

Swing Up and Stabilisation of an Inverted Pendulum on a Cart using Nonlinear Methods

Jesper Hede Christensen
Rasmus Christiansen

Master's Thesis
Control and Automation
Department of Electronic Systems
June 2017





Title:

Swing Up and Stabilisation of an Inverted Pendulum on a Cart using Nonlinear Methods

Theme:

Nonlinear control

Project period:

Spring 2017
2017-02-01 – 2017-06-08

Group number:

1030

Group member:

Jesper Hede Christensen
Rasmus Christiansen

Supervisors:

John-Josef Leth
Palle Andersen

Pages: 112

Appendices: 8

Concluded: 2017-06-08

Abstract:

This thesis describes the process of designing a nonlinear control scheme for swinging up and stabilising an inverted pendulum on a cart.

A model of the cart and pendulum dynamics has been derived using Lagrangian mechanics. The actuator dynamics are modelled by a first order transfer function. The combined model has been implemented in Simulink and verified. An Extended Kalman Filter has been developed to estimate the full state vector.

A two-step control strategy has been developed, which consists of *swing-up* controllers and *stabilising* controllers. A *catch trigger* algorithm manages the transition between the controllers. Swinging up the pendulum concerns controlling the energy of the system such that the pendulum reaches its heteroclinic orbits.

The stabilising controllers keep the cart close to a position reference while keeping the pendulum upright at its unstable equilibrium. Six controllers; three swing-up and three stabilising, have been designed, implemented and compared.

Results indicate minimal difference between the stabilising controllers when considering their ability to keep the pendulum upright and the cart at a desired position.

Preface

This Master's thesis is produced by group CA10-1030 and concludes the fourth and final semester in the Master's programme Control and Automation at the Department of Electronic Systems, Aalborg University.

Citation before a full stop regards sentence and after full stop at the end of a paragraph regards the entire paragraph. Chapters or sections starting with "The following is based on [Reference]" means the entire chapter or section are based entirely on that source.

The .zip-file handed in along with this document contains the source code developed during the course of the project:

- MATLAB/Simulink model
- C/C++ implemented code

Thanks to John-Josef Leth and Palle Andersen from Aalborg University for great support and supervision during the project.



Jesper Hede Christensen



Rasmus Christiansen

Figure 1: The members of group 17GR1030.

Resumé

Dette projekt omhandler ulineært reguleringsdesign af et omvendt pendul på en vogn, og forsøger at besvare problemstillingen:

Hvordan kan et ulineært reguleringsdesign udvikles således det kan svinge det omvendte pendul på vognen op og stabilisere det, og hvordan klarer den ulineære, stabiliserende regulerering sig med lineære reguleringsmetoder?

For at besvare dette er et fysisk inverteret pendul på en vogn, som står tilrådighed i reguleringslaboratoriet på Aalborg Universitet, blevet analyseret og modelleret. Den mekaniske del af modellen er udledt på baggrund af Lagrange-mekanik, og den elektriske del er approksimeret ved en første ordens overføringsfunktion. Den mekaniske model er opstillet på tilstandsform, og et udvidet Kalman-filter er designet således alle tilstænde kendes.

Den overordnede reguleringsstrategi er baseret på to faser, en opsvingende og en stabiliserende fase. Dertil er der designet en algoritme, som varetager overgangen mellem faserne, styret af pendulets vinkel og den mekaniske energi i pendulet.

Designet af reguleringen til opsvinget er baseret på mekanisk energi, og forsøger at tvinge pendulets faseportræt ud mod dets heterokliniske kredsløb. Pendulets friktion er udeladt fra designprocessen, hvilket gør at heuristiske metoder har vist sig nødvendige til at svinge pendulet op til dets ustabile ligevægtspunkt. Der er designet tre forskellige metoder til opsving.

Idet pendulet nærmer sig sit ustabile ligevægtspunkt, og kriterierne for overgangsalgoritmen er opfyldt, vil den stabiliserende regulering overtage styringen. Til dette er der designet et kaskadekoblet PID-reguleringsssystem, en LQG-regulator og en sliding-mode regulator, hvis formål er at stabilisere pendulet i dets ustabile ligevægtspunkt, samt at holde vognen nær en ønsket position.

I alt er der designet, implementeret, testet og sammenlignet seks forskellige reguleringsmetoder på det fysiske system. Til opsving viser metoden baseret på pendulets energi alene mest lovende resultater. Derudover viser resultaterne minimal forskel mellem de stabiliserende reguleringsdesigns.

Definitions

Notation

Throughout this thesis, general mathematical notation is used. However, in order to avoid ambiguity, the following conventions are used:

- Vectors and matrices are denoted in bold, lowercase and bold uppercase text, respectively, e.g. \mathbf{x} and \mathbf{A} .
- Trigonometric functions $\sin(\cdot)$ and $\cos(\cdot)$ taken with respect to one variable will have the parentheses omitted, e.g. $\sin x = \sin(x)$, but $\sin xk = \sin(x) \cdot k$.
- Variables sampled at time step $k \cdot T_s$ will be subscripted x_k .
Variables sampled at time step $k \cdot T_s$ and based on data from the previous sample will be subscripted $x_{k|k-1}$, or, if based on data from the current sample, $x_{k|k}$.
- Matrix indices will be referred to by $M_{(i,j)}$, meaning the index at row i and column j of matrix M .
- A matrix size is indicated by $M_{a \times b}$, where a and b denotes the row and column length respectively.
- Variable references are denoted with a superscripted asterisk, e.g. x^* .
- Variables estimates are denoted with a hat, e.g. \hat{x} .

Acronyms

ARM Advanced RISC Machine.

back-EMF back-electromotive force.

DAC Digital to Analog Converters.

EKF Extended Kalman Filter.

LQG Linear-Quadratic-Gaussian.

LQR Linear-Quadratic Regulator.

MCU Microcontroller Unit.

MSE Mean Square Error.

PCB printed circuit board.

PID Proportional-Integral-Derivative.

RMSE Root Mean Square Error.

SMC Sliding-Mode Controller.

SRAM Static Random-Access Memory.

Nomenclature

Table 1 lists the most commonly used variables and constants.

Symbol	Description	Unit
x	Cart position	m
θ	Pendulum angle	rad
\dot{x}	Cart velocity	$\frac{m}{s}$
$\dot{\theta}$	Pendulum velocity	$\frac{rad}{s}$
u	System input	F
\mathbf{x}	State vector	—
\mathbf{r}	Position vector	—
\mathbf{q}	Generalised coordinate vector with x and θ	—
\mathbf{M}	Inertial properties matrix	—
\mathbf{C}	Centrifugal force vector	—
\mathbf{B}	Friction vector	—
\mathbf{G}	Gravitational force vector	—
F	Force	N
E, E_p	System energy and pendulum energy	J
c	Coulomb friction	F
v	Viscous friction	F
g	Gravitational acceleration	$\frac{m}{s^2}$
m_c, m_p	Cart and pendulum mass	kg
l	Pendulum length	m

Table 1: List of most common variables and constants used throughout the thesis.

Contents

1	Introduction	1
2	System Description	3
2.1	Arduino Due	4
2.2	Servo amplifier	5
2.3	Optical encoder and quadrature decoder	6
2.4	Rail and cart observations	7
3	Problem Statement	9
3.1	Requirements	9
3.2	Outline	9
4	Modelling	11
4.1	Motor electromechanics	11
4.2	Cart and pendulum mechanics	13
4.3	Cart and pendulum frictions	16
4.4	State space formulation	20
4.5	Simulink implementation	21
4.6	Model verification	24
5	State Estimation	29
5.1	Numerical derivative	29
5.2	Extended Kalman Filter	30
5.3	Comparison	32
6	Control Strategy	35
6.1	Stability	36
6.2	Mechanical energy and heteroclinic orbits	36
6.3	Cart friction feedforward	39
6.4	Catch trigger	40
7	Control Design: Swing-up	41
7.1	Pendulum energy method	41
7.2	Full energy method	44
7.3	Heuristic swing-up control design	47
8	Control Design: Stabilisation	53
8.1	Proportional-Integral-Derivative control	53
8.2	Linear-Quadratic-Gaussian regulator	55
8.3	Sliding-Mode Controller	58
9	Implementation	63
9.1	Simulink	63
9.2	Arduino	64

10 Results	69
10.1 Swing-up control	69
10.2 Stabilising control	70
10.3 Robustness	71
11 Discussion	75
12 Conclusion	77
Bibliography	79
A Integer Value to Current	81
B Distributed Cart Coulomb Friction	83
C Cart Viscous Friction	87
D Test of Teflon-coated Belt	93
E Pendulum Friction Estimation	95
F Aerodynamic Drag	97
G Determination of Measurement Noise	99
H Initialisation Error Test	101

In April 2014, aerospace manufacturer *SpaceX* launches their *Falcon 9* rocket on a resupply mission to the International Space Station. The first stage, the *booster stage*, manages to perform a controlled splashdown after separation from the rest of the rocket. Later, in January 2015, *SpaceX* attempts a full, vertical landing of the booster stage onto a drone ship. While this ends with a crash, it has been attempted several times since, and *SpaceX* succeeded in landing the booster stage on land in December 2015.[1]



Figure 1.1: *Falcon 9* booster stage landing on the "Just Read the Instructions" drone ship.[2]

Landing a rocket vertically is difficult, as it involves decelerating, manoeuvring and balancing the rocket at the same time. Deceleration is achieved by providing thrust in the opposite direction of the velocity vector of the rocket, but manoeuvring requires angling the thrust vector, and this produces both linear and rotational acceleration, depending on the rocket design.[3]

This behaviour resembles balancing a pole on the fingertip – an inverted pendulum.

An inverted pendulum is a textbook example of a highly nonlinear, open-loop unstable system. It is unstable in the sense that in order to keep it upright, it needs swift, active stabilisation. Without such stabilisation, it will fall over.[4]

With a nonlinear system, the task of designing an appropriate control scheme becomes challenging. Linear control methods can be applied, but require a linearised model to design. Alternatively, nonlinear control design deals with stabilisation and control of nonlinear systems directly.[5]

This thesis describes the analysis, modelling and control of an inverted pendulum system, focusing on nonlinear control design in comparison with linear ones.

System Description 2

This chapter describes the test setup considered in this project. First, the general construction and connections are described. Then, in **section 2.1** the functionality of the used Microcontroller Unit (MCU) is presented. The description continues with the motor controller and sensors in **section 2.2** and **section 2.3** respectively. In **section 2.4**, some initial system observations are presented.

The setup is located in the control laboratory at Aalborg University and is depicted in **figure 2.1**.

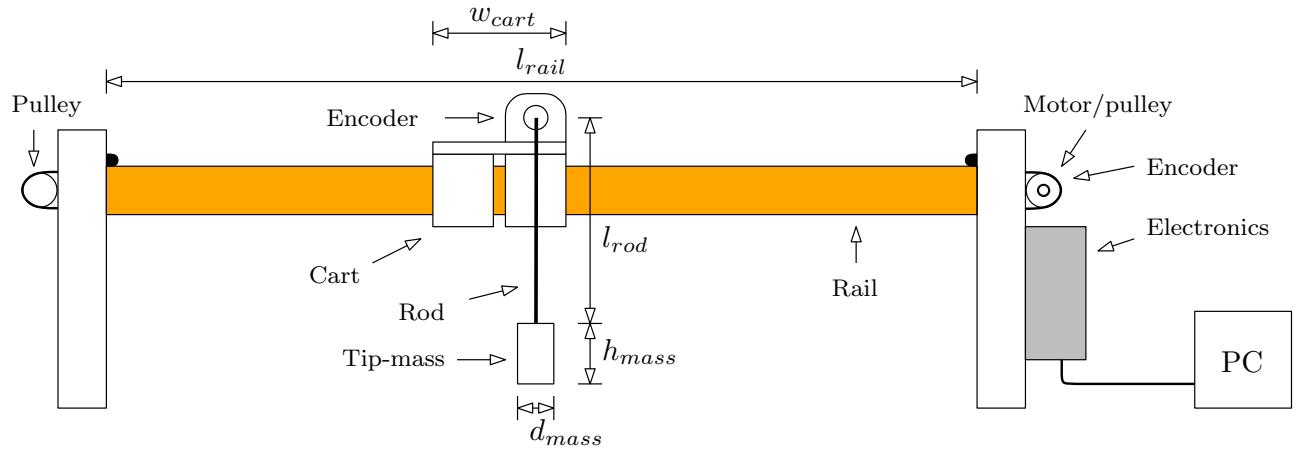


Figure 2.1: Test setup with essential physical dimensions.

It consists of a pendulum attached to a cart. The cart is attached to the rail using ball bearings, and is able to move horizontally. A toothed belt connects the cart to the pulleys at each end of the rail. The pulley at the right side is driven by a maxon DC motor, using current control, which effectively drives the cart. Attached to the motor is an incremental optical encoder which is used to measure the cart position.

The pendulum consists of a light metal rod attached to another maxon DC motor, though this is only used for the encoder attached to it in order to measure the angular displacement of the pendulum.

A cylindrical, modular mass is attached at the end of the pendulum rod. It is modular in the sense that it consist of multiple, threaded masses of 100, 75, 50 and 26 grams, which can be screwed together in any combination. By default, all masses will be used. A list of dimensions and the tip-mass is given in **table 2.1**.

The grey box on **figure 2.1** symbolises the electronics of the setup, which includes a maxon servo amplifier[6], an Arduino Due[7], a power supply unit and finally a printed circuit board (PCB), which contains a voltage shield and quadrature decoders.

The Arduino Due operates at 3.3 V, which is insufficient for controlling the DC motor through the power amplifier, as this requires a voltage range of ± 10 V at its input pin. Thus, the voltage shield is used to map the Arduino output of 0 – 3.3 V to the amplifier range. The Arduino is used to compute e.g. controller output and also serves as the interfacing device between sensors and the motor. A C++

Name	Value	Unit	Description
m_p	251	g	Tip-mass of the pendulum
l_{rod}	28.28	cm	Length of the rod between tip-mass and joint
d_{rod}	1	cm	Diameter of the rod
h_{mass}	10.4	cm	Height of tip-mass
d_{mass}	1.95	cm	Diameter of tip-mass
l_{rail}	88.9	cm	Length of the cart rail
w_{cart}	33.45	cm	Width of the cart
l	33.45	cm	Pendulum length, $l_{rod} + 0.5h_{mass}$
r	3	cm	Radius of the pulleys

Table 2.1: System parameters measured by scale or measurement tape.

library, `Joint.cpp`, has been supplied for the low-level interaction between these.

The Arduino-motor-encoder signal path is depicted in **figure 2.2**.

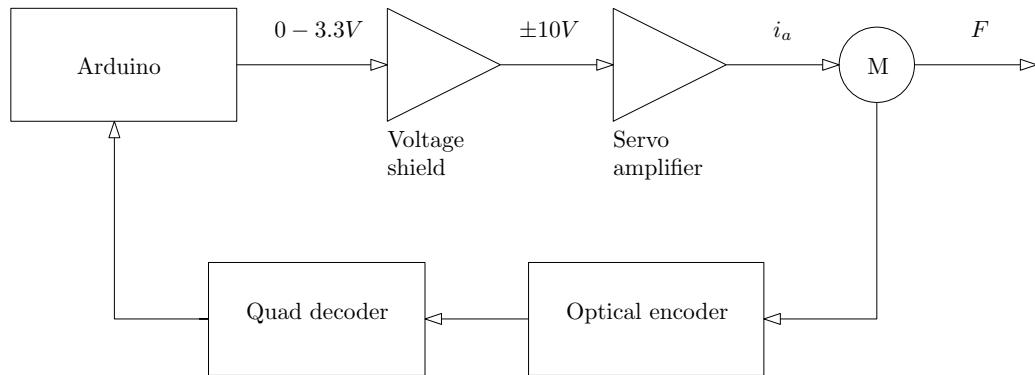


Figure 2.2: Representation of the signal path between the Arduino to the motor, and back through the encoder and decoder.

2.1 Arduino Due

The Arduino Due is an MCU with a 32-bit Atmel SAM3X8E Advanced RISC Machine (ARM) Cortex-M3 processor with a clock speed of 84 MHz. It has 512 kilobytes of flash memory and 100 kilobytes of Static Random-Access Memory (SRAM), and is thus well suited for the purpose of this project. It also features two 12-bit Digital to Analog Converters (DAC).

The connections between the Arduino and the remaining hardware is depicted in **figure 2.3**.

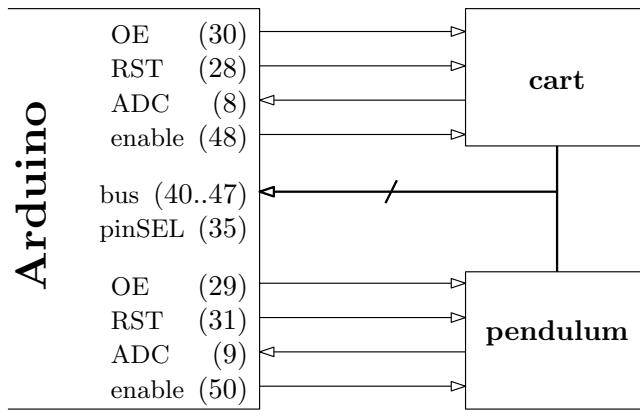


Figure 2.3: Arduino pin connection between the cart and pendulum hardware.

The connections between the Arduino and the cart and pendulum are similar. Each connection is displayed with the pin/peripheral name and pin number. The diagram is simplified such that the blocks *cart* and *pendulum* each contain the servo amplifier, motor and optical encoder, respectively. Each connection is described as follows:

- DAC** Digital to analog converter: Converts a digital number to a voltage output from the Arduino. Used to set a motor current reference for the motor amplifier.
- OE** Output enable: Enables reading from the encoder when set to binary low.
- bus** 8 pin data bus.
- pinSEL** Pin selector: Pin used to select whether to read low byte (binary high) or high byte (binary low) reading on the data bus. The low and high bytes are then combined to create a 16-bit signed integer value.
- enable** Enable pin: Used for initialization of servo amplifier output.
- RST** Reset pin: Resets the encoders.

The data bus is used to read decoded sensor values, and works as follows

1. Pin OE of the device to read from is pulled low to enable data output.
2. Low byte is requested and received.
3. High byte is requested and received.
4. Pin OE is pulled high again, ending the transaction.
5. The low and high bytes are combined to form the 16-bit integer value.

The Arduino samples the sensors with a period of 0.05 s and sends this, along with other data, via the USB serial protocol at 115200 baud to a computer.

2.2 Servo amplifier

The servo amplifier is used to provide current control of the motor. The servo amplifier reads a current reference from a ± 10 V pin, and applies this to the motor in an internal current control loop.

The current reference is set as a 12-bit integer, which is converted to an analogue value using the DAC on the Arduino board, outputting a corresponding voltage in the 0 – 3.3 V range. This is amplified by the voltage shield and applied to the servo amplifier pin in the ±10 V range.

Appendix A describes a test which has been conducted to determine the relation between the 12-bit integer value and the current supplied to the motor. The results are summarized in **figure 2.4**.

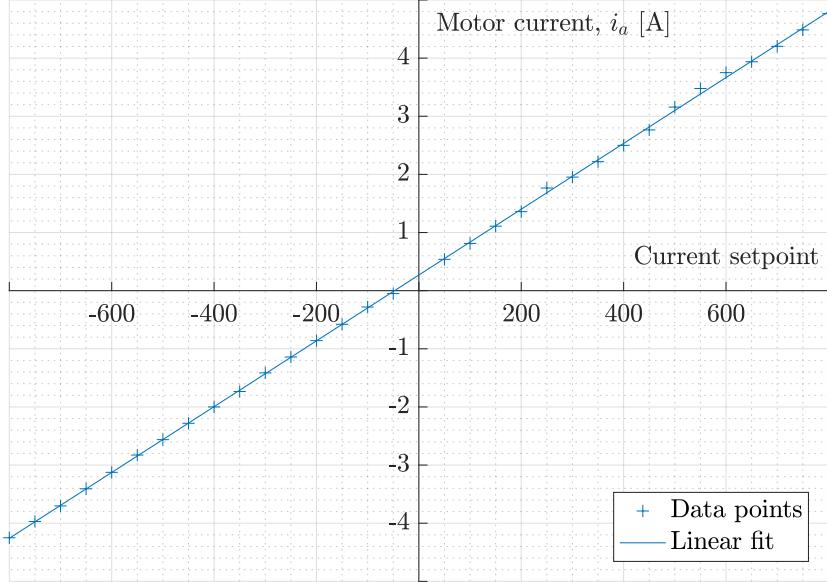


Figure 2.4: Integer to current test in order to determine direct relationship between the integer value of the current setpoint and the current.

When setting a current reference value, called the *setpoint*, the corresponding 12-bit integer is shifted by 2000 in the `Joint.cpp` library in order to be able to represent negative values and drive the motor in both directions.⁽¹⁾

A relation between the setpoint and the force applied to the cart is found using the motor torque constant, k_t [8], radius of the pulley attached to the motor, r , and the slope of the data in **figure 2.4**, α . This relation is found to be

$$F = i_a \frac{k_t}{r} = \text{setpoint} \cdot \alpha \frac{k_t}{r} \quad (2.1)$$

$$\Leftrightarrow \text{setpoint} = \frac{r}{\alpha k_t} F, \quad \alpha = 0.0057 \quad (2.2)$$

The relation is used in a modification of the `Joint.cpp` library, such that a force reference can be set instead of an integer value.

2.3 Optical encoder and quadrature decoder

Each motor has an incremental optical encoder attached to its shaft, which is used along with a quadrature decoder to measure the cart and pendulum positions by counting small increments of rotation on the motor shafts.

⁽¹⁾In **figure 2.4**, it is seen that setting the setpoint to zero does not result in a current control reference of zero ampere. This is corrected by adjusting the offset from 2000 to 1953.

The optical encoder is an Avago HEDS-5540 [9] with two channels. It contains an LED from which the light is redirected by a lens onto an encoder disc, which has a pattern of holes that lets light through in a similarly patterned fashion when rotated. The light pattern hits a group of photodiodes, where each outputs a current depending on the amount of light that hits it.

A detector circuit is used to detect the lighting pattern from the photo diodes, and outputs a quadrature signal in two channels, A and B, where channel A is 90 degrees out of phase with channel B, indicating the direction of the motor rotation. The working principle of the optical encoder is shown in figure 2.5.

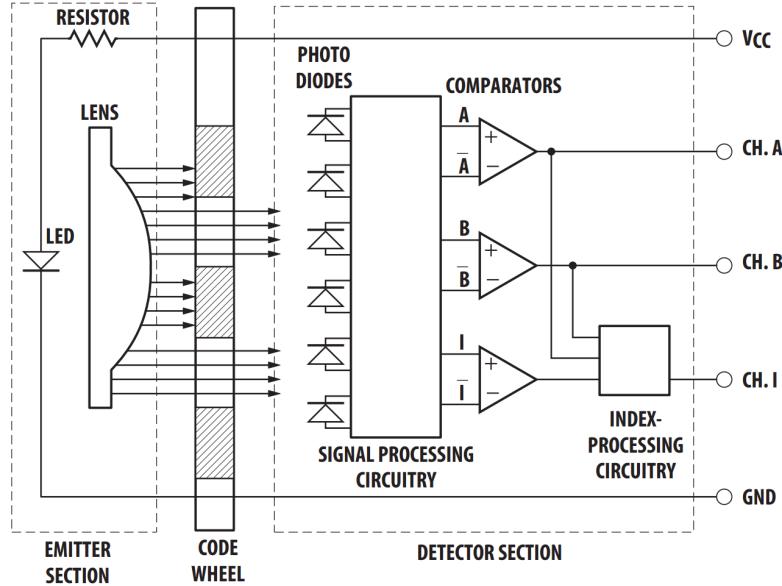


Figure 2.5: Block diagram of the optical encoder HEDS-5540[9]

The output of the optical encoder is decoded using a Avago HCTL-2021-PLC [10] quadrature decoder IC chip, which is located on the PCB with the voltage shield. This decoder features a 16-bit up/down counter and an 8-bit bus interface, which is connected to the Arduino.

The result is a *tick* count, depending on how far the motor shafts are rotated from their initial position. The decoder outputs a count of 2000 ticks per revolution, and the pendulum angle have 2π radians per revolution providing a resolution of

$$\min \Delta\theta = \frac{2\pi}{2000} = \pi \cdot 10^{-3} \frac{\text{rad}}{\text{ticks}} \quad (2.3)$$

The maximum travel length of the cart is $l_{\text{rail}} - w_{\text{cart}} = 0.769$ m, and the total number of ticks when the cart moves on the rail is 8737, providing a resolution of

$$\min \Delta x = \frac{0.769}{8737} = 88.016 \cdot 10^{-6} \frac{\text{m}}{\text{ticks}} \quad (2.4)$$

The resolution of the position measurements is thus sufficient for this project.

2.4 Rail and cart observations

The entire system has been made available *as is* in the beginning of the project. This section serves to document some initial observation about the test setup.

Initially, it has been observed that the cart does not run smoothly on the rails, and the friction depends on its position and direction.

Therefore, before proceeding to estimate any parameters, some alterations to the system have been considered:

- Leveling of the system has been adjusted.
- The rail has been polished and greased.

The cart moves on the rail using ball bearings, and is driven by a toothed belt and a motor. By detaching the belt from the cart and letting it move freely on the rail, it has been observed to move with less friction.

The ball bearings introduce some friction as well, especially at specific points where the Coulomb friction increases. That is, at low velocities the cart tends to get stuck and needs extra force to move. It is assumed that the ball bearing friction is amplified through the power transmission, i.e. the belt and pulleys.

Furthermore, the belt introduces friction between its teeth and the pulley. In an attempt to minimize this, another type of belt with a Teflon treated surface has been tested as described in **appendix D**. However, the new belt introduces more friction since the current consumption at the same velocity increased. This is assumed to be due to the new belt being stiffer, such that more mechanical work is wasted on bending the belt.

Thus, the original belt is used for the project.

Problem Statement 3

Given the topic described in **chapter 1** and the system description in **chapter 2**, the following problem statement is developed:

How can a nonlinear control scheme be developed in order to swing up and stabilise an inverted pendulum on a cart, and how does the stabilising control compare to linear approaches?

3.1 Requirements

The requirements for considering the control scheme successful is divided into two categories; functional and additional requirements.

3.1.1 Functional requirements

The functional requirements are the necessary capabilities that must be fulfilled in order to consider the scheme operational.

- 1) Able to autonomously swing up and stabilise the pendulum in its upright position.
- 2) Keep the cart from colliding with the physical barriers of the system.

3.1.2 Additional requirements

The additional requirements are features that are desirable, but not necessary.

- 1) Swing up the pendulum to within ± 10 degrees (~ 0.1745 radians) of its upright position.
- 2) Swing up the pendulum in less than 10 seconds.
- 3) Stabilise the pendulum with oscillations less than ± 2 degrees (~ 0.0349 radians).
- 4) Keep the cart within ten centimetres of the reference position while stabilising the pendulum.
- 5) Offer robustness for up to 10 percent variation in the tip-mass.

3.2 Outline

In order to fulfil the requirements stated in **section 3.1**, a system model and control laws will be derived and implemented. The derivation of the system model is presented in **chapter 4**.

A two-step control strategy is considered and described in **chapter 6**, as the control scheme required to swing the pendulum up differs from the one required to stabilise it. The proposed swing-up controllers are presented in **chapter 7** and the stabilising controllers in **chapter 8**. Several of these controllers require information of the cart and pendulum velocities, and thus an Extended Kalman Filter (EKF) is designed in **chapter 5**.

All the control schemes are implemented on the Arduino Due at the physical test setup, as described in **chapter 9**, and the results are shown in **chapter 10**.

Modelling 4

This chapter describes the modelling process of the mechanical and electrical part of the system. First, a model of the electrical part, motor and amplifier is derived in **section 4.1**, followed by a model derivation for the system dynamics using Lagrangian mechanics in **sections 4.2** and **4.3**. In **section 4.4**, the model is reformulated into state-space form for later reference and the implementation in MATLAB is described in **section 4.5**. Finally, the model is verified in **section 4.6**.

4.1 Motor electromechanics

The DC motor model is based on the circuit diagram in **figure 4.1**.

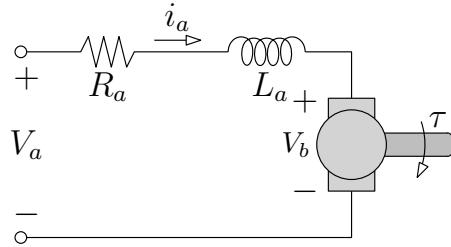


Figure 4.1: DC motor circuit diagram.

Here, i_a is the current through the armature, L_a and R_a are the motor inductance and resistance, respectively, V_b is the back-electromotive force (back-EMF), a voltage generated by the shaft motion, and V_a is the voltage applied to the motor terminals.

From Kirchhoff's voltage law, the motor dynamics are described by

$$V_a(t) = L_a \frac{di(t)}{dt} + R_a + V_b(t) \quad (4.1)$$

and the torque produced by the motor is

$$\tau(t) = k_t i(t) \quad (4.2)$$

where k_t is the motor torque constant.

A block diagram illustrating the interaction between the motor dynamics and the system mechanics is shown in **figure 4.2**.

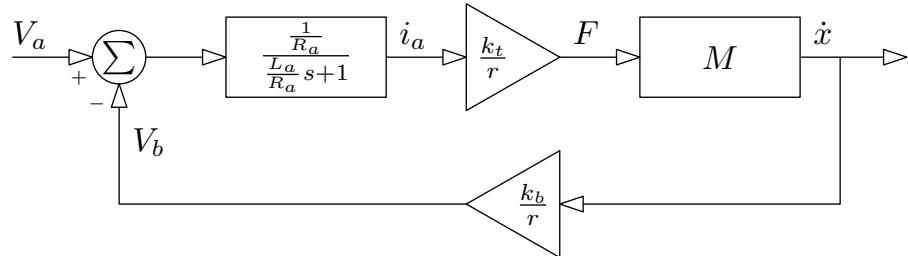


Figure 4.2: Motor and system interaction.

In this, M is the mechanical part of the system, which will be considered in **section 4.2**. The parameters k_t and k_b are the motor torque and back-EMF constants, respectively, which are equal, when assumed that the DC motor is ideal, thus $k_b = k_t$.

The servo amplifier features an internal current control for the motor, which is considered as depicted in **figure 4.3**, where the back-EMF voltage is viewed as a disturbance, and the mechanics block M has been omitted.

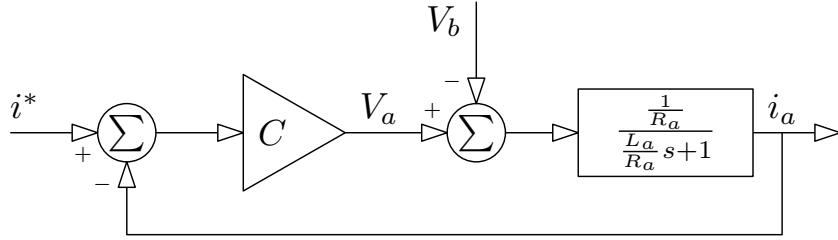


Figure 4.3: Motor model including current controller

Here, C is the current controller, and i^* is the current reference. From **figure 4.3** it is possible to establish a transfer function of the motor in the s -domain as

$$H(s) = \frac{i_a}{V_a - V_b} = \frac{\frac{1}{R_a}}{\frac{L_a}{R_a}s + 1} = \frac{K_m}{\tau_m s + 1} \quad (4.3)$$

where K_m and τ_m are constants to be determined experimentally.

In order to do this, a pulse train test was conducted where a signal pulsing between ± 4 N (approximately ± 1.3 A) with a duration of 500 ms was applied to the motor. Measuring the current directly from the servo amplifier using an oscilloscope yields the results in **figure 4.4**.

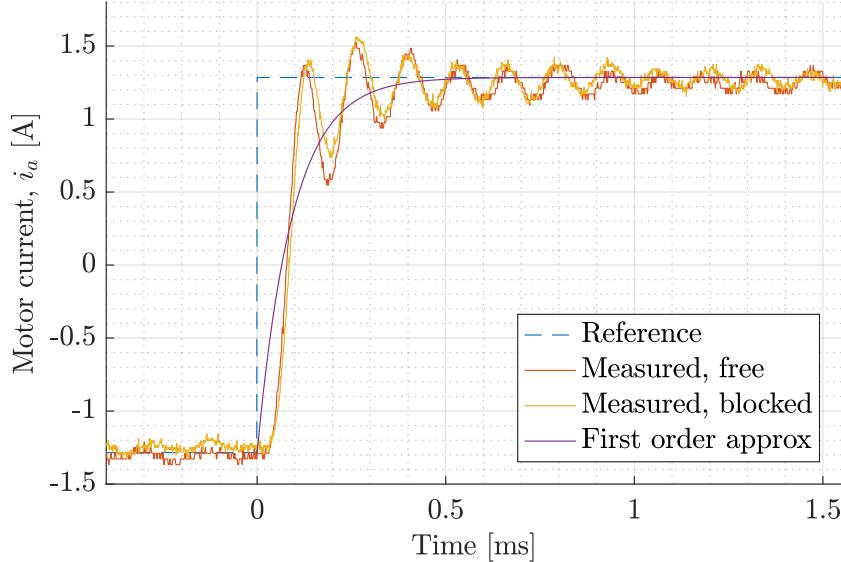


Figure 4.4: Motor step responses from two tests, where the motor have been blocked and able to run freely. The noise on the step responses is assumed to be commutator noise, which is proportional to the velocity by some constant.

The signals *Measured, free* and *Measured, blocked* are two different tests using the same input, and refers to whether the motor has been blocked or been able to move freely. This has been done to

investigate the disturbance of the back-EMF voltage. As it can be seen, the signals are almost identical, concluding that the servo amplifier is able to suppress this disturbance. The settling time of the current control less than 1 ms, which is much faster than the sampling period of the system of 0.05 ms.

A first order approximation is assumed sufficient to represent the dynamics of the current control, and is found, using MATLAB function `tfest`, to be

$$H(s) = \frac{1.065 \cdot 10^4}{s + 1.058 \cdot 10^4} \quad (4.4)$$

4.2 Cart and pendulum mechanics

The following is based on [11].

In order to model the forces acting on the cart and pendulum, Lagrangian mechanics are used. **Figure 4.5** depicts the mechanical system, where the position vector of the cart is denoted \mathbf{r}_c and the position vector of the pendulum, at its assumed centre of mass, is denoted \mathbf{r}_p .

The mass of the pendulum, assumed evenly distributed, and the distance from the pendulum joint at \mathbf{r}_c to its centre of mass at \mathbf{r}_p , are denoted by m_p and l , respectively, for which the measured values are listed in **table 2.1**. The pendulum rod is assumed massless. Finally, the angular displacement of the pendulum, from the upright position, is denoted by θ .

It should be noted that the mass of the cart, denoted m_c , is actually a unified parameter for the masses and inertias present in the entire power transmission, i.e. from the motor to the cart itself. Thus, the parameter m_c includes:

- Rotor inertia of the motor
- Pulley inertias
- Belt mass
- Cart mass

Furthermore, the belt is assumed to be non-elastic, thus no phase shift in the power transmission should occur.

It is seen from **figure 4.5** that

$$\mathbf{r}_c(t) = \begin{bmatrix} x(t) \\ 0 \end{bmatrix}, \quad \mathbf{r}_p(t) = \begin{bmatrix} x(t) - l \sin \theta(t) \\ l \cos \theta(t) \end{bmatrix} \quad (4.5)$$

$$\dot{\mathbf{r}}_c(t) = \begin{bmatrix} \dot{x}(t) \\ 0 \end{bmatrix}, \quad \dot{\mathbf{r}}_p(t) = \begin{bmatrix} \dot{x}(t) - l \cos \theta(t)\dot{\theta}(t) \\ -l \sin \theta(t)\dot{\theta}(t) \end{bmatrix} \quad (4.6)$$

In order to simplify the notation throughout the rest of this section, explicit notation of time dependency will be omitted, i.e.

$$x = x(t), \quad \theta = \theta(t) \quad (4.7)$$

$$\mathbf{r}_c = \mathbf{r}_c(t), \quad \mathbf{r}_p = \mathbf{r}_p(t) \quad (4.8)$$

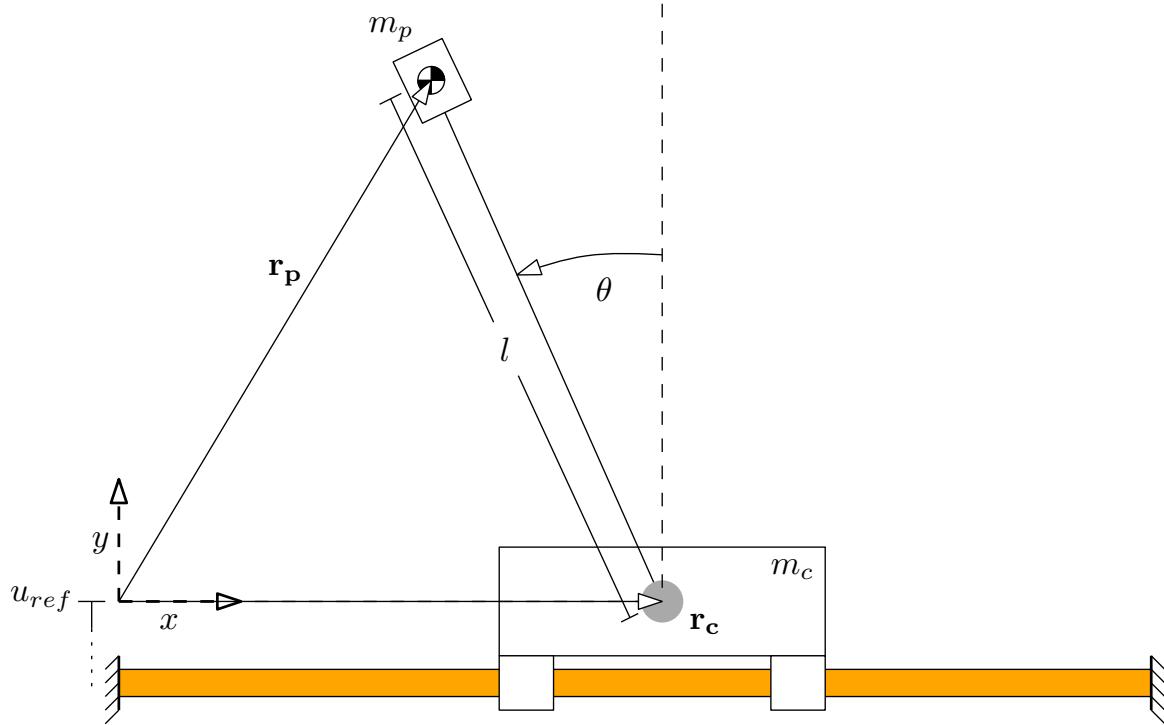


Figure 4.5: Diagram of cart, pendulum, and rail. Essential vectors, parameters and variables are shown.

and equivalently for their derivatives.

The differential equations describing the system mechanics are derived based on the kinetic energy T and the potential energy U of the system, where the *Lagrangian* is defined by

$$L = T - U = (T_c + T_p) - (U_c + U_p) \quad (4.9)$$

The potential energies of the system are defined by

$$U_c = u_{ref} \quad (4.10)$$

$$U_p = m_p g l \cos \theta + u_{ref} \quad (4.11)$$

where u_{ref} is the reference potential, i.e. the height that potential energies are relative to, and g is the gravitational constant.

The kinetic energy of the cart is defined by

$$T_c = \frac{1}{2} m_c \dot{\mathbf{r}}_c^T \dot{\mathbf{r}}_c = \frac{1}{2} m_c \begin{bmatrix} \dot{x} & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ 0 \end{bmatrix} = \frac{1}{2} m_c \dot{x}^2 \quad (4.12)$$

as the cart is only capable of moving on the horizontal axis.

The kinetic energy of the pendulum is expressed by

$$T_p = \frac{1}{2} m_p \dot{\mathbf{r}}_p^T \dot{\mathbf{r}}_p = \frac{1}{2} m_p \left((\dot{x} - l \cos \theta \dot{\theta})^2 + (-l \sin \theta \dot{\theta})^2 \right) \quad (4.13)$$

The Lagrangian thus becomes:

$$\begin{aligned} L(x, \dot{x}, \theta, \dot{\theta}) &= \frac{1}{2}m_c\dot{x}^2 + \frac{1}{2}m_p \left((\dot{x} - l \cos \theta \dot{\theta})^2 + (-l \sin \theta \dot{\theta})^2 \right) - m_p gl \cos \theta - 2u_{ref} \\ &= \frac{1}{2}(m_c + m_p)\dot{x}^2 - m_p l \cos \theta \dot{\theta} \dot{x} + \frac{1}{2}m_p l^2 \dot{\theta}^2 - m_p gl \cos \theta - 2u_{ref} \end{aligned} \quad (4.14)$$

From this, the dynamics of the system are found by the *Euler-Lagrange* equation with generalised coordinates $\mathbf{q} = [x \ \theta]^T$ and external forces \mathbf{Q} as

$$\begin{aligned} \mathbf{Q} &= \frac{d}{dt} \frac{\partial L}{\partial \dot{\mathbf{q}}} - \frac{\partial L}{\partial \mathbf{q}} \\ &= \frac{d}{dt} \begin{bmatrix} \frac{\partial L}{\partial \dot{x}} \\ \frac{\partial L}{\partial \dot{\theta}} \end{bmatrix} - \begin{bmatrix} \frac{\partial L}{\partial x} \\ \frac{\partial L}{\partial \theta} \end{bmatrix} \\ &= \frac{d}{dt} \begin{bmatrix} (m_c + m_p)\dot{x} - m_p l \cos \theta \dot{\theta} \\ m_p l^2 \dot{\theta} - m_p l \dot{x} \cos \theta \end{bmatrix} - \begin{bmatrix} 0 \\ m_p l \dot{x} \sin \theta \dot{\theta} + m_p g l \sin \theta \end{bmatrix} \\ &= \begin{bmatrix} (m_c + m_p)\ddot{x} - m_p l (\cos \theta \ddot{\theta} - \sin \theta \dot{\theta}^2) \\ m_p l^2 \ddot{\theta} - m_p l (-\dot{x} \sin \theta \dot{\theta} + \ddot{x} \cos \theta) \end{bmatrix} - \begin{bmatrix} 0 \\ m_p l \dot{x} \sin \theta \dot{\theta} + m_p g l \sin \theta \end{bmatrix} \\ &= \begin{bmatrix} (m_c + m_p)\ddot{x} - m_p l \cos \theta \ddot{\theta} + m_p l \sin \theta \dot{\theta}^2 \\ m_p l^2 \ddot{\theta} - m_p l \cos \theta \ddot{x} - m_p g l \sin \theta \end{bmatrix} \end{aligned} \quad (4.15)$$

Including externally applied actuation on the cart as well as friction in both the cart and the pendulum, \mathbf{Q} can be written as

$$\mathbf{Q} = \begin{bmatrix} F \\ 0 \end{bmatrix} - \mathbf{B}(\dot{x}, \dot{\theta}) \quad (4.16)$$

where $\mathbf{B}(\dot{x}, \dot{\theta})$ denote cart and pendulum frictions and F denotes the input forces transferred from the motor, through the belt and to the cart.

Letting **equation (4.16)** equal **equation (4.15)** and isolating the accelerations, \ddot{x} and $\ddot{\theta}$, yields the vector function form:

$$\begin{bmatrix} F \\ 0 \end{bmatrix} - \mathbf{B}(\dot{x}, \dot{\theta}) = \underbrace{\begin{bmatrix} m_c + m_p & -m_p l \cos \theta \\ -m_p l \cos \theta & m_p l^2 \end{bmatrix}}_{\mathbf{M}(\theta)} \begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} + \underbrace{\begin{bmatrix} m_p l \sin \theta \dot{\theta}^2 \\ 0 \end{bmatrix}}_{\mathbf{C}(\theta, \dot{\theta})} + \underbrace{\begin{bmatrix} 0 \\ -m_p g l \sin \theta \end{bmatrix}}_{\mathbf{G}(\theta)} \quad (4.17)$$

where $\mathbf{B}(\dot{x}, \dot{\theta})$ indicates friction, $\mathbf{M}(\theta)$ includes all inertial properties, $\mathbf{C}(\theta, \dot{\theta})$ expresses centrifugal force and $\mathbf{G}(\theta)$ expresses gravitational force.

For convenience, this can be rearranged into:

$$\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \mathbf{M}(\theta)^{-1} \left(\begin{bmatrix} F \\ 0 \end{bmatrix} - \mathbf{B}(\dot{x}, \dot{\theta}) - \mathbf{C}(\theta, \dot{\theta}) - \mathbf{G}(\theta) \right) \quad (4.18)$$

This concludes the general structure of the mechanical model. The following section will elaborate on the friction terms.

4.3 Cart and pendulum frictions

The following is based on [12].

In **section 4.2**, the dynamics related to the cart and pendulum mechanics have been derived. This includes an external force caused by friction, which will be further elaborated in this section.

Two frictional components are present in the cart/pendulum system, namely the friction force affecting the movement of the cart and the friction torque affecting the pendulum. While the latter is only expected to depend on the angular velocity of the pendulum, the cart friction expresses frictional forces and torques in the following:

- Contact between the ball bearings and the rail.
- Contact between the belt and the gears in the power transmission.
- Internal friction in the motor.

In general, friction can be modelled solely as a function of the velocity, and for this project will be based on the friction components called Coulomb and viscous frictions, assuming the Stribeck-term is negligible, which is supported by the data in **appendix B.2**. Furthermore, effects from aerodynamic drag are considered negligible, with a brief analysis of the topic conducted in **appendix F**.

In this case, the Coulomb friction represents the necessary applied force in order to achieve acceleration, and is also referred to as *static friction*. Viscous friction, on the other hand, is the force produced due to surface contact at a specific velocity.

The friction is modelled as

$$\mathbf{B}(\dot{x}, \dot{\theta}) = \begin{bmatrix} B_c(\dot{x}) \\ B_p(\dot{\theta}) \end{bmatrix} = \begin{bmatrix} c_c \operatorname{sgn}(\dot{x}) + v_c \dot{x} \\ c_p \operatorname{sgn}(\dot{\theta}) + v_p \dot{\theta} \end{bmatrix} \quad (4.19)$$

for translational friction force in the power transmission and cart, B_c , and rotational friction torque in the pendulum joint, B_p , respectively. **Figure 4.6** illustrates these components separately and combined. Similar to the mass parameter m_c , the expression for $F_{c,v}$ will represent frictions from the power transmission in addition to the cart/rail friction.

An issue persists when using the sign function in simulation, as it tends to be numerically unstable at low velocities. This is due to the numerical value of the velocity never becoming exactly zero, causing the sign function to discontinuously jump between ± 1 .

In order to solve this, a *sigmoid* can be used to approximate the sign function. For this, the hyperbolic tangent, $\tanh(\cdot)$, is used, and the frictions thus become

$$B_c(\dot{x}) = c_c \tanh(k\dot{x}) + v_c \dot{x} \quad (4.20)$$

$$B_p(\dot{\theta}) = c_p \tanh(k\dot{\theta}) + v_p \dot{\theta} \quad (4.21)$$

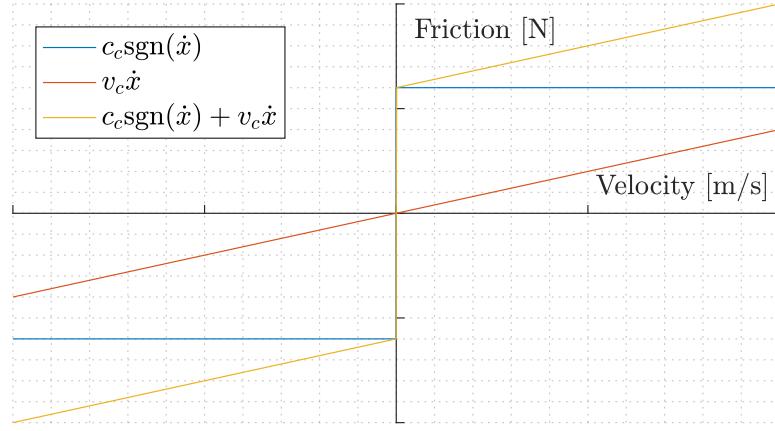


Figure 4.6: General friction model with Coulomb friction $c_c \text{sgn}(\dot{x})$, viscous friction $v_c \dot{x}$ and combined friction $c_c \text{sgn}(\dot{x}) + v_c \dot{x}$. Similarly for the pendulum friction torque.

with k being a sufficiently large parameter that determines the steepness of $\tanh(\cdot)$ through zero, and equivalently for the rotational friction. **Figure 4.6** shows a comparison of the original and the approximate expressions of the combined friction.

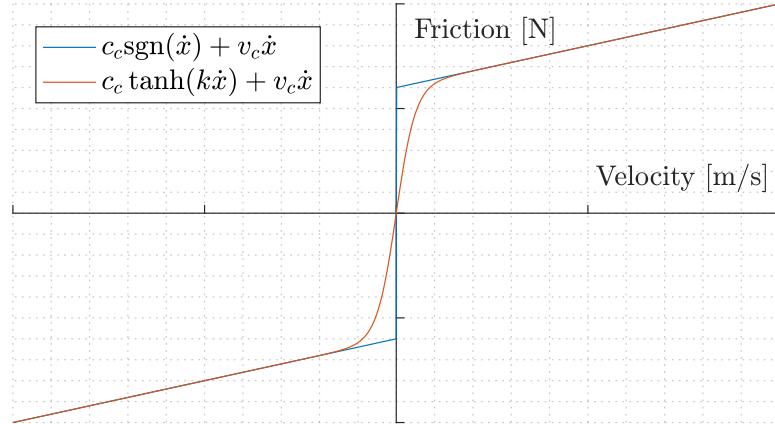


Figure 4.7: Friction model $c_c \text{sgn}(\dot{x}) + v_c \dot{x}$ and a continuous approximation, $c_c \tanh(k \dot{x}) + v_c \dot{x}$.

4.3.1 Translational friction in the cart

In order to estimate the coefficients for the translational friction, a series of tests are presented in **appendix B**. In this, the data shows that the viscous coefficient for the cart is approximately zero. However, the data regarding the translational Coulomb friction are inconsistent and are therefore disregarded.

As an alternative, the cart has instead been subjected to a pulse input, starting from either end of the rail and accelerated towards the middle before stopping due to friction. Using this data and the dynamic model, an optimisation routine using MATLAB function `fmincon` has been set up.

In this, the model in **equation (4.18)** is subjected to the same input and then compared with the data in order to determine proper parameters, namely cart mass m_c , Coulomb friction when moving right, $c_{c,R}$, and when moving left, $c_{c,L}$. The directional dependency is chosen as per observation of the physical test setup, as described in **section 2.4**.

The mass is included as it cannot be directly measured due to physical restrictions in the test setup.

The cost function used for the optimisation is the Root Mean Square Error (RMSE) of the cart position, summed for each direction:

$$C_{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_{c_i,R} - \bar{x}_{c_i,R})^2} + \sqrt{\frac{1}{n} \sum_{i=1}^n (x_{c_i,L} - \bar{x}_{c_i,L})^2} \quad (4.22)$$

with \bar{x}_c indicating the simulation while R and L indicates direction.

With this cost function, the optimisation problem is formulated as:

$$\min_{m_c, c_{c,R}, c_{c,L}} C_{RMSE} \quad (4.23)$$

subject to $m_c, c_{c,R}, c_{c,L} \in [1, 5]$

with initial values being three for all the variables.

The solution to the optimisation problem is found to be:

$$m_c = 3.8495 \quad (4.24)$$

$$c_{c,R} = 3.6192 \quad (4.25)$$

$$c_{c,L} = 3.1041 \quad (4.26)$$

with the data and simulation corresponding to the solution is shown in **figure 4.8**.

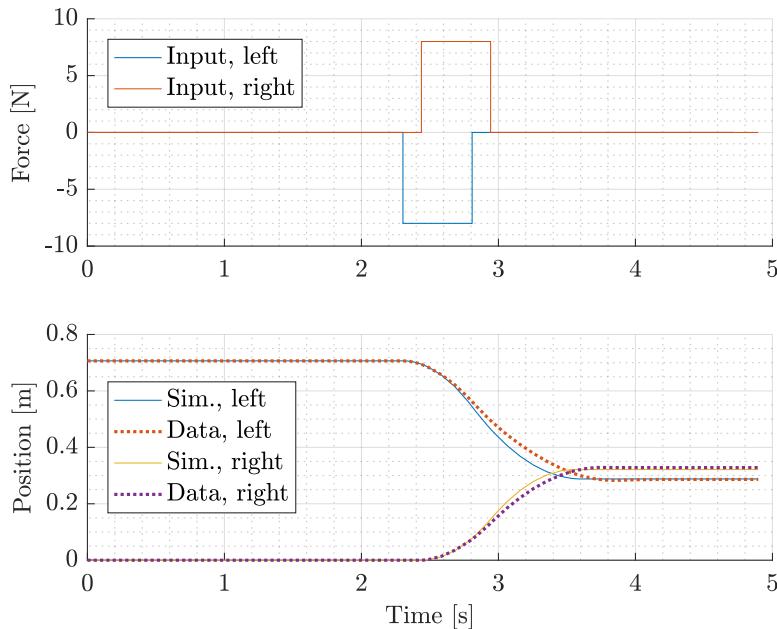


Figure 4.8: Data and simulation of the optimisation solution. Top figure shows the input signals in each direction, and the bottom figure shows the real data and simulation with the found parameters.

Conclusively, some differences between the test data and the simulation persist, but the found coefficients are assumed to suffice. The final form of the friction model thus becomes:

$$B_c(\dot{x}) = c_c \tanh(k\dot{x}), \quad c_c = \begin{cases} c_{c,R} & \text{if } \dot{x} \geq 0 \\ c_{c,L} & \text{if } \dot{x} < 0 \end{cases} \quad (4.27)$$

4.3.2 Rotational friction in the pendulum joint

For the pendulum friction model, Coulomb and viscous friction coefficients are needed. These coefficients, denoted c_p and v_p , are found by the methods described in [appendix E](#).

Figure 4.9 shows the method for finding c_p , as the pendulum is manually adjusted to find the maximum angle ϕ at which it remains stationary, using mass $m_p = 26$ g and by computing the torque, τ_c , produced by gravity.

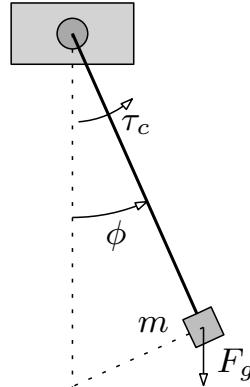


Figure 4.9: Pendulum system with variables and parameters used for Coulomb friction estimation.

In order to find v_p , the pendulum has been raised manually, then released and let oscillate until coming to a full stop. From this underdamped, harmonic motion, the coefficient is determined by fitting the peaks to an exponentially decaying function, as described in [appendix E.2](#) and shown in **figure 4.10**.

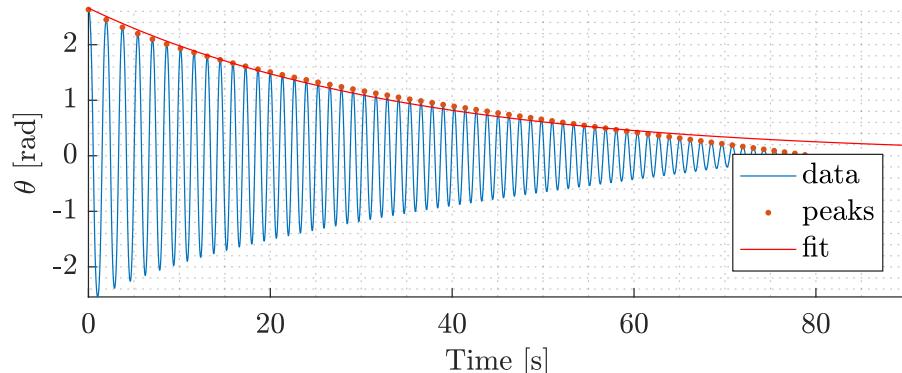


Figure 4.10: Data and exponential fit used for determining the viscous friction coefficient.

The two coefficients have been found to be:

$$\begin{aligned} c_p &= 0.0030 \\ v_p &= 0.01479 \end{aligned} \tag{4.28}$$

Since the response in **figure 4.10** are also subject to the Coulomb torque, the individually found values are adjusted to fit this.

Using the values

$$\begin{aligned} c_p &= 0.0035 \\ v_p &= 0.00067 \end{aligned} \tag{4.29}$$

yields the result shown in **figure 4.11**.

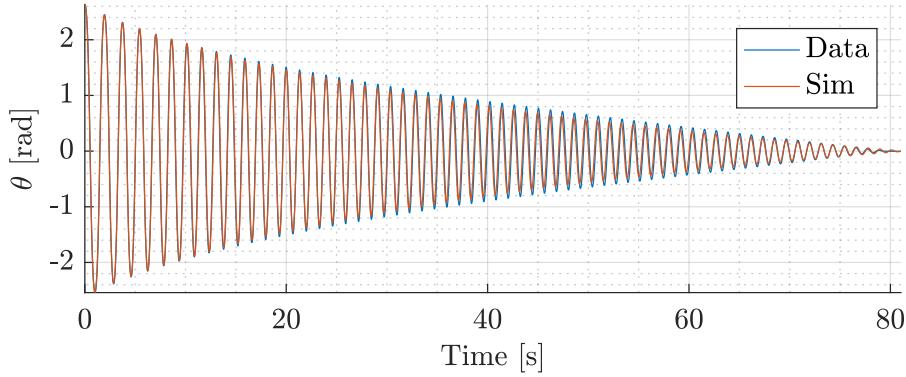


Figure 4.11: Response of the pendulum swinging freely using adjusted friction coefficients.

It is seen that the viscous coefficient is lowered significantly as the Coulomb friction torque is added to the model.

Conclusively, the friction model for the entire system is

$$\mathbf{B}(\dot{x}, \dot{\theta}) = \begin{bmatrix} B_c(\dot{x}) \\ B_p(\dot{\theta}) \end{bmatrix} = \begin{bmatrix} c_c(\dot{x}) \operatorname{sgn}(\dot{x}) \\ c_p \operatorname{sgn}(\dot{\theta}) + v_p \dot{\theta} \end{bmatrix} \quad (4.30)$$

with

$$c_p = 0.0035 \quad (4.31)$$

$$c_v = 0.00067 \quad (4.32)$$

$$c_c(\dot{x}) = \begin{cases} c_{c,R} = 3.6192 & \text{if } \dot{x} \geq 0 \\ c_{c,L} = 3.1041 & \text{if } \dot{x} < 0 \end{cases} \quad (4.33)$$

4.4 State space formulation

For convenience, the mechanical model, presented at **equation (4.18)** can be formulated as a state space system.

Considering the system state vector $\mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4]^T = [x \ \theta \ \dot{x} \ \dot{\theta}]^T$, the nonlinear state space system representation becomes:

$$\begin{aligned} \dot{\mathbf{x}} &= \begin{bmatrix} \dot{x} \\ \dot{\theta} \\ \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{\theta} \\ \mathbf{M}(\theta)^{-1} \left(\begin{bmatrix} F \\ 0 \end{bmatrix} - \mathbf{B}(\dot{x}, \dot{\theta}) - \mathbf{C}(\theta, \dot{\theta}) - \mathbf{G}(\theta) \right) \end{bmatrix} \\ &= \begin{bmatrix} \dot{x} \\ \dot{\theta} \\ -\alpha(\mathbf{x}) - \beta(\mathbf{x}) \\ -\gamma(\mathbf{x}) - \delta(\mathbf{x}) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ (m_c + m_p \sin^2 \theta)^{-1} \\ \cos \theta (l(m_c + m_p \sin^2 \theta))^{-1} \end{bmatrix} F \end{aligned} \quad (4.34)$$

where system input $u = F$ and

$$\alpha(\mathbf{x}) = \frac{c_c \tanh(k\dot{x}) + m_p l \sin \theta \dot{\theta}^2}{m_c + m_p \sin^2 \theta} \quad (4.35)$$

$$\beta(\mathbf{x}) = \cos \theta \frac{c_p \tanh(k\dot{\theta}) + v_p \dot{\theta} - m_p g l \sin \theta}{l(m_c + m_p \sin^2 \theta)} \quad (4.36)$$

$$\gamma(\mathbf{x}) = \cos \theta \frac{c_c \tanh(k\dot{x}) + m_p l \sin \theta \dot{\theta}^2}{l(m_c + m_p \sin^2 \theta)} \quad (4.37)$$

$$\delta(\mathbf{x}) = (m_c + m_p) \frac{c_p \tanh(k\dot{\theta}) + v_p \dot{\theta} - m_p g l \sin \theta}{m_p l^2 (m_c + m_p \sin^2 \theta)} \quad (4.38)$$

and the system output is

$$\mathbf{y} = h(\mathbf{x}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{x} \quad (4.39)$$

For simplicity and later reference, this can be compactly written as

$$\begin{aligned} \dot{\mathbf{x}} &= \begin{bmatrix} \dot{x} \\ \dot{\theta} \\ f_c(\mathbf{x}) \\ f_p(\mathbf{x}) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g_c(\mathbf{x}) \\ g_p(\mathbf{x}) \end{bmatrix} u \\ &= f(\mathbf{x}) + g(\mathbf{x})u \end{aligned} \quad (4.40)$$

4.5 Simulink implementation

This section concerns the MATLAB/Simulink-implementation of the system. The model is based on several files:

- `swingUpModel.slx`
The main implementation of the system model where the simulation is carried out using an ordinary differential equation solver.
- `params.m`
Initialisation script that holds the values of all parameters needed to run the simulation along with default values for some simulation settings.
- `swingUp.m`
Script used to call `params.m` and change simulation settings from their defaults.

The remainder of this section will focus on the content of `swingUpModel.slx`. The top-level of the file is shown in **figure 4.12**.

Starting from the left, the block *Input* contains several Simulink source blocks (e.g. steps, pulse trains) which is used to simulate the system response of open-loop inputs.

swingUpModel.slx

This is the main implementation of the system model including motor dynamics, mechanics, endstops, sampling, filtering and control etc.

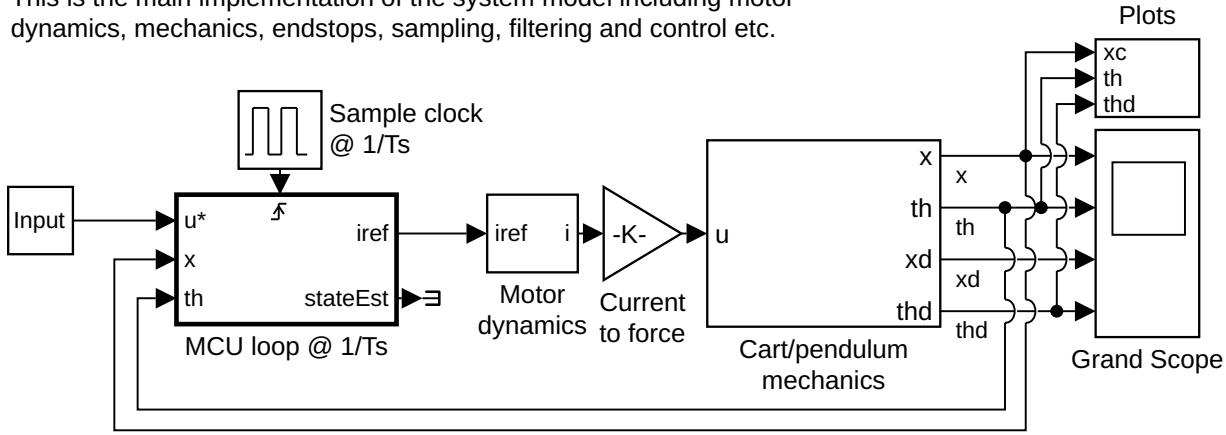


Figure 4.12: Simulink model file swingUpModel.slx, top-level.

The next block, *MCU loop*, is a triggered subsystem which is run at each rising edge of the output from the block *Sample clock*, a pulse train with a period of $T_s = 0.005$ ms. Its main purpose of *MCU loop* is to contain the data filtering and control algorithms.

The triggering is used to emulate sampling effects on the data fed into the block as well as on the control output. Additionally, though not shown, this block emulates quantisation of the inputs x and θ . The block in its entirety will be further elaborated in [section 9.1](#).

The block *Motor dynamics* contains the first-order approximated transfer function described in [section 4.1](#) and saturation based on the analysis in [appendix A](#). The gain in *Current to force* is the product of the torque coefficient of the motor, k_t , and the inverse of the pulley radius, r^{-1} .

The subsystem *Cart/pendulum mechanics* contains the core of the simulation, namely the dynamics described in [section 4.2](#). The content of this subsystem is shown in [figure 4.13](#).

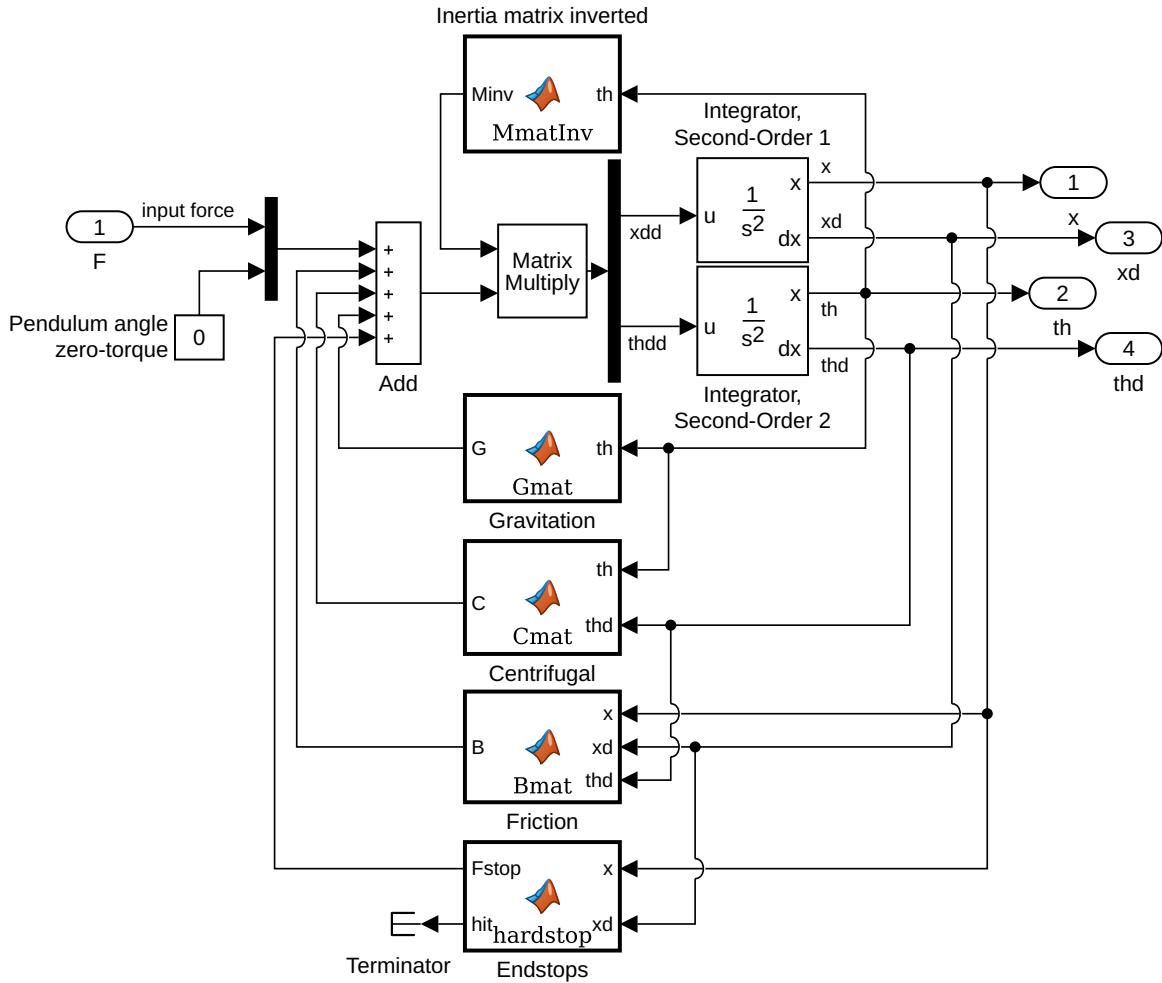


Figure 4.13: Simulink model file *swingUpModel.slx*, cart/pendulum mechanics subsystem.

As such, this is **equation (4.18)**, split up into its subcomponents, followed by double integration. In addition to this, the block *Endstops* is an implementation of the barriers at each end of the rail, modelled as two very stiff spring-damper systems with coefficient found by trial-and-error.

4.5.1 Data visualisation

To the far right of **figure 4.12**, the two blocks named *Grand Scope* and *Plots* serve the purpose of data visualisation.

Grand Scope is the visualisation of the states over time. *Plots* is a graphical animation of the cart and pendulum as well as an animated phase-plot of $(\theta, \dot{\theta})$, both of which are updated along with the progress of the simulation. This is made as a way to intuitively verify the behaviour of the system while running a simulation.

Snapshots of the two animations are shown in **figures 4.14** and **4.15**.

For real-time updates on these plots, an item from the MATLAB Community File Exchange called *Real-time Pacer* is used, as it slows down the simulation speed to approximately real-time. [13]

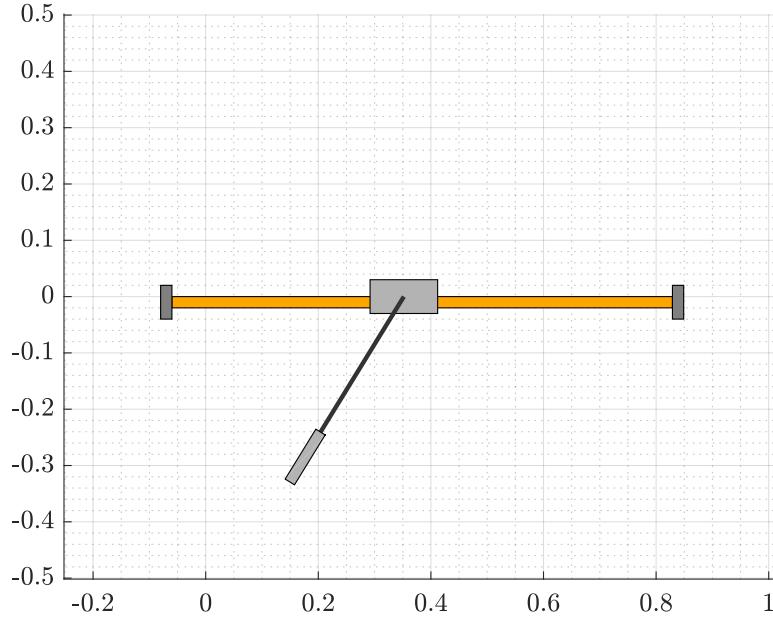


Figure 4.14: Snapshot of the system animation, mid-simulation.

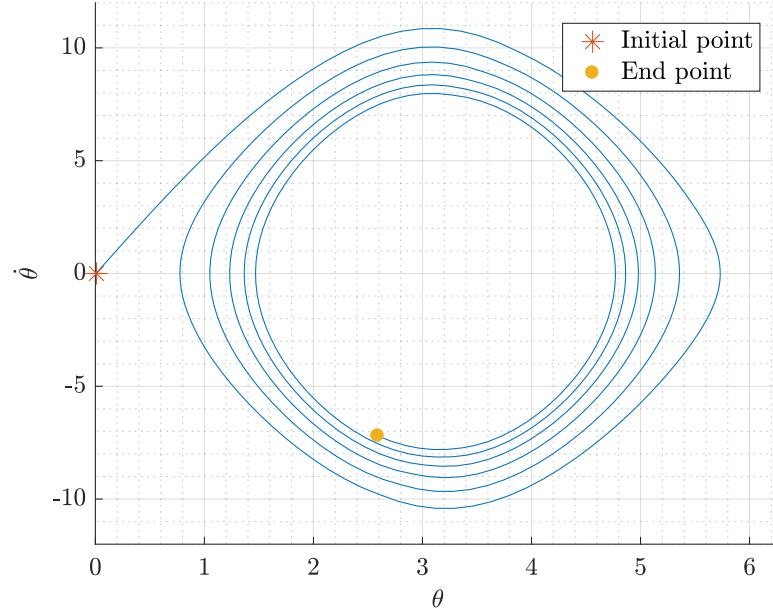


Figure 4.15: Snapshot of the pendulum phase animation, mid-simulation.

4.6 Model verification

In this section, the model with the found parameters will be tested in order to evaluate to which degree it represents the physical system. This will be done in two different scenarios:

Input Cart acceleration/deceleration with pendulum started in its stable equilibrium. Two successive input force pulses, in opposite directions, are applied to the open-loop system.

Oscillation Free pendulum motion from the vicinity of the unstable equilibrium. Without any actuation input, both the pendulum and the cart are allowed to move freely.

In both cases, the system position responses are logged along with the applied input (if any). Then, the simulation model is initialised with the same initial conditions and is subjected to the same input, after which the responses are compared.

The first scenario is considered in order to verify the effect of actuation input to the entire system, while the second is in order to check the coupled effects between the pendulum and the cart under a highly dynamic pendulum motion.

Figure 4.16 shows the results of both scenarios with the parameters initially determined, as described previously in this chapter. For the input test, it is seen that the model shows almost the same response as the data, except for a slight deviation during movement in the negative x-direction. The coupled response from the pendulum also responds according to the data.

However, for the oscillation test, a slight deviation of about five centimetres is seen in the cart position, as the modelled cart tends to drift a bit to the left. Additionally, the physical pendulum motion is initially more damped than the model.

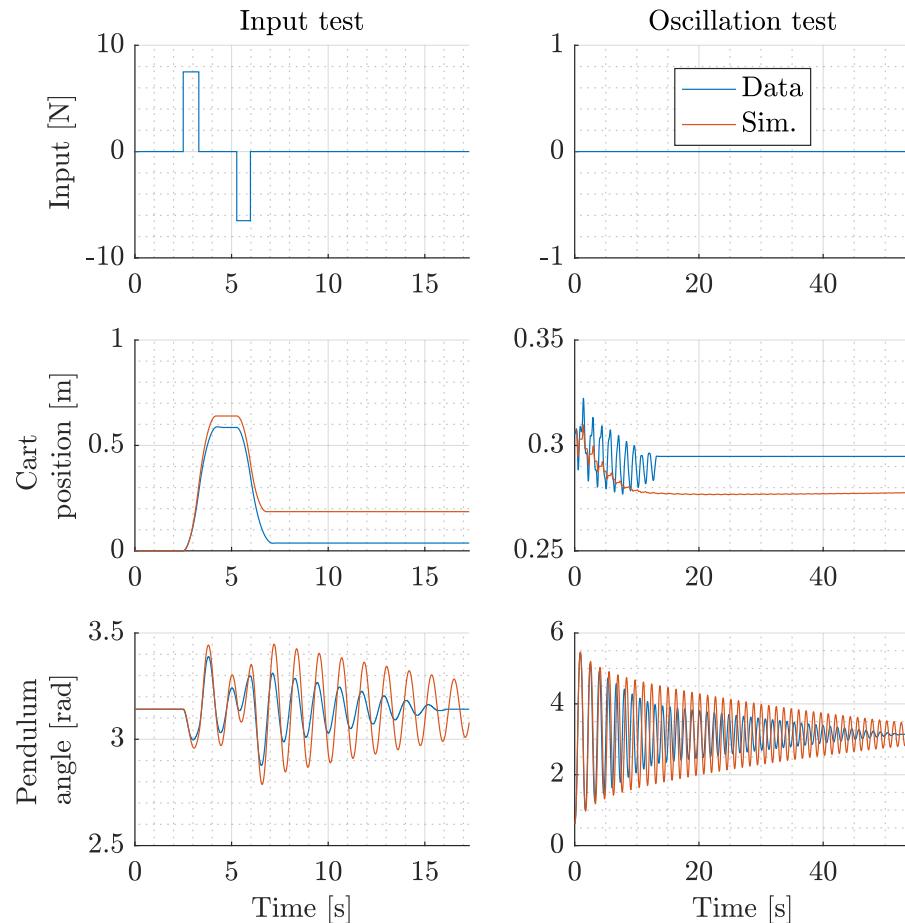


Figure 4.16: Model verification test results using the initially found parameters. The three plots on the left-hand side are from the input test, and the other three are from the oscillation test.

In order to further investigate the pendulum response data, **figure 4.17** shows two cases of the oscillation test; with and without the transmission belt attached. Initially, the pendulum tends to be damped quicker with the belt attached, but after about 22 seconds, the oscillations are synchronous.

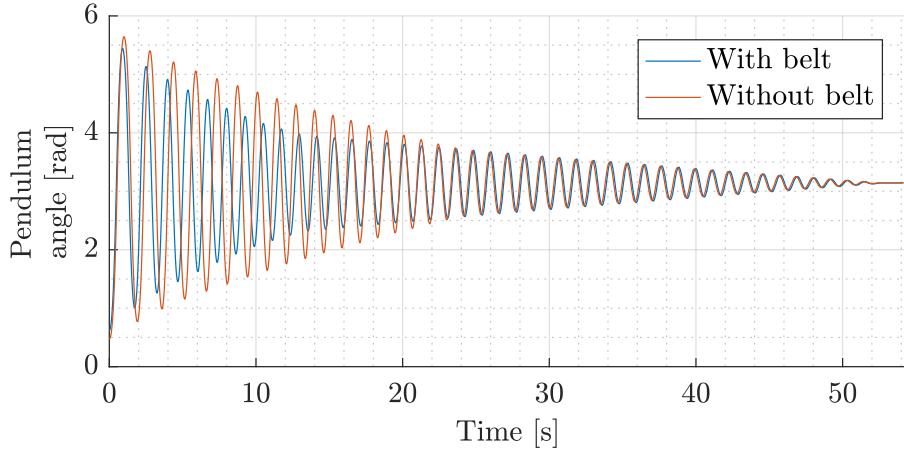


Figure 4.17: Comparison of two oscillation tests; with and without the belt attached to the cart.

Observations of the cart motion during the two tests show that the change in damping occurs when the centrifugal force decreases beyond the point where it is unable to move the cart, i.e. at about 22 seconds without the belt and 13 seconds with the belt. This is also seen in **figure 4.16** on the middle-right subfigure. The test without the belt does not have corresponding cart position data, as this is measured by the encoder on the motor.

Given the results in **figure 4.16**, some adjustments to the system parameters are made, which are presented in **table 4.1**

	Old	New	Unit
m_c	3.8495	4.75	[kg]
$c_{c,R}$	3.6192	3.4	[N]
$c_{c,L}$	3.1041	2.45	[N]
c_p	0.0038	0.0038	[Nm]
v_p	0.00067	0.00067	[Nms/rad]

Table 4.1: Parameter adjustments.

These new parameter values produce the results shown in **figure 4.18**, where it is seen that the simulated motion of both the cart and the pendulum is rather accurate with respect to the data for the input test. However, for the oscillation test, the cart drifts about 6 cm to the left as opposed to just 1.5 cm in **figure 4.16**, and the issue with the pendulum damping persists. The new parameters will replace the ones previously found. It is notable, however, that the mass is significantly higher and the friction are lower. This may be related to that fact that m_c does not only account for the cart mass, but also the inertias in the pulleys and the motor.

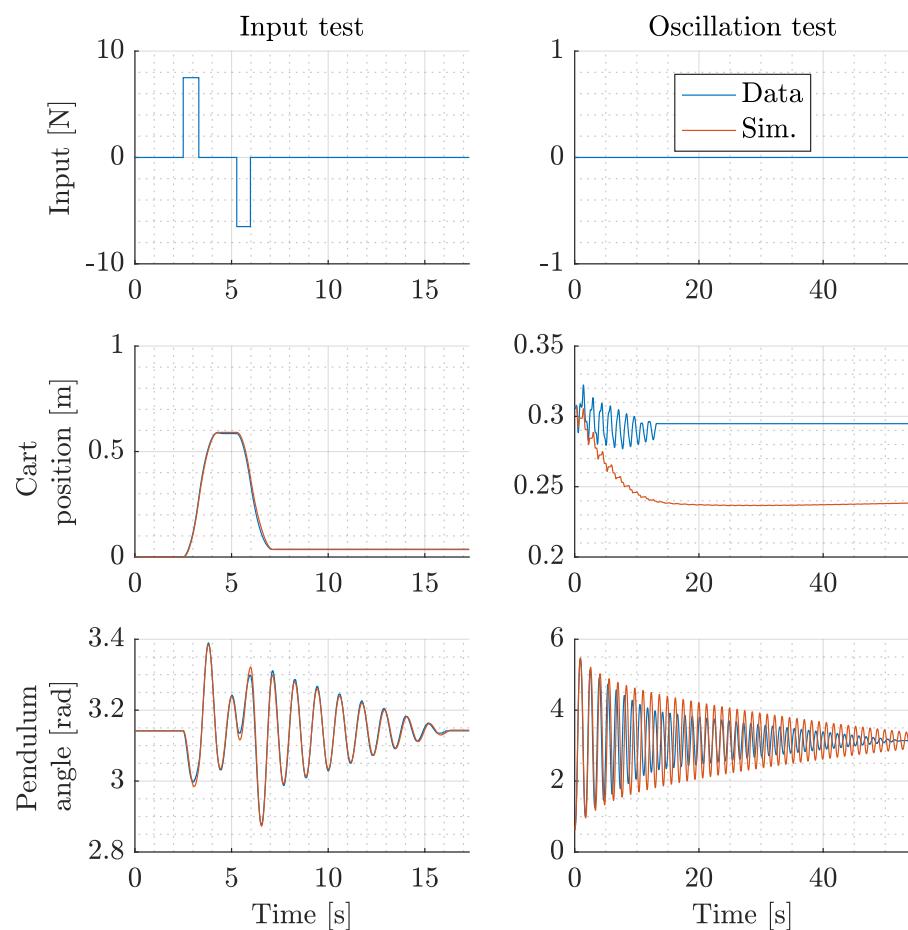


Figure 4.18: Model verification test using the adjusted parameters.

State Estimation 5

The physical test setup is equipped with relative encoders attached to the motors, and are therefore only capable of measuring displacement relative to the initial position.

Since the full state-vector is needed for the controllers designed in **chapters 7 and 8** the topic of state estimation is relevant and will be covered in this chapter, specifically to estimate the velocities, \dot{x} and $\dot{\theta}$.

5.1 Numerical derivative

Given the fact that the measured states are positions x and θ , the simplest approach to estimating their first time derivatives are by doing so numerically.

For a function of time t , $f(t)$, the derivative is (backwards) defined by:

$$\frac{df(t)}{dt} = \lim_{\Delta t \rightarrow 0} \frac{f(t) - f(t - \Delta t)}{\Delta t} \quad (5.1)$$

Given a sampling time T_s , the states \dot{x} and $\dot{\theta}$ can thus be approximated by:

$$\left. \begin{array}{l} \dot{x}(t) \approx \frac{x(t) - x(t - T_s)}{T_s} = \frac{x_k - x_{k-1}}{T_s} = \dot{x}_k \\ \dot{\theta}(t) \approx \frac{\theta(t) - \theta(t - T_s)}{T_s} = \frac{\theta_k - \theta_{k-1}}{T_s} = \dot{\theta}_k \end{array} \right\} t \in T_s \cdot \mathbb{Z}^+ \quad (5.2)$$

However, one issue arises when computing the derivative this way. Due to the resolution of the encoders, quantization of the data occurs. This is shown in **figure 5.1**, which illustrates the final few oscillations of the pendulum when let to swing freely from $\theta \approx 0$ with the resolutions described in **section 2.3**.

Quantization is an issue regarding the numerical derivative as it will limit the resolution of the velocity estimate based on the quantisation step size and the sampling time. This is shown in **section 5.3**, where the smallest non-zero value of the numerical derivative is

$$\min \Delta \dot{\theta} = \frac{\min \Delta \theta}{T_s} \approx 0.6283 \frac{\text{rad}}{\text{s}} \quad (5.3)$$

given $T_s = 0.005$ seconds.

The equivalent value for \dot{x} , although not shown, is

$$\min \Delta \dot{x} = \frac{\min \Delta x}{T_s} \approx 0.0176 \frac{\text{m}}{\text{s}} \quad (5.4)$$

In order to address this, the data can subsequently be filtered by e.g. a moving average (MA) filter. MA-filters consider a sample window and output the average of the samples:

$$\dot{\theta}_{k,MA} = \frac{1}{n} \sum_{i=0}^{n-1} \dot{\theta}_{k-i} \quad (5.5)$$

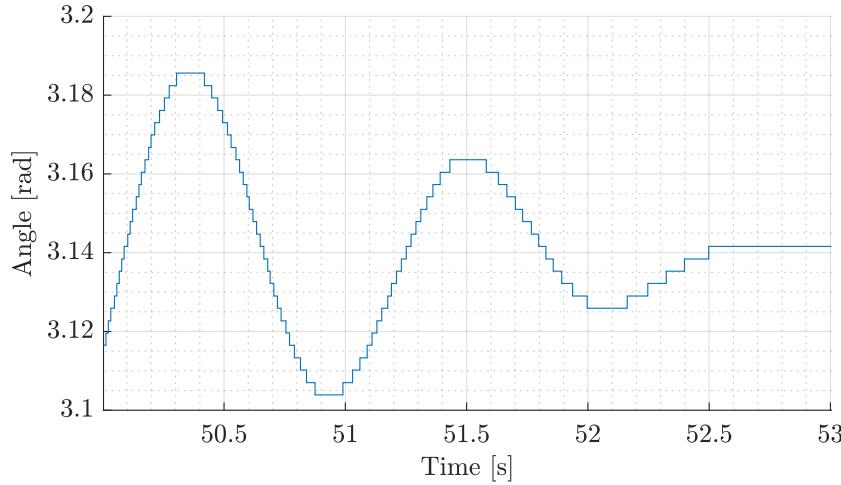


Figure 5.1: Raw pendulum angle data which shows the resolution of the measured signal.

where n is the window, i.e. the number of samples to be used in the averaging. State \dot{x}_k is found equivalently. The effect of the moving average is also seen in [section 5.3](#) for window sizes of both 10 and 50. It is seen that the 10-sample MA increases the resolution while introducing some phase delay. The 50-sample MA increases the resolution even further, but also introduces a much larger phase delay.

5.2 Extended Kalman Filter

The following is based on [14].

Another option in terms of state estimation is the Kalman filter, which is a linear prediction-correction filter that uses a system model to predict the state vector based on the previous time step and the system input, and then corrects the current estimate given a measurement.

For highly nonlinear systems, such as the one considered in this project, the linear Kalman filter becomes insufficient. Instead, an EKF is used, as it uses a nonlinear model to propagate the system states and output, and a linearisation in the most recent state vector estimate for the matrix computations.

The discrete-time system dynamics with state noise w_k and measurement noise v_k are expressed by

$$\mathbf{x}_{k+1} = f_k(\mathbf{x}_k, u_k) + w_k, \quad w_k \in \text{NID}(\mathbf{0}, \mathbf{Q}_k) \quad (5.6)$$

$$\mathbf{y}_{k+1} = h_k(\mathbf{x}_k) + v_k, \quad v_k \in \text{NID}(\mathbf{0}, \mathbf{R}_k) \quad (5.7)$$

where $f_k(\mathbf{x}_k, \mathbf{u}_k)$ is a discretisation of the continuous system dynamics in [section 4.4](#) by the forward Euler method, i.e.

$$\begin{aligned} f_k(\mathbf{x}_k, u_k) &= \mathbf{x}_k + T_s \dot{\mathbf{x}}(\mathbf{x}_k) \\ &= \mathbf{x}_k + T_s (f(\mathbf{x}_k) + g(\mathbf{x}_k)u_k) \end{aligned} \quad (5.8)$$

$$h_k(\mathbf{x}_k) = h(\mathbf{x}_k) \quad (5.9)$$

with $f(\mathbf{x}_k)$, $g(\mathbf{x}_k)$ and $h(\mathbf{x}_k)$ defined in [equation \(4.40\)](#), page 21.

The matrices \mathbf{Q}_k and \mathbf{R}_k are the covariance matrices for the model accuracy and measurement noise, respectively.

Using this model, the Kalman filter is applied in two steps, namely a measurement update (correction) and a time update (prediction).

5.2.1 Correction step

The correction step uses information from new samples of \mathbf{y}_k and u_k in addition to the state vector prediction made during the previous sample, $\hat{\mathbf{x}}_{k|k-1}$:

$$\hat{\mathbf{y}}_{k|k-1} = h_k(\hat{\mathbf{x}}_{k|k-1}) \quad (5.10)$$

This is used to compute the system output residual

$$\tilde{\mathbf{y}}_{k|k-1} = \mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1} \quad (5.11)$$

The Kalman gain, defined by

$$\mathbf{K}_k \triangleq \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (5.12)$$

where $\mathbf{P}_{k|k-1}$ is the covariance matrix and \mathbf{H}_k is the linearised output function. The Kalman gain is used to correct the state estimate by:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_{k|k-1} \quad (5.13)$$

Finally, the covariance matrix is computed by

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T \quad (5.14)$$

5.2.2 Prediction step

Following the correction step, a prediction of the state vector is made for the next sample, based on the model, current state estimate and current input, as

$$\hat{\mathbf{x}}_{k+1|k} = f_k(\hat{\mathbf{x}}_{k|k}, u_k) \quad (5.15)$$

followed by a prediction of the covariance matrix:

$$\mathbf{P}_{k+1|k} = \mathbf{F}_k \mathbf{P}_{k|k} \mathbf{F}_k^T + \mathbf{Q}_k \quad (5.16)$$

where \mathbf{F}_k is the linearised system dynamics function.

The predicted values are then saved for the next iteration of the filter, namely when the system is sampled again T_s time units later:

$$k+1|k \rightarrow k|k-1 \quad \begin{cases} \hat{\mathbf{x}}_{k+1|k} \\ \mathbf{P}_{k+1|k} \end{cases} \rightarrow \begin{cases} \hat{\mathbf{x}}_{k|k-1} \\ \mathbf{P}_{k|k-1} \end{cases} \quad (5.17)$$

which is then used in the correction step at the next sample.

5.2.3 Jacobian computation

The linearised system equations, namely matrices \mathbf{H}_k and \mathbf{F}_k , must be derived analytically or otherwise found in order to be used in the filter.

These matrices are given by their Jacobians, defined by

$$\mathbf{H}_k \triangleq \frac{\partial h_k(\mathbf{x})}{\partial \mathbf{x}^T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (5.18)$$

$$\begin{aligned} \mathbf{F}_k \triangleq \frac{\partial f_k(\mathbf{x}_k)}{\partial \mathbf{x}_k^T} &= \frac{\partial}{\partial \mathbf{x}_k^T} (\mathbf{x}_k + T_s (f(\mathbf{x}_k) + g(\mathbf{x}_k)u_k)) \\ &= \mathbf{I} + \frac{\partial}{\partial \mathbf{x}_k^T} (T_s (f(\mathbf{x}_k) + g(\mathbf{x}_k)u_k)) \end{aligned} \quad (5.19)$$

As evaluating [equation \(5.19\)](#) results in a very large matrix expression, which must be re-evaluated and re-implemented after every change to the system equations, an alternative approach is taken, namely numerical computation of the Jacobian entries:

$$\mathbf{F}_k = \begin{bmatrix} \frac{\partial f_k(\mathbf{x}_k)}{\partial x_{k,1}} & \dots & \frac{\partial f_k(\mathbf{x}_k)}{\partial x_{k,n}} \end{bmatrix}, \quad n = \dim(\mathbf{x}_k) \quad (5.20)$$

with

$$\frac{\partial f_k(\mathbf{x}_k)}{\partial x_{k,i}} = \frac{f_k(\mathbf{x}_k^{pert}) - f_k(\mathbf{x}_k)}{\epsilon}, \quad \mathbf{x}_k^{pert} = \begin{cases} \mathbf{x}_{k,j} + \epsilon \text{ if } j = i \\ \mathbf{x}_{k,j} \text{ otherwise} \end{cases}, \quad j = 1, \dots, n \quad (5.21)$$

This effectively perturbs the system equations by ϵ , one entry of the state vector at a time.

5.3 Comparison

[Figure 5.2](#) shows a simulation comparison of the estimation methods described in this chapter.

The uppermost subfigure shows several oscillations from a simulation of the pendulum swinging freely. The two smaller subfigures are close-ups of the left and right boxes marked on the uppermost one, to see the estimates of both large and small signals.

The figure shows that the numerical estimate without averaging does not suffice in estimating the velocities. Applying a moving average filter improves the resolution as the windows size increases, but at the cost of added phase delay, which is undesirable for feedback control.

Finally, the extended Kalman filter shows the best accuracy. However, one thing must be taken into account, namely that the EKF uses a "perfect" model in this case, as it is identical to the simulation model. In reality, this model will have inaccuracies compared to the real system.

In order to address this, the \mathbf{Q}_k matrix must be tuned to account for the model inaccuracies. The \mathbf{R}_k matrix must likewise be tuned in order to account for measurement noise.

Initially, \mathbf{Q}_k will be identity and tuned as found necessary. \mathbf{R}_k , on the other hand can be set according to the variance of the measurements. [Appendix G](#) shows that the quantisation is much more significant than any measurement noise, implying the variance of the signal is close to zero.

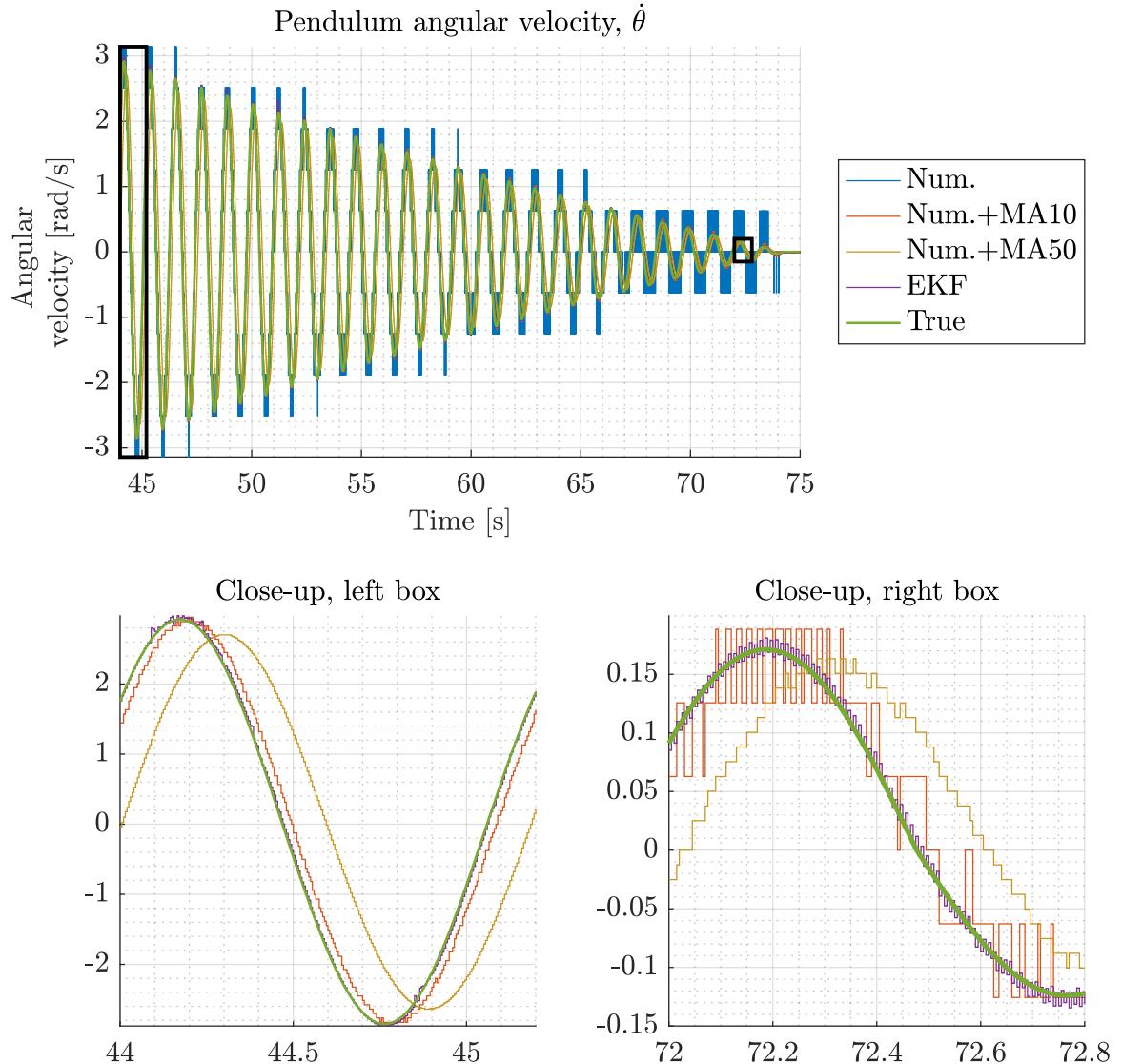


Figure 5.2: Comparison of the different estimation methods. The top figure shows the last part of a simulation, and the bottom two are close-ups of the beginning and end. The different estimation methods are; the Numerical derivative (Num.), numerical derivative with 10-sample moving average filter (Num.+MA10), numerical derivative with 50-sample moving average filter (Num.+MA50), EKF and finally the model (True). Note that the raw numerical derivative has been omitted on the close-ups for readability.

The EKF will be tested and tuned in **section 9.2**.

Control Strategy 6

This chapter will describe general control strategy for this project. The chosen approach is based on a two-step control strategy, such that one controller will swing up the pendulum, and another will stabilise it. The objective of the control is thus:

- 1) Set the pendulum in oscillation from initial state $\mathbf{x}_0 = [x_0 \ \theta_0 \ \dot{x}_0 \ \dot{\theta}_0]^T = [0 \ \pi \ 0 \ 0]^T$ and increase its mechanical energy to reach within $\theta = \pm 10^\circ$ while keeping $|x^* - x| < 10$ cm, as stated in [section 3.1.1](#).
- 2) Bring the pendulum from this oscillation to the control references, $\mathbf{x}^* = [x^* \ 0 \ 0 \ 0]^T$.

For each step, multiple controller types will be tested and compared. The controller for the first step will be referred to as the *swing-up controller* while the second one will be called the *stabilising controller*. The necessity of two controllers is based on the idea that the swing-up controller must carefully increase the mechanical energy while keeping the cart from hitting the physical end-stops. On the other hand, the stabilising controller must keep the pendulum in an upright position and the cart at its reference, and is expected to be more aggressive.

An overview of the control strategy is shown in [figure 6.1](#).

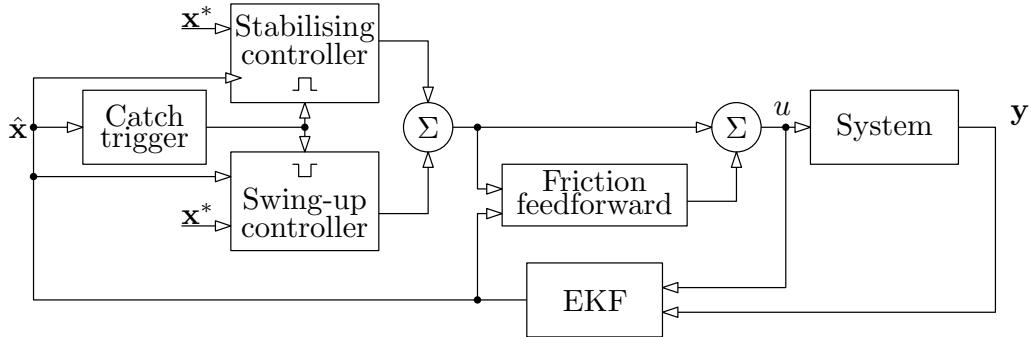


Figure 6.1: Control system overview. The icons in the controller blocks indicate when each is active based on the output of Catch trigger.

Before proceeding to the design, a few definitions and lemmas regarding system stability will be presented for later reference in [section 6.1](#). Then, an analysis of the mechanical energy of the pendulum and the concept of heteroclinic orbits will be carried out in [section 6.2](#).

As seen in [figure 6.1](#), the cart friction $B_c(\dot{x})$ is feedforwarded due to its size and discontinuous nature. This will be further described in [section 6.3](#).

The block in [figure 6.1](#) called *Catch trigger* is an algorithm that decides which one of the controllers should be active, and its functionality will be described in [section 6.4](#).

The designs of the swing-up and stabilising controllers are described in [chapter 7](#) and [chapter 8](#), respectively.

6.1 Stability

In this section, a few notions of stability and system convergence will be listed for later reference, as they will be used in the design process described in **chapter 7** (Control Design: Swing-up) and **chapter 8** (Control Design: Stabilisation).

Considering a general system of the form

$$\dot{x} = f(x) \quad (6.1)$$

the notion of stability is defined in the following along with Lyapunov's and LaSalle's theorems that will be used to prove stability.

Definition 6.1.1 (Stability). [4]

The equilibrium point $x = 0$ of **equation (6.1)** is

- stable if, for each $\epsilon > 0$, there is $\delta = \delta(\epsilon) > 0$ such that

$$\|x(0)\| < \delta \Rightarrow \|x(t)\| < \epsilon, \quad \forall t \geq 0 \quad (6.2)$$

- unstable if it is not stable.
- asymptotically stable if it is stable and δ can be chosen such that

$$\|x(0)\| < \delta \Rightarrow \lim_{t \rightarrow \infty} x(t) = 0 \quad (6.3)$$

Theorem 6.1.1 (Lyapunov's stability theorem). [4]

Let $x = 0$ be an equilibrium point for **equation (6.1)** and $D \subset \mathbb{R}^n$ be a domain containing $x = 0$. Let $V : D \rightarrow \mathbb{R}$ be a continuously differentiable function such that

$$V(0) = 0 \text{ and } V(x) > 0 \text{ in } D - \{0\} \quad (6.4)$$

$$\dot{V}(x) \leq 0 \text{ in } D \quad (6.5)$$

Then, $x = 0$ is stable. Moreover, if

$$\dot{V}(x) < 0 \text{ in } D - \{0\} \quad (6.6)$$

then $x = 0$ is asymptotically stable.

Theorem 6.1.2 (LaSalle's theorem). [4]

Let $\Omega \subset D$ be a compact set that is positively invariant with respect to **equation (6.1)**. Let $V : D \rightarrow \mathbb{R}$ be a continuously differentiable function such that $\dot{V}(x) \leq 0$ in Ω . Let E be the set of all points in Ω where $\dot{V}(x) = 0$. Let M be the largest invariant set in E . Then every solution starting in Ω approaches M as $t \rightarrow \infty$.

6.2 Mechanical energy and heteroclinic orbits

The swing-up control strategy is based on the mechanical energy of the system, and thus the reference shown in **figure 4.5** on page 14, defining the system and its parameters, is revised. The potential energy reference, u_{ref} , is placed where the pendulum is at rest in its stable equilibrium point, as depicted in **figure 6.2**, namely

$$u_{ref} = m_p g l \quad (6.7)$$

in order to avoid negative potential energy.

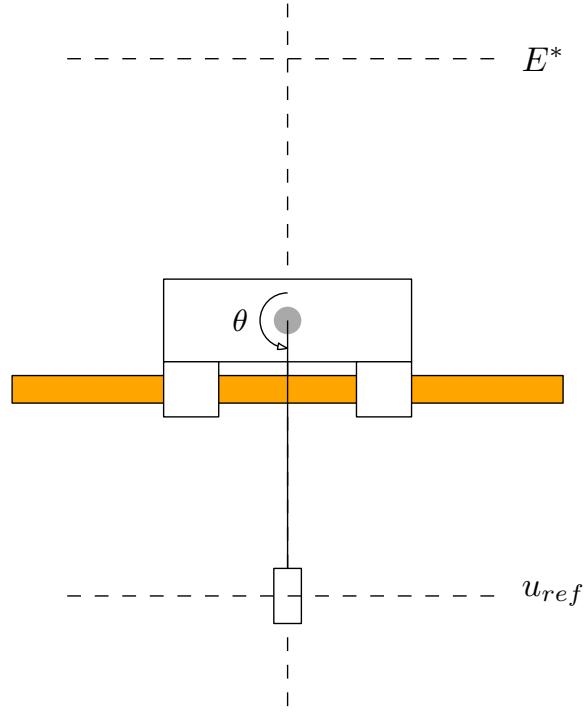


Figure 6.2: Placement of the potential energy reference, at the center of mass of the pendulum when it is hanging straight down.

This also indicates that the potential energy, when the pendulum is upright, corresponds to the energy reference

$$E^* = 2m_p gl \quad (6.8)$$

Swinging up the pendulum will specifically concern controlling the energy such that the pendulum reaches its heteroclinic orbits. These orbits are found by investigating the case where the energy of the pendulum is equal to E^* , i.e.

$$\begin{aligned} 0 &= E^* - E_p \\ &= 2m_p gl - \frac{1}{2} m_p l^2 \dot{\theta}^2 - m_p gl(\cos \theta + 1) \\ &= (1 - \cos \theta)g - \frac{1}{2} l \dot{\theta}^2 \end{aligned} \quad (6.9)$$

Solving for $\dot{\theta}$ yields the expression for the heteroclinic orbits

$$\dot{\theta} = \pm \sqrt{\frac{2g}{l}(1 - \cos \theta)} \quad (6.10)$$

This is a function of θ , and becomes zero when $\theta = 2\pi k$ for $k \in \mathbb{Z}$. The trajectories of **equation (6.10)** is visualised for $\{\theta \in \mathbb{R} \mid 0 \leq \theta \leq 2\pi\}$ in **figure 6.3** in bright red.

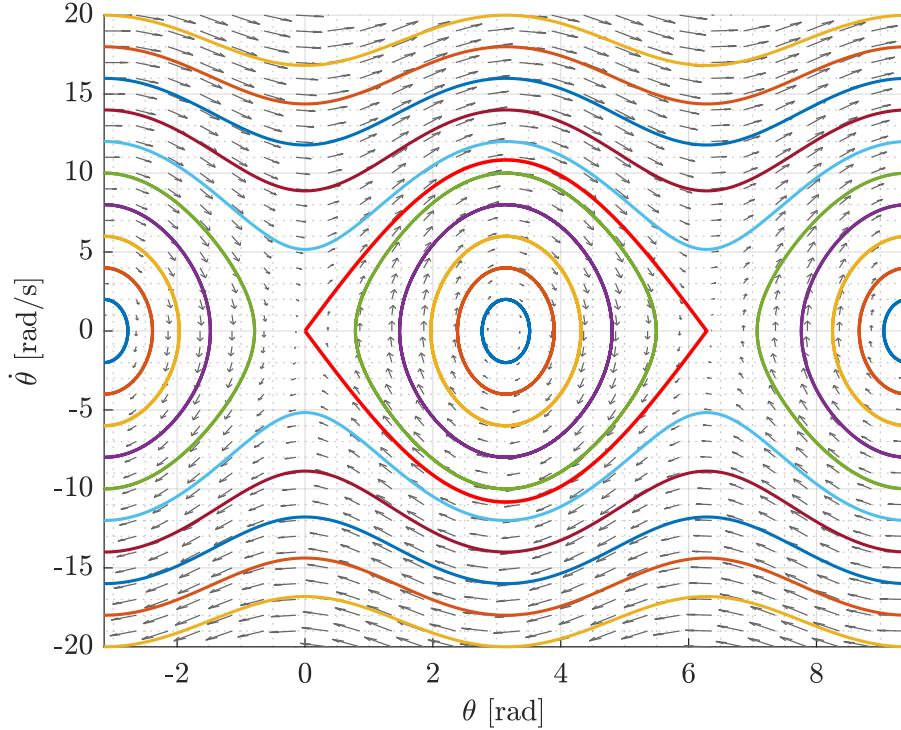


Figure 6.3: Phase plot of the system **without friction**.

At either side of a heteroclinic orbit exists an equilibrium point, specifically at $(\theta, \dot{\theta}) = (0, 0)$ and $(\theta, \dot{\theta}) = (2\pi, 0)$. Thus, a heteroclinic orbit is defined as a trajectory which connects two different equilibrium points. Physically, however, these two points corresponds to the same vertical angle of the pendulum. The trajectories outside the heteroclinic orbits are trajectories where the energy in the system exceeds the energy reference, and they will remain on the indicated paths as long as there is no friction. The trajectories inside are thus trajectories where the energy is lower than the reference.

Including friction in the system equations will yield trajectories like depicted in **figure 6.4**.

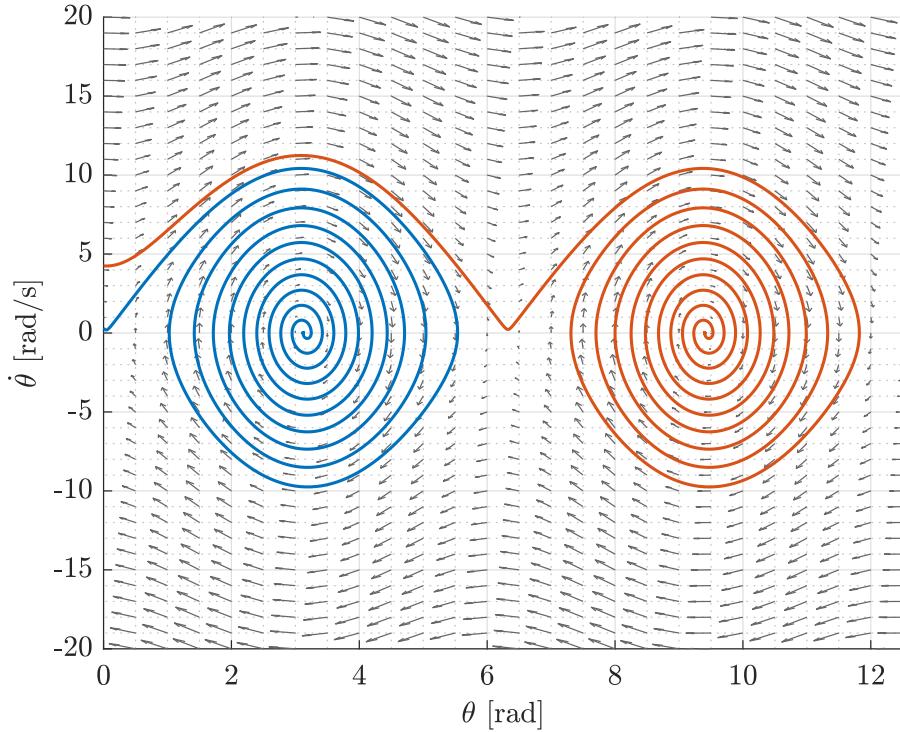


Figure 6.4: Phase plot of the pendulum system **with** friction. The friction coefficients have been chosen larger than the actual values for illustrative purposes.

The friction removes energy from the system and creates stable equilibria at the points $\theta = (2k + 1)\pi$ for $k \in \mathbb{Z}$. That is, when within the heteroclinic orbits of two equilibrium points, the system will converge towards the stable equilibrium point between them, if no control input is present.

From the above analysis, the purpose of the swing-up controller can be described as to bring the system from its initial state to the heteroclinic orbits.

6.3 Cart friction feedforward

From [section 4.3.1](#) and [appendices B](#) and [C](#), the friction exerted on the cart has proved to be troublesome. While viscous friction is insignificant, the Coulomb term is direction-dependent and shows tendencies of depending on the position as well.

To simplify the design process in [chapters 7](#) and [8](#), the Coulomb friction of the cart is omitted from the model equations, and added to the controller output.

By removing the Coulomb friction, the friction vector from [section 4.4](#) is redefined as

$$\mathbf{B}(\dot{\theta}) = \begin{bmatrix} 0 \\ c_p \tanh(k\dot{\theta}) + v_p \dot{\theta} \end{bmatrix} \quad (6.11)$$

for use in [chapters 7](#) and [8](#), and the cart friction is added by feedforward as depicted in [figure 6.1](#), with

$$B_{ff}(\dot{x}, u) = c_c(\dot{x}) \tanh(k \cdot u), \quad c_c(\dot{x}) = \begin{cases} c_{c,R} & \text{if } \dot{x} \geq 0 \\ c_{c,L} & \text{if } \dot{x} < 0 \end{cases} \quad (6.12)$$

where u is the output from the active controller and k is a tuning parameter that determines how steep $\tanh(\cdot)$ is through zero. The $\tanh(\cdot)$ in the feedforward is dependent on the sign of u rather than \dot{x} in order to avoid e.g. adding positive friction compensation to negative u , effectively counteracting the control action of the active controller.

6.4 Catch trigger

While the control scheme is initialised with the swing-up controller, an algorithm is needed to determine when to trigger the switch to the stabilising controller.

The conditions under which the stabilising controller will be activated are based on two criteria:

- Pendulum energy: $|E^* - E_p| \leq 0.05E^*$
- Angular span: $|\theta| \leq 30^\circ = 0.5236 \text{ rad}$

where $E_p = \frac{1}{2}m_p l^2 \dot{\theta}^2 + m_p g l (\cos\theta + 1)$ is the mechanical energy of the pendulum, i.e. without considering the cart.

If none, or only one of these two criterias are met, the output of the swing-up controller will be applied to the system. When both are satisfied, the output from the stabilising controller is applied to the system.

Note that when one controller is active, the other remains passive. In terms of implementation, it is beneficial to completely skip computing the output of the passive controller.

This chapter describes the different control methods investigated to swing up the pendulum to a point where the stabilising controller can take over, and keep the pendulum upright. As stated in the system requirements in **sections 3.1.1** and **3.1.2**, the objectives of the swing-up controller is to drive the cart in a motion such that the pendulum converges towards the heteroclinic orbits while keeping the cart at its reference position.

A theoretical approach based on LaSalle's invariance principle is presented from which two different control laws are derived; one in **section 7.1** based on the mechanical energy of the pendulum (omitting the cart) and one in **section 7.2**, which is based on the full system mechanical energy. These will be referred to as the *pendulum energy method* and *full energy method*, respectively.

Based on simulation results, the control laws are modified in **section 7.3** to meet the requirements, and a third method, a simpler variation to the method in **section 7.1** is introduced, which will be referred to as the *sign-based method*. Thus, three swing-up controllers are presented in this chapter and they will all be implemented and compared.

Note that all simulations presented in this chapter are done on the full nonlinear model.

7.1 Pendulum energy method

The following is based on [15].

In this section, a simple approach to the energy-based swing-up control will be derived. Consider the pendulum subsystem, with dynamics described by

$$0 - B_p(\dot{\theta}) = -m_p l \ddot{x} \cos \theta + m_p l^2 \ddot{\theta} - m_p g l \sin \theta \\ \Leftrightarrow J \ddot{\theta} = m_p l \ddot{x} \cos \theta + m_p g l \sin \theta - B_p(\dot{\theta}), \quad J = m_p l^2 \quad (7.1)$$

where $B_p(\dot{\theta})$ is the pendulum friction model. That is, the pendulum is described by the bottom row of **equation (4.17)**, page 15.

Omitting the friction term in **equation (7.1)**, and letting the cart acceleration \ddot{x} be considered the system input u , the equation becomes

$$J \ddot{\theta} = m_p l u \cos \theta + m_p g l \sin \theta \quad (7.2)$$

Since the input for this is an acceleration, but the real system requires a force, an approximate conversion from desired acceleration to force is presented in **section 7.1.1**. The mechanical energy for the pendulum, E_p , is given by the potential and kinetic energies, U_p and T_p , defined in **equation (4.11)**

and **equation (4.13)**, page 14:

$$\begin{aligned} E_p &= T_p + U_p \\ &= \frac{1}{2} J \dot{\theta}^2 + m_p g l \cos \theta + u_{ref} \\ &= \frac{1}{2} J \dot{\theta}^2 + m_p g l (\cos \theta + 1) \end{aligned} \quad (7.3)$$

Taking the time derivative of the energy yields

$$\begin{aligned} \dot{E}_p &= J \dot{\theta} \ddot{\theta} - m_p g l \dot{\theta} \sin \theta \\ &= (m_p g l \sin \theta + m_p l u \cos \theta) \dot{\theta} - m_p g l \dot{\theta} \sin \theta \\ &= m_p l u \dot{\theta} \cos \theta \end{aligned} \quad (7.4)$$

From this equation, it is observed that the mechanical energy can be controlled with input u , and that control action is lost when $\theta = \pm \frac{\pi}{2}$ or $\dot{\theta} = 0$, i.e. when the pendulum is horizontal or not in motion. On the other hand, the control input is most effective when $\theta = \{0, \pm \pi\}$ and $\dot{\theta}$ is large.

In order to increase the energy of the pendulum, control input must be positive when $\dot{\theta} \cos \theta$ is positive. Recalling the definitions of the energy error in **section 6.2**, a control law following $\dot{\theta} \cos \theta$ and seeking to minimise the energy error is proposed

$$u = k_e (E^* - E_p) \dot{\theta} \cos \theta \quad (7.5)$$

where $k_e > 0$ is a controller gain.

Considering **theorem 6.1.2** (LaSalle's theorem), Ω is chosen as the set contained within the heteroclinic orbits in **figure 6.3**, i.e.

$$\Omega = \left\{ (\theta, \dot{\theta}) \mid 0 \leq \theta \leq 2\pi, -\sqrt{\frac{2g}{l}(1 - \cos \theta)} \leq \dot{\theta} \leq \sqrt{\frac{2g}{l}(1 - \cos \theta)} \right\} \quad (7.6)$$

Using the function

$$V_1(\theta, \dot{\theta}) = \frac{1}{2} (E^* - E_p)^2 = \frac{1}{2} \left((1 - \cos \theta) g - \frac{1}{2} l \dot{\theta}^2 \right)^2 \quad (7.7)$$

yields

$$\begin{aligned} \dot{V}_1(\theta, \dot{\theta}) &= (E^* - E_p) \left(\dot{E}^* - \dot{E}_p \right) \\ &= (E^* - E_p) \left(0 - m_p l u \dot{\theta} \cos \theta \right) \\ &= -(E^* - E_p) m_p l u \dot{\theta} \cos \theta \\ &= -(E^* - E_p) m_p l k_e (E^* - E_p) \dot{\theta} \cos \theta \dot{\theta} \cos \theta \\ &= -m_p l k_e \left((E^* - E_p) \dot{\theta} \cos \theta \right)^2 \leq 0 \end{aligned} \quad (7.8)$$

from which the set \mathbb{E} is given by

$$\begin{aligned}\mathbb{E} &= \left\{ (\theta, \dot{\theta}) \in \Omega \mid \dot{V}_1(\theta, \dot{\theta}) = 0 \right\} \\ &= \left\{ (\theta, \dot{\theta}) \in \Omega \mid \dot{\theta} = 0 \right\} \cup \left\{ (\theta, \dot{\theta}) \in \Omega \mid \dot{\theta} = \pm \sqrt{\frac{2g}{l}(1 - \cos \theta)} \right\} \\ &= \mathbb{E}_1 \cup \mathbb{M}\end{aligned}\tag{7.9}$$

In this, \mathbb{M} is the largest invariant set in \mathbb{E} , and thus from **theorem 6.1.2** (LaSalle's theorem) it follows that every solution starting in Ω approaches \mathbb{M} as $t \rightarrow \infty$.

The control law in **equation (7.5)** does not include control of the cart position, and the cart must therefore be placed in the middle of the rail in order to avoid collision with the physical limits of the system.

A simulation using the control law in **equation (7.5)** is shown in **figure 7.1** and **figure 7.2** with $k_e = 0.5$

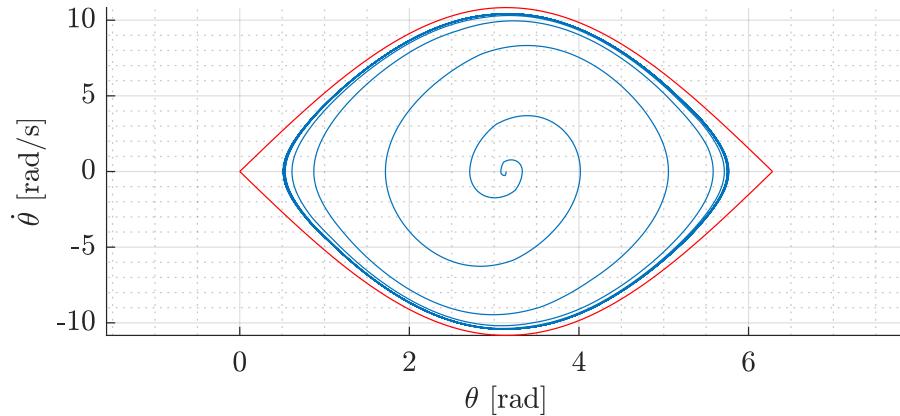


Figure 7.1: Phase plot(blue) of a simulation using the derived control law and the full system model. The heteroclinic orbits(red) are added as reference. Simulation is done on the full nonlinear model.

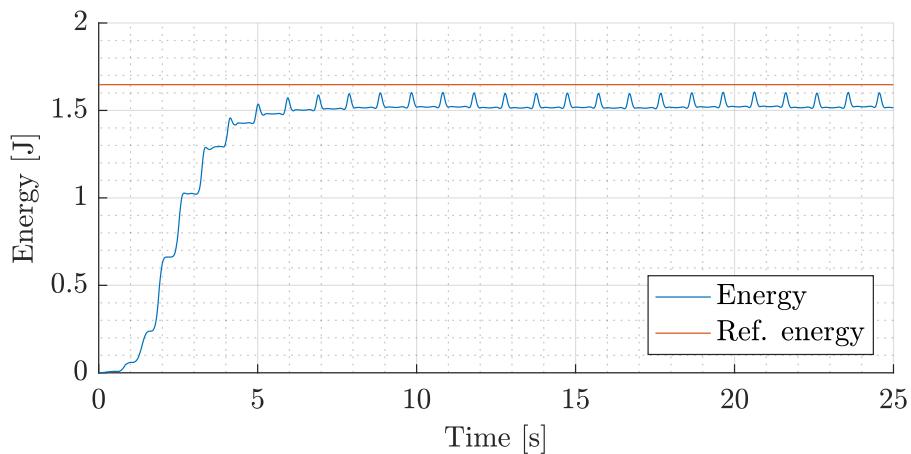


Figure 7.2: Pendulum energy when simulating the full model using the derived control law in **equation (7.5)**. Simulation is done on the full nonlinear model.

It is seen that the pendulum does not fully reach the heteroclinic orbits, thus the control law is insufficient. In order to remedy this, a different approach will be considered in **section 7.2**.

7.1.1 Cart acceleration to force

When considering the cart acceleration as the pendulum subsystem input, and deriving a control law based on this, the controller output must be converted to a force.

From **equation (4.17)** on page 15, the relation between F and \ddot{x} is

$$F = (m_c + m_p)\ddot{x} - m_p l \cos \theta \ddot{\theta} + m_p l \sin \theta \dot{\theta}^2 \quad (7.10)$$

when omitting the cart friction.

This relation is used to compute the force to apply to the system from the control signal u by replacing \ddot{x} with u and using θ , $\dot{\theta}$ and $\ddot{\theta}$ from the EKF.

7.2 Full energy method

The following is based on [16, 17].

Instead of considering just the pendulum energy, this method considers the energy of the entire system, attempting to include control of the cart in the derivation of the control law. The full system energy is defined using the potential energy of the pendulum, U_p , from **equation (4.11)** on page 14 and inertia matrix from **equation (4.17)** on page 15.

$$\begin{aligned} E &= \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M}(\theta) \dot{\mathbf{q}} + U_p \\ &= \frac{1}{2} \begin{bmatrix} \dot{x} & \dot{\theta} \end{bmatrix} \begin{bmatrix} m_c + m_p & -m_p l \cos \theta \\ -m_p l \cos \theta & m_p l^2 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix} + m_p g l (\cos \theta + 1) \\ &= \frac{1}{2} \dot{x}^2 (m_c + m_p) + \frac{1}{2} \dot{\theta}^2 m_p l^2 - \dot{x} \dot{\theta} m_p l \cos \theta + m_p g l (\cos \theta + 1) \end{aligned} \quad (7.11)$$

Furthermore, the dynamics presented in **equation (4.17)** are considered, using the generalised coordinate to express the accelerations

$$\mathbf{M}(\theta) \ddot{\mathbf{q}} = -\mathbf{C}(\theta, \dot{\theta}) - \mathbf{G}(\theta) + \boldsymbol{\tau} - \mathbf{B}(\dot{\theta}), \quad \boldsymbol{\tau} = [u \ 0]^T \quad (7.12)$$

As in **section 7.1**, the pendulum friction is omitted, effectively setting $\mathbf{B} = 0$. The time derivative of

equation (7.11) yields

$$\begin{aligned}
\dot{E} &= \frac{1}{2}\ddot{\mathbf{q}}^T \mathbf{M}(\theta)\dot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T \dot{\mathbf{M}}(\theta)\dot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T \mathbf{M}(\theta)\ddot{\mathbf{q}} + \dot{U}_p \\
&= \dot{\mathbf{q}}^T \mathbf{M}(\theta)\ddot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T \dot{\mathbf{M}}(\theta)\dot{\mathbf{q}} + \dot{\mathbf{q}}^T \mathbf{G}(\theta), \quad \frac{\partial U_p}{\partial \mathbf{q}} = \dot{\mathbf{q}}^T \mathbf{G}(\theta) \\
&= \dot{\mathbf{q}}^T (\boldsymbol{\tau} - \mathbf{C}(\theta, \dot{\theta}) - \mathbf{G}(\theta)) + \frac{1}{2}\dot{\mathbf{q}}^T \dot{\mathbf{M}}(\theta)\dot{\mathbf{q}} + \dot{\mathbf{q}}^T \mathbf{G}(\theta) \\
&= \dot{\mathbf{q}}^T \boldsymbol{\tau} - \dot{\mathbf{q}}^T \mathbf{G}(\theta) + \dot{\mathbf{q}}^T \mathbf{G}(\theta) - \dot{\mathbf{q}}^T \mathbf{C}(\theta, \dot{\theta}) + \frac{1}{2}\dot{\mathbf{q}}^T \dot{\mathbf{M}}(\theta)\dot{\mathbf{q}} \\
&= \dot{\mathbf{q}}^T \boldsymbol{\tau} - [\dot{x} \quad \dot{\theta}] \begin{bmatrix} m_p l \sin \theta \dot{\theta}^2 \\ 0 \end{bmatrix} + \frac{1}{2} [\dot{x} \quad \dot{\theta}] \begin{bmatrix} 0 & m_p l \sin \theta \dot{\theta} \\ m_p l \sin \theta \dot{\theta} & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix} \\
&= \dot{x}u
\end{aligned} \tag{7.13}$$

The following function is then proposed, including the cart position and velocity, from which a control law is found inspired by the Lyapunov redesign approach:

$$V_2(\mathbf{x}) = \frac{k_e}{2}(E^* - E)^2 + \frac{k_v}{2}\dot{x}^2 + \frac{k_x}{2}(x^* - x)^2 \tag{7.14}$$

where k_e , k_v , and k_x are positive constants, E^* and x^* are energy and position references. Taking the time derivative yields

$$\begin{aligned}
\dot{V}_2(\mathbf{x}) &= -k_e(E^* - E)\dot{E} + k_v\dot{x}\ddot{x} - k_x(x^* - x)\dot{x} \\
&= -k_e(E^* - E)\dot{x}u + k_v\dot{x}\ddot{x} - k_x(x^* - x)\dot{x} \\
&= \dot{x}(-k_e(E^* - E)u + k_v\ddot{x} + k_x(x^* - x))
\end{aligned} \tag{7.15}$$

Inserting the equation of \ddot{x} from **equation (4.40)** on page 21, the derivative extends to

$$\dot{V}_2(\mathbf{x}) = \dot{x} \left(u(-k_e(E^* - E) + k_v\dot{\alpha}(\theta)) + k_v\dot{\beta}(\theta, \dot{\theta}) - k_x(x^* - x) \right) \tag{7.16}$$

Where

$$\dot{\alpha}(\theta) = \frac{1}{m_c + m_p \sin^2 \theta} \tag{7.17}$$

$$\dot{\beta}(\theta, \dot{\theta}) = \frac{m_p \sin \theta (-l\dot{\theta}^2 + g \cos \theta)}{m_c + m_p \sin^2 \theta} \tag{7.18}$$

Based on **equation (7.16)** a control law is proposed such that

$$u(-k_e(E^* - E) + k_v\dot{\alpha}(\theta)) + k_v\dot{\beta}(\theta, \dot{\theta}) - k_x(x^* - x) = -k_d\dot{x} \tag{7.19}$$

Where k_d is a positive constant. The control law is derived by isolating u in **equation (7.19)**, which yields

$$u = \frac{-k_d\dot{x} + k_x(x^* - x) - k_v\dot{\beta}(\theta, \dot{\theta})}{-k_e(E^* - E) + k_v\dot{\alpha}(\theta)} \tag{7.20}$$

Inserting **equation (7.20)** into **equation (7.16)** results in

$$\dot{V}_2(\mathbf{x}) = -k_d \dot{x}^2 \leq 0 \quad (7.21)$$

In order to apply **theorem 6.1.2** (LaSalle's theorem) to **equation (7.21)** based on the set Ω in **equation (7.6)**, the coordinate transformation

$$T = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (7.22)$$

is used, which yields the transformed state vector $\tilde{\mathbf{x}} = [\theta \quad \dot{\theta} \quad x \quad \dot{x}]^T$. Then, the fourth-dimensional set Ω_4 can be defined as

$$\begin{aligned} \Omega_4 &= \Omega \times [x_{min}, x_{max}] \times [\dot{x}_{min}, \dot{x}_{max}] \\ &= \Omega \times [0, l_{rail} - w_{cart}] \times [\dot{x}_{min}, \dot{x}_{max}] \end{aligned} \quad (7.23)$$

From **equation (7.23)** the set \mathbb{E}_4 is defined as

$$\mathbb{E}_4 = \left\{ \tilde{\mathbf{x}} \in \Omega_4 \mid \dot{V}_2(\tilde{\mathbf{x}}) = 0 \right\} \quad (7.24)$$

As in **section 7.1**, it is desired to find the largest invariant set in \mathbb{E}_4 , denoted \mathbb{M}_4 , such that it follows from **theorem 6.1.2** (LaSalle's theorem) that every solution starting in Ω_4 approaches \mathbb{M}_4 as $t \rightarrow \infty$.

Simulations shows, as the one depicted in **figure 7.3**, that the largest invariant set is

$$\mathbb{M}_4 = \left\{ \tilde{\mathbf{x}} \in \Omega_4 \mid \dot{\theta} = \pm \sqrt{\frac{2g}{l}(1 - \cos \theta)}, \dot{x} = 0, x = x^* \right\} \quad (7.25)$$

which corresponds to the bright red orbit. The set relates to \mathbb{E}_4 by

$$\mathbb{E}_4 = \mathbb{M}_4 \cup \mathbb{E}_4 \setminus \mathbb{M}_4 \quad (7.26)$$

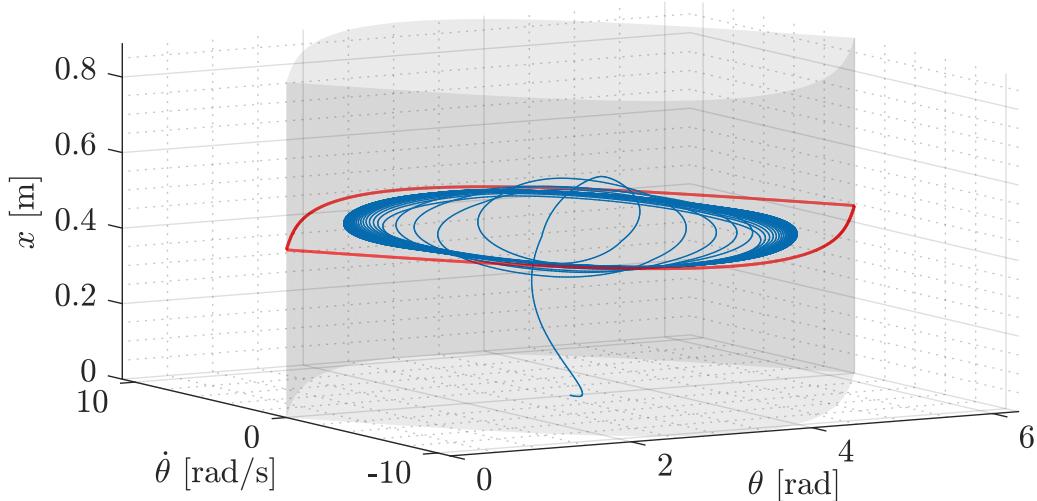


Figure 7.3: Phase plot(blue) of the derived control law simulated on the full nonlinear model. The red orbit corresponds to the set \mathbb{M}_4 . The shaded cylinder represents the set \mathbb{M}_4 where $0 \leq x^* \leq l_{rail} - w_{cart}$.

The simulation is performed on the full nonlinear model, using the following gains $k_x = 12$, $k_d = 1$, $k_v = 8.3$, and $k_e = 1$.

As the simulations in [figure 7.1](#) and [figure 7.3](#) show, neither the pendulum energy method nor the full energy method manage to reach the heteroclinic orbits. Therefore, other approaches must be considered to ensure this, such that the stabilising controller can be enabled and stabilise the pendulum.

7.3 Heuristic swing-up control design

The results in [section 7.1](#) and [section 7.2](#) indicate that the control laws are not able to reach the heteroclinic orbits. This section continues on the basis of these results, but with a heuristic approach and only concern reaching the heteroclinic orbits by modifying the control laws.

In [section 7.3.1](#) the full energy method is reconsidered by adding an integral term such that it reaches the energy reference and heteroclinic orbits. The pendulum energy method's control law is expanded to include the cart position, and an integral term is also added, which is explained [section 7.3.2](#). In [section 7.3.3](#) the sign-based method is presented, which is, as mentioned, a simpler variation of the method presented in [section 7.1](#).

Adding and tuning the integral terms must be done carefully to avoid overshooting the energy reference before the stabilising controller takes over.

7.3.1 Full energy method

This section is based on the results presented in [section 7.1](#) (Full energy method).

Considering the simulation data of the phase plot in [figure 7.3](#), the energy over time is shown in [figure 7.4](#).

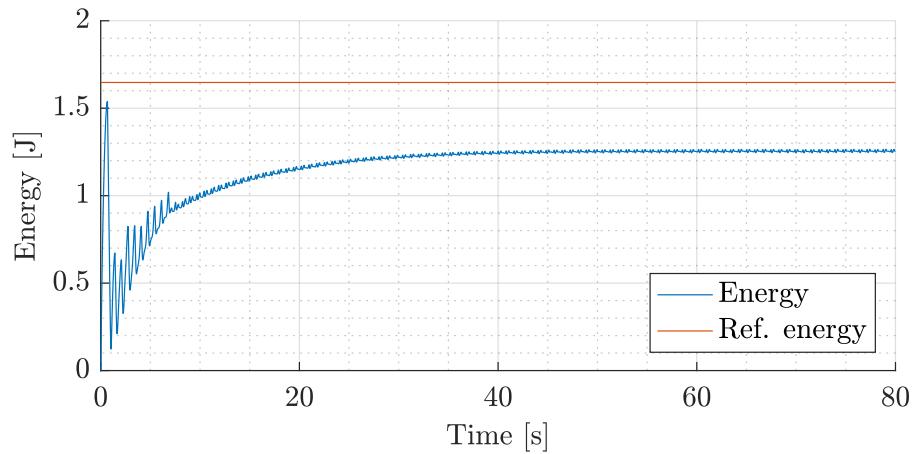


Figure 7.4: System energy when using the control derived in the Full energy method in [section 7.2](#), simulated on the full model.

Here, it is seen that it does not reach the reference energy, but settles with a steady-state error. In order to eliminate this error, an integral term is added to the existing control law in [equation \(7.20\)](#),

such that

$$u = \frac{-k_d \dot{x} + k_x(x^* - x) - k_v \dot{\beta}(\theta, \dot{\theta})}{-k_e(E^* - E) + k_v \dot{\alpha}(\theta)} + \dot{\theta} \cos \theta k_i \int_0^t e(\tau) dt \quad (7.27)$$

where $e(\tau) = E^* - E$ is the integrated energy error from time $t = 0$ until t , and k_i is a positive constant. The term $\dot{\theta} \cos \theta$ originates from [15], where it is shown that it maximises the efficiency of the control signal at the most appropriate time thus not working against the existing control signal.

Using $k_x = 12, k_d = 1, k_v = 8.3, k_e = 1$, and $k_i = 0.012$, **figure 7.5** shows that the heteroclinic orbits is reached, but also exceeded. **Figure 7.5** shows the phase, and the trajectory indicates that

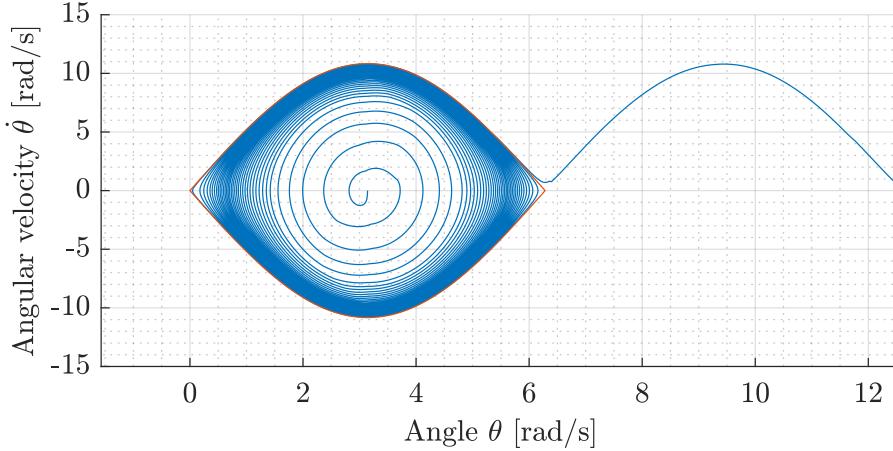


Figure 7.5: Heteroclinic orbits(red) reached using the control law in equation (7.27). The phase(blue) exceeds the heteroclinic orbits. Simulated on the full model.

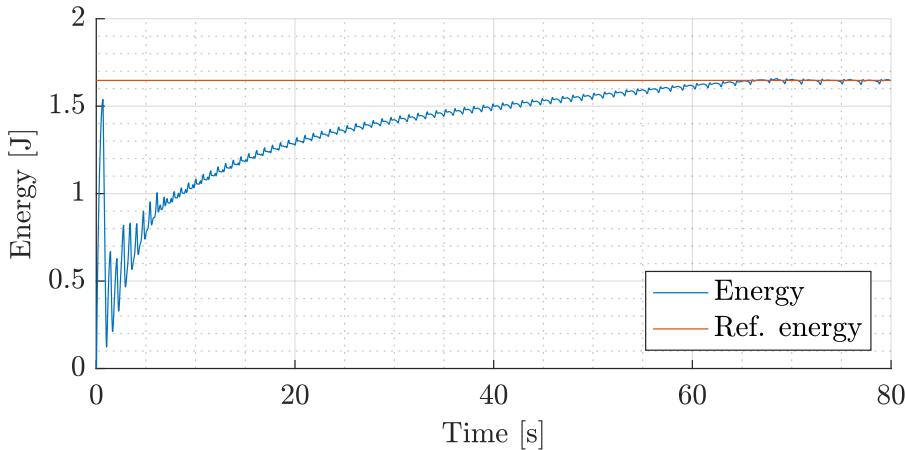


Figure 7.6: Energy reference reached using the control law in equation (7.27). Simulated on the full model.

the pendulum does a full revolution, and therefore leaves the heteroclinic orbits between $\theta = 0$ and $\theta = 2\pi$.

Figure 7.6 shows the system energy, which settles slowly but without overshooting. Increasing the integral gain improves the energy rise time, however it also introduce overshoot, and gains are therefore kept as is.

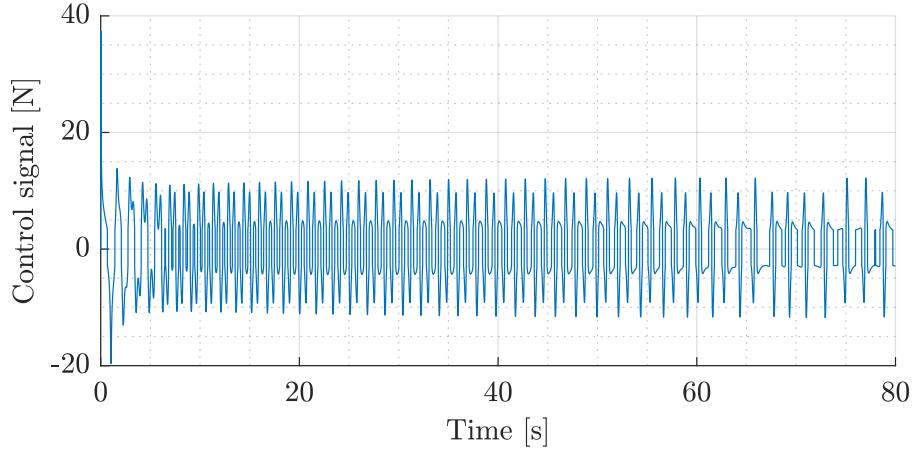


Figure 7.7: The control signal of equation (7.27) when simulated on the full model.

Finally, the k_v and k_e gains have been chosen systematically, based on the denominator in equation (7.27). To avoid singularities, the following relation can be established:

$$0 \neq -k_e(E^* - E) + \frac{k_v}{m_c + m_p \sin^2 \theta} \quad (7.28)$$

$$\frac{k_v}{k_e} \neq (E^* - E)(m_c + m_p \sin^2 \theta) \quad (7.29)$$

Inspecting the maximum value of the right hand side, i.e. when $\sin^2 \theta = 1$, and using the inequality $E^* - E \leq 2m_p g l$, gives

$$\frac{k_v}{k_e} \geq 8.2381 \quad (7.30)$$

The inequality is achieved by setting $k_v = 8.3$ and $k_e = 1$. The other gains are chosen iteratively during the design process.

7.3.2 Pendulum energy method

This section continues based on the results shown in **section 7.1** (Pendulum energy method). The results presented in **figure 7.1** do not include control of the cart position, and it is therefore required to start in the middle of the rail. In order to reach a reference position for the cart, and keep it within the limits of the physical setup, a proportional control is added to the control law defined in equation (7.5), such that

$$u = k_e(E^* - E_p)\dot{\theta} \cos \theta + k_c(x^* - x) \quad (7.31)$$

Based on the results in **section 7.1**, where the pendulum energy settles with a steady-state error, an integral term on the energy error is added, i.e.

$$\begin{aligned} u &= k_e(E^* - E_p)\dot{\theta} \cos \theta + k_c(x^* - x) + k_i \dot{\theta} \cos \theta \int_0^t e_p(\tau) dt \\ &= \left(k_e(E^* - E_p) + k_i \int_0^t e_p(\tau) dt \right) \dot{\theta} \cos \theta + k_c(x^* - x) \end{aligned} \quad (7.32)$$

where the integral error is defined as $e_p(\tau) = E^* - E_p$, from time 0 until time t .

A simulation using this revised control law are shown in **figures 7.8** and **7.9** with $k_e = 0.5$, $k_i = 0.02$ and $k_c = 1$, chosen iteratively.

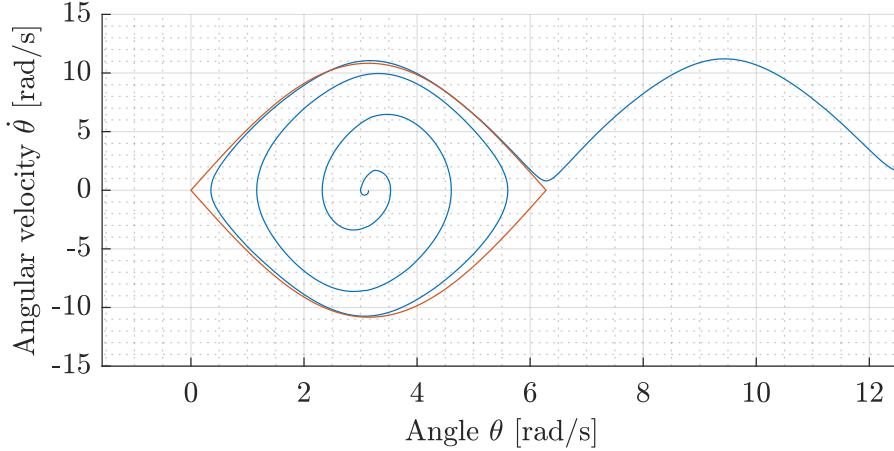


Figure 7.8: Pendulum phase plot(blue) for pendulum energy control with integral action on the full model. The heteroclinic orbits(red) are also shown.

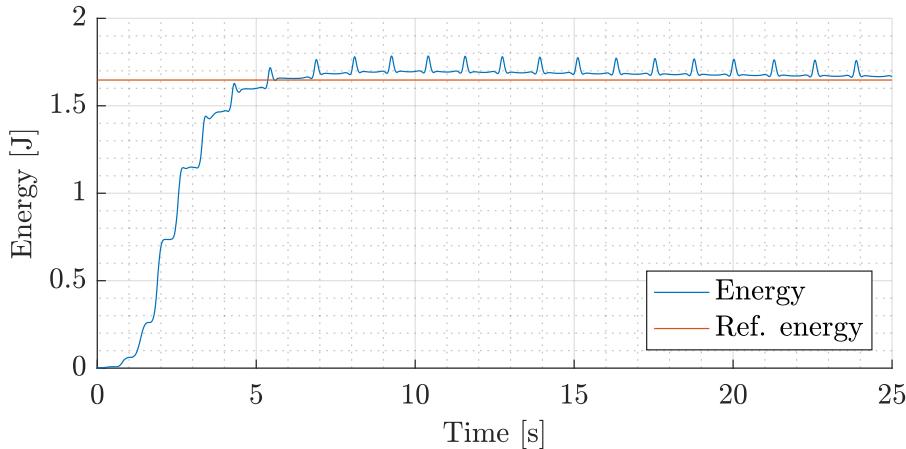


Figure 7.9: Pendulum energy for pendulum energy control with integral action. Simulated on the full model.

This shows that the mechanical energy reference is exceeded by a small margin, and the pendulum does several full revolutions once this happens.

The controller output for the simulation is shown in **figure 7.10**.

Tests with the value of k_i have shown that setting it too high will produce an overshoot, after which the energy will slowly decrease to the reference. Setting it too low will lower the rate at which the steady state error is corrected.

In general, this control approach utilises a simple expression for computing the system input, yet it struggles with reaching the energy reference, and relies on integral action to reach the value. The gain on the integral action is very sensitive, but a small overshoot is not an issue, as the stabilising control will take over once both the energy and angle criteria are met, as described in **section 6.4**.

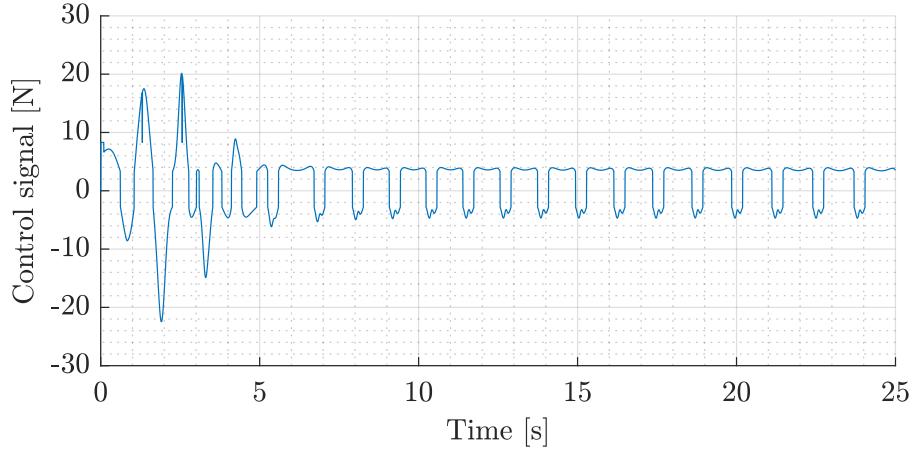


Figure 7.10: Controller output from the pendulum energy method with integral action.

7.3.3 Sign-based method

The following is based on [15].

As an alternative to the control law proposed in **equation (7.5)**, [15] suggests maximising the control input in order to increase the rate at which the energy error converges to zero.

This is achieved by defining a maximum control signal constant, u_{max} , and the control law

$$u = u_{max} \operatorname{sgn} \left((E^* - E_p) \dot{\theta} \cos \theta \right) \quad (7.33)$$

which drives $|E^* - E_p|$ towards zero.

Considering the function in **equation (7.7)**,

$$V_3(\mathbf{x}) = V_1(\mathbf{x}) = \frac{1}{2}(E^* - E_p)^2 = \frac{1}{2} \left((1 - \cos \theta)g - \frac{1}{2}l\dot{\theta}^2 \right)^2 \quad (7.34)$$

yields

$$\begin{aligned} \dot{V}_3(\mathbf{x}) &= -(E^* - E_p)m_p l u \dot{\theta} \cos \theta \\ &= -(E^* - E_p)m_p l u_{max} \operatorname{sgn} \left((E^* - E_p) \dot{\theta} \cos \theta \right) \dot{\theta} \cos \theta \end{aligned} \quad (7.35)$$

Recalling that $\operatorname{sgn}(0) = 0$, proving convergence towards a set using **theorem 6.1.2** (LaSalle's theorem) yields the sets defined in **equation (7.9)**.

The control law in **equation (7.33)** will cause chattering around $(E^* - E_p) \dot{\theta} \cos \theta = 0$. Therefore, the $\operatorname{sgn}(\cdot)$ can be replaced by e.g. a saturation function. Chattering will occur shortly when the pendulum changes direction, and as θ approaches zero. However, as this project relies on a different controller for stabilising the pendulum in the vicinity of $\theta = 0$, this is not expected to be an issue.

Additionally, for the control of the cart position, **equation (7.33)** can be extended to

$$u = u_{max} \operatorname{sgn} \left((E^* - E_p) \dot{\theta} \cos \theta \right) + k_c(x^* - x) \quad (7.36)$$

A simulation of the full nonlinear system with this control law is shown in **figures 7.11** and **7.12**, with $u_{max} = 1$ and $k_x = 1$.

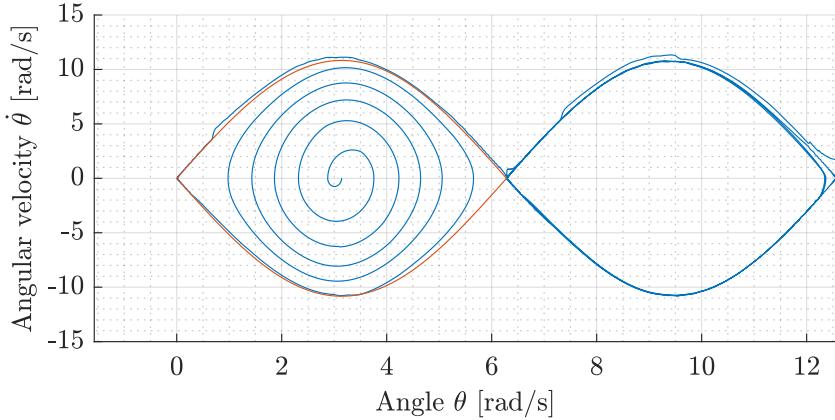


Figure 7.11: Pendulum phase plot(blue) for the sign-based method, simulated on the full model. The heteroclinic orbits(red) are shown.

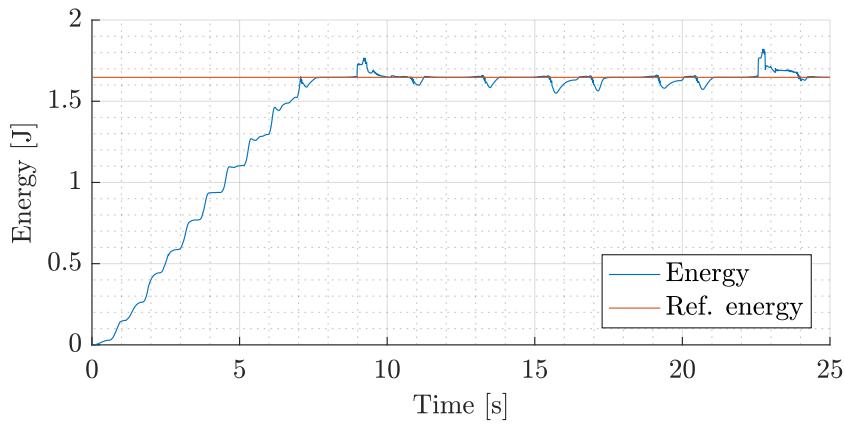


Figure 7.12: Pendulum energy for the sign-based method, simulated on the full model.

They show that this control law quickly increases the energy of the pendulum and settles very close to the reference. The system energy is generally kept, but temporarily exceeds the reference at $t \approx 9$ s, causing the pendulum to do a full revolution, as is seen on the phase plot. Additionally, this method does not require integral action.

The corresponding controller output is shown in **figure 7.13**. Here, it is evident how the $\text{sgn}(\cdot)$

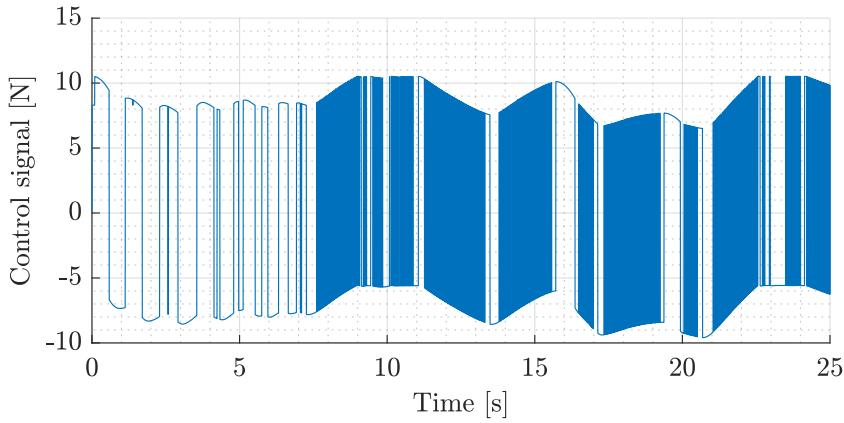


Figure 7.13: Control output of the sign-based method.

affects the controller output. At $t \approx 7.5$ s, when the energy reference is met, the chattering occurs.

Control Design: Stabilisation 8

This chapter presents the different control methods which will stabilise the pendulum in the upright position while keeping the cart in its reference position.

Three different controller types are considered. **Section 8.1** presents a cascade coupled Proportional-Integral-Derivative (PID) control scheme using decoupled dynamics of the pendulum and cart. In **section 8.2** an Linear-Quadratic-Gaussian (LQG) control scheme is derived, and lastly a nonlinear Sliding-Mode Controller (SMC) is derived in **section 8.3**. Thus three stabilising controllers will be designed, implemented, and compared.

Recall that the friction term of the cart is omitted, and added as a feedforward term, as described in **section 6.3**.

Note that all simulations presented in this chapter are done on the full nonlinear model.

8.1 Proportional-Integral-Derivative control

An PID control scheme has been developed to stabilize the pendulum and keep the cart at a desired position. Since the PID controller itself is only able to control one variable, two PID controllers have been coupled to control the x and θ variables.

Recall the dynamics defined in **equation (4.17)**:

$$\begin{bmatrix} u \\ 0 \end{bmatrix} - \mathbf{B}(\dot{x}, \dot{\theta}) = \underbrace{\begin{bmatrix} m_c + m_p & -m_p l \cos \theta \\ -m_p l \cos \theta & m_p l^2 \end{bmatrix}}_{\mathbf{M}(\theta)} \begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} + \underbrace{\begin{bmatrix} m_p l \sin \theta \dot{\theta}^2 \\ 0 \end{bmatrix}}_{\mathbf{C}(\theta, \dot{\theta})} + \underbrace{\begin{bmatrix} 0 \\ -m_p g l \sin \theta \end{bmatrix}}_{\mathbf{G}(\theta)} \quad (8.1)$$

This is simplified using the small angle approximation technique, such that

$$\begin{aligned} u &= \mathbf{M}_{(1,1)} \ddot{x} + \mathbf{M}_{(1,2)} \ddot{\theta} + \mathbf{C} \\ u &= (m_c + m_p) \ddot{x} - m_p l \ddot{\theta} + m_p l \theta \dot{\theta}^2 \end{aligned} \quad (8.2)$$

and

$$\begin{aligned} -\mathbf{B}_p(\dot{\theta}) &= \mathbf{M}_{(2,1)} \ddot{x} + \mathbf{M}_{(2,2)} \ddot{\theta} + \mathbf{G} \\ -v_p \dot{\theta} &= -m_p l \ddot{x} + m_p l^2 \ddot{\theta} - m_p g l \theta \end{aligned} \quad (8.3)$$

In order to create two decoupled transfer functions, the terms $-m_p l \ddot{\theta} + m_p l \theta \dot{\theta}^2$ are considered as disturbances in **equation (8.2)** and thus removed. The Laplace transform of this yields

$$u = (m_c + m_p) x s^2 \quad (8.4)$$

and a transfer function is then found to be

$$H_c(s) = \frac{x(s)}{u(s)} = \frac{1}{(m_c + m_p)s^2} \quad (8.5)$$

Viewing the cart acceleration as an input, i.e. $u_2 = \ddot{x}$, in **equation (8.3)**, the Laplace transform yields

$$-v_p\theta s = -m_p l u_2 + m_p l^2 \theta s^2 - m_p g l \theta \quad (8.6)$$

and a transfer function for the pendulum is found to be

$$H_p(s) = \frac{\theta(s)}{u_2(s)} = \frac{m_p l}{m_p l^2 s^2 + v_p s - m_p g l} \quad (8.7)$$

The control scheme is depicted in **figure 8.1**.

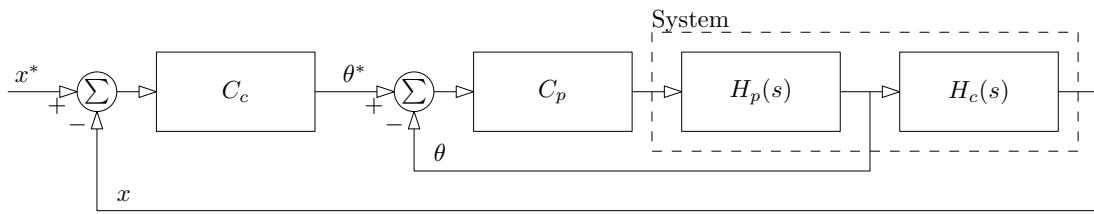


Figure 8.1: PID control scheme. One PID, C_c , to control the cart position, the output is fed as a reference to the next PID controller, C_p , which controls the pendulum angle. The system is represented by the two transfer functions $H_p(s)$ and $H_c(s)$.

Where C_c and C_p are the PID controllers on the form

$$C = \frac{K_d s^2 + K_p s + K_i}{s} \quad (8.8)$$

Where K_p , K_i , and K_d are gains for each of the control laws found iteratively and shown in **table 8.1**.

	C_c	C_p
K_p	0.2	112
K_i	0.03	15
K_d	0.15	15

Table 8.1: Gains for the two PID controllers.

When designing a cascade control scheme, the inner loop must be faster than the outer loop. The poles of the inner and outer, closed-loop systems are

$$\rho_p = \begin{bmatrix} -37.187 & -8.033 & -0.15 \end{bmatrix} \quad (8.9)$$

and

$$\rho = \begin{bmatrix} -37.16 & -8.03 & -0.137 & -0.015 & -0.0122 \pm 0.662 \end{bmatrix} \quad (8.10)$$

respectively.

The control scheme can thus be concluded to be stable. A simulation using this control scheme is shown in **figure 8.2**.

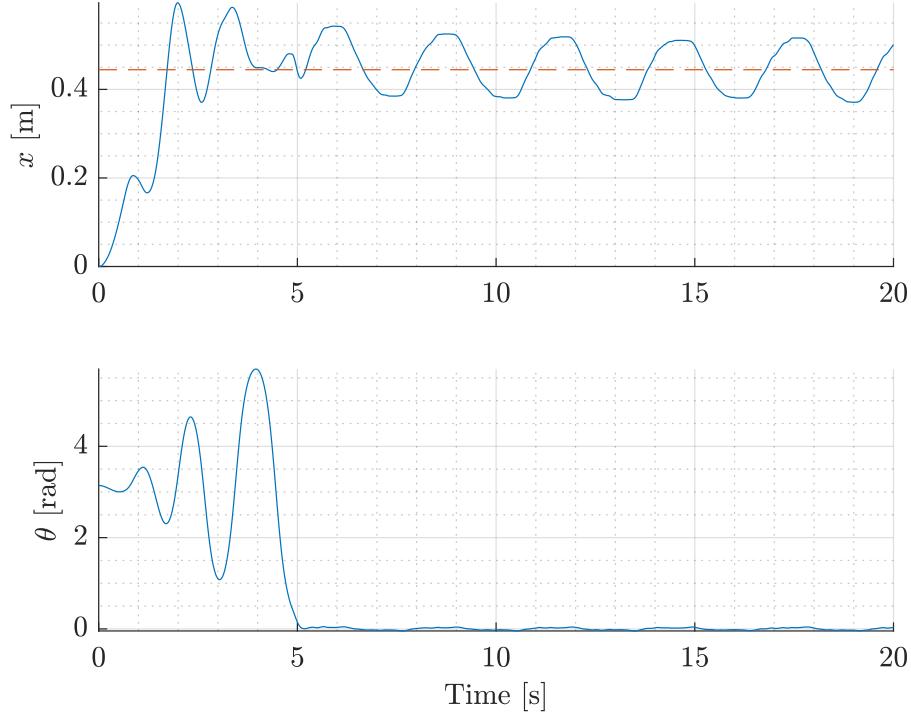


Figure 8.2: PID simulation results on the full nonlinear model. The dotted red line, on the top figure, indicates the cart reference, x^* .

The Mean Square Error (MSE) of the cart position and pendulum angle, from 10 seconds and until the end of the simulation is

$$\text{MSE}_c = 2.791 \cdot 10^{-3} \text{ m}$$

$$\text{MSE}_p = 586.8 \cdot 10^{-3} \text{ rad}$$

The values of the MSEs reflect how the cart moves ± 5 cm around the reference, and the pendulum angle moves around the zero reference with ± 0.04 rad. The results are thus satisfactory, which concludes the PID design.

8.2 Linear-Quadratic-Gaussian regulator

The LQG control scheme consists of an Linear-Quadratic Regulator (LQR) which utilizes the EKF as the state estimator, and thus requires a linear state space model, i.e.

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} &= \mathbf{Cx} + \mathbf{Du}\end{aligned}\tag{8.11}$$

where, in this case, $\mathbf{C} = \mathbf{I}$, and $\mathbf{D} = 0$.

The matrices \mathbf{A} and \mathbf{B} are obtained by linearising the nonlinear system dynamics in **equation (4.40)**, that is

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})u\tag{8.12}$$

at $\theta = 0$, which yields

$$\mathbf{A} = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \Big|_{\theta=0} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{m_p g}{m_c} & 0 & -\frac{v_\theta}{l m_c} \\ 0 & \frac{(m_c + m_p)g}{l m_c} & 0 & -\frac{(m_c + m_p)v_\theta}{l^2 m_p} \end{bmatrix} \quad (8.13)$$

and

$$\mathbf{B} = \frac{\partial g(\mathbf{x})}{\partial u} \Big|_{\theta=0} = \begin{bmatrix} 0 & 0 & \frac{1}{m_c} & \frac{1}{l m_c} \end{bmatrix}^T \quad (8.14)$$

have been found. The LQG control scheme is depicted in **figure 8.3**.

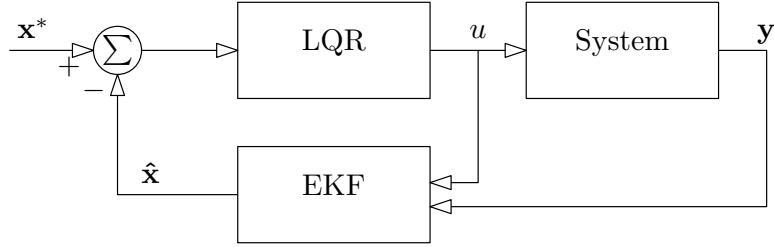


Figure 8.3: The LQG control scheme, consisting of the LQR and the EKF, and System indicates the full model.

The LQR is an optimal controller which minimizes an infinite horizon quadratic cost function given by

$$J = \int_0^\infty (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt \quad (8.15)$$

The matrices \mathbf{Q} and \mathbf{R} are weighing matrices on the states and input, respectively. The weighing matrices are bounded by $\mathbf{Q} = \mathbf{Q}^T \geq 0$ and $\mathbf{R} = \mathbf{R}^T > 0$. The input u which minimizes J is

$$u = -\mathbf{k}\mathbf{x} \quad (8.16)$$

where \mathbf{k} is given by

$$\mathbf{k} = \mathbf{R}^{-1}(\mathbf{B}^T \mathbf{P}) \quad (8.17)$$

and \mathbf{P} is found by the solution to the algebraic Riccati equation

$$\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} - (\mathbf{P} \mathbf{B}) \mathbf{R}^{-1}(\mathbf{B}^T \mathbf{P}) + \mathbf{Q} = 0 \quad (8.18)$$

The weighing matrices \mathbf{Q} and \mathbf{R} are design dependent. A good initial choice of weights can be determined by using Bryson's rule which defines the diagonal elements of the matrices as

$$\mathbf{Q}_{(k,k)} = \frac{1}{(x_k)_{max}^2}, \quad k = 1, \dots, n, \quad n = \dim \mathbf{x} \quad (8.19)$$

$$\mathbf{R} = \frac{1}{u_{max}^2} \quad (8.20)$$

Applying Bryson's rule normalizes the weights which is beneficial since they have different units. The weights for the cart and pendulum states are based on the requirements defined in **sections 3.1.1** and **3.1.2**, i.e. the pendulum is required to be ± 0.05 radians from zero, and the cart ± 10 cm from the centre:

$$\mathbf{Q}_{(2,2)} = \mathbf{Q}_{(4,4)} = \frac{1}{0.05^2} = 400 \quad (8.21)$$

$$\mathbf{Q}_{(1,1)} = \mathbf{Q}_{(3,3)} = \frac{1}{0.1^2} = 100 \quad (8.22)$$

The maximum deviation of input is set to 10 N which results in the following weight

$$\mathbf{R} = \frac{1}{10^2} = 0.01 \quad (8.23)$$

By choosing \mathbf{Q} to be diagonal individual weighing on the states are achieved. Using the MATLAB function `lqr` with the above weights, the gain vector is found as

$$\mathbf{k} = \begin{bmatrix} -100 & 1012 & -172.1 & 266.8 \end{bmatrix} \quad (8.24)$$

To ensure stability, the eigenvalues of the closed-loop system $\mathbf{A} - \mathbf{Bk}$ must have negative real parts, and are found by solving $\det(\lambda\mathbf{I} - (\mathbf{A} - \mathbf{Bk})) = 0$ for λ , which gives

$$\lambda = \begin{bmatrix} -127.86 & -1.68 \pm 0.014i & -0.01 \end{bmatrix} \quad (8.25)$$

A simulation run with the above control gives the results in **figure 8.4**.

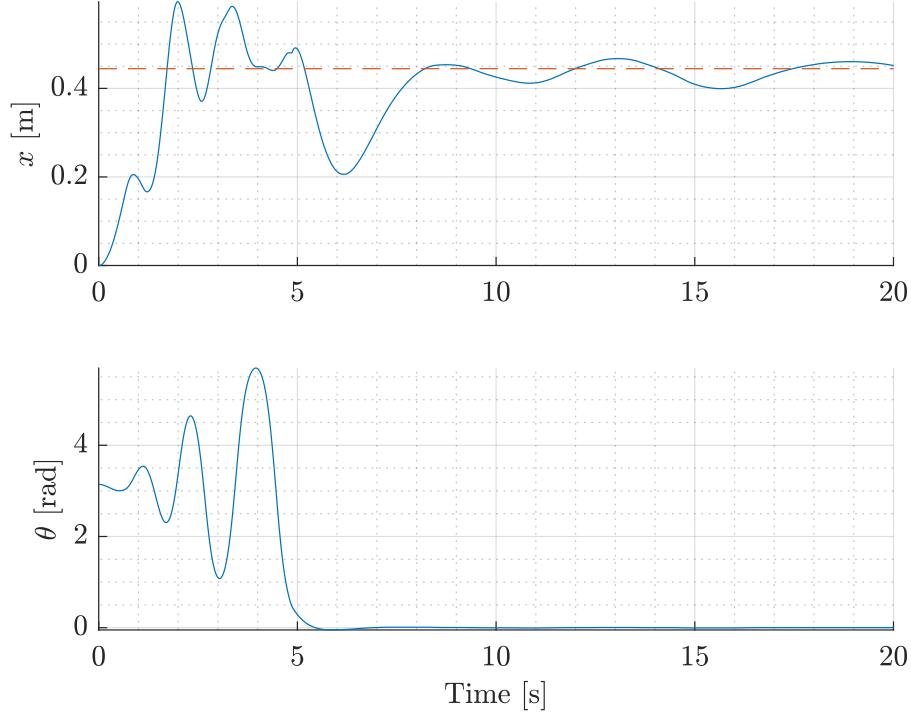


Figure 8.4: LQG simulated on the full nonlinear model. The dotted red line in the top figure indicates the cart reference, x^* .

The MSE of the cart position and pendulum angle, from 10 seconds until the end of the simulation is

$$\text{MSE}_c = 521.9 \cdot 10^{-6} \text{ m}$$

$$\text{MSE}_p = 9.460 \cdot 10^{-6} \text{ rad}$$

The cart moves ± 3 cm around the reference, and the pendulum is ± 0.005 rad from being completely upright, which the low MSE results indicate.

8.3 Sliding-Mode Controller

The following is based on [4].

The SMC is a nonlinear controller which is characterised by its way of leading the system states towards a *sliding manifold* and then along it until the manifold has converged to zero, as illustrated in figure 8.5.

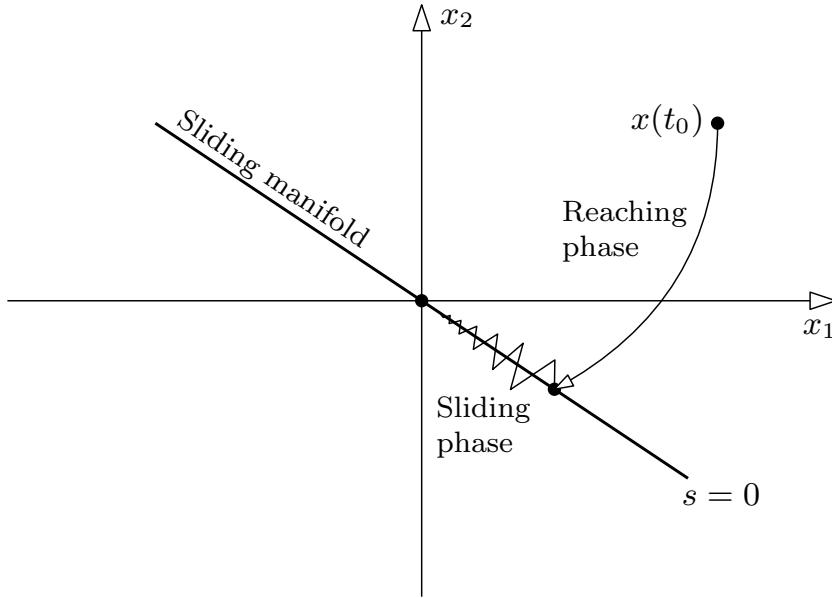


Figure 8.5: Sliding-mode controller principle

This is achieved by use of a system model, such that the system dynamics are compensated for, and the sliding manifold can be reached in finite time.

Considering the inverted pendulum system state vector $\mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4]^T = [x \ \theta \ \dot{x} \ \dot{\theta}]^T$ with $n = 4$ states and $p = 1$ input, the sliding mode control design is based on the coordinate transformation $T : D \rightarrow R^n$

$$T(\mathbf{x}) = \begin{bmatrix} x_2 \\ x_3 \frac{\cos x_2}{l} - x_4 \\ x_1 - x^* \\ \hline x_4 \end{bmatrix} = \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \\ \hline \xi \end{bmatrix} = \begin{bmatrix} \eta \\ \hline \xi \end{bmatrix}, \quad \eta \in \mathbb{R}^{n-p}, \xi \in \mathbb{R}^p \quad (8.26)$$

$T(\mathbf{x})$ transforms the system from the form in **equation (4.40)**, i.e.

$$\begin{aligned}\dot{\mathbf{x}} &= \begin{bmatrix} \dot{x} \\ \dot{\theta} \\ -\alpha(\mathbf{x}) - \beta(\mathbf{x}) \\ -\gamma(\mathbf{x}) - \delta(\mathbf{x}) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ (m_c + m_p \sin^2 \theta)^{-1} \\ \cos \theta (l(m_c + m_p \sin^2 \theta))^{-1} \end{bmatrix} u = \begin{bmatrix} \dot{x} \\ \dot{\theta} \\ f_c(\mathbf{x}) \\ f_p(\mathbf{x}) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g_c(\mathbf{x}) \\ g_p(\mathbf{x}) \end{bmatrix} u \\ &= f(\mathbf{x}) + g(\mathbf{x})u\end{aligned}\quad (8.27)$$

into the *regular form*:

$$\begin{bmatrix} \dot{\eta} \\ \dot{\xi} \end{bmatrix} = \begin{bmatrix} f_a(\mathbf{x}) \\ f_b(\mathbf{x}) + g_b(\mathbf{x})u \end{bmatrix} = \frac{\partial T(\mathbf{x})}{\partial \mathbf{x}} f(\mathbf{x}) + \frac{\partial T(\mathbf{x})}{\partial \mathbf{x}} g(\mathbf{x})u \quad (8.28)$$

with

$$\frac{\partial T(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{\sin x_2}{l} & \frac{\cos x_2}{l} & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8.29)$$

$$\frac{\partial T(\mathbf{x})}{\partial \mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} x_4 \\ -x_4 \frac{\sin x_2}{l} + \beta(\mathbf{x}) \left(\frac{1 + \frac{m_c}{m_p} - \cos^2 \theta}{\cos \theta} \right) \\ x_3 \\ f_p(\mathbf{x}) \end{bmatrix} \quad (8.30)$$

$$\frac{\partial T(\mathbf{x})}{\partial \mathbf{x}} g(\mathbf{x}) = \begin{bmatrix} 0 & 0 & 0 & g_p(\mathbf{x}) \end{bmatrix}^T \quad (8.31)$$

In (η, ξ) -coordinates, $f_a(\eta, \xi)$ and $f_b(\eta, \xi)$ then become:

$$f_a(\eta, \xi) = \begin{bmatrix} \xi \\ -\xi \frac{\sin \eta_1}{l} + \beta(\eta, \xi) \left(\frac{1 + \frac{m_c}{m_p} - \cos^2 \eta_1}{\cos \eta_1} \right) \\ \frac{l}{\cos \eta_1} (\eta_2 + \xi) \end{bmatrix} \quad (8.32)$$

$$f_b(\eta, \xi) = f_p(\eta, \xi) \quad (8.33)$$

From this, a sliding manifold $s = \xi - \phi(\eta) = 0$ is designed such that

$$\dot{\eta} = f_a(\eta, \phi(\eta)) \quad (8.34)$$

has an asymptotically stable equilibrium point at the origin when motion is restricted to the manifold.

A linear design of $\phi(\eta)$, i.e.

$$\phi(\eta) = -\mathbf{k}\eta, \quad \mathbf{k} = [k_1 \ k_2 \ k_3] \quad (8.35)$$

is achieved by pole placement for the closed-loop system in regular form, linearised in $\eta = \xi = 0$, that is

$$\dot{\eta}_{lin} = \mathbf{A}\eta + \mathbf{B}\phi(\eta) \quad (8.36)$$

with

$$\mathbf{A} = \frac{\partial f_a(\eta, \xi)}{\partial \eta} \Big|_{\eta=\xi=0} = \begin{bmatrix} 0 & 0 & 0 \\ -\frac{g}{l} & 0 & 0 \\ 0 & l & 0 \end{bmatrix} \quad (8.37)$$

$$\mathbf{B} = \frac{\partial f_b(\eta, \xi)}{\partial \xi} \Big|_{\eta=\xi=0} = \begin{bmatrix} 1 & \frac{v_{p,lin}}{l^2 m_p} & l \end{bmatrix}^T \quad (8.38)$$

which is controllable as its controllability matrix has full rank. Note that for the computation of \mathbf{A} and \mathbf{B} , the system has been assuming to only have viscous friction terms, hence $v_{p,lin}$, in order to avoid dependency on the slope coefficient k in $\tanh(k\dot{\theta})$.

The desired closed-loop poles are chosen to be -1 and $-4 \pm 2.4j$, and found via the MATLAB-function `place`, which yields the gain vector

$$\mathbf{k} \approx [10 \ -1 \ -2] \quad (8.39)$$

From this, the sliding manifold becomes

$$\begin{aligned} s &= \xi - \phi(\eta) = \xi + \mathbf{k}\eta \\ &= \xi + k_1\eta_1 + k_2\eta_2 + k_3\eta_3 \\ &= \dot{\theta} + k_1\theta + k_2 \left(\dot{x} \frac{\cos \theta}{l} - \dot{\theta} \right) + k_3(x - x^*) \end{aligned} \quad (8.40)$$

To find the sliding-mode control law, the time derivative of **equation (8.40)** is taken in order to obtain an explicit dependency of the input u :

$$\begin{aligned} \dot{s} &= \ddot{\theta} + k_1\dot{\theta} + k_2 \frac{d}{dt} \left(\dot{x} \frac{\cos \theta}{l} - \dot{\theta} \right) + k_3(\dot{x}) \\ &= \ddot{\theta} + k_1\dot{\theta} + k_2 \left(\ddot{x} \frac{\cos \theta}{l} - \dot{x} \frac{\sin \theta}{l} - \ddot{\theta} \right) + k_3(\dot{x}) \\ &= f_p(\mathbf{x}) + g_p(\mathbf{x})u + k_1\dot{\theta} + k_2 \left((f_c(\mathbf{x}) + g_c(\mathbf{x})u) \frac{\cos \theta}{l} - \dot{x} \frac{\sin \theta}{l} - f_p(\mathbf{x}) - g_p(\mathbf{x})u \right) + k_3(\dot{x}) \\ &= \underbrace{(1 - k_2)f_p(\mathbf{x}) + k_2f_c(\mathbf{x})\frac{\cos \theta}{l} + k_1\dot{\theta}}_{\Phi} + \underbrace{\left(k_3 - k_2 \frac{\sin \theta}{l} \right) \dot{x} + \left((1 - k_2)g_p(\mathbf{x}) + k_2g_c(\mathbf{x})\frac{\cos \theta}{l} \right) u}_{\Gamma} \end{aligned} \quad (8.41)$$

The control law is then defined as

$$u = -\Gamma^{-1}(\Phi + \lambda \operatorname{sgn}(s)), \quad \lambda = \left| \frac{\Phi}{\Gamma} \right| + \lambda_0, \quad \lambda_0 > 0 \quad (8.42)$$

In order to validate the stability of this control law, consider the Lyapunov function candidate

$$V(s) = \frac{1}{2}s^2 \quad (8.43)$$

with time derivative

$$\begin{aligned} \dot{V}(s) &= s\dot{s} \\ &= s(\Phi + \Gamma u) \\ &= s(\Phi - \Gamma\Gamma^{-1}(\Phi + \lambda \operatorname{sgn}(s))) \\ &= s(\Phi - \Phi - \lambda \operatorname{sgn}(s)) \\ &= -s\lambda \operatorname{sgn}(s) \end{aligned} \quad (8.44)$$

Since $\operatorname{sgn}(s) = |s|s^{-1}$ for $s \neq 0$,

$$\dot{V}(s) = \begin{cases} -|s|\lambda < 0 & \text{for } s \neq 0 \\ 0 & \text{for } s = 0 \end{cases} \quad (8.45)$$

which, according to **theorem 6.1.1** (Lyapunov's stability theorem), shows that $s = 0$ is an asymptotically stable equilibrium point. Thus, trajectories starting outside $s = 0$ will reach the manifold in finite time and stay there. A simulation using this control is shown in **figure 8.6**.

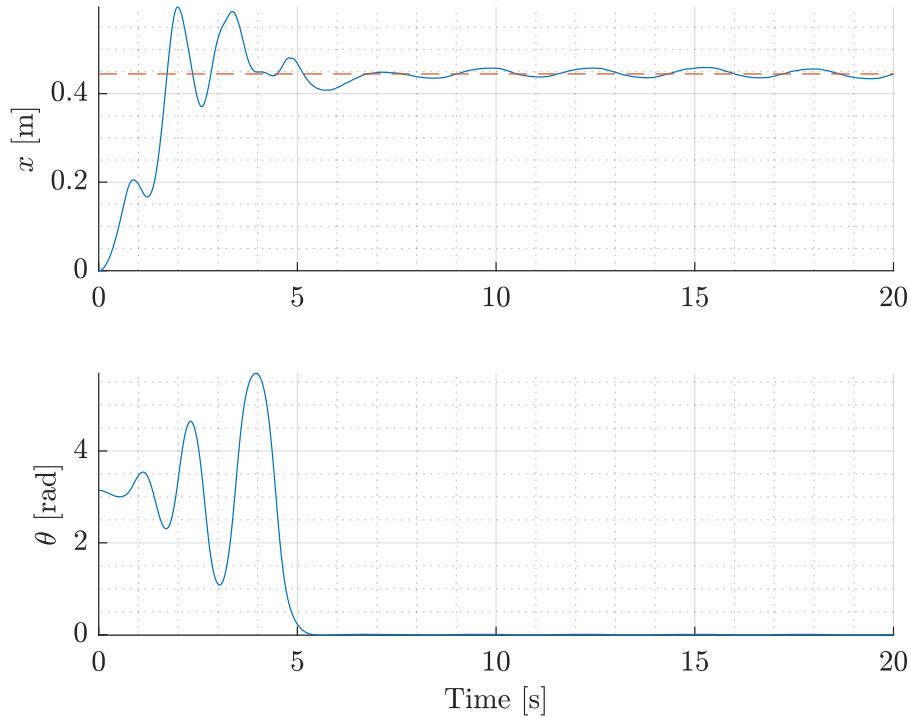


Figure 8.6: Sliding-mode simulation results on the full nonlinear model. The dotted red line in the top figure indicates the cart reference, x^* .

The MSE of the cart position and pendulum angle, from 10 s until the end of the simulation is

$$\text{MSE}_c = 65.51 \cdot 10^{-6} \text{ m}$$

$$\text{MSE}_p = 14.33 \cdot 10^{-6} \text{ rad}$$

Implementation 9

This chapter describes some core aspects regarding the implementation of the data filtering and control. In **section 9.1**, this is considered for the Simulink-model, while **section 9.2** focuses on the Arduino-implementation and verification of the EKF.

9.1 Simulink

This section describes implementation of the EKF and developed control laws in Simulink. The content of the block *MCU loop* from **figure 4.12**, page 22, is shown in **figure 9.1**.

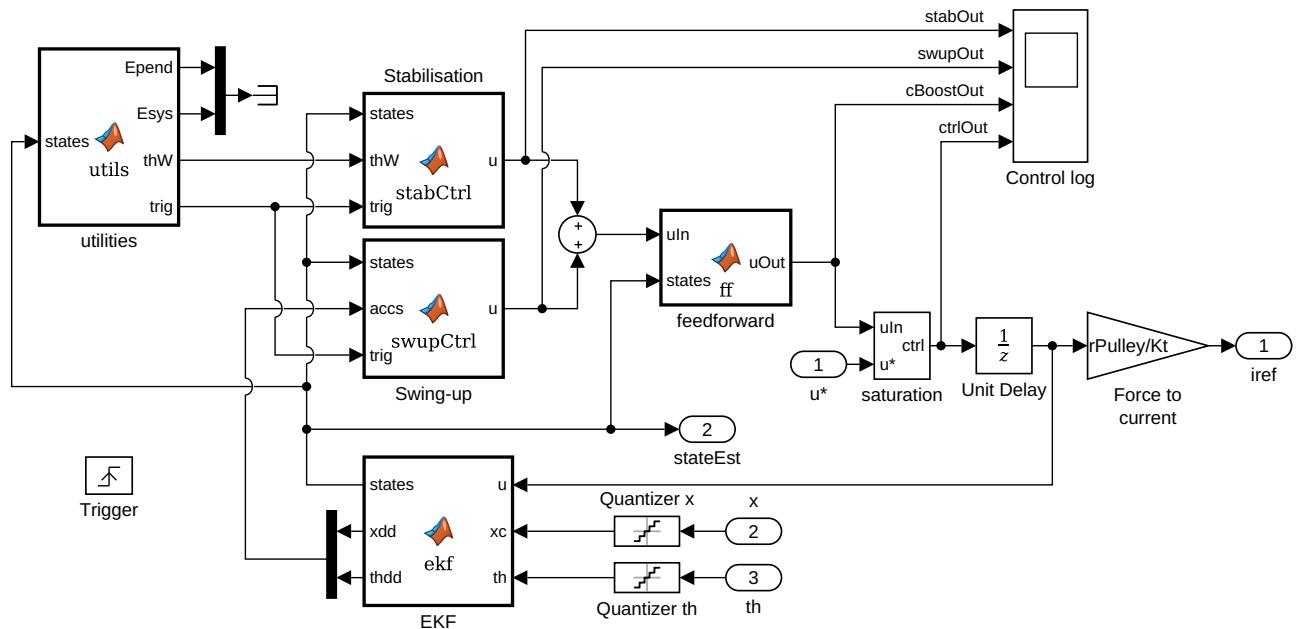


Figure 9.1: Simulink model file *swingUpModel.slx*, MCU loop subsystem.

As mentioned in **chapter 6**, only one controller is active at a time; either the swing-up or the stabilising controller. This is important to emphasise since their outputs in **figure 9.1** are summed before entering the friction feedforward block, *feedforward*.

The two controller blocks, *Stabilisation* and *Swing-up*, only perform a simple action which is a call to an .m-file which calculates the desired control signal. With this structure, changes can be made in the control calculations without changing the Simulink model, which is useful when multiple people develop using e.g. git version control and need to make changes in the same file. These are the only blocks with this configuration.

The *EKF* block is self-explanatory; it receives the quantised measurements of the cart position and pendulum angle, and outputs the state estimates along with the model-based accelerations.

The block *utilities* computes the energies in the system, and the catch trigger algorithm, thus forward-

ing a signal, $trig$, which activates either the swing-up or stabilising controller.

Another important feature within the utilities block is the calculation of a wrapped pendulum angle, θ_w , used by the stabilising controllers. This angle is wrapped between $\pm\pi$ to compensate for the relative optical encoding. If the angle is not wrapped and the pendulum happens to do a full revolution before the catch is triggered, the stabilising controller will attempt to correct this very large error, relative to zero, by producing a large control signal. This results in rapid acceleration until the physical limit of the rail is hit.

9.2 Arduino

The implementation on the Arduino is based on the structure of the Simulink implementation. The code written for the Arduino consists of several files and libraries:

- `main.ino`
The main file of the Arduino program which uses a switch-case structure as depicted in **figure 9.2**.
- `ctrl.c / ctrl.h`
A C library which calculates the control action of both the swing-up and stabilising control laws derived in **chapters 7 and 8**. It also manages the friction feedforward, the catch trigger algorithm and calculating θ_w .
- `EKF.cpp / EKF.h`
A C++-library to compute state estimations, implemented using the Eigen template library which supports matrix computations[18].
- `Joint.cpp / Joint.h`
A provided C++ library in which low-level interaction between the hardware and the control etc. is implemented. It has been modified to operate in SI units.
- `params.h`
A header file containing system-wide parameters.

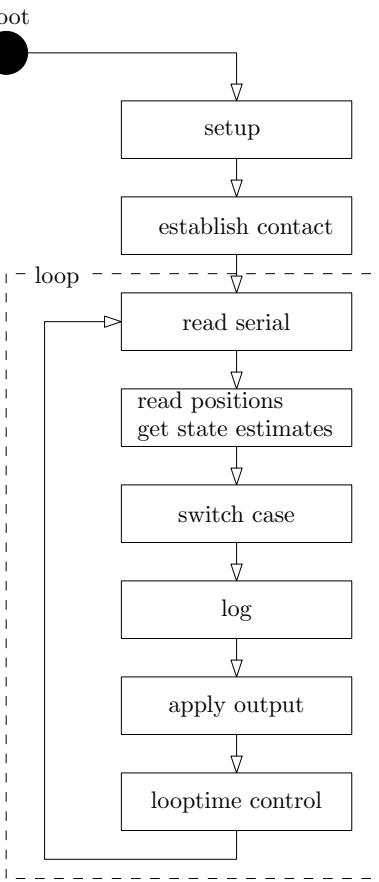


Figure 9.2: State diagram of the Arduino code.

When booted, the Arduino runs a setup protocol which creates and initializes two Joint C++ objects, *sled* and *pendulum*, and an EKF object, using the Joint and EKF library, respectively. It also initializes its peripherals, and enables the motor drivers. Then the Arduino goes into a waiting state, *establish contact*, where it listens on the serial line for a command.

When any command is received, it goes into the main loop, which is structured as follows:

- 1) In state *read serial*, any incoming command is interpreted.

The available commands are defined as follows:

L	Enable/disable data logging.
swup=x	Set swing-up controller type to x.
stab=x	Set stabilising controller type to x.
1/0	Enable/disable control.
r	Reset encoders and EKF.

- 2) Positions are read using the *Joint.cpp*-library and states are estimated.
- 3) In the switch case, commands 0/1, L and r are carried out.
If the control action is enabled, a control signal is computed.
- 4) If the log enabled, a string is created and written to the serial line.
- 5) The control signal is applied to the motor, if control action is enabled.

- 6) A function ensures that each loop has a looptime of 5 ms.
- 7) Loops back to reading the serial.

9.2.1 Parameter tuning

The directional Coulomb friction for the cart, $c_{c,R}$ and $c_{c,L}$, have been considered as tuning parameters which is justified by their inconsistency. The implemented values are $c_{c,R} = 3.3$ and $c_{c,L} = 2.8$.

The weights of \mathbf{Q} and \mathbf{R} of the EKF has been tuned to obtain satisfactory results. Since measurement noise is lower than the quantization of the measurement signal, the weights have been tuned such that the estimate and measurement of x and θ are identical, as can be seen in figure 9.3. The new weights are

$$\mathbf{R} = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.1 \end{bmatrix} \quad (9.1)$$

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 50 & 0 \\ 0 & 0 & 0 & 50 \end{bmatrix} \quad (9.2)$$

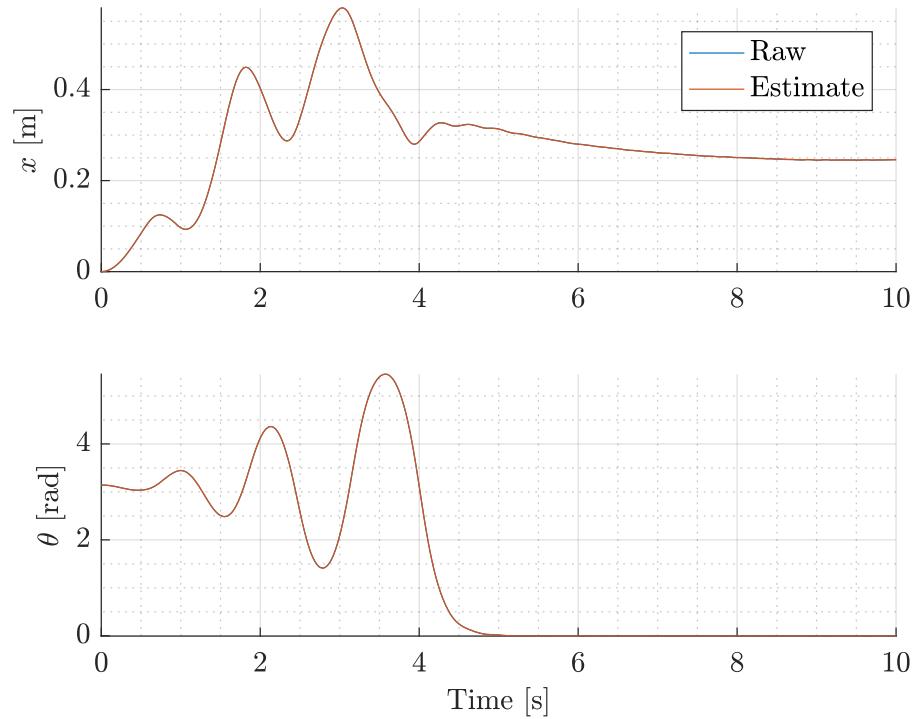


Figure 9.3: Comparison between position estimates from the EKF and raw measurements.

The estimates of the velocities are compared to a numerical derivative of the positions in **figure 9.4**.

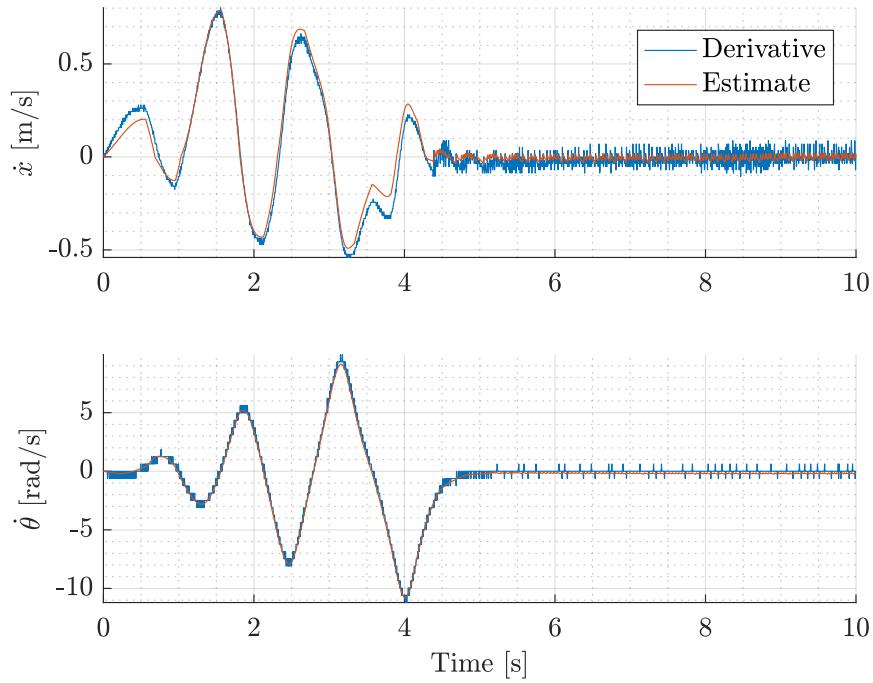


Figure 9.4: Comparison between velocity estimates from the EKF and the numerical derivative of the measured positions.

Finally, **figure 9.5** shows a comparison between a system test and an equivalent simulation using the same control configuration, namely the pendulum energy method for swing-up and LQG for stabilisation.

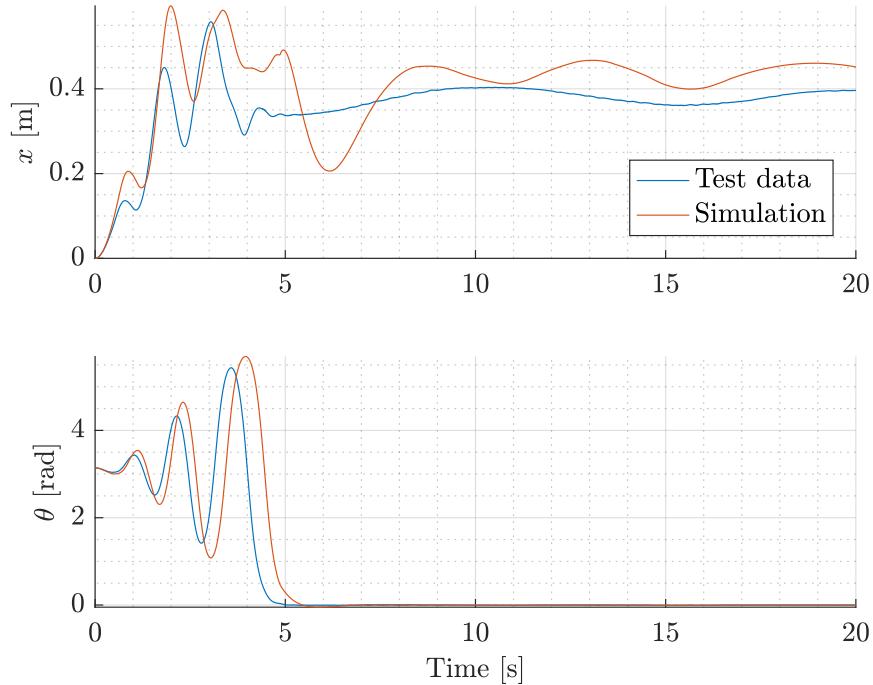


Figure 9.5: Comparison between the implementation of the pendulum energy swing-up method using the LQG for stabilisation, and a simulation using the same control methods.

In this chapter, the results of this project will be presented. The chapter will be divided into three sections, each of which will deal with the results from a test related to the swing-up phase, the stabilisation and robustness, respectively.

10.1 Swing-up control

Tests are conducted to compare the three proposed methods for swinging up the pendulum, namely the *full energy method* described in [section 7.3.1](#), the *pendulum energy method* described in [section 7.3.2](#) and finally the *sign-based method* described in [section 7.3.3](#). [Figure 10.1](#) shows the resulting data from the tests.

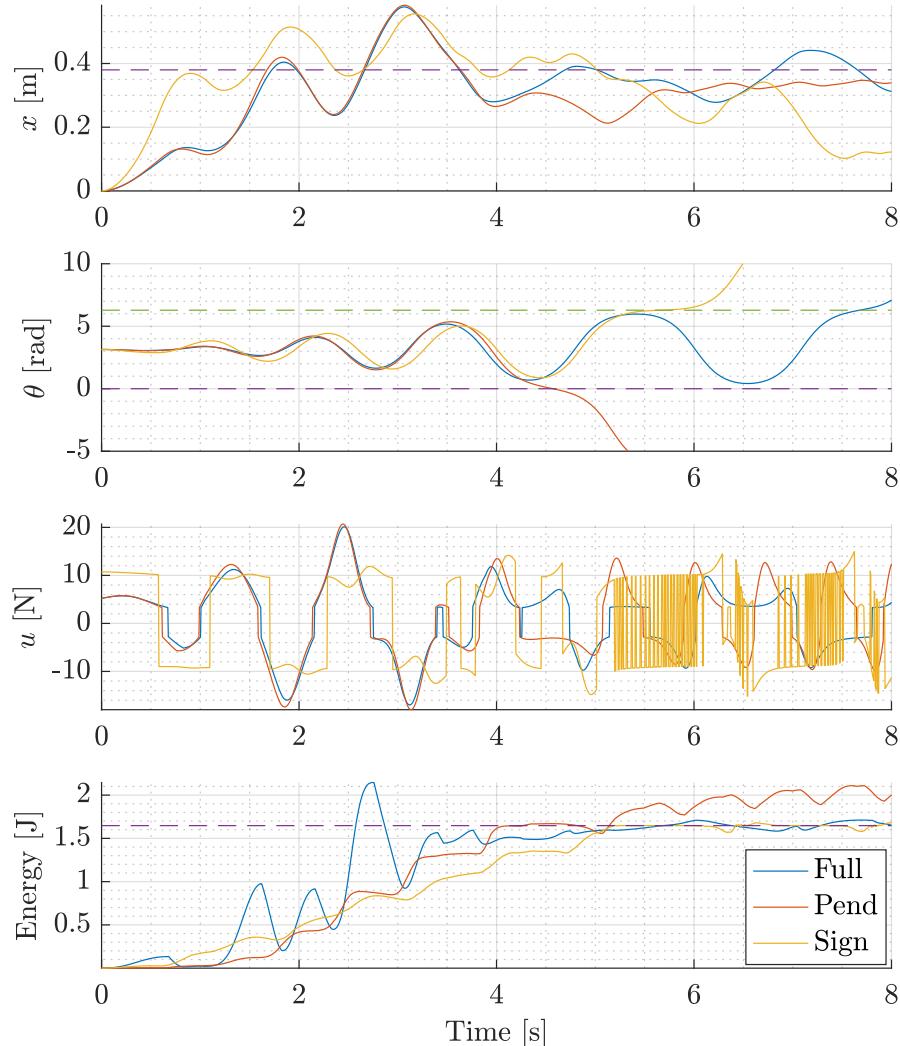


Figure 10.1: Results from the swing-up control tests. Dashed lines indicate reference values. The subplots from top to bottom are: Cart position, pendulum angle, controller output, system/pendulum energy.

The inverted pendulum is initialised in the leftmost position of the rail, hanging straight down, and the stabilising controller has been deactivated.

In terms of the cart position, all three methods keep the cart within the physical limits of the rail, though not exactly at the reference.

Additionally, all three methods manage to reach the unstable equilibria at angles zero and 2π , but due to the integral action they exactly swing beyond these points and perform a full revolution.

Considering the energy, all three methods manage to increase the energy in question to the reference value, although the pendulum energy method does overshoot. However, it also maintains an energy very close to the reference for about a second at $t = 4$ to $t = 5$, which is sufficient for the stabilising controller to catch the pendulum.

Finally, from the controller outputs u , it is evident how the sign-based method switches between ± 10 , in contrast to the two other methods.

10.2 Stabilising control

For stabilisation, a PID-controller, and LQG-regulator and a SMC have been derived in **sections 8.1 to 8.3**, respectively.

In order to test them, the inverted pendulum has been started with the same method for swing-up, namely the pendulum energy method. When the catch-trigger criteria described in **section 6.4** are met, the stabilising control is used.

The results from this test are shown in **figures 10.2 and 10.3**.

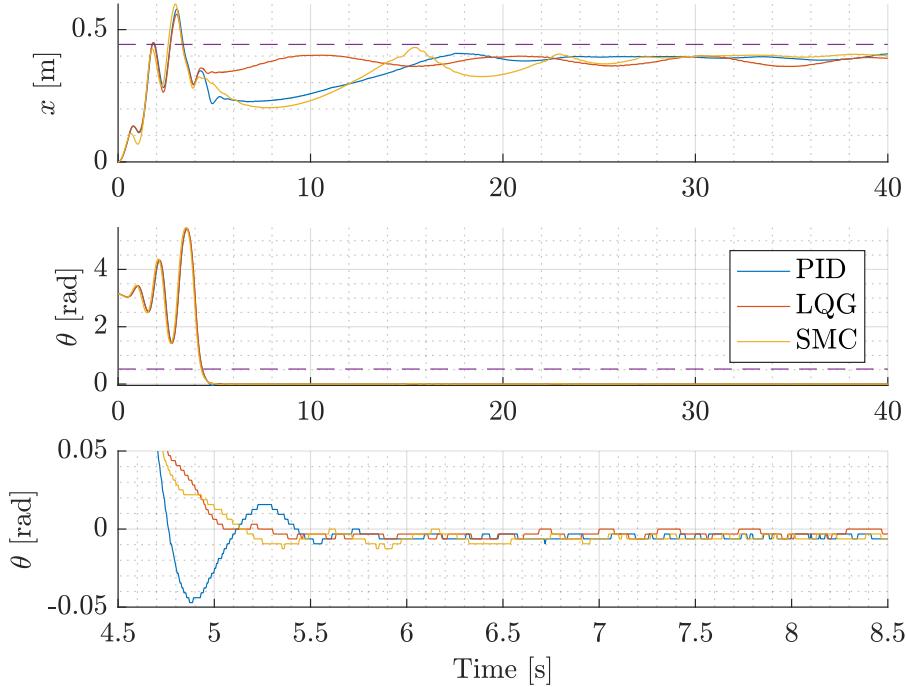


Figure 10.2: Results from testing the stabilising controllers, all using the pendulum swing-up methods. The subplots from top to bottom are: Cart position, pendulum angle, pendulum angle at stable state.

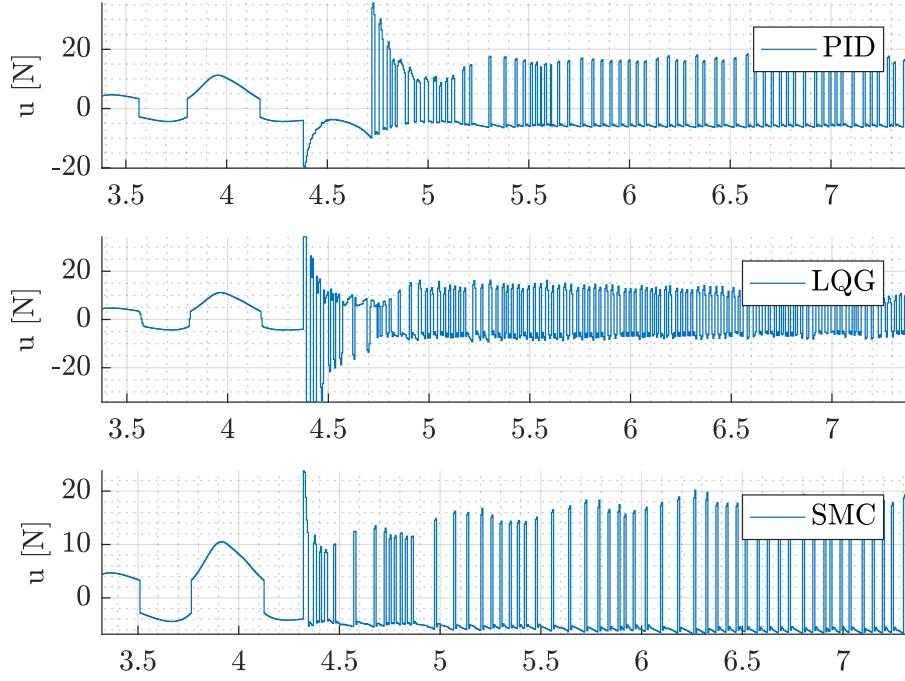


Figure 10.3: Controller outputs from the test of the stabilising controllers, with zoom on the switch between the swing-up and stabilising phases.

This test shows that all three stabilising controllers are capable of keeping the pendulum in the upright position. However, the PID tends to oscillate more than the two others. The cart is kept near the reference position, though with a small steady-state error and some oscillation.

In the controller outputs in **figure 10.3**, it is seen that the controllers all tend to chatter. This is due to the fact that the controllers must be aggressive to keep the pendulum upright, and because of the feedforward, as described in **section 6.3**. The sliding-mode control has larger peaks in the chattering, but also for short durations at a time.

Similar to the procedure in the respective ends of **sections 8.1** to **8.3**, the performance of the stabilising control at steady-state is quantified by calculating the MSEs over the final ten seconds of the data:

	PID	LQG	SMC	Unit
MSE_c	$2.640 \cdot 10^{-3}$	$4.120 \cdot 10^{-3}$	$1.811 \cdot 10^{-3}$	m
MSE_p	$20.99 \cdot 10^{-6}$	$14.00 \cdot 10^{-6}$	$16.29 \cdot 10^{-6}$	rad

10.3 Robustness

Finally, as stated in **section 3.1.2** (Additional requirements), the stabilising controllers must ensure some robustness with respect to variations in the pendulum tip-mass.

In order to test this, the tip-mass has been reduced by 26 g and 50 g, which corresponds to a parameter variation of approximately 10 and 20 percent.

The test is conducted using the pendulum energy method for swing-up. The results are shown in

figures 10.4 and 10.5.

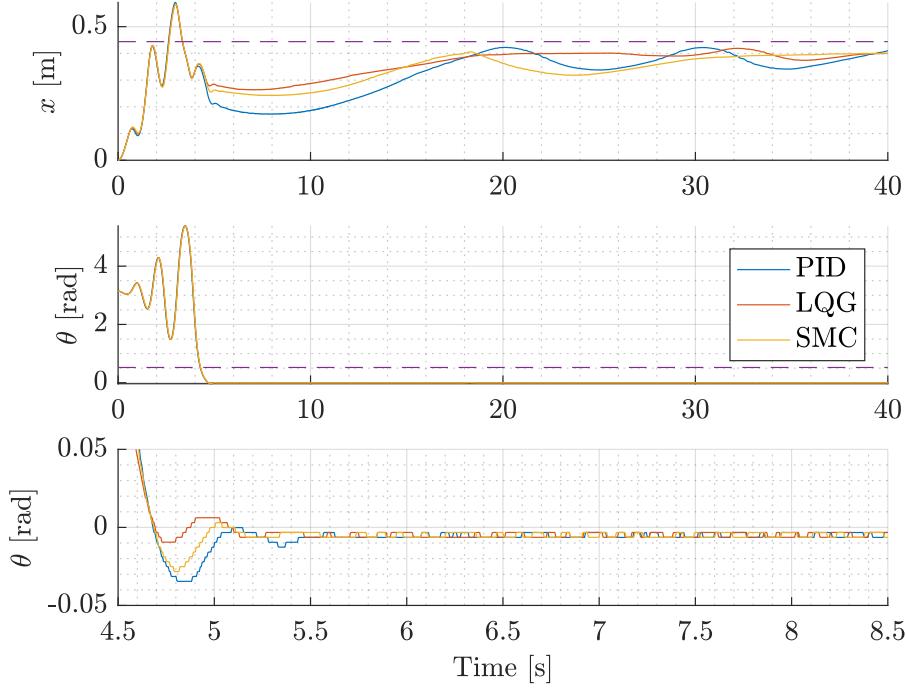


Figure 10.4: Results of robustness test with 26 g removed from the tip mass. The pendulum energy methods is used for swing-up. Dashed lines indicates references.

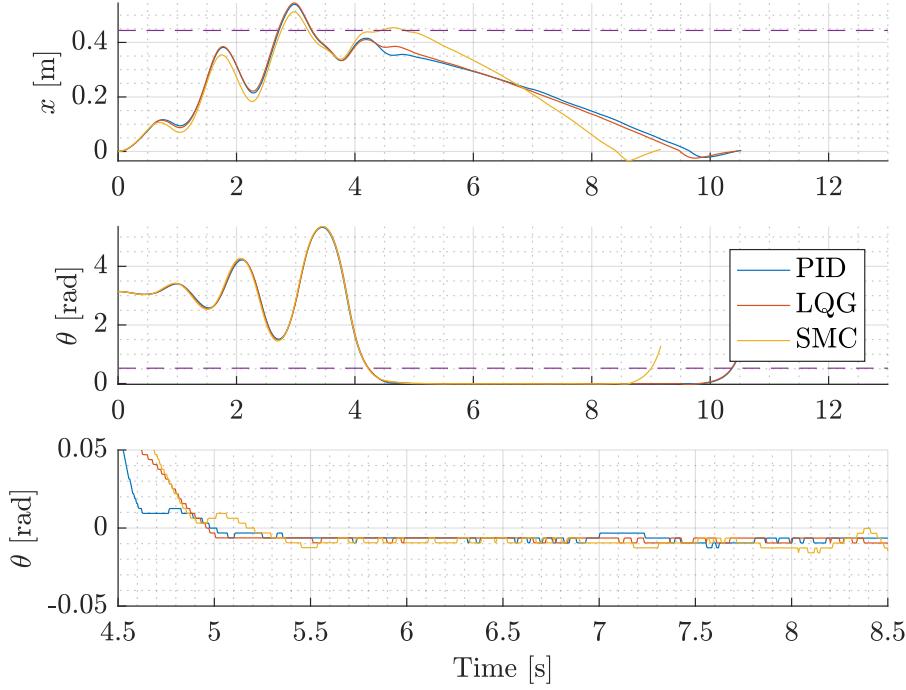


Figure 10.5: Results of robustness test with 50 g removed from the tip mass. The pendulum energy methods is used for swing-up. Dashed lines indicates references.

From this, it is seen that all controllers manage to stabilise the system when just 26 g is removed, though the LQG and SMC do oscillate more in the process, when compared to figure 10.2. The PID, on the other hand, seems to oscillate less in θ , but more in x .

Finally, when 50 g is removed, all of the controllers manage to stabilise the pendulum, but in doing so cause the cart to accelerate and hit the end of the rail.

In this chapter, several topics presented throughout the thesis are discussed.

Cart friction

As presented in **section 4.3** (Cart and pendulum frictions), the final model of the friction in the system consists of a Coulomb model for the cart, and a Coulomb and viscous model for the pendulum. Several attempts to derive a more precise model of the cart friction are described in **appendix B** and **appendix C**, but discarded due to inconclusive results. These tests and analyses were made in the beginning of the project period, and measures to minimise friction in the system were attempted afterwards, as described in **section 2.4** (Rail and cart observations).

Better results may have been obtained by deriving a friction model based on e.g. the Dahl Model, or the Bristle Model. The Dahl Model has been used for adaptive friction compensation in servo system with ball bearings, and models the stress-strain curve by a differential equation. The Bristle Model attempts to model the behaviour of microscopical contact points between two surfaces as flexible bristles with a stiffness. When the surfaces move relative to each other, the strain in the bristles increases, which makes them act springs, giving rise to a friction force. [19]

In terms of minimising friction, a mechanical reconstruction of the system has been considered. The system is built with an intention of adding another pendulum to the cart. Because of this, the cart is larger than necessary when only supporting one pendulum. Minimising the cart size, thus mass, will theoretically decrease the friction between the cart and rail. Decreasing the number of points at which the cart is fastened to the rail decreases the contact surface and thus friction.

Addressing the way the cart is actuated through a toothed belt, it may be beneficial in terms of friction to implement another type of power transmission, e.g. based on a wire or pneumatics.

Aerodynamic drag

In the model derivation it is assumed that torque produced by aerodynamic drag on the pendulum is negligible. This is addressed in **appendix F**, which shows that the produced torque is low compared to gravitational torque. However, at e.g. peak velocity of approximately 11 rad/s, this drag torque is larger than the friction torque. Including this may improve the accuracy of the pendulum model, and thus the control can be designed to compensate for it.

Control design

As presented in **section 7.1** (Pendulum energy method) and **section 7.2** (Pendulum energy method), two energy based swing-up methods are considered. These are based on a simplified system, that is, without frictions. The results in the respective sections indicate that the heteroclinic orbits and thus reference energy is not reached. This can be due omitting friction in the design, hence the control laws are not able to compensate for energy lost to friction.

Since heuristics are applied to the swing-up methods, the application of **theorem 6.1.2** (LaSalle's theorem) invalidated, and it is thus possible that the pendulum may exit the heteroclinic orbits or equivalently overshoot on the energy. However, the catch trigger algorithm provides a smooth transition between the swing-up and stabilising controllers, as seen in **chapter 10** (Results). In general, when experimenting with the catch angle during implementation, the stabilising controllers have been seen to struggle when the catch angle is less than ± 10 degrees.

Finally, during implementation of the stabilising controllers, the tuning process of the PID and LQG required more time to achieve satisfactory results. As opposed to this, the SMC required less, but it was more cumbersome to design.

Sensors

During implementation, an uncertainty has been found when initialising the optical encoder measuring the pendulum angle, as described in **appendix H**. When initialising the pendulum, resting at $\theta = \pi$, the encoder tends to come to rest at small values from a true, vertical position, presumably due to the Coulomb friction in the pendulum.

This could be optimized by using an absolute encoder, rather than the relative one, and calibrate it carefully, or by adding another sensor to the setup, e.g. an accelerometer. Having an extra sensor for the same measurement allows calibration of the encoder. Another solution could be to dismount the unused motor to which the pendulum is attached, and install a stand alone encoder directly on the pendulum, assuming the friction originates from this.

In addition to this, an accelerometer on the cart could be used to verify the assumption that the belt is non-elastic.

Implementation

The model is implemented using MATLAB's Simulink environment, as is described in **section 4.5** (Simulink implementation). The development during this project, either documenting or coding, has been managed with git version control. Unfortunately, Simulink's .slx files are not compatible with this, causing files to be overwritten simultaneously by multiple developers, since the files cannot be merged. This could have been avoided by using Simulink Projects, a team collaborative design platform integrated with source control. [20]

The code written for the Arduino Due has been structured in C and C++ libraries as mentioned in **chapter 9** (Implementation). Implementation of the EKF computations requires matrix calculations, for which the Eigen-library is used, as this is not natively supported on the Arduino. These calculations require roughly half of the available processor time. If more efficient EKF computations are desired, alternatives such as C-code generation in MATLAB, using the MATLAB Coder, can be considered.[21]

Conclusion 12

In this thesis, the topic of nonlinear control is applied to an inverted pendulum on a cart, and used to swing up and stabilise the pendulum, with the purpose of comparing different methods, both linear and nonlinear, and their performance.

The control scheme has been based on a two-step strategy, containing a swinging phase and stabilising phase. The transition between these phases are determined by a *catch trigger* algorithm with requirements on the pendulum angle and mechanical energy.

Three different nonlinear swing-up controllers have been developed, implemented, and compared. Two of the swing-up controllers are based on the energy of the pendulum, named *pendulum energy method* and *sign-based method*, and the third is based on the energy of the entire system, named *full energy method*. All of them are able to swing up the pendulum to its upright position, reaching the heteroclinic orbits of the pendulum, and meet the requirements stated in **section 3.1.2** (Additional requirements).

After implementation and testing, it has been found that the pendulum energy method provided best results, and is therefore used for further testing with the stabilising controllers. However, it is also found that the sign-based swing-up holds an advantage, as it does not need integral action on the energy error to reach the heteroclinic orbits, as the two other methods do. The integral action, while correcting the energy error, does also pose a risk to overshoot the energy reference, and thus exceed the heteroclinic orbits.

For stabilisation, three control schemes have been developed, implemented and compared: A nonlinear Sliding-Mode Controller, a Linear-Quadratic-Gaussian regulator, and a Proportional-Integral-Derivative controller. All of them are able to catch and stabilise the pendulum upright while keeping the cart close to its given reference. The results indicate minimal difference in performance in terms of stabilising the pendulum, and all the controllers meet the requirements stated in **section 3.1.2** (Additional requirements).

Finally, the stabilising controllers provide robustness with respect to 10% changes in the pendulum tip-mass.

Conclusively, system is able to autonomously swing up and stabilise the pendulum while keeping the cart safely within the physical limits of the test setup, and thus fulfils the requirements in **section 3.1.1** (Functional requirements).

Bibliography

- [1] Stav Ziv. "Timeline: A Brief History of SpaceX's Reusable Rocket Launches". In: *Newsweek* (Jan. 20, 2016). URL: [Timeline : %20A%20Brief%20History%20of%20SpaceX%E2%80%99s%20Reusable%20Rocket%20Launches](#) (visited on 06/07/2016).
- [2] SpaceX. *Official SpaceX Photos*. URL: <https://www.flickr.com/photos/spacex/> (visited on 06/05/2017).
- [3] Rhett Allain. "Why is it so difficult to land a rocket?" In: *Wired Magazine* (Jan. 19, 2015). URL: <https://www.wired.com/2015/01/difficult-land-rocket/> (visited on 06/07/2017).
- [4] H.K. Khalil. *Nonlinear Systems*. Third edition. Pearson Education. Prentice Hall, 2002.
- [5] Nahum Shimkin. *Nonlinear Control Systems*. Lecture notes. Department of Electrical Engineering, Technion – Israel Institute of Technology, Haifa, Israel, 2009.
- [6] *maxon motor control*. 4-Q-DC Servoamplifier ADS 50/10. maxon motor. July 2009.
- [7] Arduino. *Arduino Due*. URL: <https://www.arduino.cc/en/Main/arduinoBoardDue> (visited on 02/21/2017).
- [8] *maxon DC motor*. RE 50. maxon motor. Apr. 2014.
- [9] *HEDM-55xx/560x and HEDS-55xx/56xx*. AV02-1046EN. Avago Technologies. Nov. 2014.
- [10] *HCTL-2001-A00, HCTL-2017-A00/PLC, HCTL-2021-A00 / PLC*. AV01-0041EN. Avago Technologies. Feb. 2006.
- [11] Simon J.A. Malham. *An introduction to Lagrangian and Hamiltonian mechanics*. Lecture notes. Aug. 2016.
- [12] Ivan Virgala and Michal Kelemen. "Experimental Friction Identification of a DC Motor". In: *International Journal of Mechanics and Applications* 3 (1 Mar. 1, 2013), pp. 26–30.
- [13] Gautam Vallabha. *Real-Time Pacer for Simulink*. 2016. URL: <https://mathworks.com/matlabcentral/fileexchange/29107-real-time-pacer-for-simulink>.
- [14] Torben Knudsen. *Kalman filter - for nonlinear systems*. Lecture notes. Aalborg University, 2016.
- [15] K. J. Åstrøm and K. Furuta. "Swinging Up a Pendulum by Energy Control". In: *IFAC World Congress* (13 June 30, 1996), p. 42.
- [16] Isabelle Fantoni Rogelio Lozano and Dan J. Block. "Stabilization of the inverted pendulum around its homoclinic orbit". In: *Systems and Control Letters* (40 Jan. 31, 2000), pp. 197–204.
- [17] Y Nishioka M Ishitobi Y Ohta and H Kinoshita. "Swing-up of a cart-pendulum system with friction by energy control". In: *Proceedings of the Institution of Mechanical Engineers* (218 Mar. 11, 2004), Part I:J. Systems and Control Engineering.
- [18] Christian Seiler Christoph Hertzberg Jitse Niesen. *Eigen*. URL: http://eigen.tuxfamily.org/index.php?title=Main_Page (visited on 06/02/2017).
- [19] K. J. Åstrøm, H. Olsson, C. Canudas de Wit, M. Gäfvert, and P. Lischinsky. "Friction Models and Friction Compensation". In: *Nonlinear and Adaptive Control: Towards a design methodology for Physical Systems* (Nov. 28, 1997), p. 37.

- [20] MathWorks. *What Are Simulink Projects?* URL: <https://se.mathworks.com/help/simulink/ug/what-are-simulink-projects.html> (visited on 06/01/2017).
- [21] MathWorks. *C Code Generation from Matlab.* URL: <https://se.mathworks.com/help/comm/ug/code-generation-from-matlab.html> (visited on 06/01/2017).
- [22] Carl Rod Nave. *Damped Harmonic Oscillator.* Feb. 28, 2017. URL: <http://hyperphysics.phy-astr.gsu.edu/hbase/oscda.html>.
- [23] Yunus A. Çengel. *Fundamentals of Thermal-Fluid Sciences.* 4th.
ISBN-13: 978-0077422400, ISBN-10: 0077422406. McGraw-Hill, Jan. 1, 2012.

Integer Value to Current A

This appendix describes the test conducted to determine the relation between the Arduino setpoint to a current running through the motor, i_a . This is done by utilizing the provided library *Joint.cpp*, and applying the motor several different step inputs.

A.1 System overview

As described in **chapter 2**, the circuit powering the motor is defined as in **figure A.1**

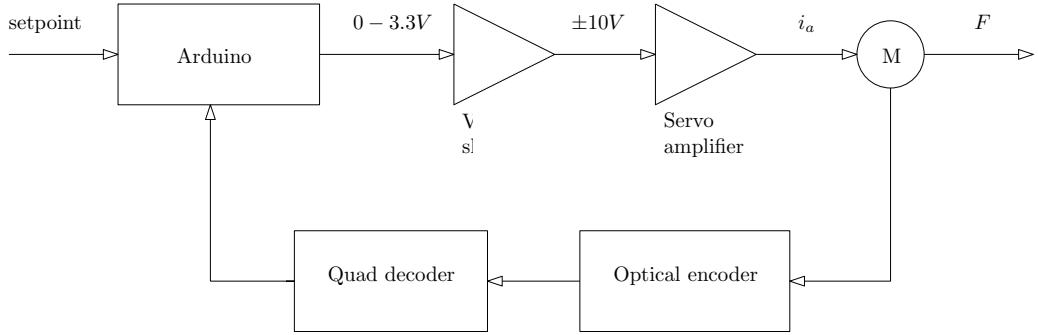


Figure A.1: The circuit powering the motor. The Arduino output is amplified to ± 10 V, which is fed into the maxon ServoAmplifier providing current to the motor.

The maxon servo amplifier allows for direct measurement of the current supplied to the motor it is running. It is therefore possible to construct a direct relationship between the setpoint in the Arduino to the motor current i_a , by simplifying **figure A.1** as depicted in **figure A.2**.

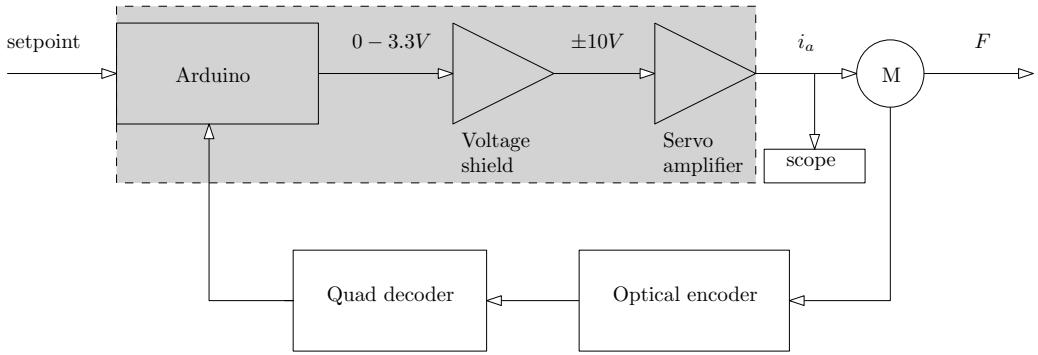


Figure A.2: Simplified electrical powering circuit. The voltage amplification stages are ignored, and the current is measured directly from the servo amplifier.

The output from the servo amplifier is a voltage with a gradient of approximately 0.4 V/A, thus a small conversion loss much be expected. **Figure A.3** displays the obtained results.

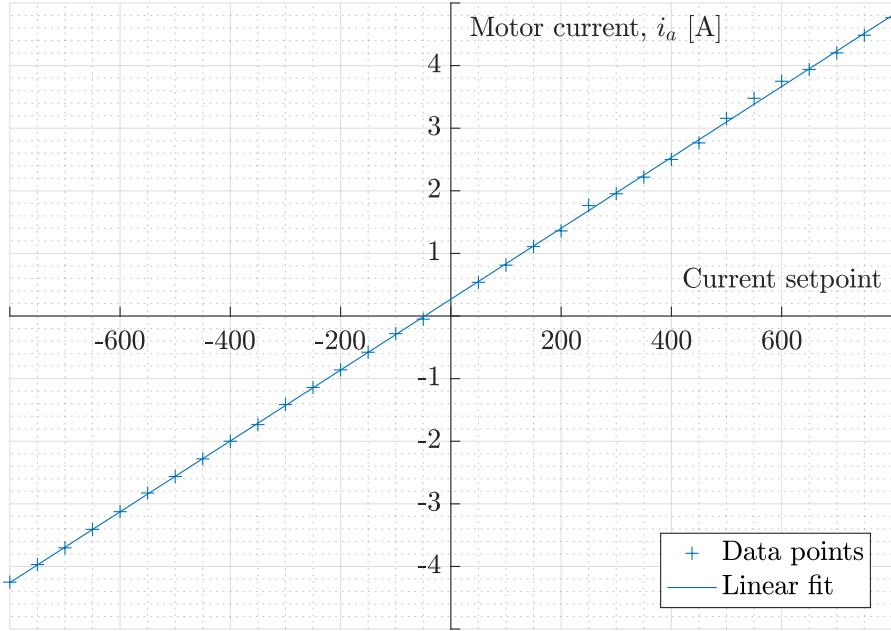


Figure A.3: Test results

The actual setpoint output in the Arduino is an 12-bit integer value between 0 and 4095, representing a voltage between 0 and 3.3 V, written on DAC0. However, to confine with an output voltage of ± 10 V after the voltage shield, an offset of 2000 is used to define 0 V out. This is incorporated in the *Joint.cpp* library. This means a negative setpoint drives the motor in one direction, and a positive setpoint drives it in the other direction. It should be noted, when inspecting the results, an integer value of 0 does not provide 0 current. The input corresponding to a zero current is ~ -47 , and thus the offset in the *Joint.cpp* library have been shifted to 1953.

A.2 Relation between Arduino setpoint and force

The force applied to the cart is defined by the motor torque divided by the length from the motor's rotor till the belt.

$$F = \frac{i_a \cdot K_M}{r} = i_a \frac{0.0934}{0.03} = i_a \cdot 3.1133 \quad (\text{A.1})$$

where K_M is the motor torque constant, and r is the distance from the motor's rotor to the belt driving the cart. The torque constant is found in the datasheet, and is defined as 93.4 mNm/A, and the distance r is measured to 3 cm. The current i_a is defined as the Arduino setpoint times the gradient found from a linear fitting of the data presented in **figure A.3**.

$$i_a = \text{setpoint} \cdot 0.0056605 \quad (\text{A.2})$$

Replacing this into **equation (A.1)** provides

$$F = \text{setpoint} \cdot 0.0056605 \cdot 3.1133 = \text{setpoint} \cdot 0.017623 \quad (\text{A.3})$$

Thus a constant of $1/0.017623 = 56.744$ can be used to convert a calculated force to a setpoint integer value.

Distributed Cart Coulomb Friction B

This appendix describes the tests conducted to determine the coulomb and viscous frictions of the motor and cart. The term "distributed" refers to how – in order to obtain precise estimations – the rail have been divided into four parts, such that frictions have been determined for each part of the rail.

B.1 System inspection

By first inspection of the system, it became clear that the coulomb friction differs depending on the placement of the cart. When moving the cart from the right-hand side towards the left, the cart experiences different sizes of friction becoming more rigid at the left-hand side. Furthermore, inspections also indicate that the viscous friction differs depending on the direction of the cart. Therefore a thorough cleaning of the rails was initiated, and after the cleaning they were also greased with MegaGlide grease, with the purpose of minimizing the frictions.

For the test, the rail have been divided into parts as depicted in **figure B.1**

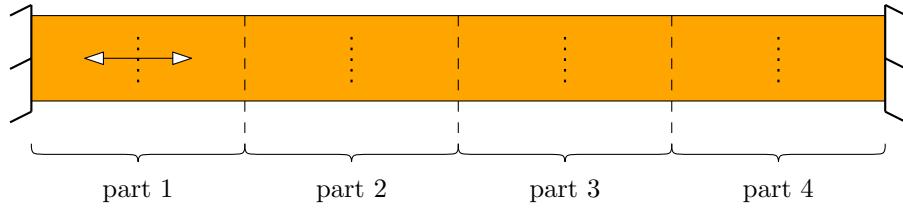


Figure B.1: The rail divided by four sections.

The cart have then been placed in the center of each part, represented by the dotted lines. In order to determine the coulomb friction, three tests have been conducted in both directions in each part using a ramp signal. In order to determine the viscous friction, a hand-tuned PI velocity controller have been developed to maintain a constant velocity. In this case, the pendulum have been placed at either end, and moved with a constant velocity across the rail.

The input signal used for testing is a force ramp scaled by the coefficient found in **appendix A**.

B.2 Test results - coulomb friction

The test results are presented as a figure for each part of the rail, and a combined table below with determined frictions. Each figure then contains six tests, three in each direction. The coulomb frictions have been estimated by selectively averaging the amount of force needed to create movement of the cart.

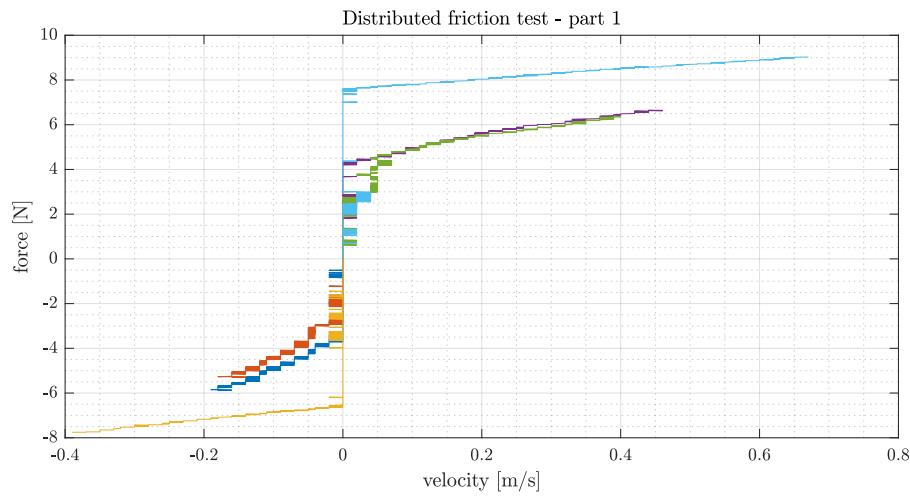


Figure B.2: Friction test results of part 1 of the rail.

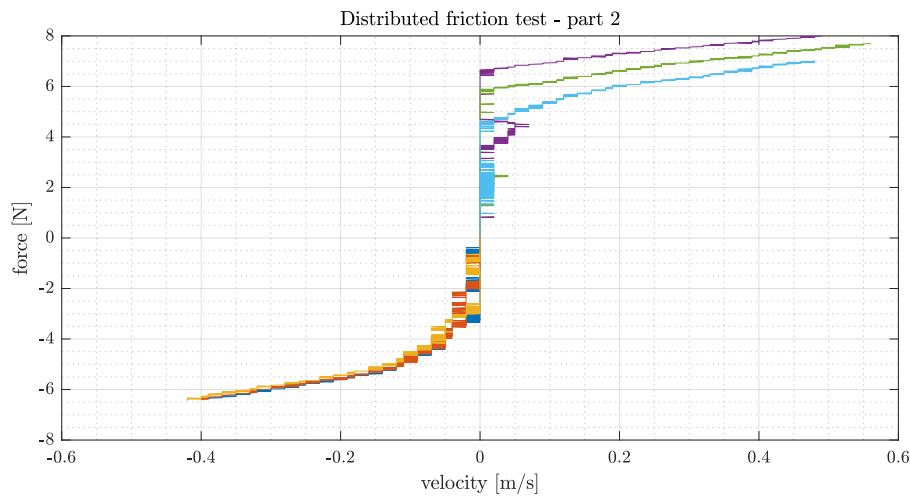


Figure B.3: Friction test results of part 2 of the rail.

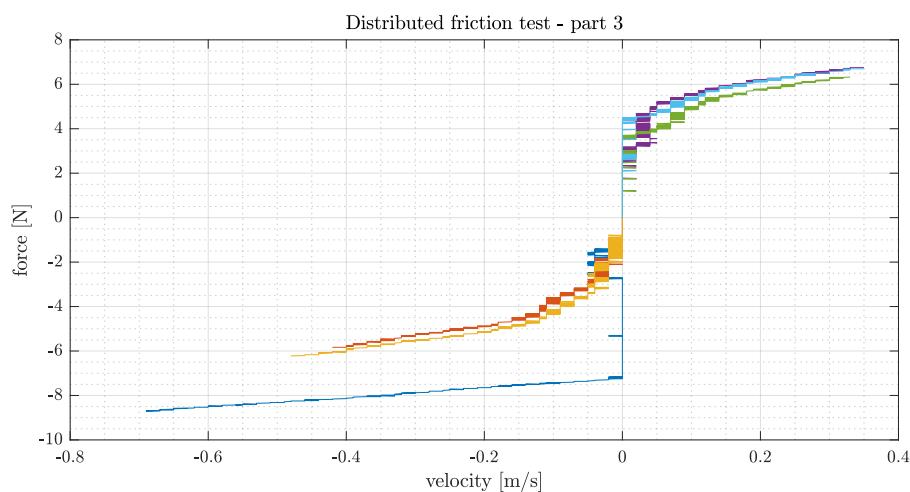


Figure B.4: Friction test results of part 3 of the rail.

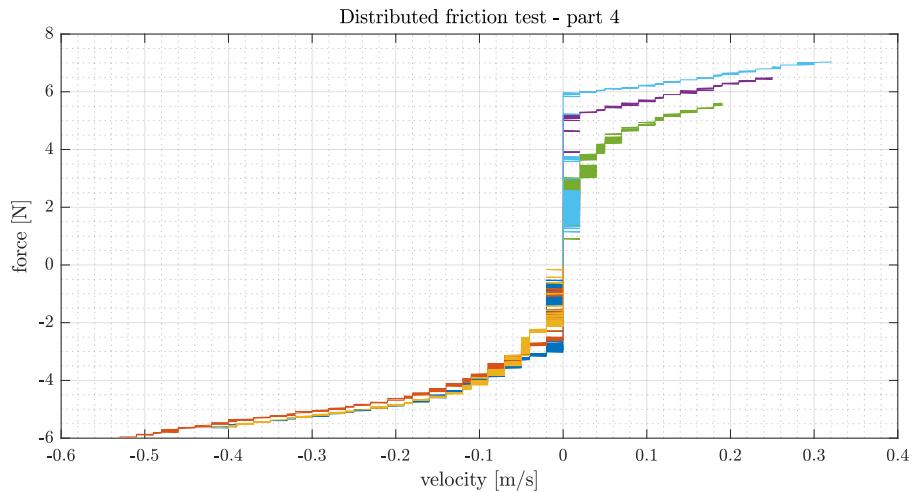


Figure B.5: Friction test results of part 4 of the rail.

Table B.1 summarizes coulomb frictions of each part and in both directions which have been determined from the presented results.

	part 1	part 2	part 3	part 4
$F_{c,r}$	2.943	2.467	1.692	1.128
$F_{c,l}$	-3.463	-2.855	-2.309	-1.956

Table B.1: Coulomb frictions in newtons [N]

The results conform well with the initial inspection of the system. Even though a thorough cleaning of rails was done, the cart still experience the prescribed frictions. The coulomb friction it is significantly higher when moving the cart the to right than moving the cart left, and is it generally lowest in the right hand side of the rail – part 3 and 4 – as mentioned.

Furthermore, as it can be seen in every figure, some tests have a larger coulomb friction than others. In **figure B.2** the yellow, and light blue lines have larger coulomb frictions and a very linear viscous friction. However, they do have velocity spikes in close vicinity of the other tests, but then return to zero velocity before enough force have been applied to overcome the coulomb friction. In **figure B.3** the purple line also have a velocity spike near the light blue line, before again returning to zero velocity. The same can also be seen for the blue line in **figure B.4**, and light blue line in **figure B.5**. These discontinuities have been determined to be caused by the ball bearing driving the cart. This have been concluded by disassembling the belt from the cart to inspect the parts individually. The motor actuating the belt wells runs smoothly, and the ball bearings of the cart runs fairly smooth, but some discontinuities can be identified. The transitions between the gears of belt contains a surprising amount of friction, which must amplify the friction from the ball bearings, creating the extra coulomb friction seen in the figures. The friction between the belt and the wheels can be experienced by mounting the belt on the wheels, and manually moving it back and forth.

After consultation with the smith Jesper, and electronic technologist Simon, it was decided to order a new belt with teflon treated surface in hopes of minimizing the friction between gear transitions.

B.3 Attempt to model Coulomb friction position dependency

This section attempts to determine if there exists a position dependency of the Coulomb friction.

For the Coulomb friction coefficient, c_x , a ramping current input has been used to increase the torque produced by the motor and thus the force applied to the cart. The Coulomb friction is then the current required to set the cart in motion. This has been repeated at several places along the rail and in both directions in order to check for consistency, as depicted in **figure B.6**.

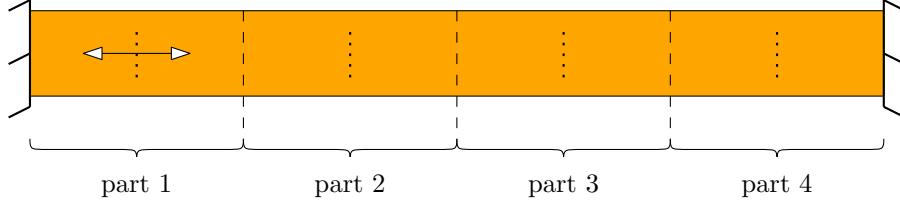


Figure B.6: Segments which the rail has been divided into.

The results of the experiment is shown in **figure B.7**.

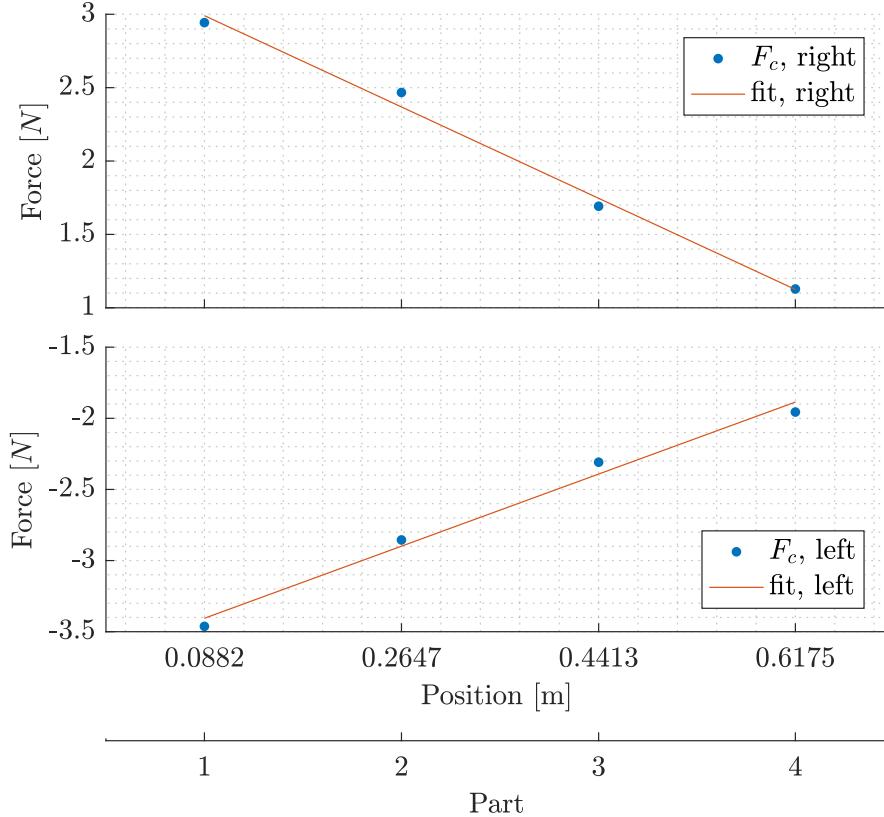


Figure B.7: Coulomb frictions determined for different locations of the rail and in both directions.

The data is fitted to the following linear functions:

$$c_{x,l}(x) = 2.8709x - 3.6588 \quad (\text{B.1})$$

$$c_{x,r}(x) = -3.5249x + 3.3015 \quad (\text{B.2})$$

Cart Viscous Friction C

This appendix describes the tests conducted in order to determine the viscous friction of the cart. A hand-tuned PI velocity controller have been developed, and used in two different test setups. The first test was conducted using MATLAB for logging purposes, and one using a Tektronix TDS2004B Oscilloscope.

A velocity controller have been chosen since a constant constant velocity equals zero acceleration, and thus the friction of the system is isolated as follows

$$F_{in} - F_{fric} = m\ddot{x} \quad (\text{C.1})$$

$$F_{in} = F_{fric} \quad (\text{C.2})$$

Thus the input force needed to move the cart is expressed by the combined friction force which is the coulomb and viscous friction. Since the coulomb friction is known at this point, the viscous friction can be isolated by doing tests at different velocities.

In order to make sure the coulomb friction is overcome, each test have been conducted at 0.4 m/s, 0.6 m/s, 0.8 m/s, and 1 m/s in both directions. Thus the cart have been placed in each end of the rail, and driven across its entire range.

The control action of the PI controller is calculated as follows

```
setpoint = -(V_ref - velSled) * V_Kp - V_acc_err * V_Ki * SAMPLINGTIME
```

where V_{ref} is the velocity reference, $velSled$ is the calculated velocity of the sled, V_{Kp} is the proportional gain, and V_{Ki} is the integral gain. The accumulated error, V_{acc_err} is calculated as

```
V_acc_err += V_ref - velSled;
```

Since the Arduino, using the library Joint.cpp, measures a position in mm, and converts this to a speed in m/s, the controller contains large gains where $V_{Kp} = 3000$, and $V_{Ki} = 4500$.

C.1 Test logged with MATLAB

When using MATLAB to log the data from the Arduino, the test setup is no different than the setup described in [chapter 2](#). MATLAB sends a start command to the Arduino, and it responds with measured data on the serial line. Using the coefficients described in [appendix A](#), the input signal have been directly converted to the motor current, i_a .

To present the results several figures have been created, and comments will follow below.

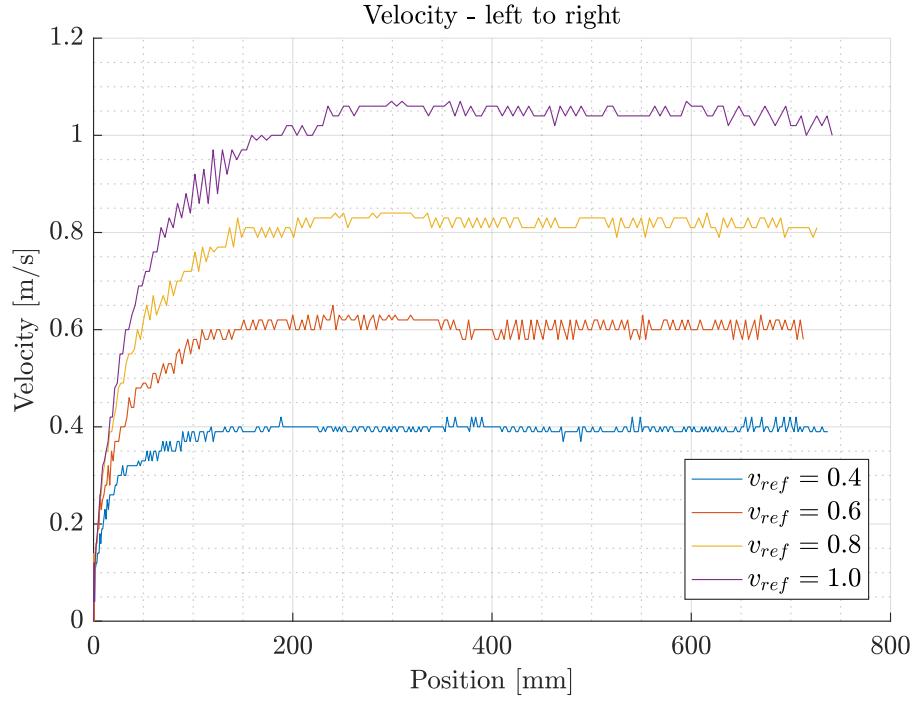


Figure C.1: Velocity references when the cart moves from the left-hand side to the right-hand side, or from part 1 to part 4.

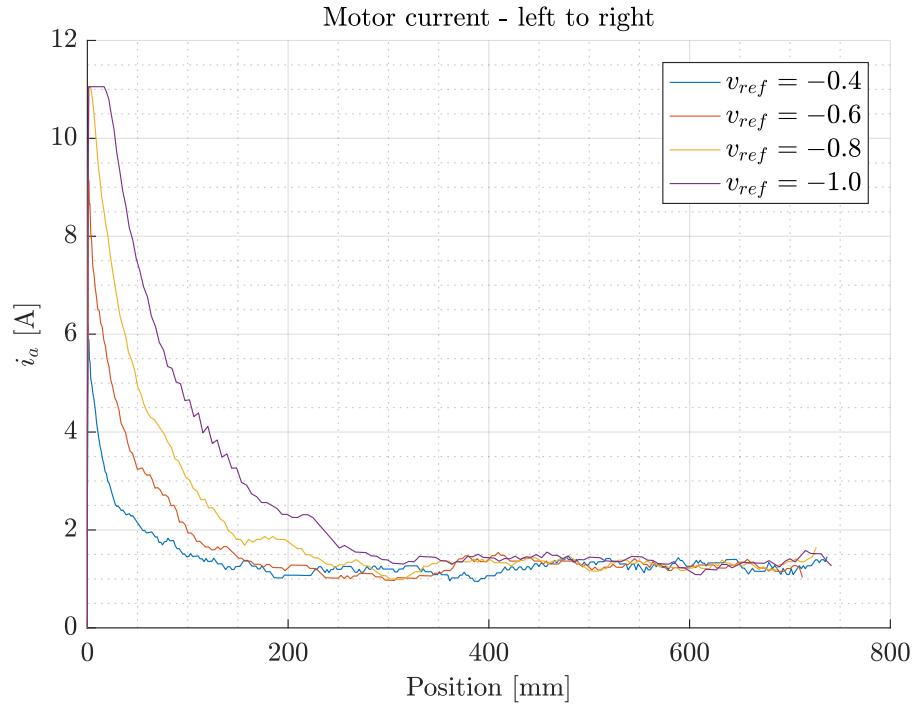


Figure C.2: Motor current used to achieve the velocities shown in figure C.1

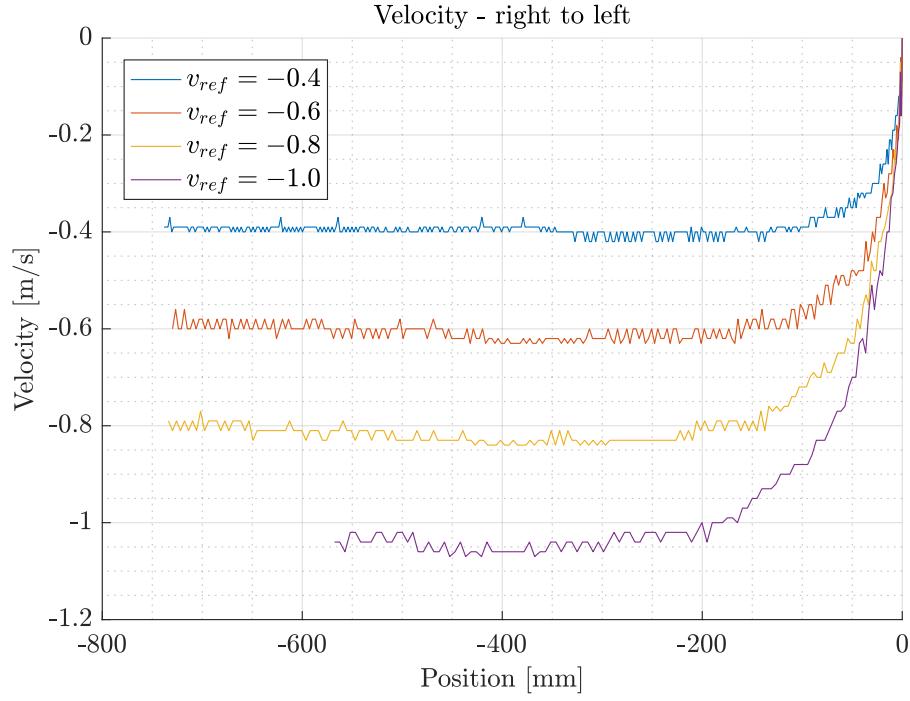


Figure C.3: Velocity references when the car moves from the right-hand side to the left-hand side, or from part 4 to part 1.

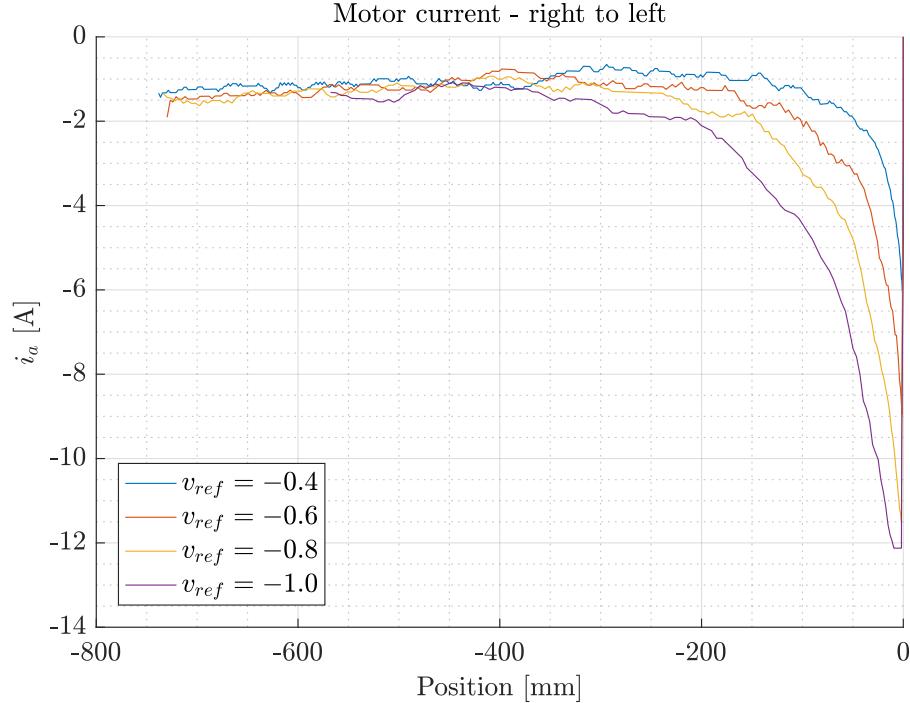


Figure C.4: Motor current used to achieve the velocities shown in figure C.3

It can be seen in **figure C.1** and **figure C.3** that the controller needs some time to obtain the reference velocity, thus the viscous friction cannot be determined in the first parts of the tests. In both figures the velocity is constant after the sled have moved roughly 400 mm. At a reference velocity of ± 1 m/s, the calculated current saturates at 11 A and -12 A respectively. This means the servo amplifier

provides maximum outputs, and tests with faster velocity references have not been conducted.

The motor current in **figure C.2** for each velocity reference becomes identical after the sled have moved 400 mm, thus giving an indication of that the viscous friction is very small, and maybe not significant. Ideally the current for each reference should settle at larger values for faster references. Inspecting the figure closely, from 450 mm to 700 mm, it can be seen that the average tendency of the current is declining, however at a small rate. This confines well with the experienced Coulomb friction described in **appendix B**, where the Coulomb friction decreases as the cart moves from part 1 to part 4.

The current in **figure C.4** for the cart moving from part 4 towards part 1 also conforms well with the described Coulomb friction of the cart. Even though the current for all references are nearly identical, they all share the same tendency. As described in **appendix B**, the Coulomb friction rises when moving from part 4 to part 1. This is verified by this test, since the velocity in **figure C.3** is held constant – but the current increases as the cart moves on the rail.

Summarizing this test, the viscous friction could not be determined using MATLAB for logging data. The difference between the currents at constant velocities is too small. This implies that the viscous friction is very small, and could be neglected.

C.2 Test logged with oscilloscope

Since the servo amplifier saturates the output in the previous tests, it would be interesting to measure the actual output, and compare it to the calculated, thus verifying the approximation done in **appendix A**. With this test, it is also hoped to achieve a higher resolution of the current and thereby being able to identify the viscous friction. Using an oscilloscope to measure the current, the test setup have changed a bit, as depicted in **figure C.5**

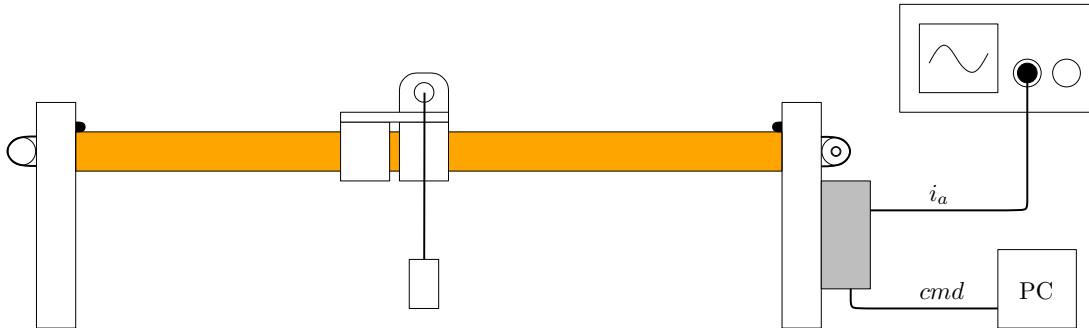


Figure C.5: Test setup using an oscilloscope to measure to motor current directly from the servo amplifier

The oscilloscope is connected directly to the servo amplifier, which outputs the motor current and needs to be scaled with a gradient of approximately 0.4 V/A. Since MATLAB is not used, the Arduino program is started by writing directly on the serial line using the Arduino IDE's Serial Monitor. The tests have been conducted using the same PI controller as in **appendix C.1**, and it is therefore known that the velocity becomes constant after the sled have moved roughly half the length of the rail – thus the current measurements stands alone.

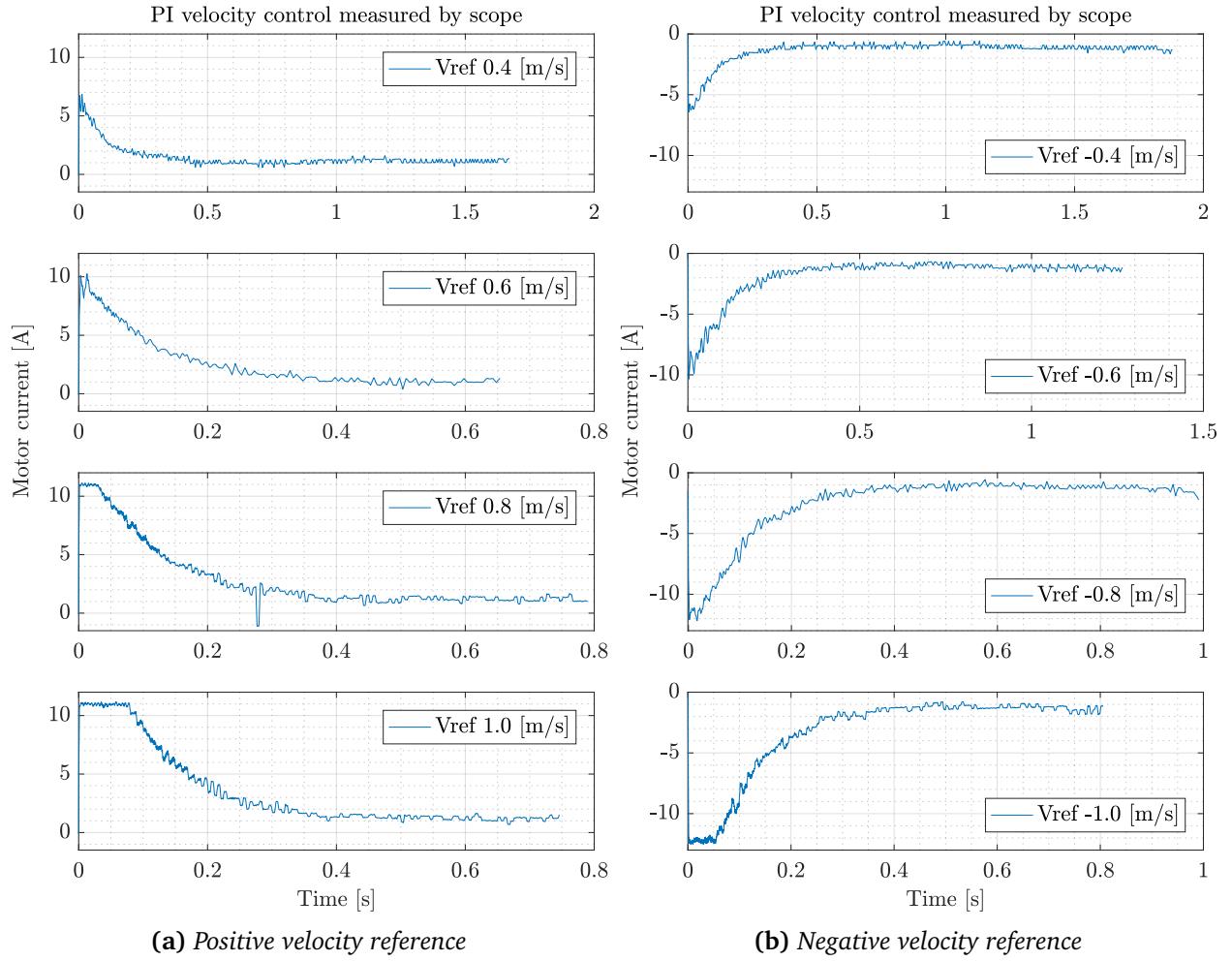


Figure C.6

Inspecting **figure C.6a** closely, when the current is constant, only the tests with a reference of 0.6 m/s and 1 m/s includes a decreasing rate of the current, implying correspondence with the estimated Coulomb friction. However, in all four tests, the current settles in too close proximity of each other – indicating again, that the viscous friction is very small.

In **figure C.6b** the current in all tests settles too close to each other, preventing an identification of the viscous friction. Inspecting the individual subplots, it can be seen, that the current increases over time confining with the identified Coulomb friction, which increases when the cart moves from part 4 to part 1.

As it can be seen in both figures, the current saturates at approximately 11 A and -12 A, which corresponds nicely to the calculated maximum and minimum current depicted in **figure C.1** and **figure C.3** respectively. Thus concluding that the conversion from an integer value to a current, described in **appendix A**, is linear in the entire spectrum. The current resolution have not improved significantly enough, concluding that the calculated motor current will suffice for further use in this project.

Test of Teflon-coated Belt D

When the new belt was installed, some viscous friction tests were conducted, to see whether the new belt had minimized the friction or not. **Figure D.1** and **figure D.2** displays the results of tests done with a velocity reference of -0.5 m/s, and 0.5 m/s, hence a test in either direction.

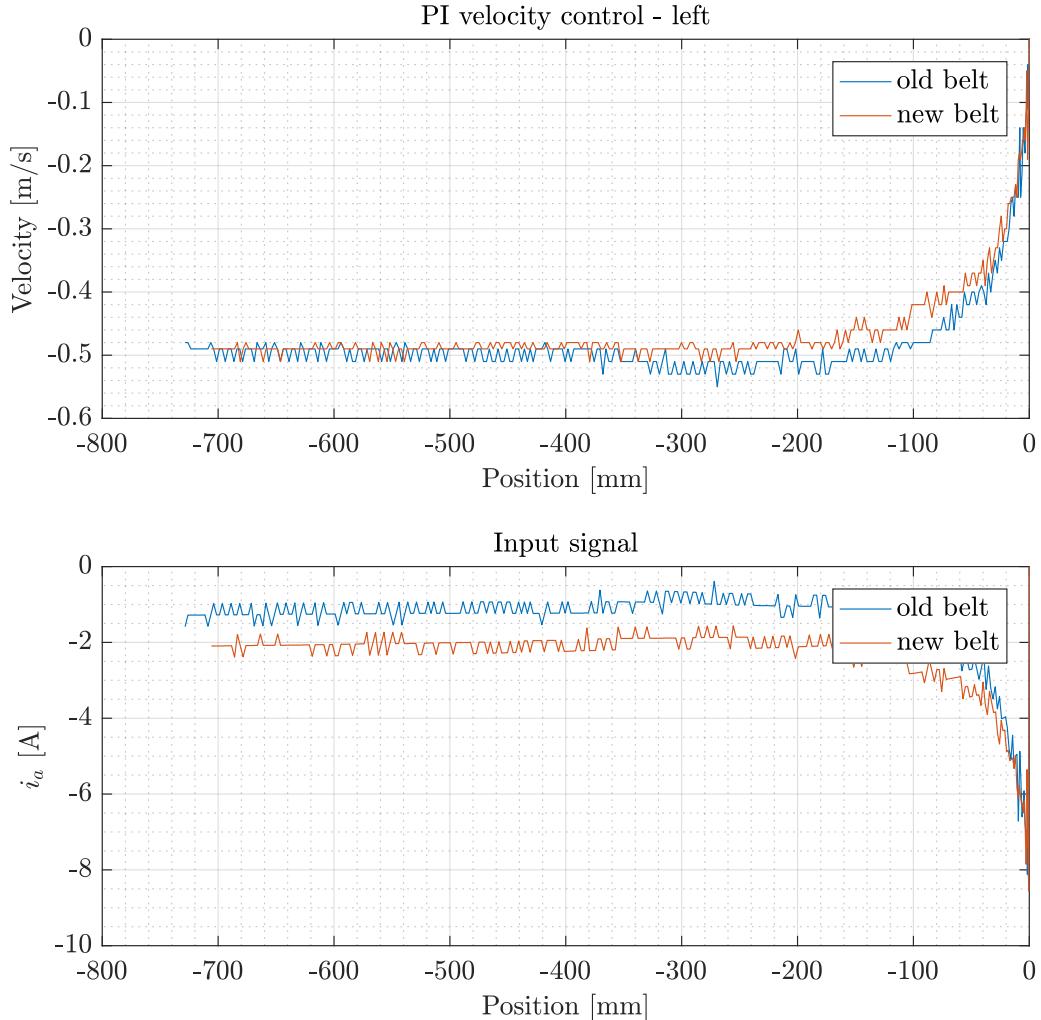


Figure D.1: PI velocity test. Notice how the red graph(new belt) have a slightly slower rise time than the blue(old belt) on the top figure. Bottom figure displays motor current, and the red curve is stable around -2 [A], where the blue is stable around -1 [A].

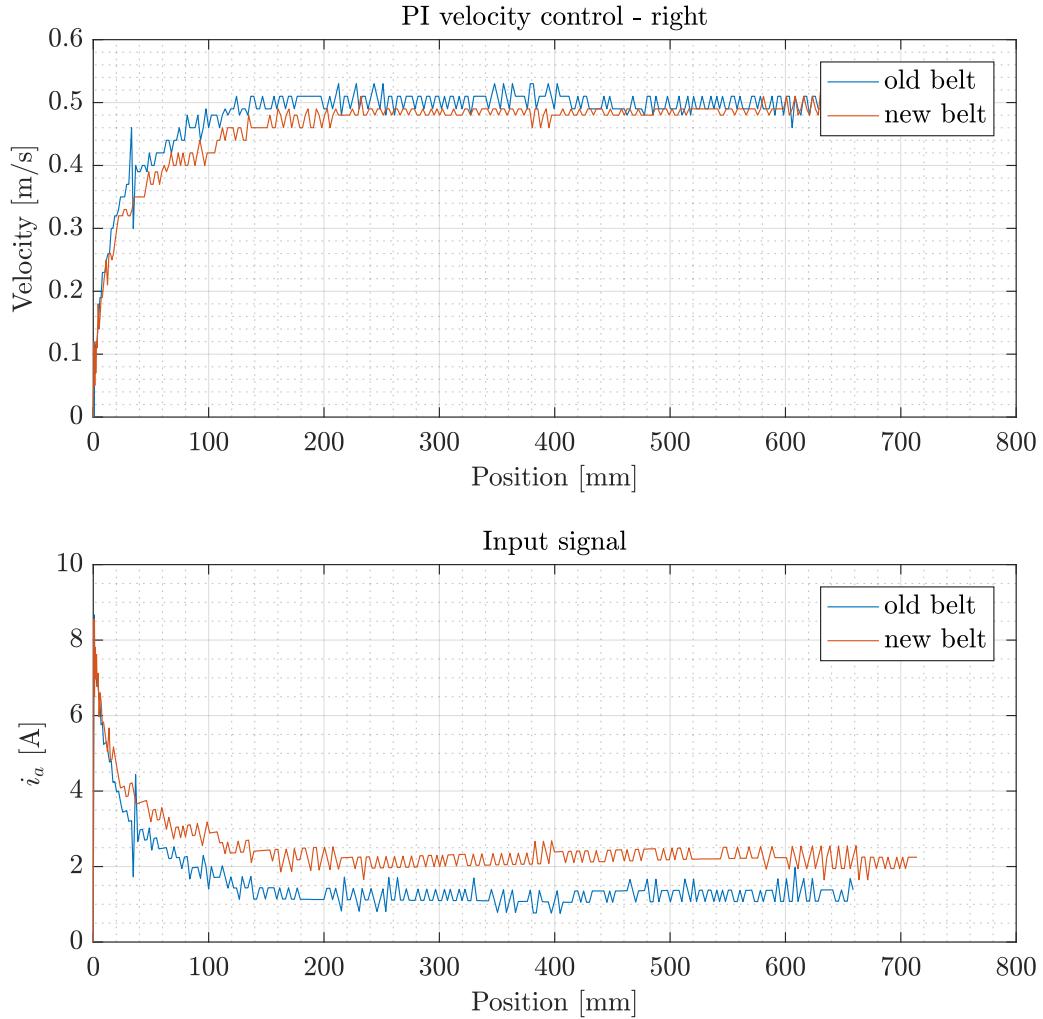


Figure D.2: Rise times on the top figure are almost identical, but from the bottom figure, it can be seen that the new belt introduces a large friction due to an increase in motor current.

As it can be seen in both figures, the new belt causes the controller to deliver more current to the motor, hence more friction is introduced using the new belt. After some considerations, and more tests, all with the same outcome, it was decided to return to the belt used in the first place.

Pendulum Friction Estimation E

This appendix describes the tests conducted in order to find the necessary coefficients for the pendulum friction model.

E.1 Coulomb coefficient

In order to determine the coulomb coefficient, c_θ , the pendulum alone is considered, as shown in figure E.1.

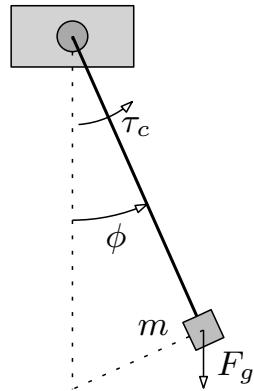


Figure E.1: Pendulum with parameters used for coulomb friction estimation.

Using a small end-mass, the pendulum is manually lifted very carefully to either the right or left side of the bottom equilibrium point, until largest possible angle at which the pendulum can hold still.

At this point, the the gravitational pull is not large enough to overcome the Coulomb friction, thus the position is maintained.

Then, the angle is measured, and the procedure repeated several times. Then, for the known mass m_p , the torque produced by the Coulomb friction is:

$$\tau_c = -\tau_g = -F_g l \sin(\phi) = m_p g l \sin(\phi) \quad (\text{E.1})$$

Using parameters $m_p = 0.026 \text{ kg}$, $g = 9.81 \text{ m/s}^2$ and $l = l_{\text{rod}} + 0.5 \cdot l_{m26g} = 0.2875 \text{ m}$, the results are:

ϕ	-0.0408	-0.0377	-0.0440	0.0377	0.0408	0.0408
τ_c	-0.0030	-0.0028	-0.0032	0.0028	0.0030	0.0030

Taking the mean of the absolute values of τ_c yields the Coulomb coefficient $c_p = 0.0030$ for the pendulum.

E.2 Viscous coefficient

In order to determine the viscous friction coefficient, v_p , the pendulum is considered like in **figure E.1**, but it is now raised manually to the vicinity of the top equilibrium point at $\phi = \pi$, and then released while the cart is kept at a fixed position.

The pendulum will start oscillating and is expected to behave like a underdamped harmonic oscillator, for which the position will vary with a exponentially declining cosine curve, i.e.:

$$\theta(t) = \theta_0 \exp^{-\gamma t} \cos(\omega t - \alpha), \quad \theta_0 = \theta(t=0) \quad (\text{E.2})$$

Considering only the amplitude of the oscillation, **equation (E.2)** becomes:

$$|\theta| = \theta_0 \exp^{-\gamma t} \quad (\text{E.3})$$

$$\gamma = \frac{v_p}{2m_p} \quad (\text{E.4})$$

Using MATLAB-function `findpeaks` to find the peaks of the oscillation, and then fitting them to **equation (E.3)**, yields the results shown in **figure E.2** for three different masses.

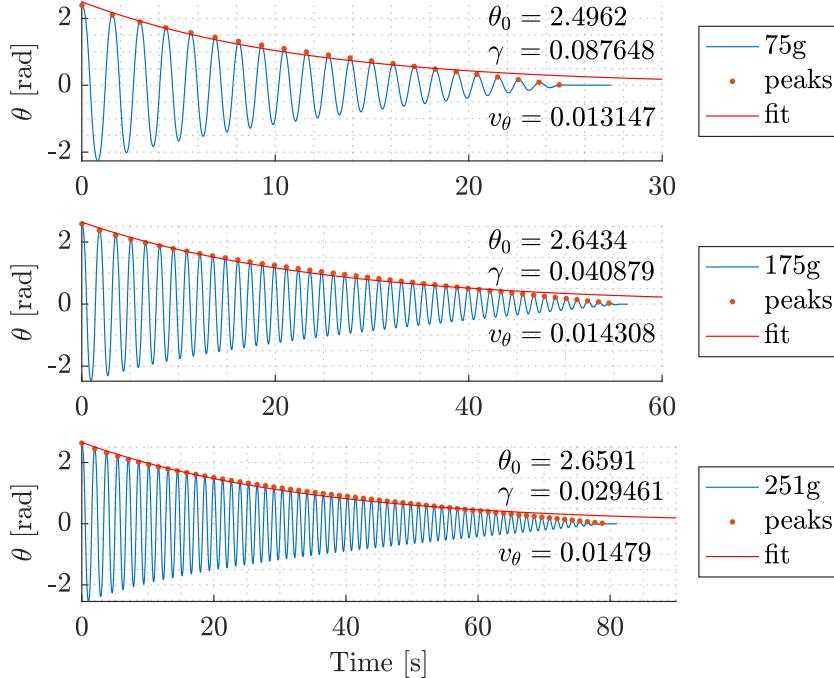


Figure E.2: Measurements for free, underdamped, harmonic oscillation with peaks, fit and fit results θ_0 , γ , v_p .

Figure E.2 shows that v_p varies slightly with the three test scenarios. However, as $m_p = 251g$ is the scenario mainly considered in this thesis, the found value for v_p for that case will be used.

Aerodynamic Drag F

In order to verify that effects from aerodynamic drag on the pendulum are negligible, an approximate torque produced by drag at various angular velocities will be presented in this appendix.

For a linear motion with v through air with density ρ , the force exerted on the object in motion is given by:

$$F_d = \frac{1}{2}\rho C_d S v^2 \quad (\text{F.1})$$

where C_d is the drag coefficient determined by the shape of the object and S is the area projected towards the direction of motion.

Using the relation between linear and angular velocity, $v = \omega l$ and considering both the pendulum rod and tip-mass as cylinders ($S = w \cdot l$), **equation (F.1)** becomes:

$$F_d = \frac{1}{2}\rho C_d w(\omega l)^2 \quad (\text{F.2})$$

$$= \frac{1}{2}\rho C_d w\omega^2 l^3 \quad (\text{F.3})$$

The differential of the aerodynamic drag is then:

$$\frac{dF_d}{dl} = \frac{1}{2}\rho C_d w\omega^2 \frac{dl^3}{dl} \quad (\text{F.4})$$

$$= \frac{1}{2}\rho C_d w\omega^2 3l^2 \quad (\text{F.5})$$

and thus

$$dF_d = \frac{1}{2}\rho C_d w\omega^2 3l^2 dl \quad (\text{F.6})$$

The torque exerted by the aerodynamic drag in the pendulum joint is then found by:

$$\tau_d = \int_{l_{min}}^{l_{max}} l \cdot dF_d \quad (\text{F.7})$$

$$= \int_{l_{min}}^{l_{max}} \frac{3}{2}\rho C_d w\omega^2 l^3 dl \quad (\text{F.8})$$

$$= \frac{3}{2}\rho C_d w\omega^2 \int_{l_{min}}^{l_{max}} l^3 dl \quad (\text{F.9})$$

$$= \frac{3}{2}\rho C_d w\omega^2 \left[\frac{1}{4}l^4 \right]_{l=l_{min}}^{l_{max}} \quad (\text{F.10})$$

$$(F.11)$$

For the rod with integral limits 0 and l_{rod} , the torque is evaluated to:

$$\tau_{d,rod} = \frac{3}{8}\rho C_d w \omega^2 l_{rod}^4 \quad (\text{F.12})$$

and for the end-mass with limits l_{rod} and $l_{rod} + l_{mass}$

$$\tau_{d,mass} = \frac{3}{8}\rho C_d w \omega^2 \left((l_{rod} + l_{mass})^4 - l_{rod}^4 \right) \quad (\text{F.13})$$

Finally, the total torque produced by aerodynamic drag is:

$$\tau_{d,total} = \tau_{d,rod} + \tau_{d,mass} \quad (\text{F.14})$$

An evaluation of **equations (F.12)** to **(F.14)**, with $C_d = 0.96$, $\rho = 1.2041 \frac{\text{kg}}{\text{m}^3}$ and the remaining parameters as given in **table 2.1**, is shown in **figure F.1**.

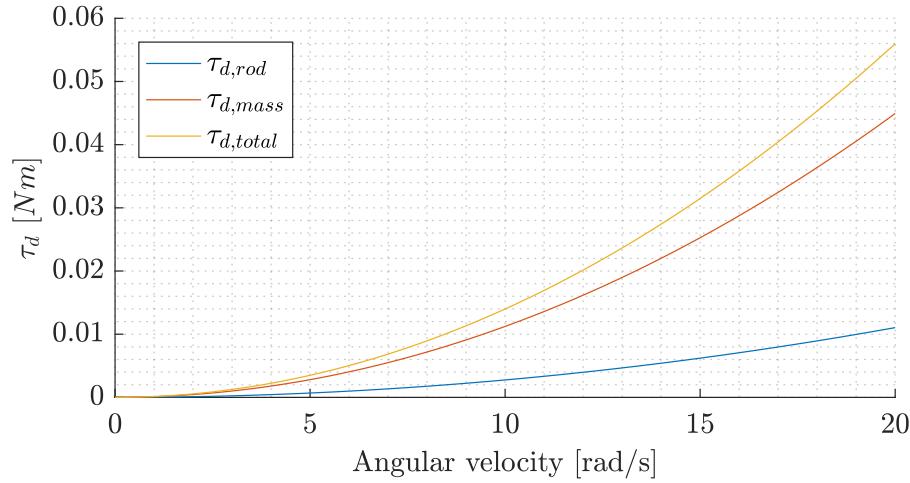


Figure F.1: Torque produced by aerodynamic drag on the pendulum.

For comparison, the torque produced by gravity at $\theta = \frac{\pi}{2}$ is:

$$\tau_g = (l_{rod} + 0.5l_{mass})m_p g \quad (\text{F.15})$$

$$= 0.8236 \text{ Nm} \quad (\text{F.16})$$

[23]

Determination of Measurement Noise G

This appendix serves as documentation of the measurement noise. The theory behind this test is, that when maintaining a constant output, then, if any measurement noise exists, it would be visible when subtracting the mean of the output from itself.

A simple test was conducted where the cart was manually moved into the middle of the rail, and then allowed to settle. This resulted in the following results

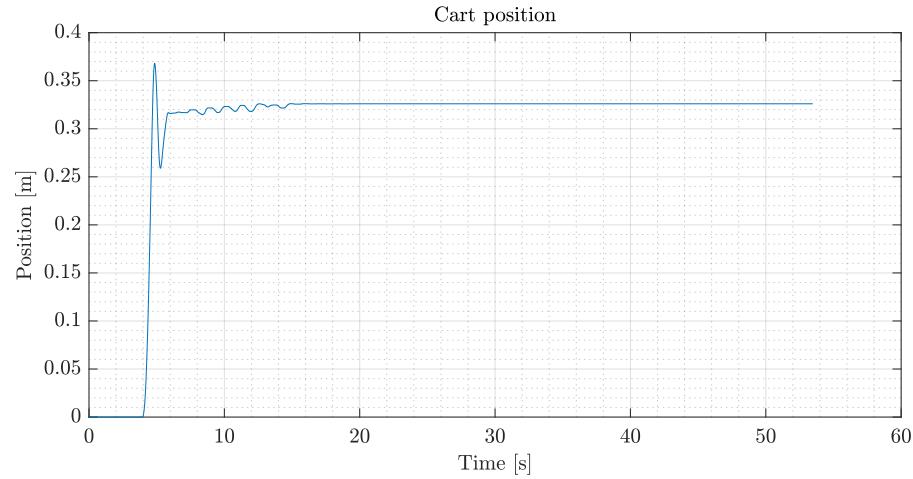


Figure G.1: Cart position

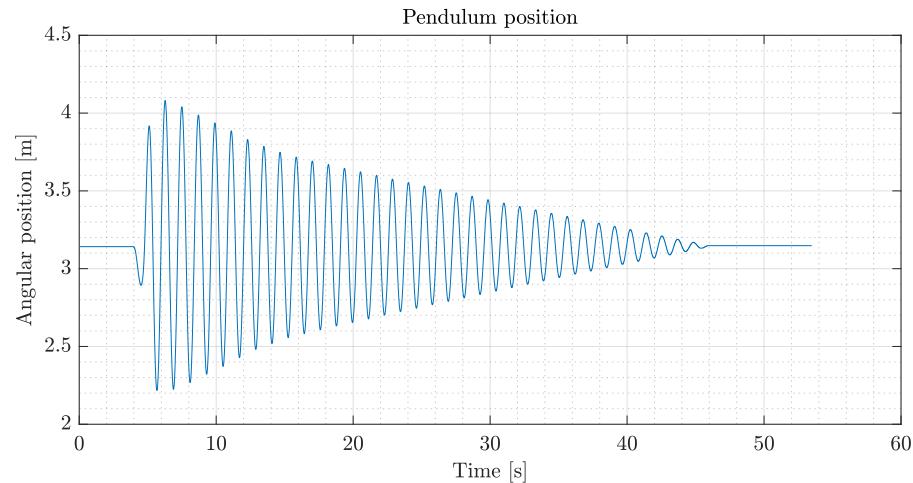


Figure G.2: Pendulum position

As it can be seen on both figures, when the system has settled, there is no deviation on the output. Subtracting the mean is therefore omitted. Since there is no deviation on the output at the end of both measurements it can be concluded that the measurement noise is smaller than the quantization, and in effect the measurement noise matrix R_k can be set to zero.

Initialisation Error Test H

The purpose of this appendix is to document the outcome when the pendulum is not carefully initialised. Upon initialisation, before activating the control, the cart must be taken to the leftmost position on the rail and the pendulum must be hanging exactly downwards.

Due to the presence of Coulomb friction in the pendulum joint, although small, the pendulum may not be hanging straight down when initialised (i.e. θ is reset to π). This will result in a response as shown in **figure H.1**.

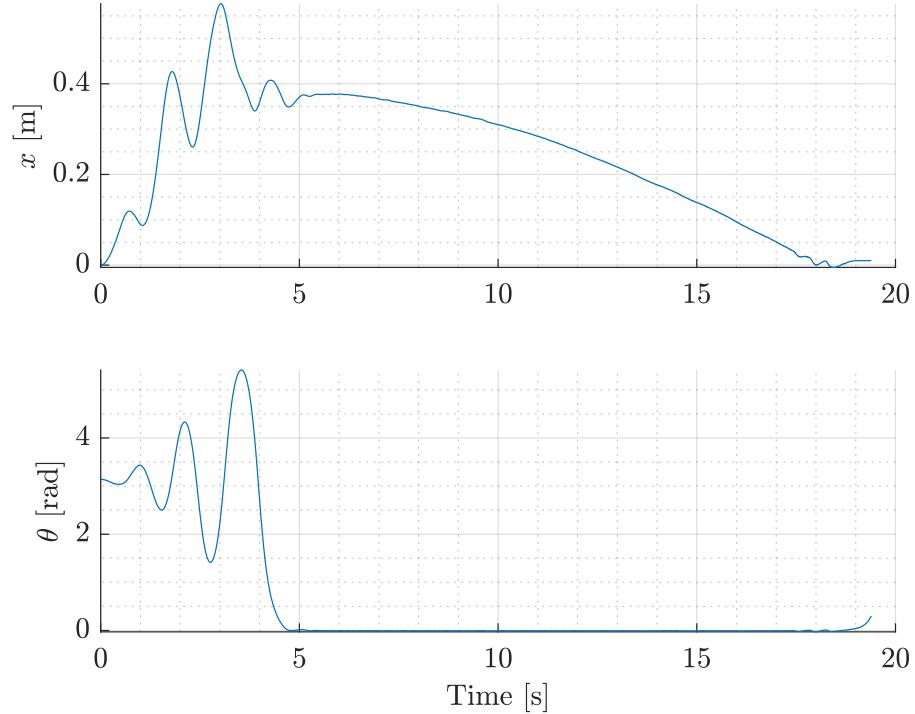


Figure H.1: Outcome of swing-up and stabilisation when pendulum is not properly initialised.

It is seen that the swing-up and catch is executed successfully, but some error persists, causing the cart to drift until it hits the end of the rail.

Carefully flicking the pendulum or pushing the table on which it is mounted tends to help prevent this.