on the actual parameters supplied and is done automatically on a case by case basis.

Once, the relevant files are properly modified, the core files of `MapleSim[Multibody]` must be replaced by the modified versions of the sub-modules and component description files. This can be easily done by redirecting the initialization files to the directory containing the modified files.

This step creates a case-specific environment. At this point, all the regular functions of `MapleSim[Multibody]` remain available to the user and can be used to generate equations from the modified model description file. However, since all the sensitivity domains are parts of the model description, the generated equations include not only the governing equations, but also the sensitivity equations with respect to all the model parameters specified by the user.

User interface

Since the implementation requires a somewhat complicated sequence of codes, it is beneficial to design a simplified user interface to carry out these steps in the proper order. To retain the functionality of MapleSim's design environment, it is also desirable to maintain the constant connection with MapleSim's graphical user interface.

MapleSim has the capability of linking its model to a Maple worksheet where further analyses can be carried out. In case of multibody models, the attached worksheet can be used to access the model description file. This capability makes it possible to create a worksheet template, which can be used for the purpose of sensitivity analysis. To illustrate the present user interface for the graph-theoretic sensitivity analysis of multibody systems, it is worthwhile to present the work flow for the current form of the worksheet template:

1. The first step is to create and validate a model for the required system in MapleSim.

2. Once the model is completed, the attachment of the worksheet template can be used to access the model description file.

3. Next, a parameter detection routine automatically determines the model parameters present in the model and asks the user to select a list of parameters for subsequent analysis.

4. Depending on the user selected model parameters, the worksheet automatically modifies the model description file, the formulation module files, and the component description files.

5. After the modification, the files are incorporated into `MapleSim[Multibody]`, which makes it capable of generating sensitivity equations.

6. Finally, the sensitivity equations are generated using the functions from the modified `MapleSim[Multibody]` module. These equations can then be simulated using Maple's built-in numerical engine.

7. The original version of `MapleSim[Multibody]` can be reinstated by simply closing the worksheet.

Using the modified version of `MapleSim[Multibody]`, a number of systems were analyzed to test the effectiveness and efficiency of the proposed algorithm. In the following sections, four examples will be presented to showcase the application of the proposed graph-theoretic algorithm for sensitivity analysis.

5.2 Example problems

The example problems are selected to showcase the validity, accuracy and efficiency of the proposed algorithm. For all the examples presented in this section, the system equations are first generated using the unmodified version of `MapleSim[Multibody]` to explore their behaviour under various initial conditions. This step also enables us to derive the governing equations for the system which are required for validation purposes.

The current version of commercially available `MapleSim[Multibody]` does not include the necessary modules to generate the governing equations for systems with electrical components. However, for the analysis of the driven slider-crank mechanism, a deprecated versions of component description files are used for the electrical components.

After the models are created in MapleSim, symbolic sensitivity equations are generated for the system using both direct differentiation and the proposed graph-theoretic algorithm, which allows the comparison of their performances and helps in the evaluation of the proposed algorithm. The sensitivity information evaluated hereby is also verified using a finite difference formulation on the governing equations.

5.2.1 A sliding pendulum mechanism

For the first example, a sliding pendulum mechanism is chosen for analysis. Figure 5.6 shows the configuration of the system.

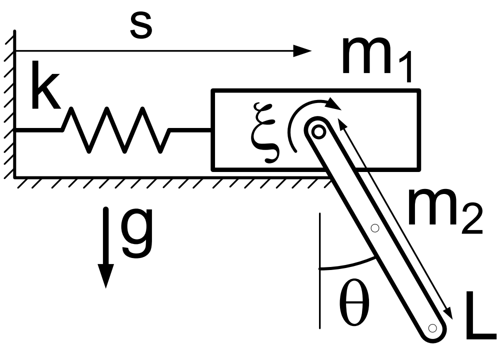


Figure 5.6: Sliding pendulum mechanism

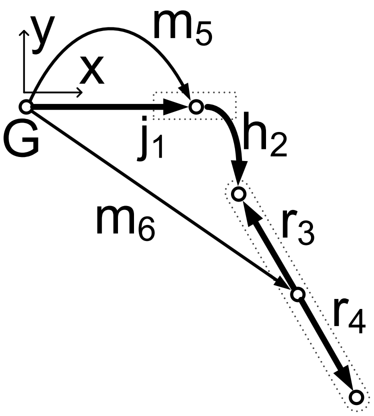


Figure 5.7: System linear graph

It constitutes two rigid bodies of masses m_1 and m_2, which are connected by a revolute joint. The revolute joint includes a torsional damper with damping coefficient ξ. The parameters L and r are the length and radius of circular cross-section of the pendulum. The sliding block is allowed to move along a prismatic joint and is connected to a linear spring of spring constant k and undeformed length s_0. The acceleration due to gravity g acts in the downward direction.

The linear graph of the system is shown in figure 5.7. By choosing the edges j_1, h_2, r_3 and r_4 as the tree branches, the governing equations for the system are generated in terms of the primary variables $s(t)$ and $\theta(t)$, which represent the displacement along the prismatic joint and the rotation of the pendulum, respectively:

$$\begin{aligned}&\left(\frac{m_2 r^2}{4} + \frac{m_2 L^2}{3}\right)\ddot{\theta} + \frac{m_2 L\cos\theta}{2}\ddot{s} + \frac{m_2 g L}{2}\sin\theta + \xi\dot{\theta} = 0\\&\frac{m_2 L\cos\theta}{2}\ddot{\theta} + (m_1 + m_2)\ddot{s} - \frac{m_2 L\sin\theta}{2}\dot{\theta}^2 + k(s - s_0) = 0\end{aligned} \tag{5.1}$$

The MapleSim implementation of the graph-theoretic algorithm is used to generate the sensitivity equations for this system. To explain the process in MapleSim, a step by step description of the process will now be presented.

To perform sensitivity analysis with respect to the model parameter L, new domains are introduced to account for the sensitivity equations. Since both rotational and translational domains are present in the system, the additional domains are named as `DOM_MR_SEN_L` and `DOM_MT_SEN_L,`to include sensitivity information from both domains.

The component description files for all the elements of the graph, i.e., the revolute joint, the rigid arm element, the rigid body, and the prismatic joint, are automatically modified to include the terminal equations for the additional domains, as described in section 5.1.2.

As mentioned in chapter 4, the first step is the generation of cutset equations for all the rigid bodies and the passive joints selected in the tree and their projection on to the respective motion spaces.

In MapleSim, this is done in a two stage process. First, the branch-transformation equations for the edge in question is generated and projected on to the motion spaces. Second the terminal equations of the through variable of the edge in question is projected on to the motion space as well. These equation fragments are then combined to create the dynamic equations. In this system, the sensitivity equations are generated from the cutset equations associated with edge j_1 for the domains `DOM_MT_SEN_L` and `DOM_MT` and the edge h_2 for the domains `DOM_MR_SEN_L` and `DOM_MR`.

For the edge j_1, the right hand side of the branch-transformation equation corresponding to the `DOM_MT` domain is given by

$$-\bar{F}^{m_5} - \bar{F}^{m_6} \tag{5.2}$$

Similarly, for the domain `DOM_MT_SEN_L` the following expression is obtained

$$-\bar{F}_L^{m_5} - \bar{F}_L^{m_6} \tag{5.3}$$

In equations (5.2) and (5.3), the superscripts denote the edges that the quantities are associated with and the subscripts refer to the parameter L with respect to which the sensitivity analysis is being carried out. The sensitivity domains are designed and implemented to hold the derivatives of the equations from the mechanical domains. Thus equation (5.3) can be easily verified to be the derivative of equation (5.2) with respect to L.

These vector expressions can be expanded using the terminal equations specified for the relevant components for the domains `DOM_MT_SEN_L` and `DOM_MT`. However, care must be

taken to ensure that before any arithmetic procedure is carried out, the vectors are expressed in a common node (frame) of the linear graph. For this example, the vector equations are expanded and expressed in the node inertial G as given below.
From the domain `DOM_MT` the expression 5.2 expands to

$$\begin{bmatrix} m_1\,\ddot{s} + \left(\ddot{s} + \frac{L\cos\theta}{2}\ddot{\theta} - \frac{L\sin\theta}{2}\,\dot{\theta}^2\right) m_2 \\ m_1\,g + m_2\,g + \left(\frac{L\sin\theta}{2}\,\ddot{\theta} + \frac{L\cos\theta}{2}\,\dot{\theta}^2\right) m_2 \\ 0 \end{bmatrix} \tag{5.4}$$

From the domain `DOM_MT_SEN_L` the expression 5.3 transforms into

$$\begin{bmatrix} m_1\,\ddot{s}_L + m_2\left(\ddot{s}_L + \frac{L\cos\theta}{2}\ddot{\theta}_L + \Upsilon_1\right) \\ m_2\left(\frac{L\sin\theta}{2}\ddot{\theta}_L + \Upsilon_2\right) \\ 0 \end{bmatrix} \tag{5.5}$$

$$\Upsilon_1 = \left(\frac{\cos\theta}{2} - \frac{L\sin\theta}{2}\theta_L\right)\ddot{\theta} - \left(L\sin\theta\,\dot{\theta}\right)\dot{\theta}_L \ - \ \left(\frac{\sin\theta}{2} + \frac{L\cos\theta}{2}\theta_L\right)\dot{\theta}^2 \tag{5.6}$$

$$\Upsilon_2 = \left(\frac{\sin\theta}{2} + \frac{L\cos\theta}{2}\theta_L\right)\ddot{\theta} + \left(L\cos\theta\,\dot{\theta}\right)\dot{\theta}_L + \ \left(\frac{\cos\theta}{2} - \frac{L\sin\theta}{2}\theta_L\right)\dot{\theta}^2 \tag{5.7}$$

These equations are associated with the edge j_1, which represents the prismatic joint with its axis of translation oriented along the ground node x-axis. The associated motion space for this component is given by the vector expressed in node G as shown below.

$$\text{Motion-Space}_{\text{prismatic joint}} = \left\{\begin{matrix} 1 & 0 & 0 \end{matrix}\right\}^T \tag{5.8}$$

For the generation of system equations, the usual procedure is to evaluate the dot product of the cutset equation and the space unit vector. However, to generate the sensitivity equations, the procedure needs to capture the chain rule of differentiation and must be properly modified as shown below.

$$\begin{bmatrix} \text{Projected} \\ \text{expression} \end{bmatrix} = \begin{bmatrix} \text{Expression} \\ \text{from} \\ \text{DOM\_MT\_SEN\_L} \end{bmatrix} \cdot \begin{bmatrix} \text{Motion} \\ \text{space} \\ \text{vector} \end{bmatrix} + \begin{bmatrix} \text{Expression} \\ \text{from} \\ \text{DOM\_MT} \end{bmatrix} \cdot \begin{bmatrix} \text{Sensitivity} \\ \text{of motion} \\ \text{space vector} \end{bmatrix} \tag{5.9}$$

The motion space vector is normally a function of the state variables and are implicitly dependent on the model parameters. For this reason, the derivatives of the space vector are important for the sensitivity equations. For this particular example however, the motion space is a constant vector and its sensitivity is a zero vector. Thus for this mechanism only, the second part of equation (5.9) can be safely ignored and the resulting projected expression can be formed by extracting the x component of the cutset equation from `DOM_MT_SEN_L`.

The next step in this derivation is the term which contains the dynamics of the edge for which the cutset equation is obtained. This term is obtained by the terminal equation corresponding to the edge in question. Since the objective is to construct the projected cutset equation, only the terminal equation for the through variable is required at this stage. For the edge j_1 two such terminal equations can be extracted, one each for the domains `DOM_MT`and `DOM_MT_SEN_L`. For the domain `DOM_MT_SEN_L` the following expression is obtained

$$\bar{F}_L^{j_1} = \left\{ \begin{array}{ccc} -k\, s_L\,(t) & {}_yF_L^{j_1}\,(t) & {}_zF_L^{j_1}\,(t) \end{array} \right\}^{\mathrm{T}} \tag{5.10}$$

The expression obtained from the domain `DOM_MT` is shown in equation (5.11). As it can be clearly seen, equation (5.10) can be derived by differentiating equation (5.11) with respect to the model parameter L.

$$\bar{F}^{j_1} = \left\{ \begin{array}{ccc} -k\,(s\,(t) - s_0) & {}_yF^{j_1}\,(t) & {}_zF^{j_1}\,(t) \end{array} \right\}^{\mathrm{T}} \tag{5.11}$$

To derive the complete projected cutset equation in the domain `DOM_MT_SEN_L`, one must project equation (5.10) onto the motion space unit vector shown in equation (5.8). To do that, one must also use the modified projection equation illustrated in (5.9). Since equation (5.8) is a constant unit vector aligned with the inertial x axis, the result of the projection can be easily determined by extracting the x component of the right hand side of the equation (5.10).

$$\left(F_L^{j_1}\right)_{\text{projected on to the motion space}} = -k\, s_L\,(t) \tag{5.12}$$

By combining the projected branch transformation and equation (5.12), the first sensitivity equation is obtained.

$$m_1\, \ddot{s}_L + m_2 \left(\ddot{s}_L + \frac{L\cos\theta}{2}\ddot{\theta}_L + \Upsilon_1 \right) + ks_L = 0 \tag{5.13}$$

where Υ_1 is given by equation (5.6).

Similarly, from the edge h_2, cutset equations can be generated for the domains DOM_MR and DOM_MR_SEN_L. The corresponding motion space is described by the vector given below.

$$\text{Motion-Space}_{\text{revolute joint}} = \left\{ \begin{array}{ccc} 0 & 0 & 1 \end{array} \right\}^T \tag{5.14}$$

Using the same method described above, the second sensitivity equation can be generated from the cutset equations for the edge h_2. Resolving every vector in the node G, the following equation is obtained.

$$\left(\frac{m_2 \cos\theta}{2} - \frac{m_2\, L \sin\theta\, \theta_L}{2} \right) \ddot{s} + \frac{m_2\, L \cos\theta}{2} \ddot{s}_L + \left(\frac{m_2\, r^2}{4} + \frac{m_2\, L^2}{3} \right) \ddot{\theta}_L + \Upsilon_3 = 0 \tag{5.15}$$

where

$$\Upsilon_3 = \frac{2m_2\, L}{3} \ddot{\theta} + \frac{m_2\, g}{2} \sin\theta + \frac{m_2\, g\, L}{2} \cos\theta\, \theta_L + \xi\, \dot{\theta}_L \tag{5.16}$$

Since the revolute joint includes a rotational damper, it is important to include the terminal equation for the domain DOM_MR_SEN_L in the projected equation. As a result the expression for the damping torque appears in equation (5.15).

Since this is a simple system, the validity of the generated sensitivity equations can be verified by inspection, which reveals that equations (5.13) and (5.15) are indeed the derivatives of the governing equations described in equation (5.1).

The sensitivity equations can also be verified by simulation. Table A.3 lists the values of the model parameters used for the simulation. The sensitivity of the state variables, i.e., $s(t)$ and $\theta(t)$ in this problem, can be evaluated by simulating equations (5.13), (5.15) and (5.1). The sensitivity results can then be verified using the finite difference method. The system was simulated numerically using "Runge-Kutta-Fehlberg method" in Maple, using the following initial conditions for the states.

$$\begin{array}{llll} s\,(0) = 0.5 & \dot{s}\,(0) = 0 & \theta\,(0) = 0 & \dot{\theta}\,(0) = 0 \\ s_L\,(0) = 0 & \dot{s}_L\,(0) = 0 & \theta_L\,(0) = 0 & \dot{\theta}_L\,(0) = 0 \end{array} \tag{5.17}$$

The figures 5.8 and 5.9 clearly identify the response of the system and validate the solution of the generated sensitivity equation. Figure 5.8 shows that the values of $s(t)$ and $\theta(t)$ settles after initial oscillations, with $s(t)$ settling to the value of s_0. This behaviour is expected because, in the sliding pendulum system, both of the principal modes of oscillations are damped and the system is destined to damp out any initial disturbances.

The behaviour of $s_L(t)$ can also be explained by considering the structure of the system. By considering the configuration of the system, it can be clearly seen that for different values of the parameter L, the resulting output of $s(t)$ always approach the same datum level s_0 asymptotically. As a result the sensitivity of $s(t)$ with respect to the parameter L also oscillates and then settles to zero.

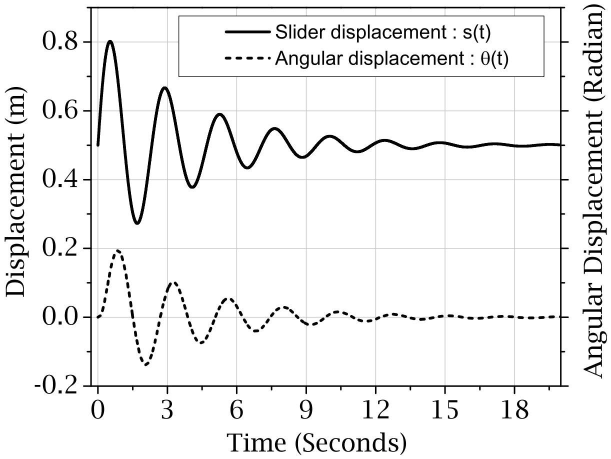


Figure 5.8: Simulation results

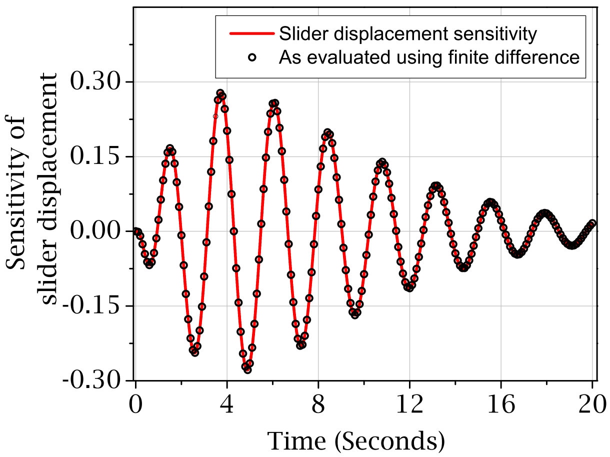


Figure 5.9: Sensitivity s_L vs time

This example of the sliding pendulum illustrates the validity of the implemented algorithm for multibody mechanisms. However, due to the simplistic nature of this system, it is worthwhile to consider other examples, where the algorithm can be tested without the simplifying circumstances. For example, the motion spaces for this system were found to be constant vectors. It needs to be verified if the algorithm works equally well with mechanisms where motion spaces are functions of time. Also, different tree selection schemes need to be considered in order to validate the algorithm for systems governed by differential algebraic equations (DAEs).

In the subsequent examples, these issues will be addressed using different mechanisms. First a three-dimensional pendulum will be presented to validate the generated equations for a system where the motion spaces are not constant vectors.

5.2.2 A three-dimensional pendulum

The three-dimensional pendulum is a rigid cylindrical mass, which is attached to a fixed point in space through a spherical joint at some distance away from its centre of mass. Due to the complete lack of rotational constraints from the joint, a three-dimensional pendulum can exhibit very complicated motions and can take up any angular orientation in space. From the standpoint of graph-theoretic system modelling, if the spherical joint is selected as the branch, the governing equations are generated in terms of the three Euler angles. For this example, a body fixed Euler 1-2-3 rotation scheme is used which results in governing equations in terms of the three sequential (1-2-3) angles ζ, η and ξ.

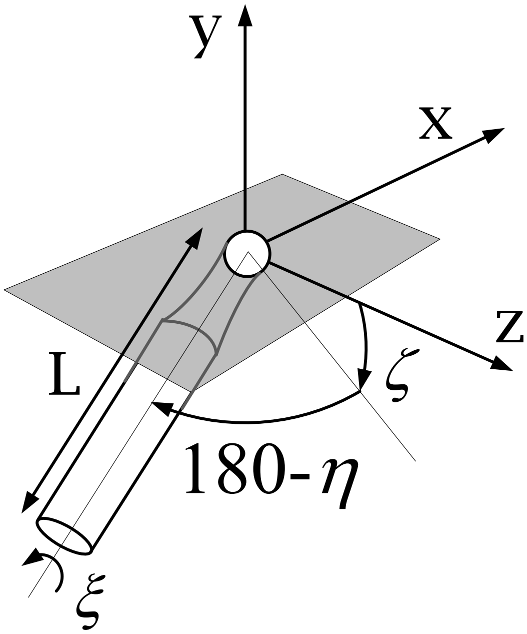


Figure 5.10: Three-dimensional pendulum

For this example, the implemented algorithm is used to study the effect of the length of the pendulum on its motion, in other words, to evaluate the sensitivities of ζ, η and ξ with respect to the length of the pendulum L.

Due to the complicated nature of the governing equations, it is not possible to present them in symbolic form in this report. However, the response of the system can be easily obtained by numerically simulating the system using the `dsolve[numeric]` routine from Maple. For this example, the simulation results are validated using the simulation tool MapleSim . The resulting sensitivity information are subsequently validated using a finite difference formulation.

The unique aspect of the spherical joint is the fact that because of the assumed Euler angles, the joint can be considered as the combination of three revolute joints connected in series, with their axes aligned to the body-fixed x, y, and z axis. As such the motion spaces for the subsequent joints become functions of the rotation of the previous joints.

For this example, if the spherical joint is selected in the tree, the resulting cutset equation must be projected on to the three motion spaces of the joint. The first motion space is the constant vector along the inertial x axis. This motion space corresponds to the joint that is connected to the ground and allows rotation about the x axis only.

However, due to rotation about this axis, the orientations of the subsequent joints are rotated and their motion spaces become functions of the state vector. For example, the second motion space for the spherical joint is defined as

$$\text{Motion Space}_2 = \left\{ \begin{array}{ccc} 0 & \cos\zeta & \sin\zeta \end{array} \right\}^{\mathrm{T}} \tag{5.18}$$

This implies that during the process of projecting the cutset vector from the sensitivity domains, the second part of equation (5.9) can no longer be ignored and the derivative of the motion space must be evaluated to complete the process. For this particular motion space shown in equation (5.18), the derivative is evaluated as

$$\text{Derivative of Motion Space}_2 = \left\{ \begin{array}{ccc} 0 & -\zeta_L \sin\zeta & \zeta_L \cos\zeta \end{array} \right\}^{\mathrm{T}} \tag{5.19}$$

The final motion space for the spherical joint is given by the third axis which has already undergone some changes in its orientation due to rotations about the first two axes. The motion space vector and the corresponding derivative is given below.

$$\begin{aligned} (\text{Motion Space}_3) &= \left\{ \begin{array}{ccc} \sin\eta & -\cos\eta\sin\zeta & \cos\eta\cos\zeta \end{array} \right\}^{T} \\ \frac{d}{dL}\left(\text{Motion Space}_3\right) &= \left\{ \begin{array}{c} (\cos\eta)\,\eta_L \\ (\sin\eta\sin\zeta)\,\eta_L - (\cos\eta\cos\zeta)\,\zeta_L \\ -(\sin\eta\cos\zeta)\,\eta_L - (\cos\eta\sin\zeta)\,\zeta_L \end{array} \right\} \end{aligned} \tag{5.20}$$

The sensitivity equations for the three-dimensional pendulum were derived using the `MapleSim` implementation of the algorithm. Since it is not possible to present the complete set of sensitivity equations, the simulation results are used to verify its accuracy.

The Runge-Kutta-Fehlberg method was used to numerically integrate the sensitivity equations along with the system equations. The values chosen for absolute and relative errors were 10^{-4} and 10^{-5} respectively. The initial conditions for the state variables were chosen to simulate a typical precessional motion of a three-dimensional pendulum. Table 5.1 lists the numerical values used as the initial conditions for the system equations.

For the sensitivity equations, all the initial conditions are taken to be zero. The decision to choose this particular initial condition can be justified by considering the nature of the variables of sensitivity equations. The model parameter values are listed in Appendix A.5.

Table 5.1: Initial conditions for the three-dimensional pendulum

Variable Name	**Initial Value**	**Unit**
$\zeta(0)$	0.5	rad
$\eta(0)$	0	rad
$\xi(0)$	0	rad
$\dot{\zeta}(0)$	0.3	rad/s
$\dot{\eta}(0)$	0	rad/s
$\dot{\xi}(0)$	0.1	rad/s

In this problem, all the state variables, i.e., ζ, η, and ξ can be specified independently. Since during the initialization, these state variables are not dependent on the model parameter L, it can be stated that the sensitivities of these quantities with respect to the model parameter L is indeed zero at the initial point. Similar logic would also be applicable for all models with independent state variables.

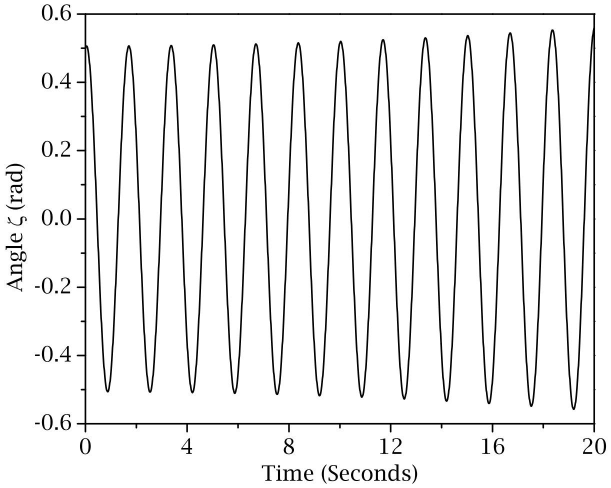


Figure 5.11: Variation of ζ

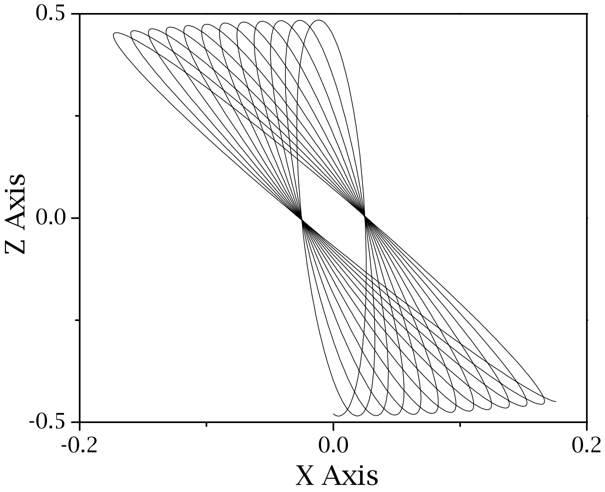


Figure 5.12: Trajectory of pendulum tip

Figures 5.11 and 5.12 show the variation of the angle ζ and the projected trajectory of the tip of the pendulum on the X - Z plane. The complex motion of the pendulum is clearly evident from these plots.

Since this is a highly non-linear system, it is difficult to generalize the nature of the behaviour of the state variables. The variations of the sensitivity quantities, i.e., ζ_L, η_L, and ξ_L are found to be highly dependent on the initial conditions and the model parameters. As a result, it is very difficult to gather physical insight from the system response.

In general, small changes in the model parameter L are found to result in completely different system responses, which are reflected in the increasing amplitude of the oscillations of sensitivity quantities. These quantities are obtained by numerically simulating the set of sensitivity equations generated by the implemented sensitivity algorithm.

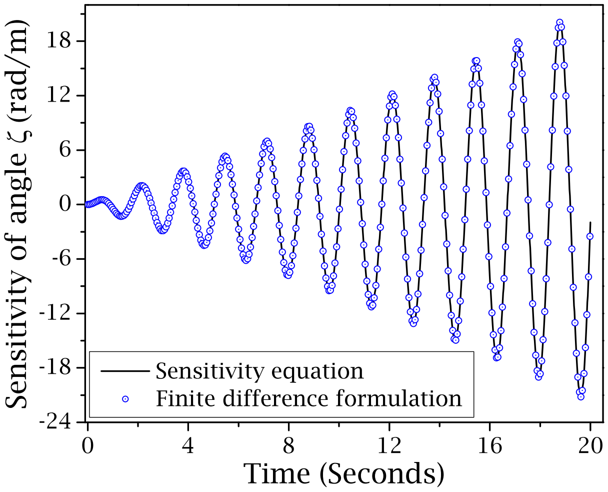


Figure 5.13: Variation of ζ_L

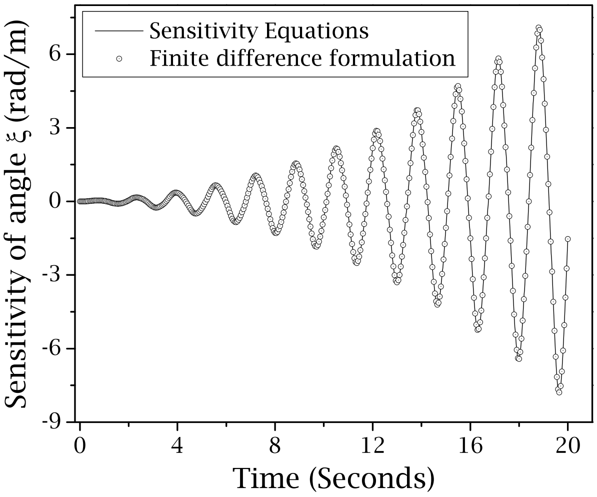


Figure 5.14: Variation of ξ_L

The evaluated sensitivity information is compared with those obtained using a finite difference formulation. The comparison between the finite difference method and the numerical solution of sensitivity equations are presented in figures 5.13 and 5.14. It clearly validates the numerical solution and the structure of the generated sensitivity equations.

So far, the presented examples involved systems modelled using sets of ordinary differential equations. To test the implemented algorithm for a broader class of problems, it is important to consider mechanisms governed by sets of differential-algebraic equations or DAEs.

5.2.3 A DC motor-driven slider-crank mechanism

In this section, an electrically driven slider-crank mechanism will be analyzed to evaluate the performance of the implemented algorithm. Figure 5.15 presents the system schematically.

The presented planar slider-crank mechanism is driven by an electric motor. This system is specifically chosen to showcase the capability of the implemented algorithm toward multi-domain systems and to test the algorithm's ability to process different tree selection schemes. This system has an electrical side where a capacitor, resister and an inductor is used along with a voltage source to drive an electro-mechanical transducer. This provides a torque to the mechanical side, where a crank and a connecting rod moves a sliding mass along a prismatic joint with viscous damping coefficient ξ.

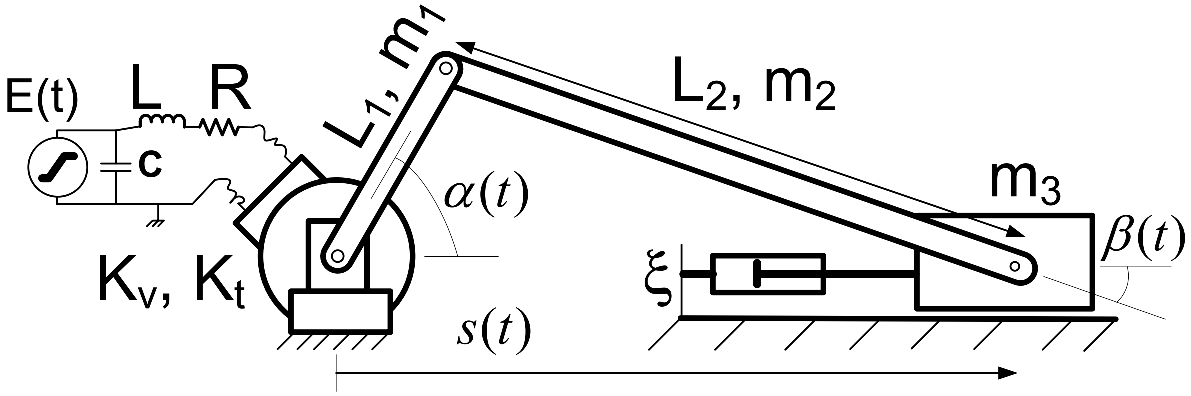


Figure 5.15: Basic schematic of the driven slider-crank mechanism

The unique component of this model is the DC motor that drives the system. It transfers energy back and forth between electrical and mechanical domains. In practical terms, this is an example of an electro-mechanical transducer that produces a torque on the mechanical side and an opposing potential difference on the electrical side. The magnitude of the torque is proportional to the current flowing through the electrical side and the magnitude of the potential difference produced is proportional to the angular velocity of the mechanical side. The constants of proportionality are known as transducer coefficients. For this model the symbols K_v and K_t are used as the transducer coefficients.

In the current system, L_1 and L_2 are the lengths of the crank and the connecting rod respectively, m_1, m_2, and m_3 are the masses of the three components, L is the inductance, R is the resistance, C is the capacitance, and $E(t)$ is the voltage signal used to drive the electrical circuit. The numerical values used for the simulation and the subsequent analysis of this system are listed in the table A.2.

The MapleSim implementation of the proposed algorithm is used to generate the sensitivity equations with respect to the model parameter L_2. A linear graph is used to represent the topology of the system in both mechanical and electrical domains.

Figure 5.16 below illustrates the graph-theoretic representation of the system.

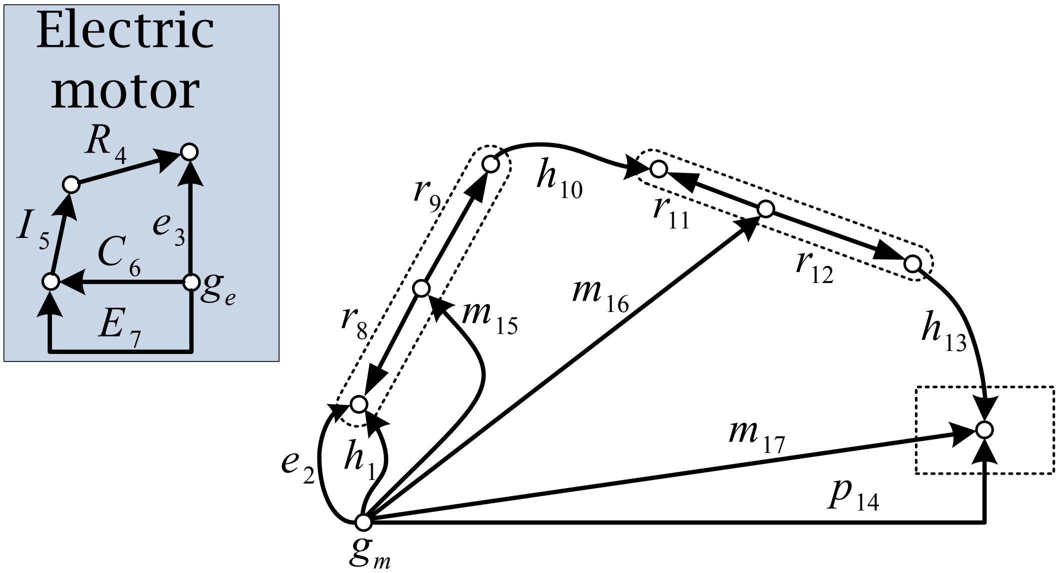


Figure 5.16: Linear graph representation of the system

In figure 5.16, the mechanical and electrical domains are shown separately. The connection between the two domains is achieved by the transducer element denoted by edges e_2 and e_3. In the mechanical domain, the edges h_1, h_{10}, and h_{13} represent revolute joints, $r_{8,9,11,12}$ are rigid arm elements, $m_{15,16,17}$ are rigid bodies, and p_{14} is a prismatic joint. On the other hand, in the electrical domain, E_7 is a voltage source, R_4 is a resistor, C_6 is the capacitor, and I_5 is the inductor.

The choice of tree branches determines the quantities with respect to which the equations are generated. For this model, the following edges are selected as the tree branches.

For the mechanical translational domain: Edges h_1, $r_{8,9}$, h_{10}, $r_{11,12}$, and h_{13}

For the mechanical rotational domain: Edges h_1, $r_{8,9}$, p_{14}, $r_{11,12}$, and h_{13}

For the electrical domain: Edges E_7, I_5, and R_4

This ensures the generation of equations in terms of the angles $\alpha(t)$, $\beta(t)$, the motor current $iM(t)$, and their sensitivities $\alpha_{L_2}(t)$, $\beta_{L_2}(t)$, and $iM_{L_2}(t)$. Since the degrees of freedom in the mechanical part is one, this particular choice of tree branches leads to the generation of one kinematic constraint equation, which results in one kinematic sensitivity equation.

The modified `MapleSim[MultiBody]` package is used to generate the sensitivity equations for this model. It results in a total of six differential equations (three system equations and three sensitivity equations) and two kinematic constraint equations.

The kinematic equations are presented below.

$$\begin{bmatrix} L_1 \sin\alpha - L_2 \cos\beta \\ L_1\, \alpha_{L_2} \cos\alpha - \cos\beta + L_2\, \beta_{L_2} \sin\beta \end{bmatrix} = 0 \tag{5.21}$$

From equation (5.21) it can be shown that the second equation is indeed the derivative of the first equation with respect to the model parameter L, which confirms the accuracy of the implemented algorithm.

To test it even further, at this point the model is analyzed again after replacing the edge h_{10} with the edge p_{14} in the mechanical translational tree. This way the equations are generated in terms of the angles $\alpha(t)$, $\beta(t)$, the motor current $iM(t)$, the slider displacement $s(t)$, and their sensitivities. When this choice of tree is used, the modified `MapleSim[Multibody]` routine generates the following set of constraint equations.

$$\begin{bmatrix} -L_2 \cos(\alpha+\beta) + s\cos\alpha - L_1 \\ L_2 \sin(\alpha+\beta) - s\sin\alpha \\ -\cos(\alpha+\beta) + L_2\,(\alpha_{L_2} + \beta_{L_2}) \sin(\alpha+\beta) + s_{L_2}\cos\alpha - s\alpha_{L_2}\sin\alpha \\ \sin(\alpha+\beta) + L_2\,(\alpha_{L_2} + \beta_{L_2}) \cos(\alpha+\beta) - s_{L_2}\sin\alpha - s\alpha_{L_2}\cos\alpha \end{bmatrix} = 0 \tag{5.22}$$

Once again, the generated equations shown in (5.22) can be easily verified to be correct, as the sensitivity constraint equations are found to be the derivatives of the system constraint equations. This further confirms the validity of the graph-theoretic implementation of sensitivity analysis.

A forward dynamic analysis is carried out using the sets of equations generated by the two distinct tree selection schemes to evaluate the variation of the sensitivities of the state variables with respect to the model parameter L_2. This requires the declaration of initial conditions for the system and demands some explanation.

In a system governed by DAEs, the initial values of the state variables can't be specified arbitrarily. For consistent initialization, the initial values must satisfy the position and velocity level constraint equations. To evaluate these initial conditions for this system, the initial value of the angle $\alpha(t)$ and its derivative is assumed to be specified.

$$\alpha(0) = \frac{Pi}{3}\, rad \qquad \dot{\alpha}(0) = 0\, rad/s \tag{5.23}$$

Since the value of α is arbitrarily specified at the initial point, it also follows that the initial value of the sensitivity of α is in fact zero, i.e., $\alpha_{L_2} = 0$ and $\dot{\alpha}_{L_2} = 0$. The initial values for the rest of the state variables must be evaluated by solving the position and velocity constraint equations.

Given the value of α and α_{L_2}, depending on the choice of the tree branches, equations (5.21) or (5.22) can be used to solve for either β and β_{L_2} or β, s and their sensitivities. For example, plugging equation (5.23) into equation (5.21) leads to the calculation of the following quantities.

$$\beta(0) = 1.27\, rad \quad and \quad \beta_{L_2}(0) = 0.201\, rad/m \tag{5.24}$$

It is interesting to note that in this example, all the sensitivity quantities are not equal to zero to start with. As evident from equation (5.24), the initial value of β is dependent on the model parameters. This in turn implies that the initial value for the sensitivity of β is non-zero, as demonstrated in the solution above. Also, the driving voltage $E(t)$ is specified as given below (5.25).

$$E\left(t\right) = \begin{cases} 0 & t \le 2 \\ 10t - 20 & 2 < t < 4 \\ 20 & 4 \le t \end{cases} \quad \text{Volts} \tag{5.25}$$

To properly initialize the values for the motor current $iM(t)$, its sensitivity $iM_{L-2}(t)$, and their derivatives, it is necessary to recognize that since the highest order of derivative of the quantity $iM(t)$ is one, the only required quantity for the integration is the value of $iM(t)$ and $iM_{L_2}(t)$ at the initial point. The values of $iM(0)$ and $iM_{L_2}(0)$ can be evaluated by solving the electrical circuit at $t = 0$ where the initial value of $\dot{\alpha}\left(0\right) = 0$ must be used as the coupling term. The following values are obtained by solving the electrical circuit at $t = 0$.

$$\begin{gathered} iM(0) = 0\, A \\ iM_{L_2}(0) = 0\, A/m \end{gathered} \tag{5.26}$$

The combined set of system and sensitivity equations obtained by both trees are integrated in Maple using built-in `dsolve[numric]` routine. The DAE extension of the "Runge-Kutta-Fehlberg" algorithm was used for the forward dynamic simulation.

Figure 5.17 demonstrates the coupling of electrical domain to the mechanical domain in the system. The rippling effect in the electrical circuit as shown in figure 5.19 is clearly evident in the plot of the crank angle α. The plot shown in 5.18 shows the effect of damping in the system, which prevents it from unbounded growth.

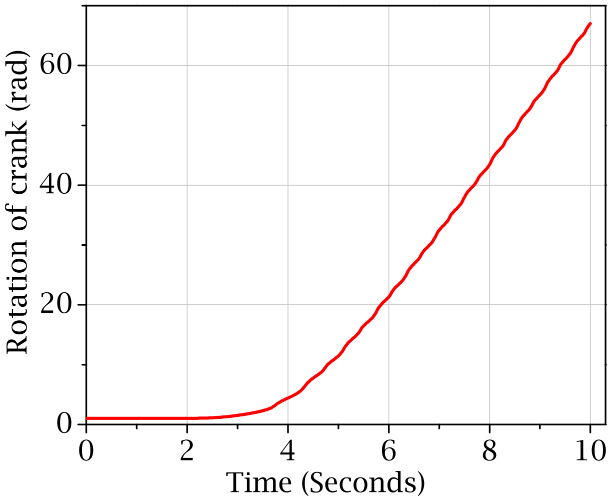


Figure 5.17: Variation of α

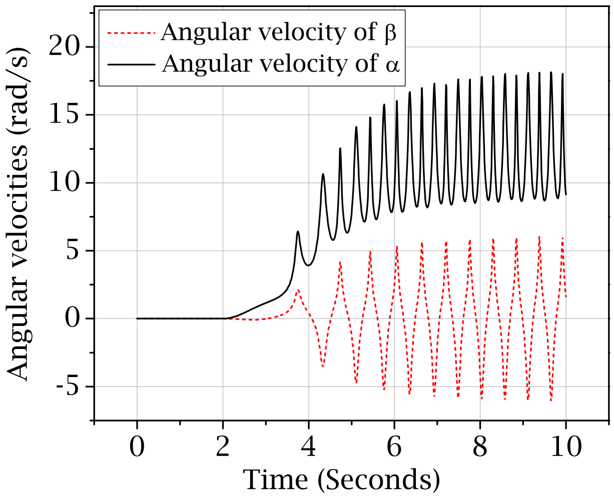


Figure 5.18: Variation of $\dot{\alpha}$ and $\dot{\beta}$

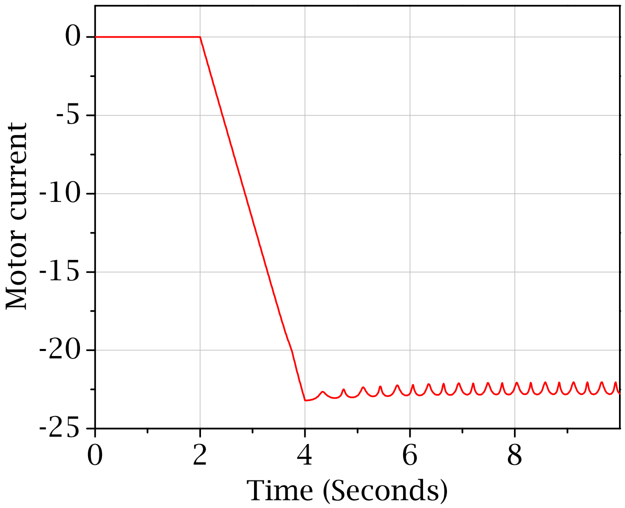


Figure 5.19: Variation of motor current

Apart from the system variables, the simulation also evaluates the sensitivities of the state vector. The implemented algorithm can be validated by comparing the solutions of α_{L_2} and β_{L_2} obtained by solving two distinct sets of equations, obtained by different choices of branches. Also the overall accuracy of the numerical solution can be checked by comparing them with the results obtained using finite difference approach.

The plots shown in figures 5.20 and 5.21 illustrate the comparison between the solution of α_{L_2} and β_{L_2} obtained by integration different sets of equations. They clearly validate the implemented algorithm as the numerical results are found to be identical to each other.

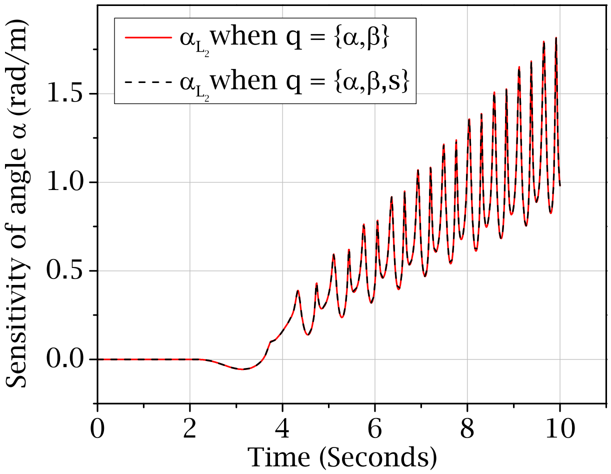


Figure 5.20: Comparison of α_{L_2}

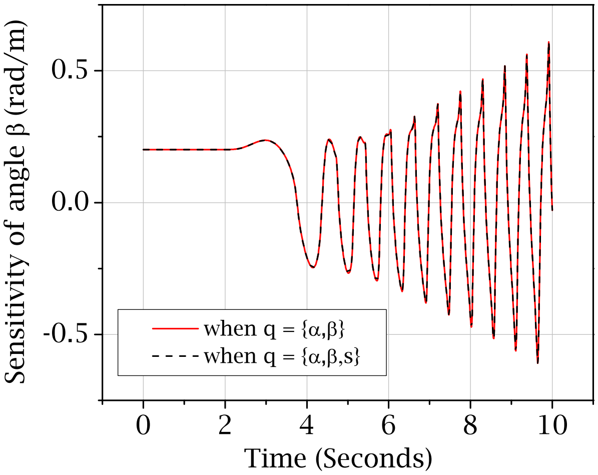


Figure 5.21: Comparison of β_{L_2}

The overall accuracy of the simulation can be validated by using a finite difference approach. The figures 5.22 and 5.23 plot the difference between the solution obtained using a finite difference approach and the solution obtained by solving these generated equations. They are found to be far less than the selected tolerances and thus confirms the validity of the sensitivity results.

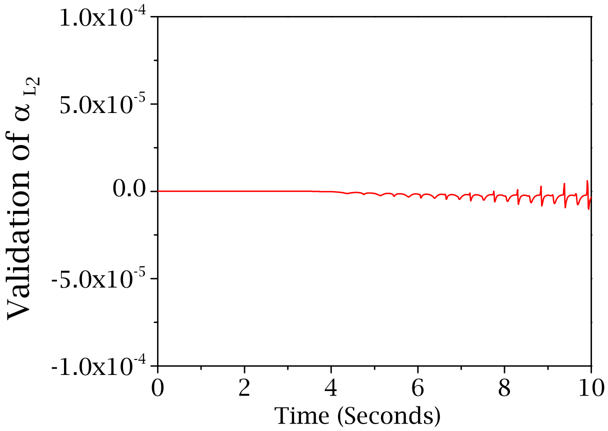


Figure 5.22: Validation of α_{L_2}

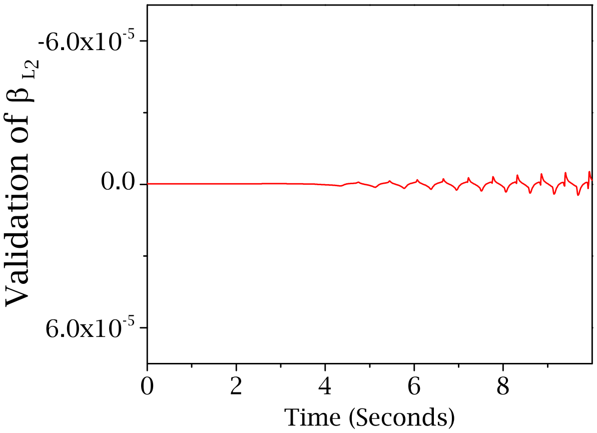


Figure 5.23: Validation of β_{L_2}

In this example, a multi-domain multibody system was used to generate the sensitivity equations using the implemented graph-theoretic sensitivity analysis algorithm. The implementation was demonstrated to be correct for the system and was found to be capable of accepting different choices of tree branches.

5.2.4 A double-wishbone suspension system

As a final example, the model of a double-wishbone suspension system will be considered to demonstrate the capability of the implemented algorithm for a multi-loop kinematic analysis. Figure 5.24 shows the schematic diagram of a typical double wishbone suspension system.

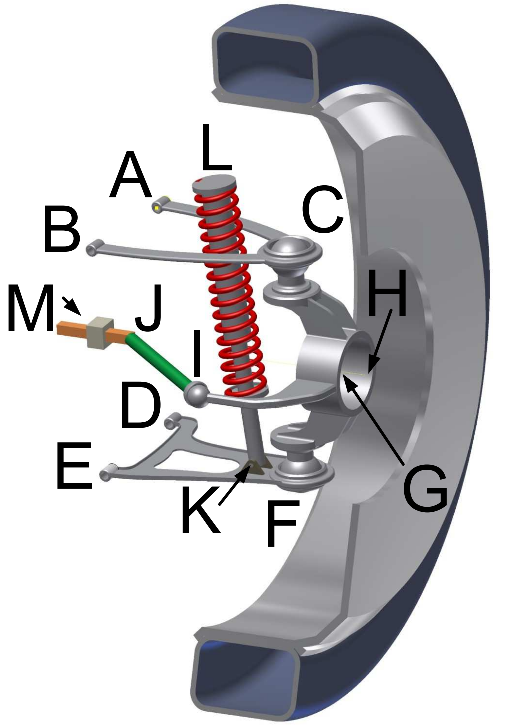


Figure 5.24: Double wishbone suspension system

The double-wishbone suspension system is defined in terms of the hard point locations. A complete list of the hard points are provided in Appendix A.4. In the figure above, M is the steering rack, which, for the purpose of the present analysis, will be assumed to be fixed, link IJ is the tie rod, links ABC and DEF are the upper and lower control arms, and the component $CFIH$ is the hub. The mechanism allows the wheel to move up and down but also forces the wheel to change its orientation while doing so.

The intention here is to study the changes in the orientation of the wheel, which is expressed in terms of the toe, caster and camber angles, and to apply the implemented graph-theoretic sensitivity analysis algorithm to generate the sensitivity equations for the system, so that it can be used to determine the effects of the hard point locations on these quantities. At this point a graph-theoretic description of the mechanism is called for.

A traditional graph-theoretic model of the complete suspension system is too complicated to be included in this thesis. For that reason a schematic graph is shown in the figure below.

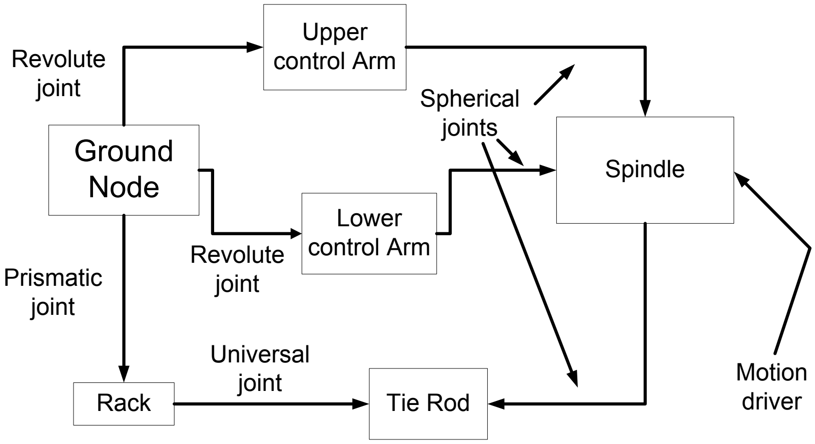


Figure 5.25: Schematic of the graph-theoretic model of the suspension system

For this example, the motion driver will be used to introduce a vertical sinusoidal displacement to the hub. Using the implemented algorithm, both governing and sensitivity equations will be generated for the system and by solving them, the sensitivities of the camber angle will be determined. The results will be validated by a finite difference formulation. At this point a mathematical definition of the camber angle is called for.

The camber angle of the wheel,which is the rotation of the wheel about the longitudinal Z axis, can be calculated from the coordinates of the hard points G and H. The following equation is used to calculate the camber angle.

$$\lambda_{camber} = \tan^{-1}\left(-\frac{H^{y} - G^{y}}{H^{x} - G^{x}}\right) \tag{5.27}$$

where the location of point H and G are functions of time and the set of model parameters $\mathbf{p}$, and are given by their x, y, and z components.

$$\begin{aligned} H(t,\mathbf{p}) &= \left[\begin{array}{ccc} H^{x} & H^{y} & H^{z} \end{array}\right]^{T} \\ G(t,\mathbf{p}) &= \left[\begin{array}{ccc} G^{x} & G^{y} & G^{z} \end{array}\right]^{T} \end{aligned} \tag{5.28}$$

To evaluate the sensitivity of the camber angle, one must differentiate equation (5.27) using chain rule of differentiation.

Equation (5.29) illustrates the expression for the sensitivity of the camber angle with respect to a scalar model parameter $b \in \mathbf{p}$.

$$\frac{d}{db}\left(\lambda_{camber}\right) = -\frac{\frac{H^{y}_{b} - G^{y}_{b}}{H^{x} - G^{x}} - \frac{\left(H^{y} - G^{y}\right)\left(H^{x}_{b} - G^{x}_{b}\right)}{\left(H^{x} - G^{x}\right)^2}}{1 + \frac{\left(H^{y} - G^{y}\right)^2}{\left(H^{x} - G^{x}\right)^2}} \tag{5.29}$$

where the subscripts refer to the parameter with respect to which the differentiation is carried out. Equation (5.30) defines the various terms used in equation (5.29).

$$H^{i}_{b} = \frac{d}{db} H^{i} \text{ and } \; G^{i}_{b} = \frac{d}{db} G^{i} \Bigg|_{i = x,\, y, \text{ and } z} \tag{5.30}$$

The suspension system has only one degree of freedom and due to the application of the motion driver, the system of equations takes the form of a set of nonlinear algebraic equations. By solving these equations at every time step, the state variables of the model can be evaluated, which can be used to determine the required camber angle and its sensitivity.

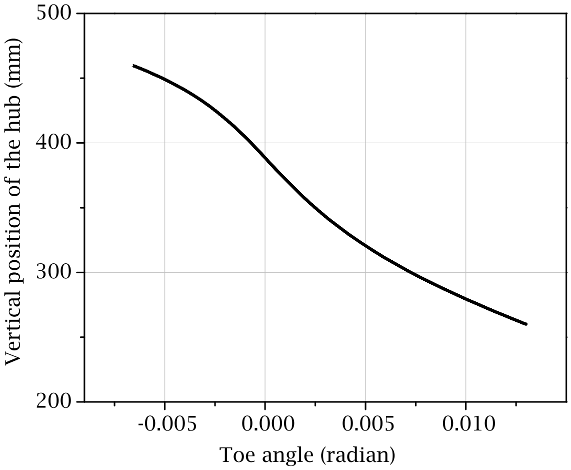


Figure 5.26: Toe angle vs. hub height

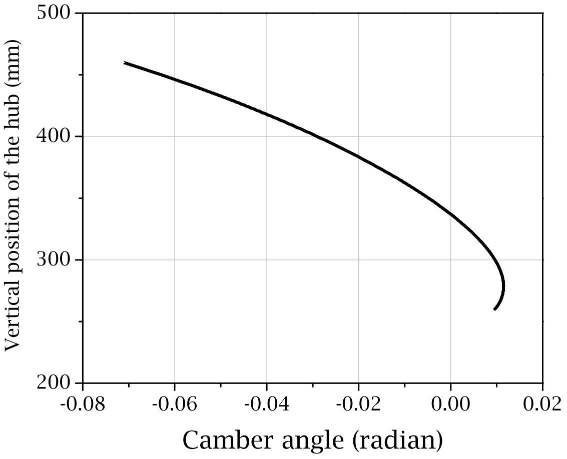


Figure 5.27: Camber angle vs. hub height

This system is modelled using nine coordinates. It includes two coordinates for the universal joint, two joint coordinates for the two control arms, and three translational coordinates for the hub and two extra coordinates to describe the location of the base of the motion actuator. The location of the point J, i.e., the inner connection point of the tie rod is selected as the model parameter for the sensitivity study. The application of the implemented graph-theoretic algorithm generates a set of equations in terms of the nine coordinates and their sensitivities.

To simulate the system, a sinusoidal motion of amplitude of 100 mm is applied to the point H; the steering input to the system is kept constant at zero. Figures 5.27 and 5.26 show the behaviour of the system.

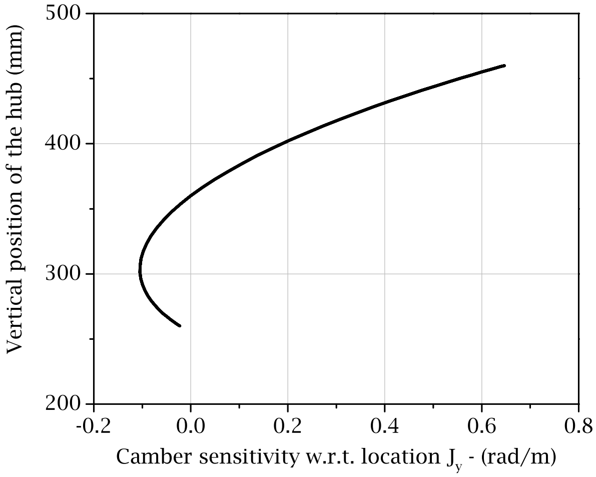


Figure 5.28: Camber sensitivity vs. hub height

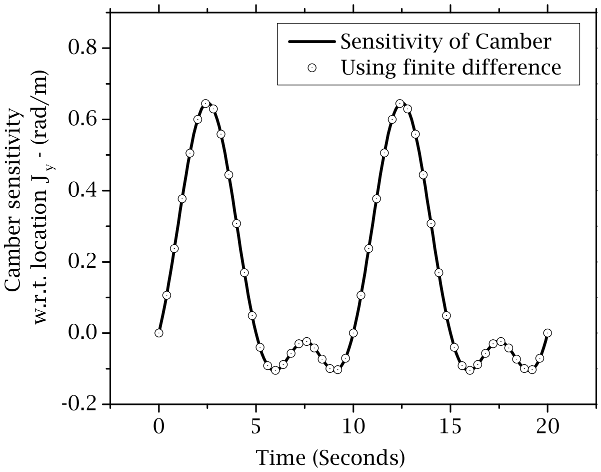


Figure 5.29: Validation of Camber sensitivity

The resulting sensitivity of the camber angle is plotted with the wheel height in figure 5.28. The plot shows that when the hub height is above 360 mm, the camber sensitivity is positive, i.e., any positive change in the y intercept of the point J is expected to result in an increase in camber angle, whereas when the hub height is less than 360 mm the opposite trend is observed. These results are validated using a finite difference formulation with a forward difference scheme, as shown in figure 5.29.

5.3 Comparison of performance

To evaluate the performance of graph-theoretic sensitivity analyses, it is necessary to develop a meaningful comparison scheme for the evaluation. For the purpose of clarity, it is also necessary to explain the alternate methods with which the comparisons are to be made.

Figure 5.30 shows a schematic representation of the various methods that can be used to obtain sensitivity information from a physical system. The central finite difference formulation is conceptually one of the simplest method for evaluation of sensitivity information. However, it involves the solution of the governing equations at three different points in the parameter space for every time step. The second route is the generation of sensitivity equations using direct differentiation of the governing symbolic equations. The generated equations are then solved along with the governing equations. The third route involves the generation of sensitivity equations and the governing equations directly from the linear graph and the subsequent numerical simulation of the same. For all of these methods, the `Maple` routine `dsolve[numeric]` is used for the numerical integration.

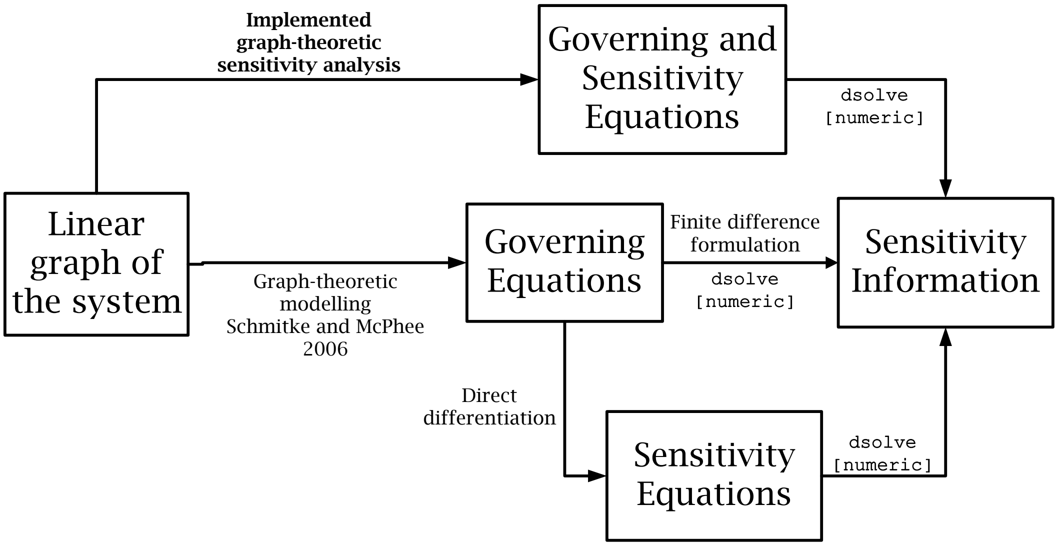

Implemented graph-theoretic sensitivity analysis
Governing and Sensitivity Equations
dsolve [numeric]
Linear graph of the system
Governing Equations
Finite difference formulation
dsolve [numeric]
Sensitivity Information
Graph-theoretic modelling Schmitke and McPhee 2006
Direct differentiation
Sensitivity Equations
dsolve [numeric]

Figure 5.30: Different methods of sensitivity analysis

To compare two sets of equations for better performance, one obvious choice is the time required to integrate them through a certain range. `Maple` does offer a command to measure the CPU time of a process. In this study it has not been considered as a reliable criterion for the comparison since its calculation in `Maple` is dependent on the internal data types used, which can be different for different integration methods.

As an alternative, the total number of numerical operations required by the set of equations are chosen as the basis of comparison. To describe the evaluation process, some elaborations are required on the relevant commands from `Maple`.

5.3.1 Commands for code optimization and computational cost

A list of equations can be optimized by using the command `‘dsolve/numeric/optimize’` available in `Maple`. This command recognizes repeating groups in a list of equations and assigns them to variables, which are then reused throughout the list of equations for a faster simulation. This also enables us to bring different sets of equations into a common platform for comparison. The total number of operations required to solve a set of equations can be evaluated by the `codegen[cost]` command. The example given below will illustrate the application of these commands.

$$\begin{bmatrix} a+b+c \\ 3a+4\sin(b+c) \\ (3b+3c)\tan(a) \end{bmatrix} = 0 \tag{5.31}$$

The `codegen[cost]`command calculates the computation cost for these equations as

$$\text{Cost} = 5 \text{ additions} + 5 \text{ multiplications} + 2 \text{ functions} \tag{5.32}$$

When `‘dsolve/numeric/optimize’` is used on equation (5.31), it readily recognizes that the term $b+c$ is repeated in the set of equations and forms an intermediate variable $t_1 = b+c$. Finally the following set of optimized equations are obtained.

$$\begin{gathered} a+b+c=0 \\ t_1 = b+c \\ 3a+4\sin(t_1)=0 \\ 3\tan(a)\,t_1 = 0 \end{gathered} \tag{5.33}$$

The cost of the optimized set of equations can be readily verified to be less than that in equation (5.32) by the `cost` command.

$$\text{Cost} = 4 \text{ additions} + \text{assignments} + 4 \text{ multiplications} + 2 \text{ functions} \tag{5.34}$$

5.3.2 Results of performance evaluation

Using the commands described above, the implemented graph-theoretic sensitivity analysis algorithm is tested against the alternate methods for the example problems presented in Section 5.2. For every model, the governing equations are optimized and their costs are calculated. Since the finite difference approach requires the solution of governing equations, this cost information is multiplied by three to arrive at its computational cost.

Next, the governing equations are symbolically differentiated to arrive at the set of sensitivity equations using direct differentiation. The combined set of governing and sensitivity equations are then optimized and the computation cost is evaluated.

Finally the MapleSim implementation of the graph-theoretic algorithm is used to generate the set of equations which contains both governing and sensitivity equations. The cost of the optimized version of this set is also recorded.

The three-dimensional pendulum model

For the three-dimensional pendulum model, the spherical joint is chosen to be the branch resulting in three governing equations and three corresponding sensitivity equations. Sensitivity analysis is carried out with respect to the model parameter L and the computational costs of the competing approaches are summarized in Table 5.2.

Table 5.2: Comparison of computation cost: Parameters $\{L\}$

	Functions	$''+''$	$''\times''$
Finite difference :	36	111	237
Direct differentiation :	18	130	270
Graph-theoretic method :	18	126	274

Since the size of the combined set of governing and sensitivity equations depends on the number of parameters being studied, the same procedure is repeated for an increasing number of parameters. The results are presented in Tables 5.3 and 5.4.

Table 5.3: Comparison of computation cost: Parameters $\{L, m\}$

	Functions	$''+''$	$''\times''$
Finite difference :	60	185	395
Direct differentiation :	24	245	482
Graph-theoretic method :	24	250	478

Table 5.4: Comparison of computation cost: Parameters $\{L, m, r\}$

	Functions	$''+''$	$''\times''$
Finite difference :	84	259	553
Direct differentiation :	30	332	628
Graph-theoretic method :	30	351	670

The numbers shown above clearly demonstrate that the computational cost associated with the finite difference method is higher than that for either direct differentiation or the proposed algorithm. For more than one parameters, this computational cost is found to escalate even further.

The direct differentiation approach and the proposed graph-theoretic approach are found to be more or less comparable in terms of the computational cost and this trend is maintained for different numbers of parameters. This behaviour is expected since the basic structure and complexity of the equations produced by the two methods are very similar to each other and in many cases they are identical. For this reason, equations produced in this fashion by these two methods can be considered equally efficient.

The driven slider-crank model

Similar analyses are carried out for the electrically driven slider-crank model. First the branches are chosen in such a way, so that the resulting equations are formed in terms of the state variables $q = \{\alpha, \beta, \alpha_{L_2}, \beta_{L_2}\}$. The table below shows the computational costs associated with the model.

Table 5.5: Comparison of computation cost: $q = \{\alpha, \beta, \alpha_{L_2}, \beta_{L_2}\}$

	Functions	$''+''$	$''\times''$
Finite difference :	27	90	201
Direct differentiation :	14	66	121
Graph-theoretic method :	14	64	113

The same comparison is conducted with a different choice of branches to ensure generation of equations in terms of the joint coordinates and its sensitivities.

Table 5.6: Comparison of computation cost: $q = \{\alpha, \beta, s, \alpha_{L_2}, \beta_{L_2}, s_{L_2}\}$

	Functions	$''+''$	$''\times''$
Finite difference :	33	66	123
Direct differentiation :	18	55	81
Graph-theoretic method :	18	55	86

Once again the counts of mathematical operations, summarized by tables 5.5 and 5.6, identify the finite difference method to be the most computationally expensive approach. They also demonstrate that direct differentiation and the proposed graph-theoretic algorithm are quite similar in performance.

Simple pendulum: Different trees for system and sensitivity equations

To illustrate the effect of branch selection on the performance of the generated equations, the example shown in section 4.2.4 is considered. For this model, the governing equations are generated by selecting the planar joint into the tree. This results in a set of state variables $q = x, y, \theta$ and reaction forces $F = \begin{bmatrix} {}_xF^h & {}_yF^h \end{bmatrix}$ governed by three differential and two algebraic equations.

Sensitivity equations for this model are derived in two separate ways. First, direct differentiation is used to generate the sensitivity equations, which results in a final set of equations in terms of ten unknown quantities.

Next the graph-theoretic method is applied to the model where the planar joint is selected as a branch for the system equations and the revolute joint is selected as a branch for the sensitivity equations. This results in a final set of equations in terms of a total of six unknown quantities. The generated equations are presented in equations (4.69) and (4.70). Table 5.7 summarizes the computational cost of the equations produced by these two methods.

Table 5.7: Comparison of computational cost: Effect of branch selection

	Functions	$''+''$	$''\times''$
Finite difference:	15	21	39
Direct differentiation:	8	20	21
Graph-theoretic method:	7	13	16

Due to the small scale of the model, the effects on the computational costs are not dramatic; however the structure of equation (4.70) and the numbers from table 5.7 show the benefit of the proposed graph-theoretic algorithm for sensitivity analysis. In this case, the graph-theoretic algorithm generates the governing and sensitivity equations in terms of a smaller number of unknown quantities and subsequently, the corresponding computational cost is found to be lower than that of the equations produced by direct differentiation.

The presented analysis reveals that the graph-theoretic method is an automated, accurate, flexible, and algorithmic approach for the derivation of sensitivity equations. At the most basic form, the graph-theoretic method forms equations which have the same complexity as the ones derived by direct differentiation. However, the graph-theoretic approach enables the users to select separate tree structures for the sensitivity equations. This results in generation of simpler sensitivity equations which are more efficient and are generally difficult to derive otherwise.

5.4 Summary

The software implementation of the proposed graph-theoretic sensitivity analysis is presented in this chapter. A brief overview of the implementation process is provided to explain the basic approach toward software deployment and the resulting user interface.

A number of examples are also presented to showcase the application of the software implementation. A sliding pendulum mechanism is considered to explain the intermediate steps of equation generation used by the algorithm. Next a dynamic simulation of a three-dimensional pendulum is presented to demonstrate key issues involving motion spaces of joints. A driven slider-crank mechanism is presented to illustrate the applicability of the algorithm for multi-domain systems. And finally, a double-wishbone suspension system is analyzed to showcase a multibody kinematic simulation.

These examples demonstrate that the current implementation of the proposed graph-theoretic sensitivity analysis algorithm is capable of generating sensitivity equation accurately and automatically. They also show that the algorithm is capable of processing different types of mechanisms and is effective for both dynamic and kinematic simulation of multibody, multi-domain systems.

Chapter 6

Conclusions and future work

In this research, an algorithm that can generate the sensitivity equations from the graph-theoretic description of a system was developed. The algorithm has been shown to be applicable for various types of dynamic systems (hydraulic, electro-chemical, mechanical etc.) and even multi-domain systems. The algorithm follows the typical step by step procedure of a graph-theoretic approach toward modelling and has been successfully implemented using a commercially available modelling and simulation tool called MapleSim. The resulting software module is capable of performing sensitivity analysis for multibody systems created in MapleSim.

The proposed method combines the simplicity of direct differentiation with the flexibility of graph-theoretic formulation. It allows users to select separate tree structures for the governing and the sensitivity equations. This results in greater control over the size and complexity of the resulting equations. The actual differentiation operation is performed at the component level to avoid differentiation of complex expressions.

The presented method has been demonstrated to be automated, flexible, accurate, and numerically stable. The efficiency of the approach has been determined to be more than that of the finite difference approach. In comparison with the direct differentiation approach, the efficiencies of the equations produced by the proposed method were found to be similar. For systems where separate trees were selected for the sensitivity equations, the proposed method was demonstrated to be capable of generating more efficient sets of equations than the ones produced by symbolic differentiation operations.

The significant contributions of this work are summarized in section 6.1; recommendations for future directions on this topic are provided in section 6.2.

6.1 Contributions

The contributions for this research have been divided into four categories as illustrated below.

Graph-theoretic sensitivity analysis of multibody systems

Starting from the work done by Savage and Carr [6, 34] the graph-theoretic approach toward sensitivity analysis was extended to be applied to multibody systems. The salient aspects of multibody systems require special considerations and some modifications to the general methodology. The resulting algorithm for graph-theoretic sensitivity analysis has the following benefits:

Algorithmic: The proposed method uses the same step by step procedures followed by the graph-theoretic methods of system modelling. As such, it is suitable for an automated implementation. This opens the possibility of the integration of the proposed algorithm with MapleSim, which uses graph-theoretic formulations to model multibody systems.

Automated: The MapleSim implementation of the proposed method can generate the symbolic governing equations and the sensitivity equations automatically from the linear graph representing the system. Minimal user intervention is required for the entire process. Especially using the software interface of MapleSim, the sensitivity equations can be generated by simply executing a specially designed Maple worksheet.

Versatile: Although the current software implementation of the algorithm is restricted to multibody systems only, the proposed algorithm can be easily applied to any dynamic system that can be modelled using a linear graph. This has been adequately demonstrated by the example problems presented in this thesis.

Flexible: The graph-theoretic approach allows the user to select a tree for the sensitivity equations, which enables the user to control the size and complexity of the resulting equations. As demonstrated by the examples, it is possible to generate sensitivity equations in terms of a different set of unknowns than those used in the governing equations, something that is very difficult to achieve using direct differentiation.

Efficient: The efficiency of the graph-theoretic approach has been demonstrated to be more than that of the numerical approach using finite difference schemes. In comparison to the direct differentiation approach, the method gives similar results in terms of efficiencies. However, it is to be noted that while direct differentiation is limited to the pre-formed structures of the governing equations, the proposed algorithm can be used to produce more efficient sensitivity equations by careful choice of the tree.

Graph-theoretic modelling of a hydrodynamic torque converter

To test the applicability of the proposed method to multi-domain systems, a graph-theoretic model was developed for a hydrodynamic torque converter. The graph-theoretic framework was developed for a rotodynamic impeller, and the complete model of a torque converter was constructed by assembling these sub-systems. The resulting acausal model was implemented using MapleSim. The model includes the dynamic effects of the mechanical impellers and the hydraulic losses in the transmission fluid.

The implemented model can be easily included into existing models of automotive transmission systems, which can result in a very high fidelity model, one that allows access to the internal assemblies of the subsystems and the physical parameters thereof.

The graph-theoretic approach used to model the torque converter can be readily extended to model other hydro-dynamic machines. The technique of assembling sub-systems can be used to model multi-stage torque converters as well as other hydraulic devices. Similar formulations can be used to develop acausal models of axial and radial flow devices.

Software Implementation

The software implementation of the proposed algorithm was an important part of this research. The objective was to make the application of this method easy and user friendly. It also means that the method can be used by users who are not directly connected to the development efforts.

The modelling and simulation tool MapleSim was used to implement the proposed algorithm. It was selected because it uses graph-theoretic formulation to create governing equations for multibody systems. At present, this implementation is restricted to multibody systems only. However, by using deprecated versions of source codes, it was demonstrated that the MapleSim implementation can be readily extended to include multi-domain systems as well.

The current implementation allows the user to create a model using MapleSim's user interface and attach it with a specially designed Maple worksheet, where the newly added functionalities can be used to generate sensitivity equations.

Graph-theoretic sensitivity analysis of electro-chemical systems

Graph-theoretic sensitivity analysis was performed on a NiMH battery model to identify important model parameters. The sensitivity equations produced by the graph-theoretic method was determined to be more efficient than those produced by the conventional methods like direct differentiation.

6.2 Future Directions

As a result of the research work that has been presented in this thesis, the following topics are identified as possible directions for future research efforts. It is to be noted that the presented research directions and issues mentioned in this section do not necessarily relate to the topic of sensitivity analysis.

Efficiency of the proposed algorithm

As mentioned in chapter 5, the efficiency of the generated set of equations was found to be more or less comparable to the efficiency of the equations generated using direct differentiation. To explain this lack of dramatic improvement of performance, the following issues must be discussed.

Projection onto the ground node: Currently the software implementation is hard-coded to resolve all vector quantities on to the ground frame. This approach is often not a very efficient method for equation generation and can result in repeated multiplications with very complicated expressions in form of composite rotation matrices. In future, the algorithm must be modified so that it can perform efficient projection of vector variables.

CPU time comparison: In this work, the CPU time was not considered as a measure of performance due to some problems regarding the implementation in Maple. It can be worthwhile to compute the actual time of integration by exporting the symbolic equations to a separate software platform that can yield reliable numbers for the time consumption.

Code optimization: To some extent the use of code optimization could be responsible for the observed performance trends. Since the code optimization routine modifies the form of the set of equations, it is possible that some structural information is lost in the process, which might have caused the performances of the two methods seem similar to one another.

Application to very large scale systems

The application of the proposed algorithm in its current form of implementation is not expected to be efficient for very large scale systems due to the hard-coded tendency to revert to the ground frame all the time. Once this issue is fixed, the algorithm can be tested on large system to test its applicability and efficiency.

Application to flexible multibody systems

One very logical extension of the current implementation is the inclusion of flexible multibody systems, which will enable users to analyze a wide variety of practical multibody systems.

Inclusion of multi-domain systems in the software environment

As demonstrated by the multi-domain models that were custom made for this thesis, the graph-theoretic algorithm is equally applicable to all dynamic systems. Currently the software implementation is restricted specifically to multibody systems since the parent software MapleSim uses *Modellica* codes for other domains. It might be useful to extend the capability of the graph-theoretic engine in MapleSim to include some components from the electrical and hydraulic domains.

Graph-theoretic modelling of hydraulic machines

By extending the methods used to model the hydrodynamic torque converter, it is possible to create high-fidelity graph-theoretic model of other hydraulic machines. In particular, the lumped parameter approach toward modelling fluid flow over curved vanes can readily be used to model hydraulic turbines, which can then be integrated with existing models of the electrical generators to assemble very high-fidelity physics-based acausal models of power generation systems.

Using experimental data to compliment the theoretical models, it is also possible to model axial flow devices like hydraulic propellers and Kaplan turbines with various configurations. The use of graph-theoretic approach can greatly improve the fidelity of the models that include these systems.

Graph-theoretic modelling and analyses of distributed systems

It might also be interesting to extend the current techniques of graph-theoretic modelling and sensitivity analysis to include systems that involve compressible flow such as wind turbines and centrifugal pumps.

6.3 Permissions and List of Publications

A summary of the published sections of this thesis is presented in this section.

Parametric importance analysis: The parametric importance analysis on NiMH battery was presented at the SAE World Congress 2011 [74].

Multibody sensitivity analysis: The application of graph-theoretic algorithm for sensitivity analysis on two-dimensional multibody systems was presented at Multibody Dynamics 2011 [76].

Torque converter modelling: The graph-theoretic model of the torque converter was presented at the MathMOD conference 2012 [21].

Design optimization: The results of design optimization on the torque converter model using sensitivity information were presented at the SAE World Congress 2012 [77].

6.4 Final thoughts

Sensitivity analysis for multibody systems is an involved, highly specialized, and fascinating area of research. Due to the ever increasing complexity of the systems being processed, the proverbial “bar” is always set at a higher level. In this research, an attempt has been made to automate the generation of sensitivity equations and to gain some control over the size of the resulting equations. However, from the discussion presented in section 6.2, it is quite clear that automated and efficient graph-theoretic sensitivity analysis remains an open topic of research.

It seems unlikely that just by improving one particular stage of the process, it would be possible to reach the required level of efficiency and accuracy from sensitivity analysis. The prudent approach appears to be to improve the overall efficiency by coming up with better methods for modelling, formulation, and numerical processing in a combined and well-coordinated fashion.

Bibliography

[1] N. Biggs, E. K. Lloyd, and R. J. Wilson, *Graph Theory, 1736-1936.* New York, NY, USA: Clarendon Press, 1986.

[2] H. E. Koenig, Y. Tokad, and H. K. Kesavan, *Analysis of discrete physical systems.* McGraw-Hill (New York), 1967.

[3] H. M. Trent, "Isomorphisms between oriented linear graphs and lumped physical systems," *The Journal of the Acoustical Society of America*, vol. 27, no. 3, pp. 500–527, 1955.

[4] S. Seshu and M. B. Reed, *Linear graphs and electrical networks.* yAddison-Wesle Pub. Co., 1961.

[5] J. J. McPhee, *Dynamics of multi-body systems: conventional and graph-theoretic approaches.* Department of Systems Design Engineering, University of Waterloo, 2000.

[6] G. J. Savage, "Automatic formulation of higher order sensitivity models," *Civil Engineering Systems*, vol. 10, no. 4, pp. 335–350, 1993.

[7] J. J. McPhee, "On the use of linear graph theory in multibody system dynamics," *Nonlinear Dynamics*, vol. 9, pp. 73–90, 1996. 10.1007/BF01833294.

[8] P. N. Sheth and J. J. Uicker, "A computer-aided design analysis system for mechanisms and linkage," *Transactions of ASME, Journal of Engineering for Industry*, vol. 94, pp. 454–464, 1972.

[9] J. Wittenburg, *Dynamics of systems of rigid bodies.* B. G. Teubner, Stuttgart, Germany, 1977.

[10] R. S. Hwang and E. J. Haug, "Topological analysis of multibody systems for recursive dynamics formulations," *Mech Struct Mach*, vol. 17, no. 2, pp. 239–258, 1989.

[11] H.-J. Lai, E. J. Haug, S.-S. Kim, and D.-S. Bae, “A decoupled flexible-relative coordinate recursive approach for flexible multibody dynamics,” *International Journal for Numerical Methods in Engineering*, vol. 32, no. 8, pp. 1669–1689, 1991.

[12] G. Andrews and H. Kesavan, “The vector-network model: A new approach to vector dynamics,” *Mechanism and Machine Theory*, vol. 10, no. 1, pp. 57 – 75, 1975.

[13] T. W. Li and G. C. Andrews, “Application of the vector-network method to constrained mechanical systems,” *Journal of Mechanisms, Transmissions, and Automation in Design*, vol. 108, pp. 471–480, 1986.

[14] G. C. Andrews, M. J. Richard, and R. J. Anderson, “A general vector-network formulation for dynamic systems with kinematic constraints,” *Mechanism and Machine Theory*, vol. 23, no. 3, pp. 243 – 256, 1988.

[15] M. Leger and J. McPhee, “Selection of modeling coordinates for forward dynamic multibody simulations,” *Multibody System Dynamics*, vol. 18, no. 2, pp. 277–297, 2007.

[16] C. Schmitke, K. Morency, and J. McPhee, “Using graph theory and symbolic computing to generate efficient models for multi-body vehicle dynamics,” *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, vol. 222, no. 4, pp. 339–352, 2008.

[17] L. Sass, J. McPhee, C. Schmitke, P. Fisette, and D. Grenier, “A comparison of different methods for modelling electromechanical multibody systems,” *Multibody System Dynamics*, vol. 12, pp. 209–250, 2004.

[18] M. Scherrer and J. McPhee, “Dynamic modelling of electromechanical multibody systems,” *Multibody System Dynamics*, vol. 9, pp. 87–115, 2003.

[19] J. McPhee, C. Schmitke, and S. Redmond, “Dynamic modelling of mechatronic multibody systems with symbolic computing and linear graph theory,” *Mathematical and Computer Modelling of Dynamical Systems*, vol. 10, no. 1, pp. 1–23, 2004.

[20] T. S. Dao and J. McPhee, “Dynamic modeling of electrochemical systems using linear graph theory,” *Journal of Power Sources*, vol. 196, no. 23, pp. 10442 – 10454, 2011.

[21] J. M. Banerjee and J. J. McPhee, “Graph-theoretic modeling and dynamic simulation of an automotive torque converter,” in *Proceedings of the 7th Vienna International Conference on Mathematical Modelling* (I. Troch, ed.), February 2012.

[22] C. Schmitke and J. McPhee, “Using linear graph theory and the principle of orthogonality to model multibody, multi-domain systems,” *Advanced Engineering Informatics*, vol. 22, pp. 147–160, April 2008.

[23] E. E. Leamer, “Let’s take the con out of econometrics, and sensitivity analysis would help,” in *Modeling Economics* (C. Granger, ed.), Oxford: Claredon Press, 1990.

[24] B. Ingalls, “Sensitivity analysis: from model parameters to system behaviour,” *Essays in Biochemistry*, vol. 45, pp. 177–193, 2008.

[25] V. Schwieger, “Sensitivity analysis as a general tool for model optimisation - examples for trajectory estimation,” *Journal of Applied Geodesy*, vol. 1, no. 1, pp. pp 27 – 34, 2007.

[26] R. Serban and E. J. Haug, “Kinematic and kinetic derivatives in multibody system analysis,” *Mechanics of Structures and Machines*, vol. 26, no. 2, pp. 145–173, 1998.

[27] C. H. Bischof, “On the automatic differentiation of computer programs and an application to multibody systems,” in *ICIAM/GAMM 95: Issue 1: Numerical Analysis, Scientific Computing, Computer Science*, pp. 267–272, 1995.

[28] D. Bestle and P. Eberhard, “Analyzing and optimizing multi-body systems,” *Mechanics of Structures and Machines*, vol. 62, pp. 181–190, 1992.

[29] K. S. Anderson and Y. Hsu, “Analytical fully-recursive sensitivity analysis for multibody dynamic chain systems,” *Multibody System Dynamics*, vol. 8, pp. 1–27, 2002. 10.1023/A:1015867515213.

[30] P. Krishnaswami, R. A. Wehage, and E. J. Haug, “Design sensitivity analysis of constrained dynamic systems by direct differentiation,” Tech. Rep. 83-5, Center for Computer-Aided Design, The University of Iowa, Iowa City, Iowa, USA, 1983.

[31] C. O. Chang and P. E. Nikravesh, “Optimal design of mechanical systems with constraint violation stabilization method,” *Journal of Mechanism, Transmissions, and Automation in Design*, vol. 107, pp. 493–498, 1985.

[32] V. J. Pesch, C. L. Hinkle, and D. A. Tortorelli, "Optimization of planar mechanism kinematics with symbolic computation," in *Proceedings of the IUTAM Symposium on Optimization of Mechanical Systems*, pp. 221–230, 1996.

[33] R. Serban and J. Freeman, "Identification and identifiability of unknown parameters in multibody dynamic systems," *Multibody System Dynamics*, vol. 5, pp. 335–350, 2001.

[34] S. Carr and G. J. Savage, "Symbolic sensitivity analysis of nonlinear physical systems using graph theoretic modeling," in *Proceedings of the Maple Summer Workshop and Symposium*, Mathematical computation with Maple V. Ideas and applications., (University of Michigan, Ann Arbor, USA), pp. 118–127, June 28 - 30 1995.

[35] R. Serban and J. S. Freeman, "Direct differentiation methods for the design sensitivity of multi-body dynamic systems," in *Proceedings of the 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, pp. 18–22, 1996.

[36] J. Dias and M. Pereira, "Sensitivity analysis of rigid-flexible multibody systems," *Multibody System Dynamics*, vol. 1, pp. 303–322, 1997.

[37] E. J. Haug, *Design sensitivity analysis of dynamic systems*, vol. F27 of *NATO ASI Series*, pp. 705–755. London, UK: Springer-Verlag, 1987. Editor C. A. Mota Soares.

[38] D. Bestle and J. Seybold, "Sensitivity analysis of constrained multibody systems," *Archive of Applied Mechanics*, vol. 62, pp. 181–190, 1992.

[39] A. Sandu, D. N. Daescu, and G. R. Carmichael, "Direct and adjoint sensitivity analysis of chemical kinetic systems with kpp: Part itheory and software tools," *Atmospheric Environment*, vol. 37, no. 36, pp. 5083 – 5096, 2003.

[40] R. Serban and A. C. Hindmarsh, "Cvodes: An ode solver with sensitivity analysis capabilities," *ACM Transactions on Mathematical Software*, 2003.

[41] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward, "Sundials: Suite of nonlinear and differential/algebraic equation solvers," *ACM Trans. Math. Softw.*, vol. 31, pp. 363–396, September 2005.

[42] L. Petzold, S. Li, Y. Cao, and R. Serban, "Sensitivity analysis of differential-algebraic equations and partial differential equations," *Computers and Chemical Engineering*, vol. 30, no. 1012, pp. 1553 – 1559, 2006.

[43] Y. Cao, S. Li, and L. Petzold, “Adjoint sensitivity analysis for differential-algebraic equations: algorithms and software,” *Journal of Computational and Applied Mathematics*, vol. 149, no. 1, pp. 171 – 191, 2002.

[44] Y. Cao, S. Li, L. Petzold, and R. Serban, “Adjoint sensitivity analysis for differential-algebraic equations: The adjoint dae system and its numerical solution,” *SIAM Journal on Scientific Computing*, vol. 24, no. 3, pp. 1076–1089, 2003.

[45] J.-Y. Ding, Z.-K. Pan, and L.-Q. Chen, “Second-order sensitivity analysis of multibody systems described by differentialz/algebraic equations: adjoint variable approach,” *Int. J. Comput. Math.*, vol. 85, pp. 899–913, June 2008.

[46] R. Serban, “A parallel computational model for sensitivity analysis in optimization for robustness,” *Optimization Methods Software*, vol. 24, pp. 105–121, February 2009.

[47] K. Bhalerao, M. Poursina, and K. Anderson, “An efficient direct differentiation approach for sensitivity analysis of flexible multibody systems,” *Multibody System Dynamics*, vol. 23, pp. 121–140, 2010.

[48] R. Mukherjee, K. Bhalerao, and K. Anderson, “A divide-and-conquer direct differentiation approach for multibody system sensitivity analysis,” *Structural and Multidisciplinary Optimization*, vol. 35, pp. 413–429, 2008.

[49] K. Anderson and Y. Hsu, “Order (n+m) direct differentiation determination of design sensitivity for constrained multibody dynamic systems,” *Structural and Multidisciplinary Optimization*, vol. 26, pp. 171–182, 2004. 10.1007/s00158-003-0336-1.

[50] Y. Hsu and K. Anderson, “Recursive sensitivity analysis for constrained multi-rigid-body dynamic systems design optimization,” *Structural and Multidisciplinary Optimization*, vol. 24, pp. 312–324, 2002.

[51] G. Corliss and Y. F. Chang, “Solving ordinary differential equations using taylor series,” *ACM Trans. Math. Softw.*, vol. 8, pp. 114–144, June 1982.

[52] Y. Chang and G. Corliss, “Atomft: solving odes and daes using taylor series,” *Computers & Mathematics with Applications*, vol. 28, no. 1012, pp. 209 – 233, 1994.

[53] J. R. Cash, "Modified extended backward differentiation formulae for the numerical solution of stiff initial value problems in odes and daes," *J. Comput. Appl. Math.*, vol. 125, pp. 117–130, December 2000.

[54] S. L. Campbell and R. Hollenbeck, "Automatic differentiation and implicit differential equations," in *Computational Differentiation: Techniques, Applications and Tools*, pp. 215–227, 1996.

[55] P. Eberhard and C. Bischof, "Automatic differentiation of numerical integration algorithms," *Mathematics of Computation*, vol. 68, no. 226, pp. pp. 717–731, 1999.

[56] R. Barrio, "Sensitivity analysis of odes/daes using the taylor series method," *SIAM Journal of Scientific Computing*, vol. 27, pp. 1929–1947, December 2005.

[57] A. Griewank, "On automatic differentiation," in *Proceedings of Mathematical Programming: Recent Developments and Applications*, 1989.

[58] J. M. Cabanellas, J. Flez, and C. Vera, "A formulation of the sensitivity analysis for dynamic systems optimization based on pseudo bond graphs," in *Proceedings of the International Conference on Bond Graph Modeling And Simulation* (F. E. Cellier and J. J. Granda, eds.), vol. 27 of *Simulation series*, (Las Vegas, USA), pp. 135–144, Society of Computer SImulation, 1995.

[59] P. J. Gawthrop, "Sensitivity bond graphs," *Journal of the Franklin Institute*, vol. 337, no. 7, pp. 907 – 922, 2000.

[60] P. J. Gawthrop and E. Ronco, "Estimation and control of mechatronic systems using sensitivity bond graphs," *Control Engineering Practice*, vol. 8, no. 11, pp. 1237 – 1248, 2000.

[61] M. Perry, M. A. Atherton, R. A. Bates, and H. P. Wynn, "Bond graph based sensitivity and uncertainty analysis modelling for micro-scale multiphysics robust engineering design," *Journal of the Franklin Institute of Engineering and Applied Mathematics*, vol. 345, no. 3, pp. 282–292, 2008.

[62] D. Hrovat and W. Tobler, "Bond graph modeling and computer simulation of automotive torque converters," *Journal of the Franklin Institute*, vol. 319, no. 12, pp. 93 – 114, 1985.

[63] J. J. McPhee, “Automatic generation of motion equations for planar mechanical systems using the new set of branch coordinates,” *Mechanism and Machine Theory*, vol. 33, no. 6, pp. 805 – 823, 1998.

[64] C. Schmitke and J. McPhee, “Forming equivalent subsystem components to facilitate the modelling of mechatronic multibody systems,” *Multibody System Dynamics*, vol. 14, pp. 81–110, 2005.

[65] J. Y. Wong, *Theory of ground vehicles.* John Wiley & Sons Inc, 2001.

[66] T. Ishihara and R. I. Emori, “Torque converter as a vibrator damper and its transient characteristics,” in *Society of Automotive Engineers*, no. 660368 in Technical papers, 1966.

[67] H. Abidi-Asl, N. L. Azad, and J. McPhee, “Math based modeling and parameteric sensitivity analysis of torque converter performance characteristics,” in *Load Simulation and Analysis in Automotive Engineering*, no. SP-2307 in Battery modeling, SAE International, 2011.

[68] E. D. Smith, F. Szidarovszky, W. J. Karnavas, and A. T. Bahill, “Sensitivity analysis, a powerful system validation technique,” *The Open Cybernetics and Systemics Journal*, vol. 2, pp. 39–56, 2008.

[69] M. Chen and G. A. Rincon-Mora, “Accurate electrical battery model capable of predicting runtime and i-v performance,” *IEEE Transactions on Energy Conversion*, vol. 21, pp. 504–511, June 2006.

[70] Z. M. Salameh, M. A. Casacca, and W. A. Lynch, “A mathematical model for lead-acid batteries,” *IEEE Transactions on Energy Conversion*, vol. 7, no. 1, pp. 93–98, 1992.

[71] A. N. Seaman and J. M, “Symbolic math-based battery modeling for electric vehicle simulation,” in *Proceedings of the ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2010.

[72] J. Newman and K. E. Thomas-Alyea, *Electrochemical Systems.* John Wiley & Sons, 3rd ed., 2004.

[73] B. Wu, M. Mohammed, D. Brigham, R. Elder, and R. E. White, "A non-isothermal model of a nickel-metal hydride cell," *A non-isothermal model of a nickel-metal hydride cell*, vol. 101, pp. 149–157, October 2001.

[74] J. M. Banerjee, T. S. Dao, and J. McPhee, "Mathematical modeling and symbolic sensitivity analysis of ni-mh batteries," in *Advanced Battery Technology SP-2310, 2011*, SAE International, 2011.

[75] J. Banerjee, H. A. asl, N. L. Azad, and J. McPhee, "Parametric importance analysis and design optimization of a torque converter model using sensitivity information," *SAE International Journal of Passenger Cars - Mechanical Systems*, vol. 5, no. 1, pp. 621–638, 2012.

[76] J. M. Banerjee and J. J. McPhee, "Graph-theoretic sensitivity analysis of multibody systems," in *Proceedings of the Multibody Dynamics 2011 ECCOMAS Thematic Conference* (J. C. Samin and P. Fisette, eds.), July 2011.

[77] J. Banerjee, H. Abidi-Asl, N. L. Azad, and J. McPhee, "Parametric importance analysis and design optimization of a torque converter model using sensitivity information," in *Proceedings of the SAE World Congress*, no. 0844 in 12PFL, Society of Automotive Engineers, 2012.

[78] G. G. Lucas and A. Rayner, "Torque converter design calculations," *Automotive Engineer*, vol. Feb, pp. 56–60, 1970.

Appendices

Appendix A

A.1 Torque Converter Parameters

Name	Description	Value	Unit
A	Flow area	0.01	m^2
α_p	Vane blade exit angle in the pump	0	Deg
α_t	Vane blade exit angle in the turbine	-58	Deg
α_s	Vane blade exit angle in the stator	65	Deg
β_p	Vane blade entry angle in the pump	-16	Deg
β_t	Vane blade entry angle in the turbine	48	Deg
β_s	Vane blade entry angle in the stator	-30	Deg
C_{flow}	Friction loss factor	0.2	N/A
C_{shock}	Shock loss factor	1	N/A
J_p	Inertia of pump element	0.106	$Kg \cdot m^2$
J_t	Inertia of turbine element	0.289	$Kg \cdot m^2$
J_s	Inertia of stator element	0.05	$Kg \cdot m^2$
$L_p^f + L_t^f + L_s^f$	Total length of fluid inertia	0.28	m
R_p	Radius of pump element	0.115	m
R_t	Radius of turbine element	0.068	m
R_s	Radius of stator element	0.088	m
ρ	Density of the hydraulic fluid	840	Kg/m^3
S_p	Design constant for pump	0.0003	m^2
S_t	Design constant for turbine	0.001	m^2
S_s	Design constant for stator	0.001	m^2

Table A.1: Torque converter parameters

A.2 Model parameters for the slider crank mechanism

Table A.2: Model parameters for the slider crank mechanism

Variable	Description	Value	Unit
K_v	Transducer coefficient	0.07	
K_t	Transducer coefficient	0.07	
L	Motor coil inductance	0.35	mH
L_1	Length of crank	0.5	m
L_2	Length of connecting rod	1.5	m
m_1	Mass of crank	0.3	Kg
m_2	Mass of connecting rod	0.6	Kg
m_3	Mass of sliding block	0.5	Kg
R	Motor coil resistance	0.85	Ω
r_1	Cross-section radius of crank	0.03	m
r_2	Cross-section radius of connecting rod	0.03	m
ξ	Damping coefficient	1.2	$N-s/m$

A.3 Model parameters for the sliding pendulum

Table A.3: Sliding pendulum parameters

Parameter	Description	Value	Unit
g	Acceleration due to gravity	9.81	m/s^2
k	Spring constant	50.00	N/m
L	Length of the pendulum	1.00	m
m_1	Mass of the slider block	1.00	Kg
m_2	Mass of the pendulum	5.00	Kg
r	Radius of cross-section of the pendulum	0.03	m
s_0	Unstretched length of the slider block	0.50	m
ξ	Rotational damping coefficient	10.00	$N-m-s/rad$

A.4 Hard point locations for the double-wishbone suspension system

Table A.4: Nominal locations of the hard points in the suspension system

Hard point	X (mm)	Y (mm)	Z (mm)
A	360.0	479.4	-86.2
B	360.0	500.9	52.1
C	540.7	498.0	-18.3
D	352.0	265.4	-77.8
E	352.0	230.8	119.2
F	615.5	229.0	17.3
G	579.0	359.5	-0.1
H	635.0	360.0	0.0
I	579.0	359.5	149.9
J	297.5	368.1	40.0
K	543.5	250.0	14.6
L	340.5	632.0	-37.5
M	0.0	368.1	40.0

A.5 Parameters for the 3-dimensional pendulum

Table A.5: Model parameters for the pendulum

Parameter	Description	Value	Unit
G	Acceleration due to gravity	9.81	m/s^2
m	Mass of the pendulum	2	Kg
L	Length of the pendulum	1	m
r	Radius of circular cross-section of the pendulum	0.03	m

A.6 NiMH Battery Parameters

Table A.6: Model parameters for the NiMH battery

Description	**Symbols**	**Value**	**Unit**
Density of nickel active material	$\rho_{\mathrm{Ni(OH)_2}}$	3.4	g/cm^3
Density of metal hydride	ρ_{MH}	7.49	(g/cm^3)
Internal resistance	R_{int}	5.0×10^{-3}	Ω
Heat capacity of the cell	c_p	30	J/gK
Conductivity of cell container	k	2.0×10^{-3}	$W/K\,cm$
Loading of nickel active material	$L_{\mathrm{Ni(OH)_2}}$	6.8×10^{-2}	g/cm^2
Loading of metal hydride	L_{MH}	1.13×10^{-1}	g/cm^2
Mass of cell	m_{cell}	67	g
Thickness of the container wall	e	0.1	cm
Gas volume inside the cell	V_{gas}	1.0×10^{-1}	cm^3
Specific electrode area of cathode	a_{pos}	4.00×10^3	cm^2/cm^3
Specific electrode area of anode	a_{neg}	3.00×10^3	cm^2/cm^3
Surface area of cathode	A_{pos}	3.25×10^2	cm^2
Surface area of anode	A_{neg}	3.60×10^2	cm^2
Thickness of positive electrode	l_{pos}	3.3×10^{-2}	cm
Thickness of negative electrode	l_{neg}	2.8×10^{-2}	cm
Concentration of KOH electrolyte	c_e	7.00×10^{-3}	mol/cm^3
Reference electrolyte concentration	$c_{\mathrm{e,ref}}$	1.00×10^{-3}	mol/cm^3
Reversible heat	$\partial U_1/\partial T$	-1.35×10^{-3}	V/K
	$\partial U_2/\partial T$	-1.68×10^{-3}	V/K
	$\partial U_3/\partial T$	-1.55×10^{-3}	V/K
	$\partial U_4/\partial T$	-1.68×10^{-3}	V/K

Table A.7: Model parameters for the NiMH battery - Continued

Description	Symbols	Value	Unit
Maximum concentration of nickel hydroxide in nickel active material	$c_{\mathrm{H^+,max}}$	3.70×10^{-2}	mol/cm^3
Maximum concentration of hydrogen in metal hydride material	$c_{\mathrm{MH,max}}$	1.00×10^{-1}	mol/cm^3
Reference concentration of nickel hydroxide in active material	$c_{\mathrm{H^+,ref}}$	$0.50 \times c_{\mathrm{H^+,max}}$	
Reference concentration of hydrogen in metal hydride material	$c_{\mathrm{MH,ref}}$	$0.50 \times c_{\mathrm{MH,max}}$	
Open-circuit voltages	$U_{1,c}, U_{1,d}$	$0.527, 0.458$	V
	U_2	0.4011	V
	U_3	-0.8279	V
	U_4	0.4011	V
Exchange current densities at reference reactant concentrations	$i_{0,1,ref}$	1×10^{-4}	Amp/cm^2
	$i_{0,2,ref}$	2×10^{-11}	Amp/cm^2
	$i_{0,3,ref}$	1×10^{-4}	Amp/cm^2
	$i_{0,4,ref}$	1×10^{-4}	Amp/cm^2

Appendix B

B.1 Matrix differentiation

Statement: If a matrix $\mathbf{X}^{m\times n}$ is differentiated with respect to a scalar b, then the result is a matrix $\mathbf{X}_b$, whose elements are obtained by differentiating the elements of matrix $\mathbf{X}$ with respect to b.

Reason: Matrices offer a convenient method of writing equations in a compact form. Naturally, matrix calculus is defined in way so that it continues to serve the purpose of being a compact form for large systems of equations. To illustrate the validity of the aforementioned statement and the merit of defining matrix differentiation in that fashion, one can consider the example of a matrix equation expressed as

$$\mathbf{X}^{m\times n}\,\mathbf{A}^{n\times 1} = 0 \tag{B.1}$$

The individual equations represented by this matrix notation are given below.

$$\left(\sum_{j=1..n} \mathbf{X}_{i,j}\mathbf{A}_j = 0\right)_{i=1..m} \tag{B.2}$$

By differentiating the equations given by (B.2) with respect to the scalar b, the following equations are obtained.

$$\left(\sum_{j=1..n} \frac{\partial \mathbf{X}_{i,j}}{\partial b}\mathbf{A}_j + \sum_{j=1..n} \mathbf{X}_{i,j}\frac{\partial \mathbf{A}_j}{\partial b} = 0\right)_{i=1..m} \tag{B.3}$$

This expression can be written in a compact matrix form by properly defining the process of differentiation of a matrix. The most obvious and convenient way is to write this equation as shown in equation (B.4).

$$\mathbf{X}_b\mathbf{A} + \mathbf{X}\mathbf{A}_b = 0 \tag{B.4}$$

The above equation uses a matrix $\mathbf{X}_b$ as the derivative of $\mathbf{X}$.

$$\frac{d}{db}\left(\mathbf{X}\right) = \mathbf{X}_b \tag{B.5}$$

Thus the correct way of defining the derivative of a matrix with respect to a scalar is to differentiate every element of the matrix with respect to the scalar. **Q.E.D**

B.2 Differentiating a rotation matrix

Statement: If a rotation matrix $\mathbf{R}$ is differentiated with respect to a scalar b, the 5result $\mathbf{R}_b$ can be expressed as the product of a skew symmetric matrix and the rotation matrix $\mathbf{R}$ itself. In other words

$$\begin{gathered}\frac{d}{db}\left(\mathbf{R}\right) = \mathbf{R}_b = \mathbf{S}\mathbf{R}\\ \mathbf{S}^T = -\mathbf{S}\end{gathered} \tag{B.6}$$

Reason: To prove this statement, one can consider a time varying vector $\mathbf{r}$ which is obtained by multiplying a rotation matrix $\mathbf{R}$ with a constant vector $\mathbf{r_0}$.

$$\mathbf{r}(t,b) = \mathbf{R}\left(t,b\right)\left\{\mathbf{r}_0\right\} \tag{B.7}$$

If vector $\mathbf{r}$ is differentiated with respect to a scalar quantity b, the derivative will be a vector that will mathematically represent the change in the position of the tip of the vector $\mathbf{r}$. By symbolic differentiation, the expression of that vector can be obtained as

$$\begin{gathered}\frac{d}{db}\mathbf{r}(t,b) = \mathbf{r}_b = \frac{d}{db}\mathbf{R}\left(t,b\right)\left\{\mathbf{r}_0\right\} = \frac{d\mathbf{R}}{db}\mathbf{R}^T\mathbf{r} = \mathbf{S}\mathbf{r}\\ \mathbf{S} = \frac{d\mathbf{R}}{db}\mathbf{R}^T\end{gathered} \tag{B.8}$$

Since the effect of $\mathbf{R}$ is to rotate the vector $\mathbf{r}_0$, any change in the value of b results in a change in the orientation of the vector. The change vector $\mathbf{r}_b$, is therefore always perpendicular to the original vector $\mathbf{r}$. From equation (B.8) it can be clearly seen that the vector $\mathbf{r}_b$ is obtained by multiplying a matrix $\mathbf{S}$ with the vector $\mathbf{r}$. From this, it is possible to infer that the matrix **S** is indeed a skew symmetric matrix, because its the only type of matrix that when multiplied with a vector, results in a vector perpendicular to the original vector. **Q.E.D**

B.3 Rotational branch transformations

Statement

The rotational sensitivity branch transformation equations can be simplified considerably using the properties of the rotation matrices and its sensitivities. As an example the following equations can be proved for the simple pendulum illustrated in figure 4.1.

Given,

$$\mathbf{R}_b^m = -(\mathbf{R}^r)^T\mathbf{R}^h\left[\left(\mathbf{R}_b^h\right)^T\mathbf{R}^r\mathbf{R}^m + \left(\mathbf{R}^h\right)^T\mathbf{R}_b^r\mathbf{R}^m\right] \tag{B.9}$$

It can be proved that

$$\mathbf{R}_b^m = (\mathbf{R}_b^r)^T\mathbf{R}^h + (\mathbf{R}^r)^T\mathbf{R}_b^h \tag{B.10}$$

Proof

The rotational circuit equation for the system states

$$\left(\mathbf{R}^h\right)^T\mathbf{R}^r\,\mathbf{R}^m = \mathbf{U}^{3\times 3} \implies \mathbf{R}^h = \mathbf{R}^r\mathbf{R}^m \tag{B.11}$$

Also, according to the discussion in appendix B.2, the following equations can be defined, where $\mathbf{S}_h$ and $\mathbf{S}_r$ are skew symmetric matrices.

$$\mathbf{R}_b^h = \mathbf{S}_h\mathbf{R}^h \text{ and } \mathbf{R}_b^r = \mathbf{S}_r\mathbf{R}^r \tag{B.12}$$

By substituting equation (B.12) into equation (B.9) the following equation is obtained.

$$\begin{aligned}\mathbf{R}_b^m &= \Big(-(\mathbf{R}^r)^T\mathbf{R}^h\left(\mathbf{S}_1\mathbf{R}^h\right)^T\mathbf{R}^r\mathbf{R}^m - (\mathbf{R}^r)^T\mathbf{R}^h\left(\mathbf{R}^h\right)^T\mathbf{S}_2\mathbf{R}^r\mathbf{R}^m\Big) \\ &= \Big((\mathbf{R}^r)^T\left(\mathbf{S}_1\mathbf{R}^h\right) + (\mathbf{R}^r)^T(\mathbf{S}_2)^T\mathbf{R}^h\Big)\end{aligned} \tag{B.13}$$

By rearranging the matrices and using equation (B.12) the following equation is obtain which proves the proposition.

$$\mathbf{R}_b^m = (\mathbf{R}^r)^T \mathbf{R}_b^h + (S_2 \mathbf{R}^r)^T \mathbf{R}^h = (\mathbf{R}^r)^T \mathbf{R}_b^h + (\mathbf{R}_b^r)^T \mathbf{R}^h \tag{B.14}$$

As an added check to this result, it should be noted that this expression can also be obtained by differentiating (4.22) with respect to the model parameter b.

B.4 Derivation of the constitutive equations for the rotodynamic impeller

A simplified impeller profile is shown in figure B.1. The flow path of the hydraulic fluid over the impeller profile is shown in figure B.2. The control volume (C.V) is the cylindrical annular area shown in figure B.1.

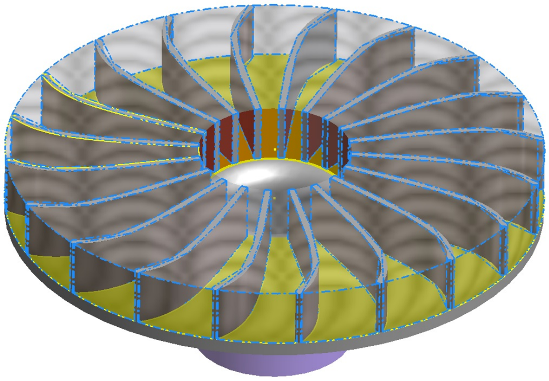


Figure B.1: Simplified impeller profile

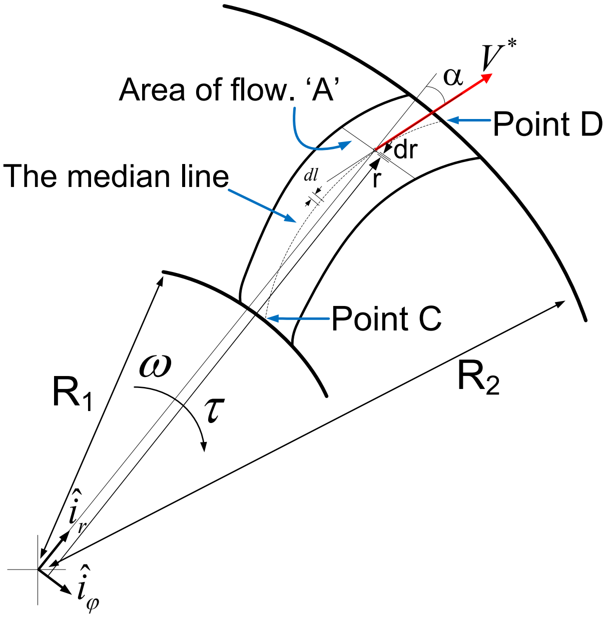


Figure B.2: Flow path over the impeller

The objective of this derivation is to establish the mathematical connection between the the net torque acting on the C.V τ, the angular velocity of the impeller ω, the flow rate of the fluid Q, and the pressure difference between the points C and D. The derivation presented here is based on the work done by Hrovat and Tobler [62].

The net torque acting on the C.V is given by the moment of momentum equation.

$$\begin{array}{c}\text{Net}\\\text{Torque}\end{array} = \begin{bmatrix}\text{Flow rate of}\\\text{angular momentum}\\\text{through control surface}\end{bmatrix} + \begin{bmatrix}\text{Time rate of change of}\\\text{angular momentum}\\\text{in the control volume}\end{bmatrix} \tag{B.15}$$

For this system and the particular choice of the control volume, the moment of momentum equation can be written as

$$\tau - J_m\dot{\omega} = \iint\limits_{CS} (\bar{r} \times \bar{V})\rho\left(\bar{V} \cdot d\bar{A}\right) + \frac{\partial}{\partial t}\iiint\limits_{CV} \rho\left(\bar{r} \times \bar{V}\right) d\forall \tag{B.16}$$

J_m Polar moment of inertia of the impeller

CV The annular cylindrical shaped control volume shown in figure 3.8

CS The corresponding control surface

ρ Density of the hydraulic fluid

$\bar{V}$ Absolute velocity of a fluid particle

A Flow area (assumed to be constant along the flow path)

$d\forall$ Differential volume. $d\forall = Adl$

The volumetric flow rate Q can be expressed in terms of the flow velocity as

$$Q = \iint\limits_{CS} V dA \tag{B.17}$$

The flow velocity of a fluid particle measured with respect to the impeller at a radial location at r with thickness dr is denoted as V^*. In terms of the unit vectors, the expression for V^* can be written as

$$\bar{V}^* = \frac{Q}{A}\hat{i}_r + \left(\frac{Q}{A}\tan\alpha\right)\hat{i}_\varphi \tag{B.18}$$

The absolute velocity of a fluid particle can be written as

$$\bar{V} = \bar{\omega} \times \bar{r} + \bar{V}^* = \frac{Q}{A}\hat{i}_r + \left(r\omega + \frac{Q}{A}\tan\alpha\right)\hat{i}_\varphi \tag{B.19}$$

Equation (B.19) is substituted into equation (B.16) and the two integrals in the right hand side are considered separately.
For the surface integral

$$\begin{aligned}
&\iint\limits_{CS} \left(r \times \tilde{V}\right) \rho \left(\tilde{V} \cdot dA\right) \\
&= \iint\limits_{\text{outlet surface}} \rho \left(r \times \tilde{V}\right) \frac{Q}{A} dA - \iint\limits_{\text{inlet surface}} \rho \left(r \times \tilde{V}\right) \frac{Q}{A} dA \\
&= \rho Q \left(\omega R_2^2 + R_2 \frac{Q}{A} \tan \alpha_e - \omega_{back} R_1^2 - R_1 \frac{Q}{A} \tan \alpha_e^{back}\right) \hat{i}_z \\
&= \psi \, Q \, \hat{i}_z
\end{aligned} \tag{B.20}$$

For the volume integral

$$\begin{aligned}
&\frac{\partial}{\partial t} \iiint\limits_{CV} \rho \left(r \times \tilde{V}\right) d\forall \\
&= \frac{\partial}{\partial t} \iiint\limits_{CV} \rho \left(r \hat{i}_r \times \left[\frac{Q}{A} \hat{i}_r + \left(r\omega + \frac{Q}{A} \tan \alpha\right) \hat{i}_\varphi\right]\right) d\forall \\
&= \left[\frac{\partial \omega}{\partial t} \iiint\limits_{CV} r^2 \rho d\forall + \rho \frac{\partial}{\partial t} \int\limits_{Streamline} r \left(\tan \alpha\right) \frac{Q}{A} A dl\right] \hat{i}_z \\
&= \left(J_f \dot{\omega} + \rho \dot{Q} S\right) \hat{i}_z
\end{aligned} \tag{B.21}$$

In the above equations, an intermediate variable ψ is introduced:

$$\psi = \rho \left(\omega R_2^2 + R_2 \frac{Q}{A} \tan \alpha_e - \omega_{back} R_1^2 - R_1 \frac{Q}{A} \tan \alpha_e^{back}\right) \tag{B.22}$$

The descriptions of the symbols used in equations (B.20) and (B.21) are given below.

α_e**:** Vane angle at point D

α_e^{back}**:** Angle of the vane leading to point C

ω_{back}**:** Angular velocity of vanes (not the impeller) before point C

J_f**:** Polar moment of inertia of the fluid mass, $J_f = \iiint\limits_{\forall_{total}} \rho \, r^2 d\forall$

S**:** Vane profile parameter, $\quad S = \int r \tan\alpha \, dl$

Equations (B.20) and (B.21) are combined together and are substituted into equation (B.16), to obtain the following equation.

$$\tau = J_m\dot{\omega} + J_f\dot{\omega} + \psi Q + \rho\, S\, \dot{Q} \tag{B.23}$$

This way the mechanical through variable τ is expressed as a function of the mechanical across variable $\dot{\omega}$ and the hydraulic through variables Q and $\dot{Q}$.

To derive the second constitutive equation, the power balance equation is applied to the system.

$$\begin{matrix}\text{Input}\\ \text{Power}\end{matrix} = \begin{bmatrix}\text{Power}\\ \text{stored}\end{bmatrix} + \begin{bmatrix}\text{Rate of work}\\ \text{done against}\\ \text{pressure gradient}\end{bmatrix} + \begin{bmatrix}\text{Lost power}\\ \text{due to friction}\\ \text{and shock effect}\end{bmatrix} \tag{B.24}$$

Using loss terms described by Hrovat [62] the power balance equation for this system is formulated.

$$\bar{\tau} \cdot \bar{\omega} = \dot{E} + (p_D - p_C)\, Q + \overbrace{P_{shock} + P_{flow}}^{\text{Lost Power}} \tag{B.25}$$

The loss term P_{flow} refers to the power lost due to shear stresses in the boundary layer and possible pressure drags. It is modelled using a semi-empirical equation described by Hrovat [62] and Lucas [78].

$$P_{flow} = \frac{1}{2}\rho\, C_{flow}\, |Q| \left(|\bar{\zeta}|\right)^2 \tag{B.26}$$

The constant C_{flow} is a dimensionless and experimentally determined quantity. The expression for $\bar{\zeta}$ is obtained by substituting the exit vane angle α_e in equation (B.18).

$$|\bar{\zeta}| = \frac{Q}{A} \sec(\alpha_e) \tag{B.27}$$

The loss term P_{shock} refers to the power lost due to non-ideal flow conditions at the interface leading to the inlet of the impeller. It is modelled using empirical equations [62].

$$P_{shock} = \frac{1}{2}\rho\, C_{shock}\, |Q|\, (V_{shock})^2 \tag{B.28}$$

The shock velocity V_{shock} is the difference between the tangential component of the absolute velocity of the fluid particles at the junction of the vane and the part leading up to it at point C. In terms of β_e (the vane angle at the point C) and α_e^{back} (the vane angle leading to point C), the shock velocity is given as

$$V_{shock} = R_1\left(\omega_{back} - \omega\right) + \frac{Q}{A}\left(\tan\alpha_e^{back} - \tan\beta_e\right) \tag{B.29}$$

The quantity $\dot{E}$ represents the rate of change of energy of the mechanical components and the hydraulic fluid in the system. The system's mechanical energy can be written in terms of the angular velocity of the impeller and the absolute velocity of the fluid particles as shown in equation (B.30).

$$\mathrm{E} = \frac{1}{2}J_m\,\omega^2 + \frac{\rho}{2}\iiint\limits_{\forall_{total}}\left(\left|\bar{V}\right|\right)^2 d\forall \tag{B.30}$$

Using equation (B.19) the following expression is obtained.

$$\begin{aligned} E &= \frac{1}{2}J_m\,\omega^2 + \frac{\rho}{2}\iiint\limits_{\forall_{total}}\left(\omega^2\,r^2 + \frac{Q^2}{A^2}\sec^2\alpha + 2\frac{Q}{A}\omega\,r\tan\alpha\right)d\forall \\ &= \frac{1}{2}J_m\,\omega^2 + \frac{1}{2}J_f\,\omega^2 + \frac{\rho Q^2}{2A}L^f + \rho Q\omega S \end{aligned} \tag{B.31}$$

where L^f is a vane profile constant given by

$$L^f = \int\limits_{Streamline}\sec^2\alpha\,dl \tag{B.32}$$

Differentiating equation (B.31) with respect to time and substituting the results in equation (B.25) the second constitutive equation of the system is obtained.

$$\left(p_C - p_D\right) = -\psi\,\omega + \frac{\rho L^f}{A}\dot{Q} + \rho\,S\,\dot{\omega} + \frac{P_{shock}}{Q} + \frac{P_{flow}}{Q} \tag{B.33}$$

Thus, the hydraulic across variable of the system, i.e., the pressure difference between the points C and D, is expressed as a function of the mechanical across variables ω, $\dot{\omega}$, the hydraulic through variable $\dot{Q}$, and the model parameters.

B.5 Normalized relative sensitivity

In any physical model, the set of model parameters contains values with different units. As a result, the mathematical concept of a unit change in the value of the parameter has different physical interpretations for different parameters. To ensure a proper level ground for comparison, Smith et al. [68] has used what is known as the relative sensitivity score. In their approach, the absolute sensitivity information is converted to a relative sensitivity measure using the expression shown in equation (B.34).

$$\mu_1 = \frac{\partial \psi}{\partial\, b} \frac{b^*}{\psi\,(b^*)} \tag{B.34}$$

In equation (B.34), μ_1 is the normalized sensitivity score, ψ is the objective function, b^* is the nominal value of the model parameter, and $\psi(b^*)$ is the value of the objective function at the nominal value of the model parameter.

$$\mu_2 = \frac{\partial \psi}{\partial\, b} \left| \frac{b^*}{\psi\,(b^*)} \right| \tag{B.35}$$

However, in this thesis the expression is modified slightly and an absolute value operator is used to calculate the relative sensitivity score as shown in equation (B.35). In this section, a brief justification of this choice of expression will be presented.

To illustrate the physical interpretations of the terms involved in equations (B.34) and (B.35), we consider the curves shown in figure B.3. In this figure, different values of the objective function and model parameters are shown at points A to H.

For each of these points, terms shown in equations (B.34) and (B.35) can be readily identified. The quantity b^* is the X-coordinate, $\psi(b^*)$ is the Y-coordinate, and $\frac{\partial \psi}{\partial\, b}$ is the gradient of the tangent line.

The relative sensitivity is a measure of the fractional change in the objective function for a fractional change in the model parameter. If the expression shown in (B.34) is used, then the notion of "*change*" becomes a function of the sign of the model parameters and the objective functions, which makes the physical interpretation of the quantity μ_1 dependent on the quadrant in which the operating point is located.

In the first quadrant, where both X and Y values are positive, μ_1 indicates a fractional *increase* in the objective function for a fractional *increase* in the model parameter. But in the second quadrant where only the parameter value is negative, μ_1 represents the fractional increase in the objective function for a fractional *decrease* in the model parameter, something completely different from the earlier interpretation.

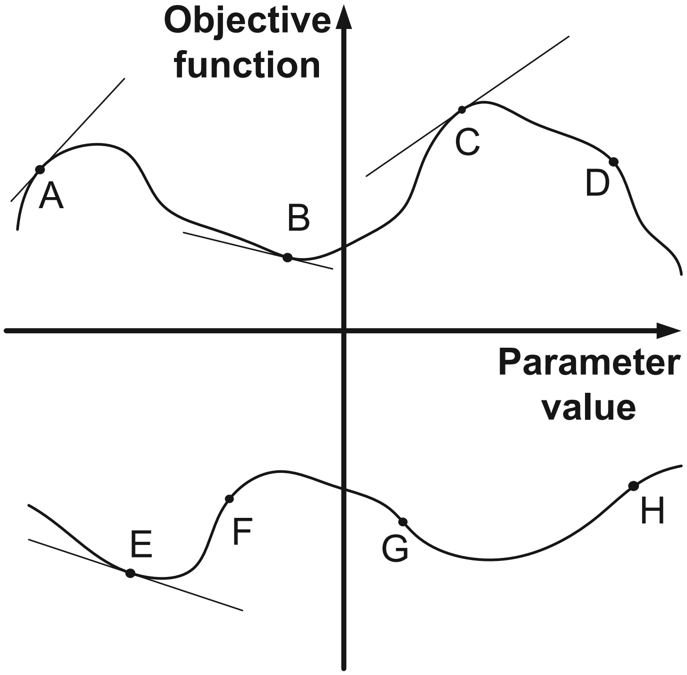


Figure B.3: Different values of objective function and model parameters

The stipulation presented above can be readily verified by looking at points C, A, and B. At point C, μ_1 is positive, which is completely in congruence with our expectations. At point A, the value of μ_1 is negative, which makes sense only if we consider the altered interpretation presented in the previous paragraph. A similar situation is observed at point B ($\mu_1 > 0$), which can be explained by the same interpretation as in case of point A. This dependence on the location of the operating point makes the equation (B.34) a particularly unsuitable method for an importance analysis.

In comparison, the quantity μ_2 always denotes the fractional *increase* in the objective function for an fractional *increase* in the model parameter. This can be shown at points C, A, F, and H where μ_2 is positive and also at points B, D, E, and G where μ_2 is negative.

In other words, the use of the absolute value sign makes the physical interpretation of μ_2 independent of the location of the operating point, which is a very desirable feature while comparing the effects of different model parameters. For this reason, the expression shown in equation (B.35) is used throughout the course of this research.