

Compte rendu du rapport de TP 1 CAVAN

Commandes Optimales Linéaires LQ/LQG

Nom	Professeur
Housseayne NADOUR	Bogdan MARINESCU
Pierre BLAUD	Mohamed BELHOCINE

1 Introduction

Ce document est le compte rendu du TP de CAVAN réalisé le lundi 11 décembre 2017 à l'Ecole Centrale de Nantes. Le TP traite de la commande LQ et LQG pour stabiliser un modèle de quart de véhicule. La première partie traitera de la commande LQR (Linear Quadratic Regulator) ensuite la seconde partie traitera du filtre de Kalman aussi appelé LQE (Linear Quadratic Estimator) et finalement la dernière partie traitera de la synthèse des deux avec la commande LQG (Linear Quadratic Gaussian).

2 Étude du système

A) simulation en boucle ouverte :

Dans un premier temps le système dynamique est simulé est constitué sur Matlab et sa réponse avec une commande u égale à zéros et les conditions initiales suivantes : $x_0 = [0.0022; -0.1146; 0.0421; -0.0183]$ est simulée. Le code Matlab est :

```
%% declaration des parametres du syst me

M1 = 30; % kg
M2 = 250; %kg
k = 2e4; %N/m
k1 = 2e5; % N/m
f = 1e3; %N.S/m

%% representation d etat du systeme

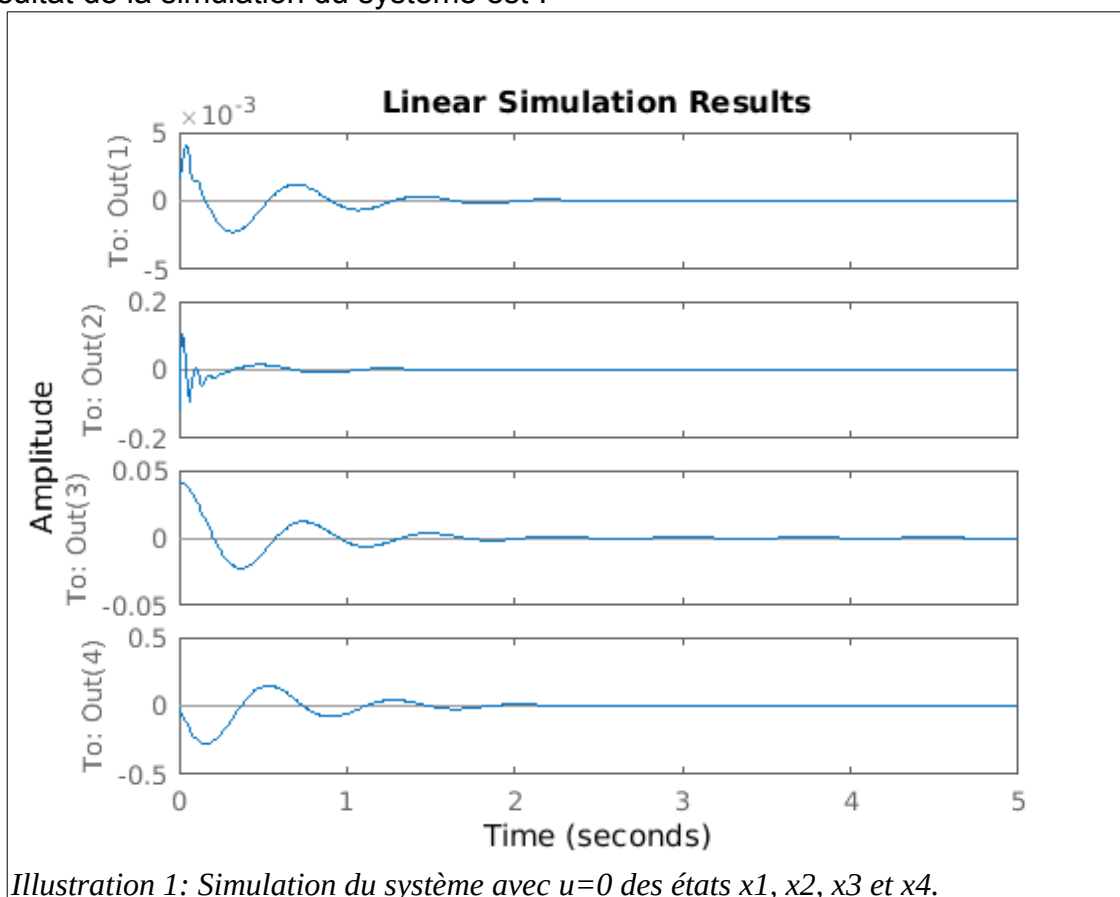
A = [ 0, 1, 0, 0, ; (- ( k1 + k) / M1), (-f/M1), k/M1, f/M1 ; 0, 0, 0, 1, ;
      k/M2, f/M2, (-k/M2), (-f/M2) ];
% Declaration de la matrice d etat A
B = [0; -1/M1; 0 ; 1/M2];
% Declaration de la matrice d etat B
Ci = eye(4);
% Declaration de la matrice Identity pour obtenir l etat X
D = [0;0;0;0];
% La matrice D est nul
```

```

%% Question A : simulation du comportement du systeme
sys_A = ss (A,B,Ci,D);
    % creation de la fonction de transfert du systeme
x0 = [0.0022; - 0.1146; 0.0421; -0.0183];
    % conditions initiales
% creation du vecteur de temps pour la fonction lsim
dt = 0.01; % step size
Tfinal = 5; % temps finale de simulation
t = 0:dt:Tfinal; % creation du vecteur temps
u = zeros ( size (t)); % la commande u est mise a zero pour toute la dur  e
% Simulation du comportement avec la commande lsim
figure(1),lsim(sys_A,u,t,x0);

```

Le r  sultat de la simulation du syst  me est :



Pour obtenir les   tats du syst  mes $x_1(t)$, $x_2(t)$, $x_3(t)$ et $x_4(t)$ la matrice C de la repr  sentation d'  tat est prise   gale    la matrice Identit  .

L'équation qui permet d'obtenir la sortie de la suspension y_s est : $z_1(t) - z_2(t)$. Le script Matlab devient avec la modification de la matrice C, appelée C2 :

```
% Question B : nous souhaitons maintenant la sortie ys avec le debatement de la
suspension
% la sortie ys est : ys = z2 - z1;

C2 = [-1 , 0 , 1, 0]; % la matrice C est modifiée
D = 0; % la matrice D est égale à 0
sys_B = ss (A,B,C2,D); % la fonction de transfert du système est créée avec la
nouvelle matrice C pour obtenir ys en sortie
% Simulation du comportement de ys avec la commande lsim
figure(2), lsim(sys_B, u,t, x0);
```

La simulation du système avec la sortie $y_s(t)$ est :

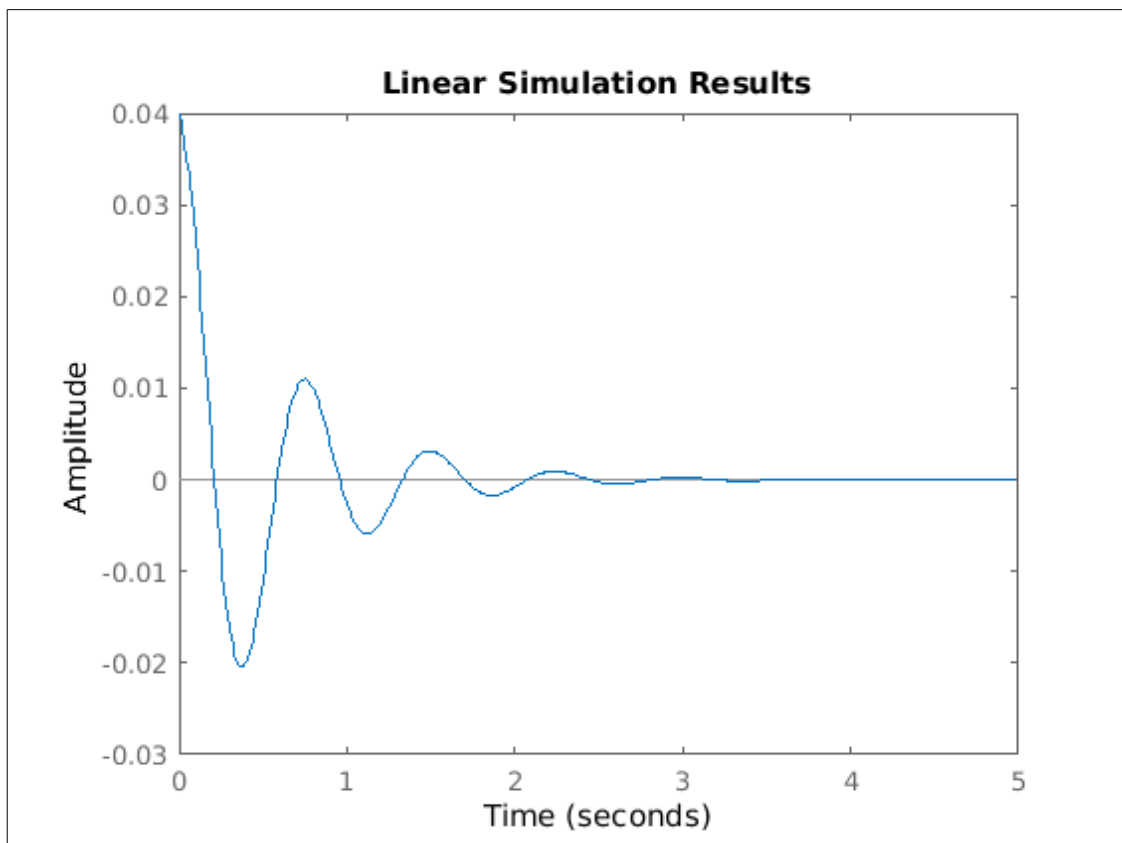


Illustration 2: Sortie $y_s(t)$

B) La sortie à commander :

$$y_s = x_3 - x_1 = C \cdot X \quad \text{tel que : } C = [-1, 0, 1, 0]$$

C) La commandabilité :

Avant de vérifier tout tentative de contrôle il faut vérifier que le système est au moins stabilisable. Pour cela le script Matlab est modifié pour calcul le rang de la matrice de contrôlabilité.

```
%% Question C: test de la controlabilite du systeme
Co = ctrb (sys_B); % la matrice de controlabilite est cree
unco = length(A) - rank(Co); % on test que le rang de la matrice de
controlabilite est egale a la dimension de la matrice A
```

Le rang de la matrice de controllabilité est égale à la dimension de A. Le système est commandable par une commande.

3 Synthèse Optimale LQ

1) L'objectif du crétere

Ce critère représente une somme pondération de l'énergie de X et U, alors l'optimisation d'un tel critère signifie que l'obtention une commande pour la quelle l'anergie fournie est minimale, et ça entraîne automatiquement que la réponse va être rapide aussi.

2) Conditions sur R et Q :

R et Q doivent etre semi defini positive parce que la puissance est toujours positive.

3,4, et 5) Calcule de K et simulation:

Le passage de l'état initiale vers l'état finale désirée du système est possible car il est contrôlable, l'introduction de J permet de réaliser ce passage avec la minimisation d'un critère appelé le coût. Le critère J choisi est la minimisation de l'énergie ys. Les critères Q et R doivent être inversibles (pas égale à zéro), car dans l'équation de Riccati, il y a un R^{-1} . Un script Matlab est utilisé pour calculé le gain du LQR avec la commande Matlab lqr(). Plusieurs critère de pondération de R sont utilisés :

```
%% Question 4.1 : synthese optimale LQR LQR (linear quadratic regulator)
% Nous utilisons la commande lqr de Matlab pour obtenir le gain K du LQR

Q = 1; % premiere ponderation Q = 1
Qx = C2' * Q * C2; % le Qx est calcule
R1 = 1; % seconde ponderation R = 1
N = 0; % est egale a zero, le critere a cout croise n est pas utilise

K1 = lqr(A,B,Qx,R1,N) ;
% le gain K1 du LQR est calcule par la commande Matlab LQR
```

```

% calcul de la boucle ferme avec le gain K1 du LQR
A1 = A - B * K1; % calcul Abf
sys1 = ss (A1,B,C2,D); % calcul de la nouvelle fonction de transfert avec Abf
avec le gain K1 du LQR

%% Question 6 : Calcul des gains du LQR avec R variable et Q = 1

    Q = 1; % la pondération Q est commune
    Qx = C2' * Q * C2; % la pondération Qx est commune
    N = 0; % est commun egale a zero

% gain du LQR avec R = 10^-4 appele K2
    R2 = 10^-4; % pondération R egale a 10^-4
    K2 = lqr(A,B,Qx,R2,N) ; % calcul du gain K2 avec LQR
    A2 = A - B * K2; % calcul de la nouvelle matrice Abf
sys2 = ss (A2,B,C2,D); % calcul de la fonction de transfert avec le gain K2

% gain du LQR avec R = 10^-8 appele K3
    R3 = 10^-8; % pondération R = 10^-8
    K3 = lqr(A,B,Qx,R3,N) ; % calcul du gain K3 avec LQR
    A3 = A - B * K3; % calcul de la nouvelle matrice Abf
sys3 = ss (A3,B,C2,D); % calcul de la fonction de transfert avec le gain k3

% gain du LQR avec R = 10^-10 appele K4
    R4 = 10^-10; % pondération R = 10^-10
    K4 = lqr(A,B,Qx,R4,N) ; % calcul du gain K4 avec LQR
    A4 = A - B * K4; % calcul de la nouvelle matrice Abf
sys4 = ss (A4,B,C2,D); % calcul de la fonction de transfert avec le gain K4

% tracer les reponses avec les R modifies pour R = 1, R = 10^-4, R = 10^-8
% et R = 10^-10

figure(3),lsim(sys_B,sys1,sys2,sys3,sys4, u,t,x0);
grid on
legend('show');

```

Par ailleurs,le critère de pondération Q_x utilisé dans le calcul du gain lqr . Pour cela le critère de pondération Q_x est calculé par la commande $Q_x = C2' * Q * C2$. Les différents gains K du LQR pour les différents valeurs de pondération sont :

Pondération R	Pondération Q	Gain du LQR
1	1	$K1 = [-0,25e-4 \ 0 \ 0,25e-4 \ 0,0625e-4]$
10^{-4}	1	$K2 = [-0,25 \ 0 \ 0,25 \ 0,0625]$
10^{-8}	1	$K3 = [-1,9242e+3 \ -0,014e+3 \ 2,3607e+3 \ 0,4683e+3]$
10^{-10}	1	$K4 = [-4,1771e+4 \ -0,0410e+4 \ 8,198e+4 \ 0,4634e+4]$

La simulation du système est réalisée avec le retour de gain du LQR pour les valeurs de gain K listées dans le tableau ci-dessus.

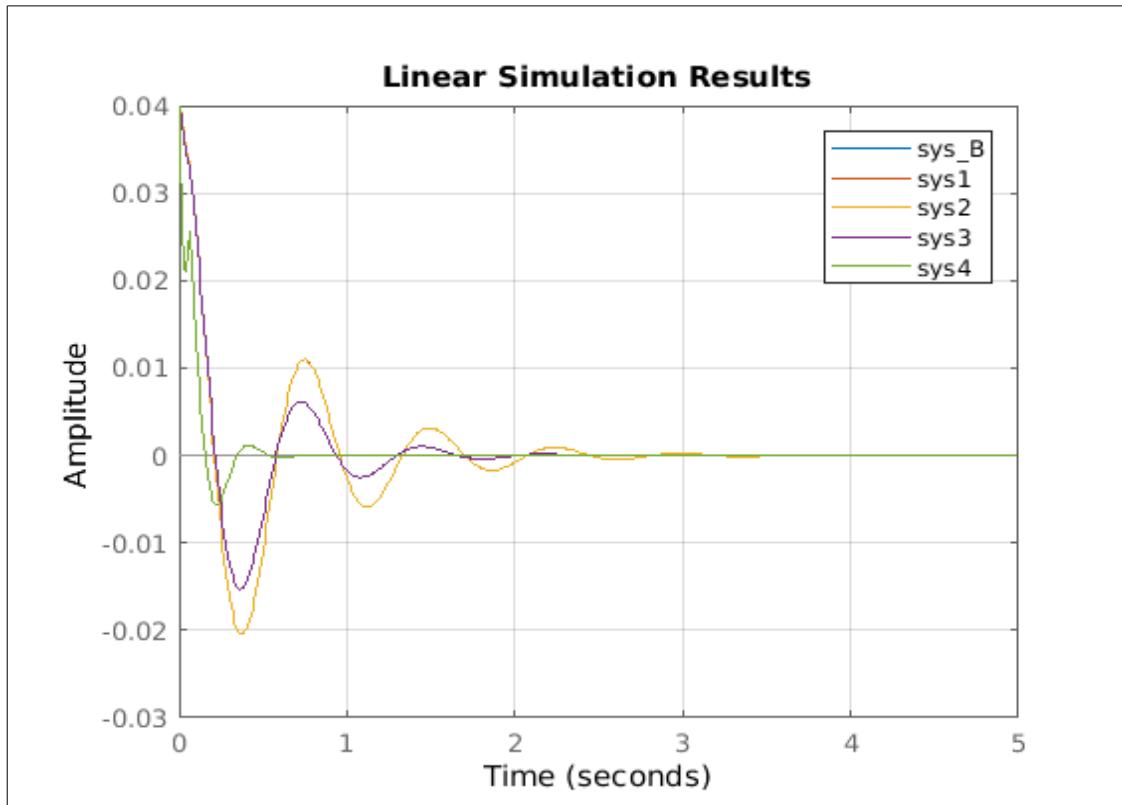


Illustration 3: Simulation du système avec le retour de gain K pour différentes valeurs du critère de pondération R .

Pour obtenir l'évolution temporelle de la commande u du système, une simulation est réalisée avec Simulink :

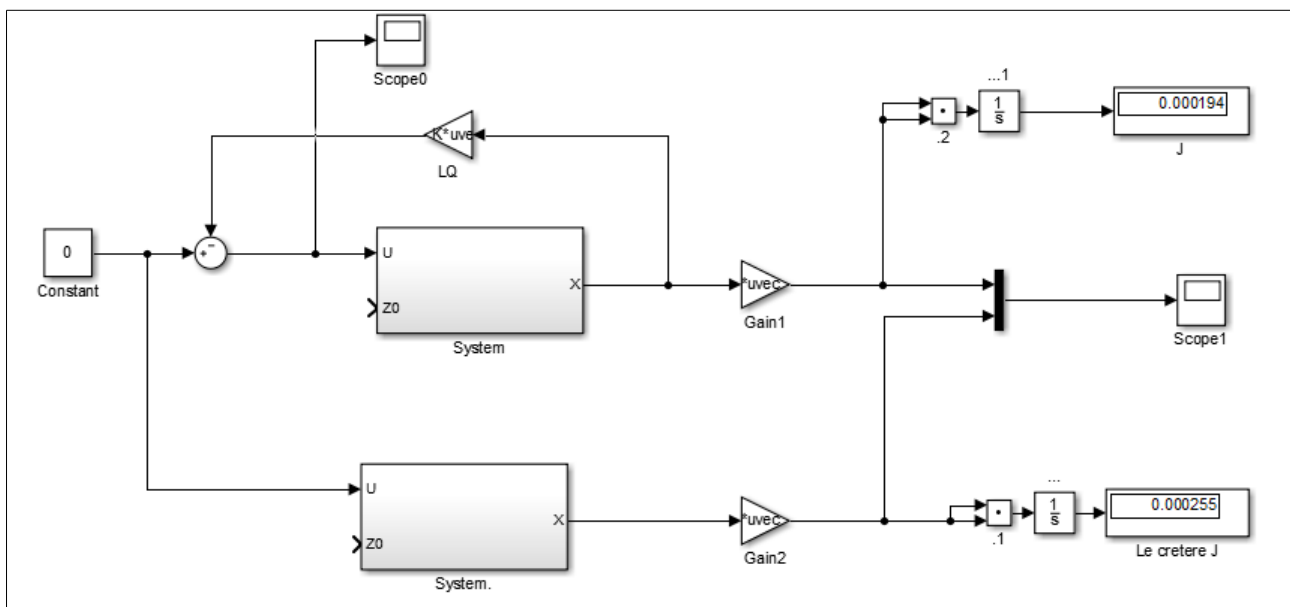


Illustration 4: Simulation simulink avec LQR et BO.

6) Les différentes évolutions temporelles de $u(t)$ selon plusieurs critères de pondération R sont :

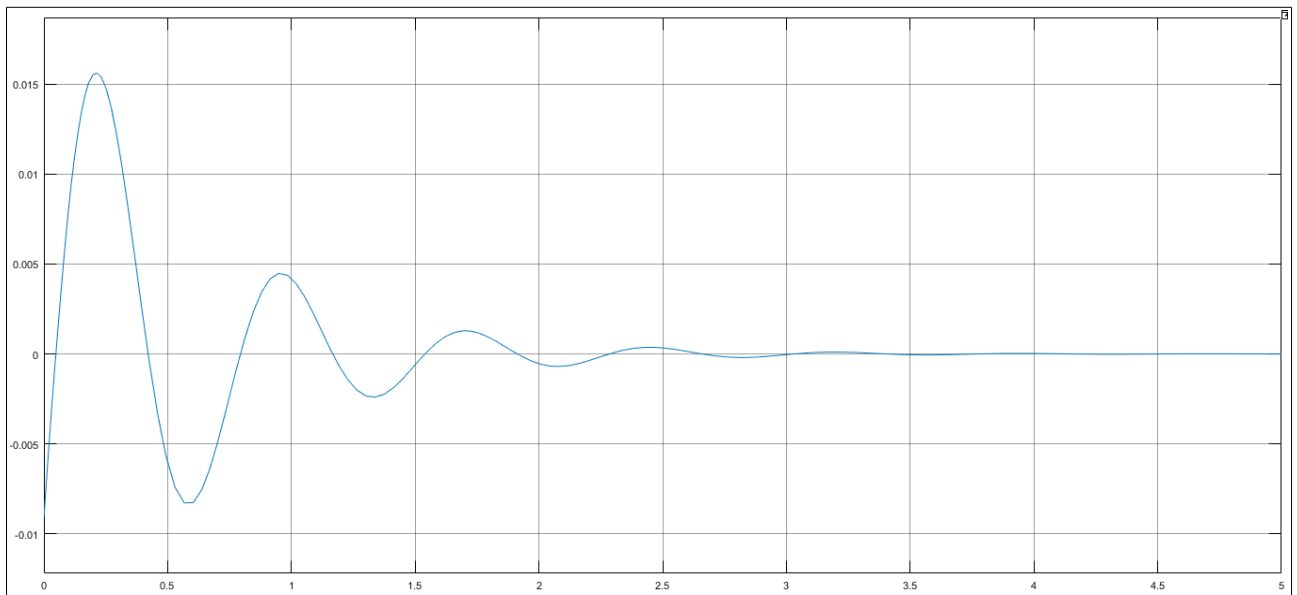


Illustration 5: Commande $u(t)$ pour critère de pondération $R=10^{-4}$

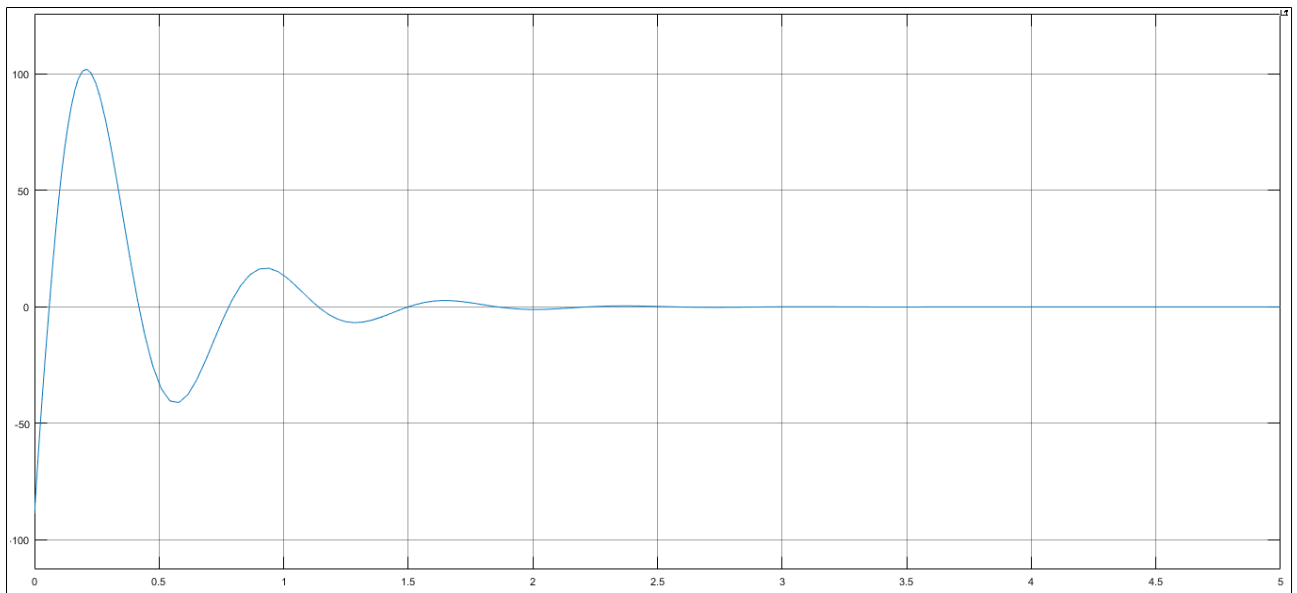


Illustration 6: Commande $u(t)$ pour un critère de pondération $R=10^{-8}$.

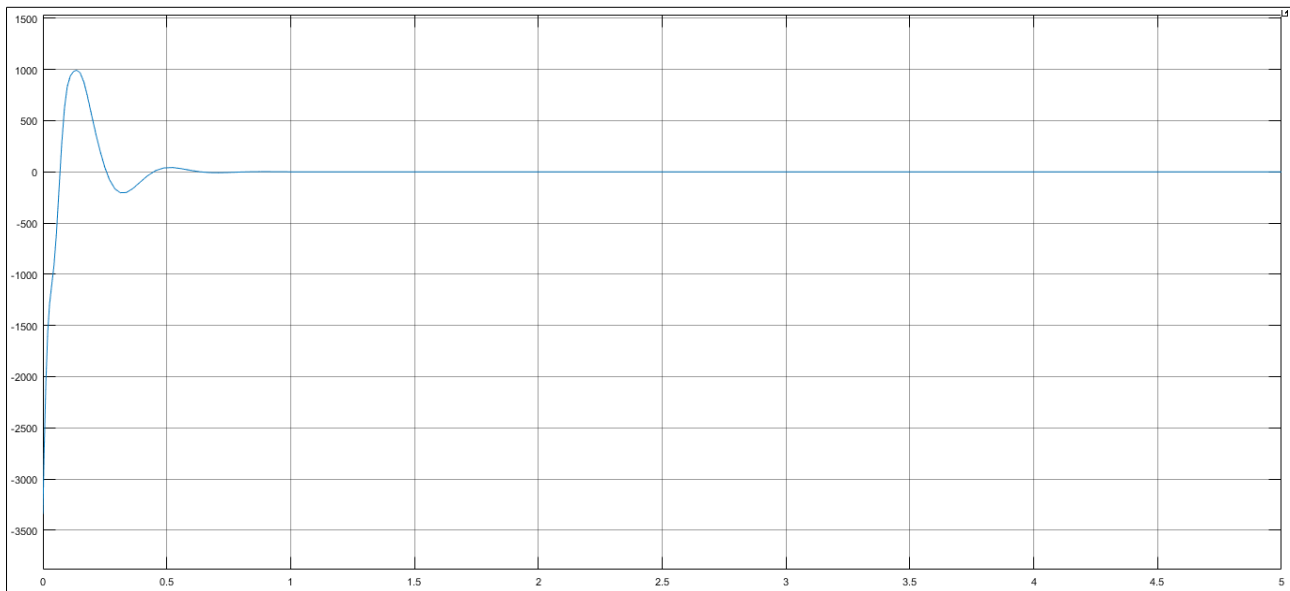


Illustration 7: Commande $u(t)$ pour un critère de pondération $R=10^{-10}$.

En observant l'évolution temporelle de $u(t)$ en fonction des critères de pondérations R modifiés, nous remarquons que $u(t)$ (aussi le gain K) augmente quand le critère R baisse. Par ailleurs, lorsque le critère de pondération R baisse, la sortie $y_s(t)$ du système atteint plus rapidement l'état finale en un temps plus court.

7) Performance dynamique selon R et Q :

La valeur du critère de pondération R permettant d'obtenir une commande $u(t)$ maximale de 102N est de 10^{-8} . Le critère de pondération Q est modifié avec 10 et R de 10^{-8} . Le nouveau script Matlab est :

```
% gain du LQR avec R = 10^-8 et Q = 10 appele K5
Q2 = 10; % la pondération Q est commune
Qx2 = C2' * Q2 * C2; % la pondération Qx est commune
R5 = 10^-8; % pondération R = 10^-10
K5 = lqr(A,B,Qx2,R5,N) ; % calcul du gain K4 avec LQR
A5 = A - B * K5; % calcul de la nouvelle matrice Abf
sys5 = ss (A5,B,C2,D); % calcul de la fonction de transfert avec le gain K4

figure(4),lsim(sys5, u,t,x0);
grid on
legend('show');
```

Le gain K du LQR obtenu est : $K5 = [-0,9583e+4 \ -0,0039e+4 \ 1,7417e+4 \ 0,1996e+4]$ La simulation du système avec le critère de pondération $Q = 10$ et $R=10^{-8}$ est :

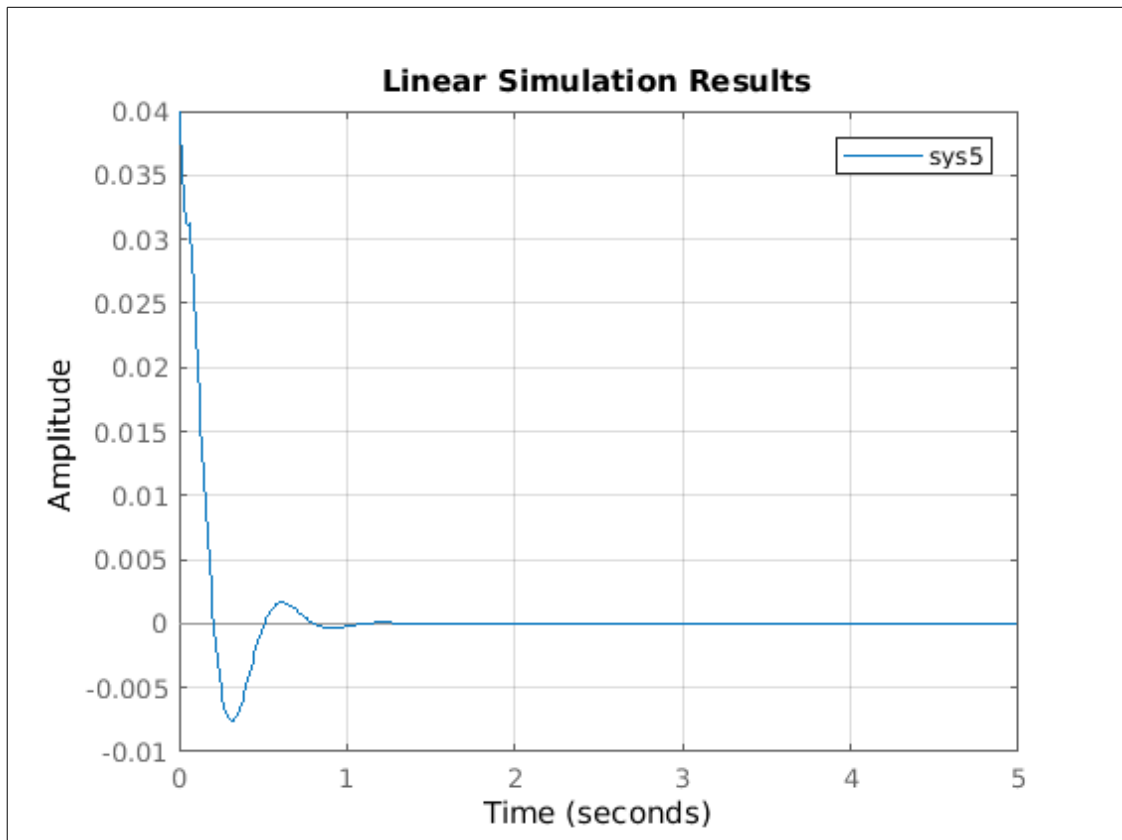


Illustration 8: Simulation du système avec le LQR, avec critère de pondération $Q=10$ et $R=10e-8$.

8 et 9) les pondérations R et Q :

L'ajout du critère de $Q = 10$ permet d'obtenir un gain du LQR proche du gain du LQR avec $Q=1$ et $R=10^{-10}$. Le critère de pondération $Q=10$ et $R=10^{-8}$ est égale aux critères de pondération $Q=1$ et $R=10^{-9}$ pour obtenir le même critère d'optimalité. Pour baisser le temps de réponse du système avec la fonction de coût J : soit on baisse le critère de pondération R , ou bien, on augmente le critère de pondération Q . Ainsi l'importance de la pondération du critère de coût J est le rapport entre Q et R . Cependant la commande $u(t)$ dépasse la valeur des 100N. Pour respecter la contrainte de $u(t) < 102N$, il faut respecter le rapport $Q/R = 10^8$. Ainsi pour un critère de pondération de $Q=10$, il faut un $R=10^{-7}$ pour respecter la contrainte de $u(t) < 102N$.

10) Minimisation le l'accélération :

Si l'on souhaite en plus minimiser le critère d'accélération de la caisse, il est nécessaire d'introduire un critère de coût croisé de la forme :

$$J = \int_0^{\infty} y^T(t) Q_1 y(t) + y_{mes}^T \cdot Q_2 y_{mes} + u^T(t) R_u u(t) dt$$

tel que $y_{mes} = C_{mes} X + D_{mes} U$ et $y = CX$

Où C_{mes} représente la dernière ligne de la matrice A et $D_{mes} = 1/M^2$

En substituant ces valeurs en J on obtient :

$$J = \int_0^{\infty} x^T \cdot Q \cdot x + 2 \cdot x^T \cdot N \cdot u + u^T R_u \cdot u \cdot dt$$

tel que :

$$Q = C^T \cdot Q_1 \cdot C + C_{mes} \cdot Q_2 \cdot C_{mes}, \quad R = D_{mes}^T \cdot Q_2 \cdot D_{mes} + R_u, \quad \text{et} \quad N = C_{mes}^T \cdot Q_2 \cdot D_{mes}$$

Avec la commande Matlab lqr(), le critère de pondération N doit être modifié.

4 Estimateur d'état optimal : filtre de Kalman

a) L'observabilité :

Un estimateur d'état peut être introduit si le système est observable. Pour tester l'observabilité du système un script Matlab est utilisé pour calculer la matrice d'observabilité du système et de calculer son rang.

Pour une sortie mesurée : $y_{mes} = \dot{x}_4 = C_{ac} \cdot X + D_{ac} \cdot U + v$ tel que C_{ac} représente la dernière ligne de la matrice A, $D_{ac} = 1/M^2$, et v est le bruit de mesure.

```
% Question 4.2 : estimateur d etat optimal filtre de Kalman LQE (linear
quadratic estimator)

% Question a : verification de l observabilite du systeme
Cac = A(4,:); % Le ymes est modifie pour egale z'' + v(t)
Ob = obsv(A,Cac); % construction de la matrice observabilite
unob = length(A)-rank(Ob); % test du rang de la matrice Obs est egale a la
dimension de la matrice A
```

Le rang de la matrice d'observabilité est égale à la dimension de la matrice A, le système est observable et un reconstruteur peut être introduit.

b) Les gains du filtre :

Un script Matlab est utilisé pour calculer le gain de l'observateur de Kalman avec dans un premier temps une espérance mathématique $\rho_0 = \rho_1 = 0,01$ puis dans un deuxième temps $\rho_0 = 1$ et $\rho_1 = 0,01$:

```
% Question B : calcul du gain du filtre de Kalman LQE : Kf
```

```

% premier cas esperance est egale a 0.01
r0 = 0.01; % ponderation egale a esperance mathematique
r1 = 0.01; % ponderation agale a esperance mathematique
Nn = 0; % Nn est egale a zero
Bd = [0; k1/M1; 0 ; 0]; % le gain du bruit de commande Z0.
C3 = A(4,:); % la nouvelle matrice C est egale a la ligne 4 de la matrice A
D = 1/M2 ;
sys6 = ss(A,B,C3,D); % la nouvelle fonction de transfert du systeme est calcule
[kest,kf1,P,M,Z] = kalman(sys6,r0,r1,Nn); % le gain optimale du filtre de
kalman Kf est calcule avec la commande Kalman
Qlqe = Qn*eye(4);
[L1,Plqe,Elqe] = lqe(A,Qlqe,C3,Qlqe,Rn); %le gain optimale du filtre de Kalman
L1 est calcule avec la commande Lqe

% deuxieme cas esperance est egale a 1, la ponderation Q est modifiee
Qn2 = 1; % ponderation egale a esperance mathematique
[kest2, kf2, P2, M2, Z2 ] = kalman(sys6, Qn2, Rn, Nn); % le gain optimale du
filtre de kalman kf est calcule
Qlqe2 = Qn2 * eye(4);
[L2,Plqe2,Elqe2] = lqe(A,Qlqe,C3,Qlqe,Rn); %le gain optimale du filtre de
Kalman L1 est calcule avec la commande Lqe

%Comparaison des poles
P1 = eig(A - kf1 * C3);
P3 = eig(A - kf2 * C3);

```

Les gains des filtres de Kalman sont :

Qn et Rn	Gain du filtre de Kalman
Qn = 0,01, Rn = 0,01	$Kf1 = 10^3 \begin{pmatrix} -0.0012 \\ 6.6577 \\ -0.0014 \\ -0.0000 \end{pmatrix}$
Qn = 1, Rn = 0,01	$Kf2 = 10^4 \begin{pmatrix} -2.6667 \\ -0.0002 \\ -0.0000 \\ -0.0000 \end{pmatrix}$

Le schéma Simulink pour observer les états.

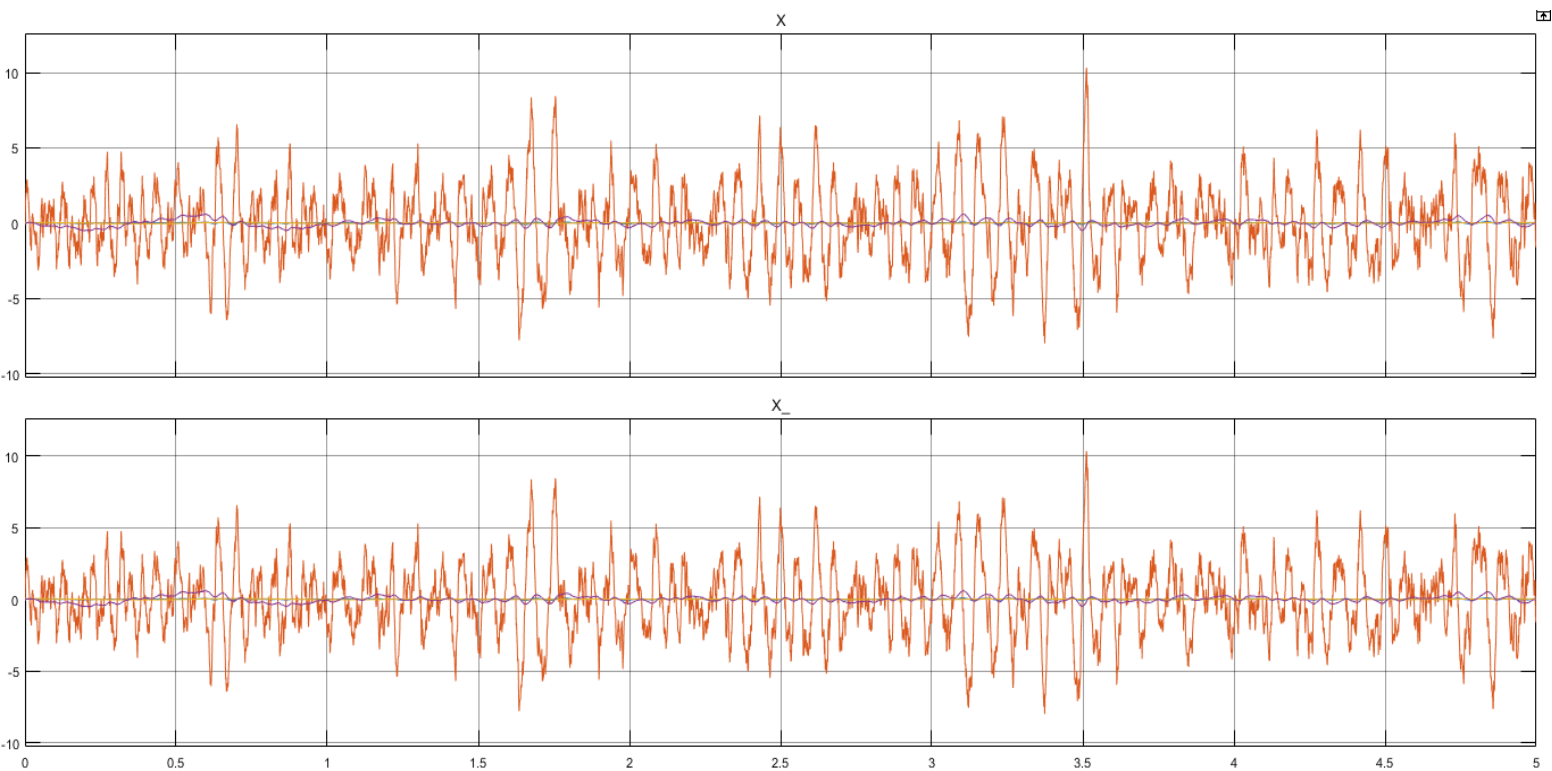


Illustration 10: Les états estimés et les vrais états du système.

La sortie réelle à commander et celle estimée sont représentées sur le graphe suivant :

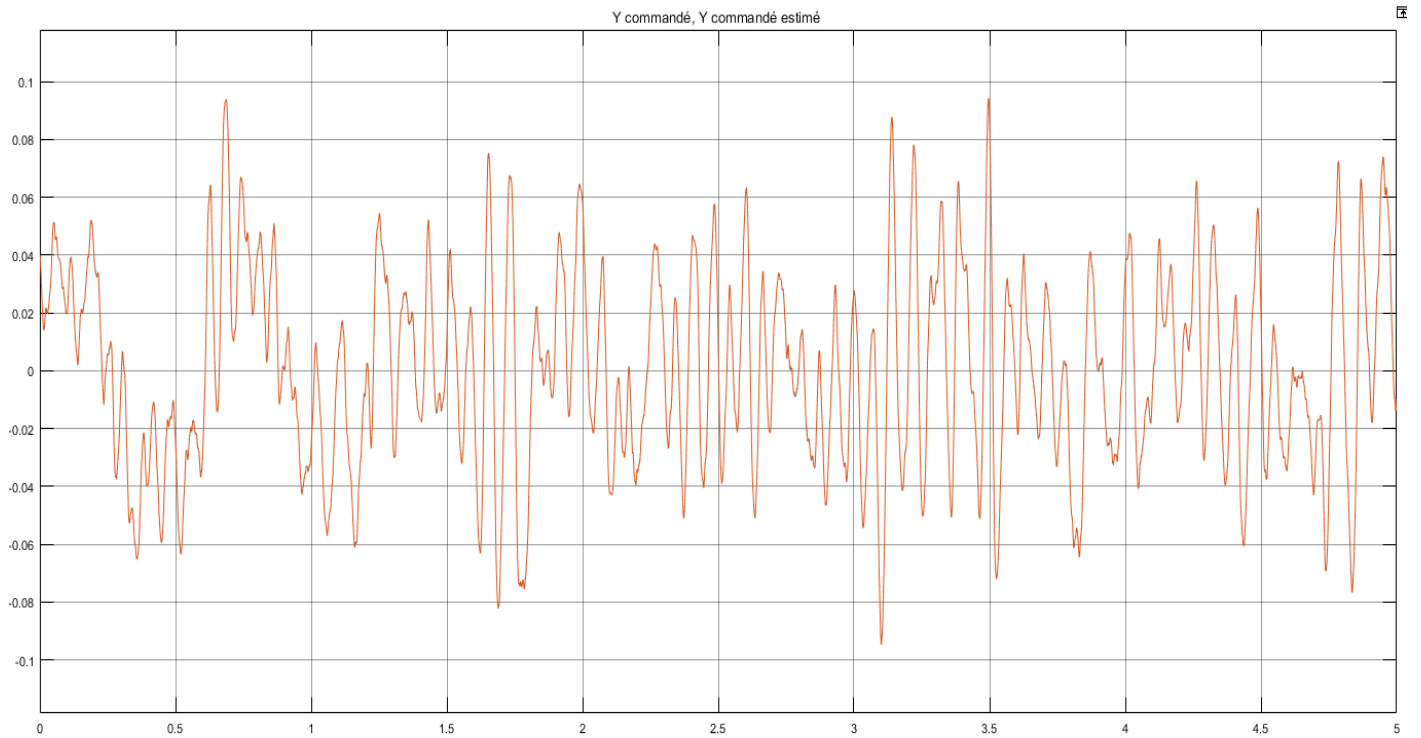


Illustration 11: Y et Y estimée

On voit clairement que la sortie à commander est confondue sur celle réelle.

Illustration sous forme la différence entre X et \hat{X} , Y et \hat{Y}

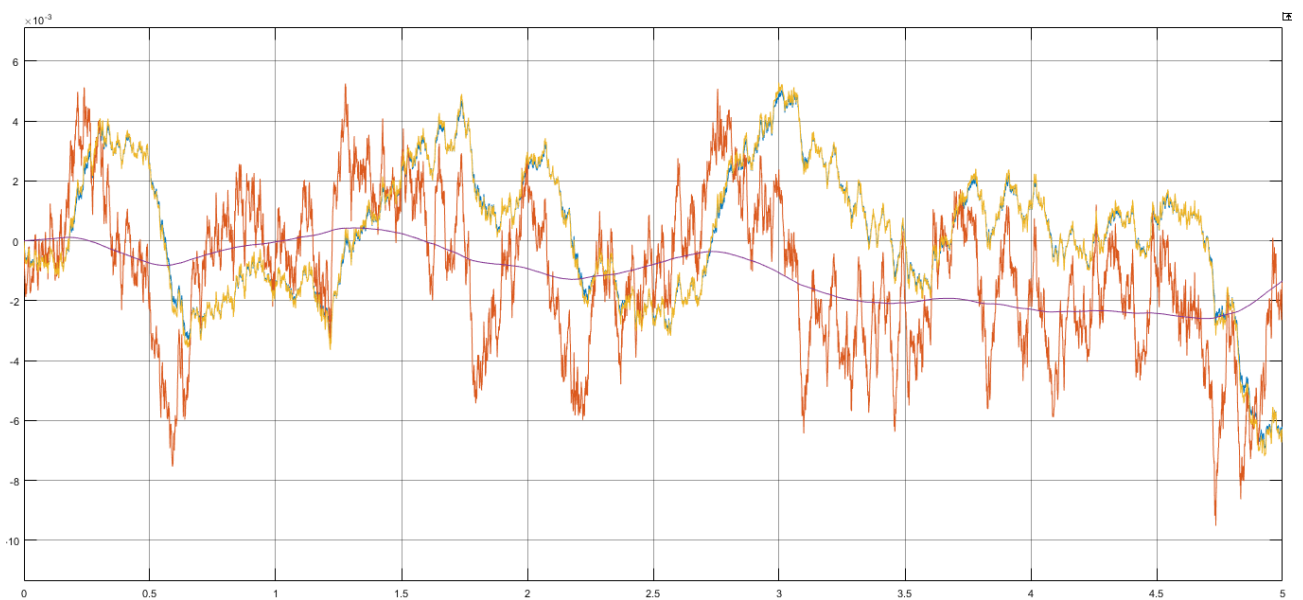


Illustration 12: La différence entre les vrais états X et celle estimées

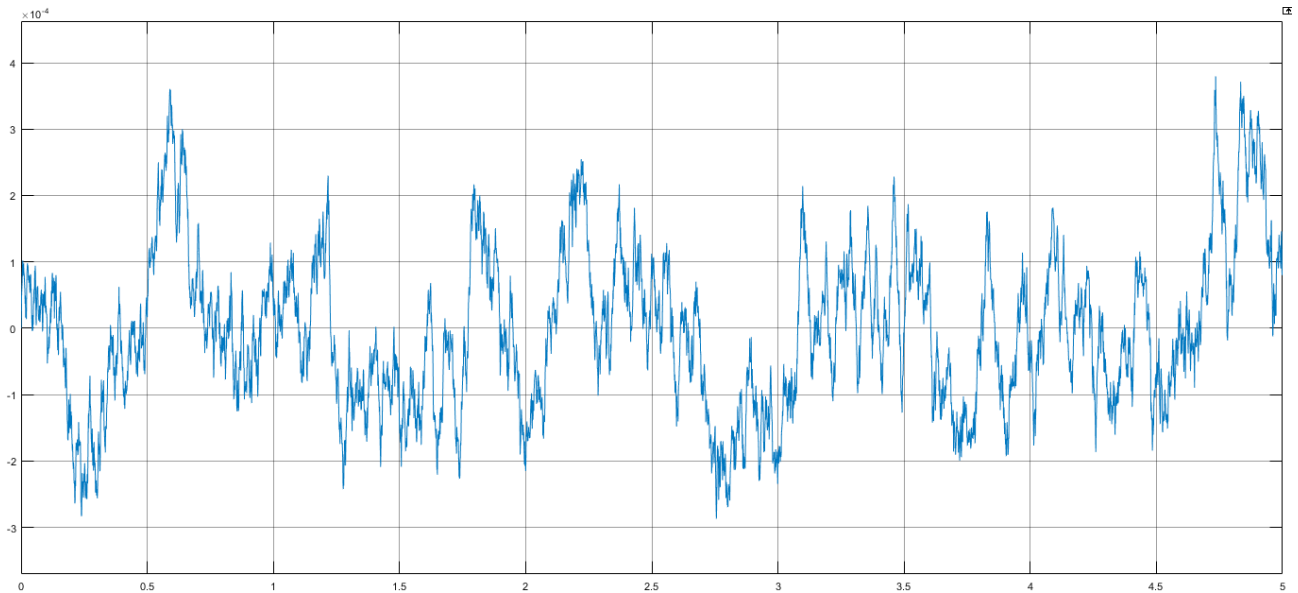


Illustration 13: $Y - \hat{Y}$

Nous remarquons que l'erreur est d'ordre 10^{-4} , qui signifie pratiquement que l'erreur est nulle, d'où la performance du filtre en observant vrais états malgré le bruit de commande et les bruits de mesure.

d) Changement de l'état initiale de l'observateur :

Les résultats des estimateurs de l'état, de la sortie Y estimée, du bruit blanc utilisé, les états du système avec un état initiale de :

$$\hat{x}_0 = x_0 - e_0 = [0,0022; -0,1145; 0,0421; -0,0183] - [0,3; 1; 0,9; 1]^T$$

Le gain de Kalman $kf1$ et $\rho_0 = 0,01$ sont :

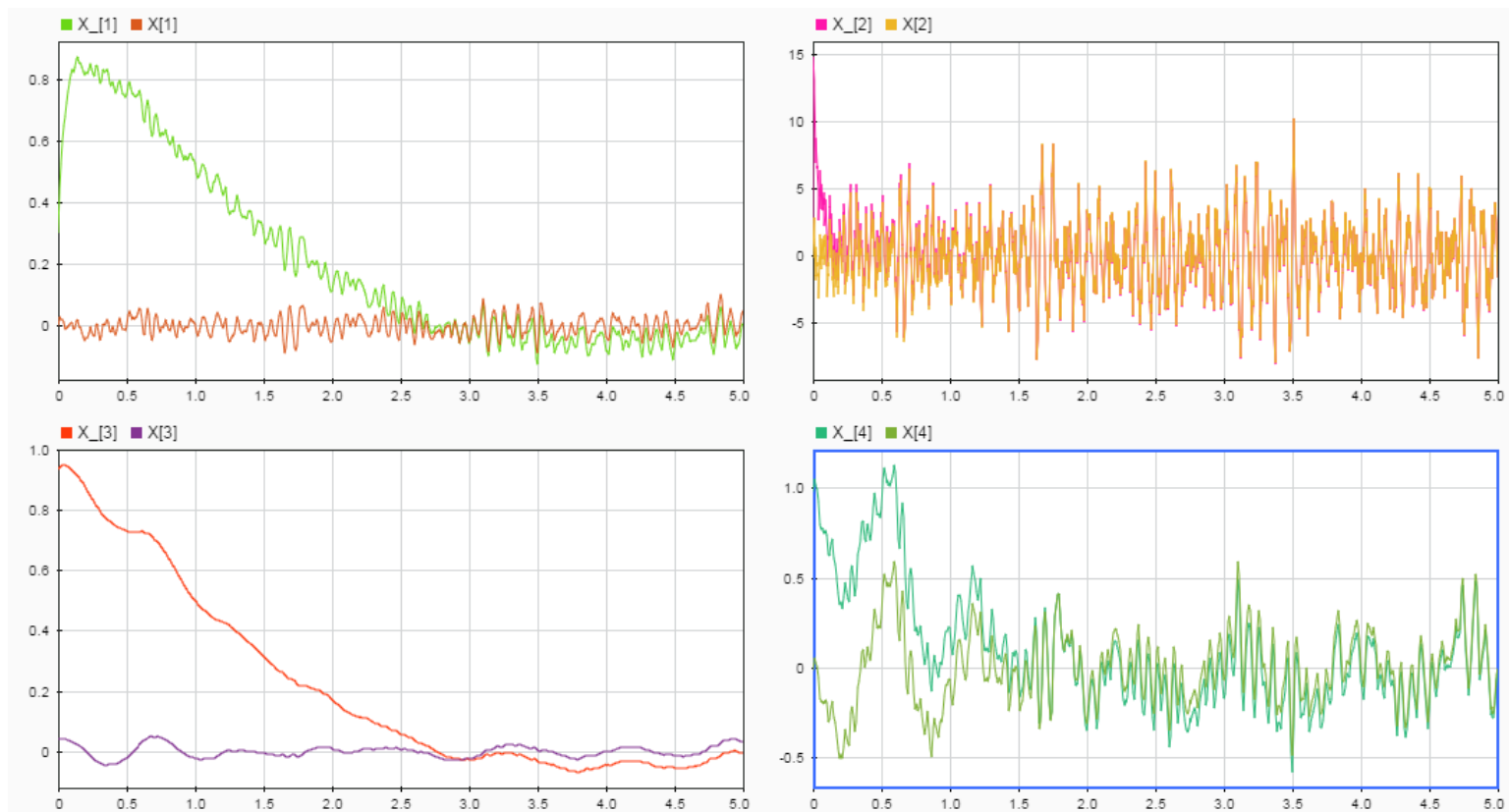


Illustration 14: Les états réelles et les états estimés

On remarque cette fois ci que l'observateur a rattrapé les états réels après environs 2.5 seconds, ce retard est le cout de ne pas savoir la valeur initiale.

On visualise aussi la difference entre X et $X_{[]}$ estimés :

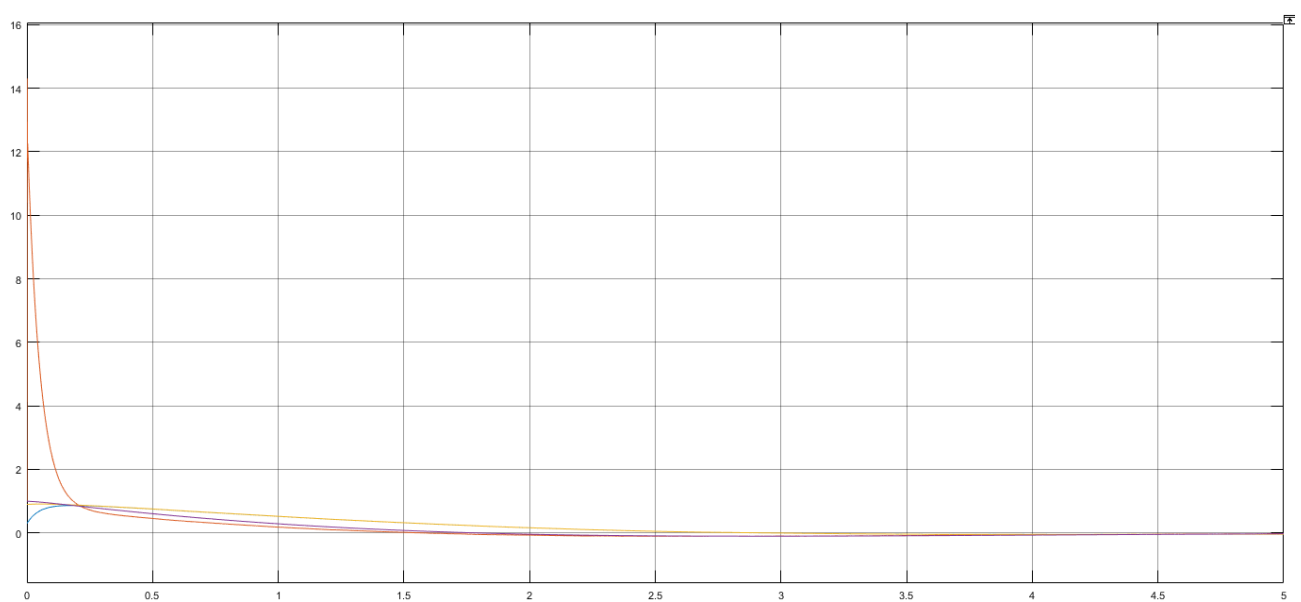


Illustration 15: la différence $e = X - X_{[]}$

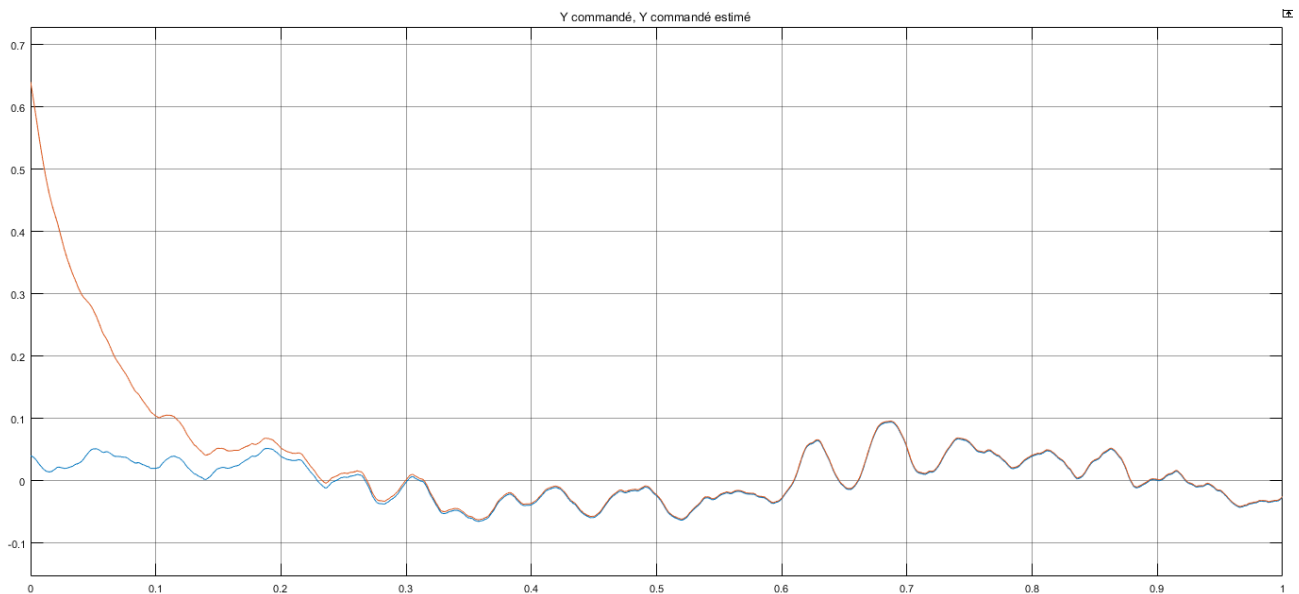


Illustration 16: La sortie à commander Y et La sortie Y_éstimer

Le filtre de Kalman a réussi à observer les états internes malgré la grande influence du bruit de commande Z_0 et le bruit de mesure V sur la sortie de à commander et la sortie mesurée.

C'est le resultat du fait que le filtre de Kaman est synthétisé pour minimiser l'erreur entre l'état estimée et celle réelle, malgré la présence du bruit de mesure et de commande.

e) Changement de la variance du bruit Z_0 vers $\rho_0=1$:

Les résultats sont obtenu pour une condition initiale :

$$\hat{x}_0 = x_0 - e_0 = [0,0022; -0,1145; 0,0421; -0,0183] - [0,3; 1; 0,9; 1]^T,$$

Pour $\rho_0 = 1$, le gain de Kalman $K_{f2} = 10^4 [-0.00046.6657 - 0.0004 - 0.0000]^T$

La simulation sous Simulink a donné les résultats suivants :

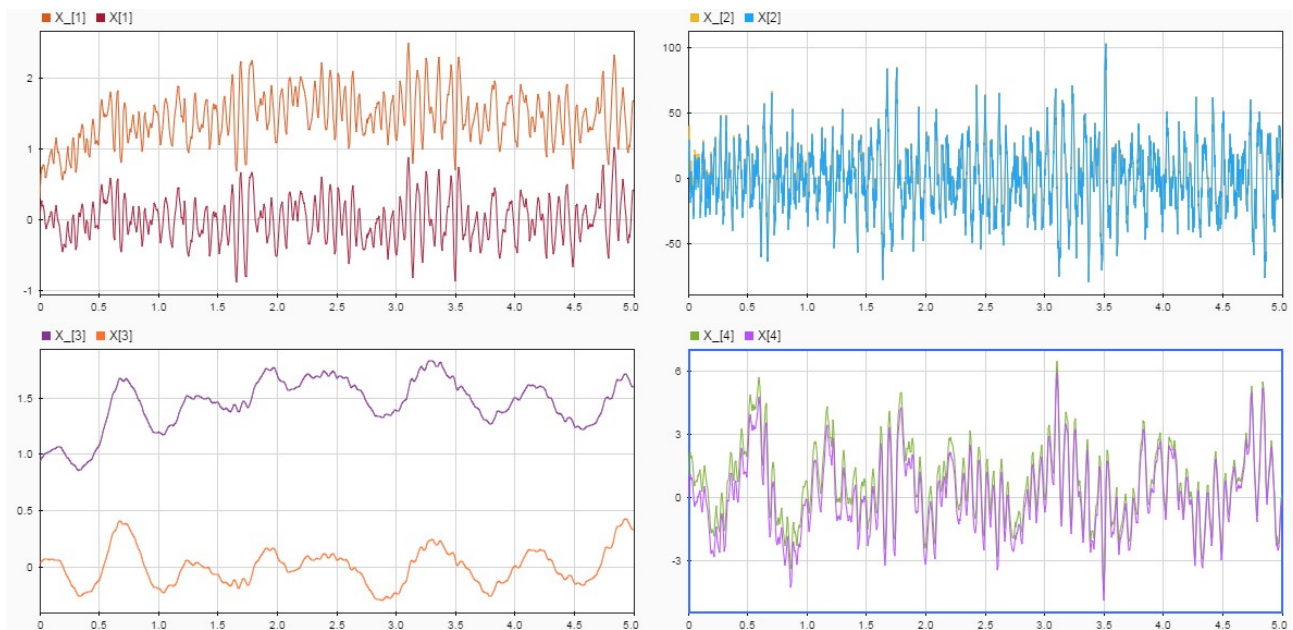


Illustration 17: Visualisation X et X_{est}

Dans ce cas il apparaît clairement une erreur statique au niveau de x_1 et x_3 estimés à cause de la valeur importante du bruit d'entrée Z_0 .

Cette erreur statique est annulée dans la sortie à commander estimée parce que cette dernière n'est que la différence entre x_1 et x_3 estimés qui portent des erreurs statiques proches. En effet le résultat de simulation nous donne :

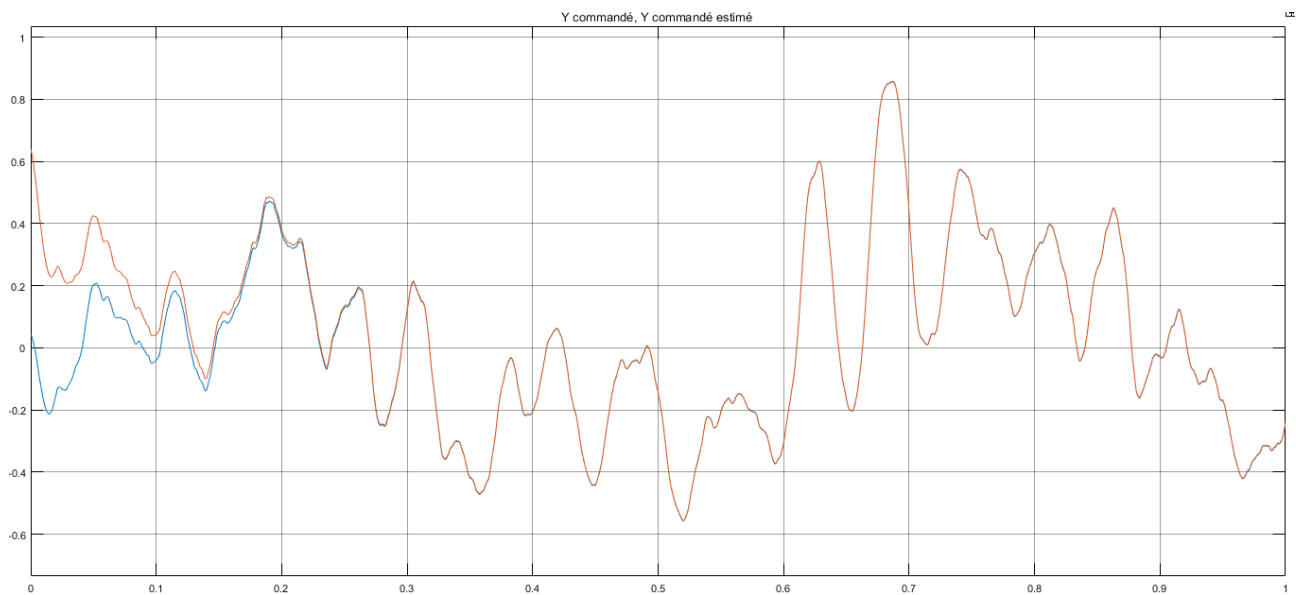


Illustration 18: Y et $Y_{\text{estimé}}$

Le filtre de Kalman a montré sa performance en rejeter les perturbations de mesure V , et à donner une bonne estimation des états internes malgré la grande influence de bruit d'entrer Z_0 sur ces états.

En plus, en prenant $\rho_0=1$ alors le filtre est synthétisé pour estimer des états influencés par des bruits assez plus importants que ceux dont $\rho_0=0.01$, ce qui signifie que sa dynamique (rapidité) s'augmente.

f) Les pôles du système :

La fonction *place* de Matlab permet de calculer le gain K d'un retour de boucle ou observateur avec les pôles désirés. Un script Matlab est utilisé :

```
%Comparaison des pÃ´les avec les differrentes methodes de calcul du gain de
%Kalman
P1 = eig(A - kf1 * C3);

%% Calcul du gain K de l'observateur de Kalman avec la fonction Place

Kplace = place (A', C3', P1);
```

La matrice de gain obtenue est:

$$K_{place} = 10^3 \begin{pmatrix} -0.0012 \\ 6.6577 \\ -0.0014 \\ -0.0000 \end{pmatrix}$$

La matrice de gain obtenue est équivalente à $kf1$.

5 Système bouclé : commande LQG

La commande LQG est réalisée avec simulink par le bloc LQE et LQR :

On remarque dans le premier graphe que les deux sortie sont identique, et la deuxième confirme que l'erreur entre eux est d'ordre 10^{-6} . d'où la commande LQG est équivalent à la commande LQ grâce à performance du filtre de Kalman qui a estimé l'état réelle avec une précision élevée tout en rejetant les bruits de mesure.

ii)) Les pôles du système bouclé :

Le système bouclé a la dynamique suivante :

$$\begin{pmatrix} \dot{x} \\ \dot{\hat{x}} \end{pmatrix} = \begin{pmatrix} A & -BF \\ K_f C & A - KC - BF \end{pmatrix} \begin{pmatrix} x \\ \hat{x} \end{pmatrix} + \begin{pmatrix} G & 0 \\ 0 & K_f \end{pmatrix} \begin{pmatrix} Z_0 \\ v \end{pmatrix}$$

'Le script suivant nous donne

```
Abf = [A , -B * K3 ; kf1 * C3 , A - kf1 * C3 - B * K3]; % Abf matrice boucle
ferme
Pbf = eig(Abf); % Calcul des poles de la matrice de boucle bf
```

les pole de ce système :

$$Pbf = 10^4 \begin{pmatrix} -2.6666 + 0.0000i \\ -0.0017 + 0.0083i \\ -0.0017 - 0.0083i \\ -0.0020 + 0.0000i \\ -0.0003 + 0.0009i \\ -0.0003 - 0.0009i \\ -0.0001 + 0.0001i \\ -0.0001 - 0.0001i \end{pmatrix}$$

Et le script suivant calcule les poles de l'observateur (LQE) et ceux de la boucle fermée LQR

```
% Calcul des poles du systeme
% pole du LQE
Po=eig(A - kf1 * C3);

% Pole du LQR
Pc = eig(A - B * K3);
```

Les pôles obtenus sont:

$$Po = 10^4 \begin{pmatrix} -2.6666 \\ -0.0020 \\ -0.0001 \\ -0.0001 \end{pmatrix}$$

et

$$P_c = \begin{pmatrix} -17.1157 + 83.3040i \\ -17.1157 - 83.3040i \\ -2.5118 + 8.7256i \\ -2.5118 - 8.7256i \end{pmatrix}$$

On remarque que Les pôles du systèmes en boucle fermé est l'union de Po et Pc. Et d'où **le principe de séparation est vérifié.**

III) Les poles du système en boucle fermée pour $\rho_0 = 1$:

Le gain de Kalman Kf2 est utilisé. Un script Matlab est réalisé :

```
% Calcul des poles du systeme question III
% pole du LQE
Po=eig(A - kf2 * C3);

% Pole du LQR
Pc = eig(A - B * K3);

Abf2 = [A , -B * K3 ; kf2 * C3 , A - kf2 * C3 - B * K3]; % Abf matrice boucle
ferme
Pbf2 = eig(Abf2); % Calcul des poles de la matrice de boucle bf
```

Les pôles obtenus sont :

$$P_o = 10^5 \begin{pmatrix} -2.6667 \\ -0.0002 \\ -0.0000 \\ -0.0000 \end{pmatrix}$$

$$\text{et } P_c = \begin{pmatrix} -17.1157 + 83.3040i \\ -17.1157 - 83.3040i \\ -2.5118 + 8.7256i \\ -2.5118 - 8.7256i \end{pmatrix}$$

Les pôles de boucle fermée obtenus sont :

$$P_{bf2} = 10^5 \begin{pmatrix} -2.6667 + 0.0000i \\ -0.0002 + 0.0008i \\ -0.0002 - 0.0008i \\ -0.0002 + 0.0000i \\ -0.0000 + 0.0001i \\ -0.0000 - 0.0001i \\ -0.0000 + 0.0000i \\ -0.0000 - 0.0000i \end{pmatrix}$$

On remarque que les pôles sont augmentés, les pôles qui ont changé sont ceux de l'observateur qui a synthétisé cette fois contre des perturbation supposées assez importantes d'où l'exigence de pole plus négative pour avoir une dynamique très rapide pour le filtre.

6 Conclusion

Ce TP1 de CAVAN a permis de constitué sur Matlab un régulateur LQG constitué du LQE aussi appelé filtre de Kalman (estimateur en présence de bruit blanc) et un LQR un régulateur retour de boucle avec un critère minimisé. Dans la première partie, nous avons pu réaliser grâce à la commande LQ une commande optimale pour optimiser le critère J. Dans la seconde partie nous avons pu réaliser un estimateur performant contre les bruits de commande et le bruit de mesure. Finalement grâce au principe de séparation nous avons synthétiser la commande LQG qui consiste à combiner LQE et LQR.