Rapport BE CSLC

BÉNISTANT Raphaël, BOURLOT Xavier, 3IMACS-AE-C, Binôme 2

30 mai 2020

Table des matières

1 Introduction

Un des objectifs de ce TP est d'introduire à l'utilisation avancée du logiciel MATLAB pour l'étude des systèmes. Il a également pour but de synthétiser l'ensemble des connaissances théorique concernant la modélisation et la commande des systèmes linéaires. Pour ce faire, nous allons étudier le comportement d'un avion en phase de croisière. Son modèle sera donné par la suite mais il est important de noter que les variables d'entrées et de sorties sont les suivantes :

- ▷ **Variables d'entrée** $u = [\delta_c, a_{prop}]^T$
  - a_{prop} : poussée réacteur, agit directement sur la vitesse de l'avion.
  - δ_c : angle de la gouverne de profondeur, le braquage de cet angle tend à faire cabrer ou piquer l'avion.
- ▷ **Variables d'état** $x = [v, \alpha, \theta, q]^T$
  - v : vitesse de l'avion.
  - α : angle dit d'attaque (entre l'axe logitudinal de l'avion et sa vitesse).
  - θ : angle dit de tangage (entre l'axe longitudinal de l'avion et le plan horizontal).
  - q : vitesse angulaire en tangage de l'avion $q = \frac{d\theta}{dt}$.
- ▷ **Variables de sortie** $y = [v, \alpha, \theta, q, \gamma]$
  - $(v, \alpha, \theta, q]^T)$: les quatre variables d'état.
  - γ : l'angle de vol $\gamma = \theta - \alpha$ ($\gamma > 0$ signifie que l'avion prend de l'altitude et $y < 0$ qu'il en perd).

Les variables les plus complexes à décrire sont illustrées sur le schéma suivant :

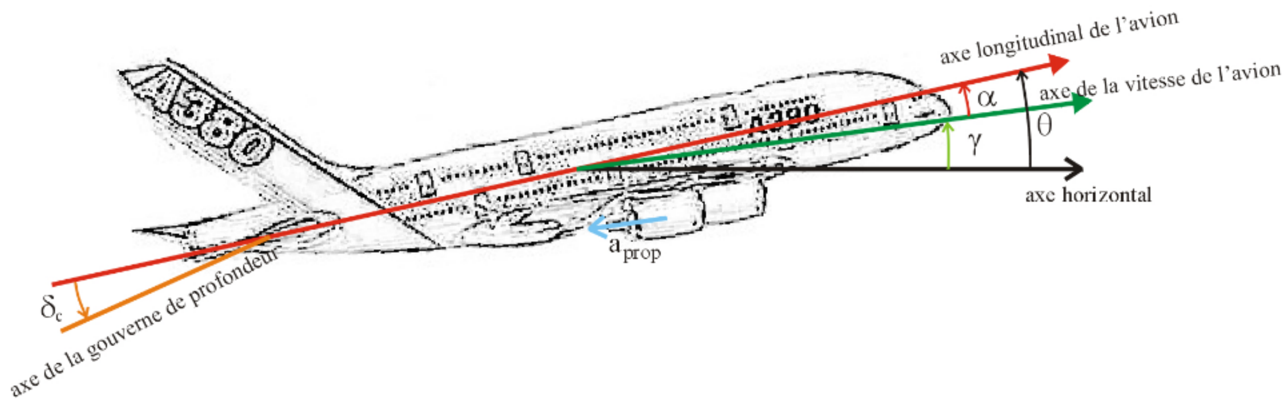

FIGURE 1 – Position des axes de l'avion

2 Etape 1 : Mise en place des outils de simulation

2.1 Question 1 :

Afin de modéliser le système dont le modèle d'état est le suivant :

$$\begin{cases} \dot{x} &= \begin{bmatrix} -0,0118223 & -0,088571 & -9,78 & 0 \\ -0,003038 & -1,2563 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0,0617 & -28,075 & 0 & -4,5937 \end{bmatrix} x + \begin{bmatrix} 0 & 1,1972 \\ 0 & -0,0012 \\ 0 & 0 \\ 7,84 & -4,05 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda & 0 \\ 0 & -\lambda & \lambda & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} u \end{cases}$$

Nous avons mis en place un modèle Simulink qui permet de simuler le comportement de celui-ci :

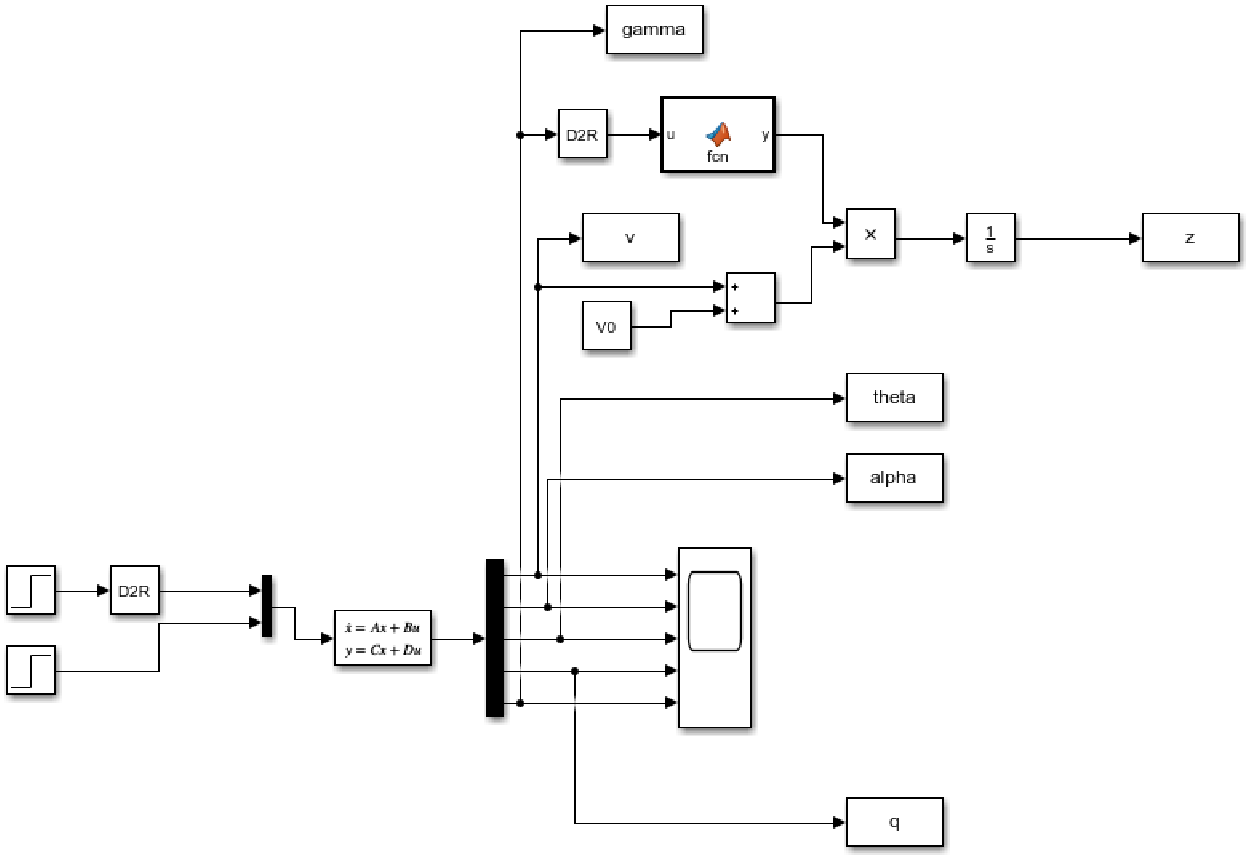

FIGURE 2 – Schéma Simulink pour le système en boucle ouverte avec 2 échelons en entrée

Après simulation on observe les résultats suivant :

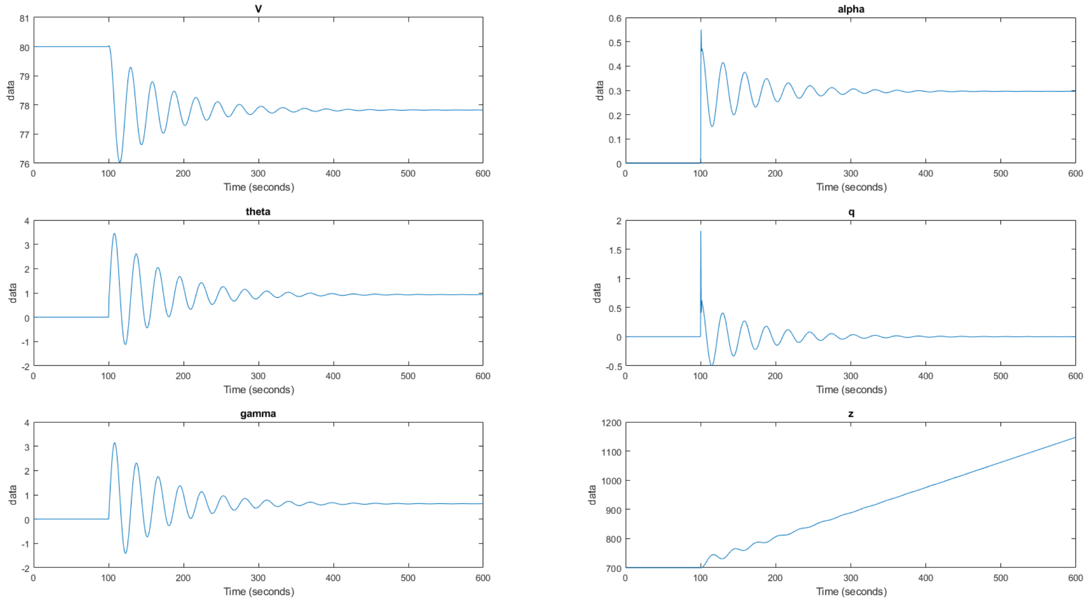

FIGURE 3 – Résultats de la simulation du modèle pour 2 échelons en entrée

On remarque les différentes variables du sytème oscillent de manière importante. Les résultats semblent cohérents puisqu'un échelon positif sur la gouverne de profondeur δ_c doit entrainer une montée de l'avion est donc une décélération. C'est bien ce que nous abservons sur nos résulats puisque z augmente et v diminue.
Un échelon sur la propulsion de l'avion a_{prop} devrait fair accélérer l'avion. Comme l'avion est ralenti par sa montée, il semble que la propulsion ne soit pas assez importante pour contrer la décélération engendrée.

2.2 Question 2 :

Nous souhaitons désormais étudier l'influence de chaque entrée sur les variables observées lors de la simulation.
Pour ce faire nous utiliserons le précédent modèle Simulink en modifiant les valeurs des échelons.

Etudions d'abord l'influence de la gouverne de profondeur δ_c. Nous fixons la valeur de la propulsion à $a_{prop} = 0.1 m.s^{-2}$ et un échelon unitaire pour la gouverne de profondeur δ_c. Après simulation, on observe les résultats suivants :

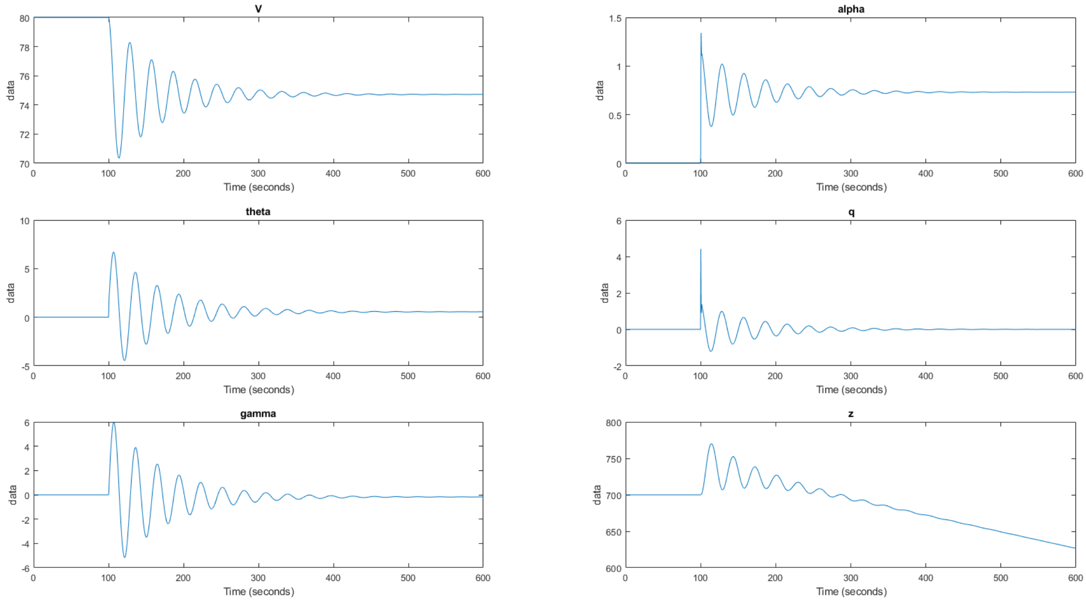

FIGURE 4 – Résultats de la simulation du modèle pour δ_c positif

Les résultats obtenus sont exactement ceux escomptés en théorie. Un braquage positif de la gouverne δ_c entraine une modification de la cambrure et donc un déplacement du point d'équilibre vers une incidence α plus grande et de ce fait une vitesse v plus faible.

Etudions maintenant l'effet d'un braquage négatif. Nous fixons la valeur de la propulsion à $a_{prop} = 0.1 m.s^{-2}$ et un échelon unitaire négatif pour la gouverne de profondeur δ_c. Après simulation, on observe les résultats suivants :

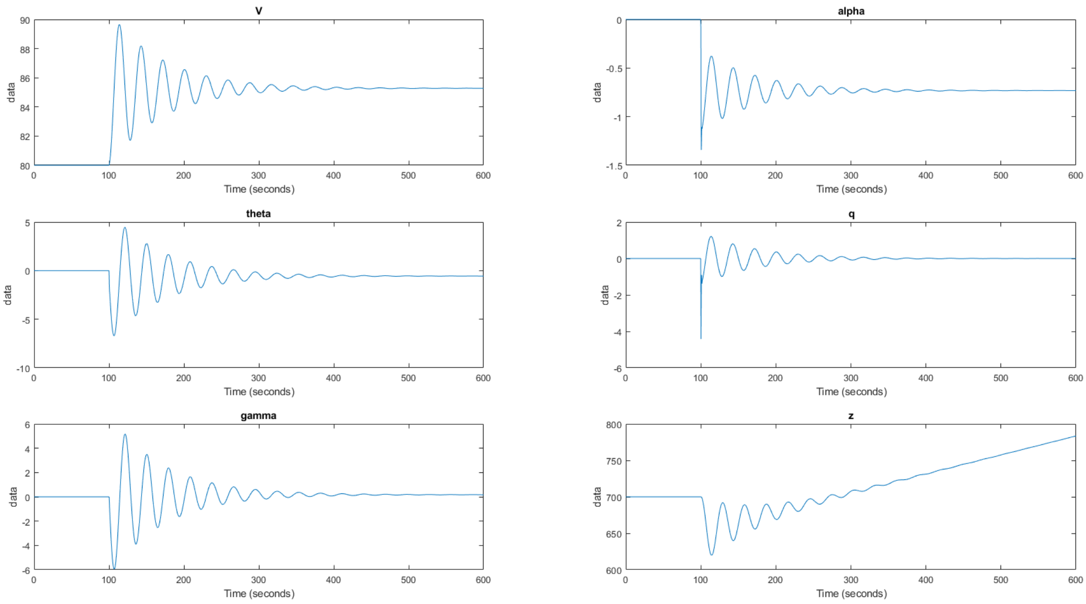

FIGURE 5 – Résultats de la simulation du modèle pour δ_c négatif

Les résultats obtenus sont exactement ceux escomptés en théorie. Les résultats sont symétriques aux résultats observés pour δ_c. Tous les phénomènes sont opposés.

Finissons maintenant par étudier l'influence de l'action sur la propulsion a_{prop}. Nous fixons la valeur de la propulsion à $a_{prop} = 0.1m.s^{-2}$ et celle de la gouverne de profondeur à $\delta_c = 0°$. Après simulation, on observe les résultats suivants :

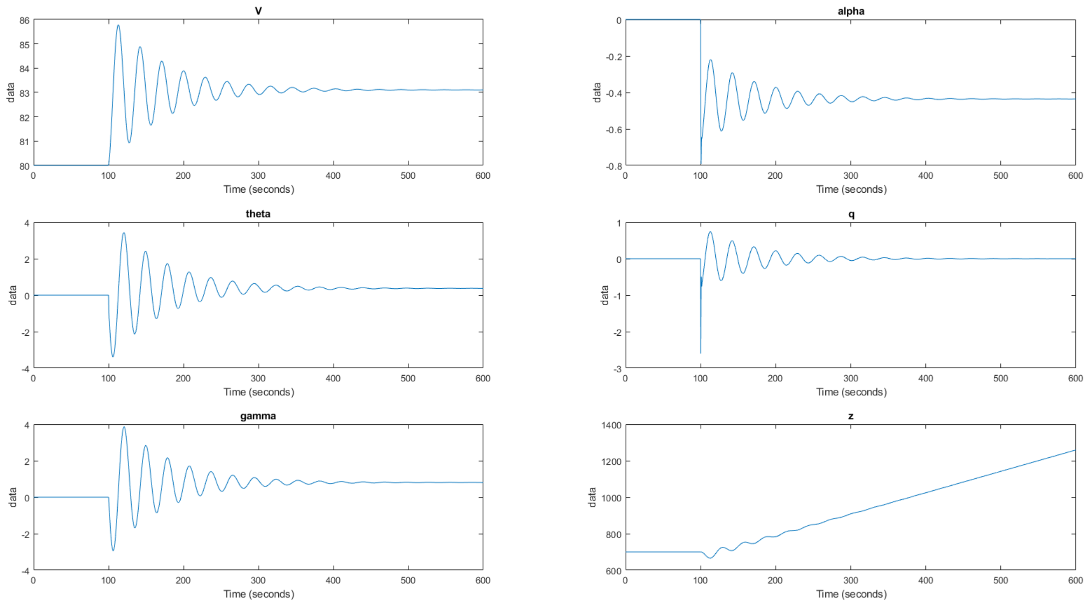

FIGURE 6 – Résultats de la simulation du modèle pour δ_c nul

L'action sur la propulsion a_{prop} a pour effet prépondérant de modifier la vitesse mais aussi de modifier les angles α et θ.

3 Etape 2 : Analyse du modèle linéaire en boucle ouverte

3.1 Question 3 :

Nous voulons désormais caractériser entièrement notre système en déterminant l'ensemble des pôles et des zéros ainsi que la valeur du gain du système en boucle ouverte.
Grâce respectivement aux fonctions "poles", "zeros" et "dc(gain)", il est possible d'obtenir les résulats précédents. On trouve :

$$poles = [-2,92 \pm 5,02j, -1,41.10^{-2} \pm 2,17.10^{-1}j]$$
$$zeros = [-2,601 + 7,558i, -2,601 - 7,558i, -0,646]$$
$$K_{0,BO} = 31.0452$$

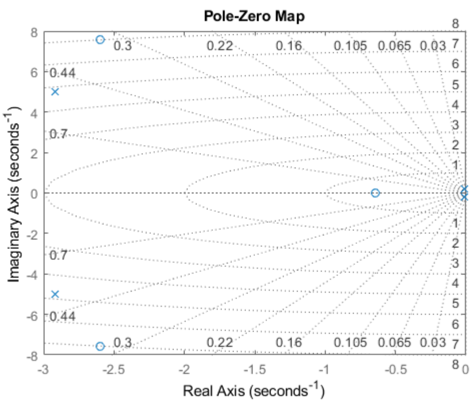

FIGURE 7 – Pôles et zéros du système en boucle ouverte

On peut alors en déduire les pôles phugoïdes qui sont les pôles les plus lents et les pôles les plus rapides :

$$poles_{phugoides} = -1,41.10^{-2} \pm 2,17.10^{-1}j$$
$$poles_{rapides} = -2,92 \pm 5,02j$$

Les réponses à un échelon respectivement du mode rapide et du mode phugoïde sont les suivantes :

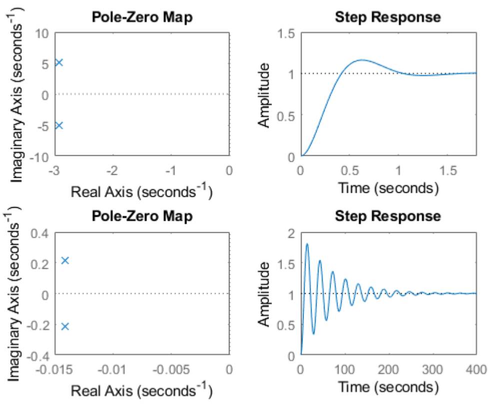

FIGURE 8 – Réponse indicielle respectivement du mode rapide et du mode phugoïde

On en déduit que le temps de réponse du mode rapide est de 9s environ contre 120s environ pour le mode phugoïde. La fonction "damp" permet de déterminer l'amortissement pour chacun des modes. On obtient les résultats suivant :

$$\xi_{phugoides} = 6,47.10^{-2}$$
$$\xi_{rapides} = 5,03.10^{-1}$$

3.2 Question 4 :

On observe le comportement des pôles phugoïdes dans le domaine fréquentiel et en réponse à un échelon. On donne en orange le système d'ordre 4 à titre de comparaison.

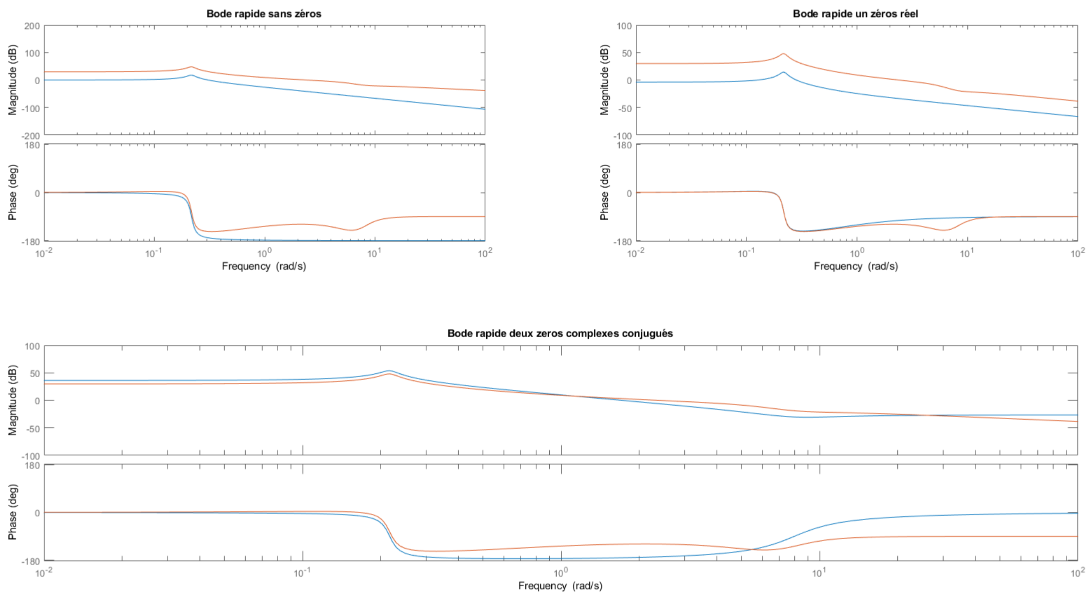

FIGURE 9 – Réponse fréquentielle du mode phugoïde avec 0, 1 et 2 zéros

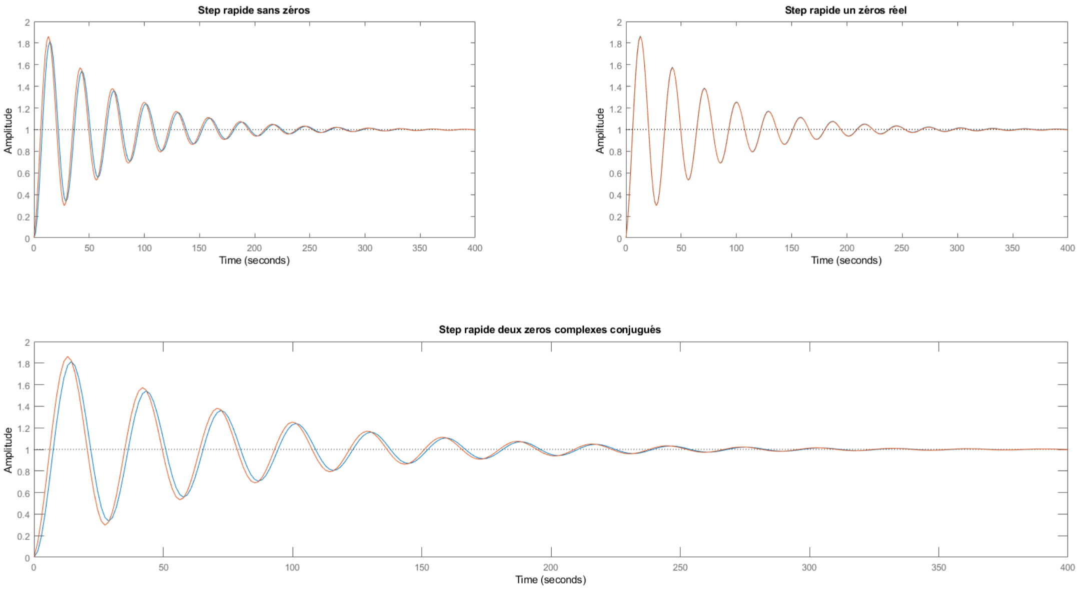

FIGURE 10 – Réponse à un échelon du mode phugoïde avec 0, 1 et 2 zéros

On observe que le mode phugoïde avec 2 zéros est celui qui approche au mieux le système d'ordre 4.

4 Etapes 3 : Régulation de la vitesse

4.1 Question 5 :

Nous souhaitons désormais mettre en place une boucle de régulation sur la vitesse v à partir de la commande de la poussée du réacteur a_{prop}. Ceci nous permettra par la suite de travailler à partir d'un vitesse de vol de l'avion constante de manière à ce qu'un changement d'altitude provoqué par une variation de l'angle de la gouverne de profondeur δ_c n'entraîne pas une modification de vitesse.
Nous mettons en place la commande $u = -K_v.v$ qui donne le modèle Simulink suivant :

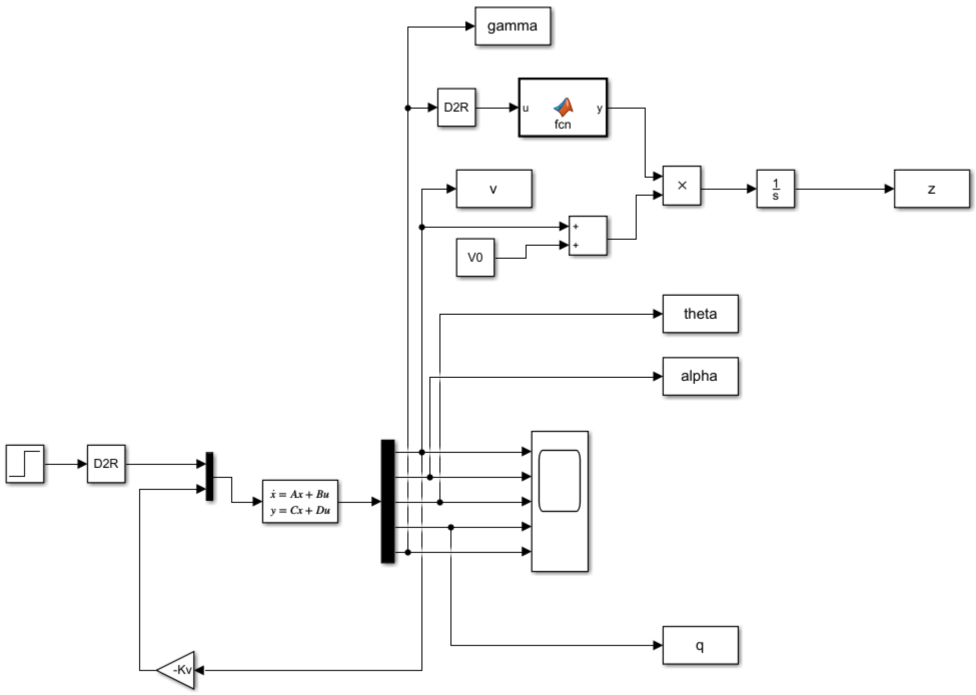

FIGURE 11 – Schéma Simulink du sytème avec commande proportionnelle à la vitesse

Après simulation, avec un échelon de 5 pour la gouverne de profondeur δ_c, nous obtenons les résultats suivants :

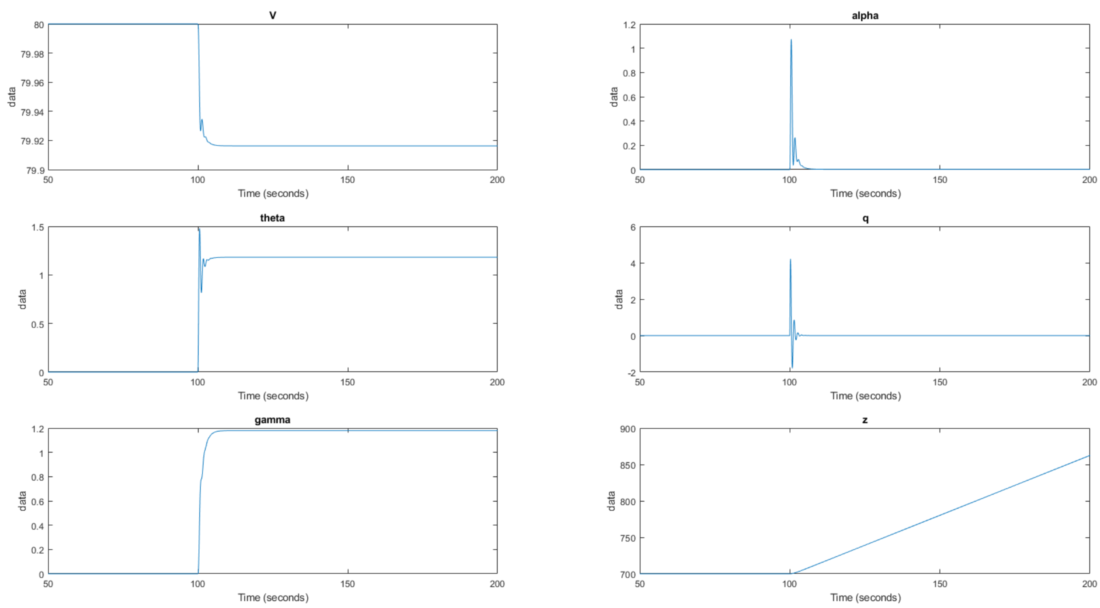

FIGURE 12 – Résultat pour le système avec une commande proprotionnelle à la vitesse

Nous remarquons que le système oscille bien moins que précédemment. La vitesse est stable et varie très peu. Pour toutes les grandeurs, on observe des dépassements très importants qui sont dûs à l'absence de correcteur.

4.2 Question 6

Afin de déterminer quelle valeur de K_v est nécessaire pour notre système, nous avons mis en place un script MATLAB qui nous a permis de tracer le lieu des racines du système. Nous avons fait en sorte que notre script s'arrête lorsque la valeur de K_v correspond au point de séparation. Les résultats son les suivants :

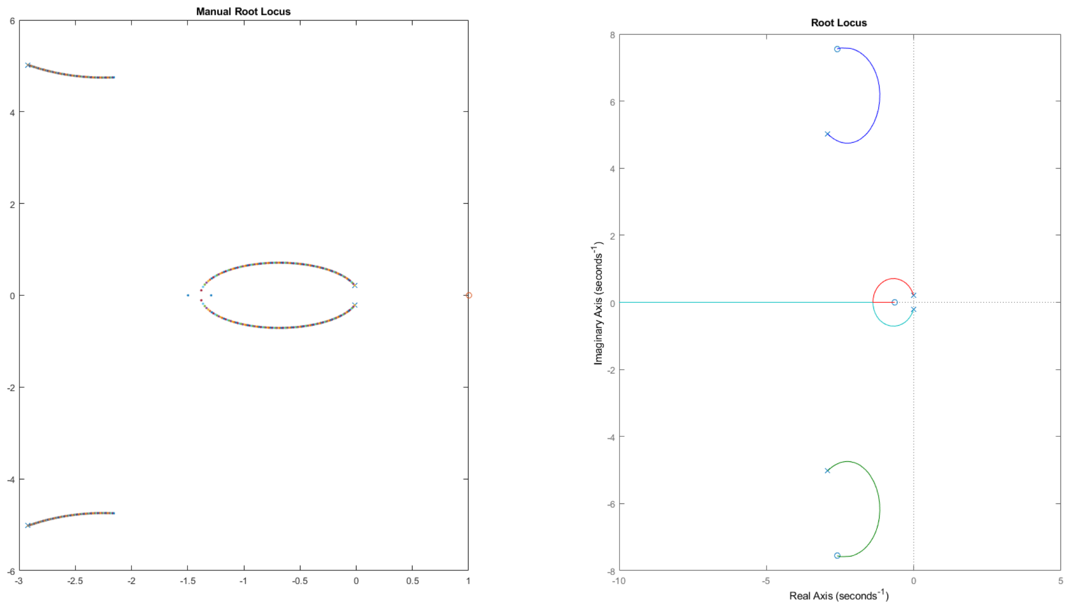

FIGURE 13 – Lieu des racines du sytème manuel et automatique

Nous obtenons les mêmes résultats que la fonction "pzmap". La valeur de K_v au point de séparation est donc :

$$K_v = 1.041$$

4.3 Question 7 :

Le mode phugoïde étant le plus lent, c'est lui qui détermine le temps de réponse global du système d'ordre 4. On choisit donc ce mode. D'après la question 4, le système avec 2 zéros fournit la meilleure approximation, on l'utilisera donc pour répondre à la question.

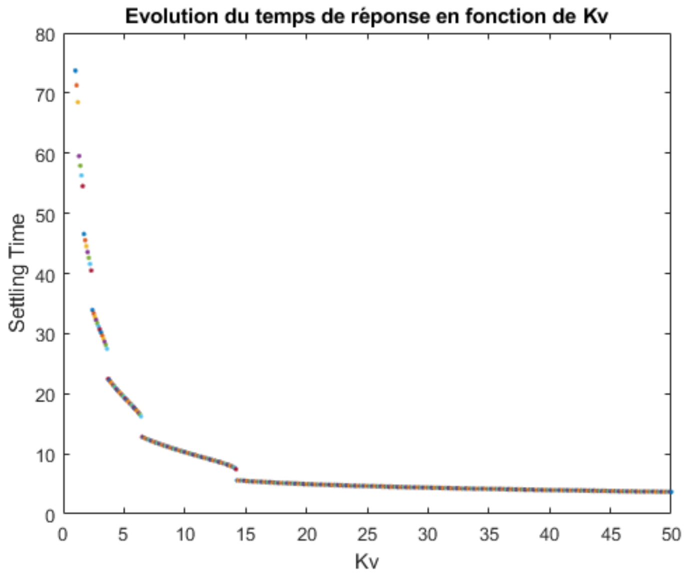

FIGURE 14 – Temps d'établissement du système approximé en fonction de la valeur de K_v

On déduit que le temps d'établissement diminue avec l'augmentation de Kv. En revanche plus Kv est grand, plus le risque de saturer les actionneurs est élevé.

4.4 Question 8 :

Nous voulons maintenant déterminer la valeur de K_v qui nous donne la paire de pôles complexes du mode phugoïde correspondant à une valeur d'amortissement de $\xi = 0,95$. Pour ce faire, nous avons utilisé la commande "sisotool" :

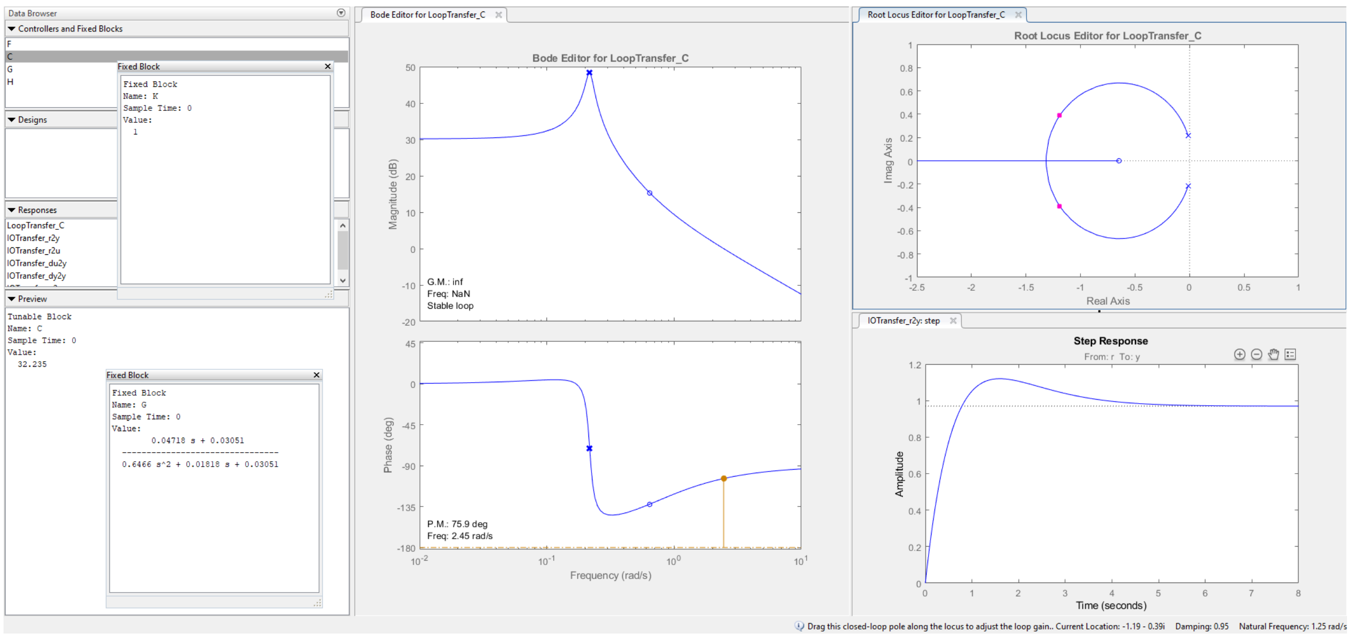

FIGURE 15 – Résultat de la commande sisotool pour notre système

On obtient une valeur de $K_v = 32.2446$
On trace alors la réponse indicielle du sytème avec cette valeur spécifique de K_v :

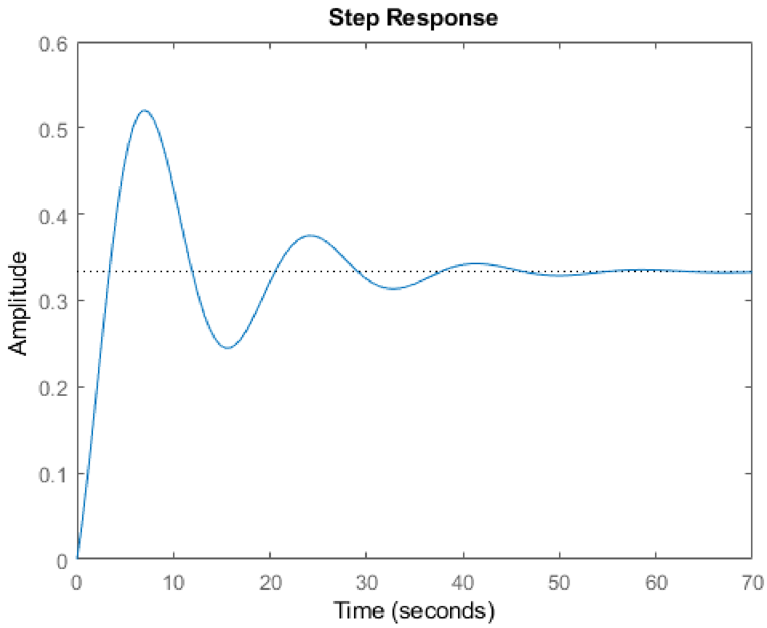

FIGURE 16 – Réponse indicielle avec K_v

4.5 Question 9 :

On répète la manipulation précédente sur le système d'ordre 4. On obtient la fenêtre sisotool suivante :

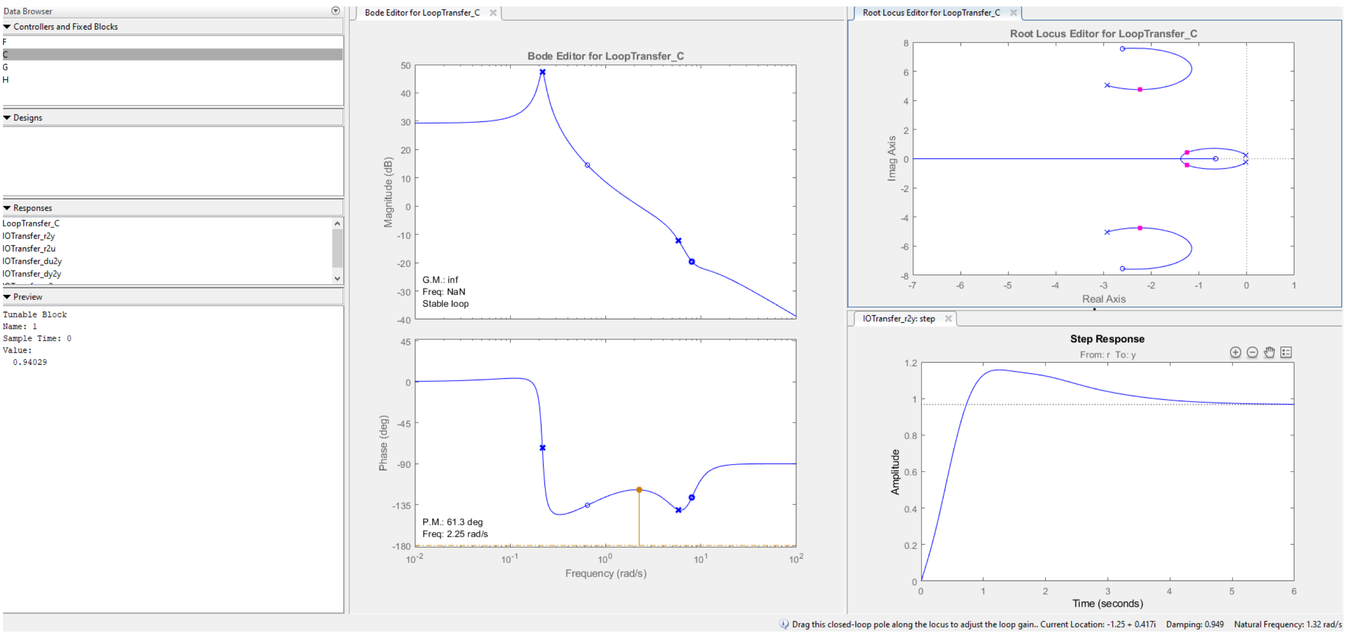

FIGURE 17 – Résultat de la commande sisotool pour notre système

On obtient un coefficient de 0.94029. On en déduit la valeur de $K_v = 33.017$

Les valeurs de K_v trouvées avec le système complet et le système d'ordre 2 sont très proches (à moins de 3%), donc notre approximation est pertinente.

4.6 Question 10 :

On utilise de nouveau sisotool pour répondre à cette question :

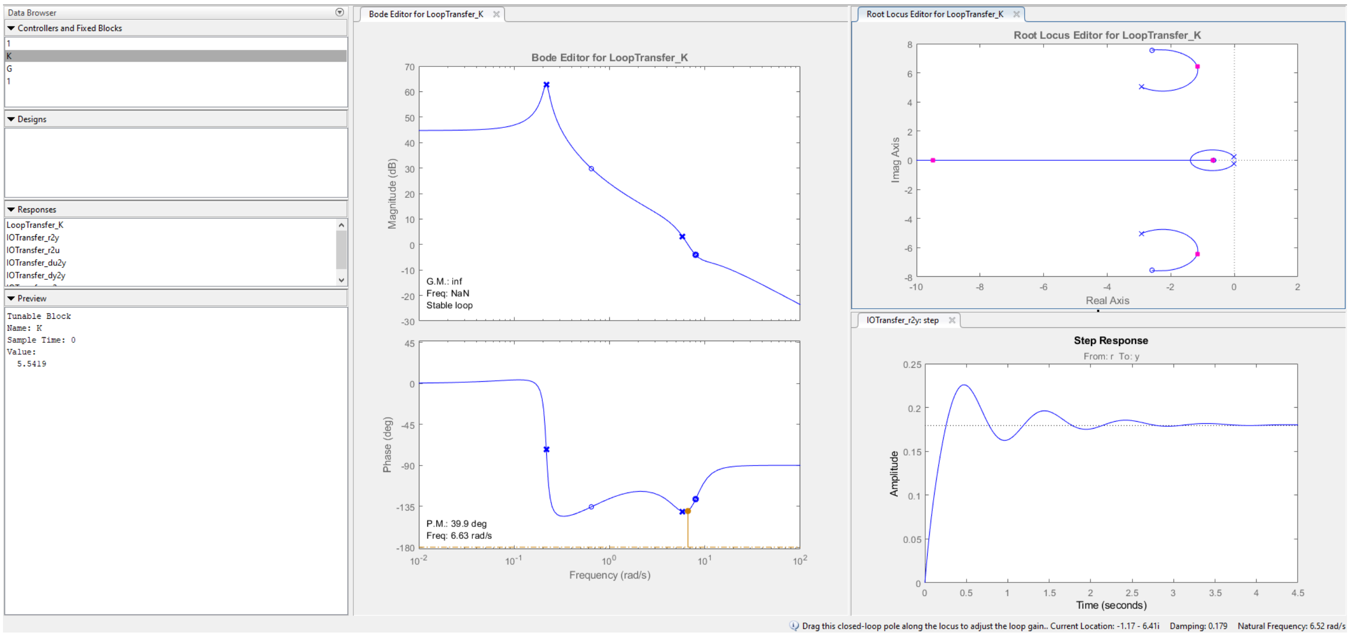

FIGURE 18 – Résultat de la commande sisotool pour notre système

On peut relever 3 coefficients pour lesquels la marge de phase est de 40 degrés : $K_v = [.107, 3.37, 5.5419]$.
On spécifie également un temps d'établissement le plus rapide possible, qui est atteint pour la plus grande valeur de K_v c.a.d pour $K_v = 5.5419$.

4.7 Question 11 :

On simule le système avec la valeur de K_v obtenue à la question précédente. Le schéma simulink est le suivant :

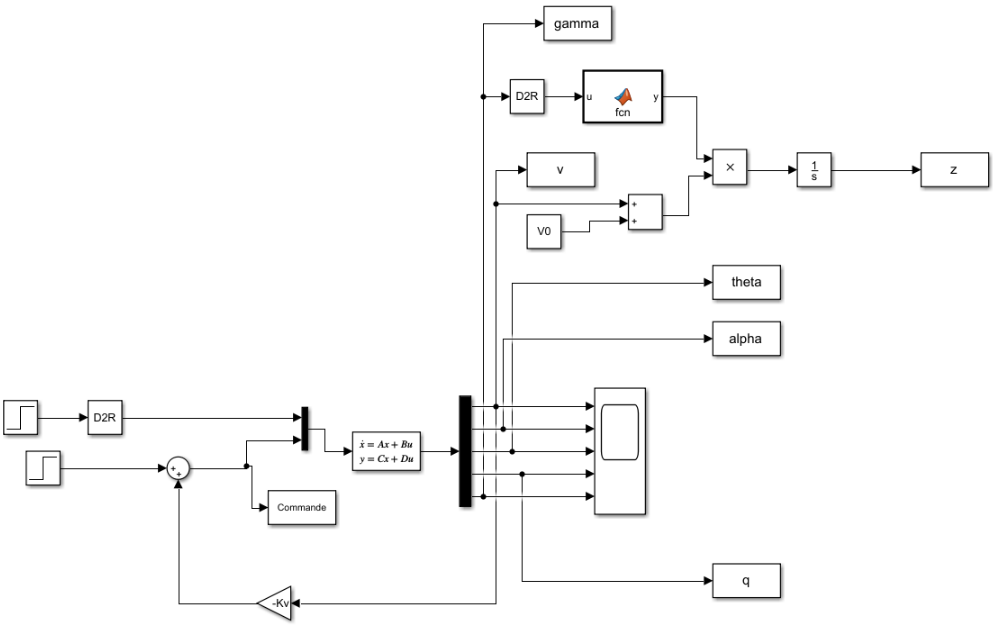

FIGURE 19 – Schéma simulink

On entre un échelon de 5 degrés. Le système semble toujours bien asservi.

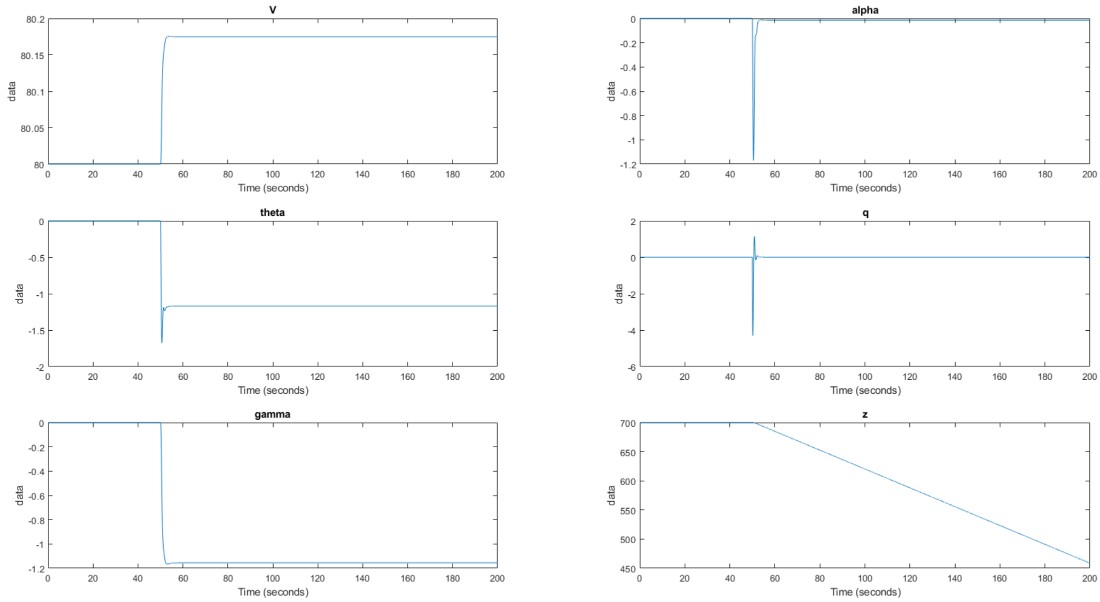

FIGURE 20 – Step de 5 degrés, réponse du système

On s'intéresse maintenant de plus près à la vitesse, et à la commande a_{prop} :

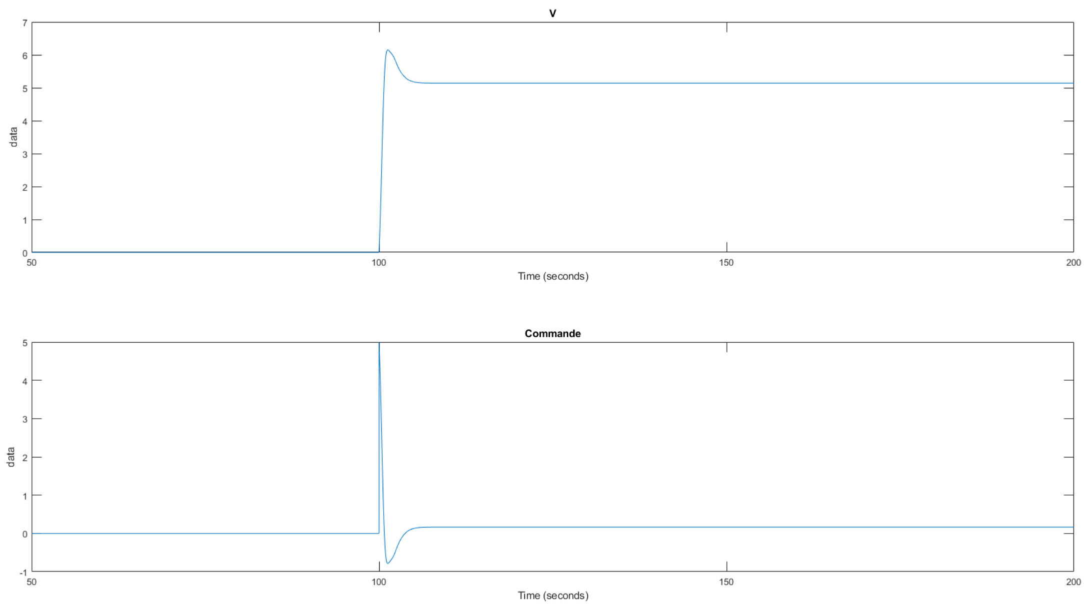

FIGURE 21 – Step de 5 degrés, vitesse et commande a_{prop}

La spécification en marge de phase imposée au système induit une forte action sur la commande. Bien que la vitesse soit régulée efficacement, on peut déplorer la contrainte importante que l'on applique sur a_{prop}. Cette spécification pousse le système à saturer les actionneurs, ce que notre modèle ne prend pas en compte, et risquerait de ne pas fonctionner en situation réelle.
IL faut donc veiller à garder un cahier des charges raisonnable, permettant une régulation effective sans trop contraindre le

système.

4.8 Question 12 :

On s'intéresse maintenant à l'énergie totale nécessaire pour la commande. C'est un critère important pour tout système réel, et nous nous proposons d'observer l'influence du choix de K_v sur cette consommation.
Pour chaque valeur de K_v, on simule le système pour une entrée constante sur a_{prop} et on récupère la sortie $v(t)$.
Ensuite, on calcule le critère de performance donné dans l'énoncé, qui prend en compte l'erreur de vitesse, mais aussi l'énergie de la commande.
On effectue ce calcul pour différentes valeurs de r, r étant un coefficient de pondération entre les deux critères de performance précédemment évoqués. On obtient les courbes suivantes :

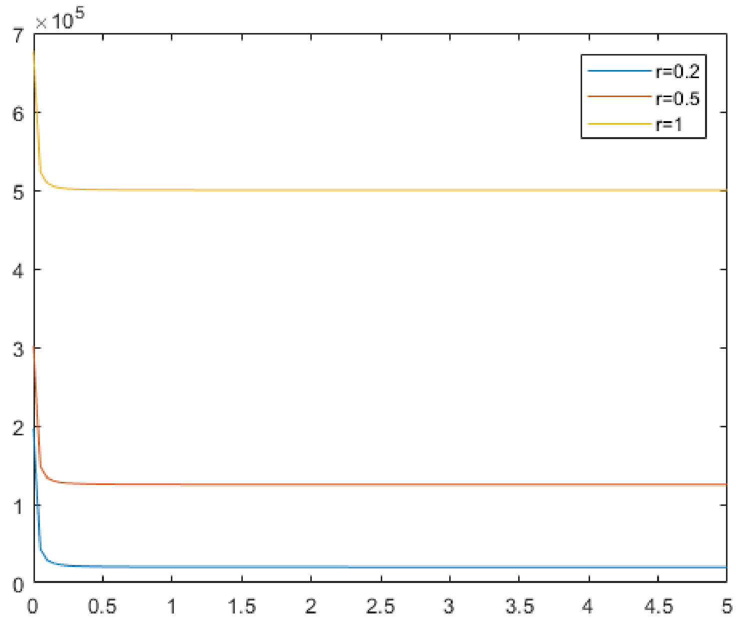

FIGURE 22 – $J_{K_v}(u)$

On remarque que cette fonction est strictement décroissante. Ainsi, il semble que le critère de performance soit satisfait pour les valeurs de K_v les plus grandes possibles, peu importe la valeur du coefficient de pondération r. Ici la simlulation s'arrête pour $K_v = 5$. On représente donc la réponse indicielle du système pour $K_v = 5$.

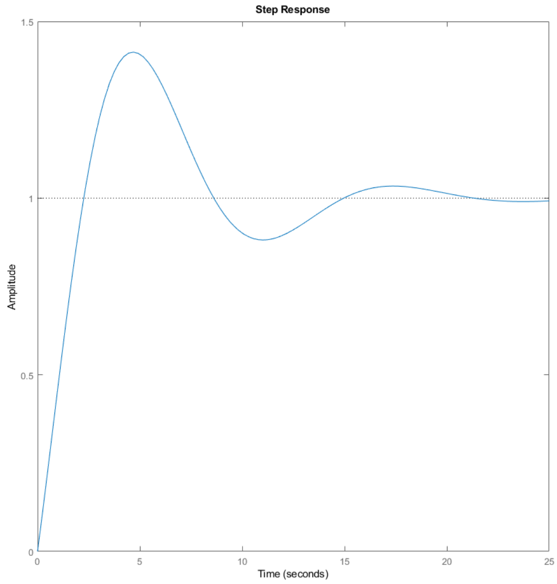

FIGURE 23 – Step du système de 5 degrés pour K_v minimisant le critère de performance

Il est curieux que ce critère de performance décroisse strictement en fonction de K_v. Nous avons peut-être commis une erreur de calcul, car il semble logique que l'énergie consommée augmente avec l'acroissement du gain.

5 Conclusion

A travers ce bureau d'étude Matlab, nous avons approché différentes méthodes d'analyse d'un système complexe. Nous avons pu utiliser des techniques variées de l'automatique, et nous familiariser avec les outils associés. L'application des méthodes vues en cours sur un cas concret nous a permis d'apprécier leur fonctionnement et leur efficacité.
L'interprétation des résultats est une part essentielle de ce travail, car il a fallu comprendre l'importance du choix d'un modèle approprié, ainsi que les limites des outils utilisés.

A Annexes

A.1 Code Matlab

breaklines

```
clear all
clc
close all

A =[-0.018223 -0.088571 -9.78 0
    -0.003038 -1.2563 0  1
    0 0 0 1
    0.0617 -28.075 0 -4.593];

B  = [ 0 1.1962
    0 -0.0012
    0 0
    7.84 -4.05];
C= [1 0 0 0
    0 57.296 0 0
    0 0 57.296  0
    0 0 0 57.296
    0 -57.296 57.296 0];
D=zeros(5,2);



% %etape1
%question 1
 simParam.StartTime='0';
 simParam.StopTime='600';
 simParam.SolverType='Fixed-step';
 simParam.Solver='ode3';
 simParam.FixedStep='1e-3';
 simParam.SaveOutput='on';
 simParam.SaveFormat='DataSet';

 step_delta=5;
 step_delta_time=100;
 step_a=0.1;
 step_a_time=100;
 V0=80;
 Z0=700;

 simOut=sim('main_sim',simParam);
 v=simOut.get('v');
 gamma=simOut.get('gamma');
 alpha=simOut.get('alpha');
 q=simOut.get('q');
 z=simOut.get('z');
 theta=simOut.get('theta');

 figure;
 subplot(321);
 plot(V0+v);
 title("V");
 subplot(322);
 plot(alpha);
```

```
title("alpha");
subplot(323);
plot(theta);
title("theta");
subplot(324)
plot(q);
title("q");
subplot(325)
plot(gamma);
title("gamma");
subplot(326)
plot(z);
title("z")

%question 2

%delta>0
step_delta=5;
step_delta_time=100;
step_a=0;
step_a_time=100;
V0=80;
Z0=700;

simOut=sim('main_sim',simParam);
v=simOut.get('v');
gamma=simOut.get('gamma');
alpha=simOut.get('alpha');
q=simOut.get('q');
z=simOut.get('z');
theta=simOut.get('theta');

figure;
subplot(321);
plot(V0+v);
title("V");
subplot(322);
plot(alpha);
title("alpha");
subplot(323);
plot(theta);
title("theta");
subplot(324)
plot(q);
title("q");
subplot(325)
plot(gamma);
title("gamma");
subplot(326)
plot(z);
title("z")

%delta<0
step_delta=-5;
step_delta_time=100;
step_a=0;
step_a_time=100;
V0=80;
```

```
Z0=700;

simOut=sim('main_sim',simParam);
v=simOut.get('v');
gamma=simOut.get('gamma');
alpha=simOut.get('alpha');
q=simOut.get('q');
z=simOut.get('z');
theta=simOut.get('theta');

figure;
subplot(321);
plot(V0+v);
title("V");
subplot(322);
plot(alpha);
title("alpha");
subplot(323);
plot(theta);
title("theta");
subplot(324)
plot(q);
title("q");
subplot(325)
plot(gamma);
title("gamma");
subplot(326)
plot(z);
title("z")

%delta=0
step_delta=0;
step_delta_time=100;
step_a=0.1;
step_a_time=100;
V0=80;
Z0=700;

simOut=sim('main_sim',simParam);
v=simOut.get('v');
gamma=simOut.get('gamma');
alpha=simOut.get('alpha');
q=simOut.get('q');
z=simOut.get('z');
theta=simOut.get('theta');

figure;
subplot(321);
plot(V0+v);
title("V");
subplot(322);
plot(alpha);
title("alpha");
subplot(323);
plot(theta);
title("theta");
subplot(324)
plot(q);
```

```
 title("q");
 subplot(325)
 plot(gamma);
 title("gamma");
 subplot(326)
 plot(z);
 title("z")

%etape 2

%Question 3

 [b,a]=ss2tf(A,B,C,D,2);
 poles=roots(a);
 tf_va = tf(b(1,:),a);

 figure;
 pzmap(tf_va);
 grid on;
 s=tf('s');

 figure;
 fasttf=1/((s-poles(1))*(s-poles(2)));
 damp(fasttf)

 subplot(221);
 pzmap(fasttf);

 subplot(222);
 step(fasttf/0.02964);

 phutf=0.0471772291240761/((s-poles(3))*(s-poles(4)));
 damp(phutf);

 subplot(223);
 pzmap(phutf);

 subplot(224)
 step(phutf);

% Question4
 zeros=roots(b(1,:));
 figure;
 subplot(221);
 bode(phutf,tf_va);
 title('Bode rapide sans zeros');

 subplot(222)
 bode((s-zeros(3))*phutf,tf_va);
 title('Bode rapide un zeros reel');

 subplot(2,2,3:4);
 bode((s-zeros(1))*(s-zeros(2))*phutf,tf_va);
 title('Bode rapide deux zeros complexes conjugues');

 figure;
 subplot(221);
 step(phutf,tf_va/31.045);
```

```
 title('Step rapide sans zeros');

 subplot(222)
 step((s-zeros(3))*phutf/.645,tf_va/31.045);
 title('Step rapide un zeros reel');

 subplot(2,2,3:4);
 step((s-zeros(1))*(s-zeros(2))*phutf/63.874,tf_va/31.045);
 title('Step rapide deux zeros complexes conjugues');

%etape3
%question 5

 simParam.StartTime='50';
 simParam.StopTime='200';
 simParam.SolverType='Fixed-step';
 simParam.Solver='ode3';
 simParam.FixedStep='1e-3';
 simParam.SaveOutput='on';
 simParam.SaveFormat='DataSet';

 step_delta=5;
 step_delta_time=100;
 V0=80;
 Z0=700;
Kv=2;

 simOut=sim('main_sim',simParam);
 v=simOut.get('v');
 gamma=simOut.get('gamma');
 alpha=simOut.get('alpha');
 q=simOut.get('q');
 z=simOut.get('z');
 theta=simOut.get('theta');

 figure;
 subplot(321);
 plot(V0+v);
 title("V");
 subplot(322);
 plot(alpha);
 title("alpha");
 subplot(323);
 plot(theta);
 title("theta");
 subplot(324)
 plot(q);
 title("q");
 subplot(325)
 plot(gamma);
 title("gamma");
 subplot(326)
 plot(z);
 title("z")

%Question 6
 figure;
 subplot(121);
```

```
 k=00.001;
 p=[i i i i];

 %On plot les poles et les zeros
 plot(poles,'x');
 hold on;
 plot(zeros,'o');
 hold on;
 title('Manual Root Locus')

 while(imag(p(1))~=0 && imag(p(2))~=0 && imag(p(3))~=0 && imag(p(4))~=0)
     ka=feedback(tf_va,k);
     p=pole(ka);

     plot(real(p),imag(p),'.');
     hold on;
     k=k+.01;
  end


 subplot(122);
 rlocus(tf_va);

%Question 7

tfphu=(s-zeros(3))*phutf;
tfphu=tfphu/dcgain(tfphu);

 figure;
 for (Kv = 1:.1:50)
     st=stepinfo(feedback(tfphu,Kv));
     t=st.SettlingTime;
     plot(Kv,t,'.');
     hold on;
 end
 xlabel('Kv');
 ylabel('Settling Time');
 title('Evolution du temps de reponse en fonction de Kv');

%On deduit que le temps d'etablissement diminue avec l'augmentation de Kv.
%En revanche plus Kv est plus, plus le risque de saturer les actionneurs est grand.

%Question 8

%Avec  sisotool et avec un objectif de z=0.95 on trouve
Kv_ordre2=32.235/dcgain(phutf)

 figure;
 step(feedback(tfphu,Kv));

%Question 9
%Ordre 4
Kv_ordre4=dcgain(tf_va)/.94029;

%Les valeurs de Kv trouvees sont tres proches donc notre apporximation est
pertinente.

%Question 10
```

```
%Avec sisotool et un objectif d'une marge de phase de 40degres
KV@40degres=[.107 3.37 5.33]
plus rapide = 5.33
Kv_phase_margin=5.3346;

%Question 11

simParam.StartTime='0';
simParam.StopTime='200';
simParam.SolverType='Fixed-step';
simParam.Solver='ode3';
simParam.FixedStep='1e-2';
simParam.SaveOutput='on';
simParam.SaveFormat='DataSet';

step_v=5;
step_v_time=50;
V0=80;
Z0=700;
K1=.94029;

simOut=sim('main_sim',simParam);
v=simOut.get('v');
gamma=simOut.get('gamma');
alpha=simOut.get('alpha');
q=simOut.get('q');
z=simOut.get('z');
Comm=simOut.get('Commande');
theta=simOut.get('theta');

figure;
subplot(211);
plot(V0+v);
title("V");
subplot(212);
plot(Comm);
title("Commande");

figure;
subplot(321);
plot(V0+v);
title("V");
subplot(322);
plot(alpha);
title("alpha");
subplot(323);
plot(theta);
title("theta");
subplot(324)
plot(q);
title("q");
subplot(325)
plot(gamma);
title("gamma");
subplot(326)
plot(z);
title("z")
```

```
%Question12
 figure;
 K=linspace(0,5,100);
 J=linspace(0,5,size(K,2));
 for r=[0.2 0.5 1]
     for index=1:size(K,2)
         K1=K(index);
         simOut=sim('main_sim',simParam);
         v=simOut.get('v');
         J(index)=sum(((v.data-v.data(size(v.data,1))).^2)+r*r*step_v*step_v);

     end
  plot(K,J);
  hold on;
 end

 legend('r=0.2','r=0.5','r=1');

%fonction decroissante
 KV_ideal=5;

 figure;
 Norm_ideal=feedback(tfphu,KV_ideal);
 Norm_ideal=Norm_ideal/dcgain(Norm_ideal);

 step(Norm_ideal);
```

A.2 Figures

Table des figures