

# Accurate control law for low-cost UAV

Aurélien Cabarbaye\*, Titouan Verdu, Jacson Miguel Olszanecki Barth and Gautier Hattenberger  
ENAC, Avenue Edouard Belin, France

## ABSTRACT

Present article proposes a method to design a control law and the associated observer to stabilize a UAV. The design is only based on the UAV geometry, mass and propulsion system characteristics which do not require any expensive facilities or software to be obtained. The control inputs are the longitudinal airspeed, the roll angle and the slope angle to ease the guidance control whether manual or automatic. The resulting control only relies on the information provided by a 6 Degrees of Freedom (DoF) Inertial measurement unit (IMU) that makes it suitable for implementation on very basic autopilot board (cf. Paparazzi Chimera, Pixhawk XRacer, ArduPilot APM...). The propeller acts indeed as an airspeed probe which makes additional sensor unnecessary. This low-cost implementation makes it of particular interest for large UAVs fleet.

## 1 INTRODUCTION

The ENAC UAV team has developed, since 2003, the Paparazzi UAV (Unmanned Aerial Vehicle) open-source drone project. It enables to convert quickly and easily an electric powered fixed-wing airframe in a semi-automatic drone. However, just like alternative autopilot, the default control law consists of various PID which gains must be tweaked manually during the first flight tests [1] [2]. Such method turns out to present some issues:

- It is time-consuming
- It requires good piloting capabilities
- It is hazardous since it does not offer any protection against unstable modes
- It requires some practices to identify which gain must be modified and how much it may be tweaked.
- the thrust control does not consider the complex behaviour of the propulsion system.
- The resulting gain adjustment does not usually offer the shortest response time or oscillation attenuation.

\*Email address: contact.aurelien.cabarbaye@enac.fr

To solve those issues, the present article proposes a more accurate control. This latter is designed from the aeroplane model detailed in section 2, page 1 whose procurement only relies on free data and calculation tools. The control law is exposed in section 3, page 4. This control depends on the state vector which is estimated by the observer, presented in section 4, page 5. The observer is based on the already known aeroplane model of section 2. It takes thus into account the aeroplane dynamics which makes it more precise than other methods that only rely on IMU data (e.g. complementary, Kalman filters [3]). However, as these alternatives, it is designed to only require information from a low-grade IMU to be implementable on a wide range of low cost existing autopilot boards (e.g. Paparazzi Apogee [4], Pixhawk XRacer [5]) which is of great interest to make up a drone swarm [6]. The resulting control law and observer are tested by simulation in section 5, page 6

The whole process is implemented on the airplane shown on figure 1



Figure 1: Experiment airplane

## 2 AIRCRAFT MODEL

The model of the UAV is constructed from its dynamics [7]:

$$\begin{aligned}
 F_X - mg \sin \theta &= m (\dot{u}^E + qw^E - rv^E) \\
 F_Y + mg \cos \theta \sin \phi &= m (\dot{v}^E + ru^E - pw^E) \\
 F_Z + mg \cos \theta \cos \phi &= m (\dot{w}^E + pv^E - qu^E) \\
 Q_L &= I_x \dot{p} - I_{zx} \dot{r} + qr (I_z - I_y) - I_{zx} pq \\
 Q_M &= I_y \dot{q} + rp (I_x - I_z) + I_{zx} (p^2 - r^2) \\
 Q_N &= I_z \dot{r} - I_{zx} \dot{p} + pq (I_y - I_x) + I_{zx} qr \\
 p &= \dot{\phi} - \dot{\psi} \sin \theta \\
 q &= \dot{\theta} \cos \phi + \dot{\psi} \cos \theta \sin \phi \\
 r &= \dot{\psi} \cos \theta \cos \phi - \dot{\theta} \sin \phi \\
 \dot{\phi} &= p + (q \sin \phi + r \cos \phi) \tan \theta \\
 \dot{\theta} &= q \cos \phi - r \sin \phi \\
 \dot{\psi} &= (q \sin \phi + r \cos \phi) \sec \theta \\
 \dot{x}^E &= u^E \cos \theta \cos \psi + v^E (\sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi) \\
 &\quad + w^E (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \\
 \dot{y}^E &= u^E \cos \theta \sin \psi + v^E (\sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi) \\
 &\quad + w^E (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \\
 \dot{z}^E &= -u^E \sin \theta + v^E \sin \phi \cos \theta + w^E \cos \phi \cos \theta \\
 u^E &= u + W_x \\
 v^E &= v + W_y \\
 w^E &= w + W_z
 \end{aligned} \tag{1}$$

where  $F_X$ ,  $F_Y$  and  $F_Z$  are the forces,  $u$ ,  $v$  and  $w$  are the velocities,  $p$  (roll),  $q$  (pitch) and  $r$  (yaw) are the rotation speed respectively and  $W_x$ ,  $W_y$  and  $W_z$  are the wind speed on the airplane axis  $X$  (upstream),  $Y$  (right wing direction),

$Z$  (downward),  $m$  and  $I = \begin{pmatrix} I_x & I_{xy} & I_{xz} \\ I_{xy} & I_y & I_{yz} \\ I_{xz} & I_{yz} & I_z \end{pmatrix}$  are the airplane mass and moments of inertia.  $g$  is the gravity acceleration.  $x$ ,  $y$ , and  $z$  are the earth reference frame axis.  $\phi$  (roll),  $\theta$  (pitch) and  $\psi$  (yaw) are the euler angles between the earth and the airplane reference frames.

The model can be simplified with the following assumptions:

**Assumption 1** Thanks to the relative symmetries around the airplane axis,  $I = \begin{pmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{pmatrix}$

**Assumption 2** The gusts are supposed to be slow so the wind parameters  $W_x$ ,  $W_y$  and  $W_z$  are constant.

**Assumption 3** Roll and slope angles are likely to be small, in the range  $(\varphi, \phi) \in [-0.5; 0.5]^2$

**Assumption 4** Roll, pitch and yaw rate angles are likely to be small

However, the yaw angle ( $\psi$ ) is not bounded. Therefore, stabilization and guidance must be addressed separately as it is currently done in paparazzi [2].

To solve the issues exposed in section 1, only the stabilization must be modified. Therefore, the state vector is reduced to:

$$X = \begin{pmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \varphi \\ \theta \end{pmatrix} \tag{2}$$

**Assumption 5**  $v \ll 1$   $w \ll u$  which imply that the sideslip angle and the angle of attack are small:  $\beta = \tan^{-1}(\frac{v}{u}) \ll 1$  (symmetric flight) and  $\alpha = \tan^{-1}(\frac{w}{u}) \ll 1$  (unstall flight). Therefore, the aerodynamics should present an almost linear behaviour.

Thanks to assumptions 3 4 and 5, the differences between a linearized model and the real dynamics should be small. The stabilization of the UAV around a flight configuration is therefore based on its linearized model. This latter is obtained in two steps. The main body is first studied. Then the propulsion system is modelled.

### 2.1 Main body

Athena Vortex Lattice (AVL) is an open source aerodynamics simulation software based on the Vortex Lattice Methods [8]. It assumes an inviscid flow which does not allow good zero-lift drag estimation. However, it seems very accurate to assess aerodynamics evolutions [9]. It is therefore used to generate the linear model of the UAV glider base.

AVL requires information about the shape and the moment of inertia of the aeroplane.

The aeroplane shape is defined with its main aerodynamics characteristic dimensions as shown on figure 2 and with the airfoil shapes.

If the airfoil shapes are unknown, they can be reverse engineered with a profile gauge with sufficient accuracy.

Thanks to assumption 2, the moment of inertia can be easily measured following the Bifilar Pendulum methodology [10] as shown on figure 3

The model provided by AVL is of the form:

$$\dot{X} = AX + BU \tag{3}$$

Since the experiment airplane only have two aerodynamic controls (elevons), the control input is defined as:

$$U = \begin{pmatrix} \delta_{elevator} \\ \delta_{aileron} \end{pmatrix} \tag{4}$$

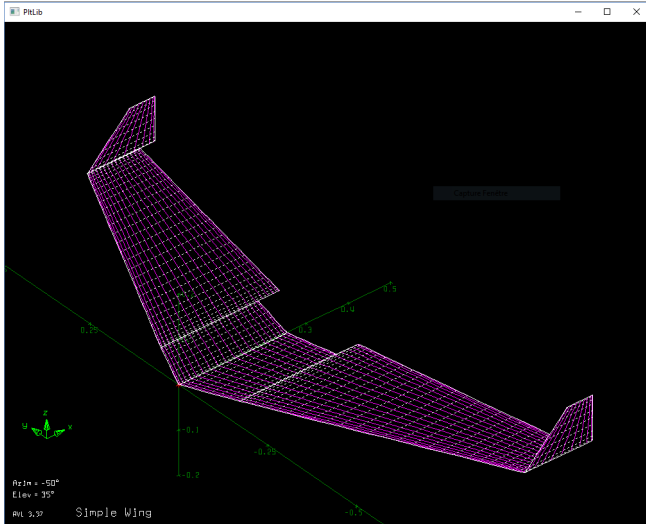


Figure 2: experiment airplane geometry in AVL

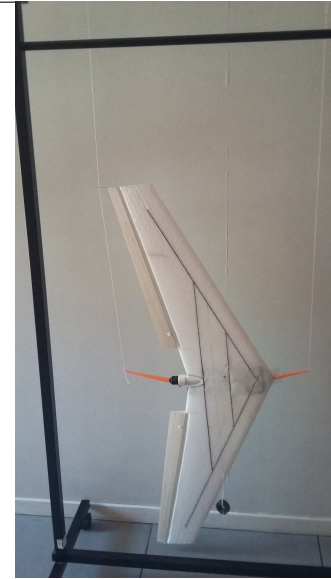


Figure 3: Moment of inertia estimation experiment

It must be noticed that the control surface deflections are managed in degree by AVL rather than in radian for the attitude angles and rates. This particularity is kept unchanged in the following of the article. The resulting elevons deflection  $\delta_{left\ elevon}$  and  $\delta_{right\ elevon}$  is computed as follows:

$$\delta_{left\ elevon} = \delta_{elevator} - \delta_{aileron} \quad (5)$$

$$\delta_{right\ elevon} = \delta_{elevator} + \delta_{aileron} \quad (6)$$

## 2.2 Propulsion system

The propulsion system is composed of a propeller and an electric motor.

**Propeller** Propeller behaviour can be described by the force  $F$  and torque  $Q$  produced.  $F$  and  $Q$  can be very well assessed from the theory resulting in the mixt between Momentum theory and blade element theory [11]. Thanks to assumption 5, the impact of the radial airspeed is neglected. The equations can be thus simplified as follow:

$$F = K f_{\omega} \omega^2 + K f_{\chi} \omega U \quad (7)$$

$$Q = K q_{\omega} \omega^2 + K q_u U^2 + K q_{\chi} \omega U \quad (8)$$

where  $\omega$  is the propeller rotation speed and  $K f_{\omega}$ ,  $K f_{\chi}$ ,  $K q_{\omega}$ ,  $K q_u$ ,  $K q_{\chi}$  are constants that must be estimated from wind tunnel test. Library of such test results are available for a wide range of propellers [12].

The simplified modelled is adjusted to the test results obtained for the APC 8x6E mounted on the experiments aeroplane. The force estimation, as well as the test data, are shown on figure 4. The model estimates very well the experimental data except for those performed in hover (on the top left-hand side). This is because the induced air velocity generated by the propeller is no more negligible compared to the general

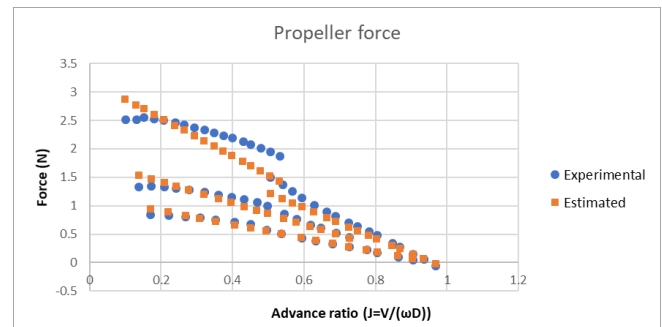


Figure 4: Force function of advance ratio  $J$  (where  $V$ ,  $\omega$  et  $D$  are respectively the airspeed, the rotation speed and the diameter of the propeller)

airspeed. However, this situation is very unlikely to happen in normal flight, so it does not call into question equation (7). The torque estimation and its corresponding test data are shown on figure 5. The very little difference between experimental and estimated data validates equation (8). The propeller proposed model can be ultimately validated with the propeller efficiency:  $\eta = \frac{F \cdot u}{Q \cdot \omega}$  shown on figure 6. The results confirm once more the suitability of equations (7) and (8).

Since these equations are very similar, propeller generated force and torque are shown together on figure 7. It can be seen that torque varies in a linear fashion with the force which gives the approximation:

$$\Delta Q = a_{Q/F} \Delta F \quad (9)$$

where  $a_{Q/F}$  is a constant.

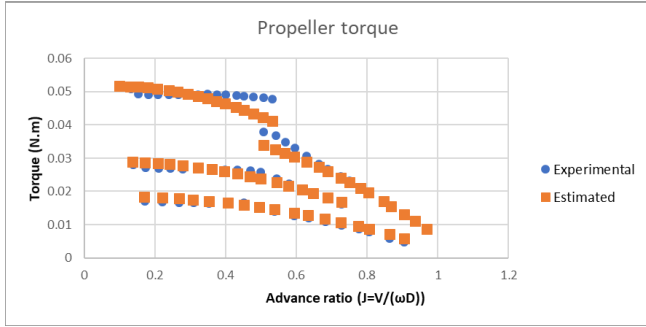


Figure 5: Torque function of advance ratio  $J$

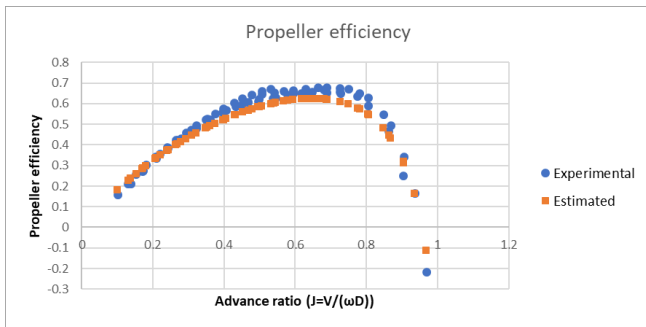


Figure 6: Efficiency function of advance ratio  $J$

**Electric motor** The electric motor behaviour is assessed thanks to two equations which gives the relation between the electrical and the mechanical parameters [13]:

$$\omega = K_V (V - RI) \quad (10)$$

$$I = K_V Q + I_0 \quad (11)$$

where  $V$  and  $I$  are respectively the voltage and the current applied to the motor and  $K_V$ ,  $I_0$  and  $R$  are respectively the motor velocity constant, zero load current and electric resistance. The model can be complexified to assess even better the motor behaviour but the adopted one seems to be sufficient [14]. Motor constants  $K_V$ ,  $I_0$  and  $R$  can be easily retrieved if not provided by the manufacturer [15].

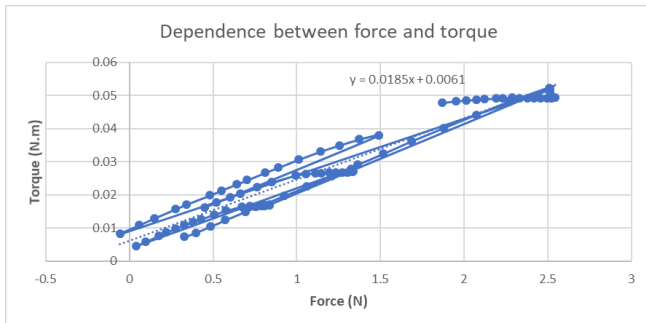


Figure 7: torque function of force

### 3 CONTROL

The aim of the control is to stabilise the aeroplane to make it as easy to steer as possible whether in manual or automatic guidance mode (cf. *AUTO1*, *AUTO2* [2]). To do so, the control must integrate a stall protection, fly the aeroplane symmetric and be easily integrated into existing autopilot.

#### 3.1 Control idea

The control states are designed first. Usually, the thrust or the power are directly controlled by the pilot in closed loop fashion. In order to ease the control, a closed loop control is built to stabilise the longitudinal velocity  $u$ .

Most aeroplane lateral controls are built to set the roll angle  $\varphi$  to a reference value. This enables an effective path following since there is a direct relation between  $\varphi$  and the turning radius.  $\varphi$  is therefore chosen as the second control state.

For the longitudinal control, there is no consensus. Aircraft are usually controlled in pitch rate, in vertical acceleration or in pitch angle [16]. A control on a derivative (cf. pitch rate, vertical acceleration) imposes more work to the guidance control. A control on pitch angle is not very precise because of the angle of attack variations during the flight. Therefore, the climb slope  $\gamma$  is chosen as third control state. Thanks to assumption 5,  $\gamma$  follows:

$$\gamma = \theta - \alpha \approx \theta - \frac{w}{u} \quad (12)$$

Applying the small disturbance linearization method [7]:

$$\Delta\gamma = \Delta\theta - \frac{\Delta w}{u_0} + \frac{w_0}{u_0^2} \Delta u = \Delta\theta - \frac{1}{u_0} \Delta w + \frac{\alpha_0}{u_0} \Delta u \quad (13)$$

Therefore, the output vector becomes:

$$Y_c = \begin{pmatrix} u \\ \varphi \\ \gamma \end{pmatrix} = C_c X \quad (14)$$

where  $C_c = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \frac{\alpha}{u_0} & 0 & -\frac{1}{u_0} & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

Three controls input are required to control the three outputs.  $F$  is chosen for the third one to remove the non-linearity of equations 7. Therefore, from equation (4)  $U$  becomes:

$$U = \begin{pmatrix} \delta_{elevator} \\ \delta_{aileron} \\ F \end{pmatrix} \quad (15)$$

And equation (3) gives

$$\dot{X} = A_c X + B_c U \quad (16)$$

$$\text{where } A_c = A \text{ and } B_c = \begin{pmatrix} \frac{1}{M} \\ 0 \\ 0 \\ B - \frac{a_{Q/F}}{I_{xx}} \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Notice that the relation between force and torque of equation (7) has been considered to compensate the torque as soon as it is applied, without the delay that would have been otherwise induced by the dynamics.

The controllability matrix  $C_m = [B_c \ A_c B_c \ A_c^2 B_c \ \dots \ A_c^7 B_c]$  has a rank equal to the length of  $A_c$ , so the system is controllable.

### 3.2 Control law

The control law is based on the method of the steady state tracking which consists in stabilising first all the states of  $X$  and then to alter some of them to converge the output to the desired value [17].

The control law is of the form:

$$U = -KX + GR \quad (17)$$

where  $K$  and  $G$  are two gain matrices and  $R$  is the desired output.

Combining equations (16) and (17) leads to:

$$\dot{X} = (A_c - B_c K) X + B_c G R \quad (18)$$

$K$  is first chosen to make all the states stable. Therefore, the poles  $p_c$  of the matrix  $(A_c - B_c K)$  are set negatives thanks to the pole placement method. In order to offer good stability properties, the poles are chosen from  $A_c$  poles. Imaginary parts are kept unchanged, positive real parts are substituted by their opposite and non-sufficiently negative real parts are changed by better ones.

Then the convergence of the output is satisfied as follows:

$$\begin{cases} \lim_{t \rightarrow \infty} (A_c - B_c K) X(t) + B_c G R = 0 \\ \lim_{t \rightarrow \infty} Y_c(t) = \lim_{t \rightarrow \infty} C_c X(t) = R \end{cases} \quad (19)$$

The system (19) leads to:

$$G = - (C_c (A_c - B_c K)^{-1} B_c)^{-1} \quad (20)$$

### 3.3 Control saturation and trim

Saturations can be added to consider the servo travel limit and the propulsion maximum power. A trim can also be added to balance the model errors. It is simply implemented modifying the values of the control required to fly in the condition of the linearization. The actual control input  $U_{trim}$  is:

$$U_{trim} = \begin{pmatrix} \delta_{elevator} + \delta_{0elevator} \\ \delta_{aileron} + \delta_{0aileron} \\ F + F_0 \end{pmatrix} \quad (21)$$

The deflections of the ailerons are obtained from equations (5) and (6). Resolving quadratic equation (7) to get  $U_{trim}$  [3] leads to  $\omega$ . Then combining equations (8), (10) and (11) leads to:

$$V = \frac{\omega}{K_V} + RK_V K_{q\omega} \omega^2 + RK_V K_{qu} U^2 + RK_V K_{q\chi} \omega U + RI_0 \quad (22)$$

where  $U = u + U_0$  and  $U_0$  is the airspeed of the linearization.

Noting that  $0 \leq V \leq V_{batt}$  and  $\delta_{min} \leq \delta_{elevation} \leq \delta_{max}$ , the control applied becomes:

$$\begin{pmatrix} V \\ \delta_{right\ elevation} \\ \delta_{left\ elevation} \end{pmatrix}_{applied} = \min \left( \begin{pmatrix} V_{batt} \\ \delta_{max} \\ \delta_{max} \end{pmatrix}, \max \left( \begin{pmatrix} V \\ \delta_{right\ elevation} \\ \delta_{left\ elevation} \end{pmatrix}, \begin{pmatrix} 0 \\ \delta_{min} \\ \delta_{min} \end{pmatrix} \right) \right) \quad (23)$$

### 3.4 Stall protection

A stall protection can be added to the control input  $\gamma$ . To balance the weight, the lift must be [18]:

$$L \cos(\varphi) = mg \cos(\gamma) \quad (24)$$

which gives

$$\gamma_{max} = \cos^{-1} \left( \frac{\frac{1}{2} \rho U^2 S C_l \cos(\varphi)}{mg} \right) \quad (25)$$

where  $\gamma_{max}$  is the limit of the desired output  $\gamma$  when  $\frac{1}{2} \rho U^2 S C_l \cos(\varphi) \leq mg$ , verifying  $\gamma_{max} \in [-\frac{\pi}{2}, 0]$ .  $\frac{1}{2} \rho U^2$  is the dynamic pressure,  $S$  is the wing surface and  $C_l$  is the lift coefficient.  $C_l$  can be for example fixed to fly, as for general aviation, at least at  $1.3 \cdot V_s$  in normal flight and at least at  $1.1 \cdot V_s$  in final approach ( $V_s$  is the stall speed). That is to say:  $C_l = \frac{C_{l_{max}}}{1.14}$  in normal flight and  $C_l = \frac{C_{l_{max}}}{1.05}$  in final approach.

## 4 OBSERVER

A good estimation of the state vector  $X$  is required by the control law (cf. equation (17)). An observer is therefore built to provide it.

### 4.1 Baseline idea

To be as low cost as possible to implement, the observer only relies on the information provided by a 6 DoF IMU. The output vector  $Y_o$  of the observer is therefore:

$$Y_o = \begin{pmatrix} \dot{u} \\ \dot{v} \\ p \\ q \\ r \end{pmatrix} = C_o X + D_o U \quad (26)$$

where  $C_o = \begin{pmatrix} A(1:3, 1:8) \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$  and  $D_o = \begin{pmatrix} B_o(1:3, 1:3) \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$  The Observability matrix  $O_m = \begin{bmatrix} C_o \\ C_o A_c \\ C_o A_c^2 \\ \dots \\ C_o A_c^7 \end{bmatrix}$  has a rank equal to the length of  $A_c$ , so the system is observable.

The dynamics of the simulated observer is [19]:

$$\dot{\hat{X}} = A_c \hat{X} + B_c U + L (Y_o - \hat{Y}_o) \quad (27)$$

where  $\hat{Y}_o = C_o \hat{X} + D_o U$  From equations (16), (26) and (27), the estimation error  $\tilde{X} = X - \hat{X}$  dynamics is:

$$\dot{\tilde{X}} = \dot{X} - \dot{\hat{X}} = (A_c - LC_o) \tilde{X} \quad (28)$$

To urge the  $\tilde{X}$  to zero, matrix  $L$  is chosen following the pole placement method with all the poles  $p_o$  of  $(A_c - LC_o)$  negatives.

#### 4.2 Observer improvements

As it will be shown in section 5, page 6, the approximation introduced in equation (7) produces an error of estimation. To improve the observer, the non-linearity of the propulsion system is considered. Equation (27) is thus modified as follows:

$$\dot{\hat{X}} = A_c \hat{X} + B_o U + EQ + L (Y_o - \hat{Y}_o) \quad (29)$$

where  $B_o = \begin{pmatrix} \frac{1}{M} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$  and  $E = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{-1}{I_{xx}} \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$  and ma-

trix  $L$  obtained in section 4.1, page 5 is kept unchanged.  $Q$  and  $F$  are computed thanks to equations (7) and (8) where  $\omega$  is computed solving quadratic equation (22), where  $V$  is the result of equation (23).

### 5 SIMULATION

A simulation is performed for the experiments aeroplane. The linearization has been made for  $u = 10 \text{ m} \cdot \text{s}^{-1}$  in trimmed steady flight. The resulting data considered for the simulation are exposed in Appendix A:, page 8.

#### 5.1 Control results

The control input of equation (17) is fixed to:

$$R = \begin{pmatrix} 5 \\ -0.5 \\ 0.5 \end{pmatrix} \quad (30)$$

The initial state is fixed arbitrarily to:

$$X_0 = \begin{pmatrix} 1 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0.5 \\ 0 \end{pmatrix} \quad (31)$$

States evolution is shown on figure 8. All the states are stable

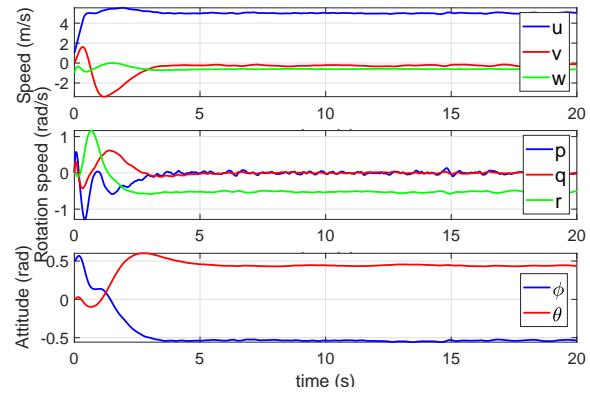


Figure 8: States  $X$

and converge asymptotically which confirms that the control law (17) can perform its first task. The control output  $Y_c$ , defined in equation (14), is shown on figure 9. It converges well

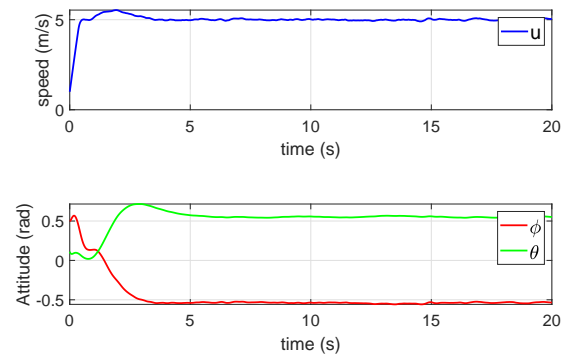


Figure 9: Control output  $Y_c$

to  $R$  fixed in equation (30), which confirms that the control law (17) performs its second task.

The control  $U$  is shown on figure 10 The saturations work

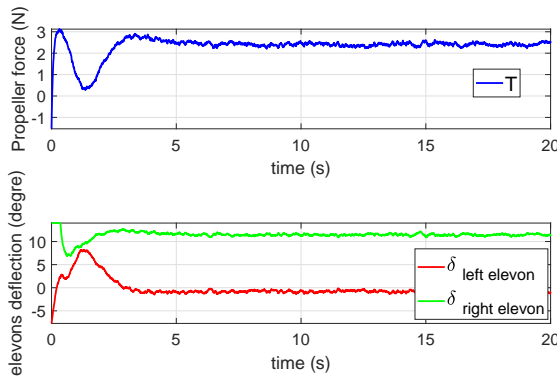


Figure 10: Control  $U$

properly (cf.  $\delta_{right-elevon}$ ).

### 5.2 Observer results

The observer is based on the information given by an IMU. This latter is supposed to be very noisy. To simulate it properly, a white noise has been added to each state of equation (26). The resulting collected information is shown on figure 11 The integration of the observer starts with the esti-

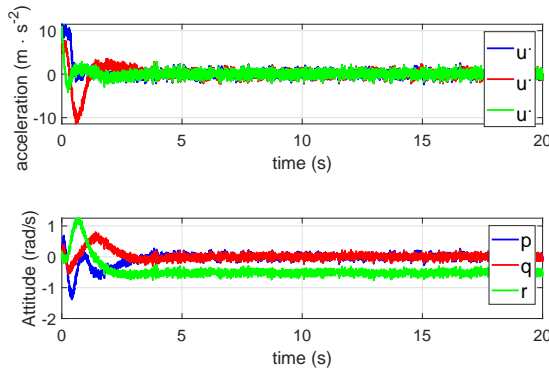


Figure 11: Observer output  $Y_o$  provided by IMU

mate states  $\hat{X}$  arbitrarily set to:

$$\hat{X}_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \quad (32)$$

Figure 12 shows what would have been the estimation error of the observer if it has been kept in its baseline form defined in equation (27). It can be noticed that if it seems to

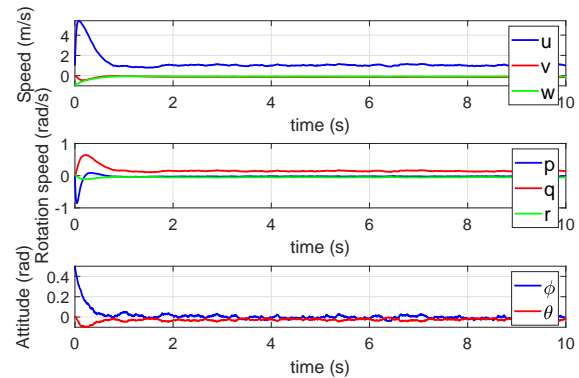


Figure 12: Estimation error  $\tilde{X}$

converge, a bias remains, in particular for the airplane longitudinal airspeed  $u$ . That is why the observer is improved in section 4.2, page 6. Figure 13 shows the estimation error of the final observer defined by equation (29). All the states

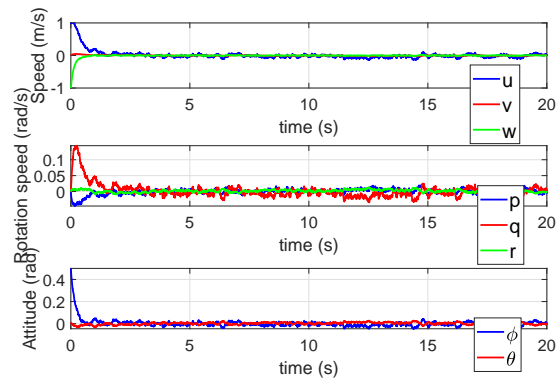


Figure 13: Estimation error  $\tilde{X}$

of equation (2) are precisely estimated.  $u$  estimation is very little bias even though no specific sensor has been used (e.g. pitot tube). That can be explained by the propeller which depends a lot on airspeed  $u$  (cf. equation (7)) and therefore acts as an alternative airspeed probe.

Moreover, the estimation errors converge very rapidly to zero (less than a second) which makes it able to compensate for most of wind gusts. If a faster convergence is needed, the poles  $p_o$  can be increased.

## 6 CONCLUSION

The aircraft model has been built in section 2 in two steps. First, the linearized main aeroplane body model is obtained from AVL software. Then the non-linear propulsion system

model is adjusted from freely available wind tunnel tests. A simplification of the propeller model enables to build a linear overall model that is used as a starting point for the control law design in section 3. This control law focusses on airspeed  $u$  and angles  $\phi$  and  $\gamma$  to ease the guidance whether manual or automatic. The states knowledge required by the control is provided by an observer designed in section 4. This observer is based on the non-linear model resulting from the mix between the linear main body model and the non-linear propulsion model, rather than on the linear model used for the control law design. This enables to track much better the states and in particular, the longitudinal speed since the propeller plays the part of an airspeed probe. In addition, the observer high convergence speed makes it suitable to deal with wind gusts. Tests of control law and observer are performed by simulation in section 5. Results seem to prove the suitability of the proposed method.

#### REFERENCES

- [1] PX4 Dev Team. Fixedwing pid tuning guide, August 2018.
- [2] ENAC Drone Team. Subsystem/control, December 2016.
- [3] Jossué Cariño Escobar, Aurélien Cabarbaye, Moisés Bonilla Estrada, and Rogelio Lozano. Quaternion kalman filter for inertial measurement units. In *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*, pages 1037–1043. IEEE, 2017.
- [4] ENAC Drone Team. Apogee/v1.00, August 2018.
- [5] PX4 Dev Team. Pixracer, September 2018.
- [6] Andrew W Sanders. Drone swarms. Technical report, US Army School for Advanced Military Studies Fort Leavenworth United States, 2017.
- [7] B. Etkin. *Dynamics of Atmospheric Flight*. Dover Books on Aeronautical Engineering. Dover Publications, 2012.
- [8] Mark Drela. Avl 3.36 user primer, February 2017.
- [9] Ramon Lopez Pereira. Validation of software for the calculation of aerodynamic coefficients: with a focus on the software package tornado. Master's thesis, Linköping Universitet, 2010.
- [10] Matthew Jardin and Eric R. Mueller. Optimized measurements of uav mass moment of inertia with a bifilar pendulum. 46:763–775, 05 2009.
- [11] W. Johnson. *Helicopter Theory*. Dover Books on Aeronautical Engineering. Dover Publications, 2012.
- [12] J.B. Brandt, G.K. Deters, R.W. Ananda, and M.S. Selig. *UIUC Propeller Database*. University of Illinois at Urbana-Champaign, August 2018.
- [13] Mark Drela. *First-Order DC Electric Motor Model*. MIT Aero & Astro, February 2007.
- [14] Mark Drela. *Second-Order DC Electric Motor Model*. MIT Aero & Astro, March 2006.
- [15] Mark Drela. *Measurement of Brushed DC Electric Motor Constants*. MIT Aero & Astro.
- [16] P.G. Hamel. *In-Flight Simulators and Fly-by-Wire/Light Demonstrators: A Historical Account of International Aeronautical Research*. Springer International Publishing, 2017.
- [17] R.L. Williams and D.A. Lawrence. *Linear State-Space Control Systems*. John Wiley & Sons, 2007.
- [18] D.P. Raymer. *Aircraft design: A conceptual approach*, 2006.
- [19] C.T. Chen. *Analog and Digital Control System Design: Transfer-Function, State-Space, and Algebraic Methods*. OUP USA, 2006.

#### APPENDIX A: DATA



$$Kf_{\omega} = 6.49343E - 06; \tag{33}$$

$$Kf_{\xi} = -0.000209663; \tag{34}$$

$$Kq_{\omega} = 1.03864E - 07; \tag{35}$$

$$Kq_{\xi} = 5.02077E - 07; \tag{36}$$

$$Kq_u = -9.98976E - 05; \tag{37}$$

$$a_{Q/F} = 0.0185 \tag{38}$$

$$K_v = 1200; \tag{39}$$

$$I_0 = 0.1; \tag{40}$$

$$R = 0.165; \tag{41}$$

$$a_{Q/F} = 0.0185 \tag{42}$$

$$A = \begin{pmatrix} -0.2153 & -0.0001 & 1.61 & 0.0003 & -1.0409 & 0 & 0 & -9.81 \\ 0 & -0.5008 & 0.0005 & 1.1056 & 0 & -9.7908 & 9.81 & 0 \\ -0.8621 & 0 & -9.4525 & 0 & 8.2192 & 0 & 0 & 0 \\ 0 & -5.76980 & -13.6429 & 0 & 2.3098 & 0 & 0 & 0 \\ 0.7595 & 0 & -7.3621 & 0 & -4.8625 & 0 & 0 & 0 \\ -0.0003 & 0.702 & -0.0154 & -2.4765 & -0.0011 & -0.309 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{43}$$

$$B = \begin{pmatrix} 2.42E - 02 & 2.74E - 05 \\ 1.33E - 05 & -3.56E - 02 \\ -0.5802 & 7.61E - 09 \\ -6.95E - 07 & -3.064 \\ -3.421 & -5.71E - 07 \\ -3.06E - 04 & -5.95E - 03 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \tag{44}$$

$$p_c = \begin{pmatrix} -7.1559 + 7.4942i \\ -7.1559 - 7.4942i \\ -13.8157 \\ -1.093 + 1.1063i \\ -1.093 - 1.1063i \\ -3.158 + 4.6121i \\ -3.158 - 4.6121i \\ -1 \end{pmatrix} \tag{45}$$

$$K = \begin{pmatrix} -0.8080 & -0.8512 & 1.7957 & -1.8668 & -0.9899 & 0.3489 & -4.6903 & -3.0280 \\ -2.1717 & 1.2099 & 0.6040 & 0.3691 & 0.0817 & 2.9449 & -10.2553 & 1.7359 \\ 7.4471 & -0.0382 & -4.6566 & 0.0428 & -0.2710 & 0.4464 & -2.1005 & -14.0328 \end{pmatrix} \tag{46}$$

$$G = \begin{pmatrix} -0.4752 & -4.7156 & -3.0280 \\ -2.3042 & -6.9212 & 0.6674 \\ 8.5137 & -1.6796 & -4.2228 \end{pmatrix} \quad (47)$$

$$p_o = 2 \begin{pmatrix} -1 \\ -1.5 \\ -2 \\ -2.5 \\ -3 \\ -3.5 \\ -4 \\ -4.5 \end{pmatrix} \quad (48)$$

$$L = \begin{pmatrix} 1.3943 & -0.0794 & -0.0633 & 2.5619 & 17.2684 & -1.0672 \\ -0.3175 & 0.9591 & -0.1471 & -5.7566 & -0.4992 & 0.3051 \\ -0.0334 & 0.0123 & 0.0628 & -0.3708 & 6.3127 & 0.1625 \\ 0.2940 & 0.0375 & 0.1486 & -1.7129 & 0.3492 & 1.4643 \\ 0.0993 & -0.0269 & 0.7555 & 0.6391 & -0.5620 & -0.3317 \\ -0.0357 & -0.0046 & -0.0165 & -3.6842 & -0.0503 & 1.7936 \\ 0.0332 & 0.5900 & -0.0037 & 0.0501 & -0.0625 & 5.8123 \\ -0.4986 & -0.0451 & -0.0852 & -0.2193 & 0.4353 & -0.4092 \end{pmatrix} \quad (49)$$