Czech Technical University

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ

Stabilized driving platform

Stabilizovaná pojízdná plošina

Bachelor’s Thesis (2023)

Author of work:	Supervisor Thesis:
David Mateo Dubrovsky	doc. Ing. Martin Novak ph.D.

Bachelor’s Program: Information and Automation Technology

Declaration

I declare that I have prepared this work independently using literary sources and information that I cite and list in the list of used literature and sources.

Date: ………………………………… Signature: ……………………………….

Abstract

This Bachelor thesis is to design, construct, control and test a self-stabilizing platform. For this purpose research into stabilizing platforms was made, moving structure was designed and constructed, a control algorithm was created. Also, the kinematics of the structure were calculated to verify moving capabilities and limits of the platform. The platform was then experimentally tested and measured.

The theorical part of the bachelor’s thesis is focused on analysing existing models of similar platform projects and analysing the control algorithms of the platform

The practical part of the bachelor’s thesis is focused on the platform holding structure design and construction. Its required electronic components are carefully chosen, and the studied control algorithm is programmed.

Keywords: balancing structure, IMU sensor

Contents

DECLARATION **2**

ABSTRACT **3**

CONTENTS **4**

LIST OF MARKS AND SYMBOLS USED **6**

LIST OF USED SOFTWARE **6**

LIST OF TABLES **7**

LIST OF FIGURES **7**

1 INTRODUCTION **8**

1.1 BACKGROUND 8
1.2 PURPOSE 8
1.3 METHOD 9
1.4 INSPIRING PROJECTS 9
1.4.1 Self-balancing platform on a mobile car: by Bushra Amer Tawfeeq 10
1.4.2 Self-Balancing Monopod: By International Institute of Information Technology 14

2 DESIGN **18**

2.1 MECHANICAL DESIGN 18
2.1.1 Structure components 24
1. Main Support (1) 24
2. 1st Bearing supports (2) 25
3. Side Support of middle platform (3) 25
4. Mid-Level Platform (4) 26
5. 2nd Bearing Support (5) 26
6. Side Support of upper platform (6) + Top Platform (7) 26

2.2 ELECTRICAL COMPONENTS 26
2.2.1 MPU 6050 28
1. Gyroscope 28
2. Accelerometer 28
3. Digital Motion Processor (DMP) 29

2.2.2 Servomotors (MG 995) 29
2.2.3 Arduino MEGA 2560 30
2.3 CONTROL DESIGN 30

2.3.1 Flowchart *31*
2.3.2 Coding *32*

3 RESULTS AND CONCLUSION **36**

3.1 FINAL DESIGN RESULT 36
3.2 TESTS RESULTS........ 37

4 CONCLUSION **43**

REFERENCES **44**

List of marks and symbols used

T_r	Torque Required
L_{CM}	Distance to Centre of Mass
W_{MS}	Weight of Movable Structure
W_{3D_MS}	Weight of 3D printed parts of Movable Structure
W_B	Weight of Bearings
W_R	Weight of Retaining rings
W_S	Weight of Servomotor MG 955
W_{BN}	Weight of Bolts and Nuts
DMP	Digital Motion Processor
$T_{Stabilizing}$	Time for Platform to Stabilize
T_{Delay}	Delay time in program for servomotors
θ_{bal}	Angle at which the servo stabilized
θ_O	Initial servo angle

List of used Software

- Fusion 360
- Microsoft Word
- Arduino IDE 2.0.2
- MATLAB (MATrix LABoratory)
- Creality Slicer 4.8.2
- Microsoft PowerPoint
- Jo.Drawio

List of Tables

TABLE 1. MG 995 FEATURES AND SPECIFICATIONS [7] 21
TABLE 2. 2125MG FEATURES AND SPECIFICATIONS [11] 23

List of figures

FIGURE 1. CAD DRAWING BY BUSHRA AMER TAWFEEQ [4] 10
FIGURE 2. SCHEMATIC DIAGRAM BY BUSHRA AMER TAWFEEQ [4] 12
FIGURE 3. SYSTEM FLOWCHART [4] 13
FIGURE 4. DESIGN OF PROPOSED PROJECT BY THE INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY, NAYA RAIPUR [5] 15
FIGURE 5. LOADS ON A SERVO MOTOR [6] 16
FIGURE 6. BLOCK DIAGRAM [5] 17
FIGURE 7. STRUCTURE 3D MODEL FULL VIEW 18
FIGURE 8. FORCES REACTIONS FROM SUPPORTS 19
FIGURE 9. STRUCTURE WITH ROLL AND PITCH AT ROUGHLY 60° ANGLE 20
FIGURE 10. LABELLED 3D MODEL 24
FIGURE 11. SCHEMATIC DIAGRAM OF ELECTRICAL CIRCUIT...27
FIGURE 12. EXAMPLES OF PMW PULSES TO CONTROL A SERVOMOTOR [15] 30
FIGURE 13. SYSTEM FLOWCHART..31
FIGURE 14. DESIGNED SELF-STABILIZING PLATFORM...36
FIGURE 15. PLATFORM ON BASE AT 30 DEGREES FOR MPU 6050 ROLL ANGLE..........................37
FIGURE 16. PLATFORM ON BASE AT 30 DEGREES FOR MPU 6050 PITCH ANGLE.........................38
FIGURE 17. MPU 6050 NOISE...39
FIGURE 18. ROLL ANGLE REACTION...40
FIGURE 19. PITCH ANGLE REACTION..41

1 Introduction

1.1 Background

Today many industries are dependent on stabilizing mechanisms. The technique is applied in everything from segways to self-stabilizing cameras [1], helicopters [2], noise reducing equipment and even in medical devices when performing precise surgery is needed. Another possible application for a stabilizing mechanism is the example we are going to focus on which is a stabilizing platform, which can be used in equipment which will self-level as an anti-motion sickness chair. [3]

This last-mentioned application has as purpose to compensate for the tilt angle that a platform may be submitted to. In order to accomplish that one a designed mechanism will read the angle of the platform by means of different possible sensors such as potentiometers, accelerometer, or gyroscopes, then compensate that angle with actuators as servo motors, DC-motors or even stepper motors. [3]

1.2 Purpose

The purpose of this bachelor's thesis is to design a mechanism that will self-balance a platform. To do so, we have as purpose to analyse how the feedback information data can be used along with the IMU sensor (composed of a 3-axis accelerometer and a 3-axis gyroscope). To accomplish that goal an Arduino platform will be used to analyse the feedback control information and accordingly activate the actuators (servo motors in our case) so the maintain a stabilised platform position.

The bachelor’s thesis questions to be answered were the following:

- How accurately did the designed mechanism stabilised the platform?
- When placed on a moving object, how fast could the object move without the mechanism having trouble stabilising the platform?

1.3 Method

After analysing multiple mechanical existing and working models, a personal design was made studying the known limitations of the model and demands for it. The model was the conceived and the control for the platform was then developed.

Once all the development has been achieved a test system had to be built in order to put to test the designed self-stabilised platform. The data obtained from the different test systems is then analysed using the software MATLAB.

1.4 Inspiring Projects

In order to begin designing the stabilising mechanism which will support the platform I will proceed to analyse other already existing projects which may have common factors with my future design and inspire me that way on how to approach more adequately the mechanism’s design.

This part is crucial as it will give me a better idea of what is and what is not possible to make and the reasons behind them, as if I were to design it without taking previous designs into account, I might encounter designing issues which could have been avoided or even realizing half-way another method of making the mechanism as there is a wide range of choices for mechanisms options.

In this part I will then analyse individually the projects I read about which has helped me in any either positive or negative way explaining its pros and cons, how they have inspire me and the working mechanics of it. Also, when possible, I will talk about the materials used in each case.

1.4.1 Self-balancing platform on a mobile car: by Bushra Amer Tawfeeq

In this prototype, as mentioned in their abstract, I can affirm that their design was successful in terms of tilt angle compensation accuracy as the platform was stable and the platform was able to overcome that error in a time no longer than 4 seconds. [4].

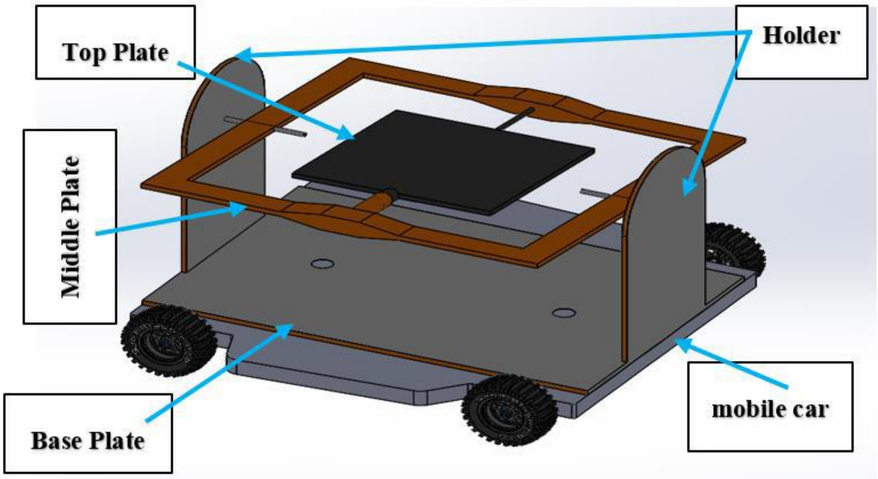

Figure 1. CAD drawing by Bushra Amer Tawfeeq [4]

In order to make the mechanical structure design they used the software SolidWorks to construct and design it. In the design they oriented themselves more for a double connected hollowed plate which are with their rotational axis on the same level as can be observed in *Figure 1*.

They opted for a design where two single holders popping out from a base plate will hold onto the middle plate. This said base plate will be the one which will be fixed to the moving car. Then the top plate will be attached to the middle plate the same way the middle one is attached to the holders. From what I can observe in the multiple pictures they have and the description of the design it seems that they opted for a bearing-less connection between plates and between plate and holder on the side opposite to the servo, instead they opted for a simple connection using a smooth thin metallic rod and a hole on the plate and holder. In my opinion I believe that using a bearing would have improved the performance conditions of the mechanism as we are talking about reducing a good amount of friction if we were to apply higher loads. Even though, it seems that they chose not to use a bearing as the platform is not intended to hold any amount of weight but just to prove the functionality of a simple working mechanism, which they performed correctly. Speaking about dimensions it’s also possible to appreciate that the surface area required for the structure to be against the surface of the platform is quite large which from my point of view that is not ideal. For example, in there the middle plate, which is the larger plate is of (40 cm length × 40 cm width), while the top plate, the stabilised platform, is of (20 cm length × 20 cm width) [4]. Meaning the ratio between middle and upper platform is of ratio 2 in this case.

Another design inconvenient I see in this case is that if you were to implement this design on a moving object which could go at steep surfaces it would mean that a new constraint would appear in the design, and this constraint is the volume of the object. The object would be constraint on size as if the middle plate would rotate to 90 degrees, the height limit on that specific scenario would be until the middle plate.

On the other hand, this specific design has some advantages whom according for what the platform is to be used might be more or less important. Therefore, the first advantage to highlight is that the structure is all relying on a large surface base attaches to the moving car. As the base is large it translates into an increase in stability for the structure compared to if we were to have a smaller base plate, and this increase on stability is mainly as the vibrations and fast movements from the car will affect less the structure. The second advantage, less important, is that the IMU sensor used is very easy to design it to be placed in the centre of the top platform and it will maintain an almost constant distance with where the sensor is connected so we are not expecting any chance that the cables will disconnect and then the platform stop working.

The four main components used in the design are the sensing unit (composed of an MPU 6050 sensor, 3-axis accelerometer + 3-axis gyroscope), the controlling unit (in this case an Arduino Mega 2560 R3 has been used), an actuator (referring to the two servo-motors, one moving the pitch angle and the other moving the roll angle) and the platform structure which for this case has been chosen to be made out of a “low weight Perspex GS cast acrylic glass” (PMMA) with a thickness of 2mm [4]. This choice of material seems in my opinion after analysing the structure not ideal for a design where the platform is required to hold any weight but again, as previously mentioned, it is excused in their case as they are not aiming this specific design to support any mass on top. It’s an effective material for their purpose plus the fact if you have available a CNC machine you can produce the parts pretty fast compared to other methods such as 3D printing.

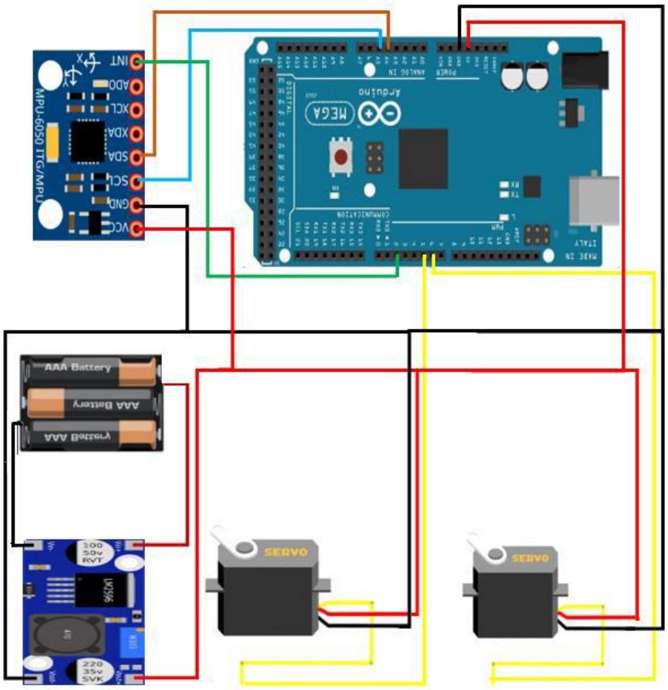

Figure 2. Schematic Diagram by Bushra Amer Tawfeeq [4]

We can appreciate in the schematic diagram they have made; the designer has used 3 × AA batteries which combined in series produced and output voltage of 7.4 Volts [4]. The user needs those 7.4V reduced to the required 5V that the system needs as an input to distribute across the servo motors, the Arduino Mega board and the MPU-6050 sensor, therefore the user chose to use a voltage regulator to accomplish that.

So how does the designed system function? Firstly the system is started which will then activate the powering of the system. Once active, the MPU 6050 sensor will take its first readings which will be sent to the Arduino board. This Arduino board will process the received information and separately for the x and y axis it will pass the values through a specific complementary filter then compare if the values are equal to 0 (which means no further system action is required) or if not, it will determine if the actuators need to move in clockwise or anticlockwise to stabilize the platform. Once the required movement is determined, it will send the necessary voltage to each servomotor in order to obtain the desired action. Straight away we will keep actioning the same procedure as a loop starting again with the MPU 6050 sensor readings and that way keep the platform stabilized in the required position. The whole process is described by the authors in a system flowchart which is the following:

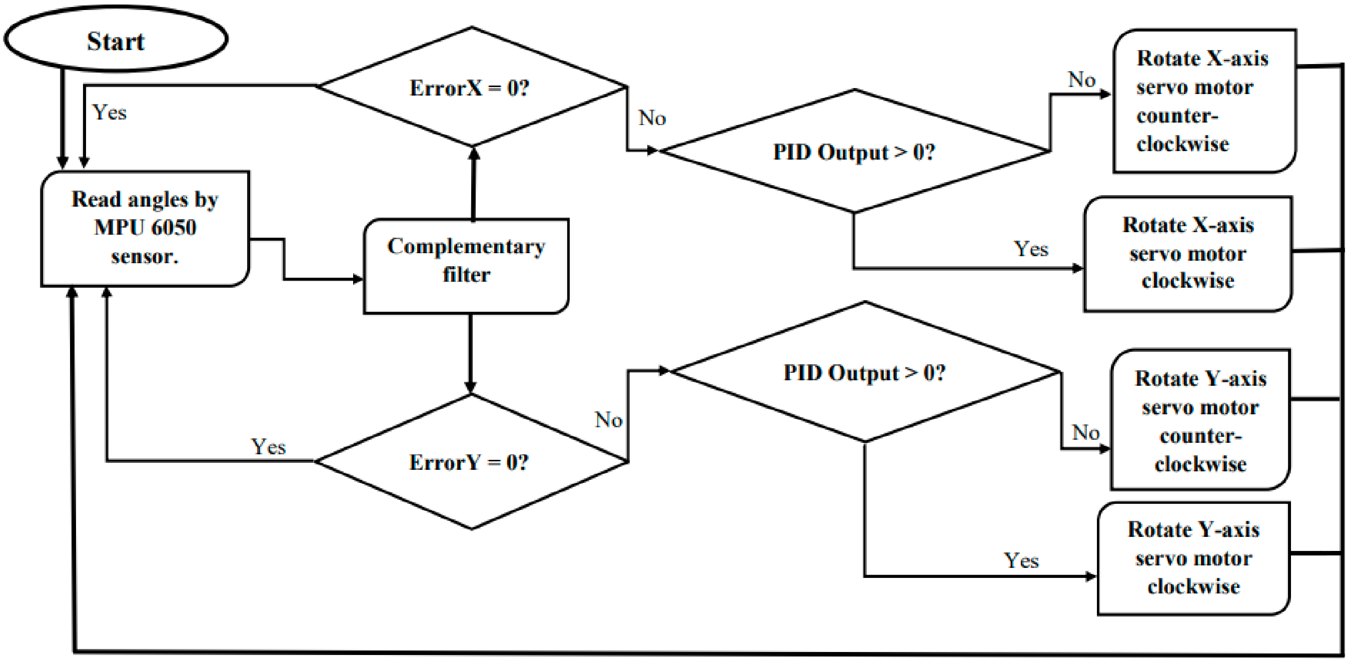

Figure 3. System Flowchart [4]

As they were using a DC-motor it was required to implement a PID controller, which was optimized with the generic algorithm (GA) [4]. They had to implement 2 different PID controllers as there are 2 different DC-motors with each one having slightly different (K_p , K_i , K_d) characteristics. If they were to use a servomotor instead it might not be needed the use of a PID controller unless a higher accuracy is needed, but because of the DC-motor, it is important to implement an adequate PID controller in order to obtain the best possible characteristics for the gain, rise time, settling time and percent overshoot.

In overall I believe the platform is a nice inspirating design from where I certainly will be able to abstract some knowledge and implement it in my own design. Even though because of the material choice I might have some constraints as the volume size of printed materials.

1.4.2 Self-Balancing Monopod: By International Institute of Information Technology

To analyse this project we will place a special focus on their mechanical design as the idea of how it is set influenced a lot my own final design.

The project is intended to have a deeper focus for the study about the control of the mechanism rather than the complexity of the mechanical design.

As we can observe in *Figure 4* the designing group opted for a monopod style mechanism where instead of having the servos aligned at the same height at the rest position, in this case we are observing that the servomotor controlling the pitch angle is on a platform above the platform where is the servomotor controlling the roll angle. In this case we are having even a third servomotor which is at the most bottom platform controlling the yaw angle. It is not specified why the chose to add that angle control, but it could be used for example to control the view orientation of any object which may be placed on top of the platform, as for example a camera, independently of what is the front view orientation of the moving device.

Figure 4. Design of proposed project by the International Institute of Information Technology, Naya Raipur [5]

Again, from *Figure 4*, we can observe that the connections made between the servomotors and also between the pitch angle controller servomotor and the top platform can lead to complications when programming the control of the platform as they have used metallic “L’s” which if some weight is applied on them, they will slightly bend cause the angle of the L not being 90 degrees anymore, leading to an inaccuracy into the stabilization of the platform as it would stop being parallel to the base. That is a problem as we can observe that they have not placed the IMU sensor on the platform but on the device’s, base moving the platform as mentioned in the report [5].

There is another visible problem on the design of this structure. It is important to mention that servo motors have a limited overhung and thrust load (check *Figure 5*) which can be applied, if this force is exceeded, it might lead to a servomotor failure by either breaking the output shaft or by damaging the internal ball bearings [6]. As we can observe, the overhang weight imposed on the servomotors is the one of all the structural weight from levels above. The servo motor which would suffer the most in this case would be the one controlling the roll angle. In order to solve that problem, it would have been ideal for the designer to add some kind of support with bearings for the connectors to transfer the forces onto the supports and that way make that the only forces that the servos would need to overcome are the inertial forces of the structure.

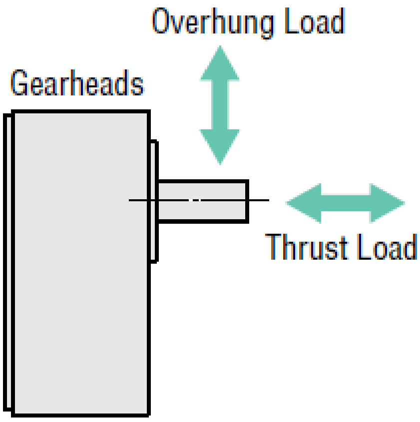

Figure 5. Loads on a servo motor [6]

The material selection of the manufacturer was quite simple as no parts were designed, but instead they chose existing parts made out of steel for the structure and a wooden square for the platform which seems like a heavy choice for the purpose.

Again, as in chapter 1.4.1, the board selection to control all the electronics has been the Arduino Mega and for the sensor the opted again for a 6-axis IMU sensor instead of a 9-axis IMU sensor. The IMU sensor is not specified but we know it is composed of ADXL345 Accelerometer and a ITG3200 Gyroscope. In this project it does not seems like there is any use of a voltage regulator in the circuit.

In order to get a more precise tilting angle data they have implemented a filter, but no further details were said. Then this data is controlled with a said PI controller, but instead I can see they have used a PID controller and not as marked in *Figure 6.* My assumption about them using a PID controller is due to the further calculations for the controller constants shown in their report.

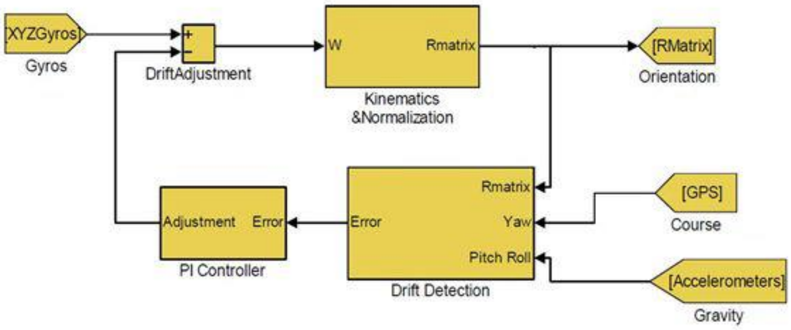

Figure 6. Block Diagram [5]

In general, taking apart the two main mentioned inconvenient in the design and the fact they are using 3 instead of 2 servomotors, I like the designed structure and believe with some improvements I could make a model similar to this one. Also it is very nicely explained all the calculations that has been made in order to implement a correct PID controller.

2 Design

2.1 Mechanical Design

After reviewing all the different options I came out with an idea of a design so the structure may correctly accomplish the task by being solid enough and by having an appropriate range of working angle.

Before I explain in what my design consists of, I think is important to mention the requirements set for the mechanical structure. As the structure is intended to be mounted on top of a moving vehicle, for example a car. It is not require for the platform to have a movement motion larger than around ± 60 degrees in each roll and pitch as the moving vehicle will very unlikely reach those position angles. Also the mounting base area shouldn't be too large as even though in real live you obviously can find vehicles which will dispose of a large mounting surface, the vehicle we are going to work with is not of such size so we should adapt.

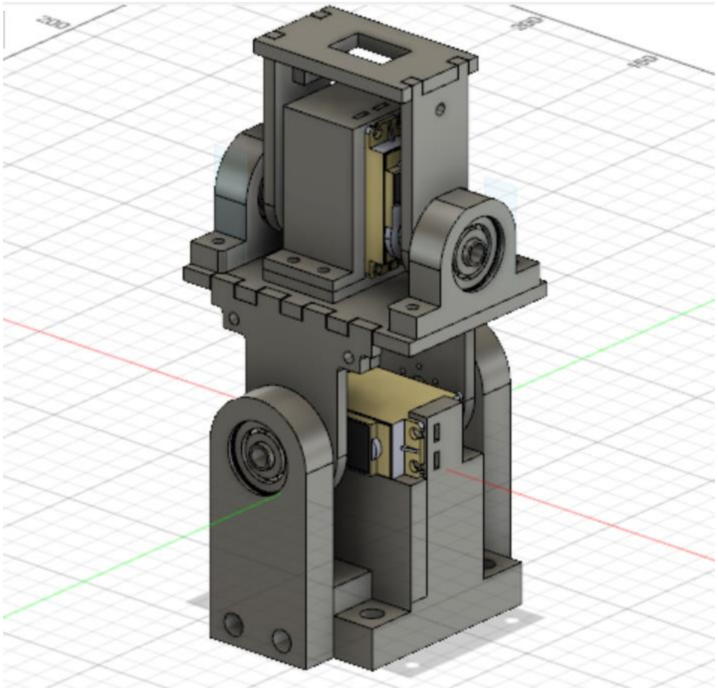

Figure 7. Structure 3D model full view

Therefore, after further analysis of the requirements I came out with a mechanical design such as the one in *Figure 7*. The structure is designed so that the mechanical mobility in both roll and pith is of ± 90 degrees, even though later on I will explain why in practise the mechanical movement will be limited to smaller ranges.

This structure has 2 different levels for the servo motors as in section 1.4.2 [5] so that way we can have a platform of considerable size without having to increase the size of the base as would happen in the example 1.4.1 [4]. This would be an advantage if we were to further design the model for carrying heavier loads.

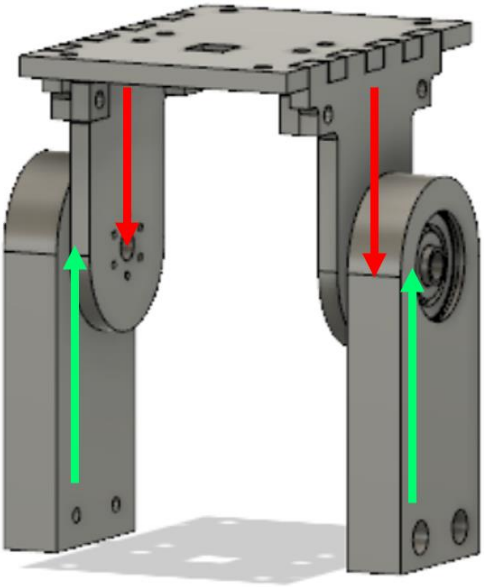

Figure 8. Forces reactions from supports

The holders for each level, are designed so that the platform levels are supported not only from the torque transmission side as in example 1.4.2 [5], but instead, they are been supported also from the opposite side as visible in *Figure 8* and in example 1.4.1 [5]. The reason behind why this support on both sides is so crucial is because if we were to remove the support in the opposite side, some torque would be generated by the platform weight, and this could cause to have the middle part of the platform bended a bit or even to break if not strong enough. This also that torque would generate a non-axial force on the output shaft of the leading servomotor and as I have already mentioned in part 1.4.2, the servomotors have a limit allowed of overhang load which commonly is quite low. Therefore, adding those supports on both sides will get rid of those two mentioned problems and therefore improve the dynamics of the system, accuracy of the system and also lifetime of the structure and servo motor.

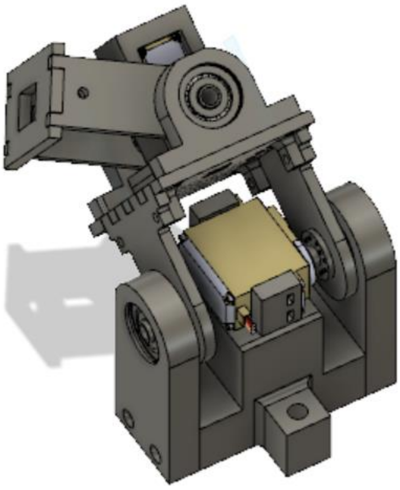

Figure 9. Structure with roll and pitch at roughly 60° angle

In *Figure 9* we can appreciate that the design would allow the requirement of motion where we are required to at least be able to move the platform to ± 60 degrees in the roll and pitch direction. This was a key factor to check before starting with the real-life manufacturing of the structure. In fact, technically speaking, it has been verified in the fusion app that the rotation of the platform is in fact larger than ± 90 degrees for each roll and pitch angles.

One crucial thing to check was that the weight of the upper part was not too large for the servomotor which had been selected (MG 995), whom specifications are shown in *Table 1*. Therefore I had to try to compress the structure as much as possible as the further away the centre of mass of the load is from the axis of rotation, the higher the amount of torque would be required to rotate the structure.

Also, in order to reduce the torque required I printed the upper parts with a lower density infill which makes the part a tiny bit lighter.

Operating Voltage:	4.8 – 7.2 Volts
Operating Temperature:	0 – 550° C
Operational Frequency:	50 Hz
Operating Motor Speed @ 4.8 V:	0.2 sec/60 degrees
Operating Motor Speed @ 6 V:	0.16 sec/60 degrees
Motor Stall torque @ 4.8 V:	8.5 kgf·cm
Motor Stall torque @ 4.8 V:	10 kgf·cm
Dead band width:	5 µs
Degree of Rotation:	180 degrees
Physical Dimensions:	40.7 x 19.7 x 42.9 mm approx.

Table 1. MG 995 Features and Specifications [7]

So after having the design done it was time to ensure that the mechanism would work properly. To verify it we were to find the rotational load on each servomotor and ensure that it is not exceeding the servomotor stall torque load. We know from the product catalogue that the servo we are using (MG 995) has a stall torque of 10kg/cm when operating at 6V. Another useful information is that the weight of the servomotor is of 55g [7]. Therefore as both servos used are going be the same, it is only needed to check the load for the lower servomotor, the one controlling the roll angle. The centre of mass of the servo's load has been obtained out of the used software "Fusion 360" and the weight of each individual printed component out of the slicer used "Creality Slicer 4.8.2". For the mas of the bearings and the retaining rings we got the weight out of their catalogue. [8] [9]. To calculate the torque required to move the upper structure we use the following formula:

$$T_r = L_{CM} \cdot W_{MS} \tag{1}$$

Where, the value of W_{MS} has been obtained as

$$W_{MS} = W_{3D\_MS} + 2 \cdot W_B + 2 \cdot W_R + W_S + W_{BN} \tag{2}$$

$$W_{MS} = 117 + 2 \cdot 20 + 2 \cdot 2 + 55 + 50 \tag{3}$$

$$W_{MS} = 266\ g \tag{4}$$

Therefore plugging *(4) into (1)* we get that the torque required to move the platform is equal to:

$$T_r = 7.120 \cdot 0.266 \tag{5}$$

$$T_r = 1.90\ kg \cdot cm \tag{6}$$

Knowing that the servomotor has a stall torque of 10kg/cm we can appreciate that the torque provided by the servomotor will be enough to move the structure and no change need to be done.

After testing the structure it was concluded that the MG995 servomotors have a little click which is not ideal for the roll angle control as the mass that is needed to carry represents too much and causes a “clicking” motion resulting on very big vibrations. Due to it, it has been chosen another servomotor which at testing had no sort of clicking and was ideal for the situation. The motor chosen was the 2125MG whom specifications are similar in speed and have a stall torque of 25 kgf· cm, which is 2.5 more than the MG 995 therefore it still satisfies the movement condition. The specifications can be appreciated in *table 2.*

Operating Voltage:	4.8-7.2V
Operational Frequency:	50-333Hz
Operating Motor Speed @ 4.8 V:	0.15 sec/60 degree
Operating Motor Speed @ 7.2 V:	0.13 sec/60 degree
Motor Stall torque @ 4.8 V:	21kg/cm
Motor Stall torque @ 7.2 V:	25kg/cm
Dead band width:	3 µs
Degree of Rotation:	180 degrees
Physical Dimensions:	40.7 x 19.7 x 42.9 mm approx.

Table 2. 2125MG Features and Specifications [11]

The chosen method of manufacturing the needed designed components is 3-D printing as it was an easy, cheap, accessible method where using the common material PLA it satisfied all my material properties requirements. Which are very basic ones as the structure is not intended to operate in any special climate, just at room temperature, also we are not looking for carrying a specific amount of weight on top of the platform so the only weight is the one from the structure itself whereas looking at the result in formula *(4)* the weight of the structure is definitely not enough to get damaged as the designed pieces have been designed to be printed at 5 mm at least which results in a solid enough component.

2.1.1 Structure components

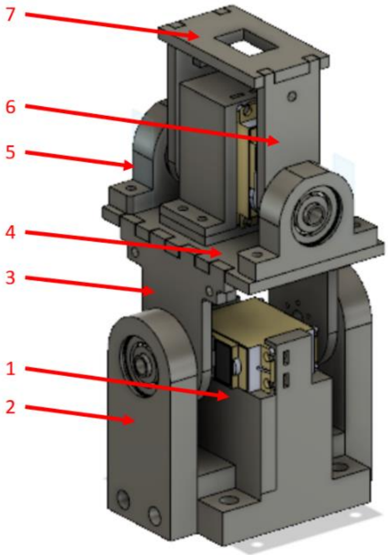

Figure 10. Labelled 3D model

1. *Main Support (1)*

The main support is intended to be the base of the structure, it would have the task of holding the structure to whichever object it is attached to. It is then crucial that the support is well rigid so it can hold tight the structure so no vibrations are caused on the structure which could lead to problems. The piece has been printed then with a higher internal density than usual as the strength of the component increased severely after hand testing on the thinner parts with 20% and 40% percent infill density. The expected tensile strength is 17.7 MPa for specimen having triangular infill pattern with 40% infill density and 0.10 mm layer height, therefore the piece was printed with those properties. [10] The component has 3 holes triangulated of size 6mm so it can fit a M5 bolt and have a more than enough strength holding to the base. The main support holds the servomotor controlling the pitch angle and the bearing supports *(2)* which are screwed into the sides of the main support by means of × 2 M3 bolts on each side.

2. 1st Bearing supports (2)

This part is doubled as two identical parts are intended to be placed on each side of the main support *(1)*. As the weight of said component is irrelevant to the motion of the platform, we can go to a higher density infill as the main support just to ensure it will not break as the component is intended to hold all the weight from the structure transferring it to the main support. It has then a hole where we are fitting the bearings which allow the rotation of the middle and upper structures, this whole was made of the exact bearing outer diameter and then sanded down until we can barely fit the bearing to ensure we have a tighter fitting, and the bearing won't have any sort of tangential movement. To restrict the bearing movement in the axial direction I added 2 slots on the components main hole, one on each side of the bearings place with a 0.5mm clearance to allow the placement of a retaining ring on each slot. This set up allow us to easy access the bearing if needed later on.

3. Side Support of middle platform (3)

Again this component is duplicated as used in both sides, can be appreciated on *Figure 10*. The use of this piece is that one of the sides of one of the components is attached to the servomotor so it can realise the torque rotational movement of the servo and therefore move the middle and upper platform on the pitch angle. The cylinder going in the bearing is intended to avoid having overhang forces on the servomotors shaft. The cylinder is hollowed to make possible the fastening of the bolt tightening the connecting wheel to the servomotor. It is possible to appreciate that a sort of teeth shape was designed so for transmission between the part *(3) and (4)* is distributed between all teeth and the two bolted connections from the upper part. From now on the density of the filling really matters as we are looking to cut in weight as much as possible therefore those components were printed at 20% infill density as we checked, and the expected tensile strength of a 20% infill density is enough as long as we do not overload the structure (which is not intended to happen)

4. Mid-Level Platform (4)

Easy to imagine from *Figure 10*, this component is mid-platform which is intended to move on the pitch angle only and therefore transmit this pitch rotation to the servomotor which is clamped to the platform. It is crucial that the whole made on the platform are not too close together, so it does not risk failing having a too small wall between wholes. This platform will also be carrying the other 2 bearing supports. Again the printing infill was made at 20% to keep a structural rigidity but still a low weight component.

5. 2nd Bearing Support (5)

Those bearing supports are intended to fix the upper bearings to the platform *(4)*, bearings which are being responsible of allowing the rotation of the top platform on the roll angle. Unlike the 1st bearing supports *(2)*, those ones are intended to keep the rotation of the servomotor as low as possible so that the centre of mass is kept closer to the pitch angle rotational axis and therefore reduce the rotational torque required to move the load.

6. Side Support of upper platform (6) + Top Platform (7)

Those two components have the same purpose as their respective one on the mid-level. The only apparent different is the size which was made smaller to reduce weight but kept at the same density infill and also that the top platform has been made with a square hole on the middle so we can place the chosen IMU sensor which will detect the inclination of the platform. The infill of the actual top platform was made of only 10% as there will be minimal forces acting on it and it is possible to save a slight quantity of mass this way.

2.2 Electrical Components

The second part of designing the self-balancing platform is doing an analysis of which electrical components are needed to accomplish the function requirement of the structure. As we could observe, not all already existing projects use the same electrical components, and this is due to the fact that some electrical components can do the same tasks as others but with a variation on the working capability. For example, it has been seen that some projects were using DC-motors while others have been using servomotors. Therefore, is crucial to take into consideration that the properties of each similar component will have some positive and negative effect on the application.

In the case of this project to perform the axial rotation on the pitch and roll angle it has been stablished that we would use the servomotor MG 995 which was the only one available in the lab. After introducing this servomotor proposition in the mechanical design it was determined that it could be used for the purpose.

The combination of components forming the electrical designed is 2 servomotors MG 995 (to control both roll and pitch angle), a 6-axis IMU sensor ("MPU 6050" to control the tilt of the platform) and an embedded board Arduino Mega 2560 (to control the servo's and IMU).

In *Figure 11* we can appreciate a schematic diagram of how the electrical components have been wired together.

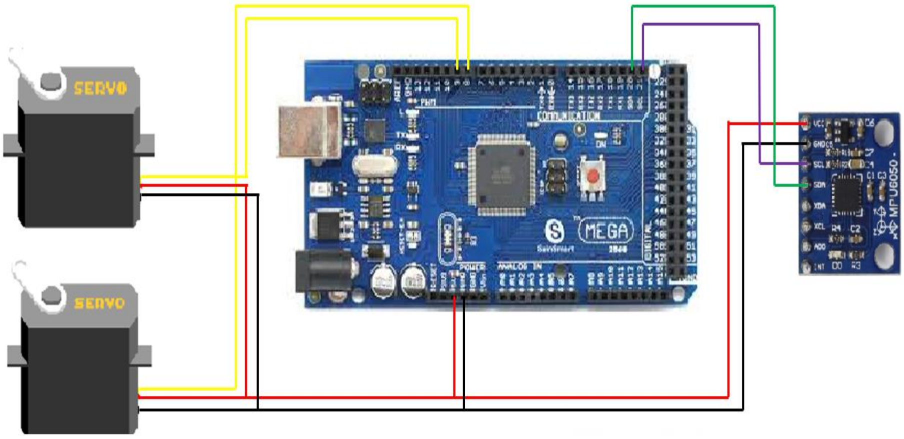

Figure 11. Schematic Diagram of Electrical Circuit

In the next few pages a deeper explanation on how does each component work, its different possible applications, and the specific application in the given project.

2.2.1 MPU 6050

The MPU-6050 is a device to track the motion of a measured object in 6 different ways. With a gyroscope that can measure how quickly a measured object is turning around three different axes (raw, pitch, yaw), it also has an accelerometer to measure how quickly the measured object is moving in three different directions (x, y and z-axes), and it also has a digital motion processor (DMP). The MPU-6050 also has a special processor that helps it understand and use the information it gets from the gyroscope and accelerometer. It can send and receive information using either I2C or SPI, which are two different ways that devices can communicate with each other using wires and in the case of this project the I2C communication is used. [11].

1. Gyroscope

A gyroscope measures angular velocity, the rate of change of orientation. It uses MEMS sensors which detect movement and output a signal proportional to the angular velocity. It can be used to determine the orientation of the device and track and control movement, for example in a robot to help maintain balance and control movements. [11] [12]

2. Accelerometer

An accelerometer in the MPU 6050 measures acceleration, the rate of change of velocity, by detecting movement of a mass on a spring using MEMS sensors. These sensors output a signal proportional to the acceleration being experienced. Accelerometers are commonly used to measure and detect acceleration, movement, orientation, and inclination in devices such as smartphones, fitness trackers, robots and drones. They can also measure gravitational force.. [11] [13]

3. *Digital Motion Processor (DMP)*

The MPU6050's digital motion processor (DMP) processes data from the gyroscope and accelerometer to provide accurate orientation and motion information. It uses complex algorithms to filter and determine the position and orientation of the object. It can also detect and follow motion in real-time, making it useful in applications such as robotics and virtual reality systems. It is an important component of the MPU6050, providing accurate and reliable information about the object's motion and orientation. [14]

After understanding the working principle of the MPU 6050 it is fair to precise that the sensor will be used to control accurately in real time the orientation and motion of the top platform. As the sensor is going to use the I2C communication method, we need to be aware that the wiring shouldn’t be too long as larger errors could be appearing. Therefore, this is why it has been checked once mounted that the sensor worked sufficiently well with those wire length conditions.

2.2.2 Servomotors (MG 995)

A servomotor is a type of actuator that allows precise control of angular position by rotating an output shaft. It consists of a motor, a reduction gear train, and a feedback sensor (usually a potentiometer). Servomotors usually have limited angle mobility but sometimes have continuous rotation. The controller of a servomotor system receives a pulse-width modulated signal specifying the angular position of the output shaft, compares it to the actual position using the feedback sensor and applies a correction signal to the motor to reach the desired position. PWM is a method of controlling power to an electronic device by rapidly turning power on and off, where the on time is called the "pulse width" and the off time is called the "duty cycle" [15]. On *Figure 12* is possible to appreciate the example of how PWM works.

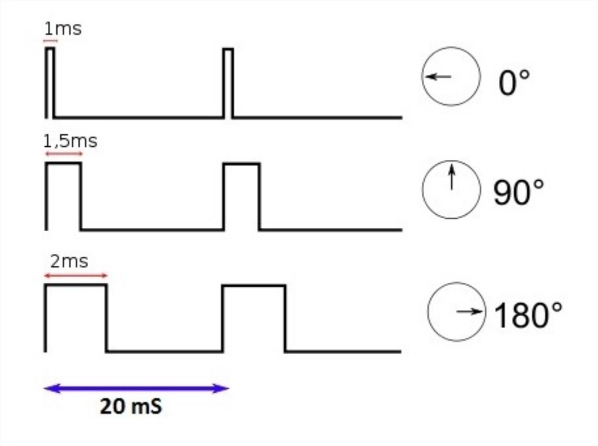

Figure 12. Examples of PMW pulses to control a servomotor [15]

2.2.3 Arduino MEGA 2560

The Arduino Mega 2560 is a microcontroller board commonly used in robotic projects for its ease of programming and community support. It is used to control servomotors and read data from an IMU sensor in a self-balancing platform project. It receives data from the IMU and uses it to run a control algorithm that generates control signals for the servomotors. The control algorithm can be a simple PID controller or a more advanced model-based method. The servomotors are controlled through a motor driver or H-bridge, and the IMU sensor is connected through an I2C or serial interface. The Arduino code may also include functions for calibrating the IMU, setting up peripherals, and communicating with other devices.

2.3 Control Design

The control of a structure is incredibly important as without it the structure would not work or would not change its electrical behaviour. Once the electrical and mechanical design have been completed it is therefore important to focus deeply into the control and programming of the system.

2.3.1 Flowchart

It is key to visualize the concept of the control before start programming it, which can be done by means of a system flowchart which shows roughly how do we want the system to work, the flowchart can be seen in *Figure 13*.

It can be appreciated in *Figure 13* that the controls complexity planned is not very complex. We are just required to physically set the platform on an equilibrated place in order to turn it on as the structure will first need to have the MPU 6050 calibrated before it can start being moved around.

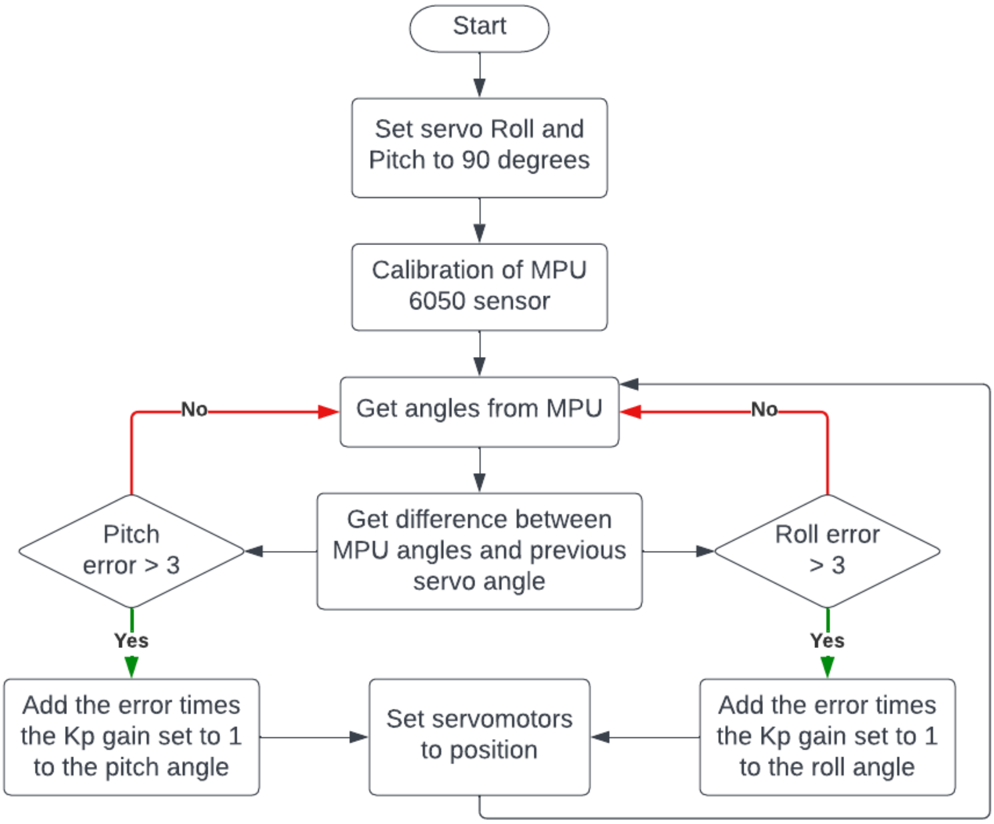

Figure 13. System Flowchart

2.3.2 Coding

Then, once it has been well visualised, we should code a program which will make the planned flowchart reality and therefore perform the effective structural movement. In the code it has used the software Arduino IDE 2.0.2 and already existing libraries in order to make the code easier.

Then because we need to use already existing libraries, first we start by importing the libraries on the code such as:

```
#include "I2Cdev.h"
#include "MPU6050_6Axis_MotionApps20.h"
// Arduino Wire library is required if I2Cdev I2CDEV_ARDUINO_WIRE
implementation
// is used in I2Cdev.h
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
    #include "Wire.h"
#endif
// Arduino Servo.h library is required to control the servomotors
#include "Servo.h"
```

Then, after initializing all the necessary variables, setting up the constants that will then be further used on the code and creating the servos instances, we reach the “set up” class. It is a class which will only be run once so the logic in it is to run whatever I need to set up or do before reaching a class which will run as a loop. Therefore, in this “set-up” class I will be first setting the servo position to what has been verified being real 90 degrees when placed on a flat equilibrated surface, then once in position we just have to give to the servomotors a few hundreds of microseconds just to ensure they reach their position before proceeding. (Previously stated: pitchServoAngle = 90 ; rollServoAngle = 95):

```
// Attaching and setting up servos
pitchServo.attach(9);
rollServo.attach(8);
pitchServo.write(pitchServoAngle);
rollServo.write(rollServoAngle);
delay(200);
```

Once the servomotors are in the respective and correct 90 degrees position the code proceeds to perform the calibration of the MPU 6050 using the following part of the code:

```
  // join I2C bus (I2Cdev library doesn't do this automatically)
  #if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
      Wire.begin();
      Wire.setClock(400000); // 400kHz I2C clock.
  #elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
      Fastwire::setup(400, true);
  #endif

  // initialize serial communication
  Serial.begin(115200);
  while (!Serial); // wait for Leonardo enumeration, others continue
immediately

  // initialize device
  mpu.initialize();

  // verify connection
  Serial.println(F("Testing device connections..."));
  Serial.println(mpu.testConnection() ? F("MPU6050 connection
successful") : F("MPU6050 connection failed"));

  // wait for ready
  delay(2000);

  // load and configure the DMP
  devStatus = mpu.dmpInitialize();

  // make sure it worked (returns 0 if so)
  if (devStatus == 0) {
      // Calibration Time: generate offsets and calibrate our MPU6050
      mpu.CalibrateAccel(6);
      mpu.CalibrateGyro(6);
      mpu.PrintActiveOffsets();
      // turn on the DMP, now that it's ready
      Serial.println(F("Enabling DMP..."));
      mpu.setDMPEnabled(true);

      // set our DMP Ready flag so the main loop() function knows it's
okay to use it
      dmpReady = true;
```

```
      // get expected DMP packet size for later comparison
      packetSize = mpu.dmpGetFIFOPacketSize();
  } else {
      Serial.print(F("DMP Initialization failed (code "));
      Serial.print(devStatus);
      Serial.println(F(")"));
  }
```

It might seem as a long code but most this amount of code will take a few seconds and if the calibration has been properly done it will set the variable dmpReady to true (logic 1) so then it can be used on the followed loop code. In the case that the variable devStatus is not equal to zero for the if condition it means that the initialization has not been done properly and therefore it would need to be reinitialized before proceeding to the calibration again. Once that’s done it steps out of the set-up class and goes into the loop class which at the beginning has the following part of code

```
  // read a packet from FIFO
  if (mpu.dmpGetCurrentFIFOPacket(fifoBuffer)) { // Get the Latest
packet

      #ifdef OUTPUT_READABLE_YAWPITCHROLL
          // display Euler angles in degrees
          mpu.dmpGetGravity(&gravity, &q);
          mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
      #endif

  }
```

This code segment is retrieving the data from the MPU getting the values for the yaw, pitch and roll angle. For the case of this design the yaw angle value will be ignored. The pitch and roll values determine the angles from the calibrated zero position on the previous code segment.

Then once the tilt angle has been retrieved from the MPU 6050 it is time to control the servos accordingly to the angle obtained.

```
  pitchServoAngle = pitchServo.read();
  rollServoAngle = rollServo.read();
```

```
  // Checking what reaction movement is required from the pitch
servomotor
  if ((ypr[1] * 180/M_PI) > 3 && pitchServoAngle < LIMIT_SERVO_MAX){
    pitchServoAngle += ypr[1] * 180/M_PI;
  } else if ((ypr[1] * 180/M_PI) < -3 && pitchServoAngle >
LIMIT_SERVO_MIN){
    pitchServoAngle -= ypr[1] * 180/M_PI;
  }

  // Checking what reaction movement is required from the roll
servomotor
  if ((ypr[2] * 180/M_PI) > 3 && rollServoAngle < LIMIT_SERVO_MAX){
    rollServoAngle -= ypr[2] * 180/M_PI;
  } else if ((ypr[2] * 180/M_PI) < -3 && rollServoAngle >
LIMIT_SERVO_MIN){
    rollServoAngle += ypr[2] * 180/M_PI;
  }

  // If values changed delay to give time to servomotors
  if(pitchServoAngle > LIMIT_SERVO_MIN && pitchServoAngle <
LIMIT_SERVO_MAX
    && rollServoAngle > LIMIT_SERVO_MIN && rollServoAngle <
LIMIT_SERVO_MAX){
      rollServo.write(rollServoAngle);
      pitchServo.write(pitchServoAngle);
      delay(80);
  }
```

The control of the platform has been kept fairly simple where we first thing to do is read the actual position of the servomotor. Once it is known we make an if statement where it is verified that the pitch and/or roll angles are larger than 3 (which is the threshold of the equilibrium position). If the angle is actually positively larger than 3 degrees, the variable previously read is added the positive difference in the case of the pitchServoAngle and the negative difference for the rollServoAngle variable, the opposite if the value read from the sensor is negatively larger than -3. In the case that one value is not larger we would leave intact the value of the servo angle.

Then, if the final output angles are in between the boundaries of the LIMIT_SERVO values the program will write the new position of the servomotors.

3 Results and conclusion

3.1 Final Design result

After all the design processes, everything has been manufactured and assemble together giving us the following final design. After testing it, it has been made a choice to change the lower servomotor controlling the roll angle to another one which did not allow the platform to have a “click” movement when it stays at a position. The new servomotor used did have the same dimensions, therefore it has been possible to use the same structure. In *Figure 14* we can appreciate the finished structure:

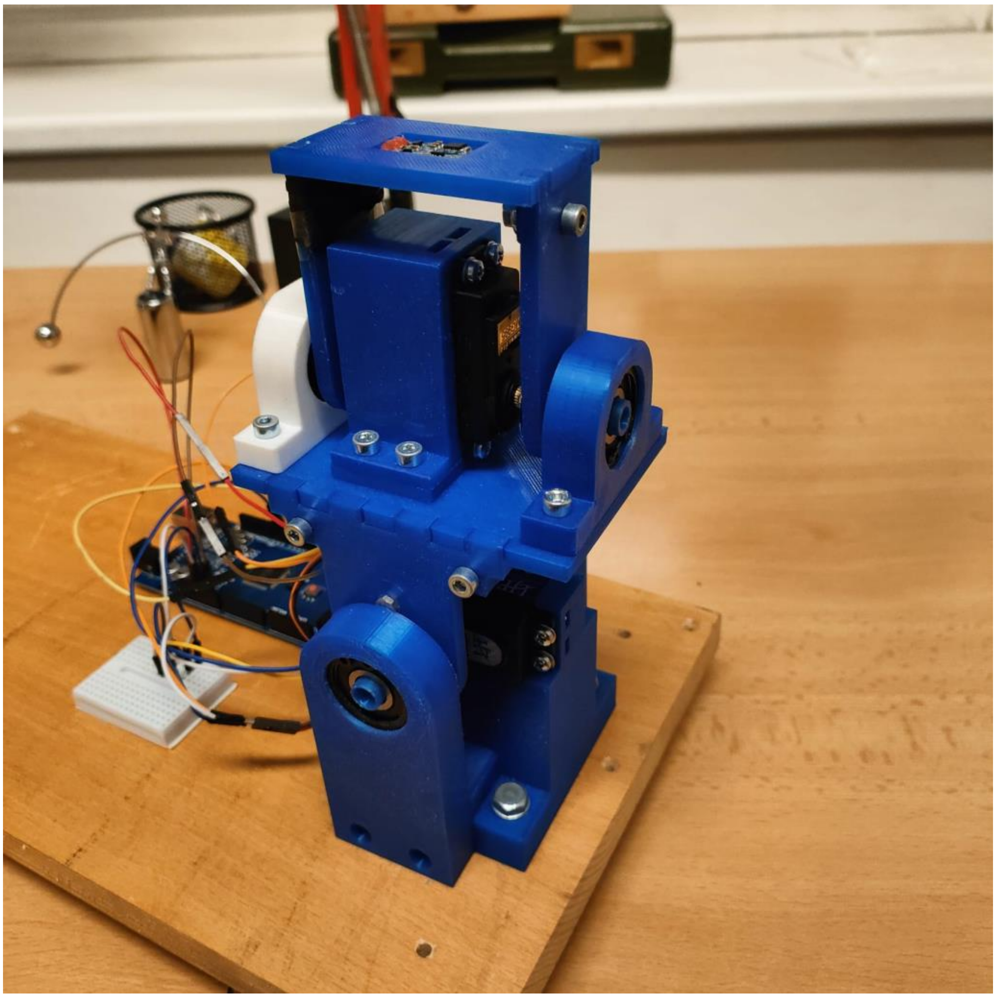

Figure 14. Designed Self-Stabilizing Platform

3.2 Tests results

In order to test the platform we have set up on top of a table a surface with a measured 30 degrees inclination. Then we perform a test where we would place the base of the structure firstly on the table so it calibrates the MPU 6050 sensor and once it is calibrated, we would move the base of the platform as quickly as possible from the 0 degrees tilt to the surface oriented at 30 degrees. Therefore we are interested in recording what are the values of the variables for the angle of each servo and the angle readings from the MPU sensor. Firstly it is interesting to check the actual angle of the homemade tilted platform using just the MPU 6050 sensor. Therefore, to perform the text we just used the same code as before but commenting out the servo moving commands and printing the values of the roll and pitch angle. The Arduino has been kept connected to the laptop so we can observe the real time graph of the variables using the Arduino serial plotter. Because of the scale not noise can be appreciated but in *Figure 17* is possible to appreciate the noise from the MPU 6050:

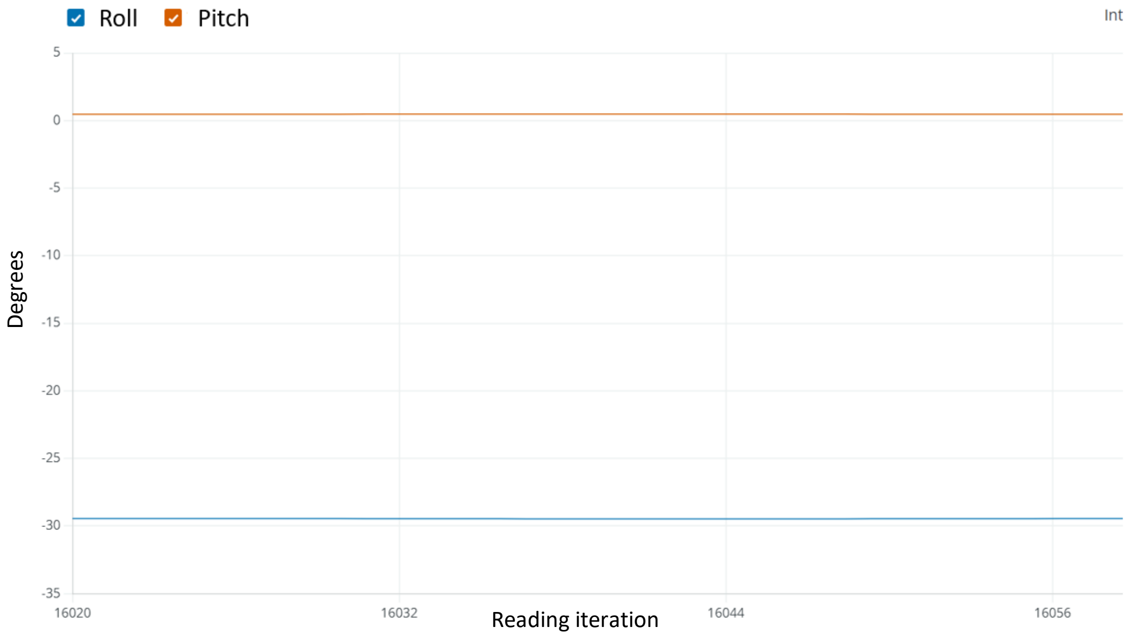

Figure 15. Platform on base at 30 degrees for MPU 6050 roll angle

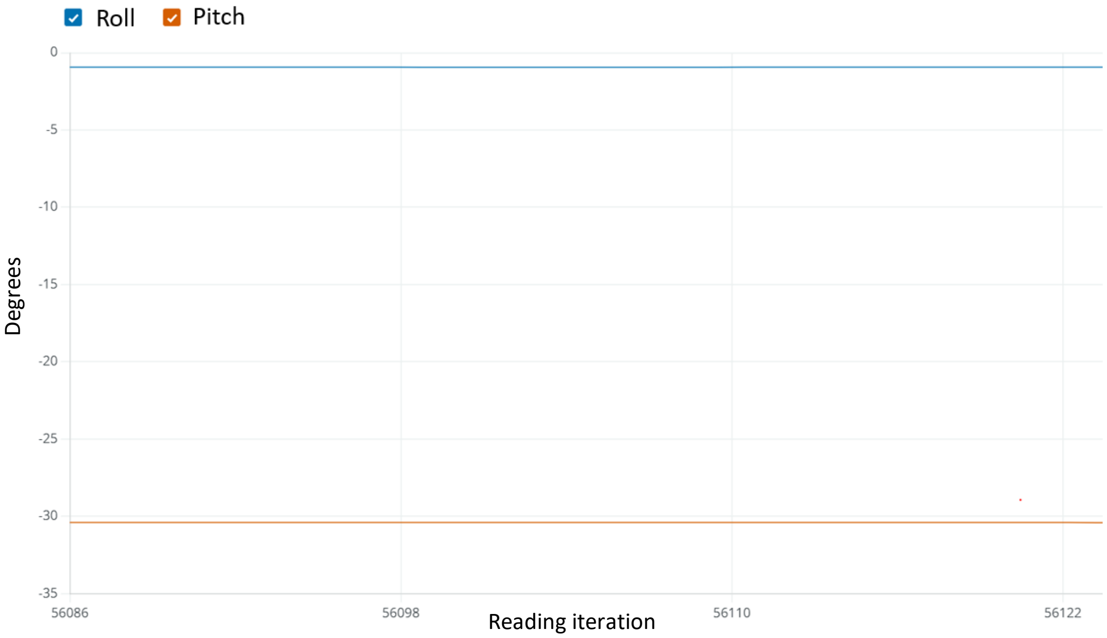

Figure 16. Platform on base at 30 degrees for MPU 6050 pitch angle

On *Figure 14* and *15* we can observe that the sensor actually reads the inclination of the platform to 30 degrees in both the roll and pitch direction. The sampling frequency is much higher too because on this case there is not the delay(80) to run the servomotors, this explains why in under 2 min the code reached 56000 iterations. It is not perfectly 30 degrees but that is due to the uncertainty of the homemade inclined plane, to the human error while placing it into position, as you can see the in both figures the variable which is supposed to remain as 0 does not remain as 0 as most likely the orientation at which the base of the platform has not been perfect. We discard that the error come from the noise, because as it can be observed in *Figure 16*, after keeping the MPU sensor at the origin position for 2 minutes the values given by the sensor did not differ from the originally given ones.

They have a range of about ± 0.04° as was observed but in the picture, time range the range is of just ± 0.02° . Those are good news because it means that the code, we use from the library of the MPU uses a good filter and we ensured to choose one which uses the DMP (Digital Motion Processor) integrated in the sensor as this one will give us a more accurate and reliable value data as it has as task to combine the raw data obtained from the gyroscope and the accelerometer.

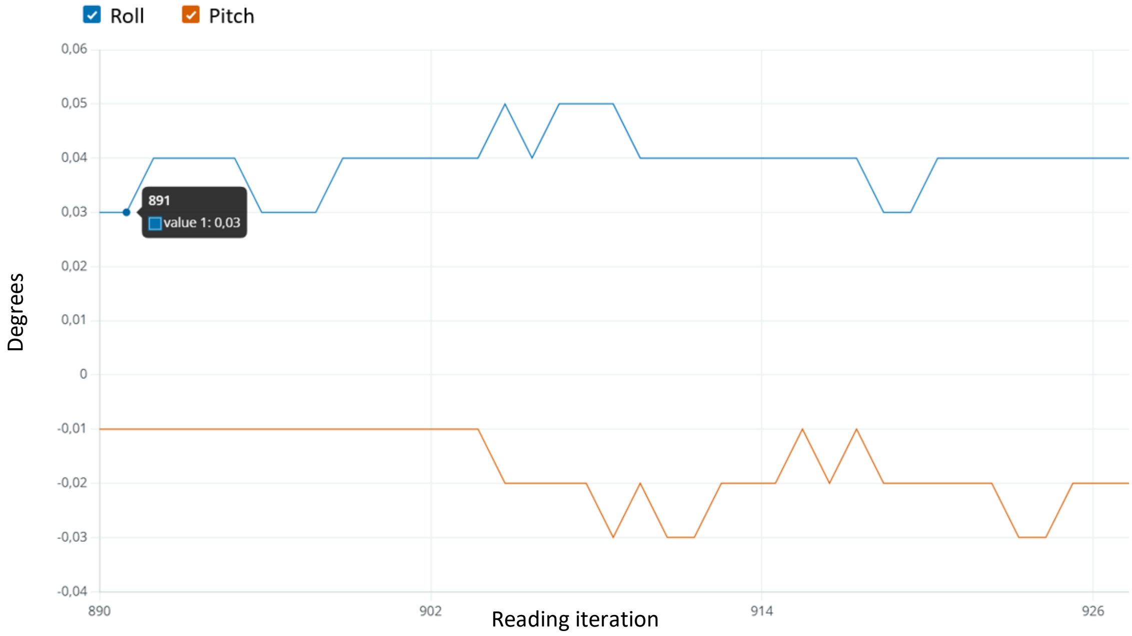

Figure 17. MPU 6050 noise

Once it has been ensured that everything was working properly, we can move on to make the actual test on the final system, with the servomotors running. It will consist of placing the structure on the 30° inclined surface manually from being previously placed at rest in a non-inclined surface. The movement is done as fast as possible to see how good the reaction from the platform is:

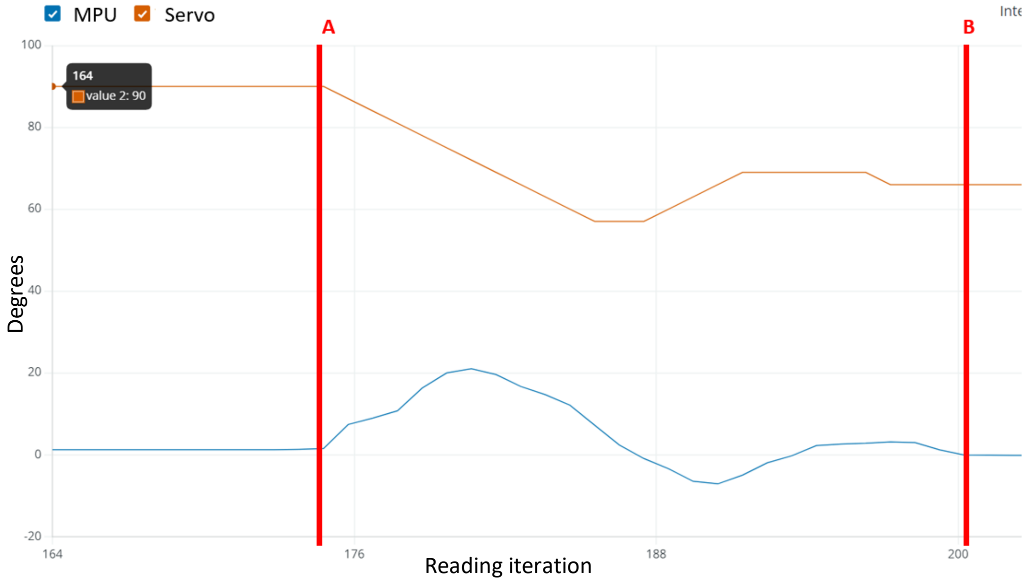

Figure 18. Roll angle reaction

As it is possible to observe in the *Figure 17*, there is the servo desired position in orange and the MPU readings in blue, those are for the roll angle control. From analysing the graph can be concluded that from point A to point B there has been 25.6 of the units on the x-axis which represents each loop with recorded data.

$$A - B\ (gap) = 25.6\ unit\ length$$

$$T_{Stabilizing} = 25.6 \cdot T_{Delay} = 25.6 \cdot 80 = 2048\ ms$$

Therefore it is possible to stablished that the time it took to the platform to reach its stabilized position is of 2 seconds. Then it is interesting also to know the precision of the movement control and more factors as the overshoot.

$$\theta_{bal} = \theta_O - (2.6 \cdot 8.8) = 90 - (2.6 \cdot 8.8) = 66.9°$$

$$\Delta\ error = |60 - \theta_{bal}| = 6.9°$$

$$\text{Overshoot} = 60 - |66.9 - (0.9 \cdot 8.8)| \approx 1°$$

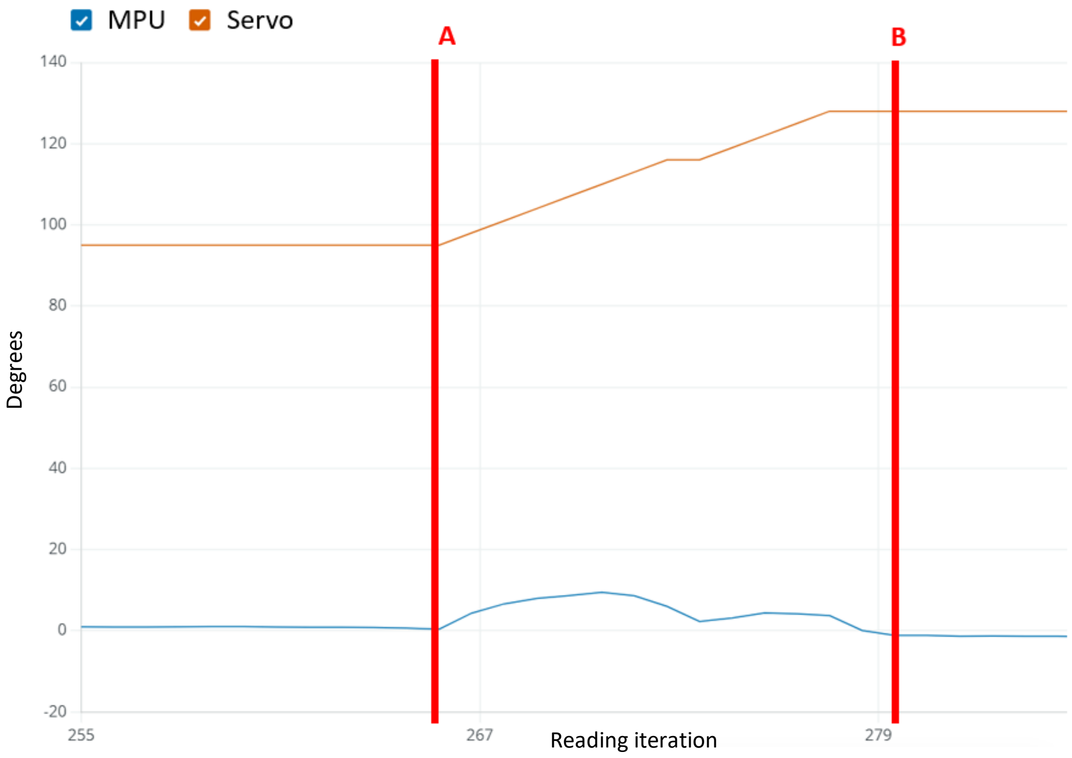

Figure 19. Pitch angle reaction

In *Figure 18*, it is possible to observe the reaction of the structural system on the pitch angle direction. In order to test it, the same procedure as for the case of *Figure 17* has been made. It is already possible to state without any calculation that the pitch angle has a better system response and accuracy. This was expected due to the fact the load for the pitch angle control servo is lower than for the roll angle control servo.

Again we have checked the distance between the start moving data to the stabilizing point data, this resulted on 13.8 units length. Therefore:

$$A - B\ (gap) = 13.8\ u.l$$

$$T_{Stabilizing} = 13.8 \cdot T_{Delay} = 13.8 \cdot 80 = 1106\ ms$$

Therefore it is possible to stablished that the time it took to the platform to reach its stabilized position is of 1.1 seconds. Then it is interesting also to know the precision of the movement control and more factors as the overshoot.

$$\theta_{bal} = \theta_O + (2.4 \cdot 11.6) = 95 + (2.4 \cdot 11.6) = 127.9°$$

$$\Delta\, error = |125 - \theta_{bal}| = 2.9°$$

$$\text{Overshoot} = |125 - 127.9| = 2.9°$$

After this, it has also been specified that the accuracy of the system for this test was with a 2.9° error. Also while the system was trying to reach 125° it did overshoot that goal by 2.9° .

4 Conclusion

By performing the tests it has been shown that even if the conception of the structure was correct, the goal response accuracy for the roll angle has not been the expected. By looking into the results from *Figure 18,* it has been stated that when the platform is moving on the rolling angle from 0 to 30 degrees, the platform has been off by 6 degrees of inaccuracy, this inaccuracy comes from either the base of the platform which might have moved the inclined base or a mistake in the control logic. The moving of the inclined base can be discarded as after repeating several times the test and getting still similar results, sometimes more accurate, sometimes as inaccurate as the one in *Figure 18.* The 5 degrees were set as aim for inaccuracy in this project, the system also had an overshoot of 1 degree when trying to reach to the 60 degrees to stabilize. Also, the response of the roll angle is twice as slow and has vibrations compared to the pitch control. On the good side, both overshoots are not of problem as they do not exceed the 3 degrees that we had set, its response is also of a medium fast speed as it moved from the 0 to 30 degrees position in only slightly over a second.

If improving this project in the future came across, a good idea would be to rethink the design now that there's more knowledge about this project. A way to rethink it would be to make it lighter, other materials could be used which with a smaller cross-section we could accomplish the same as with 3D-printing the cross-section has to be fairly large in order for it to be rigid and solid enough. It also could be possible to implement a proper PID controller instead of the way the control has been made. Even a simple P controller could work but for this it is needed servomotors to which you can change the speed of rotation and not servomotors as the ones that have been used. Typically this would require continuous rotation servomotors. If instead of servomotors, DC-motors were chosen, as long as the torque requirements are met it is also possible to implement the PID controller to it

References

[1] S.-P. Gosselin, “Development and Experimentation of a Fast 3-DOF Camera-Orienting Device,” 2003. [Online]. Available: http://ijr.sagepub.com/content/16/5/619.short.

[2] U. K. Institut fur Theoretische Elektrotechnik und Systemoptimierung, “An integrated GPS/MEMS-IMU navigation system for an autonomous helicopter.,” 2006. [Online]. Available: http://ac.els-cdn.com/S1270963806000484/1-s2.0-S1270963806000484-main.pdf?_tid=426eb69a-f7d9-11e4-9735-00000aab0f26&acdnat=1431347502_d826dff7933c69c0b57e566441853bf9.

[3] D. E. Grober, “Autonomous, self leveling, self correcting anti-motion sickness chair, bed and table,” 2009. [Online].

[4] A.-K. C. o. E. A.-J. B. Bushra Amer Tawfeeq, “International Journal of Electrical and Computer Engineering (IJECE),” *A self-balancing platform on a mobile car,* p. 12, 02 08 2022.

[5] S. P. A. P. Ritesh Nayak, “Self Balancing Monopod,” International Institute of Information Technology, 2018.

[6] J. Tang, “Oriental Motor,” 31 03 2020. [Online]. Available: https://blog.orientalmotor.com/motor-sizing-basics-part-4-radial-load-and-axial-load.

[7] “Microcontrollers Lab,” [Online]. Available: https://microcontrollerslab.com/mg995-servo-motor-pinout-interfacing-

with-arduino-features-examples/#:~:text=MG995%20Features%20and%20Specifications%201%20Operating%20Voltage%3A%204.8,8%20Dead%20band%20width%3A%205%20usec%20More%20items.

[8] TowerPro, [Online]. Available: https://www.towerpro.com.tw/product/mg995/. [Accessed 28 10 2022].

[9] “629-2RSH,” SFK, [Online]. Available: https://www.skf.com/sg/products/rolling-bearings/ball-bearings/deep-groove-ball-bearings/productid-629-2RSH.

[10] “Standards DIN 472,” [Online]. Available: https://www.fasteners.eu/standards/DIN/472/.

[11] J. Borah, “Evaluation of tensile strength of 3D printed PLA component and Taguchi optimization,” in *Neuro Quantology*, vol. 20, 2022, pp. 1714-1726.

[12] J. Joshep, “How Does the MPU6050 Accelerometer & Gyroscope Sensor Work and Interfacing It With Arduino,” 16 05 2022. [Online]. Available: https://circuitdigest.com/microcontroller-projects/interfacing-mpu6050-module-with-arduino#:~:text=The%20MPU6050%20is%20a%20Micro,of%20a%20system%20or%20object..

[13] “Gyroscope,” TechTarget, 03 2016. [Online]. Available: https://www.techtarget.com/whatis/definition/gyroscope#:~:text=A%20gyroscope%20is%20a%20device,unless%20subjected%20to%20external%20torque..

[14] C. Woodford, “Accelerometers,” 01 02 2022. [Online]. Available: https://www.explainthatstuff.com/accelerometers.html.

[15] B. S. a. A. M. Andi Adriansyah Musaab, “Design of AHRS for Quadrotor Control using Digital Motion Processor,” Department of Electrical Engineering, Faculty of Engineering, Universitas Mercu Buana, Jakarta.

[16] “Práctica 5 con Miuva. Servomotor con pulsador en PIC C,” [Online].

[17] W. Gastreich, “WHAT IS A SERVO MOTOR AND HOW IT WORKS?,” 27 08 2018. [Online].

[18] C. Pao, “What is an IMU Sensor,” 15 November 2018. [Online]. Available: https://www.ceva-dsp.com/ourblog/what-is-an-imu-sensor/#:~:text=An%20IMU%20is%20a%20specific,considered%20a%209%2Daxis%20IMU..