TP N°3 : Commande Modale

Sommaire

Introduction : 2

Gain Statique 3

Commande par retour d’état 3

Mise en pratique 4

Étude générale 4

1.3 Rejet de la perturbation échelon 6

1.4 Rejet de perturbations de type rampe : 8

1.5 Rejet de perturbation sinusoïdale. 10

1.6 Perturbation de type constante+Sinus 14

Conclusion : 18

Annexes matlab 19

Question 1.3 19

Question 1.4 20

Question 1.5 21

Question 1.6 22

# Introduction :

On considère le système.

L’enjeu de ce TP va être de réguler la sortie y autour d’une consigne échelon avec rejet des différents types de perturbations.

# Gain Statique

Pour le calcul du gain statique, le théorème des limites nous permet de trouver très rapidement ½ :

On a donc bien A=-2, B=1 et C=1

# Commande par retour d’état

On applique une commande par retour d’état, pour cela, on pose.

Il nous faut déterminer le correcteur K et le scalaire h telle que la Bf présente des pôles en -3 et un gain statique unitaire. C’est-à-dire qu’on cherche à se ramener à cette forme-là :

⬄

Par identification, il vient tout de suite h=3 et k=1.

# Mise en pratique

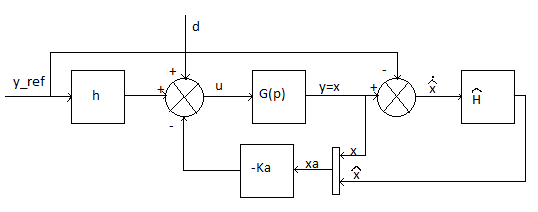
Étude générale

Nous allons dans un premier temps étudier de manière générale la réponse à une perturbation.

Représentation d’états :

Pour ce genre de système, il est nécessaire d’augmenter notre représentation d’états, de calculer un correcteur Ka et une fonction afin de rejeter les perturbations d(p) en entrée du système.

Schéma de principe du système augmenté :

**Système augmenté :**

Représentation d’états du système augmenté.

On pose et , ou la taille de la matrice dépendra de l’ordre de la fonction .

Comment calculer pour rejeter les perturbations d’entrée ?

On cherche à avoir (théorème des valeurs limites), ou S(p) est la fonction de sensibilité du système en boucle fermée :

quand

Calculons

On sait que et donc :

soit :

(1-1)

On cherche maintenant à avoir

On connais G(p) qui est de premier ordre, mais d(p) n’est pas fixe. Si d(p) possède un intégrateur, alors il sera impossible d’avoir une erreur statique nulle. On va donc chercher à annuler le dénominateur de la fonction d(p). Pour cela, on va utiliser .

La relation (1-1) peut être écrite sous la forme

On remarque que si le dénominateur de est égal au dénominateur de , alors les intégrateurs sur la perturbation seront annulés.

On dois par ailleurs avoir le numérateur de pour éviter d’avoir un intégrateur sur

Durant la suite du TP, nous aurons donc pour les différents types de perturbations une fonction

Et nous aurons une matrice de la taille de l’ordre de la fonction .

# 1.3 Rejet de la perturbation échelon

On cherche à rejeter une perturbation de type une perturbation échelon.

On a donc et

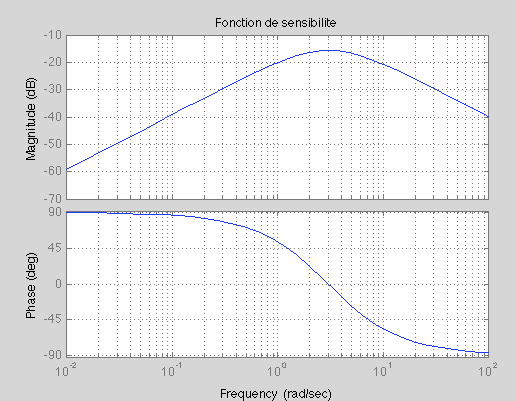
Calcul du correcteur Ka du système augmenté :

Or nous voulons que le correcteur fixe les pôles en -3 donc

Par identification,

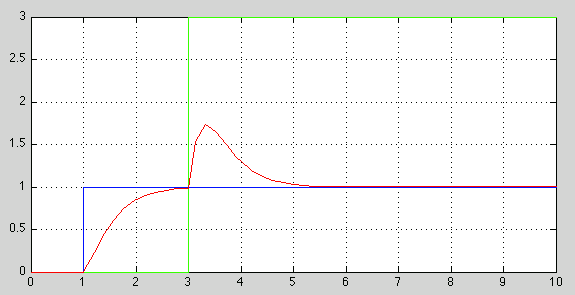
Nous implémentons notre système augmenté et trouvons via Matlab la fonction de sensibilité du système augmenté :

Avec son bode :



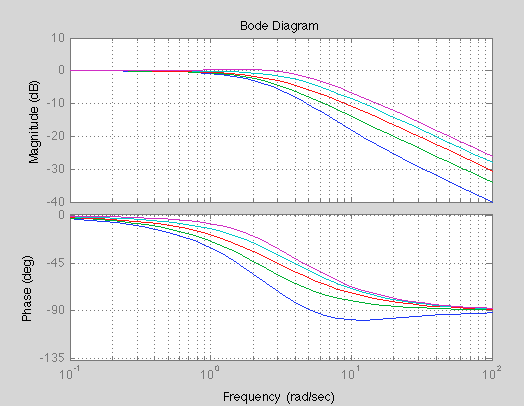
On a bien une fonction de sensibilité qui tend vers 0 quand p est proche de 0.

Nous avons ensuite simulé sous simulink la réponse à une perturbation de type échelon, avec en jaune la consigne, en violet la perturbation et en bleu le suivi de consigne :



Le système rejette bien asymptotiquement les perturbations !

Pour finir, nous avons tracé le bode de la fonction de transfert du système en boucle fermé en faisant varier « h » pour étudier l’influence de se dernier :



On observe que plus h est grand et plus la fréquence de coupure augmente. Le coefficient h n’influe donc pas sur le gain statique, mais sur la vitesse du système. Conserver h n’a donc de sens que si l’on désire avoir un système plus rapide.

Code Matlab en annexe.

# 1.4 Rejet de perturbations de type rampe :

Nous avons ici une perturbation du type

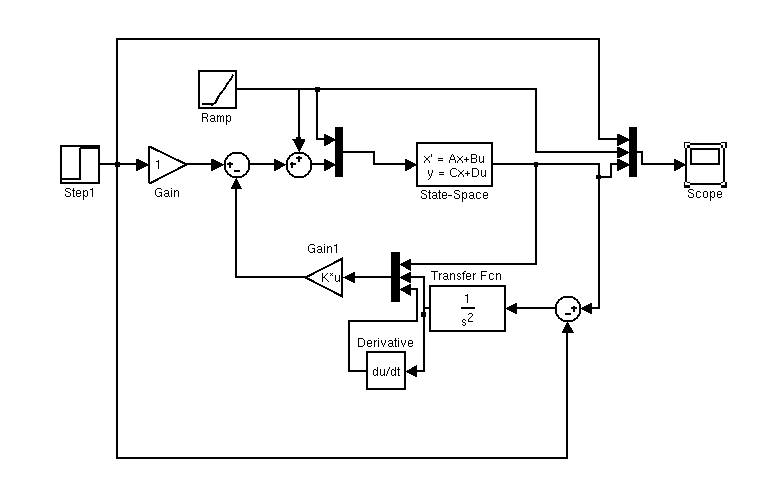
On a donc une fonction et .

On a alors la représentation d’états :

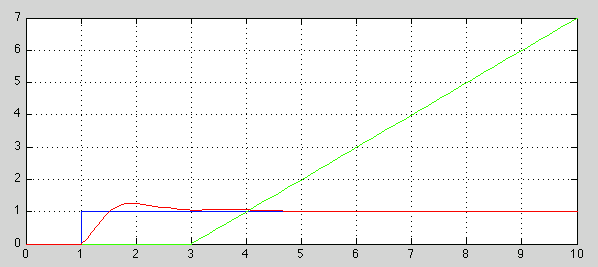
On calcule le correcteur Ka sous matlab :

Ka = 7 27 27

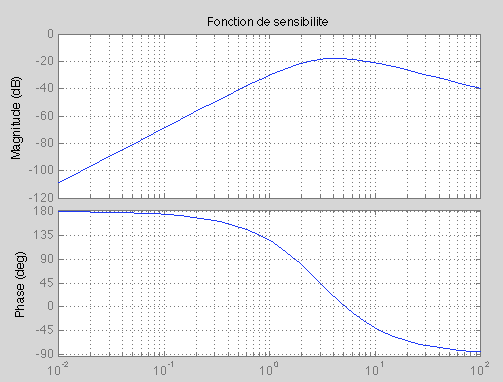
Sous simulink, nous implémentons le schéma suivant:



Et nous obtenons la réponse à une rampe suivante. La réponse est en rouge :



Fonction de sensibilité :



Ici, on remarque que le gain statique est bien nul.

Code Matlab en annexe.

# 1.5 Rejet de perturbation sinusoïdale.

On désire ici rejeter une perturbation sinusoïdale d’équation donc sur le plan de Laplace : .

On augmente de deux notre représentation d’états car nous avons encore une fois une perturbation d’ordre 2. On note le système augmenté de .

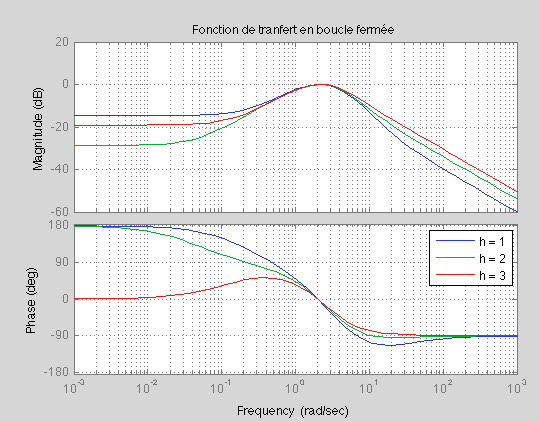
On a donc donc et on a . On ne se préoccupe pas du numérateur, la seule chose qui nous intéresse est d’avoir un dénominateur de même forme que le dénominateur de la perturbation, afin de rejeter cette perturbation.

Représentation d’états du système augmenté :

Nous implémentons donc le calcul du correcteur du système augmenté sous Matlab et obtenons le résultat suivant :

Ka = 7 -9 23

On désire ensuite connaître la valeur du gain h. En effet, selon la valeur du gain h, la réponse du système en boucle fermée change, comme nous le montre ce diagramme de Bode pour lequel nous avons fait varier la valeur de h.



Code Matlab en annexe.

On désire avoir un gain asymptotique de 1, soit ou est la fonction de transfert en boucle fermée.

En faisant varier le gain h, nous obtenons différentes valeurs sur les coefficients des fonctions de transfert, et nous en déduisons une relation entre ceux-ci et h.

Résultats matlab :

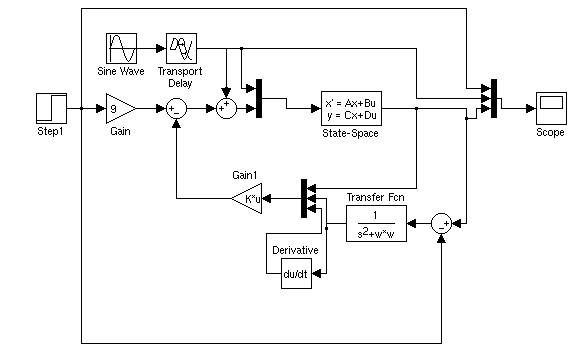
|  |  |
| --- | --- |
| Gain h = 1 | Gain h = 2 |
| Transfer function:  s^2 + 23 s - 5  -----------------------  s^3 + 9 s^2 + 27 s + 27 | Transfer function:  2 s^2 + 23 s - 1  -----------------------  s^3 + 9 s^2 + 27 s + 27 |

On en déduit la relation entre les coefficients de la fonction de transfert et h suivante :

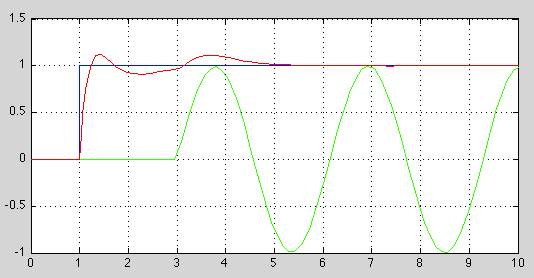
On cherche maintenant à avoir un gain statique nul, donc si donc si .

Test du gain sous simulink :

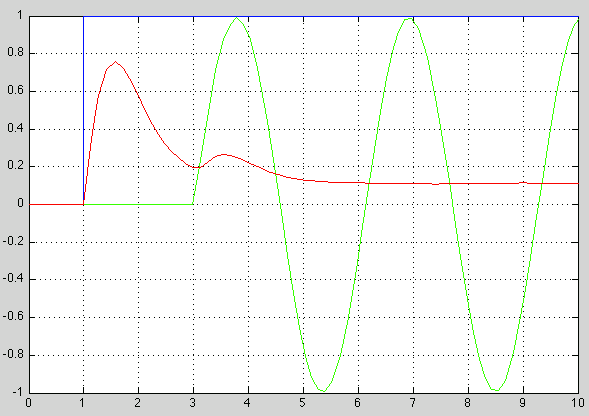
On implémente le schéma suivant :



Et nous obtenons les résultats suivant :

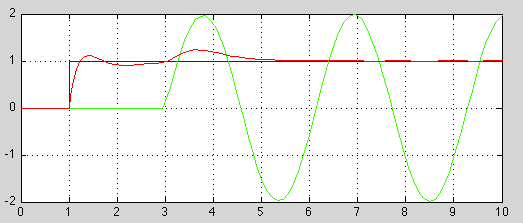


On remarque donc un gain statique nul sur la courbe rouge (réponse du système en boucle fermée).

Si nous changeons h= 3, nous obtenons la réponse suivante :

Il est donc important de garder le coefficient h dans ce cas de perturbations.

Nous remarquons par ailleurs que quelque soit l’amplitude du signa sinusoïdal perturbateur appliqué, celui-ci est rejeté :



# 1.6 Perturbation de type constante+Sinus

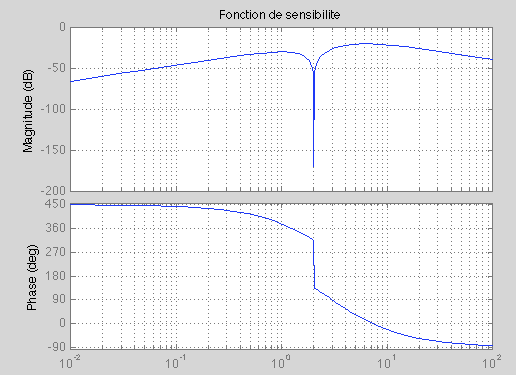
On désire ici réguler le même système, mais avec une perturbation en sinus+constante . Le dénominateur est d’ordre 3, on va donc avoir un correcteur  
 et une représentation d’états augmentée de trois états notés

Donc

Représentation d’états :

On calcule ensuite le correcteur Ka en boucle fermée et obtenons le correcteur suivant :

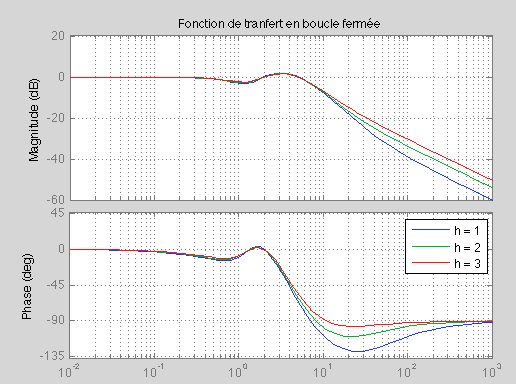
Ka = 10 81 60 50

Bode de la sensibilité du système :

A-t-on besoin du gain h ? Pour cela, nous nous basons sur le même principe que les questions précédentes et calculons plusieurs fonctions de transfert en boucle fermée en faisant varier h sous matlab. Nous obtenons le tableau suivant :

|  |  |
| --- | --- |
| Gain h = 1 | Gain h = 2 |
| Transfer function:  s^3 + 50 s^2 + 64 s + 81  ----------------------------------  s^4 + 12 s^3 + 54 s^2 + 108 s + 81 | Transfer function:  2 s^3 + 50 s^2 + 68 s + 81  ----------------------------------  s^4 + 12 s^3 + 54 s^2 + 108 s + 81 |

On cherche toujours . Et nous avons la fonction de transfert en fonction du gain h :

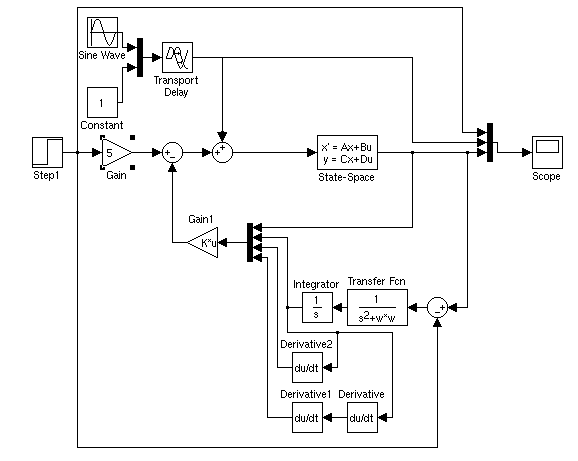
On en déduit ici que le gain h n’a pas d’influence sur le gain statique du système. On remarque cela sous matlab en traçant les diagrammes de Bode des fonctions de transfert en boucle fermée en faisant varier h :  


On remarque, que quelque soit la valeur de h, le gain statique est le même. Seule la rapidité du système change.

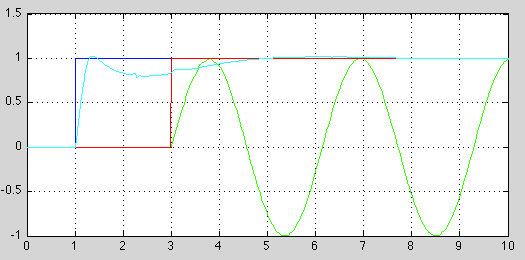
Code source Matlab en annexe 6.

Test du système en boucle fermée avec simulink :

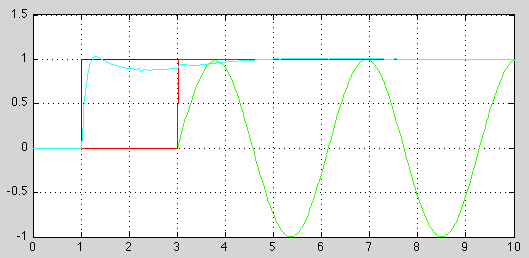
Schéma :



Réponse avec h = 1



Réponse avec h = 5



On peut confirmer que h n’influe que sur la rapidité du système, ce que nous avions remarqué sur le diagramme de Bode.

# Conclusion :

Ce TP nous a permis d’appliquer nos savoirs acquis en cours sur la commande d’un système avec une représentation d’états augmentée.

Nous avons en effet cherché à rejeter différents types de perturbations en entrée du système, et pour cela nous avons augmenté notre représentation d’états et calculé des correcteurs permettant de rejeter ces perturbations.

# Annexes matlab

Question 1.3

a = -2;

c = 1;

d = [0 0];

B = 1;

E = 1;

BE = [B E];

%Calcul de Ka:

Aa = [-2 0;1 0];

Ba = [1;0];

Ca = [1 0];

Ka = acker(Aa,Ba,[-3 -3])

% Fonction de transfert

N = 3;

h = ones(1,N)

for h = 1:N

Bbf =Ba\*h + [0;-1]

[num1,den1] = ss2tf(Aa-Ba\*Ka,Bbf,Ca,0);

SYS2 = tf(num1,den1)

figure(2)

hold on

bode(SYS2)

grid on

end

Question 1.4

% les matrices:

Aa = [-2 0 0;0 0 1;1 0 0];

Ba = [1;0;0];

Ea = [1;0;0];

Da = [0;0;-1];

Ca = [1 0 0];

a = -2;

c = 1;

d = [0 0];

B = 1;

E = 1;

BE = [B E];

% Correcteur Ka:

P = [-3 -3 -3];

Ka = acker(Aa,Ba,P)

%Fonction de sensibilite:

[num,den] = ss2tf(Aa-Ba\*Ka,Ea,Ca,0);

SYS = tf(num,den)

bode(SYS)

grid on

title('Fonction de sensibilité')

Question 1.5

a = -2;

c = 1;

d = [0 0];

B = 1;

E = 1;

BE = [B E];

w= 2;

Aa = [-2 0 0;0 0 1;1 -4 0];

Ba = [B;0;0];

Ea = [1;0;0];

Da = [0;0;-1];

Ca = [1 0 0];

% Correcteur Ka:

P = [-3 -3 -3];

Ka = acker(Aa,Ba,P)

%Fonction de sensibilite:

[num,den] = ss2tf(Aa-Ba\*Ka,Ea,Ca,0);

SYS = tf(num,den)

bode(SYS)

grid on

title('Fonction de sensibilité')

% Fonction de transfert

N = 3;

h = ones(1,N)

for h = 1:N

Bbf =Ba\*h + [0;0;-1];

[num1,den1] = ss2tf(Aa-Ba\*Ka,Bbf,Ca,0);

SYS2 = tf(num1,den1)

figure(2)

hold on

bode(SYS2)

grid on

end

legend('h = 1','h = 2','h = 3')

title('Fonction de transfert en boucle fermée')

Question 1.6

a = -2;

c = 1;

d = [0 0];

B = 1;

E = 1;

BE = [B E];

w= 2;

Aa = [-2 0 0 0;

0 0 1 0;

0 0 0 1;

1 0 -4 0];

Ba = [B;0;0;0];

Ea = [1;0;0;0];

Ca = [1 0 0 0];

% Correcteur Ka:

P = [-3 -3 -3 -3];

Ka = acker(Aa,Ba,P)

%Fonction de sensibilite:

[num,den] = ss2tf(Aa-Ba\*Ka,Ea,Ca,0);

SYS = tf(num,den)

bode(SYS)

grid on

title('Fonction de sensibilité')

% Fonction de transfert

N = 3;

for h = 1:N

Ba\_bf =Ba\*h + [0;0;0;-1];

[num1,den1] = ss2tf(Aa-Ba\*Ka,Ba\_bf,Ca,0);

SYS2 = tf(num1,den1)

figure(2)

hold on

bode(SYS2)

grid on

end

legend('h = 1','h = 2','h = 3')

title('Fonction de transfert en boucle fermée')