The Feedback Control of a Robotic Gymnast

by

Henry Kotzé

Thesis presented in partial fulfilment of the requirements for the degree of Master of Engineering (Mechanical) in the Faculty of Engineering at Stellenbosch University

Supervisor: Dr. J.A.A Engelbrecht

October 2018

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: 2018/10/27

Abstract

The Feedback Control of a Robotic Gymnast

H. Kotzé

Department of Mechanical and Mechatronic Engineering,
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.

Thesis: BEng (Mechatronics)

October 2018

Vibrating a tillage tool is an effective way of reducing the draft force required to pull it through the soil. The degree of draft force reduction is dependent on the combination of operating parameters and soil conditions. It is thus necessary to optimize the vibratory implement for different conditions.

Numerical modelling is more flexible than experimental testing and analytical models, and less costly than experimental testing. The Discrete Element Method (DEM) was specifically developed for granular materials such as soils and can be used to model a vibrating tillage tool for its design and optimization. The goal was thus to evaluate the ability of DEM to model a vibratory subsoiler and to investigate the cause of the draft force reduction.

The DEM model was evaluated against data ...

Uittreksel

Die Terugvoer Beheer van ’n Robotiese Gimnas

(“The Feedback Control of a Robotic Gymnast”)

H. Kotzé

Departement Meganiese en Megatroniese Ingenieurswese,
Universiteit van Stellenbosch,
Privaatsak X1, Matieland 7602, Suid Afrika.

Tesis: BIng (Megatronika)
Oktober 2018

Om ‘n tand implement te vibreer is ‘n effektiewe manier om die trekkrag, wat benodig word om dit deur die grond te trek, te verminder. Die graad van krag vermindering is afhanklik van die kombinasie van werks parameters en die grond toestand. Dus is dit nodig om die vibrerende implement te optimeer vir verskillende omstandighede.

Numeriese modulering is meer buigsaam en goedkoper as eksperimentele opstellings en analitiese modelle. Die Diskrete Element Metode (DEM) was spesifiek vir korrelrige materiaal, soos grond, ontwikkel en kan gebruik word vir die modellering van ‘n vibrerende implement vir die ontwerp en optimering daarvan. Die doel was dus om die vermoë van DEM om ’n vibrerende skeurploeg the modelleer, te evalueer, en om die oorsaak van die krag vermindering te ondersoek.

Die DEM model was geïvalueer teen data ...

Acknowledgements

I would like to express my sincere gratitude to the following people and organisations ...

Dedications

Hierdie tesis word opgedra aan ...

Contents

List of Figures

List of Tables

Nomenclature

Constants

g =	9.81 m/s^2

Variables

Re_{D}	Reynolds number (diameter)	[ ]
x	Coordinate	[m]
$\ddot{x}$	Acceleration	[m/s^2]
θ	Rotation angle	[rad]
τ	Moment	[N·m]

Vectors and Tensors

$\overrightarrow{\boldsymbol{v}}$	Physical vector, see equation ...

Subscripts

a	Adiabatic
a	Coordinate

Chapter 1

Introduction

1.1 Problem Statement

A feedback control system for a swinging robotic gymnast must be designed, implemented and verified. Feedback control loops must be designed that use the "legs" of the gymnast to swing the "body" of the gymnast from a "hanging" position to a "handstand" position and then balance the gymnast on top of the horizontal bar. A mathematical model for the dynamics of the swinging robotic gymnast system must be derived or sourced from literature. The dynamics must be analysed to propose an appropriate feedback control architecture that actuates the "legs" of the gymnast using feedback from sensors that measure the swinging motion of the gymnast on a horizontal bar. A practical demonstrator must be constructed and the correct operation must be demonstrated.

The feedback control of a robotic gymnast entails to model the behavior the of a gymnast hanging from a horizontal bar and tries to swing himself upwards using his legs until he balance himself in the inverted position. The problem is modelled as a double pendulum: where one end is connected to a fixed hinge, the 2 pendulums are connected with a hinge between eachother and there is some torque actuating the lower pendulum.

1.2 Literature Study

The fundamental concepts that provides the foundation for the concepts discussed in the report will be summarise here. It acts as a refresher for those familiar to control theory.

The ordinary differential equations (ODE's) describing a system can be arranged as a set of linear differential equations. Describing a system in such a way is known as the State Space design approach, where the solution is the trajectory of the chosen state variables.Gene F. Franklin (2015)

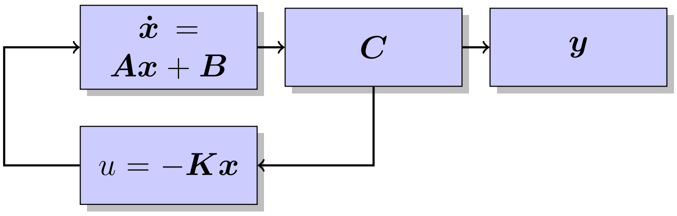
$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}$
$\boldsymbol{C}$
$\boldsymbol{y}$
$u = -\boldsymbol{K}\boldsymbol{x}$

Figure 1.1: State Space Representation with Feedback Gain

These ODE's are required to be written as vectors in the state-variable form seen in equation (1.1) and (1.2)

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}u \tag{1.1}$$

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + Du \tag{1.2}$$

The nth-column vector $\boldsymbol{x}$ is called the state of the system for a nth-order system. The **A** matrix is the system matrix, containing $n{\times}n$ elements and the input matrix is the $n{\times}1$ **B** matrix. **C** is a $1{\times}n$ row matrix called the output matrix and the scalar D is known as the direct transmission term Gene F. Franklin (2015).

A system parameter of great interest to control engineers are the poles of the system. It provides the characteristic response of the system starting at a initial condition with no forcing function. These poles,**s**, are the natural frequencies of the system and the state space representation allow these poles to be easy identified. The poles are the solution to the eigenvalue problem of the **A** matrix shown in equation (1.3) Gene F. Franklin (2015).

$$\det(s\boldsymbol{I} - \boldsymbol{A}) = 0 \tag{1.3}$$

The poles of the system can be assigned new positions to satisfy dynamic response specification by introducing feedback. The feedback is a linear combination of the state variables $\boldsymbol{x}$ resulting in the input of the system u to be transformed as seen in equation (1.4) and represented in Figure 1.1. Substituting equation (1.4) into (1.1) the characteristic equation is now describe as (1.5). The corresponding characteristic equation is:

$$\alpha_s = (s - s_1)(s - s_2) \ldots (s - s_n)$$

This shows by selecting the correct gain matrix **K** the poles of the system can be moved to a desired position.

$$u = -\boldsymbol{K}\boldsymbol{x} \tag{1.4}$$

$$\det[s\boldsymbol{I} - (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})] = 0 \tag{1.5}$$

The classical approach to controlling a system is by implementing a controller which reacts on the error of the desired state and the current state. These controllers are more commonly known as PID-controllers where the controller equation is shown in (1.6).

$$u(t) = K[e(t) + K_I \int_0^t e(\tau)d\tau + K_D \frac{de(t)}{dt}] \tag{1.6}$$

Each term represent an effect it has on the system response when the PID-controller is implemented shown in Figure **??**. If the system or plant is assumed to be a second-order differential equation represented by:

$$\dddot{q} + (2\zeta\omega_n + K_D)\ddot{q} + (\omega_n^2 + K_P)\dot{q} + K_I = 0 \tag{1.7}$$

From equation (7) it is visible that by tuning the PID constants the response of the system can controlled.

1.3 System Overview

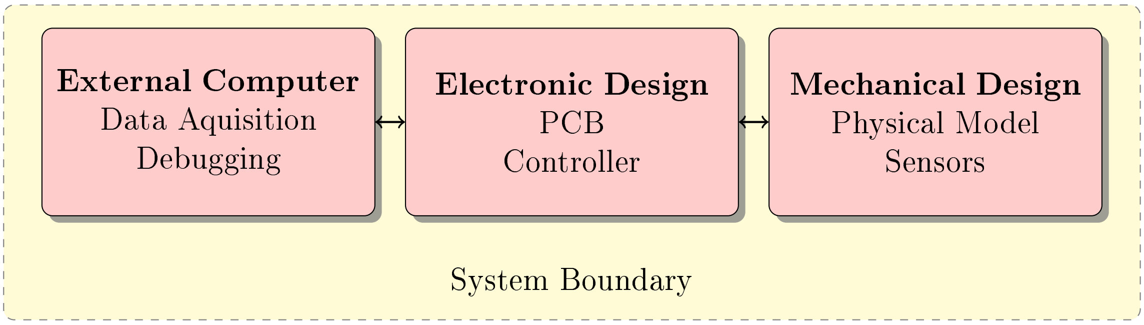

Figure 1.2: System Overview of the Feedback Control of Robotic Gymnast

Figure 1.2 provides an overview of the various subsystems the project will contain. The project is subdivided into these subsystems being developed separately with little interaction between the each other. An brief overview on each subsystem will be presented here.

The external computer communicates with the electronic design sending instruction to start the system and for debugging purposes. The external computer will receive data from the electronic design and allows the verification of system parameters.

The electronic design acts as the middle-man between the mechanical design and the external computer. It provides instructions to the mechanical design components based on the controller while sending data to the external computer. The electronic design contains the Printed Circuit Board (PCB) that conditions all the signals for processing.

The mechanical design is responsible for creating a physical model that represents the mathematical model describing the system. The correct sensors must be selected to measure the state variables and providing interfaces for the electronic design.

1.4 Project Execution

The execution of the project occurs in a sequence of steps. First the mathematical model of the system is derived by using the appropriate approaches. The derived mathematical model is then implemented on a simulation program where the dynamics of the system can be verified and inspected.

Following the successful implementation of the mathematical mode the various controllers will be designed and implemented on the simulation program. The behaviour of the simulated system will be inspected.

From the simulation results the mechanical design specification will be determined. The mechanical design will then commence to create a physical model that provides an acceptable representation of the mathematical model.

During manufacturing of the mechanical design the electronic design will start. Conceptual designs will be done to determine the various designs capable of meeting the requirements. The electronic design will be tested to ensure it performs as designed with the opportunity to create revisions of the design.

The programming of the microcontroller, from now on as μC, will start. This includes the programming of the controller, data acquisition and the conversions of the sampled data.

System identification will then occur to determine the various parameters required by the controllers. These newly determined parameters will then be implemented on the simulation program. New controllers will be designed and verified in simulation.

The new controllers are then implemented onto the μC for the system experiments to start. From these system experiments the responses of the experiment will be compared by those of the simulation.

The report was written throughout the sequence of steps mentioned above and will be completed and reviewed at the end.

1.5 Report Outline

A brief overview of each chapter in the report will provided here. It acts as primer for the reader and the easy identification of section that may interest the reader more.

Chapter 2 explain the system concepts that is reference throughout the report. It contains the mathematical derivation of the robotic gymnast and the simulated model. The system parameters with system identification test are shown and whether the simulated models is a acceptable representation of the physical model.

Chapter 3 describes the design paradigm to swinging and balancing of the robotic gymnast. The specification for the controllers and the simulated responses of the controller are provided.

Chapter 4 contains the designs of the mechanical and electronic system of the project. The various components used in the designs and it explains how the physical model state information is provided to the designed controllers.

Chapter 5 describe the software implemented to reduplicate what has been done in the project. It explains the architecture of the software and the various functions implemented.

Chapter 6 provides the practical results of the controllers implemented in chapter 3 and hypothise unexpected behaviour in the experiments.

Chapter 7 concludes the report with a summary of the report and recommendation for future endeavours on the Feedback Control of a Robotic Gymnast.

Chapter 2

Conceptualisation and Modelling

2.1 System Concepts

The report contains many variable names and use of terminology for concepts that is used throughout the report. These variables and terminologies are defined here.

The double pendulum is a under-actuated system which is defined as a system where the input to the system cannot command one of the state variables to in instantaneous change in position.

The robotic gymnast is describe as a double pendulum consisting out of an actuated- and non-actuated pendulum as seen in Figure 2.1. The position of the non-actuated pendulum is described by the angle θ whereas the actuated pendulum is described by ϕ relative to θ.

underactuation, equilibrium position, unstable stable

2.2 Mathematical Model

The approach taken to derive the mathematical model of the robotic gymnast will be discussed in this section. It is presented to allow the reader to understand parameters mentioned in the report. The swing-up of the robotic gymnast consist of non-linear behaviour and is require to fully derive the dynamics of the system. The strenuous mathematical steps are removed from the reader, which is provided in Appendix A.1, and a summary of the motivation and paradigm approach to the derivation is provided here.

The robotic gymnast is modelled as two pendulums connected together with a hinge, where each pendulum is modeled as having their mass distributed arbitrary along their axis. There is a torque actuating the lower pendulum seen in Figure 2.1 and friction is modelled as proportional to the angular velocity

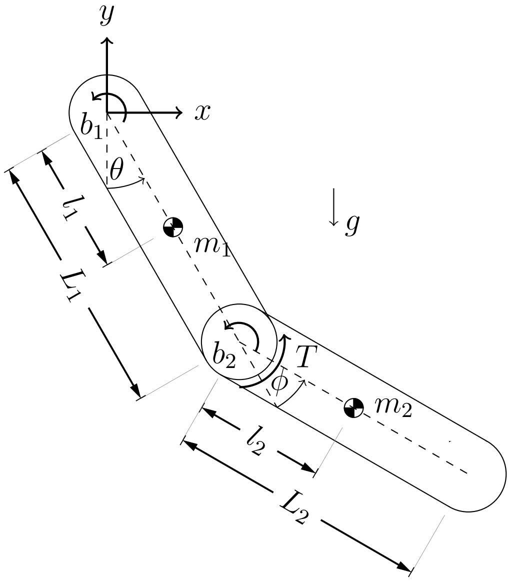

Figure 2.1: Free Body Diagram of the Double Pendulum

of the pendulums. The friction that develops at the hinge is proportional to the relative motion of the actuated pendulum and non-actuated pendulum. The angle, ϕ was purposefully chosen relative to θ to ease the identification of this friction. Figure 2.1 displays the free body diagram of the robotic gymnast.

Deriving the equation of motion of the robotic gymnast can be approached using different methods. The method chosen in the report is by using the Euler-Lagrange equation. The Euler-Lagrange equation shown in (2.1) derives the dynamics of the system by analysing the energy of the system. This approach was chosen due to the energy of the system being easily defined as the potential energy T of the 2 pendulums, and the kinetic energy V of the pendulums.

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \vec{\dot{q}}} - \frac{\partial \mathcal{L}}{\partial q} = 0 \tag{2.1}$$

$$\mathcal{L} = T - V \tag{2.2}$$

Using the Euler-Lagrange equation leads to the condense equations ofm otions shown in (2.3) and (2.4),

$$d_{11}\ddot{\theta} + d_{12}\ddot{\phi} + h_1 + \psi_1 = 0 \tag{2.3}$$

$$d_{21}\ddot{\theta} + d_{22}\ddot{\phi} + h_2 + \psi_2 = \tau \tag{2.4}$$

where the coefficients are defined as

$$d_{11} = I_a + I_b + m_2(L_1^2 + l_2^2 + 2L_1 l_2 \cos(\phi)) \tag{2.5}$$

$$d_{12} = I_b + m_2(l_2^2 L_1 l_2 \cos(\phi)) \tag{2.6}$$

$$h_1 = -m_2 L_1 l_2 \sin(\phi)\dot{\phi}^2 - 2m_2 L_1 l_2 \sin(\phi)\dot{\phi}\dot{\theta} \tag{2.7}$$

$$\psi_1 = (m_2 l_1 + m_2 L_1) g \cos(\theta) + m_2 l_2 g \cos(\theta + \phi) \tag{2.8}$$

$$d_{21} = I_b + m_2(l_2^2 + L_1 l_2 \cos(\phi)) \tag{2.9}$$

$$d_{22} = I_b + m_2 l_2^2 \tag{2.10}$$

$$h_2 = m_2 L_1 l_2 \sin(\phi)\dot{\theta}^2 \tag{2.11}$$

$$\psi_2 = m_2 l_2 g \cos(\theta + \phi) \tag{2.12}$$

2.3 Simulation Model

The mathematical model derived in the previous section is required to be implemented on a simulation program. Simulating the model allows the designer to understand how system parameters influence the dynamics of the system and the verification of controllers implemented. It will be presented by discussing how the simulation is a true representation of the system, how design specifications were determined and the influence of these parameters.

Simulation of the robotic gymnast was done using MATLAB Simulink. The differential equations shown in equation (2.3) and (2.4) were implemented using the MATLAB function box. It was required to write these equation where $\ddot{\phi}$ and $\ddot{\theta}$ is the subject in each of the *MATLAB Function* box to allow MATLAB to simulate the model.

Other non-linearities introduced by sensors and components were added such as saturation of the motor, gearbox backlash and quantisation of sensory data. These non-linearities were implemented to allow the simulation to be an acceptable representation of the physical system. The Simulink Model is shown in Figure 2.2.

2.4 System Identification

The system identification tests are done to determine the characteristics that describe the behavior of the system. These characteristics include the damping ratio's and natural frequencies of the system around the unstable equilibrium position. The system are described by 2 independent parameters, θ and ϕ which defines the position of the non-actuated and actuated pendulums.

The project started off with a previous physical model which provided realistic system parameters to allow the simulation to be a acceptable representation of reality. From using these previous system parameters the simulation provided a set of specification for the new design. The newly design model

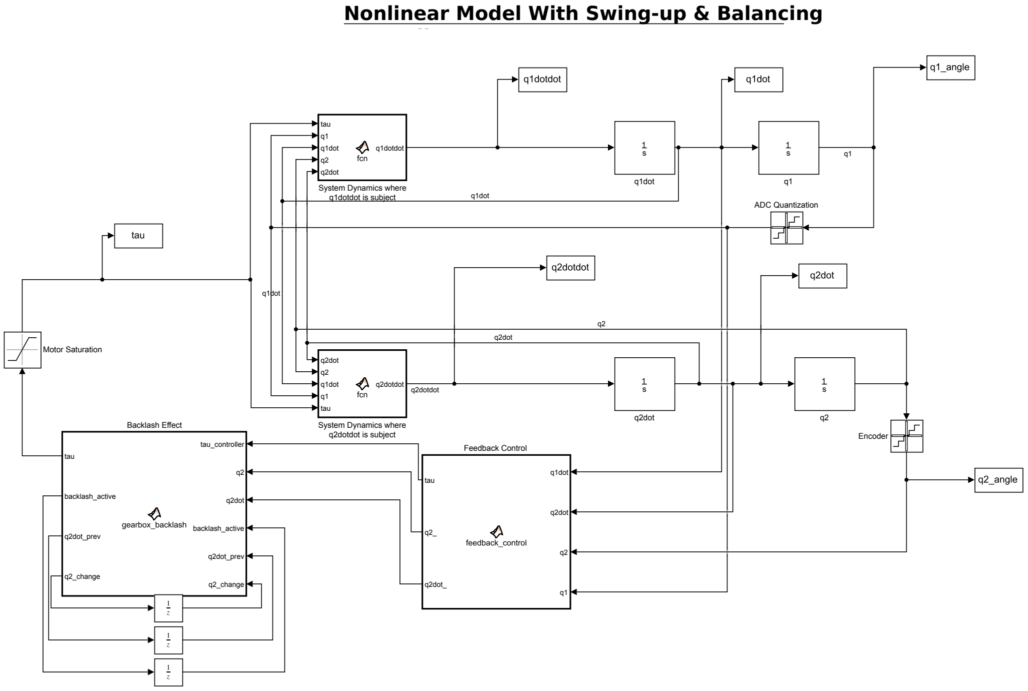

Figure 2.2: MATLAB Simulink Model

System Parameter	Value
L_1	$0.235\,\mathrm{m}$
L_2	$0.245\,\mathrm{m}$
I_A	$0.0265\,\mathrm{kg}\cdot\mathrm{m}^2$
I_B	$0.0221\,\mathrm{kg}\cdot\mathrm{m}^2$
m_1	$0.5763\,\mathrm{kg}$
m_2	$0.4928\,\mathrm{kg}$
l_1	$0.235\,\mathrm{m}$
l_2	$0.245\,\mathrm{m}$

Table 2.1: System Parameters

parameters are used throughout the report and the system identification is provided here to proof the simulation is a acceptable representation of the physical model. The physical model parameters is shown in Table 2.1.

Since the system is described by 2 independent parameters the system is thus a 2 degree of freedom (2DOF) system and is expected to contain 2 natural frequencies each accompanied by a damping coefficient.

The first natural frequency of the system is determine by inspecting the response of the system when starting at a initial condition and keeping $\phi = 0\,\mathrm{rad}$ constant throughout the response. This was done by using a lightweight PVC pipe that has negligible effect on the weight of the system. The actuated pendulum and non-actuated pendulum are constrained to this pipe to ensure the 2 pendulums stay in-line with each other and thus ensuring $\phi = 0\,\text{rad}$. The response of the system is shown in Figure 2.3 starting at a initial condition of roughly $\theta = \frac{\pi}{2}$. The accuracy of the initial conditions is of little importance, but the initial condition must allow the response to contain a few oscillation to accurately determine the parameters of interest.

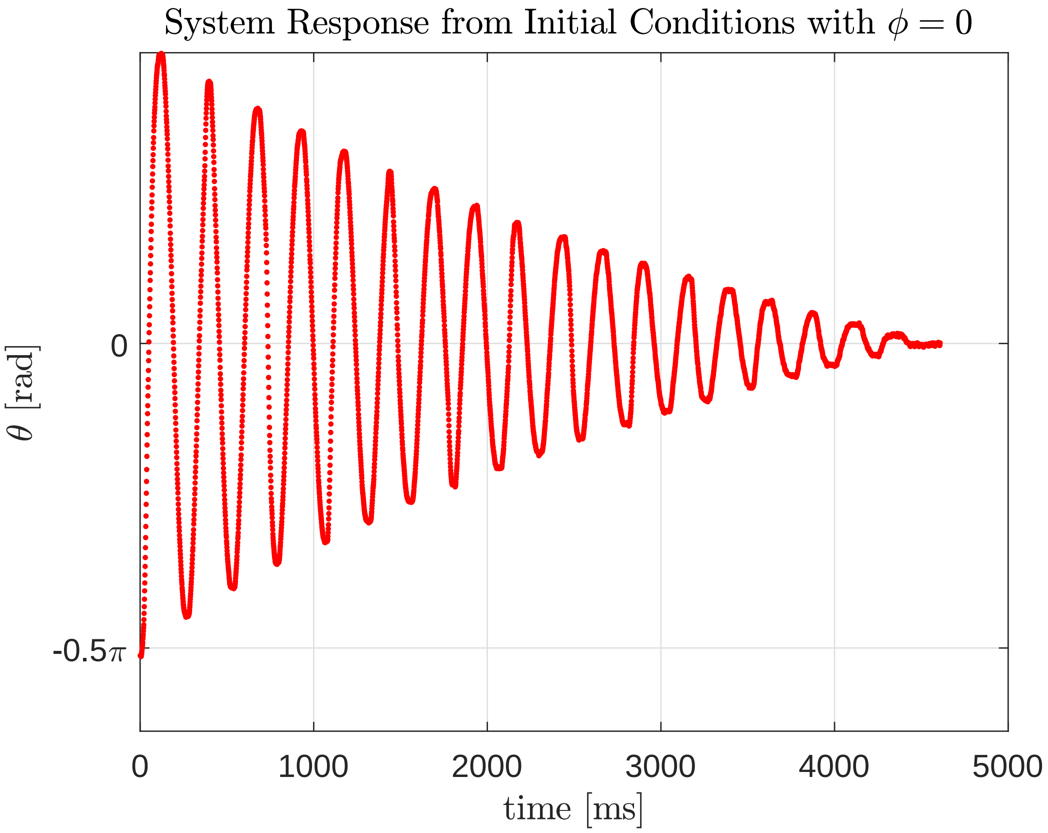

Figure 2.3: Initial Condition System Response while $\phi = 0\,\text{rad}$

The second natural frequency is determined by analysing the response of the system when ϕ starting at a initial condition and keeping $\theta = 0\,\text{rad}$ throughout the response. This was accomplished by constraining the non-actuated pendulum using hard stops. Figure 2.4 shows the measured response of the system when ϕ starts at a initial condition and keeping θ constant.

The natural frequencies of the system is identified by inspecting the frequency content of the time-domain responses. The frequency content of the initial condition responses of both experiments are shown in Figure 2.5, by applying the Fast Fourier Transform (FFT) algorithm to the time-domain signals. The FFT indicates the natural frequencies by the predominant peaks

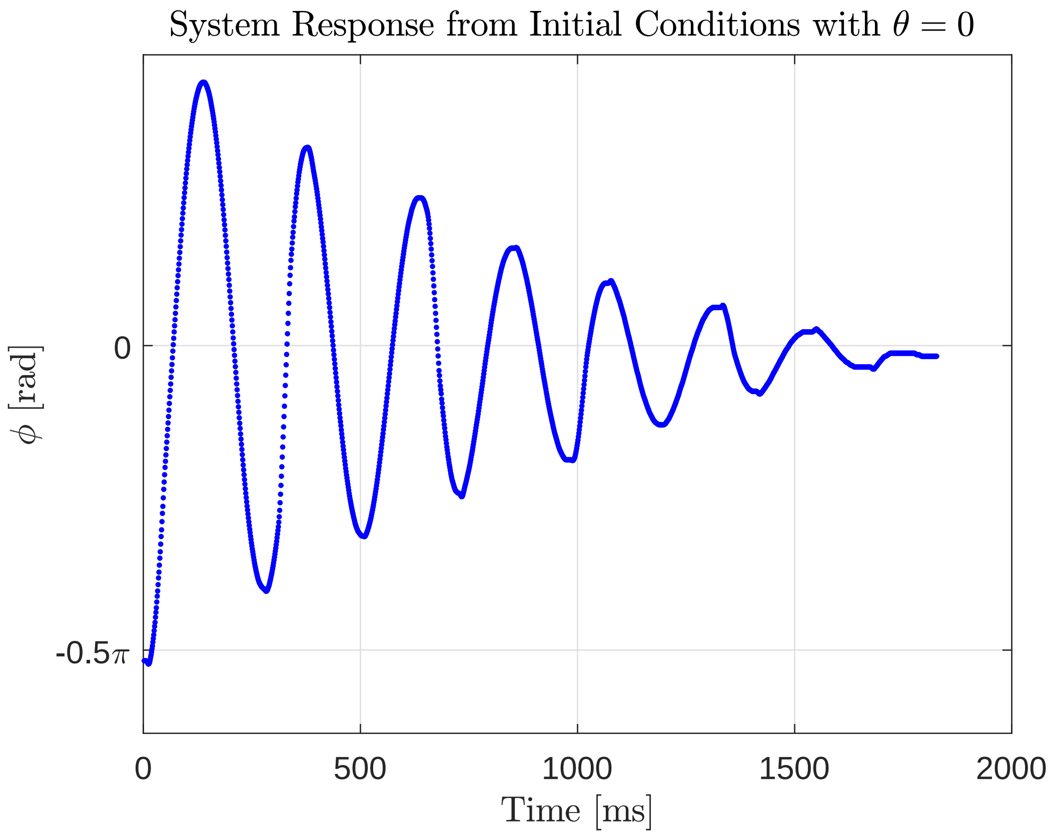

Figure 2.4: Initial Condition System Response while $\theta = 0\,\text{rad}$

across the frequency spectrum which are tabulated in Table 2.2.

The responses shown in Figure 2.3 and 2.4 are decaying with time and this decaying behaviour can be modeled by the following equation:

$$\tau(t) = -\zeta\omega_n t$$

Where ω_n is the natural frequency of the system and ζ the damping ratio of the system. The natural frequencies of the system has already been determined and thus linear regression can be used to determine the best ζ that will fit the measured data. The decaying functions are shown in Figure 2.3 and 2.4 with the best fit ζ values shown in Table 2.2. It is visible from the responses that the damping ratio changes when the response are nears steady state. This change of damping ratio is neglible due to the controllers being implement are robust to changes in ζ.

Figure 2.3 shows the decaying function with properties that are shown in Table **??** and it is visible that the calculated values provides a acceptable approximation of the measured data.

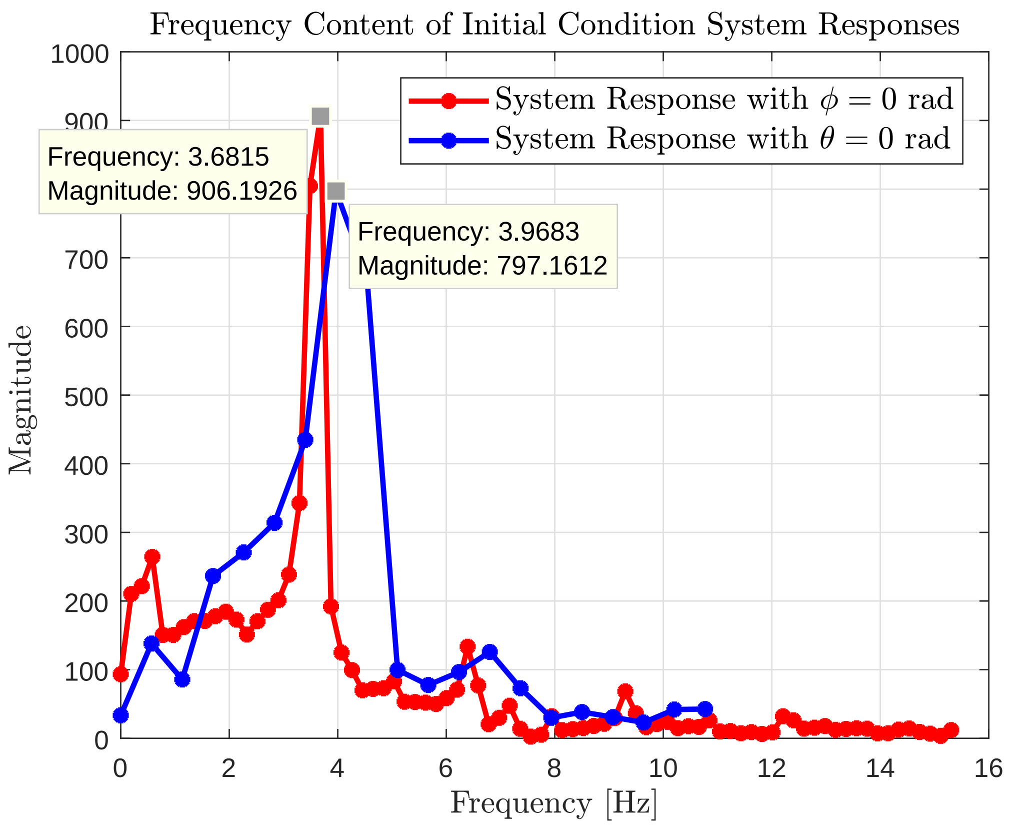

Figure 2.5: Frequency Content of Time-Domain Initial Condition Responses

System Characteristic	Mean	Standard Deviation
f_1	3.93 Hz	
f_2	3.62 Hz	0.04029
ζ_1	$1.6931 \cdot 10^{-5}$	$4.823 \cdot 10^{-6}$
ζ_2	$5.85184 \cdot 10^{-5}$	

Table 2.2: System Characteristic & their Statistical Properties from 10 Experiments

2.5 Model Validation

The model implemented in simulation must be able describe the physical model to an acceptable degree to allow any further developments on the simulated model. The simulated model will be validated by comparing the experimental system characteristic values to those attained in simulation.

System Characteristic	Experiment	Simulated	Difference
w_{n_1}	3.93 Hz		
w_{n_2}	3.62 Hz	0.04029	

Table 2.3: Comparison between Experimental- and Simulated System Characteristic

Chapter 3

Feedback Control Design

3.1 Balancing Controller

Provided the robotic gymnast is within the region of controllability, a balancing controller is required to bring the system to the unstable equilibrium position. The design approach is based on the premises that the swing-up controller will swing the robotic gymnast within the region of controllability where the balancing controller will steer the robotic gymnast to balancing.

The independent parameters, ϕ and θ, will be condensed from now on as a vector describes as

$$\vec{q} = \begin{bmatrix} \theta \\ \phi \end{bmatrix}$$

The system can be approximated as a linear system when in the vicinity of the unstable equilibrium position. The system is thus linearised at

$$\vec{Q}_s = [\vec{q_s}, \dot{\vec{q_s}}, \ddot{\vec{q_s}}]^T = [\pi, 0, 0, 0, 0, 0]$$

the unstable equilibrium position using the Taylor Series Expansion shown in detail in Appendix B. This linearised model can than be written in the state space form to implement a feedback gain. The state space variables are chosen as $\Delta\vec{q}$ and $\Delta\dot{\vec{q}}$ which results in the state space representation as:

$$\dot{\vec{q}} = \boldsymbol{A}\Delta\vec{q} + \boldsymbol{B}u$$

and

$$\vec{y} = \boldsymbol{D}\Delta\vec{q} + \mathbf{0}u$$

.

The linearised system remains a coupled system which results that the quadratic eigenvalue problem is required to be solved to identified the poles of the system. The solved quadratic eigenvalue problem results in the following poles from the system parameters in Table 2.1.

$$\vec{s} = \begin{bmatrix} -8.2 \\ -4.2 \\ 8.2 \\ 4.2 \end{bmatrix}$$

The poles of the system are pairs of positive and negative real poles that indicates an unstable system. This is expected due to the system being linearised in the unstable equilibrium position. When the linearised system is at rest, any disturbance will result in a theoretically infinite growth of the state variables, but this behaviour can be controlled by introducing feedback.

These poles will be moved to the desired position by using the method of dominant poles. The method of dominate poles chooses a pair of the poles for the closed-loop system and select the other open-loop poles to have real parts. This allows the higher-order system response to be characterised as a second-order response Gene F. Franklin (2015). The linearised model already has 2 negative real poles which is chosen to stay the same. The real positive poles were selected based on the following specifications.

These specification was selected on increasing the null controllability region with a low settling time. Due the approximation of the linearised model the null controllability region may be larger than the derived sized sim (2006).

3.1.1 Controller Architecture

3.1.2 Requirements/Specifications and Constraints

3.1.3 Plant Linearisation

3.1.4 Full State Feedback Design

3.1.5 Simulation Response

3.2 Swingup Controller

The swing up of the robotic gymnast is done by the non-linear feedback control system. For the robotic gymnast to swing from the stable equilibrium position to the unstable equilibrium position the feedback control must incorporate the non-linearities of the system. How these non-linearities of the system is incorporated and the design paradigm of the approach will be explained in following text.

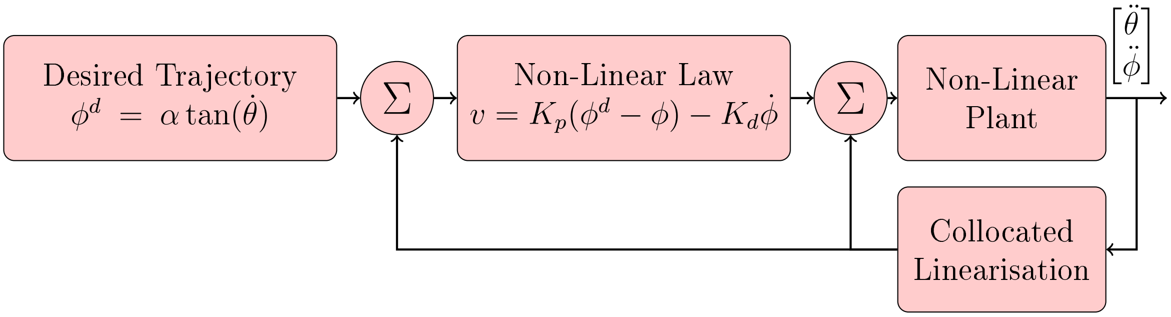
Desired Trajectory
$\phi^d = \alpha\tan(\dot{\theta})$
Σ
Non-Linear Law
$v = K_p(\phi^d - \phi) - K_d\dot{\phi}$
Σ
Non-Linear Plant
$\begin{bmatrix}\ddot{\theta}\\ \ddot{\phi}\end{bmatrix}$
Collocated Linearisation

Figure 3.1: Block Diagram of the Non-Linear Controller

3.2.1 Controller Architecture

Figure 3.1 shows a high-level block diagram of the swing-up controller. The swing-up controller consist out of multiple parts that is required to work in unison to allow the robotic gymnast to swing.

The collocated linearisation block is the foundation on which the entire controller is built. Therefore the other controller blocks are dependent on the successfull implementation of the collocated linearisation.

The swing-up controller is a combination of classical control theory and a more advance controller. The classical control theory is implemented by the proportional gain of the error and a derivative gain seen in the non-linear control law. The more advance controllers are implemented by implementing a non-linear controller.

Each of the blocks are explained in the following sections

3.2.2 Requirements/Specifications and Constraints

The requirements of the swing-up controller is to swing the robotic gymnast up under 30 seconds to the unstable equilibrium positions. In the vicinity of the unstable equilibrium position the when the non-actuated pendulum is upright $\theta = 2\pi\,\text{rad} \pm \frac{\pi}{30}$ the controller must position the actuated pendulum between $\phi \in [-5°, 5°]$.

The first requirement is set to provide a feasible solution to the swing-up of the robotic gymnast and allow the swing-up sequence to be captivating. The second requirement is set to ensure the linear approximation holds when the balancing controller is active to bring the system to the unstable equilibrium position and balance.

Constraints that is placed on the controller is the stall torque of the motor.

3.2.3 Feedback Linearisation

It has been shown that it is not possible to linearise the dynamics of the gymnast by means of static state feedback and non-linear transformation mur (1990), but it is possible to achieve a linear response from one of the state variables by implementing a non-linear feedback. This non-linear feedback is the partial feedback linearisation.

Collocated linearisation is a form of partial feedback linearisation where a non-linear control input τ is used to linearise the response of the actuated pendulum. By analysing equation (2.4), the input τ is chosen to cancel all the non-linearities of the system and add an additional outer loop control input v_2. This outer loop control input can be selected to force the actuated pendulum to follow a desired trajectory (spo, 1995). The derivation of the collocated linearisation is shown in Appendix **??**.

$$d_{11}\ddot{\theta} + h_1 + \psi_1 = -d_{12}v_2 \tag{3.1}$$

$$\ddot{\phi} = v_2 \tag{3.2}$$

The subtle practical implication of using collocated linearisation is that the system being controlled must be well defined. If this is not the case the non-linear input τ will introduce other unwanted dynamics not describe my the model.

3.2.4 Nonlinear Control Law

The ability to control the actuated pendulum to follow a desired trajectory, provides the possibility to increase the energy of the system if the correct trajectory is chosen. The increase of energy in the system will cause the pendulums to rise from their stable equilibrium position and thus start swinging upwards. The desired trajectory for ϕ is chosen as equation (3.3) determined by W.Spong in (spo, 1995).

$$\phi^d = \alpha \arctan(\dot{\theta}) \tag{3.3}$$

The outer loop control input, v_2, is chosen as

$$v_2 = K_p(\phi^d - \phi) - K_d\dot{\phi} \tag{3.4}$$

The desired trajectory is chosen to increase the energy in the system, approach is to allow the actuated pendulum swing in phase with the non-actuated pendulum. By this approach the energy of the actuated pendulum is transferred to the non-actuated pendulum spo (1995). Appendix E shows how this approach is expected to work.

Returning to the desired trajectory of ϕ, the coefficient α constrains the actuated pendulum to stay within a interval of $\phi \in [-\alpha, \alpha]$ spo (1995). This provides better control over the system to stay within the null controllability region when the system reaches the unstable inverted position.

3.2.5 Simulation Response

3.3 Simulation Results

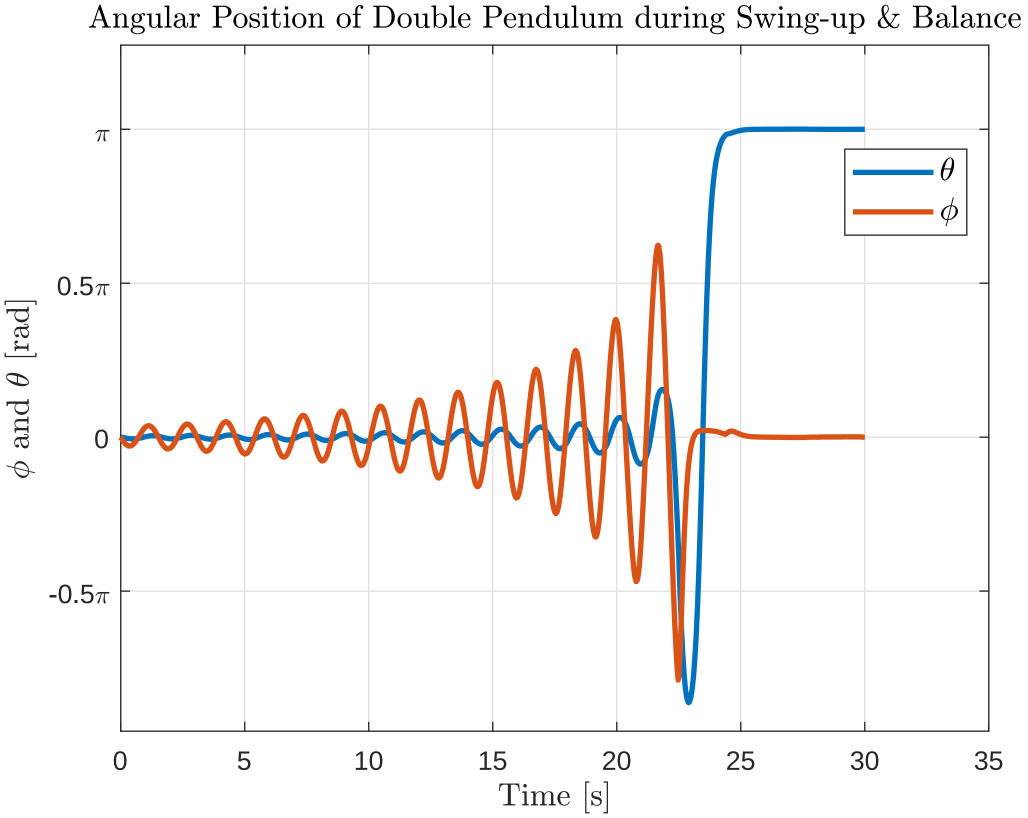

Figure 3.2: The Swing-Up & Balancing of the Robotic Gymnast

Chapter 4

Hardware Design and Implementation

4.1 Mechanical Hardware

4.1.1 Mechanical Components

4.1.2 Motor

4.1.3 Assembly

4.2 Electronic Hardware

4.2.1 System Description

Figure 4.1 provides a system overview and how the different parts functions together. The micro-controller receives the different signals that has been correctly conditioned from supporting circuitry to interpret the dynamics of the system. From the observed condition it is able to output the correct signals to instruct the next command.

The digital logic circuit that consist of logic level converters acquires the signal from the micro-controller and performs signal conditioning to interface with the motor driver and determines the correct direction to rotate the motor.

The motor driver controls the DC brushed motor based of the digital signals and provides a proportional feedback current that is delivered to unity-gain amplifier.

The motor contains a encoder that indicates the direction and position of the rotor through digital signals that is sent through a digital logic filter to retrieve only critical information from the encoder signals.

The physical model contains a potentiometer that measures the non-actuated pendulums angle and is sent to the buffer.

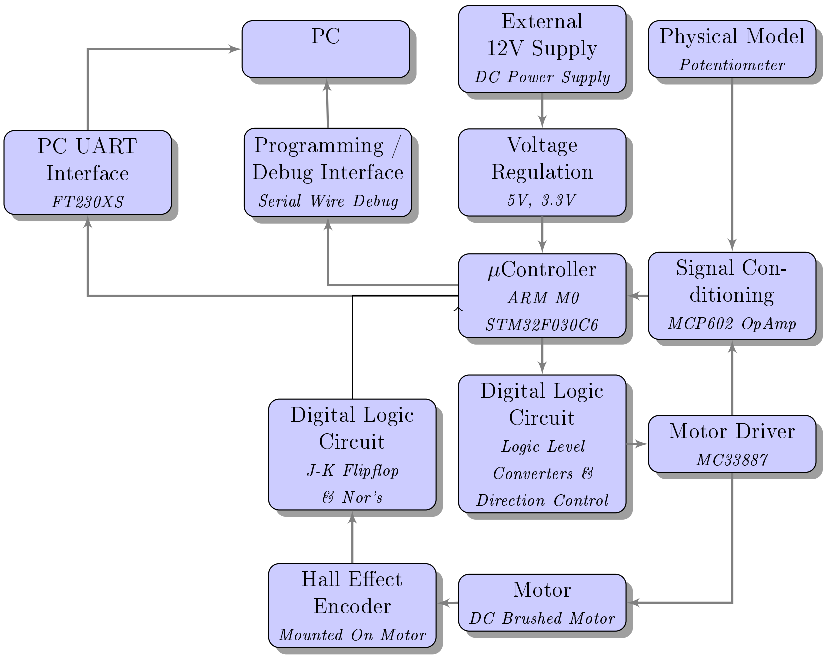

Figure 4.1: Electronic System Overview

The microcontroller will use the UART interface as it's data acquisition protocol to send the necessary information to the computer.

The micro-controller is programmed using the Serial Wire Debug (SWD) protocol to transfer the binaries from the computer.

Power is provided using a external 12V power-supply, which will power the motor, but also using a regulator to down convert/step to a 5V and 3.3V to power the microcontoller and the other peripherals.

4.2.2 Microcontroller

The microcontroller chosen is the STM32F030Mxx. The selection was done according the ease of setting up, memory size, physical dimensions and the peripherals it provided.

The STM32F030MXX is based of the ARM M0 architecture which is ARM's entry level micro-controller. It requires little support to have a up and running microcontroller requiring only the SWD protocol to program and a few by-pass capacitors.

It was difficult to determine the memory size specification for the project. This uncertainty ensured that the largest memory size the ARM M0 architecture could provide was selected.

The Electrical and Electronic Department's Printed Circuit Board (PCB) manufacturing machine can only provide a minimum track width 0.3 mm. This resulted in choosing a microcontroller whose footprint would meet the requirement.

Component	Supply Voltage [V]
Digital Logic Components	5
μController	3.3
Motor Driver	12

Table 4.1: Suppy Voltage's for the different components

Based on the conceptual design, the chosen microcontroller required to contain 2 ADC's, minimum of 5 GPIO's and 1 serial communication peripheral.

4.2.2.1 Programming / Debug Interface

The *Atollic TrueSTUDIO for ARM 8.0.0* Integrated Development Environment (IDE) is used for writing the source code which converts the source code to the Executable and Linkable Format (.elf) file. These .elf files is then written using the Serial Wire Debug (SWD) protocol to the μC. Debugging of the source code occur using the same IDE which allows the programmer to inspect variables, timers and logic.

4.2.2.2 PC UART Interface

The purpose of the UART to serial communication is for data acquisition of the system response and for debugging purposes. The data being sent follows a structure to ensure the reliability of the data. Figure (ref) shows the format of the data being sent.

The data being sent across the UART to serial circuit is retrieved by a computer executing a Python script, listening for any activity on the computer's driver ports and writing the data into a comma-separated value (csv) file that can later be use to analyse the data.

The UART to serial circuit has been tested by doing a loopback test and using a digital logic analyser to verify the data being sent. The loopback test consist of connecting the Tx and Rx lines together and forcefully echo what has been sent to the circuit to be sent back. Figure (ref) in Appendix XXX shows the digital signals sent and received and confirms the working of the UART to Serial circuit.

4.2.3 Voltage Regulation

The various components required different supply voltages in the electronic design. The diffenrent supply voltage is tabulated in table 4.1.

The table indicates 3 different supply voltages that will be required: 3.3 V, 5 V and 12 V. This was achieved by using a 5 V and 3.3 V linear voltage regulators and the 12 V is supply by a external source.

The schematic for each voltage regulator is shown in Appendix XX, where each voltage regulator circuit includes a Light Emitting Diode (LED) to ensure the minimum load is met for each regulator. The LED also acts as a visual debugging method.

4.2.4 Potentiometer Sensor

4.2.4.1 Working Principle

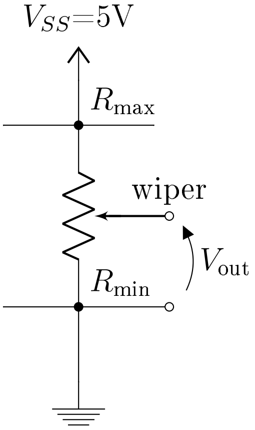

Figure 4.2: Simplified Model of a Potentiometer

The rotary position potentiometer consist out of a wiper that is attached to a rotating shaft. The wiper relative position to a internal resistor is proportionally changed with rotation. The position of the wiper will output a voltage signal that is proportional to the position of the wiper relative to the resistor.

4.2.4.2 Interface

The signal produced by the rotary potentiometer varies from 4.95V and 50mV from 360°to 0°respectively. This is signal sent through a simple voltage divider circuit to reduce the signal to 3V to 15mV to be within the sampling limits of the micro-controller. The reduce signal is then sent through a rail-to-rail unity gain amplifier before being sampled by the ADC of the microcontroller.

The scaled voltage is sent through a unity gain rail-to-rail amplifier, where the mirrored output signal is fed into the ADC. The type of ADC used in the STM32F030XX is a successive approximation register (SAR), (stm, 2017). The SAR ADC's contains internal capacitors that suffers from the effect of being depleted if the sampling frequency is to high. Using an operational amplifier reduces the risk of depleting this internal capacitor because of the low output resistance.

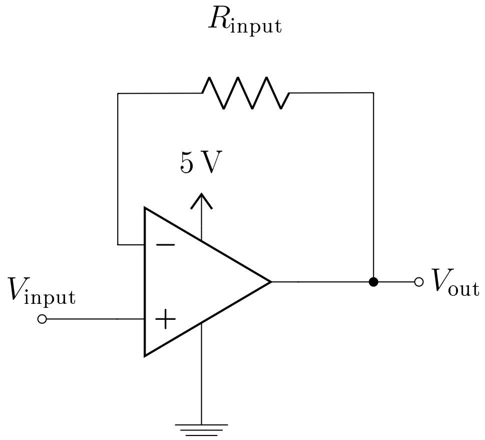

Figure 4.3: Unity Gain Amplifier Circuit

4.2.5 Magnetic Encoder

4.2.5.1 Working Principle

A rotating gear containing ferrous metal teeth is attach to the rotating shaft. The rotating metal teeth rotates near a hall-effect sensor which creates a change in the magnetic flux inside the hall-sensor. This change in magnetic flux is sensed by the hallsensor which produces a digital signal (Instruments, 2006).

4.2.5.2 Digital Interface

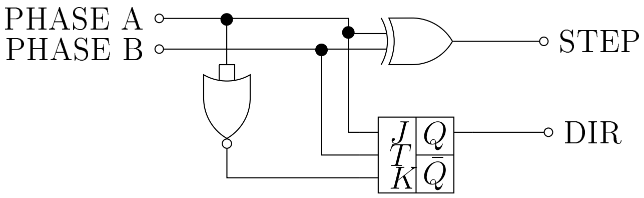

Figure 4.4: Digital Logic Circuit containing JK-Flipflops, XOR- and NOR Gates

The encoder contains a sold state hallsensor which provides 2 channels with a 90 degree phase difference (Faulhaber, 2011). These 2 signals under go a hardware filter that produces 2 signals that indicate the direction of the motor and the incremental position.

The hardware filter consist out of XOR, NOR and JK-Flipflop gates shown in Figure 4.4 and the schematic in Figure ??. The XOR gate produces the incremental position of the motor whereas the NOR and JK-Flipflop combination produces the direction of the motor by a logical 1 or 0.

The hardware filter was implemented to reduce the processing time of the micro-controller on the original signals.

These signals will be read by the μC using interrupts when a rising- & falling edge are present.

The encoder provides 16 lines per revolution, equal to a combined 32 rising- & falling edges per line. The encoder provides 2 lines increasing the resolution to 64. The motor is connected to a gearbox with a reduction stage of 14:1. The encoder will thus rotate 14 times per shaft revolution, increasing the resolution to 896 per revolution.

4.2.6 Motor Driver

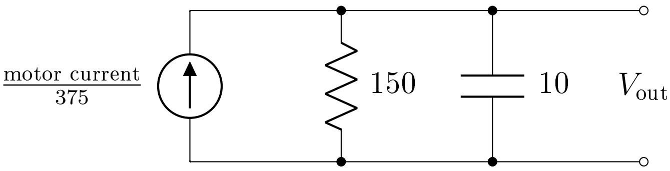

Figure 4.5: Simplified Circuit of Motor Feedback

The motor driver IC is connected directly to the motor and responsible for directional and rotational control of the brushed DC motor. The motor driver is the MC33887 and contains 2 half H-bridges that forms a full H-bridge which are Pulse-Width-Modulated (PWM) to control the speed of the motor. The PWM- and direction signal originates from the micro-controller. As discussed previously, the signals' logic level is first converted and than sent through the AND digital filter before the motor driver receives it.

The MC33887 provides a proportional current of 1/375 of the current flowing through the high-side of the full H-bridge (mot, 2007). This current is sent through a resistor of 150 Ω to provide a voltage signal to represent the current. Due to the motor being controlled using PWM, the current through the motor is periodic impulse signals. The use of a parallel capacitor is used to create a ripple voltage to allow the ADC of the microcontoller enough time to sample. This ripple voltage is sent through a unity-gain amplifier as seen in Figure 4.3 before it is sampled by the microcontroller. This closes the feedback loop to implement torque control by the control system.

The MC33887 is capable of providing up to 6A of continuous current to the motor, while withstanding the high current transients due to /the fast switching of a inductive load (mot, 2007). The motor driver IC provides the motor with 12V DC which is externally provided by a DC power supply. The schematic of the motor driver is shown in Appendix XXX.

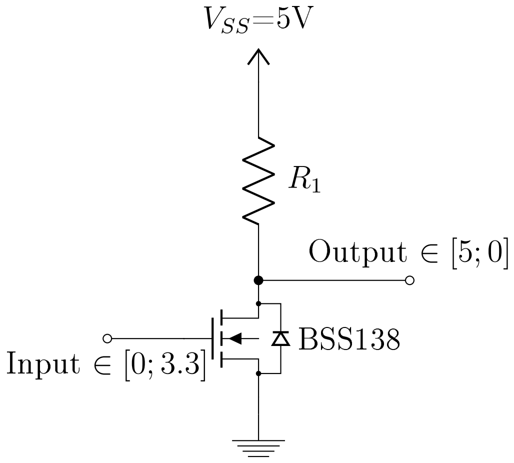

Figure 4.6: Logic Level Converter & Inverter Circuit

4.2.6.1 Logic Level Converters

The μC is required to interface with the motor driver. The μC represent a logical high and low as a 3.3 V and 0 V respectively. The motor driver IC's logical high threshold is 3.5 V. It is thus required to use a logic level converter to allow reliable communication between the two devices.

The logic level converter used is the circuit shown in Figure 4.6 using the BSS128 transistor. The circuit has the side effect of being a inverter. A logic low, 0 V by the μC will be converted to a 5 V and a logic high, 3.3 V will be converted to 0 V. This side effect is overcome by inverting the logic in software.

4.2.6.2 AND Digitial Circuit

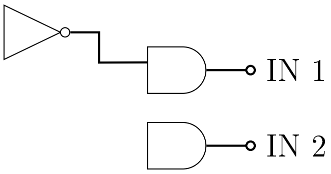

Figure 4.7: AND digital logic with inverter

The AND digital gates in combination with a inverter shown in Figure 4.7 is responsible for providing the motor IC's with the desired direction signals alongside the PWM signal produced by the μC.

The AND circuit receives 2 signals from the μC after it has been converted to the correct logic level: the PWM signal and a logic level signal indicating the desired direction. Based on the directional signal the AND circuit will switch the PWM signal between the 2 inputs of the motor IC's while holding the other low.

This switching of PWM signal between the 2 inputs of the motor IC and keeping the other signal low controls the direction of the motor. This hardware directional control was done in order to reduce the processing time of the μC is required to do.

4.2.7 Verification Tests

4.2.7.1 Angle Sensor Measurements

4.2.7.2 Optical Encoder Measurements

4.2.7.3 PWM Duty Cycle to Motor Current

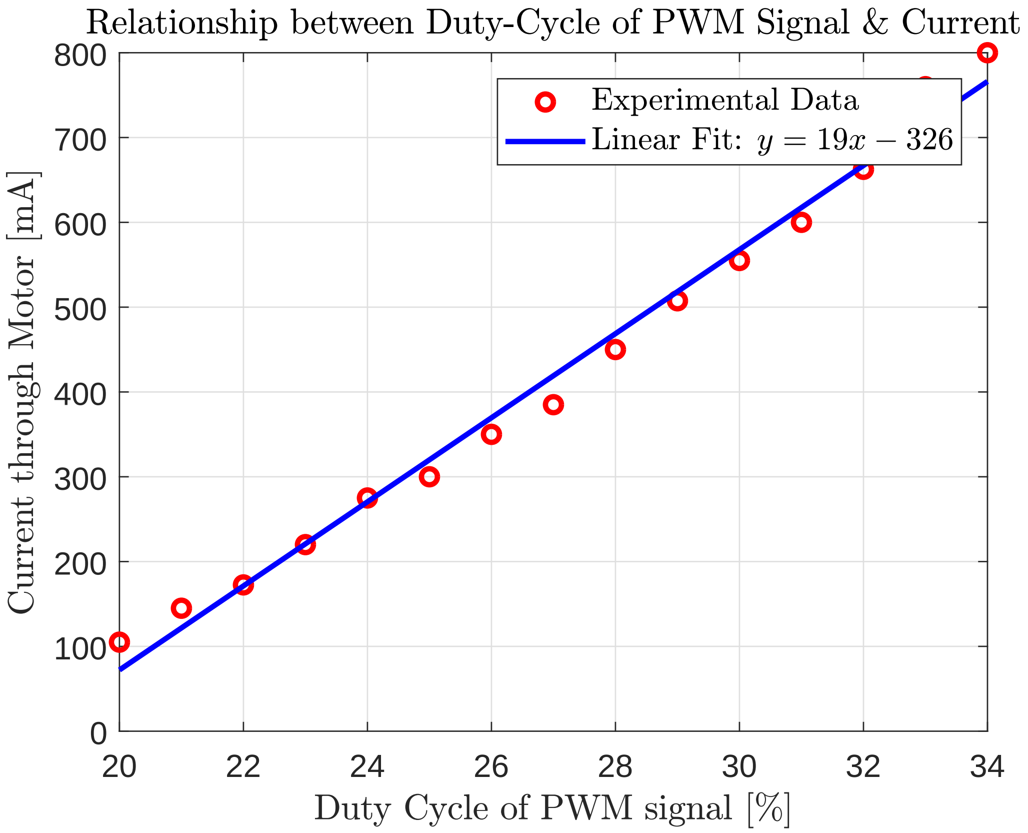

Figure 4.8: Relationship between Duty-Cycle of PWM Signal and Current through Motor

The input to the system is the torque delivered by the motor and the magnitude and direction is determined by the control laws. The model describing the system in equation 2.4 assumes the torque delivered to the system is instantaneously available. This is inaccurate due to the DC motor model describing the torque delivered by the model when applying a DC voltage. The input that the motor will be receiving is a Pulse-Width-Modulated (PWM) signal to represent a DC voltage.

Experiments were done to determine the relationship between the duty-cycle of the PWM signal and the torque delivered by the motor. These experiments are done by incrementing the duty-cycle of the PWM signal that the motor receives when the shaft is kept fix against a hard-stop. The motor-driver that controls the motor has a feedback pin that provides a proportional current of $\frac{1}{375}$ of the current flowing through the motor. This proportional current is allowed to flow through a known resistor. The voltage across the resistor is measured and provides a indication of the current flowing through the motor. The mean value of the voltage measured on the oscilloscope is taken and mapped backwards to determine the current. Figure 4.8 shows the measured data with a line of best fit. It is clear that there exist a linear relationship between the duty-cycle of the PWM signal and the torque the motor can provide.

Chapter 5

Software Design

The software required to allow data acquisition, retrieval of system state information and communication is presented here. The software design played a central role in continuing of the project pass the initial simulation phase.The software design allowed the determination of the system characteristic parameters and the verification of the simulation and controller.

5.1 Software Requirements

5.1.1 Data Aquisition

Communication with the microcontroller was implemented using the serial communication from the external computer to the microcontroller. Communication with the microntroller occures differently depending on the state of the system. A *Python* script was used for communication that runs on the external computer.

If experiments are conducted the communication is uni-directional from the microcontroller towards the external computer. The microcontroller streams the state variables of the system to the external computer in specific structure shown in Figure 5.1. The star attached to the variables indicate that they are not sent in the correct units. The reason behind the decision is to reduce the processing time of the microcontroller. Sending data types such as floats are computational hungry and thus these conversions are handled on the external computer.

0	1	2	3	4	5	6	7
$	time	,	θ^*	,	ϕ^*	,	τ^*

Figure 5.1: Data Structure for Streaming Data during Experiments

The structure used in Figure 5.1 is chosen as comma-seperated values csv) which makes it easy to write these data in a .csv file and read them later to analyse the data.

The other state in which communication occured was bi-directional used for debugging purposes. In this state the *Python* script allows the user to type commands adhering to the structure shown in Figure 5.2. Based on the command used, the microcontroller would echo the same command back if it completed the command instructed.

0	1	2	3	4	. . .	n-1	n
$	cmd	,	value	value		\r	\n

Figure 5.2: Data Structure for Sending Commands

5.1.2 Embedded Program

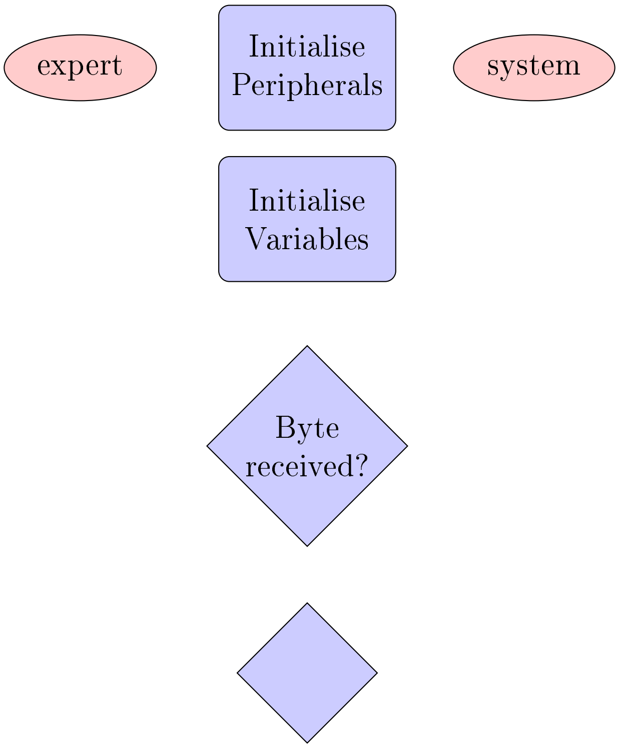

Figure 5.3: Data Structure for Sending Commands

Figure 5.3 shows the main execution flow of the microcontroller based on factors that influence their states. A brief overview of the execution flow is describe below.

On startup of the microcontroller, it will initialise the various peripherals required for operation. They include timers for PWM generation, ADC and interrupts for encoder signals. The microcontroller will than initialise the various variables required for operation.

Once initialisation is completed the microcontroller will check via an interrupt if a byte over the serial communication has been received. If a byte has been received it, the microncontroller will verify the command structure and execute based on this command.

Every 0.1ms the microcontroller will inspect the data arrays of the sampled signals. Direct Memory Access (DMA) is used for sampling and results in these sampling data being automatically refreshed by hardware.

The micrcontroller will than react whether a interrupt has occur to indicate a rising or falling edge of the encoder signal. This falling and rising edge indicates a incremental change of the position of the actuated pendulum the microcontroller will behave accordingly.

The microcontroller will than verify whether it is required to stream the data every 1ms to the external computer.

Chapter 6

Practical Results

6.1 Swingup Controller

6.2 Balancing Controller

6.3 Swingup and Balancing

Chapter 7

Conclusion

7.1 Summary

7.2 Recommendation

Appendices

Appendix A

Derivation of the Double Pendulum

A.1 Derivation of the Mathematical Model

$$x_1 = l_1 \cos(\theta)$$
$$y_1 = -l_1 \sin(\theta)$$

$$x_2 = L_1 \sin(\theta) + l_2 \sin(\theta + \phi)$$
$$y_2 = -L_1 \cos(\theta) - l_2 \cos(\theta + \phi)$$

$$\dot{x_2} = L_1 \cos(\theta)\dot{\theta} - l_2 \cos(\theta + \phi)(\dot{\theta} + \dot{\phi})$$
$$\dot{y_2} = L_1 \sin(\theta)\dot{\theta} + l_2 \sin(\theta + \phi)(\dot{\theta} + \dot{\phi})$$

$$x_2^2 = L_1^2 \cos(\theta)^2\theta^2 + l_2^2 \cos(\theta + \phi)^2(\dot{\theta} + \dot{\phi})^2 + 2L_1 l_2\dot{\theta}(\dot{\theta} + \dot{\theta}) \cos(\theta) \cos(\theta + \phi)$$
$$y_2^2 = L_1^2 \sin(\theta)^2\theta^2 + l_2^2 \sin(\theta + \phi)^2(\dot{\theta} + \dot{\phi})^2 + 2L_1 l_2\dot{\theta}(\dot{\theta} + \dot{\theta}) \sin(\theta) \sin(\theta + \phi)$$

$$x_2^2 + y_2^2 = L_1^2\theta^2[\cos(\theta)^2 + \sin(\theta)^2] + l_2^2(\dot{\theta} + \dot{\phi})^2[\cos(\theta + \phi)^2 + \sin(\theta + \phi)^2] +$$
$$2L_1 l_2\dot{\theta}(\dot{\theta} + \dot{\phi})[\cos(\theta) \cos(\theta + \phi) + \sin(\theta) \sin(\theta + \phi)]$$

Using the following trigonometric identities

$$\cos(\gamma)^2 + \sin(\gamma)^2 = 1$$

$$\cos(\gamma) \cos(\alpha) + \sin(\gamma) \sin(\alpha) = \cos(\gamma - \alpha)$$

the above equation resolves to:

$$V_2^2 = L_1\dot{\theta}^2 + l_2^2(\dot{\theta} + \dot{\phi})^2 + 2L_1l_2(\dot{\theta} + \dot{\phi})\dot{\theta}\cos(\phi)$$

The kinetic energy in the system consist of the fixed rotation of the underactuated pendulum and the rotation and velocity of the actuated pendulum.

$$T = \frac{1}{2}I_A\dot{\theta}^2 + \frac{1}{2}I_B(\dot{\theta} + \dot{\phi})^2 + \frac{1}{2}m_2V_2^2$$

$$T = \frac{1}{2}I_A\dot{\theta}^2 + \frac{1}{2}I_B(\dot{\theta} + \dot{\phi})^2 + \frac{1}{2}m_2[L_1\dot{\theta}^2 + l_2^2(\dot{\theta} + \dot{\phi})^2 + 2L_1l_2(\dot{\theta} + \dot{\phi})\dot{\theta}\cos(\phi)]^2$$

The potential energy in the system is defined as

$$V = -m_1gl_1\cos(\theta) - m_2g[L_1\cos(\theta) + l_2\cos(\theta + \phi)]$$

The Lagrange is defined as

$$\mathcal{L} = T - V$$

$$\mathcal{L} = \frac{1}{2}I_A\dot{\theta}^2 + \frac{1}{2}I_B(\dot{\theta}+\dot{\phi})^2 + \frac{1}{2}m_2[L_1\dot{\theta}^2 + l_2^2(\dot{\theta}+\dot{\phi})^2 + 2L_1l_2(\dot{\theta}+\dot{\phi})\dot{\theta}\cos(\phi)]^2 + m_1gl_1\cos(\theta) + m_2g[L_1\cos(\theta) + l_2\cos(\theta + \phi)]$$

$$\frac{\partial \mathcal{L}}{\partial \theta} = -m_1gl_1\sin(\theta) - m_2gL_2\sin(\theta) - m_2gl_2\sin(\theta + \phi)$$

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\theta}} = I_A\ddot{\theta} + I_B\ddot{\theta} + I_B\ddot{\phi} + m_2L_1^2\ddot{\theta} + m_2l_2^2\ddot{\theta} + m_2l_2\ddot{\phi} + 2m_2L_1l2\ddot{\theta}\cos(\phi) - 2m_2L_1l_2\dot{\theta}\dot{\phi}\sin(\phi) + m_2L_1l_2\ddot{\phi}\cos(\phi) - m_2L_1l_2\dot{\phi}^2\sin(\phi)$$

$$\frac{\partial \mathcal{L}}{\partial \phi} = -m_2L_1l_2(\dot{\theta} + \dot{\phi})\dot{\theta}\sin(\phi) - m_2gl_2\sin(\theta + \phi)$$

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\phi}} = I_B\ddot{\theta} + I_B\ddot{\phi} + m_2l_2^2\ddot{\theta} + m_2l_2^2\ddot{\phi} + m_2L_1l_2\ddot{\theta}\cos(\phi) - m_2L_1l_2\dot{\theta}\dot{\phi}\sin(\phi)$$

The differential equation describing the dynamics of the system is

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\vec{q}}} - \frac{\partial \mathcal{L}}{\partial q} = B(\dot{q}) + \tau(q)$$

where $q = \begin{bmatrix} \theta \\ \phi \end{bmatrix}$

A.2 Communication Structure

In Table A.1 the various commmands that are used in the command structure shown in Figure A.1 used for debugging purposes is explained with the possible value ranges that can be used.

0	1	2	3	4	...	n-1	n
$	cmd	,	value	value		\r	\n

Figure A.1: Data Structure for Sending Commands

Command	Range	Effect
A		
B		3.3
C		12

Table A.1: Summary of Communication Commands and their Effects

List of References

(1990). A case study in approximate linearization: The acrobat example. *Proc. American Control Conference.*

(1995). The swing up control problem for the acrobat. *IEEE Control Systems*, vol. 47, no. 10, pp. 50–55.

(2006). On stabilization of an inverted double pendulum with one control torque. *Journal of Compute and Systems Science International*, vol. 45, no. 3, pp. 337–334.

(2007 2). *5.0 A H-Bridge woth Load Current Feeback.* Freescale Semiconductor, 12th edn. Document Number: MC33887.

(2017). *How to get the best ADC accuracy in the STM32 microcontrollers.* ST Electronics.

Faulhaber (2011). *Encoders, Magnetic Encoder, Series IE2-16.* Faulhaber, 11th edn.

Gene F. Franklin, J. David Powell, e.a. (2015). *Feedback Control of Dynamic Systems.* 7th edn. Pearson Education Limited, Edingburgh Gate, Harlow, Essex CM20 2JE, England. ISBN 1-29-206890-6.

Instruments, N. (2006 9). *Magnetic Encoder Fundamentals.* 10. National Instruments, The address.