Development of Active Gyrostat Control Systems for land and space based applications.

Joseph W Schenk

Department of Physics
University of York
MPhys Project Dissertation

Supervisor: Dr Phil Lightfoot

js1115@york.ac.uk

May 2015

Development of Active Gyrostat Control Systems for land and space based applications.

Abstract.

The aim of this project was to investigate the properties of gyrostat control systems and culminate in a functioning active system highly flexible in application. The flywheel design and centre of mass for the gimbal axis and body are scrutinised for a passive system as stability is found to be dependent upon all three. This resulted in a flywheel with a moment of inertia of $(2870 \pm 90)\mathrm{gcm}^2$ made of aluminium. The active system consisted of an MPU6050 accelerometer, Arduino UNO microprocessor and servo, hence creating a real-time feedback loop. This is an automated version of Brennan's earlier manual system. The automation required the microprocessor to be coded and optimised, this code was based largely on effects seen during passive system testing. The active system successfully stabilised a $(405.8 \pm 0.1)\mathrm{g}$ model, similar in geometry to an unstable vehicle. Although on a smaller scale, the system is comparable to the work of Lit Motors and their two wheeled car. Due to its success, despite the MPU6050 being sensitive to vibrations, the system was later demonstrated in a marine application, stabilising a $(2.46 \pm 0.01)\mathrm{kg}$ fibreglass boat against a rolling effect. With small amounts of code modification the system would be applicable in satellite setups in replacement of the ten times heavier reaction wheel.

1. Introduction

A torque may be applied using several methods, however a self-contained system, as required in space based applications limits design to a moment wheel, mass ejection or gyroscopic control. Gyrostatic control systems offer up to 10 times more torque sustainably, using comparable amounts of mass [1] to a moment wheel, making them ideal for satellite attitude control and other stabilisation applications. They don't require refilling as in mass ejection, and their usage time between charges is considerably longer than mass ejection fill intervals.

Gyrostatic stabilisation systems have two main variants, the passive system and the active system. These stabilisation systems vary greatly in their torque output and their function, even beating heart surgery may soon be done using gyro stabilisation as seen in [2] by Gagne et al. The passive system shows limited application mainly in ground based systems, this systems properties have been investigated later and are shown to be fully functional. The active system required more development, but the stability is substantially improved and due to the control system now present it can be used in satellites, to control attitude and also in marine ships to control pitch and roll. This active system involves turning the gimbal axis, with flywheel and motor, which exerts a stabilising torque. Dependent upon the servo used to turn the gimbal axis a much larger stabilising torque can be exerted relative to the passive system, 10 times is easily achievable.

The application of rotational tops dates back to 1752, when James Short implemented an idea of John Serson and created a navigational device, measuring the location of stars relative to ships at sea [3]. On 4th September 1877 the first implementation of gyroscopic control occurred in Brennan's steerable torpedo [4]; this controlled the attitude of the torpedo and so a rudder system was used to control direction. Brennan later begun developing a monorail, part patented in 1903, Townsend et al [5]. Further notable development occurred in 1905 [6] as Brennan published a paper featuring a manual active system for improved stability [7]. This involved a driver turning the gimbal axis manually using a long bar, this would not pass current safety testing. In 1907 the monorail was exhibited [8], where many people were able to stand on one side of the monorail and it corner successfully. Whilst Brennan's monorail was a success, the financial backing for large scale commercial use was not available and so his idea was rejected. The implementation in early 1900 was not possible but now with improved manufacturing techniques and the idea being revisited due to its ability to cross steeper terrain and also thin bridges [9]. Modern manufacturing techniques can be considered more economically viable during the construction phase, a similar scale infrastructure modification has been discussed by Ormiston and Emerston in [10].

Later, ideas have been developed such as the concept car, the Ford Gyron of 1961 [11] which required a 180lb gyroscope. But most recently the Lit motors patent of 10th September 2013 which features the electronic control system for an active gyro stabilisation system [12]. Lit motors patent features arguably the most advanced system and is comparable to the active system developed here. Their proposition is that two gyrostabilisers will work side by side. Upon vehicle acceleration some of the flywheel energy could be used and during braking the energy could be recovered into

the flywheels. This design is based upon the fact that at higher speeds the wheels will have higher angular momentum and so the vehicle will be more stable and require less gyroscopic stabilisation. More physical implications such as vehicle width and height are discussed by Kidane et al [13]. Townsend, Shenoi and Murphy [14] investigate the use of gyro stabilisation in replacement of keels and other control surfaces, similarly investigated by Adams and McKenney [15]. It is from these that the plan was developed to investigate basic passive properties and later develop a system with superior physical capabilities when scaled down. This was concluded in the form of a fully automated adaptive active system.

It will be shown later that the stability of the passive system is dependent upon flywheel rotation rate, moment of inertia of the flywheel, rate of change of the gimbal axis angle, mass of the body, centre of mass of the system and also centre of mass of the gimbal axis. These findings are of great importance as it allows a more effective approach to the active system. The fundamental gyro stabiliser theoretical considerations as seen later, and where a bicycle is cornering as in [16], are confirmed.

The active system was controlled using an accelerometer and microprocessor, programmed in a C based language in the Arduino user interface. The Arduino recorded angle of body lean from the accelerometer and then sent a signal to a servo to control servo angle. This Arduino feedback loop makes the setup particularly effective.

The models used were developed from first principles, with the variables mentioned about the passive system variable by some means. This involved designing a flywheel using a computer aided design package called Pro-desktop, then adapting these designs to suit the machining processes available.

The main source of error during testing resulted from measurement of the gimbal and body lean angles, variable resistors proved reliable. During the active system development the MPU6050 accelerometers were found to be very sensitive to vibration. This was attempted to be damped using a pulley system with the chip suspended on a separate support to the flywheel, hence isolating vibration, this was successful. Consideration was also given to the source of vibration, it is still believed that it was due to flywheel imbalance, given the constraints of the equipment used to mill the flywheel vibrations were minimal. The wiring system was complex during active system development and although loose connections were rare, they did cause problems with the MPU6050. Careful labelling of wires made fault finding easier. To understand why the lean angles are so important and understand how the system functions the theory must now be examined.

2. Theoretical background of gyrostatic stabilisation

An understanding of the basic dynamics of the system, moment of inertia of the flywheel, torque required to counter balance toppling torque and how to find the toppling torque are required, equations 1 – 6 cover these fundamental concepts. First consider the torque:

$$\boldsymbol{\tau} = \mathbf{r}\mathrm{x}\mathbf{F} \qquad \text{(equation 1)}$$

Using equation 1 and figure 1 below, a more explicit form to the exact problem can be found as in equation 2. Figure 1 gives a definition of some of the variables discussed throughout.

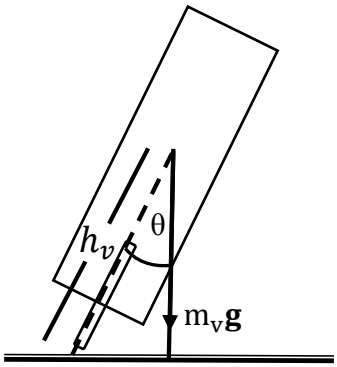


Figure 1: The basic body of the model represented as a rectangle, showing the toppling or lean angle θ and the vehicle centre of mass height.

So the toppling torque is now trivial to find as in equation 2:

$$\boldsymbol{\tau}_{\mathrm{gravity}} = \mathrm{h_v m_v}\mathbf{g}\sin(\theta) \qquad \text{(equation 2)}$$

Where the total vehicle centre of mass height is $\mathrm{h_v}$, $\mathrm{m_v}$ is the vehicle total mass and **g** the acceleration due to gravity. This torque can be divided into two components, g for gimbal and b for body. Note that as the gimbal axis rotates $\mathrm{h_v}$ becomes a variable; the effect is however small. During calculations and experimentation the simplification as in equation 2 was used. From equation 2 it is useful to note the rate of torque change with theta, equation 3, which gives an insight into how the torque changes rapidly at small toppling angles. The rate of torque change with time can be seen in equation 4.

$$\frac{\mathrm{d}\boldsymbol{\tau}_{\mathrm{gravity}}}{\mathrm{d}\theta} = \mathrm{h_v m_v}\mathbf{g}\cos(\theta) \qquad \text{(equation 3)}$$

$$\frac{\mathrm{d}\boldsymbol{\tau}_{\mathrm{gravity}}}{\mathrm{dt}} = \mathrm{h_v m_v}\mathbf{g}\cos(\theta)\dot{\theta} \qquad \text{(equation 4)}$$

Now consider how to create a counter-torque to stop the vehicle falling. To understand flywheel geometry first consider some fundamental mathematics. Equation 5 shows the kinetic energy of a rotating object with rotation rate **ω.**

$$\mathrm{E_K} = \frac{1}{2}\mathrm{I_F}\boldsymbol{\omega}\cdot\boldsymbol{\omega} \qquad \text{(equation 5)}$$

The moment of inertia of a rotating disk can be found using equation 6, note R the radius and its inertia relation compared to the mass dependence.

$$\mathrm{I_F} = \frac{1}{2}m_f\mathrm{R}^2 \qquad \text{(equation 6)}$$

Equation 6 forms the basis for the flywheel design investigation and the CAD that proceeds any testing.

Now the basic flywheel mathematics have been treated the full system can be considered. This is done using the lagrangian, L = K – U, where K and U are the kinetic and potential energies respectively. For the most thorough treatment the two main rigid components should be treated respectively, the flywheel, f and vehicle body, b. A turning motion will also be considered with a yaw angle κ as seen in figure 2. Treatment without κ can be seen in [17].

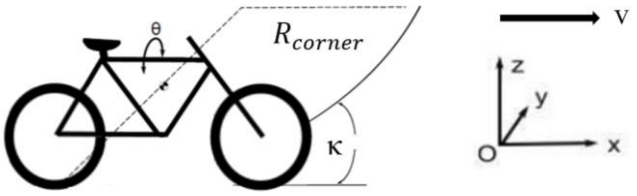


Figure 2: Diagram showing the yaw angle, κ and the direction of travel with speed v, note also the axis direction.

Next the total kinetic energy for the system should be found. Note that the system is assumed to not be counter-steering as at higher speed this is more to initiate the turn [18]. This is not of concern here and is an investigation in itself seen in [19]. Figure 2 is derived from [16] and the mathematics are drawn from [16] and shall later be simplified for the system in question. The total vehicle kinetic energy is shown in equation 7, note the toppling component.

$$K_b = \frac{1}{2}m_b\left(v^2 + \dot{\kappa}^2\sin^2(\theta)h_b^2 + \dot{\kappa}v\sin(\theta)h_b + \dot{\theta}h_b^2\right) + \frac{1}{2}\left(\dot{\theta}^2 I_{bx} + I_{by}\dot{\kappa}^2\sin^2(\theta) + I_{bz}\dot{\kappa}^2\cos^2(\theta)\right) \qquad \text{(equation 7)}$$

In equation 8 consider the gimbal axis kinetic energy, note the translational energy and the rotational kinetic energy components. Here φ is the gimbal lean angle.

$$\begin{aligned} K_f = {} & \frac{1}{2}m_f\left(v^2 + \dot{\kappa}^2\sin^2(\theta)h_f^2 + \dot{\kappa}v\sin(\theta)h_f + \dot{\theta}^2h_f^2\right) \\ & + \frac{1}{2}\left(I_{fx}\left(\dot{\theta}^2\cos^2(\varphi) + \dot{\kappa}^2\cos^2(\theta)\sin^2(\varphi)\right.\right. \\ & - 2\dot{\theta}\dot{\kappa}\cos(\theta)\cos(\varphi)\sin(\varphi) \\ & + I_{fy}\left(\dot{\kappa}^2\sin^2(\theta) + \dot{\varphi}^2 + 2\dot{\kappa}\dot{\varphi}\sin(\theta)\right) \\ & + I_{fz}\left(\dot{\theta}^2\sin^2(\varphi) + \dot{\kappa}^2\cos^2(\theta)\cos^2(\varphi) + \omega^2 + 2\omega\dot{\theta}\sin(\varphi)\right. \\ & \left.\left. + 2\dot{\theta}\dot{\kappa}\cos(\theta)\sin(\varphi)\cos(\varphi) + 2\omega\dot{\kappa}\cos(\theta)\cos(\varphi)\right)\right) \end{aligned} \qquad \text{(equation 8)}$$

Using the lagrangian the toppling angle acceleration can be found. For ease equation 9 has been laid out into its constituent parts, gravitational force component, centrifugal effect component, turning or pivot component and precession effect component , $A_{grav}, A_{cen}, A_{piv}$ and A_{pre} respectively.

$$\ddot{\theta} = A_{grav} + A_{cen} + A_{piv} - A_{pre} \qquad \text{(equation 9)}$$

Where:

$$A_{grav} = \frac{(m_b h_b + m_f h_f) g \sin(\theta)}{I_{bx} + I_f + m_b h_b^2 + m_f h_f^2}$$

$$A_{cen} = \frac{\frac{v^2}{R_{corner}} (m_b h_b + m_f h_f) \cos(\theta)}{2(I_{bx} + I_f + m_b h_b^2 + m_f h_f^2)}$$

$$A_{piv} = \frac{\frac{v^2}{R_{corner}^2} (I_{by} - I_{bz} + I_f + m_b h_b^2 + m_f h_f^2) \sin(\theta) \cos(\theta) + I_f \frac{v}{R_{corner}} \dot{\varphi} \cos(\theta)}{I_{bx} + I_f + m_b h_b^2 + m_f h_f^2}$$

$$A_{pre} = \frac{I_f \omega \left(\dot{\varphi} + \frac{v}{R_{corner}} \sin(\theta) \right) \cos\varphi}{I_{bx} + I_f + m_b h_b^2 + m_f h_f^2}$$

This equation and its constituent components can be seen in work of Yetkin and Ozguner [16] for further reference Spry and Girard [6] have a comparably thorough treatment. For the systems used in this experimental setup equation 9 can be reduced to equation 10, the net torque.

$$\tau = h_v m_v \mathbf{g} \sin(\theta) - I_f \omega \dot{\varphi} \cos(\varphi) \qquad \text{(equation 10)}$$

This assumes the centre of mass height is fixed regardless of gimbal lean angle. This assumption is suitable as gimbal rotation invokes little change in total centre of mass height. Furthermore equation 10 assumes the system has no translational motion, which is suitable for use in a static environment.

To find h_v a moments method is used as seen in figure 3 and then equation 11 can be found using some basic mathematics.

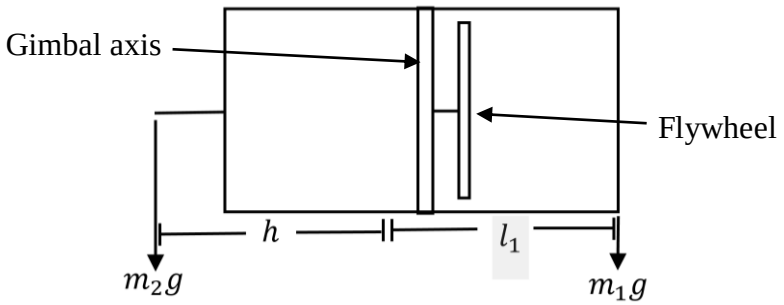


Figure 3: Diagram of the simple model on its side, where h is the COM height.

$$h = \frac{m_1 L}{m_1 + m_2} \qquad \text{(equation 11)}$$

Equation 11 now allows the centre of mass height to be found as all the values in the equation are measureable, here L is the total length of the model or the sum of h and l_1.

2.1. Initial testing and component design

Initial testing involved confirming proportionality as in equations 2, 9 and 10; varying flywheel rotation rate and flywheel mass and then noting its effect on stability. This initial testing was used to familiarise with the processional behaviour of the passive system before attempting to design any components for the later passive system

testing, where more detailed investigations into centre of mass took place. Design criteria ensured the initial system was simple and lightweight and so a low power motor was used.

From equation 10 it is seen that to remain stable τ would be negative. The moment of inertia of the flywheel is therefore key and is governed by equation 6. When designing the flywheel it was necessary to minimise the mass of the flywheel, m_f to keep m_v small and maximise the radius of the flywheel, R. So a design with a lightweight centre and a higher mass circumference was preferable. Equation 12 shows the moment of inertia of a hoop, the extreme of moving the mass of a disk to the outer surface.

$$I_H = m_f R^2 \qquad \text{(equation 12)}$$

For a fixed mass and radius it can be seen that the inertia of a hoop is double that of a disk, this will be used during the CAD design to maximise the moment of inertia. To measure the moment of inertia a rotational experiment involving dropping mass and measuring rotational acceleration as the mass falls was carried out. This has errors from measuring the rate, which was computer controlled, also in measurement of the mass dropped and in string winding radius variation as the mass falls. Equation 13 shows how the moment of inertia of the flywheel was found. In equation 13 m is the dropped mass, r is the radius of the string windings and a_T is the angular acceleration.

$$I = \frac{r^2 m(g-\alpha)}{a_T} \qquad \text{(equation 13)}$$

Motor output relative to the mass of the flywheel is important. A less massive flywheel means a smaller, less powerful, but lighter motor is required, hence minimising m_v. Initially, a lightweight motor was used but after testing it became apparent that a motor with a higher sustainable rotation rate was required. A graph of flywheel rotation rate against power will show a curve, with a gradient dependent upon how close to maximum rotation rate the system is operating. Motor suitability was judged on these curves.

The body of the system, as with the gimbal axis should have varying centre of mass heights, i.e. from one centre of mass height to another, thus making centre of mass measurements, varying h_v, more simple. It was expected and shown that greater h_v led to a more unstable system.

Measurement of angles was primarily made at later stages using an MPU6050 but contained within the system were also two variable resistors, one on the gimbal lean axis and one on the body lean axis. These variable resistors were used to measure gimbal and body lean angle. For the passive component this forms the main angle measurement method. Measurement of these angles will be the primary source of error, measurement of the resistance and also obtaining accurate chip data can be complex.

2.2. Passive testing

Having designed the equipment following the criteria from section 2.1 passive testing could begin. Theoretically this section follows section 2.1 closely, examining proportionalities, but more importantly the emphasis was on gaining information about physical behaviour for use during the active system setup.

A more unstable gimbal system should give a larger $\dot{\varphi}$, which will result in a more stable system according to equation 10. It would however be expected that this will at some point become so unstable it causes instability in the gimbal axis.

2.3. Arduino Uno and servo control

The Arduino Uno is programmed in a C based language. By using the Wire library it is possible to get the Arduino and the MPU6050 communicating. The Servo library should also be used to save time getting the servo to communicate, so that the servo is initially testable directly from code or from MPU6050 data.

It is now known that MPU6050 is susceptible to vibration and damping should be used. Code can be used to minimise noise and hence increase accuracy. The magnitude of this noise reduction is variable by implementing a memory function, where the code stores previous angle values.

2.4.Active system development

With the MPU6050 indirectly connected to the servo through an Arduino UNO microprocessor the toppling or lean angle, θ, can be used to input a servo torque on the gimbal axis. This setup triggers an increase in $\dot{\varphi}$ in response to a toppling torque, hence stabilising the system. This system should require lower ω to obtain stability compared to the passive system due to this gimbal axis control, seen in equation 10.

Various inputs can be used for the servo angle, initially for testing it was matched to the toppling angle, making any fault in programming or hardware clear. A modified version of equation 10 is finally used including constants that can be changed to vary sensitivity to body toppling angle. Dependence of servo input code on rate of toppling angle change and also instantaneous toppling angle are investigated, due to precession effects witnessed during passive testing.

3. Method

3.1. Initial testing and component design

Using equation 6 the moment of inertia of a compound flywheel consisting of two compact discs and a backing alloy component was found. The alloy component was found to have a moment of inertia of $(5.1 \pm 0.1)\text{gcm}^2$ which was found by measuring the radius and measuring the mass, repeated for CDs the total moment of inertia was found to be $(545.1 \pm 0.1)\text{gcm}^2$.

Throughout the project the centre of mass height was found using moments as in figure 3 and equation 11. The centre of mass of this initial model was found to be $(10.5 \pm 0.3)\text{cm}$ where the large error occurs due to parallax when measuring length using a ruler. To confirm the torque methods reliability the model was simply balanced on its centre of mass height and the value of h_v taken as $(10.5 \pm 0.5)\text{cm}$.

The two methods are in agreement although the error on the torque method is smaller and so this was the method used in all later experimentation.

Figures 4 and 5 are diagrams of the lightweight model used in this initial testing. Figure 4 shows the gimbal axis out of page and figure 5 shows the body lean axis out of page.

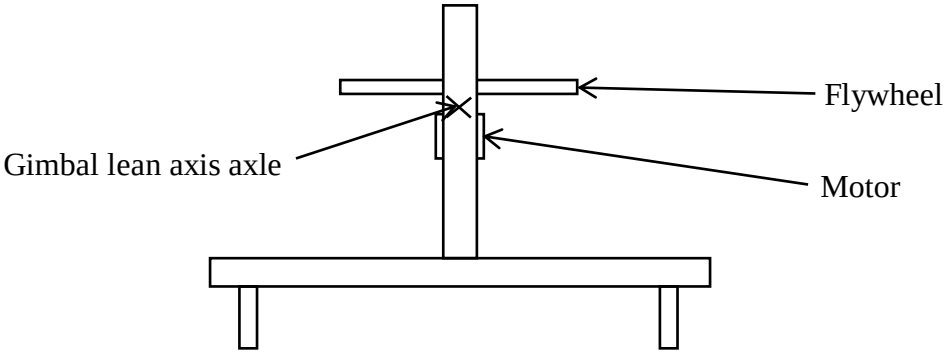


Figure 4: Lightweight model looking along the gimbal axis, the cross denotes the gimbal pivot point axle.

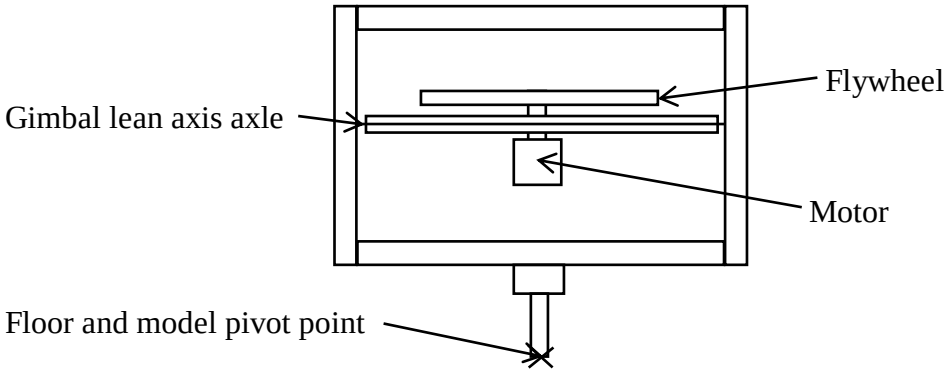


Figure 5: Lightweight model looking along the body lean axis, the cross denotes the pivot point on the floor.

The small model motor was used in this lightweight model and rotation rate measured against power in. Power was taken from the power supply using P = IV. The flywheel rotation rate was measured using a tachometer, a small dark strip was coloured on the CDs to allow its use. Due to the gimbal axis being held still here the rotation rate was measured with an error of ± 50 RPM, which given the large values of several thousand rotations per minute tested, this error is reasonable. When starting the initial stability tests the body was held by hand and then released. This introduced additional error as the model in some releases would be closer to upright than others, by methodical release this was minimised.

Flywheels require safety and as such a safety cage was constructed. It was constructed in steel with sliding observation Perspex on two sides, enabling easy access. The basic model was tested at various rotation rates and time until the model fell over was taken, using this method a point of stability was found. Rotation rate and gimbal angle were then obtained. The gimbal angle was measured using a protractor. Data obtained

was purely used to gain a greater understanding of the dynamics and so the large error associated with the angle was not of concern.

Flywheel geometry was investigated using equation 6 and 12, hence a flywheel with mass near the circumference would be most efficient. Flywheel geometry involved consideration of equation 10, so that τ was negative and stability was witnessed at an achievable flywheel rotation rate. Initial estimates of mass 100g with a centre of mass at 8cm were made. The centre of mass was adjustable in the final model so this estimation is suitable.

Due to limitations of the materials available the flywheel needed to be 5mm thick. This was a machined depth and so has a small error. It was suggested initial investigations occur with radius of (6.0 ± 0.1)cm. A micrometre was used for radius measurements and so the error was related to finding the line through the centre point and not the measuring scale. Two materials were recognised as suitable, steel and aluminium, with densities of $(8.0 \pm 0.1)\text{gcm}^{-3}$ and $(2.7 \pm 0.1)\text{gcm}^{-3}$ respectively. From the motor experiment in graph 16 it was decided an operating rate of 6000RPM should be used. Assuming a plain disk as in equation 6 it was found a mass of 62.6g was required given an inertia of 1127gcm^2. The masses for the steel and aluminium disks were found to have theoretical masses of (452 ± 6)g and (153 ± 6)g respectively, this is much larger than the required 62.6g and so as it is lighter aluminium was chosen as the material.

Various drilled hole size mass savings were found and an approximation of two different disk thicknesses, i.e. the centre was of a different density out to a distance $r = 4.1$cm from the centre, and the remainder 1.9cm was a different density. This density variation would be achieved by creating holes in the flywheel at the design stage. During machining however it was more effective to create a recessed section in the flywheel. This proposed flywheel design can be seen in Figure 6.

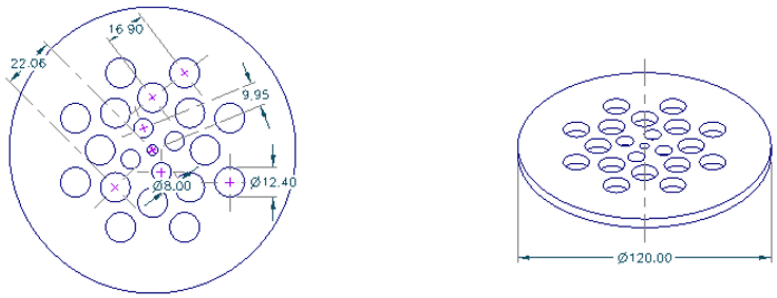


Figure 6: The proposed design of flywheel using computer aided design, the measurements are given in millimetres.

The theoretical moment of inertia for this flywheel was found to be $(2515 \pm 13)\text{gcm}^2$. This flywheel added to the proposed system raises the centre of mass to (0.11 ± 0.01)m. Inputting this into equation 10 yields a $\tau = 0$ value at (6484 ± 611)RPM upto a body lean angle of 45^{o}. A similar investigation with steel was also made, but aluminium was chosen and so these figures are of greater importance.

To find the constructed flywheel moment of inertia an angular velocity measurement was made using a sealed unit. A mass of (8.47 ± 0.01)g was attached to a string wrapped around the pulley. Also attached to the pulley was the flywheel of mass (137.98 ± 0.01)g and radius (60 ± 1)mm. Note the pulley with string wrapped around is of radius r, (12.5 ± 0.5)mm. The error is large because as the number of string windings increases so r increases. Using equation 13 the moment of inertia of the flywheel was found.

3.2. Passive testing

An outline of the new alloy bodied system is shown in figures 7 and 8 depicting the gimbal axis view and the body axis view respectively. Figure 9 shows the actual passive model.

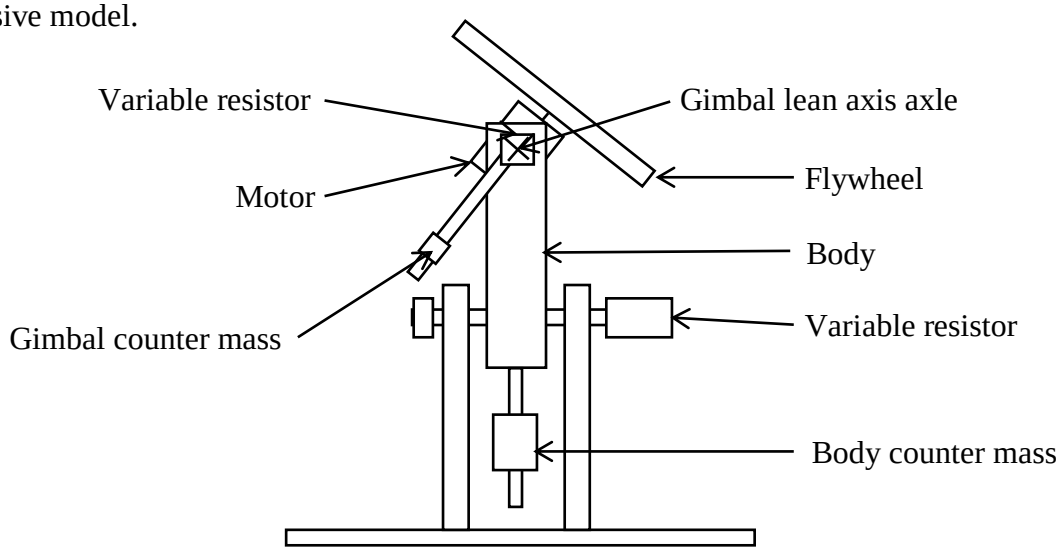


Figure 7: Alloy bodied system prior to development, viewed looking along the gimbal axis. Note the variable centre of mass of the body and the gimbal axis. The cross here denotes the gimbal pivot point axle.

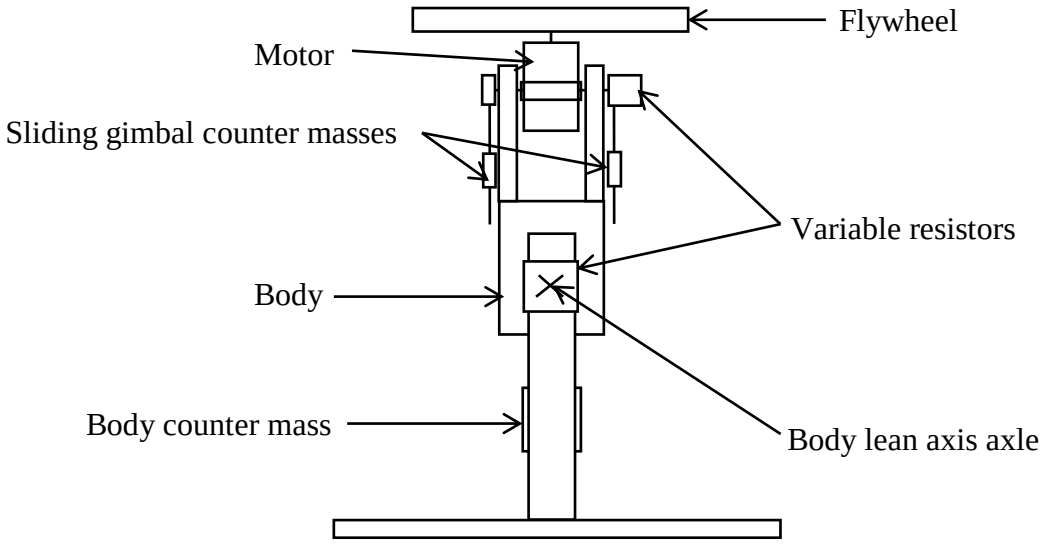


Figure 8: Alloy bodied system prior to development, viewed looking along the body lean axis. Note the two sliding gimbal axis masses. The cross here denotes the body lean pivot point axle.

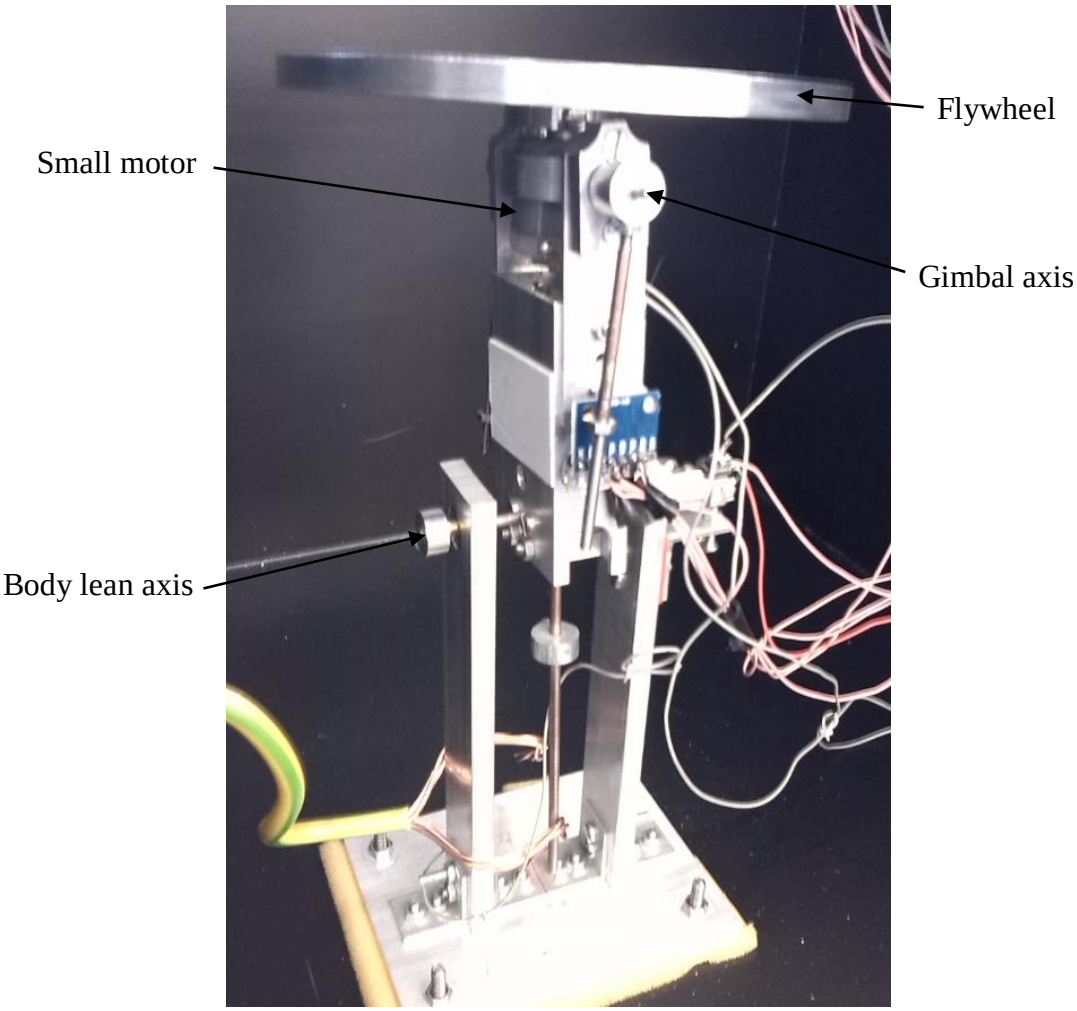


Figure 9: Passive model without development, note in this image the counter masses are not present.

Initially the new alloy bodied system with adjustable centre of mass was bolted directly to the safety cage. After the first run it became apparent due to the high rotation rate that the tiny imbalance of the flywheel caused large amounts of vibration. To reduce vibration effects foam was placed at all joins between the metal and Perspex, and between the alloy of the model and steel of the safety cage, bolts had lock-nuts added to stop the vibration working the mounting points loose. Holes were also drilled through the safety cage to allow access for the wiring of the circuitry. Furthermore a thick piece of mains earth wiring was used to aid the release of the system to improve safety. The complete setup can be seen in figure 10.

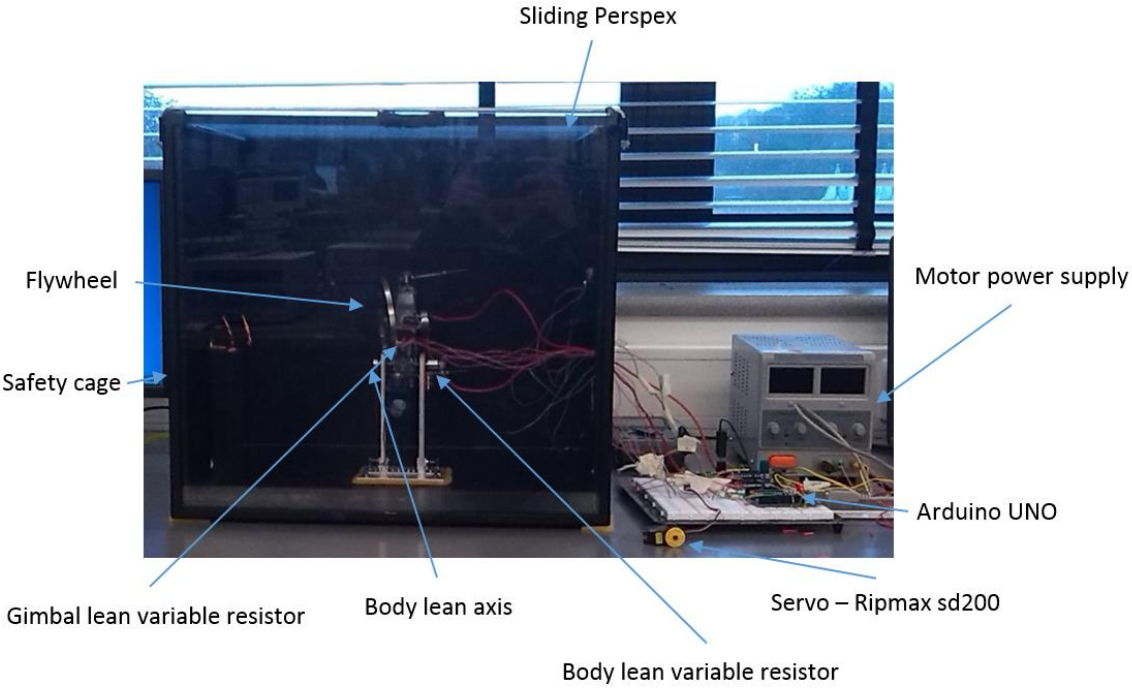


Figure 10: The complete system, note the gimbal lean angle here is large due to the flywheel mass and low resistance bearings.

In early runs resonance in the body of the model seemed a major fault and the gimbal seemed unresponsive, the precession effect was not initially witnessed. To increase responsiveness the centre of mass of the gimbal axis was raised to decrease its stability. To test the systems rigidity slower runs were made until approximately 2400RPM was used and the precession was witnessed.

It was realised after several runs that the small motor was unable to cope with these large amounts of torque being produced and so a Permax 400 motor was used. This new motor required significant redesign of the gimbal axis and its pivot points. The counter mass was also now adjustable for the body left and right, and so the system oscillates about an approximate zero degrees, prior to this oscillations were about an approximate fifteen degree toppling angle.

The MPU6050 did not respond well to vibration and so for passive testing the variable resistors were used. Initially a Metex data logger was used, it soon became apparent that this equipment was not able to record data fast enough and so Cassy lab, recording every 0.2s was employed.

With the flywheel motor switched off data was taken using the variable resistor against lean angle for both the gimbal and body lean axis. These graphs were heavily relied upon to move from units of kΩ to degrees. During every run at the second oscillation the motor RPM, voltage and current draw are recorded, this gathers data to confirm the Permax motor suitability. A comparison was made between loaded, the system in operation and unloaded, the system held steady. It was however noted that after one full cycle when the motor was started at (6450 ± 250)RPM rotation rate

dropped to (6162 ± 250)RPM. For angle against time plots curve form was found to be as in equation 14.

$$y = a\sin(bx + c)\exp(dx) + f \qquad \text{(equation 14)}$$

Graphs at six different rotation rates were plotted. Gimbal against body lean angle for two of these different rotation rates were plotted to demonstrate the phase difference and gain understanding of stability, as more points in a cycle denotes greater stability.

Now a (17.15 ± 0.01)g mass was added to the gimbal axis and moved so the optimum location for the mass is found, this would be where the exponential component of equation 14 was lowest. To find this point flywheel rotation rate was held constant at (5000 ± 250)RPM, producing reasonable stabilising torque without motor degradation.

It was assumed that the oscillation time period was constant, found using:

$$T = \frac{2\pi}{b} \qquad \text{(equation 15)}$$

Where b in equation 15 is the coefficient in equation 14 found from the graph's best fit curve. By differentiating the function and setting to zero a function of x in terms of the coefficients could be found for each graph. Then the time at which the maximum and minimum occurred was found and so the modulus of the true angle taken, having included f from equation 14 as necessary.

Plotting the maxima and minima points gave exponential curves. These exponential coefficients were then plotted against counter mass location, the turning point of this graph was found, this is the point where the system is most stable. The mass was not physically able to be set at this position so a different mass was used, which was set to create the same gimbal centre of mass height.

Using the torque method as in figure 3 a body centre of mass height was found to be (9.26 ± 0.2)cm. This would be varied using a mass of (324.2 ± 0.01)g to produce centre of mass heights at 2.7cm, 2.1cm, 1.8cm, 1.2cm and 0.8cm. It was expected that the exponential coefficient, d, would be lowest at lower centre of mass heights. Note the centre of mass height seems small compared to initial design estimation, but with the new motor system, the body and flywheel the combined mass is (405.75 ± 0.01)g and operating at (5000 ± 250)RPM it would be expected the system would be less stable with comparable h_v.

3.3. Arduino Uno and servo control

Initially familiarisation with the Arduino took place which involved making an LED turn on and off and using pulse width modulation hence creating a fade effect. Next the accelerometer and MPU6050 were linked to the system. The first lines of the code to initiate the chip were used from John Chi [20], these lines and the rest of the code can be seen in the appendix.

At this point the code printed numbers. These were related to angle but were not linear and so a graph was plotted to gain a function for angles in degrees in terms of these values. The real angle in degrees was measured using a protractor. At this time the chip was fixed to a breadboard and using the desk as a pivot, reading accurate values was straightforward.

The code to use all six axis in the chip is far more complex and so a decision to use only a single axis was chosen. Whilst slightly less versatile the code is less susceptible to problems. At this stage the chip was unable to differ between left and right, this is later developed. To help monitor the magnitude of these angles an LED was used, varying brightness with angle. A coded test system was also created in case a fault should occur with respect to the range of angles produced.

To obtain a polarity measurement to differ between left and right, two main possible solutions were considered. Firstly using the gradient of previous points, however, this became unreliable as increase in lean produces the same gradient regardless of orientation. This was then built upon and a zero point incorporated, should the system pass through the zero point zone, roughly 12 degrees, so the angle polarity would change. Setting up this system required a storage function and using if statements so as the system passes through zero and the gradient changes so the sign changes. A graph was plotted demonstrating its effectiveness.

At this point a servo was introduced to the system in the form of a Ripmax sd200, which has the same operation of the larger Sanwa servo fitted to the main model, however this is less bulky for testing. The operating values were found to be $(0 \rightarrow 180)°$, so 90° is the vertical and optimum angle.

3.4. Active system development

Initially the MPU6050 was unable to cope with the vibration, this led to large amounts of time spent in an attempt to dampen these vibrations using various mounting methods. The MPU6050s became functional after they were mounted to a pulley attached to a clamp stand which was linked to the model using a polymer belt, hence dampening more vibrations.

Now the MPU6050 can be linked to body lean angle and a servo attached to the gimbal axis. Initially the servo was to turn to the body lean axis angle, this proved difficult but it was achieved. Once again the MPU6050 proved too sensitive for this application, it is however believed with enhancement this sensitivity could be useful in increasing sensitivity/precision of lean angle values. The code was modified after several runs so that should the modulus of two previous values be greater than 15° apart then the last value is ignored, this acted as a dampener. Figure 11 is the complete active system, as successfully run.

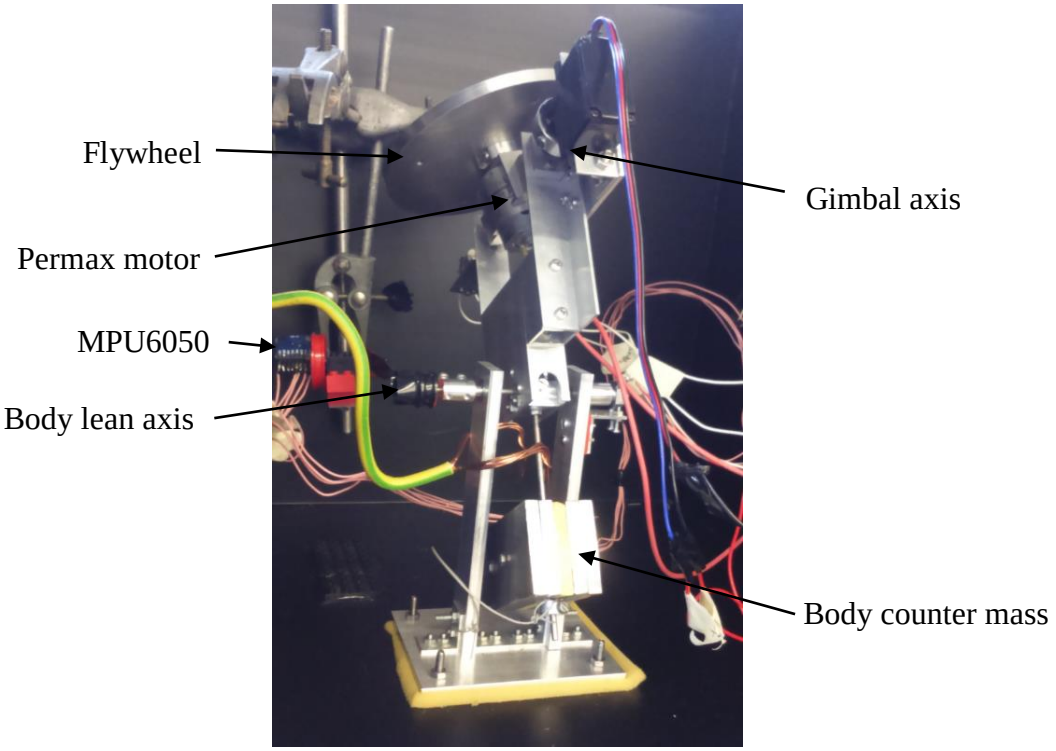


Figure 11: The active system, note the pulley, large servo, Permax motor and the clamp stands holding the MPU6050 which is detached from vibration.

Initially the Sanwa servo outputted the same values as those inputted, this was later developed in the form of a function closely related to equation 10 seen in code part 1. Note the safety feature on the limitation of lock angle, here called correction and the print action in the serial monitor.

```
constant = 0.5;                                          //Roughly mgh changable for different systems
torque = constant * sin(twist*(3.1415926535/180));       //Torque to be countered
Serial.print(" torque");
Serial.print(torque);
gimconstant = 0.02;                                      //Gimbal Iw. roughly 0.1 - 1
extra = 0.0002;                                          //Power variable
omega = ((torque) + extra)/(gimconstant);                //Extra - for more torque power
Serial.print("omega =");
Serial.print(omega);

correction = omega*(wait/100) + correction1;

if (correction < 30){
  correction = 35;
  Serial.print("Warning minimum lock 30 degrees");
}
if (correction > 150){
  correction = 145;                                      //To remove lock to lock, safety component
  Serial.print("Warning maximum lock 150 degrees");
}

Serial.print("correction gimbal angle=");
servo.write(correction);
Serial.print(correction);
```

Code part 1: Stabilising function with values to be varied to change the severity of the response of the servo, note also the safety feature in correction value limitation.

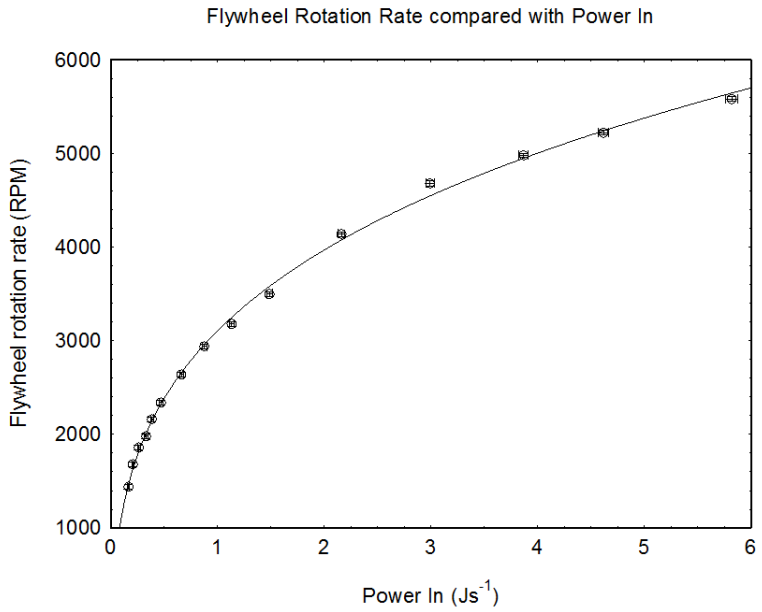
After the active systems success in a lab environment it was tested in a fibreglass boat where stability was witnessed, reducing the effect of waves and disturbances. The system responded, however with further development of the angle input, i.e. using a different chip so that response to angle change was more reliable and less susceptible to vibrations, so the systems effectiveness could be improved greatly. Vibrations were

limited using slow flywheel rotation rates. Rotation rate was fully variable in the boat using a speed controller to vary the power input to motor, this was in turn controlled via a receiver and remote control.

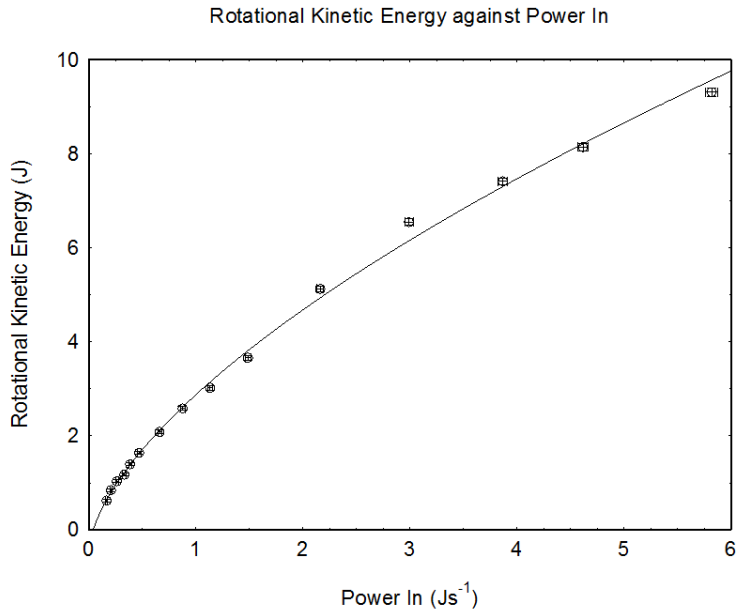
4. Results

4.1. Initial Testing and component design

The properties of the small motor in the simple model were examined. Graph 1 shows rotation rate against power in and graph 2 shows rotational kinetic energy against power in. The two graphs are linked via equation 5, hence the change in curve form between the two.



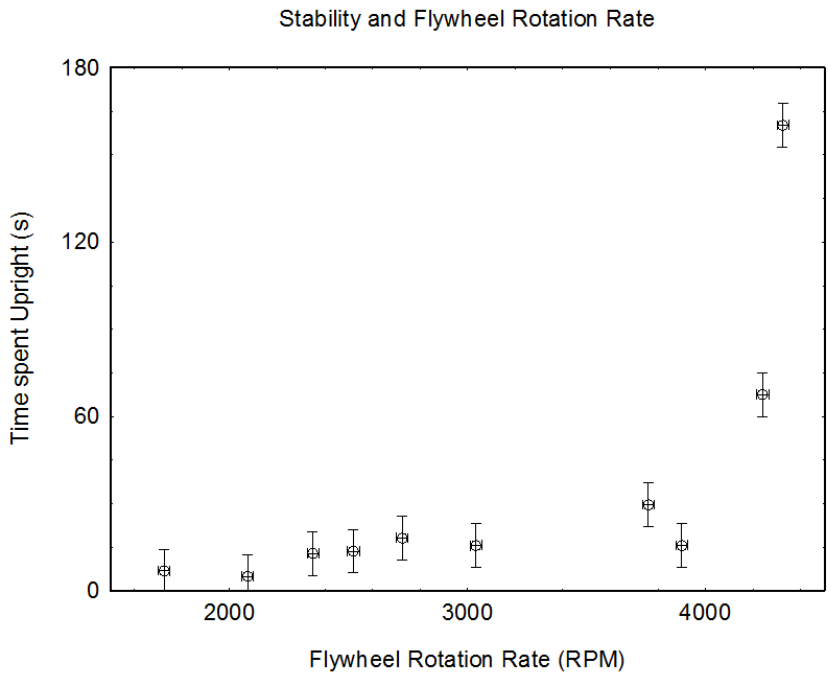
Graph 1: Rotation rate against power in for the small motor used in the lightweight model.



Graph 2: Rotational kinetic energy against power in for the small motor used in the lightweight model.

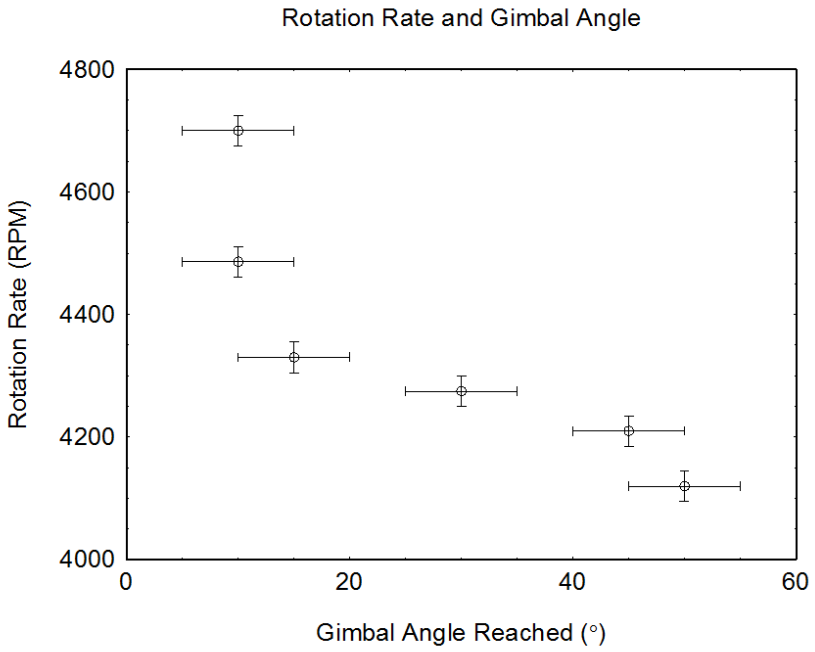
Note in graphs 1 and 2 the small error in flywheel rotation rate, this is because when the CD flywheel is held still the reading is easily taken and repeat readings yield values with a small range. From graph 1 it is apparent that the motor becomes less efficient at higher rotation rates and it can be seen maximum useable rotation rate is at around 5500RPM.

Now the rotation rate of the flywheel was varied for the lightweight model and graph 3 shows the point of stability to be (4300 ± 100)RPM.



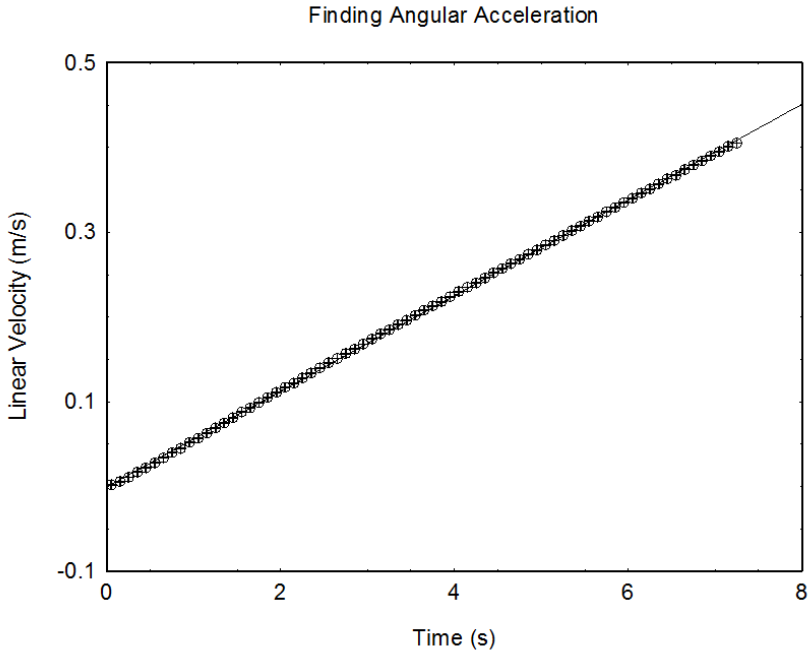
Graph 3: Time spent upright against flywheel rotation rate, notice the rapid rise in time spent upright at the point of stability.

Deviation from the smooth curve in graph 3 was due to the release not being identical for each run, however the trend is clearly visible. As expected from equation 10, graph 4 shows that as rotation rate increased so gimbal axis movement required for stabilisation decreased. The large error in gimbal angle was due to the use of a protractor, here this was the most significant error.



Graph 4: Rotation rate against gimbal angle reached for the lightweight model. As expected from equation 10 a higher flywheel rotation rate requires less movement of gimbal axis for stabilisation.

Now a greater understanding of the basic passive system was achieved so testing of the new flywheel begun. To find the new flywheel's physical moment of inertia from equation 13 so graph 5 was plotted and the gradient was taken to find angular acceleration.



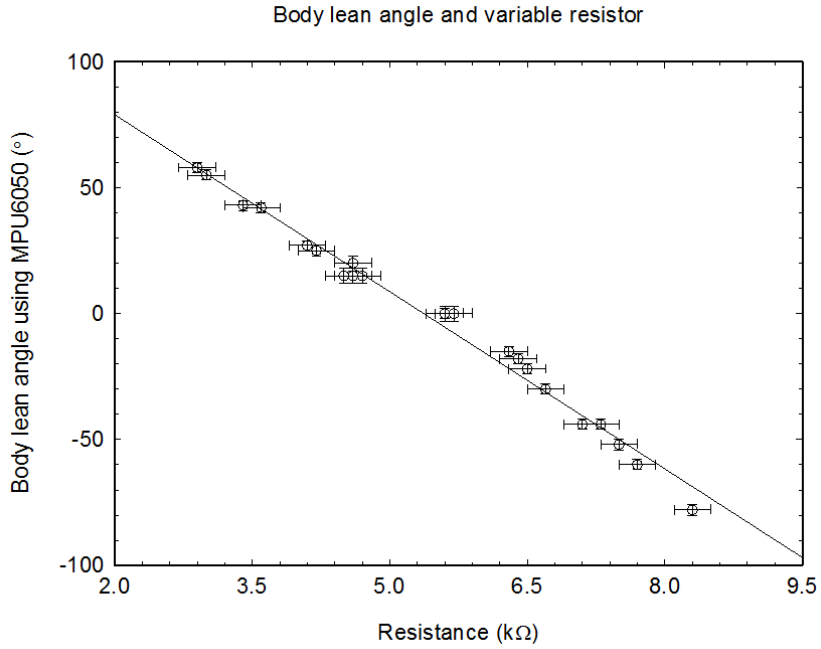
Graph 5: Linear velocity against time, the gradient is $0.0564 \mathrm{ms}^{-2}$ with negligible error.

Combining the gradient, angular acceleration and equation 13, the moment of inertia was experimentally found to be $(2870 \pm 90) \mathrm{gcm}^2$. This is different from the theoretical value of $(2515 \pm 13) \mathrm{gcm}^2$ due to the final design being milled not drilled.

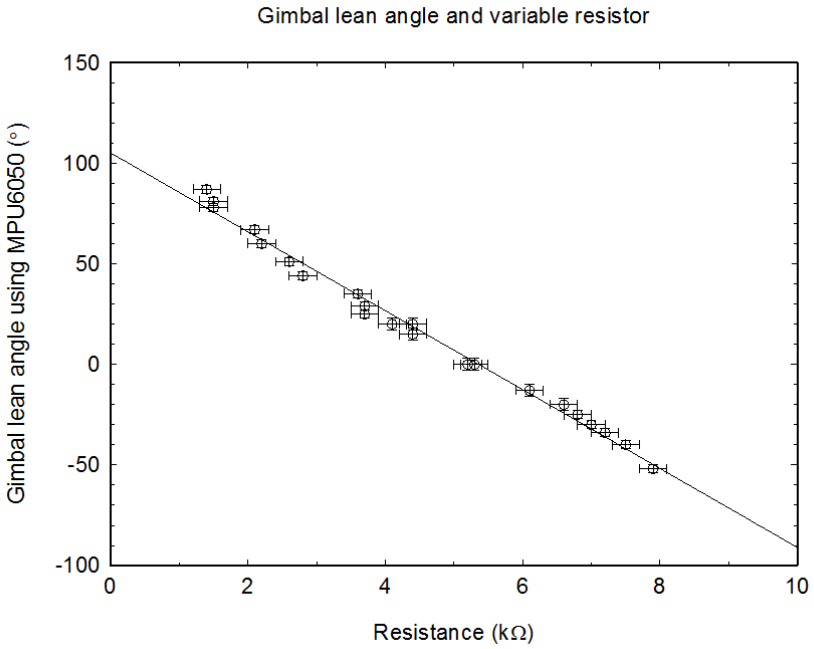
Figure 9 of section 3.2 shows the passive model at this stage prior to development, note the unsuccessful MPU6050 mounting.

4.2. Passive testing

Due to initial complications with the MPU6050 variable resistors were initially used to measure angle. Graphs 6 and 7 are the calibration of the body lean angle variable resistor and the gimbal lean angle variable resistor respectively. For body and gimbal axis the equations were $y = -23.5x + 126$ and $y = -19.6x + 105$ respectively.

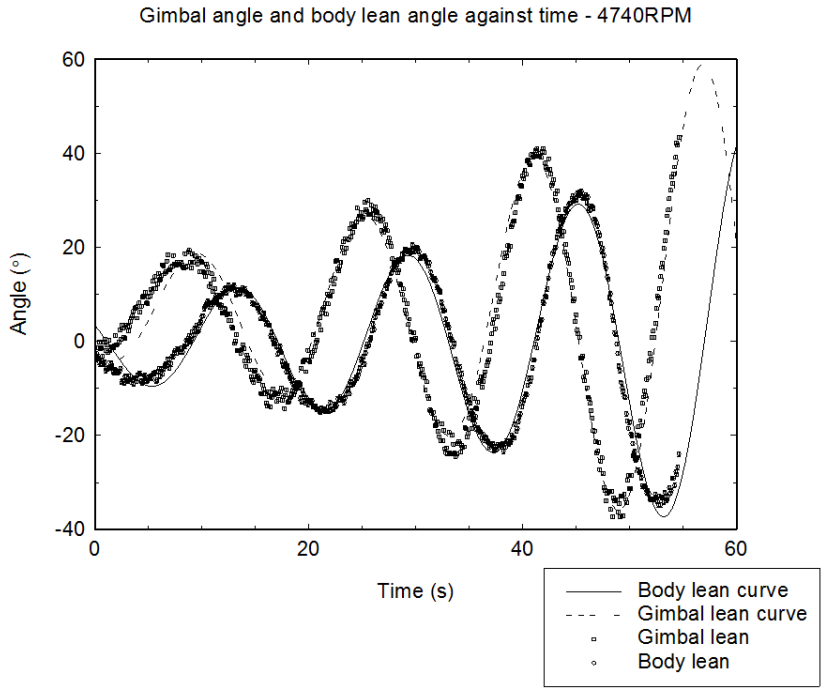


Graph 6: Body lean angle using MPU6050 against resistance of the variable resistor.

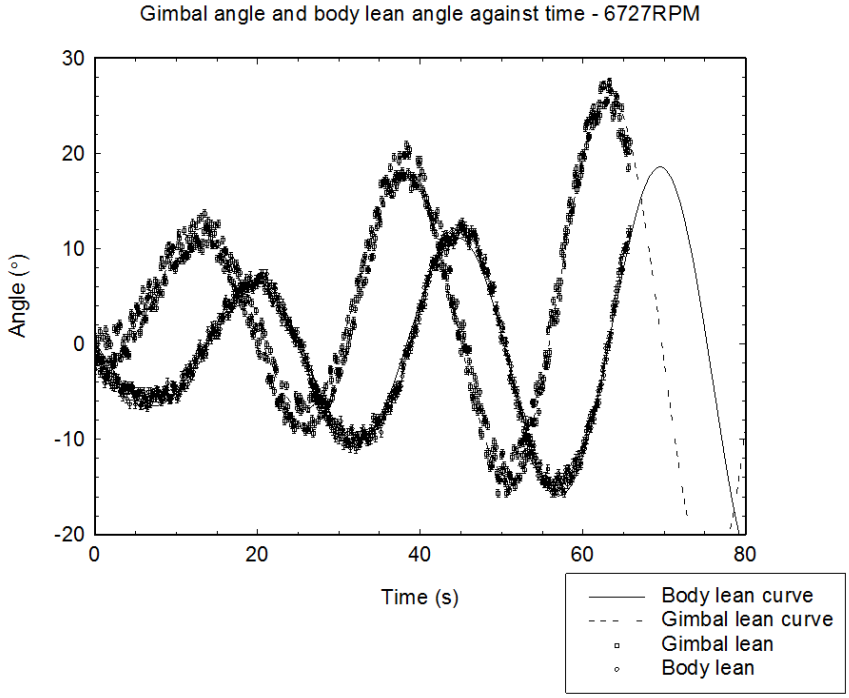


Graph 7: Gimbal lean angle using MPU6050 against resistance of the variable resistor.

Now graphs at six different rotation rates were plotted, the most significant being 4740RPM and 6727RPM as a plot is later made of gimbal against body lean angle. These two rotation rates were chosen as they have a significant rotation rate difference. In graphs 8 and 9 the general form seen in all angle against time data can be witnessed for flywheel rotation rates of (4740 ± 250)RPM and (6727 ± 250)RPM respectively. Note the phase difference between gimbal and body lean angles, this is the case at all rotation rates tested.

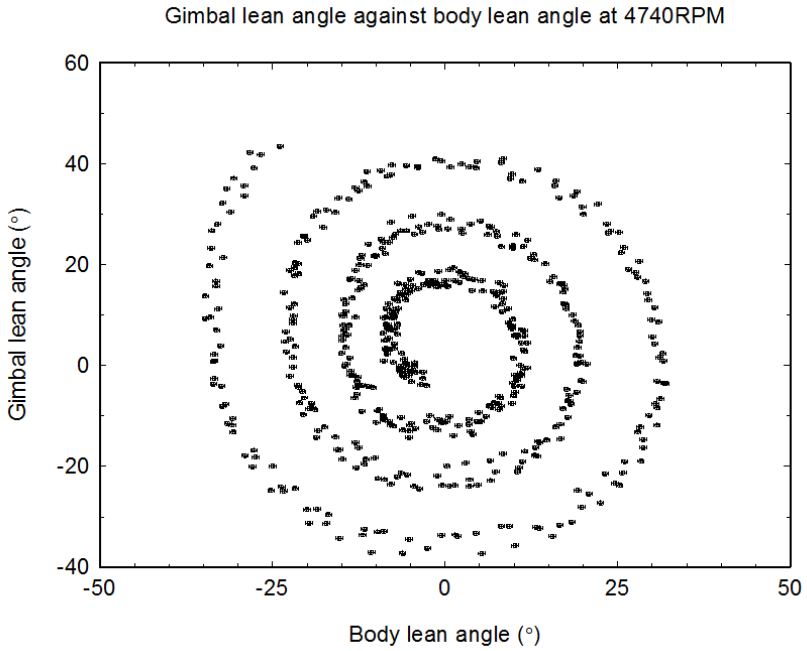


Graph 8: Angle against time for a flywheel rotation rate of (4740 ± 250)RPM.

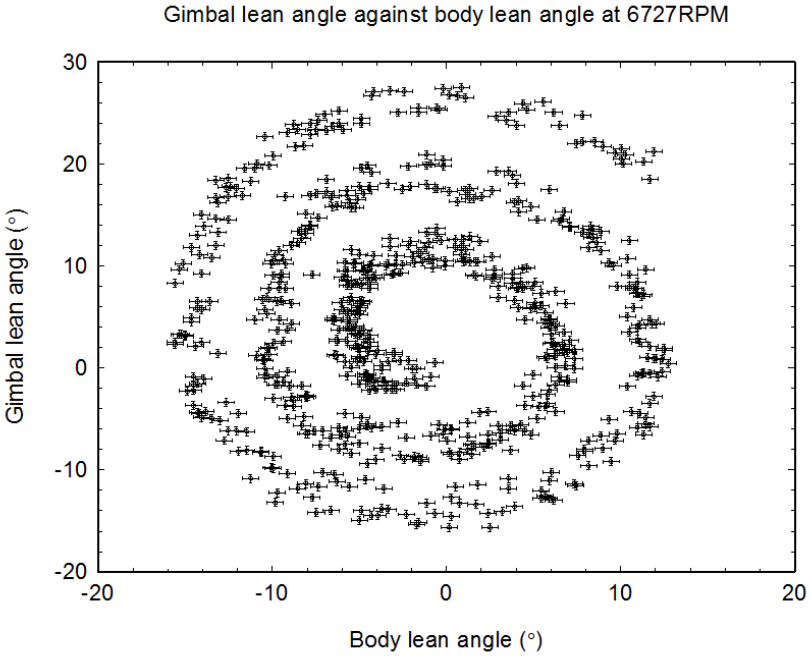


Graph 9: Angle against time for a flywheel rotation rate of (6727 ± 250)RPM.

For investigating the link between gimbal and body lean angle graphs 10 and 11 were plotted for the two rotation rates as in graphs 8 and 9 respectively, the phase difference is demonstrated to be $\frac{\pi}{2}$.



Graph 10: Gimbal lean angle against body lean angle for a rotation rate of (4740 ± 250)RPM, notice the density of points.

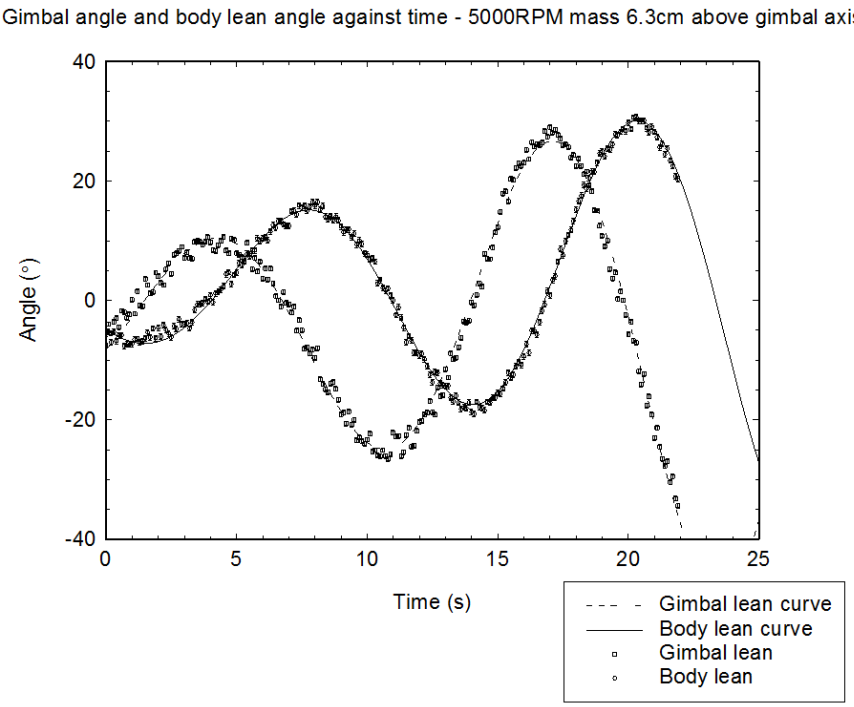


Graph 11: Gimbal lean angle against body lean angle for a rotation rate of (6727 ± 250)RPM, notice the increased density of points.

The increased density of points in graph 11 compared to graph 10 shows that the lean angles change at a greater rate at lower flywheel rotation rate, hence illustrating

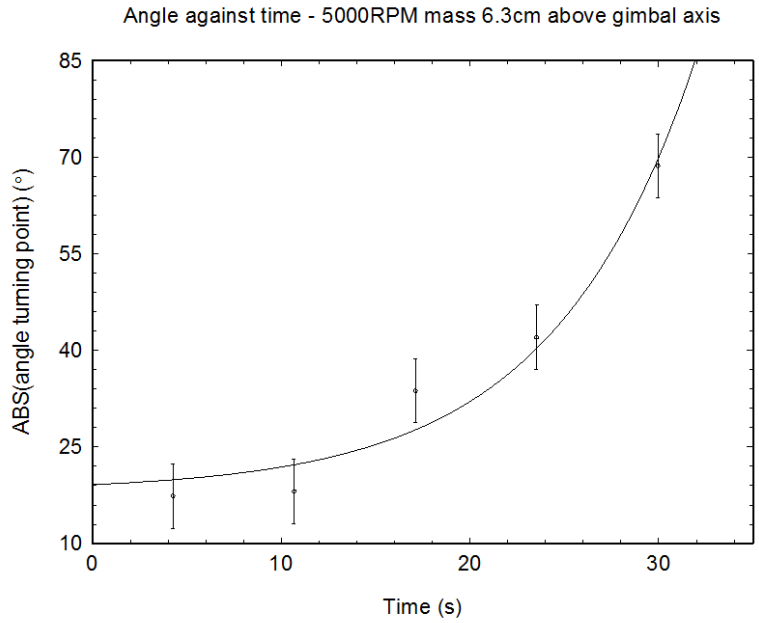
improved stability with higher rotation rate. This is the case as readings are taken every 0.2s in both instances and a greater number of points are present in graph 11 between any two given angles.

Ten graphs were taken varying gimbal added mass ((17.15 ± 0.01)g) height and rotation rate. Having found the hypothesised behaviour and with the gimbal centre of mass height optimise-able, graphs were taken with the rotation rate held at (5000 ± 250)RPM. The gimbal added mass heights of 3cm below axis, 1.9cm below axis, no mass (0cm equivalent), 1.9cm above, 2.8cm above, 3.6cm above, 4.5cm above, 5.3cm above, 5.6cm above and 6.3cm were taken and graphs of the form of graph 12 were plotted for each. Note the error in height measurements was 0.2cm, this was due to gauging the centre of the mass added to the gimbal.



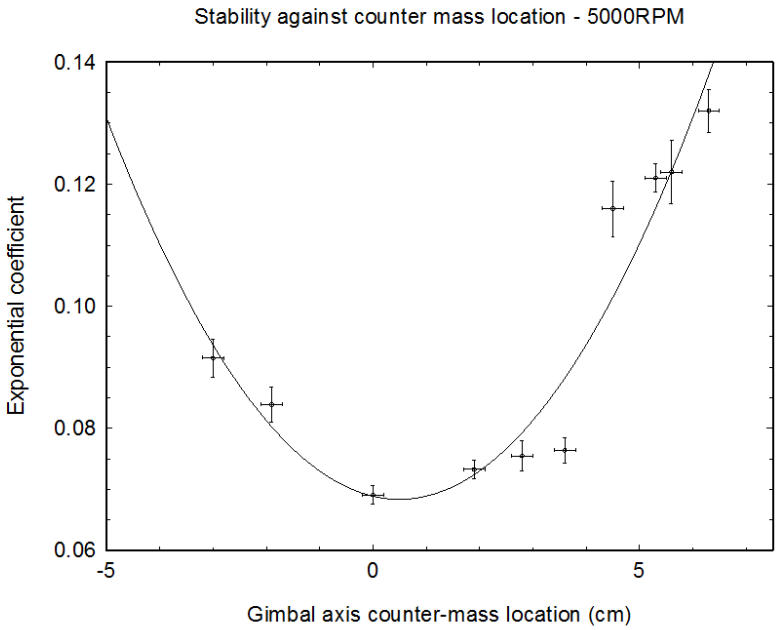
Graph 12: Angle against time for one of the gimbal added mass locations, here (6.3 ± 0.2)cm is shown. Which was the most top heavy configuration.

Then for every height the best fit curve for the body had its turning points found and this resulted in one graph with an exponential fit for each of the mass heights. The exponential fit for (6.3 ± 0.2)cm is shown as in graph 13.



Graph 13: The absolute value of the turning points against time with a best fit exponential fitted.

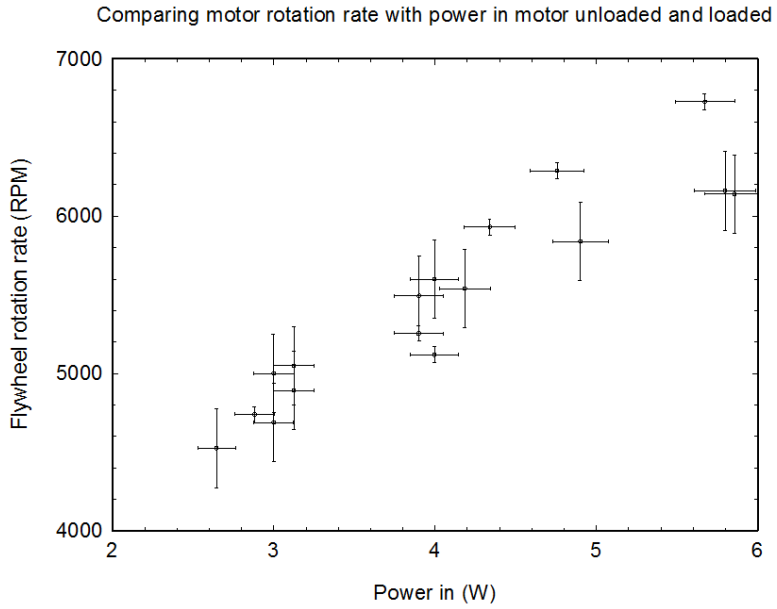
Graph 13 shows how f, the intercept correction value from equation 14 appears to not fit the data correctly, because a staggering can be seen in the data points. Here only five data points exist due to the short lived stability. Plotting all the graphs like 13 for the other values of gimbal mass height resulted in the following exponential coefficients, seen in graph 14.



Graph 14: Comparing added gimbal mass height and the exponential coefficients, giving rise to the optimum point for the added mass to be located.

The minima of the curve in graph 14 is the optimum location of the gimbal added mass, found to be (0.5 ± 0.1)cm above the gimbal axis. Note the best fit curve has equation $y = 0.00207x^2 - 0.00205x + 0.689$.

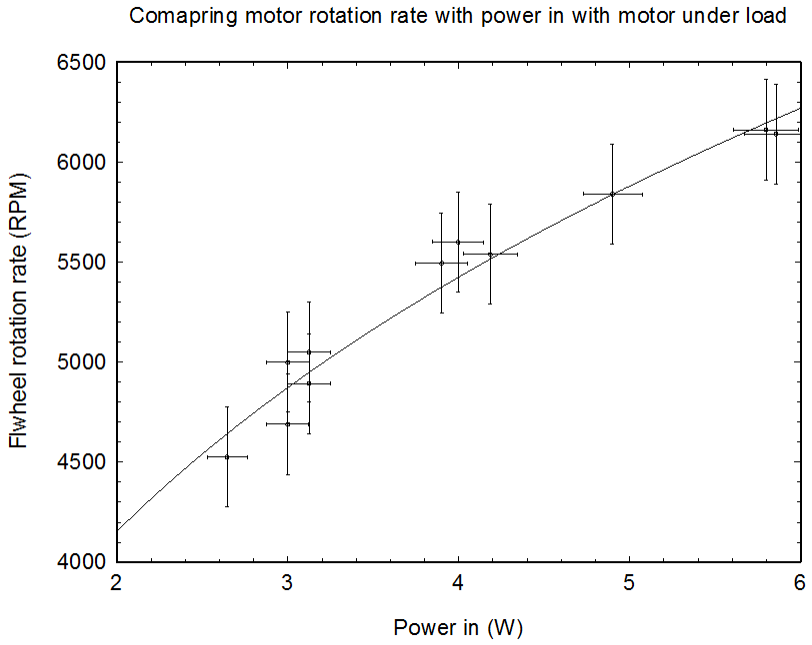
At this stage enough data with regards to the Permax motor was available to assess its suitability. Graph 15 shows all the Permax motor data available, both with motor under-load and motor unloaded.



Graph 15: Rotation rate of flywheel against power in, some data of system operating and some without operation.

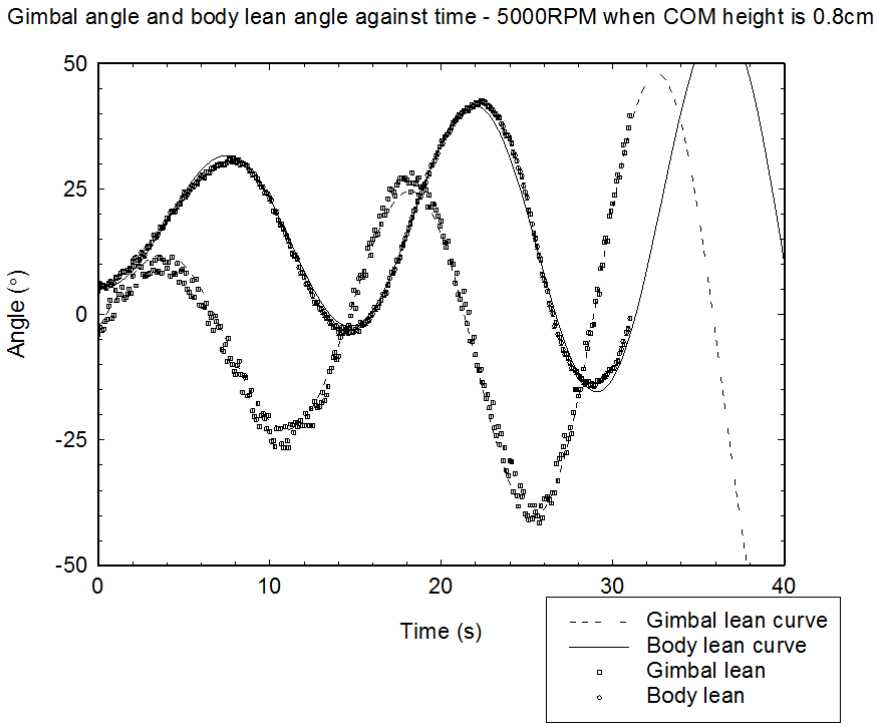
Graph 16 was plotted removing all data where the system was not in operation. Comparing graph 15 with graph 16 it is evident that when not under-load the motor responds differently and produces higher rotation rates for the same power in. This is likely due to energy losses in the form of heat due to friction between the motor shaft and bushes when under load. So for heavier systems a motor with bearings may be required.

Note how the drop in rate of increase in rotation rate in graph 16 is much less than that in graph 1 with increasing power in, so the Permax motor is concluded more suitable for this operating range.



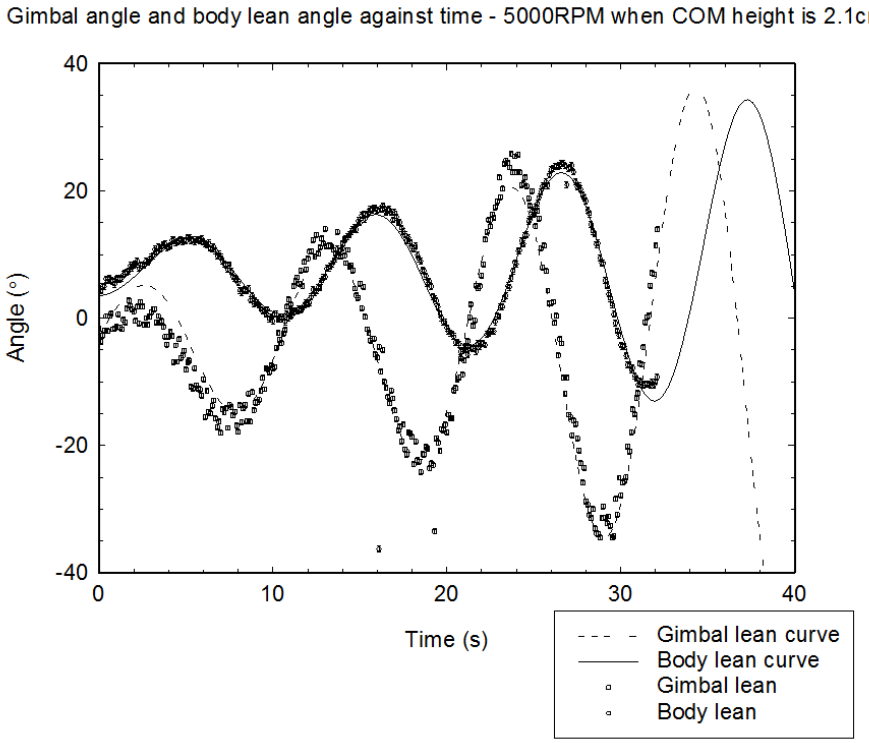
Graph 16: Rotation rate against power in when system is under-load and in operation.

Now the gimbal axis efficiency has been maximised so an investigation into the centre of mass of the body can be made, the following centre of mass heights were used: (0.8 ± 0.1)cm, (1.2 ± 0.1)cm, (1.8 ± 0.2)cm, (2.1 ± 0.2)cm and (2.7 ± 0.3)cm. Two of these shall be included here in full (0.8 ± 0.1)cm and (2.1 ± 0.2)cm, graphs 17 and 18 show the angle against time graphs for these two centre of mass heights respectively.



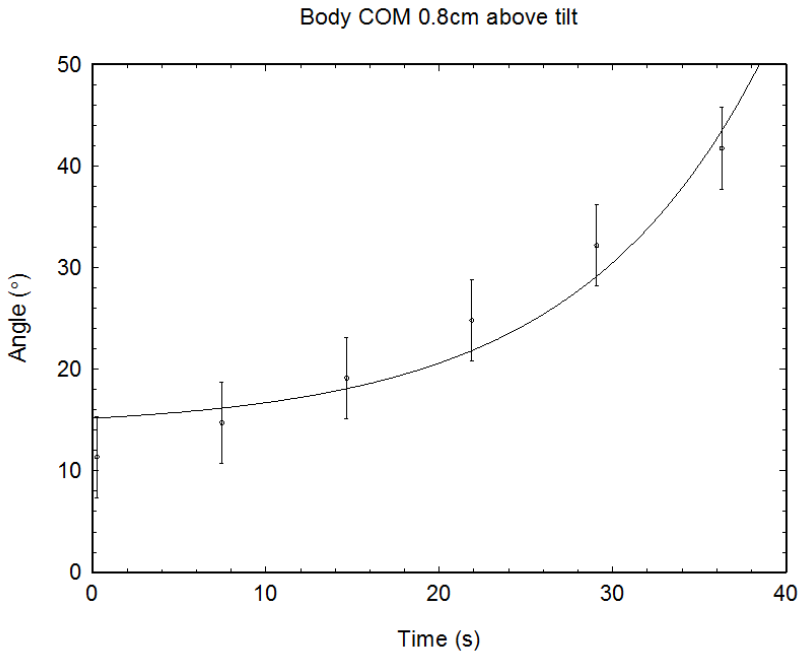
Graph 17: Angle against time for body centre of mass height at (0.8 ± 0.1)cm.

In graph 18 anomalous points were seen for the first time and they are believed to have occurred from either malfunctioning variable resistors or a less likely Cassy lab malfunction.

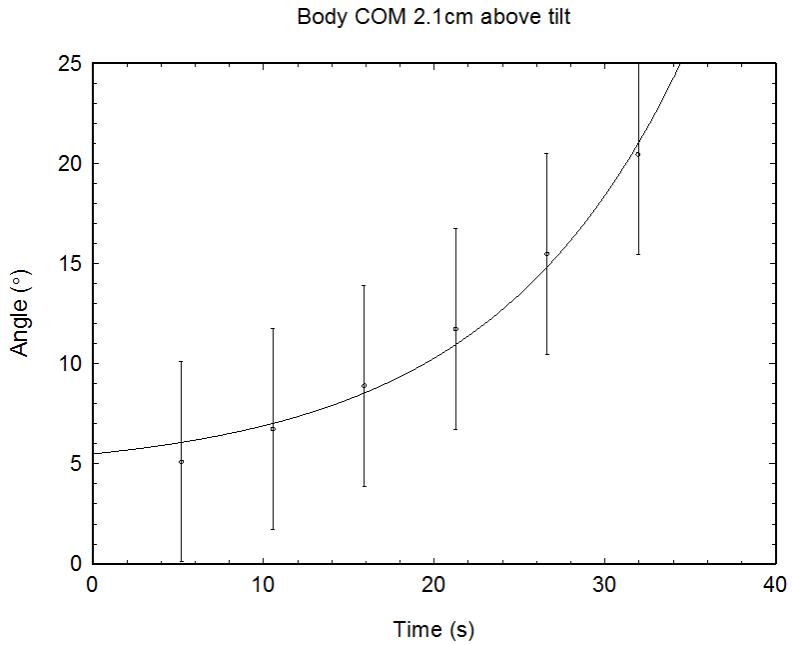


Graph 18: Angle against time for body centre of mass height at (2.1 ± 0.2)cm.

Now for all centre of mass heights the exponentials were plotted, graphs for (0.8 ± 0.1)cm and (2.1 ± 0.2)cm are included in the form of graphs 19 and 20.

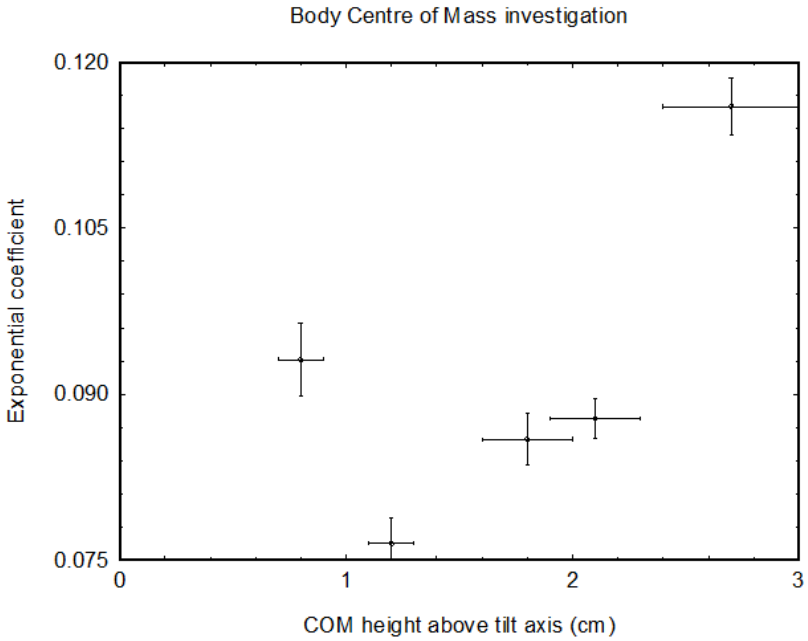


Graph 19: Angle against time for the turning points from graph 17 for body centre of mass height at (0.8 ± 0.1)cm. Note the particularly ill-fitting exponential curve.



Graph 20: Angle against time for the turning points from graph 18 for body centre of mass height at (2.1 ± 0.2)cm. Note the better exponential fit than that in graph 19.

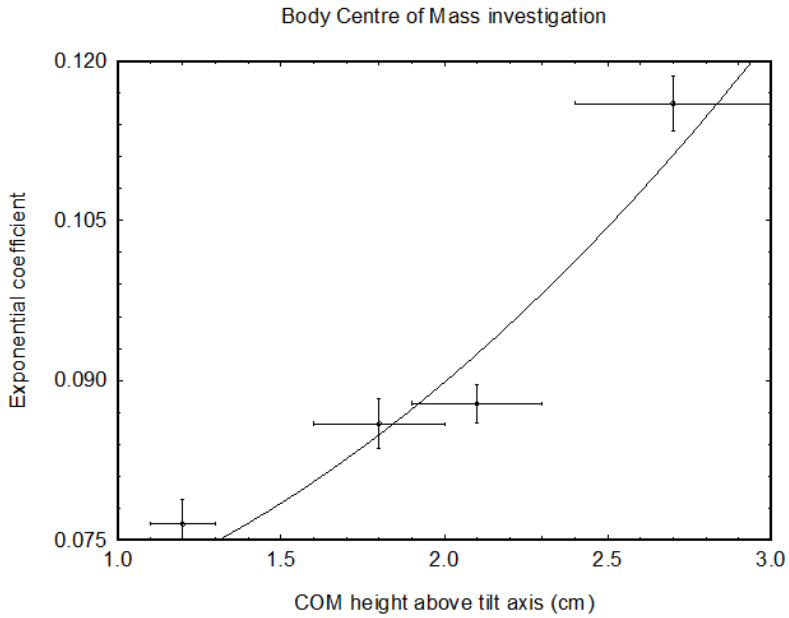
Plotting these exponential coefficients against body centre of mass height leads to graph 21. As seen in graph 19 in the form of an ill-fitting curve the point at (0.8 ± 0.1)cm doesn't fit the trend.



Graph 21: Exponential coefficients against centre of mass height, note the ill-fitting point at (0.8 ± 0.1)cm.

Removing the ill-fitting point results in graph 22 and fitting a curve to this data yields an equation of $y = 0.00649x^2 + 0.0639$. As expected the higher the centre of mass

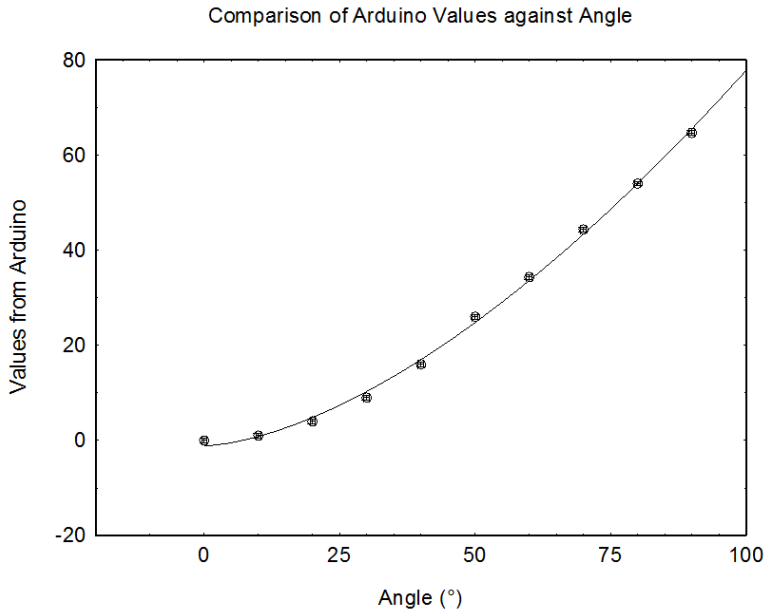
so the greater the instability. The intercept should be negligible because if h_v is zero then there would be no toppling torque.



Graph 22: Exponential coefficient against centre of mass height with curve fitted.

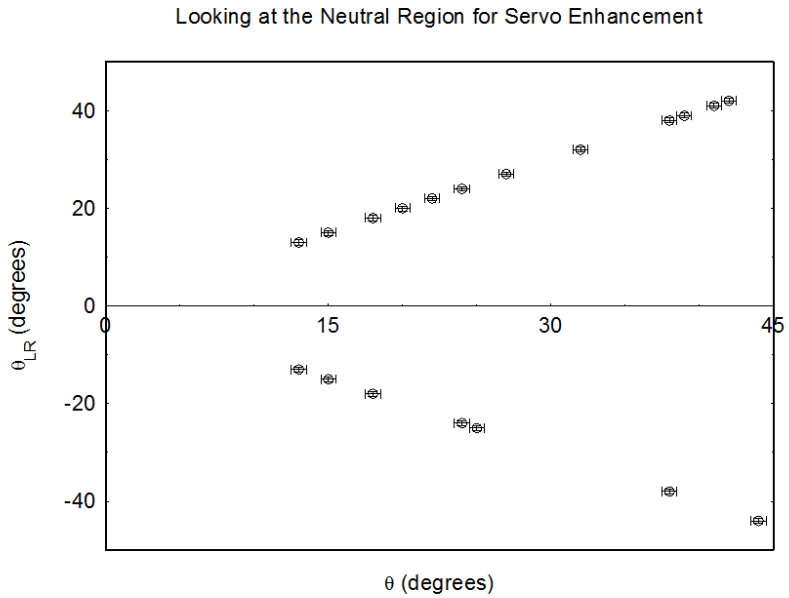
4.3. Arduino Uno and servo control

Initially the Arduino values received from the MPU6050 were not useable values and these had to be fitted to a function as seen in graph 23. The line of best fit has equation $y = 0.0482x^{1.61} - 1.09$, this was added to the code so that the MPU6050 output was converted to an angle in degrees.



Graph 23: Arduino values against angle in degrees, used to create a useful output from the Arduino.

Note in graph 23 that all angles are positive. The code was now altered as described earlier to read left and right, this successful method is demonstrated in graph 24.



Graph 24: Successfully showing the chips ability to decide whether it is leaning to the left or the right based on the zero region.

Note however the neutral region does not print any values, this is due to the large error in the chip around the zero point and the method of left and right differentiation. The chipset is most accurate at angles whose modulus is 88°.

4.4. Active system development

Whilst graphical data was not obtained for this section the active system was successful and the model remained stable. Using the final dampening method involving rubber belts and exterior mounting the MPU6050 successfully read angles as in graph 24 and this was inputted into the Sanwa servo.

The most effective function was found to be closely related to equation 10. Whist oscillatory motion was seen its amplitude was minimised, limited in stability by the accuracy of the MPU6050 at small angles and its sensitivity to rapid angle change.

The final system can be seen earlier in figure 11 in section 3.4 as successfully run. Note the features mentioned earlier, Permax motor, aluminium flywheel, suspended MPU6050 connected via polymer tape, fully adjustable centre of mass and Sanwa servo. A video of this system can be viewed on youtube by following the link [21], note the oscillatory motion due to MPU6050 sensitivity issues as described earlier.

By comparison with figure 9 the large amount of development of the model can be witnessed, not only is it more effective but the mass to be stabilised is over four times the initial predication of 100g, this is evidence of the systems effectiveness. For a complete demonstration the system was placed in a (2.46 ± 0.01)kg boat and stability was witnessed. Figures 12 and 13 show the system mounted in the boat and figures 14 and 15 show the system in operation with covers.

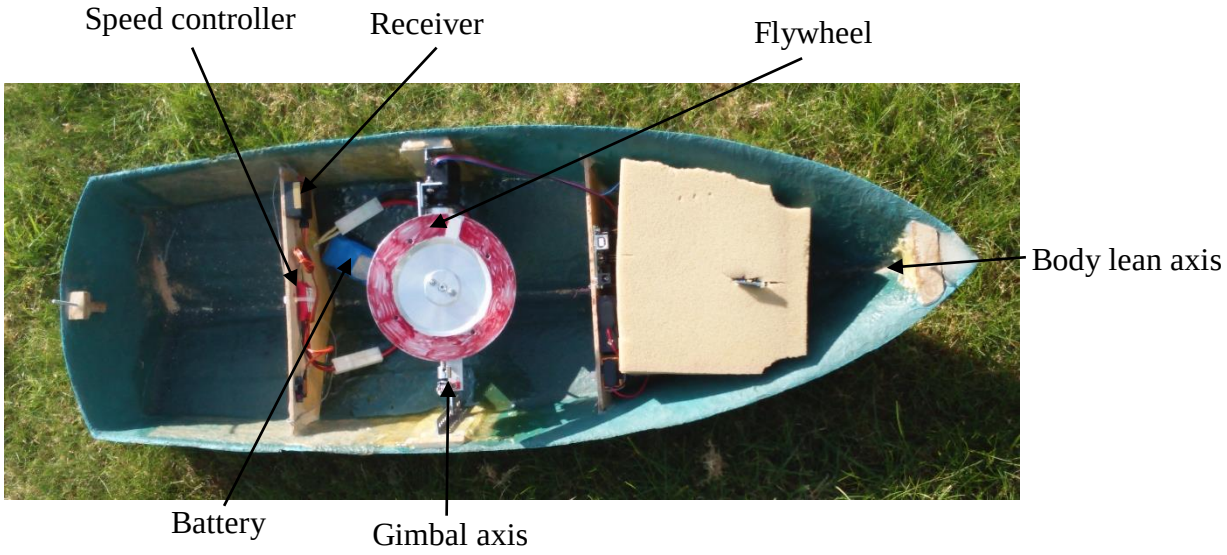


Figure 12: The system mounted in the boat, note the speed controller and receiver location as well as the battery.

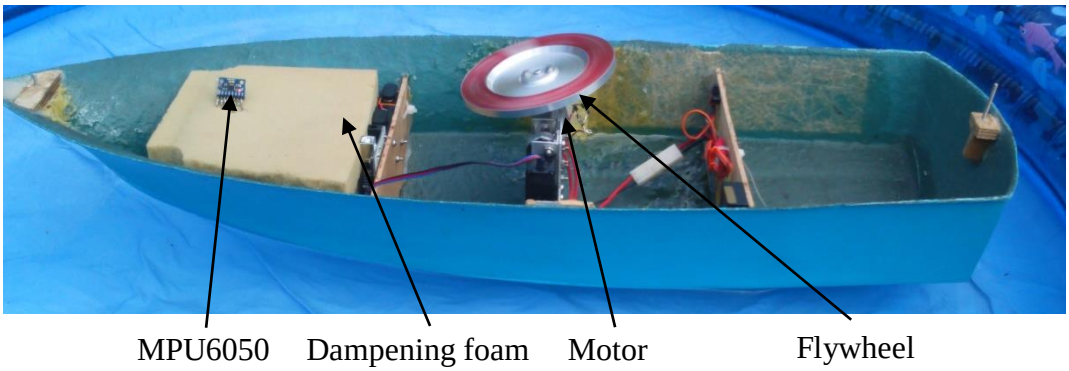


Figure 13: The system mounted in the boat in action, note the dampening foam around the MPU6050.

Figure 14: The complete system with covers as stabilised, note the remote control for variable flywheel rotation rate and also note the controlled environment in which the system was tested.

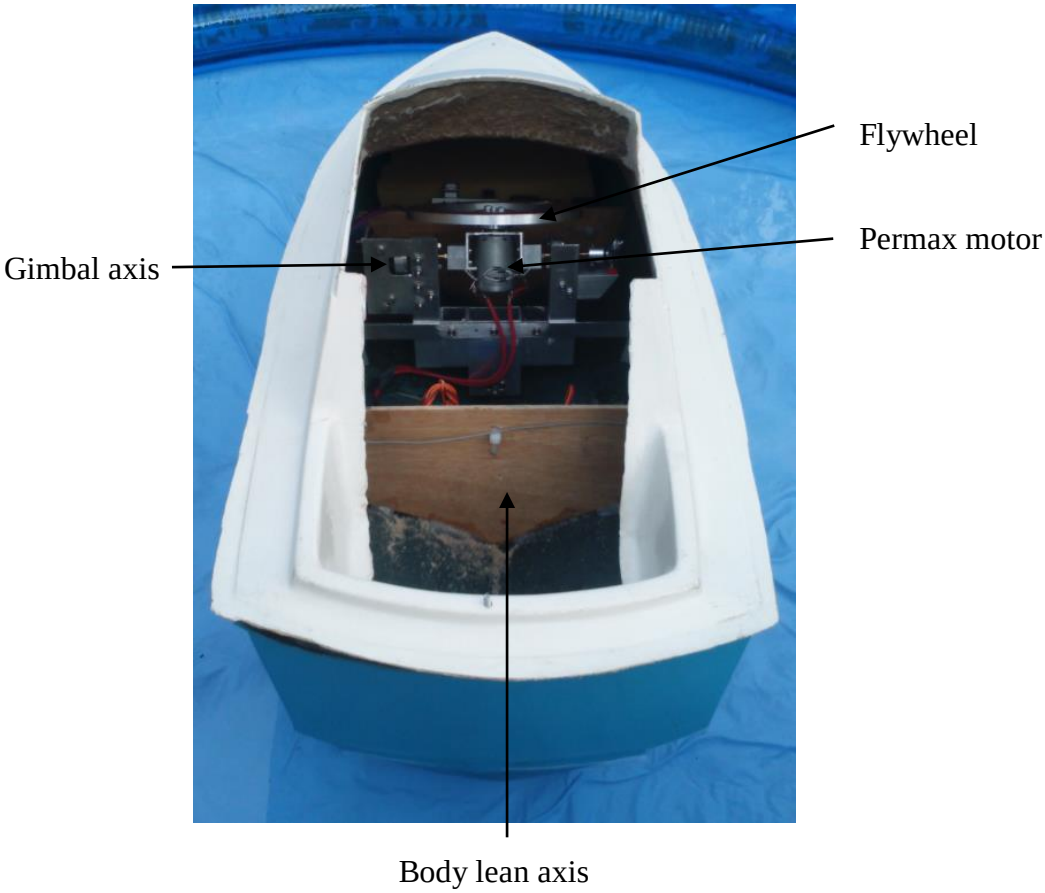


Figure 15: The complete setup with boat cover as tested with success, the system still corrects tilt despite the raising of the centre of mass.

The isolation padding around the MPU6050 was effective at lower rotation rates, which were ample due to the higher $\dot{\varphi}$. The boats oscillatory rolling motion when nudged was not instantly damped but took a complete oscillation or so before stopping, this is believed to be due to MPU6050 sensitivity. This behaviour can be witnessed in the video file which can be viewed at [21]. When the gimbal axis rotated rapidly it induced a vibration which affected the MPU6050. Vibration was limited using lower flywheel rotation rates.

5. Discussion

5.1. Initial testing and component design

Graphs 1 and 16 show the operational differences of the small and Permax motor respectively. Whilst for the lightweight model the small motors operating range was suitable, the Permax motor was required for the larger model with higher load application. The power in error resulted from the scale increments on the digital power supply. The rotation rate error was due to fluctuations when using the digital tachometer. Whilst the small motor data was taken unloaded it emphasises the difference between the two motors. Graph 2 simply links graph 1 and equation 5, so the shape of the curve is different from that in graph 1. From graph 1 it is evident the maximum operating range is around 5500RPM.

Graph 3 shows the point of stability at (4300 ± 100)RPM. The error in time is from human reaction time of pressing the stopwatch and any model release discrepancies.

The variation in point location from a smooth curve is likely to also be due to the model release method, some runs may be released more stable than others. This was improved in the more accurate passive testing by mechanical release using a mains earth cable as a holder and then simple removal ensures a more reliable, constant release. Graph 4 supports equation 10 in that a higher flywheel rotation rate requires reduced gimbal angle movement rate for the same stabilising torque. The large error on the gimbal angle measurement is due to using a protractor to measure the angle.

Graph 5 and equation 13 combine to find the moment of inertia of the alloy flywheel. The main error sources in the moment of inertia were due to resistance in the angular rotation measuring device and the value of r varying as the string unwinds. The experimental moment of inertia value was found to be $(2.87 \pm 0.09) \times 10^{-4}\text{kgm}^2$ which is close to the theoretical, original design value of $(2.515 \pm 0.013) \times 10^{-4}\text{kgm}^2$. The flywheel was hence concluded suitable for the application it was designed for.

5.2. Passive testing

Variable resistors were deemed more accurate initially and graphs 6 and 7 allowed calibration of the variable resistors against angle for body and gimbal lean. Error in MPU6050 reading was low due to the flywheel being stationary during calibration. Error in resistance was obtained from the fluctuations during measurement by the Cassy system.

The angle measurement error in graphs such as 8, 9, 12, 17, 18 and others as mentioned but not included here originated from both the variable resistors and also the error from the calibration experiment, the former being the largest error. (4740 ± 250)RPM and (6727 ± 250)RPM were used so that in graphs 10 and 11 the phase difference between gimbal and body lean can be witnessed clearly as $\frac{\pi}{2}$. It was also found that gimbal angle and body lean angle rate of change was reduced with higher flywheel rotation rate, seen by noting data point density.

From the 10 graphs varying rotation rate and gimbal axis height it was found that the gimbal axis height was related to stability, hence a thorough investigation was initiated using (5000 ± 250)RPM. Various mass heights led to graphs such as 12, for each of these graphs an exponential graph was plotted as in graph 13. These were then combined to form graph 14, where the optimum location was found to be (0.5 ± 0.1)cm for the gimbal mass of (17.15 ± 0.01)g. This result is reliable, whilst graphs such as 13 can be scrutinised due to the apparent pairing effect when the negative angles are modulated and shifted by the f value from equation 14. The mass was not placed at the optimum position as this was not possible, however the centre of mass at this optimum position was obtained using different masses. This did not induce any significant system change compared to using the one single mass located optimally.

Graph 15 supports the earlier conclusion made in section 5.1 that when motors are unloaded they operate at higher rotation rate for the same power in, as expected. So this validates the comparison between graph 16, the Permax motor and graph 1 the small model motor.

Flywheel rotation rate of (5000 ± 250)RPM was used to investigate the effect of body centre of mass on system stability. Graph 19 was scrutinised after graph 21 pointed to ill-fitment. The graph plotting package was unable to provide an improved fit on the data set in graph 19 and so the point was discarded and graph 22 was plotted. As expected from equation 10 a larger value of h_v leads to greater instability.

5.3. Arduino Uno and servo control

Error in MPU6050 readings are minimal even after conversion via the fit of graph 23 providing vibrations are kept to a minimum. Due to this reliability time was invested to create a left and right distinction function, as shown in graph 24, where the small error in the MPU6050 can be witnessed.

Note in Graph 24 that in the region below the modulus of 13° there are no data points, this is due to the chip being unable to register small angles. This would certainly be a point for improvement if a system refinement was to be made, this later led to jittery movements in the active system. Jittery movements were reduced using noise minimisation in the code. For system safeguarding purposes should extreme gimbal values of $< 30°$ or $>150°$ be reached the code is limited and further angles cannot be inputted until back inside the range, a warning is also shown in the serial monitor. Throughout testing faults were solved and during this print functions were used to aid fault location, this at times led to lengthy testing but had a high success rate.

5.4. Active system development

The final physical dampening solution to the sensitivity problem with the MPU6050 was successful, largely isolating the chip from flywheel vibrations. The damping was further improved when combined with the coded dampener as the MPU6050 reading becomes reliable under a variety of rotation rate conditions. Before this coded dampener was used servo movements were jittery and caused instability. To improve the setup further an angle measurement chip less susceptible to vibration would be used. This improvement would lead to the ability to test using higher flywheel rotation rates and so the system would be effective in heavier applications, such as larger marine vessels.

By controlling $\dot{\varphi}$ using a servo stability with a heavier model than originally expected was witnessed, but at lower flywheel rotation rates. The successful active system was placed in a (2.46 ± 0.01)kg fibreglass hulled boat to test in a real world environment. The system was effective, however at least a single oscillation occurred each time, this was believed to be due to MPU6050 sensitivity. The MPU6050 was also affected by the gimbal axis rapid rotation in response to the initial disturbance. Townsend et al [14] discuss the development of a similar system, however this system differs as its development involved consideration of satellite control (lightweight flywheel) and land based control, hence despite its lightweight construction counter-torque available is considerable.

6. Conclusions

The behaviour and characteristics described in previous works by Spry and Girard [6], and Brennan [7] were tested and passive behaviour was witnessed. Whilst stability was not reached the systems optimum gimbal axis centre of mass point was found.

The design of potential flywheels and their materials were scrutinised and a suitable design made. A higher centre of mass system was confirmed more unstable and greater gimbal axis motion was witnessed.

In [7] Brennan designed a manual active stability system, whereas here the system constructed is automated. The Arduino UNO was programmed to produce an angle from an MPU6050 input, then from passive testing the equation was found to link servo output to body lean angle. The system was tested in a lab environment successfully and so it was then fitted into a (2.46 ± 0.01)kg boat and shown to reduce the impact of waves. The response could be improved in future but would require an MPU6050 replacement, a chip with higher sensitivity to angle change and less sensitivity to high frequency vibration. Changing the chip would also lead to re-optimising the code for the new setup. The system was optimised according to the behaviour witnessed during testing and the particular conditions of use. This optimisation initially involved testing basic models, but the system is now applicable in marine, land and with a little modification to the code, satellite application also.

Acknowledgements

Thanks to the laboratory technicians for helping to locate some standard equipment when required and a special thanks to Dr Phil Lightfoot for machining the components and for any advice given during the project.

References

[1] X. Roser and M. Sghedoni November 1996 Control moment gyroscopes (CMG's) and their application in future scientific missions (Spacecraft Guidance, Navigation and Control Systems, Proceedings of the 3rd ESA international conference held 26th-29th November 1996) p52

[2] Julien Gagne, Edouard Laroche, Olivier Piccin and Jacques Gangloff An active cardiac stabiliser based on gyroscopic effect September 2009 (Engineering in Medicine and Biological Society, EMBC 2009, Annual international conference of the IEEE) p6769-6772

[3] S.Schaffer Artificial Horizon accessed 7th September 2014 History and Philosophy of Science Cambridge Digital Library

[4] Edwyn Gray 19th Century Tornedos and their Inventors (Naval Institute Press, Annapolis, MD21402(2004)) p155-156

[5] N.C Townsend, A.J. Murphy and R.A. Shenoi 11th June 2006 A new active gyrostabiliser system for ride control of marine vehicles (Ocean Engineering 34 (2007)) p1607-161

[6] Stephen C. Spry and Anouck R. Girard 28th January 2009 Gyroscopic stabilisation of unstable vehicles: configurations, dynamics, and control (Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility 46:S1) p247-260

[7] Louis Brennan 8th August 1905 Means for imparting stability to unstable bodies United States Patent, Patent Number 796893

[8] B.V.Elsevier 18th November 1909 The Brennan Mono-rail System (Nature 82) p79

[9] Henry T. Eddy The Mechanical Principles of Brennan's Mono-rail Car June 1910 (Journal of the Franklin Institute, Volume 169, Issue 6) p467

[10] H. Ormiston and A. Emerston Informal Discussion. Railway Track for High Speed and Heavy Traffic 1st August 1973 (ICE Proceedings Volume 54, Issue 3) p563-565
[11] Marshall Schuon 2nd December 1990 About Cars: Detroit's Dreams and what they mean (The New York Times Section 8) p12
[12] Kim et al. 10th September 2013 Electronic Control System for Gyroscopic Stabilized Vehicle United States Patent, patent Number 8,532,915 B2
[13] S.Kidane, L.Alexander, R.Rajamani, P.Starr and M.Donath 7th March 2008 A fundamental investigation of tilt control systems for narrow commuter vehicles (International Journal of Vehicle Mechanics and Mobility, Volume 46, Issue 4)
[14] N.C Townsend, A.J. Murphy and R.A. Shenoi 11th June 2006 A new active gyrostabiliser system for ride control of marine vehicles (Ocean Engineering 34 (2007)) p1607-1617
[15] John D. Adams and Shepard W. McKenney 13th December 2005 Gyroscopic roll stabiliser for boats United States Patent, Patent Number 6973847 B2
[16] Harun Yetkin and Umit Ozguner Stabilizing Control of an Autonomous Bicycle June 2013 (Control Conference (ASCC) 2013 9th Asian, IEEE) p1-6
[17] Harun Yetkin, Simon Kalouche, Michael Vernier, Gregory Colvin, Keith Redmill and Umit Ozguner Gyroscopic Stabilization of an Unmanned Bicycle 6th June 2014 (American Control Conference 2014) p4549-4554
[18] Robin S. Sharp Stability, Control and Steering Responses of Motorcycles 9th August 2010 (Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility, 35:4-5) p291-318
[19] A.L.Schwab, J.P.Meijaard and J.D.G.Kooijman Lateral dynamics of a bicycle with a passive rider model: stability and controllability 20th September 2011 (International Journal of Vehicle Mechanics and Mobility, Volume 50, Issue 8) p1209-1224
[20] John Chi Short example sketch 17th August 2014 (playground.Arduino Public domain, Arduino)
[21] https://www.youtube.com/watch?v=o1rFL5E_cuE&feature=youtu.be

Appendix

The complete code is included below for reference as used in the successful active system.

```
#include<Wire.h>                                          //use the wire and servo libraries
#include<Servo.h>

Servo servo;                                              //Creating the object for the code, call it
servo
int led = 11;
int fade = 2, level, brightness;

const int MPU=0x68;                                       //The location of the MPU

int16_t MPUGyroX;
                                                          //Essentially denote the following as 16 bit
integer
                                                          //The setup function shall define initial
starting values
void setup() {                                            //Void setup, when the rest button is pressed
the program is to be restarted, for fault finding
                                                          //Now the code begins begins
  Wire.begin(MPU);                                        //Required to start I2C communication, uses
twi.init function
  Wire.beginTransmission(MPU);                            //Start communication
  Wire.write(0);                                          //Arduino turned on for first time
  Wire.write(0x6B);
  pinMode(led,OUTPUT);
  servo.attach(9);                                        //tell the arduino the location of the servo
  Serial.begin(9600);
}                                                         //Transmit at 9600 bits per second (baud)
```

```
int zerocorrection;                                                    //Define some variables to be used later

int x = 1, MPUGyroXP, zerocorrectionX, Read1, Read2, Read3, Read4;   //define further variables
int count = 1, count1, count2, wait = 200, servoin, ticker, servoinput, correction, servoangle = 90, twist2,
twist1, sinthetadot;
int counter = 0, count3, count4, twist;
float thetadot, sintdot, torque, omega, extra, gimconstant, constant, correction1 = 90;

void loop() {                                                          //The loop function for updating data
  Wire.beginTransmission(MPU);                                         //Restart communication
  delay(wait);                                                         //Pause code for wait time (in ms)
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU, 1, true);
  MPUGyroX = Wire.read();


  MPUGyroXP = 65 - MPUGyroX;


  if (x==4){                                                           //If statement used to find correction factor
    Read4 = MPUGyroXP;
    zerocorrection=(Read1+Read2+Read3+Read4)/4;                        //Average the first four values
    zerocorrectionX = pow((zerocorrection + 1.09)/0.0482, (1/1.61));
    Serial.print ("Zero Correction Found:    ");                      //Inform user zero correction known
    Serial.println (zerocorrection);

    x=x+1;
  }
  if (x==3){
    Read3 = MPUGyroXP;
    x=x+1;
  }

  if (x==2){
    Read2 = MPUGyroXP;
    x=x+1;
  }
  if (x==1){
    Read1 = MPUGyroXP;
    x=x+1;
  }
  if (x > 4){
    Wire.beginTransmission(MPU);                                       //Start communication of device and arduino
    Wire.write(0x3B);
    Wire.endTransmission(false);
    Wire.requestFrom(MPU, 14, true);                                   //Begin process (true)

    MPUGyroX = Wire.read();


    MPUGyroXP = pow((65-MPUGyroX + 1.09)/0.0482, (1/1.61));            //Using function found, find angle in degrees.

    if (MPUGyroXP - zerocorrection < 0){
      MPUGyroXP = 0;

    }                                                                  //If the tilt angle is less than when it was
started no output required

    if (MPUGyroXP > 10){                                               //If tilt angle>10 degrees correction required
      level = MPUGyroXP - 10;
      brightness = level*2;                                            //Brightness as a function of tilt
      analogWrite(led, brightness);
      if (brightness > 253){
        Serial.print ("***** ERROR IN LED OUTPUT *****");
      }    //Fault detection for error in ouput calculation.
    }

    count = count + 1;                                                 //Which run through of the code is occuring

    if (count % 2 == 0){
      count2 = MPUGyroXP;
    }
    else{
      count1 = MPUGyroXP;                                             //Count through code is an even or odd count
    }

    if (abs(MPUGyroXP) >= 12){

      if(ticker % 2 == 0){
        twist = (-1)*abs(MPUGyroXP);
        counter = 0;
      }

      else{
        twist = abs(MPUGyroXP);
        counter = 0;
      }
    }
    servoin = (twist);

    if (abs(MPUGyroXP) < 12){                                         //This allows the system to recognise when it
has passed through zero and hence when the values should use the reverse polarity function contained
      if (counter == 0){
        ticker = ticker +1;
        count3 = count1;
        count4 = count2;
        count1 = count4;
        count2 = count3;                                              //Swap count values, a storage idea is used
        counter = 0;
        counter = counter + 1;
```

```
      }
    }


    Serial.print("    twist=");
    Serial.println(twist);

    if (count % 2 == 0){
      twist2 = twist;
    }
    else{
      twist1 = twist;
    }
    if ((abs(twist2) - abs(twist1)) > 15){                          //Remove noise via programming
      if (count % 2 == 0){
        twist2 = twist1;
      }

      else{
        twist1 = twist2;
      }
    }
    Serial.print("*******NOISE MEASUREMENT:  ");
    Serial.print(abs(twist2)-abs(twist1));
    thetadot = (twist1 + twist2)/(2);                               //Rate of theta change
    Serial.print(" theta dot (average value)");
    Serial.print(thetadot);                                         //Useful for later development

    constant = 0.5;                                                 //Roughly mgh changable for different systems
    torque = constant * sin(twist*(3.1415926535/180));              //Torque to be countered
    Serial.print(" torque");
    Serial.print(torque);
    gimconstant = 0.02;                                             //Gimbal Iw. roughly 0.1 - 1
    extra = 0.0002;                                                 //Power variable
    omega = ((torque) + extra)/(gimconstant);                       //Extra - for more torque power
    Serial.print("omega =");
    Serial.print(omega);

    correction = omega*(wait/100) + correction1;

    if (correction < 30){
      correction = 35;
      Serial.print("Warning minimum lock 30 degrees");
    }
    if (correction > 150){
      correction = 145;                                            //To remove lock to lock, safety component
      Serial.print("Warning maximum lock 150 degrees");
    }

    Serial.print("correction gimbal angle=");
    servo.write(correction);
    Serial.print(correction);

    // correction1 = correction*0.5;                                //Store the value for next time, not required
here but again, useful in development

  }
}
```