CentraleSupélec

CentraleSupélec
Département Automatique

Laboratory work - Control Theory course

Helicopter tilt control

Preliminary remark: **this study includes work to be done, outside the class-hours, between the two sessions scheduled in the timetable. It is essential to treat this part seriously in order to complete the work requested during the second session. The preparation will be checked individually by the supervisors at the beginning of the second session.**

The report, which must covers both sessions as well as the preparation between the two sessions, must be returned 7 days after the end of the last session. It must be submitted in a PDF file through your cohort EDUNAO page.

Context and description of the physical system

This laboratory work deals with the stabilization of a platform with two degrees of freedom. Although studied in an academic context, this type of system can model the problems posed by the stabilization or control of various physical processes such as a helicopter rotor plane, a drone, etc. (Figure 1).

Figure 1: Stabilization of an aerodynamic platform

The system used in this study is built on (Figure 2):

- a platform driven by two propellers allowing to control the yaw and tilt angles of the platform plane;
- two motor-based actuation chains allowing the speed of the propellers to be controlled to a reference value;
- a set of sensors to measure propeller rotational speed, yaw angles, bank angles and inertial measurements (speed and acceleration) using a dedicated unit.

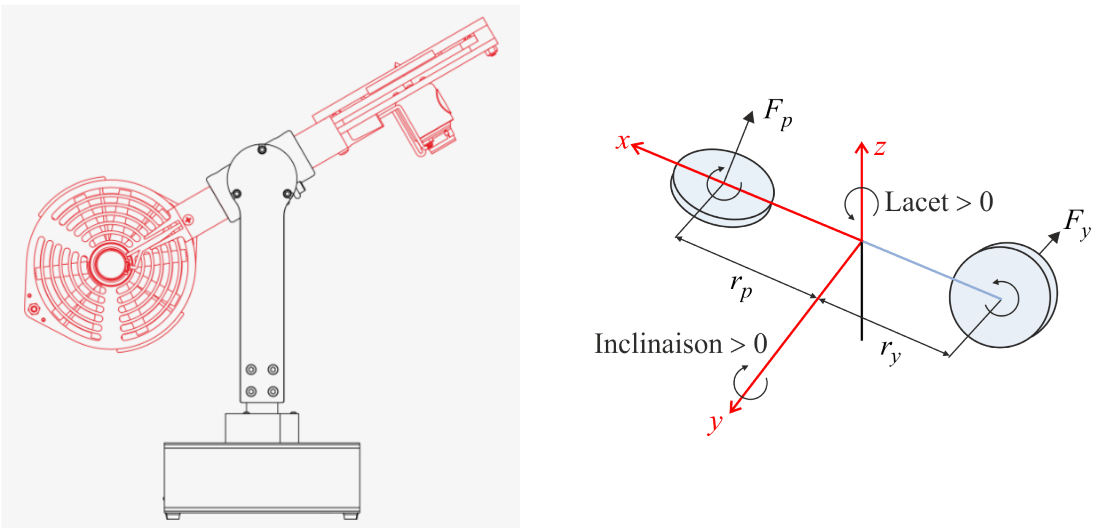


Figure 2: Two degrees of freedom Quanser Aero platform

In this study, only the tilt angle of the platform is considered. The yaw angle is blocked in order to obtain a mono-variable system. The objective is to determine the control laws allowing to stabilize the platform at a reference tilt angle and to compare the performances. For example, in the case of a drone, $\theta = 0$ corresponds to a stationary flight and $\theta = \theta_0 \neq 0$ corresponds to a displacement in the xy plane.

The control signal here is the voltage at the input of the motor driving the propeller.

In the following we denote:

- θ the tilt angle (rotation around $\vec{y}$),
- ω_h the rotation speed of the propeller,
- v_p the control voltage of the motor acting on the tilt.

The present study is organized along several axes of work distributed in two sessions of three hours with the objective of designing a control loop for the tilt.

In the first session, you will identify the parameters of the model relating the tilt motor control voltage and the tilt angle θ, analyze the dynamic characteristics of the open-loop system and design a state feedback controller to adjust the propeller rotation speed.

The second session deals with the control of the tilt angle towards a reference value by means of a Linear Quadratic optimal control law and by introducing an observer. A preparatory work is required between the first and the second session.

During the two laboratory work sessions it is important to test the control laws obtained through simulation with the model developed during the first session before effectively deploying it on the platform.

1 Modeling

The files necessary for the work are available on the Desktop in the directory "**Fichiers Elèves2A_AERO**".

1.1 Description of the tilt motion

Figure 3 shows the scheme used for modeling the tilt dynamics of the platform. The voltage v_p applied to the motor lead the propeller motor to rotate at speed ω_h. This rotation speed produces a thrust force F_0. The relationship between the rotation speed and the force F_0 is non-linear and will be identified experimentally. Finally, the force F_0 produces a torque that will tilt the platform.

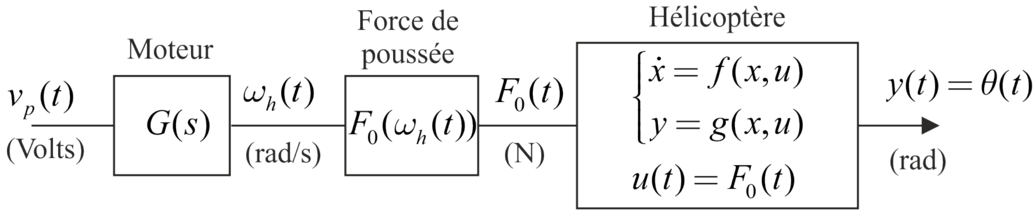


Figure 3: Block diagram of the tilt dynamics

1.2 A priori knowledge - Modeling

In order to determine adequate control laws to stabilize the platform, it is first necessary to have a dynamic model of the system. The figure 4 shows the tilting torques exerted by the weight and the propeller drive.

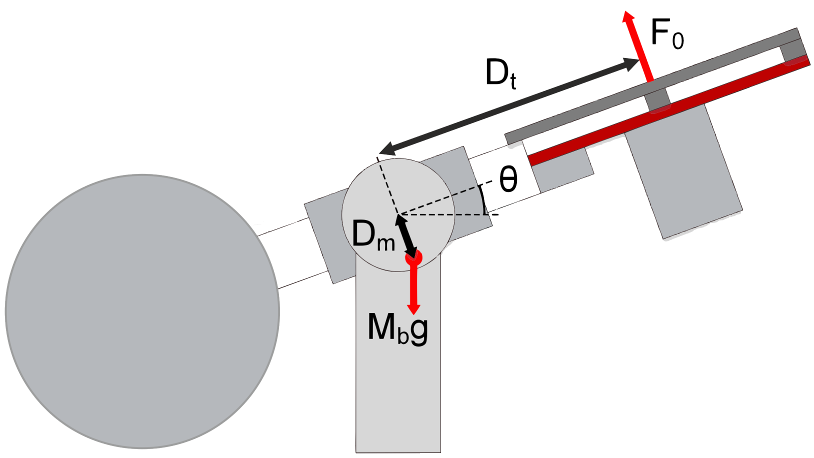


Figure 4: Diagram for the platform tilt

With the fundamental law of dynamics, and considering a viscous friction in the rotation, the motion in tilt is described by:

$$J_p\ddot{\theta}(t) = F_0(\omega_h(t))D_t - D_p\dot{\theta}(t) - M_b g D_m \sin\theta(t), \tag{1}$$

where:

- J_p is the moment of inertia about the tilt axis,

- D_p is the coefficient of viscous friction on the tilt axis,
- M_b is the mass of the rigid body subject to tilt,
- D_t is the distance of the thrust from the axis of rotation,
- D_m is the distance from the center of gravity to the axis of rotation.

A part of the model parameters is known. The numerical values are given in the table 1.

Table 1: Nominal numerical values of the model parameters

Symbol	Description	Value
DC motor		
V_{nom}	Nominal voltage	18 V
τ_{nom}	Rated torque	$22 \cdot 10^{-3}$ Nm
ω_{nom}	Rated speed	3050 rpm
I_{nom}	Rated current	0,54 A
Aero Body		
M_b	Body mass	1,15 kg
D_t	Thrust distance	0,158 m
D_m	Distance center of gravity	0,0071 m
v_{pmax}	Voltage range	$\pm 24V$
Motor encoder		
	Nb. points per channel	512 points/tour
	Nb. two-way points in quadrature	2048 points/tour

The file **"InitCommande_Aero.m"** is used to initialise the platform parameters and can be completed for the synthesis of the control laws and the definition of the parameters necessary for the implementation of real-time control, such as the sampling periods T_{e1} and T_{e2}.

First laboratory session

2 Parameter identification for the motor drive

2.1 Transfer function of the motor drive

We will consider the relationship between the voltage input and the speed of the propeller as an output. This relationship is considered to be governed by a first order transfer function. The following form is proposed:

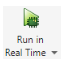
$$G(s) = \frac{\Omega_h(s)}{V_p(s)} = \frac{K_v}{1 + \tau p} \tag{2}$$

with K_v the gain of the motor and τ the time constant related to the mechanical load. With $x_1(t) = \omega_h(t)$ and $y(t) = \omega_h(t)$, the following state space model is obtained :

$$\begin{array}{rcl} \dot{x}_1(t) & = & -\frac{1}{\tau}x_1(t) + \frac{K_v}{\tau}v_p(t) \\ y(t) & = & x_1(t). \end{array} \tag{3}$$

For real-time identification of the system response, the Simulink file **"Identification_G.slx"** for recording the real-time response is provided. This real time control file allows to record the following quantities, in the form of column vectors:

- control signal voltage (**"cmd_vp"** variable);
- motor drive rotation speed in rad/s (**"omega"** variable);
- time vector in seconds (**"temps"** variable);

2.1.1. Record the response to a step of amplitude $0 - 6V$ peak to peak. To use the file **"Identification_G.xls"** you have to run the real time task with the button 

.

2.1.2. Based on these measurements, determine K_v and τ.

2.1.3. Validation of the identified parameters: compare the measured response to the ones obtained using the simulation of the identified model. For this comparison the file **"Validation_G.m"** is provided.

2.1.4. Validation of the model with another input: record the response to voltage input steps $(2,5V, 5V, 7,5V, 10V)$ by storing the velocity measurement ω. Using the file "Validation_G.m", compare the response obtained with the simulation based on the identified model. Is the assumption of linearity verified?

2.2 Motor speed control

We wish to control the angular speed of the motor to a set point ω_h^{ref}, in order to ensure the following specifications:

- steady state error zero,
- time of establishment in closed loop $t_e \approx 200$ms

- degree of pole damping $\xi = 0.7$.

From an experimental point of view, the implementation of the control law will be realized numerically by sampling the speed measurements $\omega_h(t)$ at the sampling period $T_{e1} = 2$ ms.

2.2.1 Synthesis of the motor speed control

Taking the state vector $x_a(t) = [\omega_h(t);\ z(t)]^\top$, with $z(t) = \int_0^t (\omega_h - \omega_h^{ref})dt$:

- establish a state representation of the motorization chain ;
- determine a feedback control law $u(t) = -K_a x_a(t)$ to control the speed $\omega_h(t)$ on the setpoint $\omega_h^{ref}(t)$ allowing closed lood dynamics given by the eigenvalues: $[-\xi\omega_0 - i\omega_0\sqrt{1-\xi^2}; -\xi\omega_0 + i\omega_0\sqrt{1-\xi^2}]$ avec $\omega_0 = 22$ et $\xi = 0.7$;
- using the Matlab features, and exploiting the continuous time models of the process and the controller, find the closed loop responses to a setpoint step of amplitude $\omega_h^{ref}(t) = 150$ rad/s. Plot the speed $\omega_h(t)$ the motor control voltage $v_p(t)$ and the reference signal $\omega_h^{ref}(t)$. Verify that the control signal respects the constraint $\pm 24V$. The imposed specification are verified ?
- comment on the behaviour obtained with respect to the eigenvalues of the closed loop.

2.3 Digital implementation of the speed controller - Experimental implementation

For the implementation in real time of the control law, the Simulink file **"Implantation_Cmd_MCC _Ka.slx"** allowing to realize the control in real time, see figure 5, and the file **"InitCommande_Aero.m"** allowing to initialize the parameters of the control are provided. This real time command file allows the following quantities to be stored in the form of column matrices:

- control voltage (variable **"cmd_vp"**);
- motor rotation speed in rad/s (variable **"omega"**);
- speed setpoint in rad/s (variable **"omega_ref"**);
- time vector in seconds (variable **"time"**);

2.3.1 Initialization of control parameters and experimentation

- Complete the file **"InitCommande_Aero.m"** with the parameters of the control law, $K_a = [K_\omega; K_z]$ and T_{e1}. Run the file to define these variables in Matlab ;
- Carry out the servo control for step setpoints with an amplitude of 150 rad/s ;
  Executar na planta
- Compare the performances obtained experimentally with those obtained in simulation. Gather the different results (simulation and experimental data) and conclude on the validity of the identified model and on the performances of the servo system. Análise dos dados
- Study the influence of the sampling period by carrying out an experiment with $T_{e1} = 50$ ms. Executar de novo na planta + Análise dos dados

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

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

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

$$A = \begin{bmatrix} -\frac{1}{\tau} & 0 \\ 1 & 0 \end{bmatrix}$$
$$B_1 = \begin{bmatrix} \frac{K_v}{\tau} \\ 0 \end{bmatrix}$$
$$B_2 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$
$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

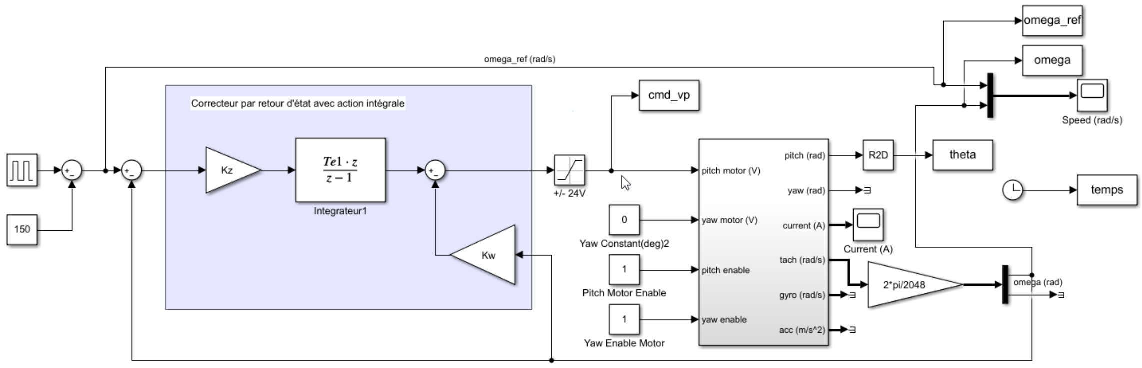


Figure 5: Simulink block used for the implementation.

3 Identification of tilt motion parameters

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

3.1.1. Following the functional description in Figure 3, write as a state representation the dynamic model of the system having ω_h as input signal and the angle θ as output variable. teoria

3.1.2. Study the equilibrium by explicitly writing the relationship between the constant input $\bar{v}_p$ and the output at the equilibrium $\bar{\theta}$. Use the intermediate quantities $\bar{\omega}_h$ and $F_0(\bar{\omega}_h)$. teoria

3.1.3. Show that by using the measurements of $\bar{\omega}_h$ and $\bar{\theta}$ we can obtain the force $F_0(\bar{\omega}_h)$ necessary to maintain an equilibrium position. Plot graphically $F_0(\bar{\omega}_h)$ by depicting the measurements related to the equilibrium points: teoria

$$\bar{\omega}_h \in \{-250; -200; -150; 150; 200; 250\}$$

planta

The provided Simulink file **"Identification_F.slx"** allows to record the real-time response. This file considers the previously calculated speed control.

3.1.4. Starting from the knowledge on $F_0(\bar{\omega}_h)$ and by approximating the relationship between the spped ω_h and the force F_0 by a static gain:

$$F_0(\omega_h) = K_t \omega_h,$$

Análise dos dados

find the numerical value for K_t. Compare the relationship $F_0(\bar{\omega}_h)$ measured with the resulting linear approximation on the entire range of speed variation.

3.2 Linearization of the dynamics around the position $\bar{\theta} = 0$

3.2.1. Linearize the dynamic model representing the relationship between the force F_0 and the tilt angle θ around the equilibrium position $\bar{\theta} = 0$. teoria

3.2.2. Show that in the free response ($F_0 = 0$) of the linear differential equation describing the evolution of the tilt angle in these conditions under the hypothesis that $\sin(\theta) \approx \theta$ leads to:

$$\ddot{\theta} + 2\zeta\omega_0\dot{\theta} + \omega_0^2\theta = 0 \qquad (4)$$

by specifying the relation between the parameters ζ and ω_0 and the physical parameters of the tilt motion.

By solving this equation, one can show that the evolution of θ exhibits a pseudo-periodic motion with $T = \frac{2\pi}{\omega_0\sqrt{1-\zeta^2}}$ and successive extrema such that $\ln\left(\frac{|\theta(0)|}{|\theta(\frac{kT}{2})|}\right) = k\eta$ with $\eta = \frac{\pi\zeta}{\sqrt{1-\zeta^2}}$ and k a positive integer. teoria

3.2.3. Record the free response (control voltage $v_p = 0$) of the platform from an initial condition close to $\theta_0 = 30°$. planta (coleta de dados)

3.2.4. From the pseudo-period of oscillation (T) and the damping of oscillations identify the values of J_p and D_p. Teoria após

3.2.5. Validation of the identified parameters J_p and D_p: write the state-space representation of the linear model with F_0 as input signal and θ as output and compare the free response in simulation with the recorded measurements. The Matlab function `initial` can be useful in this endeavour.[1] Analise de dados

3.3 Validation of the global model

The previous approach enabled us to identify the parameters K_v, τ, J_p, D_p, the function $F_0(\omega_h)$ in tabular form and the gain K_t corresponding to a linear approximation of $F_0(\omega_h)$. With these parameters and the speed feedback control system $v_p = K_a x(t)$, the aim in this section is to validate the system models. We will consider two models, the first, which we will call the non-linear model, takes into account the non-linearities of the system (equation (1) and the function $F_0(\omega_h)$ in tabular form) as well as the saturation in the signal v_p. The second model, called the linear or synthesis model, corresponds to the linear model around the equilibrium position $\bar{theta} = 0$.

3.3.1. Non-linear model
The file **"SimulateurNL_AERO_Cmd_Ka.slx"** is used to simulate the non-linear model obtained and the speed control of the propellers. To use this file, you need to fill in the identified values for K_v, τ, J_p, D_p, F_O_table and W_table. A double click in the block shown in figure 6 displays the parameter block where the identified values are filled in. The data F_O_table and W_table are vectors with the values F_0 and ω_h obtained in **3.1.3**.
Simulate a step response between a reference speed ω_h^{ref} and the angle θ of the model obtained and compare it with the experimental response measured with the platform. A velocity step of 150 rad/s will be used.

3.3.2. Linear model or synthesis model
Considering the linear model of **3.2.5**, and the gain K_t, establish a linear model between the input $\omega_h(t)$ and the output $\theta(t)$. Simulate in Matlab-Simulik the step response between a reference speed ω_h^{ref} and the angle θ of the model obtained. Compare with the response obtained with the non-linear model and the experimental response measured with the platform. A velocity step of 150 rad/s will be used for this. Conclude on the validity of the linear model.

[1] It should be noted that for a vector `temps` and a constant `a`, the command `index = temps>a; temps_tronc = temps(index);` generates a vector `temps_tronc` composed of the elements of `temps` having a value larger than `a`.

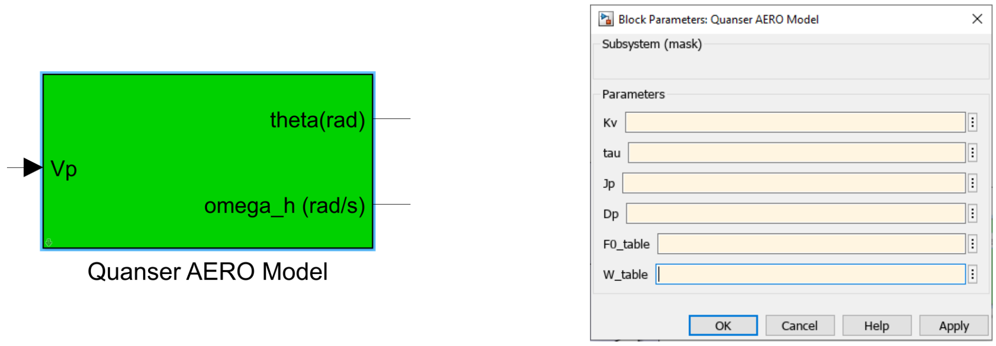


Figure 6: Non linear Simulator of Quanser AERO

To go further ... for those who wish to deepen the previous study in simulation and/or experimentally and to explore frequency-based control design approaches (c.f. Appendix of the Automation handout)

3.4 Speed control with a PI controller

In this part it is proposed to exploit the frequency-based approaches to establish a control law by considering the same specifications in order to maintain the same level of performance of the position servo. The synthesis of a continuous time corrector will be carried out by considering the looped structure characterized by the block diagram shown in figure 7. For the synthesis of the PI controller:

- Translate the constraints of the specifications into specifications on the frequency response in open loop;
- Determine a proportional-integral controller ensuring the phase margin and the cut-off pulsation necessary for the expressed specifications;
- Verify in simulation the performance of the controller;
- experimentally validate the results obtained: the control scheme "**Implantation_Cmd_MCC_PI.slx**" can be used by modifying, according to the desired test, the sampling period.

Design approaches for the synthesis of the PI controller are presented in appendix A of the handout "Control of dynamic systems".

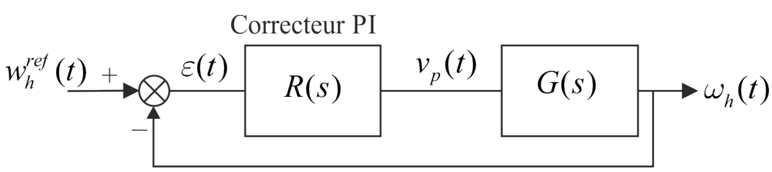


Figure 7: Speed control with a PI controller

4 Work to be done between the two laboratory sessions

This work must be done before the beginning of the second lab session.

Principle of tilt control for a helicopter

In order to control the tilt angle in an adequate manner, a cascade control structure is envisaged. It is composed of:

- a servo-control of the propeller rotation speed to a set value in order to avoid various disturbances (offset on the power amplifier, friction, gain differences observed for different amplitudes of the set point steps, ...),
- a strategy to elaborate the speed set point of the previous servo in order to obtain the desired behavior for the tilt of the helicopter: return to a set point position by limiting the oscillations.

By maintaining the speed loop, we now wish to elaborate the speed reference signal ω_h^{ref} to be imposed to the motor which will allow to bring the platform to a given tilt without inducing oscillations and with a response time of the order of $2s$ and a very low overshoot. The speed loop being fast compared to the dynamics required to control the tilt angle, we will approach this internal loop by a unitary gain. We consider the model of Figure 8.

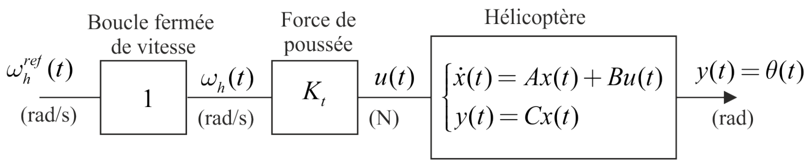


Figure 8: Modèle avec approximation de la boucle interne de vitesse

4.1 Specifications

We wish to control the angular speed of the motor to a setpoint ω_h^{ref}, respecting the following behavior, following a numerical realization of the control at a rate T_{e1} greater than or equal to 0.5 ms:

- zero steady-state error,
- overshoot inferior to 10%,
- rising time $200ms$.

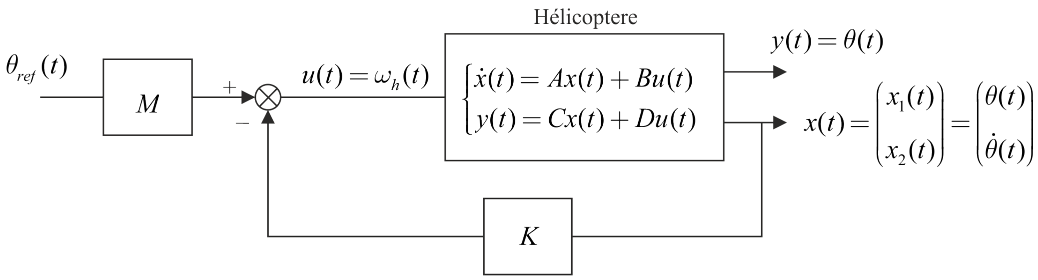


Figure 9: Closed loop with a state feedback controller

4.2 Tilt control: a state-space approach

In this approach, the control will be designed based on a state-space representation of the identified system. The closed loop with the state feedback controller is depicted in figure 9.

4.2.1. Starting from the identified linear model, infer the state representation of the model by considering as state vector $x(t) = (\theta(t),\ \dot{\theta}(t))^\top$.

4.2.2. Check the commandability condition in Matlab.

4.2.3. Control without integral action.
Using the LQ approach, determining state feedback control

$$u(t) = -K_{LQ}\, z(t) + M_{LQ}\, x_{ref}$$

to check the expected specifications.
Matlab functions will be used to synthesise the LQ controller. The command file **"InitCommande_Aero.m"** can be completed for the synthesis of the control law. The final validation in simulation could be done with the simulation model **"SimulateurNL_AERO_Cmd_LQ.slx"**.

Note : The synthesis of the LQ controller minimises the following criterion:

$$J = \int_0^\infty (z^\top Q z + u_c^\top R u_c) dt \tag{5}$$

where the weighting matrices Q and R need to be determined in order to obtain a control system that meets the specifications. In order to find the Q and R matrices, one possible approach is to normalise the state variables with respect to their maximum values and consider only the outputs to be controlled. Thus, $\tilde{x}_i = \frac{x_i}{x_{imax}}$ and with $\tilde{y} = C\tilde{x}$ and expressing the criterion with the normalised variables, we have :

$$J = \int_0^\infty (\tilde{y}^\top \tilde{Q}\tilde{y} + \tilde{u_c}^\top \tilde{R}\tilde{u_c}) dt = \int_0^\infty (\tilde{z}^\top C^\top \tilde{Q} C \tilde{z} + \tilde{u_c}^\top \tilde{R}\tilde{u_c}) dt \tag{6}$$

Unitary diagonal matrices for the $\tilde{Q}$ and $\tilde{R}$ usually provide a good starting point for iterative adjustment. For slower overall behaviour we increase $\tilde{R}$ and for faster overall behaviour we decrease $\tilde{R}$. If we want to make a particular state variable faster or slower, we respectively decrease or increase the corresponding $\tilde{Q}_i$ weighting. We then return to the criterion in its initial form (5) using matrices with normalisation terms.

$$\tilde{z}^\top C^\top \tilde{Q} C \tilde{z} = z^\top \underbrace{C^\top Q_n \tilde{Q} Q_n C}_{Q}\, \tilde{z}$$

$$\tilde{u}_c^\top \tilde{R} \tilde{u}_c = u_c^\top \underbrace{R_n \tilde{R} R_n}_{R}\, \tilde{u}_c$$

With:

$$Q_n = \begin{bmatrix} \frac{1}{y_{1\,\max}} & 0 & 0 \\ & \ddots & \vdots \\ 0 & & \vdots \\ 0 & \cdots & \frac{1}{y_{n\,\max}} \end{bmatrix}$$

$$R_n = \begin{bmatrix} \frac{1}{u_{1\,\max}} & 0 & 0 \\ & \ddots & \vdots \\ 0 & & \vdots \\ 0 & \cdots & \frac{1}{u_{m\,\max}} \end{bmatrix}$$

In the case of the Aero platform, an initial choice is to start with $\tilde{y}(t) = [\tilde{\theta}]$ with, $\tilde{Q} = [1]$ and $\tilde{R} = 1$. The following maximum values will be considered:

$$\theta_{max} = \pi/4 rad,$$
$$\omega_{hmax} = 500 rad/s.$$

4.2.4. Control with integral action.
In practice, the system includes resisting torques (dry friction in particular) which can be modeled as a global disturbance acting against the controlobjectives. Let the model be :

$$\dot{x}(t) = Ax(t) + Bu(t) + Bd(t)$$

where $d(t)$ models the perturbation. In order to get rid of the effect of the disturbance we complete the control model by introducing as a state variable $z_i(t) = \int_0^t (\theta - \theta_{ref})dt$;

- considering the augmented state vector $x_a = (x \;\; z_i)^\top = (\theta(t),\, \dot{\theta}(t),\, z_i(t))^\top$ write the augmented state representation;
- Using the LQ approach, determine the state feedback $u(t) = -K_{LQI}x_a(t)$ to verify the expected specifications. In a similar way to the control without integral action, the Matlab functions will be used to synthesise the LQ controller. The command file **"InitCommande_Aero.m"** can be completed to synthesise the control law. The final simulation validation can be carried out using the simulation model **"SimulateurNL_AERO_Cmd _LQI.slx"**. For the choice of weights, consider $\tilde{y}(t) = [\tilde{\theta} \quad z_i]^\top$ and $z_{imax} = 0.15$.

2nd laboratory session

The aim of this session is to experimentally implement the control laws that were the subject of the preparatory work. This work must have been done.

5 Numerical implementation of the LQ control

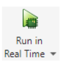
For the implementation in real time of the law of control by state feedback the Simulink file **"Implantation_Cmd_LQ.slx"** allowing to realize the control in real time and the file **"InitCommande_Aero.m"** allowing to initialize the parameters of the control are provided. The real-time control environment makes it possible to store in the form of column matrices the following quantities:

- control voltage (variable **"cmd_vp"**);
- tilt position in degrees (variable **"theta"**); tilt speed in degrees per second (variable **"theta_p"**);
- tilt motor angular speed reference (variable **"omega_ref"**);
- tilt angular position setpoint in degrees (variable **"theta_ref"**);
- time vector in seconds (variable **"time"**);

From an experimental point of view, the implementation of the control law will be realized numerically by sampling the position measurements $\theta(t)$ at the sampling period $T_{e2} = 20$ ms.

5.1 Control without integral action

Complete the file **"InitCommande_Aero.m"** [2] with the parameters of the corrector, K and M (which must be called `K_lq`, `M_lq` in MATLAB) Run the file to define these variables in Matlab.

Launch the file **"Implantation_comm_LQ.xls"** allowing to implement the control in real time. To use the file you have to run the real time task with the button 

.

Display the signals **"theta"**, **"theta_ref"** and **"omega_ref"** for a reference position given by a step with amplitude $\theta_{ref} = 10°$.

Compare the results obtained with the result obtained in simulation.

[2]This file includes the internal speed loop calculated previously. The integral action is set up with an "anti wind-up" action to take into account the saturation of the command signal at ± 24 Volts.

5.2 Command with integral action

- Consider the file **"Implantation_Cmd_LQ.slx"** and add an integral action, so as to implement the LQ command with integral action determined in the preparatory work.
- Display the signals **"theta"**, **"theta_ref"** and **"omega_ref"** for a reference position in step of amplitude $\theta_{ref} = 10°$.
- Compare the performances with the result obtained in simulation and with the behavior obtained with the LQ command without integral action.

6 Control by state feedback and observer

We want to study here the possibility of performing the control without the derivative of the angular position by completing the state feedback with an observer, according to the diagram of figure 10, for the control without integral action, or figure 11 for the control with an integral action

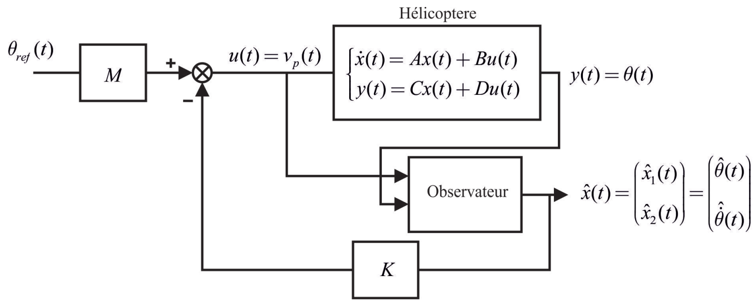


Figure 10: Closed loop with state feedback corrector and observer

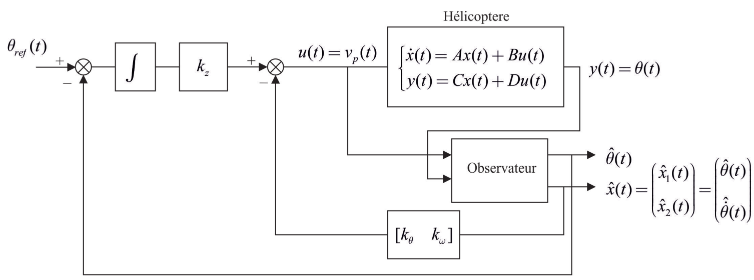


Figure 11: Closed loop with state feedback corrector with integral action and observer

6.1 Observer synthesis

Construct an observer. To do this:

6.1.1. Write the literal equations of the observer allowing to reconstruct the state variables $\theta(t)$ and $\dot{\theta}(t)$ from the control signal $\omega_h(t)$ and the measurement of $\theta(t)$.

6.1.2. Considering the desired time-performance and anticipating the fact that the control law will then be implemented numerically at the rate T_{e2}, justify the choice of the following eigenvalues for the observer:

$$\lambda_1, \lambda_2 = -\omega_0\xi_0 \pm j\,\omega_0\sqrt{1-\xi_0^2},$$

with $\omega_0 = 15$ rad/s and $\xi_0 = 0,9$.

6.1.3. Compute the observer's gain matrix L using the appropriate MATLAB function.

6.1.4. Plot with MATLAB the responses of the output and the control to a setpoint step.

6.1.5. Final simulation validation can be carried out using the simulation model **"SimulateurNL_AERO_Cmd_LQG.slx"**

6.2 Experimental implementation

The observer is realized in discrete form by performing the following approximation (which corresponds to the Euler transform):

$$\frac{\mathrm{d}\hat{X}(t)}{\mathrm{d}t} \approx \frac{\hat{X}\left((k+1)T_e\right) - \hat{X}\left(k\,T_e\right)}{T_e}$$

from which we deduce the discrete equation of state:

$$\hat{X}\left((k+1)T_e\right) = \left[I + (A - LC)T_e\right]\,\hat{X}\left(k\,T_e\right) + B\,T_e\,u\left(k\,T_e\right) + L\,T_e y\left(k\,T_e\right)$$

and the control form:

$$u\left(k\,T_e\right) = -K\,\hat{X}(k\,T_e) + M\,\theta_{ref}(k\,T_e)$$

For the real-time implementation of the control law by state feedback and observer, the Simulink file **"Implantation_Cmd_LQG.slxs"** allowing to implement the controller, see figure 12, and the file **"InitCommande_Aero.m"** allowing to initialize the parameters of the control are provided. This real-time control file allows the following quantities to be stored in the form of column matrices:

- control voltage (variable **"cmd_vp"**);
- tilt position in degrees (variable **"theta"**);
- estimated tilt position in rad (variable **"theta_obs"**);
- estimated tilt speed in rad per second (variable **"theta_p_obs"**);
- tilt motor angular velocity reference (variable **"omega_ref"**);
- tilt angular position setpoint in degrees (variable **"theta_ref"**);
- time vector in seconds (variable **"temps"**);

6.2.1. Keep the same sampling period T_{e2} as in the case of the state feedback control law.

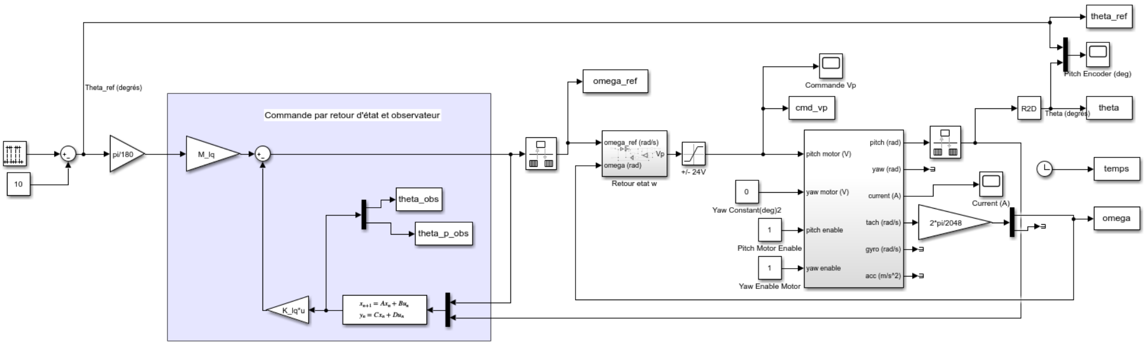
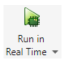


Figure 12: Simulink block used for simulation.

Complete the file **"InitCommande_Aero.m"** with the values of the sampling period T_{e2} and the parameters of the corrector and the observer, namely A, B, C, K_lq, L and M_lq. Run the file to define these variables in Matlab.

Launch the file **"Implantation_Cmd_LQG.slx"** allowing to implement the control in real time. To use the file you have to run the real time task with the button 

.

Plot the signals **"theta"**, **"theta_ref"** and **"omega_ref"**.

Verify the performance obtained with the LQ control without integral action and the observer and compare this result with the one obtained with the state feedback control without observer.

6.3 Command with integral action and observer

- Consider again the file **"Implantation_Cmd_LQG.slx"** by adding an integral action, such that the LQ control benefit from both the integrator and the observer features.
- Plot the signals **"theta"**, **"theta_ref"** and **"omega_ref"** for a step reference position with amplitude $\theta_{ref} = 10°$.
- Compare the performance with the result obtained in simulation and with the behavior obtained with the LQ control without integral action.

6.4 Analysis of stability margins

In order to establish a comparative study, as complete as possible, this part consists of analyzing the stability margins of the control laws without and with observer. Please refer to Sections 5.6 and 5.7 of the "Commande des systèmes dynamiques" handout.

6.4.1 Margins with state feedback

By exploiting the state representation in open loop

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ r(t) = Kx(t) \end{cases}$$

where K is the state feedback matrix. Find the gain and phase stability margins.

6.4.2 Margins with the state feedback and the observer

In this part, the effect of the observer on the stability margins is taken into account. For this analysis it will first be necessary to establish the state representation of the system augmented by the observer. We note for that $x_a(t) = [x^\top; \hat{x}^\top]^\top$ the vector of augmented state composed of the state variables of the process and those of the observer.

- Establish the open loop state representation of the system with observer

$$\begin{cases} \dot{x}_a(t) = A_a x_a(t) + B_a u(t) \\ r(t) = K_a x_a(t) \end{cases}$$

  The matrices A_a and B_a are obtained according to the model of the process and the representation of the observer, i.e. according to the matrices A, B, A_{obs} et B_{obs}.

- Find the stability margins using this model.

7 Conclusions

Write a summary of the work done, resuming the approach taken for each control design, comparing the two types of feedback and comparing the experimental results with the results obtained in simulation.

To go further ... it is useful as personal work for those who wish to deepen the previous study in simulation and/or experimentally and to explore the frequency approaches (c.f. Appendix of the Automation handout)

8 Tilt control: frequency approach

In this part, it is proposed to exploit the frequency approaches to establish a control law while maintaining the same level of performance of the tilt servo. The voltage control law of the tilt motor will be realized from a PID corrector. The closed loop with the PID controller is shown in the figure 13.

8.1 Design of a PID controller

- Translate the constraints of the specifications into specifications on the frequency response in open loop.
- Find a PID controller allowing to obtain a response to a step with less than 10% of overshoot and a time of the first maximum $t_m = 1s$. The PID controller considered corresponds to the following transfer function:

$$R(s) = K_p \left(1 + \frac{1}{T_i\ s} + \frac{T_d\ s}{1 + \tau_d\ s}\right)$$

- Validate the synthesis by plotting, using a Matlab script, the frequency response in open loop and the index response in closed loop.
- Experimentally validate the results obtained: the Simulink control scheme **"Implantation_PID.slx"** can be used by modifying, according to the desired test, the sampling period.

8.2 Analysis of stability margins

- Compare the stability margins obtained with the LQ control with integral action and observer and the PID control.

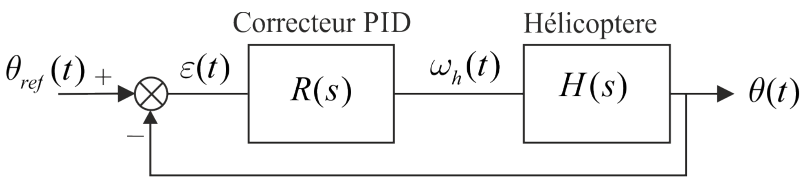


Figure 13: Closed loop with PID controller