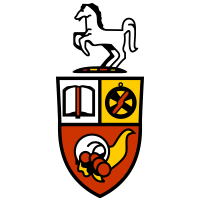
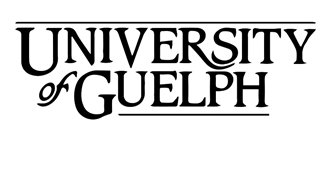
**ENGG\*4480 Advanced Mechatronics Design**

**Final Project Report**



A picture containing table

Description automatically generated

ENGG\*4480 Advanced Mechatronics Design

Wednesday, January 29th, 2020

Group Number # 7

Connor Fitzsimons (0990526)

Mirza Gowher Baig (0913336)

Ahmed Waheed (0926021)

Muhammad Azim (0938465)

Mohammed Al Fakhri (0982745)

Daniel Tadrous (0915887)

Table of Contents

[1.0 Summary 1](#_Toc36845153)

[1.1 Background 1](#_Toc36845154)

[1.2 Objectives 2](#_Toc36845155)

[1.3 Required Functionality 2](#_Toc36845156)

[2.0 Setup 3](#_Toc36845157)

[3.0 List of materials 6](#_Toc36845158)

[3.1 Specifications 6](#_Toc36845159)

[4.0 Methodology 9](#_Toc36845160)

[5.0 Preliminary Laboratory Ideas 11](#_Toc36845161)

[6.0 Preliminary Results 13](#_Toc36845162)

[7.0 Conclusion 14](#_Toc36845163)

[8.0 References 15](#_Toc36845164)

[9.0 Appendix 16](#_Toc36845165)

# Summary

## 1.1 Background

Gyroscopes are devices used to measure the change in angular acceleration. This angular acceleration can be treated as an input so that one or more motors can be used to control or maintain a set orientation of a platform that the device is attached to. This stabilization effect has multiple real-world applications such as a camera operator needing to maintain a level camera on a grip using a device called a gimbal as seen in Figure 1 below, or a cup holder in a car needing to stabilize itself from external forces to avoid spilling a drink.



Figure 1: A typical model of a Gimbal [1]

This lab will explore the use of an accelerometer as its functioning gyroscope. The device will always use two motors to control the platform keeping it parallel to the ground. This entire device will be mounted upon a small car which can then be used to simulate an external force when moved by hand. This sets up a control problem that can be solved using advanced mechatronics techniques.

## 1.2 Objectives

The main objectives of this lab are to get students familiar with sensor integration, programming an Arduino, system modeling, and controller design. By the end of the lab, students should be able to...

* Describe the platform stabilization control problem
* Understand how to derive the equations of motion for a system using Lagrange
* Importance of linearizing systems about set variables
* Describe the purpose and benefits of implementing a PID controller
* Know two different approaches to tune a PID controller

## 1.3 Required Functionality

The first half of this lab will deal with students assembling the device on top of the car using the provided kit. An Arduino is then further connected to act as a microcontroller to stabilize the platform along the x-axis and y-axis. The second half will be purely virtual, and the system will be modeled on MATLAB following the provided template. In the virtual set up a PID controller will also be implemented to tune and acquire better control of the system.

# Setup

Figure 2 below is a 3D CAD model to show how the setup of our project looks like. As shown in the CAD model below, the project is mounted on a car base which was borrowed from the Engineering Equipment Library. The wooden pieces, the metal rods and the plexiglass walls were cut and machined by the group in the machine shop.

A picture containing table

Description automatically generated

Figure 2: 3D Cad model of the project

A close up of a device

Description automatically generated

Figure 3: 3D Exploded View of the project

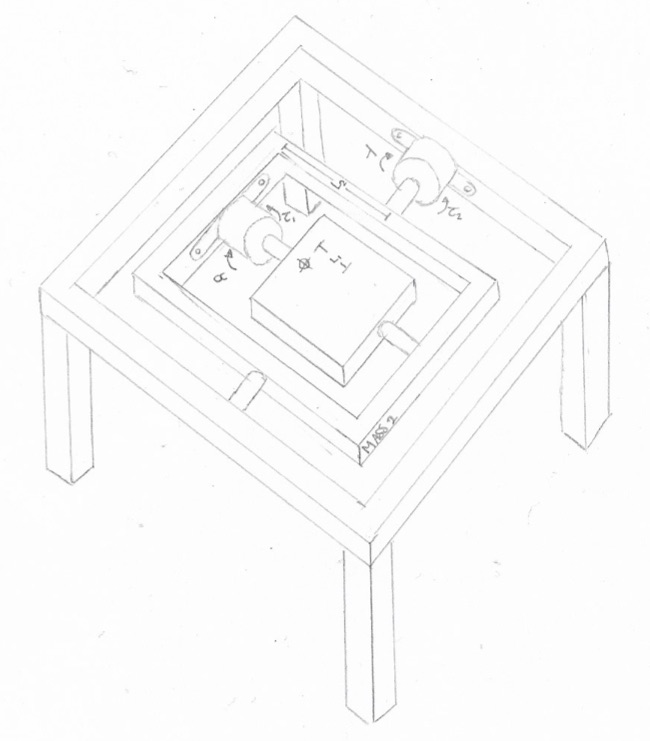


Figure 4: Sketch of the project

|  |  |
| --- | --- |
| Symbol | Definition |
| Alpha | Angle of Mass 1 with respect to horizontal |
| Lambda | Angle of Mass 2 with respect to horizontal |
| Mass2 | Mass of base mounting the second motor |
| Mass Motor2 | Mass of the second motor |
| IMass1 | Inertia of the platform to be controlled |
| IMass2 | Inertia of Mass2 |
| L1 | Displacement from center of platform to center of mass2 |
| L2 | Displacement from center of platform to center of mass motor 2 |
| g | Gravity |

Table 1: Variables and its representation

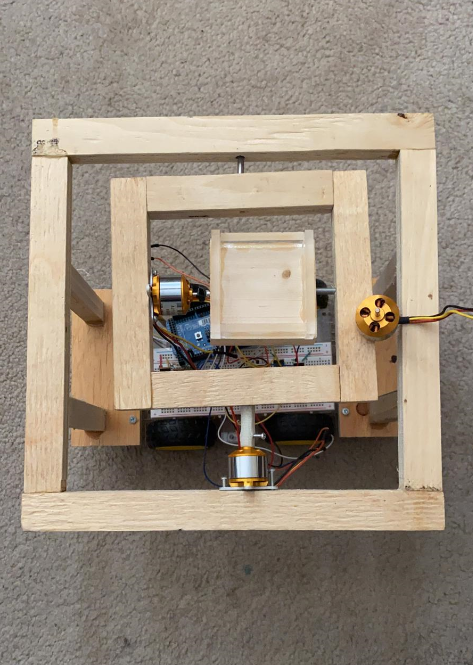


Figure 5: Actual build of the project

# List of materials

The following components were used to successfully build the self-balancing cup holder:

* 1 Bread Board
* 1 Arduino Mega 2560
* 2 Brushless Motor
* 1 IMU Sensor
* 6-P-Channel-Mosfet
* 6-N-Channel-Mosfet
* 1 Wooden Base
* Assorted wires
* 12 – 22 K ΩResistors

## 3.1 Specifications

The tables below provide the specifications of the Arduino Mega 2560, Brushless motor, P-Channel-Mosfet, N-Channel-Mosfet, and the IMU sensor used in the setup.

|  |  |
| --- | --- |
| Arduino Mega 2560 | |
| Microcontroller | ATmega2560 |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage (limits) | 6-20V |
| Digital I/O Pins | 54 (including 14 PWM output) |
| Analog Input Pins | 16 |
| DC Current per I/O Pin | 40 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 256 KB of which 8 KB used by bootloader |
| SRAM | 8 KB |
| EEPROM | 4 KB |
| Clock Speed | 16 MHz |

Table 2: Arduino Mega 2560 specification [2]

|  |  |
| --- | --- |
| A2212 Brushless Motor | |
| Model Number | H6450 |
| Operating Voltage | 4.8V – 6.0V |
| Max Efficiency | 80% |
| Max Efficiency Current | 4-10A (>75%) |
| No - Load Current @ 10V | 0.5 A |
| Current Capacity | 12A/60sec |
| Output Current | 30A |
| Size | 45mm × 24mm × 11mm |
| Weight | 25g |
| Operating Temperature | -20°C - 60°C |
| Direction w/ increasing PWM signal | Clockwise |

Table 3: Brushless Motors specifications [3]

|  |  |
| --- | --- |
| IRF9510 P-Channel-Mosfet | |
| Drain-Source Voltage | -100V |
| Drain Current @ 25°C | -4A |
| Drain Current @ 100°C | -4A |
| Drain Current Pulsed | -2.8V |
| Power Dissipation | 45W |
| Response Time | 10ns |
| Input Capacitance | 200pF |
| Output Capacitance | 94pF |
| Operating Temperature Range | -55 - +175 °C |
| Size (L X W X H) | 31.24mm X 10.67mm × 4.83 mm |
| Weight | 3g |

Table 4: P-Channel-Mosfet specification [4]

|  |  |
| --- | --- |
| RFP30N06L N-Channel-Mosfet | |
| Drain-Source Voltage | 60V |
| Drain Current @ 25°C | 32A |
| Drain Current @ 100°C | 22.6A |
| Drain Current Pulsed | 128V |
| Power Dissipation | 79W |
| Response Time | 15ns |
| Input Capacitance | 800pF |
| Output Capacitance | 270pF |
| Operating Temperature Range | -55 - +175 °C |
| Size (L X W X H) | 29.03mm × 9.9mm × 4.5 mm |
| Weight | 3g |

Table 5: N-Channel-Mosfet specification [5]

|  |  |
| --- | --- |
| IMU Sensor Gyroscope (MPU-9255) | |
| Supply Voltage (Min) | 2.4V |
| Supply Voltage (Max) | 3.6V |
| Offset Voltage @25°C (Max) | 3mV |
| Operating Temperature | -40 - +175 °C |
| Operating Current | 3.2 mA |
| Sleep Current | 8 µA |
| Operating Temperature Range | -40 - +85 °C |
| Size (L X W X H) | 3mm × 3mm × 1mm |

Table 6: IMU Sensor specification [6]

# Methodology

This self-balancing cup holder setup enables the user to have a better understanding as to how to create a platform stabilization system using basic system control theory and utilizing simple everyday materials to do so. This project is primarily comprising of two brushless motors, 1 6DOF gyroscope, and a microcontroller (see section 3 for a full list of materials and specifications). This relatively simple setup will use the x and y angular acceleration data collected from the gyroscope and feed the data into the Arduino microcontroller which then gives the sufficient commands to the motors to spin in a certain angle to keep the platform in one stable position. The gyroscope is used in many applications in today's world, such as video game controllers, smart devices (ex. Smartphones, tablets), aerospace flight operations, and automobiles racing industry. This project can give a big insight as to how the sensor integration works in unison with different kinds of motors. This project can also be simulated in Matlab, which will give future users an even greater dept in as to how simulations can be important in any future career prospects. Furthermore, future iterations of the project can provide students with a chance to control the position of a targeted object around a set-point in different scenarios similar to the self-balancing cup holder implementing a combination of PID and other controllers as well as a different range of sensors. This will help students develop even more in-depth knowledge in the field of mechatronics while learning advanced principles of system control incorporating sensor integration.

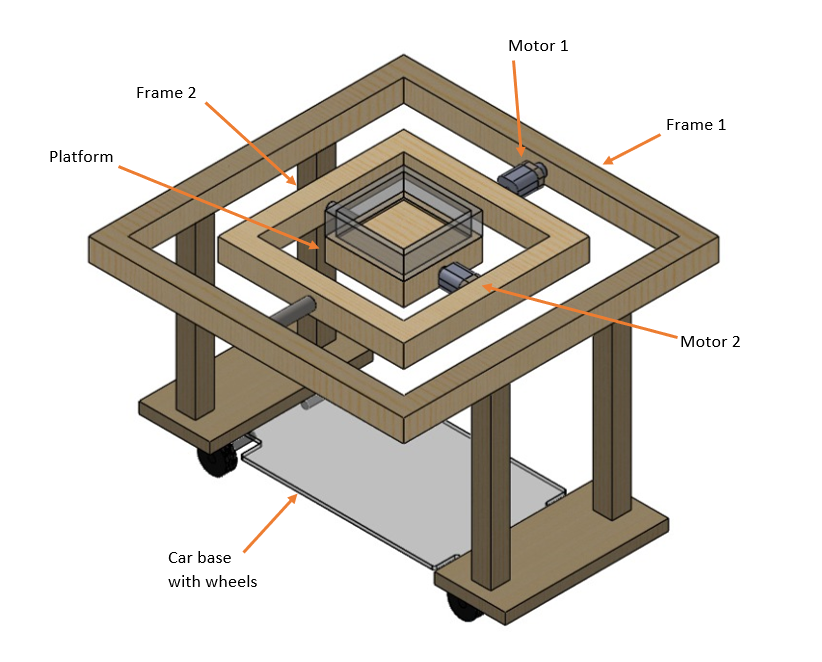


Figure 6: SolidWorks model of the project

As seen in Figure 6 above, the build is focused around a central platform which will be used to secure the cup. The center platform is connected to Frame 2 via a brushless DC motor 2. Frame 2 is connected to Frame 1 via brushless DC motor 1. The axis of rotation of both motors is perpendicular to each other to give the project the ability to manipulate the x and y-axis to stabilize the platform in between. This entire project sits on a car base with wheels in order to simulate a real-world situation, where the project will be attached to the cup holder. Furthermore, the project is also placed on the car base to give it mobility and to store the excess wires, microcontroller, and the 6DOF gyroscope.

# Preliminary Laboratory Ideas

**PID control implementation in Arduino**

As mentioned in section 4.0 above, future iterations of the project can provide students with a chance to control the position of a targeted object around a set-point in different scenarios similar to the self-balancing cup holder implementing a combination of PID and other controllers as well as a different range of sensors. The future students can learn how to create a customer PID controller based on their parameters and criteria. This can give them exposure to the purpose of implementing a PID controller in control systems. The students can also compare the system simulation with and without the presence of a PID controller in order to further understand the importance and role of a PID controller. This will in turn help students develop even more in-depth knowledge in the field of mechatronics while learning advanced principles of system control incorporating sensor integration and other features of mechatronics.

**Derivation of equations of motion using Lagrange**

The Lagrangian is a powerful tool used to acquire the equations of motion for a mechanical system, it is defined as follows...

L = Ksys – Usys

Where Ksys is the total kinetic energy of the system and Usys is the total potential energy of the system. Using this formula partial derivatives can be taken at each output (Alpha and Lambda) to calculate the torque needed to be applied to the motors as follows...

Tau = (d/dt(dL/dqdot)) - dL/dq

Use these formulas to derive the equations of motion for the system in terms of Tau1 and Tau2. The expression must only use the variables listed in the Table 1 above in Section 2.

*Instructor Note:* The solution to the two equations of motion can be found in Appendix A. The solutions were derived using MATLAB.

**Converting to State Space and Discretizing**

MATLAB is a powerful programming tool that can be used to model our system. Now that we have our equations of motion the system can be linearized so that it can be modeled in state space. Taking this state-space model and discretizing it will allow it to be used in MATLAB. Follow the given template to take these steps and acquire the state-space matrices.

*Instructor Note:* The template being refereed to is labeled in Appendix B and will be used for this section and the next.

**Tuning a PID controller on MATLAB using Zeigler-Nicholas Method 1 and 2**

Now that the system is set up it is time to model its output by first modeling the system under the effects of no controller. Inertia values for mass 1 and mass 2 can be calculated as those for a rectangular prism and a hollow rectangular prism. The inertia value for motor 2 can be calculated assuming the motor is a cylinder.

Next, implement a PID controller by following the Z-N method 1, and further fine-tuning the values to achieve a percent overshoot of no more than 10%. Repeat the same process using the Z-N method 2, once again further fine-tuning the values to reduce the steady-state error to zero. The provided MATLAB template’s setup presents a similar setup for linearization and simulation as the code provided from Lab 2 Winter 2020 [7] and Lecture 08 Winter 2020 [8].

# Preliminary Results

The device is controlled using the PID controller. The PID controller is used to help the device reach a steady state with correct damping at the right time. The device main purpose is to counter the motion of the water inside a cup when a vehicle is moving. The curve should be able to react smoothly to an impulse step signal and reach the steady-state with no oscillation in both axes. The output should look as shown in Figure 7. PID tuning method to be used is the Zigler-Nichols method of tuning. The inputs are the position detected by the filtered IMU sensor output, which tells the angle to the microcontroller. The microcontroller should then correct the angle by activating the motors in the right direction.

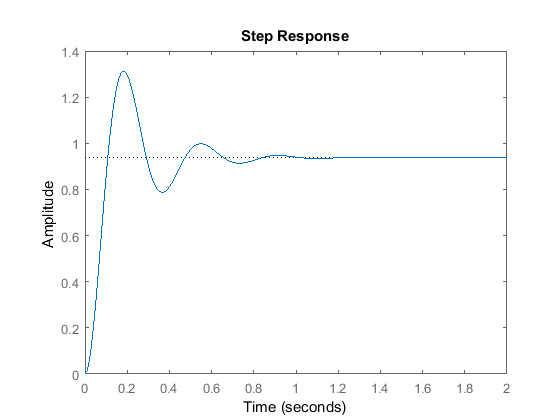


Figure 7: PID Step response output [9].

# Conclusion

In conclusion, the project covers range of the topics discussed in this course from system modeling to PID control and filters. Controlling brushless DC motors is a challenging part of the project. Tuning the PID and building the circuit will enhance the technical skills of the students trying to attempt this project. Building this project will allow the students to have a better exposure to position control and circuit building techniques.

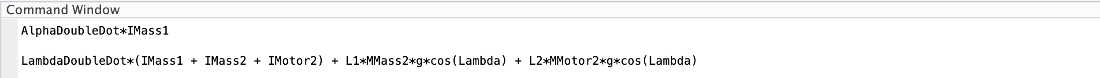
# References

|  |  |
| --- | --- |
| [1] | Aluratek, "3 Axis Handheld Gimbal for Smartphones," 05 2018. [Online]. Available: https://aluratek.com/pub/media/catalog/product/cache/e4d64343b1bc593f1c5348fe05efa4a6/3/-/3-axis-handheld-gimbal-smartphones-angle.jpg. [Accessed 03 April 2020]. |
| [2] | Arduino, "Arduino Mega 2560 Rev3. (n.d.).," [Online]. Available: https://store.arduino.cc/usa/mega-2560-r3. [Accessed 02 April 2020]. |
| [3] | ZJchao, "A2212/13T TECHNICAL DATA.," April 2017. [Online]. Available: https://www.rhydolabz.com/documents/26/BLDC\_A2212\_13T.pdf. [Accessed 02 April 2020]. |
| [4] | Sparkfun, "FQP27P06 60V P-Channel MOSFET.," May 2001. [Online]. Available: https://www.sparkfun.com/datasheets/Components/General/FQP27P06.pdf . [Accessed 02 April 2020 ]. |
| [5] | Sparkfun, "FQP30N06L 60V LOGIC N-Channel MOSFET.," May 2001. [Online]. Available: http://cdn.sparkfun.com/datasheets/Components/General/FQP30N06L.pdf. [Accessed 02 April 2020]. |
| [6] | I. Inc., "MPU-9255," [Online]. Available: https://stanford.edu/class/ee267/misc/MPU9255-Datasheet.pdf.. [Accessed 02 April 2020]. |
| [7] | Gadsden, A. , "Lab\_2\_Self\_Balancing\_Template.," 02 April 2020. [Online]. Available: https://courselink.uoguelph.ca/d21/le/content/605668/viewContent/2329049/View . |
| [8] | Gadsden, A. , "L08m.m.," [Online]. Available: https://courselink.uoguelph.ca/d21/le/content/605668/viewContent/2330291/View. [Accessed 02 April 2020]. |
| [9] | "Introduction: PID Controller Design," Controls Tutorials for Matlab & Simulink, 2016. [Online]. Available: http://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlPID. [Accessed 03 April 2020]. |

# Appendix

**Appendix A – EOM’s from LaGrange**





**Appendix B – MATLAB Template**

