INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600
NOTE TO USERS

This reproduction is the best copy available.

UMI
RICE UNIVERSITY

A Singular Perturbation Approach to Modeling Closed Kinematic Chains

by

Jorge Alberto González García

A Thesis Submitted
in Partial Fulfillment of the Requirements for the Degree
Master of Science

Approved, Thesis Committee:

[Signatures with names and titles]

Dr. Fathi Ghorbel, Chairman,
Associate Professor
Mechanical Engineering and
Materials Science

Dr. Andrew J. Neade,
Associate Professor
Mechanical Engineering and
Materials Science

Dr. Athanasios C. Antoulas
Professor
Computer and Electrical Engineering

Dr. James B. Dabney
Assistant Professor
University of Houston-Clear Lake

Houston, Texas
August, 2000
A Singular Perturbation Approach to Modeling Closed Kinematic Chains

Jorge Alberto González García

Abstract

The purpose of this work is to develop a singular perturbation-based approach to modeling closed kinematic chains. First, a kinematical analysis is developed to show that closed kinematic chains cannot be modeled in general using only independent generalized coordinates. Second, the Lagrangian formulation is used to develop the DAE system for closed kinematic chains. Next, differential algebraic equations (DAEs) are described, followed by discussion of standard techniques for their solution and the limitations of the standard techniques with respect to model-based control design. Then, a singular perturbation approach to solving the DAE that arise from closed kinematic chains is developed. Using this model makes it possible to solve an ODE which is an approximation of the DAE. Finally, the technique is illustrated using the Rice Planar Delta Robot.
To my mother Silvia
for her inspiration and eternal love
&
to my father Abelardo
for his affection and care
&
to my siblings Julissa, Silvia and Abelardo
for their words of wisdom and profound understanding.
Acknowledgments

Thanks above all to God.

Thanks to my advisor Dr. Fathi Ghorbel for his advice and support during the course of this research, also thanks to my co-advisor Dr. James Dabney for his guidance, and thanks to Dr. Andrew J. Meade and Dr. Athanasios C. Antoulas for serving in my thesis committee. Thanks to Dr. Ruvinda Gunawardana for his assistance during this project. Thanks to Fulbright, the IIE and CONACYT (ID 119134) in Mexico for supporting me in this project.

Thanks to my colleagues and friends in Houston because they made my stay an enjoyable experience. Thanks to Kun, Athanasios, Prassana and Remzi in the Dynamics group. Thanks to my friends Arturo, Cristina, Karen, Nancy, Alejandro, Eli, Obdulia, Esteban, Yadira, Fernando, Eduardo, Daniel and Doña Mari. Thanks to all the people of MESAR 1999-2000. Thanks for sharing to all of you.

Thanks to my mother Silvia for her teachings inspiration and love, to my father Abelardo and my siblings Julissa, Silvia and Abelardo for their constant care and understanding. Thanks to all my family specially my aunt Chata. Thanks to my everlasting friends: Luis, Nicolás, Carmina, Carlos, Jorge, Luis Guillermo, Jaime and Rafael simply for their friendship. Thanks to all my friends who always offer the best: Héctor, Yari, Lourdes, Carlos, Patricia, Salvador, Beatriz, Enrique, Ofelia, Carlos, Paco, Paloma and Fernando. Thanks to everybody who has offered me sincere friendship. Excuse me if I forgot your name. This list would not be complete unless I thank Lucidalia who made a world of difference for her sharing and support.
# Table of Contents

Abstract ii

Acknowledgments iv

1 Introduction 1

1.1 Advantages of closed kinematic chains 1

1.2 Difficulties in modeling closed kinematic chains 2

1.3 Special characteristics of DAEs 3

1.4 Thesis Overview 4

2 Kinematics of Constrained Multibody Systems 6

2.1 Open Kinematic Chains 7

2.1.1 Particles 8

2.1.2 A Free Rigid Body 9

2.1.3 A Constrained Rigid Body 10

2.1.4 Multi-body Mechanisms Forming an Open Chain 12

2.2 Closed Kinematic Chains 15

2.2.1 Multi-body Mechanisms Forming a Closed Chain 15

3 Dynamics of Constrained Multibody Systems 19

3.1 Lagrangian Mechanics 19

3.1.1 Lagrange's Equations of Motion with Independent Generalized Coordinates 21
3.1.2 Lagrange's Equations of Motion with Constrained Generalized Coordinates ........................................... 22

3.2 Mechanical Systems and the Reduced Model ........................................ 24

4 Differential Algebraic Equations ........................................ 27

4.1 Basic Definitions ........................................ 28

4.2 Types of DAEs ........................................ 29

4.3 Comparison between DAEs and ODEs ........................................ 30

4.4 Index of General DAE Systems ........................................ 32

4.4.1 The Differential Index of Mechanical Systems ........................................ 34

5 Numerical Solution of DAEs ........................................ 36

5.1 Multistep Methods ........................................ 37

5.2 Other Approaches ........................................ 38

6 Singular Perturbation Formulation ........................................ 43

6.1 Standard Singular Perturbation Model ........................................ 43

6.2 Transformation from DAE to SPF ........................................ 44

6.3 Example: A Robotic Arm ........................................ 46

6.3.1 Experimentation ........................................ 48

7 Application of the SPF to the Rice Planar Delta Robot (RPDR) ........................................ 58

7.1 Derivation of the DAEs for the RPDR ........................................ 58

7.2 The Reduced Model ........................................ 63

7.3 The Singular Perturbation Formulation ........................................ 65

7.4 Simulation: Implementation and Results ........................................ 66

8 Conclusions and Future Work ........................................ 81
Chapter 1

Introduction

Closed kinematic chains are an important class of mechanical systems that have several useful properties. However, closed kinematic chains possess mathematical properties that present special challenges to modeling and control. This thesis describes an innovative means of modeling closed kinematic chains via singular perturbations that addresses these special challenges and which may be quite useful for feedback control of these systems.

A closed kinematic chain is a multibody mechanical system containing at least one kinematical loop. A multi-body system is a a collection of subsystems called bodies, components or substructures in which the motion of the subsystems is kinematically constrained by different types of joints. Figure 1.1 (a) illustrates a typical open kinematic chain, a serial robot. Notice that there is only one path from any reference point in the serial robot to every other point in the robot. Figure 1.1 (b) illustrates a closed kinematic chain, a parallel robot. Here, for example, there are two paths from joint 1 to the end effector: one path consisting of link 1 and link 3, the other consisting of the base, link 2 and link 4.

1.1 Advantages of closed kinematic chains

Closed kinematic chains have several advantages relative to open chains in certain applications: For example, not all joints of parallel robots have to be actuated, so actuators can be placed closer to the base. This makes the moving part to possess lower mass inertia when moving and greater accelerations can be achieved on the
end-effector. Also, the weight rigidity ratio is considerably augmented because of
the multiple supports of the end-effector. Parallel robots are very suitable for fast
assembly lines. Examples of this are the three degree of freedom Swiss Delta robot [1]
and its six degree of freedom extension the Hexa robot [2]. Other examples of closed
kinematic chains are the delta robot, the slider-crank mechanism, and the Stewart
platform.

![Diagram](image_url)

**Figure 1.1:** Examples of Parallel and Serial Robots [3]

1.2 Difficulties in modeling closed kinematic chains

A closed kinematic chain is much more complicated to model than an open kinematic
chain. Additionally controller design is not trivial because we might end up with
more equations of motion than degrees of freedom. One way of modeling closed
kinematic chains is through Differential Algebraic Equations (DAEs). There are a
few other approaches that different authors have followed but these techniques refer
to specific systems and include equations which are implicit in nature (see [3] for more detail). An open kinematic chain can be modeled using differential equations because each body in the system can be described by a set of coordinates which is independent from coordinates of other bodies. A closed kinematic chain is modeled using Differential Algebraic Equations because it is not possible to model them with independent generalized coordinates. A common procedure to obtain the differential algebraic equations that represent a closed kinematic chain, consists in cutting open the closed chain at a non-actuated link. From here the equations of motion are obtained for the different open link chains and the constraint equations are added to the system. The result is a system of equations of differential and algebraic equations. The result will also have more generalized coordinates than degrees of freedom in the system. To reduce the system that represents a closed kinematic chain using only the same number of coordinates as the number of degrees of freedom it is necessary to manipulate mathematically the equations of motion. In general it is not possible to obtain explicitly these equations and we will show this in chapter 2. To model the equations of motion of a closed kinematic chain we could stick to the DAEs and try solving them. There are different techniques to solve DAEs which will be presented in chapter 5. DAEs present a series of challenges because their special characteristics.

1.3 Special characteristics of DAEs

Our challenge is to study DAEs analyze and simulate them effectively. The term differential-algebraic equation (DAE) as we already mentioned, refers to those differential equations whose variables are algebraically constrained. Differential algebraic equations unlike ordinary differential equations present a series of problems that must be studied to solve them. DAEs are classified according to their differential index. The differential index is the number of times an algebraic equation has to be differen-
tiated to obtain the derivative of the algebraic term explicitly (see section 4.4). When the index of a DAE system exceeds 2, the algebraic equations must be differentiated to reduce the index and solve them more efficiently. In a closed kinematic chain the algebraic equation represents a physical constraint that must be followed at every point. Differentiating this constraint could drift away the solution from the original constraint due to the error introduced by the numerical computations. Furthermore, DAEs and not ODEs must be given an initial condition that must comply with the algebraic constraints. This is a consequence of having simultaneously differential and algebraic equations in a mathematical model. Finally DAEs must have inputs that must be sufficiently smooth because unlike ODEs their solution also depend on the inputs.

1.4 Thesis Overview

This thesis is organized as follows. In Chapter 2, we will show that closed kinematic chains, unlike open kinematic chains, cannot in general be described with generalized coordinates equal in number to the degrees of freedom of the system. In Chapter 3, the Lagrangian formulation will be presented and the a representation of constrained mechanical systems and closed kinematic chains will be studied. We will also reduce the model of the mechanical system in terms of constrained variables to independent variables with local results as was explained in [3]. In Chapter 4, DAEs will be defined and the differences between DAEs and ODEs will be pointed out. Also, we will define rigorously the differential index. In Chapter 5 we will explain standard numerical methods used today to solve DAEs and observe how all these methods require precise initial conditions and are not suitable for controller design. In Chapter 6 we will develop the singular perturbation approach that will permit us have a model suitable for control design. We will also see that the approximate solution is very close to
the exact solution given a small error in the initial conditions. In Chapter 7 the Rice Planar Delta Robot (RPDR) is used to illustrate the results obtained experimentally. Finally, chapter 8 presents conclusions and recommendations for future work.
Chapter 2

Kinematics of Constrained Multibody Systems

The purpose of this chapter is to present an overview of the kinematics of open and closed kinematic chains. We will show that in general open kinematic chains can always be modeled using independent generalized coordinates and that closed kinematic chains cannot in general be modeled using independent generalized coordinates.

Independent generalized coordinates are geometrical quantities that uniquely define the position of a kinematical system. When a system is stated using more generalized coordinates than degrees of freedom, constraint equations are needed to completely characterize the system. The number of degrees of freedom is equal to the number of generalized coordinates minus the number of independent equations of constraint. It is important to point out that the number of degrees of freedom is a characteristic of the system itself, and does not depend upon the particular set of coordinates used [5]. We will show that in general we cannot obtain a model of closed kinematic chains using as many variables as degrees of freedom because mathematically the equations of constraints do not permit to obtain explicitly and unambiguously all variables.

The notation that we will be using throughout this thesis is as follows: \( \mathbb{R} \) denotes the set of real numbers, \( \mathbb{R}^n \) denotes the \( n \)-dimensional vector space over \( \mathbb{R} \) endowed with the Euclidean norm \( ||x|| = \left\{ \sum_{i=1}^{n} x_i^2 \right\}^{\frac{1}{2}} \), and \( \mathbb{R}^{n \times m} \) denotes the set of all \( n \times m \) matrices of real numbers. We will also use the dot notation to specify a derivative with respect to time. For instance: \( \dot{x} = dx/dt \).
In section 2.1, we will start with the analysis of open kinematic chains. We will define the parameterization \( q' = \sigma(q) \), where \( q' \) are dependent generalized coordinates and \( q \) are independent generalized coordinates, and show that this parameterization always holds explicitly. To show this, we will analyze a particle, a free body, a constrained body and a series of constrained bodies. In section 2.2 we will analyze a series of constrained bodies forming a closed kinematic chain to show how the parameterization \( q' = \sigma(q) \) is not possible to obtain explicitly. Constraints bring certain challenges in the analysis of the kinematics of closed chains. We will conclude that closed kinematic chains cannot be modeled using only independent generalized coordinates equal in number to the degrees of freedom of the system.

2.1 Open Kinematic Chains

Generalized coordinates are usually chosen to have a geometrical significance. The configuration of a free mechanical system can be described using coordinates that we will call \( q' \) as was done in [3]. The set of coordinates \( q' \) are in general readily visual. When we are not able to select coordinates \( q' \) which are independent we will call the constraints among the coordinates \( q' \):

\[
\phi(q') = 0. \tag{2.1}
\]

In general we would like to describe a system using independent generalized coordinates \( q \) which are equal in number to the number of DOF (\( n \)). To obtain \( q \) from \( q' \) there is

\[
q = \alpha(q') \tag{2.2}
\]

This parameterization is usually easy to obtain explicitly, we may simply choose the appropriate components of the vector \( q' \) which may correspond to the actuated joints [3].
Since we would like to describe the system using only independent generalized coordinates \( \mathbf{q} \), we are interested in obtaining \( \mathbf{q}' \) from \( \mathbf{q} \). The following parameterization defines this statement [3]

\[
\mathbf{q}' = \sigma(\mathbf{q}) \tag{2.3}
\]

This parameterization is always possible to obtain explicitly for open kinematic chains, but it is not in general possible for a closed kinematic chain.

If we are able to obtain explicitly and uniquely the parameterization \( \mathbf{q}' = \sigma(\mathbf{q}) \) we can derive and substitute for \( \mathbf{q}' \) in the Lagrangian formulation of the equations of motion for a dependent system. Then, we can always solve a system of equations to obtain an independent system in terms of \( \mathbf{q} \).

Now we will show how the parameterization \( \mathbf{q}' = \sigma(\mathbf{q}) \) is always explicit when we talk about open kinematic chains.

### 2.1.1 Particles

First, we would like to describe the position of the most basic thing, a particle. A particle is an idealization of a material body whose dimensions are very small when compared to the distance to other bodies. It is regarded as a point mass. In our analysis we assume the existence of an inertial system of reference (i.e. systems of reference that are either at rest or moving with uniform velocity relative to a fixed reference frame). The position of a free particle can be described in space by three parameters: \( x, y \) and \( z \) which define the coordinates of the particle in a fixed Cartesian reference frame. For this case the parameterization \( \mathbf{q}' = \sigma(\mathbf{q}) \) is automatically satisfied. For example, we may choose \( \mathbf{q}' = \mathbf{q} = [x \ y \ z]^T \), which means there are three DOF and three independent generalized coordinates.
2.1.2 A Free Rigid Body

A rigid body is a system of mass points subject to the holonomic constraints that the distance between all pairs of points remains constant throughout the motion [7]. The position of a rigid body can be described in space by six parameters: for example $x$, $y$, $z$ defining the coordinates of the origin of an axis fixed to the rigid body (such as the center of mass of the body) in the Cartesian frame, and $\theta$, $\gamma$ and $\psi$ as Euler angles in reference to coordinate system finer to the body. Here, $\theta$ is a counterclockwise rotation about the $z$ axis, $\gamma$ is a counterclockwise rotation about the current $x$ axis, and $\psi$ is a counterclockwise rotation about the current $y$ axis [8]. We know that the number of coordinates required to specify the position of the system is the number of degrees of freedom of that system, therefore according to our previous discussion, we can conclude a free three-dimensional body is a physical system with six degrees of freedom. The parameterization $\mathbf{q}' = \sigma(\mathbf{q})$ is also automatically satisfied as in the case for the particle because the body is unconstrained. For example,
\( \mathbf{q}' = \mathbf{q} = [x \ y \ z \ \theta \ \gamma \ \psi]^T \), which means there are 6 DOF and 6 independent generalized coordinates.

![Diagram of a Rigid Body in a Cartesian Frame](image)

**Figure 2.2: A Rigid Body in a Cartesian Frame**

### 2.1.3 A Constrained Rigid Body

A constrained rigid body is a free rigid body with additional constraints with respect to other bodies. A constrained rigid body can be defined in space by six coordinates \((\mathbf{q}')\) like a free rigid body, but the coordinates must satisfy the additional constraint equations \(\phi(\mathbf{q}') = 0\). This means that not all coordinates will be independent, and each scalar constraint equation reduces the degrees of freedom of the system by one. The position of any point in a rigid body can be determined using the position of a reference point and its orientation coordinates. The constraint equations can be obtained mathematically using these position equations. These equations computed for each link represent the parameterization \(\mathbf{q}' = \sigma(\mathbf{q})\) and is always possible to obtain \(\mathbf{q}'\) explicitly because there is only one equation that defines the position of a determined point in a constrained rigid body.
To illustrate, suppose a robot link is pinned at one end using a revolute joint as shown in figure 2.3. We have constrained its movement to an $xy$ plane, and constrained point $A$ to coincide with the origin, $O$, of our reference axis. We have in fact applied five constraints to the system: $A_x = 0$ and $A_y = 0$, $A_z = 0$, $\gamma = 0$ and $\psi = 0$. These constraints can be transformed using as a reference the center of mass of the rigid body. Then, we have $x = L/2 \cos \theta$ and $y = L/2 \sin \theta$ where $x$ and $y$ are the distances from the $x$ and $y$ axis of our fixed reference to the center of mass, $C$, of the rigid body and $L/2$ is the distance from $O$ to $C$. We define $L$ as the total length of the link and assume the link has a mass uniformly distributed. Since we have six generalized coordinates with five scalar constraints, we now have a rigid body with one DOF. For this case the parameterization $q' = \sigma(q)$ gets a bit more complicated than for the case of a free body. Choosing $q' = [x \ y \ z \ \theta \ \gamma \ \psi]^T$ as in the case for the free system (notice that the system might be described by three coordinates instead of six because there are three DOF in a system constrained to move in a plane). Now $q$ is not equal to $q'$ because we have additional constraints. A variable that describes completely the position of this constrained link is $\theta$, so we
choose \( q = [\theta] \). To obtain the parameterization \( q' = \sigma(q) \) we simply observe the constraints that we have applied and conclude

\[
q' = \begin{bmatrix}
x \\
y \\
z \\
\theta \\
\gamma \\
\psi
\end{bmatrix} = \begin{bmatrix}
L/2 \cos \theta \\
L/2 \sin \theta \\
0 \\
\theta \\
0 \\
0
\end{bmatrix}
\]  \hspace{1cm} (2.4)

Now the parameterization is explicitly defined. We may express the entire system in terms of independent generalized coordinates only since the number of explicit variables \( q \) is equal to the degrees of freedom of the system.

2.1.4 Multi-body Mechanisms Forming an Open Chain

Multi-body mechanisms are a series of rigid bodies with constraints among each other. The term open-chain refers to the fact that there is only one path on the chain in the entire system to go from any point in the system to any other point in the system. The approach that we followed for a single rigid body can be generalized to multi-body mechanical systems. Let us consider the most general case first. We have \( m \) free bodies each of them described by 6 parameters, for example, let us choose the three Cartesian coordinates \( x, y \) and \( z \) and the Euler angles \( \theta, \gamma \) and \( \psi \). Hence the system is totally described by \( 6m \) parameters or DOF. However these parameters have algebraic relations between them that reduce the DOF of the system.

When we have \( m \) bodies connected sequentially, we have \( m \) relationships shown in figure 2.4

\[
\begin{align*}
1A &= O \\
1B &= 2A \\
2B &= 3A \\
\vdots \\
\{m - 1\}B &= \{m\}A
\end{align*}
\]  \hspace{1cm} (1-3)
The free system composed of \( m \) bodies is described by \( 6m \) parameters:

\[
    x_1 \quad y_1 \quad z_1 \quad \theta_1 \quad \gamma_1 \quad \psi_1 \quad x_2 \quad y_2 \quad \theta_2 \quad \gamma_2 \quad \psi_2 \quad \ldots \quad x_m \quad y_m \quad z_m \quad \theta_m \quad \gamma_m \quad \psi_m
\]

where \( x_i, y_i \) and \( z_i \) are the Cartesian coordinates of the center of mass of each link, and \( \theta_i, \gamma_i \) and \( \psi_i \) are the Euler angles.

In open kinematic chains, as we said before, there is one and only one path to obtain the position of any point in a link in the system with respect to any reference point in the chain. In other words, there is exactly one non-redundant equation that mathematically defines the position of any point in the system with respect to a reference point. These position definitions are the constraint equations of the system.

To illustrate, consider figure 2.4. We obtain the following equations of constraint

\[
    x_1 = \frac{L}{2} \cos \theta_1
\]

\[
    y_1 = \frac{L}{2} \sin \theta_1
\]
\begin{align*}
z_1 &= 0 \\
x_2 &= L \cos \theta_1 + \frac{L}{2} \cos \theta_2 \\
y_2 &= L \sin \theta_1 + \frac{L}{2} \sin \theta_2 \\
z_2 &= 0 \\
\vdots \\
x_m &= L \cos \theta_1 + L \cos \theta_2 + \ldots + \frac{L}{2} \cos \theta_m \\
y_m &= L \sin \theta_1 + L \sin \theta_2 + \ldots + \frac{L}{2} \sin \theta_m \\
z_m &= 0
\end{align*}

If we constrain the system using prismatic joints the equations are even simpler choosing the appropriate coordinates.

Since there is only one equation that defines the position of each link, then we can say that each link is unambiguously identified. As is the case for a single rigid body, the parameterization \( \mathbf{q}' = \sigma(\mathbf{q}) \) is equivalent to the position constraints. Then, we can choose

\[
\mathbf{q}' = \begin{bmatrix}
x_1 \\
y_1 \\
\theta_1 \\
x_2 \\
y_2 \\
\theta_2 \\
\vdots \\
x_m \\
y_m \\
\theta_m
\end{bmatrix} = \begin{bmatrix}
L/2 \cos \theta_1 \\
L/2 \sin \theta_1 \\
\theta_1 \\
L \cos \theta_1 + L/2 \cos \theta_2 \\
L \sin \theta_1 + L/2 \sin \theta_2 \\
\theta_2 \\
\vdots \\
L \cos \theta_1 + L \cos \theta_2 + \ldots + L/2 \cos \theta_m \\
L \sin \theta_1 + L \sin \theta_2 + \ldots + L/2 \sin \theta_m \\
\theta_m
\end{bmatrix}
\tag{2.5}
\]

Here we chose \( \mathbf{q} = [\theta_1 \ \theta_2 \ \ldots \ \theta_m] \). Now the parameterization is explicitly defined. We may express the entire system in terms of independent generalized coordinates.
only since the number of explicit variables $q$ is equal to the degrees of freedom of the system.

2.2 Closed Kinematic Chains

2.2.1 Multi-body Mechanisms Forming a Closed Chain

Closed kinematic chains are systems with links containing at least one kinematic loop. This means that for some points on the system there are two or more paths on the chain from some reference point in the chain. Multi-body mechanisms containing a closed loop are called closed kinematic chains. We will show that in general closed kinematic chains cannot be modeled using independent generalized coordinates. We will show that the parameterization $q' = \sigma(q)$ is not explicit for closed kinematic chains and therefore we cannot obtain the equations of motion of a system in terms of independent generalized coordinates ($q$).

The analysis developed in the previous section for open kinematic chains can be extended to be used in the analysis of closed kinematic chains. For closed chain mechanisms when we have $m$ bodies connected sequentially, we have $m + 1$ relationships as shown in figure 2.5.

\[
\begin{align*}
1A &= O \\
1B &= 2A \\
2B &= 3A \\
\vdots \\
\{m-1\}B &= \{m\}A \\
\{m\}B &= O2
\end{align*}
\]

The free system composed of $m$ bodies is described by the same $6m$ parameters used in the modeling of open kinematic chains, that is

\[
x_1 \ y_1 \ z_1 \ \theta_1 \ \gamma_1 \ \psi_1 \ x_2 \ y_2 \ \theta_2 \ \gamma_2 \ \psi_2 \ \ldots \ x_m \ y_m \ z_m \ \theta_m \ \gamma_m \ \psi_m
\]
Figure 2.5: Example of multi bodies with added constraints forming a closed loop

where \( x_i, y_i \) and \( z_i \) are the Cartesian coordinates of the center of mass of each link, and \( \theta_i, \gamma_i \) and \( \psi_i \) are the Euler angles.

As said before, in closed kinematic chains there are two or more paths to obtain the position of some points in a link in the system with respect to a reference point in the system. In other words, there is more than one equation that mathematically constrains the position of some points in the system with respect to a reference point. To illustrate, we constrain a system as shown in figure 2.5, and we obtain the following \( 2m + 2 \) equations of constraint

\[
\begin{align*}
  x_1 &= \frac{L}{2} \cos \theta_1 \\
  y_1 &= \frac{L}{2} \sin \theta_1 \\
  x_2 &= L \cos \theta_1 + \frac{L}{2} \cos \theta_2 \\
  y_2 &= L \sin \theta_1 + \frac{L}{2} \sin \theta_2
\end{align*}
\]
\[
\begin{align*}
    x_m &= L \cos \theta_1 + L \cos \theta_2 + \ldots + \frac{L}{2} \cos \theta_m \\
    y_m &= L \sin \theta_1 + L \sin \theta_2 + \ldots + \frac{L}{2} \sin \theta_m \\
    x_m &= Cx - \frac{L}{2} \cos \theta_m \\
    y_m &= Cy - \frac{L}{2} \sin \theta_m
\end{align*}
\]

Closed kinematic chains are more complicated to model than open kinematic chains because there is more than one way to solve for the position of a point in any body of the system. In the previous equations we can see that we have chosen to describe the additional constraints on link \( m \). The last four equations show this fact. The point here is that these equations must be satisfied simultaneously. For open kinematic chains we did not have to solve any system of simultaneous equations because each link is unambiguously characterized by one equation.

There are problems we encounter when trying to solve these equations simultaneously: there is not a unique solution and the mathematical procedure to obtain explicitly the \( q' \) variables is not in general feasible. In addition, the solution of these equations present singularities. There is not a unique solution because both equations in general contain trigonometric functions which we know have the same numerical value for more than one angle. We can see in Figure 2.6 this issue.

In conclusion, we are not able to obtain the parameterization \( q' = \sigma(q) \) for open kinematic chains because the equations of constraint are not defined explicitly, are not unique, and there are singularities in their solution. Therefore if we cannot obtain explicitly and globally \( q' \), we are unable to obtain the system using only independent generalized coordinates, \( q \).

Modeling closed chain mechanisms in general has to be done through differential algebraic equations (DAEs), and with more generalized coordinates than degrees of
freedom. Extending the research on open-chains to close-chains is not trivial because we end up with more equations of motion than DOF and the resulting system is a system of DAEs. In chapter 4 we will understand how we can obtain a model based on DAEs.
Chapter 3

Dynamics of Constrained Multibody Systems

We have studied the kinematics of constrained multibody systems. Now we will study the dynamics through a review of Lagrangian mechanics. We will also show a representation of the equations of motion for mechanical systems and we will go through the development of a reduced model obtained in [3].

3.1 Lagrangian Mechanics

The Lagrange formulation makes it possible to obtain the equations of motion of a system without actually solving for the reaction forces. We will start defining the different types of constraints that exist. Constraint equations can be written in the following form:

$$\phi_i(q_1, q_2, \ldots, q_M, t) = 0$$  \hspace{1cm} (3.1)

as it was pointed out in chapter 2.

We may use instead a velocity constraint, which imposes an identical constraint as equation (3.1) if the initial position is specified. The velocity constraint is obtained differentiating (3.1):

$$\dot{\phi}_i = \sum_{j=1}^{M} \left[ \frac{\partial}{\partial q_j} \phi_i(q_1, q_2, \ldots, q_M, t) \right] \dot{q}_j + \frac{\partial}{\partial t} \phi_i(q_1, q_2, \ldots, q_M, t) = 0$$  \hspace{1cm} (3.2)

A more general form of constraint equation (3.2) is sometimes necessary in mechanical systems

$$\sum_{j=1}^{M} a_{ij} (q_1, q_2, \ldots, q_M, t) \dot{q}_j + b_i(q_1, q_2, \ldots, q_M, t) = 0$$  \hspace{1cm} (3.3)
A velocity constraint (3.2) is identical to the constraints illustrated by (3.3) if the coefficients of equation (3.2) and the coefficients of equation (3.3) are identical within a multiplicative factor. This factor could be a function of the generalized coordinates and time, \( g_i = g_i(q_1, q_2, ..., q_M, t) \). This means that the following equations must be satisfied

\[
a_{ij}g_i = \frac{\partial \phi_i}{\partial q_j}, \quad b_ig_i = \frac{\partial \phi_i}{\partial t} \quad (3.4)
\]

The constraint equation (3.3) is said to be holonomic if equation (3.4) is satisfied, and is said to be nonholonomic if it is not satisfied. The terms \( \frac{\partial \phi_i}{\partial q_j} \) constitute the Jacobian of a set of holonomic constraints. The coefficients \( a_{ij} \) are referred as the Jacobian Constraint Matrix, even when the constraints are nonholonomic [9]. Another variable that we have to take into account is time \( t \). Constraints in which time does not appear explicitly are known as scleronomic constraints. On the other hand, if constraints are given as an explicit function of time, the constraints are classified as rheonomic. These terms can also be applied to mechanical systems. A system is scleronomic if (1) none of the constraint equations contain \( t \) explicitly and (2) the transformation equations in terms of the generalized coordinates do not contain \( t \) [5]. Holonomic scleronomic constraints are the constraints that we will be using in general throughout this thesis.

We will introduce a few additional concepts that will help in the understanding of how the Lagrangian formulation works out.

**Virtual Displacements**

Consider a system of \( N \) particles and define the virtual displacements \( \delta x_1, \delta y_1, \delta z_1, ..., \delta z_N \) as infinitesimal changes in the coordinates \( x_1, y_1, z_1, x_2, ..., z_N \). Virtual displacements are consistent with the constraints of the system, but otherwise arbitrary, and they take place without change in time. This means that they are not real displacements, and no forces and no constraints change during this process [10].
Virtual Work
Consider a particle under a variety of forces. When this particle is subjected to a virtual displacement $\delta r$, the forces acting on the particle do virtual work, $\delta W$. Virtual work is infinitesimal because the virtual displacement is also infinitesimal.

Reaction Forces
A force acting on a system is a reaction force associated with a constraint condition if, and only if, the reaction force does no virtual work in a virtual displacement of the system.

3.1.1 Lagrange's Equations of Motion with Independent Generalized Coordinates

Lagrange's Equations of motion for the case when the number of generalized coordinates equals the number of degrees of freedom of the system reduces to the following equations

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{q}_j} \right) - \frac{\partial T}{\partial q_j} + \frac{\partial V}{\partial q_j} = Q_j, \quad j = 1, 2, ..., M \quad (3.5)$$

where $T$ is the kinetic energy of the system, $V$ is the potential energy, and $Q$ are the nonconservative forces. Since the potential energy does not depend on the generalized velocities $\dot{q}_j (j = 1, 2, ..., M)$, we can define the lagrangian as

$$L = T - V \quad (3.6)$$

and therefore simplify equation (3.5) to

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} = Q_j, \quad j = 1, 2, ..., M \quad (3.7)$$

Many problems do not involve nonconservative forces in which cases $Q_j = 0 \ (j = 1, 2, ..., n)$, and Lagrange's equation reduces to

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} = 0, \quad j = 1, 2, ..., M \quad (3.8)$$
The complete derivation of Lagrange's equation can be reviewed in [9], [11], [10], or [8] among others.

### 3.1.2 Lagrange's Equations of Motion with Constrained Generalized Coordinates

Now we will start to concentrate in the fact that we might formulate a problem using more generalized coordinates than degrees of freedom, so we have constrained generalized coordinates. Each constraint condition would be violated if the generalized coordinates were chosen arbitrarily. The constraint force associated with each generalized coordinate is an unknown reaction that does no virtual work, and therefore appears as an unknown in the generalized force equation. Lagrange's equations for constrained generalized coordinates are written as

$$\frac{d}{dt} \left( \frac{\partial L}{\partial q_j} \right) - \frac{\partial L}{\partial q_j} = Q_j + R_j, \quad j = 1, 2, ..., M$$  \hspace{1cm} (3.9)

where $Q_j$ is the contribution of the given applied forces to the $j$th generalized force, and $R_j$ is the contribution of all reactions. To completely define the system we need $M - N$ constraint conditions that must be explicitly satisfied. The constraint conditions have the form of equation (3.3) and are written

$$\sum_{j=1}^{M} a_{ij} \dot{q}_j + b_i = 0 \quad i = 1, 2, ..., M - N.$$  \hspace{1cm} (3.10)

If there is no need to obtain the reaction forces, then it is possible to account for them through the Lagrange Multipliers [9]. The constraint equations (3.10) can be rewritten as

$$\sum_{j=1}^{M} a_{ij} \dot{q}_j + b_i dt = 0 \quad i = 1, 2, ..., M - N.$$  \hspace{1cm} (3.11)

This is the Pfaffian form of the velocity constraint. Since we are considering a virtual displacement, time is held constant, $dt = 0$, and all the differential increments
are substituted by virtual increments. So we have

\[ \sum_{j=1}^{M} a_{ij} \delta q_j = 0 \quad i = 1, 2, ..., M - N. \]  \tag{3.12} \]

Now we will let \( R^{(i)} \) be the vector of reaction forces attributable to constraint condition \( i \). The component \( R_j^{(i)} \) of this vector is the contribution of the reaction associated with this constraint to the \( j \)th generalized force. Since a reaction force does no work in a virtual movement, then we can conclude that \( R \) must be perpendicular to any \( \delta r \) consistent with the \( i \) constraint condition. Thus we have

\[ \sum_{j=1}^{M} R_j^{(i)} \delta q_j = 0 \]  \tag{3.13} \]

If we multiply equation (3.12) by a factor \( \lambda_i \) which is the \textit{Lagrange Multiplier}, we obtain \( M - N \) equations.

\[ \lambda_i \sum_{j=1}^{M} a_{ij} \delta q_j = 0 \quad \tag{3.14} \]

Now, we will subtract the sum of equation (3.14) from equation (3.13) and obtain

\[ \sum_{j=1}^{M} \left( R_j^{(i)} - \sum_{i=1}^{M-N} \lambda_i a_{ij} \right) \delta q_j = 0 \]  \tag{3.15} \]

Up to this point, \( \lambda \)'s have been considered to be arbitrary, and \( \delta q \)'s must conform with equation (3.12). If we choose \( \lambda \)'s such that

\[ R_j^{(i)} = \sum_{i=1}^{M-N} \lambda_i a_{ij} \]  \tag{3.16} \]
then equation (3.15) will apply for any set of \( \delta q \)'s, which means they can be chosen independently.

It follows that equation (3.9) becomes

\[ \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} = Q_j + \sum_{i=1}^{M-N} \lambda_i a_{ij}, \quad j = 1, 2, ..., M \]  \tag{3.17} \]

This is the standard form of Lagrange’s equations. In addition to this equation, we have to solve the \( M - N \) holonomic constraint equations (3.10).
3.2 Mechanical Systems and the Reduced Model

Using Lagrangian dynamics, equation (3.7), we can obtain a model of the equations of motion of a mechanical system when we use the same number of coordinates as degrees of freedom. In the absence of friction and other disturbances, Spong [12] proposed the following $n$-degree of freedom mechanical system representation of the equations of motion:

$$ D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = 0 \quad (3.18) $$

where $q \in \mathbb{R}^n$ is the vector of generalized coordinates, $D(q)$ is the $n \times n$ inertia matrix, $C(q, \dot{q})\dot{q}$ represents the Coriolis and centrifugal terms and $g(q)$ represents the gravitational terms.

For closed kinematic chains, Ghorbel et al. [3] considered a holonomic dynamical model called the free system, which represents a collection of interconnected open chain rigid bodies with $n'$ DOF. Equation (3.17) can be rewritten as follows:

$$ D'(q')\ddot{q}' + C'(q', \dot{q}')\dot{q}' + g'(q') = 0 \quad (3.19) $$

where $q' \in \mathbb{R}^n$ is the vector of generalized coordinates, $D'(q')$ is the $n' \times n'$ inertia matrix, $C'(q', \dot{q}')\dot{q}'$ represents the Coriolis and centrifugal terms, $g'(q')$ represent the gravitational terms. Then, in [3], the authors applied $(n' - n)$ scleronomic holonomic constraints as it was explained in chapter 2 and given by $\phi(q') = 0$, equation (3.10).

Our next step is to reduce the system as was explained in [3]. By reducing the system, it is understood that will end up with the same number of equations of motion than degrees of freedom. We will go through a review of the procedure taken in [3] because we will use the reduced model to solve the system in the following chapters.

Now by combining the equations of constraints $\phi(q') = 0$, (2.1), and the parameterization $\alpha(q') = q$, (2.2),

$$ \psi(q') = \begin{bmatrix} \phi(q') \\ \alpha(q') \end{bmatrix} \quad (3.20) $$
as was defined in [3]. Differentiating with respect to \( q' \),

\[
\psi^\prime(q') = \frac{\partial \psi}{\partial q'}
\]  

(3.21)

We are interested in obtaining the system in terms of \( q \), therefore the parameterization \( q' = \sigma(q) \), previously defined in chapter 2, is required to reduce the model. We now want to obtain \( \dot{q}' \) since the system depends on it. We will take the time derivative of \( \psi(q') \)

\[
\psi^\prime(q') \dot{q}' = \begin{bmatrix} 0_{(n'-n)\times n} \\ I_{n \times n} \end{bmatrix} \dot{q}
\]

(3.22)

Rewriting it as follows

\[
\dot{q}' = \rho(q') \dot{q}
\]

(3.23)

where

\[
\rho(q') = \psi^{-1}_{q'}(q') \begin{bmatrix} 0_{(n'-n)\times n} \\ I_{n \times n} \end{bmatrix}.
\]

(3.24)

We are assuming that \( \psi(q') \) is of full rank. We will also need the derivative of \( \rho(q') \) to reduced the model. Since \( \rho(q') \) is in terms of an inverse matrix, it is not easy to take the time derivative. Therefore, we use the following expression for \( \dot{\rho}(q', \dot{q}') \), which can be easily obtained by pre-multiplying \( \rho(q') \) with \( \psi^\prime_q(q') \) and taking the time derivative:

\[
\dot{\rho}(q', \dot{q}') = -\psi^{-1}_{q'}(q') \psi^\prime_q(q', \dot{q}') \rho(q'),
\]

(3.25)

where \( \psi^\prime_q(q', \dot{q}') \) can be obtained by differentiating (3.21) with respect to time.

We now consider the virtual work expression \( \delta W \) associated with the dynamics of the constrained system (3.19) as it was done in [3], namely,

\[
\delta W = \delta q'^T \begin{bmatrix} D'(q') \dot{q}' + C'(q', \dot{q}') \dot{q}' + g'(q') - \phi^T_q(q') \lambda \end{bmatrix}
\]

(3.26)

We now by the definition of virtual work that \( \delta W = 0 \) for any set of \( \delta q' \). Since \( \delta q' \) conforms to the instantaneous constraints, then the virtual work due to the constraints
(reaction forces) vanishes that is, \( \delta q^T \phi_{q'}^T \lambda = 0 \), and consequently

\[
\delta W = \delta q^T [D'(q')q' + C'(q', q')\dot{q}' + g'(q')] = 0
\]

(3.27)

From equation (3.23) and assuming a virtual displacement we can conclude that

\[
\delta q' = \rho(q')\delta q
\]

(3.28)

recalling that we need \( \psi(q') \) to be of full rank. It follows, using equations (3.27) and (3.28) that

\[
\delta W = \delta q^T \left[ \rho(q')^T D'(q')q' + \rho(q')^T C'(q', q')\dot{q}' + \rho(q')^T g'(q') \right] = 0
\]

(3.29)

Since the generalized coordinates \( q \) are independent, we obtain

\[
\rho(q')^T D'(q')q' + \rho(q')^T C'(q', q')\dot{q}' + \rho(q')^T g'(q') = 0
\]

(3.30)

The reduced system is:

\[
D(q')\ddot{q} + C(q', q')\dot{q} + g(q') = 0,
\]

(3.31)

where

\[
D(q') = \rho(q')^T D'(q')\rho(q'),
\]

\[
C(q', q') = \rho(q')^T C'(q', q')\rho(q') + \rho(q')^T D'(q')\dot{q}',
\]

\[
g(q') = \rho(q')^T g'(q'),
\]

In [3] and [13], equation (3.31) was used in conjunction with a parameterization

\[
\dot{q}' = \rho(q')\dot{q},
\]

\[
q' = \sigma(q)
\]

to solve the system. In general, it is not possible to derive an analytic expression for \( \sigma(q) \) in closed kinematic chains as it was explained in chapter 2. Usually it must be computed using numerical methods.
Chapter 4

Differential Algebraic Equations

Differential equations arise in many instances in areas of engineering, science and economics to describe the way physical systems change. Differential equations that describe physical phenomena are called mathematical models. These models often involve more than one differential equation, which means we have a system of differential equations. Our interest in this thesis will concern a special system of equations which involve differential equations coupled with algebraic equations (equations which do not contain differential variables), and this is due to the nature of our application. These equations are called Differential Algebraic Equations (DAEs). A more rigorous definition will be given in the following section.

Many physical systems are described by DAEs (e.g. constrained mechanical systems, fluid dynamics, the kinetic of chemical reactions, electrical networks, and control engineering), which explains the interest for research activity in this field. Depending on the area, DAEs have been called singular, implicit, differential-algebraic, descriptor, generalized state space, noncanonic, noncasual, degenerate, semistate, constrained, reduced order model, and nonstandard systems [14]. The literature emphasizes the numerical solution of DAEs because the interest them has arisen from analyzing and modeling actual physical systems. Therefore numerical computations are used to simulate system behaviors.

We will begin by stating basic mathematical definitions and the implications that we will assume about differential equations before we get into the type of systems to be used.
4.1 Basic Definitions

We begin by defining a system of Implicit Ordinary Differential Equations according to [15]:

Definition 4.1 Implicit Ordinary Differential Equations (ODEs)

A system of equations containing the ordinary derivatives (i.e. not partial derivatives) of one or more dependent variables, with respect to a single independent variable, is said to be a system of implicit ODEs. The general form of a system of implicit ODE is:

\[ f(\dot{x}, x, t) = 0 \]  \hspace{1cm} (4.1)

where \( x(t) \in \mathbb{R}^n, t \in \mathbb{R} \) is the independent variable, and \( f \in \mathbb{R}^n \) is a function supposed to have sufficient bounded derivatives, and its initial condition, \( x(0) \), is specified.

ODEs are typically set in the form of definition 2.2 as a result of mathematical manipulation or simply because ODEs models arise typically in this manner.

Definition 4.2 Explicit Ordinary Differential Equation

If the ODE from definition 2.1 can be written as

\[ \dot{x} = f(x, t) \]  \hspace{1cm} (4.2)

then equation (4.2) is called system of explicit ODEs.

Systems of simultaneous equations can sometimes include ODEs and pure algebraic equations (i.e. equations that do not contain any kind of derivatives) depending on the application or the way the model is approached. This leads us to the following definition:
**Definition 4.3** Differential Algebraic Equations (DAEs)

If a system of equations contains differential and algebraic equations (equations with no derivatives) or in other terms, the Jacobian of \( f \) in equation (4.1), denoted by \( \partial f / \partial \dot{x} \), is singular, then we can say the system is a system of DAEs.

### 4.2 Types of DAEs

It is important to differentiate between the different forms of DAEs because solutions to them depend on their form. The simplest form a DAE might take is the **linear constant coefficient form** [14]:

\[
A \dot{x}(t) + Bx(t) = f(t) \tag{4.3}
\]

where \( x \in \mathbb{R}^n \) is the vector of space variables, \( f \in \mathbb{R}^n \) is arbitrary, and \( A, B \in \mathbb{R}^{n \times n} \) are constant matrices.

The second form of DAEs is very similar to the first with the difference that matrices \( A \) and \( B \) from equation (4.3) can be functions of time. A **linear time varying** DAE system can be characterized by the following equation [14]:

\[
A(t) \dot{x}(t) + B(t)x(t) = f(t) \tag{4.4}
\]

where \( x \in \mathbb{R}^n \) is the vector of state variables, \( f \in \mathbb{R}^n \) is arbitrary, and \( A, B \in \mathbb{R}^{n \times n} \) are matrices involving time, \( t \). Linear time varying DAEs exhibit most of the behavior that nonlinear DAEs have, but the linearity many times facilitates the analysis.

If the differential and algebraic equations are separated, the system is a semi-explicit DAE. The form of a **semi-explicit linear time varying** DAE is as follows [14]:

\[
\dot{x} = B_{11}(t)x(t) + B_{12}(t)z(t) + f_1(t) \tag{4.5}
\]

\[
0 = B_{21}(t)x(t) + B_{22}(t)z(t) + f_2(t) \tag{4.6}
\]
where $x \in \mathbb{R}^n$ is the differential variable, $z \in \mathbb{R}^p$ is the algebraic variable, $t \in \mathbb{R}$, $B_{11}$ and $B_{21} \in \mathbb{R}^{(n \times n)}$, $B_{12}$ and $B_{22} \in \mathbb{R}^{(p \times p)}$, $f_1(t) \in \mathbb{R}^n$, and $f_2(t) \in \mathbb{R}^p$.

For nonlinear systems the implicit form of equation (4.1) does not allow the development of explicit controller synthesis results. Furthermore, the majority of mechanical applications (as well as other engineering applications) are modeled by DAEs systems in the semi-explicit form where there is a clear separation between the differential and algebraic equations. This leads us to consider the following general nonlinear semi-explicit system [14]:

$$
\dot{x} = f(x, z, t) \quad (4.7)
$$

$$
0 = g(x, z, t) \quad (4.8)
$$

where $x \in \mathbb{R}^n$ is the differentiable variable, $z \in \mathbb{R}^p$ is the algebraic variable, $t \in \mathbb{R}$, and $f(x, z, t)$ and $g(x, z, t)$ are appropriately dimensioned functions into $\mathbb{R}^n$ and $\mathbb{R}^p$ respectively.

Understanding the challenges that DAEs present when solving them is important because it will tell us about the feasibility of simulating the physical system that is many times helpful for understanding its behavior. The following section makes a comparison between DAEs and ODEs which will help us understand DAEs mathematical characteristics much better.

### 4.3 Comparison between DAEs and ODEs

Consider an explicit DAE described by equations (4.7) and (4.8). There are basic characteristics that makes DAEs different from ODEs when we talk about their solution and we will summarize them as follows:

- The singular algebraic equations in (4.8) imply the presence of underlying algebraic constraints in the differential variables $x$. For some systems of DAEs the
algebraic equations have to be differentiated several times to obtain a solution for the algebraic variable, \( z \), and the differentiation process imposes additional constraints in \( x \). Depending on the algorithm chosen to solve the DAE, it could be possible that the original constraints are not met due to the error introduced by the numerical computation on the differentiated constraint. If the constraint represents a physical characteristic of a system such a deviation could have critical importance.

- As a consequence of the previous point, arbitrary initial conditions \( x(0) \) do not yield smooth solutions but an impulsive behavior at time \( t = 0 \), and this is due to the fact that physically this is not possible. To better explain this point, consider a system composed only of algebraic equations. When we solve this system, we obtain a finite number of solutions depending only on the equations and not on any initial condition. Now consider a system composed only of differential equations. An initial condition together with the differential equations define a solution, and there exists an infinite number of solutions depending on the initial conditions. When a system of differential-algebraic equations arises from a physical plant, we have to input an initial condition required to solve the differential part taking into account that the algebraic equations have a definite solution and therefore the initial condition must comply with such a constraint to make converge must of the numerical methods already available.

- DAE systems, unlike ODE systems, might have a solution, \( x(t), z(t) \) that are functions of the time derivatives of the inputs, in which case, these inputs must be sufficiently smooth [16].

As we have seen there are challenges in the solution of DAEs compared with the well developed solution algorithms of ODEs. In section 5 we will mention different
approaches mentioned in the literature to have a better understanding of the numerical solution of DAEs. Now we will focus on important subjects that will help us in the analysis of systems that are modeled through DAEs.

### 4.4 Index of General DAE Systems

Various indices have been defined in the literature, but the most referred is the differential index, $v_d$, or simply called the index of a DAE system. The higher the index is, the less reliable simulations are with classical numerical methods. In section 5 we will list methods proposed in the literature to solve DAEs and explain why solutions of higher index DAEs are not as reliable as those of index 1 or 2. Now let us concentrate on the definition of differential index.

**Definition 4.4** The minimum number of times that all or part of (4.1) must be differentiated with respect to $t$, in order to determine $\dot{x}$ as a continuous function of $x$ and $t$, is the differential or global index of the DAE.

An explicit ODE system is a special class of DAE systems with an index 0.

The differential index provides a measure of the "singularity" of the algebraic equations and their differences with respect to ODEs systems. In semi-explicit systems the index is determined by differentiating only the algebraic part [17], as shown in the following examples. Considering equation (4.8), if only one differentiation of the algebraic equations in (4.8) or the solution of $z$ yields the differential equations for $z$, then the system is of index $v_d = 1$.

The simplest situation is that of a system of the form

$$
\begin{align*}
\dot{x} &= f(x, z) \\
0 &= g(x, z)
\end{align*}
$$
by differentiating \( 0 = g(x, z) \) with respect to \( t \), one obtains

\[
\frac{\partial g}{\partial x} \dot{x} + \frac{\partial g}{\partial z} \dot{z} = 0
\]

Hence we can solve for \( \dot{x} \) and \( \dot{z} \) provided that \( \frac{\partial g}{\partial z} \neq 0 \). Which will result in:

\[
\dot{x} = f \\
\dot{z} = \left[ \frac{\partial g}{\partial z} \right]^{-1} \left[ -\frac{\partial g}{\partial x} f \right]
\]

Since we have differentiated one time in order to obtain this result, the system is of index 1.

Another more elaborate situation could be:

\[
\dot{x} = f(x, z) \\
0 = g(x)
\]

By differentiating \( 0 = g(x) \) with respect to \( t \), one obtains

\[
\frac{dg}{dx} \dot{x} = 0
\]

substituting \( \dot{x} \) yields:

\[
\frac{dg}{dx} f(x, z) = 0
\]

Hence, by differentiating this equation with respect to \( t \), one obtains

\[
\left[ \frac{dg}{dx} \frac{\partial f}{\partial x} + \frac{d^2 g}{dx^2} f \right] \dot{x} + \left[ \frac{dg}{dx} \frac{\partial f}{\partial z} \right] \dot{z} = 0
\]

assuming \( \frac{dg}{dx} \frac{\partial f}{\partial z} \neq 0 \), we can then solve for \( \dot{x} \) and \( \dot{z} \). Indeed:

\[
\dot{x} = f \\
\dot{z} = \left[ \frac{dg}{dx} \frac{\partial f}{\partial z} \right]^{-1} \left[ -\frac{dg}{dx} \frac{\partial f}{\partial x} - \frac{d^2 g}{dx^2} f \right] f
\]

Since we have differentiated twice in order to obtain this result, the system is of index 2.
The concept of index is not intended to show how many times we have to differentiate in order to have an equivalent ODE system. The differential index is important because it gives a notion of the feasibility for the numerical computation as it will be explained in section 5.

4.4.1 The Differential Index of Mechanical Systems

In chapter 3 we introduced the classical mechanics description of the equations of motion of the first kind to model rigid bodies. We will repeat here for convenience

\[ D'(q')\ddot{q}' + C'(q', \dot{q}')\dot{q}' + g'(q') = \phi_q^T(q')\lambda \]  \hspace{1cm} (4.9)

\[ \phi(q') = 0 \]  \hspace{1cm} (4.10)

They can be rewritten in the so called “descriptor form” or semi-explicit form

\[ \dot{q}' = v' \]  \hspace{1cm} (4.11)

\[ D'(q')\dot{v}' = -C''(q', v')v' - g'(q') + \phi_q^T(q')\lambda \]  \hspace{1cm} (4.12)

\[ 0 = \phi(q') \]  \hspace{1cm} (4.13)

where \( q' \) is the vector of generalized position coordinates, \( v' \) is the velocity vector corresponding to each \( q' \) and \( \lambda \) is the lagrange multiplier vector.

Index according to the constraint used:

- Equations (4.11), (4.12) and (4.13) model a constrained mechanical system as an index 3 DAE.

- When we obtain kinematical constraints (i.e. derive (4.13) once with respect to time, \( \phi_q'(q'')q' = 0 \)) we have modeled the constrained mechanical system as an index 2 DAE.
• If we derive the constraint, equation (4.13), twice with respect to time, $\phi_{q'}(q')\ddot{q}' + \dot{\phi}_{q'}(q', \dot{q}')\dot{q}' = 0$, we have the representation of the constrained mechanical system as an index 1 system.

Notice that to obtain an index 0 DAE (i.e. an ODE) we must do mathematical manipulation using the acceleration constraints (i.e. the constraints that gave us an index 1 DAE) and get rid of the Lagrange multiplier, $\lambda$ which is an algebraic variable in the system.
Chapter 5

Numerical Solution of DAEs

As explained in section 4.4, lower index DAEs are easier to solve numerically than higher index DAEs. DAEs which cannot be solved explicitly for the algebraic variable have a high differential index, $\nu_d > 1$, and cannot be readily reduced into an ODE system. The algebraic equations in (4.8) imply the presence of underlying algebraic constraints in the differentiable variables $x$. If the differential index exceeds two, the algebraic equations have to be differentiated several times to obtain a solution for the purely algebraic variable $z$, therefore the differentiation process imposes additional constraints in $x$. In other words, although the differentiation imposes constraints that are equivalent to the original ones, both constraints are not exactly the same due to the error introduced by the numerical computations. This is of vital importance for models in which the constraints represent a physical characteristic of the system.

Furthermore, DAE systems have a smooth solution in a numerical computation for $x(t)$ and $z(t)$, if the initial condition $x(0)$ satisfies the algebraic constraints. In contrast, a smooth solution exists for an ODE system for any initial condition $x(0)$. Also, unlike ODE systems, the solution $x(t)$ and $z(t)$ of high index DAE systems (4.7) and (4.8) may also depend on the time derivatives of the inputs $u(t)$, in which case the requirement is to have sufficiently smooth inputs [16]. These are common causes of failure in the numerical solution for DAE systems.
5.1 Multistep Methods

The most popular method to solve DAEs is the backward differentiation formula (BDF), which is a multistep method. Gear [18] proposed in 1971 one of the first techniques to solve DAEs numerically using BDF. The simplest first order BDF is the implicit Euler method, which consists in replacing the derivative, \( \dot{x} \) by a backward difference:

\[
\frac{x_m - x_{m-1}}{h}
\]

where \( h = t_m - t_{m-1} \) is the step size. The resulting nonlinear algebraic system is then solved by Newton’s method. Another method to solve DAEs is the \( k \)-step (constant step-size) BDF, which consists in replacing \( \dot{x} \) by a derivative defined by

\[
\frac{\rho x_m}{h}
\]

where \( \rho x_m = \sum_{i=0}^{k} \alpha_i x_{m-i} \) and \( \alpha_i, i = 0, 1, \ldots, k \) are the coefficients of the BDF. The reader is referred to [19], [20], [21] for explanation on the properties of BDF and to [22] for an example on BDF implementation on DAEs.

Söderlind [23] developed another multistep method for semi-explicit index 1 systems, where the system is partitioned in stiff and non-stiff subsystems. The non-stiff subsystem was solved using the classical fourth-order Runge-Kutta method, while the stiff subsystem was solved using the three-step BDF. Liniger [24] considered a two-step one leg formula for index 1 DAEs where the algebraic constraints are clearly identified. The advantage of this method comes in the stability properties in the case the system is very nonlinear or when frequent step sizes are needed. Griepentrog and März [25] conducted an analysis for the convergence for linear multistep methods.

Index 1 systems are primarily solved using BDF, and this is due to the extraordinary stability and accuracy properties of BDF which is extended to the solution of many higher index DAEs. In the 1980s a second generation of BDF was developed as
a result of recognizing the importance of DAEs in engineering applications. Examples of these codes are DASSL and LSODI. Many DAEs are solved using this codes, but still not all DAEs can be solved using BDF.

Most of the integration methods for DAEs consist in adapting the standard methods for ODEs to the implicit differential equation

$$f(\dot{x}, x, t) = 0$$

(D.1)

DASSL is one of the first codes developed to solve DAEs and is based on the following approach. For a typical time step from \( t_{m-1} \) to \( t_m \), a backward differentiation formula replaces the derivative \( \dot{x} \) in (5.1) by a difference operator defined by

$$\rho x_m = \frac{\alpha_0}{h} x_m + \frac{1}{h} \sum_{i=1}^{l} \alpha_i x_{m-i}$$

(5.2)

where \( \alpha_i \in \mathbb{R} \) and \( h \) is the step size. On every step the code DASSL chooses the order of the BDF and the step size based on the behavior of the solution. This way the solution of a system of implicit differential equation is reduced to the solution of a system of nonlinear algebraic equations

$$F(\rho x_m, x_m, t_m) = 0$$

(5.3)

which can be solved using Newton’s method. Newton’s method converges most rapidly when the initial guess \( x_n(0) \) is accurate. To obtain the initial guess, DASSL evaluates the polynomial which interpolates the computed solution at the last \( k+1 \) (\( k \) is the BDF order and ranges from 1 to 5) times \( t_{n-1}, t_{n-2}, \ldots, t_{n-(k+1)} \), at the current time \( t_n \). Details of the method and algorithms used are discussed in [26].

5.2 Other Approaches

One-step methods, more specifically the Implicit Runge-Kutta method (IRK) has also been the focus of some interesting research. IRK methods show significant advantages
over BDF when DAEs exhibit frequent discontinuities. They are potentially more
efficient because multi-step methods have to be restarted at a low order after every
discontinuity and IRK methods can start at a higher order. IRK can also be used to
generate accurate starting values for higher BDF methods, in consequence exploiting
the advantages of each method [14].

The interested reader concerned with the issue of potential error on the original
constraints might find interesting the following methodologies for solving DAEs.

Gear [27] proposed a way of introducing constraints that have been lost through
differentiations or other manipulations to have a system with a structure which is
computationally tractable. The framework of this idea is as follows. The classical
mechanics Lagrange equations of the first kind that we have been using to describe
the motion of rigid bodies are

\[
D'(q')\ddot{q}' + C'(q', \dot{q}')\dot{q}' + g'(q') = u + \phi_q^T(q')\lambda \tag{5.4}
\]

\[
\phi(q') = 0 \tag{5.5}
\]

and they can be rewritten in the so called "descriptor form"

\[
\dot{q}' = v' \tag{5.6}
\]

\[
D'(q')v' = -C'(q', v')v' - g'(q) + u + \phi_q^T(q')\lambda \tag{5.7}
\]

\[
0 = \phi(q') \tag{5.8}
\]

where \(q'\) is the vector of generalized position coordinates, \(v'\) is the velocity vector
corresponding to each \(q'\), \(\lambda\) is the lagrange multiplier and \(u\) is the nonconservative
input torque. Applying the formulation we are trying to obtain we get

\[
\dot{q}' = v' + \phi_q^T(q')\mu \tag{5.9}
\]

\[
D'(q')v' = -C'(q', v')v' - g'(q') + u + \phi_q^T(q')\lambda \tag{5.10}
\]

\[
0 = \phi(q') \tag{5.11}
\]
0 = \phi_{q'}(q')v' \hfill (5.12)

This is called the "stabilized index two formulation". Its advantage is the fact that
the position and velocity level constraints are automatically enforced thus eliminating
the drift from the original physical constraints.

Führer and Leimkuhler [28] and Rheinboldt [29] describe the numerical treatment
of a DAE formulation taking into account the constraints and also one or more deriva-
tives. This approach results in an over-determined system of differential-algebraic
equations. If we use the standard form that we have been using for a mechanically
constrained systems, and adding constraints on position, velocity and acceleration we
have a system:

\begin{align}
D'(q')\ddot{q}' + C'(q', \dot{q}')\dot{q}' + g'(q') &= u + \phi_q^T(q')\lambda \\
\phi(q') &= 0 \\
\phi_{q'}(q')\dot{q}' &= 0 \\
\phi_{q''}(q')\ddot{q}' + \dot{\phi}_{q''}(q', \dot{q}')\dot{q}' &= 0
\end{align} \hfill (5.13-5.16)

For consistent initial values and with usual smoothness requirements, the over-determined
system has a unique solution. To solve numerically over-determined DAEs first the
system is transformed into a problem of nonlinear algebraic equations and then lin-
earized applying Newton's schemes. A classical method to solve these is through
Gauss-Newton iteration. A problem it presents is the fact that nothing is known
about the propagation of errors. It would first appear that the numerical computa-
tion required was not efficient because of the linear algebra involved, but this is not
the case because the stabilized index two formulation is extended to maintain also
the acceleration level by means of additional Lagrange multipliers. Since the purpose
of this chapter is only to highlight possible solutions options, the reader is referred to
[28] and [29] for a in depth explanation of the numerical method and for techniques to improve the efficiency of this approach.

Finally, the singular perturbation approach is a way to solve DAEs as stiff ODEs, and it is the approach we will be concentrating in the following chapter.

Methods to solve DAEs most of the time cannot be used directly to solve the equations of motion due to the following reasons:

- Most of the methods are convergent for index 1 systems only. In the case of high index DAEs, there is a loss of approximation order on the algebraic variables, which causes standard techniques of error estimation and step-size selection to fail.

- When integrating the solution of an index reduced DAE, the numerical solution tends to drift away from a constraint manifold. In other words, the numerical solution does not satisfy the constraints on position level, which is very important in a mechanical system. Additionally, the index transformation may cause changes in the stability of the DAE relative to perturbations in the solutions.

- Computational effort is increased because the methods do not exploit the special structure of the equations of motion.

As we have seen, there exist in the literature many formulations for solving index 1 systems. Many of these algorithms have been extrapolated for higher index DAEs, but they only solve higher index DAEs with specific characteristics. Higher index DAEs require more complex calculations than lower index DAEs because they undergo numerical differentiation and additional constraints have to be taken into account. Common to all of these techniques are the measures to accurately determine a consistent set of initial conditions, and measures to continuously maintain the constrain satisfaction. These measures are computationally inefficient. Moreover, the
numerical solution of DAEs, presented this far, is not suitable for controller design because they leave the system in DAE form and therefore not suitable for model-base control design. Our focus in the following chapter will be in obtaining an ODE as a singular perturbation model using a general DAE system. This new approach overcomes these difficulties.
Chapter 6

Singular Perturbation Formulation

As we have seen in the previous chapters, closed kinematic chains are modeled using index 3 DAEs. Solving high index DAEs (index greater than 1) presents many disadvantages, therefore, we will now study an approximation that simplifies the solution of the equations of motion. This model is based on a singular perturbation.

Singular Perturbation ODE systems originate as models of chemical, biochemical, electric circuits systems, etc. The connection that exists between singular perturbation systems and DAEs is evident if we take a look at the equations of motion. The analysis of closed chain mechanisms does not generate a singular perturbation model, but the solution of the DAE representing the closed kinematic chain can be very well approximated in that manner. Singular perturbation systems are characterized by the presence of a small parameter $\epsilon$. When the parameter $\epsilon$ is approximated to zero, the system becomes a system of DAEs. We will do the inverse: we will add a variable that is multiplied by a small parameter $\epsilon$ to convert our DAEs into a singular perturbation model. The model depends on how small is $\epsilon$ and that is the reason it is an approximation.

6.1 Standard Singular Perturbation Model

The standard singular perturbation model has to comply with certain conditions. It can be described by the following equations

$$\dot{x} = f(t, x, z, u, \epsilon), \quad x \in \mathbb{R}^n$$

(6.1)
\[ \epsilon \dot{z} = g(t, x, z, u, \epsilon), \quad z \in \mathbb{R}^m \] (6.2)

where we assume that \( f \) and \( g \) are continuously differentiable, and \( x \in \mathcal{Z} \subset \mathbb{R}^n \), \( z \in \mathcal{Y} \subset \mathbb{R}^m \) are the state variables with \( \mathcal{Z}, \mathcal{Y} \) being open and connected sets, and \( u \in \mathbb{R}^m \) is the vector of control inputs.

As mentioned before the small parameter \( \epsilon \) is the singular perturbation parameter. When \( \epsilon \to 0 \) the differential equation system reduces to a system of DAEs. Equation (6.2) becomes

\[ 0 = g(t, x, z, u, \epsilon) \] (6.3)

We say we have a singular perturbation model whenever equation (6.3) has \( k \geq 1 \) isolated roots

\[ z = h_i(t, x, u), \quad i = 1, 2, \ldots, k \] (6.4)

Our objective is to transform a DAE into a singular perturbation model in a logical manner and this will be discussed in the following section.

### 6.2 Transformation from DAE to SPF

As explained in Chapter 4, the dynamics of a closed kinematic chain can be fully characterized by a semi-explicit system of DAEs. To facilitate the development of the singular perturbation form, let us choose matrices \( \mathbf{A} \) and \( \mathbf{B} \) such that we extend the previous parameterization given by \( \mathbf{q} = \alpha(\mathbf{q'}) \) in chapter 2 to the linear parameterization

\[ \begin{bmatrix} \mathbf{q} \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \mathbf{q'} \] (6.5)

where \( \mathbf{q} \) is partitioned in two parts \((\mathbf{q}, \mathbf{z})\) and the combined selector matrix is full rank as was explained in [30]. Here \( \mathbf{q} \) is the coordinates that we want to use to describe the system.
Now we can rewrite the system used to describe a closed chain in chapter 3 as follows

\[ D(q, z)\ddot{q} + C(q, \dot{q}, z)\dot{q} + g(q, z) = 0 \]  
\[ \phi(q, z) = 0 \]  

Following the reasoning in [31] and [30], we define

\[ \mathbf{w} = \phi(q, z) \]  

where \( \mathbf{w} \) is an artificial state variable that represents the constraint error. Since \( \mathbf{w} \) is arbitrary, we are free to choose its dynamics. We wish for \( \mathbf{w} \) to asymptotically approach zero, so we define

\[ \dot{\mathbf{w}} = -\frac{1}{\epsilon} \mathbf{w} \]  

where \( \epsilon \) is a small previously selected positive number. We can also obtain \( \dot{\mathbf{w}} \) from (6.8)

\[ \dot{\mathbf{w}} = \phi_q \dot{q} + \phi_z \dot{z} \]  

so that

\[ \phi_z \dot{z} = -\frac{1}{\epsilon} \phi - \phi_q \dot{q} \]  

Thus, the pair (6.6) and (6.11) together form an ODE representation of the system dynamics. Due to the use of the small parameter \( \epsilon \), this system is in singular perturbation form. Also, due to (6.9), the dynamics of the constraint error are stable, and therefore this formulation is robust to small errors in initial conditions and disturbances.

We now put the system in the form of equations introduced in [3] and explained in [30]. In [3] the system is reduced but the parameterization \( \mathbf{q}' = \sigma(q) \) as discussed in chapter 2 is not in general explicit. The equations of motion of the reduced model
explained in chapter 3 using the notation of separating the algebraic and differential variables reduce to:

\[
D(q, z) = \rho(q, z)^TD'(q, z)\rho(q, z) \tag{6.12}
\]

\[
C(q, z, \dot{q}, \dot{z}) = \rho(q, z)^TC'(q, z, \dot{q}, \dot{z})\rho(q, z) + \rho(q, z)^TD'(q, z)\dot{\rho}(q, z, \dot{q}, \dot{z}) \tag{6.13}
\]

\[
g(q, z) = \rho(q, z)^Tg'(q, z) \tag{6.14}
\]

where \(D'(q, z)\) is the inertia matrix, \(C'(q, z, \dot{q}, \dot{z})\) is the centrifugal and coriolis terms \(g(q, z)\) is the gravity vector. Next we compute \(\dot{z}\). We compute the Jacobian matrices \(\phi_q\) and \(\phi_z\) and pick a small positive number \(\epsilon\). Now we have everything to compute the system

\[
D(q, z)\dot{q} = -C(q, \dot{q}, z, \dot{z})\dot{q} - g(q, z) \tag{6.16}
\]

\[
\phi_z\dot{z} = -\frac{1}{\epsilon}\phi - \phi_q\dot{q} \tag{6.17}
\]

Notice that we do not compute \(\ddot{z}\), only \(\dot{z}\). Also notice that this procedure results in an explicit model.

### 6.3 Example: A Robotic Arm

Consider a simple robotic arm illustrated in figure 6.1. We will solve its equations of motion through the Singular Perturbation Formulation using constrained generalized coordinates. So, \(x\) and \(y\) represent the coordinates of the center of mass of the arm. The inertia matrix is

\[
D' = \begin{bmatrix}
m + \frac{I_g}{l^2-x^2} & 0 \\
0 & m
\end{bmatrix} \tag{6.18}
\]

the coriolis and centrifugal terms

\[
C' = \begin{bmatrix}
\frac{I_g x \ddot{x} + C_g(l^2-x^2)}{(l^2-x^2)^2} & 0 \\
0 & 0
\end{bmatrix} \tag{6.19}
\]
and the gravity vector is

\[ g' = \begin{bmatrix} 0 \\ mg \end{bmatrix} \tag{6.20} \]

where \( m \) is the mass of the robotic arm, \( I_g \) is the moment of inertia of the robotic arm in a line through the center of mass and perpendicular to the plane of movement, \( l \) is the distance from the center of rotation to the center of mass, \( g = 9.81 m/s^2 \) is the gravitational acceleration constant and \( C_o \) is the damping coefficient. The damping coefficient is a viscous damping coefficient added to the system to model more closely its real physical behavior. We chose to model viscous damping because the physical robotic arm that we will use in our experiment is attached to the base using a pair of lubricated ball bearings.

![Diagram of a Robotic Arm](image)

**Figure 6.1: A Robotic Arm**

We form

\[ \psi(q') = \begin{bmatrix} \phi(q') \\ \alpha(q') \end{bmatrix} = \begin{bmatrix} x^2 + y^2 - l^2 \\ x \end{bmatrix}, \tag{6.21} \]

noticing that using the parameterization \( q = \alpha(q') \) we picked \( x \) as the coordinate to be used to solve the system. We now compute

\[ \psi_{q,z} = \begin{bmatrix} 2x & 2y \\ 1 & 0 \end{bmatrix} \tag{6.22} \]
and
\[
\frac{d}{dt} \psi_{q,z} = \begin{bmatrix} 2\dot{x} & 2\dot{y} \\ 0 & 0 \end{bmatrix}.
\] (6.23)

From here we can compute \( \rho \) and \( \dot{\rho} \) using equations (3.24) and (3.25).

We know the constraint is
\[
\phi = x^2 + y^2 - l^2 = 0
\] (6.24)

and therefore
\[
\phi_q = 2x
\] (6.25)

and
\[
\phi_z = 2y.
\] (6.26)

We now have all the data to solve equations (6.16) and (6.17).

6.3.1 Experimentation

We are going to compare our simulation results with real data measured in a laboratory experiment. Apparatus that we used in our experiment is a link of the Rice Planar Delta Robot (RPDR). The RPDR will be used in Chapter 7 to develop an experiment and compare it with the SPF. The robotic arm is attached to the base by two grease lubricated ball bearings. There is a sprocket attached to the arm with the center coinciding with the center of rotation of the arm. For this experiment the apparatus was not actuated; the drive chain was removed.

We used a model with one independent generalized coordinate: \( \theta \). The position \( \theta \) was traced using a program and an optical encoder connected to a PC computer through a DSP board. We ran a series of 5 experiments to obtain real data. In each experiment we arbitrarily chose an initial condition with an angle \( \theta \) close to 90 degrees and fixed it using a brake, as seen in figure 6.1. After starting the tracing program we released the brake. With the files containing the traced data, we made some manual
manipulation on each file eliminating the dead time and the data traced after the pendulum had stopped. The derivatives of the angles $\theta_i$ were computed numerically. This data was used to estimate the moment of inertia of the robotic arm in a line through the center of rotation, $I_o$, and the viscous damping coefficient, $C_o$. We fit the equations of motion of an exact model to the real data using a least square fit. We used one of the experiments to plot the results and then compare them with the theoretical results. We used as the mass of the arm $m = 0.7951$ Kg. and the distance from the center of rotation to the center of mass $l = 0.045$ m. We used Matlab [32] to make all numerical computations.

The equation of motion for the robotic arm considering damping is:

$$I_o \ddot{\theta} + C_o \dot{\theta} + mgl \cos \theta = 0 \quad (6.27)$$

Note that this equation is in terms of the independent generalized coordinate $\theta$, which is much simpler than the previous equations developed in this chapter. This is because we are simplifying the process to obtain $I_o$ and $C_o$.

We now rewrite the above equation as:

$$A_\theta b + mgl \ C = 0 \quad (6.28)$$

where

$$A = \begin{bmatrix} \ddot{\theta}_1 & \dot{\theta}_1 \\ \ddot{\theta}_2 & \dot{\theta}_2 \\ \vdots & \vdots \\ \ddot{\theta}_i & \dot{\theta}_i \end{bmatrix}, \quad b = \begin{bmatrix} I_o \\ C_o \end{bmatrix}, \quad C = \begin{bmatrix} \cos \theta_1 \\ \cos \theta_2 \\ \vdots \\ \cos \theta_i \end{bmatrix},$$

and $\theta_1, \theta_2, \ldots, \theta_i$ are the values of the real data measured at different instants of times.

We now minimize

$$\frac{1}{2} (A_\theta b + mgl \ C)^T (A_\theta b + mgl \ C) \quad (6.29)$$

and get

$$A_\theta^T (A_\theta b + mgl \ C) = 0 \quad (6.30)$$
So
\[ \mathbf{b} = -[\mathbf{A}^T \mathbf{A}]^{-1} \mathbf{A}^T \mathbf{m}_{gl} \mathbf{C} \]  
\hspace{1cm} (6.31)

We got as the numerical values \( I_o = 0.0087 \) Kg-m\(^2\) and \( C_o = 0.0032 \) N-m-s/rad.

To choose \( \epsilon \) we ran a series of experiments. Table 6.1 shows different values of \( \epsilon \), the number of floating point operations (flops) in the solution of the SPF and the number of steps required for the error of \( y \) to be reduced 98%. To do this experiment we had to choose an initial condition for \( y \) to be off the condition imposed by the constraints. The initial condition given is \( y(0) = -0.0450\text{m} \), and the initial condition imposed by the constraints is \( y(0) = -0.0416\text{m} \). Subtracting these two numbers we have the absolute initial error 3.4 \( \times \) 10\(^{-3}\)m. Reducing the error by 98% means to obtain an absolute error less than 6.8 \( \times \) 10\(^{-5}\)m. The value of \( \epsilon \) determines how fast the solution \( y \) via the singular perturbation formulation approaches the solution of \( y \) via the exact model. As \( \epsilon \) is chosen smaller, the system becomes stiffer and more floating point operations will be required to solve the system. As we can see, the solution does not converge much more rapidly for an \( \epsilon < 0.004 \), and that is the reason we chose \( \epsilon \) to be 0.004.

<table>
<thead>
<tr>
<th>( \epsilon )</th>
<th>flops</th>
<th>steps (98%)</th>
<th>( t ) (sec) (98%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>68978</td>
<td>14</td>
<td>0.224</td>
</tr>
<tr>
<td>0.04</td>
<td>71248</td>
<td>11</td>
<td>0.176</td>
</tr>
<tr>
<td>0.03</td>
<td>71249</td>
<td>8</td>
<td>0.128</td>
</tr>
<tr>
<td>0.02</td>
<td>72887</td>
<td>5</td>
<td>0.080</td>
</tr>
<tr>
<td>0.01</td>
<td>76166</td>
<td>3</td>
<td>0.048</td>
</tr>
<tr>
<td>0.007</td>
<td>76543</td>
<td>2</td>
<td>0.032</td>
</tr>
<tr>
<td>0.005</td>
<td>80452</td>
<td>2</td>
<td>0.032</td>
</tr>
<tr>
<td>0.004</td>
<td>82091</td>
<td>1</td>
<td>0.016</td>
</tr>
<tr>
<td>0.003</td>
<td>82117</td>
<td>1</td>
<td>0.016</td>
</tr>
</tbody>
</table>

**Table 6.1:** Experimentation varying \( \epsilon \) to get number floating point operations and time of error reduction by 98%.
We went further and also studied the behavior of the error as $\epsilon$ was reduced dramatically. To calculate the error ($P$) we used the following normalized equation

$$P = \frac{\int_0^1 (e)^T(e) dt}{\int_0^1 z^T_z z_r dt}$$  \hspace{1cm} (6.32)$$

where $e = y - y_r$, $y$ is the solution vector using the SPF, and $y_r$ is the data computed from real data measured.

![Graph showing $P$ vs $\epsilon$](image)

**Figure 6.2:** Plot of the error vs $\epsilon$

In Figure 6.2 we can observe how in general the error gets smaller as $\epsilon$ decreases. In Figure 6.3 we can observe that the error due to numerical computations is taking importance as $\epsilon$ is decreased even further. As we already discussed observing Table
6.1 we know that the number of floating point operations increases as $\epsilon$ decreases. Our choice of $\epsilon$ was based on the time it took for the initial error to be reduced 98%. We can conclude that there are three different factors to be taken into account when choosing a value for $\epsilon$: the speed of the offset error to converge, the numerical computations fee (floating point operations) and the error due to the numerical work.

We also made a comparison of the error along a trajectory of the SPF with respect to real data, the error of the exact model with respect to real data, and the error of the SPF with respect to the exact model. To do this analysis we fixed $\epsilon$ at 0.004 and calculated the solution via the SPF using an exact initial condition for $y$. The
error was calculated using equation 6.32. For the error of the SPF with respect to real data, we obtained a value of $2.8 \times 10^{-3}$, for the error of the exact model with respect to real data, we obtained a value of $2.6 \times 10^{-3}$, and for the error of the SPF with respect to the exact model we obtained a valued of $2.4 \times 10^{-4}$. We can see that the error between the real data and the SPF and the error between the exact model and the real data is one order of magnitude greater than the error between the exact model and the SPF. This tells us that the error arising from the SPF is small relative to the error due to uncertainties and unmodeled effects.

In Figure 6.4 we present graphs of $x$ and $y$ versus time making a comparison between the real data and the singular perturbation approach using a value of $\epsilon = 0.004$. The error was also plotted to get a better idea of the precision of the solution. It is important to notice that the initial condition of $x$ is the condition that is used to determine the physical position of the pendulum. The initial condition given for $y$ was arbitrarily chosen at $y(0) = -0.0450$. In figure 6.5 we present graphs of $x$ and $y$ versus time making a comparison between the exact model and the singular perturbation approach using a value of $\epsilon = 0.004$. Here again the initial condition given for $y$ was arbitrarily chosen at $y(0) = -0.0450$. Finally in figure 6.6 we present graphs of $x$ and $y$ versus time making a comparison between the exact model and the real data.

In Figure 6.4 we can see how the exact model has a small error in the order of $1 \times 10^{-3}$. This error is due to other physical factors as Coulomb friction and hysteresis. Figure 6.6 makes a comparison between the exact model and the SPF since we are interested in the results of the latter. The plots of the SPF show how the initial condition of $y$ should have been $y(0) = 0.0416$. We can see how the solution of $y$ through the singular perturbation approach converges rapidly to the exact solution. The solution of $y$ is obtained from the constraint equation as represented by equation
(6.11). If we would have applied an initial condition for \( y \) identical to the condition imposed by the constraints, that is \( y(0) = -0.0416 \), we would not have the large initial error that appears in Figures 6.4 and 6.5. This behavior appears because the SPF formulation forces the system to converge to the solution since \( \dot{w} = (1/\epsilon)\phi \). We have found an efficient way of solving DAEs using only ODE. This approach presents the advantage of not necessarily having an exact initial condition for the variable defined by the constraint equations. Since we have a system of ODE we have the wide variety of solvers available including solvers for stiff systems.
Figure 6.4: Plots comparing the real data and the Singular Perturbation Formulation
Figure 6.5: Plots comparing the exact solution and the Singular Perturbation Formulation
Figure 6.6: Plots comparing the exact solution and the real data
Chapter 7

Application of the SPF to the Rice Planar Delta Robot (RPDR)

In this chapter first we will derive the equations of motion of the Rice Planar Delta Robot (RPDR), we reduced the model as was done in [3] and explained in chapter 3 and then apply the singular perturbation approach developed in chapter 6 to obtain an approximation of the solution of its equations of motion.

The RPDR was designed and built at Rice University to perform experiments on closed chain mechanisms. A sketch of the RPDR appears in figure 7.1. The RPDR has two links connected through revolute joints. Two of the links (links 1 and 2) are actuated with DC motors while the other two motors (links 3 and 4) are passive. The robot has 2-DOF, so the inputs match the number of dof.

7.1 Derivation of the DAEs for the RPDR

The RPDR can be seen as two serial robots connected at the end by a free joint as seen in figure 7.2. Each one of the serial robots is completely defined by a set of two second order ordinary differential equations (o.d.e.), which makes a total of four o.d.e. If we add the constraint equations, we will have two more algebraic equations that are added to the system dynamic definition. The equations of motion will be in the form as shown in [12]:

\[
D'(q')\ddot{q}' + C'(q', \dot{q}')\dot{q}' + g'(q') = \phi_q^T(q')\lambda \tag{7.1}
\]

\[
\phi_{q'}(q')\dot{q}' = 0 \tag{7.2}
\]
In which $q' = [q_1 q_2 q_3 q_4]^T$ is the generalized coordinate vector of the free system (the vector of coordinates representing the angles in each one of the four links of the RPDR), $D'(q')$ is the 4x4 inertia matrix, $C'(q', \dot{q}')$ is the 4x4 matrix representing the centrifugal and Coriolis terms, $g'(q')$ is the gravity vector, and $\phi_q(q')$ is the jacobian of $\phi(q')$ as it was pointed out in [3].

The detailed derivation of these equations, using the Lagrange-Euler method is as follows:

As defined in Figure 7.2, $m_i$, $l_i$, and $a_i$ are respectively the mass, distance to the center of mass, and length of link $i$. The inertia of link $i$ about the line through the center of mass parallel to the axis of rotation is denoted by $I_i$. The parameters corresponding to link 1 and link 3 are defined similar to the parameters of link 2 and link 4.
The position of the center of gravity of links 1 and 3 is:

\[
\mathbf{l}_{c1} = [l_{c1} \cos(q_1)]\mathbf{i} + [l_{c1} \sin(q_1)]\mathbf{j} \tag{7.3}
\]

\[
\mathbf{l}_{c3} = [a_1 \cos(q_1) + l_{c3} \cos(q_1 + q_3)]\mathbf{i} + [a_1 \sin(q_1) + l_{c3} \sin(q_1 + q_3)]\mathbf{j} \tag{7.4}
\]

The velocity is:

\[
\mathbf{v}_1 = [-l_{c1} \sin(q_1)\dot{q}_1]\mathbf{i} + [l_{c1} \cos(q_1)\dot{q}_1]\mathbf{j} \tag{7.5}
\]

\[
\mathbf{v}_3 = [-a_1 \sin(q_1)\dot{q}_1 - l_{c3} \sin(q_1 + q_3)\dot{q}_1 - l_{c3} \sin(q_1 + q_3)\dot{q}_3]\mathbf{i} +
[a_1 \cos(q_1)\dot{q}_1 + l_{c3} \cos(q_1 + q_3)\dot{q}_1 + l_{c3} \cos(q_1 + q_3)\dot{q}_3]\mathbf{j} \tag{7.6}
\]

The velocity square is:

\[
v_1^2 = l_{c1}^2 \dot{q}_1^2 \tag{7.7}
\]
\[ v_3^2 = [a_1^2 + 2a_1l_{c3}\cos(q_3) + l_{c3}^2][q_1]^2 + [2a_1l_{c3}\cos(q_3) + 2l_{c3}^2][q_1]^2[q_3]^2 + [l_{c3}^2][q_3]^2 \quad (7.8) \]

The Kinetic Energy is: \( K = K_1 + K_3 \)

\[ K_1 = \frac{1}{2}m_1v_1^2 + \frac{1}{2}I_1q_1^2 \quad (7.9) \]
\[ K_3 = \frac{1}{2}m_3v_3^2 + \frac{1}{2}I_3(q_1 + q_3)^2 \quad (7.10) \]
\[ K = \frac{1}{2} [m_1l_{c1}^2 + m_3(a_1^2 + l_{c3}^2 + 2a_1l_{c3}\cos(q_3)) + I_1 + I_3][q_1]^2 + \]
\[ [m_3(a_1l_{c3}\cos(q_3) + 2l_{c3}^2) + I_3][q_1][q_3] + \frac{1}{2}[m_3l_{c3}^2 + I_3][q_3]^2 \quad (7.11) \]

The Potential Energy is:

\[ V_1 = m_1gl_{c1}\sin(q_1) \quad (7.12) \]
\[ V_3 = m_3g(a_1\sin(q_1) + l_{c3}\sin(q_1 + q_3)) \quad (7.13) \]
\[ V = m_1gl_{c1}\sin(q_1) + m_3g(a_1\sin(q_1) + l_{c3}\sin(q_1 + q_3)) \quad (7.14) \]

The derivatives entering into Lagrange’s equations are as follows:

\[ \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_1} \right) = [m_1l_{c1}^2 + m_3(a_1^2 + l_{c3}^2 + 2a_1l_{c3}\cos(q_3)) + I_1 + I_3][\dot{q}_1] - \]
\[ [2m_3a_1l_{c3}\sin(q_3)][\dot{q}_1][\dot{q}_3] + [m_3(a_1l_{c3}\cos(q_3) + l_{c3}^2) + I_3][\dot{q}_3] - \]
\[ [m_3a_1l_{c3}\sin(q_3)][\dot{q}_3]^2 \quad (7.15) \]
\[ \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_3} \right) = [m_3(a_1l_{c3}\cos(q_3) + l_{c3}^2) + I_3][\dot{q}_1] - [m_3a_1l_{c3}\sin(q_3)][\dot{q}_3][\dot{q}_1] + \]
\[ [m_3l_{c3}^2 + I_3][\dot{q}_3] \quad (7.16) \]
\[ \frac{\partial T}{\partial q_1} = 0 \quad (7.17) \]
\[ \frac{\partial T}{\partial q_3} = -m_3a_1l_{c3}\sin(q_3)[\dot{q}_1]^2 - m_3a_1l_{c3}\sin(q_3)[\dot{q}_1][\dot{q}_3] \quad (7.18) \]
\[ \frac{\partial V}{\partial q_1} = m_1gl_{c1}\cos(q_1) + m_3ga_1\cos(q_1) + m_3gl_{c3}\cos(q_1 + q_3) \quad (7.19) \]
\[ \frac{\partial V}{\partial q_3} = m_3gl_{c3}\cos(q_1 + q_3) \quad (7.20) \]
The same derivation that has been presented for the serial robot of links 1 and 3 is done for the serial robot conformed of links 2 and 4. If we substitute into the Lagrange's equations and arrange the terms we get the same result as [3]:

\[
D'(q') = \begin{bmatrix}
  d_{1,1} & 0 & d_{1,3} & 0 \\
  0 & d_{2,2} & 0 & d_{2,4} \\
  d_{3,1} & 0 & d_{3,3} & 0 \\
  0 & d_{4,2} & 0 & d_{4,4}
\end{bmatrix}
\] (7.21)

where, \( d_{1,1} = m_1l_{c1}^2 + m_3(a_1^2 + l_{c3}^2 + 2a_1l_{c3} \cos(q_3)) + I_1 + I_3 \), \( d_{1,3} = m_3(l_{c3}^2 + a_1l_{c3} \cos(q_3)) + I_3 \), \( d_{2,2} = m_2l_{c2}^2 + m_4(a_2^2 + l_{c4}^2 + 2a_2l_{c4} \cos(q_4)) + I_2 + I_4 \), \( d_{2,4} = m_4(l_{c4}^2 + a_2l_{c4} \cos(q_4)) + I_4 \),

\[
C'(q') = \begin{bmatrix}
  h_1\dot{q}_3 & 0 & h_1(\dot{q}_1 + \dot{q}_3) & 0 \\
  0 & h_2\dot{q}_4 & 0 & h_2(\dot{q}_2 + \dot{q}_4) \\
  -h_1\dot{q}_1 & 0 & 0 & 0 \\
  0 & -h_2\dot{q}_2 & 0 & 0
\end{bmatrix}
\] (7.22)

where \( h_1 = -m_3a_1l_{c3} \sin(q_3) \), and \( h_2 = -m_4a_2l_{c4} \sin(q_4) \), and

\[
g'(q') = \begin{bmatrix}
  (m_1l_{c1} + m_3a_1) \cos(q_1) + m_3l_{c3} \cos(q_1 + q_3) \\
  (m_2l_{c2} + m_4a_2) \cos(q_2) + m_4l_{c4} \cos(q_2 + q_4) \\
  m_3l_{c3} \cos(q_1 + q_3) \\
  m_4l_{c4} \cos(q_2 + q_4)
\end{bmatrix}g
\] (7.23)

where \( g = 9.81 \text{ m/sec}^2 \) is the gravitational acceleration constant.

Until now we have derived and arranged in matrix form the 4 o.d.e. that describe the unconstrained system. The next step is to write the algebraic constraint equations. The constraint equations come from the fact that point \( E \) and point \( F \) in Figure 7.2 are coincident. We do it by inspection and obtain:

\[
\phi(q) = \begin{bmatrix}
  a_1 \cos(q_1) + a_3 \cos(q_1 + q_3) - c - a_2 \cos(q_2) - a_4 \cos(q_2 + q_4) \\
  a_1 \sin(q_1) + a_3 \sin(q_1 + q_3) - a_2 \sin(q_2) - a_4 \sin(q_2 + q_4)
\end{bmatrix}
\] (7.24)

Since we have more generalized coordinates than degrees of freedom, we can obtain by classical dynamics a representation of the constrained system in terms of the
Lagrange multipliers [5]. The standard form of the Lagrange's equations including the Lagrange multipliers is:

\[
\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} = Q_j + \sum_{i=1}^{M-N} \lambda_i a_{ij}, \quad j = 1, 2, ..., M
\]

where,

\[ a_{ij} = \phi_q = \frac{\partial \phi_j}{\partial q_i} \]

and with \( \lambda_i \) being the Lagrange multipliers.

Evaluating \( \phi_q \), we get:

\[
\phi_q^T(q) = \begin{bmatrix}
-a_1 \sin(q_1) - a_3 \sin(q_1 + q_3) & a_1 \cos(q_1) + a_3 \cos(q_1 + q_3) \\
 a_2 \sin(q_2) + a_4 \sin(q_2 + q_4) & -a_2 \cos(q_2) - a_4 \cos(q_2 + q_4) \\
-a_3 \sin(q_1 + q_3) & a_3 \cos(q_1 + q_3) \\
a_4 \sin(q_2 + q_4) & -a_4 \cos(q_2 + q_4)
\end{bmatrix}
\] (7.25)

Now we have all the terms in equations (7.1) and (7.2). With this example we have modeled a closed kinematic chain using DAEs. It is clear that more generalized coordinates than degrees of freedom of the system had to be used. In the derivation of the equations of motion for the RPDR, a velocity constrain was used, which in return reduce the differential index of the system.

### 7.2 The Reduced Model

Our next step is to reduce the system as it was explained in [3] and repeated in chapter 3. We will use the reduced model to solve the system via the SPF. Furthermore, we will make a comparison of the solution of the equations developed in [3] which is an exact model, and the solution via the singular perturbation formulation explained in chapter 6.

We choose the generalized coordinate vector of the constrained system to be \( q = [ q_1 \quad q_2 ]^T \) because these are the actuated variables. We begin with the pa-
rameterization \( \alpha(q') = q \), which is given by,

\[
\alpha(q') = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} q' = q.
\] (7.26)

Now by combining the equations of constraints, (7.24), and the parameterization \( \alpha(q') = q \), (7.26), we have

\[
\psi(q') = \begin{bmatrix} \phi(q') \\ \alpha(q') \end{bmatrix}
\] (7.27)

as it was defined in [3].

Differentiating with respect to \( q' \), we obtain the following expression for \( \psi_{q'}(q') \):

\[
\psi_{q'}(q') = \begin{bmatrix} \psi_{q'}(1, 1) & \psi_{q'}(1, 2) & -a_3\sin(q_1 + q_3) & a_4\sin(q_2 + q_4) \\ \psi_{q'}(2, 1) & \psi_{q'}(2, 2) & a_3\cos(q_1 + q_3) & -a_4\cos(q_2 + q_4) \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},
\] (7.28)

where, \( \psi_{q'}(1, 1) = -a_1\sin(q_1) - a_3\sin(q_1 + q_3), \psi_{q'}(1, 2) = a_2\sin(q_2) + a_4\sin(q_2 + q_4), \psi_{q'}(2, 1) = a_1\cos(q_1) + a_3\cos(q_1 + q_3), \) and \( \psi_{q'}(2, 2) = -a_2\cos(q_2) - a_4\cos(q_2 + q_4). \)

Repeating from [3] and chapter 3 we have the following expression:

\[
\rho(q') = \psi_{q'}^{-1}(q') \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.
\] (7.29)

and

\[
\dot{\rho}(q', \dot{q}') = -\psi_{q'}^{-1}(q')\dot{\psi}_{q'}(q', \dot{q}')\rho(q'),
\] (7.30)

where \( \dot{\psi}_{q'}(q', \dot{q}') \) can be obtained by differentiating (7.28) with respect to time.

The reduced system is repeated here for conivienience:

\[
D(q')\ddot{q} + C(q', q')\dot{q} + g(q') = 0,
\] (7.31)

where

\[
D(q') = \rho(q')^T D'(q')\rho(q'),
\]
\[ C(q', \dot{q}') = \rho(q')^T C'(q', \dot{q}') \rho(q') + \rho(q')^T D'(q') \dot{\rho}(q', \dot{q}'), \]
\[ g(q') = \rho(q')^T g'(q'), \]
\[ D'(q'), C'(q', \dot{q}'), g'(q'), \dot{\rho}(q', \dot{q}'), \text{ and } \rho(q') \text{ for the RPDR are defined in (7.21), (7.22) (7.23), (7.30), (7.29).} \]

In [3] and [13], equation (7.31) was used in conjunction with the parameterization
\[ \dot{q}' = \rho(q') \dot{q}, \]
\[ q' = \sigma(q) \]
to solve the system. In general, it is not possible to derive an analytic expression for \( \sigma(q) \) in closed kinematic chains as it was explained in chapter 2. Usually it must be computed using numerical methods. For the R.P.D.R. however, it is derived in [3] an explicit expression solving the position constraints (7.24) to obtain the following:
\[ q_4 = \tan^{-1} \left[ \frac{B(q_1, q_2)}{A(q_1, q_2)} \right] + \tan^{-1} \left[ \frac{\pm \sqrt{A(q_1, q_2)^2 + B(q_1, q_2)^2 - C(q_1, q_2)^2}}{C(q_1, q_2)} \right] - q_2, \]  
(7.32)
where \( A(q_1, q_2) = 2a_4 \lambda(q_1, q_2), B(q_1, q_2) = 2a_4 \mu(q_1, q_2), \) and \( C(q_1, q_2) = a_3^2 - a_4^2 - \lambda(q_1, q_2)^2 - \mu(q_1, q_2)^2, \) and \( \lambda(q_1, q_2) = a_2 \cos(q_2) - a_1 \cos(q_1) + c, \mu(q_1, q_2) = a_2 \sin(q_2) - a_1 \sin(q_1). \) Finally,
\[ q_3 = \tan^{-1} \left[ \frac{\mu(q_1, q_2) + a_4 \sin(q_2 + q_4)}{\lambda(q_1, q_2) + a_4 \cos(q_2 + q_4)} \right] - q_1. \]  
(7.33)
Hence, (7.32) and (7.33) combined represent the parameterization \( q' = \sigma(q). \)

### 7.3 The Singular Perturbation Formulation

To obtain a solution via the singular perturbation approach we need to solve the system (7.31) together with equation (6.17), which is the model of the singular perturbation formulation developed in chapter 6 and we will repeat here for convenience:
\[ \phi_z \dot{z} = -\frac{1}{\epsilon} \phi - \phi_q \dot{q} \]
In order to solve this system, we need to compute $\phi_q$ and $\phi_z$. We will use the notation introduced in chapter 6 which means more specifically for the RPDR $q = [q_1 \quad q_2]^T$ and $z = [q_3 \quad q_4]^T$. Hence,

$$
\phi_q(q, z) = \begin{bmatrix}
\psi_q(1, 1) & \psi_q(1, 2) \\
\psi_q(2, 1) & \psi_q(2, 2)
\end{bmatrix},
$$

(7.34)

where, $\psi_q(1, 1) = -a_1 \sin(q_1) - a_3 \sin(q_1 + q_3)$, $\psi_q(1, 2) = a_2 \sin(q_2) + a_4 \sin(q_2 + q_4)$, $\psi_q(2, 1) = a_1 \cos(q_1) + a_3 \cos(q_1 + q_3)$, and $\psi_q(2, 2) = -a_2 \cos(q_2) - a_4 \cos(q_2 + q_4)$. And

$$
\phi_z(q, z) = \begin{bmatrix}
-a_3 \sin(q_1 + q_3) & a_4 \sin(q_2 + q_4) \\
a_3 \cos(q_1 + q_3) & -a_4 \cos(q_2 + q_4)
\end{bmatrix},
$$

(7.35)

Notice that $\phi_q$ and $\phi_z$ are sub-matrices of $\psi_q$.

Now we are ready to solve the system via the singular perturbation approach which consists in solving

$$
D(q, z) \ddot{q} = -C(q, \dot{q}, z, \dot{z}) \dot{q} - g(q, z)
$$

(7.36)

$$
\phi_z \ddot{z} = -\frac{1}{\epsilon} \phi - \phi_q \dot{q}
$$

(7.37)

Notice that we do not compute $\ddot{z}$, only $\dot{z}$. Also notice that this procedure results in an explicit model and the parameterization $q' = \sigma(q)$ was not necessary to compute.

### 7.4 Simulation: Implementation and Results

We designed a physical experiment in which we gave the system an initial position and zero velocity and let the system swing freely. We compared the experimental results with the exact solution of the reduced model derived in [3] and [13], and also compared experimental results with the solution using the SPF. The real data was traced using a program and an optical encoder connected to a PC computer through a DSP board. We ran a series of 10 experiments to obtain the real data. In each experiment we arbitrarily fixed an initial condition using a braking system.
with angles close to their equilibrium position as seen in Figure 7.3. After starting the tracing program we released the brake. With the files containing the traced data, we made some manual manipulation on each file eliminating the dead time and the data traced after the pendulum had stopped. The derivatives of the angles $\theta_i$ were computed numerically. To plot the results we chose 1 of the 10 experiments.

![Figure 7.3: RPDR before releasing to trace the experiment data](image)

The parameters we used in the simulation are summarized in table 7.1. These parameters are very similar to the ones presented in [3], but we had to adjust the value of the constants for links 1 and 2 to contemplate the kinematics of the braking system and the sprocket attached to these links. The distances between axis of joints 1 and 2 was measured to be $c = 0.3048$. We also added a viscous damping term ($C_o$) to account for friction of this type. To calculate the damping constant we developed a program in which the solution of our differential system was fit to the real measured
<table>
<thead>
<tr>
<th>Link $i$</th>
<th>$m_i$ (kg)</th>
<th>$a_i$ (m)</th>
<th>$l_{ci}$ (m)</th>
<th>$I_i$ (kg m$^2$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.7951</td>
<td>0.2794</td>
<td>0.0452</td>
<td>$7.459 \times 10^{-3}$</td>
</tr>
<tr>
<td>2</td>
<td>0.7862</td>
<td>0.2794</td>
<td>0.0452</td>
<td>$7.459 \times 10^{-3}$</td>
</tr>
<tr>
<td>3</td>
<td>0.2611</td>
<td>0.3048</td>
<td>0.1581</td>
<td>$3.476 \times 10^{-3}$</td>
</tr>
<tr>
<td>4</td>
<td>0.2611</td>
<td>0.3048</td>
<td>0.1467</td>
<td>$3.476 \times 10^{-3}$</td>
</tr>
</tbody>
</table>

**Table 7.1:** Kinematic Link Parameters of the Rice Planar Delta Robot

Data using a least square fit as it was done for the robotic arm example in chapter 6. For all numerical computations we used Matlab [32]. The value we obtained for this constant is $C_o = 0.002$ N-m-s/rad. For our real data graphs the initial position of the chain was chosen to be a small angle close to its equilibrium position. The initial configuration we chose is given by $q_1 = -56$ and $q_2 = -73$ with zero velocity.

To choose $\epsilon$ we ran a series of experiments. Table 7.2 shows different values of $\epsilon$, the number of floating point operations in the solution of the SPF, the number of steps required for the error of $z_1$ and $z_2$ via the SPF to be reduced by 98%. To get these results we had to choose an initial condition for $z_1$ and $z_2$ to be off the condition imposed by the constraints. The value of $\epsilon$ determines how fast the solution of $z_1$ and $z_2$ via the singular perturbation formulation converges to the solution via the exact model. As $\epsilon$ is chosen smaller, the system becomes stiffer and more floating point operations will be required to solve the system. As we can see, the solution does not converge much more rapidly for an $\epsilon < 0.004$, and that is the reason we chose $\epsilon$ to be 0.004.

The plots of each link comparing the SPF and the real data and the error are presented in Figures 7.5 and 7.6. Figure 7.5 (a) and (c) show $q_1$ and $q_2$ respectively. In each graph the dotted line represents the SPF solution and the solid line represents the real data. The difference between the real data and the SPF is plotted in Figures 7.5 (b) and (d). We can observe a simulation that does not vary at any point for
<table>
<thead>
<tr>
<th>$\epsilon$</th>
<th>flops</th>
<th>$z_1$ steps (98%)</th>
<th>$z_1$ t (sec) (98%)</th>
<th>$z_2$ steps (98%)</th>
<th>$z_2$ t (sec) (98%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>226920</td>
<td>3</td>
<td>0.048</td>
<td>11</td>
<td>0.176</td>
</tr>
<tr>
<td>0.04</td>
<td>235647</td>
<td>3</td>
<td>0.048</td>
<td>9</td>
<td>0.144</td>
</tr>
<tr>
<td>0.03</td>
<td>304945</td>
<td>3</td>
<td>0.048</td>
<td>7</td>
<td>0.112</td>
</tr>
<tr>
<td>0.02</td>
<td>357242</td>
<td>3</td>
<td>0.048</td>
<td>5</td>
<td>0.080</td>
</tr>
<tr>
<td>0.01</td>
<td>504155</td>
<td>2</td>
<td>0.032</td>
<td>3</td>
<td>0.048</td>
</tr>
<tr>
<td>0.005</td>
<td>756102</td>
<td>1</td>
<td>0.016</td>
<td>2</td>
<td>0.032</td>
</tr>
<tr>
<td>0.004</td>
<td>886235</td>
<td>1</td>
<td>0.016</td>
<td>1</td>
<td>0.016</td>
</tr>
<tr>
<td>0.003</td>
<td>1068175</td>
<td>1</td>
<td>0.016</td>
<td>1</td>
<td>0.016</td>
</tr>
</tbody>
</table>

**Table 7.2:** Experimentation varying $\epsilon$ to get number floating point operations and time for reduction of error by 98%.

more than 2 degrees. Figure 7.6 (a) and (c) show $z_1$ and $z_2$ respectively. Notice that $z_1$ and $z_2$ represent the position of links 3 and 4 respectively. In each graph the dotted line represents the SPF solution and the solid line represents the real data. The difference between the real data and the SPF is plotted in Figures 7.6 (b) and (d). We have on purpose given an arbitrary initial condition: $z_1 = 0$ and $z_2 = 0$ to show how the SPF adjust to the real solution in less 0.02 sec. We can also observe this in the error graphs so the error at time 0 is $Ez_1(0) = 19$ and $Ez_2(0) = 48$. The plots of each link comparing the SPF and the exact solution of the differential equations of the system (from [3]) and the error are presented in Figures 7.7 and 7.8. Figure 7.7 (a) and (c) show $q_1$ and $q_2$ respectively. In each graph the dotted line represents the SPF solution and the solid line represents the exact solution. The difference between the exact solution of the equations of motion of the RPDR and the SPF is plotted in Figures 7.7 (b) and (d). We can observe that the SPF is practically the same plot as the exact solution. Figure 7.8 (a) and (c) show $z_1$ and $z_2$ respectively. In each graph the dotted line represents the SPF solution and the solid line represents the exact solution. The difference between the exact solution and the SPF is plotted in Figures 7.8 (b) and (d). Here, the error was also plotted to get an idea of the precision. We
can observe from these figures how the solution given by the SPF converges to the exact solution. We present plots of the exact solution and the real data in Figures 7.9 and 7.10. Finally, in Figures 7.11 and 7.12 we plotted again the SPF and the real data, but this time using the exact real initial condition for $z_1$ and $z_2$. With this plots we can observe the error of the SPF and compare it with the error of the exact model.

In Figures 7.9 and 7.10 we can see how the exact model has a small error of less than 3 degrees. This error is due to other physical factors like coulomb friction and hysteresis. Figures 7.7 and 7.8 make a comparison between the exact model and the SPF since we are interested in the results of the latter. The plots of the SPF show how the initial condition of $z_1$ and $z_2$ are off the initial condition imposed by the constraints and how the solution converges rapidly to the exact solution. If we would have applied an initial condition for $z_1$ and $z_2$ identical to the condition imposed by the constraints, that is $z_1(0) = 19$ and $z_2(0) = 48$ respectively, we would not have the large initial error that appears in Figures 7.6 and 7.8. This behavior appears because the SPF formulation forces the system to converge to the solution since $\dot{w} = -(1/\epsilon) \phi$. We have found an efficient way of solving DAEs using only ODE. This approach present the advantage of not necessarily having an exact initial condition for the variable defined by the constraint equations. Besides the original constraint is part of the model, so the solution does not drift away from the constraints due to differentiations.

We went further and also studied the behavior of the error as $\epsilon$ was reduced. To calculate the error ($P$) we used the following normalized equation, which is equation (6.32) used in Chapter 6,

$$P = \frac{\int_0^1 (e)^T (e) \, dt}{\int_0^1 z_r^T z_r \, dt}$$  \hspace{1cm} (7.38)
where $e = z - z_r$, $z$ is the solution vector using the exact model or the SPF, and $z_r$ is the data computed from real data measured.

![Plot of the error vs $\epsilon$](image)

**Figure 7.4:** Plot of the error vs $\epsilon$

In Figure 7.4 we can see how the error decreases as we make $\epsilon$ smaller. As we already observed in table 7.2, the number of floating point operations increases as $\epsilon$ decreases. There is a numerical fee for making the error of the initial condition to approach faster the exact solution.

We also made an analysis of the error introduced by the SPF with initial conditions that comply with the constraints in comparison with the error introduced by the exact model. We obtained the error $P$ along the trajectory using equation (7.38). For the
error of the SPF with respect to real data we obtained a value of 1.9793, for the error of the exact model with respect to real data an error value of 1.9838, and for the error of the SPF with respect to the exact model an error value of 0.0299. Since the error of the solution through the SPF is in magnitude smaller than the error of the solution through the exact model, we can conclude that the SPF formulation does not introduce any additional error when the initial condition given coincides with the condition imposed by the constraints. The error on both models is due to factors not considered such as hysteresis or Coulomb friction as already mentioned or uncertainties in the parameters.
Figure 7.5: Plots of $q_1$ and $q_2$ comparing the real data of the RPDR and the Singularity Perturbed Formulation
Figure 7.6: Plots of $z_1$ and $z_2$ comparing the real data of the RPDR and the Singularity Perturbed Formulation
Figure 7.7: Plots of $q_1$ and $q_2$ comparing the exact solution of the equations of motion of the RPDR and the Singularly Perturbed Formulation
Figure 7.8: Plots of $z_1$ and $z_2$ comparing the exact solution of the equations of motion of the RPDR and the Singularly Perturbed Formulation
Figure 7.9: Plots of $q_1$ and $q_2$ comparing the exact solution of the equations of motion of the RPDR real data.
Figure 7.10: Plots of $z_1$ and $z_2$ comparing the exact solution of the equations of motion of the RPDR and the real data
Figure 7.11: Plots of $q_1$ and $q_2$ comparing the real data of the RPDR and the Singularity Perturbed Formulation using an exact initial condition.
Figure 7.12: Plots of $z_1$ and $z_2$ comparing the real data of the RPDR and the Singularity Perturbed Formulation using an exact initial condition
Chapter 8

Conclusions and Future Work

In this chapter we summarize the contributions of this thesis and discuss future extensions of this research.

First, we studied the kinematics of kinematic chains and we were able to show why closed kinematic chains cannot in general be modeled using independent generalized coordinates. We also studied the closed relation that mechanical systems and in particular closed kinematic chains have with differential algebraic equations. A model was established to understand the way the equations of motion were obtained. Through examples we were able to compare equations of motion using independent generalized coordinates, and equations of motion using constrained generalized coordinates. Besides, a reduced model for closed kinematic chains was derived and later used to solve the equations of motion.

We were able to characterize and define differential algebraic equations which is currently in the literature, a topic of extensive research being developed extensively. We have seen that DAEs describe physical systems and that their solution present certain differences compared to the solution of ODEs. In particular, DAEs imply the presence of underlying algebraic constraints which are differentiated to obtain a solution using standard methods. The differentiation drives-off the original constraints due to the error introduced by numerical computations. Many times in mechanical systems this implies that physical position constraints are not fulfilled. Furthermore, standard DAE solvers require an accurate initial condition which satisfies the constraints to get a solution that does not present impulsive behaviors.
Finally we derived a singular perturbation formulation to give an approximation of the equations of motion using constrained generalized coordinates. This formulation presents many advantages, and is highly numerically efficient. The formulation converges to the exact solution even though small error in the initial conditions are introduced. To compare and verify the results of this thesis, an experimental exercise was done using the Rice Planar Delta Robot (RPDR). The reduced model previously defined was used to model the RPDR and we first solved the equations of motion using an exact model and later the singular perturbation approach. We made an analysis of the small parameter $\epsilon$ used in the SPF. From this analysis we learned that $\epsilon$ defines how rapidly the constraint is met when an initial error in the constraint is introduced. As $\epsilon$ is decreased the error on the solution approaches the exact solution more rapidly as long as the error due to numerical computations is not significant. We also learned that the number of floating point operation increases as $\epsilon$ is decreased. We observed that the error due to the SPF is much smaller that the error due to the physical factors not considered in the analysis.

Future extension of this research is recommended in the area of control to observe how efficient the approximation behaves. Also, research on the compliance of the original constraint can be done. Differential algebraic equations are a constant challenge when we talk about directly implementing control on them. Other work can be done in the area of error analysis to determine how close SPF are to the real model and in comparison with other methodologies to solve DAEs. In the same manner a study of the numerical efficiency of this technique could be broadened.
Bibliography


