Hugo QUIN
Simon ALIX

CDI

Au511 - Practical Work

Control of Aircraft

Pierre Nouaille

07/01/2024

Table des matières

In this practical work, we will model an autopilot control system using Python and libraries: control, sisopy31 and slycot.

In-Flight Operating Point

We were attributed different in-flight operating points for each group during this work, defined on this table :

Alt (ft) \ Mach	0.78	0.96	1.21	1.35	1.52	1.72	1.97
510	11	12	13	14	15	16	17
2880	21	22	23	24	25	26	27
6380	31	32	33	34	35	36	37
11050	41	42	43	44	45	46	47
13500	51	52	53	54	55	56	57
15100	61	62	63	64	65	66	67
18100	71	72	73	74	75	76	77
24580	81	82	83	84	85	86	87

As group number 43, we have the following values :

Altitude : 11050 ft

Mach : 1.21

Aircraft Characteristics

The considered aircraft is a fighter aircraft of MIRAGE III class.

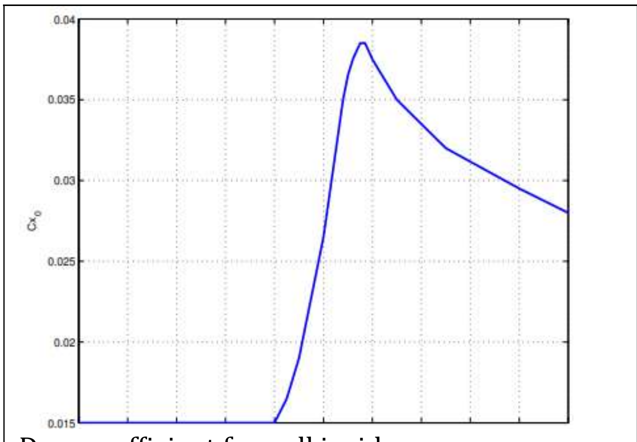
- Mass m=8400 kg
- Aircraft centering (center of gravity position) c=52 % (as % of total length)
- Reference surface (Wings) S=34 m^2
- Radius of gyration : 2,65 m
- Reference length : 5,24 m
- Total length : 3/2 * Reference length

For the calculus of air density and sound speed as a function of altitude, we will use the US standard atmosphere 76 model.

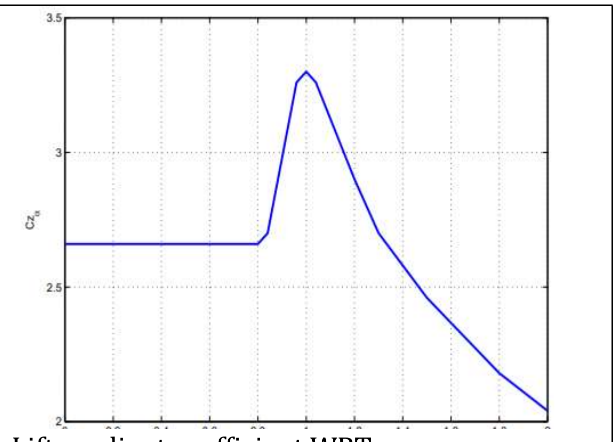
The hypothesis about the aircraft are the following :

- Symmetrical flight, in the vertical plane (null sideslip and roll)
- Thrust axis merged with aircraft longitudinal axis.
- Inertia principal axis = aircraft transverse axis (diagonal inertia matrix)
- Fin control loop: its dynamics will be neglected for the controller synthesis.
- The altitude sensor is modelled by a 1st order transfer function with a time constant τ = 0.75 s

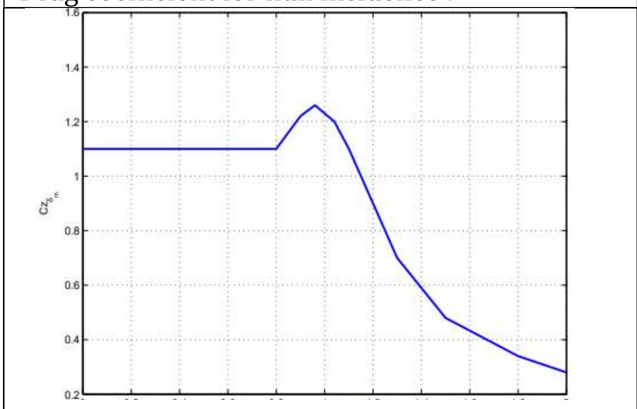
We will now define the aircraft aerodynamic coefficients for the longitudinal motion according to our operation point, referring to the given functions in the subject.

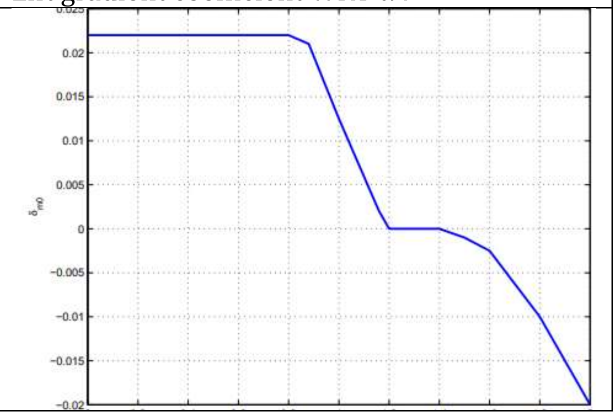


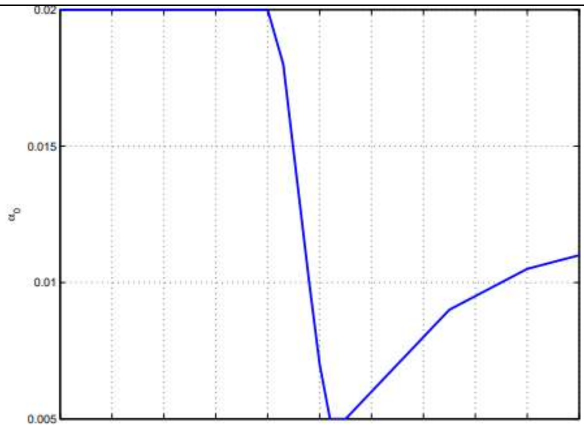
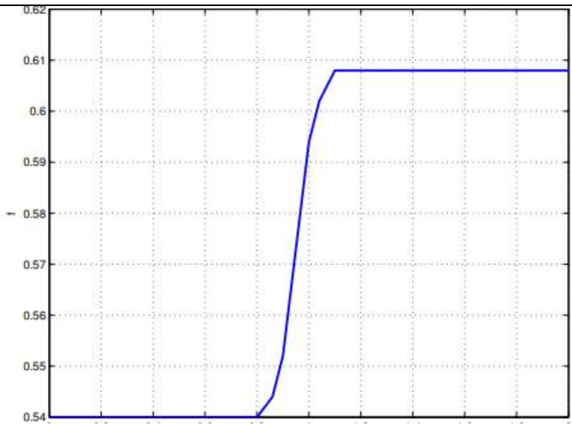
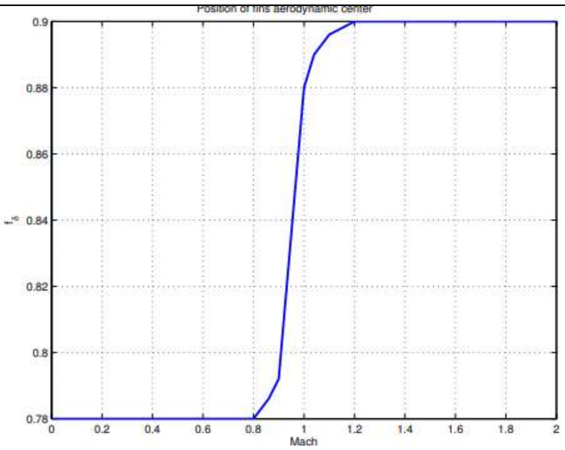
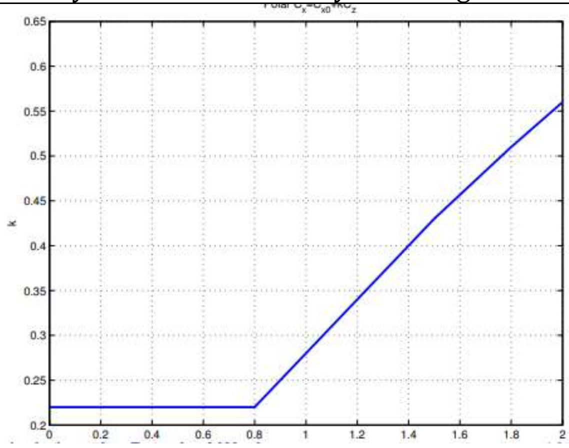
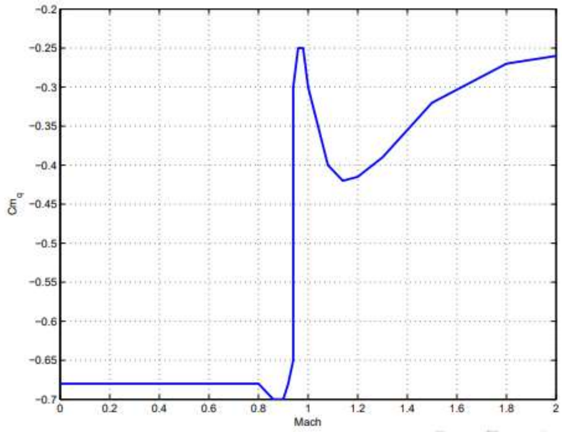
Drag coefficient for null incidence :



Lift gradient coefficient WRT α :

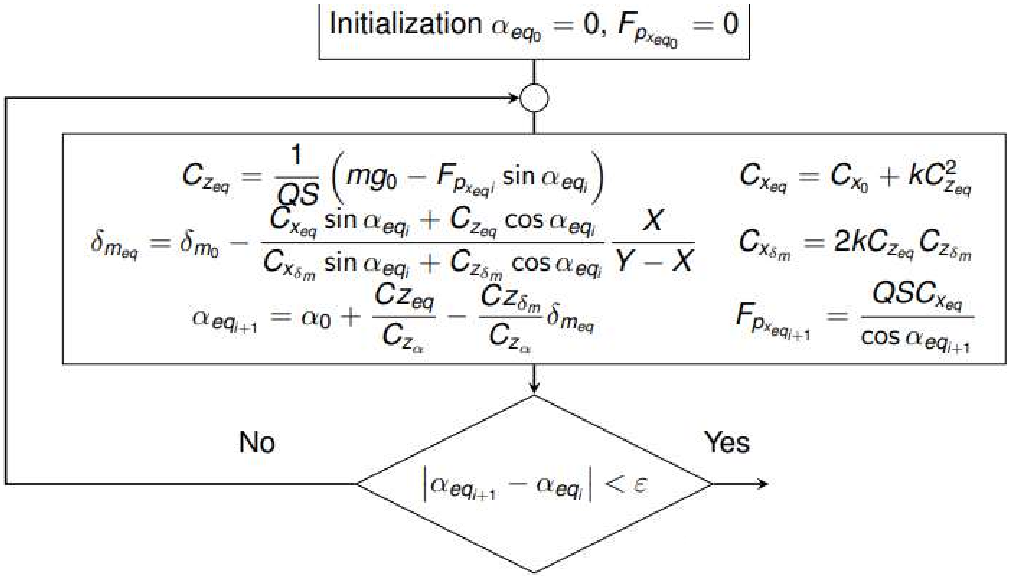




Lift gradient coefficient δm :	Equilibrium fin deflection for null lift :
 Incidence for null lift and fin deflection	 Aerodynamic center of body and wings
 Aerodynamic center of fins :	 Polar coefficient
 Damping coefficient	

Study of the Uncontrolled Aircraft

To study the equilibrium point of the aircraft, we will implement the following algorithm as a python function.



We can now computev the matrices A and B of the state space representation of the model with the following coefficient formulas :

$$X_V = \frac{2QSC_{X_{eq}}}{mV_{eq}} \qquad m_V = 0 \qquad Z_V = \frac{2QSC_{Z_{eq}}}{mV_{eq}} \approx \frac{2g_0}{V_{eq}}$$

$$X_\alpha = \frac{F_{eq}}{mV_{eq}} \sin\alpha_{eq} + \frac{QSC_{X_\alpha}}{mV_{eq}} \qquad m_\alpha = \frac{QS\ell_{ref} C_{m_\alpha}}{I_{YY}} \qquad Z_\alpha = \frac{F_{eq}}{mV_{eq}} \cos\alpha_{eq} + \frac{QSC_{Z_\alpha}}{mV_{eq}}$$

$$X_\gamma = \frac{g_0 \cos\gamma_{eq}}{V_{eq}} \qquad m_q = \frac{QS\ell_{ref}^2 C_{m_q}}{V_{eq} I_{YY}} \qquad Z_\gamma = \frac{g_0 \sin\gamma_{eq}}{V_{eq}}$$

$$X_{\delta_m} = \frac{QSC_{X_{\delta m}}}{mV_{eq}} \qquad m_{\delta_m} = \frac{QS\ell_{ref} C_{m_{\delta m}}}{I_{YY}} \qquad Z_{\delta_m} = \frac{QSC_{Z_{\delta m}}}{mV_{eq}}$$

$$X_\tau = -\frac{F_\tau \cos\alpha_{eq}}{mV_{eq}} \qquad Z_\tau = \frac{F_\tau \sin\alpha_{eq}}{mV_{eq}}$$

With this, we find :

```
A = [[ -0.0705   -0.0239   -0.0436     0.         0.      0.   ]      B = [[   0.   ]
     [  0.047     0.        2.7032     0.         0.      0.   ]           [   0.8893]
     [ -0.047     0.       -2.7032     1.         0.      0.   ]           [  -0.8893]
     [  0.        0.     -107.8169    -1.8545     0.      0.   ]           [-155.191 ]
     [  0.        0.        0.         1.         0.      0.   ]           [   0.    ]
     [  0.      410.2523    0.         0.         0.      0.   ]]          [   0.    ]]
```

We can now compute the poles, damping factors and proper pulsations using the “damp” function from the control library.

Poles	Damping	Proper Pulsation
(0; 0)	ξ=1.0	w=0 rad/s
(-2.279+j10.375; -2.279-j10.375)	ξ=0.215	w=10.622 rad/s
(-0.0462;	ξ=1.0	w=0.0462 rad/s
-0.0243)	ξ=1.0	w=0.0243 rad/s

We find that the system possesses a fast mode (short period) at 10.622 rad/s with a low damping ratio at 0.215.

The system also possesses a slow mode (phugoid) at approximately 0.03 rad/s with a damping ratio at 1.

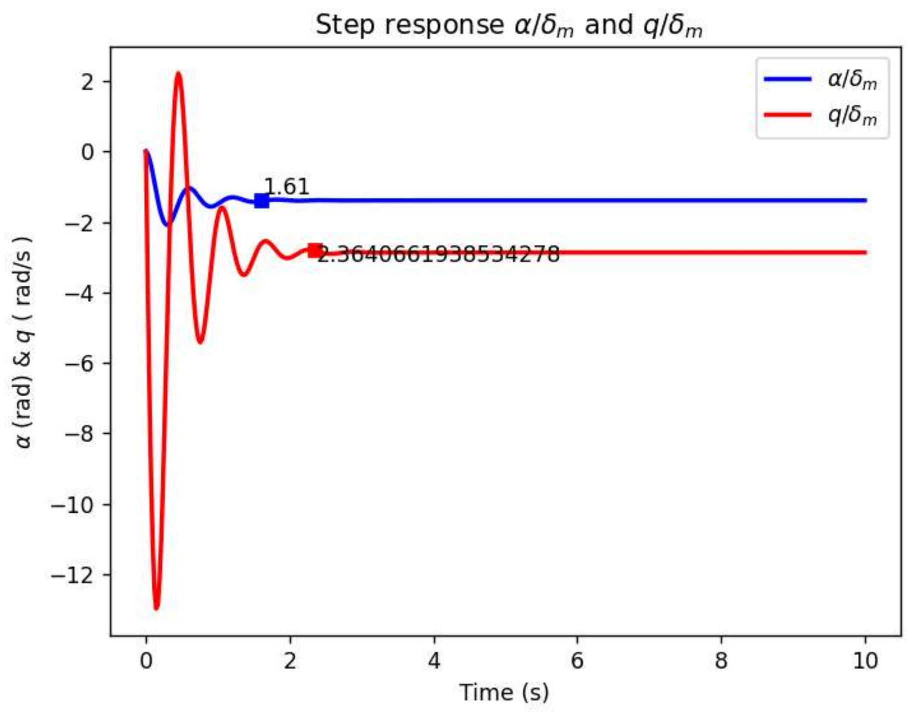
In Short-Period mode

Transfer function q / delta_m

$$\frac{-155.2\,s - 323.6}{s^2 + 4.558\,s + 112.8}$$

Transfer function alpha / delta_m

$$\frac{-0.8893\,s - 156.8}{s^2 + 4.558\,s + 112.8}$$



On this graph, we can identify the settling times for each variable.

- α settling time 5% = 1.607565 s
- q Settling time 5% = 2.364066 s

We can see that for the short-period mode, the step responses are very unstable, with a relatively slow settling time.

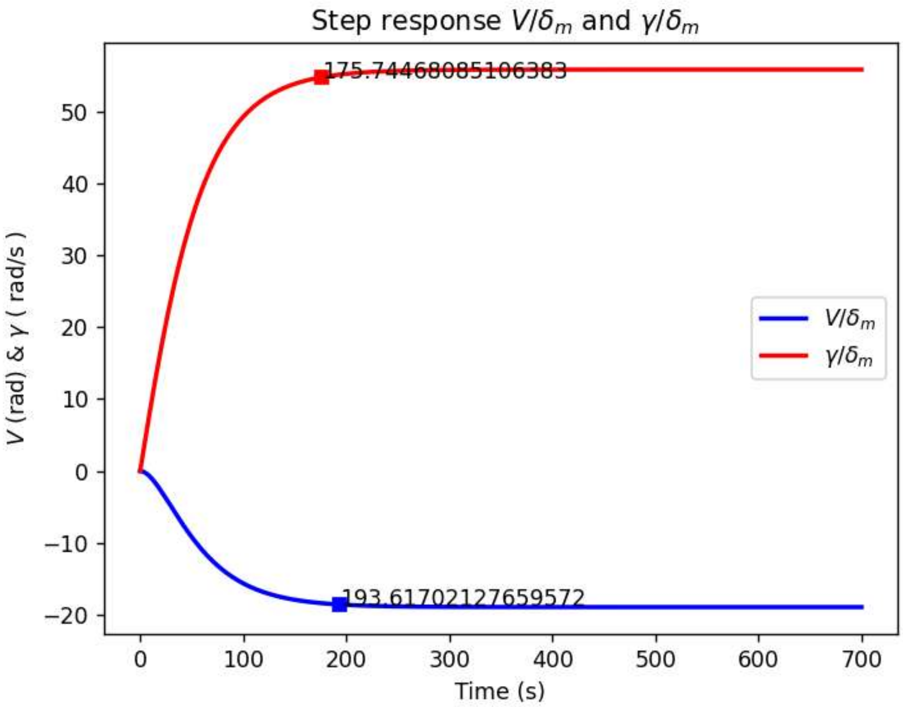
In Phugoid mode

Transfer function v/delta_m

$$\frac{-0.02127}{s^2 + 0.07054\,s + 0.001124}$$

Transfer function γ/delta_m =

$$\frac{0.8893\,s + 0.06273}{s^2 + 0.07054\,s + 0.001124}$$



On this graph :

- v settling time 5% = 193.617021 s
- γ settling time 5% = 175.744681 s

For the phugoid mode, the step response is excessively unaccurate, unstable, and with a terribly long settling time.

Every variable is either wrong or unstable. This is why we will need to add control feedback loops to our open loop system.

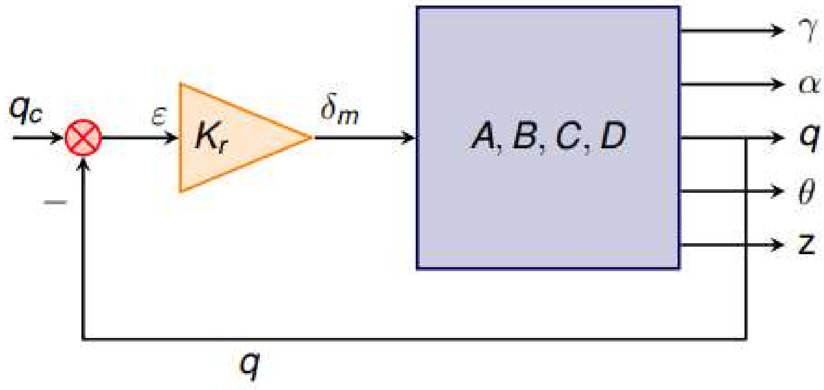
Before we go onto the next step, we can also remove the speed from the state vector, as its variation is null in our case of study. This gives us :

$$X = \begin{pmatrix} \gamma & \alpha & q & \theta & z \end{pmatrix}^T$$

Controllers Synthesis

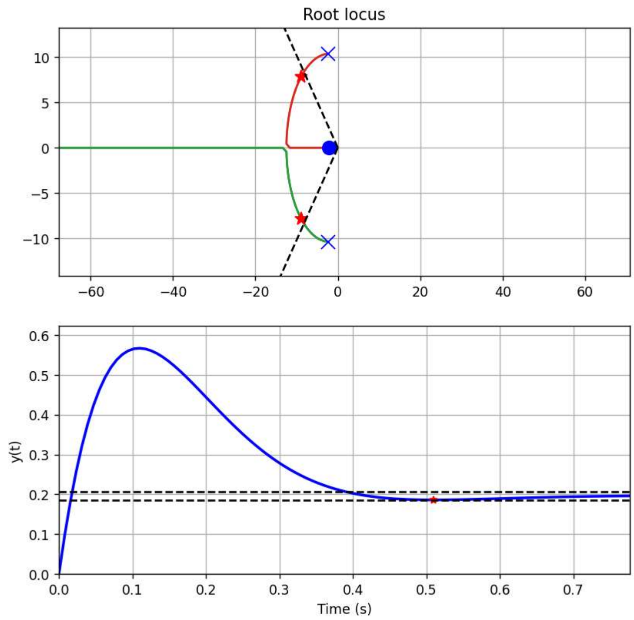
The *q* Feedback Loop

We start to build our control system by adding a gyrometric feedback loop on q.



To implement the closed loop, we need to find a gain Kr to compute the new closed loop state space representation.

We do this by first using the root locus (function part of the slycot and control libraries) to find the correct gain and damping coefficients. This method is much more practical and allows us to skip all the complicated calculations with the transfer functions.



With this method, we search for a gain value to obtain a damping ratio of 0.75.

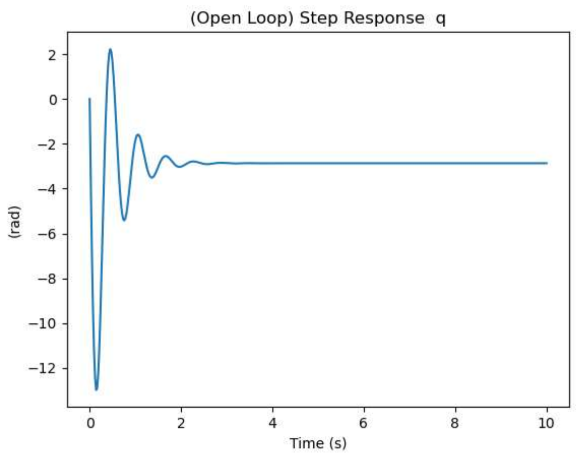
With this, we find Kr = -0.08528.

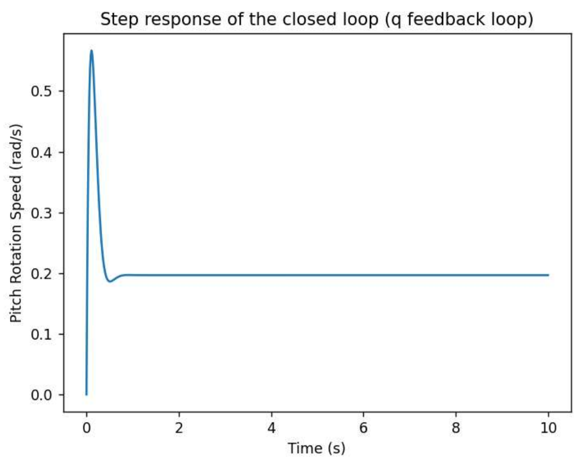
We can use this value to compute the new closed loop state space representation with :

$$A_k = A - K_r B C_q$$
$$B_k = K_r B$$
$$C_k = C_{out} = C_q$$
$$D_k = K_r D$$

State space representation of closed loop.

```
Aq : [[   0.        2.7032    0.0758    0.       0.   ]
      [   0.       -2.7032    0.9242    0.       0.   ]
      [   0.     -107.8169  -15.0894    0.       0.   ]
      [   0.        0.        1.        0.       0.   ]
      [ 410.2523    0.        0.        0.       0.   ]]
Bq : [[-0.0758]
      [ 0.0758]
      [13.2349]
      [-0.    ]
      [-0.    ]]
```

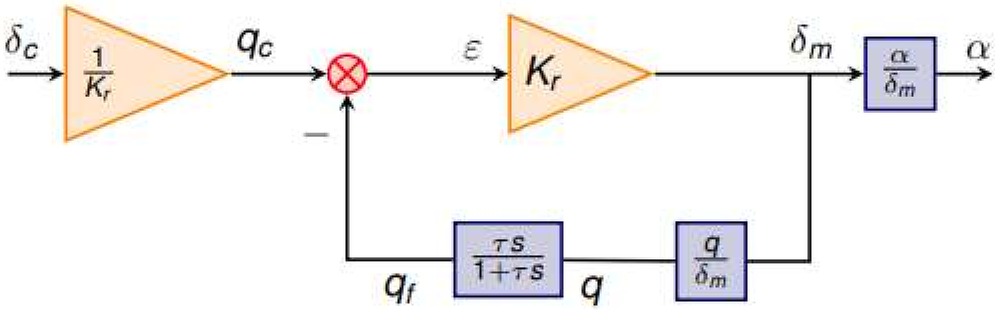




By implementing the q feedback loop, this is the step response we obtain on the right, compared with the uncontrolled response on the left.

Even though it still has a considerable overshoot, the response is stable, and the settling time is short.

Another method to stabilize the system is to use a washout filter. We will follow this example only for q.

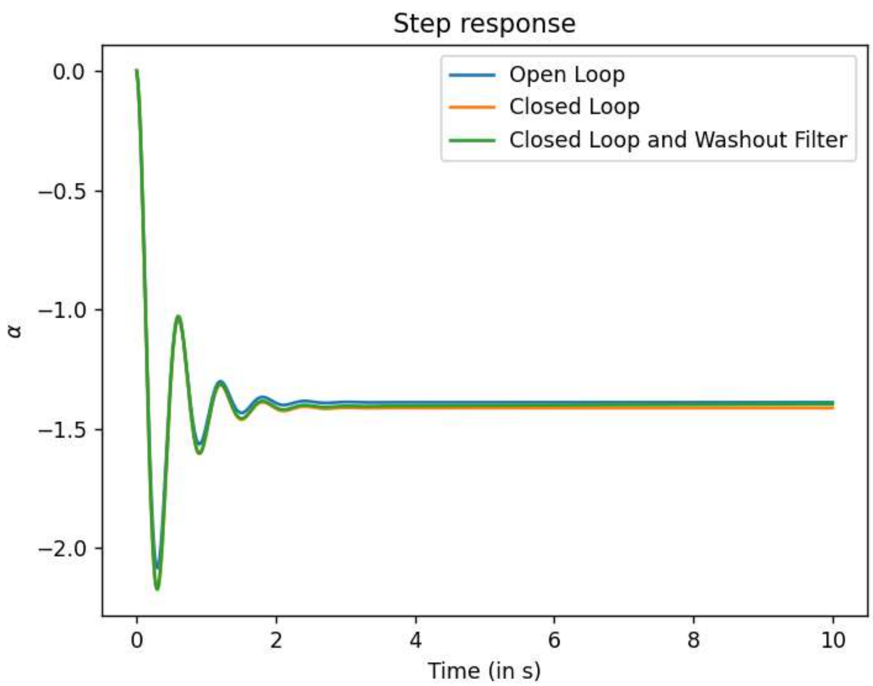


This method goes by defining a constant τ for which the washout filter is added.

The wahsout filter is defined by :

$$\frac{\tau s}{1+\tau s}$$

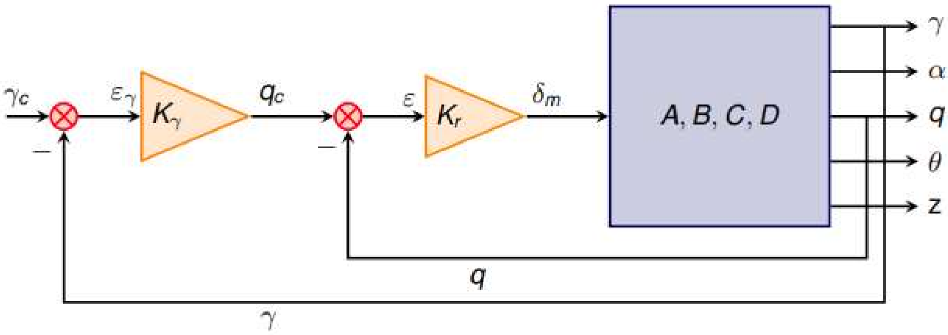
It allows to have the same steady state gain for α with or without the q feedback.



As we can see on this graph, the system with the washout filter (green curve) does not seem to be of very good use here, as there isn't much difference with the uncontrolled response. However, these results could probably be wrong due to our misunderstanding on how to implement the washout filter.

The γ Feedback Loop

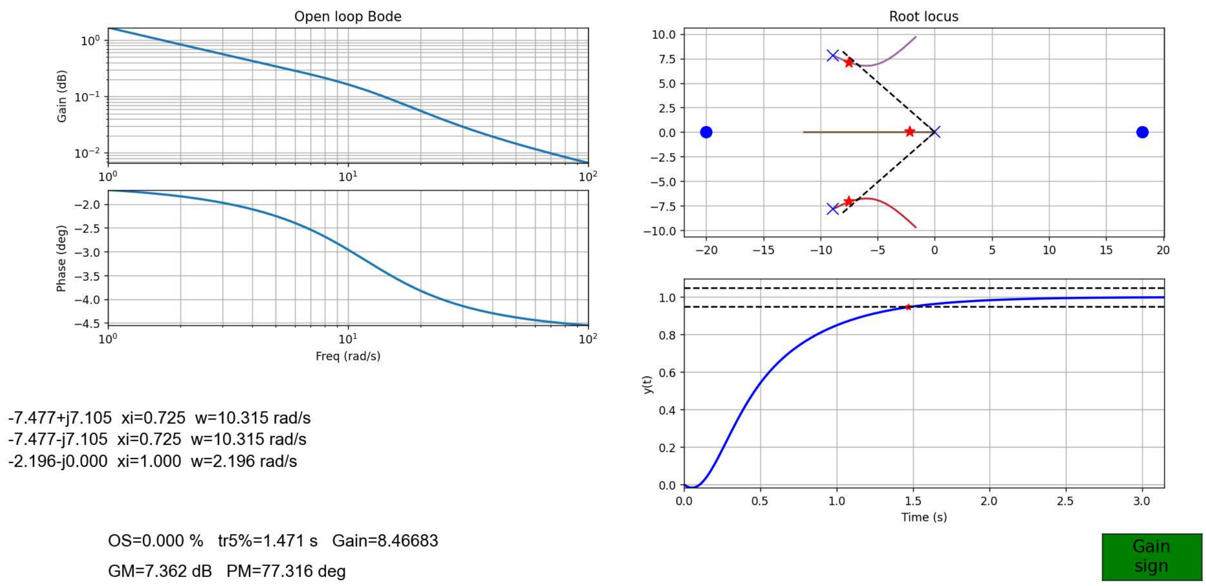
In the same way as previously with q, we now add a flight path angle feedback loop on γ as we look for the gain K_γ. We still consider that the auto-throttle perfectly ensures that the speed is constant, so $\dot{v} = 0$.



We start by finding a gain respecting the following requirements :

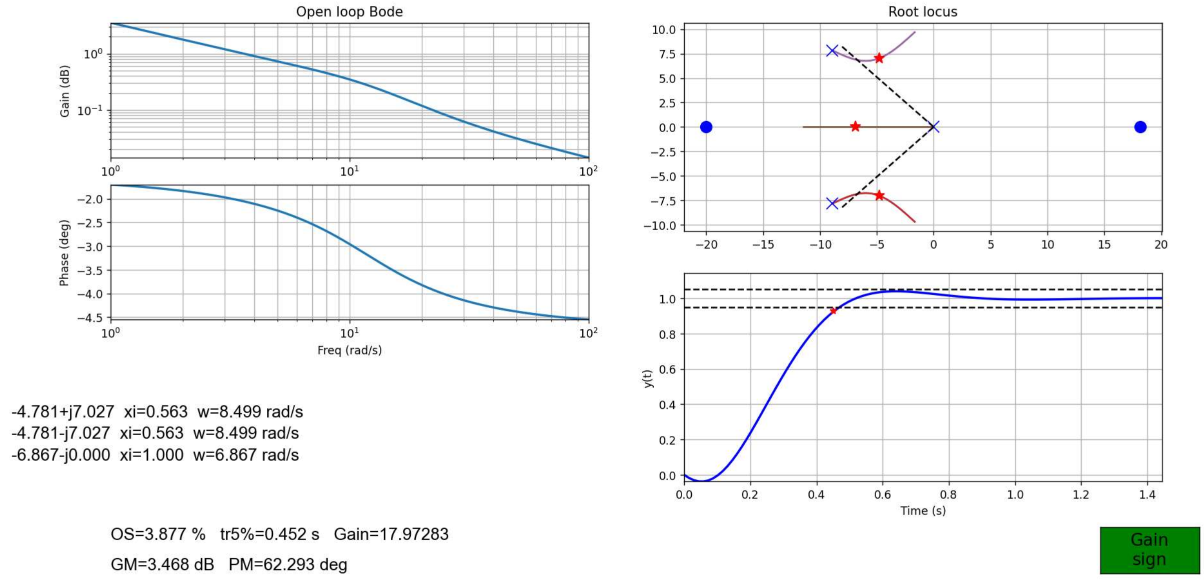
- Gain Magrin > 7dB
- Phase Margin > 45°
- Minimum tr5% (settling time)

With a gain $K_\gamma = 8.4668$, the requirements are respected, and we get an time response of 1.471 seconds.

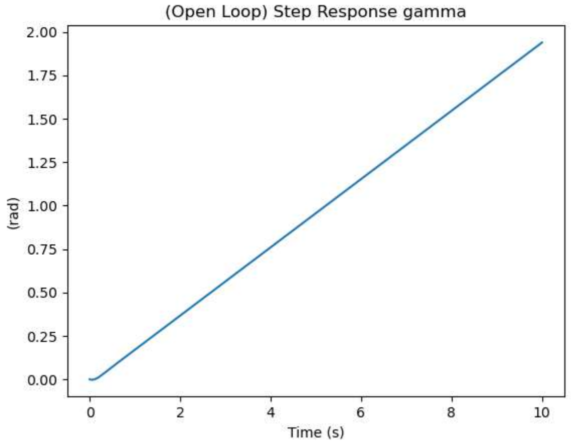


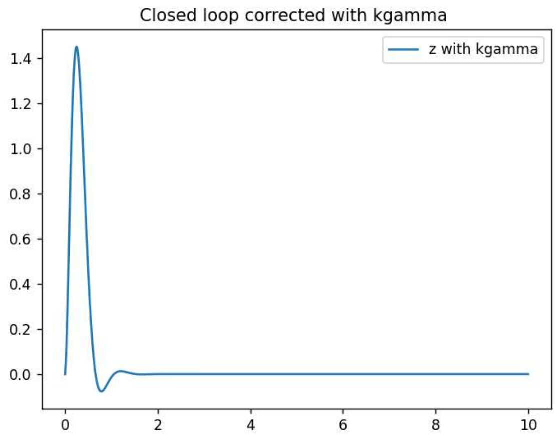
As we can see on the graph, the response is precise and stable. However, the settling time can still be improved. This is why we will look for a second tuning, this time adding new requirements :

- Maximum overshoot < 5%
- ξ > 0.5



By minimizing settling time to 0.452 seconds, we have an overshoot of 3.877%. The response is now stable, precise, and fast.





We can also visualise the state space representation and transfer function :

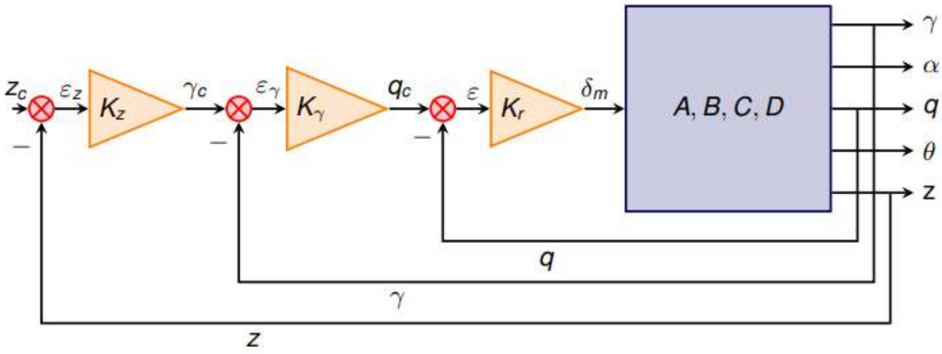
$$A_\gamma : \quad \begin{bmatrix} 1.3631 & 2.7032 & 0.0758 & 0. & 0. \\ -1.3631 & -2.7032 & 0.9242 & 0. & 0. \\ -237.8687 & -107.8169 & -15.0894 & 0. & 0. \\ 0. & 0. & 1. & 0. & 0. \\ 410.2523 & 0. & 0. & 0. & 0. \end{bmatrix} \qquad B_\gamma : \quad \begin{bmatrix} -1.3631 \\ 1.3631 \\ 237.8687 \\ -0. \\ -0. \end{bmatrix}$$

Transfer function :

$$\frac{-2.192\,s^2 - 4.065\,s + 797.7}{s^4 + 16.43\,s^3 + 135.7\,s^2 + 492\,s + 797.7}$$

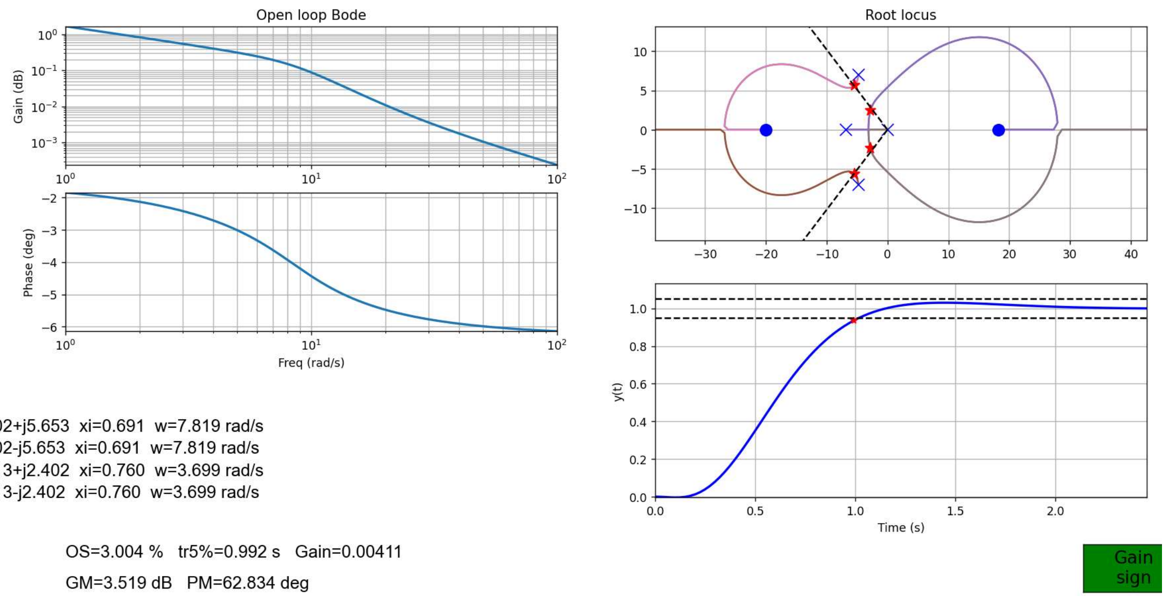
The z feedback loop

Finally, while keeping both previous tunings (Kr and K_γ), we add a final feedback loop on z.

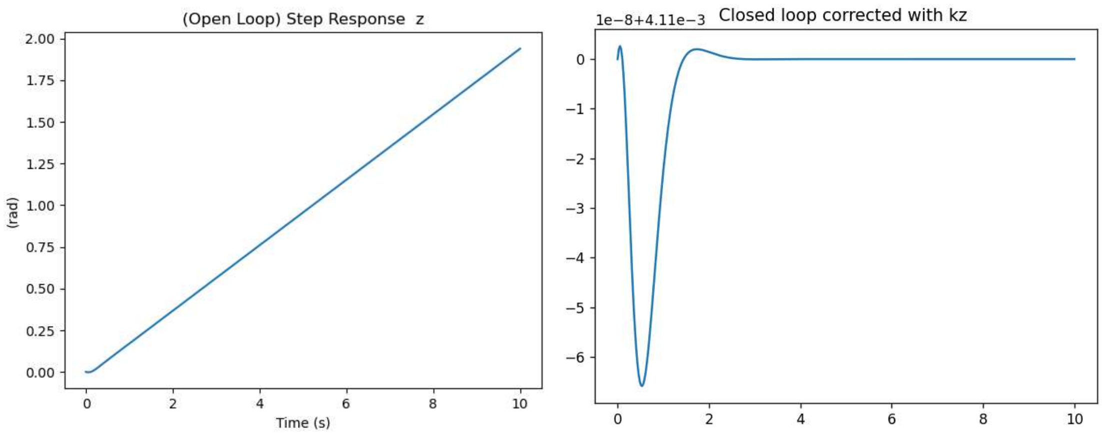


For this final feedback loop, the expected performances are :

- Overshoot $D_1 < 5\%$
- Minimum settling time (tr5%)
- $\xi > 0.5$



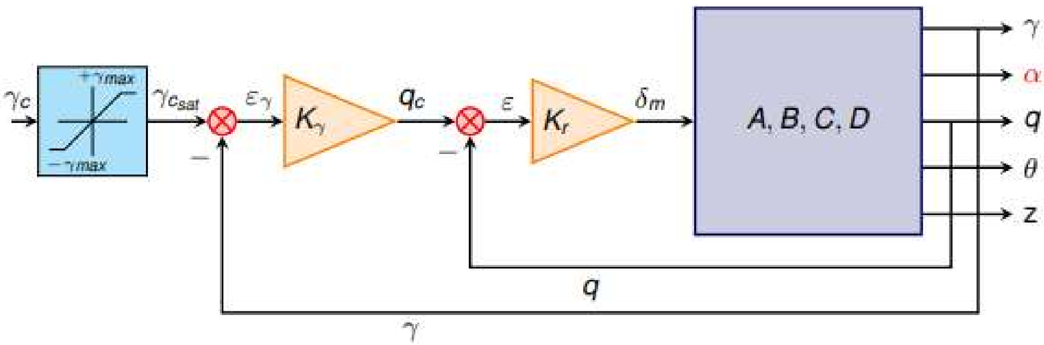
By meeting all the requirements, we get Kz = 0.00411 and manage to get a settling time just under 1 second. (0.992 second)



Here, we can observe our corrected step response. The results seem unrealistically good, as we only have a microscopic overshoot.

Saturation in the γ control loop

A saturation is added at the input of the γ feedback loop. In this question, we are going to determine the value of $\gamma_{C_{sat}}$, but we will not implement the non-linear simulation of the saturated autopilot.



To do so, we will need to build the state space representation of the closed loop between $\gamma_{C_{sat}}$ and α, which includes both q and γ feedback loops.

We will evaluate α_{max} knowing the maximum transverse load factor that we need is $\Delta n_z = 3.1\ g$.

With this we can find a saturation value γ_{max} of the flight path angle using the Newton method :

$$\gamma_{max} = 14.282689952077863\,°$$

Even though this method works, there are easier ways to find the saturation value of the flight path angle. For this type of variable, a graphical method, for example, would work just as well. Since we do not need a perfectly accurate value as we are looking for a maximum, we could plot the whole function and find the maximum graphically.

Synthesis with a new Centre of Gravity (C.O.G)

We define a new position of the centre of gravity of the aircraft by modifying the value of c.
c=f*1.1

It modifies the values of X and Y, and therefore changes the state space representation of the aircraft.

The new state space representation we get is :

A = [[0. 2.7032 0. 0. 0.]
[0. -2.7032 1. 0. 0.]
[0. -107.8169 -1.8545 0. 0.]
[0. 0. 1. 0. 0.]
[410.2523 0. 0. 0. 0.]]

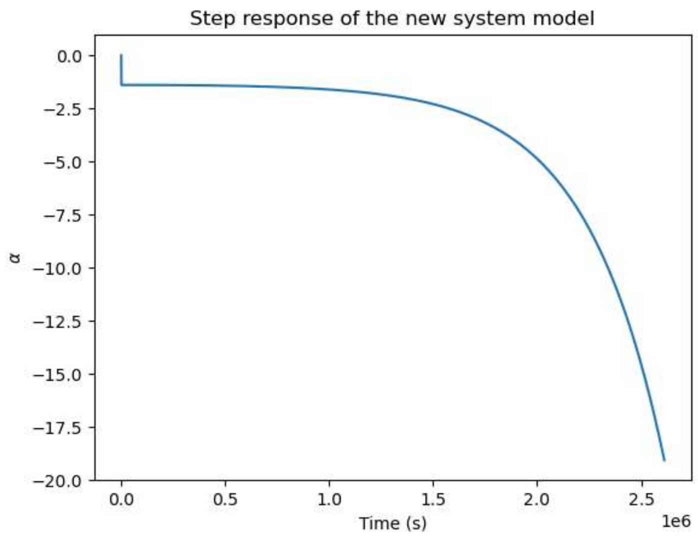
B = [[0.8893]
[-0.8893]
[-155.191]
[0.]
[0.]]

And the transfer function is the following :

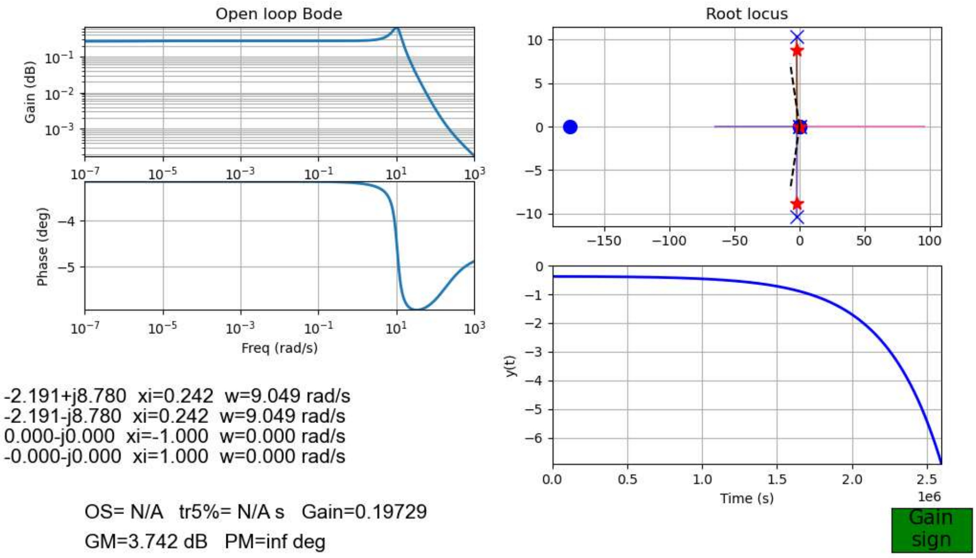
$$\frac{-0.8893\, s^3 - 156.8\, s^2 + 2.138 10^{-1}\, s + 1.07.10^{-0}}{s^4 + 4.558\, s^3 + 112.8\, s^2 - 4.682 10^{-11}\, s - 7.896.10^{-10}}$$

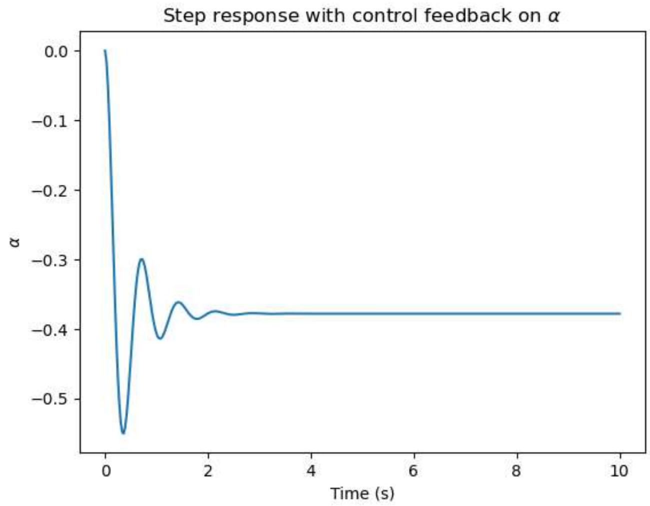
We can see that we obtain almost the same damping and poles as the previous model.

Poles	Damping	Proper Pulsation
(0; 0)	ξ=1.0	w=0 rad/s
(-2.279+j10.37; -2.279-j10.37)	ξ=0.215	w=10.62 rad/s



The model is once again unstable and diverges. We will need to find a tuning for Kα.





With Kα = 0.19729, we have a pulsation of 9 rad/s, and the system is corrected with still some flaws, such as a high overshoot, and quite a slow settling time.

Conclusion

In this practical work, even though we were unable to reach the end because of the very long start we had and difficulties of installing libraries, functions not working properly, we finally were able to produce and deliver the analysis and correction/automatization for output variables of an aircraft.

We did so by calculating, for each variable, the step responses and then correcting them by using closed loop state space representation method and by tuning parameters accordingly.