TP N°1 : Commande d’un système mecanique : Le pendule inversé

Sommaire

[Introduction : 1](#_Toc295048111)

[Modelisation non lineaire (NL) 3](#_Toc295048112)

[Approche Lagrangienne : 3](#_Toc295048113)

[Modelisation lineaire 6](#_Toc295048114)

[Recherche des états d’équilibre 6](#_Toc295048115)

[Linéarisation autour du point d’équilibre : 6](#_Toc295048116)

[Changement de base 9](#_Toc295048117)

[Propriétés de commande et d’observation 10](#_Toc295048118)

[Propriétés de stabilité 11](#_Toc295048119)

[Modèle NL sous Matlab, comparaison et conclusion 12](#_Toc295048120)

[Conclusion : 13](#_Toc295048121)

# Introduction :

Le but de ce TP d’automatique est de réussir à synthétiser la commande d’un pendule inversé. Pour cela, le sujet du TP nous aide en fixant une démarche basée sur 5 objectifs dans l’ordre :

1. Modélisation du système par les lois classiques de la mécanique.
2. Linéarisation des équations trouvées en (1) autour d’un point d’équilibre.
3. Représentation du système sous forme d’états internes.
4. Analyse des propriétés de commandabilité, d’observabilité et de stabilité.
5. Et enfin, synthèse d’une loi de commande par retour d’états.

# Modelisation non lineaire (NL)

Le pendule inversé peut être modélisé comme étant deux masses en mouvement (le chariot et la masselotte du pendule) et reliées entre elles par une tige de masse négligeable :

## Approche Lagrangienne :

On cherche à mettre notre système sous la forme suivante :

Avec :

* est la position du chariot
* est la vitesse du chariot
* est la position angulaire
* est la vitesse angulaire
* 🡪
* 🡪

**Energie cinétique :**

Soit l’énergie cinétique du chariot et et les composantes de l’énergie cinétique de la masselotte selon et

Le **chariot** ne peut se déplacer que selon  :

La **masselotte** peut se déplacer selon ξ et  :

L’énergie cinétique totale du système est donc :

**Energie potentielle :**

Seule la masselotte est concernée, en effet, le chariot ne peut avoir de mouvements verticaux.

**Calcul du Lagrangien et de la forme d’état non linaire :**

Calcul du Lagrangien :

Nous pouvons maintenant utiliser les formules d’Euler Lagrange pour nous ramener à une forme  :

Euler Lagrange :

Déterminons  :

Ainsi,

Donc,

Avec les forces et couples extérieurs au système. Nous avons donc :

**En effet, la seule force appliquée à notre système est la force u. Il n’y a pas de couples externes appliqué au système (un chariot qui se déplace linéairement sur un plan horizontal) vaut donc 0.**

A partir de ce couple d’équation, nous allons essayer de nous retrouver sous cette forme :

Pour cela, nous allons dans un premier temps isoler les du système d’équation précédent :

**Cette dernière formule, mise sous une autre forme donne** :

**CQFT ! Ouf…**

# 1.2 Modelisation lineaire

L’objectif ici est d’avoir une commande linéaire. Il faut donc qu’on détermine une modélisation linéaire à partir du système non linaire trouvé précédemment.

## Recherche des états d’équilibre

Les états d’équilibres sont déterminés quand leurs dérivés s’annulent :

Donc on peut écrire le système suivant :

Les termes ~~barrés~~ sont nuls

Si on admet que est non nul, on peut écrire :

⬄

⬄

Nous ne nous intéressons au cas car cela nous donne donc qui correspondent physiquement à un résultat interprétable : le pendule est vertical, orienté vers le haut (ou vers le bas (. Évidemment, nous ne retiendrons que le premier cas, car dans notre système, il existe des butées qui empêchent le pendule de subir un angle trop important.

## Linéarisation autour du point d’équilibre :

Autour du point d’équilibre, la formule de Taylor permet d’écrire :

Nous voulons la représentation donc :

rappel :

**Calcul des  :**

(Après développement et simplification…)

Or, au voisinage du point d’équilibre, on considère que et il vient donc

De même, nous calculons

Ainsi, on obtient la matrice

**Calcul des  :**

Tout de suite, il vient :

Or,  :

Donc,

Nous obtenons le système linéarisé suivant :

Qui représente le système d’équation :

En faisant une combinaison linéaire de ces deux équations, nous trouvons :

L’objectif, va maintenant être de trouver la fonction de transfert

Donc, en passant en Laplace :

La fonction de transfert est :

Nous pouvons l’arranger :

Les pôles de ce système sont : et .

On peut tout de suite prédire que le système est instable en boucle ouverte car l’un des pôles est à partie réelle négative. L’enjeu de la partie suivante va être de trouver un correcteur PD qui puisse réguler ce système instable par nature.

**Correcteur PD :**

Nous allons chercher un correcteur de type Proportionnel Dérivable (PD) qui puisse asservir en position et en angle notre pendule. Le retour d’information sera assuré par un capteur d’angle uniquement :

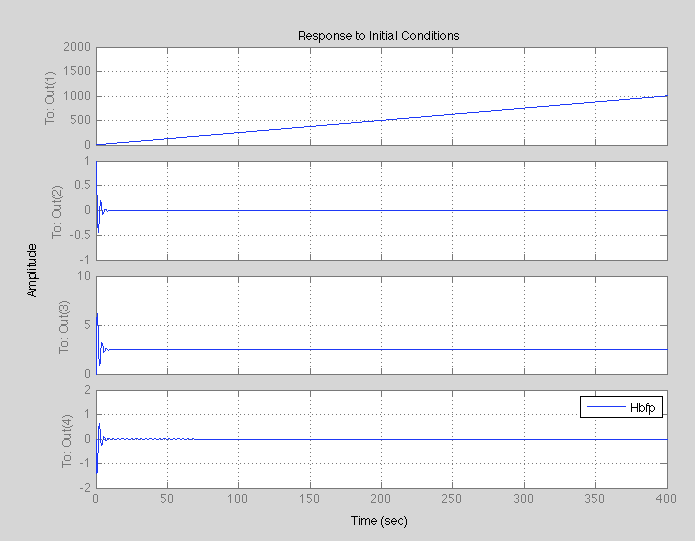
Le but étant de trouver k1 et k2 tel que soit stable.

Nous obtenons la fonction de transfert en boucle fermée suivante :

Pour déterminer et nous allons étudier les pôles de. Utilisons le critère de Routh. Celui-ci nous donne les conditions suivantes pour la stabilité :

Nous choisissons  et.

Utilisons Matlab pour visualiser la réponse du système à partir de sa représentation d’état :



Les 3 courbes représentent les sorties dans l’ordre.

On remarque que le pendule parvient à se stabiliser à la verticale car θ tend rapidement vers 0° et se maintient à 0°. Cependant cette condition est obtenue avec le déplacement du chariot à vitesse constante. Dans le système réel, cette condition n’est pas réalisable, le chariot irait rapidement en butée.

Il faut donc asservir le système en angle mais aussi en position.

# 1.3 Changement de base

Posons  :

Avec P la matrice de transfert de la base dans la base

On utilise Matlab pour faire le calcul de la matrice inverse à notre place :

Matlab nous donne :

Pour passer dans la nouvelle base on applique les relations suivantes :

On obtient les nouvelles matrices suivantes :

On obtient alors sur matlab les fonctions de transfert suivantes :

Fonction de transfert H1

Transfer function:

-s^2

--------------

s^4 - 20.6 s^2

Changement de base :

Fonction de transfert H1

Transfer function:

8.882e-016 s^3 - s^2

-------------------------------

s^4 - 8.882e-016 s^3 - 20.6 s^2

Fonction de transfert H2

Transfer function:

6.217e-015 s^3 + 0.5 s^2 + 1.288e-014 s - 9.81

----------------------------------------------

s^4 - 20.6 s^2

Changement de base :

Fonction de transfert H2

Transfer function:

1.193e-015 s^3 + 0.5 s^2 + 1.242e-014 s - 9.81

----------------------------------------------

s^4 - 8.882e-016 s^3 - 20.6 s^2

En négligeant les coefficients proches de 0, nous remarquons que celles-ci sont identiques à celles obtenues précédemment dans l’autre base. On en conclu qu’il existe une seule fonction de transfert alors qu’il existe plusieurs représentations d’état, suivant la base utilisée.

# 1.4 Propriétés de commande et d’observation

Etude de la commandabilité du système 1-3 :

Afin d’étudier la commandabilité du système 1-3, nous nous basons sur le théorème de Kalman. On utilise donc les fonctions ctrb(), qui donne la matrice de contrôlabilité [B AB]. On calcule ensuite le rang de cette matrice avec la fonction matlab rank(). On obtient alors le résultat suivant :

RG\_sys1 = 4 Le range de la matrice de contrôlabilité est égal à la taille de la matrice de commandabilité. Le système est donc complètement commandable d’après Kalman.

Étude de l’observabilité du système 1-3 :

Nous allons maintenant étudier l’observabilité des états dans le cas a) et dans le cas b) du système 1-3. On aura donc respectivement les matrices C :

et .

On se sert de la commande obsv() sous matlab, qui nous renvoie la matrice d’observabilité . On calcule ensuite le rang de ces matrices avec la fonction rank() de matlab et on obtient les résultats suivants :

RG\_sys2\_H1 = 2

RG\_sys2\_H2 = 4

Ici, le rang de la matrice d’observabilité est de 2 pour l’état , cet état n’est donc pas observable. En revanche, l’état est observable car le rang de sa matrice d’observabilité est 4.

Maintenant, il est intéressant de voir si la nouvelle base possède les mêmes propriétés d’observabilité. Après une simulation sous matlab, nous obtenons les résultats suivants :

RG\_sysZ\_H1 = 2

RG\_sysZ\_H2 = 4

On obtient les mêmes résultats que la représentation d’états 1-3. Changer de base n’altère donc pas les propriétés d’observabilité d’un système.

# 1.5 Propriétés de stabilité

Étude de stabilité du système : on se sert de la fonction eig() de matlab qui nous donne les valeurs propres d’une matrice. On obtient donc les valeurs propres de A sous matlab :

Stab =

0

0

4.5388

-4.5388

Une des valeurs propres est positive, le système est donc instable.

Nous cherchons maintenant à calculer un correcteur K afin de rendre stable le système.

Avec la commande et le système , le système en boucle fermée devient :

On souhaite déterminer la matrice K du retour d’état. Il faut que K agisse sur tous les états. Pour cela nous utilisons la fonction *acker()* qui permet de placer les pôles du correcteur. On choisis de les placer en -1.

Le correcteur K renvoyé par acker() est le suivant : K = -0.1019 -26.6520 -0.4077 -4.2039 et la matrice (A-B\*K) est possède alors les valeurs propres suivantes :

-1.0002 + 0.0002i

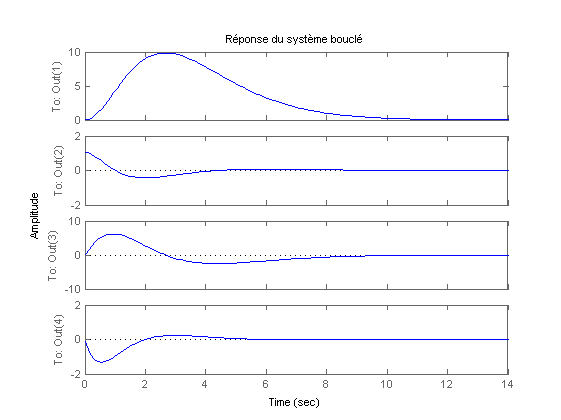
-1.0002 - 0.0002i

-0.9998 + 0.0002i

-0.9998 - 0.0002i

Il y a donc une certaine erreur sous matlab, mais si on néglige les coefficients très petits (>1/100) alors on obtiens quasiment les valeurs propres demandées.

Nous obtenons la réponse suivante avec la fonction initial() :



Nous remarquons que tous nos états sont maintenant stables en boucle fermée.

# Modèle NL sous Matlab, comparaison et conclusion

# Conclusion :