**Compte rendu**

**TP6**

**Commande optimale : le pendule inversé**

Contents

[I. Introduction 1](#_Toc126588209)

[II. Commande LQ (retour d’état). 1](#_Toc126588210)

[II.1 Modèle du système 1](#_Toc126588211)

[II.2 Recherche des matrices Q et R. 1](#_Toc126588212)

[III. Commande LQ avec action intégrale. 2](#_Toc126588213)

[IV. Commande par placement des pôles. 2](#_Toc126588214)

[V. Commande par placement de pôles avec action intégrale. 2](#_Toc126588215)

[VI. Conclusion 2](#_Toc126588216)

# Introduction

Lors de ce TP, nous allons travailler sur le robot Lego EV3 qui peut être assimilé à un pendule inversé. Tout l’enjeu de ce TP est de mettre en place une commande optimale afin de réguler ce robot qui est instable. Pour tenter de faire tenir le robot sur ces deux roues, nous allons utiliser quatre commandes différentes : LQ, LQ avec action intégrale, par placement des pôles, et enfin par placement des pôles avec action intégrale. Chaque commande sera d’abord vérifiée par simulation, puis expérimentalement.

# Commande LQ (retour d’état).

Afin de mettre en place une commande LQ, il faut définir les matrices Q et R afin de calculer le coefficient K en minimisant un critère quadratique.

## Modèle du système

L’état du système peut être défini à partir de 4 variables d’état qui sont : Ψ l’angle d’inclinaison du robot par rapport à l’axe z (en degrés), Ψ’, 𝜃 l’angle des roues (en degrés) et 𝜃’. On obtient alors le vecteur d’état : x=. De plus, le système possède deux sorties qui sont vL et vR ², respectivement la vitesse de la roue gauche et droite. On a alors le vecteur de sortie : y= [vL vR].

## Recherche des matrices Q et R.

La taille de la matrice Q correspond à l’ordre du système soit 4. La taille de la matrice R est donnée par le nombre de sortie soit 2. Cependant, le couple des moteurs appliqué à chaque roue est le même (travaille réalisé en ligne droite) donc vL = vR. Ainsi, on peut considérer que le système ne possède qu’une seule sortie. La matrice devient alors un scalaire.

Après avoir complété le bloc « commande » dans le fichier Simulink de la même manière que présenté en Annexe 1, nous pouvons chercher les meilleures valeurs de Q et R en simulation grâce au script Matlab donné en annexe B de l’énoncé. Le tableau en Annexe 2 présente les caractéristiques de la réponse obtenue pour différentes valeurs de Q et R. De ce tableau, on peut en déduire que les meilleures valeurs à prendre pour Q et pour R sont :

Q = R=

La réponse obtenue avec ces valeurs de Q et R est présentée en Annexe 3.

En effet, on remarque que le second coefficient sur la diagonal de Q est essentiel pour avoir le meilleur comportement du robot. Lorsque ce coefficient augmente, la position debout du robot est plus stable. Cependant si ce coefficient devient trop grand, le comportement du robot devient trop saccadé même si la commande paraît optimale. On prend donc un juste milieu avec une valeur de 1000.

Pour tous les autres coefficients, les augmenter a plutôt un effet négatif sur le comportement du robot. On fait donc le choix de tous les laisser à 1.

Avec l’implémentation du correcteur obtenue, le robot en réel est très stable. Les observations sont conformes à la simulation. Le robot est même capable de résister à de faibles perturbations. La réponse obtenue en pratique est présentée en Annexe 4.

# Commande LQ avec action intégrale.

Avec seulement une commande LQ, on remarque une dérive du robot selon l’axe y comme on peut le voir sur l’Annexe 5. Pour corriger ce problème, on ajoute une action intégrale sur l’angle 𝜃. On modifie alors le bloc « commande » du fichier Simulink de la même manière que présenté en Annexe 6.

Afin de pondérer l’action intégrale ajoutée, on rajoute un coefficient sur la diagonale de la matrice Q. On simule ensuite le modèle pour plusieurs valeurs de ce coefficient pour choisir la meilleure. Les résultats de ces tests sont présentés dans le tableau en Annexe 7. D’après ce tableau, le coefficient le plus adapté pour l’action intégrale est 1.

Décrire comportement du robot avec la commande implémentée (annexe 8)

# Commande par placement des pôles.

Avec la commande LQ, les valeurs propres du système sont indépendantes du gain K. Pour pouvoir contrôler au mieux la dynamique de notre système, on cherche les valeurs propres de la matrice dynamique en boucle fermée Abf = A – BK. Pour cela, on utilise la commande « eig(Abf) » de Matlab.

On trouve donc les valeurs propres suivantes : mettre les vp

-188.7591

-0.3991

-7.4982

-6.2521

Maintenant que nous connaissons les pôles, nous allons les modifier et essayer de trouver un placement de pôle optimal. Grâce à la fonction « place(A,B,pôles) » de Matlab, nous modifions le pôle le plus lent, c’est-à-dire le plus proche de l’axe imaginaire. Les résultats obtenus sont présentés en Annexe 9. On remarque que si l’on diminue trop le pôle dominant, le système devient trop oscillatoire. La valeur de -1 semble donc être un bon compromis.

En réel, on observe avec cette commande que le robot n’est pas très stable et oscille beaucoup. On observe encore une fois une déviation par rapport à l’axe y, comme l’Annexe 10 en témoigne.

# Commande par placement de pôles avec action intégrale.

Pour corriger la déviation du robot dans l’espace, nous ajoutons une action intégrale. On calcule les valeurs propres de la matrice en boucle fermée. On obtient alors… On remarque que ces valeurs propres sont toutes négatives, notre système est bien stable. On remarque également la présence d’un pôle complexe conjugué. Pour obtenir une réponse plus amortie, on ramène ces 2 pôles sur l’axe réel. Les résultats obtenus pour les différentes valeurs des pôles sont présentés en Annexe 11.

Grâce à ce tableau, on sélectionne la réponse obtenue avec les pôles … En réel, le robot …

# Conclusion

Au travers de ce TP, nous avons pu mettre en place différentes commandes et les comparer entres elles. Nous avons pu faire l’étude de la commande LQ qui ne nécessite pas de choisir les valeurs propres tout en restant robuste. Le choix des matrices Q et R est un choix primordial dans cette méthode car il permet la pondération des coefficients. Cependant cette commande n’est pas suffisante pour éviter une dérive selon l’axe y. C’est pourquoi la commande LQ avec action intégrale a été implémentée. Cette commande efface effectivement la dérive mais a augmenté les oscillations du robot. Nous avons donc essayé par la suite une commande par placement de pôles en prenant comme base les pôles trouvés avec la commande LQ. On obtient alors une réponse plus rapide tout en gardant le problème de la dérive selon l’axe y. On a enfin ajouté une action intégrale à cette commande qui a permit d’effacer cette dérive tout en gardant des oscillations faibles. Les résultats finaux obtenus sont très satisfaisant et répondent à la problématique de départ.