



ROYAL INSTITUTE
OF TECHNOLOGY

A computer-based approach to Lagrangian mechanics

SA114X Degree Project in Engineering Physics, First Level

FILIP STRAND AND JAKOB ARNOLDSSON

Bachelor's Thesis at the Department of Mechanics
Supervisor: Nicholas Apazidis
Examiner: Mårten Olsson

May 20th 2016

Abstract

Classical mechanics is the branch of physics concerned with describing the motion of bodies. The subject is based on three simple axioms relating forces and movement. These axioms were first postulated by Newton in the 17th century and are known as his three laws of motion.

Lagrangian mechanics is a restatement of the Newtonian formulation. It deals with energy quantities and paths-of-motion instead of forces. This often makes it simpler to use when working with non-trivial mechanical systems. In this thesis, we use the Lagrangian method to model two such systems; A rotating torus and a variant of the classical double pendulum.

It soon becomes clear that the complexity of these systems make them difficult to attack by hand. For this reason, we take a computer-based approach. We use a software-package called Sophia which is a plug-in to the computer algebra system MapleTM. Sofia was developed at the Department of Mechanics at KTH for the specific purpose of modeling mechanical problems using Lagrange's method. We demonstrate that this method can be successfully applied to the analysis of motion of complex mechanical systems. The complete equations of motion are derived in a symbolic form and then integrated numerically. The motion of the system is finally visualized by means of 3D graphics software BlenderTM.

Contents

1	Introduction	1
1.1	Background	1
1.2	Main objective	3
1.3	Problems	4
1.3.1	<i>Boston Hoop</i>	4
1.3.2	<i>Double pendulum with dual springs</i>	4
2	Theory	5
2.1	Lagrangian formalism: Definitions and notation	5
2.2	The Principle of Least Action	6
2.3	Special case of the Euler-Lagrange equations	6
2.4	General case of the Euler-Lagrange equations	8
2.4.1	Alternate formulation	10
2.4.2	Application to rigid bodies	11
2.5	Conservation laws	13
2.6	Chaos	15
2.7	Mechanics and geometry	15
2.7.1	Coordinate systems and rotation	16
2.7.2	Kinetic energy and moment of inertia	17
3	Method	19
3.1	Software	19
3.1.1	Maple and Sophia	19
3.1.2	Blender	19
3.2	General procedure	20
3.3	<i>Boston Hoop</i>	20
3.3.1	Geometry	20
3.3.2	Mechanics	21
3.3.2.1	Moment of inertia	22
3.3.2.2	Kinetic energy	22
3.3.2.3	Potential energy	23
3.3.2.4	Angular momentum	23
3.3.3	Cases	24

3.4	<i>Double pendulum with dual springs</i>	25
3.4.1	Geometry	25
3.4.2	Mechanics	27
3.4.2.1	Moment of inertia	27
3.4.2.2	Kinetic energy	27
3.4.2.3	Potential energy	28
3.4.2.4	Generalised forces and friction	29
3.4.3	Cases	32
4	Results	33
4.1	<i>Boston Hoop</i>	33
4.1.1	Generalized coordinates and phase plots	33
4.1.2	Conserved quantities	35
4.1.3	Energy- and angular momentum transfer between different parts of the system	36
4.2	<i>Double pendulum with dual springs</i>	37
4.2.1	Generalized coordinates and phase plots	37
4.2.2	Sensibility to initial conditions	39
4.2.2.1	Deviations due to numerical integration	42
4.2.3	Friction	45
5	Discussion	47
5.1	<i>Boston Hoop</i>	47
5.2	<i>Double pendulum with dual springs</i>	48
5.3	The power of this method	50
6	Summary and conclusion	51
Bibliography		52
Appendices		53
A	Maple and Sophia Code	53
A.1	<i>Boston hoop</i>	53
A.2	<i>Double pendulum with dual springs</i> (no friction)	56
A.3	<i>Double pendulum with dual springs</i> (friction)	59
B	Blender Code	63

Chapter 1

Introduction

This work is a Bachelor thesis in engineering physics at KTH in Stockholm Sweden at the department of mechanics, 2016. This thesis consist of a presentation of theory, course of action and analysis in solving complex mechanical systems. This is done by using computer algebra and analytical mechanics to study two complex systems and their movement over time with different types of initial conditions and with or without friction. The Maple based software Sofia developed at the department of mechanics at KTH is used to find and solve the differential equations for the motion of the two different systems [1].

1.1 Background

The analysis of mechanical systems isn't something new. Since the fifteenth century, starting with Newton's three laws of motion:

- **1st law:** When viewed in an inertial reference frame, an object either remains at rest or continues to move at a constant velocity, unless acted upon by a force.
- **2nd law:** The vector sum of the forces \mathbf{F} on an object is equal to the mass m of that object multiplied by the acceleration vector \mathbf{a} of the object: $\mathbf{F} = m\mathbf{a}$.
- **3rd law:** When one body exerts a force on a second body, the second body simultaneously exerts a force equal in magnitude and opposite in direction on the first body.

Scientists and engineers have been able to analyze and conclude important results of different aspects of mechanical systems. This not only for their behavior at a specific point in time but also over a period of time. This Newtonian approach is based on vector relations between vector quantities, which works fine for systems with simple geometry. But in 1788 a new and groundbreaking theory, that change the field of mechanics, was introduced in the book *Mecanique Analytique* written by

CHAPTER 1. INTRODUCTION

the Italian mathematician Joseph Lagrange [2]. It was the now famous Lagrangian method.

The new theory made it simple to derive the equations of motion for systems of rigid bodies or particles that in some way or another were connected to each other. Furthermore, this new way of looking at mechanical problems was, unlike the Newtonian theory, based on scalar quantities and their relations. In the Lagrangian method the vector quantity force is substituted by the scalar quantity energy. To derive the equations of motion one would no longer have to use vectors but instead differentiate the expression for energy of the system, which allow systems with more complex geometries to be solved efficiently. One of the main advantages that comes with the Lagrangian method is that one has the freedom to use arbitrary coordinates as long as they together describe the full configuration of the system. This together with the algorithmic structure of the method makes it a pioneering advancement in the field of mechanics.

The power and possible applications of the method, although great and significant, is also limited. Already for rather simple systems with several degrees of freedom the computations of the equations of motion rapidly reach unreasonable levels of difficulty. This before the task of solving these equations has even been dealt with, which often require numerical integration.

With the advancements in technology the field of computer algebra has developed and grown, and in past decades also made its entrance into the field of mechanics. Softwares as the KTH developed Sofia open new opportunities in the task for both computing and solving the equations of motion of a wide range of mechanical problems. Sofia is named after the Russian mathematician Sofia Kovalevskaya partially for her contributions in the mechanics behind the motion of rigid bodies about a fix point. The software is a specific system used to solve problems with systems of rigid bodies and their motion. It is not limited to the Lagrangian method but provide a powerful tool in solving complex mechanical problems with the Lagrangian method [1].

A quite simple and classical system of rigid bodies is the double pendulum and variations of it. Although this system can be thought of as a quite simple construction at first, it exhibits a complex pattern of motion. This makes it suitable as a system for analysing complex dynamics and to display the power that lies in the mix of Lagrangian mechanics and computer algebra.

Nevertheless, to show and analyze interesting aspects of the physics of a dynamic system, the actual system itself does not have to be as complex as the double pendulum. The *Boston Hoop* is a system which, although its rather simple 3D geometry can be solved by the Lagrangian method.

1.2 Main objective

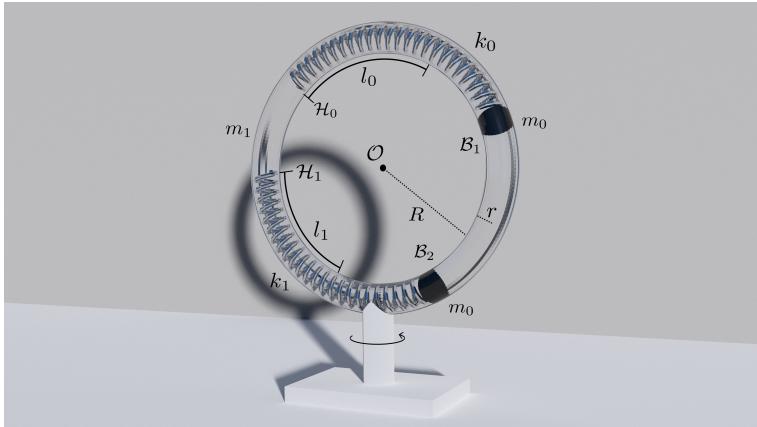
The main goals with this thesis is to successfully learn and apply the Lagrangian method on two complex systems using the Sofia software and study the motion of the systems. More precisely, conservation of quantities as energy and angular momentum, sensitivity of initial conditions (chaos) and the effect of dissipative forces will be analysed for the two systems.

Another goal of this work is to familiarise and learn to us the animation and physics based software Blender so that animations and figures of the systems and the results of the simulations in Sofia could be presented and intuitively analysed in an neat manner.

1.3 Problems

In this section, the two problem formulations along with 3D-renderings are presented.

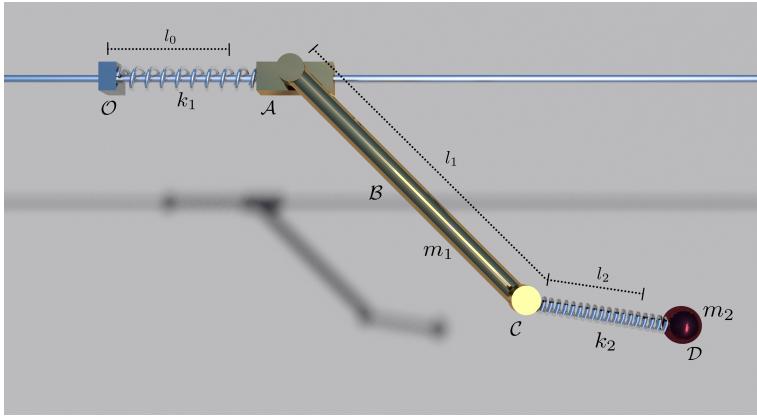
1.3.1 *Boston Hoop*



natural length l_0 and spring-constant k_0 . The other ball is attached to a spring fixed at \mathcal{H}_1 with natural length l_1 and spring-constant k_1 . We analyze conserved quantities of motion and look at the transfer of energy between different parts of the system through time.

Consider two balls of mass m_0 moving inside a smooth hollow torus with outer radius R , inner radius r and mass m_1 . The torus can rotate freely about the vertical symmetry axis. One of the balls is attached to a spring fixed at \mathcal{H}_0 with natural length l_0 and spring-constant k_0 . The other ball is attached to a spring fixed at \mathcal{H}_1 with natural length l_1 and spring-constant k_1 .

1.3.2 *Double pendulum with dual springs*



The rigid rod of mass m_1 (with center of mass at \mathcal{B}) and length l_1 is pivoted at \mathcal{A} so that it may rotate freely in the two-dimensional plane. The section at \mathcal{A} is also connected to a spring of natural length l_0 and spring-constant k_1 from point \mathcal{O} . The spring can move horizontally along the \mathcal{OA} . At point \mathcal{C} , another spring with natural length l_2 and spring-constant k_2 , is attached to the lower part of the massive rod. The other end of this spring is attached to a ball of mass m_2 at point \mathcal{D} . The second spring is restricted to move in the same plane as the massive rod. We analyze the system's sensibility to initial conditions and response to friction.

Chapter 2

Theory

In the following sections the theory is presented in more detail.

2.1 Lagrangian formalism: Definitions and notation

The following definitions are useful when working with the Lagrangian formalism:

Generalized coordinate

A generalized coordinate, labeled as $q_i, i = 1, \dots, n$, is a scalar parameter used to describe (parts of) the configuration of a system. Some examples are the usual three-dimensional Cartesian coordinates ($q_1 = x, q_2 = y, q_3 = z$), or the cylindrical-coordinate system ($q_1 = r, q_2 = \theta, q_3 = z$) etc. In general, there are many different choices of generalized coordinates for describing a single system.

Configuration-space

The configuration-space is the space of all generalized coordinates.

Configuration-path

By a configuration-path, which we denote $\mathbf{q}(t)$, we mean a time dependent coordinate-tuple

$$\mathbf{q}(t) = [\mathbf{q}] = \begin{bmatrix} q_1(t) \\ q_2(t) \\ \vdots \\ q_n(t) \end{bmatrix} \quad (2.1)$$

As time runs, this traces out a path in the configuration space. This should not be confused with the actual-path.

Actual-path

The actual-path is the path in real-space that the particle, or system of particles, actually takes. It is denoted by \mathbf{r} and is sometimes also referred to as the *position*

CHAPTER 2. THEORY

vector. In general, \mathbf{r} is a time dependent, three dimensional vector expressed in some coordinate system

$$\mathbf{r}(t, q_1(t), \dots, q_n(t)) = [\mathbf{r}] = \begin{bmatrix} f_1(t, q_1(t), \dots, q_n(t)) \\ f_2(t, q_1(t), \dots, q_n(t)) \\ f_3(t, q_1(t), \dots, q_n(t)) \end{bmatrix} \quad (2.2)$$

As we see, \mathbf{r} depends explicitly on the generalized coordinates. The generalized coordinates themselves depend on time, making \mathbf{r} implicitly time dependent.¹

2.2 The Principle of Least Action

The fundamental concept underlying the Lagrangian formalism is the Principle of Least Action. It states that:

*Out of all possible paths of motion, $\mathbf{q}(t)$, a body can take from time t_1 to t_2 , the realizable path is the one which minimizes the action S , where*²

$$S[q_1, \dots, q_n] = \int_{t_1}^{t_2} \mathcal{L}(q_1, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n) dt \quad (2.3)$$

Here, \mathcal{L} is known as the *Lagrangian* of the system. It can be shown that if $\mathcal{L} = T - V$, where T is the kinetic energy and V is the potential energy of the system, this statement ultimately reduces down to Newton's equations of motion.

However, because the principle is phrased as an optimization problem, it only provides an implicit statement for the correct path of motion. This is not very useful for direct calculations. Fortunately, it turns out that if a path minimizes the action, it must also satisfy a set of differential equations known as the *Euler-Lagrange equations*.[5] As we will see, this solves our problem.

2.3 Special case of the Euler-Lagrange equations

The special case of the Euler-Lagrange equations can be written [3]

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i} = 0 \quad (2.4)$$

These equations are a direct consequence of the principle of least action. They can also be derived from the more general case as we will see later. We show the derivation for a two-dimensional configuration-path but it can be generalized to any dimension. Suppose we have the following configuration-path $\mathbf{q}(t)$:

$$\mathbf{q}(t) = \begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix} \quad (2.5)$$

¹In general, \mathbf{r} may also explicitly depend on time

²A more proper name would be 'The principle of *stationary* action' as the theory formally requires an extremum of S which can also include saddle-points.

CHAPTER 2. THEORY

We can vary the path by adding a small perturbation $\boldsymbol{\eta}(t)$ so that

$$\mathbf{q}(t) + \boldsymbol{\eta}(t) = \begin{bmatrix} q_1(t) + \eta_1(t) \\ q_2(t) + \eta_2(t) \end{bmatrix}, \quad \dot{\mathbf{q}}(t) + \dot{\boldsymbol{\eta}}(t) = \begin{bmatrix} \dot{q}_1(t) + \dot{\eta}_1(t) \\ \dot{q}_2(t) + \dot{\eta}_2(t) \end{bmatrix} \quad (2.6)$$

We want to study all possible paths from point $\mathbf{q}(t_1)$ to $\mathbf{q}(t_2)$. Formally, this means that $\boldsymbol{\eta}(t_1) = \boldsymbol{\eta}(t_2) = \mathbf{0}$. $\boldsymbol{\eta}(t)$ can be any path as long as this condition is satisfied. We now make the main claim by forcing the action to be stationary³

$$0 = \delta S \quad (2.7)$$

$$= S[(q_1 + \eta_1), (q_2 + \eta_2)] - S[q_1, q_2] \quad (2.8)$$

$$= \int_{t_1}^{t_2} \mathcal{L}\left((q_1 + \eta_1), (q_2 + \eta_2), (\dot{q}_1 + \dot{\eta}_1), (\dot{q}_2 + \dot{\eta}_2)\right) - \mathcal{L}(q_1, q_2, \dot{q}_1, \dot{q}_2) dt \quad (2.9)$$

The Lagrangian $\mathcal{L}(q_1, q_2, \dot{q}_1, \dot{q}_2)$ is a standard multidimensional function and can be Taylor-expanded as such

$$\begin{aligned} \mathcal{L}\left((q_1 + \Delta q_1), (q_2 + \Delta q_2), (\dot{q}_1 + \Delta \dot{q}_1), (\dot{q}_2 + \Delta \dot{q}_2)\right) &= \mathcal{L}(q_1, q_2, \dot{q}_1, \dot{q}_2) + \\ &+ \frac{\partial \mathcal{L}}{\partial q_1} \Delta q_1 + \frac{\partial \mathcal{L}}{\partial q_2} \Delta q_2 + \frac{\partial \mathcal{L}}{\partial \dot{q}_1} \Delta \dot{q}_1 + \frac{\partial \mathcal{L}}{\partial \dot{q}_2} \Delta \dot{q}_2 + (\mathcal{O}^2) \end{aligned}$$

We only care about first order approximations when comparing the action of two neighboring paths. The first term of (2.9) can thus be rewritten using the linear terms in the Taylor-expansion. In this case, $\Delta q_1, \Delta q_2, \Delta \dot{q}_1, \Delta \dot{q}_2$ must clearly equal $\eta_1, \eta_2, \dot{\eta}_1, \dot{\eta}_2$ respectively. Doing this substitution and some simplification yields

$$0 = \int_{t_1}^{t_2} \underbrace{\frac{\partial \mathcal{L}}{\partial q_1} \eta_1(t)}_{\boxed{1}} + \underbrace{\frac{\partial \mathcal{L}}{\partial q_2} \eta_2(t)}_{\boxed{2}} + \underbrace{\frac{\partial \mathcal{L}}{\partial \dot{q}_1} \dot{\eta}_1(t)}_{\boxed{1}} + \underbrace{\frac{\partial \mathcal{L}}{\partial \dot{q}_2} \dot{\eta}_2(t)}_{\boxed{2}} dt \quad (2.10)$$

The terms $\boxed{1}$ and $\boxed{2}$ can be rewritten using partial integration

$$\boxed{1}: \quad \int_{t_1}^{t_2} \frac{\partial \mathcal{L}}{\partial \dot{q}_1} \dot{\eta}_1(t) dt = \underbrace{\left[\frac{\partial \mathcal{L}}{\partial \dot{q}_1} \eta_1(t) \right]_{t_1}^{t_2}}_0 - \int_{t_1}^{t_2} \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_1} \right) \eta_1(t) dt \quad (2.11)$$

$$\boxed{2}: \quad \int_{t_1}^{t_2} \frac{\partial \mathcal{L}}{\partial \dot{q}_2} \dot{\eta}_2(t) dt = \underbrace{\left[\frac{\partial \mathcal{L}}{\partial \dot{q}_2} \eta_2(t) \right]_{t_1}^{t_2}}_0 - \int_{t_1}^{t_2} \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_2} \right) \eta_2(t) dt \quad (2.12)$$

³This is also known as Hamilton's principle

CHAPTER 2. THEORY

Because of the restrictions $\boldsymbol{\eta}(t_1) = \boldsymbol{\eta}(t_2) = \mathbf{0}$, the endpoints vanish. Substituting the non-zero terms of (2.11) and (2.12) into (2.10) and factoring out $\eta_1(t)$ and $\eta_2(t)$ gives

$$0 = \int_{t_1}^{t_2} \left(\frac{\partial \mathcal{L}}{\partial q_1} - \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_1} \right) \right) \eta_1(t) + \left(\frac{\partial \mathcal{L}}{\partial q_2} - \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_2} \right) \right) \eta_2(t) dt \quad (2.13)$$

The integral above can only stay zero for all possible choices of $\eta_1(t), \eta_2(t)$ if the integrand is identically zero. Thus, we get that

$$0 = \left(\frac{\partial \mathcal{L}}{\partial q_1} - \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_1} \right) \right) \eta_1(t) + \left(\frac{\partial \mathcal{L}}{\partial q_2} - \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_2} \right) \right) \eta_2(t) \quad (2.14)$$

This equation can only hold for all $\eta_1(t), \eta_2(t)$ if:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial q_1} - \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_1} \right) &= 0 \\ \frac{\partial \mathcal{L}}{\partial q_2} - \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_2} \right) &= 0 \end{aligned} \quad (2.15)$$

which are the Euler-Lagrange equations for a two-dimensional path.⁴

This special case of the Euler-Lagrange equations are the equations of motion for a system subjected exclusively to *conservative* forces (this is shown in the next section). As stated above, they can now be integrated to find the path of motion for some set of initial conditions.

The Euler-Lagrange equations also bridge an important philosophical gap arising from the least action statement: The principle effectively talks about a body examining all possible paths to finally settle on the one which minimize the action. Trying out every possible path in reality would not be very practical however. With these equations, we have thus saved ourselves a great deal of trouble by instead relying on analysis to provide a recipe for what the time-evolution of the correct path necessarily must be.

2.4 General case of the Euler-Lagrange equations

The general Euler-Lagrange equations can be written as [4]:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_i} \right) - \frac{\partial T}{\partial q_i} = \sum_{j=1}^N \mathbf{F}_j \cdot \frac{\partial \mathbf{r}_j}{\partial q_i} \quad (2.16)$$

There are two main differences compared to the special case above: The right-hand side is non-zero and we have T in place of \mathcal{L} . $\sum_{j=1}^N \mathbf{F}_j \cdot \frac{\partial \mathbf{r}_j}{\partial q_i}$ are known as the *generalized forces* and are explained in the derivation below.

⁴If we instead have n generalized coordinates, there would be n Euler-Lagrange equations.

CHAPTER 2. THEORY

This more general statement is useful when working with systems that include non-conservative forces, such as friction. Unlike the special case, the general Euler-Lagrange equations do not follow from the principle of least-action. However, we later show how the special case can be derived from the general statement by considering only conservative forces.

We derive the equations for a system of N particles described by n generalized coordinates. The position vector for the j :th particle in the system can in general be written as

$$\mathbf{r}_j(t, q_1, q_2, \dots, q_n) = [\mathbf{r}_j] = \begin{bmatrix} f_j(t, q_1, q_2, \dots, q_n) \\ g_j(t, q_1, q_2, \dots, q_n) \\ h_j(t, q_1, q_2, \dots, q_n) \end{bmatrix} \quad (2.17)$$

We begin by looking at Newton's second law for the j :th particle:

$$[\mathbf{F}_j] = m_j [\mathbf{a}_j] \quad (2.18)$$

$$= m_j \left[\frac{d^2 \mathbf{r}_j}{dt^2} \right] \quad (2.19)$$

Taking the dot-product of both sides of (2.19) with $\frac{\partial \mathbf{r}_j}{\partial q_i}$ yields

$$[\mathbf{F}_j] \cdot \left[\frac{\partial \mathbf{r}_j}{\partial q_i} \right] = m_j \left[\frac{d^2 \mathbf{r}_j}{dt^2} \right] \cdot \left[\frac{\partial \mathbf{r}_j}{\partial q_i} \right] \quad (2.20)$$

We can rewrite the dot-product of the RHS of (2.20) using the product rule

$$\left[\frac{d^2 \mathbf{r}_j}{dt^2} \right] \cdot \left[\frac{\partial \mathbf{r}_j}{\partial q_i} \right] = \frac{d}{dt} \left(\left[\frac{d \mathbf{r}_j}{dt} \right] \cdot \left[\frac{\partial \mathbf{r}_j}{\partial q_i} \right] \right) - \left[\frac{d \mathbf{r}_j}{dt} \right] \cdot \frac{d}{dt} \left[\frac{\partial \mathbf{r}_j}{\partial q_i} \right] \quad (2.21)$$

$$= \frac{d}{dt} \left([\mathbf{v}_j] \cdot \underbrace{\left[\frac{\partial \mathbf{r}_j}{\partial q_i} \right]}_{\mathbf{I}} \right) - [\mathbf{v}_j] \cdot \left[\frac{\partial \mathbf{v}_j}{\partial q_i} \right] \quad (2.22)$$

In (2.22) we have changed the order of differentiation in the last factor of the last term and replaced all $\frac{d \mathbf{r}_j}{dt}$ for \mathbf{v}_j . We can rewrite $[\mathbf{I}]$ as $\frac{\partial \mathbf{v}_j}{\partial q_i}$, to show this we look at

$$\mathbf{v}_j(t, q_1, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n) = \left[\frac{d \mathbf{r}_j}{dt} \right] \quad (2.23)$$

$$= \left[\frac{\partial \mathbf{r}_j}{\partial t} \right] \underbrace{\frac{dt}{dt}}_1 + \left[\frac{\partial \mathbf{r}_j}{\partial q_1} \right] \underbrace{\frac{dq_1}{dt}}_{\dot{q}_1} + \left[\frac{\partial \mathbf{r}_j}{\partial q_2} \right] \underbrace{\frac{dq_2}{dt}}_{\dot{q}_2} + \dots \quad (2.24)$$

$$= \left[\frac{\partial \mathbf{r}_j}{\partial t} \right] + \sum_{k=1}^n \left[\frac{\partial \mathbf{r}_j}{\partial q_k} \right] \dot{q}_k \quad (2.25)$$

CHAPTER 2. THEORY

Taking $\frac{\partial}{\partial \dot{q}_i}$ of the last equation gives us:

$$\left[\frac{\partial \mathbf{v}_j}{\partial \dot{q}_i} \right] = \left[\frac{\partial \mathbf{r}_j}{\partial q_i} \right] \quad (2.26)$$

Inserting this back into (2.22) gives us:

$$\left[\frac{d^2 \mathbf{r}_j}{dt^2} \right] \cdot \left[\frac{\partial \mathbf{r}_j}{\partial q_i} \right] = \frac{d}{dt} \left(\left[\mathbf{v}_j \right] \cdot \left[\frac{\partial \mathbf{v}_j}{\partial \dot{q}_i} \right] \right) - \left[\mathbf{v}_j \right] \cdot \left[\frac{\partial \mathbf{v}_j}{\partial q_i} \right] \quad (2.27)$$

$$= \frac{d}{dt} \left(\frac{1}{2} \frac{\partial}{\partial \dot{q}_i} \left\{ \left[\mathbf{v}_j \right] \cdot \left[\mathbf{v}_j \right] \right\} \right) - \frac{1}{2} \frac{\partial}{\partial q_i} \left\{ \left[\mathbf{v}_j \right] \cdot \left[\mathbf{v}_j \right] \right\} \quad (2.28)$$

where we just use the product rule again on both terms.

Inserting this back into (2.20) gives

$$\left[\mathbf{F}_j \right] \cdot \left[\frac{\partial \mathbf{r}_j}{\partial q_i} \right] = m_j \frac{d}{dt} \left(\frac{1}{2} \frac{\partial}{\partial \dot{q}_i} \left\{ \left[\mathbf{v}_j \right] \cdot \left[\mathbf{v}_j \right] \right\} \right) - m_j \frac{1}{2} \frac{\partial}{\partial q_i} \left\{ \left[\mathbf{v}_j \right] \cdot \left[\mathbf{v}_j \right] \right\} \quad (2.29)$$

$$= \frac{d}{dt} \left(\frac{\partial}{\partial \dot{q}_i} \underbrace{\left\{ \frac{1}{2} m_j \left[\mathbf{v}_j \right] \cdot \left[\mathbf{v}_j \right] \right\}}_{T_j} \right) - \frac{\partial}{\partial q_i} \underbrace{\left\{ \frac{1}{2} m_j \left[\mathbf{v}_j \right] \cdot \left[\mathbf{v}_j \right] \right\}}_{T_j} \quad (2.30)$$

This gives us the general Euler-Lagrange equations for the j :th particle

$$\frac{d}{dt} \left(\frac{\partial T_j}{\partial \dot{q}_i} \right) - \frac{\partial T_j}{\partial q_i} = \mathbf{F}_j \cdot \frac{\partial \mathbf{r}_j}{\partial q_i} \quad (2.31)$$

Finally, we sum over all N particles to obtain

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_i} \right) - \frac{\partial T}{\partial q_i} = \sum_{j=1}^N \mathbf{F}_j \cdot \frac{\partial \mathbf{r}_j}{\partial q_i} \quad (2.32)$$

where $T = \sum_{j=1}^N T_j$ is the total kinetic energy of the system.

2.4.1 Alternate formulation

If we make a distinction between conservative⁵- and non-conservative forces, we can write the RHS of (2.32) as⁶

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_i} \right) - \frac{\partial T}{\partial q_i} = - \frac{\partial V}{\partial q_i} + \sum_j^N \mathbf{F}_j^{NC} \cdot \frac{\partial \mathbf{r}_j}{\partial q_i} \quad (2.33)$$

⁵Conservative forces are forces that can be written as minus the gradient of the potential function V .

⁶Note that V is the *total* potential energy for the system - it may consist of several parts such as a gravitational potential, spring-potential etc. In general, the term $\frac{\partial V}{\partial q_i}$ may thus be a sum of many terms from different potentials.

CHAPTER 2. THEORY

where \mathbf{F}_j^{NC} is a non-conservative force. We now construct a function \mathcal{L} so that

$$\mathcal{L}(q_1, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n) = T(q_1, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n) - V(q_1, \dots, q_n) \quad (2.34)$$

By substituting $T = \mathcal{L} + V$ into (2.33), we get

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i} - \frac{\partial V}{\partial q_i} = -\frac{\partial V}{\partial q_i} + \sum_j^N \mathbf{F}_j^{NC} \cdot \frac{\partial \mathbf{r}_j}{\partial q_i} \quad (2.35)$$

which simplifies to

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i} = \sum_j^N \mathbf{F}_j^{NC} \cdot \frac{\partial \mathbf{r}_j}{\partial q_i} \quad (2.36)$$

which is an alternate formulation of the general Euler-Lagrange equations. One advantage with this form over (2.32) is that it removes the need to explicitly state conservative forces since they have already been accounted for in the Lagrange function.

We can now easily arrive at the special case of the Euler-Lagrange equations by considering a case with only conservative forces. This means that $\sum_j^N \mathbf{F}_j^{NC} \cdot \frac{\partial \mathbf{r}_j}{\partial q_i} = 0$ and we recover (2.4).

2.4.2 Application to rigid bodies

We can easily confirm that the general Euler-Lagrange equations is valid even for rigid bodies. The equations of motion for a two-dimensional rigid body moving in the xy a plane are given by:

$$\begin{bmatrix} F_x \\ F_y \end{bmatrix} = m \begin{bmatrix} a_{G_x} \\ a_{G_y} \end{bmatrix} \quad (2.37)$$

$$\frac{d}{dt} (I_G \omega_z) = M_z \quad (2.38)$$

The most general two dimensional rigid body can be described using three generalized coordinates (two for x, y translation and the other for rotation). Suppose the center of mass G is located at $[q_1(t), q_2(t)]^T$ and that the body can be rotated an angle $q_3(t)$ about the z -axis. An arbitrary point on the body can be specified with position vector \mathbf{r}_j :

$$\mathbf{r}_j = \begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix} + \begin{bmatrix} \cos(q_3(t)) & -\sin(q_3(t)) \\ \sin(q_3(t)) & \cos(q_3(t)) \end{bmatrix} \begin{bmatrix} f_1(u_j, v_j) \\ f_2(u_j, v_j) \end{bmatrix} \quad (2.39)$$

CHAPTER 2. THEORY

where $f_1(u_j, v_j)$ and $f_2(u_j, v_j)$ are arbitrary functions of the two parameters u_j, v_j .⁷
The general Euler-Lagrange equations for this system can be stated as follows

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_1} \right) - \frac{\partial T}{\partial q_1} = \sum_{j=1}^N \mathbf{F}_j \cdot \frac{\partial \mathbf{r}_j}{\partial q_1} \quad (2.40)$$

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_2} \right) - \frac{\partial T}{\partial q_2} = \sum_{j=1}^N \mathbf{F}_j \cdot \frac{\partial \mathbf{r}_j}{\partial q_2} \quad (2.41)$$

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_3} \right) - \frac{\partial T}{\partial q_3} = \sum_{j=1}^N \mathbf{F}_j \cdot \frac{\partial \mathbf{r}_j}{\partial q_3} \quad (2.42)$$

The RHS of (2.40) and (2.41) can be written as

$$\sum_{j=1}^N \mathbf{F}_j \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = F_x \quad (2.43)$$

$$\sum_{j=1}^N \mathbf{F}_j \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = F_y \quad (2.44)$$

where F_x and F_y are the net-forces in the respective directions. The kinetic energy T for the body can be written as

$$T = \frac{1}{2} m \mathbf{v}_{\mathcal{G}} \cdot \mathbf{v}_{\mathcal{G}} + \frac{1}{2} I_{\mathcal{G}} \omega^2 \quad (2.45)$$

$$= \frac{1}{2} m (\dot{q}_1^2(t) + \dot{q}_2^2(t)) + \frac{1}{2} I_{\mathcal{G}} \dot{q}_3^2(t) \quad (2.46)$$

The LHS of (2.40) and (2.41) can be written as

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_1} \right) - \frac{\partial T}{\partial q_1} = m \ddot{q}_1(t) \quad (2.47)$$

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_2} \right) - \frac{\partial T}{\partial q_2} = m \ddot{q}_2(t) \quad (2.48)$$

Thus (2.40) and (2.41) are the first vector equation

$$m \ddot{q}_1(t) = F_x \quad (2.49)$$

$$m \ddot{q}_2(t) = F_y \quad (2.50)$$

The LHS of (2.42) can be computed as

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_3} \right) - \frac{\partial T}{\partial q_3} = \frac{d}{dt} (I_{\mathcal{G}} \dot{q}_3(t)) \quad (2.51)$$

⁷For example, a circle of radius R has $f_1(u, v) = u \cdot \cos(v)$ and $f_2(u, v) = u \cdot \sin(v)$, where $u = [0, R], v = [0, 2\pi]$.

CHAPTER 2. THEORY

We now show that the last equation will give momentum-equation (2.38). Taking the partial derivative in the RHS of (2.42) gives us

$$\mathbf{F}_j \cdot \frac{\partial \mathbf{r}_j}{\partial q_3} = \begin{bmatrix} F_{x,j} \\ F_{y,j} \end{bmatrix} \cdot \begin{bmatrix} -\sin(q_3(t)) & -\cos(q_3(t)) \\ \cos(q_3(t)) & -\sin(q_3(t)) \end{bmatrix} \begin{bmatrix} f_1(u_j, v_j) \\ f_2(u_j, v_j) \end{bmatrix} \quad (2.52)$$

We can show that this indeed equals the moment around \mathcal{G} (in the z direction) due to the force \mathbf{F}_j applied at \mathbf{r}_j ⁸

$$M_{z,j} = \mathbf{r}_{\mathcal{G} \rightarrow j} \times \mathbf{F}_j \quad (2.53)$$

$$= \begin{vmatrix} f_1\mathcal{C}(q_3(t)) - f_2\mathcal{S}(q_3(t)) & f_1\mathcal{S}(q_3(t)) + f_2\mathcal{C}(q_3(t)) \\ F_{x,j} & F_{y,j} \end{vmatrix} \quad (2.54)$$

$$= F_{y,j} \left\{ f_1\mathcal{C}(q_3(t)) - f_2\mathcal{S}(q_3(t)) \right\} - F_{x,j} \left\{ f_1\mathcal{S}(q_3(t)) + f_2\mathcal{C}(q_3(t)) \right\} \quad (2.55)$$

$$= F_{x,j} \left\{ -f_1\mathcal{S}(q_3(t)) - f_2\mathcal{C}(q_3(t)) \right\} + F_{y,j} \left\{ f_1\mathcal{C}(q_3(t)) - f_2\mathcal{S}(q_3(t)) \right\} \quad (2.56)$$

We see that (2.56) is indeed equal to (2.52). This means that

$$\mathbf{F}_j \cdot \frac{\partial \mathbf{r}_j}{\partial q_3} = M_{z,j} \quad (2.57)$$

Summing over all N forces gives us the total momentum M_z around \mathcal{G}

$$\sum_{j=1}^N \mathbf{F}_j \cdot \frac{\partial \mathbf{r}_j}{\partial q_3} = M_z \quad (2.58)$$

Finally joining (2.51) and (2.58) gives us the momentum equation

$$\frac{d}{dt} (I_{\mathcal{G}} \dot{q}_3(t)) = M_z \quad (2.59)$$

2.5 Conservation laws

One of the most fundamental principles in physics is the laws of conserved quantities. A conserved quantity is a quantity that is constant in time. All quantities in physics are not conserved and those who are, are not always conserved under arbitrary conditions.

The conserved quantities in physics that are relevant for classical mechanical systems are for example energy, momentum and angular momentum. These are

⁸We use the short-hand notation f_1 , f_2 for $f_1(u_j, v_j)$, $f_2(u_j, v_j)$ and $\mathcal{C}(q_3(t))$, $\mathcal{S}(q_3(t))$ for $\cos(q_3(t))$ and $\sin(q_3(t))$ respectively. Note also that when the moment around \mathcal{G} is calculated we do not want to use the absolute position vector \mathbf{r}_j but rather $\mathbf{r}_{\mathcal{G} \rightarrow j}$ which is the position vector relative to the center of mass. Formally $\mathbf{r}_{\mathcal{G} \rightarrow j} = \mathbf{r}_j - \mathbf{r}_{\mathcal{G}} = \mathbf{r}_j - [q_1(t), q_2(t)]^T$.

CHAPTER 2. THEORY

conserved in an isolated system, a system that does not interact with the surrounding environment in any way. Isolated systems are today not known to actually exist in reality. There is no such thing as a shield against gravity for example; it is infinite in range. But it is useful when focusing on basic principles in the physics of a system or in clarify the laws of physics. Furthermore the laws of conservation can, although the theoretical assumption of an isolated system, be presumed to be exact under certain circumstances. One such law is the law of conservation of energy, which applies for all classical mechanical problems as long as no dissipative forces exists, as for example friction [6].

In this work the conservation of energy and angular momentum will be analysed and as already mentioned the energy of the two systems in this work should be constant in the non-dissipative case. In the case of conservation of angular momentum it will be constant in both magnitude and direction and is considered as an absolute symmetry of nature. But this claim assumes that the net torque on the system is zero and that the interactions between the particles in the system is along the lines joining them together.

To prove this let m_i denote the mass of particle i and \mathbf{r}_i denote the position of particle i , then the angular momentum for the system, with respect to a fix point O , is given by

$$\mathbf{H}_O = \sum_i \mathbf{r}_i \times (m_i \dot{\mathbf{r}}_i) \quad (2.60)$$

with time derivative

$$\dot{\mathbf{H}}_O = \sum_i \mathbf{r}_i \times \mathbf{F}_i \quad (2.61)$$

where we have \mathbf{F}_i as the net force on each particle in the system. If the net force now is divide into the external \mathbf{F}_i^e and internal forces \mathbf{f}_{ij} , the force on particle i due to particle j , as

$$\mathbf{F}_i = \mathbf{F}_i^e + \sum_j \mathbf{f}_{ij} \quad (2.62)$$

and we get

$$\dot{\mathbf{H}}_O = \sum_i \mathbf{r}_i \times \mathbf{F}_i^e + \sum_{ij} \mathbf{r}_i \times \mathbf{f}_{ij} \quad (2.63)$$

Now, by Newtons third law we have $\mathbf{f}_{ij} = -\mathbf{f}_{ji}$ which means that the second term above vanishes provided that the interactions between the particles point along the lines joining them. This leaves

$$\dot{\mathbf{H}}_O = \sum_i \mathbf{r}_i \times \mathbf{F}_i^e = \mathbf{M}_O \quad (2.64)$$

where \mathbf{M}_O is precisely the expression of the external torque on the system. Thus if the external torque is equal to 0 the angular momentum will be conserved. Note

CHAPTER 2. THEORY

that the angular momentum do not need to be conserved in all directions, but can be conserved in a specific direction, the direction where the external torque is 0 [7].

2.6 Chaos

Chaos theory of dynamical systems involves the study of the behavior and sensitivity to initial conditions of mechanical systems. Even though the system itself might seem simple and have a deterministic solution, meaning that the initial conditions determine the future behavior of the system, it might become chaotic after a certain period of time. This often yields a wide range of outcomes, making the prediction of the dynamics of the system near to impossible. The behavior of the system seems to become stochastic. Just because a system have a deterministic nature do not make it predictable for all future time.

This claim might seem to contradict the deterministic nature of the system, but it isn't the case. Often the amount of information one have about the system at the start is finite, for example the accuracies of the initial conditions, which implies that the prediction of the dynamics will be weaker farther in to the future. This yields a stochastic outcome rather than a deterministic although the nature of the solutions of the dynamics of the system themselves might be deterministic [8].

There are mainly three ways of studying the chaotic or periodic behavior of a dynamical system, phase-plots, Poincare-maps and sensitivity to initial conditions. A phase-plot is a plot of one generalized coordinate against the corresponding generalized velocity. All generalized coordinates and their respective velocities make up the phase-space that describes periodic and stable qualities of the system, which can be analysed in the 2 dimensional phase-plots. Here a closed curve indicates a periodic behavior of the system, where as if the phase-plane is filled and we get a non-closed curve we probably have an chaotic, non-periodic system.

In this thesis the phase-plots will be studied. However, to further analyze and study the chaotic and periodic behavior of the dynamics of systems Poincare-maps provides a powerful tool. Nevertheless, this is not studied in this thesis. Furthermore, the study of the sensitivity to initial conditions do also give an indication if the system is chaotic or not. If the initial conditions are subjected to small disturbances that results in great deviations in the dynamics of the system, one can say that the system is of a chaotic nature.

There exists no universally excepted definition of when a classical mechanical system is chaotic today, but most scientists agree with the stated methods of classifying systems as chaotic. Consequently we look at phase-plots and sensitivity to initial conditions in the analysis of the solutions of the problems in this work [8].

2.7 Mechanics and geometry

In this section the basic theory behind rotation of coordinate systems and kinetic energy of particles and rigid bodies is presented in short.

2.7.1 Coordinate systems and rotation

Rotations are a central part when dealing with the descriptions of physical systems. Here, we briefly present some useful tools for working with rotations.

The following rotation-matrices rotates a vector $[x, y, z]^T$ around the $[1, 0, 0]^T$, $[0, 1, 0]^T$ and $[0, 0, 1]^T$ direction respectively:

$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}, \quad (2.65)$$

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, \quad (2.66)$$

$$\mathbf{R}_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.67)$$

By successive multiplication of $\mathbf{R}_x(\theta), \mathbf{R}_y(\theta)$ and $\mathbf{R}_z(\theta)$ we can form rotations relative to coordinate systems other than the fixed one. This technique is convenient when describing some geometric parts of a system in relation to other parts. We use of this technique when formulating the geometry of our problems, see **sections 3.3.1 and 3.4.1**. We proceed by an example:

Let $\boxed{\mathcal{N}}$ be the fixed Cartesian system. We describe vectors in this system by

$$\boxed{\mathcal{N}} : \begin{bmatrix} x_{\mathcal{N}} \\ y_{\mathcal{N}} \\ z_{\mathcal{N}} \end{bmatrix} \quad (2.68)$$

Let $\boxed{\mathcal{A}}$ be a system that is rotated an angle θ (clockwise) about the x -axis from $\boxed{\mathcal{N}}$. We denote vectors written in the $\boxed{\mathcal{A}}$ system by $[x_{\mathcal{A}}, y_{\mathcal{A}}, z_{\mathcal{A}}]^T$. The linear transformation that takes us from $\boxed{\mathcal{A}}$ to the fixed coordinate system is

$$\boxed{\mathcal{A} \rightarrow \mathcal{N}} : \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}}_{\mathbf{R}_x(\theta)} \begin{bmatrix} x_{\mathcal{A}} \\ y_{\mathcal{A}} \\ z_{\mathcal{A}} \end{bmatrix} \quad (2.69)$$

Let $\boxed{\mathcal{B}}$ be a coordinate system that is rotated an angle φ (clockwise) about the y -axis of $\boxed{\mathcal{A}}$. We denote vectors written in the $\boxed{\mathcal{B}}$ system by $[x_{\mathcal{B}}, y_{\mathcal{B}}, z_{\mathcal{B}}]^T$. The

CHAPTER 2. THEORY

transformation that takes us from $\boxed{\mathcal{B}}$ to the fixed coordinate system is

$$\boxed{\mathcal{B} \rightarrow \mathcal{N}} : \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}}_{R_x(\theta)} \underbrace{\begin{bmatrix} \cos(\varphi) & 0 & -\sin(\varphi) \\ 0 & 1 & 0 \\ \sin(\varphi) & 0 & \cos(\varphi) \end{bmatrix}}_{R_y(\varphi)} \begin{bmatrix} x_{\mathcal{B}} \\ y_{\mathcal{B}} \\ z_{\mathcal{B}} \end{bmatrix} \quad (2.70)$$

In Sophia, the systems $\boxed{\mathcal{A}}$ and $\boxed{\mathcal{B}}$ can be written as

```
&rot([N, A, 1, theta]);
&rot([A, B, 2, phi]);
```

As we see, the matrix-notation is abstracted away for simplicity. It is useful, however, to explicitly show what happens behind the scenes when drawing the geometry of the system in other software with different notion.

Note also that we always provide a description of going from any system to the fixed system $\boxed{\mathcal{N}}$ (even though, for example, system $\boxed{\mathcal{B}}$ is defined in terms of $\boxed{\mathcal{A}}$ and $\boxed{\mathcal{N}}$ is not mentioned explicitly). This is because we always measure the absolute motion relative the fixed coordinate system. For example, the kinetic energy of a particle obviously depends on the square of its velocity vector and this vector must ultimately be expressed in the $\boxed{\mathcal{N}}$ system.

2.7.2 Kinetic energy and moment of inertia

When calculating the kinetic energy for a system we have to take in consideration all different parts of the system which contributes to the kinetic energy. These are mainly the components that have mass.

For a single particle we have that the kinetic energy can be written as

$$T_{particle} = \frac{1}{2}m(\mathbf{v} \cdot \mathbf{v}) \quad (2.71)$$

where m is the mass and \mathbf{v} is the velocity of the particle. For a rigid body in 3 dimensions we have that the kinetic energy for a arbitrary motion is given by

$$T_{rigidbody} = \frac{1}{2}m(\mathbf{v}_{\mathcal{G}} \cdot \mathbf{v}_{\mathcal{G}}) + \frac{1}{2}\boldsymbol{\omega}^T \mathbf{I}_{\mathcal{G}} \boldsymbol{\omega} \quad (2.72)$$

where \mathcal{G} is the center of mass, $\mathbf{I}_{\mathcal{G}}$ is the moment of inertia with respect to \mathcal{G} and $\boldsymbol{\omega}$ is the angular velocity of the body. As may be seen, the rigid body has two contribution to its kinetic energy. One from the translation of the body and one from its rotation around its center of mass.

What also is of interest for this thesis is how the moment of inertia can be calculated in special cases. For a rigid body we have that the components of the moment of inertia in 2 dimensions are

$$I_x = \int (y^2 + z^2) dm \quad (2.73)$$

CHAPTER 2. THEORY

$$I_y = \int (x^2 + z^2) dm \quad (2.74)$$

$$I_z = \int (x^2 + y^2) dm \quad (2.75)$$

If we also want to study the 3 dimensional case we have to use matrix notation for the moment of inertia where we also get components $I_{xy}, I_{xz}, I_{yz}, I_{yx}, I_{zx}$ and I_{zy} . Then a moment of inertia tensor can be formed as

$$I = \begin{pmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{pmatrix} \quad (2.76)$$

In the case of a solid torus with uniform mass distribution, all these new components become equal to 0 and we are left with the moment of inertia tensor for a torus as

$$I = \begin{pmatrix} (\frac{3}{4}r^2 + R^2)M & 0 & 0 \\ 0 & (\frac{5}{8}r^2 + \frac{1}{2}R^2)M & 0 \\ 0 & 0 & (\frac{5}{8}r^2 + \frac{1}{2}R^2)M \end{pmatrix} \quad (2.77)$$

where M is the total mass of the torus, r its inner radius and R its outer radius [9].

To get the moment of inertia tensor for a hollow torus one simply subtract the tensor corresponding to the hollow part of the hollow torus, with the same density of mass as the solid torus, from the tensor describing the solid torus. Furthermore if the thickness of the hollow torus is set to t, the tensor for the moment of inertia for the corresponding hollow part is the same as for the solid torus given that $r = r - t$ and $R = R + t$.

Chapter 3

Method

In this section a brief explanation of all tools involved is given and how the solutions of the two mechanical problems where carried out using these tools.

3.1 Software

The tools used in solving the two mechanical problems are briefly presented below.

3.1.1 Maple and Sophia

Maple is a computer software for symbolical solving of mathematical as well as physical problems. Although Maple is one of the largest platforms for modern computer algebra it lacks specific tools in solving advanced mechanical problems.

However, together with the Sofia package which is a specific software for treating problems in rigid body mechanics the problems in this work could be treated. Sofia consists of a set of procedures that make the process of solving a wide rang of rigid body problems more efficient and simpler. In this thesis the Maple version of Sofia is used and the specific use of it can be seen in the code in **appendix A**.¹

3.1.2 Blender

Blender is a free and open source software for 3D creation [10]. It can be used to model physical problems, render animations, video editing and creating graphics for games. However, in this thesis Blender is mainly used to render animations of the data produced in Sofia. The 3D environment in each problem including the problems themselves are created through a Python script included in **appendix B** and run in blender where the details of the quality of the rendering of each picture and some properties of the camera are set.

¹The Sophia plug-in can be found at: http://www.mech.kth.se/~nap/F_fk/sophia/

3.2 General procedure

The general procedure for modeling and solving the two mechanical problems can be summarized in the following 4 steps:

1. Define the geometry using generalized coordinates and constants.
 2. Calculate the kinetic energy T .
 3. Depending on the problem:
 - 3.1. *Without Friction (only conservative forces)*
 - 3.1.1. Calculate the potential energy V .
 - 3.1.2. Construct the Lagrangian $\mathcal{L} = T - V$.
 - 3.1.3. Set up the special case of the Euler-Lagrange equations.
-

- 3.1. *With Friction (or non-conservative forces)*
 - 3.1.1. Find explicit expression for all active forces \mathbf{F}_j and the points at which they are applied \mathbf{r}_j .
 - 3.1.2. Compute the sum $\sum_j \mathbf{F}_j \cdot \frac{\partial \mathbf{r}_j}{\partial q_i}$ for all $i = 1, 2, \dots$
 - 3.1.3. Set up the general case of the Euler-Lagrange equations.

4. Numerically integrate the Euler-Lagrange equations with specified initial conditions.

The sections below give more detailed explanations of these steps.

3.3 *Boston Hoop*

3.3.1 Geometry

The Boston Hoop is a three dimensional system with three degrees of freedom. We will use the generalized coordinates $q_1(t)$, $q_2(t)$ and $q_3(t)$ to describe it. The main points of interest are the location of the two balls, \mathcal{B}_1 and \mathcal{B}_2 . Here we present the geometry for the problem:

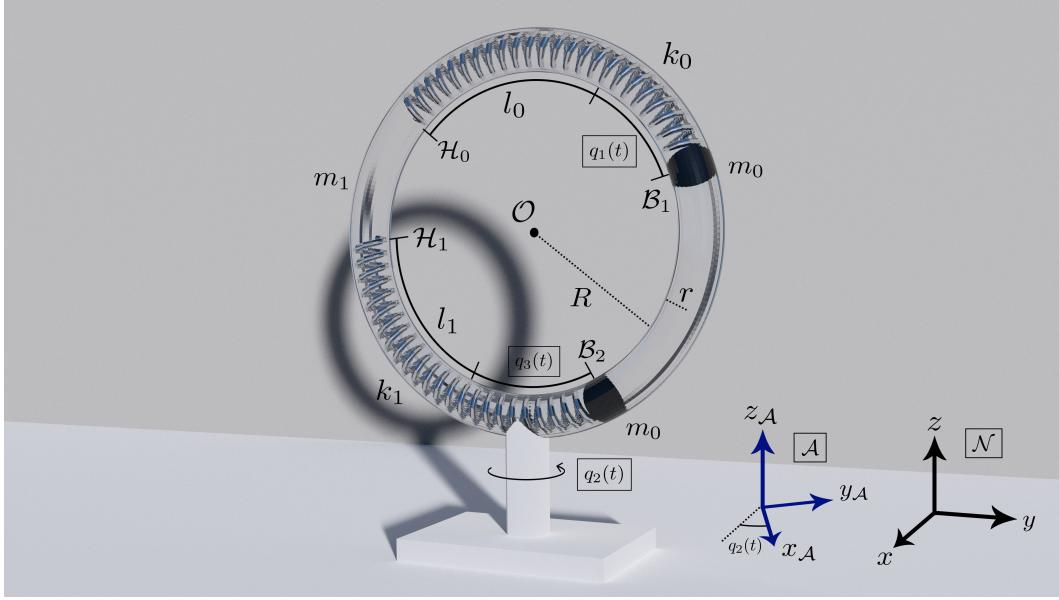


Figure 3.1. A three dimensional render of the Boston Hoop

Because of the nature of the problem, we use polar coordinates to describe position of the balls inside the torus. The torus itself is allowed to rotate around the z -axis and so we need to multiply the positions vectors of the balls with $\mathbf{R}_z(q_2(t))$.

The position of ball B_1 is uniquely defined by the generalized coordinate $q_1(t)$. It represents the deviation in angle from the location of the ball when the spring is unstretched. B_1 can be written as

$$\mathbf{r}_{B_1} = \mathbf{R}_z(q_2(t)) \begin{bmatrix} R \cdot \cos\left(\mathcal{H}_0 - \frac{l_0}{R} - q_1(t)\right) \\ 0 \\ R \cdot \sin\left(\mathcal{H}_0 - \frac{l_0}{R} - q_1(t)\right) \end{bmatrix} \quad (3.1)$$

The position of ball B_2 is almost the same as in the previous case. Analogously, we have $q_3(t)$ in place of $q_1(t)$ and some other differences such as constants and signs. B_2 may be written as

$$\mathbf{r}_{B_2} = \mathbf{R}_z(q_2(t)) \begin{bmatrix} R \cdot \cos\left(\mathcal{H}_1 + \frac{l_1}{R} + q_3(t)\right) \\ 0 \\ R \cdot \sin\left(\mathcal{H}_1 + \frac{l_1}{R} + q_3(t)\right) \end{bmatrix} \quad (3.2)$$

3.3.2 Mechanics

In this section the basic mechanics of the *Boston Hoop* problem are treated. The moment of inertia, kinetic energy and potential energy are all computed using theory stated in the previous chapter.

3.3.2.1 Moment of inertia

From the geometry of this problem we see that the torus itself rotates only around its diametrical axis. If we now approximate the torus as a solid torus with uniformly distributed mass we get the moment of inertia as the tensor described in **section 2.7.2** as

$$\mathbf{I} = \begin{pmatrix} (\frac{3}{4}r^2 + R^2)M & 0 & 0 \\ 0 & (\frac{5}{8}r^2 + \frac{1}{2}R^2)M & 0 \\ 0 & 0 & (\frac{5}{8}r^2 + \frac{1}{2}R^2)M \end{pmatrix} \quad (3.3)$$

The approximation to a solid torus in this case is made because no thickness t is given in the problem description and as an effort in having the complexity level of the problem at a reasonable level. The difference when calculating with a hollow torus turns out not to be of importance for the results of the problem.

Because the torus only rotates around its diametrical axis we see from the the tensor that the moment of inertia will be

$$I_{hoop} = \left(\frac{5}{8}r^2 + \frac{1}{2}R^2 \right) M \quad (3.4)$$

where M is the total mass of the torus, r is the inner radius and R is the outer radius of the torus. The two other masses in the problem are modeled as point masses and will not have a moment of inertia.

3.3.2.2 Kinetic energy

The kinetic energy of the *Boston Hoop* problem can now be calculated as the sum of the kinetic energy of the hoop and the two point masses or balls within the hoop as

$$T = T_{hoop} + T_{balls} \quad (3.5)$$

The kinetic energy of the hoop can be calculated using the formula for kinetic energy for a rigid body stated in equation (2.72). In this case we have that $\mathbf{v}_G = \mathbf{0}$ i.e. the hoop itself do not move in space it only rotates around its center of mass. The angular velocity is, with the choice of generalized coordinates made in the geometry section, $\boldsymbol{\omega} = \dot{q}_2$ where \dot{q}_2 is the time derivative of the generalized coordinate q_2 . This gives us the kinetic energy for the hoop as

$$T_{hoop} = \frac{1}{2}\boldsymbol{\omega}^T \mathbf{I}_G \boldsymbol{\omega} = \frac{1}{2} \cdot \left(\frac{5}{8}r^2 + \frac{1}{2}R^2 \right) m_1 \cdot \dot{q}_2^2 \quad (3.6)$$

The kinetic energy for the two balls can now be calculated as

$$T_{balls} = T_{B_1} + T_{B_2} = \frac{1}{2}m_{B_1}(\mathbf{v}_{B_1} \cdot \mathbf{v}_{B_1}) + \frac{1}{2}m_{B_2}(\mathbf{v}_{B_2} \cdot \mathbf{v}_{B_2}) \quad (3.7)$$

where in this case $m_{B_1} = m_{B_2} = m_0$ and $\mathbf{v}_{B_1}, \mathbf{v}_{B_2}$ are calculated with respect to the Newtonian fixed coordinate system $\boxed{\mathcal{N}}$ using Sofia commands, see code in

appendix A.1. Here \mathcal{B}_1 and \mathcal{B}_2 stands for ball 1 and 2. This give us the total kinetic energy for the *Boston Hoop* as

$$T = \frac{1}{16} \cdot (5r^2 + 4R^2) m_1 \cdot \dot{q}_2^2 + \frac{1}{2} m_0 \cdot v_{\mathcal{B}_1}^2 + \frac{1}{2} m_0 \cdot v_{\mathcal{B}_2}^2 \quad (3.8)$$

3.3.2.3 Potential energy

The potential energy of the this system is given by the sum of the potential energy of the height of the balls and the deviation from the natural length of the springs for each ball. Therefore, we can write the potential energy as

$$V = V_{\mathcal{B}_1} + V_{\mathcal{B}_2} + V_{\mathcal{S}_1} + V_{\mathcal{S}_2} \quad (3.9)$$

where \mathcal{B}_1 and \mathcal{B}_2 stands for the potential energy due to the height of the balls, and \mathcal{S}_1 , \mathcal{S}_2 are their corresponding potential energy due to the springs they are attached to. These are calculated in Sofia using the position vectors of the balls and special Sofia commands, see code in **appendix A.1** for details. But the general idea behind the calculations are the following. If h_1 is the height of ball 1 and h_2 is the height of ball 2 and their corresponding springs have spring constants k_1 and k_2 , the four contributions to the potential energy can be calculated as

$$V_{\mathcal{B}_1} = m_0 \cdot g \cdot h_1 \quad (3.10)$$

$$V_{\mathcal{B}_2} = m_0 \cdot g \cdot h_2 \quad (3.11)$$

$$V_{\mathcal{S}_1} = \frac{1}{2} \cdot k_0 \cdot (R \cdot q_1(t))^2 \quad (3.12)$$

$$V_{\mathcal{S}_2} = \frac{1}{2} \cdot k_1 \cdot (R \cdot q_3(t))^2 \quad (3.13)$$

where we used the geometric setup described previously for ball 1 and 2. Note that the potential for the springs are circular in this case and because of this the factor describing the deviation in length from the natural length of the springs become R times the generalized coordinate describing the ball in question. This gives us the length of interest because of the way the coordinates were chosen. For clarification of constants and coordinates see **sections 1.3.1** and **3.3.1**.

3.3.2.4 Angular momentum

The angular moment of a system of particles and rigid bodies can be calculated as the sum of the angular moment of the individual parts. In this system we have two balls and one torus that contributes to the angular momentum, thus we need to calculate the angular momentum separately for the two balls and the torus.

CHAPTER 3. METHOD

The angular momentum of a particle, in this case the balls, can be calculated as the cross product of the positions vector of the particle and the momentum of the particle. The positions vectors of the two balls in the system are already defined in the geometry of the system and the momentum of the balls was calculated using Sofia commands. In this case the commands used where those that get the velocity of the balls in respect to the fix coordinate system \mathcal{N} . If we denote the position vectors to the ball one and two as $\mathbf{r}_{\mathcal{B}_1}, \mathbf{r}_{\mathcal{B}_2}$ with their respective velocities $\mathbf{v}_{\mathcal{B}_1}, \mathbf{v}_{\mathcal{B}_2}$ we can write the angular momentum \mathbf{H} for the two balls as

$$\mathbf{H}_{\mathcal{B}_1} = \mathbf{r}_{\mathcal{B}_1} \times m\mathbf{v}_{\mathcal{B}_1} \quad (3.14)$$

$$\mathbf{H}_{\mathcal{B}_2} = \mathbf{r}_{\mathcal{B}_2} \times m\mathbf{v}_{\mathcal{B}_2} \quad (3.15)$$

The angular momentum of the hoop was calculated as the product between the moment of inertia and the angular velocity vector. The angular velocity was constructed from the geometry of the system as

$$\boldsymbol{\omega} = (0, 0, \dot{q}_2) \quad (3.16)$$

where \dot{q}_2 is the time derivative of the generalized coordinate q_2 . The moment of inertia I_{hoop} was calculated according to the procedure in the previous section, and we get the angular momentum for the hoop as

$$\mathbf{H}_{hoop} = I_{hoop}\boldsymbol{\omega} \quad (3.17)$$

The total angular momentum can now be expressed as

$$\mathbf{H} = \mathbf{r}_{\mathcal{B}_1} \times m\mathbf{v}_{\mathcal{B}_1} + \mathbf{r}_{\mathcal{B}_2} \times m\mathbf{v}_{\mathcal{B}_2} + I_{hoop}\boldsymbol{\omega} \quad (3.18)$$

Furthermore, if the equation describing the connection between the angular momentum and the torque is used we see that in this case we only get components of the torque in the plane orthogonal to the rotation of the torus. This due to the gravitational forces in this system always being in the same plane as the hoop. This means that the angular momentum only is conserved in the z-direction in the fixed system \mathcal{N} . Because of this we only look at the z-component of the angular momentum when analyzing it.

3.3.3 Cases

To effectively analyse the solution of the system, we divided the analysis into three different cases. The first case we looked at was the plots of the generalized coordinates over time for certain values of the constants and initial conditions of the system. These were kept constant through most of the steps and cases of the analysis. Phase portraits of the generalized coordinates where also made and a shallow analysis of the dynamic behavior of the system was carried out.

CHAPTER 3. METHOD

The second case was to look at the conserved quantities of the system. These included the conservation of energy and angular momentum. Plots of the two were made and a conclusion considering the conservation of them was stated.

The third and last case was to analyse how the energy and angular momentum in the system moved over time between different parts of the system. Plots over these quantities for the specific parts of the system where made and a brief analysis of them. Furthermore a short discussion about the results of the different cases was carried out and the importance of the Lagrangian method together with computer algebra discussed.

3.4 ***Double pendulum with dual springs***

In this section the geometry and basic mechanics of the *Double pendulum with dual springs* problem are presented. With these calculated and chosen the procedure of solving the problem is done according to the procedure described in the previous section 3.2.

3.4.1 **Geometry**

The pendulum is a two dimensional system with four degrees of freedom². We will use the generalized coordinates $q_1(t)$, $q_2(t)$, $q_3(t)$ and $q_4(t)$ to describe it. The system has 4 main points of physical interest: \mathcal{A} , \mathcal{B} , \mathcal{C} and \mathcal{D} as shown in the figure below. We now present the mathematical description of the points³:

²Even though this is a 2D problem, we use 3D notation through out this thesis for consistency.

³We present the description relative the fixed Cartesian system $[\mathcal{N}]$.

CHAPTER 3. METHOD

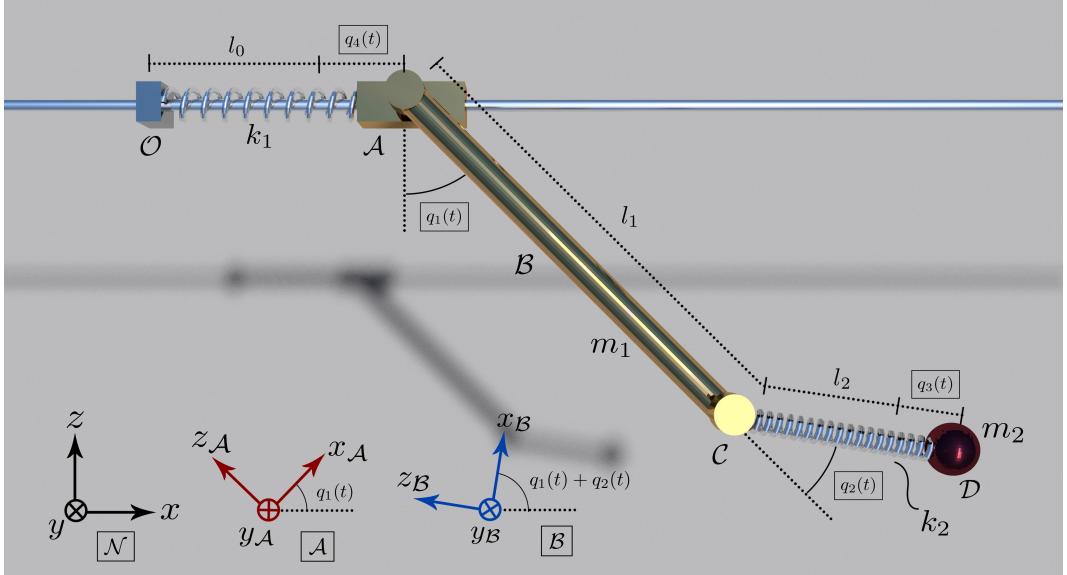


Figure 3.2. A three dimensional render of the double pendulum

The point \mathcal{O} is the origin

$$\mathbf{r}_{\mathcal{O}} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.19)$$

The point \mathcal{A} is attached to the upper spring and its position depends on the stretching of the spring. $q_4(t)$ is the deviated length from its natural length. \mathcal{A} can be described as

$$\mathbf{r}_{\mathcal{A}} = \begin{bmatrix} l_0 + q_4(t) \\ 0 \\ 0 \end{bmatrix} \quad (3.20)$$

The point \mathcal{B} indicates the center mass of the rod and is located along half its length. The rod itself is rotated an angle $q_1(t)$ about the y -axis and \mathcal{B} can therefore be written as

$$\mathbf{r}_{\mathcal{B}} = \begin{bmatrix} l_0 + q_4(t) \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_y(q_1(t)) \begin{bmatrix} 0 \\ 0 \\ -\frac{l_1}{2} \end{bmatrix} \quad (3.21)$$

where we have used the rotation notation presented in **section 2.7.1**. Point \mathcal{C} is nearly identical to \mathcal{B} but is located along at the bottom of the rod instead of in the

CHAPTER 3. METHOD

middle, as such

$$\mathbf{r}_C = \begin{bmatrix} l_0 + q_4(t) \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_y(q_1(t)) \begin{bmatrix} 0 \\ 0 \\ -l_1 \end{bmatrix} \quad (3.22)$$

Point \mathcal{D} describes the location of mass m_2 . It is further rotated an angle $q_2(t)$ about the same axis and stretched an amount $q_3(t)$ along the length of the spring so that

$$\mathbf{r}_D = \begin{bmatrix} l_0 + q_4(t) \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_y(q_1(t)) \begin{bmatrix} 0 \\ 0 \\ -l_1 \end{bmatrix} + \mathbf{R}_y(q_1(t)) \mathbf{R}_y(q_2(t)) \begin{bmatrix} 0 \\ 0 \\ -(l_2 + q_3(t)) \end{bmatrix} \quad (3.23)$$

With the position vector to each of these four points defined, we can now set up the necessary mechanical properties to model the system.

3.4.2 Mechanics

In this section the basic mechanics of the system in question are presented.

3.4.2.1 Moment of inertia

The moment of inertia of interest in this system is that of the rigid rod. In this case we are interested of the moment of inertia for a 2 dimensional problem where the rod in question is rotating about its center of mass. To calculate this we use the theory presented in [section 2.7.2](#) and the fact that the mass of the rod is uniformly distributed over its length and that the rod itself is infinitely thin. If the total mass is m_1 and the length of the rod is l_1 , the density of the rod per length becomes $\frac{m_1}{l_1}$. Now the moment of inertia can be calculated as

$$I = \int_{-l_1/2}^{l_1/2} x^2 \cdot \frac{m_1}{l_1} dx = \frac{1}{12} m_1 \cdot l_1^2 \quad (3.24)$$

3.4.2.2 Kinetic energy

The kinetic energy for the system in question can be calculated by considering the sum of the kinetic energy of the ball and the rigid rod. In our solution we used the Sofia commands to calculate the velocity of the ball and the rigid rod from their respective position vectors for their center of mass. If we denote the velocity for the ball as \mathbf{v}_B and the velocity for the centre of mass of the rigid rod as \mathbf{v}_{rg} we get the kinetic energy for the system as

$$T = T_{ball} + T_{rod} = \frac{1}{2} m_1 \cdot (\mathbf{v}_B \cdot \mathbf{v}_B) + \frac{1}{2} m_2 \cdot (\mathbf{v}_{rg} \cdot \mathbf{v}_{rg}) + \frac{1}{2} I_{rod} \omega^2 \quad (3.25)$$

CHAPTER 3. METHOD

where I_{rod} is the moment of inertia of the rigid rod and ω is the angular velocity of the rigid rod around its center of mass. The moment of inertia for the rod was calculated in the previous section and the angular velocity can be determined from the geometry of the system as the time derivative of the generalized coordinate q_1 i.e. $\omega = \dot{q}_1$.

3.4.2.3 Potential energy

The potential energy due to the gravitational field the system is in can be calculated as the sum of the potential energy for the ball and the rigid rod. These calculations were done by using Sofia commands to get the height of the ball and the center of mass of the rod. In the system we also have two springs (labeled S_1 and S_2) that contribute to the potential energy. These are also necessary to summarize over when calculating the total potential energy of the system. Thus the total potential energy can be calculated as

$$V = V_{ball} + V_{rod} + V_{S_1} + V_{S_2} \quad (3.26)$$

To calculate the first two terms of the total potential energy above the height of the two components were calculated be taking the scalar product of the position vectors of the components with the unit vector in the z-direction in the fixed system \mathcal{N} . If we now denote h_{rod} as the height of the center of mass of the rigid rod and h_{ball} as the height of the ball, we get the first two terms of the total potential energy as

$$V_{ball} + V_{rod} = \frac{1}{2}m_1 \cdot g \cdot h_{ball} + \frac{1}{2}m_2 \cdot g \cdot h_{rod} \quad (3.27)$$

Then the potential energy for the two springs could be calculated as following, from the way the geometry of the system was set up

$$V_{S_1} + V_{S_2} = \frac{1}{2} \cdot k_1 \cdot q_4(t)^2 + \frac{1}{2} \cdot k_2 \cdot q_3(t)^2 \quad (3.28)$$

where k_1 and k_2 are the spring constants for the two springs. The total potential energy for the system can now be written as

$$V = \frac{1}{2} \left(m_1 \cdot g \cdot h_{ball} + m_2 \cdot g \cdot h_{rod} + k_1 \cdot q_4(t)^2 + k_2 \cdot q_3(t)^2 \right) \quad (3.29)$$

3.4.2.4 Generalised forces and friction

In this section we look at a case where there is friction present at the two rotating joints \mathcal{A} and \mathcal{C} . We model these frictionous forces to be proportional to the respective angular velocities \dot{q}_1 and \dot{q}_2 .

When dealing with friction, we must use the general form of the Euler-Lagrange equations as stated in section (2.4). This gives us two choices as to which formulation to use: (2.16) or (2.36). Here, we chose not to build a Lagrangian function but to instead only consider the kinetic energy T and explicitly state all active forces on the system using (2.16).⁴ Figure 3.3 illustrates the situation.⁵

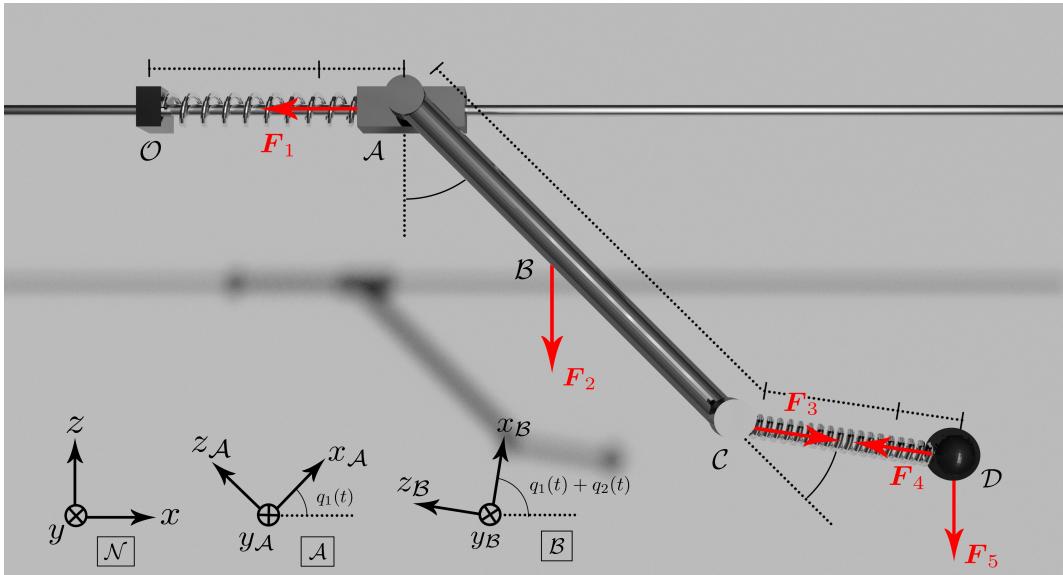


Figure 3.3. Illustration of the active conservative forces on system.

To model the system, we want to compute the right hand side of (2.16):

$$\sum_{j=1}^5 \mathbf{F}_j \cdot \frac{\partial \mathbf{r}_j}{\partial q_i}, \quad i = 1, 2, 3, 4 \quad (3.30)$$

In this expression, \mathbf{F}_j is the j :th force and is applied at the point \mathbf{r}_j . What follows are the explicit expression for all active conservative forces:

⁴Equation (2.36) is admittedly simpler because it discards all conservative forces. However, we chose the more inconvenient way to show the procedure of setting up all the forces.

⁵Note that we do not display *all* forces on the system but only all *active* forces. For example, the vertical normal force at \mathcal{A} is not displayed and do not enter into the calculations. This is because it is perpendicular to the purely horizontal movement at \mathcal{A} - formally, this means that the dot-product $\mathbf{F}_{normal} \cdot \frac{\partial \mathbf{r}_A}{\partial q_i}$ is zero and it does not contribute to the movement of the system. For this reason, forces like these are considered to be inactive. (It can be thought of as being included in the geometry of the set-up).

CHAPTER 3. METHOD

\mathbf{F}_1 is the force acting on the upper end of the rigid rod (\mathcal{A}) due to the spring. As usual, the strength of a spring-force is modeled to be proportional to its stretching:

$$\mathbf{F}_1 = \begin{bmatrix} -k_1 \cdot q_4(t) \\ 0 \\ 0 \end{bmatrix} \quad (3.31)$$

\mathbf{F}_2 the gravitational force on the rod, attacking at its center of mass (\mathcal{B}):

$$\mathbf{F}_2 = \begin{bmatrix} 0 \\ 0 \\ -m_1 g \end{bmatrix} \quad (3.32)$$

\mathbf{F}_3 is the force acting on the lower end of the rigid rod (\mathcal{C}) due to the spring:⁶

$$\mathbf{F}_3 = \mathbf{R}_y(q_1(t)) \mathbf{R}_y(q_2(t)) \begin{bmatrix} 0 \\ 0 \\ -k_2 \cdot q_3(t) \end{bmatrix} \quad (3.33)$$

\mathbf{F}_4 is the reaction force to \mathbf{F}_3 and is acting on the ball at (\mathcal{D}). It is equal in magnitude to \mathbf{F}_3 but oppositely directed:

$$\mathbf{F}_4 = \mathbf{R}_y(q_1(t)) \mathbf{R}_y(q_2(t)) \begin{bmatrix} 0 \\ 0 \\ k_2 \cdot q_3(t) \end{bmatrix} \quad (3.34)$$

Finally, \mathbf{F}_5 is the force on the ball due to gravity and is also applied at (\mathcal{D}):

$$\mathbf{F}_5 = \begin{bmatrix} 0 \\ 0 \\ -m_2 g \end{bmatrix} \quad (3.35)$$

Up to this point, we have only included the conservative forces for the system. In this case, the special- and general formulation of the Euler-Lagrange equations are mathematically equivalent. However, as is discussed further down in Chapter 4, there will still be some numerical differences between the two formulations.

We proceed by adding friction. The aim is to find the expressions for two forces counter-acting the rotation in the two joints \mathcal{A} and \mathcal{C} . We adopt the usual convention of modeling air-resistance on a body by claiming it can be represented as a force proportional to the velocity of the body. In our case, we make these two forces proportional to the respective angular velocities \dot{q}_1 and \dot{q}_2 . We make them perpendicular to the *rotation-diameter* and make them rotate with the system by multiplying with the respective rotation matrices. This way of setting up the forces

⁶Note the use of rotation matrices when we describe the forces in the fixed cartesian system \mathcal{N} . In Sophia, we simply define this force as $[0, 0, -k_1 \cdot q_3(t)]^T$ where the vector is expressed in the \mathcal{B} -system that is rotated relative to the fixed system. The same goes for \mathbf{F}_4 .

CHAPTER 3. METHOD

is of course equivalent to applying the corresponding torque to the system. We define the forces as:

$$\mathbf{F}_{\dot{q}_1} = \mathbf{R}_y(q_1(t)) \begin{bmatrix} -c_1 \dot{q}_1(t) \\ 0 \\ 0 \end{bmatrix}, \quad \text{applied at } \mathcal{C} \quad (3.36)$$

$$\mathbf{F}_{\dot{q}_2} = \mathbf{R}_y(q_1(t)) \mathbf{R}_y(q_2(t)) \begin{bmatrix} -c_2 \dot{q}_2(t) \\ 0 \\ 0 \end{bmatrix}, \quad \text{applied at } \mathcal{D} \quad (3.37)$$

Computing the dot-product for $\mathbf{F}_{\dot{q}_1}$ gives us

$$\mathbf{F}_{\dot{q}_1} \cdot \frac{\partial \mathbf{r}_{\mathcal{C}}}{\partial q_i} = -c_1 \dot{q}_1(t) \begin{bmatrix} \cos(q_1(t)) \\ 0 \\ \sin(q_1(t)) \end{bmatrix} \cdot \frac{\partial}{\partial q_i} \left(\begin{bmatrix} l_0 + q_4(t) \\ 0 \\ 0 \end{bmatrix} - l_1 \begin{bmatrix} -\sin(q_1(t)) \\ 0 \\ \cos(q_1(t)) \end{bmatrix} \right) \quad (3.38)$$

The partial derivative is non-zero only when $i = 1$. Looking at this case we get

$$\mathbf{F}_{\dot{q}_1} \cdot \frac{\partial \mathbf{r}_{\mathcal{C}}}{\partial q_1} = -c_1 \dot{q}_1(t) \begin{bmatrix} \cos(q_1(t)) \\ 0 \\ \sin(q_1(t)) \end{bmatrix} \cdot \begin{pmatrix} -l_1 \begin{bmatrix} -\cos(q_1(t)) \\ 0 \\ -\sin(q_1(t)) \end{bmatrix} \end{pmatrix} \quad (3.39)$$

$$= -c_1 l_1 \dot{q}_1(t) \cdot \left\{ \cos^2(q_1(t)) + \sin^2(q_1(t)) \right\} \quad (3.40)$$

$$= -C_1 \cdot \dot{q}_1(t) \quad (3.41)$$

where we have used the fact that $C_1 = c_1 l_1$ is an arbitrary constant and the trigonometric identity. Using the same argument, we can also show that

$$\mathbf{F}_{\dot{q}_2} \cdot \frac{\partial \mathbf{r}_{\mathcal{D}}}{\partial q_2} = -C_2 \cdot \dot{q}_2(t) \quad (3.42)$$

CHAPTER 3. METHOD

3.4.3 Cases

To systematically analyze the solution of the dynamics of the system we divide the analysis into three cases. In the first case we set the parameters of the system and plotted the generalized coordinates over time. Phase portraits for all generalized coordinates where made and a brief analysis of the dynamics of the system was carried out. A phase portrait of q_1 over a longer time period was also made to study the behavior over long time periods.

The second case consisted of analysing the sensitivity to initial conditions of the system at different levels of energy and initial conditions. Furthermore, the importance of numerical errors was discussed and how this can effect the correctness of the solution over time.

The last case was to introduce friction in the system and analyze the effects of the dynamics and energy. A comparison to the non-friction case was carried out and how the chaotic behavior of the system have been affected by the introduction of friction.

Chapter 4

Results

In this chapter the results of the solutions for the two systems and graphs over the properties of interest, according to what is stated in the cases section for each problem, are presented.

4.1 *Boston Hoop*

4.1.1 Generalized coordinates and phase plots

In **Figure 4.1** below, the solution for the generalized coordinates for the system over a time period of 36 seconds is presented. The initial conditions and parameter values used where

Constants:

$$R = 5, \ r = 0.6, \ \mathcal{H}_0 = \pi - 0.1, \ \mathcal{H}_1 = \pi + 0.1, \ l_0 = \frac{5\pi}{2}, \ l_1 = \frac{5\pi}{2}, \\ m_0 = 100, \ m_1 = 10, \ k_0 = 900, \ k_1 = 300, \ g = 9.82$$

Initial conditions:

$$q_1(0) = \frac{\pi}{2} - 0.3, \ q_2(0) = 0, \ q_3(0) = 0.1, \ u_1(0) = 0, \ u_2(0) = 1, \ u_3(0) = 0$$

CHAPTER 4. RESULTS

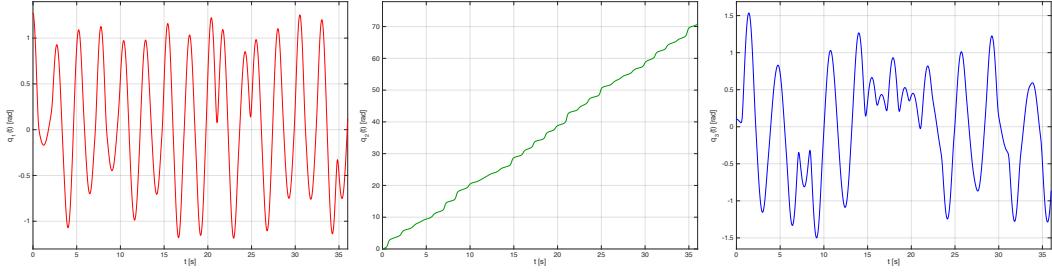


Figure 4.1. Solution for the generalized coordinates over 36 seconds. From left to right, q_1 , q_2 , q_3

In **Figure 4.1** we see the solution for the q_1 coordinate to the left and q_3 to the right. In the middle we see the solution for the generalized coordinate q_2 . None of the coordinates are seen to exhibit a strictly periodic behavior, although a quasi-periodic behavior can be suspected in the q_1 coordinate.

The phase portraits of the solution for the system with the same initial conditions and parameter values as above can be seen in **Figure 4.2** below.

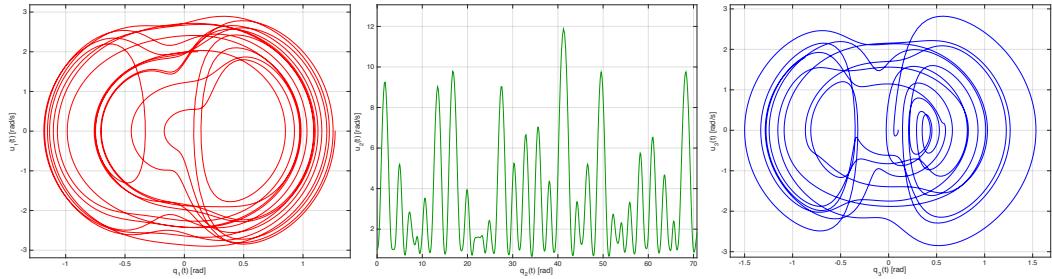


Figure 4.2. Phase portraits for the generalized coordinates and their corresponding velocities under a time period of 36 seconds. From left to right, q_1 , q_2 , q_3

Figure 4.3 shows the phase portraits for an extended time of 1000 seconds

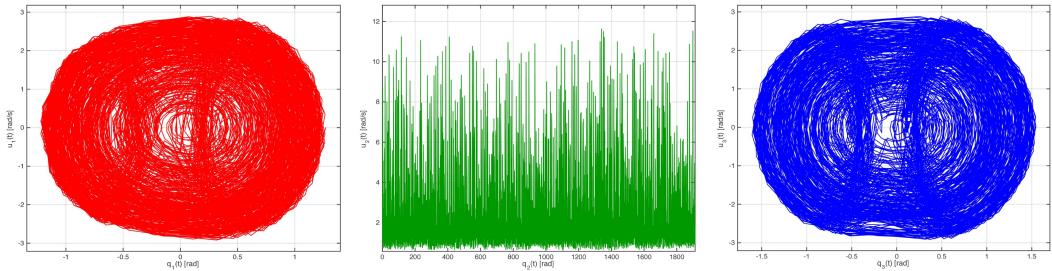


Figure 4.3. Phase portraits for the generalized coordinates and their corresponding velocities under 1000 seconds. From left to right, q_1 , q_2 , q_3

The phase portraits clearly show that they do not follow a certain pattern or

CHAPTER 4. RESULTS

create closed curves. With increasing time one can see that the whole phase-space becomes covered.

4.1.2 Conserved quantities

A study of the conserved quantities of the system where made which included the total energy and angular momentum in the z-direction , along the diametrical axis of the torus parallel to the gravitational force. The results of these two quantities are shown in **Figure 4.4** below.

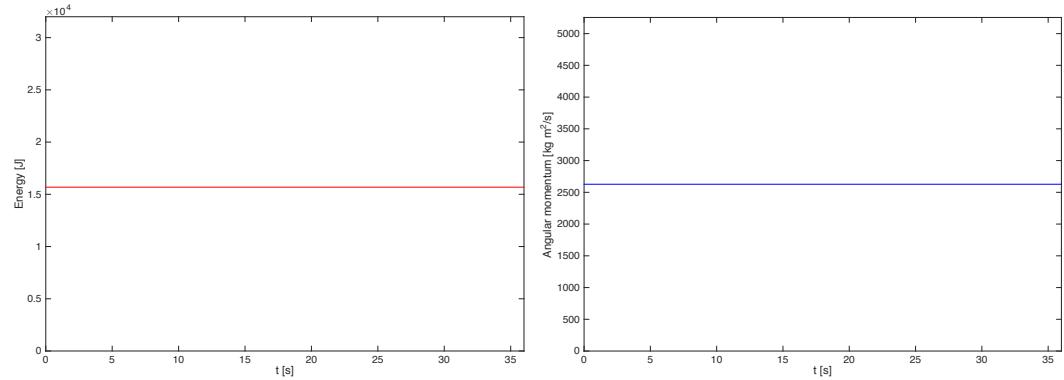


Figure 4.4. Total energy and angular momentum over 36 seconds

CHAPTER 4. RESULTS

4.1.3 Energy- and angular momentum transfer between different parts of the system

To analyse how the energy and angular momentum is distributed between the different parts of the system over time the three graphs in **Figure 4.5** below were made. The first graph shows the kinetic energy for the hoop and for the two balls.

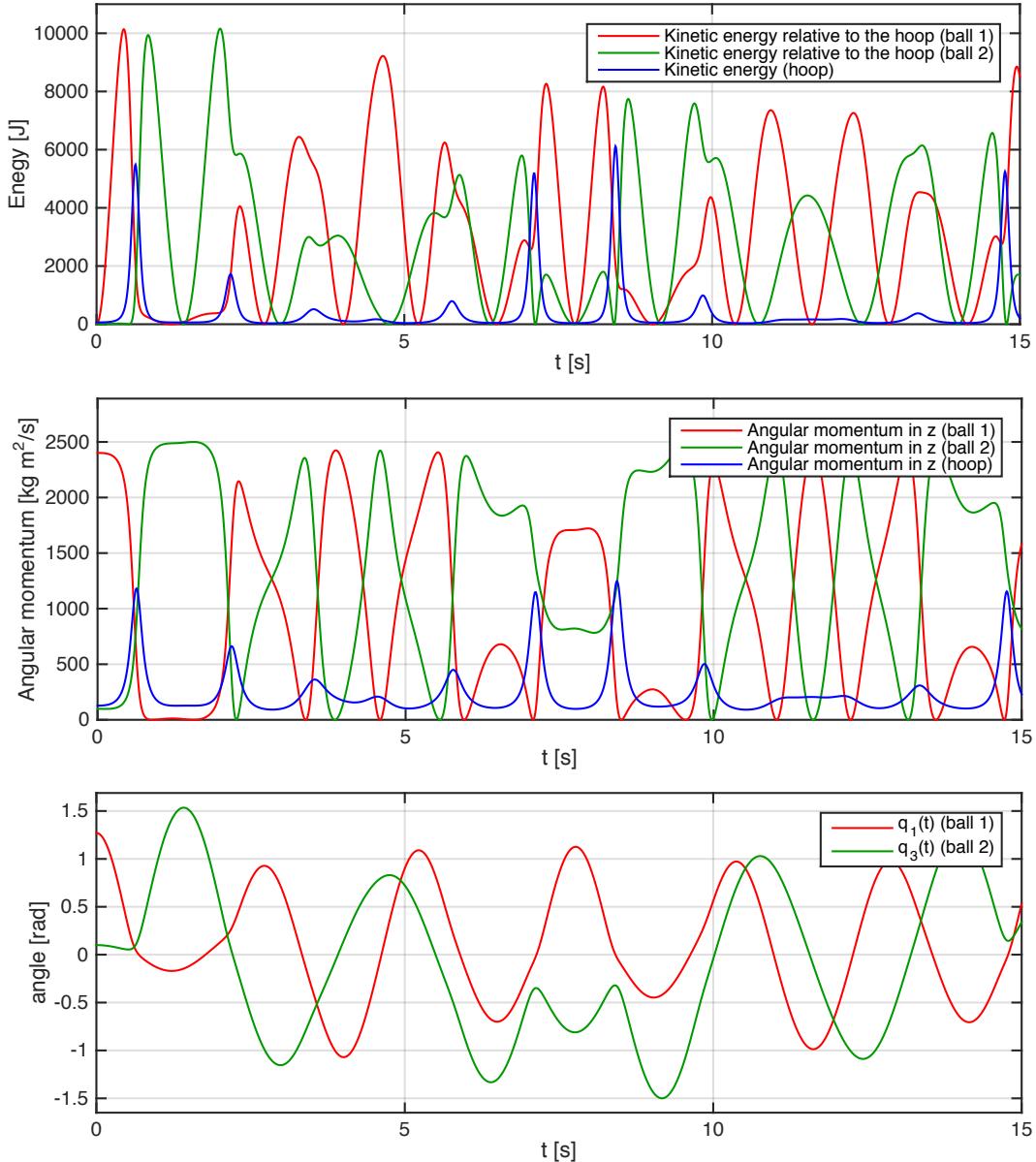


Figure 4.5. From top to bottom: Kinetic energy for the two balls and hoop, Angular momentum for the two balls and the hoop, and positions of the two balls relative the hoop. All three plots are taken over 15 seconds.

CHAPTER 4. RESULTS

However, it is important to notice that the kinetic energy for the balls in this graph is relative to the hoop. This gives an indication of the velocity of the balls in the hoop and not relative the fixed coordinate system \mathcal{N} .

The second graph shows the angular momentum of the three components, hoop and balls, in the z-direction of the fixed coordinate system. Lastly, the third graph show the positions of the two balls relative the hoop.

4.2 Double pendulum with dual springs

4.2.1 Generalized coordinates and phase plots

In **Figure 4.6** the dynamics of the system is displayed in four graphs of the generalized coordinates over a time period of 30 seconds.

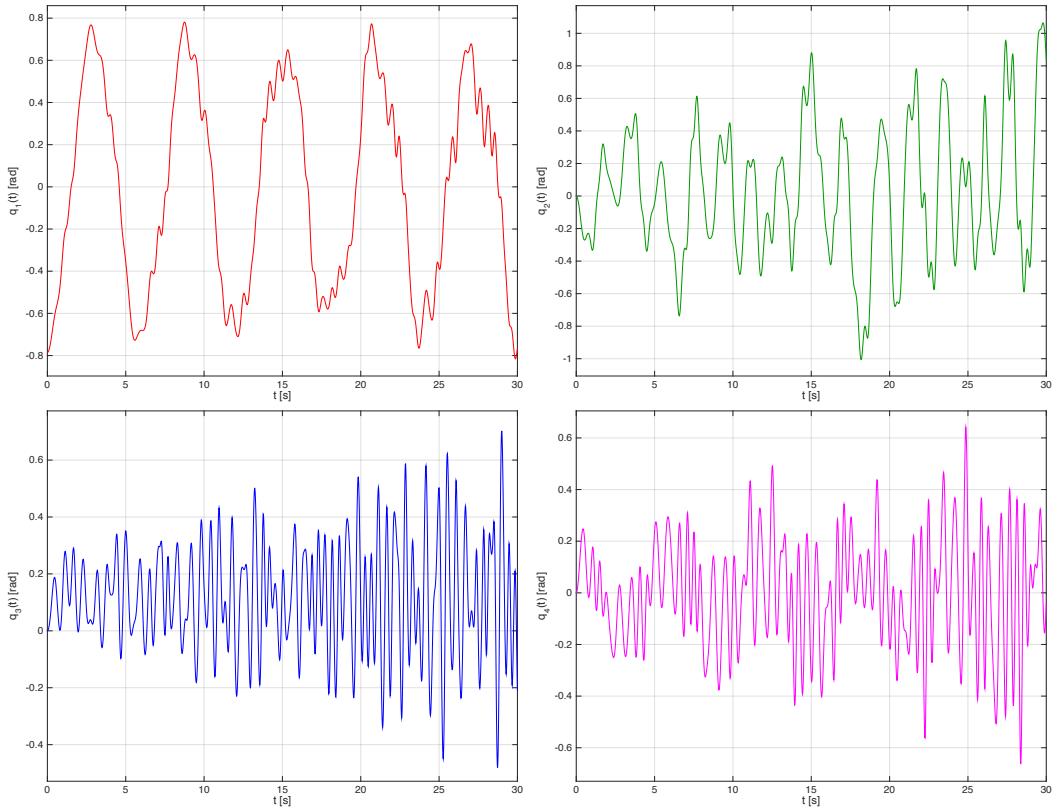


Figure 4.6. Solutions of the generalized coordinates over 30 seconds. From left to right: on top q_1 and q_2 , on bottom q_3 and q_4 .

In the top left corner we have q_1 and in the top right corner we have q_2 over time. In the bottom left corner q_3 is displayed and in the bottom right corner we

CHAPTER 4. RESULTS

have q_4 . Non of the four coordinates show a strictly periodic behavior, however one can suspect a quasi periodic behavior especially in the q_1 coordinate.

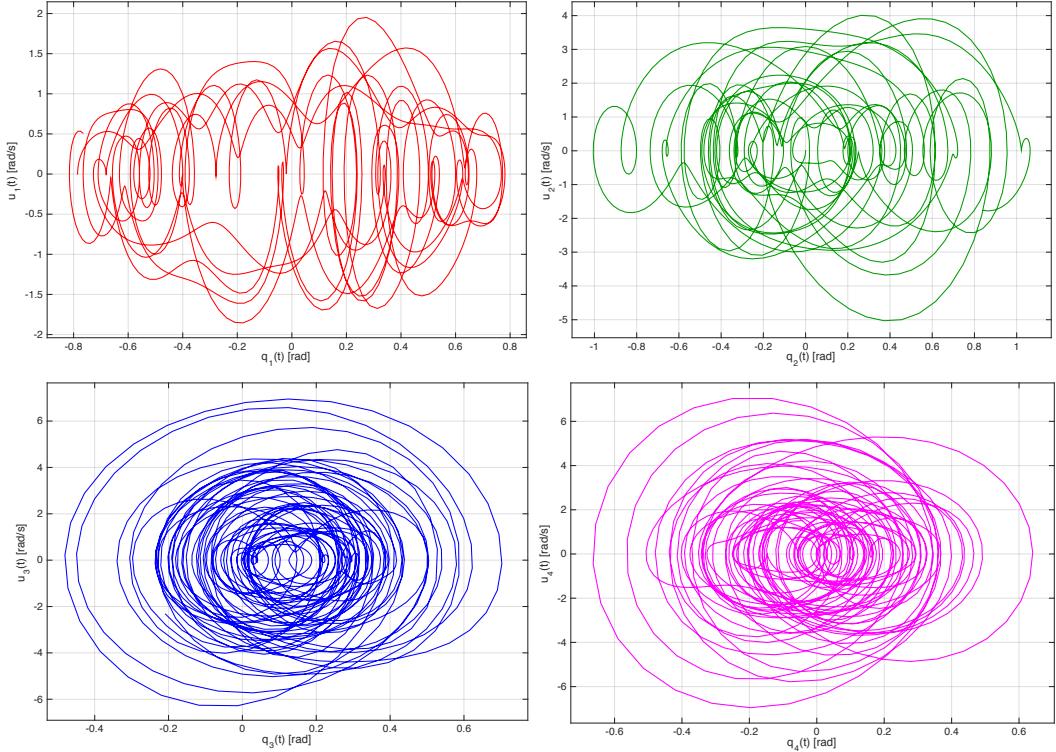


Figure 4.7. Phase portraits of the generalized coordinates and their respective velocities, over 30 seconds. From left to right: on top q_1 and q_2 , on bottom q_3 and q_4 .

In **Figure 4.7** above, the phase portraits for the four generalized coordinates and their respective velocities are shown. They are positioned in the same manner as the graphs over the four generalized coordinates alone. All four phase portraits are generated with the following initial conditions and parameter values:

Constants:

$$\boxed{l_0 = 3, l_1 = 7, l_2 = 3, m_1 = 1, m_2 = 0.5, \\ k_1 = 50, k_2 = 50, g = 9.82}$$

Initial conditions:

$$\boxed{q_1(0) = -\frac{\pi}{4}, q_2(0) = 0, q_3(0) = 0, q_4(0) = 0, \\ u_1(0) = 0, u_2(0) = 0, u_3(0) = 0, u_4(0) = 0}$$

CHAPTER 4. RESULTS

Below, in **Figure 4.8**, we see the phase portrait for q_1 over a time period of 500 seconds.

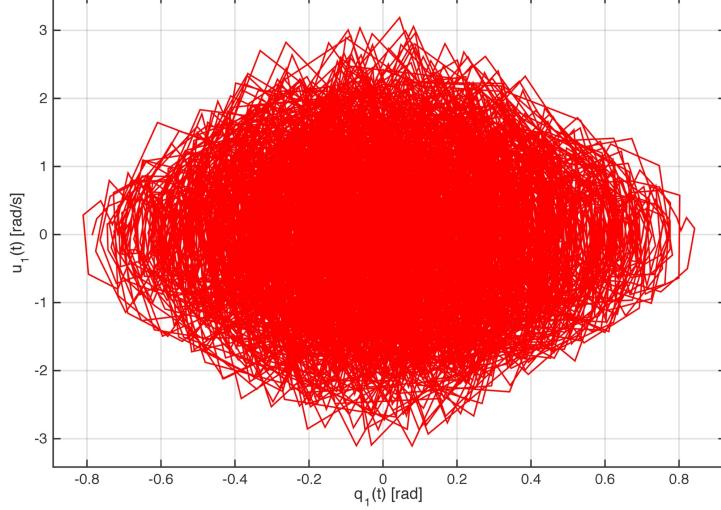


Figure 4.8. Phase portrait of the generalized coordinate q_1 and the corresponding velocity u_1 over 500 seconds.

As in the case of the *Boston hoop* problem no apparent pattern that repeat it self can be recognized in the phase portraits above. Furthermore, we can see that the phase-space will become covered over time.

4.2.2 Sensibility to initial conditions

When studying the sensitivity to initial conditions three graphs where made at which the initial condition of the angle in q_1 was shifted 0.1 and 0.01 radians from the first value of q_1 in each graph. The three graphs begin with q_1 in either -0.1 , $-\frac{\pi}{4}$ or $-\frac{\pi}{2}$ radians. This to analyze how the sensitivity is connected to energy and distance from equilibrium.

CHAPTER 4. RESULTS

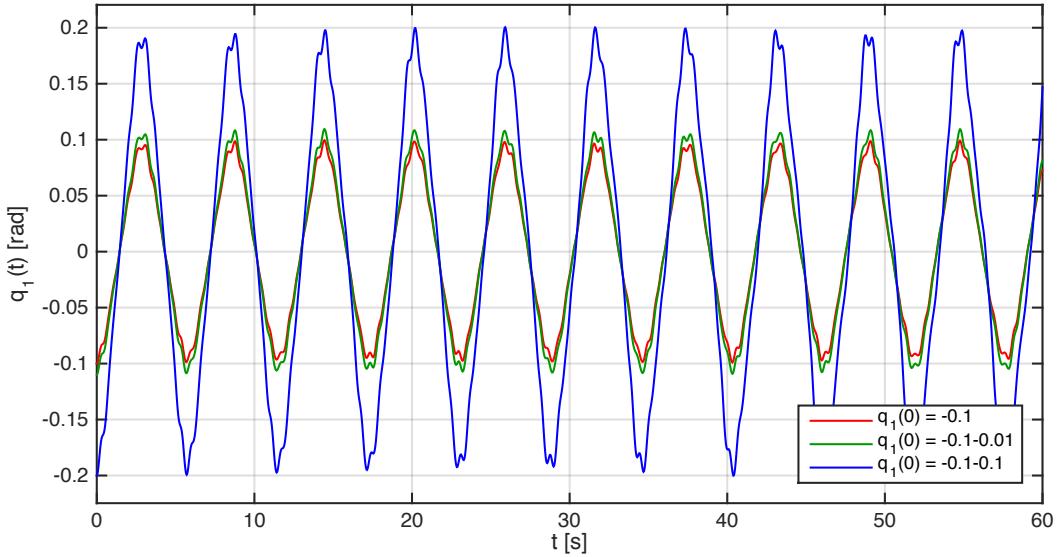


Figure 4.9. Solutions for the generalized coordinate q_1 over 60 seconds for three different initial conditions. Close to equilibrium, low energy set up.

In **Figure 4.9** we see the effect of the initial shift in q_1 near the equilibrium position, with low energy in the system, over a time period of 60 seconds.

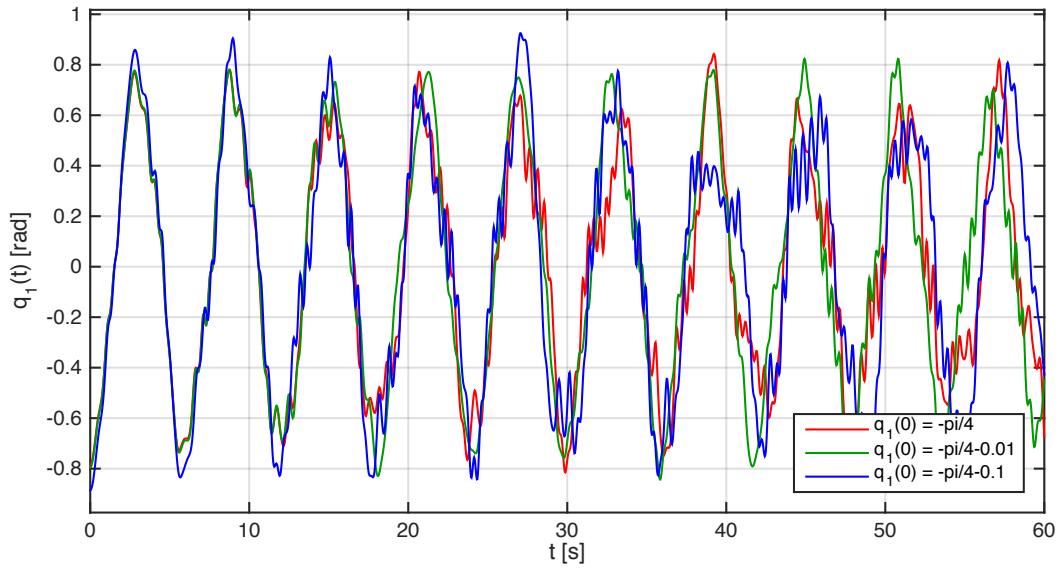


Figure 4.10. Solutions for the generalized coordinate q_1 over 60 seconds for three different initial conditions. Medium energy set up.

In **Figure 4.10** the effect of the initial shift in q_1 is shown if the shift occurs

CHAPTER 4. RESULTS

around $q_1 = -\frac{\pi}{4}$. At last we have in **Figure 4.11** below the effect of the initial shift in q_1 for the high energy case where we start with $q_1 = -\frac{\pi}{2}$.

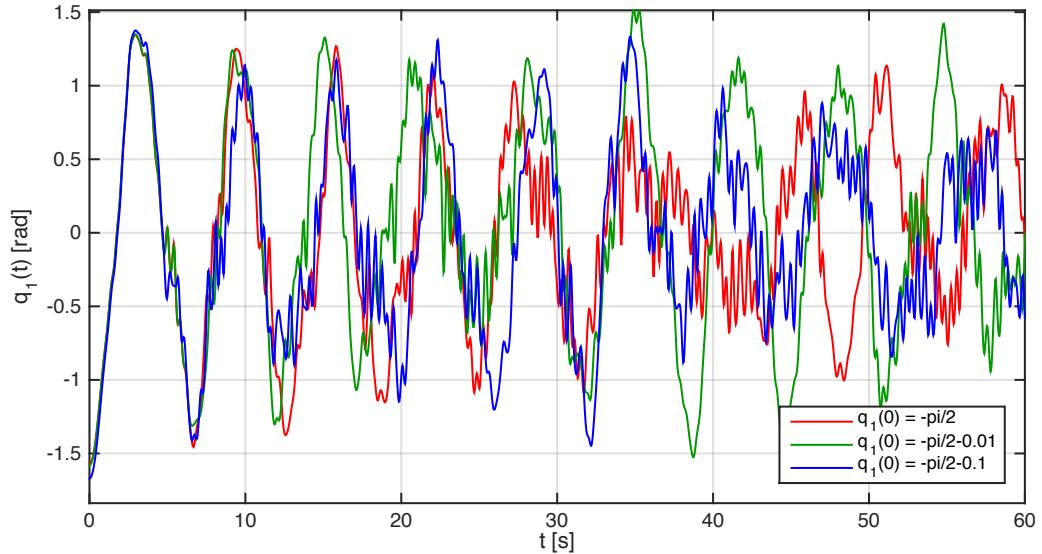


Figure 4.11. Solutions for the generalized coordinate q_1 over 60 seconds for three different initial conditions. High energy set up.

CHAPTER 4. RESULTS

4.2.2.1 Deviations due to numerical integration

A source of changes in the position of the system over time can be due to numerical errors, which can play an important part in chaotic and sensitive systems. To show this three figures where made.

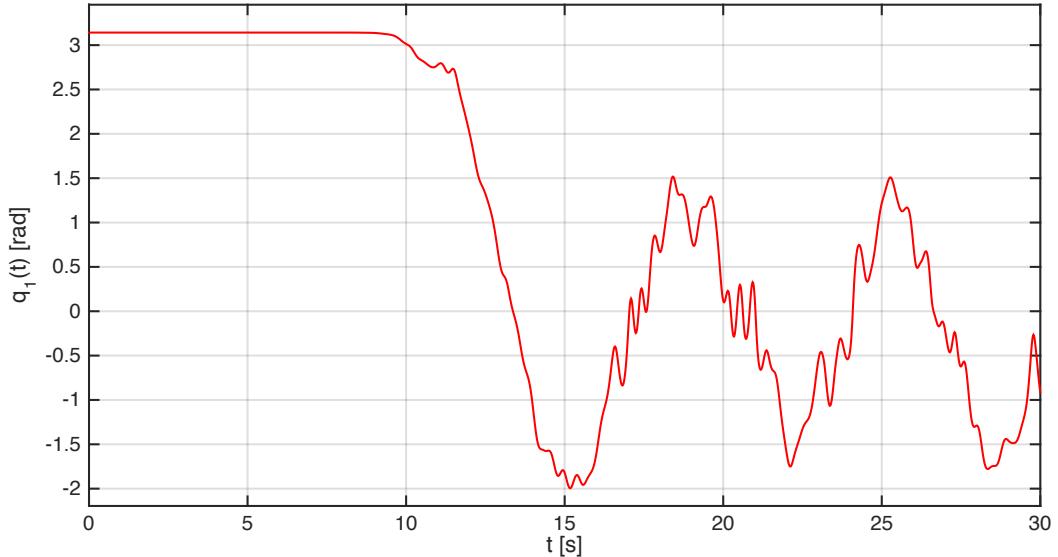


Figure 4.12. The generalized coordinate q_1 over a time period of 30 seconds.

In **Figure 4.12** we have set the pendulum in an inverted position and left it there. We used the following initial conditions

Initial conditions:

$$\boxed{\begin{aligned} q_1(0) &= \pi, & q_2(0) &= 0, & q_3(0) &= 0, & q_4(0) &= 0, \\ u_1(0) &= 0, & u_2(0) &= 0, & u_3(0) &= 0, & u_4(0) &= 0 \end{aligned}}$$

The figure shows the generalized coordinate q_1 over time and we see that the pendulum falls from the perfect inverted position although no extra external forces exists.

CHAPTER 4. RESULTS

Below, in **Figure 4.13** and **Figure 4.14**, we see two graphs of the generalized coordinate q_1 over a time period of 60 seconds.

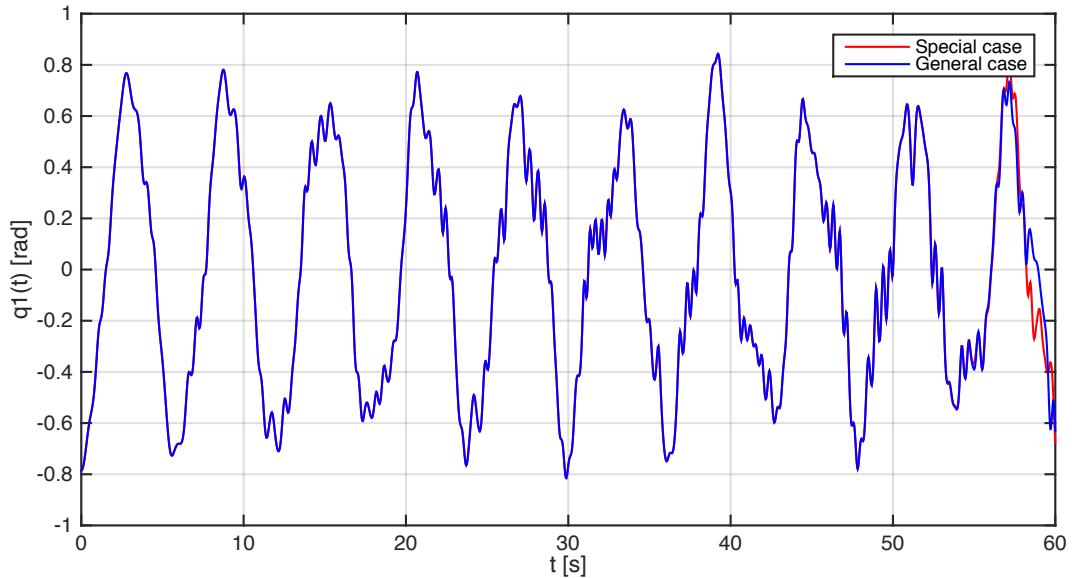


Figure 4.13. q_1 over a time period of 60 seconds. Here, "special case" and "general case" stands for the different formulations of the Euler-Lagrange equations. Both formulations are stated with the same initial conditions and parameters. In the genera case, the active forces where accounted for explicitly but no friction was added.

CHAPTER 4. RESULTS

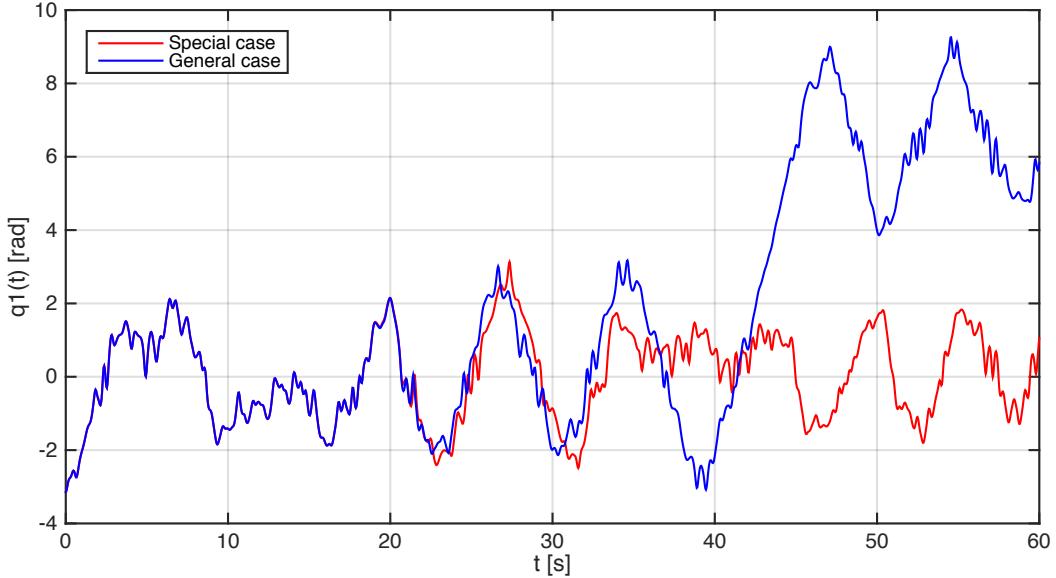


Figure 4.14. q_1 over a time period of 60 seconds for other initial conditions. Note the large separation around 40 seconds - this is because q_1 measures as an angle and we see a complete revolution for one of the cases causing a sudden increase in value.

In each graph the value of the coordinate is plotted for the two different formulations of the Euler-Lagrange equations. One of the special case and one of the general case, but with the same initial conditions and values of parameters. In the general case no friction have be added, both formulations describe the same system and situation. Nevertheless, we see that they are identical up to a point where they separate and, especially in the second case, become completely different. The initial conditions for the two graphs are as follows:

Initial conditions used in figure 4.13:

$$\begin{aligned} q_1(0) &= -\frac{\pi}{4}, \quad q_2(0) = 0, \quad q_3(0) = 0, \quad q_4(0) = 0, \\ u_1(0) &= 0, \quad u_2(0) = 0, \quad u_3(0) = 0, \quad u_4(0) = 0 \end{aligned}$$

Initial conditions used in figure 4.14:

$$\begin{aligned} q_1(0) &= -\pi, \quad q_2(0) = 0, \quad q_3(0) = 1, \quad q_4(0) = 1, \\ u_1(0) &= 0, \quad u_2(0) = 0, \quad u_3(0) = 0, \quad u_4(0) = 0 \end{aligned}$$

CHAPTER 4. RESULTS

4.2.3 Friction

We used the following parameter values and initial conditions to study the effect of introducing dissipative forces in the form of friction to the system

Constants:

$$\begin{aligned} l_0 &= 3, \quad l_1 = 7, \quad l_2 = 3, \quad m_1 = 1, \quad m_2 = 0.5, \\ k_1 &= 50, \quad k_2 = 50, \quad C_1 = 10, \quad C_2 = 10, \quad g = 9.82 \end{aligned}$$

Initial conditions:

$$\begin{aligned} q_1(0) &= -\frac{\pi}{4}, \quad q_2(0) = 0, \quad q_3(0) = 0, \quad q_4(0) = 0, \\ u_1(0) &= 0, \quad u_2(0) = 0, \quad u_3(0) = 0, \quad u_4(0) = 0 \end{aligned}$$

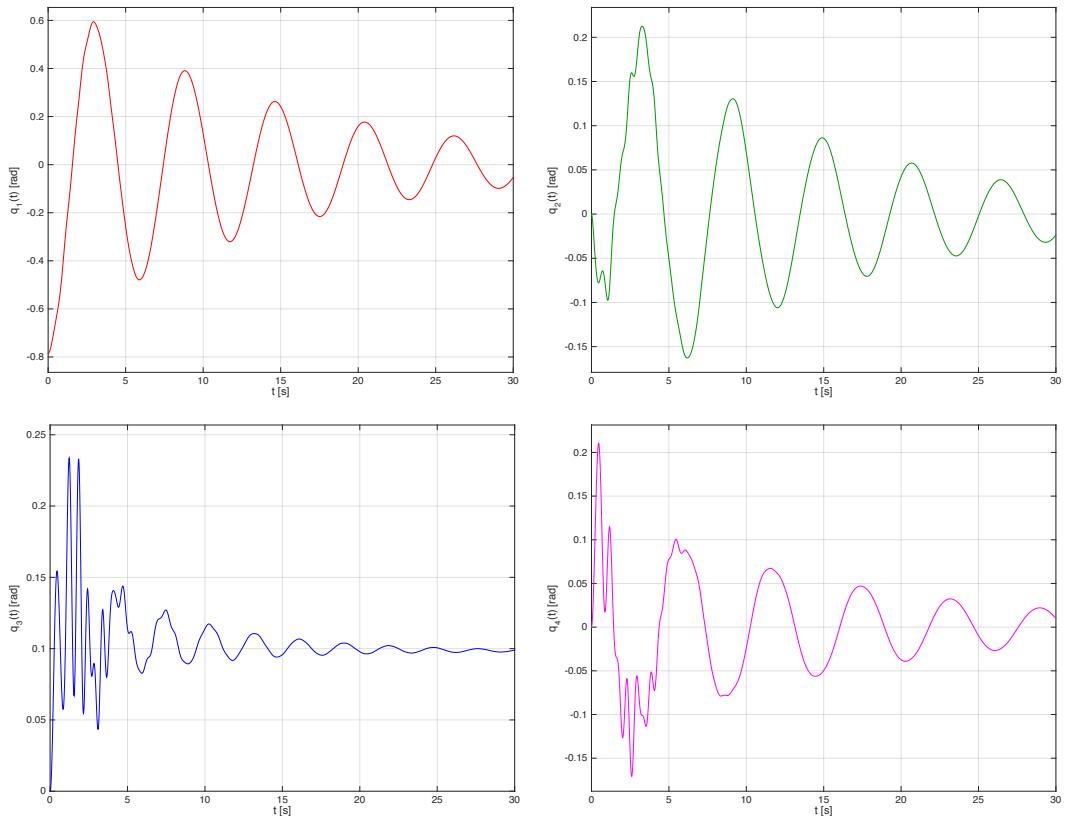


Figure 4.15. All four generalized coordinates with friction added. Friction coefficients set to $c_1 = 10$ and $c_2 = 10$

CHAPTER 4. RESULTS

In the case where we added friction in two points of the system we get the behavior as shown in **Figure 4.15** above. Here both of the friction coefficients were set to 10. All generalized coordinates are clearly tending to their equilibrium positions. To clearly see the difference in the dynamics of the system we plot the generalized coordinate q_1 with and without friction in **Figure 4.16** below.

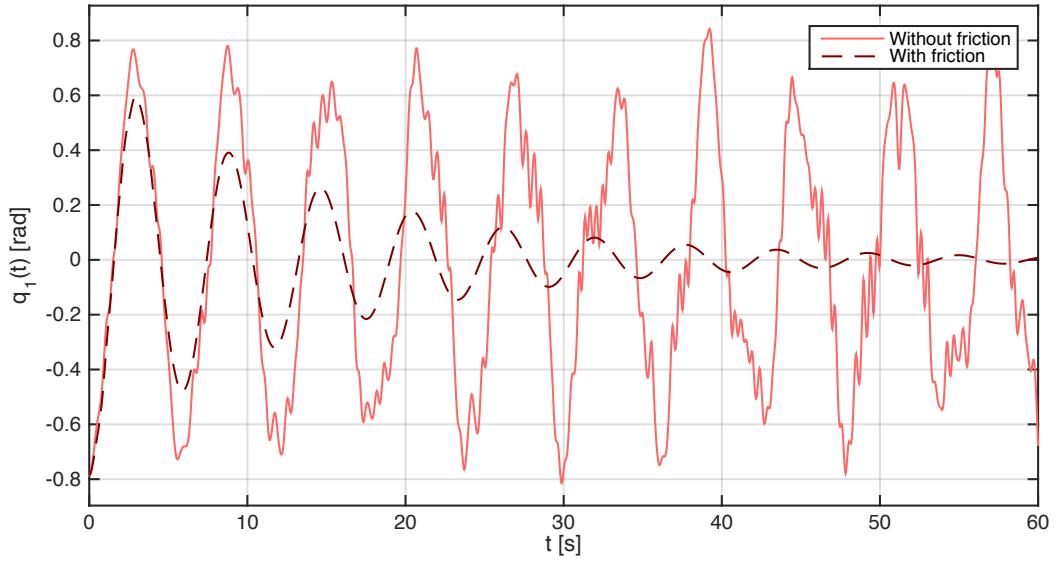


Figure 4.16. The generalized coordinate q_1 over a time period of 60 seconds with and without friction.

We also look at the total energy, with and without friction in **Figure 4.17**.

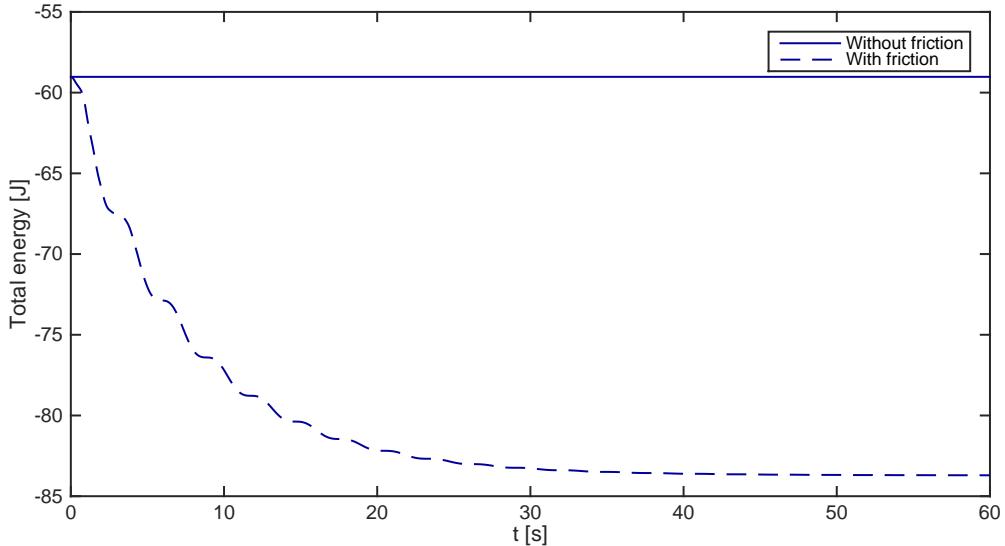


Figure 4.17. Total energy for the system, with and without friction.

Chapter 5

Discussion

In this chapter we discuss the results showcased in the previous section and what conclusions about the two problems and their dynamics can be stated. We also discuss the power of the Lagrangian mechanics together with computer algebra.

5.1 *Boston Hoop*

First and foremost we notice in **Figure 4.4** that the total energy and angular momentum for this system are conserved. Furthermore, the graphs of the generalized coordinates in **Figure 4.1** do not exhibit any impossible or unlikely properties as discontinuities or extreme values. The animations done for this system seem also to generate a plausible result, hence the geometry and mechanics of the problem seem to be correctly set up.

In addition, the graphs in **Figure 4.5** showcase properties of the dynamics of the system that agrees with stated theory and with what can be expected from a system as the *Boston Hoop*. In the first few seconds of the simulation we see that q_1 accelerates whilst q_3 that starts near the equilibrium position do not. At the same time the hoop it self is not effected and continue spinning with the same velocity it started with. When the first ball corresponding to q_1 reaches the equilibrium point relative to the spring it is attached to, when q_1 is zero, the contribution to the angular moment of the ball becomes zero. Because of the fact that the total angular momentum must be conserved this sets the hoop spinning faster and thus gaining kinetic energy. At the same time the ball loses energy and moves slowly relative the hoop. We can also see that the other ball that corresponds to q_3 that started near the equilibrium position relative to the spring is thrown outwards, increasing q_3 , due to the hoop accelerating and thus gaining angular momentum.

This pattern where the kinetic energy and angular momentum is shifted between the different components of the system can be seen throughout the graphs shown in **Figure 4.5**. All according to the stated theory, which consequently strengthens the conclusion that the solution showcased in the previous section is correct.

Furthermore, from **Figure 4.1** and **Figure 4.2** we can observe that the sys-

CHAPTER 5. DISCUSSION

tem do not tend to have any strictly periodic behavior, neither the graph of the generalized coordinates nor the graph of the phase portraits indicates a periodic behavior. The phase portraits do not display a closed curve that is repeated over time, moreover they seem to cover the whole of the phase-space. This is further shown in **Figure 4.3** where the simulation time is extend to 1000 seconds. This together with the fact that the graphs of the generalized coordinates do not display any periodicity in time leads to the conclusion that this system probably is chaotic.

5.2 Double pendulum with dual springs

In **Figure 4.6** we can see the four generalized coordinates plotted over a time period of 30 seconds. There seem to be no apparent strict periodicity in any of the graphs but, especially for the generalized coordinate q_1 , we can notice some kind of quasi periodicity. Furthermore, if we observe the phase portraits in **Figure 4.7** we don't see a repeated pattern that gets thicker with time. But instead we see that the curves tend to cover the whole of the phase space. This statement is further confirmed when we look at the phase portrait of the generalized coordinate q_1 over a time period of 500 seconds in **Figure 4.8**. Thus we can say that the system in question seems to be aperiodic and chaotic in its dynamics.

When further analysing the chaotic behavior of the system an important aspect is its sensitivity to initial conditions or disturbances. This was studied in **section 4.2.2** where three plots of the sensitivity where displayed. In **Figure 4.11** where we tested the systems sensitivity at an high level of energy we see that the movement in q_1 separates for small deviations in initial conditions in q_1 , already within the first 10 seconds of the simulation. This, although the deviation in the initial condition of q_1 are small, approximately 1% of the undisturbed case. If we look at the movement of the system in q_1 further along the time line we see that the position of the system completely separated from each-other due to these small deviations in initial conditions.

In **Figure 4.9** and **Figure 4.10** are done in the same manner and with the same absolute deviation in initial condition in q_1 as the previous mentioned case, but now around two new different undisturbed values of q_1 . Both for systems with smaller amount of energy. Here we see the same pattern, that small disturbances in the initial condition will give a whole new dynamic of the system but not as quickly and prominent as in the earlier mentioned case. In Figure 4.9 where the system is placed near to its equilibrium position we see that the difference in the dynamics of the system are much smaller and happens more slowly than in the two other cases.

This further indicates that the system is of an chaotic nature and that the chaotic behavior is more prominent when the system contains more energy. We also see that the sensitivity to initial conditions becomes weaker with decreasing energy in the system.

A cause of error to the solutions of the generalized coordinates of the system, i.e. the dynamics of the system, is the numerical integration of the equations of

CHAPTER 5. DISCUSSION

motion derived in the Lagrangian method. Because we have an chaotic system that is sensitive to disturbances it is of interest to analyse if the numerical errors play an important roll in the final solution of this specific system. To study this three figures where made, shown in **section 4.2.2.1**.

In **Figure 4.12** we have set the pendulum in an perfect inverted position. If the solutions would be perfect this figure would have displayed an constant line over time. This is although not what happens, instead the pendulum falls after approximately 10 seconds even though the conditions are ideal. We see that the numerical error have an effect on the system in its unstable positions, which isn't especially unanticipated. This is expected for non-chaotic system as well. But it shows that the numerical error can completely alter the future behavior of the system.

In **Figure 4.13** and **Figure 4.14** we have displayed the generalized coordinate q_1 over a time period of 60 seconds, computed with the special case of the Euler-Lagrange equations and the general case of the Euler-Lagrange equations. Both describe the same system under the same conditions, no friction added in the general case. They should according to theory be equivalent and one would expect the two curves in each graph to coincide with each other. But this isn't the case for the two figures. We see that in the first figure, **Figure 4.13**, the two curves are essentially identical up to a point around 55 seconds where they separate. In the second figure **Figure 4.14**, which is a case where the system inhibits more energy than the case displayed in the first figure, we see the two curves separate completely after around 40 seconds. This due to the pendulum in the general case crossing the inverted position whilst the special case do not.

The explanation to why we see the separation of the two solutions that are expected to be equivalent is a combination of the symbolic nature of Sofia, numerical errors and that the system is chaotic and thus sensitive to disturbances. Although the two cases of the Lagrangian method are equivalent and describe the same system under the same conditions the symbolic presentation of the solutions are different. They are mathematically equivalent but not symbolically. As an example we have that

$$\sin(\alpha + \beta) = \sin(\alpha)\cos(\beta) + \sin(\beta)\cos(\alpha) \quad (5.1)$$

where the expression on the right and left side are mathematically equivalent but not symbolically equal. This is the case when we compute the solution for the dynamics of the system with the special and general Eular-Lagrange equations. They are equal, otherwise they would not coincide perfectly in the first few seconds of the simulations, but not in an symbolical sense. When the equations then are numerically integrated the error produced will not be the same in the two cases. This error can then be seen as a small disturbance to the solution of the system. Although small, it will be enough to alter the future dynamics of the system. This is what can be seen in the two figures, due to the chaotic nature of the system or in other words this shows that the system in question actually is of a chaotic nature.

CHAPTER 5. DISCUSSION

The last point studied was the effect of introducing dissipative forces in form of friction in two joints of the system. In **Figure 4.15** we see all four generalized coordinates plotted against time for a period of 30 seconds. We see that the amplitude of each coordinate slowly decreases towards the equilibrium position with time. Precisely what could be expected when friction is introduced to the system.

A figure to showcase the difference of the dynamics of the system when friction is introduced can be seen in **Figure 4.16**. We see that the motion of the system becomes somewhat more periodic and less chaotic over time than in the case without dissipative forces. We can also see that the total energy of the system decreases with time to a minimum in **Figure 4.17**, precisely as expected.

5.3 The power of this method

The two systems studied in this work have a complex geometry and the equations of motion for both systems turns out to be at least as complex as the geometry and furthermore need numerical integration to be solved. For this reason alone computer assistance is a necessary tool in solving the dynamics of these systems. Sophia provides useful tools that do not already exist in Maple and make the implementation of the algorithmic structure of the Lagrangian method easy and efficient despite the complexity of the problems studied. Precisely the combination of the algorithmic structure of the Lagrangian mechanics and the simplicity of the procedures provided by Sophia makes this method so powerful and versatile. A wide range of otherwise tough and difficult classical mechanical systems can be quickly and efficiently solved by this method, which in part is indicated by the problems solved in this thesis.

The quite easy way of exporting data of the solutions from Sophia to Blender where the ability to script the 3D setup of the problem in python makes the visualization of the solutions simple and quick. The high quality and ability to produce animations of picture film quality was priceless in understanding and determining the plausibility of the dynamics of the systems. It also opens up for new ways of simulating the system in the future where 3D printing and VR-technology maybe become more common and developed.

Chapter 6

Summary and conclusion

In this work we took an entirely computer-based approach to modeling and solving mechanical systems. We presented two complex systems with several degrees of freedom and different geometric constraints and described how to model them using the Lagrangian theory. We then showed how to implement this theory in code using Maple and the plug-in package Sophia.

A brief analysis of both systems was also made to get a qualitative understanding of their motion. Some of the things we looked at were: Sensitivity to initial conditions, response to friction and the transfer of energy between different parts of a system over time.

To complement the analysis of motion-data, we also made 3D-visualizations of both systems using Blender. We found two main advantages of using Blender compared to ordinary plotting tools such as those found in Maple or Matlab: First, the resulting image- and video renderings have an unmatched aesthetic quality (making them suitable for presentation). Second, it turns out that once the geometric model for a system is set up (a one-time process), we were able to achieve very high frame-rate animations (~ 60 fps in wire-frame-mode). This meant that we could maintain a very small time-step in the numerical solution and still get a real-time animation of the system. To quickly be able to see the physical movement is key to modeling, tweaking and analyzing problems of this kind. Videos for the systems can be found at:

<https://github.com/filipstrand/lagrangian-mechanics>

Bibliography

- [1] Apazidis, Nicholas, N.A, 2016. *Computer Algebra Assisted Mechanics*. KTH,Mechanics
- [2] Apazidis, Nicholas, N.A, 2012. *Mekanik II, Partikelsystem, stel kropp och analytisk mekanik*. Lund: StudentLitteratur AB
- [3] Essén, Hanno, 2006. *The Theory of Lagrange's method*. KTH Mechanics
- [4] Thor, Anders J., 1980. *Analytisk Mekanik*. KTH, Institutionen för Mekanik
- [5] Sussman, Gerald and Wisdom, Jack, 2000. *Structure and Interpretation of Classical Mechanics*. London: The MIT Press
- [6] R.Nave, 2016. *Conservation Laws* [online] Available at:<<http://hyperphysics.phy-astr.gsu.edu/hbase/conser.html>> [Accessed 12 April 2016]
- [7] J. Peraire, S. Widnall, 2008. *2D Rigid Body Dynamics*.[pdf] Massachusetts Institute of Technology. Available at: <http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-07-dynamics-fall-2009/lecture-notes/MIT16_07F09_Lec21.pdf> [Accessed 20 Mars 2016]
- [8] Standford Encyclopedia of Philosophy, 2008. *Chaos* [online](13 October 2015) Available at:<<http://plato.stanford.edu/entries/chaos/#DefChaDetNonSenDep>> [Accessed 15 April 2016]
- [9] Weisstein Eric, 2016. *Torus* [online](22 April 2016) Available at: <<http://mathworld.wolfram.com/Torus.html>> [Accessed 15 February 2016]
- [10] Blender Foundation, 2016. *Blender*™(2.77a). [computer program] Blender Foundation. Available at: <<https://www.blender.org>> [Accessed 2 Mars 2016]

Appendix A

Maple and Sophia Code

Here we present the Maple and Sophia code for the two systems. The code is also available for download at

<https://github.com/filipstrand/lagrangian-mechanics>

A.1 *Boston hoop*

```
read "/path/to/SophiaV6.txt";
read "/path/to/Graphics.txt";
with(plots): with(plottools): with(linalg): with(ArrayTools):

# -----Declaring time dependence-----
dependsTime(q1, q2, q3, u1, u2, u3);

# -----Coordinate Systems-----
&rot([N, A, 3, q2]):

# -----Position vector to balls-----
r_B1 := (A &ev [R*cos(H0-10/R-q1),0,R*sin(H0-10/R-q1)]):
r_B2 := (A &ev [R*cos(H1+11/R+q3),0,R*sin(H1+11/R+q3)]):

# -----Velocity vectors of the balls-----
v_B1 := simplify(N &fdt r_B1):
v_B2 := simplify(N &fdt r_B2):

# -----Angular momentum (H)-----
omega := <0,0,q2t>;
I_hoop := << ((3/4)*r^2+R^2)*m1 | 0 | 0 >,
           < 0 | ((5/8)*r^2+(1/2)*R^2)*m1 | 0 >,
           < 0 | 0 | ((5/8)*r^2+(1/2)*R^2)*m1 >>;
H_hoop := I_hoop.omega;
H_B1 := Ecross(r_B1,m0 &**v_B1);
H_B2 := Ecross(r_B2,m0 &**v_B2);

# -----Kinetic energy (T)-----
T_hoop_rotation := (1/2)*(omega^+).H_hoop;
T_B1_translation := (1/2)*m0*(v_B1 &o v_B1);
T_B2_translation := (1/2)*m0*(v_B2 &o v_B2);
```

APPENDIX A. MAPLE AND SOPHIA CODE

```

T := T_hoop_rotation + T_B1_translation + T_B2_translation:

# -----Potential energy (V)-----
height_of_B1 := r_B1 &o (N &ev [0, 0, 1]):
height_of_B2 := r_B2 &o (N &ev [0, 0, 1]):
V_B1 := m0*g*height_of_B1:
V_B2 := m0*g*height_of_B2:
V_S1 := (1/2)*k0*(R*q1)^2:
V_S2 := (1/2)*k1*(R*q3)^2:
V := V_B1 + V_B2 + V_S1 + V_S2:

# -----Construct the Lagraingian-----
L := T-V:
L := subs(q1t = u1, q2t = u2, q3t = u3, L):

# -----Construct the Euler-Lagrange equations-----
dLdq1 := diff(L, q1):
dLdu1 := diff(L, u1):
dtdLdu1 := &dt(dLdu1):
eq1 := dtdLdu1-dLdq1 = 0:
eq1 := subs(q1t = u1, q2t = u2, q3t = u3, eq1):

dLdq2 := diff(L, q2):
dLdu2 := diff(L, u2):
dtdLdu2 := &dt(dLdu2):
eq2 := dtdLdu2-dLdq2 = 0:
eq2 := subs(q1t = u1, q2t = u2, q3t = u3, eq2):

dLdq3 := diff(L, q3):
dLdu3 := diff(L, u3):
dtdLdu3 := &dt(dLdu3):
eq3 := dtdLdu3-dLdq3 = 0:
eq3 := subs(q1t = u1, q2t = u2, q3t = u3, eq3):

kde := {q1t = u1, q2t = u2, q3t = u3}:

eq1 := {u1t = solve(eq1, u1t)}:
eq2 := {u2t = solve(eq2, u2t)}:
eq3 := {u3t = solve(eq3, u3t)}:

eqs := eq1 union eq2 union eq3 union kde:

# -----The initial conditions-----
Initcond := {q1(0) = Pi/2-0.3,
             q2(0) = 0,
             q3(0) = 0.1,
             u1(0) = 0,
             u2(0) = 1,
             u3(0) = 0}:

eqst := subs(toTimeFunction, eqs):

# -----The constants-----
param := {R = 5,
          r = 0.6,
          H0 = Pi-0.1,
          H1 = Pi+0.1,
          l0 = 5*Pi/2,
          l1 = 5*Pi/2,

```

APPENDIX A. MAPLE AND SOPHIA CODE

```

m0 = 100,
m1 = 10,
k0 = 900,
k1 = 300,
g = 9.82}:

eqst := subs(param, eqst):

# -----Numericall solve the system-----
ff := dsolve(eqst union Initcond, {q1(t),
                                     q2(t),
                                     q3(t),
                                     u1(t),
                                     u2(t),
                                     u3(t)}, type = numeric, maxfun = 0):

numberOfPoints := 1800:
integrationTime := 30:
_plot := odeplot(ff,
                  [[t,q1(t)],
                   [t,q2(t)],
                   [t,q3(t)],
                   [t,u1(t)],
                   [t,u2(t)],
                   [t,u3(t)]],
                  0..integrationTime,numpoints=numberOfPoints):

# -----Exporting the data to a .txt file-----
# The code below simply exports the data in _plot
# in a text file name 'maple_data.txt' with the
# following format (columns are seperated by space):
# | t | q_1 | ... | q_n | u_1 | ... | u_n |
# |---+---+---+---+---+---+---|
# | t_0 |     |     |     |     |     |     |
# | t_1 |     |     |     |     |     |     |
# | .  |     |     |     |     |     |     |
# | .  |     |     |     |     |     |     |
# | t_k |     |     |     |     |     |     |

t_data := ((plottools[getdata](_plot, "points")[1])[3])[[1..numberOfPoints],1]:
q1_data := ((plottools[getdata](_plot, "points")[1])[3])[[1..numberOfPoints],2]:
q2_data := ((plottools[getdata](_plot, "points")[2])[3])[[1..numberOfPoints],2]:
q3_data := ((plottools[getdata](_plot, "points")[3])[3])[[1..numberOfPoints],2]:
u1_data := ((plottools[getdata](_plot, "points")[4])[3])[[1..numberOfPoints],2]:
u2_data := ((plottools[getdata](_plot, "points")[5])[3])[[1..numberOfPoints],2]:
u3_data := ((plottools[getdata](_plot, "points")[6])[3])[[1..numberOfPoints],2]:

_data := Concatenate(2,
                      t_data,
                      q1_data,
                      q2_data,
                      q3_data,
                      u1_data,
                      u2_data,
                      u3_data):

writedata("/path/to/maple_data.txt",convert(_data,array),float):

```

APPENDIX A. MAPLE AND SOPHIA CODE

A.2 *Double pendulum with dual springs (no friction)*

```

read "/path/to/SophiaV6.txt":
read "/path/to/Graphics.txt":
with(plots): with(plottools): with(linalg): with(ArrayTools):

# -----Declaring time dependence-----
dependsTime(q1, q2, q3, q4, u1, u2, u3, u4):

# -----Coordinate Systems-----
&rot([N, A, 2, q1]):
&rot([A, B, 2, q2]):

# -----Position vectors-----
r_0 := (N &ev [0,0,0]):
r_A := (N &ev [10+q4,0,0]): 
r_B := (N &ev [10+q4,0,0]) &++ (A &ev [0, 0, -11/2]):
r_C := (N &ev [10+q4,0,0]) &++ (A &ev [0, 0, -11]):
r_D := (N &ev [10+q4,0,0]) &++ (A &ev [0, 0, -11]) &++ (B &ev [0, 0, -(12 + q3)]):

# -----Velocity vectors -----
v_B := simplify(N &fdt r_B):
v_D := simplify(N &fdt r_D):

# -----Angular momentum (H)-----
omega := <0,q1t,0>:
I_rod := << (1/12)*m1*l1^2 | 0 | 0 ,
          < 0 | (1/12)*m1*l1^2 | 0 >,
          < 0 | 0 | 0 >>:
H_rod := I_rod.omega:

# -----Kinetic energy (T)-----
T_rod_rotation := (1/2)*(omega^+).H_rod:
T_rod_translation := (1/2)*m1*(v_B &o v_B):
T_ball_translation := (1/2)*m2*(v_D &o v_D):
T := T_rod_rotation + T_rod_translation + T_ball_translation:

# -----Potential energy (V)-----
height_of_r_B := r_B &o (N &ev [0, 0, 1]):
height_of_r_D := r_D &o (N &ev [0, 0, 1]):
V_rod := m1*g*height_of_r_B:
V_ball := m2*g*height_of_r_D:
V_S1 := (1/2)*k1*q4^2:
V_S2 := (1/2)*k2*q3^2:
V := V_rod + V_ball + V_S1 + V_S2:

# -----Construct the Lagrangian-----
L := T-V:
L := subs(q1t = u1, q2t = u2, q3t = u3, q4t = u4, L):

# -----Construct the Euler-Lagrange equations-----
dLdq1 := diff(L, q1):
dLdu1 := diff(L, u1):
dtdLdu1 := &dt(dLdu1):
eq1 := dtdLdu1-dLdq1 = 0:
eq1 := subs(q1t = u1, q2t = u2, q3t = u3, q4t = u4, eq1):

dLdq2 := diff(L, q2):

```

APPENDIX A. MAPLE AND SOPHIA CODE

```

dLdu2 := diff(L, u2):
dtdLdu2 := &dt(dLdu2):
eq2 := dtdLdu2-dLdq2 = 0:
eq2 := subs(q1t = u1, q2t = u2, q3t = u3, q4t = u4, eq2):

dLdq3 := diff(L, q3):
dLdu3 := diff(L, u3):
dtdLdu3 := &dt(dLdu3):
eq3 := dtdLdu3-dLdq3 = 0:
eq3 := subs(q1t = u1, q2t = u2, q3t = u3, q4t = u4, eq3):

dLdq4 := diff(L, q4):
dLdu4 := diff(L, u4):
dtdLdu4 := &dt(dLdu4):
eq4 := dtdLdu4-dLdq4 = 0:
eq4 := subs(q1t = u1, q2t = u2, q3t = u3, q4t = u4, eq4):

kde := {q1t = u1, q2t = u2, q3t = u3, q4t = u4}:

eq1 := {u1t = solve(eq1, u1t)}:
eq2 := {u2t = solve(eq2, u2t)}:
eq3 := {u3t = solve(eq3, u3t)}:
eq4 := {u4t = solve(eq4, u4t)}:

eqs := eq1 union eq2 union eq3 union eq4 union kde:

# -----The initial conditions-----
Initcond := {q1(0) = -Pi/4,
             q2(0) = 0,
             q3(0) = 0,
             q4(0) = 0,
             u1(0) = 0,
             u2(0) = 0,
             u3(0) = 0,
             u4(0) = 0}:

eqst := subs(toTimeFunction, eqs):

# -----The constants-----
param := {l0 = 3,
           l1 = 7,
           l2 = 3,
           m1 = 1,
           m2 = 0.5,
           k1 = 50,
           k2 = 50,
           g = 9.82}:

eqst := subs(param, eqst):

# -----Numerically solve the system-----
ff := dsolve(eqst union Initcond, {q1(t),
                                      q2(t),
                                      q3(t),
                                      q4(t),
                                      u1(t),
                                      u2(t),
                                      u3(t),
                                      u4(t)}, type = numeric, maxfun = 0):

```

APPENDIX A. MAPLE AND SOPHIA CODE

```

numberOfPoints:= 720:
integrationTime:= 30:
_pplot := odeplot(ff,
    [[t,q1(t)],
     [t,q2(t)],
     [t,q3(t)],
     [t,q4(t)],
     [t,u1(t)],
     [t,u2(t)],
     [t,u3(t)],
     [t,u4(t)]],0..integrationTime,numpoints=numberOfPoints):

# -----Exporting the data to a .txt file-----
# The code below simply exports the data in _plot
# in a text file name 'maple_data.txt' with the
# following format (columns are seperated by space):

# | t | q_1 | ... | q_n | u_1 | ... | u_n |
# |-----+-----+-----+-----+-----+-----|
# | t_0 |   |   |   |   |   |   |
# | t_1 |   |   |   |   |   |   |
# | . |   |   |   |   |   |   |
# | . |   |   |   |   |   |   |
# | t_k |   |   |   |   |   |   |

t_data := ((plottools[getdata](_plot, "points")[1])[3])[[1..numberOfPoints],1]:
q1_data := ((plottools[getdata](_plot, "points")[1])[3])[[1..numberOfPoints],2]:
q2_data := ((plottools[getdata](_plot, "points")[2])[3])[[1..numberOfPoints],2]:
q3_data := ((plottools[getdata](_plot, "points")[3])[3])[[1..numberOfPoints],2]:
q4_data := ((plottools[getdata](_plot, "points")[4])[3])[[1..numberOfPoints],2]:
u1_data := ((plottools[getdata](_plot, "points")[5])[3])[[1..numberOfPoints],2]:
u2_data := ((plottools[getdata](_plot, "points")[6])[3])[[1..numberOfPoints],2]:
u3_data := ((plottools[getdata](_plot, "points")[7])[3])[[1..numberOfPoints],2]:
u4_data := ((plottools[getdata](_plot, "points")[8])[3])[[1..numberOfPoints],2]:

_data := Concatenate(2,
    t_data,
    q1_data,
    q2_data,
    q3_data,
    q4_data,
    u1_data,
    u2_data,
    u3_data,
    u4_data):

writedata("/path/to/maple_data.txt",convert(_data,array),float):

```

APPENDIX A. MAPLE AND SOPHIA CODE

A.3 Double pendulum with dual springs (friction)

```

read "/path/to/SophiaV6.txt":
read "/path/to/Graphics.txt":
with(plots): with(plottools): with(linalg): with(ArrayTools):

# -----Declaring time dependence-----
dependsTime(q1, q2, q3, q4, u1, u2, u3, u4):

# -----Coordinate Systems-----
&rot([N, A, 2, q1]):
&rot([A, B, 2, q2]):

# -----Position vectors-----
r_0 := (N &ev [0,0,0]):
r_A := (N &ev [10+q4,0,0]):
r_B := ((N &ev [10+q4,0,0]) &++ (A &ev [0, 0, -11/2])):
r_C := ((N &ev [10+q4,0,0]) &++ (A &ev [0, 0, -11])):
r_D := ((N &ev [10+q4,0,0]) &++ (A &ev [0, 0, -11]) &++ (B &ev [0, 0, -(12 + q3)])):

# -----Velocity vectors -----
v_B := simplify(N &fdt r_B):
v_D := simplify(N &fdt r_D):

# -----Angular momentum (H)-----
omega := <0,q1t,0>:
I_rod := << (1/12)*m1*l1^2 | 0 | 0 >,
           < 0 | (1/12)*m1*l1^2 | 0 >,
           < 0 | 0 | 0 >>:
H_rod := I_rod.omega:

# -----Kinetic energy (T)-----
T_rod_rotation := (1/2)*(omega^+).H_rod:
T_rod_translation := (1/2)*m1*(v_B &o v_B):
T_ball_translation := (1/2)*m2*(v_D &o v_D):
T := T_rod_rotation + T_rod_translation + T_ball_translation:

# -----Construct the Generalized forces-----
F1 := N &to (N &ev [-k1*q4,0,0]):
F2 := N &to (N &ev [0,0,-m1*g]):
F3 := N &to (B &ev [0,0,-k2*q3]):
F4 := N &to (B &ev [0,0,k2*q3]):
F5 := N &to (N &ev [0,0,-m2*g]):

r_0 := N &to r_0:
r_A := N &to r_A:
r_B := N &to r_B:
r_C := N &to r_C:
r_D := N &to r_D:

Q1 := F1 &o [map(diff,r_A[1],q1),N] +
      F2 &o [map(diff,r_B[1],q1),N] +
      F3 &o [map(diff,r_C[1],q1),N] +
      F4 &o [map(diff,r_D[1],q1),N] +
      F5 &o [map(diff,r_D[1],q1),N] -
      C1*q1t:

Q2 := F1 &o [map(diff,r_A[1],q2),N] +

```

APPENDIX A. MAPLE AND SOPHIA CODE

```

F2 &o [map(diff,r_B[1],q2),N] +
F3 &o [map(diff,r_C[1],q2),N] +
F4 &o [map(diff,r_D[1],q2),N] +
F5 &o [map(diff,r_D[1],q2),N] -
C2*q2t:

Q3 := F1 &o [map(diff,r_A[1],q3),N] +
F2 &o [map(diff,r_B[1],q3),N] +
F3 &o [map(diff,r_C[1],q3),N] +
F4 &o [map(diff,r_D[1],q3),N] +
F5 &o [map(diff,r_D[1],q3),N]:

Q4 := F1 &o [map(diff,r_A[1],q4),N] +
F2 &o [map(diff,r_B[1],q4),N] +
F3 &o [map(diff,r_C[1],q4),N] +
F4 &o [map(diff,r_D[1],q4),N] +
F5 &o [map(diff,r_D[1],q4),N]:

# -----Construct the Euler-Lagrange equations-----
T := subs(q1t = u1, q2t = u2, q3t = u3, q4t = u4, T):

dTdq1 := diff(T, q1):
dTdu1 := diff(T, u1):
dtdTdu1 := &dt(dTdu1):
eq1 := dtdTdu1-dTdq1-Q1 = 0:
eq1 := subs(q1t = u1, q2t = u2, q3t = u3, q4t = u4, eq1):

dTdq2 := diff(T, q2):
dTdu2 := diff(T, u2):
dtdTdu2 := &dt(dTdu2):
eq2 := dtdTdu2-dTdq2-Q2 = 0:
eq2 := subs(q1t = u1, q2t = u2, q3t = u3, q4t = u4, eq2):

dTdq3 := diff(T, q3):
dTdu3 := diff(T, u3):
dtdTdu3 := &dt(dTdu3):
eq3 := dtdTdu3-dTdq3-Q3 = 0:
eq3 := subs(q1t = u1, q2t = u2, q3t = u3, q4t = u4, eq3):

dTdq4 := diff(T, q4):
dTdu4 := diff(T, u4):
dtdTdu4 := &dt(dTdu4):
eq4 := dtdTdu4-dTdq4-Q4 = 0:
eq4 := subs(q1t = u1, q2t = u2, q3t = u3, q4t = u4, eq4):

kde := {q1t = u1, q2t = u2, q3t = u3, q4t = u4}:

eq1 := {u1t = solve(eq1, u1t)}:
eq2 := {u2t = solve(eq2, u2t)}:
eq3 := {u3t = solve(eq3, u3t)}:
eq4 := {u4t = solve(eq4, u4t)}:

eqs := eq1 union eq2 union eq3 union eq4 union kde:

# -----The initial conditions-----
Initcond := {q1(0) = -Pi/4,
            q2(0) = 0,
            q3(0) = 0,
            q4(0) = 0,

```

APPENDIX A. MAPLE AND SOPHIA CODE

```

u1(0) = 0,
u2(0) = 0,
u3(0) = 0,
u4(0) = 0}:

eqst := subs(toTimeFunction, eqs):

# -----The constants-----
param := {l0 = 3,
           l1 = 7,
           l2 = 3,
           m1 = 1,
           m2 = 0.5,
           k1 = 50,
           k2 = 50,
           C1 = 10,
           C2 = 10,
           g = 9.82}:

eqst := subs(param, eqst):

# -----Numerically solve the system-----
ff := dsolve(eqst union Initcond, {q1(t),
                                      q2(t),
                                      q3(t),
                                      q4(t),
                                      u1(t),
                                      u2(t),
                                      u3(t),
                                      u4(t)}, type = numeric, maxfun = 0):

numberOfPoints:= 300:
integrationTime:= 10:
_plot := odeplot(ff,
                  [[t,q1(t)],
                   [t,q2(t)],
                   [t,q3(t)],
                   [t,q4(t)],
                   [t,u1(t)],
                   [t,u2(t)],
                   [t,u3(t)],
                   [t,u4(t)]],0..integrationTime,numpoints=numberOfPoints):

# -----Exporting the data to a .txt file-----
# The code below simply exports the data in _plot
# in a text file name 'maple_data.txt' with the
# following format (columns are separated by space):
# | t    | q_1 | ... | q_n | u_1 | ... | u_n |
# |-----+----+----+----+----+----+----|
# | t_0 |     |     |     |     |     |     |
# | t_1 |     |     |     |     |     |     |
# | .   |     |     |     |     |     |     |
# | .   |     |     |     |     |     |     |
# | t_k |     |     |     |     |     |     |

t_data := ((plottools[getdata](_plot, "points")[1])[3])[[1..numberOfPoints],1]:
q1_data := ((plottools[getdata](_plot, "points")[1])[3])[[1..numberOfPoints],2]:
q2_data := ((plottools[getdata](_plot, "points")[2])[3])[[1..numberOfPoints],2]:

```

APPENDIX A. MAPLE AND SOPHIA CODE

```
q3_data := ((plottools[getdata](_plot, "points")[3])[3])[[1..numberOfPoints],2]:  
q4_data := ((plottools[getdata](_plot, "points")[4])[3])[[1..numberOfPoints],2]:  
u1_data := ((plottools[getdata](_plot, "points")[5])[3])[[1..numberOfPoints],2]:  
u2_data := ((plottools[getdata](_plot, "points")[6])[3])[[1..numberOfPoints],2]:  
u3_data := ((plottools[getdata](_plot, "points")[7])[3])[[1..numberOfPoints],2]:  
u4_data := ((plottools[getdata](_plot, "points")[8])[3])[[1..numberOfPoints],2]:  
  
_data := Concatenate(2,  
                     t_data,  
                     q1_data,  
                     q2_data,  
                     q3_data,  
                     q4_data,  
                     u1_data,  
                     u2_data,  
                     u3_data,  
                     u4_data):  
  
writedata("/path/to/maple_data.txt",convert(_data,array),float):
```

Appendix B

Blender Code

We also include the Blender code used to animate the two systems in this report. However, for practical reasons we chose not to include the code in the document but it can be found at

<https://github.com/filipstrand/lagrangian-mechanics>