A Project-Oriented Course on Guidance and Control of Autonomous Aerial Vehicles [⋆]

F. Nejjari, [*] B. Morcego [*] V. Puig [*] and C. Trapiello [*]

[*] *Advanced Control Systems Group*
Universitat Polièècnica de Catalunya (UPC)
Rambla Sant Nebridi 10, Terrassa 08222, Spain
e-mail:
{fatiha.nejjari,bernardo.morcego,vicenc.puig,carlos.trapiello}@upc.edu

Abstract: In this paper, a project oriented course on guidance and control of autonomous aerial vehicles (UAVs) is presented. The paper describes the different modules of the course and how they are addressed using the project oriented approach. The project uses a quadrotor UAV that is used as the case study during the course. In this course, the students learn how to mathematically model quadrotor UAV flight characteristics, develop feedback control algorithms to enable stable flight control and fuse sensor measurements using Kalman filter techniques to estimate the UAV position and orientation. Students develop these concepts through both simulation and interaction with UAV real measurements. Throughout the course, students build a full 6-degree-of-freedom simulation of controlled quadrotor UAV flight using MATLAB and Simulink.

Keywords: Control education, Project Based Learning, UAVs, Kalman Filter, LPV Control, Geometrical Path Following

1. INTRODUCTION

The School of Industrial, Aerospace and Audiovisual Engineering of Terrassa (ESEIAAT) offers study programmes in Industrial Engineering, Aerospace Engineering and Telecommunications Engineering. The aerospace branch of programmes contains the BSc in Aerospace Technology Engineering and the BSc in Aerospace Vehicle Engineering, besides three MSc and a PhD programme.

The course described in this paper belongs to the intensification in UAV, which is an optional set of courses of the aerospace BSc programmes. The intensification contains 6 courses of 3 ECTS each that cover most of the relevant and distinguishing aspects of UAV engineering, such as: regulations, hardware, sensors, design and control. Each course takes 5h/week during 6 weeks.

In this course, the students learn how to mathematically model quadrotor UAV flight characteristics and develop and tune feedback control algorithms to enable stable flight control, and fuse sensor measurements using Kalman filter techniques to estimate the UAV position and orientation. Students develop these concepts through both simulation and interaction with UAV real measurements. Throughout the course, students build a full 6-degree-of-freedom simulation of controlled quadrotor UAV flight using MATLAB and Simulink.

[⋆] This work has been partially funded by the Spanish State Research Agency (AEI) and the European Regional Development Fund (ERFD) through the project SCAV (ref. MINECO DPI2017-88403-R) and and by SMART Project (ref. num. EFA153/16 Interreg Cooperation Program POCTEFA 2014-2020).

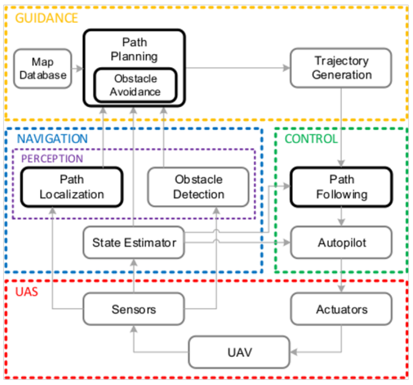

Figure 1. Control diagram of a UAV

When preparing the course, professors discussed about the possible teaching methodologies to achieve the proposed learning goals. So far, labs in most of the courses at UPC have been organised in quite a standard way. They all agreed that students “should learn by doing” and that“real-world problems capture students’ interest at the same time the students acquire and apply new knowledge in a problem-solving context”. This is in fact the key element that Project-Based Learning (PBL) methodology is focused on. For this reason, they decided to dedicate aproximately 60% to the lab activities. This methodology has proved to be an efficient way to address different areas in engineering teaching (Lamar et al., 2012).

2. QUADROTOR UAV MATHEMATICAL MODELLING

Mathematical modelling and simulation of the quadrotor is a vital part of the control system development. The aim of this module is to provide the students with the skills and knowledge that are necessary for system modelling and simulation.

2.1 Quadrotor description

The quadrotor is a vehicle that has four propellers in a cross configuration. Two propellers can rotate in a clockwise direction, while the other two can rotate anticlockwisely. The quadrotor is moved by changing the rotor speeds. For example, by increasing or decreasing together the four propeller speeds, a vertical motion is achieved. Changing only the speeds of the propellers situated oppositely produces either roll or lateral motions. Finally, a yaw rotation results from the difference in the counter-torque between each pair of propellers.

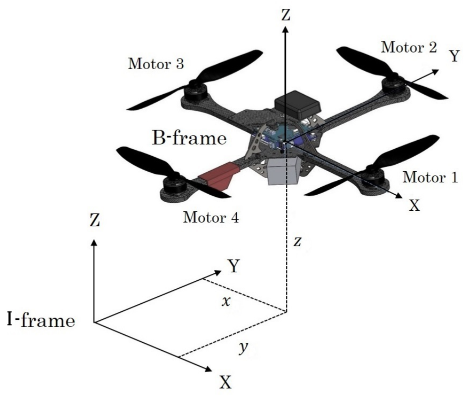

Figure 2. Quadorotor modelling frames

2.2 Mathematical model

The quadrotor model is divided in generic and specific parts. The generic part describes the dynamics of the quadrotor and the specific part defines how the structure (rotor localization and rotation direction) interacts with the system.

To describe the dynamics of the quadrotor, it is necessary to define the two frames in which it will operate: inertial frame and body frame. The inertial frame $\{\mathbf{I}\}$ is static and represents the reference of the multirotor while the body frame $\{\mathbf{B}\}$ is defined by the orientation of the quadrotor and is situated in its center of mass (see Figure 2). The two frames are related by the rotation matrix (1). $\mathbf{R}_B^I$ transforms a vector in body reference to a vector in inertial reference. In this case, the Euler angles, namely roll angle (ϕ), pitch angle (θ) and yaw angle (ψ), are used to model this rotation following the sequence $z-y-x$:

$$\mathbf{R}_B^I = \begin{bmatrix} c(\psi)c(\theta) & c(\psi)s(\theta)s(\phi)-s(\psi)c(\phi) & c(\psi)s(\theta)c(\phi)+s(\psi)s(\phi) \\ s(\psi)c(\theta) & s(\psi)s(\theta)s(\phi)+c(\psi)c(\phi) & s(\psi)s(\theta)c(\phi)-c(\psi)s(\phi) \\ -s(\theta) & c(\theta)s(\phi) & c(\theta)c(\phi) \end{bmatrix} \quad (1)$$

where $s(\cdot)$ and $c(\cdot)$ denote $\sin(\cdot)$ and $\cos(\cdot)$, respectively.

The dynamics of the quadrotor are defined using the Newton - Euler modelling approach that describe the translation and rotation of a rigid body

$$\begin{cases} \dot{\boldsymbol{\xi}}_I = \mathbf{v}_I \\ \dot{\mathbf{v}}_I = \dfrac{1}{m}(\mathbf{f}_I) \\ \dot{\boldsymbol{\eta}}_I = \mathbf{W}_\eta \boldsymbol{\omega}_B \\ \dot{\boldsymbol{\omega}}_B = \dfrac{1}{\mathbf{J}}(\boldsymbol{\tau}_B - \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega}) \end{cases} \quad (2)$$

where, $\boldsymbol{\xi} = [x\ y\ z]^T$ is the position vector in the inertial frame, $\mathbf{v}_I = [v_x\ v_y\ v_z]^T$ is the linear speed vector in the inertial frame, $\boldsymbol{\eta}_I = [\phi\ \theta\ \psi]^T$ is the orientation vector, $\boldsymbol{\omega} = [p\ q\ r]^T$ is the body angular speed vector, m is the mass of the vehicle, $\mathbf{J}$ is the inertia tensor, $\mathbf{f}_I$ and $\boldsymbol{\tau}_B$ represent the external forces and torques applied to the UAV, and $\mathbf{W}_\eta$, which represents the transformation matrix of angular velocities from the body frame to the inertial frame, is given by Blakelock (1991):

$$\mathbf{W}_\eta = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix}$$

The model equations (2) has been derived under the following assumptions (Freddi et al., 2011):

- The structure is rigid and symmetric and the propellers are also rigid.
- The thrust and the drag are proportional to the square of speed.
- The motor dynamics can be neglected since they are relatively fast.
- The inertia tensor of the body is diagonal $J = diag(J_{xx}, J_{yy}, J_{zz})$

The lift of the rotors (f_z), the translational drag and the gravity (g) and the torques generated by the rotors ($\tau_m = [\tau_x\ \tau_y\ \tau_z]^T$) and rotational drag represent the external forces and torques that interact with the vehicle.

The quadrotor model is obtained by expanding the equations (2), doing the transformations from body to inertial frame (1), and applying the previous assumptions:

$$\begin{cases} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{z} = v_z \\ \dot{v}_x = \dfrac{1}{m}[\cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi)]f_z \\ \dot{v}_y = \dfrac{1}{m}[\cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi)]f_z \\ \dot{v}_z = \dfrac{1}{m}[\cos(\phi)\cos(\theta)]f_z - g \\ \dot{\phi} = p + \sin(\phi)\tan(\theta)q + \cos(\phi)\tan(\theta)r \\ \dot{\theta} = \cos(\phi)q - \sin(\phi)r \\ \dot{\psi} = \dfrac{\sin(\phi)}{\cos(\theta)}q + \dfrac{\cos(\phi)}{\cos(\theta)}r \\ \dot{p} = \dfrac{1}{J_{xx}}[-(J_{zz} - J_{yy})qr - J_p q\Omega_p + \tau_x] \\ \dot{q} = \dfrac{1}{J_{yy}}[(J_{zz} - J_{xx})pr + J_p p\Omega_p + \tau_y] \\ \dot{r} = \dfrac{1}{J_{zz}}[-(J_{yy} - J_{xx})pq + \tau_z] \end{cases} \quad (3)$$

where J_p is the inertia moment of the rotor (rotating parts) and the propeller around Z axis and Ω_p:

$$\Omega_p = \sum_{i=1}^{4} \Omega_i \tag{4}$$

where Ω_i denotes the angular velocity of the rotor i.

2.3 Exercises (Duration: 6 hours)

In this module the students are first asked to implement the nonlinear model of the quadrotor in Simulink/Matlab and then linearize the system around a generic operating point. The parameters of the model are adjusted using real measurements from a real UAV usind system identification tools.

3. CONTROL

3.1 Control-oriented Model

The linear parameter varying (LPV) control technique is proposed because it allows designing a gain-scheduling controller with a linear-like representation of the UAV nonlinear model. Hence, an LPV modelling stage is required in order to transform the original nonlinear model (3) into an LPV system. An LPV system is a linear time-varying system whose matrices depend on a vector of time varying parameters that vary in a known interval. In a pure LPV system, the varying parameters only depend on exogenous signals. However if the varying parameters are function of the states, as in this case, it is named a quasi-LPV system (qLPV). The main advantage of these systems is that they allow to embed the system nonlinearities into the varying parameters.

Following the parameter nonlinear embedding approach proposed in Kwiatkowski et al. (2006), the set of varying parameters $\Gamma = [\gamma_1, \gamma_2, \gamma_3]^T = [\dot{\phi},\ \dot{\theta},\ \Omega]^T$ is chosen in order to embed the attitude subsystem nonlinearities. Thus, the state space representation of the attitude subsystem of (3) can be expressed in a quasi-LPV form as follows

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(\boldsymbol{\Gamma})\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \qquad \boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} \tag{5}$$

where only the state matrix is dependent of the varying parameters $\boldsymbol{\Gamma}$.

The proposed control scheme is presented in 1. In the next two subsections, the attitude and velocity control are presented. In Section 5, the position (guidance) control is described.

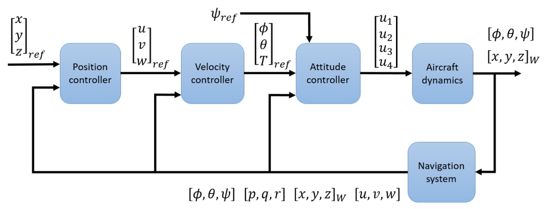
$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{ref}$
Position controller
$\begin{bmatrix} u \\ v \\ w \end{bmatrix}_{ref}$
Velocity controller
ψ_{ref}
$\begin{bmatrix} \phi \\ \theta \\ T \end{bmatrix}_{ref}$
Attitude controller
$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}$
Aircraft dynamics
$[\phi, \theta, \psi]$
$[x, y, z]_W$
Navigation system
$[\phi, \theta, \psi]$ $[p, q, r]$ $[x, y, z]_W$ $[u, v, w]$

Figure 3. Cascade control loop of a UAV

3.2 Attitude Control

To design the state feedback controller gain $\boldsymbol{K}_A$, a polytopic representation of the state matrix in (5) is used

$$\boldsymbol{A}_A(\boldsymbol{\Gamma}) = \sum_{i=1}^{2^{n_\Gamma}} \pi_i(\boldsymbol{\Gamma}) \boldsymbol{A}_{Ai} \tag{6}$$

where $\boldsymbol{A}_{Ai}$ are obtained using the bounding box approach, $n_\Gamma = 3$ is the number of scheduling variables, and $\pi_i(\boldsymbol{\Gamma})$ is given by

$$\pi_i(\boldsymbol{\Gamma}) = \prod_{j=1}^{n_\Gamma} \xi_{ij}(\eta_0^j, \eta_1^j)\ , \qquad i = \{1, ..., 2^{n_\Gamma}\} \tag{7}$$

$$\begin{aligned} \eta_0^j &= \frac{\overline{\gamma_j} - \gamma_j(t)}{\overline{\gamma_j} - \underline{\gamma_j}} \\ \eta_1^j &= 1 - \eta_0^j\ , \qquad j = \{1, ..., n_\Gamma\} \end{aligned} \tag{8}$$

where each varying parameter γ_j is known, and varies in a defined interval $\gamma_j(t) \in \left[\underline{\gamma_j}, \overline{\gamma_j}\right]$.

Then, designing a state feedback controller, the controller gain $\boldsymbol{K}_A(\boldsymbol{\Gamma})$ is given by

$$\boldsymbol{K}_A(\boldsymbol{\Gamma}) = \sum_{i=1}^{2^{n_\Gamma}} \pi_i(\boldsymbol{\Gamma}) \boldsymbol{K}_{Ai} \tag{9}$$

where $\boldsymbol{K}_{Ai}$ are the controller gains computed for the vertex qLPV system, that is, for each possible permutation of upper and lower bound of the elements in $\boldsymbol{\Gamma}$.

For designing the vertex controller gains, the LQR optimal control problem is formulated as a convex optimization problem with Linear Matrix Inequalities (LMIs) constraints. In Duan and Yu (2013), it is demonstrated how the LQR control problem for a linear system, can be reformulated into an H_2 performance problem, and hence solved via LMI techniques.

Given the LQR parameters $\boldsymbol{Q} = \boldsymbol{Q}^T \geq 0, \boldsymbol{R} = \boldsymbol{R}^T > 0$, the optimal performance bound μ (such that $J(x,u) < \mu$), the matrices $\boldsymbol{A}_{Ai}$ obtained from the vertexes of (5), and an imposed decay rate η. Then, the polytopic control gains $\boldsymbol{K}_{Ai}$ are obtained by finding $\boldsymbol{P} \in \mathbb{S}^{n_x}, \boldsymbol{W}_i \in \mathbb{S}^{n_u \times n_x}$ and $\boldsymbol{Y} \in \mathbb{S}^{n_u}$, with $n_x = 9$ and $n_u = 3$, that minimize μ while satisfying the following LMIs

$$\begin{aligned} (\boldsymbol{A}_{Ai}\boldsymbol{P} + \boldsymbol{B}_A\boldsymbol{W}_i) + (\boldsymbol{A}_{Ai}\boldsymbol{P} + \boldsymbol{B}_A\boldsymbol{W}_i)^T + 2\eta\boldsymbol{P} \leq 0 \\ trace(\boldsymbol{Q}^{\frac{1}{2}}\boldsymbol{P}(\boldsymbol{Q}^{\frac{1}{2}})^T) + trace(\boldsymbol{Y}) \leq \mu \\ \begin{bmatrix} -\boldsymbol{Y} & \boldsymbol{R}^{\frac{1}{2}}\boldsymbol{W}_i \\ (\boldsymbol{R}^{\frac{1}{2}}\boldsymbol{W}_i)^T & -\boldsymbol{P} \end{bmatrix} \leq 0 \\ i = \{1, ..., 2^{n_\Gamma}\} \end{aligned} \tag{10}$$

and applying the transformation $\boldsymbol{K}_{Ai} = -\boldsymbol{W}_i\boldsymbol{P}^{-1}$.

An external decay rate term η is introduced, in order to force the required fast dynamics of the attitude control loop. The model (5) in polytopic form is used for solving the previous LMIs. The chosen scheduling variables $\boldsymbol{\Gamma} = [\dot{\phi}, \dot{\theta}, \Omega]^T$ are bounded in the following intervals

$$\dot{\phi} \in [-2, 2]\ \frac{rad}{s}\quad \dot{\theta} \in [-2, 2]\ \frac{rad}{s}\quad and\ \ \Omega \in [-100, 100]\ \frac{rad}{s}$$

The solution of (10) returns the polytopic controller gains $\boldsymbol{K}_{Ai}$. Then, at every time step, the scheduled controller gain can be interpolated by means of (9).

3.3 Velocities Control

According to Figure 3, the second layer of control is a velocities controller that is developed in the B-frame. Expressing in B-frame, the translational subsystem is the following (see Bresciani (2008))

$$\begin{aligned} \dot{u} &= (vr - wq) + g \sin\theta \\ \dot{v} &= (wp - ur) - g \cos\theta \sin\phi \\ \dot{w} &= (uq - vp) - g \cos\theta \cos\phi + \frac{U_1}{m} \end{aligned} \tag{11}$$

where apart from the gravity projection and the thrust force, the Coriolis-centripetal term is reflected in B-frame axes. The state space representation defines the following states: $\boldsymbol{x}_v = [x_{v1}, x_{v2}, x_{v3}]^T = [u, v, w]^T$.

In the design of the velocities controller in B-frame, it was assumed that reliable measurements of the B-frame velocities $[u, v, w]^T$ are available from quadrotor set of sensors, or derived from an observer, after an E-frame $\rightarrow$ B-frame reference change.

The trigonometrical nonlinearities that the subsystem presents, prevents from directly applying LPV control techniques. However, after carrying out a feedback linearization task which defines the new input vector $\boldsymbol{u}_{fl2} = [\alpha_x, \alpha_y, \alpha_z]^T$ as

$$\begin{aligned} \alpha_x &= g \sin\theta \\ \alpha_y &= -g \cos\theta \sin\phi \\ \alpha_z &= -g \cos\theta \cos\phi + \frac{U_1}{m} \end{aligned} \tag{12}$$

Following the hypothesis presented for the attitude controller, that the estimations of rotational velocities $[p, q, r]^T$ are obtained in real time, a LPV control is designed by considering $\boldsymbol{\Gamma}_v = [\gamma_{v1}, \gamma_{v2}, \gamma_{v3}]^T = [p, q, r]^T$ as the varying parameters.

Expanding the states of the system in order to include the integral action of the velocities error: $\boldsymbol{q}_v = [q_{v1}, q_{v2}, q_{v3}]^T = \boldsymbol{C}_v \boldsymbol{x}_v - \boldsymbol{r}_{vel}$, the resulting LPV system is expressed as follows

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{q}_{1v} \\ \dot{q}_{2v} \\ \dot{q}_{3v} \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{A}_v(\boldsymbol{\Gamma}_v) & 0 \\ \boldsymbol{C}_v & 0 \end{bmatrix}}_{\boldsymbol{A}_V(\boldsymbol{\Gamma}_v)} \begin{bmatrix} u \\ v \\ w \\ q_{1v} \\ q_{2v} \\ q_{3v} \end{bmatrix} + \underbrace{\begin{bmatrix} \boldsymbol{B}_v \\ 0 \end{bmatrix}}_{\boldsymbol{B}_V} \begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \end{bmatrix} - \begin{bmatrix} 0 \\ \boldsymbol{I} \end{bmatrix} \boldsymbol{r}_{vel} \tag{13}$$

where

$$\boldsymbol{A}_v(\boldsymbol{\Gamma}_v) = \begin{bmatrix} 0 & \gamma_{v3} & -\gamma_{v2} \\ -\gamma_{v3} & 0 & \gamma_{v1} \\ \gamma_{v2} & -\gamma_{v1} & 0 \end{bmatrix} \qquad \boldsymbol{B}_v = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$\boldsymbol{C}_v = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{r}_{vel} = \begin{bmatrix} u_r \\ v_r \\ w_r \end{bmatrix} \tag{14}$$

According to (9), the controller gain $\boldsymbol{K}_v$ is scheduled as a weighted addition of the $\boldsymbol{K}_{vi}$ controller gains, which are computed for the vertexes of the LPV system (13). The LQR control problem computed for the vertex systems is obtained minimizing the parameter γ, while satisfying the LMIs expressed in (10), but for the new system matrices $\boldsymbol{A}_V$ and $\boldsymbol{B}_V$ instead. The chosen scheduling variables $\boldsymbol{\Gamma}_v = [p, q, r]^T$ are bounded within the following intervals

$$p \in [-2, 2]\ \frac{rad}{s}\quad q \in [-2, 2]\ \frac{rad}{s}\quad and\ \ r \in [-2, 2]\ \frac{rad}{s}$$

The aforementioned control action is complemented with the addition of a feedforward action as follows

$$\boldsymbol{u}_{fl2} = \begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \end{bmatrix} = -\boldsymbol{K}_v \begin{bmatrix} \boldsymbol{x}_v \\ \boldsymbol{q}_v \end{bmatrix} + \boldsymbol{N}_{ffv} \boldsymbol{r}_{vel} \tag{15}$$

In this case, the derivation of the actual control actions (ϕ, θ, U_1) from the obtained $(\alpha_x, \alpha_y, \alpha_z)$ is performed as follows

$$\begin{aligned} \theta &= \arcsin(\frac{\alpha_x}{g}) \\ \phi &= \arcsin(\frac{-\alpha_y}{\cos\theta g}) \\ U_1 &= (\alpha_z + \cos\phi \cos\theta g)\, m \end{aligned} \tag{16}$$

The particular case $\alpha_x = \pm g$, which corresponds to a pitch angle perpendicular to the ground ($\theta = \pm\frac{\pi}{2}$), leads to an indetermination in the roll angle ϕ. The possible singularities can be avoided as in the position controller case, setting slow dynamics for the velocities controller by means of a bound rate β, that assures small values for the obtained (ϕ , θ) angles, which together with (ψ_r) are set as the reference angles for the attitude controller.

3.4 Exercises (Duration: 6 hours)

The practical sessions of this part consist in first obtaining the UAV LPV model (5) from the non-linear model (3). Then, the attitude LPV model is transformed to the polytopic LPV form (6) to design an LPV LQR controller by solving the LMIs (10). This controller is implemented using the polytopic interpolation (9) and tested in the non-linear simulator of the UAV in Simulink. Figure 4 presents the attitude control response obtained.

Similarly, after the feedback linearisation (11), the velocities LPV model is transformed to the polytopic LPV form (6) to design an LPV LQR controller by solving the LMIs (10). This controller is also implemented using the polytopic interpolation (9) and tested in the non-linear

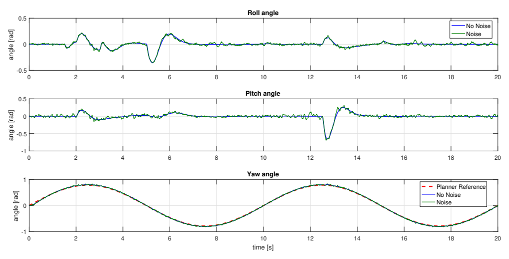

Figure 4. Attitude control response

simulator of the UAV in Simulink. Figure 5 presents the velocities control response.

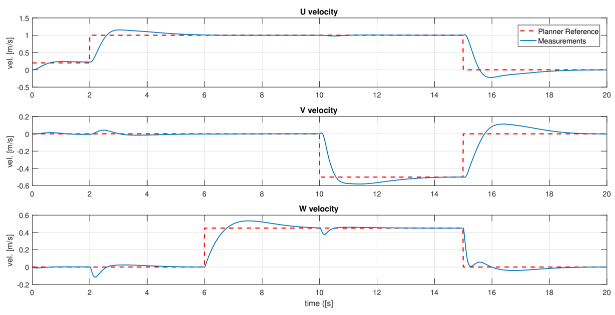

Figure 5. Velocities control response

4. SENSOR DATA FUSION

4.1 Sensor data fusion

To perform adequate control actions, a reliable estimation of the state variables must be available in real time. The aim of this module is to provide an overview of sensor fusion algorithms and applications in the context of UAV guidance and control. The main emphasis is on the Kalman Filter algorithm together with some variants and generalisations.

The IMU represents the physical device which provides information about the quadrotor's attitude and heading. It is composed of the following sensors: an accelerometer, a magnetometer, a gyroscope, a barometer and a temperature sensor.

Every sensor has its issues, the accelerometer drifts over time; the gyroscope is very sensible to the vibrations generated by the motors; the magnetometer measure is distorted by ferromagnetic materials; the GPS has a very poor accuracy and a slow update rate. Each sensor by itself is very limited, but they can be combined to compensate for their limitations. The filter takes care of this by integrating all sensors in order to obtain a more accurate state estimation.

4.2 Exercises (Duration: 6 hours)

The practical session of this part consists in developing some algorithms using a Kalman Filter and an Extended Kalman Filter to fuse the data from the accelerometer, gyroscope and magnetometer. The students use real data in this case. Figure 6 shows the estimation of the yaw angle using the kalman filter and fusing data from the magnetometer and the gyroscope.

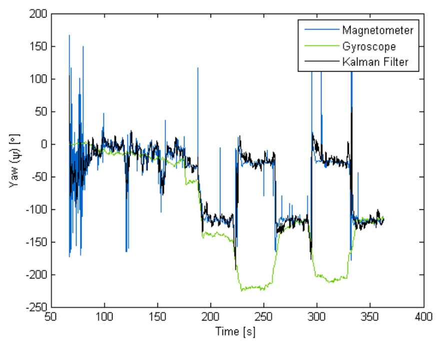

Figure 6. Yaw angle estimation

5. GUIDANCE

5.1 Guidance approach

The guidance module of the course is devoted to the position control of the vehicle, from a control perspective. Regarding Figure 1, it involves algorithms that are located in the Guidance and Control blocks. Position control is the problem of generating references for velocity and attitude controllers given mission commands, see Figure 3.

However, position control includes many control problems which can be solved using several approaches. Path planning together with obstacle avoidance and trajectory control comprise the most prominent problems to study. All these problems are discussed in class.

The problem of trajectory control is considered in depth. Trajectory control is defined as making a vehicle follow a pre-established path in space. This problem can be addressed with time constraints or without them, leading to the trajectory tracking approach and the path following approach. In the trajectory tracking problem a reference specified in time is tracked, where the references of the path are given by a temporal evolution of each space coordinate. Whereas path following (PF) handles the problem of following a path with no preassigned timing information, thus any time dependence of the problem is removed.

One example, that is described in depth in class, is the Pure Pursuit and Line-Of-Sight (PLOS) algorithm. This algorithm is simple and effective. To calculate the desired heading of the vehicle, ψ_d, it only requires two waypoints (W_i and W_{i+1}), the present position of the quadrotor (p), and the value of the only parameter it has, d_{th}, the threshold distance.

The computations are purely geometrical, no dynamics of the vehicle are needed, and that results in a very straightforward algorithm. Taking Figure 7 as a reference, the first objective consists in calculating ψ_{LOS} and ψ_P, the orientation that would take the vehicle to the closest point in the Line-Of-Sight, q, and the one that would take it to the next waypoint, W_{i+1}. the calculations are:

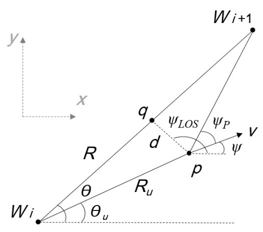

Figure 7. PLOS variables

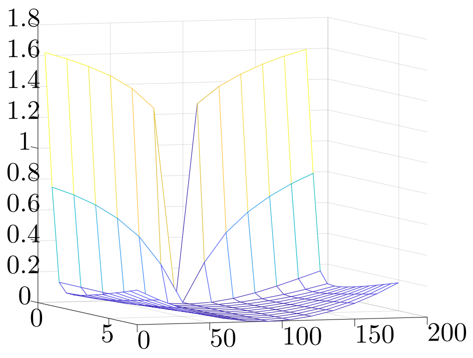

Figure 8. Carrot chasing algorithm evaluated. x-axis is the parameter value, y-axis is the initial heading of the vehicle and z-axis is the cross-track error

$$
\begin{aligned}
&\theta = \arctan(W_{i+1}(y) - W_i(y), W_{i+1}(x) - W_i(x))\\
&\theta_u = \arctan(p(y) - W_i(y), p(x) - W_i(x))\\
&R_u = \sqrt{(p(x) - W_i(x))^2 + (p(y) - W_i(y))^2}\\
&d = R_u * \sin(|\theta - \theta_u|)\\
&R = \sqrt{(R_u^2 - d^2)}\\
&q = [R * \cos(\theta) + W_i(x), R * \sin(\theta) + W_i(y)]\\
&\psi_{LOS} = \arctan(q(y) - p(y), q(x) - p(x))\\
&\psi_P = \arctan(W_{i+1}(y) - p(y), W_{i+1}(x) - p(x))
\end{aligned}
\qquad (17)
$$

After that, when the vehicle is too far from q (i.e., $d > d_{th}$) $\psi_{ref} = \psi_{LOS}$. Otherwise,

$$\psi_{ref} = \alpha\psi_{LOS} + (1 - \alpha)\psi_P \qquad (18)$$

where $\alpha = 1 - (d_{th} - d)/d_{th}$. So, when $d < d_{th}$ ψ_{ref} is a linear combination between ψ_{LOS} and ψ_P.

5.2 Exercises (Duration: 6 hours)

The practical sessions of this part consist in programming a geometric PF algorithm and evaluating it according to the mean squared error criterion. The error considered is the cross-track error, a standard way of measuring the distance of the vehicle to the path, defined as the norm of the perpendicular vector connecting the vehicle and the path, (d in Figure 7).

The PF algorithms are taken from Sujit et al. (2014). Carrot-chasing, Non Linear Guidance Law and Vector Field (each algorithm has a straight line and a loiter version) are distributed among groups of two students. They have to implement them using the simulator described in Rubí et al. (2019).

When the algorithm is programmed and tested, it has to be evaluated with different initial conditions of the aircraft and different values of its parameters. Figure 8 shows a sample evaluation of the algorithm. In this case, the vehicle follows a straight trajectory beginning at the initial waypoint with orientations that range from 0° to 180° and the performance parameter of the algorithm ranges from 0.5, where the trajectory is very unstable, to 7.

Finally, each group describes its algorithm and results to the rest of the class. This way, each student has an overview of the eight algorithms and knows how to program them and how to evaluate their performance.

6. CONCLUSIONS

In this paper, a project oriented course on Guidance and control of UAVs has been presented. In this course, the students learn how to mathematically model quadrotor UAV flight characteristics, develop and tune feedback control algorithms to enable stable flight control, and fuse sensor measurements using Kalman filter techniques to estimate the UAV position and orientation.

The PBL methodology is used throughout because the course was organized this way from the beginning. No comparison with other methodologies has been carried out, but our learning results are very encouraging because all the students passed the course with higher mean grades than in other courses of the school where a more standard methodology is followed. As future work, a comparison with other control techniques and an implementation of the developped algorithms to a real quadrotor could be performed.

REFERENCES

Blakelock, J.H. (1991). *Automatic control of aircraft and missiles.* John Wiley & Sons.

Bresciani, T. (2008). Modelling, identification and control of a quadrotor helicopter. *MSc Theses.*

Duan, G.R. and Yu, H.H. (2013). *LMIs in control systems: analysis, design and applications.* CRC press.

Freddi, A., Lanzon, A., and Longhi, S. (2011). A feedback linearization approach to fault tolerance in quadrotor vehicles. *IFAC Proceedings Volumes*, 44(1), 5413–5418.

Kwiatkowski, A., Boll, M.T., and Werner, H. (2006). Automated generation and assessment of affine lpv models. In *Decision and Control, 2006 45th IEEE Conference on*, 6690–6695. IEEE.

Lamar, D.G., Miaja, P.F., Arias, M., Rodríguez, A., Rodríguez, M., Vázquez, A., Hernando, M.M., and Sebastián, J. (2012). Experiences in the application of project-based learning in a switching-mode power supplies course. *IEEE TRANSACTIONS ON EDUCATION*, 55(1), 69–77.

Rubí, B., Ruiz, A., Pérez, R., and Morcego, B. (2019). Path-flyer: A benchmark of quadrotor path following algorithms. In *2019 15th IEEE International Conference on Control and Automation.*

Sujit, P.B., Saripalli, S., and Sousa, J.B. (2014). Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicless. *IEEE Control Systems*, 34(1), 42–59. doi: 10.1109/MCS.2013.2287568.