

Development of an Active Gyroscopic Stabilisation System for a Motorcycle

James S Scott

Department of Physics
University of York
MPhys Project Dissertation
Supervisor: Dr. Phil Lightfoot

jss514@york.ac.uk

May 2017

Abstract

The aim of this project was to develop an active balance control system using gyroscopes that could be applied to motorcycles. A steel flywheel with a moment of inertia of $(19850 \pm 253)\text{gcm}^2$ was designed and manufactured for use in a control moment gyroscope to produce a restoring force to act against any destabilising motion experienced by the vehicle. A smaller flywheel with a moment of inertia of $(1270 \pm 1.733)\text{gcm}^2$ was used for initial trials, tuning and comparison. The active control system uses a Genuino 101 microcontroller with code optimised to the hardware to record all data, measure the angular position of the vehicle frame and determine a response using a PID algorithm. The PID algorithm required careful calibration of tuning variables to optimise system response, which is outputted via servo motors to control the gyroscopic precession angle and, by extension, the restoring force. Both passive and active control regimes were tested. The passive system failed to produce full stabilisation as expected, while the active system successfully stabilised a test vehicle with a moment of inertia of $(283810 \pm 253)\text{gcm}^2$ to stable equilibrium within (5.00 ± 0.19) degrees with the smaller flywheel, and within (1.11 ± 0.04) degrees with the larger flywheel. The active system also successfully implemented in a single track vehicle of $(4.25 \pm 0.05)\text{kg}$ for (3.797 ± 0.327) seconds.

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 3 |
| 2 | Theory | 5 |
| 2.1 | Gyroscopic Theory | 5 |
| 2.2 | Genuino 101 and Control Code | 6 |
| 2.3 | Circuit Design | 6 |
| 2.4 | Theory Of PID System | 8 |
| 3 | Methodology | 9 |
| 3.1 | Vehicle Test Frame | 9 |
| 3.2 | Designing The Flywheel | 10 |
| 3.3 | Measuring Inertia | 11 |
| 3.4 | Control Code | 12 |
| 3.5 | Passive System | 13 |
| 3.6 | Active System And PID Tuning | 14 |
| 4 | Results | 14 |
| 4.1 | Determination of Inertia | 14 |
| 4.2 | Small Flywheel Tuning | 15 |
| 4.3 | Passive System | 19 |
| 4.4 | System Performance With Larger Flywheel | 20 |
| 4.5 | Stabilisation Of The Bike | 21 |
| 4.6 | Dropout Identification | 23 |
| 4.7 | Positional Sensor Noise and Effect of Vibration | 25 |
| 5 | Discussion | 25 |
| 5.1 | Investigation Issues | 26 |
| 5.2 | Further Investigation Objectives | 27 |
| 6 | Conclusions | 27 |
| 7 | Appendix | 30 |

1 Introduction

Since the early twentieth century there has been work into gyroscope applications for a wide range of scenarios in different environments, offering a controlled method of influencing a body's dynamics, from satellites to luxury yachts. With a growing industrial interest in anti-roll stabilisation systems for land-based vehicles one solution is using a control moment gyroscope (CMG), which uses the moment of inertia of a flywheel to generate a restoring force. Controlling the orientation by applying an external torque to the output axis is also commonly known as an anti-roll gyroscope (ARG). Vehicle instability is a major issue for manufacturers and is a limiting factor in how vehicles and other structures can be designed, and commercial market leaders such as BMW are developing technologies to achieve self-balancing motorcycles [1]. Single-track vehicles offer a number of benefits, especially for areas with high population density suffering from traffic and air quality problems, which can be mitigated by smaller personal transport vehicles. However, single-track vehicles compared to cars are unsafe, due to other road users and weather conditions, and gyroscopic stabilisation could offer solutions to these [2].

A gyroscope consists of a solid (typically disc shaped) mass rotating with an angular velocity enclosed within an arrangement of two low friction gimbals. When implemented into a frame attached to one of the gimbals the degrees of freedom of the gyroscope is reduced. The restoring force produced by the flywheel acts on the frame to bring the system back to a level position. The gyroscope acts as a torque transducer, with applied torque from the electric motor in one axis and restoring force output from the flywheel in another [3]. The restoring force delivered by the gyroscope is dependent on the precession angle and the angular velocity of the flywheel. In this investigation the torque was varied by changing precession angle only.

There are two operational regimes of a CMG; passive and active. A passive system uses the natural restoring force generated by the spinning flywheel, with the gyroscope responding to a change in orientation. An active system manually controls the gyroscopic precession angle, dependent on supplied information of vehicle dynamics. The active system can be more effective than the passive system because it can offer a calibrated response to a change in position to bring a body back to equilibrium more efficiently, avoiding the angular momentum of the body moving past equilibrium on return. The active system can operate at a lower angular velocity, decreasing energy usage and increasing safety.

Active systems are more accessible than ever before due to the availability of small and relatively cheap computers in the form of microcontrollers. These handle lightweight computational tasks with the ability to interpret analog and digital signals, with a large array of compatible devices to suit any purpose. Previous limitations also arose from the additional weight from a mechanical drivetrain, requiring extra power [3]. This prevented a system being developed for single-track vehicles, which are traditionally used due to a reduced size and increased efficiency, but new electric vehicle technologies are increasing energy storage and transfer options.

The earliest notable effort of gyroscopically stabilized vehicle was around 1904, when Louis Brennan registered a patent for his idea of implementing one or more CMGs to maintain the attitude of single-track, marine and aerial vehicles [4]. To control the torque supplied by the gyroscope the precession angle was to be manually controlled by an operator using a lever [4]. This concept was successfully applied to a monorail, carrying 50 passengers on a circular track, using two gyrostats [5]. Despite the success, it was

deemed economically unviable, with rail infrastructure already established for two tracks [5]. The two gyrostats act on the same axis but with opposite spins and precession angles, the reasoning of this will be highlighted later. A similar project in 1911 by Shilovsky used similar technology, implementing a clutch controlled gyrostat in a single-track vehicle called a gyrocar[6].

More recent work on the topic was carried out by Yetkin et al. in a paper on gyroscopic stabilization of an unmanned vehicle, using a vertically oriented gyrostat for stabilisation [7]. In this investigation there was analysis of the dynamics of the bicycle with the CMG, deriving equations to describe behaviour from the system energetics. This mathematical analysis was supported by the earlier work by Spry and Girard, which fully describes all system dynamics from first principles for both single and double gyroscope control systems [6].

Notable commercial work in applying CMG technology to vehicles has recently been carried out by Lit Motors and their C-1 vehicle [8]. Operating like a motorcycle, it uses two gyroscopes to control the attitude of the vehicle, with user control similar to a car steering wheel [9]. The CMG system is used as the control scheme, prevents steering beyond the rollover point and protects against perpendicular impacts by maintaining an upright position in the event of crash detection [10]. An advanced CMG system such as this can offer the stability of a four-wheeled vehicle in a two-wheeled vehicle, using sensory input from a number of vehicle points to determine total vehicle dynamics and computing a response via flywheel precession angles and angular velocity [9],[10]. An additional advantage of the CMG makes use of the flywheel as an energy storage device, implementing a kinetic energy recovery system (KERS) under braking to maximise efficiency, as seen in Formula 1 racing cars. Lit Motors also cite a number of benefits to the C-1, offering personal transportation with a low-carbon footprint and high efficiency to keep running costs low [11]. A CMG system supplements the physical advantages of single-track vehicles (reduced weight, drag, and friction surfaces) with solutions to safety concerns, operating skill and weather exposure [11].

Joseph Schenk investigated the prospect of using gyrostat control systems for flexible purposes, culminating in a more general analysis of the topic [12]. The research was similar to what will be carried out in this investigation, with passive and active regimes using a microprocessor (Arduino UNO) in conjunction with a servo motor to create a real-time feedback loop [12]. This was to further Brennan's manual system to create an automatic response, and used a single flywheel powered by an electric motor freely rotating in one axis [12]. It was noted that there were issues with the accelerometer (MPU6050) arising from sensitivity to vibrations from the spinning flywheel. The apparatus was converted for use in a fibreglass boat to minimise rolling. Further to Schenk's work, Sammy Jackson developed a gyrostat system for use in satellites, using a pyramidal arrangement of small flywheels to control orientation in space [13].

This project focuses on producing an adaptive stability mechanism for motorcycles by integrating a control system with gyroscopes, making them safer and more accessible for users. The investigation involved analysis of the benefits of an active system over a passive system, establishing and quantifying the performance envelopes of both. In this paper there is an analysis of gyroscopic theory and relevant vehicle dynamics for both control regimes, supporting the development of a functional test bed. As the project was started from the ground up there is full description of the circuit design and microcontroller usage, including control code.

2 Theory

2.1 Gyroscopic Theory

When the angular velocity of the gyroscope flywheel is much greater than the gyroscopic precession rate the total angular momentum, L , can be approximated to the angular momentum of the flywheel, L_f . This is the gyroscopic approximation. When there is a change in the orientation of the rotational axis of a rotating body there is a precession caused by a precession torque. This occurs due to the conservation of angular momentum, with the flywheel maintaining both the magnitude and direction in space (with no externally applied torque). Note that in this investigation the flywheels are operated with a vertical spin axis, the rotation of the vehicle frame on its axis bearings is referred to as roll, and that all angular velocities are in radians per second.

If the spinning flywheel is caused to rotate by a couple acting on the spin bearings (a torque to rotate the vehicle) the rate of rotation induces a torque parallel to the plane of the spin axis and perpendicular to the applied torque. This new torque causes the CMG to rotate backwards or forwards in the vehicle frame, and is dependent upon the gyroscopic precession rate, angular momentum of the flywheel and the rotation rate of the vehicle frame. The precession torque, τ_{prec} , is given in equation 1, where I_f is the flywheel inertia in the spin plane, ω_f is the angular velocity of the flywheel and ω_{roll} is the angular velocity of the vehicle.

$$\tau_{prec} = I_f \omega_f \omega_{roll} \quad (1)$$

The gyroscope produces a restoring force, τ_{gyro} , that acts against the rotation of the vehicle frame, inducing stabilisation. This is dependent on the instantaneous precession rate of the flywheel, ω_{prec} , the moment of inertia of the flywheel and the angular velocity of the flywheel. This gives equation 2.

$$\tau_{gyro} = I_f \omega_f \omega_{prec} \quad (2)$$

The induced precession torque by vehicle motion is only relevant to the passive regime as the servo in the active regime exerts an external torque to control the precession rate. It is the rate of roll that directly induces the precession torque, hence the passive system response is dependent on how fast the vehicle frame is falling. The active control system is different, with a manually induced precession combined with the angular momentum of the flywheel to induce a gyro torque stabilising the vehicle frame. The plane this torque acts in moves with the precession angle of the flywheel. While the flywheel is horizontal all gyro torque produced acts to stabilise the vehicle frame. However, as the gyro torque moves with the CMG and the flywheel is not horizontal, only a component of the gyro torque acts as a stabilising torque. Likewise there is a component that works against the frame of the vehicle in the horizontal plane, the vehicle yaw. This yaw torque, τ_{yaw} , is why the vehicle was constructed to use two CMGs, one spinning clockwise and one anti-clockwise, to nullify any net torque that would act to destabilise a motorcycle and effect control. The stabilising torque and yaw torque are given in equations 3 and 4 respectively, where θ is the precession angle of the flywheel.

$$\tau_{stab} = \tau_{gyro} \cos(\theta) \quad (3)$$

$$\tau_{yaw} = \tau_{gyro} \sin(\theta) \quad (4)$$

Using these equations, the τ_{stab} output of the CMG to stabilise the vehicle against an applied destabilising torque can be found by equating them. Expressing equation 3 in full and using the known relation $\tau = I\alpha$, where α is the angular acceleration experienced, we get equation 5.

$$I_f \omega_f \omega_p \cos(\theta) = -I_v \alpha_{roll} \quad (5)$$

If we then express α_{roll} as $\frac{a_T \cos(\theta_{roll})}{R_v}$, assume that the frame and precession angle is near the horizontal (so we can ignore the trigonometric contribution), and using the standard equation $L = I\omega$, equation 5 is rearranged to equation 6.

$$L_f = I_f \omega_f = \frac{I_v g}{R_v} \quad (6)$$

Note that ω_p is also omitted. This is because the rate of precession is negligible compared to the flywheel angular velocity ($\omega_p \ll \omega_f$), known as the gyroscopic approximation. This equation allows the determination of the CMG flywheel with operational angular velocity to stabilise a body of known moment of inertia.

2.2 Genuino 101 and Control Code

The Genuino 101 is a microcontroller similar to Arduino boards, and is programmed using C++ scripts which can be developed and uploaded via the Arduino IDE. The code can make use of available libraries which allow easy use of components or processes. The Genuino 101 was used in this investigation as the centralized controller that outputs a response with respect to vehicle position. The Genuino 101 was chosen for the built in 6-axis accelerometer/gyroscope to measure positional data and capability to output four separate pulse-width modulation (PWM) signals (pins signified by " ~ "), which are needed to control the servo and DC flywheel motors.

2.3 Circuit Design

Designing the circuit was a large section of this research project. By accurately and precisely producing a circuit, the likelihood of experimental errors stemming from the electronic components would be reduced. It was decided early in the project that for the project to be fully independent and self-stabilising it must use on-board power - i.e. batteries. This would be determined once the power requirements of the circuit were established. The circuit started with the devices that were built into the mechanical framework: a pair of Como Drills RE-380 DC motors and a pair of Parallax Standard Servo motors. These were to drive the flywheels and control the gyroscopic precession respectively. The datasheets for the servo stated that the operating voltage of the servo was a range of 4-6V, while the datasheet for the DC motor offered nominal operating voltages, not limits, of up to 7.2V. Due to the angular momentum of the flywheels they will continue to rotate when the motor is no longer powered. This causes the motor to function as a generator, requiring a diode to protect the components from the back E.M.F. and the voltage spike from the inductive load. A Fairchild SB1245 Schottky Barrier Rectifier was chosen, capable of blocking 45 V. The flyback diodes were soldered directly across the motor connections as indicated in figure 1.

The control circuit of the servo motors is simple, with three connections for ground, power and the controlling PWM signal - which is processed internally. The DC motor

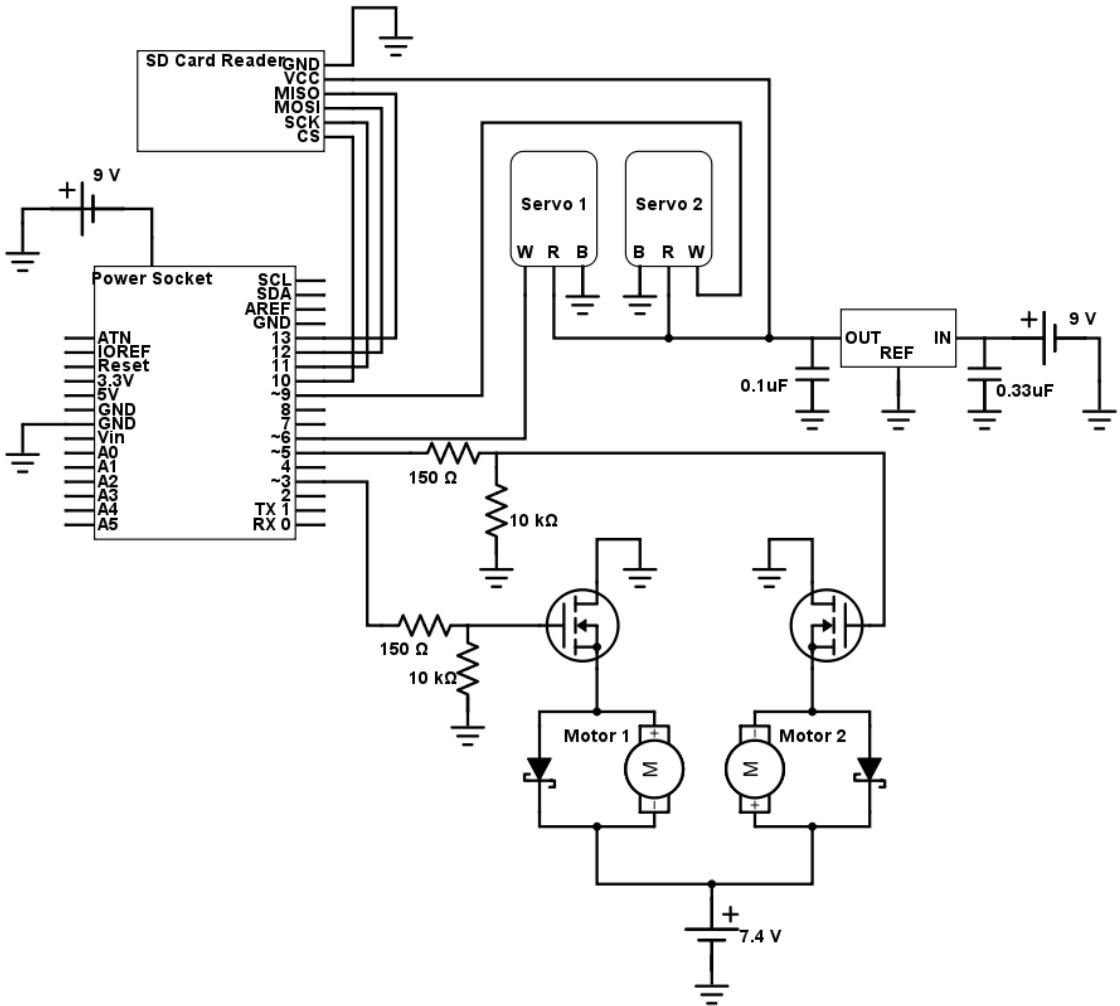


Figure 1: Full circuit diagram of all components used during testing, including the microcontroller, servos, DC flywheel motors, flyback diodes and MOSFETs. The SD card reader is powered off of the voltage regulator output for stable 5V.

control system is more complex as the motors are equipped with just a positive and negative terminal, lacking an internal control circuit. The RPM of the motors is controlled by a PWM signal which must be driven manually by controlling the current flow. To rapidly switch the current on and off to replicate a PWM signal a metal oxide semiconductor field effect transistor (MOSFET) is used. This transistor has three pins: source, drain, and gate. When a voltage greater than the threshold voltage is applied to the gate then current is allowed to flow from the source to the drain. When no voltage is applied the high internal resistance of the transistor prevents any current flowing. The MOSFET is capable of switching at very high frequencies, and by applying a PWM signal from the Genuino 101 to the gate the MOSFET reproduces this signal input through the current. The critical value of a chosen MOSFET is the threshold gate voltage ($V_{G_{th}}$), ensuring it is lower than the applied 3.3V PWM signal voltage from the Genuino 101. This limited the component selection to logic level MOSFETs, which are commonly used with microprocessors. The ST Electronics STP55NF06L was chosen due to its low $V_{G_{th}}$ and ability to handle $\pm 16V$ gate-source voltage and continuous drain current of 55A, both

far in excess of requirements. Note that in figure 1 that there are value labelled resistors in the DC motor control circuit. The resistor of 150Ω is a current limiting resistor and protects the microprocessor if the high capacitance of the MOSFET causes a current draw of more than 40mA (maximum output current on a single pin). The $10k\Omega$ resistor is a pull-down resistor and ensures that the PWM signal reads low correctly, otherwise the pin can float and cause the MOSFET to be open permanently. This would cause the MOSFET to operate in the Ohmic region of the characteristic curve, causing significant heating which would lead to damage to the MOSFET and surrounding components. The DC motors and servo motors could operate off two separate sources but to save weight, space, and cost they were designed to share a source. It was decided that an Overlander 7.4V battery would be appropriate for the size of the framework and a suitable power source [14]. In this arrangement, the battery is capable of powering the DC motors and can be used with a voltage regulator to power the servo motors. The voltage regulator was a Texas Instruments TO-220KC, which regulates a supply 5V. This component has a 2V voltage dropout, so requires a supply to deliver a minimum of 7V to maintain a stable voltage output. In figure 1 it is shown that there are capacitors to ground before and after the voltage regulator. These stabilise and filter noise from the regulated power signal, leading to a cleaner power supply for the servos and SD card reader and theoretically better operational performance. The voltage regulator was verified to supply a stable voltage of 4.97V all sources.

The power requirements of the Genuino 101 are such that a source with a range of 7-12V is sufficient to power all processes and signals. This can be applied via the power jack or the V_{in} pin. Note that the microprocessor runs off 5V, but the built in voltage regulator has a dropout voltage requiring that only a supply in the 7-20V range produces stable operation, with 12-20V generating high temperatures in the regulator. A separate 9V battery was used to power the Genuino 101 via the power jack to keep it separate from the main circuit, stabilising the input power against potentially large draws from other components.

2.4 Theory Of PID System

A PID system is a relatively simple method of determining an output for a system based on the difference between the value of the system's current state and a desired setpoint (the error). The controller uses the output to minimise the error, with system behaviour calibrated using three distinct tuning variables. PID systems use equation 7 to calculate the response, where $e = Setpoint - Input$, and the tuning variables have to be manually adjusted based on observed system performance.

$$Output = K_P e(t) + K_I \int e(t) dt + K_D \frac{d}{dt} e(t) \quad (7)$$

There are three distinct control modes within the PID system; proportional, integral and derivative control. The proportional mode has the greatest impact on system response, changing controller output in proportion to the error. The integral control mode influences how quickly the system reacts to error. If the error is large then the system will react quicker, helping to keep the system closer to the setpoint. The derivative control mode influences the output with respect to the rate of change of error. In this investigation that would be in response to the frame falling, with the rate of error increasing as it falls under gravitational acceleration. The derivative mode is very sensitive to noise

in the input. For this investigation the input, setpoint and output were set to the angular position data, zero degrees and servo angle respectively. The PID system takes the angular position data, determines the error between current position and zero, runs it through equation 7 and outputs a variable which can be used to control servo position.

3 Methodology

3.1 Vehicle Test Frame

A test frame was designed and constructed specifically for this investigation by Dr. Phil Lightfoot. A rectangular frame with space inside for two CMGs and an area in the middle for components and power is suspended on bearings at both ends and is shown in figure 3. This is contained within a stiff metal cubic structure. The bearings allow the vehicle to freely rotate along the central axis. Additional shielding could be attached to the structure for safety in the event of structural failure of the gyro cages. The gyro cages are fixed to the frame via conductive bearings, enabling low-friction precession rotation of the flywheel and producing a safe route for the DC motor power cables, as any loose cables could get caught in the CMG.

The prototype arrangement included two sets of flywheel with a known inertia coupled to a driven axis or shaft which receives a torque from an electric motor. The flywheel was held inside a solid box arrangement which could rotate freely on bearings or be controlled via a servo in an axis perpendicular to the input torque, acting as a driven gimbal. The two flywheels can be driven by equal and opposite torques, and subsequently respond in opposite directions. This will maintain total weight distribution and reduce vertical displacement allowing a smaller vertical profile and reduce the centre of mass of the prototype.

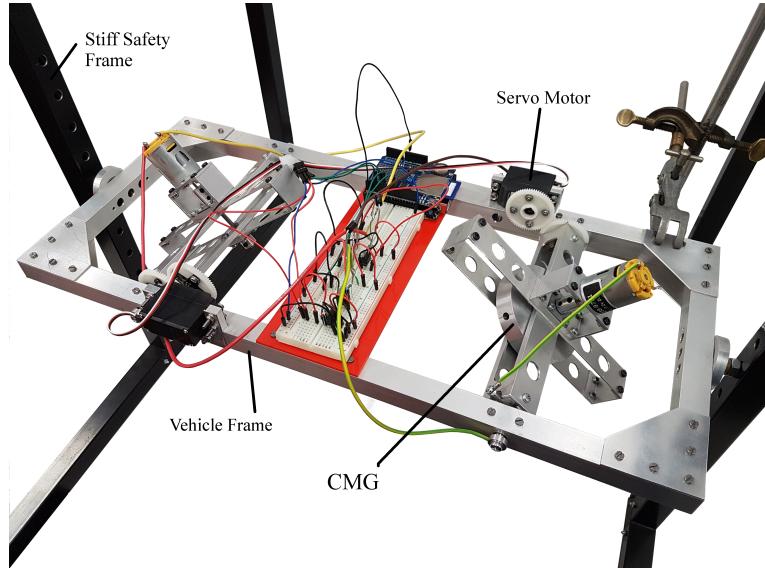


Figure 2: A picture of the vehicle frame in the safety cage, showing the vehicle frame containing two CMGs. Note that a clamp was used to hold the vehicle level and the background has been edited out for clarity.

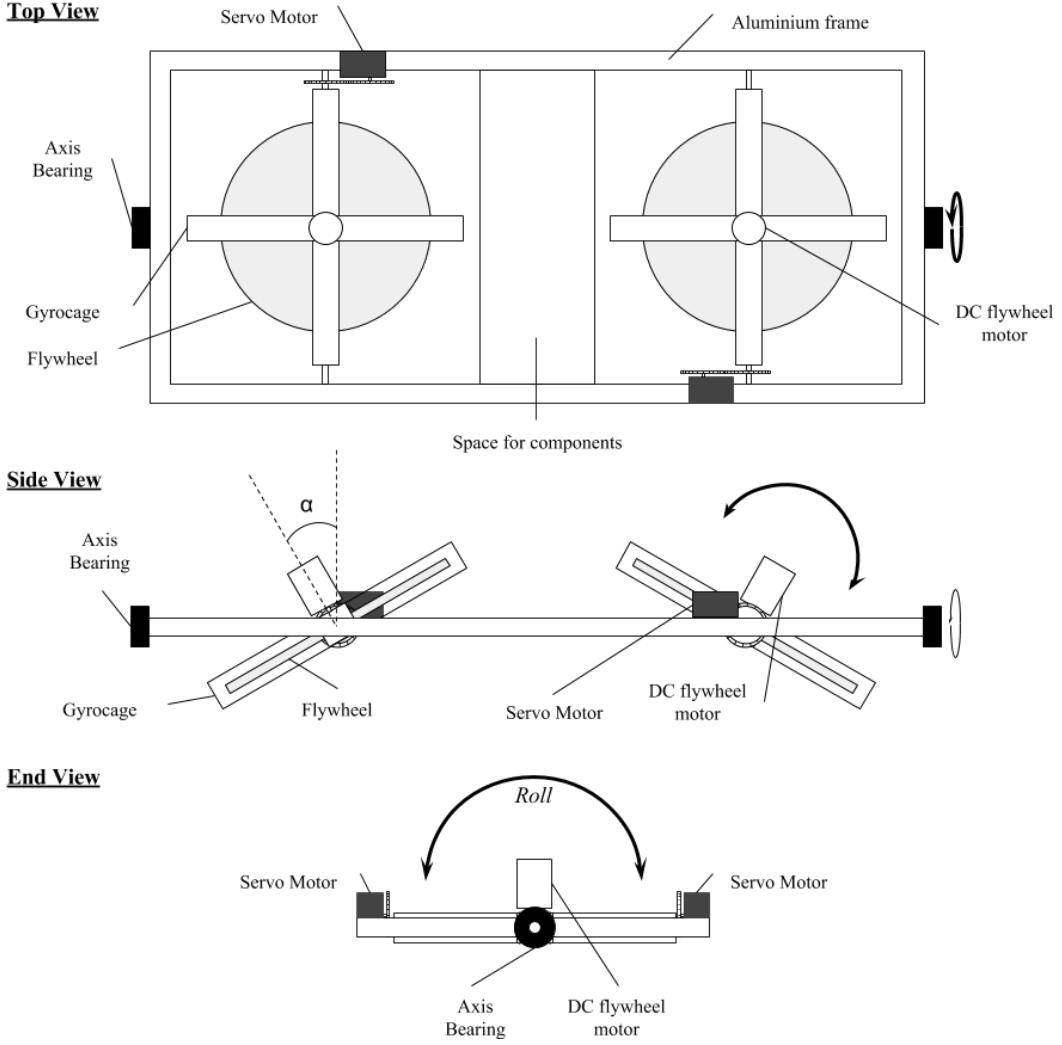


Figure 3: A diagram of the vehicle test from from top, side and end views. Arrows indicate the rotation direction, α describes flywheel precession angle, and *Roll* indicates the vehicle roll as dicussed in section 2.1.

3.2 Designing The Flywheel

The flywheel to be used in the gyroscope was designed using CAD. The size and shape was decided from equation EQ. The moment of inertia, I , of a ring is a factor 2 greater than the moment of inertia of a solid disk. While it is not possible to constrain the mass exactly to a thin ring an accurate theoretical inertia can be determined using equation 8, where M is the mass of the shape and a and b are the internal and external radius respectively. The volume of the designed flywheel was found using the CAD program which was combined with the density of steel to determine the mass.

$$I = mr^2 \quad (8)$$

Where I = inertia, m = mass of flywheel, r = radius of flywheel

$$I = \frac{1}{2}M(a^2 + b^2) \quad (9)$$

The dimensions of the vehicle frame constrained the maximum diameter of the flywheel to 145mm and the maximum thickness to 12mm. Stress and strain of flywheel

designs are only necessary at very high rotation rate, thus the main design consideration is to move the mass to the edge. A four spoked design of $(525.58 \pm 0.001)g$ was produced, but due to incompatible workshop machining times with project timescales only one flywheel could be manufactured for the investigation, which was produced to a simpler design: a disc with large circular holes drilled into it. This shape would have a lower moment of inertia than the designed shape, and would be more comparable to the moment of inertia of a thick hoop as in equation 9.

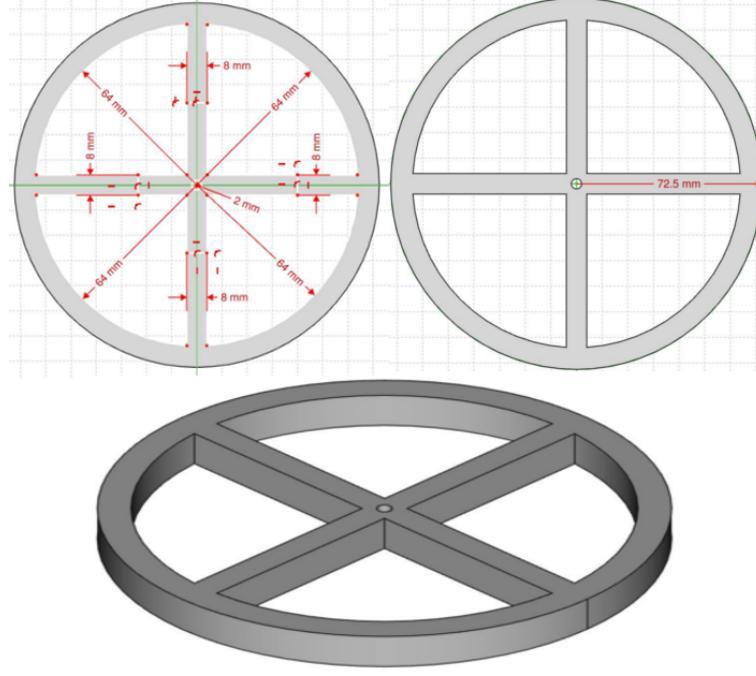


Figure 4: CAD flywheel with dimensions. Note that this was not the final flywheel used.

3.3 Measuring Inertia

The inertia of the flywheel was determined experimentally by mounting it on a PASPORT Rotary Motion Sensor which allows the flywheel to spin horizontally on a spindle. The spindle has wheels to attach a known mass to by a thread. The dropped mass exerts a known tangential acceleration on the flywheel and the rotation of the flywheel is measured and recorded by a computer. Derived from the conservation of energy shown in equation 10 and converting angular velocity to linear velocity, equation 11 calculates the moment of inertia of the flywheel, I_f , where m is the dropped mass, r is the radius of the axle the mass is wrapped around, g is the acceleration due to gravity and h is the height at which the mass is dropped.

$$mgh = \frac{1}{2}I\omega^2 + \frac{1}{2}mv^2 \quad (10)$$

$$I = mr^2 \left(\frac{2gh}{v^2} - 1 \right) \quad (11)$$

The velocity of the mass was found by timing the descent from the data collected. The test was carried out with the large flywheel by stacking it on top of a light flat aluminium disc - as the large flywheel alone was susceptible to lateral movement - and

an aluminium securing piece on top to connect to the axle. The test was then run with this testbed. As moments of inertia are additive, the moment of inertia of the flywheel is found simply by $I_f = I_{f+testbed} - I_{testbed}$.

The moment of inertia of the vehicle frame was determined experimentally by attaching a known mass on a string to the bearing that allows it to spin on an axis. There is no need to account for the frictional effect of the bearing because any force exerted against the motion of the system will be exerted during testing, so the determined inertia would be representative of the testing system.

All component and circuit tests were carried out using a power supply, but passive and active system tests on the frame were powered using battery power. Using batteries is essential to make the system fully independent as wires would influence the dynamics of the vehicle. The final testing used an Overlander Supersport Pro 35C Lipoly battery, which is capable of supplying 7.4V with a 2200mAh capacity, to drive the flywheel motors. The larger battery offers improved performance over standard commercially available batteries by fulfilling motor power demands while reducing the vehicle mass. A battery alarm was used with the Overlander to monitor cell charge, as the cells become unstable at a low enough capacity and if the cells discharge at different rates. A 15A fuse was used inline with the cable running from the battery to the breadboard to protect the battery and components in the event of a short circuit.

3.4 Control Code

The control code was developed throughout the project, adding new functionality and testing as required. Individual control scripts were produced to verify component and microcontroller functionality, with successful code integrated into the final script. After initial familiarisation with the device using the example scripts available in the Arduino IDE each active component was tested individually with the Genuino 101. The servo motors were operated using the available *Servo* library. The servo motor responds to an inputted PWM signal, rotating to a position as needed. This library allows the user to control the angular position of the servo motor by using simple commands and input a value of degrees. The library handles the generation of the controlling PWM signal, although the user can choose to dictate the PWM pulse size in microseconds if more precise control is required. Servo functionality was verified using the sweep example code for both degrees and microseconds under various power sources. The sensors can measure position every 2ms, and as such the main limiting factor to loop frequency is the time taken for the servo motor to move to position.

The DC flywheel motor speed is also controlled via PWM, but uses high speed switching through the use of the MOSFET to produce the signal pulses. The Genuino 101 outputs a PWM signal to the MOSFET, controlling the motor switching through the component. The functionality of the MOSFET was verified with the circuit shown in figure 5, using the microprocessor to read the value of a potentiometer to vary the PWM output, in turn controlling the DC motor speed.

Using the *Wire* library the Genuino 101 can interface with an SD card reader. This adds the functionality of recording all data to a microSD storage device allowing full data collection and ensuring there are no wires attached to the vehicle. Without this data would have to be extracted via the serial port over USB, disrupting vehicle dynamics and preventing accurate assessment of system performance. The SD card reader was verified using the *Read/Write* example code.

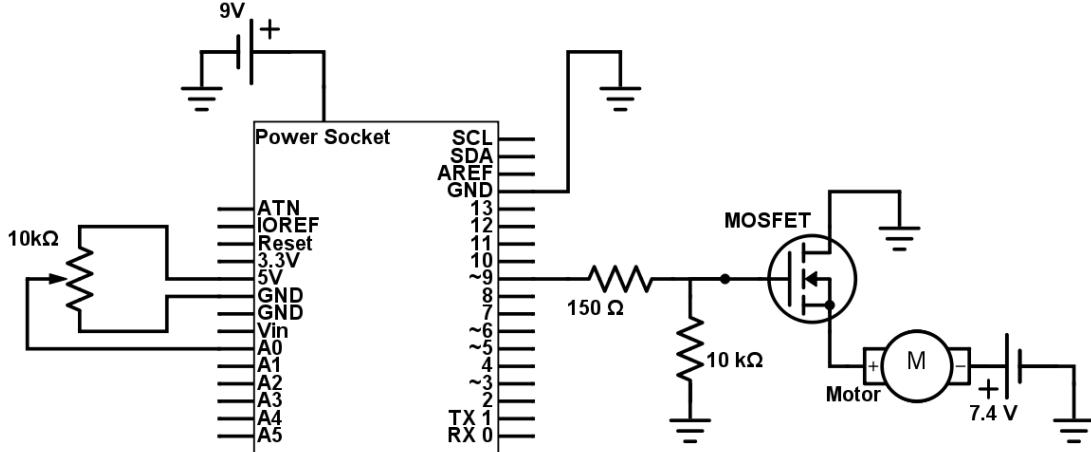


Figure 5: MOSFET test circuit designed to verify operation and confirm that DC motor speed can be fully controlled with the microcontroller. The microcontroller and motor power sources are separate to ensure the board does not suffer from current fluctuations induced by the motor.

The Genuino 101 has both a gyroscope and an accelerometer. Using the *CurieIMU* library that was specifically developed for the Intel CurieTM chip, the raw data from these sensors can be accessed. The sensors have different performance regimes in time, with the gyroscope precise for a short time and the accelerometer less accurate for position but is more reliable for longer periods of time. The effect of the sensor drift can be mitigated using a filter. The data from both sensors was optimally fused using the Madgwick filter to determine vehicle orientation[15]. The Madgwick filter was used (via the *MadgwickAHRS* library) because it offers better filtering against sensor drift than a normal Kalman filter and has been successfully implemented on this board before, ensuring compatibility and reducing complexity. There is an outputted roll of the vehicle in degrees.

The PID system was implemented using the *PID_v1* library, providing a simple interface with a set of functions to calibrate the PID algorithm. The *SetSampleTime* function is used to keep the PID updating regularly, but in this investigation it was set to 1ms, effectively setting the PID to update every loop. Note the default update time for the PID system is 20ms. The *SetOutputLimits* function changes the size of the PID output, and can be used to ensure the system can write a compatible value to the servos. There are two operating modes for the PID, manual and automatic, equating to off and on using the *SetMode* function. For the full control code see appendix.

3.5 Passive System

The passive system was tested by detaching the servo gear wheel from the servo horn so the the gyro cage could rotate freely. The flywheel motor was driven at full speed directly from the 7.4V battery with the frame and gyro cage held until the flywheel stopped accelerating. The system was released and the angular position of the vehicle was recorded to allow direct comparison of the physical characteristics of the passive and active systems. Analysing the data from the passive runs also gives insight into the effect of vibrations on the positional sensors of the Genuino 101. As the movement of the frame is slow and smooth a graph of angular position against time should also show a smooth line, and any deviation would be due to an error in the positioning data. This can be

compared to positional data with time when the device is stationary and no vibrations are present, returning the error due to vibration.

3.6 Active System And PID Tuning

For the first stage of testing the smaller flywheel was used for safety, as the system behaviour was still unknown. The active system requires the servo to be connected to the gyro cage via the nylon gear wheels. The servo was adjusted such that the gyro cage (and by extension the flywheel) was horizontal when the servo moved to 90° degrees. This was done by removing the gear wheel from the servo, using the microcontroller to rotate the servo to 0° or 180° and reattaching the gyro cage at the corresponding vertical position. This was found to be easier than carrying out the same process at 90° as the servo could rest against an endpoint, whereas at 90° it may move slightly when reattaching the gear wheel. The PID system uses three tuning variables to calibrate the effect of the proportional, integral and derivative responses; K_P , K_I , and K_D . To test the system response to different setups the frame was held horizontal while the flywheel motors spun up to speed and then released cleanly. The angular position of the frame and the demanded servo output was recorded to profile system performance. It was important to check the weight distribution of the vehicle frame regularly to ensure it was balanced. This was done by holding the frame, dropping it and observing the rotation, rearranging the components if necessary. If the frame is not balanced the restoring force provided by the gyroscope does not correct the frame equally so will fail to right itself properly, leading to destabilisation. Likewise, all cables were taped down as moving them would cause variable mass distribution. To start, K_P was varied until the system produced a self-stabilising oscillation. This is the most basic form of stabilisation the system can produce and can only be considered stable enough to not rotate past a point of no return. Having identified a rough value of K_P to produce this state the value was varied at a finer scale to produce better performance as a basis for further tuning, determining the K_P value at which all further tests will be carried out. The derivative tuning variable, K_D , was then varied and the response recorded. The length of time that the system was stable for was plotted against K_D and response chosen on the value that offered optimum stability. The integral variable was tuned similarly, sampling the stability decays and assessing the performance. The standard deviation of the angular position of the frame was plotted against K_I , where the standard deviation relates to vehicle stability by describing the typical oscillation size.

Once the performance of the active system was characterised the system was changed to incorporate the larger flywheel. The cage holding the smaller flywheel was completely removed as this was a new testing regime and the vehicle structure could be optimised. Reducing the mass of the frame would improve performance as it would require less torque to right itself.

4 Results

4.1 Determination of Inertia

The disc shaped small flywheel inertia was approximated to $(1270 \pm 1.733)\text{gcm}^2$ using equation 8. Using the method from section 3.3 the moment of inertia of the larger flywheel of mass 588.7g was found to be $(19850 \pm 253)\text{gcm}^2$, nearly 16 times larger than the smaller

flywheel. Using the same method, the moment of inertia of the vehicle frame when inside the testing frame was determined to be $(283810 \pm 253)\text{gcm}^2$.

4.2 Small Flywheel Tuning

Figure 6 shows the how the system behaviour varied with K_P , with sampled stability decays from repeated runs for each system. From the graph it is clear that of the values plotted that a K_P of 5.00 is the most stable. Working from this the system was tested around this, observing performance and plotting behaviour. This resulted in a final K_P value of 5.50, which gave a high quality oscillation and relatively long lasting stability before decaying, as shown in figure 7. The standard deviation of this system was determined to be (10.55 ± 0.52) .

The results of tuning K_D are shown in figures 8 and 9. The standard deviation of the angular displacement of the vehicle is indicative of how well the vehicle is stabilised, describing the typical variation of the frame from the equilibrium. From figure 8 of standard deviation the optimum value of those shown would be 0.10, as this has the lowest typical angular displacement. However, figure 9 shows that the most stable system

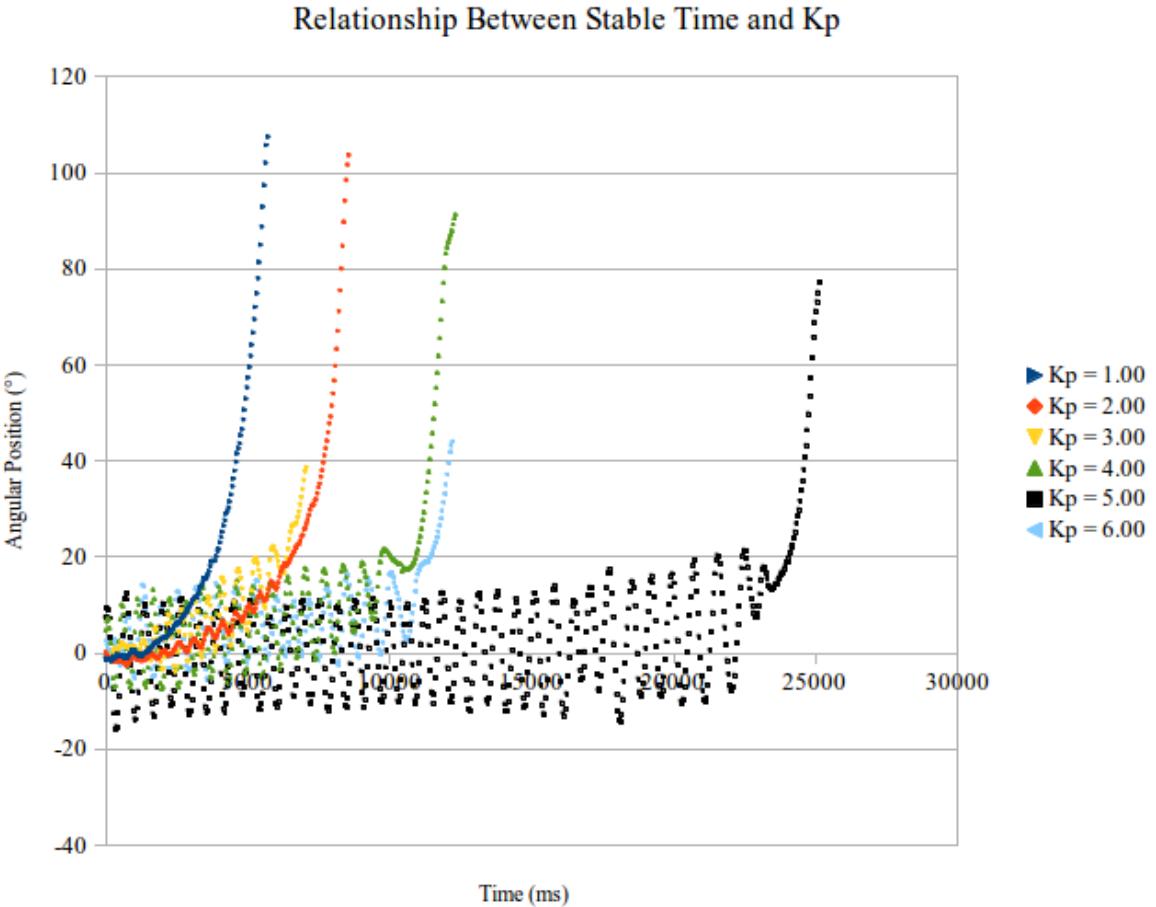


Figure 6: The plotted relationship from sampled stability decays for different values of K_p . The system that is stabilised best is the system that sustains a position around equilibrium for the longest period of time. This information was used to determine the chosen value of K_p , with further sampling carried out to ensure tuning optimisation.

PID Tuning

$K_p = 5.50, K_i = 0.00, K_d = 0.00$, Motor Speed ≈ 11000 RPM

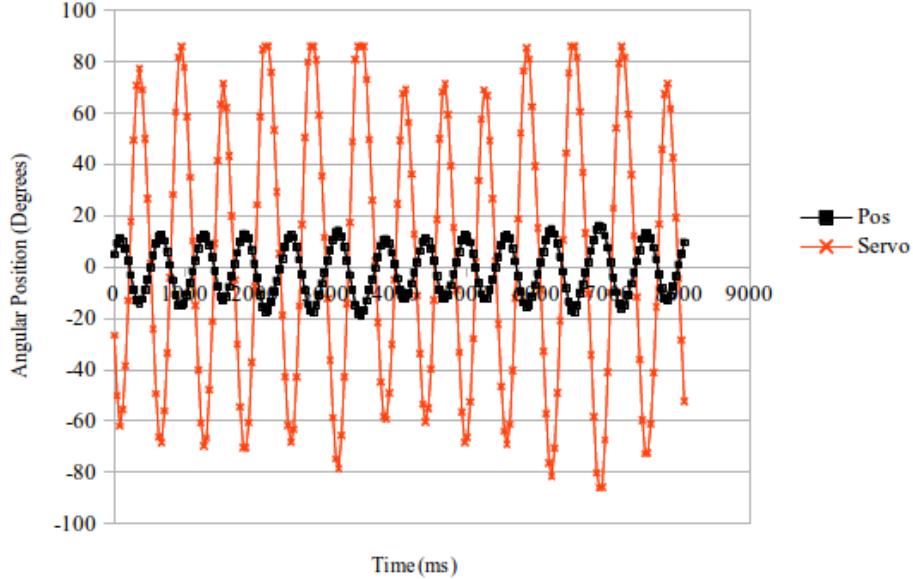


Figure 7: A representative sampled stability decay showing vehicle oscillation around the equilibrium point of the vehicle frame. The demanded servo position clearly mirrors the behaviour of the vehicle, acting opposite to the motion. The analysis of the vehicle behaviour determined the choice of $K_P = 5.50$ for continued tuning.

is at $K_D = 0.03$, with the system stable for nearly factor 3 longer. This value was chosen for further tuning.

The testing with the final tuning variable, K_I , for the smaller flywheel was decided using the standard deviation, with the aim of minimising the displacement from the equilibrium. Figure 10 shows the standard variation with K_I , with a plotted trend line. The optimum value with the lowest standard deviation is shown to be 0.30. It should be noted that the error bars on this graph are large, therefore any conclusions drawn from the trend should not be treated as fact. However, the difference between the values are low and a value for K_I had to be chosen, and with the values producing similar stabilisation times these results determined the choice+.

A sampled stabilisation from the fully tuned active system with the smaller flywheel is shown in figure 11 describes the typical behaviour of the vehicle, with smaller oscillations around the equilibrium point than in figure 7, with a standard deviation of (5.00 ± 0.19) .

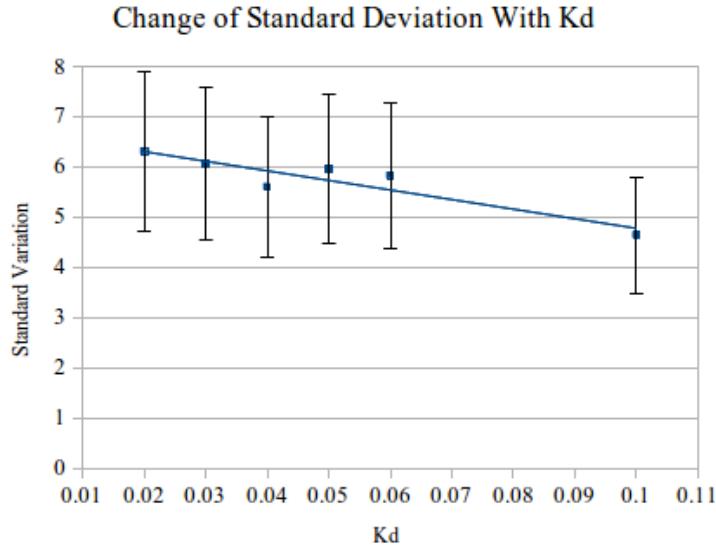


Figure 8: A graph of the determined standard deviation of the vehicle frame from the equilibrium point with varying K_D . The trend indicates that the most appropriate choice would be the higher values of K_D . The higher the value of K_D the faster the PID algorithm responds to a change in vehicle position. This would make the servo response at low roll angles larger, causing a decreased standard deviation around the equilibrium.

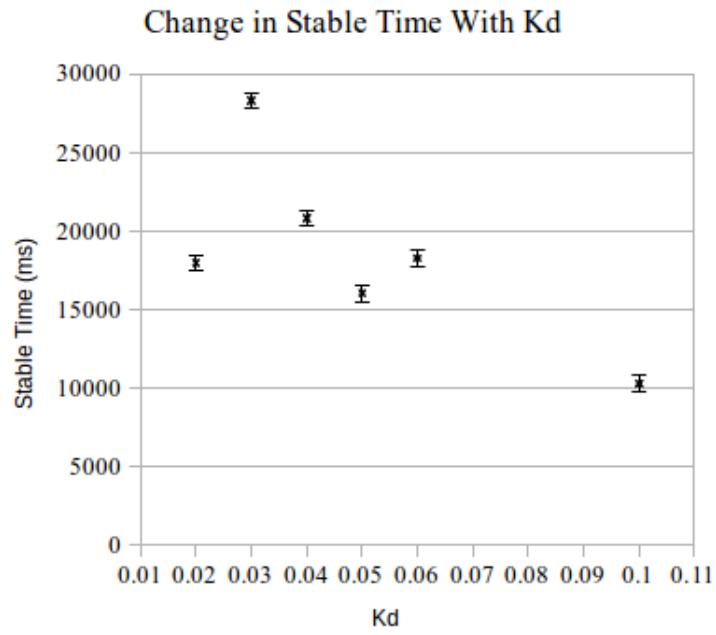


Figure 9: A graph showing typical stable time of sampled stability decays for varying values of K_D . The stable time was prioritised over standard deviation as last tuning variable was yet to be tested and could be used to reduce this instead. This behaviour is due to the system responding too much to the vehicle falling, resulting in the CMG generating a larger than required restoring force. This results in the system instigating its own destabilisation.

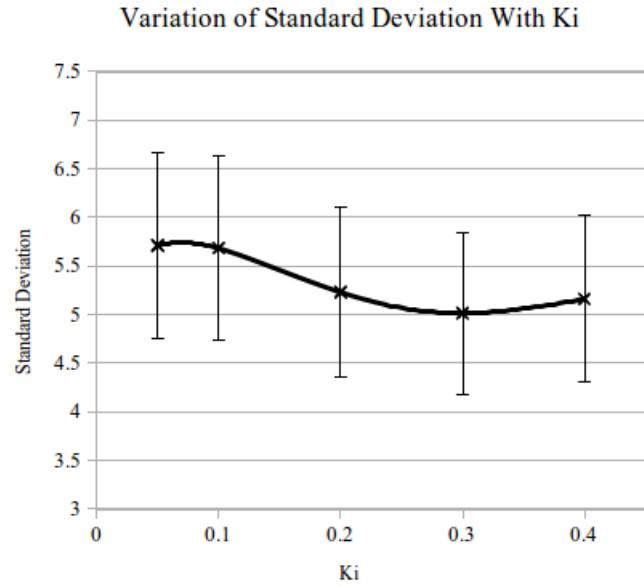


Figure 10: This graph shows the use of the K_I tuning variable to reduce the size of the vehicle oscillations around the equilibrium, establishing a complete tuning for the active system with the smaller flywheel. The influence of K_I on the response is to change how quickly the system reacts to small errors. If the value of K_I is too high then the vehicle oscillation around the equilibrium point will increase. The fitted curve highlights this trend, with the optimum value at the lowest standard deviation.

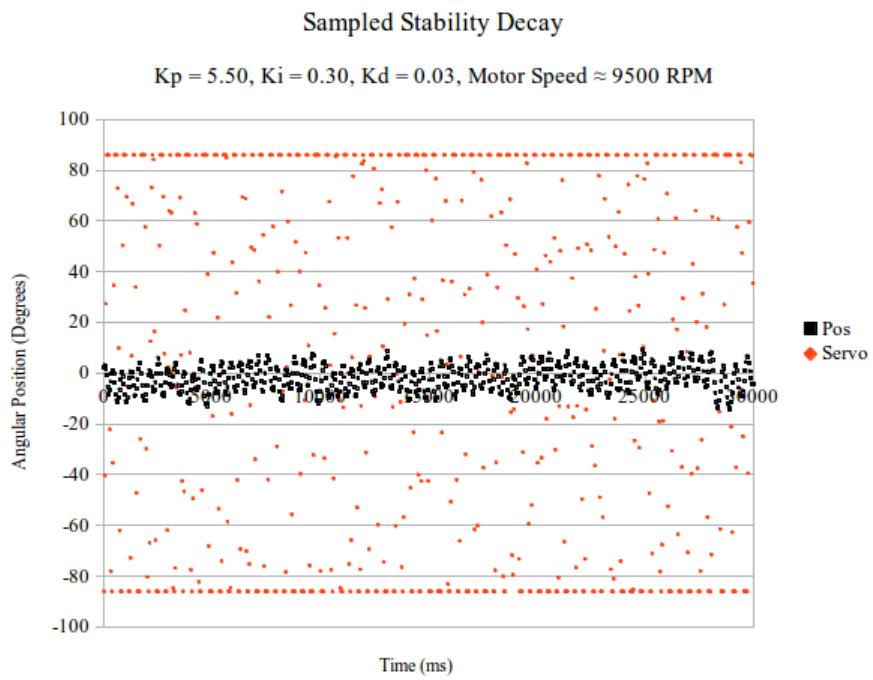


Figure 11: A sampled stabilisation from the fully tuned system, showing sustained stability for more than 30 seconds. The standard deviation of the vehicle position was found to be (5.00 ± 0.19) , less than half of the standard deviation attained with proportional tuning only.

4.3 Passive System

The results of the passive system testing is shown in figures 12 and 13. The angular displacement was plotted against time for two values of RPM, showing how the system stabilisation is improved with higher flywheel angular velocity as there is more available torque. The higher RPM flywheel maintains an angular position less than $\pm(20.0 \pm 0.005)$ degrees for (231.78 ± 0.024) seconds, compared to the smaller flywheel performance of (170.64 ± 0.024) seconds. Figures 12 and 13 also demonstrate the advantages of an active system over a passive system, with the system decaying over time and failing to stabilise even at high RPM, with slow gyroscopic response to vehicle roll. Passive stabilisation was not achieved with the smaller flywheel, even at high RPM.

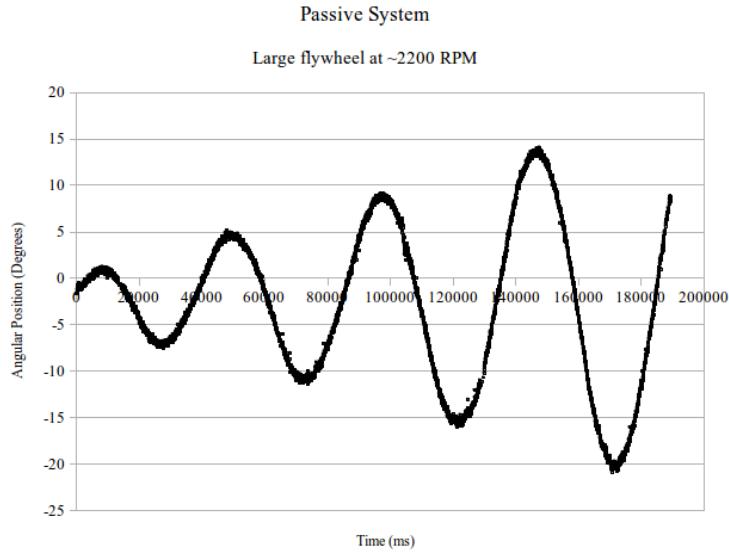


Figure 12: The passive response of the CMG with a flywheel RPM of (2190 ± 55.5) RPM.

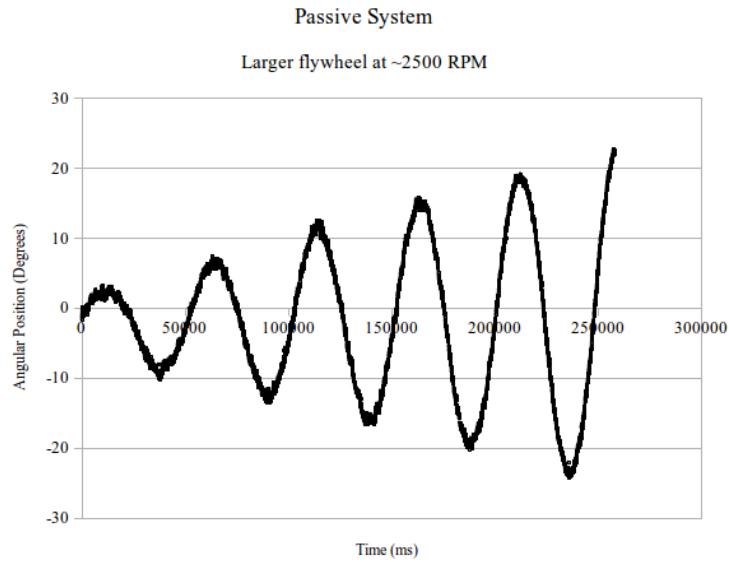


Figure 13: The passive response of the CMG with a flywheel RPM of (2510 ± 94.1) RPM.

4.4 System Performance With Larger Flywheel

The smaller flywheel tuning was used as a primary proof of concept to be carried forward to the designed system with the larger flywheel. Tuning was achieved by observing the effect of changing the variables first hand, not using data. This method may not have resulted in a perfectly tuned system, but did produce good performance, high quality stabilisation and allows direct comparison with the passive and smaller flywheel systems. Figure 14 shows the final tuned system and figure 15 compares the use of the small and large flywheel. At 2500 RPM the system produced stabilisation with a standard deviation of (1.11 ± 0.04) , a factor 4.5 improvement over the smaller flywheel. The stable time was also excellent, typically decaying after 30 seconds of stability.

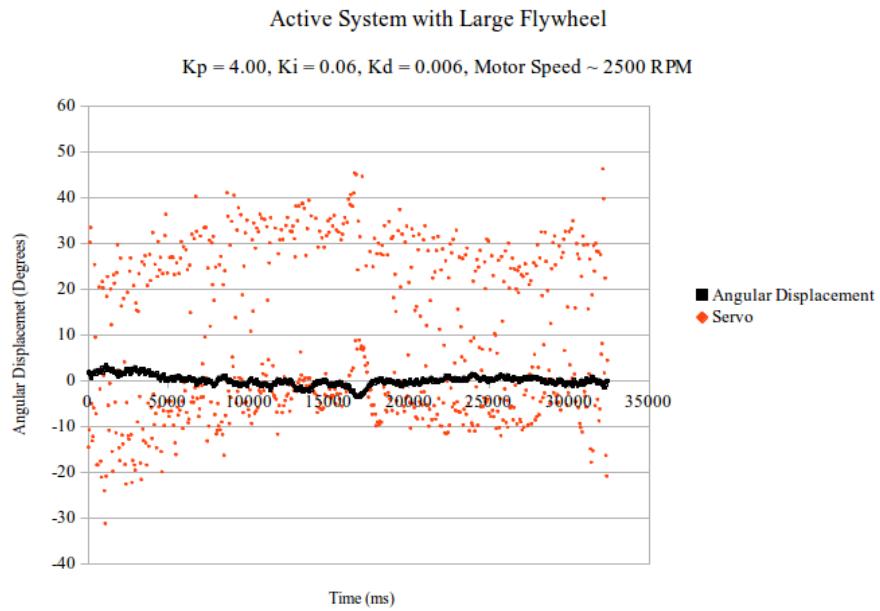


Figure 14: A sampled stability decay of the fully tuned active system using the larger flywheel, showing how the vehicle frame stays closer to the equilibrium than the system with the smaller flywheel (figure 7). The vehicle stayed at equilibrium within (1.11 ± 0.04) degrees.

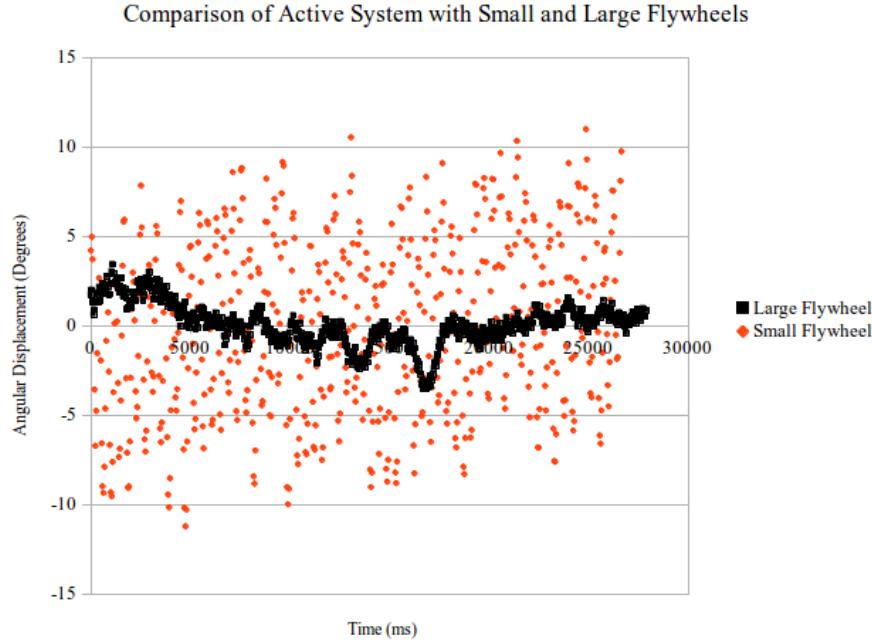


Figure 15: Comparing the two induced stabilisations of the tuned active systems with the smaller and larger flywheel. The two positional datasets are overlaid to allow direct comparison. It is clear that the large flywheel allows stabilisation closer to the equilibrium point due to the larger available restoring force providing a faster response.

The improved response is due to the larger restoring force that can be produced using the larger flywheel, allowing the system to induce stabilisation against vehicle motion faster as servo position is not changing as much. This highlights how the quality of servo is directly linked to system performance, with higher quality servo motors capable of delivering higher torques and therefore producing a more sensitive active system.

4.5 Stabilisation Of The Bike

Figure 16 shows the stabilisation achieved when the model was converted to a bike. The stabilisation is brief and does not maintain equilibrium but the active control system does keep the bike from falling for a few seconds. The large decrease in performance is due to the additional $(1.7433 \pm 0.0005)\text{kg}$ of the wheels. Figure 17 shows the complete model vehicle, with a total mass of $(4.25 \pm 0.05)\text{kg}$. The background has been removed for clarity.

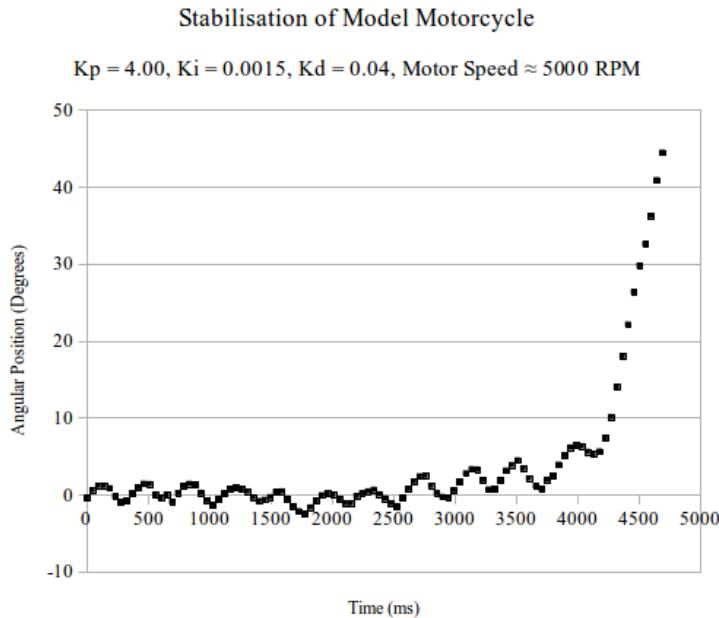


Figure 16: Final system stabilisation with wheels attached to the vehicle frame. Long term stabilisation was not achieved due to the increased vehicle moment of inertia, but the active system maintained a position around equilibrium for (3.797 ± 0.327) seconds.

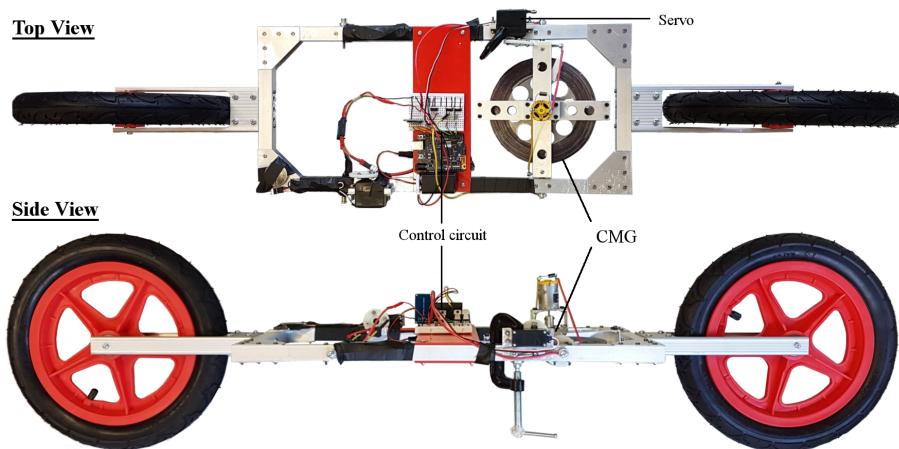


Figure 17: Images of the complete test vehicle from the top and the side. The background has been removed.

4.6 Dropout Identification

Analysing the stability decays of multiple runs with different tunings revealed a possible cause of a number of destabilizations. As shown in figure 18, before the system loses control of the vehicle attitude there is a distinct gap in the data where no position is recorded and no servo signal is sent. Figure 19 shows the loop time of the sampled data. Before the dropout the average loop time was (49.5 ± 0.022) ms, but during the dropout the average loop time increases to (147.2 ± 85.0) ms. Collecting dropout data from 7 different runs gave an average total dropout time of (943 ± 180) ms, enough time for the vehicle frame to rotate beyond an irrecoverable position. At this point the CMG system cannot induce a high enough torque to fully right the vehicle.

The observed dropout was likely due to inconsistent performance of the SD card reader, causing variable write time. As data is written every loop the chance of experiencing dropout is high. This could be mitigated by writing the data from multiple loops to a buffer and then writing this single unit of data.

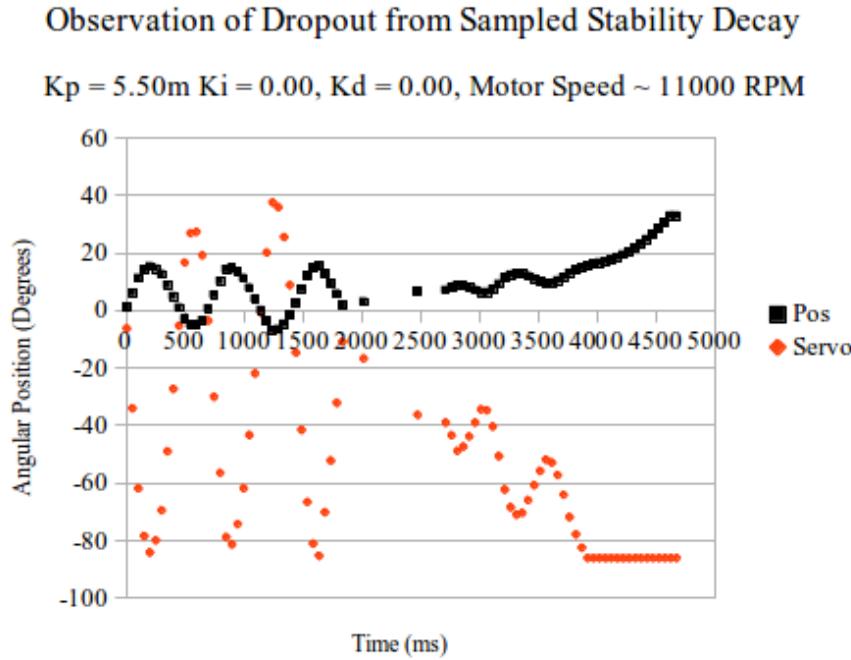


Figure 18: Sampled system behaviour from an active system run with the smaller flywheel. The missing data from the centre of the graph is the effect of the dropout, after which the system cannot recover to stabilise. This is likely caused by read/write time variation on the SD card stalling the control script.

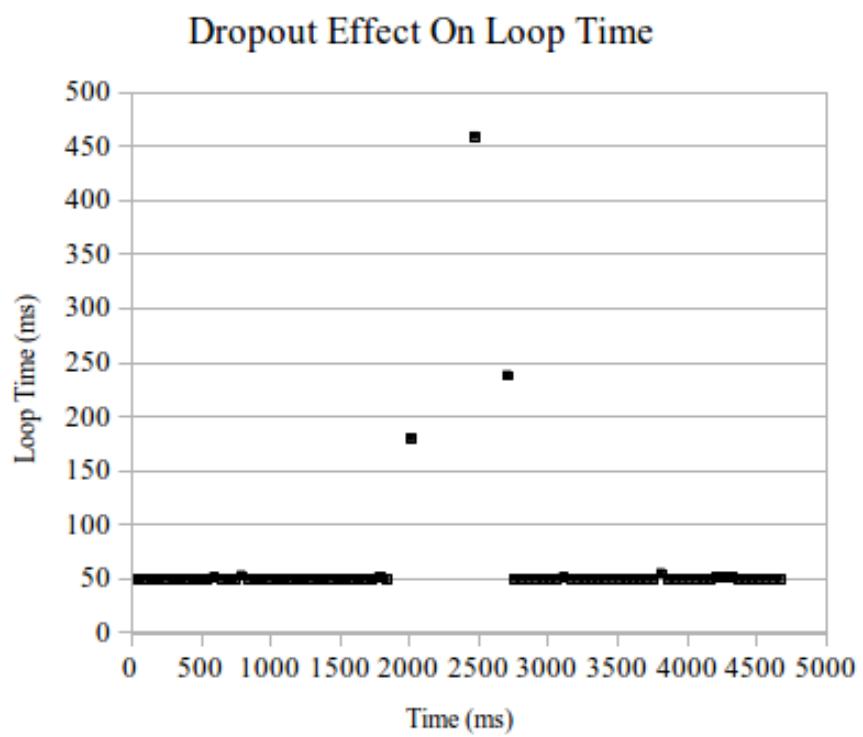


Figure 19: This graph shows the loop time throughout the stability decay shown in figure 18, clearly displaying the significant increase in loop time due to the dropout. From a number of sampled decays the average droupout time was found to be $(943 \pm 180)\text{ms}$, a significant amount of time for the active system to not respond.

4.7 Positional Sensor Noise and Effect of Vibration

Shown in figure 20, sensor data from the microcontroller under stable conditions with no vibrations is overlayed with sensor data with the microcontroller experiencing vibrations. Both data sets were collected when using the Madgwick filter. The figure shows that even with no external influence there is still an error in the recorded position, found to be (0.147 ± 0.013) . With the vibration the error increases to (0.297 ± 0.027) , more than doubling the error in position. Any error in position would cause the PID algorithm to produce the incorrect response to the true vehicle dynamics, and with larger error the more the outputted PID response will act against system stabilisation.

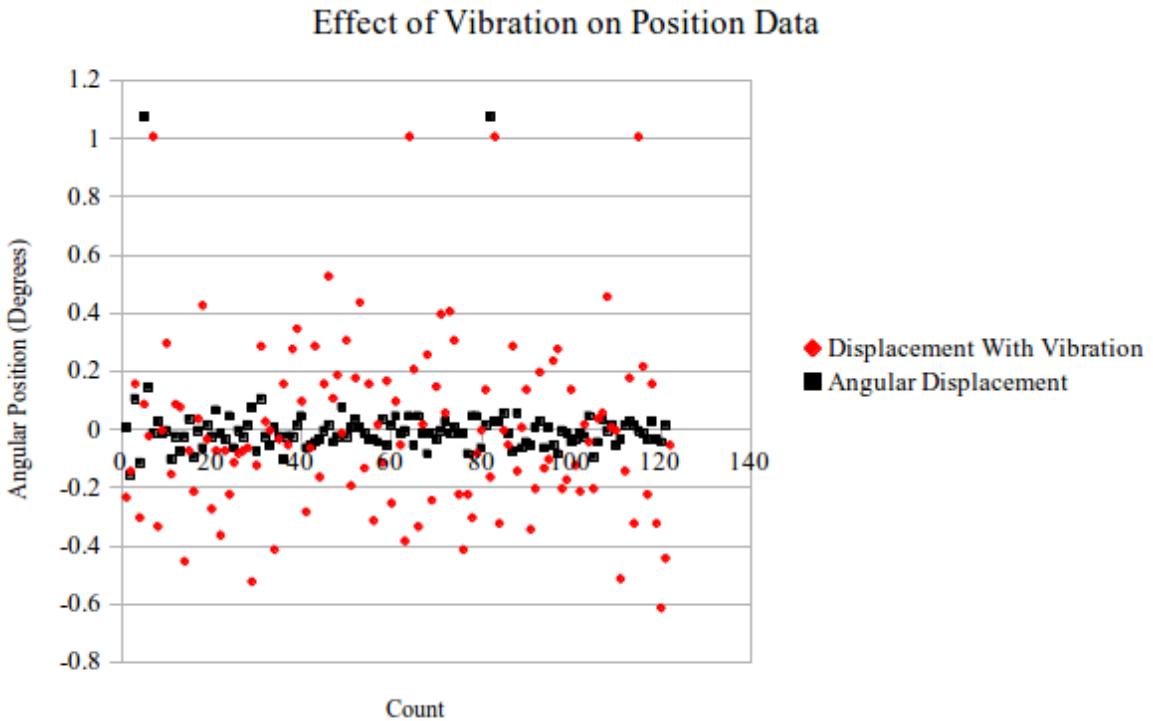


Figure 20: Sampled angular position data from microcontroller when exposed to and free from vibrations, overlayed to make a comparison clearer. The error in these scenarios was found to be (0.297 ± 0.027) and (0.147 ± 0.013) respectively. The tiny magnitude of these errors show that the effect of vibrations on the sensor data can be ignored, with the outputted servo response only accurate to 0.5 degrees.

The magnitude of this error would not have a large effect on the PID response, as the actual servo response is not accurate enough (to 0.5 degrees) for any data offset to effect the CMG response, hence the influence of vibrations on the sensor data can be ignored with respect to system performance. If the addition of another flywheel induces unmanageable noise additional sensor input from different points in the frame would allow determination of an average angular position, reducing any effect of the noise.

5 Discussion

The passive system was conclusively shown to be inferior to the active system, failing to induce controlled stabilisation with the same RPM as the active system, caused by the slow natural precession rate of the gyroscope. Investigating the passive control could be

improved by adding a mass to the gimbal above the flywheel, inducing a larger precession torque for a faster response and better comparison with the active system.

Active stabilisation was achieved with both the small and large flywheel, maintaining a position about the equilibrium within (5.00 ± 0.19) and (1.11 ± 0.04) degrees respectively. The smaller flywheel was predominantly used for safe testing and experimentation, while the larger flywheel was designed to produce the correct gyroscopic torque.

Whilst stabilisation was achieved a method of determining optimum tuning should be developed. The methods used in this investigation had significant errors due to performance variation. The magnitude of these errors would be reduced by carrying out more runs and quantifying the average behaviour of the system. This could not be carried out to the required scale for certainty due to time constraints, but the results with the smaller flywheel confirm notable performance improvements with tuning, achieving an active stabilisation system capable of maintaining the vehicle frame position within (1.11 ± 0.04) degrees.

The performance of the system to stabilise was only tested around the vehicle equilibrium and was not verified to operate against unexpected, applied destabilising forces. In theory, however, the PID system would fail at higher roll angles, because the algorithm relies on the assumption that the system dynamics are linear. As the gyroscopic torque used decays with $\cos(\theta)$ the PID would fail to provide the required response at high rolls. This effect could be mitigated by increasing the angular momentum of the flywheel with precession angle.

Testing the system within the bike structure shows that the system is capable of operating in situ, a significant achievement for this investigation. Similar to the large flywheel tuning, this system was quickly tuned by observing the response first hand. Restricted to one flywheel when it was designed to use two, brief stabilisation for (3.797 ± 0.327) seconds was produced.

5.1 Investigation Issues

The inability of the servo motor to move to the required position fast enough was a significant limiting factor in producing stabilisation, and would be a larger barrier to investigating the system response to an applied destabilisation. This is shown in figure 7, where the servo data shows that the microcontroller demands the full range of servo position, but this was never physically achieved by the servo. Future investigations should fully quantify servo performance to establish system limitations, ensuring that servo motion is completed, and implement servo motors with higher torque. This system would react faster to input signals for a more sensitive response. Note that for calibrating servo position there are available servo protractors [16]. Further to this, reducing loop frequency may elicit better stabilisation, reducing the likelihood of microcontroller clock interference.

During the investigation the angular velocity of the flywheel decreased with increasing precession angle of the CMG due to a component of the flywheel weight acting on the motor axis, requiring more power to maintain axle RPM. This, combined with the decreasing stabilisation component of the gyro torque (see equation 3), reduced the magnitude of the restoring force. This prevents the system from recovering at higher roll angles, resulting in destabilisation. Further work would benefit from quantifying RPM change with precession angle, controlling the motor signal to counter it. This would require a separate microcontroller as accurate control relies heavily on clock timing.

5.2 Further Investigation Objectives

There are a number of aspects of this project that could be investigated further, improvements that should be made, and retrospective changes in the way it was carried out. To properly develop a functioning active system for a motorcycle future work on the project must involve control for a moving vehicle with a greater than 0 centre of mass, requiring a full understanding of the more complex dynamics. This would need a non-linear control system, for which the suitability of a PID system would have to be assessed, likely requiring the development of a new control scheme or algorithm adaptation to include a non-linear term. This is beyond the scope of the PID library. Further to this, the behaviour of the vehicle would vary with velocity, requiring an adaptive control scheme. If used, the PID algorithm is compatible with variable tuning dependent on system state. Continued development to include directional changes requires full understanding of single-track steering dynamics (see [6]), with far more variables to consider. At low velocities the position of the bike would usually be upright with steering used to change direction, but at high speeds a significant rotation of the handlebars is dangerous with steering achieved by counter steering and leaning. These scenarios require calibrated response regimes from the CMG, which would be configured to allow leaning, increasing control code complexity. This problem is avoided by Lit Motors by using the CMG to control direction, with the additional advantage of reducing changes in lateral mass distribution.

6 Conclusions

A fully independent active gyroscopic stabilisation system was successfully developed for use in a single-track vehicle. A stabilisation around the equilibrium within (1.11 ± 0.04) degrees was achieved using a flywheel with a moment of inertia of $(19850 \pm 253)\text{gcm}^2$ and a PID-based control scheme under test conditions. The test system was converted for use in a wheeled vehicle and demonstrated stabilisation for (3.797 ± 0.327) seconds. The suitability of the Genuino 101 for producing and interpreting positional data was confirmed and controlled servo and DC motors, writing all data to SD. The active system was verified to be superior to a passive regime, which failed to fully stabilise the vehicle.

Acknowledgements

Thank you to the lab technicians at the University of York for providing the equipment and space to carry out the experiment. I would also like to give special thanks to Dr. Phil Lightfoot for designing and constructing the experimental framework, flywheels and for any and all support and advice throughout the project.

Programs Used

FreeCAD
DataStudio
Arduino IDE

References

- [1] BMW. THE BMW MOTORRAD VISION NEXT 100.. <https://news.bmw.co.uk/article/bmw-motorrad-vision-next-100/> (accessed 13/10/16).
- [2] Department for Transport. Infrastructure and Cyclist Safety. https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/3731/infrastructure-and-cyclist-safety.pdf (accessed 13/10/16).
- [3] Kim, D.K.Y., Miller, E.. Gyroscopic system in vehicle suspension. <https://www.google.com/patents/US8919788>; US Patent 8,919,788, 2014
- [4] Brennan, L. Means for imparting stability to unstable bodies, [shhttps://www.google.com/patents/US796893](https://www.google.com/patents/US796893), US Patent 796,893, 1905
- [5] H. T. Eddy. The mechanical principles of brennan's mono-rail car. Journal of the Franklin Institute1910; 169(6): . <http://www.sciencedirect.com/science/article/pii/S0016003210900045> (accessed 13/10/16).
- [6] S. C. Spry, A. R. Girard. Gyroscopic Stabilization of Unstable Vehicles: Configurations, Dynamics, and Control. <http://www.umich.edu/arclab/macccs/media/gyrovehicle.pdf>: University of Michigan; 2008.
- [7] H. Yetkin, S. Kalouche, M. Vernier, G. Colvin, K. Redmill, U. Ozguner. Gyroscopic Stabilization of an Unmanned Bicycle. : Carnegie Mellon University; 2014. http://www.andrew.cmu.edu/user/skalouch/docs/ACC_2014.pdf (accessed 13/10/16).
- [8] Lit Motors Corporation. C-1 Lit Motors. <http://litmotors.com/c1/> (accessed 12/10/16).
- [9] Kim, D.K.Y. and Bretney, K. and Shao, A. and Tsang, A.L.. Electronic control system for gyroscopic stabilized vehicle. <https://www.google.com/patents/US8532915>; US Patent 8,532,915, 2013
- [9] Kim, D.K.Y., Millett, M.R.. Vehicle collision mitigation system. <https://www.google.com/patents/US8930128>; US Patent 8,930,128, 2015
- [11] Kim, D.K.Y. and Bretney, K. and Tsang, A.L.. Gyroscopic stabilized vehicle. <https://www.google.com/patents/US8706390>; US Patent 8,706,390, 2014
- [12] Joseph W Schenk. Development of Active Gyrostat Control Systems for land and space based applications (MPhys Project Dissertation). Department of Physics: University of York; 2015.
- [13] Sammy Jackson. Development of a Gyrostat Control System for Space Vehicles (MPhys Project Dissertation). Department of Physics: University of York; 2016.
- [14] Overlander Batteries. 2200mAh 2S 7.4v 35C LiPo Battery - Overlander Supersport Pro.

<https://www.overlander.co.uk/batteries/lipo-batteries/sport-2200-2s-7-4v-25c.html> (accessed 08 May 2017).

[15] Sebastian O.H. Madgwick. An efficient orientation filter for inertial and inertial/-magnetic sensor arrays. University of Bristol: x-io Technologies; 2010.

http://x-io.co.uk/res/doc/madgwick_internal_report.pdf (accessed 08 May 2017).

[16] Servo City. Servo Protractor. <https://www.servocity.com/servo-protractor> (accessed 08 May 2017).

7 Appendix

```

// List of all libraries needed
#include <SPI.h>
#include <SD.h>
#include <Servo.h>
#include "CurieIMU.h"
#include <MadgwickAHRS.h>
#include <PID_v1.h>

// All global scope variables
const int chipSelect = 10;                                // SD card CS pin
int servo1pin=9;                                         // Servo1 PWM pin
int mot1pin=5, revs=170;                                  // Motor1 PWM pin
unsigned long microsPerReading, microsPrevious;           // Looptime variables
float accelScale, gyroScale;                             // Scale factors
double Setpoint, Input, Output;                           // PID variables

// PID Tuning Variables
double Kp = 4.00;
double Ki = 0.00;
double Kd = 0.040;

// Used objects
Madgwick filter;
Servo servo1, servo2;
PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);
File myFile;

void setup() {
    // Program will not run without SD card
    if (!SD.begin(chipSelect)) {
        return;
    }

    // Write dataset header
    myFile = SD.open("Final.txt", FILE_WRITE);
    if (myFile) {
        myFile.print("Motor_Speed:");
        myFile.println(revs);
        myFile.print("Tuning:");
        myFile.print(Kp); myFile.print(",");
        myFile.print(Ki); myFile.print(",");
        myFile.println(Kd);
        myFile.print("Time"); myFile.print("\t");
        myFile.print("Pos"); myFile.print("\t");
        myFile.println("Servo");
        myFile.close();
    } else {
        return;
    }
}

```

```

}

// Attach objects
servo1.attach(servo1pin);
pinMode(mot1pin, OUTPUT);

// Function to bring flywheel up to speed
motorstart(revs);

// PID Settings
myPID.SetSampleTime(1);
myPID.SetOutputLimits(-86,86);
Setpoint = 0.00;
myPID.SetMode(AUTOMATIC);

// Position Data
CurieIMU.begin();
CurieIMU.setGyroRate(25);
CurieIMU.setAccelerometerRate(25);
filter.begin(25);
CurieIMU.setGyroRange(250);
CurieIMU.setAccelerometerRange(2);
}

void loop() {
    int aix, aiy, aiz;
    int gix, giy, giz;
    float ax, ay, az;
    float gx, gy, gz;
    float roll, t;
    int servopos = 90;
    unsigned long microsNow;

    // check if it's time to read data and update the filter
    microsNow = micros();
    if (microsNow - microsPrevious >= microsPerReading) {

        // Raw Data from CurieIMU
        CurieIMU.readMotionSensor(aix, aiy, aiz, gix, giy, giz);

        // Convert Raw Data to Gravity and Degrees/Second
        ax = convertRawAcceleration(aix);
        ay = convertRawAcceleration(aiy);
        az = convertRawAcceleration(aiz);
        gx = convertRawGyro(gix);
        gy = convertRawGyro(giy);
        gz = convertRawGyro(giz);
        t = millis();

        filter.updateIMU(gx, gy, gz, ax, ay, az);
        roll = filter.getRoll();
    }
}

```

```

Input = roll;           // Input position data into PID
myPID.Compute();        // Calculate a PID response

// Limit servo output to prevent slowdown of servo against endpoint
if (servopos+Output <=86 || servopos+Output >=-86){
    servol.write(servopos+Output);
}
else if (servopos+Output <=-86){
    servol.write(-86);
}

else if (servopos+Output >=86){
    servol.write(86);
}

// Short time for servo to move
delay(10);
// Drive flywheel motor
analogWrite(mot1pin, revs);

// Write all data to file
myFile = SD.open("Final.txt", FILE_WRITE);
if (myFile) {
    myFile.print(t);
    myFile.print("\t");
    myFile.print(roll);
    myFile.print("\t");
    myFile.println(Output);
    myFile.close();
} else {
    return;
}

// Increment Time, Keep Known Pace
microsPrevious = microsPrevious + microsPerReading;
}
}

float convertRawAcceleration(int aRaw) {
    // +/-2G Range Maps to Raw Value of +/-32768
    float a = (aRaw * 2.0) / 32768.0;
    return a;
}
float convertRawGyro(int gRaw) {
    // +/-250 Degrees/Seconds Range Maps to Raw Value of +/-32768
    float g = (gRaw * 250.0) / 32768.0;
    return g;
}

void motorstart(int revs){

```

```
for (int rpm = 0; rpm <= revs; rpm += 10) {
    servol.write(90);
    delay(10);
    analogWrite(mot1pin, rpm);
    delay(500);
}
}
```