

**Université de Strasbourg
Télécom Physique**

**ROBUST FLIGHT CONTROL
PROJECT REPORT**

NKIMO LENOU Randolph

February 26, 2024

Contents

1	Question 1.1: Flight dynamics	5
2	Question 1.2: Model construction & analysis	6
2.1	Model construction	6
2.2	Analysis of the open loop	7
3	Question 2.1: Damping gain design	8
3.1	Output transfert function and analysis	8
4	Question 2.2: Scaling gain design	8
5	Question 2.3: Integral gain design	10
6	Question 3A.1: Weighting filter discussion	12
7	Question 3A.2: Weighting filter computation	13
8	Question 3B.1: Reference model computation	14
9	Question 3C.1: Controller design	15
9.1	First exercise: weighted closed loop	15
9.2	Second exercise: weighted closed loop	16
9.3	Third exercise : H_∞ synthesis	17
9.4	Fourth exercise: Optimize the constraints	18
10	Question 3C.2: Controller order reduction	19
11	Question 3C.3: Controller analysis & simulation	24
11.1	First exercise	24
11.2	Second exercise	25
12	Question 3D.1: Controller design	27
13	Question 3D.2: Controller analysis & simulation (5%)	29
14	Question 3E.1: Controller design (5%)	31
15	Question 3E.2: Controller order reduction (5%)	32
16	Question 3E.3: Controller analysis & simulation (5%)	34

List of Figures

1	State space of actuator + missile model	6
2	Representation of zeros and poles of G_{am}	7
3	Poles and zeros of $G_{am}(1, 1)$	8
4	Poles and zeros of $G_{clq-unsc}$	8
5	Step Response of G	9
6	Loop gain, root locus and step response of G_{ol-nz} with 60° phase margin	10
7	Loop gain, root locus and step response of G_{ol-nz} with smallest settling time	11
8	inverse of performance weight (red) and inverse of control weight (blue)	13
9	Step Response of T_d	15
10	$\sigma(T_{wz})$ et $\sigma(T_{wzi})$, $i = 1, 2, 3$	18
11	$\sigma(T_{wz})$ et $\sigma(T_{wzi})$, $i = 1, 2, 3$	19
12	bode plot of C_{e-min} & $C0_e$	21
13	bode plot of C_{i-min} & C_{i-red}	22
14	poles-zeros of C_{i-min} & C_{i-red}	23
15	2x3 plots	24
16	Open loop margin	25
17	2x2 times plots	26
18	$\sigma(T_{wz})$ et $\sigma(T_{wzi})$, $i = 1, 2, 3$, (<i>hfinstruct</i>)	28
19	bode plot of C_{i-min} , C_{i-red} & C_{i-red}^* (<i>hfinstruct</i>)	28
20	2x3 plots (<i>hfinstruct</i>)	29
21	Open loop margin (<i>hfinstruct</i>)	30
22	2x2 times plots (<i>hfinstruct</i>)	30
23	F_{f-init} sigma value plot	31
24	zeros and poles of F_{f-init}	32
25	singular values of F_{f-lf} & F_{f-init}	33

26	Comparaison of pole zero map	34
27	Comparison of the singular values - Model Matching Error . .	35
28	Comparison of the singular values - Controller	36
29	Comparison of the step responses - Closed Loop	37
30	Comparison of the step responses - Actuator	38
31	2 x 2 Time Frequency plots	39

1 Question 1.1: Flight dynamics

The procedure of computing the equilibrium points or trajectories of a non-linear system is the first step towards computing some kind of linearized model. For a aerospace flight control we often want to calculate the equilibrium point (or trim) around an **imposed** condition $\rho(t) = \bar{\rho}$. At this condition we impose by definition that: $\dot{\bar{x}} \stackrel{\text{def}}{=} 0$.

For a missile, as in our case, the equilibrium points are determined following the steps below:

1. Step 1: Solve for the control from the output equation as a function of the equilibrium point vector (known) and AoA (unknown)

$$\delta_m = \delta_m(\rho_y, \alpha)$$

2. Step 2: Replace the control inside the moment equation and solve for the AoA and for the control

$$\begin{aligned} C_m[\alpha, M, \delta_m(\rho_y, \alpha)] &= 0 \iff C_m[\alpha, \rho_y] = 0 \\ \implies \alpha &= f_\alpha(\rho_y), \quad \delta_m = \delta_m[(\rho_y), f_\alpha(\rho_y)] \end{aligned}$$

3. Step 3: Replace the AoA and the control inside the force equation to compute the pitch rate at equilibrium

$$q = f_q(\rho_y)$$

The angle of attack α and the Mach number M (or airspeed V) have a certain **operating envelope** $\Gamma_{\sigma x}$ which is given by:

$$-20^\circ \leq \alpha \leq 20^\circ, \quad 1.5 \leq M \leq 3$$

Once the operating point has been determined, we can proceed to linearize the system around this point using the tangent linearization method. This method involves approximating the non-linear state matrices around the equilibrium point by evaluating their Jacobians at this point. In other words, we aim to characterize the local behavior of the system by focusing on changes around the equilibrium point, thereby simplifying the analysis and deriving linear models to study the behavior of the system in its immediate vicinity.

This can be achieved by using the tool model linearizer of **MATLAB/SIMULINK** to find those operating point and also use the same tool to compute the linear model around the operating points.

2 Question 1.2: Model construction & analysis

2.1 Model construction

```
>> G_am

G_am =

A =
      x1          x2          x3          x4
x1   -1.305       1   -0.114       0
x2   -300.4       0   -131.4       0
x3     0       0       0       1
x4     0       0 -2.25e+04    -210

B =
      u_cmd
x1     0
x2     0
x3     0
x4 2.25e+04

C =
      x1          x2          x3          x4
y1 -146.3       0   -11.73       0
y2     0       1       0       0

D =
      u_cmd
y1     0
y2     0

Continuous-time state-space model.
Model Properties
```

Figure 1: State space of actuator + missile model

Transfert function from input u_{cmd} to output y_1 :

$$G_{am(1,1)} = \frac{-2.6397 \times 10^5 (s+36.52)(s-36.64)}{(s^2 + 1.305s + 300.4)(s^2 + 210s + 2.25 \times 10^4)}$$

Transfert function from input u_{cmd} to output y_2 :

$$G_{am(2,1)} = \frac{-2.9564 \times 10^6 (s+1.044)}{(s^2 + 1.305s + 300.4)(s^2 + 210s + 2.25 \times 10^4)}$$

2.2 Analysis of the open loop

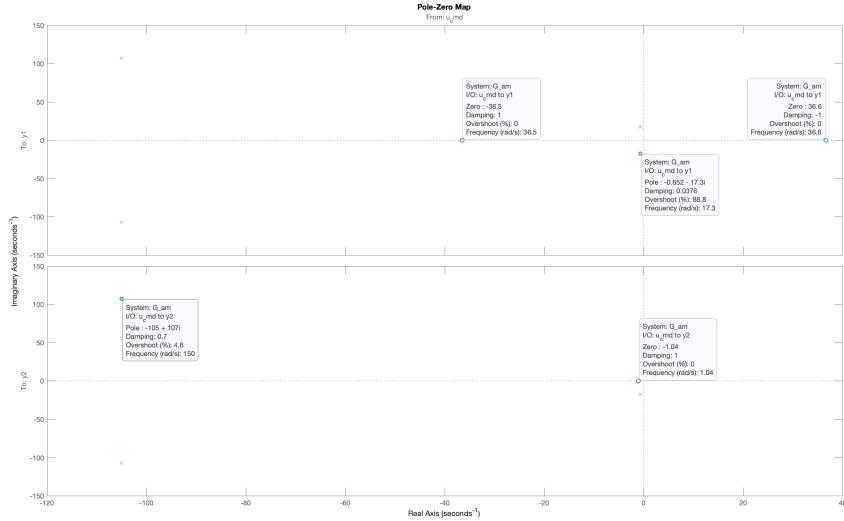


Figure 2: Representation of zeros and poles of G_{am}

The system is stable because all poles are located at strictly negative real parts. Furthermore, the frequency separation between the actuator and missile poles is due to the independence of the two systems. Although the output of the actuator is the missile control, the dynamics imposed by their respective poles ensures good frequency response separation.

The system has positive real part **zeros**, indicating that it is **non-minimum phase**.

3 Question 2.1: Damping gain design

3.1 Output transfert function and analysis

The value of control gain C_q which gives a closed-loop damping of about 0.7
 $C_q = -0.15769$.

Transfert function from input u_{unsc} to output y_1 :

$$G_{clq-unsc} = \frac{-2.6397 \times 10^5 (s+36.64)(s+36.52)}{(s^2+27.41s+411.2)(s^2+183.9s+1.762 \times 10^4)}$$

Transfert function from input u_{cmd} to output y_1 :

$$G_{am(1,1)} = \frac{-2.6397 \times 10^5 (s+36.52)(s+36.64)}{(s^2+1.305s+300.4)(s^2+210s+2.25 \times 10^4)}$$

The damping gain is calculated as $C_q = -0.15769$, and the missile's dominant poles are shifted from $p_{1,2} = -0.652 \pm 17.3j$ to $p_{1,2} = -17.7 \pm 14.9j$, representing a significant increase in both system damping and bandwidth. The actuator poles are also shifted to a lower frequency, which has a somewhat limiting effect on the closed-loop system (from a natural frequency of 150, rad/s to 133, rad/s).

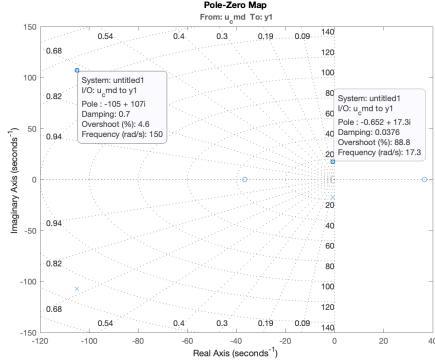


Figure 3: Poles and zeros of $G_{am}(1,1)$

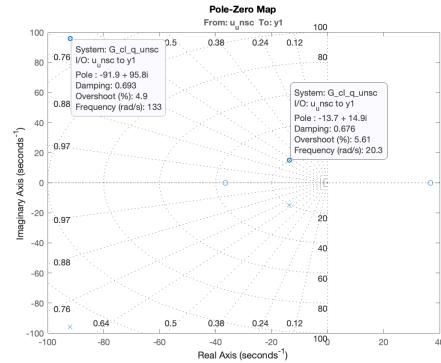


Figure 4: Poles and zeros of $G_{clq-unsc}$

4 Question 2.2: Scaling gain design

The value of scaling gain C_{sc} is 0.020514

The 5% settling time: 0.256 secondes.

The transfert fonction of total pre-compensated inner loop G:

From input u_p to output $y1$:

$$G(s) = \frac{-5415(s - 36.64)(s + 36.52)}{(s^2 + 27.41s + 411.2)(s^2 + 183.9s + 1.762e04)}$$

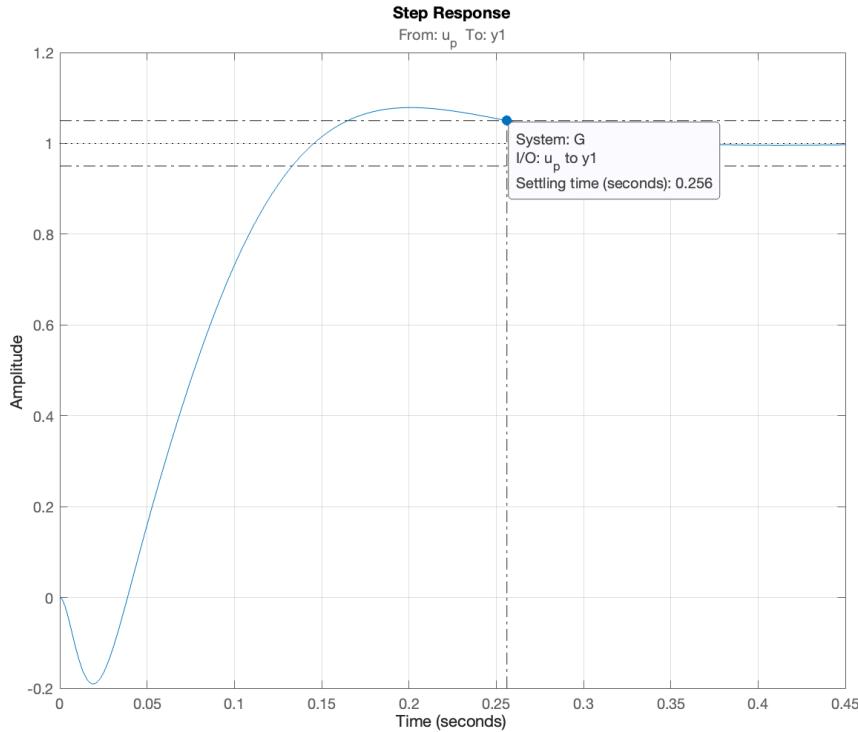


Figure 5: Step Response of G

The disturbance on $y1$ is not rejected, as the transfer function from input u_p to output $y1$ has no integral term (i.e. a pole at 0). Consequently, any disturbance at the output will be directly reflected at the $y1$ output.

To remedy this problem, it is necessary to add an **integrator** into the control law upstream of the disturbance. The integrator will generate a corrective action that compensates for the disturbance and thus stabilizes the $y1$ output. By introducing this integrator, constant errors will be corrected, improving system performance in terms of disturbance rejection and reference tracking.

5 Question 2.3: Integral gain design

Here us the output of the transfer function $zpk(G_{ol-nz})$ when the integral gain C_i is unitary.

From input $e1_f$ to output $y1$:

$$G_{ol-nz} = \frac{-5415(s+36.52)(s-36.64)}{s(s^2+27.41s+411.2)(s^2+183.9s+1.762 \times 10^4)}$$

However we want to adjust the integral gain C_i to guarantee a 60° phase margin or to obtain the smallest 5% settling time. To do so, we will use the Matlab's tool sisotool. First of all we'll study the gain for which we guarantee 60° phase margin.

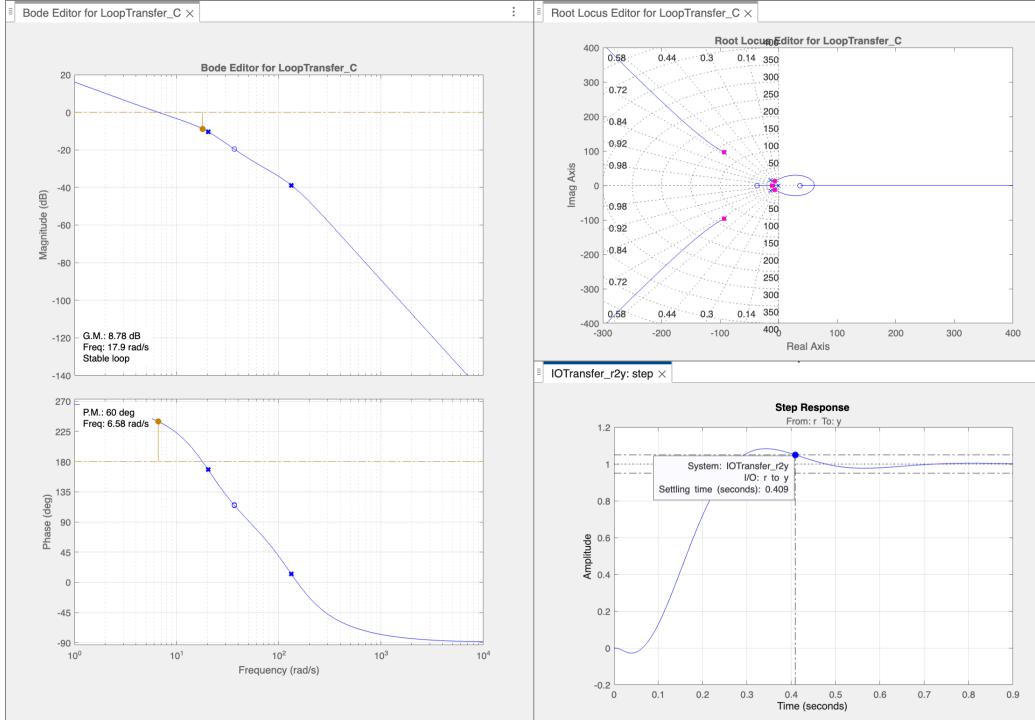


Figure 6: Loop gain, root locus and step response of G_{ol-nz} with 60° phase margin

With the help of the tool we can obtain the following integral gain $C_i = 6.351$.

In this second case the parameters have been adjusted to obtain the smallest

5% settling time.

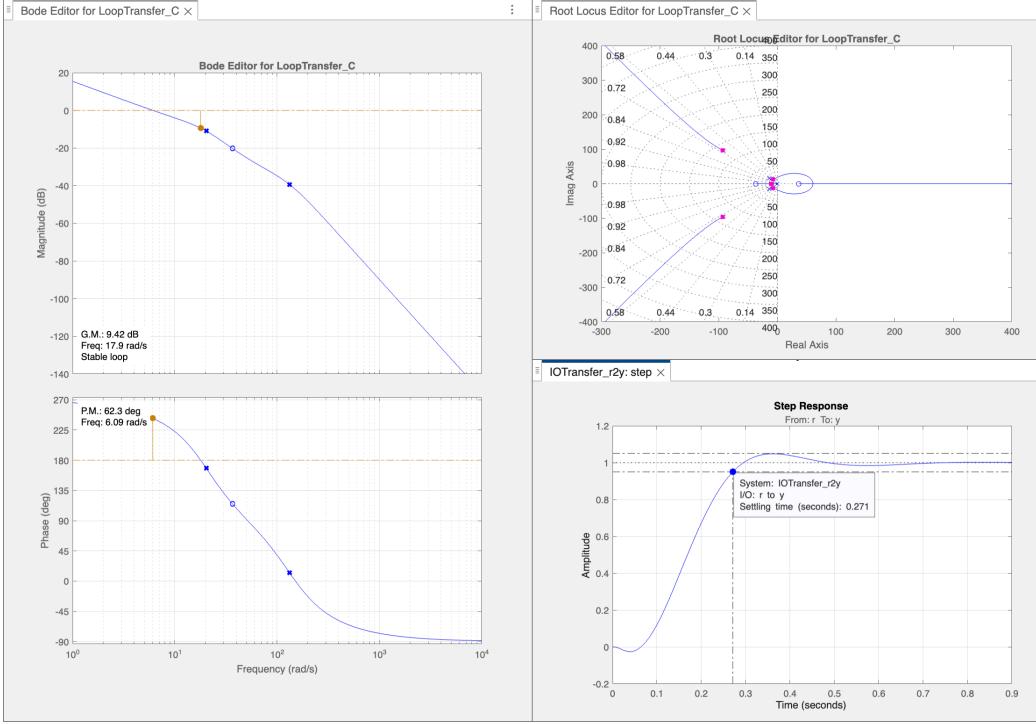


Figure 7: Loop gain, root locus and step response of G_{ol-nz} with smallest settling time

With the help of the tool we can obtain the following integral gain $C_i = 5.9041$. With this gain the closed loop zpk(T) is:

$$T(s) = \frac{-31971(s+36.52)(s-36.64)}{(s+10.65)(s^2+14.56s+223)(s^2+186.1s+1.8 \times 10^4)}$$

In this section we have developed a model capable of rejecting disturbances. However, we can't consider it optimal in any way. Question 3B.1 shows that it is possible to obtain a 5% settling time inferior to $t_{5\%} < 0.18$ s. This means that it is possible to obtain a more optimal model with a different type of feedback. We will treat in the next sections how to optimize the model even further.

6 Question 3A.1: Weighting filter discussion

To characterize the filters, we'll give the characteristics of their inverse.

- W_1^{-1} is a **high pass** filter
 - Low frequency: we have a gain of -40 dB
 - High frequency: the ideal gain is less than 2 ($M_s < 2$).
- W_2^{-1} is a **low pass** filter
 - Low frequency: we have a gain of +40 dB
 - High frequency: M_s

The tool used to design the filters is Matlab's `makeweight` command, which is used to build the template for these filters by specifying the low-, medium- and high-frequency characteristics.

7 Question 3A.2: Weighting filter computation

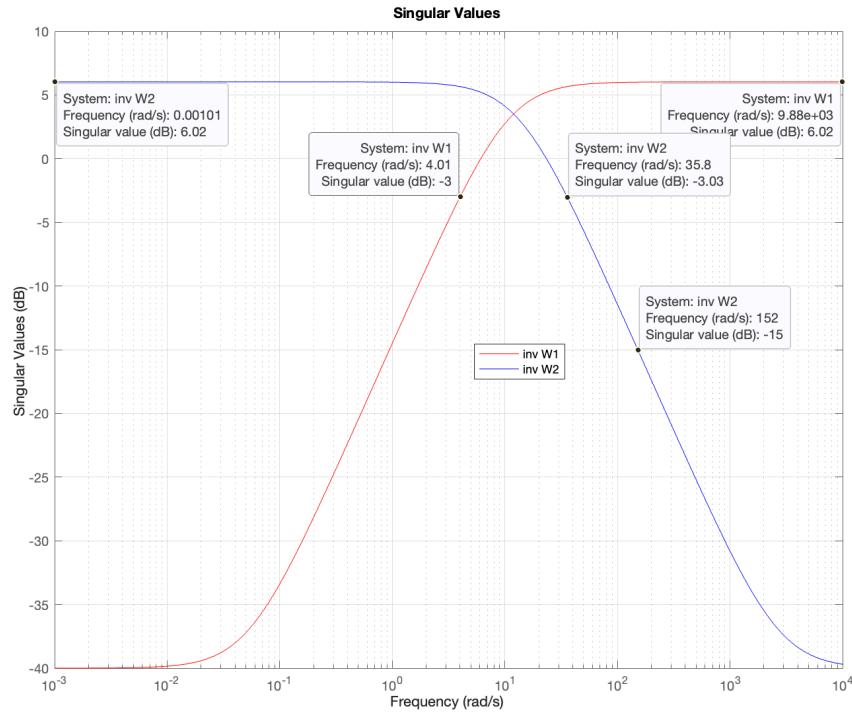


Figure 8: inverse of performance weight (red) and inverse of control weight (blue)

The parameters A_i , M_i , and ω_i .

i	1	2
M_i	2	0.01
$\omega_i(\text{rad/s})$	5.2918	27.00567
A_i	0.01	2

A_i , M_i , and w_i represent respectively the gain in low frequency (DC gain), the gain in high frequency (HF gain), and the bandwidth. More precisely, w_i is not exactly the bandwidth but an approximation as we can see on Figure 8. w_i is the frequency where the asymptote of $|W^{-1}|$ crosses 1 (0 dB). It's therefore a good approximation of the bandwidth. Note that w_1 is approximately the desired closed-loop bandwidth and w_2 is approximately the desired controller bandwidth.

8 Question 3B.1: Reference model computation

The ω_d and ζ_d parameters of the reference model were determined using an optimization with Matlab's `fmincon` function.

The transfert function of reference model is: $T_d(s) = \frac{-9.0819(s-36.64)}{(s^2+25.52s+332.8)}$ with $\omega_d = 18.2417 \text{ rad/s}$ and $\zeta_d = 0.6996$.

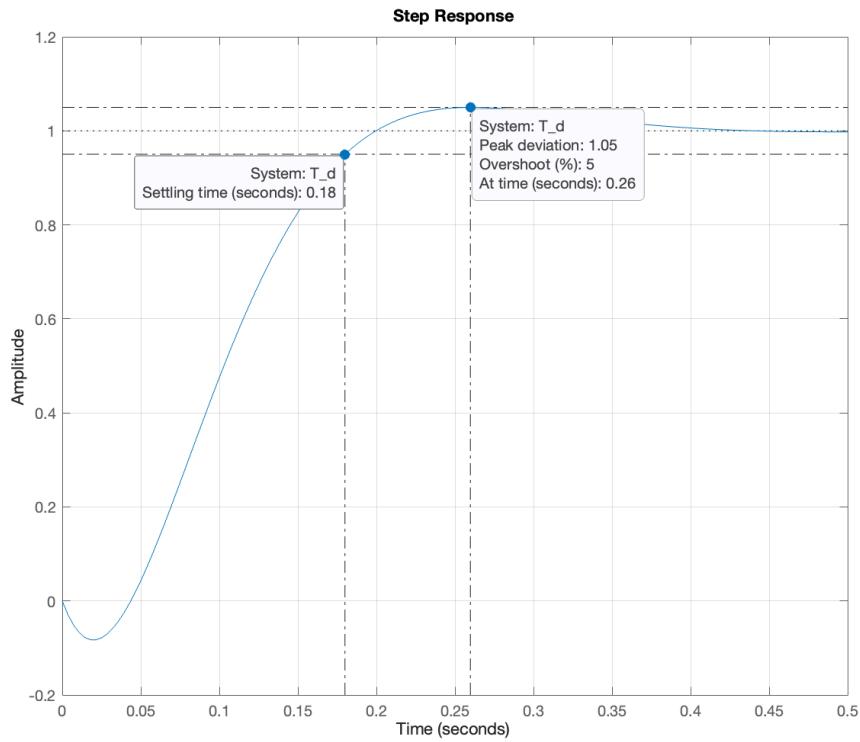


Figure 9: Step Response of T_d

9 Question 3C.1: Controller design

9.1 First exercise: weighted closed loop

Proof. First we proof that $T_{wz} = \begin{bmatrix} W_1 S_o \\ W_2 C_e S_o \\ W_3 (T_d - T_o) \end{bmatrix}$.

We have $z_1 = W_1 e_1$ and $e_1 = w - y_1 = w - G C_e v = w - G C_e e_1$, from these equation we have

$$z_1 = W_1 \frac{1}{(1+C_e G)} w = W_1 S_o w$$

We have $z_2 = W_2 u$ and $u = C_e e_1 = C_e S_o w$, then

$$z_2 = W_2 C_e S_o w$$

We have $z_3 = W_3 e_{1d}$ and $e_{1d} = y_{1d} - y_1 = T_d w - T_o w$, then

$$z_3 = W_3 (T_d - T_o) w$$

note that T_o is the closed-loop transfert function between w and y_1 . finally

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} W_1 S_o \\ W_2 C_e S_o \\ W_3 (T_d - T_o) \end{bmatrix} w$$

Proof. In second place we proof that $P(s) = \begin{bmatrix} W_1 & -W_1 G \\ 0 & W_2 \\ W_3 T_d & -W_3 G \\ 1 & -G \end{bmatrix}$

We have

$$z_1 = W_1 e_1 = W_1 (w - y_1) = W_1 w - W_1 G u$$

$$z_2 = W_2 u$$

$$z_3 = W_3 e_{1d} = W_3 (y_{1d} - y_1) = W_3 T_d w - W_3 G u$$

$$v = e_1 = w - y_1 = w - G u$$

finally, by grouping the previous relations together

$$\begin{bmatrix} z \\ v \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} W_1 & -W_1 G \\ 0 & W_2 \\ W_3 T_d & -W_3 G \\ 1 & -G \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix}$$

9.2 Second exercise: weighted closed loop

Transfert function $P(s)$ for $W_3 = W_1$

From input w to output :

$$\begin{aligned} z1 &: \frac{0.5(s + 10.58)}{(s + 0.05292)} \\ z2 &: 0 \\ z3 &: \frac{-4.5409(s - 36.64)(s + 10.58)}{(s + 0.05292)(s^2 + 25.52s + 332.8)} \\ v &: 1 \end{aligned}$$

From input u to output :

$$\begin{aligned} z1 &: \frac{2707.5(s - 36.64)(s + 36.52)(s + 10.58)}{(s + 0.05292)(s^2 + 27.41s + 411.2)(s^2 + 183.9s + 1.762e04)} \\ z2 &: \frac{100(s + 13.5)}{(s + 2701)} \\ z3 &: \frac{2707.5(s - 36.64)(s + 36.52)(s + 10.58)}{(s + 0.05292)(s^2 + 27.41s + 411.2)(s^2 + 183.9s + 1.762e04)} \\ v &: \frac{5415(s - 36.64)(s + 36.52)}{(s^2 + 27.41s + 411.2)(s^2 + 183.9s + 1.762e04)} \end{aligned}$$

9.3 Third exercise : H_∞ synthesis

The value of γ is 0.9688.

γ_1	0.8323
γ_2	0.8388
γ_3	0.5880

The previous results show that the smallest value is γ_3 corresponding to the less violated constraint which is the model matching constraint.

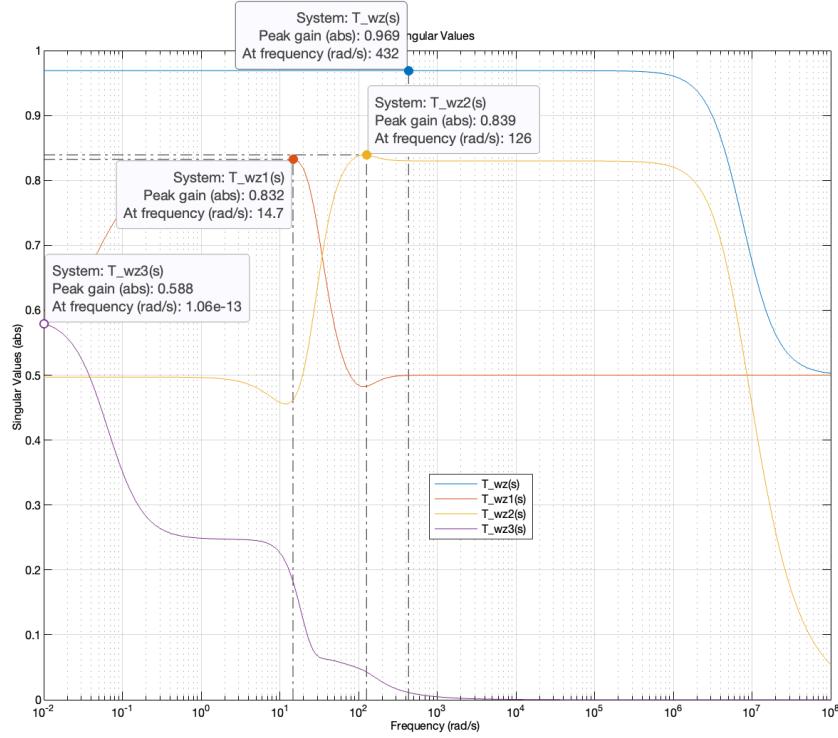


Figure 10: $\sigma(T_{wz})$ et $\sigma(T_{wzi})$, $i = 1, 2, 3$

9.4 Fourth exercise: Optimize the constraints

After tightening the constraints on the reference tracking model, we obtain a $\gamma = 1.1135$.

$W_1(s) = \frac{0.5(s+10.58)}{(s+0.05292)}$	$\gamma_1 = 0.8928$
$W_2(s) = \frac{100(s+13.5)}{(s+2701)}$	$\gamma_2 = 1.0037$
$W_3(s) = \frac{0.5(s+80.3)}{(s+0.4015)}$	$\gamma_3 = 0.7561$

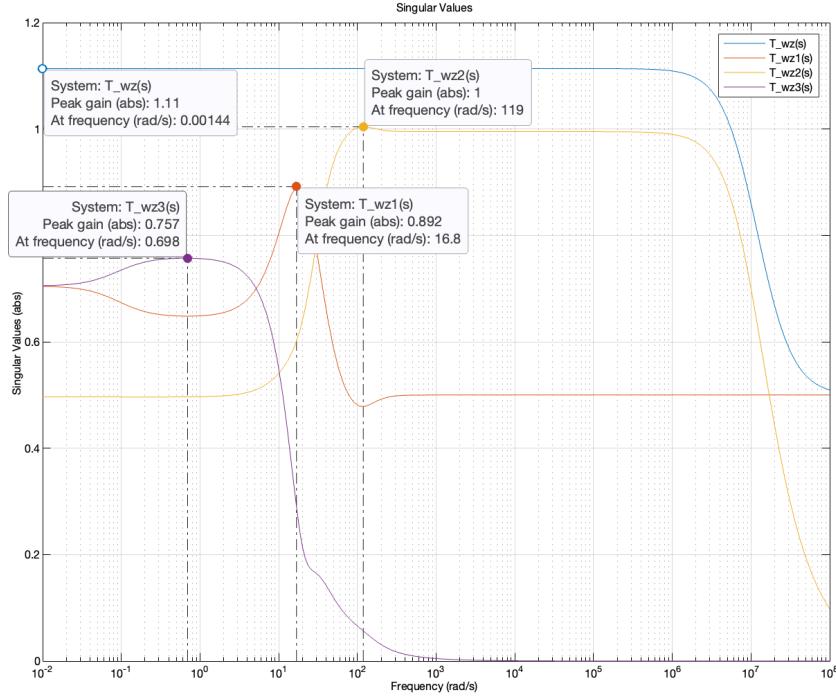


Figure 11: $\sigma(T_{wz})$ et $\sigma(T_{wzi})$, $i = 1, 2, 3$

10 Question 3C.2: Controller order reduction

The controller $C0_e$ designed with the function `hinfsyn` is of the same order as the augmented system \mathbf{P} , therefore an order of **9**.

$$C0_e(s) = \frac{98589(s+2701)(s+0.09117)(s^2+31.47s+406)(s^2+27.41s+411.2)(s^2+183.9s+1.762e04)}{(s+9.908e06)(s+0.09941)(s+0.05292)(s^2+30.38s+376.5)(s^2+71.86s+1697)(s^2+165.2s+1.522e04)}$$

The controller's **poles** :

$$1.0e + 06 \times$$

$$(-9.908332502201731 + 0.0000000000000000i)$$

$$(-0.000082575186173 + 0.000091654990977i)$$

$$(-0.000082575186173 - 0.000091654990977i)$$

$$(-0.000035931407524 + 0.000020151980770i)$$

$$(-0.000035931407524 - 0.000020151980770i)$$

$$(-0.000015189625906 + 0.000012074909223i)$$

$$(-0.000015189625906 - 0.000012074909223i)$$

$$(-0.000000099410692 + 0.000000000000000i)$$

$$(-0.000000052918230 + 0.000000000000000i)$$

In blue we have the **very low frequency** poles and in red the **very high frequency** pole.

The controllers' **zeros**

$$1.0e + 03 \times$$

$$(-2.700567081759474 + 0.000000000000000i)$$

$$(-0.091947990152624 + 0.095751980828497i)$$

$$(-0.091947990152624 - 0.095751980828497i)$$

$$(-0.015736214157757 + 0.012585612684609i)$$

$$(-0.015736214157757 - 0.012585612684609i)$$

$$(-0.013704263145001 + 0.014945505014132i)$$

$$(-0.013704263145001 - 0.014945505014132i)$$

$$(-0.000091172514078 + 0.000000000000000i)$$

In blue we have the **very low frequency** zero and in red the **very high frequency** zero.

$$C_{e-min}(s) = \frac{24.644(s^2+31.47s+406)(s^2+27.41s+411.2)(s^2+183.9s+1.762e04)}{(s+0.05292)(s^2+30.38s+376.5)(s^2+71.86s+1697)(s^2+165.2s+1.522e04)}$$

After reduction we can see that it remains 6 zeros and 6 poles for C_{i-min}

$$C_{i-min}(s) = \frac{24.644(s^2+31.47s+406)(s^2+27.41s+411.2)(s^2+183.9s+1.762e04)}{(s^2+30.38s+376.5)(s^2+71.86s+1697)(s^2+165.2s+1.522e04)}$$

$$C_{i-red}(s) = \frac{25.228(s^2 + 33.57s + 636.5)}{s^2 + 62.7s + 2154}$$

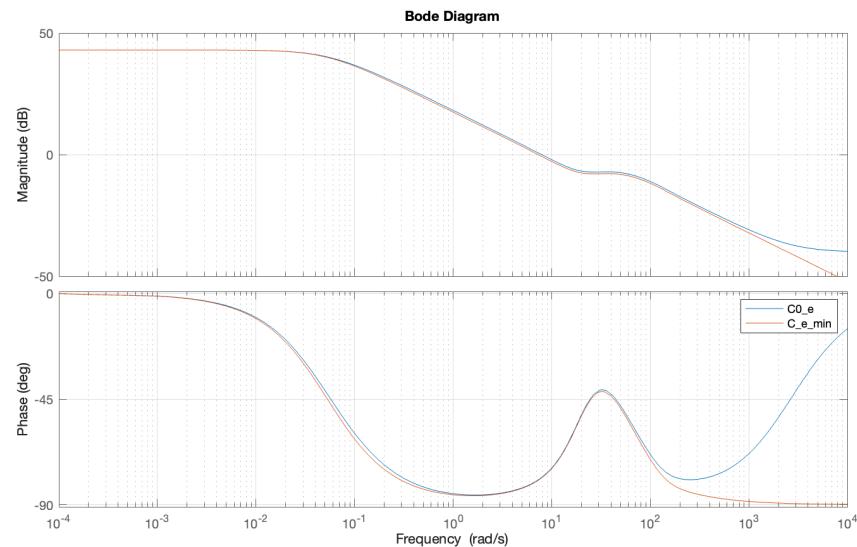


Figure 12: bode plot of C_{e-min} & $C0_e$

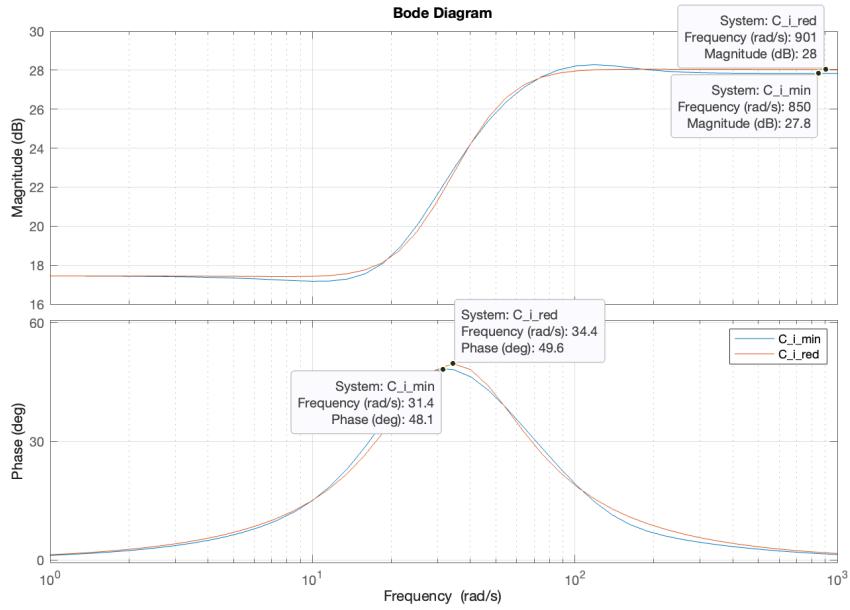


Figure 13: bode plot of C_{i-min} & C_{i-red}

From the figure 17 we can see that the controller C_{i-red} and the controller C_{i-min} are very close in term of gain and phase, they are slightly different in middle frequency range. We can see that the phase advance provide by the controller is around **49 deg**.

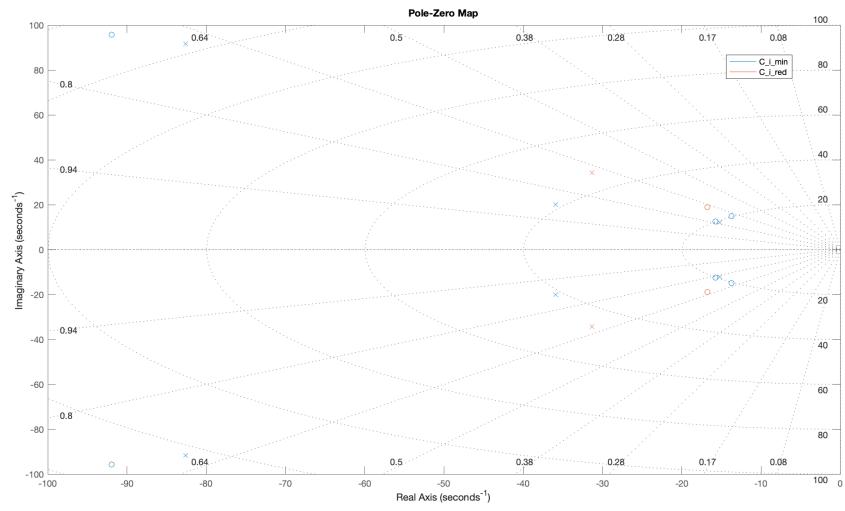


Figure 14: poles-zeros of $C_{i-\min}$ & $C_{i-\text{red}}$

11 Question 3C.3: Controller analysis & simulation

11.1 First exercise

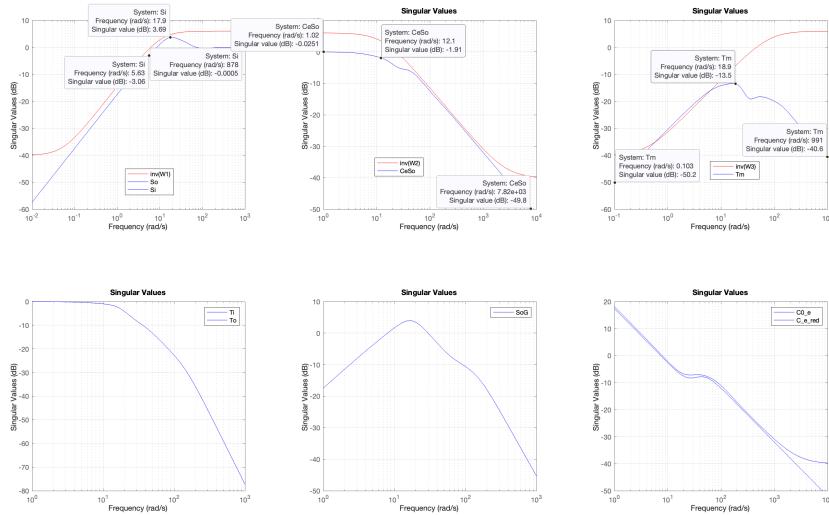


Figure 15: 2x3 plots

Figure 19 shows various results. the top-right figure shows that the model matching constraint is violated. We can also see that $C0_e$ and C_{e-red} are very close.

11.2 Second exercise

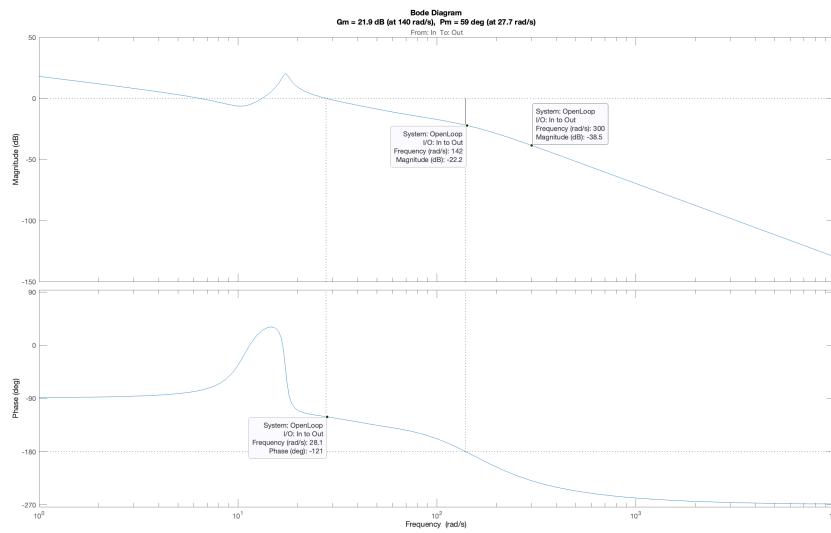


Figure 16: Open loop margin

The delay margin is $DM = \frac{59 \cdot \frac{\pi}{180}}{27.7} \cdot 1000 = 37.1749$ ms. We can see from Figure 20 that the gain of the system at 300 rad/s is -38.5 dB < -30 dB.

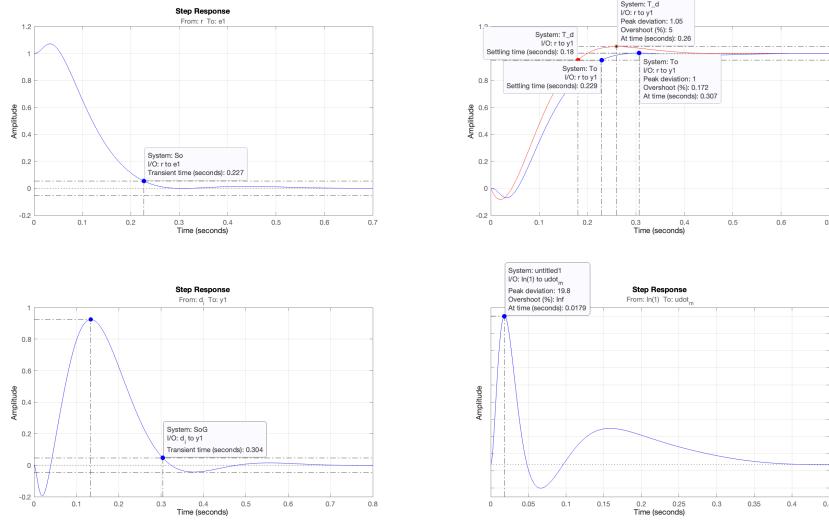


Figure 17: 2x2 times plots

System	Settling time (5%)	Transient time	Peak response
S_o	/	0.227 s	/
T_d	0.18 s	/	5%
T_o	0.229 s	/	0.172%
Sog	/	0.304 s	/
$T_{r \rightarrow \dot{u}_m}$	/	/	19.8 deg/s

We can see also that the settling time of our closed-loop system is greater than 0.18s (the settling time of our model).Indeed the model matching constraint impact the closeness of T_o and T_d . When we try to reduce the gain of W3 we obtain a slower system with a settling time much greater than 0.18s.

12 Question 3D.1: Controller design

Table with the parameter of both controllers.

25.228	27.72
33.57	27.54
636.5	446
62.7	61.94
2154	1527

The values in the previous table look pretty close. *hinstruct* found also the same controller we obtain in the previous part. With *hinstruct* we obtain $\gamma = 1.1289$ which worse than the case with *hinsyn*. It's slightly bigger because use *hinstruct* by imposing a controller structure with less flexibility than the case with *hinsyn*, where the controller was a 9th order controller.

γ_{So}^*	0.9005
γ_{CeSo}^*	1.0122
$\gamma_{T_m}^*$	0.1447

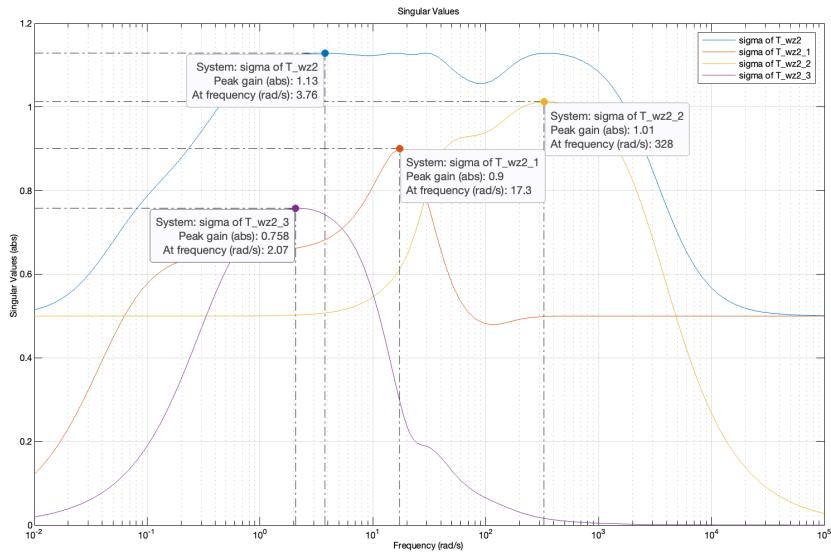


Figure 18: $\sigma(T_{wz})$ et $\sigma(T_{wzi})$, $i = 1, 2, 3$, (*hfinstruct*)

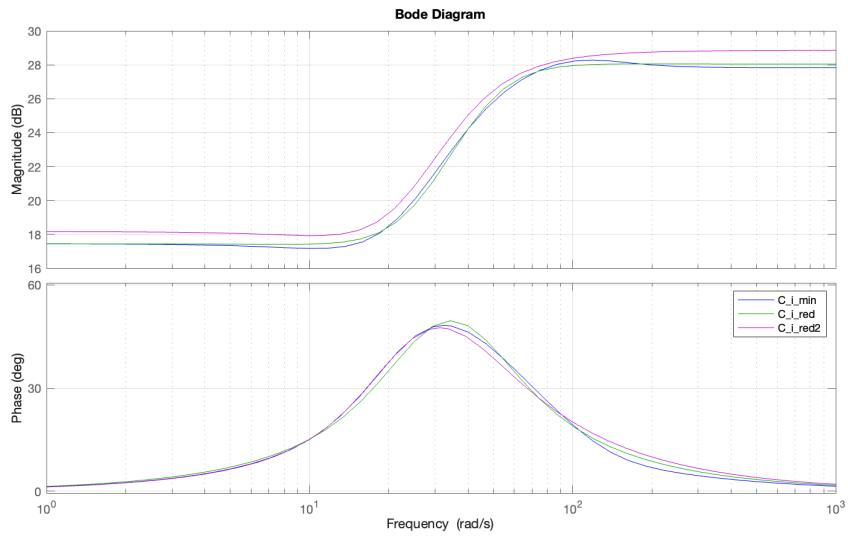


Figure 19: bode plot of C_{i-min} , C_{i-red} & C_{i-red}^* (*hfinstruct*)

The results of the previous show that the controllers are pretty close, but the gain of C_{i-red}^* in magenta is slightly bigger than the gain of the other controller and the advance phase it provides is slightly smaller than the one of other controllers.

13 Question 3D.2: Controller analysis & simulation (5%)

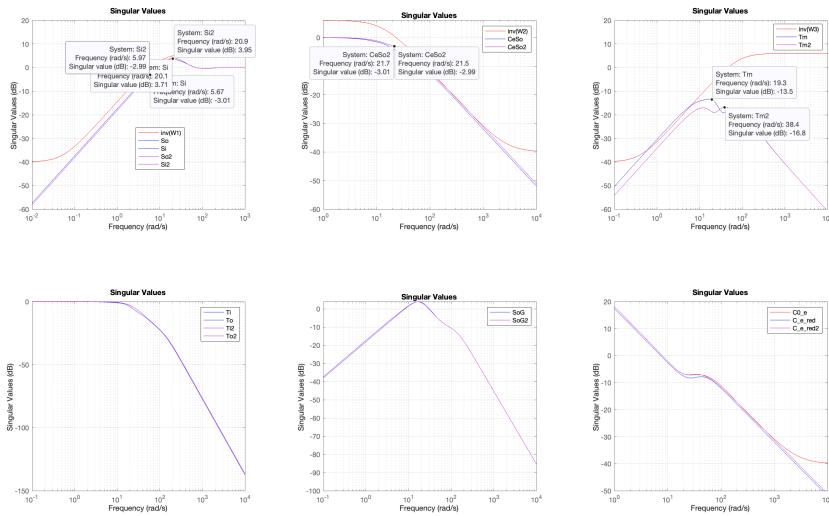


Figure 20: 2x3 plots (*hfinstruct*)

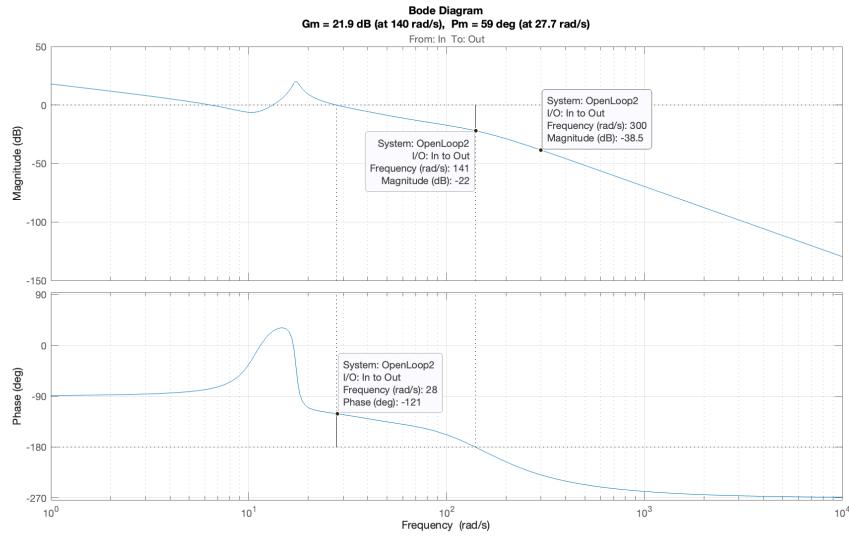


Figure 21: Open loop margin (*hfinstruct*)

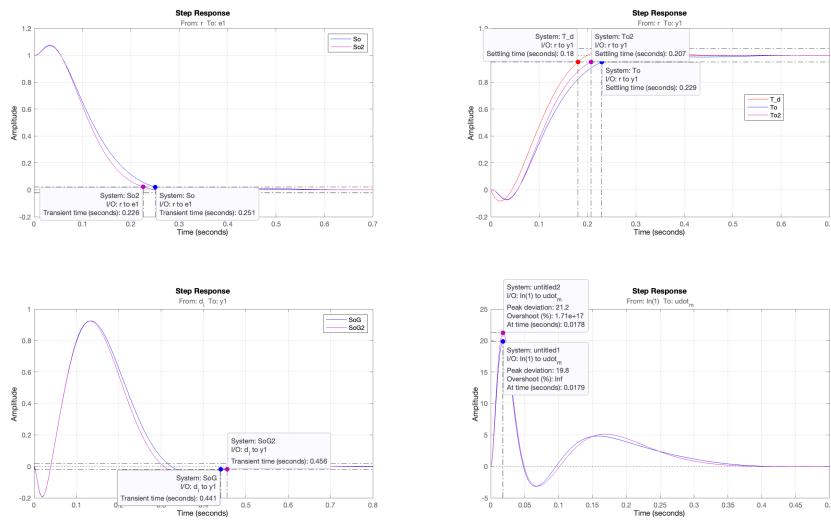


Figure 22: 2x2 times plots (*hfinstruct*)

All the results are pretty close. we can see that the model matching constraint is not violated this time. We have a slightly less margin phase than with hinfsyn. We have also a closed-loop system with a settling time more closed to the settling time of our model. By zooming out on the figure of *CeSo* we can see that the control constraint is violated for the case of *hinfstruct*.

14 Question 3E.1: Controller design (5%)

The first feedforward controller calculated has the following output.
From input "y1" to output:

$$F_{f-init}(s) = \frac{6.0505 \times 10^{-5} (s+15.57)(s-36.64)(s^2+22.37s+330.9)(s^2+38.52s+886.7)(s^2+196.8s+1.962 \times 10^4)}{(s-36.64)(s+36.52)(s^2+25.52s+332.8)(s^2+27.54s+446)}$$

The sigma value plot of the controller

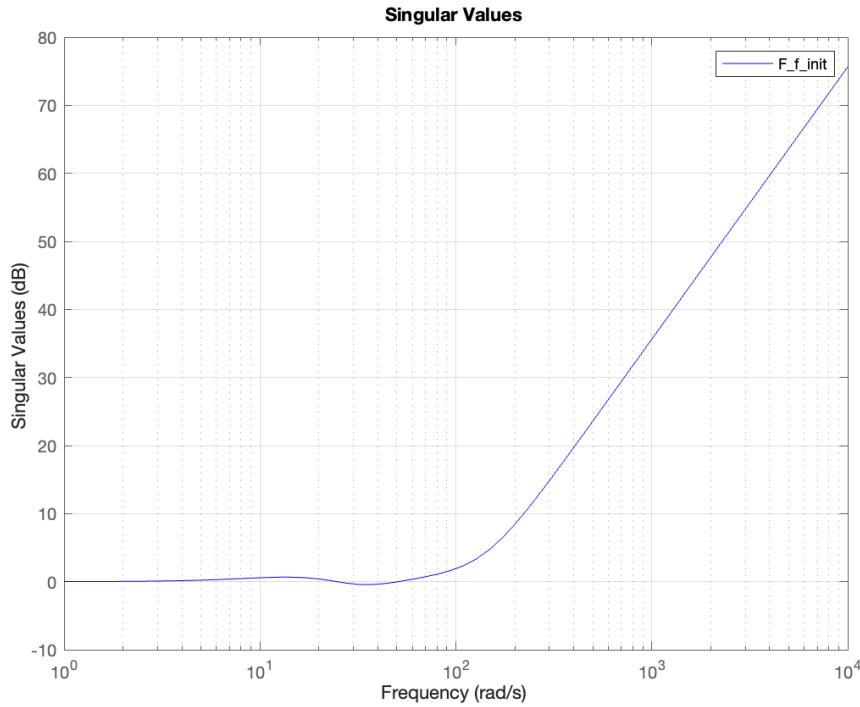


Figure 23: F_{f-init} sigma value plot

As it can be seen on the graph, the values of the graph diverge to infinity when treating the high frequencies. Because of this response, the controller F_{f-init} can't be implemented.

15 Question 3E.2: Controller order reduction (5%)

We are required to reduce F_{f-init} 's order. To do so, we have to remove the RHP pole- zero pole, as well as the pair of high frequency zeros that can be truncated. We can see the zeros and poles on the following figure.

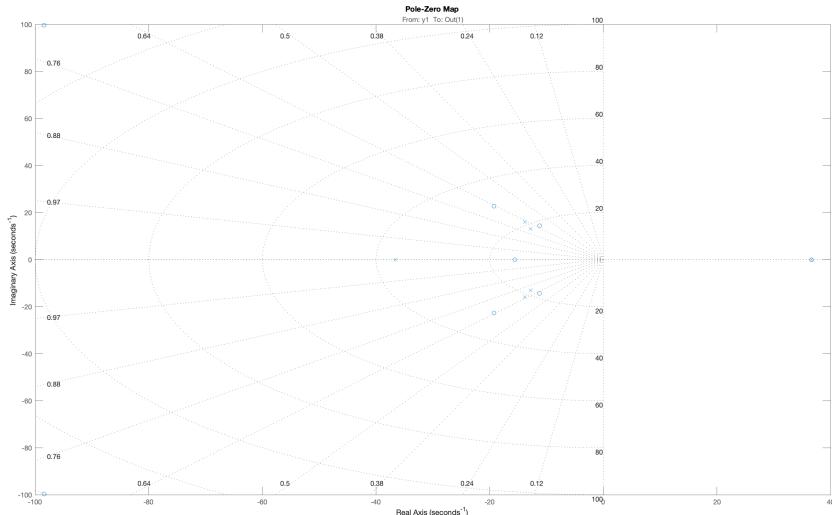


Figure 24: zeros and poles of F_{f-init}

After removing the three zeros and the pole, as well, as compensating the gain, we obtain the truncated controller F_{f-lf} .

$$F_{f-lf}(s) = \frac{1.187(s+15.57)(s^2+22.37s+330.9)(s^2+38.52s+886.7)}{(s+36.52)(s^2+25.52s+332.8)(s^2+27.54s+446)}$$

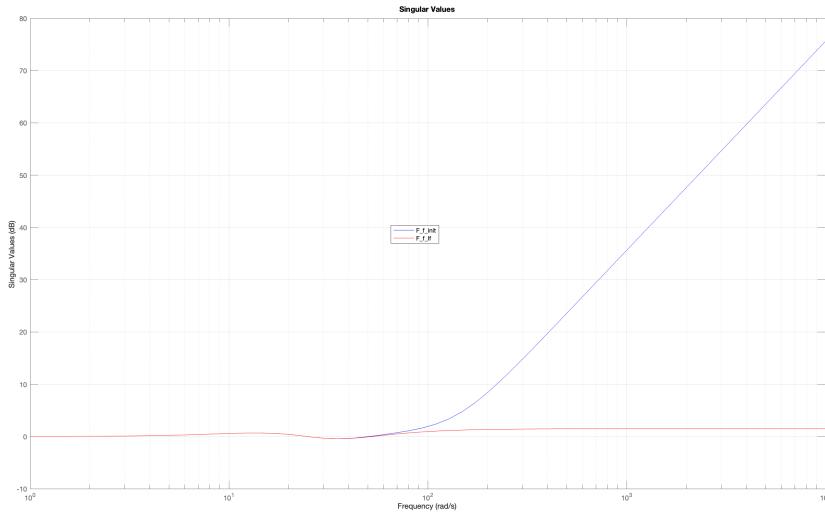


Figure 25: singular values of F_{f-lf} & F_{f-init}

The divergence seen on F_{f-init} has been corrected.
 We can further reduce the order by using model-order reduction techniques.
 We can reduce the model to the order 3. Downgrading the order more would
 degrade significantly the controller frequency at low-mid frequencies.

The output of the final model is:

$$F_f(s) = \frac{1.1856(s+13.39)(s^2+42.3s+1083)}{(s+23.14)(s^2+45.35s+742.7)}$$

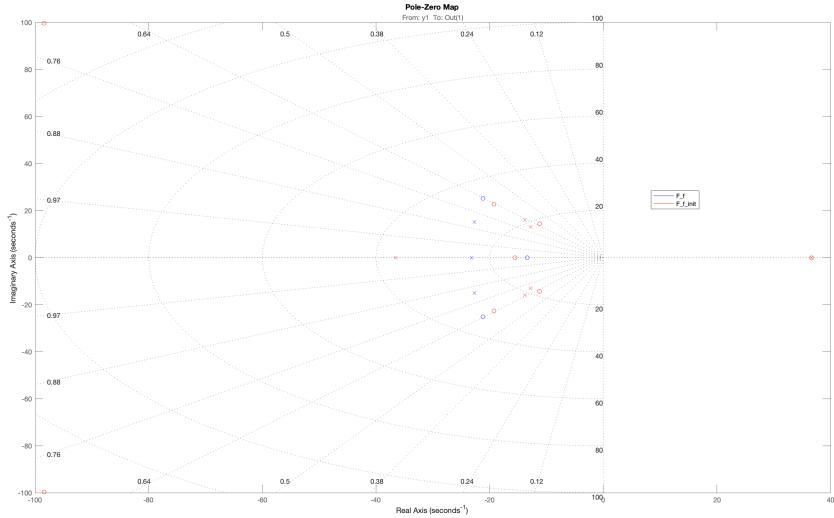


Figure 26: Comparaison of pole zero map

16 Question 3E.3: Controller analysis & simulation (5%)

During this project we have created multiple model matching error functions. The next figures shows the singular values of the 4 models: the weighting filter inverse W_3^{-1} (red), the result from *hinfsyn* (blue), the result from *hinfstruct* (magenta), and the result from *feedforward* (green). We recall that the bandwith ω_1 has a value of 5.2918.

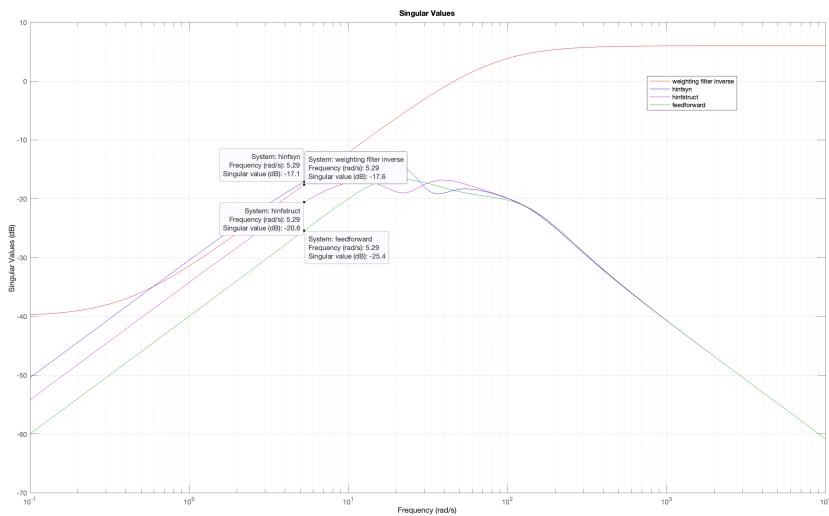


Figure 27: Comparison of the singular values - Model Matching Error

We can see that we have gained from 8.3dB to 4.8dB (range of dB gained) with respect to the no feedforward cases.

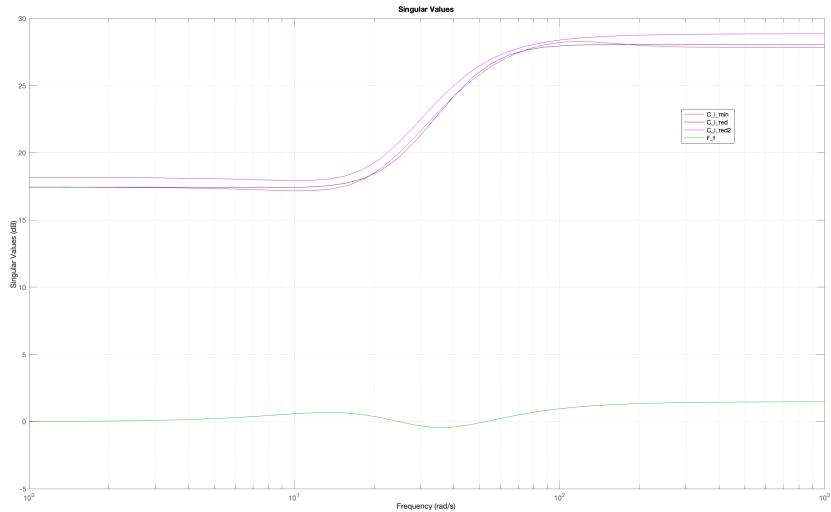


Figure 28: Comparison of the singular values - Controller

All four controllers have a similar trajectory. However we can clearly see a big difference between the gain of the feedforward controller and the rest of the controllers.

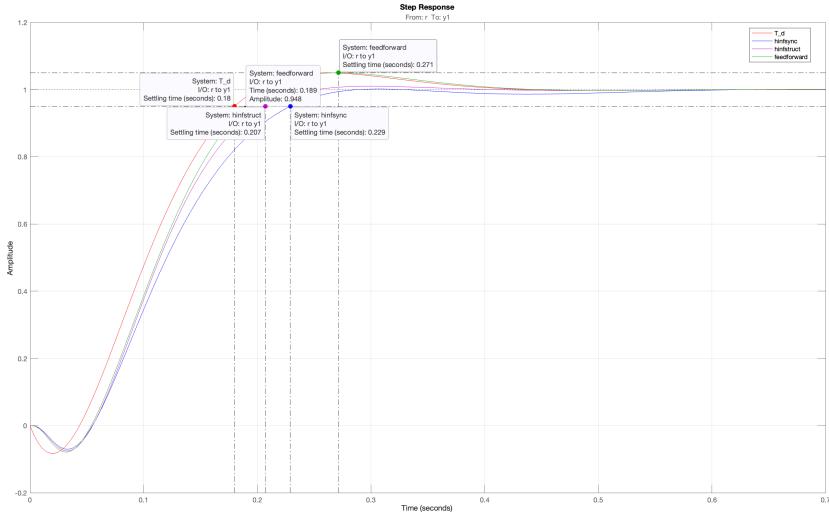


Figure 29: Comparison of the step responses - Closed Loop

It can be seen that the last model with is a little more abrupt in the transient period, resulting in a slight output in $\pm 5\%$ of the final value, so the response time can be approximated to that corresponding to the input in the $\pm 5\%$ of the value, i.e. 0.189s, so we have gained 40ms in terms of $5 \pm \%$ settling time.

However, the response doesn't match the reference model. The reason behind this is the multiple approximations we were forced to do. The initial feed-forward controller, calculated using system inversion, had more zeros than poles. This meant that the model couldn't be implemented in our answer. Therefore, we have a difference in the settling time.

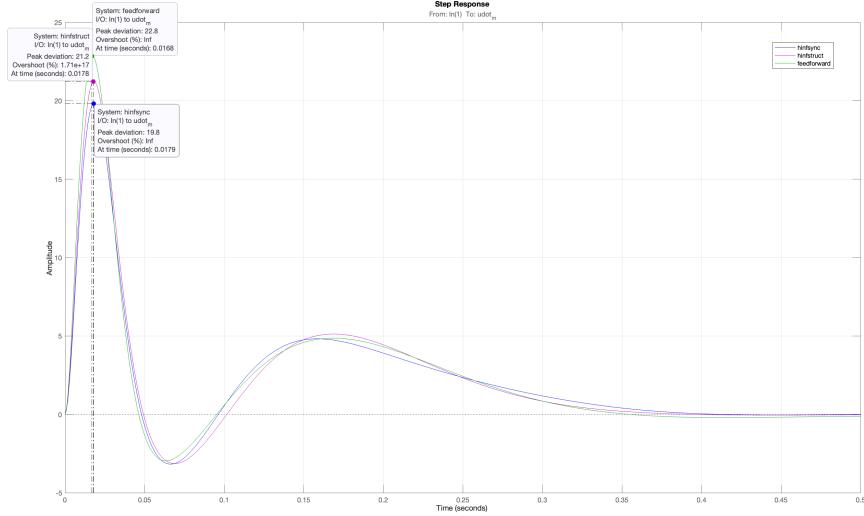


Figure 30: Comparison of the step responses - Actuator

We can clearly measure that the peak responses of all the created models don't reach the saturation level of $25^{\circ}/sec$. The closest we are to the saturation level is $22.8^{\circ}/sec$, which means a significant $2.2^{\circ}/sec$ difference. Finally, after presenting the four graphs to compare all the developed models, we can regroup them into the following graph.

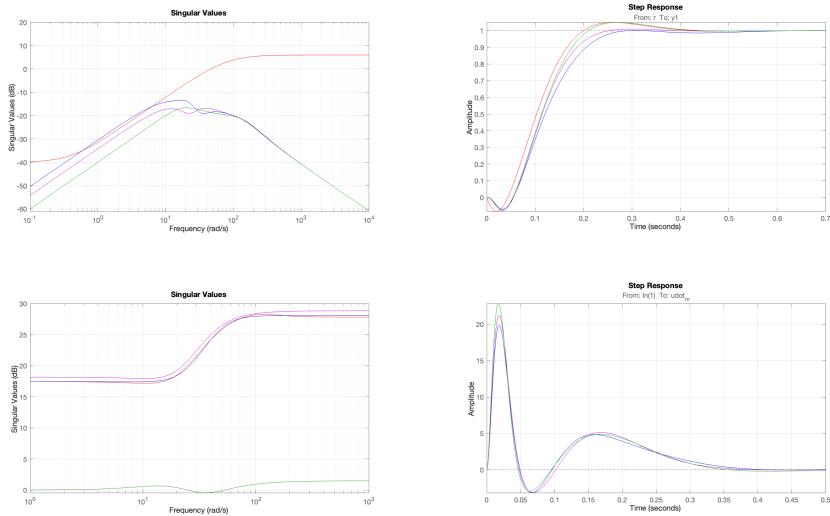


Figure 31: 2 x 2 Time Frequency plots