Numerical Solution of Differential Algebraic Equations and Applications.

ed. Per Grove Thomsen

November 21, 2005

Contents

Chapter 1

Preface

These lecture notes have been written as part of a special course on the numerical solution of Differential Algebraic Equations and applications . The course was held at IMM in the spring of 2005. The authors of the different chapters have all taken part in the course and the chapters are written as part of their contribution to the course.
It is hoped that coming courses in the Numerical Solution of DAE's will benefit from these lecture notes.
The students participating in the course and contributing with their chapters are the following:

- Marie Bro
- Michaël Gineste
- Mark Hoffmann
- Rasmus Frank Kristensen
- Kristina Hoffmann Larsen
- Peter Larsen
- Carsten Vølcker

Chapter 2

Introduction

The theory of numerical solution of ordinary differential equations (ODE's) has been developed since the early part of this century beginning with Adams and Runge and Kutta. At the present time the theory is well understood and the development of software has reached a state where robust methods are available for a large variety of problems. The theory for Differential Algebraic Equations (DAE's) has not been stydied to the same extent , early attempts have been made by Gear and Petzold in the early 1970'es. Not only are the problems harder to solve but the theory is also harder to understand.
The problems that lead to DAE's are found in many applications of which some are mentioned in later chapters of these lecture notes. The choice of sources for problems has been selected by the students following the course. It is the intention to illustrate the application of DAE-theory in practical environments spanning a wide range of engineering fields rather than covering a specific field more densely.
For a detailed study of the methods that may be applied to solve the different example problems we refer to the references [1, 18, 20]

2.1 Definitions

The problems considered are in the most general form a fully implicit equation of the form

$$F(y', y) = 0 \tag{2.1.1}$$

F and y are of dimension n and F is assumed to be sufficiently smooth. This is the autonomous form , a non-autonomous case is defined by

$$F(y', y, x) = 0 \tag{2.1.2}$$

Since the non-autonomous equation may be made autonomous by adding the equation $x' = 1$ we need not consider the non-autonomous form seperately. (this may be subject to debate since the non-autonomous case can have special features)
A special case arises when we can solve for the y'-variable since we can make the equation explicit and obtain a system of ODE's, the condition for this is that $\frac{\partial F}{\partial y'}$ is nonsingular. The case when this process does not work is the case we wish to consider in these notes it is the case when $\frac{\partial F}{\partial y'}$ is singular. We talk about systems of Differential Algebraic Equations or DAE's for short.

2.1.1 Semi explicit DAE's

The simplest form of problem is the one where we can write the system in the form

$$\begin{aligned} y' &= f(y,z) \\ 0 &= g(y,z) \end{aligned}$$

and g_z has a bounded inverse in a neighbourhood of the solution.
Assuming we have a set of consistent initial values (y_o, z_o) it may be shown that in this case z can be found as a function of y. This implies local existence, uniqueness and regularity of the solution.

2.1.2 Index

Numerous examples exist where the conditions above do not hold. These cases have general interest and below we give a couple of examples from applications.

- Example 1.1: Singular algebraic constraint.
  We consider the problem defined by the system of three equations

$$y_1' = y_3 \tag{2.1.3}$$

$$0 = y_2(1 - y_2) \tag{2.1.4}$$

$$0 = y_1 y_2 + y_3(1 - y_2) - t \tag{2.1.5}$$

  The second equation has two solutions $y_2 = 0 \text{and} y_2 = 1$ and we may get different situations depending on the initial conditions. t is a parameter of our choice. case 1: if $y_2 = 0$ we get $y_3 = t$ from the last equation and we can solve the first equation for y.
  case 2: Setting $y_2 = 1$ we get $y_1 = t$ from the last equation and $y_3 = 1$ comes out of the first one.

- Example 1.2: implicit algebraic variable.

  $$y' = f(y, z) \tag{2.1.6}$$

  $$0 = g(y) \tag{2.1.7}$$

  In this case we have that $g_z = 0$ and the condition of boundedness of the inverse does not hold. however if the condition that $g_y f_z$ has a bounded inverse holds we can do the trick of differentiating the second equation leading to

  $$0 = g_y(y)f(y, z) \tag{2.1.8}$$

  and this will then be like the semi explicit case where the conditions are satisfied. We now introduce the definition of *Index*

- *definition: Differential index.* For general DAE-systems we define the index along the solution path as the minimum number of differentiations of the systems that is required to reduce the system to a set of ODE's for the variable y.

- *description* The concept of index has been introduced in order to qualify the level of difficulty that is involved in the solution of the DAE. This must be understood in the way that methods that converge for one index may not be useful for higher index. Index is a concept that indicates the degree of difficulty.

- *definition: Perturbation index.* The perturbation index is defined as beeing complementary to the differential index [18] Problem (1) has perturbation index m along a solution y if m is the smallest integer such that , for all functions $\hat{y}$ having a defect

  $$f(y', y) = \delta(x) \tag{2.1.9}$$

  there exists an estimate

  $$\parallel \hat{y}(x) - y(x) \parallel \leq C(\parallel \hat{y}(0) - y(0) \parallel + max \parallel \delta(\xi) \parallel + \ldots + max \parallel \ \delta^{(m-1)}(\xi) \parallel) \tag{2.1.10}$$

  whenever the expression on the right hand side is sufficiently small. C is a constant that depends only on the function f and the length of the interval. If we consider the ODE-case (1.1), the lemma by Gronwall (see ref HWN1 p. 62) gives the bound

  $$\parallel \hat{y}(x) - y(x) \parallel \leq C(\parallel \hat{y}(0) - y(0) \parallel + \max_{0\leq\xi\leq x} \parallel \int_0^\xi \delta(t)dt \parallel). \tag{2.1.11}$$

  If we interpret this in order to find the perturbation index it is obviously zero.

- *Index reduction* A process called index reduction may be applied to a system for lowering the index from an initially high value down to f.ex. index one. This reduction is performed by successive differentiation and is often used in theoretical contexts. It is illustrated by an example.
- example 2.1 We look at example 1.2, (8),(9) and differentiate equation (9) two times whereby we obtain

  $$0 = g_y f_x + g_{yy} f^2 + g_y f_z z' \tag{2.1.12}$$

  $$z' = \frac{g_y f_x + g_{yy} f^2}{g_y f_z} \tag{2.1.13}$$

  The two differentiations have reduced the system to one of index zero and we can solve the resulting ODE-system using well known methods.

  If we want to find the perturbation index we may look at the equations for the perturbed system

  $$\hat{y}' = f(\hat{y}, \hat{z}) + \delta(x) \tag{2.1.14}$$

  $$0 = g(\hat{y}) + \theta(x) \tag{2.1.15}$$

  Using differentiation on the second equation leads to

  $$0 = g_y(\hat{y}) f(\hat{y}, \hat{z}) + g_y(\hat{y})\delta(x) + \theta'(x) \tag{2.1.16}$$

  ¿From the estimates and using Gronwalls lemma 2.1.11 like we did above we can now obtain

  $$\| \hat{y}(x) - y(x) \| \leq C(\| \hat{y}(0) - y(0) \| + \int_0^x (\| \delta(\xi) \| + \| \theta'(\xi) \|) d\xi \tag{2.1.17}$$

  $$\| \hat{z}(x) - z(x) \| \leq C(\| \hat{y}(0) - y(0) \| + max \| \delta(\xi) \| + max \| \theta'(\xi) \|) \tag{2.1.18}$$

  All the max- values are to be taken over $0 \leq \xi \leq x$.

2.1.3 Singular Perturbations.

One important source for DAE-problems comes from using singular perturbations [17] Here we look at systems of the form

$$y' = f(y, z) \tag{2.1.19}$$

$$\epsilon z' = g(y, z), 0 < \epsilon \ll 1 \tag{2.1.20}$$

Letting ϵ go to zero ($\epsilon \rightarrow 0$) and differentiating the second equation we obtain an index one problem in semi-explicit form. This system may be proven to have an ϵ - expansion where the expansion coefficients are solution to the system of DAE-s that we get in the limit (20,21). We will not here go into more detail around singular perturbations but refer to later examples.

2.2 The Van der Pol equation

A very famous test problem for systems of ODE's is the Van der Pol equation defined by the second order differential equation

$$y" = \mu(1 - y^2)y' - y$$

This equation may be treated in different ways, the most straigtforward is to split the equation into two by introducing a new variable for the derivative

$$\begin{aligned} &y_1 = y, y_2 = y' \\ &{y_1}' = y_2, {y_2}' = \mu(1 - {y_1}^2)y_2 - y_1 \end{aligned}$$

The system of two ODE's may be solved by any standard library solver but the outcome will depend on the solver and on the parameter μ. If we divide the second of the equations by μ we get an equation that has the character of a singular perturbation problem. Letting $\mu \to \infty$ we see that this corresponds to $\epsilon \to 0$ in equation (21).
Several other approaches may show other aspects of the nature of this problem for example [1] introduces the transformation $t = x/\mu$ after a scaling of y_2 by μ we get

$$\begin{aligned} &y_1' = y_2 \\ &y_2' = \mu^2((1 - {y_1}^2)y_2 - y_1) \end{aligned}$$

The introduction of $\epsilon = 1/\mu^2$ finally results in a problem in singular perturbation form

$$\begin{aligned} &y_1' = y_2 \\ &\epsilon y_2' = (1 - {y_1}^2)y_2 - y_1 \end{aligned}$$

As explained previously this problem will approach a DAE if $\epsilon \to 0$ Using terminology from ODE's the stiffness of the problem increases as ϵ gets smaller giving rise to stability problems for numerical ODE-solvers that are explicit while we expect methods that are A- or L-stable to perform better. Although the Van der Pol equation is not really a singular perturbation problem by nature it shares the characteristics of one as $\epsilon \to 0$.

Chapter 3

The Plane Double Pendulum

by : **Marie Bro**

3.1 Introduction

This chapter deals with the plane double pendulum and the problems that arise when treating it as a DAE-system. The double pendulum is interesting because it exhibits interesting and very complicated behaviour and when generalized to a pendulum consisting of n joints it can be used to model the motions of a rope or a long chain hanging from a fixed point.

The plane double pendulum consists of two masses m_1 and m_2 moving in the plane $\mathbb{R}^2$ connected by a bar of length ℓ_2 with no mass. The mass m_1 is connected to a fixed point by a bar without mass of length ℓ_1. The coordinate system is chosen so that the fixed point is at the origin (see figure 3.1). The only external force on the system is the gravitational force, since is ignored.

It should be quite obvious that this system can be formulated as a DAE-system. The two masses move in the plane, but they are confined to move on certain manifolds. It is clear that m_1 will move on a circle with radius ℓ_1, and that m_2 is confined to the annulus with inner radius $\ell_2 - \ell_1$ and outer radius $\ell_2 + \ell_1$. However it is possible to formulate the double pendulum as an ODE-system. If it is formulated in polar coordinates it is a system with two degrees of freedom described by four ODE's, whereas in cartesian coordinates it will be a system with 4 space coordinates desribed by 8 ODE's with two algebraic constraints making it a system with 2 degrees of freedom i.e. it is a DAE system.(sections 3.2 and 3.4)

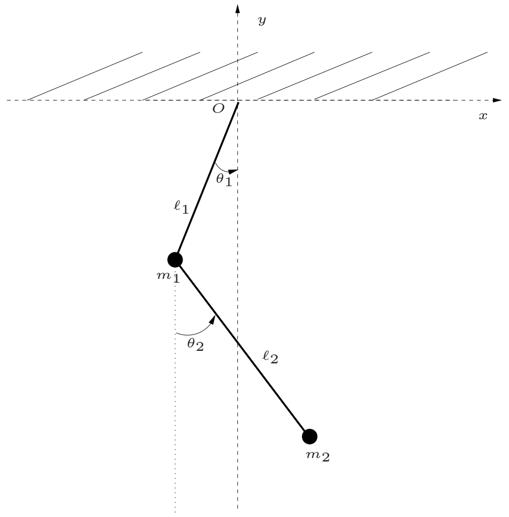


Figure 3.1: *The plane double pendulum*

It is not very clear why it should be interesting to treat this problem in cartesian coordinates since the dimension of the system is doubled doubled compared to polar coordinates and on top of that a DAE-formulation is harder to deal with than an ODE-formulation. It turns out however that the equations in cartesian coordinates seems much simpler, and it is easier to add more masses. In section 3.4 it is shown that when formulated in cartesian coordinates analytically it is no harder to append the n'th mass than it is to append the second mass i.e. turning the simple plane pendulum into a double pendulum.

However the cartesian formulation cause some problems since it gives rise to singularities which has nothing to with the physics of the problem. They related to the choice of coordinates. The main goal of this chapter will be to deal with these singularities and to investigate whether they make the formulation in cartesian coordinates too hard to handle numerically. Even though if is possible to handle the singularities for the simple pendulum and for the double pendulum, it might not be possible to handle the singularities for a pendulum with a large number of joints in practice, since the system has a singularity for each joint.

3.2 Formulation and reduction of the DAE

The double pendulum can be formulated in the Hamiltonian formalism [13]. The kinetic energy, T, of the system is given by:

$$T = \frac{m_1}{2}\left(\dot{x}_1^2 + \dot{y}_1^2\right) + \frac{m_2}{2}\left(\dot{x}_2^2 + \dot{y}_2^2\right) \tag{3.2.1}$$

the potentiental energy, V, is given by:

$$V = m_1 g y_1 + m_2 g y_2 \tag{3.2.2}$$

this gives us the Hamiltonian, H:

$$H = \frac{{p_{x_1}}^2 + {p_{y_1}}^2}{m_1} + \frac{{p_{x_2}}^2 + {p_{y_2}}^2}{m_2} + m_1 g y_1 + m_2 g y_2$$

from which we obtain the dynamical equations:

$$\begin{aligned} \dot{x}_1 &= \frac{p_{x_1}}{m_1} \qquad & \dot{p}_{x_1} &= 0 & \quad (3.2.3a)\\ \dot{y}_1 &= \frac{p_{y_1}}{m_1} \qquad & \dot{p}_{y_1} &= m_1 g & \quad (3.2.3b)\\ \dot{x}_2 &= \frac{p_{x_2}}{m_2} \qquad & \dot{p}_{x_2} &= 0 & \quad (3.2.3c)\\ \dot{y}_1 &= \frac{p_{y_2}}{m_2} \qquad & \dot{p}_{y_2} &= m_2 g & \quad (3.2.3d) \end{aligned}$$

These differential equations is just a description of two independent masses in a gravitational force field. To describe the double pendulum they must be combined with the algebraic constraints

$$\begin{aligned} {x_1}^2 + {x_2}^2 - {\ell_1}^2 &= 0 & \quad (3.2.4a)\\ (x_2 - x_1)^2 + (y_2 - y_1)^2 - {\ell_2}^2 &= 0 & \quad (3.2.4b) \end{aligned}$$

which, as we will see shortly introduce horizontal forces in the system.

The approach to solve the DAE is index reduction i.e. we diffrentiate the constraints with respect to time in order to obtain differential equations instead of numerical equations. Differentiation of the first constaint (equation (3.2.4a)) gives:

$$\begin{aligned}
2x_1\dot{x}_1 + 2y_1\dot{y}_1 &= 0 \quad \Leftrightarrow \\
\frac{2}{m_1}x_1p_{x_1} + \frac{2}{m_1}y_1p_{y_1} &= 0 \quad \Leftrightarrow \\
x_1p_{x_1} + y_1p_{y_1} &= 0
\end{aligned}$$

This means that the position vector and the momentum vector of m_1 are orthogonal which is equivalent to ℓ_1 being fixed. However this constraint does not reveal the length of the bar only that it is constant. The actual length of the bar will come from the initial conditions.

It is still an algebraic constraint so we differentiate once more:

$$\begin{aligned}
\dot{x}_1p_{x_1} + x_1\dot{p}_{x_1} + \dot{y}_1p_{y_1} + x_1\dot{p}_{y_1} &= 0 \quad \Leftrightarrow \\
\frac{{p_{x_1}}^2}{m_1} + x_1\dot{p}_{x_1} + \frac{{p_{y_1}}^2}{m_1} + m_1gy_1 &= 0 \quad \Leftrightarrow \\
x_1\dot{p}_{x_1} &= -\frac{{p_{x_1}}^2}{m_1} - \frac{{p_{y_1}}^2}{m_1} - m_1gy_1
\end{aligned}$$

Differentiating the second constraint (equation (3.2.4b)) gives:

$$(x_2 - x_1)\left(\frac{p_{x_2}}{m_2} - \frac{p_{x_1}}{m_1}\right) + (y_2 - y_1)\left(\frac{p_{y_2}}{m_2} - \frac{p_{y_1}}{m_1}\right) = 0$$

Just like above we get a new algebraic constraint, and again it expresses orthogonality between a position vector of a point and its velocity vector. This time it is the position and the velocity of m_2 in a coordinate system centered at m_1. Like the case for ℓ_1 this implies that ℓ_2 is fixed but the length of the bar is no longer given by the equations. It will be supplied by the initial conditions. The second differentiation gives:

$$\begin{aligned}
&(\dot{x}_2 - \dot{x}_1)\left(\frac{p_{x_2}}{m_2} - \frac{p_{x_1}}{m_1}\right) + (x_2 - x_1)\left(\frac{\dot{p}_{x_2}}{m_2} - \frac{\dot{p}_{x_1}}{m_1}\right) \\
&\qquad + (\dot{y}_2 - \dot{y}_1)\left(\frac{p_{y_2}}{m_2} - \frac{p_{y_1}}{m_1}\right) + (y_2 - y_1)\left(\frac{\dot{p}_{y_2}}{m_2} - \frac{\dot{p}_{y_1}}{m_1}\right) = 0
\end{aligned}$$

Using equations (3.2.3) we can substitute the time derivatives of the posistion coordinates with the momenta:

$$\left(\frac{p_{x_2}}{m_2}-\frac{p_{x_1}}{m_1}\right)^2+(x_2-x_1)\left(\frac{\dot{p}_{x_2}}{m_2}-\frac{\dot{p}_{x_1}}{m_1}\right)+\left(\frac{p_{y_2}}{m_2}-\frac{p_{y_1}}{m_1}\right)^2+(y_2-y_1)\left(\frac{m_2g}{m_2}-\frac{m_1g}{m_1}\right)=0$$

From this we obtain:

$$(x_2-x_1)\left(\frac{\dot{p}_{x_2}}{m_2}-\frac{\dot{p}_{x_1}}{m_1}\right)=-\left(\frac{p_{x_2}}{m_2}-\frac{p_{x_1}}{m_1}\right)^2-\left(\frac{p_{y_2}}{m_2}-\frac{p_{y_1}}{m_1}\right)^2$$

The system can now be written as:

$$\mathbf{M}\begin{bmatrix}\dot{x}_1\\ \dot{y}_1\\ \dot{x}_2\\ \dot{y}_2\\ \dot{p}_{x_1}\\ \dot{p}_{y_1}\\ \dot{p}_{x_2}\\ \dot{p}_{y_2}\end{bmatrix}=\begin{bmatrix}p_{x_1}\\ p_{y_1}\\ p_{x_2}\\ p_{y_2}\\ -{p_{x_1}}^2-{p_{y_1}}^2-{m_1}^2gy_1\\ g\\ -\left(\frac{p_{x_2}}{m_2}-\frac{p_{x_1}}{m_1}\right)^2-\left(\frac{p_{y_2}}{m_2}-\frac{p_{y_1}}{m_1}\right)^2\\ g\end{bmatrix}$$

where $\mathbf{M}$ is the mass matrix:

$$\mathbf{M}=\begin{bmatrix}m_1&0&0&0&0&0&0&0\\ 0&m_1&0&0&0&0&0&0\\ 0&0&m_2&0&0&0&0&0\\ 0&0&0&m_2&0&0&0&0\\ 0&0&0&0&m_1x_1&0&0&0\\ 0&0&0&0&0&\frac{1}{m_1}&0&0\\ 0&0&0&0&\frac{x_1-x_2}{m_1}&0&\frac{x_2-x_1}{m_2}&0\\ 0&0&0&0&0&0&0&\frac{1}{m_2}\end{bmatrix}$$

The system apparently have index 2, but there are two singularities in the reduced system. The mass matrix is singular for $x_1=0$ for $x_1=x_2$, i.e. when either one of the bars is in a vertical position the system has a singularity. This means that it will occur all the time and therefore it has to be dealt with. It cannot be deleted by altering the orientation of the coordinate system. This will only make another direction singular which it not solving our problem since we want to be able to model all kinds of trajectories for the double pendulum.

At the singularities the system is actually still a DAE i.e. the system does not have the same index in all points. These singularities has nothing to do with the properties of the pendulum it stems from the choice of coordinates.

3.3 Consistent initial conditions

As indicated in the introduction not all inital conditions are consistent with the system. First of all the position vectors must be right: The mass m_1 must be positioned on a circle centered at the origin with radius ℓ_1 and m_2 must be positioned a distance ℓ_2 from m_1. Next not all momenta are allowed: m_1 must stay on its circle and m_2 must stay at the exact distance ℓ_2 from m_1.

An easy way to obtain consistent initial condition in cartesian coordinates is to give the initial conditions in polar coordinates, and then calculate the corresponding cartesian coordinates.
The position in cartesian coordinates is given by (see figure 3.1):

$$x_1 = \ell_1 \sin(\theta_1) \tag{3.3.1a}$$

$$y_1 = -\ell_1 \cos(\theta_1) \tag{3.3.1b}$$

$$x_2 = \ell_1 \sin(\theta_1) + \ell_2 \sin(\theta_2) \tag{3.3.1c}$$

$$y_2 = -\ell_1 \cos(\theta_1) - \ell_2 \cos(\theta_2) \tag{3.3.1d}$$

The corresponding momenta are obtained by differentiating the position with respect to time:

$$p_{x_1} = m\dot{x}_1 = m_1\dot{\theta}_1\ell_1 \cos(\theta_1) \tag{3.3.2a}$$

$$p_{y_1} = m\dot{y}_1 = m_1\dot{\theta}_1\ell_1 \sin(\theta_1) \tag{3.3.2b}$$

$$p_{x_2} = m\dot{x}_2 = m_1\dot{\theta}_1\ell_1 \cos(\theta_1) + m_2\dot{\theta}_2\ell_2 \cos(\theta_2) \tag{3.3.2c}$$

$$p_{y_2} = m\dot{y}_2 = m_1\dot{\theta}_1\ell_1 \sin(\theta_1) + m_2\dot{\theta}_2\ell_2 \sin(\theta_2) \tag{3.3.2d}$$

Thus we can obtain the sought initial, $(x_{1,0}, y_{1,0}, p_{x_{1,0}}, p_{y_{1,0}})$ and $(x_{2,0}, y_{2,0}, p_{x_{2,0}}, p_{y_{2,0}})$, by giving the corresponding polar initial conditions $(\theta_{1,0}, \dot{\theta}_{1,0})$ and $(\theta_{2,0}, \dot{\theta}_{2,0})$. In polar coordinates all initial conditions are consistent, since all angles and all angular velocisties are allowed.

3.4 The n-mass pendulum

In this section we extend the system to have as many joints as we want it to have. We start by adding a third joint consisting of a mass m_3 connected to m_2 by a massless bar of length ℓ_3. This will add four differential equations and an algebraic constraint to our system. The ODE's are the equations for m_3 in a gravitational force field:

$$\begin{aligned} \dot{x}_3 &= \frac{p_{x_3}}{m_3} & \dot{p}_{x_3} &= 0 \\ \dot{y}_3 &= \frac{p_{y_3}}{m_3} & \dot{p}_{y_3} &= m_3 g \end{aligned}$$

and the algebraic constraint is that m_3 has the constant distance of ℓ_3 to m_2:

$$(x_3 - x_2)^2 + (y_3 - y_2)^2 - {\ell_3}^2 = 0$$

Apart from the indices, the equations for m_3 are identical to the equations for m_2. Thus it is no different to add the third joint than it is to add the second joint. From this we can conclude, that if we can model the double pendulum in cartesian coordinates we can in principle model a pendulum with any number of joints without further conceptual difficulties. The size of the system might though give numerical problems.

The n'th ($n \in \mathbb{N}$) joint will add the following equations to the reduced system

$$\dot{x}_n = \frac{p_{x_n}}{m_n} \tag{3.4.1a}$$

$$\dot{y}_n = \frac{p_{y_n}}{m_n} \tag{3.4.1b}$$

$$(x_n - x_{n-1})\left(\frac{\dot{p}_{x_n}}{m_n} - \frac{\dot{p}_{x_{n-1}}}{m_{n-1}}\right) = -\left(\frac{p_{x_n}}{m_n} - \frac{p_{x_{n-1}}}{m_{n-1}}\right)^2 - \left(\frac{p_{y_n}}{m_n} - \frac{p_{y_{n-1}}}{m_{n-1}}\right)^2 \tag{3.4.1c}$$

$$\dot{p}_{y_n} = m_n g \tag{3.4.1d}$$

This shows us two things: The first one is that for each joint another singularity is added. Whenever a bar passes a vertical position there is a singularity. The second one is that the only direct influence on a particular mass in the system comes from the two neighbouring masses and is supplied by the vertical forces given by equation (3.4.1c). As a consequence of that the already obtained equations are not altered by the addition of another joint.

Now we will look into the polar formulation of the pendulum and try to add more joints. We will mainly look at the energy of the system, since it provides the dynamic equations and they are very complicated

The energy for the double pendulum expressed in polar coordinates can be obtained by substituting the coordinate and momentum transformations from cartesian to polar coordinates (equations (3.3.1) and (3.3.2)) into the expressions for the energy (3.2.1) and (3.2.2):

$$T = \frac{1}{2}(m_1 + m_2){\ell_1}^2\dot{\theta}_1^2 + \frac{1}{2}m_2{\ell_2}^2\dot{\theta}_2^2 + m_2\ell_1\ell_2\dot{\theta}_1\dot{\theta}_2\cos(\theta_2 - \theta_1) \tag{3.4.2}$$

$$V = -m_1 g\ell_1\cos(\theta_1) - m_2 g\left(\ell_1\cos(\theta_1) + \ell_2\cos(\theta_2)\right) \tag{3.4.3}$$

The equations for the time derivatives of the angles, θ_i, (i.e. the angular velocities) comes from the terms in the energy functions that depend on $\dot{\theta}_i$. The equations for the time derivatives of the angular momenta, p_i, (i.e. the forces) comes from the terms that depend on θ_i. The reader is refered to the [13] for a derivation of the Hamiltonian formalism and the procedure when formulating a Hamiltonian system.

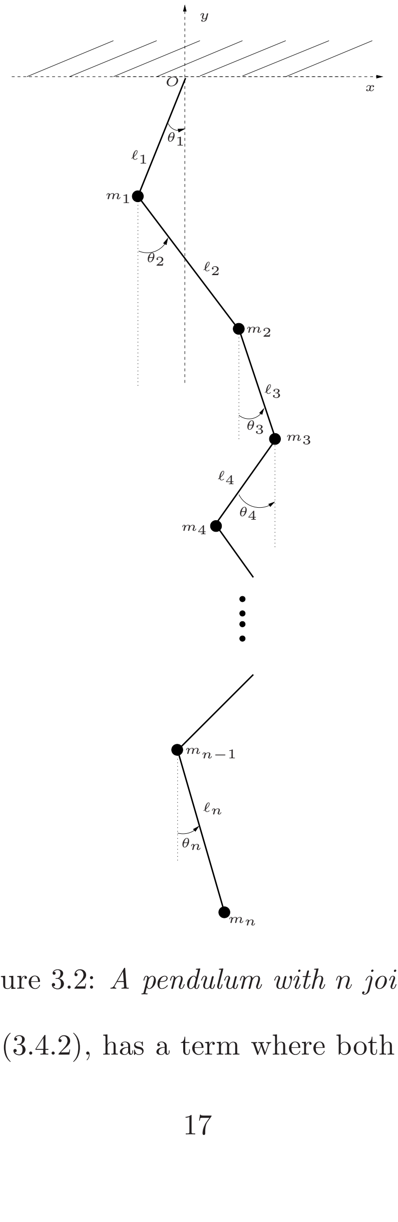


Figure 3.2: *A pendulum with n joints*

The kinetic energy, equation (3.4.2), has a term where both $\dot{\theta}_1$ and $\dot{\theta}_2$ appears. It means that the time evolution of the two angles are coupled to each other through the equations for the angular velocities. The same term also depends on both θ_1 and θ_2, so the time evolution is also coupled through the force equations.

The potential energy, equation (3.4.2), does not contain any terms dependent on more than one dynamic variable so it does not provide much coupling allthough it does introduce m_2 into the equation of $\dot{\theta}_1$, as does the first term of the kinetic energy.

Now we look at the kinetic energy of the the 3-joint pendulum (the potential does not contain any interesting extra terms):

$$\begin{aligned} T =& \frac{1}{2}(m_1 + m_2 + m_3){\ell_1}^2\dot{\theta}_1^2 + \frac{1}{2}(m_2 + m_3){\ell_2}^2\dot{\theta}_2^2 + \frac{1}{2}m_3{\ell_3}^2\dot{\theta}_3^2 \\ & + (m_2 + m_3)\ell_1\ell_2\dot{\theta}_1\dot{\theta}_2 \cos(\theta_2 - \theta_1) + m_3\ell_1\ell_3\dot{\theta}_1\dot{\theta}_3 \cos(\theta_3 - \theta_1) \\ & + m_3\ell_1\ell_3\dot{\theta}_2\dot{\theta}_3 \cos(\theta_3 - \theta_2) \end{aligned}$$

It is obtained from the kinetic energy in cartesian coordinates the same way as for the double pendulum. In stead of just one term dependent of four dynamical variables it has three. The time evolution of each of the angles is coupled directly to each of the other angles and not just to the neigbouring ones. This means that when working in polar coordinates the pendulum with three joints is not obtained by just adding to new equations to the system of the double pendulum. All the equations are new and that makes it more difficult to extend the system. However it does not make the numerical calculations that much harder. The three joint pendulum only consist of two more equations than the double pendulum, and no singularities are added.

The kinetic energy of the 4-joint pendulum is even worse. This is due the fact the the extra terms stems from squaring a quantity of four terms. In the n-joint pendulum the extra terms stem from squaring a quantity of n terms.

3.5 Implementation

The strategy is to start out by implementing the simple plane pendulum and find out how to handle the singularity. Then the rest of the joints can be appended one at a time. For each joint a new singularity has to be handled by the solver. Hopefully it will be the first two joints that gives the most dificulties. Once the singularities of the double pendulum are handled it should not be a problem writing the code for handling the rest of the singularities. Numerical problems might arise though, as the number of singularities is raised.

The simple pendulum, in reduced form, is given by:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{p}_x \\ \dot{p}_y \end{bmatrix} = \begin{bmatrix} \frac{p_x}{m} \\ \frac{p_y}{m} \\ \frac{-1}{x}\left(\frac{{p_x}^2+{p_y}^2}{m} + mgy\right) \\ mg \end{bmatrix}$$

Note that it is not formulated as a system with a mass matrix. The standard ODE-solvers of MatLab can all be applied to systems formulated with a mass matrix, but none of them are applicable to this problem because the mass matrix is singular. Some of them can handle an ODE-system with a singular mass matrix (i.e. a DAE) but it has to be index one and apperently the index of this system is greater than one at the singularity. It is not possible to use MatLabs event detection in handling the singularity since it cannot detect events where the mass matrix is singular. Therefore it is nescesarry to write a method that can handle the singularity.

A first approach is to use a simple method with constant stepsize. This will make the risk of hitting the singularity smaller than when the stepsize is variable, in which case the stepsize will be controlled by the size of the error. When the solution approaches the singularity the error grows and the stepsize is reduced and thus it almost certain that the singularity is hit. With constant stepsize it should be possible to step over the singularity.

We try with the trapezoidal rule:

$$y_{n+1} = y_n + \frac{h}{2}(f_n + f_{n+1}) \tag{3.5.1}$$

where the ODE is given by $y' = f(y)$ and $f_n = f(y_n)$. The method is implicit so we have to use iterations to take a step along the solution curve. It turns out that fixed point iterations are very fast, when the mass is not too close to the singularity. When $|x| > 0.15$ and the stepsize is $h = 10^{-3}$ it takes only iterations to obtain an accuracy of size 10^{-6}. When $|x| < 0.15$ the stepsize has to be smaller other wise the fixed point iteration diverge quite fast. In fact to avoid divergence of the fixed point iteration when close to the singularity one has to keep decreasing the stepsize, and this will make it impossible to step over the singularity.

Newton iterations might then be interesting. When solving the nonlinear equation

$$y = \phi(y)$$

using Newtons method the iteration scheme is given by [**?**]:

$$y^{v+1} = y^v - \left[I - \frac{\partial \phi}{\partial y}(y^v) \right]^{-1} [y^v - \phi(y^v)]$$

where $\frac{\partial \phi}{\partial y}(y^v)$ is the differential (i.e. the Jacobi-matrix) of ϕ with respect to y at y^v. In our case ϕ is the righthand side of equation (3.5.1) and we are solving for y_{n+1}. Therefore the scheme is:

$$\begin{aligned} y_{n+1}^{v+1} &= y_{n+1}^v - \left[I - \frac{\partial \phi}{\partial y_{n+1}}(y_{n+1}^v) \right]^{-1} [y_{n+1}^v - (y_n + \frac{h}{2}(f_n + f(y_{n+1}^v)))] \\ &= y_{n+1}^v - \left[I - \frac{h}{2}\frac{\partial f}{\partial y}(y_{n+1}^v) \right]^{-1} [y_{n+1}^v - \left(y_n + \frac{h}{2}\left(f_n + f(y_{n+1}^v) \right) \right)] \end{aligned}$$

Thus it is not a problem that the Jacobi matrix is singular because it is subtracted from the unit matrix before inversion.

After stepping over the singularity we use the continuous extension [14] of trapezoidal rule to determine the exact time passing the singularity. In this manner we can avoid drifting. The solution obtained when stepping over the singularity is probaly not very accurate, and by going back a step and calculating the time of crossing using a continuous version of the trapezoidal rule it should be possible to get a more accurate value of the dynamical variables right after the crossing.

The continuous extension is given by:

$$y(t_n + \theta h) = y_n + h \left(k_1 \theta \left(1 - \frac{\theta}{2} \right) + k_2 \theta \right)$$

where $y_{n+1} = y(t_n + h)$ and $\theta \in [0;1]$ is parameter. The quantities k_1 and k_2 stems from viewing the trapezoidal rule as a two-stage Runge-Kutta method. They are thus given by:

$$\begin{aligned} k_1 &= f_n \\ k_2 &= f_{n+1} \end{aligned}$$

Note that we have already made the step so therefore k_2 is known. We know that the singularity occurs when $x = 0$ therefore we can obtain the parameter value θ by solving the scalar equation:

$$\begin{aligned} x(t_n + \theta h) &= 0 \qquad \Leftrightarrow \\ y_n(1) + h \left(f_n(1) \theta \left(1 - \frac{\theta}{2} \right) + f_{n+1}(1) \theta \right) &= 0 \end{aligned}$$

where the number 1 in parentheses refers to the first element of the vector. The solution, θ^* to this equation reveals the time for the occurence of the singularity. By setting $y_{n+1} = y(t_n + (\theta^* + \varepsilon)h)$ we get a starting point after the singularity which is very close to the singularity and this should give us a better solution curve.

A last comment in this section is concerneing the stepsize. Allthough stated in the beginning it is not nescesarry to keep stepsize completely constant. When crossing the singularity it is important to have quite small stepsize, but away from the singularity there is no reason for this. As long as there is a lower bound for the stepsize there is no problem in using differentstepsizes in different domains. In this implementation the stepsize is not determined by the error but by the size of the x-coordinate. We operate with three different stepsizes.

3.6 Numerical results

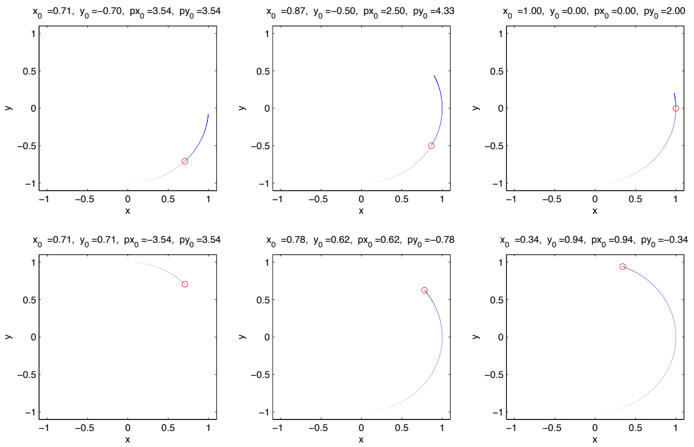


Figure 3.3: *The trajectories for the pendulum with different initial conditions without attemps to step over singularity. Thus the stepsize is not changed as the singularity is approached and the fixed point iterartions fail at some point. Before this happens the mass stay on a circular trajectory. The small circle marks the starting point.*

The numerical results are not very extensive. The singularity in the simple pendulum has been passed but there has been made no continuous extension. The double pendulum has not been implemented.

There are good news though. When the solution curve is not close to the singularity it displays the kind of dynamics we expect. It follows a circular path and it is affected by gravity, so it looses speed and in some cases changes direction when going upwards (see figures 3.3 og 3.4). The figures have been obtained by solving the system without making any attempts to overcome the singularity. Just to see how far the fixed point iteration gets us. Of course it might happen that changing the stepsize will get us past the singularity, but figure 3.5 shows that this is not the case. It is not posssible to pass the singularity using the trapezoidal rule and fixed point iterations. This is due to the fact that the fixed point iteration diverge sufficiently close to the singularity. Figure 3.6 is the graphs of the length of the pendulum as a function of time.The graphs give a hint that the accuracy of this method, as expected, is increased as the stepsize is reduced. Especially the graph for $h = 10^{-5}$ is impressive, but it is not really possible to use this stepsize, since the computations take too long. It seems optimal to use the stepsize $h = 10^{-3}$ away from the singularity, reducing it to $h = 10^{-4}$ for a while close to the singularity and then switch to Newton iterations. This method have not yet really had succes. Although the singularity has been passed the timestep needed in order to make the Newton iterations converge is very small and therefore the computation time is very large, in fact so large that unless it can somehow be decreased appreciably it is not realistic to use this model to simulate the double pendulum. The long computation time is probably also related to the fact that the matrix that is inverted when using the Newton method is very close to singular.

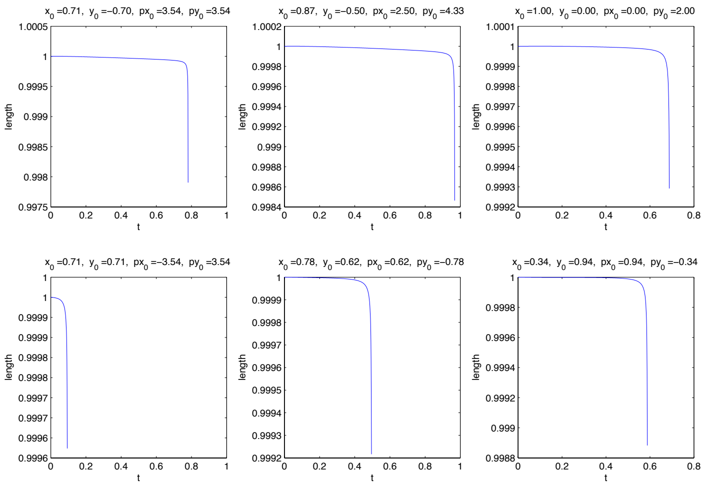


Figure 3.4: *These graphs shows the length of the bar as a function of time. It is seen to be as good as constant until the singularity is approached.*

Another approach to overcoming singularities in DAE's can be found in [16]. This article describes a method for transforming the singular ODE into a nonsingular ODE by making a transformation in the independent variable. However it will take some work to make this operational since the approach in the article is rather abstract an no concrete examples are given.

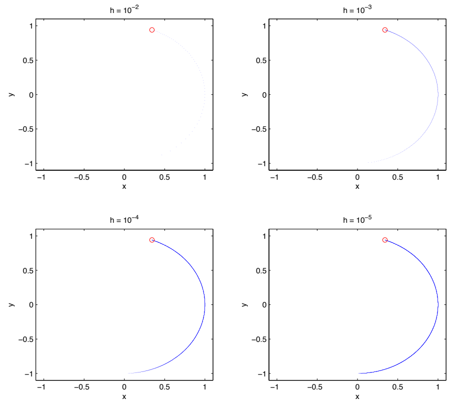


Figure 3.5: *The trajectory for the initial conditions* $(x_{1,0}, y_{1,0}, p_{x_{1,0}}, p_{y_{1,0}}) = (0.34, 0.94, 0.94, -0.34)$ *with different stepsizes. Obviously the smaller the stepsize the closer to the singularity we can get, but does not seem possible to pass the singularity.*

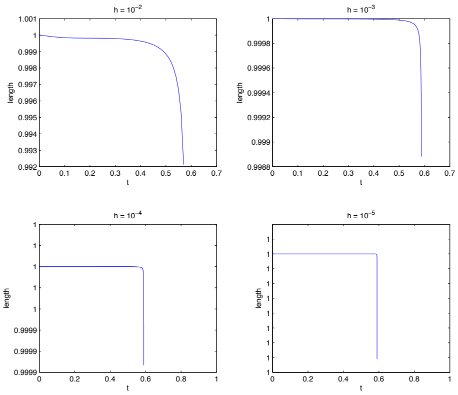


Figure 3.6: *The graphs of the length as a function of time for different stepsizes for the initial condition* $(x_{1,0}, y_{1,0}, p_{x_{1,0}}, p_{y_{1,0}}) = (0.34, 0.94, 0.94, -0.34)$

3.7 Conclusion

The double pendulum has been formulated in cartesian coordinates which is justified by the fact, that the equations are easier to survey than the polar formulation. However it gives rise to numerical difficulties, because the equations contains singularities which are due to the choice of coordinates. The strategy for overcoming these singularities has been to use a simple method with censtant stepsize and then simply just 'step over' the singularity. It has not had great succes because the computation time is too long.

Chapter 4

Different approaches to ODE systems with invariants.

by : **Michaël Gineste**

4.1 Introduction

This report will focus on a system of ordinary differential equations with an invariant belonging to it. This invariant is a property of the system, some instance which (usually) is constant over time.
The dynamical system in regard is the restricted three body problem which has an energy conservation property.

Many ODE or DAE systems has one or more invariants along with the model of the system, some being just extra information on the solution, others an important quantity preferably conserved.

This report will investigate how this invariant can be used to improve the numerical solution of the system. This report is somehow an implementation of the methods presented in [29], and an attempt to evaluate their performence. The approaches differ in fundamental idea, so a detailed description of arguments and theory behind each approach is not presented, but should easily be found in the litterature.

4.2 The model

The restricted three body problem is used as test problem, which expresses a particle with neglible mass in the gravitational field of two much larger masses, in this case a satellite moving in the Earth-Moon system. In figure 4.1 the reference system of the dynamics is shown, the position of earth is $(-\mu, 0)$ and the moon is positioned in $(1-\mu, 0)$, where μ is the mass ratio between moon and earth. Here origo is in the center of gravity.

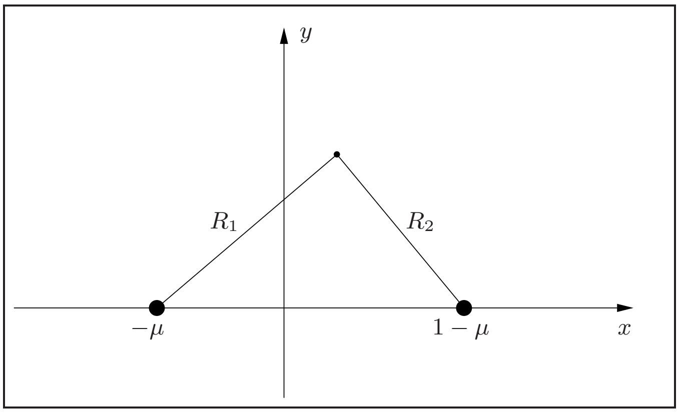


Figure 4.1: Model of restricted three body problem

Newtonian formulation

The differential equations describing the satellite dynamics are

$$x'' = x + 2y' - \beta\frac{x+\mu}{R_1^3} - \mu\frac{x-\beta}{R_2^3} \tag{4.2.1a}$$

$$y'' = y - 2x' - \beta\frac{y}{R_1^3} - \mu\frac{y}{R_2^3} \tag{4.2.1b}$$

where $\beta = 1-\mu$, and the distances

$$R_1 = \left((x+\mu)^2 + y^2\right)^{\frac{1}{2}}, \quad R_2 = \left((x-\beta)^2 + y^2\right)^{\frac{1}{2}} \quad .$$

Formulated as an first order ODE system with $\boldsymbol{y} = (x, y, x', y')^T$ the system (4.2.1) can be written as a autonomous system

$$\boldsymbol{y}' = \boldsymbol{f}(\boldsymbol{y}) = \begin{pmatrix} y_3 \\ y_4 \\ y_1 + 2y_4 - \beta\frac{y_1+\mu}{R_1^3} - \mu\frac{y_1-\beta}{R_2^3} \\ y_2 - 2y_3 - \beta\frac{y_2}{R_1^3} - \mu\frac{y_2}{R_2^3} \end{pmatrix} \tag{4.2.2}$$

The system (4.2.1) has a known first integral invariant, know as the Jacobi Integral

$$I = \tfrac{1}{2}\left(x'^2 + y'^2 - x^2 - y^2\right) - \frac{\beta}{R_1} - \frac{\mu}{R_2} \tag{4.2.3}$$

which describes the conservation of energy of the moving satellite.

This invariant should be constant over time, so this requirement can be regarded as a 'constraint' on the ODE system, i.e. the solution to (4.2.2) should be on the manifold

$$\mathcal{M} = \left\{ \boldsymbol{y} \in \mathbb{R}^4 \mid g(\boldsymbol{y}) = I(\boldsymbol{y}(t)) - I(\boldsymbol{y}(0)) = 0 \right\} \qquad .$$

4.3 The approaches

The overall goal with this report is to investigate how different formulation of the numerical solution can affect the solution, especially with long-time integration.

Different formulation is meant in a broader meaning, the physical problem is still the same, but the numerical method is changed.

The cases for evaluation are:

1. Plain ODE system.
2. Formulation as DAE problem by inclusion of invariant.
3. Postprocessing of solution to ODE system (4.2.1) by use of invariant (4.2.3).
4. Reformulation of model as Hamiltonian system, applied with symplectic integrator.

The numerical methods used to integrate the solution will be of an explicit and implicit kind with fixed stepsizes, the Improved Euler method and the implicit midpoint rule (IMR). These methods are of order two, and are used to better recognize the effects of the different approaches. Each of the initiatives is applied to calculations with both the explicit and the implicit method.

One could argue that such low order methods should not be used for long integration of this dynamical system (and especially not with fixed stepsize), but in test with use of the MATLAB solver `ode45`, the numerical method was simply too good to see the effects of the initiatives.

The Butcher tableau of the two methods are presented in figure 4.2, since these methods are computed as such (i.e. as RK methods).

$$\begin{array}{c|cc} 0 & 0 & \\ 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

(a) Improved Euler

$$\begin{array}{c|c} \frac{1}{2} & \frac{1}{2} \\ \hline & 1 \end{array}$$

(b) Implicit Midpoint Rule

Figure 4.2: Used numerical schemes

The implicit midpoint rule is a symplectic method (since it is a one stage Gauss-Legendre method) and hence useful to evaluate the reformulation of the system (4.2.2) into a Hamiltonian system, see section 4.3.4.

4.3.1 Plain ODE system

In this case, nothing is done to the system (4.2.2). Instead the circumstances under which the tests are performed are elaborated.

Evaluation of the numerical solution somehow requires a periodic solution which doesn't pass too close to the singular points $(x, y) = (1 - \mu, 0)$ or $(-\mu, 0)$, since a fixed stepsize is used in this report. A more or less classic periodic solution is used, which has the initial conditions:

$$\begin{aligned} x(0) &= 1.2 & x'(0) &= 0 \\ y(0) &= 0 & y'(0) &= -1.04935750983031990726 \end{aligned}$$

along with the ratio μ and period T

$$\mu = 1/82.45 \qquad T = 6.19216933131963970674 \quad .$$

The solution to this IVP can be seen in figure 4.3.

The error of the numerical solution is chosen to be the drift in position in one dimension i.e. when the periodic solution crosses the line $y = 0$, the error is as

$$\text{error}(k) = |x(0) - x_n(kT)| \tag{4.3.1}$$

where $x_n(kT)$ is the solution found (interpolated) at $y = 0$ for the k'th period. The numerical solution isn't always near the axis at a multiplum of the period, so its the period number that counts.

This error is used to evaluate the numerical method over 'long-time' integration (10 times a period).

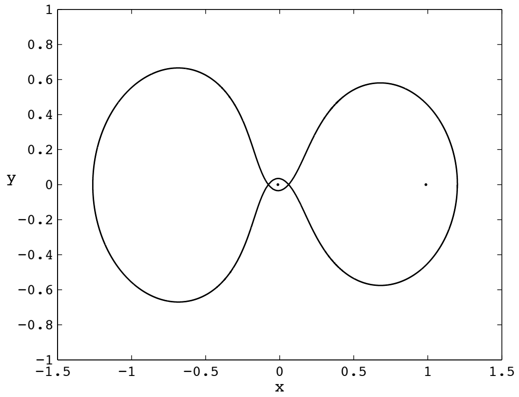


Figure 4.3: Orbit

4.3.2 DAE formulation

A way of including the invariant-'constraint' into the ODE system, turning it into a DAE system, is by use of Lagrange multipliers. The multiplier λ is introduced in (4.2.2) as

$$\begin{aligned} \boldsymbol{y}' &= \boldsymbol{f}(\boldsymbol{y}) - \boldsymbol{G}(\boldsymbol{y})\lambda \\ 0 &= g(\boldsymbol{y}) \end{aligned} \tag{4.3.2}$$

where $g(\boldsymbol{y}) = I(\boldsymbol{y}) - I(\boldsymbol{y}_0)$ and the function $\boldsymbol{G}(\boldsymbol{y})$ is chosen to be $(\partial g/\partial \boldsymbol{y})^T$, which gives an orthogonal projection of the solution onto the constraint manifold $\mathcal{M}$.

Performing index reduction on the constraint would give λ in terms of $\boldsymbol{y}$

$$\lambda = \frac{g_{\boldsymbol{y}}\boldsymbol{f}}{g_{\boldsymbol{y}}\boldsymbol{G}}$$

giving an index-1 DAE system, for which an appropriate solver could be applied, for example the MATLAB solver `ode15s` or `ode23t`.

Unfortunately, a property of the definition of a first integral to the system (4.2.2), is that

$$\frac{\partial I}{\partial \boldsymbol{y}} f = 0$$

and since $\partial I/\partial \boldsymbol{y} = g_{\boldsymbol{y}}$, the introduction of the multiplier into the ODE system gives nothing.

4.3.3 Postprocessing

By postprocessing we will mean taking a time-step in the solution of the ODE system with some method, and then 'pulling back' or projecting this estimate onto the constraint manifold. Different ways of doing this are present in the litterature, an approach in [29] is: having solution value $\boldsymbol{y}_{n-1}$, advancing one step with an ODE method ϕ_h giving an estimate $\tilde{\boldsymbol{y}}$ and then solving

$$\begin{aligned} \|\tilde{\boldsymbol{y}} - \boldsymbol{y}_n\|_2 &= \min_{\boldsymbol{y}_n} \\ g(\boldsymbol{y}_n) &= 0 \end{aligned}$$

which constitutes a nonlinear constrained least squares problem.

This approach was implemented by use of the Matlab built-in function `fmincon` which "finds a constrained minimum of a function of several variables" (from `fmincon` help) by the setup

$$\texttt{fmincon}(\|\tilde{\boldsymbol{y}} - \boldsymbol{y}_n\|, \tilde{\boldsymbol{y}}, [\,], [\,], [\,], [\,], [\,], [\,], g(\boldsymbol{y}_n))$$

where $g(\boldsymbol{y})$ is as in (4.3.2).

Note: The use of this function is quite evaluation expensive, implementation of an efficient algorithm set up for this specific problem could probably make this solution form more attractive.

Another way of doing postprocessing is to introduce a stabilizing term in the system such that the ODE system becomes

$$\boldsymbol{y}' = \boldsymbol{f}(\boldsymbol{y}) - \gamma \boldsymbol{F}(\boldsymbol{y})\left(I(\boldsymbol{y}) - I_0\right) \tag{4.3.3}$$

where the term $\boldsymbol{F}$ is

$$\boldsymbol{F} = \boldsymbol{E}^T \left(\boldsymbol{E}\boldsymbol{E}^T\right)^{-1} \quad , \quad \boldsymbol{E}(\boldsymbol{y}) = \frac{\partial I}{\partial \boldsymbol{y}}$$

such that for the system (4.2.2) the stabilization function $\boldsymbol{F}$ becomes

$$\begin{aligned} \boldsymbol{F}(\boldsymbol{y}) &= \begin{pmatrix} \left(-y_1 + \beta\frac{y_1+\mu}{R_1^3} + \mu\frac{y_1-\beta}{R_2^3}\right)/D(\boldsymbol{y}) \\ \left(y_2\left(-1 + \frac{\beta}{R_1^3} + \frac{\mu}{R_2^3}\right)\right)/D(\boldsymbol{y}) \\ y_3/D(\boldsymbol{y}) \\ y_4/D(\boldsymbol{y}) \end{pmatrix} \\ D(\boldsymbol{y}) &= \left(y_2^2\left(-1 + \frac{\beta}{R_1^3} + \frac{\mu}{R_2^3}\right)^2 + \left(-y_1 + \beta\frac{y_1+\mu}{R_1^3} + \mu\frac{y_1-\beta}{R_2^3}\right)^2 + y_3^2 + y_4^2\right) \end{aligned} \tag{4.3.4}$$

Note that this stabilization term vanishes on the constraint manifold.

This stabilization is applied after each advancement by the ODE method ϕ_h, such that the scheme becomes

$$\begin{aligned}\tilde{\boldsymbol{y}}_{n+1} &= \phi_h(\boldsymbol{y}_n)\\ \boldsymbol{y}_{n+1} &= \tilde{\boldsymbol{y}}_{n+1} - \gamma \boldsymbol{F}(\tilde{\boldsymbol{y}}_{n+1})(I(\tilde{\boldsymbol{y}}_{n+1}) - I_0)\end{aligned}$$

The constant $\gamma = 1$ is chosen, which should be an optimal value, according to [28] which also holds more information on this stabilization technique.

Since the stabilization term is a vector function, its inclusion doesn't increase the computational load significantly.

4.3.4 Hamiltonian reformulation

Reformulating the system (4.2.2) into a Hamiltonian form, and applying a symplectic solver to this, could improve the solution with regard to drift in position, since the symplectic method tends to mimic the flow of the Hamiltonian[1] system.

A Hamiltonian system is defined as

$$\boldsymbol{q}' = \frac{\partial H}{\partial \boldsymbol{p}} , \qquad \boldsymbol{p}' = -\frac{\partial H}{\partial \boldsymbol{q}}$$

or equivalently, with $\boldsymbol{y} = \big(\boldsymbol{q}, \boldsymbol{p}\big)^T$

$$\boldsymbol{y}' = J\nabla H , \qquad J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$$

where H is the Hamiltonian (scalar) function.

The state variables in the Hamiltonian form of the ODE system (4.2.2) are

$$\boldsymbol{q} = \begin{pmatrix} x \\ y \end{pmatrix} , \qquad \boldsymbol{p} = \begin{pmatrix} x' - y \\ y' + x \end{pmatrix}$$

so that the system is given by

$$\boldsymbol{q}' = \begin{pmatrix} p_1 + q_2 \\ p_2 - q_1 \end{pmatrix} \tag{4.3.5a}$$

$$\boldsymbol{p}' = \begin{pmatrix} p_2 - \beta\frac{q_1+\mu}{R_1^3} - \mu\frac{q_1-\beta}{R_2^3} \\ -p_1 - \frac{\beta q_2}{R_1^3} - \frac{\mu q_2}{R_2^3} \end{pmatrix} \tag{4.3.5b}$$

[1] A Hamiltonian system is symplectic by nature

with the Hamiltonian function

$$H(\boldsymbol{q},\boldsymbol{p}) = \tfrac{1}{2}\big(p_1^2 + p_2^2\big) + q_2 p_1 - q_1 p_2 - \frac{\beta}{R_1} - \frac{\mu}{R_2} \qquad , \tag{4.3.6}$$

with $R_i(\boldsymbol{q})$ as in (4.2.1).

The Hamiltonian system is very close to the Newtonian considered before, the invariant H is actually the same as the Jacobi Integral I, and the ODE system (4.3.5) differs only with the factor 2 on the derivatives x', y' in $\boldsymbol{p}'$. So it can be expected that these two systems behaves much alike.

4.4 Results

This problem appeared less useful for the evaluation of the different approaches than first expected, since the different methods performed more or less equally well in terms of error-estimate over the periods, in reference to the specific integration method.
However there are some remarks to be made.

Instead of examine the error over each period, tests on convergence order for the methods is applied. The convergence measure is taken to be the mean of error-estimates over the integration interval, i.e. ten times an orbit-period.

The explicit method with no additional initiatives diverges, or else very small stepsizes are needed. The stepsizes used for testing (5e-4 $\leq h \leq$ 2e-3) was too large to obtain convergence for the explicit method, but in this case, solving the least squares problem or stabilizing the solution ensures convergence. The use of the mentioned function `fmincon` can not be recommended in terms of calculation time, the improvement obtained is small compared to poststabilizing. In figure 4.4, the mean error is shown for different stepsizes ($h =$ 2e-3), where the least squares method only is included for $h/2$, the results for the stabilization techique is marked with crosses ($\times$).

The implicit midpoint rule performs very well though, it is not clear in figure 4.4, but the mean error of this method is similar when applied to the Newtonian and the Hamiltonian version of the problem. The solution to the Hamiltonian system is named `gls` in figure 4.4, in order to emphasize the symplecticity.
The close resemblance of the Newtonian and Hamiltonian formulation is probably the reason why the IMR method does so well in both cases. The Newtonian formulation is not

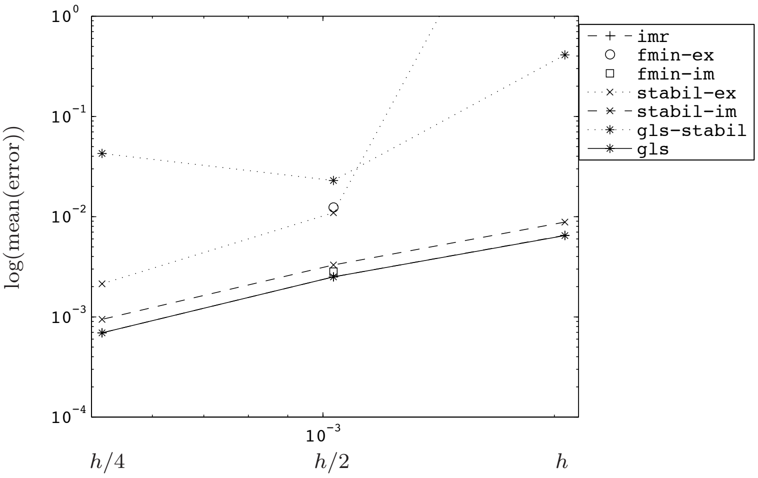


Figure 4.4: Convergence of different computations

an symplectic transformation, but not very far from the symplectic, Hamiltonian system (4.3.5).
It should be mentioned that in terms of overall position, the Hamiltonian formulation behaves exactly like the Newtonian with the same stepsize, the orbits are identical.

An odd behavior with regards to the value of the invariant during the integration was observed, which were common to all the used numerical methods. The Jacobi Integral drops immediately to a lower level (about 5 % lower than the initial value), where its stays the majority of the integration time. Except for passages near the planets, where dips occur because of the reciprocal distances. The phenomenon can be seen in figure 4.5. This behavior and the magnitude of deviation is the same whether it is the Newtonian or the Hamiltonian system, which isn't a big surprise since the invariants are the same, but it was anticipated that the symplectic property of the Hamiltonian system along with a symplectic solver, would improve the invariant conservation.

Since the solution to the Hamiltonian formulation has the same deviation, magnitude and pattern, in regards to its invariant, as the different other methods applied to the Newtonian formulation has, one might think that it was possible to improve the invariant conservation by stabilizing the system the same way as in (4.3.3). So a term equivalent to (4.3.3) was introduced in (4.3.5). And this improves conservation of the Hamiltonian significantly – see figure 4.6(a) compared to figure 4.5, and yes, the scales are the same – but introducing this stabilization, the solution orbit deteriorates. It doesn't diverge within the used integration

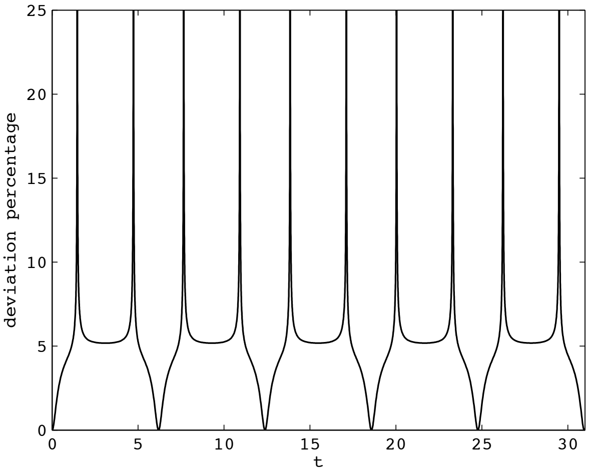


Figure 4.5: Deviation of Jacobi Integral (4.2.3) in percent from initial value.

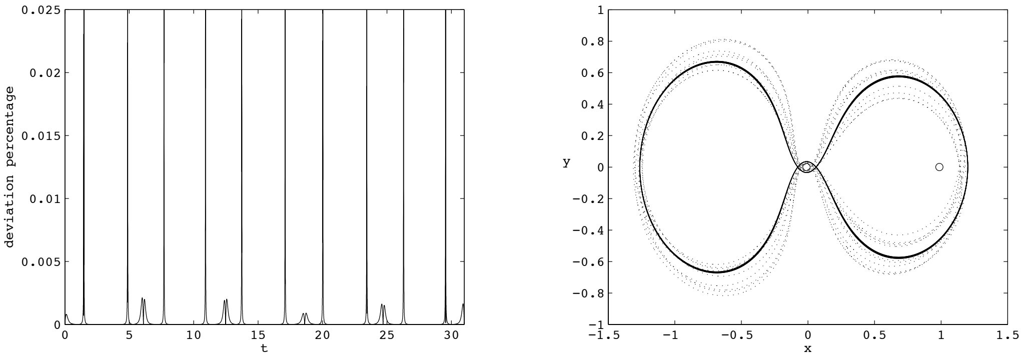
time (ten periods), but it doesn't look as good. The drift in x-coordinate i.e. the error-estimate (4.3.1) is worse, this is shown in figure 4.4 as `gls-stabil`.

The solution orbit computed with the midpoint rule and with the stabilization function applied to it can be seen in figure 4.6(b). And what is even more strange is that the passages where the invariant deviation goes to zero, the orbit deviates the most, which is in contrast to the influence of the stabilizing term which should vanish on the constraint manifold.

The reason for this deterioration in position-solution remains an unanswered question at the end of this project.

Concluding remarks

For this specific problem, none of the initiatives seems worthwhile if it is invariant conservation that is sougt. For the ODE solution, the best option seems either to do nothing or to reformulate it into a Hamiltonian system and use this model along with an symplectic solver. In this report, an implicit solver was used which gave good results, but if one wants to avoid solving the nonlinear system, it could be an idea to use an explicit, symplectic, possibly of higher order method and apply the stabilization technique, which seems a reasonable price for obtaining a better converging solution.



(a) Percentage deviation from initial invariant value H_0.

(b) Solution orbit over 10 periods, full line IMR, dotted line IMR with stabilization.

Figure 4.6: Effect of stabilization of Hamiltonian system.

Chapter 5

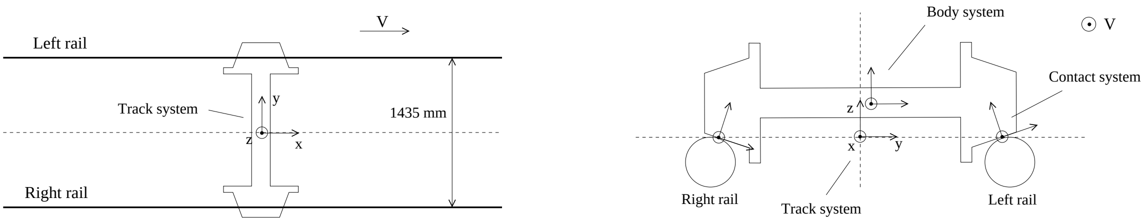
Wheel–Rail Contact Model

by : Mark Hoffmann

5.1 Introduction

In this project a single wheelset with conical profiles is simulated on a straight and level track. The wheel–rail contact is considered to be a rigid constraint, which eventually yields a differential–algebraic equation system (DAE) describing the motion of the wheelset. The model is implemented in Matlab and a detailed description of the solution process is given.

5.2 Multibody System



(a) Top view

(b) Front view

Figure 5.1: Wheelset on a straight track

The German `BA004` [2] wheelset is simulated. The mass is 1032 kg and the moments of inertia are $I_{xx} = I_{zz} = 529.8$ kg m^2 and $I_{yy} = 89.9$ kg m^2. The nominal rolling radius is $r_0 = 0.46$ m. The lateral distance from the center of mass to the circular wheel section having radius r_0 is 0.75 m. The track gauge is 1435 mm, which is measured 14 mm below the top of the rails.

The multibody system is illustrated in Figure 5.1. Three bodies are considered : 1) Wheelset 2) Left rail 3) Right rail. The motion of the wheelset is defined relative to the track system, which moves with the constant speed V along the track. Furthermore, a body system is defined for the wheelset. The origin is in the center of mass and the axes are aligned with the principal axes of the wheelset. The body system is obtained from the track system by two successive rotations : 1) ψ (yaw) around the vertical axis 2) ϕ (roll) around the longitudinal axis. For the computation of the tangential contact forces it is convenient to define a contact coordinate system for each contact point (see Figure 5.1 (b)). The origin is in the contact point and the axes are obtained by rotating the body system around its longitudinal axis into the contact plane. All defined coordinate systems are right hand systems.

For simplicity and in order to avoid multiple contacts on a single wheel it is assumed that the wheel profiles are conical and the rail profiles are circular (see Figure 5.1). The contact is also assumed to be two dimensional and hence the yaw angle is neglected in the computation of the normal contact forces. The conicity of the wheel is δ and the radius of the circular rails is $R = 0.21$ m.

The profiles are described in parametric form by $\bar{\mathbf{u}}^{ik}$, where i and k are body and contact numbers, respectively. This numbering is given in Table 5.1. For each profile, the vector $\bar{\mathbf{u}}^{ik}$ refers to a local system, i.e. the wheel profiles are defined in the body system and the rail profiles are defined in the track system. This yields the simplest (and most natural) representation of the profiles. The wheel profile parameter measures the lateral distance w.r.t. to a reference point at r_0, and the rail profile parameter measures the angle with vertical (see Figure 5.2). Using this strategy the profiles are found to be

$$\bar{\mathbf{u}}^{11} = \begin{bmatrix} y_0^{11} + s^{11} \\ -r_0 + s^{11}\tan\delta \end{bmatrix} \quad \bar{\mathbf{u}}^{12} = \begin{bmatrix} y_0^{12} + s^{12} \\ -r_0 - s^{12}\tan\delta \end{bmatrix}$$

$$\bar{\mathbf{u}}^{21} = \begin{bmatrix} y_0^{21} - R\sin s^{21} \\ R(\cos s^{21} - 1) \end{bmatrix} \quad \bar{\mathbf{u}}^{32} = \begin{bmatrix} y_0^{32} - R\sin s^{32} \\ R(\cos s^{32} - 1) \end{bmatrix}$$

where

$$y_0^{11} = 0.75 \text{ m} \qquad y_0^{12} = -0.75 \text{ m} \qquad y_0^{21} = 0.79 \text{ m} \qquad y_0^{32} = -0.79 \text{ m}$$

	1	**2**	**3**
i (body index)	Wheelset	Left rail	Right rail
k (contact index)	Left contact point	Right contact point	–

Table 5.1: Body and contact numbering

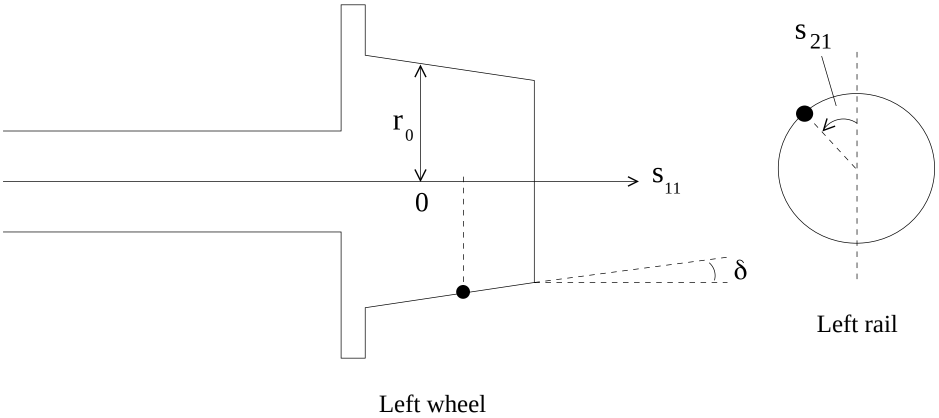


Figure 5.2: Profile parameters (front view)

5.3 Equations of Motion

The motion of any multibody system can be found using Newton–Euler equations [5, 6, 7]. For a single wheelset with fixed speed on a straight track the equations of motion are

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{q}} &= \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}) \\ \dot{\beta} &= h(\mathbf{q}, \dot{\mathbf{q}}) \end{aligned} \tag{5.3.1}$$

$$\begin{aligned} \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}) &= \begin{bmatrix} F_y \\ F_z \\ I_{yy}\dot{\psi}V/r_0 + M_x \\ -I_{yy}\dot{\phi}V/r_0 + M_z \end{bmatrix} \\ h(\mathbf{q}, \dot{\mathbf{q}}) &= M_y/I_{yy} \end{aligned}$$

The position and orientation of the wheelset is defined by $\mathbf{q} = [y, z, \phi, \psi]^T$. The x–coordinate is fixed relative to the track system, hence the wheelset is constrained to move with the speed V along the track. This yields an energy input and is a simple way to provide a steady motion of the wheelset. β is called the spin perturbation and defines the difference between the actual spin and the nominal value of V/r_0. It is seen that the wheelset has 5 degrees of freedom. The mass matrix is $\mathbf{M} = \mathrm{diag}(m, m, I_{xx}, I_{zz})$ and the external forces and moments affecting the wheelset are $\mathbf{F}_{\mathrm{ext}} = [F_x, F_y, F_z]^T$ and $\mathbf{M}_{\mathrm{ext}} = [M_x, M_y, M_z]^T$.

5.4 Wheel–Rail Constraints

By imposing two rail constraints the number of degrees of freedom is reduced. The position of the wheelset is still defined by $\mathbf{q} = [y, z, \phi, \psi]^T$, however, these coordinates are only descriptor variables and not the degrees of freedom because y, z and ϕ are connected through the wheel–rail constraints. Two conditions are imposed for each wheel–rail contact

1. The contact point on the wheel and rail should be identical in space (contact point constraint)
2. The normal vectors to the wheel and rail profiles are aligned (orientation constraint)

These two conditions are written in the contact constraint equation

$$\mathbf{C}(\mathbf{q}, \mathbf{s}) = \begin{bmatrix} \mathbf{R}^1 + \mathbf{A}^1\bar{\mathbf{u}}^{11} - \bar{\mathbf{u}}^{21} \\ \mathbf{R}^1 + \mathbf{A}^1\bar{\mathbf{u}}^{12} - \bar{\mathbf{u}}^{32} \\ \bar{\mathbf{n}}^{21T}\mathbf{A}^1\bar{\mathbf{t}}^{11} \\ \bar{\mathbf{n}}^{32T}\mathbf{A}^1\bar{\mathbf{t}}^{12} \end{bmatrix} = \mathbf{0}^{6\times1} \tag{5.4.1}$$

where

$$\bar{\mathbf{t}}^{ik} = (\bar{\mathbf{u}}^{ik})' \qquad \bar{\mathbf{n}}^{ik} = \hat{\bar{\mathbf{t}}}^{ik} = (\hat{\bar{\mathbf{u}}}^{ik})'$$

and

$$\mathbf{q} = \begin{bmatrix} y \\ z \\ \phi \end{bmatrix} \qquad \mathbf{R}^1 = \begin{bmatrix} y \\ r_0 + z \end{bmatrix} \qquad \mathbf{A}^1 = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}$$

The profiles are described by the four parameters $\mathbf{s} = [s^{11}, s^{21}, s^{12}, s^{32}]^T$. These parameters are determined through the contact constraint equation (5.4.1). Since there are four unknown parameters and six constraint equations the number of degrees of freedom are reduced by two, that is $z(y)$ and $\phi(y)$.

5.5 DAE Formulation

The normal contact forces are taken into account by imposing *Lagrange multipliers* to the equations of motion [3, 4, 5, 7]

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}_{\mathbf{q}}^T\boldsymbol{\lambda} = \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{s}, \dot{\mathbf{s}}, \boldsymbol{\lambda}) \tag{5.5.1}$$

where $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers. The equations of motion given in equation (5.5.1) together with the contact constraint equation (5.4.1) is a DAE system of differential index 3 [29] and not directly solvable. To come around this problem a reformulation is presented using the augmented Lagrangian form [4]. The contact constraint equation is differentiated twice w.r.t. time (index reduction)

$$\begin{aligned} \mathbf{C}(\mathbf{q},\mathbf{s}) &= \mathbf{0} & (5.5.2)\\ \mathbf{C}_\mathbf{q}\dot{\mathbf{q}} + \mathbf{C}_\mathbf{s}\dot{\mathbf{s}} &= \mathbf{0} & (5.5.3)\\ \mathbf{C}_\mathbf{q}\ddot{\mathbf{q}} + \mathbf{C}_\mathbf{s}\ddot{\mathbf{s}} &= -(\mathbf{C}_\mathbf{q}\dot{\mathbf{q}})_\mathbf{q}\dot{\mathbf{q}} - (\mathbf{C}_\mathbf{q}\dot{\mathbf{q}})_\mathbf{s}\dot{\mathbf{s}} - (\mathbf{C}_\mathbf{s}\dot{\mathbf{s}})_\mathbf{q}\dot{\mathbf{q}} - (\mathbf{C}_\mathbf{s}\dot{\mathbf{s}})_\mathbf{s}\dot{\mathbf{s}} =: \mathbf{Q}_d & (5.5.4) \end{aligned}$$

Since the normal contact forces are defined to be acting normal to the constraint manifold it is required that [4]

$$\mathbf{C}_\mathbf{s}^T\boldsymbol{\lambda} = \mathbf{0} \qquad (5.5.5)$$

Combining equation (5.5.1), (5.5.4) and (5.5.5) it is found that

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} & \mathbf{C}_\mathbf{q}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_\mathbf{s}^T \\ \mathbf{C}_\mathbf{q} & \mathbf{C}_\mathbf{s} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \ddot{\mathbf{s}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}(\mathbf{q},\dot{\mathbf{q}},\mathbf{s},\dot{\mathbf{s}},\boldsymbol{\lambda}) \\ \mathbf{0} \\ \mathbf{Q}_d(\mathbf{q},\dot{\mathbf{q}},\mathbf{s},\dot{\mathbf{s}}) \end{bmatrix} \quad , \quad \begin{aligned} &\mathbf{M} = \mathrm{diag}(m,m,I_{xx}) \\ &\mathbf{q} = [y,z,\phi]^T \\ &\mathbf{s} = [s^{11},s^{21},s^{12},s^{32}]^T \end{aligned} \qquad (5.5.6)$$

This equation can be solved in order to determine the accelerations and the Lagrange multipliers. Note that the right hand side is dependent on the Lagrange multipliers because the tangential contact forces depends nonlinearly on the normal load. Thus equation (5.5.6) is nonlinear in $\boldsymbol{\lambda}$.

In order to solve equation (5.5.6) it is necessary to know $\mathbf{C}_\mathbf{q}$, $\mathbf{C}_\mathbf{s}$ and $\mathbf{Q}_d$. It is seen that $\mathbf{A}_\phi^1\mathbf{v} = \mathbf{A}^1\hat{\mathbf{v}}$ where $\hat{\mathbf{v}} = [-v_2, v_1]^T$. Utilizing this information it is found that

$$\mathbf{C}_\mathbf{q} = \begin{bmatrix} \mathbf{I}^{2\times2} & \mathbf{A}^1\hat{\bar{\mathbf{u}}}^{11} \\ \mathbf{I}^{2\times2} & \mathbf{A}^1\hat{\bar{\mathbf{u}}}^{12} \\ \mathbf{0}^{1\times2} & \bar{\mathbf{n}}^{21T}\mathbf{A}^1\bar{\mathbf{n}}^{11} \\ \mathbf{0}^{1\times2} & \bar{\mathbf{n}}^{32T}\mathbf{A}^1\bar{\mathbf{n}}^{12} \end{bmatrix}$$

$$\mathbf{C}_\mathbf{s} = \begin{bmatrix} \mathbf{A}^1\bar{\mathbf{t}}^{11} & -\bar{\mathbf{t}}^{21} & \mathbf{0}^{2\times1} & \mathbf{0}^{2\times1} \\ \mathbf{0}^{2\times1} & \mathbf{0}^{2\times1} & \mathbf{A}^1\bar{\mathbf{t}}^{12} & -\bar{\mathbf{t}}^{32} \\ \bar{\mathbf{n}}^{21T}\mathbf{A}^1(\bar{\mathbf{t}}^{11})' & \bar{\mathbf{t}}^{11T}\mathbf{A}^{1T}(\bar{\mathbf{n}}^{21})' & 0 & 0 \\ 0 & 0 & \bar{\mathbf{n}}^{32T}\mathbf{A}^1(\bar{\mathbf{t}}^{12})' & \bar{\mathbf{t}}^{12T}\mathbf{A}^{1T}(\bar{\mathbf{n}}^{32})' \end{bmatrix}$$

$$(\mathbf{C}_\mathbf{q}\dot{\mathbf{q}})_\mathbf{q}\dot{\mathbf{q}} = \begin{bmatrix} -\mathbf{A}^1\bar{\mathbf{u}}^{11}\dot{\phi}^2 \\ -\mathbf{A}^1\bar{\mathbf{u}}^{12}\dot{\phi}^2 \\ -\bar{\mathbf{n}}^{21T}\mathbf{A}^1\bar{\mathbf{t}}^{11}\dot{\phi}^2 \\ -\bar{\mathbf{n}}^{32T}\mathbf{A}^1\bar{\mathbf{t}}^{12}\dot{\phi}^2 \end{bmatrix}$$

$$(\mathbf{C_q\dot{q}})_\mathbf{s}\mathbf{\dot{s}} = (\mathbf{C_s\dot{s}})_\mathbf{q}\mathbf{\dot{q}} = \begin{bmatrix} \mathbf{A}^1\bar{\mathbf{n}}^{11}\dot{\phi}\dot{s}^{11} \\ \mathbf{A}^1\bar{\mathbf{n}}^{12}\dot{\phi}\dot{s}^{12} \\ \bar{\mathbf{n}}^{21T}\mathbf{A}^1(\bar{\mathbf{n}}^{11})'\dot{\phi}\dot{s}^{11} + \bar{\mathbf{n}}^{11T}\mathbf{A}^{1T}(\bar{\mathbf{n}}^{21})'\dot{\phi}\dot{s}^{21} \\ \bar{\mathbf{n}}^{32T}\mathbf{A}^1(\bar{\mathbf{n}}^{12})'\dot{\phi}\dot{s}^{12} + \bar{\mathbf{n}}^{12T}\mathbf{A}^{1T}(\bar{\mathbf{n}}^{32})'\dot{\phi}\dot{s}^{32} \end{bmatrix}$$

$$(\mathbf{C_s\dot{s}})_\mathbf{s}\mathbf{\dot{s}} = \begin{bmatrix} \mathbf{A}^1(\bar{\mathbf{t}}^{11})'(\dot{s}^{11})^2 - (\bar{\mathbf{t}}^{21})'(\dot{s}^{21})^2 \\ \mathbf{A}^1(\bar{\mathbf{t}}^{12})'(\dot{s}^{12})^2 - (\bar{\mathbf{t}}^{32})'(\dot{s}^{32})^2 \\ \bar{\mathbf{n}}^{21T}\mathbf{A}^1(\bar{\mathbf{t}}^{11})''(\dot{s}^{11})^2 + 2(\bar{\mathbf{n}}^{21T})'\mathbf{A}^1(\bar{\mathbf{t}}^{11})'\dot{s}^{11}\dot{s}^{21} + \bar{\mathbf{t}}^{11T}\mathbf{A}^{1T}(\bar{\mathbf{n}}^{21})''(\dot{s}^{21})^2 \\ \bar{\mathbf{n}}^{32T}\mathbf{A}^1(\bar{\mathbf{t}}^{12})''(\dot{s}^{12})^2 + 2(\bar{\mathbf{n}}^{32T})'\mathbf{A}^1(\bar{\mathbf{t}}^{12})'\dot{s}^{12}\dot{s}^{32} + \bar{\mathbf{t}}^{12T}\mathbf{A}^{1T}(\bar{\mathbf{n}}^{32})''(\dot{s}^{32})^2 \end{bmatrix}$$

For conical wheel profiles the following is valid

$$(\bar{\mathbf{t}}^{11})' = (\bar{\mathbf{t}}^{12})' = (\bar{\mathbf{t}}^{11})'' = (\bar{\mathbf{t}}^{12})'' = (\bar{\mathbf{n}}^{11})' = (\bar{\mathbf{n}}^{12})' = (\bar{\mathbf{n}}^{11})'' = (\bar{\mathbf{n}}^{12})'' = \mathbf{0} \tag{5.5.7}$$

By inserting $\bar{\mathbf{t}}^{ik} = (\bar{\mathbf{u}}^{ik})'$, $\bar{\mathbf{n}}^{ik} = \hat{\bar{\mathbf{t}}}^{ik} = (\hat{\bar{\mathbf{u}}}^{ik})'$ and equation (5.5.7) it is found

$$\mathbf{C(q,s)} = \begin{bmatrix} \mathbf{R}^1 + \mathbf{A}^1\bar{\mathbf{u}}^{11} - \bar{\mathbf{u}}^{21} \\ \mathbf{R}^1 + \mathbf{A}^1\bar{\mathbf{u}}^{12} - \bar{\mathbf{u}}^{32} \\ (\hat{\bar{\mathbf{u}}}^{21T})'\mathbf{A}^1(\bar{\mathbf{u}}^{11})' \\ (\hat{\bar{\mathbf{u}}}^{32T})'\mathbf{A}^1(\bar{\mathbf{u}}^{12})' \end{bmatrix}$$

$$\mathbf{C_q} = \begin{bmatrix} \mathbf{I}^{2\times2} & \mathbf{A}^1\hat{\bar{\mathbf{u}}}^{11} \\ \mathbf{I}^{2\times2} & \mathbf{A}^1\hat{\bar{\mathbf{u}}}^{12} \\ \mathbf{0}^{1\times2} & (\hat{\bar{\mathbf{u}}}^{21T})'\mathbf{A}^1(\hat{\bar{\mathbf{u}}}^{11})' \\ \mathbf{0}^{1\times2} & (\hat{\bar{\mathbf{u}}}^{32T})'\mathbf{A}^1(\hat{\bar{\mathbf{u}}}^{12})' \end{bmatrix}$$

$$\mathbf{C_s} = \begin{bmatrix} \mathbf{A}^1(\bar{\mathbf{u}}^{11})' & -(\bar{\mathbf{u}}^{21})' & \mathbf{0}^{2\times1} & \mathbf{0}^{2\times1} \\ \mathbf{0}^{2\times1} & \mathbf{0}^{2\times1} & \mathbf{A}^1(\bar{\mathbf{u}}^{12})' & -(\bar{\mathbf{u}}^{32})' \\ 0 & (\bar{\mathbf{u}}^{11T})'\mathbf{A}^{1T}(\hat{\bar{\mathbf{u}}}^{21})'' & 0 & 0 \\ 0 & 0 & 0 & (\bar{\mathbf{u}}^{12T})'\mathbf{A}^{1T}(\hat{\bar{\mathbf{u}}}^{32})'' \end{bmatrix}$$

$$(\mathbf{C_q\dot{q}})_\mathbf{q}\mathbf{\dot{q}} = \begin{bmatrix} -\mathbf{A}^1\bar{\mathbf{u}}^{11}\dot{\phi}^2 \\ -\mathbf{A}^1\bar{\mathbf{u}}^{12}\dot{\phi}^2 \\ -(\hat{\bar{\mathbf{u}}}^{21T})'\mathbf{A}^1(\bar{\mathbf{u}}^{11})'\dot{\phi}^2 \\ -(\hat{\bar{\mathbf{u}}}^{32T})'\mathbf{A}^1(\bar{\mathbf{u}}^{12})'\dot{\phi}^2 \end{bmatrix}$$

$$(\mathbf{C_q\dot{q}})_\mathbf{s}\mathbf{\dot{s}} = (\mathbf{C_s\dot{s}})_\mathbf{q}\mathbf{\dot{q}} = \begin{bmatrix} \mathbf{A}^1(\hat{\bar{\mathbf{u}}}^{11})'\dot{\phi}\dot{s}^{11} \\ \mathbf{A}^1(\hat{\bar{\mathbf{u}}}^{12})'\dot{\phi}\dot{s}^{12} \\ (\hat{\bar{\mathbf{u}}}^{11T})'\mathbf{A}^{1T}(\hat{\bar{\mathbf{u}}}^{21})''\dot{\phi}\dot{s}^{21} \\ (\hat{\bar{\mathbf{u}}}^{12T})'\mathbf{A}^{1T}(\hat{\bar{\mathbf{u}}}^{32})''\dot{\phi}\dot{s}^{32} \end{bmatrix}$$

$$(\mathbf{C_s\dot{s}})_\mathbf{s}\mathbf{\dot{s}} = \begin{bmatrix} -(\bar{\mathbf{u}}^{21})''(\dot{s}^{21})^2 \\ -(\bar{\mathbf{u}}^{32})''(\dot{s}^{32})^2 \\ (\bar{\mathbf{u}}^{11T})'\mathbf{A}^{1T}(\hat{\bar{\mathbf{u}}}^{21})'''(\dot{s}^{21})^2 \\ (\bar{\mathbf{u}}^{12T})'\mathbf{A}^{1T}(\hat{\bar{\mathbf{u}}}^{32})'''(\dot{s}^{32})^2 \end{bmatrix}$$

5.6 Forces

The external forces and moments are given by

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \mathbf{F}_g + \mathbf{F}_s + {}_T\mathbf{A}_b(\mathbf{F}_{fl} + \mathbf{F}_{fr}) + {}_T\mathbf{A}_{cl}\mathbf{F}_{cl} + {}_T\mathbf{A}_{cr}\mathbf{F}_{cr}$$

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \mathbf{R}_{cl} \times ({}_b\mathbf{A}_{cl}\mathbf{F}_{cl} + \mathbf{F}_{fl}) + \mathbf{R}_{cr} \times ({}_b\mathbf{A}_{cr}\mathbf{F}_{cr} + \mathbf{F}_{fr})$$

$\mathbf{F}_g$ is the gravitational force, $\mathbf{F}_s$ is a lateral suspension force, $\mathbf{F}_{fl}$, $\mathbf{F}_{fr}$ are flange forces, ${}_T\mathbf{A}_b$, ${}_T\mathbf{A}_{cl}, {}_T\mathbf{A}_{cr}, {}_b\mathbf{A}_{cl}, {}_b\mathbf{A}_{cr}$ are rotation matrices and $\mathbf{R}_{cl}$, $\mathbf{R}_{cr}$ are position vectors from the center of mass of the wheelset to the contact points. The gravitational force is

$$\mathbf{F}_g = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix}$$

The primary suspension is modelled by

$$\mathbf{F}_s = \begin{bmatrix} 0 \\ -k_s y \\ 0 \end{bmatrix}$$

The flange is modelled by a stiff spring with a dead band

$$\mathbf{F}_{fl} = \begin{cases} [0, -k_f(y - y_f), 0]^T & y > y_f \\ \mathbf{0} & y \leq y_f \end{cases} \qquad \mathbf{F}_{fr} = \begin{cases} [0, -k_f(y + y_f), 0]^T & y < -y_f \\ \mathbf{0} & y \geq -y_f \end{cases}$$

$k_s = 1.823$ MN/m, $k_f = 14.60$ MN/m and $y_f = 0.0091$ m. These value are from Cooperrider's bogie [10, 11]. The rotation matrices are

$$\begin{aligned}
{}_T\mathbf{A}_b &= \begin{bmatrix} \cos\psi & -\sin\psi\cos\phi & \sin\psi\sin\phi \\ \sin\psi & \cos\psi\cos\phi & -\cos\psi\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \\
{}_T\mathbf{A}_{cl} &= \begin{bmatrix} \cos\psi & -\sin\psi\cos(\phi+\delta) & \sin\psi\sin(\phi+\delta) \\ \sin\psi & \cos\psi\cos(\phi+\delta) & -\cos\psi\sin(\phi+\delta) \\ 0 & \sin(\phi+\delta) & \cos(\phi+\delta) \end{bmatrix} \\
{}_T\mathbf{A}_{cr} &= \begin{bmatrix} \cos\psi & -\sin\psi\cos(\phi-\delta) & \sin\psi\sin(\phi-\delta) \\ \sin\psi & \cos\psi\cos(\phi-\delta) & -\cos\psi\sin(\phi-\delta) \\ 0 & \sin(\phi-\delta) & \cos(\phi-\delta) \end{bmatrix} \\
{}_b\mathbf{A}_{cl} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\delta) & -\sin(\delta) \\ 0 & \sin(\delta) & \cos(\delta) \end{bmatrix} \\
{}_b\mathbf{A}_{cl} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\delta) & \sin(\delta) \\ 0 & -\sin(\delta) & \cos(\delta) \end{bmatrix}
\end{aligned}$$

The position of the contact point are found using the profile parameters

$$\begin{aligned}
\mathbf{R}_{cl} &= \begin{bmatrix} 0 \\ y_0^{11} + s^{11} \\ -r_0 + s^{11}\tan\delta \end{bmatrix} \\
\mathbf{R}_{cr} &= \begin{bmatrix} 0 \\ y_0^{12} + s^{12} \\ -r_0 - s^{12}\tan\delta \end{bmatrix}
\end{aligned}$$

The normal forces are found using the Lagrange multipliers

$$N_l = \sqrt{\lambda_1^2 + \lambda_2^2} \qquad N_r = \sqrt{\lambda_3^2 + \lambda_4^2}$$

The tangential contact forces (creep forces) depends on the relative velocity between the wheel and rail (creep). The creep is computed using the following approximations.

$$\begin{aligned}
\xi_{x,cl} &\approx 1 + ((V/r_0 + \beta)\bar{u}_2^{11} - \dot{\psi}\bar{u}_1^{11})/V \\
\xi_{x,cr} &\approx 1 + ((V/r_0 + \beta)\bar{u}_2^{12} - \dot{\psi}\bar{u}_1^{12})/V \\
\xi_{y,cl} &\approx (-\psi V + \dot{y} - \dot{\phi}\bar{u}_2^{11})/(V\cos\delta) \\
\xi_{y,cr} &\approx (-\psi V + \dot{y} - \dot{\phi}\bar{u}_2^{12})/(V\cos\delta) \\
\xi_{s,cl} &\approx (\dot{\psi}\cos\delta - (V/r_0 + \beta)\sin\delta)/V \\
\xi_{s,cr} &\approx (\dot{\psi}\cos\delta + (V/r_0 + \beta)\sin\delta)/V
\end{aligned}$$

The creep forces T_x and T_y are calculated using the nonlinear model proposed by Shen–Hedrick–Elkins [9]

$$\begin{aligned}
T_x &= \epsilon\tilde{F}_x \\
T_y &= \epsilon\tilde{F}_y
\end{aligned}$$

where

$$a = a_0(N/N_0)^{1/3} \qquad b = b_0(N/N_0)^{1/3}$$

$$\begin{aligned}
\tilde{F}_x &= -abGC_{11}\xi_x \\
\tilde{F}_y &= -abG\left(C_{22}\xi_y + \sqrt{ab}C_{23}\xi_s\right)
\end{aligned}$$

$$|\tilde{\mathbf{F}}| = \sqrt{\tilde{F}_x^2 + \tilde{F}_y^2}$$

$$|\mathbf{F}| = \begin{cases} \mu N\left(u - \frac{1}{3}u^2 + \frac{1}{27}u^3\right) & u < 3 \\ \mu N & u \geq 3 \end{cases} \quad , \quad u = \frac{|\tilde{\mathbf{F}}|}{\mu N}$$

$$\epsilon = |\mathbf{F}|/|\tilde{\mathbf{F}}|$$

and

$$\begin{aligned}
a_0 &= 2.8134 \text{ mm} \\
b_0 &= 1.6745 \text{ mm} \\
N_0 &= 5073 \text{ N} \\
C_{11} &= 4.8530 \\
C_{22} &= 4.5548 \\
C_{23} &= 2.2666 \\
G &= 8.27 \cdot 10^{10} \text{ N/m}^2 \\
\nu &= 0.27 \\
\mu &= 0.30
\end{aligned}$$

where C_{11}, C_{22}, C_{23} are Kalker's creepage coefficients [12]. The contact forces are

$$\mathbf{F}_{cl} = [T_{x,cl}, T_{y,cl}, N_l]^T \qquad \mathbf{F}_{cr} = [T_{x,cr}, T_{y,cr}, N_r]^T$$

It should be noted that the term $\mathbf{Q}$ in equation (5.5.6) only includes the tangential contact forces because the normal forces are taken into account using the Lagrange multipliers.

5.7 Numerical Integration

1. *Initial condition* : The coordinates $y_0, \psi_0, \dot{y}_0, \dot{\psi}_0, \beta_0$ are specified. The dependent coordinates $\mathbf{q}_{d,0} = [z_0, \phi_0]^T$, $\dot{\mathbf{q}}_{d,0} = [\dot{z}_0, \dot{\phi}_0]^T$, $\mathbf{s}$ and $\dot{\mathbf{s}}$ are determined by solving equation (5.5.2) and (5.5.3). Equation (5.5.2) is solved using Newton–Raphson's method (see e.g. [8]). From a simple geometric consideration the following initial value is found appropriate

$$\begin{bmatrix} z \\ \phi \\ s^{11} \\ s^{21} \\ s^{12} \\ s^{32} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ (y_0^{21} - y_0^{11}) - R\sin\delta \\ \delta \\ (y_0^{32} - y_0^{12}) + R\sin\delta \\ -\delta \end{bmatrix}$$

and the Jacobi matrix is

$$\mathbf{J} = [\mathbf{C}_{\mathbf{q}_d}, \mathbf{C}_{\mathbf{s}}] \qquad , \qquad \mathbf{q}_d = [z, \phi]^T$$

Equation (5.5.3) is solved by exploiting the linearity, i.e.

$$[\mathbf{C}_{\mathbf{q}_d}, \mathbf{C}_{\mathbf{s}}] \begin{bmatrix} \dot{\mathbf{q}}_d \\ \dot{\mathbf{s}} \end{bmatrix} = -\mathbf{C}_{\mathbf{q}_i}\dot{\mathbf{q}}_i \qquad , \qquad \begin{array}{c} \mathbf{q}_i = y \\ \mathbf{q}_d = [z, \phi]^T \end{array}$$

2. Equation (5.5.6) is solved using fixed–point iteration. This gives the current value of $\ddot{\mathbf{q}}$ and $\boldsymbol{\lambda}$. The static load is used as a first initial guess, however, during simulation the previous value gives a better estimate. From the static load it is found

$$\boldsymbol{\lambda}_0 = \left[\frac{mg}{2}\tan\delta, -\frac{mg}{2}, -\frac{mg}{2}\tan\delta, -\frac{mg}{2}, 0, 0\right]^T$$

3. The independent accelerations are integrated one step forward in time
4. The new value of the dependent coordinates are found through solving equation (5.5.2) and (5.5.3)
5. Repeat step 2. to 4. until the end of the simulation is reached

5.8 Matlab

In order to integrate the DAE system in Matlab the following formulation of the system is presented

$$\mathbf{y} = [y, \psi, \dot{y}, \dot{\psi}, \beta, z, \phi, s^{11}, s^{21}, s^{12}, s^{32}]^T$$

$$\mathbf{M}\dot{\mathbf{y}} = \begin{cases} y_3 \\ y_4 \\ \ddot{y} \qquad \text{(known from equation (5.5.6))} \\ (-I_{yy}\dot{\phi}V/r_0 + M_z)/I_{zz} \\ M_y/I_{yy} \\ \mathbf{C}(\mathbf{q}, \mathbf{s}) \end{cases}$$

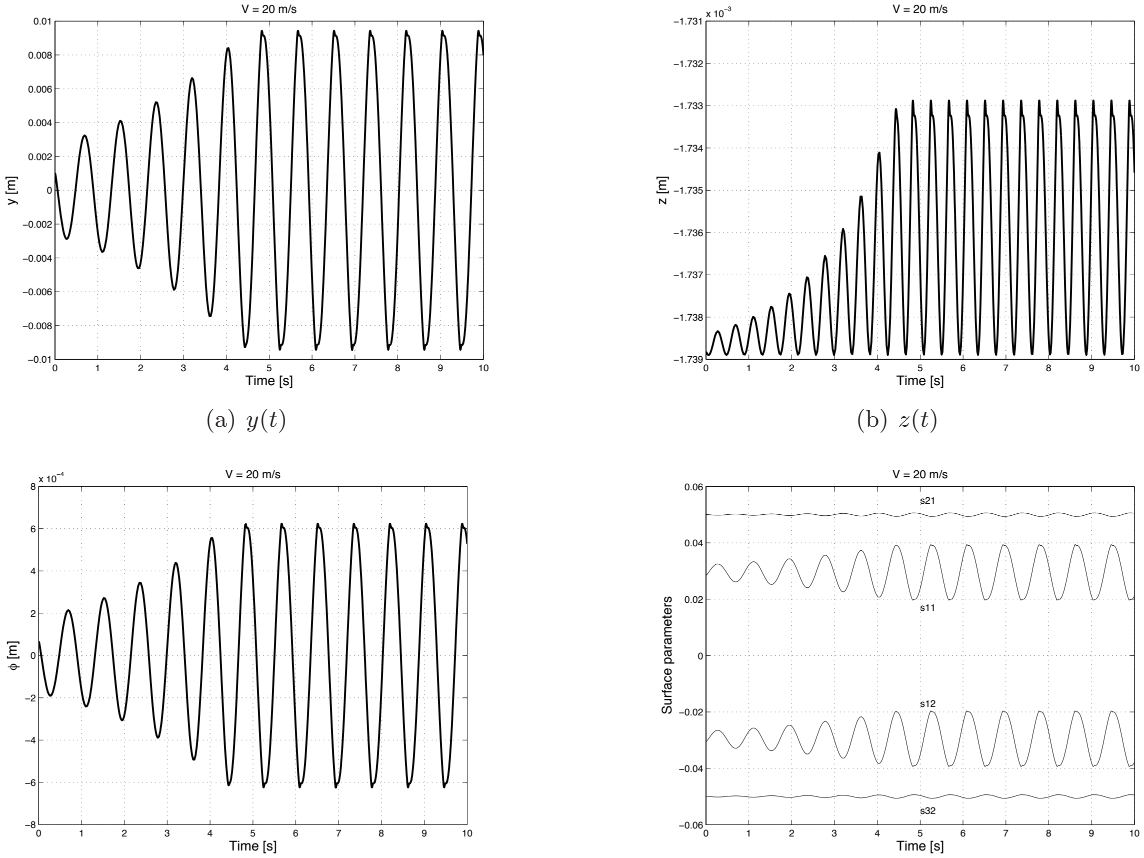
where

$$\mathbf{q} = [y_1, y_6, y_7]^T \qquad \mathbf{s} = [y_8, y_9, y_{10}, y_{11}]^T$$

and $\mathbf{M}$ is the *singular* mass matrix providing zeros on the left hand side in the 6 last equations. The dependencies of the right hand side function are important for an efficient computation of the Jacobi matrix, which is needed when using an implicit numerical integrator. The dependencies are given in $\mathbf{J}_{\text{pattern}}$. The system is integrated using the Matlab solver `ode15s`.

$$\mathbf{M} = \left[\begin{array}{ccccccccccc} 1&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0 \\ \hline 0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0 \end{array}\right] \mathbf{J}_{\text{pattern}} = \left[\begin{array}{ccccccccccc} 0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0 \\ 1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1 \\ 1&0&0&0&0&0&1&1&1&0&0 \\ 0&0&0&0&0&1&1&1&1&0&0 \\ 1&0&0&0&0&0&1&0&0&1&1 \\ 0&0&0&0&0&1&1&0&0&1&1 \\ 0&0&0&0&0&0&1&0&1&0&0 \\ 0&0&0&0&0&0&1&0&0&0&1 \end{array}\right]$$

5.9 Results



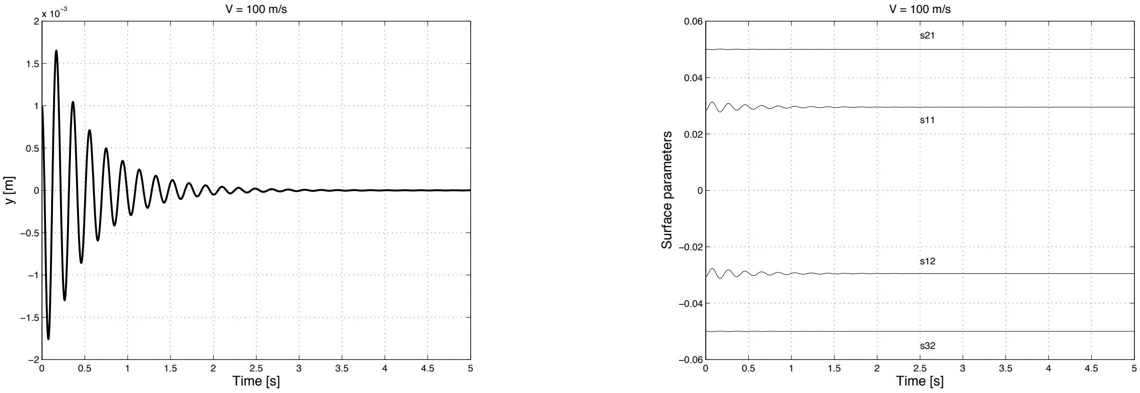
(a) $y(t)$

(b) $z(t)$

(c) $\phi(t)$

(d) $s_{ij}(t)$

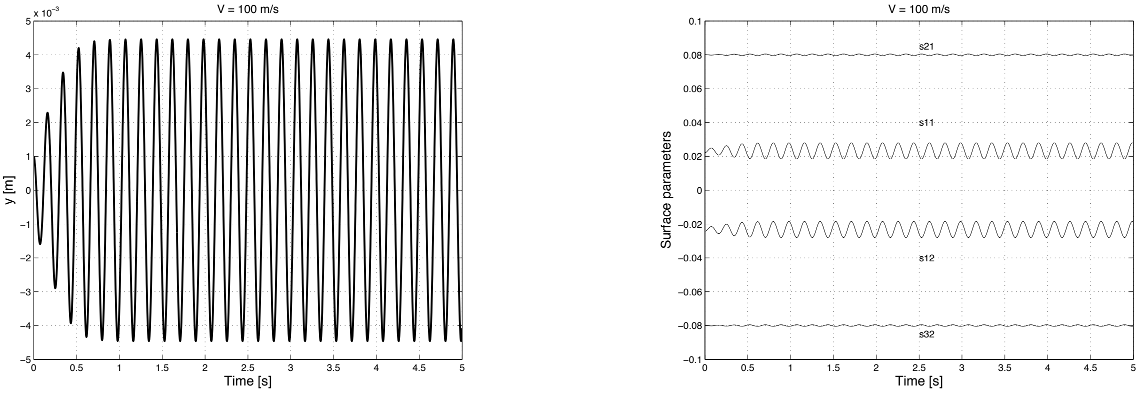
Figure 5.3: Simulation results at $V = 20$ m/s. The wheelset is not suspended. The conicity is $\delta = 0.05$. The center track solution is unstable, however, the amplitude of the oscillations are limited by the flange



(a) $y(t)$

(b) $s_{ij}(t)$

Figure 5.4: Simulation results at $V = 100$ m/s. The wheelset is suspended, $k_s = 1.823$ MN/m. The conicity is $\delta = 0.05$. The center track solution is stable. Note that $\delta + \phi = s^{21} \approx 0.05$ because the wheel profile is conical and the rails are circular. Similarly, $\delta - \phi = -s^{32} \approx 0.05$



(a) $y(t)$

(b) $s_{ij}(t)$

Figure 5.5: Simulation results at $V = 100$ m/s. The wheelset is suspended, $k_s = 1.823$ MN/m. The conicity is $\delta = 0.08$. The center track solution is unstable. Again, $\delta + \phi = s^{21} \approx 0.08$ and $\delta - \phi = -s^{32} \approx 0.08$. Figure 5.4 and Figure 5.5 illustrates that high speed trains need a low conicity to ensure stability

Chapter 6

Bicycle model.

by : **Rasmus Frank Kristensen**

6.1 Introduction

This paper deals with the stability of an uncontrolled bicycle. By uncontrolled means the stability is analyzed without any rider-gained input. In a real life situation this analysis corresponds to riding a bicycle with no hands. The dynamic system of a bicycle is interesting since a bicycle is unstable at low speeds, but stable at high speeds. The system corresponds to an inverted pendulum in the static state, which is highly unstable, but due to moments of inertia and angular momentums the system gets stable when the forward speed reaches a certain value.

Anybody who has ridden a bicycle can reconize these behaviours - for instance at a certain speed it is possible to ride the bike with no hands and yet keep the bicycle relatively easy stabilized. The opposite situation, when trying to stabilize a bicycle at very low speed requires a great amount of rider-gained input.

The theoretical basis for this model is found in a series of articles by Papadopoulos and Hand. The original basis for this paper was found in two models for a motorcycle by Sharp [26]. These models are rather complicated and are found using the numerical software AutoSim to obtain the equations of motion. The model presented here is the model derived by Hand in his MSc Thesis [25] and later simplified by Papadopoulos [24]. This model is found using the paper-pencil approach, but is later validated numerically by comparison with equations of motion found using sotfware by Papadopoulos et. al. in [23]. So though not as advanced as Sharps very real-life alike model this model displays some of the qualitative behaviours for bicycle dynamics despite the simplified equations of motion.

6.2 Mechanical Model

Figure 6.1 shows the mechanical model of the bicycle. The model consists of four rigid bodies, front frame ff, front wheel fw, rear frame rw and rear wheel rw. Each part consists of an applied mass, which are shown as filled circles in figure 6.1.

Two additional parameters are used. The pependicular distance f between the front fork and the front wheel contact point p_f determined by the angle of the steering axis. The position $(l_t,\ h_t)$ of the total mass m_t in the y and z direction respectively. The total mass is the sum of all the applied masses and the position is calculated with respect to the individual parts contribution to the total mass. The method is explained in appendix 6.8.2.

The distance f is called the *mechanical trail* of the bicycle. This parameter plays an important role in the stability of the model. A great deal of Hands thesis analyzes this parameter. The reason for the importance of the mechanical trail is that it determines how the bicycle is leaning when it is steered. A bicycle can never be stable if the mechanical trail is negative. This means the front fork must have a certain forward angle. Negative mechanical trail corresponds to the front fork is angled backwards.

The reference coordinate system is taken with the origin at the rear wheel contact point p_r. This point is later refered to as O though not appering in figure 6.1.

The wheel base w is the distance between the two contact points for the front and rear wheel, p_f and p_r respectively. The front frame has the angle α between the baseline.

Neglecting the vetical movements i the z-direction and setting the movement in the y-direction as a forward speed parameter the system consists of four degrees of freedom. A translation in the x-direction, which determines the sidewards movement. The *yaw*, which is the rotation about the z-axis at the rear contact point. The *lean*, which is the rotation about the y-axis. The last variable ψ is the applied steering and correspond to the rotation of the front frame about the steering axis.

The wheels are modeled as knife-egde wheels, which mean they act as rigid discs and the wheel-road contact is applied in the two points p_r and p_f only. The wheel-road contact modelling is assumed to with no slip in any direction. This enables the number of variables to be reduced to two. The reduced set of equations is derived in section 6.4. A discussion about these assumptions can be found in section 6.6.

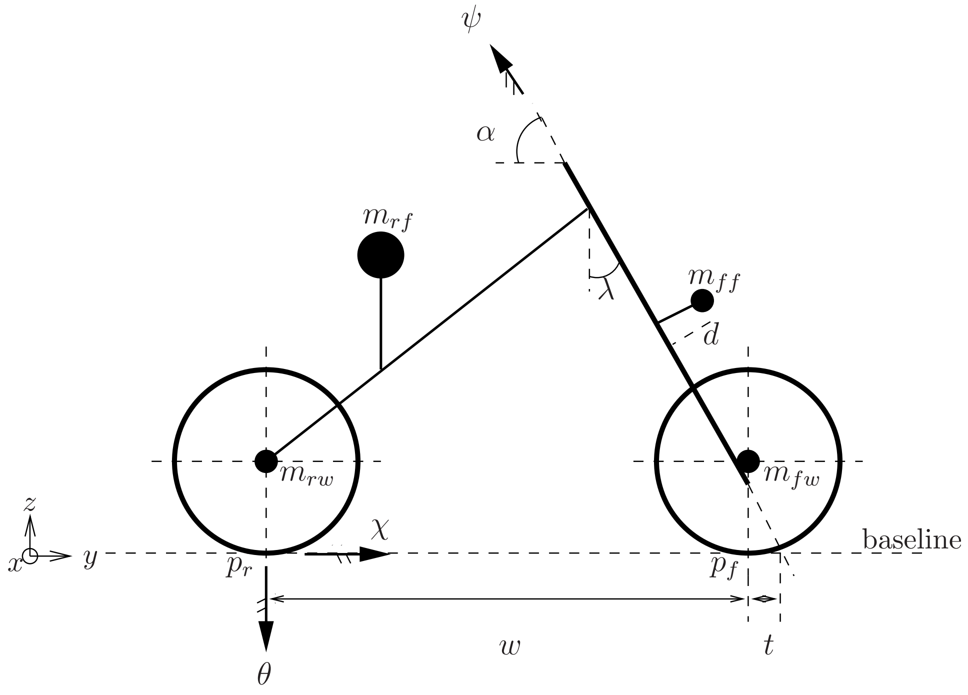


Figure 6.1: Bicycle model.

6.3 Mathematical Model

The equations of motion is derived in the following sections by the use of Lagrange's equation

$$\frac{\partial}{\partial t}\left(\frac{\partial E_{kin}}{\partial q_i}\right) - \left(\frac{\partial E_{kin}}{\partial q_i}\right) + \left(\frac{\partial E_{pot}}{\partial q_i}\right) = \mathcal{F}_i \qquad , i = 1, 2, \cdots, n \tag{6.3.1}$$

where the index notation corresponds to the particular degree of freedom. The right hand side in (6.3.1) corresponds to the external forces applied to the system. Integrating and evaluating the energy terms in (6.3.1) yields a set n (coupled) second order differential equations. The lagrangian formulation for the bicycle problem is derived in [25], but since a complete nonlinear formulation is cumbersome, and the model was intented to be analysed analytically, various linearization has been made.

The derivation of the equations of motion will not be perfomed here, so the equations presented here has the origin in Hand's Thesis. These equations are later simplified by Papadopoulos and these simplified equations are used in this paper. Though it might sound as a simple derivation of the equation of motion both authors refer this system as 'the border for the paper-pencil approach'. In the following presentation of the equations of motion the angles lean χ and steering ψ are assumed to be small.

Starting out by taking the sum of the forces action in the x-direction on the bike in figure 6.1 yields

$$m_t\ddot{x} + m_t h_t\ddot{\chi} - m_t l_t\ddot{\theta} - m_f d\ddot{\psi} = F_{xr} + F_{xf}$$

where m_t is the total mass, $m_t = m_{ff} + m_{fw} + m_{rf} + m_{rw}$.

The contributions are displacements x, steering ψ, yaw θ and lean χ. To obtain equilibrium two unknown forces F_{xr} and F_{xf} are applied at the contact points p_f and p_r for the front and rear wheel respectively. These forces will be treated in section 6.4.

The second equation corresponds to the total lean moment χ for all the applied masses.

$$m_t h_t\ddot{x} + T_{yy}\ddot{\chi} + T_{yz}\ddot{\theta} + F'_{\lambda u}\ddot{\psi} - H_t\dot{\theta} - H_f\cos\lambda\dot{\psi} = g m_t h_t\chi - g\nu\psi$$

where ν is defined as:

$$\nu = m_f d + m_t \tfrac{l_t}{w} f \tag{6.3.2}$$

l_t and h_t are the position of the total mass measured from the rear contact point p_r with respect to the y and z direction respectively.

The contributions are the χ moment to accelerate total mass m_t, angular rotations about the rear contact point p_r, T_{yy} and T_{yz}, angular acceleration about the steering axis, $F'_{\lambda u}$, gyroscopic contributions from the spinning wheels, H_t and H_f. The right hand side consists of the moment of gravity, moments of vertical front contact force due to offset from steering.

The total yaw moment θ is given by:

$$-m_t l_t\ddot{x} + T_{zy}\ddot{\psi} + T_{zz}\ddot{\theta} + F''_{\lambda z}\ddot{\psi} + H_t\dot{\chi} - H_f\sin\lambda\dot{\psi} = -wF_{xf}$$

The contributions are nearly the same as for the lean χ, angular rotations and gyroscopic moments, but about a different axis. The righthand side is the moment about the rear contact point p_r from the yet unknown contact force F_{xf} at the front wheel contact point p_f.

The last equation of motion is the contributions from steering the bicycle. The moment about the steering axis ψ.

$$-m_f d\ddot{x} + F'_{\lambda y}\ddot{\chi} + F''_{\lambda z}\ddot{\theta} + F'_{\lambda\lambda}\ddot{\psi} + H_f\left(\dot{\chi}\cos\lambda + \dot{\theta}\sin\lambda\right) = M_\psi + fF_{xf} - g\nu\chi + g(\sin\lambda)\nu\psi$$

The left hand side consists acceleration of the front frame, moments about the steering axis for angular acceleration. The right hand side consists of a applied steering moment M_{psi}, tyre-ground force on the front wheel contact point p_f and moment contributions from the actual steering of the bicycle. These terms will be adressed in section 6.4.

Setting up the four equations of motion yields:

$$\begin{aligned}
m_t\ddot{x} + m_t h_t\ddot{\chi} - m_t l_t\ddot{\theta} - m_f d\ddot{\psi} &= F_{xr} + F_{xf} && (6.3.3a)\\
m_t h_t\ddot{x} + T_{yy}\ddot{\chi} + T_{yz}\ddot{\theta} + F'_{\lambda u}\ddot{\psi} - H_t\dot{\theta} - H_f\cos\lambda\dot{\psi} &= g m_t h_t\chi - g\nu\psi && (6.3.3b)\\
-m_t l_t\ddot{x} + T_{zy}\ddot{\psi} + T_{zz}\ddot{\theta} + F''_{\lambda z}\ddot{\psi} + H_t\dot{\chi} - H_f\sin\lambda\dot{\psi} &= -wF_{xf} && (6.3.3c)\\
-m_f d\ddot{x} + F'_{\lambda y}\ddot{\chi} + F''_{\lambda z}\ddot{\theta} + F'_{\lambda\lambda}\ddot{\psi} + H_f\left(\dot{\chi}\cos\lambda + \dot{\theta}\sin\lambda\right) &= M_\psi + fF_{xf} && (6.3.3d)\\
&\quad -g\nu\chi + g(\sin\lambda)\nu\psi
\end{aligned}$$

where ν is defined in (6.3.2).

The wheel-road forces F_{xr} and F_{xf} are unknown. These magnitude are determined by the relation (6.3.3a), which assumes equilibrium. The rest of the parameters can be found using the geometry of the bike. Parameters are found for the modeled bike in [23] and are shown in table 6.8.1 in appendix 6.8.1.

6.4 Constraints and reduced set of equations

The mathematical model presented in section 6.3 only describes the motion regarding the origin of the reference coordinate system at p_r. The system actually has got nine variables, but only four presented in equation (6.3.3), since the motion of the front wheel contact point p_f is assumed to match the motion of the rear contact point p_f. This is actually not needed so the system can be completely described by the variables x_r x_f,y_r, y_f, θ_r, θ_f, χ_r, χ_f and the steering angle ψ. These variables for the front contact point will however be eliminated as constraints are introduced.

The approach for solving multibody dynamical systems is to derive the energy expressions for the n set of bodies and thereby obtaining the equations of motion. When the contributions for the bodies are derived the system is constrained to match the desired physical conditions. Constraints to a multibody system can be generalized to two types of constraints. An unconstrained multibody system has n degress of freedom characterized by the general coordinates $q_r(r = 1, 2, \cdots, n)$. The system can be physically restricted by positions or geometric constraints. These constraints are named *holonomic constraints*. They occur in the form:

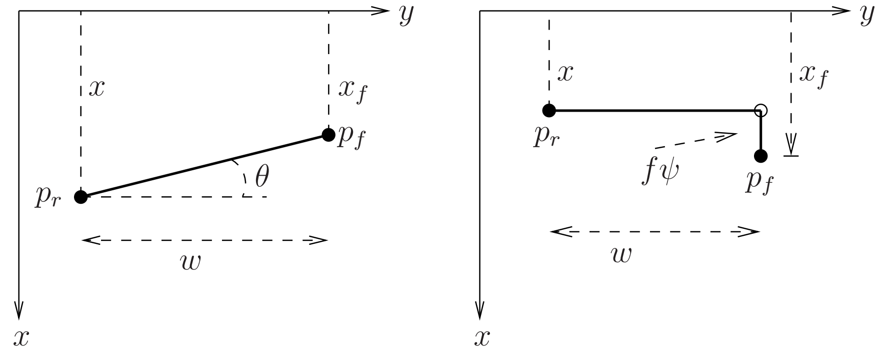
$$g_i(q_r, t) = 0 \qquad , \quad i = 1, 2, \cdots, m \quad r = 1, 2, \cdots, n \qquad (6.4.1)$$

where t is the time, m the number of constraints and n the total number of degrees of freedom. If a system is fully constrainted $m = n$ there are no degrees of freedom. The avaliable degrees of freedom can be written as $n - m$. An example of a holonomic system can be two bodies interacting through a frictionless joint.

Alternative the motion can be restricted physically by kinematic constraints. These constraints are called *non-holonomic constraints* and has the form:

$$g_i(q_r, \dot{q_r}, \ddot{q_r}, \cdots, t) = 0 \qquad , \quad i = 1, 2, \cdots, m \quad r = 1, 2, \cdots, n \tag{6.4.2}$$

The constraints in this paper is primary nonholonomic constraints. To analyse the constraints for the system we start out by examining the front wheel position with respect to the yawing motion. This can be seen in figure 6.2.



(a) Position of front wheel with respect to yaw.

(b) Position of front wheel with respect to the steering angle.

Figure 6.2: Postion of front wheel determined by yaw and steering.

Figure 6.2(a) shows the postion of the front wheel contact point p_f with no steering $\psi = 0$. Assuming small angles the relation between the rear position x and the front position x_f for the two contact points p_r and p_f respectively the relation can be written as (6.4.3).

$$\begin{aligned} x_f &= x - w \cdot \sin(\theta) \quad , \qquad \text{assuming small rotations} \qquad \sin(\theta) \simeq \theta \qquad \Rightarrow \\ x_f &\simeq x - w \cdot \theta \end{aligned} \tag{6.4.3}$$

The opposite situation is shown in figure 6.2(b). The position x_f with no yaw, $\theta = 0$, and an applied steering angle ψ can be written as:

$$x_f \simeq x + f \cdot \psi \tag{6.4.4}$$

Again small angle approximations has been made. Combining the two expressions for the front wheel position yields

$$x_f \simeq x - w \cdot \theta + f \cdot \psi \tag{6.4.5}$$

The yaw of the front wheel, θ_f combined with steering ψ is shown in figure 6.3.

Following [24] the expression for this case yields the relation in (6.4.6)

$$\theta_f \simeq \theta + \psi \cdot \cos(\lambda) \tag{6.4.6}$$

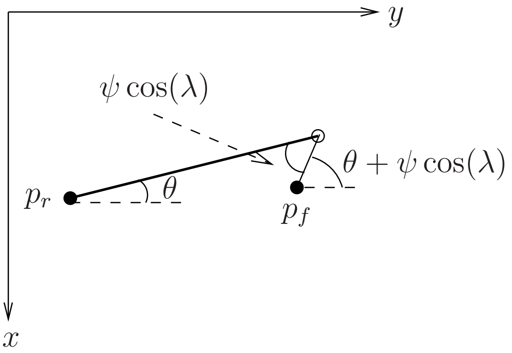


Figure 6.3: Combined yaw of front wheel and steering.

The coordinates for the front wheel can now be described by four relations:

$$x_f - x = -w\theta + f \cdot \psi \tag{6.4.7a}$$

$$y_f - y = w \tag{6.4.7b}$$

$$\theta_f - \theta = \psi \cos\lambda \tag{6.4.7c}$$

$$\chi_f - \chi = -\psi \sin\lambda \tag{6.4.7d}$$

The last relation (6.4.7d) is derived analogous to (6.4.7c), but is not needed for further analysis. The relation in (6.4.7b) assumes the wheels each rotate with the same velocity. The assumption is fairly reasonble since the wheels on a bicycle each nearly rotates at a constant rate. The cases where this assumption is invalid is however way beyond the scope of this model.

There is now two expressions for the position of the front wheel. The forward velocity for the front and rear wheel is shown in figure 6.4.

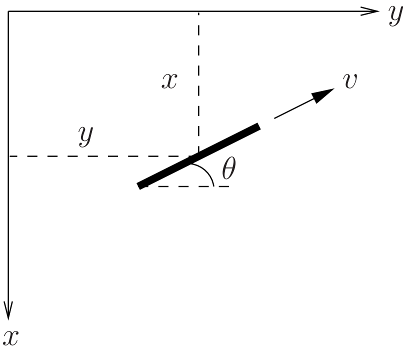


Figure 6.4: Velocity parameters.

The wheels are assumed to be thin knife-edge wheels with no slip in any direction. Again assuming small rotations the position with zero slip can be written as:

$$\begin{aligned} \frac{\Delta x}{\Delta y} &= -\tan(\theta) \qquad \text{, assuming small rotations} \quad \tan(\theta) \simeq \theta \Rightarrow \\ \dot{x} &\simeq -v \cdot \theta \end{aligned} \tag{6.4.8}$$

The increment in the y-direction Δy is taken as the forward speed v, and the corresponding increment in the x-direction is taken as the velocity of x. The increments can thereby be written as $\Delta y = v$ and $\Delta x = \dot{x}$. A similar approach can be used to determine the front wheel velocity. Analogous to (6.4.8) the velocity for the front wheel is written as:

$$\dot{x}_f = -v \cdot \theta_f \tag{6.4.9}$$

This analogy is true since the forward speed must be equal for both parts since w must be constant as seen in (6.4.7b). Differentiating (6.4.7a) yields:

$$\dot{x}_f - \dot{x} = -w\dot{\theta} + f\dot{\psi} \tag{6.4.10}$$

The two expressions for the wheel-road contact can be inserted into (6.4.10).

$$-v\theta - (-v\theta_f) = -w\dot{\theta} + f\dot{\psi} \tag{6.4.11}$$

The yaw θ for the front wheel is found in (6.4.7c). Substituting this into (6.4.11) gives:

$$\begin{aligned} v(-\theta + (\theta + \psi\cos\lambda) &= -w\dot{\theta} + f\dot{\psi} \qquad \Rightarrow \\ \dot{\theta} &= \tfrac{f}{w}\dot{\psi} + v\tfrac{\cos\lambda}{w}\psi \end{aligned} \tag{6.4.12}$$

Differentiating (6.4.12) once more

$$\ddot{\theta} = \tfrac{f}{w}\ddot{\psi} + V\tfrac{\cos\lambda}{w}\dot{\psi} \tag{6.4.13}$$

By using the relations in (6.4.7) the yaw θ is now completely determined by (6.4.7c), (6.4.12) and (6.4.13). This means the variables in the equations of motion now can be described by three variables. The variable x can easily be eliminated by differentation of rear constraint in (6.4.8):

$$\ddot{x} = -v\dot{\theta} = -v\tfrac{f}{w}\dot{\psi} - v^2\tfrac{\cos\lambda}{w}\psi \tag{6.4.14}$$

This reduces the variables to two. The last term in (6.4.14) is found by substituting θ with the relation in (6.4.12). The system can now be completely described by the steering angle ψ and the lean χ. The equations of motion are now:

$$M_{\chi\chi}\ddot{\chi} + K_{\chi\chi}\chi + M_{\chi\psi}\ddot{\psi} + C_{\chi\psi}\dot{\psi} + K_{\chi\psi}\psi = 0 \qquad \text{lean} \tag{6.4.15a}$$

$$M_{\psi\chi}\ddot{\chi} + C_{\psi\chi}\dot{\chi} + K_{\chi\psi}\chi + M_{\psi\psi}\ddot{\psi} + C_{\psi\psi}\dot{\psi} + K_{\psi\psi}\psi = M_{\psi} \qquad \text{steer} \tag{6.4.15b}$$

Since the expressions for the equations of motion are rather complicated the equations of motion are shown in the general form in (6.4.15). The terms can be found using the algorithmic approach in [23]. The algorithmic approach is written in appendix 6.8.2. The two equations of motion will be refered as the *lean* equation, (6.4.15a) and the *steer* equation, (6.4.15b). The applied moment M_ψ is the rider gained input to the system. This term is however ignored for the further analysis.

6.5 Numerical Solution

The reduced set of equations of motion in (6.4.15) are written as a matrix formulation.

$$\underbrace{\begin{bmatrix} M_{\chi\chi} & M_{\chi\psi} \\ M_{\psi\chi} & M_{\psi\psi} \end{bmatrix}}_{\mathbf{M}} \begin{bmatrix} \ddot{\chi} \\ \ddot{\psi} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & C_{\chi\psi} \\ C_{\psi\chi} & C_{\psi\psi} \end{bmatrix}}_{\mathbf{C}} \begin{bmatrix} \dot{\chi} \\ \dot{\psi} \end{bmatrix} + \underbrace{\begin{bmatrix} K_{\psi\psi} & K_{\chi\psi} \\ K_{\psi\chi} & K_{\psi\psi} \end{bmatrix}}_{\tilde{\mathbf{K}}} \begin{bmatrix} \chi \\ \psi \end{bmatrix} = \underbrace{\begin{bmatrix} 0 \\ M_\psi \end{bmatrix}}_{\tilde{\mathbf{F}}} \qquad (6.5.1)$$

The coefficients in (6.5.1) are determined by the algorithmic approach shown in appendix 6.8.2. Introducing the vector **q** as:

$$\mathbf{q} = \begin{bmatrix} \dot{\chi} \; \dot{\psi} \; \chi \; \psi \end{bmatrix}^T \qquad , \qquad \dot{\mathbf{q}} = \begin{bmatrix} \ddot{\chi} \; \ddot{\psi} \; \dot{\chi} \; \dot{\psi} \end{bmatrix}^T$$

The 'damping matrix' **C** is velocity dependent, see appendix 6.8.2, so **C** is multiplied with the forward velocity. The 'stiffness matrix' $\tilde{\mathbf{K}}$ consists of terms with no velocity and terms which are dependent with the squared forward speed. The $\tilde{\mathbf{K}}$ is split into two matrices according to [23]. The two terms are:

$$\begin{aligned} \tilde{\mathbf{C}} &= \mathbf{C} \cdot v \\ \tilde{\mathbf{K}} &= \mathbf{K}_1 + \mathbf{K}_2 \cdot v^2 \end{aligned}$$

Expressions for the entries in these two matrices are found as well in appendix 6.8.2. The **C** matrix is named the 'damping' matrix, since it takes the same form as the damping parameter in ordinary dynamical analysis. This matrix however does not dissipate energy, and the 'damping' naming convention is therefore not entirely correct. Instead of dissipating the energy is transfers the energy from one plane to a different plane in the system, but the transfer of energy displays the same phenomenae as, for instance viscous damping ($d\dot{q}$), with a exponential decay of amplitudes, but the energy remains in the system. Explanations about gyroscopic and inertia effects in this form is found in [27].

The coupling from second order for a set of first order differential equations are done using a state-space formulation. The state-space formulation can be seen in (6.5.3).

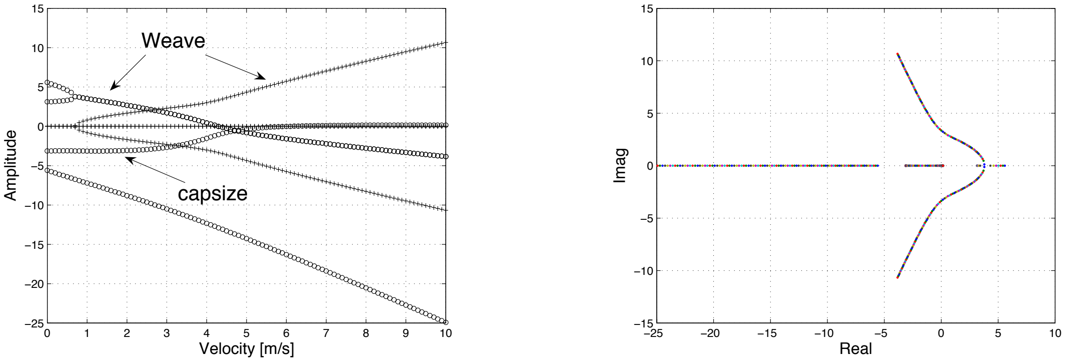
$$\begin{aligned} \begin{bmatrix} \mathbf{M} & \tilde{\mathbf{C}} \\ \mathbf{0} & \mathbf{M} \end{bmatrix} \dot{\mathbf{q}} + \begin{bmatrix} \mathbf{0} & \tilde{\mathbf{K}} \\ -\mathbf{M} & \mathbf{0} \end{bmatrix} \mathbf{q} &= \tilde{\mathbf{F}} \qquad (6.5.2) \\ \mathbf{A} \cdot \dot{\mathbf{q}} + \mathbf{B} \cdot \mathbf{q} &= \tilde{\mathbf{F}} \qquad (6.5.3) \end{aligned}$$

(6.5.3) can be solved by applying a suited ode solver to the problem. The problem stated in (6.5.3) does not contribute to any numerical problem regarding the numerical time integration. The following results are produced with MATLAB´s `ode45`.

6.5.1 Stability

To study the stability of the bicycle with no rider gained control the state space formulation is analysed regarding the systems eigenvalues for a range of forward velocities.

Following the terminology for bicycle/motorcycle stability there is two modes[1]. The first mode is named 'capsize'. This mode corresponds to the bicycle falls over due to lean. The capsize motion is however non-oscillatory. It just falls over. The second mode is named 'weave', which are oscillations about the headed direction. These two modes corresponds to the reduced set of variables derived in section 6.4, the lean χ and steering ψ. the eigenvalues for these modes are shown in figure 6.5(a).



(a) Real and complex eigenvalues for different forward velocities. The + marks the real eigenvalues and the o marks the imaginary eigenvalues.

(b) Real and complex eigenvalues shown in the complex plane.

Figure 6.5: Root Locus plot for the eigenvalues.

The weave motion is the most interesting in figure 6.5(a). For the inteval 0 to around 0.6 m/s the weave motion is highly unstable. The primary physical interpretation correponds

[1] there is actually three - the last mode 'wobble' is not present in this model.

to an inverted pendulum. Around $v = 0.6$ m/s the weave motion starts oscillating. The real eigenvalues becomes identical for the weave motion and forms a complex conjugated pair. The complex conjugated pairs corresponds to the oscillatory mode. Still one real eigenvalue remains positive so the mode is yet unstable.

At $v = 4.3$ m/s the real part of the weave motion becomes negative and remains negative until infinity. At this point the weave motion becomes stable. The capsize motion shown in figure 6.5(a) corresponds to the real eigenvalue, which starts out by being negative. Around $v = 6$ m/s this eigenvalue crosses the real axis and becomes mildly unstable. However the real part approaches 0 as velocity is increased. This means the bicycle is completely stable in the interval between these two velocities. Since the capsize motion is mildly unstable it is reasonable to assume in practice that the bicycle remains stable as soon as the weave motion becomes stable around $v = 4.3$ m/s. This corresponds to approx. 16 km/h.

6.5.2 Response with initial conditions

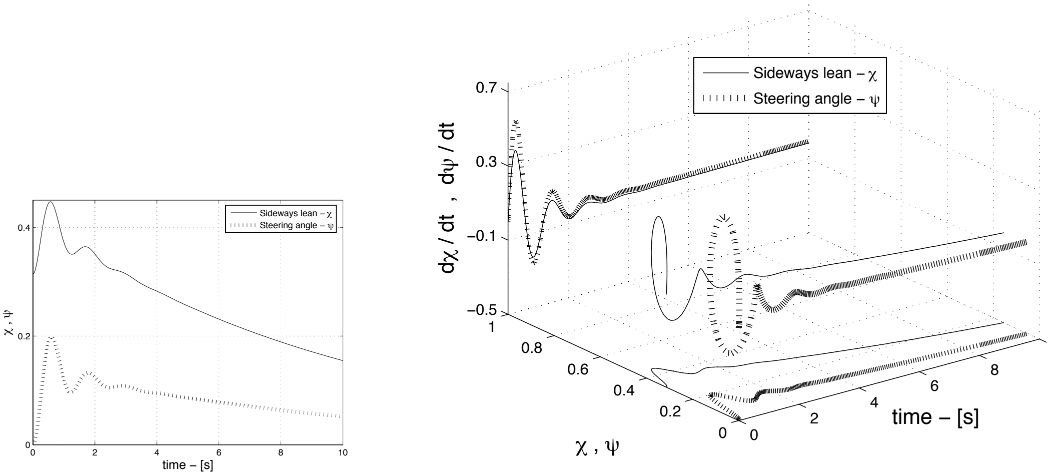
The response is limited to the range of $\chi \in [\ -\frac{\pi}{2}\ ,\ \frac{\pi}{2}\ ]$. This is not directly implemented in the numerical scheme, but the interpretation of the results are only valid in this region.

However the region can be extended. This corresponds to some extremely bold rider, who chooses to ride the bike with a reasonable high forward velocity, on a thin wire, with no hands, and furthermore choose to apply a sideways moment so the bike undergoes a $2 \cdot pi$ sideways spin around the wire. This kind of modelling is way beyond the scope of this paper. Anyway, if this modelling is used some serious questions regarding the linearization of the equations of motion should be adressed here.

The first simulation is made with an applied lean angle $\chi = \frac{\pi}{10}$ an an initial condition. The forward velocity is set to $v = 5.6\ m/s$. This velocity is in the stable range for both the capsize and weave motion. The figures in 6.6 shows the steering angle ψ and the lean χ as function of the forward speed.

The figures in 6.6 clearly displays the self-stabilizing of the bicycle. Minor oscillations about the equilibrium position occurs in the first two seconds. Afterwards is graduately decays towards equilibium. After around $t = 40\ s$ the system is back to the straight forward motion. Figure 6.6(b) shows the state-space response for the forward speed range. The dotted line corresponds to the steering angle ψ and the solid line is the lean angle χ. The oscillations are quite visible in the figure and it can be seen that the derivatives $\dot{\chi}$ and $\dot{\psi}$ show the same oscillation. However the displacement between derivatives is not as large as for the angles χ and ψ. The lean angle has larger amplitudes than the steering angle.

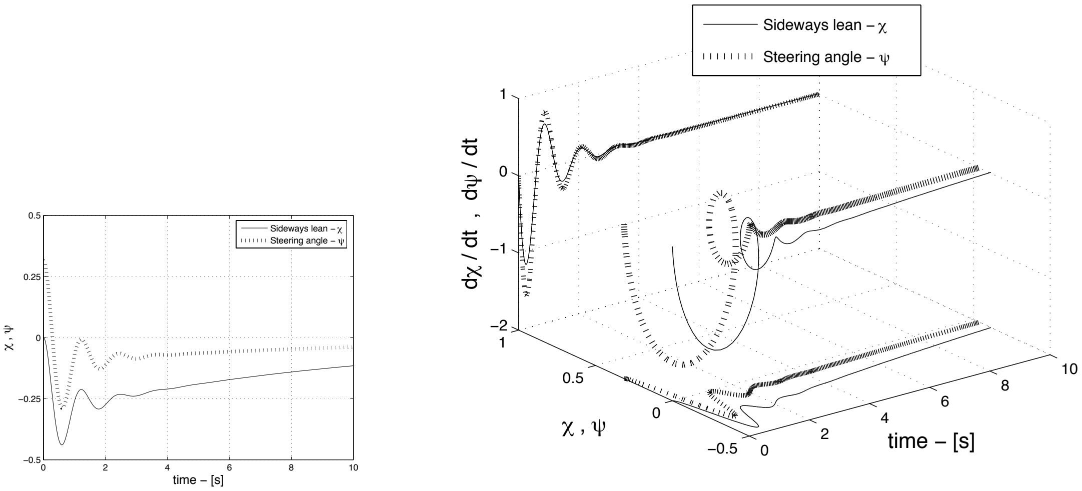
The next simulation is made with an applied steering angle $\psi = \frac{\pi}{10}$ as an initial condition. The results are shown in figure 6.7.



(a) Steering and lean as function of the forward speed.

(b) Angles and derivatives as function of the forward speed.

Figure 6.6: Results for the bicycle with an applied lean.



(a) Steering and lean as function of the forward speed.

(b) Angles and derivatives as function of the forward speed.

Figure 6.7: Results for the bicycle with an applied lean.

The self-stabilization is clearly seen in this case, and the bicycle returns faster to the original forward state with no lean than the simulation with an applied lean as an initial condition.

6.6 Extending the equations

Various different constraints has been applied to the model for the purpose of actually meet the subject of this course. Unfortunalely none of the experiments provided any reasonable results. The use of extra constraints makes the system overdeterminated, and in many cases gives singularities in the matrices in the state-space formulation. Advanced numerical time integrators has to be used, but the problem can be solved.

The primary reason for this is the reduction of the equations of motion represented in (6.4.15) in section 6.4. The reduced set of equations is derived using geometric relations to reduce the system variables to two instead of nine. This reduction is based on Hand work in [25]. The new set of equations enables the two unknown wheel-road forces to be eliminated.

The first test was to include the equation of motion in the x-direction, refered as equation (6.3.2) in section 6.3. Though it is not needed to describe the stability issues of the uncontrolled bike it could be used for a more advanced model of the wheel-road modelling. The goal was to enable some sort of possibility for a sideways slip to occur when the lean angle combined with the steering was a certain amount.

The first equation (6.3.2) can be used to find the unknown road-wheel force acting at the rear contact point p_r and a friction modelling using Coulomb friction was applied, but for all cases the results turned out false for the problem. In most cases the the bicycle reached a higher sideways speed than the forward speed - most likely due to pooly modelling.

A different constraint was implemented to restrict the sideways rotation, yaw, of the total mass (or the rider mass) to the corresponding inverted pendulum state. This constraint was implemented because the physics could be applied to the reduced set of equations, but yet again the constraint seemed meaningless.

Since the reduced set of equations only variables is lean χ and steering ψ the restriction of the movement rider as the inverted pendulum becomes obsolete simply because it was restricted to follow the path given by $\tilde{x}^2 + \tilde{y}^2 - h_t^2 = 0$, where $\tilde{x}$ and $\tilde{y}$ are introduced coordinate transforms from the lean χ polar representation into some arbitrary $(\tilde{x}, \tilde{y})$ plane. But since the transformation from the polar χ to the cartesian $(\tilde{x}, \tilde{y})$ can be written as $(\tilde{x}, \tilde{y}) = (\cos(\chi), \sin(\chi))$ and using the constrainst described above the problem is always true because of the identity $\cos(\chi)^2 + \sin(\chi)^2 = 1$. The is quite a meaningless constraint to introduce in the system.

The motion described by the yaw angle θ is also geometrical restricted. This follows the same restriction as the inverted pendulum $\tilde{x}^2 + \tilde{y}^2 - h_t^2 = 0$, but again the problem arises with the elimination of the sideways contact forces F_{xr} and F_{xf}.

The conclusion of the experiments with the extra constraints is the reduced set of equations limits the extension of the model. Advanced models for the wheel-road interaction is found in [26] but these models are described by many the degress of freedom. To actually extend the equations of motion presented in this paper requires a complete non-linear model and more degress of freedom. Alternatively a more simple model of the bicycle combined with a fully cartesian (x, y, z) space formulation of the motion could be extended to include algebraic constrains.

6.7 Conclusion

The model presented here is simple, relatively easy for a numerical implementation and it provides reasonable results regaring the stability analysis of an uncontrolled bicycle. The variables are reduced to two which means the model can be completely described by two coupled second order differential equations. The model represents the behaviours of a real-life bicycle well.

However the implementation of additional constraints has not been succesfull. The flipside for the simplicity of the model is the lack of possiblities for extensions. The point being is this particular model and stability analysis is relatively simple and easy to solve, but due to the restrictions provided by the analysis of the non-holonomic constrains additional (reasonable) constraints are not introduced easily. For a more differential algebraic approach a more simple geometry could be analysed combined with a cartesian representation. Furthermore the stability issues could be neglected and more emphasis put on the control of the bicycle, since the stability of the bicycle is closely related to the geometry and parameters regarding the bicycle model, which in this case limits the possibility of extending the equations of motion.

6.8 Appendix

6.8.1 Parameters for the numerical simulation

Parameter	Symbol	Value
Wheel base $[m]$	w	1.02
Trail $[m]$	t	0.08
Head Angle	α	arctan(3)
Gravity $[N/kg]$	g	9.81
Rear wheel	Symbol	Value
Radius $[m]$	R_{rw}	0.3
Mass $[kg]$	m_{rw}	2
Mass Moments of inertia - $[kgm^2]$	(A_{yy}, A_{xx}, A_{zz})	(0.12, 0.06, 0.06)
Rear frame	Symbol	Value
Postion centre of mass $[m]$	$x_{rf}, (y_{rf}, z_{rf})$	(0, 0.3, 0.9)
Mass $[kg]$	m_{rf}	85
Mass Moments of inertia $[kgm^2]$	$\begin{bmatrix} B_{xx} & 0 & 0 \\ & B_{yy} & B_{yz} \\ & & B_{xx} \end{bmatrix}$	$\begin{bmatrix} 11 & 0 & 0 \\ & 9.2 & 2.4 \\ & & 2.8 \end{bmatrix}$
Front frame	Symbol	Value
Postion centre of mass $[m]$	$x_{rf}, (y_{rf}, z_{rf})$	(0, 0.9, 0.7)
Mass $[kg]$	m_{rf}	4
Mass Moments of inertia $[kgm^2]$	$\begin{bmatrix} C_{xx} & 0 & 0 \\ & C_{yy} & C_{yz} \\ & & C_{xx} \end{bmatrix}$	$\begin{bmatrix} 0.06 & 0 & 0 \\ & 0.0546 & -0.0162 \\ & & 0.0114 \end{bmatrix}$
Front wheel	Symbol	Value
Radius $[m]$	R_{fw}	0.35
Mass $[kg]$	m_{fw}	3
Mass Moments of inertia - $[kgm^2]$	(D_{yy}, D_{xx}, D_{zz})	(0.28, 0.14, 0.14)

Table 6.1: Parameters for the numerical simulation.

6.8.2 Algorithm for deriving the equations of motion

Algorithmic approach to derive the equation og motion.

Calculate the total mass and the center of mass with respect to the origin O.

$$\begin{aligned}
m_t &= m_{rw} + m_{rf} + m_{fw} + m_{ff} \\
y_t &= \left(y_{rf} \cdot m_{rf} + y_{ff} \cdot m_{ff} + w \cdot m_{fw}\right)/m_t \\
z_t &= \left(-R_{rw} \cdot m_{rw} + z_{rf} \cdot m_{rf} + z_{ff} \cdot m_{ff} - R_{fw} \cdot m_{fw}\right)/m_t
\end{aligned}$$

Calculate the relevant mass moments and products of inertia at the origin O along the global axes:

$$\begin{aligned}
T_{yy} &= A_{yy} + B_{yy} + C_{yy} + D_{yy} + m_{rw}R_{rw}^2 + m_{rf}z_{rf}^2 + m_{ff}z_{ff}^2 + m_{fw}R_{fw}^2 \\
T_{yz} &= B_{yz} + C_{yz} - m_{rf}y_{rf}z_{rf} - m_{ff}y_{ff}z_{ff} + m_{fw}wR_{fw} \\
T_{zz} &= A_{zz} + B_{zz} + C_{zz} + D_{zz} + m_{rf}y_{rf}^2 + m_{ff}y_{ff}^2 + m_{fw}w^2
\end{aligned}$$

The same properties for the front part are calculated:

$$\begin{aligned}
m_f &= m_{ff} + m_{fw} \\
y_f &= \left(y_{ff}m_{ff} + wm_{fw}\right)/m_f \\
z_f &= \left(z_{ff}m_{ff} + R_{fw}m_{fw}\right)/m_f
\end{aligned}$$

Mass moments and products of inertia for the front part.

$$\begin{aligned}
F_{yy} &= C_{yy} + D_{yy} + m_{ff}(z_{ff} - z_f)^2 + m_{fw}(R_{fw} + z_f)^2 \\
F_{yz} &= C_{yz} - m_{ff}(y_{ff} - y_f) \cdot (z_{ff} - z_f) + m_{fw}(w - x_f) \cdot (R_{fw+z_f}) \\
F_{zz} &= C_{zz} + D_{zz} + m_{ff}(y_{ff} - y_f)^2 + m_{fw}(w - y_f)^2
\end{aligned}$$

The angle of the steering axis and the z-axis are defined as:

$$\begin{aligned}
\lambda &= \tfrac{\pi}{2} - \alpha \\
\boldsymbol{\lambda} &= [\sin(\lambda),\ 0,\ \cos(\lambda)]^T
\end{aligned}$$

Calculate the penpendicular distance d for the center of mass of the front assembly is ahead of the steering axis

$$d = (y_f - w - t)\cos(\lambda) - z_f \sin(\lambda)$$

The relevant mass moments and products of inertia along the stering axis are calculated as:

$$\begin{aligned}
F_{\lambda\lambda} &= m_f d^2 + F_{yy}\sin(\lambda)^2 + 2F_{yz}\sin(\lambda)\cos\lambda + F_{zz}\cos\lambda^2 \\
F_{\lambda y} &= -m_f d z_f + F_{yy}\sin(\lambda)F_{yz}\cos\lambda \\
F_{\lambda z} &= m_f d y_f + F_{yz}\sin(\lambda)F_{zz}\cos\lambda
\end{aligned}$$

The mechanical trail, which is the pependicular distance between the front wheel contact point and and the steering axis, are calculated as:

$$f = \frac{t \cdot \cos(\lambda)}{w}$$

The angular momentum along the x axis for the rear and front wheels are:

$$\begin{aligned} S_r &= \frac{A_{xx}}{R_{rw}} \\ S_f &= \frac{D_{xx}}{R_{fw}} \\ S_t &= S_f + S_r \end{aligned}$$

A static moment is defined as follows

$$S_d = m_f \cdot d + f \cdot m_t \cdot y_t$$

Using all the relations calculated the equations of motion can be found. The degrees of freedom are $\mathbf{q} = [\chi \ \psi]^T$.

$$\mathbf{M}\ddot{\mathbf{q}} + (\mathbf{C} \cdot v)\,\dot{\mathbf{q}} + \left(\mathbf{K}_1 + \mathbf{K}_2 \cdot v^2\right)\mathbf{q} = \mathbf{F} \tag{6.8.1}$$

The entries in the matrices in (6.8.1) are:

$$\begin{aligned} M(1,1) &= T_{yy} \\ M(1,2) &= M(2,1) \ = \ F_{\lambda y} + fT_{yz} \\ M(2,2) &= F_{\lambda\lambda} + 2fF_{\lambda z} + f^2 T_{zz} \end{aligned} \tag{6.8.2}$$

$$\begin{aligned} K_1(1,1) &= gm_t z_t \\ K_1(1,2) &= K_1(2,1) \ = \ -gS_d \\ K_1(2,2) &= -gS_d \sin(\lambda) \end{aligned} \tag{6.8.3}$$

$$\begin{aligned} K_2(1,1) &= K_2(2,1) \ = \ 0 \\ K_2(1,2) &= (S_t - m_t z_t)\cos(\lambda)/w \\ K_2(2,2) &= (S_d + S_f \sin(\lambda))\cos(\lambda)/w \end{aligned} \tag{6.8.4}$$

$$\begin{aligned} C(1,1) &= 0 \\ C(1,2) &= fS_t + S_f \cos(\lambda) T_{yz} \cos(\lambda)/w - fm_t z_t \\ C(2,1) &= -(fS_t + S_f \cos(\lambda)) \\ C(2,2) &= F_{\lambda z} \cos(\lambda)/w + f(S_u + T_{zz} \cos(\lambda)/w) \end{aligned} \tag{6.8.5}$$

Chapter 7

Parameter estimation in Differential Algebraic Equations

by : **Kristina Hoffmann**

7.1 Theory

This chapter is about estimating parameters in DAE-models, where there is some measured data $y(t)$. The DAE-model is described by

$$\frac{d}{dt}x(t) = f(x(t), u(t), \theta) \tag{7.1.1}$$

$$\hat{y}(t|\theta) = h(x(t), u(t), \theta) \ , \tag{7.1.2}$$

where θ is the unknown parameters, to be estimated and $\hat{y}(t|\theta)$ is predicted values of $y(t)$ from the model.

The θ's is found by minimizing the difference between the measured data $y(t)$ and the predicted values. The difference is

$$\varepsilon(t, \theta) = y(t) - \hat{y}(t|\theta) \ , \tag{7.1.3}$$

and the function to minimize is

$$V_N(\theta) = \frac{1}{N}\sum_{t=1}^{N} \varepsilon^2(t,\theta) \tag{7.1.4}$$

$$= \frac{1}{N}\sum_{k=1}^{N} \|y(t_k) - \hat{y}(t_k|\theta)\|^2 \tag{7.1.5}$$

$$= \frac{1}{N}\sum_{k=1}^{N} \frac{1}{2}\left(y(t_k) - \hat{y}(t_k|\theta)\right)^2 \quad . \tag{7.1.6}$$

Here the 2-norm is used, but other norms can also be used. Then the θ's which minimizes $V_N(\theta)$ has to be found

$$\hat{\theta}_N = \arg(\min(V_N(\theta))) \quad . \tag{7.1.7}$$

Newton-Raphsons method is used to find the minimum

$$\frac{d}{d\theta}V_N(\theta) = 0 \Rightarrow \tag{7.1.8}$$

$$\hat{\theta}^{(i+1)} = \hat{\theta}^{(i)} - \mu^{(i)}\left[V_N''(\hat{\theta}^{(i)})\right]^{-1} V_N'(\hat{\theta}^{(i)}) \quad , \tag{7.1.9}$$

where $V_N'(\theta)$ is the gradient of $V_N(\theta)$ and $V_N''(\theta)$ is the hessian of $V_N(\theta)$. The step-length $\mu^{(i)}$ is found by making sure that

$$V_N(\hat{\theta}^{(i+1)}) < V_N(\hat{\theta}^{(i)}) \quad . \tag{7.1.10}$$

This is done by assuming that $V_N(\theta)$ is a quadratic polynomial

$$V_N(\theta) = \alpha_0 + \alpha_1\theta + \alpha_2\theta^2 \quad , \tag{7.1.11}$$

and then inserting 3 values for θ in $V_N(\theta)$ to find α_0, α_1 and α_2. When this is done the θ-value which minimizes $V_N(\theta)$ can be found from

$$\theta_{min} = \frac{-\alpha_1}{2\alpha_2} \quad , \tag{7.1.12}$$

and then $\mu^{(i)}$ is found as

$$\mu^{(i)} = \left(\theta^{(i)} - \theta_{min}\right) \Delta\theta \quad . \tag{7.1.13}$$

When the norm in equation (7.1.6) is used, it is found that the derivatives of $V_N(\theta)$ is

$$V_N'(\theta) = \frac{1}{N} \sum_{k=1}^{N} - \left(\frac{d}{d\theta} \hat{y}(t_k|\theta) \right) (y(t_k) - \hat{y}(t_k|\theta)) \tag{7.1.14}$$

and

$$V_N''(\theta) = \frac{1}{N} \sum_{k=1}^{N} \left(\frac{d}{d\theta} \hat{y}(t_k|\theta) \right) \left(\frac{d}{d\theta} \hat{y}(t_k|\theta) \right)^T \tag{7.1.15}$$

$$+ \frac{1}{N} \sum_{k=1}^{N} \left(\frac{d^2}{d\theta^2} \hat{y}(t_k|\theta) \right) (y(t_k) - \hat{y}(t_k|\theta)) \quad . \tag{7.1.16}$$

The second sum in the second derivative is often eliminated, because when you are close to the solution, $\frac{d^2}{d\theta^2}\hat{y}(t|\theta)$ is close to zero and then it is not necessary to find $\frac{d^2}{d\theta^2}\hat{y}(t|\theta)$, which can be very difficult. Then the only thing missing is to find $\frac{d}{d\theta}\hat{y}(t|\theta)$ from equation (7.1.2).

If it is not possible to find $\frac{d}{d\theta}\hat{y}(t|\theta)$ analytically, the following steps can be used to find some coupled differential equations which gives $\frac{d}{d\theta}\hat{y}(t|\theta)$ as a solution. From equation (7.1.2) it is found that

$$\frac{d}{d\theta}\hat{y}(t|\theta) = \frac{d}{d\theta} \left(h(x(t,\theta), u(t), \theta \right) \tag{7.1.17}$$

$$= H_1 \left(x(t,\theta), u(t), \theta \right) z(t,\theta) + H_2 \left(x(t,\theta), u(t), \theta \right) \quad , \tag{7.1.18}$$

where

$$\begin{aligned}
H_1(x,u,\theta) &= \frac{d}{dx}h(x,u,\theta) && (7.1.19)\\
z(t,\theta) &= \frac{d}{d\theta}x(t,\theta) && (7.1.20)\\
H_2(x,u,\theta) &= \frac{d}{d\theta}h(x,u,\theta) \ . && (7.1.21)
\end{aligned}$$

From equation (7.1.1) and equation (7.1.20) it is found that

$$\begin{aligned}
\frac{d}{dt}z(t,\theta) &= \frac{d}{dt}\frac{d}{d\theta}x(t,\theta) && (7.1.22)\\
&= \frac{d}{d\theta}\frac{d}{dt}x(t,\theta) \qquad \text{If } x(t,\theta) \text{ is two times continous differentiable.} && (7.1.23)\\
&= F_1\left(x(t,\theta),u(t),\theta\right)z(t,\theta) + F_2\left(x(t,\theta),u(t),\theta\right) \ , && (7.1.24)
\end{aligned}$$

where

$$\begin{aligned}
F_1(x,u,\theta) &= \frac{d}{dx}f(x,u,\theta) && (7.1.25)\\
F_2(x,u,\theta) &= \frac{d}{d\theta}f(x,u,\theta) \ . && (7.1.26)
\end{aligned}$$

7.2 Example

The use of this theory is shown on the following example, a water tank with free outlet. The cross section area of the tank is A measured in (m^2) and the area of the outlet hole is a also in (m^2). The water level in the tank is h measured in (m). The flow into the tank is u measured in $(\frac{m^3}{s})$ and the flow out of the tank is q also in $(\frac{m^3}{s})$.

The law of Bernoulli describe the outlet velocity v measured in $(\frac{m}{s})$ as a function of the water level h in the tank

$$v(t) = \sqrt{2gh(t)} \ , \qquad (7.2.1)$$

where g is the gravitational acceleration. The connection between the flow out of the tank q and the outlet velocity v is per definition

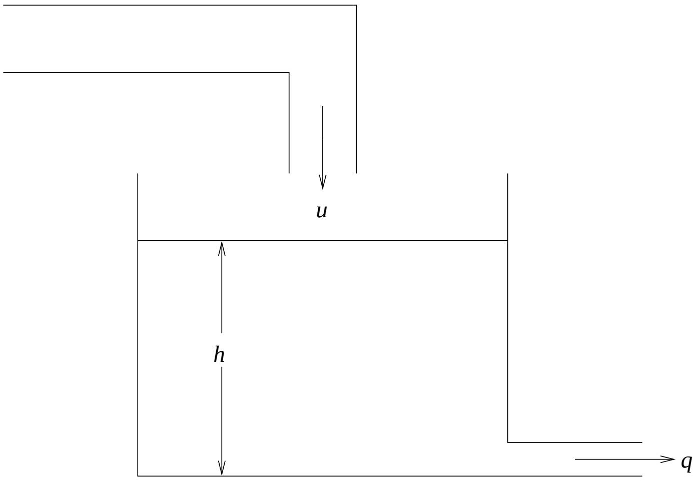


Figure 7.1: Model of the tank used in this example.

$$q(t) = a \cdot v(t) \ \ . \tag{7.2.2}$$

The volume of the water in the tank by the time t is

$$V(t) = A \cdot h(t) \ \ , \tag{7.2.3}$$

measured in (m^3). The volume changes in connection with the difference between the flow into and out of the tank

$$\frac{d}{dt}Ah(t) = u(t) - q(t) \ \ . \tag{7.2.4}$$

The three equations (7.2.1), (7.2.2) and (7.2.4) describe a model for the tank on figure 7.1. Equation (7.2.1) and equation (7.2.2) is inserted in equation (7.2.4) to get an explicit equation for the water level in the tank.

$$\frac{d}{dt}h(t) = -\frac{a\sqrt{2g}}{A}\sqrt{h(t)} + \frac{1}{A}u(t) \ \ . \tag{7.2.5}$$

From this equation it is possible to determine $h(t)$ because the flow into the tank $u(t)$ is known. Now the flow out of the tank $q(t)$ can be determined from equation (7.2.2), equation (7.2.1) and the solution to equation (7.2.5) as

$$q(t) = a\sqrt{2g}\sqrt{h(t)} \ . \tag{7.2.6}$$

To make the model a little simpler the parameter A is chosen to 1 and $a = \theta$ is used as the unknown parameter. This gives the model

$$\begin{aligned} \dot{h(t)} &= -\theta\sqrt{2g}\sqrt{h(t)} + u(t) & (7.2.7)\\ \hat{q}(t|\theta) &= \theta\sqrt{2g}\sqrt{h(t)} \ . & (7.2.8) \end{aligned}$$

When the model from equation (7.2.7) and (7.2.8) is inserted in equation (7.1.6) $V_N(\theta)$ becomes

$$V_N(\theta) = \frac{1}{N}\sum_{k=1}^{N}\frac{1}{2}\left(q(t_k) - \hat{q}(t_k|\theta)\right)^2 \ , \tag{7.2.9}$$

where $\hat{q}(t_k|\theta)$ is the predicted values of the flow out of the tank from the model and $q(t_k)$ is the measured values of the flow out of the tank for the experiment, se figure 7.2.

Because this model is simple the derivative can be found analytically from equation (7.1.14) and equation (7.1.16) as

$$\frac{d}{d\theta}\hat{q}(t|\theta) = \sqrt{2g}\sqrt{h(t)} \ , \tag{7.2.10}$$

and

$$\frac{d^2}{d\theta^2}\hat{q}(t|\theta) = 0 \ . \tag{7.2.11}$$

So here it is not only an assumption that the second sum in the second derivative is close to zero.

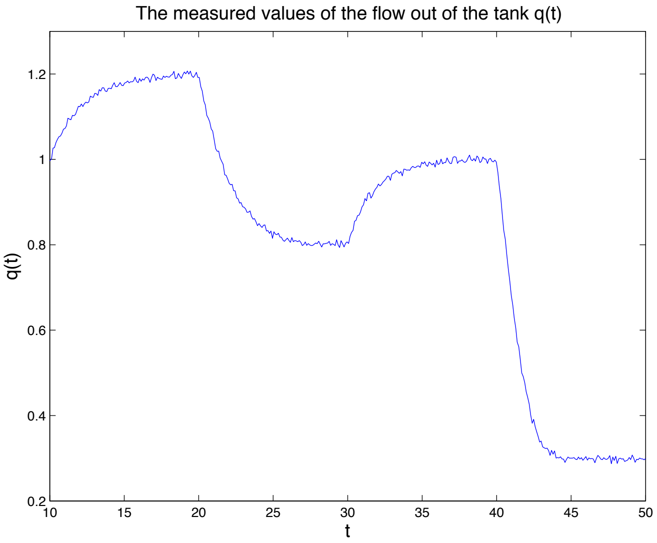


Figure 7.2: The measured values of the flow out of the tank $q(t_k)$ for the experiment.

Because it is so easy to find the derivative analytically it is not necessary to use the equations (7.1.17)-(7.1.26) to find $\frac{d}{d\theta}\hat{q}(t|\theta)$.

In this experiment the area of the outlet hole is $a = \frac{1}{\sqrt{2g}} \approx 0.226$. The flow into the tank $u(t)$ for the experiment can be seen on figure 7.3.

$V_N(\theta)$ is plotted as a function of θ and is shown on figure 7.4.

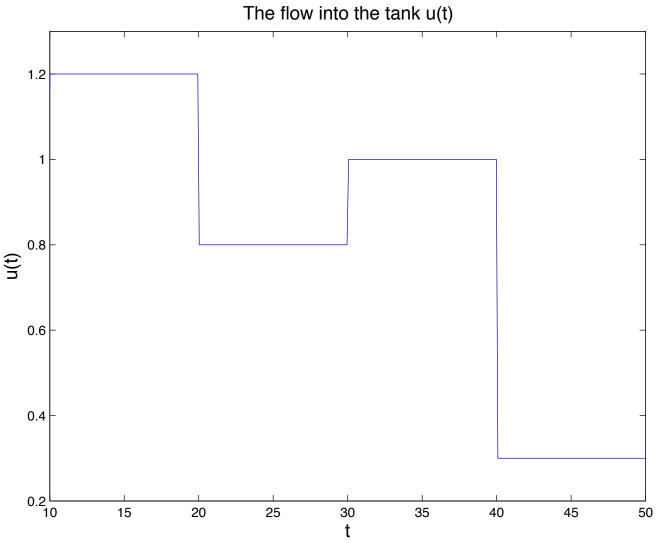


Figure 7.3: The flow into the tank $u(t)$ for the experiment.

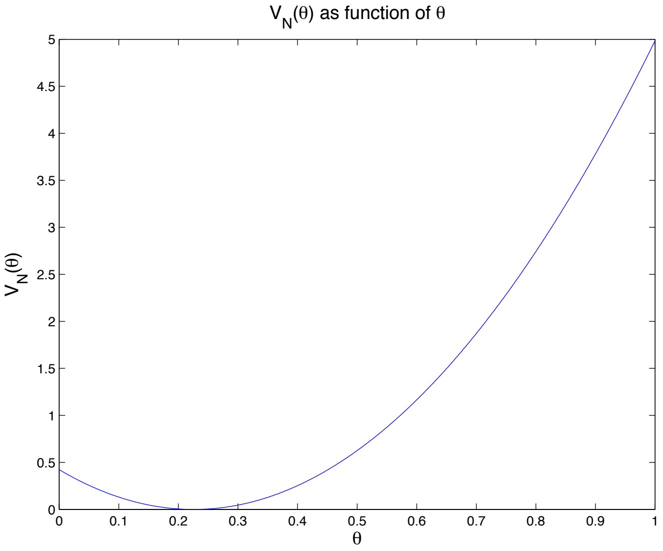


Figure 7.4: $V_N(\theta)$ from equation (7.2.9) as a function of θ.

As it can be seen $V_N(\theta)$ has the minimum value for $a \approx 0.225$. This is very close to the actual value which is $a \approx 0.226$.

To use the theory for finding the parameters in the experiment the model as described above is used, but instead of using the measured values of the flow out of the tank, $q(t)$ is made by using the solution from solving $\dot{h}(t)$ to find $\hat{q}(t|\theta)$ with the right θ-value in the model and then add a random error. The predefined function in Matlab called `randn.m` is used, this function find a random number chosen from a normal distribution with mean zero, variance one and standard deviation one. In this way it is possible to control the size of the error and thereby examine the implementation of the theory.

At first the implementation is run without any error. In this way it is possible to se in which level of tolerance of $V_N(\theta)$ the parameter can be found in and how many iterations it takes.

If the tolerance is set to 10^{-4}, the implementation finds that $\theta = 0.2401$ in 40 iterations. If the tolerance is set to 10^{-5}, θ is found to 0.2303 in 51 iterations. When the tolerance is set to 10^{-6} the implementation can not converge so there is no solution. This means that the implementation is expected to converge for a tolerance around $10^{-4} - 10^{-5}$. The error is squared in $V_N(\theta)$ in equation (7.2.9) so for $tol = 10^{-4}$ the error can be around 0.01 and for $tol = 10^{-5}$ the error can be around 0.003.

Then the error is set to be 0.0001 times the random numbers. The absolute maximum of these errors is then around 0.0003, so this should not give any problems. The results is shown in table 7.1.

$error = 0.0001 * randn$

$tol = 10^{-4}$		$tol = 10^{-5}$	
θ	$iterations$	θ	$iterations$
0.2406	40	0.2303	54
0.2409	41	0.2308	49
0.2407	41	0.2301	43
0.2406	40	0.2301	48
0.2401	42	0.2303	44

Table 7.1: Results for $error = 0.0001 * randn$.

As it can be seen the results is approximately the same as when there is no error. Then the error is made 5 times larger, $error = 0.0005 * randn$. The absolute maximum of these errors is then around 0.001. This gives the results in table 7.2.

With this error the results is also approximately the same as when there is no error. The error is now set to $error = 0.001 * randn$ which gives the absolute maximum of these errors around 0.003 and this gives the results in table 7.3.

Again it can be seen that the results is approximately the same as when there is no error. The error is now at the limit for what the implementation is expected to handle. If the error is set to $error = 0.01 * randn$, which gives a absolute maximum error around 0.03,

$error = 0.0005 * randn$

$tol = 10^{-4}$		$tol = 10^{-5}$	
θ	$iterations$	θ	$iterations$
0.2406	39	0.2302	51
0.2409	42	0.2297	56
0.2407	42	0.2305	48
0.2407	42	0.2302	58
0.2399	46	0.2302	55

Table 7.2: Results for $error = 0.0005 * randn$.

$error = 0.001 * randn$

$tol = 10^{-4}$		$tol = 10^{-5}$	
θ	$iterations$	θ	$iterations$
0.2401	40	0.2303	51
0.2410	39	0.2298	50
0.2337	41	0.2303	55
0.2407	40	0.2301	52
0.2407	41	0.2302	55

Table 7.3: Results for $error = 0.001 * randn$.

this is above the error-limit for $tol = 10^{-5}$ and on the limit for $tol = 10^{-4}$. This is also what the implementation gives as results. For $tol = 10^{-5}$ there is no convergence and for $tol = 10^{-4}$ the result is almost the same as if there was no error.

7.3 Conclusion

The implementation of the theory for finding parameters in Differential Algebraic Equation models is very good. If the tolerance, which gives convergence for the model without any error, is used and the error is smaller than what this tolerance can handle as error, the implementation finds the same parameters for the model with error as if there was no error. So as long as the tolerance and the error is adjusted to the model without error, the implementation finds the correct parameters.

Chapter 8

Trajectory Prescribed Path Control - TPPC

by : **Carsten Vølcker and Peter Larsen**

8.1 Introduction

In this report we consider a *Trajectory Prescribed Path Control - TPPC* problem. Imagine a space shuttle that returning from its mission has to reenter the atmosphere. This is a very critical point of any mission in space, and there are many parameters to be taken into account. When reaching the atmosphere the vehicle has high velocity, since the vacuum of outer space does not allow it to decellerate. We will focus our analysis on a specific model taken from *Brenan et al.* [20] which describes a trajectory to be followed by the vehicle during reentry. Two control parameters *roll* and *pitch* can be modified to stabilize the vehicle and keep it on the prescribed path. We do not consider heating of the shuttle nor the specific control system and devices.

First of all we present the theoretical model and explain some of the aerodynamic parameters involved. We then move on to the implementation in Matlab, which is followed by the main results of the simulation. The last two sections are discussion and conclusion.

8.2 Theoretical model

The point of departure for the theoretical model is quite basic - it is Newton's second law of motion. The following three forces affect the propulsion free vehicle:

- Gravitational pull mg
- Aerodynamic lift force L
- Aerodynamic drag force D

Let $\overline{r}$ denote the position vector and m the mass, then it adds up to

$$m\overline{r}'' = m\overline{g} + \overline{L} + \overline{D}, \tag{8.2.1}$$

or equivalently by introducing the velocity vector $\overline{v}$

$$\begin{aligned} \overline{r}' &= \overline{v} \\ \overline{v}' &= \overline{g} + \frac{1}{m}(\overline{L} + \overline{D}), \end{aligned} \tag{8.2.2}$$

where the prime denotes the derivative with respect to time. The magnitude of the drag force is given by

$$D = \frac{1}{2}\rho V_R^2 S C_D, \tag{8.2.3}$$

and its direction is opposite the velocity of the vehicle. Here ρ is the density of air (which is a function of altitude), S is the cross sectional reference area of the vehicle, V_R is the magnitude of the velocity relative to the rotation of the earth. The lift force is similarly given by

$$L = \frac{1}{2}\rho V_R^2 S C_L, \tag{8.2.4}$$

but with *lift* coefficient C_L. It is perpendicular to $\overline{D}$. The coefficients C_D and C_L will be discussed in section 8.2.2.

The dynamical model will be described in three different coordinate systems. The first one is spherical with origin at the center of the earth, and it rotates at the angular velocity of the earth Ω. In this coordinate system the variables that represent the position of the center of mass of the vehicle are measured; the altitude above earth surface H, the azimuth angle ε, and the zenith angle λ. This is seen in figure 8.1.

The second system is for measuring the velocity variables. It is measured in a local spherical system with origin at the position of the lift center of the vehicle. This system is also shown in figure 8.1. The x-axis is always towards the north, the y-axis towards the east,

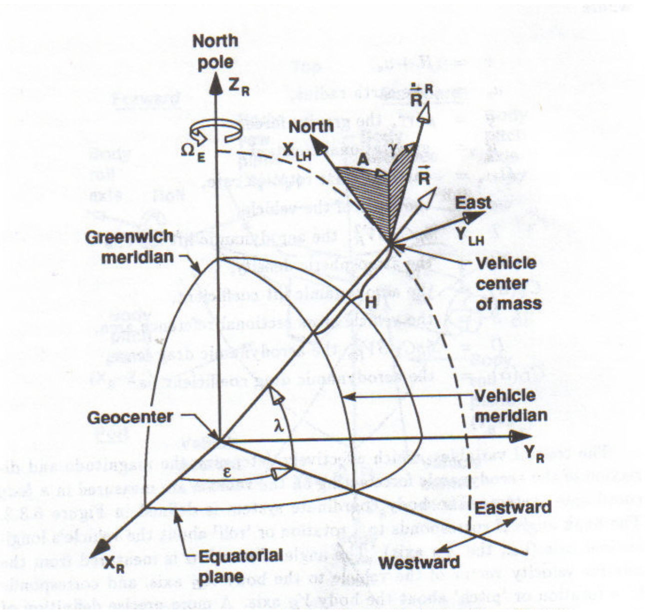


Figure 8.1: Figure 6.3.1 from Brenan et al. [20].

and the z-axis points towards the center of the earth. This system is locally horizontal with respect to gravity, since the z-axis points to the center of the earth. The first system is relative to the rotating earth and hence the velocity is referred to as the relative velocity. The magnitude of the velocity vector in this system is V_R, the azimuth angle is A, and the zenith angle is denoted γ.

The third and last system is for describing the orientation of the vehicle relative to the velocity vector from the lift center of the vehicle - see figure 8.2. Let's place the vehicle with the nose and tail along the x-axis, the y-axis is from side to side, and the z-axis from top to bottom. The three angles of rotation are called pitch or α about the y-axis, roll or β about the x-axis, and yaw about the z-axis. In this context yaw is not considered. The variables in this last system are called the control parameters (CP), since they are the ones that enable us to steer the vehicle.

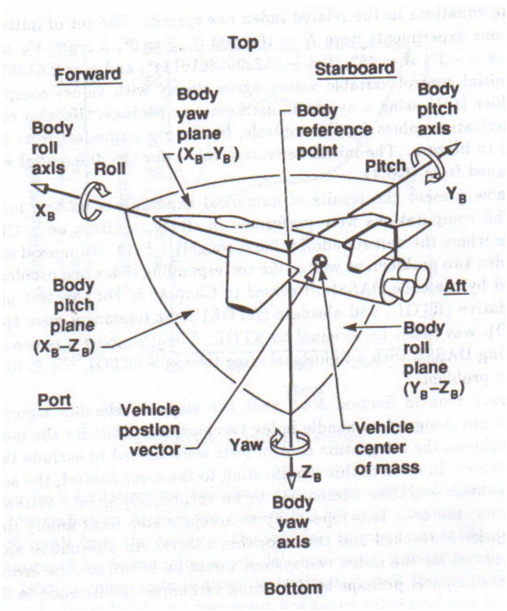


Figure 8.2: Figure 6.3.2 from Brenan et al. [20].

With the three coordinate systems we can describe the dynamics of the vehicle with a 6 dimensional system of first order differential equations:

$$
\begin{aligned}
H' &= V_R \sin(\gamma) && (8.2.5)\\
\varepsilon' &= \frac{V_R \cos(\gamma)\sin(A)}{r\cos(\lambda)} && (8.2.6)\\
\lambda' &= \frac{V_R}{r}\cos(\gamma)\cos(A) && (8.2.7)\\
V_R' &= -\frac{D}{m} - g\sin(\gamma) \\
&\quad - \Omega^2 r\cos(\lambda)\Big(\sin(\lambda)\cos(A)\cos(\gamma) - \cos(\lambda)\sin(\gamma)\Big) && (8.2.8)\\
\gamma' &= \frac{L\cos(\beta)}{mV_R} + \frac{\cos(\gamma)}{V_R}\left(\frac{V_R^2}{r} - g\right) + 2\Omega\cos(\lambda)\sin(A) \\
&\quad + \frac{\Omega^2 r\cos(\lambda)}{V_R}\Big(\sin(\lambda)\cos(A)\sin(\gamma) - \cos(\lambda)\cos(\gamma)\Big) && (8.2.9)\\
A' &= \frac{L\sin(\beta)}{mV_R\cos(\gamma)} + \frac{V_R}{r}\cos(\gamma)\sin(A)\tan(\lambda) \\
&\quad - 2\Omega\Big(\cos(\lambda)\cos(A)\tan(\gamma) - \sin(\lambda)\Big) \\
&\quad + \frac{\Omega^2 r\cos(\lambda)\sin(\lambda)\sin(A)}{V_R\cos(\gamma)}\,. && (8.2.10)
\end{aligned}
$$

Equations (8.2.5) through (8.2.10) contain a lot of trigonometric functions which are due to projections in the two interlinked spherical coordinate systems. All terms proportional to g express the gravitational pull. Whenever the factor Ω^2 is present it is a centrifugal force term, and similarly for the factor Ω it is a coriolis force term. The L terms in (8.2.9) and (8.2.10) have a trigonometric dependence on β. If β is zero we have maximum lift and maximum influence on the zenith angle γ and zero influence on the azimuth angle A in the local coordinate system. The azimuth angle A has to be symmetrical about $\beta = 0$ which explains the above choice of trigonometric dependencies - $\cos(\beta)$ in (8.2.9) and $\sin(\beta)$ in (8.2.10). In (8.2.8) $\overline{D}$ always has the opposite direction of the velocity, hence the minus sign. The leftover force terms are due to coordinate transformations.

As mentioned earlier m is the mass of the vehicle. The angular velocity of the earth is Ω. H is the altitude above the surface of the earth, and r is the distance from the center of the earth, so they are related by $r = H + a_e$ where a_e is the earth radius. The gravitational acceleration g is given by $g = \mu/r^2$, where μ is the gravitational constant. The density of atmospheric air ρ which has an influence on lift and drag depends on H. Lift and drag coefficients C_L and C_D depend on the pitch angle α and also on the velocity V_R and the altitude H.

8.2.1 Constraints

In *Brenan et al.* [20] the constraints for the trajectory to be followed by the vehicle are given only in terms of γ and A

$$\gamma + 1 + 9\left(\frac{t}{300}\right)^2, \tag{8.2.11}$$

$$A - 45 - 90\left(\frac{t}{300}\right)^2. \tag{8.2.12}$$

Put into words, γ varies from -1 to -10 degrees in a parabolic fashion from 0 to 300 seconds. A varies from 45 to 135 degrees in the same way. Where equations (8.2.5) through (8.2.10) make up the differential part of the DAE problem, these constraints are the algebraic part. With (8.2.11) and (8.2.12) the whole problem has differential index two. This can be argued in the following way. If you differentiate once with respect to time you end with γ' and A' which are given by equations (8.2.9) and (8.2.10). These two equations have trigonometric terms in α and β. This means that differentiating a second time will yield time derivatives of α and β which corresponds to an 8 dimensional ordinary differential equations system or an order zero DAE-system.

The constraints might seem rather strange, since they include neither altitude nor velocity. Equation (8.2.11) describes the angular difference from a locally horizontal plane. The

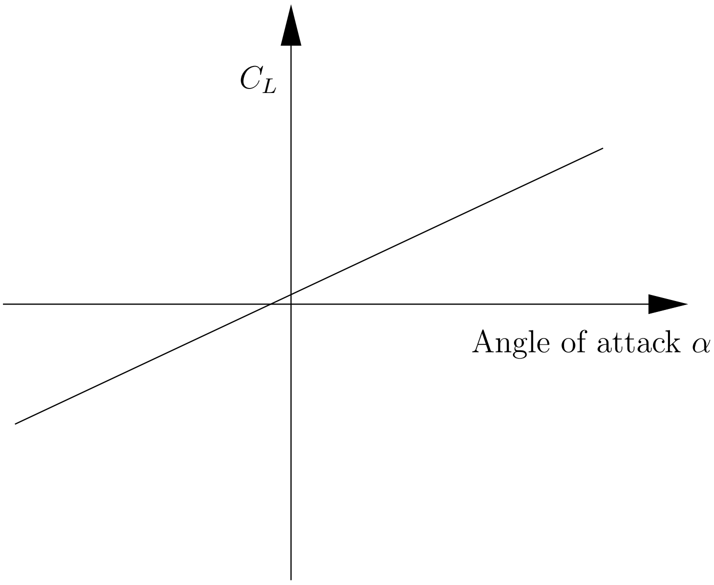


Figure 8.3: Liftcoefficient C_L as a function of the angle of attack α.

increasing negative angle means that the steepness of the descent will increase. Equation (8.2.12) describes the orientation on the compass with 0 degrees being north. So starting from north-east at 45 degrees, the vehicle ends up flying south-east at 135 degrees.

8.2.2 Lift and drag coefficients

The two aerodynamic forces *lift* and *drag* are given by equations (8.2.3) and (8.2.4). In each equation either the lift coefficient or the drag coefficient intervenes along with some other parameters. As mentioned earlier C_D and C_L are functions of angle of attack α and the Mach number. The Mach number is the ratio of V_R and the speed of sound. If we consider the atmosphere to be a ideal gas, the speed of sound will only depend on temperature. For a specific aircraft C_L and C_D should be determined experimentally. However a the dependance on α and M always has the same qualitative behavior.

For constant M C_L is linear in α with a small constant term which is positive for most M. This is seen in figure 8.3. For the larger α the linear behavior stops and we get to what is called stall, where a further increase will create no more lift. For larger M the slope has a tendency to become smaller.

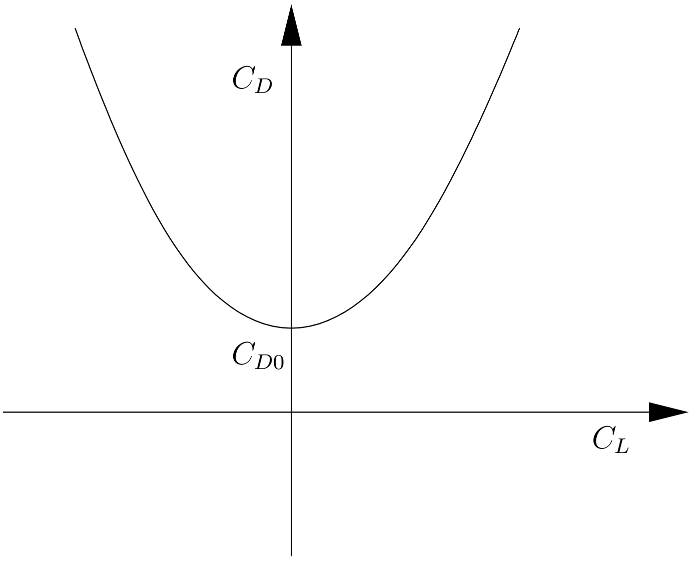


Figure 8.4: A socalled polar plot of the dragcoefficient C_D as a function of the liftcoefficient C_L.

The standard way of plotting the dragcoefficient is in a so called polar plot as a function of C_L. In general this yields a parabolic shape with a strictly positive minimum as seen in figure 8.4. The dragcoefficient can never become negative since this would the drag force would accelerate the aircraft.

The dependance on M is not as "nice" as on α. There a different regimes that have distinctively different behaviors. These are the subsonic ($M < 1$), supersonic ($M > 1$), and hypersonic ($M \gg 1$) regimes. The hypersonic regime is not as clearly defined as sub- and supersonic, but it must be considered when treating reentry problems, since the velocity of the reentering shuttle is of the order ten times the speed of sound.

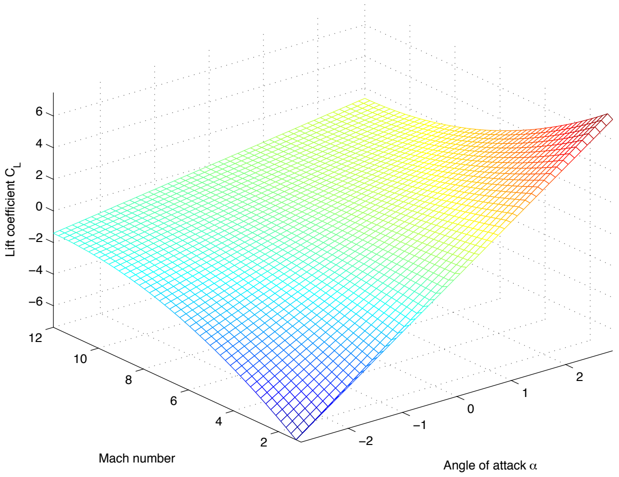


Figure 8.5: Lift coefficient C_L from *Scott and Olds* [21] as a function of the angle of attack α and the Mach number.

8.3 Implementation

8.3.1 Parameters

The temperature and the density of air are both taken from tables in *Databog i fysik & kemi* [19]. Polynomial fits have been used to implement them as functions in our program.

When it comes to lift and drag coefficients things are a bit more complicated, since relevant data are not readily available for specific shuttle types. We have based our coefficients on a theoretically calculated example taken from *Scott and Olds* [21]. The lift coefficient is a linear equation of type

$$C_L = C_{L0} + s\alpha,$$

where C_{L0} and s are parametrized by second order polynomials in the Mach number M and the aspect ratio AR of the shuttle. There are two different parametrizations for the subsonic and super/hypersonic regimes. The lift coefficient is shown in figure 8.5 as a function of α and M for $AR = 1.86$. One clearly sees for constant M we have a straight line, and the slope decreases with increasing M. There is also a discontinuity at $M = 1$ because of the change from sub- to supersonic.

In the same way the drag coefficient which approximately has a parabolic shape is given

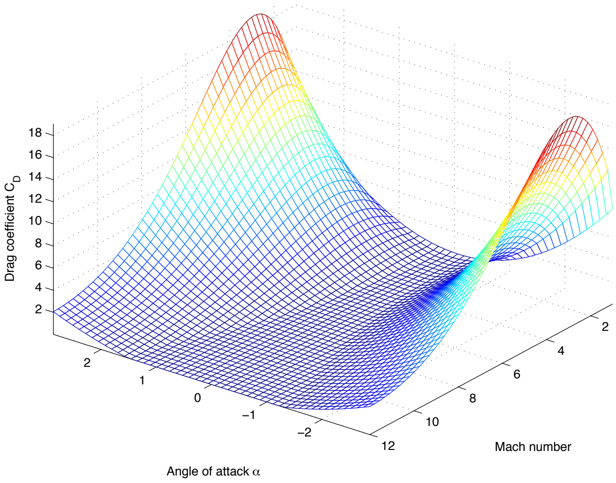


Figure 8.6: Drag coefficient C_D from *Scott and Olds* [21] as a function of the angle of attack α and the Mach number.

by a the equation

$$C_D = C_{D0} + K_1 C_L + K_2 C_L^2,$$

where this time C_{D0}, K_1, and K_2 are parametrized to second order by M and AR. In figure 8.6 this is presented as function of α and M for $AR = 1.86$. Again there are different parametrizations for sub- and supersonic. One notices the "flat bottom" down the middle of the graph. This is due to the fact that the coefficient actually becomes negative for small α, and this of course we cannot have. We have fixed this problem by setting all values of C_D below 0.04 to 0.04. This of course is very far from an optimal solution, and in a more realistic simulation this would have to be reconsidered in more detail.

When it comes to the mass m and the cross sectional reference area S, these have been estimated from data sheets obtained from the *NASA website* [22]. The aspect ratio is taken from *Scott and Olds* [21]. The following values have been used for the constant parameters:

- Mass $m = 200,000\,\mathrm{lbs}$,
- Cross sectional reference area $S = 362.1\,\mathrm{m}^2$,
- Aspect ratio $AR = 1.86$,
- Earth radius $a_e = 6378,137\,\mathrm{km}$,

- Gravitational constant $\mu = 3.9848 \cdot 10^{14}\,\mathrm{m}^3/\mathrm{s}^2$,
- Earth angular velocity $\Omega = 0.004167\,$degrees/s.

8.3.2 Solving by The Use of AE's

Because γ and A are give explicitely by the AE's (8.2.11) and (8.2.12), the DAE-system is solved in practice by substituting the two ODE's (8.2.9) and (8.2.10) with the AE's.

8.3.3 Solving by Index Reduction

Theoretically, the problem can also be solved by index reduction to order zero, which in practice means to differentiate the AE's several times. Because our system has the index 2, maximum two differentations are possible. The most straight forward way is to differentate untill each dynamical variable is represented by its own ODE. To obtain the two extra ODE's for finding α and β we would have to first differentiate (8.2.11) and (8.2.12) two times, and secondly we would need to differentiate (8.2.9) and (8.2.10) once, and by merging these four equations we would finally obtain the two extra ODE's. By the use of this procedure we would still be able to maintain the constraints by using the AE's to find γ and A explicitly. However it is not a good idea to reduce the DAE-system down to index 0, because this introduces numerical drift on the solution of α and β. And furthermore in this case it is impossible to isolate α' and β' because of nonlinearities in the dynamical equations. So instead we only reduce the system to index 1 and use Newton-Raphson to find the CP's.

8.3.4 Finding The Control Parameters α and β

By differentiating the constraints (8.2.11) and (8.2.12) one time the system is reduced to index 1, and the index reduced constraints become as follows:

$$\gamma' + 18\frac{t}{300^2} = 0\,, \tag{8.3.1}$$

$$A' - 180\frac{t}{300^2} = 0\,. \tag{8.3.2}$$

The control parameters α and β are both represented in (8.2.9) and (8.2.10), α is indirectly represented by the liftforce L, so by substituting γ' and A' the CP's can be found by the

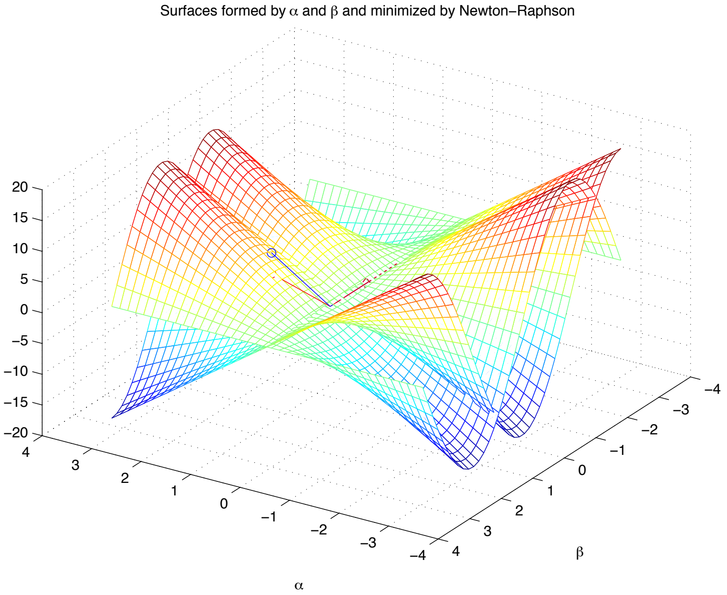


Figure 8.7: Minimizing (8.3.3) and (8.3.4) to find CP's.

equations:

$$\frac{L\cos(\beta)}{mV_R} + \frac{\cos(\gamma)}{V_R}\left(\frac{V_R^2}{r} - g\right) + 2\Omega\cos(\lambda)\sin(A) + \frac{\Omega^2 r\cos(\lambda)}{V_R}\Big(\sin(\lambda)\cos(A)\sin(\gamma) - \cos(\lambda)\cos(\gamma)\Big) + 18\frac{t}{300^2} = 0\,, \quad (8.3.3)$$

$$\frac{L\sin(\beta)}{mV_R\cos(\gamma)} + \frac{V_R}{r}\cos(\gamma)\sin(A)\tan(\lambda) - 2\Omega\Big(\cos(\lambda)\cos(A)\tan(\gamma) - \sin(\lambda)\Big) + \frac{\Omega^2 r\cos(\lambda)\sin(\lambda)\sin(A)}{V_R\cos(\gamma)} - 180\frac{t}{300^2} = 0\,. \quad (8.3.4)$$

As seen in the figures 8.7 and 8.8, (8.3.3) and (8.3.4) form two surfaces from α and β. So the CP's are found by minimizing (8.3.3) and (8.3.4) with Newton-Raphson, and the solution will be found where the two surfaces intersect. In the figures 8.7 and 8.8 the starting guesses of α and β are shown with circles, and the solution found by Newton-Raphson is shown with the diamond.

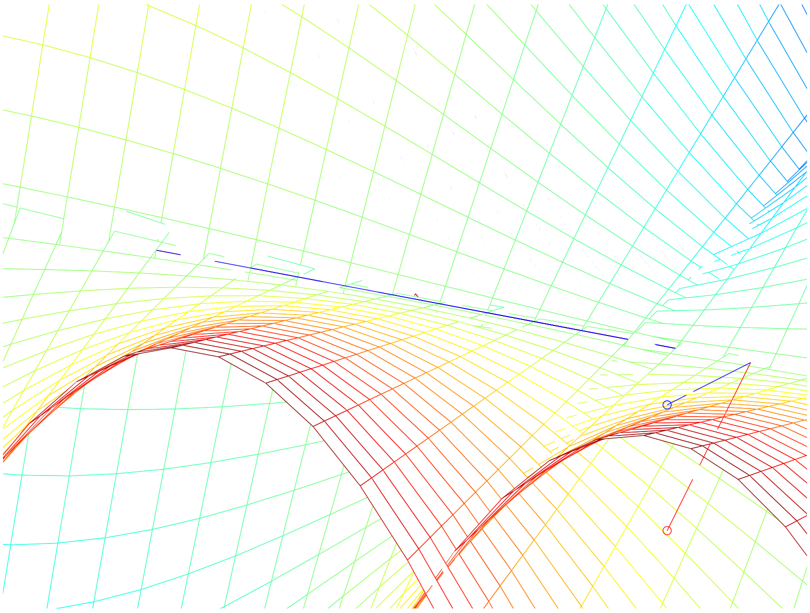


Figure 8.8: Close up of figure 8.7.

8.3.5 Numerical implementation

The procedure is divided into four main parts which are finding the physical parameters, the constraints (AE's), solving the ODE's and finding the CP's. A flow diagram for the program is seen in figure 8.9. To different integrators have been used, ERK4 with fixed time step and ESDIRK23 with variable time step. The two methods yield the very similar solutions, which could be expected, since we assumed our system not to be stiff (this being a reason to use a variable timestep). This assumption was made on the fact, that it would not be physical correct to have a shuttle to make sudden changes in the dynamical variables, and furthermore we where looking a smooth change in the variables. To use ESDIRK23 it is necessary to know the Jacobian of the system. Because of the coupling of the physical parameters it would be to complex to find the Jacobian analytically and probably not as accurate as desired, since the parameters are approximated by polynomial functions. Therefor we have implemented numerical approximation. This is however quite time consuming, so our choice of integrator is the ERK4 which does not use the jacobian. It must be mentioned, that all code is selfmade, so no Matlab functions are used.

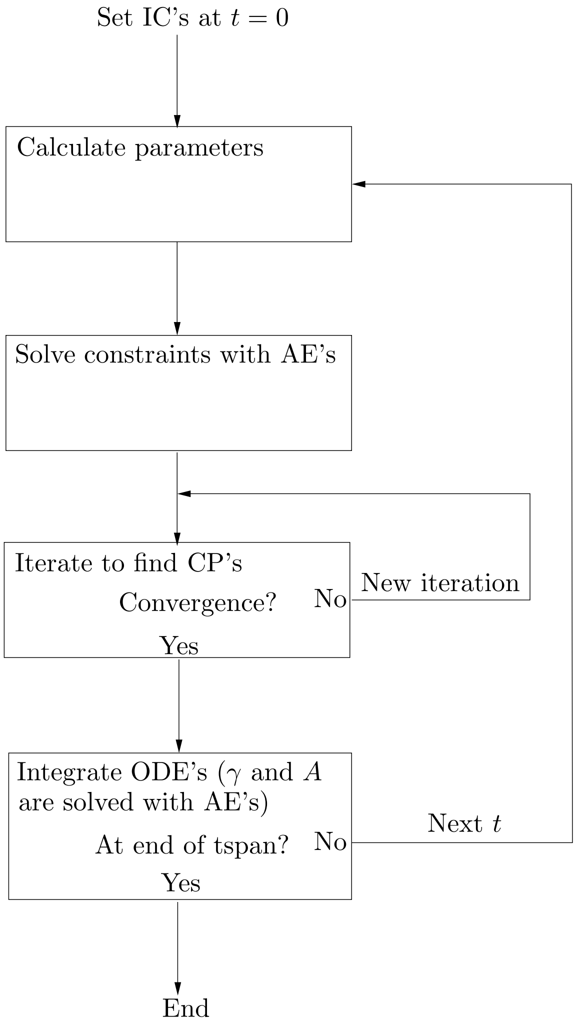


Figure 8.9: Flow diagram for the numerical simulation.

8.4 Results

In this section we will present the results from our program, and we will perform an analysis of the dependence of the error on the time step.

For our calculations we have used the following initial conditions:

- $H_0 = 100,000\,\text{feet} = 30,480\,\text{m}$ (height of the atmosphere)
- $V_{R0} = 12,000\,\text{feet/s} = 3657.6\,\text{m/s}$
- $\varepsilon_0 = 0°$ (Greenwich meridian)
- $\lambda_0 = 0°$ (Equator)
- $\gamma_0 = -1°$ (given by prescribed trajectory)
- $A = 45°$ (given by prescribed trajectory)

The initial guess for α and β is taken from *Brenan et al.* [20]; $\alpha_0 \simeq 2.67°$ and $\beta_0 \simeq -0.0522096°$. By use of Newton Raphson algorithm consistent values are found to be $\alpha_0 = 0.09001603930206°$ and $\beta_0 = -0.05221600531589°$. With these initial conditions and using the ERK4 with different time steps and a Newton-Raphson iteration the problem has been solved as described in figure 8.9.

In figures 8.10 and 8.11 the altitude and the velocity are shown. The shuttle descends from 30 km to 10 km. At the same time it decelerates from 3657.6 m/s or Mach 12 to 391.9 m/s which is nearly Mach 1. This shows that after all the vehicle does not reach the subsonic regime. Though not shown here λ is clearly controlled by the the A-part of the prescribed trajectory. When flying north-east in the beginning λ increases and begins to decrease only when the direction becomes south east. The last dynamic variable ε increases all the time corresponding to the vehicle flying in only eastern directions.

The results for α and β are shown in figures 8.12 and 8.13. Qualitatively the roll angle β corresponds very well to the result obtained in *Brenan et al.* [20]. The pitch angle α, however, decreases, and this is not in agreement with *Brenan et al.* [20]. One would expect the shuttle to increase its pitch angle, since this would increase drag, and thereby decelerate the shuttle altogether. This is probably due to the incoherence between different data and parameters that we have used.

8.4.1 Error analysis

We have used a time step of $h = 0.01\,\text{s}$ to calculate reference solution for the error analysis. We consider the system at the end of the simulation $t = 300\,\text{s}$ and calculate the absolute (abs) and relative (rel) error from the reference solution of a solution with a larger time

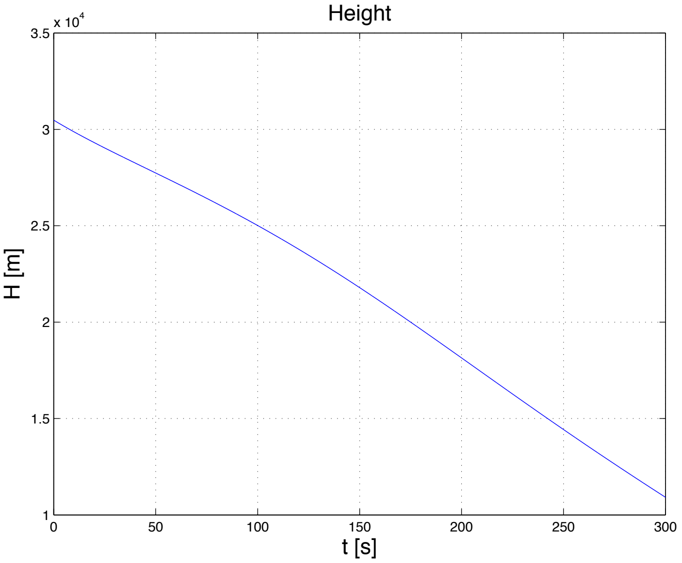


Figure 8.10: Altitude H as a function of time.

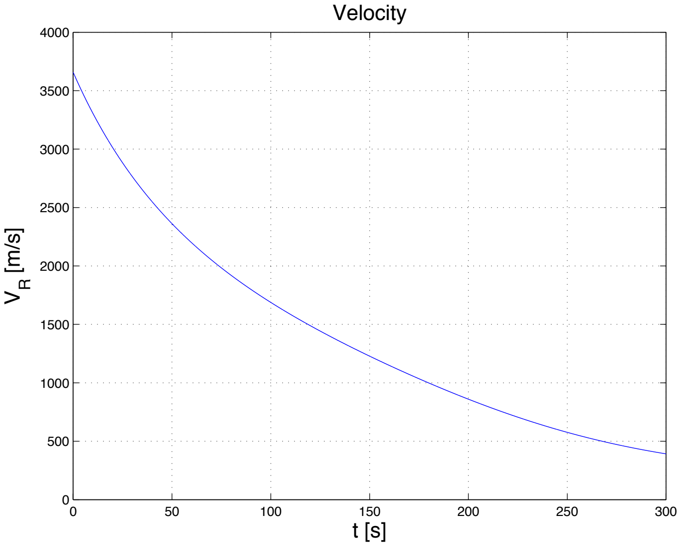


Figure 8.11: Velocity V_R as a function of time.

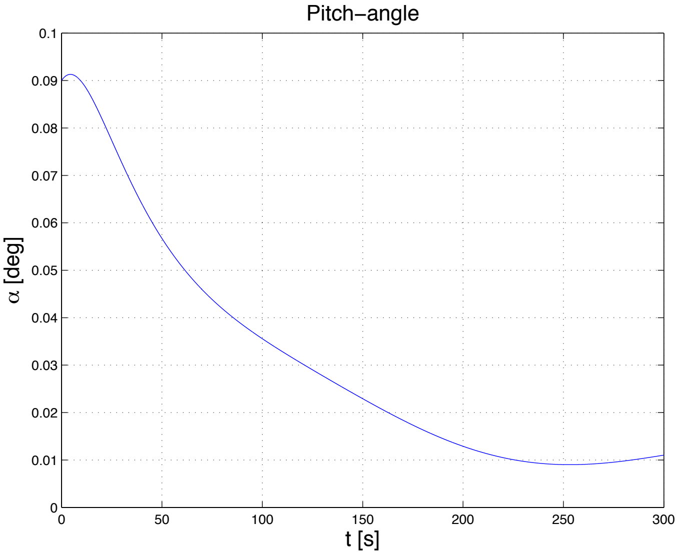


Figure 8.12: Angle of attack α as a function of time.

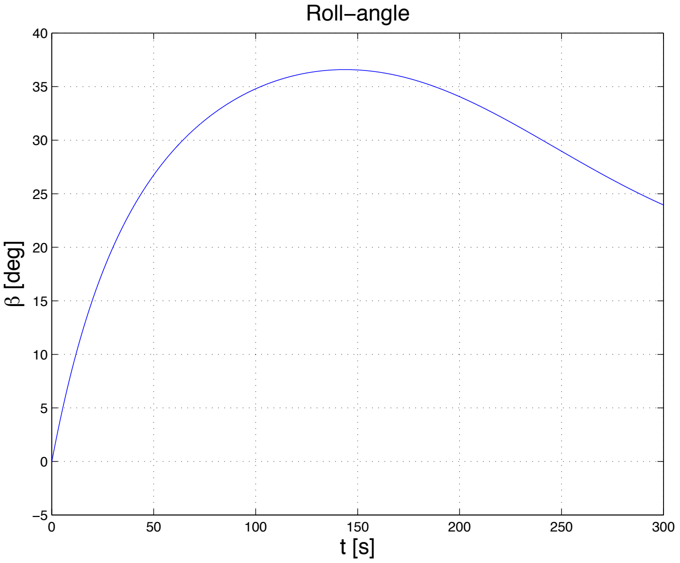


Figure 8.13: Roll angle β as a function of time.

	Relative error	Absolute error
H	$1.270 \cdot 10^{-3}$	$13.9\,\mathrm{m}$
ε	$0.140 \cdot 10^{-3}$	$0.0004°$
λ	$1.086 \cdot 10^{-3}$	$0.00193°$
V_R	$1.230 \cdot 10^{-3}$	$0.48\,\mathrm{m/s}$
α	$0.768 \cdot 10^{-3}$	$(8.4 \cdot 10^{-6})°$
β	$4.207 \cdot 10^{-3}$	$0.1°$

Table 8.1: Relative and absolute error for $h = 1\,\mathrm{s}$.

step. In table 8.1 the results for $h = 1\,\mathrm{s}$ are shown. The relative error is very similar for all variables, though considerably higher for the roll angle β. This could reflect that the roll angle being a degree of freedom that is very hard to control. The absolute errors on especially H, V_R, and β are quite higher.

8.5 Discussion

The simulations presented in the previous sections are based on the model proposed in [20]. One could question how realistic the constraints that are introduced actually are. The ones used are of course very simplified. From an intuitive point of view it would be more logical to introduce constraints on other variables like altitude and velocity. Another aspect is that in a more elaborated model one would have to consider the effects of thermal heating which mainly depend on the velocity and air density and thereby altitude. However, introducing constraints on for example altitude would complicate the system considerably by increasing the index to 3. This is because the altitude ODE does not directly depend on α nor β, and only by differentiating once more would it be possible to reduce the index of the system to the same as for the constraints that we have used.

In this problem it is convenient that the constraints are stated as functions of time. It allows for a reduction of the dimension of the dynamical part of the system from 6 to 4. But what happens if we do not use this advantage explicitly? This is simply done by integrating all six ODEs and using the values of A and γ obtained from the integration instead of the ones they are supposed to be. The results for γ is seen in figure 8.14 along with the prescribed trajectory. It is clear that the two trajectories are not alike, but they have the same slope in almost all points. This is due to the index reduction of the constraints. The CP's α and β are found using the time derivative of the original constraints, and therefore they only fulfill that the calculated trajectory must have the same slope. A constant of integration is lost when differentiating. By varying different parameters one can also observe that the trajectory jumps to a different path also with the same slope. These observations indicate that there are several stable trajectories, and this emphasizes the need for efficient and

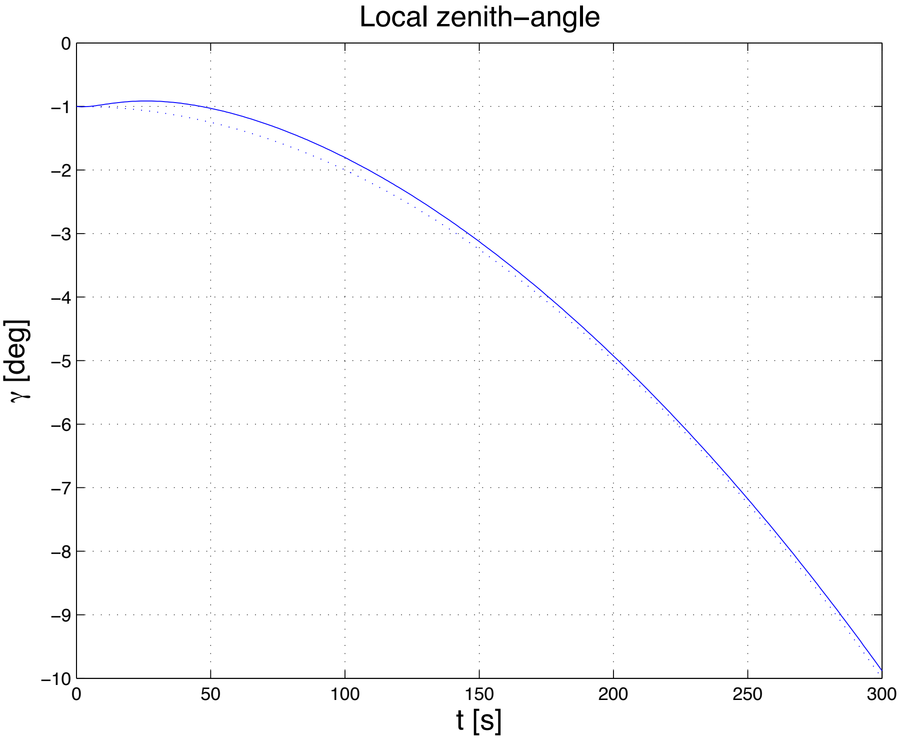


Figure 8.14: Prescribed (dotted line) and numerically integrated local zenith angle γ as a function of time.

reliable control in reentry problems.

8.6 Conclusion

TPPC problems very quickly become complicated. This is due to both the DAE formulation and the difficulties encountered in the science of aeronautics. What concerns the numerical approach, we have succeeded in implementing a simple and efficient method that combines a Runge-Kutta solver for the dynamical part and a Newton-Raphson iteration for the algebraic part. The error analysis shows that small time steps ($h = 0.01\,\mathrm{s}$) should be used, since in particular altitude, velocity, and roll angle deviate quite considerably.

The physical parameters for the vehicle that we have used must be far more consistent if the aim of the simulation is to obtain precise and realistic results. In our calculations the control parameter α behaves in an even counterintuitive way, and this is of course far from satisfactory. We suspect that the lift and drag coefficient in the hypersonic regime have a major influence on whether the simulation is realistic or not. In order to improve the calculations a complete dataset for a specific vehicle would be necessary.

Bibliography

[1] E.Hairer and G. Wanner, : Solving Ordinary Differential Equations II, Springer -Verlag, 2.nd edition, 1996.

[2] www.rafil-gmbh.de/produkte/gueter/gueterw.htm

[3] A. A. Shabana, M. Berzeri, J. R. Sany : Numerical Procedure for the Simulation of Wheel/Rail Contact Dynamics, ASME Vol. 123, 168–178, 2001.

[4] A. A. Shabana, J. R. Sany : An Augmented Formulation for Mechanical Systems with Non–Generalized Coordinates: Application to Rigid Body Contact Problems, Kluwer Academic Publishers, Nonlinear Dynamics 24, 183–204, 2001.

[5] A. A. Shabana : Dynamics of Multibody Systems (second edition), Cambridge University Press 1998.

[6] F. Pfeiffer, C. Glocker : Multibody Dynamics with Unilateral Contacts, John Wiley & Sons, 1996.

[7] E. Eich–Soellner, C. Führer : Numerical Methods in Multibody Dynamics, B.G. Teubner Stuttgart, 1998.

[8] L. F. Shampine *et al* : Fundamentals of Numerical Computing, John Wiley & Sons, 1997.

[9] Z. Y. Shen, J. K. Hedrick, J. A. Elkins : A Comparison of Alternative Creep Force Models for Rail Vehicle Dynamic Analysis, Proc. of 8^{th} IAVSD Symposium, 591–605, Cambridge, 1983.

[10] N. K. Cooperrider : The Hunting Behavior of Conventional Railway Trucks, ASME J. Eng. Industry 94, 752–762, 1972.

[11] C. Kaas–Petersen : Chaos in a Railway Bogie, ACTA Mechanica 61, 89–108, Springer–Verlag, 1986.

[12] J. J. Kalker : Three–Dimensional Elastic Bodies in Rolling Contact, Kluwer Academic Publishers, 1990.

[13] V. I. Arnold, *Mathematical methods of Classical Mechanics*, Springer-Verlag, 1980

[14] Per Grove Thomsen, *Solution methods for ODE's with changes of state*, IMM, Danmarks Tekniske Universitet, marts 2005

[15] J.D.Lambert, *Numerical Methods for Ordinary Differential Systems -The Initial Value Problem*, Wiley, 1991

[16] Patrick J Rabier, *On the Computation of Impasse Points of Quasi-Linear Differential-Algebraic Equations*, Mathematics of Computation, vol.62 no.205,january 1994

[17] R.E. O'Malley . *Introduction to Singular Perturbations.* Academic Press, New York. 1974.

[18] E.Hairer, Ch. Lubich and M.Roche . *The numerical solution of differential-algebraic systems by Runge-Kutta methods.* Lecture notes in Mathematics 1409, Springer Verlag . 1989.

[19] E. S. Andersen, P. Jespersgaard, and O. G. Østergaard. *Databog fysik & kemi*, volume 2nd. F&K forlaget, 2nd edition, 1983.

[20] K. E. Brenan, S. L. Campbell, and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations.* Number 14 in Classics in applied mathematics. Society for Industrial and Applied Mathematics - SIAM, 1996.

[21] J. M. Scott and J. R. Olds. Transforming aerodynamic datasets into parametric equations for use in multidisciplinary design optimization. In *Defense and Civil Space Programs Conference and Exhibit.* American Institute of Aeronautics and Astronautic - AIAA, October 1998.

[22] NASA website. `http://www.nasa.gov`.

[23] A. Schwab, J. Meijaard, J. Papadopoulos
Benchmark Results on the Linearized Equations of Motion of an Uncontrolled Bicycle
Proceedings of ACMD'04

[24] J. Papadopoulos
Bicycle Steering Dynamics and Self-Stability: A Summary Report on Work in Progress
Preliminary draft - Cornell University 1987

[25] R.S. Hand
Comparisons and Stability Analysis of Linearized Equations of Motion for a Basic Bicycle Model
MSc Thesis, Cornell University 1988

[26] Robin S. Sharp, David J.N. Limebeer
A Motorcycle Model for Stability and Control Analysis
Multibody System Dynamics 6-2001

[27] F.M.L. Amirouche
Computational Methods in Multibody Dynamics
Prentice-Hall 1992

[28] U.M. Ascher, H. Chin, S. Reich
Stabilization of DAEs and invariant manifolds
Numerische Mathematik 67:131-149 (1994)

[29] Editors : P.G.Thomsen, C. Bendtsen
Numerical Solution of Differential Algebraic Equations
Technical Report, 1999, IMM, DTU, IMM-REP-1999-8.