CentraleSupélec | université PARIS-SACLAY

TP1 final report
Control Theory

Laboratory work - Control Theory course
Helicopter tilt control

Voie A - Groupe 23:
CHIQUETTO DO LAGO, Ricardo
HABIB FRAGOSO COSTA, Renato
REGAZZI FERREIRA DA SILVA, João Pedro

September – October 2023

Contents

1 Modeling

2 Parameter identification for the motor drive

2.1 Transfer function of the motor drive

Initially, we recorded the motor's response to an input step ranging from 0 to 6V peak to peak. Figure 1, depicted below, illustrates the motor speed over time.

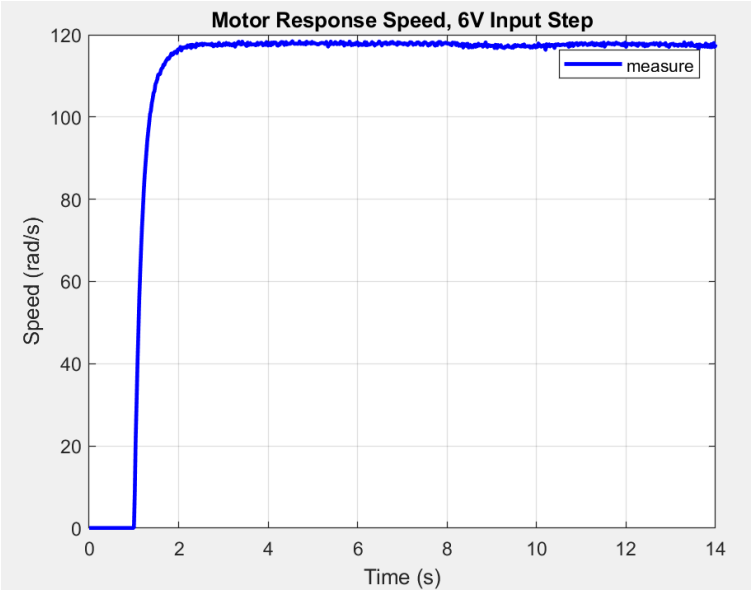


Figure 1: Motor speed over time. Input step with amplitude from 0V to 6V.

2.1.1 Determining K_v and τ from experiment data

Knowing the following transfer function of the motor drive and the input $v_p(t)$

$$G(s) = \frac{\Omega_h(s)}{V_p(s)} = \frac{K_v}{1+\tau s} \tag{1}$$

$$v_p(t) = H(t-1) * 6, \;\; H(t) = \begin{cases} 0, & \text{if } t < 0 \\ 1, & \text{if } t \geq 0 \end{cases} \tag{2}$$

We can use the laplace transform to find an expression of $\omega_h(t)$ in terms of K_v

$$\mathcal{L}\{v_p(s)\} = \mathcal{L}\{6 \cdot H(t-1)\} = 6 \cdot \frac{e^{-s}}{s} = V_p(s)$$

$$\Omega_h(s) = V_p(s) \cdot G(s) = \frac{K_v}{\tau} \cdot \frac{1}{\frac{1}{\tau}+s} \cdot \frac{6e^{-s}}{s} = \frac{6K_v}{\tau} \cdot \frac{e^{-s}}{s} \cdot \frac{1}{\frac{1}{\tau}+s}$$

$$\Omega_h(s) = \left(\frac{R_1}{s} + \frac{R_2}{\frac{1}{\tau} + s} \right) \cdot e^{-s}$$

$$f(s) = \frac{\Omega_h(s)}{e^{-s}} = \left(\frac{R_1}{s} + \frac{R_2}{\frac{1}{\tau} + s} \right)$$

$$R_1 = f(s) \cdot s \Big|_{s=0} = \frac{6K_v}{\tau} \cdot \frac{1}{\frac{1}{\tau} + 0} = 6K_v$$

$$R_2 = f(s) \cdot \left(\frac{1}{\tau} + s \right) \Big|_{s=\frac{-1}{\tau}} = \frac{6K_v}{\tau} \cdot \frac{1}{\frac{-1}{\tau}} = -6K_v$$

Therefore,

$$\Omega_h(s) = \left(\frac{6K_v}{s} - \frac{6K_v}{\frac{1}{\tau} + s} \right) \cdot e^{-s} \tag{3}$$

$$\omega_h(t) = 6K_v \cdot H(t-1) \cdot (1 - e^{\frac{-1}{\tau}(t-1)}) \tag{4}$$

Hence, knowing that τ must be greater than zero, the $\lim_{t\to\infty} \omega_h(t) = 6K_v$. In this regard, K_v is obtainable from Figure 1 as the one-sixth of the value at which the ω_h stabilizes. Our experiment resulted in $K_v = 19.5939$.

The parameter τ was easier to determine, beeing just the value of $t-1$ when $\omega_h(t)$ reaches 63.2% of its maximum value. From the results shown in Figure 1, $\tau = 0.18537$.

2.1.2 Validation of the identified parameters

To validate K_v and τ, a comparison was made between the measured motor response and the results derived from simulating the identified model. As depicted in Figure 2, the simulation closely aligns with the measured data, signifying the accuracy of the parameters determined for the input step with an amplitude of $0 - 6V$.

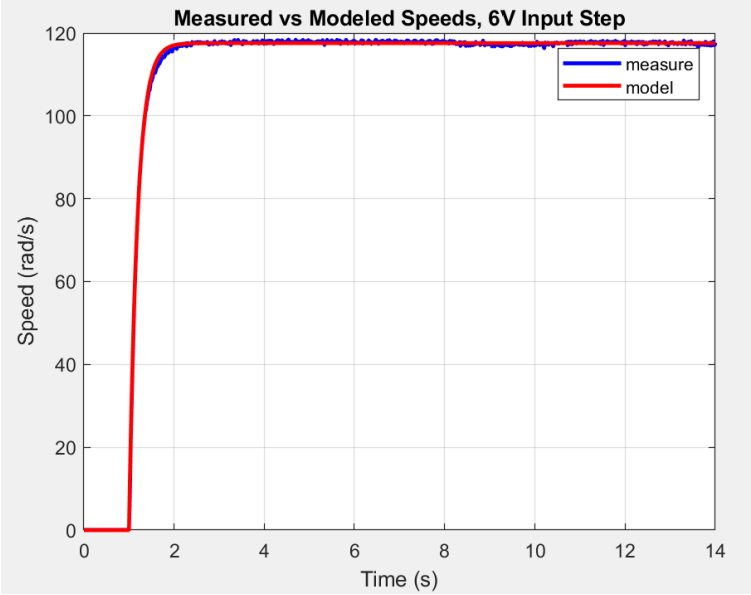


Figure 2: Motor speed over time. Input step with amplitude from 0V to 6V.

2.1.3 Validation of the model for other inputs

While the K_v and τ values produced a model with accurate results, as demonstrated in the comparison in Figure 2, it is imperative to verify their validity across a range of input variations within the motor system. With that in mind, we can observe from the charts that the assumption of linearity does not hold true. Higher input increments cause the model's results to exceed the actual measurements, and conversely, the defined K_v loses its accuracy with varying inputs.

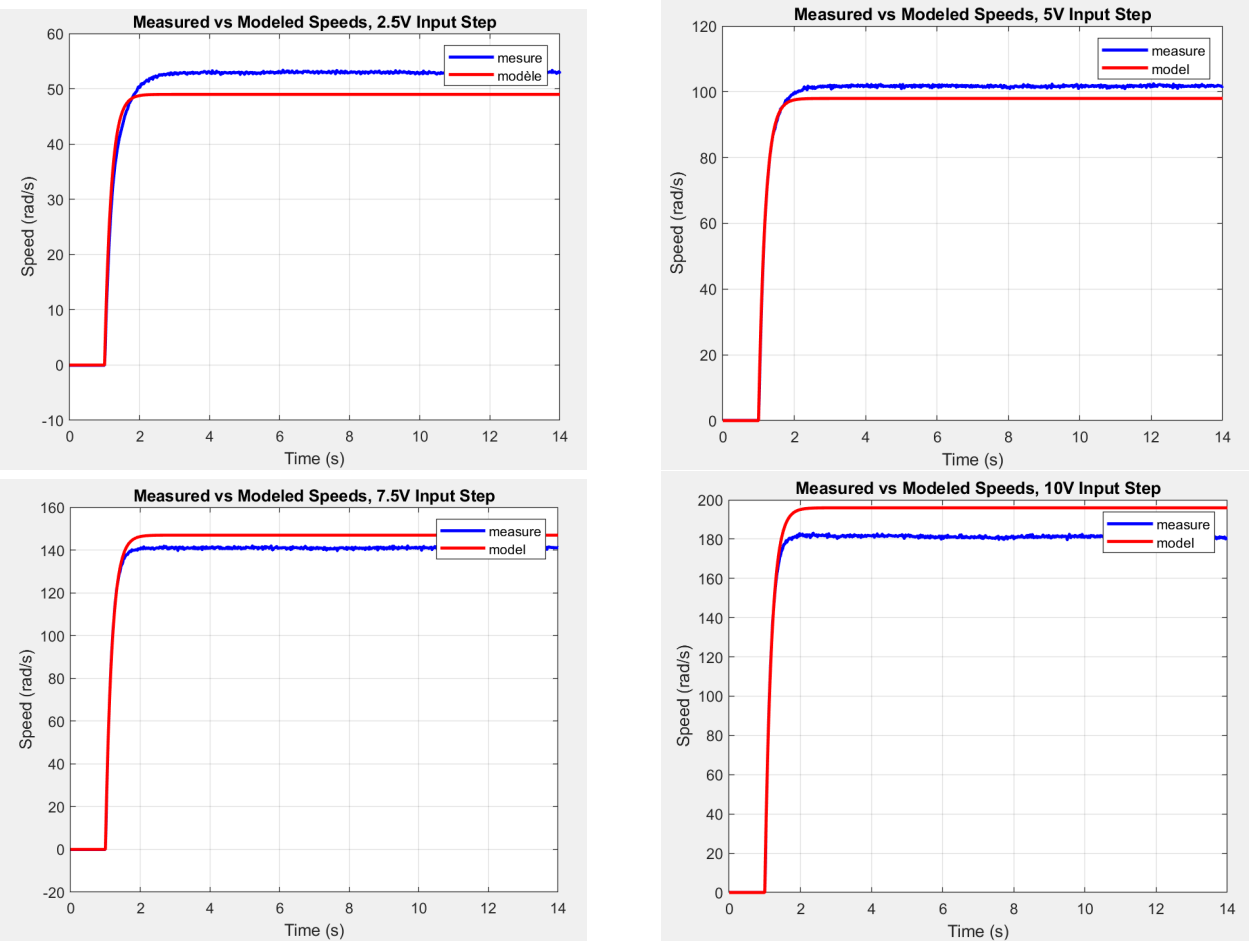


Figure 3: Measured motor response compared to model for different voltage inputs.

2.2 Motor speed control

2.2.1 State representation of the motorization chain

From the definitions, we have that:

$$\begin{cases} \dot{x}_1(t) = -\frac{1}{\tau}x_1(t) + \frac{K_v}{\tau}v_p(t) \\ \dot{z}(t) = \omega_h - \omega_h^{ref} \\ y(t) = x_1(t) = \omega_h(t) \end{cases} \tag{5}$$

Knowing that $x_1(t) = \omega_h$ and rearranging the equations in matrices:

$$\begin{cases} \begin{bmatrix} \dot{\omega}_h(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau} & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \omega_h(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} \frac{K_v}{\tau} \\ 0 \end{bmatrix} v_p(t) + \begin{bmatrix} 0 \\ -1 \end{bmatrix} \omega_h^{ref} \\ y(t) = x_1(t) = \omega_h(t) \end{cases} \tag{6}$$

Then we can define the matrices $A = \begin{bmatrix} -\frac{1}{\tau} & 0 \\ 1 & 0 \end{bmatrix}$, $B_1 = \begin{bmatrix} \frac{K_v}{\tau} \\ 0 \end{bmatrix}$, $B_2 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$ and $C = \begin{bmatrix} 1 & 0 \end{bmatrix}$, so that:

$$\begin{cases} \dot{x}_a(t) = A \cdot x_a(t) + B_1 \cdot v_p(t) + B_2 \cdot \omega_h^{ref} \\ y(t) = C \cdot x_a(t) \end{cases} \tag{7}$$

2.2.2 Synthesis of the motor speed control

When defined the feedback control law as $u(t) = -K_a x_a(t)$, $u(t) = [v_p(t)]$, because the input w_h^{ref} is not part of the state feedback control law:

$$\begin{cases} \dot{x}_a(t) = A \cdot x_a(t) + B_1 \cdot (-K_a x_a(t)) + B_2 \cdot \omega_h^{ref} \\ y(t) = \omega_h(t) \end{cases} \tag{8}$$

$$\Rightarrow \begin{cases} \dot{x}_a(t) = (A - B_1 K_a) \cdot x_a(t) + B_2 \cdot \omega_h^{ref} \\ y(t) = \omega_h(t) \end{cases} \tag{9}$$

Then, the closed loop system can be observed like a SISO system, where w_h^{ref} is the input and w_h is the output, so that $A' = (A - B_1 K_a)$. Subsequently,the following eigenvalues values were set using the MATLAB *place* function $[\xi\omega_0 - i\omega_0\sqrt{1-\xi^2}, -\xi\omega_0 + i\omega_0\sqrt{1-\xi^2}]$, with $\omega_0 = 22$ and $\xi = 0.7$. Consequently, we found:

$$K_a = [0.2404, 4.5789] \tag{10}$$

2.2.3 Analysis of closed loop responses

As we can see in Figure 4, the control signal respects the determined constraint of $\pm 24V$. On top of that, it's also possible to observe in Figure 5 that the motor speed ω_h rapidly stabilizes as the same value of the reference signal speed ω_h^{ref}. This behavior is consistent with the selected eigenvalues because both of them are in the left side of the complex plane, which means that the system is controlable. Therefore, all the imposed specifications are verified.

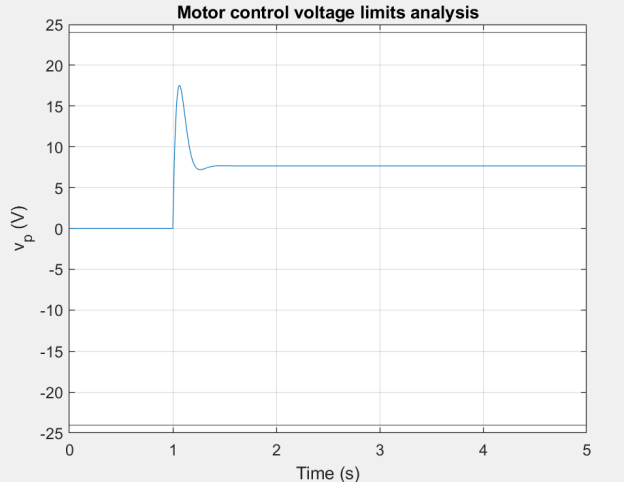


Figure 4: Motor control voltage compared to constraint of $\pm 24V$.

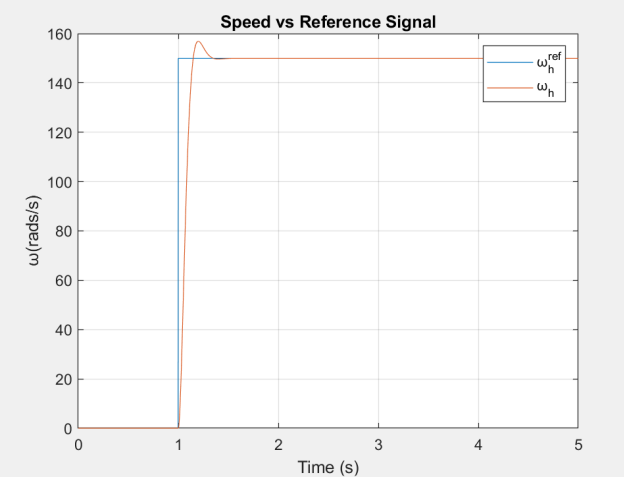


Figure 5: Simulated motor speed over time compared to reference signal of 150 rad/s.

2.3 Digital implementation of the speed controller

2.3.1 Initialization of control parameters

Figure 6 presents the results of the servo control for setpoints with an amplitude of $150 rad/s$.

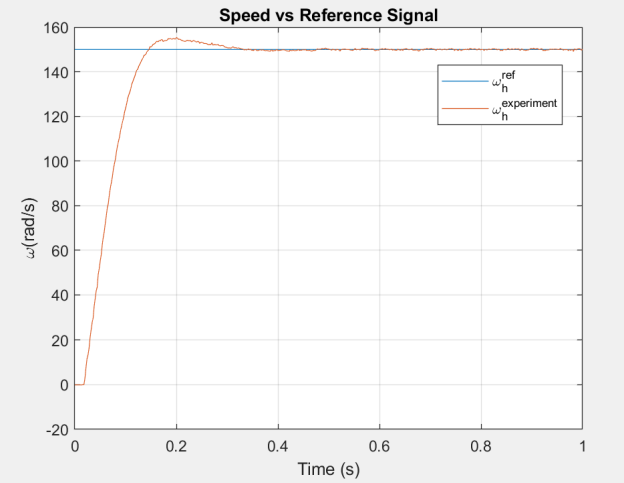


Figure 6: Motor speed over time compared to reference signal with an amplitude of 150 rad/s.

Additionally, Figure 7 shows the comparison between simulation and experimental data. As we can see, both curves are close together, indicating the success of the modeling. Control saturation does not result in a deviation from the specified time of establishment, which is approximately 200 ms in a closed loop. In the other hand, with larger step setpoints amplitudes, the effects of saturation would become more evident.

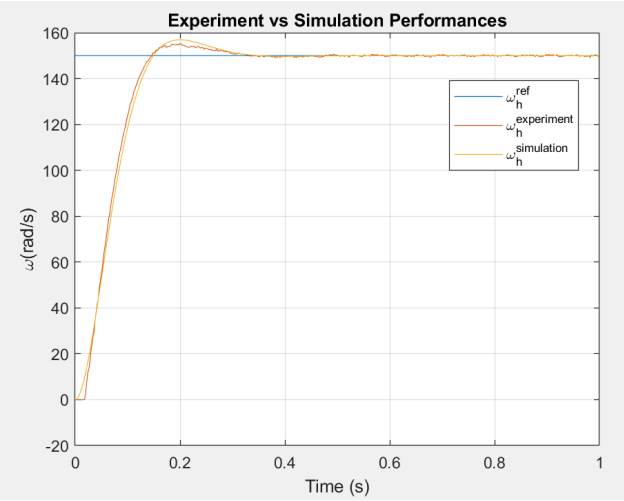


Figure 7: Comparison between simulation and experiment motor speeds over time and reference signal with an amplitude of 150 rad/s.

The increase in the sample time within the control system had a significant impact on its performance, resulting in a longer time to stabilize and larger fluctuations. This effect is primarily due to the discrete nature of digital control systems. When you increase the sample time, the controller updates it's output less frequently, which means it has less information about the system's current state. As a result, the controller's response becomes less agile, and it may not be able to

correct deviations from the setpoint as quickly or accurately. This delayed response can lead to overshooting or oscillations, manifesting as larger fluctuations.

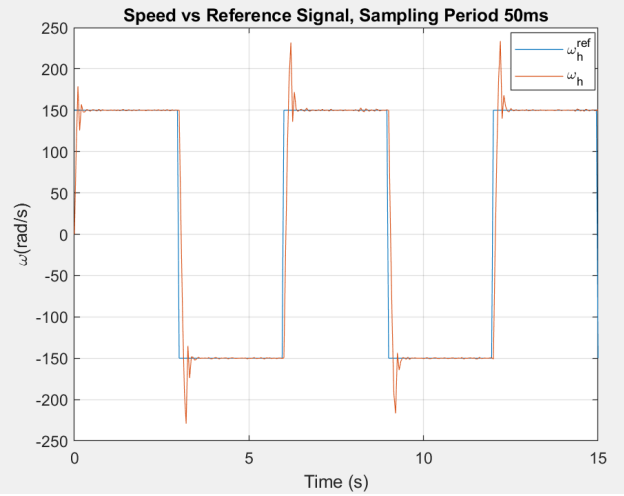


Figure 8: Motor speed over time compared to reference signal with an amplitude of 150 rad/s. Sampling period of $50ms$.

3 Identification of tilt motion parameters

3.1 Study of the dynamic behavior around one equilibrium position $\tilde{\theta}$ and $F_0(\omega_h)$ identification.

3.1.1 State representation

Firstly,

$$\begin{aligned} u(t) &= \omega_h(t) \\ y(t) &= \theta(t) \end{aligned} \tag{11}$$

and we can define the state x as

$$x(t) = \begin{bmatrix} \theta(t) \\ \dot{\theta}(t) \end{bmatrix} \tag{12}$$

.

Then, with the information given by the equation 13 we can write the state representation as the equation 14:

$$J_p \cdot \ddot{\theta} = F_0 \cdot D_t - D_p\dot{\theta} - M_b \cdot g \cdot D_m sin(\theta) \tag{13}$$

$$\begin{cases} \dot{x(t)} = f'(x(t), u(t)) = \begin{bmatrix} \dot{\theta}(t) \\ \frac{1}{Jp}[F_0(t)D_t - D_p\dot{\theta}(t) - M_b g D_m sin(\theta(t))] \end{bmatrix} \\ y(t) = g'(x(t), u(t)) = \theta(t) \end{cases} \tag{14}$$

3.1.2 Relationship between $\bar{v_p}$ and $\bar{\theta}$

At the equilibrium, all derivates are null, so it can rewrite the state equation of the state representation 5 and obtain this relation:

$$0 = -\frac{1}{\tau}\bar{\omega}_h + \frac{K_v}{\tau}\bar{v}_p \Rightarrow \bar{\omega}_h = K_v\bar{v}_p \tag{15}$$

Similarly, from the state representation 14 that is possible to write

$$0 = \bar{F}_0 D_t - 0 - M_b g_m sin(\bar{\theta}) \tag{16}$$

So, using $F_0(\omega_h)$ to connect both equations,

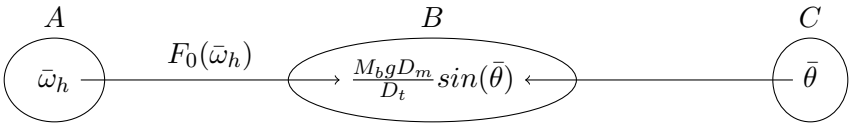
$$\bar{F}_0 = F_0(\bar{\omega}_h) \Rightarrow F_0(K_v\bar{v}_p)D_t = M_b g D_m sin(\bar{\theta}) \Rightarrow sin(\bar{\theta}) = \frac{F_0(K_v\bar{v}_p)D_t}{M_b g D_m} \tag{17}$$

3.1.3 Further studies and measurements of $F_0(\bar{\omega}_h)$

From the equation 16,

$$F_0(\bar{\omega}_h) = \frac{M_b g D_m}{D_t} sin(\bar{\theta}) \tag{18}$$

Then, it is possible to define the relation between $\bar{\omega}_h$, $\bar{\theta}$ and $F_0(\bar{\omega}_h)$ when the last one is considered as a function:



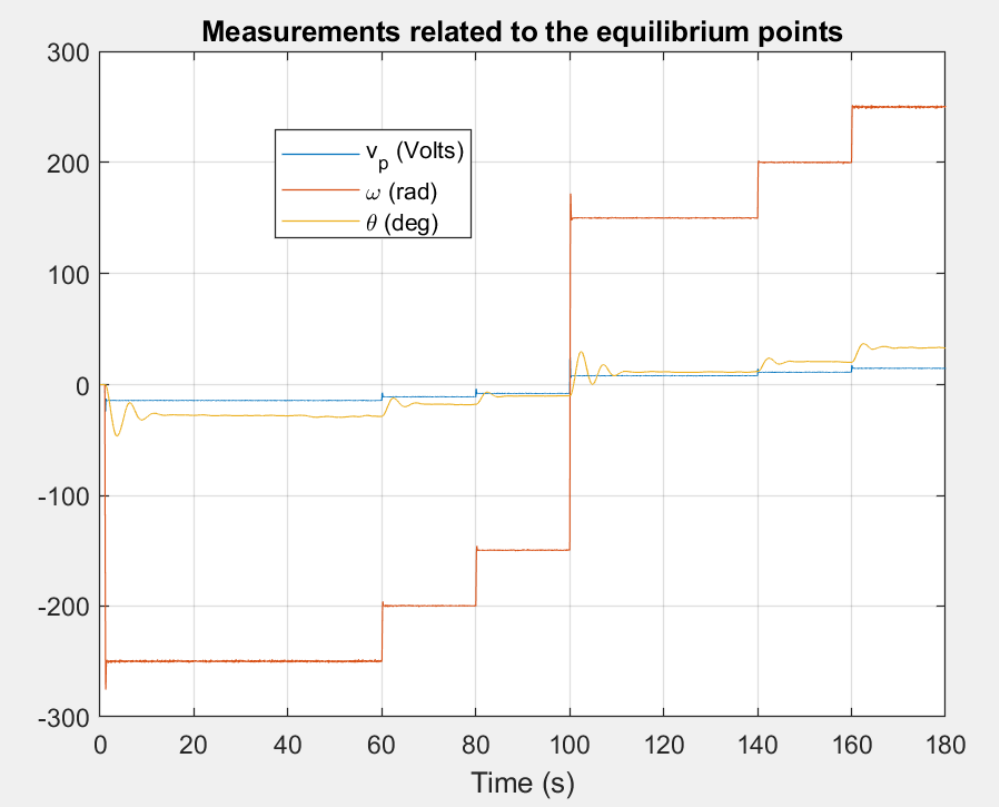


Figure 9: Measures related to the equilibrium points $\omega_h \in \{-250; -200; -150; 150; 200; 250\}$.

Additionaly, values found for $\overline{\theta}$ are $[-27.80, -17.48, -10.24, 11.50, 20.73, 33.36]$.

3.1.4 Obtainging K_t

On this experiment, this approximation will be considered:

$$F_0(\omega_h) = K_t \omega_h \tag{19}$$

So, from equation 18 we have that:

$$\frac{M_b g D_m}{D_t} sin(\bar{\theta}) = K_t \omega_h \tag{20}$$

In this sense, using the linear regression operator in Matlab ($Kt = x \setminus y$), $K_t \approx 0,009$.

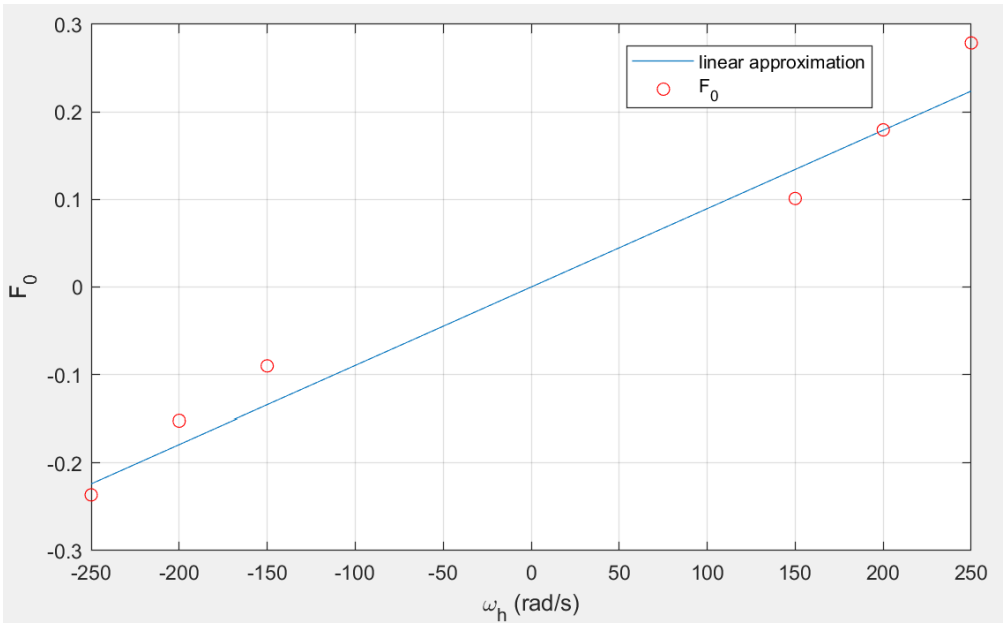


Figure 10: Comparison of experiment and linear approximation.

3.2 Linear Model

3.2.1 Linearization around $\bar{\theta} = 0$

First, isolating $\ddot{\theta}$ from the given equation it is obtained:

$$\ddot{\theta} = \frac{1}{J_p}(F_0 \cdot D_t - D_p\dot{\theta} - M_b \cdot g \cdot D_m sin(\theta)) \tag{21}$$

Then, using the formula expressed in figure 11 to approximate the system linearly around $\bar{\theta} = 0$, it is possible to obtain a linearization of $\ddot{\theta}$ as in equation 22.

$$\dot{x}(t) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\bar{x},\bar{u},t) & \frac{\partial f_1}{\partial x_n}(\bar{x},\bar{u},t) \\ \frac{\partial f_n}{\partial x_1}(\bar{x},\bar{u},t) & \frac{\partial f_n}{\partial x_n}(\bar{x},\bar{u},t) \end{bmatrix} x(t) + \begin{bmatrix} \frac{\partial f_1}{\partial u_1}(\bar{x},\bar{u},t) & \frac{\partial f_1}{\partial u_m}(\bar{x},\bar{u},t) \\ \frac{\partial f_n}{\partial u_1}(\bar{x},\bar{u},t) & \frac{\partial f_n}{\partial u_m}(\bar{x},\bar{u},t) \end{bmatrix} u(t)$$

$$\tilde{y}(t) = \begin{bmatrix} \frac{\partial h_1}{\partial x_1}(\bar{x},\bar{u},t) & \cdots & \frac{\partial h_1}{\partial x_n}(\bar{x},\bar{u},t) \\ \vdots & \ddots & \vdots \\ \frac{\partial h_l}{\partial x_1}(\bar{x},\bar{u},t) & & \frac{\partial h_l}{\partial x_n}(\bar{x},\bar{u},t) \end{bmatrix} x(t) + \begin{bmatrix} \frac{\partial h_1}{\partial u_1}(\bar{x},\bar{u},t) & \cdots & \frac{\partial h_1}{\partial u_m}(\bar{x},\bar{u},t) \\ \vdots & \ddots & \vdots \\ \frac{\partial h_l}{\partial u_1}(\bar{x},\bar{u},t) & & \frac{\partial h_l}{\partial u_m}(\bar{x},\bar{u},t) \end{bmatrix} u(t)$$

Figure 11: First class formula for model's linear approximation.

$$\frac{\partial}{\partial t}\dot{\theta} = \left[\left.\frac{\partial \ddot{\theta}}{\partial \theta}\right|_{\bar{\theta}=0} \quad \left.\frac{\partial \ddot{\theta}}{\partial \dot{\theta}}\right|_{\bar{\theta}=0}\right] \begin{bmatrix}\theta \\ \dot{\theta}\end{bmatrix} + \left.\frac{\partial \ddot{\theta}}{\partial F_0}\right|_{F_0=\bar{F}_0} \cdot F_0 \tag{22}$$

So, solving these formulas the result is:

$$\frac{\partial}{\partial t}\dot{\theta}(t) = \frac{1}{J_p}\left(\begin{bmatrix}-M_b \cdot g \cdot D_m & -D_p\end{bmatrix}\begin{bmatrix}\theta(t) \\ \dot{\theta}(t)\end{bmatrix} + D_t \cdot F_0(t)\right) \tag{23}$$

3.2.2 Approximation to quadratic equation in free response

From the equation 23 we have:

$$\ddot{\theta} + 2\frac{D_p}{J_p}\dot{\theta} + \frac{M_b g D_m}{J_p}\theta = 0 \tag{24}$$

And acording to the standard formulation

$$\ddot{\theta} + 2\xi\omega_0\dot{\theta} + \omega^2\theta = 0 \tag{25}$$

We find the following:

$$\omega_0^2 = \frac{M_b g D_m}{J_p}, \;\; \xi\omega = \frac{D_p}{J_p} \tag{26}$$

3.2.3 Data recording

To record the data in Figure 12, we used a control voltage $v_p = 0$. The system was then set to $\theta = 30°$ and released. The data was gathered until the system it stabilized.

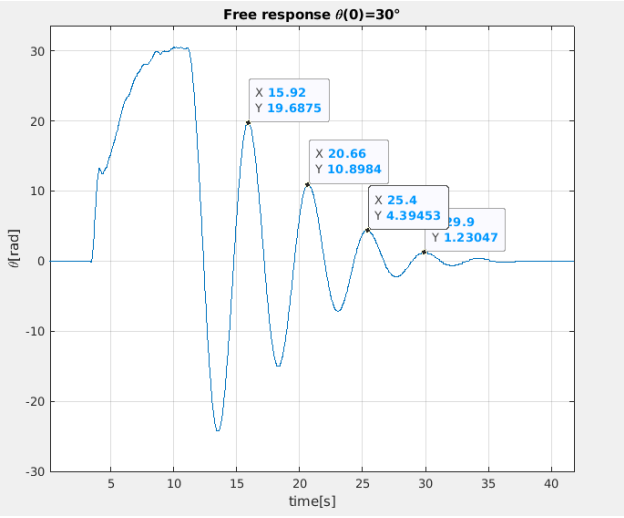


Figure 12: Free response of system iniciating in $\theta = 30°$.

3.2.4 Identification of J_p and D_p

The following code was implemented to identify the solicited values.

```
%% 3.2.4
x0=15.92;
y0=19.6875;
x1=20.66;
y1=10.8984;
x2=25.4;
y2=4.39453;

T = (-x0+x1-x0+x2)/3;
k = 4;
eta = 1/k*log(abs(y0)/abs(y2));
xi = eta/pi/sqrt(1+(eta/pi)^2);
omega0 = 2*pi/(T*sqrt(1-xi^2));

Jp = Mb*g*Dm/omega0^2;
Dp = 2*xi*omega0*Jp;
```

Results:

$$
\begin{gathered}
T = 4,7400 \\
k = 4 \\
\eta = 0,3749 \\
\xi = 0,1185 \\
\omega_0 = 1,3350 \\
J_p = 0,0449 \\
D_p = 0,0142
\end{gathered}
$$

3.2.5 Validation

Starting with the linearization made in 23, it is possible to rewrite the state-space written in 14 as:

$$
\begin{cases}
\frac{\partial}{\partial t}\begin{bmatrix}\theta(t)\\ \dot{\theta}(t)\end{bmatrix} = \begin{bmatrix}0 & 1\\ \frac{-M_b \cdot g \cdot D_m}{J_p} & \frac{-D_p}{J_p}\end{bmatrix}\begin{bmatrix}\theta(t)\\ \dot{\theta}(t)\end{bmatrix} + \begin{bmatrix}0\\ \frac{D_t}{J_p}\end{bmatrix} F_0(t) \\
y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}\theta(t)\\ \dot{\theta}(t)\end{bmatrix}
\end{cases}
\tag{27}
$$

Subsequently, using the recommend Matlab function, the linearization was successfully validated as shown in figure 13.

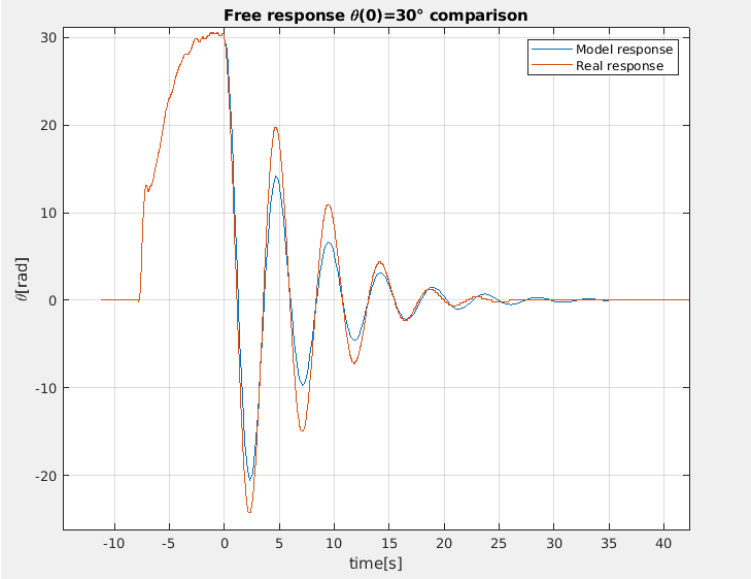


Figure 13: Comparison between the linear model and the real free system response.

3.3 Validation of the global model

3.3.1 Non-linear model simulation

We deployed the given non-linear model in order to simulate a step response using a velocity step of $150 rad/s$. The obtained results are presented in figure 14.

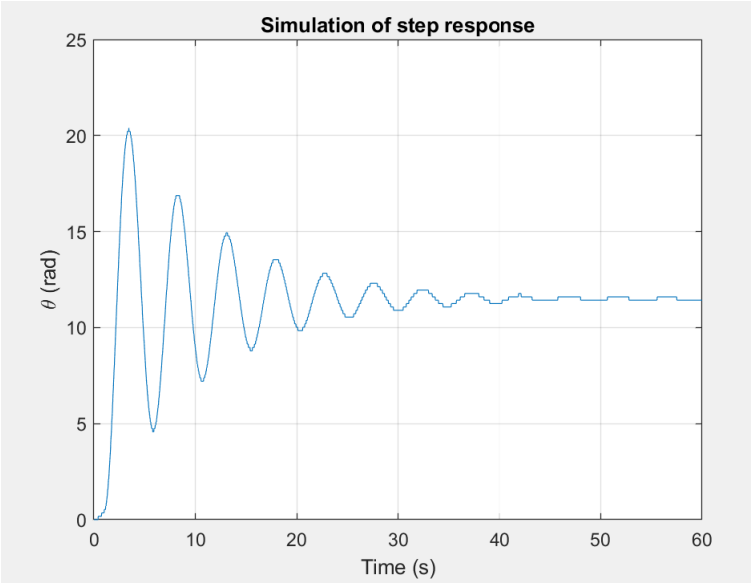


Figure 14: Non-linear model simulation results.

3.3.2 Linear model simulation

We changed the given model in order to simulate a linear model step response using a velocity step of $150 rad/s$. The obtained results are presented in figure 15.

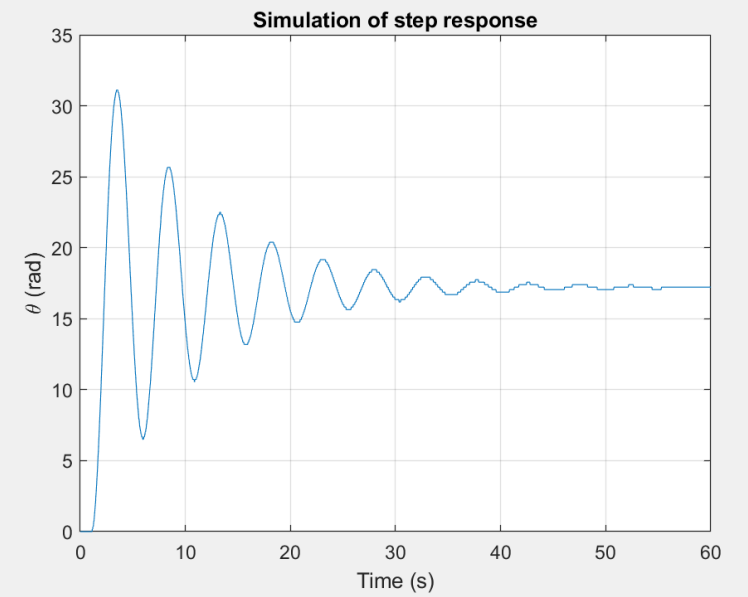


Figure 15: Linear model simulation results.

By combining all the models we've previously structured, we introduce a comprehensive global model (as shown in Figure 16). This model enables us to assess both the step response performance and the input control value, v_p. Observing Figure 14 and 15, it is possible to conclude that the simulation results closely represents the experimental data.

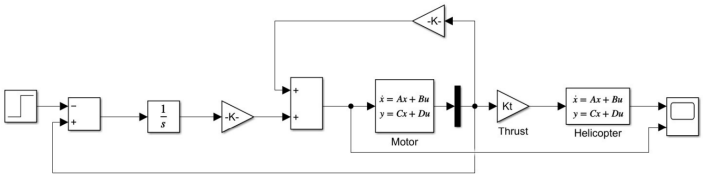


Figure 16: Global model.

4 Middle Work

4.1 Specifications

In this part of the experiment, the goal is to regulate the tilt angle, which can be done by implementing a control system that have 2 main components: Propeller Rotation Speed Servo-Control (ensuring that the propeller's rotation speed is kept at a set value, mitigating disturbances)

and Tilt Behavior Control Strategy (which devises a strategy to determinate the speed set point to propeller rotation servo), optimazing the control of a helicopter's tilt angle.

4.2 Tilt control: a state-space approach

4.2.1 System Linearization

As $F_0(t) = K_{th}(t)$, the linearized system expressed in 27 becomes:

$$\begin{cases} \frac{\partial}{\partial t}\begin{bmatrix}\theta(t)\\ \dot{\theta}(t)\end{bmatrix} = \begin{bmatrix}0 & 1\\ \frac{-M_b \cdot g \cdot D_m}{J_p} & \frac{-D_p}{J_p}\end{bmatrix}\begin{bmatrix}\theta(t)\\ \dot{\theta}(t)\end{bmatrix} + \begin{bmatrix}0\\ \frac{D_t \cdot K_t}{J_p}\end{bmatrix}\omega_H(t) \\ y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}\theta(t)\\ \dot{\theta}(t)\end{bmatrix} \end{cases} \tag{28}$$

$$\Rightarrow A = \begin{bmatrix}0 & 1\\ \frac{-M_b \cdot g \cdot D_m}{J_p} & \frac{-D_p}{J_p}\end{bmatrix},\ B = \begin{bmatrix}0\\ \frac{D_t \cdot K_t}{J_p}\end{bmatrix},\ C = \begin{bmatrix}1 & 0\end{bmatrix} \tag{29}$$

4.2.2 controllability

To check the controllability, the matrix Q_c is created:

$$Qc = \begin{bmatrix}B & A*B\end{bmatrix} = \begin{bmatrix}0 & 0,0031\\ 0,0031 & -0,0010\end{bmatrix} \tag{30}$$

As, rank(Qc) = 2, the system is controllable.

4.2.3 LQR

Because of Matlab features, it was simple to calculate the gain K_lq. The biggest decision in the process was to define the relation between the normalized values of q and r. As a starting point, it was decided to use $\frac{q}{r} = 7$

$K_{lq} = \begin{bmatrix}306,8564 & 337,4107\end{bmatrix}$

After that, as defined in course material, $M = -\left[C(A-BK)^{-1}B\right]^{-1}$. So,

$M_{lq} = 813,8099$

With these values, that result was obtained in simulation:

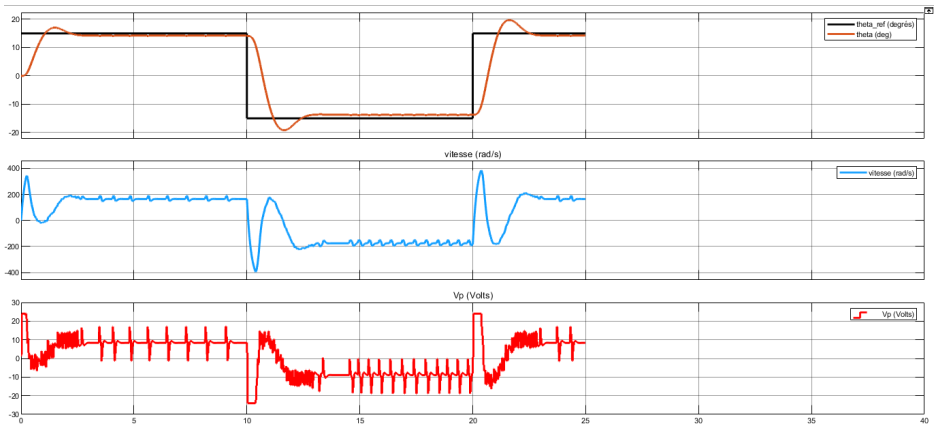


Figure 17: Simulation LQR $T_{e1} = 0.5ms$.

During the simulator test, it could be possible to note that it is not possible to attempt to the specifications. Firstly, the steady-state error depends on the θ_{ref} desired and is not null for all. Secondly, the over shoot and the rising time are attempted ideally if $\frac{q}{r}$ is very high, but obtaining this values is not possible because of v_p saturation in 24 voltz.

4.2.4 LQI

Adding the new state variable, the system 28 becomes as 31.

$$\begin{cases} \frac{\partial}{\partial t}\begin{bmatrix}\theta(t)\\ \dot{\theta}(t)\\ z_i\end{bmatrix} = \begin{bmatrix}0 & 1 & 0\\ \frac{-M_b \cdot g \cdot D_m}{J_p} & \frac{-D_p}{J_p} & 0\\ 1 & 0 & 0\end{bmatrix}\begin{bmatrix}\theta(t)\\ \dot{\theta}(t)\\ z_i\end{bmatrix} + \begin{bmatrix}0\\ \frac{D_t \cdot K_t}{J_p}\\ 0\end{bmatrix}\omega_H(t) + \begin{bmatrix}0\\ 0\\ -1\end{bmatrix}\theta_{ref}(t) + B' \cdot d \\ y = \begin{bmatrix}1 & 0 & 0\end{bmatrix}\begin{bmatrix}\theta(t)\\ \dot{\theta}(t)\\ z_i\end{bmatrix} \end{cases} \tag{31}$$

Then, choosing the normalized values of q, r and z as $\frac{q}{r} = 4$ and $\frac{q}{r} = 1$, it was simple to obtain the gains values to the system 31 with Matlab features.

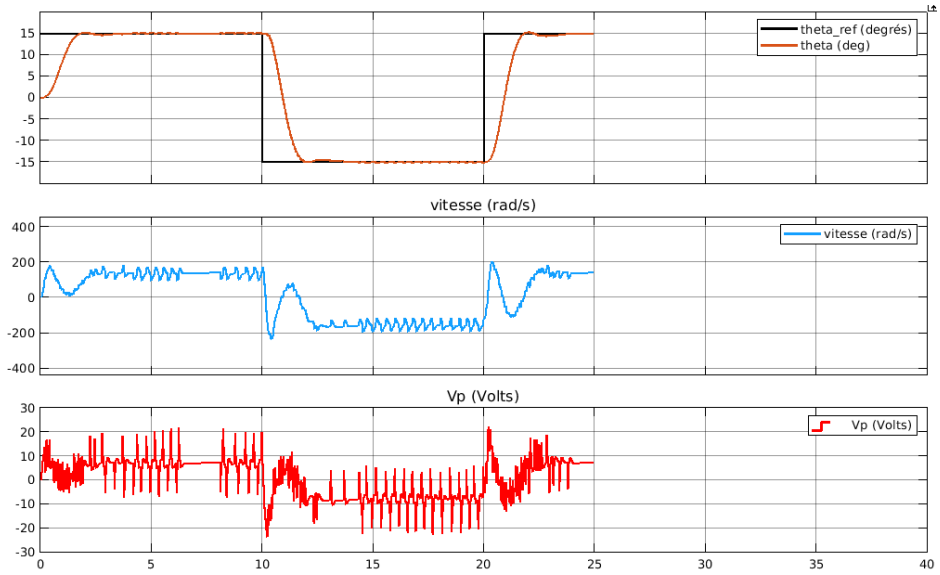


Figure 18: Simulation LQI $T_{e1} = 0.5ms$.

Then, steady state error and overshoot were achieved. However, the rising time was not because it would be needed a highter maximum power. The saturated V_p value was in 24 voltz, and it would be needed highter to reach this goal.

5 Helicopter LQ experiments

5.1 LQR

After using the LQR control system on the AERO physical model, the factor $\frac{q}{r} = 14$ was more effective and there are the results:

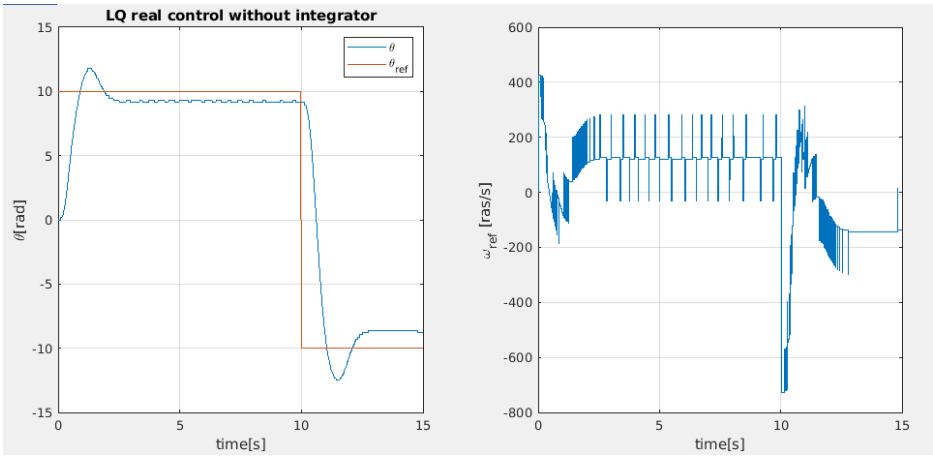


Figure 19: Results of LQR control in AERO system.

As predicted in simulation of 4.2.3, this control strategy do not result in a steady state error and, so, do not attempt the specifications. Because of that, it is necessary to use the LQI control system, as shown in the next subsection.

Just to verify the simulation predictability, the comparison between both with the same parameters, is plotted below.

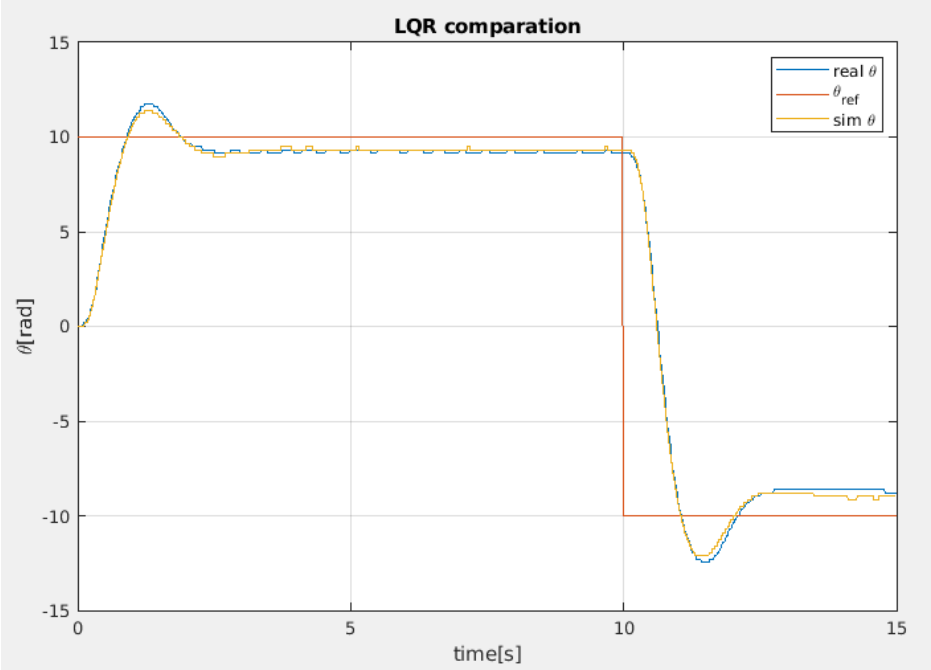


Figure 20: Comparison of LQR control between physical system and simulation.

After this comparison, it is concluded that the non-linear simulation is very close to the real system, but there are some little differences, for example the overshoot.

5.2 LQI

In order to add an integral action, it was necessary to add an integrator and change the system's gains, as the figure 21

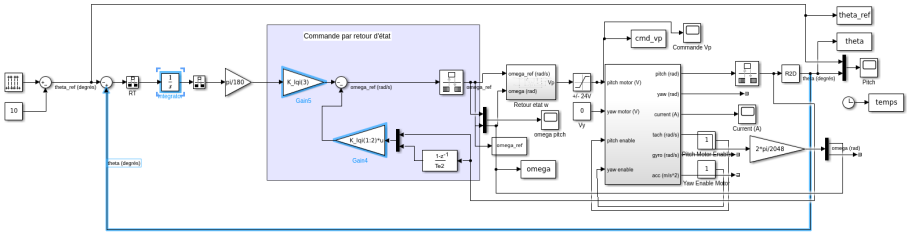


Figure 21: Simulink's structure implemented with integral action.

With these modifications, the result of control action was captured in figure 22. Considering q, z and r as the normalized weights of $X(t)$, $z(t)$ and $u(t)$, the group decided to use $q/r = 10$ and $z/r = 4$, because there were fewer measure's noising and it was desired to reach closer to the rising time aimed.

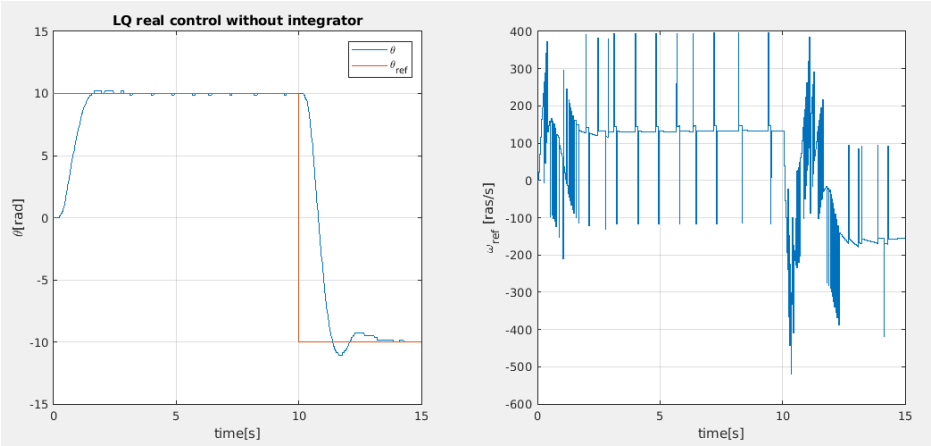


Figure 22: Results of control action on the physical system.

It is noted that the overshoot and the steady state error conditions were finally attempted. The only criteria imposed and not reached was the rising time, because of system power limitations.

To verify the accuracy of the simulation model, the figure 23 was created. There are some differences, but the mean comportment is very close.

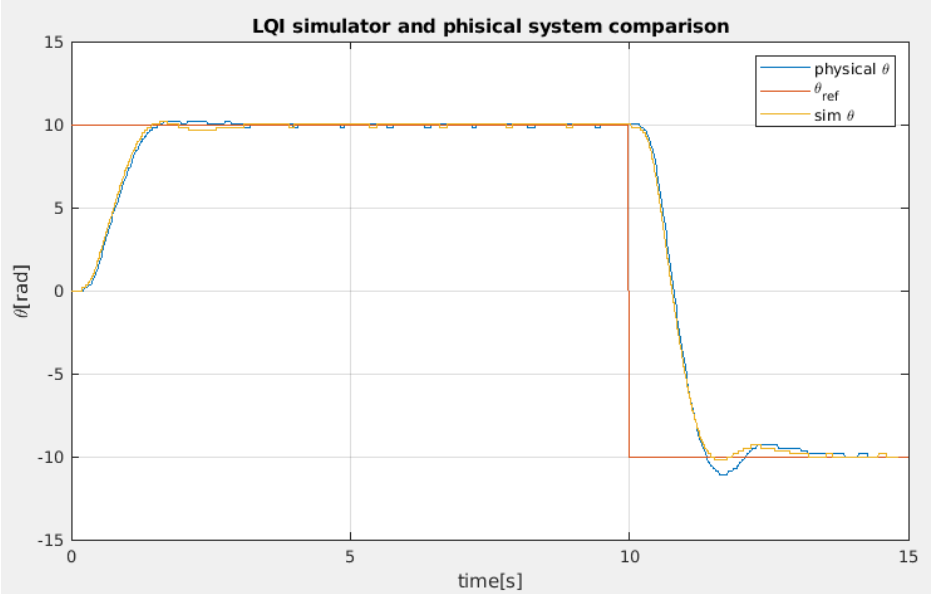


Figure 23: Comparison between simulation and physical results.

At the end, in figure 24 shows the difference of control LQR and LQI. The rising time is pretty much the same, because both of them are limited with the same power saturation. Although, the integrator corrects the results in terms of overshoot and steady state error.

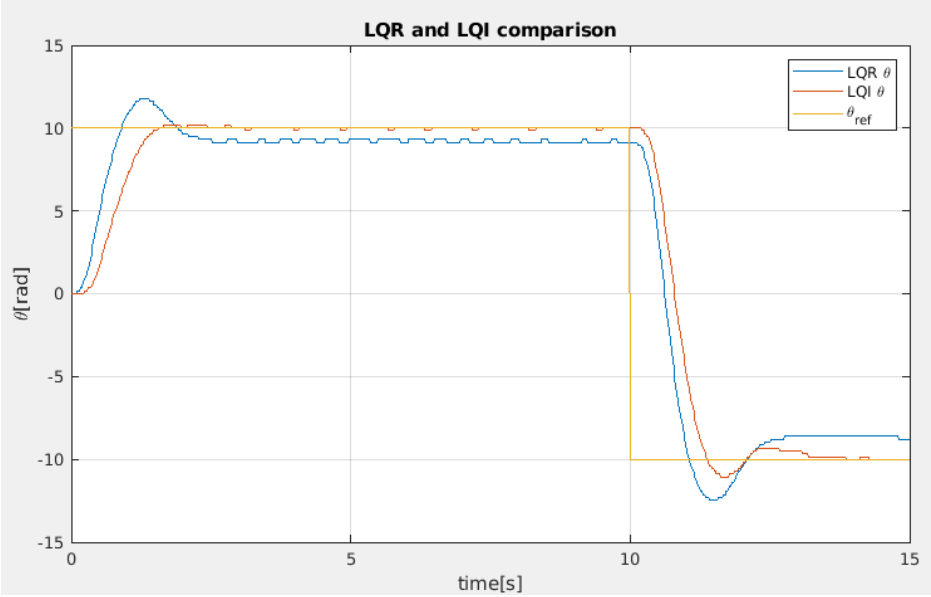


Figure 24: Comparison between LQR and LQI results.

6 Observers addition

6.1 Observer Synthesis

6.1.1 Literal Equations

Considering

$$\begin{aligned} \dot{x} &= \begin{bmatrix} \hat{\theta} \\ \dot{\hat{\theta}} \end{bmatrix} \\ \hat{y} &= C\hat{x} \end{aligned} \tag{32}$$

it is just necessary to make a substitution in the observer's state equation:

$$\begin{aligned} \dot{\hat{\theta}} &= A\hat{x} + Bu + L(y - \hat{y}) \\ &= A\hat{x} + Bu + Ly - LC\hat{x} \\ \Rightarrow \dot{\hat{\theta}} &= (A - LC)\hat{x} + Bu + Ly \end{aligned} \tag{33}$$

6.1.2 Justification of the choice of eigenvalues

In order to understand the choice made, it is possible to calculate the the real part of observer eigenvalues:

$$Real\{\lambda_1\} = Real\{\lambda_2\} = -\omega_0\xi_0 = -13,5 \tag{34}$$

In addition, the real part of feedback control gain's eigenvalue is actually: -1.7282 (calculated via Matlab features from the matrix $A - BK$).

In this way, it is plausible to confirm that the observer could be applied to the feedback control, because of the value of the observer dynamic that is almost eight times the speed of the feedback control.

6.1.3 Matrix L Calculation

The code used on Matlab is the following one:

```
% poles
xi2 = 0.9;
w02 = 15;
p = [w02*(-xi2-1i*sqrt(1-xi2^2)) w02*(-xi2+1i*sqrt(1-xi2^2))]';

% Observateur Matrix
L = place(Asys.', Csys.', p).'; % Gain de l'observateur, vecteur
    (1x2) a calculer avec "place"
```

Then, the result was this matrix:

$$L = \begin{bmatrix} 26.6836 \\ 214.7758 \end{bmatrix} \tag{35}$$

6.1.4 Responses of the output and the control to a set-point step

The idea, at first, is to construct a state representation with observer, but no integrator. So, without feedback, that is the system:

$$\begin{cases} \begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & 0 \\ LC & A - LC \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} B \\ B \end{bmatrix} u(t) \\ y = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} \end{cases} \tag{36}$$

After, there is the control law:

$$u(t) = \begin{bmatrix} 0 & -K \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + M\theta_{ref} \tag{37}$$

By those equations, it is able to formulate the closed system on this states' equation:

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & -BK \\ LC & A - LC - BK \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} B \\ B \end{bmatrix} M\theta_{\text{ref}} \tag{38}$$

In figure 25, there is a step input response and the corresponding control value. The output θ perform as expected.

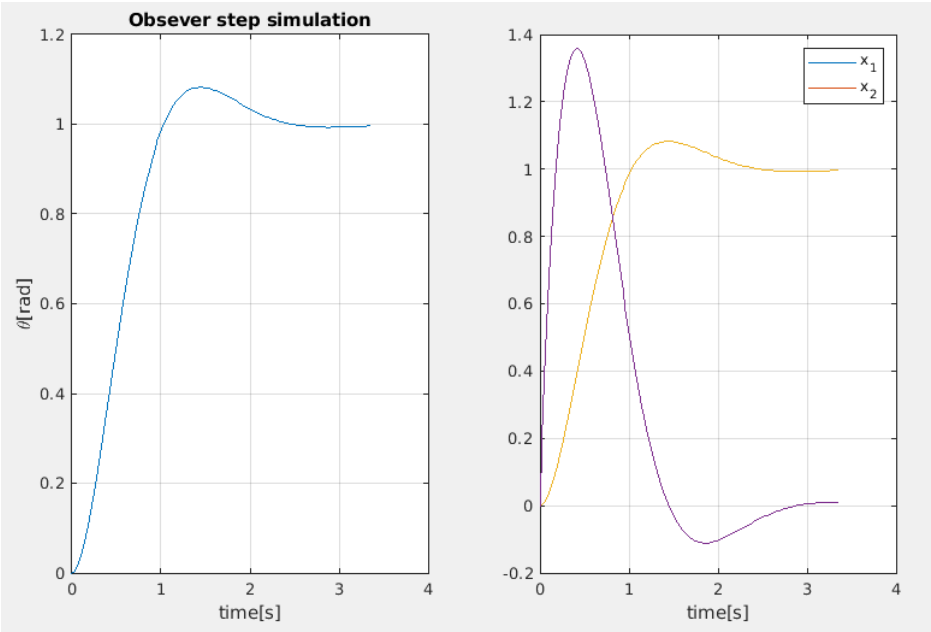


Figure 25: θ and ω_h in simulation with only observer

6.1.5 Complete simulation

To verify the control model using observers more accurately, a simulation with the nonlinear system presented below was carried out. This allowed for a better estimation of the plant's behavior and the assessment of the presence of power limitations due to v_p saturations.

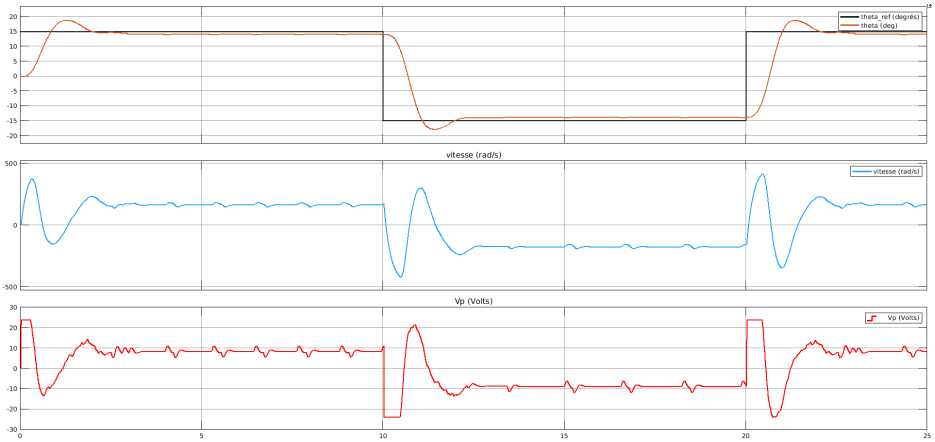


Figure 26: θ and ω_h in experiment with only observer

6.2 Command LQR with Observer

With the assistance of the provided file, data collection was performed for the same LQR model, but now with the presence of an observer.

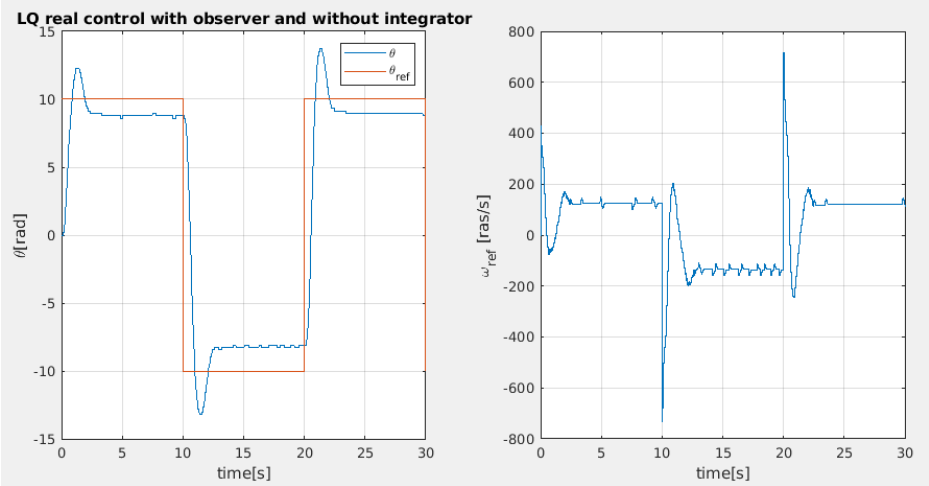


Figure 27: LQ real control with observer and without integrator

The results were very close to what had already been obtained in the LQR without an integrator. Therefore, to more precisely verify the difference in the presence of the observer, both sets of results were plotted together.

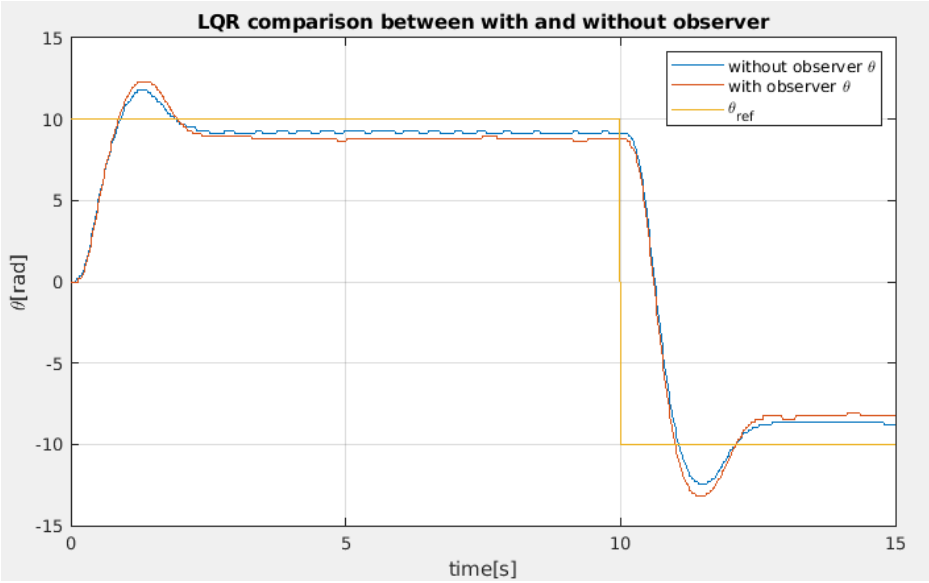


Figure 28: LQR comparison between with and without observer

In this graphic, it is possible to confirm that for $\theta_{ref} = 10$, the case with observer came closer to

the reference than the one without observer. This happens because the observer approximates the states, what would actually help in case of having a lot of noise (or lac of information). However, it was not a problem on this situation, because the interferences were not that high.

6.3 Command LQI with Observer

An integral action was added to the system, that was already with the observer. So, it was necessary just changing the gains values and andding the difference-integral action.

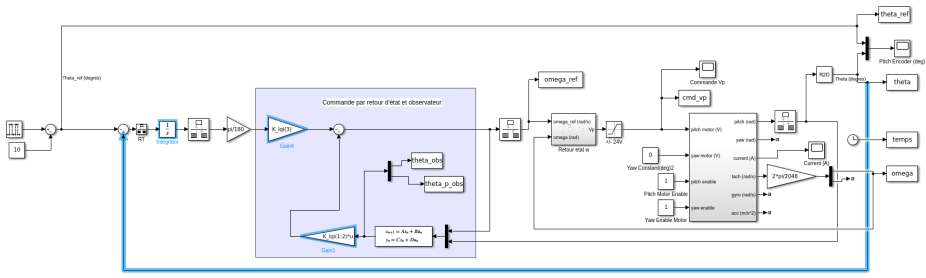


Figure 29: Simulink controller model with observer and integral action

After the simulation, the plot was made, having the following results:

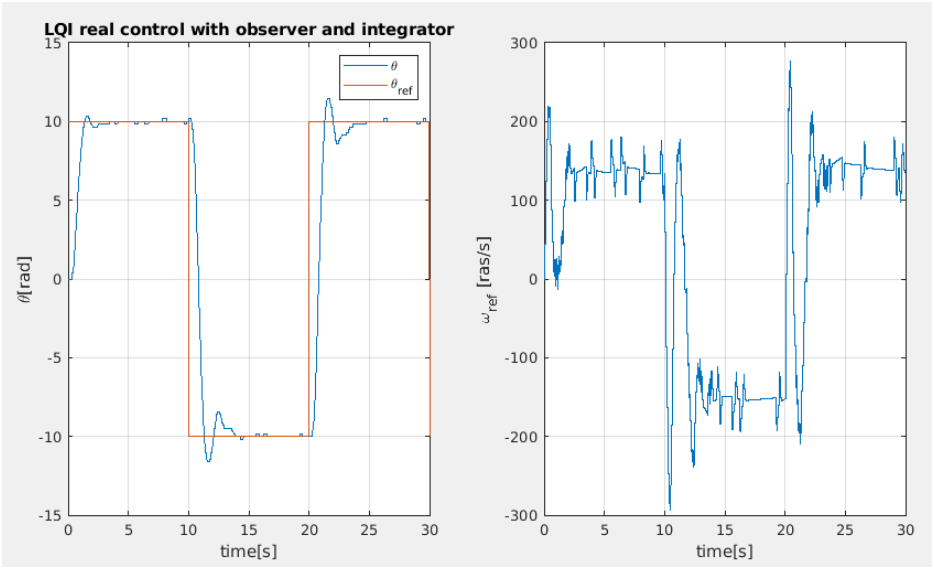


Figure 30: LQI real control with observer and integrator

In order to analise the results, in the figure 31 has a plot of the results of θ of the last experiments.

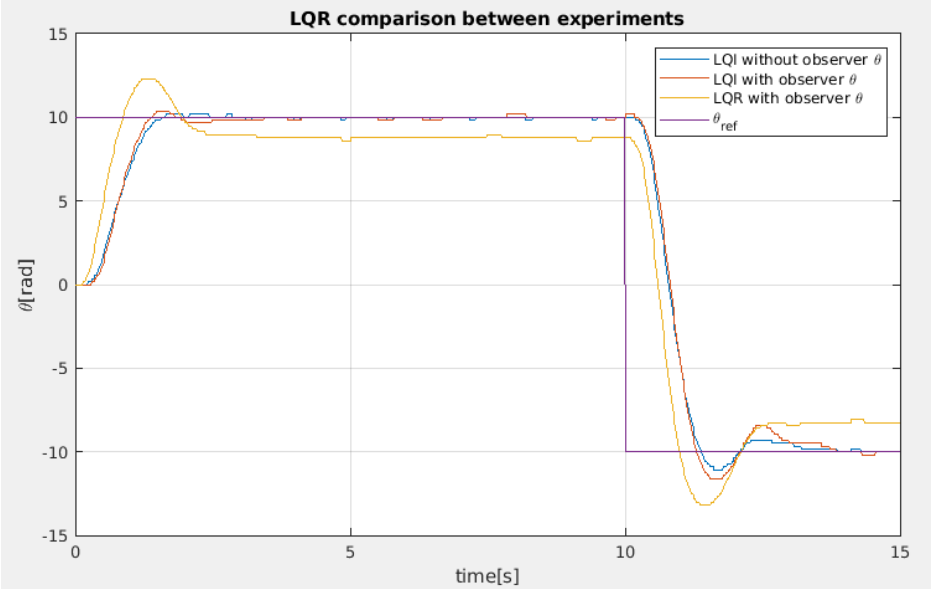


Figure 31: LQR comparison between LQI without and with observer and LQR with observer

It is observed that after the addition of an integrate integral action, the state error and the overshoot were reduced. The addition of an observer in the system was negative because the internal data was already acquirable without estimation. Although, the existence of integrator is always better to the results.

6.4 Analysis of stability margins

6.4.1 LQR model

First, it is necessary to obtain T_{OL}:

$$T_{OL} = \frac{R(s)}{U(s)} = K_{lq}X(s) \cdot X^{-1}(s)(sI-)^{-1}B = K_{lq}(sI - A)^{-1}B \tag{39}$$

Then, using Matlab features, was possible to plot the Bode Diagram and the Nyquist Diagram. The Nyquist one, figure 32, shows that there is no gain margin, because the system is stable what ever the gain's addiction. Furthermore, the bode's diagram, figure 33, helps us to verify the same sentence about the gain's margin and to obtain the phase margin: 70°.

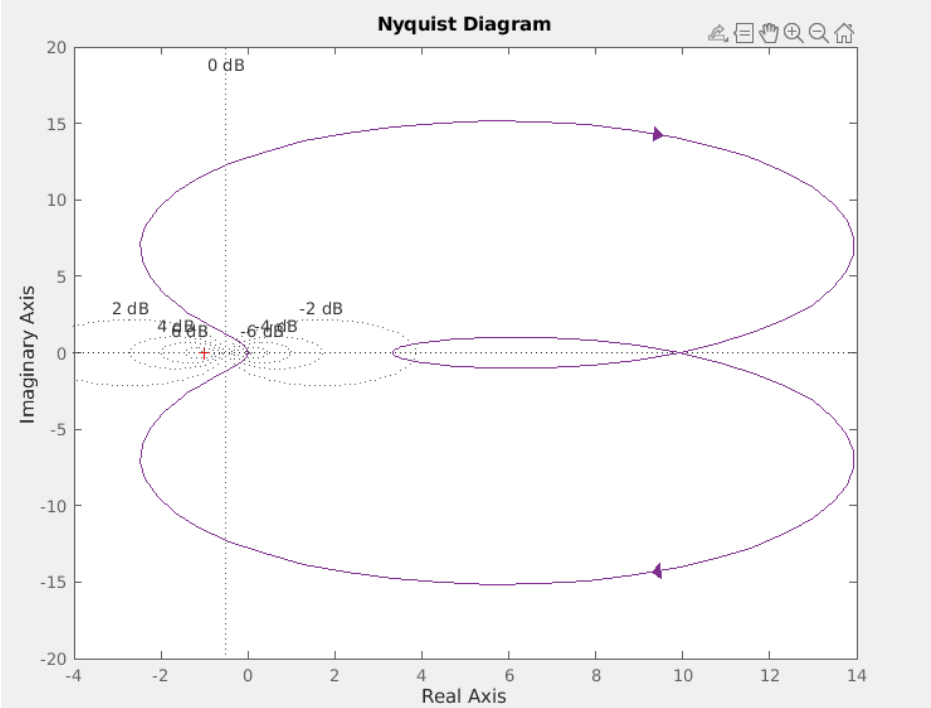


Figure 32: Nyquist diagram of LQR model

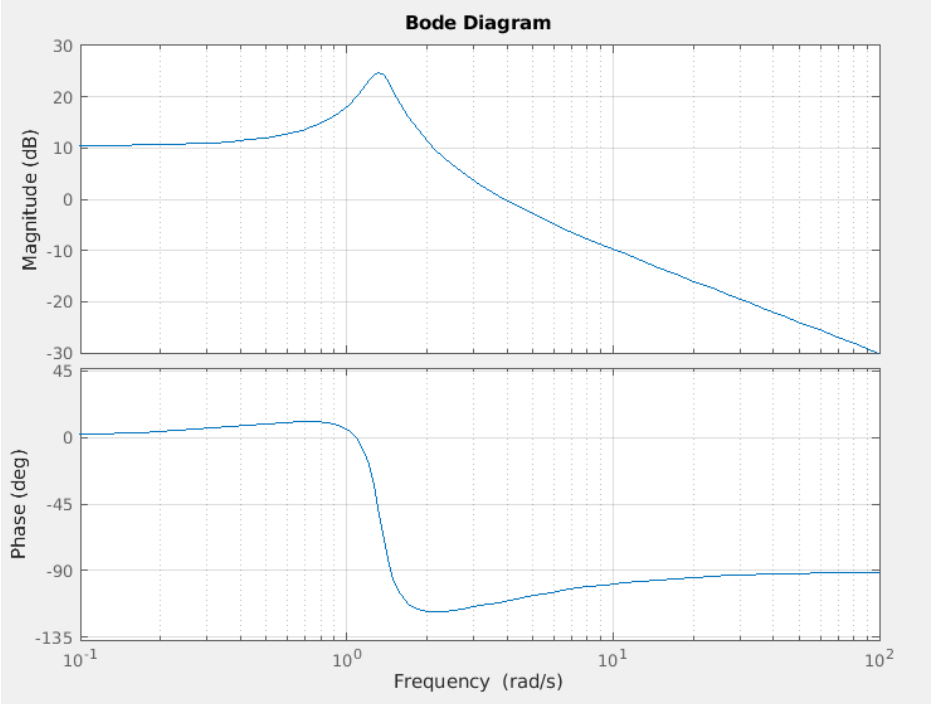


Figure 33: Bode diagram of LQR model

6.4.2 Observed model

Analogy,

$$T_{OL} = \frac{R(s)}{U(s)} = K_a(sI - A_a)^{-1}B_a \tag{40}$$

From the equation system 38:

$$\begin{aligned} A_a &= \begin{bmatrix} A & -BK_{lq} \\ LC & A - LC - BK_{lq} \end{bmatrix} \\ B_a &= \begin{bmatrix} B \\ B \end{bmatrix} \\ K_a &= \begin{bmatrix} 0 \\ 0 \\ K_{lq} \end{bmatrix} \end{aligned} \tag{41}$$

So again, using Matlab features, that is visible in Nyquist Diagram, figure 34 that the margin gain is infinite. But, now, the phase margin is bigger: about 90°, as visible in the Bode diagram, figure 35.

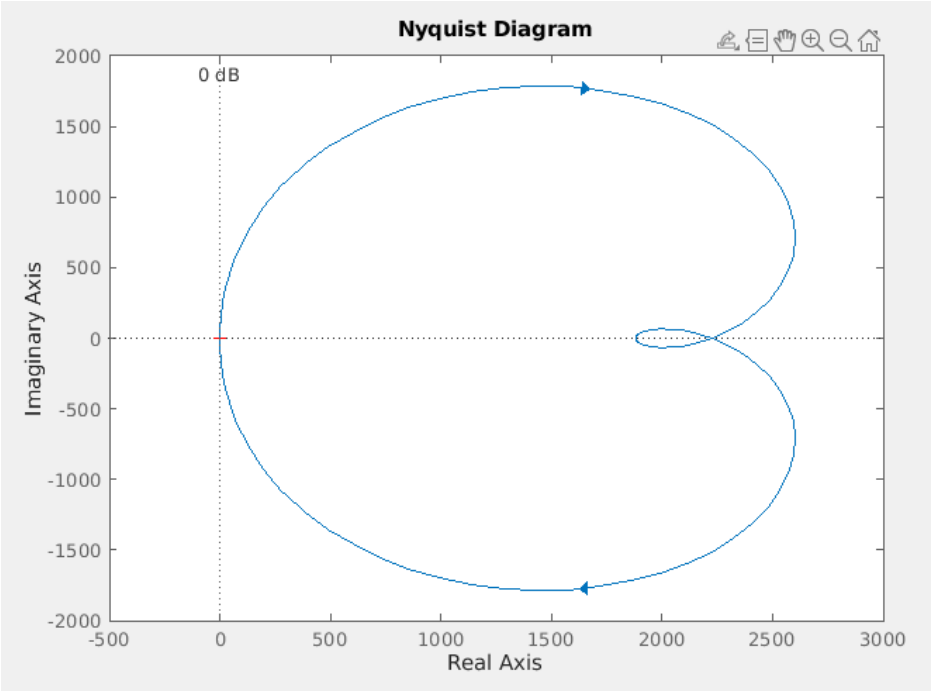


Figure 34: Nyquist diagram of LQR observed model

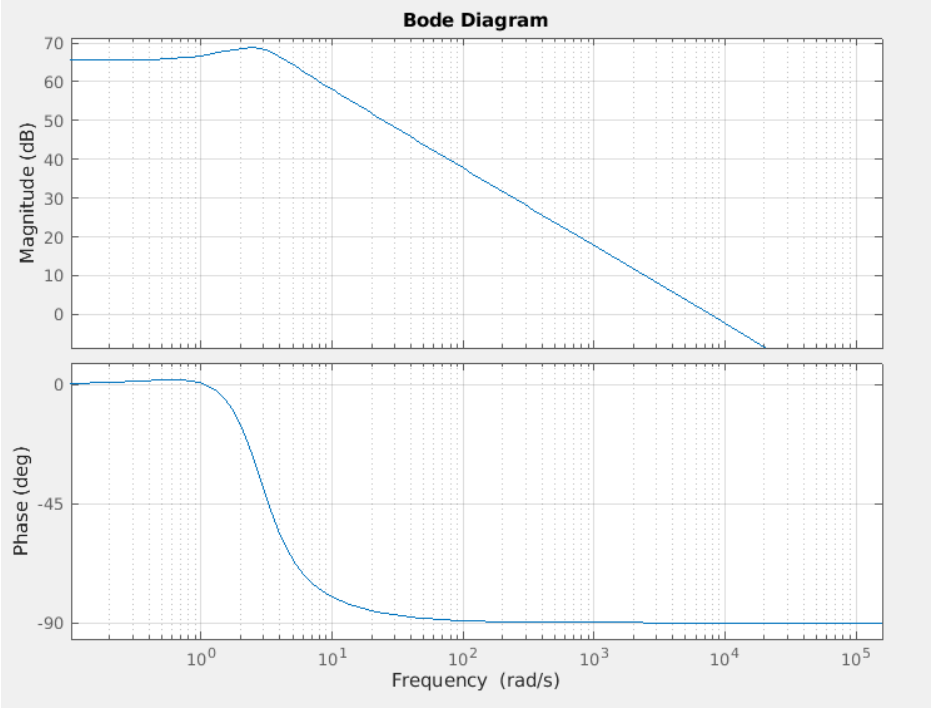


Figure 35: Bode diagram of LQR observed model

7 Conclusion

Throughout this laboratory work, we have undertaken an exploration of the control system for an experimental helicopter platform. Our work began with the study of the motor drive, where we conducted a comprehensive analysis of the voltage step response to identify the parameters of the motor's speed transfer function. This phase revealed the challenge posed by the non-linear gain of the system. To address this, it was designed a state feedback controller with integral action using the pole placement technique, resulting in the meeting of the desired specifications of alignment between experimental and simulated results and rapidity for regulating the motor's speed.

Additionally, tilt motion parameters were identified by conducting a comprehensive study of the equilibrium and dynamics of the system. This phase enabled us to establish a vital relationship between thrust force and motor speed and led to the identification of the remaining parameters in the system's model. The final model was validated through a series of experimental tests, confirming its accuracy.

Furthermore, the state-space control design brought forth the development of a Linear Quadratic Regulator (LQR) for reference angle tracking. In this sense, we implemented controllers with and without integral action, highlighting their capabilities through detailed simulations. Significantly, the inclusion of integral action played a essential role in eliminating steady-state errors and ensuring the system's performance.

In conclusion, our laboratory work underscores our adeptness in modeling, validating, and controlling the experimental helicopter platform. This report represents the application of principles

learned the lectures.

A "initCommande_AERO.m"

```
%
    ------------------------------------------------------------------------

%% Initialisation des parametres de la plateforme AERO
%
    ------------------------------------------------------------------------


clear all    % Effade toutes les varaibles de la memoire
close all    % Fermeture de toutes les figures

%
    ------------------------------------------------------------------------

% Quanser Aero Parameters
%
    ------------------------------------------------------------------------

Mb = 1.15;% Masse du corps rigide
Dm = 0.0071; %Distance du centre de gravite a l'axe de rotation
Dt = 0.158; % Distance de la poussee a l'axe de rotation
g = 9.81;

%
    ------------------------------------------------------------------------

%% Boucle de vitesse
%
    ------------------------------------------------------------------------


% Periode d'echantillonnage boucle de vitesse
Te1 = 0.002; % 2ms

% Parulametres Kv et tau
Kv = 19.5939;
tau = 0.18537;

% Representation d'etat de la chaine de motorisation xa = [wh z]';
Am = [-1/tau 0; 1 0];
B1m = [Kv/tau ; 0];
B2m = [0 ; -1];
Bm = [B1m, B2m];
Cm = [1 0];
```

```
33
34 % Retour d'etat (2.2.1)
35 xi = 0.7;
36 w0 = 22;
37 p = [w0*(-xi-1i*sqrt(1-xi^2)) w0*(-xi+1i*sqrt(1-xi^2))]';
38
39 Ka = place(Am, B1m, p); % Gain du retour d'etat, vecteur (1x2) a
       calculer avec "place"
40 Kw = Ka(1); % Gain pour la vitesse, x1 = wh
41 Kz = Ka(2); % Gain pour l'action integrale, w2 = z
42
43 sysm=ss(Am-B1m*Ka,Bm,Cm,[0 0]);
44 ym=step(sysm);
45
46 %
       ------------------------------------------------------------------------

47 %% Identification des parametres du mouvement en inclinaison
48 %
       ------------------------------------------------------------------------

49 omega_h = [-250 -200 -150 150 200 250]; % position theta en radians
50 theta_h = [-27.7964 -17.4788 -10.2367 11.5001 20.7327 33.3521]; %
       position theta en radians a completer
51 F0_h = Mb*g*Dm/Dt*sin(theta_h*pi/180); % a completer
52 Kt =  omega_h.' \ F0_h.'; % a completer
53
54 %
       ------------------------------------------------------------------------

55 % Linearisation et determination de Jp et Dp
56 %
       ------------------------------------------------------------------------

57 %% 3.2.4
58 x0=15.92;
59 y0=19.6875;
60 x1=20.66;
61 y1=10.8984;
62 x2=25.4;
63 y2=4.39453;
64
65 T = (-x0+x1-x0+x2)/3;
66 k = 4;
67 eta = 1/k*log(abs(y0)/abs(y2));
```

```
xi = eta/pi/sqrt(1+(eta/pi)^2);
omega0 = 2*pi/(T*sqrt(1-xi^2));

Jp = Mb*g*Dm/omega0^2;
Dp = 2*xi*omega0*Jp;

%
    ------------------------------------------------------------------------

% Modele Lineaire
%
    ------------------------------------------------------------------------

A = [0 1;-Mb*g*Dm/Jp -Dp/Jp];
B = [0; Dt*Kt/Jp];
C = [1 0];
D = 0;
sys = ss(A, B, C, D);

%
    ------------------------------------------------------------------------

% Commande LQ
%
    ------------------------------------------------------------------------

Te2 = 0.02; % 20ms
%Te1 = 0.05;
% intermediaires
theta_max = pi/4;
omega_h_max=500;
q= 14; % define here the weight
Q=C.'*q/theta_max^2*C;
R=1/omega_h_max^2;

% retour d'etat
K_lq = lqr(A,B,Q,R);  % Gain du retour d'etat, vecteur (1x2)
M_lq = -(C*(A-B*K_lq)^-1*B)^-1;  % Gain pour la reference (1x1)

% cleaning space
clear theta_max
clear omega_h_max
clear zi_max
clear q
clear Q
```

```
clear R

%
    ------------------------------------------------------------------------

% Commande LQI
%
    ------------------------------------------------------------------------

%% steady state
Aa = [A [0;0];1 0 0];
B1a = [B;0];
B2a = [0;0;-1];
Ca = [1 0 0;0 0 1];
Da = [0;0];

% intermediaires
theta_max = pi/4;
omega_h_max=500;
zi_max = 0.15;
q=10; % define here the weight
z=4; % define here the weight
r=1; % define here the weight
Qn=[1/theta_max 0;0 1/zi_max];
Q=Ca.'*Qn*[q 0;0 z]*Qn*Ca;
R=r/omega_h_max^2;

K_lqi = lqr(Aa, B1a, Q, R);  % Gain du retour d'etat avec action
    integrale, vecteur (1x3)

% cleaning space
clear theta_max
clear omega_h_max
clear zi_max
clear q z r
clear Q Qn
clear R

%
    ------------------------------------------------------------------------

% Observateur
%
    ------------------------------------------------------------------------
```

```
141
142 % poles
143 xi2 = 0.9;
144 w02 = 15;
145 p = [w02*(-xi2-1i*sqrt(1-xi2^2)) w02*(-xi2+1i*sqrt(1-xi2^2))]';
146
147 %Modele
148 Asys = A;% Matrice A modele lineaire (dynamique)
149 Bsys = B;% Matrice B modele lineaire (de Commande)
150 Csys = C;% Matrice C du modele lineaire (de Sortie)
151
152 % Observateur Matrix
153 L = place(Asys.', Csys.', p).'; % Gain de l'observateur, vecteur (1x2
      ) a calculer avec "place"
154
155 Ao = [A -B*K_lq; L*C A-L*C-B*K_lq];
156 Bo = [B;B]* M_lq;
157 Co = [C 0 0];
158 Do = 0;
159 syso = ss(Ao, Bo, Co, Do);
160
161 %
      ------------------------------------------------------------------------

162 % Correcteur PID
163 %
      ------------------------------------------------------------------------

164 Kp = 1;% Action proportionnelle
165 Ti = 1; % Action integrale
166 Td = 1; % Action derivee
167 tau_d = 1 ;% Filtage de l'action derivee
```