Bachelor's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Ivo Tredal
Bjørn K.T. Solheim
Ludvig Vartdal
Leidulv Tønnesland

Modeling, Development, and Design of an Inverted Pendulum System

Bachelor's thesis in Electrical Engineering
Supervisor: Christian Fredrik Sætre
May 2023

NTNU
Norwegian University of
Science and Technology

Ivo Tredal
Bjørn K.T. Solheim
Ludvig Vartdal
Leidulv Tønnesland

Modeling, Development, and Design of an Inverted Pendulum System

Bachelor's thesis in Electrical Engineering
Supervisor: Christian Fredrik Sætre
May 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Abstract

The inverted pendulum has made its mark as a staple in the field of control theory, making an appearance in classic mechanical systems such as the Furuta pendulum and the cart-pendulum system. This thesis focuses on the reaction wheel pendulum, more specifically the modeling, development and design of an inverted spherical pendulum actuated by two perpendicular reaction wheels. The system is developed with the overarching objective of functioning as an interactive learning platform for testing and experimenting with various control configurations.

The modeling of the system is centered around the derivation of an expression for the dynamics, which is done through the utilization of the Euler-Lagrange equations. The design process integrates a hardware and mechanical design approach, aiming to achieve optimum functionality while minimizing expenditure. Fabrication of the mechanical components entails 3D printing, and the hardware components are intricately connected onto a bespoke PCB. The implementation of the operational functionalities, e.g., Euler angle estimation and control system algorithms, is executed through the usage of the ESP32 family-based microcontroller. Additionally, wireless communication between the microcontroller and a Python-based user interface is established, making the product intuitive and user-friendly.

Keywords: Lagrangian Mechanics, Automatic Control, PCB, HMI, 3D Printing.

Sammendrag

Den inverterte pendelen er et stempel innenfor reguleringsteknikken, og er gjenkjennbar i klassiske mekaniske systemer som Furuta-pendelen og vogn-pendel-systemet. Denne oppgaven legger fokuset på reaksjonshjulpendelen - mer spesifikt modelleringen, utviklingen og designet av en invertert sfærisk pendel aktuert av to vinkelrette reaksjonshjul. Systemet er utviklet med det overordnede formålet om å fungere som en interaktiv læringsplattform som kan brukes til utprøving av diverse reguleringstekniske konsepter.

Ved modelleringen av systemet vil Euler-Lagrange-likningene brukes til å utlede et uttrykk for dynamikken. Designprosessen kombinerer maskinvare og mekanisk designtilnærming med formålet om å oppnå optimal funksjonalitet, samtidig som det vektlegges å minimere kostnadene. Mekaniske komponenter produseres ved hjelp av 3D-printing, og maskinvare er integrert i et spesialtilpasset kretskort. Algoritmene som definerer systemets funksjonalitet, deriblant estimeringen av Euler-vinkler, samt algoritmene knyttet til reguleringssystemet, implementeres på en mikrokontroller tilhørende ESP32-familien. Dessuten er det lagt opp til trådløs kommunikasjon med et eksternt Python-basert brukergrensesnitt, som gjør sluttproduktet intuitivt og brukervennlig.

Nøkkelord: Lagrange-mekanikk, Reguleringsteknikk, PCB, HMI, 3D-printing.

Preface

This thesis is written during the spring semester of 2023 as a compulsory part of the Bachelor of Science program in Electrical Engineering at the Norwegian University of Science and Technology (NTNU). The group consists of three members with the study specialization Automatic Control and Robotics, as well as one member specializing in Electronics and Sensor Systems. The multidisciplinary team structure has proven useful, as the project required solutions from both fields. The rather steep learning curve, in addtion to the challenge of applying our prior knowledge acquired throughout our studies, has been enriching. Overall, the project has been a very valuable and rewarding experience.

We would, first and foremost, like to thank Associate Professor Christian Fredrik Sætre at the Department of Engineering Cybernetics (ITK) for his guidance throughout the project. Our frequent meetings were always insightful and, indeed, very beneficial. We would also like to extend our gratitude to Senior Engineer Ingulf Helland at the Department of Electronic Systems (IES) for taking the time to provide advice and insight regarding our electronic circuit design. This was meaningful, as ordering a malfunctioning PCB would be quite detrimental to our progress.

Ivo Tredal, Bjørn K.T. Solheim, Ludvig Vartdal, Leidulv Tønnesland

Trondheim, May 2023

Contents

List of Figures **ix**

Acronyms **x**

1 Introduction **1**

1.1 Background and Motivation 1
1.2 Problem Description 1
1.3 Outline 2
1.4 Notation 2

2 Modeling **3**

2.1 Rigid Body Dynamics 3
2.2 Single-Axis Reaction Wheel Pendulum Dynamics 4
2.3 Dual-Axis Reaction Wheel Pendulum Dynamics 6
2.4 DC Motor Dynamics 9
2.5 Complete State-Space Representation of the Pendula 10
2.6 Linearization 11

3 Realization **13**

3.1 Hardware 13
3.2 Composite System Prerequisites 16
3.3 PCB Design and Implementation 17
3.4 Motor Control 20
3.5 Mechanical Design 22
3.6 Bluetooth Communication 25
3.7 HMI Design 25

4 Implementation **27**

4.1 Thermal Sensing 27
4.2 Bias Compensation and Noise Filtering 28
4.3 Inertial Sensor Fusion 29
4.4 Automatic Stabilization 31

5 Conclusions **34**
5.1 Summary . . . 34
5.2 Further Work . . . 34
5.3 Concluding Remarks . . . 36

References **37**

Appendix **40**
A PCB Specifications and Design . . . 40
A.1 Schematic . . . 40
A.2 Circuit Diagram . . . 41
B CAD-Model Schematics . . . 42
C Arduino Code Snippets . . . 44
C.1 Section of IMU variables structure . . . 44
C.2 Interrupt Routine for Motor 1 Encoder . . . 44
C.3 Get Motor RPM Function . . . 44
C.4 Bluetooth Transmission Function . . . 45
D Budget . . . 46

List of Figures

2.1 Inverted reaction wheel pendulum with one axis of rotation. 4
2.2 The reaction wheel pendulum modeled as a robotic manipulator. 5
2.3 Inverted spherical pendulum in three-dimensional Euclidean space. 7
2.4 Projection of the spherical reaction wheel pendulum. 8
2.5 DC motor model. 9

3.1 ESP32-WROOM-32D development board, reprinted from [8]. 14
3.2 The DC motors and motor drivers employed in the system. 14
3.3 MPU-6050 6-DoF accelerometer and gyroscope, reprinted from [14]. 14
3.4 10K NTC thermistor, reprinted from [16]. 15
3.5 Magnetic encoder, reprinted from [17]. 15
3.6 16
3.7 Buck converter, inspired by [24]. 17
3.8 H-bridge circuit, inspired by [26]. 18
3.9 3D model of the PCB. 20
3.10 Relation between duty cycle and RPM. 21
3.11 CAD models of main mounts. 22
3.12 CAD model of the reaction wheel. 23
3.13 Reaction wheel iterations. 23
3.14 CAD models of the components related to the single-axis system. 24
3.15 Photographs of the realized pendulum system. 24
3.16 Human Machine Interface 26

4.1 Result of ordinary least squares estimation. The data is fetched from [35]. . 28
4.2 Magnitude response of the digital filter $H(z)$. 29
4.3 Effect of smoothing the scaled accelerometer data. 30
4.4 Comparison of the Euler angle estimates. 31
4.5 Block diagram of a causal PID controller with anti-windup, inspired by [37]. 32
4.6 Attempted stabilization of the single-axis system. 33

Acronyms

BIBO Bounded Input, Bounded Output.

BMS Battery Management System.

CAD Computer-Aided Design.

DC Direct Current.

DoF Degrees of Freedom.

EMF Electromotive Force.

EMI Electromagnetic Interference.

FIR Finite Impulse Response.

GUI Graphical User Interface.

HMI Human Machine Interface.

IC Integrated Circuit.

IIR Infinite Impulse Response.

IMU Inertial Measurement Unit.

LTI Linear Time-Invariant.

MIMO Multiple Input, Multiple Output.

NTC Negative Temperature Coefficient.

ODE Ordinary Differential Equation.

PCB Printed Circuit Board.

PID Proportional-Integral-Derivative.

PLA Polyactic Acid.

PWM Pulse-Width Modulation.

RoHS Restrictions of Hazardous Substances.

RPM Revolutions Per Minute.

SISO Single Input, Single Output.

SMD Surface-Mount Device.

Chapter 1

Introduction

1.1 Background and Motivation

The pendulum has, ever since the first experiments of the pendulum conducted by Galileo Galilei took place, been researched extensively. It has made its mark as one of the most notable topics in physics throughout history, both by being a tool for measuring the acceleration due to gravity and by laying the foundation for innovative solutions such as the pendulum clock. Following the rise of control theory, the pendulum has proven itself useful and relevant, as many physical systems may be modeled as pendula. In humanoid robotics, for instance, bipedal locomotion may be modeled as two coupled pendula, whereof the stance leg corresponds to an inverted pendulum and the swing leg corresponds to a freely hanging pendulum.

The stabilization of inverted pendula is a frequently used example in research and education in the field of control theory, due to their nonlinear nature. Because of this, the inverted pendulum is an excellent system to be used as a benchmark for testing different control schemes. The inverted reaction wheel pendulum is an example of a mechanical system that admits underactuation, and is one of the simplest examples of this being the case. Underactuation is a property of increasing relevance following the rise of robotics, as both biped walkers and the majority of flying and swimming robots are examples of underactuated systems [1].

1.2 Problem Description

The goal of the project is to develop a mechatronic system that functions as a learning platform for experimenting with both basic and advanced concepts in the field of control engineering. The system is required to be relatively cheap, reproducible and safe, and it must be deemed viable for educational purposes, e.g., as laboratory work.

The system to be developed is a pendulum with two perpendicular reaction wheels attached to it, each driven by a DC motor. This encompasses the design of an electronic system that enables the ability to control actuators and read sensor data through the usage of a microcontroller, as well as the design and construction of the physical pendulum and the reaction wheels. Once these tasks are accomplished, one can implement a controller that automatically computes the DC motor input voltage.

In addition to developing the unconstrained reaction wheel pendulum, it is also desirable to develop a solution that limits the pendulum to only rotate about a single axis, allowing for stabilization using only a single reaction wheel. A mathematical model of the dynamics must be derived for both systems. This is due to the project goal of developing

a product that can be used for further experimentation, as a model is required for the implementation of several control configurations, e.g., model predictive control and the linear-quadratic regulator. Furthermore, analysing the mathematical models will allow for a deeper understanding of the systems, and may provide critical insight on how the control system performance may be enhanced.

1.3 Outline

Following the introductory chapter, models describing the dynamics of both the SISO and MIMO systems will be derived in Chapter 2. Here, a summary of the results from physics, mathematics and robotics that make up the prerequisites required to derive such expressions for rigid bodies will be provided as well. Next, the thesis will focus on the physical realization of the system in Chapter 3. This includes hardware and electronics, as well as the construction of the rigid mechanical system holding everything together. Additionally, the design of the HMI will be documented in this chapter. Subsequently, the algorithms that will be implemented on the microcontroller are described in Chapter 4. At last, the final remarks and the conclusion of the thesis is stated in Chapter 5.

1.4 Notation

Column vectors are denoted by bold lowercase letters, e.g., $\mathbf{x} \in \mathbb{R}^n$, while matrices are denoted by bold uppercase letters, e.g., $\mathbf{A} \in \mathbb{R}^{n \times m}$. The identity matrix is denoted by $\boldsymbol{\mathcal{I}}$, and its dimension is implicitly given by the context. $\hat{\mathbf{e}}_1$, $\hat{\mathbf{e}}_2$ and $\hat{\mathbf{e}}_3$ denotes the unit vectors in the x-, y- and z-directions, respectively, and again - the dimension is given by the context. $\mathbf{u} \otimes \mathbf{v}$ denotes the outer product of $\mathbf{u}$ and $\mathbf{v}$, and $\mathbf{u} \times \mathbf{v}$ denotes the cross product of $\mathbf{u}$ and $\mathbf{v}$. A random vector $\mathbf{z} \in \mathbb{R}^m$ is normally distributed with the expected value $\boldsymbol{\mu} \in \mathbb{R}^m$ and the covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{m \times m}$ if $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. $\mathrm{SO}(n)$ denotes the special orthogonal group in dimension n. $\mathrm{atan2}(y, x)$ is the 2-argument arctangent function. The derivative of a function $x(t)$ with respect to time t is denoted $\dot{x}(t) = \frac{\mathrm{d}x}{\mathrm{d}t}$. The Laplace domain representation of a continuous-time signal $x(t)$ is denoted $X(s) = \mathcal{L}\{x(t)\}$. The z-domain representation of a discrete-time signal $x[n]$ is denoted $Y(z) = \mathcal{Z}\{y[n]\}$.

Chapter 2

Modeling

This chapter documents the derivation of the kinematics and dynamics of both the aforementioned single-axis and dual-axis pendulum systems, resulting in nonlinear state-space models that ensure realistic simulations. These will finally be linearized, which will allow for experimentation using linear mathematical methods such as convolution and the Laplace transform.

2.1 Rigid Body Dynamics

A sufficiently accurate system model may be acquired by utilizing Lagrangian mechanics, where the Lagrangian is defined as the difference between the system's kinetic and potential energy,

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) \coloneqq \mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) - \mathcal{P}(\mathbf{q}). \tag{2.1}$$

The kinetic energy of a kinematic chain with the generalized coordinates $\mathbf{q} = \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix}^\top$, i.e., an n-link system with n corresponding joint variables, is given as

$$\mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \sum_{i=1}^{n} \left[m_i \mathbf{v}_i^\top \mathbf{v}_i + \boldsymbol{\omega}_i^\top \mathbf{R}_i^0 \boldsymbol{\mathcal{J}}_i (\mathbf{R}_i^0)^\top \boldsymbol{\omega}_i \right], \tag{2.2}$$

where m_i is the mass of body i, and $\mathbf{v}_i \in \mathbb{R}^3$ and $\boldsymbol{\omega}_i \in \mathbb{R}^3$ are the corresponding translational and angular velocity vectors. Additionally, $\mathrm{SO}(3) \ni \mathbf{R}_i^0 = \prod_{j=1}^{i} \mathbf{R}_j^{j-1}$ is the orientation transformation between the i-th body-attached frame and the inertial frame, and $\boldsymbol{\mathcal{J}}_i \in \mathbb{R}^{3\times3}$ is the inertia tensor of body i, expressed in the body-attached frame. While $\mathbf{v}_i$ is simply given by computing the time derivative of the distance from the origin in the inertial frame to the center of mass of the i-th body, $\mathbf{r}_i$, $\boldsymbol{\omega}_i$ may be determined by the corresponding skew-symmetric matrix $\mathbb{R}^{3\times3} \ni [\boldsymbol{\omega}]_\times = -[\boldsymbol{\omega}]_\times^\top$, defined as

$$\boldsymbol{\omega}_i = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \implies [\boldsymbol{\omega_i}]_\times \coloneqq \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} = \sum_{j=1}^{3} (\boldsymbol{\omega}_i \times \hat{\mathbf{e}}_j) \otimes \hat{\mathbf{e}}_j = \frac{\mathrm{d}\mathbf{R}_i^0}{\mathrm{d}t} (\mathbf{R}_i^0)^\top. \tag{2.3}$$

In the case of a continuous mass distribution characterized by the mass density function $\rho(\mathbf{r}_i^\star)$, the inertia tensor may be expressed as

$$\boldsymbol{\mathcal{J}}_i = \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{yx} & J_{yy} & J_{yz} \\ J_{zx} & J_{zy} & J_{zz} \end{bmatrix} = \iiint_{\mathcal{B}} \rho(\mathbf{r}_i^\star) [\mathbf{r}_i^\star]_\times^\top [\mathbf{r}_i^\star]_\times \, \mathrm{d}V, \tag{2.4}$$

where $\mathcal{B}$ is the bounded region in space containing the body and $\mathbf{r}_i^\star$ is the distance from the origin in the given body-attached frame to the center of mass of the body. The potential

energy of the system is simply given by gravity's effect on each link,

$$\mathcal{P}(\mathbf{q}) = -g \sum_{i=1}^{n} m_i \mathbf{r}_i^\top \hat{\mathbf{e}}_g. \tag{2.5}$$

Here, g is the gravitational acceleration, while $\hat{\mathbf{e}}_g$ is a unit vector in the direction of the force of gravity. Subsequently, the dynamics of the system is given by the Euler-Lagrange equations,

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{q}_k}\right] - \frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial q_k} = \tau_k, \qquad k = 1, \ldots, n, \tag{2.6}$$

where τ_k represents the generalized force corresponding to q_k. This yields a set of n coupled second order ODEs of the form

$$\sum_{j=1}^{n} d_{kj}\ddot{q}_j + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left[\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k}\right]\dot{q}_i\dot{q}_j + \frac{\partial \mathcal{P}}{\partial q_k} = \tau_k, \qquad k = 1, \ldots, n, \tag{2.7}$$

where d_{ij} denotes the elements in the inertia matrix $\mathbf{D}(\mathbf{q}) \in \mathbb{R}^{n\times n}$. This can be expressed as

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}, \tag{2.8}$$

where $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n\times n}$ is the Coriolis matrix and $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$ is the gravity vector [2].

2.2 Single-Axis Reaction Wheel Pendulum Dynamics

The inverted reaction wheel pendulum where the pendulum is constrained to only rotate about a single axis is depicted in Fig. 2.1. The generalized coordinates are

$$\mathbf{q} = \begin{bmatrix}\theta_p \\ \theta_w\end{bmatrix}, \tag{2.9}$$

and their meaning is shown in the figure.

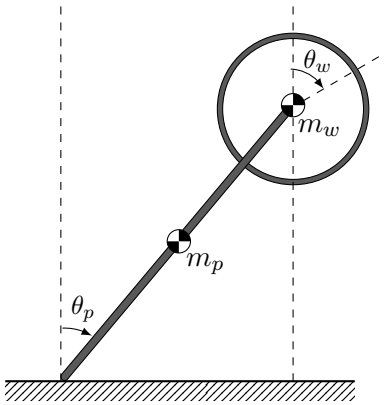


Figure 2.1. Inverted reaction wheel pendulum with one axis of rotation.

The system may be modeled as a 2-DoF robotic manipulator, as shown in Fig. 2.2. In the figure, the z_0-, z_1- and z_2-axes are not drawn, as they are implicitly given by the right-hand rule, i.e., $\hat{\mathbf{e}}_3 = \hat{\mathbf{e}}_1 \times \hat{\mathbf{e}}_2$. The orientation of the pendulum corresponds to a clockwise

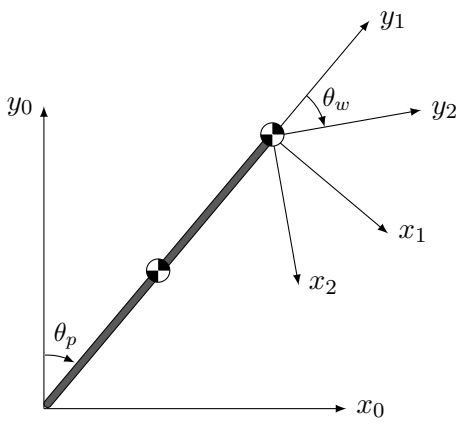


Figure 2.2. The reaction wheel pendulum modeled as a robotic manipulator.

θ_p-rotation about the z_0-axis, resulting in the transformation matrix that relates the initial frame and the body attached frame $o_1x_1y_1z_1$

$$\mathbf{R}_1^0 = \begin{bmatrix} \cos\theta_p & \sin\theta_p & 0 \\ -\sin\theta_p & \cos\theta_p & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.10a}$$

Similarly, the angle of the reaction wheel corresponds to a clockwise θ_w-rotation about the z_1-axis, thus giving the orientation of $o_2x_2y_2z_2$ relative to $o_1x_1y_1z_1$ as

$$\mathbf{R}_2^1 = \begin{bmatrix} \cos\theta_w & \sin\theta_w & 0 \\ -\sin\theta_w & \cos\theta_w & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.10b}$$

The position of any point along the pendulum a distance ℓ away from the pivot is

$$\mathbf{r} = \begin{bmatrix} 0 & \ell & 0 \end{bmatrix} \mathbf{R}_1^0 = \ell \begin{bmatrix} \sin\theta_p \\ \cos\theta_p \\ 0 \end{bmatrix}, \tag{2.11}$$

and therefore, the corresponding translational velocity vector is

$$\mathbf{v} = \frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t} = \ell\dot{\theta}_p \begin{bmatrix} \cos\theta_p \\ -\sin\theta_p \\ 0 \end{bmatrix}. \tag{2.12}$$

Hence, the translational kinetic energy of the system is given by

$$\mathcal{K}_{\text{trans}}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} m_p d_p^2 \dot{\theta}_p^2 + \frac{1}{2} m_w d_w^2 \dot{\theta}_p^2, \tag{2.13}$$

where m_p is the mass of the reaction wheel and m_w is the mass of the pendulum. Additionally, d_p and d_w are the distances from the pivot to the corresponding centers of mass. The rotational velocity of the pendulum, as well as the rotational velocity of the reaction wheel, is given by the skew-symmetric matrices

$$[\boldsymbol{\omega}_p]_\times = \frac{\mathrm{d}\mathbf{R}_1^0}{\mathrm{d}t} (\mathbf{R}_1^0)^\top = \begin{bmatrix} 0 & \dot{\theta}_p & 0 \\ -\dot{\theta}_p & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{2.14a}$$

$$[\boldsymbol{\omega}_w]_\times = \frac{\mathrm{d}\mathbf{R}_2^0}{\mathrm{d}t}(\mathbf{R}_2^0)^\top = \begin{bmatrix} 0 & \dot{\theta}_p + \dot{\theta}_w & 0 \\ -\dot{\theta}_p - \dot{\theta}_w & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{2.14b}$$

respectively. Due to the z_0- and z_1-axes being the only axes of rotation, the moment of inertia tensor of the pendulum, expressed in $o_1x_1y_1z_1$, is given by

$$\boldsymbol{\mathcal{J}}_p = \mathrm{diag}(0, 0, J_p), \tag{2.15a}$$

while the inertia tensor of the reaction wheel, expressed in $o_2x_2y_2z_2$, is simply

$$\boldsymbol{\mathcal{J}}_w = \mathrm{diag}(0, 0, J_w). \tag{2.15b}$$

The rotational kinetic energy of the system can now be expressed as

$$\mathcal{K}_{\mathrm{rot}}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2}J_p\dot{\theta}_p^2 + \frac{1}{2}J_w(\dot{\theta}_p + \dot{\theta}_w)^2. \tag{2.16}$$

Combining the expressions for the translational and rotational kinetic energy, the total kinetic energy of the system is given by

$$\mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2}\left(m_pJ_p^2 + m_wJ_w^2 + J_p + J_w\right)\dot{\theta}_p^2 + J_w\dot{\theta}_p\dot{\theta}_w + \frac{1}{2}J_w\dot{\theta}_w^2, \tag{2.17}$$

and, given that $\mathcal{P} = 0$ when $\theta_p = \pm\frac{\pi}{2}$, the potential energy is

$$\mathcal{P}(\mathbf{q}) = (m_pd_p + m_wd_w)\, g\cos\theta_p. \tag{2.18}$$

The Euler-Lagrange equations yield the system model

$$\begin{aligned} \gamma\ddot{\theta}_p + J_w\ddot{\theta}_w &= \nu\sin\theta_p, \\ J_w\ddot{\theta}_p + J_w\ddot{\theta}_w &= \tau_w, \end{aligned} \tag{2.19}$$

where $\gamma = m_pJ_p^2 + m_wJ_w^2 + J_p + J_w$, $\nu = g\,(m_pd_p + m_wd_w)$, τ_w is the torque applied on the reaction wheel, and it is assumed that the friction acting on the pendulum is negligible due to the pivot being fixed to the origin in the inertial frame. Based on this, the pendulum is in equilibrium when $\theta_p = \pi k\,\forall k \in \mathbb{Z}$, of which the equilibria in the upright position are unstable (i.e., not simply stable[1]) and the equilibria in the downward position are asymptotically stable (i.e., simultaneously simply stable and convergent[2]) [3].

2.3 Dual-Axis Reaction Wheel Pendulum Dynamics

Deriving the equations of motion when no constraints are placed on the pendulum is a far more troublesome task. In the single-axis case, both the inertia tensor of the pendulum and the reaction wheel only depended on the principal moments of inertia about the z_i-axes. This is not the case in three dimensions, where the cross products of inertia are present as well. Hence, applying a torque in a single direction may induce angular acceleration in multiple directions, as implied by the rotational form of Newton's second law,

$$\boldsymbol{\tau} = \boldsymbol{\mathcal{J}}\dot{\boldsymbol{\omega}}. \tag{2.20}$$

The derivation can, however, be simplified by making reasonable assumptions. Just like in the SISO case, it is assumed that the pivot is fixed to the origin in the world frame, which means that the friction acting on the pendulum is negligible. Additionally, it is assumed

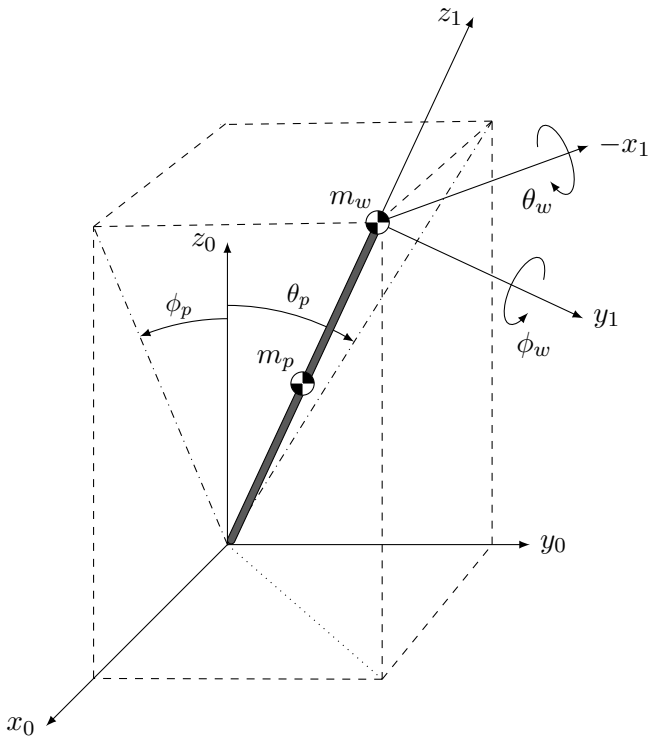


Figure 2.3. Inverted spherical pendulum in three-dimensional Euclidean space.

that the center of mass of each reaction wheel coincides at the same point in space, as shown in Fig. 2.3. In this case, the generalized coordinates are

$$\mathbf{q} = \begin{bmatrix} \theta_p \\ \theta_w \\ \phi_p \\ \phi_w \end{bmatrix}, \tag{2.21}$$

where θ_w and ϕ_w are the angles related to the angular velocity of the reaction wheels, and θ_p is the angle between the z_0-axis and the projection of the pendulum onto the x_0z_0-plane. Similarly, ϕ_p is the angle between the z_0-axis and the projection of the pendulum onto the y_0z_0-plane. This corresponds to a counterclockwise ϕ_p-rotaion about the fixed y_0-axis, followed by a clockwise θ_p-rotation about the fixed x_0-axis. The orientation transformation between the coordinate frames $o_1x_1y_1z_1$ and $o_0x_0y_0z_0$ is given by

$$\mathbf{R}_1^0 = \begin{bmatrix} \cos\phi_p & 0 & \sin\phi_p \\ -\sin\theta_p\sin\phi_p & \cos\theta_p & \sin\theta_p\cos\phi_p \\ -\cos\theta_p\sin\phi_p & -\sin\theta_p & \cos\theta_p\cos\phi_p \end{bmatrix}, \tag{2.22}$$

which implies that the position of the pendulum, expressed in the world frame, is

$$\mathbf{r} = \begin{bmatrix} 0 & 0 & \ell \end{bmatrix} \mathbf{R}_1^0 = \ell \begin{bmatrix} \sin\phi_p \\ \sin\theta_p\cos\phi_p \\ \cos\theta_p\cos\phi_p \end{bmatrix}, \tag{2.23}$$

[1] An equilibrium point $\mathbf{x}_{\text{eq}}$ is simply stable if $\forall\varepsilon > 0\,\exists\delta > 0$ such that if $\|\mathbf{x}(0) - \mathbf{x}_{\text{eq}}\| \leq \delta$ then $\|\mathbf{x}(t) - \mathbf{x}_{\text{eq}}\| \leq \varepsilon\,\forall t \geq 0$.

[2] An equilibrium point $\mathbf{x}_{\text{eq}}$ is convergent if $\exists\delta > 0$ such that if $\|\mathbf{x}(0) - \mathbf{x}_{\text{eq}}\| \leq \delta$ then $\mathbf{x}(t) \xrightarrow{t\to\infty} \mathbf{x}_{\text{eq}}$.

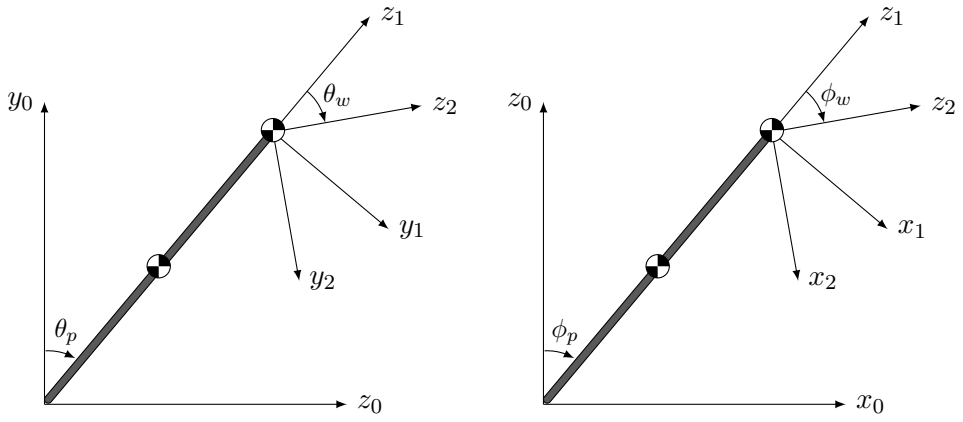
where ℓ is the distance from the pivot point to some point on the pendulum. Hence, the translational velocity of the given point in space is

$$\mathbf{v} = \frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t} = \ell \begin{bmatrix} \dot{\phi}_p \cos\phi_p \\ \dot{\theta}_p \cos\theta_p \cos\phi_p - \dot{\phi}_p \sin\theta_p \sin\phi_p \\ -\dot{\theta}_p \sin\theta_p \cos\phi_p - \dot{\phi}_p \cos\theta_p \sin\phi_p \end{bmatrix}. \tag{2.24}$$

Based on this, the translational kinetic energy is given as

$$\mathcal{K}_{\mathrm{trans}}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2}\left(m_p d_p^2 + m_w d_2^2\right)\left(\dot{\theta}_p^2 \cos^2\phi_p + \dot{\phi}_p^2\right), \tag{2.25}$$

where m_p and m_w are the masses of the pendulum and the reaction wheel, and d_m and d_w are the distances from the pivot to the corresponding centers of mass. It is presumed that the pendulum's orientation won't deviate too much from being upright as a result of the implementation of a controller, implying that $\theta_p \cong \phi_p \cong 0$ at all times. Consequently, it is reasonable to assume that x_0 is parallel with x_1 and y_0 is parallel with y_1 at all times. This assumption, as well as the fact that the reaction wheels are orthogonal, simplifies the problem by allowing for separate treatment of the axes, as depicted in Fig. 2.4a and 2.4b. This allows for a derivation of the expression for the rotational kinetic energy that is quite similar to the derivation in the single-axis case.



(a) Projection onto the y_0z_0-plane. (b) Projection onto the x_0z_0-plane.

Figure 2.4. Projection of the spherical reaction wheel pendulum.

Neglecting the coupling of the axes, the rotational kinetic energy is

$$\mathcal{K}_{\mathrm{rot}}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2}J_{p,x_1}\dot{\theta}_p^2 + \frac{1}{2}J_{p,y_1}\dot{\phi}_p^2 + \frac{1}{2}J_{w,x_2}(\dot{\theta}_p + \dot{\theta}_w)^2 + \frac{1}{2}J_{w,y_2}(\dot{\phi}_p + \dot{\phi}_w)^2, \tag{2.26}$$

where J_{p,x_1} and J_{p,y_1} are the moments of inertia of the pendulum about the x_1- and y_1-axes, while J_{w,x_2} and J_{w,y_2} are the moments of inertia of the reaction wheels about the x_2- and y_2-axes. For convenience, it is assumed that $J_{p,x_1} = J_{p,y_1} =: J_p$ and $J_{w,x_2} = J_{w,y_2} =: J_w$. Thus, the expression for the rotational kinetic energy of the system reduces to

$$\mathcal{K}_{\mathrm{rot}}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2}J_p(\dot{\theta}_p^2 + \dot{\phi}_p^2) + \frac{1}{2}J_w\left[(\dot{\theta}_p + \dot{\theta}_w)^2 + (\dot{\phi}_p + \dot{\phi}_w)^2\right]. \tag{2.27}$$

The potential energy is given by

$$\mathcal{P}(\mathbf{q}) = (m_p d_p + m_w d_w)\, g \cos\theta_p \cos\phi_p, \tag{2.28}$$

given that $\mathcal{P} = 0$ when $\theta_p = \phi_p = \pm\frac{\pi}{2}$. Finally, the Euler-Lagrange equations yield the system of ODEs

$$\begin{aligned}(J_p + J_w + \mu\cos^2\phi_p)\ddot{\theta}_p + J_w\ddot{\theta}_w &= \eta\cos\phi_p\sin\theta_p + \dot{\theta}_p\dot{\phi}_p\sin 2\phi_p,\\ (J_p + J_w + \mu)\ddot{\phi}_p + J_w\ddot{\phi}_w &= \eta\cos\theta_p\sin\phi_p - \frac{1}{2}\mu\dot{\theta}_p^2\sin 2\phi_p,\\ J_w\ddot{\theta}_p + J_w\ddot{\theta}_w &= \tau_\theta,\\ J_w\ddot{\phi}_p + J_w\ddot{\phi}_w &= \tau_\phi,\end{aligned} \tag{2.29}$$

where $\mu = m_p d_p^2 + m_w d_w^2$, $\eta = g(m_p d_p + d_w m_w)$, τ_θ is the torque associated with θ_w, and τ_ϕ is the torque associated with ϕ_w. From this, one can show that the set of asymptotically stable equilibria is given by

$$\mathfrak{S} = \{(\theta_p, \theta_w) \in \mathbb{R}\times\mathbb{R} \,:\, \theta_p = \pi k_1 \wedge \theta_w = \pi k_2,\ k_1, k_2 \in \mathbb{Z}\}\,, \tag{2.30a}$$

while the set of unstable equilibria is given by

$$\mathfrak{U} = \left\{(\theta_p, \theta_w) \in \mathbb{R}\times\mathbb{R} \,:\, \theta_p = \pi k_1 - \frac{\pi}{2} \wedge \theta_w = \pi k_2 - \frac{\pi}{2},\ k_1, k_2 \in \mathbb{Z}\right\}. \tag{2.30b}$$

2.4 DC Motor Dynamics

Considering the dynamics of the DC motors is an important step towards increasing the accuracy of the model. A model representing the conversion of electrical to mechanical energy is shown in Fig. 2.5.

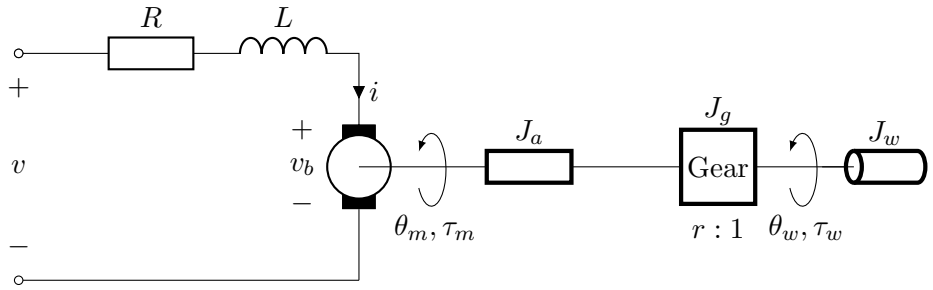


Figure 2.5. DC motor model.

The dynamics of the electric circuit is given by

$$\frac{\mathrm{d}i}{\mathrm{d}t} = \frac{1}{L}v(t) - \frac{R}{L}i(t) - \frac{1}{L}v_b(t), \tag{2.31}$$

where L is the armature inductance, R is the armature resistance, v is the armature voltage, i is the armature current, and the induced back EMF $v_b(t) = K_b\dot{\theta}_m(t)$. Additionally, the equation of motion for the mechanical system is

$$\frac{\mathrm{d}^2\theta_m}{\mathrm{d}t^2} + \frac{B_m}{J_m}\frac{\mathrm{d}\theta_m}{\mathrm{d}t} = \frac{1}{J_m}\tau_m(t) - \frac{1}{J_m r}\tau_w(t), \tag{2.32}$$

where $\theta_m = \frac{1}{r}\theta_w$ is the rotor position, τ_w is the load torque, B_m is the friction coefficient, J_m is the sum of the actuator and gear inertias, i.e., $J_m = J_a + J_g$, and the generated

torque $\tau_m(t) = K_m i_a(t)$. Hence, the system can be characterized by the transfer function from the armature voltage to the rotor position,

$$\frac{\Theta_m(s)}{V(s)} = \frac{K_m}{s\left[(Ls+R)(J_m s + B_m) + K_b K_m\right]}, \tag{2.33}$$

as well as the transfer function from the torque applied on the reaction wheel to the rotor position,

$$\frac{\Theta_m(s)}{T_w(s)} = \frac{-(Ls+R)/r}{s\left[(Ls+R)(J_m s + B_m) + K_b K_m\right]}, \tag{2.34}$$

where $\Theta_m(s) = \mathcal{L}\{\theta_m(t)\}$, $V(s) = \mathcal{L}\{v(t)\}$ and $T_w(s) = \mathcal{L}\{\tau_w(t)\}$. Furthermore, it is reasonable to assume that the "electrical time constant" $\frac{L}{R}$ is much smaller than the "mechanical time constant" $\frac{J_m}{B_m}$ [2]. Considering this, one may approximate $\frac{L}{R} \equiv 0$, and hence, the DC motor model reduces to

$$J_m \ddot{\theta}_w(t) + B\dot{\theta}_w(t) = \kappa v(t) - \tau_w(t), \tag{2.35}$$

where $\kappa = r\frac{K_m}{R}$ and $B = B_m + \frac{K_b K_m}{R}$ is the effective damping [4]. In the MIMO system, where two DC motors are used, one may consider

$$\begin{aligned} J_m \ddot{\theta}_w(t) + B\dot{\theta}_w(t) &= \kappa v_\theta(t) - \tau_\theta(t) \\ J_m \ddot{\phi}_w(t) + B\dot{\phi}_w(t) &= \kappa v_\phi(t) - \tau_\phi(t) \end{aligned} \tag{2.36}$$

as a description of the DC motor dynamics. Here, v_θ and v_ϕ are the armature voltages associated with $\phi_w = r\phi_m$ and $\theta_w = r\theta_m$. This, of course, assumes that both of the motors have identical parameters.

2.5 Complete State-Space Representation of the Pendula

Combining the inverted pendulum dynamics and the DC motor dynamics results in a model where the armature voltage is the input. In the case of the pendulum rotating about a single axis, this is done by combining Eq. (2.19) and Eq. (2.35). This can be expressed in state-space form by defining the input $u \coloneqq v$, the output $y \coloneqq \theta_p$ and the state vector $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^\top \coloneqq \begin{bmatrix} \theta_p & \dot{\theta}_p & \dot{\theta}_w \end{bmatrix}^\top$:

$$\begin{aligned} \dot{\mathbf{x}} &= \begin{bmatrix} x_2 \\ (J_w + J_m)\varepsilon\nu \sin x_1 + BJ_w\varepsilon x_3 \\ -J_w\varepsilon\nu \sin x_1 - B\varepsilon\gamma x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ -J_w\varepsilon\kappa \\ \varepsilon\gamma\kappa \end{bmatrix} u, \\ y &= x_1, \end{aligned} \tag{2.37}$$

where $\varepsilon = \left[(m_p J_p^2 + m_w J_w^2 + J_p + J_w)(J_w + J_m) - J_w^2\right]^{-1}$. For the dual-axis system, on the other hand, the equations to combine are Eq. (2.29) and Eq. (2.36). In this case, the input vector is defined as $\mathbf{u} = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^\top \coloneqq \begin{bmatrix} v_\theta & v_\phi \end{bmatrix}^\top$, the state vector is defined as $\mathbf{x} = \begin{bmatrix} x_1 & \cdots & x_6 \end{bmatrix}^\top \coloneqq \begin{bmatrix} \theta_p & \dot{\theta}_p & \dot{\theta}_w & \phi_p & \dot{\phi}_p & \dot{\phi}_w \end{bmatrix}^\top$, and lastly; the output vector is defined as $\mathbf{y} = \begin{bmatrix} y_1 & y_2 \end{bmatrix}^\top \coloneqq \begin{bmatrix} \theta_p & \phi_p \end{bmatrix}^\top$. Subsequently, the state-space representation of

the system is given as

$$\begin{aligned}\dot{\mathbf{x}} &= \begin{bmatrix} x_2 \\ \frac{1}{\upsilon+\lambda\mu\cos^2 x_5}\left[\lambda(x_2 x_5 \sin 2x_4 + \eta\cos x_4 \sin x_1) + J_w(Bx_3 - \kappa u_1)\right] \\ \frac{1}{\upsilon+\lambda\mu\cos^2 x_5}\left[(\kappa u_1 - Bx_3)(\sigma + \mu\cos^2 x_4) - J_w(\eta\cos x_4 \sin x_1 + x_2 x_5 \sin 2x_4)\right] \\ x_5 \\ \frac{1}{\upsilon+\lambda\mu}\left[\lambda\left(\eta\cos x_1 \sin x_4 - \frac{1}{2}\mu x_2^2 \sin 2x_4\right) + J_w(Bx_6 - \kappa u_2)\right] \\ \frac{1}{\upsilon+\lambda\mu}\left[J_w\left(\frac{1}{2}\mu x_2^2 \sin 2x_4 - \eta\cos x_1 \sin x_4\right) + (\sigma+\mu)(\kappa u_2 - Bx_6)\right] \end{bmatrix}, \\ \mathbf{y} &= \begin{bmatrix} x_1 \\ x_4 \end{bmatrix},\end{aligned} \tag{2.38}$$

where $\upsilon = J_m J_p + J_m J_w + J_p J_w$, $\lambda = J_m + J_w$ and $\sigma = J_p + J_w$.

2.6 Linearization

The general nonlinear dynamical system of the form

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} &= \mathbf{g}(\mathbf{x})\end{aligned} \tag{2.39}$$

may be linearized about an equilibrium point $(\mathbf{x}_{\text{eq}}, \mathbf{u}_{\text{eq}})$ where $\mathbf{f}(\mathbf{x}_{\text{eq}}, \mathbf{u}_{\text{eq}}) \equiv \mathbf{0}$:

$$\begin{aligned}\Delta\dot{\mathbf{x}} &= \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\substack{\mathbf{x}=\mathbf{x}_{\text{eq}} \\ \mathbf{u}=\mathbf{u}_{\text{eq}}}} \Delta\mathbf{x} + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\substack{\mathbf{x}=\mathbf{x}_{\text{eq}} \\ \mathbf{u}=\mathbf{u}_{\text{eq}}}} \Delta\mathbf{u}, \\ \Delta\mathbf{y} &= \left.\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right|_{\substack{\mathbf{x}=\mathbf{x}_{\text{eq}} \\ \mathbf{u}=\mathbf{u}_{\text{eq}}}} \Delta\mathbf{x},\end{aligned} \tag{2.40}$$

where $\Delta\mathbf{x} \cong \mathbf{x} - \mathbf{x}_{\text{eq}} =: \widetilde{\mathbf{x}}$, $\Delta\mathbf{u} \cong \mathbf{u} - \mathbf{u}_{\text{eq}} =: \widetilde{\mathbf{u}}$, and $\Delta\mathbf{y} \cong \mathbf{y} - \mathbf{C}\mathbf{x}_{\text{eq}} =: \widetilde{\mathbf{y}}$. This results in the approximated LTI state-space system with n states, m inputs and p outputs,

$$\begin{aligned}\dot{\widetilde{\mathbf{x}}} &= \mathbf{A}\widetilde{\mathbf{x}} + \mathbf{B}\widetilde{\mathbf{u}}, \\ \widetilde{\mathbf{y}} &= \mathbf{C}\widetilde{\mathbf{x}},\end{aligned} \tag{2.41}$$

where the state matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, the input matrix $\mathbf{B} \in \mathbb{R}^{n\times m}$, and the output matrix $\mathbf{C} \in \mathbb{R}^{p\times n}$ are defined by the corresponding Jacobian matrices. This may be a good approximation, given that the curvature of $\mathbf{f}$ is not too big and that only a sufficiently close proximity of the linearization point is considered [5]. Linearizing the SISO system about its upright equilibrium yields

$$\begin{aligned}\dot{\widetilde{\mathbf{x}}} &= \begin{bmatrix} 0 & 1 & 0 \\ \varepsilon\nu(J_w + J_m) & 0 & BJ_w\varepsilon \\ -J_w\varepsilon\nu & 0 & -B\varepsilon\gamma \end{bmatrix}\widetilde{\mathbf{x}} + \begin{bmatrix} 0 \\ -J_w\varepsilon\kappa \\ \varepsilon\gamma\kappa \end{bmatrix}\widetilde{u}, \\ \widetilde{y} &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}\widetilde{\mathbf{x}},\end{aligned} \tag{2.42}$$

while linearizing the MIMO system about its upright equilibrium results in the approximated LTI system given as

$$\begin{aligned}\dot{\widetilde{\mathbf{x}}} &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ a_3 & 0 & a_4 & 0 & 0 & 0 \\ a_2 & 0 & a_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & a_3 & 0 & a_4 \\ 0 & 0 & 0 & a_2 & 0 & a_1 \end{bmatrix} \widetilde{\mathbf{x}} + \begin{bmatrix} 0 & 0 \\ b_1 & 0 \\ b_2 & 0 \\ 0 & 0 \\ 0 & b_1 \\ 0 & b_2 \end{bmatrix} \widetilde{\mathbf{u}}, \\ \widetilde{\mathbf{y}} &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \widetilde{\mathbf{x}},\end{aligned} \tag{2.43}$$

with:

$$\begin{aligned} a_1 &= \frac{-B(\sigma+\mu)}{\upsilon+\lambda\mu}, \\ a_2 &= \frac{-J_w\eta}{\upsilon+\lambda\mu}, \\ a_3 &= \frac{\lambda\eta}{\upsilon+\lambda\mu}, \\ a_4 &= \frac{BJ_w}{\upsilon+\lambda\mu}, \\ b_1 &= \frac{-J_w\kappa}{\upsilon+\lambda\mu}, \\ b_2 &= \frac{\kappa(\sigma+\mu)}{\upsilon+\lambda\mu}. \end{aligned}$$

The approximated LTI models may be expressed as transfer matrices,

$$\mathbf{H}(s) = \mathbf{C}\frac{\operatorname{adj}(s\boldsymbol{\mathcal{I}} - \mathbf{A})}{\det(s\boldsymbol{\mathcal{I}} - \mathbf{A})}\mathbf{B}, \tag{2.44}$$

thus the SISO system corresponds to the transfer function

$$H(s) = \frac{-J_w\varepsilon\kappa s}{s^3 + B\gamma\varepsilon s^2 - \varepsilon\nu\left(J_m + J_w\right)s - B\varepsilon^2\nu\left[\gamma\left(J_m + J_w\right) - J_w^2\right]}, \tag{2.45}$$

and similarly, the MIMO system corresponds to the transfer matrix

$$\mathbf{H}(s) = \begin{bmatrix} \frac{b_1 s + a_4 b_2 - a_1 b_1}{s^3 - a_1 s^2 - a_3 s + a_1 a_3 - a_2 a_4} & 0 \\ 0 & \frac{b_1 s + a_4 b_2 - a_1 b_1}{s^3 - a_1 s^2 - a_3 s + a_1 a_3 - a_2 a_4} \end{bmatrix}. \tag{2.46}$$

Chapter 3

Realization

The purpose of this chapter is to develop the physical system, based on the models that were derived in Chapter 2. This includes the construction of the mechanical system, as well as the design of the electronic system and the graphical interface that enables user-friendly experimentation.

3.1 Hardware

Physically realizing an adequate development platform requires the use of electronics. An electronic system that achieves a viable product could in simple terms consist of a microcontroller, motors with encoders, motor drivers, batteries, thermistors and an IMU. When selecting these main components, it is crucial to consider physical limitations such as weight and size, but also safety measures. Hence, all of the electronic components used follow the RoHS standard. The components were carefully selected with a budgetary constraint of approximately 3000 NOK from retailers with reasonable shipping times. The full list of purchases is shown in Appendix D.

Microcontroller

The ESP32 series System-on-chip microcontroller by Espressif was selected as the device family of choice to control and perform all necessary computations of the system. Its low power consumption, small size, and low cost make it ideal for this application [6]. The specifically selected ESP32-WROOM-32D, shown in Fig. 3.1, is equipped with an Xtensa dual-core processor running at 240 MHz and is capable of Wi-Fi and Bluetooth communication. It can handle real-time control tasks and communicate with multiple sensors and actuators in a fast and reliable manner. This is a vital property when stabilizing the inverted pendulum. Its ease of use and support with the open-source software Arduino IDE was a fundamental requirement for the project [7].

DC Motors and Motor Drivers

The selection of DC motors for the platform had certain requirements. With the purpose of increasing the accuracy of the model, it was seen as important that the motors were of the common brushed standard. They also needed to be fitted with encoders for positional readings and have relatively high torque in relation to low weight and cost. Out of the few possible options, the 290-006 motor by Sha Yang Ye Industrial, shown in Fig. 3.2a, was seen as most suitable with a rated torque of 0.3 kgf.cm. With a 12 V rating and 2.4 A stall current, the selection of this motor impacted design choices related to both system

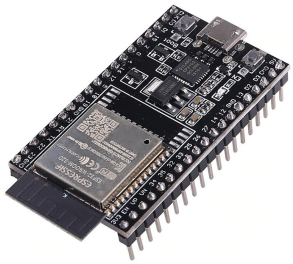
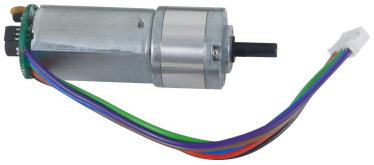


Figure 3.1. ESP32-WROOM-32D development board, reprinted from [8].



(a) 290-006 DC motor, reprinted from [11].

(b) MC33886PVW motor driver, adapted from [12].

Figure 3.2. The DC motors and motor drivers employed in the system.

power delivery and the required motor drivers. Since the motors can change direction, it is predicated that a current spike of 4.8 A is possible per motor with rapid change [9]. To make sure the system can handle this, the MC33886PVW H-bridge motor driver by NXP Semiconductors, shown in Fig. 3.2b, was selected for its 5 A load rating [10].

IMU

With the purpose of attaining the requisite levels of precision and accuracy in accelerometer and gyroscope measurements, the selection of the MPU6050, shown in Fig. 3.3, was seen as suitable for employment in the system. The device is based upon Micro-Electro-Mechanical-System technology, and integrates a 3-axis accelerometer and a 3-axis gyroscope, with the capacity of measuring an object's acceleration and angular velocity at a resolution of 16 bits. To obtain the accelerometer and gyroscope data in units of m/s^2 and rad/s, respectively, a scaling factor based on the sensor sensitivity is required. The default sensitivity settings for the accelerometer and gyroscope are $2g$ and 250 deg/s, respectively, which is appropriate for the intended application [13].

Figure 3.3. MPU-6050 6-DoF accelerometer and gyroscope, reprinted from [14].

Figure 3.4. 10K NTC thermistor, reprinted from [16].

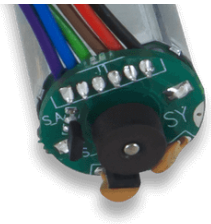


Figure 3.5. Magnetic encoder, reprinted from [17].

NTC Thermistors

In the context of a voltage-powered multisensor system, it is pertinent to consider the potential risks of damage to the electrical components. With the specific aim of avoiding heat damage, it is necessary to obtain a high degree of sensitivity and reliability in temperature measurements. Multiple 10 kΩ NTC thermistors, shown in Fig. 3.4, have thus been employed to monitor key regions of the system. The sensors exhibit a reduction in resistance as the temperature increases, making them ideal for temperature sensing and control. The areas that have been identified as critical regions in need of thermal sensing are the motor drivers, motors and batteries, being the components that undergo frequent and rapid rotational adjustments. If the temperature readings from the motors exceed 60 °C, it is advised to shut the system off [15]. The thermistor of choice is displayed in Fig. 3.4.

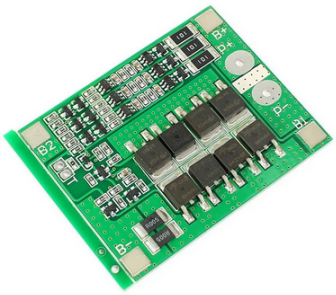
Magnetic Encoders

The accurate measurement of the position and rotation of the motors is achieved by the use of magnetic encoders, shown in Fig. 3.5, which are mounted on the rear end of the chosen motors. The magnet disc attached to the rear axle generates changes in the magnetic field, which is sensed by a pair of Hall effect sensors positioned 90 degrees apart. The selected motor's magnet discs are divided into six distinct sections of magnetic poles consisting of three north poles and three south poles. The Hall effect sensor outputs a logic 1 for a north pole and a logic 0 for a south pole. Given the possibility of mechanical vibrations and shocks, the implementation of a magnetic encoder was considered the most suitable option due to its capability to prevent disturbances. This is relevant, as the pendulum requires fast and frequent corrections, thereby increasing the probability of such disturbances occurring [18].

Batteries and BMS

To ensure powerful, yet portable power delivery to the system, it was seen as suitable to incorporate the use of 18650 lithium-ion batteries, shown in Fig. 3.6a. These batteries are rechargeable, and the selected EFEST IMR 18650 cells were chosen in particular for their high capacity of 3000 mAh and maximum pulse discharge rate of 35 A [19]. Lithium-ion

(a) 18650 lithium-ion battery, reprinted from [22].



(b) BMS, reprinted from [23].

Figure 3.6

batteries have a nominal voltage of 3.7 V with a typically safe working range between 2-3 V to 4.2 V. The minimum voltage is generally model specific and varies between vendors [20]. With three of these batteries in series, a typical voltage range between 11.1-12.6 V is achievable and within acceptable tolerance of what the intended system would require regarding the motor specification of 12 V $\pm 10\%$ [15]. The motors are most critical in regard to power delivery, and these batteries were chosen to mitigate any problems related to high current spikes when changing direction, but also due to the availability of high discharge rate batteries [21]. To safely use these batteries, a 3-cell BMS circuit, shown in Fig. 3.6b, was acquired with essential features such as cell balancing, overcharge-, over-discharge, and short circuit protection. The BMS is rated for a max pulse discharge rate of 35-40 A and a continuous rate up to 20 A, which approximately matches the tolerances of the batteries.

3.2 Composite System Prerequisites

With a general overview of the principal hardware components, as well as considerations for features, availability, shipping time and cost, it was seen as a more viable option to incorporate everything into a custom PCB design rather than relying on multiple off-the-shelf board solutions. By doing this, the control system could be more compact and tailored towards the specific use case. To better understand parts of the system's functionality, some general concepts related to its components are necessary to cover.

A fundamental buck converter/step-down regulator, as illustrated in Fig. 3.7, is based on the concept of rapid transistor switching. When the transistor is closed, the inductor and capacitor are charged, therefore no current flows through the diode D_1 due to reverse bias. At the opening of the transistor, the stored current in the inductor decreases along with the magnetic field and induces a voltage with opposite polarity, acting as a current source. If the voltage across the inductor is higher than the capacitor voltage, the capacitor gets charged. If both voltages are equal, they discharge through the load R_L and back through D_1. In other words, the capacitor helps to smoothen V_{OUT}. As a result of this switching functionality, the total current received by R_L becomes greater than the current received by V_{IN}. Therefore, the buck converter steps up the current while stepping down the voltage. To control the transistor, an external circuit that compares a determined reference voltage to a divided voltage of the output is used. This helps to decrease noise pollution, and changes the frequency of a pulse-width modulator connected to the transistor [24].

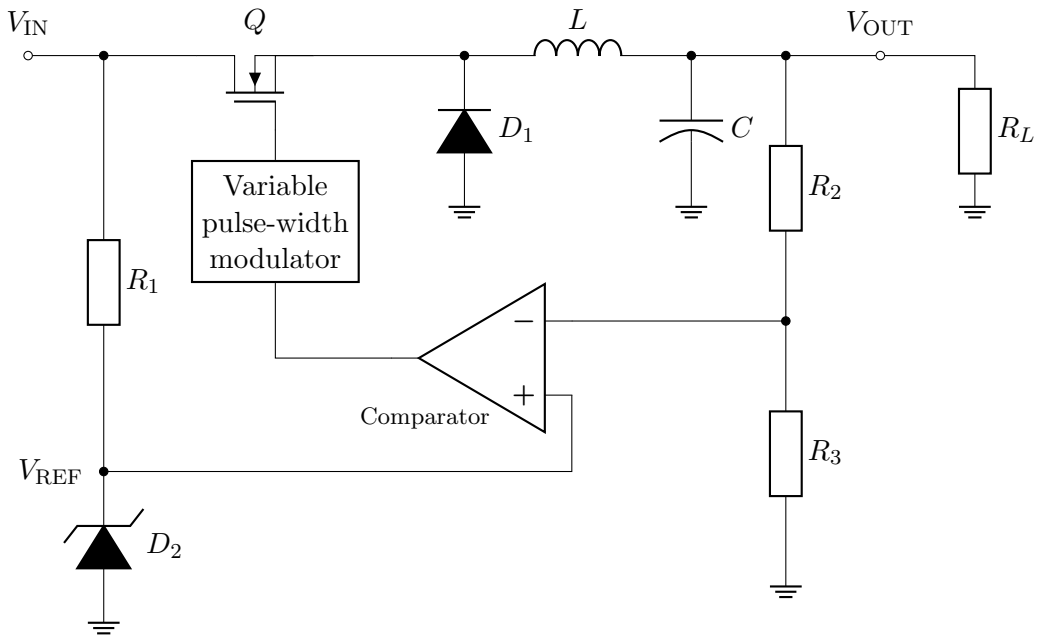


Figure 3.7. Buck converter, inspired by [24].

The H-Bridge circuit in Fig. 3.8 consists of four transistors that can be configured to have the voltage polarity changed of the central load connected between them. This type of circuit is typically used for controlling both the direction and speed of DC motors. Direction control is done by either closing Q_1 and Q_4 together or by closing Q_3 and Q_2 together. This results in opposite current paths through the motor. The closed/open states of the transistors can also be used to either coast, break, or short-circuit the motor. Speed control of the motor can be done through the usage of PWM signaling to specific transistors with certain on/off timings. These timing intervals are known as duty cycle and are well suited for inertial loads like motors, as they are prone to be less susceptible to discrete transistor switching [25]. To deal with the effect of slightly inconsistent transistor opening/closing timings that results in voltage build-up, each transistor is fitted with return path diodes to prevent the gates from being damaged [26].

3.3 PCB Design and Implementation

The PCB design was created using Altium Designer, manufactured by JLCPCB, and soldered by Elektronikk og prototypelaboratoriet at NTNU. The complete schematic and circuit design can be found in Appendix A.1 and A.2, respectively. The PCB incorporates all aforementioned hardware components, in addition to a buck converter circuit, logic-level converters, a multiplexer, an external battery charging circuit and other components related to noise filtering. Most components selected are of the SMD type which helps reduce the overall board size, since SMD components usually are smaller, but also reduces cost due to the need for fewer holes in the PCB [27]. To improve the potential for debugging and component replacements, the microcontroller, IMU and motors are mounted using sockets, and there are screw terminals for connecting each thermistor and the 12 V input. An on/off switch is also incorporated into the design to make general use of the system more practical, along with four M3-sized mounting holes for attachment to the mechanical system. The circuit board itself is a two-layer design, where the top layer consists of the

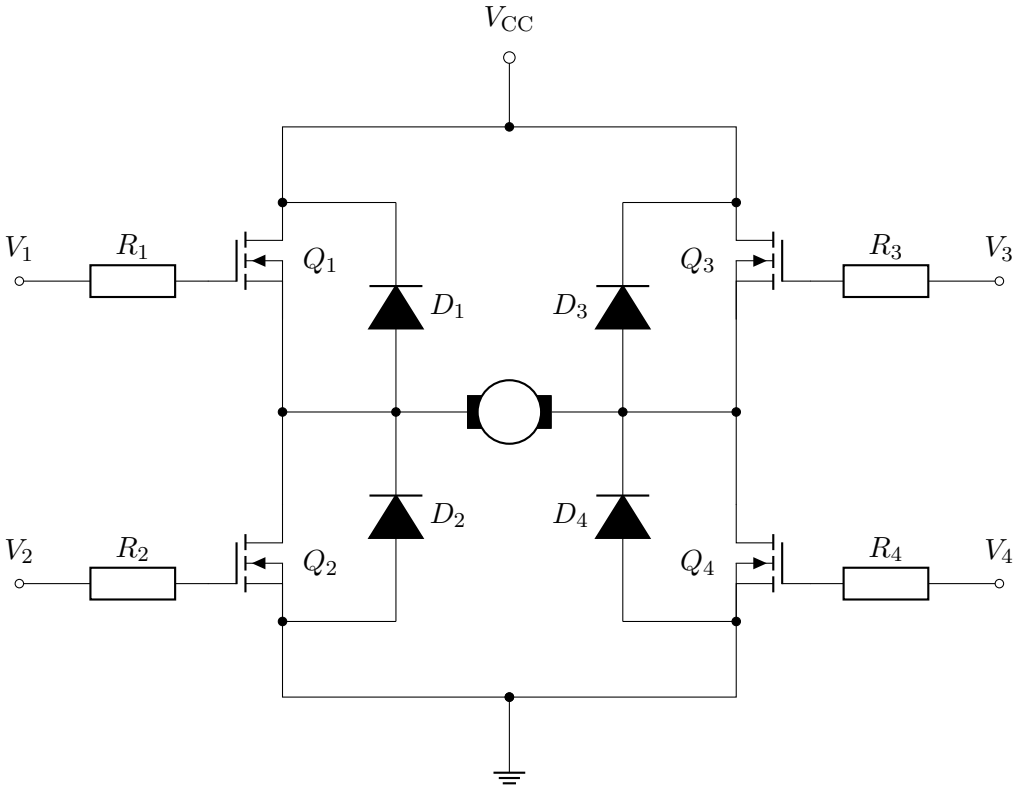


Figure 3.8. H-bridge circuit, inspired by [26].

trace routings, while the bottom layer is primarily a ground plane. Traces are routed in both planes using vias with an emphasis on maximizing width wherever possible. The ground plane is split into power ground (PGND), analog ground (AGND), and digital ground (DGND) sections with specific paths between them to reduce potential noise and interference [28].

As mentioned previously in the batteries and BMS subsection the input voltage for the system is intended to be 12 V ± 10%. This was done to match the required voltage of the motors since they constitute the largest load of the system. It was easier from a technical standpoint to step-down voltage for the system's other low-power components, rather than stepping up the voltage to the motors, while still being able to output a high enough current, due to the nature of such regulators [24]. For this reason, the PCB houses a buck-converter circuit to power the microcontroller, logic-level converters and motor encoders with 5 V. The microcontroller's internal regulator is also used and supplies 3.3 V to the logic-level converters, IMU and the thermistors pull-up resistors.

The onboard buck converter circuit uses the LMR51420 IC by Texas Instruments and converts the system's input voltage to a 5 V output. The IC model was selected for its suitable characteristics and TI's renowned reputation in addition to datasheet quality. The implementation of the IC carefully follows TI's recommendations and resembles, from a schematic view, figure 9-1 in the datasheet [29]. The component values in the implemented circuit are similar, but with some discrepancies to better align with the datasheet suggestions of section 9.2.2. Two input 4.7 µF ceramic decoupling capacitors of type X7R dielectric with a voltage rating of 50 V, to compensate for the derating of ceramic capacitors, were for instance selected along with an 0.1 µF ceramic decoupling capacitor

for high-frequency filtering. Both of these were placed as close ut the input IC terminal as possible to minimize voltage fluctuations [30]. To reduce output ripple voltage to a desirable target of around 30 mV, two 22 µF X7R dielectric capacitors were used. The minimum required capacitance was estimated using a $\Delta i_L/I_{\text{OUT}}$ ratio of 30% and a 5% overshoot/undershoot target in relation to the desired 5V output. The minimum C_{OUT} was calculated with the formula

$$C_{\text{OUT}} > \frac{1}{2} \cdot \frac{8 \cdot (I_{\text{OH}} - I_{\text{OL}})}{f_{\text{SW}} \cdot \Delta V_{\text{OUT\_SHOOT}}}, \tag{3.1}$$

where I_{OL} and I_{OH} are the minimum and maximum current output values for transient loads, respectively. With a f_{SW} of 500 kHz, and datasheet suggested values of 0.5 A and 1.5 A for I_{OL} and I_{OH}, a minimum C_{OUT} was estimated to be 32 µF. However, to leave some margin, a total value of 44 µF was instead used [29]. The inductor value was in a similar fashion calculated using the datasheet provided formula

$$L_{\text{MIN}} = \frac{V_{\text{IN\_MAX}} - V_{\text{OUT}}}{I_{\text{OUT}} \cdot K_{\text{IND}}} \cdot \frac{V_{\text{OUT}}}{V_{\text{IN\_MAX}} \cdot f_{\text{SW}}}. \tag{3.2}$$

Here, K_{IND} is the inductor ripple current in relation to the maximum output current, and was selected to a fitting value of 0.3. The minimum required inductance was calculated to be 10 µH, and an inductor of this kind was selected with both shielding, for EMI reduction, and a current rating of 4 A RMS. The same method was also applied to the selection of the C_{BOOT} capacitor which was from the datasheet suggested to have a voltage rating of at least 16 V and a dielectric temperature range of either X5R or X7R. It is worth mentioning that during the selection of these components, an effort was made to use components from well-known manufacturers with certain quality standards. The AEC-Q200 automotive standard is a well-known example and most chosen capacitors and resistors, from companies such as Panasonic or TDK, were selected with this qualification in mind [31, 32].

To implement the two selected motor drivers into the PCB design a 47 µF electrolytic decoupling capacitor connected to the V+ pin and a 33 nF ceramic decoupling capacitor connected to the internal charge pump pin, were used for each driver. This was done following the suggested typical application from the driver datasheet [10]. In addition to this, 0.1 µF decoupling capacitors were added to each of the driver's V+ pins for high-frequency filtering. All of these capacitors were placed as reasonably close to the driver pins as possible to minimize voltage fluctuations [30]. Furthermore in order to minimize EMI, as a result of sparking between the brushes and commutator in the DC-motors, a 0.1 µF ceramic capacitor was placed in parallel to each of the driver's outputs [33] To interface the drivers each is equipped with four logical low/high input pins. By applying different combinations of low/high to these pins, as shown in the datasheet, different operation modes can be selected. However, the drivers require a logic level of 5 V, while the microcontroller is only capable of 3.3 V. To overcome this, TXS0104E 4-bit logic-level converters by Texas Instruments were implemented. 0.1 µF ceramic decoupling capacitors were also added to both the 3.3 V and 5 V inputs, for high-frequency filtering, for each logic-level converter.

When incorporating the use of 5 thermistors into the PCB design there was a shortage of available input/output pins on the microcontroller. To overcome this the 8-channel analog multiplexer SN74LV4051A by Texas Instruments was added into the design for simple logic switching between each thermistor. To assure accurate measurements, from a hardware perspective, 10 kΩ pull-up resistors were selected with a deviation of 1%.

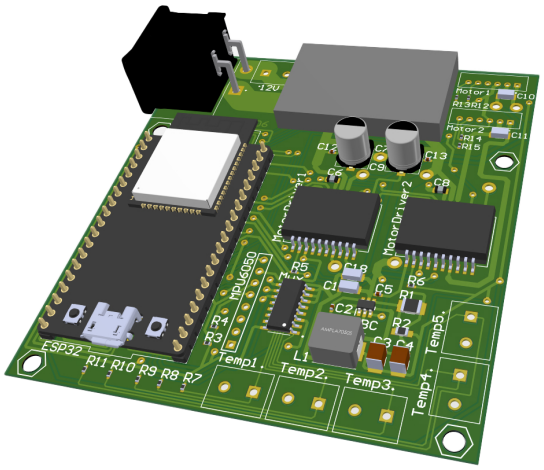


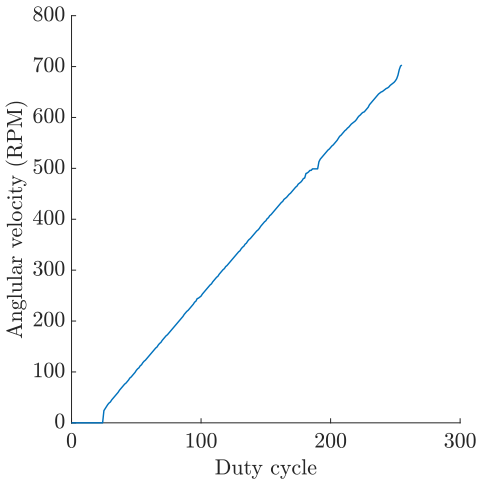
Figure 3.9. 3D model of the PCB.

3.4 Motor Control

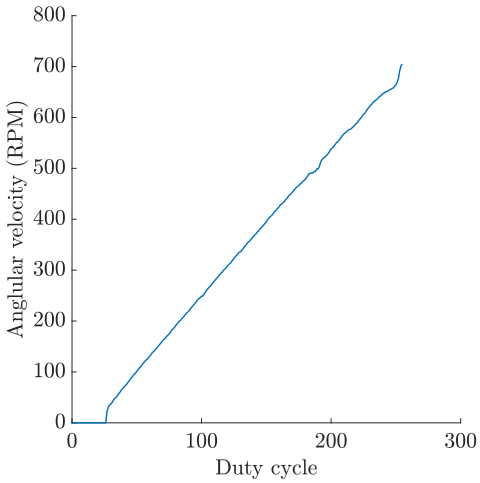
When trying to achieve stabilization of the aforementioned system, it is crucial to ensure sufficient motor control. To discover the motor's characteristics it was seen as important to measure the relation between duty cycle and achieved RPM. The motor's RPM can be obtained by using the magnetic encoders. By counting the number of pulses detected by one of the motor's Hall effect sensors over a specific time period, a basic calculation can be done to retrieve the RPM. An increase in the frequency of pulses corresponds to an increase in the RPM value of the motor. To discern the direction of rotation of the motors, the collective output of the two Hall effect sensors is compared. A detection order of North-south indicates one direction, while a detection order of South-north indicates the opposite. To implement this on the microcontroller it was necessary to incorporate the use of hardware interrupts, since the program would otherwise not be able to detect every single pulse, with standard program flow. Interrupt functions are implemented for each encoder, and are set to halt the program when one of the encoder outputs changes from a logical high to low or vice versa. The interrupt routine function for motor encoder 1 can be found in Appendix C.2. A hardware interrupt routine requires fast program execution. For this reason, an approximation was made in regard to the minimum angle increment detectable. By using the formula

$$\varphi_{\text{min}} = \frac{360}{N_{\text{poles}}} r \tag{3.3}$$

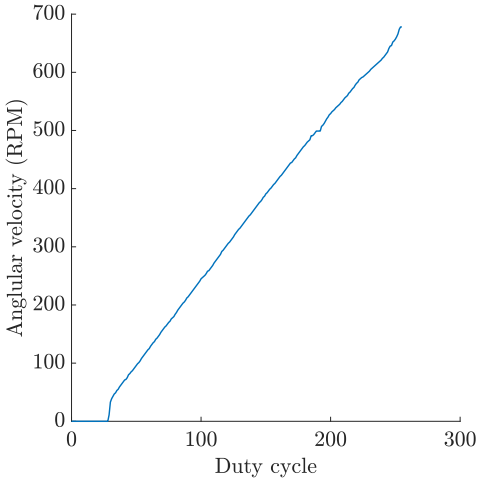
and the datasheet given gearing ratio of $r = 1/19.225$, in addition to an observed 6 poles per encoder, the minimum angle increment was calculated and rounded to be 3° [15]. To calculate the RPM for each motor the program keeps track of the total incremented angeled measured every 100ms. The implemented code function for this can be found in Appendix C.3. The measured RPM to duty cycle characteristics can be found in Fig. 3.10. Based on the results from these figures a relatively clear linearity can be observed from a duty cycle value of approximately 30. To overcome the motors duty cycle dead zone, an offset is added. To still achieve the same 8-bit resolution as before, the controller's output values are scaled to fit this new working range. With this adjustment, the linearity is suitable for the implementation of a controller.



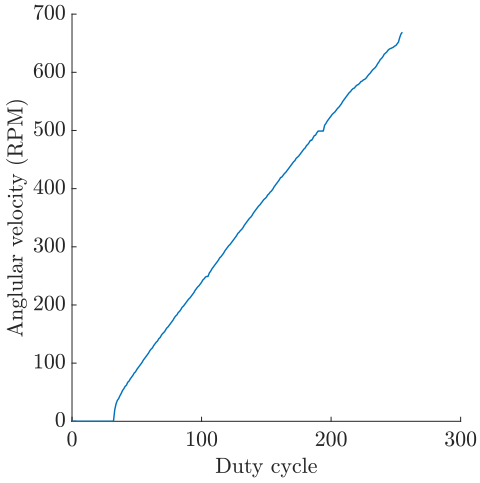
(a) Motor 1 clockwise direction.



(b) Motor 1 counterclockwise direction.

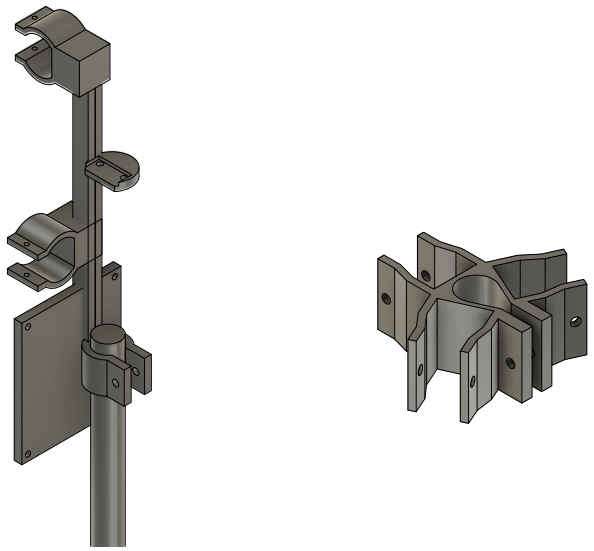


(c) Motor 2 clockwise direction.



(d) Motor 2 counterclockwise direction.

Figure 3.10. Relation between duty cycle and RPM.



(a) The frame. (b) The battery mount.

Figure 3.11. CAD models of main mounts.

3.5 Mechanical Design

The mechanical components tie the electronics and the pendulum together. The specified criteria posed are that of a cheap and effortlessly reproducible system. Therefore the development of the mechanical system was done with Fusion 360, and thereafter produced by 3D-printing in PLA. The printing was done for free through the student organization MAKE NTNU, which made trial and error throughout the process feasible. The design is based around the necessary electrical components with emphasis on a centered weight distribution close to the pendulum's vertical. More in-depth dimensional breakdowns are presented in Appendix B.

The mechanical system is split into three main parts - the frame, the inertia wheels, and the pendulum. The frame is further based on individually developed components that have been combined. This includes, among other things, holds for the motors and mounting platforms for the PCB and IMU. As seen in Fig. 3.11a, the holds for the electrical components are adjacent to the pendulum's vertical. The motors are attached at different points along the z-axis to keep the center of mass in close proximity of the pendulum's vertical. Additionally, the difference in height allows for the reaction wheels having a greater radius, without moving the motors away from the z-axis. The design of the frame also makes changing the pendulum or adjusting the mounting point seamless. There is also developed a separate mount for the batteries, as seen in Fig. 3.11b. Although it isn't physically connected, it's counted as an extension of the frame, making it possible to switch between a portable system and a system with an external power supply.

It is necessary to construct the reaction wheel such that its moment of inertia is sufficiently large, as its purpose is to generate a torque that can change the orientation of the pendulum and resist the force of gravity. This was done by increasing the radius and mass, as the moment of inertia of an annular cylinder rotating about a single perpendicular axis

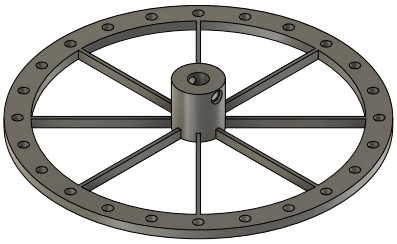


Figure 3.12. CAD model of the reaction wheel.

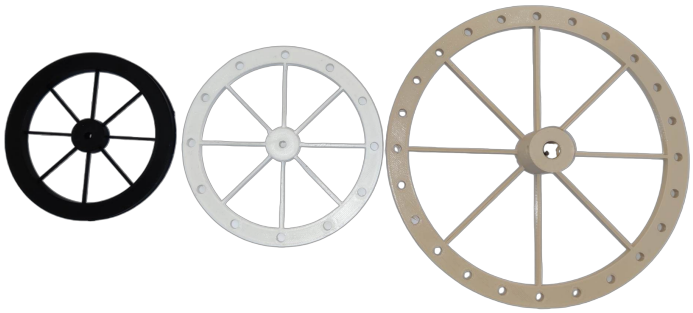


Figure 3.13. Reaction wheel iterations.

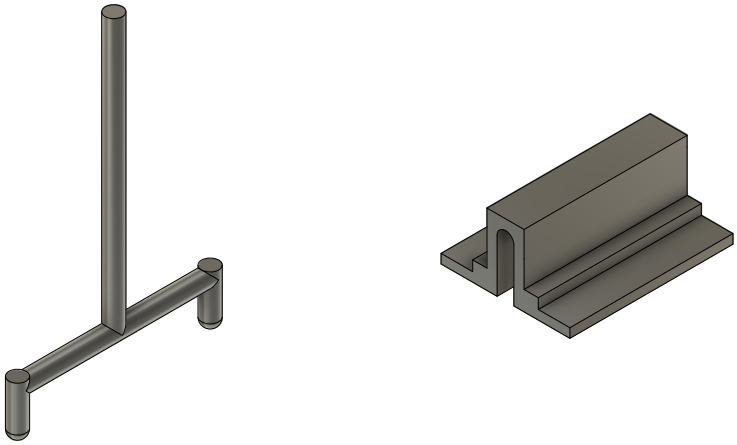
intersecting its center is given as

$$J_{\perp} = \frac{M}{2}\left(R_1^2 + R_2^2\right), \tag{3.4}$$

where M is the mass, R_1 is the inner radius, and R_2 is the outer radius. As seen in Fig. 3.12, the wheel is designed to have a modular weight, where the holes on the outer rim are made for mounting screws. During the course of the development phase, iterative experimental testing showed the need for adjustments in the reaction wheel design, where the modifications were centered around the radius and weight distribution, as shown in Fig. 3.13. The first iteration heavily underestimated the required weight and radius. Thus, the second iteration added modular weight, and the final model saw a substantial increase in both the radius and amount of holes.

In addition to the standard pendulum with no constraints placed on it, a pendulum that only rotates about a single axis was designed. Alongside the single-axis pendulum, a calibration tool that keeps the system perpendicular during initiation was developed. The CAD models of the double-ended pendulum and the calibration tool is displayed in Fig. 3.14a. Additionally, the finalized system is shown in Fig. 3.15.

For the assembly of the physical system, minor modifications were necessary. This pertains mainly to friction between different components or parts. In the physically realized system, seen in Fig. 3.15a, electrical tape was utilized to increase friction between the holds and the components. This was applied to the motors, pendulum and battery holder. In addition to this, rubber domes were placed on all of the legs of the pendula, with the purpose of increasing the friction. The reaction wheels were also altered, to facilitate efficient force transmission between the motor axle and the reaction wheel, an aluminum tube was installed at the center of the reaction wheel. The primary function of the tube is to secure the motor axle using two horizontal screws, ensuring a robust and secure lockdown. To make sure that the screws would not unwind themself while the wheel was rotating, Loctite was used.



(a) Double-ended pendulum. (b) Calibration tool.

Figure 3.14. CAD models of the components related to the single-axis system.

(a) The single-axis inverted pendulum system.

(b) The dual-axis inverted pendulum system.

Figure 3.15. Photographs of the realized pendulum system.

3.6 Bluetooth Communication

Bluetooth was selected as the communication standard of choice in regards to the HMI described in Section 3.7. The implemented communication features some unique solutions. On the microcontroller side, most variables are stored in struct objects to improve usability and overall comprehension. However, when implementing the transmit or receive functionality, each struct value would have had to be individually called to a corresponding input or output array of data. Instead of doing this, the union operator was implemented and encapsulated around every non-nested struct object, along with the addition of an array. An example of this structure setup is shown in Appendix C.1. With this implementation, the union operator allows the struct- and array datatypes to share the same address space in memory while being different types, meaning that each item in the struct could also be indexed like an array [34]. With this functionality, simple iteration over the union arrays was possible for updating desirable elements. The full Bluetooth transmission function can be found in Appendix C.4.

The receive and transmit functions use their own custom communication layouts for data, with only comma-separated values. This was done to help improve timing performance and reduce overhead. The impacts of sending data continuously from the microcontroller were measured to make sure that loop-iteration timings were not unreasonable, preventing the system from being responsive. The total calculation time for the inertial sensor fusion algorithm described in Section 4.3 was measured to approximately 10 µs. Bluetooth transmission time was measured to about 2.8 ms, with a total loop time for the entire system of approximately 2.9 ms while sending data. With this in mind, the data transmission was selected to be in periodic intervals of 100 ms, since this seemed to be a relatively reasonable compromise between low average loop timings and update rates in the HMI.

3.7 HMI Design

To control and monitor the complete system, an application was built using Python. The PC program has a graphical user interface and uses Bluetooth to communicate with the system. This was made possible by utilizing libraries such as CustomTkinter, to create the custom GUI and pySerial for Bluetooth communication. The program features real-time graphs and variables of the system's IMU measurements, angular rates of the motors, and temperature sensor readings. There are PID parameter input fields for the system's two controllers, buttons for sending different commands, and a notification window for events and statuses. Additionally, there are input fields for changing the filter constants related to the signal processing of the IMU data. This is further described in Chapter 4.

A typical use case of the program follows the procedure of first making sure that Bluetooth is enabled, and that the device named "ESP32-E2306" is connected. Thereafter one can press the connect button to establish true communication with the system. The buttons are intended to be self-explanatory, and actions made become visible in the notification window. After connecting, it's typical to proceed with pressing the calibrate button, which auto-calibrates all 6-DoF of the IMU, and then enter PID parameters. By selecting the enable checkboxes next to each controller, one can choose between one or two axial operation. It is also possible to input different filter constants. Known-good default values will be used if these fields are left empty. The updated button sends the current values of every input field to the system. To initiate monitoring the start button can be pressed, which will result in all data variables and graphs being updated in real-time. There are some caveats, however. The updates only happen every 100 ms, and due to performance problems related to the rendering updates of the graphs are often even less frequent. The

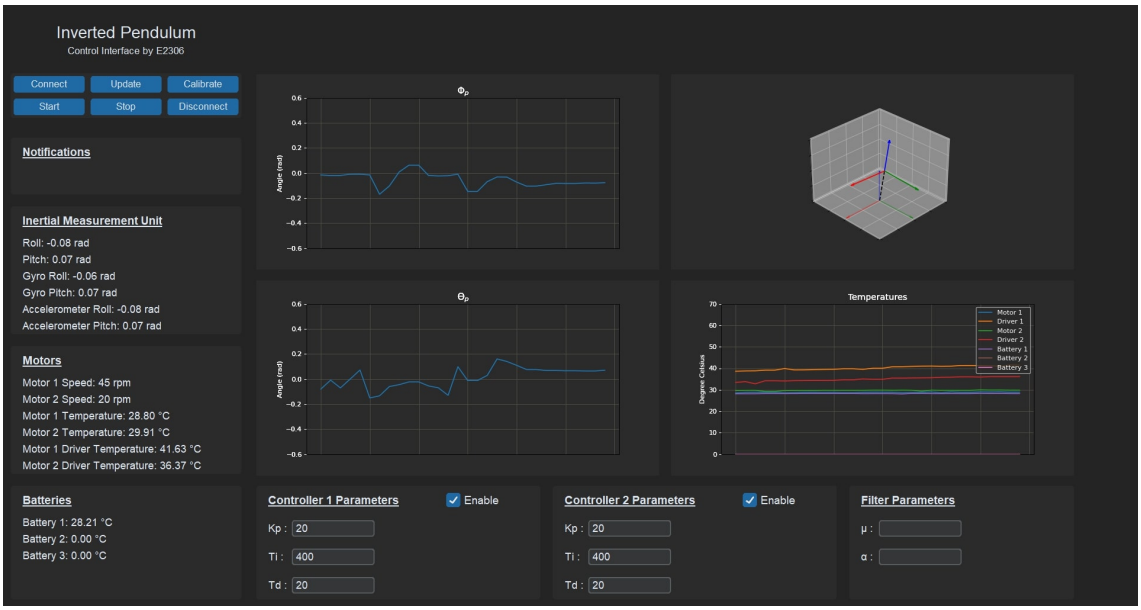


Figure 3.16. Human Machine Interface

complete program can be found as an executable file under the Github link provided in Section 5.3 along with the source code. The program is intended to work on any Windows 10/11 PC with a resolution of full HD and a scaling factor set to 100%.

Chapter 4

Implementation

This chapter focuses on providing a description of the algorithms running on the microcontroller related to the signals in the system. These algorithms will, in addition to enabling monitoring of the system in the HMI, lay the foundation for further testing and implementation of various control configurations.

4.1 Thermal Sensing

The general Steinhart-Hart equation with the unknown coefficients $\bar{\boldsymbol{\beta}}$, given by

$$\frac{1}{T} = \sum_{i=0}^{\infty} \bar{\beta}_{i+1} \ln^i R, \tag{4.1}$$

relates the resistance R to its temperature T, expressed in kelvin. In practice, simply considering $\bar{\beta}_i \equiv 0 \, \forall i \geq 5$ is a sufficient approximation, due to the higher order terms being significantly less weighted. Now, let $\boldsymbol{\beta} \in \mathbb{R}^4$ denote the new reduced order parameter vector, and let $x = T^{-1}$. For m observations $(R_1, x_1), \ldots, (R_m, x_m)$ with $m > 4$, the data follows the linear regression model

$$\mathbf{x} = \mathbf{f}(\mathbf{R}; \boldsymbol{\beta}) = \begin{bmatrix} 1 & \ln R_1 & \ln^2 R_1 & \ln^3 R_1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \ln R_m & \ln^2 R_m & \ln^3 R_m \end{bmatrix} \boldsymbol{\beta} + \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix} =: \mathbf{X}\boldsymbol{\beta} + \mathbf{w}, \tag{4.2}$$

where $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{\mathcal{I}})$ is additive white Gaussian noise. The parameter vector estimate is given by solving the quadratic minimization problem,

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{x} - \mathbf{X}\boldsymbol{\beta}\|^2 = \mathbf{X}^{\dagger}\mathbf{x}, \tag{4.3}$$

where $\mathbf{X}^{\dagger} \coloneqq \left(\mathbf{X}^{\top}\mathbf{X}\right)^{-1}\mathbf{X}^{\top}$ is the left Moore-Penrose pseudoinverse of $\mathbf{X}$. This yields

$$\begin{aligned} \hat{\beta}_1 &= 0.001152861267543, \\ \hat{\beta}_2 &= 2.272144502092691 \cdot 10^{-4}, \\ \hat{\beta}_3 &= 6.108642392225680 \cdot 10^{-7}, \\ \hat{\beta}_4 &= 7.243392408492539 \cdot 10^{-8}. \end{aligned}$$

Using these coefficients, the estimated temperature expressed as a function of resistance, i.e., $\hat{T} = g(R; \hat{\boldsymbol{\beta}})$, is shown in Fig. 4.1.

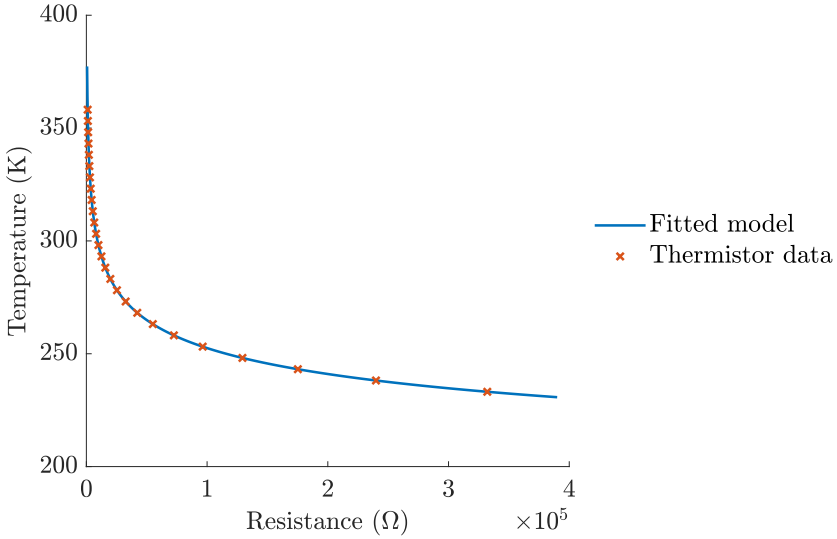


Figure 4.1. Result of ordinary least squares estimation. The data is fetched from [35].

4.2 Bias Compensation and Noise Filtering

Let $\mathbf{a} = \begin{bmatrix} a_x & a_y & a_z \end{bmatrix}^\top$ be the acceleration along each axis, and let $\boldsymbol{\omega}^\star = \begin{bmatrix} \omega_x^\star & \omega_y^\star & \omega_z^\star \end{bmatrix}^\top$ be the angular velocity about each axis, where both of the vectors are expressed in the body-attached frame. Now, let

$$\mathbf{z} = \begin{bmatrix} \mathbf{a} \\ \boldsymbol{\omega}^\star \end{bmatrix} \tag{4.4}$$

be a column vector in $\mathbb{R}^6$ that contains both the acceleration and angular rate data. The measured data, $\hat{\mathbf{z}}$, differs from the actual data, i.e.,

$$\hat{\mathbf{z}}[n] = \mathbf{z}[n] + \mathbf{w}[n] + \mathbf{b}[n], \tag{4.5}$$

with $\mathbf{w}[n] \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ and $\mathbf{b}[n]$ being the bias. The offset can be estimated by holding the pendulum still in its upright equilibrium such that $\mathbf{a} = \begin{bmatrix} 0 & 0 & -g \end{bmatrix}^\top$ and $\boldsymbol{\omega}^\star = \mathbf{0}$, and then computing the mean of $\mathfrak{H} + 1$ sensor readings,

$$\mathbf{b} \cong \frac{1}{\mathfrak{H}+1} \sum_{n=0}^{\mathfrak{H}} \hat{\mathbf{z}}[n] + g\hat{\mathbf{e}}_3, \tag{4.6}$$

assuming that the bias is constant $\forall n \geq 0$. Indeed, the reliability of the measured data is expected to increase by subtracting the approximated bias from the measurements,

$$\mathbf{z}[n] \cong \hat{\mathbf{z}}[n] - \mathbf{b}\vartheta[n], \tag{4.7}$$

where $\vartheta : \mathbb{Z} \to \{0, 1\}$ is the Heaviside step function. The bias may, however, exhibit temporal variability, and it is therefore deemed necessary to calibrate the IMU by computing a new approximation of $\mathbf{b}$ each time the microcontroller is booted.

While the white Gaussian noise exhibited by the gyroscope is less noticable and therefore deemed negligible, the presence of high-frequency noise in the accelerometer readings is quite prominent. This may be combated by implementing an exponential moving average filter, given by the difference equation

$$y[n] = \mu x[n] + (1 - \mu)y[n - 1], \tag{4.8}$$

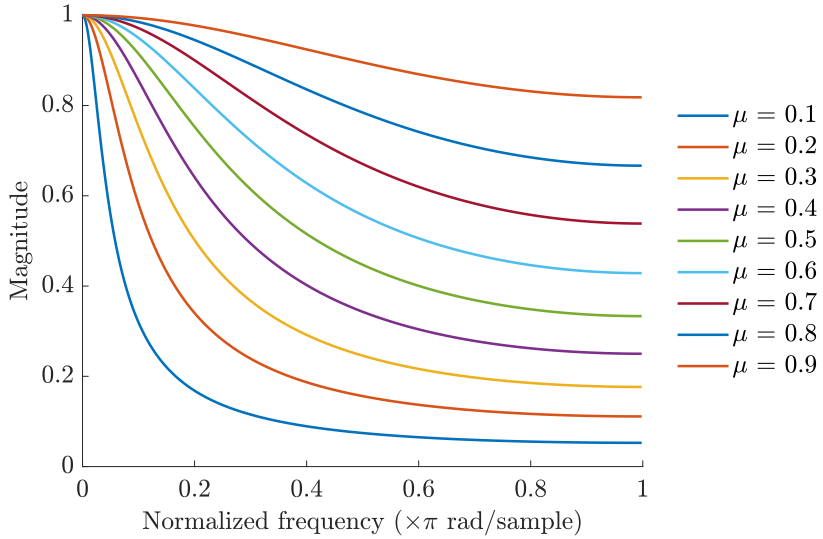


Figure 4.2. Magnitude response of the digital filter $H(z)$.

where μ is the smoothing factor. This corresponds to the IIR transfer function

$$H(z) = \frac{\mu}{1-(1-\mu)z^{-1}}, \tag{4.9}$$

implying that the filter is BIBO stable[1] $\forall \mu \in (0,1]$, though it is an all-pass filter in the limiting case where $\mu = 1$. The magnitude response $\left|H(e^{j\omega})\right|$ for different values of μ is shown in Fig. 4.2. As seen in the figure, the attenuation of high-frequency noise increases as μ approaches 0. This will, however, give more weight to older observations. It is therefore necessary to compromise between smooth data and responsiveness. A comparison between the filtered and unfiltered accelerometer data with $\mu = 0.3$ is shown in Fig. 4.3. Here, $\{\widetilde{a}_x, \widetilde{a}_y, \widetilde{a}_z\}$ is the set of unfiltered data, while $\{a_x, a_y, a_z\}$ is the set of filtered data. The reason for not opting for an FIR filter such as the simple moving average filter is that IIR filters generally can, for a given filter order, satisfy a tighter specification than FIR filters, implying lower computational complexity [36]. Besides, FIR filters generally require a higher filter order, and thus, they admit more delay.

4.3 Inertial Sensor Fusion

Using the IMU, the orientation of the pendulum can be described using the 3-2-1 set of Euler angles corresponding to the following sequence of basic rotations:

1. A rotation of ψ about the fixed z-axis (yaw).
2. A rotation of θ about the fixed y-axis (pitch).
3. A rotation of ϕ about the fixed x-axis (roll).

[1] An LTI system is BIBO stable if any bounded input (i.e., $\exists \gamma_u$ such that $||\mathbf{u}|| \leq \gamma_u$) yields a bounded output (i.e., $\exists \gamma_y$ such that $||\mathbf{y}|| \leq \gamma_y$).

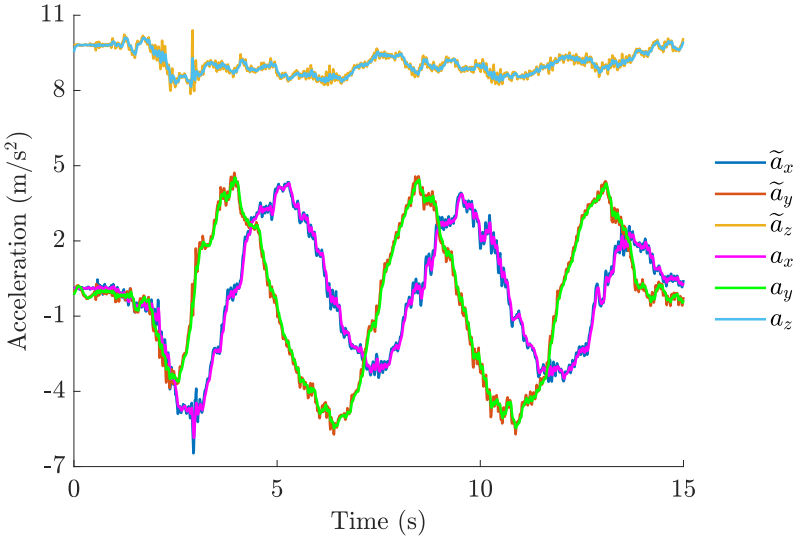


Figure 4.3. Effect of smoothing the scaled accelerometer data.

Therefore, the orientation transformation between the body-attached frame $o_1x_1y_1z_1$ and the inertial frame $o_0x_0y_0z_0$ is given by

$$\mathbf{R}_1^0 = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix}. \quad (4.10)$$

The accelerometer readings relate to the Euler angles by rotating the gravity vector, i.e.,

$$\mathbf{a} = \mathbf{R}_1^0 \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} = g \begin{bmatrix} \sin\theta \\ -\sin\phi\cos\theta \\ -\cos\phi\cos\theta \end{bmatrix}, \quad (4.11)$$

assuming that the accelerometer is at rest. From this, one can show that the roll and pitch angles can be estimated from the accelerometer readings:

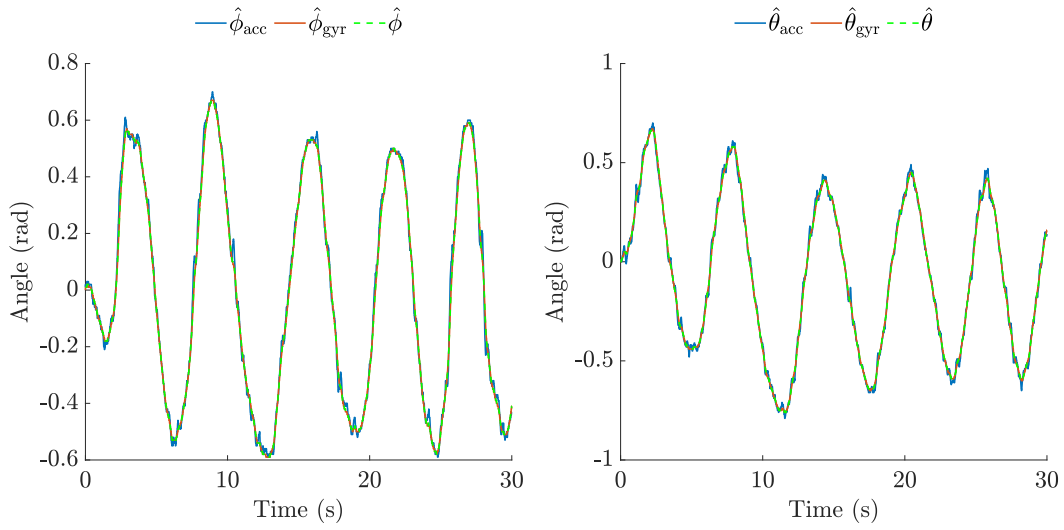
$$\begin{bmatrix} \hat{\phi}_{\text{acc}} \\ \hat{\theta}_{\text{acc}} \end{bmatrix} = \begin{bmatrix} \text{atan2}\left(a_y, a_z\right) \\ \text{atan2}\left(-a_x, \sqrt{a_y^2 + a_z^2}\right) \end{bmatrix}. \quad (4.12)$$

Let $\boldsymbol{\omega} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}$ denote the angular velocities about each axis in the inertial frame, with the correlation to the rates in the body-attached frame given by the transformation

$$\boldsymbol{\omega} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \boldsymbol{\omega}^\star. \quad (4.13)$$

Please note that the singularities in the matrix are inconsequential, as automatic stabilization requires the system to be in close proximity of the equilibrium at all times. For practical implementations, the angular velocities may be converted to angles through the usage of the backward Euler discrete-time integrator, given by the transfer function

$$G(z) = \frac{T}{1 - z^{-1}}, \quad (4.14)$$



(a) Estimation of the roll angle.

(b) Estimation of the pitch angle.

Figure 4.4. Comparison of the Euler angle estimates.

where $T > 0$ is the sampling period. Hence, the gyroscope's pitch and roll angle estimates are given by the difference equations

$$\begin{bmatrix} \hat{\phi}_{\text{gyr}}[n] \\ \hat{\theta}_{\text{gyr}}[n] \end{bmatrix} = T \begin{bmatrix} \omega_x[n] \\ \omega_y[n] \end{bmatrix} + \begin{bmatrix} \hat{\phi}_{\text{gyr}}[n-1] \\ \hat{\theta}_{\text{gyr}}[n-1] \end{bmatrix}, \tag{4.15}$$

and now, the pitch and roll estimates given by the gyroscope and the accelerometer can be combined by applying the complementary filter,

$$\begin{bmatrix} \hat{\phi}[n] \\ \hat{\theta}[n] \end{bmatrix} = \alpha \begin{bmatrix} \hat{\phi}_{\text{acc}}[n] \\ \hat{\theta}_{\text{acc}}[n] \end{bmatrix} + (1-\alpha) \begin{bmatrix} \hat{\phi}_{\text{gyr}}[n] \\ \hat{\theta}_{\text{gyr}}[n] \end{bmatrix}, \tag{4.16}$$

where the weighting factor $\alpha \in [0, 1]$. Typically, α is close to zero, as gyroscopes generally provide better estimates. The result of selecting $\alpha = 0.05$ is shown in Fig. 4.4, where an arbitrary rotation of the pendulum has been performed. Finally, the correlation between the Euler angle estimates and the generalized coordinates describing the orientation of the pendulum, given the orientation of the IMU in the mechanical system, is given as

$$\theta_p \cong \hat{\theta} \quad ; \quad \phi_p \cong \hat{\phi}, \tag{4.17}$$

and the control goal is to keep $\begin{bmatrix} \theta_p & \phi_p \end{bmatrix}^\top$ close to $\mathbf{0}$ at all times.

4.4 Automatic Stabilization

The pendulum may be stabilized by implementing a control law that automatically corrects the control error $e = r - y = -y$, e.g., the PID controller given by

$$u(t) = -K_p y(t) - \frac{K_p}{T_i} \int_0^t y(\tau)\,\mathrm{d}\tau - K_p T_d \frac{\mathrm{d}y}{\mathrm{d}t}. \tag{4.18}$$

Here, K_p is the proportional gain, T_i is the integral time, and T_d is the derivative time. Indeed, in its current form, the controller is non-causal due to the transfer function from

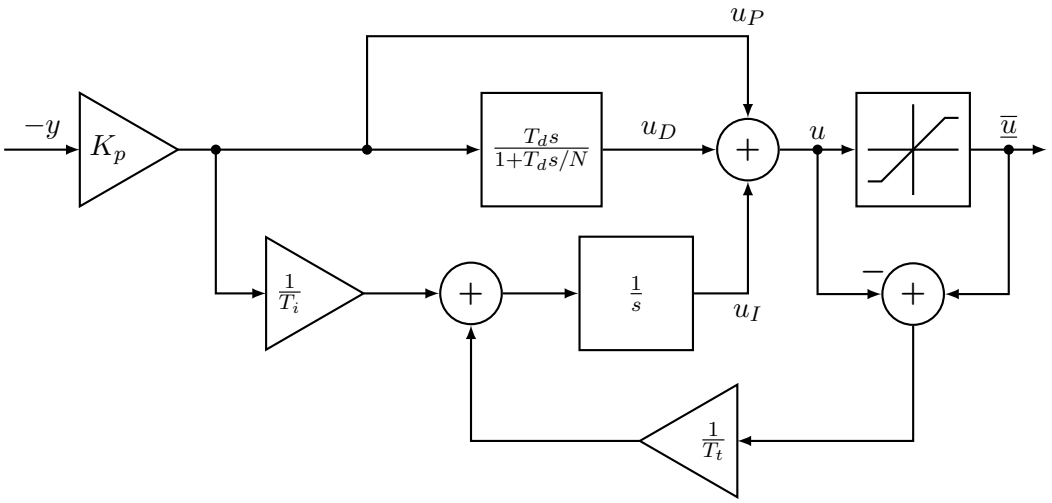


Figure 4.5. Block diagram of a causal PID controller with anti-windup, inspired by [37].

the output to the derivative term being improper. This can be combated by implementing it as a first order high-pass filter, namely

$$G(s) = \frac{U_D(s)}{Y(s)} = -K_p \frac{T_d s}{1 + \frac{T_d}{N} s}, \tag{4.19}$$

where $N \in [2, 20]$ are typical values for defining the filter constant [37]. It is also desirable to constrain the control signal by some upper and lower bound,

$$\underline{\overline{u}}(t) \coloneqq \min\left(u^\star, \max\left(u(t), -u^\star\right)\right), \tag{4.20}$$

as this prevents integer overflow. Here, $u^\star$ is the chosen maximum duty cycle of the DC motors. It may be the case that $|u(t)| > u^\star$ for some values of t, which motivates the implementation of anti-windup compensation. Thus, the integral term is implemented as

$$u_I(t) = \int_0^t \frac{1}{T_t} \left[\underline{\overline{u}}(\tau) - u(\tau)\right] - \frac{K_p}{T_i} y(\tau) \, \mathrm{d}\tau, \tag{4.21}$$

where $T_d < T_t < T_i$ is the tracking time constant. A rule of thumb is to simply select $T_t = \sqrt{T_i T_d}$ [37]. The control configuration to be implemented in the microcontroller is depicted in Fig. 4.5, and indeed, doing so requires the algorithm to be represented in discrete-time. The conformal mapping

$$z = e^{sT} \iff s = \frac{1}{T} \ln z \tag{4.22}$$

may be used to discretize the derivative term. By simply considering the first order Padé approximant, i.e., the bilinear transformation

$$s \cong \frac{2}{T} \frac{z-1}{z+1}, \tag{4.23}$$

the discrete transfer function from $Y(z)$ to $U_D(z)$ is given as

$$G(z) = -K_p \zeta \frac{1 - z^{-1}}{1 + \xi z^{-1}}, \tag{4.24}$$

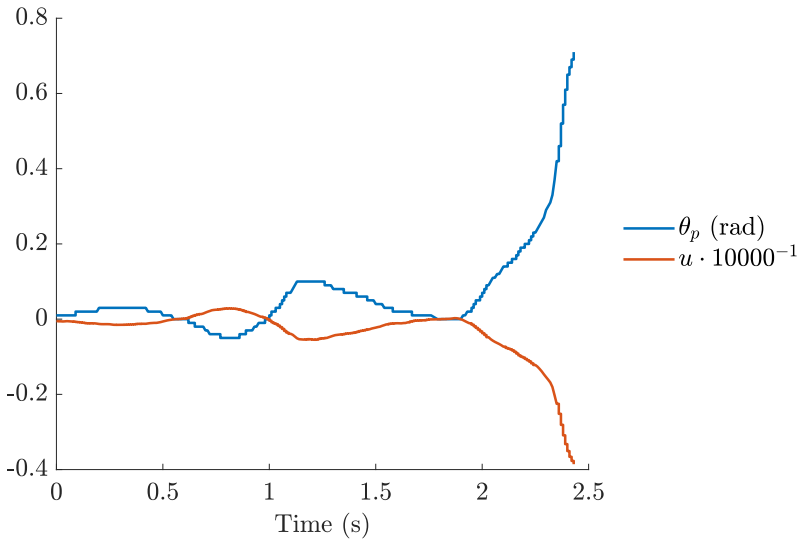


Figure 4.6. Attempted stabilization of the single-axis system.

where $\xi = \frac{TN-2T_d}{TN+2T_d}$ and $[0,1) \ni \zeta = \frac{2T_dN}{TN+2T_d}$. Note that setting $s = j\Omega$ and $z = e^{j\omega}$ yields the frequency mapping

$$\Omega = \frac{2}{T} \tan \frac{\omega T}{2}, \tag{4.25}$$

implying that the bilinear transformation exhibits a nonlinear distortion of the frequency response. It is, however, assumed that the effect of the warping is negligible, as the approximation $\Omega \cong \omega$ holds when $\omega \ll \frac{\pi}{T}$ [38]. The discrete transfer function corresponds to the discrete-time difference equation

$$u_D[n] = -\xi u_D[n-1] - K_p\zeta \left(y[n] - y[n-1]\right). \tag{4.26}$$

The proportional term can be transformed directly,

$$u_P[n] = -K_p y[n], \tag{4.27}$$

and lastly, the integral term may be approximated by a Riemann sum, yielding

$$u_I[n] = u_I[n-1] + \chi\left(\underline{u}[n] - u[n]\right) - K_p\sigma y[n], \tag{4.28}$$

where $\chi = \frac{T}{T_t}$ is the discrete tracking time and $\sigma = \frac{T}{T_i}$ is the discrete integral time [39]. Fig. 4.6 shows the performance of the controller for both the SISO and MIMO systems. As seen in the figure, closed-loop stability is not obtained due to the need for rather aggressive PID parameters. The mechanical system may, in its current state, not be possible to stabilize using PID control, as the moment of inertia of the reaction wheel is not sufficiently large. This implies that generating a torque with the ability to change the orientation of the pendulum relies on the control signal being overwhelmingly large even when the control error is small, leaving the output quite oscillatory.

Chapter 5

Conclusions

5.1 Summary

In Chapter 2, the models describing the dynamics of both the SISO and MIMO systems was derived. This was done by using the Euler-Lagrange equations. Then, the models were linearized about the upright equilibrium, which allows for implementation using linear model-based control configurations. In Chapter 3, a description of all selected hardware components was provided, along with considerations for the implemented PCB design. The mechanical components that facilitate the connection between the electronics and the pendulum, as well as the realized system, were also presented. Furthermore, a description of the HMI was given, in addition to an insight into different Arduino C code snippets that are crucial for the system's functionality. In Chapter 4, the algorithms implemented on the microcontroller were presented. This mainly involved signal processing of the sensor data, which laid the foundation for the development of the control system. The control configuration consisted of a causal PID controller with back-calculation implemented.

5.2 Further Work

The purpose of the system is to lay a foundation for further experimentation and development. Improvements in the reaction wheel design is a critical step towards automatically stabilizing both the single-axis and dual-axis systems. Subsequently, achieving closed-loop stability using optimal control laws such as the linear-quadratic regulator would be a reasonable goal. Indeed, full state feedback would require all the states of the system to be known. As $\dot{\theta}_p$ and $\dot{\phi}_p$ are the only states that are not being measured, they may simply be approximated using backward difference approximation. Implementing a state observer such as the Kalman filter may be a viable strategy as well. Furthermore, fusing the inertial sensors using the Kalman filter may provide better performance, as the reliability of the Euler angle estimates are expected to increase as a result of this.

As of yet, model based control schemes are not implementable due to parameters in the models derived in Chapter 2 being unknown. Hence, applying methods from the field of system identification is necessary, before the model discretization process can proceed. Once this is done, the model can be implemented in the code. Although the model describing the dynamics of the dual-axis reaction wheel pendulum is likely sufficient for stabilization and disturbance rejection, developing a more accurate expression for the rotational kinetic energy may yield a model that enhances the performance of the control system. This is especially necessary if it is desired to operate further away from the upright equilibrium, for instance by experimenting with swing-up control, as the model assumes

that $\theta_p(t) \cong \phi_p(t) \cong 0 \, \forall t \geq 0$.

As stated in Chapter 3, the selection of brushed motors in the system was grounded on the basis of modeling analysis. However, extensive testing has revealed the occurrence of wear and sudden variations in rotation within the brushed motors, leading to a decrease in performance and accuracy. A potential enhancement for the system entails the integration of brushless motors, which can yield higher efficiency, reliability and smoother operation [40].

Further development of the mechanical design should explore the possible utilization of different build material. The main material used in this system is PLA, but there are other options for 3D-printing such as resin. After conducting several assembly processes, it has been observed that the durability of the motor holds is compromised when repeatedly inserting and removing the motor from the holds. There is also noticeable damage done during prolonged stress testing of the system. Hence, it may be necessary to consider the implementation of a flexible motor holder mechanism capable of facilitating the opening and closing of the holder around the motors, with the aim of improving the durability of the system.

Occasionally, there may arise instances where the reaction wheels exhibit a tendency to loosen from the motor shaft, necessitating the periodic tightening of the screws in order to maintain proper functionality. To establish a more reliable and long-term solution, alternative solution such as the application of adhesive bonding or the utilization of a shaft inserted through the motor shaft can be considered to securely position the reaction wheel. The usage of metallic reaction wheels could lower the needed radius which would make the system more compact. For further increase of the system's overall usability, there is a necessity for development of a platform that stops the pendulum from falling over, as well as a calibration tool for the dual-axis pendulum system.

There are some clear areas of improvement regarding the PCB design. The J5019 charging circuit used, while technically functional, was nowhere near fast enough for achieving a reasonable charging time. Instead of using USB charging, the new revision should instead have a standard 12 V barrel jack plug. In regards to the thermistors, one significant change was made in terms of use after the PCB was completed. Originally, each battery and motor was intended to have its own thermistor, but it was later seen as more important to monitor the motor drivers rather than the batteries. Since the PCB was only designed to have five total thermistor inputs, it was only possible to monitor one of the batteries. In an updated design, two more 10 kΩ pull-up resistors and screw terminals can be added, and appropriately routed to the existing 8-channel multiplexer. In addition to this, it is worth mentioning that having the system powered on while a USB cable is connected to the microcontroller for re-programming will result in the microcontroller's onboard regulator being damaged due to small differences in voltage. To prevent this possible scenario from happening, a Schottky diode may be added to the output of the PCB's buck converter.

The HMI program's main disadvantage is related to the performance of updating the real-time graphs. The performance can increase with the use of powerful PC hardware, but it is more desirable optimize graph rendering. The technique of blitting is currently used along with multiprocessing, but these may not be utilized to their full potential. The GUI can also be improved to support a wider range of resolutions and scaling factors. From a monitoring perspective, it could be beneficial to add a real-time graph of the control signal. The implemented functions for handling errors and edge cases catch a reasonable amount of scenarios, but there are likely still improvements to be made in this regard as well. A final restructuring of the source code with a more object-oriented programming approach may make the program more comprehensible.

In order to further enhance the system as an advanced learning platform, it is useful to devise a versatile system that facilitates seamless adaptation, allowing for removal of electrical components from the pendulum. The primary constraint within the system lies in the requirement for an adequate length of cables to accommodate the inclusion of the IMU, motors and thermistors. The introduction of longer cables enables the capability to detach the PCB and batteries from the pendulum, resulting in the motors, IMU and thermistors as the sole remaining electrical components on the pendulum. This mechanical configuration could possibly facilitate the testing of different stabilization strategies and enhancing the system's overall versatility.

A notable aspect of the thesis involved maintaining cost adherence within the designated budget of 3000 NOK. However, it is conceivable that alternative construction approaches exist, offering potential avenues to attain an even more economically viable implementation of the system. Exploring alternatives such as employing lower-cost thermistors may offer potential economic savings, though the accuracy of the measurements may decrease as a result of this. Another alternative avenue to be explored encompasses the disassembly of the PCB board and batteries from the pendulum, effectively eliminating the necessity for the implementation of expensive and powerful motors. With this approach, it becomes feasible to substitute the current motors with more economical and comparatively less powerful alternatives.

5.3 Concluding Remarks

To enable others to engage in further study and development of the conducted work, unrestricted access to all written code for both the microcontroller and the HMI is granted. The self-designed PCB, and CAD files for 3D-printing are also made available for further development. All the aforementioned files have been organized in a Github repository, accessible through the following hyperlink: **Github repository**.

References

[1] Russ Tedrake. *Underactuated Robotics. Algorithms for Walking, Running, Swimming, Flying, and Manipulation*. 2023. URL: https://underactuated.csail.mit.edu.

[2] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. 2nd ed. John Wiley & Sons, Inc., 2020.

[3] Damiano Varagnolo. *Equilibria*. TTK4225 Systems Theory, Introduction. NTNU, Department of Engineering Cybernetics, 2022.

[4] Torleif Anstensrud. *Tema 7 - Baneførging*. IELET2107 Robotics. NTNU, Department of Engineering Cybernetics, 2022.

[5] Damiano Varagnolo. *Nonlinear systems*. TTK4225 Systems Theory, Introduction. NTNU, Department of Engineering Cybernetics, 2022.

[6] Espressif. *ESP32*. URL: https://www.espressif.com/en/products/socs/esp32 (visited on May 11, 2023).

[7] Ravi Teja. *Getting Started with ESP32 — Introduction to ESP32*. URL: https://www.electronicshub.org/getting-started-with-esp32/ (visited on May 8, 2023).

[8] PROBOTS. *ESP-WROOM-32D ESP32 NodeMCU WiFi + BLE IoT Development Board*. URL: https://probots.co.in/esp32-wroom-32d-wifi-ble-bluetooth-iot-development.html (visited on May 18, 2023).

[9] Pololu Robotics & Electronics. *4.1. Choosing the motor, power supply, and Jrk*. URL: https://www.pololu.com/docs/0J73/4.1 (visited on May 13, 2023).

[10] NXP Semiconductor. *5.0 A H-Bridge*. URL: https://no.mouser.com/datasheet/2/302/MC33886-3138398.pdf (visited on May 13, 2023).

[11] Farnell. *290-006*. URL: https://no.farnell.com/productimages/large/en_GB/2528804-40.jpg (visited on May 13, 2023).

[12] Sky Tech Electronics. *MC33886PVW*. URL: https://www.skytech.ir/product_details.aspx?ID_Parts=9461&MC33886PVW (visited on May 19, 2023).

[13] zahidali. *Introduction to MPU6050*. URL: https://www.theengineeringprojects.com/2019/02/introduction-to-mpu6050.html (visited on May 2, 2023).

[14] Indiamart. *MPU-6050 3 Axis Accelerometer & Gyrometer Module*. URL: https://www.indiamart.com/proddetail/mpu-6050-3-axis-accelerometer-gyrometer-module-23923171473.html (visited on May 15, 2023).

[15] Sha Yang Ye Industrial CO. LTD. *Technical Data 1G220019X00015R*. URL: https://www.elfadistrelec.no/Web/Downloads/_t/ds/Digilent_MototrGearbox-1_19_eng_tds.pdf (visited on May 12, 2023).

[16] Elfa distrelec. *NTC Thermistor with Insulated Leads NTC 10kOhm*. URL: https://www.elfadistrelec.no/en/ntc-thermistor-with-insulated-leads-ntc-10kohm-3435k-40mm-vishay-ntcle413e2103f102l/p/30153876 (visited on May 15, 2023).

[17] Digilent. *DC Motor/Gearbox (1:19 Gear Ratio): Custom 12V Motor Designed for Digilent Robot Kits*. URL: https://digilent.com/shop/dc-motor-gearbox-1-19-gear-ratio-custom-12v-motor-designed-for-digilent-robot-kits/ (visited on May 18, 2023).

[18] Mark Lacriox, John Santos, and Rob Stiffler. *The Advantages of Magnetic Encoder Yechnology in Harsh Operating Environments*. Timken Company, 2011. URL: https://www.timken.com/resources/the-advantages-of-magnetic-encoder-technology-in-harsh-operating-pamphlet/.

[19] EFEST. *Efest IMR 18650 3000mAh 35A flat top battery*. URL: http://www.efestpower.com/index.php?ac=article&at=read&did=396 (visited on May 11, 2023).

[20] August Neverman. *Everything You Need to Know About the 18650 Battery*. URL: https://commonsensehome.com/18650-battery/ (visited on May 11, 2023).

[21] Bulkbattery. *Efest 18650 3000mAh 35A IMR Battery*. URL: https://www.bulkbattery.com/products/efest-18650-3000mah-35a-imr-battery (visited on May 8, 2023).

[22] ismokeking. *Efest 18650 - 3000 mAh*. URL: https://ismokeking.no/produkt/efest-18650-3000-mah-35-a/ (visited on May 15, 2023).

[23] YTDMEN. *3S 20A 25A Li-ion Lithium Battery 18650 Charger PCB BMS*. URL: https://ali.onl/2aIP (visited on May 18, 2023).

[24] Thomas L. Floyd. *Electronic Devices*. 10th ed. Conventional Current Version. Pearson, 2017.

[25] Wikipedia. *Pulse-width modulation*. URL: https://en.wikipedia.org/wiki/Pulse-width_modulation (visited on May 14, 2023).

[26] Jake Hertz. *H-bridge DC Motor Control Using Complementary PWM, Shoot-through, and Dead-time*. URL: https://www.allaboutcircuits.com/technical-articles/h-bridge-dc-motor-control-complementary-pulse-width-modulation-pwm-shoot-through-dead-time-pwm/ (visited on May 14, 2023).

[27] OurPCB. *10 Differences Between Through Hole And SMT Assembly*. URL: https://www.ourpcb.com/10-differences-between-through-hole-and-smt-assembly.html (visited on May 19, 2023).

[28] Frederik Dostal. *When Grounds Are Separated*. URL: https://www.analog.com/en/analog-dialogue/raqs/raq-issue-159.html (visited on May 20, 2023).

[29] Texas Instruments. *LMR51420 SIMPLE SWITCHER® Power Converter 4.5-V to 36-V, 2-A, Synchronous Buck Converter in a SOT-23 Package*. URL: https://www.ti.com/lit/ds/symlink/lmr51420.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-wwe%20ts=1684527196926ref_url=https%5C%253A%5C%252F%5C%252Fno.mouser.com%5C%252F (visited on May 19, 2023).

[30] Zachariah Peterson. *Decoupling Capacitor and Bypass Placement Guidelines*. URL: https://resources.altium.com/p/bypass-and-decoupling-capacitor-placement-guidelines (visited on May 19, 2023).

[31] Panasonic Industry. *Automotive Solutions*. URL: https://na.industrial.panasonic.com/automotive-solutions (visited on May 19, 2023).

[32] TDK. *Automotive Electronic Components for Safety Applications*. URL: https://www.tdk-electronics.tdk.com/download/140992/c57ef45a71967b5b91128491fcfa502d/ag-automotive-safety.pdf (visited on May 19, 2023).

[33] LTD CHANGZHOU JKONGMOTOR CO. *How to eliminate the noise of DC motor?* URL: https://www.jkongmotor.com/new_detail/nid/77757.html (visited on May 20, 2023).

[34] Brian W. Kernighan and Dennis M. Ritchie. *The C programming language*. 2nd ed. Prentice Hall, 1988.

[35] Vishay BCcomponents. *NTC Thermistors, Pipe PVC Long Leads Sensors*. URL: https://www.vishay.com/docs/29060/ntclp100.pdf (visited on May 15, 2023).

[36] Stefan Werner. *Design of Digital Filters: IIR*. TTT4120 Digital Signal Processing. NTNU, Department of Electronic Systems, 2020.

[37] Karl J. Åström and Tore Hägglund. *Advanced PID Control*. ISA - Instrumentation, Systems, and Automation Society, 2006.

[38] M. Sami Fadali and Antonio Visiolo. *Digital Control Engineering: Analysis and Design*. 3rd ed. Academic Press, 2019.

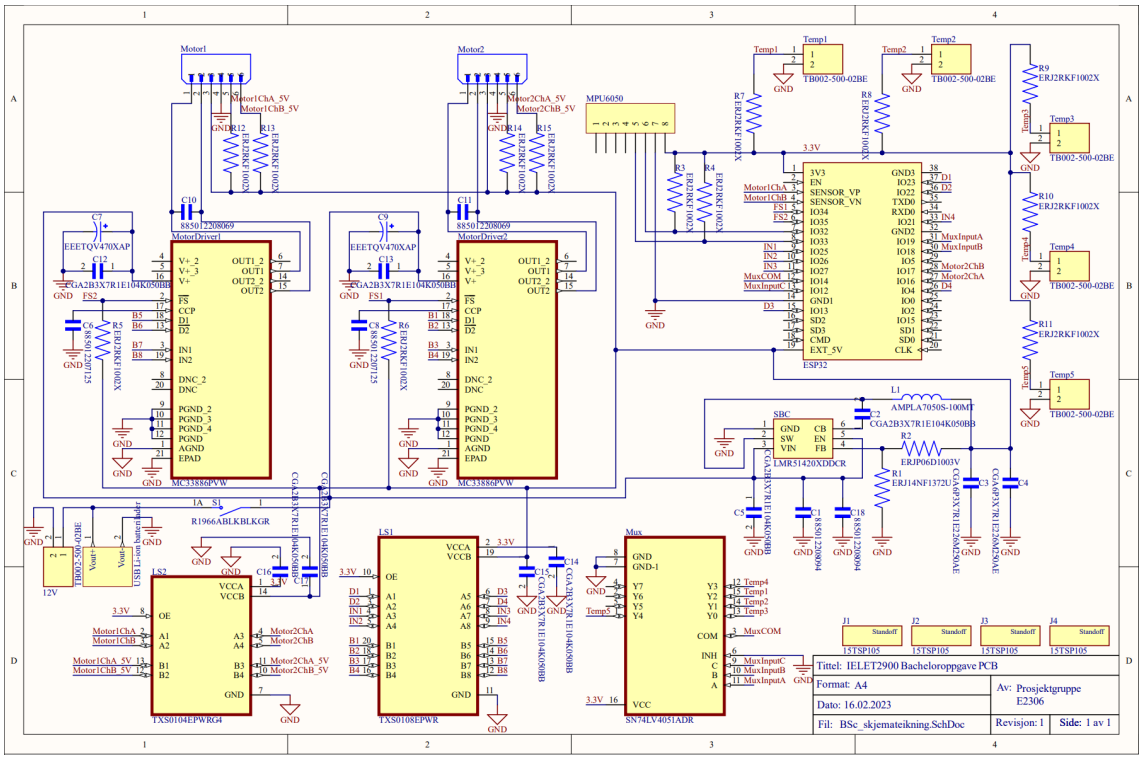
[39] Torleif Anstensrud. *Tema 10 - Digitale regulatorer*. IELET2102 Digital Control Systems. NTNU, Department of Engineering Cybernetics, 2022.

[40] KDEDirect. *Brushless Vs Brushed Motors*. URL: https://www.kdedirect.com/blogs/news/brushless-vs-brushed-motors (visited on May 18, 2023).

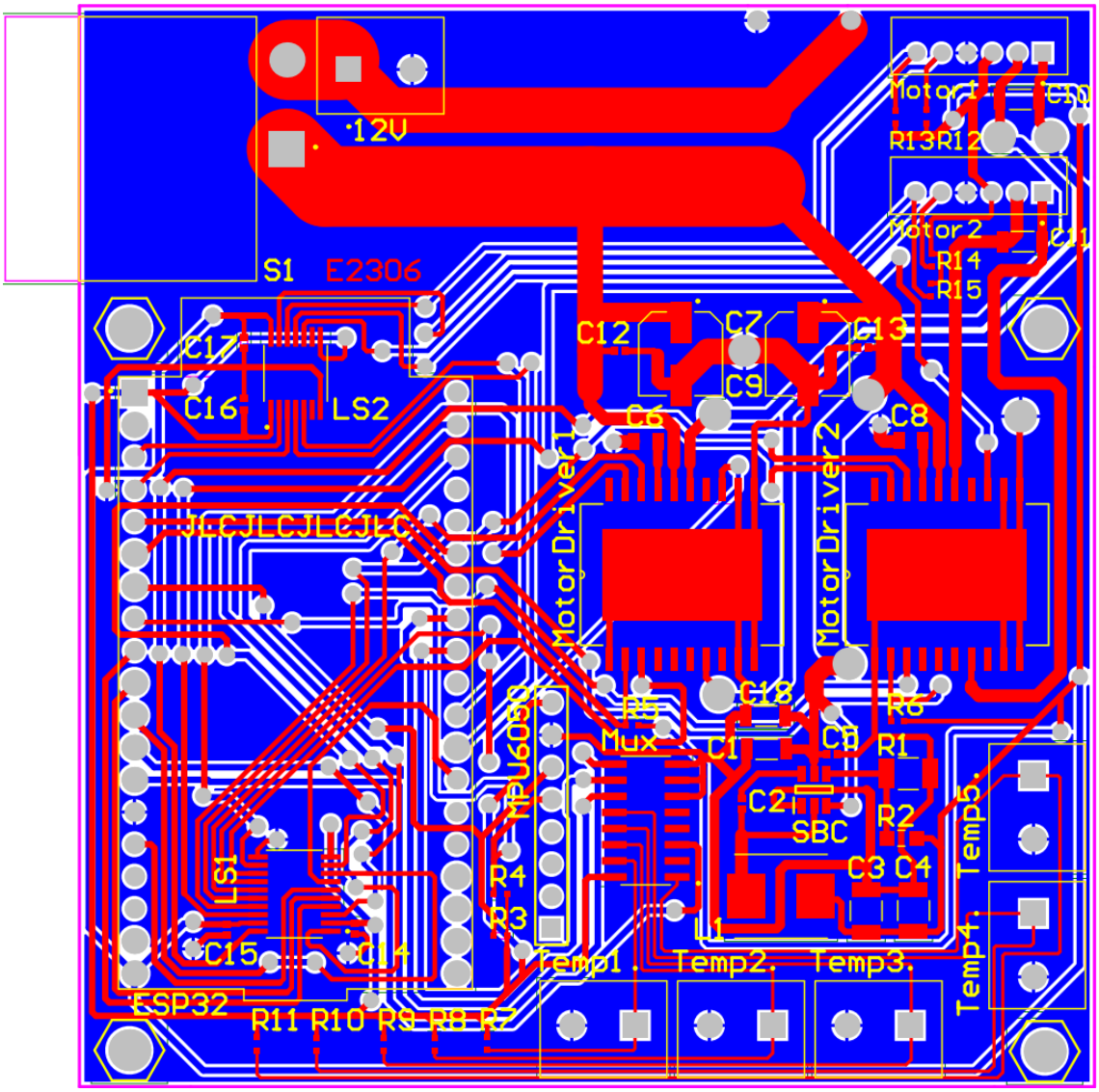
Appendix

A PCB Specifications and Design

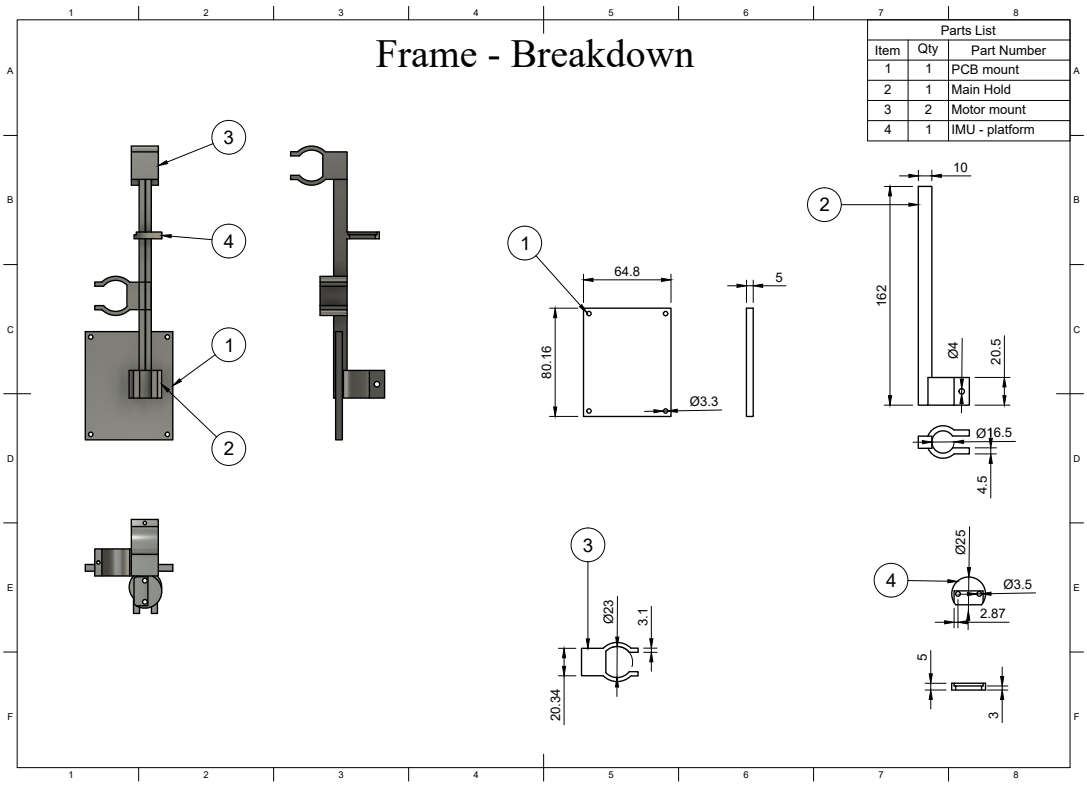
A.1 Schematic

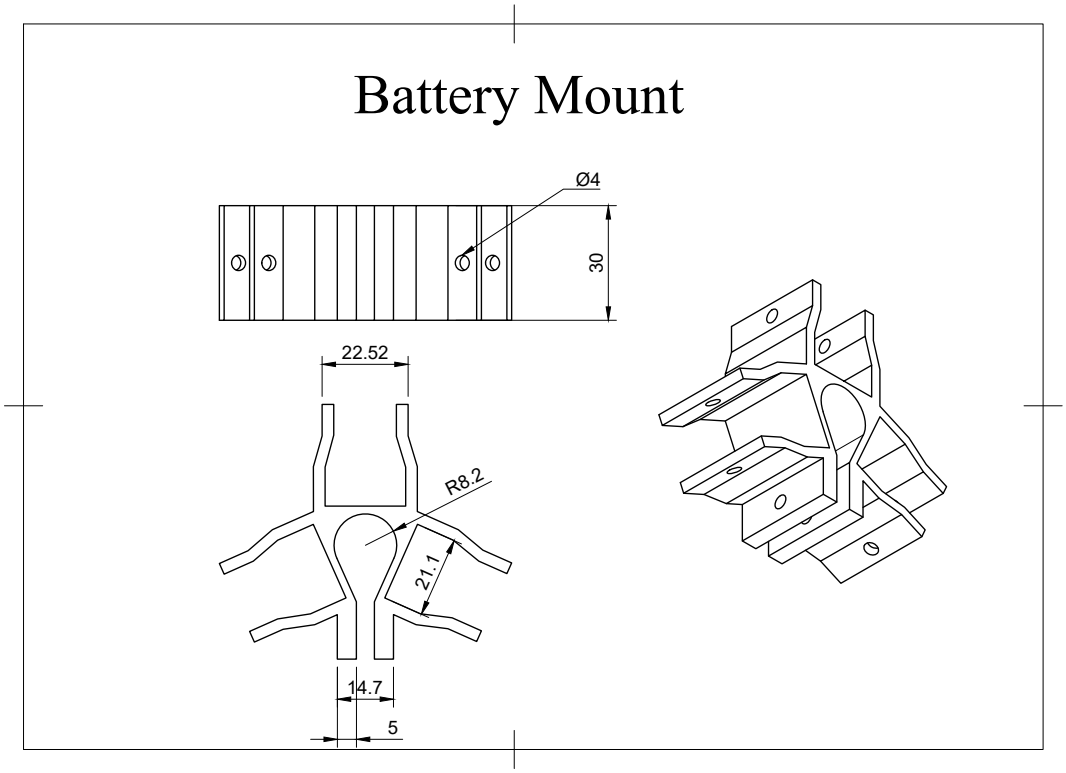


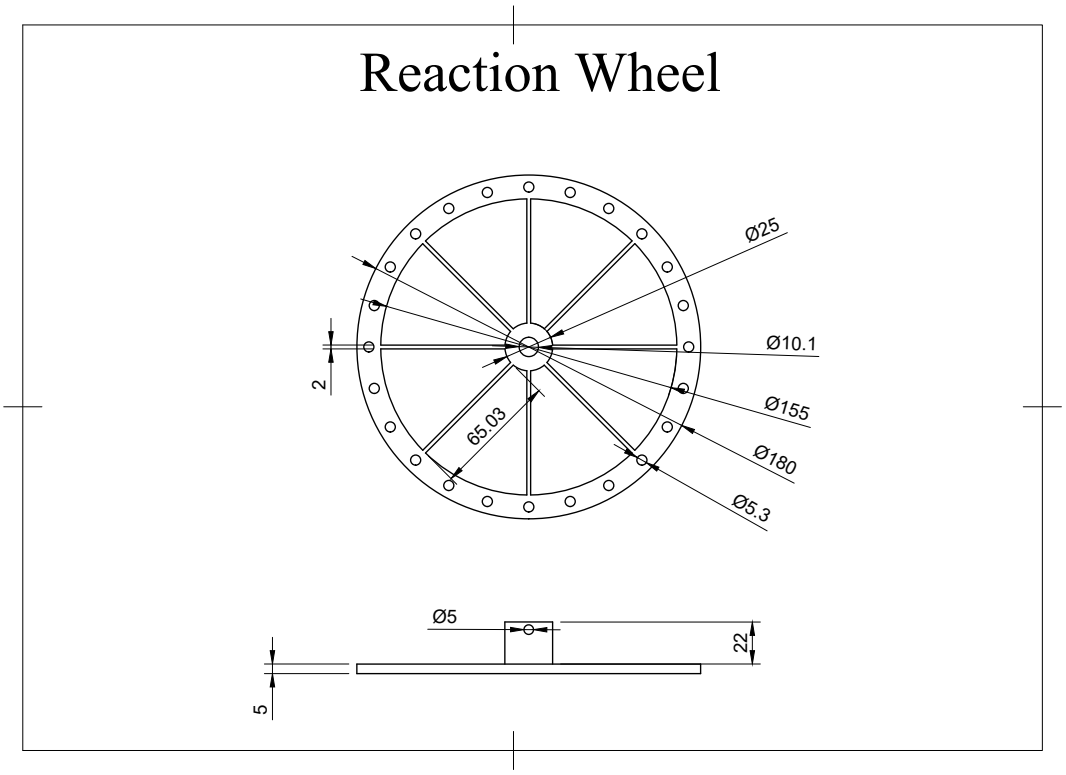
A.2 Circuit Diagram

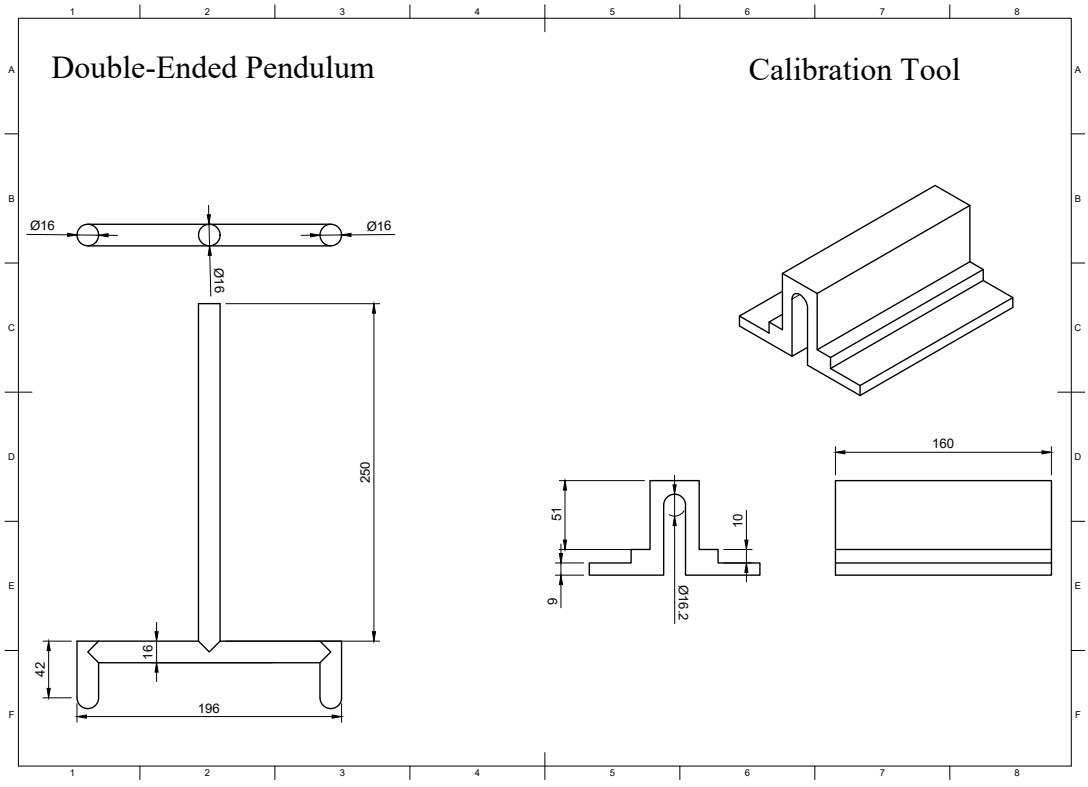


B CAD-Model Schematics






Reaction Wheel
Ø25
Ø10.1
Ø155
Ø180
Ø5.3
65.03
2
Ø5
22
5


Double-Ended Pendulum
Calibration Tool
Ø16
Ø16
Ø16
250
16
42
196
160
51
10
9
Ø16.2

C Arduino Code Snippets

C.1 Section of IMU variables structure

```
typedef union imu_t {
    struct imu_s{
        gyroProperties gyro;
        accelProperties accel;
        float roll;
        float pitch;
        float rollComp;
        float pitchComp;
        float rollPrev;
        float pitchPrev;
    }data;
    float imu_a[32];
};
```

C.2 Interrupt Routine for Motor 1 Encoder

```
void isrMotor1Encoder() {
    currStateMotor1EncoderA = digitalRead(motor1EncoderChannelA);
    if (currStateMotor1EncoderA != prevStateMotor1EncoderA) {
        if (digitalRead(motor1EncoderChannelB) != currStateMotor1EncoderA) {
            motor1.data.angle += angleIncrement;
            motor1.data.runningAngle += angleIncrement;
            motor1.data.rotation = cwRotation;
        }
        else {
            motor1.data.angle -= angleIncrement;
            motor1.data.runningAngle -= angleIncrement;
            motor1.data.rotation = ccwRotation;
        }
    }
    if (motor1.data.angle >= OneRotation || motor1.data.angle <= -(OneRotation)) {
        motor1.data.angle = 0;
    }
    prevStateMotor1EncoderA = currStateMotor1EncoderA;
}
```

C.3 Get Motor RPM Function

```
void getMotorsRPM() {
    unsigned long currentRPMTimer = esp_timer_get_time();

    if (currentRPMTimer - prevRPMTimer >= rpmSampleTime) {
        motor1.data.rpm = ((float)abs(motor1.data.runningAngle)/(float)OneRotation) *
↪   (float)oneMinute * ((float)rpmSampleTime/(float)1e4);
        motor2.data.rpm = ((float)abs(motor2.data.runningAngle)/(float)OneRotation) *
↪   (float)oneMinute * ((float)rpmSampleTime/(float)1e4);
        motor1.data.runningAngle = 0;
        motor2.data.runningAngle = 0;
```

```
9
10          prevRPMTimer = esp_timer_get_time();
11      }
12  }
```

C.4 Bluetooth Transmission Function

```
1   void sendDataOverBluetooth() {
2       if (esp_timer_get_time() - prevBluetoothTimer >= bluetoothUpdateInterval) {
3           float dataToSend[sendDataElements];
4
5           // Update IMU data to send
6           for (int i = 0; i < imuDataElements; i++) {
7               dataToSend[i] = imu.imu_a[imuElementsToSend[i]];
8           }
9           // Update Motor data to send
10          for (int i = 0; i < motorDataElements; i++) {
11              dataToSend[i+motor1DataElementOffset] =
    ↪   motor1.motor_a[motorElementsToSend[i]];
12              dataToSend[i+motor2DataElementOffset] =
    ↪   motor2.motor_a[motorElementsToSend[i]];
13          }
14          // Update Battery data to send
15          for (int i = 0; i < batteryDataElements; i++) {
16               dataToSend[i+batterElementOffset] = bat.batteries_a[i];
17          }
18          // Send all specified data elements
19          for (int i = 0; i < sendDataElements; i++) {
20              SerialBT.print(dataToSend[i], decimalAccuracy);
21              if (i < sendDataElements - 1) {
22                  SerialBT.print(" , ");
23              }
24              else {
25                  SerialBT.println("");
26              }
27          }
28          prevBluetoothTimer = esp_timer_get_time();
29      }
30  }
```

D Budget

Component	Description	Manufacturer	Qty.	Price*
MPU6050	IMU	TDK	1	17.00
ESP32-WROOM-32D	Microcontroller	Espressif Systems	1	98.00
J5019	Battery charger	N/A	1	39.00
3S BMS 20A	BMS	N/A	2	100.00
290-006	DC motor	Digilent, Inc.	2	401.00
B6B-PH-K-S(LF)(SN)	Motor socket	JST American Sales	2	7.22
PPTC191LFBN-RC	MCU socket	SullinsCorp	3	35.22
1043	Battery holder	Keystone Electronics	3	85.72
LMR51420XDDCR	Switching regulator	Texas Instruments	1	16.32
MC33886PVW	Motor driver	NXP	2	257.96
SN74LV4051ADR	Multiplexer	Texas Instruments	1	5.84
NTCLE400E3104HA	NTC Thermistor	Vishay	5	166.73
TB002-500-02BE	Screw terminal block	CUI Devices	6	35.70
R1966ABLKBLKGR	Button	E-Switch	1	19.65
AMPLA7050S-100MT	Inductor	ABRACON	1	17.87
885012207125	Capacitor	Wurth Elektronik	2	2.28
EEE-TQV470XAP	Capacitor	Panasonic	2	26.20
***	Capacitor	TDK	2	35.74
****	Capacitor	TDK	8	10.00
885012208069	Capacitor	Wurth Elektronik	2	12.52
885012208094	Capacitor	Wurth Elektronik	2	19.85
ERJ-P06D1003V	Resistor	Panasonic	1	3.44
ERJ-14NF1372U	Resistor	Panasonic	1	4.60
ERJ-2RKF1002X	Resistor	Panasonic	13	4.47
TXS0108EPWR	Logic level converter	Texas Instruments	1	16.43
TXS0104EPWRG4	Logic lever converter	Texas Instruments	1	13.32
15TSP105	Standoff	Essentra	4	23.80
010440R018	Screw	Essentra	10	16.90
PCB	Custom	JLCPCB	5	37.76
18650 3000MAH	Battery	EFEST	4	349
Std. EU power cord	Battery wires	N/A	1	70.00
1460-1297-ND	Motor cable	TMC GmbH	2	41.56
RPC2058-ND	Rubber dome	Essentra Components	12	40.38

Components were purchased from Digi-key Electronics, Mouser Electronics, Elkim, Clas Ohlson, VapeNorge, and JLCPCB. Shipping, customs and transaction fees added up to 856.91 NOK.

* Prices for the number of components in NOK.
** CGA6P3X7R1E226M250AE.
*** CGA2B3X7R1E104K050BB.

Component cost: 2031.45 NOK

Total cost: 2888.36 NOK

Modeling, Development, and Design of an Inverted Pendulum System

NTNU
Norwegian University of Science and Technology

Ivo Tredal [1] Bjørn K.T. Solheim [2] Ludvig Vartdal [1] Leidulv Tønnesland [1]

[1]Department of Engineering Cybernetics [2]Department of Electronic Systems

Introduction

The purpose of the project is to develop a mechatronic system that functions as a learning platform for experimenting with both basic and advanced concepts in the field of control engineering. This thesis focuses on the reaction wheel pendulum, more specifically the modeling, development and design of an inverted pendulum actuated by two perpendicular reaction wheels, with the overarching objective of developing an learning platform for control systems. The task at hand is to design an electronic system that enables the ability to control actuators and read sensor data, as well as the design and construction of the physical pendulum and the reaction wheels.

System Dynamics

In the case where the pendulum is restricted to only rotate about a single axis, the generalized coordinates are $\mathbf{q} = \begin{bmatrix}\theta_p & \theta_w\end{bmatrix}^\top$. The dynamics of the system is given by

$$\begin{aligned} \gamma\ddot{\theta}_p + J_w\ddot{\theta}_w &= \nu \sin\theta_p, \\ J_w\ddot{\theta}_p + (J_m + J_w)\ddot{\theta}_w &= \kappa v - B\dot{\theta}_w. \end{aligned} \tag{1}$$

For the unconstrained pendulum shown in Fig. 1, on the other hand, the generalized coordinates are $\mathbf{q} = \begin{bmatrix}\theta_p & \theta_w & \phi_p & \phi_w\end{bmatrix}^\top$.

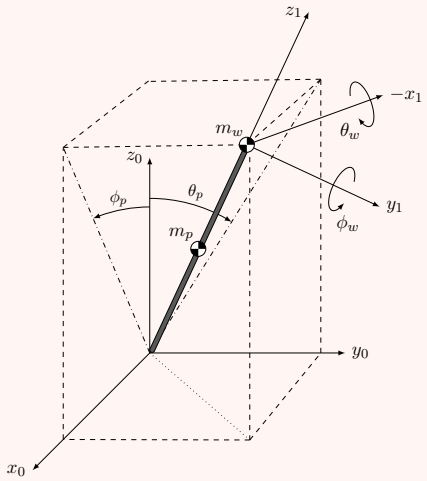


Figure 1. Dual-axis reaction wheel pendulum.

Here, the equations of motions are

$$\begin{aligned} (J_p + J_w + \mu\cos^2\phi_p)\ddot{\theta}_p + J_w\ddot{\theta}_w &= \eta\cos\phi_p\sin\theta_p + \dot{\theta}_p\dot{\phi}_p\sin 2\phi_p, \\ (J_p + J_w + \mu)\ddot{\phi}_p + J_w\ddot{\phi}_w &= \eta\cos\theta_p\sin\phi_p - \frac{1}{2}\mu\dot{\theta}_p^2\sin 2\phi_p, \\ J_w\ddot{\theta}_p + (J_m + J_w)\ddot{\theta}_w + B\dot{\theta}_w &= \kappa v_\theta, \\ J_w\ddot{\phi}_p + (J_m + J_w)\ddot{\phi}_w + B\dot{\theta}_w &= \kappa v_\phi. \end{aligned} \tag{2}$$

The parameters in the models originates from the kinetic and potential energy of the systems, as well as the DC motor dynamics. v, v_θ, and v_ϕ are the input voltages applied to the DC motor.

Electronic System Design

An electronic system is designed and implemented to be the foundation of the platform. The system incorporates an ESP32-family microcontroller, motors with encoders, motor drivers, an IMU, a custom buck converter circuit, a charging circuit, and thermistors connected to an analog multiplexer. A 3D model of the PCB is shown in Fig. 2.

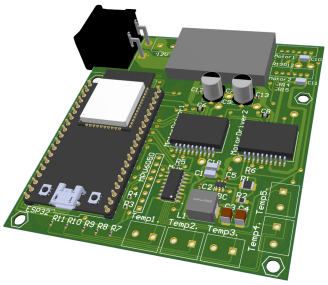


Figure 2. 3D model of the PCB.

All of these hardware components, along with logic-level converters, components for noise filtering, and a divided ground plane for power, analog signals, and digital signals are integrated into a custom PCB design. The

3D Printing

The mechanical components facilitates the connection between the electronics and the pendulum, as shown in Fig. 3. The system is realized through 3D-printing with PLA material.

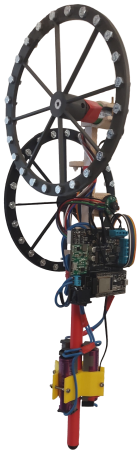


Figure 3. The dual-axis inverted pendulum system.

The mechanical system is split into three main parts; the frame, the inertia wheels, and the pendulum. The inertia wheels are designed to have modular weight, and the frame is designed such that different pendula are easily interchangeable.

Human Machine Interface

To monitor and operate the complete system a PC program, shown in Fig. 4, is built using Python. The program has a graphical user interface and uses Bluetooth to communicate with the electronic system.

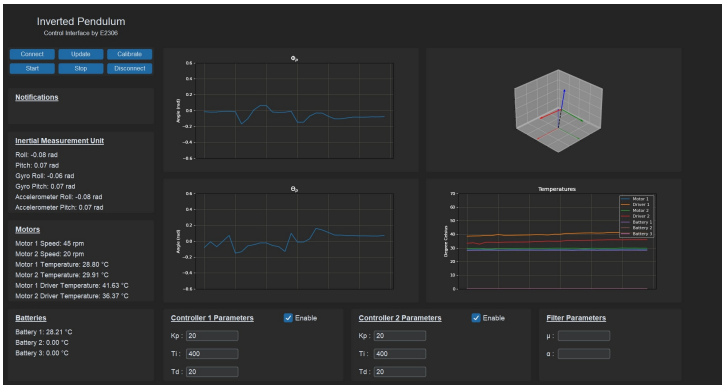


Figure 4. Developed PC program.

Graphs and values are updated in real-time, and PID values can be typed in for each controller, along with filter parameters related to the IMU.

Control System Design

The control figuration implemented in the microcontroller is depicted in Fig. 5.

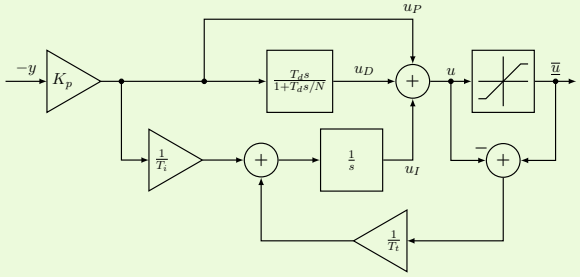
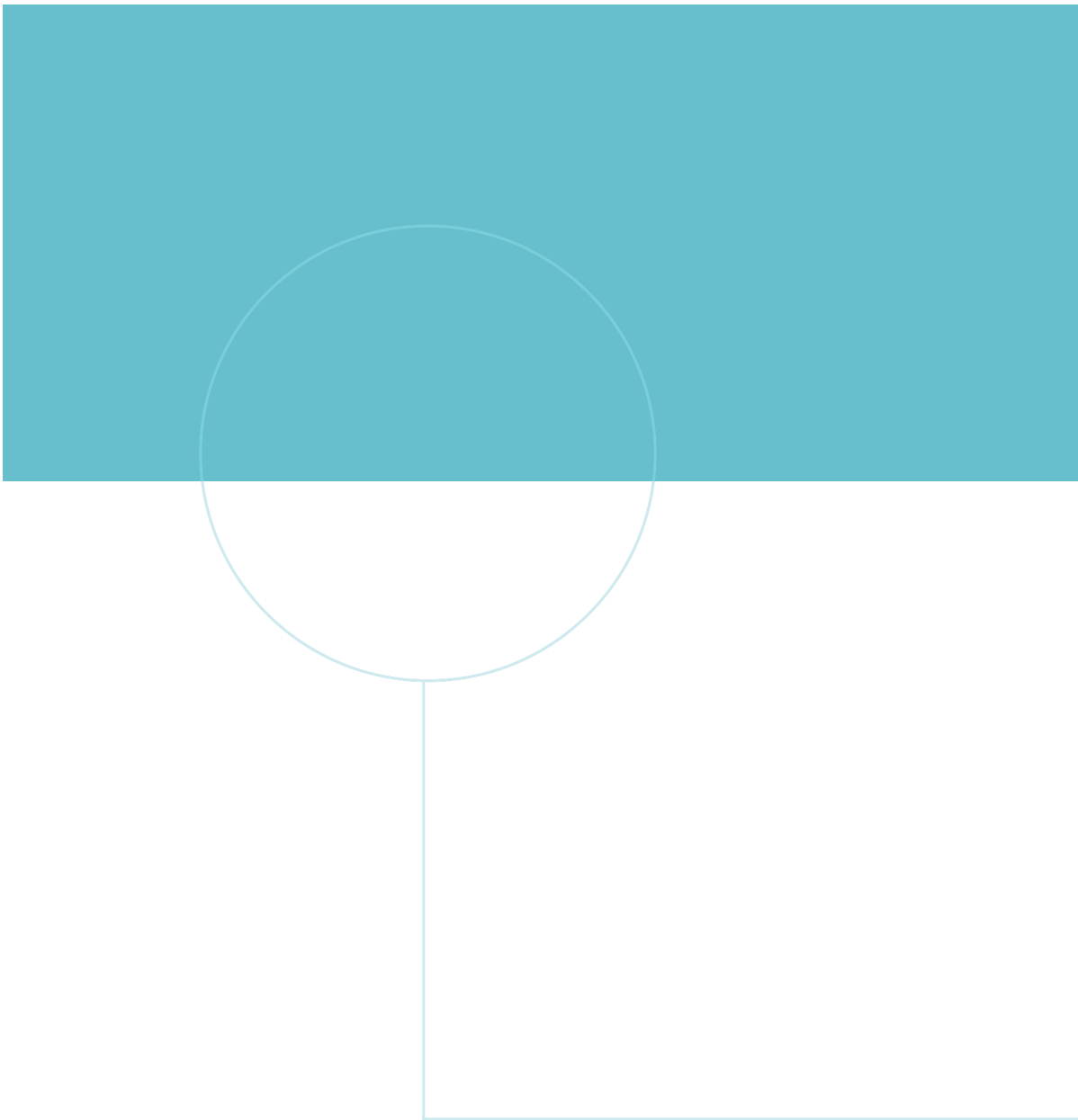


Figure 5. PID controller with back-calculation.

Here, the derivative term is approximated to a high-pass filter in order to obtain causality, and back-calculation is implemented in order to avoid integrator windup.



NTNU

Norwegian University of
Science and Technology