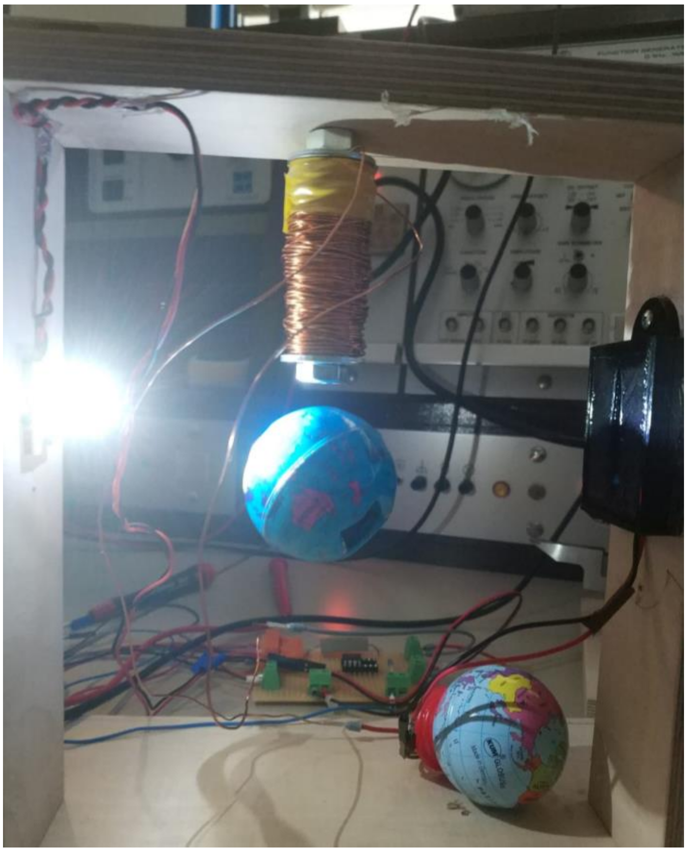
Werkzeugmaschinen und Steuerungen
TU KAISERSLAUTERN

MASTER THESIS

Design and Development of a Digitally Controlled Magnetic Levitation System



Author: Juan Sebastián Lafuente Larrañaga

Enrolment Number: 411962

Supervisor: M.Sc. Jonas Weigand

Datum: 09.10.2019

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst und ohne unerlaubte, fremde Hilfe angefertigt habe. Alle Ausführungen, die anderen Schriften wörtlich oder sinngemäß übernommen wurden, sind kenntlich gemacht. Alle verwendeten Quellen sind im Literaturverzeichnis zitiert.

Kaiserslautern, den 09. Oct. 2019

____

Juan Sebastián Lafuente Larrañaga

Sperrvermerk

Falls die Arbeit der Geheimhaltung unterliegt, wie es bei externen Arbeiten üblich ist, wird an dieser Stelle ein Sperrvermerk eingefügt. Andernfalls ist diese Seite zu entfernen.

Table of Contents

Introduction

In this project the study and the design of a magnetic levitation control system has been developed. The magnetic system that is studied in this project consist on an electromagnet which has been coiled with cupper wire making a winding and a levitating object with spherical (ball) shape and made of a ferromagnetic material.

To be able to make the ball levitate, a control system is required. A sensor is needed to measure the location of the levitated object. A solar cell is used as a sensor, which works together with a LED light emitter. The location of the ball will be related with the output voltage of the sensor.

The sensor provides a microcontroller the information of the measurements. The microcontroller, based on a Proportional Derivative (PD) control strategy, calculates the current or the voltage that has to be applied to the electromagnet in order to maintain the ball levitating. Arduino is used as a microcontroller.

Simulation tools based on the model of the system are used in order to study and get information about the behaviour of the system. MATLAB Simulink has been used as a simulation software.

Abstract

In this document, the development of the magnetic levitation system is explained and how it has been done. The goals of the project, can be summarized in two main objectives.

- Development of a physical prototype where a controlled magnetic levitation system is represented.
- Study and document the magnetic levitation system that has been created based on a previous simulation. This involves the study of the magnetic systems in order to create a model of the system and simulate its behaviour.

This document has been divided in different chapters based on the phases that have been followed to develop the system. The phases of the project are the followings.

- Study and documentation of the magnetic levitation systems from a theoretical point of view.
- Development of a model that represents the system.
- Simulation of the model.
- Designing, building and assembly of the prototype.
- Tuning of the parameters of the controller.

Symbol Index

Abbreviation	Variable	Unit
L_C	Coil Inductance	[H]
R_C	Coil Resistance	[Ω]
U_C	Applied Voltage to the Coil	[VDC]
$i_C(t)$	Instant Current in the Coil	[A]
m	Mass of the Levitated Object	[kg]
$\ddot{y}$	Acceleration of the Levitated Object	[m/s^2]
$\dot{y}$	Velocity of the Levitated Object	[m/s]
y	Location of the Levitated Object	[m]
F_{Mag}	Magnetic Force	[N]
F_G	Gravitational Force	[N]
g	Gravitational Constant	[m/s^2]
L(y)	Variable Inductance	[H]
a	Geometrical Constant	[m]
W(i,y)	Magnetic Energy in the Electromagnet	[W]
I	Current in the Electromagnet	[A]
K_C	Magnetic Constant	[$N*m^2/A^2$]
N	Number of Turns	[-]
A	Gap Area	[m^2]
μ_0	Magnetic Permeability	[N/A^2]
I_0	Reference Current	[A]
y_0	Reference Location	[m]

ΔI	Current Increment	[A]
Δy	Location Increment	[m]
R	Resistor in Series with the Electromagnet	[Ω]
R_{Total}	Resistance of the Channel 1	[Ω]
V_{in}	Solar Cell Input Voltage	[VDC]
V_{out}	Solar Cell Amplified Output Voltage	[VDC]
R_f	Resistor of the Amplifier 1	[Ω]
R_g	Resistor of the Amplifier 2	[Ω]
$U_{CH.1}$	Voltage in the Channel 1	[VDC]
$U_{CH.2}$	Voltage in the Channel 2	[VDC]
t_{Filter}	Time Constant of The Filter	[s]
K_P	Proportional Gain	[VDC/m]
T_D	Derivative Time	[-]
$u_{Control}$	Control Signal Voltage	[-]
$i_{Control}$	Control Signal Current	[-]
i_{PWM}	Control Signal PWM Current	[-]
$r(s)$	Reference	[m]
$e(s)$	Error	[m]
$u(s)$	Control Signal Voltage (s domain)	[VDC]
$i(s)$	Control Signal Current (s domain)	[A]
$d(s)$	Disturbance	[m]
$G_{CL}(s)$	Close Loop Transfer Function	[-]
$G_{PD}(s)$	PD Controller Transfer Function	[VDC/m]
$G_{Elec}(s)$	Electric Block Transfer Function	[A/VDC]

$G_{Mag}(s)$	Magnetic Force Transfer Function	[m/A]
$G_{Sensor}(s)$	Sensor Filter Transfer Function	[-]
FM	Filtered Measurement	[VDC]
FM_{Old}	Previous Filtered Measurement	[VDC]
NFM	Non-Filtered Measurement	[VDC]
dt	Sample Time	[s]
τ_{fs}	Time Constant of the Sensor Filter	[s]
τ_{fd}	Time Constant of the Derivative Filter	[s]
T_{fo}	Time Constant of the Output Filter	[s]
FO	Filtered Output	[A]
FO_{Old}	Previous Filtered Output	[A]
NFO	Non-Filtered Output	[A]

List of Abbreviations

WSKL	Lehrstuhl für Werkzeugmaschinen und Steuerungen Kaiserslautern
MagLev	Magnetic Levitation
PWM	Pulse Width Modulation
PD	Proportional Derivative

1 Contents

1.1 Magnetic Levitation Context

Magnetism has always been a phenomenon which has aroused interest in humans. According the historical sources [1], the magnetic phenomenon was documented for the first time in the ancient Greek by Thales of Miletus (about 585B.C.). Since that moment, many scientists have observed and written about the magnetic attraction and repulsion. Many applications were developed, like the magnetic compass, which improved the navigation routes possibilities. It was not until the XIX century when the Danish physicist and chemist Hans Christian Ørsted [2] discovered that electric currents were able to create disturbances in magnets. That was the first time that a connection between the electricity and the magnetic fields was discover. Since that moment, many other scientists in the following decades were going to developed the fundaments of the electromagnetism.

The technical applications of the magnetic levitation are wide, from magnetic levitating trains to magnetic bearings. The main advantage, is that is possible to suppress the friction when an object is moving on a surface. The friction is the main responsible of the loss of the energy when there is a relative movement between two objects in contact. With this technology, highest ranges of efficiency can be achieved.

Magnetic fields, are an example of a nonlinear and instable processes when they are used to counterbalance the weight of a levitated object. An electrical current, creates a magnetic field around it [3], using a coil the magnetic field can be oriented in a desired direction as it can be seen in Figure 1.

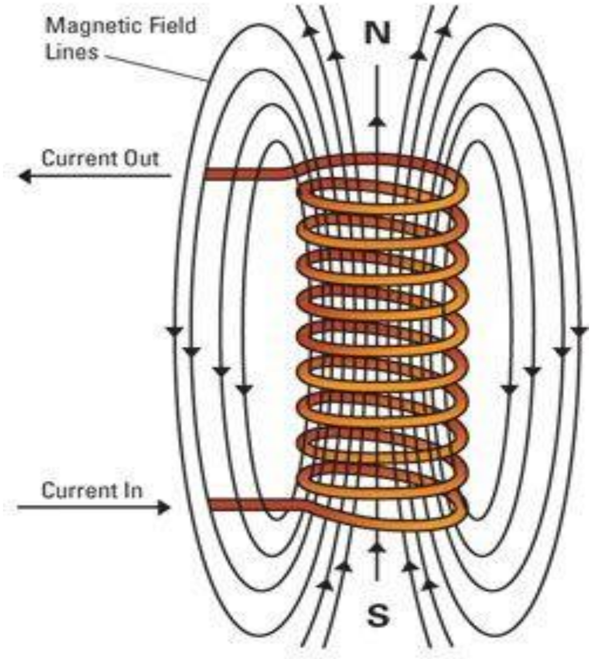


Figure 1 Magnetic Field Lines in an Electromagnet (From [4])

In Figure 1 it is appreciated, how the magnetic field lines are distributed when a coil is excited with a current. The density of those magnetic field lines that are crossing a determined area, will determine which is the magnetic force that would appear in a material (located in the mentioned area) with a determined ferromagnetic characteristic.

Can be observed in Figure 1, that inside the coil the magnetic field lines are approximately constant among the section, so the magnetic force which a ferromagnetic object would experiment could be considered as a linear process. Nevertheless, once the magnetic field lines leave the coil, the linearity of the process starts decreasing. In this project, the levitating object is going to be located under the electromagnet, as it can be seen in the Figure 2, so the process that describes the dynamics of the system is not linear. The process can be linearized around an operation point, creating a range of values where the process approaches to a linear system. In the following chapters the linearization of the system and how the levitation is managed are going to be described.

Figure 2 Example of Magnetic Levitation Using an Electromagnet (From [5])

In order to find a range of stability for magnetic fields, many control strategies have been designed during the past decades. Those control systems were using the new technologies available, mainly they are not only based on PID control systems, but also on advanced strategies like the predictive control [6].

The research and development of the magnetic fields represents a challenge for scientific community in the following years. The reduction of the friction between the surfaces represents a step forward in the transportations systems, making possible a more efficient way of moving. One example of it, is the magnetic levitating train, which has been the object of numerous investments in the last years [7].

1.2 Fundaments of the Magnetic Levitation

The magnetic levitation system of this project, consists on a ferromagnetic object which is levitating due to the attraction force that an electromagnet creates. The system is instable and requires an active control system to work properly. The electromagnet creates the magnetic field thanks to the current that is flowing in the coil. It can be seen therefore, that there are two subsystems which are related one with each other. First the electrical subsystem and second the magnetic subsystem. In this chapter, the relation between the current that flows in the electromagnet and the magnetic field or force that is inducted is going to be analyse from a theoretical point of view.

1.2.1 Electrical subsystem

The electromagnet produces the magnetic field thanks to the current that flows on it. Electromagnets consist of wire wound into a coil. A coil is represented as an inductance L_C [H] with a resistance R_C [Ω], as it can be seen in the Figure 3.

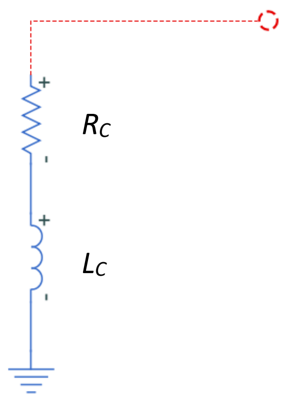


Figure 3 Electrical representation of the electromagnet (Tool Simulink)

When a Direct Current (DC) voltage U_C [V] is applied to the system, a current $i_C(t)$ [A] will flow. The relation between the voltage U_C and the current $i_C(t)$ is given in the Equation (1) and is the result of applying Ohm's Law.

$$U_C = i_C(t)R_C + L_C\frac{di_C(t)}{dt} \quad (1)$$

If the Laplace transform is considered, the equation becomes Equation (2).

$$U_C(s) = R_C I_C(s) + L_C s I_C(s) \tag{2}$$

So, the transfer function that relates the applied voltage input (U_C) and the output current (I_C) in the coil or electromagnet would be Equation (3):

$$G_{Elec}(s) = \frac{I_C(s)}{U_C(s)} = \frac{1}{R_C + L_C s} = \frac{1/R_C}{1 + L_C s/R_C} \tag{3}$$

In Equation (3) can be seen, that the subsystem is a first order system [8], where the resistance and the inductance of the coil will impose the time constant and the gain of the system. Those parameters will determine the stability of the system. L_C and R_C are always positive so the subsystem will be stable. The resistance and the inductance of the coil can be measured with a multimeter and with an inductance test [9] respectively. It has been seen experimentally, that once a voltage is applied to the coil, it reaches the nominal current in a short time which has been considered valid for the project. Therefore, a control system would not be necessary for this subsystem.

1.2.2 Magnetic subsystem

The electromagnet creates the magnetic field that induces the magnetic force (F_{Mag}[N]). The force balance is applied to the system of the Figure 4, resulting in the Equation (4).

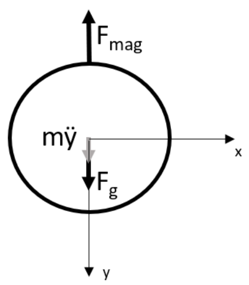


Figure 4 Force Balance (Own made)

$$m\ddot{y} = F_{Mag} - F_G \tag{4}$$

Where m [kg] is the mass of the levitated object, $\ddot{y}$ [m/s^2] is the acceleration of the levitated object, F_{Mag} [N] is the magnetic force and F_G [N] is the gravitational force. Considering that the levitating object remains static Equation (5):

$$F_{Mag} = F_G \tag{5}$$

The gravitational force can be obtained applying Newton's second law:

$$F_G = mg \tag{6}$$

Where *g* is the gravitational constant *g* = 9.81 [m/s^2].

The levitated object is a ferromagnetic material. For this reason, the parameters that define its magnetic behaviour are three. The mass of the object *m* [kg], the distance from the object to the electromagnet *y*[m], and the contribution to the magnetic field that the coil or the electromagnet creates. This last parameter is represented as a variable inductance *L(y)* [H], that changes with the position of the object [10]. The value of the inductance is maximum when the levitating object is next to the magnet *y(0)*, and minimum when all the contribution to the magnetic field is made by the electromagnet *y(∞)*.

To obtain *L(y)*, there are many experimental models in the bibliography. For this project the following one has been selected [11] Equation (7).

$$L(y) = L(\infty) + \frac{L(0)}{1 + \frac{y}{a}} \tag{7}$$

Where *a* [m] is a geometrical constant that should be experimentally determined.

To obtain the magnetic force that appears in the ferromagnetic object, the magnetic energy in the electromagnet *W(i,y)* can be obtained [12]. Where *I* [A] is the current flowing in the electromagnet.

$$W(I,y) = \frac{1}{2} L(y) I^2 \tag{8}$$

The magnetic force that an electromagnet induces to a ferromagnetic sphere can be obtained according [10]. The magnetic force can be obtained as the gradient of the Equation (8).

$$F_{Mag} = \frac{\partial W(I,y)}{\partial y} = -\frac{L(0)aI^2}{2(y+a)^2} = -k_c \frac{I^2}{(y+a)^2} \tag{9}$$

Where *kc* [N*m^2/A^2] is the magnetic constant, *I* [A] is the current that flows in the electromagnet and *y* [m] is the distance of the ferromagnetic levitated object to the electromagnet. The position constant *a*, is related with the validity range. In this project, this parameter can be approximated to zero because as it will be seen in the following chapter 1.3 about the physical model, the working range is not wide. On top of that, calculating this parameter requires a big testing effort, which is not the goal of this project and does not add any relevant value to the quality of the model. After the simplification the magnetic force can be defined as Equation (10).

$$F_{Mag} = k_c \frac{I^2}{y^2} \tag{10}$$

The magnetic constant can be obtained using the Maxwell formula shown in the Equation (11) [13] [14]. This formula can be applied when the electromagnet consist on a cylindrical coil and the object that is going to be under the magnetic field is a ferromagnetic material [15].

$$k_c = \frac{\mu_0 A\, N^2}{4} \tag{11}$$

A [m^2] is the air gap area, N [-] represents the number of turns that the wire has and μ_0 [N/A^2] is the magnetic permeability value which is $4\pi * 10^{-7}$[N/A^2].

As it can be seen, the magnetic force is a nonlinear equation because it has two multiplied variables, the current in the coil *I* [A]and the distance between the magnet and the levitating object, which will be defined as *y* [m].

$$F_{mag} = f(I, y) \tag{12}$$

Now, the Equation (4) can be written again considering Equations (6) and (9) as:

$$m\ddot{y} = k_c \frac{I^2}{y^2} - mg \tag{13}$$

The Equation (13), can be linearized with the Taylor method [16]. To use this method, first of all an equilibrium point has to be defined. In the equilibrium point the levitated object is not moving, that means that there is no acceleration so the left side of the Equation (13) is equal to zero. The reference current that flows in the electromagnet I_0 and the reference location of the levitating object y_0 are defining the equilibrium point.

Equation (14) represents the Taylor Series linearization [16] of a nonlinear function with two variables.

$$f(\Delta I, \Delta y) = f(I_0, y_0) + \left(\left. \frac{\partial f(I,y)}{\partial I} \right|_{\substack{I=I_0 \\ y=y_0}} \right) \Delta I + \left(\left. \frac{\partial f(I,y)}{\partial y} \right|_{\substack{I=I_0 \\ y=y_0}} \right) \Delta y \tag{14}$$

Where:

$$\begin{cases} \Delta I = I - I_0 \\ \Delta y = y - y_0 \end{cases} \tag{15}$$

Which can be graphically represented as it can be seen in the following Figure 5. It can be seen that the approach is valid when the values are close to the operation point, which will be this case.

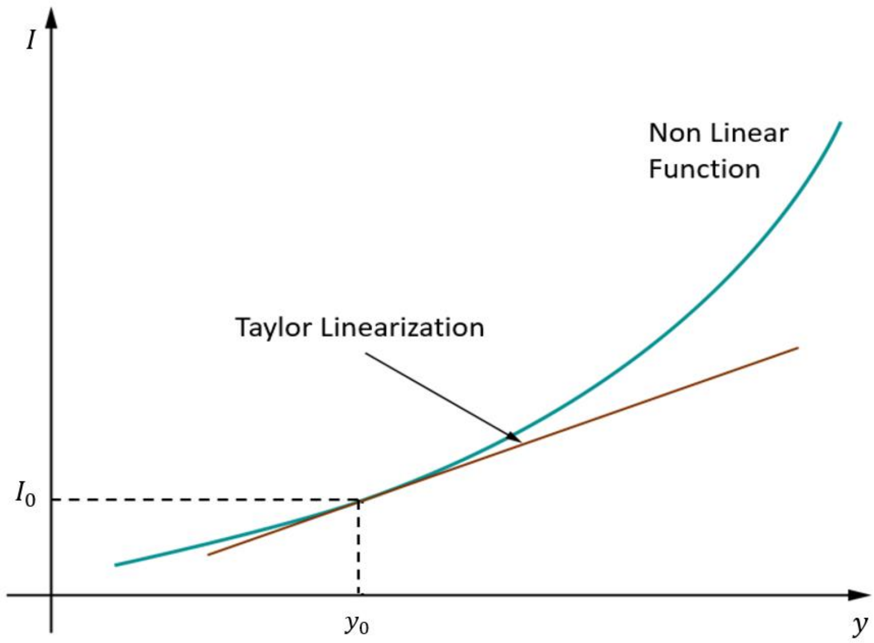


Figure 5 Graphic representation of Taylor Series Linearization (Own made)

Obtaining the partial derivates:

$$\begin{aligned} \left. \frac{\partial f(I,y)}{\partial I} \right|_{\substack{I_0 \\ y_0}} &= -k_c \frac{2I_0}{{y_0}^2} \\ \left. \frac{\partial f(I,y)}{\partial y} \right|_{\substack{I_0 \\ y_0}} &= k_c \frac{2{I_0}^2}{{y_0}^3} \end{aligned} \tag{16}$$

This way, the magnetic force can be linearized and the equilibrium Equation (13) can be rewritten as Equation (17):

$$m\ddot{y} = f(\Delta I, \Delta y) - mg$$
$$m\ddot{y} = f(I_0, y_0) - k_c \frac{2I_0}{{y_0}^2}\Delta I + \ k_c \frac{2{I_0}^2}{{y_0}^3}\Delta y - mg \tag{17}$$

In the equilibrium point the magnetic force $f(I_0,y_0)$ is equal to the gravitational force *mg*. So, the Equation (17) becomes:

$$m\ddot{y} = -k_c \frac{2I_0}{{y_0}^2}\Delta I + \ k_c \frac{2{I_0}^2}{{y_0}^3}\Delta y \tag{18}$$

Taking the Laplace transform,

$$ms^2Y(s) = -k_c \frac{2I_0}{{y_0}^2} I(s) + k_c \frac{2{I_0}^2}{{y_0}^3} Y(s) \tag{19}$$

The transfer function of the magnetic force system (Equation(20)), relates the location of the levitated object (output) and the current that flows in the electromagnet (input).

$$G_{Mag}(s) = \frac{I(s)}{Y(s)} = \frac{-k_c \frac{2I_0}{{y_0}^2}}{ms^2 - \left[k_c \frac{2{I_0}^2}{{y_0}^3}\right]} = \frac{-k_c \frac{2I_0}{m{y_0}^2}}{s^2 - \left[k_c \frac{2{I_0}^2}{m{y_0}^3}\right]} \tag{20}$$

Equation (20) represents the dynamics of the electromagnet. It can be seen that it has two poles and that one of the is located in the positive half of the complex plane. The values of the poles are:

$$p_{1,2} = \ \pm \sqrt{k_c \frac{2{I_0}^2}{m{y_0}^3}} \tag{21}$$

That means that the open-loop system is unstable. In the control system justification of the chapter 1.5, the control strategy is going to be described. The values I_0 and y_0 have been obtained experimentally placing the ball in a desired location and measuring the current that is required to make the object levitate. The values of I_0 and y_0 are shown in the Equation (22).

$$\begin{cases} I_0 = 0.4\ [A] \\ y_0 = \ 0.002\ [m] \end{cases} \tag{22}$$

1.3 Physical Model

This project is an example where the magnetic levitation effect is represented. In the following chapter, all the components that are used in the prototype are going to be described and the reason of every choice is going to be explained. The election of the hardware is very important because every component has its own physical characteristics and those affect directly to the behaviour of the system, to the model and to the control system design.

The physical system of the hardware can be divided in five blocks which are related to each other. Those blocks are resumed in the following Figure 6. Every block has its own dynamic and characteristics, which are going to be described.

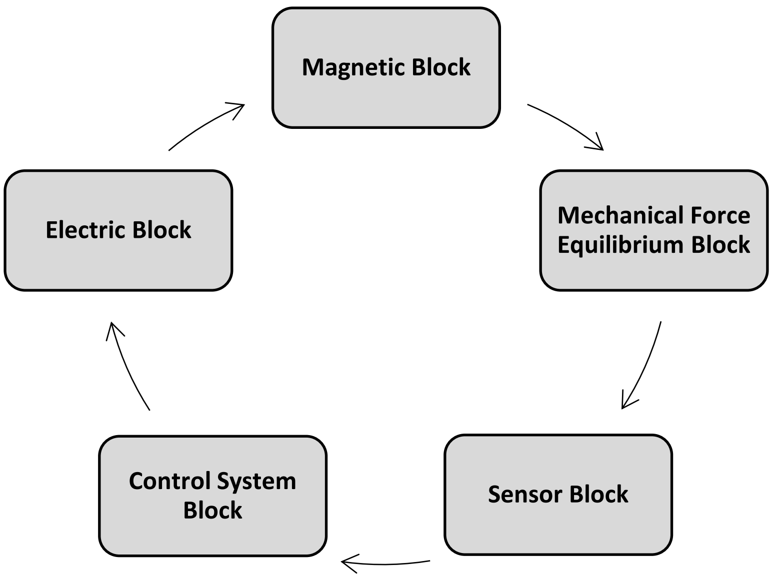


Figure 6 Blocks of the Hardware (Own made)

1.3.1 Electromagnet

The electromagnet is the component in charge of making the object levitate, therefore its characteristics will determine the levitating conditions. To build the electromagnet the following recommendations has been followed [17].

A cupper wire is coiled around a ferromagnet material making a winding. As the physical characteristics of the magnet are related with its geometry and number of turns (Equation (11)), is important to set parameters of the magnet and build it in order to have a magnet with known characteristics.

First of all, in this project has been considered a maximum levitation distance requirement for the electromagnet. That means, setting the maximum distance where the electromagnet is able to defeat the gravitational force and make the sphere levitate. In this limit case, the current flowing in the electromagnet would be the maximum possible. It has been decided a maximum distance of 8mm.

Knowing that while the object is levitating around an equilibrium point, the magnetic force is equal to the gravitational force (from Equation (13)):

$$F_G = \frac{4\pi * 10^{-7} A\, N^2 I^2}{4\, y_{max}^2} \tag{23}$$

It has been experimentally obtained the weight of the levitated sphere, which is 0.06 [N]. An electromagnet was designed and built according the Equation (23), which was supposed to fulfil the 8 mm maximum distance requirement that was commented in the previous paragraph. The parameters that this first electromagnet had can be seen in the Equation (24). The current of the test was limited by linear power supply source.

$$\begin{cases} N = 400\ [-] \\ I = 0.6\ [A] \\ A = 100 * 10^{-6} [m^2] \end{cases} \tag{24}$$

But experimentally was observed, that the electromagnet was not following the dynamics of the Equation (23). The maximum distance where the electromagnet was able to defeat the gravitational force was 2 mm, which means an error of 75% comparing with the value of 8 mm that theoretically should be. Because of this reason, a bigger and more powerful electromagnet was designed. After the new electromagnet was built, it was validated with a test, where the 8 mm maximum lift requirement was fulfilled. The geometry of the coil that has been chosen can be seen in the Figure 7. Some parameters of the Equation (23), like the weight of the sphere (F_G), the maximum distance lift (y_{Max}) , and the gap area (A) , shown in the Equation (25).

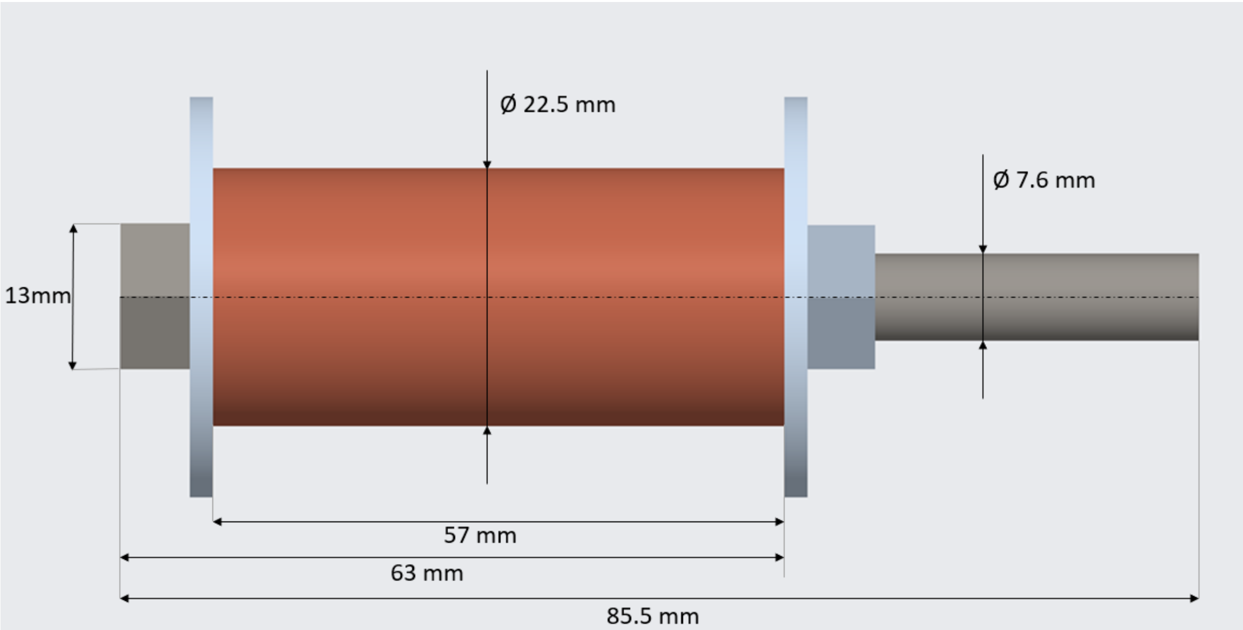


Figure 7 Geometry of the Electromagnet (Tool Creo Parametric)

$$\begin{cases} F_G = 0.06\ [N] \\ y_{max} = 0.008\ [m] \\ A = 4.536 * 10^{-3}\ [m^2] \end{cases} \tag{25}$$

Taking those parameters (Equation (25)) to the Equation (23), it can be seen that the current and the number of turns are still variables. To set the number of turns, a 28 meters length coil [18] has been used. The number of turns is determined by the amount of turns that can be done with the mentioned coil. The choice has been made due to the availability of the coil and the simplicity of the winding.

In the Figure 8 and Figure 9 can be seen how the magnet was built using the revolution movement given by a driller machine. To measure the number of turns, a hall effect sensor [19] was implemented to the driller in a fixed position, while a neodymium magnet was located in the rotating part. Every time that the magnet was crossing the hall effect sensor, a digital output signal was created by the sensor, which was processed by Arduino. This way, every revolution was counted.

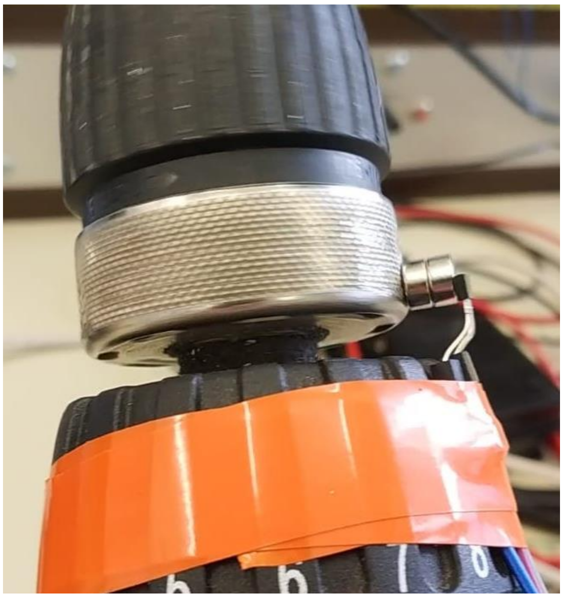


Figure 8 Driller with fixed hall sensor and rotating neodymium magnet (Own made)

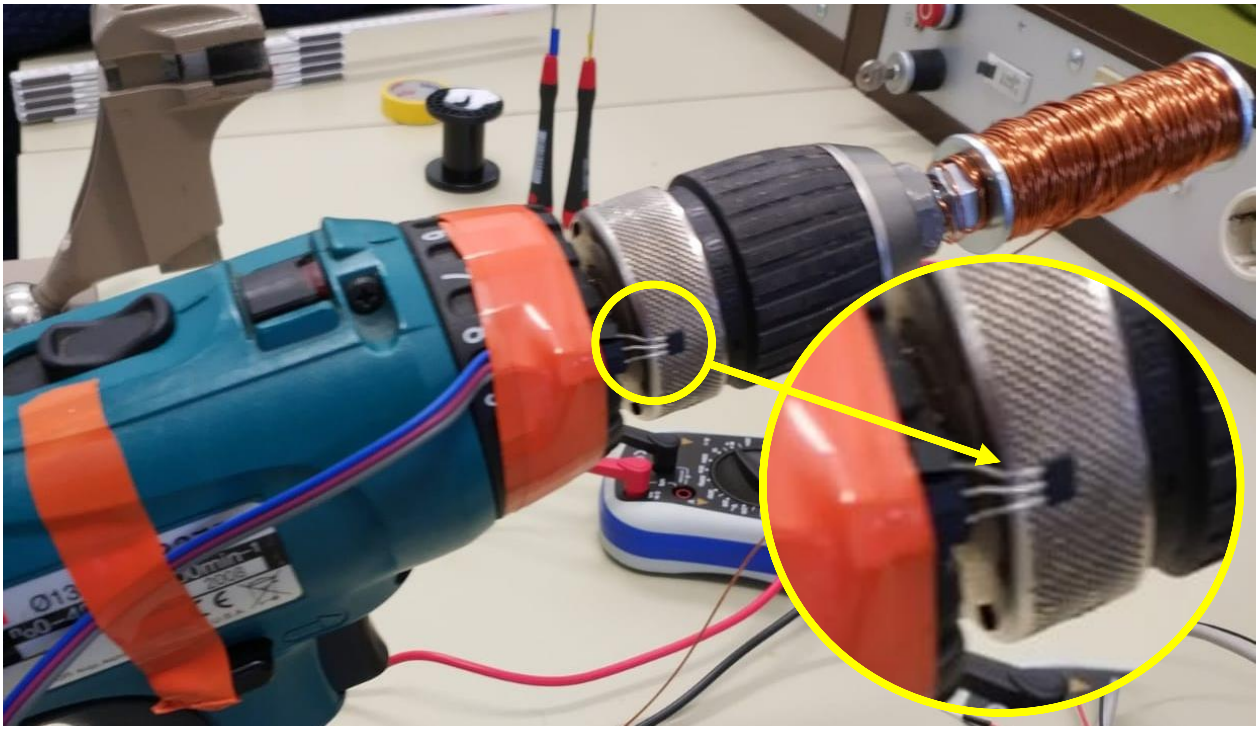


Figure 9 Winding of the electromagnet (Own made)

The code for this revolution counting has been developed using Arduino [20], and can be seen in the Appendix as Revolution Counter Code. The result was a number of turns of:

$$N = 726\ [-] \tag{26}$$

After having winded 28 meters of wire, the electrical resistance R_{Coil} [Ω] was measured with the multimeter. The result was the following Equation (27):

$$R_C = 2.2\ [\Omega] \tag{27}$$

In order to have safe working conditions while a relatively strong magnet, 12VDC voltage has been chosen as an input power source. To handle this voltage, Arduino Motor Shield Rev3 [21] has been implemented on Arduino, which allows to work with higher voltages and currents as it can be seen in the datasheet. This voltage will be varied during the levitation process by the control system (Arduino). The maximum current that Arduino Motor Shield Rev3 can work with is 2 Amperes [21]. For this reason, a resistor should be added on series to the electromagnet not to exceed the maximum current. The Equation (28) shows the value of the resistor [22]:

$$R = 4.7\ [\Omega] \tag{28}$$

As the resistors are in series, both of them are summed to obtain the resistance of the circuit of the electromagnet.

$$R_{Total} = 2.2\ [\Omega] + 4.7\ [\Omega] = 6.9\ [\Omega] \tag{29}$$

When the Ohm's Law is applied to the circuit, the following Equation (30) is obtained.

$$I_{Max} = \frac{U_{Max}}{R} = \frac{12\ [VDC]}{2.2\ [\Omega] + 4.7\ [\Omega\ ]} = 1.74\ [A] \tag{30}$$

In the Equation (30) can be seen that the maximum current is not going to exceed the capacity of Arduino.

For the core of the winding a standard screw has been used for availability and practical reasons. Apart from that, using a ferromagnetic material as a core is an advantage in the sense that the magnetic field that the electromagnet creates can be increased [23].

In the following Figure 10, the finished and built electromagnet can be seen.

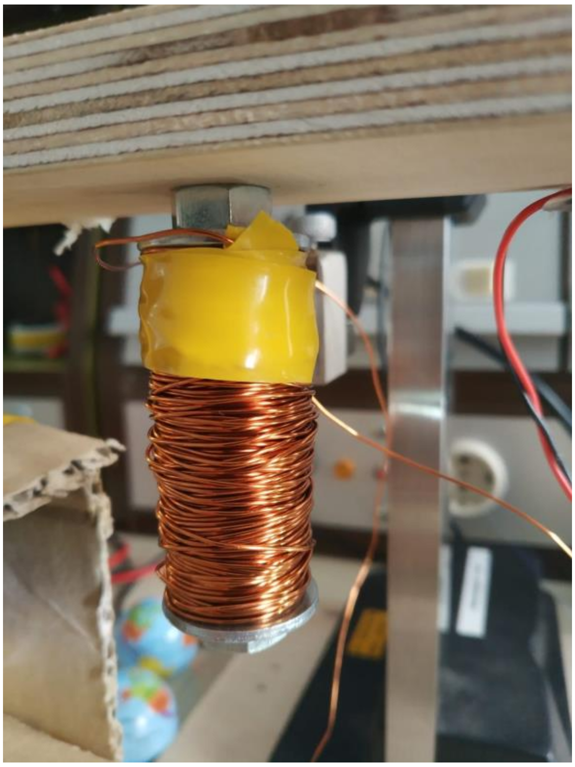


Figure 10 Electromagnet (Own made)

1.3.2 Sensors

The levitation system cannot be achieved without the use of sensors. Thanks to the measurements that the sensor makes during the levitation, the levitating parameters can be controlled. The control system will base and generate the control signal according to the information that the sensors are providing (1.5). Because of this, the controlled variable of this project has to be measured with sensors. In the case of this project, the main variable that has to be controlled is the distance between the electromagnet and the ferromagnetic levitating material. For security and fine-tuning reasons, a current sensor can be also used. Arduino Motor Shield rev3 includes current sensors for its outputs, so the current sensor can be implemented without the need of any extra hardware.

The sensor that is used in the project is the following:

- Light sensor

It consist on a solar cell [24], which has been filtered phisically to avoid the horizontal disturbances. The sensor is located in a fixed position in the frame of the prototype, and works in convination with a LED light emiter located also in a fixed position in the frame in front of the sensor. The light sensor with its filter and the LED emiter can be seen in Figure 11.

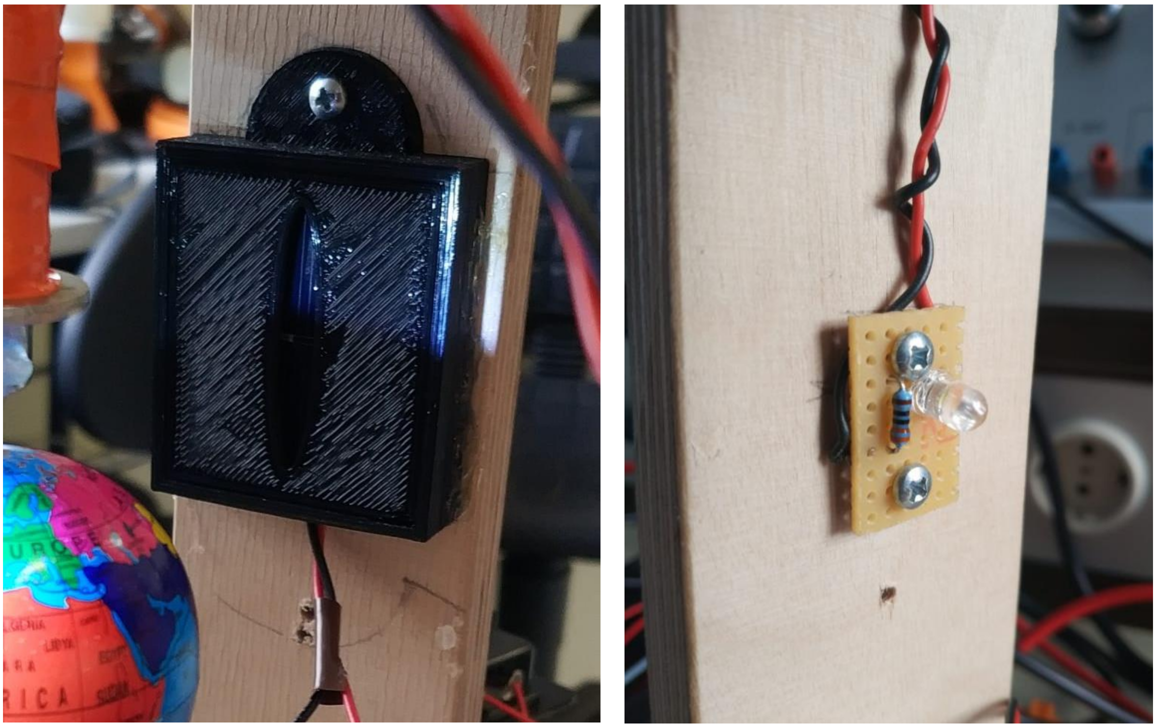


Figure 11 Solar Cell (Velleman sol3n) and LED emitter (Own made)

The light from the LED emitter, fall upon the levitating object, consecuently a shade that depends on the levitating objects location appears on the surface of the solar cell. The sade covers a bigger surface when the distance bewteen the levitating object and the electromagnet is decreases and the other way round. The output from the light sensor consist on an electrical potential difference that depends on the amount of the direct light radiation that the sensor is recieving from the LED emitter. The output voltage from the solar cell sensor is a value between 0 and 1 [VDC]. This way, a relation between the location of the ball and the light sensors output voltage can be found.

If the output voltage from the solar cell sensor is measured for different locations of the levitating sphere, can be seen that for the selected range of movement of the sphere, the voltage variation is 0.245 [VDC] (from the spehre in the $y(0)$ [m] position to $y(\infty)$ [m] position). These measurements have to be processed by Arduino's analogic input. The resolution of Arduino's analogic input is 1024 units for 5 Volts. That means, that the range of movement of the sphere

(0.245 [VDC]) is using 50 steps out of 1023 (Equation (31)). As a result, the measurements are not consider accurate enough for the project. This fact justifies the need of an operational amplifyer.

$$Arduino\ Steps = \frac{Voltage\ Rage}{Resolution} = \frac{0.245\ [VDC]}{5\ [VDC]\ /1024\ [Steps]} \cong 50\ [Steps] \quad (31)$$

Operational amplifiers are linear devices that have all the properties required for nearly ideal DC amplification [25]. The output voltage signal from the solar cell sensor can be amplifyed with an operational amplifier. In this project the OP 27 Operational Amplifier [26] has been used. The electronic scheme that represents the operational amplifyer can be seen in the following Figure 12.

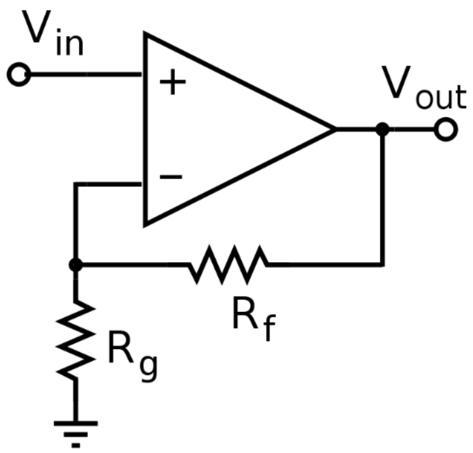


Figure 12 Operational Amplifier (From [27])

Figure 12 represents a Closed-loop amplifier. The relation between the solar cell input voltage V_{in} [VDC] and the amplified output voltage V_{out} [VDC] can be manipulated to a desired value using the resistors R_f and R_g [Ω]. The relation can be seen in the Equation (32).

$$V_{out} = V_{in}\left(1 + \frac{R_f}{R_g}\right) \quad (32)$$

The values of R_f and R_g have been selected based on two criteria. Firstly, the amplification factor has to be high enough to permit sufficient units of measurements in Arduino's analogical input. Secondly the values of the resistors have to be high enough so the current that flows among them does not exceed the nominal and recommended values of the operational amplifier. The values that have been considered are represented in the Equation

$$\begin{cases} R_f = 50\ [k\Omega] \\ R_g = 3.9\ [k\Omega] \\ Amplification\ factor = 13.83\ [-] \end{cases} \quad (33)$$

With the implementation of the amplifier, the amplified output signal varies in 2.2 [VDC] for the complete range of movement of the levitating object. That means that 450 steps out of 1024 are available (Equation (34)) to represent the range of the movement of the sphere, which has been considered valid from the accuracy point of view.

$$Arduino\ Steps = \frac{Voltage\ Rage}{Resolution} = \frac{2.2\ [VDC]}{5\ [VDC]\ /1024\ [Steps]} \cong 450\ [Steps] \quad (34)$$

In order to know which is the relation between the location of the ball and the light sensors output voltage, a calibration test has been done. Using a height gauge, the ball has been located in a different known position and the output voltage from the sensor has been measured. This procedure has been repeated for 20 points. In the following Figure 13, the lay out of the test can be seen.

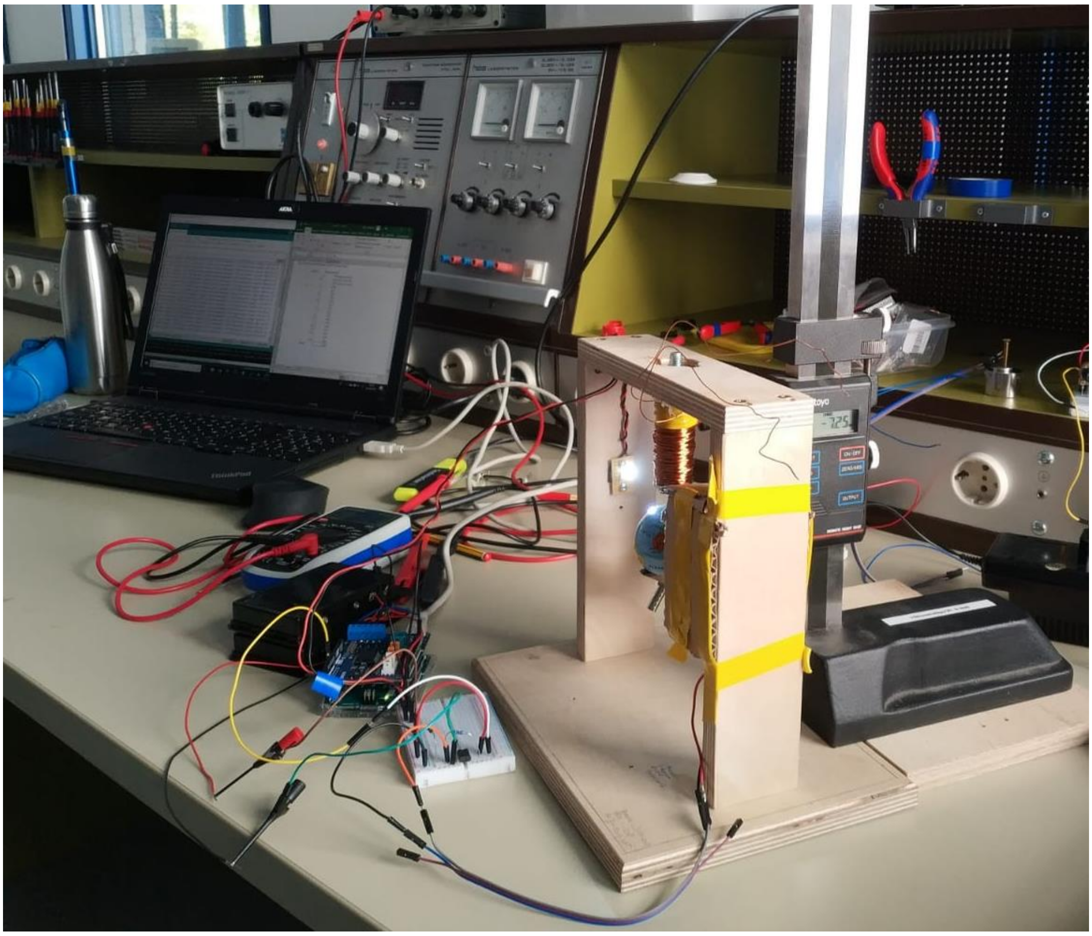
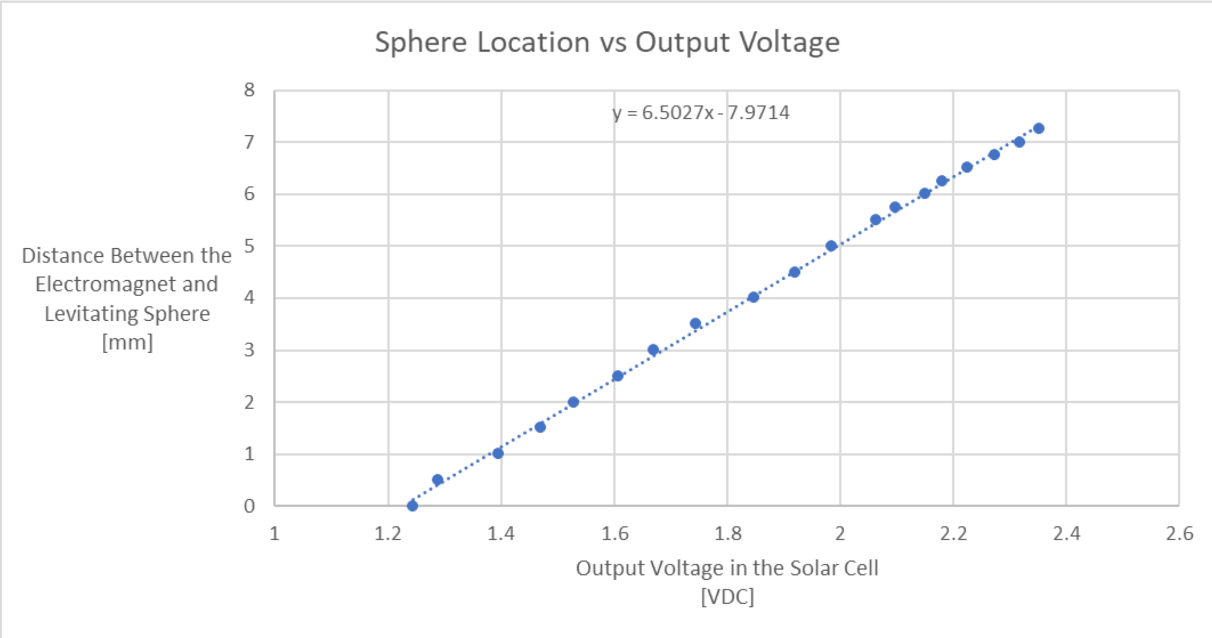


Figure 13 Sensor Calibration Test (Own made)

The results of the text can be seen in the following Graph 1.



Graph 1 Sensor Calibration Test

In Graph 1 can be seen the linear behaviour of the system, which will simplify the calculations of the control system.

1.3.3 Arduino

Arduino [20] is a free hardware based on a low-cost microcontroller. Arduino is not considered to be a high capability microcontroller regarding the speed of the system (cycle time, memory, capacity, pins readout speed and range). Nevertheless, due to its simplicity when it comes to programming and the reasonably good characteristics for the requirements of this project, the use of Arduino is a good choice. Arduinos software is based on C/C++.

Arduino will process the information of the sensors. Both the sensors are analogical inputs for Arduino. The control system parameters will be programmed in Arduino as a code. Arduino will read the controlled variable and will calculate an output for the transistor which controls the input voltage of the electromagnet. This way the control of the position of the ball will be done manipulating the mentioned voltage.

In this project “Arduino Mega” [28] will be used for a matter of availability. A simpler model like “Arduino Uno” [29] would be more adequate in the sense that is adapting better to the simple and low requirements of the system.

Figure 14 Arduino Mega (Own made)

One of the limitations of Arduino mega, is that is not able to handle voltages bigger than 5 [VDC]. As in this project there are components that are fed with 12 [VDC], Arduino should be able to manipulate this voltage. For this reason, Arduino Motor Shield rev3 [21] has been implemented. Arduino Motor Shield rev3 is able to feed two channels with different voltages independently at the same time, adapting to the requirements that are needed in the project as it will be seen in the following chapter 1.3.6, where the electric circuit is explained.

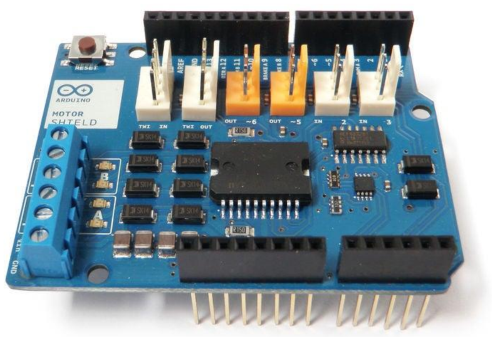


Figure 15 Arduino Motor Shield rev3 (from [21])

1.3.4 Levitated Object

The object that will be make levitate is a void sphere made of a ferromagnetic material. As it can be seen in the Figure 16.

Figure 16 Levitating Object

The possibility to use a magnetic material as a levitation object was analysed. Using material with magnetic properties changes the system. First, the magnetic force that appear in the levitating object is stronger than the one in a ferromagnetic material for the same conditions [30] (distance between the electromagnet and the levitating material and current flowing in the electromagnet), making possible to increase the range of the movement of the object or to reduce the current in the electromagnet for the same range of movement. On the other hand, if the magnetic force is increased, it could be more difficult to design a control system because the dynamics of the system would be faster.

Secondly, sensors based on magnetic fields measurements could be used to measure the magnetic field created by the levitating object and relate it with the distance. Which is a simpler lay out than using light sensor in the sense that an emitter is not needed.

The fact of working with magnets changes the nature of the system itself and makes the equations described in the chapter 1.2.2 not valid. Because of these reasons, the idea of working with a magnetic material instead of a ferromagnetic was dismissed.

1.3.5 Frame

The frame, is the structure where all the previous components are attached. It requires to be stiff and sturdy.

The frame is made of wood and has a "bridge" shape. In the Figure 17, the frame can be seen.

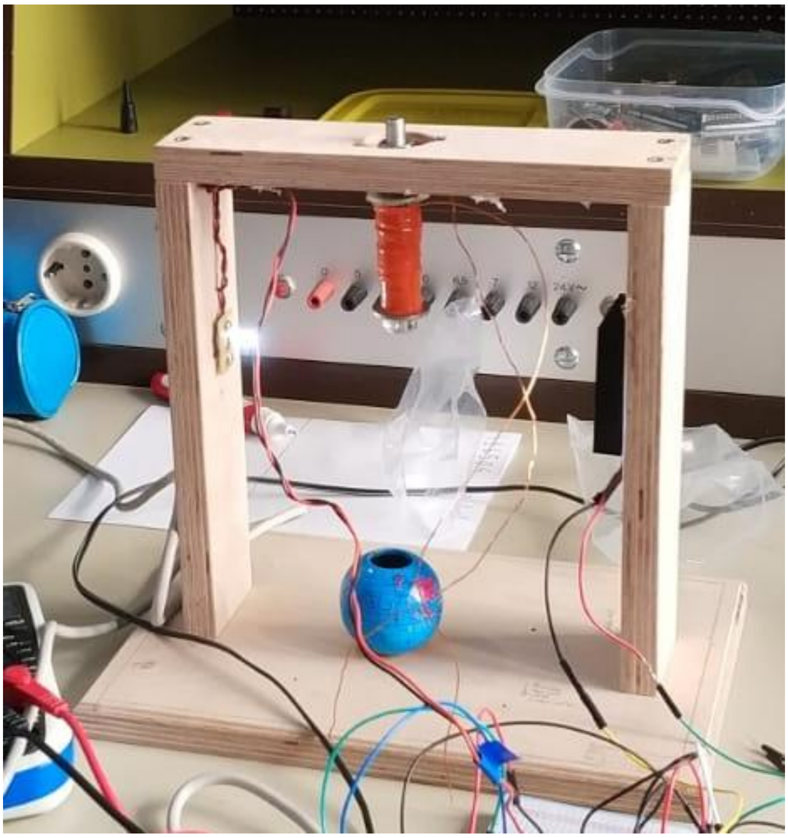


Figure 17 Frame (Own made)

1.3.6 Electric Circuit

To connect every component to its appropriate connection an electric circuit has to be used. The electric circuit collects all the electrical components that are needed in the project in an ergonomic way. The components have been welded in a welding board as it can be seen in the Figure 18.

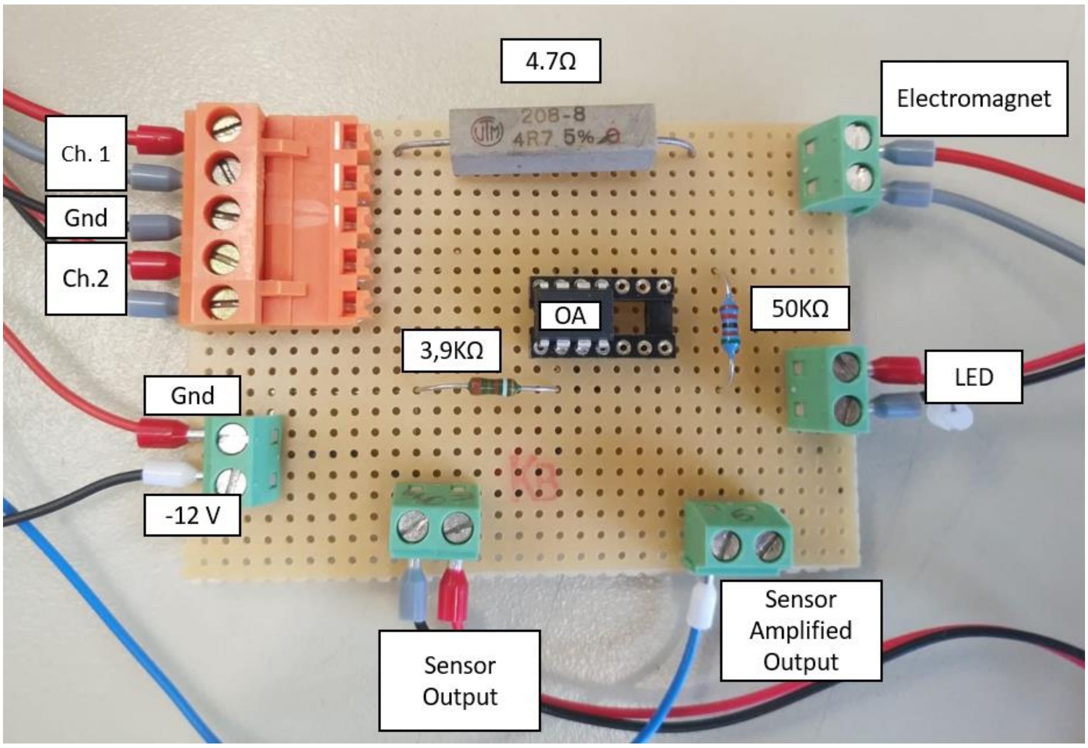


Figure 18 Electric Circuit

The electric circuit can be divided in two sub-circuits, one for the electromagnet (Channel 1) and another one for the sensor and the LED emitter (Channel 2).

Channel 1 (Ch.1) feeds the electromagnet. The voltage that feeds the Ch.1 is manipulated by Arduino and it is the manipulated variable of the system. The algorithm that has been used to calculate the control voltage signal is described in the chapter where the control system is explained 1.5. In the Figure 19 the lay out of the Ch.1 can be seen.

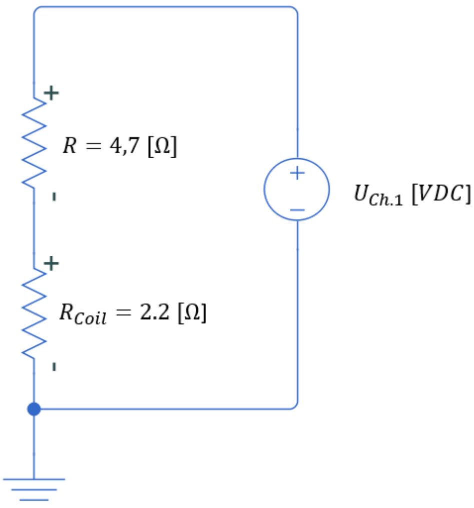


Figure 19 Channel 1 (Tool Simulink)

Ch.1 is fed by Arduino with a Pulse Width Modulation (PWM) signal. That means that the voltage and the current it this channel will vary. The amplifier (OP in Figure 18) and the LED emitter though, require a constant voltage. Because of this a second channel (Ch.2) is required. The lay out of the channel 2 is shown in the Figure 20.

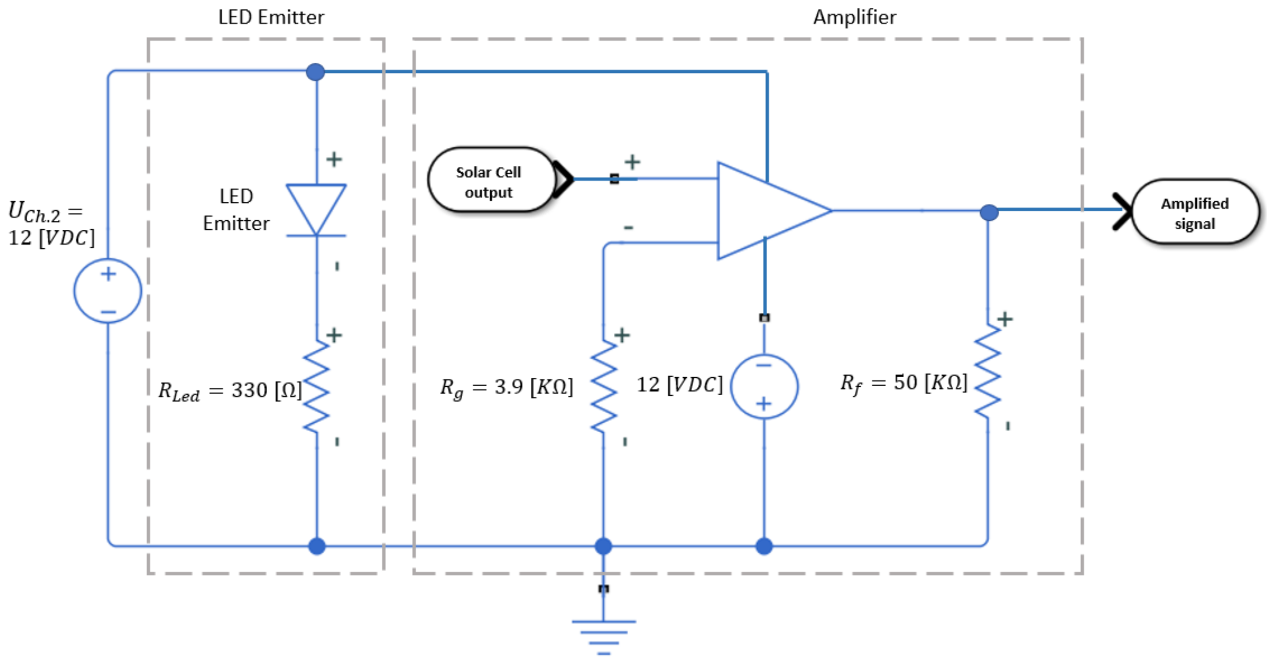


Figure 20 Channel 2 (Tool Simulink)

1.4 Simulink Model

According the developer of the software MathWorks [31], "Simulink is a block diagram environment for multidomain simulation and Model-Based Design. It supports system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems. "

Simulink belongs to MATLAB software and all the functionalities from MATLAB can be easily implemented in Simulink. That is the reason why for the design of the control system and in order to understand the behaviour of the system, a model in Simulink has been built. This model is based on the theory that has been summarised in the first chapter of this document 1.2.

As it was mentioned above, it is difficult to build a reliable model of the electromagnet because the dynamics of the real system does not approach to the theoretical Equation (23). For this reason, the goal of using a Simulink model is not to accurately represent the system, but help to understand the system and its dynamics. Simulink's simulation tool allows to change the parameters of the system and the control, making possible to identify tendencies and relate them with variables. The tuning of the control system, has been based on the results obtained in Simulink's simulations.

Summarizing, the levitating system can be divided in several block. In the first block the levitating phenomena is represented, and modelizes the magnetic force. Secondly, the mechanical force balance block determines the location of the ball. After that, the sensor block measures the location of the ball. The control system block reads the measurements from the previous block and compares it with the set point, calculating a signal which will control the electromagnets input voltage. Finally, in the electric block the electrical behaviour of the system is modelized. This is represented in the Figure 21. An additional block could be added, representing the noise and the disturbances.

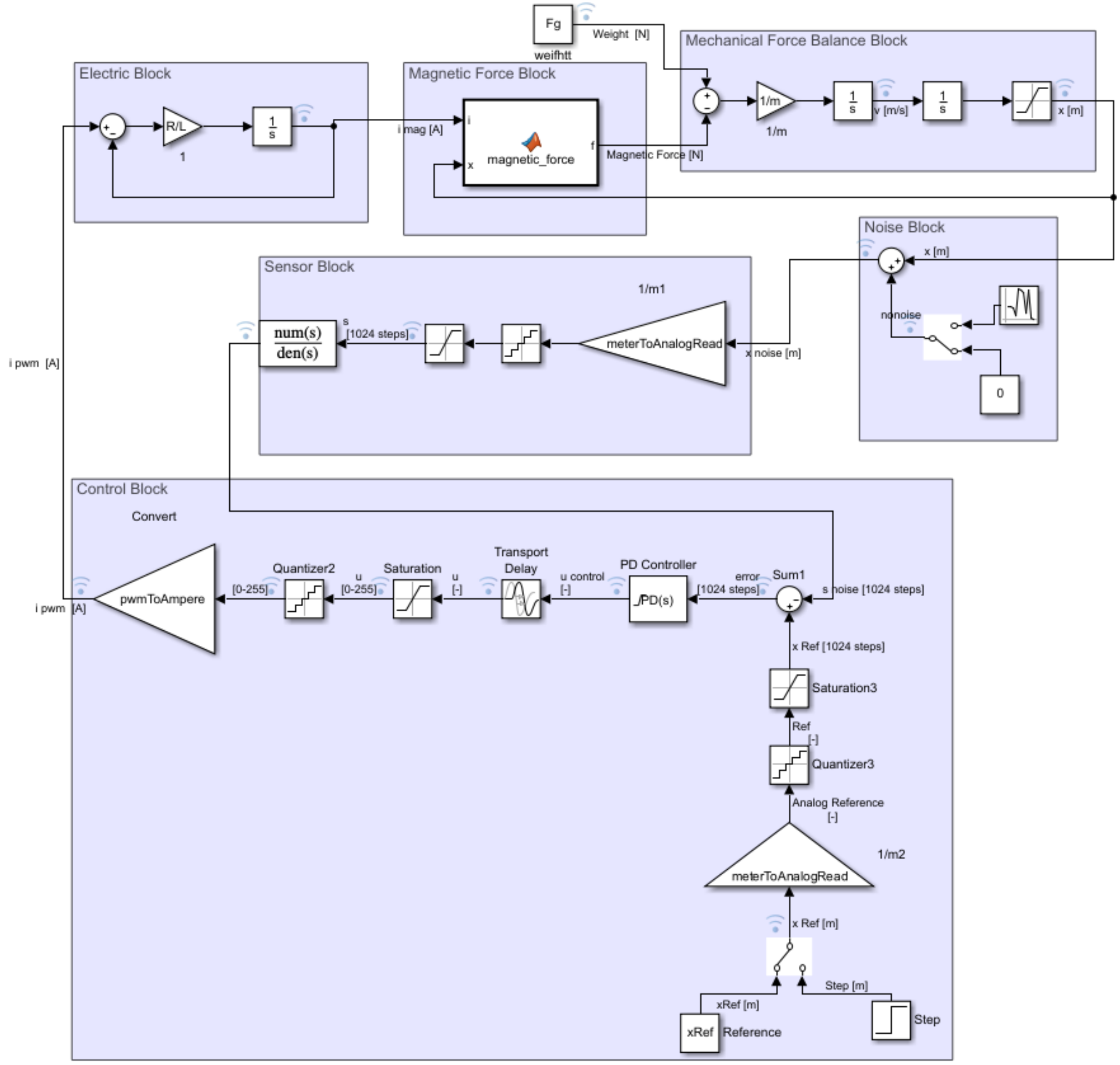


Figure 21 Simulink Model (Tool Simulink)

1.4.1 Magnetic Block

In the magnetic block, the magnetic phenomena between the electromagnet and a ferromagnetic material is modelized. In the 1.2.2 chapter of this document, the equations and the science behind this technology can be seen.

In the Simulink model, the Equation (10) is written as a MATLAB function. The code which calculates the magnetic force is based on the Equation (23) and can be seen in the appendix as Magnetic Force Code.

The constant value k_c [N/A^2] has been obtained theoretically using the Equation (11). Using the values obtained in Equation (25) and (26), the value of value of k_c [N/A^2] is Equation (35).

$$k_C = 7.5 * 10^{-4} \,[\mathrm{N/A^2}] \tag{35}$$

The block is represented in Simulink as a simplified block as it can be seen in the following figure.

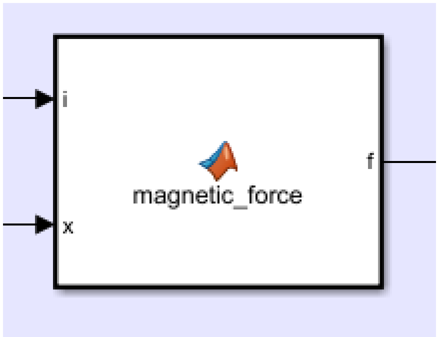


Figure 22 Magnetic block in Simulink (Tool Simulink)

It can be seen that there are two inputs. One is the position of the levitating object and the second one is the current flowing in the electromagnet. The output would be the magnetic force that acts in the sphere in Newtons.

1.4.2 Mechanical Force Balance Block

In this block, the Equation (4) is modelized. As it was said, this equation comes from applying the Newtons second law to the system.

In Simulink this balance can be represented graphically as a sum of blocks as it can be seen in the Figure 23.

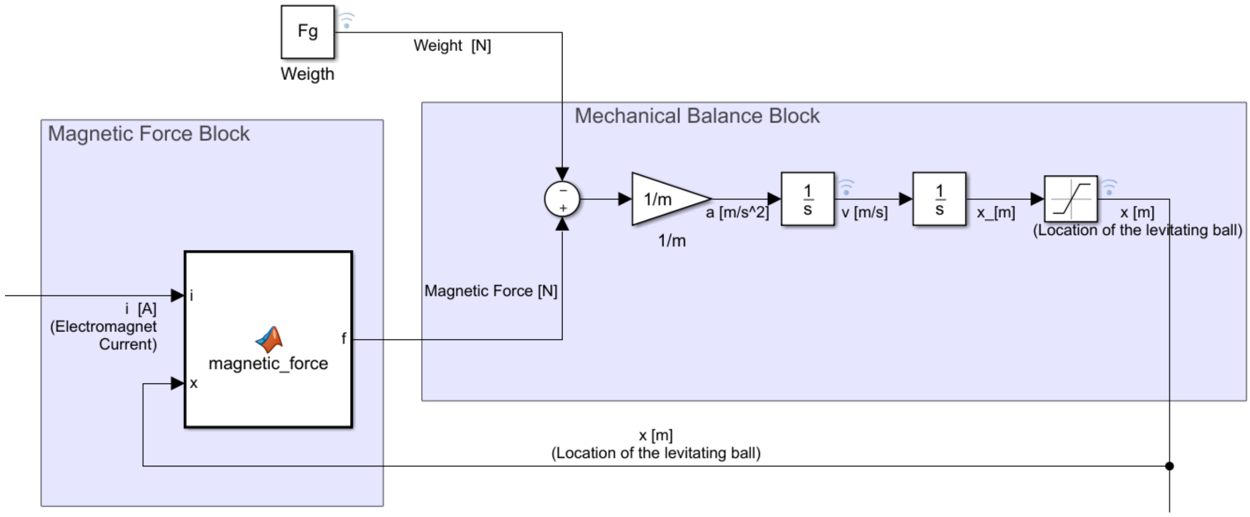


Figure 23 Magnetic Force and Mechanical Balance Block (Tool Simulink)

The weight of the levitating ball, which has been defined as a constant, is subtracted with the magnetic force that was coming from the magnetic force function block. That results in an accelerated mass. In the following steps of the block, the location of the levitating object is obtained after dividing the sum of forces ($F_G - F_{Mag}$) into the mass m and integrating the acceleration two times.

A saturation block is added to represent the geometrical limits of the real working range. This is necessary, because according to Equation (10) , the magnetic force (F_{Mag}) tends to infinity when the distance approaches zero. Considering real magnets, the magnetic force is saturated when the distance diminishes. Those limits should not be crossed because the system loses the linear behaviour. Once in the non-linear range, the magnet is not strong enough to make the ball levitate when crossing the upper bound, or the magnetic force increases to uncontrolled high values quickly when crossing the lower bound. Both situations are not stable and once there, the stabilization of the system is not possible.

This fact makes the levitating system sensitive against disturbances. At the same time, make the sensors and the control system to require fast processing requirements. Both the sensor and Arduino, have experimentally demonstrated being fast enough for the requirements of the project.

1.4.3 Noise and Disturbances Block

In the magnetic levitating system of this project, relevant mechanical disturbances in the system are not expected. Nevertheless, the sensor in charge of measuring the location of the ball is a source of disturbances. The origin of the disturbances depends on the technology of the sensor, in the case of a light sensor, the main disturbance in the measurements is the light that illuminates the room where the experiment is taking part.

To make a model of the mentioned disturbances, the noise and disturbances block is implemented in Simulink. The graphical appearance of the block is seen in Figure 24.

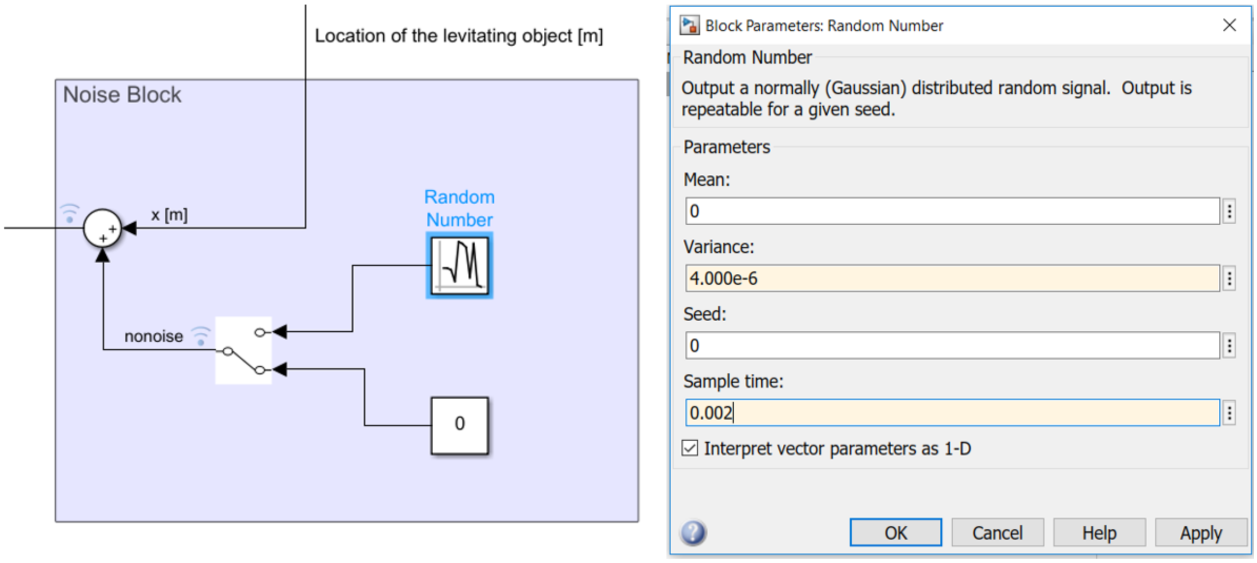


Figure 24 Noise and Disturbance Block (Tool Simulink)

As it can be seen the disturbance is represented as a random number, where the variance and the sample time are modelized. The values of the variance and the sample have been estimated and can be seen in the Figure 24 .

1.4.4 Sensor Block

In the Simulink's sensor block, the sensor that is used in the prototype is modelized. The most important parameters of the sensor are the range of the measurements, the sensibility and the time constant. The sensor can be represented in Simulink as it can graphically be seen in the Figure 25. The input of this block is connected to the output of the noise block which was explained in the previous subchapter 1.4.3.

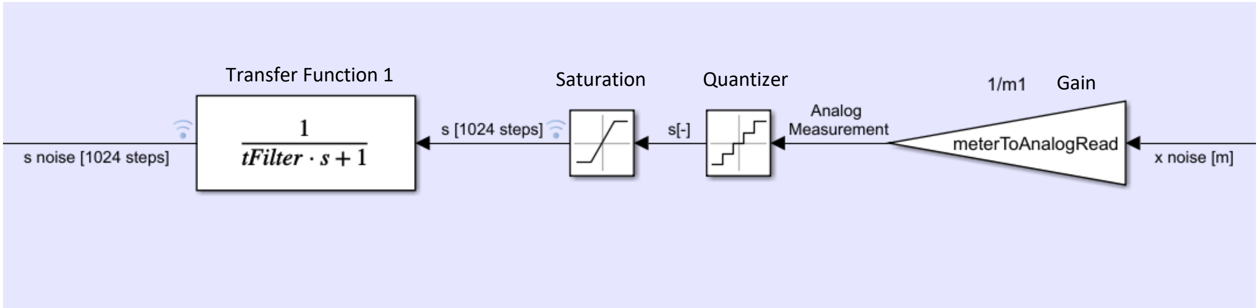


Figure 25 Sensor Block (Tool Simulink)

In the chapter where the sensor was explained (1.3.2) was mentioned, that there is a linear relation between that voltage and the position of the ball. That relation is modelized as a gain in the "meterToAnalogRead" block. The relation has based on the calibration of the sensor which has been described before in the chapter of the sensor 1.3.2 and defined in the Graph 1 Sensor Calibration Test. Its value can be seen in the following Equation (36).

$$Analog\ Measurement\ [-] = \frac{Location\ of\ the\ Sphere\ [m]}{6.5027} + 7.9714 \tag{36}$$

The measurement of the sensor will be read by Arduino and the analogical inputs on Arduino have a size of 10 bits. Which means a resolution between readings of 5 volts / 1024 units. This discretization is represented in the block with the stair appearance "Quantizer" [32] block which follows the gain block. After the Quantizer there is the saturation block, which stablishes the limits of the measurement.

The "Transfer Function 1" that can be seen after the saturation block is a filter. In order to not be highly influence by the disturbances in the measurements. This block, represents a low pass filter and decreases the influence of high frequency oscillations. The parameters that defines the block is t_{Filter} [s] and its value is.

$$t_{Filter} = 0.005\ [s] \tag{37}$$

1.4.5 Control System Block

In this block the control action is modelized. In the magnetic levitating system of this project, the controlled variable is the location of the ferromagnetic sphere. The manipulated variable is the voltage that feeds the electromagnet. This voltage is controlled through Pulse Width Modulation

(PWM) technology. The Control System Block is graphically represented in Simulink as it can be seen in the Figure 26.

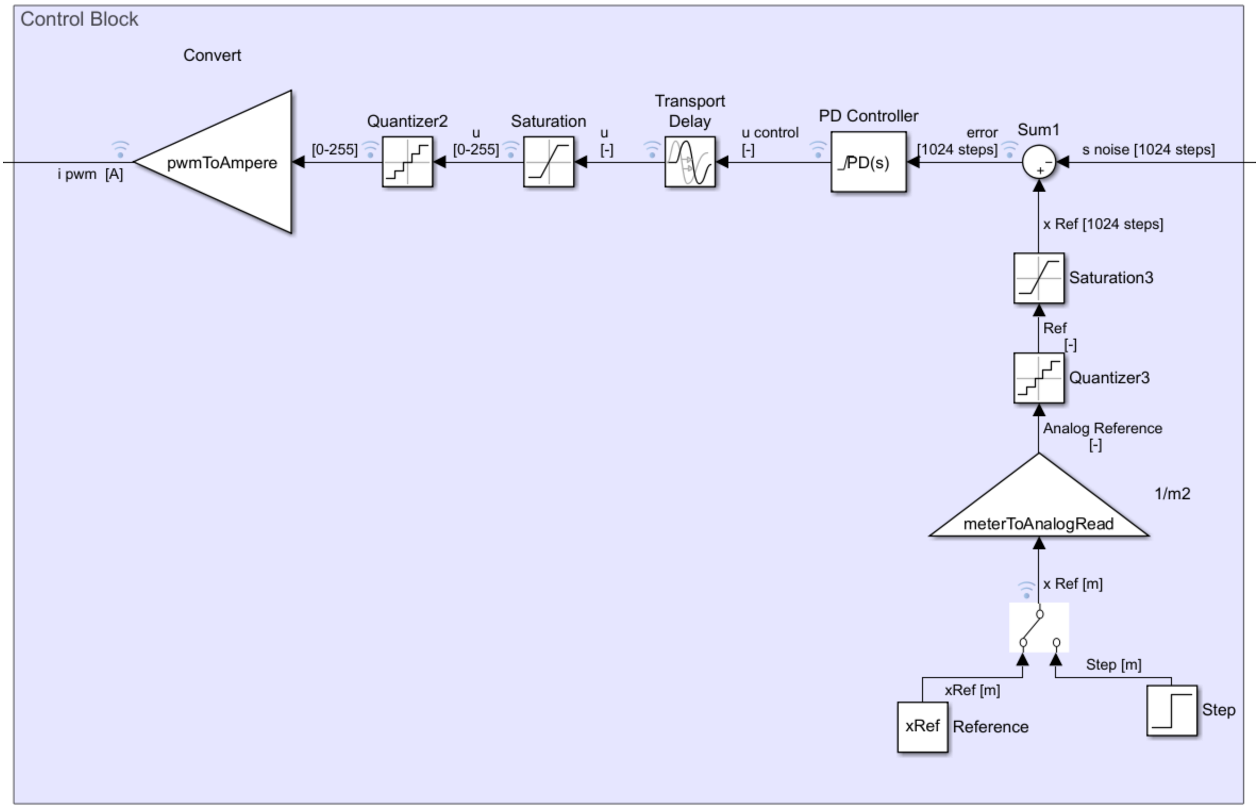


Figure 26 Control System Block (Tool Simulink)

First of all, the distance that is desired between the electromagnet and the levitating object is stablished as a Set Point (xRef [m]). In the following steps, the reference distance or location is converted to an analogical value through the same gain (meterToAnalogRead) that was used in the sensor block (Equation (36)). After that, the signal is discretized to an integer value in the Quantizer3. The Saturation3, set the limits of the signal to a value between 0 and 1023. The result of this subblock, is a reference signal value between 0 and 1023 that represents the desired location of the levitating object.

In the Sum block, the measurement signal that was made by the sensor (chapter 1.4.4) and the reference Set Point signal are compared. The resulting signal is an error. This error represents the difference that there is in the system between the desired position for the levitating object and the real position.

The error signal that has been obtained feeds the block that represents the control system (PD controller). As it will be explained and justified with more details in the following chapter 1.5, the control system is based on a Proportional and Derivative technology. The PD Controller, can be schematically represented as it can be seen in the appendix PD Controller and simplified in the following Figure 27.

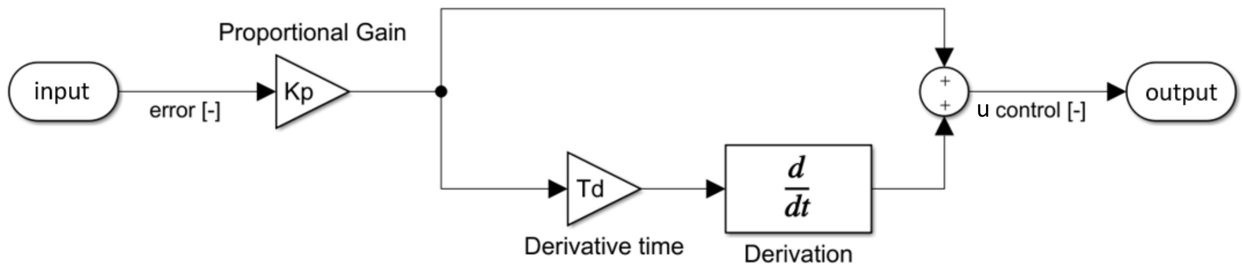


Figure 27 PD Control simplified (Tool Simulink)

It can be seen, that the control response multiplies the previously mentioned error per a Proportional Gain K_p factor. On the other hand, it also derivates de error signal, obtaining the slope, and multiplies per the Derivative time gain T_d. Both signals are summed and the result is the control system response.

After the PD control block, the control signal ($i_{Control}$) is delayed in a “Transport Delay” block. In this block the time that the control system needs for processing the data is modelized.

In the following step, Saturation block sets the limits of the electromagnets supply voltage. The maximum voltage of 12 volts would be represented as an analogical value of 255 steps. In the following Quantizer (Quantizer2) block of the Figure 26, the signal is approach to an integer number in order to represent the non-continuous behaviour of Arduino’s power supply.

The resulting signal (i PWM) feeds the gain that follows in the Figure 26. Arduino Motor Shield is able to vary the input voltage of the electromagnet through a Pulse Width Modulation technology. The output signal from the PD controller is a value that represents the average value of a PWM voltage signal. This voltage, $u_{Control}$ [-], is converted to a current in the Simulink model in the “pwmToAmpere” block of the Figure 26. The conversion is based on the application of the Ohm’s Law (Figure 19).

$$i_{Control}\ [-] = \frac{u_{Control}[-]}{R_C + R} \tag{38}$$

For this reason, the VoltageToCurrent block has a value of Equation (13).

$$VoltageToCurrent[-] = \frac{i_{Control}[-]}{u_{Control}[-]} = \frac{1\ [-]}{R_C + R} = \frac{1\ [-]}{2.2 + 4.7} = 0.145[-] \tag{39}$$

Summarizing, the control system has as input the location of the levitating object, and after comparing the measurement with the Set Point value, a controlled current that will feed the electromagnet goes as an output.

1.4.6 Electric Block

The electric block represents the dynamical influence of the resistor and the inductance of the coil. Arduino manipulates the input voltage of the electromagnet, the values that the voltage can take goes from 0 to 12 volts with 255 steps between them. The applied voltage generates a current flow. It has been explained in this document, how to get the relation between the current and the applied voltage (Equation (3)). As has been explained above, a resistor has been added in series to the electromagnet.

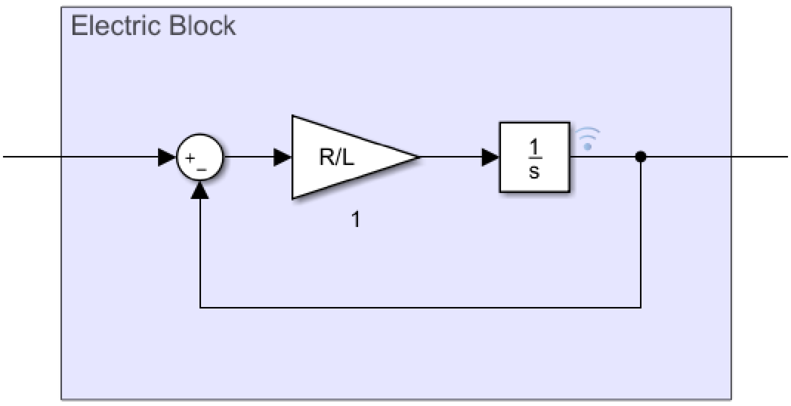


Figure 28 Electric Block (Tool Simulink)

The resistance of the circuit, Figure 19, and the inductance can be seen in the following Equation (43). The inductance has been estimated.

$$R_{Total} = 6.9\ [\Omega] \tag{40}$$
$$L_C = 0.003[H]$$

1.4.7 Results of the Simulation

Simulink tool helps the understanding of the dynamical system and allows to identify tendencies in the response with the modification of the different variables that affects to the system. In this subchapter, how the variables affect the stability of the system is going to be study. Among the variables that can be study, the variables of the hardware like the electromagnets input voltage, the size of the electromagnet, the electrical circuit, the intensity of the LED emitter, or the characteristics of the sensor will remain fixed and the ones that can be modify with the software will be study. The reason of this strategy of simulation is because the aim of the Simulink model is to identify the tendencies that affects to a model which has been already built. This way, the tuning of the control parameters in Arduino are based on the behaviour that can be expected according the simulations. The parameters of the simulations can be seen in the code which is shown in the appendix MATLAB Code.

- Effect of the proportional gain K_P

The proportional gain of the control block is proportional to the error between the measurement and the set point. In the Figure 29 can be seen, how the K_P affect to the response. The rest of the parameters of the simulation remain constant.

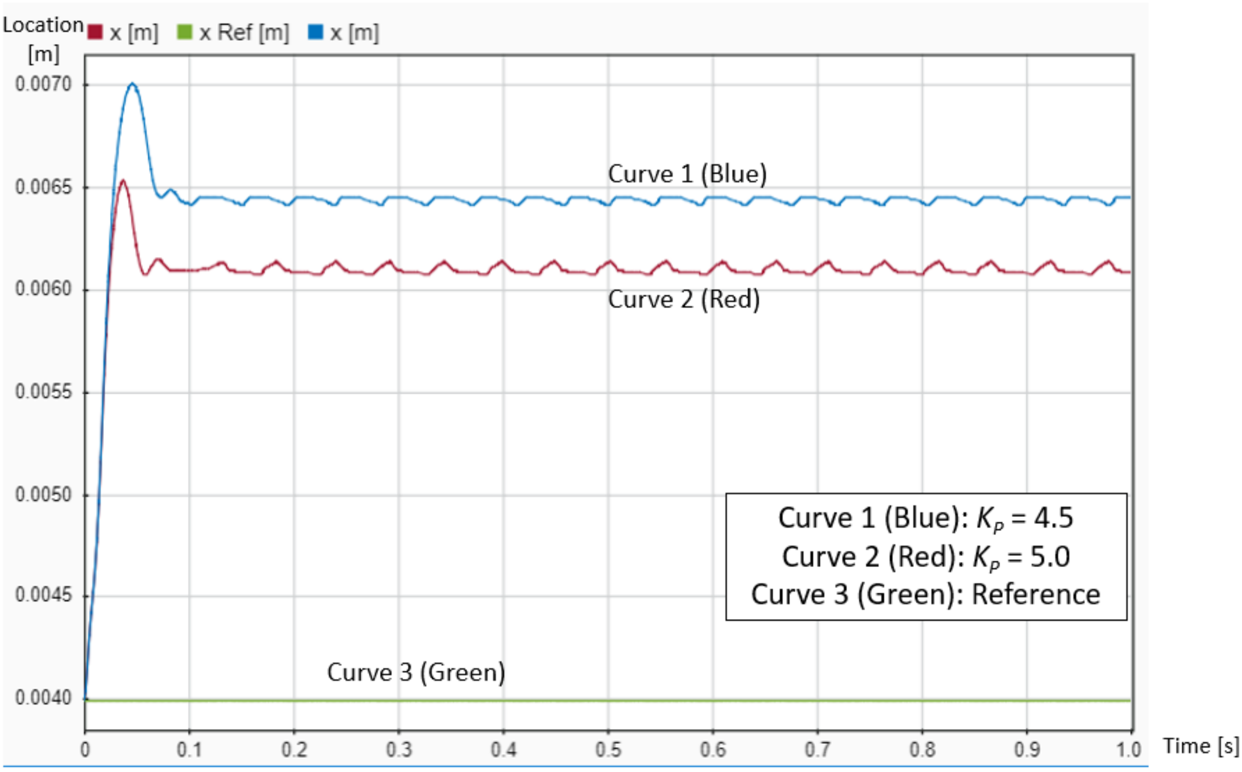


Figure 29 K_P Effect (tool Simulink)

It can be seen how a higher gain makes the answer reach the equilibrium point faster, but also increases the amplitude of the vibrations. As the control signal is proportional to the error, a higher K_P makes the equilibrium point being closer from the Set Point (reference).

- Effect of the derivative time T_D

The derivative action acts as a damping for the system as it can be seen in the Figure 30.

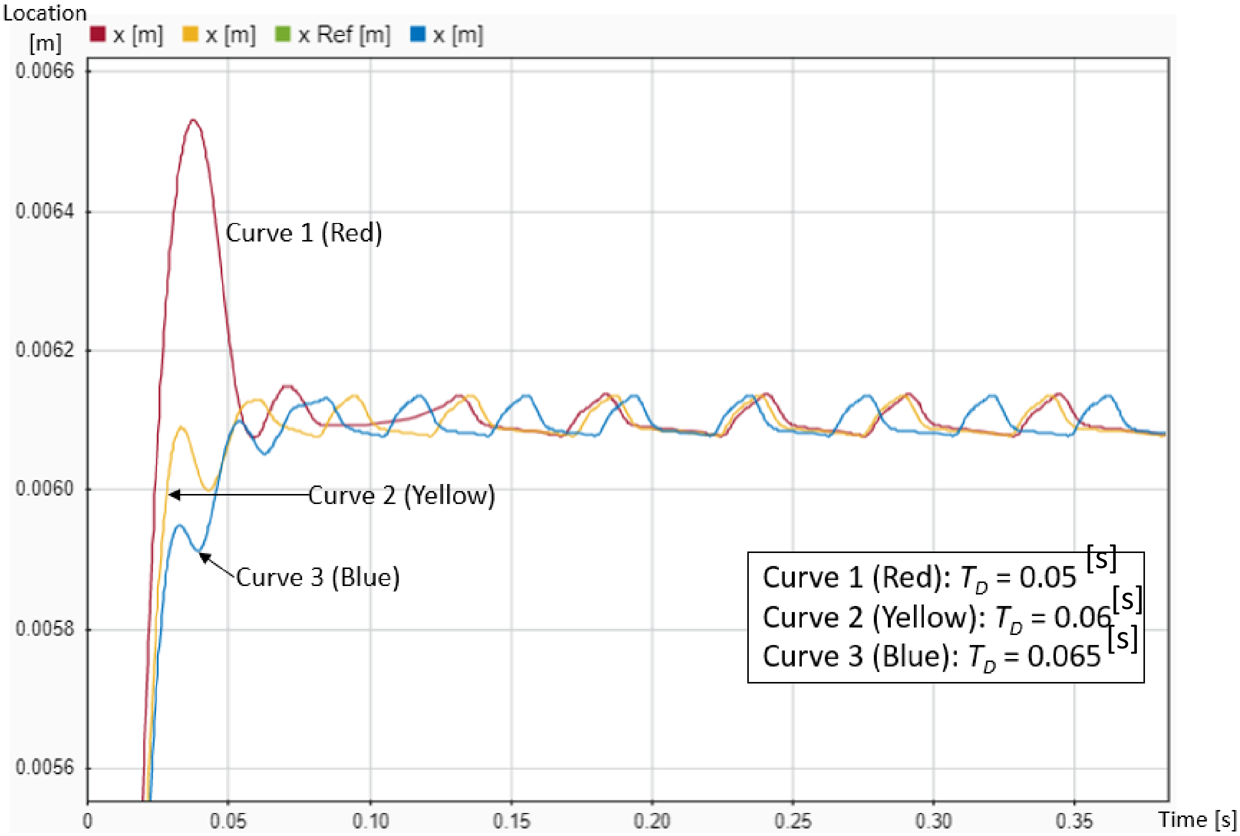


Figure 30 T_D Effect (Tool Simulink)

It can be seen that when the T_D is higher, the response is damped.

- Effect of the noise

The noise makes the derivative part of the controller to increase the control signal. For this reason, the equilibrium point of the sphere decreases. That can be seen in the Figure 31.

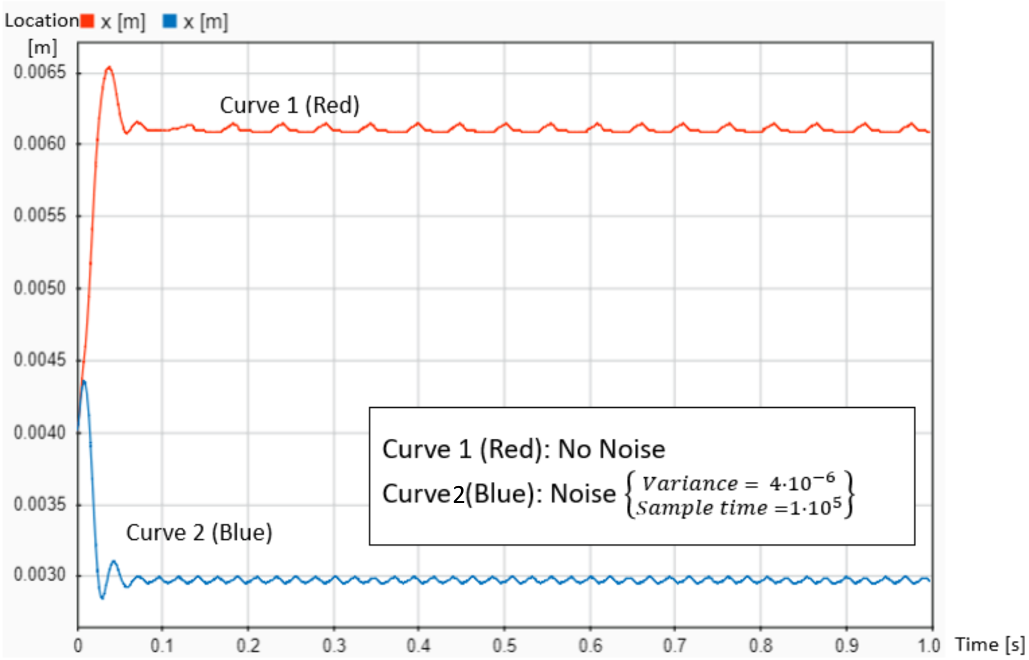


Figure 31 Effect of the noise (Tool Simulink)

- Weight of the levitated object

If the weight of the levitating object increases, the equilibrium point is located in a higher distance. This fact can be seen in Figure 32.

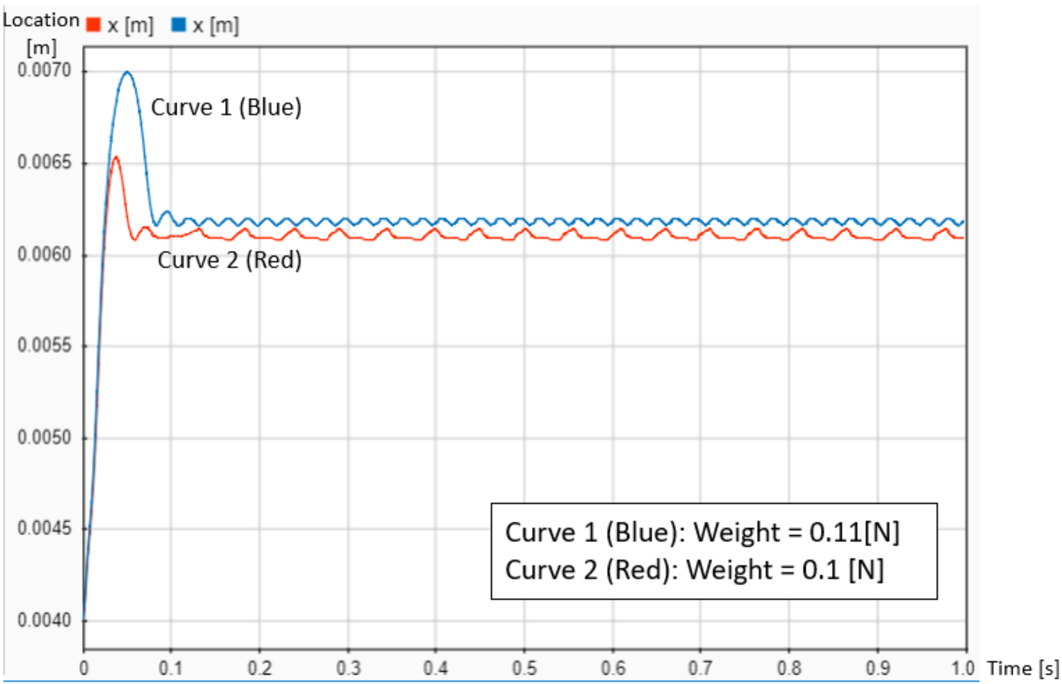


Figure 32 Weight effect (Tool Simulink)

- Measurement output filter

The measurement filter increases the first overshoot when is filter time constant T_f is low. When T_f increases the first overshoot decreases but the response in the steady state increases its fluctuations. The results can be observed in the Figure 33.

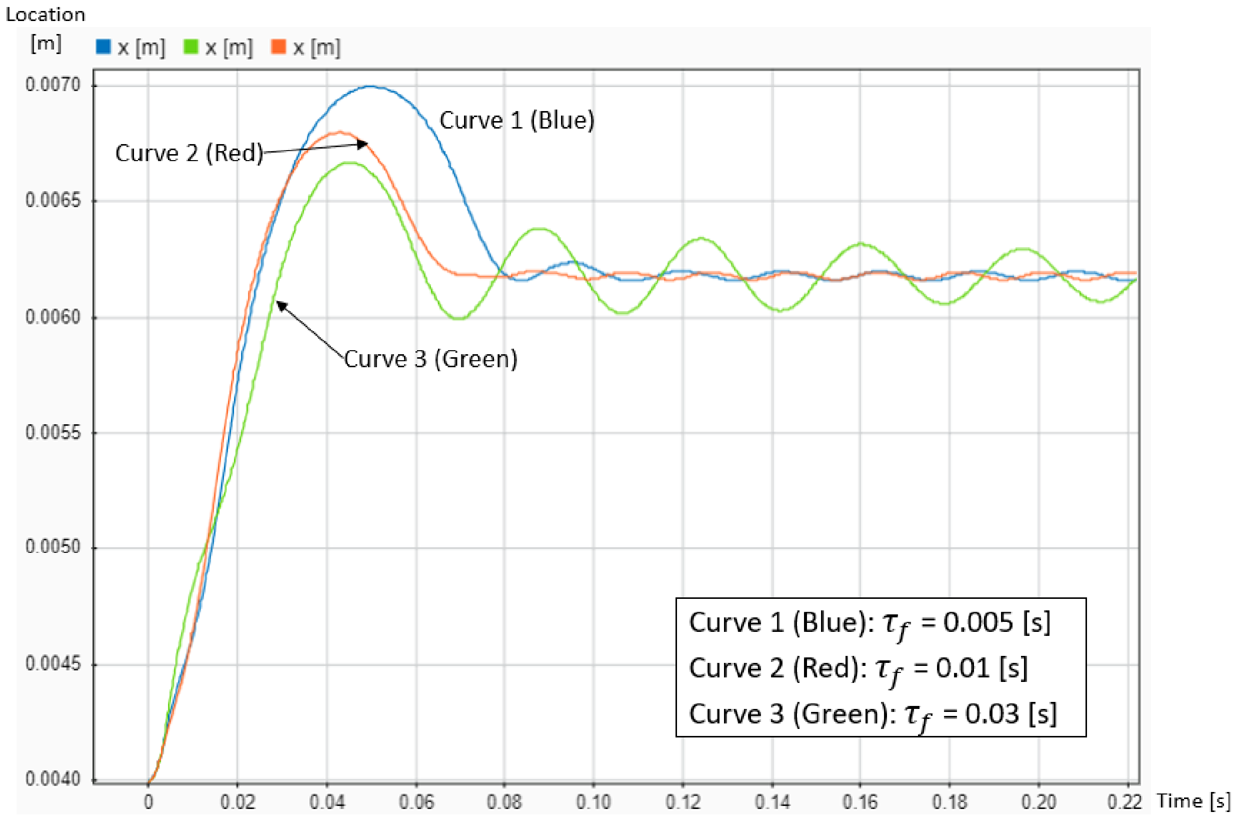


Figure 33 Measurements output filter

1.5 Control System

Theoretically, the magnetic force induced by the current that flows in the electromagnet is equal to the weight of the ball, making the ball levitate. In practice, there are many sources of disturbances. For example, the environmental light affects the measurements of the sensor, on the other hand the inductance of the electromagnet's disturbs the electromagnetic force. Those disturbances can originate acceleration forces on the levitating material, breaking the force balance. The goal of the control system is to maintain the static and dynamic equilibrium and face the disturbances, using the information provided by the sensor.

The control system designed for this project has two parts. First of all, the proportional part, that generates a control signal which is proportional to the difference between the set point and the measurement. Second, the derivative part, generates a signal which is proportional to how the error signal changes. The derivative part makes the response faster and it can be considered as a predictive controller in the sense that is using the slope of the error curve. The derivative signal does not response to a constant error, for this reason, it is combined with a proportional part. How the control signal is obtained can be seen in the Equation (41).

$$Control\ signal\ = K_P \left(error + \ T_D \frac{d(error)}{dt} \right) \tag{41}$$

The control system diagram can be seen in the following Figure 34. Parts that are used in the hardware like the operational amplifier or Arduino's analogue units are neglect in the study of the control system because they do not influence in the essence of the control system.

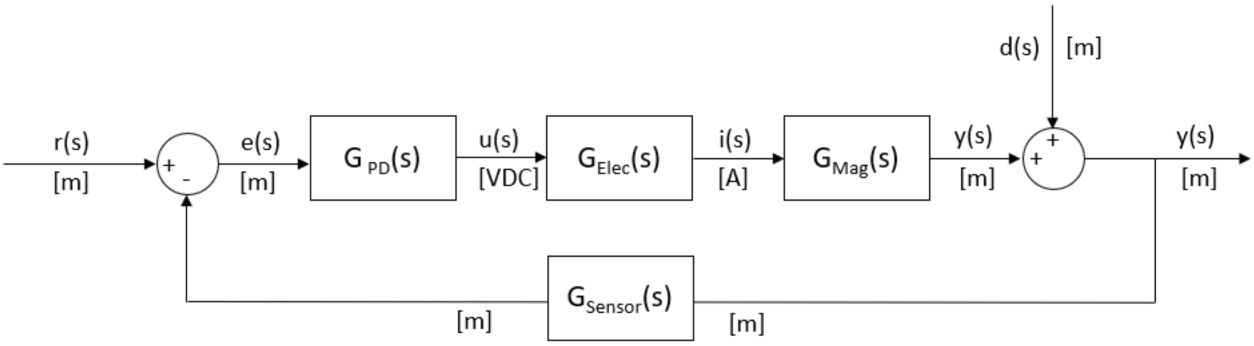


Figure 34 Control Loop

$$G_{PD}(s) = K_P\,(1 + T_D s)$$

$$G_{Elec}(s) = \frac{1/R_C}{1 + L_C s/R_C}$$

$$G_{Mag}(s) = \frac{-k_c \dfrac{2I_0}{m{y_0}^2}}{s^2 - \left[k_c \dfrac{2{I_0}^2}{m{y_0}^3}\right]} = \frac{\alpha}{s^2 - \beta} \tag{42}$$

$$G_{Sensor} = \frac{1}{\tau_f s + 1}$$

Where:

- r(s): Set Point [m]
- e(s): Error [m]
- u(s): Control Signal [VDC]
- i(s): Current in the electromagnet [A]
- y(s): Location of the levitating object [m]
- d(s): Disturbances [m]

The transfer functions $G_{Elec}(s)$ relates the voltage and the current in the electromagnet and comes from Equation (3). $G_{Mag}(s)$ comes from Equation (20) and represents the magnetic force. In the Equation (42) the function has been simplified. The function $G_{PD}(s)$ represents the proportional derivative controller. As it was explained, it consists in two parts. K_P represents the proportional gain and T_D the derivative gain.

$G_{Sensor}(s)$ modelizes the sensor with a filter. A filter in the measurements is required because the sensor measurements signal is noisy. A noisy signal as an input of the control system would cause that the slope of the error signal would be noisy as well, so the derivative part of the controller would amplify the noise of the system. To decrease that effect, the measurements will combine past measurements with on time measurements. This action is implemented as a code in Arduino and can be represented as a first order transfer function. T_f is the time constant of the transfer function and the gain is equal to 1 because the aim of this block is not to amplify the measurement.

The disturbances, *d(s),* cannot be predicted, so it can assume that they affect directly to the location of the object.

Now, the stability of the linearized system is going to be analysed. The open loop transfer function $G_{OL}(s)$ can be calculated Equation (43).

$$G_{OL}(s) = G_{PD}(s)\, G_{Elec}(s)\, G_{Mag}(s) \tag{43}$$

A system is stable only when all the poles of the transfer function of the system (close loop) are in the left half of the complex plan. Because of this, the close loop transfer function $G_{CL}(s)$ has to be obtained in order to analyse the stability.

$$G_{CL}(s) = \frac{G_{PD}(s)\, G_{Elec}(s)\, G_{Mag}(s)}{1 + G_{PD}(s)\, G_{Elec}(s)\, G_{Mag}(s)\, G_{Sensor}(s)} \tag{44}$$

Operating, Equation (45) is obtained.

$$G_{CL}(s) = \\ = \frac{\frac{\alpha k_p}{R_C}\left[T_D \tau_f\, s^2 + \left(T_D + \tau_f\right)s + 1\right]}{\frac{L_C \tau_f}{R_C} s^4 + \left(\tau_f + \frac{L_C}{R_C}\right)s^3 + \left(1 - \frac{L_C \tau_f \beta}{R_C}\right)s^2 + \left(\frac{k_P T_D \alpha}{R_C} - \tau_f \beta - \frac{L_C \beta}{R_C}\right)s + \frac{k_P \alpha}{R_C} - \beta} \tag{45}$$

The stability of the system can be analysed with the Routh-Hurwitz stability criterion [33]. The denominator of the Equation (45) represents the characteristic polynomial, and it can be expressed in the following way Equation (46). The aim of using this method, is to identify where are poles of the system. If the poles are in the left half of the complex plan the system will be stable.

$$P(s) = a_n s^n + \; a_{n-1} s^{n-1} + a_{n-2} s^{n-2} + \cdots + a_1 s + a_0 \tag{46}$$

If the previous polynomial has a negative coefficient a_i, the system would not be stable. Apart from that, the algorithm that is shown in the Figure 35 has also to be checked to demonstrate that the system is stable. The coefficients of the Equation (46) are placed in the indicated places and the coefficient α_1 ,α_2, β_1, β_2 … are obtained.

$$\begin{array}{c|ccccc} s^n & a_n & a_{n-2} & a_{n-4} & \cdots \\ s^{n-1} & a_{n-1} & a_{n-3} & a_{n-5} & \cdots \\ s^{n-2} & \alpha_1 & \alpha_2 & \alpha_3 & \cdots \\ s^{n-3} & \beta_1 & \beta_2 & \beta_3 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s^0 & \delta_1 & & & \end{array}$$

$$\alpha_1 = \frac{(a_{n-1} \cdot a_{n-2}) - (a_n \cdot a_{n-3})}{a_{n-1}} \qquad \beta_1 = \frac{(\alpha_1 \cdot a_{n-3}) - (a_{n-1} \cdot \alpha_2)}{\alpha_1} \quad \cdots$$

$$\alpha_2 = \frac{(a_{n-1} \cdot a_{n-4}) - (a_n \cdot a_{n-5})}{a_{n-1}} \qquad \beta_2 = \frac{(\alpha_1 \cdot a_{n-5}) - (a_{n-1} \cdot \alpha_3)}{\alpha_1} \quad \cdots$$

$$\alpha_3 = \frac{(a_{n-1} \cdot a_{n-6}) - (a_n \cdot a_{n-7})}{a_{n-1}} \qquad \beta_3 = \frac{(\alpha_1 \cdot a_{n-7}) - (a_{n-1} \cdot \alpha_4)}{\alpha_1} \quad \cdots$$

$$\vdots \qquad\qquad \vdots$$

Figure 35 Routh-Hurwitz Algorithm (From [34])

Routh-Hurwitz criterion says that the number of poles in the right half of the complex plan is equal to the number of times that the first column coefficients (α_1, β_1, δ_1…) are changing of sign. In order to have a stable system, none of the poles can be in the right side, for this reason all the coefficient from the first column have to be positive.

The characteristic polynomial of Equation (45) depends on the control parameters K_P and T_D. The rest of the values which appear in the characteristic polynomial are positive. That means that using a PD controller, if the adequate values are given to K_P and T_D, the system can be stable.

As simulation tools are used in this project, the study of the stability of the system is done calculating the poles of the transfer function and checking that all the poles are in the left side of the complex plane. This procedure has been done simulating the close loop transfer function and is explained in the following chapter 1.6.

1.6 Simulation of the Linearized Model

The simulation of the linearized transfer function of the system is studied in this chapter. In the previous chapter 1.4, has been explained the simulation of the non-linear system with Simulink, in this chapter though, the linearized transfer function of the system is simulated. The aim of the simulation of the linearized transfer function is to study the influence of the filters and to analyse the stability of the system.

The linearized close loop transfer function can be seen in the appendix named Transfer Function Code. This transfer function is built the same way as the one shown in the Equation (45) of the previous chapter 1.5, but in this function, apart from the sensor filter that already was in Equation (45), a derivative filter and an output filter have been implemented. Those filters are modelized as first order transfer functions with a time constant.

$$\begin{cases} \tau_{fs} : \text{Time constant of the sensor filter} \\ \tau_{fd} : \text{Time constant of the derivative filter} \\ \tau_{fo} : \text{Time constant of the output filter} \end{cases}$$

The code that has been used can be seen in the appendix Transfer Function Code. This code defines all the variables that appears are involved in the close loop and in the Equations (42). It calculates the poles of the close loop system. That way the stability of the system can be study. Depending on the values that the variables take, the poles of the system can be in the left or right side of the complex plane. The position of the poles will determine the stability of the system. To be stable, all the poles have to be in the left side of the complex plane.

In order to study the influence of the variables in the response of the system, several simulations have been done changing each parameters that affects the system. To analyse the answer of the system, a step input signal has been imposed.

- Results of the simulation
    - Influence of the sensor filter

The sensor filter is used in order to decrease the noise of the measurements. The algorithm of the sensor filter can be seen in the Equation (47).

$$FM = FM_{Old} + \frac{dt}{\tau_{fs}} (UNFM - FM_{Old}) \tag{47}$$

Where:

$$\begin{cases} FM: \ Filtered\ Measurement\ [-] \\ FM_{Old}: Previous\ Value\ of\ Filtered\ Measurement\ [-] \\ dt: Loop\ Sample\ time\ [s] \\ \tau_{fs}: Time\ Constant\ of\ the\ Sensor\ Filter\ [s] \\ UNFM: Unfiltered\ Measurement\ [A] \end{cases}$$

In the following picture can be seen, how for a higher value of the time constant can done the system lose the stability.

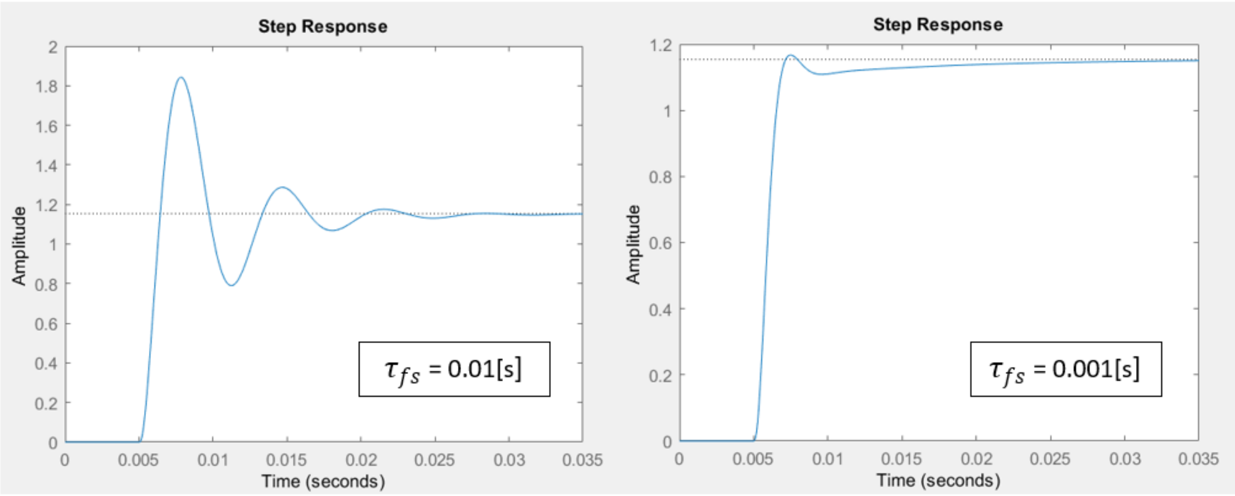


Figure 36 Influence of the sensor filter (Tool MATLAB)

- Influence of the derivative filter

The derivative part of the controller generates a control signal which proportionally depends on how the measurement signal changes. That means that the noise is amplified. In order to make this effect decrease, the derivative filter is used.

- Influence of the output filter

The output filter has been implemented to the code in order to have a softer controlled signal. The controller is sensitive to the noise, generating a controlled signal that can highly grow in a short time. In order to damped the control signal, which is the current that feeds the electromagnet, the output filter is used. The algorithm that describes the output filter is based on the same principle as the one described for the sensor filter and can be seen in the Equation (48).

$$FO = FO_{Old} + \frac{dt}{\tau_{fo}} (UNFO - FO_{Old}) \tag{48}$$

Where:

$$\begin{cases} FO : \ Filtered\ Output\ Signal\ [A] \\ FO_{Old} : Previous\ Value\ of\ Filtered\ Output\ Signal\ [A] \\ dt : Loop\ Sample\ time\ [s] \\ \tau_{fo} : Time\ Constant\ of\ the\ Output\ Filter\ [s] \\ UNFO : Unfiltered\ Output\ Signal\ [A] \end{cases}$$

In the Figure 37 can be seen that the influence of the output filter has not a great effect in the response. If an output filter is used, the response is slightly faster as it can be seen in Figure 37.

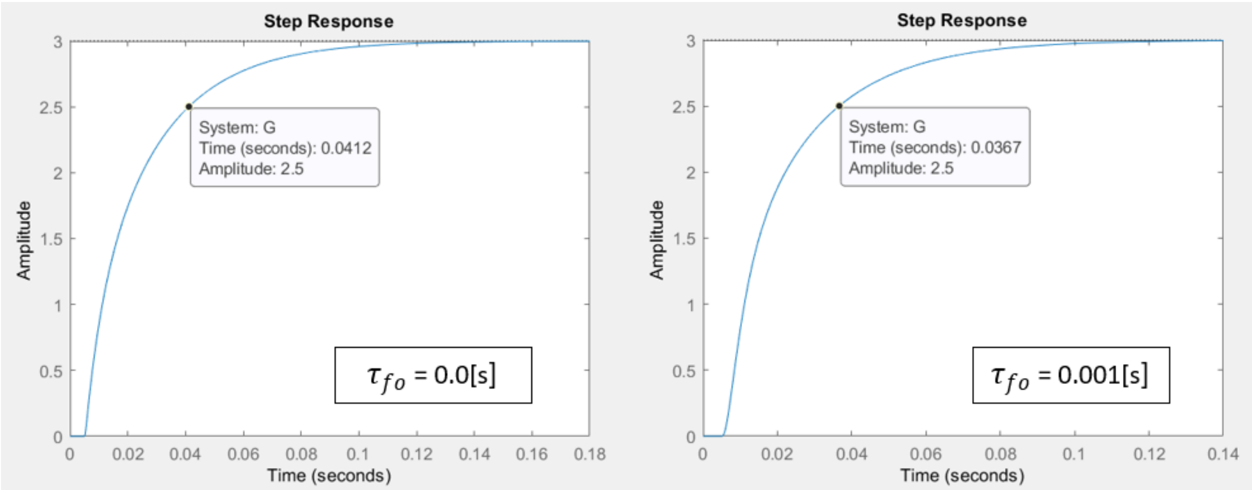


Figure 37 Influence of the Output Filter in the Response (Tool MATLAB)

1.7 Implementation of the Control System in Arduino

The code that represents the control system in Arduino can be seen in the appendix Arduino Code. The code language is C++. In the code are comments in the right side that explain the meaning of the code or the assigned variables.

First of all, the variables and the constants that the program will use are defined. Also, the analogic and the digital pins are defined.

In the “void setup” part of the code, the operations that are going to execute once are defined. The brakes and the direction of the Arduino Motor Shield are defined. Also, the 12 volts constant feeding that the LED emitter and the amplifier require are set. A serial communication [35] is initiated. This will allow to check the measurements of the sensor and the current that is flowing in the two channels of the Arduino Motor Shield.

In the “void loop” the operations that are constantly repeating are written. This includes the measurement of the analogic read of the sensor. Then the measurement is filtered. The filter uses part of the previous measurements to calculate the filtered measurement. With the filtered value the control signal is calculated based on the PD controller (1.5). Finally, the measurement of the sensor, the desired set point and the current flowing in the electromagnet are shown with the serial communication.

1.8 Results and Conclusions

The main goals of the project have been defined in the Abstract of this document. Those goals could be sumarized in were two:

- Developement of a phisical prototype where a controlled magnetic levitation system is represented.
- Study and document the magnetic levitation system that has been created based on a previous simulations. This involves the study of the magnetic systems in order to create a model of the system and simulate its behaviour.

Both goals could be considered as achieved. It has been possible to build a prototype where the sphere was levitated under controlled parameters. On the other hand, the levitating system is based on the theory described in this document about magnetic fields and the simulation tools have helped the tuning of the controller.

It has been also proved, that the formulation found in the references about the magnetic constant (Equation (11)), does not represent accurately the magnetic levitation system.

It also has also to be underlined, that the when a setpoint is introduced in the code of the control system, it can not be changed unless the levitation phenomena is stopped. That means, that the distance between the ball and the electromagnet can change due to disturbances in the light of the room where the experiment is taking part. These disturbances can make the ball either fall or lift until the electromagnet.

References

[1] M. Fowler, U Va Physics, [Online]. Available: http://galileoandeinstein.physics.virginia.edu/more_stuff/E&M_Hist.html.

[2] APS NEWS, [Online]. Available: https://www.aps.org/publications/apsnews/200807/physicshistory.cfm.

[3] lumen learning, “https://courses.lumenlearning.com/boundless-physics/chapter/magnetism-and-magnetic-fields/,” Lumen Learnig, [Online]. Available: https://courses.lumenlearning.com/boundless-physics/chapter/magnetism-and-magnetic-fields/.

[4] Los Alamos National Laboratory, [Online]. Available: https://www.pinterest.de/pin/63261569737776585/?nic=1.

[5] VIRTUAL VEHICLE, [Online]. Available: https://www.youtube.com/watch?v=wumR-1SSQ2s.

[6] S. H. K. a. G. Thomas Bächle, “Science Direct,” September 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0967066113000828.

[7] The Economic Times, 26 September 2006. [Online]. Available: https://economictimes.indiatimes.com/news/international/japan-to-invest-3-1bn-to-develop-maglev-trains/articleshow/2027096.cms.

[8] The University of Jordan, [Online]. Available: http://engineering.ju.edu.jo/Laboratories/04%20-%20First%20Order%20System.pdf.

[9] D. Herres, TEST & MEASUREMENT, 03 May 2017. [Online]. Available: https://www.testandmeasurementtips.com/how-to-measure-inductance/.

[10] H. H. a. J. R. M. E. D. Woodson, Electromechanical Dynamics, Massachusetts Institute of Technology: MIT OpenCourseWare.

[11] S. F. S. Ferreir, “Electromagnetic study of a variable inductor controlled by a DC current,” Universidada de Coimbra, January of 2016, pp. 14-20.

[12] “lumenlearning.com,” LumenCandela, [Online]. Available: https://courses.lumenlearning.com/boundless-physics/chapter/magnetic-fields-and-maxwell-revisited/.

[13] Z. Z. a. X. Li, “Real-Time Adaptive Control of a Magnetic Levitation System with a Large Range of Load Disturbance,” *Sensors,* May 2018.

[14] A. C. a. L. H. Zhangab Sue, “Nonlinear analysis of a maglev system with time-delayed feedback control,” in *Physica D: Nonlinear Phenomena*, 15 October 2011, pp. 1761-1770.

[15] Electrical4U, “Electrical4U,” 2019. [Online]. Available: https://www.electrical4u.com/ferromagnetic-materials/.

[16] Uniersity of Colorado Denver, Uniersity of Colorado Denver, [Online]. Available: http://www.math.ucdenver.edu/~esulliva/Calculus3/Taylor.pdf.

[17] UZZORS2K, Creative Commons, 28 July 2009. [Online]. Available: http://uzzors2k.4hv.org/index.php?page=magneticlevitation.

[18] “CUL 100-0.68 Datasheet,” [Online]. Available: https://cdn-reichelt.de/documents/datenblatt/D800/KATALOGDATENBLATT_CUL.pdf.

[19] “A3141 Datasheet,” [Online]. Available: https://www.elecrow.com/download/A3141-2-3-4-Datasheet.pdf.

[20] “Arduino,” [Online]. Available: https://www.arduino.cc/.

[21] “Auduino Motor Shield Rev3,” [Online]. Available: https://store.arduino.cc/arduino-motor-shield-rev3.

[22] “208-8 4R7 Datasheet,” [Online]. Available: https://www.uni-kl.de/elektronik-lager/423330.

[23] P. Dohrman, “Sciencing,” 17 April 2018. [Online]. Available: https://sciencing.com/three-ways-make-electromagnet-stronger-5498690.html.

[24] «Sol3N Datasheet,» [En línea]. Available: https://www.uni-kl.de/elektronik-lager/423330.

[25] “Electronics Tutorials,” [Online]. Available: https://www.electronics-tutorials.ws/opamp/opamp_1.html.

[26] «OP 27 Datasheet,» [En línea]. Available: https://www.uni-kl.de/elektronik-lager/416504.

[27] “Wikipedia,” September 2009. [Online]. Available: https://en.wikipedia.org/wiki/Operational_amplifier#/media/File:Operational_amplifier_noninverting.svg.

[28] “Arduino Mega datasheet,” [Online]. Available: https://store.arduino.cc/mega-2560-r3.

[29] “Arduino Uno,” [Online]. Available: http://arduino.cl/arduino-uno/.

[30] H. Föll, “Uni Kiel,” [Online]. Available: https://www.tf.uni-kiel.de/matwis/amat/iss/kap_4/illustr/s4_4_3.html.

[31] MathWorks, [Online]. Available: https://www.mathworks.com/help/simulink/.

[32] MathWorks, “Simulink Block Libraries,” [Online]. Available: https://www.mathworks.com/help/simulink/block-libraries.html.

[33] E. J. Routh, A Treatise on the Stability of a Given State of Motion: Particularly Steady Motion, Macmillan, 1877.

[34] J. A. Torres, “Wikipedia,” 21 May 2007. [Online]. Available: https://es.wikipedia.org/wiki/Teorema_de_Routh-Hurwitz#/media/Archivo:Criterio_de_Routh_01.JPG.

[35] Arduino, “Serial Communication,” [Online]. Available: https://www.arduino.cc/reference/en/language/functions/communication/serial/.

Appendix

A1 Revolution Counter Code

Code 1 Revolution Counter Code

```
int switchPin =5; // choose the input pin
int val = 0; // variable for reading the pin status
int counter = 0;
int currentState = 0;
int previousState = 0;

void setup() {
//pinMode(ledPin, OUTPUT); // declare LED as output
pinMode(switchPin, INPUT); // declare pushbutton as input
Serial.begin(9600);
}

void loop(){
val = digitalRead(switchPin); // read input value
// Serial.println(val);
if (val == HIGH) { // check if the input is HIGH (button released)
//digitalWrite(ledPin, HIGH); // turn LED on
currentState = 1;
}
else {
//digitalWrite(ledPin, LOW); // turn LED off
currentState = 0;
}
if(currentState != previousState){
if(currentState == 1){
```

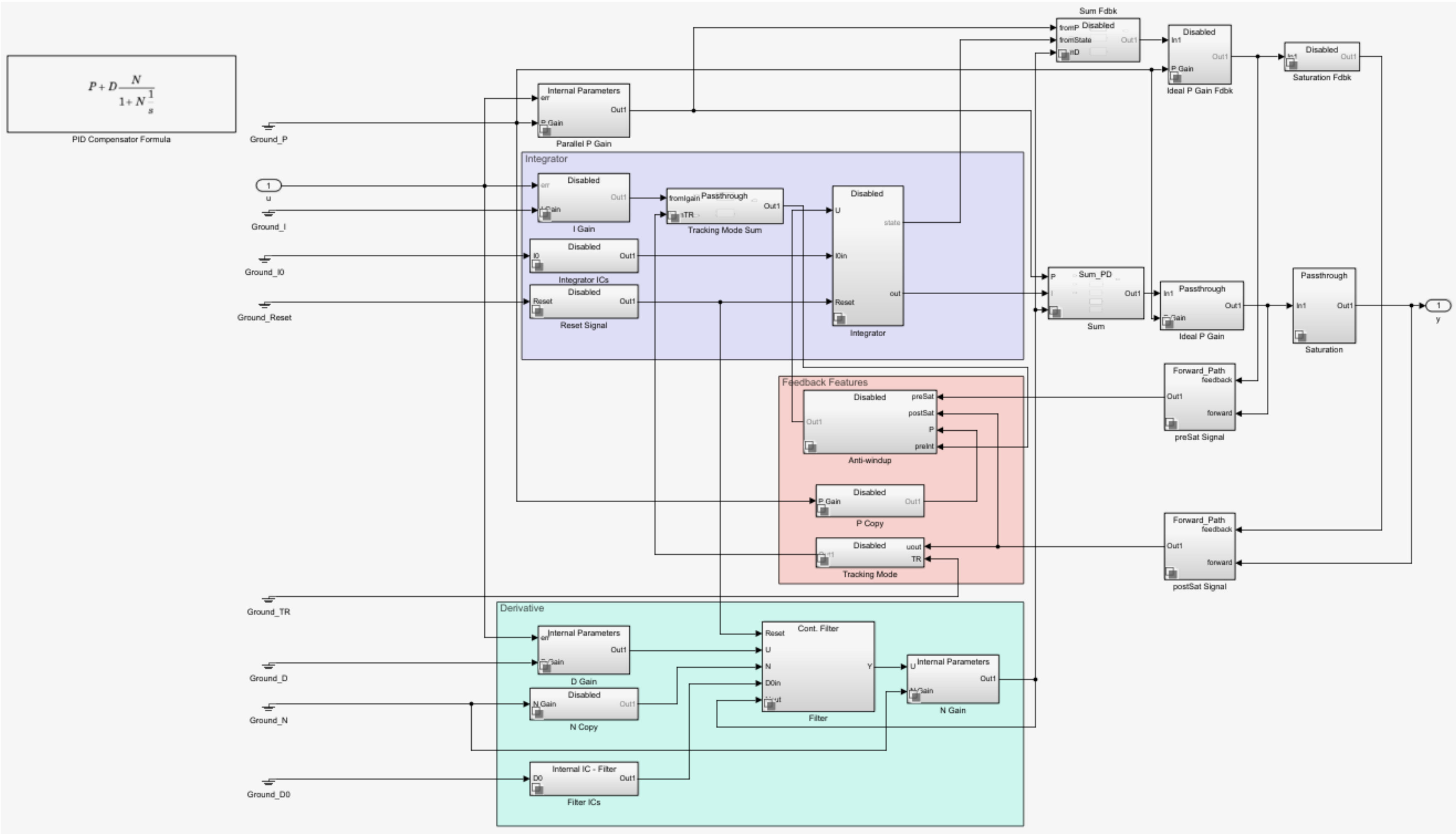
```
counter = counter + 1;
Serial.println(counter);
}
}
previousState = currentState;
delay(5);
```

A2 Magnetic Force Code

Code 2 Magentic Force Code

```
function f = magnetic_force(i, x, km, fMax)
% calculates the magnetic force
f = kc*(i^2) / (x^2);
% saturation
f = min(f, fMax);
end
```

A3 PD Controller



A4 MATLAB Code

```
clear all;

g = 9.81;% Graviationskonstante in m/s^2
Fg = 0.1; % Gewichtskraft Kugel in Newton
m = Fg/g; % Masse der Kugel in kg

L = 0.003; % Spuleninduktivität in H
R = 2.2 + 4.7; % Spulenwiderstand in Ohm

km = 7.5 * 1e-4;

controllerDelay = 180/0.002;

%pwmToAmpere = 0,0062745098;
%meterToAnalogRead = 6.5027; % 10 cm max

pwmToAmpere = 2/255;
meterToAnalogRead = 1024 / 0.05;

scaleErrorForControl = 1 / 1024;
descaleErrorForControl = 255 / 1;

xMax = 0.05;
xMin = 0;
% xMax = inf;
% xMin = -inf;

sensorMax = 1024;
sensorMin = 0;
% sensorMax = inf;
```

```
% sensorMin = -inf;

tFilter = 0.005;

pwmMax = 255;
pwmMin = 0;

x0 = 0.003;
v0 = 0;
i0 = 0.25;

xRef = 0.002; % Referenzposition in Meter
iRef = xRef * sqrt(Fg/km); %Zugehöiger Referenzstrom in Ampere

ki = 2*km*iRef/xRef^2;
kx = -2*km*iRef^2/xRef^3;

xRandVariance = 0.001 * xRef;

xRefAfterStep = 0.006;

t_start = 0;
t_end = 1;

pwm_sample_time = 1/490; % s

% dt = pwm_sample_time / 255;
dt = pwm_sample_time;

time_cont_vec = t_start:dt:t_end;

t_end = time_cont_vec(end);
nt = numel(time_cont_vec);
```

```
fMax = 2;
```

A5 Arduino Code

a. Magnetic Levitation

```
#include "ControlProgramCycle.h"


const long CYCLE_TIME_FILTER  = 200;              // cycle time of programm
[micro-seconds]
const long CYCLE_TIME_RUN     = 1000;             // cycle time of programm
[micro-seconds]
const long CYCLE_TIME_DEBUG   = 100000;           // cycle time of programm
[micro-seconds]



int digPin = 3; // PWM pin
int anaPin = A2; // Sensor pin

double anaVal = 0;   // Analogic Valie
double unfilteredVal = 0;   // Analogic Valie
double digVal = 0;   // Digital Value
double output = 0;   // Digital Value for PWM after saturation
  int setpoint = 110;  // bigger value, less power, more distance magnet ball.
Just change this
double derivative = 0.;
double error = 0.;
double previous_error = 0.;
double dt = 0.;
  double Kp = 2;
  double Td = 0.04;
  double Tf_o = 0.001;  // output filter in seconds
double Tf_d = 0.0; // derivative filter in seconds
double Tf_s = 0.000001; // solar cell filter in seconds

ControlProgramCycle     ControlProgramCycle_Debug( CYCLE_TIME_DEBUG );
```

```
ControlProgramCycle     ControlProgramCycle_Run( CYCLE_TIME_RUN );
ControlProgramCycle     ControlProgramCycle_Filter( CYCLE_TIME_FILTER );

//--------------------------------------------------------|
void setup()
{
  //Setup Channel A
  pinMode(12, OUTPUT); //Initiates Motor Channel A pin
  pinMode(9, OUTPUT); //Initiates Brake Channel A pin

  //Setup Channel B
  pinMode(13, OUTPUT); //Initiates Motor Channel B pin
  pinMode(8, OUTPUT); //Initiates Brake Channel B pin

   Serial.begin(57600);
   pinMode(digPin, OUTPUT);

   delay(500);

     //Chanel A brake and direction
  digitalWrite(12, HIGH); //Establishes forward direction of Channel A
  digitalWrite(9, LOW);   //Disengage the Brake for Channel A

  //Chanel B brake and direction
  digitalWrite(13, HIGH); //Establishes forward direction of Channel B
  digitalWrite(8, LOW);   //Disengage the Brake for Channel B
  // Lamp activation
  analogWrite(11,255);   //12 V for the LED and sensor


   Serial.println("Started.");

   dt = CYCLE_TIME_RUN *1e-6; // convert from micro-seconds to seconds
```

```
   if (Tf_s < dt) Tf_s=dt; // make sure filter is ok
   if (Tf_o < dt) Tf_o=dt; // make sure filter is ok

}
//-----------------------------------------------------------|
void loop() // PID
{

/*
  if ( ControlProgramCycle_Filter.Activate() ) {

   // read the sensor
   unfilteredVal = analogRead(anaPin);

   // filter the sensor
   anaVal = (1 - dt/Tf_s) * anaVal + dt/Tf_s* unfilteredVal;

  }

  /*
   *
   */
  if ( ControlProgramCycle_Run.Activate() ) {



   // read the sensor
   unfilteredVal = analogRead(anaPin);

   // filter the sensor
   // anaVal = (1 - dt/Tf_s) * anaVal + dt/Tf_s* unfilteredVal;
    anaVal = unfilteredVal;
```

```
   // PD calculations
   error = -setpoint + anaVal;
   derivative = (error - previous_error);
   digVal  =  Kp*(error  +   Td*derivative/(  dt  +  Tf_d  *  derivative));
//integralpart shoul be cancelled
   previous_error = error;

      // filter the output
      output = digVal;
    // output = (1 - dt/Tf_o) * output + dt/Tf_o* digVal;
   //output=output+50;
   // Check the value for levitation point;
    // output = digVal;
    if (output < 0) output=0;
    if (output > 255) output=255;

   analogWrite(digPin, output);

  }

  // print debug messages
  if ( ControlProgramCycle_Debug.Activate() ) {
       // Show the Sensor Value;
   Serial.print("anaVal=[");
   Serial.print(anaVal);
   Serial.print("]-");
   // Show the Error;
   Serial.print("error=[");
   Serial.print(error);
   Serial.print("]-");
   // Show the Electromagnet;
```

```
    Serial.print("output=[");
    Serial.print(output);
    Serial.println("]");
  }
}
```

b. Control Program cycle .ccp

```
/*
  ControlProgrammCycle.cpp
  Created by Jonas Weigand, 30.08.2019
*/

#include "Arduino.h"
#include "ControlProgramCycle.h"


    ControlProgramCycle::ControlProgramCycle(long cycle_time )
    {
      // save initial constants
      _cycle_time = cycle_time; // [micro-seconds]

      // initialise internal states (the ones required)
      _previous_mircos = 0;
      CYCLE_ENABLE    = true;
    }

    bool ControlProgramCycle::Activate() {

      // get current time
      _current_micros = micros();  // [micro-seconds]

      if ((_current_micros - _previous_mircos >= _cycle_time) &&
CYCLE_ENABLE) {
```

```
        _previous_mircos = _current_micros;    // Remember the time of the last cycle
        return true;
      } else
      {
        return false;
      }
    }
```

c. Control Program cycle

```
/*
  ControlProgrammCycle.h
  Created by Jonas Weigand, 30.08.2019
*/

// #ifndef ControlProgramCycle_h
// #define ControlProgramCycle_h

#include "Arduino.h"


class ControlProgramCycle
{

  public:

    ControlProgramCycle(long cycle_time );

    bool Activate();

    bool CYCLE_ENABLE;
```

```
  private:

    // Class Member Variables
    long _cycle_time;  // [micro-seconds]
    long _previous_mircos;  // [micro-seconds]
    long _current_micros;  // [micro-seconds]

};

// #endif
```

A6 Transfer Function Code

```
%% Simulate Levitating Ball
% Jonas Weigand, 16.09.19

clear all;

%% Define Filter and Time Parameters
Tfs = 0.0000001; % Time constant of the sensor filter, [s]
Tfd = 0.0000001; % Time constant of the derivative filter, [s]
Tfo = 0.0000001;   % Time constant of the output filter, [s]

Tdelay = 0.005; % Delay Time needed by the microcontroller, [s]

%% In order to use a transfer function, linearise the model

%  define a working point
y0 = 0.003;  % distance in [m]
i0 = 1;      % current in [A]

% m*g = km * i0^2/y0^2 = kx*y0 + ki*i0
% kx = -2*km * i0^2 / y0^3
% ki = km*i0/y0^2
km = 2e-6;
% derive ki and ky using linearisation
ki = km*i0/y0^2;
kx = -2*km * i0^2 / y0^3;



%% Define model and control parameters
% Note that the distance y is defined in [m].
% The controller output is defined in [A].
```

```
% In the Arduino Code, both units are different and must be converted!

P = -1000;
Td = 0.01;
D = P*Td;


%% Define model and control parameters
L = 0.002;  % inductivity of the coil, [H]
R = 7.2;    % resistance of the coil, [ohm]
m = 0.01;   % mass of the ball, [kg]
km = 2e-6;  % magnetic constant, [N*m^2/A^2]



%% Set up the transfer function
s = tf('s');


G = ( (Tfs*s + 1)*(-s*ki*D/(Tfd*s + 1) - ki*P) )/ ...
    ( (L/R*s + 1)*(Tfs*s + 1)*(Tfo*s + 1)*(m*s^2 +kx) - ki*(P + D*s/(Tfd*s +
1)) ) *...
    exp(-Tdelay*s);



%% Visualise and Print

poles           = pole(G) %#ok
is_stable       = isstable(G) %#ok

step(G)
```