NUMERICAL METHODS FOR SIMULATION AND OPTIMIZATION OF MECHANICAL SYSTEMS

by

Dan I. Coroian

<u>An Abstract</u>

Of a thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Mathematics in the
Graduate College of The
University of Iowa

August 1997

Thesis supervisor: Professor Florian A. Potra

ABSTRACT

The equations of motion of constrained mechanical systems, often called the Euler-Lagrange equations, are mixed differential-algebraic equations of index 3, and their numerical solution is substantially more difficult than the numerical solution of ordinary differential equations. A numerical method for solving the overdetermined differential-algebraic equations that arise in multi-body dynamics is proposed. The method is based on Newton-type iterations using outer inverses. Some previously proposed methods are shown to be special cases of this method. A method for obtaining outer inverses of any rectangular matrix is described. A numerical algorithm for variable fidelity differential-algebraic equation model corelation is presented, and numerical results are discussed. In the context of workspace analysis for mechanisms and manipulators, a numerical method for obtaining all tangent vectors for a class of multiple bifurcation points that occur when using continuation methods is developed. Also, an existence and uniqueness result for a class of delay differential-algebraic equations is proven.

Abstract approved: ______________________________
Thesis supervisor

____
Title and department

____
Date

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

PH.D. THESIS

This is to certify that the Ph.D. thesis of

Dan I. Coroian

has been approved by the Examining Committee for the thesis requirement for the Doctor of Philosophy degree in Mathematics at the August 1997 graduation.

Thesis committee: ______________________
Thesis supervisor

____
Member

____
Member

____
Member

____
Member

NUMERICAL METHODS FOR SIMULATION AND OPTIMIZATION OF MECHANICAL SYSTEMS

by

Dan I. Coroian

A thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Mathematics in the
Graduate College of The
University of Iowa

August 1997

Thesis supervisor: Professor Florian A. Potra

To my parents,
Iulian and Felicia Coroian

ACKNOWLEDGEMENTS

My deepest respect, gratitude and appreciation to my advisor and mentor, Professor Florian A. Potra, for his many years of patience, encouragement, thoughtful suggestions, and critical analysis, which helped me during the preparation of this thesis.

I would also like to give special thanks and appreciation to Professor Edward J. Haug, who introduced and guided me through new areas of research, and whose competent advice helped me shape this dissertation. Additional appreciation to Professor Kendall E. Atkinson for his continuous guidance and support during my graduate years.

I am grateful to my family, especially to my mother and father, Felicia and Iulian Coroian, and to my sister Lia and brother-in-law Boris for their love, support, encouragement, and confidence.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

There has been a great deal of work, in recent years, in developing faster and more reliable numerical methods for the simulation and optimization of constrained mechanical systems. The equations of motion for these systems, often called the Euler-Lagrange equations, are mixed differential-algebraic equations of index 3. Their numerical solution has proven to be substantially more difficult than the numerical solution of ordinary differential equations.

Standard numerical integrators for ordinary differential equations, such as linear multi-step methods, can be applied to solve the equations of constrained motion since, in principle, any differential-algebraic equation can be reduced to an ordinary differential equation by repeated differentiation of the constraints. However, the numerical solution obtained no longer satisfies the kinematic constraints, whose preservation is essential for realistic simulation.

In a series of recent papers [17, 15, 16], it is shown that many of the numerical methods for solving the Euler-Lagrange equations lead to *generalized solutions* of overdetermined differential-algebraic equations. Here, the index is reduced and all the constraints are included, but their discretizations lead to overdetermined nonlinear algebraic systems, which are inconsistent in general.

In the past, overdetermined differential-algebraic equations were solved in *a least square sense* [16], by *tangent space parameterization* [39], or using the *ssf-inverse method* [16]. Recently, Nashed and Chen [34] proposed a Newton-like method using outer inverses for solving singular operator equations.

In Chapter 2, outer and inner inverses of linear operators between Banach spaces are defined. A general method for obtaining outer and inner inverses for arbitrary rectangular matrices is proposed. The Newton-like method using outer inverses proposed by Nashed and Chen [34] is presented.

Chapter 3 presents some basic theoretical aspects related to differential-algebraic equations. Also an existence and uniqueness result for a class of delay differential-algebraic equations is given. The result was proven using an existence and uniqueness theorem proved by Potra and Rheinboldt [41], by means of differential-geometric techniques.

The fourth chapter of the thesis discusses the Euler-Lagrange equations and their numerical solution. The *ssf-inverse method* and the *tangent space parameterization* are presented. In this chapter, a class of Newton-like methods using outer inverses for solving the overdetermined nonlinear system arising in multi-body system simulation is described. It is then shown that the methods proposed by Potra [39] and by Führer and Leimkuhler [16] can be interpreted as particular procedures belonging to this class. New procedures for overdetermined differential-algebraic equations are proposed and the numerical results obtained using them on some examples are discussed.

In Chapter 5, a problem that came from determining the boundary surfaces of working domains of mechanisms is solved. This chapter presents a numerical algorithm that can be used to obtain all tangent vectors for a class of multiple bifurcation points that occur when using numerical continuation methods to map the boundary surface.

There are situations in which mechanical systems involving mechanisms that are controlled by a human operator, must be modeled at two different levels of abstraction; one suitable for real-time simulation and one with a higher level of fidelity for design purposes. Both models are described by differential-algebraic equations. It is necessary to find an input for the high-fidelity model that will yield a behaviour of its solution that is close to the one of the low-fidelity model. Chapter 6 contains the analytical formulation of the above problem, and the solution of the above optimization problem using sensitivity analysis. The numerical algorithm proposed is then tested on examples, and results are discussed.

CHAPTER 2
OUTER INVERSES

2.1 Outer and Inner Inverses

Let X and Y be Banach spaces and let A be a bounded linear operator on X into Y. A linear operator $B : Y \to X$ is said to be an *inner inverse* of A ([33]) if

$$ABA = A \tag{2.1}$$

A linear operator $B : Y \to X$ is an *outer inverse* of A if

$$BAB = B \tag{2.2}$$

Inner and outer inverses are sometimes called generalized inverses. Only nonzero outer inverses are considered (the zero operator is always an outer inverse).

Let A be an $m \times n$ matrix, $m > n$. Any outer inverse B of A will be an $n \times m$ matrix, and the following result allows one to obtain a large class of outer inverses for a given $m \times n$ matrix.

Lemma 2.1.1 *(a). If $rank(A) = n$, then A can be written as*

$$A = E \begin{bmatrix} I \\ 0 \end{bmatrix} \tag{2.3}$$

where I is the $n \times n$ identity matrix and E is an $m \times m$ invertible matrix. The $n \times m$ matrix

$$B = [I \quad P]\ E^{-1} \tag{2.4}$$

is an outer inverse of A for any $n \times (m-n)$ matrix P.

(b). If $rank(A) = r < n$, then A can be written as

$$A = E \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} F \tag{2.5}$$

where E is an $m \times m$ invertible matrix, I is the $r \times r$ identity matrix, and F is an $n \times n$ invertible matrix. If M is an outer (inner) inverse of the matrix $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$, then the $n \times m$ matrix

$$B = F^{-1}\ M\ E^{-1} \tag{2.6}$$

is an outer (inner) inverse of A.

Proof (a). For any $n \times (m-n)$ matrix P, we have

$$\begin{aligned} BAB &= [I \quad P]\ E^{-1} E \begin{bmatrix} I \\ 0 \end{bmatrix} [I \quad P]\ E^{-1} \\ &= [I \quad P] \begin{bmatrix} I \\ 0 \end{bmatrix} [I \quad P]\ E^{-1} \\ &= I\ [I \quad P]\ E^{-1} = B \end{aligned} \tag{2.7}$$

so that B is an outer inverse of A.

(b). Let M be an outer inverse of $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$, i.e.

$$M \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} M = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \tag{2.8}$$

Then

$$
\begin{aligned}
BAB &= F^{-1}\ M\ E^{-1}E \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} FF^{-1}\ M\ E^{-1} \\
&= F^{-1}\ M \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} M\ E^{-1} \\
&= F^{-1}\ M\ E^{-1} = B \qquad (2.9)
\end{aligned}
$$

so that B is an outer inverse of A for any outer inverse M of $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$. The proof for the inner inverse is similar. $\square$

Remarks. (i) It is immediate to show that a matrix M is both an inner and an outer inverse of $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ if and only if M can be written in the form:

$$
M = \begin{bmatrix} I & B \\ C & CB \end{bmatrix} = \begin{bmatrix} I \\ C \end{bmatrix} \begin{bmatrix} I & B \end{bmatrix}. \qquad (2.10)
$$

(ii) For any $(n-r)\times r$ matrix Q, the matrix $M = \begin{bmatrix} I & 0 \\ Q & 0 \end{bmatrix}$ is an outer inverse of $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$, since we have the identity

$$
\begin{bmatrix} I & 0 \\ Q & 0 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} I & 0 \\ Q & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ Q & 0 \end{bmatrix}.
$$

In practice, to obtain outer inverses of an $m \times n$ matrix A, with $m > n$, one can use the singular-value decomposition of the matrix A, to write it in the form needed in Lemma 2.1.1. Let $rank(A) = r \leq n$, and let E, S, and V be the matrices in the singular-value decomposition of the matrix A; i.e.,

$$
A = ESV^T \qquad (2.11)
$$

where E is an $m \times m$ unitary matrix, V is an $n \times n$ unitary matrix, and S is a $m \times n$ diagonal matrix of the same dimension as A, of the form $S = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}$ with

$$D = \begin{bmatrix} \alpha_1 & 0 & \dots & 0 \\ 0 & \alpha_2 & \dots & 0 \\ . & . & . & . \\ 0 & 0 & \dots & \alpha_r \end{bmatrix}$$

where $\alpha_1, \alpha_2, ..., \alpha_r$ are the singular values of A. Define a diagonal square matrix W of size n with the first r diagonal elements coinciding with α_i and the last $n-r$ equal to 1, if $r = rank(A) < n$. Clearly W is invertible. Let $F = WV^T$ and $S_1 = SW^{-1}$. Then, the matrix A can be written as

$$A = ES_1F$$

and S_1 has the form $\begin{bmatrix} I \\ 0 \end{bmatrix}$ if $rank(A) = n$, and $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ if $rank(A) = r < n$. Using Lemma 2.1.1, inner and outer inverses can now be easily constructed. Also, Lemma 2.1.1 can be modified to accommodate arbitrary nonsingular diagonal matrices, instead of I in the upper left corner. Therefore, forcing an identity matrix in that position is not actually necessary.

2.2 Newton-like Methods Using Outer Inverses

Nashed and Chen [34] present a class of Newton-like methods that use outer inverses to solve singular operator equations in Banach spaces. The main result of their paper is the Kantorovich-type theorem stated below. Let X and Y be Banach spaces. Denote by $S(x_0, r)$ the open ball with center x_0 and radius r, and by $\bar{S}$ the closure of S. Let $\mathcal{R}(A)$ denote the range of an operator A.

Theorem 2.2.1 *([34]) Let $F : D \subset X \to Y$ be Frechet differentiable and let $A(x)$ be a bounded linear approximation of $F'(x)$. Assume that there exist an open convex* *subset D_0 of D; $x_0 \in D_0$; a bounded outer inverse $A^\#$ of $A = A(x_0)$; positive constants η and K, and nonnegative constants $M, L, \mu,$and $l \geq 0$ such that for all x and $y \in D_0$ the following conditions hold:*

$$\|A^\# F(x_0)\| \leq \eta \tag{2.12}$$

$$\|A^\#(F'(x) - F'(y))\| \leq K\|x - y\| \tag{2.13}$$

$$\|A^\#(F'(x) - A(x))\| \leq M\|x - x_0\| + \mu \tag{2.14}$$

$$\|A^\#(A(x) - A)\| \leq L\|x - x_0\| + l \tag{2.15}$$

$$b := \mu + l < 1 \tag{2.16}$$

Also, assume that $h := \sigma\eta \leq \frac{1}{2}(1-b)^2$, where $\sigma := max(K, M + L)$ and $\bar{S} = \bar{S}(x_0, t^*) \subset D_0$, where $t^* = \left(1 - b - \sqrt{(1-b)^2 - 2h}\,\right)/\sigma$. Then,

(i) the sequence x_k defined by

$$x_{k+1} = x_k - A(x_k)^\# F(x_k) \tag{2.17}$$

with $A(x_k)^\# = (I + A^\#(A(x_k) - A))^{-1}A^\#$, lies in $S(x_0, t^*)$ and converges to a solution $x^* \in \bar{S}$ of $A^\# F(x) = 0$;

(ii) the equation $A^\# F(x) = 0$ has a unique solution in $\hat{S} \cap \{\mathcal{R}(A^\#) + x_0\}$, where

$$\hat{S} = \begin{cases} \bar{S}(x_0, t^*) \cap D_0, & if\ \ h = \frac{1}{2}(1-b)^2 \\ S(x_0, t^{**} \cap D_0, & if\ \ h < \frac{1}{2}(1-b)^2 \end{cases}$$

$$\mathcal{R}(A^\#) + x_0 := \{x + x_0 : x \in \mathcal{R}(A^\#)\}$$

and

$$t^{**} = \left(1 - b + \sqrt{(1-b)^2 - 2h}\,\right)/\sigma$$

The following consequence of this result can be used to solve, in a generalized sense, the overdetermined nonlinear equations that appear when looking for the numerical solutions of the Euler-Lagrange equations.

Corollary 2.2.1 *([34]) The Newton method using outer inverses*

$$x_{k+1} = x_k - F'(x_k)^{\#} F(x_k) \tag{2.18}$$

converges quadratically to a solution $x^* \in \hat{S} \cap \{\mathcal{R}(F'(x_0)^{\#}) + x_0\}$ *of* $F'(x_0)^{\#} F(x) = 0$ *under the conditions of Theorem 2.2.1, with* $A(x) = F'(x)$.

CHAPTER 3
DIFFERENTIAL-ALGEBRAIC EQUATIONS

3.1 Differential-Algebraic Equations

Consider the implicit differential equation

$$F(t, x, \dot{x}) = 0 \tag{3.1}$$

where $F : \Re \times \Re^{n_x} \times \Re^{n_x} \rightarrow \Re^{n_x}$, and the dot ($\dot{}$) represents differentiation with respect to t. A differential equation of the form (3.1) is called a *differential-algebraic equation* (DAE). In a differential-algebraic equation there are algebraic constraints on the variables. The constraints may appear explicitly as in equation (3.2b) below

$$F(t, x, y, \dot{x}) = 0 \tag{3.2a}$$

$$G(t, x, y) = 0 \tag{3.2b}$$

where the Jacobian matrix of F with respect to $\dot{x}$ is nonsingular, or they may arise because the Jacobian $F_{\dot{x}}$ in equation (3.1) is singular.

Definition 3.1.1 *([10]) The* differential index *of (3.1) is the minimum number of times that all or part of (3.1) must be differentiated with respect to t in order to determine $\dot{x}$ as a continuous function of x and t.*

In view of this definition, explicit ordinary differential equations of the form $y' = f(t, y)$, as well as implicit differential equations of the form (3.1) with nonsingular Jacobian matrix $F_{\dot{x}}$, have index zero. A semi-explicit differential-algebraic equation of the form

$$\dot{x} = f(t, x, y) \tag{3.3a}$$

$$0 = g(t, x, y) \tag{3.3b}$$

has index one if g_y is nonsingular, and higher than one otherwise. It turns out that the higher the index, the more difficult the numerical solution [10].

Definition 3.1.2 *([19]) The* perturbation index *of a DAE is the minimum integer k such that the following holds: For every solution x of (3.1) and all functions $\tilde{x}$ having a defect*

$$F(t, \tilde{x}, \dot{\tilde{x}}) = \delta(t)$$

there exists on $[0, T]$ an estimate

$$\|\tilde{x}(t) - x(t)\| \leq C \left(\|\tilde{x}(0) - x(0)\| + \max_{0 \leq \zeta \leq t} \|\delta(\zeta)\| + \ldots + \max_{0 \leq \zeta \leq t} \|\delta^{k-1}(\zeta)\| \right)$$

whenever the expression on the right side is sufficiently small, where C is a constant that depends only on F and the length of the time interval.

The perturbation index is a measure of the sensitivity of solutions with respect to perturbations in the equation. Therefore it plays an important role in studying the behaviour of the numerical solutions of DAE's.

3.2 Delay-Differential-Algebraic Equations: Existence and Uniqueness

Although there has been a great deal of work, recently, related to the study of differen-

tial-algebraic equations, there is very little literature concerning the study of delay-differential-algebraic equations. The delay-DAE's appear in a variety of scientific and engineering applications, such as the real time simulation of mechanical systems, optimal control, circuit analysis, chemical process simulation, etc. This section presents an existence and uniqueness result for a class of delay-DAE's.

3.2.1 A Class of Non-Autonomous DAE's

Consider the following first-order autonomous DAE

$$F(x) = b \tag{3.4a}$$

$$A(x)x' + B(x)z + G(x) = 0 \tag{3.4b}$$

where $F : E_x \subset \Re^n \to \Re^r$, $A : E_x \to L(\Re^n, \Re^s)$, $B : E_x \to L(\Re^n, \Re^s)$ and $G : E_x \to \Re^s$ are of class C^ρ, $\rho \geq 3$ on the open non-empty set $E_x \subset \Re^n$.

Assume that

$$r < n \leq r + s = n + m \tag{3.5}$$

and that for each $x \in E_x$ the matrix

$$\begin{bmatrix} DF(x) & 0 \\ A(x) & B(x) \end{bmatrix} \tag{3.6}$$

is non-singular.

By means of differential-geometric techniques, Potra and Rheinboldt proved [41] the following theorem.

Theorem 3.2.1 *If conditions (3.5) and (3.6) hold for (3.4), then for any given* $x_0 \in E_x$*, there exists a unique, maximally extended* C^ρ*-solution* $x : J \subset \Re \to E_x$ *and* $z : J \to E_x$ *of (3.4),* J *an interval containing 0, that satisfies the initial conditions*

$x(0) = x_0$, $x'(0) = \eta(x_0)$, *and* $z(0) = \zeta(x_0)$, *where*

$$\begin{bmatrix} \eta(x) \\ \zeta(x) \end{bmatrix} = \begin{bmatrix} DF(x) & 0 \\ A(x) & B(x) \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ -G(x) \end{bmatrix} \tag{3.7}$$

Now, this result is extended to the non-autonomous case. Consider the first-order non-autonomous DAE :

$$F(t,x) = b \tag{3.8a}$$

$$A(t,x)x' + B(t,x)z + G(t,x) = 0 \tag{3.8b}$$

where $F : [0,\infty) \times E_x \subset \Re^{n+1} \to \Re^r$, $A : [0,\infty) \times E_x \to L(\Re^n, \Re^s)$, $B : [0,\infty) \times E_x \to L(\Re^m, \Re^s)$, $G : [0,\infty) \times E_x \to \Re^s$ are of class $C^\rho, \rho \geq 3$, and E_x is a non-empty open set in $\Re^n$. Here, as usual, $L(\Re^n, \Re^m)$ is the space of all linear mappings with domain $\Re^n$ and range in $\Re^m$.

Assume that

$$r \leq n \leq r + s = n + m \tag{3.9}$$

and that, for each $x \in E_x$, $t \geq 0$, the matrix

$$\begin{bmatrix} D_xF(t,x) & 0 & D_tF(t,x) \\ A(t,x) & B(t,x) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.10}$$

is non-singular.

In the standard way, add to the system (3.8) the equation $t' = 1$, to transform it into an autonomous system, and let

$x_1 = (t,x) \in E_{x_1} = [0,\infty) \times E_x \subset \Re^{n+1}$

$x_1' = (t', x') \in E_{p_1} = \Re \times E_p \subset \Re^{n+1}$
$\mathcal{A} : E_{x_1} \to L(\Re^n, \Re^{s+1}), \mathcal{A}(x_1) = \begin{pmatrix} 1 & 0 \\ 0 & A(x_1) \end{pmatrix}$
$\mathcal{B} : E_{x_1} \to L(\Re^n, \Re^{s+1}), \mathcal{B}(x_1) = \begin{pmatrix} 0 \\ B(x_1) \end{pmatrix}$
$\mathcal{G} : E_{x_1} \to \Re^{s+1}, \mathcal{G}(x_1) = \begin{pmatrix} 1 \\ G(x_1) \end{pmatrix}$
With these notations, the system (3.8) becomes

$$F(x_1) = b \tag{3.11a}$$

$$\mathcal{A}(x_1)x_1' + \mathcal{B}(x_1)z + \mathcal{G}(x_1) = 0 \tag{3.11b}$$

which is of the form (3.4). Conditions (3.9) and (3.10) imply that

$$r < n + 1 \leq r + (s + 1) = (n + 1) + m \tag{3.12}$$

and that, for all $x_1 \in E_{x_1}$,

$$\begin{bmatrix} DF(x_1) & 0 \\ \mathcal{A}(x_1) & \mathcal{B}(x_1) \end{bmatrix} = \begin{bmatrix} D_tF(t,x) & D_xF(t,x) & 0 \\ 1 & 0 & 0 \\ 0 & A(t,x) & B(t,x) \end{bmatrix} \tag{3.13}$$

which is non-singular by (3.10).

Thus, Theorem 3.2.1 can be applied, and it follows that if (3.9) and (3.10) hold, then for any given $x_{1_0} \in E_{x_1}$, there exists a unique, maximally extended C^ρ-solution $x_1 : J \subset \Re \to E_{x_1}$, $z : J \to E_z \subset \Re^m$ of (3.11), that satisfies the initial conditions $x_1(0) = x_{1_0}$, $x_1'(0) = \eta_1(x_{1_0})$, and $z(0) = \zeta_1(x_{1_0})$, where η_1, and ζ_1 are given by an equation analogous to (3.7). Now, since x_1 is the unique solution of (3.11), and since its first component satisfies the differential equation $y' = 1$ with initial condition $y(0) = 0$, it is clear that x_1 is of the form $x_1(t) = (t, x(t))$.

Thus, we have proven the following

Theorem 3.2.2 *If conditions (3.9) and (3.10) hold for (3.8), then, for any given $x_0 \in E_x$, there exists a unique, maximally extended C^ρ-solution $x : J \subset \Re \to E_x$, $z : J \to E_z$ of (3.8), J an interval containing 0, that satisfies the initial conditions $x(0) = x_0$, $x'(0) = \eta(x_0)$, and $z(0) = \zeta(x_0)$, where η and ζ are given by an equation analogous to (3.7).*

This is the corresponding of Theorem 3.2.1 for the non-autonomous case.

3.2.2 A Class of Non-Autonomous First Order Delay-DAEs

Consider the following Initial Value Problem for a first order delay-DAE of retarded type:

$$F(t, x) = b \tag{3.14a}$$

$$A(t, x)x'(t) + B(t, x)x(t-1) + C(t, x)z(t) + D(t, x) = 0 \tag{3.14b}$$

with the initial condition

$$x(t) = f(t), \quad -1 \leq t \leq 0 \tag{3.15}$$

where $F : [0, \infty) \times E_x \subset \Re^{n+1} \to \Re^r, A : [0, \infty) \times E_x \to L(\Re^n, \Re^s)$, $B : [0, \infty) \times E_x \to L(\Re^n, \Re^s), C : [0, \infty) \times E_x \to L(\Re^m, \Re^s), D : [0, \infty) \times E_x \to \Re^s, f : [-1, 0] \to \Re$ are of class $C^\rho, \rho \geq 3$, E_x is an open non-empty set in $\Re^n$, and $z \in E_z \subset \Re^m$, E_z open. Without loss of generality, the delay is set to 1, but identical results can be obtained for any constant delay.

Assume that

$$r \leq n \leq r + s = n + m \tag{3.16}$$

and that for each $x \in E_x$ and $t \leq 0$, the matrix

$$\begin{bmatrix} D_xF(t,x) & 0 & D_tF(t,x) \\ A(t,x) & C(t,x) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.17}$$

is non-singular.

On the interval $[0,1]$, using the initial condition (3.15), equation (3.14b) becomes

$$A(t,x)x'(t) + B(t,x)f(t-1) + C(t,x)z(t) + D(t,x) = 0 \tag{3.18}$$

Let $G_0(t,x) = B(t,x)f(t-1) + D(t,x)$, for $0 \leq t \leq 1$, which is of class C^ρ on [0,1]. Then, on $[0,1]$, the system (3.14) becomes

$$F(t,x) = b \tag{3.19a}$$

$$A(t,x)x'(t) + C(t,x)z(t) + G_0(t,x) = 0 \tag{3.19b}$$

By Theorem 3.2.2, there exists a unique C^ρ-solution (x_0, z_0) of (3.19), on [0,1], that satisfies the initial conditions $x_0(0) = f(0)$, and $z_0(0) = \zeta(f(0))$, where x_0, and z_0 are defined on an interval containing 0, and ζ is given by an equation analogous to (3.7).

At the next step, consider the system (3.14) on the interval $[0,2]$. Let $f_1 : [-1,1] \to E_x$ be given by

$$f_1(t) = \begin{cases} f(t), & on \quad [-1,0] \\ x_0(t), & on \quad [0,1] \end{cases}$$

which is of class C^ρ, since x_0 is of class C^ρ on an interval containing 0. The equation (3.14b) becomes, on $[0,2]$,

$$A(t,x)x'(t) + B(t,x)f_1(t-1) + C(t,x)z(t) + D(t,x) = 0 \tag{3.20}$$

Let $G_1(t,x) = B(t,x)f_1(t-1) + D(t,x)$ for $0 \leq t \leq 2$, which is of class C^ρ on $[0,2]$. The system (3.14) becomes, on $[0,2]$,

$$F(t,x) = b \tag{3.21a}$$

$$A(t,x)x'(t) + C(t,x)z(t) + G_1(t,x) = 0 \tag{3.21b}$$

By Theorem 3.2.2, there exists a unique C^ρ solution (x_1, z_1) of (3.21) on $[0,2]$, that satisfies the initial conditions $x_1(0) = f(0)$, and $z_1(0) = \zeta(f(0))$.

Since this process can be continued indefinitely, the following is proved.

Theorem 3.2.3 *If conditions (3.16) and (3.17) hold for (3.14), then there exists a unique C^ρ-solution $x : J \to E_x, z : J \to E_z$ of (3.14) that satisfies the initial conditions $x(0) = f(0)$, and $z(0) = \zeta(f(0))$, where ζ is given by an equation analogous to (3.7).*

CHAPTER 4
NUMERICAL METHODS FOR CONSTRAINED MECHANICAL SYSTEMS

4.1 The Euler-Lagrange Equations

The equations of motion of constrained multi-body systems can be written, when holonomic constraint equations are considered, in the following general form

$$M(t,p)\ddot{p} + G(t,p)^T\lambda = f(t,p,\dot{p}) \tag{4.1a}$$

$$g(t,p) = 0 \tag{4.1b}$$

Here, the *position vector* $p \in \Re^{n_p}$ contains *generalized coordinates* chosen to describe the multi-body system. The number n_p of coordinates, as well as their physical interpretation, depends on the particular formulation. Correspondingly, $M(t,p)$ is a symmetric $n_p \times n_p$ matrix called the *generalized mass matrix*, $\lambda \in \Re^{n_\lambda}$ is the vector of *Lagrange multipliers* that define *constraint forces* in the system, $f(t,p,\dot{p})$ is the vector of *generalized external forces*, $g : \Re \times \Re^{n_p} \to \Re^{n_\lambda}$ is the mapping defining kinematic (position level) constraints, and $G(t,p) = D_p g(t,p)$ is the Jacobian matrix of $g(t,p)$ with respect to p.

By differentiating the position level constraints (4.1b) with respect to time, the velocity level constraints are obtained.

$$G(t,p)\dot{p} + D_t g(t,p) = 0 \tag{4.2}$$

One additonal differentiation gives the acceleration level constraints,

$$G(t,p)\ddot{p} + z(t,p,\dot{p}) = 0 \tag{4.3}$$

where

$$z(t,p,\dot{p}) = (D_pG(t,p)\dot{p})\dot{p} + 2(D_tG(t,p))\dot{p} + D_{tt}g(t,p)$$

With some formulations, the matrix $M(t,p)$ is symmetric and positive definite, while with others $M(t,p)$ may be only symmetric and positive semidefinite. If $G(t,p)$ has full row rank; i.e.,

$$\text{rank } G(t,p) = n_\lambda \tag{4.4}$$

and that

$$a^T M(t,p)a > 0, \quad \forall a \in \ker G(t,p) \tag{4.5}$$

then the Euler-Lagrange equations have a unique maximally extended solution for any consistent initial values $p(0)$, and $\dot{p}(0)$ [23]. Clearly, any such solution will also satisfy the constraints (4.2) and (4.3). By the assumptions (4.4) and (4.5), the matrix

$$\begin{bmatrix} M(t,p) & G(t,p)^T \\ G(t,p) & 0 \end{bmatrix} \tag{4.6}$$

is nonsingular, so the linear system in $\ddot{p}$ and λ formed by (4.1a) and (4.3) has a unique solution

$$\ddot{p} = \eta(t,p,\dot{p}) \tag{4.7}$$

$$\lambda = \zeta(t,p,\dot{p}) \tag{4.8}$$

Now, for any initial conditions $p_0, \dot{p}_0 \in \Re^{n_p}$, the ordinary differential equation

(4.7) has a unique solution that satisfies $p(0) = p_0$ and $\dot{p}(0) = \dot{p}_0$. If the initial values are consistent; i.e., if the constraints (4.1b) and (4.2) are satisfied for $t = 0, p = p_0$, and $\dot{p} = \dot{p}_0$, then this unique solution, together with λ determined by (4.8), represents a solution of the Euler-Lagrange equations (4.1). Thus, in this case, theoretically, the problem of solving the Euler-Lagrange equations (4.1) is reduced to the problem of solving an ordinary differential equation (ODE) with consistent initial values [38]. However, the numerical solution obtained by applying standard ODE solvers no longer satisfies the kinematic constraints (4.1b), which is not admissible in most practical situations.

A different approach is to generate, by differentiating the constraints, an overdetermined lower-index differential-algebraic equation, and view the solution of the Euler-Lagrange equations as a *generalized solution* of this overdetermined differential equation. For this purpose, it is convenient to rewrite the Euler-Lagrange equations (4.1) as a first-order system, by introducing the variable $v = \dot{p}$. Equations (4.1), (4.2), and (4.3) become

$$\dot{p} - v = 0 \tag{4.9a}$$

$$M(t,p)\dot{v} + G(t,p)^T\lambda - f(t,p,v) = 0 \tag{4.9b}$$

$$G(t,p)\dot{v} + z(t,p,v) = 0 \tag{4.9c}$$

$$G(t,p)v + D_t g(t,p) = 0 \tag{4.9d}$$

$$g(t,p) = 0 \tag{4.9e}$$

Consider the vector $x = (p, v, \lambda)$ of dimension $n_x = 2n_p + n_\lambda$, and denote by $F(t, x, \dot{x})$ the vector defined by the left sides of equations (4.9a), (4.9b), and (4.9e). Then the

Euler-Lagrange equations can be written as

$$F(t, x, \dot{x}) = 0 \tag{4.10}$$

where F is a given mapping of $\Re \times \Re^{n_x} \times \Re^{n_x}$ into $\Re^{n_x}$. If the conditions (4.4) and (4.5) are satisfied, the index of the differential-algebraic equation (4.10) is 3, since by differentiating (4.1) twice, we have obtained equations (4.7) and (4.8), and one more differentiation with respect to t, combined with (4.7), will give an explicit expression of $\dot{x}$ as a function of x and t.

On the other hand, the system (4.9) can be written in the form

$$\Psi(t, x, \dot{x}) = 0 \tag{4.11}$$

where Ψ is a mapping of $\Re \times \Re^{n_x} \times \Re^{n_x}$ into $\Re^{\gamma}$, with $\gamma = n_x + 2n_\lambda$. Any equation of the form (4.11), with $\Psi : \Re^{2n_x+1} \to \Re^{\gamma}$, is called an overdetermined differential-algebraic equation (ODAE) [15, 16, 17]. The index of a general ODAE of the form (4.11) can be defined in the same way the index was defined for a DAE, so in this case, by the argument above, the ODAE (4.11), and hence the system (4.9), has index 1, since one differentiation will determine $\dot{x}$ as a continuous function of x and t.

There are many ways in which a lower index ODAE can be generated by a DAE. For example the following ODAE generated by the Euler-Lagrange equations (4.9a,4.9b,4.9e) is considered in [16], under the supplementary assumption that the matrix $M(t, p)$ is symmetric and positive definite

$$\dot{p} - v = 0 \tag{4.12a}$$

$$M(t, p)\dot{v} + G(t, p)^T \lambda - f(t, p, v) = 0 \tag{4.12b}$$

$$G(t, p)M(t, p)^{-1} f(t, p, v) - H(t, p)\lambda + z(t, p, v) = 0 \tag{4.12c}$$

$$G(t,p)v + D_t g(t,p) = 0 \tag{4.12d}$$

$$g(t,p) = 0 \tag{4.12e}$$

where

$$z(t,p,v) = (D_p G(t,p)v)v + 2(D_t G(t,p))v + D_{tt} g(t,p) \tag{4.13a}$$

$$H(t,p) = G(t,p)M(t,p)^{-1}G(t,p)^T \tag{4.13b}$$

Equations (4.9) and (4.12) are identical, except equations (c) of both formulations. Equation (4.9c) is a differential equation, while (4.12c) is an algebraic equation. Equation (4.12c) is obtained by solving (4.9b) for $\dot{v}$ and then substituting into (4.9c). Note that if $M(t,p)$ is symmetric positive definite, then so is $H(t,p)$. It is convenient to write (4.12) in the form

$$\hat{\Psi}(t,x,\dot{x}) = 0 \tag{4.14}$$

where the mapping $\hat{\Psi} : \Re \times \Re^{n_x} \times \Re^{n_x} \to \Re^{\gamma}$, $\gamma = n_x + 2n_\lambda$, is defined by the left side of (4.12).

The equations (4.11) and (4.14) are overdetermined, but consistent, because they are satisfied by any solution of (4.1). This is no longer the case once they are discretized using an integration formula. The system obtained is an overdetermined nonlinear algebraic system, and is inconsistent in general, so it has no classical solution. By requiring that the solution satisfies the overdetermined system in a weaker sense, several ways of solving the problem have been proposed, with good numerical results.

Applying a backward differentiation formula (BDF) to (4.14) replaces the differentiation operator $\dot{x}(t_n) = D_t x(t_n)$ with its corresponding difference operator

$\dot{x}_n = \rho x_n$, defined by

$$\dot{x}_n := \rho x_n := \frac{\alpha_0}{h} x_n + \frac{1}{h} \sum_{i=1}^{k} \alpha_i x_{n-i} \tag{4.15}$$

where α_i are the BDF-coefficients and h is the time step used in integration. Knowing the past values x_{n-i}, for $i = 1, 2, ..., k$, the current value x_n of the numerical solution of (4.14) is obtained as the "solution" of the nonlinear system

$$\hat{\psi}(x_n) := \hat{\Psi}(t_n, x_n, \rho x_n) = 0 \tag{4.16}$$

Solving this overdetermined system requires first the definition of an appropriate generalized solution. If Newton's method is formally applied to (4.16), a solution is defined by the limit of a sequence of solutions of the linear problems in Δx

$$D_x \hat{\psi}(x_n^{(m)}) \Delta x + \hat{\psi}(x_n^{(m)}) = 0 \tag{4.17}$$

with

$$x_n^{(m+1)} = x_n^{(m)} + \Delta x \tag{4.18}$$

Again, the linear system (4.17) is an overdetermined inconsistent linear system, and it cannot be expected to have a classical solution.

4.2 The SSF-Solution

In [16] and [17] the authors propose solving the overdetermined linear system (4.17) by introducing the notion of a *state-space-form solution*, or *ssf-solution* as follows. If A is an $m \times n$ matrix with $m > n$, and if A is partitioned into a square $n \times n$ matrix A_1, and an $(m-n) \times n$ matrix A_2, then x^* is the ssf-solution of the overdetermined linear system

$$\begin{bmatrix} A_1 \\ A_2 \end{bmatrix} x = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}, \tag{4.19}$$

if the following conditions are satisfied

$$\begin{aligned} A_1x^* - a_1 &\in \text{Range}A_2^T && (4.20a)\\ A_2x^* &= a_2 && (4.20b) \end{aligned}$$

It is then proven that the above conditions determine x^* uniquely, provided the matrix V^TA_1V is nonsingular, where V is a matrix whose columns form a basis for the null-space of A_2.

The system (4.17) is of the form (4.19), with A_1 being the sub-Jacobian corresponding to the first three equations in (4.12). If Δx is chosen to be the ssf-solution of the linear system (4.17), and if the sequence $\{x_n^{(m)}\}_{m=0}^{\infty}$ defined by (4.17)-(4.18) converges to x_n^*, then x_n^* is called the *ssf-solution of the nonlinear system* (4.14).

If A_2 has full-row rank, which is the case when solving the Euler-Lagrange equations of motion, let $A_2^\dagger$ be the Moore-Penrose inverse of A_2, defined by $A_2^\dagger = A_2^T(A_2A_2^T)^{-1}$, which is a right inverse of A_2, i.e. $A_2A_2^\dagger = I$.

Lemma 4.2.1 *([16]) If V is a full column rank matrix spanning the null-space of A_2, and if V^TA_1V is nonsingular, then the ssf-solution of (4.20) is given by*

$$x^* = A_2^\dagger a_2 + V(V^TA_1V)^{-1}V^T(a_1 - A_1A_2^\dagger a_2) \tag{4.21}$$

or, equivalently, by

$$x^* = A_1^{-1}(I - A_2^TQ^{-1}A_2A_1^{-1})a_1 + A_1^{-1}A_2^TQ^{-1}a_2 \tag{4.22}$$

if A_1 is nonsingular, where $Q = A_2 A_1^{-1} A_2^T$.

Let V be a full column rank matrix spanning the null-space of A_2, $ker(A_2)$, such that $V^T A_1 V$ is nonsingular. Let $B : \Re^m \rightarrow \Re^n$ be the operator defined by

$$Ba = x^* \tag{4.23}$$

for any $a \in \Re^m$, where x^* is the ssf-solution of (4.20), given by equation (4.21). If a is partitioned as $a = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$, with $a_1 \in \Re^n$, and $a_2 \in \Re^{m-n}$, then B can be written as

$$B \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = A_2^\dagger a_2 + V(V^T A_1 V)^{-1} V^T (a_1 - A_1 A_2^\dagger a_2) \tag{4.24}$$

or, in matrix form

$$B = \begin{bmatrix} V(V^T A_1 V)^{-1} V^T & A_2^\dagger - V(V^T A_1 V)^{-1} V^T A_1 A_2^\dagger \end{bmatrix} \tag{4.25}$$

and B is an $n \times m$ matrix. We next establish the following result.

Proposition 4.2.1 *The matrix B is an inner and an outer inverse for the matrix A; i.e., the ssf-solution of the overdetermined system (4.20) can be regarded as an inner and an outer inverse solution.*

Proof. It is useful to denote $U = V(V^T A_1 V)^{-1} V^T$. It is immediate that $U A_1 U = U$. The following holds

$$\begin{aligned} ABA &= \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \begin{bmatrix} U & A_2^\dagger - U A_1 A_2^\dagger \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \\ &= \begin{bmatrix} A_1 U & A_1 A_2^\dagger - A_1 U A_1 A_2^\dagger \\ A_2 U & A_2 A_2^\dagger - A_2 U A_1 A_2^\dagger \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \end{aligned} \tag{4.26}$$

Since V spans $ker(A_2)$, $A_2 V = 0$, and $A_2 U = 0$. Therefore

$$ABA = \begin{bmatrix} A_1U & A_1A_2^\dagger - A_1UA_1A_2^\dagger \\ 0 & I \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$$

$$= \begin{bmatrix} A_1UA_1 + A_1A_2^\dagger A_2 - A_1UA_1A_2^\dagger A_2 \\ A_2 \end{bmatrix} \tag{4.27}$$

To show that $ABA = A$, it suffices to prove the identity

$$A_1UA_1 + A_1A_2^\dagger A_2 - A_1UA_1A_2^\dagger A_2 = A_1 \tag{4.28}$$

which is equivalent to

$$A_1UA_1(I - A_2^\dagger A_2)x = A_1(I - A_2^\dagger A_2)x \tag{4.29}$$

for any $x \in \Re^n$. Let $y := (I - A_2^\dagger A_2)x$. Then

$$A_2y = A_2x - A_2A_2^\dagger A_2x = A_2x - A_2x = 0 \tag{4.30}$$

so $y \in ker(A_2)$, therefore $y = Vz$, for some z of an appropriate dimension, or $(I - A_2^\dagger A_2)x = Vz$. Equation (4.29) becomes

$$A_1UA_1Vz = A_1Vz \tag{4.31}$$

which is true, since $UA_1V = V$. Thus $ABA = A$; i.e., B is an inner inverse of A. For proving the outer inverse property, the following holds

$$BAB = \begin{bmatrix} U & A_2^\dagger - UA_1A_2^\dagger \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \begin{bmatrix} U & A_2^\dagger - UA_1A_2^\dagger \end{bmatrix}$$

$$= \begin{bmatrix} U & A_2^\dagger - UA_1A_2^\dagger \end{bmatrix} \begin{bmatrix} A_1U & A_1A_2^\dagger - A_1UA_1A_2^\dagger \\ 0 & I \end{bmatrix} \tag{4.32}$$

since $A_2U = A_2V(V^TA_1V)^{-1}V^T = 0$. It follows that

$$BAB = \begin{bmatrix} UA_1U & UA_1A_2^\dagger - UA_1UA_1A_2^\dagger + A_2^\dagger - UA_1A_2^\dagger \end{bmatrix} \tag{4.33}$$

Now,

$$UA_1U = V(V^TA_1V)^{-1}V^TA_1V(V^TA_1V)^{-1}V^T = V(V^TA_1V)^{-1}V^T = U \tag{4.34}$$

and

$$\begin{aligned}
&UA_1A_2^\dagger - UA_1UA_1A_2^\dagger + A_2^\dagger - UA_1A_2^\dagger \\
&= V(V^TA_1V)^{-1}V^TA_1A_2^\dagger - V(V^TA_1V)^{-1}V^TA_1V(V^TA_1V)^{-1}V^TA_1A_2^\dagger \\
&\quad + A_2^\dagger - V(V^TA_1V)^{-1}V^TA_1A_2^\dagger = A_2^\dagger - V(V^TA_1V)^{-1}V^TA_1A_2^\dagger
\end{aligned} \tag{4.35}$$

so that

$$BAB = \begin{bmatrix} V(V^TA_1V)^{-1}V^T & A_2^\dagger - V(V^TA_1V)^{-1}V^TA_1A_2^\dagger \end{bmatrix} = B \tag{4.36}$$

which shows that B is an outer inverse of A. □

4.3 Tangent Space Parametrization

Another way of solving the ODAE generated by the Euler-Lagrange equations of motion is proposed in [39], and the idea is to "square" the overdetermined system by projecting its differential part and obtaining in this fashion an index one DAE.

First, by letting $\dot{v} = w$, $x = (p, v, w, \lambda) \in \Re^{3n_p + n_\lambda}$, and defining

$$\Gamma(t,x) = \begin{bmatrix} M(t,p)w + G(t,p)^T\lambda - f(t,p,v) \\ g(t,p) \\ G(t,p)v + D_t g(t,p) \\ G(t,p)w + z(t,p,v) \end{bmatrix} \tag{4.37}$$

the Euler-Lagrange equations (4.9) can be written as

$$\dot{p} - v = 0 \tag{4.38a}$$

$$\dot{v} - w = 0 \tag{4.38b}$$

$$\Gamma(t,p,v,w,\lambda) = 0 \tag{4.38c}$$

The system (4.38) is overdetermined, but consistent, because it is satisfied by any solution of (4.9). Applying a linear multi-step discretization scheme of the form

$$y_n = \sum_{i=1}^{r} \alpha_i y_{n-i} + h \sum_{i=0}^{r} \beta_i \dot{y}_{n-i} \tag{4.39}$$

to (4.38), the following nonlinear system has to be solved at each iteration step:

$$p - h\beta_0 v - \hat{p}_n = 0 \tag{4.40a}$$

$$v - h\beta_0 w - \hat{v}_n = 0 \tag{4.40b}$$

$$\Gamma(t_n,p,v,w,\lambda) = 0 \tag{4.40c}$$

where $\hat{p}_n$, and $\hat{v}_n$ are explicitly defined values given by the multi-step scheme. This is an overdetermined nonlinear system and it is inconsistent in general, so it has no classical solution. Equation (4.40) can be written in the form

$$\Phi(t_n, x) := \begin{bmatrix} F(x) \\ \Gamma(t_n, x) \end{bmatrix} = 0 \tag{4.41}$$

Let $H(t_n, x)$ be the Jacobian matrix of Φ with respect to x. Then,

$$H(t_n, x) := \begin{bmatrix} H_1(x) \\ D_x\Gamma(t_n, x) \end{bmatrix} \tag{4.42}$$

where

$$H_1(x) = \begin{bmatrix} I & -h\beta_0 I & 0 & 0 \\ 0 & I & -h\beta_0 I & 0 \end{bmatrix} \tag{4.43}$$

Formally applying Newton's method to (4.41) leads to solving an overdetermined linear system of the form

$$D_x\Phi(t_n, x_n^{(m)})\Delta x = -\Phi(t_n, x_n^{(m)}) \tag{4.44}$$

which can be written as

$$\begin{bmatrix} H_1 \\ H_2 \end{bmatrix} y = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \tag{4.45}$$

with $H_1 = H_1(x_n^{(m)}), H_2 = D_x\Gamma(t_n, x_n^{(m)}), y = \Delta x$, and $\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = -\Phi(t_n, x_n^{(m)})$.

The method proposed in [39] is to project first the differential part of (4.38) on an appropriate subspace, to obtain a DAE of the form

$$A_1^T(\dot{p} - v) = 0 \tag{4.46a}$$

$$A_2^T(\dot{v} - w) = 0 \tag{4.46b}$$

$$\Gamma(t, p, v, w, \lambda) = 0 \tag{4.46c}$$

where A_1, and A_2 are $n_p \times (n_p - n_\lambda)$ matrices, chosen such that

$$\begin{bmatrix} A_i^T \\ G(t_c, x_c) \end{bmatrix}, \quad i = 1, 2 \tag{4.47}$$

are nonsingular, where (t_c, x_c) are some consistent values; i.e., $\Gamma(t_c, x_c) = 0$. Discretizing (4.46) in the same way as above, the nonlinear system to be solved at each iteration becomes

$$A_1^T(p - h\beta_0 v - \hat{p}_n) = 0 \tag{4.48a}$$

$$A_2^T(v - h\beta_0 w - \hat{v}_n) = 0 \tag{4.48b}$$

$$\Gamma(t_n, p, v, w, \lambda) = 0 \tag{4.48c}$$

which is now a square system that can be solved in the usual way, by applying Newton's method. This leads to solving a square linear system of the form

$$\begin{bmatrix} A^T H_1 \\ H_2 \end{bmatrix} y = \begin{bmatrix} A^T a_1 \\ a_2 \end{bmatrix} \tag{4.49}$$

where the same notations as above were used, and $A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$.

Theorem 4.3.1 *If the matrices A_1, and A_2 are chosen such that the matrix*

$$\begin{bmatrix} A^T H_1 \\ H_2 \end{bmatrix} \tag{4.50}$$

is nonsingular, where $A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$, $H_1 = H_1(x_n^{(m)})$, and $H_2 = D_x\Gamma(t_n, x_n^{(m)})$, then the *projection method for solving the linear system*

$$\begin{bmatrix} H_1 \\ H_2 \end{bmatrix} y = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \tag{4.51}$$

is both an outer inverse and an inner inverse method.

Proof. Since the matrix $\begin{bmatrix} A^T H_1 \\ H_2 \end{bmatrix}$ is invertible, the linear system (4.49) can be solved to get

$$y = \begin{bmatrix} A^T H_1 \\ H_2 \end{bmatrix}^{-1} \begin{bmatrix} A^T a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} A^T H_1 \\ H_2 \end{bmatrix}^{-1} \begin{bmatrix} A^T & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \tag{4.52}$$

Let $H^{\#} = \begin{bmatrix} A^T H_1 \\ H_2 \end{bmatrix}^{-1} \begin{bmatrix} A^T & 0 \\ 0 & I \end{bmatrix}$. We have

$$\begin{aligned} H^{\#} H &= \begin{bmatrix} A^T H_1 \\ H_2 \end{bmatrix}^{-1} \begin{bmatrix} A^T & 0 \\ 0 & I \end{bmatrix} \\ &= \begin{bmatrix} A^T H_1 \\ H_2 \end{bmatrix}^{-1} \begin{bmatrix} A^T H_1 \\ H_2 \end{bmatrix} = I \end{aligned} \tag{4.53}$$

Therefore, $H^{\#} H H^{\#} = H^{\#}$ and $H H^{\#} H = H$. This proves that $H^{\#}$ is an inner and outer inverse of the matrix H, which is the Jacobian matrix of Φ with respect to x. □

Remarks. (i) If H_1 is a square matrix, by choosing $A = V$, with V generating the null-space of H_2, the *ssf-method* is obtained. If strictly considering procedures of building outer inverses, the method from the last theorem can be regarded as more general than the *ssf-method*. On the other hand, in a multi-body dynamics context, the two methods try to preserve different manifolds, and therefore they are qualitatively different (H_1 cannot be square in the framework of [39]).

(*ii*) The above result says that the method proposed in [39] is equivalent to a Newton-like method using outer inverses, as described in Corollary 2.2.1.

4.4 Numerical Methods using Outer Inverses

Consider the ODAE generated by the Euler-Lagrange equations

$$\dot{p} - v = 0 \tag{4.54a}$$

$$M(t,p)\dot{v} + G(t,p)^T\lambda - f(t,p,v) = 0 \tag{4.54b}$$

$$G(t,p)\dot{v} + z(t,p,v) = 0 \tag{4.54c}$$

$$G(t,p)v + D_t g(t,p) = 0 \tag{4.54d}$$

$$g(t,p) = 0 \tag{4.54e}$$

which can be written in the form

$$\Psi(t,x,\dot{x}) = 0 \tag{4.55}$$

Applying a BDF-scheme $\dot{x}_n = \rho x_n$ to (4.55), the current value of the numerical solution is obtained as the "solution" of an overdetermined nonlinear algebraic system of the form

$$\psi(x_n) := \Psi(t_n, x_n, \rho x_n) = 0 \tag{4.56}$$

which is inconsistent in general, so it has no classical solution.

We will solve this system in the *outer inverse sense*, by applying the Newton-like method using outer inverses of Theorem 2.2.1 and Corollary 2.2.1. The solution of the overdetermined system (4.56) is defined as the limit of the sequence

$$x_n^{(k+1)} = x_n^{(k)} - \psi'(x_n^{(k)})^{\#}\psi(x_n^{(k)}) \tag{4.57}$$

where $\psi'(x_n^{(k)})^{\#}$ is an outer inverse of $\psi'(x_n^{(k)})$. Under the conditions of Theorem 2.2.1, this sequence converges quadratically to a solution x_n of the equation $\psi'(x_n^{(0)})^{\#}\psi(x) = 0$, where $x_n^{(0)}$ is the starting point for the iterative process.

To compute the outer inverse $\psi'(x_n^{(k)})^{\#}$ at each k, we can use the updating formula given by Theorem 2.2.1. If we denote $A := \psi'(x_n^{(0)})$, then an outer inverse of $\psi'(x_n^{(k)})$ can be found by

$$\psi'(x_n^{(k)})^{\#} = (I + A^{\#}(A(x_n^{(k)}) - A))^{-1}A^{\#} \tag{4.58}$$

A suitable outer inverse of A, $A^{\#} = \psi'(x_n^{(0)})^{\#}$, can be found by the method described in Chapter 2.

4.5 Numerical Results

This section presents some numerical results obtained by applying the method described in the previous section to determine the motion of two classical examples of constrained mechanical systems: the pendulum and the slider-crank mechanism.

The purpose of the numerical experiments presented in this section is to illustrate the approach for solving DAE of multi-body dynamics based on outer inverses. Therefore, the integration was performed using Euler's method with constant step size. The outer inverse was determined using the method described in the remarks following Lemma 2.1.1. For an $m \times n$ matrix A, with $m > n$ and $rank(A) = r < n$, written as

$$A = E \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} F \tag{4.59}$$

the outer inverse used is

$$A^{\#} = F^{-1} \begin{bmatrix} I & 0 \\ Q & 0 \end{bmatrix} E^{-1}, \tag{4.60}$$

where Q is any $(n-r) \times r$ matrix. Experimented were performed with matrices Q of the form

$$Q = a \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ . & . & \dots & . \\ 1 & 1 & \dots & 1 \end{bmatrix} \tag{4.61}$$

for different choices of the real parameter a.

We compared our results with a reference solution which was obtained using the code ODASSL with a very small time step [17]. The error listed in Tables 1 and 2 is defined as the square root of the sum of the square of the errors at each integration step. "Tolerance" is a parameter that defines the stopping criteria for the Newton step.

The Simple Pendulum. The Euler-Lagrange equations that describe the motion of the mathematical pendulum of mass 1 and radius 1 are shown in Figure 1

$$\dot{x} = v \tag{4.62a}$$

$$\dot{y} = w \tag{4.62b}$$

$$\dot{v} + \lambda x = 0 \tag{4.62c}$$

$$\dot{w} + \lambda y + g = 0 \tag{4.62d}$$

$$x^2 + y^2 = 1 \tag{4.62e}$$

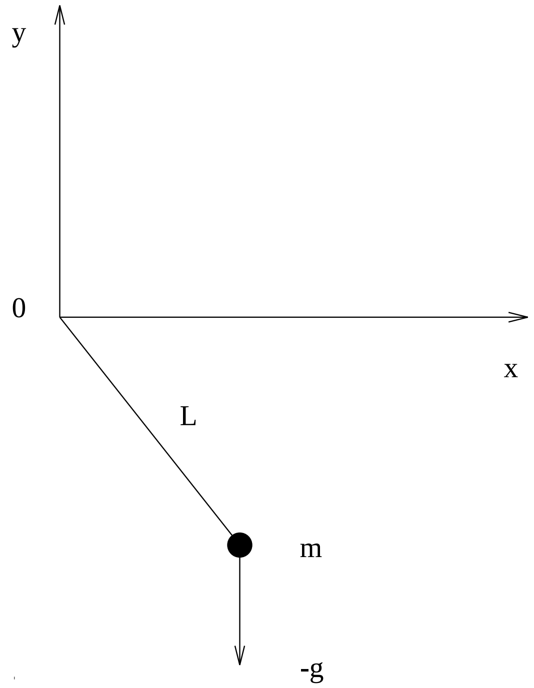


Figure 1: Simple pendulum

$$xv + yw = 0 \tag{4.62f}$$

$$v^2 + w^2 + x\dot{v} + y\dot{w} = 0 \tag{4.62g}$$

The results for this simulation are gathered in Table 1. They show that the method proposed is not sensitive relative to the choice of the outer inverse used, as reflected by changes in the parameter a. This opens the possiblity of finding an outer inverse with convenient properties (like sparsity), that can reduce computing time. Smaller stesizes lead to better accuracy in the solution, for different choices of the outer inverse.

The Slider-Crank Mechanism. Consider the motion of the Slider-Crank mechanism shown in Figure 2. The corresponding equations of motion are

a	h	Error	flops	tolerance
0.0001	0.001	2.0481e-04	6,142,082	0.0004
	0.01	0.0056	660,616	0.005
	0.1	0.0507	69,330	0.27
0.001	0.001	2.0481e-04	6,142,132	0.0004
	0.01	0.0056	660,592	0.005
	0.1	0.0507	69,330	0.27
0.01	0.001	2.0481e-04	6,142,132	0.0004
	0.01	0.0056	660,592	0.005
	0.1	0.0507	69,330	0.27
0.1	0.001	2.1635e-04	6,142,104	0.0004
	0.01	0.0059	664,624	0.005
	0.1	0.0507	69,330	0.27
1	0.001	9.9956e-04	6,223,027	0.0004
	0.01	0.0108	680,671	0.009
	0.1	0.0488	69,372	0.4

Table 1: Numerical results for the pendulum

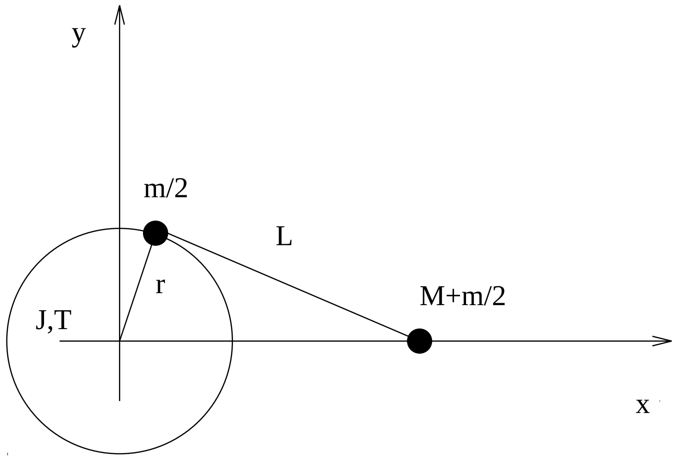


Figure 2: Slider-Crank mechanism

$$\dot{\theta} = v \tag{4.63a}$$

$$\dot{x} = w \tag{4.63b}$$

$$(J + \frac{m}{2})\dot{v} + 2rx\sin\theta\ \lambda = T - \frac{m}{2}gr\cos\theta \tag{4.63c}$$

$$(M + \frac{m}{2})\dot{w} + 2(x - r\cos\theta)\lambda = 0 \tag{4.63d}$$

$$x^2 - 2rx\cos\theta - r^2 - L^2 = 0 \tag{4.63e}$$

$$xw - rw\cos\theta + rxv\sin\theta = 0 \tag{4.63f}$$

$$rx\dot{v}\sin\theta + (x - r\cos\theta)\dot{w} + w^2 + rxv^2\cos\theta + 2rvw\sin\theta = 0 \tag{4.63g}$$

where θ is the crank angle, x is position of the slider on the x-axis, J is the inertia of the disc, and T is a constant torque applied on the crank. For numerical experiments, the following data have been used: $r = 1, m = 10, M = 10, T = 8, L = 2$, and $J = 5$. The equations of motion become

$$\dot{\theta} = v \tag{4.64a}$$

$$\dot{x} = w \tag{4.64b}$$

$$10\dot{v} + 2x \sin\theta\ \lambda = 8 - \frac{5g}{2}\cos\theta \tag{4.64c}$$

$$15\dot{w} + 2(x - \cos\theta)\lambda = 0 \tag{4.64d}$$

$$x^2 - 2x\cos\theta - 3 = 0 \tag{4.64e}$$

$$xw - w\cos\theta + xv\sin\theta = 0 \tag{4.64f}$$

$$x\dot{v}\sin\theta + (x - \cos\theta)\dot{w} + w^2 + xv^2\cos\theta + 2vw\sin\theta = 0 \tag{4.64g}$$

The numerical results obtained are summarized in Table 2. They agree with the results obtained for the simulation of the pendulum, in the sense that they show that the method is relatively independent to the choice of the outer inverse used.

a	h	Error	flops	tolerance
0.0001	0.001	1.1479e-4	6,241,631	0.0004
	0.01	0.0023	636,314	0.005
	0.1	0.0291	72,801	0.13
0.001	0.001	1.1478e-4	6,241,710	0.0004
	0.01	0.0023	636,263	0.005
	0.1	0.0291	72,801	0.13
0.01	0.001	1.1487e-4	6,245,574	0.0004
	0.01	0.0023	636,353	0.005
	0.1	0.0291	72,801	0.13
0.1	0.001	1.4686e-04	6,245,574	0.0004
	0.01	0.0024	641,198	0.005
	0.1	0.0291	72,801	0.13
1	0.001	9.4202e-05	6,240,283	0.002
	0.01	0.0044	646,817	0.02
	0.1	0.1764	100,165	0.4

Table 2: Numerical results for the slider-crank

CHAPTER 5
WORKSPACE ANALYSIS AND CONTINUATION METHODS

A very important aspect related to the simulation and optimization of mechanical systems is the determination of the boundary surfaces of their working domains, which are implicitly defined by a set of non-linear algebraic constraints. Workspace analysis for mechanisms and manipulators has conventionally focused on determining the set of all points in space that can be reached by a specific point on a working body of the mechanism or manipulator. Development of numerical methods for workspace analysis has been fuelled by advances in the numerical analysis of systems of nonlinear equations and related bifurcation issues. Numerical continuation is the basis for many of the solution methods. The solution sets to systems of nonlinear equations often have bifurcation points, where the solution splits into two or more components. At these points, tangent vectors are needed to implement the continuation method. These aspects have been solved for simple bifurcation points by Haug, Wang, and Wu [28], and Qiu, Luh, and Haug [44]. This chapter presents a numerical method to obtain all tangent vectors for a class of multiple bifurcation points of continuation curves that define the boundary of the $\mathbf{u}$-space solution set of $\mathbf{\Phi}(\mathbf{u}, \mathbf{w}, \mathbf{v}) = \mathbf{0}$ where, for $\mathbf{q} = \left[\mathbf{u}^T, \mathbf{w}^T, \mathbf{v}^T\right]^T$, $\mathbf{\Phi}(\mathbf{q})$ is of class C^2, $\mathbf{\Phi} : \Re^{n_q} \mapsto \Re^m$ and $\mathbf{q} \in \Re^{n_q}$. The class of multiple bifurcation points covered here occurs when the Jacobian matrix $\mathbf{\Phi}_{\mathbf{w},\mathbf{v}}$ is row rank deficient three. This case is of particular interest for the application of mapping the boundary of manipulator workspaces.

5.1 Introduction

Let $\mathbf{\Phi} : \Re^{n_q} \mapsto \Re^m$ denote the kinematic constraint vector function for a mechanism whose constraint equations are

$$\mathbf{\Phi}(\mathbf{q}) = \mathbf{0} \tag{5.1}$$

where $\mathbf{q} = \left[q_1, q_2, ..., q_{n_q}\right]^T$ is the vector of generalized coordinates that characterize the position and orientation of each body in the mechanism. The generalized coordinates are divided into three categories, associated with the use of the mechanism as a manipulator, as follows: $\mathbf{q} = \left[\mathbf{u}^T, \mathbf{w}^T, \mathbf{v}^T\right]^T$, where $\mathbf{v}$ is the vector of the input coordinates, $\mathbf{u}$ represents the output coordinates, and $\mathbf{w}$ is the collection of intermediate coordinates that are introduced for modeling convenience and are neither input nor output coordinates.

The *accessible output set* is the set of $\mathbf{u}$ values where $\mathbf{\Phi}(\mathbf{u}, \mathbf{w}, \mathbf{v}) = \mathbf{0}$. Thus, it is the set

$$A = \{\mathbf{u} \in \Re^{n_u} : \mathbf{\Phi}(\mathbf{u}, \mathbf{w}, \mathbf{v}) = \mathbf{0} \text{ for some } \mathbf{w} \text{ and } \mathbf{v}\} \tag{5.2}$$

A necessary condition for a point $\mathbf{q}^*$ to be on the boundary of the accessible output set is [28, 55] that the sub-Jacobian $\mathbf{\Phi}_{\mathbf{w},\mathbf{v}}(\mathbf{q}^*)$ is row rank deficient, which is equivalent to the existence of a unit vector $\boldsymbol{\xi} \in \Re^m$ such that

$$\mathbf{\Phi_z}^T\boldsymbol{\xi} = \mathbf{0} \tag{5.3a}$$

$$\boldsymbol{\xi}^T\boldsymbol{\xi} = 1 \tag{5.3b}$$

where, for notational convenience, $\left[\mathbf{w}^T, \mathbf{v}^T\right]^T$ is denoted by $\mathbf{z}$.

Since points on the boundary must satisfy the constraint equations (5.1), denoting $\mathbf{x} := \left[\mathbf{q}^T, \boldsymbol{\xi}^T\right]^T$, the set of equations that characterize candidate points on the boundary of the accessible output set is

$$\mathbf{G}(\mathbf{x}) = \begin{bmatrix} \boldsymbol{\Phi}(\mathbf{q}) \\ \boldsymbol{\Phi}_{\mathbf{z}}(\mathbf{q})^T \boldsymbol{\xi} \\ \boldsymbol{\xi}^T \boldsymbol{\xi} - 1 \end{bmatrix} = \mathbf{0} \tag{5.4}$$

The practical approach used in [27] to map the boundary, when $dim\ \mathbf{u} \geq 3$, is to define a plane (or a hyperplane for $dim\ \mathbf{u} > 3$) that intersects the boundary in one-dimensional sets and to use continuation methods to map these planar curves.

As noted in [55], the continuation method can be reliably used to map closed curves on the boundary of the accessible output set, as long as the matrix $\mathbf{G}_{\mathbf{x}}$ has full row rank. However, cases arise in applications in which points are encountered on the continuation curves at which $\mathbf{G}_{\mathbf{x}}$ is row rank deficient, called bifurcation points.

In several real applications [27], points were found on continuation curves for which the sub-Jacobian matrix $\boldsymbol{\Phi}_{\mathbf{z}}$ appearing in equations (5.3a) and (5.4) has row rank deficiency two or three. At bifurcation points where $\boldsymbol{\Phi}_{\mathbf{z}}$ has row rank deficiency one or two, methods for calculation of the tangent vectors of the continuation curves are given in [55] and [44]. The purpose of this chapter is to provide a method for finding tangents at the bifurcation points for which $\boldsymbol{\Phi}_{\mathbf{z}}$ has row rank deficiency three and $\mathbf{G}_{\mathbf{x}}$ is row rank deficient.

5.2 Finding All $\boldsymbol{\xi}$ Vectors at a Bifurcation Point

Consider a point $\mathbf{q}^* = \left[\mathbf{u}^{*T}, \mathbf{w}^{*T}, \mathbf{v}^{*T}\right]^T$ on the boundary of the accessible output set, such that $\boldsymbol{\Phi}_{\mathbf{z}} := \boldsymbol{\Phi}_{\mathbf{z}}(\mathbf{q}^*)$ has row rank deficiency three. This case was encountered in several practical applications, and such a point is a multiple bifurcation point. Since the row rank deficiency is three, $dim(ker(\mathbf{\Phi}_{\mathbf{z}}^T(\mathbf{q}^*))) = 3$, so there exist orthonormal vectors $\boldsymbol{\xi}_1$, $\boldsymbol{\xi}_2$, and $\boldsymbol{\xi}_3$ such that

$$\mathbf{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi}_i = \mathbf{0} \tag{5.5a}$$

$$\boldsymbol{\xi}_i^T\boldsymbol{\xi}_j = \delta_{ij},\ \ i,j = 1,2,3 \tag{5.5b}$$

Every unit vector $\boldsymbol{\xi}$ in $ker(\mathbf{\Phi}_{\mathbf{z}}^T(\mathbf{q}^*))$, can be represented as a linear combination of $\boldsymbol{\xi}_1$, $\boldsymbol{\xi}_2$, and $\boldsymbol{\xi}_3$. For convenience, spherical coordinates are used for the coefficients, which have the advantage of preserving the unitary norm, and allow simplified algebraic manipulations; i.e.,

$$\boldsymbol{\xi} \equiv \boldsymbol{\xi}(\phi,\theta) = \sin(\phi)\cos(\theta)\boldsymbol{\xi}_1 + \sin(\phi)\sin(\theta)\boldsymbol{\xi}_2 + \cos(\phi)\boldsymbol{\xi}_3 \tag{5.6}$$

where $0 \leq \phi \leq \pi$, $0 \leq \theta < 2\pi$.

The goal is to find all vectors $\boldsymbol{\xi} \in ker(\mathbf{\Phi}_{\mathbf{z}}^T(\mathbf{q}^*))$ for which the Jacobian matrix $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*,\boldsymbol{\xi})$ is row rank deficient; i.e.,

$$\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*,\boldsymbol{\xi}) = \begin{bmatrix} \mathbf{\Phi}_{\mathbf{y}} & \mathbf{\Phi}_{\mathbf{v}} & \mathbf{0} \\ (\mathbf{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi})_{\mathbf{y}} & (\mathbf{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi})_{\mathbf{v}} & \mathbf{\Phi}_{\mathbf{z}}^T \\ \mathbf{0} & \mathbf{0} & 2\boldsymbol{\xi}^T \end{bmatrix} \tag{5.7}$$

is row rank deficient, where the notation $\mathbf{y} = \left[\mathbf{u}^T, \mathbf{w}^T\right]^T$ is used.

The following result holds (see [44]):

Lemma 5.2.1 $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*,\boldsymbol{\xi})$ *is row rank deficient if and only if* $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*,-\boldsymbol{\xi})$ *is row rank deficient.*

Because of this result, the search for $\boldsymbol{\xi}$ can be restricted to the upper half-space; i.e., $0 \leq \phi \leq \pi/2$.

Now, $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\xi})$ is row rank deficient if and only if there exists a non-zero vector $\mathbf{c} = \left[\mathbf{c}_1{}^T, \mathbf{c}_2{}^T, c_3\right]^T$, where $\mathbf{c}_1 \in \Re^m, \mathbf{c}_2 \in \Re^{n_z}$, and $c_3 \in \Re$, such that

$$\mathbf{c}^T \mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\xi}) = \mathbf{0} \tag{5.8}$$

which means, by equation (5.7), that

$$\mathbf{c}_1{}^T \boldsymbol{\Phi}_{\mathbf{y}} + \mathbf{c}_2{}^T (\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi})_{\mathbf{y}} = \mathbf{0} \tag{5.9a}$$

$$\mathbf{c}_1{}^T \boldsymbol{\Phi}_{\mathbf{v}} + \mathbf{c}_2{}^T (\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi})_{\mathbf{v}} = \mathbf{0} \tag{5.9b}$$

$$\mathbf{c}_2{}^T \boldsymbol{\Phi}_{\mathbf{z}}{}^T + 2c_3 \boldsymbol{\xi}^T = \mathbf{0} \tag{5.9c}$$

Multiplying equation (5.9c) on the right by $\boldsymbol{\xi}$, and using the fact that $\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi} = \mathbf{0}$, shows that $c_3 = 0$. Therefore, from equation (5.9c), $\boldsymbol{\Phi}_{\mathbf{z}} \mathbf{c}_2 = \mathbf{0}$; i.e., $\mathbf{c}_2$ is in $ker(\boldsymbol{\Phi}_{\mathbf{z}})$. Also, $\mathbf{c}_2 \neq \mathbf{0}$. Otherwise, by equation (5.9a), since $\boldsymbol{\Phi}_{\mathbf{y}}$ is invertible, $\mathbf{c}_1 = \mathbf{0}$, which contradicts the fact that $\mathbf{c} \neq \mathbf{0}$.

Now, $\boldsymbol{\Phi}_{\mathbf{z}} : \Re^{n_z} \to \Re^m$ has row rank deficiency three, so $dim(ker(\boldsymbol{\Phi}_{\mathbf{z}})) = n_z + 3 - m := k$. Thus, there exists an orthonormal basis of $ker(\boldsymbol{\Phi}_{\mathbf{z}})$, say $\{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_k\}$ such that, since $\mathbf{c}_2 \in ker(\boldsymbol{\Phi}_{\mathbf{z}})$,

$$\mathbf{c}_2 = d_1 \boldsymbol{\alpha}_1 + d_2 \boldsymbol{\alpha}_2 + ... + d_k \boldsymbol{\alpha}_k := \boldsymbol{\alpha} \mathbf{d} \tag{5.10}$$

where $\mathbf{d} = [d_1, d_2, ..., d_k]^T$ and $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_k]$, with $\mathbf{d} \neq \mathbf{0}$ since $\mathbf{c}_2 \neq \mathbf{0}$.

From equation (5.9a), since $\boldsymbol{\Phi}_{\mathbf{y}}$ is invertible,

$$\mathbf{c}_1{}^T = -\mathbf{c}_2{}^T (\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi})_{\mathbf{y}} (\boldsymbol{\Phi}_{\mathbf{y}})^{-1} \tag{5.11}$$

which, substituted in equation (5.9b), yields

$$\mathbf{c}_2^T \left[(\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi})_{\mathbf{y}} (-\boldsymbol{\Phi}_{\mathbf{y}}^{-1} \boldsymbol{\Phi}_{\mathbf{v}}) + (\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi})_{\mathbf{v}} \right] = \mathbf{0} \tag{5.12}$$

Substituting equation (5.10) into equation (5.12) gives

$$\mathbf{d}^T \boldsymbol{\alpha}^T \left[(\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi})_{\mathbf{y}} (-\boldsymbol{\Phi}_{\mathbf{y}}^{-1} \boldsymbol{\Phi}_{\mathbf{v}}) + (\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi})_{\mathbf{v}} \right] = \mathbf{0} \tag{5.13}$$

Defining

$$\mathbf{H}(\boldsymbol{\xi}(\phi, \theta)) := \boldsymbol{\alpha}^T \left[(\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi})_{\mathbf{y}} (-\boldsymbol{\Phi}_{\mathbf{y}}^{-1} \boldsymbol{\Phi}_{\mathbf{v}}) + (\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi})_{\mathbf{v}} \right] \tag{5.14}$$

and since $\mathbf{d} \neq \mathbf{0}$, the above shows that $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\xi}(\phi, \theta))$ row rank deficient implies that $\mathbf{H}(\boldsymbol{\xi}(\phi, \theta))$ is row rank deficient.

Assume now that $\mathbf{H}(\boldsymbol{\xi}(\phi, \theta))$ is row rank deficient. Thus there is a vector $\mathbf{d} = [d_1, d_2, ..., d_k]^T \in \Re^k$, $\mathbf{d} \neq \mathbf{0}$, such that

$$\mathbf{d}^T \mathbf{H} = \mathbf{0} \tag{5.15}$$

or,

$$\mathbf{d}^T \boldsymbol{\alpha}^T \left[(\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi})_{\mathbf{y}} (-\boldsymbol{\Phi}_{\mathbf{y}}^{-1} \boldsymbol{\Phi}_{\mathbf{v}}) + (\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi})_{\mathbf{v}} \right] = \mathbf{0} \tag{5.16}$$

where $\{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_k\}$ is the above basis of $ker(\boldsymbol{\Phi}_{\mathbf{z}})$ and $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_k]$.

Let $\mathbf{c}_2 \in ker(\boldsymbol{\Phi}_{\mathbf{z}})$ be defined by

$$\mathbf{c}_2 = \boldsymbol{\alpha}\mathbf{d} = d_1\boldsymbol{\alpha}_1 + d_2\boldsymbol{\alpha}_2 + ... + d_k\boldsymbol{\alpha}_k \tag{5.17}$$

Now, since $\mathbf{d} \neq \mathbf{0}$, $\mathbf{c}_2 \neq \mathbf{0}$, and then equation 5.15 becomes

$$\mathbf{c}_2^T \left[(\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi})_{\mathbf{y}} (-\boldsymbol{\Phi}_{\mathbf{y}}^{-1} \boldsymbol{\Phi}_{\mathbf{v}}) + (\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi})_{\mathbf{v}} \right] = \mathbf{0} \tag{5.18}$$

Let $\mathbf{c}_1 \in \Re^m$ be defined by

$$\mathbf{c}_1{}^T = -\mathbf{c}_2{}^T(\mathbf{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi})_{\mathbf{y}}(\mathbf{\Phi}_{\mathbf{y}})^{-1} \tag{5.19}$$

Then, equation (5.18) can be rewritten as

$$\mathbf{c}_1{}^T\mathbf{\Phi}_{\mathbf{v}} + \mathbf{c}_2{}^T(\mathbf{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi})_{\mathbf{v}} = \mathbf{0} \tag{5.20}$$

Now, since $\mathbf{c}_2 \in ker(\mathbf{\Phi}_{\mathbf{z}})$, $\mathbf{\Phi}_{\mathbf{z}}\mathbf{c}_2 = \mathbf{0}$, or

$$\mathbf{c}_2{}^T\mathbf{\Phi}_{\mathbf{z}}{}^T = \mathbf{0} \tag{5.21}$$

Let $c_3 = 0$. Then, from equation (5.21),

$$\mathbf{c}_2{}^T\mathbf{\Phi}_{\mathbf{z}}{}^T + 2c_3\boldsymbol{\xi}^T = \mathbf{0} \tag{5.22}$$

Also,

$$\mathbf{c}_1{}^T\mathbf{\Phi}_{\mathbf{y}} + \mathbf{c}_2{}^T(\mathbf{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi})_{\mathbf{y}} = -\mathbf{c}_2{}^T(\mathbf{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi})_{\mathbf{y}}(\mathbf{\Phi}_{\mathbf{y}})^{-1}\mathbf{\Phi}_{\mathbf{y}} + \mathbf{c}_2{}^T(\mathbf{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi})_{\mathbf{y}} = \mathbf{0} \tag{5.23}$$

Combining equations (5.23), (5.20), and (5.22), yields

$$\mathbf{c}_1{}^T\mathbf{\Phi}_{\mathbf{y}} + \mathbf{c}_2{}^T(\mathbf{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi})_{\mathbf{y}} = \mathbf{0} \tag{5.24a}$$

$$\mathbf{c}_1{}^T\mathbf{\Phi}_{\mathbf{v}} + \mathbf{c}_2{}^T(\mathbf{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi})_{\mathbf{v}} = \mathbf{0} \tag{5.24b}$$

$$\mathbf{c}_2{}^T\mathbf{\Phi}_{\mathbf{z}}{}^T + 2c_3\boldsymbol{\xi}^T = \mathbf{0} \tag{5.24c}$$

which, by equation (5.7), is equivalent to

$$\mathbf{c}^T\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\xi}) = \mathbf{0} \tag{5.25}$$

where $\mathbf{c} \neq \mathbf{0}$, since $\mathbf{c}_2 \neq \mathbf{0}$. This implies $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\xi})$ is row rank deficient.

The above analysis establishes the following result (see also [44]) :

Lemma 5.2.2 $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\xi}(\phi, \theta))$ *is row rank deficient if and only if* $\mathbf{H}(\boldsymbol{\xi}(\phi, \theta))$ *is row* *rank deficient.*

Using the expression for $\boldsymbol{\xi}$ from equation (5.6), in equation (5.14) gives

$$\mathbf{H}(\boldsymbol{\xi}(\phi,\theta)) = \sin(\phi)\cos(\theta)\boldsymbol{\alpha}^T\left[(\boldsymbol{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi}_1)_{\mathbf{y}}(-\boldsymbol{\Phi}_{\mathbf{y}}^{-1}\boldsymbol{\Phi}_{\mathbf{v}}) + (\boldsymbol{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi}_1)_{\mathbf{v}}\right]$$

$$+ \sin(\phi)\sin(\theta)\boldsymbol{\alpha}^T\left[(\boldsymbol{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi}_2)_{\mathbf{y}}(-\boldsymbol{\Phi}_{\mathbf{y}}^{-1}\boldsymbol{\Phi}_{\mathbf{v}}) + (\boldsymbol{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi}_2)_{\mathbf{v}}\right]$$

$$+ \cos(\phi)\boldsymbol{\alpha}^T\left[(\boldsymbol{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi}_3)_{\mathbf{y}}(-\boldsymbol{\Phi}_{\mathbf{y}}^{-1}\boldsymbol{\Phi}_{\mathbf{v}}) + (\boldsymbol{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi}_3)_{\mathbf{v}}\right] \tag{5.26}$$

Denoting

$$\mathbf{A} := \boldsymbol{\alpha}^T\left[(\boldsymbol{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi}_1)_{\mathbf{y}}(-\boldsymbol{\Phi}_{\mathbf{y}}^{-1}\boldsymbol{\Phi}_{\mathbf{v}}) + (\boldsymbol{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi}_1)_{\mathbf{v}}\right] \tag{5.27}$$

$$\mathbf{B} := \boldsymbol{\alpha}^T\left[(\boldsymbol{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi}_2)_{\mathbf{y}}(-\boldsymbol{\Phi}_{\mathbf{y}}^{-1}\boldsymbol{\Phi}_{\mathbf{v}}) + (\boldsymbol{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi}_2)_{\mathbf{v}}\right] \tag{5.28}$$

$$\mathbf{C} := \boldsymbol{\alpha}^T\left[(\boldsymbol{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi}_3)_{\mathbf{y}}(-\boldsymbol{\Phi}_{\mathbf{y}}^{-1}\boldsymbol{\Phi}_{\mathbf{v}}) + (\boldsymbol{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi}_3)_{\mathbf{v}}\right] \tag{5.29}$$

simplifies equation (5.26) to

$$\mathbf{H}(\boldsymbol{\xi}(\phi,\theta)) = \sin(\phi)\cos(\theta)\mathbf{A} + \sin(\phi)\sin(\theta)\mathbf{B} + \cos(\phi)\mathbf{C} \tag{5.30}$$

Lemma 5.2.3 *If* $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\xi}_3)$ *has full row rank, then* $\mathbf{C}$ *is right invertible, and a right inverse of* $\mathbf{C}$ *is the Moore-Penrose inverse (see [2]);* $\mathbf{C}^+ = \mathbf{C}^T(\mathbf{C}\mathbf{C}^T)^{-1}$.

Proof: Since $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\xi}_3)$ has full row rank, by Lemma 5.2.2, $\mathbf{H}(\boldsymbol{\xi}_3)$ has full row rank. But, by equation (5.6), $\boldsymbol{\xi}_3 = \boldsymbol{\xi}(0,\theta)$ for any $0 \le \theta < 2\pi$, so $\mathbf{C} = \mathbf{H}(\boldsymbol{\xi}(0,\theta)) = \mathbf{H}(\boldsymbol{\xi}_3)$ has full row rank. This implies that $\mathbf{C}\mathbf{C}^T$ is invertible, so $\mathbf{C}^+ = \mathbf{C}^T(\mathbf{C}\mathbf{C}^T)^{-1}$ exists and

$$\mathbf{C}\mathbf{C}^+ = \mathbf{C}(\mathbf{C}^T(\mathbf{C}\mathbf{C}^T)^{-1}) = (\mathbf{C}\mathbf{C}^T)(\mathbf{C}\mathbf{C}^T)^{-1} = \mathbf{I}$$

Thus $\mathbf{C}$ is right invertible, with right inverse $\mathbf{C}^+$. $\square$

The problem of finding all ϕ and θ such that $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\xi}(\phi,\theta))$ is row rank deficient is equivalent, by Lemma 5.2.2, to finding all ϕ and θ such that $\mathbf{H}(\boldsymbol{\xi}(\phi,\theta))$ is row rank deficient, or

$$\sin(\phi)\cos(\theta)\mathbf{A} + \sin(\phi)\sin(\theta)\mathbf{B} + \cos(\phi)\mathbf{C} \;\text{ is row rank deficient} \tag{5.31}$$

If $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\xi})$ is row rank deficient for every $\boldsymbol{\xi}$ in $ker(\boldsymbol{\Phi}_{\mathbf{z}}^T(\mathbf{q}^*))$, then the problem of finding $\boldsymbol{\xi}$ is solved. Assume, otherwise, that $\boldsymbol{\xi}_3$ is chosen such that $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\xi}_3)$ has full row rank. Then ϕ in equation (5.6) cannot be 0, so equation(5.31) is equivalent to

$$\cos(\theta)\mathbf{A} + \sin(\theta)\mathbf{B} + \cot(\phi)\mathbf{C} \;\text{ is row rank deficient} \tag{5.32}$$

Also, by Lemma 5.2.3, $\mathbf{C}$ is right invertible, so equation (5.32) is equivalent to

$$\cot(\phi)\mathbf{I} + [\cos(\theta)\mathbf{A} + \sin(\theta)\mathbf{B}]\,\mathbf{C}^+ \;\text{ is row rank deficient} \tag{5.33}$$

Denoting

$$\mathbf{D}(\theta) := [\cos(\theta)\mathbf{A} + \sin(\theta)\mathbf{B}] \tag{5.34}$$

the following result holds:

Theorem 5.2.1 *For each θ in $[0, 2\pi)$, $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\xi}(\phi,\theta))$ is row rank deficient if and only if $\cot(\phi)$ is an eigenvalue of $-\mathbf{D}(\theta)\mathbf{C}^+$.*

Proof: Since $\boldsymbol{\xi}_3$ was chosen such that $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\xi}_3)$ has full row rank, by equation (5.33), $\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\xi}(\phi,\theta))$ is row rank deficient if and only if

$$\cot(\phi)\mathbf{I} + \mathbf{D}(\theta)\mathbf{C}^+ \;\text{ is row rank deficient} \tag{5.35}$$

Since $\mathbf{D}(\theta)\mathbf{C}^+$ is a square matrix, this is equivalent to $\cot(\phi)$ being an eigenvalue of

$-\mathbf{D}(\theta)\mathbf{C}^+$, for each $0 \leq \theta < 2\pi$. $\square$

5.3 Finding All Tangents to Continuation Curves

To find the tangents at multiple bifurcation points, proceed as follows. Let θ be fixed in $[0, 2\pi)$. Let $\boldsymbol{\eta}$ be an eigenvector of $-\mathbf{D}(\theta)\mathbf{C}^+$ corresponding to this value of θ; i.e.,

$$-\mathbf{D}(\theta)\mathbf{C}^+\boldsymbol{\eta} = \cot(\phi)\boldsymbol{\eta} \tag{5.36}$$

Note that only the real eigenvalues need to be considered, since $\sin(\phi)$ and $\cos(\phi)$ appear as coefficients in the expansion of the real vector $\boldsymbol{\xi}$ in terms of the elements of the basis of $ker(\boldsymbol{\Phi}_{\mathbf{z}}^T(\mathbf{q}^*))$.

Defining $\boldsymbol{\zeta}_2 := \mathbf{C}^+\boldsymbol{\eta}$, then $\boldsymbol{\eta} = \mathbf{C}\boldsymbol{\zeta}_2$ and equation (5.36) becomes

$$-\mathbf{D}(\theta)\boldsymbol{\zeta}_2 = \cot(\phi)\mathbf{C}\boldsymbol{\zeta}_2 \tag{5.37}$$

or,

$$\left[\cot(\phi)\mathbf{C} + \mathbf{D}(\theta)\right]\boldsymbol{\zeta}_2 = \mathbf{0} \tag{5.38}$$

i.e.,

$$\left[\cot(\phi)\mathbf{C} + \cos(\theta)\mathbf{A} + \sin(\theta)\mathbf{B}\right]\boldsymbol{\zeta}_2 = \mathbf{0} \tag{5.39}$$

Since $\sin(\phi) \neq 0$ by the choice of $\boldsymbol{\xi}_3$, this is equivalent to

$$\left[\cos(\phi)\mathbf{C} + \sin(\phi)\cos(\theta)\mathbf{A} + \sin(\phi)\sin(\theta)\mathbf{B}\right]\boldsymbol{\zeta}_2 = \mathbf{0} \tag{5.40}$$

Using equations (5.14) and (5.30), equation (5.40) can be written as

$$\boldsymbol{\alpha}^T\left[(\boldsymbol{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi})_{\mathbf{y}}(-\boldsymbol{\Phi}_{\mathbf{y}}^{-1}\boldsymbol{\Phi}_{\mathbf{v}}) + (\boldsymbol{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi})_{\mathbf{v}}\right]\boldsymbol{\zeta}_2 = \mathbf{0} \tag{5.41}$$

and, denoting

$$\mathbf{s} = \left[(\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi})_{\mathbf{y}} (-\mathbf{\Phi}_{\mathbf{y}}^{-1} \mathbf{\Phi}_{\mathbf{v}}) + (\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi})_{\mathbf{v}} \right] \boldsymbol{\zeta}_2 \tag{5.42}$$

equation (5.41) becomes

$$\boldsymbol{\alpha}^T \mathbf{s} = \mathbf{0} \tag{5.43}$$

which implies that $\mathbf{s}$ is orthogonal to $\boldsymbol{\alpha}_i, i = 1, 2, ..., k$. Since $\{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_k\}$ is an orthonormal basis of $ker(\mathbf{\Phi}_{\mathbf{z}})$, $\mathbf{s} \in (ker(\mathbf{\Phi}_{\mathbf{z}}))^{\perp} = range(\mathbf{\Phi}_{\mathbf{z}}{}^T)$ (see [35]). Restricting the domain of $\mathbf{\Phi}_{\mathbf{z}}{}^T$ to $(ker(\mathbf{\Phi}_{\mathbf{z}}))^{\perp}$, then it can be regarded as one-to-one and onto linear transformation from $(ker(\mathbf{\Phi}_{\mathbf{z}}))^{\perp}$ onto $range(\mathbf{\Phi}_{\mathbf{z}}{}^T)$ (see [35]). Thus, there exists a unique vector $\boldsymbol{\zeta}_3 \in (ker(\mathbf{\Phi}_{\mathbf{z}}{}^T))^{\perp} = range(\mathbf{\Phi}_{\mathbf{z}})$ such that

$$\mathbf{s} = -\mathbf{\Phi}_{\mathbf{z}}{}^T \boldsymbol{\zeta}_3 \tag{5.44}$$

Using the definition of $\mathbf{s}$ in equation (5.42),

$$\left[(\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi})_{\mathbf{y}} (-\mathbf{\Phi}_{\mathbf{y}}^{-1} \mathbf{\Phi}_{\mathbf{v}}) + (\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi})_{\mathbf{v}} \right] \boldsymbol{\zeta}_2 = -\mathbf{\Phi}_{\mathbf{z}}{}^T \boldsymbol{\zeta}_3 \tag{5.45}$$

and defining

$$\boldsymbol{\zeta}_1 := (-\mathbf{\Phi}_{\mathbf{y}}^{-1} \mathbf{\Phi}_{\mathbf{v}}) \boldsymbol{\zeta}_2 \tag{5.46}$$

equations (5.46) and (5.45) yield the following system of equations:

$$\mathbf{\Phi}_{\mathbf{y}} \boldsymbol{\zeta}_1 + \mathbf{\Phi}_{\mathbf{v}} \boldsymbol{\zeta}_2 = \mathbf{0} \tag{5.47a}$$

$$(\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi}(\phi, \theta))_{\mathbf{y}} \boldsymbol{\zeta}_1 + (\mathbf{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi}(\phi, \theta))_{\mathbf{v}} \boldsymbol{\zeta}_2 + \mathbf{\Phi}_{\mathbf{z}}{}^T \boldsymbol{\zeta}_3 = \mathbf{0} \tag{5.47b}$$

$$\boldsymbol{\xi}(\phi, \theta)^T \boldsymbol{\zeta}_3 = \mathbf{0} \tag{5.47c}$$

the last being a consequence of the fact that $\boldsymbol{\zeta}_3 \in (ker(\mathbf{\Phi}_{\mathbf{z}}{}^T))^{\perp}$ and $\boldsymbol{\xi}$ is in $ker(\mathbf{\Phi}_{\mathbf{z}}^T)$.

After multiplying the last equation by 2, this system can be rewritten as

$$\begin{bmatrix} \mathbf{\Phi_y} & \mathbf{\Phi_v} & \mathbf{0} \\ (\mathbf{\Phi_z}^T\boldsymbol{\xi}(\phi,\theta))_{\mathbf{y}} & (\mathbf{\Phi_z}^T\boldsymbol{\xi}(\phi,\theta))_{\mathbf{v}} & \mathbf{\Phi_z}^T \\ \mathbf{0} & \mathbf{0} & 2\boldsymbol{\xi}^T(\phi,\theta) \end{bmatrix} \begin{bmatrix} \boldsymbol{\zeta}_1 \\ \boldsymbol{\zeta}_2 \\ \boldsymbol{\zeta}_3 \end{bmatrix} = \mathbf{0} \tag{5.48}$$

which can be written, using equation (5.7), as

$$\mathbf{G_x}(\mathbf{q}^*,\gamma(\phi,\theta))\boldsymbol{\zeta} = \mathbf{0} \tag{5.49}$$

where $\boldsymbol{\zeta} = \left[\boldsymbol{\zeta}_1{}^T, \boldsymbol{\zeta}_2{}^T, \boldsymbol{\zeta}_3{}^T\right]^T \in \Re^{m+n_q}$. This implies that $\boldsymbol{\zeta}$ is in the null space of $\mathbf{G_x}(\mathbf{q}^*,\gamma(\phi,\theta))$, $ker(\mathbf{G_x})$. This is the first-order necessary condition for $\boldsymbol{\zeta}$ to be tangent to a continuation curve at $\mathbf{q}^*$. Normalize the vector $\boldsymbol{\zeta}$, and denote the unit vector obtained also by $\boldsymbol{\zeta}$. Recall that since $\boldsymbol{\eta}$ is an eigenvector, $\boldsymbol{\zeta}_2 \neq \mathbf{0}$, so $\boldsymbol{\zeta} \neq \mathbf{0}$.

A second-order necessary condition for a vector $\boldsymbol{\zeta} \in ker(\mathbf{G_x}(\mathbf{q}^*,\boldsymbol{\xi}))$ to be tangent to a continuation curve at $(\mathbf{q}^*,\boldsymbol{\xi})$ is (see [2]) that

$$\boldsymbol{\zeta}^T(\mathbf{e}_i^T\mathbf{G_x}(\mathbf{q}^*,\boldsymbol{\xi})_{\mathbf{x}}\boldsymbol{\zeta} = \mathbf{0}, \quad \text{i=1,2,...,n} \tag{5.50}$$

where $\{\mathbf{e}_1,\mathbf{e}_2,...,\mathbf{e}_n\}$ spans $ker(\mathbf{G_x}{}^T(\mathbf{q}^*,\boldsymbol{\xi}))$, and $\mathbf{G_x}(\mathbf{q}^*,\boldsymbol{\xi})$ has row rank deficiency n.

Proposition 5.3.1 *If $\mathbf{\Phi_z}^T(\mathbf{q}^*)$ has row rank deficiency 3, then $\mathbf{G_x}(\mathbf{q}^*,\boldsymbol{\xi}(\phi,\theta))$ has row rank deficiency greater than or equal to 2.*

Proof. To prove this proposition, it is enough to show that

$$dim(ker(\mathbf{G_x}(\mathbf{q}^*,\boldsymbol{\xi}(\phi,\theta)))) \geq 3$$

i.e., there are three linearly independent vectors in $ker(\mathbf{G_x}(\mathbf{q}^*,\boldsymbol{\xi}(\phi,\theta)))$.

Recalling that θ is being kept fixed in $[0,2\pi)$, define $\boldsymbol{\zeta}^* := \left[\mathbf{0},\mathbf{0},(\frac{d\boldsymbol{\xi}}{d\phi})^T\right]^T \in \Re^{n_y+n_v+m}$, where

$$\frac{d\boldsymbol{\xi}}{d\phi} = \cos(\phi)\cos(\theta)\boldsymbol{\xi}_1 + \cos(\phi)\sin(\theta)\boldsymbol{\xi}_2 - \sin(\phi)\boldsymbol{\xi}_3 \tag{5.51}$$

Then,

$$\begin{aligned}\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\xi}(\phi,\theta))\boldsymbol{\zeta}^* &= \begin{bmatrix} \boldsymbol{\Phi}_\mathbf{y} & \boldsymbol{\Phi}_\mathbf{v} & \mathbf{0} \\ (\boldsymbol{\Phi}_\mathbf{z}^T\boldsymbol{\xi}(\phi,\theta))_\mathbf{y} & (\boldsymbol{\Phi}_\mathbf{z}^T\boldsymbol{\xi}(\phi,\theta))_\mathbf{v} & \boldsymbol{\Phi}_\mathbf{z}^T \\ \mathbf{0} & \mathbf{0} & 2\boldsymbol{\xi}^T(\phi,\theta) \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \frac{d\boldsymbol{\xi}}{d\phi} \end{bmatrix} = \\ &= \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\Phi}_\mathbf{z}^T \frac{d\boldsymbol{\xi}}{d\phi} \\ 2\boldsymbol{\xi}^T \frac{d\boldsymbol{\xi}}{d\phi} \end{bmatrix}\end{aligned} \tag{5.52}$$

Now,

$$\boldsymbol{\Phi}_\mathbf{z}^T \frac{d\boldsymbol{\xi}}{d\phi} = \boldsymbol{\Phi}_\mathbf{z}^T(\cos(\phi)\cos(\theta)\boldsymbol{\xi}_1 + \cos(\phi)\sin(\theta)\boldsymbol{\xi}_2 - \sin(\phi)\boldsymbol{\xi}_3) = \mathbf{0} \tag{5.53}$$

since $\boldsymbol{\xi}_i$ is, by definition, in $ker(\boldsymbol{\Phi}_\mathbf{z}^T(\mathbf{q}^*))$, $i = 1, 2, 3$. Also, since $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2$, and $\boldsymbol{\xi}_3$ are mutually orthonormal,

$$\begin{aligned}2\boldsymbol{\xi}^T \tfrac{d\boldsymbol{\xi}}{d\phi} &= 2(\sin(\phi)\cos(\theta)\boldsymbol{\xi}_1 + \sin(\phi)\sin(\theta)\boldsymbol{\xi}_2 + \cos(\phi)\boldsymbol{\xi}_3) \\ &\quad (\cos(\phi)\cos(\theta)\boldsymbol{\xi}_1 + \cos(\phi)\sin(\theta)\boldsymbol{\xi}_2 - \sin(\phi)\boldsymbol{\xi}_3) \\ &= 2(\sin(\phi)\cos(\phi)\cos^2(\theta) + \sin(\phi)\cos(\phi)\sin^2(\theta) - \sin(\phi)\cos(\phi)) \\ &= 2\sin(\phi)\cos(\phi)(\cos^2(\theta) + \sin^2(\theta) - 1) = 0\end{aligned} \tag{5.54}$$

which proves, by equation (5.52), that

$$\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\xi}(\phi,\theta))\boldsymbol{\zeta}^* = 0 \tag{5.55}$$

or $\boldsymbol{\zeta}^* \in ker(\mathbf{G_x}(\mathbf{q}^*, \boldsymbol{\xi}(\phi,\theta)))$.

Similarly, define $\boldsymbol{\zeta}_* := \left[\mathbf{0}, \mathbf{0}, \boldsymbol{\zeta}_{*3}^T\right]^T \in \Re^{n_y+n_v+m}$, where

$$\boldsymbol{\zeta}_{*3}^T = -\sin(\phi)\sin(\theta)\boldsymbol{\xi}_1 + \sin(\phi)\cos(\theta)\boldsymbol{\xi}_2 \tag{5.56}$$

It can be shown, as above, that

$$\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\xi}(\phi,\theta))\boldsymbol{\zeta}_* = 0 \tag{5.57}$$

so $\boldsymbol{\zeta}_* \in ker(\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\xi}(\phi,\theta)))$.

It is easy to verify that the vectors $\boldsymbol{\zeta}$, $\boldsymbol{\zeta}^*$, and $\boldsymbol{\zeta}_*$ in $ker(\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\xi}(\phi,\theta)))$ are linearly independent, so $dim(ker(\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\xi}(\phi,\theta))))$ is at least 3, which proves that $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\xi}(\phi,\theta))$ has row rank deficiency greater than or equal to 2. $\square$

There are two cases.

Case 1. Assume that $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\xi}(\phi,\theta))$ has row rank deficiency 2.

By the proof of Proposition 5.3.1, any tangent to a continuation curve at $\mathbf{q}^*$, associated with $\boldsymbol{\xi}(\phi,\theta)$, will be of the form

$$\boldsymbol{\tau} = a\boldsymbol{\zeta} + b\boldsymbol{\zeta}^* + c\boldsymbol{\zeta}_* = \begin{bmatrix} a\boldsymbol{\zeta}_1 \\ a\boldsymbol{\zeta}_2 \\ a\boldsymbol{\zeta}_3 + b\frac{d\boldsymbol{\xi}}{d\phi} + c\boldsymbol{\zeta}_{*3} \end{bmatrix} \tag{5.58}$$

where $a, b, c \in \Re$, and $\boldsymbol{\zeta}_*$ has been normalized. Clearly, $\boldsymbol{\zeta}^*$ is unitary and $\boldsymbol{\zeta}$ was normalized earlier. By the above, the following result is proved.

Theorem 5.3.1 *For each θ in $[0, 2\pi)$, let $\phi \in [0, \pi/2]$ be such that cot(ϕ) is an eigenvalue of the matrix $-\mathbf{D}(\theta)\mathbf{C}^+$. Let $\boldsymbol{\eta}$ be a corresponding eigenvector of $-\mathbf{D}(\theta)\mathbf{C}^+$. Define $\boldsymbol{\zeta} = \left[\boldsymbol{\zeta}_1^T, \boldsymbol{\zeta}_2^T, \boldsymbol{\zeta}_3^T\right]^T$, where $\boldsymbol{\zeta}_1 := (-\boldsymbol{\Phi}_{\mathbf{y}}^{-1}\boldsymbol{\Phi}_{\mathbf{v}})\boldsymbol{\zeta}_2$, $\boldsymbol{\zeta}_2 := \mathbf{C}^+\boldsymbol{\eta}$ and $\boldsymbol{\zeta}_3$ is the unique solution of*

$$\mathbf{\Phi}_{\mathbf{z}}^T\boldsymbol{\zeta}_3 = (\mathbf{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi})_{\mathbf{y}}\left(-\mathbf{\Phi}_{\mathbf{y}}^{-1}\mathbf{\Phi}_{\mathbf{v}}\right) + (\mathbf{\Phi}_{\mathbf{z}}^T\boldsymbol{\xi})_{\mathbf{v}}\boldsymbol{\zeta}_2 \tag{5.59a}$$

$$\boldsymbol{\xi}_1^T\boldsymbol{\zeta}_3 = 0 \tag{5.59b}$$

$$\boldsymbol{\xi}_2^T\boldsymbol{\zeta}_3 = 0 \tag{5.59c}$$

$$\boldsymbol{\xi}_3^T\boldsymbol{\zeta}_3 = 0 \tag{5.59d}$$

Then, any tangent vector to a continuation curve at the multiple bifurcation point $\mathbf{q}^*$ *is of the form*

$$\boldsymbol{\tau} = a\boldsymbol{\zeta} + b\boldsymbol{\zeta}^* + c\boldsymbol{\zeta}_* = \begin{bmatrix} a\boldsymbol{\zeta}_1 \\ a\boldsymbol{\zeta}_2 \\ a\boldsymbol{\zeta}_3 + b\frac{d\boldsymbol{\xi}}{d\phi} + c\boldsymbol{\zeta}_{*3} \end{bmatrix} \tag{5.60}$$

The second-order necessary condition of equation (5.50), applied to $\boldsymbol{\tau}$, becomes

$$\boldsymbol{\tau}^T(\mathbf{e}_i^T\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*,\boldsymbol{\xi})_{\mathbf{x}}\boldsymbol{\tau} = \mathbf{0}, \qquad \text{i} = 1,2 \tag{5.61}$$

where $\{\mathbf{e}_1, \mathbf{e}_2\}$ span $ker({\mathbf{G}_{\mathbf{x}}}^T(\mathbf{q}^*,\boldsymbol{\xi}))$. Recall that $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*,\boldsymbol{\xi})$ has row rank deficiency 2 for **Case 1**, so the dimension of $ker({\mathbf{G}_{\mathbf{x}}}^T(\mathbf{q}^*,\boldsymbol{\xi}))$ is also 2.

The equation above can be used in practice to determine a, b, and c in equation (5.60), combined with the condition that $\|\boldsymbol{\tau}\| = 1$.

Case 2. Assume now that $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*,\boldsymbol{\xi}(\phi,\theta))$ has row rank deficiency greater than or equal to 3.

In this case, when solving for the eigenvalues of $-\mathbf{D}(\theta)\mathbf{C}^+$ in equation (5.35), repeated eigenvalues are obtained. Let $\boldsymbol{\eta}^1, \boldsymbol{\eta}^2, ..., \boldsymbol{\eta}^s$ be linearly independent eigenvectors corresponding to an eigenvalue $\cot(\phi)$ with multiplicity s. For each $\boldsymbol{\eta}^i$, find the corresponding $\boldsymbol{\zeta}^i$ as in Theorem 5.3.1. Then, the following holds

Theorem 5.3.2 *In the above conditions, any tangent vector to a continuation curve* *at* $\mathbf{q}^*$, *associated with* $\boldsymbol{\xi}(\phi, \theta)$, *can be written as*

$$\boldsymbol{\tau} = a_1\boldsymbol{\zeta}^1 + a_2\boldsymbol{\zeta}^2 + ... + a_s\boldsymbol{\zeta}^s + b\boldsymbol{\zeta}^* + c\boldsymbol{\zeta}_* \tag{5.62}$$

where $a_1, a_2, ..., a_s, b, c \in \Re$.

To determine the coefficients $a_1, a_2, ..., a_s, b$, and c, the second-order necessary condition can be used, together with the condition $\|\boldsymbol{\tau}\| = 1$.

Because of the nature of the bifurcation point, there is a continuum of continuation curves at such a point $\mathbf{q}^*$. In practice, the goal is to find the exterior tangent from the collection of tangents existing at this multiple bifurcation point and also to find the interior, cone-shaped region covered by such continuation curves. The next section proposes an algorithm to be called each time the continuation method reaches a multiple bifurcation point at which $\boldsymbol{\Phi}_{\mathbf{z}}$ has row rank deficiency three and $\mathbf{G}_{\mathbf{x}}$ is row rank deficient.

5.4 Numerical Algorithm

To determine tangents at a multiple bifurcation point $\mathbf{q}^*$ with $\boldsymbol{\Phi}_{\mathbf{z}}$ row rank deficient three, the following algorithm can be used.

Algorithm 5.4.1 *At a bifurcation point* $\mathbf{q}^*$, *with orthonormal basis* $\{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3\}$ *of* $ker({\boldsymbol{\Phi}_{\mathbf{z}}}^T(\mathbf{q}^*))$ *:*

1. *If* $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\xi}_3)$ *is row rank deficient, choose new* $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2$, *and* $\boldsymbol{\xi}_3$ *such that* $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\xi}_3)$ *is full row rank. If* $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\xi})$ *is row rank deficient for every* $\boldsymbol{\xi} \in ker(\boldsymbol{\Phi}_{\mathbf{z}}^T(\mathbf{q}^*))$, *then go to* **8**.

2. *Define a grid on* $\theta \in [0, 2\pi)$, *and for each* θ *do the following:*

3. *Find an orthonormal basis* $\{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_k\}$ *of* $ker(\boldsymbol{\Phi}_{\mathbf{z}})$ *by solving the system* $\boldsymbol{\Phi}_{\mathbf{z}}\boldsymbol{\alpha}_i = \mathbf{0}$, *and form the matrix* $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_k]$.

4. *Evaluate*

$$\mathbf{A} = \boldsymbol{\alpha}^T \left[(\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi}_1)_{\mathbf{y}} (-\boldsymbol{\Phi}_{\mathbf{y}}^{-1} \boldsymbol{\Phi}_{\mathbf{v}}) + (\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi}_1)_{\mathbf{v}} \right] \tag{5.63}$$

$$\mathbf{B} = \boldsymbol{\alpha}^T \left[(\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi}_2)_{\mathbf{y}} (-\boldsymbol{\Phi}_{\mathbf{y}}^{-1} \boldsymbol{\Phi}_{\mathbf{v}}) + (\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi}_2)_{\mathbf{v}} \right] \tag{5.64}$$

$$\mathbf{C} = \boldsymbol{\alpha}^T \left[(\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi}_3)_{\mathbf{y}} (-\boldsymbol{\Phi}_{\mathbf{y}}^{-1} \boldsymbol{\Phi}_{\mathbf{v}}) + (\boldsymbol{\Phi}_{\mathbf{z}}^T \boldsymbol{\xi}_3)_{\mathbf{v}} \right] \tag{5.65}$$

$$\mathbf{D}(\theta) = [\cos(\theta)\mathbf{A} + \sin(\theta)\mathbf{B}] \tag{5.66}$$

5. *Evaluate the right inverse of* $\mathbf{C}$*,*

$$\mathbf{C}^+ = \mathbf{C}^T(\mathbf{C}\mathbf{C})^{-1} \tag{5.67}$$

6. *Evaluate* $-\mathbf{D}(\theta)\mathbf{C}^+$*.*

7. *Evaluate the row rank deficiency of* $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\xi}(\phi, \theta))$*.*

7a. *If* $\mathbf{G}_{\mathbf{x}}(\mathbf{q}^*, \boldsymbol{\xi}(\phi, \theta))$ *has row rank deficiency 2, then,*

7a.1. *Determine the eigenvalues and eigenvectors of the matrix* $-\mathbf{D}(\theta)\mathbf{C}^+$*.*

7a.2. *For each eigenvalue* $\cot(\phi)$ *and corresponding real eigenvectors* $\boldsymbol{\eta}^{(1)}$*,* $\boldsymbol{\eta}^{(2)}, \ldots,$ *compute* $\boldsymbol{\zeta}_2 = \mathbf{C}^+\boldsymbol{\eta}^{(i)}$*,* $\boldsymbol{\zeta}_1 = (-\boldsymbol{\Phi}_{\mathbf{y}}^{-1}\boldsymbol{\Phi}_{\mathbf{v}})\boldsymbol{\zeta}_2$*, and* $\boldsymbol{\zeta}_3$ *as defined in Theorem 5.3.1. Also compute* $\boldsymbol{\zeta}^* := \left[\mathbf{0}, \mathbf{0}, (\frac{d\boldsymbol{\xi}}{d\phi})^T\right]^T$ *and* $\boldsymbol{\zeta}_* := \left[\mathbf{0}, \mathbf{0}, \boldsymbol{\zeta}_{*3}^{\;T}\right]^T$*, as defined in equations (5.51) and (5.56), and normalize* $\boldsymbol{\zeta}_*$*.*

7a.3. *Normalize the vector* $\left[\boldsymbol{\zeta}_1^{\;T}, \boldsymbol{\zeta}_2^{\;T}, \boldsymbol{\zeta}_3^{\;T}\right]^T$ *and denote by* $\boldsymbol{\zeta}$ *the unit vector obtained.*

7a.4. *Write the tangent to a continuation curve at* $\mathbf{q}^*$ *as*

$$\boldsymbol{\tau} = a\boldsymbol{\zeta} + b\boldsymbol{\zeta}^* + c\boldsymbol{\zeta}_* \tag{5.68}$$

and determine the coefficients a, b, and c, using the second-order necessary condition given in equation (5.50) and the condition $\|\boldsymbol{\tau}\| = 1$*.*

7b. *If* $\mathbf{G}_\mathbf{x}(\mathbf{q}^*, \boldsymbol{\xi}(\phi, \theta))$ *has row rank deficiency greater than or equal to 3, then,*

7b.1. *Determine the eigenvalues and eigenvectors of the matrix* $-\mathbf{D}(\theta)\mathbf{C}^+$. *When obtaining a multiple eigenvalue, determine corresponding linearly independent eigenvectors* $\boldsymbol{\eta}^1, \boldsymbol{\eta}^2, ..., \boldsymbol{\eta}^s$.

7b.2. *For each* $\boldsymbol{\eta}^i$, *compute the corresponding* $\boldsymbol{\zeta}^i := \left[{\boldsymbol{\zeta}^i_1}^T, {\boldsymbol{\zeta}^i_2}^T, {\boldsymbol{\zeta}^i_3}^T\right]^T$ *as in Theorem 5.3.1 and normalize it. Also, for each eigenvalue, compute the corresponding* $\boldsymbol{\zeta}^* := \left[\mathbf{0}, \mathbf{0}, (\frac{d\boldsymbol{\xi}}{d\phi})^T\right]^T$ *and* $\boldsymbol{\zeta}_* := \left[\mathbf{0}, \mathbf{0}, \boldsymbol{\zeta}_{*3}^T\right]^T$ *as defined in equations (5.51) and (5.56) and normalize* $\boldsymbol{\zeta}_*$.

7b.3. *Write the tangent to a continuation curve at* $\mathbf{q}^*$ *as*

$$\boldsymbol{\tau} = a_1\boldsymbol{\zeta}^1 + a_2\boldsymbol{\zeta}^2 + ... + a_s\boldsymbol{\zeta}^s + b\boldsymbol{\zeta}^* + c\boldsymbol{\zeta}_* \tag{5.69}$$

and determine the coefficients $a_1, a_2, ..., a_s, b,$ *and* c, *using the second-order necessary condition given in equation (5.50) and the normalization condition* $\|\boldsymbol{\tau}\| = 1$.

8. *Define a grid on* $(\phi, \theta) \in [0, \pi/2] \times [0, 2\pi)$, *and for each* (ϕ, θ) *do the following:*

8a. *Determine a basis of the null space of the matrix* $\mathbf{G}_\mathbf{x}(\mathbf{q}^*, \boldsymbol{\xi}(\phi, \theta))$.

8b. *Write the tangent to a continuation curve at* $\mathbf{q}^*$ *as a linear combination of vectors in this basis.*

8c. *Determine the coefficients in this expression of the tangent vector, using the second-order necessary condition and a normalization condition.*

In practice, to continue to follow the boundary it is required to keep track of the angles between the previous tangent at the point $\mathbf{q}^*$ and the tangents computed using Algorithm 5.4.1, and select the exterior tangent. Also, the interior region covered by continuation curves starting at $\mathbf{q}^*$ can be determined in this fashion.

CHAPTER 6
VARIABLE FIDELITY DIFFERENTIAL-ALGEBRAIC EQUATION MODEL CORELATION

6.1 Introduction

It is often necessary to model mechanical systems involving vehicles and equipment that are controlled by a human operator, at a level of abstraction suitable for real-time simulation, and subsequently model the same system at a higher level of fidelity for design purposes. Both models will generally be described by differential-algebraic equations of dynamics, with human input and feedback control. To be more specific, consider the case of real-time driver in the loop simulation of a vehicle over a winding road, in which the driver actions are recorded, along with vehicle position and orientation relative to the road surface on which the driver guides the vehicle. Such a simulation provides designers with realistic duty cycle information that is needed for component and subsystem analysis and design ([24]). The model used for driver-in-the-loop real-time simulation, however, is not at a level of fidelity suitable for accurate prediction of loads on components for purposes of design. It is thus necessary to resort to simulation of the same vehicle and duty cycle with a high fidelity model that yields accurate loads and stresses for design calculation.

The foregoing situation leads to a fundamental difficulty in high-fidelity simulation of numerous classes of human operated systems. If the operator inputs used to control a system, as predicted by a lower fidelity model suitable for real-time simulation, are input to the high fidelity model, and simulation is carried out, differences in the two models will generally result in a trajectory that does not follow the one generated in the operator-in-the-loop simulation. It is thus necessary to impose modified operator inputs that will lead to a trajectory that closely approximates the results of operator-in-the-loop real-time simulation. The fact that the resulting operator inputs are somewhat different from those actually employed by the specific individual that operated the equipment in the simulator is not of concern, since different operators would in fact behave somewhat differently. It is desirable, however, that the deviation in operator behaviour be as small as possible.

The problem addressed here includes diverse applications with human control, such as tele-robotics, construction equipment, manufacturing equipment, and ground vehicles. The vehicle application is singled out in this thesis as the focus for development for several reasons. First, it is pervasive in both life and work of everyone in the society. Second, the influence of human behaviour and judgement is dominant and impossible to model. Third, virtual proving grounds are emerging ([24]), that link a driver-in-the-loop simulator with design analysis codes to permit rapid-test iterations.

The variable fidelity model corelation addressed in this chapter is required to realize the potential for a revolutionary new virtual proving ground-based design-test technology.

6.2 Analytical Formulation

Two model formulations are considered, both differential-algebraic equations that represent low and high fidelity models of the same vehicle system. In each model, there are systems of differential equations of motion and algebraic equations of constraint, that are coupled using Lagrange multipliers. In addition, each model contains differential equations that represent dynamics of steering, braking, and propulsion subsystems. While the physical state variables and control states in each of the models are different, it is presumed that each model depends on the same driver input variables and predicts vehicle motion. The low and high fidelity differential-algebraic equation models, with their associated controllers, are summarized, and an error function that is to be minimized in selecting modified driver inputs for the high fidelity model is defined.

6.2.1 The Low-Fidelity Model

For simulation in real-time, we employ a system of second order differential-algebraic equations of the form ([23])

$$\mathbf{L}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{\Psi_q}^T\boldsymbol{\gamma} = \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u}^*, \mathbf{z}) \tag{6.1a}$$

$$\mathbf{\Psi}(\mathbf{q}, \mathbf{d}^*) = \mathbf{0} \tag{6.1b}$$

with appropriate initial conditions. Here, the generalized coordinate vector $\mathbf{q}$ defines the position and orientation of all bodies in the system, $\boldsymbol{\gamma}$ is the vector of Lagrange multipliers, $\mathbf{u}^*$ is the vector of driver inputs that influences generalized forces, such as, for example, the accelerator pedal position, $\mathbf{z}$ is the vector of control states, and $\mathbf{d}^*$ is the vector of driver inputs that influences kinematic constraints, such as, for example, control of the steering rack position. The control equations for this model are given in first order form

$$\dot{\mathbf{z}} = \mathbf{g}(\mathbf{z}, \mathbf{u}^*, \mathbf{q}) \tag{6.2}$$

Under modest and realistic hypotheses, and with kinematically admissible initial conditions on $\mathbf{q}$ and $\dot{\mathbf{q}}$, and initial conditions on $\mathbf{z}$, the differential-algebraic system defined by 6.1 and 6.2 has a unique solution for $\mathbf{q}$, $\boldsymbol{\gamma}$, and $\mathbf{z}$ as functions of time for given input histories $\mathbf{u}^*(t)$ and $\mathbf{d}^*(t)$. The physical position and orientation of the vehicle on the roadway are thus determined as functions of the solution $\mathbf{q}(t)$.

6.2.2 The High-Fidelity Model

For convenience in subsequent analysis, the high-fidelity model is specified in Cartesian coordinate form as a system of first order differential-algebraic equations,

$$\mathbf{M}\dot{\mathbf{s}} + \mathbf{\Phi}_{\mathbf{r}}^T \boldsymbol{\lambda} - \mathbf{F}^A(\mathbf{r}, \mathbf{s}, \mathbf{p}, \mathbf{v}, \mathbf{u}, \mathbf{w}) = \mathbf{0} \tag{6.3a}$$

$$\mathbf{J}(\mathbf{p})\dot{\mathbf{v}} + \mathbf{\Phi}_{\mathbf{p}}^T \boldsymbol{\lambda} - \mathbf{N}^A(\mathbf{r}, \mathbf{s}, \mathbf{p}, \mathbf{v}, \mathbf{u}, \mathbf{w}) = \mathbf{0} \tag{6.3b}$$

$$\dot{\mathbf{r}} - \mathbf{s} = \mathbf{0} \tag{6.3c}$$

$$\dot{\mathbf{p}} - \mathbf{v} = \mathbf{0} \tag{6.3d}$$

$$\mathbf{\Phi}(\mathbf{r}, \mathbf{p}, \mathbf{d}) = \mathbf{0} \tag{6.3e}$$

where $\mathbf{r}$ is the vector of position coordinates of all bodies in the system, $\mathbf{s}$ is its time derivative, $\mathbf{p}$ is the vector of Euler parameters for all the bodies in the system (see [23]), $\mathbf{v}$ is its time derivative, $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers, $\mathbf{u}$ is the vector of driver inputs that influence the control system and generalized forces, and $\mathbf{d}$ is the vector of driver inputs that influence kinematic constraints. The applied forces and torques $\mathbf{F}^A$ and $\mathbf{N}^A$ depend on driver and control input, $\mathbf{M}$ is a constant diagonal mass matrix, $\mathbf{J}(\mathbf{p}) = 4\mathbf{G}^T\mathbf{J}'\mathbf{G}$, where $\mathbf{J}'$ is a block diagonal matrix containing all body inertia tensors, $\mathbf{G}$ depends only on Euler parameters (see [23]), and the constraints in equation 6.3e include Euler parameter normalization constraints. The control equations for this model, in first order form, are

$$\dot{\mathbf{w}} - \mathbf{f}(\mathbf{w}, \mathbf{u}, \mathbf{r}, \mathbf{p}) = \mathbf{0} \tag{6.4}$$

where $\mathbf{w}$ is the vector of control states.

As in the low-fidelity model described by 6.1 and 6.2, the differential-algebraic system given in 6.3 and 6.4 has a unique solution for $\mathbf{r}$, $\mathbf{p}$, $\mathbf{s}$, $\mathbf{v}$, $\boldsymbol{\lambda}$ and $\mathbf{w}$, as functions of the driver inputs $\mathbf{u}$ and $\mathbf{d}$, once kinematically admissible initial conditions and initial conditions on controller state are specified. Once $\mathbf{r}$ and $\mathbf{p}$ are known, the position and orientation of the vehicle relative to the roadway, predicted by the high-fidelity model can be determined. As noted above, it is assumed that $\mathbf{u}$ and $\mathbf{u}^*$, as well as $\mathbf{d}$ and $\mathbf{d}^*$, have the same physical significance in both models.

6.2.3 The Error Function

In order to quantify the deviation between (1) trajectories predicted by the low and high fidelity models, (2) control states, and (3) driver inputs, an error measure is needed. For purposes of error minimization, a scalar measure of error is formulated as

$$e = \int_{t_0}^{t_1} [a(\mathbf{r}, \mathbf{p}; \mathbf{q}) + b(\mathbf{w}; \mathbf{z}) + c(\mathbf{u}, \mathbf{d}; \mathbf{u}^*, \mathbf{d}^*)]dt \tag{6.5}$$

where t_0 is the beginning of the simulation and t_1 is the end time. The first term in the integrand quantifies deviation in vehicle position and orientation on the roadway, the second term quantifies deviation in control states, and the third term quantifies deviation in driver inputs.

The objective in variable fidelity model correlation is to select $\mathbf{u}$ and $\mathbf{d}$, given $\mathbf{u}^*$ and $\mathbf{d}^*$ and the associated solution of equations (6.1) and (6.2) from the low-fidelity simulation, to minimize the error function of equation (6.5). In this formulation, the first order differential-algebraic equations and control equations for the high-fidelity model of equations (6.3) and (6.4) become constraints in the optimization process.

6.3 Sensitivity Analysis

In order to solve the error minimization problem summarized above, the differential (or sensitivity) of the error function in (6.5) with respect to $\mathbf{u}$ and $\mathbf{d}$ must be computed. The adjoint variable method of sensitivity analysis ([22], [8]) for differential-algebraic equations is used for this purpose. Adjoint identities are first introduced, in terms of undetermined adjoint variables. The variation of each of the identities is then calculated and the results are summed to create a comprehensive adjoint differential identity that is subsequently employed to reduce the differential the error function to a relation that is explicit in variations of $\mathbf{u}$ and $\mathbf{d}$. The resulting elimination process yields a system of differential-algebraic equations with terminal conditions at t_1, which must thus be integrated backward to t_0 to obtain adjoint variables that may be used to evaluate the explicit sensitivity of the error function due to variations in $\mathbf{u}$ and $\mathbf{d}$.

6.3.1 Adjoint Identities

Multiplying both sides of equation (6.3a) by $\boldsymbol{\mu}^T$ and integrating over the interval (t_0, t_1) yields

$$\int_{t_0}^{t_1} \boldsymbol{\mu}^T[\mathbf{M}\dot{\mathbf{s}} + \boldsymbol{\Phi}_{\mathbf{r}}^T\boldsymbol{\lambda} - \mathbf{F}^A]dt = 0 \tag{6.6}$$

Multiplying both sides of equation (6.3b) by $\boldsymbol{\nu}^T$ and integrating yields

$$\int_{t_0}^{t_1} \boldsymbol{\nu}^T[\mathbf{J}\dot{\mathbf{v}} + \boldsymbol{\Phi}_{\mathbf{p}}^T\boldsymbol{\lambda} - \mathbf{N}^A]dt = 0 \tag{6.7}$$

Multiplying both sides of equation (6.3c) by ξ^T and integrating yields

$$\int_{t_0}^{t_1} \xi^T[\dot{\mathbf{r}} - \mathbf{s}]dt = 0 \tag{6.8}$$

Multiplying both sides of equation (6.3d) by ζ^T and integrating yields

$$\int_{t_0}^{t_1} \zeta^T[\dot{\mathbf{p}} - \mathbf{v}]dt = 0 \tag{6.9}$$

Multiplying both sides of equation (6.3e) by $\boldsymbol{\eta}^T$ and integrating yields

$$\int_{t_0}^{t_1} \boldsymbol{\eta}^T \boldsymbol{\Phi} dt = 0 \tag{6.10}$$

Finally, multiplying both sides of equation (6.4) by $\boldsymbol{\beta}^T$ and integrating yields

$$\int_{t_0}^{t_1} \boldsymbol{\beta}^T[\dot{\mathbf{w}} - \mathbf{f}]dt = 0 \tag{6.11}$$

Equations (6.6) through (6.11) are required to hold for arbitrary values of the adjoint variables $\boldsymbol{\mu}, \boldsymbol{\nu}, \xi, \zeta, \boldsymbol{\eta}$, and $\boldsymbol{\beta}$. They provide identities that will be used to reduce the variation of the error function to an explicit differential form in variations of $\mathbf{u}$ and $\mathbf{d}$.

6.3.2 Variations of Adjoint Identities

Equations (6.6) through (6.11) hold for any values of the adjoint variables, provided that equations (6.3) and (6.4) are satisfied. The variation of each of these equations is first taken with respect to all variables that appear, except the adjoint variables. Integration by parts is then employed to create a system of identities involving variations in the state variables, but not their derivatives.

Taking the variation of equation (6.6) and integrating by parts yields

$$\int_{t_0}^{t_1} \{-\dot{\boldsymbol{\mu}}\mathbf{M}\delta\mathbf{s} + \boldsymbol{\mu}^T[\boldsymbol{\Phi}_{\mathbf{r}}^T\delta\boldsymbol{\lambda} + (\boldsymbol{\Phi}_{\mathbf{r}}^T\boldsymbol{\lambda})_{\mathbf{r}}\delta\mathbf{r} + (\boldsymbol{\Phi}_{\mathbf{r}}^T\boldsymbol{\lambda})_{\mathbf{p}}\delta\mathbf{p} + (\boldsymbol{\Phi}_{\mathbf{r}}^T\boldsymbol{\lambda})_{\mathbf{d}}\delta\mathbf{d} - \mathbf{F}_{\mathbf{r}}^A\delta\mathbf{r}$$

$$-\mathbf{F}_{\mathbf{s}}^A\delta\mathbf{s} - \mathbf{F}_{\mathbf{p}}^A\delta\mathbf{p} - \mathbf{F}_{\mathbf{v}}^A\delta\mathbf{v} - \mathbf{F}_{\mathbf{u}}^A\delta\mathbf{u} - \mathbf{F}_{\mathbf{w}}^A\delta\mathbf{w}]\}dt + \boldsymbol{\mu}^T\mathbf{M}\delta\mathbf{s}|_{t_0}^{t_1} = 0 \tag{6.12}$$

Taking the variation of equation (6.7) and integrating by parts yields

$$\int_{t_0}^{t_1}\{-\dot{\boldsymbol{\nu}}\mathbf{J}\delta\mathbf{v}+\boldsymbol{\nu}^T[-\dot{\mathbf{J}}\delta\mathbf{v}+(\mathbf{J}\dot{\mathbf{v}})_{\mathbf{p}}\delta\mathbf{p}+\boldsymbol{\Phi}_{\mathbf{p}}^T\delta\boldsymbol{\lambda}+(\boldsymbol{\Phi}_{\mathbf{p}}^T\boldsymbol{\lambda})_{\mathbf{r}}\delta\mathbf{r}+(\boldsymbol{\Phi}_{\mathbf{p}}^T\boldsymbol{\lambda})_{\mathbf{p}}\delta\mathbf{p}+(\boldsymbol{\Phi}_{\mathbf{p}}^T\boldsymbol{\lambda})_{\mathbf{d}}\delta\mathbf{d}$$

$$-\mathbf{N}_{\mathbf{r}}^A\delta\mathbf{r}-\mathbf{N}_{\mathbf{p}}^A\delta\mathbf{p}-\mathbf{N}_{\mathbf{v}}^A\delta\mathbf{v}-\mathbf{N}_{\mathbf{s}}^A\delta\mathbf{s}-\mathbf{N}_{\mathbf{u}}^A\delta\mathbf{u}-\mathbf{N}_{\mathbf{w}}^A\delta\mathbf{w}]\}dt+\boldsymbol{\nu}^T\mathbf{J}\delta\mathbf{v}|_{t_0}^{t_1}=0 \tag{6.13}$$

Taking the variation of equation (6.8) and integrating by parts yields

$$\int_{t_0}^{t_1}[-\dot{\xi}^T\delta\mathbf{r}-\xi^T\delta\mathbf{s}]dt+\xi^T\delta\mathbf{r}|_{t_0}^{t_1}=0 \tag{6.14}$$

Taking the variation of equation (6.9) and integrating by parts yields

$$\int_{t_0}^{t_1}[-\dot{\zeta}^T\delta\mathbf{p}-\zeta^T\delta\mathbf{v}]dt+\zeta^T\delta\mathbf{p}|_{t_0}^{t_1}=0 \tag{6.15}$$

Taking the variation of equation (6.10) and integrating by parts yields

$$\int_{t_0}^{t_1}[\boldsymbol{\Phi}_{\mathbf{r}}^T\delta\mathbf{r}-\boldsymbol{\Phi}_{\mathbf{p}}^T\delta\mathbf{p}+\boldsymbol{\Phi}_{\mathbf{d}}\delta\mathbf{p}]dt=0 \tag{6.16}$$

Finally, taking the variation of equation (6.11) and integrating by parts yields

$$\int_{t_0}^{t_1}\{-\dot{\boldsymbol{\beta}}^T\delta\mathbf{w}-\boldsymbol{\beta}^T[\mathbf{f}_{\mathbf{w}}\delta\mathbf{w}+\mathbf{f}_{\mathbf{u}}\delta\mathbf{u}+\mathbf{f}_{\mathbf{r}}\delta\mathbf{r}+\mathbf{f}_{\mathbf{p}}\delta\mathbf{p}]\}dt+\boldsymbol{\beta}^T\delta\mathbf{w}|_{t_0}^{t_1}=0 \tag{6.17}$$

6.3.3 Variation of the Error Function

Taking the variation of equation (6.5) yields

$$\delta e=\int_{t_0}^{t_1}[a_{\mathbf{r}}\delta\mathbf{r}+a_{\mathbf{p}}\delta\mathbf{p}+b_{\mathbf{w}}\delta\mathbf{w}+c_{\mathbf{u}}\delta\mathbf{u}+c_{\mathbf{d}}\delta\mathbf{d}]dt \tag{6.18}$$

Adding the variational identities of equations (6.6) through (6.11) and equating the resulting coefficients of the variations in $\mathbf{s}$, $\mathbf{v}$, $\mathbf{r}$, $\mathbf{p}$, $\boldsymbol{\lambda}$, and $\mathbf{w}$ with those in equation (6.18), both within the interval (t_0, t_1) and at t_1, respectively, yields the following adjoint equations inside the interval (t_0, t_1):

$$-\mathbf{M}\dot{\boldsymbol{\mu}} - \mathbf{F_s^{A}}^T\boldsymbol{\mu} - \mathbf{N_s^{A}}^T\boldsymbol{\nu} - \xi = \mathbf{0} \tag{6.19}$$

$$-\mathbf{J}\dot{\boldsymbol{\nu}} - \mathbf{F_v^{A}}^T\boldsymbol{\mu} - \dot{\mathbf{J}}\boldsymbol{\nu} - \mathbf{N_v^{A}}^T\boldsymbol{\nu} - \zeta = \mathbf{0} \tag{6.20}$$

$$-\dot{\xi} + (\boldsymbol{\Phi}_\mathbf{r}^T\boldsymbol{\lambda})_\mathbf{r}^T\boldsymbol{\mu} - \mathbf{F_r^{A}}^T\boldsymbol{\mu} + (\boldsymbol{\Phi}_\mathbf{p}^T\boldsymbol{\lambda})_\mathbf{r}^T\boldsymbol{\nu} - \mathbf{N_r^{A}}^T\boldsymbol{\nu} + \boldsymbol{\Phi}_\mathbf{r}^T\boldsymbol{\eta} - \mathbf{F}_\mathbf{r}^T\boldsymbol{\beta} = a_\mathbf{r}^T \tag{6.21}$$

$$-\dot{\zeta} + (\boldsymbol{\Phi}_\mathbf{r}^T\boldsymbol{\lambda})_\mathbf{p}^T\boldsymbol{\mu} - \mathbf{F_p^{A}}^T\boldsymbol{\mu} + (\boldsymbol{\Phi}_\mathbf{p}^T\boldsymbol{\lambda})_\mathbf{p}^T\boldsymbol{\nu} + (\mathbf{J}\dot{\mathbf{v}})_\mathbf{p}^T\boldsymbol{\nu} - \mathbf{N_p^{A}}^T\boldsymbol{\nu} + \boldsymbol{\Phi}_\mathbf{p}^T\boldsymbol{\eta} - \mathbf{f}_\mathbf{p}^T\boldsymbol{\beta} = a_\mathbf{p}^T \tag{6.22}$$

$$\boldsymbol{\Phi}_\mathbf{r}\boldsymbol{\mu} + \boldsymbol{\Phi}_\mathbf{p}\boldsymbol{\nu} = \mathbf{0} \tag{6.23}$$

$$-\dot{\boldsymbol{\beta}} - \mathbf{F_w^{A}}^T\boldsymbol{\mu} - \mathbf{N_w^{A}}^T\boldsymbol{\nu} - \mathbf{f}_\mathbf{w}^T\boldsymbol{\beta} = b_\mathbf{w}^T \tag{6.24}$$

and the following equations at t_1:

$$\mathbf{M}\boldsymbol{\mu}(t_1) = \mathbf{0} \tag{6.25}$$

$$\mathbf{J}\boldsymbol{\nu}(t_1) = \mathbf{0} \tag{6.26}$$

$$\xi(t_1) = 0 \tag{6.27}$$

$$\zeta(t_1) = 0 \tag{6.28}$$

$$\boldsymbol{\beta}(t_1) = \mathbf{0} \tag{6.29}$$

The remaining terms in the sum of equations (6.6) through (6.11) are the desired result,

$$\delta e = \int_{t_0}^{t_1} \{[c_{\mathbf{u}} + \boldsymbol{\mu}^T \mathbf{F}_{\mathbf{u}}^A + \boldsymbol{\nu}^T \mathbf{N}_{\mathbf{u}}^A + \boldsymbol{\beta}^T \mathbf{f}_{\mathbf{u}}]\delta \mathbf{u} + [c_{\mathbf{d}} - \boldsymbol{\mu}^T (\boldsymbol{\Phi}_{\mathbf{r}}^T \boldsymbol{\lambda})_{\mathbf{d}} - \boldsymbol{\nu}^T (\boldsymbol{\Phi}_{\mathbf{p}}^T \boldsymbol{\lambda})_{\mathbf{d}} - \boldsymbol{\eta}^T \boldsymbol{\Phi}_{\mathbf{d}}]\delta \mathbf{d}\} dt \tag{6.30}$$

If the terminal-value problem of equations (6.19) through (6.29) can be integrated from t_1 backward to t_0 for all the adjoint variables, then, equation (6.30) can be evaluated as the explicit differential of the error function with respect to $\mathbf{u}$ and $\mathbf{d}$. Note that equations (6.19) to (6.24) are a system of differential-algebraic equations in the adjoint variables, with terminal conditions at t_1 given by equations (6.25) through (6.29). It remains to show that a unique solution of this terminal-value problem in fact exists.

Differentiating equation (6.23) yields

$$\boldsymbol{\Phi}_{\mathbf{r}} \dot{\boldsymbol{\mu}} + \boldsymbol{\Phi}_{\mathbf{p}} \dot{\boldsymbol{\nu}} = -\dot{\boldsymbol{\Phi}}_{\mathbf{r}} \boldsymbol{\mu} - \dot{\boldsymbol{\Phi}}_{\mathbf{p}} \boldsymbol{\nu} \tag{6.31}$$

and differentiating equation (6.31) yields

$$\boldsymbol{\Phi}_{\mathbf{r}} \ddot{\boldsymbol{\mu}} + \boldsymbol{\Phi}_{\mathbf{p}} \ddot{\boldsymbol{\nu}} \equiv \boldsymbol{\alpha} \tag{6.32}$$

where the right side of equation (6.32) is quadratic in first order derivatives of adjoint variables.

Equations (6.19) and (6.20) can be differentiated and the derivatives of ξ and ζ from equations (6.21) and (6.22) can be substituted to obtain the following second order equations:

$$-\mathbf{M}\ddot{\boldsymbol{\mu}} - {\mathbf{F}_{\mathbf{s}}}^{A^T} \dot{\boldsymbol{\mu}} - \dot{\mathbf{F}}_{\mathbf{s}}^{A^T} \boldsymbol{\mu} - {\mathbf{N}_{\mathbf{s}}}^{A^T} \dot{\boldsymbol{\nu}} - \dot{\mathbf{N}}_{\mathbf{s}}^{A^T} \boldsymbol{\nu} = (\boldsymbol{\Phi}_{\mathbf{r}}^T \boldsymbol{\lambda})_{\mathbf{r}}^T \boldsymbol{\mu} - {\mathbf{F}_{\mathbf{r}}}^{A^T} \boldsymbol{\mu}$$

$$+ (\boldsymbol{\Phi}_{\mathbf{p}}^T \boldsymbol{\lambda})_{\mathbf{r}}^T \boldsymbol{\nu} - {\mathbf{N}_{\mathbf{r}}}^{A^T} \boldsymbol{\nu} + \boldsymbol{\Phi}_{\mathbf{r}}^T \boldsymbol{\eta} - \mathbf{f}_{\mathbf{r}}^T \boldsymbol{\beta} - a_{\mathbf{r}}^T \tag{6.33}$$

$$-\mathbf{J}\ddot{\boldsymbol{\nu}} - 2\dot{\mathbf{J}}\dot{\boldsymbol{\nu}} - \mathbf{F_v}^{A^T}\dot{\boldsymbol{\mu}} - \dot{\mathbf{F}}_{\mathbf{v}}^{A^T}\boldsymbol{\mu} - \ddot{\mathbf{J}}\boldsymbol{\nu} - \mathbf{N_v}^{A^T}\dot{\boldsymbol{\nu}} - \dot{\mathbf{N}}_{\mathbf{v}}^{A^T}\boldsymbol{\nu} = (\boldsymbol{\Phi}_{\mathbf{p}}^T\boldsymbol{\lambda})_{\mathbf{p}}^T\boldsymbol{\mu}$$

$$-\mathbf{F_p}^{A^T}\boldsymbol{\mu} + (\boldsymbol{\Phi}_{\mathbf{p}}^T\boldsymbol{\lambda})_{\mathbf{p}}^T\boldsymbol{\nu} - (\mathbf{J}\dot{\boldsymbol{\nu}})_{\mathbf{p}}^T - \mathbf{N_p}^{A^T}\boldsymbol{\nu} + \boldsymbol{\Phi}_{\mathbf{p}}^T\boldsymbol{\eta} - \mathbf{f}_{\mathbf{p}}^T\boldsymbol{\beta} - a_{\mathbf{p}}^T \tag{6.34}$$

In matrix form, equations (6.33), (6.34), and (6.32) can be written as

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} & \boldsymbol{\Phi}_{\mathbf{r}}^T \\ \mathbf{0} & \mathbf{J} & \boldsymbol{\Phi}_{\mathbf{p}}^T \\ \boldsymbol{\Phi}_{\mathbf{r}} & \boldsymbol{\Phi}_{\mathbf{p}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\mu}} \\ \ddot{\boldsymbol{\nu}} \\ \boldsymbol{\eta} \end{bmatrix} = \mathbf{RHS} \tag{6.35}$$

where **RHS** contains variables other that $\ddot{\boldsymbol{\mu}}$, $\ddot{\boldsymbol{\nu}}$, and $\mathbf{q}$. The coefficient matrix of equation (6.35) is the same as that for the second order high-fidelity differential-algebraic equation of motion. Thus, with appropriate terminal conditions, they have a unique solution ([23]).

To see that the terminal conditions of equations (6.25) through (6.29) uniquely determine $\boldsymbol{\mu}(t_1), \boldsymbol{\nu}(t_1), \dot{\boldsymbol{\mu}}(t_1), \dot{\boldsymbol{\nu}}(t_1)$, and $\boldsymbol{\beta}(t_1)$, note first that equation (6.23) at t_1 may be interpreted as specifying that $\boldsymbol{\mu}(t_1)$ and $\boldsymbol{\nu}(t_1)$ are kinematically admissible virtual velocities. Premultiplying equations (6.25) and (6.26) by $\frac{1}{2}\boldsymbol{\mu}^T(t_1)$ and $\frac{1}{2}\boldsymbol{\nu}^T(t_1)$, respectively, and adding shows that the associated kinetic energy is zero. Thus, $\boldsymbol{\mu}(t_1) = \boldsymbol{\nu}(t_1) = \mathbf{0}$ ([23]). With these results and equations (6.27) and (6.28), equations (6.19) and (6.20) at t_1 yield $\mathbf{M}\dot{\boldsymbol{\mu}}(t_1) = \mathbf{0}$ and $\mathbf{J}\dot{\boldsymbol{\nu}}(t_1) = \mathbf{0}$, and equation (6.31) at t_1 yields $\boldsymbol{\Phi}_{\mathbf{r}}\dot{\boldsymbol{\mu}}(t_1) + \boldsymbol{\Phi}_{\mathbf{p}}\dot{\boldsymbol{\nu}}(t_1) = \mathbf{0}$. This implies, by the same argument, that $\dot{\boldsymbol{\mu}}(t_1) = \dot{\boldsymbol{\nu}}(t_1) = \mathbf{0}$. The terminal conditions on $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ are thus

$$\boldsymbol{\mu}(t_1) = \mathbf{0} \tag{6.36a}$$

$$\boldsymbol{\nu}(t_1) = \mathbf{0} \tag{6.36b}$$

$$\dot{\boldsymbol{\mu}}(t_1) = \mathbf{0} \tag{6.36c}$$

$$\dot{\boldsymbol{\nu}}(t_1) = \mathbf{0} \tag{6.36d}$$

The terminal-value differential-algebraic equations (6.35), (6.23), (6.31) and (6.24), with zero terminal conditions (6.36) and (6.29), thus have a unique solution. This solution may be substituted into equation (6.30) to obtain the sensitivities (derivatives) of the error function e with respect to the driver inputs $\mathbf{u}$ and $\mathbf{d}$.

6.4 Numerical Implementation

The following algorithm can be used to find the inputs $\mathbf{u}$ and $\mathbf{d}$ for the high-fidelity model, that minimize the error functional of equation (6.5):

Algorithm 6.4.1

Step 1. *Solve the low-fidelity model described in equations (6.1) and (6.2) over the entire time interval* $[t_0, t_1]$ *and store the solution.*

Step 2. *Integrate the high-fidelity model of equations (6.3) and (6.4) from the initial time* t_0*, with initial estimates* $\mathbf{u} = \mathbf{u}^*$*, and* $\mathbf{d} = \mathbf{d}^*$ *for the inputs. At each time step, evaluate the pointwise difference between the solution of the high-fidelity model and the solution of the low-fidelity model. Stop the integration when the pointwise error between the two models exceeds some chosen tolerance, and denote by* t_f *the corresponding time value.*

Step 3. *Solve the differential-algebraic equations (6.35), (6.23), (6.31), and (6.24), with zero terminal conditions at* t_f*, from* t_f *to* t_0 *for the adjoint variables* $\boldsymbol{\mu}$*,* $\boldsymbol{\nu}$*, and* $\boldsymbol{\eta}$*. These are used in equation (6.30) with* t_f *in place of* t_1*, to obtain the gradient of the error function* e *of equation (6.5) with respect to the inputs* $\mathbf{u}$ *and* $\mathbf{d}$*, which is then supplied to the optimization routine*

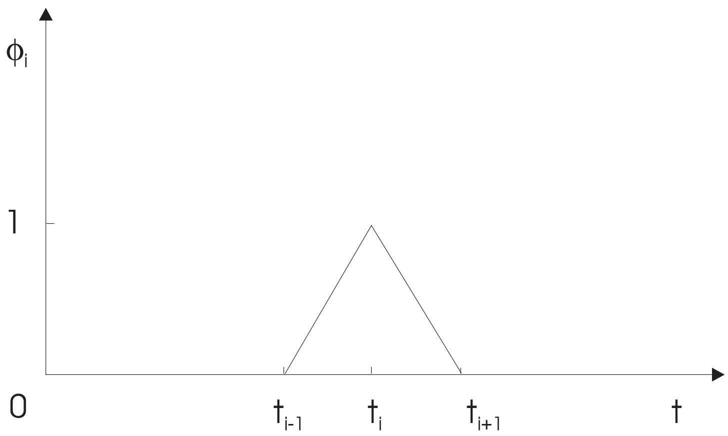


Figure 3: Piecewise linear basis function ϕ_i

Step 4. *Iteratively minimize the error function e, with respect to the inputs* $\mathbf{u}$ *and* $\mathbf{d}$*, over the interval* $[t_0, t_1]$*. If* $t_f < t_1$ *return to* **Step 4***.*

Remarks. (i) To reduce oscillation in the solution of the optimization problem, parametrize the inputs $\mathbf{u}$ and $\mathbf{d}$ before applying the (discrete) optimization routine. This way, the number of variables is reduced and the process is accelerated. The input can be parameterized as

$$T_h(t) = \sum_{i=1}^{N} a_i \phi_i(t) \tag{6.37}$$

where ϕ_i are piecewise linear basis functions, defined by

$$\phi_i(t) = \begin{cases} \frac{t-t_{i-1}}{t_i-t_{i-1}}, & if \quad t \in [t_{i-1}, t_i] \\ \frac{t_{i+1}-t}{t_{i+1}-t_i}, & if \quad t \in [t_i, t_{i+1}] \\ 0, & otherwise \end{cases} \tag{6.38}$$

the graph of a basis function ϕ_i is shown in Figure 3

(ii) Another way of solving the problem is to minimize the error e over disjoint intervals of the form $[T_0, t_1]$, where initially $T_0 = t_0$ and at the end of **Step 4**, if

$t_f < t_1$, let $T_0 = t_f$ and go to **Step 2**. This will make the optimization process faster, since the objective function e is now a function of fewer variables, but the solution obtained is suboptimal over the entire interval and may no longer be smooth. This solution can be smoothened by doing a final optimization of the error function e over the entire interval $[t_0, t_1]$, using the solution obtained by minimizing over disjoint intervals as the starting estimate for iteration. In numerical experiments performed, the solution obtained in this fashion was not as smooth as the one obtained by using the above algorithm. However, in some practical situations in which the time interval is long, this method may be a better approach to solve the problem than the more conservative algorithm given.

6.5 Numerical Examples

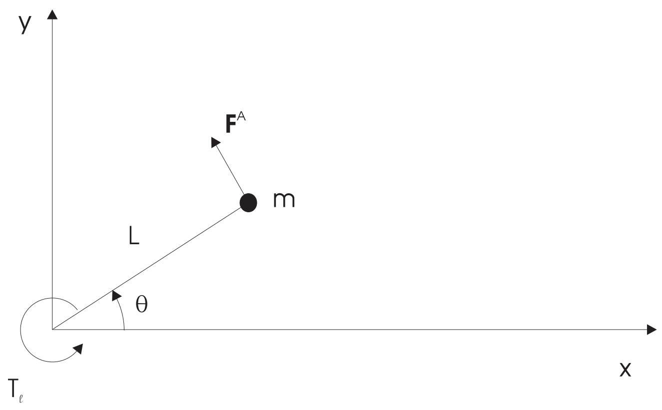
6.5.1 The Simple Pendulum

Consider as the low-fidelity model the pendulum shown in Figure 4, where $F^A = x\dot{y} + \dot{x}y$ is an applied force and the applied torque $\mathbf{u}^* = T_l$ is chosen as the input. The low-fidelity model is described by the second-order ordinary differential equation

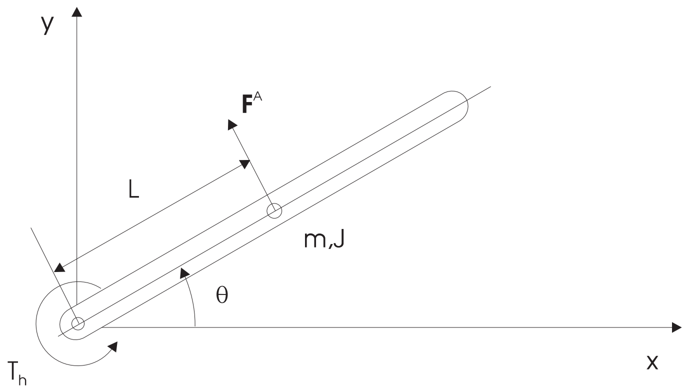
$$mL^2\ddot{\theta} = F^A L - mg\cos\theta + T_l - c_1\dot{\theta} \tag{6.39}$$

where T_l is a constant torque and c_1 is a damping constant. The high-fidelity model is also shown in Figure 4 and is described by the following system of differential-algebraic equations

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & J \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ L\sin\theta & -L\cos\theta \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} -F^A\sin\theta \\ -mg + F^A\cos\theta \\ T_h(t) - c_2\dot{\theta} \end{bmatrix} \tag{6.40}$$



Low-fidelity pendulum



High-fidelity pendulum

Figure 4: Low-fidelity and high-fidelity pendulum

$$\mathbf{\Phi} = \begin{bmatrix} x - L\cos\theta \\ y - L\sin\theta \end{bmatrix} = \mathbf{0} \tag{6.41}$$

where c_2 is a damping constant, and T_h will be determined such that the evolution in time of the two models is as close as possible, i.e. the error functional defined by

$$e = \int_{t_0}^{t_1} \{[(x_h(t) - x_l(t))^2 + (y_h(t) - y_l(t))^2] + [\alpha(T_h(t) - T_l)^2]\}dt \tag{6.42}$$

is minimized. Here the indices h and l correspond to the solution of the high-fidelity and low-fidelity models, respectively. The choice of the parameter α in the expression for the error can emphasize differences in position between the two models over differences in input, or it could keep the input T_h close to the input of the low-fidelity model T_l.

To define the right endpoint t_f of the optimization subinterval, the following pointwise error was used:

$$E(t) = (x_h(t) - x_l)^2 + (y_h(t) - y_l)^2 \tag{6.43}$$

The adjoint variables $\boldsymbol{\mu}$ and $\boldsymbol{\eta}$ are determined by solving the system of differential-algebraic equations

$$\begin{bmatrix} m & 0 & 0 & 1 & 0 \\ 0 & m & 0 & 0 & 1 \\ 0 & 0 & J & L\sin\theta & -L\cos\theta \\ 1 & 0 & L\sin\theta & 0 & 0 \\ 0 & 1 & -L\cos\theta & 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\mu}} \\ bfeta \end{bmatrix} = RHS$$

$$\mathbf{\Phi_q}\boldsymbol{\mu} = \begin{bmatrix} \mu_1 + L\sin\theta\mu_3 \\ \mu_2 - L\cos\theta\mu_3 \end{bmatrix} = \mathbf{0} \tag{6.44}$$

$$\mathbf{\Phi_q}\dot{\boldsymbol{\mu}} + \dot{\mathbf{\Phi}}_\mathbf{q}\boldsymbol{\mu} = \begin{bmatrix} \dot{\mu}_1 + L\sin\theta\dot{\mu}_3 + L\cos\theta\dot{\theta}\mu_3 \\ \dot{\mu}_2 - L\cos\theta\dot{\mu}_3 + L\sin\theta\dot{\theta}\mu_3 \end{bmatrix} = \mathbf{0}$$

where $\mathbf{q} = [x, y, \theta]$, with terminal conditions at t_f

$$\boldsymbol{\mu}(t_f) = \mathbf{0} \text{ and } \dot{\boldsymbol{\mu}}(t_f) = \mathbf{0} \tag{6.45}$$

The right side of equation (6.44) is

$$RHS(1) = 2(x - x_l) + y\sin\theta(\dot{\mu}_1 + \dot{\theta}\mu_2) - y\cos\theta(\dot{\mu}_2 - \dot{\theta}\mu_1)$$

$$RHS(2) = 2(y - y_l) + x\sin\theta(\dot{\mu}_1 + \dot{\theta}\mu_2) - x\cos\theta(\dot{\mu}_2 - \dot{\theta}\mu_1)$$

$$RHS(3) = c_2\dot{\mu}_3 - (\dot{x}y + x\dot{y})(\mu_1\cos\theta + \mu_2\sin\theta) - (L\lambda_1\cos\theta + L\lambda_2\sin\theta)\mu_3$$

$$RHS(4) = -2L\cos\theta\dot{\theta}\dot{\mu}_3 + L\sin\theta\dot{\theta}^2\mu_3 - L\cos\theta\ddot{\theta}\mu_3$$

$$RHS(5) = -2L\sin\theta\dot{\theta}\dot{\mu}_3 - L\cos\theta\dot{\theta}^2\mu_3 - L\sin\theta\ddot{\theta}\mu_3$$

Here, x, y, and θ stand for x_h, y_h, and θ_h. The adjoint variable $\boldsymbol{\mu}$ can then be used to compute the variation of the error functional,

$$\delta e = \int_{t_0}^{t_1} [2\alpha(T_h(t) - T_l) + \boldsymbol{\mu}^T\mathbf{F_u}]\delta\mathbf{u}dt = \int_{t_0}^{t_1} [2\alpha(T_h(t) - T_l(t)) + \mu_1]\sum_{i=1}^{N}\phi_i\delta a_i dt$$

$$= \sum_{i=1}^{N}\phi_i\int_{t_0}^{t_1}[2\alpha(T_h(t) - T_l(t)) + \mu_1]dt\delta a_i \tag{6.46}$$

Therefore, the gradient of the error function e with respect to the parameters a_i can be obtained by:

$$\frac{\partial e}{\partial a_i} = \sum_{i=1}^{N}\phi_i\int_{t_0}^{t_1}[2\alpha(T_h(t) - T_l(t)) + \mu_1]dt \tag{6.47}$$

The algorithm 6.4.1 was employed to find the input torque T_h that will keep the high-fidelity model close to the low-fidelity model. For optimization, the BFGS method was used on each interval $[t_0, t_f]$. Initially, the constant α in the definition

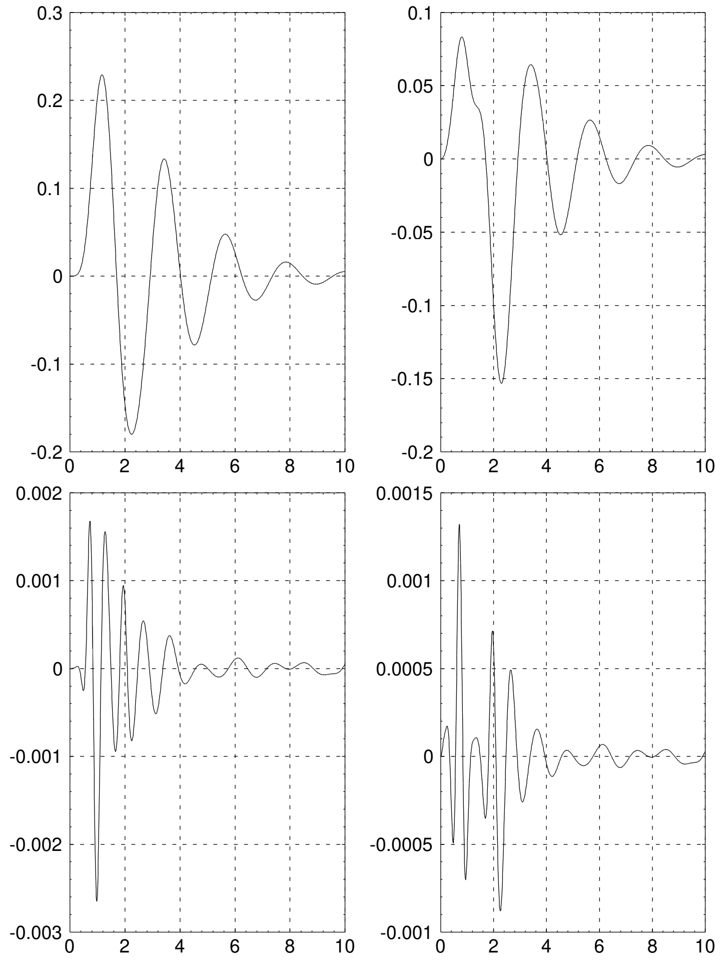


Figure 5: Pendulum. Deviations of the high-fidelity model from the low-fidelity model for $\alpha = 0, t_1 = 10s$: $x_h - x_l$ and $y_h - y_l$ before (up), and after (down) optimization

of the error function e was set to zero, to emphasize the difference-in-positions term, and then, α was increased to put more weight on the difference-in-inputs term in equation (6.42).

Figure 5 shows the difference between the states of the two models before and after optimization. In this case, $\alpha = 0$, which assured a very small deviation in positions between the high-fidelity and the low-fidelity model after optimization. The high-fidelity input T_h given by the algorithm is shown in Figure 6.

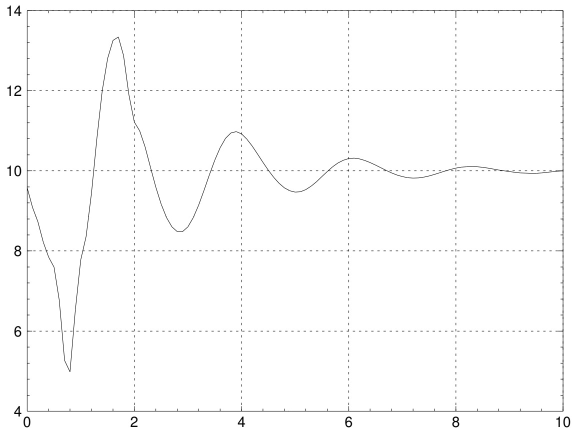


Figure 6: Pendulum. Optimized input T_h, for $\alpha = 0$, and $T_l = 10$

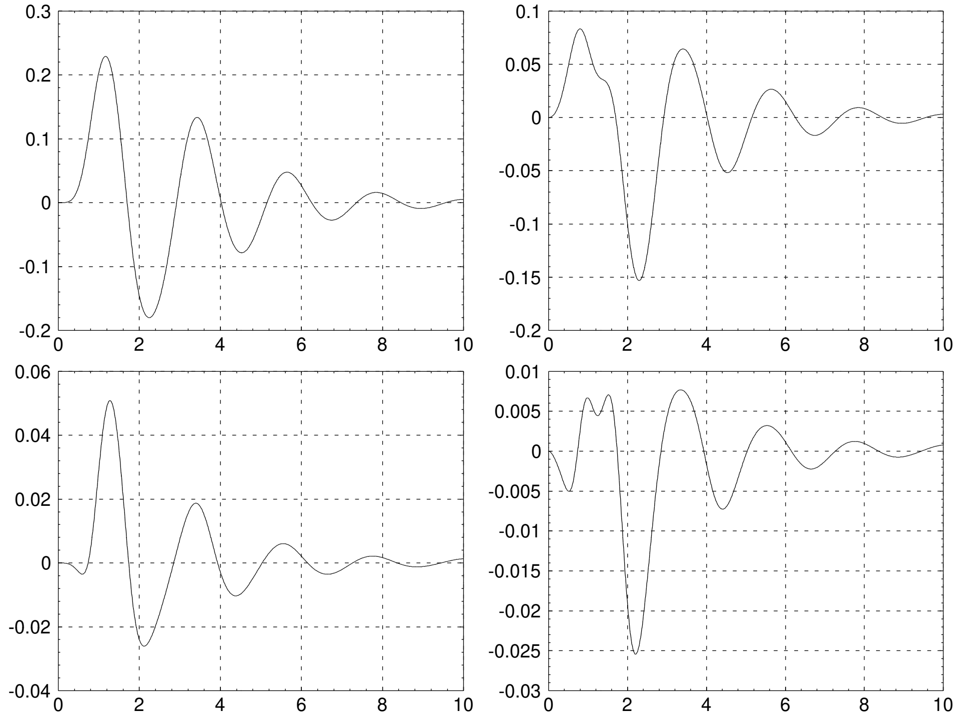


Figure 7: Pendulum. Deviations of the high-fidelity model from the low-fidelity model for $\alpha = 0.001, t_1 = 10s$: $x_h - x_l$ and $y_h - y_l$ before (up), and after (down) optimization

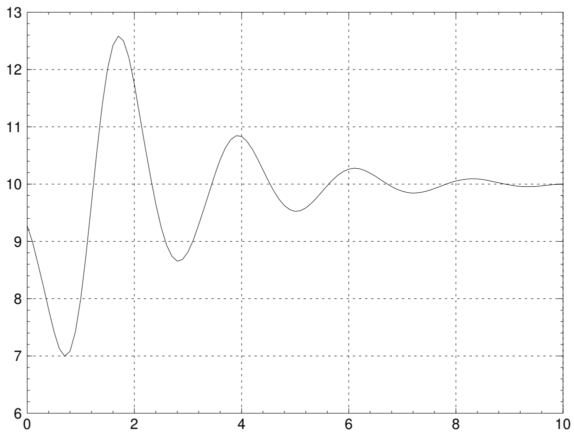


Figure 8: Pendulum. Optimized input T_h, for $\alpha = 0.001$, and $T_l = 10$

When the constant α was increased to 0.001, more weight is put on the difference-in-input term in equation (6.42), and as it can be seen in Figures 7 and 8, the optimized input T_h for the high-fidelity model is closer to the input of the low-fidelity model, while the error between the states of the two models increases.

The same experiments were performed using instead of the algorithm 6.4.1, optimization on adjacent intervals $[T_0, t_f]$, as in the second remark in section 6.4, where initially $T_0 = t_0$, and after determining the new t_f, $T_0 = t_{fold}$. After reaching the end of the time interval, a second stage optimization is performed, now over the entire interval $[t_0, t_1]$, with the solution obtained in the first stage as input. Since the objective function is a function of fewer variables a_i's, the optimization performs faster and, as shown in Figures 9-12, the states of the models are reasonably close together after optimization, with the error between states just slightly bigger than above, when the algorithm 6.4.1 was used. The problem with this procedure is that the optimized input T_h is no longer a smooth function of time. The second stage optimization is successful in removing some of the spikes in T_h, but the final solution is not differentiable, and if compared to the solution given by the algorithm, it is

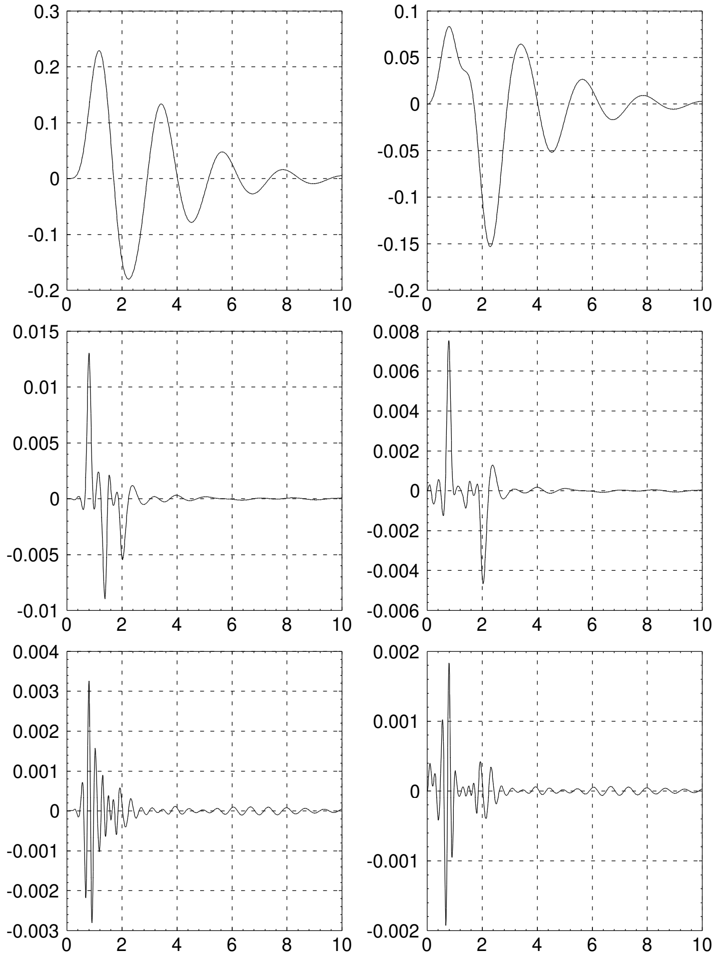


Figure 9: Pendulum. Deviations of the high-fidelity model from the low-fidelity model for $\alpha = 0, t_1 = 10s$ using the 2-stage optimization: $x_h - x_l$ and $y_h - y_l$ before (up), after the first stage (middle), and after the second stage (down) of optimization

further apart from the input of the low-fidelity model.

Initially α was set to zero, thus more weight is put on the difference between states terms in the objective function e. As Figure 9 shows, after the two-stage optimization process, the error in positions between the two models is made small, while the input necessary to achieve this is not differentiable as seen in Figure 10. Figures 11 and 12 show the results obtained with $\alpha = 0.001$. While the states of

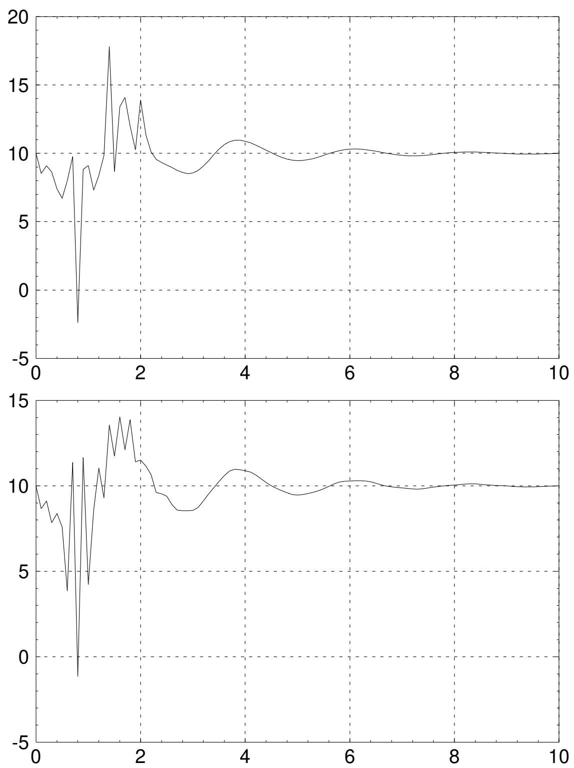


Figure 10: Pendulum. Optimized input T_h, after the first stage (up), and after the second stage (down) of optimization, for $\alpha = 0$, and $T_l = 10$

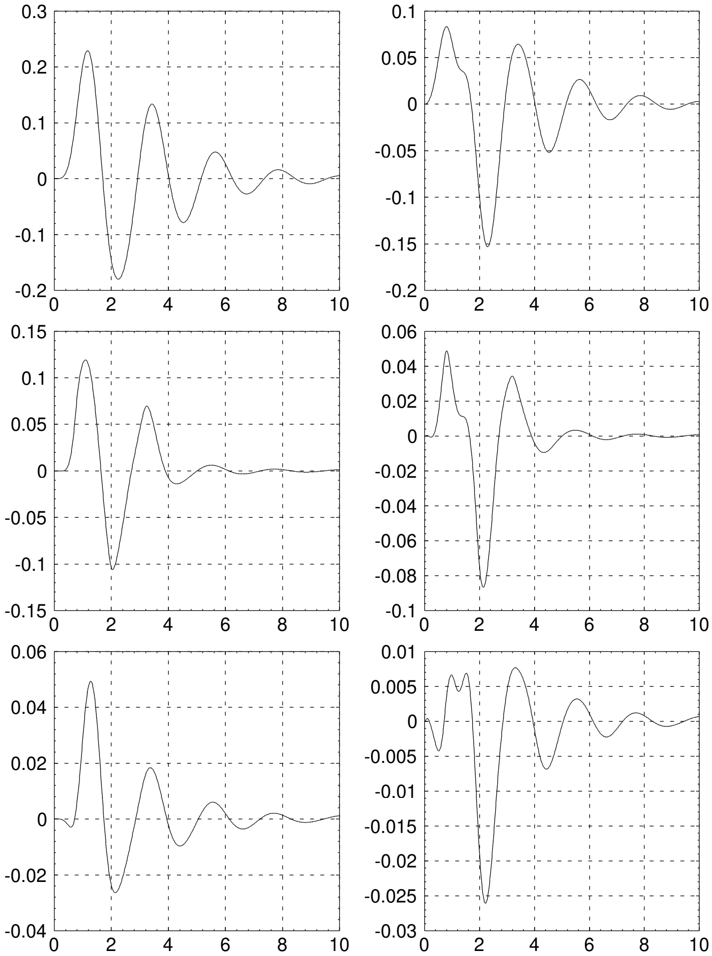


Figure 11: Pendulum. Deviations of the high-fidelity model from the low-fidelity model for $\alpha = 0.001, t_1 = 10s$ using the 2-stage optimization: $x_h - x_l$ and $y_h - y_l$ before (up), after the first stage (middle), and after the second stage (down) of optimization

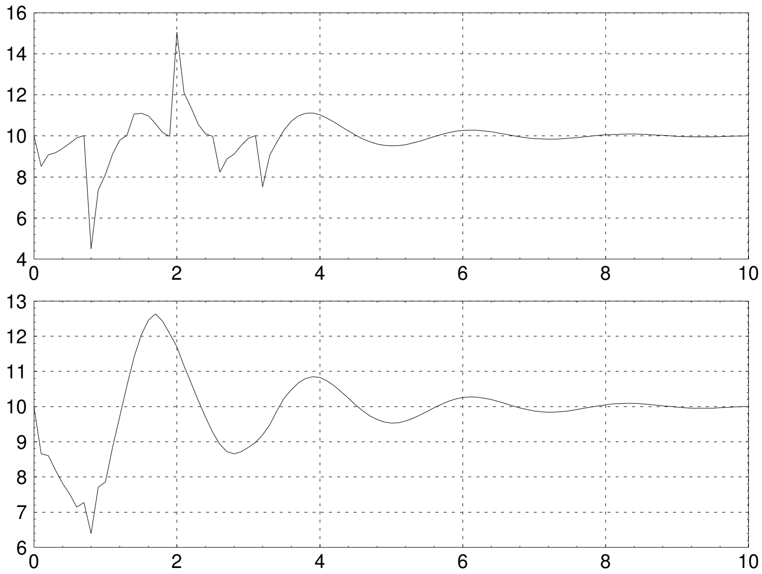
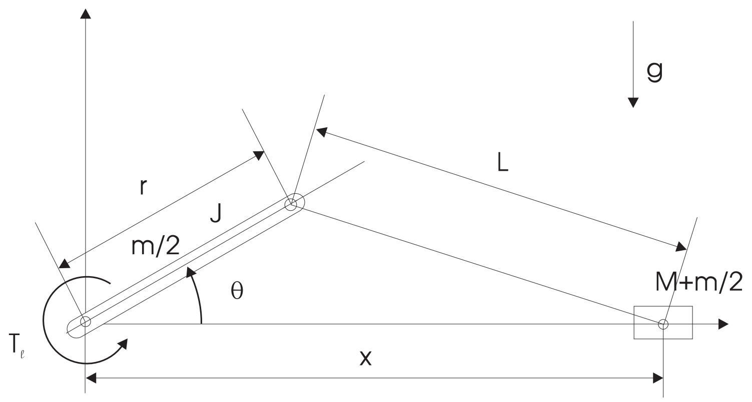
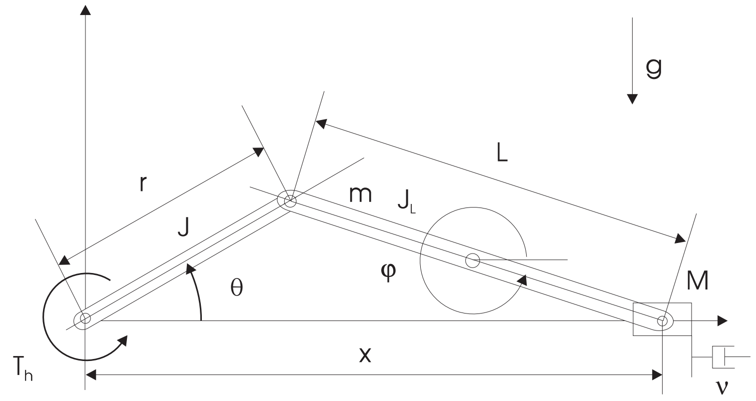


Figure 12: Pendulum. Optimized input T_h, after the first stage (up), and after the second stage (down) of optimization for $\alpha = 0.001$, and $T_l = 10$

the two models are further apart than before, the optimized input for the high-fidelity model, T_h, is closer to the input of the low-fidelity model, but it is still not differentiable.



Low-fidelity model for the slider-crank



High-fidelity model for the slider-crank

Figure 13: Low-fidelity and high-fidelity slider-crank

6.5.2 The Slider-Crank

The low-fidelity model for the slider-crank mechanism is shown in Figure 13. It models the connecting rod as a massless constraint and lumps the actual mass of the rod at its ends. Its equation of motion are

$$(J + \tfrac{mr^2}{2})\ddot{\theta} + 2rx\sin\theta\lambda = T_l - \tfrac{1}{2}mgr\cos\theta \tag{6.48a}$$

$$(M + \tfrac{m}{2})\ddot{x} + (2x - 2r\cos\theta)\lambda = 0 \tag{6.48b}$$

$$x^2 - 2rx\cos\theta + r^2 - L^2 = 0 \tag{6.49}$$

where the applied torque T_l is considered to be the input $\mathbf{u}^*$, and was set to be constant.

For the high-fidelity model, consider also a slider-crank mechanism, but with friction and a connecting rod body accounted for, as shown in Figure 13. The high-fidelity model is described by the following equations:

$$\begin{aligned}(J + \frac{mr^2}{2})\ddot{\theta} + \frac{1}{2}mrL\cos(\theta-\varphi)\ddot{\varphi} - r\sin\theta\lambda_1 + r\cos\theta\lambda_2 &= T_h \\ -mgr\cos\theta - \frac{1}{2}mrL\sin(\theta-\varphi)\dot{\varphi}^2 & \\ M\ddot{x} - \lambda_1 = F_f = -\nu\dot{x} & \\ \frac{1}{2}mrL\cos(\theta-\varphi) + (J + \frac{mL^2}{4})\dot{\varphi} - L\sin\varphi\lambda_1 + L\cos\varphi\lambda_2 &= -\frac{1}{2}mgL\cos\varphi \\ +\frac{1}{2}mrL\sin(\theta-\varphi)\dot{\varphi}^2 &\end{aligned} \tag{6.50}$$

with kinematic constraints

$$\mathbf{\Phi}(\theta, x, \phi) = \begin{bmatrix} r\cos\theta + L\cos\varphi - x \\ r\sin\theta + L\sin\varphi \end{bmatrix} = \mathbf{0} \tag{6.51}$$

The input for the high-fidelity model is the applied torque $T_h = \mathbf{u}$, which is determined using the optimization algorithm 6.4.1. The goal of the optimization process is to find an input for the high-fidelity model that yields an evolution of this model as close as possible to that of the low-fidelity model. The error function e used for this model, which is the cost function for the optimization problem, is

$$e = \int_{t_0}^{t_1} \{[(\theta_h(t) - \theta_l(t))^2 + (x_h(t) - x_l(t))^2] + [\alpha(T_h(t) - T_l)^2]\}dt \tag{6.52}$$

where indices h and l correspond to the high-fidelity and low-fidelity models, respectively, and t_0 and t_f are the endpoints of the interval. With the parametrization of the input T_h as in (6.37), optimization reduces to selection of the parameters a_i at each control point $t_i,\ i = 1, 2, ..., N$. The pointwise error used in **Step 2** of the numerical algorithm 6.4.1 to determine the right endpoint of the optimization intervals $[t_0, t_f]$ is defined by

$$E(t) = (\theta_h(t) - \theta_l(t))^2 + (x_h(t) - x_l(t))^2 \tag{6.53}$$

The parameter α in equation (6.52) can be chosen to emphasize differences in position over differences in input, which will keep the two models close, or α can be chosen to keep inputs of the two models as close as possible, which is also desirable in practical situations.

Performing sensitivity calculations, the adjoint variables $\boldsymbol{\mu}$ and $\boldsymbol{\eta}$ are obtained by solving the following differential-algebraic equation (analogous to equations (6.35), (6.26), (6.31)) on the interval $[t_0, t_f]$:

$$\begin{gathered}
\begin{bmatrix} \mathbf{M} & \boldsymbol{\Phi}_{\mathbf{q}}^T \\ \boldsymbol{\Phi}_{\mathbf{q}} & 0 \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\mu}} \\ \boldsymbol{\eta} \end{bmatrix} = RHS \\
\boldsymbol{\Phi}_{\mathbf{q}}\boldsymbol{\mu} = \begin{bmatrix} -r\sin\theta\mu_1 - \mu_2 - L\sin\varphi\mu_3 \\ r\cos\theta\mu_1 + L\cos\varphi\mu_3 \end{bmatrix} = \mathbf{0} \\
\boldsymbol{\Phi}_{\mathbf{q}}\dot{\boldsymbol{\mu}} + \dot{\boldsymbol{\Phi}}_{\mathbf{q}}\boldsymbol{\mu} = \begin{bmatrix} -r\sin\theta\dot{\mu}_1 - r\cos\theta\dot{\theta}\mu_1 - \dot{\mu}_2 - L\sin\varphi\dot{\mu}_3 - L\cos\varphi\dot{\varphi}\mu_3 \\ r\cos\theta\dot{\mu}_1 - r\sin\theta\dot{\theta}\mu_1 + L\cos\varphi\mu_3 - L\sin\varphi\dot{\varphi}\mu_3 \end{bmatrix} = \mathbf{0}
\end{gathered} \tag{6.54}$$

where $\mathbf{q} = [\theta, x, \varphi]^T$ and

$$\mathbf{M} = \begin{bmatrix} J + mr^2 & 0 & \frac{1}{2}mrL\cos(\theta-\varphi) \\ 0 & M & 0 \\ \frac{1}{2}mrL\cos(\theta-\varphi) & 0 & J_1 + \frac{1}{4}mL^2 \end{bmatrix}$$

$$\mathbf{\Phi} = \begin{bmatrix} r\cos\theta + L\cos\varphi - x \\ r\sin\theta + L\sin\varphi \end{bmatrix} \tag{6.55}$$

and

$$RHS(1) = 2(\theta-\theta_1) + \frac{1}{2}mrL\sin(\theta-\varphi)\ddot{\varphi}(\mu_1-\mu_3) - \frac{1}{2}mrL\cos(\theta-\varphi)\dot{\varphi}^2(\mu_1-\mu_3)$$
$$-mrL\sin(\theta-\varphi)\dot{\varphi}\dot{\mu}_3 + (r\cos\theta\lambda_1 + r\sin\theta\lambda_2)\mu_1 + mgr\sin\theta\mu_1$$
$$RHS(2) = 2(x-x_l) + \nu\dot{\mu}_2$$

$$RHS(3) = \frac{1}{2}mrL\sin(\theta-\varphi)\ddot{\theta}(\mu_1-\mu_3) + \frac{1}{2}mrL\cos(\theta-\varphi)\dot{\theta}^2(\mu_1-\mu_3) \tag{6.56}$$
$$+mrL\sin(\theta-\varphi)\dot{\theta}\mu_1 + (L\cos\varphi\lambda_1 + L\sin\varphi\lambda_2)\mu_3 + \frac{1}{2}mgL\sin\varphi\mu_3$$
$$RHS(4) = 2r\cos\theta\dot{\theta}\dot{\mu}_1 + 2L\cos\varphi\dot{\varphi}\dot{\mu}_3 - (-r\cos\theta\ddot{\theta} + r\sin\theta\dot{\theta}^2)\mu_1 - (-L\cos\varphi\ddot{\varphi}$$
$$+L\sin\varphi\dot{\varphi}^2)\mu_3$$
$$RHS(5) = 2r\sin\theta\dot{\theta}\dot{\mu}_1 + 2L\sin\varphi\dot{\varphi}\dot{\mu}_3 - (-r\sin\theta\ddot{\theta} + r\cos\theta\dot{\theta}^2)\mu_1 - (-L\sin\varphi\ddot{\varphi}$$
$$-L\cos\varphi\dot{\varphi}^2)\mu_3$$

The terminal conditions at t_f for the above differential- algebraic equations are

$$\boldsymbol{\mu}(t_f) = \mathbf{0} \tag{6.57a}$$

$$\dot{\boldsymbol{\mu}}(t_f) = \mathbf{0} \tag{6.57b}$$

The variation of the error in this case is

$$\delta e = \int_{t_0}^{t_f} [2\alpha(T_h(t) - T_l(t)) + \mu_1]\delta\mathbf{u}dt = \int_{t_0}^{t_f} [2\alpha(T_h(t) - T_l(t)) + \mu_1] \sum_{i=1}^{N} \phi_i \delta a_i dt$$

$$= \sum_{i=1}^{N} \phi_i \int_{t_0}^{t_f} [2\alpha(T_h(t) - T_l(t)) + \mu_1] dt \delta a_i \tag{6.58}$$

Using the solution of the differential-algebraic equation (6.54), the gradient of the error function with respect to the input parameters a_i is computed as

$$\frac{\partial e}{\partial a_i} = \sum_{i=1}^{N} \phi_i \int_{t_0}^{t_f} [2\alpha(T_h(t) - T_l(t)) + \mu_1] dt \tag{6.59}$$

It is then supplied to the optimization routine. The BFGS algorithm was used for optimization on each interval $[t_0, t_f]$.

Two different experiments were performed using the algorithm 6.4.1. In the first, the damping coefficient ν in the high-fidelity model was set to zero. In this case, differences between the low-fidelity and the high-fidelity models are due to different mass distributions among the bodies of the mechanism. For the second experiment, the damping coefficient was chosen to be $\nu = 3$. Therefore, the high-fidelity model is dissipating energy, and the input needed to mimic the behaviour of the low-fidelity model is expected to differ from the initial estimate more than in the first case.

A rather delicate issue is the choice of the coefficient α in equation (6.52). This coefficient determines how much weight is put on the deviation of the high-fidelity input from the low-fidelity input. If $\alpha = 0$, the focus is only on the states of the two models. As numerical results show, a good reproduction of the evolution of the low-fidelity model is obtained from the high-fidelity model. However, a penalty is paid in terms of high variations in the input for the high-fidelity model. Too large a value for α causes the high-fidelity input to be close to the value of the low-fidelity input, but allows large variations of the states of the high-fidelity model from the desired values. A reasonable choice for α is one that assures comparable magnitudes of the two parts of the right side of equation (6.52). Finding the value of α a-priori is not possible. Therefore, experiments were carried out with different values of α, and the deviation of the states of the high-fidelity model was computed. For each of these experiments, results are presented for $\alpha = 0$ and for larger values of α, for which the deviation in states is within acceptable limits. Note that the latter value for α depends only on the high-fidelity model used.

Experiment 1:

With the damping coefficient ν in the high-fidelity model set to zero, and for various values of α, the results are shown in Figures 14, 15, 16, and 17. As seen in Figures 14 and 15, when the constant α in the definition of the error function is set to zero, the algorithm keeps the state of the high-fidelity model close to the state of the low-fidelity model, while the input, T_h, needed, exhibits a large deviation from the input $T_l = 10$ of the low-fidelity model.

If greater weight is put on the difference-in-input term in equation (6.52), then the optimized input T_h for the high-fidelity model is closer to the input of the low-fidelity model, while the deviation in position increases (see Figures 16 and 17).

Experiment 2:

In this case, the damping coefficient ν in the high-fidelity model is set to $\nu = 3$. As a consequence, the high-fidelity model deviates more from the low-fidelity model (see Figure 18). Thus, the input torque T_h needed to keep the models close is larger (see Figure 19). The algorithm still performs well in keeping the states of the two models close when $\alpha = 0$ (see Figure 18). As seen in Figures 20) and 21), an increase in the coefficient α brings the optimized input for the high-fidelity model closer to the input of the low-fidelity model, while the deviation in position increases.

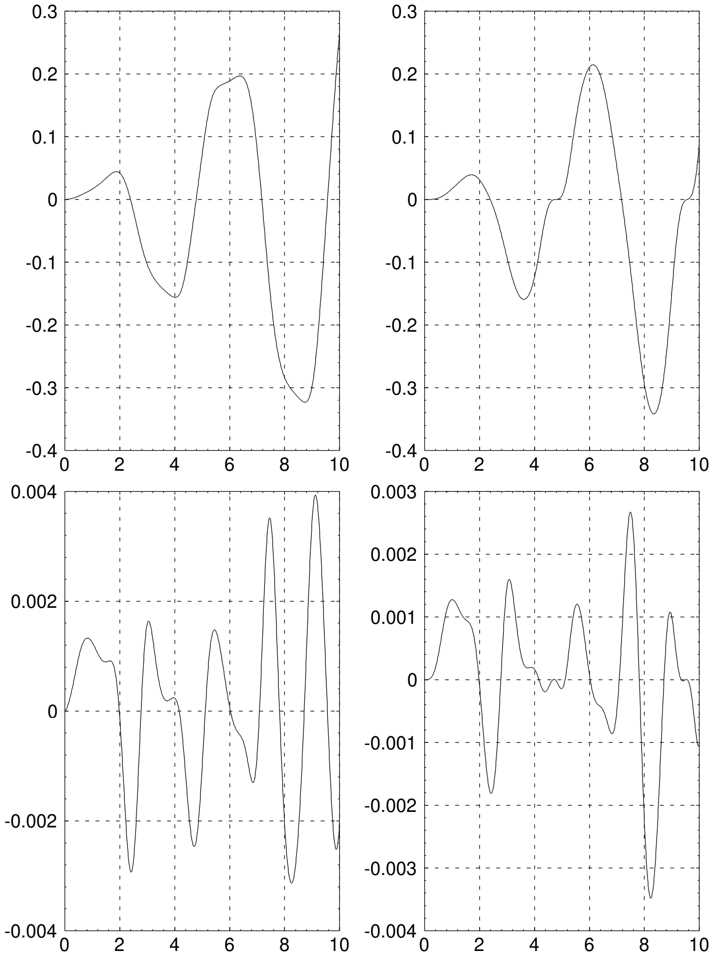


Figure 14: Slider-Crank. Deviations of the high-fidelity solution from the low-fidelity solution for $\nu = 0, \alpha = 0$, and $t_1 = 10s$: $\theta_h - \theta_l$ and $x_h - x_l$ before (up), and after (down) optimization

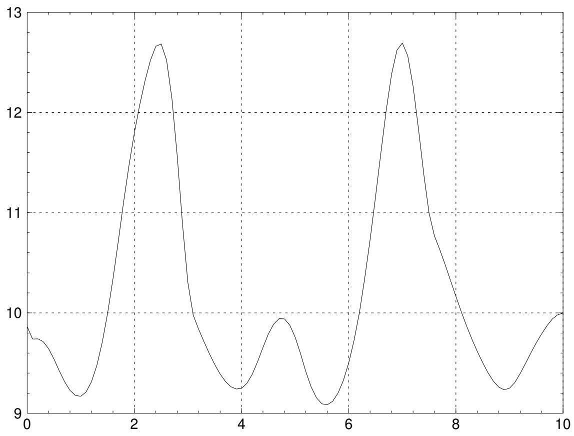


Figure 15: Slider-Crank. Optimized input T_h for the high-fidelity model, for $\nu = 0, \alpha = 0$, and $T_l = 10$

The same experiments were performed using instead of the algorithm proposed 6.4.1, optimization on adjacent intervals $[T_0, t_f]$, as in the second remark in section 6.4. At the beginning, $T_0 = t_0$, and after determining the new t_f, T_0 is defined to be equal to the previous t_f, $T_0 = t_{fold}$. Because the objective function is a function of fewer variables, the optimization process is faster, but the solution losses differentiability. A second optimization is performed over the entire time interval $[t_0, t_1]$, using the solution obtained in the first stage as input.

As before, the damping coefficient ν in the high-fidelity model was initially set to zero (see Figures 22 to 25). Here, the first row of plots corresponds to $T_h = T_l$. The next row of plots is obtained with the high-fidelity input given by the first round of optimization, over separate intervals. The plots of the last row are obtained with the high-fidelity input given by the second stage optimization, over the entire time interval.

As expected, the optimized input is not smooth. The spikes correspond to endpoints of the intervals on which optimization is performed on the first stage.

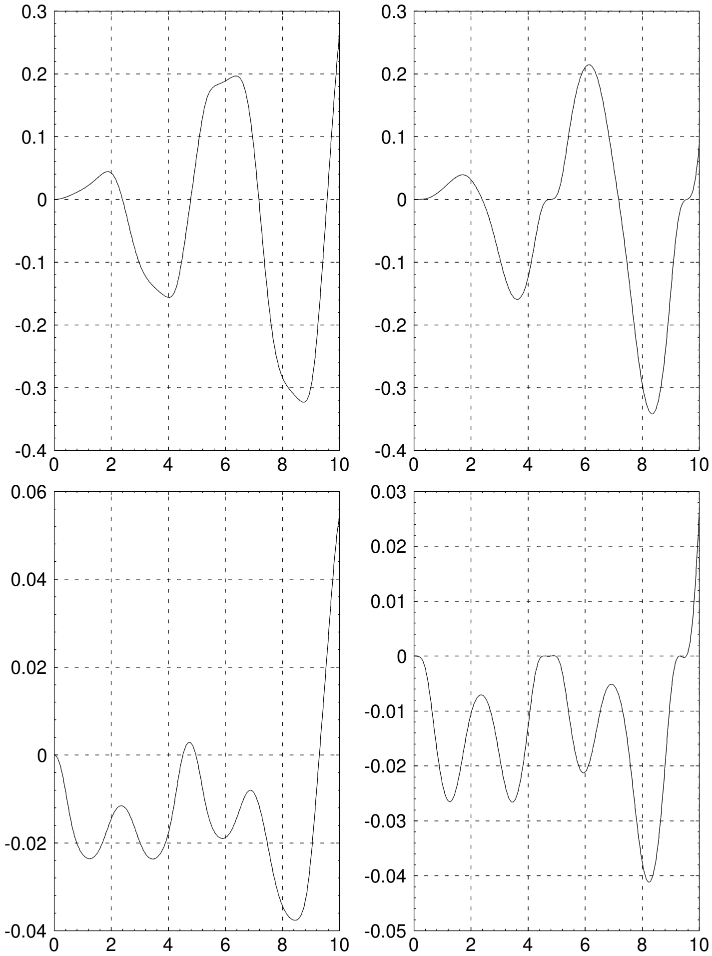


Figure 16: Slider-Crank. Deviations of the high-fidelity solution from the low-fidelity solution for $\nu = 0, \alpha = 0.005$, and $t_1 = 10s$: $\theta_h - \theta_l$ and $x_h - x_l$ before (up), and after (down) optimization

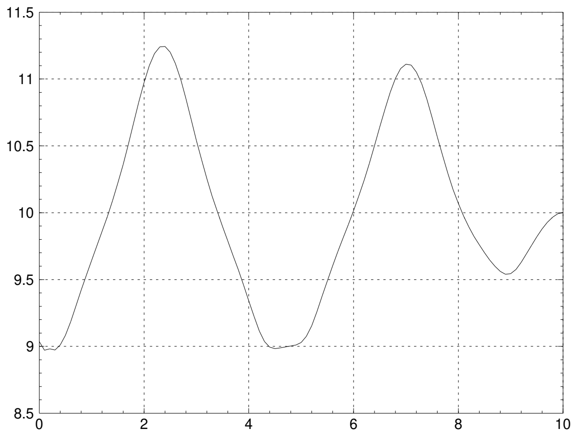


Figure 17: Slider-Crank. Optimized input T_h for the high-fidelity model, for $\nu = 0, \alpha = 0.005$, and $T_l = 10$

They are due to the parametrization of T_h in (6.37), as $T_h = \sum_{i=1}^{N} a_i \phi_i(t)$. Indeed, the optimization procedure here is designed not to allow changes in the high-fidelity input at the first point of a given interval. Changing the a_i corresponding to the first point of an interval would affect the solution of the high-fidelity model on the previous interval. On the other hand, the values of the input at the beginning of an interval have the largest influence on the solution on that interval. This is why the optimization procedure yields large values for the a_i corresponding to the second point of each interval, which is the first value considered for modification.

For $\alpha = 0.005$, results were shown in Figures 24 and 25. Note that, since more intervals were selected in the first stage of the optimization, spikes in the solution were smaller and the second round of optimization almost eliminated them.

A large damping coefficient $\nu = 3$ in the high-fidelity model yields results presented in Figures 26 to 29. If no weight is put on the deviation of the high-fidelity input from the low-fidelity input, i.e. if $\alpha = 0$, the high-fidelity model can be caused to behave close to the low-fidelity model, as illustrated in Figure 26. As expected,

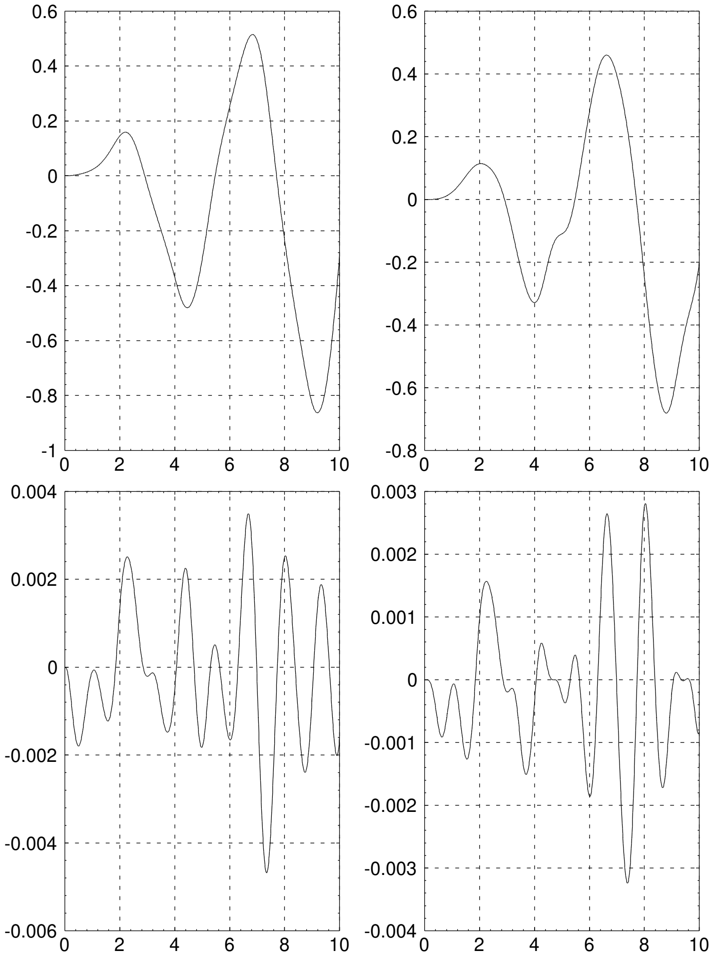


Figure 18: Slider-Crank. Deviations of the high-fidelity solution from the low-fidelity solution for $\nu = 3, \alpha = 0$, and $t_1 = 10s$: $\theta_h - \theta_l$ and $x_h - x_l$ before (up), and after (down) optimization

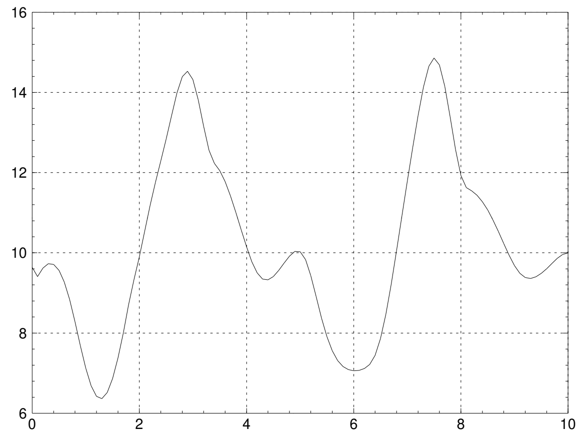


Figure 19: Slider-Crank. Optimized input T_h for the high-fidelity model, for $\nu = 3, \alpha = 0$, and $T_l = 10$

the required high-fidelity input varies in a large range, as seen in Figure 27. With $\alpha = 0.001$ in equation (6.37), the high-fidelity input can be maintained in a narrower range (see Figure 28), with the penalty of allowing larger deviations of the high-fidelity solution from the low-fidelity solution (see Figure 29).

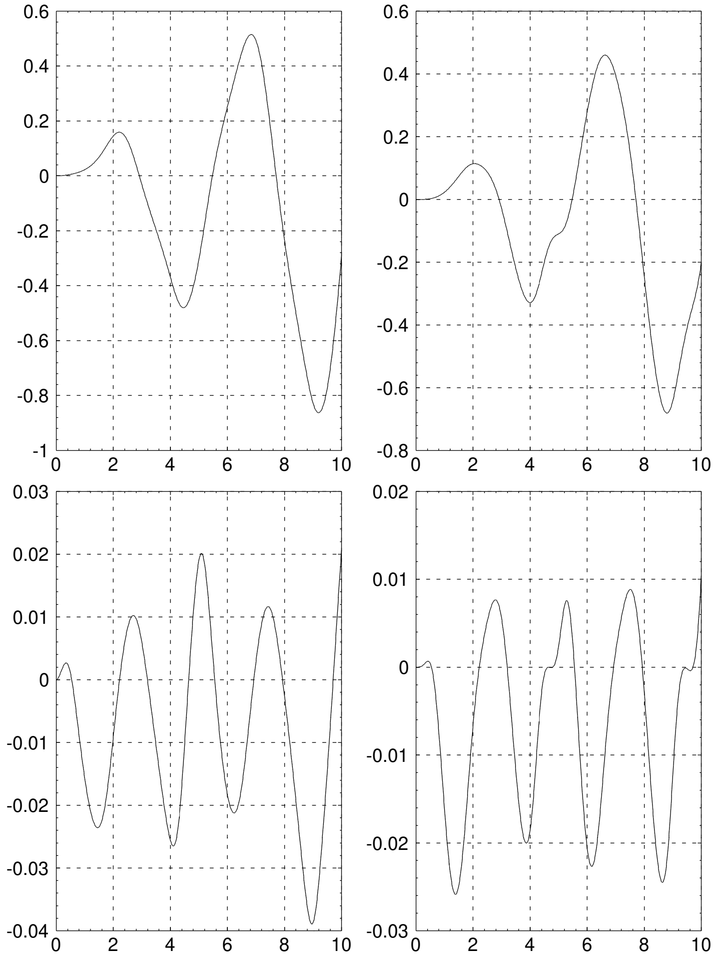


Figure 20: Slider-Crank. Deviations of the high-fidelity solution from the low-fidelity solution for $\nu = 3, \alpha = 0.001$, and $t_1 = 10s$: $\theta_h - \theta_l$ and $x_h - x_l$ before (up), and after (down) optimization

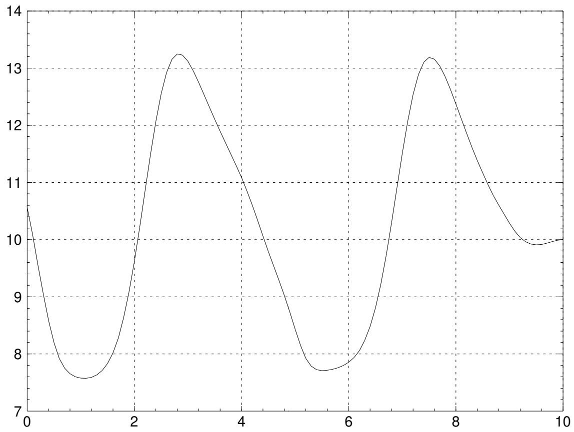


Figure 21: Slider-Crank. Optimized input T_h for the high-fidelity model, for $\nu = 3, \alpha = 0.001$, and $T_l = 10$

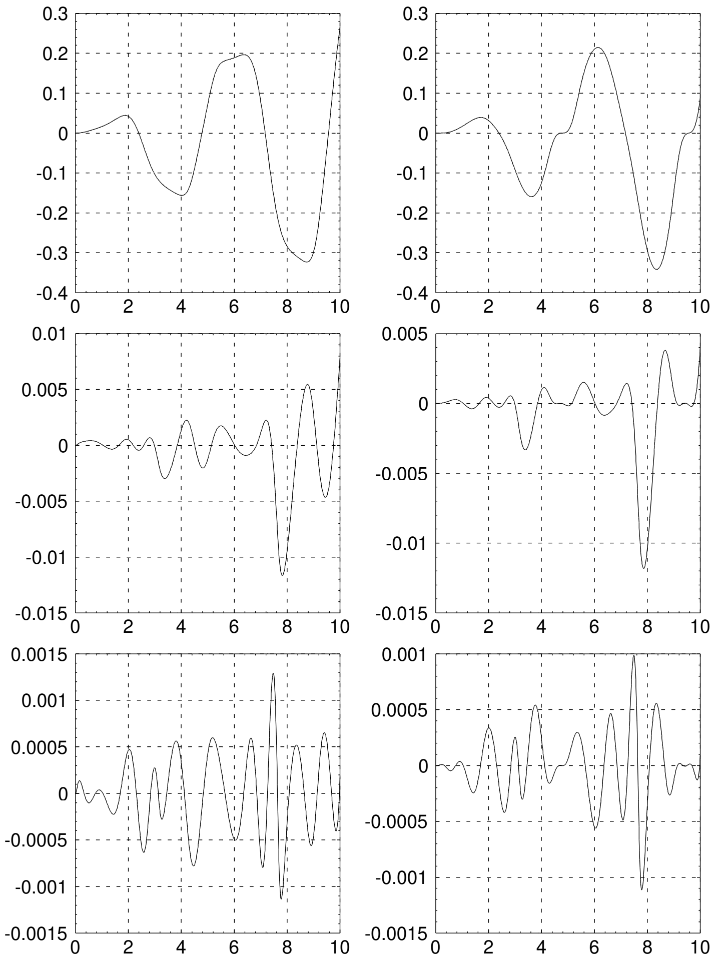


Figure 22: Slider-Crank. Deviations of the high-fidelity solution from the low-fidelity solution using the 2-stage optimization, for $\nu = 0, \alpha = 0$, and $t_1 = 10s$: $\theta_h - \theta_l$ and $x_h - x_l$ before optimization (up), after the first stage (middle), and after the second stage (down) of optimization

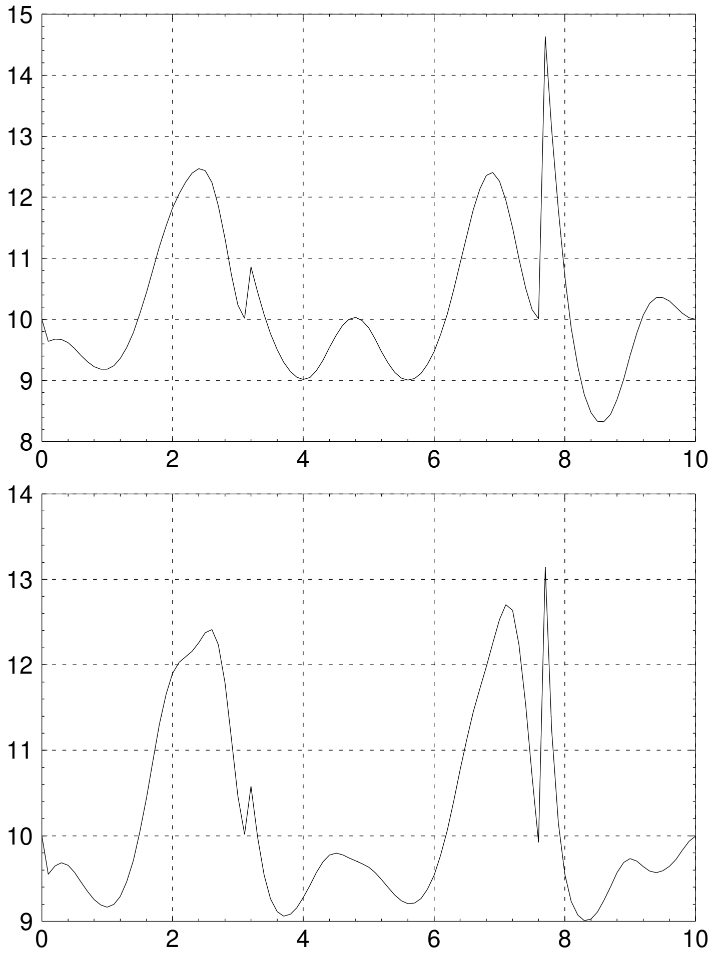


Figure 23: Slider-Crank. First stage (up) and second stage (down) optimized input T_h for the high-fidelity model, for $\nu = 0, \alpha = 0$, and $T_l = 10$

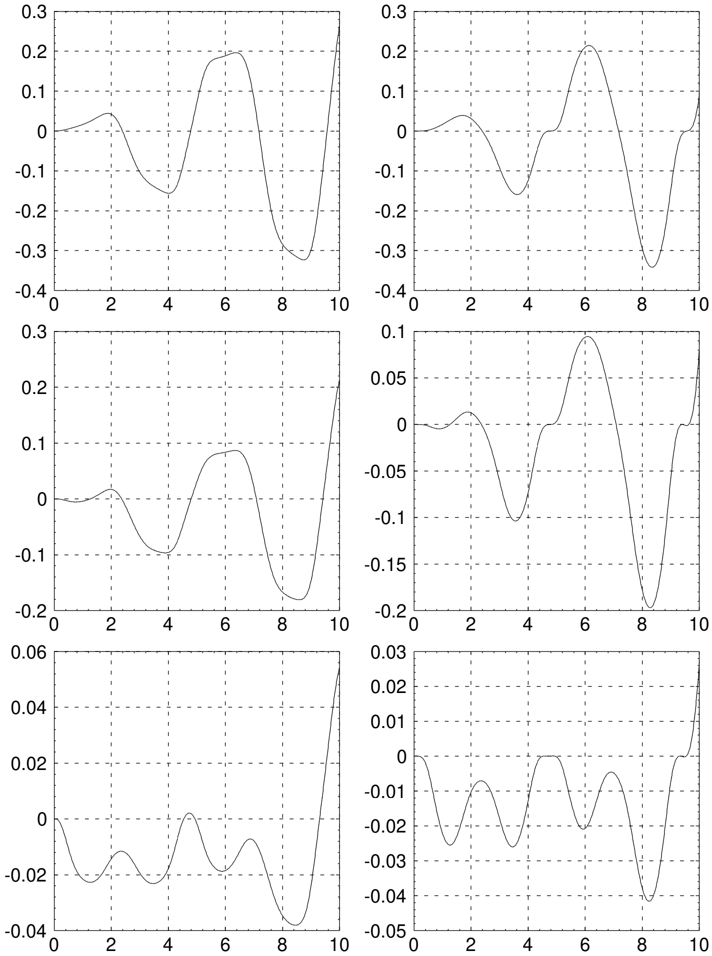


Figure 24: Slider-Crank. Deviations of the high-fidelity solution from the low-fidelity solution using the 2-stage optimization, for $\nu = 0, \alpha = 0.005$, and $t_1 = 10s$: $\theta_h - \theta_l$ and $x_h - x_l$ before optimization (up), after the first stage (middle), and after the second stage (down) of optimization

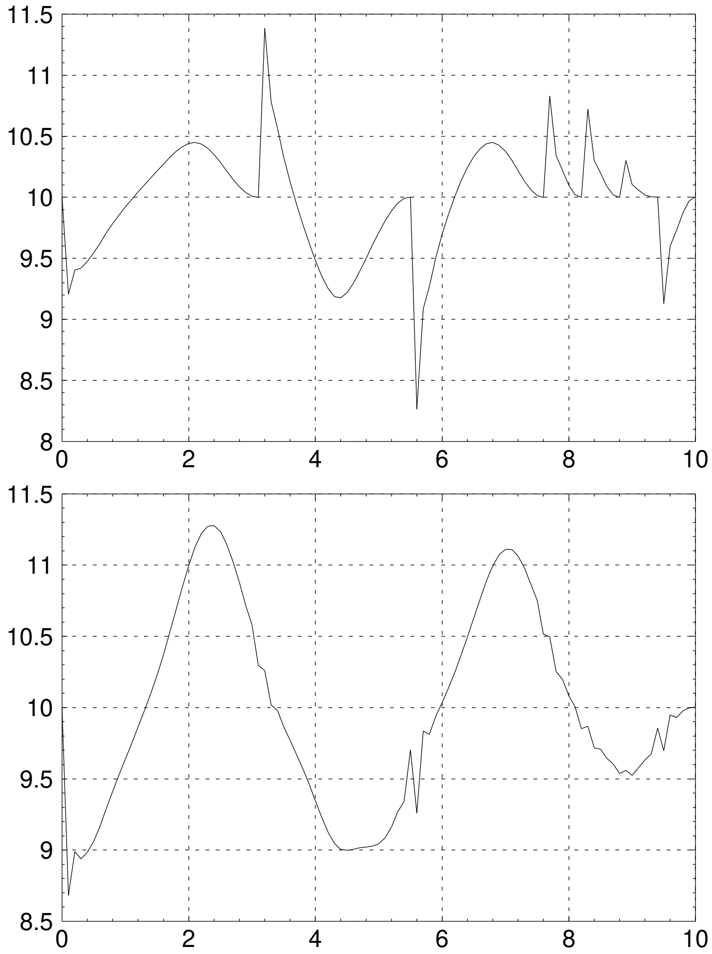


Figure 25: Slider-Crank. First stage (up) and second stage (down) optimized input T_h for the high-fidelity model, for $\nu = 0, \alpha = 0.005$, and $T_l = 10$

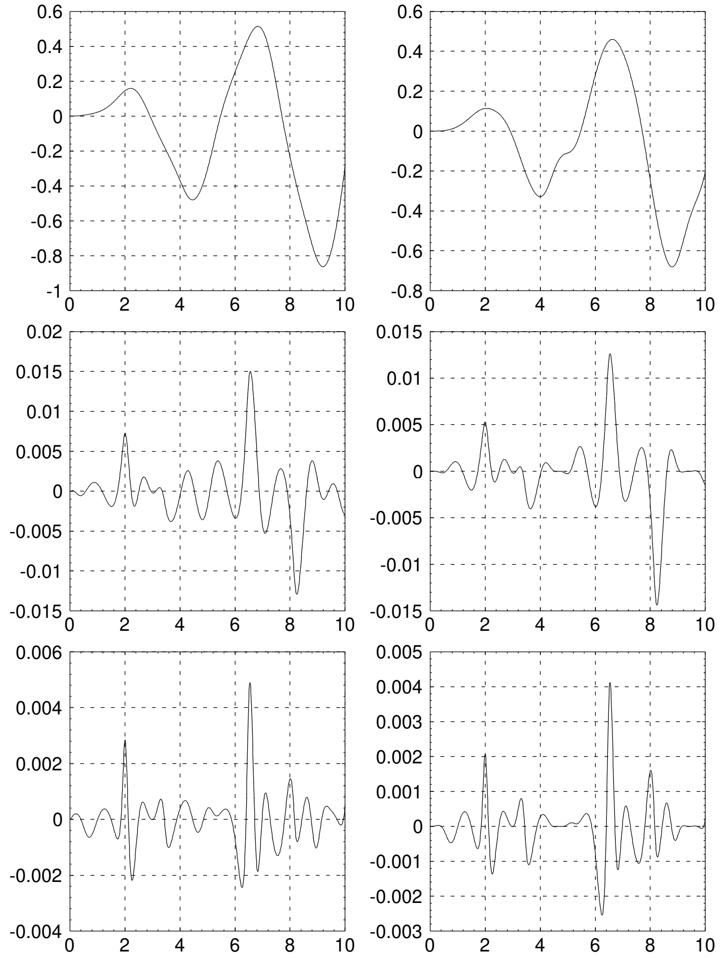


Figure 26: Slider-Crank. Deviations of the high-fidelity solution from the low-fidelity solution using the 2-stage optimization, for $\nu = 3, \alpha = 0$, and $t_1 = 10s$: $\theta_h - \theta_l$ and $x_h - x_l$ before optimization (up), after the first stage (middle), and after the second stage (down) of optimization

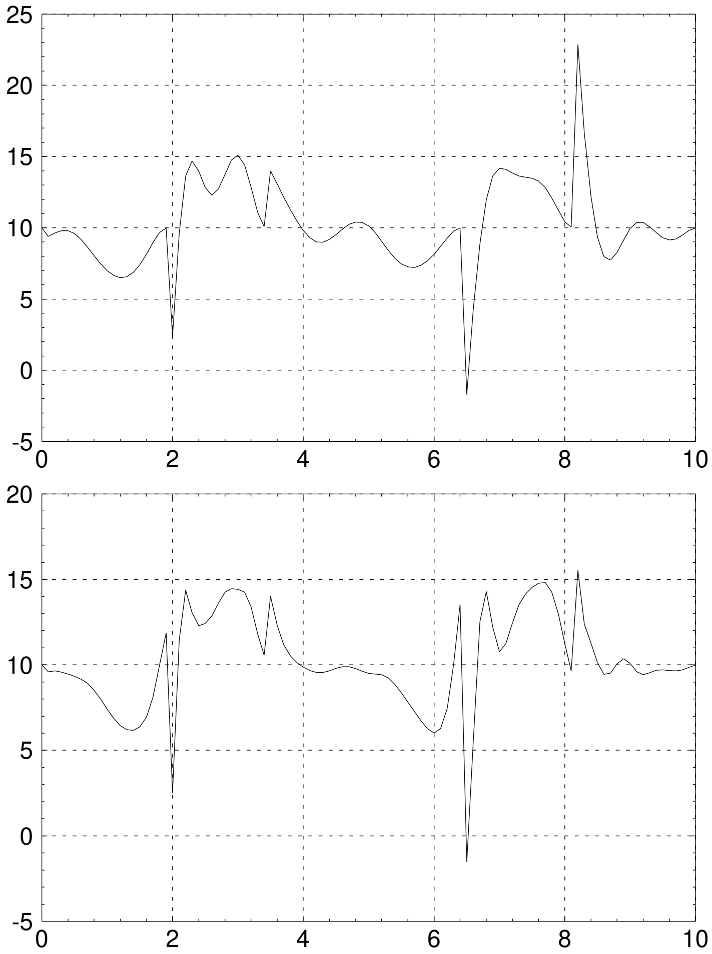


Figure 27: Slider-Crank. First stage (up) and second stage (down) optimized input T_h for the high-fidelity model, for $\nu = 3, \alpha = 0$, and $T_l = 10$

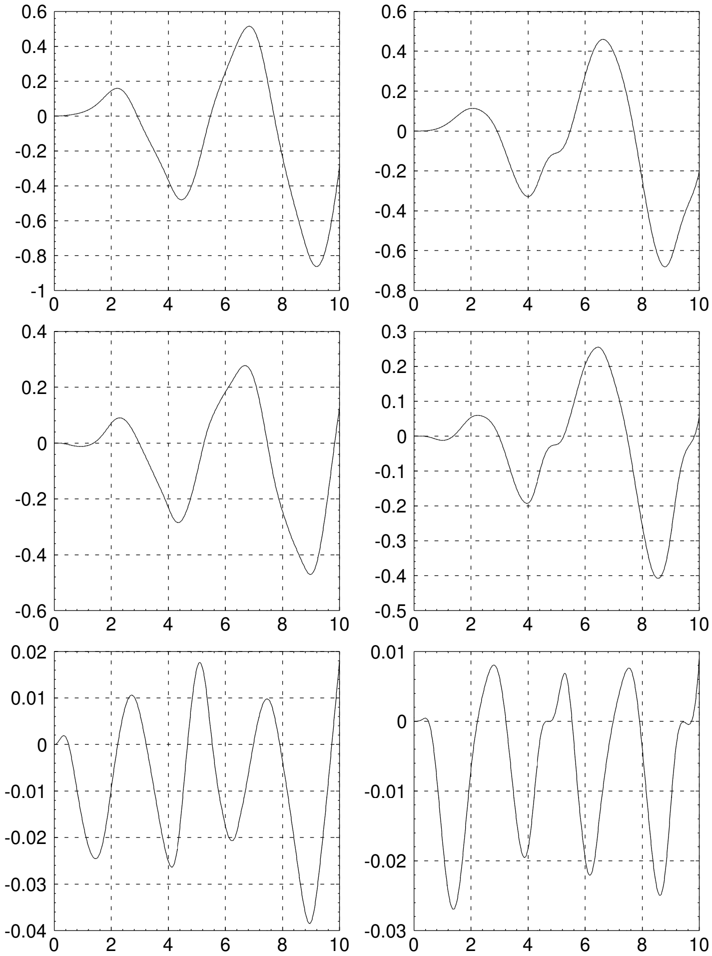


Figure 28: Slider-Crank. Deviations of the high-fidelity solution from the low-fidelity solution using the 2-stage optimization, for $\nu = 3, \alpha = 0.001$, and $t_1 = 10s$: $\theta_h - \theta_l$ and $x_h - x_l$ before optimization (up), after the first stage (middle), and after the second stage (down) of optimization

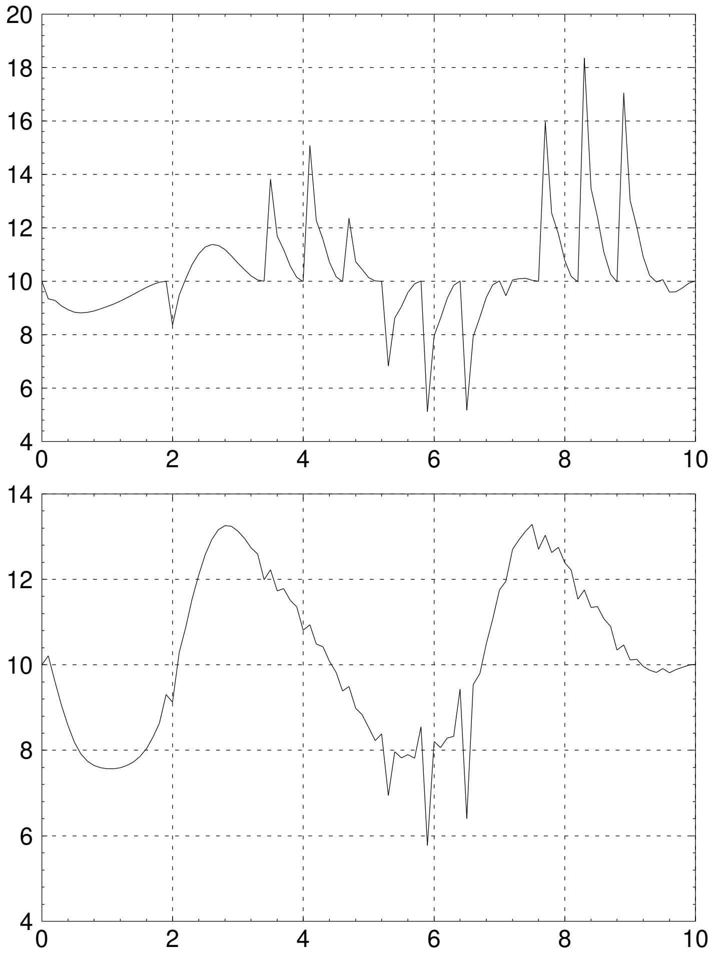


Figure 29: Slider-Crank. First stage (up) and second stage (down) optimized input T_h for the high-fidelity model, for $\nu = 3, \alpha = 0.001$, and $T_l = 10$

CHAPTER 7
CONCLUSIONS

Some results on the relationships between generalized inverses and overdetermined systems, in the context of numerical integration for differential-algebraic equations, have been presented. It was proved that a large class of outer inverses can be used to solve, in a generalized sense, the overdetermined nonlinear system obtained by discretizing the ODAE. Convergence is guaranteed, provided the initial estimate is sufficiently close to the solution. Outer inverses for arbitrary rectangular matrices can be obtained using the technique presented in the second chapter of the thesis. This procedure for finding outer inverses can be implemented numerically, and it has proven to be fast and reliable.

For large time steps, an interesting behavior was observed. When the Newton method is triggered (i.e., the tolerance of the residual exceeds a certain level), the accuracy will improve at the beginning of the simulation, but will become worse at the end of the simulation. One explanation could be that, contrary to some other DAE methods, the numerical method proposed does not impose that a part of the system is strictly satisfied. Therefore, a drift off the manifold is experienced and the errors accumulate quickly. This problem could be remedied by using, from time to time, an outer inverse that forces equality on the manifold, or by stopping integrating variables that should be determined algebraically. An interesting direction of research could be on how to choose a suitable outer inverse for a specific problem.

The numerical algorithm proposed in the fifth chapter of the thesis for finding all the tangent vectors for a class of multiple bifurcation points, that occur when using continuation methods, can be easily implemented and applied to real problems, where the described type of bifurcation occurs.

The accuracy of error sensitivity calculated, for variable fidelity differential-algebraic equation model corelation, using the proposed adjoint variable method for differential-algebraic equations is demonstrated to be adequate for iterative minimization of error functions representing deviation in response between models of differing fidelity. The piecewise linear class of input function parameterizations used in the examples presented demonstrates its effectiveness in representing time varying inputs that are to be determined in the optimization process, hence permitting direct use of established nonlinear programming computer programs. Piecewise optimization of input over subintervals of the period of simulation is shown to reduce computer time required for solution, but leads to approximate solutions that are less regular than those obtained by optimization over the entire period of simulation.

REFERENCES

[1] F. A. Adkins and D. I. Coroian. Boundary calculation and reduction of dimension by intersecting with planes are commutative processes. Technical Report R-197, Center for Computer Aided Design, University of Iowa, Iowa City, Iowa 52242, USA, 1995.

[2] E. L. Allgower and K. Georg. *Numerical Continuation Methods: An Introduction.* Springer-Verlag, Berlin, 1990.

[3] Th. Andrzejewski, H. G. Bock, E. Eich, and R. von Schwerin. Recent advances in the numerical integration of multibody systems. In W. Schiehlen, editor, *Advanced Multibody System Dynamics, Simulation and Software Tools*, pages 127–152. Kluwer Academic Publishers, London, 1993.

[4] M. Anitescu, D. I. Coroian, M. Z. Nashed, and F. A. Potra. Outer inverses and multi-body system simulation. *Numer. Funct. Anal. Optimization*, 17:661–678, 1996.

[5] U. Ascher and L. Petzold. The numerical solution of delay-differential-algebraic equations of retarded and neutral type. , University of Minnesota, 1992. submitted to SINUM.U.

[6] U. Ascher and L. Petzold. Stability of computational methods for constrained dynamics systems. *SIAM J. Sci. Stat. Comput.*, 14:95–120, 1993.

[7] D. S. Bae and E. J. Haug. A recursive formulation for constrained mechanical system dynamics. I and II. *Mech. Struct. & Mach.*, 15:359–382 and 481–506, 1987.

[8] D. Bestle. *Analyse und Optimierung von Mehrkörpersystemen.* Springer-Verlag, Berlin, 1994.

[9] V. Brasay and E. Hairer. Half-explicit runge kutta methods for differential-algebraic systems of index 2. *SIAM J. Numer. Anal.*, 2:538–552, 1993.

[10] K. E. Brenan, S. L. Campbell, and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations.* Elsevier Science Publishers, 1989.

[11] K. E. Brenan and B. E. Enquist. Backward differentiation approximations of nonlinear differential/algebraic equations,and suplement. *Math.Comp.*, 51:659–676,S7–S16, 88.

[12] J. C. Butcher. Initial value problems: Numerical methods and mathematics. *Computers Math.Applic.*, 28:1–16, 1994.

[13] G. D. Byrne and P. R. Ponzi. Differential-algebraic systems,their applications and solutions. *Computers and Chemical Engineering*, 12:377–382, 1988.

[14] D. I. Coroian and F. A. Adkins. Calculation of continuation curves at multiple bifurcation points when mapping the boundaries of manipulator workspaces. Technical Report R-199, Center for Computer Aided Design, University of Iowa, Iowa City, Iowa 52242, USA, 1996.

[15] C. Führer and B. Leimkuhler. Formulation and numerical solution of the equations of constrained mechanical motion. DFVLR-FB 89-09, Munich, 1989.

[16] C. Führer and B. Leimkuhler. A new class of generalized inverses for the solution of discretized Euler-Lagrange equations. In E. Haug and R. Deyo, editors, *Real-Time Integration Methods for Mechanical System Simulation*, pages 143–154. Springer-Verlag,Berlin, 1991.

[17] C. Führer and B. Leimkuhler. Numerical solution of differential-algebraic equations for constrained mechanical motion. *Numerische Mathematik*, 59:55–69, 1991.

[18] C. W. Gear, G. K. Gupta, and B. J. Leimkuhler. Automatic integration of the Euler-Lagrange equations with constraints. *J. Comp. Appl. Math.*, 12 & 13:77–90, 1985.

[19] E. Hairer, Ch. Lubich, and M. Roche. *The numerical solution of differential-algebraic systems by Runge-Kutta methods.* Springer, LNM 1409, Berlin, 1989.

[20] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I. Nonstiff Problems.* Springer-Verlag, Berlin, Berlin, 1993. second revised edition.

[21] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems.* Springer-Verlag, Berlin, Berlin, 1991. second revised edition.

[22] E. J. Haug. Design sensitivity analysis of dynamic systems. In C. A. Mota Soares, editor, *Computer-Aided Optimal Design: Mechanical Systems*, pages 705–755. Springer-Verlag, 1987.

[23] E. J. Haug. *Computer Aided Kinematics and Dynamics of Mechanical Systems, Vol. I: Basic Methods.* Allyn and Bacon, Boston, 1989.

[24] E. J. Haug, K. K. Choi, J. G. Kuhl, and J. Wargo. Virtual prototyping simulation for design of mechanical systems. *J. of Mechanical Design*, 117:63–70, 1995.

[25] E. J. Haug, D. I. Coroian, and R. Serban. Variable fidelity differential-algebraic equation model correlation. *Mech. Struct. Mach.*, 25(1), 1996.

[26] E. J. Haug and R. Deyo. *Real-time integration methods for mechanical system simulation.* Springer-Verlag,Berlin, Berlin, 1991.

[27] E. J. Haug, C.-M. Luh, F. A. Adkins, and J. Y. Wang. Numerical algorithms for mapping boundaries of manipulator workspaces. *Journal of Mechanical Design*, 118(2):228–234, 1996.

[28] E. J. Haug, J. Y. Wang, and J. K. Wu. Dextrous workspaces of manipulators, Part I: Analytical criteria. *Mechanics of Structures and Machines*, 20(3):321–361, 1992.

[29] Ch. Lubich, U. Nowak, U. Pöle, and Ch. Engstler. Mexx-numerical software for the integration of constrained mecahnical multibody systems. Preprint Sc 92-12, ZIB, Berlin, Berlin, 1992.

[30] R. März. Some new results concerning index-3 differential-algebraic equations. *J. Math. Anal. Appl*, 140:177–199, 1989.

[31] R. März and C. Tischendorf. Solving more general index-2 differential algebraic equations. *Computers Math. Applic.*, 28:77–105, 1994.

[32] P. C. Müller, P. Rentrop, W. Kortüm, and C. Führer. Constrained mechanical systems in descriptor form: identification, simulation and control. In W. Schiehlen, editor, *Advanced Multibody System Dynamics*, pages 451–456. Kluwer Academic Publishers, Stuttgart, 1993.

[33] M. Z. Nashed. *Generalized Inverses and Applications: proceedings of an advanced seminar.* Academic Press, New York, 1976.

[34] M. Z. Nashed and X. Chen. Convergence of Newton-like methods for singular operator equations using outer inverses. *Numer. Math.*, 66:235–257, 1993.

[35] E. D. Nering. *Linear Algebra and Matrix Theory.* John Wiley & Sons, New York, 2nd edition, 1970.

[36] L. R. Petzold. Differential-algebraic equations are not ODEs. *SIAM J. Scien. Stat. Comput.***3**, 3:367–384, 1982.

[37] L. R. Petzold. Numerical solution of differential-algebraic equations in mechanical systems simulation. *Physica D*, 60:269–279, 1992.

[38] L. R. Petzold and F. A. Potra. ODAE methods for the numerical solution of Euler-Lagrange equations. *Applied Numerical Mathematics*, 10:397–413, 1992.

[39] F. A. Potra. Implementation of linear multistep methods for solving constrained equations of motion. *SIAM J. Numer. Anal.*, 30(3):774–789, 1993.

[40] F. A. Potra. Numerical methods for differential-algebraic systems with application to real-time simulation of mechanical systems. *ZAMM*, 74(3):177–187, 1994.

[41] F. A. Potra and W. C. Rheinboldt. Differential geometric techniques for solving differential algebraic equations. In E. Haug and R. Deyo, editors, *Real– Time Integration Methods for Mechanical Systems Simulation*, pages 155–192. Springer, Berlin, 1991.

[42] F. A. Potra and W. C. Rheinboldt. On the numerical solution of Euler-Lagrange equations. *Mech. Struct. Mach.*, 19(1):155–192, 1991.

[43] F. A. Potra and J. Yen. Implicit numerical integration for Euler-Lagrange equations via tangent space parametrization. *Mech. Struct. Mach.*, 19(1):77–98, 1991.

[44] C. C. Qiu, C.-M. Luh, and E. J. Haug. Dextrous workspaces of manipulators, Part III: Calculation of continuation curves at bifurcation points. *Mechanics of Structures and Machines*, 23(1):115–130, 1995.

[45] P. J. Rabier and W. C. Rheinboldt. A geometric treatment of implicit differential-algebraic equations. Tech. Rept. TR-ICMA-162, Inst. for Comp. Math. and Appl., Univ. of Pittsburg, June 1991. to appearV in J. Diff. Equations.

[46] P. J. Rabier and W. C. Rheinboldt. On the numerical solution of the Euler-Lagrange equations. Technical Report ICMA-93-177, Univ. of Pittsburg, 1993.

[47] S. Reich. On an existence and uniqueness theory for nonlinear differential-algebraic equations. *Circ. Sys. Sig. Processing*, (10):343–359, 1991.

[48] W. C. Rheinboldt. Differential-algebraic systems as differential equations on manifolds. *Math. Comp.*, (43):473–482, 1984.

[49] W. C. Rheinboldt and B. Simeon. Performance analysis of some methods for solving Euler-Lagrange equations. 1994. To appear in Appl. Math. Lett.

[50] W. Schiehlen. *Multibody Systems Handbook*. Springer-Verlag, Berlin, 1990.

[51] W. Schiehlen. *Advanced Multibody System Dynamics, Simulation and Software Tools*. Kluwer Academic Publishers,London, 1993.

[52] A. A. Shabana. *Dynamics of Multibody Systems*. Wiley-Interscience, New York, 1989.

[53] B. Simeon, C. Führer, and P. Rentrop. Differential algebraic equations in vehicle system dynamics. *Surv. on Math. in Ind.*, (1):1–37, 1991.

[54] B. Simeon, C. Führer, and P. Rentrop. Differential-algebraic equations in vehicle system dynamics. *Surv. Math. Ind.*, (I):1–37, 1991.

[55] J. Y. Wang and J. K. Wu. Dextrous workspaces of manipulators, Part II: Computational methods. *Mechanics of Structures and Machines*, 21(4):471–506, 1993.

[56] J. Yen. Constrained equations of motion in multibody dynamics as ODEs on manifolds. *SIAM J. Numer. Anal.*, 30(2):553–568, 1993.

[57] M. Zennaro. Delay differential equations: Theory and numerics. Lecture Notes, Dipartimento di Scienze Matematiche, Universita di Trieste, 34100 Trieste, Italy, April 1994.