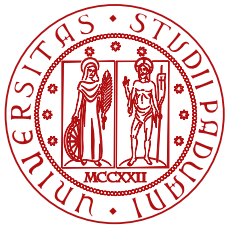


Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Bachelor Thesis in Electronic Engineering

Re-design of a magnetic levitation platform: from an early prototype to a working system

Candidate

Alberto Morselli

Student ID 1219601

Supervisor

Prof. Damiano Varagnolo

University of Padua

Co-supervisor

PhD Candidate Hans A. Engmark

NTNU - Norwegian University of Science and Technology

Academic Year 2021/2022

14 November 2022

Abstract

This work documents the modelling and re-designing of a magnetic levitation (maglev) platform intended to become a take-home lab for students interested in control engineering. The platform, initially designed by a team of Norwegian students, presented a series of inefficiencies and limitations that this work seeks to remove. Thus, the final goal of this project is to fully develop and test an improved maglev platform.

The thesis introduces both the theoretical framework behind the modeling of the system, and the specific frameworks for designing and analysing the various electronic subsystems that were deemed to be improvable. In doing so, this work discusses those subsystems, lists a set of potential solutions together with their pros and cons, and selects potential choices one may take, given the techno-economic constraints surrounding the project. In addition, practical implementation issues are discussed, giving in this way an overview of which parts of the systems should be further analyzed and fine tuned before arriving at a system that is ready for production.

The most challenging part of this project has been entering into it without clear guidelines of the design choices made by the previous team. It is precisely for this reason that this work does not just aim at reporting on the project itself, but also at offering documentation that is as clear as possible, intended to help a potential future team bringing the project to its next phase.

Con questo lavoro si vuole documentare la modellizzazione e la riprogettazione di una piattaforma a levitazione magnetica con l'obiettivo che possa divenire in futuro un take-home lab di controlli automatici. Il prototipo, originariamente progettato da un team di studenti norvegesi, presenta però una serie di imprecisioni e limitazioni che, con questo lavoro, si intende eliminare. L'obiettivo finale del progetto è perciò quello di arrivare allo sviluppo e al collaudo di un sistema maglev migliorato.

La tesi introduce sia il contesto teorico alla base della modellizzazione del sistema, sia i contesti specifici per l'analisi e la progettazione dei vari sottosistemi elettronici ritenuti perfezionabili. Nel fare ciò vengono prima discussi tali sottosistemi, quindi elencate una serie di potenziali soluzioni con i relativi pro e contro e infine individuata una soluzione ottimale in base ai vincoli tecnico-economici del progetto. Vengono inoltre descritti i problemi di natura pratico-implementativa che potrebbero sorgere, fornendo così una panoramica di quali elementi del sistema andrebbero ulteriormente discussi e ottimizzati prima di arrivare ad un'ipotetica fase di produzione.

La parte più impegnativa è stata entrare nell'ottica del progetto senza disporre di chiare linee guida sulle scelte progettuali adottate dal precedente team. Proprio su questo motivo si basano le linee esplicative di questo lavoro di tesi, che non vuole essere soltanto un report del lavoro fatto, ma che vuole proporsi come un'effettiva documentazione in grado di aiutare il più possibile un potenziale futuro team a condurre il progetto verso la sua fase successiva.

Contents

List of Figures

List of Tables

List of Code snippets

List of Acronyms

AC/DC Alternate Current - Direct Current
ADC Analog to Digital Converter
DC Direct Current
dof degrees of freedom
ic integrated circuit
in-amp intrumentation amplifier
levmag levitating magnet
LQR linear-quadratic regulator
maglev magnetic levitation
MIMO Multiple-Inputs Multiple-Outputs
nl non-linear
op-amp operational amplifier
pcb printed circuit board
PID Proportional-Integral-Derivative
PWM pulse width modulation
RGA Relative Gain Array
SISO Single-Input Single-Output
SMD surface mounted chip
vref voltage reference

1 Introduction

1.1 Background

Magnetic levitation is a highly advanced technology used in a variety of different applications. It is used today in several different areas, such as transportation systems (maglev trains), wind turbines, civil engineering (elevators), toys and so on.

Although the first magnets were discovered by the Ancient Greeks, it was not until 1842 that Samuel Earnshaw's theorem proved conclusively that it is not possible to levitate stably using only static paramagnetic fields. The forces acting on any magnetic object in any combination of gravitational, electrostatic, and magnetostatic fields will make the object's position unstable along at least one axis. For example, a maglev system with two simple dipole magnets repelling each other is highly unstable, since the magnet being levitated can slide sideways or flip over, and it turns out that no configuration of magnets can produce stable levitation [26].

For successful levitation and control of all 6 degrees of freedom (dof), 3 translational and 3 rotational, a combination of permanent magnets and electromagnets can be used [26]. This is usually achieved by actively controlling solenoids using sensor feedback.

The magnetic levitation platform designed and tested in this project is based on the same principles as the examples above, and aims to actively control a group of solenoids to stabilize a levitating magnet disc.

1.2 About the project

The maglev platform used in this project consists of 8 neodymium permanent magnets arranged in a ring pattern, a group of 4 solenoids driven by two L298N motor drivers to actively stabilize the levitating magnet (levmag), a sensor-board of hall-effect transducers to estimate its position and the electronic circuitry required to handle it all. Finally, the heart of the system: a Teensy 4.0 microcontroller mounting an ARM Cortex-M7 chip running at 600 MHz, one of the fastest on the market.

The maglev system, shown in figure (1.1), is arranged in a plexiglass frame on four levels:

1. the lowest level houses all the electronics;
2. the second level contains the 8 neodymium permanent magnets, necessary to provide a static magnetic field to stabilize the levmag on the z-axis;
3. the third one contains two pairs of solenoids, to respectively stabilize along x-axis and y-axis;
4. the top level is below where the levmag floats, and from which the sensor board emerges to estimate its position.

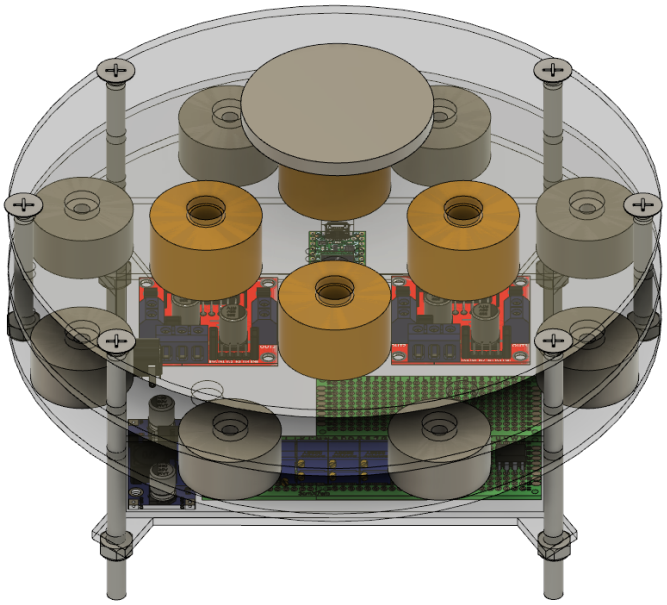


Figure 1.1: Fusion model of the maglev system designed

This project was originally designed by another team of Norwegian students and aims to become a take-home control engineering lab. To achieve this, several redesigns and analysis of the various subsystems are necessary to arrive to an optimized final product. Unfortunately the original platform suffered some physical damage and other electronic parts were completely burnt out as a result of a non-optimal design. The one documented in this work is the first of several potential future re-designs and required to entirely rebuild the model, and as such aims to:

- Re-design the electronic circuitry deemed improvable, such as the amplification stage, add a zener diodes protection stage and discuss the original configuration and placement of the sensors, as can be seen in chapter 4.
- Provide clean and accurate guidelines about how the project works, the design choices taken, and the elements that could be improved in a future redesign. This is intended to help the next team get into the project perspective quickly without spending too much time figuring out how the project works, and immediately focus on the improvements to be realized. In order to do so, this work provides precise schematics, rich documentation, consistency between the various stages (design, code, implementation) and guidance through the use of the modelling and simulation scripts.
- Provide cleaner, modular and commented code, along with testing routines, as shown in chapter 6.

1.3 Structure of the report

The report is organized in several main chapters, each one dealing with a different topic:

1. This first chapter is an introduction to the subject, that includes background information relevant to the task, gives an overview of the project, and also contains a description of the structure of the thesis.
2. Here, the mathematical model behind the system is introduced, along with a description of the physical laws and the variables that make up the model.
3. This chapter explains how the mathematical model presented in the previous chapter was implemented in MATLAB. It explain how to use the developed scripts to simulate and analyze the system and reports the results obtained.
4. This is the largest chapter, where the design phase of each electronic subsystem is presented in detail, along with the design choices taken, the pros and cons of every potential solution, and where finally a selection of an optimal choice is made given the techno-economic constraints surrounding the project. It also describes how the different subsystems are interconnected and gives an overview of the overall design.

5. This chapter presents the control strategies for the Proportional-Integral-Derivative (PID) control algorithm, intended to actively drive the solenoids. This chapter also discuss the difficulties encountered trying to tie together the MATLAB model to the actual system.
6. This chapter presents the code designed for the microcontroller, the digital choices taken and the routines designed precisely for the testing phase.
7. In this chapter, testing procedures for the several electronics subsystems are quickly presented.
8. Here we discuss potential improvements and future work that could be done to further analyze and improve the maglev system.
9. The last chapter contains conclusions on the results of the project.

This thesis does not intend to cover the basic fundamentals of electronics, physics, embedded programming and control systems in order to reserve more space for the treatment, discussion and application of these in our specific system. For example, it is not explained what a pulse width modulation (PWM) is, but rather why a specific switching frequency or resolution was used. For the less experienced reader the following links, organised by topic, provide sufficient background for an almost complete understanding of this thesis.

Physics		**Electronics**	
Newton-Eulers equations of motion	[27]	Hall-effect sensors	[8]
Lorentz force	[25]	Operational amplifiers	[11]
Biot-Savart law	[22]	Instrumentation amplifiers	[21]
Laplace force law	[22]	Potentiometers	[13]
Solenoids	[32]	Zener diodes	[20]
		Shottky diodes	[31]
		Buck converters	[23]

Embedded programming		**Control System**	
ADC	[2]	Control theory	[24]
PWM	[29]	State space model	[18]
		Transfer function	[9]
		Equilibrium	[7]
		Linearization	[6]
		PID controller	[28]
		Kalman Filter	[4]

Table 1.1: Suggested links for an exhaustive background

2

Model Description

The mathematical model is based on the existing model described in the internal document *Model Description: Magnetic Levitation System (R. Doshmanziari, H.A. Engmark, and K.T. Hoang, 2021)* [5]. However, the document models a magnetic levitation platform similar to the one used in this project, but with some differences. The paper in fact describes a system with a ring magnet instead of one with 8 permanent magnets, modelled as a bias current in the solenoids instead of a "permanent" solenoid (this will become clearer after reading section 2.1). Lastly, the paper describes a system that assumes full state measurements, which our system does not have. The measurements are instead the magnetic field in x-, y- and z-direction, measured by the hall-effect sensors placed on the sensor board. As a result, the full description adapted for our specific model is given here below. This chapter will give you the necessary theoretical framework to understand the MATLAB implementation of the model presented in chapter 3.

2.1 Model assumptions and simplifications

The maglev system is shown in figure (2.1). The system contains 8 permanent magnets and a set of four electromagnetic solenoids, whereas the object being lifted is a permanent disk magnet. Hereafter they will be referred to as the magnets, the solenoids and the levmag, respectively. The magnets provide most of the lift, whereas the solenoids provide stabilization [5].

The levmag dynamics can be fully described by the 12×1 state vector and a 4×1 input vector:

$$\eta = \left[x_m, y_m, z_m, \psi_m, \theta_m, \varphi_m, \dot{x}_m, \dot{y}_m, \dot{z}_m, \dot{\psi}_m, \dot{\theta}_m, \dot{\varphi}_m\right]^T \quad \in \mathbb{R}^{12\times 1}$$
$$u = \left[I_s^1, I_s^2, I_s^3, I_s^4\right]^T \quad \in \mathbb{R}^{4\times 1}$$

"where $\left[x_m, y_m, x_m\right]^T$ and $\left[\psi_m, \theta_m, \varphi_m\right]^T$ are the Cartesian coordinates and the Euler rotation angles of the center of mass of the levmag, and I_s^j are the currents in each solenoid" [5].

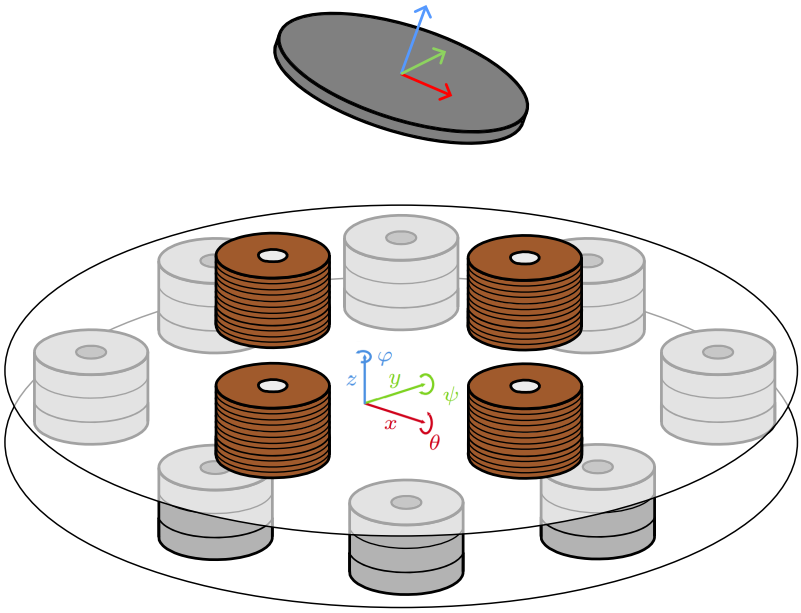


Figure 2.1: Illustration of the maglev system

"Throughout the modelling, only the magnetic and mechanical properties are considered; it is assumed that other external effects are negligible" [5]. Accordingly, the following modelling simplifications are made:

1. the magnets are modelled as solenoids with a constant current;
2. the levmag is modelled in the same way;
3. the solenoids (and everything else modelled as such) are modeled as thin wire loops.

"The first and second simplification are reasonable when the distance between the magnets is sufficiently large. Moreover, a permanent magnet is magnetically equivalent to an electromagnet when only considering the external field, justifying the third simplification". [5]

2.2 Physical laws

As in the paper, the levmag movement is modeled using the Newton-Euler equations of motion, described below

$$\begin{bmatrix} F_b(\eta, u) \\ \tau_b(\eta, u) \end{bmatrix} = \begin{bmatrix} m\mathbf{I}_3 & 0 \\ 0 & \mathcal{I} \end{bmatrix} \cdot \begin{bmatrix} a \\ \alpha \end{bmatrix} + \begin{bmatrix} 0 \\ \omega \times \mathcal{I}\omega \end{bmatrix} \tag{2.1}$$

where m is mass, a is linear acceleration ($[\ddot{x}_m, \ddot{y}_m, \ddot{z}_m]^T$ in the space vector representation), $F_b(\eta, u)$ is the total force acting on the levmag, $\mathcal{I}$ is an inertia matrix, α is angular acceleration ($[\ddot{\psi}_m, \ddot{\theta}_m, \ddot{\varphi}_m]^T$ in the space vector representation), ω is angular velocity ($[\dot{\psi}_m, \dot{\theta}_m, \dot{\varphi}_m]^T$ in the space vector representation) and τ_b is the torque acting on it.

The magnetic forces $F_b(\eta, u)$ for current carrying wires (keep in mind that we are modelling the levmag as a solenoid, so as current carrying wire loops) are modelled using Laplace force law:

$$F_b(\eta, u) = I \int_C \mathrm{d}\ell \times \mathbf{B}(p) \tag{2.2}$$

where $\mathbf{B}(p)$, the magnetic field value in a point p of the wire C, is calculated using Biot-Savart law:

$$\mathbf{B}(p) = \frac{\mu_0}{4\pi} I \int_C \frac{d\ell \times (p - \ell)}{|(p - \ell)|^3} \tag{2.3}$$

Here, I is current in a wire, $d\ell$ is a differential vector along the line (wire) C, $\mathbf{B}$ is the magnetic field, μ_0 is the permeability of air, p is a point in Cartesian coordinates, and ℓ is point on the line C.

2.3 Solution to Biot-Savart and Laplace force law

As reported by the paper, the Biot-Savart law and the Laplace force law include integrals that in general can not be solved analytically. However, González and Cárdenas [1] present the following solution of the Biot-Savart law when the current

carrying wire is assumed to be a thin loop:

$$\begin{aligned}\mathbf{B}(\rho,z) =& \frac{\mu_0 I}{2\pi\sqrt{\left(\rho'+\rho\right)^2+(z-z')^2}}\cdot \\ &\cdot\left\{\frac{(z-z')}{\rho}\cdot\left[\frac{\rho'^2+\rho^2+(z-z')^2}{\left(\rho-\rho'\right)^2+(z-z')^2}\mathrm{E}(k)-\mathrm{K}(k)\right]\boldsymbol{e}_\rho\right. \\ &\left.-\left[\frac{\rho^2-\rho'^2+(z-z')^2}{\left(\rho-\rho'\right)^2+(z-z')^2}\mathrm{E}(k)-\mathrm{K}(k)\right]\boldsymbol{e}_z\right\}\end{aligned} \tag{2.4}$$

where

$$k=\frac{4\rho'\rho}{\left(\rho'+\rho\right)^2+(z-z')^2} \tag{2.5}$$

Here, ρ, ϕ and z are the cylindrical coordinates of the point of interest, and ρ' and z' are the radius and the height of a current loop centered around the z-axis. $K(k)$ and $E(k)$ are the complete elliptical integrals of the first and second kind, respectively. The limit case of equation (2.4) when $\rho \longrightarrow 0$ is also presented as:

$$\lim_{\rho\to 0}\mathbf{B}(\rho,z)=\mathbf{B_z}(z)=\frac{\mu_0\rho'^2 I}{2\left[\rho'^2+(z-z')^2\right]^{3/2}} \tag{2.6}$$

"To approximate the solution of Laplace force law for a current carrying loop in a magnetic field, Wang and Ren [19] suggest using discretization to divide the loop into n linear segments. Assuming uniform field strength along each segment, this allows the Laplace force law to be represented as a sum instead of an integral (see (2.12) and (2.13). Wang and Ren show that this approximation achieves less than 0.07% relative error for $n \geq 100$" [5].

2.4 Complete model description

As stated in the paper and suggested by Wang and Ren [19] the levmag is discretized into n points p_i, using the discretization

$$p_i = R\left(\psi_m,\theta_m\right)\cdot\begin{bmatrix} r_m\cos\left(i/n\cdot 2\pi\right) \\ r_m\sin\left(i/n\cdot 2\pi\right) \\ 0\end{bmatrix}+\left[x_m,y_m,z_m\right]^T \qquad i=1,\ldots,n \tag{2.7}$$

where r_m is the radius of the levmag, and $[x_m, y_m, z_m]^T$ the center, whereas $R\left(\psi_m, \theta_m\right)$ is the rotation matrix in the specific case where $\varphi_m = 0$ (due to the symmetries in the system) used to perform a rotation in Euclidean space (see [30] and [14] for more details).

$$R(\psi, \theta) = \begin{bmatrix} \cos(\theta) & \sin(\psi)\sin(\theta) & \cos(\psi)\sin(\theta) \\ 0 & \cos(\psi) & -\sin(\psi) \\ -\sin(\theta) & \cos(\theta)\sin(\psi) & \cos(\psi)\cos(\theta) \end{bmatrix} \tag{2.8}$$

Using the simplification for the Biot-Savart law, the magnetic field is computed at a point $p_i = [x, y, z]^T$ in Cartesian coordinates as

$$\begin{aligned} \mathbf{B}_x(p_i) &= \mathbf{B}(\rho, z) \cdot \cos\phi \\ \mathbf{B}_y(p_i) &= \mathbf{B}(\rho, z) \cdot \sin\phi \\ \mathbf{B}_z(p_i) &= \mathbf{B_z}(z) \end{aligned} \tag{2.9}$$

using the coordinate transformation from the cylindrical ones

$$\begin{aligned} \rho(p_i) &= \sqrt{x_i^2 + y_i^2} \\ \phi(p_i) &= \arctan\frac{y_i}{x_i} \\ z(p_i) &= z_i \end{aligned} \tag{2.10}$$

With the principle of superposition, the total magnetic vector field $\overrightarrow{\mathbf{B}}_i(\eta, u, p_i)$ produced by all the solenoids and all the permanent magnets in a point p_i becomes

$$\overrightarrow{\mathbf{B}}_i(\eta, u, p_i) := \sum_{h=1}^{4} \begin{bmatrix} \mathbf{B}_x\left(p_i - p_s^h\right) \\ \mathbf{B}_y\left(p_i - p_s^h\right) \\ \mathbf{B}_z\left(p_i - p_s^h\right) \end{bmatrix} + \sum_{k=1}^{8} \begin{bmatrix} \mathbf{B}_x\left(p_i - p_m^k\right) \\ \mathbf{B}_y\left(p_i - p_m^k\right) \\ \mathbf{B}_z\left(p_i - p_m^k\right) \end{bmatrix} \tag{2.11}$$

where p_s^h is the Cartesian coordinates of the center of the h'th solenoid and p_m^k the center of the k'th permanent magnet. Then, according to the Laplace force law, the total magnetic force on the levmag from the equation (2.2) is now approximated by a sum of terms:

$$F_b(\eta, u) = I \int_C \mathrm{d}\ell \times \mathbf{B}(p) \approx I_m \sum_{i=1}^{n} \vec{\ell}_i \times \overrightarrow{\mathbf{B}}_i(\eta, u, p_i) \tag{2.12}$$

where

$$\vec{\ell}_i = \frac{p_{i+1} - p_{i-1}}{4}$$

and $\overrightarrow{\mathbf{B}}_i(\eta, u, p_i)$ the total magnetic field vector produced by all the solenoids and magnets found with equation (2.11). Similarly, the magnetic torque is approximated by

$$\tau_b = I_m \sum_{i=1}^{n} \vec{r}_i \times \vec{\ell}_i \times \overrightarrow{\mathbf{B}}_i(\eta, u, p_i) \tag{2.13}$$

where

$$\vec{r}_i = p_i - \left[x_m, y_m, z_m\right]^T$$

The moment of inertia of the levmag is

$$\mathcal{I} = \text{diag}\left(\left[\frac{1}{4}m \cdot r_m^2, \frac{1}{4}m \cdot r_m^2, \frac{1}{2}m \cdot r_m^2\right]\right) \tag{2.14}$$

where m is the mass of the magnet.

"The entire system can be compactly represented by the state space representation" [5]

$$\begin{aligned} \dot{\eta} &= A\eta + B\phi(\eta, u) \\ y &= h(\eta, u) \end{aligned} \tag{2.15}$$

where

$$A = \begin{bmatrix} O_{6\times6} & I_6 \\ O_{6\times6} & O_{6\times6} \end{bmatrix} \qquad B = \begin{bmatrix} O_{6\times6} \\ I_6 \end{bmatrix}$$

where I_n and $O_{n\times m}$ are identity and zero matrices, so that

$$\begin{aligned} \dot{\eta}\,[1-6]^T &= \left[\dot{x}_m, \dot{y}_m, \ldots, \dot{\varphi}_m\right]^T = \eta\,[7-12]^T \\ \dot{\eta}\,[7-12]^T &= \left[\ddot{x}_m, \ddot{y}_m, \ldots, \ddot{\varphi}_m\right]^T = \phi(\eta, u) \end{aligned}$$

with $\phi(\eta, u)$ and $h(\eta, u)$ non-linear (nl) functions that are respectively

$$\begin{aligned} \phi(\eta, u) &= \begin{bmatrix} mI_3 & O_{3\times3} \\ O_{3\times3} & \mathcal{I} \end{bmatrix}^{-1} \begin{bmatrix} F_b(\eta, u) \\ \tau_b(\eta, u) \end{bmatrix} - \begin{bmatrix} O_{2\times1} \\ g \\ O_{3\times1} \end{bmatrix} \\ h(\eta, u) &= \left[\overrightarrow{\mathbf{B}}(p_{s_1}), \overrightarrow{\mathbf{B}}(p_{s_2}), \ldots, \overrightarrow{\mathbf{B}}(p_{s_j})\right] \end{aligned} \tag{2.16}$$

where g is the gravitational force subtracted from $\dot{z}_m$ and $h(\eta, u)$ represent a vector concatenation of the magnetic field measured by the j'th sensor according to where it's placed, so the dimensions of y depend on the number of sensors used (it will be a 3xj vector).

The next chapter will cover the implementation of this mathematical model in a MATLAB environment, the simulation scripts created with the model here explained and how to use them to analyze our system.

3

MATLAB implementation

The mathematical model described just before has been implemented in a MATLAB environment to analyze and simulate the system. A GitHub repository [1] has been made, containing all the original code (adapted for our project) written by the authors [5] as well as the code written specifically for this project. This chapter will go through how the code works, its usage, and its applications.

3.1 Class definition

The class `maglevSystem` has methods for computing the magnetic field of the solenoids and the permanent magnets modelled as such, the force and torque on the levitating magnet, the state derivative $\dot{\eta}$, and the output y read by the sensors. An object of this class can be used to simulate the trajectory of the levmag, but this is made possible only after modelling all the elements of the system that we want to simulate. This modelling process includes:

- the current that flows in each of the four solenoids;
- the physical parameters of each element (inner radius, outer radius, mass, etc.);
- the geometric layout of the various elements (distance from the center, placement coordinates, number of elements present in the system);
- decide the equivalent solenoid modelling parameters for the magnets;

[1] `https://github.com/albertomors/maglev22`

- the equivalent current that correctly model each permanent magnet in its corresponding solenoid. In order to correctly simulate the magnets (in terms of the electromagnetic forces generated), it is necessary to match the number of wire loops with the equivalent current of its corresponding solenoid.

3.2 Quick overview

After having found all these values, an object of this class might be created with these simple commands:

```
approximationType = 1; % 0=fast, 1=accurate
load('params.mat');

% setting initial conditions
x0 = zeros(12,1);
x0(3) = 0.0576; % z-coordinate

sys = maglevSystem(x0, params, approximationType);
hold on; draw(sys, 'fancy'); hold off;
```

Listing 3.1: Creating a maglevSystem object

that displays the following image of the system created with the params used

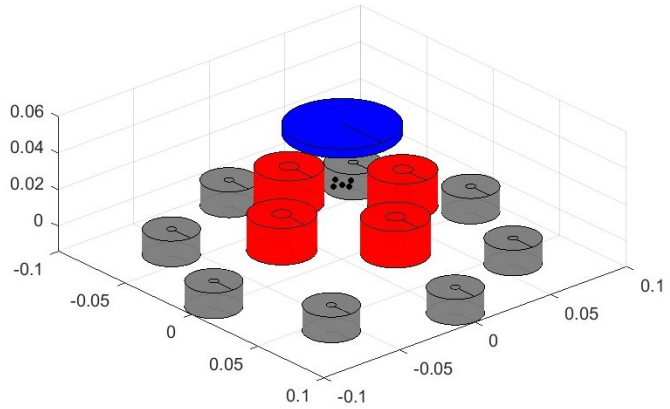
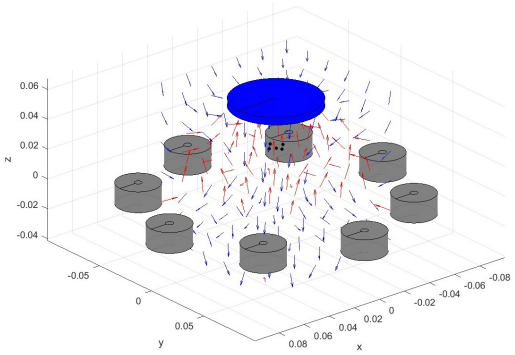


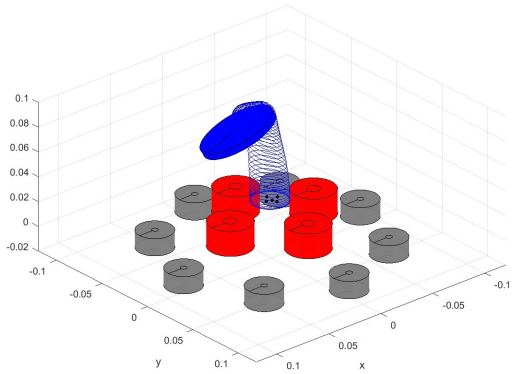
Figure 3.1: Drawing method of the maglevSystem class (the origin is aligned with the top of the permanent magnets)

As seen in the code snippet above, the class has 2 different modes of simulating and computing the forces: one fast and one accurate (which require a longer computation time), described in section 3.3. The `params` variable contains all

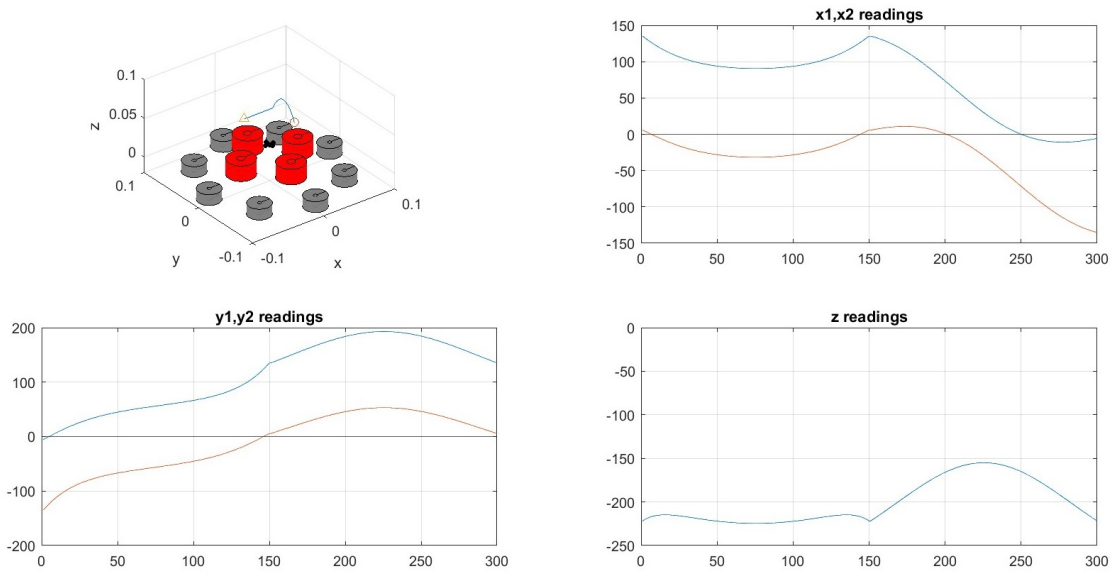
relevant parameters presented in section 3.1, whereas $x0$ are the initial conditions of the space vector representing the levmag state. Along with drawing the system, the class can also compute $\dot{\eta}$ with the function `sys.f(x0,u)`, where u is the 4×1 vector representing the values of the currents flowing in the solenoids, and lastly with the function `sys.h(x0,u)` you can have access to the sensor readings. With these methods and a bit of coding skills it's possible to do a lot of things, such as:



(a) Plot the overall force acting on the magnet according to its position



(b) Simulate the levmag evolution according to the initial conditions



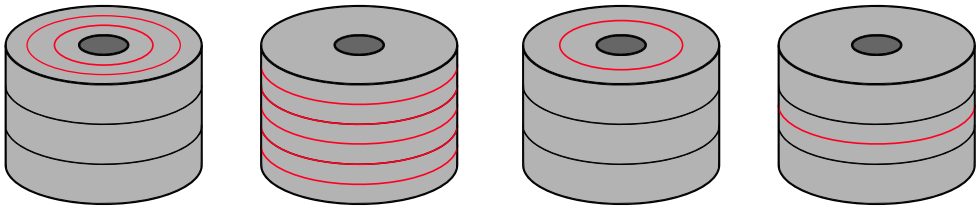
(c) Plot the sensor readings while moving the magnet along a particular path

All the code of the class `maglevSystem`, along with the sub-class and the side methods used by it can be found in the Github repository at `https://github.com/albertomors/maglev22`.

3.3 Accurate vs fast simulations

As mentioned earlier, the `maglevSystem` class has two different modes to approximate the magnetic field: one fast and one accurate that require a longer computation time. When using the accurate mode each winding is treated as separate wire loops distributed across the volume of the solenoid, with nr concentric loops in width and nh loops in height as shown in figure (3.3a). This is a more time consuming way to simulate the system as the magnetic field has to be computed separately for each winding, but it also gives a more accurate representation of the actual system if the number of windings set is consistent with the real system.

Instead, the fast mode models each solenoid as one discretized wire loop centered in the center of mass, as can be seen in figure (3.3b). The computed force and torque are then scaled up by the number of windings in the solenoid, where $windings = nr \cdot nh$. The magnetic field here has to be computed just one time, but gives slightly less accurate results. Both methods then discretize the loop in nl linear segments, as suggested by Wang and Ren [19].



(a) Accurate (b) Fast

Figure 3.3: Comparison between the two approximation methods (In this example $nr = 2$ and $nh = 5$).

3.4 Modelling the physical components

After this short introduction, we'll now go through the actual physical modelling process of the components. This is done in the most part by measuring the physical dimensions of the permanent magnets, the levitating magnet and the solenoids, their weight and so on. All these values can be set, edited as required and stored in the `params.mat` file. An exhaustive description of `params` can be found in the appendix E. After these preliminary measures you can move onto the stage of finding the right value of the equivalent current. First of all has to be

decided the values of nr, nh and nl for each of the magnets (or used the actual values for the solenoids). Setting a high value of windings lead to a more reliable simulation and more accurate results, but also requires a longer computation time. These modelling parameters must be decided with a trade-off between these two things. After these we can move on finding the right amount of current such that the field strength of the modelled solenoids matched the field strength of the real magnets. Of course, this value is strongly dependent by the $nr \cdot nh$ parameters set (nl too), so changing the windings parameters requires to compute again the value of the equivalent current.

3.4.1 Permanent magnets

As first will be presented the physical parameters of the permanent magnets (both measured and set), present in the `params.mat` file under the `magnets` keyword. This is shown in table 3.1.

Variable	Description	Value	Unit
ri	Inner radius	0.0025	m
ro	Outer radius	0.0145	m
h	Height	0.0140	m
offset	Placement offset angle[2]	0	degrees
N	Number of permanent magnets	8	
R	Radius of permanent magnet circle	0.085	m
nr	Number of rings in radius	15	
nh	Number of rings in height	15	
nl	Number of discretizations	100	
I_eq	Equivalent current	to be computed	A

Table 3.1: Specifications for the neodymium permanent magnets

Since the permanent magnets are grouped in stacks of three, it is natural to model a stack as a single solenoid. To find the equivalent current, a simpler maglev system is used: it consists of two stacks of three neodymium magnets where one was fastened to the surface, acting as the permanent magnet, while the other was suspended above, acting as the levitating magnet (see figure 3.4). There will be a height where free-levitation occurs (in an equilibrium point along the

[2]additional variable to rotate the magnets configuration by a pre-set offset (if present)

z-axis) and the corresponding repulsive electromagnetic forces and gravity force balance each other.

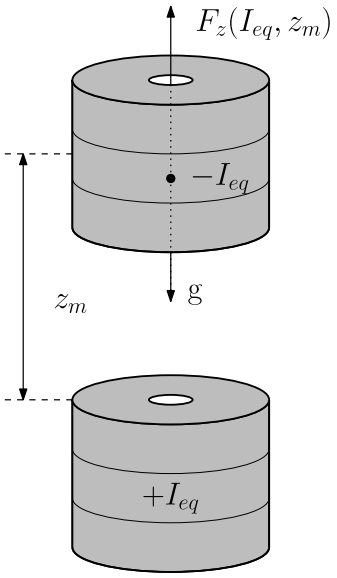


Figure 3.4: Measuring levitation height between two magnets

By measuring experimentally the height where free-levitation occurs, the only unknown is the two currents (equals and opposite) needed to create the force F_z. This current, in fact, flows in one direction on the lower magnet (establishing a north pole on its top) and in the opposite direction on the levitating one (establishing a south pole on its top and a north pole on its bottom) such that the two magnets repel each other.

$$\frac{F_z(I_{eq}, z_m)}{m} - g = 0 \tag{3.1}$$

The value `meas_height` is measured to be $\approx 0.0765\ m$. By simulating an equivalent system in MATLAB it is possible to find the current numerically, using the `maglevSystem` class to graph $\dot{z}_m$, while varying I_{eq}, searching where it crosses zero.

Two scripts has been developed appositely to do this, to be used consecutively. The first one, `eq_curr_neo.m` is a trial-error tool to approximately guess the right value by successive attempts. The code, after overriding some parameters to simulate this new specific system (see listing 3.2), set the equivalent currents of the magnets and graph $\dot{z}_m$ varying z_m in search of the value that approximately cross the zero in `meas_height`.

```
...
%% Overwrite solenoid parameters
params.solenoids.ri = 0;
params.solenoids.ro = 0; % ...and don't draw it

%% Overwrite neodymium magnets parameters as a single centered one
params.magnets.N = 1; % just one...
params.magnets.R = 0; % ...placed at center

%% Overwrite floating magnet parameters as a neodymium magnet
params.levitatingmagnet = params.magnets;
...
params.magnets.I = test_current;
params.levitatingmagnet.I  = -test_current; % same current but repelling
sys = maglevSystem(x0, params, approximationType);
...
for i = 1:length(Zrs)
    %simulate system evolution in the height range
    temp = sys.f([0,0,Zrs(i),zeros(1,9)]',[0 0 0 0]');
    Fzs(i) = temp(9); %zm_dot should = 0
end
...
plot(Zrs,Fzs);
...
```

Listing 3.2: Overriding the parameters to simulate the new maglev system

The tool provides the following plots shown in figure (3.5).

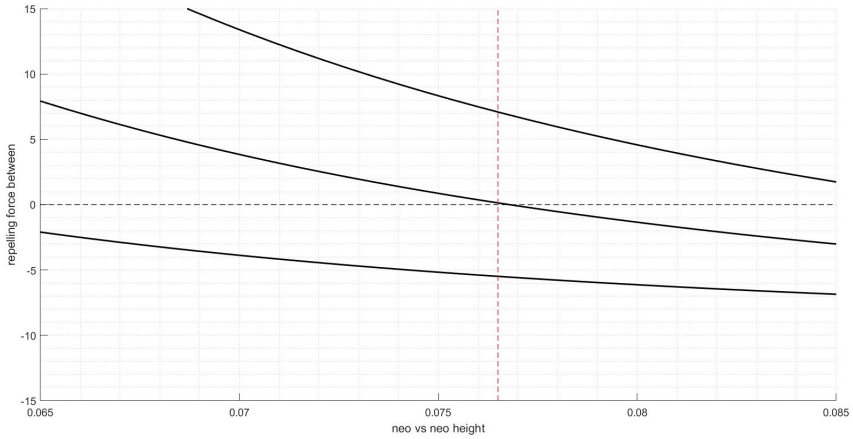


Figure 3.5: Results from the script eq_curr_neo.m

When the value set is sufficiently close to the right value the code saves this last value in a support internal file called `results.m`. Launching now the second script, `eq_curr_neo_AUTO.m`, will automatically search in a closer range with higher accuracy the best current value that better fit the desired results. The

searching parameters are adjustable in the code headers as desired to allow recursive script usage every time with higher accuracy in a smaller range. This is shown in listing 3.3.

```
1 ...
2 %% Searching parameters [EDIT HERE]
3 ...
4 current_range = 10; % Ampere around stimated correct value
5 min_current = last_curr-current_range/2;
6 max_current = last_curr+current_range/2;
7 current_steps = 256;
8 ...
```

Listing 3.3: Adjusting the searching parameters for recursive usage

The following commands (see figure 3.6) shows how the plots of figure (3.5) were obtained. As can be seen the third attempt was considered sufficient precise to launch the relative automatic searching tool.

```
>> eq_curr_neo
approxType [0/1]> 0
current [-1 when happy]> 89
current [-1 when happy]> 176
current [-1 when happy]> 135
current [-1 when happy]> -1
>> eq_curr_neo_AUTO
approxType [0/1]> 0
LAST: 135.000000 A
NOW : 134.193548 A
```

Figure 3.6: MATLAB automatic tool to search for the equivalent current

The fast approximation type results in a value of 134.1935 A whereas the accurate one results in 91.2903 A. The difference here is considerable and should not be underestimated.

3.4.2 Levitating magnet

As before, the levitating magnet specifications are presented in table (3.2), found in `params.mat` under `levitatingmagnet` keyword.

The same techniques used just before can be used to model the equivalent current of the levitating magnet. The system to use this time is the one shown in figure (3.7). Again, the height at which to search for the balance of forces has to be measured experimentally. This time `meas_height` was estimated to be

Variable	Description	Value	Unit
ri	Inner radius	0	m
ro	Outer radius	0.03	m
m	Mass	0.072	kg
nr	Number of rings in radius	30	
nh	Number of rings in height	5	
nl	Number of discretizations	100	
I_eq	Equivalent current	to be computed	A

Table 3.2: Specifications for the levitating magnet

0.074 m. Although the dof of this system are now three (current of the magnet I_{mag}, current of the levmag I_{eq} and height z_m) the value of I_{mag} was estimated with the procedure just seen in section 3.4.1 and can be taken for known.

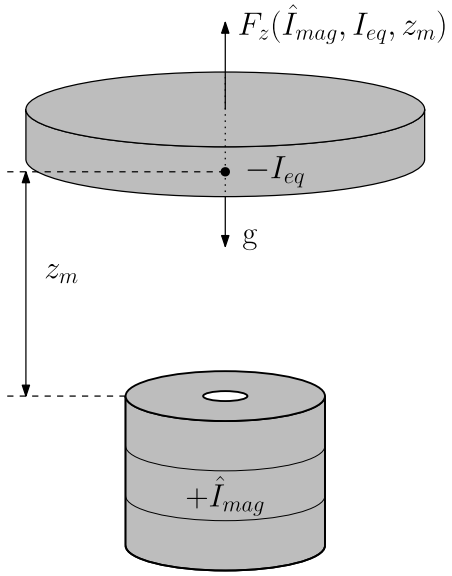


Figure 3.7: Measuring levitating height between a magnet and the levmag

There is only one important thing to keep in mind: the same approximation conventions must be kept to preserve consistency with the modelling process, due to the strongly dependency of the model by the approximation type used. So if we now want to model the levmag with a fast approximation type we have to use the I_{fast} value (134.1935 A) of the magnet's equivalent current, if instead we want to use the accurate one we have to use the I_{acc} value (91.2903 A). This is done by the script that initially loads from `results.m` the right values as can be seen in listing 3.4.

```
1 ...
2 if(approximationType == 0)
3     params.magnets.I = results.neo_vs_neo.curr_fst;
4 else
5     params.magnets.I = results.neo_vs_neo.curr_acc;
6 end
7 ...
```

Listing 3.4: Keeping consistency between the modelling process

whereas for the rest of the script operates in the same way of the ones presented before, leading to a value of 50.4839 A following the fast approximation method and to 39.4839 A following the accurate one.

3.4.3 Solenoids

Here is presented the specifications for the solenoids, shown in table (3.3). Here the values nr and nh are not modelling values set by us, but the real number of coils of the solenoids. These values can be found in `params.mat` under `solenoids` keyword.

Variable	Description	Value	Unit
ri	Inner radius	0.0050	m
ro	Outer radius	0.0175	m
h	Height	0.0200	m
N	Number of solenoids	4	
R	Radius of solenoid circle	0.040	m
zs	Offset along z-axis[3]	0.023	m
nr	Number of rings in radius	20	
nh	Number of rings in height	50	
nl	Number of discretizations	100	

Table 3.3: Specifications for the solenoids

3.4.4 Sensors

The `params.mat` file, finally contains information about the hall-effect sensors placement, that in the code has been implemented as measure of the magnetic

[3]Keep in mind that solenoids are placed one level above the permanent magnets. zs is the distance between the two tops (including thickness of the plexiglass level between)

field in the point where it's placed the real sensor and are drawn as black dots as can be seen in figure (3.8). As will be seen in chapter 4.3 it was decided to use a total of 5 sensors: two for x-axis, two for y-axis and one for z-axis arranged in a cross pattern with the z-sensors facing the top to improve the symmetry of the measurements.

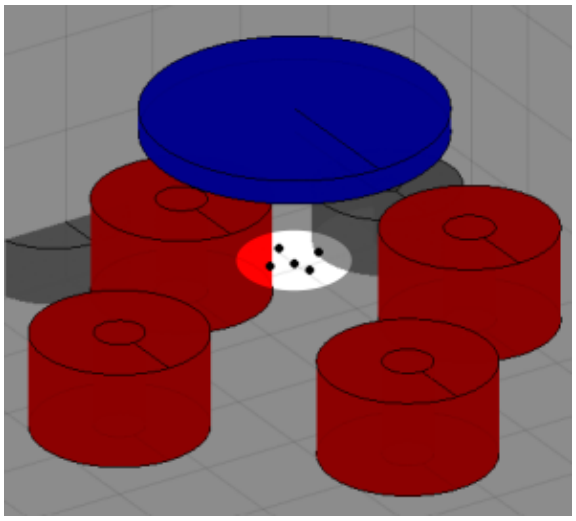


Figure 3.8: Hall-effect sensors in MATLAB

Inside the MATLAB implementation, these "virtual" sensors return a vector value of the magnetic field in (x, y, z), whereas in the reality return only the value of the magnetic field where they are pointing towards. The `maglevSystem` method `sys.h(x0,u)` then returns a vector concatenation of the measurements of these 5 sensors, which in order are: (x_1, x_2, y_1, y_2, z). So in the simulations, the values not corresponding to the purpose of the sensor has to be trashed ($y-, z-$ measures of the x-sensors and so on). Table (3.4) reports the coordinates of the 5 sensors.

Variable	**Description**	**Value** $(\mathbf{x_1}, \mathbf{x_2}, \mathbf{y_1}, \mathbf{y_2}, \mathbf{z})$	**Unit**
x	x-coordinates	[5.5,-5.5,0,0,0]	mm
y	y-coordinates	[0,0,5.55,-5.5,0]	mm
z	z-coordinates	[2.75,2.75,2.75,2.75,2.65]	mm

Table 3.4: Specifications for the sensors

The modelling phase of the various elements is now completed. We will now start to search for an operating equilibrium point of the entire system, and use some developed tools to analyze it.

3.5 Geometrical layout

The positioning of the magnets and solenoids was decided by the previous team and, as they reported, the optimal choice was made after a careful analysis of the problem. The radius of the circle affects both the stability of the system and the flexibility of the levitating magnet, i.e. the force required to move the magnet away from the equilibrium point and back towards it. A lower radius means a more rigid equilibrium, since the force of the permanent magnets is stronger with small deviations. Instead a higher radius produces a more flexible system, requiring less force from the solenoids to manipulate the position of the levitating magnet. However, a too high radius means that the system will no longer have an equilibrium. As was reported by them, in the end, a radius of 0.085 m was chosen for the magnets, and one of 0.040 m for the solenoids, as this configuration in theory allows for some movement in the z-direction without being close to instability. Furthermore, this configuration left room for a greater distance between the solenoids, which was assumed to make the control even more flexible. A discussion on the choice of this configuration used can be found in chapter 8.8.

3.6 Equilibrium point

Analysing if and where the system is in equilibrium is useful when deciding on an operating point for the controller. The system is in equilibrium if:

$$\dot{\eta} = A\eta + B\phi(\eta, u) = 0 \tag{3.2}$$

which for an autonomous system ($u = 0$), and using equations (2.15) and (2.16) becomes:

$$\left[\dot{x}_m, \dot{y}_m, \dot{z}_m, \dot{\psi}_m, \dot{\theta}_m, \dot{\phi}_m, \frac{F_x}{m}, \frac{F_y}{m}, \frac{F_z}{m} - g, \frac{\tau_\psi}{I}, \frac{\tau_\theta}{I}, \frac{\tau_\phi}{I}\right]^T = \mathbf{0} \tag{3.3}$$

Due to symmetries in the magnetic field, if:

$$x_m = y_m = \psi_m = \theta_m = \phi_m = 0 \tag{3.4}$$

then

$$F_x = F_y = \tau_\psi = \tau_\theta = \tau_\phi = 0 \tag{3.5}$$

By also assuming the following stationary conditions

$$\dot{x}_m = \dot{y}_m = \dot{z}_m = \dot{\psi}_m = \dot{\theta}_m = \dot{\phi}_m = 0 \tag{3.6}$$

the only non-zero element in (3.3) is $F_z/m - g$. Given the states in equation (3.4) and (3.6) with $u = \mathbf{0}$, F_z only varies with z_m. This means that if there exist an z_{m_eq} such that $F_z/m - g = 0$, then there exists an unstable equilibrium in (η_{eq}, u_{eq}):

$$\begin{aligned} \eta_{eq} &= \begin{bmatrix} 0 & 0 & z_{m\_eq} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \\ u_{eq} &= \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T \end{aligned} \quad \Rightarrow \quad \dot{\eta} = \mathbf{0} \tag{3.7}$$

Finding z_{m_eq} is done numerically by the script `z_graph.m`, that operates in a similar way to the ones already presented: loads the correct values from `results.m` according to the approximation method used, but this time simulate the entire maglev system (the one with 8 magnets) searching for the equilibrium points where the levmag floats freely. With the current configuration the equilibrium in the center can be found only by flipping the levmag, such that its north pole points upwards like the permanent magnets (which is the opposite orientation used for the equivalent currents). This can be noticed both from simulations as well as experimentally and the reasons of this are better explained in section 8.8. The values of the two currents in this simulation are thus both positive.

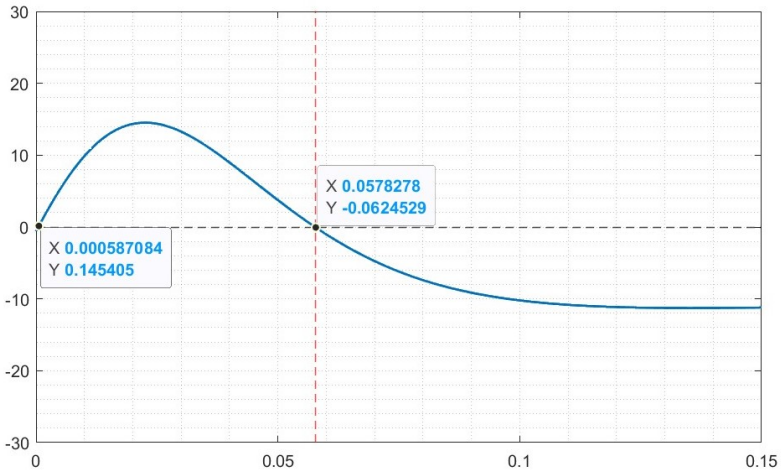


Figure 3.9: Equilibrium points along z-axis of the system

As can be seen in figure (3.9) there's one unstable equilibrium located close to the origin (top of the permanent magnets) and one stable (along z-axis) at around 5.78 cm from the origin. It is therefore logical to choose the latter as the operating point of the system. The one shown in figure (3.9) is the result obtained following the fast approximation method. The accurate one leads instead to a value of 5.89 cm, slightly higher. Considering the accurate result as a reference, the fast methods achieves less than 1.8% relative error.

3.7 Theoretical system controllability

We would now analyze if the solenoids, along with their location, are sufficiently strong and well placed to control our system. For this issue has been developed the script `solenoids_controlling_power.m` that wants to analyze the controllability of the system. In order to do this, set the x-solenoids[4] at maximum power (which in chapter 4.1 is explained to be 0.5 A) and by moving the levmag along the x-axis estimates how far the solenoids force is strong enough to bring it back towards the center. A short code snippet is shown in listing 3.5 for greater clarity.

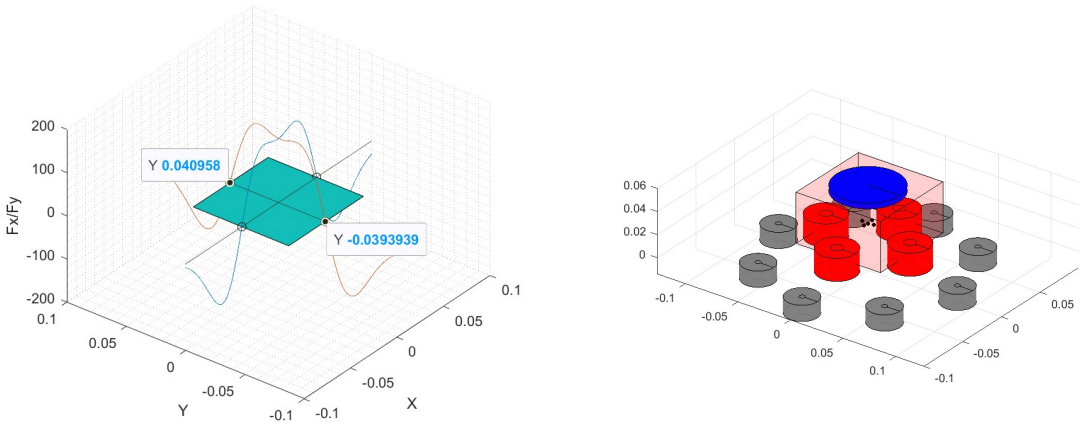
```
1 ...
2 for i = 1:length(Xs)
3     x0(1) = Xs(i);
4     temp = sys.f(x0,[.5 0 -.5 0]');
5                     %x1 y1 x2 y2
6                     %x+ y+ x- y-
7     Fxs(i) = temp(7); %x_dot
8 end
9 ...
```

Listing 3.5: X-solenoids running at max power

It then does the same with the other pair of solenoids (the y-) and moves the levmag along y-axis. Lastly uses all four solenoids together[5] to check controllability on z-axis and show a controllability zone obtained with these three results, shown in figure (3.10).

[4] as will be seen in chapter 5.2 the control strategy chosen is a dual solenoid setting for x- and y-axis (one attracts while the other repels or vice versa).

[5] chapter 5.2 explain the control strategy for z-axis



(a) x- and y-axis separate controllability (b) box plot of the controllability zone

Figure 3.10: Theoretical controllability zone plot of the system in the best case

The controllability zone obtained using these three dimensions is a box that delimits the actual zone in a best-case scenario where the levitating magnet has controllability axes independent from each other. In the reality from this box must be removed surely the edges and the farthest parts. As can be seen, the solenoids should in theory be able to correct a deviation from the origin along the x- and y-axis of up to $\approx 0.04\ m$, which was assumed to be more than enough for a system of this size. One last useful controllability metric (condition number) can be found in chapter 5.3.3 and a discussion about it can be found in chapter 8.8.

The modelling phase is now terminated. Some others scripts not presented in this chapter will be described inside the specific ones related to the stage for which they were developed, like the sensor readings estimation performance presented in chapter 4.3.2 or the linearization process presented in chapter 5.3.1.

4

Design

The electronic partial re-design results from several problems encountered with the original schematic and parts (the microcontroller and some other integrated circuits were completely burnt out due to non-optimal design choices and should have been replaced in any case).

This chapter will cover all the design process of the electronic stage. As the intention was to create a modular system, the electrical circuit was divided into several removable boards to facilitate testing, component replacement, reduce sources of error and allow individual stages to be replaced in the event of a future redesign. Electronics is essentially made up of the sub-parts shown in figure (4.1).

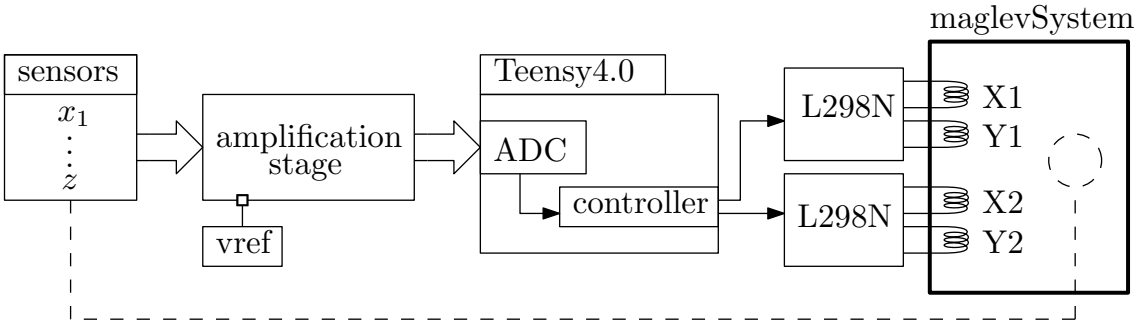


Figure 4.1: Electronics block diagram

The different boards are connected to each other with wires using pinheaders or jst-connectors. All the data sheets of the several parts here mentioned can also be found on the Github repository at `https://github.com/albertomors/maglev22/tree/main/datasheets`.

4.1 Power supply

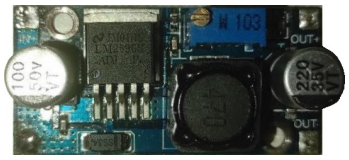


Figure 4.2: LM2596S

It was decided to supply the coils with 12 V, a commonly used value for this purpose. A female DC barrel jack was used as the power supply input connector, allowing the user to easily disconnect the power supply from the system at any time during operation. From the barrel jack, the 12 V goes directly to supply the L298N motor drivers. The rest of the circuit, however, needs just 5 or 3.3 V according to the implementation developed (see section 4.4.6). In order to obtain these power supplies an LM2596S step-down voltage regulator can be used. To select the needed AC/DC power supply the estimated maximum power consumption of the system has to be calculated. Connecting the coils to a 12 V power supply makes them absorb a maximum of 0.7 A. However, as will be seen later in section 4.2, the maximum current absorbed by each solenoid is 0.5 A. It was decided to use a 12 V, 3 A, AC/DC power supply. 3 A are sufficient to power all the four solenoids (for a total of 2 A) at the maximum current limit, while maintaining sufficient power (1 A) for the remaining electronics.

4.2 L298N controllers

To control the direction and power of the solenoids was used a L298N Dual H-Bridge Motor Driver. It can controls up to two solenoids independently, so two of these has been used. It works by receiving, for each solenoid, two digital inputs to determine the direction of the current and one PWM input to determine the voltage. One downside to the L298N was that they have a considerable voltage drop from the input voltage to the output voltage. However, this aspect was taken into account earlier and the maximum output voltage was expected to be 10.3 V. Thus, the new maximum current passing through the solenoids will be 0.5 instead of 0.7 A.

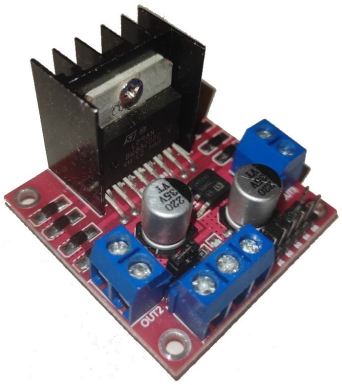


Figure 4.3: L298N

4.3 Sensors

Between the many possible sensors which would be able to determine the position of the levmag, e.g. motion sensors based on either sound or laser, or cameras, was decided to use linear hall-effect sensors due to the fact that they are inexpensive, small, require no extensive setup, and they are easy available for purchase on the market. They do not directly measure the position of the magnet but instead changes in the magnetic field. This can be used to estimate the position of the magnet by observing the changes caused by the positioning of the levmag, after establishing a reference value when it is in its equilibrium point chosen. More about the pros and the cons of this design choice can be seen in chapter 8.7.

4.3.1 OH49E

The integrated circuit (ic) selected is the OH49E. It operates in this way: when there is no outside magnetic field ($B = 0$ GS), the output voltage is one-half the supply voltage. If a south magnetic pole approaches to the front face (the side hereafter marked with an arrow), the circuit will drive the output voltage higher. Contrary, a north magnetic pole will drive the output voltage lower. The reading range goes from -1200 GS to 1200 GS, which is enough for the magnetic fields present in our system, whereas the typical sensitivity declared is 2 mV/GS.

4.3.2 Sensor board configuration

The previous team used a configuration of 3 hall effect sensors glued together at the center of the platform, one for every axis. Ideally the sensors should stay all centered in $(0, 0, z)$ with z height that allows both a good reading of the magnetic field generated by the levmag and to be far enough away to avoid damages from the levmag oscillations. However, it's physically impossible to put all them together in the center, and in fact the old ones were slightly off-centered. To solve this problem and improve the symmetry of the readings a new cross configuration that uses 5 sensors instead of 3 was developed, shown in figure (4.4).

The central sensor facing upwards is obviously the z-one. A few millimeters away from it were arranged two pairs of sensors, one for x-axis and one for y-axis. The two sensors of a pair point both outwards the sensor board, so in opposite

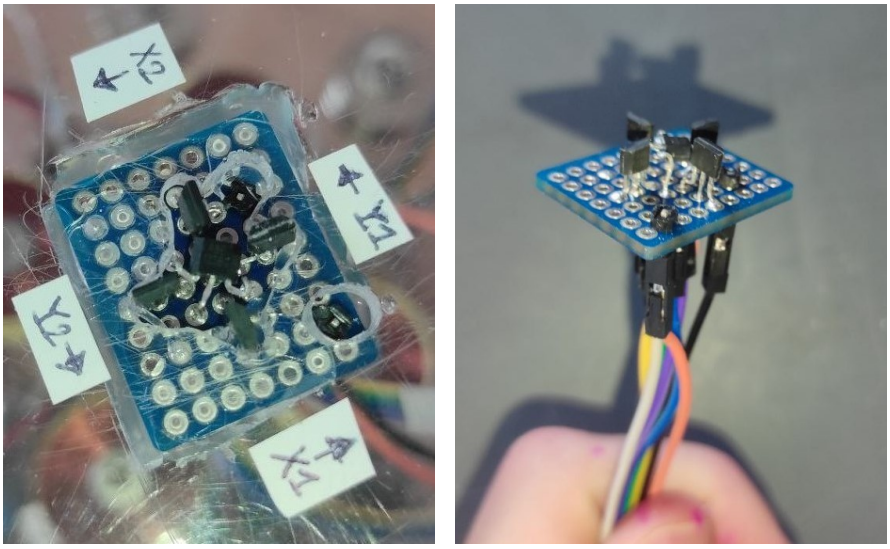


Figure 4.4: 5-sensors cross configuration

directions, in order to maximize the symmetry of the readings. The values from x- and y-axis is then obtained with a simple average[1] between the two sensors to estimate the x- and y- deviations. With this configuration, if the levmag goes down, z-sensor will drive its output voltage higher. The same thing happens to the corresponding sensors if it goes towards the direction pointed by x_1 or y_1.

We wanted to analyze, using MATLAB, how much this solution deviates the readings from a virtual 3-axes sensor positioned exactly at the center. The script `y_error` simulates two systems, one with the sensor configuration above described and one with the virtual sensor, moving the levmag along a specific path and comparing the measurements along it. The results are shown in figure (4.5), where `<x>` is the calculated average from the real sensors and `X` the ideal value measured in the center by the virtual 3-axes sensor.

As can be seen, this method only slightly errs the value of the magnetic field measured, which can be considered a success.

4.4 Amplification design

Due to the low sensitivity of the sensors, the output voltage was too insufficient to measure all the movements of the levitating magnet. To solve this an

[1] with the right signs, accordingly to the direction

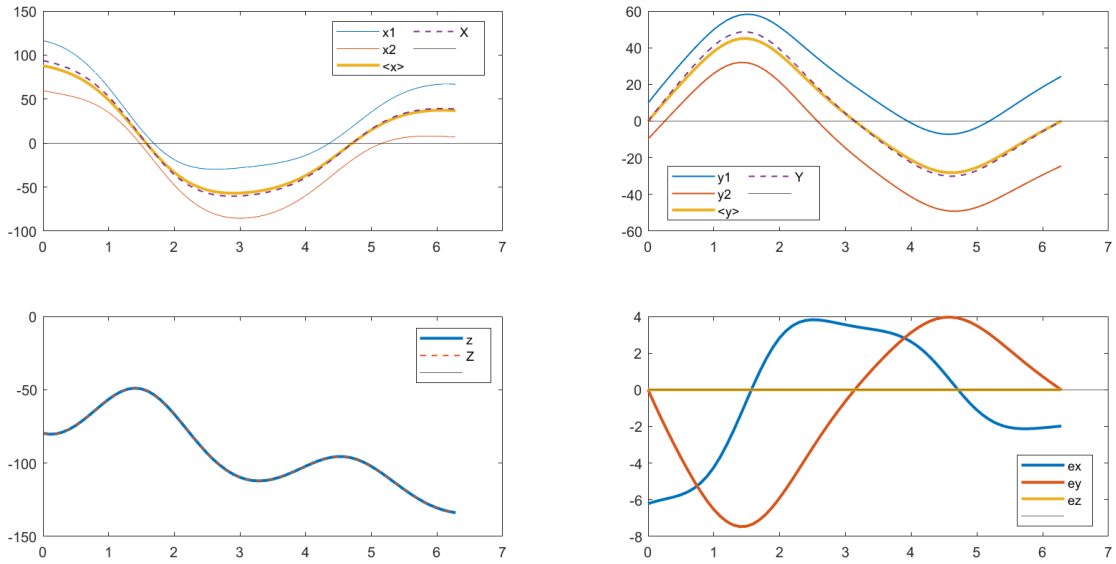
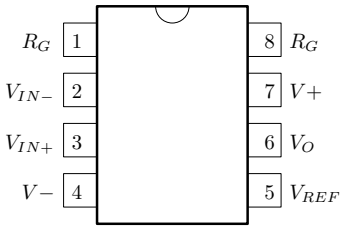


Figure 4.5: 5-sensor cross configuration vs virtual sensor in the center

intrumentation amplifier (in-amp) was added to the output of each sensor. The in-amp is used for several reasons: amplify the readings, eliminate the measurement offset caused by the presence of the permanent magnets, and adapt the output voltage for the microcontroller. The removal of the offset is done by connecting a potentiometer to the negative input pin of the in-amp. When the levmag is in equilibrium, the potentiometer is adjusted until the differential input voltage across the in-amp is zero.



Since the output has to be read by the Analog to Digital Converter (ADC) of the microcontroller, it has to be adapted to match the reading range of the internal ADC of the microcontroller, equal to 0–3.3V. Teensy data sheet says that not only is it unable to read higher voltages, but that applying them is also dangerous cause it may burn the input pin, and therefore something is needed to prevent the output voltage from exceeding this limit. Since the input voltage can vary either positively (when the magnet moves in one direction) as well as negatively (when it moves in the opposite direction), so does the output differential voltage. To translate the output voltage by a precise offset to the centre of the reading range (1.65 V) there is a specific pin on which to apply reference input voltage (how to obtain it is explained in section 4.4.5). This voltage reference input establishes the common-mode voltage of the output. The block diagram of

the amplification chain can be seen in figure (4.6).

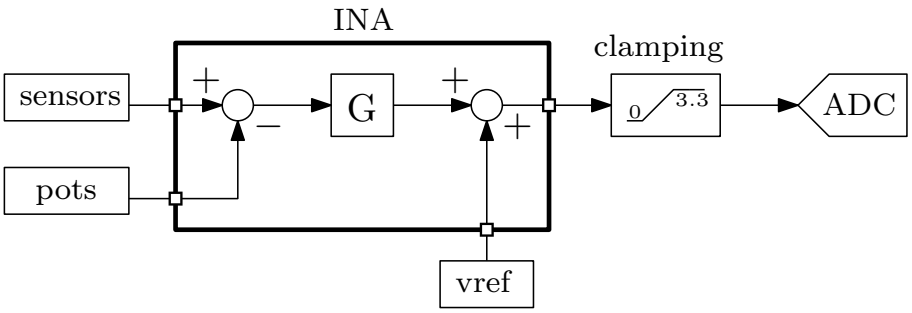


Figure 4.6: Block diagram of the amplification chain

The amplification chain design process took up a significant portion of the project time due to a number of problems encountered. The following is a chronological discussion organised in problem-solution sections which traces all the individual stages in search of the best solution adopted at the end.

4.4.1 Previous INA126 implementation

The amplification chain of the original model was realised with some INA126 fed with a single-supply of 0–5 V and characterised by a gain value of $5+80k\Omega/R_G$ settable by an external resistor R_G connected to pins 1 and 8. However, the minimum gain value of 5 was too high so a voltage divider stage was inserted downstream to reduce this voltage by 1/3, resulting in a total gain of $5 \cdot 2/3 \approx 3.3$[2]. This obviously compromised the accuracy achieved, especially when realized with a second voltage divider stage using resistors with a tolerance of ±5%. Also the reference voltage was obtained with a trivial voltage divider using two fixed resistors, once again inaccurate.

4.4.2 INA128 first approach

As a first approach to the problem were used some INA128, for the amusing reason that they were the wrong components sent us with the model by the previous team, but characterised by a gain of $1 + 50k\Omega/R_G$, again settable with an external resistor R_G, and thus allowing the gain of 3.3 to be achieved in a single

[2]The gain value choosing method will be discussed in depth in chapter 7.2, for now the same one designed by the previous team will be used as a reference for the design phase

step instead of two. As reported by its data sheet and simulated with multisim, this solution has two main problems:

- first, the minimum supply voltage for the positive supply pin is 4.5 V, so we can't supply it with 3.3 V to prevent the output voltage from going higher, and we need some additional protection stage to do this
- second, the positive and the negative output voltage swing is 1.4 V smaller than the power supply range (both positively and negatively). This means the output voltage, with a single-supply feed of 0-5 V, can't go lower than 1.4 V or higher than 3.6 V. The positive saturation is not a problem, cause we don't need to go higher than 3.3 V, but the negative is.

Fly-back converter solution

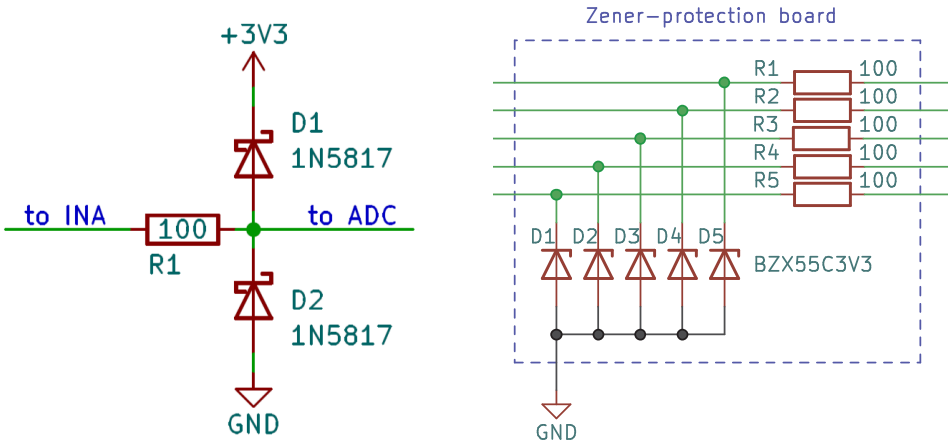
All the in-amps presented here can operate both in single or dual power supply. The first solution that immediately comes to mind is to somehow obtain the negative supply voltage of -5 V and supply the INA128 with ±5 V. With this solution, the output voltage can swing properly between a linear output range of ±3.6 V.

The −5 V supply voltage could be obtained with an integrated fly-back converter that provides an isolated 5 V output from the 12 V voltage (which, being isolated, can be connected with the positive output terminal to ground (0 V), thus realising a negative supply voltage of -6 V on its negative output terminal). This solution allows the output to be correctly driven in the desired linear range, at the cost of having to limit the output voltage not only on the positive range, but also on the negative range (the voltage read by the ADC cannot even go below 0 V).

Clamping circuits

Among the several existing clamping circuits the simplest and cheapest one is realized using two shottky diodes. A shottky diode is characterized by having a very low forward voltage drop (150-450 mV instead of 600-700 mV) and a very fast switching action. This lower forward voltage drop allows it to be used in these clamping applications. The application circuit is shown in figure (4.7a).

Another possible clamping circuit, but one that can only be used on the positive limit, is realized with a 3.3 V zener diode with the cathode facing the ADC input



(a) Shottky's clamping circuit (b) Zener's clamping circuit

Figure 4.7

pin. This protection circuit, shown in figure (4.7b) is the one used in the selected final implementation with AD623 in-amp discussed in chapter 4.4.6.

The full developed circuit simulated using multisim during his positive clamping action can be seen in figure (4.8). The negative clamping acts in the same way, but turning on the lower diode.

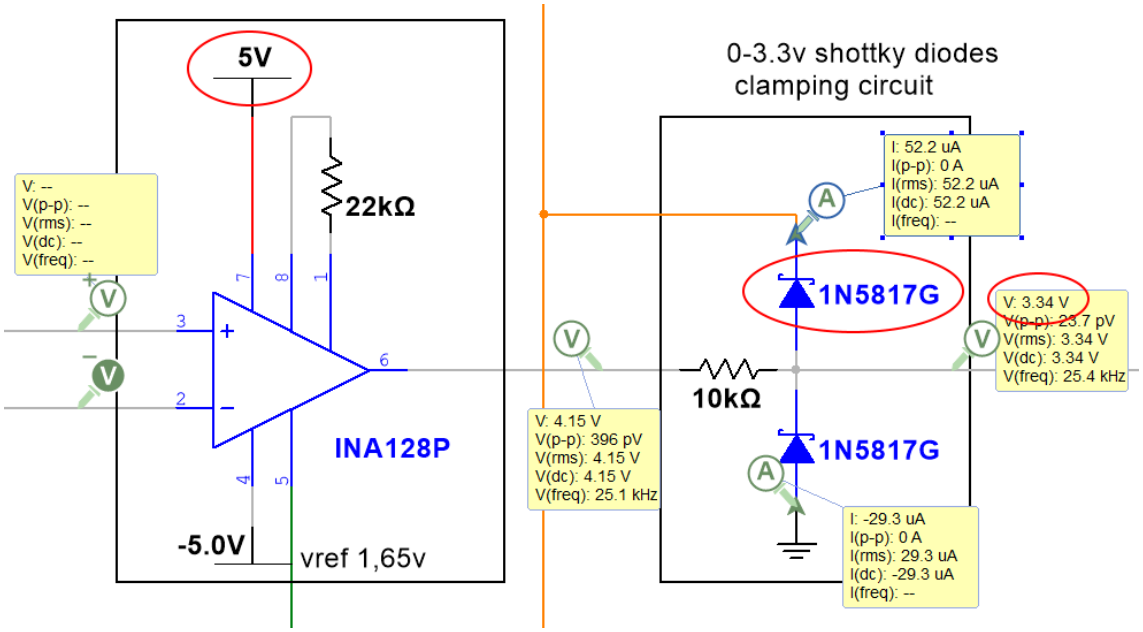


Figure 4.8: INA128 full schematic

4.4.3 INA326 rail-to-rail solution

A second solution was developed with the INA326. Its main feature is that it is a rail-to-rail in-amp, i.e. its output can swing almost (10 mV less) up to the power supply rails. Moreover, it can be supplied with a voltage between 2.7 and 5.5 V, allowing us to supply it with 3.3 V. Another important feature is the input voltage range: the data sheet states that the voltages applied to its input pins can safely range from 20 mV below the negative rail to 100 mV above the positive rail. It is important that the voltages of the potentiometers and the sensors output never exceed these limits, so the power supply of amplifier and sensors must be decided together to respect these constraints. All these features seem to make it an undisputed winner.

Design

Choosing a power supply of 0–3.3 V for the in-amp prevents the output voltage to exceed the ADC input limits without any additional stage. The same voltage is applied to supply the sensors and the potentiometers. The fact that the input voltages can go 20 mV below the negative rail (-0.02 V) to 100 mV above the positive rail (3.1 V) makes the input stage intrinsically very robust and capable of withstanding any combination of the input voltages, which cannot vary more higher than 3.3 V or lower than 0 V. The gain is settable with 2 external resistors connected to some of its pins as seen in figure (4.9) calculated as $G = 2 \cdot (R_2/R_1)$. Application notes on its data sheet suggest using an $R_2 \approx 200\ k\Omega$ (along with a 0.5 nF C_2 capacitor connected in parallel[3]) for desired G around 2. It was then chosen $R_2 = 200\ k\Omega$ and to use a 200 $k\Omega$ potentiometer as R_1, which under ideal conditions would be set at $\approx 60.61\%$ (121.21 $k\Omega$), but under realistic conditions allows the gain to be precisely adjusted to achieve the desired value of 3.3. As also suggested by the application notes it was added a 0.1 μF capacitor placed close to and across the power supply pins for highest accuracy, along with a R_0C_0[4] output filter that minimizes auto-correction circuitry noise and also acts as an anti-aliasing filter for a subsequent ADC, which is precisely our case. The full schematic during his positive clamping intrinsic saturation is shown in figure (4.9).

[3] C_2 and C_0 combine to form a 2-pole response that is 3 dB at 1 kHz. Each individual pole is at 1.5 kHz.

[4] $R_0 = 100\ \Omega$, $C_0 = 1\mu$F

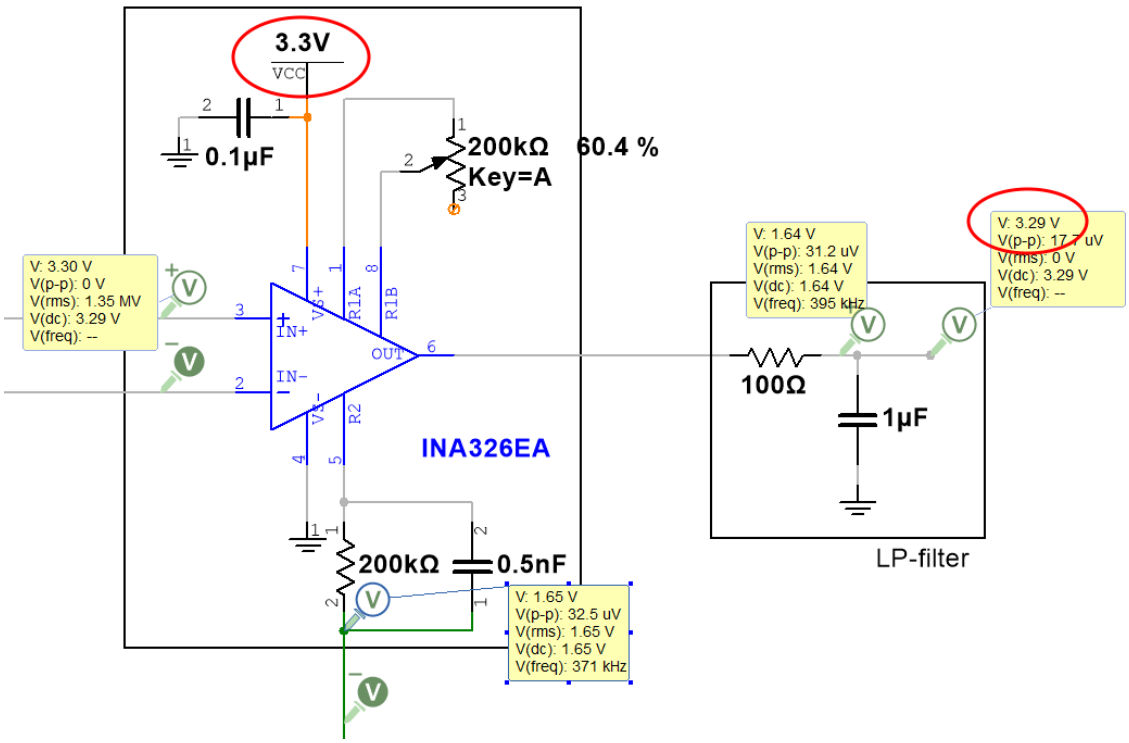


Figure 4.9: INA326 full schematic

Components availability

Unfortunately, we had to deal with the fact that they were not available anywhere on the market (probably because it is discontinued), and that, even if we had found them, the only existing model is in the form of a surface mounted chip (SMD), suitable for being mounted on a printed circuit board (pcb) but not for a stripboard like the one used, something we had not taken into consideration during the design phase in multisim. It was also considered to look for adapter sockets from SMD to DIP8 but we decided to give up because of the major difficulty of unavailability of the part on the market.

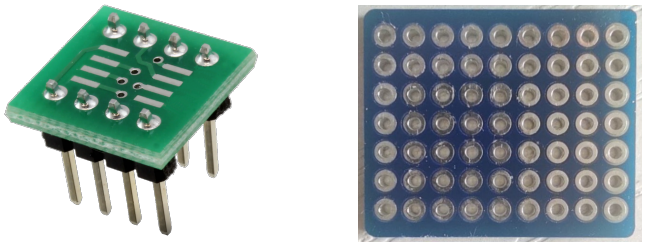


Figure 4.10: SMD to DIP8 socket and stripboard examples

4.4.4 AD623 TRADE-OFF

After extensive online research on a worthy INA326 substitute we came to the conclusion that we could not find another in-amp with the same rail-to-rail characteristics. Most of the parts found offered the same output characteristics, but with limitations on the input range. The best candidate found was the AD623. It features a rail-to-rail output swing from $V_- + 50\ mV$ to $V_+ - 150\ mV$ but an input voltage range that can swing from $V_- - 150\ mV$ to $V_+ - 1.50\ V$, where the single supply voltage V_+ can be a voltage between 2.7 and 12 V. Supplying it with 3.3 V to prevent output exceeding the ADC safe limits means that the input voltages could not go over $3.3 - 1.5 = 2.7\ V$. This is an enormous limitation to be dealt with. The only solution is to supply the AD623 with 5 V, so that the input voltages could go from 0 to $5 - 1.5 = 3.5$ safely. Sensors and potentiometers are then supplied with 3.3 V, so as to remain in safe voltages zone, and the output must, however, be clamped only positively to 3.3 V, because has the possibility of reaching up to 4.85 V.

DESIGN

The gain of the AD623 can be set by one external resistor connected between pins 1 and 8 of the in-amp, according to the equation $G = 1 + 100k\Omega/R_G$, where R_G is a potentiometer (to allow an accurate gain adjustment, and, as will be seen in chapter 7.2, the possibility of changing it as required), set to 43.48 $k\Omega$ to obtain the desired gain of 3.3. For optimal results, application notes suggest adding a 0.1 μF ceramic capacitors and a 10 μF electrolytic tantalum capacitors between pins 4 and 7 to capacitively decouple power supplies close to the power pins, recommendation followed by us, as can be seen in figure (4.11). The output is then clamped with the zener protection stage shown and discussed in section 4.4.2, as only a positive clamping action is required.

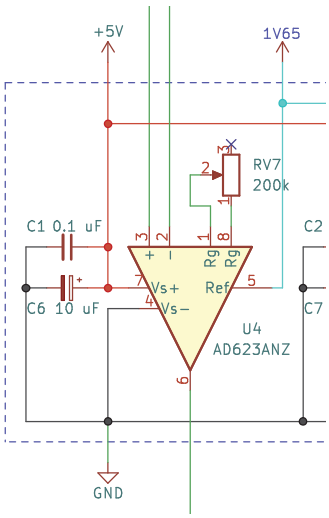
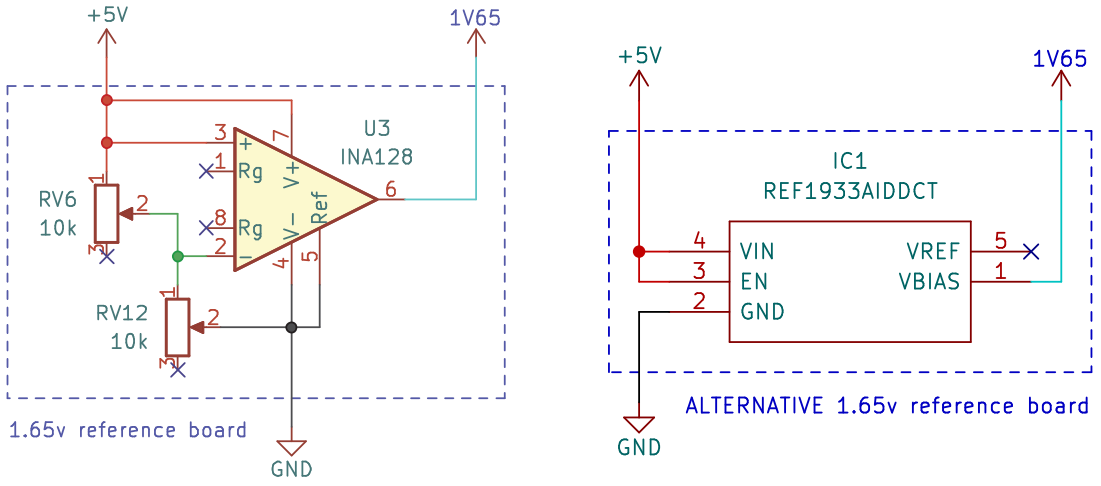


Figure 4.11: AD623 full schematic

4.4.5 Voltage reference

Regarding the voltage reference (vref), the old design used a trivial voltage divider to obtain the half-supply voltage. This is not properly correct, because, as also stated in the data sheet of the in-amps used, the pin vref must be driven by a low impedance or connected to ground if not used. In order to achieve this is necessary to use an output stage of another operational amplifier (op-amp). For convenience, as we already had them at our disposal, we used an INA128 with two 10 kΩ potentiometers as shown in figure (4.12a). By adjusting the two potentiometers, the desired vref value of 1.65 V can be obtained.

A more elegant and compact solution, especially in case it will be decided to make a pcb in the future, can be achieved by using the REF1933. It is a 3.3 V ic voltage reference, whose distinguishing feature is a dual voltage output: VREF of 3.3 V, and the second VBIAS which is half the voltage of VREF, i.e. 1.65 V, the exact value needed by us.



(a) INA128 used vref schematic

(b) suggested alternative schematic

Figure 4.12

4.4.6 Adopted solution

At the end was decided to keep this last solution with the AD623, clamping the outputs with a zener-diodes protection stage. In fact, considering the techno-economic constraints, it is the solution that buying fewer components gives us the best results (the first solution would have required the purchase of 10 additional

shottky diodes and a fly-back converter, occupying also additional space). We then need two supply voltages (3.3 for potentiometers and sensors and 5 V for the in-amps, and as we will see, for the Teensy), obtained with two different LM2596S from the 12 V as shown in figure (4.13). The voltage reference is obtained with an INA128 as shown in section 4.4.5.

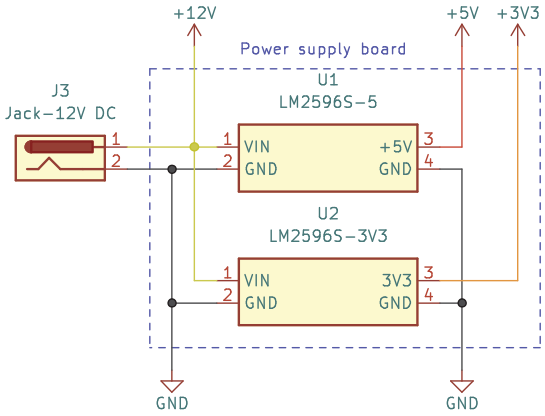


Figure 4.13: Necessary power supply stage for the adopted solution

The amplification stage was later mounted on a breadboard (shown in figure 4.14), tested, and gave the excellent results expected.

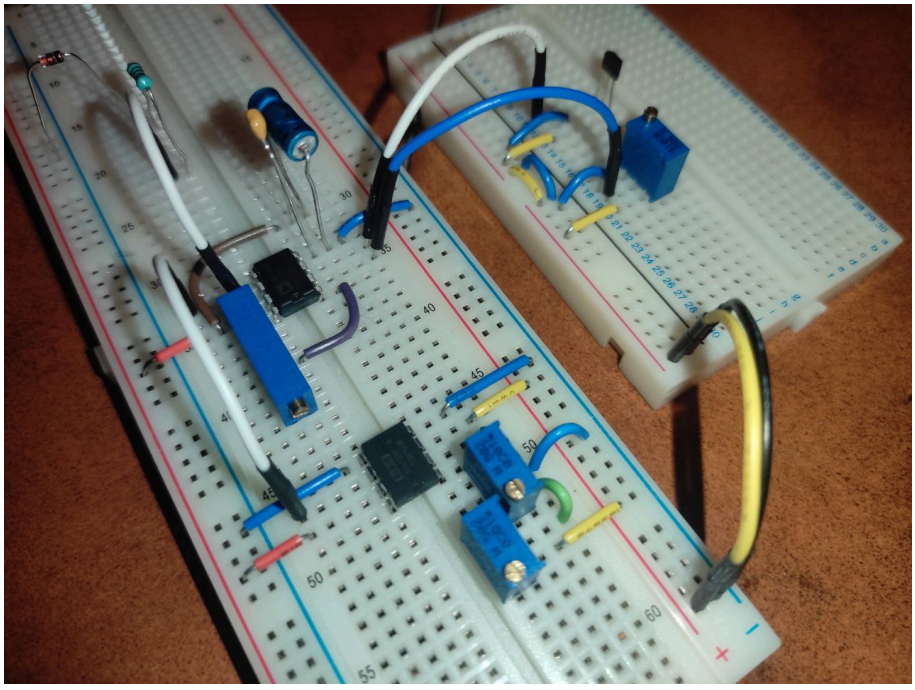


Figure 4.14: Testing on breadboard the amplification stage

4.5 Microcontroller's wiring

The Teensy pinout is clearly described by figure (4.15). Pins 2 to 7 are dedicated for the SX motor driver, which controls X1- and Y1-coils, the ones considered positive, while pin 8 to 12 plus 14 are dedicated for the DX motor driver, which controls X2- and Y2-coils. Pin 13 was skipped because it's dedicated to Serial communication. As can be seen in appendix B, all the pins here mentioned support PWM modulation (in the Teensy pinout called *PWM_pins*), so the pin order is kept consistent with that of the motor driver, without crossing wires. Pins 15 to 19, chosen from the available *Analog_pins*, are used to read the 5 voltages coming from the amplification stage, while pin 20 is used to read the in-amp vref value, so that the potentiometers can be easily adjusted with a dedicated testing routine (see chapter 6.3) without the use of any measuring instruments.

Teensy can be powered either via USB cable as well as using an external power supply or a battery by connecting it to pin V_{IN}. But since these two are internally connected together, it is not possible to use both at the same time. It is therefore important to disconnect them by cutting the connection between the dedicated pads V_{USB} and V_{IN} if you decide to use an external power source and at the same time maintain serial communication with the PC. This was done by us at the right time, after finishing assembling and testing the power supply section. The full electrical schematic can be seen in appendix A.

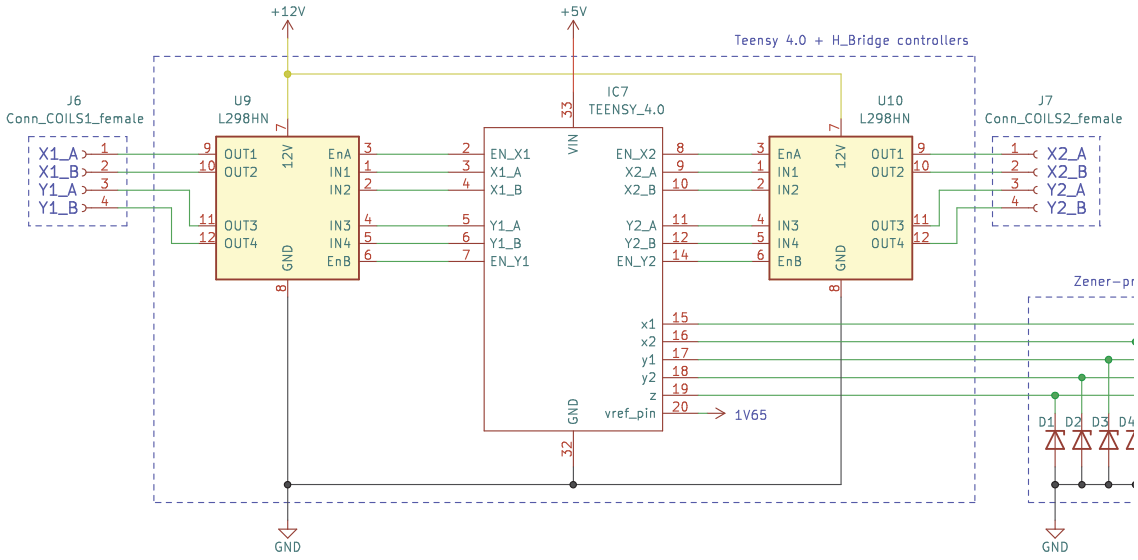


Figure 4.15: Teensy - L298N wirings

4.6 System housing and plexiglass frame

To design the layout of the components a Fritzing schematic of components, wiring and paths (shown in appendix C) was first realized in order to get an idea for the successive practical realization of each circuit on stripboard. This meticulous procedure allowed us to fit all the new boards into the same frame of the old project (with three more boards). The sensor board was raised one layer, now emerging from the last plexiglass level, so we had to drill a new cross-shaped hole (see figure 4.16) in the level to insert it. The mounting holes for the circuit boards were also drilled by hand, once their position had been decided. A detailed description with measures and so on about the plexiglass frame can be found in appendix D.

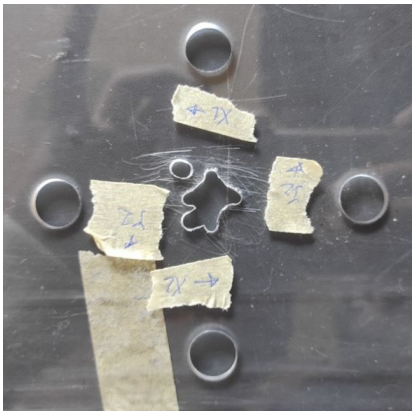


Figure 4.16: Sensor board cross-shaped hole

4.7 Building the platform

The process of soldering and realizing the single boards gave really good results (see figure 4.17), also thanks to the meticulous preparation of the Fritzing schematics, which allowed us to go off without a hitch, having already studied all the component locations with their wiring on the stripboard. Sockets were also soldered for critical parts such as in-amp and microcontroller, so as not to thermally stress the ic during the soldering process and to allow easy removal and/or replacement in case of fault.

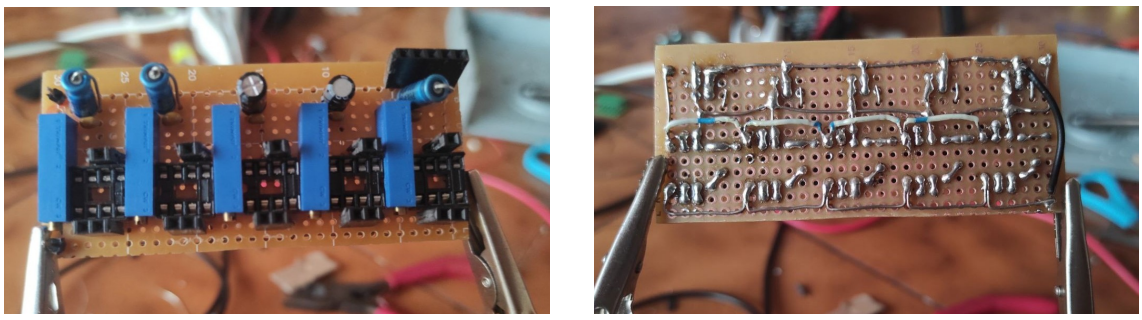
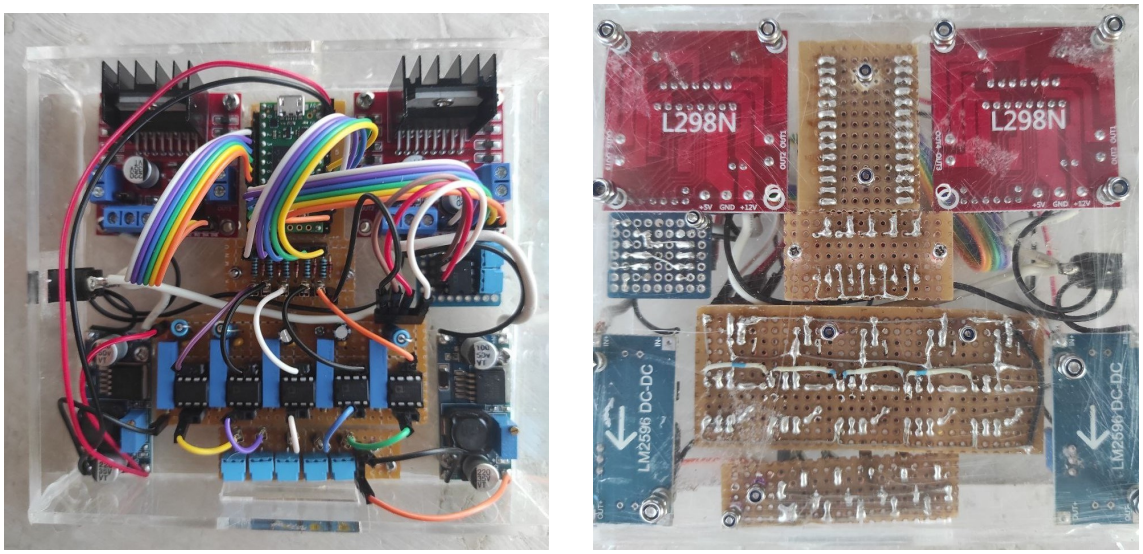


Figure 4.17: Soldering

After finishing soldering and assembling all the electronics these are the beautiful results, shown in figure (4.18).



(a) Top (b) Bottom

Figure 4.18

When mounting the rest of the platform (solenoids, magnets, sensors) on the remaining two levels, some simple rules of good behaviour were followed. As first, connectors were made to easy connect/disconnect the various peripherals (solenoids, sensors), to facilitate the mounting phase and to maximise the modularity of the electronics (see figure 4.19).

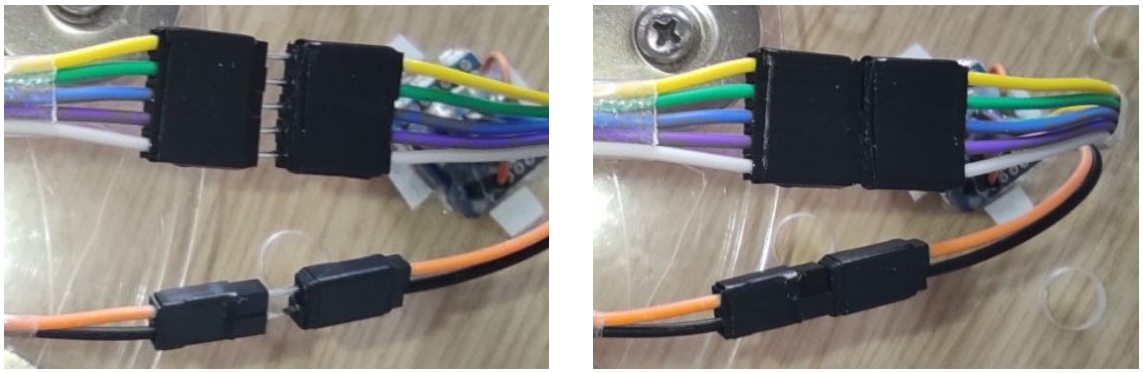


Figure 4.19: Connectors made for the sensors

Long cables were used for these peripherals, so that the system could operate with its different layers separated, which was useful during the testing phase as will be seen in chapter 7 (see figure 4.20).

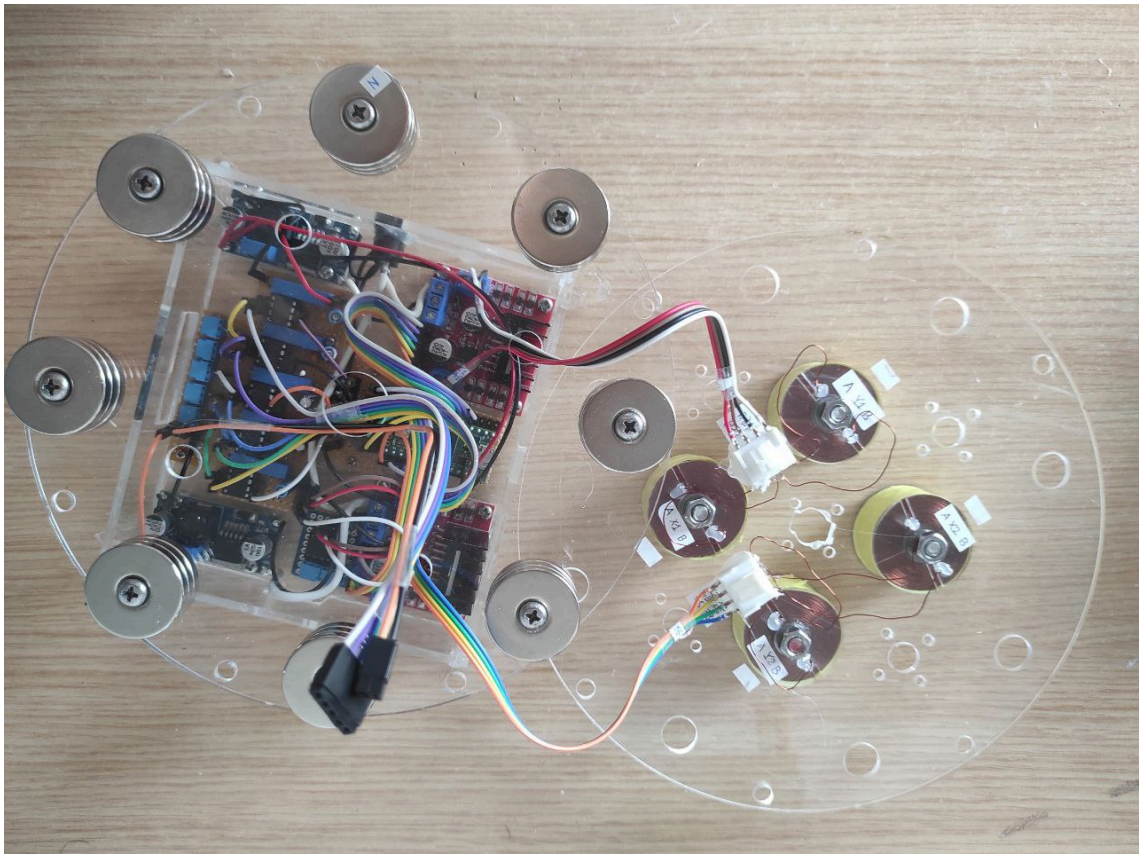


Figure 4.20: Platform opened

All these cables and connectors have been labelled to reduce the possibility of error (see figure 4.21). Lastly, the magnets were also labelled indicating their north pole, and solenoids and sensors with their names, to ease debugging.

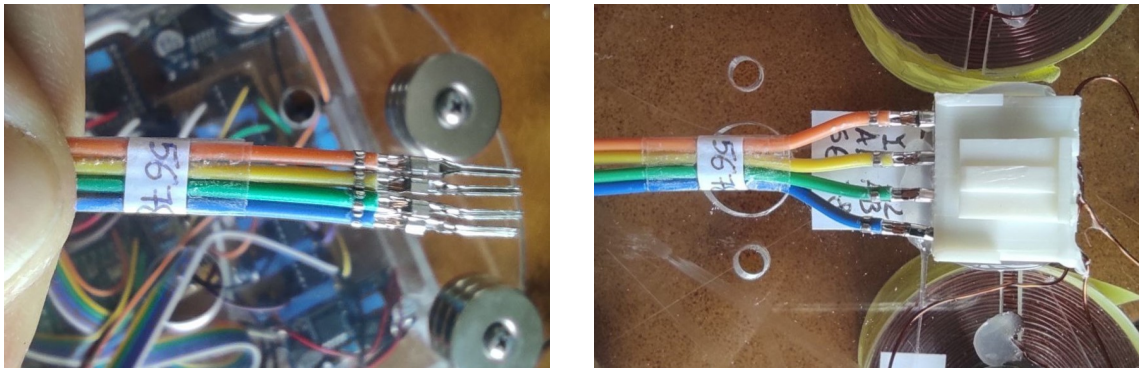
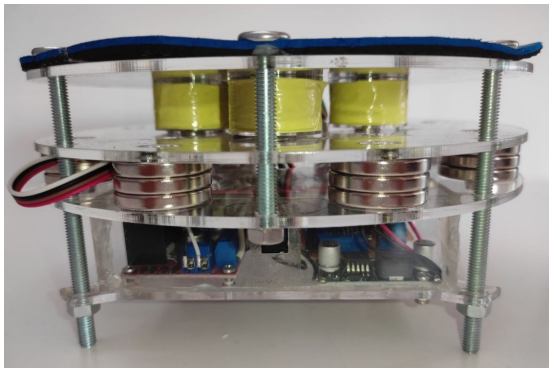


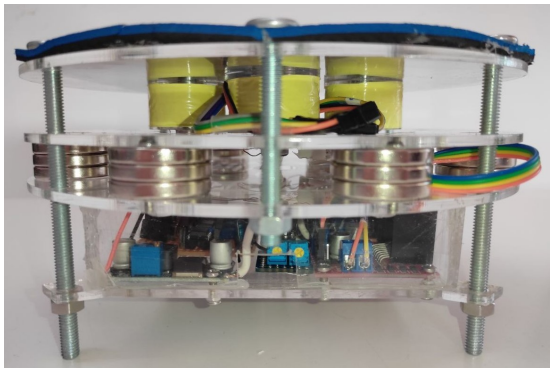
Figure 4.21: Labelling solenoid's wires and connectors

4.8 Results

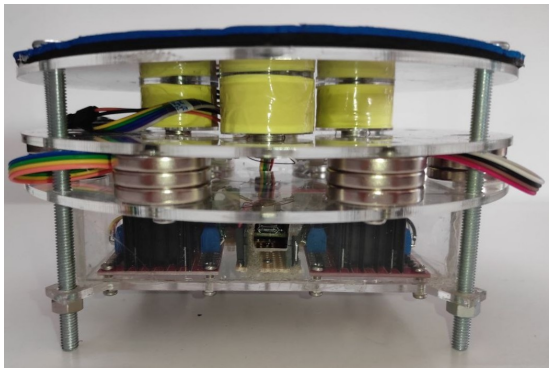
The platform fully build is shown in figure 4.22. The five potentiometers used as reference for the sensors, the two to adjust the voltage reference and the Teensy micro-usb port are accessible from the outside of the plexiglass frame thanks to some windows made appositely.



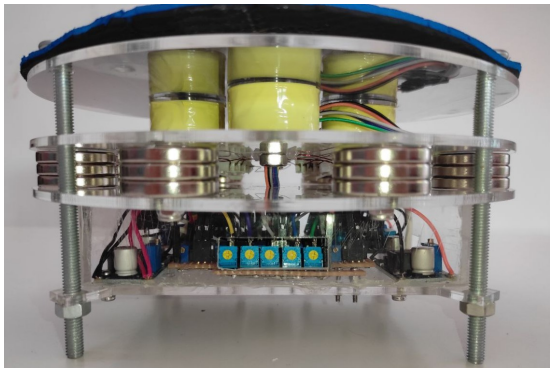
(a) DC-barrel jack side (left)



(b) vref side (right)



(c) micro-usb side (top)



(d) potentiometers side (bottom)

Figure 4.22

5

PID controller design

When being introduced to a plant or process that has to be controlled, it is important to choose a good strategy for the control structure. This includes choosing the type of the controller, the design to implement it on the microcontroller used and the tuning process to obtain the desired characteristics. This chapter will therefore describe and discuss the choices that were made in relation to controller design.

5.1 Choice of the control structure

The controller used was not changed from the one designed by the previous team, due to time constraints of the project but it will be discussed in chapter 8.11. It was set up with three parallel PID-controllers receiving a total of three inputs representing the magnetic field deviations from the equilibrium on the three axes. Actually, as the levitating magnet moves along the xy-plane, the magnetic field in the x- and y-sensors change. The same thing occurs if the levitating magnet moves along the z-axis. The three inputs can somewhat be an expression of the position of the magnet. The microcontroller sends out four PWM output values to the L298N modules that supply the currents for the solenoids (output algorithm represented by Hc) The magnetic levitation platform is therefore a 5×4 Multiple-Inputs Multiple-Outputs (MIMO) system, as shown in figure 5.1.

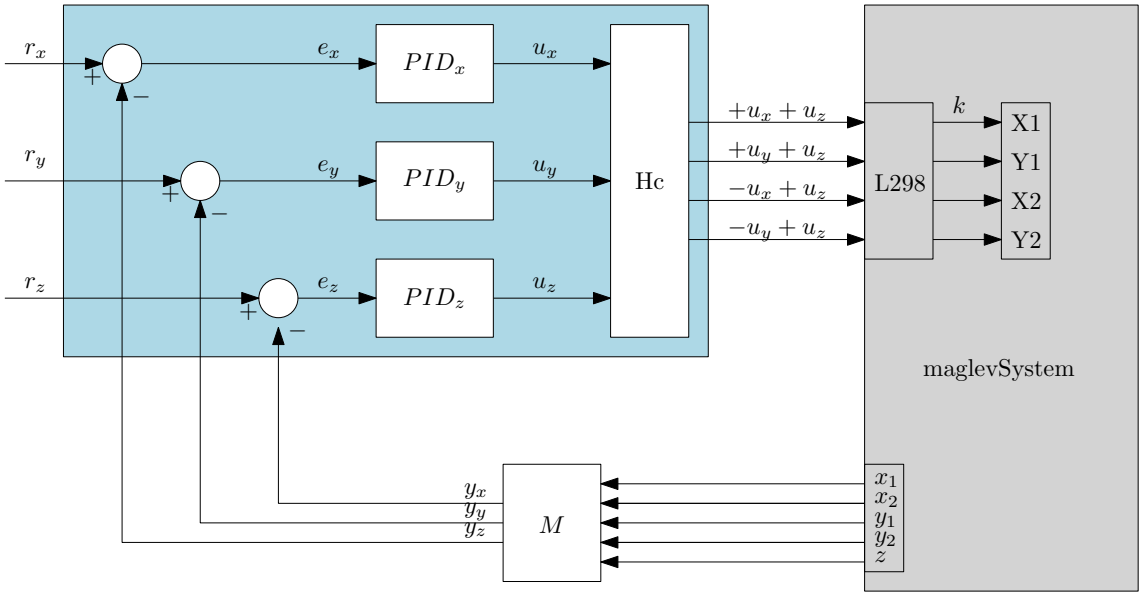


Figure 5.1: Block diagram control structure

5.2 Output algorithm

The selected control strategy consists of 3 individual Single-Input Single-Output (SISO) PID-controllers. As explained in section 5.3.3 this is theoretically a good control structure. The selected strategy is to control the solenoids in a dual mode to correct the deviations along the x- and y-axis, while using them all together to correct the deviations along the z-axis. For example, if the levitating magnet moves toward one the solenoid $X1$, it will push him back towards equilibrium while the dual solenoid $X2$ will pull it, and vice versa. The same thing is done for y-axis. On the other hand with respect to the control along the z-axis, if the levitating magnet goes down, all the four solenoids will try to push him back up, and vice versa. For the levitating magnet to be pushed, considering that its north pole points downwards, it is necessary to create a north pole on the top of the solenoids, running a positive current through the solenoid. On the other hand for pulling a negative current has to be run on the solenoids. The correctness of this choice and whether it is the best strategy will also be discussed in chapter 8.8.

5.3 Linear analysis

As said before the system is strongly non-linear. To make the system easier to analyze and implement a generic controller the system has to be linearized

around the equilibrium chosen as operating point found in chapter 3.6, in this section referred as $(\bar{\eta}, \bar{u})$. The linearized system can then be expressed by the simple equation:

$$\begin{aligned} \dot{\eta} &= A\eta + Bu \\ y &= C\eta + Du \qquad \text{with } D := \mathbf{0} \end{aligned} \tag{5.1}$$

5.3.1 Linearization

The matrices A, B and C can be found by taking the partial derivatives of the existing system nl functions shown in equation (5.2).

$$\begin{aligned} \dot{\eta} &= f(\eta, u) \\ y &= h(\eta, u) \end{aligned} \tag{5.2}$$

with respect to η and u:

$$\begin{aligned} A &= \begin{bmatrix} \left.\frac{\partial f}{\partial \eta_1}\right|_{(\bar{\eta},\bar{u})} & \left.\frac{\partial f}{\partial \eta_2}\right|_{(\bar{\eta},\bar{u})} & \cdots & \left.\frac{\partial f}{\partial \eta_{12}}\right|_{(\bar{\eta},\bar{u})} \end{bmatrix} \\ B &= \begin{bmatrix} \left.\frac{\partial f}{\partial u_1}\right|_{(\bar{\eta},\bar{u})} & \left.\frac{\partial f}{\partial u_2}\right|_{(\bar{\eta},\bar{u})} & \cdots & \left.\frac{\partial f}{\partial u_4}\right|_{(\bar{\eta},\bar{u})} \end{bmatrix} \\ C &= \begin{bmatrix} \left.\frac{\partial h}{\partial \eta_1}\right|_{(\bar{\eta},\bar{u})} & \left.\frac{\partial h}{\partial \eta_2}\right|_{(\bar{\eta},\bar{u})} & \cdots & \left.\frac{\partial h}{\partial \eta_{12}}\right|_{(\bar{\eta},\bar{u})} \end{bmatrix} \end{aligned} \tag{5.3}$$

The partial derivatives could not be solved analytically and has to be approximated using the following finite difference approximation:

$$\frac{\partial f(\eta, u)}{\partial \eta} \approx \frac{f(\eta + \Delta, u) - f(\eta - \Delta, u)}{2\Delta} \tag{5.4}$$

Where Δ is the step size. The process is done numerically in MATLAB by the script `RGA_analysis.m`, shown in listing (5.1).

```
1 ...
2 delta = 1e-4;
3 dimX = 12;
4 dimU = params.solenoids.N; % 4
5 dimY = 3*length(params.sensor.x); % 3x5=15
6 ...
7 % d(x/dx) empty matrix of dimension x*x (12x12) where
```

```
% on rows there's df(i=1:12)/.
% on columns there's ./dx(i=1:12)
A = zeros(dimX,dimX);
for i = 1:dimX % for every column
    % linearize the entire column
    A(:,i) = (sys.f(xLp+(i==1:dimX)'*delta,uLp) ...
              -sys.f(xLp-(i==1:dimX)'*delta,uLp)) ...
              /(2*delta);
end

% d(x/du) empty matrix of dimension x*u (12x4)
B = zeros(dimX,dimU);
for i = 1:dimU
    B(:,i) = (sys.f(xLp,uLp+(i==1:dimU)'*delta) ...
              -sys.f(xLp,uLp-(i==1:dimU)'*delta)) ...
              /(2*delta);
end

% d(y/dx) empty matrix of dimension y*x (15x12)
C = zeros(dimY,dimX);
for i = 1:dimX
    C(:,i) = (sys.h(xLp+(i==1:dimX)'*delta, uLp) ...
              -sys.h(xLp-(i==1:dimX)'*delta, uLp)) ...
              /(2*delta);
end

% d(y/du) = 0
D = zeros(dimY, dimU);
```

Listing 5.1: Linearization process

5.3.2 Transfer function matrix

Using the linearized model from section 5.3.1, the transfer function matrix $G(s)$ from u to y such that $y = G \cdot u$ can be found as:

$$G(s) = C[sI - A]^{-1}B \tag{5.5}$$

With $G(s)$ being a 5×4 matrix, one row for every output and one column for every input. In MATLAB, the y value is the vector concatenation of the 5 virtual sensor measurements x_1, x_2, y_1, y_2, z, so a 15×1 vector instead of 5×1, because going back to chapter 3.4.4 we remember that the virtual implementation of the sensors measure the magnetic field of every axis. We so proceed to trash the values not corresponding to the purpose of the sensor. To use instead the averages value from x and y we use a matrix M to get the transfer function from u to the three values of magnetic field measured $(y_x, y_y, y_z) := y2$ shown in figure (5.2).

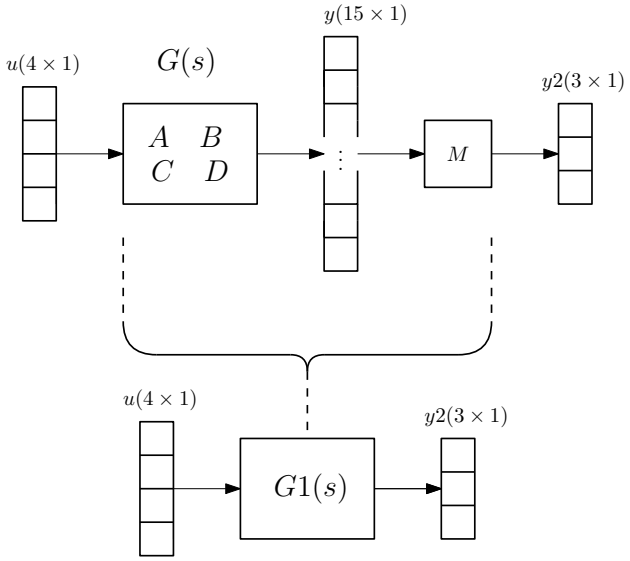


Figure 5.2: G1(s) transfer function

The measurements in y are concatenated 3 by 3, so it looks like (x-measure of sensor x_1, y-measure of x_1, z-measure of x_1, x-measure of x_2, ...). In order to model the input algorithm we need a matrix M such that:

$$M \cdot y = M \cdot \begin{bmatrix} x_{1\_x} \\ \vdots \\ z_z \end{bmatrix} = \begin{bmatrix} 0.5 \cdot x_{1\_x} + 0.5 \cdot x_{2\_x} \\ 0.5 \cdot y_{1\_y} + 0.5 \cdot y_{2\_y} \\ z \end{bmatrix} = \begin{bmatrix} 0.5 \cdot y(1) + 0.5 \cdot y(4) \\ 0.5 \cdot y(8) + 0.5 \cdot y(11) \\ y(15) \end{bmatrix} \tag{5.6}$$

which corresponds to this matrix:

$$M = \begin{bmatrix} 0.5 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.7}$$

As explained, the controller was set up with three parallel PID-controllers actuating not the four current inputs $u(4 \times 1)$ but instead the values (u_x, u_y, u_z). To model the output algorithm implemented on the microcontroller we need to find the link from the input values set by the PID-controller to the solenoids input currents u as shown in figure (5.3).

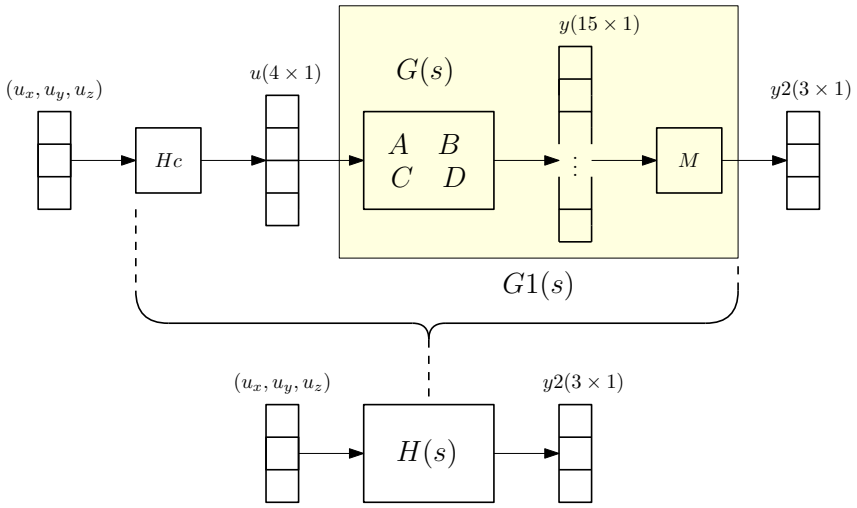


Figure 5.3: H(s) transfer function

This can be done with a matrix, here called Hc, such that:

$$Hc \cdot u = Hc \cdot \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = k \cdot \begin{bmatrix} u_x + u_z \\ u_y + u_z \\ -u_x + u_z \\ -u_y + u_z \end{bmatrix} \tag{5.8}$$

where $k = 0.5/255$ is the PWM value to currents gain. It then corresponds to the matrix:

$$Hc = \frac{0.5}{255} \cdot \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \end{bmatrix} \tag{5.9}$$

The code snippet where you can see this process is reported in listing (5.2).

```
1  ...
2  %% State space to transfer function
3  ssModel = ss(A,B,C,D);  % State-space model
4  G = tf(ssModel);  % Creating a transfer matrix
5  ...
6  %    1     4       8     11      15
7  M = [1 0 0 1 0 0 0 0 0 0 0 0 0 0 0;
8       0 0 0 0 0 0 0 1 0 0 1 0 0 0 0;
9       0 0 0 0 0 0 0 0 0 0 0 0 0 0 1]./[2; 2; 1];
10
11 G1 = M*G;
12 ...
```

```
Hc = 0.5/255 .* [1  0  1;
                 0  1  1;
                -1  0  1;
                 0 -1  1];

% final tf from (ux,uy,uz)' to y2
H = G1*Hc; % transfer matrix of the chosen control strategy
...
```

Listing 5.2: Transfer function H(s)

$H(s)$, $G(s)$ and the state space matrices can be found in the GitHub repository as `ss_matrices.mat`. The high order and the pole-zero-placement of the transfer functions made it difficult to use any familiar techniques to analyze the system stability. Even though the linearization process was not necessary to implement a PID controller, it was still done in case a different type of controller is implemented in the future.

5.3.3 Condition number and RGA

The linearized system can be instead analyzed using singular value analysis [17] to estimate a controllability metric and Bristol's Relative Gain Array Method [3, 15]. This was done to determine the level of interaction between loops and to verify that the input-output pairings chosen was reasonable. The *condition number* is given by:

$$\gamma(H_0) = \frac{\bar{\sigma}(H_0)}{\underline{\sigma}(H_0)} \qquad \text{with} \qquad H_0 = \lim_{s \to 0} sH(s) \tag{5.10}$$

where $\bar{\sigma}(H_0)$ and $\underline{\sigma}(H_0)$ are, respectively, the maximum and minimum singular values of H_0. A large condition number (Skogestad suggests larger than 10) indicates that the system might be hard to control. On the opposite, a small condition number indicates that the system is insensitive to multi-variable disturbances. Using the MATLAB commands `cond(H0)` returns $\gamma(H_0) = 3.299$. Based on this, the system in its current configuration is of the latter type.

The Relative Gain Array (RGA), Λ, is traditionally used as an interaction measure between the loops of a multi-variable system and as a tool to find the best input-output pairings. The RGA is given by:

$$\Lambda = H_0 \otimes (H_0^{-1})^T = \begin{bmatrix} 1.00 & -5.815 \cdot 10^{-6} & 2.319 \cdot 10^{-6} \\ -5.637 \cdot 10^{-8} & 1.00 & 2.855 \cdot 10^{-8} \\ 7.883 \cdot 10^{-7} & 1.260 \cdot 10^{-6} & 1.00 \end{bmatrix} \tag{5.11}$$

Where each row corresponds to an output and each column corresponds to an input. Ideally, the inputs and outputs should be coupled so that the relative gain is close to one. In this case it is trivial to see that pairing u_x with y_x, u_y with y_y and u_z with y_z is close to optimal.

5.4 Digital PID design

The generalized form of a continuous-time PID controller is described by the equation 5.12.

$$u(t) = K_P \cdot e(t) + K_I \int_0^t e(t)\, dt + K_D \cdot \frac{de(t)}{dt} \tag{5.12}$$

When implementing this controller algorithm into a digital microcontroller, the algorithm needs to be discretized to an incremental equivalent form, described by the equation (5.13). See [28] for more details.

$$\Delta u[n] = u[n] - u[n-1] = \Delta u_P[n] + \Delta u_I[n] + \Delta u_D[n] \tag{5.13}$$

where

$$\begin{aligned} \Delta u_P[n] &= K_P \cdot \Delta e[n] = K_P \cdot (e[n] - e[n-1]) \\ \Delta u_I[n] &= K_I \cdot e[n] \\ \Delta u_D[n] &= K_D \cdot (\Delta e[n] - \Delta e[n-1]) \\ &= K_D \cdot (e[n] - 2 \cdot e[n-1] + e[n-2]) \end{aligned} \tag{5.14}$$

Putting all together gives the full incremental PID-algorithm:

$$\begin{aligned} u[n] = u[n-1] &+ K_P \cdot (e[n] - e[n-1]) + K_I \cdot e[n] \\ &+ K_D \cdot (e[n] - 2 \cdot e[n-1] + e[n-2]) \end{aligned} \tag{5.15}$$

At the end was decided to use a digital PD-controller. This was mainly done based on the fact that the controller needs to be as rapid as possible, in order to correct the position of the levitating magnet. Implementing a PD-controller in the code can be done easily setting the integral parameter $K_I = 0$.

5.5 Tuning

Since the system is non linear, there weren't found any straightforward method to tune the controllers. It was not possible to use any of the standard methods e.g. the Ziegler & Nichols method or Skogestads SIMC. The parameters for the proportional and derivative gain where therefore found by the previous team with a systematic trial-and-error method.

Unfortunately, the fine-tuning process was much more complicated than expected and did not lead to appreciable results. During the testings the system remains stable for a couple of seconds before starting to oscillate until the levitating magnet falls to the side. Even the smallest changes in the system (magnet weight, disturbance, sensor placement) and in the values of the controller's parameters make an enormous difference in stability, so its not a given that the old values found by the previous team will work for this system, unfortunately. A tuning process such as this would in fact require much more time than we had available and a more accurate analysis. The reasons for this difficult and uncompleted phase will be discussed in more detail in chapter 8.1. Due to time constraints of the project and with the knowledge that the primary goal of this project was the re-design and re-building of the platform for the next team (see chapter 8.11) it was finally decided to abandon the fine-tuning attempts.

6 Code

This chapter will present all the design choices made about the microcontroller settings, along with a description of the critical steps in the code. Different versions of parts of the code have also been developed, explained and discussed here with their pros and cons. The testing routines used in the chapter 7 will also be introduced here.

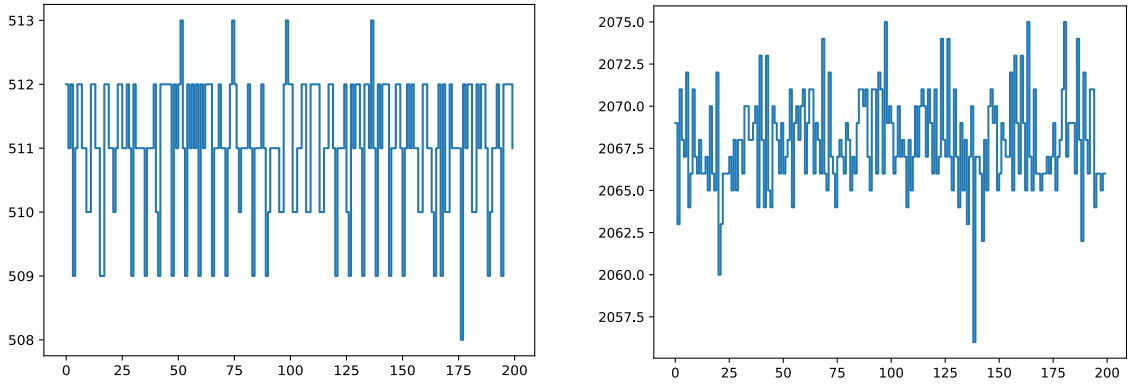
6.1 Microcontroller's settings

The microcontroller used is a Teensy4.0. It was chosen mainly for its speed and the fact that it runs *Teensyduino*, a plugin for the Arduino IDE. There are several settings about the Teensy that can be changed, to suit different purposes. Some settings have been left at the default values, while others have been changed to suit the system or to make it run more smoothly.

6.1.1 ADC

Regarding the sensor readings, although the ADC can be set to a reading resolution of up to 12-bit, it was left set to 10-bit. This is due to the fact that the 12-bit readout presents more noise and slows down the system further. This is also demonstrated in the Teensy data sheet (see [16], page 64), as it shows that the actual number of bits is 10.7. To further reduce noise, the read averaging has been set to 4. This means that the controller reads the sensor data four times and

averages it before giving the input to the program. A comparison between a 10-bit reading and a 12-bit reading, with an average of 4, is shown in figure (6.1).

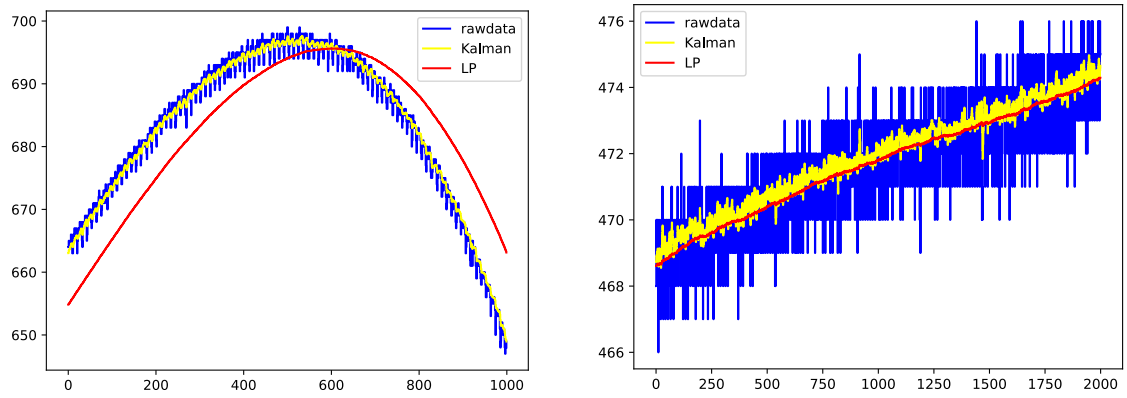


(a) 10-bits (b) 12-bits

Figure 6.1

6.1.2 Filter design

The input readings of the sensors contain noise. It was therefore necessary to filter out this noise in order to get the most accurate readings possible. Two different filters were implemented to see which type would fit the system: an LP filter (LinnesLab 2020, see [10]) and a simple Kalman filter (denyssene 2020, see [4]). The tests, shown in figure 6.2 highlight that the LP filter may have more stable readings, but also adds a delay of about 100 samples. This is equivalent to a delay of 3.2 ms, which could be detrimental to the control of the system. The one used in the final implementation of the code is the simple Kalman filter.



(a) Test 1 (b) Test 2

Figure 6.2: Filter analysis

6.1.3 PWM settings

The default PWM settings are a frequency of 4482 Hz [12] with a resolution of 8 bits. However, the total controller execution time is approximately $t_c = 32\ \mu s$ (measured using simple time functions in the code). This means that the controller can send different outputs at approximately $f_c = 1/t_c \approx 31.250$ kHz. Due to this speed discrepancy, the frequency of the signal PWM was increased to 31.250 kHz. Although it could have been increased more, this has not been done due to the limitation of the motor drivers to 40 kHz.

6.2 Code description

The code is organized in several modular sections, both to make the code cleaner and more easily understandable and to facilitate the editing and re-writing of its different sub-parts. They consist of:

- defines.h – it contains all pin definitions, together with the constants used in the code and the pid parameters;
- attract_repel.ino – basic definitions of which motor direction means repel() and which attract() according to the orientation of the system's magnets and solenoid connections (shown in listing 6.1);
- turn_X.ino – contains all developed versions of turn_X available for the use in main.ino (see section 6.2.2);
- turn_Y.ino – as above, but for the y-axis;
- main.ino – explained in section 6.2.3.

6.2.1 attract_repel.ino

```
1 /*
2  * with the current setup based on the coil connection made [A-B = EXT-INT] and
3  * the levmag orientation [NORTH POLE pointing up], a repulsion is made setting
4  * the coil with a SOUTH POLE on the top, aka setting A-B = HIGH-LOW. An attr-
5  * action is instead made setting A-B = LOW-HIGH, creating a NORTH-POLE on the
6  * top. Change the repel() and attract() functions depending on the setup used.
7  */
8
9 //SOUTH-POLE on the TOP
```

```
10 void repel(byte ca, byte cb, byte c_enable, int value){
11   digitalWrite(ca, HIGH);
12   digitalWrite(cb, LOW);
13   analogWrite(c_enable, value);
14 }
15
16 //NORTH-POLE on the TOP
17 void attract(byte ca, byte cb, byte c_enable, int value){
18   repel(cb,ca,c_enable,value);
19 }
```

Listing 6.1: attract_repel.ino

6.2.2 TURN FUNCTIONS

Two different versions of the code were developed for each axis, one with the suffix *_OLD* and the other with the suffix *_NEW*, which in turn was developed into a debug version using Serial.print() commands and a final performing version without them (identified with the additional suffix *_final*).

The *_OLD* version always sets the coils always in a dual behaviour (one R-epel and the other A-ttract) by giving priority to the value u_x and adding u_z afterwards. When u_z is much greater than u_x, this can lead to to the undesired situation in which one of the two PWM values saturates to 0, as explained and discussed later. The algorithm first, based on the sign of u_x, chooses whether coil X1 should R-epel or A-ttract, while coil X2 will do the opposite. The following table illustrates the behaviour (R/A) and PWM values of the coils according to the sign of u_x.

Condition	**Coils behaviour**
$u_x \geq 0$	X1-R $= +u_x + u_z$ X2-A $= +u_x - u_z$
$u_x < 0$	X1-A $= -u_x - u_z$ X2-R $= -u_x + u_z$

Table 6.1: Turn_X_OLD algorithm table

The code snippet is here below showed in listing 6.2.

```
1 ...
2 void turn_X_OLD_final(int ux, int uz)
3 {
4     if(ux≥0) // X1-R = +ux+uz # X2-A = +ux-uz
5     {
```

```
6           value = +ux +uz; value = constrain(value, 0, 255);
7           repel(X1_A,X1_B,ENABLE_X1,value); //X1-R
8
9           value = +ux -uz; value = constrain(value, 0, 255);
10          attract(X2_A,X2_B,ENABLE_X2,value); //X2-A
11      }
12
13      else // (ux<0) X1-A = -ux-uz # X2-R = -ux+uz
14      {
15          value = -ux -uz; value = constrain(value, 0, 255);
16          attract(X1_A,X1_B,ENABLE_X1,value); //X1-A
17
18          value = -ux +uz; value = constrain(value, 0, 255);
19          repel(X2_A,X2_B,ENABLE_X2,value); //X2-R
20      }
21  }
22  ...
```

Listing 6.2: turn_X_OLD.ino

On the other hand, the *_NEW* version first calculates the sums $+u_x + u_z$ for X1 and $-u_x + u_z$ for X2 (dual action for the x-axis but concordant for the z-axis). The two results are then constrained between -255 and +255 and, depending on the sign, if the result is ≥ 0 the corresponding coil will repels with value = +res, otherwise (< 0) the coil will attract with value = $-$ res. This means that in some cases both coils can A-attract or R-repel. The following table describe the possibilities, which, as can be seen, this time are four instead of two.

s_1 \ s_2	< 0	≥ 0
< 0	X1-A = $-u_x - u_z$ X2-A = $+u_x - u_z$	X1-A = $-u_x - u_z$ X2-R = $-u_x + u_z$
≥ 0	X1-R = $+u_x + u_z$ X2-A = $+u_x - u_z$	X1-R = $+u_x + u_z$ X2-R = $-u_x + u_z$

Table 6.2: Turn_X_NEW algorithm table

The code snippet is here below showed in listing 6.3.

```
1  ...
2  void turn_X_NEW_final(int ux, int uz)
3  {
4      int sumx1 = +ux +uz; sumx1 = constrain(sumx1,-255,+255);
5      int sumx2 = -ux +uz; sumx2 = constrain(sumx2,-255,+255);
6
7      if(sumx1≥0)
8      {   repel(X1_A,X1_B,ENABLE_X1,sumx1);   //X1-R
```

```
9       }
10      else
11      {   sumx1 = abs(sumx1);
12      attract(X1_A,X1_B,ENABLE_X1,sumx1);      //X1-A
13      }
14
15      if(sumx2≥0)
16      {   repel(X2_A,X2_B,ENABLE_X2,sumx2);    //X2-R
17      }
18      else
19      { sumx2=abs(sumx2);
20          attract(X2_A,X2_B,ENABLE_X2,sumx2); //X2-A
21      }
22  }
23  ...
```

Listing 6.3: turn_X_NEW.ino

The more intuitive and proper of the two is clearly the *_NEW* version, but, from what the previous team reported, the *_OLD* one led to better stability performance, so we decided to report it for completeness. To better understand the differences between the two versions, let us use an example:

```
1  EXAMPLE 1: the levmag is slightly moved towards X1-coil and much lower than z_eq
2             ux=5, uz=50 [uz >> ux]
3
4  _OLD ### ux≥0 => X1-R while X2-A
5                   set X1-R with +5+50 = +55 => clamp (ok)   => R with pwm = +55
6                       X2-A with +5-50 = -45 => clamped to 0 => A with pwm = 0
7
8  _NEW ### sumx1 = +5+50 = +55 => clamp (ok) => R with pwm = +55
9           sumx2 = -5+50 = +45 => clamp (ok) => R with pwm = +45
10
11 ---
12
13 EXAMPLE 2: ux=50, uz=5 [ux >> uz]
14
15 _OLD ### ux≥0 => X1-R while X2-A
16                  set X1-R with +50+5 = +55 => clamp (ok) => R with pwm = +55
17                      X2-A with +50-5 = +45 => clamp (ok) => A with pwm = +45
18
19 _NEW ### sumx1 = +50+5 = +55 => clamp (ok) => R with pwm = +55
20          sumx2 = -50+5 = -45 => clamp (ok) => R with pwm = +45
```

As said before, the two versions acts in the same way when $u_x \gg u_z$, whereas they act quite differently if $u_x \ll u_z$. Both versions are available in its current file so you can choose which one to use by changing a single instruction in the main.

6.2.3 MAIN

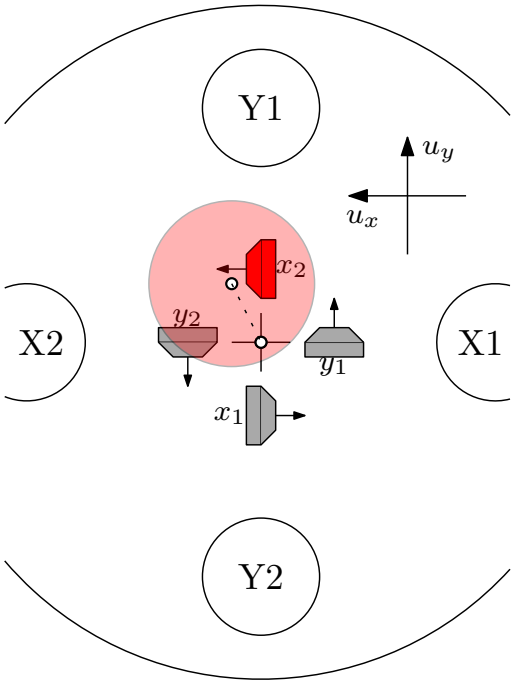


Figure 6.3: Weighted-average sensor estimation

The main.ino file is richly commented and needs no special explanation. After an initial setup as explained in section 6.1, it reads and converts the voltages coming from the amplification stage into their corresponding gauss values. The values from the sensors $x_1 \dots z$ are, in order, stored in the array `dv_in[]`, and are those that would ideally be used to calculate the three u_x, u_y, u_z values by a simple average. In reality, however, it was noted that the sensor of the pair closer to the levmag reacted slightly better than their counterparts further away. It was then decided to use a weighted average instead of a simple one, prioritising the closest sensor of the pair. To estimate which sensor between x_1 and x_2 prioritize the y ones were used, as shown in listing 6.4. In figure (6.3) there's an example with the levmag slightly off centered towards Y1. The y-sensors are used to estimate this deviation to prioritize x_2 sensors.

```
1  ...
2  // use y to estimate the X-weights [how much priority give to x2 than to x1]
3  dwx2 = (dv_in[2] - dv_in[3]) * 0.5 / 0.4;
4  wx2 = 0.5 + dwx2 * 0.5;
5  wx1 = 1 - wx2;
6
7  // instead use x to estimate the Y-weights [how much priority give to y1 than to y2]
8  dwy1 = (dv_in[0] - dv_in[1]) * 0.5 / 0.4;
9  wy1 = 0.5 + dwy1 * 0.5;
10 wy2 = 1 - wy1;
11 ...
12 u[0] = dv_in[0]*wx1 - dv_in[1]*wx2;
13 u[1] = dv_in[2]*wy1 - dv_in[3]*wy2;
14 u[2] = dv_in[4];
15 ...
```

Listing 6.4: Weighted average

Another relevant thing to be discussed is this output limitation done just before passing the values to the turn functions:

```
1 \\ give priorities to the controller along the axis
2 \\ [contrain between [-255≤-LIM:+LIM≤255]
3 ux[0] = round(max(ux[0],MAX_OUTPUT_X);
4 uy[0] = round(max(uy[0],MAX_OUTPUT_Y);
5 uz[0] = round(max(uz[0],MAX_OUTPUT_Z); //100
```

Listing 6.5: Output limitation

The reason for this was to make sure that the solenoids had a greater controllability along the xy-plane. The z-output was saturated to approximately 40%(±100) of the maximum output (±255), in order to make the effects of the x and y controllers greater.

6.3 Testing routines

Besides the primary code, two specific test routines were also developed. The first, `potcalibration.ino`, is dedicated to sensory reading, and is a tool for adjusting potentiometer references while reading their values. Used while the levmag is centererd on $[x_m, y_m, z_m]^T = [0, 0, z_{eq}]^T$ allows the references to be set correctly, making sure that the readings of `dv_in[]` are as close to zero as possible. It also reads the voltage reference value from pin 20, to adjust this as well.

The second tool developed tests the coils and the behaviour of the levmag separately. It allows the solenoids to be freely set to ensure that the wiring and the conventions of repel() and attract() functions are correctly implemented.

All the codes, along with the testing routines, can also be found on the Github Repository `https://github.com/albertomors/maglev22` in the folder named `code`.

7 Testing

This chapter outlines the procedures to be followed to set up the various electronic sub-parts and the order to be followed to do it, discusses the choices made and explain how they can and should be changed in the future if other choices are to be made. The order to be followed is as below:

1. set-up and measurement of power supplies;
2. test of the zener diodes protection stage;
3. adjustment of the voltage reference;
4. decision of reading range, determining the in-amps gain and last settings in the code;
5. adjusting of the in-amps gain;
6. connection of the Teensy4.0 to the circuit;
7. calibration of the references (potentiometers);
8. test of the coils;

7.1 In detail

As said, the first thing to do is to adjust the two LM2596S using their potentiometers to obtain and measure 5 and 3.3 V on their output stage, respectively. Then the reference voltage must be adjusted to the center of the reading range of the ADC. Ideally, this value is 3.3/2 = 1.65 V, but after testing the zener diode

protection stage we discovered that it introduces a slight saturation when voltages approach 3.3 V. In fact, the maximum saturation value reached is not 3.3 but 3.168 V, measured under saturated output conditions. Consequently, in order to make the best and most symmetrical use of the effectively used ADC range, it was decided to set the voltage reference to a value of 3.168/2 = 1.584 V. In order to do so, a simple voltmeter can be used, as well as the specific program developed to do so.

7.2 Optimal gain

By moving the magnet while measuring the sensor voltages on the V_{IN-} pins of the in-amps, it is possible to get an idea of the estimation capability of the sensors. Although the simulations showed the levmag can be brought back up to a maximum distance of 4 cm (see section 3.7), we have to keep in mind that the simulation only shows the $[x_m, y_m, z_m]$ state variables, and shows them independently, but does not describe *how* it is brought back to the centre. It does not in fact provide us with information on the other 9 state variables of the levmag, so there is certainly a risk of the levmag flipping, rotating, wobbling and so on, as demonstrated experimentally. It is therefore not "controllable", or rather it is not "controllable" in this way[1], on the other 9 state variables.

The optimal solution therefore seems to choose an operating range (both of measurement and of controllability of the levmag) around the equilibrium point that is small enough to preserve roughly the properties of the equilibrium point, but large enough to accommodate and be able to measure the oscillations and movement of the magnet that the control fails to correct. The area chosen is therefore a circle of radius 1 cm around the center. Measuring now the semi-deviation of the sensor voltage at 1 cm from the center with respect to the value it has at the center allow us to find the optimal value of the in-amps gain, in order to fully benefit from the reading range of the ADC. The deviation from the center is measured to be $\approx 0.45\ V$ and the optimal gain can be found according to the equation:

$$\begin{aligned} \Delta v_{out_{MAX}} &= 1/2 \cdot \mathrm{ADC}_{\mathrm{eff\_RANGE}} = 3.168\ V/2 (= \mathrm{vref}) \\ &= 1.584\ V = G \cdot \Delta v_{in_{0--1\ cm}} = G \cdot 0.45\ V \end{aligned} \tag{7.1}$$

[1] with the control strategies implemented, and with the controller used

where the result is a gain $G = 1.584/0.45 = 3.52$ that can be realized adjusting the external potentiometers connected between pins 1 and 8 of the in-amps according to the equation:

$$G = 1 + \frac{100\ k\Omega}{R_G} \implies R_G = \frac{100\ k\Omega}{G-1} = \frac{100\ k\Omega}{2.52} \approx 39.68\ k\Omega \tag{7.2}$$

Several other ways are potentially correct and should be tested. With the current choices, the movements of the magnet over 1 cm from the center saturate at the same voltage value as when it is at 1 cm. Increasing the chosen operating zone means having a wider reading range at the cost of a lower resolution on small deviations from the center, because the gain must be decreased accordingly.

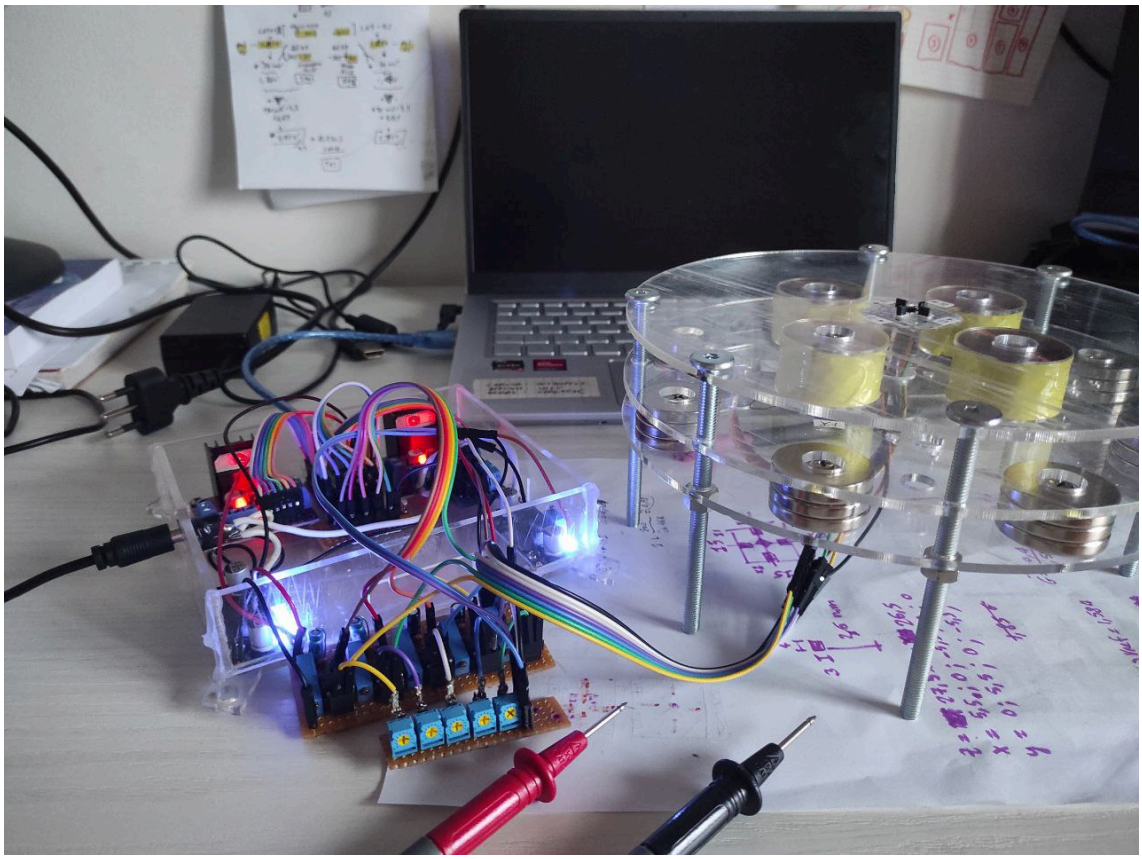


Figure 7.1: Testing the electronics

There are two ways to adjust the gain of the in-amps:

1. Measure the resistance value (for example with an ohmmeter) between pins 1 and 8 of the in-amps socket (without the in-amps connected) and adjust the potentiometer until the correct value is read.
2. Setting a small voltage value (small enough not to saturate the output with the chosen gain), adjust the potentiometer until the differential output (which can be measured with a voltmeter by placing the two test leads on v_{out} and v_{ref}) will be exactly $G \cdot v_{IN}$.

Lastly, the code has to be adjusted accordingly, setting the value `i2o` (inside defines.h) equal to the set gain.

7.3 Sensor testing

At this time the Teensy can be connected to the circuit, after ensuring that everything works in the right way. Figure (7.2) shows a simple test of the sensor board manually moving the levitating magnet to check that the dual sensor behaviour is working properly.

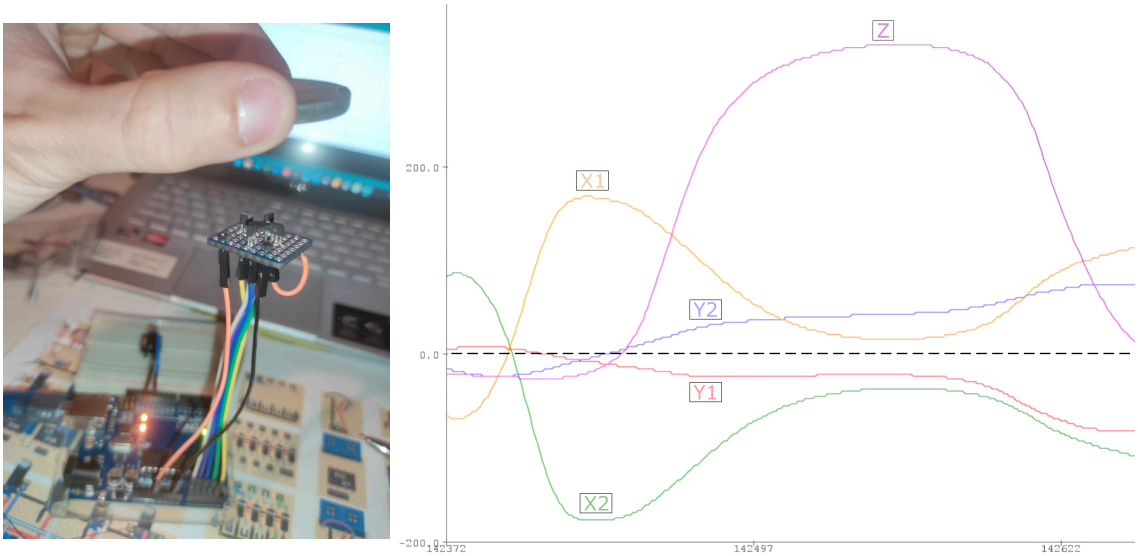


Figure 7.2: Testing the sensor board

7.4 References calibration

The potentiometers calibration and the test of the coils can be easily done with the scripts developed, described by the previous chapter. Since the adjusting of

the references has to be done with the levmag in $[x, y, z] = [0, 0, z_{eq}]$ a magnet holder with an height of z_{eq} was specifically made to help centering the magnet during this phase, as can be seen in figure (7.3).

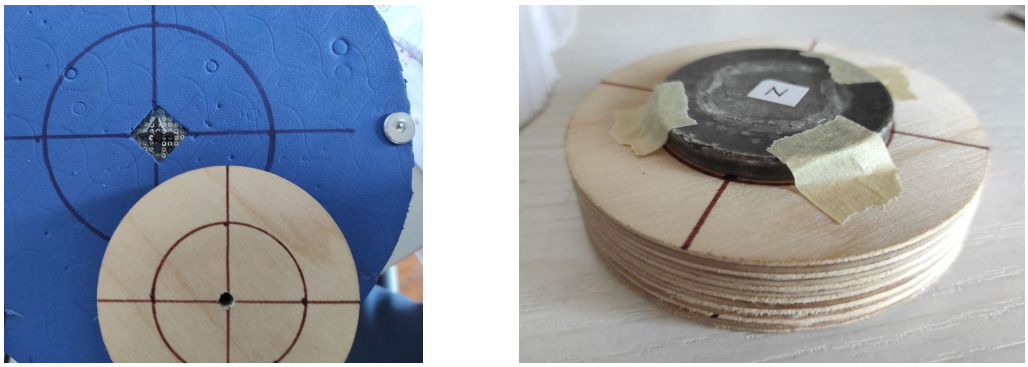


Figure 7.3: Magnet holder

The 10 kΩ potentiometers used due to space constraints among those available are of the single-turn type and consequently hardly adjustable to a high level of accuracy. With much effort, however, we were able to obtain a value with an absolute error of ± 2 mV on the equilibrium point for all five sensors. An external board (shown in figure 7.4), was then realized to be connected in parallel to the existing one, using the multi-turn 200 kΩ potentiometers used for in-amps. The much higher resistance value and the different type together allow a much more precise regulation of the reference voltages, with an achieved result of ± 0.1 mV.

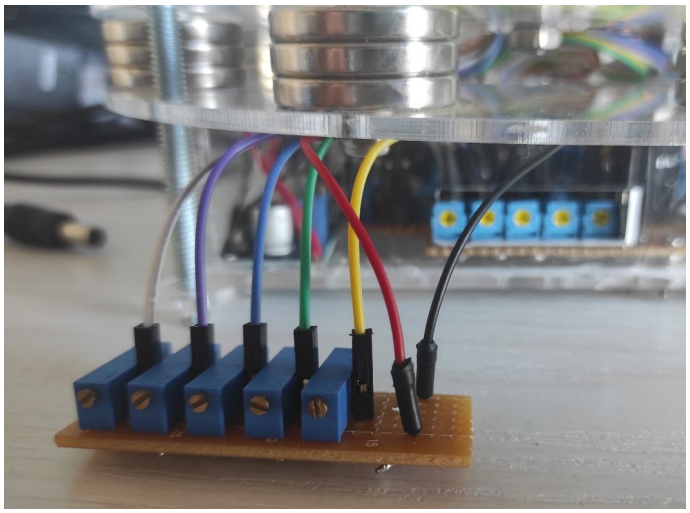


Figure 7.4: External reference board

We suggest purchasing and replacing the existing 10 kΩ single-turn potentiometers with multiple-turn ones in the future.

8 Discussion

This chapter discusses all the design choices that were not changed from the previous project due to time constraints or impossibility to change them, along with those made in the current re-design. It also discusses all the suggested improvements to be implemented in the future, which were only noticed during the last testing phase.

8.1 Tuning difficulties

As said before the tuning of the controller was not completed. The first trial-and-error attempts were soon abandoned after a few days of failed trials, to search for the reason of the excessive instability in other areas. Attempts have also been made with MATLAB PID Tuner, which, however, failed to stabilize the function $H(s)$. Using a different control structure could improve this problem, for example by using a cascade control. The internal feedback loop could then stabilise the poles of the right half-plane while an external loop controller gives the system the desired characteristics.

8.2 Mathematical model

The mathematical model itself has been discussed, because although it seems to represent the physical system well, it has not been systematically tested to confirm it. With more time available and a stable system model, testing would

have been a priority, as a good model is crucial for the further development of the platform. With more time and more accurate sensors, the modelling results could be systematically verified by making a test setup in which the magnetic field produced by the magnets is measured with hall-effect sensors. The same setup can be then recreated with the `maglevSystem` class and, by implementing virtual sensors in the same places as the real ones, the physically measured and simulated magnetic field values can be compared.

8.3 Levitating magnet damages

Another topic worth to be analized is that the problem might actually reside in the physical properties of the levitating magnet, because during the previous project, the levitating magnet suffered some superficial damage. As reported, every time new controller parameters were tested, the magnet, initially positioned in the centre, would eventually fall out. The strong magnetic field in some cases caused the magnet to violently clash with the top plexiglass plate, causing some minor damage to the magnet (see figure 8.1). The potential damages to the levitating magnet were eventually mitigated by the implementation of a soft mat over the top layer.

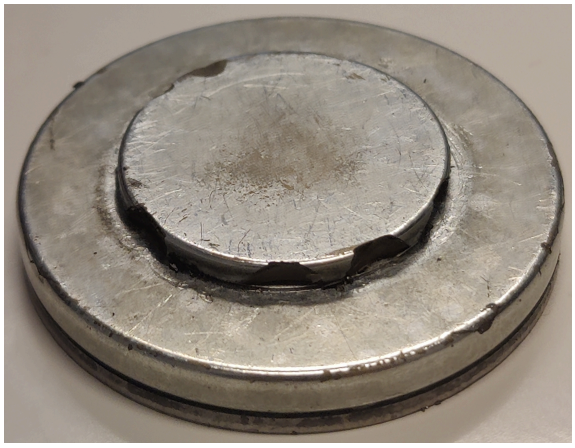


Figure 8.1: Damages on the levitating magnet

8.4 Structure of the levitating magnet

During the current project, however, in one of the first tests the levitating magnet split into three pieces, revealing it to be made up of several parts. The old

levitating magnet consisted of a middle magnetic disc of larger radius (the one that still remains), a lower magnetic disc of smaller radius, and a glued-on upper plate that appeared to be diamagnetic or ferromagnetic, shown in figure (8.2).

Figure 8.2: Old levitating magnet structure vs. the current one

Unfortunately the lower disc broke into several pieces, and cannot be recovered, while the top plate just slipped away. The idea that the problem may be the changed dynamics of the levitating magnet no longer being "appropriate" comes from online researches for similar magnetic levitation platforms, all showing that typical structure of the levitating magnet. Such a structure may possibly facilitate its control with a similar system, certainly modifying its inertia and suchlike (and if the intuition about the top plate is correct, also the electromagnetic properties).

To prove this, some tests were carried out with a neodymium magnet attached under what remains of the levitating magnet (the central disc) that revealed a stronger stability of the magnet in the center, which seems to obstruct it from getting out of the equilibrium, but at the same time increases the difficulty in controlling it, as bigger forces are needed to correct its movements.

8.5 Solenoids analysis

The screws use to fix the solenoids to the plexiglass plate had an important effect on the strength of the magnetic field generated by the solenoids. Having no information on the type of steel the screws were made of, it was rather difficult to understand the permeability of the core. However, the use of the screws with the presumably higher permeability added some problems to the system: the levitating magnet was attracted to the screws, changing the behaviour of the magnetic field. The greater the permeability, the greater the inductance of an electromagnet and the response time of electromagnets. This can seriously affect the response speed of the controller, which, however fast it is, has to contend with the much slower electromagnets.

8.6 Observability constraints and noise involved

The fact that the sensor reading must be restricted to a chosen area to improve accuracy for small deviations from the centre makes the controller less flexible. The sensors also read oscillations along an axis and movements along it in the same way, and there is no way to discern these two different behaviours of the levitating magnet.

The noise generated by the solenoids more or less alters the sensor readings. The measured electromagnetic actions should be analysed further by coding a magnetic field reader that checks all possible combinations of solenoid inputs while reading the sensor values. Storing these values in a function to be used on the PID controller to subtract the solenoids electromagnetic effects from the sensor readings could give better performance and reduce the noise involved.

8.7 Alternatives to sensors chosen

The read and control actions should be separated into two physically different channels in order not to alter the readings with controller corrections. Some suggested sensors are for example ultrasonic sensors or, even better, a camera. A camera is one of the most compact, advanced, and versatile sensors that could be used for this purpose. It could in fact estimate all 12 state variables of the magnet (allowing full state feedback) using specific image-processing techniques. This though would levitate also the cost of the platform.

8.8 Controllability analysis

The wide radius of the circle where the permanent magnets are arranged, chosen by the previous team, causes the magnetic field to have enough space to fold in on the center, which is why the magnet must be placed upside down (north pole pointing upwards, like the permanent ones) to find the unstable balance chosen as the operating point. This also changes the control strategies required to control levmag along the z-axis, so that what we call repulsion along the x- and y-axes is attraction along the z-axis. This greatly reduces controllability when x-, y- and z-axes are controlled together by the same solenoids (as in our case), cause the two actions partially cancel each other out. A strategy that might

work could be a solenoid in the center to control only the z-axis, but this would make it impossible to place the sensors in the center, unless they were placed on two different levels and the technique explained in section 8.6 is used to clean the readings from the electromagnetic action of the z-solenoid.

One idea attempted to solve this problem, during the testing and development period, was to use only two controllers instead of three. That is, excluding the z-controller and concentrating only on stabilizing a two-dimensional system. Unfortunately, no differences were found in the instability of the system, both with and without the z-controller. With hindsight, it would be wise to focus on two-dimensional control and stabilization before moving towards z-axis control.

Another fact to think about is whether the dual control strategy chosen is the right one. The choices made in fact reduce the degrees of freedom on solenoid control from 4 to 3. With more sophisticated techniques and more time available this would need to be explored in more detail.

8.9 Potential improvements regarding electronics

Solenoids become very hot when the system is left on for a while, so much that the screws with which they are attached to the plexiglass by thermal expansion come loose. To prevent the system from overheating in the future, it is recommended to install cooling fans to take heat out from the solenoid level. They are typically built to operate on 12 V, perfect as we have this power supply available in our circuit.

Another suggested improvement is the design and realization of a pcb. This could improve to better fasten the connection points, reduce the risk of shorts and to potentially reduce noise across the circuit.

The last suggested improvements concern the replacement of certain components. First of all, the voltage reference could be implemented in the way explained in chapter 4.4.5 and the single-axis hall-effect sensors could be replaced by an individual 3-axis hall-effect sensor available on the market in a typical SMD packaging, so a pcb has to be realized to improve this. A printed circuit board design could also allow in-amps to be replaced by INA326s to reduce circuitry, electronics costs and the time spent assembling the various boards. Finally, as already mentioned, we recommend replacing the potentiometer board with some multi-turns to improve the achievable accuracy of the references.

8.10 Suggestions for improving the controller

The linearization process is only valid in the equilibrium point. One idea to discuss is, due to the symmetry of the system, to linearize the system at a few relevant points around the equilibrium, and possibly design a versatile controller with different parameters designed specifically for each point.

Another strategy based on the same idea is to tune (successfully) the PID controller with two different types of parameters to obtain a smooth version (to be used when the levitating magnet is close to equilibrium) and an aggressive one (to be used when the distance increases and the necessary corrections need to be more decisive). Depending on the situation, the PID controller can then change its parameters k_P, k_I, k_D between those of the smooth version and those of the aggressive version

8.11 Future work

There is reason to believe that the implementation of a state observer would greatly improve the control performance of the system. Previous studies and simulations on the subject have shown that if there is at least one other sensor in the system that is not centrally located, a high-gain non-linear observer may theoretically be able to estimate all the states, which could allow for full state feedback, thus enabling the implementation of a linear-quadratic regulator (LQR) controller, which in a system such as this one can greatly improve performance.

This idea is currently being developed by another student, working on this same project. His work includes a Kalman filter for state estimation (all states except angle of rotation φ_m and angular velocity $\dot{\varphi}_m$ along z-axis, which cannot be observed with the sensors used) consequently used by an appropriately designed LQR controller, allows the levitating magnet to be stabilized on the equilibrium point along all observable states. It is hoped in the future to be able to implement this controller, once finished, on the system designed in this thesis.

Throughout the writing of this thesis, the internal document *Model Description* is also receiving a revision, that replaces the equivalent current with a much more accurate equivalent surface current, based on the magnetic polarization of permanent magnets. The model now also includes the precession of the levitating magnet (previously not considered). The first numerical simulations performed

show that the consistency between the two approximations is now much greater. Hopefully, this work will be completed in the future and integrated into the next studies and re-design of the project.

9

Conclusions

The primary goal of this project, as described in the introduction, was the re-design and re-building of an existing original magnetic levitation platform. In doing so, we sought to provide precise design guidelines for the next team that will continue this development process. Although many of the initial goals have been achieved, there is still much that can be done in the various areas.

The process of studying, comprehending, modelling and simulating the system took much more time than expected, due to initial difficulties with the documentation from the previous team, located in Norway, and a lack of precise guidelines on the specifics on the project. This issue generated the necessity for a comprehensive guide – something that this thesis hopes to achieve – to explain the project to a consequent team of developers. The electronic design phase and the search for the optimal techno-economic solution went more or less as planned, and led to the results described in the previous chapters. Unfortunately, it was not possible to find reasonable parameters for the controller, due to time constraints to wrap up the project, but given the initial goals of this thesis and the initial condition from which this work started we feel satisfied with the outcome.

The platform therefore has all the fully functional systems working in order to implement a controller, but is currently unable to stabilize the levitating magnet for more than a few seconds. We believe that a version with better stability properties can be achieved by continuing with further tests, corrections and improvements to the state observer and levitation controller algorithms.

We note that we underestimated the complexity and the time initially thought

this project would take, but we also consider it an invaluable learning experience that has allowed to develop (or improve) working methodologies, abilities and know-how in a cross-disciplinary framework that ties together electronics, numerical computing, embedded programming, handwork and control systems.

References

[1] Migdonio Alberto González and Dorindo Elám Cárdenas. "Analytical Expressions for the Magnetic Field Generated by a Circular Arc Filament Carrying a Direct Current". In: *IEEE Access* 9 (2021), pp. 7483–7495. DOI: 10.1109/ACCESS.2020.3044871.

[2] *Analogue to Digital Converter*. https://www.electronics-tutorials.ws/combination/analogue-to-digital-converter.html. [Online; accessed 1-November-2022].

[3] E. Bristol. "On a new measure of interaction for multivariable process control". In: *IEEE transactions on automatic control* 11.1 (1966), pp. 133–134.

[4] denyssene. *Simple Kalman Filter Library*. https://github.com/denyssene/SimpleKalmanFilter (Accessed: 27.10.2022). 2020.

[5] R. Doshmanziari, H.A. Engmark, and K.T. Hoang. *Model Description: Magnetic Levitation System*. https://folk.ntnu.no/hansae/Maglev_System_Description.pdf. [Online; accessed 18-October-2022]. 2021.

[6] *Dynamic Systems and Control Linearization of Nonlinear Systems*. https://www.control.utoronto.ca/~broucke/ece311s/Handouts/linearization.pdf. [Online; accessed 1-November-2022]. 2007.

[7] *Equilibrium*. http://www.scholarpedia.org/article/Equilibrium. [Online; accessed 1-November-2022].

[8] *Hall Effect Sensor*. https://www.electronics-tutorials.ws/electromagnetism/hall-effect.html. [Online; accessed 1-November-2022].

[9] *How to find the transfer function of a system*. https://x-engineer.org/transfer-function. [Online; accessed 1-November-2022].

[10] LinnesLab. *KickFiltersRT*. https://github.com/LinnesLab/KickFiltersRT (Accessed: 18.05.2022). 2020.

[11] *Operational Amplifier Basics*. https://www.electronics-tutorials.ws/opamp/opamp_1.html. [Online; accessed 1-November-2022].

[12] PJRC. *Teensyduino, Pulsed Output: PWM & Tone*. https://www.pjrc.com/teensy/td_pulse.html (Accessed: 27.10.2022). 2022.

[13] *Potentiometer*. https://eepower.com/resistor-guide/resistor-types/potentiometer. [Online; accessed 1-November-2022].

[14] Rainer Goebel. *Rotation matrix — Wikipedia, The Free Encyclopedia*. https://www.brainvoyager.com/bv/doc/UsersGuide/CoordsAndTransforms/SpatialTransformationMatrices.html. [Online; accessed 20-October-2022]. 2020.

[15] D. E. Seborg et al. *Process Dynamics and Control*. Wiley, 2016, pp. 331–338.

[16] NXP Semiconductors. *i.MX RT1060 Crossover Processors for Consumer Products*. https://www.pjrc.com/teensy/IMXRT1060CEC_rev0_1.pdf (Accessed: 27.10.2022). 2019.

[17] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control*. Wiley, 2007.

[18] *State Space Representations of Linear Physical Systems*. https://lpsa.swarthmore.edu/Representations/SysRepSS.html. [Online; accessed 1-November-2022].

[19] Z.J. Wang and Yong Ren. "Magnetic Force and Torque Calculation Between Circular Coils With Nonparallel Axes". In: *Applied Superconductivity, IEEE Transactions on* 24 (Aug. 2014), pp. 1–5. DOI: 10.1109/TASC.2014.2311412.

[20] *What Are Zener Diodes?* https://www.allaboutcircuits.com/textbook/semiconductors/chpt-3/zener-diodes/. [Online; accessed 1-November-2022].

[21] *What is an Instrumentation Amplifier? Circuit Diagram, Advantages, and Applications*. https://www.elprocus.com/what-is-an-instrumentation-amplifier-circuit-diagram-advantages-and-applications/. [Online; accessed 1-November-2022].

[22] Wikipedia contributors. *BiotSavart law — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Biot%E2%80%93Savart_law&oldid=1118579430. [Online; accessed 1-November-2022]. 2022.

[23] Wikipedia contributors. *Buck converter — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Buck_converter&oldid=1119236883. [Online; accessed 1-November-2022]. 2022.

[24] Wikipedia contributors. *Control theory — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Control_theory&oldid=1111121972. [Online; accessed 1-November-2022]. 2022.

[25] Wikipedia contributors. *Lorentz force — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Lorentz_force&oldid=1113983259. [Online; accessed 1-November-2022]. 2022.

[26] Wikipedia contributors. *Magnetic levitation — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Magnetic_levitation&oldid=1112700831. [Online; accessed 17-October-2022]. 2022.

[27] Wikipedia contributors. *NewtonEuler equations — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Newton%E2%80%93Euler_equations. [Online; accessed 1-November-2022]. 2021.

[28] Wikipedia contributors. *PID controller — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=PID_controller&oldid=1112744285. [Online; accessed 24-October-2022]. 2022.

[29] Wikipedia contributors. *Pulse-width modulation — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Pulse-width_modulation&oldid=1112641448. [Online; accessed 1-November-2022]. 2022.

[30] Wikipedia contributors. *Rotation matrix — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Rotation_matrix&oldid=1106415981. [Online; accessed 18-October-2022]. 2022.

[31] Wikipedia contributors. *Schottky diode — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Schottky_diode&oldid=1085155572. [Online; accessed 1-November-2022]. 2022.

[32] Wikipedia contributors. *Solenoid — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Solenoid&oldid=1107669093. [Online; accessed 1-November-2022]. 2022.

Sources for used 3D assets in Fusion360

As per grabCAD licence; https://grabcad.com/terms the following models has and can be used for NON-commercial purposes when sources is given:

Potentiometer:
https://grabcad.com/library/potenciometro-2
Jesús A. Pérez Rincón

LM2596 DC-DC StepDown Converter (Buck converter):
https://grabcad.com/library/lm2596-dc-dc-stepdown-converter-1
Ivano De Marchi

DS-210 2.1mm DC Female Barrel Jack:
https://grabcad.com/library/ds-210-2-1mm-dc-female-barrel-jack-1
Vasily Kashirin

Breadboard 30 x 70 mm:
https://grabcad.com/library/breadboard-30-x-70-mm-1
Doctor Rozum

Prototyping Breadboard:
https://grabcad.com/library/prototyping-breadboard-1
Gabriel Tóth

Teensy 4.0
https://grabcad.com/library/teensy-4-0-1
Simon Nylund

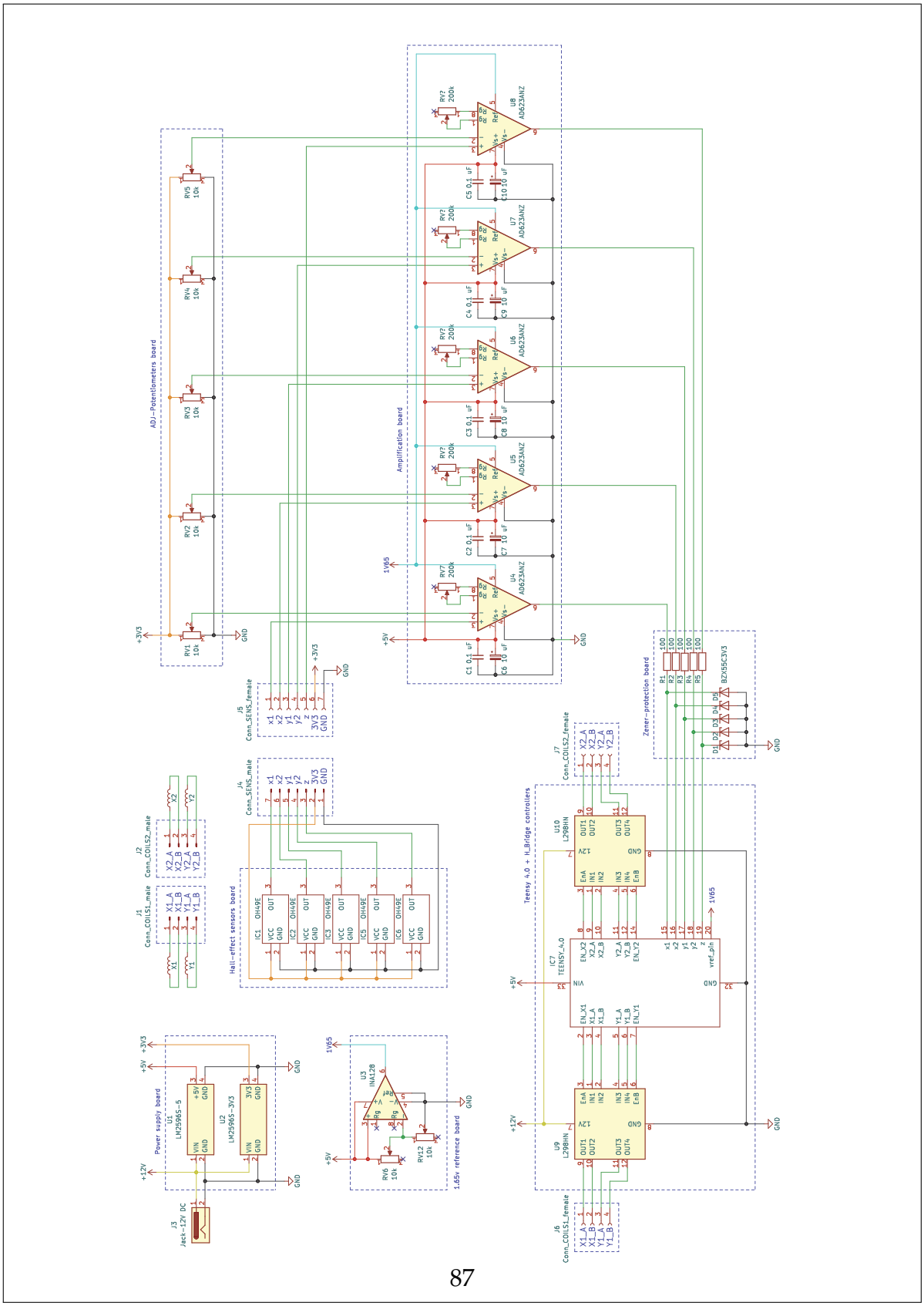
As per snapEDA license; https://www.snapeda.com/about/terms/ the following models has and can be used for NON-commercial purposes:

INA 128(Instrumental amplifier):
https://www.snapeda.com/parts/INA128PA/Texas%20Instruments/view-part/?company=Nuntamp;t=INA128amp;welcome=homeamp;ref=search

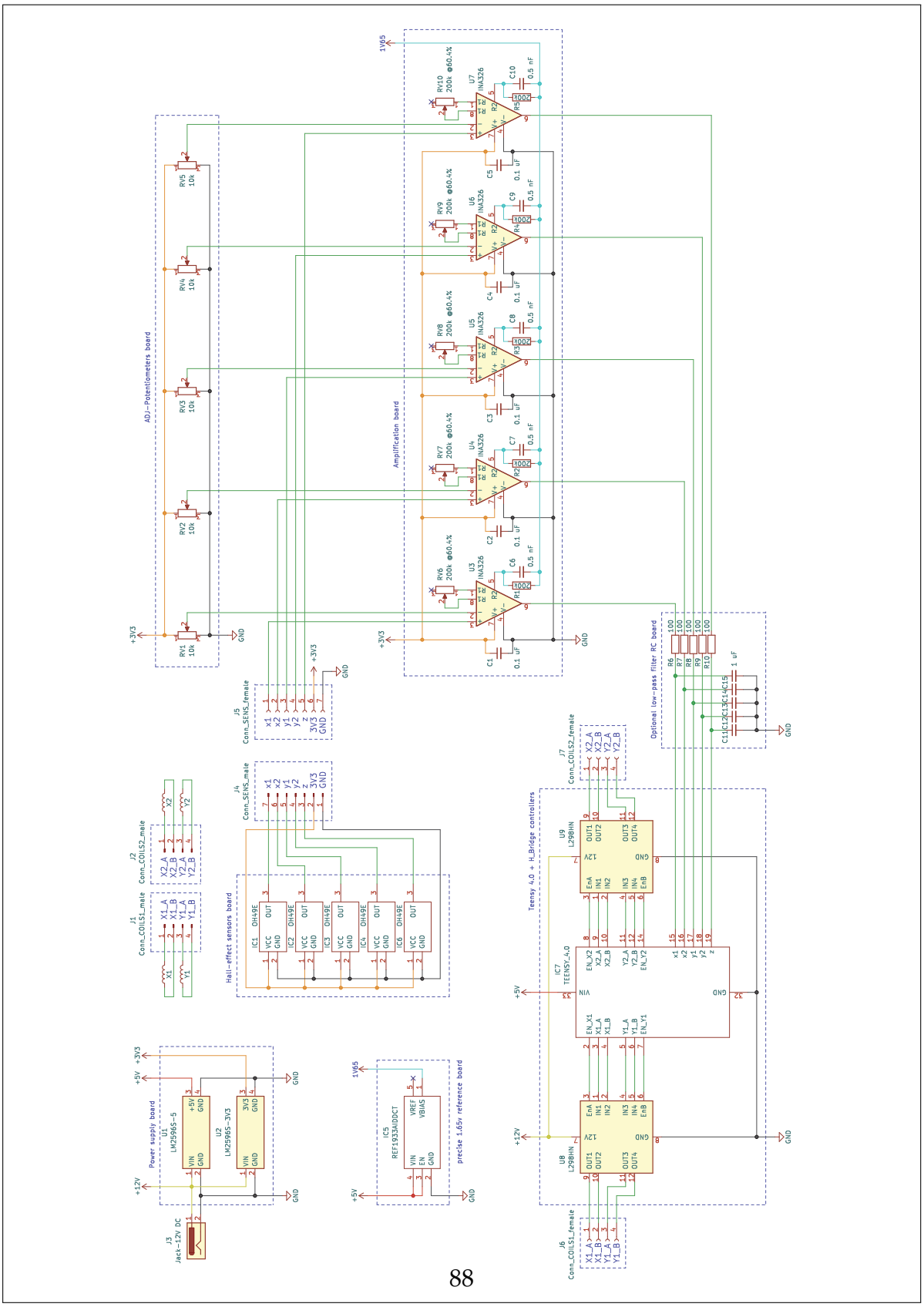
As per autodesk license; https://www.autodesk.com/company/legal-notices-trademarks/website-terms-of-use/terms-of-use-english the following models has and can be used for NON-commercial purposes:

L298N motor driver:
https://gallery.autodesk.com/projects/133940/l298n?searched=
Tecnologiasmagallanes Ies Magallanes

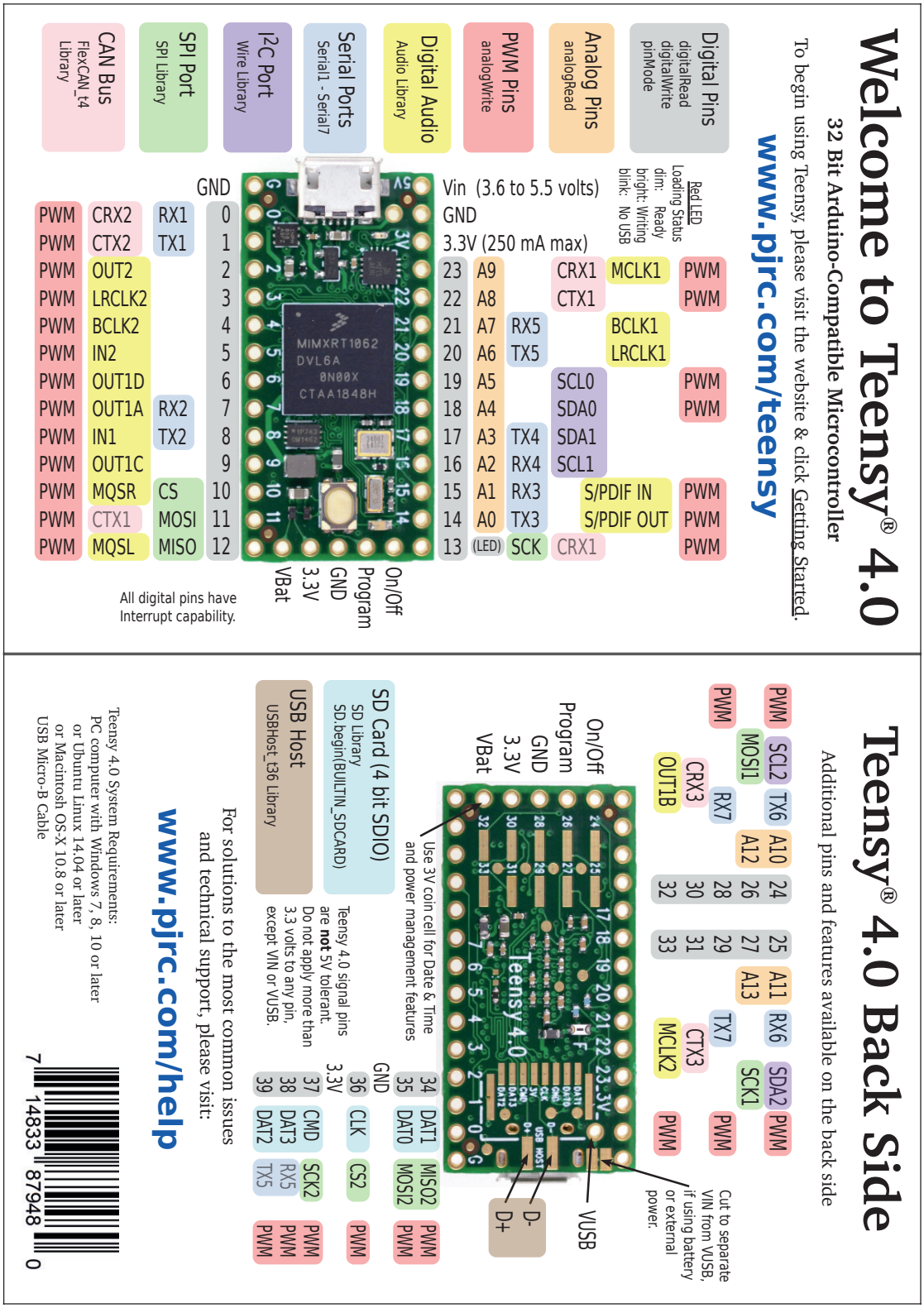
A Electrical schematic



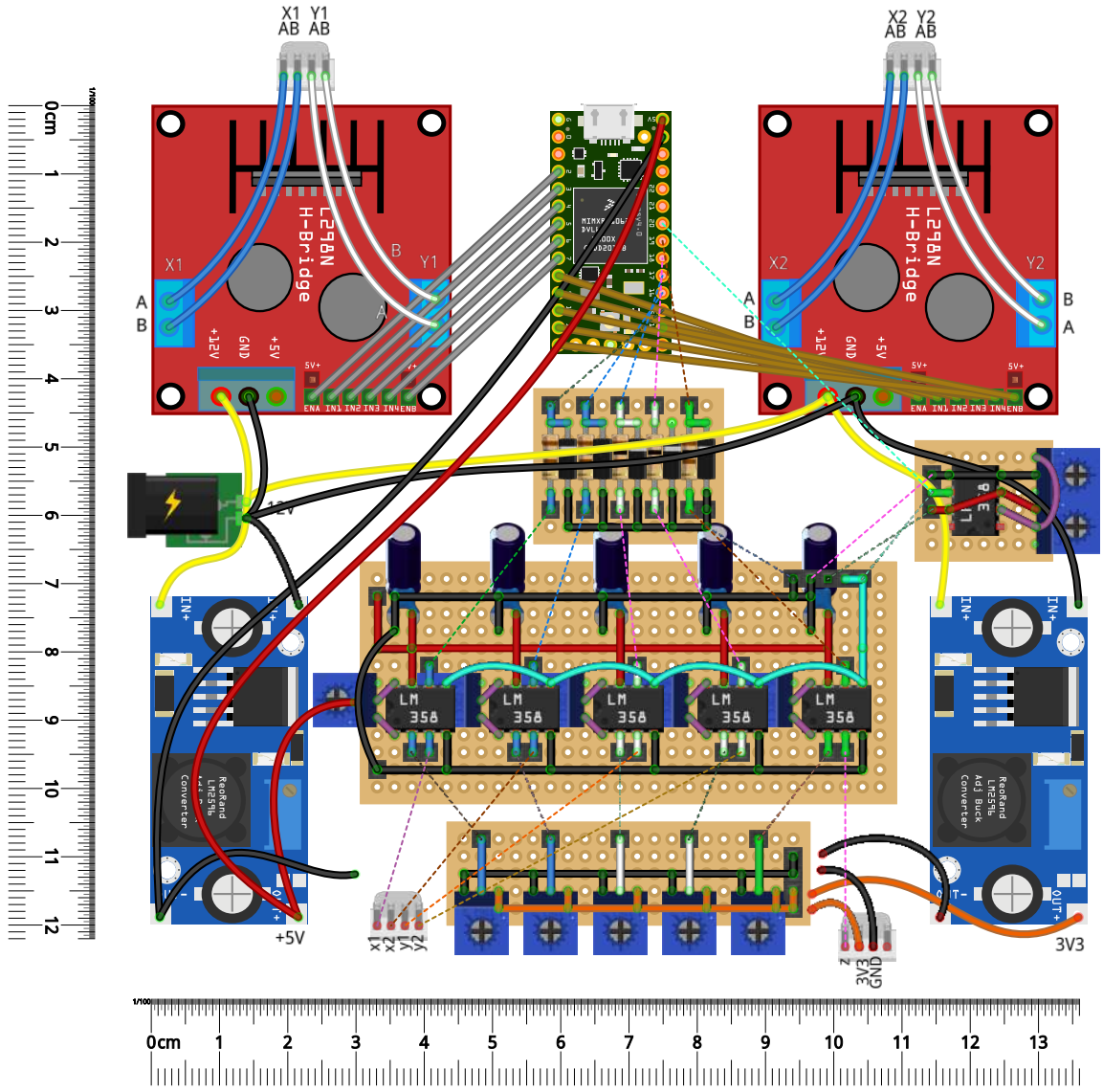
The INA326 schematic is also presented as a suggestion for future pcb design. It also includes REF1933 voltage reference.



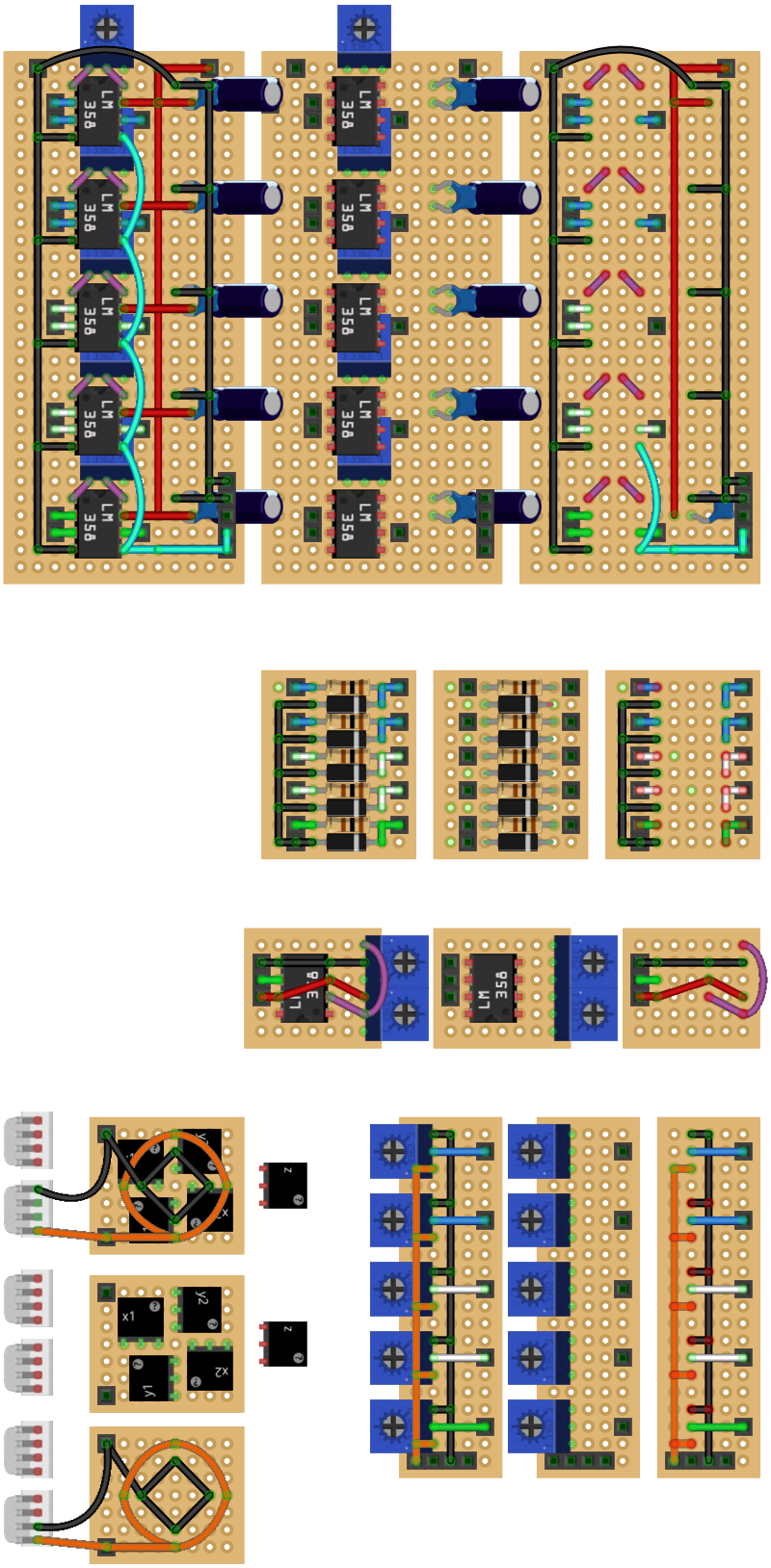
B TEENSY PINOUT



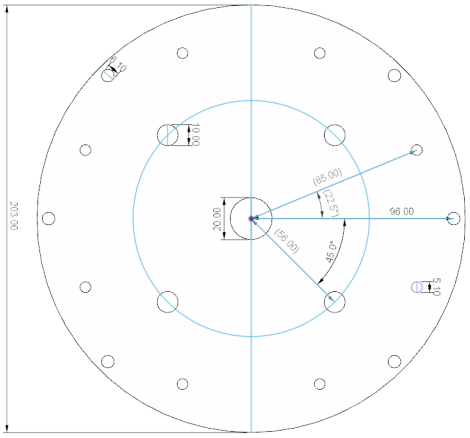
Electrical housing



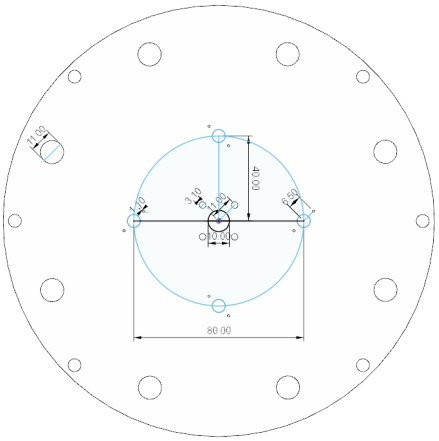
Each board is shown here in three different ways: only wiring, only components and mixed.



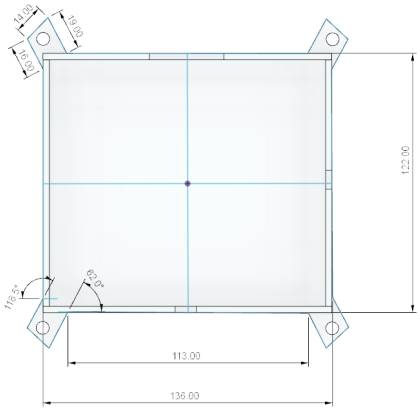
D Plexiglass frame design



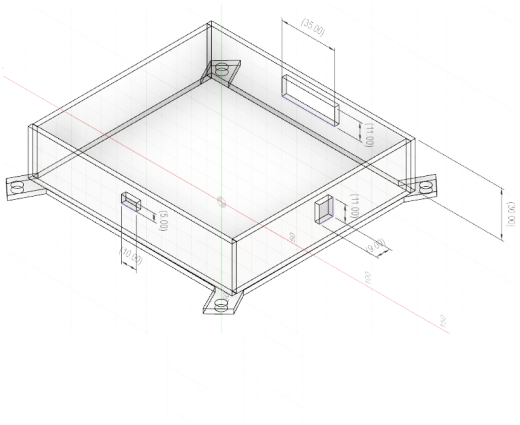
(a) Magnet plate



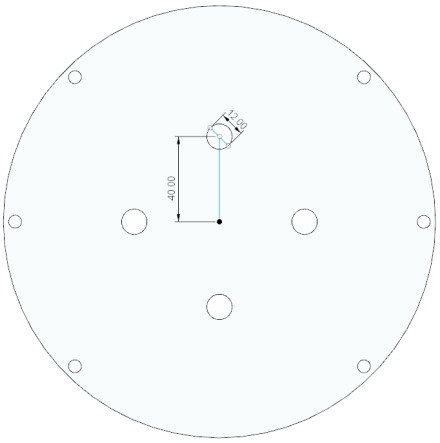
(b) Solenoid plate



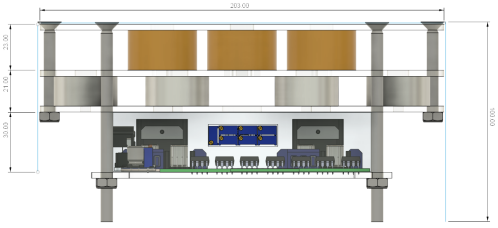
(c) Electrical housing above



(d) Electrical housing side



(e) Top plate



(f) System from side

E PARAMS.M

params	Variable	Description	Value	Unit
solenoids	ri	Inner radius	0.0050	m
	ro	Outer radius	0.0175	m
	h	Height	0.0200	m
	N	Number of solenoids	4	
	R	Radius of solenoid circle	0.040	m
	zs	Offset along z-axis	0.023	m
	nr	Number of rings in radius	20	
	nh	Number of rings in height	50	
	nl	Number of discretizations	100	
magnets	ri	Inner radius	0.0025	m
	ro	Outer radius	0.0145	m
	h	Height	0.0140	m
	N	Number of permanent magnets	8	
	R	Radius of permanent magnet circle	0.085	m
	offset	Placement offset angle	0	degrees
	nr	Number of rings in radius	15	
	nh	Number of rings in height	15	
	nl	Number of discretizations	100	
	meas_height	Measured levitation height[1]	0.0765	m
	I_fast	Equivalent current [fast]	134.1935	A
	I_acc	Equivalent current [accurate]	91.2903	A
levmag	ri	Inner radius	0	m
	ro	Outer radius	0.03	m
	h	Height	0.005	m
	m	Mass	0.072	kg
	nr	Number of rings in radius	30	
	nh	Number of rings in height	5	
	nl	Number of discretizations	100	
	meas_height	Measured levitation height[2]	0.074	m
	I_fast	Equivalent current [fast]	50.4839	A
	I_acc	Equivalent current [accurate]	39.4839	A
sensors	x	x-coordinates	see below	
	y	y-coordinates		
	z	z-coordinates		

x-coordinates	[0.0055,-0.0055,0,0,0]	m
y-coordinates	[0,0,0.0055,-0.0055,0]	m
z-coordinates	[0.0275,0.0275,0.0275,0.0275,0.0265]	m

[1] neodymium magnet vs neodymium magnet simplified system
[2] neodymium magnet vs levmag simplified system