AE 6210A

Advanced Dynamics I

Computer Aided Design Project

Jean-Guillaume Durand
jean-guillaume.durand@gatech.edu

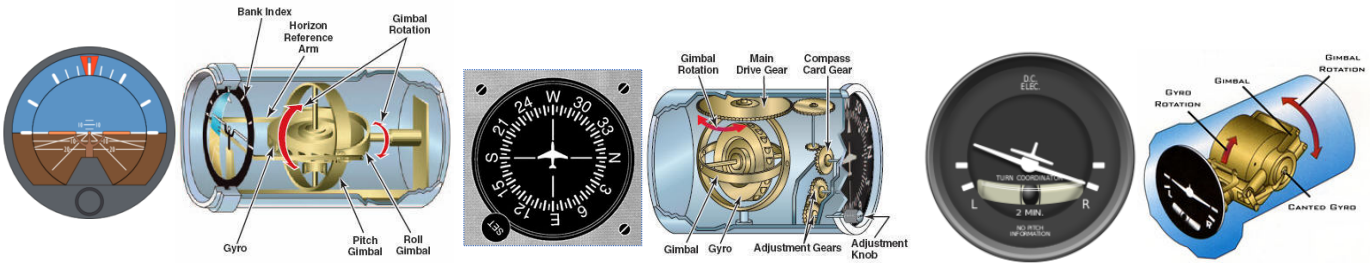
Kaivalya S Bakshi
kbakshi@gatech.edu

December 6, 2012

Contents

1 Introduction

Rate gyroscopes are used in many instruments of an aircraft's cockpit to give the pilot informations about the orientation of the plane in space as well as the rotation rates. Figure 1 from [2] shows the use of gyroscopes in the attitude indicator 1(a), the heading indicator 1(b) and in the turn coordinator 1(c).



(a) Attitude indicator (b) Heading indicator (c) Turn coordinator

Figure 1: Use of gyroscopes in the cockpit

In this project, we propose to design a single axis gyroscope in order to measure the pitch rate of a small unmanned air vehicle (UAV). The gyroscope will be first modelized and we will validate the model to insure the correctness of the different equations. Then, parametric design trade studies will be carried out to adapt the gyroscope to the small UAV and insure the quality of the measurements. The output of such a gyroscope could then be used for angular velocity feedback inside the flight control system of the vehicle.

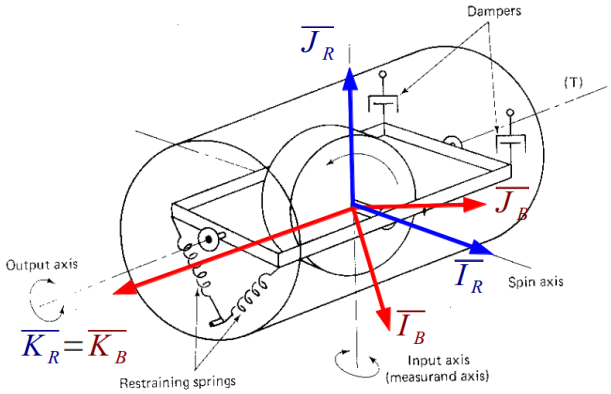
2 Model Description

In this section, we propose to describe the dynamic model of the single axis gyroscope attached to an aircraft capable of general 6 degrees of freedom (DOF) motion. Very few assumptions will be made at first to allow complete further design of the gyroscope (masses, size, damping and stiffness, placement and orientation...). A Newton-Euler method will be used.

2.1 Complete model

The main assumption used is that the hinge joints are frictionless.

2.1.1 Frames and geometry



- I-frame fixed to the ground
- P-frame fixed to airplane
- B-frame fixed to device
- R-frame fixed to disk support

The orientation of the device with respect to the airplane is parameterized by α, β, γ such that the rotation matrix from P-frame to B-frame is given by $R_B^P = R_I(\alpha)R_J(\beta)R_K(\gamma)$.

With that orientation we have

$$\begin{aligned}\overline{\omega}_{B/I} &= p\overline{I}_P + q\overline{J}_P + r\overline{K}_P \\ &= \Omega_X\overline{I}_B + \Omega_Y\overline{J}_B + \Omega_Z\overline{K}_B\end{aligned}$$

Complete formulas are

$$\begin{aligned}\Omega_X &= \Omega_X(\alpha,\beta,\gamma,p,q,r) \\ \Omega_Y &= \Omega_Y(\alpha,\beta,\gamma,p,q,r) \\ \Omega_Z &= \Omega_Z(\alpha,\beta,\gamma,p,q,r)\end{aligned}$$

with

$$\begin{aligned}\Omega_X(\alpha,\beta,\gamma,p,q,r) &= \{r(\text{Cos}[\theta]\text{Sin}[\beta] + \text{Cos}[\beta]\text{Sin}[\gamma]\text{Sin}[\theta]) + p(\text{Cos}[\alpha]\text{Cos}[\beta]\text{Cos}[\theta] - (\text{Cos}[\gamma]\text{Sin}[\alpha] + \text{Cos}[\alpha]\text{Sin}[\beta]\text{Sin}[\gamma])\text{Sin}[\theta]) \\ &\quad + q(-\text{Cos}[\beta]\text{Cos}[\theta]\text{Sin}[\alpha] - (\text{Cos}[\alpha]\text{Cos}[\gamma] - \text{Sin}[\alpha]\text{Sin}[\beta]\text{Sin}[\gamma])\text{Sin}[\theta])\} \\ \Omega_Y(\alpha,\beta,\gamma,p,q,r) &= \{p(\text{Cos}[\theta](\text{Cos}[\gamma]\text{Sin}[\alpha] + \text{Cos}[\alpha]\text{Sin}[\beta]\text{Sin}[\gamma]) + \text{Cos}[\alpha]\text{Cos}[\beta]\text{Sin}[\theta]) + q(\text{Cos}[\theta](\text{Cos}[\alpha]\text{Cos}[\gamma] \\ &\quad - \text{Sin}[\alpha]\text{Sin}[\beta]\text{Sin}[\gamma]) - \text{Cos}[\beta]\text{Sin}[\alpha]\text{Sin}[\theta]) + r(-\text{Cos}[\beta]\text{Cos}[\theta]\text{Sin}[\gamma] + \text{Sin}[\beta]\text{Sin}[\theta])\} \\ \Omega_Z(\alpha,\beta,\gamma,p,q,r) &= \{r\text{Cos}[\beta]\text{Cos}[\gamma] + q(\text{Cos}[\gamma]\text{Sin}[\alpha]\text{Sin}[\beta] + \text{Cos}[\alpha]\text{Sin}[\gamma]) + p(-\text{Cos}[\alpha]\text{Cos}[\gamma]\text{Sin}[\beta] + \text{Sin}[\alpha]\text{Sin}[\gamma])\}\end{aligned}$$

Those combinations will allow to decouple the equations by changing the orientation of the device in the UAV. Indeed, we would like to measure only q.

2.1.2 Inertia matrices

The inertia matrix of the disk is the usual matrix for a hollow cylinder but the matrix for the support has to be derived with compouned body formulas. We consider the support as compouned of cuboids and a central cylinder as the axis. Using the notations given in figure 2 we obtain the two matrices in the R-frame. **The hinge joints are neglected.**

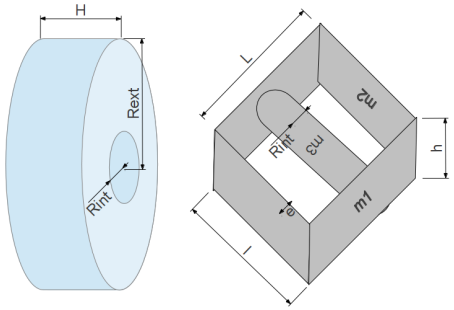


Figure 2: Geometry notations used for disk and support

$$I_{Disk/R} = \begin{pmatrix} \frac{m_D}{2}\left(R^2 + R_{int}^2\right) & 0 & 0 \\ 0 & \frac{m_D}{4}\left(R^2 + R_{int}^2 + \frac{H^2}{3}\right) & 0 \\ 0 & 0 & \frac{m_D}{4}\left(R^2 + R_{int}^2 + \frac{H^2}{3}\right) \end{pmatrix} = \begin{pmatrix} I_{XX} & 0 & 0 \\ 0 & I_{YY} & 0 \\ 0 & 0 & I_{ZZ} \end{pmatrix}$$

$$I_{Support/R} = \begin{pmatrix} I_{SX} & 0 & 0 \\ 0 & I_{SY} & 0 \\ 0 & 0 & I_{SZ} \end{pmatrix}$$

with

$$\begin{aligned}I_{SX} &= m_1\left(\frac{L^2}{6} + \frac{e^2}{6} + \frac{(L-e)^2}{2}\right) + m_2\left(\frac{h^2}{6} + \frac{L^2}{6}\right) + m_3\left(\frac{R_{int}^2}{2}\right) \\ I_{SY} &= m_1\left(\frac{e^2}{6} + \frac{l^2}{6} + \frac{(L-e)^2}{2}\right) + m_2\left(\frac{e^2}{6} + \frac{L^2}{6} + \frac{(l-e)^2}{2}\right) + m_3\left(\frac{R_{int}^2}{4} + \frac{(l-e)^2}{12}\right) \\ I_{SZ} &= m_1\left(\frac{l^2}{6} + \frac{h^2}{6}\right) + m_2\left(\frac{h^2}{6} + \frac{e^2}{6} + \frac{(l-e)^2}{2}\right) + m_3\left(\frac{R_{int}^2}{4} + \frac{(l-e)^2}{12}\right)\end{aligned}$$

2.1.3 Accelerations and Momentums

The only equations that will be used to derive our model are along $\overline{K}_B$ thus we only write the measure numbers of the quantities along $\overline{K}_B$.

$$\frac{{}^I d\overline{H}^{\oplus}_{Support/I}}{dt} \cdot \overline{K}_B = I_{SZ}\left(\ddot{\theta} + \dot{\Omega}_Z\right) + (I_{SY} - I_{SX})\left(\Omega_X \cos(\theta) + \Omega_Y \sin(\theta)\right)\left(\Omega_Y \cos(\theta) - \Omega_X \sin(\theta)\right)$$

$$\frac{{}^I d\overline{H}^{\oplus}_{Disk/I}}{dt} \cdot \overline{K}_B = I_{ZZ}\left(\ddot{\theta} + \dot{\Omega}_Z\right) + (I_{YY} - I_{XX})\left(\Omega_X \cos(\theta) + \Omega_Y \sin(\theta)\right)\left(\Omega_Y \cos(\theta) - \Omega_X \sin(\theta)\right) - I_{XX}\omega\left(\Omega_Y \cos(\theta) - \Omega_X \sin(\theta)\right)$$

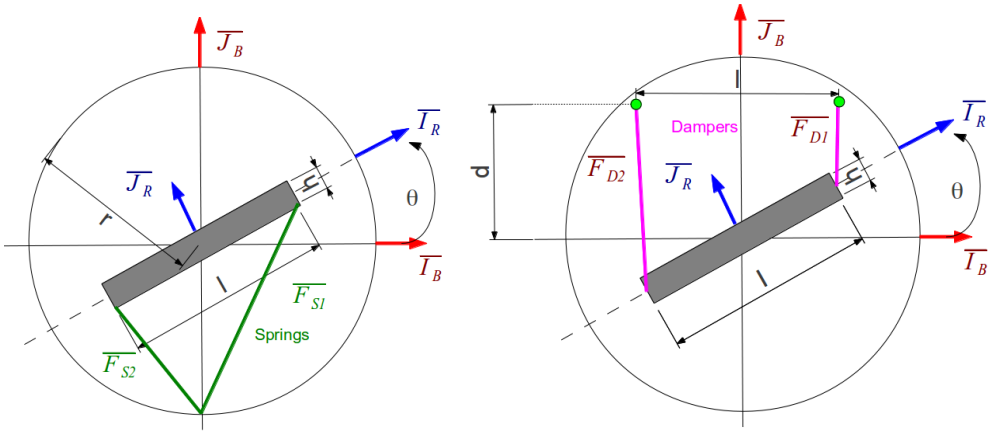
2.1.4 Forces and moments

On the disk

- weight $\overline{W}_D = -mg\overline{J}_B$
- axis reaction (frictionless) $\overline{R}_D = R_X\overline{I}_R + R_Y\overline{J}_R + R_Z\overline{K}_R$ and $\overline{M}^O_D = M_Y\overline{J}_R + M_Z\overline{K}_R$

On the support

- weight $\overline{W}_S = -m_S g\overline{J}_B$
- using the moment transport formula, we show that the hinge joints in A and B are equivalent to one (frictionless) $\overline{T} = T_X\overline{I}_B + T_Y\overline{J}_B + T_Z\overline{K}_B$ and $\overline{Q}^O_D = Q_X\overline{I}_B + Q_Y\overline{J}_B$
- Springs $\overline{F}_{S1}$ and $\overline{F}_{S2}$ and moments $\overline{M}^O_{S1}$ and $\overline{M}^O_{S2}$
- Damping $\overline{F}_{D1}$ and $\overline{F}_{D2}$ and moments $\overline{M}^O_{D1}$ and $\overline{M}^O_{D2}$
- Reaction of disk $-\overline{R}_D$ and $-\overline{M}^O_D$



(a) Springs (b) Dampers

Figure 3: Geometry assumptions used for moments calculation

A preliminary analysis without writing down the expression of the moment due to springs and dampers shows that we will be using the equation of motion along $\overline{K}_B$ axis to derive the dynamic model. Only those components will be studied then and using the geometry described on figure 3, we get the following expression for this moment due to the dampers and the springs :

$$\begin{aligned}\overline{M}^O_{Springs} &= \frac{1}{2}klr\sqrt{l^2+4r^2}\mathrm{Cos}[\theta]\left(-\frac{1}{\sqrt{l^2+4r^2-4lr\mathrm{Sin}[\theta]}} + \frac{1}{\sqrt{l^2+4r^2+4lr\mathrm{Sin}[\theta]}}\right)\overline{K}_B \\ &= \left(\frac{-2\left(kl^2r^2\right)\theta}{l^2+4r^2} + O[\theta]^2\right)\overline{K}_B\end{aligned}$$

$$\overline{M}^{O}_{Dampers} = \frac{-\left(cl^2\dot{\theta}\left(-l^2\left(20d^2+l^2\right)\text{Cos}[\theta]+2\left(2d^2+l^2\right)\left(4d^2+l^2+\left(4d^2-l^2\right)\text{Cos}[2\theta]\right)+l^2\left(4d^2+l^2\right)\text{Cos}[3\theta]\right)\right)}{\left(8\left(8d^4+4d^2l^2+3l^4+l^2\left(-4\left(2d^2+l^2\right)\text{Cos}[\theta]+\left(4d^2+l^2\right)\text{Cos}[2\theta]\right)\right)\right)}\overline{K}_B$$
$$= \left(-\frac{1}{2}cl^2\dot{\theta}+O[\theta]^2\right)\overline{K}_B$$

The complete moment is written $M'_z(\theta) = \overline{M}^{O}_{Dampers} + \overline{M}^{O}_{Springs}$. The second expression are the linearized one when a small angles approximation is used. In this latter case, we have the following simplification $M'_z(\theta) = -C\dot{\theta} - K\theta$. With $K = \frac{-2\left(kl^2r^2\right)}{l^2+4r^2}, C = \frac{1}{2}cl^2$.

NB: we can see that the distance d from the frame to the dampers does not act on the moment generated by the dampers.

2.1.5 Equations of motion

The equation that is going to be our dynamic model is the one yielding from the mass momentums along axis $\overline{K}_R$. Writing the equations we obtain :

$$\mathbb{C}_B\left(\frac{{}^I d\overline{H}^{\oplus}_{Support/I}}{dt}\right) = \left\{\begin{array}{c}\bullet\\ \bullet\\ -M_Z + M'_z(\theta)\end{array}\right\} \text{ and } \mathbb{C}_B\left(\frac{{}^I d\overline{H}^{\oplus}_{Disk/I}}{dt}\right) = \left\{\begin{array}{c}\bullet\\ \bullet\\ M_Z\end{array}\right\}$$

with $M'_z(\theta) = \frac{kl}{2}\left[-(x_1-x_0)\sin(\gamma_1)+(x_2-x_0)\cos(\gamma_2)\right] + \left(\frac{l^2c}{4}(1-\cos(\theta))\sin(\theta)+\frac{lc}{2}\left(h-\frac{l}{2}\sin(\theta)\right)\cos(\theta)\right)\left[\frac{\dot{x_1}}{\|\overline{AA'}\|}+\frac{\dot{x_2}}{\|\overline{BB'}\|}\right]$

Solving for M_Z in the equations on the left (support) and injecting it in the expression on the left (disk) gives the final equation :

$$\boxed{(I_{ZZ}+I_{SZ})\left(\ddot{\theta}+\dot{\Omega}_Z\right)+\left[(I_{YY}+I_{SY})-(I_{XX}+I_{SX})\right](\Omega_X\cos(\theta)+\Omega_Y\sin(\theta))(\Omega_Y\cos(\theta)-\Omega_X\sin(\theta))-I_{XX}\omega(\Omega_Y\cos(\theta)-\Omega_X\sin(\theta)) = M'_z(\theta)} \quad (1)$$

2.2 Model simplification

As we have to design the gyroscopic system, we can choose $\omega \gg \Omega_X, \Omega_Y, \Omega_Z$. Then, we will choose the dynamics of the system to be such that $\ddot{\theta} \gg \dot{\Omega}_Z$. The orientation of the gyroscope in the UAV is chosen with $\alpha = 0, \beta = \frac{\pi}{2}, \gamma = 0$ such that $(\Omega_Y\cos(\theta) - \Omega_X\sin(\theta)) = q$. Finally, the inertia of the support along $\overline{K}_R$ can also be choosen such that $I_{SZ} \gg I_{ZZ}$. Consequently, the following steps are summarized in the following equations to yield the simplified 1 DOF final dynamic model.

$$\begin{aligned}(1) &\Rightarrow (I_{ZZ}+I_{SZ})\left(\ddot{\theta}+\dot{\Omega}_Z\right) - I_{XX}\omega(\Omega_Y\cos(\theta)-\Omega_X\sin(\theta)) = M'_z(\theta)\\ &\Rightarrow (I_{ZZ}+I_{SZ})\left(\ddot{\theta}+\dot{\Omega}_Z\right) - I_{XX}\omega q = M'_z(\theta)\\ &\Rightarrow (I_{ZZ}+I_{SZ})\ddot{\theta} - I_{XX}\omega q = M'_z(\theta)\\ &\Rightarrow\end{aligned}$$

$$\boxed{I_{ZZ}\ddot{\theta} - I_{XX}\omega q = M'_z(\theta)} \quad (2)$$

Equation 2 is our 1 DOF dynamic model for the gyroscopic device.

Using the linearized expression of $M'_z(\theta)$ we obtain $\boxed{I_{ZZ}\ddot{\theta} + C\dot{\theta} + K\theta = I_{XX}\omega q}$. This linearization is legit if we choose the damping and the stiffness in a way that it limits the amplitude of θ and thus enables to apply **small angle approximation**.

3 Model Validation

Before going further into the design of the gyroscopic device, we need to make sure the derived dynamic model is correct. To do so, we can use a different method to derive the equations of motion, check the consistency of the equations and make sure the coding is correct with unit tests.

3.1 Lagrange method

Since a Newton-Euler method was used to derive the dynamic model, a good method to check the equations is to derive them using the Lagrange method on the system $\{Support + Disk\}$ with only 1 DOF, θ. For this derivation, we choose to use the linearized expressions.

Kinetic energy
$T_{S/I} = \frac{1}{2}(I_{ZZ}+I_{SZ})\left[\dot{\theta}+\Omega_Z\right]^2+\frac{1}{2}(I_{YY}+I_{SY})\left[\Omega_Y\cos(\theta)-\Omega_X\sin(\theta)\right]^2+\frac{1}{2}I_{XX}\left[\omega+\Omega_X\cos(\theta)+\Omega_Y\sin(\theta)\right]^2+\frac{1}{2}I_{SX}\left[\Omega_X\cos(\theta)+\Omega_Y\sin(\theta)\right]^2$
Potential energy $V_{S/I} = \frac{1}{2}K\theta^2$
Generalized force $Q_\theta = -C\dot{\theta}$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}_{S/I}}{\partial \dot{\theta}}\right) - \frac{\partial \mathcal{L}_{S/I}}{\partial \theta} = Q_\theta^{NC}$$
$$\Rightarrow (I_{ZZ}+I_{SZ})\left(\ddot{\theta}+\dot{\Omega}_Z\right)+\left[(I_{YY}+I_{SY})-(I_{XX}+I_{SX})\right](\Omega_X\cos(\theta)+\Omega_Y\sin(\theta))(\Omega_Y\cos(\theta)-\Omega_X\sin(\theta))-I_{XX}\omega(\Omega_Y\cos(\theta)-\Omega_X\sin(\theta))+K\theta=-C\dot{\theta}$$

The obtained equation is the same as the one derived with Newton-Euler's method so we have all reasons to believe that the derived dynamic model is correct now. The same assumptions can be made to derive the simpler model.

3.2 Model consistency

A simple way to verify the consistency of the equations is first to make sure they are homogeneous in terms of units.

Here we have $[M]\cdot[L]^2\cdot[T]^{-2}$ on both sides thus they are homogeneous and correspond to a moment dimension which is physically correct.

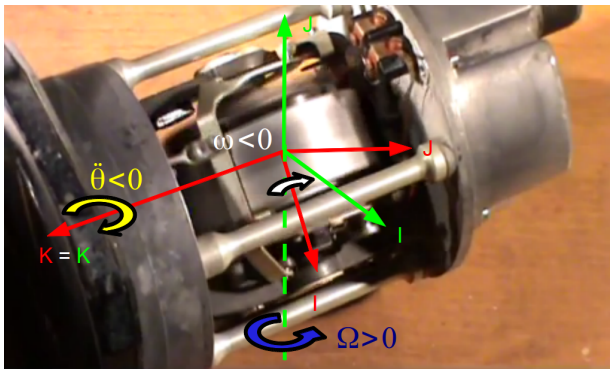


Figure 4: Video [3] used to observe dynamic behavior of a real gyroscope similar to our device

Secondly, one can check the consistency of the model by comparing it to a real physical model. We used [3] to verify the correct evolution of θ for a given input. The signs were checked to make sure the dynamic model responded in the same way as a real physical gyroscopic device for the same input. In [3] represented in figure 4 a positive rotation of the device along the input axis would give a negative acceleration response in $\ddot{\theta}$ (with a negative rotation speed of the disk ω as can be seen in the video). With our model $I_{ZZ}\ddot{\theta} - I_{XX}\omega\Omega_Y = M'_z(\theta)$ the same evolution can be observed, a positive input Ω would give a negative acceleration output $\ddot{\theta}$ (still with a negative ω). Thus, our model is consistent with a real physical model of a gyroscope.

3.3 Unit tests

Even if the simplified model is not long to code, a method has to check the correctness of the coded equations. To do so, we coded unit tests which are the basic methodology to verify the coding of functions. Those unit tests were coded for particular values of the design variables to make sure that the inertia coding was correct for instance as well as the signs of the coefficients. Finally a display of the coded equation enables to detect errors at a glance.

3.4 Code checking

Using the limit case when $K = 0$, we obtain what is called an integrating gyroscope. Indeed, using the Laplace transform on the corresponding equation :

$$I_{ZZ}\ddot{\theta} + C\dot{\theta} = I_{XX} * \omega * \Omega_Y \underbrace{\longrightarrow}_{\mathcal{L}} \Theta(s)\left(I_{ZZ}s^2 + Cs\right) = I_{XX} * \omega * \Omega_Y(s)$$

$$\Rightarrow \Theta(s) = \frac{I_{XX} * \omega * \Omega_Y(s)}{s\left(I_{ZZ}s + C\right)}$$

$$\Rightarrow \Theta(s) = \frac{I_{XX} * \omega}{c}\Omega_Y(s)\left[\frac{1}{s} - \frac{1}{s + \frac{C}{I_{ZZ}}}\right]$$

$$\underbrace{\longrightarrow}_{\mathcal{L}^{-1}} \theta(t) = \frac{I_{XX} * \omega}{c}\int_0^t \Omega_Y(\tau)\left[1 - exp\left(-\frac{C}{I_{ZZ}}(t-\tau)\right)\right]d\tau$$

Using a high damping C we derive a simplified expression $\theta(t) = \frac{I_{XX}*\omega}{c}\int_0^t \Omega_Y(\tau)d\tau$. Thus θ represents directly the pitch Euler angle (and not the pitch rate, for $\Omega_Y = p$. Comparing the output θ of the gyroscope, the calculated integral and the pitch Euler angle provided in the flight data, we are able to check if we obtain the same results (up to scale, figure 5).

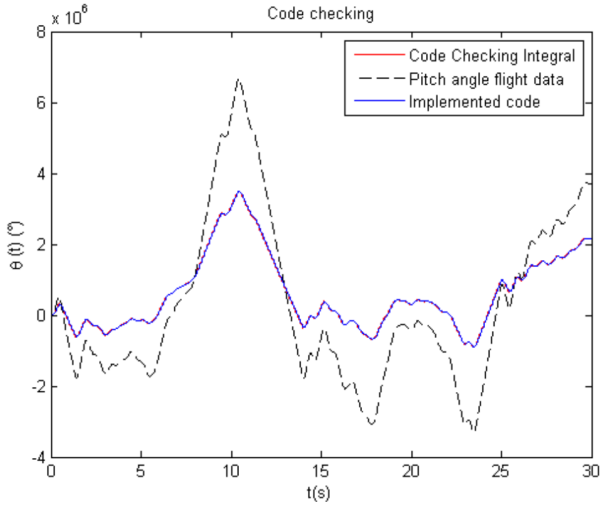


Figure 5: Integrating gyroscope to check coding

Finally, for a given set of standard parameters for the gyroscopic device, we checked the output of our system with the output of the team composed of Florian Hauer and Emmanuel Boidot. We obtained the same variations thus believe our coding is correct.

4 Parametric Design Trade Studies

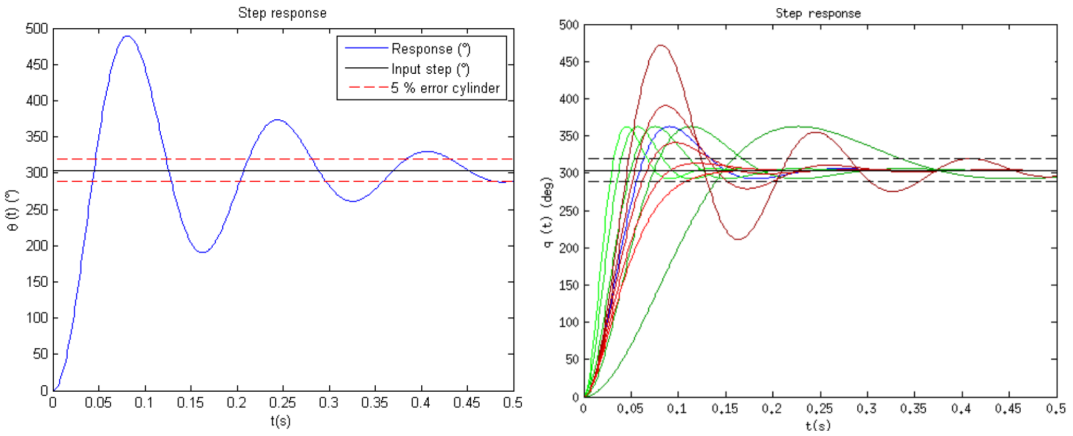
4.1 Pitch step response

The dynamic model (2) can be seen as a second order system, using the Laplace transform we can easily determine its step response. We assume the initial conditions are $\dot{\theta}(t=0) = 0rad.s^{-1}, \theta(t=0) = 0rad$.

$Eq.(2) \underbrace{\longrightarrow}_{\mathcal{L}aplace} \Theta(p)\left(I_{ZZ}p^2 + Cp + K\right) = I_{XX}\omega\Omega_Y(p)$. The transfer function then becomes $H(p) = \frac{G_0}{1+\frac{2\xi}{\omega_0}p+\frac{1}{\omega_0{}^2}p^2}$ with $G_0 = \frac{I_{XX}\omega}{K}, \omega_0 = \sqrt{\frac{K}{I_{ZZ}}}, \xi = \frac{C}{2\sqrt{KI_{ZZ}}}$. The step response of a second order system is completely parameterized via those three parameters that will be our design trade studies variables. Using the inverse Laplace transform of $H(p)$ for a step input $\Omega_Y(p) = \frac{1}{p}$ we obtain the time response for a pitch step input :

$$-\frac{G_0}{2\sqrt{\left(\xi^2-1\right)\omega_0{}^2}}e^{-t\left(\xi\omega_0+\sqrt{\left(\xi^2-1\right)\omega_0{}^2}\right)}\left(\left(-1+e^{2t\sqrt{\left(\xi^2-1\right)\omega_0{}^2}}\right)\xi\omega_0+\left(1+e^{2t\sqrt{\left(\xi^2-1\right)\omega_0{}^2}}-2e^{t\left(\xi\omega_0+\sqrt{\left(\xi^2-1\right)\omega_0{}^2}\right)}\right)\sqrt{\left(\xi^2-1\right)\omega_0{}^2}\right)$$

Such a response can be represented as seen on figure 8, and regenerated using the parameters in Table 7.



(a) Step response of our second order system

(b) Influence of ω_0 (green) and ξ (red), the brighter the color, the greater the value

Figure 6: Step response analysis

As we can see on figure 8, the response is not optimized at all, the first shootout is about 61 % at t = 0.012561 s. **The parameters are $G_0 = 5.302686, \omega_0 = 39.101295 rad/s, \xi = 0.154462 (\neq 0.69)$.**

NB: [4] classes of automatics were used to understand the basic tradeoffs of the parameters and the automatics theory which we do not discuss here, especially optimization of a second order system step response (damping, oscillations, reactivity...).

4.2 Trade off studies

As the response to a step in pitch rate is perfectly know in the linear case, we can characterize the response in terms of precision, rapidity and stability with only the design parameters. If the step response is correct in those terms, we have all reasons to believe that the system output will nicely follow the pitch rate input in the general case. In every automatic system, the step response is often the occasion to get the first design numbers and figures.

To do so, we chose first to work at fixed geometry by taking into account the size of the aicraft. Only k and c were left to decide for optimizing the step response. The best tradeoff in the step response of a second order system is when $\xi = 0.69$ because it represents only one overshoot with 5% relative error and an optimal response time [4]. Then, the pulsation ω_0 has to be high to insure reactivity of the system, as we are designing a measure instrument to be used for control laws with high frequencies, it needs to be highly reactive.

With these theory elements and the associated equations of ω_0 and ξ with only inertia terms depending on geometry (and masses) that are fixed, K and C depending also on geometry and the design parameters k and c, we are able to get the optimal k and c at a given geometry to obtain optimal step response in terms of reactivity and precision. Increasing the pulsation is done by increasing k, thus making the springs stiffer and hereby restraining the angle θ amplitude. Limiting the amplitude of the output angle is not a good thing because the angle has to be measured by electric device thus better precision and resolution will be achieved if we have large amplitudes of θ.

At this point, we have a reactive and precise system but lacking amplitude. The gain G_0 can be changed with the I_{XX} disk inertia and ω the angular speed of the disk. Those paremeters were fixed in the geometry first assessment and can now be changed to adjust the gain to obtain reasonable angle amplitude. Nonetheless, those parameters already at high values could not be changed a lot without being unfeasible for ω or sizing of the disk. System phase and gain margins were also checked (see [4]) to insure the stability of the system. Indeed, having a very good response on the given data does not guarantee a very good response on other data. For instance, we tested our

gyroscope system on the yaw and roll flight data (as pitch rate input) to make sure our device was able to perform well on other datasets.

Finally to come up with a solution with a high gain, we realized that even if the step response was far from the optimal case, the response in the experiment was fairly correct. Drastically reducing k and c we were able to get pretty reasonable amplitude of around 30 to 40 deg and optimized on k to reduce delay (phase) to 10-20 ms. Hence, we achieved our design needs with the following method:

- Fix geometry relatively to UAV size
- Obtain best tradeoff in k and c for an optimal time response
- Adjust gain with limits of feasible ω and inertia to allow good measurement precision and resolution
- To continue increasing gain, reduce optimal k to loosen the frame and adjust damping c to trade between reactivity and relative error on pitch rate measurement

Iterative Trade Study Values	
k (N/m)	c
2	0.4
5.1	0.63
20.41	1.27
10.47	0.52
25	0.1
25	0.3

The feasability of the values also has to be evaluated. Indeed, wanting to increase the pulsation can lead to very high values of the springs constant. The important parameter is in fact K where k and geometry factors are present. Thus, to keep k reasonable we can act on geometry to increase K without having unfeasible k values. The speed of the disk is limited to eletric motors capacities like 20 000-40 000 RPM.

5 Final Design

Before considering the dimensions of the device, we have to know its position and orientation in the UAV. Since no other forces or moments than $M'_z(\theta)$ (due to dampers and springs) are applied on our model, **we can put the gyroscope anywhere in the UAV** : it will not affect its behaviour. Indeed all the joints unknowns are not present in the dynamic model along $\overline{K}_R$. However, the orientation is important since we want to combine α, β, γ in order to decouple p,q and r and measure only q. We determined earlier in the dynamic model derivation that the combination $\alpha = 0, \beta = \frac{\pi}{2}, \gamma = 0$ satisfies our requirement to have only q as an input.

Now that we are certain to measure only the pitch rate of the UAV we can focus on sizing the device with our parametric design trade studies.

The final design variables are given in the following table :

Parameter	Value	Parameter	Value
Casing			
r	0.04 m	d	0.03 m
Disk			
mD	0.1 kg	R	0.03 m
R_{int}	0.01 m	H	0.01 m
RPM	40 000	ω	4187 rad/s
Frame			
m1	0.005 kg	m2	0.005 kg
m3	0.005 kg	h	0.01 m
e	0.0005 m	l	0.079 m
Dampers and Springs			
c	0.3 N/m/s	k	25 N/m

Figure 7: Final design values

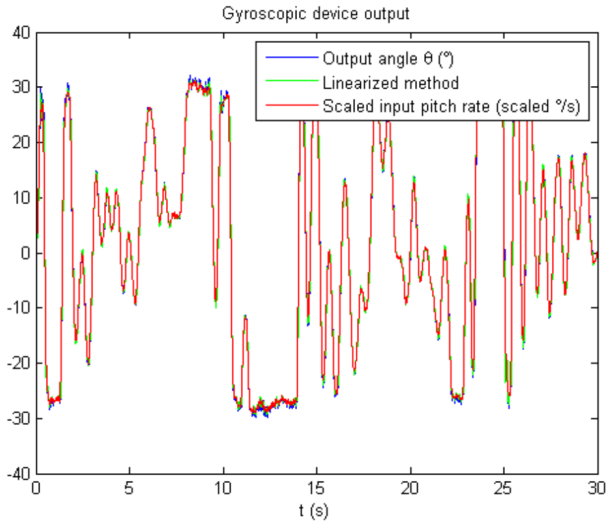


Figure 8: Final design response to flight data

6 Conclusion

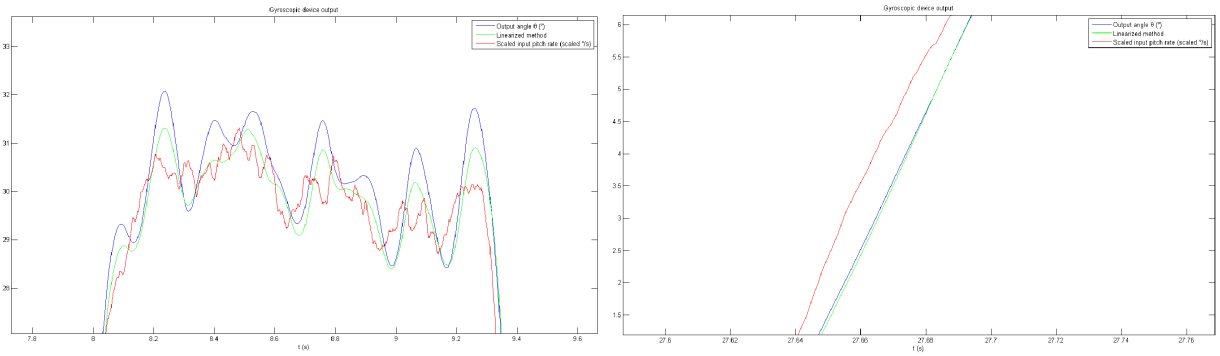
Our design needs for this gyroscopic device were :

- reactivity, delay limited to 10 ms to allow good further control laws
- precision below 5%
- output amplitude of about 30-40 deg to enable easy measurement precision and resolution
- size limited to fit in the UAV
- mass limited not to alter the UAV's flight behavior
- feasible solution (k,c,mass, ω ...)

Using the method described in the previous section, we were able to respect those requirements. We tested a linearized method using the small angles approximation (for angles below 25 deg) and a complete method without linearizing. We were able to assess the difference between the two methods as shown on figure 9. As expected, the linearized method encounters difficulties in precision for angles above 20 deg. We were also able to check that our assumptions were reasonable. Indeed, given the high value of ω, the assumptions made to derive equation 1 are respected ($\frac{p}{\omega} = 8.2816e-005, \frac{q}{\omega} = 2.4852e-005, \frac{r}{\omega} = 7.1620e-005$). Nevertheless, one of our assumption was not respected, the frame inertia has the same order of magnitude as the disk inertia (along K-axis) so cannot be neglected to derive equation 1. I_{SZ} was thus included in our calculations.

The obtained results are summarized in the following table :

Error (%)	Linearized Case	Complete model
Mean	8.3	9.1
Median	2.9	3.7



(a) Precision and reactivity close look

(b) 10-20 ms reactivity

Figure 9: Results

The size of the device is adapted to the UAV and the gyroscopic induced effect of it was checked to have negligible effect on the dynamics of the UAV. Indeed the gyroscopic effect can be written as $\overline{\omega}_{Aircraft/I} \times I_{Disk}.\omega_{Disk}$. The maximum value of the gyroscopic torque for our final design is : 0.0218 N.m. Compared to the aerodynamical torques on the aircraft. For instance the torque on the roll axis is $C_l \frac{1}{2}\rho V^2 Sb = -6.555377 N.m$ (with b the span of the UAV). Then, the gyroscopic effect is negligible (0.3 %).

Moreover, in the derivation of the model 1 we assumed that $\ddot{\theta} \gg \dot{\Omega}_Z$. Using the orientation chosen for the gyroscopic device, we have $\dot{\Omega}_Z = p$. Interpolating the roll rate data to compute $\dot{\Omega}_Z$ we were able to include that correction term in the equations. The final result is not affected by this new term in the case where $\ddot{\theta} \gg \dot{\Omega}_Z$. Hence, our preliminary assumption was justified.

NB: if the system were to be designed for visual feedback only (attitude indicator for instance), the reactivity could be of around 100 ms. Allowing greater precision and amplitude.

References

[1] M. Costello, *AE 6210A Advanced Dynamics I class at Georgia Institute of Technology*, 2012

[2] *Gyroscopic instruments*, Flight Instructor Wiki, http://cfi-wiki.net/w/Gyroscopic_Instruments

[3] *How Gyroscopic Turn & Slip Indicator Works*, Youtube video, `http://www.youtube.com/watch?v=0sRrSkSJc7w&feature=related`

[4] Stéphane Genouel, *04-Systèmes linéaires continus et invariants*, classes of automatics Lycée Chateaubriand Rennes, France, 2009-2010

[5] D. Bruce Owens , Development of a Low-Cost Sub-Scale Aircraft for Flight Research: The FASER Project, NASA Langley Research Center

7 Appendix : Code

7.1 Main

```
% AE - 6210 Advanced Dynamics I
% Computer Aided Project
% Unit tests on implemented functions

%% Cleaning and preparing the workspace
clc, close all, clear all;

% Global values to give to the model function
global tData
global qData
global pData

% Linear mode
global C
global Ixx
global Izz
global K
global omega
% Non linear
global k
global l
global r
global d
global c

%% Loading data
data = load('../data/Flight_Data.txt');

tData = data(:,1);
xData = data(:,2);
yData = data(:,3);
zData = data(:,4);
uData = data(:,8);
vData = data(:,9);
wData = data(:,10);
phiData = data(:,5);
thetaData = data(:,6);
psiData = data(:,7);
pData = data(:,11);
qData = data(:,12);
rData = data(:,13);

%% Design variables
% Units in USI

% Casing
r = 0.04;
d = 0.03;

% Disk
mD = 0.1;
R = 0.03;
Rint = 0.01;
H = 0.01;
RPM = 40000;
omega = RPM*pi/30;

% Frame
m1 = 0.005;
m2 = 0.005;
m3 = 0.005;
h = 0.01;
e = 0.0005;
l = 0.079;

% Dampers
% c = 0.4;
% c = 0.63365
% c = 1.2673;
% c = 0.517008;
% c = 0.1;
c = .3;

% Springs
% k = 2;
% k = 5.10185;
% k = 20.4074;
```

```
% k = 10.471;
% k = 25;
k = 25;

%% Intermediate variables
Ixx = mD/2*(R^2+Rint^2);
Izz = mD/4*(R^2+Rint^2+H^2/3);
Isz = m1*(l^2/6 + h^2/6) + m2*(h^2/6 + e^2/6 + (l - e)^2/2) + m3*(Rint^2/4 + (l - e)^2/12);

K = 1/2 * l^2 * (4*k*r^2)/(l^2 + 4*r^2);
% K = 0;   % For code checking
C = 1/2*c*l^2;

fprintf('Intermediate variables :\n');
fprintf('Ixx = %f kg.m^2\n',Ixx);
fprintf('Izz = %f kg.m^2\n',Izz);
fprintf('Isz = %f kg.m^2\n',Isz);
fprintf('K = %f kg.m^2/s^2\n',K);
fprintf('C = %f kg.m^2/s\n',C);

%% Second order system variables
xi = C/(2*sqrt(K*Izz));
omega0 = sqrt(K/Izz);
G0 = Ixx*omega/K;

fprintf('\nSecond order system variables :\n');
fprintf('xi = %f\n',xi);
fprintf('omega0 = %f\n',omega0);
fprintf('G0 = %f\n',G0);

%% Gyroscopic effect

fprintf('Gyroscopic torque is : %f N.m\n',max(qData)*Ixx*omega);
fprintf('Moment on roll axis is : %f N.m\n', -0.02*0.5*1.225*20^2*0.6558*2.04);

%% Step response analysis
tStep = 0:0.001:0.5;
figure, plot(tStep,180/pi*stepResponse(tStep,G0,omega0,xi),'b');
hold on,
plot(tStep,180/pi*G0*ones(size(tStep)),'k')
plot(tStep,180/pi*0.95*G0*ones(size(tStep)),'r--')
plot(tStep,180/pi*1.05*G0*ones(size(tStep)),'r--')
legend('Response (deg)','Input step (deg)','5 % error cylinder');
title('Step response');
xlabel('t(s)');
ylabel('\theta (t) (deg)');
hold off;

% Relative error on first pic with correspondent time
D1rel = exp(-xi*pi/sqrt(1-xi^2));
t1 = xi*pi/(omega0*sqrt(1-xi^2));

% Resonance
omegaR = omega0*sqrt(1-2*xi^2);
Gr = G0/(2*xi*sqrt(1-xi^2));

fprintf('\nStep response caracterization :\n');
fprintf('D1rel = %f at t = %f s\n',D1rel,t1);
fprintf('omega0 = %f rad/s\n',omega0);
fprintf('G0 = %f\n',G0);
if (xi < 0.7)
    fprintf('\tResonance :\n');
    fprintf('\tomegaR = %f rad/s',omegaR);
    fprintf('\tGr = %f\n',Gr);
end

%% Numerical simulation

% Initialization
tspan = [0 tData(end)];
z0 = [0 0];
% Iterations
[tLin, zLin] = ode45(@projfuncLin, tspan, z0);
[t, z] = ode45(@projfunc, tspan, z0);

%% Display
figure, plot(t,180/pi*z(:,1),'b')
hold on,
plot(tLin,180/pi*zLin(:,1),'g')
plot(tData,180/pi*G0*qData,'r')
legend('Output angle \theta (deg)','Linearized method','Scaled input pitch rate (scaled deg/s)');
title('Gyroscopic device output');
xlabel('t (s)');
hold off;
```

```

%% Error
qDataBisLin = interp1(tData,qData,tLin);
errorLin = min(abs((zLin(:,1)-G0*qDataBisLin)./(G0*qDataBisLin)),0.5);
qDataBis = interp1(tData,qData,t);
error = min(abs((z(:,1)-G0*qDataBis)./(G0*qDataBis)),0.5);

figure, plot(t,error,'b');
hold on;
plot(tLin,errorLin,'g');
hold off;
legend('Relative error','Relative error for linearized method');
title('Error');

% Display
fprintf('\n Error :\n');
fprintf('Mean error = %f \t Median error = %f\n',mean(error),median(error));
fprintf('Mean errorLin = %f \t Median errorLin = %f\n',mean(errorLin),median(errorLin));

%% Code checking
thetaCheck = Ixx*omega/C*cumtrapz(qData);
figure, plot(tData,180/pi*thetaCheck,'r');
hold on,
plot(tData,0.5*1e6*180/pi*thetaData,'--k');
plot(tLin,1e3*180/pi*zLin(:,1),'b')
legend('Code Checking Integral','Pitch angle flight data','Implemented code');
title('Code checking');
xlabel('t(s)');
ylabel('\theta (t) (deg)');
hold off;
```

7.2 Step Response

```
function output = stepResponse(t,G0,Omega,Xi)
    output = -(1/(2.*sqrt((-1 + Xi^2).*Omega^2))).*exp(-t.*(Xi.*Omega + sqrt((-1 + Xi^2).*Omega^2)))...
        .*...
        G0.*((-1 + exp(2.*t.*sqrt((-1 + Xi^2).*Omega^2))).*Xi.*Omega + (1 + exp(2.*t.*sqrt((-1 + Xi^2)...
            .*Omega^2)) -...
        2.*exp(t.*(Xi.*Omega + sqrt((-1 + Xi^2).*Omega^2)))).*sqrt((-1 + Xi^2).*Omega^2));
end
```

7.3 Moment functions

7.3.1 Non-linearized

```
function output = moment(z, k,l,r, d,c)
    output = ...
        (1/2*k*l*r*sqrt(l^2+4*r^2)*cos(z(1))*(-1/sqrt(l^2+4*r^2-4*r*l*sin(z(1))) + 1/sqrt(l^2+4*r^2+4*...
            r*l*sin(z(1)))))+ ... % Springs moment
        ... % Dampers moment
        -(c*l^2*z(2)*(-l^2*(20*d^2 + l^2)*cos(z(1)) + ...
     2*(2*d^2 + l^2)*(4*d^2 + l^2 + (4*d^2 - l^2)*cos(2*z(1))) + ...
     l^2*(4*d^2 + l^2)*cos(3*z(1))))/(8*(8*d^4 + 4*d^2*l^2 + 3*l^4 + ...
     l^2*(-4*(2*d^2 + l^2)*cos(z(1)) + (4*d^2 + l^2)*cos(2*z(1))))));
end
```

7.3.2 Linearized

```
function output = momentLin(z, K, C)
    output = -C*z(2)- K*z(1);
end
```

7.4 Dynamic model functions for ODE45

7.4.1 Non-linearized

```
function zdot = projfunc( t, z )

% Get global variables for design
% Linear mode
global Ixx
global Izz
global omega
% Non linear
global k
global l
global r
global d
global c

global tData
global qData

% Interpolation of the input data to fit with simulation time step
q = interp1(tData, qData, t);
% State update
zdot(1,1) = z(2);
% Complete moment expression
zdot(2,1) = moment(z, k,l,r, d,c)/Izz + Ixx*omega*q/Izz;

end
```

7.4.2 Linearized

```
function zdot = projfuncLin( t, z )

% Get global variables for design
% Linear mode
global C
global Ixx
global Izz
global K
global omega

global tData
global qData
global pData

% Interpolation of the input data to fit with simulation time step
q = interp1(tData, qData, t);

pdot = (interp1(tData, pData, t +0.001) - interp1(tData, pData,t))/0.001;

% State update
zdot(1,1) = z(2);
% Linearized moment
zdot(2,1) = momentLin(z, K, C)/Izz + Ixx*omega*q/Izz - pdot;
end
```