Camera stabilization

LUDWIG NYBERG

MARCUS TJELLANDER
Abstract

A way to maintain a fixed position in three dimensional space is the use of a stabilization system. The use of this technology has greatly increased over the years and can be found in products like cameras and quadrocopters. This research focuses on a handheld gyro-stabilized mount where the purpose is to stabilize a camera from disturbances using combined methods, and to analyze its behaviour.

This report will describe the theory behind self-stabilization and construction of the prototype. Method for modeling, construction and testing for this problem will also be discussed.

The prototype is equipped with an IMU sensor that calculate the angles changes on the body in the space. The IMU consists of three accelerometer and three gyroscope, one for each axis to sense rotation and acceleration. Three motors are used, one on each axis for movement in the three dimensions and are controlled by an Arduino. The accelerometers and gyroscopes are used together with a PID controller to achieve a feedback-system.

Using both the accelerometer and gyroscope to complement each others disadvantages and a Kalman filter to filter out noise from the raw data, resulted in a stable PID controller and a functional stabilizer.
Sammanfattning

Ett sätt att behålla ett fast läge i det tredimensionella rummet är att använda ett stabiliseringsystem. Teknologin har börjat användas mycket mer de senaste åren och kan hittas i föremål som kameror och quadrocopters. Den här forskningen fokuserar på ett handhållt gyro-stabiliserad stativ, där syftet är att stabilisera en kamera från störningar genom kombinerade metoder och sedan att analysera dess beteende.

Den här rapporten kommer beskriva teorin bakom självstabilisering och konstruktionen av prototypen. Metoder för modellering, konstruktion och tester för det här problemet kommer också diskuteras.

Prototypen är utrustad med en IMU sensor som beräknar vinkelförändringar på kroppen i rymden. IMU sensorn består av tre accelerometerar och tre gyroskop, en för varje axel som känner av rotation och acceleration. Tre motorer används, en till varje axel för att styra rotationen i de tre dimensionerna, dessa styrs av en Arduino. Kombinering av accelerometern och gyroskopet tillsammans med en PID kontroll implementeras för att uppnå ett återkopplingssystem.

Användandet av både accelerometern och gyroskopet kompletterar varandras brister och ett kalman filter som filterar bort brus från radatan, resulterar i en stabil PID kontroller och ett fungerade stabiliseringsystem.
Preface

We would like to thank Simon Gärtner, Fredrika Kringberg, Marcus Olsson and Philip Pulli for providing assistance in the lab with both machining parts and guidance with programming and designing the prototype. The help has been more than what has been expected of them even during times when they have not been working. We would also like to thank Staffan Qvarnström and Tomas Östberg for helping us with acquiring parts and guidance on using the machines. Without anyone of them this project would not have been possible.
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>i</td>
</tr>
<tr>
<td>Sammanfattning</td>
<td>ii</td>
</tr>
<tr>
<td>Preface</td>
<td>iii</td>
</tr>
<tr>
<td><strong>1 Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.1 Background</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Purpose</td>
<td>1</td>
</tr>
<tr>
<td>1.3 Scope</td>
<td>1</td>
</tr>
<tr>
<td>1.4 Method</td>
<td>2</td>
</tr>
<tr>
<td><strong>2 Theory</strong></td>
<td>3</td>
</tr>
<tr>
<td>2.1 Control theory</td>
<td>3</td>
</tr>
<tr>
<td>2.2 Camera stabilization</td>
<td>3</td>
</tr>
<tr>
<td><strong>3 Demonstrator</strong></td>
<td>5</td>
</tr>
<tr>
<td>3.1 Software</td>
<td>5</td>
</tr>
<tr>
<td>3.1.1 PID controller</td>
<td>6</td>
</tr>
<tr>
<td>3.1.2 Internal measurement Unit</td>
<td>6</td>
</tr>
<tr>
<td>3.1.3 Filtering and error propagation</td>
<td>7</td>
</tr>
<tr>
<td>3.1.4 Simulink</td>
<td>8</td>
</tr>
<tr>
<td>3.2 Electronics</td>
<td>9</td>
</tr>
<tr>
<td>3.2.1 IMU - MPU6050</td>
<td>9</td>
</tr>
<tr>
<td>3.2.2 Motor</td>
<td>9</td>
</tr>
<tr>
<td>3.2.3 H-bridge DRV8833</td>
<td>10</td>
</tr>
<tr>
<td>3.2.4 Arduino Uno</td>
<td>10</td>
</tr>
<tr>
<td>3.2.5 Joystick</td>
<td>11</td>
</tr>
<tr>
<td>3.3 Hardware</td>
<td>11</td>
</tr>
<tr>
<td>3.4 Testing</td>
<td>12</td>
</tr>
<tr>
<td>3.4.1 Simulink</td>
<td>12</td>
</tr>
<tr>
<td><strong>4 Discussion</strong></td>
<td>14</td>
</tr>
<tr>
<td>4.1 Choice of feedback</td>
<td>14</td>
</tr>
<tr>
<td>4.2 Movement along the axes</td>
<td>14</td>
</tr>
<tr>
<td>4.3 Errors</td>
<td>15</td>
</tr>
<tr>
<td><strong>5 Conclusion</strong></td>
<td>16</td>
</tr>
<tr>
<td>5.1 Recommendations and future work</td>
<td>16</td>
</tr>
</tbody>
</table>
Appendix

A. Code parts
Abbreviations

DC  Direct Current
DOF  Degrees of Freedom
IMU  Internal Measurement Unit
ISP  Internally Stabilized Platforms
LOS  Line of Sight
MEMS  Microelectromechanical Systems
PCB  Printed Circuit Board
PID  Proportional Integral Derivative
PWM  Pulse Width Modulation
Introduction

1.1 Background

Stabilization mechanisms is a widely used tool in today’s industry\footnote{[1]}. The system can be found in many different areas and products, such as segways or quadcopters, where the purpose usually is to keep something stable or to isolate something from movements or vibrations. A simple explanation, is to maintain a fixed position or direction in space while outer forces are interfering. This project involves a camera-stabilization mount with a function to eliminate small vibrations and to smooth the movement of the camera.

1.2 Purpose

The purpose of this project is to create a stabilization system that will stabilize movement with three degrees of freedom. This goal of the project is to answer two questions. First one is more theoretically orientated and the second more practically.

- “How does the choice of feedback, gyroscope or accelerometer, impact the stabilization?”
- “How will the controller system react to movement along the three axis x, y and z?”

1.3 Scope

This project will look at three degrees of freedom. Rotation around the x-axis, roll, around the y-axis, pitch, and around the z-axis, yaw. A recreation of an existing system, but with some limitations. Since the main goal is to implement the system and study the behaviour and stability, the design will be kept as simple as possible while still making sure it is robust.
1.4 Method

The project will be divided into three major parts. Design and construction of the prototype, the control implementations and the testing. As for the design model, a 2D model is made for machining the parts. Since the test rig should have a simple body design, the construction of the body will be made in light metal for robustness and minimal weight. The control implementation is done through a PID-controller that uses the IMU sensors as input and sends the output to the motors to counteract the angular changes of the camera platform. To achieve an optimal controller for the system, simulation will be done and then verified by testing the PID-controller.
Theory

2.1 Control theory

Control theory expands where mathematics and methods works together to create a body with applied mathematics. The word control has two main meanings. The first one states that a device behaviour is guaranteed to react to a specific desire. The second one satisfies a physical or mathematical behaviour when actively testing or checking are used\(^2\). Most systems are based on feedback control where a control signal and a reference signal is compared through mathematical calculations. The relation between the first and second signal creates a circular argument where they influence each other\(^3\). Sensors, camera and thermometers are a few electronic parts with feedback implications.

Most innovations in autonomous systems relay on electronics but are rarely pure electronic. Communication systems and information processing like mobile phone and navigation systems, rely more on pure electronic but some systems are closely tied to mechanical parts\(^4\). Mechatronic is a way to design "helpful machinery"\(^4\) where subtasks work in synergy, be it mechanically, electronically, or by software. For control systems with increased number of degrees, a complexity where higher computing power and more sensors are interacting on a higher plain.

All variations for internally stabilized platforms (ISP) have a common goal, which is to hold or control the line of sight (LOS) of one object relative to space or another object\(^5\). It is not always the case to have a fixed LOS relative to an object depending on its task. Its primitive use is to prevent rotation for its own internal space but it is important to have the sensor controlled in a specific manner, depending on what the system is attempting to accomplish\(^5\).

2.2 Camera stabilization

Today camera stabilizing platforms are used in a wide variety of applications. Unmanned and manned aerial vehicles (UAV’s and MAV’s) with mounted camera systems, tracking of moving objects and autonomous navigation are just a
few examples[6]. The need for stable video are crucial in these kinds of environments and areas of use. Another application when this technique is used today is when filming movies, to get smooth movement and eliminate hand-held vibrations. The stabilization can be achieved in two ways, mechanically or electro-mechanically. The electromechanical option is preferred since it can be made smaller and faster[6][7]. The electromechanical system uses either DC motors or servos to counteract the rotation of the system. The system needs a sensor to determine the relative position of the camera, this kind of sensor is discussed in the section 3.1.2. There is a third option when trying to stabilize a video, and that is to do it purely in the software. This is a technique that is under development [8], due to this it will not be discussed in this report.

Simply explained, the stabilization is achieved using motors that counter the movement of the platform that is set to be stabilized. If the platform is tilted, the motor will counteract this movement by rotation in the opposite direction. This is where control theory comes in handy. If the motor needs to compensate for both small and large disturbances in an effective way it can not react to these two states in the same way. The system needs to compensate fast and accurate. This is why the system needs to slow down the motor rotation when it approaches the desired position, if not the result will be an overshoot. There are several techniques to achieve this. The most common one is to use a PID controller [9], the PID controller is further discussed in section 3.1.1. The purpose of control theory is to calculate an input that results in a desired output.
Demonstrator

3.1 Software

To be able to control the prototype, a software is needed. The software is divided into three codes. The Kalman filter, the Calibration and the main code. Both the Kalman filter code and the calibration code are open source codes from the Arduino community[10][11]. The main code calls the Kalman filter code on every loop to get the Kalman filtered data. Below, a pseudo code can be seen. This is the basic structure the code need to have. The loop has to preform calculations for both x and y axis. The IMU is not be able to provide sufficient data to stabilize the z axis, this is further discussed in section[1.3]

**Algorithm 1** Pseudo code representation

1: Initialize sensor;
2: Calibrate sensor;
3: Set desired angle for x and y axis;
4: loop
5: Read raw data from sensor;
6: Filter data;
7: Fuse data from accelerometer and gyroscope;
8: Convert data to Euler angle;
9: if Desired angle ≠ Read angle:
10: Calculate the error;
11: Compute the PID term using equation 3.1;
12: PWM = Constrain( -255 ≤ PID term ≥ 255);
13: if PWM < 0:
14: Rotate CCW;
15: else if PWM > 0:
16: Rotate CW;
17: end if
18: end if
19: end loop

In the following sections the main parts of the code will be explained in de-

|Algorithm 1 Pseudo code representation|
3.1.1 PID controller

A proportional-integral-derivative (PID) controller is a closed loop feedback algorithm, that takes an error input and returns an output that sends back in the algorithm until it satisfies a desired value. The controller in this case will be used to regulate the current to the motors. The proportional (P) part eliminates the error signal and works proportional to the difference between the error and the output signal. If the error in the input is large the control signal output will also have a large error. One problem is that when the error gets small, the P-part alone will never be able to eliminate the error entirely. This small remaining error is called a steady-state error.

To eliminate the steady-state error, an integral (I) part is used, which sums the error over time. This will make the steady-state error smaller, but while doing so the stability of the system will also decrease. Meaning that, for example, the overshoot and settling time will rise.

To compensate for the decreased stability, the derivative (D) part is implemented to shorten the overshoot and thereby hastening the recovery from disturbances\[1\]. It also helps correct for the phase lag caused by the I-part.

The Sum of the PID controller can be written as:

\[
 u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}
\]  

(3.1)

Where \( K_p, K_i \) and \( K_d \) are the tweaking parameters to the P, I and D-parts, \( e(t) \) is the error signal and \( u(t) \) is the output signal. These parameters are frequently tuned to achieve low Overshoot, rise time and settling time.

3.1.2 Internal measurement Unit

To be able to correct for movements of the camera in the three dimensional space, the system needs to constantly feed real time data to the controller in order to regulate the motors. In order to get this feedback an internal measurement unit (IMU) is used. The IMU is a small circuit board containing three accelerometers and three gyroscopes. More advanced IMUs exists but will not be used during this project. The accelerometers measures physical acceleration of the unit itself. This is done using small masses that is allowed to move freely along one axis each. The movement of the mass it then measured with a capacitive sensor.\[12\] The accelerometers makes it possible to measure movement along the axis.

As a compliment to the accelerometers, the unit also contain three gyroscopes. A conventional gyroscope uses a spinning wheel that is mounted in the center of three free collar rings. Due to Newton’s third law of motion the rotating wheel will conserve the angular momentum and therefore stay in the same position even though the collars are rotated. However this system is hard to make in such a small scale that it would be unsuitable for an IMU. Therefore a Microelectromechanical system (MEMS) gyroscope is used instead. The MEMS
gyroscope is a vibrating structure gyroscope that uses a tuning fork configuration. Two fork shaped masses oscillate in opposite directions, and the capacity between them are measured. When exposed to an angular velocity the Coriolis force of each mass also acts in opposite directions. This results in a capacity change that is proportional to the angular velocity\cite{13}. This allows for measurement of rotation around the axis.

### 3.1.3 Filtering and error propagation

Generally the accelerometer output signal from an IMU, contains a lot of noise and is sensitive to small external interferences\cite{14}. This can be a problem when using this output as a feed-back to a controller, since the entire system relays on that an accurate reading can be obtained. Even the small vibrations caused by the user holding the stabilizer, is enough to interfere with the sensitive accelerometers in the IMU.

The gyroscopes however are not sensitive to vibrations, instead they can suffer from a problem called drifting. Drifting is when the actual rotation and the read rotation from the gyroscope differ. Using a signal containing a small error as an control signal for the system will give an error in the output signal, this is called error propagation. When the input error propagates to the output, and then the output is used as a reference the error will continue to increase. When this occurs it is hard to get accurate readings of at what angle the IMU has rotated with.

![Figure 3.1: Raw and filtered X-data plot gathered from the Arduino software](image)
Both accelerometers and gyroscopes have their advantages and disadvantages. To get around this both can be used in combination with each other to provide feedback to the system. It is more complicated but will benefit the accuracy greatly\cite{14}. To achieve this, both output data from the two sensors needs to be merged, this is called data fusion. More of it later in section \ref{sec:4}.

One way to handle these problems is to use a filter. The two most common filters used for this type of applications are Kalman filter, and the complementary filters\cite{15}. The Kalman filter is an iterative filter that requires more computational power than the complementary filter and is more mathematically complex. Both filters will accomplish the same thing, they will fuse the data and by doing so keeping the errors small\cite{14,15,16}.

### 3.1.4 Simulink

Simulink is a multidomain simulation and model-based design for block diagram environment. The application is integrated with MATLAB, and is a helping tool for the PID tuner to achieve robust design and low response time. The software have an auto PID calculator for optimum values and can be set to adjust the given system. The implementation of the PID controller into the system will result in the following flow.
3.2 Electronics

3.2.1 IMU - MPU6050

The IMU GY-521 MPU6050 is a three-axes gyroscope and accelerometer (6 DOF) with 3V-5V operating voltage and 16 bit data output[18]. This will be placed under the camera mount and will measure the angle and acceleration of the camera mount. Limited by a small budget and time, a more advanced IMU with 9 DOF could have been an alternative.

![GY-521 MPU6050](image)

3.2.2 Motor

Two Micro Motors L149.6.21 and one L149.6.90 have been used. All three motors are brushed DC motors with gearboxes to increase the torque. These are not the optimal kind of motors and are discussed further in section 4. The reason for selecting said motors is that they where available at the time when
needed. They also had sufficient torque to hold the camera in position. Some data about the motor, taken from the data sheet can be found below.

<table>
<thead>
<tr>
<th>Detail</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal voltage</td>
<td>6</td>
<td>V</td>
</tr>
<tr>
<td>Length</td>
<td>36</td>
<td>mm</td>
</tr>
<tr>
<td>Ratio to :1</td>
<td>20.8</td>
<td></td>
</tr>
<tr>
<td>Nominal torque</td>
<td>2.5</td>
<td>Ncm</td>
</tr>
<tr>
<td>Speed at nominal torque</td>
<td>60</td>
<td>rpm</td>
</tr>
<tr>
<td>Current at nominal torque</td>
<td>85</td>
<td>mA</td>
</tr>
</tbody>
</table>

Table 1. Data sheet[19].

3.2.3 H-bridge DRV8833

H-bridge is a circuit that enables voltage to be applied across a load in either of the directions. The motors are driven by a DRV8833 Dual Motor Driver Board. Each bridge can handle two motors with a operating voltage 2.7 V to 10.8 V and peaks with a 2.0 A current[20]. Since each axis has one motor attached to itself, two H-bridges are necessary to handle all three axis. To save time and avoid chances of error in building one, buying two where the better choice. It also made it possible to keep the structure smaller since the machines used to make our own bridge are not precise enough to make small PCBs.

Figure 3.5: DRV8833 Dual Motor Driver Carrier[20]

3.2.4 Arduino Uno

The arduino uno SMD with microcontroller ATmega328 is a open source computer hardware, for controlling digital and analog devices. It operates at 5V, 14 dialog I/O pins where 6 are PWM output pins and clocks at 490 Hz[21]. In this case the pins where overclocked to be able to send PWM signals fast enough, the overclock was set at approximately 32 MHz. The Arduino’s 5 V is not sufficient to drive the motors, insted the Arduino is suitable to control the IMU and H-bridges, in other words handling input and output. An external power supply is used to power the motors via the H-bridge.

3.2.5 Joystick

A 2-axis analog thumb joystick[22] with a select button is used to control the z- and y-axis. It is mounted on top of the handle and can rotate the z-axis 180°
and move the y-axis 15° up and down. This is used to prevent the z-axis from drifting.

### 3.3 Hardware

The hardware has been designed to fit the motors and to be able to hold the camera. Due to this the design is not optimal, further discussed in section 4.

Figure 3.6: Profile view of the prototype.
3.4 Testing

3.4.1 Simulink

To be able to tune the PID controller MATLAB and SIMULINK has been used. To simulate the system a step response test has been set up. To begin with a low PWM voltage is sent to the motor, after a predetermined time a higher PWM voltage signal is sent to the motor. This results in a angle change of the IMU. When knowing both the input (PWM) and the output (IMU angle reading) signals, MATLAB can use the built in PID tuner to set the optimal gain values. The tuner then estimates the PID parameters for an optimal rise-, settling-time and overshoot. These parameters can be set by the user.
Figure 3.8: X-axis motor, Left: Tuned PID reference tracking. Right: Step response with fitted one pole plant. Both taken from the Matlab software. 

Figure 3.9: Y-axis motor, Left: Tuned PID reference tracking. Right: Step response with fitted one pole plant. Both taken from the Matlab software. 

<table>
<thead>
<tr>
<th>Parameter</th>
<th>X-axis</th>
<th>Y-axis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rise Time [s]</td>
<td>0.0143</td>
<td>0.00649</td>
</tr>
<tr>
<td>Settling Time [s]</td>
<td>0.0938</td>
<td>0.0427</td>
</tr>
<tr>
<td>Overshoot [%]</td>
<td>4.78</td>
<td>4.78</td>
</tr>
<tr>
<td>$K_P$</td>
<td>0.984</td>
<td>1.496</td>
</tr>
<tr>
<td>$K_I$</td>
<td>18.927</td>
<td>53.241</td>
</tr>
<tr>
<td>$K_D$</td>
<td>$5.949 \times 10^{-4}$</td>
<td>$4.115 \times 10^{-4}$</td>
</tr>
</tbody>
</table>

Table 2. PID controller parameters.
Discussion

During the project some compromises has been made. One of the major ones has been the design of the hardware structure. The structures compactness has been limited by the size of the DC motors. Since the motors was chosen early in the project, the design had to be adapted accordingly. This resulted in a bulkier structure than what was planned for.

4.1 Choice of feedback

The first research question "How does the choice of feedback, gyroscope or accelerometer, impact the stabilization?" that is mention in the section [12] is solved by combining the gyro and accelerometer where they complement each others disadvantages. When the IMU starts to tilt, the accelerometer for the three axis can not detect small values very accurately due to noise from small vibrations. However the gyro is much more sensitive in the rotation and will quickly detect when an axis start rotating. But while the gyro is accurate at small errors, it suffers from drifting and the error will increase overtime. So combining the two feedback options will result in a more accurate angle reading.

4.2 Movement along the axes

The second research question "How will the controller system react to movement along the three axis x, y and z?" where behaving as expected in theory. The acceleration forces for the corresponding axes when moving the arm parallel to the accelerometer, gave almost none or very small disturbances. The test was done by moving the arm vertically and horizontally while standing still. Tests like these are difficult to perform without a proper test rig, since the movement from the arm was not moving in a direct line, so no real measurement could been gathered or compared with. However the second test, when turning the axes gave a stronger reading for the angles when the raw and filtered data are compared. So the answer to the second research question is that the movement
along the axes have no impact on the system when exposed to external forces theoretically, but no data could be collected.

4.3 Errors

Mentioning in the demonstrator, another choice of motors and IMU would have improved the results. Brushless motors with higher speed and no voltage drop have a better response time and can react to angle changes with better precision without voltage loss that are distributed to all motors. The motors that were used had all a gearbox. This results in friction and since all the gears have margin of errors while produced a small gap is introduced into the system. This is a problem when adjusting for small angle variations since the motor will rotate but the gearbox will not be able to translate this rotation to the axis, due to the play between the gears.

Another problem that occurred was that if one motor aligns with another axis then its own, the motor will start to rotate. For example, if the y-axis motor aligns with the z-axis the IMU sends data to the controller that an angle offset exists. This will result in an output to the y-axis motor to rotate but since the y-axis motor is not aligned with the y-axis the IMU will not change its angle as it should. This will cause the I-part of the controller to gradually increase, resulting in a fast rotation in one direction. One way to solve this is to use a more advanced IMU, as discussed above. With three data streams for each axis a condition can be set so that the structure no matter how the motors are aligned it will be stable.
Conclusion

This study shows that both research questions can be answered, but are limited in some areas. The implementation for the first research question is achieved theoretically and combines the gyroscope with the accelerometer. Although the two can be implemented separately, combining both for this project works better. The answer to the second question gave almost no behaviour impact during testing, which is discussed in section 4. The gopro mount fulfilled its purpose as an ISP, that adds functionality to its user for various activities.

5.1 Recommendations and future work

For improvement of the stabilization, an alternative is to use an IMU with 6 DOF that also has 3 magnetometers. This solves the drifting problem on the z-axis, this will also give better precision on the other axis. The existing IMU with 6 DOF completed with the joystick prevents the drifting problem but limits the z-axis to be controlled rather than stabilized.
Bibliography


Appendix
Code parts

Code part containing the PID implementation for the x-axis, the y-axis is similar, however the gain values are different.

```c
// PID Controller code by Ludwig Nyberg & Marcus Tjellander 2017.
// Camera Stabilization
int updatePidX(int targetValue, int currentValue) { // compute PWM value
  int pidTerm = 0; // PID correction
double error = 0; // Reset error
  double Kp = 0.984, Ki = 18.93, Kd = 0.0005949; // PID gain values from simulink
  static float last_error = 0; // Saves the error between loops
  error = targetValue - currentValue;
  PTerm = Kp * error;
  ITerm += constrain(Ki * error, -255, 255);
  DTerm = Kd * (error - last_error);
  pidTerm = PTerm + ITerm + DTerm;
  last_error = error;
  return constrain(pidTerm, -255, 255); // PWM can only take values from 0 to 255
}

PWMX = updatePidX(desAngX, kalAngleX);

if (PWMX > 0) {
  analogWrite(Xpin1, 0);
  analogWrite(Xpin2, abs(PWMX));
}
else if (PWMX < 0) {
  analogWrite(Xpin2, 0);
  analogWrite(Xpin1, abs(PWMX));
}
```