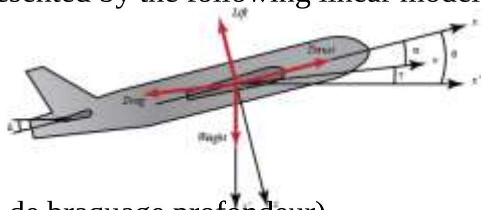
Au 412 : Applied control system

Exercice Session – Aircraft Pitch : Control design

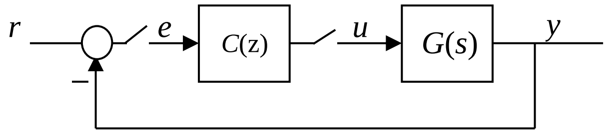
Consider pitch angle dynamics of an airplane approximately represented by the following linear model [1]:

$$TF(s) = G(s) = \frac{1.151\,s + 0.1774}{s^3 + 0.739\,s^2 + 0.921\,s}$$



where the input $\delta(t)$ or $\Delta(s)$ is the elevator deflection angle (angle de braquage profondeur)
and the output $\theta(t)$ or $\Theta(s)$ is the aircraft pitch angle theta (angle de tangage de l'avion).

We will use the following control scheme:



0) Create a Matlab script file: **`AU412_TD.m`**
   Give the continuous transfer function $G(s)$ of the linear plant.

1) **Root Locus. Example of designing a simple controller**
Consider a proportional controller: $C(s) = K$.

- Compute the closed loop transfer function.
- Sketch the root locus corresponding to the controlled system.
- Analyze the stability with respect to K
- Give the values of K at which the system is oscillating.
- Simulate the system with Simulink and try to confirm your analysis study.

Answer the same questions when taking an integral action: $C(s) = K/s$.

2) **Control design. Method 1: Design a <u>continuous</u> controller C(s) then <u>convert</u> C(s)→C(z)**

a) Consider a step input of 0.2 radians, design a PID or PID filtered continuous controller with **`sisotool`** (find the best parameters of the controllers):

$$C(s) = K_p + \frac{K_p}{s} + K_p s \quad \text{or} \quad C(s) = K_p + \frac{K_p}{s} + K_p \frac{s}{s+\alpha}$$

so that the step response fulfils the following conditions:

- Overshoot (dépassement maximal) is less than 10%
- Rise time (temps de montée) is less than à 2 seconds
- Settle time (temps d'établissement) of 5% is less than 10 seconds
- Steady state error (erreur en régime permanent) is less than 2%

<u>Procedure:</u>

- Tab *Compensator Editor* →Compensator → $F = 0.2$ (the input r is multiplied by 0.2)

---

[1] Ref : http://control.me.cmu.edu/CTMS/Content/Aircraft_Pitch/System/Modeling/html/Aircraft_Pitch_Modeling.html
http://control.me.cmu.edu/CTMS/Content/Aircraft_Pitch/Control/PID/html/Aircraft_Pitch_PID.html

- Tab *Analysis Plots* → Plot 1: Step → Contents of plots : Closed loop *r* to *y*.
- Tab *Automated Tuning*:

  Design Method: PID Tuning
  Controller Type: PID or PID filtred
  Tuning algorithm: Parameter search (to find adequate values of the PID parameters)
  Performance metric: Integral Square Error, or other (the algorithm use this criterion in finding the best parameters)
  Update Compensator
- Check if your requirements are satisfied. If not, do manual adjustments by using for example: Tab *Graphical Tuning* → Plot 1: Open Loop: Rlocus → Show Design Plot (At the same time, open the Step response window) → adjust the zeros/… until you obtain a satisfactory response.
  Note: You can place poles/zeros ×, **O (in red)** of C(s), but not those **(in blue)** of the plant of G(s).

b) **Simulation with Simulink.** Save the obtained PID:
  - File → Export → C → ToWorkspce

Simulate the controlled system (plant + continuous controller) in Simulink. Use one of the blocs *Zero-Pole*, *Transfer Fcn*, *State-Space* from *Simulink/Continuous* group, or *LTI System* from *Control System Toolbox*.

c) **Discretization C(s)→C(z).** Give the discrete version of your PID controller (**`c2d`**), after choosing a suitable sample time (use **`bode`**). To do so, plot the Bode diagram of $G(s)$, and decide on which sample frequency is suitable that fulfil Shannon criterion. For example, choose a sample frequency after which the gain of $G(s)$ is less than -30dB.

Simulate the system (plant + discrete PID) in Simulink.
Compare these two controllers with different sample rates.

3) **Method 2. Design the discrete controller C(z) with sisotool.** It is also possible to shorten the procedure and design the discrete controller $C(z)$ with *sisotool*. To do this, give a discrete approximation of the plant, $G(s)$ → $G(z)$, with **`c2d`** (method='zoh'). Use *sisotool* and consider the discrete system $G(z)$ instead of the continuous system in designing the controller, do the necessary adjustments to fulfil the requirements. Save the obtained controller to Workspace. Test the new controller in Simulink (plant + discrete controller).

4) **Implementing a discrete controller**
Write the equations (recurrence equations) that allow you to implement the obtained controller.
In Simulink, replace the controller bloc with delay blocs + an Embedded Matlab function.

5) **Design in state space**
Write the discrete system G(s) in state space representation (use **`ss`**) and give a discrete approximation.
In Matlab: **`sysc=ss(Gs); sysd=c2d(sysc,T,'zoh')`**
or: **`Gz=c2d(Gs,T,'zoh'); sysd = ss(Gz)`**

Compute the gain K that place the poles of the controlled system (closed loop system) to $\{1\ ;\ 0.5 \pm 0.5i\}$. Une the command **`place`**.

Simulate, in Simulink, the behaviour of the controlled system (plant + controller). Add a zero order hold 'zoh' bloc to simulate DAC.
Give a temporal analysis of the controlled system. Plot the error between the desired input (consigne) and the output. Comments.