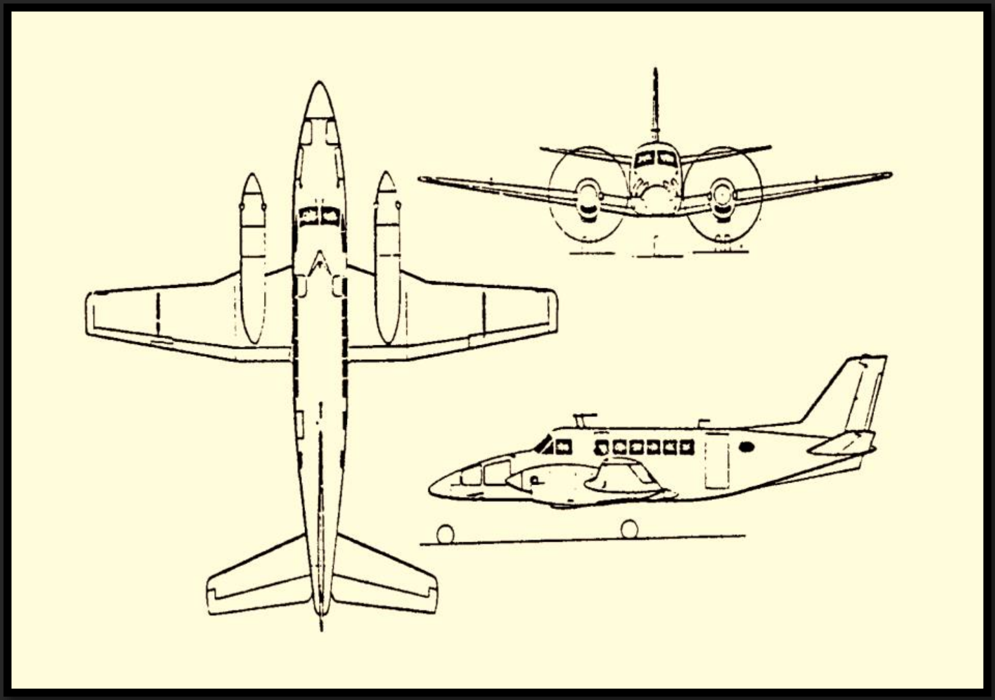
Homework of Dynamic Stability
Airplane B (case 3 – Normal cruise)



Supervised by: Professor Yazigi.

Student: Chaouki Zakaria.

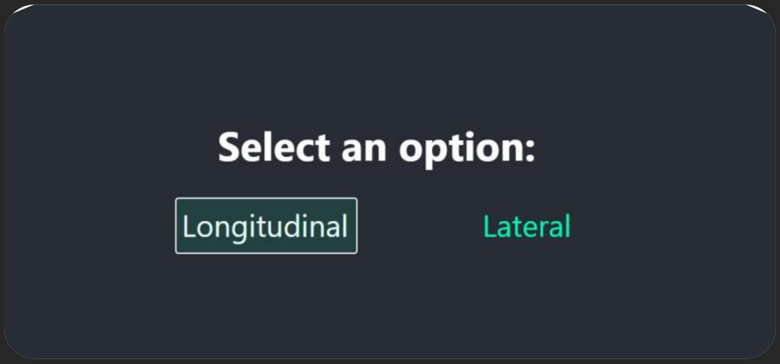
2023

Table of contents

Introduction

The main goal of this project is to make calculations on aircraft to know if it is stable or not accordingly with the lateral and longitudinal axis.
To this purpose, we will explain in details all the steps to do so and some graphical representations will complete those explanations.

First, I would like to thank the teacher for this project which allowed me to put my practical skills in programming as well as all the theory learnt with the different courses into a small application called AS-APP (Aircraft Stability Application).



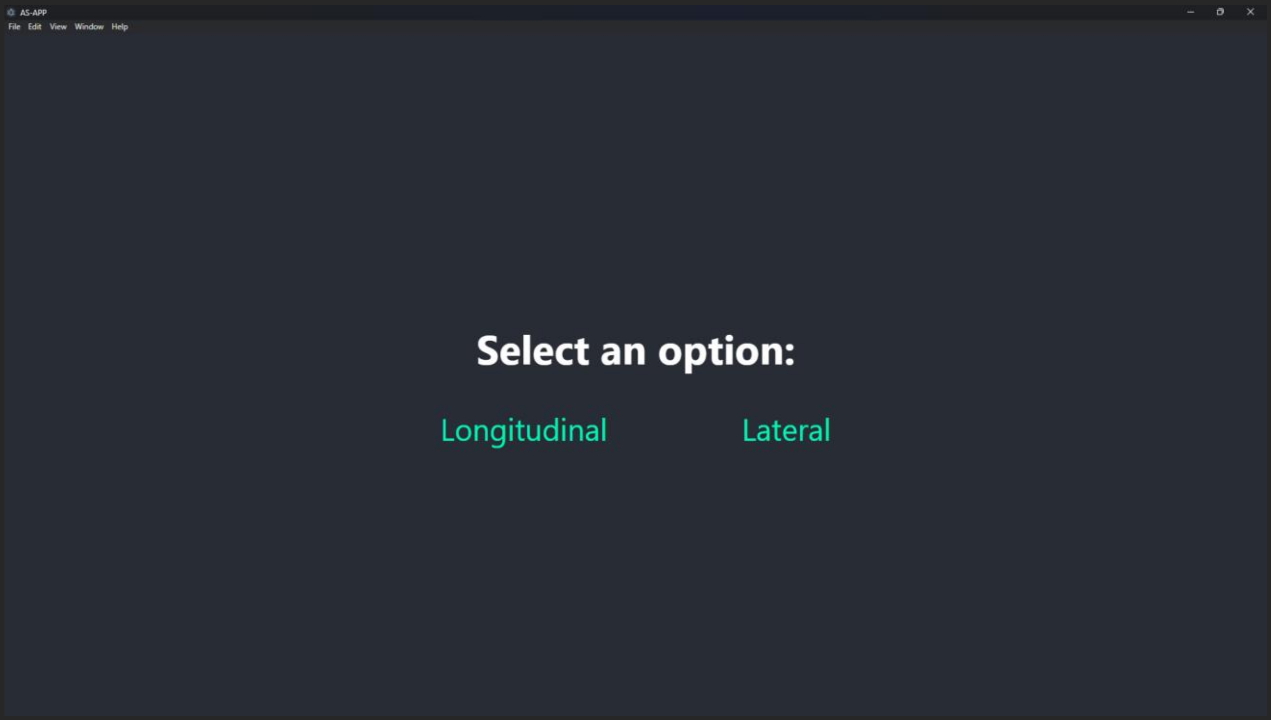
Let's introduce all the steps required for the calculation process.

We study the cruise mode conditions with steady flight conditions (stability right before perturbations). To do so we need to calculate multiple derivatives parameters with respect to the longitudinal and lateral axis. They are mainly linked to the elevator, rudder, and throttle settings. However, as we are in steady flight conditions, throttle settings will not be changing on cruise mode.

In this task we will be working on a small airplane with 2 rotors and a max capacity of 19 passengers.
The calculation example will be made on a normal cruise flight. All the data can be seen in the annexes of this report.

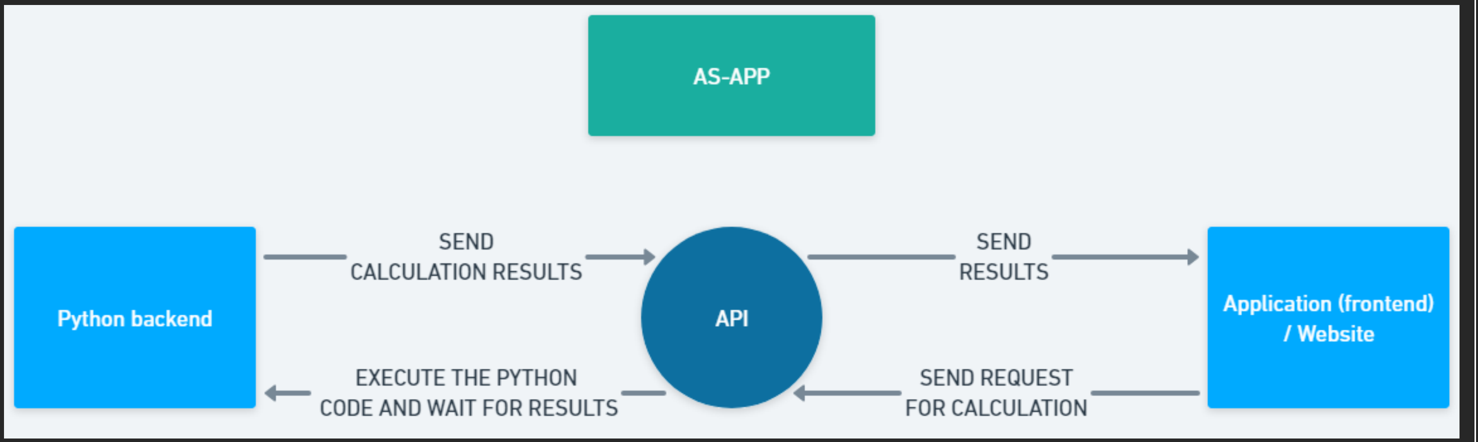
Desktop application for aircraft stability calculation

Thanks to this homework, I was able to develop an application called AS-APP (Aircraft Stability APP).

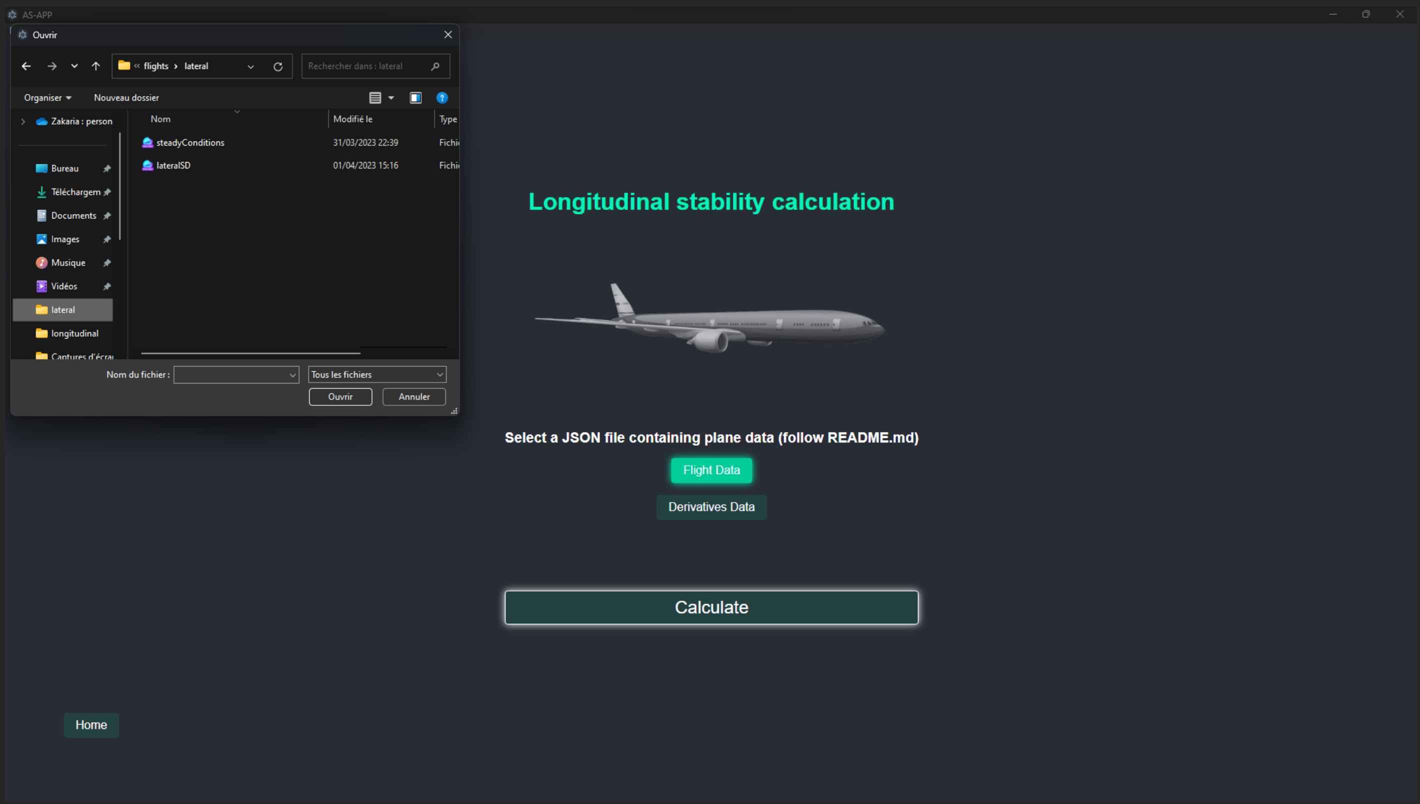


This application was built using Javascript and Python. I used the popular framework react JS which allows to create website as well as application (thanks to electron JS which allows to have a cross-platform desktop application).

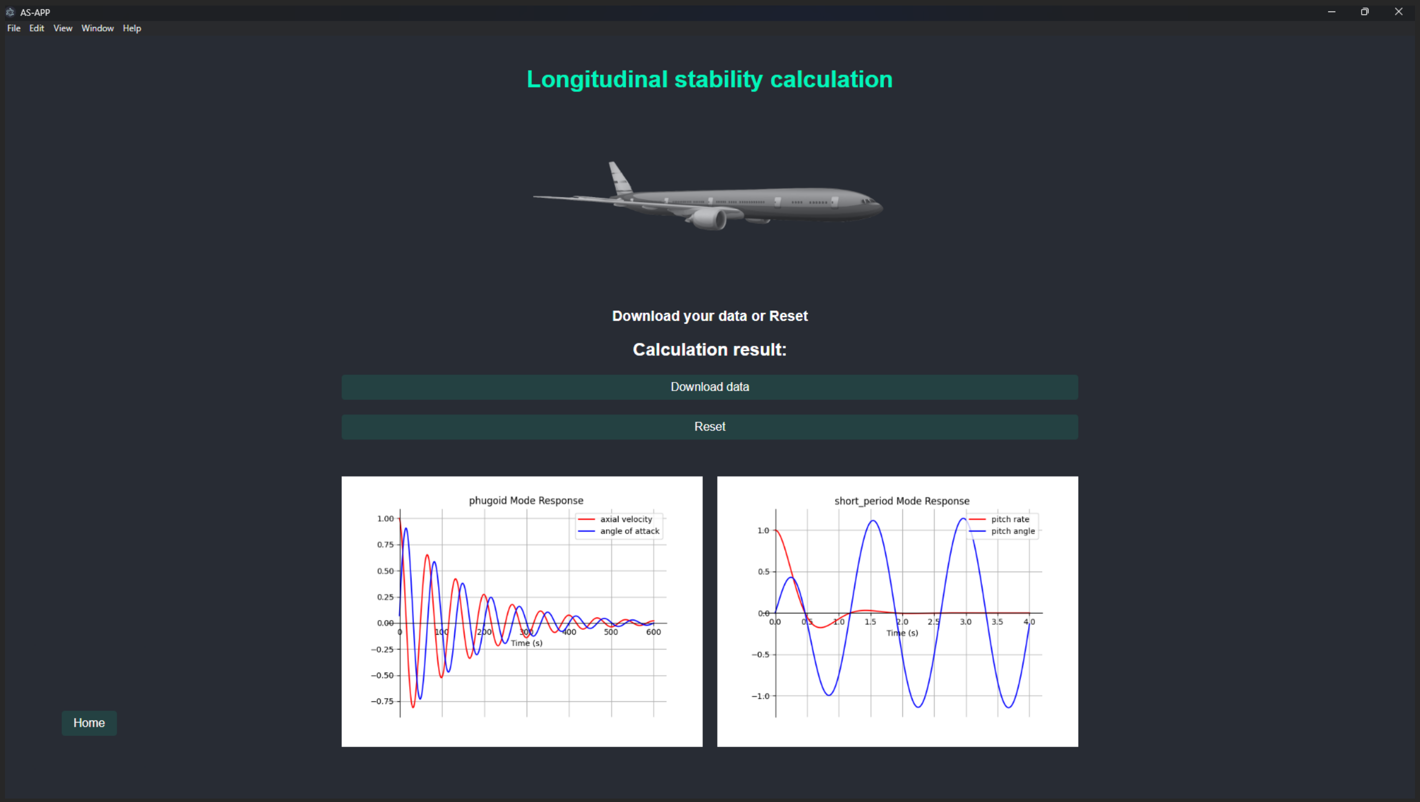
The AS-APP was built following those steps:



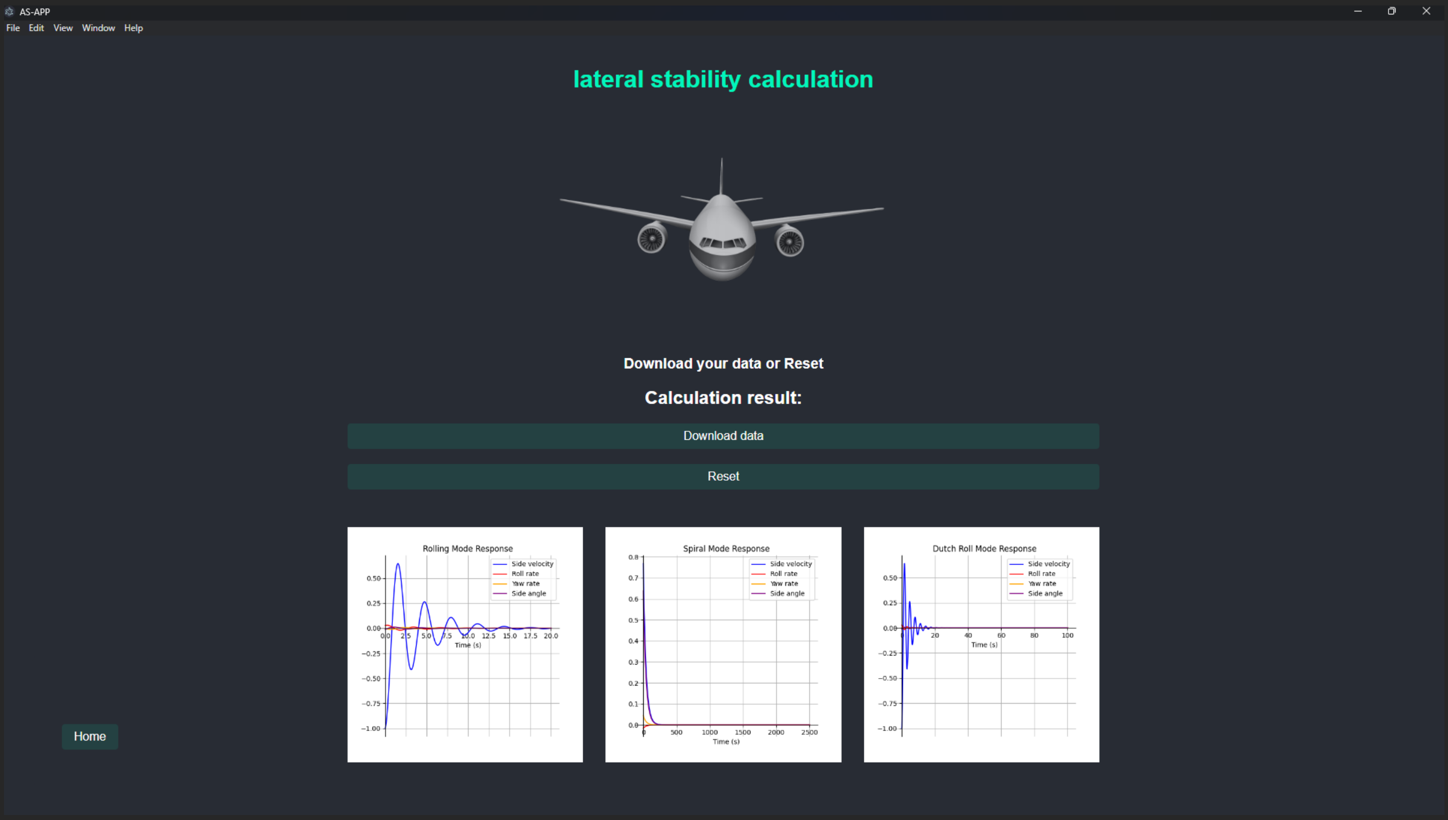
For longitudinal calculation we get:



When clicking on longitudinal we obtain the above page. We can then select our data files with the two buttons **flight data** and **derivatives data.** However, the files must respect the template given in the source code **(api/flights/)**



For lateral calculation we get:



As for longitudinal calculation, we can download all data in a .txt format. You can find all that data at the end of the report. Also, everything will be detailed in this report.

The full project can be found here: https://github.com/PhantHive/aircraft-stability

Please go to the next page (Jump to page 11 if you desire to see the python calculation).

PART I – Longitudinal Dynamic Stability of an Airplane

1. Aircraft Matrix (A) and Control Matrix (B)

In this first part we will be calculating both A and B matrices. To do so, let's start with calculating the $\bar{q}$ the mean dynamic pressure.

$$\bar{q} = \frac{1}{2} \cdot \rho \cdot {U_0}^2$$

N.A (Numerical Application):

$$\bar{q} = \frac{1}{2} \cdot 0.654 \cdot 137.2^2$$

So,

$$\bar{q} = 6155.39568$$

a. *Aircraft Matrix*

Now we need to dress the table of longitudinal stability derivatives.

We will base our calculations on those formulas given in the course:

Equation	*X*	*Z*	*M*
u	$X_u = -\frac{\bar{q}\,S}{mU_0}\left(2C_{D_0} + C_{D_u}\right)$	$Z_u = -\frac{\bar{q}\,S}{mU_0}\left(2C_{L_0} + C_{L_u}\right)$	$M_u = \frac{\bar{q}S\,\bar{c}}{I_{yy}U_0}C_{m_u}$
w	$X_w = \frac{\bar{q}\,S}{mU_0}\left(C_{L_0}\left(1 - \frac{2}{\pi e A}C_{L_\alpha}\right)\right)$	$Z_w = -\frac{\bar{q}\,S}{mU_0}\left(C_{D_0} + C_{L_\alpha}\right)$	$M_w = \frac{\bar{q}S\,\bar{c}}{I_{yy}U_0}C_{m_\alpha}$
$\dot{w}$	*0*	$Z_{\dot{w}} = \frac{\bar{q}\,S\,\bar{c}}{2m{U_0}^2}\left[C_{D_0} + C_{L_{\dot{\alpha}}}\right]$	$M_{\dot{w}} = \frac{\bar{q}S\,\bar{c}^2}{2I_{yy}{U_0}^2}C_{m_{\dot{\alpha}}}$
q	*0*	$Z_q = \frac{\bar{q}\,S\,\bar{c}}{2mU_0}C_{L_q}$	$M_q = \frac{\bar{q}\,S\,\bar{c}^2}{2I_{yy}U_0}C_{m_q}$

- X derivatives:

$$X_u = -\frac{6155.39568 \cdot 26.01}{4990 \cdot 137.2} * (2 \cdot 0.0298 + 0)$$

$$X_u = -0.01393$$

$$X_w = -\frac{6155.39568 \cdot 26.01}{4990 \cdot 137.2} * \left(0.3 \cdot \left(1 - \frac{2}{\pi \cdot 0.83 \cdot 7.56} \cdot 5.48\right)\right)$$

$$X_w = 0.03251$$

- Z derivatives

$$Z_u = -\frac{6155.39568 \cdot 26.01}{4990 \cdot 137.2} \cdot (2 \cdot 0.3 + 0.02)$$

$$Z_u = -0.144988$$

$$Z_w = -\frac{6155.39568 \cdot 26.01}{4990 \cdot 137.2} \cdot (0.0298 + 5.48)$$

$$Z_w = -1.28847$$

$$Z_{\dot{w}} = \frac{6155.39568 \cdot 26.01 \cdot 1.98}{2 \cdot 4990 \cdot 137.2^2} \cdot (0.0298 + 2.5)$$

$$Z_{\dot{w}} = 0.00426$$

$$Z_q = \frac{6155.39568 \cdot 26.01 \cdot 1.98}{2 \cdot 4990 \cdot 137.2} \cdot 8.1$$

$$Z_q = 1.87526$$

- M derivatives

$$M_u = \frac{6155.39568 \cdot 26.01 \cdot 1.98}{27467 \cdot 137.2} \cdot 0$$

$$M_u = 0$$

$$M_w = \frac{6155.39568 \cdot 26.01 \cdot 1.98}{27467 \cdot 137.2} \cdot (-1.89)$$

$$M_w = -0.158985$$

$$M_{\dot{w}} = \frac{6155.39568 \cdot 26.01 \cdot 1.98^2}{2 \cdot 27467 \cdot 137.2^2} \cdot (-9.1)$$

$$M_{\dot{w}} = -0.00552$$

$$M_q = \frac{6155.39568 \cdot 26.01 \cdot 1.98^2}{2 \cdot 27467 \cdot 137.2} \cdot (-34)$$

$$M_q = -2.83145$$

Let’s regroup all numerical applications of those longitudinal derivatives into this table:

	X	Z	M
u	$X_u = -0.01393$	$Z_u = -0.144988$	0
w	$X_w = 0.03251$	$Z_w = -1.28847$	$M_w = -0.158985$
$\dot{w}$	0	$Z_{\dot{w}} = 0.00426$	$M_{\dot{w}} = -0.00552$
q	0	$Z_q = 1.87526$	$M_q = -2.83145$

The aircraft matrix for longitudinal stability is described as:

$$A = \begin{bmatrix} X_u & X_w & 0 & -g\cos\theta_0 \\ Z_u & Z_w & U_0 & -g\sin\theta_0 \\ M_u + Z_u M_{\dot{w}} & M_w + Z_w M_{\dot{w}} & M_q + U_0 M_{\dot{w}} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} -0.01393 & 0.03251 & 0 & -9.81 \\ 0.14498 & -1.28847 & 137.2 & 0 \\ 0.0008 & -0.15186 & -3.58879 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

b. Control matrix

To calculate the control matrix with respect to the longitudinal axis we need to calculate both elevator and throttle derivatives.

- Elevator derivatives:

$$X_{\delta_e} = \frac{6155.39568 \cdot 26.01}{4990 \cdot 137.2} \cdot 0$$
$$X_{\delta_e} = 0$$

$$Z_{\delta_e} = \frac{6155.39568 \cdot 26.01}{4990 \cdot 137.2} \cdot 0.6$$
$$Z_{\delta_e} = 0.14031$$

$$M_{\delta_e} = \frac{6155.39568 \cdot 26.01 \cdot 1.98}{27467 \cdot 137.2} \cdot (-2)$$
$$M_{\delta_e} = -0.16823$$

- Throttle derivatives:

$$X_{\delta_T} = \frac{6155.39568 \cdot 26.01}{4990 \cdot 137.2} \cdot 0$$
$$X_{\delta_T} = 0$$

$$Z_{\delta_T} = \frac{6155.39568 \cdot 26.01}{4990 \cdot 137.2} \cdot 0$$
$$Z_{\delta_T} = 0$$

$$M_{\delta_T} = \frac{6155.39568 \cdot 26.01 \cdot 1.98}{27467 \cdot 137.2} \cdot 0$$
$$M_{\delta_T} = 0$$

Observation: while on cruise and with steady flight conditions, the acceleration of the airplane will be equal to zero. Consequently, the throttle derivatives will be equal to zero as the thrust won't be changing because there is no need to adjust the throttle settings.

The control matrix for longitudinal stability is described as:

$$B = \begin{bmatrix} X_{\delta_e} & X_{\delta_T} \\ Z_{\delta_e} & Z_{\delta_T} \\ M_{\delta_e} + Z_{\delta_e} M_{\dot{w}} & M_{\delta_T} + Z_{\delta_T} M_{\dot{w}} \\ 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0.14031 & 0 \\ -0.16901 & 0 \\ 0 & 0 \end{bmatrix}$$

c. *Calculation using python*

Using python, we can calculate the aircraft and control matrix more efficiently.

To do so, we will create functions that apply the different formulas. We will also use a data file where all derivatives, geometry of the plane and steady flight conditions are stored.

The aircraft matrix, also known as the state-space matrix, gives us a lot of information over the aircraft stability as it describes the behavior of the aircraft like the position, velocity, angular rate etc.

Computing those formulas on python give us this matrix:

```
==================================
Aircraft matrix for longitudinal stability:

[[ -0.0139376    0.03251104   0.          -9.81      ]
 [ -0.14498843  -1.28847948 137.2          0.        ]
 [  0.00080085  -0.15186868  -3.58929078   0.        ]
 [  0.           0.           1.           0.        ]]
==================================
```

As well as for the aircraft matrix, we can compute the control matrix which gives us direct information on the inputs related to the elevator and throttle in the case of longitudinal motion. See the result below:

```
================================
Control matrix (Elevator/Throttle) for longitudinal stability
[[ 0.          0.        ]
 [ 0.14031139  0.        ]
 [-0.16901382  0.        ]
 [ 0.          0.        ]]
================================
```

We can see that the throttle values are at 0. As said in the manual calculation part, with steady flight conditions, no acceleration is needed.

2. The characteristic equation

a. Manual calculation

To get the characteristic equation, let's calculate the following determinant:

$$\boldsymbol{det(\lambda I - A) = \begin{vmatrix} \lambda + 0.01393 & -0.03251 & 0 & 9.81 \\ 0.14498 & \lambda + 1.28847 & -137.2 & 0 \\ -0.0008 & 0.15186 & \lambda + 3.58879 & 0 \\ 0 & 0 & -1 & \lambda \end{vmatrix}}$$

We will take the row with the more zeros which is the 4th one.

We can then make the calculation as follow:

$$\boldsymbol{det(\lambda I - A) = -1 \cdot \begin{vmatrix} \lambda + 0.01393 & -0.03251 & -9.81 \\ 0.14498 & \lambda + 1.28847 & 0 \\ -0.0008 & 0.15186 & 0 \end{vmatrix} + \lambda \cdot \begin{vmatrix} \lambda + 0.01393 & -0.03251 & 0 \\ 0.14498 & \lambda + 1.28847 & -137.2 \\ -0.0008 & -0.15186 & \lambda + 3.58879 \end{vmatrix}}$$

$$\det(\lambda \mathrm{I} - \mathrm{A}) = 9.81 \cdot \begin{vmatrix} 0.14498 & \lambda + 1.28847 \\ -0.0008 & 0.15186 \end{vmatrix}$$

$$+\lambda \cdot \left[137.2 \cdot \begin{vmatrix} \lambda + 0.01393 & 0.03251 \\ -0.0008 & 0.15186 \end{vmatrix} + (\lambda + 3.58879) \cdot \begin{vmatrix} \lambda + 0.01393 & -0.03251 \\ -0.14498 & \lambda + 1.28847 \end{vmatrix}\right]$$

$$\begin{aligned} \det(\lambda \mathrm{I} - \mathrm{A}) = 9.81 \cdot (0.0220 + 0.0008\lambda + 0.00103) + \lambda \\ \cdot [137.2 \cdot (0.15186\lambda + 0.00211 + 0.000026) + (\lambda + 3.58879) \\ \cdot (\lambda^2 + 1.3024\lambda + 0.01794 - 0.00471)\,] \end{aligned}$$

$$\begin{aligned} det(\lambda I - A) = 9.81 \cdot (0.0008\lambda + 0.02303) + \lambda \\ \cdot [137.2 \cdot (0.15186\lambda + 0.002136) + (\lambda + 3.58879) \cdot (\lambda^2 + 1.3024\lambda + 0.01323)\,] \end{aligned}$$

$$det(\lambda I - A) = 0.0078\lambda + 0.2259 + \lambda \cdot [(20.83519\lambda + 0.29305) + (\lambda^3 + 1.3024\lambda^2 + 0.01323\lambda + 3.58879\lambda^2 + 4.674\lambda + 0.04747)]$$

$$det(\lambda I - A) = 0.0078\lambda - 0.22059 + \lambda \cdot (\lambda^3 + 4.89119\lambda^2 + 25.52242\lambda + 0.34052)$$

$$det(\lambda I - A) = \lambda^4 + 4.89119\lambda^3 + 25.52242\lambda^2 + 0.34832\lambda + 0.22059$$

b. Calculation using python

Using python, we will obtain the following result (I used **sympy** for symbolic calculation so that the “s” appears):

```
Characteristic equation:
1.0 * s^ 4  + 4.891707857578721 * s^ 3  + 25.533809056898715 * s^ 2  + 0.3760697964645441 * s^ 1  + 0.22613112879192954
```

This confirms our previous calculation. However, using python is much more efficient as the library NumPy use powerful algorithms to calculate the characteristic equation.

3. Eigenvalues of the system

To find the eigen values of the system we must resolve the characteristic equation:

$$\boldsymbol{det(\lambda I - A) = 0}$$

As the polynom is of degree 4 it is hard to calculate by hand, that’s why we will use programming to do the task.

Thanks to NumPy we are able to calculate effectively the roots of the system that are 4 complex.

```
Eigenvalues = [-2.43931965+4.41711337j -2.43931965-4.41711337j -0.00653428+0.09401444j
 -0.00653428-0.09401444j]
```

4. Different modes of longitudinal stability

For both short period mode and phugoid mode we will find the natural frequency and the damping factor necessary to draw the curves.

```
=================================
Natural frequencies:
[5.04590635 0.09424124]
Damping ratios:
[0.48342547 0.06933562]
=================================
```

The python program will return 2 lists. Each list contains both modes natural frequencies respectively for the short period and the phugoid mode.

a. Short period mode

Natural frequency for short period is:

$$\omega_{sp} = 5.045$$

Damping ratio for short period is:

$$\zeta_{sp} = 0.483$$

b. Phugoid mode

Natural frequency for short period is:

$$N_p = 0.094$$

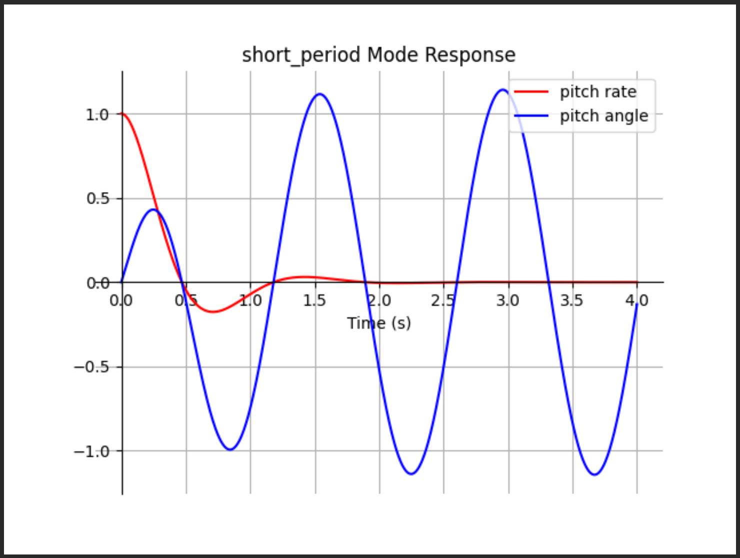
Damping ratio for short period is:

$$N_p = 0.069$$

5. Curves of longitudinal motion

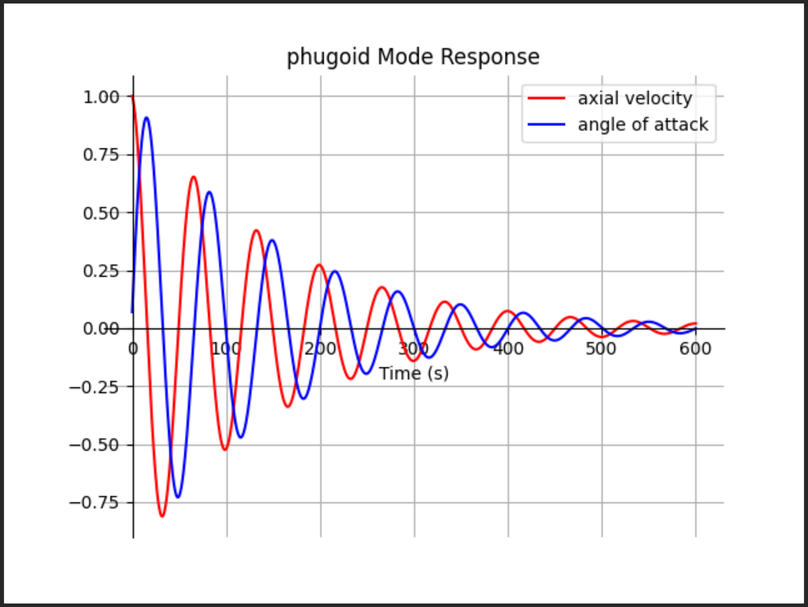
a. Short period mode curves

By computing the formula for pitch rate and pitch angle we obtain the short period mode:



b. Phugoid mode curves

By computing the formula for axial velocity, and angle of attack we obtain the phugoid mode:



6. Transfer Functions

Using the library sympy we can calculate the transfer functions.

As learnt in during class, to find the transfer functions of the system we need to resolve the following system:

$$\frac{X(s)}{\eta(s)} = (sI - A)^{-1} \cdot B$$

We can do that this way with sympy:

```
s = sympy.symbols('s')
tfs = (s * sympy.eye(4) - self.aircraft_matrix).inv() @ self.control_matrix
```

a. Python calculation of elevator and throttle tfs

We obtain the following transfer functions for the elevator:

```
Transfer functions for Elevator:

0-> u(s)/delta_e(s):
 9.58937816716585e-5*s^4 + 0.019475533685236*s^3 + 0.0745083649230235*s^2 + 0.0646525517515994*s + 0.00111781254822341
---------------------------------------------------------------------------------------------------------------------------------------------------------
0.0210216460879131*s^6 + 0.130210702144366*s^5 + 0.671169129469272*s^4 + 0.709325912849518*s^3 + 0.0272195465881832*s^2 + 0.00637046921256218*s + 0.000107774876909573


1-> w(s)/delta_e(s):
 0.020343528472174*s^5 - 3.26229465429033*s^4 - 4.32875734918422*s^3 - 0.168963296327355*s^2 - 0.046226167820995*s - 0.000786597147037475
---------------------------------------------------------------------------------------------------------------------------------------------------
0.144988434324649*s^6 + 0.8980764758988*s^5 + 4.62912185096386*s^4 + 4.89229307257183*s^3 + 0.187735985390634*s^2 + 0.0439377750524418*s + 0.000743334303950413


2-> q(s)/delta_e(s):
 s*(-0.0245050495064712*s^2 - 0.0350053397799749*s - 0.000598110450438957)
-----------------------------------------------------------------------------------------------------------
0.144988434324649*s^4 + 0.709241063443924*s^3 + 3.7021069975043*s^2 + 0.0545257709861833*s + 0.0327863983156074


3-> theta(s)/delta_e(s):
 -0.0245050495064712*s^2 - 0.0350053397799749*s - 0.000598110450438957
-----------------------------------------------------------------------------------------------------------
0.144988434324649*s^4 + 0.709241063443924*s^3 + 3.7021069975043*s^2 + 0.0545257709861833*s + 0.0327863983156074
```

Due to how sympy does the symbolic calculation simplification we got transfer functions with power degree of 6. However, if we had to factorize those transfer functions, we could make them at the power degree of 4.

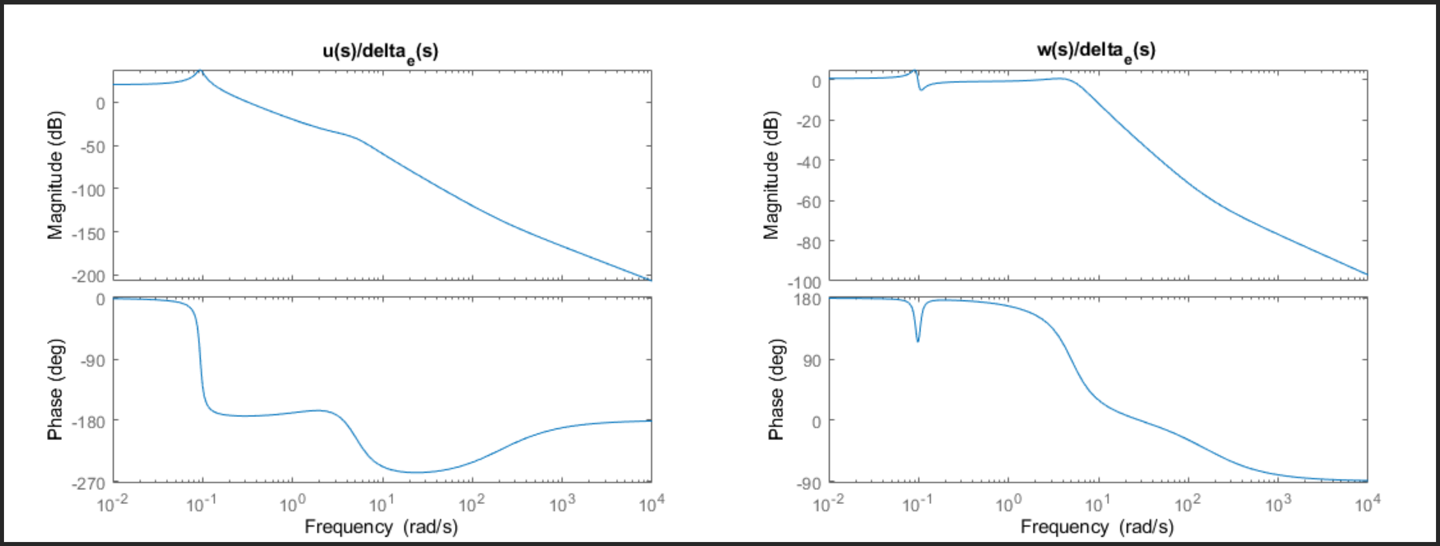
It is important to note that in this report we will make all the necessary verification (using matlab and bode diagram representation) to make sure that the given transfer functions are correct.

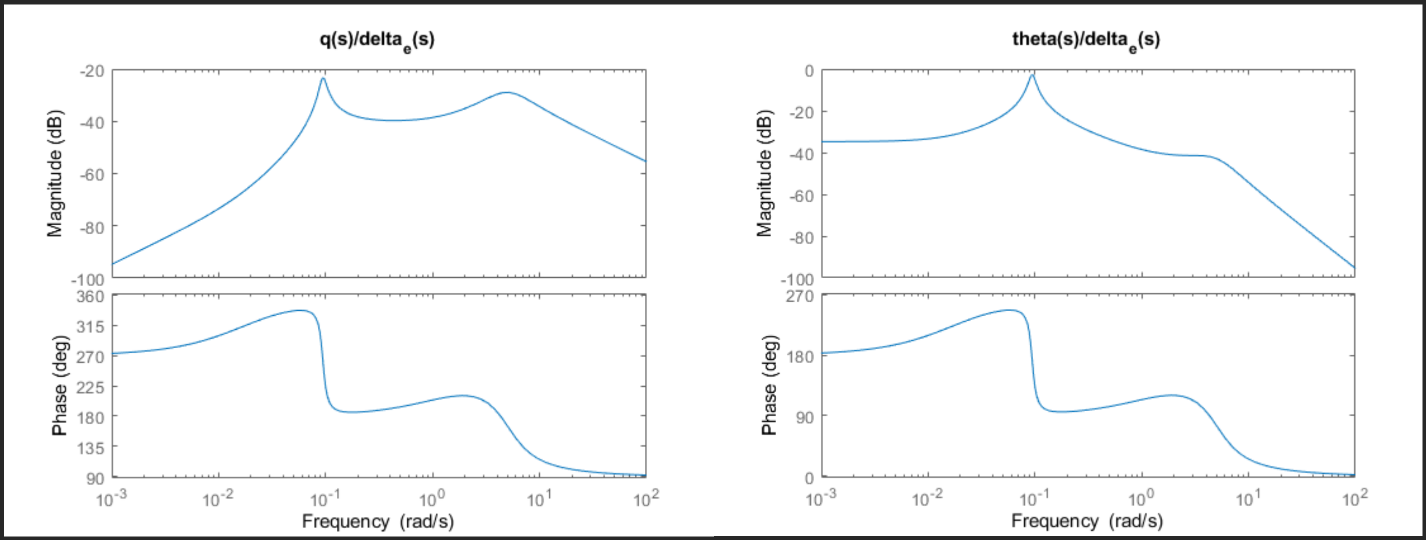
For the throttle, as expected we obtain 0 everywhere:

```
Transfer functions for Throttle:

0-> u(s)/delta_t(s):
 0


1-> w(s)/delta_t(s):
 0


2-> q(s)/delta_t(s):
 0


3-> theta(s)/delta_t(s):
 0
```

b. Matlab calculation for bode diagram

As it is simpler to use matlab for plotting bode diagram with the built-in function “bode(tf)” I will be using matlab.





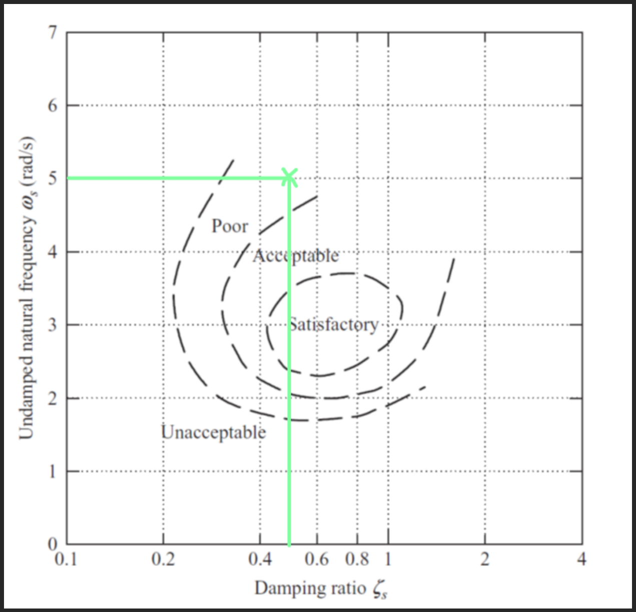
Those bode diagrams were made using matlab "bode" function.

7. Analyze: flying qualities requirements

First, we need to check if the short periode mode damping is acceptable.

Given that: $\omega_{sp} = 5.045 \; and \;\; \zeta_{sp} = 0.483$

We have on the graph:



We can see that we are in a relatively poor area which means that the aircraft might have a slightly slow response to pitch disturbance.

we can further analyze the short period mode by calculating the time constant (τ_{sp}) and the period (T_{sp}) of the mode.

The time constant (τ_{sp}) is given by:

$\tau_{sp} = 1 / (\omega_{sp} * \zeta_{sp})$

So, $\tau_{sp} = 0.410$ seconds

The period (T_{sp}) is given by:

$T_{sp} = 2\pi / \omega_{sp}$

$T_{sp} = 1.245$ seconds

Now, we can compare the time constant and period of the short period mode to the expected values for a stable aircraft. Typically, the time constant for the short period mode should be between 0.3 and 0.5 seconds, and the period should be between 1.1 and 1.6 seconds.

In this case, the calculated values of τ_{sp} and T_{sp} fall within the expected range, indicating that the short period mode is stable. However, as mentioned earlier, the damping ratio of 0.483 is relatively low, indicating a slow response to pitch disturbance.

Finally, the phugoid mode gives us some indication on the aircraft stability. The natural frequency of 0.094 indicates that the phugoid mode oscillation is relatively slow. Also, the damping ratio of 0.069 indicates that the oscillations will persist for a relatively long time which may suggest that control inputs need to be adjusted to maintain stability on the longitudinal axis.

Please go to the next page for PART II.

PART II – Lateral Dynamic Stability of an Airplane

1. Aircraft Matrix (A) and Control Matrix (B)

Remainder:

$$\bar{q} = 6155.39568$$

a. aircraft matrix

As we did for longitudinal stability, we will calculate the aircraft matrix for lateral dynamic stability using the following formulas:

Equation	*Y*	*L*	*N*
v	$Y_v = \frac{\bar{q}\,S}{mU_0} C_{Y_\beta}$	$L_v = \frac{\bar{q}\,S\,b}{I_{xx}U_0} C_{l_\beta}$	$N_v = \frac{\bar{q}\,S\,b}{I_{zz}U_0} C_{n_\beta}$
p	$Y_p = \frac{\bar{q}\,Sb}{2mU_0} C_{Y_p}$	$L_p = \frac{\bar{q}\,S\,b^2}{2I_{xx}U_0} C_{l_p}$	$N_p = \frac{\bar{q}\,S\,b^2}{2I_{zz}U_0} C_{n_p}$
r	$Y_r = \frac{\bar{q}\,Sb}{2mU_0} C_{Y_r}$	$L_r = \frac{\bar{q}\,S\,b^2}{2I_{xx}U_0} C_{l_r}$	$N_r = \frac{\bar{q}\,S\,b^2}{2I_{zz}U_0} C_{n_r}$

- Y derivatives:

$$Y_v = \frac{6155.39568 \cdot 26.01}{4990 \cdot 137.2} \cdot (-0.59)$$

$$Y_v = -0.1379$$

$$Y_p = \frac{6155.39568 \cdot 26.01 \cdot 14.02}{2 \cdot 4990 \cdot 137.2} \cdot (-0.19)$$

$$Y_p = -0.3114$$

$$Y_r = \frac{6155.39568 \cdot 26.01 \cdot 14.02}{2 \cdot 4990 \cdot 137.2} \cdot 0.39$$
$$Y_r = 0.6393$$

- L derivatives:

$$L_v = \frac{6155.39568 \cdot 26.01 \cdot 14.02}{20602 \cdot 137.2} \cdot (-0.13)$$
$$L_v = -0.1032$$

$$L_p = \frac{6155.39568 \cdot 26.01 \cdot 14.02^2}{2 \cdot 20602 \cdot 137.2} \cdot (-0.5)$$
$$L_p = -2.7833$$

$$L_r = \frac{6155.39568 \cdot 26.01 \cdot 14.02^2}{2 \cdot 20602 \cdot 137.2} \cdot (0.14)$$
$$L_p = 0.7793$$

- N derivatives

$$N_v = \frac{6155.39568 \cdot 26.01 \cdot 14.02}{46308 \cdot 137.2} \cdot 0.08$$
$$N_v = 0.0282$$

$$N_p = \frac{6155.39568 \cdot 26.01 \cdot 14.02^2}{2 \cdot 46308 \cdot 137.2} \cdot 0.019$$
$$N_p = 0.0470$$

$$N_r = \frac{6155.39568 \cdot 26.01 \cdot 14.02^2}{2 \cdot 46308 \cdot 137.2} \cdot (-0.197)$$
$$N_r = -0.4878$$

Now we can calculate the aircraft matrix as follow:

$$A = \begin{bmatrix} Y_v & Y_p & -(U_0 - Y_r) & g\cos\theta_0 \\ L_v & L_p & L_r & 0 \\ N_v & N_p & N_r & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} -0.1379 & -0.3114 & -136.5607 & 9.81 \\ -0.1032 & -2.7833 & 0.7793 & 0 \\ 0.0282 & 0.0470 & -0.4878 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

b. *Control matrix*

To calculate the control matrix, we need to calculate both rudder and aileron derivatives:

- Rudder derivatives

$$Y_{\delta_r} = \frac{6155.39568 \cdot 26.01}{4990 \cdot 137.2} \cdot 0.144$$

$$Y_{\delta_r} = 0.0336$$

$$L_{\delta_r} = \frac{6155.39568 \cdot 26.01 \cdot 14.02}{20602 \cdot 137.2} \cdot 0.0106$$

$$L_{\delta_r} = 0.0084$$

$$N_{\delta_r} = \frac{6155.39568 \cdot 26.01 \cdot 14.02}{46308 \cdot 137.2} \cdot (-0.0758)$$

$$N_{\delta_r} = -0.02677$$

❖ Aileron derivatives

$$Y_{\delta_a} = \frac{6155.39568 \cdot 26.01}{4990 \cdot 137.2} \cdot 0$$

$$Y_{\delta_a} = 0$$

$$L_{\delta_a} = \frac{6155.39568 \cdot 26.01 \cdot 14.02}{20602 \cdot 137.2} \cdot 0.156$$

$$L_{\delta_a} = 0.1238$$

$$N_{\delta_a} = \frac{6155.39568 \cdot 26.01 \cdot 14.02}{46308 \cdot 137.2} \cdot (-0.0012)$$

$$N_{\delta_a} = -0.0004$$

c. *Using python*

Using python code, we obtain:

```
================================
Aircraft matrix for longitudinal stability:

[[  -0.13797286   -0.3114679  -136.56067116    9.81      ]
 [  -0.10323434   -2.78335672    0.77933988    0.        ]
 [   0.02826339    0.04705501   -0.4878861     0.        ]
 [   0.            1.            0.            0.        ]]
================================
Control matrix (Rudder/Throttle) for lateral stability
[[ 0.03367473  0.        ]
 [ 0.00841757  0.12388121]
 [-0.02677956 -0.00042395]
 [ 0.          0.        ]]
================================
```

2. The characteristic equation

a. *Manual calculation*

Now that we have the aircraft matrix, we can calculate the characteristic equation:

$$\boldsymbol{det(\lambda I - A)} = \begin{vmatrix} \boldsymbol{\lambda + 0.1379} & \boldsymbol{0.3114} & \boldsymbol{136.5606} & \boldsymbol{-9.81} \\ \boldsymbol{0.1032} & \boldsymbol{\lambda + 2.7833} & \boldsymbol{-0.7793} & \boldsymbol{0} \\ \boldsymbol{-0.0282} & \boldsymbol{0.047} & \boldsymbol{\lambda + 0.4878} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{-1} & \boldsymbol{0} & \boldsymbol{\lambda} \end{vmatrix}$$

$$\boldsymbol{det(\lambda I - A)} = \boldsymbol{-1} \cdot \begin{vmatrix} \boldsymbol{\lambda + 0.1379} & \boldsymbol{136.5606} & \boldsymbol{-9.81} \\ \boldsymbol{0.1032} & \boldsymbol{-0.7793} & \boldsymbol{0} \\ \boldsymbol{-0.0282} & \boldsymbol{\lambda + 0.4878} & \boldsymbol{0} \end{vmatrix} + \boldsymbol{\lambda} \cdot \begin{vmatrix} \boldsymbol{\lambda + 0.1379} & \boldsymbol{0.3114} & \boldsymbol{136.5606} \\ \boldsymbol{0.1032} & \boldsymbol{\lambda + 2.7833} & \boldsymbol{-0.7793} \\ \boldsymbol{-0.0282} & \boldsymbol{0.047} & \boldsymbol{\lambda + 0.4878} \end{vmatrix}$$

$$\begin{aligned} \det(\lambda I - A) = 9.81 \cdot \begin{vmatrix} 0.1032 & -0.7793 \\ -0.0282 & \lambda + 0.4878 \end{vmatrix} + \lambda \\ \cdot \left[-0.0282 \cdot \begin{vmatrix} 0.3114 & 136.5606 \\ \lambda + 2.7833 & -0.7793 \end{vmatrix} + 0.047 \cdot \begin{vmatrix} \lambda + 0.1379 & 136.5606 \\ 0.1032 & -0.7793 \end{vmatrix} + (\lambda + 0.4878) \right. \\ \left. \cdot \begin{vmatrix} \lambda + 0.1379 & 0.3114 \\ 0.1032 & \lambda + 2.7833 \end{vmatrix}\right] \end{aligned}$$

$$\begin{aligned} \det(\lambda I - A) = 9.81 \cdot (0.1032\lambda + 0.0503 - 0.02197) + \lambda \\ \cdot [-0.0282 \cdot (-0.2426 - 136.5606\lambda - 380.089) + 0.047 \cdot (-0.7793\lambda - 0.1074 - 14.09) \\ + (\lambda + 0.4878) \cdot (\lambda^2 + 2.9212\lambda + 0.3838 - 0.0321)] \end{aligned}$$

$$\begin{aligned} \det(\lambda I - A) = 1.01\lambda + 0.277 + \lambda \cdot [-0.0282 \cdot (-380.3316 - 136.5606\lambda) + 0.047 \cdot (-0.7793\lambda - 14.1974) + (\lambda^3 \\ + 2.9212\lambda^2 + 0.3517\lambda + 0.4878\lambda^2 + 1.4249\lambda + 0.1715] \end{aligned}$$

$$\det(\lambda I - A) = 1.01\lambda + 0.277 + \lambda \cdot (\lambda^3 + 3.409\lambda^2 + 5.5909\lambda + 10.2295)$$

$$\det(\lambda I - A) = \lambda^4 + 3.40\lambda^3 + 5.59\lambda^2 + 11.23\lambda + 0.277$$

b. Calculation using python

By computing the aircraft matrix we get the following characteristic equation:

```
Lateral Characteristic equation:
1.0 * s^ 4  + 3.4092156862391976 * s^ 3  + 5.60014476763606 * s^ 2  + 11.265664290382915 * s^ 1  + 0.2780136108582089
--------------------------------
```

3. Eigenvalues of the system

To find the eigen values of the system we must resolve the characteristic equation:

$$\boldsymbol{det(\lambda I - A) = 0}$$

As the polynom is of degree 4 it is hard to calculate by hand, that's why we will use programming to do the task.

a. Calculation using python

Thanks to NumPy we can effectively calculate the roots of the system that are 4 complexes.

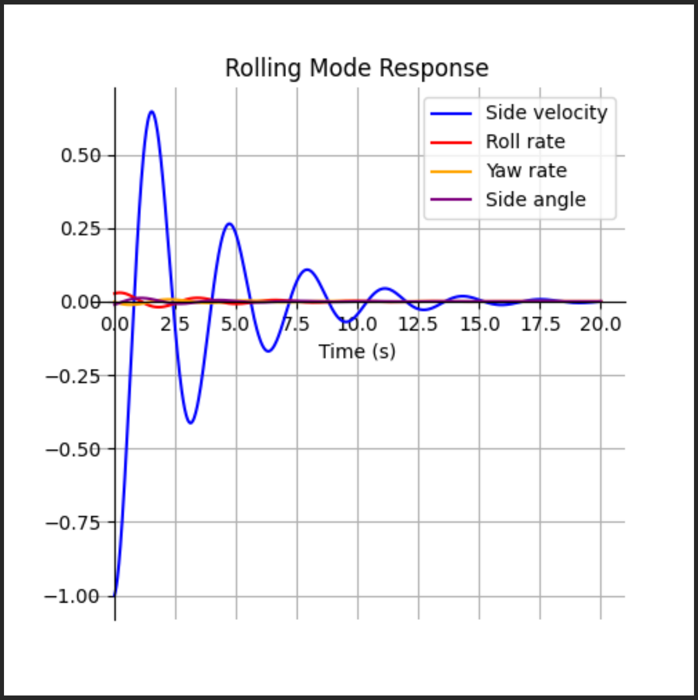
```
================================
Eigen values:
[-2.82590244+0.j         -0.27916485+1.96465681j -0.27916485-1.96465681j
 -0.02498355+0.j        ]
```

4. Different modes of longitudinal stability

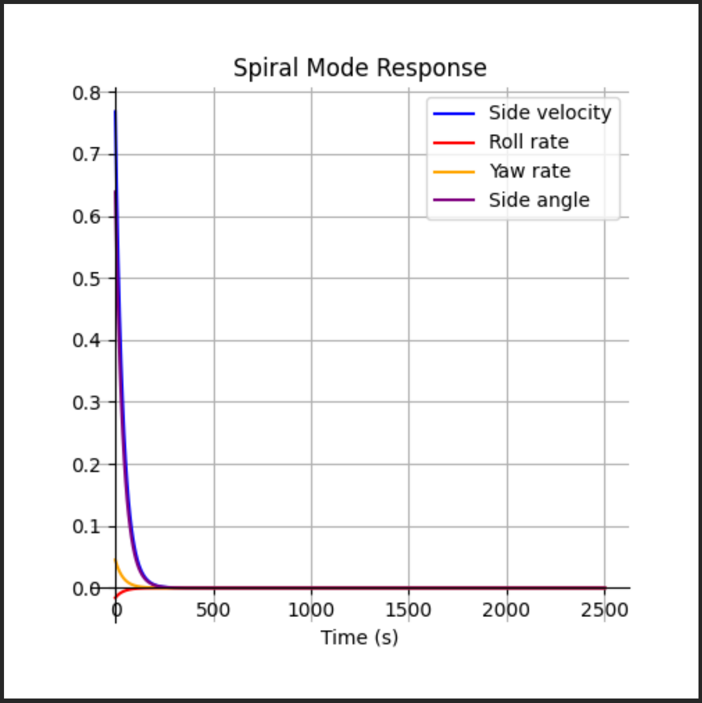
```
================================

Rolling mode parameter:
Lambda_roll (-0.2791648466500354+1.9646568071744046j)


Spiral mode parameter:
Lambda_spiral (-0.02498355321492144+0j)


Dutch roll mode parameters:
(Natural frequency) wn_dutch_roll =  1.9843914385982075
(Damping factor) zeta_dutch_roll =  0.1406803321260245
```

5. Curves of lateral motion

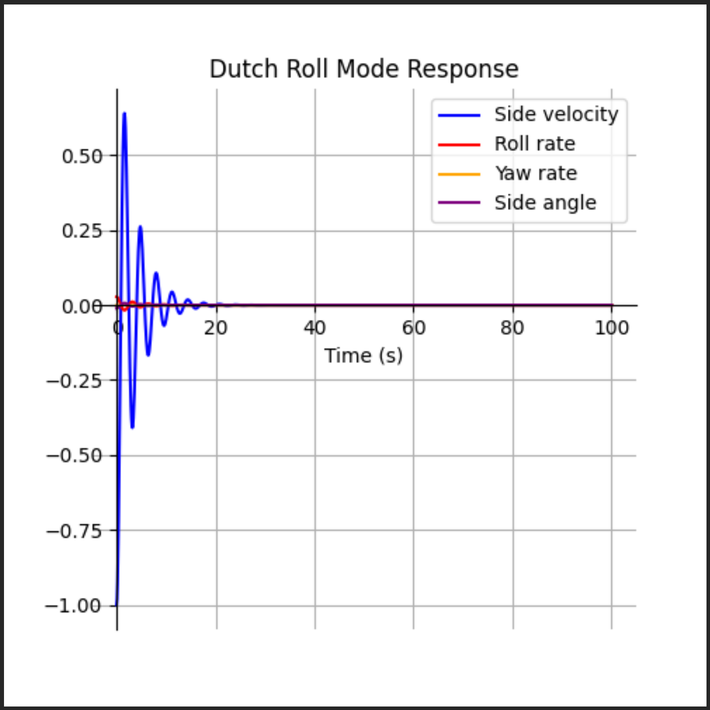
Rolling mode:



Spiral mode:



Dutch roll mode:



We will explain why there is those variation during the analysis part.

6. Transfer Functions

Using the library sympy we can calculate the transfer functions.

As for the longitudinal part, we will resolve the following system using python:

$$\frac{X(s)}{\eta(s)} = (sI - A)^{-1} \cdot B$$

Please go to the next page to see the computing of transfer functions.

a. Python calculation of aileron and rudder tfs

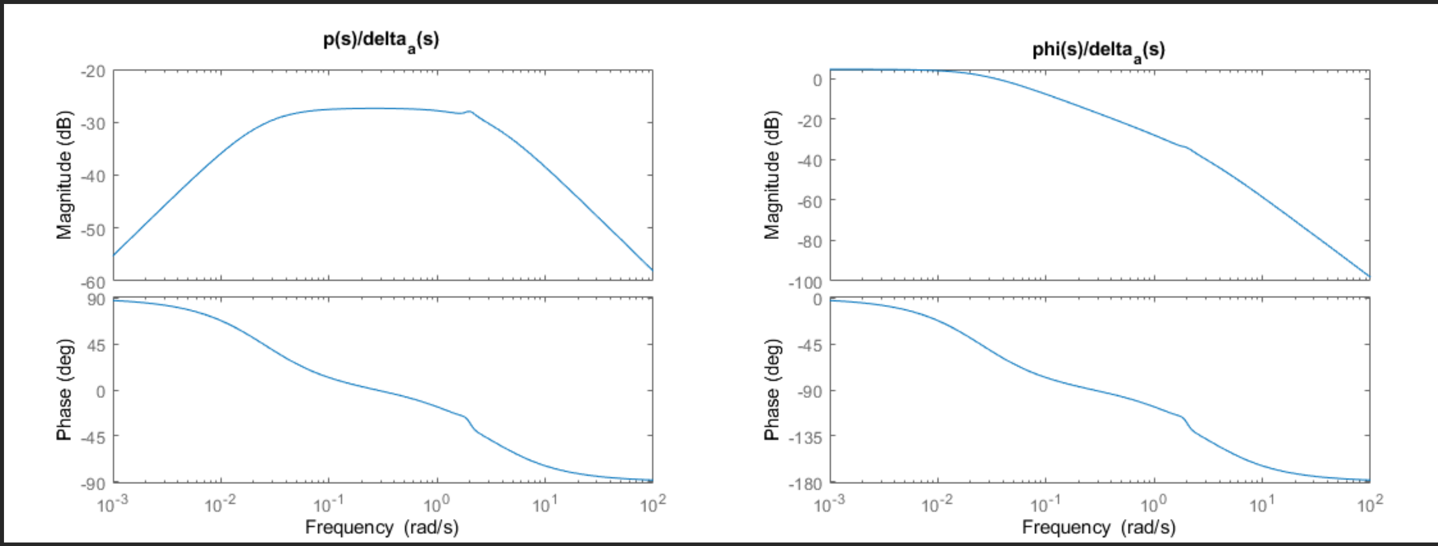
```
================================
Transfer functions for Aileron:

0->u(s)/delta_a(s):
 -8.59598586206201e-19*s^4 + 2.12448915381096e-5*s^3 + 0.000623762198611258*s^2 + 0.000818394914359862*s + 0.000178097456988085
---------------------------------------------------------------------------------------------------------------------------------------------
0.00110020242532102*s^5 + 0.0040528541444754*s^4 + 0.00719096728483494*s^3 + 0.0140859048558168*s^2 + 0.00370840353695944*s + 8.39675551367069e-5


1->w(s)/delta_a(s):
 s*(0.0127887955558147*s^2 + 0.00796987360136672*s + 0.0495996580483583)
--------------------------------------------------------------------------------------------------------------
0.103234343267372*s^4 + 0.351948142425727*s^3 + 0.578127267289119*s^2 + 1.16300345448837*s + 0.0287005525363379


2->q(s)/delta_a(s):
 -4.518183340934e-6*s^4 + 4.76846056985344e-5*s^3 + 8.79016020816073e-6*s^2 + 0.000360205939956122*s + 9.92332007454649e-5
------------------------------------------------------------------------------------------------------------------------------------------
0.0106573296298456*s^5 + 0.0392587778078725*s^4 + 0.0696567349318112*s^3 + 0.136445919158268*s^2 + 0.0359221884848422*s + 0.000813368424490626


3->theta(s)/delta_a(s):
 0.0127887955558147*s^2 + 0.00796987360136672*s + 0.0495996580483583
--------------------------------------------------------------------------------------------------------------
0.103234343267372*s^4 + 0.351948142425727*s^3 + 0.578127267289119*s^2 + 1.16300345448837*s + 0.0287005525363379
```

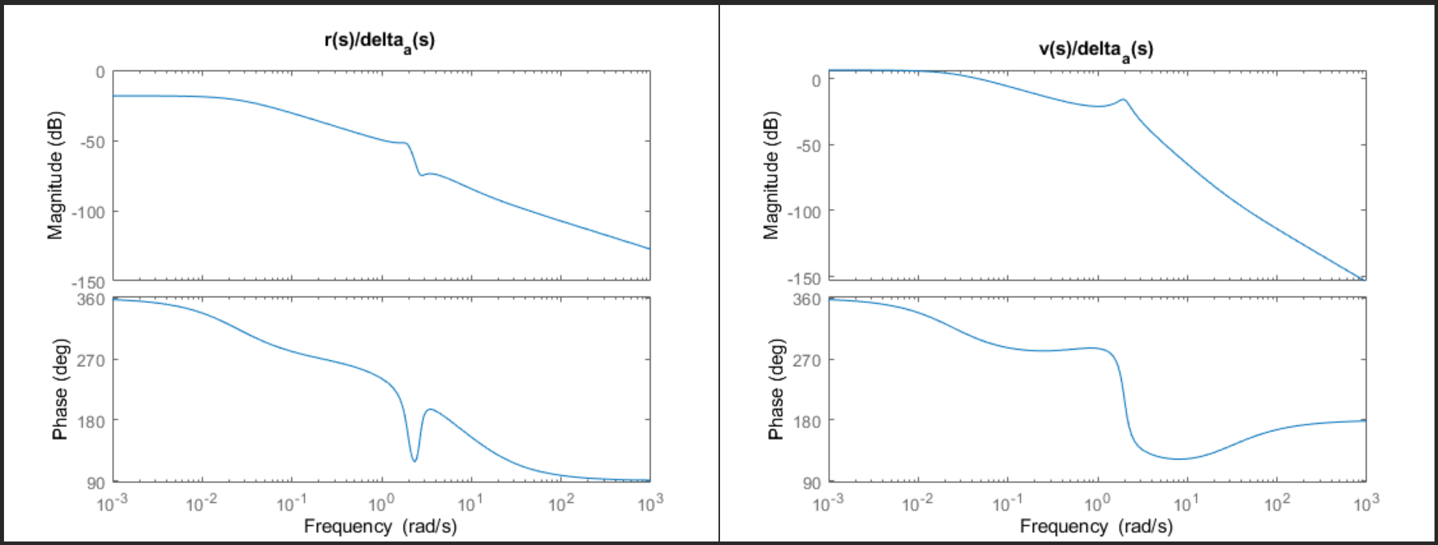
```
Transfer functions for Rudder:

0->u(s)/delta_r(s):
 3.70490230654783e-5*s^4 + 0.00415196069747085*s^3 + 0.012421813828965*s^2 + 0.00291696974371307*s - 4.96684695679571e-5
---------------------------------------------------------------------------------------------------------------------------------------------
0.00110020242532102*s^5 + 0.0040528541444754*s^4 + 0.00719096728483494*s^3 + 0.0140859048558168*s^2 + 0.00370840353695944*s + 8.39675551367069e-5


1->w(s)/delta_r(s):
 s*(0.000868982262125873*s^2 - 0.00196956213740166*s - 0.0359575323006617)
--------------------------------------------------------------------------------------------------------------
0.103234343267372*s^4 + 0.351948142425727*s^3 + 0.578127267289119*s^2 + 1.16300345448837*s + 0.0287005525363379


2->q(s)/delta_r(s):
 -0.000285398581035664*s^4 - 0.000897726240497923*s^3 - 0.000299077873002638*s^2 - 0.000284511954720931*s - 7.25165697755321e-5
------------------------------------------------------------------------------------------------------------------------------------------
0.0106573296298456*s^5 + 0.0392587778078725*s^4 + 0.0696567349318112*s^3 + 0.136445919158268*s^2 + 0.0359221884848422*s + 0.000813368424490626


3->theta(s)/delta_r(s):
 0.000868982262125873*s^2 - 0.00196956213740166*s - 0.0359575323006617
--------------------------------------------------------------------------------------------------------------
0.103234343267372*s^4 + 0.351948142425727*s^3 + 0.578127267289119*s^2 + 1.16300345448837*s + 0.0287005525363379
```

As said for the longitudinal part, due to how the sympy algorithm for symbolic calculation works, we get transfer functions with higher power than 4. But expressions can be factorize to get a max degree of 4.

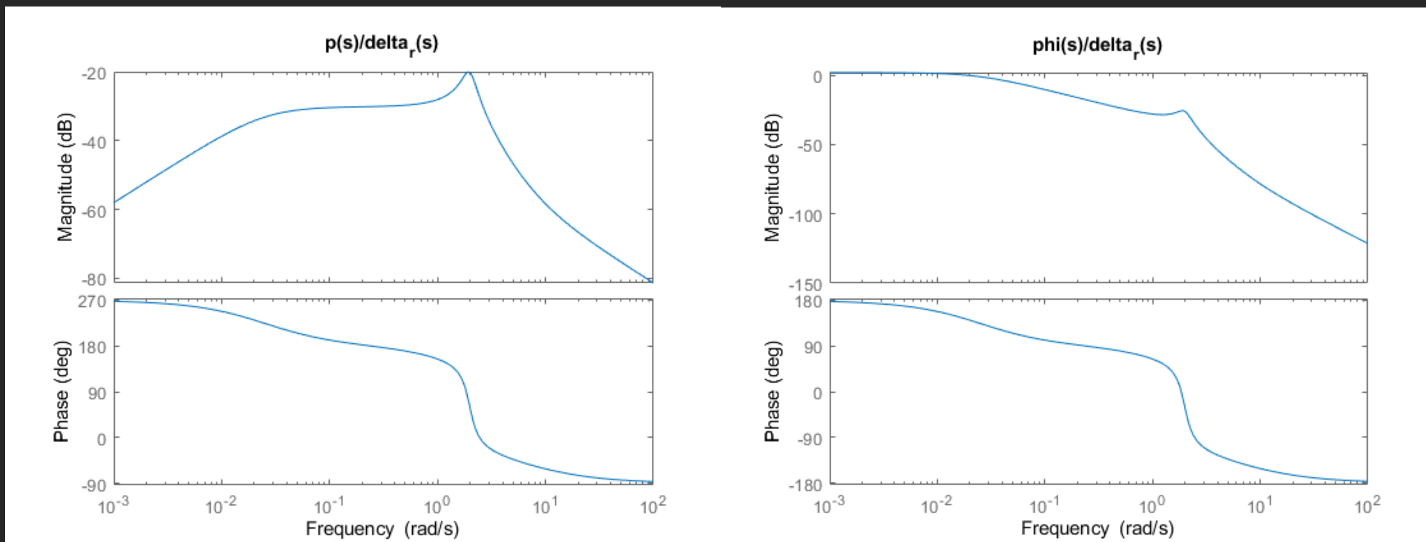
b. *Matlab calculation for bode diagram*

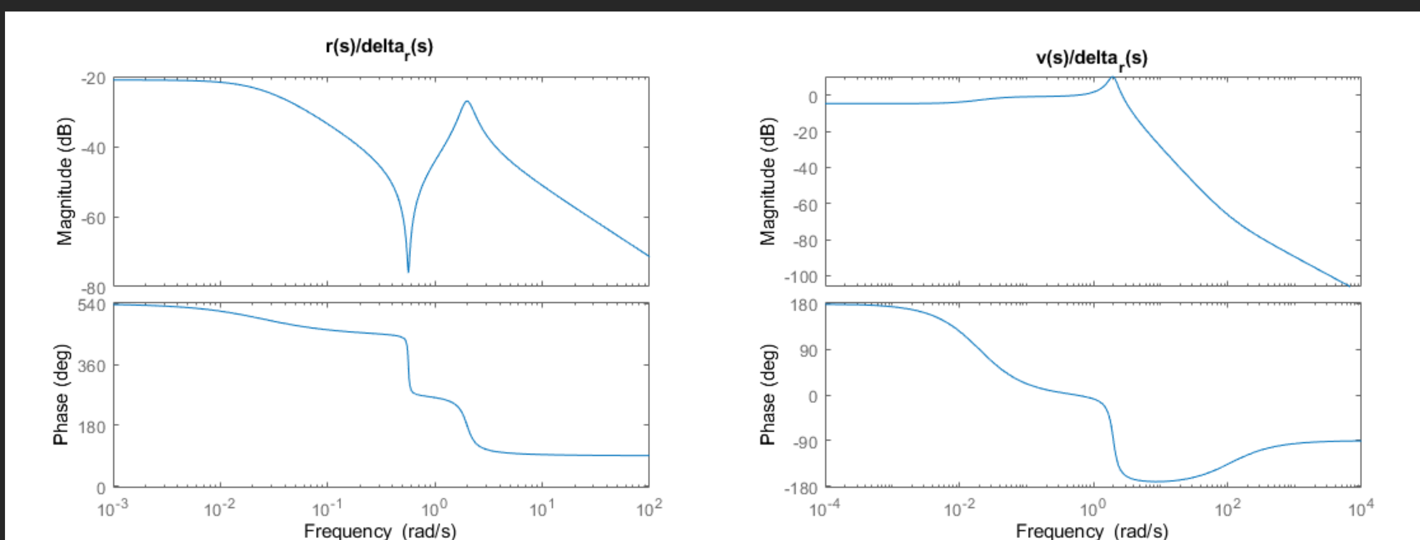
- Aileron bodes





❖ Rudder bodes





Please go to the next page to see the analytic part.

7. Analyze: flying qualities requirements

From the eigenvalues, we can see that there are two complex conjugate poles and two real poles. The real poles are negative and therefore stable, while the complex conjugate poles have a negative real part, which means that they are also stable.

Overall, the lateral stability of the aircraft seems to be satisfactory. The real poles are negative, and the complex conjugate poles have a negative real part, indicating stability. The rolling mode and spiral mode parameters are also within acceptable ranges. The natural frequency of the Dutch roll mode is also within an acceptable range, but the damping factor is somewhat low, indicating that the aircraft may have a slow response to lateral disturbances. Once again, it might be interesting to play with the control inputs for both rudder and aileron to adjust the stability of the aircraft.

Conclusion

After analyzing the longitudinal and lateral stability of the aircraft, it appears that there may be some issues with the aircraft's response to disturbances. In the longitudinal axis, while the short period mode is stable, the damping ratio is relatively low, which indicates that the aircraft may have a slow response to pitch disturbance. Similarly, in the lateral axis, while the Dutch roll mode natural frequency is within an acceptable range, the damping factor is somewhat low, which suggests that the aircraft may have a slow response to lateral disturbances. These factors indicate that the aircraft may not be responsive enough to disturbances, which could cause instability.

To address this, it may be necessary to adjust the control inputs for both rudder and aileron to improve the aircraft's stability. Additionally, it may be worth considering implementing other measures such as learnt during our aeronautic class:

For example, a stability augmentation system to obtain the desired performance by adjusting the gains.

ANNEXE

Flight conditions		
Altitude H	6096	m
Air density ρ	0,654	kg/m3
Speed U0	137,2	m/s
Mach number M	0,4	
Center of gravity GC		
Initial attitude θ0	0	rad

Steady state conditions	
CL0	0,3
CD0	0,0298
CTx0	0,0298
Cm0	0
Cmt0	0

Geometric data		
wing area S	26,01	m2
wing span b	14,02	m
wing mean chord $\bar{c}$	1,98	m
Aspect ratio A	7,56	
Oswald Number e	0,86	

Longitudinal Aerodynamic deravatives	
Cmu	0
Cm alpha	-1,89
Cm alpha dot	-9,1
Cmq	-34
Cm t u	0
Cm t alpha	0
C L u	0,02
C L alpha	5,48
C L alpha dot	2,5
CLq	8,1

Inertial data		
Mass m	4990	Kg
Ixx	20602	kg.m2
Iyy	27467	kg.m2
Izz	46308	kg.m2
Ixz	5929	kg.m2

C D u	0
C D alpha	0,131
C D alpha dot	0
C Dq	0
C T u	-0,0596
C L delta e	0,6
C D delta e	0
C m delta e	-2

Lateral Aerodynamic derivatives	
C lβ	-0,13
C lp	-0,5
C lr	0,14
C l delta a	0,156
C l delta r	0,0106
C nβ	0,08
C np	0,019
C nr	-0,197
C n delta a	-0,0012
C n delta r	-0,0758
C yβ	-0,59
C yp	-0,19
C yr	0,39
C y delta a	0
C y delta r	0,144