Trabajo Fin de Grado en Física

Attitude Measurement and Control Techniques on the Earth's Surface

Jorge Pretel Villanueva
Universidad de Granada

Julio de 2019

UNIVERSIDAD DE GRANADA

Universidad de Granada

Abstract

The following bachelor thesis introduces the basis of the attitude representation and measurement theory on Earth's surface from inertial and magnetic measurement units data, as well as the use of sensor fusion algorithms as the optimal and complementary filters. Moreover, a review of the control system formalism is presented, where the stability and the reach of the controllers is analyzed, as well as two of the most used control techniques at the present time: the Proportional Integral Derivative controller and the Linear Quadratic Regulator. Finally, a study of the measurement and filtering methods on a real inverted pendulum with reaction wheel is conducted.

Resumen

En el presente trabajo se realiza una introducción a la teoría de la medida de la orientación en la superficie terrestre mediante el uso de sensores inerciales y magnéticos con técnicas de fusión de señales, a través del uso de filtros óptimos y complementarios. A continuación se hace una presentación del formalismo de sistemas de control, en la cual se analiza la estabilidad y el alcance de dichos sistemas, así como dos de los métodos de control más comunes, el controlador Proporcional Integral Derivativo y el Regulador Lineal Cuadrático. Por último, se hace un estudio de la medida para un sistema real, el péndulo invertido con volante de inercia.

Agradecimientos

Quiero expresar mi agradecimiento a todo el equipo de Granasat, por su ayuda incondicional y la alegría que han sabido transmitir cada día, incluso cuando más inmersos están en sus cables y sus osciloscopios. A Andrés, mi tutor, por brindar a jóvenes estudiantes la oportunidad de soltar el libro y, a golpe de martillo y con sangre y sudor, materializar lo que algún día sólo fueron ideas.

A toda mi familia, a los que han llegado y también a los que se fueron. En especial a mi madre, por su salmorejo, fuente de inspiración y gozo en mis jornadas de trabajo.

A los amigos con los que he terminado esta carrera, con los que tantos buenos y malos ratos he compartido.

A la gente de Monachil. Al Marqués de Saldaña: sin él el mundo sería un poco peor.

En último lugar, a Lasas.

A todos y a ninguno, gracias.

Contents

1 Introduction

The study of attitude measurement and control has been indispensable for the developement of fields as aeronautical and aerospace engineering.

Every satellite has incorporated systems that allow them to know their orientation respect to the Earth, the sun, or distant stars and galaxies in the case of astronomical satellites, as well as tools to control that orientation, like reaction wheels, gimballs, thrusters, magnetorquers, ect.

Figure 1: Control moment gyroscope installed in the International Space Station [25]

Before being sent into orbit, spacecrafts are tested on ground, generally by using test platforms with air bearing systems [7], which can be constructed in such a way that the center of mass of the suspended system is coincident with the center of rotation. This way, the effect of gravitational forces and torques is reduced considerably, simulating an almost perfect zero-gravity enviroment for testing and developement of spacecraft attitude control software. Other applications of orientation measurement and control theory include the developement of autonomous navigation systems and self balancing vehicles, among others.

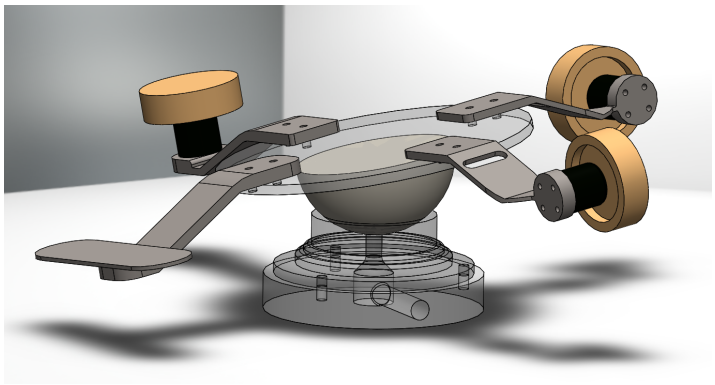

Figure 2: Test platform with air bearing designed by the Granasat team [21]

At the beginning of this work, a brief overview of the mathematical representation of the attitude is provided, as well as the different transformation rules and temporal evolution of the representation elements. After that, the different inertial measurement sensors are introduced, and the relation between the measured data and the orientation of the system respect to a ground frame is determined.

In particular, sensor fusion algorithms will be introduced for the data analysis. Sensor fusion is the combination of data from different noisy or imprecise sensors that are measuring the value of certain magnitude, in order to obtain information that is more precise than that from the individual sources. This combination can be done by making use of statistical methods and optimal estimators like the Kalman filter [25], one of the most famous and widely used filters nowadays, or taking into account some characteristics of the noise from each individual sensor and trying to take advantage of the useful part of it's data. In the case of orientation measurement, this filtering process leads to the reduction of noise from the accelerometer and magnetometer data, as well as the minimization of the characteristic bias errors that arise when gyroscope data is integrated. This constitutes a great improvement with respect to the non filtered values.

Once the orientation data is known, it can be fed into a controller that is able to manipulate the state of a certain system, turning or moving it, by using passive or active actuators. There is a wide set of control approaches available and, depending on the particular system, different methods can be used.

This work introduces the basics of such control systems from a theoretical and practical point of view.

2 Orientation representation

The orientation of a body-fixed frame of reference respect to a laboratory or Earth-surface reference frame and it's temporal evolution can be represented in many ways.

(Note: vectors are written in bold letters to distinguish them from scalars.)

2.1 Euler angles

According to Euer's rotation theorem, any rotation in $\mathbb{R}^n$ space can be described by a sequence of no more than n rotations around coordinate vectors, such that consecutive rotations are performed around distinct vectors.

Those vectors are usually chosen to be the three basis vectors $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{y}}$ and $\hat{\boldsymbol{z}}$ (but could also be $\hat{\boldsymbol{y}}$, $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{z}}$, or $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{z}}$ and $\hat{\boldsymbol{x}}$ again, etc).

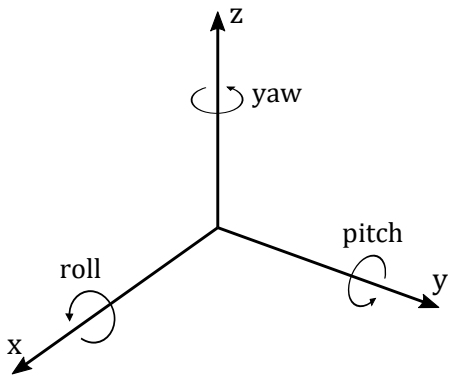

Figure 3: Roll, pitch and yaw angles

The angles of rotation around $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{y}}$ and $\hat{\boldsymbol{z}}$ are called roll (ϕ), pitch (θ) and yaw (ψ).

The change of attitude around each axis can be defined by the following three rotation matrices [1].:

$$R_x(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c(\phi) & -s(\phi) \\ 0 & s(\phi) & c(\phi) \end{pmatrix} \quad R_y(\theta) = \begin{pmatrix} c(\theta) & 0 & s(\theta) \\ 0 & 1 & 0 \\ -s(\theta) & 0 & c(\theta) \end{pmatrix}$$
$$R_z(\psi) = \begin{pmatrix} c(\psi) & s(\psi) & 0 \\ -s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.1}$$

Where $s(\alpha)$ and $c(\alpha)$ represent the sine and cosine functions. The complete rotation after the consecutive individual rotations (2.1) is given by it's product:

$$R(\phi,\theta,\psi) = \begin{pmatrix} c(\theta)c(\psi) & s(\phi)c(\psi) - c(\theta)s(\psi) & c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi) \\ c(\theta)s(\psi) & s(\phi)s(\theta)s(\psi) + c(\phi)c(\psi) & c(\phi)s(\theta)s(\psi) - s(\phi)s(\psi) \\ -s(\theta) & s(\phi)c(\theta) & c(\phi)c(\theta) \end{pmatrix} \tag{2.2}$$

This transformation $R \in SO(3)$ has the following properties:

$$R^{-1} = R^T \quad RR^T = R^T R = \mathbb{I}^{3\times 3} \tag{2.3}$$

that is, the inverse rotation transformation is described by the transpose of R: $R(-\phi,-\theta,-\psi) = R^T(\phi,\theta,\psi)$.

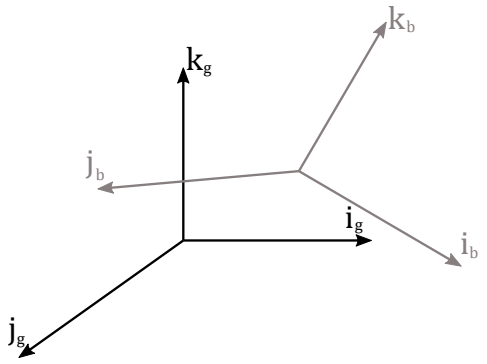

Figure 4: Ground and body frames

In general, a rotation matrix that defines the orientation of a body-fixed reference frame respect to a laboratory (ground) frame can be constructed as the projection of each one of the basis vectors $\hat{\boldsymbol{i}}_b$, $\hat{\boldsymbol{j}}_b$ and $\hat{\boldsymbol{k}}_b$ of the body frame onto each basis vector $\hat{\boldsymbol{i}}_g$, $\hat{\boldsymbol{j}}_g$ and $\hat{\boldsymbol{k}}_g$ of the ground frame:

$$R^{BG} = \begin{pmatrix} \hat{i}_g \cdot \hat{i}_b & \hat{j}_g \cdot \hat{i}_b & \hat{k}_g \cdot \hat{i}_b \\ \hat{i}_g \cdot \hat{j}_b & \hat{j}_g \cdot \hat{j}_b & \hat{k}_g \cdot \hat{j}_b \\ \hat{i}_g \cdot \hat{k}_b & \hat{j}_g \cdot \hat{k}_b & \hat{k}_g \cdot \hat{k}_b \end{pmatrix} \tag{2.4}$$

This matrix is called the direction cosine matrix (DCM), as the scalar product of each pair of unitary basis vectors is the cosine of the angle between both. This matrix is equivalent to (2.2).

A vector $\boldsymbol{v}^b$ expressed in the body reference frame can be expressed in the ground frame as:

$$\boldsymbol{v}^g = R^{BG}\boldsymbol{v}^b \tag{2.5}$$

and from ground frame to body frame:

$$\boldsymbol{v}^b = R^{GB}\boldsymbol{v}^g, \quad R^{BG}R^{GB} = \mathbb{I}^{3\times 3} \tag{2.6}$$

2.2 Quaternions

Quaternions are an extension to complex numbers with one real and three imaginary parts [10]:

$$q = s + iq_1 + jq_2 + kq_3 \tag{2.7}$$

with $i^2 = j^2 = k^2 = ijk = -1$. They are usually represented in form of a 4-dimensional vector:

$$\boldsymbol{q} = (s, \boldsymbol{v}) \tag{2.8}$$

with $\boldsymbol{v} = (q_1, q_2, q_3)$. Since only 3 values are required to fully describe a 3D rotation, the use of quaternions in the orientation description is limited to unitary vectors:

$$|\boldsymbol{q}| = \sqrt{s^2 + q_1^2 + q_2^2 + q_3^2} = 1 \tag{2.9}$$

A rotation by an angle α around an unit vector $\boldsymbol{v}$ is represented in quaternion notation as:

$$\boldsymbol{q}_{\alpha,\boldsymbol{v}} = \left(cos\frac{\alpha}{2},\, \boldsymbol{v}sin\frac{\alpha}{2}\right) \tag{2.10}$$

so the first component represents the magnitude of the rotation and the last three correspond with the rotation direction.

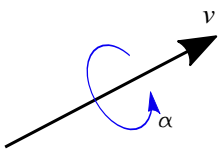

Figure 5: Quaternion representation of a rotation

The conjugate of an unit quaternion is:

$$\boldsymbol{q}^*\,(\alpha, \boldsymbol{v}) = \boldsymbol{q}\,(\alpha, -\boldsymbol{v}) \tag{2.11}$$

Similarly to Euler angles case, a rotation matrix can be obtained for a given quaternion:

$$R_q = \begin{pmatrix} 1-2q_2^2-2q_3^2 & 2q_1q_2+2sq_3 & 2q_1q_3-2sq_2 \\ 2q_1q_2+2sq_3 & 1-2q_1^2-2q_3^2 & 2q_2q_3-2sq_1 \\ 2q_1q_3-2sq_2 & 2q_2q_3-2sq_1 & 1-2q_1^2-2q_2^2 \end{pmatrix} \tag{2.12}$$

The composition ($\circ$) of two rotations $\boldsymbol{q}_a$ and $\boldsymbol{q}_b$ is given by the quaternion product:

$$\boldsymbol{q}_a \circ \boldsymbol{q}_b = (s_a, \boldsymbol{v}_a) \circ (s_b, \boldsymbol{v}_b) = (s_a s_b - \boldsymbol{v}_a \boldsymbol{v}_b,\ s_a \boldsymbol{v}_b + \boldsymbol{v}_a s_b + \boldsymbol{v}_a \times \boldsymbol{v}_b) \tag{2.13}$$

In general, the composition of quaternions does not commute.

An important thing to take into account when computing quaternions is that any precision error (floating error) can lead to not normalized results which don't represent proper rotations, so algorithms must implement a constant normalization process.

Conversion from quaternion notation to Euler angles can be done as:

$$\begin{cases} \phi = atan\left(\frac{2sq_1+q_2q_3}{1-2\left(q_1^2+q_2^2\right)}\right) \\ \theta = asin\left(2\left(sq_2-q_3q_1\right)\right) \\ \psi = atan\left(\frac{2sq_3+q_1q_2}{1-2\left(q_2^2+q_3^2\right)}\right) \end{cases} \tag{2.14}$$

Each representation has some advantages and some drawbacks. The use of Euler angles is more intuitive and all the information is contemplated in only 3 values, but it suffers from a problem called gimbal lock whereby the system loses a degree of freedom for certain continuous rotations. They are also more ambiguous than quaternions in the sense that Euler angles can be defined for different sets of axes.

Quaternions are less intuitive but they get rid of the gimbal lock problem and the computational cost associated to it's use is smaller.

3 Orientation measurement

3.1 Measurement devices

In order to obtain information relative to the orientation of a body, we have to be able to measure certain magnitudes whose directions are known to us in a ground-fixed frame. In Earth's surface, those magnitudes are the Earth's gravity and magnetic field vectors. In other enviroments like the outer space, further methods like star tracking systems can be applied. In any case, the following formalism can be applied for any orientation measurement problem in general.

3.1.1 Accelerometer

In a static situation, an accelerometer measurement provides a constant actualization of the gravity vector's direction. When the body is subjected to other accelerations, if a model of the body's motion is available, some corrections can be made in order to separate the gravitational from the rest of the dynamical accelerations. Sometimes those accelerations are small compared to gravity and can therefore be neglected. Even when the accelerations

are not small we can still make use of the mean measured value $\langle \boldsymbol{a} \rangle$ to have a rough idea of where downwards is.

The accelerometer static measurement vector is:

$$\boldsymbol{a} = (a_x, a_y, a_z)^T \tag{3.1}$$

and it's equal to the gravity vector expressed in the ground frame multiplied by the corresponding rotation matrix:

$$\boldsymbol{a} = R^T(\phi, \theta, \psi)\,(0, 0, -g)^T \tag{3.2}$$

From (2.2) we have the following correspondence:

$$\begin{pmatrix} -s(\theta)\,g \\ s(\phi)\,c(\theta)\,g \\ c(\phi)\,c(\theta)\,g \end{pmatrix} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \tag{3.3}$$

From the quotient between a_y and a_z we can obtain the roll angle:

$$\phi = atan2\left(\frac{a_y}{a_z}\right) \tag{3.4}$$

where the two argument arctangent function $atan2$ is here used to ensure that the angle remains in the $\pm\pi$ range.

From a_x and realizing that $\sqrt{a_y^2 + a_z^2} = c(\theta)\,g$, we get the pitch angle:

$$\theta = atan2\left(\frac{-a_x}{\sqrt{a_y^2 + a_z^2}}\right) \tag{3.5}$$

3.1.2 Magnetometer

Once we know the roll and pitch angles from the accelerometer measurement, the yaw angle can be computed from those angles and the magnetometer measurement.

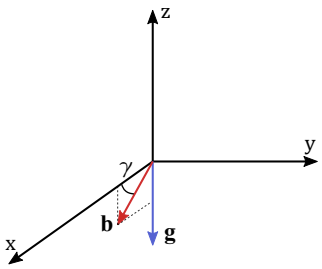

Figure 6: Gravity and magnetic vectors in ground frame

The magnetic field vector in Earth's surface has the form [9]:

$$\boldsymbol{b} = (Bcos\gamma,\ 0,\ -Bsin\gamma)^T \tag{3.6}$$

where γ is the inclination of the vector, which varies depending on the measurement location.

The vector measured by the magnetometer is:

$$\boldsymbol{m} = (m_x, m_y, m_z)^T \tag{3.7}$$

If we have already $R_x(\phi)$ and $R_y(\theta)$ matrices, we get the relation:

$$R_y(\theta) R_x(\phi) \boldsymbol{m} = R_z^T(\psi) \boldsymbol{b} = \boldsymbol{b}^r \tag{3.8}$$

where $\boldsymbol{b}^r$ represents the magnetic vector (3.6) after performing the rotation $R_z^T(\psi)$ that we want to determine. From (3.8) we have:

$$\begin{pmatrix} c(\theta) m_x + s(\phi) s(\theta) m_y + c(\phi) s(\theta) m_z \\ c(\phi) m_y - s(\phi) m_z \\ -s(\theta) m_x + s(\phi) c(\theta) m_y + c(\phi) c(\theta) m_z \end{pmatrix} = \begin{pmatrix} c(\psi) c(\gamma) \\ -s(\psi) c(\gamma) \\ -s(\gamma) \end{pmatrix} \tag{3.9}$$

from the first and second rows of (3.9) we can finally compute the yaw angle:

$$\psi = atan2\left(\frac{c(\phi) m_y + s(\phi) m_z}{c(\theta) m_x + s(\phi) s(\theta) m_y + c(\phi) s(\theta) m_z}\right) \tag{3.10}$$

3.1.3 Gyroscope

Triaxial gyroscopes measure the angular velocity vector $\boldsymbol{\omega}$ of the body expressed in the body-fixed frame. In order to get the orientation from the gyroscope measurements, the signal has to be integrated over time starting from some initial conditions.

If $\triangle t$ is the sampling time of the measuring device, one might assume that the variation of the Euler angles can be iteratively computed as the current value of the angles plus the angular velocity times the sampling time:

$$\begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix}_{k+1} = \begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix}_{k} + \triangle t \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}_{k} \tag{3.11}$$

as ω_x, ω_y and ω_z represent the rate of change of the angular position respect to x, y and z axes in the body frame. Nevertheless, the Euler angles define a rotation as a sequence of three rotations, while $\boldsymbol{\omega}$ describes the effect of three rotations simultaneously.

Given that rotations in general are not commutative, the result would be different if instead of chosing the Euler angles as a sequence of rotations about $x \to y \to z$, we chose the sequence $y \to z \to x$ or $x \to z \to y$.

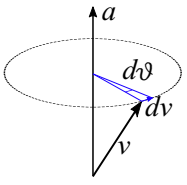

Figure 7: Infinitesimal rotation of a vector

An infinitesimal rotation of $d\theta$ of a vector $\boldsymbol{v}$ about an unitary vector $\boldsymbol{a}$ in $\mathbb{R}^3$can be expressed as [1]:

$$d\boldsymbol{u} = (\boldsymbol{a} \times \boldsymbol{u})\, d\theta \tag{3.12}$$

We can express the cross product as a skew symmetric matrix:

$$\boldsymbol{a}\times \equiv [\boldsymbol{a}\times] = \begin{pmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{pmatrix} \tag{3.13}$$

Rearranging:

$$\frac{d\boldsymbol{u}}{d\theta} = [\boldsymbol{a}\times]\, \boldsymbol{u} \tag{3.14}$$

Integrating both sides we get:

$$\boldsymbol{u}\,(\theta) = e^{\theta[\boldsymbol{a}\times]}\boldsymbol{u}\,(0) \tag{3.15}$$

The skew symmetric matrix for each basis vector is called the infinitesimal generator of rotations for that vector:

$$L_x = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \; L_y = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \; L_z = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{3.16}$$

and any skew symmetric matrix can be expressed as a linear combination of (3.16). It's said that the matrices form a basis of the 3D rotation (SO3) Lie group.

To prove the non commutativity of rotations, two sucesive rotations can be performed:

$$\begin{aligned} e^{\theta_1 A}e^{\theta_2 B} &= \left(1 + \theta_1 A + \tfrac{\theta_1^2 A^2}{2!} + ...\right)\left(1 + \theta_2 B + \tfrac{\theta_2^2 B^2}{2!} + ...\right) \\ &= 1 + \theta_1 A + \tfrac{\theta_1^2 A^2}{2!} + \theta_2 B + \tfrac{\theta_2^2 B^2}{2!} + \theta_1 A\theta_2 B + \theta_2 B\theta_1 A + ... \\ &= e^{\theta_2 B}e^{\theta_1 A}e^{\left(\theta_1\theta_2\{A,B\}+O\left(\theta^3\right)\right)} \end{aligned} \tag{3.17}$$

where $\{A, B\}$ is the anticommutator of A and B. As seen above, there is an exponential term that makes the result differ when the rotations are performed in different order, with very few exceptions for which that exponential is equal to the identity matrix.

For infinitesimal rotations ($\theta \ll 1$), that exponential goes fast to zero and the rotations are indeed commutative.

If the sampling time is short enough so that the rotation angle for each iteration is very small, then (3.11) can be a good approximation.

The time evolution of a DCM is given by:

$$\frac{dR^{BG}}{dt} = R^{BG}\,[\boldsymbol{\omega}\,(t)\,\times] \tag{3.18}$$

or in a discrete iterative form:

$$R^{BG}(t+\triangle t) = R^{BG}(t)\left(\mathbb{I}^{3\times 3} + \triangle t\left[\boldsymbol{\omega}(t)\times\right]\right) \tag{3.19}$$

From the elements of the DCM, comparing (2.4) with (2.2), the Euler angles can be obtained.

In the case of quaternions, the evolution is given by:

$$\dot{\boldsymbol{q}} = \boldsymbol{q} \circ (0, \boldsymbol{\omega}(t)) \tag{3.20}$$

3.2 Sensor fusion

Let us imagine that at time t_0 we measure a given variable x with some sensor, getting as a result a value of x_0 with an error or uncertainty that we can describe by a gaussian probability density function with mean at $\hat{x}_0$ and variance σ_0^2. Then later, at time t_1, we make another measurement with a different sensor, getting a value of $\hat{x}_1$ with assocated variance σ_1^2.

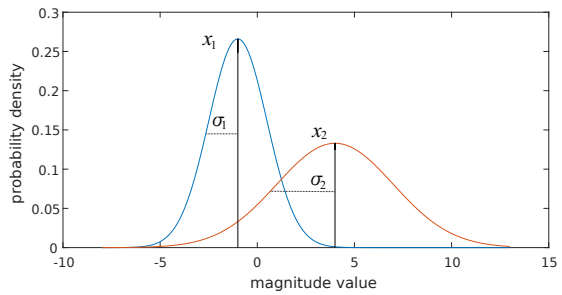

Figure 8: Density probability distributions associated with measurements $\hat{x}_1$ and $\hat{x}_2$

Then we can make use of Bayesian statistics to make an inference about the real value of x.

The Bayes theorem for contiuous distributions reads [20]:

$$f(x \mid \hat{x}_0, \hat{x}_1) = \frac{f\left(x \mid \mu_0, \sigma_0^2\right) f\left(x \mid \mu_1, \sigma_1^2\right)}{\int_{-\infty}^{\infty} f\left(x \mid \mu_0, \sigma_0^2\right) f\left(x \mid \mu_1, \sigma_1^2\right)} \tag{3.21}$$

That is, the *a posteriori* probability distribution is given by the normalized product of the two distributions.

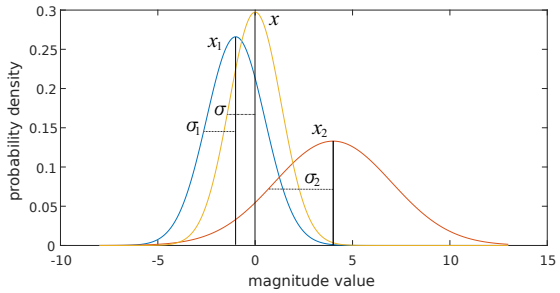

Figure 9: A posteriori distribution obtained from the two measurements

This new probability distribution has the following parameters [22]:

$$\mu = \frac{\sigma_0^2}{\sigma_0^2 + \sigma_1^2}\mu_0 + \frac{\sigma_1^2}{\sigma_0^2 + \sigma_1^2}\mu_1 \tag{3.22}$$

$$\sigma^2 = \frac{\sigma_0^2 \sigma_1^2}{\sigma_0^2 + \sigma_1^2} \tag{3.23}$$

where $\sigma^2 \leq \sigma_0^2, \sigma_0^2$. This means that from two different sources of information with two given uncertainties we can get a measure that is more reliable, more close to the real value (statistically at least) than the last two.

Another way to obtain this result is generating an optimal estimator.

If we want to estimate the value of x, we can do it as a linear combination of $\hat{x}_0$ and $\hat{x}_1$[21]:

$$x = \alpha\hat{x}_0 + \beta\hat{x}_1 = \alpha\hat{x}_0 + (1-\alpha)\,\hat{x}_1 \tag{3.24}$$

where $\alpha + \beta = 1$ so if both measures are equal, then x is also equal to them.

We define the optimal estimate as the one that presents the minimum variance:

$$\sigma^2(\alpha) = \alpha^2\sigma_0^2 + (1-\alpha)^2\,\sigma_1^2 \tag{3.25}$$

$$\frac{d\sigma^2(\alpha)}{d\alpha} = 0 \;\rightarrow\; \alpha = \frac{\sigma_0^2}{\sigma_0^2 + \sigma_1^2} \tag{3.26}$$

therefore

$$\sigma^2 = \frac{\sigma_0^2 \sigma_1^2}{\sigma_0^2 + \sigma_1^2} \tag{3.27}$$

the same result as (3.23).

This linear optimal estimate is the basis of optimal sensor fusion method called Kalman filter.

3.2.1 Kalman filter

The Kalman filter is an iterative method for finding the optimal estimate for a set of magnitudes from certain system, making use of two data:

- A priori estimate $\boldsymbol{x}_k$: is the obtained value for the state vector from the previous estimate and a model that describes it's evolution over time:

$$\boldsymbol{x}_k = A\boldsymbol{x}_{k-1} + B\boldsymbol{u}_{k-1} + \boldsymbol{w}_{k-1} \tag{3.28}$$

where matrices A and B, called state and control matrices, predict the change of the state variables in sucesive iterations (for a further explanation of the state space representation, read section 4.2).

$\boldsymbol{w}_k$ is the process noise, and it represents the deviation of the state from the model prediction (this can be due to random noises and internal/external factors that keep the system from behaving as it should theoretically). This process noise is described as a multidimensional gaussian probability distribution with a covariance matrix Q, called process noise covariance [22]:

$$Q = \begin{pmatrix} \sigma_{1,1}^2 & \sigma_{1,2}^2 & \cdots & \sigma_{1,n}^2 \\ \sigma_{21}^2 & \sigma_{2,2}^2 & \cdots & \sigma_{2,n}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n,1}^2 & \sigma_{n,2}^2 & \cdots & \sigma_{n,n}^2 \end{pmatrix} \tag{3.29}$$

where

$$\sigma_{x,y}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})\,(y_i - \bar{y}) \tag{3.30}$$

The diagonal terms of the covariance matrix are simply the variances for each state variable, while the off diagonal terms show the correlation between the different errors, which in practice are usually set equal to zero for simplicity. Essentialy, if the terms in Q are large, it means that the model is not doing well predicting the current state from the previous one.

- A measurement $\boldsymbol{y}_k$: is obtained by measuring the current state of the system:

$$\boldsymbol{y}_k = H\boldsymbol{x}_k + \boldsymbol{v}_k \tag{3.31}$$

where H is called observation or output matrix, and it determines the portion of the state that can be measured. $\boldsymbol{v}_k$ is known as measurement noise, and it, as it's name says, describes the error of the measurement for each variable. The asociated covariance matrix is called measurement noise covariance, R.

The algorithm is then divided into two parts or phases:

1. Prediction phase: the previous state is propagated (3.28) to get an a priori estimate $\left(\boldsymbol{x}_k^-\right)$ of the actual state:

$$\boldsymbol{x}_k^- = A\boldsymbol{x}_{k-1} + B\boldsymbol{u}_{k-1} \tag{3.32}$$

as well as the estimate error covariance matrix, which represents the prediction of the error associated to the predicted state $\boldsymbol{x}_k^-$:

$$P_k^- = E\left[\boldsymbol{e}_k \boldsymbol{e}_k^T\right] = AP_{k-1}A^T + Q \tag{3.33}$$

1. Correction phase: a new measurement of the state is made (3.31), and the a posteriori predicted state is computed as a linear combination of both the predicted and the measured values:

$$\boldsymbol{x}_k = \boldsymbol{x}_k^- + K_k\left(\boldsymbol{y}_k - H\boldsymbol{x}_k^-\right) \tag{3.34}$$

which bears resemblance to the monodimensional case (3.24). Here, the weighting parameter is given by the matrix K, known as the Kalman gain. As in (3.24), it represents the gain that minimizes the error covariance matrix:

$$K_k = \frac{P_k^- H^T}{HP_k^- H^T + R} \tag{3.35}$$

If the measurement error covariance R is small, the value of the a posteriori predicted state stays closer to the measurement. If P_k^- is smaller, then the a posteriori predicted state approaches more to the a priori prediction.

Finally, the a posteriori estimate error covariance matrix P is computed:

$$P_k = (I - K_k H)\, P_k^- \tag{3.36}$$

The next iteration, the a posteriori predicted state $\boldsymbol{x}_k$ becomes the new previous state $\boldsymbol{x}_{k-1}$, and matrix P_k becomes the new previous P_{k-1}.

3.2.2 Complementary filter

A much simpler version of a linear estimator (3.24) is the complementary filter.

When an angle is measured by integrating the gyroscope signal over time (3.19), the integration of the measurement errors along with the error intrinsic to the discretizacion in digital signals produces a bias that grows over time and causes the measure to deviate from the real value in the long term. On the other hand, the accelerometer provides a unbiased measurement of the orientation that is accurate over long periods of time, but the signal is submited to a large amount of noise. We can say then that the accelerometer signal is more valuable at low frequencies and the gyroscope data has more desirable properties at low frequencies.

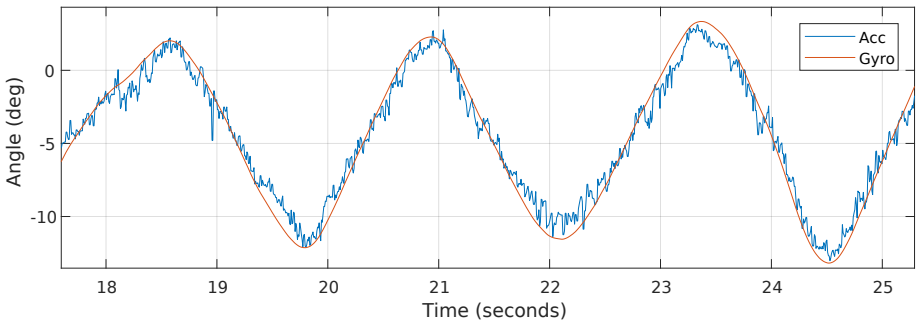

Figure 10: Example of gyroscope and accelerometer signals in angle measurement

The complementary filter exploits this frequency features by adding the pass-filtered data from each sensor: a low pass filter reduces the high frequency noise from the accelerometer, and a high pass filter minimizes the effect of the bias of the gyroscope measurement, so from the addition of both filtered data we can a measurement that is useful and reliable in the entire frequency spectrum. The complementary high and low pass filters in the Laplace domain are given by [23]:

$$G_{low}(s) = \frac{\alpha}{1+\alpha s} \quad G_{high}(s) = 1 - G_{low}(s) \tag{3.37}$$

The measured angle in the Laplace domain $\theta(s)$ is computed as the sum of the filtered measurements from the accelerometer, $\theta_{acc}(s)$, and from the gyroscope, $\theta_{gyro}(s)$:

$$\theta(s) = G_{low}(s)\,\theta_{gyro}(s) + (1 - G_{low}(s))\,\theta_{acc}(s) \tag{3.38}$$

performing the inverse Laplace transform to express the function in time domain and discretizing [6]:

$$\theta_k = \alpha\theta_{gyro,k} + (1-\alpha)\,\theta_{acc,k} = \alpha\,(\theta_{k-1} + \triangle t\omega_{k-1}) + (1-\alpha)\,\theta_{acc,k} \tag{3.39}$$

with

$$\alpha = \frac{\tau}{\tau + \triangle t} \tag{3.40}$$

where τ is the time constant of the filter.

4 Control theory

The control theory, as it's name indicates, is based on the study of the control of the dynamic state of a given system (also called plant) through the action of elements which are capable of modifying it's behavior, as well as the analysis of the stability and the reach of such control. These elements are called actuators, and they're operated by an element called controller.

In order to control a system, an aim state or setpoint is defined, that is, a specific set of values for the state variables that we want our system to reach.

From the difference between the current state and the aim state an error is defined, and from this error signal and through the use of a particular control law, the controller decides how the actuators will operate.

This kind of control in which there is a feedback from the state is called closed loop control. In the case of attitude control on Earth's surface, the feedback will come from the signal of the different inertial measurement sensors attached to the body, which provide a constant uptate of the state of the system.

4.1 Linear time-invariant systems

Although the vast majority of physical systems present inherent nonlinear characteristics, they can be approximately described by a linear approximation in the neighbourhood of an equilibrium point by performing a first order Taylor expansion of the dynamic functions. The reason behind this linearization is the amount of vastly simpler methods for studying control and it's stability compared to the nonlinear case.

In particular, for linear systems the total reaction $y(t)$ to a set of inputs/actions $f_i(t_i)$ (for example, a voltage step function applied to a circuit, or a certain amount of impulse generated by a thruster in a spacecraft) is the sum of the individual responses for every input. This property is known as superposition.

The impulse response of a system, $h(t)$, is defined as the output of the system corresponding to a delta function input $f(t) = \delta(t)$ for zero-boundary conditions. Because of linearity, the response of the system is [14]:

$$y(t) = \int_0^t f(\tau)h(t-\tau)d\tau \tag{4.1}$$

This equation is known as the convolution integral, also represented by the convolution operator $y(t) = (u * h)(t)$. The idea behind this comes from expressing any input $f(t)$ as a set of n scaled and time-shifted delta impulses, and bringing the expression to the continuum limit ($n \to \infty$).

To visualize this superposition feature, we take as an exaple a simple mass-spring-damper model with an input force, described by the linear differential equation $m\ddot{y}(t) + b\dot{y}(t) + ky(t) = u(t)$. The impulse response to a delta function at $t = 0$, $f(t) = c\delta(t)$, is in this case equal to a instant change in the momentum of the system $m\dot{y}(0^+) = c$. The solution to the differential equation reads:

$$h(t) = e^{\frac{-b}{2m}t} sin(\omega' t) \quad \omega' = \sqrt{\frac{k}{m} - \left(\frac{b}{2m}\right)^2} \tag{4.2}$$

If the system is subjected to an input force $f(t) = sin(10t),\ t \in \left[0, \frac{\pi}{\omega'}\right]$, we can approximate it by a set of n scaled and time-shifted delta impulses. Here, these inpulses could be intuitively seen as "knocks" of a very short duration that change the momentum of the mass instantaneously.

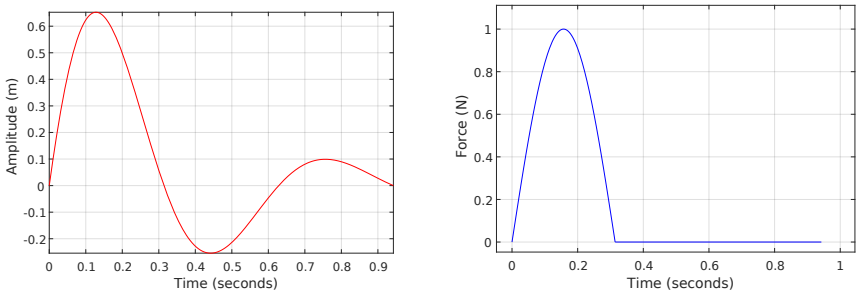

Figure 11: Impulse function $h(t)$ for the mass-spring system and input function $f(t)$

As n becomes larger, the sum of the individual responses becomes closer to the real response.

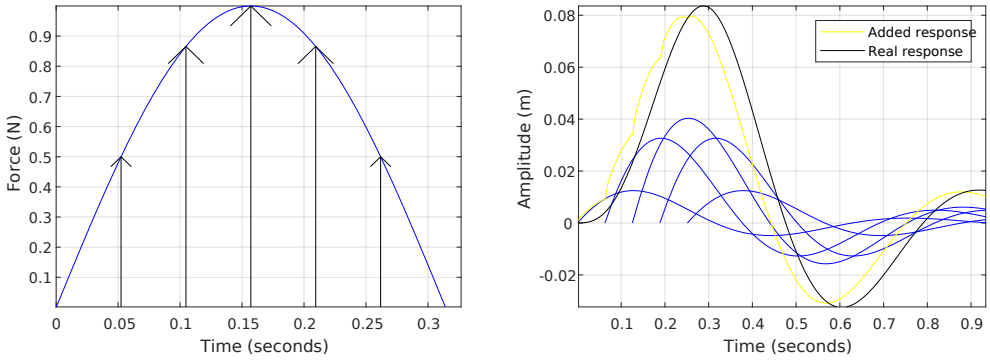

Figure 12: Delta inpulse approximation for $n = 5$ and added response compared to real

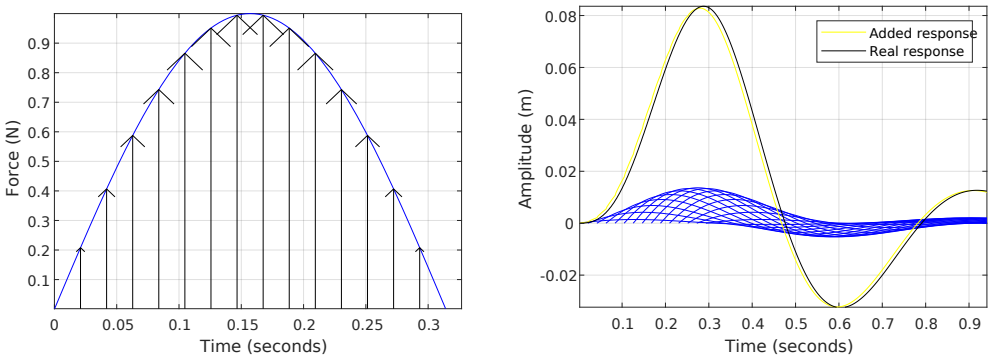

Figure 13: Delta inpulse approximation for $n = 14$ and added response compared to real

When n tends to infinity, the sum of every individual response becomes the convolution of $f(t)$ and $h(t)$.

In this property lies the simplicity and potential of linear time-invariant systems (LTI): the behavior of the system as response to any perturbation or control force is completely determined by a function $h(t)$.

4.2 System representation

4.2.1 Ordinary differential equations

As in the mass-spring example, the physical description of linear systems is generally represented by a set of ordinary differential equations that determine the temporal evolution of the generalized coordinates of the system:

$$x_i(t) = \sum_n a_{n,i}(t) x_i^{(n)} + \sum_n b_{n,i}(t) u_i^{(n)} \tag{4.3}$$

where x_i represents the value of the generalized coordinates that define the state of the system, and u_i represent the variables that define an external action or force from the actuators. In a complete description, the number of equations is equal to the number of degrees of freedom of the system.

4.2.2 Laplace domain

A system that responds to a certain signal or input can be seen as a black box that transforms in a specific way the incoming input into another output signal. In case of a LTI system, this transformation is completely described by it's impulse function $h(t)$.

The Laplace transform of $h(t)$ is called transfer function, $H(s)$, and it represents the relation between output (Y) and input (U) in the Laplace domain. In it's more general form [13]:

$$Y(s) = H(s)U(s) \tag{4.4}$$

and as in the case of $h(t)$, it completely defines the behavior of the system.

The advantage of working in the Laplace domain is that the response of a number of connected interacting systems to an initial signal is computed as simply the product of the individual transfer functions times the signal, whereas in time domain this response comes from the convolution (4.1), which can be more arduous to perform.

The transfer function representation also allows us to obtain a large number of characteristics of the system's dynamics in a relatively easy way.

The transfer function of a system is usually denoted with capital letters.

4.2.3 Block diagrams

In engineering fields, another typical way of describing a system is through block diagrams, which visually display the different parts of the system and their connections, representing them by their particular transfer functions. (here the term signal may not strictly refer to an electric signal, but to any output of an element that generates a reaction on another)

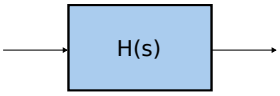

Figure 14: System component represented by a block with associated transfer function $H(s)$

The case shown below is the most general case of closed loop system: a controller connected to a plant, and a feedback signal that is substracted to a reference signal to determine the error that is then sent into the controller.

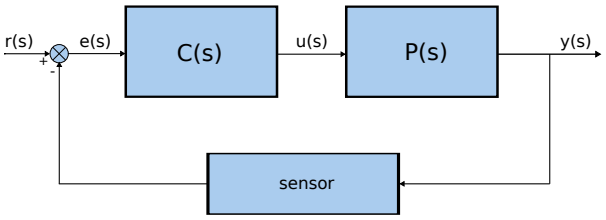

Figure 15: Block diagram for closed loop system

The transfer function for a closed loop system can be obtained by noticing the following relations:

$$\begin{aligned} E(s) &= R(s) - Y(s) \\ Y(s) &= P(s)C(s)E(s) \end{aligned} \tag{4.5}$$

Subtituing the first equation into the second we get the closed loop transfer function:

$$H_{cl}(s) = \frac{Y(s)}{R(s)} = \frac{C(s)P(s)}{1 + C(s)P(s)} \tag{4.6}$$

4.2.4 State space representation

The evolution of the state variables can be expressed in form of matrix equations [11]:

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= A\boldsymbol{x}(t) + B\boldsymbol{u}(t) \\ \boldsymbol{y}(t) &= C\boldsymbol{x}(t) + D\boldsymbol{u}(t) \end{aligned} \tag{4.7}$$

where $\boldsymbol{x}(t)$ is the state vector, which comprises the variables that define the state of the system at a certain time t, $\boldsymbol{y}(t)$ is the output vector, and $\boldsymbol{u}(t)$ is the input vector from the controller signal.

The matrix A is called state matrix, and it determines how the current state affects the variation of the state variables, and B is the control matrix, which describes how the input affects these variables. C and D, known as output and feed-forward matrices, define the output vector in terms of the current state. Intuitively, C determines the portion of the system variables that can be observed.

For closed loop systems, and in particular when the state feedback is constant, $\boldsymbol{u}$ can be expressed as a function of $\boldsymbol{x}$. In general we have:

$$\boldsymbol{u}(t) = K\boldsymbol{x}(t) \tag{4.8}$$

where K is known as closed loop feedback matrix or gain matrix.

To change from state space representation to Laplace domain representation, we perform the Laplace transform to the state equations (4.7):

$$\begin{gathered} s\boldsymbol{x}(s) = A\boldsymbol{x}(s) + B\boldsymbol{u}(s) \;\rightarrow\; (sI - A)\boldsymbol{x}(s) = B\boldsymbol{u}(s) \\ \boldsymbol{y}(s) = C\boldsymbol{x}(s) + D\boldsymbol{u}(s) \end{gathered} \tag{4.9}$$

then multiplying both sides of the first by $(sI - A)^{-1}$ and subtituing in the second:

$$\boldsymbol{y}(s) = \left[C\left(sI - A\right)^{-1} B + D\right] \boldsymbol{u}(s) \tag{4.10}$$

where $H(s) = \left[C\left(sI - A\right)^{-1} B + D\right]$ is equal to the transfer function of the system.

4.3 Control analysis

The solution to the set of differential equations that appear in the State Space description (4.8) can be divided in two separate terms: an homogeneous solution $\boldsymbol{x}_h$ for zero input that describes the response of the system to some initial conditions $\boldsymbol{x}_0$, and a particular solution $\boldsymbol{x}_p$ that corresponds with nonzero input and zero boundary conditions.

The solution to the homogeneous equation $\boldsymbol{x}'(t) = A\boldsymbol{x}(t)$ has an exponential form:

$$\boldsymbol{x}_h(t) = e^{At}\boldsymbol{x}(0) \tag{4.11}$$

where the square matrix exponential is computed as the usual power series expansion:

$$e^{At} = \sum_{n=0}^{\infty} \frac{(At)^n}{n!} = I + At + \frac{A^2t^2}{2!} + \frac{A^3t^3}{3!} + ... \tag{4.12}$$

The matrix $\Phi = e^{At}$ is called the state transition matrix of the system. It presents the following properties [24]:

- $\Phi(0) = I$
- $\Phi(-t) = \Phi^{-1}(t)$
- $\Phi(t_1)\Phi(t_2) = \Phi(t_1 + t_2)$

The particular solution is obtained by multiplying both sides of the first equation in (4.7) by e^{-At} and integrating:

$$\int_0^t \left[e^{-At}\boldsymbol{x}'(t) - e^{-At}A\boldsymbol{x}(t)\right] = \int_0^t \left[\frac{d}{dt}\left(e^{-At}\boldsymbol{x}(t)\right)\right] = \int_0^t e^{-At}B\boldsymbol{u}(t) \tag{4.13}$$

rearranging the terms:

$$\boldsymbol{x}(t) = e^{At}\boldsymbol{x}(0) + \int_0^t e^{A(t-\tau)}B\boldsymbol{u}(t) \tag{4.14}$$

where the first term on the right is the homogeneous solution $\boldsymbol{x}_h(t)$, also called *zero input response*, and the second is the convolution integral, also known as the *zero state response* of the system.

4.3.1 Stability of linear systems

In general, a linear system is said to be stable if the state vector doesn't diverge as time goes to infinity. In particular, a system is asymptotically stable if the state variables tend to 0 [22] :

$$\lim_{t\to\infty} x_i(t) = 0 \tag{4.15}$$

and marginally stable if the state remains bounded:

$$\sup_{t\geq 0} \|x_i(t)\| < k_i \tag{4.16}$$

Given a transfer function

$$H(s) = \frac{\sum_{i=0}^{n} b_n s^n}{\sum_{i=0}^{m} a_m s^m} \tag{4.17}$$

the function can be expanded into a sum of partial fractions:

$$H(s) = \sum_{i=0}^{n} \frac{r_i}{s - p_i} \tag{4.18}$$

where p_i and r_i are called the poles and the residues of the transfer function respectively. By performing the inverse Laplace transform we get the system impulse response:

$$\mathcal{L}^{-1}(H(s)) = h(t) = \sum_{i=0}^{n} r_i e^{p_i t} \tag{4.19}$$

We can see that if the real part of a pole is positive, then the response of the system diverges to infinity with time. This fact implies the following property: the system is asypmtotically stable if and only if all poles of the transfer function have a negative real part. If the real part of a pole is zero, then the system is marginally stable.

From (4.10) it can be proved that the poles of the transfer function and the eigenvalues of the state matrix A coincide, so the same criterion can be applied for the real part of the eigenvalues.

4.3.2 Observability

A system is said to be *observable* if the system state $\boldsymbol{x}(t)$ can be known from the input $\boldsymbol{u}(t)$ and output $\boldsymbol{y}(t)$ vectors at a given time t and the initial state $\boldsymbol{x}(t_0)$.

A system is observable if and only if the observability matrix $O\,(A, C)$ has rank n, equal to the dimension of A [20]:

$$O = \begin{pmatrix} C & CA & CA^2 & \cdots & CA^{n-1} \end{pmatrix}^T \tag{4.20}$$

This ensures that we can obtain an independent system of equations that determines each value of $\boldsymbol{x}(t)$.

4.3.3 Controllability

A system is said to be *controllable* if for any given time t, and any given initial and final states $\boldsymbol{x}(t_0)$ and $\boldsymbol{x}(t)$, there is some input $\boldsymbol{u}(t)$ that can take the state from $\boldsymbol{x}(t_0)$ to $\boldsymbol{x}(t)$.

A sufficient condition for controllability is that the controllability matrix $Q\,(A, B)$ has rank n:

$$Q = \begin{pmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{pmatrix}^T \tag{4.21}$$

4.3.4 Inverted pendulum

The inverted pendulum is a very useful system when it comes to studying attitude control and control systems in general. On one hand, it represents the archetypal model of a system that is subjected to forces or torques that tend to unstabilize it (balancing problem). On the other hand, because it's an observable system, the stability can be achieved by using a wide set of possible control approaches, which will be later discussed in more detail.

The simplest version of an inverted pendulum consists of a body attached to a pivot, with a constant distance between the body's center of mass and the pivot. It's (called) inverted because the center of mass is above the rotation point of the system. The system in this first approximation is assumed to be frictionless.

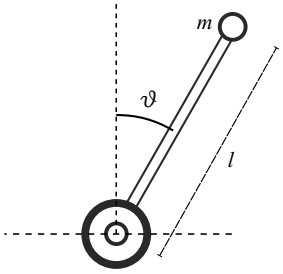

Figure 16: Inverted pendulum

We also assume that the center of gravity coincides with the center of the mass.

The torque produced by the gravity is $\tau = mlgsin\theta$, with the associated differential equation $\ddot{\theta} + \frac{g}{l}sin\theta = 0$. To linearize the equation, we perform the first order Taylor expansion of the sine function around the top equilibrium point

$$\ddot{\theta} - \frac{g}{l}\theta = 0 \tag{4.22}$$

Expressed in state space representation and taking θ and $\dot{\theta}$ as state variables:

$$\begin{gathered} \frac{d\boldsymbol{x}}{dt} = \begin{pmatrix} \dot{\theta} \\ \ddot{\theta} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ \frac{g}{l} & 0 \end{pmatrix} \begin{pmatrix} \theta \\ \dot{\theta} \end{pmatrix} \\ y = \begin{pmatrix} 1 & 0 \end{pmatrix} \boldsymbol{x} \end{gathered} \tag{4.23}$$

The eigenvalues of the state matrix are $\lambda_1 = \frac{\sqrt{g}}{l}$ and $\lambda_2 = -\frac{\sqrt{g}}{l}$, and therefore the point is unstable and the body tends to fall when it's at the top, as expected.

In order to stabilize the system, we define as error the difference between the current angle of the pendulum and the top position, in our case equal to θ. The goal of this control will then be to bring the state as close as possible to the top equilibrium point $\theta = \dot{\theta} = 0$. First, the functional dependence between an external torque τ_{ext} that will control the pendulum and the error has to be set, that is, a control law has to be defined.

Given that the gravitational torque increases with θ, we can assume a proportional dependence with θ as a first ansatz [19]:

$$\ddot{\theta} - \frac{g}{l}\theta = \tau_{ext} = \alpha\theta \tag{4.24}$$

In the state space representation, this feedback can be expressed as:

$$u(t) = K\boldsymbol{x}(t) = \begin{pmatrix} \alpha & 0 \end{pmatrix} \boldsymbol{x}(t) \tag{4.25}$$

The state space equations then read:

$$\begin{gathered} \dot{\boldsymbol{x}} = \begin{pmatrix} \dot{\theta} \\ \ddot{\theta} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ \frac{g}{l} & 0 \end{pmatrix} \begin{pmatrix} \theta \\ \dot{\theta} \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} \alpha & 0 \end{pmatrix} \begin{pmatrix} \theta \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ \frac{g}{l} + \alpha & 0 \end{pmatrix} \begin{pmatrix} \theta \\ \dot{\theta} \end{pmatrix} \\ \boldsymbol{y} = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} \theta \\ \dot{\theta} \end{pmatrix} \end{gathered} \tag{4.26}$$

For constant gain closed loop, the state matrix becomes $A_{cl} = A + BK$, and the system equation can be treated as for the open loop control case ,$\dot{\boldsymbol{x}} = A_{cl}\boldsymbol{x}$.

In this case, the eigenvalues of the state matrix can either be a positive-negative pair or a pure imaginary pair. In the first case the system response diverges, and in the second the imaginary pair creates a constant oscillatory response (4.19). Therefore the control law (4.25) is not useful.

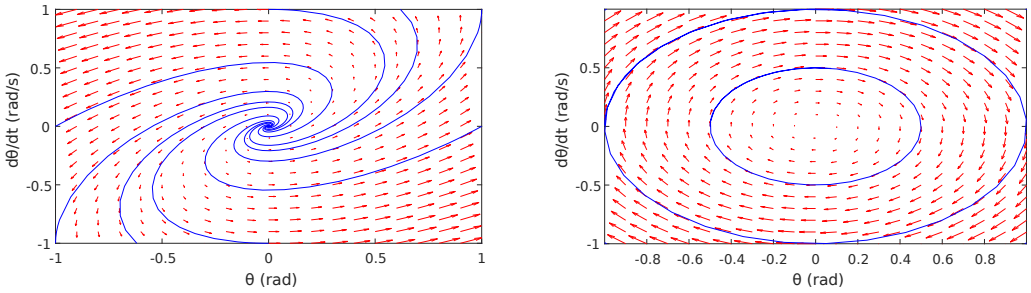

Figure 17: Phase portraits of the system for divergent and constant responses

This divergence of the dynamic variables corresponds to the linearized equations; in the real inverted pendulum system, the divergence is translated into a proggresive shift away from the top unstable equilibrium position towards the lower stable point.

As a second attempt, we can introduce in the feedback a negative dependence with $\dot{\theta}$:

$$\ddot{\theta} - \frac{g}{l}\theta = \tau_{ext} = \alpha\theta - \beta\dot{\theta} \tag{4.27}$$

The characteristic equation then reads:

$$\lambda(\beta - \lambda) - \frac{g}{l} + \alpha = 0 \tag{4.28}$$

This second order equation can return complex values with positive real part, and the system can therefore be stabilized.

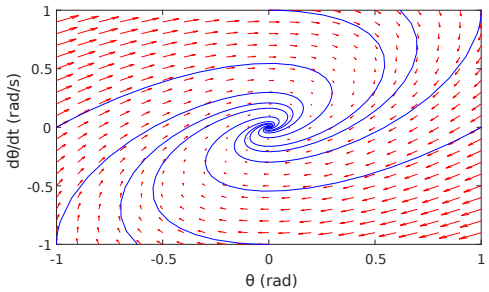

Figure 18: Phase portrait for convergent response

As we see, for certain systems a proportional control (that is, the gain K is proportional to the error) is not enough. In the oscillatory situation the torque produced by the actuator makes the pendulum pass the equilibrium point at a high velocity, producing an overshoot that repeats towards both sides throughout the motion: the inverted pendulum becomes a simple pendulum, but pointing upwards instead of downwards.

By adding a derivative term, the overshoot reduces over time, making the system converge to the equilibrium point, just like a regular pendulum with friction would do.

This example serves as an introduction to one of the most used control approaches in the world of industrial processes and automatization, and in control systems in general: the Proportional-Integral-Derivative control, also known as PID control.

4.4 Controllers

4.4.1 PID controller

A PID controller makes use of three different terms to calculate the control function from the error $e(t)$ [23]:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \tag{4.29}$$

where K_p, K_i and K_d are called proportional, integral and derivative constants respectively.

- The proportional term depends on the actual value of the error: if the value of the controlled variable is far from the setpoint, the action introduced will be large. If they are equal, the proportional term will be zero.
- The derivative term depends on the rate of change of the error, that is, it's influenced by the speed with which the state is getting closer or further from the setpoint. The derivative term has a predictive nature, given that it acts on the error trend,

preventing the system from overshooting. For example, if a car wants to reach a certain position and stop at that point, it will have to begin decelerating before arriving or it will inevitably pass the point at nonzero speed.

- The integral term, as it's name says, is proportional to the error integrated over time. It's useful in the situations in which there is a disturbance that prevents the state to reach the setpoint. This disturbance appears for example in cases where the setpoint monotonically changes over time and the proportional-derivative control is not able to make the system reach a nonzero error. The integral term would then increase until the disturbance is overcome.

To see the effect of the PID constants K_i, we can write a second order homogeneous differential equation in a canonical form [1]:

$$\ddot{x}(t) + 2\zeta\omega_n \dot{x}(t) + \omega_n^2 x(t) = u(t) \tag{4.30}$$

Where ζ is called damping ratio and ω_n is the natural frequency of the undamped system. The transfer function reads:

$$P(s) = \frac{1}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{4.31}$$

If we add a PID control to the system, the closed loop transfer function can is computed as (4.6):

$$H(s) = \frac{\left(K_p + \frac{1}{s}K_i + sK_d\right)\frac{1}{s^2+2\zeta\omega_n s+\omega_n^2}}{1 + \left(K_p + \frac{1}{s}K_i + sK_d\right)\frac{1}{s^2+2\zeta\omega_n s+\omega_n^2}} \tag{4.32}$$

The charasteristic equation for this transfer function is:

$$s\left(s^2 + 2\zeta\omega_n s + \omega_n^2\right) + K_p s + K_i + K_d s = 0 \tag{4.33}$$

rearranging terms:

$$s^3 + (2\zeta\omega_n + K_d)\, s^2 + \left(\omega_n^2 + K_p\right) s + K_i = 0 \tag{4.34}$$

That is, by tuning the three PID constants, we can change the value of the different poles of the system. This allows us to directly modify the dynamic properties of the system and make it stable, or even impose some characteristics on it's movement, given that the poles of the system determine it's behavior (4.19).

For example, in the case of the inverted pendulum (4.22) with a PD control, the system approaches the setpoint in different ways depending on the value K_p and K_d.

The transfer function of the pendulum in that case is (4.6) (4.10):

$$H(s) = \frac{(K_p + sK_d)\frac{s}{s^2+\frac{l}{g}}}{1 + (K_p + sK_d)\frac{s}{s^2+\frac{l}{g}}} = \frac{s\,(K_p + sK_d)}{s^2\,(1 + K_d) + sK_p + \frac{l}{g}} \tag{4.35}$$

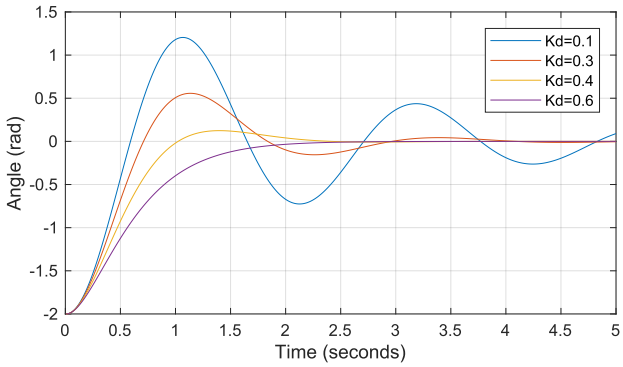

Figure 19: Evolution of angle for fixed K_p and different values of K_d

As we can see, the controller can be configurated in many ways. The optimallity of the control depends on various factors; on the one hand, we want the system to converge to the setpoint as fast as possible and, on the other, we may want the system to overshoot as little as possible.

Comparing (4.35) with (4.31):

$$\zeta = \frac{K_p\sqrt{g}}{\sqrt{l\,(1+K_d)}} \quad \omega_n^2 = \frac{l}{g\,(1+K_d)} \tag{4.36}$$

Theoretically, by tuning K_d and K_p we can make the system reach the wanted state virtually as fast as we want, and make the overshoot as low as we want.

But in practice real systems and actuators present technical and physical limitations that maight not be overcome.

Another thing of great importance and that needs to be addresed is the energy consumption of the actuator in the control process. The PID control approach doesn't directly take into account this fact, but more modern techniques do, as we will see in the next section.

4.4.2 LQR controller

As an example, we can imagine a satellite with a thruster that can change the object's angular momentum and make it rotate" around a certain axis $\hat{n}$.

Now we want to change the orientation and turn the satellite by α degrees around $\hat{n}$. To reach this new rotated state, the thruster has to be activated twice: one to accelerate the body so it begins rotating, and another to stop it, so the final state is stationary.

We assume for simplicity that the accelerating and decelerating impulses are instantaneous, one at the beginning and other at the end of the motion. We also assume that the mass of fuel used each time is directly proportional to the angular impulse, that is, the variation in the angular momentum: $m_f = \gamma \triangle L = \gamma I_{\hat{n}} \triangle \omega$.

Then the time that it takes for the system to reach the setpoint is $\frac{\alpha\gamma I_{\hat{n}}}{m_f}$. That means that the fastest we want our control to respond, the more fuel we will have to spend.

In general, if we want our control system to perform better dinamically, it will be at the expense of a higher energy consumption. If we want to go to work, taking an helicopter

might be much faster than walking or driving, but the cost associated with this mean of transportation is considerably higher.

In real systems, the adequacy of the control depends both on the state regulation capability and the magnitude of the control action.

the Linear Quadratic Regulator (LQR) control consists of contructing a cost function that takes into acount the two terms and finding a state feedback control law that minimizes it.

The control and action costs are computed as quadratic forms of the state and input vectors and their associated weighting matrices:

$$\boldsymbol{x}^T(t)\,Q\boldsymbol{x}(t) \qquad \boldsymbol{u}^T(t)\,R\boldsymbol{u}(t) \tag{4.37}$$

where weighting matrices Q and R are real, symmetric and positive definite. By tuning Q and R we are able to either penalize states that lie far from it's setpoint and maximize convergence rate of certain value x_i, or punish the use of an actuator, therefore minimizing certain control action throughout the control process. The cost function is then defined as the integral over time of both quadratic costs [21]:

$$J = \int_0^{\infty} \left(\boldsymbol{x}^T(\tau)\,Q\boldsymbol{x}(\tau) + \boldsymbol{u}^T(\tau)\,R\boldsymbol{u}(\tau)\right) d\tau \tag{4.38}$$

In order to find the optimal control $\boldsymbol{u}^\star$ for a given set of matrices $\{A,\, B,\, Q,\, R\}$ that minimizes J, we start by defining a *value function*:

$$V(x(t),t) = \min_u \left\{ \int_t^{t_f} \left(\boldsymbol{x}^T(\tau)\,Q\boldsymbol{x}(\tau) + \boldsymbol{u}^T(\tau)\,R\boldsymbol{u}(\tau)\right) d\tau \right\} \equiv \min_u \left\{ \int_t^{t_f} g(\boldsymbol{x}(t),\boldsymbol{u}(t),\tau)\, d\tau \right\} \tag{4.39}$$

This function represents the smallest possible value of the cost function associated with bringing the state from $\boldsymbol{x}(t)$ to some final state $\boldsymbol{x}(t_f)$.

Now we make use of the Bellman optimallity principle. The principle states that every segment of an optimal path (here path is refered to a control signal $u(t)$ fed into the system over time) is itself optimal. This simply means that at each time step, the optimal control makes the cost function evolve in a way that it's extremal. This can be expressed as [19]:

$$V(\boldsymbol{x}(t+dt),t+dt) = \min_u \left\{ V(\boldsymbol{x}(t),t) + \int_t^{t+dt} g(\boldsymbol{x}(\tau),\boldsymbol{u}(\tau),\tau)\, d\tau \right\} \tag{4.40}$$

or in an alternative form:

$$\min_u \left\{ \dot{V}(\boldsymbol{x},t) + g(\boldsymbol{x},\boldsymbol{u},t) \right\} = 0 \tag{4.41}$$

This functional equation is called *Hamilton-Jacobi-Bellman* equation, widely used in calculus of variations.

As an ansatz for the value function, we define it as a quadratic form $V(\boldsymbol{x},t) = \boldsymbol{x}^T(t)\,P\boldsymbol{x}(t)$. Substituing into (4.41) and making use of the chain rule (4.7):

$$\min_u \left\{ \boldsymbol{x}^T \dot{P}\boldsymbol{x} + 2\boldsymbol{x}^T P\left(A\boldsymbol{x} + B\boldsymbol{u}\right) + \boldsymbol{x}^T Q\boldsymbol{x} + \boldsymbol{u}^T R\boldsymbol{u} \right\} = 0 \tag{4.42}$$

For LTI systems, $\dot{P} = 0$ due to the time invariance of V. Now differentiating respect to $\boldsymbol{u}$ to find the minimum:

$$2\boldsymbol{u}^T R + \boldsymbol{x}^T PB = 0 \tag{4.43}$$

so

$$\boldsymbol{u} = -R^{-1}B^T P\boldsymbol{x} = K\boldsymbol{x}(t) \tag{4.44}$$

As we see above, the optimal control is given by a state feedback control with a gain matrix $K = -R^{-1}B^T P$. To find the matrix P, we substitute (4.44) into (4.41) again and simplify to get

$$\boldsymbol{x}^T \left[A^T P + PA - PBR^{-1}B^T P + Q\right]\boldsymbol{x} = 0 \tag{4.45}$$

this only holds if:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{4.46}$$

This equation is called the *algebraic Riccati equation*, and it has to be solved in order to find K, usually by using numerical methods.

As we said earlier, there are many feasible ways to control a system. In the LQR problem, this translates into choosing the individual terms of Q and R matrices that suit best our requirements, and this is generally done by trial and error.

As a starting point for the weighting marices, the so called Bryson's rule is usually used: if the approximated largest possible values of the state and control variables are $\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_n$ and $\tilde{u}_1, \tilde{u}_2, ..., \tilde{u}_m$ respectively, then:

$$Q = \begin{pmatrix} \frac{1}{\tilde{x}_1^2} & 0 & \dots & 0 \\ 0 & \frac{1}{\tilde{x}_2^2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\tilde{x}_n^2} \end{pmatrix} \qquad R = \begin{pmatrix} \frac{1}{\tilde{u}_1^2} & 0 & \dots & 0 \\ 0 & \frac{1}{\tilde{u}_2^2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\tilde{u}_m^2} \end{pmatrix} \tag{4.47}$$

that is, the cost for each variable is normalized by it's magnitude, so the control is roughly adjusted to the characteristics of the specific studied system.

5 Inverted pendulum with reaction wheel

A prototype of an inverted pendulum with a reaction wheel integrated was constructed for future attitude measurement and control testing.

The structure consists of a plastic face attached to a static mount by a ball bearing that allows the face to rotate around it's axis.

This face presents two brackets at different heights to hold two low cost inertial measurement units (IMUs) and a central hole to place a motor with a wheel attached to it's axis.

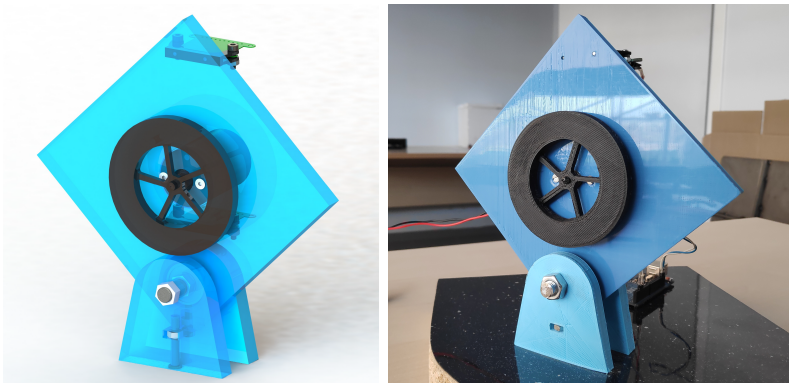

Figure 20: 3D model and finished prototype

The structure was designed in Solidworks and eventually printed using a 3D printer. Each IMU has incorporated a triaxial accelerometer, magnetormeter and gyroscope. The serial communication with the devices as well as the signal processing and algorithms execution is performed by an Arduino UNO microcontroller.

The motor along with the reaction wheel work as the actuator of the system, applying a controlled torque that can modify the orientation of the body.

5.1 Sensor calibration and characterization

Inertial sensors may be subject to different types of measurement errors, not only due to the limitations on the precision with which those measurements are made, but also because of a possible lack (or excess) of sensitivity of the sensor and steady-state related errors among others.

Those two can be modelled as follows:

$$X_{meas} = \Delta_{scal} \left(X_{real} + \Delta_{steady} \right) \tag{5.1}$$

that is, Δ_{scal} represents an error that scales proportional to the measurement magnitude, and Δ_{steady} is the bias that comes out when the real measured magnitude is zero.

The steady-state error can be obtained by measuring the sensor signal while the system remains still.

For the gyroscope, the values measured in the y-axis (face's rotation axis) are grouped around a mean value of $-1.764°/s$.

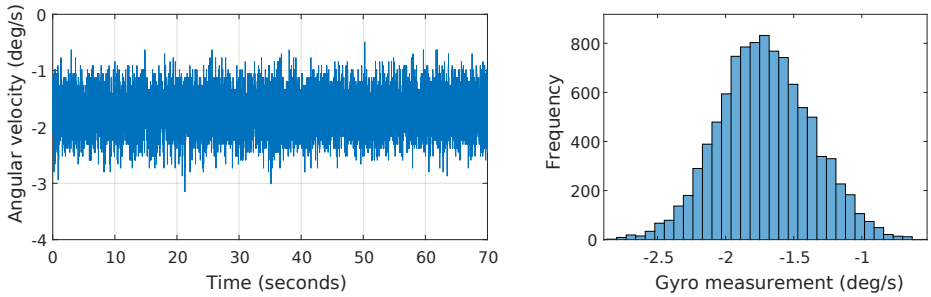

Figure 21: Gyroscope static measurements and value histogram

The gyroscope noise can be well approximated by a normal probability density distribution with variance $\sigma^2_{gyro} = 0.1166$.

To obtain the scaling error, a rotation of 90° is performed on the system and the angle is integrated. The value obtained from the measurement is 91.5°, so $\Delta_{scal} = \frac{91.5}{90} = 1.0167$.x

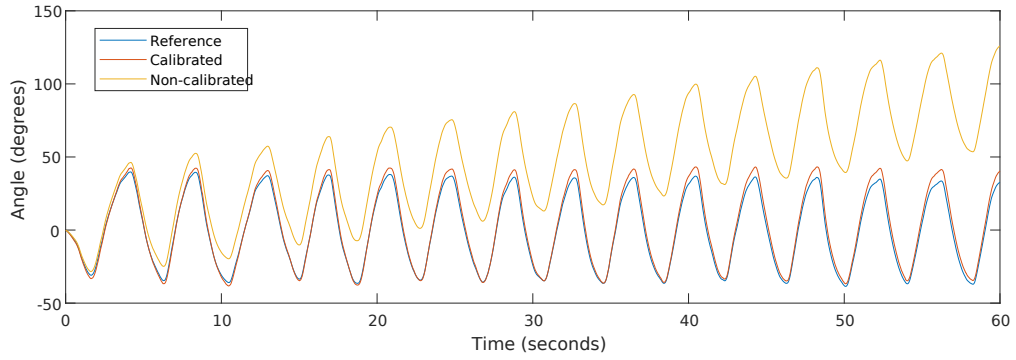

Figure 22: Orientation measurement from gyroscope before and after calibration compared to reference

A quick method for calibrating the accelerometer is to use the orthogonality property of the axes to determine the maximum gravity output value for each individual axis. This is done by slowly rotating the sensor in a way that the measured acceleration in two of the axes becomes zero at some point, in which the third axis displays the absolute value of the gravitational acceleration.

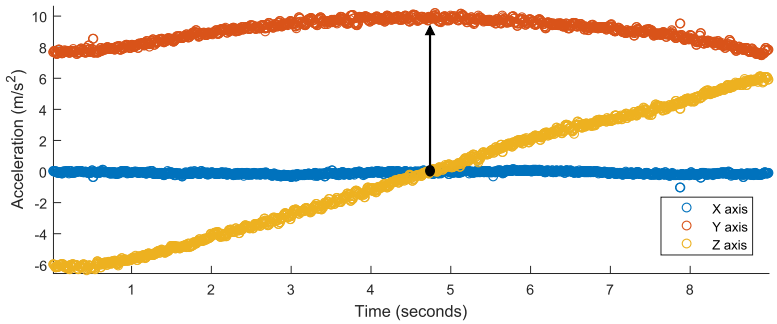

Figure 23: Acceleration measurement during rotation. The Y axis reaches it's maximum value at the junction point

Acc 1	Positive (m/s^2)	Negative (m/s^2)	Zero state error (m/s^2)	Scaling error
X axis	9.60	-10.31	-0.35	1.02
Y axis	10.06	-10.15	-0.05	1.03
Z axis	10.00	-10.37	-0.19	1.04

Table 1: Maximum values of acceleration, zero-state bias and scaling error for accelerometer 1

Acc 2	Positive (m/s^2)	Negative (m/s^2)	Zero state error (m/s^2)	Scaling error
X axis	9.96	-10.25	-0.16	1.04
Y axis	9.80	-10.06	-0.13	1.01
Z axis	9.65	-9.92	-0.14	9.99

Table 2: Maximum values of acceleration, zero-state bias and scaling error for accelerometer 2

The noise asociated with the accelerometer's angle measurement (and with accelerometers in general) is not constant; it presents a dependence with the motion of the system and becomes higher as the magnitude of the accelerations grow.

To model the acceleration noise, a set of different measurements is made. For each measurement, the system is subjected to an oscillatory movement with different frequency for a certain period of time. The higher the frequency is, the faster the system oscillates and therefore the larger the noise becomes.

The noise error is computed as the diference between the angle from the accelerometer measurement (5.5) minus the angle obtained by filtering the data with a complementary filter. Then this noise is associated to the squared average of the gyroscope signal's rate of change, a measure of the sharpness of the motion.

$$\bar{\alpha}^2 = \frac{1}{n^2} \sum_{k=1}^{n} \left(\frac{\omega_{k+1} - \omega_k}{\triangle t} \right)^2 \tag{5.2}$$

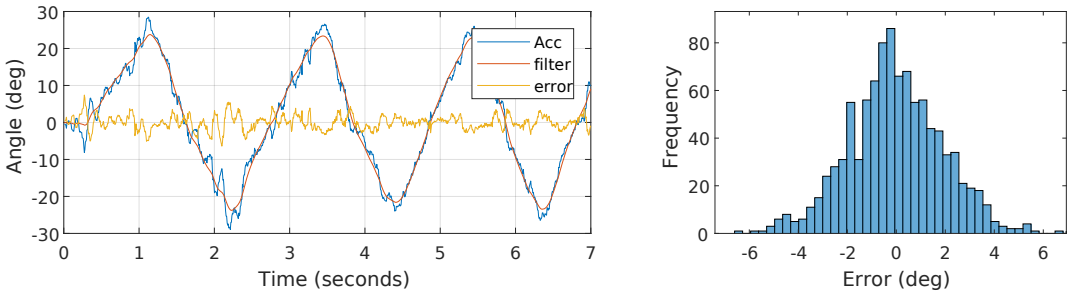

Figure 24: Accelerometer measurement error for smooth motion and error histogram. The yellow line represents the difference between accelerometer and filter data.

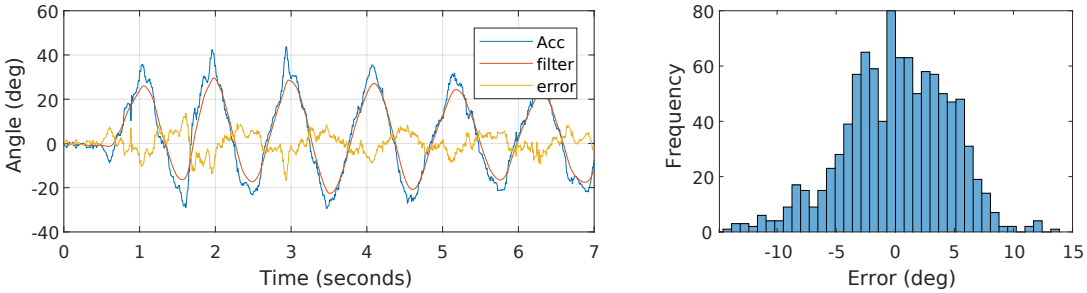

Figure 25: Accelerometer measurement error for sharp motion and error histogram

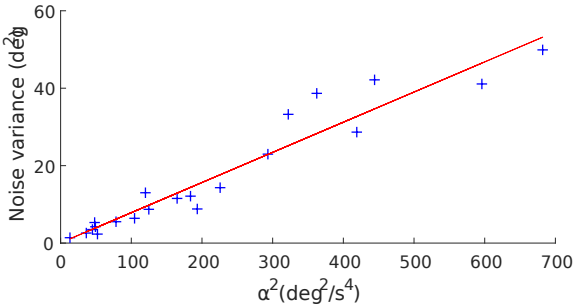

Figure 26: Relation between noise variance and sharpness parameter

The relation between the variance σ_{acc}^2 and $\bar{\alpha}^2$ is given by:

$$\sigma_{acc}^2 \approx 0.078\bar{\alpha}^2 + 0.1442 \tag{5.3}$$

5.2 Orientation measurement

In order to decouple the gravity part of the acceleration measurement from other dynamical terms [23], in this case the centripetal acceleration caused by the rotation motion, the two IMUs are placed at a different distance from the rotating point.

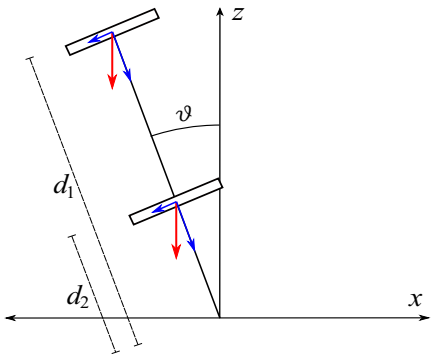

Figure 27: Accelerometers configuration scheme

If the distances are d_1 and d_2, the accelerometer measurement vectors (x and z axes) read:

$$\boldsymbol{a}_n = (a_{n,1}, a_{n,1}) = \left(d_n\ddot{\theta} + gsin\theta, -\dot{\theta}^2 d_n - gcos\theta\right) \tag{5.4}$$

Cancelling the terms of order higher than 2 from the two vectors $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$, we can compute the angle (3.4):

$$\theta = atan2\left(\frac{a_{1,1} - \mu a_{2,1}}{-a_{1,2} - \mu a_{2,2}}\right) \tag{5.5}$$

where $\mu = \frac{d_1}{d_2} = 4.17$.

To make sure that the gravity vectors measured by both accelerometers are indeed parallel, first a measurement of the gravity vector is performed for both sensors. Then, the angle between both is computed and one measurement vector is multiplied by the convenient rotation matrix.

Then, a signal fusion is performed between both accelerometers and a gyroscope to minimize the noise and the bias.

A state vector is defined for the Kalman filter; a common approach is to chose as variables the angle θ and the gyroscope bias b [6]:

$$\boldsymbol{x}_k = \begin{pmatrix} \theta_k \\ b_k \end{pmatrix} \tag{5.6}$$

The last one is not observable, so the output matrix presents only one nonzero term:

$$H = (1, 0) \tag{5.7}$$

During the observation phase, the previous state is propagated using the gyroscope data to obtain an a priori estimate (3.28) of the current state:

$$\boldsymbol{x}_k^- = \begin{pmatrix} \theta_k^- \\ b_k^- \end{pmatrix} = \begin{pmatrix} 1 & -\triangle t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \theta_{k-1} \\ b_{k-1} \end{pmatrix} + \begin{pmatrix} \triangle t \\ 0 \end{pmatrix} \omega_k \tag{5.8}$$

The measurement noise covariance matrix R in this case is given by the variance of the accelerometer's angle measurement noise.

Finally, for the determination of the process noise covariance matrix Q, we can assume that the error associated with the prediction is proportional to the sampling period (the longer the time between the measurements, the more the noise can affect the variables) and to the gyroscope noise variance previously computed:

$$Q = \begin{pmatrix} \triangle t \sigma_{gyro} & 0 \\ 0 & \triangle t \sigma_{gyro} \end{pmatrix} \tag{5.9}$$

The complementary filter is tuned by trial and error. The final value for the gain constant c is 0.97. That means that the 97% of the measurement value comes from the gyroscope data.

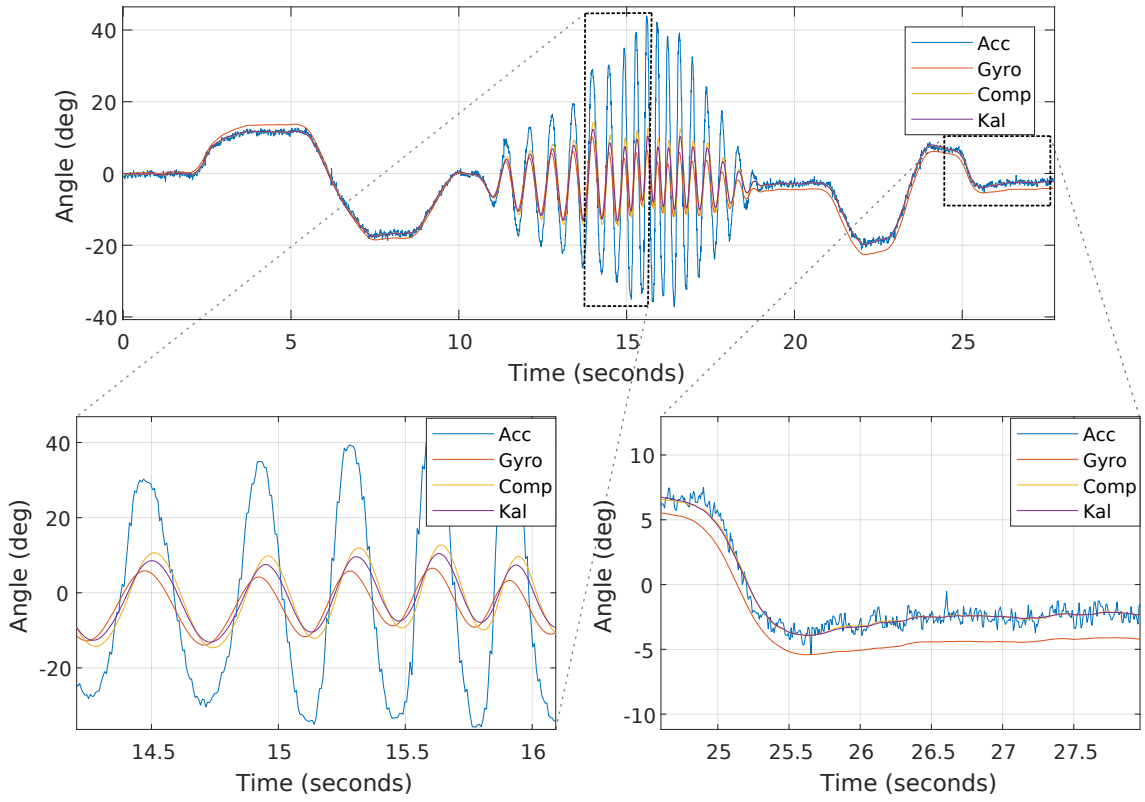

Figure 28: Angle measurement by accelerometer, gyroscope, complementary filter and Kalman filter

5.3 Discussion

From figure (28) we can see that the accelerometer pair presents a small amplitude noise when the body remains fixed, but this noise gets larger when the system moves abruptly, to the point that the angle obtained by it's measurement reaches twice the value from the gyroscope. This is due, as it's shown in (5.3), to the fact that accelerometers become unstable when the accelerations vary rapidly in time. Nevertheless, the two parallel accelerometers system is fully able to decouple the signal correspondant to the gravity vector and the rest of accelerations.

The effect of the bias in the gyroscope measurement is also noticeable. After 25 seconds, the difference between the real angle and the biased is of 2.5° even when the gyroscope has been previously calibrated. Even though this bias doesn't seem disproportionate, it's effect in the long term may become large.

The running time for each filtering algorithm is shown in the following figure:

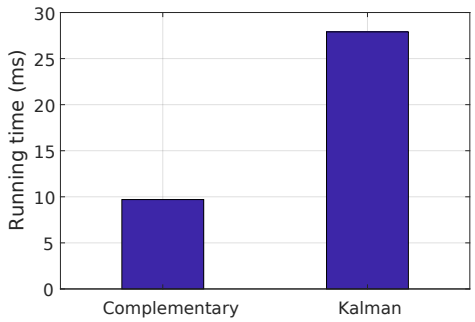

Figure 29: Running times for Kalman and complementary filters

The Kalman filter algorithm takes 3.24 times more time to run than the complementary filter.

The signals from the Kalman and complementary filters stay practically the same, but the complementary filter signal seems to differ slightly in the oscillating period.

Lastly, a small phase shift arises in both filtered signals, which could be overcome by using a faster microcontroller.

6 Conclusions and future work

The performance of the Kalman filter and the complementary filter are very similar in the studied case. This is due to the fact that the gyroscope itself provides a very accurate value of the orientation, so the data is provided mostly from it in both cases. The accelerometers only correct the bias in the long term, but don't affect much the instantaneous measurement of the device. Given that the running time of the complementary filter is more than three times lower, it's use could be more appropiate for simple applications like the one studied here.

Also, even though the Kalman filter is optimal, it's optimallity depends completely on knowing the right value of the elements of each covariance matrix, which can be difficult to get in some occasions.

Because of this, Kalman filter will be more useful for systems that present more complexity, and where a higher degree of accuracy is needed, or where the system parameters change with time and the weights must be adjusted.

We have a similar situation for the controllers. The PID controller is very easy to implement, as only 3 parameters need to be tuned to configurate the whole control. This simplicity makes of the PID a good option when the system's behavior is not precisely known and trial and error methods could be used for the adjustement of the parameters, even when there is any analytical model available. On the other hand, the LQR control is much more complex when it comes to the tuning process; not only a bigger set of parameters have to be defined, but also the dynamics of the system have to be determined (a model has to be available). Another consequence of the state-feedback nature of the LQR is that the system has to be observable (otherwise the dynamics of some variable could not affect the cost function). Nevertheless, the LQR method allows us to perform a more complete and optimized control, not only of the different state variables but also of the energy/resources consumption of the system. It also can manage a bigger set of inputs/outputs, whereas the classical PID controller is designed to control a single error parameter (although further improvements can be done).

The next step in this study will be to finish the inverted pendulum prototype and test the control algorithms. By measuring features like the settling time or the overshot of the system, we can have a good idea of which control approach is more convinient for a given system, as well as the real difficulties that can arise in the implementation and tuning processes.

References

[1] H. Goldstein, C. Poole and J. Safko. *Classical mechanics.* Adison Wesley, 2002.

[2] A. Zisserman. *Lectures on Estimators.* Oxford University, 2007.

[3] Y. Pei, S. Biswas, D. Fussell, K. Pingali. *An elementary introduction to Kalman Filtering.* CoRR, 2017.

[4] G. Welch, G. Bishop. *An introduction to the Kalman Filter.* Proc. Siggraph Course, 2006.

[5] M. Kok, J. Hol, T. Schön. *Using inertial sensors for position and orientation estimation.* Foundation and Trends of Signal Processing, 2017.

[6] P. Gui, L. Tang, S. Mukhopadhyay. *MEMS based IMU for tilting measurement: comparison of complementary and kalman filter based data fusion.* Proceedings of the 10th IEEE Conference on Industrial Electronics and Applications, 2015.

[7] J. Schwarz, M. Peck, C. Hall. *Historical review of Air-Bearing spacecraft simulators.* Journal of Guidance, Control and Dynamics, 2003.

[8] E. Kalman, Rudolph & others. *A new approach to linear filtering and prediction problems.* Journal of basic Engineering, 1960.

[9] R. Bieda, J. Krzysztof. *Determining of an object orientation in 3D space using direction cosine matrix and non-stationary Kalman filter.* Archives of Control Sciences, 2016.

[10] M. Blanke, M. Larsen. *Satellite Dynamics and control in a Quaternion Formulation (2nd edition).* Technical University of Denmark, Departament of Electrical Engineering, 2010.

[11] H. Werner. *Control Systems Theory and Design,* 2019.

[12] Wikibooks. *Control Systems.* 2019. https://en.wikibooks.org/w/index.php?title=Control_Systems&oldid=3482645.

[13] R. Jurgen. *Standard Handbook of Electronic Engineering, Fifth Edition.* Mc Graw Hill. 2005.

[14] C. Graham, S, Goodwin, M. Salgado. *Control System Design.* Valparaíso, 2000.

[15] I. Rosu. *The Bellman Principle of Optimallity.* 2002. https://pdfs.semanticscholar.org/bc80/746fb737a4de056e4a49515829bd07fd00b3.pdf

[16] I. Manchester. *Guidance, Navigation and Control.* Course notes, University of Sydney, 2013.

[17] P. Maybeck. *Stochastic Models, Estimation and Control.* Academic Press Inc, 1999.

[18] M. Gajamohan, M. Merz, I. Thommen, R. D'Andrea. *The Cubli: a cube that can jump up and balance.* IEE/RJS International Conference on Intelligent Robots and Systems, 2012.

[19] E. D. Sontag. *Mathematical Control Theory: Deterministic Finite-Dimensional Systems.* Springer-Verlag New York, 1998.

[20] J. Bellon. *Riccati Equations in Optimal Control Theory.* 2008.

[21] Granasat Electronics Aerospace Group website. https://granasat.ugr.es/

[22] E. W. Kamen. *The poles and zeros of a linear time-varying system:* Linear Algebra and it's Applications, volume 98, 1988.

[23] N. S. Nise. *Control Systems Engineering.* Wiley, 2010.

[24] S. Shinners. *Modern Control System Theory and Design.* Wiley, 1998.

[25] International Space Station website.
https://www.nasa.gov/mission_pages/station/main/index.html