Control of Aircraft

AU511

The goal of this project is to study the longitudinal stability of a Mirage III around an equilibirum point and to synthesize controllers.

Calka Magdalena
Pichon Corentin

IPSA, Institut Polytechnique des Sciences Avancées
2020

Contents

1 Introduction

In this project we will study the Mirage III aircraft around operating point 74, with a Mach number of 1.4 and an altitude of 24 000 feet. And then synthesize controllers for the longitudinal modes of this aircraft. For this project we first considered the following hypothesis :

- Symmetrical flight in the vertical plane (null sideslip and roll)
- Thrust axis merged with aircraft longitudinal axis
- Inertia principal axis is aircraft transverse axis (diagonal inertia matrix)
- Fin control loop: it's dynamics will be neglected for the controller synthesis
- The altitude sensor is modeled by a 1^{st} order transfer function with a time constant $\tau = 0.71\ s$

The first step of this project was to enter the different parameter needed in python. Some of the aircraft characteristics are general and were already given :

- Total length: $l_t = \frac{3}{2} \cdot l_{ref}$
- Reference length: $l_{ref} = 5.24\ m$
- Position of CoG: $c = 52\ \%$
- Mass: $m = 8400\ kg$
- Reference surface (wings): $S = 34\ m^2$
- Radius of gyration: $r_g = 2.65\ m$

Some other aerodynamic are dependant of the speed, Mach number, and we did an numerical interpolation on the given curves to find the right values for the chosen operating point. Giving use the following results :

- Drag coefficient for null incidence: $Cx_0 = 0.033$
- Lift gradient coefficient wrt α: $Cz_\alpha = 2.58\ rad^{-1}$
- Lift gradient coefficient wrt δ_m: $Cz_{\delta m} = 0.59\ rad^{-1}$
- Equilibrium fin deflection for null lift: $\delta m_0 = 0\ rad$
- Incidence for null lift and null fin deflection: $\alpha_0 = 0.008\ rad$
- Aerodynamic center of body and wings: $f = 0.608$
- Aerodynamic center of fins (pitch axis): $f_\delta = 0.9$
- Polar coefficient: $k = 0.4$
- Damping coefficient: $Cm_q = -0.36\ s/rad$

2 Study of the uncontrolled aircraft

2.1 Equilibrium conditions

Equation used to ensure control of the aircraft have been linearized to facilitate data processing and numerical resolution. To warrant the stability of the aircraft the linearization have to be done at an equilibrium point.

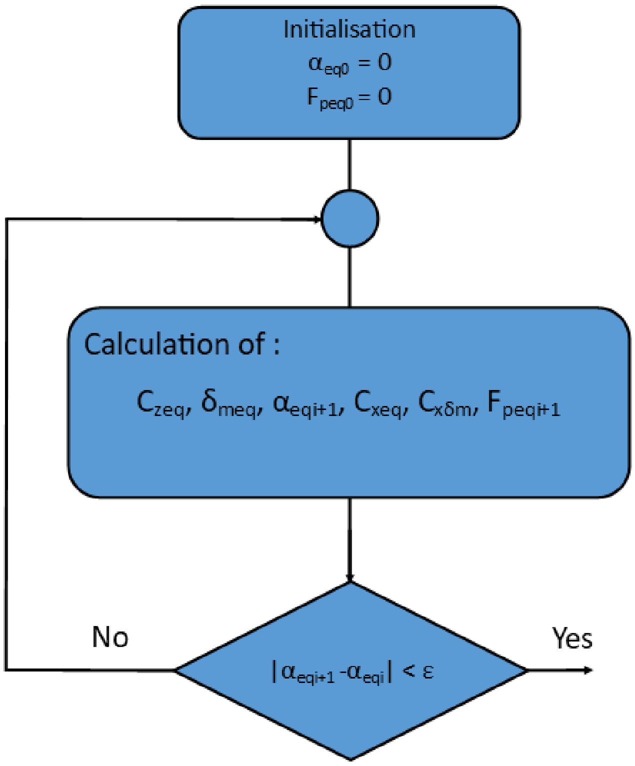


Figure 1: Algorithm to find equilibrium point

So, we used algorithm present above to calculate this point. Needed information about this point for the implementation of the step response of the system are:

- Lift coefficient at the equilibrium point C_{zeq}
- Angle of attack at the equilibrium point α
- Norm of the thrust vector F_{px}

These values were obtain though the following equations and data:

$$
\begin{aligned}
C_{zeq} &= \frac{1}{QS}(mg_0 - F_{pxeqi}\sin\alpha_{eqi}) \\
\delta_{meq} = \delta_{m0} &= -\frac{C_{xeq}\sin\alpha_{eqi} + C_{zeq}\cos\alpha_{eqi}}{C_{x\delta m}\sin\alpha_{eqi} + C_{z\delta m}\cos\alpha_{eqi}}\,\frac{X}{Y-X} \\
\alpha_{aqi+1} &= \alpha_0 + \frac{C_{Zeq}}{C_{z\alpha}} - \frac{C_{Z\delta m}}{C_{Z\alpha}}\delta_{meq} \\
C_{Xeq} &= C_{X0} + kC^2 Zeq \\
C_{x\delta m} &= 2kC_{zeq}C_{Z\delta m} \\
F_{pxeqi+1} &= \frac{QSC_{Xeq}}{\cos\alpha_{aqi+1}}
\end{aligned}
$$

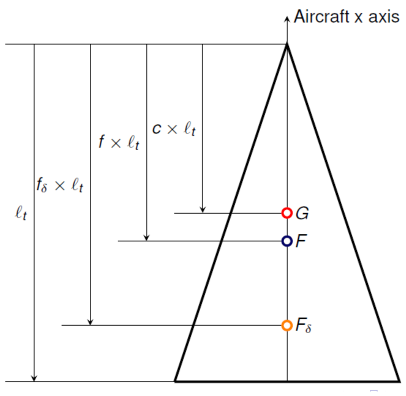


Figure 2: Geometry, Source: "tp_pilotage_auto2020.pdf"

With:

$$X = -l_t(f - C)$$

$$Y = -l_t(f_\delta - C)$$

Others variables where find with interpolation, the curves present in "tp_pilotage_auto2020.pdf" were processed to recover points, then these points were interpolated which allowed us to find results devoid of errors due to reading the graphs. The following python algorithm was used for this purpose:

```python
import pandas as pa
import scipy.interpolate as sip

L = ["a0","Cx0","Cza","f"]

for i in L:
    df = pa.read_csv(i+".csv",names=["x","y"],delimiter=";")
    x, y = [float(j) for j in df.x.to_list()], [float(h) for h in df.y.to_list()]
    f = sip.interp1d(x, y)
    print(i," : ",f(1.4))
```

So with the use of those variables and the algorithm presented in the Figure 1, we have our python code to find equilibrium point. It compute alpha until it's variations tends to zero:

```python
alpha, Fpx = 0, 0
alphaList = []
alphaList.append(alpha)

alpha = alpha_0 + Cz_eq/C_zalpha - Cz_deltam/C_zalpha*dela_meq
alphaList.append(alpha)
Fpx = Q*S*Cx_eq/(np.cos(alpha))
i = 1
epsilon = 0.00000000000000000000001
while abs(alphaList[i] - alphaList[i-1])>epsilon:
    Cz_eq = 1/(Q*S)*(m*g0-Fpx*np.sin(alpha))
    Cx_eq = C_x0 + k*pow(Cz_eq, 2)
    Cx_deltam = 2*k*Cz_eq*Cz_deltam
    delta_meq = delta_m0 -(Cx_eq*np.sin(alpha) +
    Cz_eq*np.cos(alpha))/(Cx_deltam*np.sin(alpha)+Cz_deltam*np.cos(alpha))
    dela_meq = delta_meq*(X/(Y-X))

    alpha = alpha_0 + Cz_eq/C_zalpha - Cz_deltam/C_zalpha*dela_meq
    alphaList.append(alpha)
    Fpx = Q*S*Cx_eq/(np.cos(alpha))
    i +=1
```

2.2 First model

We will in the first time build a full model of the longitudinal aircraft stability with the 6 variables. This model will be built considering small variation about the equilibrium point previously given. Using this simplified longitudinal model we obtain the following equations :

$$X_V = \frac{2QSC_{xeq}}{mV_{eq}}$$

$$X_\alpha = \frac{F_{eq}}{mV_{eq}} sin(\alpha_{eq}) + \frac{QSC_{X\alpha}}{mV_{eq}}$$

$$X_\gamma = \frac{g_0 cos(\gamma_{eq})}{V_{eq}}$$

$$X_{\delta m} = \frac{QSC_{X\delta m}}{mV_{eq}}$$

$$X_\tau = -\frac{F_\tau cos(\alpha_{eq})}{mV_{eq}}$$

$$m_V = 0$$

$$m_\alpha = \frac{QSl_{ref}C_{m\alpha}}{I_{YY}}$$

$$m_q = \frac{QSl_{ref}^2 C_{mq}}{V_{eq} I_{YY}}$$

$$m_{\delta m} = \frac{QSl_{ref}C_{m\delta m}}{I_{YY}}$$

$$Z_V = \frac{2QSC_{zeq}}{mV_{eq}} \approx \frac{2g_0}{V_{eq}}$$

$$Z_\alpha = \frac{F_{eq}}{mV_{eq}} \cos(\alpha_{eq}) + \frac{QSC_{Z\alpha}}{mV_{eq}}$$

$$Z_\gamma = \frac{g_0 \sin(\gamma_{eq})}{V_{eq}}$$

$$Z_{\delta m} = \frac{QSC_{Z\delta m}}{mV_{eq}}$$

$$Z_\tau = \frac{F_\tau \sin(\alpha_{eq})}{mV_{eq}}$$

Then we find the following state space model :

$$\begin{pmatrix} \dot{V} \\ \dot{\gamma} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} -X_V & -X_\gamma & -X_\alpha & 0 & 0 & 0 \\ Z_V & 0 & Z_\alpha & 0 & 0 & 0 \\ -Z_V & 0 & -Z_\alpha & 1 & 0 & 0 \\ 0 & 0 & m_\alpha & m_q & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & V_{eq} & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} V \\ \gamma \\ \alpha \\ q \\ \theta \\ z \end{pmatrix} + \begin{pmatrix} 0 \\ Z_{\delta m} \\ -Z_{\delta m} \\ m_{\delta m} \\ 0 \\ 0 \end{pmatrix} (\delta m)$$

Replacing with the value that we discussed in the introduction we have the following matrix :

$$A = \begin{pmatrix} -0.0228 & -0.0225 & -0.0459 & 0 & 0 & 0 \\ 0.044 & 0 & 1.3097 & 0 & 0 & 0 \\ -0.044 & 0 & -1.3097 & 1 & 0 & 0 \\ 0 & 0 & -55.4522 & -0.7051 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 435.3323 & 0 & 0 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 \\ 0.2956 \\ -0.2956 \\ -54.7586 \\ 0 \\ 0 \end{pmatrix}$$

The C and D matrix will then depend and which variables we want to get as an output.

2.3 Open loop modes studies

After building the model we can use the damp function of control Matlab python library. We found the following results.
For Phugoid :

- λ = -1.007+7.44j ξ = 0.1342 f = 7.508 Hz
- λ = -1.007-7.44j ξ = 0.1342 f = 7.508 Hz

For Short period :

- λ = -0.01689+0.02626j ξ = 0.5411 f = 0.03122 Hz
- λ = -0.01689-0.02626j ξ = 0.5411 f = 0.03122 Hz

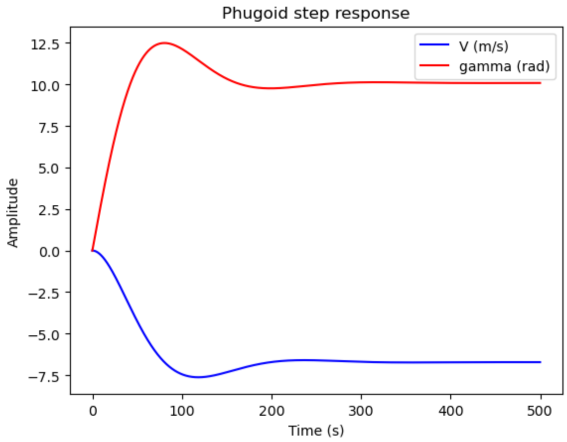
2.4 Uncontrolled aircraft transient phase studies

When we take the eigenvalues of the matrix we clearly see that their is one pole per variable. But we have two pair of conjugated poles. Those are the short period and phugoid modes. The short period is the one with the bigger pulsation and the phugoid is the other. We can then take as an hypothesis that we neglect the coupling between the two pair of variables and the other variables too. Allowing to make a simplified model for each mode.

2.4.1 Phugoid

$$\begin{pmatrix} \dot{V} \\ \dot{\gamma} \end{pmatrix} = \begin{pmatrix} -X_V & -X_\gamma \\ Z_V & 0 \end{pmatrix} \begin{pmatrix} V \\ \gamma \end{pmatrix} + \begin{pmatrix} 0 \\ Z_{\delta m} \end{pmatrix} \delta_m$$

$$\begin{pmatrix} \dot{V} \\ \dot{\gamma} \end{pmatrix} = \begin{pmatrix} -0.0338 & -0.0225 \\ 0.044 & 0 \end{pmatrix} \begin{pmatrix} V \\ \gamma \end{pmatrix} + \begin{pmatrix} 0 \\ 0.2956 \end{pmatrix} \delta_m$$



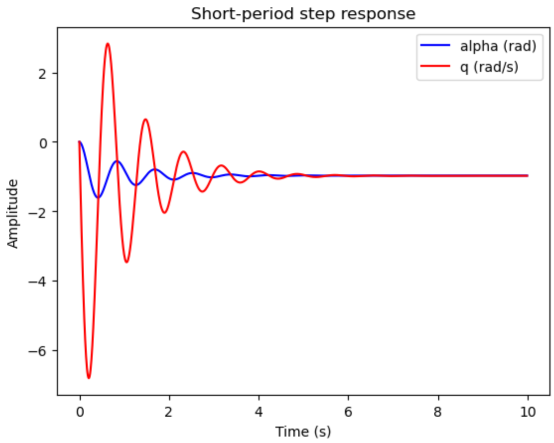
$$\frac{V}{\delta_m} = \frac{-0.00666}{s^2 + 0.03384s + 0.0009912}$$
$$\frac{\gamma}{\delta_m} = \frac{0.2956s + 0.01}{s^2 + 0.03384s + 0.0009912}$$

Figure 3: Phugoid step response

2.4.2 Short-period

$$\begin{pmatrix} \dot{\alpha} \\ \dot{q} \end{pmatrix} = \begin{pmatrix} -Z_\alpha & 1 \\ m_\alpha & m_q \end{pmatrix} \begin{pmatrix} \alpha \\ q \end{pmatrix} + \begin{pmatrix} -Z_{\delta m} \\ m_{\delta m} \end{pmatrix} \delta_m$$

$$\begin{pmatrix} \dot{\alpha} \\ \dot{q} \end{pmatrix} = \begin{pmatrix} -1.3097 & 1 \\ -55.4522 & -0.7051 \end{pmatrix} \begin{pmatrix} \alpha \\ q \end{pmatrix} + \begin{pmatrix} -0.2956 \\ -54.7586 \end{pmatrix} \delta_m$$



$$\frac{\alpha}{\delta_m} = \frac{-0.2956s - 54.97}{s^2 + 2.015s + 56.38}$$
$$\frac{q}{\delta_m} = \frac{-54.76s - 55.33}{s^2 + 2.015s + 56.38}$$

Figure 4: Short period step response

2.5 Reduced model

We now consider that we have a perfect auto-throttle so the speed is control and we can get it out of the model. And we have the following :

$$\begin{pmatrix} \dot{\gamma} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} 0 & Z_\alpha & 0 & 0 & 0 \\ 0 & -Z_\alpha & 1 & 0 & 0 \\ 0 & m_\alpha & m_q & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ V_{eq} & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \gamma \\ \alpha \\ q \\ \theta \\ z \end{pmatrix} + \begin{pmatrix} Z_{\delta m} \\ -Z_{\delta m} \\ m_{\delta m} \\ 0 \\ 0 \end{pmatrix} (\delta m)$$

The matrix are as following :

$$A = \begin{pmatrix} 0 & 1.3097 & 0 & 0 & 0 \\ 0 & -1.3097 & 1 & 0 & 0 \\ 0 & -55.4522 & -0.7051 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 435.3323 & 0 & 0 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0.2956 \\ -0.2956 \\ -54.7586 \\ 0 \\ 0 \end{pmatrix}$$

3 Controllers synthesis

3.1 q feedback loop

To start building the autopilot for this aircraft we will create a first feedback loop on the pitch rotation speed. We will need to find a gain K_r for this loop. To do so we take the open loop system that can be simplified as the transfer function of q over δ_m. We use the sisotool implementation with this system to get the root locus and all the useful information.

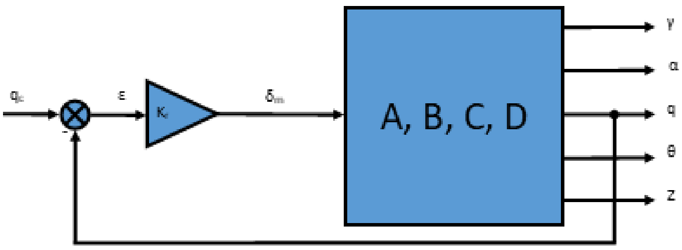


Figure 5: q feedback loop

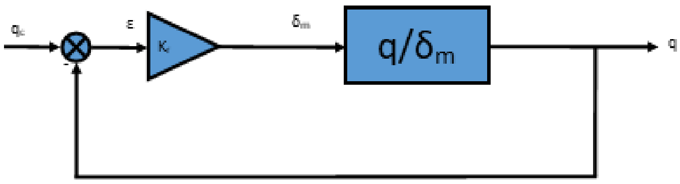


Figure 6: Simplified q feedback loop

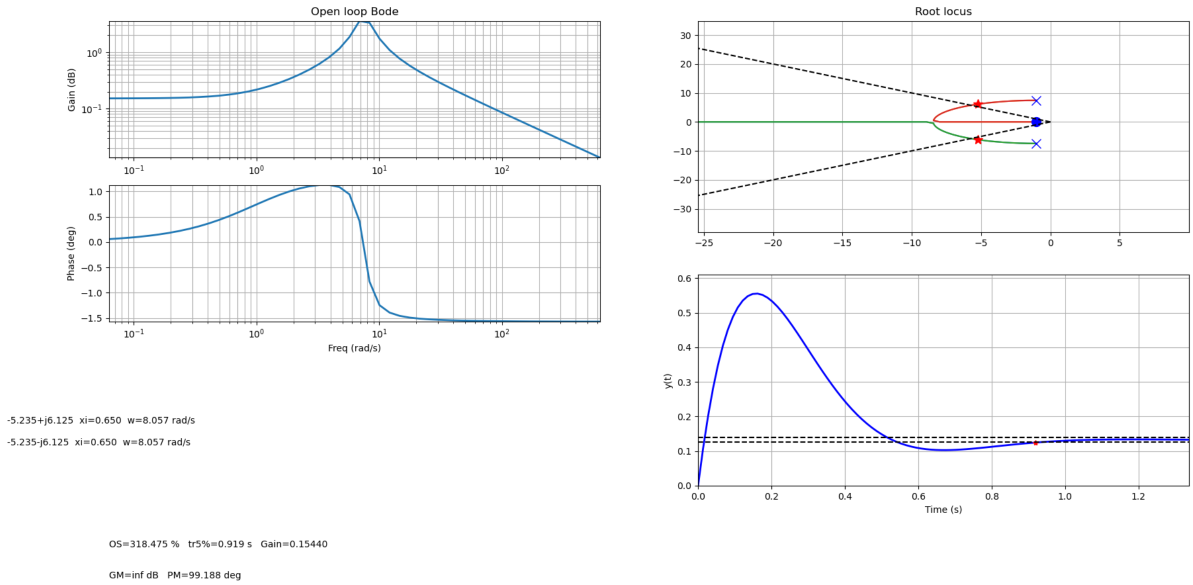


Figure 7: Sisotool of q feedback loop

We actually used sisotool with a minus in front of the transfer function because the result first use of sisotool showed unstable poles with any gain. Thus, we will have to put a minus in front of the gain when we will get it.

We want to get a damping ratio ξ of 0.65 using this feedback loop. To do so with sisotool we can change the gain applied with the root locus graph by moving from zeros to poles. We see that a K_r

of 0.1544 gives us two conjugated poles with the damping ratio required. Then we just need to build the feedback loop around the system with a gain K_r of -0.1544.

As we want to avoid any problem in the feedback with the control Matlab library we will use the direct calculation method to build the new state space with feedback loop. We use the following equations :

$$A_q = A - K_r B C_q$$
$$B_q = K_r B$$
$$C_q = C_{out}$$
$$D_q = 0$$

We can change the C matrix depending on which variable we want to see as an output.

So, the state space for the q feedback loop is the following :

$$A = \begin{pmatrix} 0 & 1.3097 & 0.0459 & 0 & 0 \\ 0 & -1.3097 & 0.9544 & 0 & 0 \\ 0 & -55.4522 & -9.1599 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 435.3323 & 0 & 0 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} -0.0456 \\ 0.0456 \\ 8.4547 \\ 0 \\ 0 \end{pmatrix} \quad C = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

And the transfer function is :

$$\frac{q}{q_c} = \frac{8.455s + 8.542}{s^2 + 10.47s + 64.92}$$

The poles and their characteristics are :

- λ : -5.235+6.125j ξ : 0.6497   f : 8.057 Hz
- λ : -5.235-6.125j ξ : 0.6497   f : 8.057 Hz

The step response obtained is :

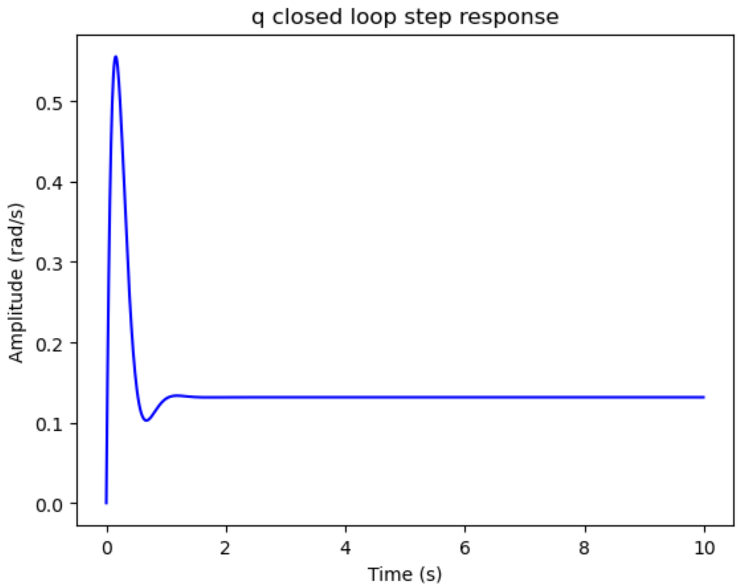


Figure 8: q feedback loop step response

We want to get the same steady state gain for α with or without the q feedback loop previously built to do so we will introduce a washout filter in the loop.

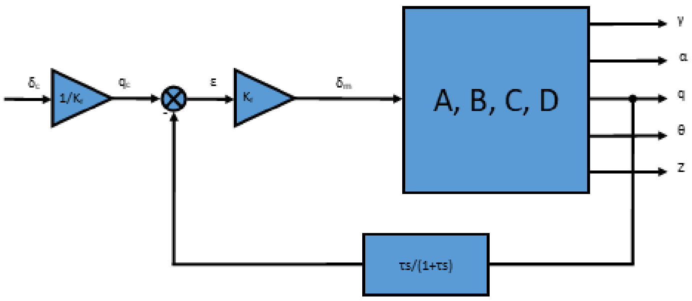


Figure 9: q feedback loop with washout

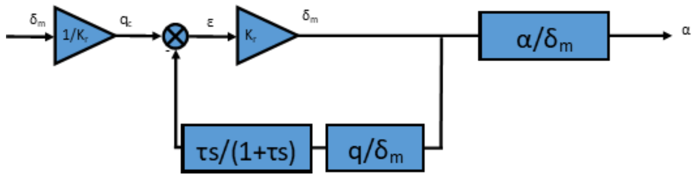


Figure 10: Simplified q feedback loop with washout

To choose the τ we first use the following equation : $\frac{1}{\tau} < \frac{\omega}{2}$. So taking the pulsation of the short period mode we find that $\tau > 0.266$. After trying a few different value we have found that with a τ of 0.5 we have a good response of the system. Giving use the following step responses.

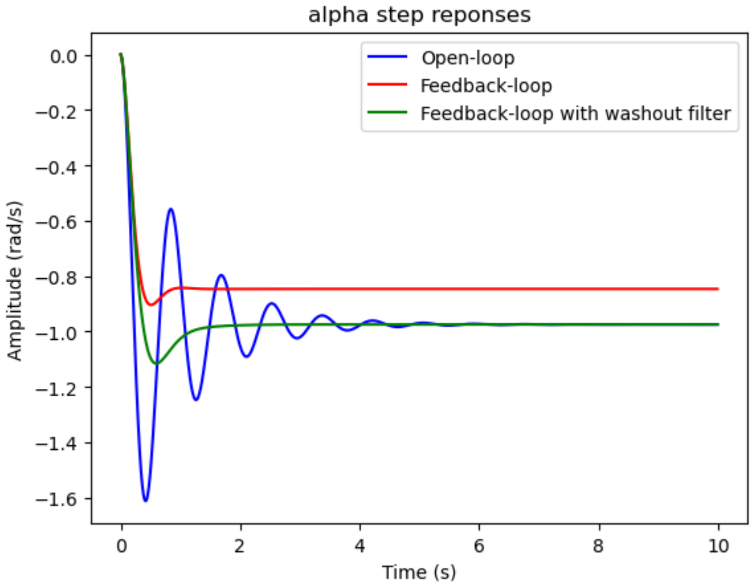


Figure 11: Alpha step responses

3.2 γ feedback loop

In this second part we will build a second feedback loop on the flight path angle γ. To do so we will use sisotool on the previous system to find the K_γ of the feedback.

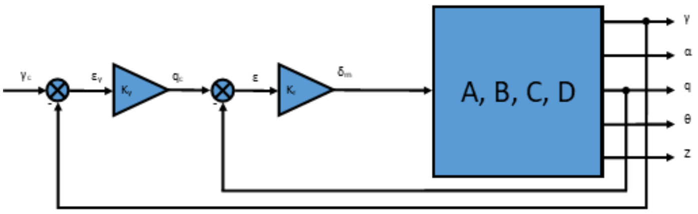


Figure 12: γ feedback loop

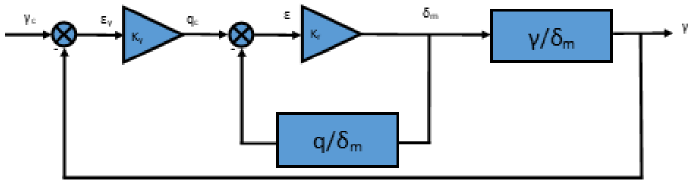


Figure 13: Simplified γ feedback loop

For the state space calculation we used the same method as the Q feedback loop which gives use the following equations :

$$A_\gamma = A_q - K_\gamma B_q C_\gamma$$

$$B_\gamma = K_\gamma B_q$$

$$C_\gamma = C_{out}$$

$$D_\gamma = 0$$

So the state space for the γ feedback loop is the following :

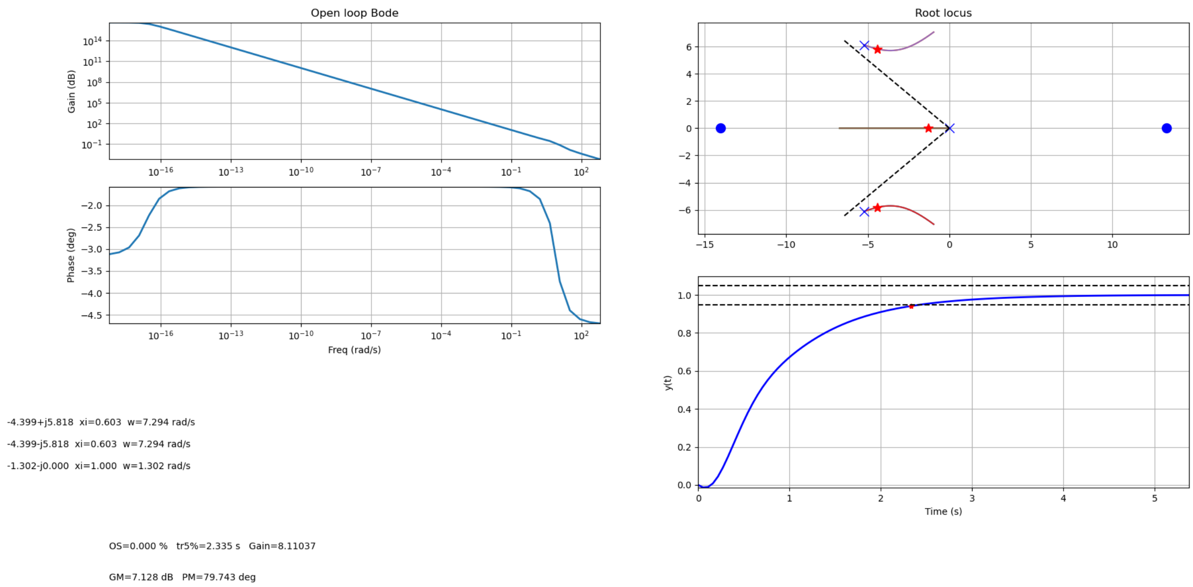


Figure 14: Sisotool of gamma feedback loop

$$A = \begin{pmatrix} 0.3701 & 1.3097 & 0.0456 & 0 & 0 \\ -0.3701 & -1.3097 & 0.9544 & 0 & 0 \\ -68.5709 & -55.4522 & -9.1599 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 435.3323 & 0 & 0 & 0 & 0 \end{pmatrix} B = \begin{pmatrix} -0.3701 \\ 0.3701 \\ 68.5709 \\ 0 \\ 0 \end{pmatrix} C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

And the transfer function is :

$$\frac{\gamma}{\gamma_c} = \frac{-0.3701s^2 - 0.261s + 69.28}{s^3 + 10.1s^2 + 64.66s + 69.28}$$

The poles and their characteristics are :

- λ : -4.399+5.818j ξ : 0.603 f : 7.294 Hz
- λ : -4.399-5.818j ξ : 0.603 f : 7.294 Hz
- λ : -1.302 ξ : 1 f : 1.302 Hz

The step response obtained is :

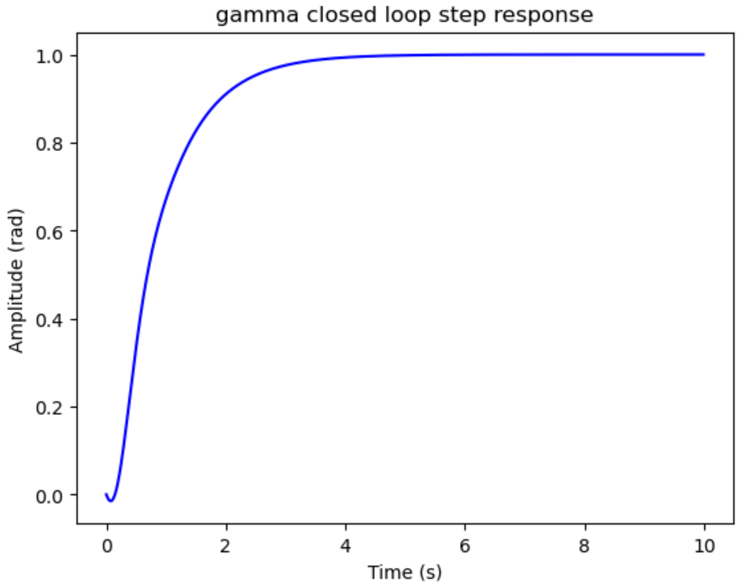


Figure 15: γ feedback loop step response

3.3 z feedback loop

In this part we will add a last feedback loop for the altitude z. To build this feedback loop we will use sisotool on the last system with two loops. This will allow us to find a gain K_z for this loop.

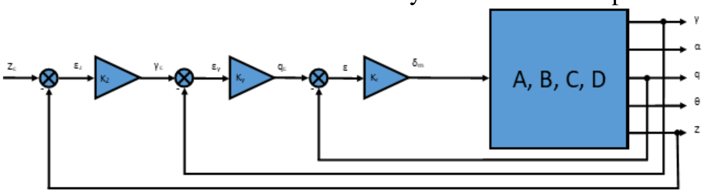


Figure 16: z feedback loop

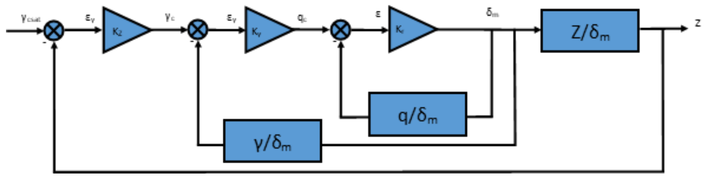


Figure 17: Simplified z feedback loop

Again we used the same principle giving us the following system:

$$A_z = A_\gamma - K_z B_\gamma C_z$$
$$B_z = K_z B_\gamma$$
$$C_z = C_{out}$$
$$D_z = 0$$

So the state space for the altitude z feedback loop is the following :

$$A = \begin{pmatrix} 0.3701 & 1.3097 & 0.0456 & 0 & 0.0005 \\ -0.3701 & -1.3097 & 0.9544 & 0 & -0.0005 \\ -68.5709 & -55.4522 & -9.1599 & 0 & -0.0837 \\ 0 & 0 & 1 & 0 & 0 \\ 435.3323 & 0 & 0 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} -0.0005 \\ 0.0005 \\ 0.0837 \\ 0 \\ 0 \end{pmatrix} \quad C = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

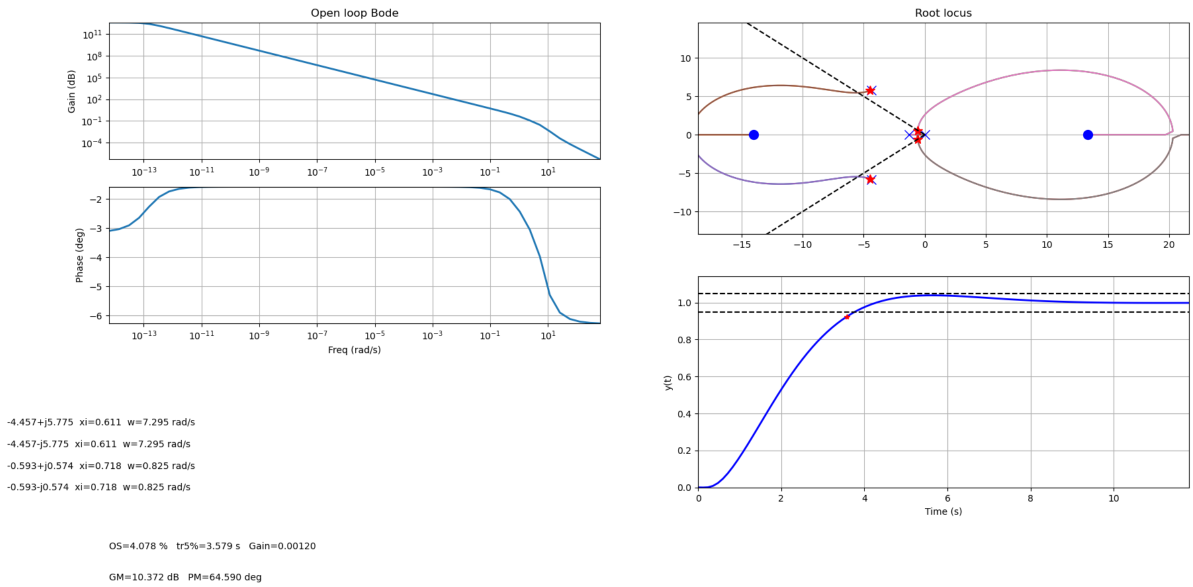


Figure 18: Sisotool of z feedback loop

And the transfer function is :

$$\frac{z}{z_c} = \frac{-0.1966s^2 + 0.1386s + 36.8}{s^4 + 10.1s^3 + 64.46s^2 + 69.14s + 36.8}$$

The poles and their characteristics are :

- λ : -4.458+5.774j ξ : 0.6111 f : 7.295 Hz
- λ : -4.458-5.774j ξ : 0.6111 f : 7.295 Hz
- λ : -0.5917+0.5842j ξ : 0.7116 f : 0.8315 Hz
- λ : -0.5917-0.5842j ξ : 0.7116 f : 0.8315 Hz

The step response obtained is :

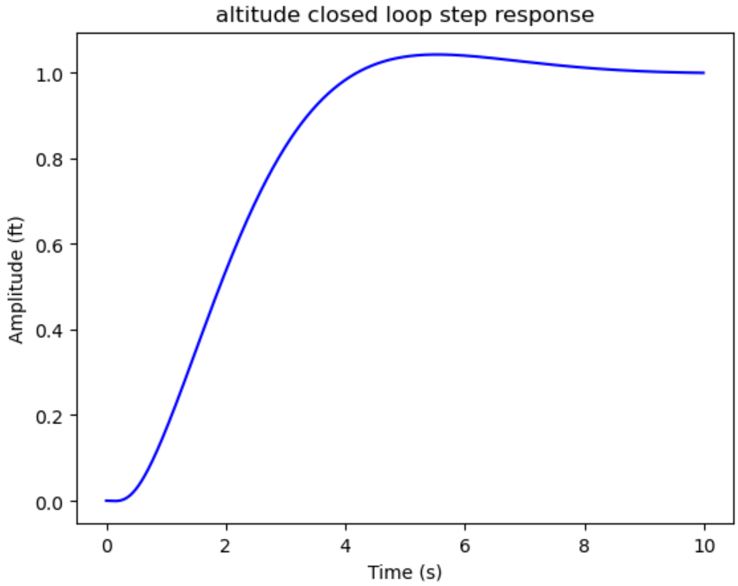


Figure 19: altitude z feedback loop step response

3.4 Saturation

The next step to build a control system on an aircraft would be to put saturation on each variable to prevent going in dangerous condition. An example is to put a saturation on the γ to block to a maximum reachable transverse load factor of $\Delta n_z = 2.8g = 27.468m/s^2$. This correspond to searching the value for which we will obtain an α_{max} that create this load factor.

$$n_z = \frac{\alpha - \alpha_0}{\alpha_{eq} - \alpha_0} = 1 + \frac{\alpha - \alpha_{eq}}{\alpha_{eq} - \alpha_0} \Rightarrow \Delta n_z = \frac{\alpha - \alpha_{eq}}{\alpha_{eq} - \alpha_0} \Rightarrow \alpha_{max} = \alpha_{eq} + (\alpha_{eq} - \alpha_0)\Delta n_z$$

SO we get $\alpha_{max} = 36.34deg$. Then we will enter a γ and see maximum of the step response of the system in α and use a bijection method until we get the result where γ_{max} give us α_{max}. At the end of the algorithm we got $\gamma_{max} = 35.51deg$. With the following step response :

3.5 Flight management

To create the script for flight management we used the model with three control loops. This allow us to control the aircraft by giving an altitude command. To do so we need to create a command u that contain a list of altitude we went to reach during a time period. And we also need to define the initial condition x0. Then we use the lsim function of control.matlab that take those entry and give us the state space vector X and Y at all time.

A first simulation correspond to an ascent phase at constant flight path angle, a cruise flight and a descent phase at constant flight path angle. We choose to create a pattern of command that combine the three phases. We created a first linspace from 10000ft to 24000ft then constant cruise at 24000ft then descent from 24000ft to 10000ft.

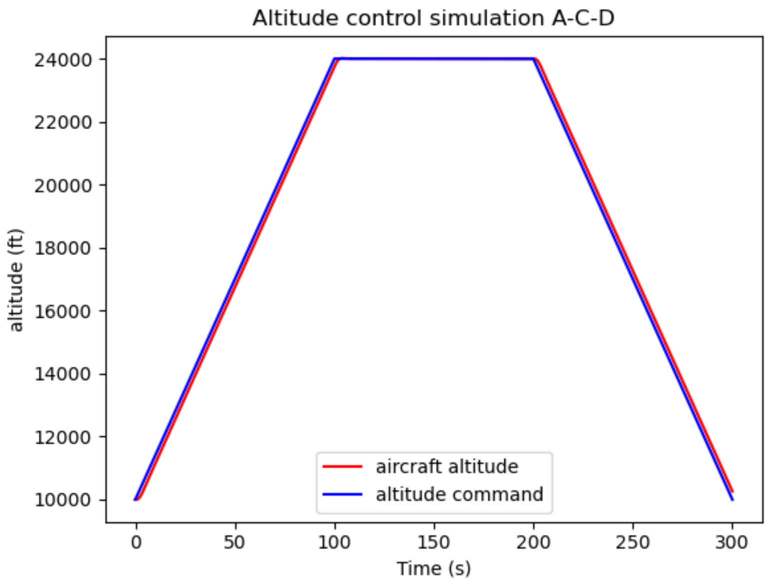


Figure 20: figure
command and response of altitude in ascent/constant/descent

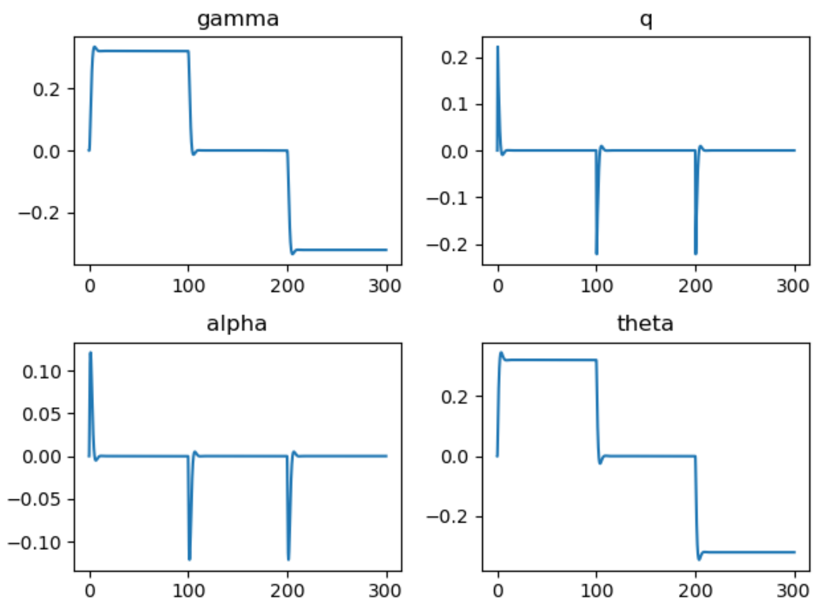


Figure 21: other variables response in ascent/constant/descent

We can see that the command are respected except we have a small delay time due to the response of the system during the switch of command type. But with altitude control we still obtain a constant flight path overall on the different trajectory.

The second trajectory is a constant altitude then a final flare. We decide to use an altitude control like on the part above. But the altitude of final flare is more complicated than the basic altitude changed we have made. We have found on a paper titled "Control and simulation of arbitrary flight trajectory-tracking" [1] an equation that can give the altitude of a flare at time t depending on the maximum decision altitude h_0 and a parameter τ that depends on multiple things like for example the runway length.

$$h(t) = h_0 \exp\left(-\frac{t}{\tau}\right)$$

As parameter we choose a classic decision altitude of 600ft and then we tried different value of τ to meet the requirement of a final flare for a fighter aircraft and we have chosen to take 10s.

$$h(t) = 600 \exp\left(-\frac{t}{10}\right)$$

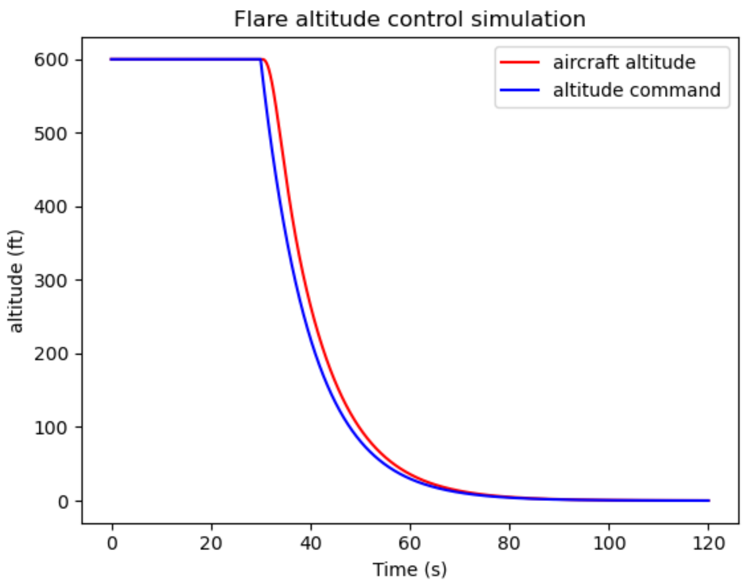


Figure 22: figure
command and response of altitude in final flare

[1]https://www.sciencedirect.com/science/article/abs/pii/S0967066104001108

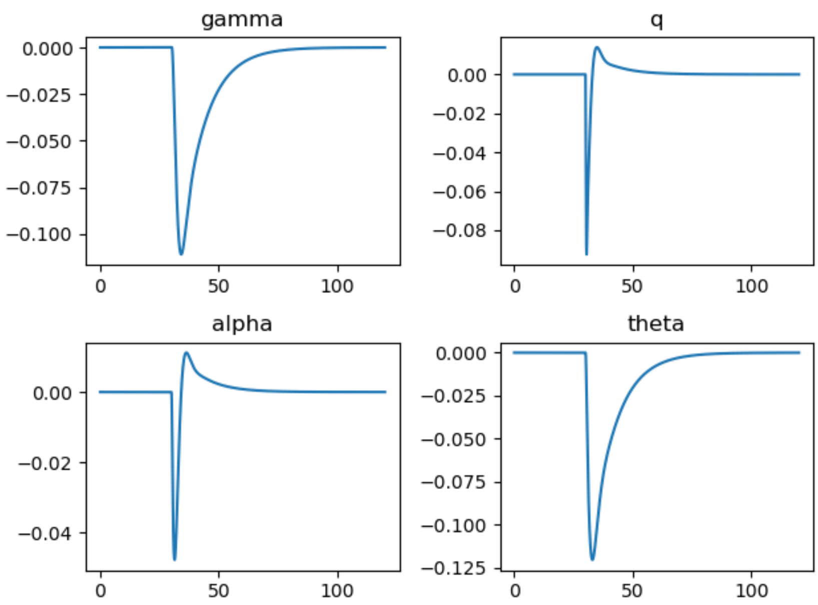


Figure 23: figure
other variables response in final flare