NEWT

Gyro stabilizing platform for design evaluation

ANTON GEBERT
KAJ JONSSON
ALEJANDRO KURATOMI
MAX LINDGREN
STANISLAV MINKO
FREDRIK SANDAHL
HENRIK STRÖMQVIST
ANDRÉ SÄLL

Mechatronics Advanced Course, MF2059
Date: December 16, 2018
Coach: Didem Gürdür
Course Responsible: Björn Möller
KTH-Royal Institute of Technology, Machine Design

Abstract

This project covers the design of a self balancing platform. The balancing is carried out by active stabilization, by using a control moment gyroscope. The platform is designed to handle small impulse and step disturbances, and weighs approximately 100 kg. The report mainly focuses on material that is important to someone inheriting the project. The prototype successfully balanced for several hours, and managed to withstand disturbances such as small kicks, pushes, and weights placed besides the balancing point.

Acknowledgements

Many thanks to Ivan Stenius, Nicholas Honeth, Didem Gürdür, Staffan Qvarnström, Tomas Östberg, Uddeholm AB, Anton Kviberg, Anton Boström and Jan Stamer, for all the help and support they have provided.

Contents

Chapter 1

Introduction

1.1 Background

With the continued trend of increasing population in the worlds largest and most crowded cities, and the need of environmentally friendly ways of transport, new solutions are needed. One such solution is the NEWT project. The NEWT project is the vision of a small two-wheeled, one-person vehicle with both land and water traversing capabilities. Such a vehicle would be better suited for the average commuter than the common car, taking up less space, needing less energy, and being able to use the potential of water ways, commonly found in the vicinity of large cities but largely unused by commuters.

The most common two-wheeled vehicle used in road traffic is the motorbike, which is arguably harder to control than a car, especially at low speeds and standstill. It also exposes the driver to the elements and other vehicles through the lack of an enclosed driver position. However, providing the driver with an enclosed position, thus shielding the driver, further increases the difficulty of handling the vehicle, by preventing the driver from using his or her legs to stabilize the vehicle. An enclosed, two-wheeled vehicle would therefore need some other way of balancing.

1.2 Project Description

The aim of this project is to explore, implement, and evaluate the possibility of balancing an unstable, two-wheeled vehicle such as the NEWT vehicle, through an active control moment gyroscope. This will be done by conducting a literature study, as well as implementing the balancing system on a full scale experimental platform.

1.3 Requirements

The projects' stakeholders initially stated a set of requirements for the experimental platform which was to be developed. These were reformulated into technical requirements. Usage of the word "shall" indicates a necessary property, while "should" indicates a desired property.

1.3.1 Stakeholder Requirements

1. The experimental platform must be full scale in size.
2. The experimental platform must balance by itself on two wheels.
3. The experimental platform must be safe to operate.
4. The development cost should not exceed 50,000 SEK.
5. The balancing system should be light-weight and compact.
6. The experimental platform should be able to provide energy harvesting.
7. The balancing system should be silent for both driver and surroundings.

1.3.2 Technical Requirements

1.1 The experimental rig must be able to hold one person weighing 100 kg.

2.1 The experimental platform must be able to balance by itself while at stand still.

2.2 The balancing system must withstand roll disturbances.

2.3 Torque orthogonal to the vehicle's driving direction generated by the balancing system must be avoided.

2.4 The experimental platform must be able to balance on land.

2.5 The balancing solution should not cause disturbing forces while driving.

2.6 The balancing solution should be able to work on an amphibian platform.

3.1 The experimental platform must have a manual emergency brake.

3.2 The experimental platform must have a safety seat belt.

3.3 The experimental platform must have driver roll protection.

1.4 Delimitations

Although the initial requirements were such as they are described in the previous section, they were soon deemed to be too comprehensive, if all requirements were to be achieved in the time given, when building the experimental platform. Due to limitations in time, money, access to key manufacturing resources, and safety concerns, the following delimitations were made.

The originally devised experimental platform was changed from being a two-wheeled platform with means of propulsion and steering, and seating for a driver, into an unmanned platform, balancing on two points of contacts which were not wheels. The driver and driving related requirements were thus scrapped. The energy harvesting part was also scrapped. Because it was only possible to manufacture one gimbal containing the flywheel, the technical requirement 2.3 regarding torque orthogonal to the vehicles driving direction, were also scrapped, although adding another gimbal should mean achieving this requirement.

1.5 Reader's Guide

From this point on, the document will be divided in 6 chapters, following the chronological order of the whole development process. First, a thorough state of the art review will be carried out regarding the most important elements of the self-balancing mechanisms and related systems used. Second, a description of the methodology used, specifically the team organization, and strategy to design and build the solution is given. Third, a detailed specification on the implementation of the prototype and its subsystems, namely dynamics, control, mechanics, electronics and software is stated. In the next section, verification and validation, some of these implemented subsystems, specifically Control and Electronics are in-depth tested to verify their functionalities and limitations, and the most relevant tests are described. Finally, the results from the final testing and future work to be carried out within the field and on this prototype are shown and discussed.

Chapter 2

State of the art

In this chapter, existing designs and applications of gyroscope balanced vehicles are presented. The difference between passive and active gyroscope stabilization is explained. Different means of measuring the roll rate of the platform to be balanced are also investigated.

2.1 Passive gyros

Passive balancing gyro systems make use of the gyros balancing properties, but they do not actively control the precession of the gyro. Instead, the system lets the gyro react naturally to the disturbances, were the high inertia produced by the gyro resists change in motion which produces the counter torque.

The stabilization solution emerged in the 1920s were Louis Brennan developed a full scale prototype of a monorail train car [1], shown in Figure 2.1. The train had two vertical gyros spinning in opposite directions which in turn precessed in opposite directions.

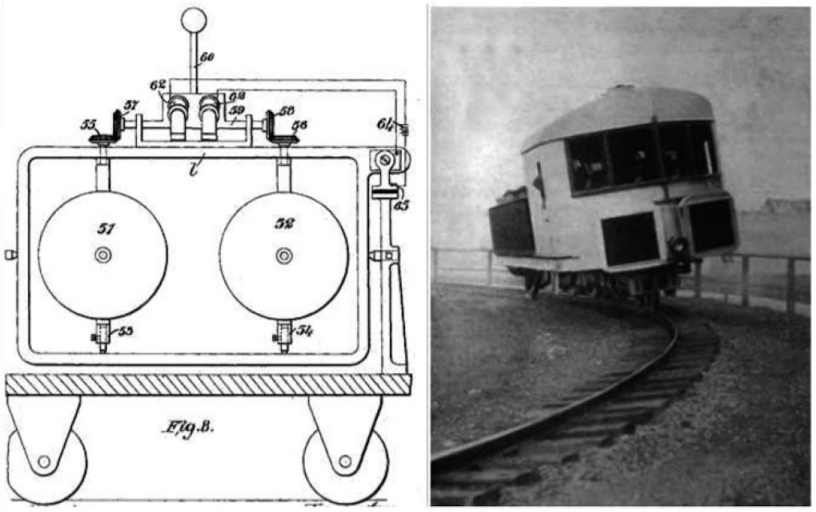


Figure 2.1: The structure of the two gyros coupling (left), and the monorail prototype (right). [2]

This coupling made the gyros cancel out any unwanted produced torque apart from the torque in the roll direction. Due to cut of funding the invention did not make it through the prototype stage. But it's viability were shown.

Today a common use for passive gyro has been the anti-roll stabilization gyro for larger sea vessels such as yachts [3]. The flywheel has then been mounted to spin around the vertical axis with the precession angle in the ships fore and aft directions. Because of the large inertia of ships in the yaw axis the torque produced in this direction can thus be neglected, therefore this solution can rely on one gyro [4]. In those implementations, both passive and active solutions have been used, where the passive gyro solution has been less complex and easier to implement compared to active controlled gyro. By investigating the difference with a passive gyro and an actively controlled gyro for a ship vessel for reduced roll, it was shown that the active gyro clearly outperforms the passive for reduced roll motion [4], as shown in Figure 2.2 .

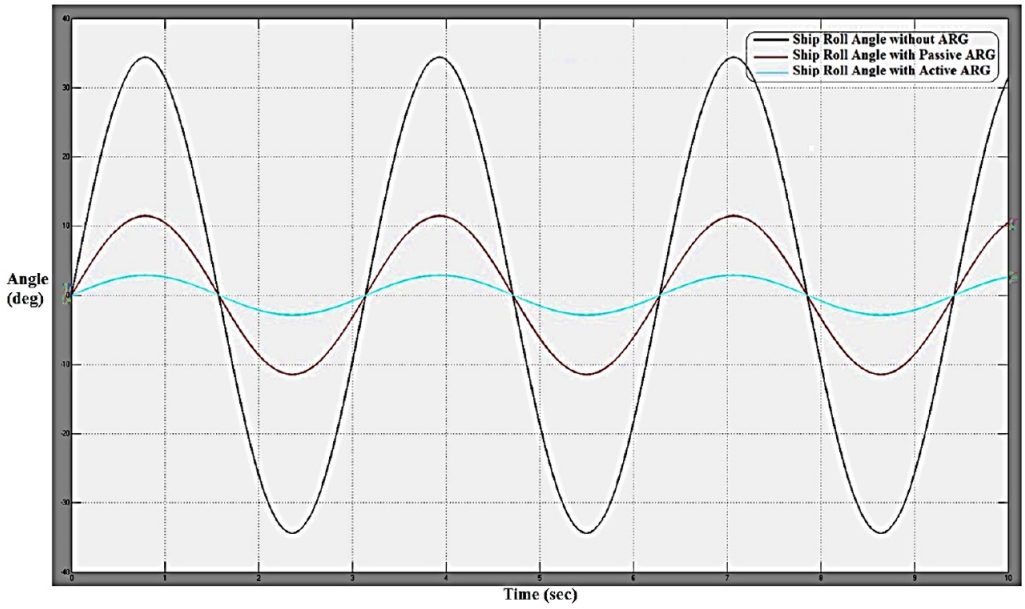


Figure 2.2: The difference in roll angle of a ship vessel induced by waves over time, for active and passive gyro. ARG - Anti roll gyroscope [3].

2.2 Active Gyros

Adding to the existing passive gyro stabilization principle where the precession angle is naturally changed to counter disturbances, it may also be adjusted through external input. What is meant by this is that the angular velocity of the precession may be changed with a servo or motor, resulting in a torque around the roll axis, which may be used to counter disturbances.

It is also important to add that passive gyros may only reject a certain amount of energy in disturbances before they reach a complete change of 90 degrees rotation around the precession axis and loses the torque generating ability around the roll axis. It could be seen as a type of "battery" in terms of disturbance rejection; a finite amount of energy to counter the work excerted by disturbance forces or torques. With an active control of the precession angle, an input of energy in terms of a servo or motor acting on the gyro this "finite" amount of energy in the gyro is no longer a problem and more mechanical work in the form of disturbances may be excerted on the system.

2.2.1 LitMotors

A contemporary example of this principle is the LitMotors concept, currently in the latter stages of development, which should also be mentioned is a big inspiration for the project. A two-wheeled vehicle is balanced by two active gyros placed beneath the drivers seat and act as roll stabilizers, therefore not needing a driver to use their legs to balance at standstill, for example which is the case when using a motorcycle. This in turn gives an opportunity to enclose the driver interface more resembling a car. Pictures of the LitMotor platform are shown in Figure 2.3 and Figure 2.4.

Figure 2.3: The LitMotors concept vehicle C-1 [5].

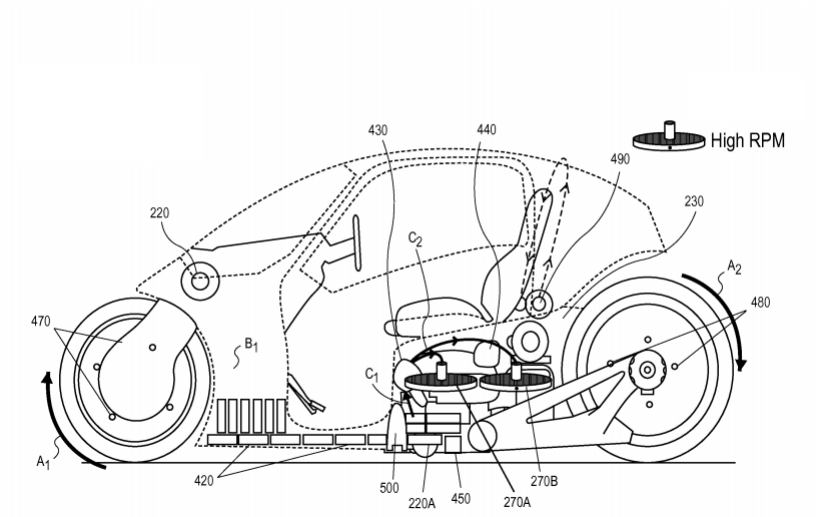


Figure 2.4: A sketch from the LitMotor system patent [6] showing the setup of the gyro stabilizing system.

The gyros in this system are mirrored to one another, i.e the precession angle is inverted compared to the other. This is done as to counter the torque around the pitch axis as followed by the dynamics of a gyro creating torque around multiple axes when moving the precession angle. An unwanted torque around the yaw-axis could cause unwanted behavior, especially if the project in later stages includes water-based transport; pitch torque would lift either the fore or aft when moving the precession angle. Also having two gyros is also a practical way of dividing the flywheel inertia and thus lowering requirements of the motor spinning it.

2.2.2 SeaKeeper

Another usage of active gyroscopic stabilizers, as mentioned before, is within the naval industry. One example is the company SeaKeeper Inc. which builds active gyros for stabilizing different naval vessels to dampen rolling motions which can be seen in picture 2.5 [7].

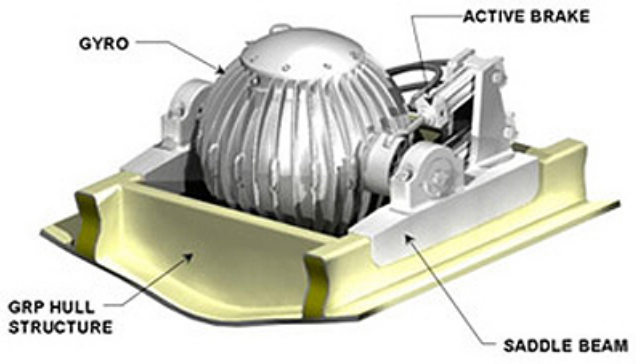


Figure 2.5: Model of the SeaKeeper concept of roll dampening unit for a naval vessel [7].

Although slightly different in terms of size, mechanical problem (balancing inverted pendulum or a floating vessel) and disturbances (gusts of wind, vehicle turning or low frequency sea waves) it is still showing proof of the many possibilities of using active gyroscopic stabilizers.

2.2.3 Angular Velocity Measurement

A gyroscope can be used to measure angular velocity. Since the range of tasks when angular velocity or position is required varies dramatically there are several types of gyroscopes depending on your application. First gyroscopes to be developed were

the passive mechanical gyroscopes. Independence of magnetic field made them frequently used for navigation once ships were build of steel rather then wood. Main disadvantages are the allocated space by physical construction and it's precision.

Therefore a more accurate type of gyroscope, ring laser gyroscope, were developed in 1963 [8]. Its construction consists of ring laser and three mirrors. The beams travel same path but in opposite directions and the difference in frequencies shows us the rotation, see Figure 2.6 below.

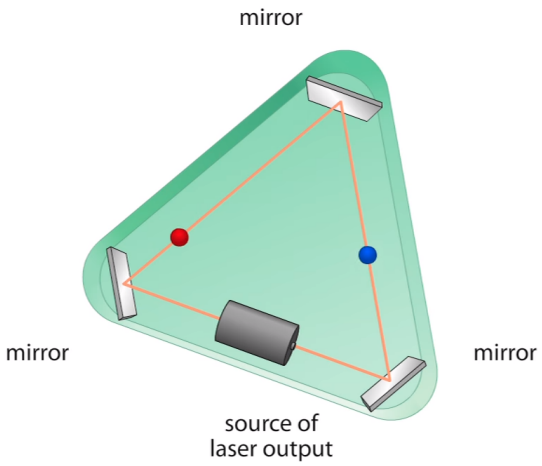


Figure 2.6: Illustration of ring laser gyroscope (RLG).

Ring laser gyroscope increased the precision of measurements, it also lowered the weight of the construction and it is free from moving parts. Drawback of this solution is it's high cost and size, if we talk about smartphones applications.

Microelectromechanical gyroscopes operate in same principal as MEMS accelerometers. Continuously moving mass that will cause change in capacitance once object rotates is represented in figure 2.7 below. Once object will start to rotate a difference in capacitance will be detected.

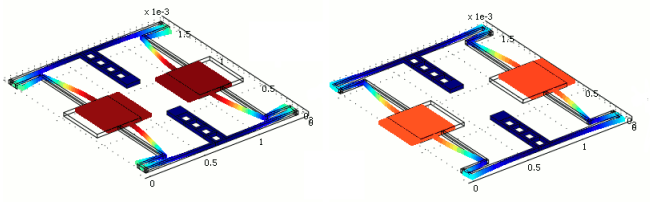


Figure 2.7: Illustration of moving mass inside silicon chip of MEMS gyroscope [9].

MEMS gyroscopes are extremely small, they fit in tiny silicone chip, lightweight and

easy to manufacture making them a perfect choice for consumer electronic products. The drawback is the precision provided by them and low disturbance rejection.

Finally the last option of the gyroscope is fiber optical gyro (FOG). The idea is the same as in ring laser gyroscope but now we send beam trough fiber optical coil allowing cheaper construction and faster response[10].

2.3 IMU

Accelerometer data consists of a motion acceleration and gravity component. These cannot be separated, unless combined with gyroscope to help remove the gravity component from the accelerometer output. The error due to the gravity component of the accelerometer data can quickly become large after the required integration process to determine position from acceleration. Due to accumulating error, a gyroscope alone is not sufficient for determining position. Gyroscopes do not sense gravity, so they can be used as a support sensor along with an accelerometer.

A combination of accelerometer and gyroscope sensors is called inertial measurement unit (IMU). Basic setup consists of 6 degrees of freedom and measures acceleration and velocity in 3 axis each. IMUs are widely used in all sorts of navigation systems[11]. Figure 2.8 shows IMU used in intermediate range ballistic missile (IRBM) and a modern spacecraft.

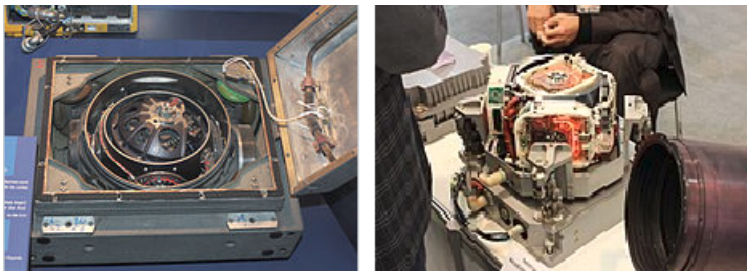


Figure 2.8: Picture of IMU for IRBM (left)[12] and a spacecraft (right).

More advanced IMU versions also include 3 magnetic field sensors, pressure sensor and even GPS allowing up to 11 degrees of freedom measurements. Figure 2.9 below is a modern ArduIMU+ V3 MEMS 11 DOF IMU with a total weight of only 18.1 g [13].

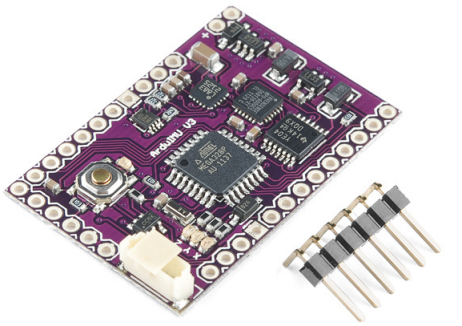


Figure 2.9: Picture of ArduIMU+ V3 MEMS IMU.

Since the accelerometer measures all forces it will be affected not only by the g force but also by every other force acting on the system. All the disturbances forces will affect measures so a low pass filer is needed. Further the gyroscope will drift due to constant integration of the velocity and it's data is reliable only during short time. One of following filers are normally implemented to increase reliability of received data: Kalman filter, Complementary filter or Mahony filter[14].

Chapter 3

Methodology

3.1 Organization

This project has been carried out by eight people and the work was divided into different subgroups, electrical, mechanical, software and control. The subgroup each person was assigned to was determined by their knowledge and the amount of work needed for each part. If one of the groups was at risk of missing their deadline, resources were allocated to that group. Most often during the project people worked in their area of expertise. There was one group leader during the whole project, but every decision was mutually agreed upon by the hole team or the subgroup that would be affected by the decision. There was one group meeting every week to see the progress of every subgroup, and to ensure that the deadline was going to be met.

3.2 Workflow

The first plan for the project was to have at least two iterations and then add features to the test rig depending on the time available. Due to safety concern and the energy stored in the flywheel a safe and robust construction was needed. Because the limited time only one prototype was constructed to ensure that it would be safe to run the prototype. This led the project to be developed using the waterfall model with some modification, every separate component has been tested before integration and every subsystem has been tested separately as well. The workflow can be seen in figure 3.1.

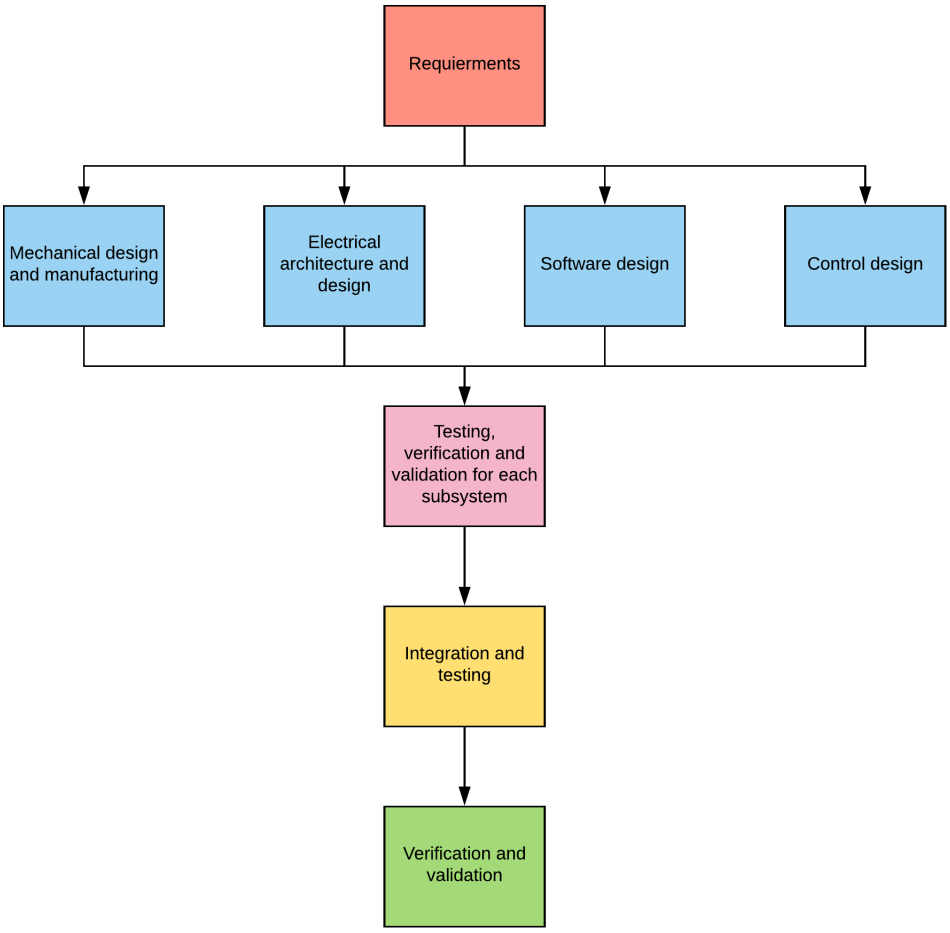


Figure 3.1: The workflow during this project.

3.2.1 Mechanical

The prototype was first designed with the Computer-aided design(CAD) program Solid Edge ST10 to see that every part would fit together and to ensure that the prototype would become as compact as possible. From the CAD, all the necessary parameters could be provided to the simulation group so they could develop a controller simultaneously as all the components where manufactured. Most of the mechanical components on the balancing platform has been constructed by the people in this project.

3.2.2 Control

To ensure that the model of the system was correct, a simulation of the system was created in Simulink and Simscape. The Simulink and Simscape model gave a similar

result. When accurate parameter for the rig was provided by the Mechanical group a controller could be developed that could handle the disturbance set by the requirements. Both a PID and an LQR controller was develop. From the simulation, the minimum velocity and torque for the flywheel and precession motor could be decided. The requirements for the motors was then provided to the Electrical group so the necessary motors could be obtained.

3.2.3 Electrical and Software design

An electrical architecture and software design was developed in tandem. The sensors that was implemented was one IMU whose function was to output the roll-rate and angle of the platform and an encoder that measured the angle of the flywheel. One micro-controller(Arduino Mega) for controlling the motors and reading data from the sensors and one computer(Raspberry pi 3b) for storing the data and sending commands to Arduino was implemented together. When the required torque and speed for the motors was acquired from the control group the motors could be obtained and tested.

3.2.4 Implementation and testing

Once every component was developed and tested independently, implementation and testing of each subsystem started. Every subsystem was tested separately, for example first the rotation of the flywheel was tested using the flywheel motor and then the encoder was tested when it was mounted on the precession axis with everything else turned off. Once every subsystem was tested independently they could be integrated together.

3.2.5 Verification and validation

When the experimental platform was completed and was balancing the system was tested to see if the it met the requirements and to see that the platform behaved the same as the simulation.

Chapter 4

Implementation

4.1 System overview

The overall design is made up of a double gimbal to able the motion of the flywheel to rotate in two planes, to constrain the rotation of the flywheel in the third plane, and to prevent any translational motion. The whole construction is able to tilt on either side to simulate the two-wheeled vehicle to tip over. Two actuators is used. One to produce the precession motion of the flywheel, i.e. to control the pitch motion of the flywheel. The other actuator control the rotational spinning motion of the flywheel which can be seen in pictures 4.1 and 4.2.

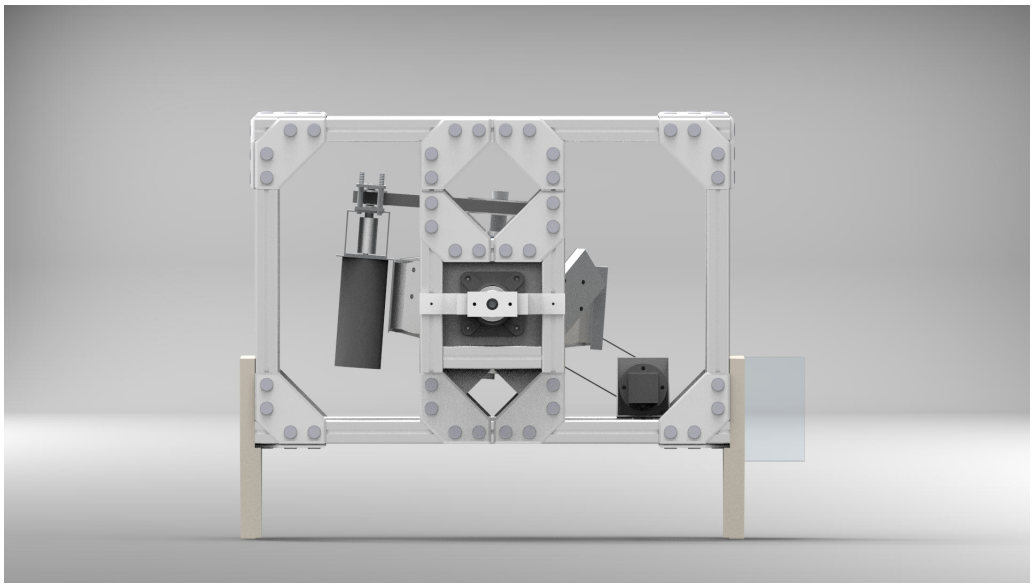


Figure 4.1: The full system visualized in CAD software, showing the side view of the platform, *Solid Edge ST10*.

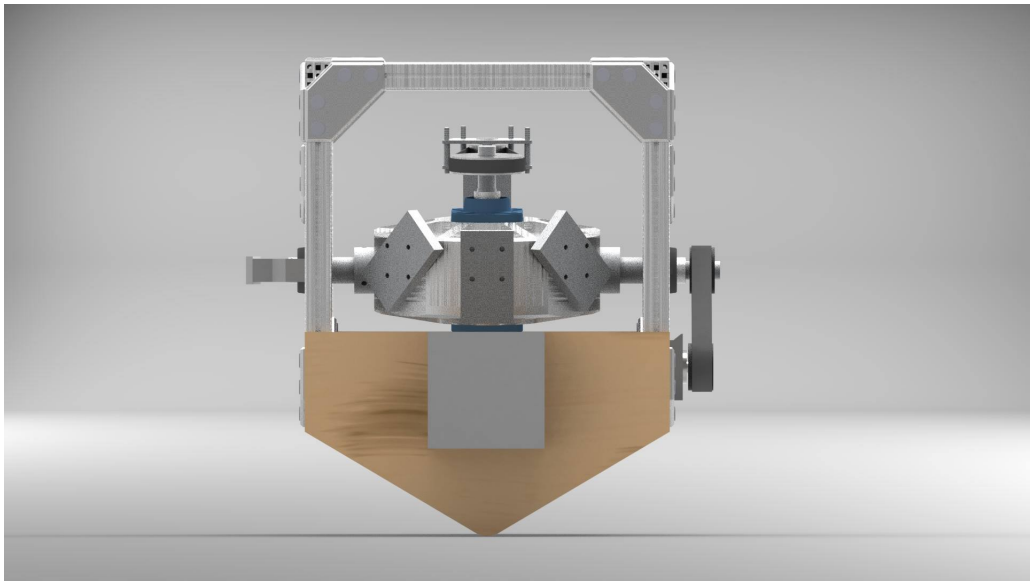


Figure 4.2: The full system visualized in CAD software, showing the front view of the platform, *Solid Edge ST10*.

4.2 Dynamics & Modeling

To solidify the foundation of the project and showing its feasibility, a firm grasp on the dynamical system is needed together with models to support it. The idea with this chapter is to explain more into detail how a gyroscope interacts with a platform with only two contact points.

4.2.1 Gyro dynamics

Figure 4.3 below shows the typical components of a gyroscope. The main component is the rotor, a flywheel rotating with high angular velocity around its spin axis, giving it an angular momentum of

$$\vec{L} = I \cdot \vec{\Omega} \, [\mathrm{kg} \cdot \mathrm{m}^2/\mathrm{s}] \tag{4.1}$$

where I [kg$\cdot$m^2] is the flywheels moment of inertia around its spin axis, and $\vec{\Omega}$ [rad/s] is its angular velocity. Assuming there is no friction in between the gimbal and gyroscope frame, the flywheel will maintain its spin axis no matter the change of orientation of the gimbal and frame. This is due to conservation of angular momentum; as a consequence of Newton's first law of motion, the angular momentum of a body will be constant unless acted upon by an external torque.

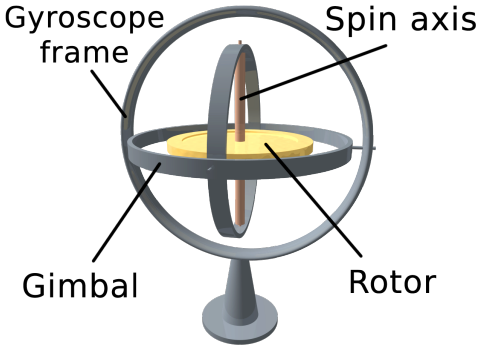


Figure 4.3: A 3D-rendering of a gyroscope. [15]

Forcing the flywheels spin axis to change by controlling its mounting creates an interesting reaction. The flywheel will exert a torque orthogonal to its precession axis, i.e. the change of spin axis, and its spin axis. This torque is equal to

$$\vec{T}_{gyro} = \vec{L} \times \vec{\omega} \, [\mathrm{N} \cdot \mathrm{m}] \tag{4.2}$$

where ω [rad/s] is the precession velocity [16]. This reaction is visualized in Figure 4.4.

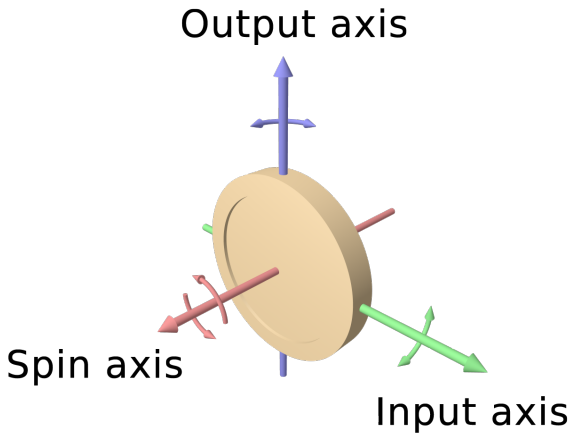


Figure 4.4: A gyro flywheel precessing giving rise to a reaction torque. The spin axis vector corresponds to $\vec{\Omega}$, the input axis vector to $\vec{\omega}$, and the output axis vector to $\vec{T}_{gyro}$ [17].

Conversely, the inverse is true as well; if a torque $\vec{T}_{gyro}$ is applied to the gyro flywheel, the flywheel will yield a torque which seeks to precess the flywheel with $\vec{\omega}$, thus reacting to the applied torque with a torque of equal magnitude in opposite direction [2]. This gyroscopic behavior, both actively forcing the gyro to precess thus yielding a reaction torque, as well as inversely applying a torque on the gyro thus passively yielding precession, can be used for stabilizing applications.

In Figure 4.5 below a flywheel is mounted in a single axis gimbal. Assuming this was mounted in a two wheeled vehicle, with the mounting points in gray to the sides of the vehicle, this setup would be able to stabilize roll of the vehicle. This could be done either passively, by letting the flywheel's natural precession counter disturbance torques on the roll axis, or actively by forcing the flywheel to precess, e.g. with an electric actuator, to yield a counter torque at the designer's discretion. In the later case, the setup would be called a control moment gyroscope (CMG)[18].

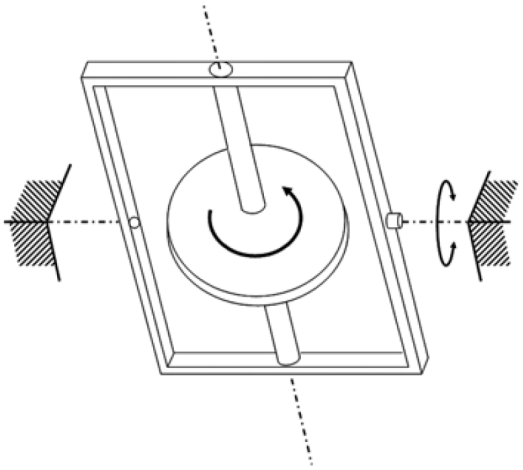


Figure 4.5: A single gimbal gyroscope[2].

4.2.2 Platform dynamics

The entire mechanical system can be described through the usage of Lagrangian mechanics equations; this method uses generalized coordinates and forces/torques which in the end simplifies derivation of the equations of motion, it also assures that no terms of mechanical effects (like Coriolis) are missed and signs are correctly assigned. Using the classic Newtonian method which views each component separately, kinematic constraints and reaction forces must be set up between each part and solved to derive the equations of motion for the entire system. In the end it will be adding more complexity to the problem as the risk for error increases when connecting the separate systems.

The Euler-Lagrange derivation is defined as [19]:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i \tag{4.3}$$

Where L is equal to the difference between potential and kinetic energy T and V:

$$L = T - V \tag{4.4}$$

The q contains the generalized coordinate which in this case will correspond to roll and precession angles, θ and α. The Q_i contains the generalized forces acting on the system, in this instance meaning the precession torque and the disturbance torque on the roll-axis. The total kinetic energy of the system consists of multiple terms of inertias multiplied with angular velocities where the following angular velocities are defined as:

$$\begin{aligned} \omega_1 &= \dot{\theta} \\ \omega_2 &= \dot{\theta} cos\alpha \\ \omega_3 &= \dot{\alpha} \\ \omega_4 &= \dot{\theta} sin\alpha \end{aligned} \tag{4.5}$$

Where ω_1 is the roll rate of the vehicle (i.e the rotation around the x-axis). ω_2 is the angular velocity of the gyroscope with respect to the x-axis, ω_3 the precession rate of the gyroscope and ω_4 the angular velocity of it with respect to the z-axis. Since the problem is only concerning roll stabilization, pitch and yaw velocities may be omitted for this project. The end result with a test rig will be mounted to the floor, thus preventing any such motion. A simplified model of the system is presented in figure 4.6.

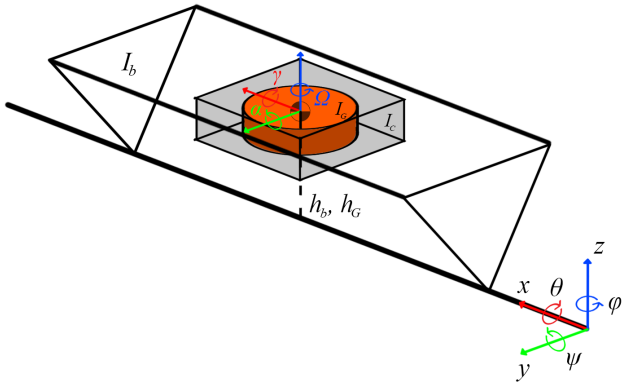


Figure 4.6: A simplified visualization of the platform setup.

To summarize, the following kinetic energy is present in the system.

$$T = \frac{1}{2}(I_{bx}\omega_1^2 + (I_{Cx} + I_{Gx})\omega_2^2 + (I_{Cy} + I_{Gy})\omega_3^2 + I_{Cz}\omega_4^2 + I_{Gz}(\omega_4 + \Omega)^2) \tag{4.6}$$

The potential energy is simplified by a model of an inverted pendulum 4.7:

$$V = (h_b m_b + h_G(m_G + m_C))gcos\theta \tag{4.7}$$

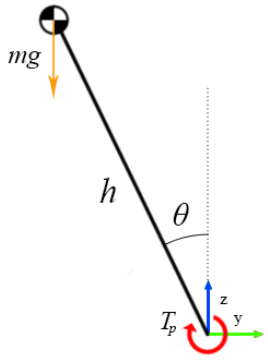


Figure 4.7: A visualization of the generalized inverted pendulum problem.

With equations (4.3) and (4.4) two differential equations may be defined:

$$\left(\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\theta}}\right) - \frac{\partial T}{\partial \theta}\right) - \left(\frac{d}{dt}\left(\frac{\partial V}{\partial \dot{\theta}}\right) - \frac{\partial V}{\partial \theta}\right) = \tau_{disturbance} \tag{4.8}$$

$$\left(\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\alpha}}\right) - \frac{\partial T}{\partial \alpha}\right) - \left(\frac{d}{dt}\left(\frac{\partial V}{\partial \dot{\alpha}}\right) - \frac{\partial V}{\partial \alpha}\right) = \tau_{control} \tag{4.9}$$

To make the equations clearer and shorter some variable changes are done.

$$\begin{gathered} k_1 = I_{Cx} + I_{Gx} \\ k_2 = I_{Cy} + I_{Gy} \\ k_3 = I_{Cz} + I_{Gz} \\ k_4 = I_{bx} \\ k_5 = k1 - k3 \\ k_6 = h_b = h_c = h_g \\ k_7 = m_G + m_b + m_C \\ cos = c \\ sin = s \end{gathered} \tag{4.10}$$

$$(k_4 + k_1 c\alpha^2 + k_3 s\alpha^2 + k_6^2 k_7)\ddot{\theta} + 2\dot{\alpha} s\alpha c\alpha k_5 + c\alpha \Omega I_{Gz} - k_6 k_7 g s\theta = \tau_{disturbance} \tag{4.11}$$

$$k_2\ddot{\alpha} + k_5\dot{\theta}^2 s\alpha c\alpha - I_{Gz}\Omega c\alpha\dot{\theta} = \tau_{control} \tag{4.12}$$

The term of $k_6^2 k_7\ddot{\theta}$ must be added as the system is modeled from the x-axis where the vehicle has its contact points; thus needing to apply the Parallel Axis Theorem to model the correct amount of inertia of the system [20]. Also take note that the center of mass placements are equal for all parts of the system, this is assumed since the majority of the weight will come from the gyroscope and its cage on the test rig, thus defining the overall center of mass. This simplification also helps the modeling in later stages.

To make it possible to derive transfer functions and state-space representations a linearization around a stable point must be done. For this case, the point with $\theta = 0$ and $\alpha = 0$ is chosen as this is the standing up position and with as much precession balancing effort around the roll axis.

$$\begin{aligned} sin(\theta) &\approx \theta \;\; for \;\; \theta \approx 0 \\ cos(\alpha) &\approx 1 \;\; for \;\; \alpha \approx 0 \end{aligned} \tag{4.13}$$

Thus giving the following:

$$\ddot{\theta} = \frac{\theta g k_6 k_7 - \Omega I_{Gz}\dot{\alpha}}{k_1 + k_6^2 k_7 + k_4} + \frac{\tau_{disturbance}}{k_1 + k_6^2 k_7 + k_4} \tag{4.14}$$

$$\ddot{\alpha} = \frac{I_{Gz}\Omega\dot{\theta}}{k_2} + \frac{\tau_{control}}{k_2} \tag{4.15}$$

Putting it in a state-space representation assuming that the outside disturbance is equal to zero, it looks like the following.

$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + B\boldsymbol{u}(t) \tag{4.16}$$

$$\boldsymbol{y}(t) = C\boldsymbol{x}(t) \tag{4.17}$$

$$\boldsymbol{x}(t) = \begin{bmatrix} \theta \\ \alpha \\ \dot{\theta} \\ \dot{\alpha} \end{bmatrix} \qquad \boldsymbol{u}(t) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \tau_{control} \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{gk_6k_7}{k_1+k_4+k_6^2k_7} & 0 & 0 & \frac{-\Omega I_{Gz}}{k_1+k_4+k_6^2k_7} \\ 0 & 0 & \frac{\Omega I_{Gz}}{k_2} & 0 \end{bmatrix} \qquad B = \frac{1}{k_2}$$

4.2.3 Deriving transfer functions

There are two transfer functions that are needed for the later control design, first the function between control torque $\tau_{control}$ and roll rate and secondly the function between roll-rate and precession angle. A more detailed calculation may be accessed in Appendix C.

The first transfer function is easily calculated by the transformation equation going from state space to transfer function [21].

$$\frac{\dot{\theta}}{\tau_{control}} = \frac{-I_{Gz}\Omega}{(k_1k_2 + k_2k_4 + k_2k_6^2k_7)s^2 - gk_2k_6 + I_{Gz}^2\Omega^2} \tag{4.18}$$

The other transfer function is derived from a Laplace-transform of equation (4.14).

$$\frac{\alpha}{\dot{\theta}} = \frac{(k_1 + k_6^2k_7 + k_4)s^2 - gk_6k_7}{-\Omega I_{Gz}s^2} \tag{4.19}$$

4.3 Control Design

4.3.1 Introduction

As previously stated in the gyro dynamics chapter, balancing the platform is possible in a passive manner; however only for a short while until the precession angle reaches 90 degrees. At that point there is no acting torque from the gyro around the roll axis resulting in the platform tipping. To make better use of the balancing dynamics an active controller must be implemented. Based on previous studies[16] the first goal in terms of control is to implement a PID-based solution to achieve a system capable of tackling disturbance torque like shifting center of mass or external forces. From the previous definition of stabilizing torque from a gyroscope it is easy to see that the precession velocity and angle dictates the amount of output torque around the roll axis.

A higher precession velocity will give more torque but over a shorter period of time since it will reach 90 degrees quicker.

Then comes the next question of to what extent the precession shall be controlled. Modeling and controlling the entire system by a voltage as input applied to a DC-motor is preferable as it gives full control of the entire cascaded loop. However this necessitates that the DC-motor be modeled in the equations and increases the complexity. To avoid this and to simplify things a pre-made controller integrated in a motor solution can be implemented, also known as a servo. With a servo the input could be either position, velocity, acceleration or torque instead which leaves the current and voltage control to the servo.

4.3.2 Cascaded PID

The idea behind the cascaded PID-controller is to first control the roll rate of the system, following a reference of zero i.e keeping it still. This however could be reached with a passive gyro precessing naturally but only for a while as mentioned before. Therefore a second requirement is needed and that is to keep the precession angle α at zero. With this configuration the system tries to balance with the gyroscope while trying to keep the precession at zero as this gives the most balancing torque on the roll axis.

In figure 4.8 below a simplified control loop is suggested, sensor noise and outer disturbances are omitted just to show the cascaded control aspect, however these are of course part of the control problem. As method of choosing suitable controllers, MATLAB's PID-tuner program was used for a more streamlined process. This in the end gave as a result two filtered PID-controllers in an error feedback setup following what can be seen in figure 4.8.

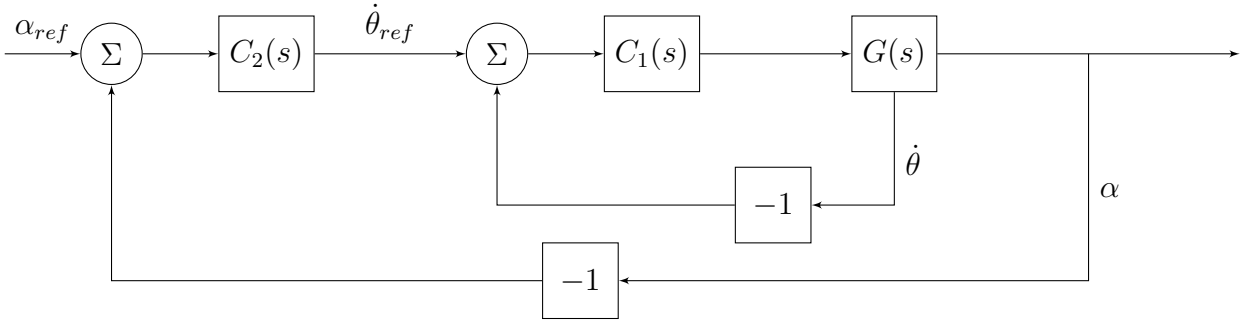


Figure 4.8: Suggested control algorithm

Discretization of the controller was done by the 'Euler forward'-method through tools in MATLAB with a sampling frequency of 100 Hz. The choice of sampling frequency

stems from the electronic devices mounted on the platform such as encoder and IMU which at the time seemed to not being able to handle higher frequencies. At the later stages of the project some improvements were made to increase the performance. The sampling frequency remained however since simulations showed stability and seemed like a reasonable time to let the control loop have a large margin of WCE (worst-case execution time).

4.3.3 LQR

Since the actual manufacturing time of the platform largely exceeded the expected thus pushing the final testing even closer to deadline, an effort to mitigate the risk of a failing control algorithm was made. Based on previous work [16], an LQR-based controller was designed and simulated as means of having a back-up in case of a failing PID. This incorporated reading all states of the system, i.e roll angle/velocity and precession angle/velocity. Choosing optimization variables Q, R were made with Bryson's rule [16]. This process was also made through the use of MATLAB's tools which solves the inherent Riccati equation of the system giving the optimal gain vector. The reason why this wasn't expanded to a LQG-controller which incorporates a Kalman-filter [22] is that all the states of the system are measurable and the expected sensor noise to be at a minimum. For example the chosen IMU which already has its own Kalman-filter.

4.4 Mechanics

This sections covers the mechanical structure that holds the double gimbal that will control the system and the design choices that has been made. The mechanical structure can be divided into two main subsystems. The outer cage, that controls the precession of the flywheel and the flywheel encasing that holds and controls the spinning of the flywheel.

4.4.1 Outer cage

The idea of the outer cage is to provide a stable frame, to hold the flywheel encasing and to be able to tilt around the x-axis.

Frame

The outer cage is designed using aluminum profiles (40x40 mm). This choice was made because the aluminum profiles provides good stability while still allows for modulation. To mount the aluminium profiles together, 90° brackets were used as seen in fig. 4.9b.

Tilting of the frame

As the end vision of the whole project (but not included in the scope of this report) is aimed towards a two-wheeled commuting vehicle, this behavior needs to be simulated. Therefore two wooden supports is placed on either side of the rig fig 4.9b. These supports allow the rig to produce a tilting motion and when upright, will be in an unstable position. Much in the sense of a inverted pendulum in 2D-space. The supports were constructed so that the total tilting angle were constrained to 30° on either side.

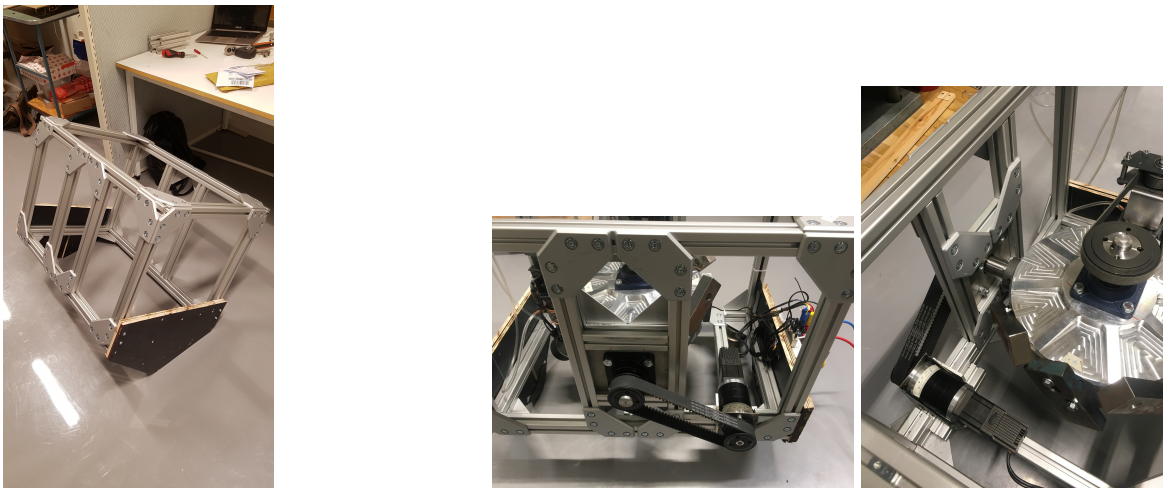
The supports will also provide safety during testing, both for personnel operating in vicinity to the rig but also to the rig itself, to prevent any damage caused by the rig falling over.

Precession of the gimbal

To produce the counter torque needed for the balancing, the controller need to be able to precess the flywheel. Constructionwise this comes in two steps. For one, the encasing needs to be mounted on bearings for the angular motion to be smooth enough. And secondly, this needs to be connected to an actuator, to produce the angular motion needed.

To mount the bearings onto the frame, two metal plates is produced which thickness matches the rails of the aluminium profiles. These can therefore be mounted into

the profiles acting as beams holding it in place, see fig 4.9c.



(a) (b) (c)

Figure 4.9: The frame that holds everything together with the supports that allows for the tilting motion of the frame (a). The transmission from the actuator to precess the encasing (b).

4.4.2 Flywheel encasing

Encasing

To be able to mount a drive and to precess the flywheel, an encasing was made. The encasing is also there to act as a first layer of protection, if something were to happen to the spinning disc.

The encasing was visualized in CAD and then cut out in a CNC machine. The material for the encasing, Alumec 89, was partly sponsored by *Uddeholm AB*. The encasing can be seen in fig. 4.10a.

On the top and bottom of the encasing, bearings was places that will hold and able the spinning motion of the flywheel.

On the sides, three sets of four holes were made to keep design choices open for were motor mounts and counter weights were to be put. Due to the properties of the aluminium material, helicoils was added to the screw holes to better support the screws and to prevent them from destroying the threadings when tightened.

Flywheel and flywheel axis

The flywheel itself was given by *KTH Combustion Engine Lab*. This was then the dimensional factor for the project. As everything later on had to be designed in accordance to the size and inertia of the flywheel.

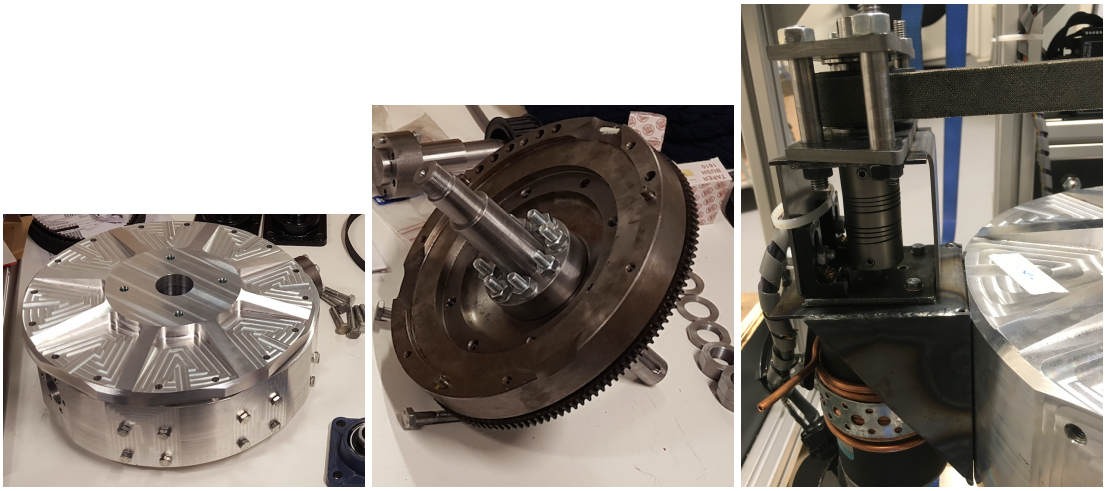
The shaft itself was lathed down to fit in bearings on the encasing, but was also to act as a hub for the fastening of the flywheel, see fig. 4.10b.

Precession beams in the encasing

To mount the two precession axis on the encasing, hubs were fastened with radial screws on the axis, which in turn could be screwed in place in the encasing.

Motor transmission

The motor was chosen to be mounted on the side of the encasing. This was firstly because of the size of the motor, which would be complicated to mount with a direct drive on the flywheel axis. But also secondly, the motor required additional gearing to overcome the friction from the bearings. Additionally, since the motor shaft was not designed to bear radial load, a transmission had to be made to support the driving shaft. The transmission shaft was supported by two additional bearings and between them and the flywheel axis, pulley wheels was mounted to drive the axis with belt a drive, see fig. 4.10c.



(a) The encasing (b) The Flywheel (c) Transmission

Figure 4.10: The encasing that holds the flywheel (a). The flywheel mounted on the flywheel axis (b). The transmission from the motor that spins up the flywheel, which is supported by two bearings, and geared 2:1 through pulley wheels with belt transmission (c).

4.5 Electronics

This section covers the required electrical design for interfacing the control software to the physical system. The electrical system is divided into three sub chapters covering, actuator, sensor and control unit.

4.5.1 Actuators

In the control system there are actuators driving, precession of the flywheel and rotation of the flywheel. The precession of the flywheel is driven by a planetary geared electrical motor. The use of an electrical motor was due to, the easy implementation and the already existing infrastructure; electrical power train. The gear is important to the motor, because the precession is a slow speed and high torque actuation. The other option was to use a stepper motor, but this was disregarded because the motor would need both additional gearing and an external driver.

The motor of choice was a $120Nm$ peak torque, $40RPM$ brushless dc motor from Crouzet, a suitable choice because of the complete drive-train; gearbox, motor and driver, as seen in figure 4.11. The motor is designed mainly for industrial applications such as conveyor-belts, heavy gates and controlled valves, applications similar to the precession motion.

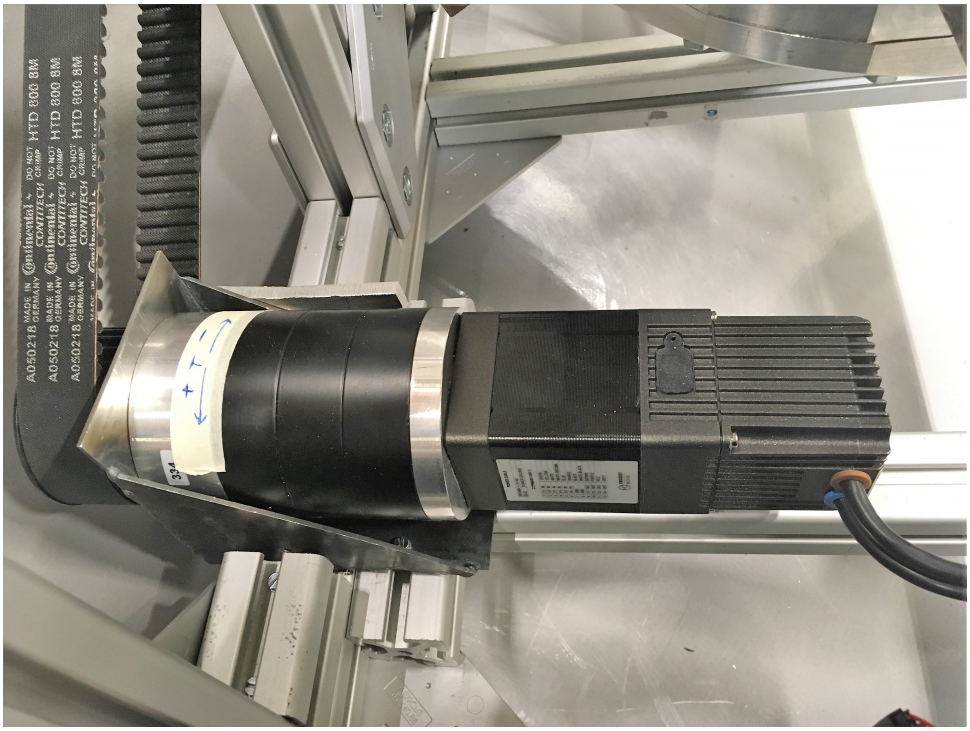


Figure 4.11: Geared brushless motor from Crouzet, used to precess the gyroscope. Picture: Crouzet.

The rotational speed of the flywheel is generated by a geared electrical motor; the mechanical system design is presented in the section on flywheel velocity. The electrical motor requires to achieve a flywheel speed of 2000 RPM, at a torque of 0.5 Nm to overcome to the friction in the bearings; this at a maximum voltage of 60 volts and preferred low current. The motor is also required to have a smooth torque to not generate vibrations to the flywheel. The solution to this was a permanent magnet motor with a suitable controller to produce a smooth continuous torque.

To choose a suitable motor a simplified motor model can be used to achieve approximate values. The following model describes the motor characteristics under steady state operation. The characteristic values K_v describes the back-electro-motive-force (BEMF), the constant K_t describes the current required for a specific torque and the value R is the resistance in the coil wires. All values are in root-mean-square values due to the use of a BLDC motor, therefore the maximum required voltage is the peak voltage of V_{in}.

$$V_{in} = \sqrt{2}(\frac{RPM}{K_v} - R\frac{\tau}{K_t}) \tag{4.20}$$

The equation links a higher rotational speed to a higher required supply voltage V_{in}.

The equations below relates the characteristic values K_v and K_t.

$$K_t = \frac{60}{2\pi K_v} \tag{4.21}$$

The equation shows how a large BEMF constant K_v leads to a small torque constant K_t, this leads to the intuitive voltage and current relationship. The power generated by a electric motor is the voltage multiplied by the current, therefore a smaller current requires a larger voltage. The table below exemplifies possible motors and their suitability for the flywheel application.

Motor	τ	RPM	V_{in}	K_v	K_t	R	I [A]
Reliance Motors	0.5	4000	46.2	120	0.08	0.5	6.25
TPPower	0.5	4000	22.6	250	0.038	0.2	10

In the table above two different motors are presented, both motors are reasonable for the application with one difference - the current consumption. The current consumption is a problem to many drivers because high currents causes larger losses and the need for more cooling on the driver power stages. The driver which was suitable for this application has a maximum current limit of 10 A and 72V, therefore the use of the Reliance motor was preferred, see figure 4.12.

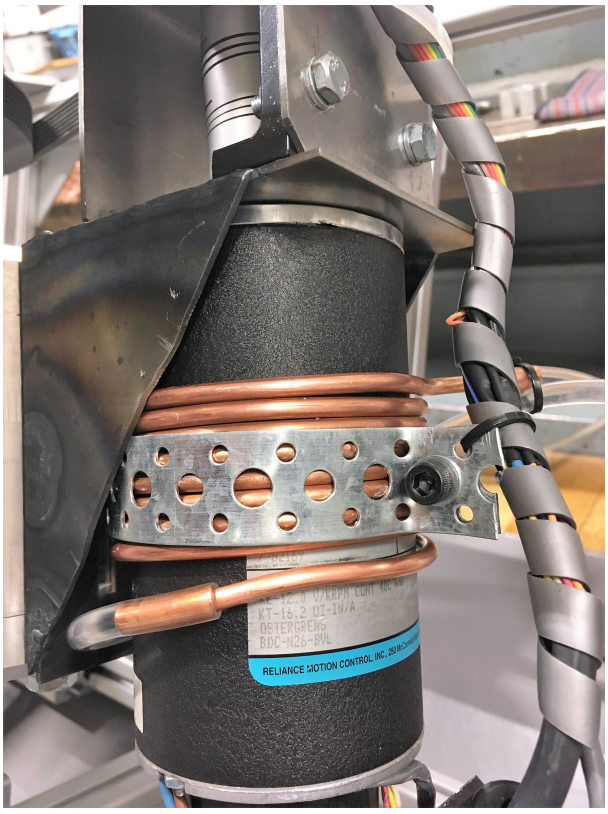


Figure 4.12: The velocity motor used with its additional water cooling system.

The motor required additional cooling which was achieved by attaching a copper cooling coil around the motor. Through the coil water was circulated which easily transported the heat away from the motor. The power loss can be approximated with the coil resistance and current flow.

$$P_{loss} = 3RI_{RMS}^2 \tag{4.22}$$

The simplified losses are then assumed to be no less than 59 watts. There are additional losses which is due to the motor not running optimally, and additional physical phenomena which takes place in the motor. The water cooling is intended to be able to remove twice the power approximated from the resistance.

4.5.2 Sensors

In the control section the control algorithms was presented, the control algorithms require four dynamic values of the system. The control algorithm requires precession, precession rate, roll and roll rate. To attain these values the an IMU and an encoder is used, the IMU collects information about the roll and the encoder monitors the gyros precession. The encoder can provide: the velocity which then can be integrated to the position or the position which can be derived to the speed. The optimal is for an encoder to both provide speed and position for this specific implementation.

The encoder can either be a simple pulse or absolute encoder. A pulse encoder uses the principle of a high or low output signal, and the position is determined by counting the pulses. This has a few drawbacks, first it cannot save a reference position and it also requires an interrupt which is occurring at a non-fixed rate. Benefits are the easy implementation and the resolution which can be several thousand of a turn. The absolute encoder has a different working principle and therefore remembers each specific position. Integrating the absolute encoder requires a communication protocol and dedicated hardware. This allows for a more efficient data transfer and a predefined timing, which leads to a more predetermined system. The encoder used for the stabilizing platform was from Netzer a DS-37, see figure 4.13.

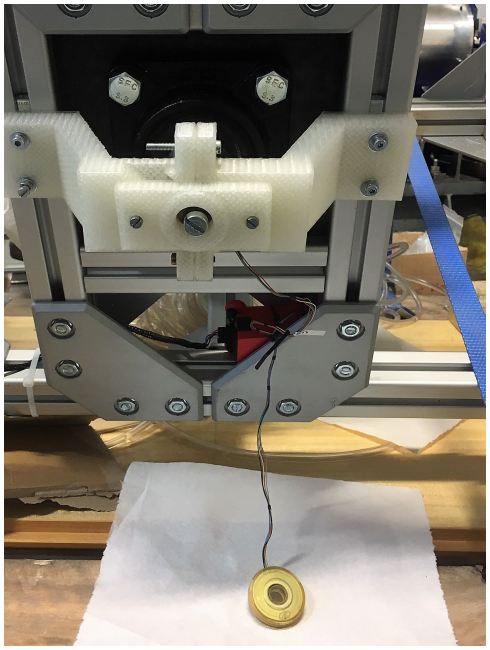


Figure 4.13: The encoder is mounted onto the precession axis, and holds in place by a plastic cover. The encoder is an absolute position encoder from Netzer.

Using an absolute encoder does have challenges, the sampling and response time of an absolute encoder can vary. This leads to an uncertainty in when the gyro was in a certain position. If the sampling is assumed to happen at the same frequency but is not, then the angle difference can be perceived as noise. In figure 4.14 it is clear how a time difference in sampling time gives rise to an error which is increasing with the rotational speed.

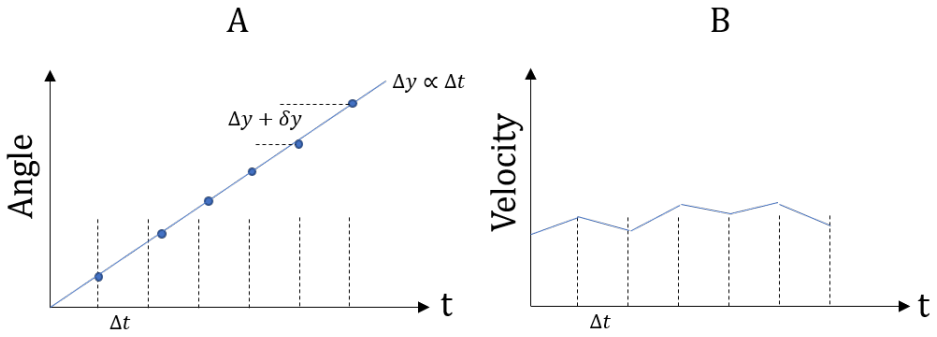


Figure 4.14: In plot A there is a constant angular motion which is sampled with a non-constant frequency. This leads to an obvious problem presented in plot B which isn't flat as the velocity of a constant speed should be.

The problem of receiving data with uneven intervals is clear if the position is derived

to receive the angular velocity. In the equation below, it is clear how the late sample leads to an error which grows proportional to the angular velocity. The delay in the measurement leads to the sample being larger or smaller by a δy. The size of δy depends on the velocity v and the delay δt.

$$\frac{\Delta y + \delta y}{\Delta t} = \frac{\Delta y}{\Delta t} + \frac{\delta y}{\Delta t} = \frac{\Delta y}{\Delta t} + \frac{v\delta t}{\Delta t} \tag{4.23}$$

The absolute encoder used in the newt project has a maximum delay of 0.1 milliseconds and a sampling time of 10 milliseconds. The gyro-cage on the newt platform can rotate with a maximum of 240 degrees per second. The error could then maximum be

$$\frac{2.4[deg]}{10[ms]} + \frac{0.24[deg/ms] * 0.1[ms]}{10[ms]} = 240 + 2.4[deg/s] \tag{4.24}$$

It is clear how the delayed reading can give varying errors. This method of measuring the speed and assuming a constant sampling time is not possible. A simple solution to this problem is to measure the actual time which elapse between samplings. This gives a $\Delta t'$ which is compensated for the delay and completely removes the problem. The only additional requirement is a clock to measure time. For position the problem is not as sever, the error in degrees can maximum be the velocity times the delay.

$$\Delta y = v(\Delta t + \delta t) \tag{4.25}$$

The position error is always corrected because the absolute encoder always keeps track of the absolute position. This could also be compensated for by monitoring the delay and compensating for it. By utilizing this clocking for the measurement the delays in the control-loop is not an issue.

The remaining information about the system is the roll and roll-rate which is sampled via the IMU. The IMU used is a "3-Space embedded IMU", it is made by Yost-labs, figure 4.15 and designed for multiple applications but with a top-frequency of 1700Hz sampling rate. From the system simulations a sampling of 250Hz is sufficient. The IMU has a filter and processing unit included which corrects for drift and noise.

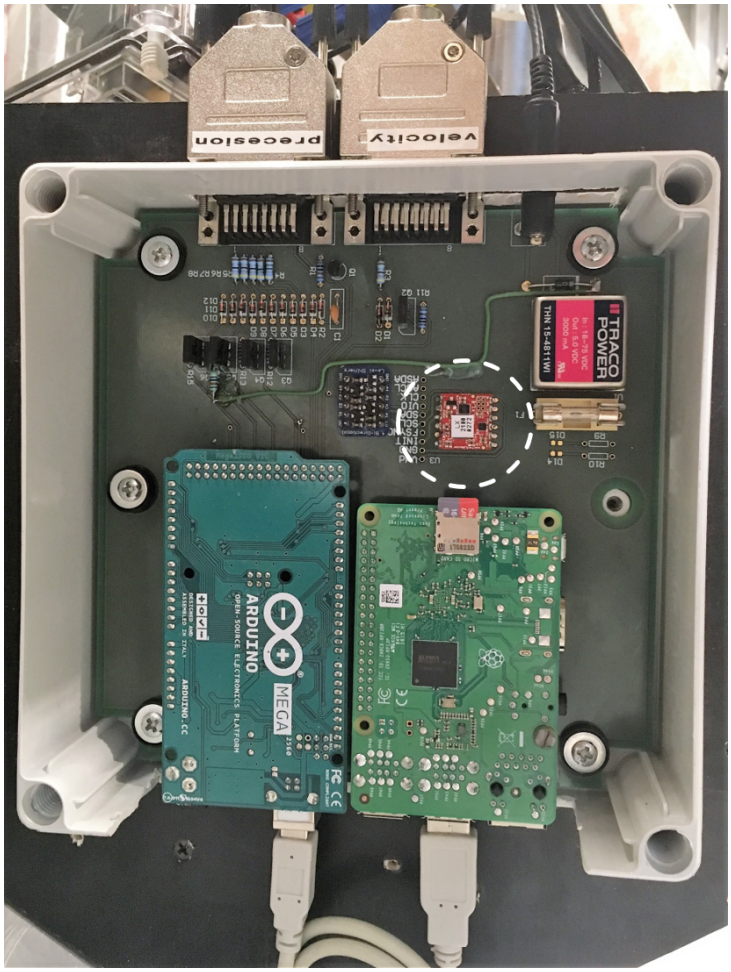


Figure 4.15: IMU unit located on the main board, the IMU is located inside the white dotted circle

The IMU is mounted directly to the main board, this results in a short distance for the communication to travel which improves the electrical signal integrity. By mounting the IMU to the board a good voltage source can be provided which improves of the systems robustness. The IMU requires a 3.3V communication and therefore a level converter was required. The level converter was designed by Sparkfun and had all the required bidirectional circuitry. Located to the left of the IMU, in figure 4.15.

4.5.3 ECU

The control-box holds the main computer, a control processor and additional circuitry, such as power supply and connection interfaces. The encoder sensor is not placed on the board, because it needs a mechanical connection to the precession axis. The encoder is interfaced via a D-Sub connector in the top of the board. The connectors also interfaces the precession motor and velocity motor with the control board. The power

to the system is connected to the top of the control-box and fused before entering the remaining part of the system, see figure 4.16. The precession motor has a 20A fuse and the velocity motor a 40A fuse, these should allow for required peak currents but limit any usage which can damage the remaining electrical system.

Figure 4.16: The closed control-box with main switch and power in and outlets at the top side.

The main power to precession and velocity motors can be disconnected by a manual switch located next to the power outlets on the control box. This is beneficial during testing to run the system but with actuators turned off.

4.6 Software

The software setup was designed to allow for access to and control over the machine while not requiring a computer to be tethered to it. The communication with the precession motor controller, the sensors and calculation of control signal was done on an Arduino Mega. A Raspberry Pi was setup to be able to flash new code on to the Arduino using a serial interface over USB. This connection also allowed for serial communication while running the machine. The Raspberry Pi was further set up to automatically connect to WiFi and over the network share it's internal memory as a network drive using Samba. This made it possible to connect to the Raspberry Pi via SSH and also copy compiled Arduino code over to it from another computer. An overview of this setup can be seen below in figure 4.17.

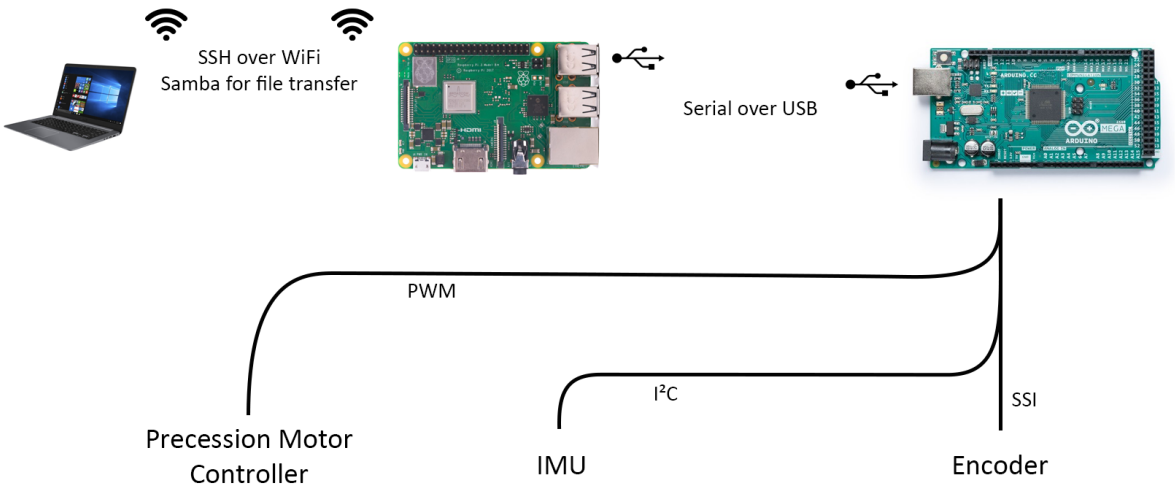


Figure 4.17: Communication setup, allowing control over the whole system wirelessly from a laptop or desktop computer.

The code for the Arduino was written and built using Atmel Studio to avoid any overhead introduced by the Arduino IDE. For the Raspberry a program to communicate with the Arduino while running was created using Python. This allowed for a more controlled start up sequence and logging of sensor data and control signal in real time.

Chapter 5

Validation & Verification

5.1 Simulation

To validate the controller designed, two nonlinear dynamic models were built, one using Simscape Multibody and one Simulink. Both were tested and compared, both for open and closed loop control. Both systems should behave similarly, in both scenarios and be able to stabilize the vehicle.

The Simscape model is the closest approximation to the physical system, as we don´t need to translate the equations directly into blocks, but the model will take care of the dynamic behavior once all components are linked together. The nonlinear model is built using Simulink blocks directly, representing the derived Lagrangian equations.

The most important parts of the Simscape model are shown below in 5.1 and 5.2:

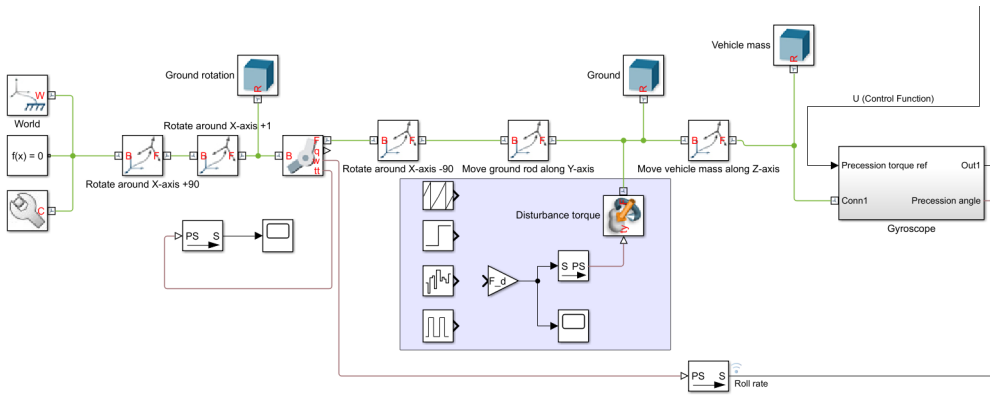


Figure 5.1: Testing platform simplified Simscape model showing the reference coordinate system, equation solver, ground and vehicle mass.

The purple area in the model is the simulink model for the torque disturbances to be

applied to the vehicle model, being possible to implement a ramp, step, white noise, and pulsating torque disturbances. The initial offset angle is set in the Revolute Joint block directly.

Inside the testing platform model, there is a gyroscope subsystem. The inside of the gyroscope subsystem is shown below in figure 5.2:

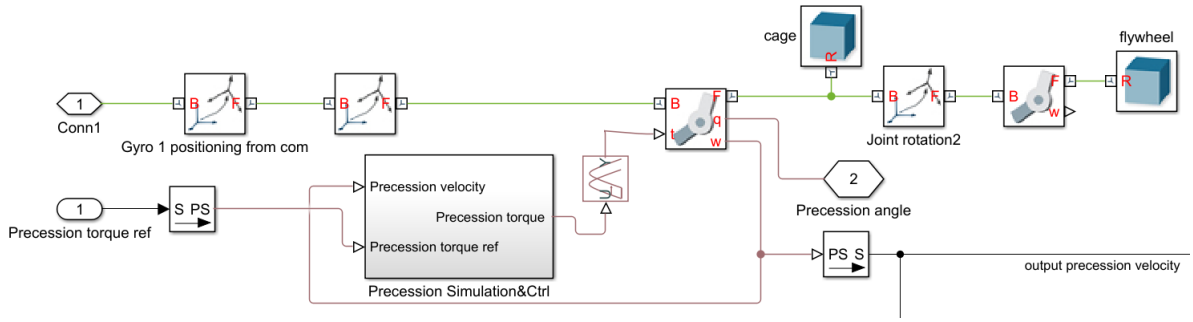


Figure 5.2: Gyroscope Simscape model.

Inside the gyroscope Simscape model, there is a motor subsystem. The inside of the motor subsystem is shown below in figure 5.3:

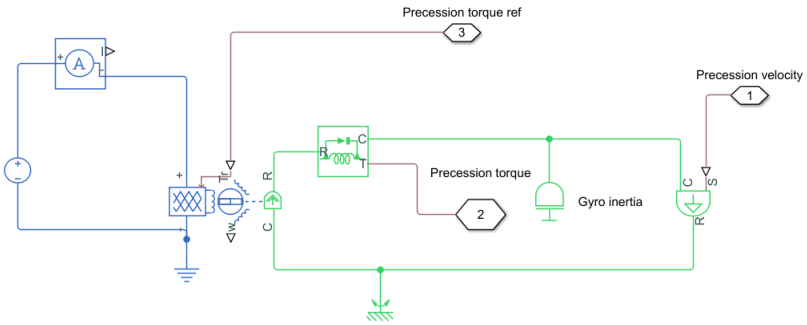


Figure 5.3: Motor Simscape model.

The motor is modelled after a simple servomotor controlled by torque input. The non-linear model is built based on the Lagrangian dynamic equations and built-in Simulink blocks, and is illustrated in the following diagram in figure 5.4.

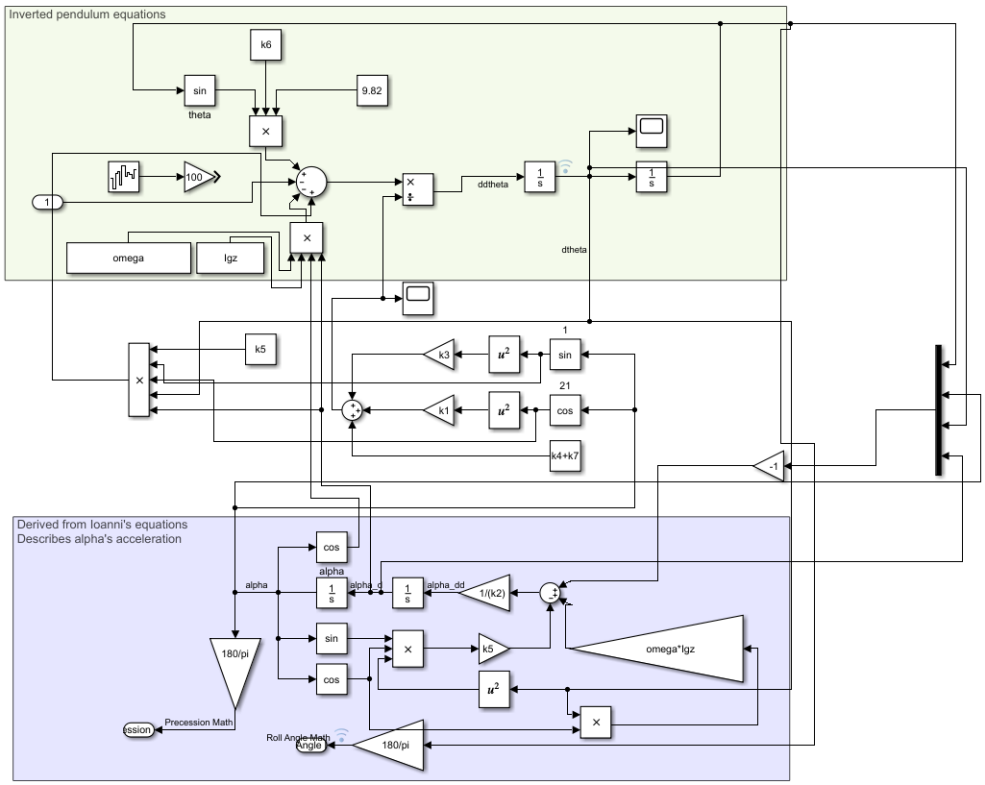


Figure 5.4: Motor Simscape model.

Both modelled systems (Simscape and Simulink) as depicted in figure 5.4 should behave similarly. Open loop simulations with an initial 5 degree and 20 degree offset in roll angle, i.e. the platform starts from a slightly tilted position, are run and results are presented in figures 5.5 and 5.6:

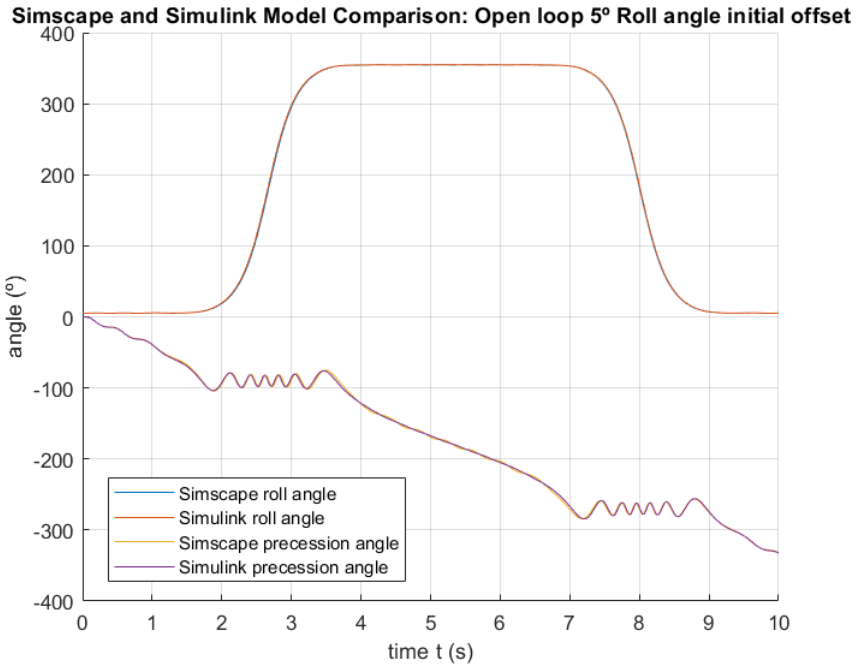


Figure 5.5: Dynamic behavior comparison for a 5° offset roll position

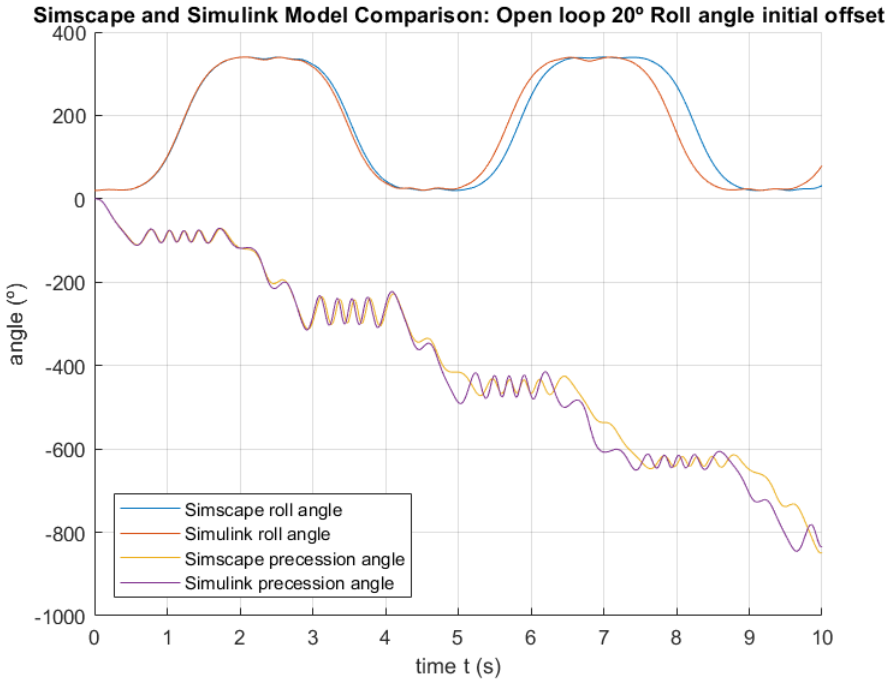


Figure 5.6: Dynamic behavior comparison for a 20° offset roll position

The parameters used for the following simulations are shown in the table below:

Gyroscope			
Variable Name	*MATLAB Symbol*	*Units*	*Value*
Gyro base center of gravity height	hb	m	0.4
Gyro base inertia	Igbx	kgm^2	0.051
Gyro flywheel inertia x Axis	Igx	kgm^2	0.094
Gyro flywheel inertia y Axis	Igy	kgm^2	0.094
Gyro flywheel inertia z Axis	Igz	kgm^2	0.172
Gyro cage inertia x Axis	Icx	kgm^2	0.6
Gyro cage inertia y Axis	Icy	kgm^2	0.7
Gyro cage inertia z Axis	Icz	kgm^2	1.1
Gyro center of gravity height	hg	m	0.4
Density	rho	kg/m^3	6700 (*)
Gyro outer radius (m)	R	m	0.17
Gyro inner radius	r	m	0.0475
Gyro mass	mg	kg	13.74
Gyro cage mass	mc	kg	37.123
Flywheel rotational speed	omega_rpm	rpm	4000
Flywheel rotational speed	omega	rad/s	418.879
Vehicle (Considers motor inertia and mass)			
Variable Name	*MATLAB Symbol*	*Units*	*Value*
Vehicle mass	mb	kg	50
Vehicle height	hb	m	0.4
Vehicle inertia	Ibx	kgm^2	8
Physical / Disturbance related variables			
Variable Name	*MATLAB Symbol*	*Units*	*Value*
Gravity	g	m/s^2	9.81
Sample time	Ts	s	0.010
Disturbance torque magnitude	F_d	Nm	170

(*) Density was modified to decrease the mass of the flywheel to match the one from the real flywheel.

Closed loop simulations with an initial 5 degree and 10 degree offset in roll angle, i.e. the platform starts from a slightly tilted position, are run and results are presented for cascaded PID control in figures 5.7, 5.8 and 5.9.

Cascaded PID Controller

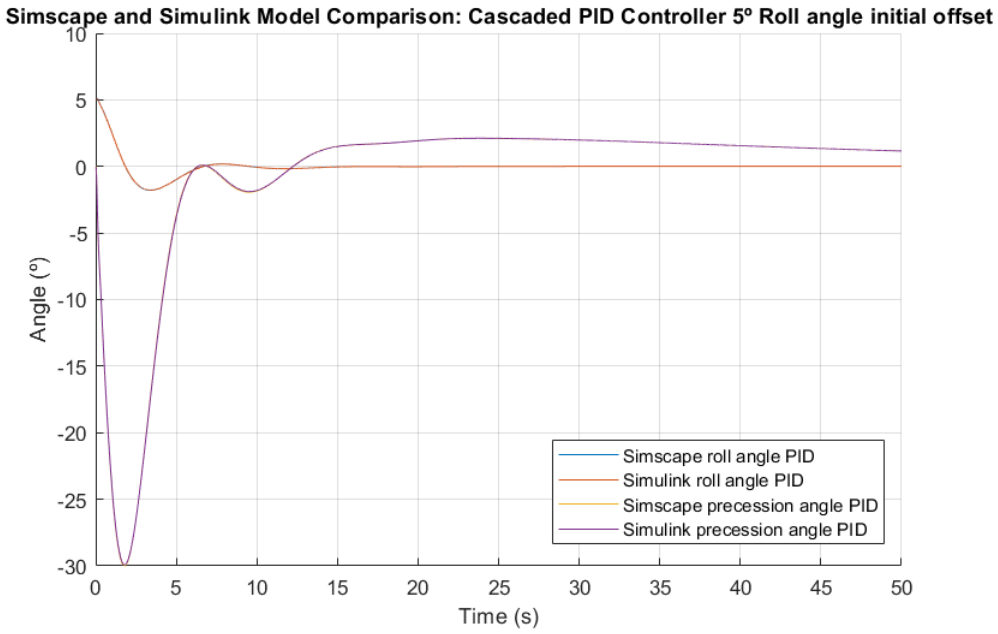


Figure 5.7: PID Control performance comparison for a 5º offset roll position

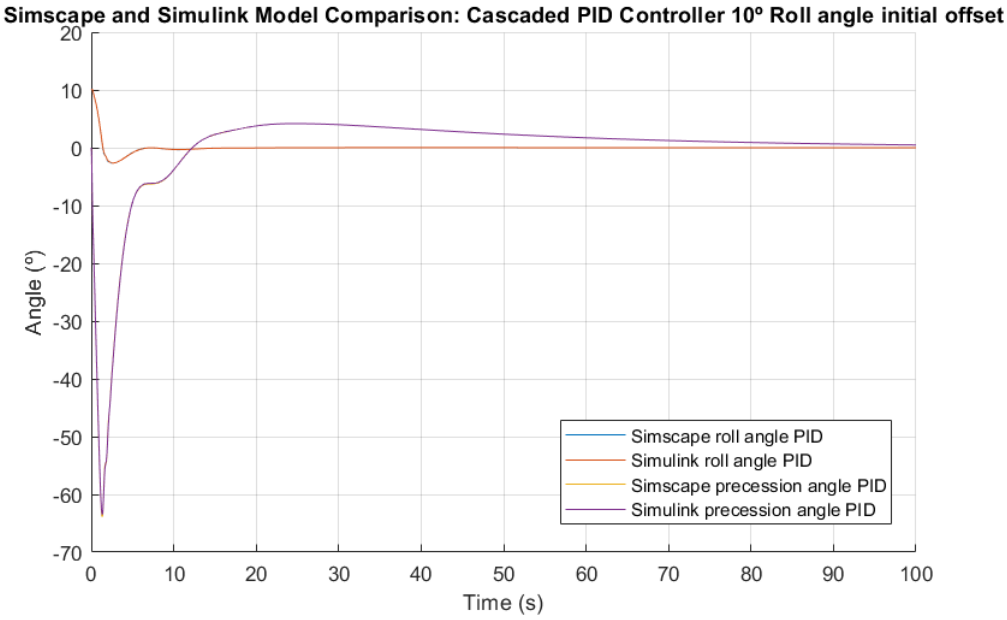


Figure 5.8: PID control performance comparison for a 10º offset roll position

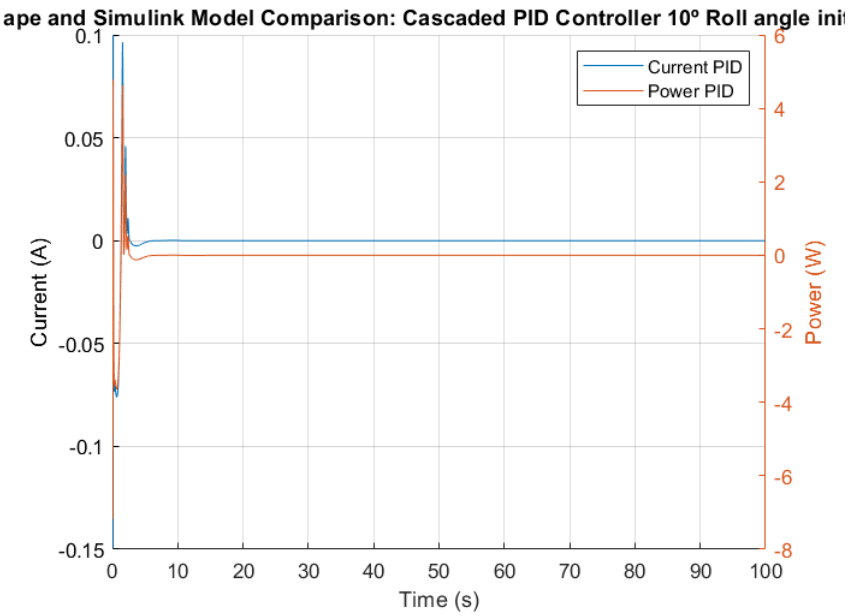


Figure 5.9: Simscape Cascaded PID precession motor current and power performance for a 10º offset roll position

The slightly different behaviors between the two models are most likely due to variables considered by Simscape and not by Simulink blocks. As explained previously, Simscape is a physical based modelling system, and it considers complex, more realistic dynamics of components used, while Simulink is a simplified block model. Variables like inertias and mass distribution, friction, efficiencies in mechanical and electrical devices, are set by default into the Simscape model, while Simulink does not.

The controllers obtained are then programmed into C functions for testing. First, Software In The Loop tests are run inside the models as MATLAB functions, translating the constants and variables properly when required into discrete time domain. Afterwards, Processor in the Loop test are run by using MATLAB Embedded Functions.

5.2 Ideal conditions

5.2.1 Model uncertainty & Disturbances

The designed and validated controllers must be tested for different conditions that will affect the real system:

- Time delays: time delay blocks can be introduced to simulate mechanical response, software or electrical delays in the real system.

- Sample time: it is desirable to have quick sampling time, however there is a minimum limit on the period between samples (due to the microcontroller clock frequency and code efficiency). maximum sample time may be tested for a specified disturbance.
- Motor response time: Real motors will have torque and velocity time constant. It is possible to make an estimate of the real motor time constants through testing. In the models, it is also possible to investigate maximum motor response time to withstand an specific disturbance.
- Noise: Sensors will never deliver an exact value on the variables being measured. There will always be a difference between the measurement and real value. Noises in the sensors may be simulated by adding white noise blocks in the feedback loops of designed controllers.

5.3 SIL & PIL

Designed controllers in Simulink must be converted into executable software, and later on, built into a microcontroller to be executed on the final prototype. Therefore, Software-in-the-loop (SIL) and Processor-in-the-loop (PIL) testing was done to investigate whether the controller, once translated into a language understandable by the microcontroller, performed well enough or not.

For the software in the Loop test, you may see the code implemented as a MATLAB C Function block in Appendix section 8.3. This same code is later implemented into a MATLAB S-Function to upload it directly into a microcontroller. Then the microcontroller runs the algorithm and results can be seen in the nonlinear model built in MATLAB. The SIL and PIL results are the same.

The board used for Processor in the Loop testing is the **Arduino DUE**, which is based on the Atmel SAM3X8E ARM Cortex-M3 CPU, with a 32-bit ARM core [23].

A standard 10 ms sample time is implemented into the discrete controller algorithm. Initially, no disturbances are imposed into the system (zero noise, 0.01 ms response time from the motor, and zero time delay). Results for PIL/SIL test for control performance with 5 degree offset and 10 degree offset initial angle are shown in figures 5.10,5.11,5.12,5.13,5.14 and 5.15.

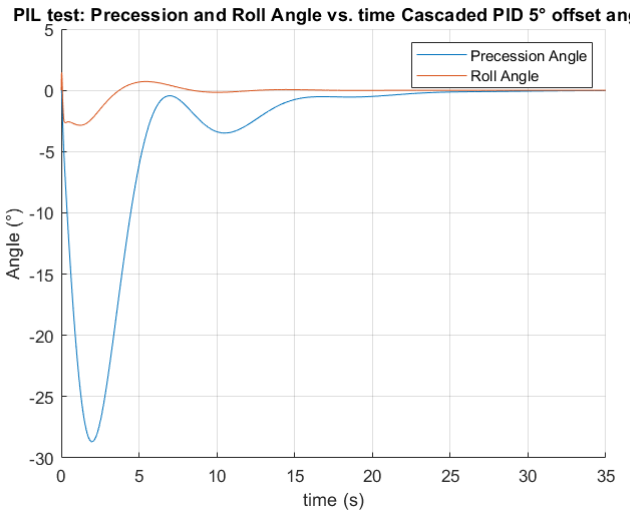


Figure 5.10: Angle position behavior in PIL/SIL controller performance test for a 5º offset roll position.

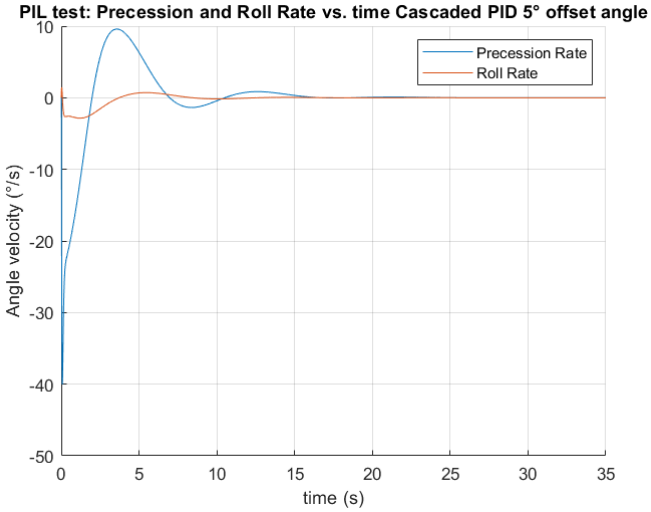


Figure 5.11: Angle velocity behavior in PIL/SIL controller performance test for a 5º offset roll position.

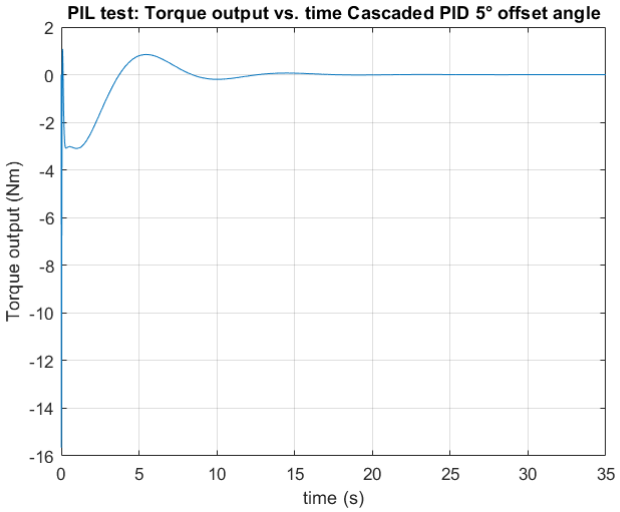


Figure 5.12: Torque output behavior in PIL/SIL controller performance test for a 5º offset roll position.

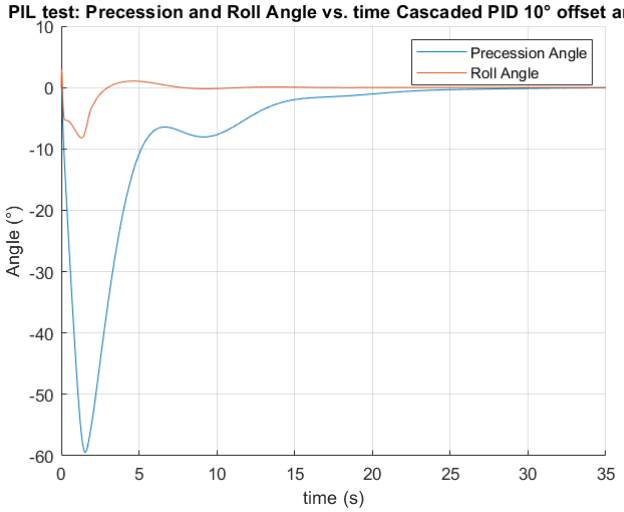


Figure 5.13: Angle position behavior in PIL/SIL controller performance test for a 10º offset roll position.

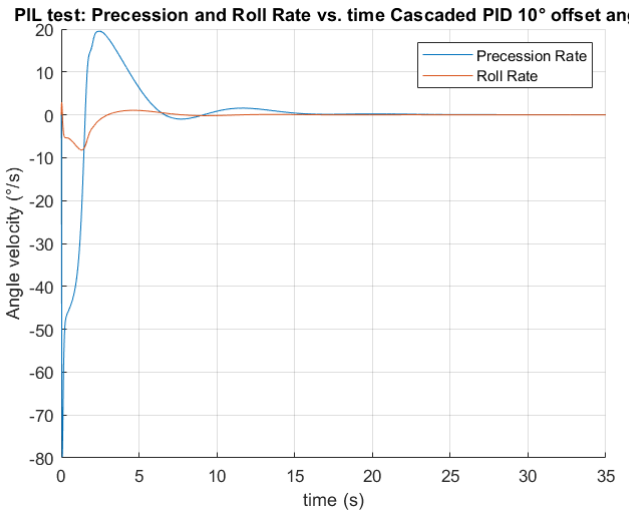


Figure 5.14: Angle velocity behavior in PIL/SIL controller performance test for a 10º offset roll position.

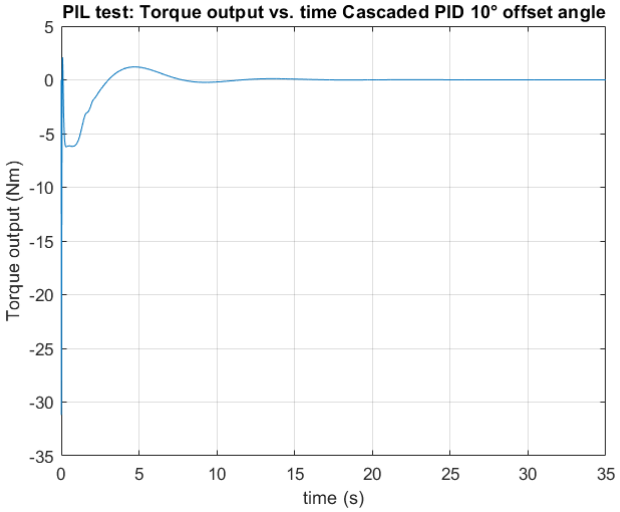


Figure 5.15: Torque output behavior in PIL/SIL controller performance test for a 10º offset roll position.

Observe that maximum precession angle is less than 60 degrees for the 10º initial offset angle, and then returns back to zero before 30 seconds. Notice also that the roll angle of the vehicle returns to zero in less than 8 seconds. Also, notice that the torque output from the motor, is, in the worst case, less than 35Nm (maximum continuous torque available in the real motor is 69Nm). These results show that the controller algorithm performs well in the microcontroller, and is ready to load into the test rig.

Adding noise with 0.00001 PSD (power spectral density) and same frequency as sampling rate, a 15 ms time constant for the motor torque, and a delay of 10 ms in control response, it is possible to obtain the following behavior presented in figures 5.16, 5.17 and 5.18 for a 10º initial offset angle. Observe that these parameters are arbitrarily chosen to reflect what is believed to be the worst case.

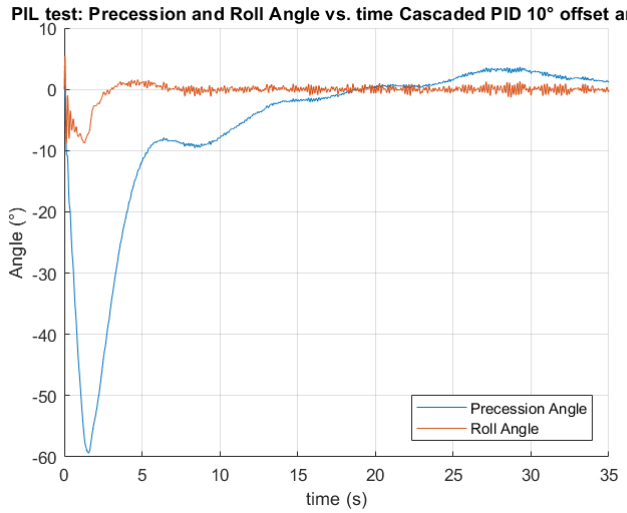


Figure 5.16: Angle position behavior in PIL/SIL controller performance test for a 10º offset roll position with disturbances.

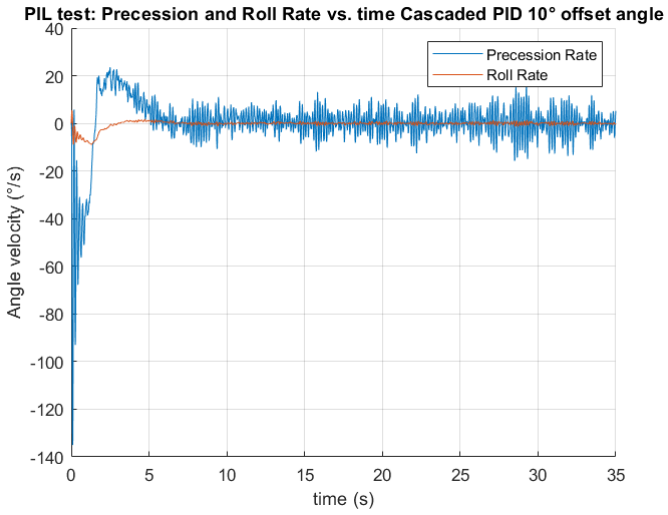


Figure 5.17: Angle velocity behavior in PIL/SIL controller performance test for a 10º offset roll position with disturbances.

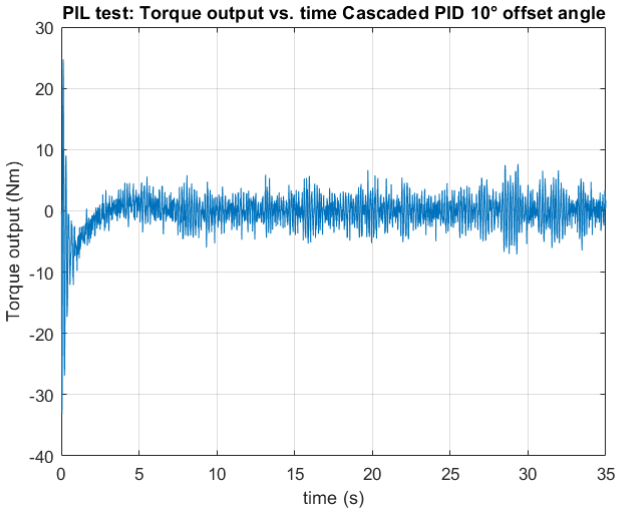


Figure 5.18: Torque output behavior in PIL/SIL controller performance test for a 10º offset roll position with disturbances.

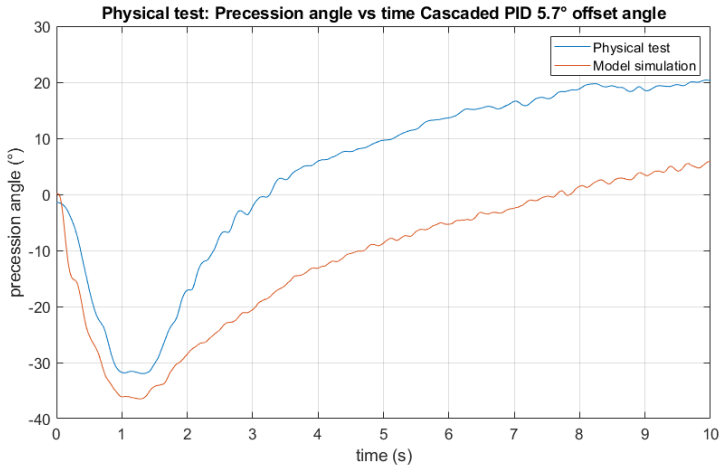
Behavior has changed from non-disturbed system, but still remains stable for the long run with an acceptable torque output. The controller is ready to be implemented in the final test rig.

Chapter 6

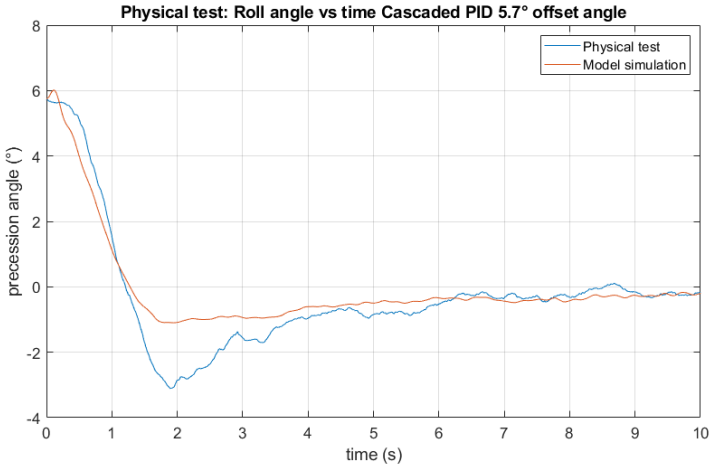
Integration and Final Testing

6.1 Tests & Results

Physical tests where performed mimicking the test used throughout creating the simulation model. The gyro was spun up and set at a precession angle near zero. The IMU's roll angle was zeroed at the balance point and the whole machine was then leaned against something to create a starting offset angle. Simulation using the model was then done at the same starting offset angle and this was done for both the Cascaded PID and the LQR controller. Comparative graphs can be seen in figure 6.1a, 6.1b, 6.2a and 6.2b below.

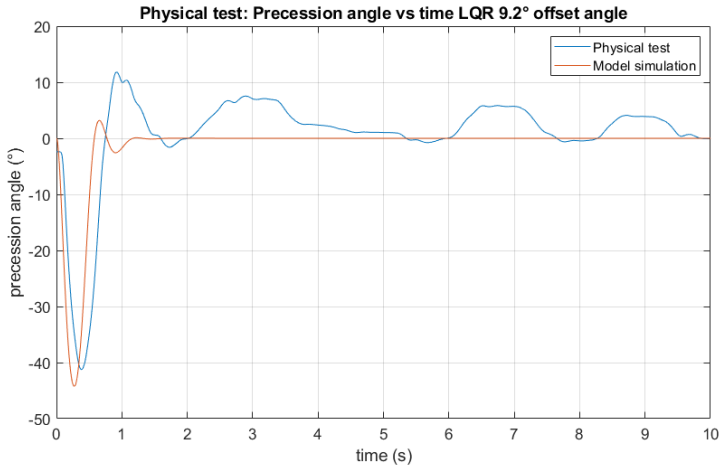


(a) Precession angle of the encasing with the PID controller

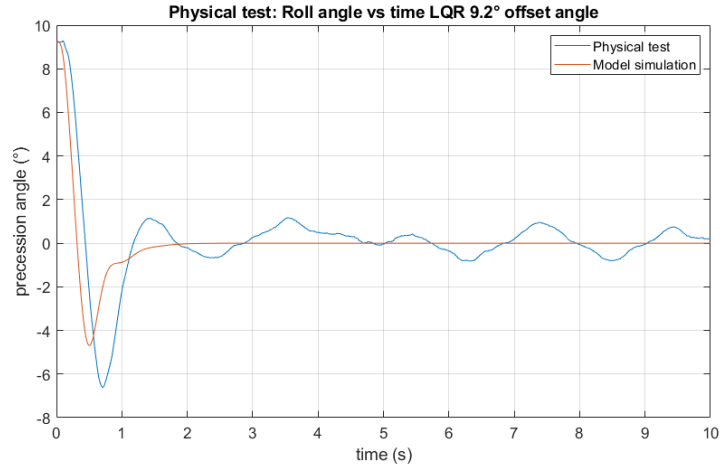


(b) Roll angle of the platform with the PID controller

Figure 6.1: The performance when starting at an inclined angle of $5.7°$ of the platform with a goal angle of $0°$. The performance is compared between the real case and the simulation.



(a) Precession angle of the encasing with the LQR controller



(b) Roll angle of the platform with the LQR controller

Figure 6.2: The performance when starting at an inclined angle of $9.2°$ of the platform with a goal angle of $0°$. The performance is compared between the real case and the simulation.

Comparing the test results with the simulated ones show slight deviation, this is probably caused by modelling errors owing to the fact that it is simplified, it should also be added that the sensors are not perfect and could deviate. The IMU for example has an orientation deviation of ± 2 degrees [24]. For the PID-controller test a strange phenomena emerges as the precession finds an equilibrium point while tilting. This was something that could not be solved during the project however still giving satisfying results as the platform could balance. The LQR tests showed more promising results with a quicker response, however still with smaller but deviating behaviour.

Chapter 7

Discussion & Conclusions

The goal of the Newt project is to develop a two wheeled vehicle, with the ability to balance autonomously. The intention with the this project was to design one possible wheel-based balancing system. This was not possible due to a tight time schedule and budget, therefore the self-balancing seesaw-box was designed instead. The time required to design a solution with two gyros was considered too large, which left the prototype using only one gyroscope. The time needed to manufacture all the required parts ended up limiting the design freedom. With the down-scaled final prototype, with one gyro and no wheels, the balancing functionality could still be proven and evaluated. The platform stands as a foundation for further testing and development. The aluminum profile structure simplifies possible changes and different control algorithms may be simulated and physically tested to verify and validate the functionality.

The initial request from the stakeholders was not delivered in the final prototype, this was due to unrealistic expectations on such a relatively short project. The final prototype included the essential parts required to evaluate the balancing system, a system which the stakeholders had proposed and tested in a earlier project [16] without success. The prototype was able to validate different load-cases which are likely to occur in the final wheeled prototype and validate models used for control design.

Chapter 8

Future work

8.1 Mechanical

A flywheel from an old car has been used during this project. The reason for using that flywheel was due to limited time and budget. The flywheel was rated for 6000rpm and was already balanced which greatly help the progress of the project. The consequence of that decision was that the hole prototype became bigger than necessary. Ideally, the flywheel should be custom made for the desired application, but to design a new flywheel a good knowledge in solid mechanics is needed, which is not an area of expertise of any person in this project. A hub motor should be used to reduce the footprint of the gyro and the encasing. The transmission between the velocity motor and the flywheel caused a lot of friction. These losses would be overcome with the use of a hub motor. To avoid yaw motion that will be introduced by the precession of the gyro, it is recommended to use two smaller gyros that are mechanically coupled. By using two flywheels, that mirrors each others movement, the undesired yaw motion of the flywheels will be cancelled out while still providing a roll motion to balance the vehicle. This is especially important for the water application. To be able to provide enough torque while making the balancing solution as small as possible emphasis should be on making the flywheels spin as fast as possible.

8.2 Control

The static tilting of the precession in the PID-control loop couldn't be solved within the set time frame of the project so this is something to further look into. The LQR-controller is working but more effort could be put into improving the performance by tuning the matrix weight parameters Q and R. The overall control problem is a highly non-linear one, using only linear control algorithms is therefore sub-optimal. Starting at higher offset roll angles will create problems with these methods unless some adaptive element is incorporated. The platform is now only standing still but mounted on a vehicle the controller must be expanded to be able to handle different states like varying velocities.

8.3 Electronics

With the knowledge of the final system design certain electrical design choices should be reevaluated. The velocity motor should be switched to be able to reach higher velocities, this would result in changing the controller to allow for at least 20-30 amperes of continuous current. The precession motor works well, but it is larger than required. This leads to additional weight which is unnecessary. The electronics in the control-box worked well but a redesign with isolated output stages and the ability to shift output signals between 3.3V, 5V and system voltage would be preferable. The encoder had problems with it's settings, something which indicated a malfunction of the part, it should be changed for a new one with the differential pair electronic interface located on the control board. For further improvement, the Arduino and Raspberry should have its components directly placed on the control board to minimize its size, shorten communication traces.

Bibliography

[1] F. A. Leve, B. J. Hamilton, and M. A. Peck, *Spacecraft Momentum Control Systems*, vol. 1010 of *Space Technology Library*. Cham: Springer International Publishing, 2015.

[2] N. Townsend and R. Shenoi, "Gyrostabilizer vehicular technology," *Applied Mechanics Reviews*, vol. 64, 01 2011.

[3] H. Takeuchi, K. Umemura, and S. Maeda, "Development of the anti rolling gyro 375t (rolling stabilizer for yachts) using space control technology," *Mitsubishi Heavy Industries Technical Review*, vol. 48, pp. 70–75, December 2011.

[4] M. Talha, F. Asghar, and S. H. Kim, "Design of fuzzy tuned pid controller for anti rolling gyro (arg) stabilizer in ships," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 17, pp. 210–220, September 2017.

[5] "Lit motors c1 apple." `https://newmotorspot.co/lit-motors-c1-apple/`. Accessed: 2018-12-15.

[6] D. K. Y. Kim and K. Bretney, "Electrical system for gyroscopic stabilized vehicle." United States Patent, Patent No.: US 8,918,239 B2, December 23 2014.

[7] J. Craig, "Gyro stabilizer refit for smoother sailing." Published at Yachtworld.com, August 2015. Accessed at Yachtworld.com 2018-12-15.

[8] W. M. Macek and J. D. T. M. Davis, "Rotation rate sensing with traveling-wave ring lasers," 1963.

[9] "Gyroscope sensors." `https://www.slideshare.net/ShereefShehata/gyroscopesensors`. Accessed: 2018-12-15.

[10] H. C. Lefèvre, *The Fiber-Optic Gyroscope*. Artech House, 1993.

[11] A. Noureldin, T. B. Karamat, and J. Georgy, *Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration*. Springer Verlag, 2013 ed., 2013.

[12] "File:centrale-intertielle missile s3 musee du bourget p1010652.jpg." https://commons.wikimedia.org/wiki/File:Centrale-intertielle_missile_S3_Musee_du_Bourget_P1010652.JPG. Accessed: 2018-12-15.

[13] "Xzn arduimu 9 degrees of freedom." https://www.geeetech.com/xzn-arduimu-9-degrees-of-freedom-p-535.html. Accessed: 2018-12-15.

[14] "Imu data fusing: Complementary, kalman, and mahony filter." http://www.olliw.eu/2013/imu-data-fusing/. Accessed: 2018-12-15.

[15] Wikipedia, the free encyclopedia, "3d-rendering of a gyroscope," 2006. [Online; accessed December 11, 2018].

[16] I. Karagiannis, "Design of gyro based roll-stabilization controller for a concept amphibious commuter vehicle," 2015.

[17] Wikipedia, the free encyclopedia, "Gyroscope flywheel diagram," 2006. [Online; accessed December 11, 2018].

[18] C. Gurrisi, R. Seidel, S. Dickerson, S. Didziulis, P. Frantz, and K. Ferguson, "Space station control moment gyroscope lessons learned," 12 2018.

[19] A. Brizard, *An Introduction to Lagrangian Mechanics: Second Edition*, ch. 2. World Scientific Publishing Company, 2014.

[20] A. E. Haas, *Introduction to theoretical physics*, p. 104. Constable Company ltd, 1928.

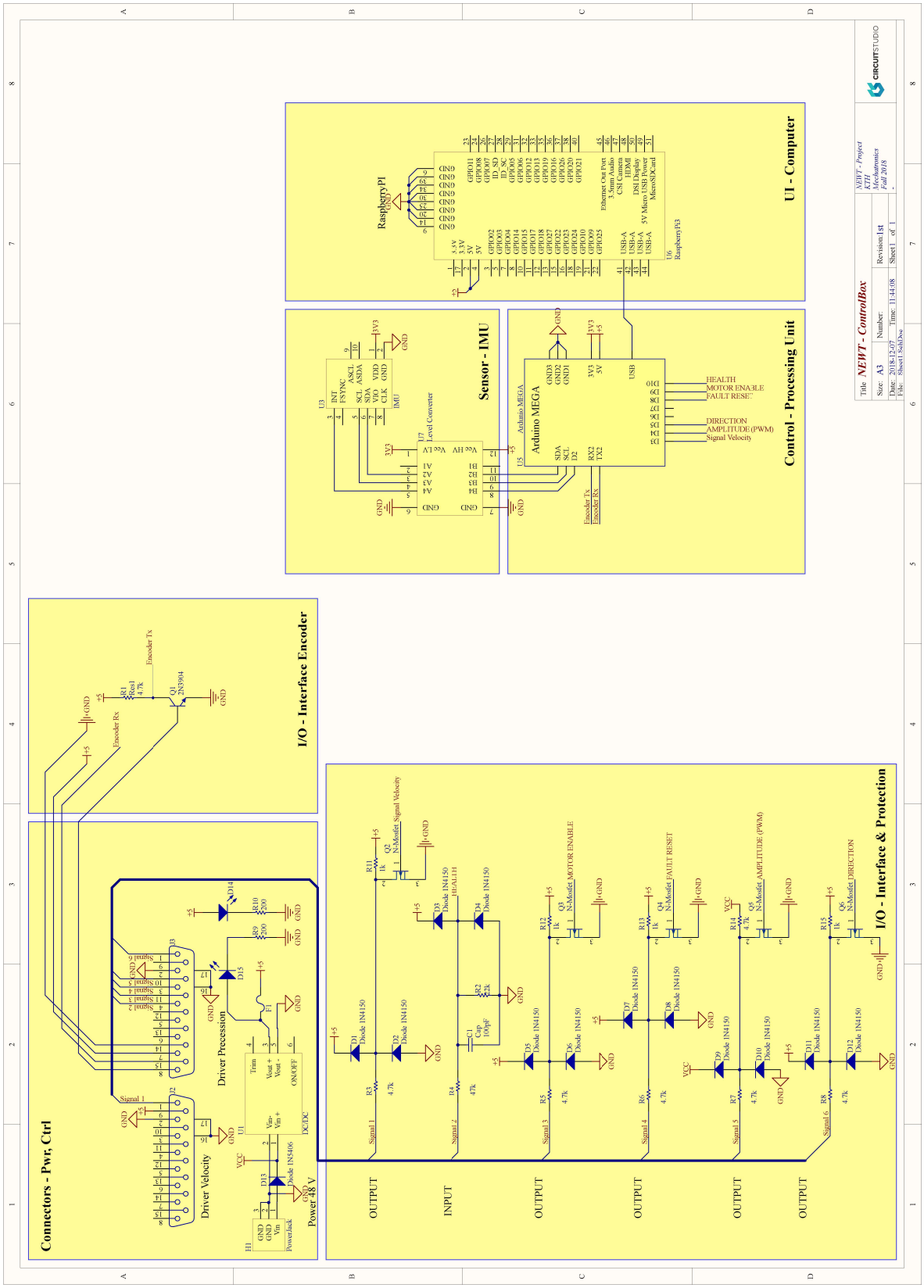
[21] E. Cheever, "Transformation: Transfer function state space," 2015. [Online; accessed December 11, 2018].

[22] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and design*, p. 359. John Wiley & Sons, 2001.

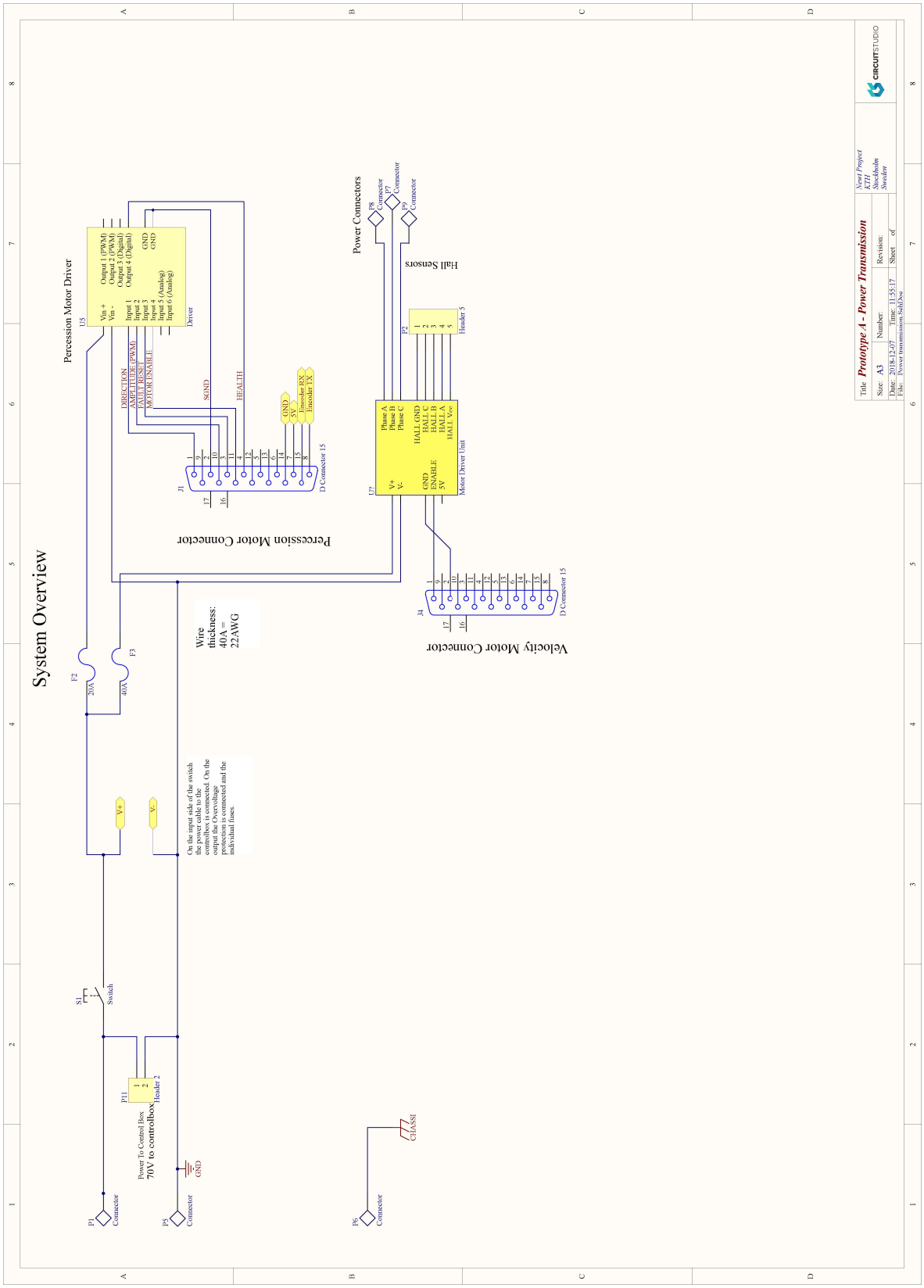
[23] A. Store, "Arduino store," 2018. [Online; accessed December 12, 2018].

[24] Y. Labs, *3-Space Sensor LX Manual*, p. 6. Yost Labs, Inc.

Appendix A


UI - Computer
Raspberry PI
Sensor - IMU
Arduino MEGA
Control - Processing Unit
I/O - Interface Encoder
Connectors - Pwr, Ctrl
I/O - Interface & Protection
Driver Precession
Driver Velocity
Power 48 V
OUTPUT
INPUT
OUTPUT
OUTPUT
OUTPUT
OUTPUT
Title: NEWT - ControlBox
Size: A3
Number:
Revision:1st
Date: 2018-12-07
Time: 11:44:08
Sheet:1 of 1
File: Sheet1.SchDoc
NEWT - Project
KTH
Mechatronics
Fall 2018
-

Appendix B


System Overview
Percession Motor Driver
Power Connectors
Hall Sensors
Percession Motor Connector
Velocity Motor Connector
Wire thickness: 40A = 22AWG
Power To Control Box 70V to controlbox
Motor Driver Unit
Header 5
Header 2
D Connector 15
Connector
Switch
Prototype A - Power Transmission
Power transmission SubDoc

Appendix C

Derivation of inner transfer function

$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + B\boldsymbol{u}(t) \tag{1}$$

$$\boldsymbol{y}(t) = C\boldsymbol{x}(t) \tag{2}$$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{gk_6k_7}{k_1+k_4+k_6^2k_7} & 0 & 0 & \frac{-\Omega I_{Gz}}{k_1+k_4+k_6^2k_7} \\ 0 & 0 & \frac{\Omega I_{Gz}}{k_2} & 0 \end{bmatrix} \qquad B = \frac{1}{k_2} \qquad C = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}$$

$$H_{inner}(s) = \frac{\dot{\theta}}{\tau_{control}} = C(sI - A)^{-1}B = \frac{-I_{Gz}\Omega}{(k_1k_2 + k_2k_4 + k_2k_6^2k_7)s^2 - gk_2k_6 + I_{Gz}^2\Omega^2} \tag{3}$$

Derivation of outer transfer function

$$f(t) = \ddot{\theta} = \frac{\theta g k_6 k_7 - \Omega I_{Gz}\dot{\alpha}}{k_1 + k_6^2 k_7 + k_4} \tag{4}$$

$$\mathcal{L}\{f(t)\} = \theta s^2 = \frac{\theta g k_6 k_7}{k_1 + k_6^2 k_7 + k_4} - \frac{\Omega I_{Gz}\alpha}{k_1 + k_6^2 k_7 + k_4}s \tag{5}$$

$$\frac{\theta}{\alpha} = \frac{-\Omega I_{Gz}s}{(k_1 + k_6^2 k_7 + k_4)s^2 - gk_6k_7} \tag{6}$$

However the inverse is needed as the precession is to be controlled by means of roll!

$$\frac{\alpha}{\theta} = \frac{(k_1 + k_6^2 k_7 + k_4)s^2 - gk_6k_7}{-\Omega I_{Gz}s} \tag{7}$$

But this is impossible as the expression becomes improper, therefore it needs to be multiplied with an integrator $\frac{1}{s}$.

$$\frac{\alpha}{\theta s} = \frac{(k_1 + k_6^2 k_7 + k_4)s^2 - gk_6k_7}{-\Omega I_{Gz}s^2} \tag{8}$$

Provided that the starting condition is equal to zero the following equation holds.

$$H_{outer}(s) = \frac{\alpha}{\dot{\theta}} = \frac{(k_1 + k_6^2 k_7 + k_4)s^2 - gk_6k_7}{-\Omega I_{Gz}s^2} \tag{9}$$

Matlab C Function Code for SIL

```
function [c,r,cin] = fcn(ref, prec, roll_rate)

persistent error_old1;
persistent error_old2;
persistent controlin_old2;
persistent controlin_old1;

persistent ref_inner_old1;
persistent ref_inner_old2;
persistent control_old1;
persistent control_old2;

if(isempty(error_old1))
    error_old1 = 0;
end

if(isempty(error_old2))
    error_old2 = 0;
end

if(isempty(controlin_old1))
    controlin_old1 = 0;
end

if(isempty(controlin_old2))
    controlin_old2 = 0;
end

if(isempty(ref_inner_old1))
    ref_inner_old1 = 0;
end

if(isempty(ref_inner_old2))
    ref_inner_old2 = 0;
end

if(isempty(control_old1))
    control_old1 = 0;
end
```

```
40
41 if(isempty(control_old2))
42     control_old2 = 0;
43 end
44
45 % Outer Loop
46 C_out_1 = -0.9568;
47 C_out_2 = 1.9120;
48 C_out_3 = -0.9552;
49 den_out_2 = -1.9615;
50 den_out_3 = 0.9615;
51
52
53 % Inner Loop
54 C_in_1 = 1233.1;
55 C_in_2 =  -2428.8;
56 C_in_3 = 1196.2;
57 den_in_2 = -1.0467;
58 den_in_3 = 0.0467;
59
60
61 error = ref - prec;
62
63 controlin = C_out_1*error + C_out_2*error_old1 + C_out_3*
     error_old2 - den_out_2*controlin_old1 - den_out_3*
     controlin_old2;
64
65 ref_inner = controlin - roll_rate;
66
67 control = C_in_1*ref_inner + C_in_2*ref_inner_old1 + C_in_3*
     ref_inner_old2 - den_in_2*control_old1 - den_in_3*
     control_old2;
68
69 %Update all variables:
70 error_old2 = error_old1;
71 error_old1 = error;
72 controlin_old2 = controlin_old1;
73 controlin_old1 = controlin;
74 ref_inner_old2 = ref_inner_old1;
75 ref_inner_old1 = ref_inner;
76 control_old2 = control_old1;
```

```
77 control_old1 = control;
78
79 c = control;
80 r = ref_inner;
81 cin = controlin;
```