DET NATUR- OG
BIOVIDENSKABELIGE
FAKULTET
KØBENHAVNS UNIVERSITET

University of Copenhagen

First year project

Project number: 2016-20

Classical analysis of chaotic pendulum

Christian Schiøtt, ID: BCN852

Rasmus Nielsen, ID: JBZ701

Thue Nikolajsen, ID: QRD689

supervised by

Professor Peter Henrik Hansen

Email: phansen@nbi.ku.dk

Number of pages: 15 Number of pages in appendix: X

The report was sent as a pdf document on Saturday 28th April, 2018

Abstract

In this project a chaotic double pendulum system is analysed. First, the *equations of motion* for the system is found and numerically integrated to conclude chaotic behaviour. An experiment is then conducted, finding that the lab system exhibited chaotic behaviour. Furthermore, a linear model for small oscillations of the system is developed, and theoretical values for the *eigenmode frequencies* produced. An experiment is then conducted to find these frequencies. The experiment and the theory agreed on the mean-valued frequencies $0.86\,Hz$ and $0.65\,Hz$.

Resumé

I dette projekt analyseres et kaotisk dobbelt pendulsystem. Først findes *bevægelsesligningerne* for vores system og integrerer disse numerisk for at påvise kaotisk opførsel. Dernæst udføres et eksperiment som påviste kaotisk opførsel i et virkeligt dobbelt pendulsystem. Derudover udvikles en lineær model for små udsving af systemet, og teoretiske værdier for *eigensvingsnings-frekvenserne* bestemmes. Dernæst udføres et eksperiment for også at finde disse frekvenser empirisk. Eksperimentet og teorien gav begge middelfrekvenserne $0.86\,Hz$ og $0.65\,Hz$.

Contents

Introduction

In this project, we will analyze the movement of a chaotic double pendulum system. The double pendulum system in question consists of two connected pendulums; an upper pendulum and a lower pendulum. The upper pendulum rotates around a fixed pivot point, while the lower pendulum rotates around the end point of the upper pendulum. Even though this system in entirely classical, it is very difficult to predict the motion of the system. The reason for this can be explained by the concept of *chaos*.

Determinism and chaos

All classical systems are what is called *deterministic*. This means that if you know the *state* of the system (the values of the degrees of freedom) at one time, you can predict the motion of the system, both forwards and backwards in time. You do this by finding the Equations which governs the time evolution of the system. This is in theory true for all classical systems. However, some *nonlinear systems* are so sensitive to initial conditions that it becomes practically impossible to measure these initial conditions precisely enough to make long term predictions. This is what is call chaotic behaviour, and the double pendulum system exhibits such behaviour.

The focus of the project

We have chosen to focus our analysis on to major things. Those things are

1. *showing that the double pendulum behaves chaotic.*
2. *finding a linear approximation to the system for small oscillations.*

This allows us to show both qualitative and quantitative behaviour of the chaotic double pendulum system. Furthermore, knowing how the system behaves both in the linear and nonlinear regime gives us a good understanding of the overall behaviour of the system.

The theoretical analysis

In this chapter we are going to find the equations governing the time evolution of our chaotic pendulum system. Once we have the equations we seek, we will find a linear approximation to the system, and analyse how it behaves when the oscillations of the pendulums are small.

Finding the equations of motion

In order to properly analyse the behaviour of our chaotic pendulum system, we need to know how it changes with time. In other words; we need to find the *equations of motion (EOMs)* describing the system. There are several different ways to go about finding them, but the approach we took was that of *Lagrangian Mechanics*. The reason being, that Lagrangian Mechanics gives a step by step recipe to finding the the equations of motion. Furthermore, with Lagrange Mechanics, we can find the EOMs knowing only the total kinetic and potential energy of the system [1].
The procedure is as follow (here $\dot{x}$ means the derivative of x with respect to time):

1. *Find a set of degrees of freedom $q_1, ..., q_n$ sufficient to describe the system.*

2. *Find the Lagrange of the system, $L = T - U$ where T is the total kinetic energy of the system and U is the total potential energy of the system.*

3. *Find the conjugate momenta p_i to all q_i defined as follows:*

$$p_i \equiv \frac{\partial L}{\partial \dot{q}_i} \tag{2.1}$$

4. *Use all the equations (1.1) to solve for all the $\dot{q}_i$ i.e. find the EOM's for all q_i.*

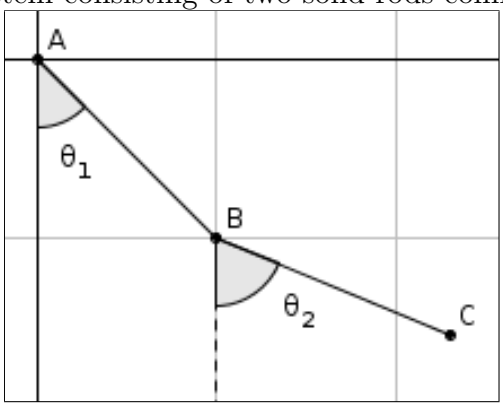
$$\dot{q}_i = f_i(q_1, ..., q_n, p_1, ..., p_n) \tag{2.2}$$

5. *Now find the EOM's for all p_i using the Euler-Lagrange Equation.*

$$\dot{q}_i = \frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} = \frac{\partial L}{\partial q_i} \tag{2.3}$$

According to step 1 we have to find a sufficient set of degrees of freedom (also called coordinates) to describe our system. As our system consists of two connected pendulums, it seems natural to use angles with respect to vertical(one for each pendulum) as our coordinates. We have chosen our coordinate θ_1 and θ_2 as seen on the figure bellow.

Figure 2.1: Our system consisting of two solid rods connected to each other.



Now to the second step; finding the Lagrange of our system. To do this, we first need to find T and U. Lets start by finding T, the total kinetic energy. To do this we recall that the total kinetic energy of an object, which is both translating and rotating about its center of mass (COM), is

$$T = \frac{1}{2}mv_{cm}^2 + \frac{1}{2}I\dot{\theta}^2$$

where m is the mass of the object, v_{cm} is the speed of the COM, I is the moment of inertia about the COM and $\dot{\theta}$ is the angular velocity about the COM. The kinetic energy of the first pendulum rod now becomes easy to find. Since the rod is moving in a circle about A, $v_{cm} = L_1\dot{\theta_1}$, where L_1 is the length from A to the COM of the pendulum rod. Furthermore, the rod is also rotating about its COM with an angular velocity $\dot{\theta_1}$. The kinetic energy T_1 of the first pendulum rod can thus be written as

$$T_1 = \frac{1}{2}m_1(L_1\dot{\theta_1})^2 + \frac{1}{2}I_1\dot{\theta_1}^2 = \frac{1}{2}(m_1L_1^2 + I_1)\dot{\theta_1}^2 \tag{2.4}$$

Now to the kinetic energy of the second pendulum rod. Since the COM of this pendulum rod is moving in a more complicated way, we need to do something else to find its speed. The trick is to write the position of the COM in Cartesian coordinates.

$$X = R_1\sin(\theta_1) + L_2\sin(\theta_2) \quad , \quad Y = R_1\cos(\theta_1) + L_2\cos(\theta_2)$$

Where L_2 is the length from B to the COM of the second rod. We now differentiate the above equations with respect to time.

$$\dot{X} = R_1\cos(\theta_1)\dot{\theta_1} + L_2\cos(\theta_2)\dot{\theta_2} \quad , \quad \dot{Y} = -R_1\sin(\theta_1)\dot{\theta_1} - L_2\sin(\theta_2)\dot{\theta_2}$$

We now recall that in Cartesian coordinates, $v_{cm}^2 = \dot{X}^2 + \dot{Y}^2$, which means that we can now find an expression from v_{cm}^2. Using appropriate trigonometric identities, we find that

$$v_{cm}^2 = (R_1\dot{\theta_1})^2 + (L_2\dot{\theta_2})^2 + 2R_1L_2\dot{\theta_1}\dot{\theta_2}\cos(\Delta\theta)$$

Where $\Delta\theta = \theta_2 - \theta_1$ Since we know that the second pendulum rod is rotating about its COM with an angular velocity $\dot{\theta_1} + \dot{\theta_2}$ we can now write the total kinetic energy of the second pendulum rod T_2 as

$$T_2 = \frac{1}{2}m_2R_1^2\dot{\theta_1}^2 + \frac{1}{2}(m_2L_2^2 + I_2)\dot{\theta_2}^2 + m_2R_1L_2\dot{\theta_1}\dot{\theta_2}\cos(\Delta\theta) \tag{2.5}$$

Now that we how both T_1 and T_2 we can write an expression for the total kinetic energy T of the system.

$$T = \frac{1}{2}(m_1L_1^2 + m_2R_1^2 + I_1)\dot{\theta_1}^2 + \frac{1}{2}(m_2L_2^2 + I_2)\dot{\theta_2}^2 + m_2R_1L_2\dot{\theta_1}\dot{\theta_2}\cos(\Delta\theta)$$

Lets define some constant, so that we can write T in a more compact form. Let

$$\alpha = m_1L_1^2 + m_2R_1^2 + I_1 \quad , \quad \beta = m_2L_2^2 + I_2 \quad , \quad \gamma = m_2R_1L_2 \tag{2.6}$$

We can then write T as follow.

$$T = \frac{1}{2}\alpha\dot{\theta_1}^2 + \frac{1}{2}\beta\dot{\theta_2}^2 + \gamma\cos(\Delta\theta)\dot{\theta_1}\dot{\theta_2} \tag{2.7}$$

Now that we finally have T we can move on to finding U. The only potential energy in our system is the gravitational potential which is given by $U_g = mgh$. Here m is the mass of the object, g is the local gravity constant on Earth and h is the height of the COM of the object above $U_g = 0$. If we chose $U_g = 0$ to be at the height of A (see figure 2.1) we can write the potential energy of the two pendulum rods as follow.

$$U_1 = -m_1gL_1\cos(\theta_1) \quad , \quad U_2 = -m_2g(R_1\cos(\theta_1) + L_2\cos(\theta_2)) \tag{2.8}$$

Where U_1 and U_2 are the potential energies of the first and second pendulum rod respectively, and L_1 is the length from A to the COM of the first pendulum rod. We can now write the total potential energy U of our system as follow.

$$U = -g[(m_1L_1 + m_2R_1)\cos(\theta_1) + m_2L_2\cos(\theta_2)]$$

Lets define some new constants, so that we can write U in a more compact form. Let

$$\delta = g(m_1L_1 + m_2R_1) \quad , \quad \epsilon = gm_2L_2 \tag{2.9}$$

We can now write U as follow.

$$U = -[\delta\cos(\theta_1) + \epsilon\cos(\theta_2)] \tag{2.10}$$

Now we can finally write the Lagrange L of our system. It is

$$L = \frac{1}{2}\alpha\dot{\theta_1}^2 + \frac{1}{2}\beta\dot{\theta_2}^2 + \gamma\cos(\Delta\theta)\dot{\theta_1}\dot{\theta_2} + [\delta\cos(\theta_1) + \epsilon\cos(\theta_2)] \tag{2.11}$$

Now we can move on to step 3; finding all the conjugate momenta p_i. In our case there are two, and we just find them by differentiating L with respect to $\dot{\theta_1}$ and $\dot{\theta_2}$.

$$p_1 = \frac{\partial L}{\partial\dot{\theta_1}} = \alpha\dot{\theta_1} + \gamma\cos(\Delta\theta)\dot{\theta_2} \tag{2.12}$$

$$p_2 = \frac{\partial L}{\partial\dot{\theta_2}} = \beta\dot{\theta_2} + \gamma\cos(\Delta\theta)\dot{\theta_1} \tag{2.13}$$

Now we move on to step 4; finding the EOMs for all q_i. This means that we need to find $\dot{\theta_1}$ and $\dot{\theta_2}$ in terms of p_1 and p_2. To do so, we solve the linear system of equations consisting of equations (2.21) and (2.22). The solutions for $\dot{\theta_1}$ and $\dot{\theta_2}$ are.

$$\dot{\theta_1} = \frac{p_1\beta - p_2\gamma\cos(\Delta\theta)}{\alpha\beta - \gamma^2\cos(\Delta\theta)^2} \tag{2.14}$$

$$\dot{\theta_2} = \frac{p_2\alpha - p_1\gamma\cos(\Delta\theta)}{\alpha\beta - \gamma^2\cos(\Delta\theta)^2} \tag{2.15}$$

At long last we have reached step 5; finding the EOMs for all p_i. To do this we just use the Euler-Lagrange equation.

$$\dot{p_1} = \frac{\partial L}{\partial\theta_1} = \gamma\sin(\Delta\theta)\dot{\theta_1}\dot{\theta_2} - \delta\sin(\theta_1) \tag{2.16}$$

$$\dot{p_2} = \frac{\partial L}{\partial\theta_2} = -\gamma\sin(\Delta\theta)\dot{\theta_1}\dot{\theta_2} - \epsilon\sin(\theta_2) \tag{2.17}$$

Now that we have the EOMs of our system, we are ready to move on with the analysis of our chaotic pendulum system.

Finding a linear approximation

In the previous section we find the EOMs of our chaotic pendulum system. Lets take a closer look at those equations. The first thing we notice is that the equations are not linear. They are so called *nonlinear ordinary differential equations*. This usually suggests that we **can not** find analytic solutions to the equations. However, one thing we can do is *linearize* the nonlinear system in the vicinity of a *fixed point* [2]. First, what is a fixed point? A fixed point is a set of values

$$\underline{X}^* = (q_1^*, ..., q_n^*, p_1^*, ..., p_n^*) \quad \text{such that} \quad \dot{q}_i = 0, \quad \dot{p}_i = 0 \quad \forall i$$

In other word, the fixed points are the configurations of the system which does not change over time. Second, what does it mean to linearize the nonlinear system in the vicinity of a fixed point? To answer that, we start by writing the EOMs of the system as a vector differential equation of the form.

$$\dot{\underline{X}} = f(\underline{X}) \tag{2.18}$$

Equation (2.19) can also be expressed in component form.

$$\begin{pmatrix} \dot{x_1} \\ \vdots \\ \dot{x_n} \end{pmatrix} = \begin{pmatrix} f_1(x_1, ..., x_n) \\ \vdots \\ f_n(x_1, ..., x_n) \end{pmatrix}$$

Here $f_1, ..., f_n$ are the EOMs of the system. Now, what happens if we make a small perturbation $\underline{\eta}$ away from a fixed point, in such a way that

$$\underline{X} = \underline{X}^* + \underline{\eta} \tag{2.19}$$

First we notice that $\dot{\underline{X}} = \dot{\underline{\eta}}$ since $\underline{X}^*$ is constant. This means that the EOMs for $\underline{\eta}$ and $\underline{X}$ are the same. Let us now look at the EOM for η_i and x_i.

$$\dot{\eta_i} = \dot{x_i} = f_i(x_1, ..., x_n) = f_i(x_1^* + \eta_1, ..., x_n^* + \eta_n) \tag{2.20}$$

Using multi variable Taylor series expansion about $\underline{X}^*$, we find that

$$\dot{\eta_i} = \left.\frac{\partial f_i}{\partial x_1}\right|_{\underline{X}^*} \eta_i + ... + \left.\frac{\partial f_i}{\partial x_n}\right|_{\underline{X}^*} \eta_n + \text{ Higher Order Terms}$$

If we neglect the small higher order terms, we now have a system of linear differential equations which tells us how $\underline{\eta}$ (and by extension $\underline{X}$) changes with time near $\underline{X}^*$. This is what it means to linearize in the vicinity of a fixed point. We can write the system of equations in matrix form as shown bellow.

$$\dot{\underline{\eta}} = \underline{\underline{J}}\,\underline{\eta} \tag{2.21}$$

The matrix $\underline{\underline{J}}$ is called the *Jacobian matrix*, and it's given by

$$\underline{\underline{J}} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}_{\underline{X}^*}$$

Now we have the tools to study the behaviour of our chaotic pendulum system near any fixed point. We choose to focus on the one stable fixed point of our system (stable meaning that perturbations do not grow exponentially at that fixed point). That is the point

$$\underline{X}_0^* = (\theta_1, \theta_2, p_1, p_2) = (0, 0, 0, 0)$$

This is just the point at which the two pendulum rods are hanging completely still and pointing straight down. Now we can try to find the solutions to equation (2.21) for our particular $\underline{\underline{J}}$ matrix. Since we are dealing with a system of first-order equations, lets test if $\underline{\eta} = \underline{v}e^{\lambda t}$ is a solution to (2.21).

$$\dot{\underline{\eta}} = \underline{\underline{J}}\,\underline{\eta} \quad \Rightarrow \quad \lambda\underline{v}e^{\lambda t} = \underline{\underline{J}}\,\underline{v}e^{\lambda t} \quad , \quad e^{\lambda t} \neq 0 \quad \forall t \quad \Rightarrow \quad \lambda\underline{v} = \underline{\underline{J}}\,\underline{v} \tag{2.22}$$

Hence $\underline{\eta} = \underline{v}e^{\lambda t}$ is a solution to (2.21) if $\underline{v}$ is an eigenvector of $\underline{\underline{J}}$ with the corresponding eigenvalue λ. Since (2.21) is a linear equation, we know that linear combinations of solutions are also solutions. This means that

$$\underline{\eta} = \sum_{i=1}^{n} c_i \underline{v}_i e^{\lambda_i t} \tag{2.23}$$

is a solution. It turns out that this is also *the complete solution* to equation (2.21). Now that we can solve the linear system of equations, we can find out how any small perturbation η away from $\underline{X}_0^*$ evolves with time. We will now look further into some very specific perturbations; *the eigenmodes of our system.*

Eigenmodes and their periods

Now that we know how to linearize our system about its stable fixed point, we can begin to look further into the behaviour of the system. Lets start by looking at the complete solution to a linearized system we found in the last section. If we choose our initial condition to be $\underline{\eta}(0) = c_i\,\underline{v}_i$ we get the specific solution

$$\underline{\eta}_i(t) = c_i\,\underline{v}_i e^{\lambda_i t} \quad , \quad c_i \in \mathbb{C} \tag{2.24}$$

This mean, that if we start the system at any initial condition $\underline{\eta}_0$ for which $\underline{\eta}_0 \in \text{span}(\underline{v}_i)$ then $\underline{\eta}(t) \in \text{span}(\underline{v}_i)\ \forall t$. The set of solutions for which $\underline{\eta}(t) \in \text{span}(\underline{v}_i)\ \forall t$, we will call the *eigenmode* corresponding to $\underline{v}_i$. Because all of the eigenvectors are linearly independent, we know that $\text{span}(\underline{v}_i) \cap \text{span}(\underline{v}_j) = \{\underline{0}\}\ \ \forall i,\, j$. Therefore solutions from different eigenmodes must be linearly independent. Now, lets look at the eigenmodes for a special case of system. These systems all have eigenvectors which comes in pairs of $\underline{a}_j \pm i\underline{b}_j$ with corresponding eigenvalues $\lambda_j = \pm i\omega_j$. Using Euler's formula: $e^{i\theta} = \cos(\theta) + i\ \sin(\theta)$, we can rewrite the *real part* of (2.24) as follow.

$$\underline{\eta} = \sum_{i=j}^{n} r_j^1\ [\underline{a}_j \cos(\omega_j t) - \underline{b}_j \sin(\omega_j t)] + r_j^2\ [\underline{a}_j \cos(\omega_j t) + \underline{b}_j \sin(\omega_j t)] \tag{2.25}$$

Where $r_j^1, r_j^2 \in \mathbb{R}$. Like the general solution to linear systems, the above equation is a sum of solutions from different eigenmodes. For these particular systems, we can pair up eigenmode solutions with corrresponding eigenvalues $\pm i\omega_j$ to make *harmonic oscillations* with the *frequencies* given by

$$f_j = \frac{\omega_j}{2\pi} \tag{2.26}$$

Equations (2.25) and (2.26) are important to us because they give us a way to test the predictions of our linear model (e.i. the eigenmode frequencies f_i). To do this, we would of cause need to measure the eigenmode frequencies experimentally. One could imagine doing this by first filming small oscillations of a chaotic pendulum. One could then use software to track the motion of the pendulum and use *Fourier-analysis* to decompose its trajectory and find the eigenmode frequencies. We will be doing this in the following chapters.

Simulation of the system

In this chapter we will try to simulate the movements of the double pendulum. All simulations are done assuming that the two pendulums are uniform rods with a mass of 1 kg and a length of 2 m. Using these properties, the minimum total energy of the system is -39.3 J.

Finding a method for simulating the system

Using the equations of motion we just derived, we simulate the motion of the pendulums in MatLab using the standard Euler method [2], so that:

$$\theta_{1,n+1} = \theta_{1,n} + \dot{\theta}_1\, dt \quad , \quad \theta_{2,n+1} = \theta_{2,n} + \dot{\theta}_2\, dt$$

$$p_{1,n+1} = p_{1,n} + \dot{p}_1\, dt \quad , \quad p_{2,n+1} = p_{2,n} + \dot{p}_2\, dt$$

Where dt is a small increase in time. We set the value of dt to $0.01\,s$. This approach seems to result in unexpected behaviour. To check if something is wrong with the simulator, we plot the total energy of the system as a function of time. Since we do not take any non-conservative forces into account, we expect the total energy to remain unchanged.

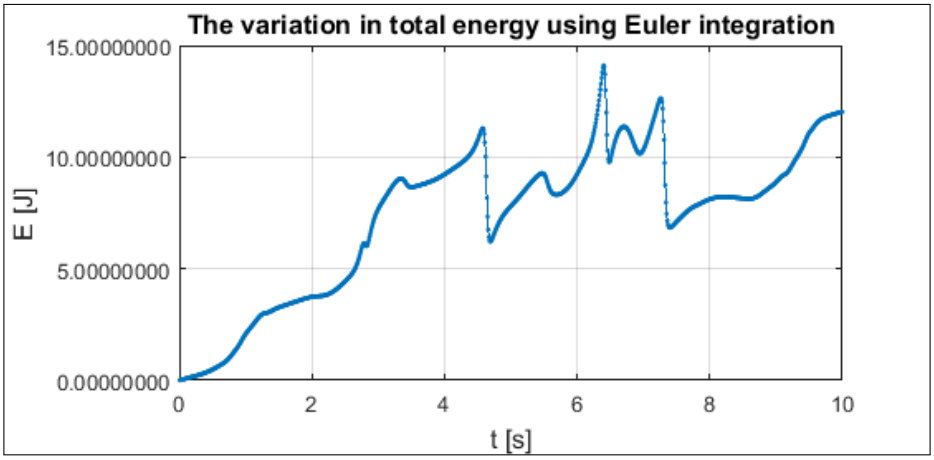


Figure 3.1: The variation in energy over time using Euler integration with the following initial conditions: $\theta_1 = \frac{\pi}{2}$, $\theta_2 = \frac{\pi}{2}$, $p_1 = 0$, $p_2 = 0$.

We see that the total energy of the system does in fact decrease significantly over time, which can only be explained by either a mistake in our equations or our program, or a general fault in our method. One such general fault could be that Euler integration is not good enough if we use a dt of $0.01\,s$. Choosing a smaller dt is not an option, as that would cause our simulation to run too slowly. This means that we have to look to other methods than Euler integration. One such method is the Runge-Kutta Method (RK4) for numerical integration of a set of ordinary differential equations [2]. This method uses a weighted average of four increments for each step in the following way:

$$f(\vec{x}_n) = \begin{pmatrix} f_1(\vec{x}_n) \\ f_2(\vec{x}_n) \\ f_3(\vec{x}_n) \\ f_4(\vec{x}_n) \end{pmatrix}$$

$$\vec{a} = f(\vec{x}_0) \quad , \quad \vec{b} = f\left(\vec{x}_0 + \vec{a}\,\frac{dt}{2}\right) \quad , \quad \vec{c} = f\left(\vec{x}_0 + \vec{b}\,\frac{dt}{2}\right) \quad , \quad \vec{d} = f(\vec{x}_0 + \vec{c}\,dt)$$

$$x_{n+1} = x_n + \left(\vec{a} + 2\vec{b} + 2\vec{c} + \vec{d}\right)\frac{dt}{6}$$

Where f_1, f_2, f_3 and f_4 are the equations of motion (2.15), (2.16), (2.17) and (2.18). We test this method in the same way as we tested Euler integration, by looking at the change in the total energy over time:

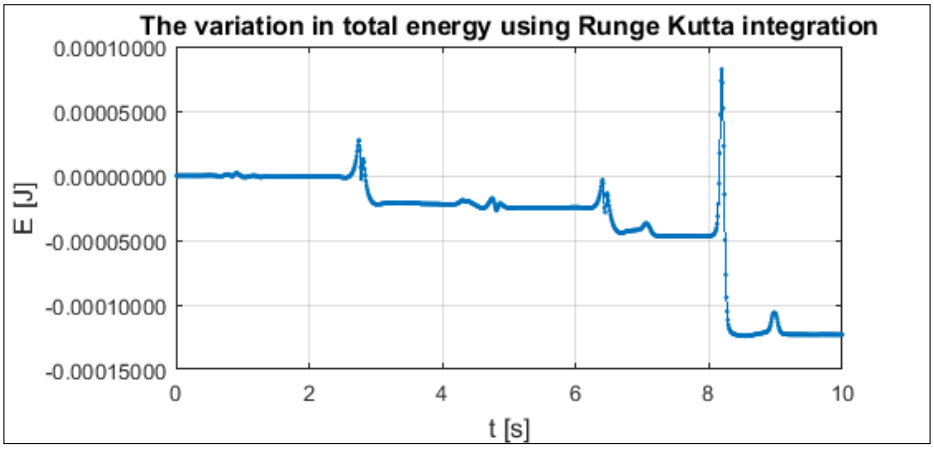


Figure 3.2: The variation in energy over time using RK4 with the following initial conditions: $\theta_1 = \frac{\pi}{2} \quad , \quad \theta_2 = \frac{\pi}{2} \quad , \quad p_1 = 0 \quad , \quad p_2 = 0$.

Since the change in the total energy of the system is in order of magnitude 10^{-5} and the minimum total energy of the system is in order of magnitude 10, we can accept the RK4 method and assume that the total energy of the system will not change significantly.

Simulating the chaotic system

To show chaotic behaviour, we run the simulation twice with slightly different initial conditions. If the system is in fact chaotic, we expect slightly different initial conditions to result in trajectories, which suddenly begin to deviate greatly from one another at a certain point in time. We choose to look at the lower pendulum, since it moves more freely than the upper pendulum.

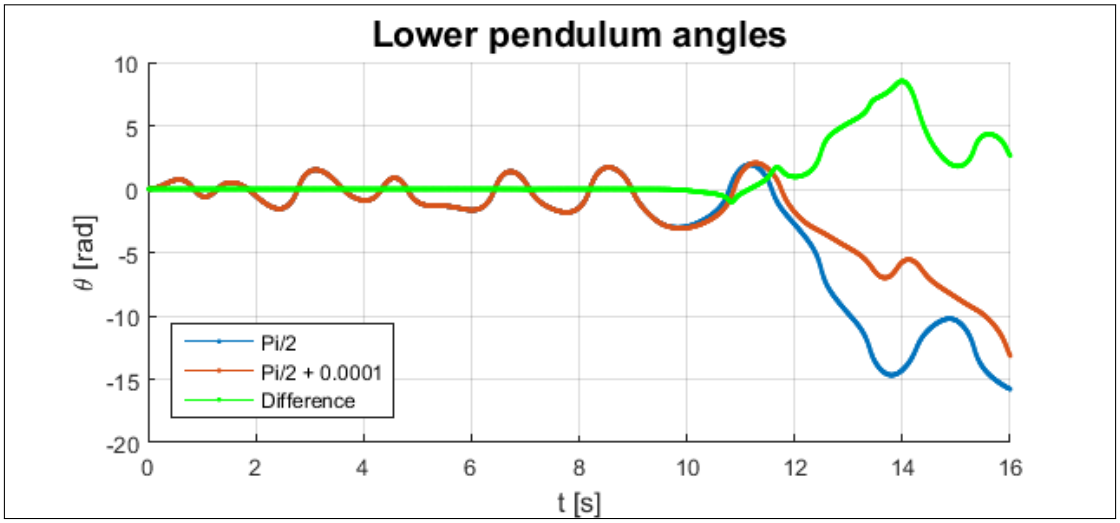


Figure 3.3: The variation in energy over time using RK4 with the following initial conditions: $\theta_1 = 2 \quad , \quad \theta_2 = 2 \quad , \quad p_1 = 0 \quad , \quad p_2 = 0$.

We see that the two experiments do in fact have trajectories, which suddenly begin to deviate greatly from one another at a certain point in time, and thus we can conclude that the system is chaotic.

The experimental aspect

In the interest of coupling theory with reality, two experiments are carried out in the lab on an actual double pendulum. Firstly, we wish to document chaotic behaviour in the pendulum, and secondly, we wish to numerically find the eigenmodes from experiments within the linear regime, and compare them to the theoretical values.

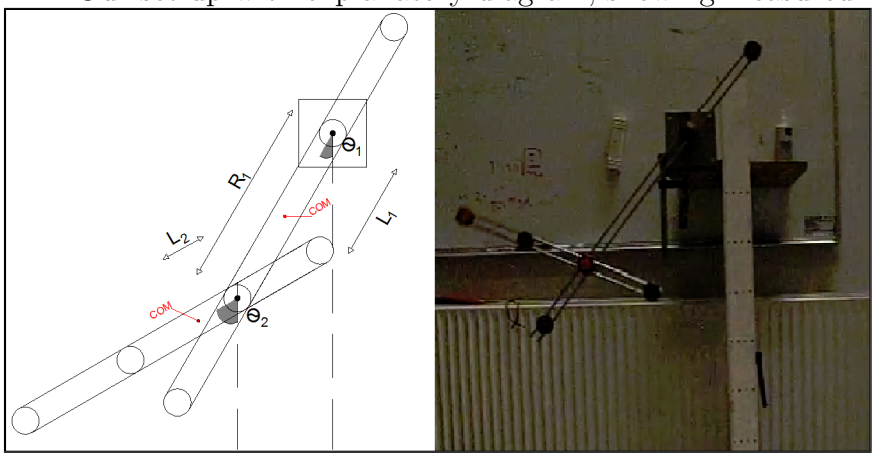
Set-up

A double pendulum, as seen on Fig. 4.1, is mounted on a free axis roughly 1 meter from a high-speed camera. Filming in 300 frames per second, it allows for clear images in 568x433 resolution, even though only every 10th frame is used in the analysis to reduce computing time. In order to gain simple X/Y-values and angular data from the video, the freeware program Tracker is used to pinpoint the coordinates of the end of the lower pendulum rod, with an uncertainty of $1\,mm$.

Masses and lengths of the rods are measured directly. The moments of inertia are deduced from the period of each pendulum when separated from the other. To do that, we use the pendulum period formula for small oscillations and the parallel axes theorem.

$$T = 2\pi\sqrt{\frac{mgL}{I_{end}}} \quad , \quad I_{end} = I_{COM} + m\,L^2 \quad \Rightarrow \quad I_{COM} = \frac{T^2}{4\pi^2}\,m\,g\,L - m\,g\,L^2 \tag{4.1}$$

Figure 4.1: Our set-up with explanatory diagram, showing measured lengths.



Where m is the mass of the respective rods, T is the period for small oscillations, I_{end} is the moments of inertia about the pivot points for each rod, I_{COM} is the moments of inertia about the COM. The COM of each rod is in turn found by balancing it on a taut string

	Upper pendulum	*Lower pendulum*
m	$(888.3 \pm 0.1)g$	$(630.6 \pm 0.1)g$
T	$(1.508 \pm 16 \cdot 10^{-4})s$	$(1.395 \cdot 18 \cdot 10^{-4})s$
I_{COM}	$(0.0576 \pm 4 \cdot 10^{-4})kg \cdot m^2$	$(0.0183 \pm 4 \cdot 10^{-4})kg \cdot m^2$
L	$(16.0 \pm 0.1)cm$	$(7.0 \pm 0.1)cm$

Furthermore, the distance between the pivot points R_1 is measured as $R_1 = (35,6 \pm 0.1)\,cm$.

Insanity experiment

To confirm chaotic behaviour in the pendulum, two iterations of the same experiment are carried out[1] with the pendulums in the state of $(-\frac{\pi}{2}, \frac{\pi}{2}, 0, 0)$, and letting the system go. This, of course, is with the expectation that we will be given entirely different results.

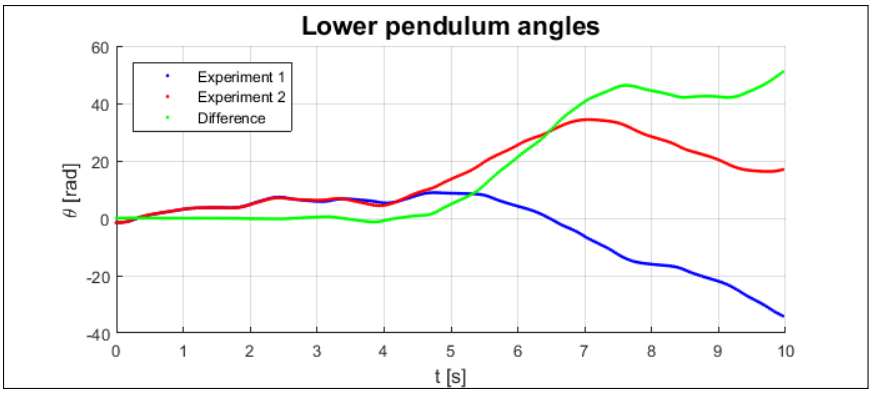


Figure 4.2: The plot of each experiments' θ_2 and the difference between them vs. time.

[1] Video documentation available at bit.ly/1UalYi4 and bit.ly/1UalYi4

Using video tracking, plots of θ_2 as a function of time are produced as seen on Fig. 4.2. Making sure the paths are started from the same point, the difference of the two functions are also seen. Clearly they are similar in the beginning, but the difference rises suddenly and quickly around 5 seconds into the experiment.

Finding the eigenmode frequencies

The theoretically derived eigenmode frequencies, cf. the earlier chapter, is given by looking at the Jacobian matrix for the fixed point $\underline{X}_0^*$. For our system, that will be a 4×4 matrix. Computing all partial derivatives and evaluating them at $\underline{X}_0^*$ we get the following matrix:

$$\underline{\underline{J}} = \begin{pmatrix} 0 & 0 & \beta' & -\gamma' \\ 0 & 0 & -\gamma' & \alpha' \\ -\delta & 0 & 0 & 0 \\ 0 & -\epsilon & 0 & 0 \end{pmatrix}_{\underline{X}^*}, \quad \alpha' = \frac{\alpha}{\alpha\beta - \gamma^2}, \quad \beta' = \frac{\beta}{\alpha\beta - \gamma^2}, \quad \gamma' = \frac{\gamma}{\alpha\beta - \gamma^2} \tag{4.2}$$

Where α, β, and γ are defined in chapter 2 section 1. Using the definition of the constants, we now construct a *Matlab-script* to find the eigenvalues of the Jacobian matrix for our system. The eigenmode frequencies are then found using the equations $\lambda_j = \pm i\omega_j$ and $f_j = \frac{\omega_j}{2\pi}$. The uncertainty of these eigenvalues are difficult to ascertain analytically, and so they are found numerically. We simply computed the eigenvalues 10^6 times, using slightly different values of measured quantities, taken from a norm-dist defined by their uncertainties. The resulting outputs are used to form a new norm-dist, and their standard deviations is taken to be the uncertainties. The theoretical eigenmode frequencies are then:

$$f_1 = (0.651 \pm 0.001)\, Hz \qquad f_2 = (0.862 \pm 0.002)\, Hz \tag{4.3}$$

An experiment well within the linear regime (an angle certainly below 5 degrees), is carried out[2] and the data analyzed in Tracker. Through its built-in Fourier Analysis tool, the eigenmode frequencies are found to be:

$$f_1 = (0.66 \pm 0.05)\, Hz \qquad f_2 = (0.85 \pm 0.05)\, Hz \tag{4.4}$$

We now see, that both theoretical intervals lie within the experimental intervals.

[2]Video documentation available at bit.ly/1R2GKx6

Conclusion

We have successfully derived the equations of motion for the double pendulum system. Using these equations and the Runge-Kutta method for numerical integration, we have created a working simulation in MatLab. By starting the simulation with two slightly different sets of initial conditions, we have shown chaotic behaviour. This chaotic behaviour has also been demonstrated in the lab. We have found a linear approximation to the system for small angles and found the frequencies of the eigenmodes both theoretically and experimentally, and we have shown that the experimental values do agree with our expectations from the theoretical calculations.

Bibliography

[1] Alexander L. Fetter, John Dirk Walecka. *Theoretical Mechanics of Particles and Continua*. ISBN: 0-486-43261-0

[2] Steven H. Strogatz. *Nonlinear Dynamics and Chaos*. ISBN: 0-201-54344-3

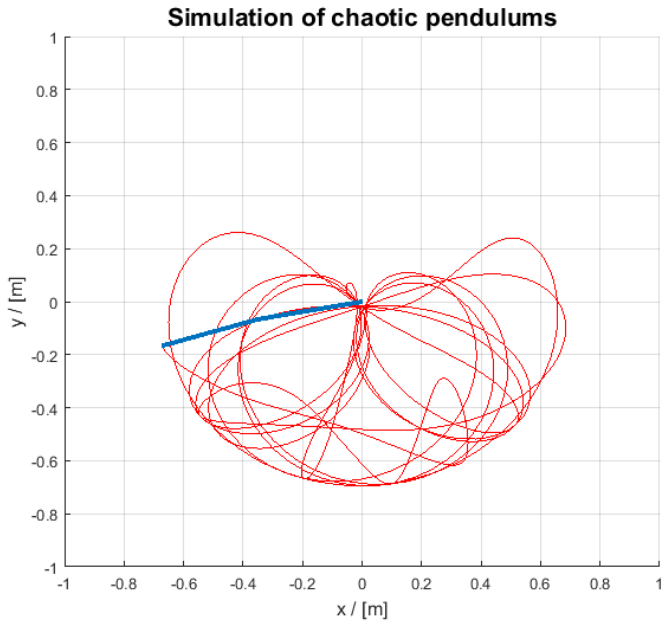
Simulation x,y-plot

```
% First year project "The Double Pendulum"
% Authors:
% Christian Schioett - BCN852
% Rasmus Nielsen - JBZ701
% Thue Nikolajsen - QRD689
% Date 18/03 2016

close all; clear all; clc;
% This is the beginning of the program

% Variables get set
g = 9.82;       % local gravity constant
dt = 0.01;      % time step size

m = [888.3 630.6]*10^-3;
R = [35.6 33.9]*10^-2;
L = [16.02 7.01]*10^-2;
P = [1.5068 1.3951];
I(1) = ( P(1)^2 * g/(4*pi*pi) - L(1) ) * m(1) * L(1);
I(2) = ( P(2)^2 * g/(4*pi*pi) - L(2) ) * m(2) * L(2);

% state-vector
S = [pi/2, -pi/2, 0, 0];

% Constants important for computations
k(1) = m(1)*L(1)^2 + m(2)*R(1)^2 + I(1);
```

```
k(2) = m(2)*L(2)^2 + I(2);
k(3) = m(2)*R(1)*L(2);
k(4) = g*m(1)*L(1) + g*m(2)*R(1);
k(5) = g*m(2)*L(2);

% Conversion to Cartesian coordinates
x(1) = sin(S(1))*R(1); y(1) = -cos(S(1))*R(1);
x(2) = x(1) + sin(S(2))*R(2); y(2) = y(1) - cos(S(2))*R(2);

% Setting up window
h = figure; set(gca,'Color','black'); set(gca,'fontsize',10);
hold on; grid on; set(gcf,'Position',[0 0 650 550])
xlim([-1; 1]); ylim([-1; 1]);
t = title('Simulation of chaotic pendulums'); t.FontSize = 16;
lx = xlabel('x / [m]'); ly = ylabel('y / [m]');
lx.FontSize = 12; ly.FontSize = 12;

% Equations of motion (and other functions)
D = @(X) ( k(1)*k(2) - k(3)^2 * cos(X(2)-X(1))^2 );
f{1} = @(X) ( X(3)*k(2) - X(4)*k(3)*cos(X(2)-X(1)) ) / D(X);
f{2} = @(X) ( X(4)*k(1) - X(3)*k(3)*cos(X(2)-X(1)) ) / D(X);
f{3} = @(X) k(3)*sin(X(2)-X(1))*f{1}(X)*f{2}(X) - k(4)*sin(X(1));
f{4} = @(X) -k(3)*sin(X(2)-X(1))*f{1}(X)*f{2}(X) - k(5)*sin(X(2));

% Variables to track path of tip of second pendulum are set
X = []; Y = [];

% The simulation is set to begin at 0 time.
T = 0;
```

```
while T <= 10;
    % Start of Runge Kutta integration
    for i = 1:4
        a(i) = f{i}(S);
    end
    for i = 1:4
        b(i) = f{i}(S + dt/2*a);
    end
    for i = 1:4
        c(i) = f{i}(S + dt/2*b);
    end
    for i = 1:4
        d(i) = f{i}(S + dt*c);
    end

    S = S + dt/6 * (a + 2*b + 2*c + d);
    % End of Runge Kutta integration

    % Conversion to Cartesian coordinates
    x(1) = sin(S(1))*R(1); y(1) = -cos(S(1))*R(1);
    x(2) = x(1) + sin(S(2))*R(2); y(2) = y(1) - cos(S(2))*R(2);

    % Tracks the motion of the tip of the second pendulum
    X(length(X)+1) = x(2); Y(length(Y)+1) = y(2);

    % If the window is open, update graphics
    if ishandle(h)
        cla;                                          % Clear screen
        plot(X, Y, 'r-')                              % Plot path
        line([0 x(1)], [0 y(1)], 'LineWidth', 3)      % Draw pendulum 1
```

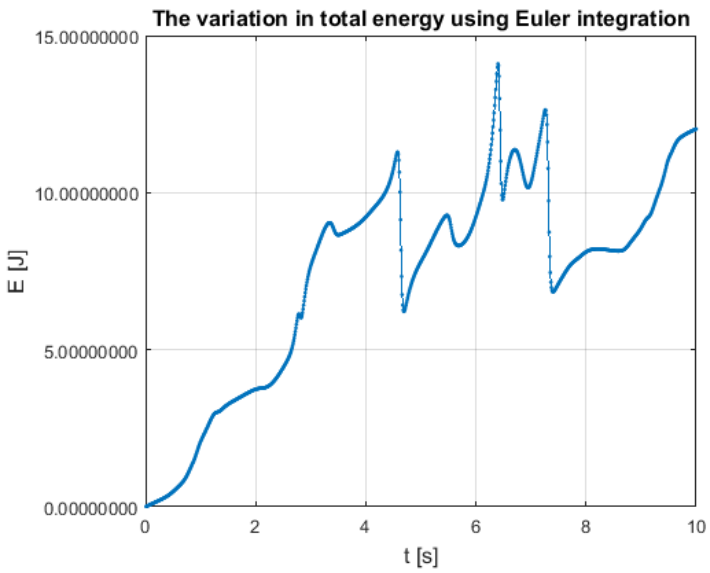
```
        line([x(1) x(2)], [y(1) y(2)], 'LineWidth', 3)  % Draw pendulum 2
    end

    % Update time the simulation has been running
    T = T + dt;

    pause(10^-6);
end
% This is the end of the program
```



Energy plot for Euler and RK4

```
% First year project "The Double Pendulum"
% Authors:
% Christian Schioett - BCN852
% Rasmus Nielsen - JBZ701
% Thue Nikolajsen - QRD689
% Date 18/03 2016

clear all; clc;
% This is the beginning of the program

% Variabler deklareres
g = 9.82;
m = [1 1];
dt = 0.01;

% Startbetingelser angives
R = [2 2];
L = R/2;
I = [1/12*m(1)*R(1)^2 1/12*m(2)*R(2)^2];

% State-vector is set
S = [pi/2 pi/2 0 0];

% Constants important for computations
k(1) = m(1)*L(1)^2 + m(2)*R(1)^2 + I(1);
k(2) = m(2)*L(2)^2 + I(2);
```

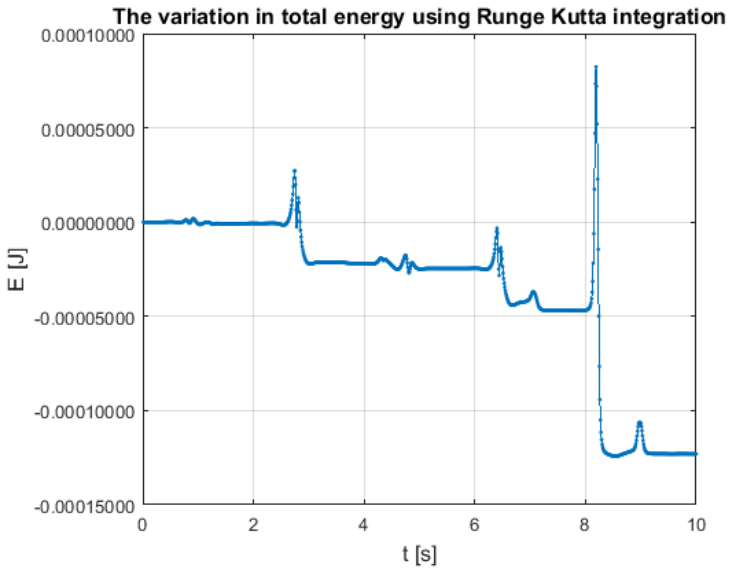
```
k(3) = m(2)*R(1)*L(2);
k(4) = g*m(1)*L(1) + g*m(2)*R(1);
k(5) = g*m(2)*L(2);

% Equations of motion (and other functions) (X = S)
D = @(X) ( k(1)*k(2) - k(3)^2 * cos(X(2)-X(1))^2 );
f{1} = @(X) ( X(3)*k(2) - X(4)*k(3)*cos(X(2)-X(1)) ) / D(X);
f{2} = @(X) ( X(4)*k(1) - X(3)*k(3)*cos(X(2)-X(1)) ) / D(X);
f{3} = @(X) k(3)*sin(X(2)-X(1))*f{1}(X)*f{2}(X) - k(4)*sin(X(1));
f{4} = @(X) -k(3)*sin(X(2)-X(1))*f{1}(X)*f{2}(X) - k(5)*sin(X(2));
H = @(X) (f{1}(X)*X(3) + f{2}(X)*X(4))/2 - k(4)*cos(X(1)) - k(5)*cos(X(2));

for i=1:1:1000          % The loop is executed if the figure window is open
    for a=1:4
        u(a) = f{a}(S);
    end
    S = S + dt * u;

    E(i) = H(S);
    T(i) = i*dt;
end

% Setting up window
figure
plot(T,E,'- .')
set(gca, 'YTickLabel', num2str(get(gca, 'YTick')', '%.8f'));
t2 = title('The variation in total energy using Euler integration'); t2.FontSize = 12;
lx = xlabel('t [s]'); ly = ylabel('E [J]');
lx.FontSize = 12; ly.FontSize = 12;
grid on
```


The variation in total energy using Euler integration
15.00000000
10.00000000
5.00000000
0.00000000
E [J]
0
2
4
6
8
10
t [s]

```
% First year project "The Double Pendulum"
% Authors:
% Christian Schioett - BCN852
% Rasmus Nielsen - JBZ701
% Thue Nikolajsen - QRD689
% Date 18/03 2016

clear all; clc; close all;
% This is the beginning of the program

% Variables get set
g = 9.82;
m = [1 1];
dt = 0.01;

% Startbetingelser angives
R = [2 2];
L = R/2;
I = [1/12*m(1)*R(1)^2 1/12*m(2)*R(2)^2];

% State-vector is set
S = [pi/2 pi/2 0 0];

% Constants important for computations
k(1) = m(1)*L(1)^2 + m(2)*R(1)^2 + I(1);
k(2) = m(2)*L(2)^2 + I(2);
k(3) = m(2)*R(1)*L(2);
k(4) = g*m(1)*L(1) + g*m(2)*R(1);
k(5) = g*m(2)*L(2);
```

```
% Equations of motion (and other functions) (X = S)
D = @(X) ( k(1)*k(2) - k(3)^2 * cos(X(2)-X(1))^2 );
f{1} = @(X) ( X(3)*k(2) - X(4)*k(3)*cos(X(2)-X(1)) ) / D(X);
f{2} = @(X) ( X(4)*k(1) - X(3)*k(3)*cos(X(2)-X(1)) ) / D(X);
f{3} = @(X) k(3)*sin(X(2)-X(1))*f{1}(X)*f{2}(X) - k(4)*sin(X(1));
f{4} = @(X) -k(3)*sin(X(2)-X(1))*f{1}(X)*f{2}(X) - k(5)*sin(X(2));
H = @(X) (f{1}(X)*X(3) + f{2}(X)*X(4))/2 - k(4)*cos(X(1)) - k(5)*cos(X(2));

for i=1:1:1000
    % Start of Runge Kutta integration
    for o = 1:4
        a(o) = f{o}(S);
    end
    for o = 1:4
        b(o) = f{o}(S + dt/2*a);
    end
    for o = 1:4
        c(o) = f{o}(S + dt/2*b);
    end
    for o = 1:4
        d(o) = f{o}(S + dt*c);
    end

    S = S + dt/6 * (a + 2*b + 2*c + d);
    % End of Runge Kutta integration

    E(i) = H(S);
    T(i) = i*dt;
end
```

```
% Setting up window
figure
plot(T,E,'- .')
set(gca, 'YTickLabel', num2str(get(gca, 'YTick')', '%.8f'));
t2 = title('The variation in total energy using Runge Kutta integration'); t2.FontSize =
lx = xlabel('t [s]'); ly = ylabel('E [J]');
lx.FontSize = 12; ly.FontSize = 12;
grid on
```

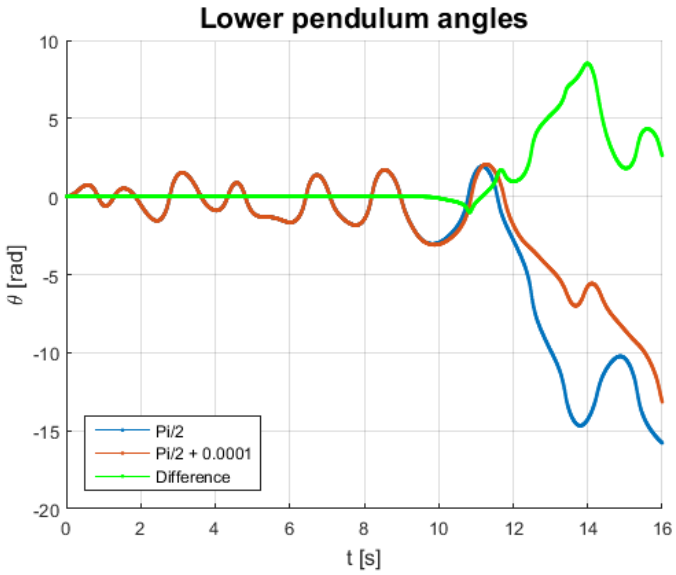


Simulation chaos experiment

```
% First year project "The Double Pendulum"
% Authors:
% Christian Schioett - BCN852
% Rasmus Nielsen - JBZ701
% Thue Nikolajsen - QRD689
% Date 18/03 2016

close all; clear all; clc;
% This is the beginning of the program

% Variables get set
g = 9.82;
m = [1 1];
dt = 0.01;

% Startbetingelser angives
R = [2 2];
L = R/2;
I = [1/12*m(1)*R(1)^2 1/12*m(2)*R(2)^2];
dif = [0 0.0001]; % This is the small difference in angle that we will add

% Constants important for computations
k(1) = m(1)*L(1)^2 + m(2)*R(1)^2 + I(1);
k(2) = m(2)*L(2)^2 + I(2);
k(3) = m(2)*R(1)*L(2);
k(4) = g*m(1)*L(1) + g*m(2)*R(1);
```

```
k(5) = g*m(2)*L(2);

% Setting up window
h = figure;
hold on; grid on;
t = title('Lower pendulum angles'); t.FontSize = 16;
lx = xlabel('t [s]'); ly = ylabel('\theta [rad]');
lx.FontSize = 12; ly.FontSize = 12;

% Equations of motion (and other functions) (X = S)
D = @(X) ( k(1)*k(2) - k(3)^2 * cos(X(2)-X(1))^2 );
f{1} = @(X) ( X(3)*k(2) - X(4)*k(3)*cos(X(2)-X(1)) ) / D(X);
f{2} = @(X) ( X(4)*k(1) - X(3)*k(3)*cos(X(2)-X(1)) ) / D(X);
f{3} = @(X) k(3)*sin(X(2)-X(1))*f{1}(X)*f{2}(X) - k(4)*sin(X(1));
f{4} = @(X) -k(3)*sin(X(2)-X(1))*f{1}(X)*f{2}(X) - k(5)*sin(X(2));

for j=1:1:length(dif)
    S = [pi/2 pi/2+dif(j) 0 0];

    for i=1:1:1600
        % Start of Runge Kutta integration
        for o = 1:4
            a(o) = f{o}(S);
        end
        for o = 1:4
            b(o) = f{o}(S + dt/2*a);
        end
        for o = 1:4
            c(o) = f{o}(S + dt/2*b);
        end
```

```
        for o = 1:4
            d(o) = f{o}(S + dt*c);
        end

        S = S + dt/6 * (a + 2*b + 2*c + d);
        % End of Runge Kutta integration

        v1(i) = S(1);
        v2(i) = S(2);
        T(i) = i*dt;
        V(j,i) = S(2) - S(1);
    end

    plot(T, v2 - v1,'- .')
end

plot(T,V(2,:)-V(1,:),'-g.')
legend('Pi/2','Pi/2 + 0.0001','Difference','Location','southwest')
```



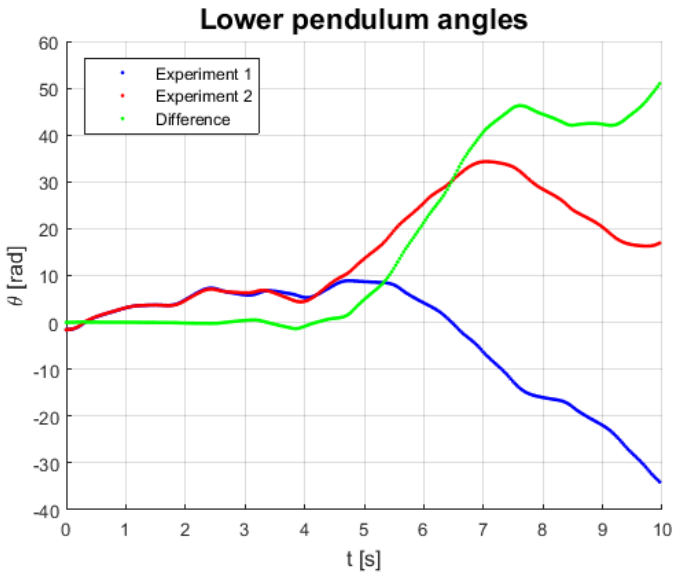
Data processing code

```
% First year project "The Double Pendulum"
% Authors:
% Christian Schioett - BCN852
% Rasmus Nielsen - JBZ701
% Thue Nikolajsen - QRD689
% Date 99/99 2016

close all; clear all; clc;
% This is the beginning of the program

% We load the data
D1 = importdata('real_data_one.txt');
D2 = importdata('real_data_two.txt');
T = D1.data(:,1);

x1_B = D1.data(:,2);
y1_B = D1.data(:,3);
x1_A = D1.data(:,4);
y1_A = D1.data(:,5);

% We take loops into acount
lapsB = 0;
lapsA = 0;
for i = 1:length(T)
   theta1_B(i) = atan2(x1_B(i), -y1_B(i));
   theta1_A(i) = atan2(x1_A(i) - x1_B(i), y1_B(i) - y1_A(i));
```

```
   if i >= 2
       dv = theta1_B(i) + 2*pi*lapsB - theta1_B(i-1);
       if abs( dv ) > pi/2
           if sign(dv) == -1
                lapsB = lapsB + 1;
           else
                lapsB = lapsB - 1;
           end
       end
       dv = theta1_A(i) + 2*pi*lapsA - theta1_A(i-1);
       if abs( dv ) > pi/2
           if sign(dv) == -1
                lapsA = lapsA + 1;
           else
                lapsA = lapsA - 1;
           end
       end
   end

   theta1_B(i) = 2*pi*lapsB + theta1_B(i);
   theta1_A(i) = 2*pi*lapsA + theta1_A(i);
end

x2_B = D2.data(:,2);
y2_B = D2.data(:,3);
x2_A = D2.data(:,4);
y2_A = D2.data(:,5);

lapsB = 0;
```

```
lapsA = 0;
for i = 1:length(T)
   theta2_B(i) = atan2(x2_B(i), -y2_B(i));
   theta2_A(i) = atan2(x2_A(i) - x2_B(i), y2_B(i) - y2_A(i));

   if i >= 2
       dv = theta2_B(i) + 2*pi*lapsB - theta2_B(i-1);
       if abs( dv ) > pi/2
           if sign(dv) == -1
                lapsB = lapsB + 1;
           else
                lapsB = lapsB - 1;
           end
       end
       dv = theta2_A(i) + 2*pi*lapsA - theta2_A(i-1);
       if abs( dv ) > pi/2
           if sign(dv) == -1
                lapsA = lapsA + 1;
           else
                lapsA = lapsA - 1;
           end
       end
   end

   theta2_B(i) = 2*pi*lapsB + theta2_B(i);
   theta2_A(i) = 2*pi*lapsA + theta2_A(i);
end

% Calculating the differences between the data sets
delta_theta_A = theta2_A - theta1_A;
```

```
% Setting up window
h = figure; set(gca,'Color','white'); set(gca,'fontsize',10);
hold on; grid on;
t = title('Lower pendulum angles'); t.FontSize = 16;
lx = xlabel('t [s]'); ly = ylabel('\theta [rad]');
lx.FontSize = 12; ly.FontSize = 12;

% We plot the difference over time
plot(T,theta1_A, 'b.')
plot(T,theta2_A, 'r.')
plot(T,delta_theta_A, 'g.')
legend('Experiment 1','Experiment 2','Difference','location','northwest')

% This is the end of the program
```



Frequencies and moments of inertia code

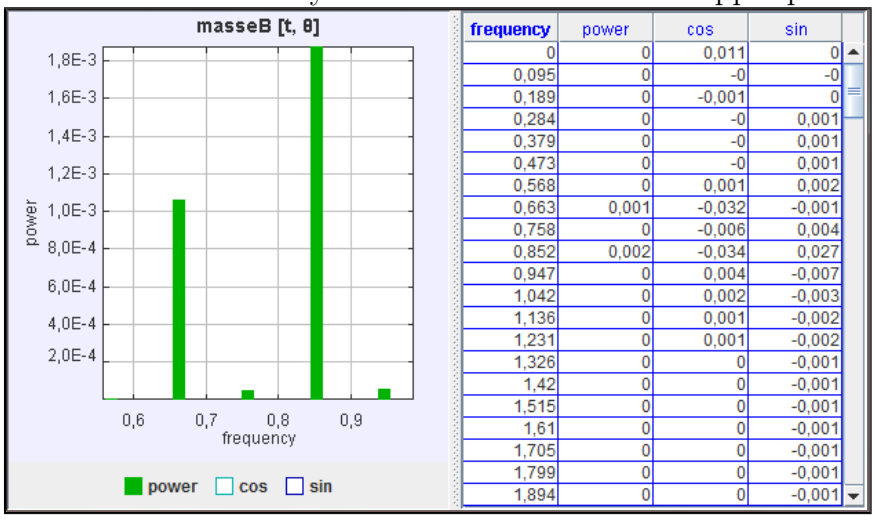
```
% First year project "The Double Pendulum"
% Authors:
% Christian Schioett - BCN852
% Rasmus Nielsen - JBZ701
% Thue Nikolajsen - QRD689
% Date 99/99 2016

close all; clear all; clc;
% This is the beginning of the program

% Variables get set
g = 9.82;                       % local gravity constant

% Physical properties are set
M = [888.3, 630.6]*10^-3;
R = [35.6, 33.9]*10^-2;
L = [16.0, 7.0]*10^-2;
T = [1.5068, 1.3951];

% Uncertainties in the physical properties
sigmaM = [0.01, 0.01]*10^-3;
sigmaR = [0.1, 0.1]*10^-2;
sigmaL = [0.2, 0.2]*10^-2;
sigmaT = [0.0001, 0.0001];

N = 10^6;
```

```
for i = 1:N
    % Choose values from norm-dist
    m_i(1) = randn()*sigmaM(1) + M(1);
    m_i(2) = randn()*sigmaM(2) + M(2);
    R_i(1) = randn()*sigmaR(1) + R(1);
    R_i(2) = randn()*sigmaR(2) + R(2);
    L_i(1) = randn()*sigmaL(1) + L(1);
    L_i(2) = randn()*sigmaL(2) + L(2);
    T_i(1) = randn()*sigmaT(1) + T(1);
    T_i(2) = randn()*sigmaT(2) + T(2);

    % Compute the moments of inertia
    I_i(1) = ( T_i(1)^2 * g/(4*pi^2) - L_i(1) ) * m_i(1) * L_i(1);
    I_i(2) = ( T_i(2)^2 * g/(4*pi^2) - L_i(2) ) * m_i(2) * L_i(2);

    % Constants important for computations
    k(1) = m_i(1)*L_i(1)^2 + m_i(2)*R_i(1)^2 + I_i(1);
    k(2) = m_i(2)*L_i(2)^2 + I_i(2);
    k(3) = m_i(2)*R_i(1)*L_i(2);
    k(4) = g*m_i(1)*L_i(1) + g*m_i(2)*R_i(1);
    k(5) = g*m_i(2)*L_i(2);

    n(1) = k(1) / (k(1)*k(2) - k(3)^2);
    n(2) = k(2) / (k(1)*k(2) - k(3)^2);
    n(3) = k(3) / (k(1)*k(2) - k(3)^2);

    % Coefficient matrix
    A = [[0,0,n(2),-n(3)]; [0,0,-n(3),n(1)]; [-k(4),0,0,0]; [0,-k(5),0,0]];
```

```
    % Find eigenvalues and eigenvectors of A
    [V,D] = eig(A);

    % Store the frquencies and moments of inertia
    f1(i) = imag( D(1,1) ) / (2*pi);
    f2(i) = imag( D(3,3) ) / (2*pi);
    I1(i) = I_i(1);
    I2(i) = I_i(2);
end
```

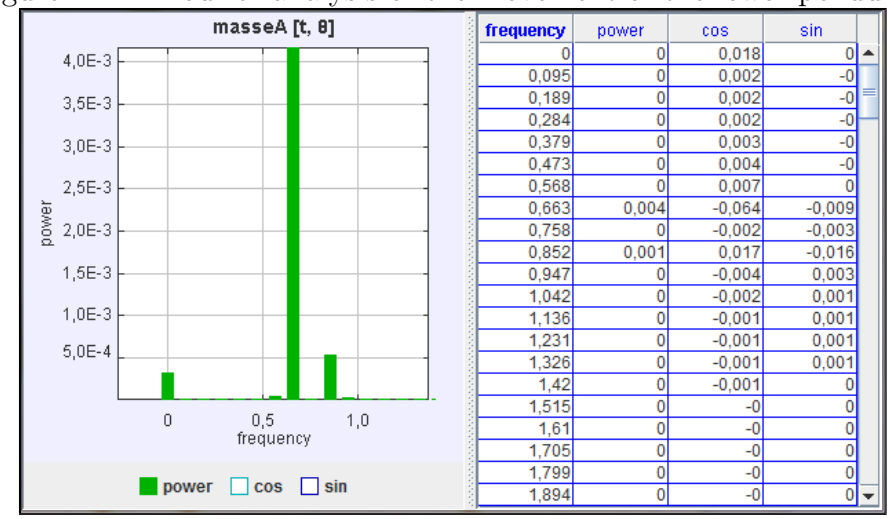
Fourier data

Figure F.1: A fourier analysis of the movement of the upper pendulum.



frequency	power	cos	sin
0	0	0,011	0
0,095	0	-0	-0
0,189	0	-0,001	0
0,284	0	-0	0,001
0,379	0	-0	0,001
0,473	0	-0	0,001
0,568	0	0,001	0,002
0,663	0,001	-0,032	-0,001
0,758	0	-0,006	0,004
0,852	0,002	-0,034	0,027
0,947	0	0,004	-0,007
1,042	0	0,002	-0,003
1,136	0	0,001	-0,002
1,231	0	0,001	-0,002
1,326	0	0	-0,001
1,42	0	0	-0,001
1,515	0	0	-0,001
1,61	0	0	-0,001
1,705	0	0	-0,001
1,799	0	0	-0,001
1,894	0	0	-0,001

Figure F.2: A fourier analysis of the movement of the lower pendulum.



frequency	power	cos	sin
0	0	0,018	0
0,095	0	0,002	-0
0,189	0	0,002	-0
0,284	0	0,002	-0
0,379	0	0,003	-0
0,473	0	0,004	-0
0,568	0	0,007	0
0,663	0,004	-0,064	-0,009
0,758	0	-0,002	-0,003
0,852	0,001	0,017	-0,016
0,947	0	-0,004	0,003
1,042	0	-0,002	0,001
1,136	0	-0,001	0,001
1,231	0	-0,001	0,001
1,326	0	-0,001	0,001
1,42	0	-0,001	0
1,515	0	-0	0
1,61	0	-0	0
1,705	0	-0	0
1,799	0	-0	0
1,894	0	-0	0