

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud MAMMERI, Tizi-Ouzou



Faculté de Génie Electrique et d'Informatique
Département d'Automatique

Mémoire de Fin d'Etudes

En vue de l'obtention du diplôme

De Master académique en Automatique
Option Commande des systèmes

Thème

*Etude comparative des commandes LQR, LQG et
placement de pôles appliquées à un système mécanique.*

Proposé par : S. DJENNOUNE

Présenté par :

Karima CHAIBI

Dirigé par : S. DJENNOUNE

Soutenu le : 13 /07 /2010

Promotion 2010

Remerciement :

Je remercie mon promoteur de m'avoir fait confiance en m'accordant de son temps et de ses connaissances, je remercie ma famille qui m'a soutenue tout au long de ce projet et qui a supporté mes hauts et mes bas ; je remercie mon fiancé de m'avoir apporté une aide inestimable à l'aboutissement de mon travail. Je remercie les membres du jury de m'avoir fait l'honneur d'évaluer mon travail

Je remercie tous mes amis et toute personne ayant contribué de près ou de loin à l'aboutissement de ce mémoire.

A mes très chers parents qui ont toujours été là pour moi, et qui m'ont donné un magnifique modèle de labeur et de persévérance. J'espère qu'ils trouveront dans ce travail toute ma reconnaissance et tout mon amour.

A mes très chère sœur KAHINA et MOUNA.

A mon très cher petit frère ZAKARI

A mon très cher fiancé SOFIANE

A ma belle famille

A mon grand père qui m'a apporté tant d'assurance avec ses prières

A tout (es) mes amis (es) pour leurs soutien moral

Je dédie ce mémoire.

Résumé :

Notre projet va consister en l'étude comparative de trois commandes par retour d'état. Ces commandes sont la commande par placement de pôles, la commande LQR et la commande LQG. Elles seront appliquées au système ball and beam linéarisé autour de l'origine. L'étude de la stabilité des asservissements est aussi étudiée dans ce projet.

Nous verrons aussi la manière de réaliser une simulation animée de notre système mise en œuvre sous le logiciel Matlab permettant de simuler le déplacement de la balle sur la barre après application de la commande par placement de pôles.

Table des matières :

Table des matières	1
Table des figures	5
Liste des symboles	9
Introduction générale	11
 CHAPITRE 1 : Présentation et modélisation du système	 15
1.1 Introduction	17
1.2 Présentation du système	17
1.2.1 Différentes configurations du système « Ball and beam ».....	17
1.2.2 Objectif de commande	18
1.2.3 Exemples pratiques analogiques au système	19
1.3 Modélisation du système.....	19
1.3.1 Description physique	20
1.3.2 Modélisation du mouvement de la balle	20
1.3.3 Modélisation du moteur à courant continu.....	22
1.3.4 Modélisation des couples appliqués sur la barre.....	23
1.3.5 Modèle d'état non linéaire global	24
1.4 Modèle linéaire et propriétés.....	29
1.4.1 Linéarisation autour d'un point d'équilibre	29
1.4.2 Test de commandabilité et observabilité	32
1.4.3 Réduction de l'ordre du système	32
1.5 Conclusion	34
 CHAPITRE 2 : Etude théorique des commandes développées	 35
2.1 Placement de pôles.....	37
2.2 Observateur d'état	38

2.3	Commande LQR.....	39
2.4	Commande LQG	41
2.5	Commande LQG/LTR (Loop Transfer Recovery).....	43
CHAPITRE 3 : Résultats des simulations		45
3.1	Introduction	47
3.2	Commande par placement de pôles	47
3.2.1	Placement des pôles appliqué au modèle sans la présence de bruits ni perturbations (cas parfait)	47
3.2.2	Placement des pôles appliqué au modèle en présence de bruits seuls.....	52
3.3	Commande LQR.....	68
3.3.1	Cas parfait (modèle sans perturbation et bruit).....	68
3.3.2	Cas avec bruits	73
3.4	Commande LQG	82
3.4.1	Commande LQG avec observateur de KALMAN.....	82
3.4.2	Commande LQG avec un observateur réduit de KALMAN.....	87
3.5	Conclusion.....	92
CHAPITRE 4 : Etude des performances		95
4.1	Introduction	97
4.2	Fonctions de transfert en boucles fermée du modèle corrigé	97
4.2.1	Commande par placement de pôles.....	97
4.2.2	Commande par placement de pôles utilisée avec un observateur d'état.....	98
4.2.3	Commande LQR	100
4.2.4	Commande LQG.....	102
4.3	Critère de Nyquist	104
4.4	Lieu de Nyquist des fonctions de transfert en boucle ouverte $L(s)$	105
4.4.1	Stabilité de l'asservissement par placement de pôles	106

4.4.2 Stabilité de l'asservissement par placement de pôles en présence de l'observateur d'état.....	108
4.4.3 Stabilité de l'asservissement par la commande LQR.....	111
4.4.4 Stabilité de l'asservissement pour la commande LQG.....	112
4.5 Conclusion :	115
CHAPITRE 5 : Présentation de l'interface du modèle sous MATLAB.....	117
5.1 Introduction	119
5.2 Objets graphiques Matlab	119
5.2.1 Hiérarchie des objets graphiques.....	119
5.2.2 Propriétés des objets graphiques	120
5.3 Programmation des interfaces.....	120
5.4 Interface du modèle ball and beam.....	122
5.5 Conclusion	125
Conclusion générale.....	127
Bibliographie	131
Annexe	137

Table des figures :

Figure 1.1 : la première configuration du système Ball and beam.	17
Figure 1.2 : la deuxième configuration du système Ball and Beam.	18
Figure 1.3 : Représentation du système ball and beam étudié.	20
Figure 1.4 : Représentation des forces appliquées sur la balle.	21
Figure 1.5 : Schéma du moteur à courant continu et son équivalent.	22
Figure 1.6 : Réponse du modèle non linéaire.	27
Figure 1.7 : Modèle non linéaire développé sous Simulink (MATLAB)	28
Figure 1.8 : Réponse du modèle linéaire à 5 variables d'état.	31
Figure 1.9 : Réponse du modèle réduit (la position) en boucle ouverte.	33
Figure 2.1 : Schéma de la commande par placement de pôles.	38
Figure 2.2 : Schéma de l'observateur d'état.	39
Figure 2.3 : Schéma de la commande LQG.	43
Figure 3.1 : Schéma de la commande par placement de pôles appliquée au système parfait. .	50
Figure 3.2 : Réponse du modèle à un échelon après application de la commande par placement de pôles.	51
Figure 3.3 : Représentation du signal de commande u	52
Figure 3.4 : Réponse du modèle à la commande par placement de pôles.	53
Figure 3.5 : Représentation du signal de commande u	53
Figure 3.6 : Représentation d'un bruit blanc.	54
Figure 3.7 : Réponse indicielle du modèle à la commande par placement de pôles en présence des bruits.	55
Figure 3.8 : Représentation du signal de commande u bruité.	56
Figure 3.9 : Schéma de commande par placement de pôles appliquée au système en présence des bruits.	57
Figure 3.10 : Réponse du modèle à la commande par placement de pôles en présence des bruits.	58
Figure 3.11 : Représentation du signal de commande u bruité.	58
Figure 3.12 : Signal de la perturbation appliquée au modèle.	59
Figure 3.13 : Schéma de la commande par placement de pôle appliquée au modèle en présence des perturbations.	60
Figure 3.14 : Réponse du modèle à la commande par placement de pôles en présence de la perturbation.	61

Figure 3.15 : Signal de commande u en présence de perturbation.	62
Figure 3.16 : Simulation de la sortie en présence de la perturbation.	63
Figure 3.17 : Simulation de la commande u délivrée au moteur en présence de perturbation.	63
Figure 3.18 : Représentation du signal de commande u	65
Figure 3.19 : Réponse du modèle à la commande par placement de pôles.	66
Figure 3.20 : Représentation du signal de commande u	67
Figure 3.21 : Réponse du modèle à la commande par placement de pôles.	67
Figure 3.22 : Schéma de commande par placement de pôles en présence d'un observateur d'état.	69
Figure 3.23 : La réponse du modèle pour la commande LQR.....	70
Figure 3.24 : Signal de commande u après l'application de la commande LQR.....	71
Figure 3.25 : Schéma de la commande LQR appliquée au système.	72
Figure 3.26 : Signal de commande après l'application de la commande LQR.....	73
Figure 3.27 : Réponse du modèle à la commande LQR.....	74
Figure 3.28 : Signal de commande après l'application de la commande LQR.....	75
Figure 3.29 : Réponse du modèle à la commande LQR.....	76
Figure 3.30 : Réponse du modèle à la commande LQR.....	77
Figure 3.31 : Signal de commande après l'application de la commande LQR.	77
Figure 3.32: Système de commande LQR en présence de bruits.	79
Figure 3.33 : Signal de commande u	80
Figure 3.34 : Réponse du modèle après application de la commande LQR en présence de perturbation.	81
Figure 3.35 : Réponse du modèle après application de la commande LQR en présence de perturbation.	82
Figure 3.36 : signal de commande u	83
Figure 3.37 : Schéma de la commande LQG appliquée au système.	85
Figure 3.38 : Réponse du modèle à la commande LQG.	86
Figure 3.39 : Signal de commande u après application de la commande LQG.....	86
Figure 3.40 : Réponse du système à la commande LQG.	88
Figure 3.41 : Signal de commande u après application de la commande LQG.....	88
Figure 3.42 : Schéma de la commande LQG avec un observateur de KALMAN réduit.....	90
Figure 3.43 : Réponse du système à la commande LQG avec observateur réduit.....	91
Figure 3.44 : Signal de commande u après l'application de la commande LQG avec observateur réduit.	92

Figure 3.45 : Réponse du système à la commande LQG avec observateur réduit.....	93
Figure 3.46 : Signal de commande u après application de la commande LQG avec un observateur réduit.....	94
Figure 4.1 : Sens trigonométrique.....	105
Figure 4.2 : Lieu de Nyquist de la fonction de transfert $1 + L1(s)$	107
Figure 4.3 : Lieu de Nyquist de la fonction de transfert $1 + L2(s)$	108
Figure 4.4 : Lieu de Nyquist de la fonction de transfert $1 + L3(s)$	109
Figure 4.5 : Lieu de Nyquist de la fonction de transfert $1 + L4(s)$	110
Figure 4.6 : Lieu de Nyquist de la fonction de transfert $1 + L5(s)$	111
Figure 4.7 : Lieu de Nyquist de la fonction de transfert $1 + L6(s)$	112
Figure 4.8 : Lieu de Nyquist de la fonction de transfert $1 + L7(s)$	113
Figure 4.9 : Lieu de Nyquist de la fonction de transfert $1 + L8(s)$	115
Figure 5.1 : Hiérarchie des objets graphiques sous Matlab.....	119
Figure 5.2 : Fenêtre de l'outil GUIDE	120
Figure 5.3 : l'espace de programmation GUI.....	121
Figure 5.4 : Intreface du modèle Ball and Beam.	125

Liste des symboles

- A : Matrice d'état. (Matrice de dynamique) $A \in \mathbb{R}^{n \times n}$
- B : Matrice de commande $B \in \mathbb{R}^{n \times p}$
- b : les frottements entre la balle et la barre
- C : Matrice d'observation. $C \in \mathbb{R}^{q \times n}$
- Co : Matrice de commandabilité
- D : Matrice d'action directe $D \in \mathbb{R}^{q \times p}$
- F_{ex} : Les forces extérieures appliquées à la balle
- g : Gravité terrestre
- I : Courant parcourant le moteur à courant continu
- J_{bl} : Moment d'inertie de la balle
- J_{bm} : Moment d'inertie de la barre
- K_p : Retour d'état pour le placement de pôles
- K_c : Retour d'état pour la commande LQR
- K_f : Gain du filtre de KALMAN
- K_l : Gain de l'observateur
- K : Constante de la f.e.m
- K_m : Constante électromagnétique
- l : Longueur de la barre
- L : L'inductance du moteur à courant continu
- M_{bl} : Masse de la balle
- N : Précompensateur
- Ob : Matrice d'observabilité
- P_c : Solution de Riccati correspondante au gain K_c
- P_f : Solution de l'équation de Riccati correspondante au gain K_f
- P : Vecteur des pôles du système.
- Q : Matrice de pondération des états
- R : Matrice de pondération de l'entrée
- R_e : Résistance du moteur à courant continu

R_{bl} : Rayon de la balle
 r_0 : Nouvelle consigne
 r : Distance parcourue par la balle en partant de l'origine
 t_r : Temps de réponse
 u : L'entrée du système et la tension appliquée au moteur $u \in \mathbb{R}^p$
 V : Espérance mathématique du bruit blanc v
 W : Espérance mathématique du bruit blanc w
 w, v : Bruit blanc
 x : Vecteur d'état. $x \in \mathbb{R}^n$
 $\hat{x}$: Vecteur des estimée du vecteur d'état x
 y : Vecteur des sorties $y \in \mathbb{R}^q$
 $\hat{y}$: Vecteur des estimée du vecteur de sortie y
 α : L'angle correspondant au roulement de la balle
 θ : L'angle θ correspondant à l'inclinaison de la barre avec l'horizontale
 ρ : Distance entre le centre de la balle et la partie supérieure de la barre

Introduction générale :

Introduction générale

Un système mécanique est un ensemble d'objets massiques en mouvement ou à l'équilibre inter réagissant entre eux selon des principes ou règles afférant à la physique, dont la cinématique, statique, dynamique, thermodynamique, etc. font partie...

Dans notre projet nous allons nous intéresser à l'étude d'un système mécanique bien défini.

Il s'agit du système dit « *Ball and Beam* » qui est composé d'une poutre (barre) sur laquelle roule une balle. Il constitue un procédé pilote (benchmark) dans plusieurs laboratoires d'automatique.

L'objectif de cette étude est de stabiliser le système même en présence des bruits et perturbations.

La description détaillée, les exemples pratiques et l'objectif de commande seront détaillés dans le premier chapitre. Nous modéliserons le système en utilisant toutes les lois physiques, cinématiques... Comme la plupart des systèmes mécaniques sont non linéaires, dont celui-ci, la linéarisation autour d'un point d'équilibre est nécessaire ainsi que l'étude de la commandabilité et l'observabilité du système afin de pouvoir appliquer des commandes linéaires.

Dans notre projet, nous nous sommes intéressés aux commandes par retour d'état.

La commande par placement de pôles, la commande LQR et la commande LQG seront présentées dans le deuxième chapitre. La commande robuste LQG/LTR sera aussi présentée ainsi que quelques observateurs d'état.

Le but principal du projet est de comparer les résultats et les performances obtenus pour chaque commande, cela sera détaillé dans le troisième et quatrième chapitres.

Le cinquième chapitre sera consacré à la présentation de l'interface du modèle sous le logiciel MATLAB.

CHAPITRE 1 : Présentation et modélisation du système.

1.1 Introduction

Le système « ball and beam » est un système mécanique composé d'une barre (poutre) sur laquelle glisse une bille. C'est un système qui possède un comportement non linéaire.

Il constitue un procédé pilote (benchmark) dans plusieurs laboratoires d'automatique.

1.2 Présentation du système

1.2.1 Différentes configurations du système « Ball and beam »

Il existe deux configurations pour le système « ball and beam » [Wan]:

La première configuration illustre le fait que la barre est prise en charge par son milieu par un moteur à courant continu, la balle tourne autour de l'axe central de la barre comme le démontre la figure 1.1, la plupart des systèmes « ball and beam utilisent ce type de configuration. L'avantage de cette configuration est qu'elle est facile à construire, et le modèle mathématique est relativement simple.[Well]

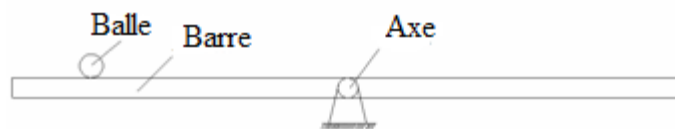


Figure 1.1 : la première configuration du système Ball and beam.

La deuxième configuration illustrée dans la figure 1.2 fait part d'une barre rattachée par ses deux extrémités par des bras l'un d'eux fixe et l'autre est reliée à un moteur qui fait balancer la balle sur la barre.

L'inconvénient de cette configuration est qu'on doit prendre en considération les différentes parties mécaniques ce qui peut ajouter de la complexité au modèle mathématique. Son avantage par contre est qu'un moteur relativement modeste suffit en raison de l'effet du levier.

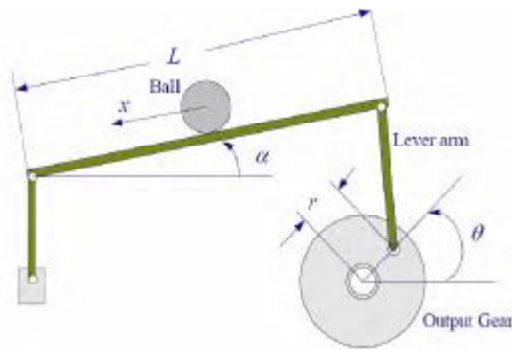


Figure 1.2 : la deuxième configuration du système Ball and Beam.

1.2.2 Objectif de commande

L'étude de ce système est d'autant plus intéressante vu sa simplicité et son authenticité. Sachant que c'est un système non linéaire, il nous permet de mieux nous familiariser avec les différentes méthodes de linéarisation, et de son instabilité en boucle ouverte il nous permet de mieux étudier les commandes permettant de le stabiliser et au final de comparer entre les performances de chacune d'elles.

La tâche principale est de trouver un régulateur pour commander le moteur à courant continu, ainsi le couple produit des signaux permettant de faire tourner la barre avec un certain angle d'inclinaison afin de faire déplacer la balle à la position désirée tout en rejetant les effets indésirables des perturbations et des bruits. Le signal de commande peut être dérivé en retournant l'information de position de la balle.

La propriété non linéaire n'est pas significative quand la barre ne s'incline que d'un petit angle par rapport à l'horizontale. Dans ce cas-ci, il est possible de linéariser le système autour d'un point d'équilibre. Le problème de la linéarité commencera à poser problème dans le cas où l'angle d'inclinaison de la barre dépassera les $+ ou - 30^\circ$, alors dans ce cas de figure une technique plus avancée de commande telle que la commande non linéaire est à prévoir.

Il est judicieux d'introduire une saturation afin de protéger le moteur à courant continu des surcharges qui peuvent être fatales. Les valeurs considérées dans ce rapport pour la saturation

sont $\pm 20 \text{ volts}$ sachant que le moteur étudié supporte un intervalle de tension allant de -24 volts à $+24 \text{ volts}$.

1.2.3 Exemples pratiques analogiques au système

On peut assimiler par analogie plusieurs procédés physiques au système étudié, parmi eux nous pouvons citer les processus suivant :

- Les réactions chimiques exothermiques :

Si une réaction chimique produit de la chaleur et que la production augmente au fur et à mesure que la chaleur augmente, alors dans ce cas si on doit employer une commande permettant de stabiliser la température de cette réaction pour éviter une réaction 'd'emballement'. [Well]

- Le décollage/atterrissage vertical des avions type VTOL (Vertical Take-Off and Landing aircraft). L'acronyme français correspondant est ADAV (Aéronef à Décollage et Atterrissage Verticaux) :

L'aéronef décolle à la verticale comme une fusée, bascule à l'horizontale pour le vol normal, puis rebasculé à la verticale pour l'atterrissage. On doit employer une commande afin de le stabiliser durant les phases de décollage et d'atterrissage.

La condition d'existence d'un tel aéronef est, bien entendu, que la force de propulsion de ce moteur dépasse son poids total, pendant toute la durée du vol ascensionnel.[Wiki2]

1.3 Modélisation du système

La modélisation regroupe un ensemble de techniques permettant de disposer d'une représentation mathématique du système à étudier.

Cette étape est importante : la réalisation d'une commande nécessite un modèle adapté, issu d'un compromis entre la complexité et la représentativité du système.

1.3.1 Description physique

La configuration choisie pour être étudiée est la première configuration qui a été présentée dans un paragraphe précédent (section 1.2.1)

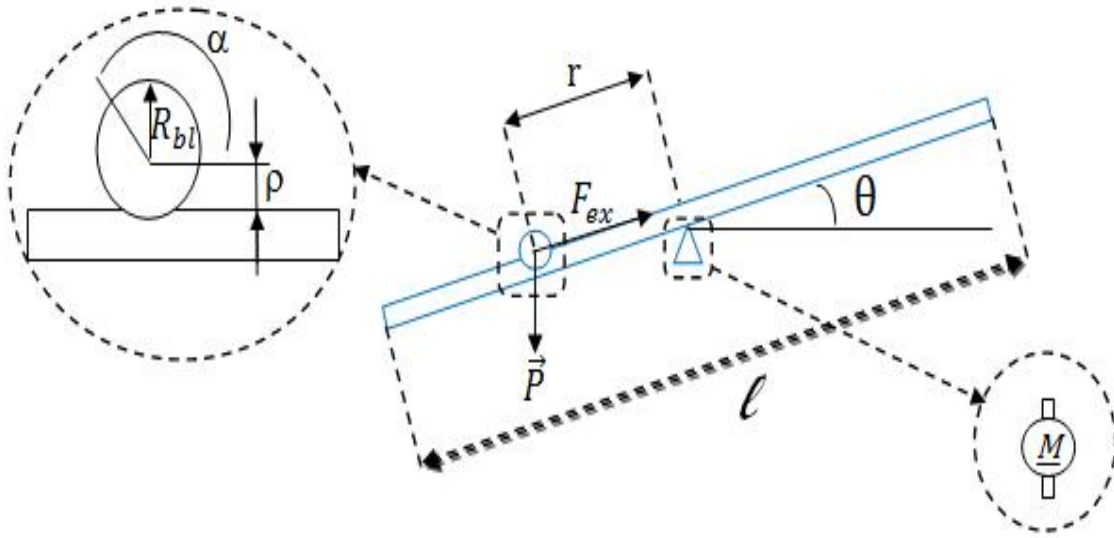


Figure 1.3 : Représentation du système ball and beam étudié.

1.3.2 Modélisation du mouvement de la balle

D'après le principe fondamental de la dynamique décrit par l'équation 1.1

$$\sum F = m * \gamma \quad (1.1)$$

Et d'après la figure 1.4 on obtient l'équation suivante :

$$M_{bl} * \gamma = M_{bl} * g * \sin\theta - F_{ex} - M_{bl} * r * (\dot{\theta})^2 \quad (1.2)$$

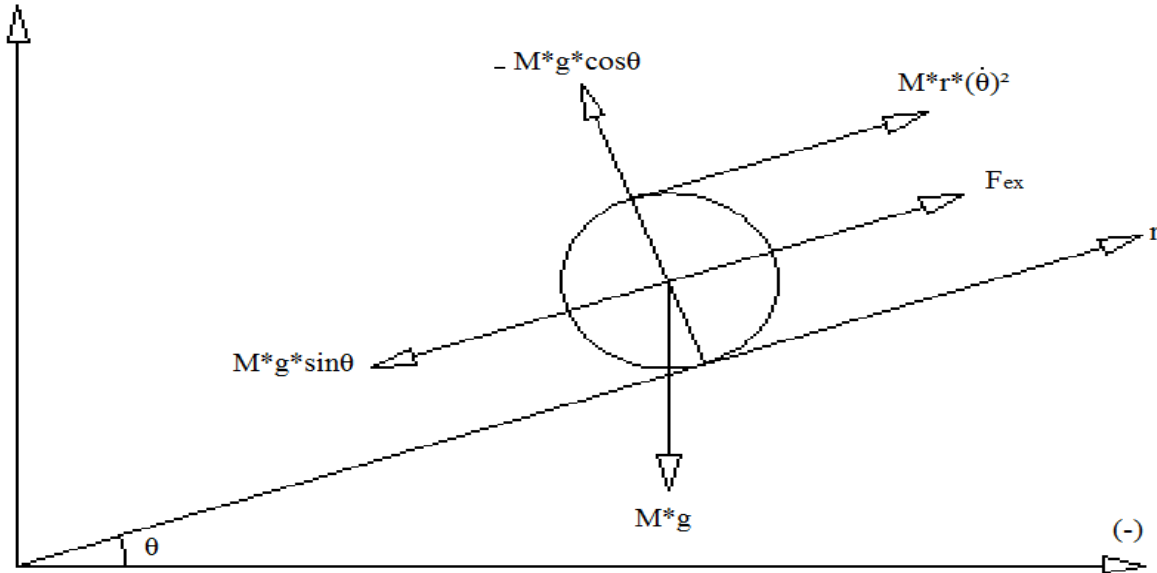


Figure 1.4 : Représentation des forces appliquées sur la balle.

Avec:

M_{bl} : Masse de la balle

γ : L'accélération

g : La gravitation terrestre

θ : L'angle que forme la barre avec l'horizontale

F_{ex} : L'ensemble des forces externes appliquées à la balle

v : La vitesse de la balle

On a les relations suivantes entre les différentes variables vues précédemment:

$$\gamma = \ddot{r} \quad (1.3)$$

$$v = \dot{r} \quad (1.4)$$

$$r = a * r \quad (1.5)$$

La somme des couples appliqués sur la balle se traduit par l'équation suivante :

$$\sum C_{bl} = J_{bl} * \gamma = F_{ex} * r \quad (1.6)$$

On a aussi l'expression du moment d'inertie J_{bl} d'une balle pleine de rayon R_{bl} et de masse M_{bl} qui est égale à :

$$J_{bl} = \frac{2}{5} M_{bl} * R_{bl}^2 \quad (1.7)$$

En remplaçant l'équation (1.7) dans la (1.6) on obtient la nouvelle équation:

$$F_{ex} = \frac{2}{5} M_{bl} * \frac{R_{bl}^2}{r} * \ddot{r} \quad (1.8)$$

Avec ρ qui représente la distance entre le centre de la balle et la plus haute extrémité de la barre. (Voir la figure 1.3)

Le remplacement des équations (1.3) (1.8) dans (1.2) nous donne l'équation suivante:

$$\ddot{r} = \frac{1}{(1 + \frac{2}{5} * (\frac{R_{bl}}{r})^2)} (g * \sin \theta - r * (\dot{\theta})^2) \quad (1.9)$$

1.3.3 Modélisation du moteur à courant continu

De la figure 1.5 on déduit l'équation caractérisant le moteur à courant continu :

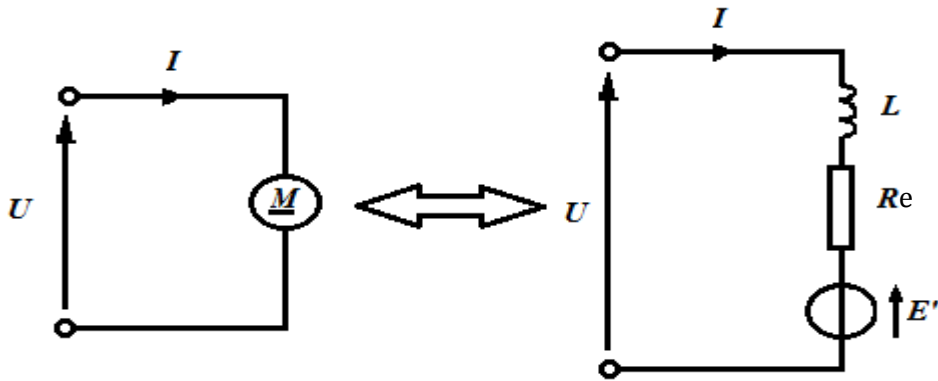


Figure 1.5: Schéma du moteur à courant continu et son équivalent.

CHAPITRE 1 : Présentation et modélisation du système

$$u = E' + R_e * I + L * \frac{dI}{dt} \quad (1.10)$$

Avec :

u : la tension aux bornes du moteur,

E' : la f.e.m. du moteur

R_e : la résistance interne du moteur,

I : le courant traversant le moteur,

L : l'inductance du moteur.

On a aussi l'expression de la f.e.m. du moteur qui peut s'écrire comme suit

$$E' = K_m * \dot{\theta} \quad (1.11)$$

Avec :

K_m : constante électromagnétique.

Ainsi, en remplaçant l'équation 1.11 dans l'équation 1.10 on obtient l'expression suivante :

$$\dot{I} = -\frac{R_e}{L} * I + \frac{1}{L} * u - \frac{K_m}{L} * \dot{\theta} \quad (1.12)$$

1.3.4 Modélisation des couples appliqués sur la barre

Pour procéder à la modélisation de la barre, nous allons utiliser les équations des couples agissants sur celle-ci par le moteur à courant continu. A cet effet, nous utiliserons le principe fondamental de la dynamique [Wan]:

$$J \frac{d\omega}{dt} = Cm - Cr \quad (1.13)$$

En remplaçant les expressions du couple moteur Cm et du couple résistant Cr on parvient à obtenir l'expression suivante :

CHAPITRE 1 : Présentation et modélisation du système

$$J_m * \ddot{\theta} = K * I - b * \dot{\theta} - C_{charge} \quad (1.14)$$

$$C_{charge} = C_{bm} - C_{bl} \quad (1.15)$$

Avec :

J_m : Le moment d'inertie du moteur.

K : Constante du moteur.

b : Coefficient de frottement.

C_{charge} : Couple engendré par la charge.

C_{bm} : Couple de la barre.

C_{bl} : Couple de la balle.

Nous avons aussi les équations suivantes :

$$C_{bm} = J_{bm} * \ddot{\theta} \quad (1.16)$$

Avec :

J_{bm} : Moment d'inertie de la barre

$$C_{bl} = F * r = - r * M_{bl} * g * \cos \theta \quad (1.17)$$

En remplaçant ces dernières expressions dans l'équation (1.14) nous obtenons finalement:

$$(J_m + J_{bm}) * \ddot{\theta} = K * I - b * \dot{\theta} - r * M_{bl} * g * \cos \theta \quad (1.18)$$

1.3.5 Modèle d'état non linéaire global

Par les équations (1.9), (1.12) et (1.18), nous constituons le modèle non linéaire du système ball and beam :

$$\begin{cases} \ddot{r} = \frac{1}{\left(1 + \frac{2}{5} * \left(\frac{R_{bl}l}{r}\right)^2\right)} (g * \sin \theta - r * (\dot{\theta})^2) \\ \dot{I} = -\frac{R_e}{L} * I + \frac{1}{L} * u - \frac{K_m}{L} * \dot{\theta} \\ \ddot{\theta} = \frac{K}{J_m + J_{bm}} I - \frac{b}{J_m + J_{bm}} \dot{\theta} - \frac{M_{bl} * g}{J_m + J_{bm}} r * \cos \theta \end{cases} \quad (1.19)$$

Nous posons le vecteur d'état suivant :

$$x^T = [x_1 \ x_2 \ x_3 \ x_4 \ x_5] = [r \ \dot{r} \ \theta \ \dot{\theta} \ I] \quad (1.20)$$

Le modèle d'état s'écrit donc sous la forme :

$$\begin{cases} \dot{x} = F(x) + g(x)u(t) \\ y = C x(t) \end{cases} \quad (1.21)$$

Avec :

$$F(x) = \begin{bmatrix} x_2 \\ \frac{g}{\left(1 + \frac{2}{5} * \left(\frac{R_{bl}}{r}\right)^2\right)} \sin(x_3) - \frac{1}{\left(1 + \frac{2}{5} * \left(\frac{R_{bl}}{r}\right)^2\right)} x_1 * x_4^2 \\ x_4 \\ \frac{K}{J_m + J_{bm}} x_5 - \frac{b}{J_m + J_{bm}} x_4 - \frac{M_{bl} * g}{J_m + J_{bm}} x_1 * \cos(x_3) \\ -\frac{R_e}{L} * x_5 - \frac{K_m}{L} * x_4 \end{bmatrix} \quad (1.22)$$

$$g(x) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} \quad (1.23)$$

Les sorties considérées sont r et θ .

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (1.24)$$

Pour toutes les simulations qui suivront, dans ce rapport, nous utiliserons les valeurs numériques données dans le tableau récapitulatif (**1.1**) suivant [Wan]:

CHAPITRE 1 : Présentation et modélisation du système

Paramètres	Symbole	Unité de mesure	Valeur
L'angle de la barre	θ	<i>rad</i>	$(-\frac{\pi}{6}, \frac{\pi}{6})$
Masse de la balle	M_{bl}	<i>Kg</i>	0.0327
Résistance du moteur à courant continu	R_e	<i>Ohms</i>	4.7
Rayon de la balle	R_{bl}	<i>m</i>	0.01
L'inductance électrique	<i>L</i>	<i>H</i>	0.0016
Constante de la f.é.m.	<i>K</i>	<i>Nm/A</i>	4.91
Tension appliquée	<i>u</i>	<i>Volt</i>	$(-24, 24)$
Position de la balle	<i>r</i>	<i>m</i>	$(-0.35, 0.35)$
Facteur d'amortissement du système mécanique	<i>b</i>	<i>Nm/ (rad/s)</i>	1.5279
Accélération de la pesanteur	<i>g</i>	<i>m/s²</i>	9.81
Langueur de la barre	<i>l</i>	<i>m</i>	0.7
Masse de la barre	M_{bm}	<i>Kg</i>	0.381
Moment d'inertie de la barre	J_{bm}	<i>Kg.m²</i>	0.019
Moment d'inertie du moteur	J_m	<i>Kg.m²</i>	0.043
Rayon de rotation de la balle	<i>r</i>	<i>m</i>	0.005
Constante électromagnétique	K_m	<i>Volts/ (rad/s)</i>	4.77

Tableau 1.1: Les valeurs utilisées dans les simulations.

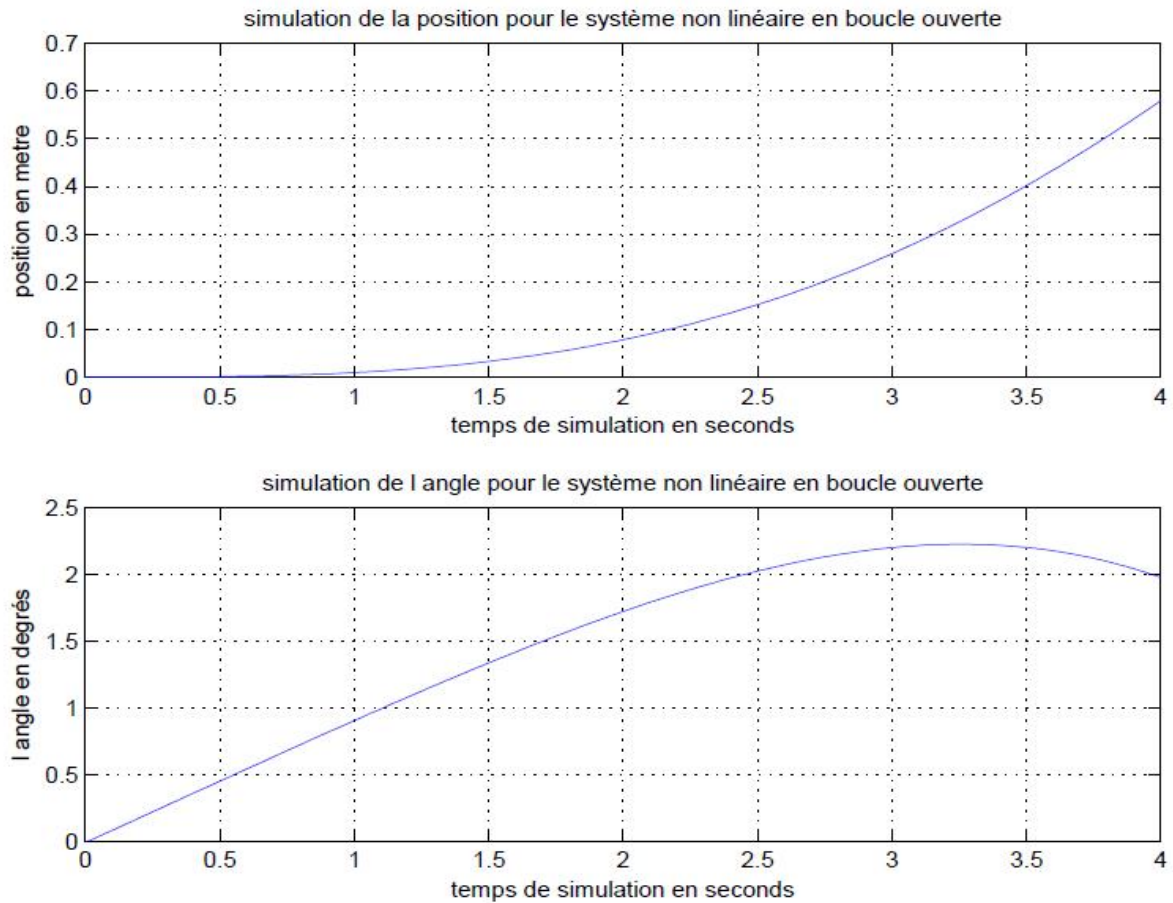


Figure 1.6 : Réponse du modèle non linéaire.

La réponse indicielle du système non linéaire à un échelon d'amplitude 0.1 volts représentée dans la figure 1.6, et sans conditions initiales, démontre bien que le système est fortement instable vu que normalement il ne devra pas dépasser la valeur max de 0.35 qui équivaut à la limite (positive) de la barre sur laquelle la balle roule. En d'autres termes la balle est tombée de la barre à l'instant 3.25s (environ)

Dans la figure 1.7 nous donnons le schéma du modèle non linéaire réalisé sous Matlab Simulink

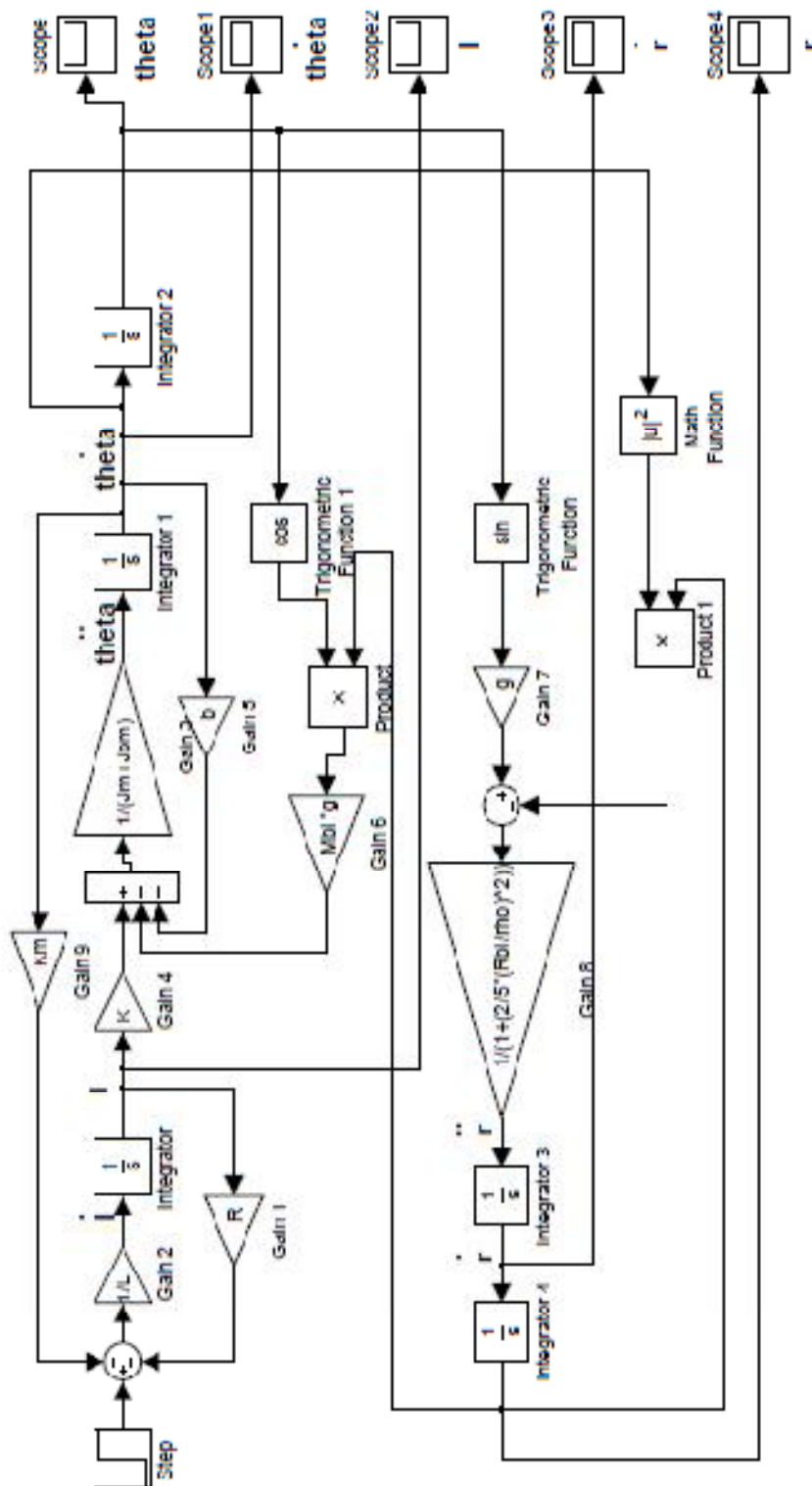


Figure 1.7 : Modèle non linéaire développé sous Simulink (MATLAB)

1.4 Modèle linéaire et propriétés

1.4.1 Linéarisation autour d'un point d'équilibre

La linéarisation consiste à obtenir un modèle linéaire autour d'un point d'équilibre. Les points d'équilibre x_e sont solution de l'équation (1.25)

$$\begin{cases} \dot{x}_e = 0 \\ u_e = 0 \end{cases} \quad (1.25)$$

La résolution de l'équation (1.25) avec (1.21) donne deux points d'équilibre :

- L'origine :

$$\begin{cases} x_{1e} = 0 \\ x_{2e} = 0 \\ x_{3e} = 0 \\ x_{4e} = 0 \\ x_{5e} = 0 \end{cases} \quad (1.26)$$

- Le point :

$$\begin{cases} x_{1e} = 0 \\ x_{2e} = 0 \\ x_{3e} = \pi \\ x_{4e} = 0 \\ x_{5e} = 0 \end{cases} \quad (1.27)$$

Nous considérons le premier point d'équilibre qui est à l'origine. Le modèle d'état linéaire s'écrit sous forme:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases} \quad (1.28)$$

Avec

$$A = \left. \frac{d\dot{x}}{dx} \right|_{x=0} \quad (1.29)$$

$$B = \left. \frac{d\dot{x}}{du} \right|_{x=0} \quad (1.30)$$

Le modèle d'état linéaire s'écrit alors :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{g}{1 + \frac{2}{5} \left(\frac{Rbl}{\rho} \right)^2} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -\frac{Mbl * g}{Jm + Jbm} & 0 & 0 & -\frac{b}{Jm + Jbm} & -\frac{K}{Jm + Jbm} \\ 0 & 0 & 0 & -\frac{Km}{L} & -\frac{Re}{L} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} U \quad (1.31)$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} X$$

Cela peut être vérifié car pour un mouvement décrit avec des angles très petits (autour de zéro) on obtient les relations suivantes:

$$\sin(x_3) \approx x_3 \quad (1.32)$$

Et

$$\cos(x_3) \approx 1 \quad (1.33)$$

Ce qui coïncide bien avec les résultats trouvés pour l'équation (1.26).

La réponse du modèle linéaire pour une entrée échelon d'amplitude 0.1 volts est comme indiqué par la figure (1.8) suivante :

CHAPITRE 1 : Présentation et modélisation du système

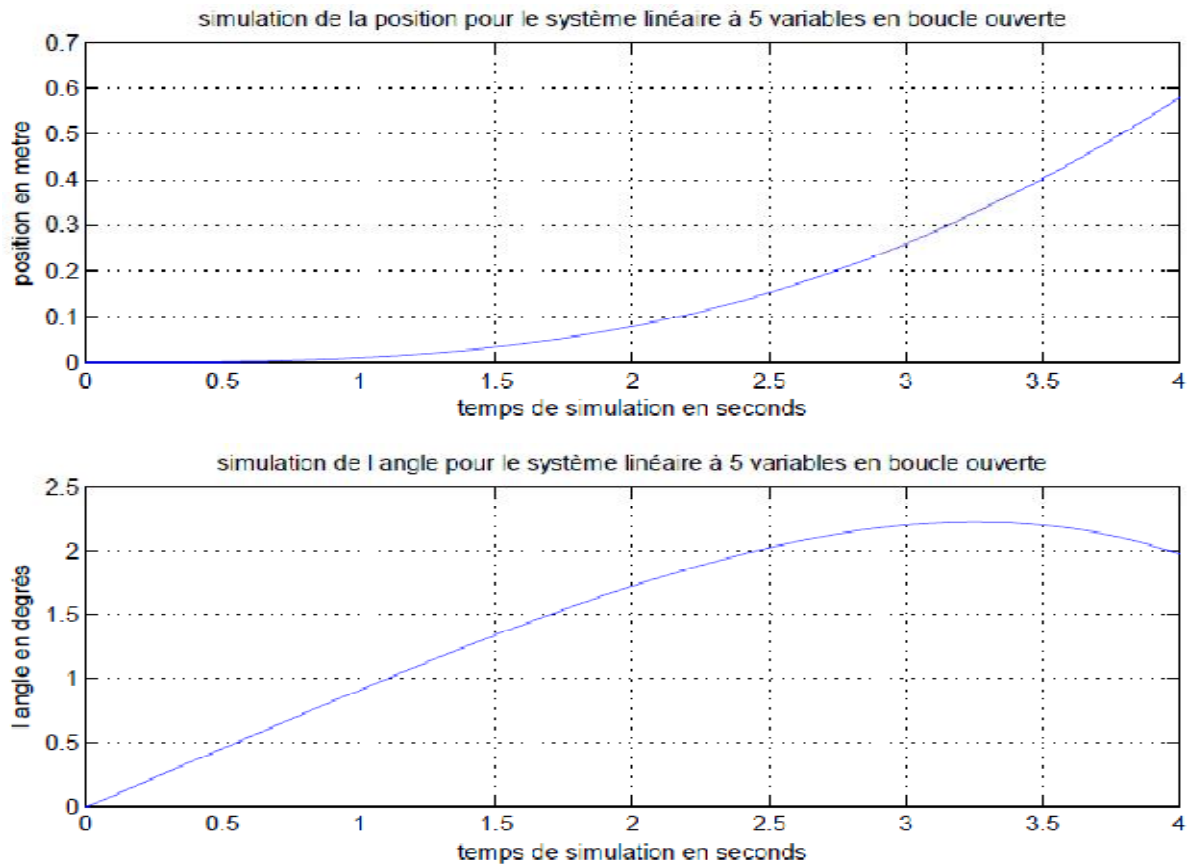


Figure 1.8 : Réponse du modèle linéaire à 5 variables d'état.

Nous remarquons en comparant les figure 1.8 et 1.6 que la réponse est identique pour les valeurs proches de l'origine.

Les valeurs propres du modèle sont :

P_bo=

1.0e+003 *
-2.8541
-0.1081
-0.0006
0.0003 + 0.0005i
0.0003 - 0.0005i

Le système en boucle ouverte possède deux pôles instables.

1.4.2 Test de commandabilité et observabilité

Le test de commandabilité et observabilité va nous permettre de savoir si on peut implémenter une commande par retour d'état.

Pour cela nous devons nous assurer que le système est commandable et observable.

Ceci sera calculé en utilisant les commandes de MATLAB :

ctrb : Pour le calcul de la matrice de commandabilité C_o ;

obsv : Pour calculer la matrice d'observabilité O_b ;

rank : Afin de calculer le rang des matrices de commandabilité et observabilité.

```
>>rang_Ob =  
5
```

```
>>rang_Co =  
2
```

1.4.3 Réduction de l'ordre du système

Ainsi nous remarquons que le modèle n'est pas commandable, ce qui nous posera problème pour l'application des commandes. Pour y remédier nous allons simplifier le système en supposant que l'inductance L , étant très petite, comme nulle ainsi on obtiendra l'équation suivante [Wan]:

$$u = K_m \cdot \dot{\theta} + R_e \cdot I \quad (1.34)$$

Après extraction du terme I du courant du moteur nous obtenons l'expression suivante :

$$I = \frac{1}{R_e} \cdot u - \frac{K_m}{R_e} \cdot \dot{\theta} \quad (1.35)$$

CHAPITRE 1 : Présentation et modélisation du système

En remplaçant cette dernière équation dans le modèle d'état défini par l'équation (1.31) on obtient le nouveau système :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{g}{1 + \frac{2}{5} \left(\frac{Rbl}{\rho} \right)^2} & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{Mbl g}{Jm + Jbm} & 0 & 0 & -\left(\frac{K Km}{Re} + b \right) \frac{1}{Jm + Jbm} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{K}{Re(Jm + Jbm)} \end{bmatrix} V \quad (1.36)$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} X$$

L'instabilité du système même après la réduction de son ordre va persister, ce qui va être représenté dans les figures 1.9 suivantes :

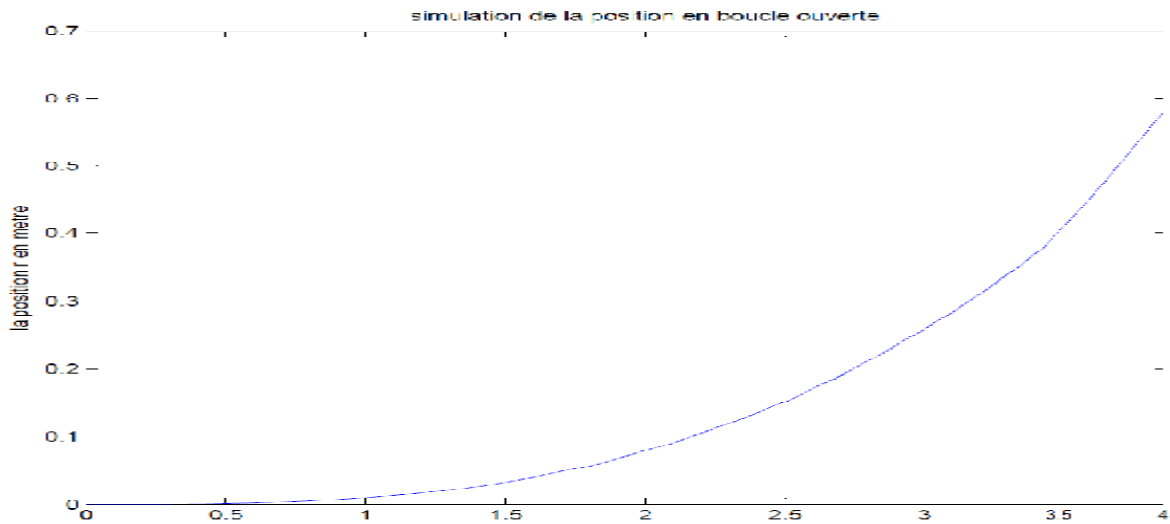


Figure 1.9: Réponse du modèle réduit (la position) en boucle ouverte.

Les valeurs propres du modèle en boucle ouverte sont :

P_Bo =

1.0e+002 *

-1.0502

-0.0057

0.0029 + 0.0049i

0.0029 - 0.0049i

Nous remarquons bien que le système reste instable.

Le teste de la commandabilité et l'observabilité du nouveau système nous donne les résultats suivant :

```
>> rang_ Ob
```

```
rang_ Ob =
```

```
4
```

```
>> rang_ Co
```

```
rang_ Co =
```

```
4
```

```
>>rang_A
```

```
Rang_A=
```

```
4
```

Dans ce cas là le système est commandable et observable et donc l'application des commandes basées sur le retour d'état est possible.

1.5 Conclusion

Le système étudié est instable. L'objectif de commande est de stabiliser la balle sur une position donnée.

Nous allons utiliser le modèle linéaire réduit pour développer des commandes stabilisantes par retour d'état.

CHAPITRE 2 : Etude théorique des commandes développées

L'obtention des performances désirées nous conduit à devoir insérer un correcteur au système. Plusieurs types de celui-ci ont été mis au point, chacun d'eux, avec ses propres avantages et inconvénients, donne des résultats avec un certain degré de complexité. Nous pouvons citer entre autre les correcteurs PID, RST, tout ou rien...

Dans ce chapitre nous ferons l'étude de quelques correcteurs à retour d'état (Placement de pôles, LQR, et LQG). La réalisation de ce type de commande nécessite la disposition des variables d'état aux mesures. Ceci peut être un handicap du fait qu'on ne peut pas toujours avoir accès à certaines des variables d'état, ou bien que le nombre de capteurs utilisés est très grand et donc très coûteux, ou encore que l'insertion d'un capteur peut modifier le fonctionnement du système. Pour palier à cet inconvénient, on utilise des observateurs d'état. Pour cela on étudiera aussi différents types d'observateurs d'état.

2.1 Placement de pôles

Soit l'équation d'état du système étudié :

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \quad (2.1)$$

Où $x \in \mathbb{R}^n$ représente l'état du système, $u(t) \in \mathbb{R}^p$ est la commande et $y(t) \in \mathbb{R}^q$ est la sortie mesurée du système. Le vecteur $D = 0$ dans notre cas.

La méthode du placement de pôles consiste à choisir la matrice de retour d'état $K_p \in \mathbb{R}^{m \times n}$, constante, de telle sorte que les valeurs propres $\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_n$ du système en boucle fermée, prennent des emplacements donnés à l'avance dans le plan complexe.

Le retour d'état s'écrit comme suit :

$$u(t) = -K_p x + r_0(t) \quad (2.2)$$

En remplaçant l'équation (2.2) dans le système d'équation (2.1) on obtiendra le nouveau système suivant :

CHAPITRE 2 : Etude théorique des commandes développées

$$\begin{cases} \dot{x} = (A - B K_p)x + B r_0(t) \\ y = Cx \end{cases} \quad (2.3)$$

La fonction de transfert du système en boucle fermée s'écrit [Gou] :

$$G_{bf}(s) = C(sI - A + B K_p)^{-1} B N \quad (2.4)$$

Avec :

$$r_0(t) = N y_{ref}(t) \quad (2.5)$$

N : étant le précompensateur.

Supposons que $y_{ref}(t) = y_{ref}$ soit une constante.

Pour que $\lim_{t \rightarrow \infty} y(t) = y_{ref}$ il est donc nécessaire que $G_{bf}(0) = 1$ (gain statique en boucle fermée), c'est-à-dire

$$N = \text{inv}(C(-A + B K_p)^{-1} B) \quad (2.6)$$

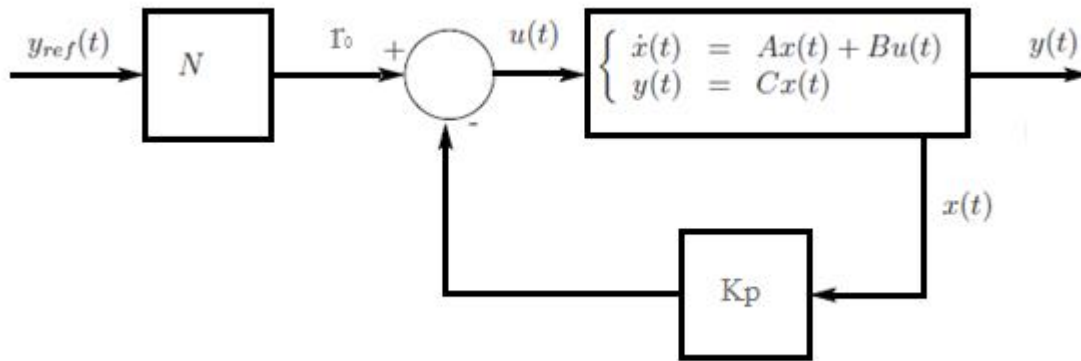


Figure 2.1 : Schéma de la commande par placement de pôles.

2.2 Observateur d'état

Soit le système d'équation linéaire (2.1).

La mise en œuvre de la commande par retour d'état $u(x)$ a besoin de capteurs permettant de donner à chaque instant t une valeur approximative de l'état $x(t)$. Deux types de capteurs de natures différentes mais équivalentes sont utilisées. Des capteurs physiques provenant de

CHAPITRE 2 : Etude théorique des commandes développées

l'instrumentation; Ces capteurs sont parfois trop coûteux ou difficiles à réaliser pour des raisons techniques. Pour cette raison, nous nous orientons vers Les capteurs logiciels dits aussi *Observateurs*; Ces derniers sont des algorithmes basés sur un modèle du système et utilise une information pertinente donnée par des capteurs physiques. Ces capteurs logiciels délivrent à chaque instant t une estimation en ligne des variables d'état non mesurées du système.[Wiki]

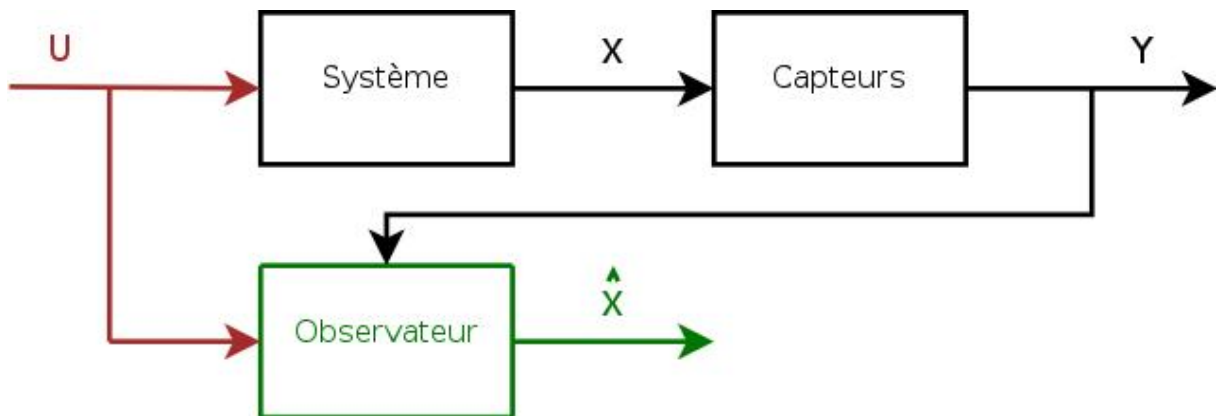


Figure 2.2: Schéma de l'observateur d'état.

Un observateur dynamique a la forme suivante :

$$\begin{cases} \dot{\hat{x}} = A\hat{x} + Bu + K_l(y - \hat{y}) \\ \hat{y} = C\hat{x} \end{cases} \quad (2.7)$$

On corrige l'évolution de l'état grâce au modèle en fonction de l'écart constaté entre la sortie observée et la sortie reconstruite par l'observateur : $(y - \hat{y})$

L'observateur d'état peut se réécrire comme suit :

$$\dot{\hat{x}} = (A - K_l C)\hat{x} + Bu + K_l y \quad (2.8)$$

2.3 Commande LQR (Linear Quadratic Regulator)

Soit le système linéaire étudié (2.1).

CHAPITRE 2 : Etude théorique des commandes développées

L'hypothèse suivante est faite :

La paire (A,B) est stabilisable, c'est-à-dire qu'il n'y a pas de modes instables et ingouvernables dans le système.

La synthèse Linéaire Quadratique dénommée LQ ou LQR (*Linear Quadratic Regulator*) consiste en la recherche d'un vecteur gain K_c telle que la commande par retour d'état :

$$u(t) = -K_c * x(t) + r_0(t) \quad (2.9)$$

Stabilise le système et minimise le critère quadratique suivant :

$$J = \frac{1}{2} \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (2.10)$$

Où les matrices de pondérations Q et R satisfont :

$$Q = Q^T \geq 0 \quad (2.11)$$

$$R = R^T > 0 \quad (2.12)$$

On a aussi le vecteur K_c qui s'écrit comme suit :

$$K_c = R^{-1} B^T P_c \quad (2.13)$$

Avec P_c qui obéit à l'équation algébrique de Riccati :

$$P_c A + A^T P_c - P_c B R^{-1} B^T P_c + Q = 0 \quad (2.14)$$

L'obtention du gain K_c passe donc par la recherche de la solution P_c symétrique positive de l'équation de Riccati, qui rapportée dans l'équation (2.13) elle fournit le vecteur gain K_c [Alaz1] [Alaz2][Berg].

2.4 Commande LQG (Linear Quadratic Gaussian)

La méthode LQR exige la connaissance du vecteur d'état. Dans la majorité des problèmes de commande on ne dispose que d'une connaissance partielle du vecteur d'état. La synthèse LQG consiste donc à rechercher à partir de cette mesure partielle, un régulateur qui minimise un critère quadratique de nature stochastique [Alaz1][Berg].

Soit le système dynamique stochastique d'équation d'état:

$$\begin{cases} \dot{x} = Ax + Bu + Mw \\ y = Cx + v \end{cases} \quad (2.15)$$

Où w et v représentent des bruits blancs, de moyenne nulle, indépendants, avec respectivement pour matrice de covariance W et V .

$$E[w(t)w(t + \tau)^T] = W\delta(\tau) \quad (2.16)$$

$$E[v(t)v(t + \tau)^T] = V\delta(\tau) \quad (2.17)$$

$$E[w(t)v(t + \tau)^T] = 0 \quad (2.18)$$

Avec : $W \geq 0$ et $V > 0$

On note aussi la matrice de covariance du bruit d'état :

$$W_x = MWM^T \quad (2.19)$$

A partir du vecteur y de mesures bruitées (retour de sortie), nous recherchons une loi de commande qui minimise le critère de l'équation suivante **(2.20)**.

$$J(x_0, u) = \lim_{tf \rightarrow \infty} E \left\{ \frac{1}{tf} \int_{t_0}^{tf} (x^T Q x + u^T R u) dt \right\} \quad (2.20)$$

CHAPITRE 2 : Etude théorique des commandes développées

La solution de ce problème s'appuie sur le principe de séparation qui établit que la commande optimale LQG est obtenue [Alaz1][Alaz2] :

- a) En recherchant l'estimé optimale $\hat{x}$ (au sens de la variance d'erreur minimale) de l'état x par la méthode du Filtre de KALMAN, c'est-à-dire on estime l'état x par l'équation classique du filtre de KALMAN à condition que le triplet $(A, MW^{1/2}, C)$ soit détectable et stabilisable.

$$\dot{\hat{x}} = A\hat{x} + Bu + K_f(y - C\hat{x}) \quad (2.21)$$

Avec $K_f = P_f C^T V^{-1}$ où P_f obéit à l'équation de Riccati suivante :

$$P_f A + A^T P_f - P_f B R^{-1} B^T P_f + M W M^T = 0 \quad (2.22)$$

Avec P_f positive symétrique.

- b) En employant cet estimé comme s'il était la mesure exacte du vecteur d'état, pour résoudre le problème de commande optimale linéaire déterministe (méthode LQR) ; soit (si $(A, B, Q^{1/2} N)$ est détectable et stabilisable) :

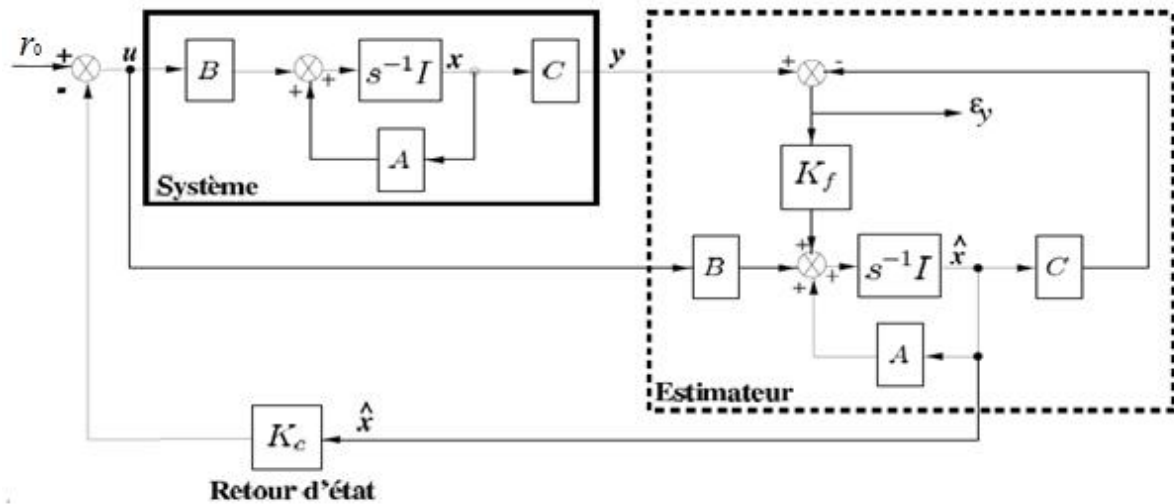


Figure 2.3: Schéma de la commande LQG.

2.5 Commande LQG/LTR (Loop Transfer Recovery)

L'objectif de la procédure de réglage LTR est de restaurer les marges de stabilité "idéale" du retour d'état de la commande LQ (ou duale du filtre de KALMAN) [Alaz2].

La procédure de synthèse LQG/LTR consiste à :

a) Synthétiser, dans une première étape, le correcteur LQ par un choix approprié des pondérations Q_x et R obéissant aux exigences du cahier des charges. Les aspects de cette première synthèse concernent le comportement basse fréquence des valeurs singulières du transfert $-K_c(sI - A)^{-1}B$, les fréquences de coupures correspondantes, l'affaiblissement en hautes fréquences, etc. ...

b) Dans une seconde étape, à partir d'un réglage nominal W_{x_0} et V_0 du filtre de KALMAN, on augmentera le paramètre q du nouveau réglage :

$$W_x = W_{x_0} + qBB^T \quad (2.23)$$

$$V = V_0 \quad (2.24)$$

Jusqu'à ce que le transfert de boucle $K(s)G(s)$ du correcteur LQG recouvre, sur une bande de fréquence suffisamment large, le transfert de boucle de retour d'état LQ :

$$\lim_{q \rightarrow \infty} K(s)G(s) = -K_c(sI - A)^{-1}B \quad (2.25)$$

Cela se fait sous les hypothèses suivantes :

- $M = B$ (Bruit d'état dans la direction de l'entrée).
- $C(sI - A)^{-1}B$ à minimum de phase.
- $m = p$ (nombre égal d'entrée et de sortie)

CHAPITRE 3 : Résultats des simulations

3.1 Introduction

Pour l'implémentation des commandes, nous utilisons le modèle réduit qui est commandable et observable.

Dans ce chapitre, nous présenterons les simulations effectuées sous le logiciel Matlab/Simulink. Les sorties considérées sont la position et l'angle θ ; la consigne prendra comme valeur 0.1 m tout au long de ce chapitre.

La simulation est une étape importante dans la conception des systèmes de commande. Elle permet de régler les paramètres de la commande (matrice de pondération, choix des pôles,...) afin d'aboutir à des résultats sur les réponses du système satisfaisants.

D'un autre côté, elle permet aussi de mieux illustrer l'impact de ces paramètres sur les performances du système.

Afin de protéger le moteur contre les surcharges nous avons insérés un bloc de saturation ayant pour intervalle $[-20\ 20]\text{volts}$ au niveau de la commande appliqué au moteur. cet ajout permet de mieux représenter la réalité.

3.2 Commande par placement de pôles

Comme décrit dans le chapitre précédent la commande par placement de pôles consiste à choisir des pôles en boucle fermée dans la partie gauche du plan de phase afin de stabiliser le système. Cette commande ne prend pas en charge les zéros du système en boucle fermée.

3.2.1 Placement des pôles appliqué au modèle sans la présence de bruits ni perturbations (cas parfait)

Dans cette étude ci nous ne ferons pas l'introduction des bruits et perturbations afin de mieux comprendre l'évolution du système corrigé. Le schéma de commande l'illustrant est représenté dans la figure 3.1.

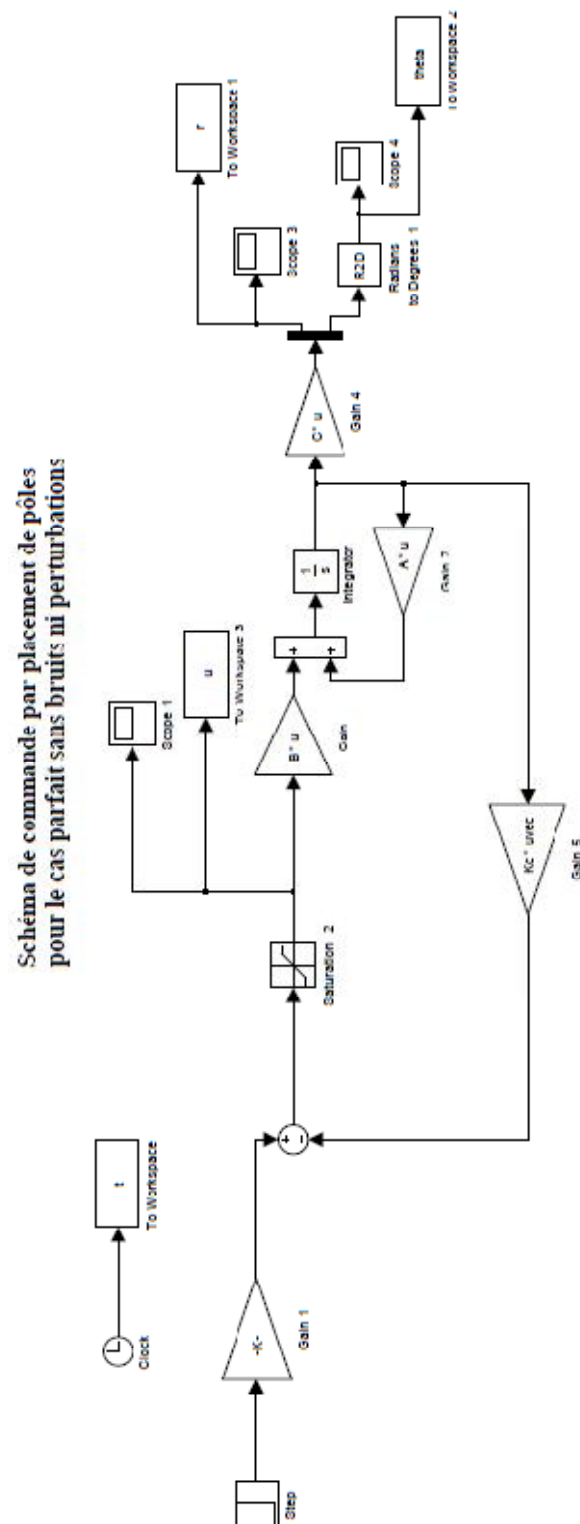


Figure 3.1 : Schéma de la commande par placement de pôles appliquée au système parfait.

i. Le premier choix des pôles

Pour cette partie nous avons préférés choisir des pôles réels (à partie imaginaire nulle) et pas très “ éloigné ” de l’origine :

$$P = [-2 \quad -3 \quad -5 \quad -10] \quad (3.1)$$

Pour ce premier choix des pôles nous obtenons un vecteur gain présenté ainsi :

$$K_p = [4.4118 \quad 5.3480 \quad 7.7746 \quad -5.0456] \quad (3.2)$$

La réponse à cette première commande est représentée dans la figure 3.2 :

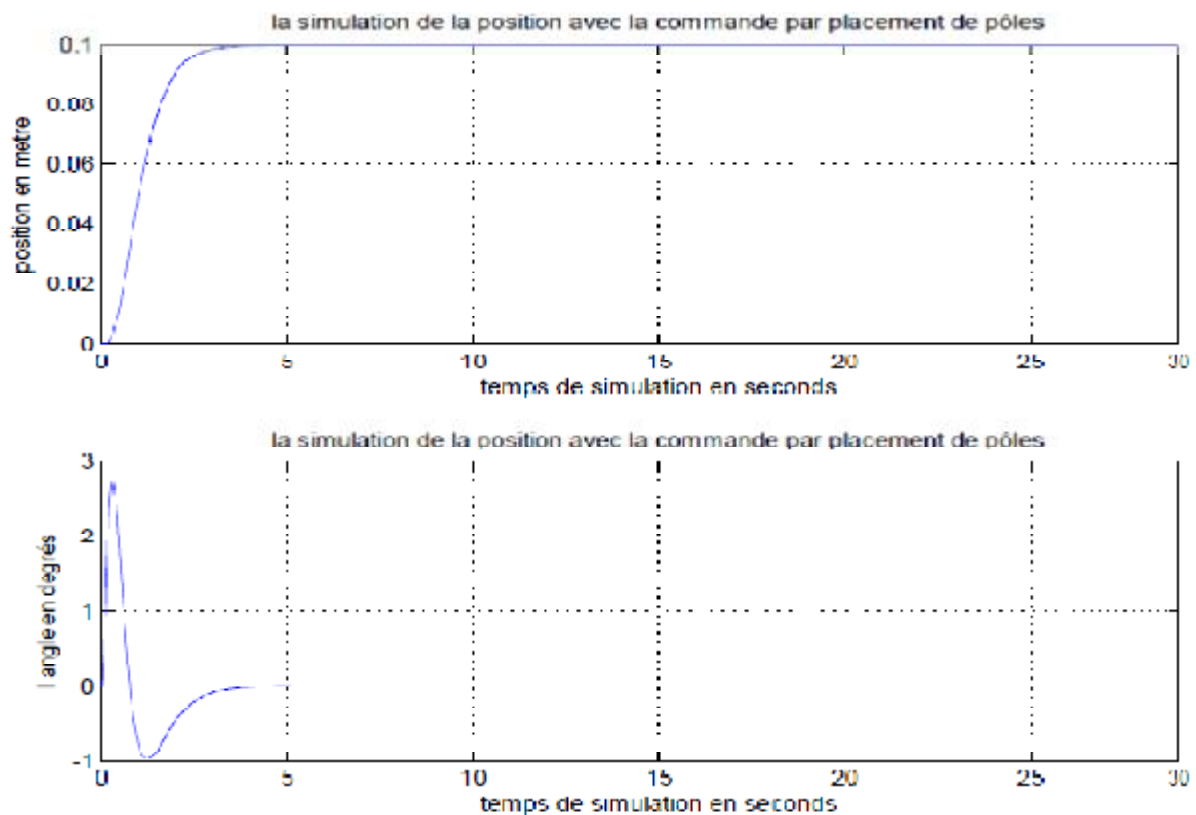


Figure 3.2: Réponse du modèle à un échelon après application de la commande par placement de pôles.

Le signal de commande délivré au moteur à courant continu donnant la réponse illustrée dans la figure 3.2 est représenté ci après dans la figure 3.3 :

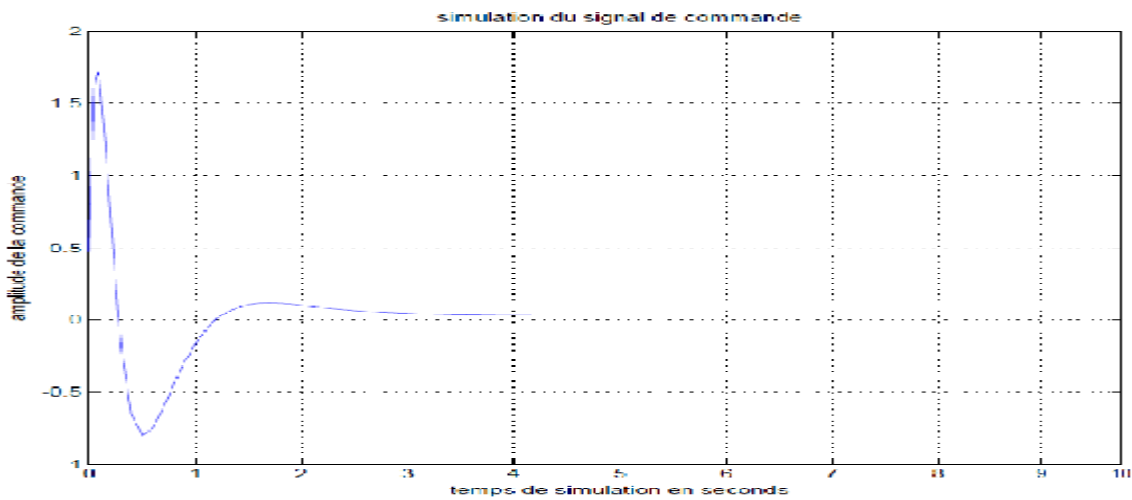


Figure 3.3: Représentation du signal de commande u .

Concernant la figure 3.2 :

Le graphe de la position illustré en premier présente une réponse sans dépassement avec un temps de réponse $t_r = 2.36$ s et une erreur statique nulle. La balle se stabilise complètement à la position désirée à l'instant $t = 4.5$ s.

Quant au graphe de l'angle θ représenté en second il présente un pic à l'instant $t_p = 0.3$ s d'une valeur de 2.73 degrés et une erreur statique nulle.

En ce qui s'agit de la figure 3.3 illustrant la commande $u(t)$ appliquée au moteur, elle aussi présente un pic à l'instant $t = 0.085$ s pour une valeur de $u = 1.72$ volts

ii. Le deuxième choix des pôles

Pour ce deuxième choix des pôles il été intéressant de connaître la réponse du système en présence de pôles mixtes (avec et sans partie imaginaire), à cet effet les pôles ont été choisis ainsi :

$$P = [-4 + j8 \quad -4 - j8 \quad -5 \quad -10] \quad (3.3)$$

On obtient le vecteur gain suivant pour ce second choix des pôles:

$$K_p = [62.6106 \quad 25.1671 \quad 14.8371 \quad -4.8675] \quad (3.4)$$

La réponse du modèle est illustrée dans la figure 3.4. Le signal de commande délivré au moteur est représenté dans la figure 3.5 :

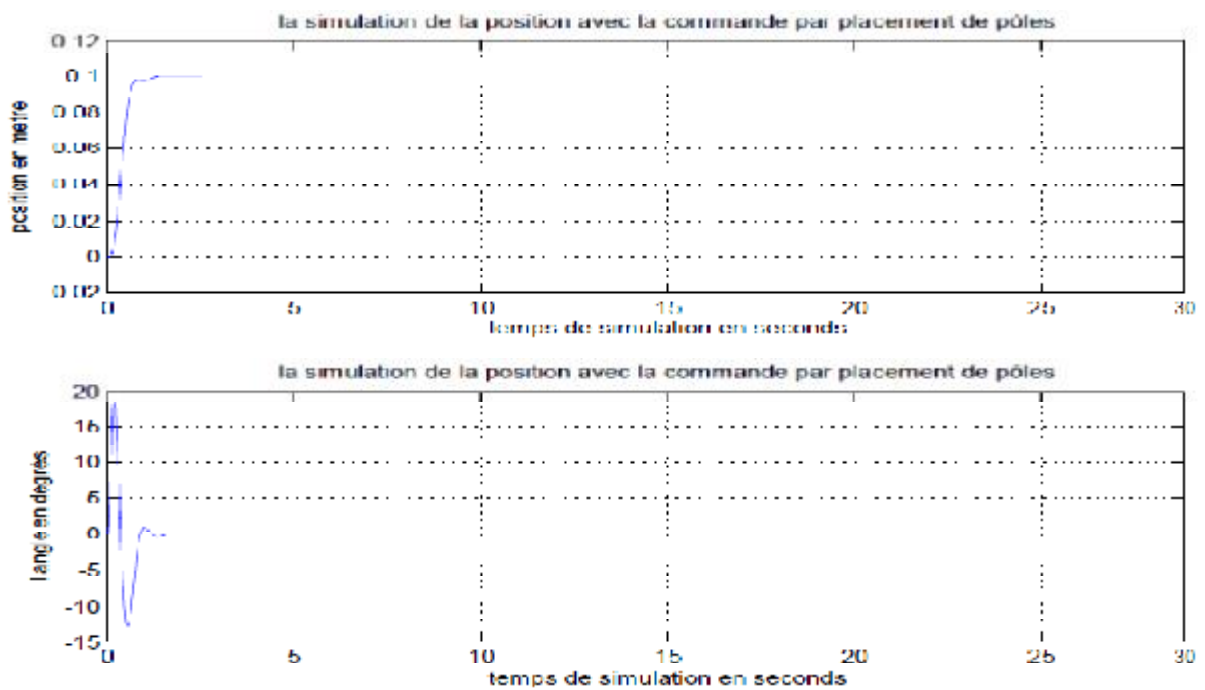


Figure 3.4: Réponse du modèle à la commande par placement de pôles.

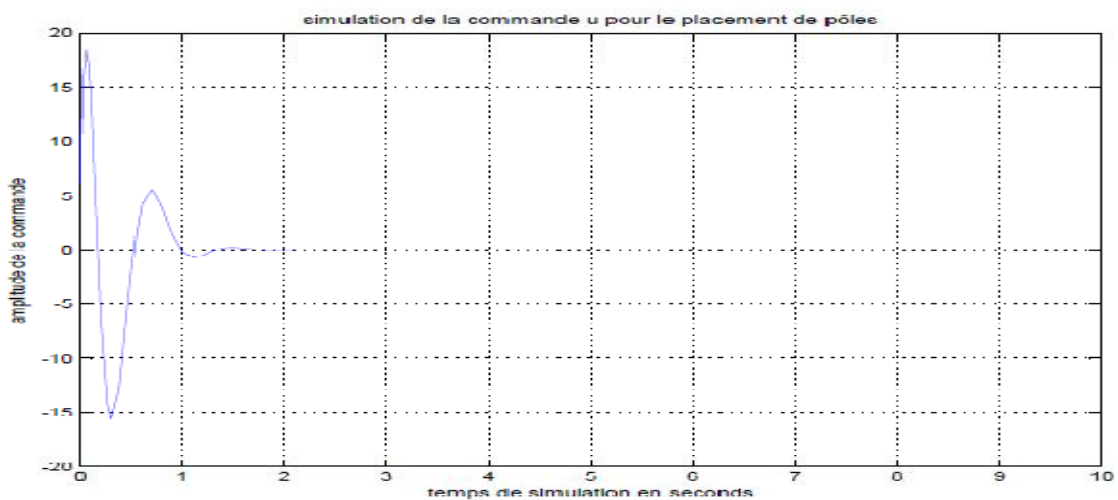


Figure 3.5: Représentation du signal de commande u .

Concernant la figure 3.4 :

Le graphe de la position illustré en premier présente une réponse sans dépassement avec un temps de réponse $t_r = 0.68 \text{ s}$ et une erreur statique nulle.

Quant au graphe de l'angle θ , représenté en second, il présente un pic très élevé à l'instant $t_p = 0.162 \text{ s}$ d'une valeur de 18.5 degrés et une erreur statique nulle dans le régime permanent. Cette réponse est oscillatoire avant de se stabiliser à partir de l'instant 1.7 s

En ce qui s'agit de la figure 3.5 illustrant la commande $u(t)$ appliquée au moteur, elle aussi présente un pic élevé à l'instant $t = 0.05 \text{ s}$ pour une valeur de $u = 18.5 \text{ volts}$.

Le temps de stabilisation de la balle à la position 0.1 m à l'instant $t = 1.5 \text{ s}$, qui est très rapide exige une commande élevée ce qui explique le grand pic de l'angle θ et la commande u .

3.2.2 Placement des pôles appliqué au modèle en présence de bruits seuls

i. Premier choix des pôles :

L'introduction du bruit blanc de moyenne nulle se fera au niveau de la commande u (bruit de commande) avec une énergie égale à 0.001 watt/Hz . Quant aux bruits de mesure, nous en introduirons à la sortie de chacun de la position et de l'angle θ avec des énergies respectivement de 0.00001 watt/Hz et $0.0000001 \text{ watt/Hz}$. [Wan]

La forme du bruit blanc utilisé est illustrée dans la figure 3.6 :

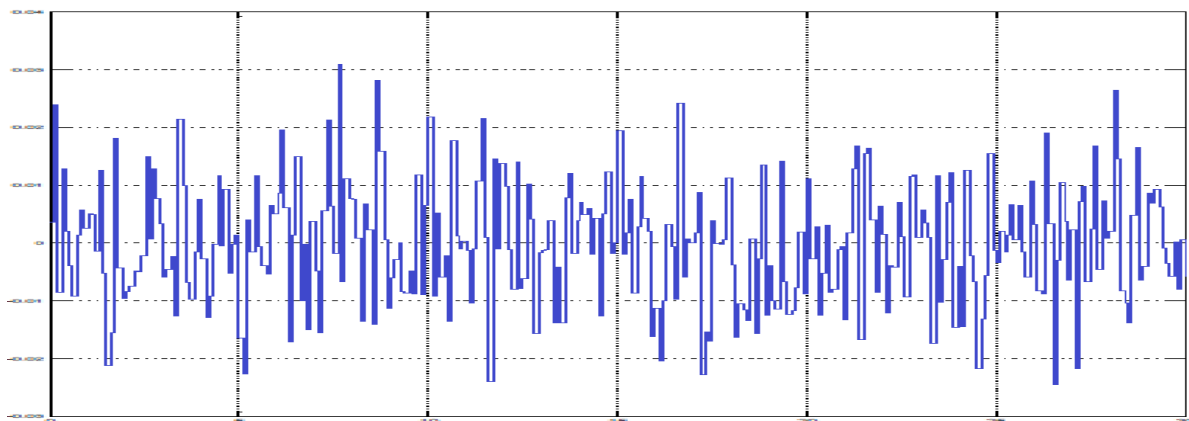


Figure 3.6 : Représentation d'un bruit blanc

Les valeurs des pôles sont ceux présenté dans l'expression (3.1), le vecteur d'état correspondant est donné dans l'expression (3.2)

Le graphe de la position et de l'angle est donné dans la figure 3.7. Le graphe de la commande u correspondante à l'application de cette commande est illustré dans la figure 3.8

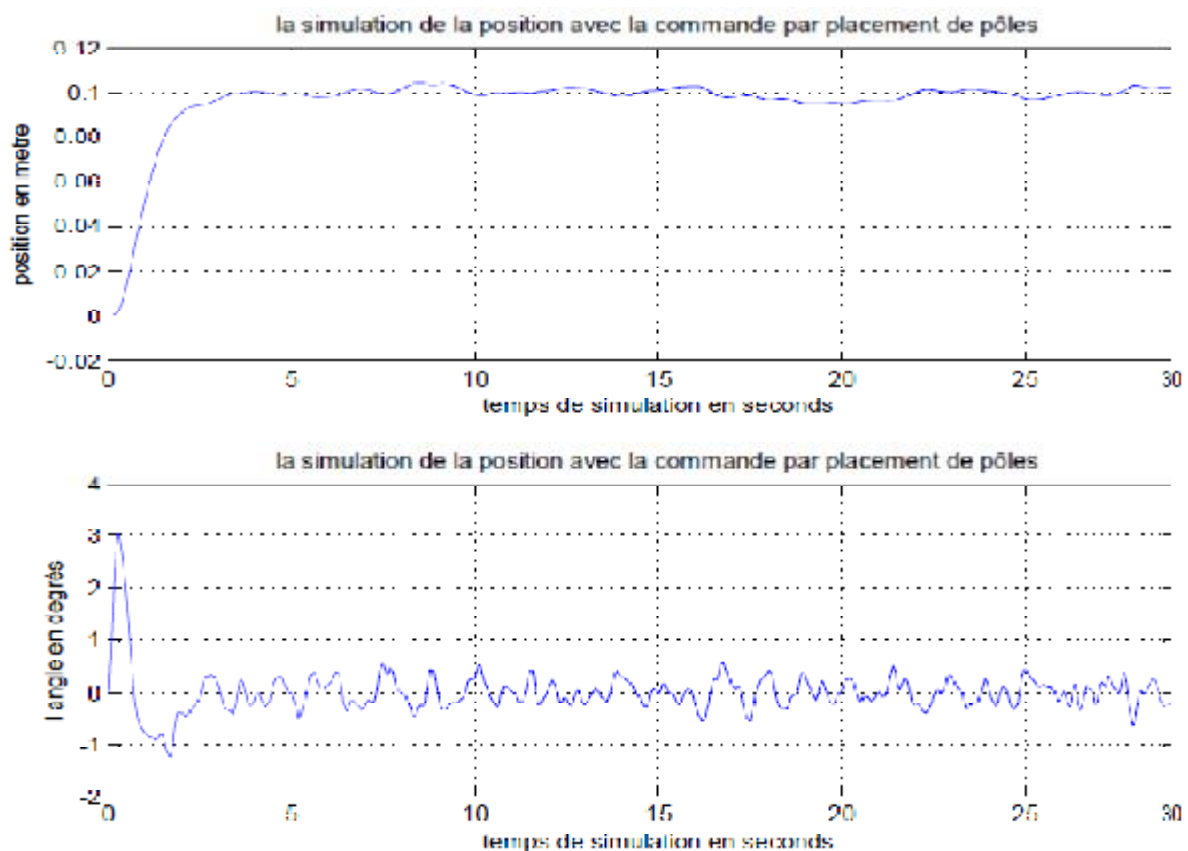


Figure 3.7: Réponse indicielle du modèle à la commande par placement de pôles en présence des bruits.

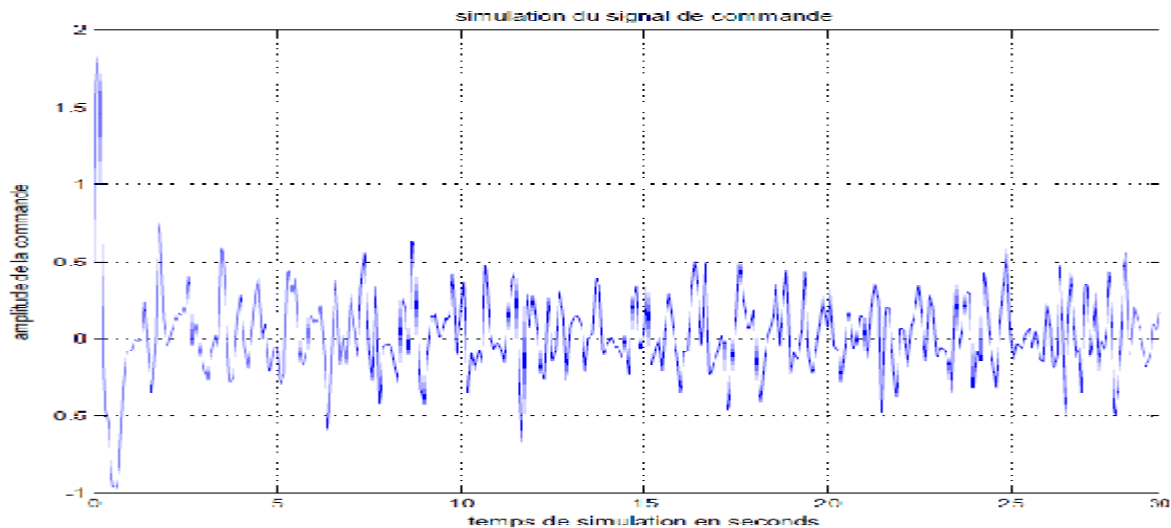


Figure 3.8: Représentation du signal de commande u bruité.

Pour la réponse du modèle à cette commande en présence des bruits de commande et de mesures, nous arrivons à obtenir de bon résultats pour la position, figure 3.5, vu que la balle oscille dans un tube de $\pm 10\%$ de la valeur de la position désirée. Quand à l'angle θ la réponse au régime permanent varie entre ± 0.6 degrés ce qui est très satisfaisant.

En ce qui concerne la commande u , figure 3.6, le signal maximal délivré dans le régime transitoire étant de 2 volts il reste assez raisonnable quant au régime permanent le signal commande varie entre ± 1 volts mais rapidement.

ii. Le deuxième choix des pôles :

Dans ce cas ci, nous avons choisi les pôles de l'expression (3.3) qui nous ont donné le vecteur de retour d'état de l'expression (3.4).

Les graphes de position et d'angle sont illustrés dans la figure 3.10.

Le signal de commande délivré au moteur à courant continu afin d'atteindre la consigne de 0.1 m pour la position est représenté dans la figure 3.11.

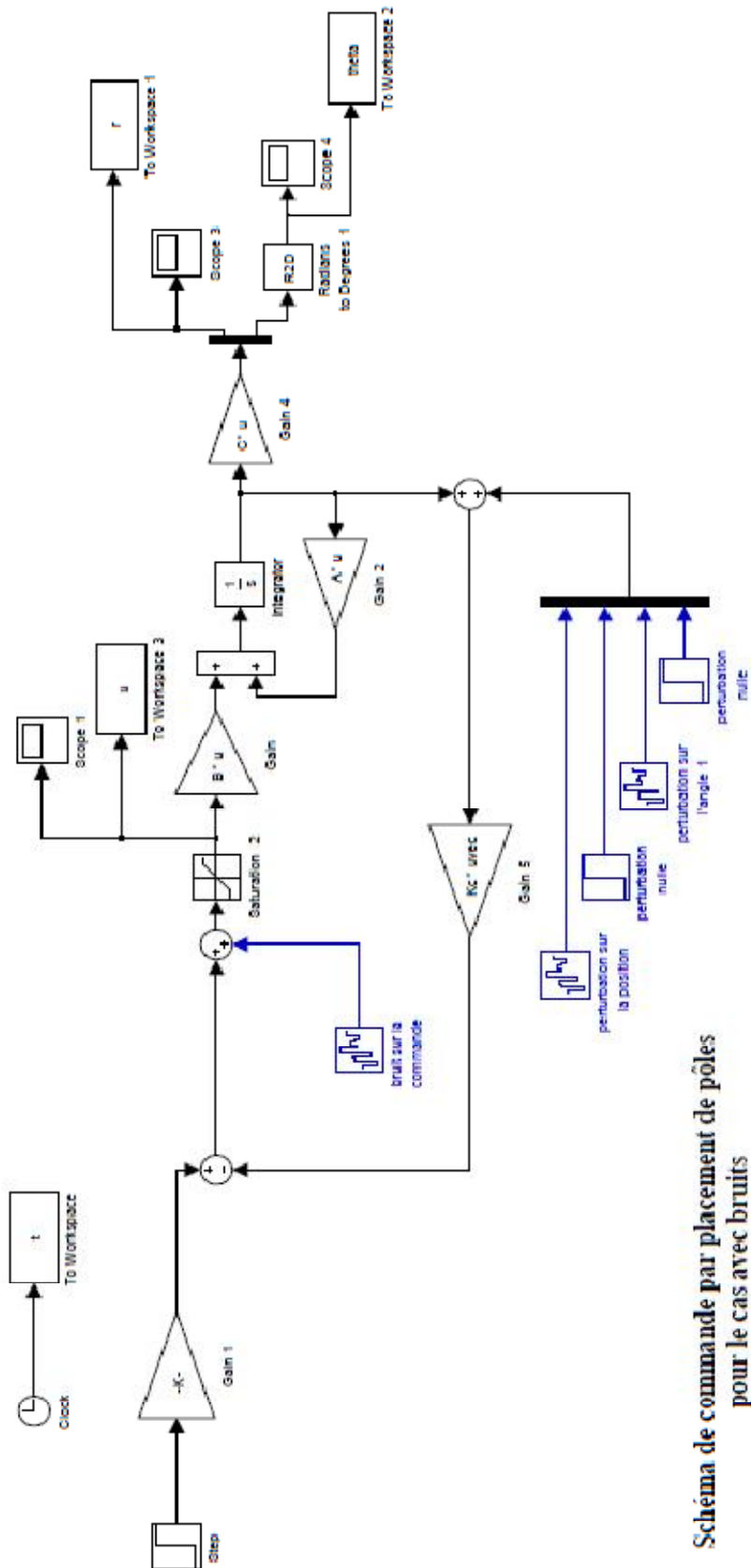


Figure 3.9 : Schéma de commande par placement de pôles appliquée au système en présence des bruits.

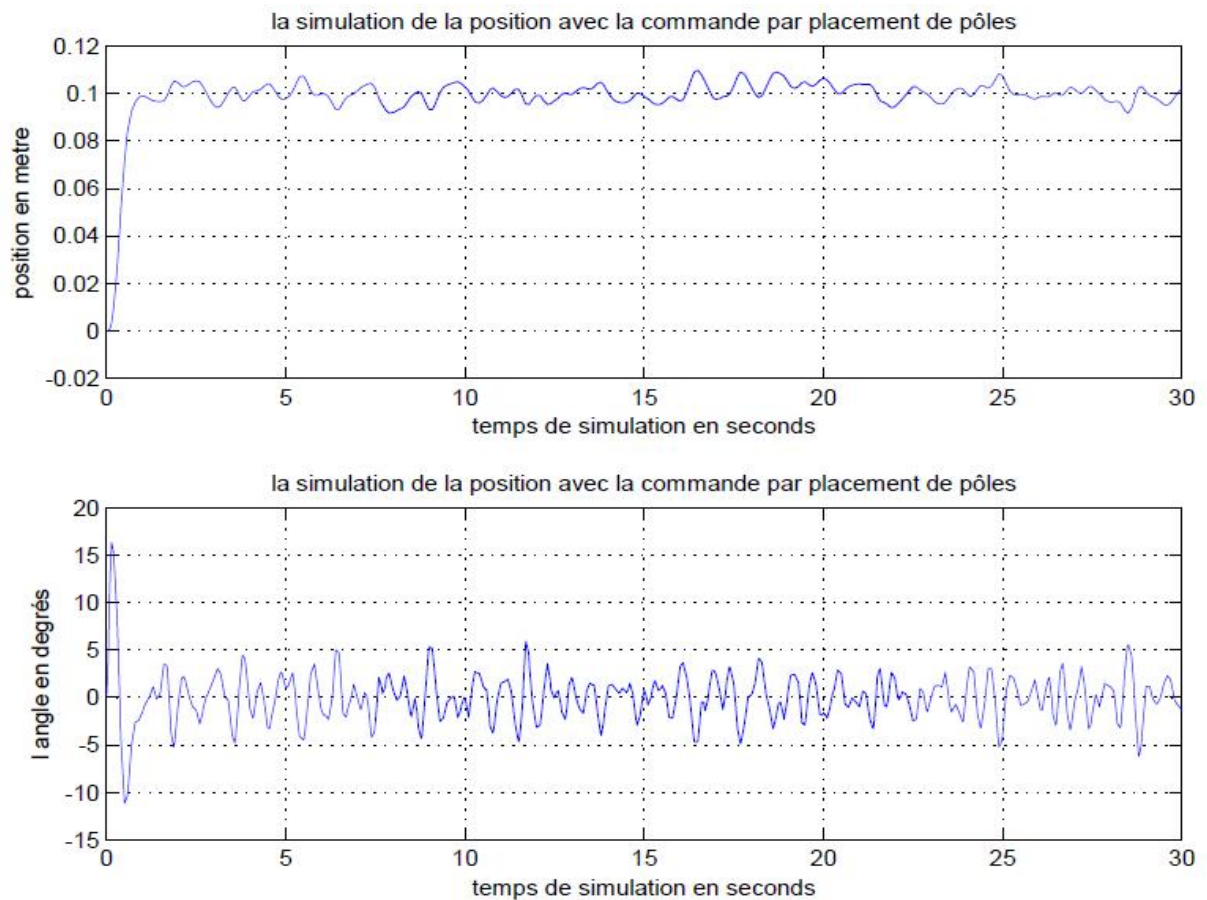


Figure 3.10: Réponse du modèle à la commande par placement de pôles en présence des bruits.

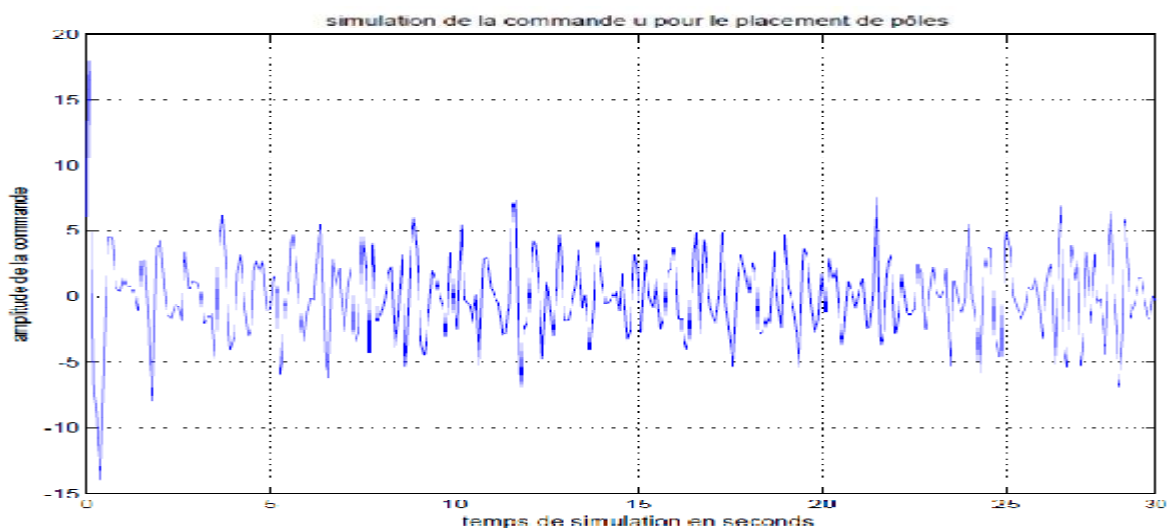


Figure 3.11: Représentation du signal de commande u bruité.

Pour la réponse du modèle à cette seconde commande en présence des bruits, figure 3.10, nous parvenons à obtenir des résultats moins bons que précédemment. Pour la position, la balle oscille dans un intervalle allant de 0.09 à 0.115 m, quant à l'angle θ , que fait la barre avec l'horizontale, il varie entre + ou - 6 degrés au régime permanent et le pic au régime transitoire atteint environ 20 degrés.

En ce qui concerne la commande u , figure 3.11, le signal maximal délivré dans le régime transitoire étant d'environ 18 volts à l'instant $t = 0.05$ s, il redescend aussi rapidement à -15 volts à $t = 0.4$ s, cela est un très grand intervalle de commande en un très petit temps, il est à éviter. S'agissant du régime permanent la commande varie entre + ou - 10 volts mais assez rapidement.

3.2.3 Placement de pôles appliqué au modèle en présence de perturbations

La perturbation représentée dans la figure 3.12 est introduite au niveau du vecteur d'état x . Celle-ci est un créneau d'une largeur de 2 s et d'amplitude 0.05.

La figure 3.13 illustre le schéma de la commande par placement de pôles appliquée au modèle en présence de perturbation.

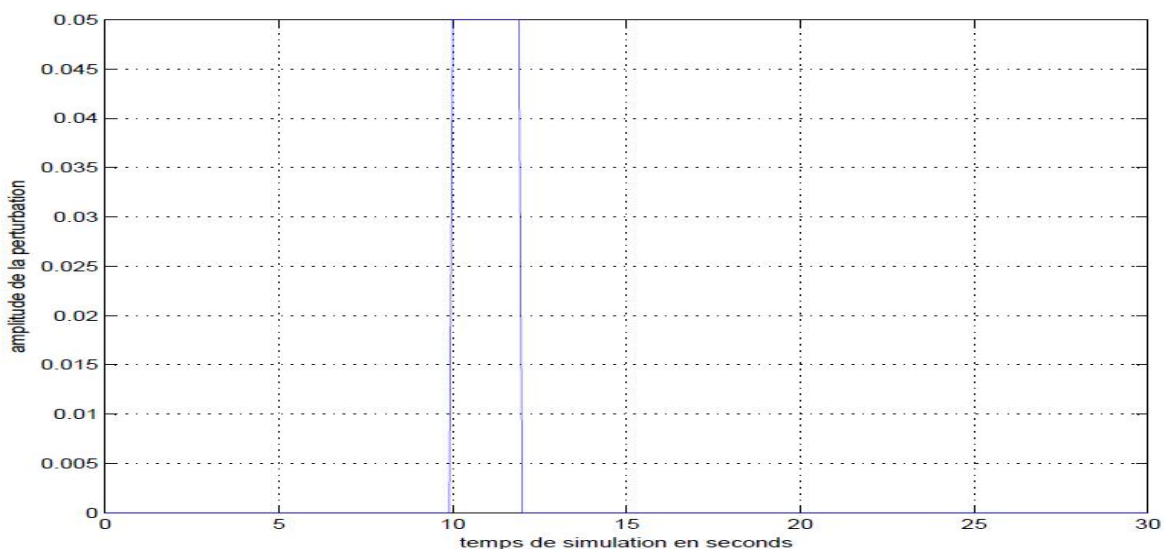


Figure 3.12 : Signal de la perturbation appliquée au modèle.

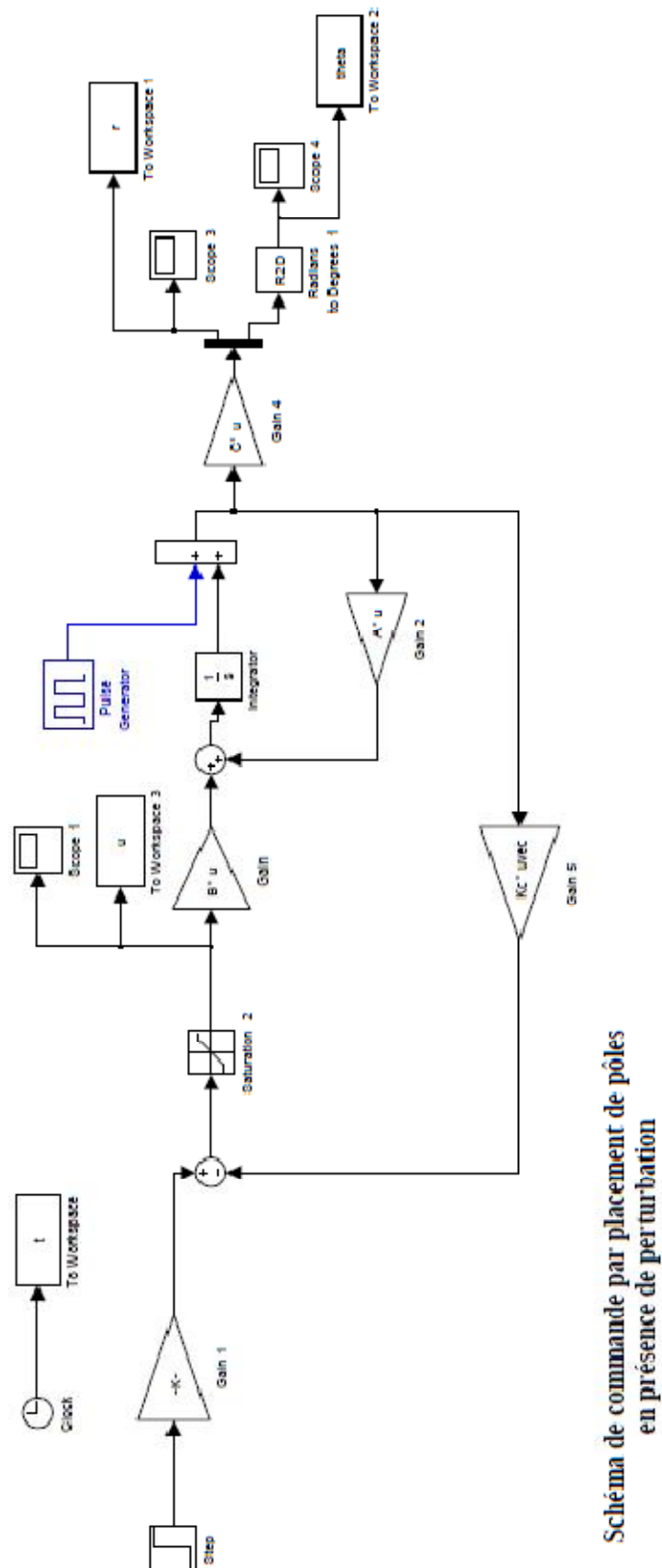


Schéma de commande par placement de pôles en présence de perturbation

Figure 3.13 : Schéma de la commande par placement de pôle appliquée au modèle en présence des perturbations

i. Le premier choix des pôles :

Les valeurs des pôles choisis sont les mêmes que ceux de la section (3.2.1.i).

Ainsi que le vecteur gain, qui est exprimé dans l'équation (3.2).

La figure 3.14 représente la réponse des sorties du modèle ; et la figure 3.15 le signal de commande délivré au moteur

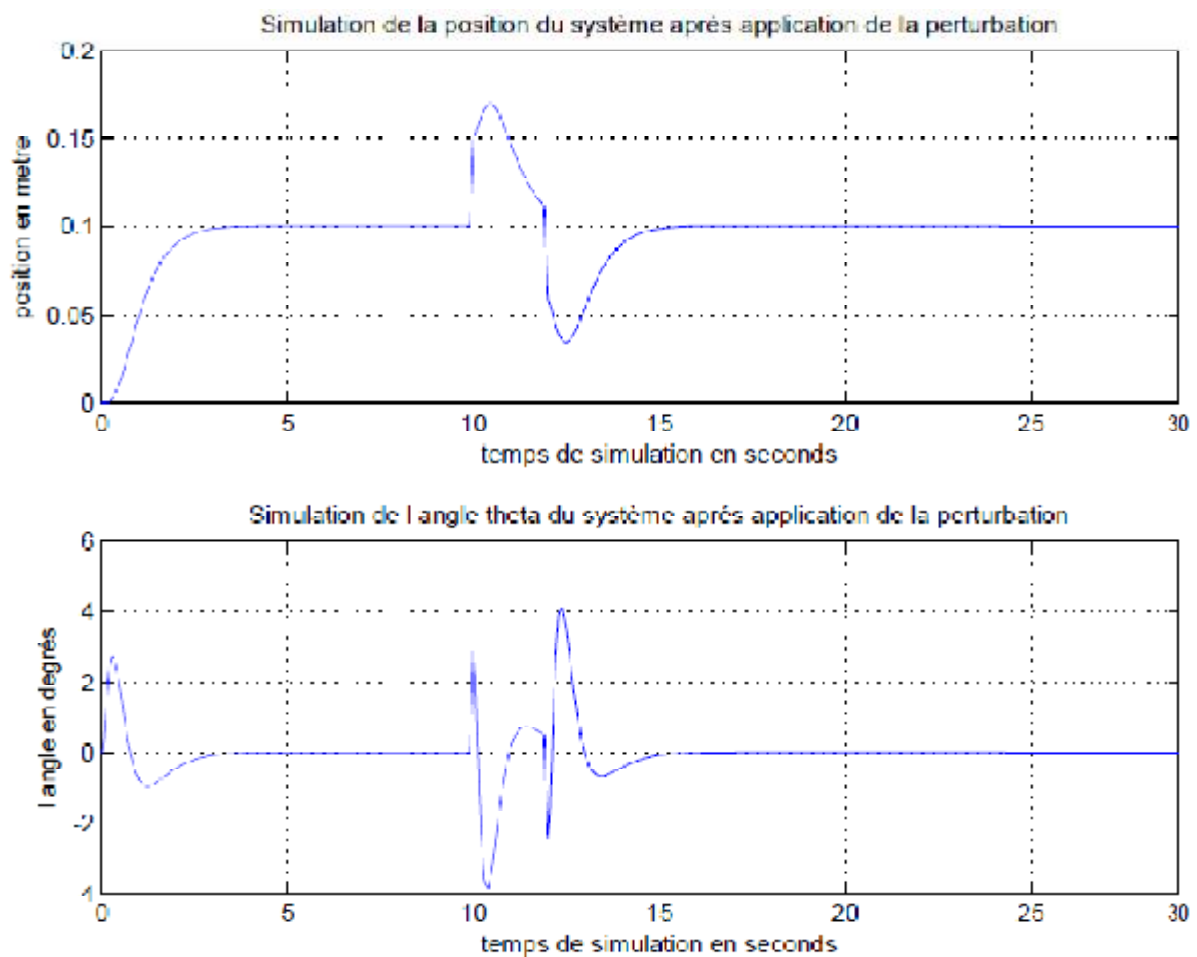


Figure 3.14 : Réponse du modèle à la commande par placement de pôles en présence de la perturbation

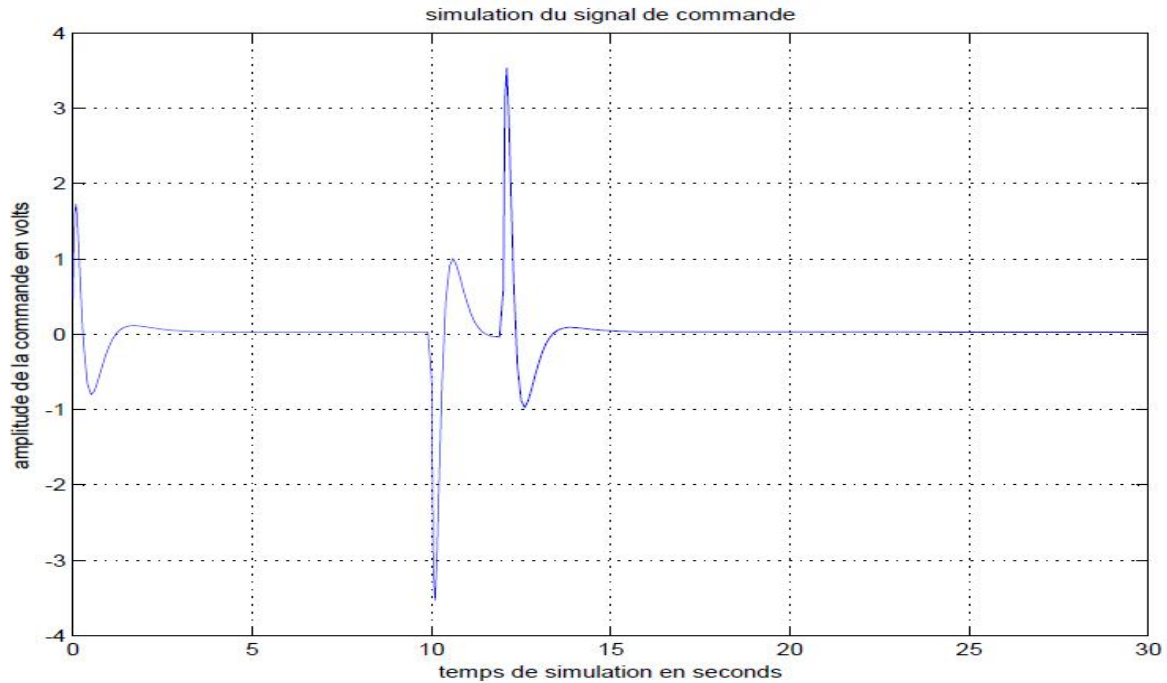


Figure 3.15 : Signal de commande u en présence de perturbation.

Le temps nécessaire pour le rejet de la perturbation, dans ce premier cas ci, est de 5 seconds sachant que les pics maximaux après l'application de la perturbation atteignent 0.17 m pour la position, 4.1 degrés pour l'angle θ et 3.55 volts pour la commande u . Par contre les minimaux sont de 0.034 m pour la position, -3.85 degrés et enfin -3.35 volts pour le signal de commande u .

ii. Le deuxième choix des pôles

Dans ce cas ci on choisira les pôles de l'expression (3.3)

$$P = [-4 + j8 \quad -4 - j8 \quad -5 \quad -10]$$

Ainsi on obtient le même vecteur gain que celui de l'expression (3.4) :

$$K_p = [62.6106 \quad 25.1671 \quad 14.8371 \quad -4.8675]$$

La figure 3.16 représente la simulation des sorties et la figure 3.17 le signal commande u .

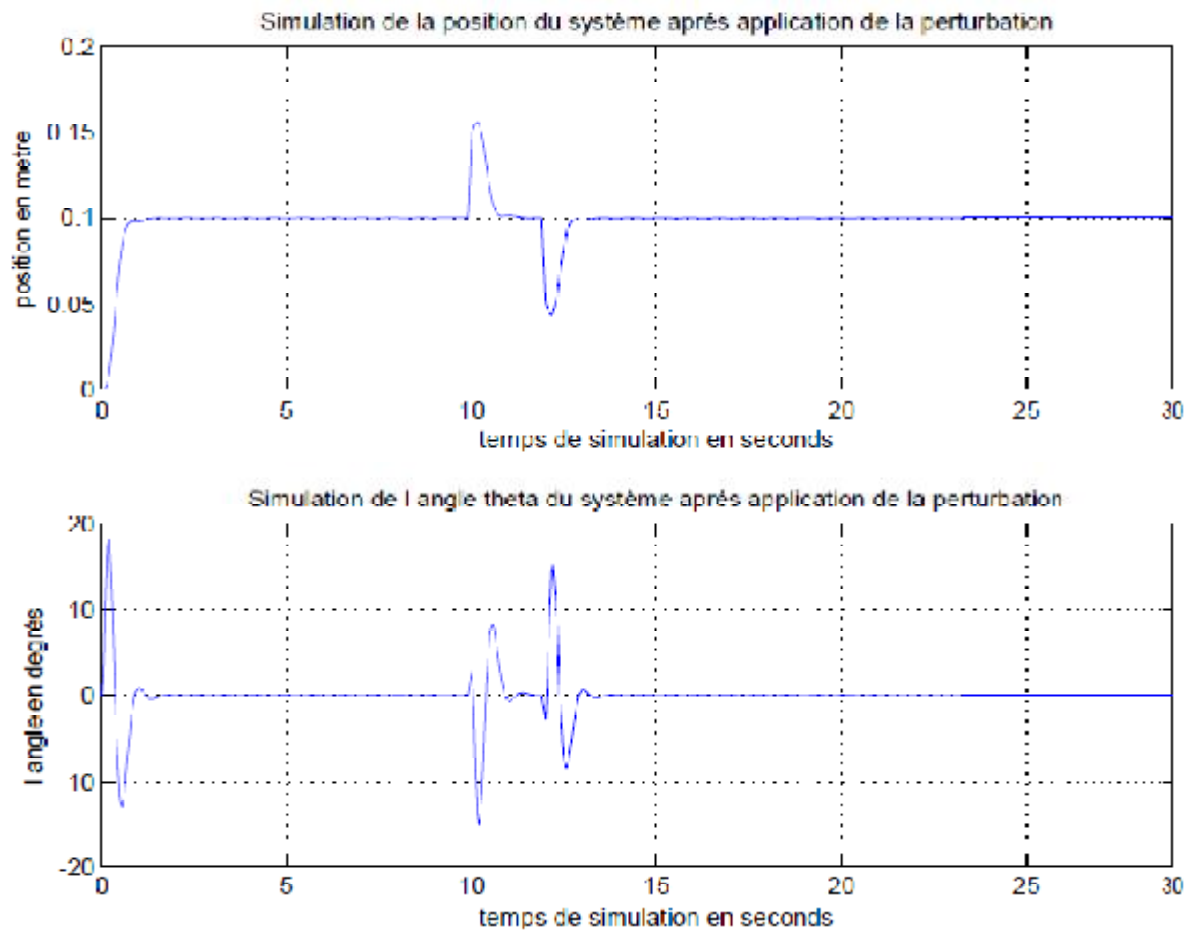


Figure 3.16 : Simulation de la sortie en présence de la perturbation.

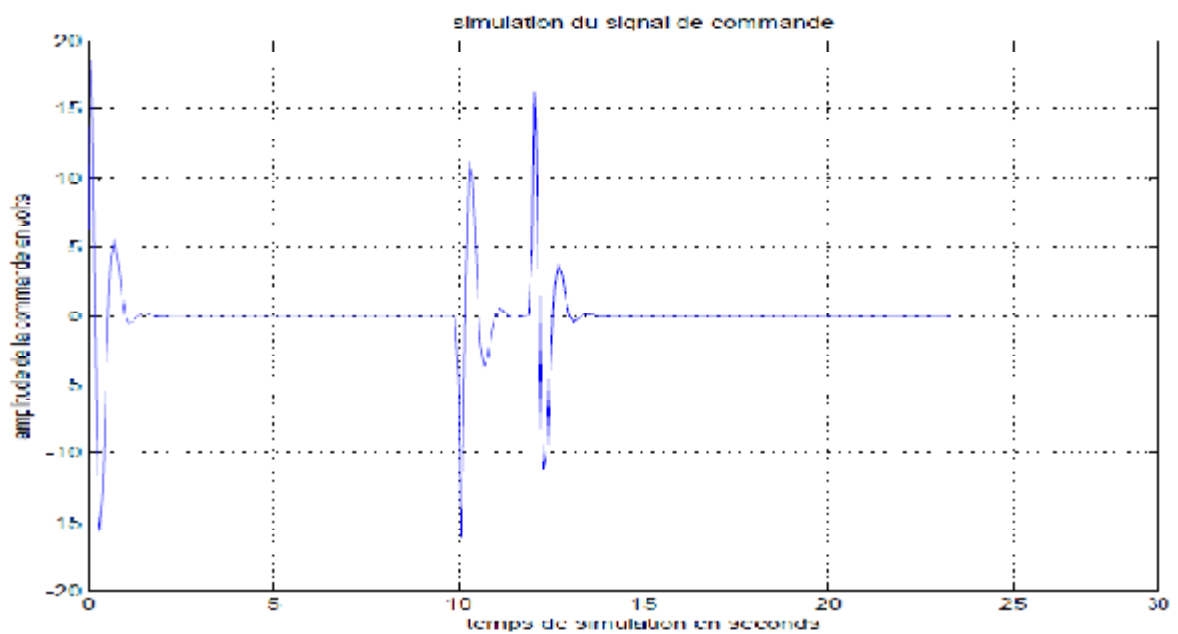


Figure 3.17 : Simulation de la commande u délivrée au moteur en présence de perturbation.

Dans ce cas ci, le temps nécessaire au rejet de la perturbation est de 4 seconds. C'est mieux que pour le cas précédent sauf que la commande délivrée au moteur est plus importante ici.

Les pics maximaux après l'application de la perturbation atteignent 0.156 m pour la position, 15 degrés pour l'angle θ et 16.4 volts pour la commande u . Par contre les minimaux sont de 0.045 m pour la position, -15 degrés et enfin -15.5 volts pour le signal de commande u .

3.2.4 Commande par placement de pôle en présence de l'observateur d'état

Dans ce cas-ci, on gardera les mêmes valeurs pour les bruits de commande et de mesure que ceux de la section précédente.

Le gain de l'observateur est calculé pour les pôles suivants choisis plus grand (environ 20 fois plus grand) que ce du modèle :

$$P_{\text{observateur}} = [-44 \quad -44 \quad -22 \quad -110]$$

Ainsi, en utilisant la commande *place* du logiciel MATLAB, on obtient le gain de l'observateur d'état comme ceci :

$$K_l = \begin{bmatrix} 66.0000 & 0.0000 \\ 968.0000 & 3.7731 \\ -0.0000 & 48.9834 \\ -5.1740 & -304.0715 \end{bmatrix}$$

La réponse du modèle, en présence de l'observateur d'état, à la commande par placement de pôles, pour le cas parfait sans bruit ni perturbation, est la même que dans le paragraphe 3.2.1 cela pour les deux choix des pôles.

Le schéma de simulation illustrant la commande par placement de pôle en présence de l'observateur d'état est représenté dans la figure 3.22

i. Le premier choix des pôles

Pour cette partie, on choisira les mêmes pôles réels que ceux de la première partie précédente (pas de pôles à partie imaginaire):

$$P = [-2 \quad -3 \quad -5 \quad -10]$$

Le gain étant le même que précédemment :

$$K_p = [4.4118 \quad 5.3480 \quad 7.7746 \quad -5.0456]$$

Le signal de commande délivré au moteur à courant continu permettant d'atteindre les consignes est représenté dans la figure 3.18. Quant au graphe de la position et de l'angle, il est donné dans la figure 3.19.

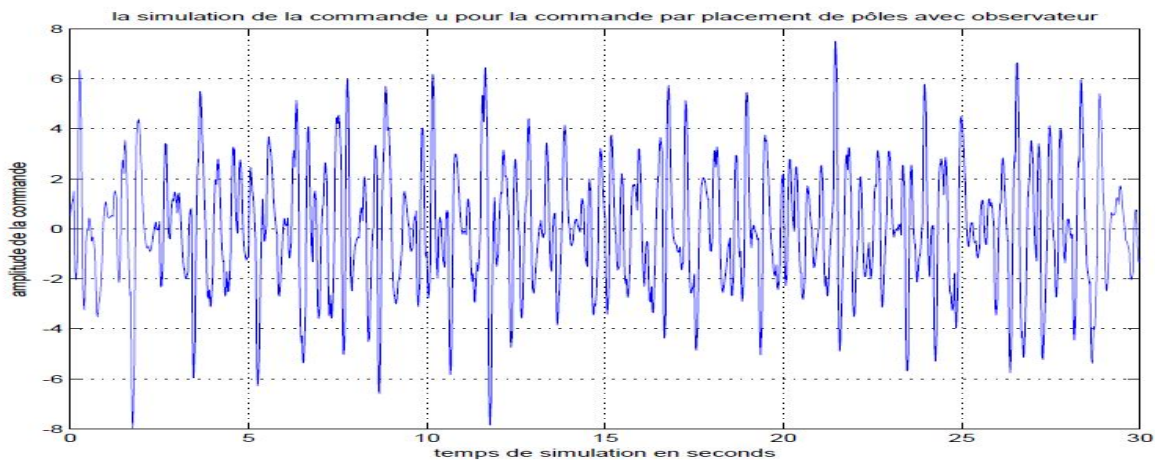


Figure 3.18: Représentation du signal de commande u .

Le signal commande u , figure 3.18, délivré est très rapide dans le temps pour une variation entre + ou - 8 volts, cela est assez grand et bien sur néfaste pour le moteur à courant continu.

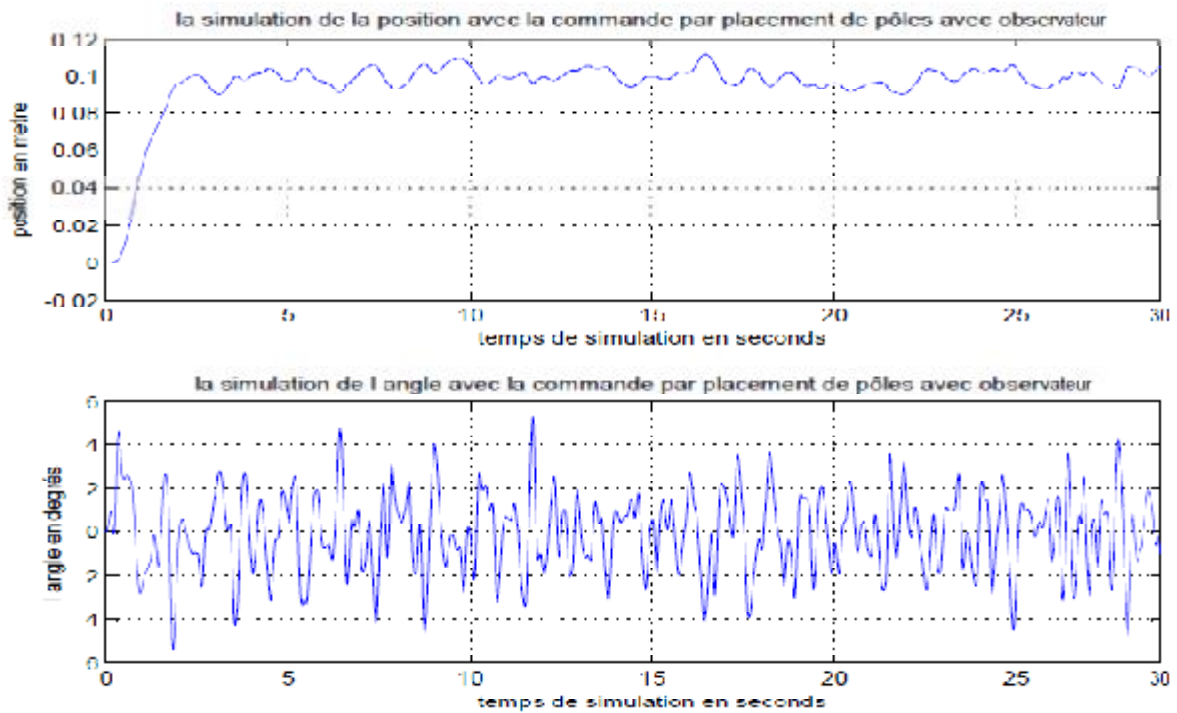


Figure 3.19: Réponse du modèle à la commande par placement de pôles.

La réponse du modèle à cette commande après l'introduction de l'observateur d'état est très mitigé, pour la réponse de l'angle θ qui comme le démontre la figure 3.19 varie dans un intervalle plus grand entre + ou – 6 degrés avec une alternance très rapide ; quant à la réponse de la position elle est assez acceptable vu que la balle oscille, pour son régime permanent, dans un tube allant de 0.09 à 0.112 m.

ii. Le deuxième choix des pôles

Dans ce cas ci, nous choisirons les pôles suivant :

$$P = [-4 + j8 \quad -4 - j8 \quad -5 \quad -10]$$

Ainsi, on obtient le vecteur gain suivant :

$$K_p = [62.6106 \quad 25.1671 \quad 14.8371 \quad -4.8675]$$

Le signal de commande délivré au moteur à courant continu afin d'atteindre les consignes est représenté dans la figure 3.20. Le graphe de position et d'angle est représenté dans la figure 3.21.

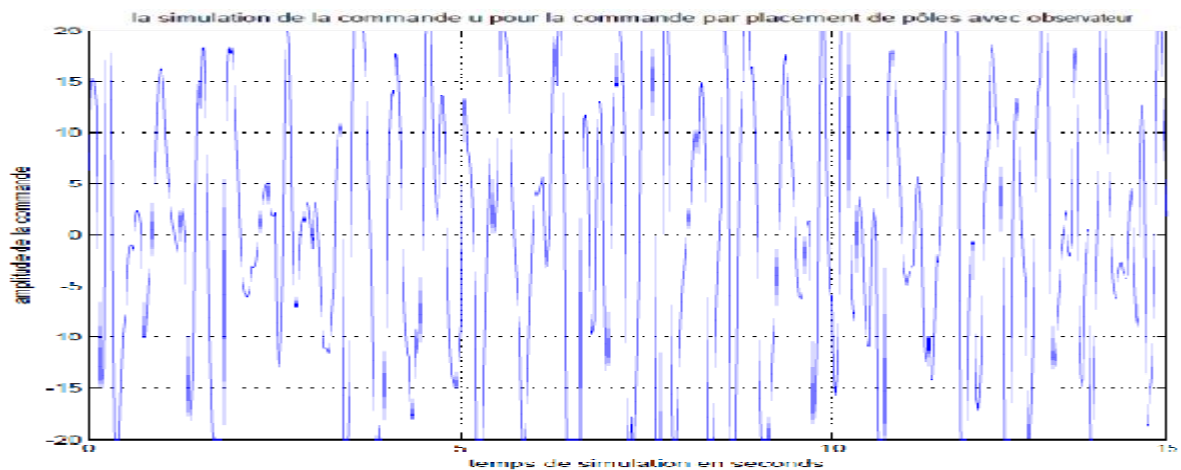


Figure 3.20: Représentation du signal de commande u .

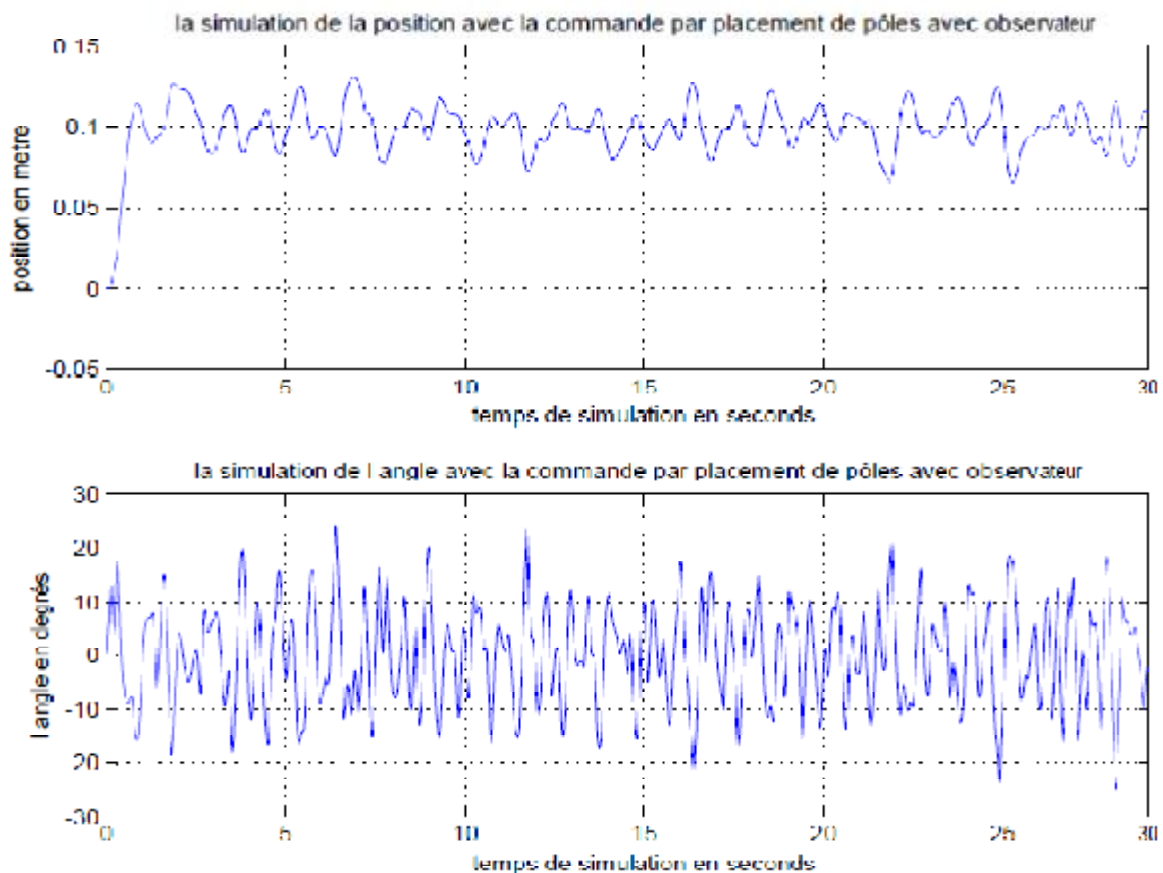


Figure 3.21: Réponse du modèle à la commande par placement de pôles.

Pour cette seconde commande en présence de l'observateur d'état, la réponse du modèle, illustrée dans la figure 3.21, est moins performante que précédemment ; pour la position le tube de variation du système est large vu qu'il s'étale entre 0.065 et 0.132 m et pour l'angle θ , qui est instable durant son régime permanent, il varie très rapidement entre + ou -30 degrés.

Conclusion :

L'introduction de l'observateur d'état à la commande par placement de pôles a influencé négativement les performances du système, on remarque bien dans les deux cas mais avec des degrés différents que les variations sont trop brutales surtout pour le signal de commande ce qui est très néfaste pour le moteur.

Cela est dû au fait que les bruits agissent à la fois sur l'observateur d'état en détériorant son pronostique et en même temps sur le modèle d'état.

Remarque :

- A chaque fois qu'on augmente la valeur des pôles (en valeur absolue) on obtient de meilleures performances, sauf qu'en même temps le signal de commande délivré au moteur va augmenter considérablement et en présence de la saturation on risque d'avoir des réponses divergentes donc il faut faire un compromis entre les valeurs des pôles choisies et le signal de commande.

- La réponse délivrée pour la commande par placement de pôles reste satisfaisante en générale, la présence de mode oscillatoire avec le bruit peut déstabiliser le système.

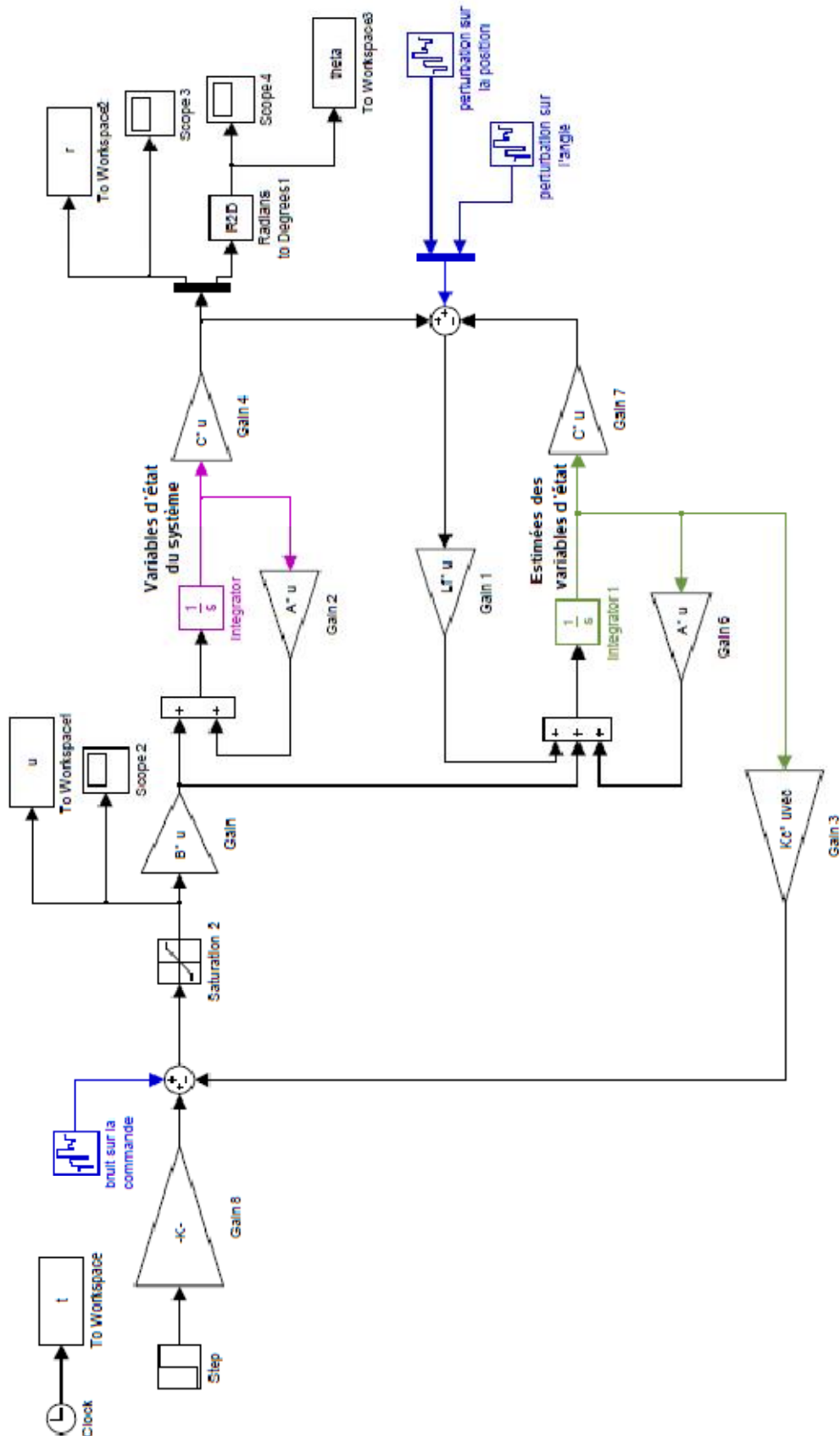


Figure 3.22: Schéma de commande par placement de pôles en présence d'un observateur d'état.

3.3 Commande LQR

3.3.1 Cas parfait (modèle sans perturbation et bruit)

i. Le premier choix des matrices de pondérations

En choisissant les matrices de pondération Q et R dans ce cas-ci comme suit : [Wan]

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R = 0.049$$

Le vecteur gain K_c est obtenu comme suit en appliquant la commande *lqr* du logiciel MATLAB : [Von]

$$K_c = [13.9819 \quad 16.8488 \quad 34.7857 \quad 1.7287]$$

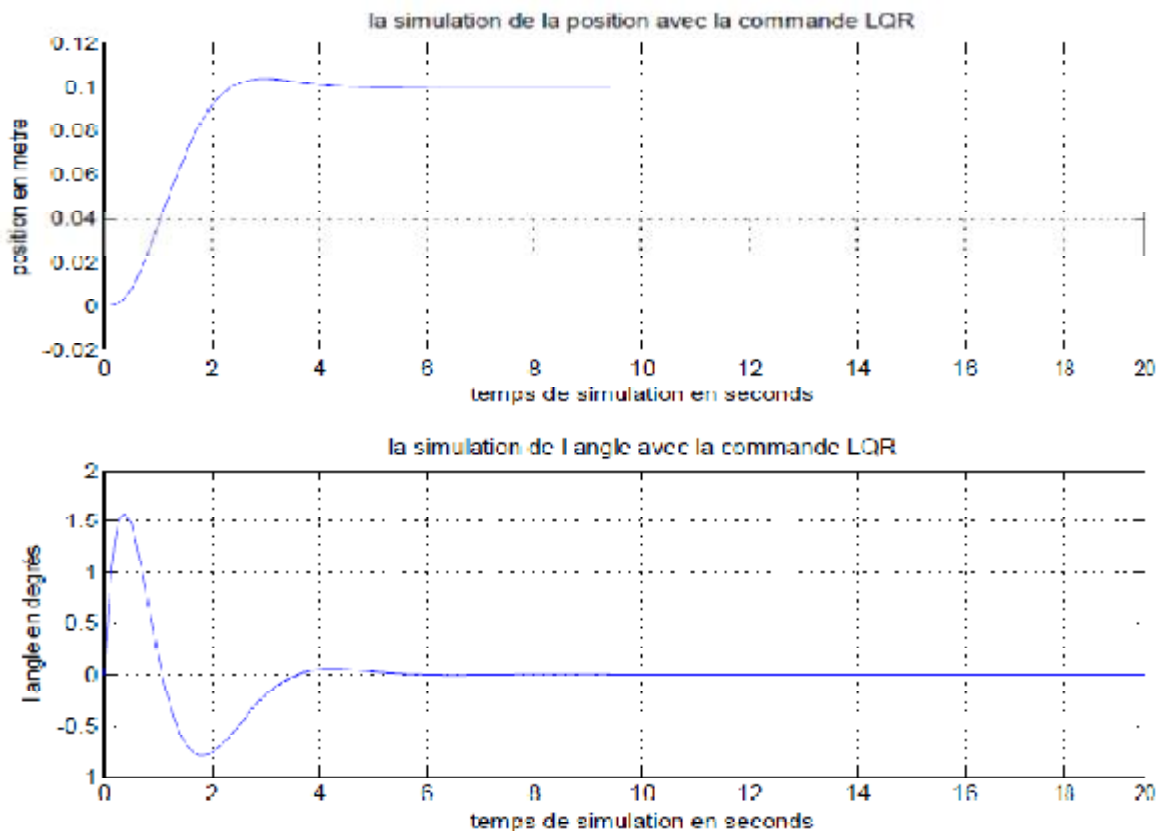


Figure 3.23 : La réponse du modèle pour la commande LQR

Le signal de commande délivré au moteur pour ce choix des matrices de pondération est représenté dans la figure suivante :

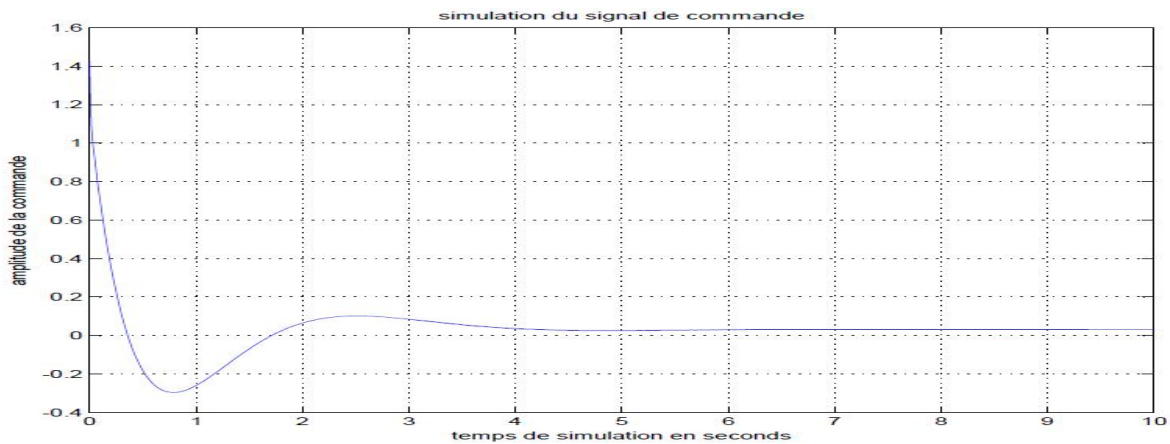


Figure 3.24: Signal de commande u après l'application de la commande LQR

La réponse du modèle à cette commande LQR pour le premier choix des pondérations est très satisfaisant, figure 3.23 ; en ce qui concerne la position nous obtenons une sortie très avec un léger dépassement de prêt de 4.1%, un temps de réponse $t_r = 2.125\text{ s}$ et une erreur statique nulle dans le régime permanent. Quant à la réponse de l'angle θ on observe un dépassement atteignant l'amplitude de 1.56 degrés au début mais une erreur statique nulle au régime permanent aussi.

En ce qui concerne la commande u , illustrée dans la figure 3.24, premier pic "très rapide" risque de ne pas être suivi par le moteur quoi qu'il soit d'assez petite amplitude vu que le max atteint 1,4volts.

Le schéma de la commande LQR est illustré dans la figure 3.25

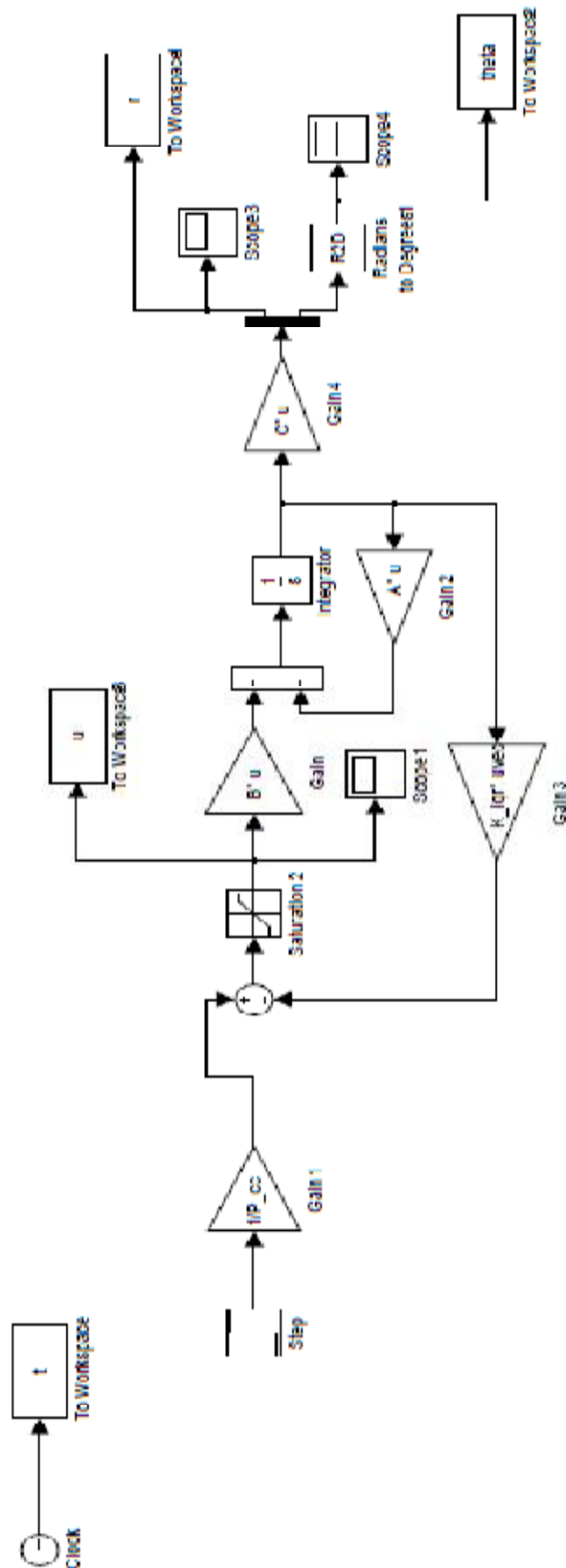


Figure 3.25 : Schéma de la commande LQR appliquée au système.

ii. Le deuxième choix des matrices de pondérations

Les nouvelles matrices de pondération sont données ci-dessous :

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}, \quad R = 0.05$$

Le gain correspondant à ce choix de matrice est comme suit :

$$K_c = [13.8384 \quad 24.9002 \quad 56.0167 \quad 9.4357]$$

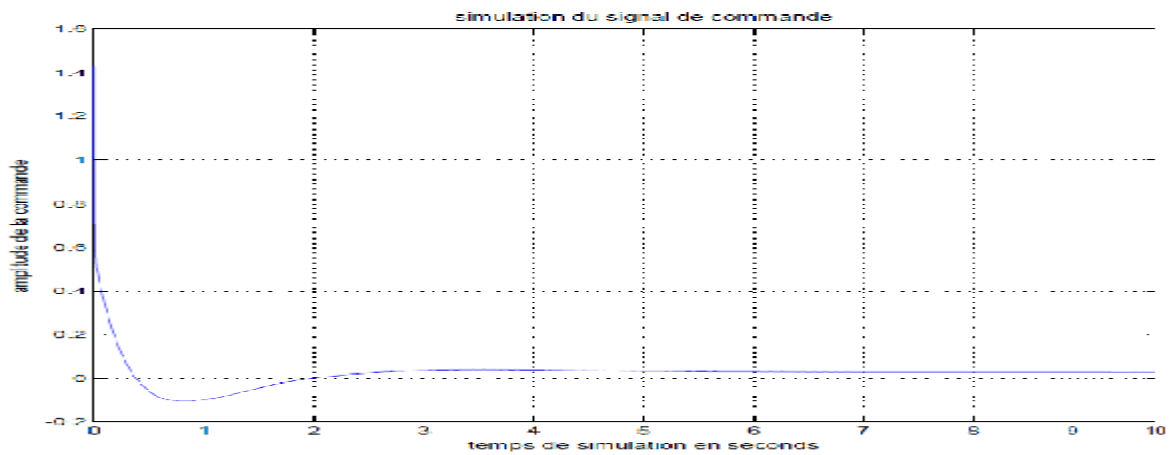


Figure 3.26: Signal de commande après l'application de la commande LQR

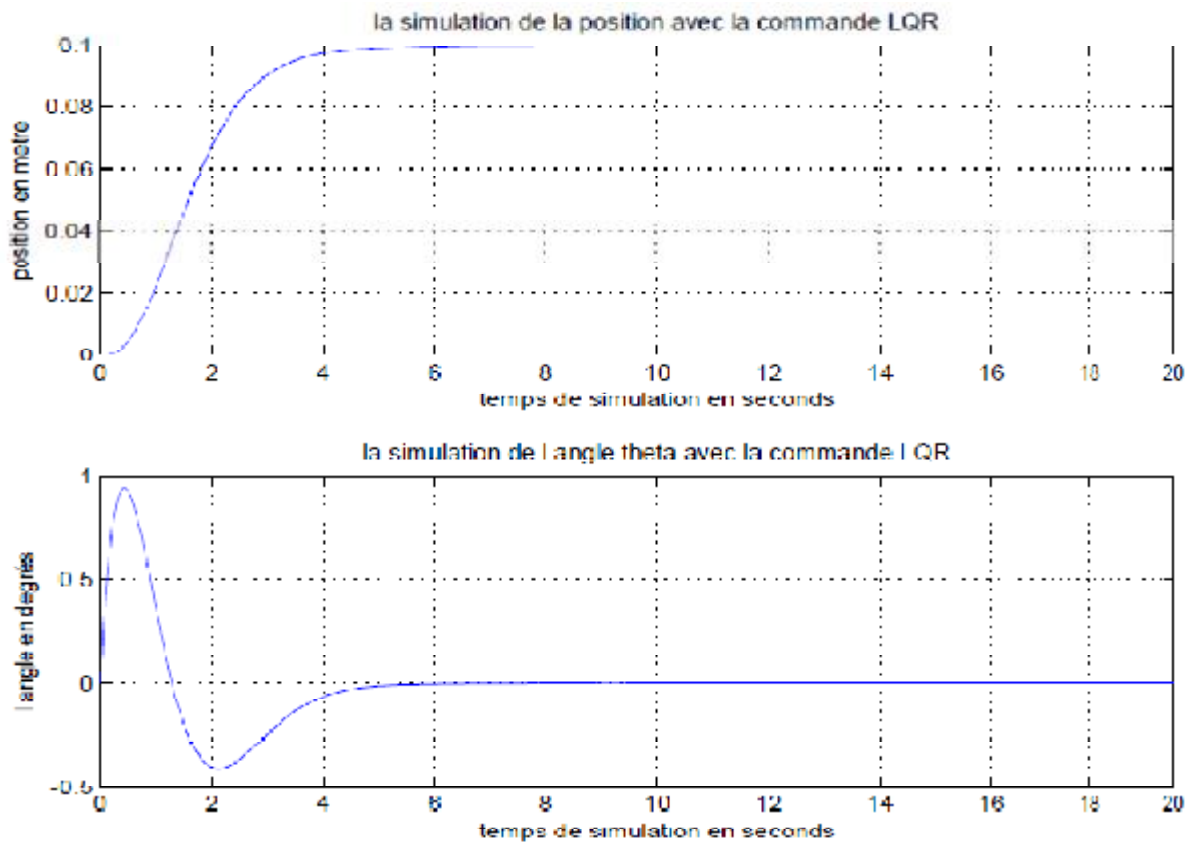


Figure 3.27 : Réponse du modèle à la commande LQR

La réponse du modèle à cette commande LQR, pour le deuxième choix des pondérations, est illustrée dans la figure 3.27, est très satisfaisant ; en ce qui concerne la position, on obtient une sortie sans dépassement, un temps de réponse $t_r = 3.5\text{ s}$ et une erreur statique nulle.

Quant à la réponse de l'angle θ , on observe un dépassement atteignant l'amplitude de 0.94 degrés au début et une erreur statique nulle aussi.

Concernant la commande u illustrée dans la figure 3.26, la rapidité du premier pic, survenu à l'instant $t = 0\text{ s}$, risque de ne pas être supporté par le moteur quoi qu'il soit d'assez petite amplitude vu que le max atteint 1,4volts.

Conclusion :

Pour le deuxième choix des pondérations, on évite le dépassement qui peut être gênant par moment et même inacceptable dans plusieurs autres cas.

3.3.2 Cas avec bruits

i. Le premier choix des matrices de pondérations

Comme pour le cas parfait on prendra les matrices de pondération suivante :

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R = 0.049$$

Le vecteur gain K_c est obtenu comme suit :

$$K_c = [13.9819 \quad 16.8488 \quad 34.7857 \quad 1.7287]$$

L'introduction du bruit se fera au niveau de la commande u (bruit de commande) avec une énergie égale à 0.1 watt/Hz . Quant aux bruits de mesure, nous en introduirons à la sortie de chacun de la position et de l'angle θ avec des valeurs d'énergies respectivement 0.00001 watt/Hz pour la position et $0.0000001 \text{ watt/Hz}$ pour l'angle θ .

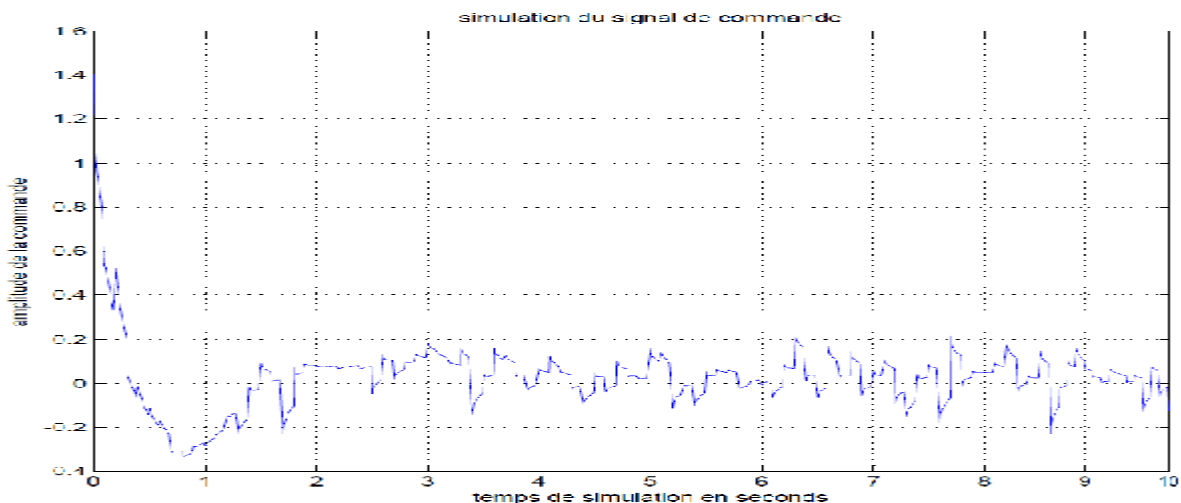


Figure 3.28: Signal de commande après l'application de la commande LQR

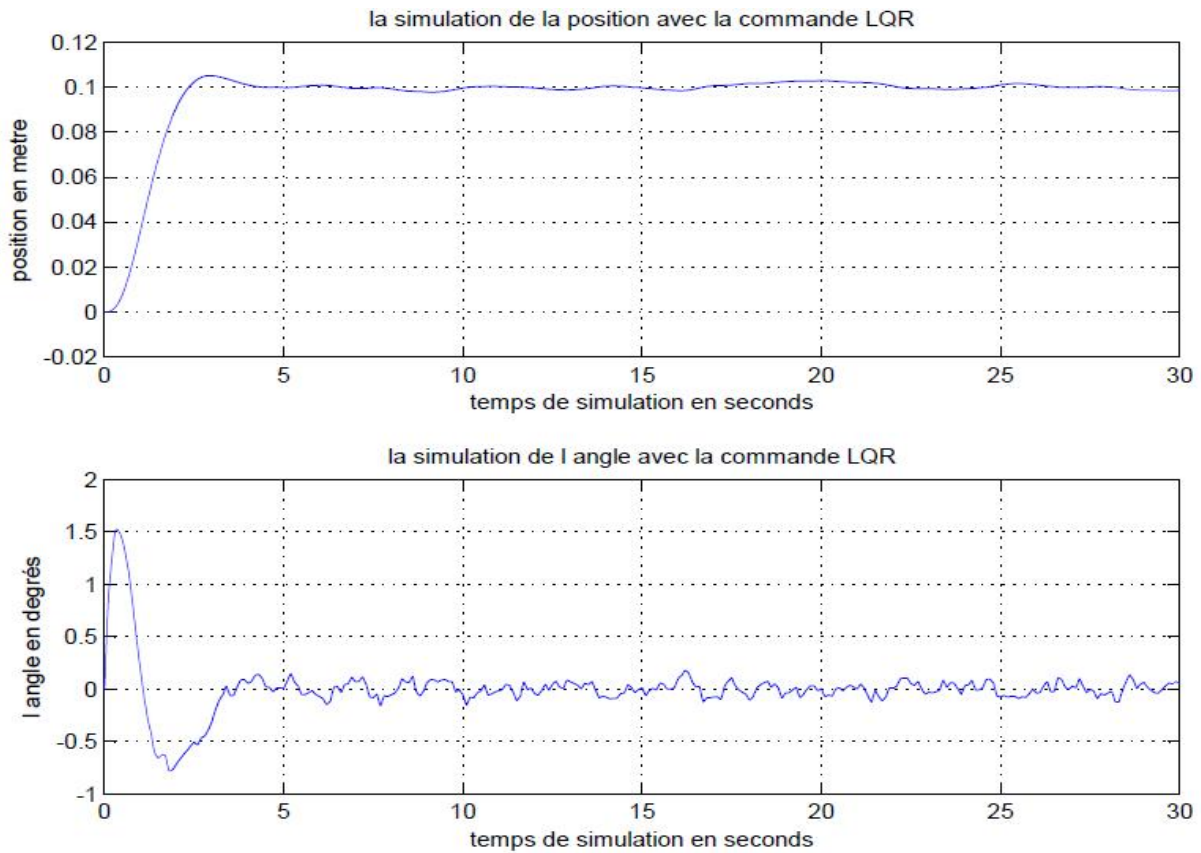


Figure 3.29: Réponse du modèle à la commande LQR.

La réponse du modèle à cette commande LQR pour le premier choix des pondérations en présence des bruits est très satisfaisant (figure 3.29) ; en ce qui concerne la position nous obtenons une sortie avec un léger dépassement de près de 5.1% et un bon rejet des bruits vu que la balle ne dépasse pas le tube de $\pm 5\%$ de la consigne dans le régime permanent. Quant à la réponse de l'angle θ on observe un dépassement atteignant l'amplitude de 1.5 degrés au début mais au régime permanent la barre oscille avec de très petit angle de ± 0.2 degrés.

Concernant la commande u (figure 3.28), le régime transitoire présente un pic qui risque de ne pas être supporté par le moteur quoi qu'il soit d'assez petite amplitude vu que le max atteint 1.4volts et pour le régime permanent le signal de commande varie entre ± 0.3 volts cela étant une assez petite commande.

ii. Le deuxième choix des matrices de pondération

Les matrices de pondération sont :

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}, \quad R = 0.05$$

Le gain correspondant à ce choix de matrice est comme suit :

$$K_c = [13.8384 \quad 24.9002 \quad 56.0167 \quad 9.4357]$$

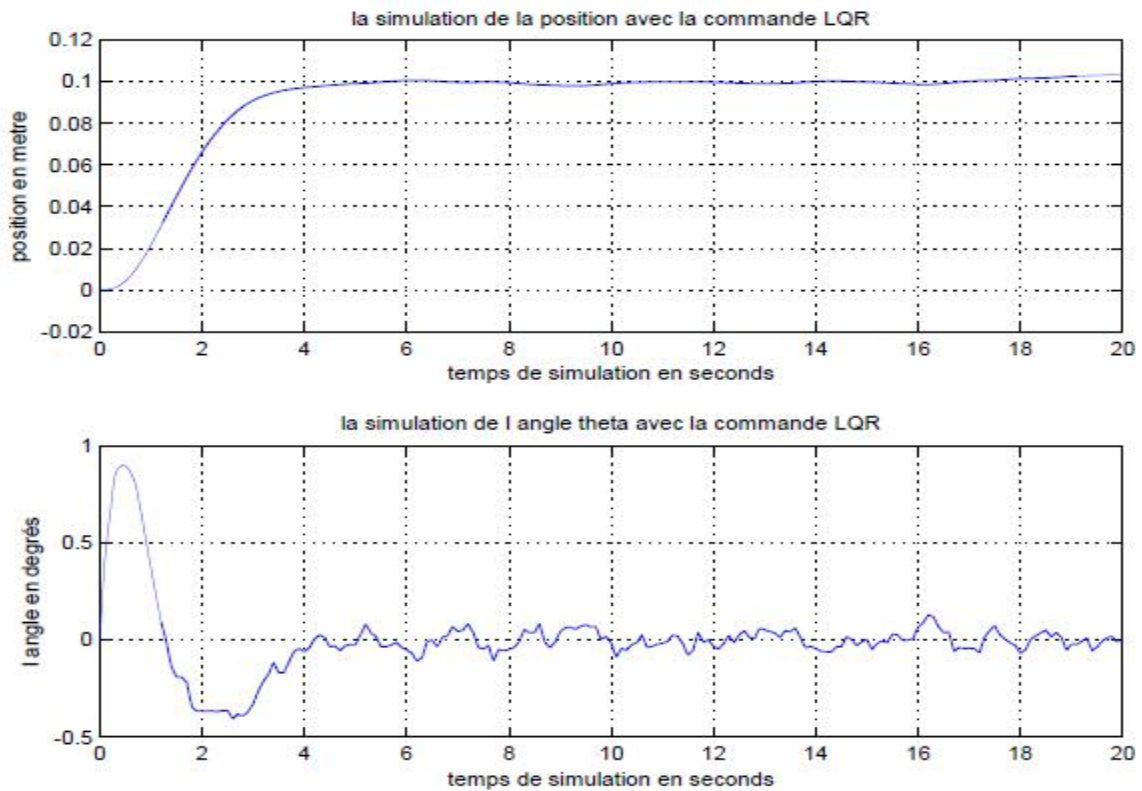


Figure 3.30 : Réponse du modèle à la commande LQR.

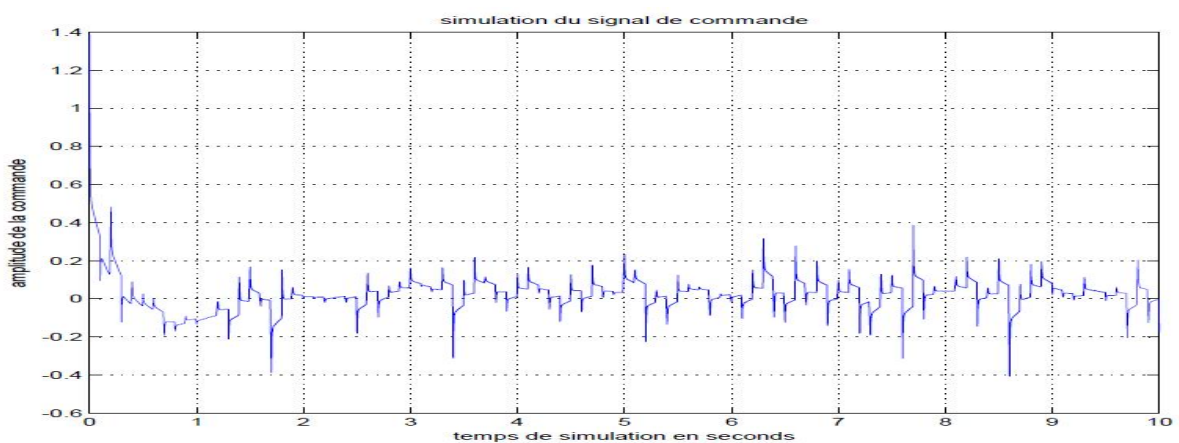


Figure 3.31 : Signal de commande après l'application de la commande LQR.

La réponse du modèle à cette commande LQR pour le deuxième choix des pondérations en présence des bruits représentée dans la figure 3.30 est très satisfaisant ; en ce qui concerne la position on obtient une sortie sans dépassement et un très bon rejet des bruits vu que ça ne dépasse pas le tube de $\pm 5\%$ de la consigne, qui est de 0.1 m , dans le régime permanent. Quant à la réponse de l'angle θ on observe un dépassement atteignant l'amplitude de 0.9 degrés au début. Mais au régime permanent, la sortie ne dépasse pas un tube ayant pour intervalle $[-0.11\ 0.15]$ degrés.

Concernant la commande u , qui est illustrée dans la figure 3.31, nous obtenons un signal “presque identique” à celui représenté dans la figure 3.28 avec des variations plus prononcées dans ce cas ci.

Conclusion :

Le second choix des matrices de pondération donne de meilleurs résultats par rapport au premier choix sans pour autant qu'il soit mauvais. En effet les deux choix des pondérations donnent une excellente performance cela est dû à la grande robustesse de la commande LQR.

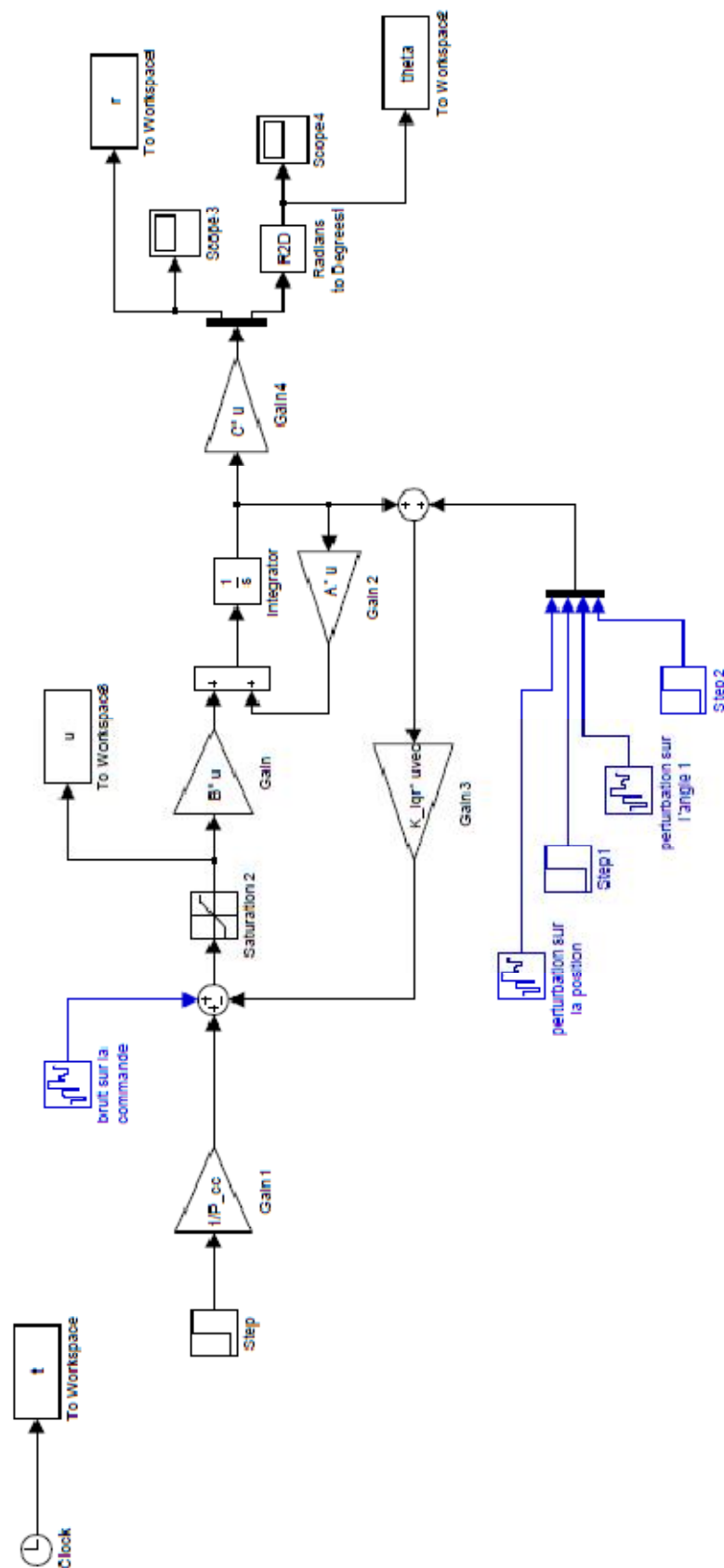


Figure 3.32: Système de commande LQR en présence de bruits.

3.3.3 Cas avec perturbation seule

La perturbation choisie pour cette section est représentée dans la figure 3.12

i. Premier choix des matrices de pondération :

Les matrices de pondération prises pour cette section sont :

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R = 0.049$$

Le vecteur gain K_c est obtenu comme suit :

$$K_c = [13.9819 \quad 16.8488 \quad 34.7857 \quad 1.7287]$$

Le signal de commande est illustré dans la figure 3.33 :

Le graphe représentant la réponse du système après l'application de la perturbation pour ce premier choix des matrices de pondération est illustré dans la figure 3.34.

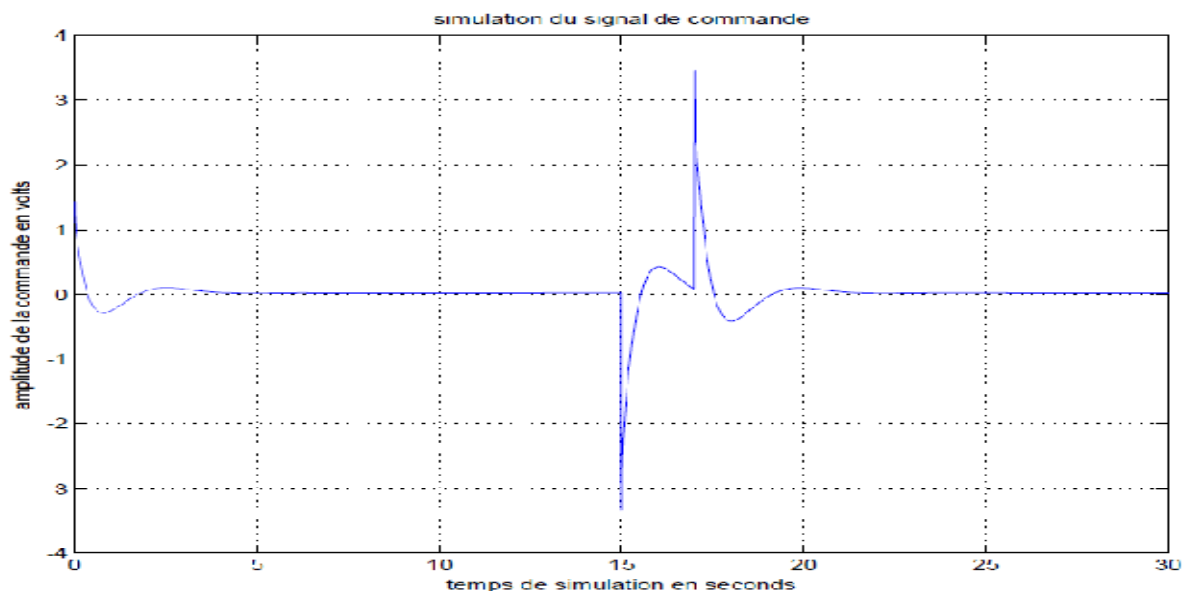


Figure 3.33 : Signal de commande u .

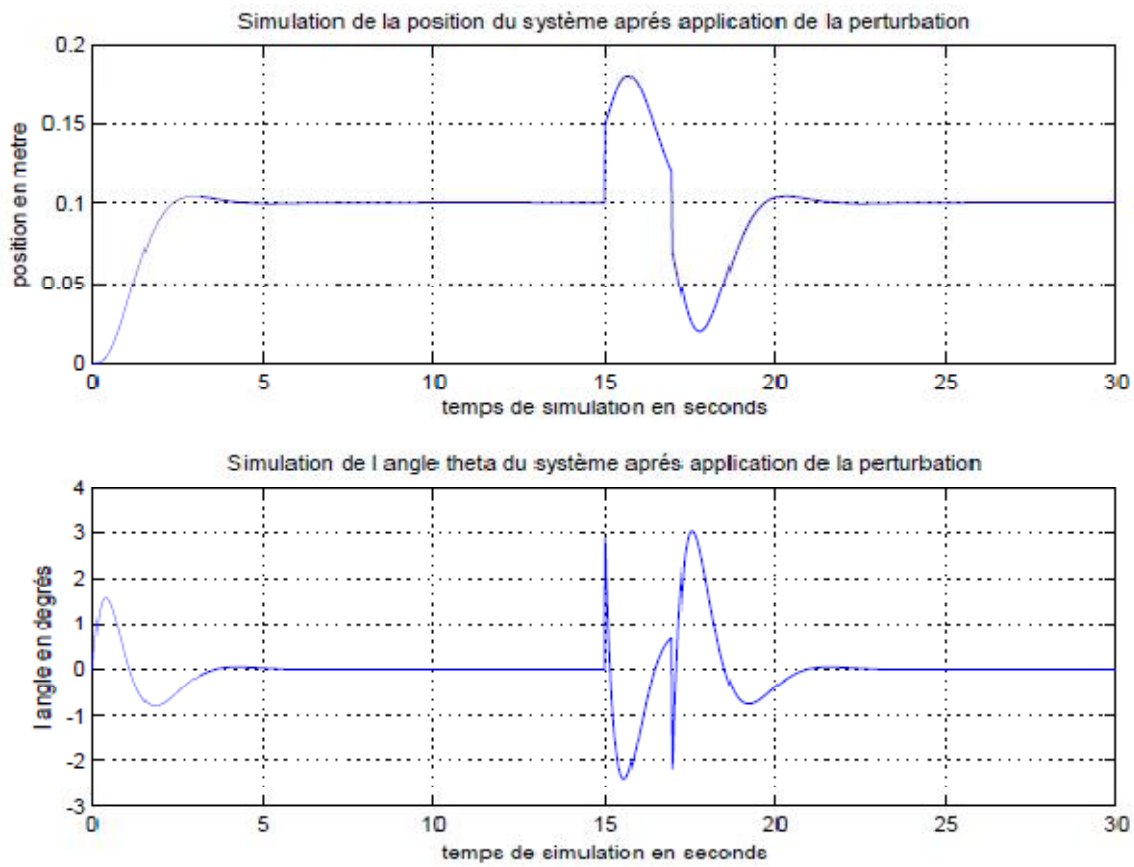


Figure 3.34 : Réponse du modèle après application de la commande LQR en présence de perturbation.

Dans ce cas ci, le temps nécessaire au rejet de la perturbation est de prêt de seconds.

Les pics maximaux après l'application de la perturbation atteignent 0.18 m pour la position, 3 degrés pour l'angle θ et 3.5 volts pour la commande u . Par contre les minimaux sont de 0.025 m pour la position, -2.4 degrés et enfin -3.3 volts pour le signal de commande u .

ii. Deuxième choix des matrices de pondération

Les matrices de pondération sont :

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}, \quad R = 0.05$$

Le gain correspondant à ce choix de matrice est comme suit :

$$K_c = [13.8384 \quad 24.9002 \quad 56.0167 \quad 9.4357]$$

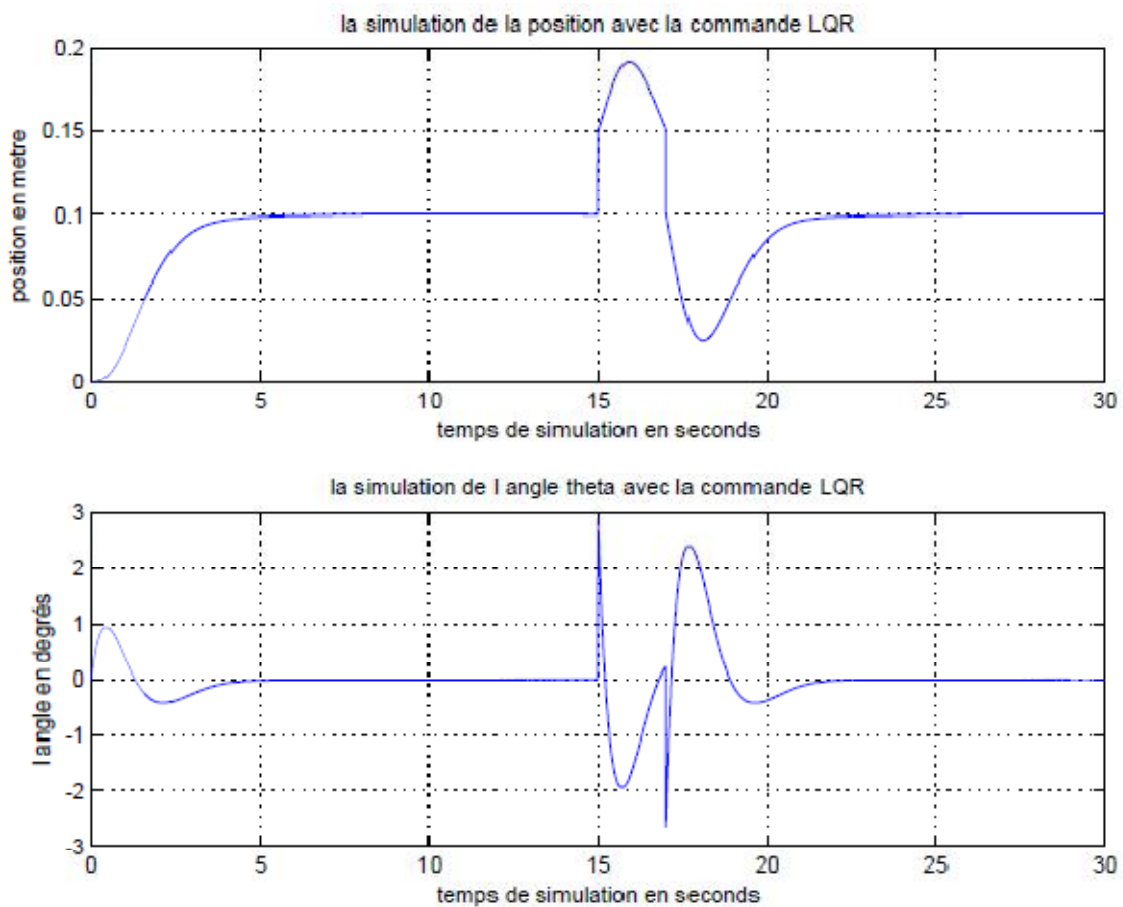


Figure 3.35 : Réponse du modèle après application de la commande LQR en présence de perturbation.

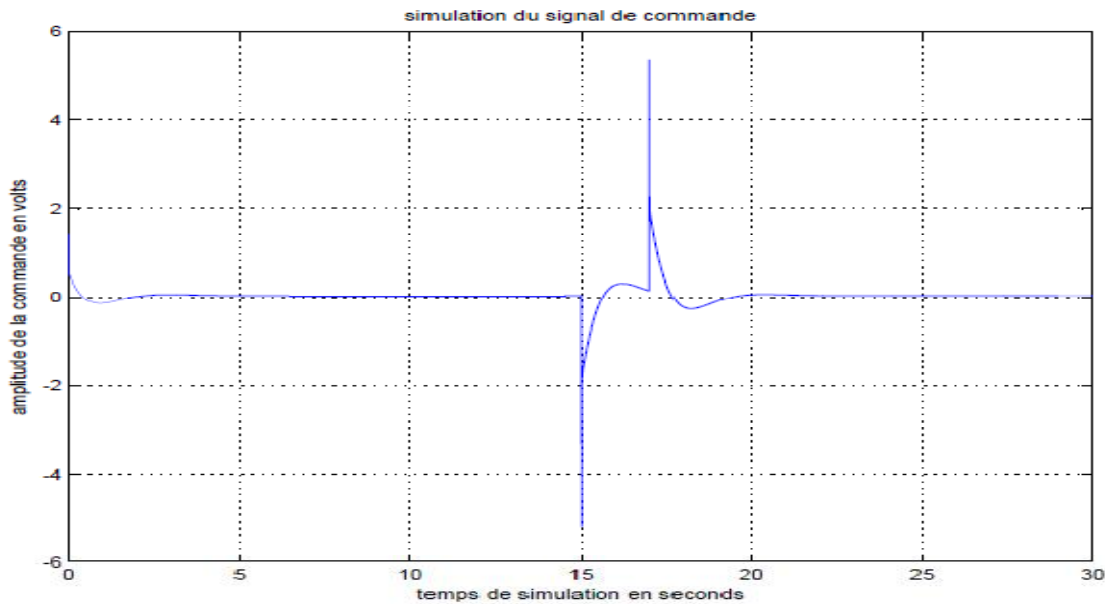


Figure 3.36 : signal de commande u .

Dans ce cas ci, le temps nécessaire au rejet de la perturbation est de prêt de 5 seconds. C'est comme pour le cas précédent sauf que la commande délivrée au moteur est un peu plus importante ici.

Les pics maximaux après l'application de la perturbation, ils atteignent 0.19 m pour la position, 3 degrés pour l'angle θ et 5.2 volts pour la commande u . Par contre les minimaux sont de 0.03 m pour la position, -2.5 degrés et enfin -5.2 volts pour le signal de commande u .

Pour les deux choix des matrices de pondération, le rejet des perturbations s'effectue en un temps environnant les 5 seconds avec une allure presque identique avec des valeurs de pic plus ou moins différentes.

Conclusion :

Le fait de vouloir avoir une erreur statique nulle avec un temps de réponse faible et pas de dépassement conduit à devoir délivrer des commandes, par le moteur, avec des énergies assez fortes.

3.4 Commande LQG

3.4.1 Commande LQG avec observateur de KALMAN

Comme dans les commandes précédentes on utilisera les mêmes valeurs pour les bruits. [Wan]

$$W = 100 \quad , \quad V = 0.0001 * \begin{bmatrix} 0.5 & 0 \\ 0 & 1 \end{bmatrix}$$

On obtient ceci pour le calcul du gain de KALMAN :

$$K_f = 10^3 * \begin{bmatrix} 1.1551 & 0.5770 \\ 0.0031 & 0.0007 \\ 1.1540 & 0.5781 \\ -0.0033 & -0.0016 \end{bmatrix}$$

Nous garderons aussi les valeurs calculées par la méthode LQR pour le retour d'état ainsi nous aurons deux choix de commande.

Le schéma de la commande LQG appliquée à notre système est représenté dans la figure 3.37.

i. Le premier choix du gain de la commande LQR

En choisissant les matrices de pondération Q et R dans ce cas-ci comme suit :

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad , \quad R = 0.049$$

Le vecteur gain K_c est obtenu comme suit :

$$K_c = [13.9819 \quad 16.8488 \quad 34.7857 \quad 1.7287]$$

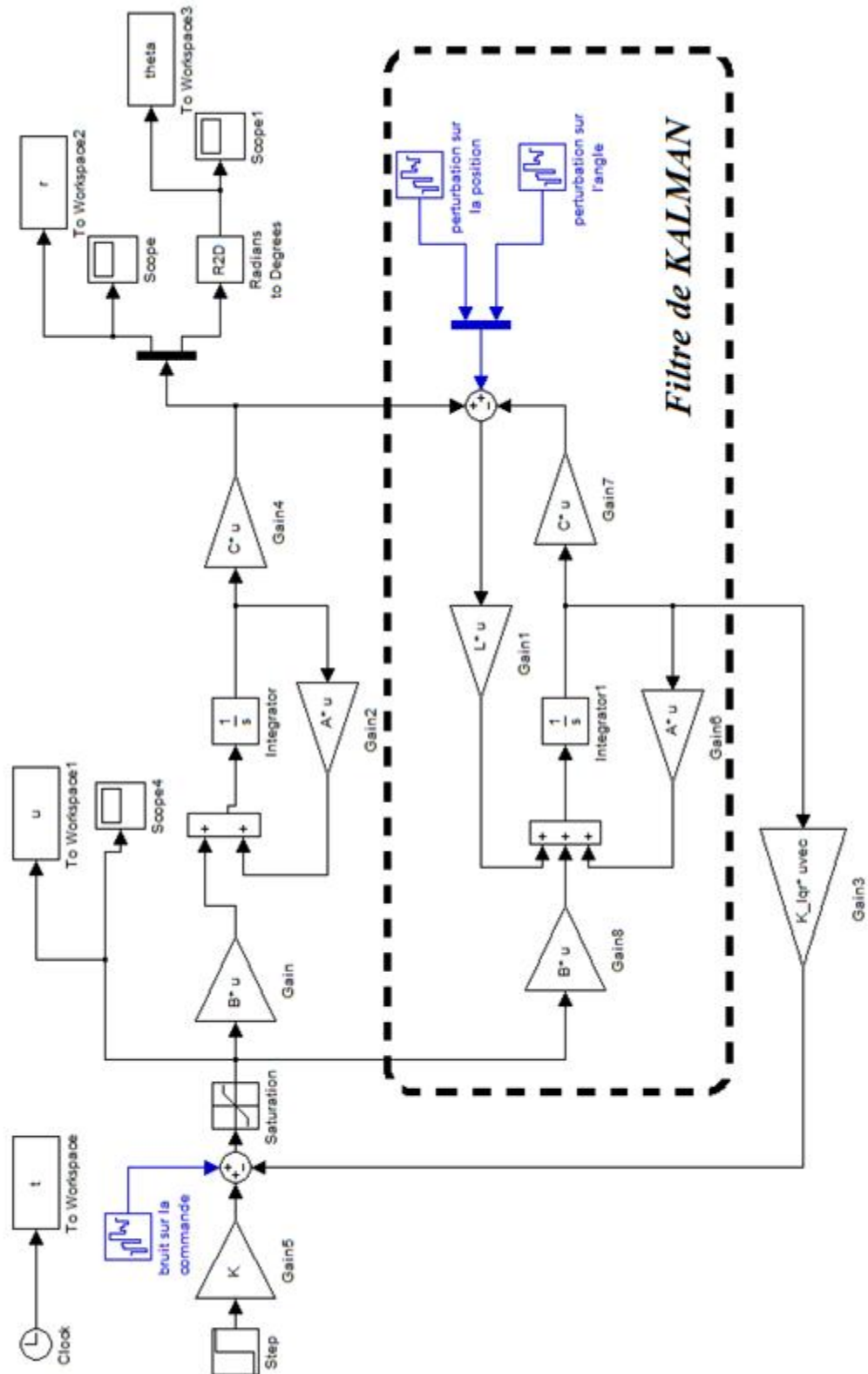


Figure 3.37 : Schéma de la commande LQG appliquée au système.

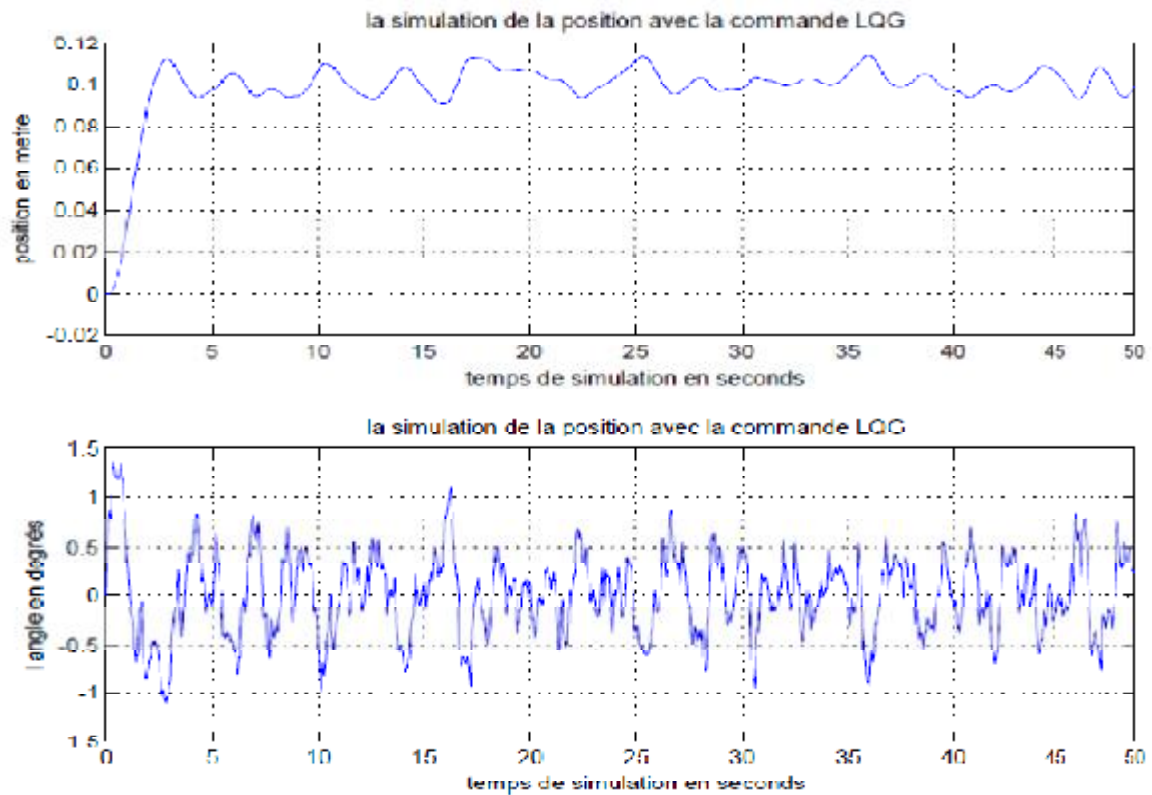


Figure 3.38 : Réponse du modèle à la commande LQG.

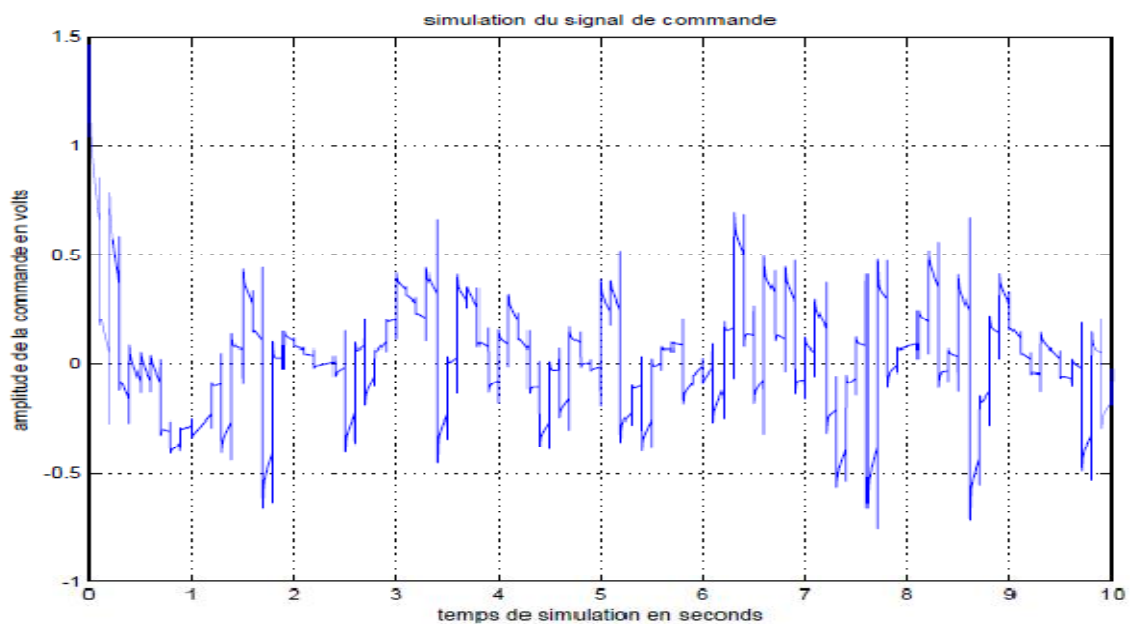


Figure 3.39 : Signal de commande u après application de la commande LQG.

On constate par rapport à la réponse de la position dans son régime permanent, illustrée dans la figure 3.38, une oscillation qui varie entre 0.09 et 0.115 mais assez lentement par rapport à la réponse du modèle étudiée précédemment (Commande LQR) ; de ce qui s'agit de la réponse de l'angle θ , la variation est située entre + et - 1.5 degrés cela reste très petit comme variation.

Quant à la figure 3.39 qui illustre la commande u , on remarque bien qu'elle est très instable avec sa variation trop brutale entre les valeurs maximal +1.5 volts et minimal -1 volt. Cela est à éviter sinon on risque de surchauffer le moteur ou bien que le moteur ne suive plus la commande.

ii. Le deuxième choix des matrices de pondérations

Pour le deuxième choix des matrices de pondérations on prendra les mêmes que ceux de la section précédentes (section 3.3.1.ii)

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}, \quad R = 0.05$$

Le gain correspondant à ce choix de matrice est comme suit :

$$K_c = [13.8384 \quad 24.9002 \quad 56.0167 \quad 9.4357]$$

La réponse du système pour cette commande est visualisée dans la figure suivante :

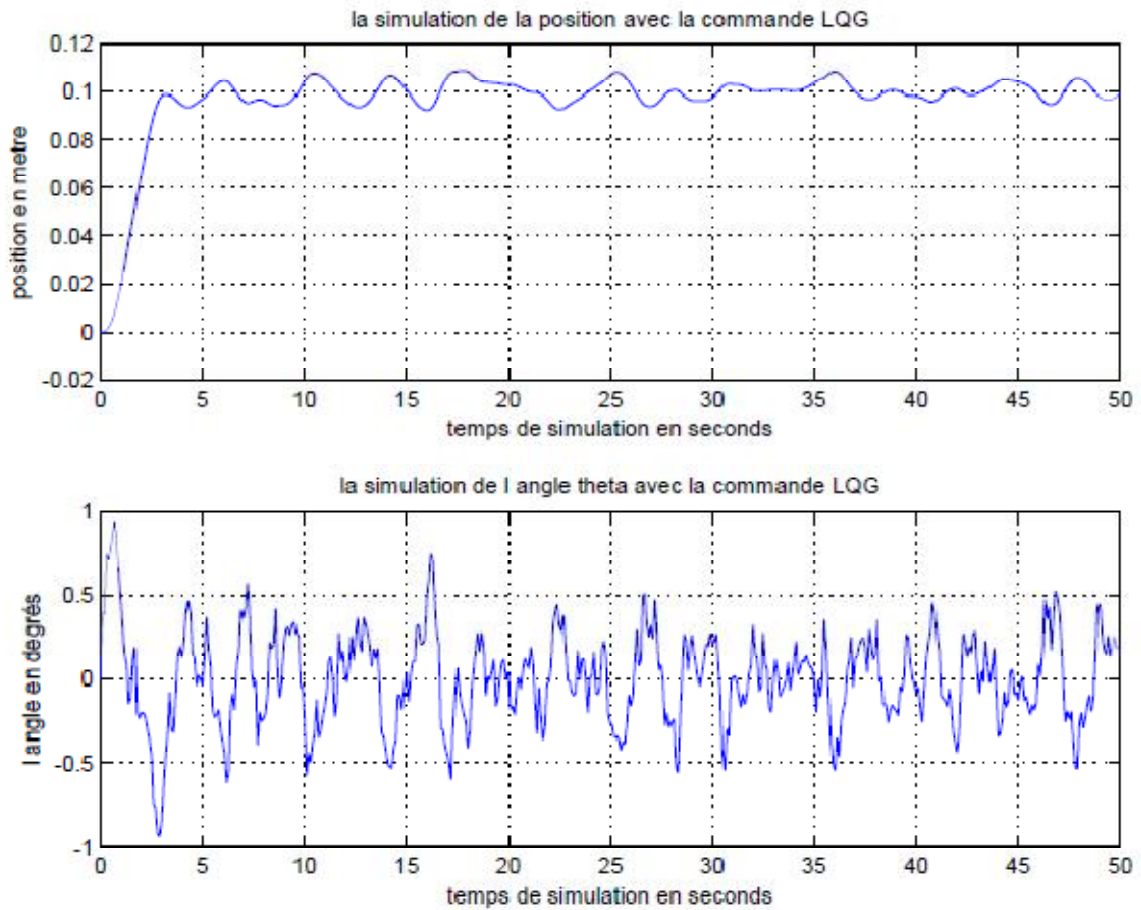


Figure 3.40 : Réponse du système à la commande LQG.

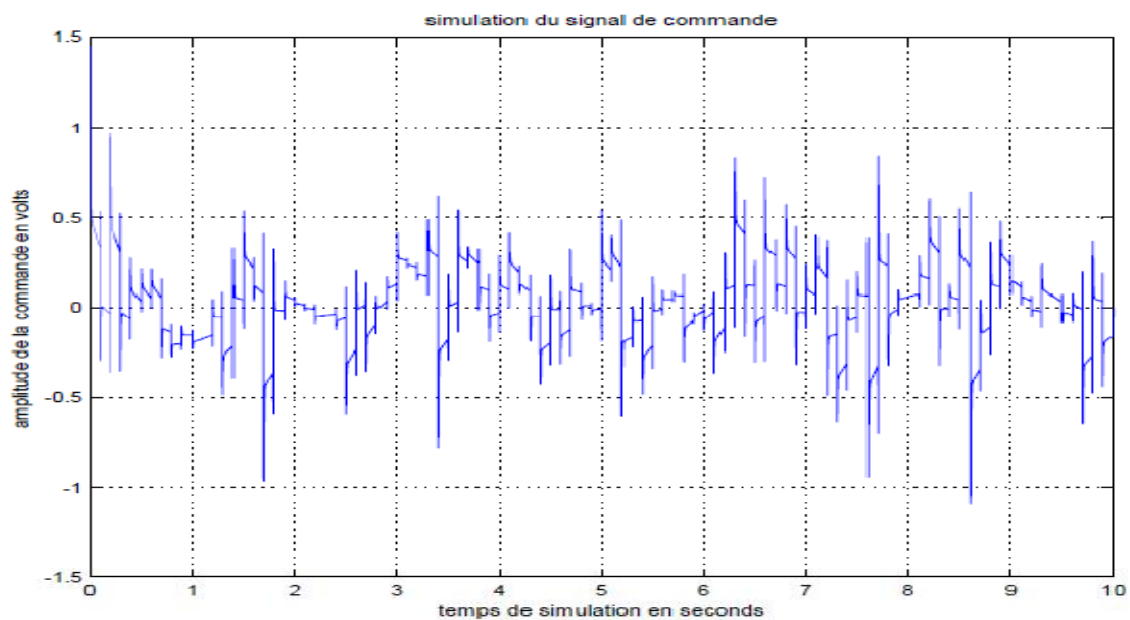


Figure 3.41 : Signal de commande u après application de la commande LQG.

Pour la figure 3.40, on constate par rapporte à la réponse de la position, dans son régime permanent, une oscillation qui varie entre 0.09 et 0.11; de ce qui s'agit de la réponse de l'angle θ , la variation est située entre + et – 1 degré cela reste assez petit comme variation.

La balle oscille incessamment autour de la position désirée sans pour autant se stabilise complètement.

Concernant la commande u , qui est illustrée dans la figure 3.41, nous obtenons un signal “presque identique” à celui représenté dans la figure 3.39 avec des variations plus ou moins prononcé dans ce cas ci.

3.4.2 Commande LQG avec un observateur réduit de KALMAN

On considérant que les états position et angle peuvent être mesurable alors dans ce cas on construira un observateur pour les états vitesses normale et angulaire.

Les paramètres suivants ont été choisis pour le calcul du gain de l'observateur [Wan]:

$$W = 100 \quad , \quad V = 0.0001 * \begin{bmatrix} 0.5 & 0 \\ 0 & 1 \end{bmatrix}$$

On remarque bien que c'est les mêmes paramètres choisis pour le calcul de l'observateur de KALMAN précédemment.

Le gain de l'observateur est obtenu comme suit :

$$K_{f_réduit} = 10^3 * \begin{bmatrix} 1.1813 & 0.5498 \\ 1.0996 & 0.5325 \end{bmatrix}$$

Le schéma de commande correspondant est illustré dans la figure 3.42 :

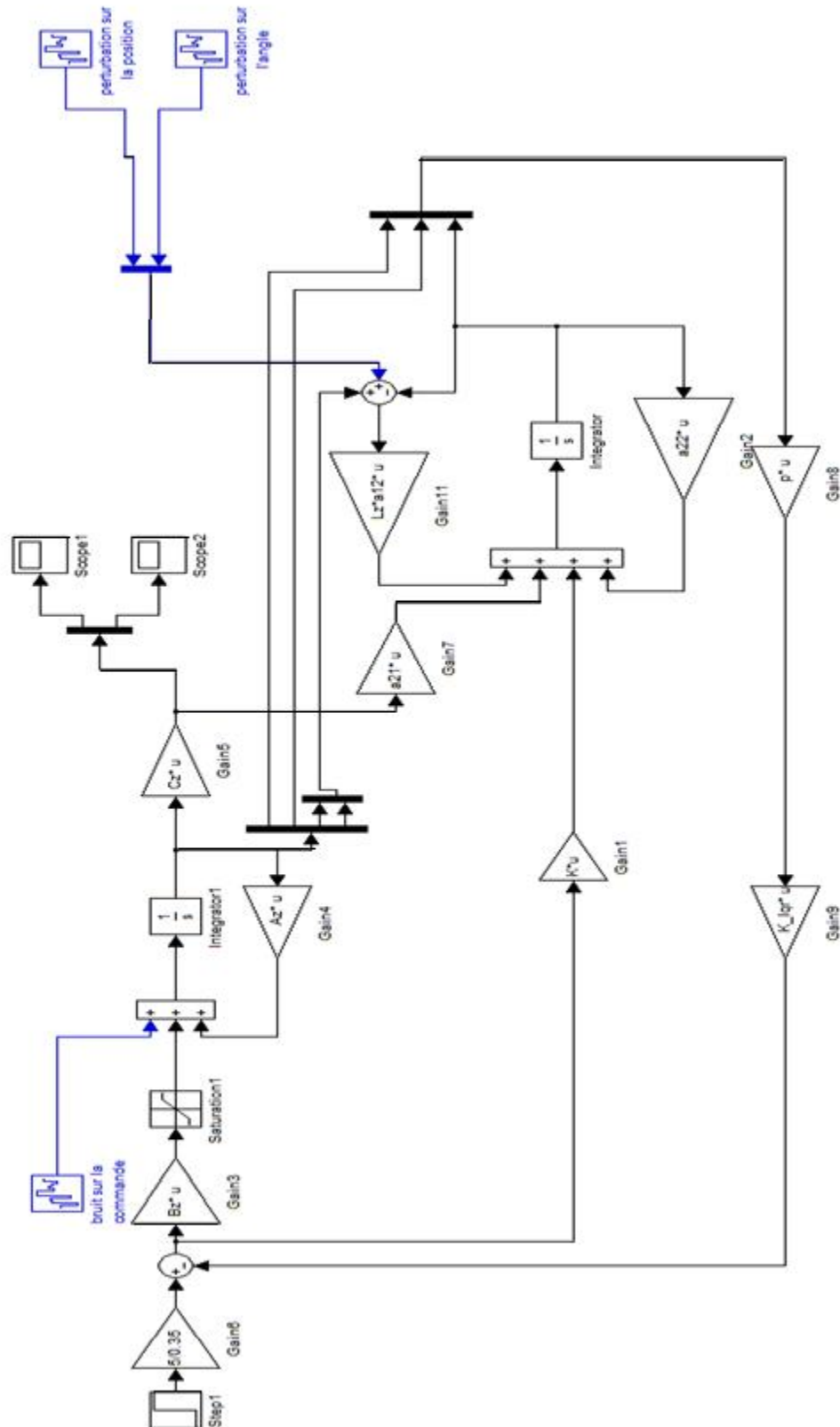


Figure 3.42 : Schéma de la commande LQG avec un observateur de KALMAN réduit.

i. Le premier choix du gain de la commande LQR

En choisissant les matrices de pondération Q et R dans ce cas-ci comme suit :

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R = 0.049$$

Le vecteur gain K_c est obtenu comme suit :

$$K_c = [13.9819 \quad 16.8488 \quad 34.7857 \quad 1.7287]$$

La réponse du système à cette commande est représentée dans la figure 3.43:

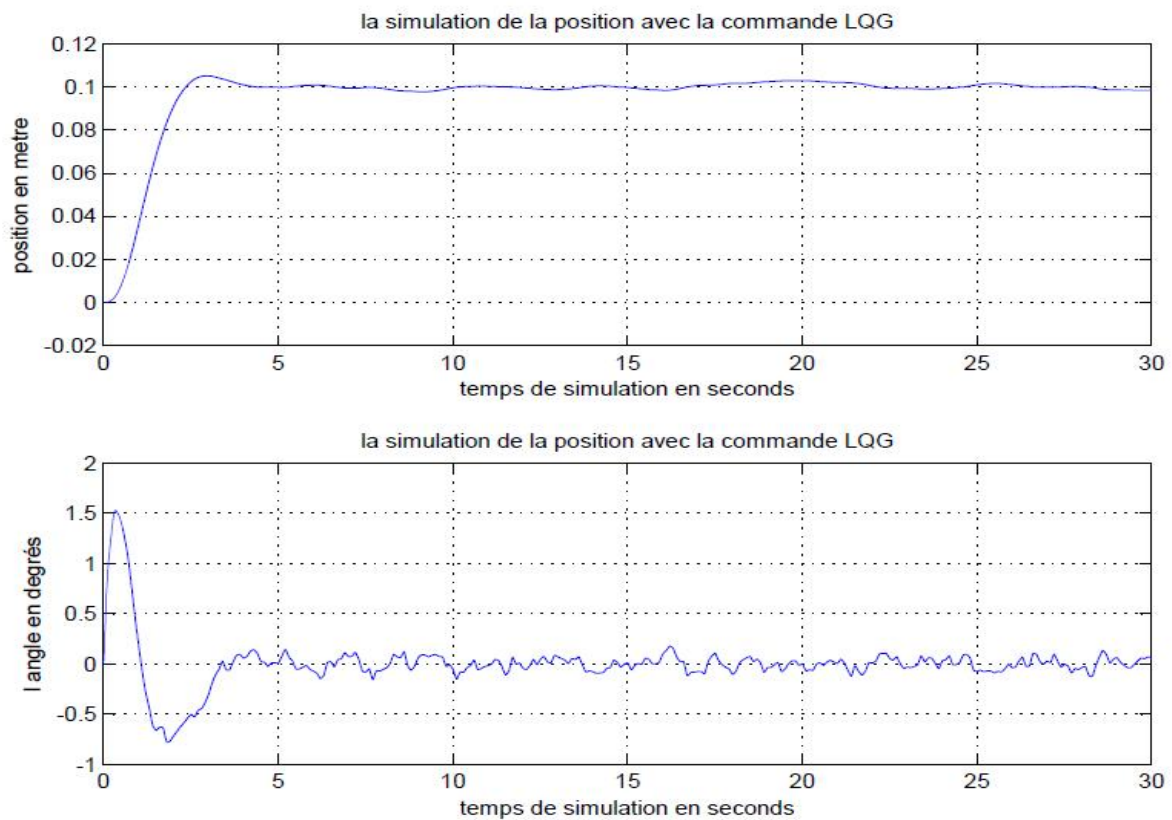


Figure 3.43: Réponse du système à la commande LQG avec observateur réduit.

Le signal de commande u après l'application de cette dernière commande est représenté dans la figure 3.44

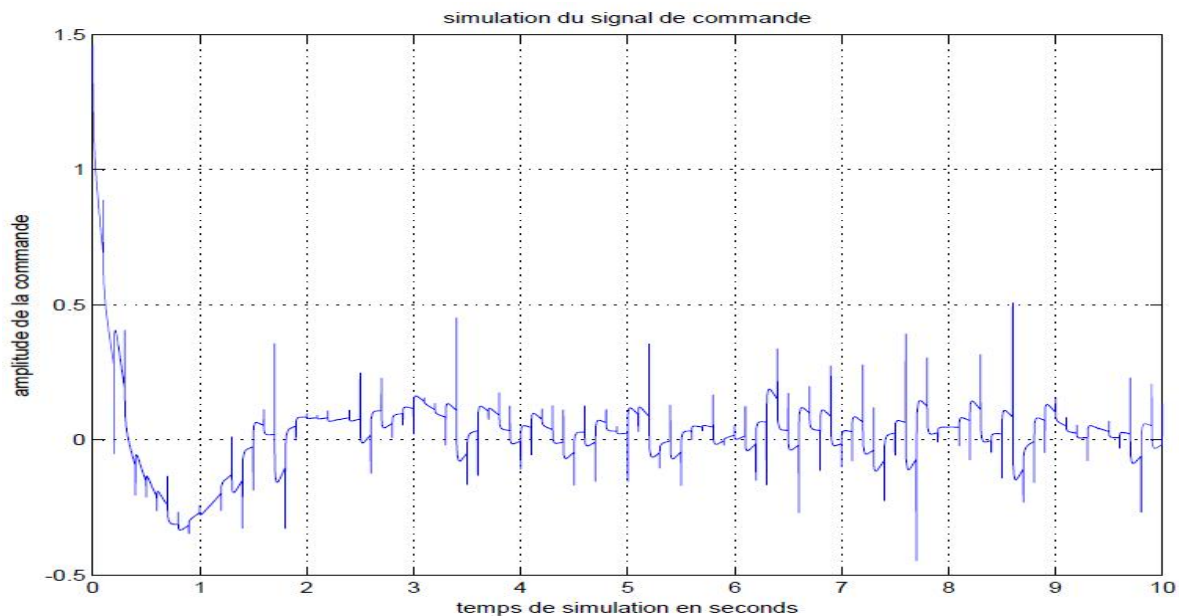


Figure 3.44 : Signal de commande u après l'application de la commande LQG avec observateur réduit.

On constate par rapporte à la réponse de la position un dépassement de prêt de 5,2%, quant au régime permanent on remarque bien que la réponse varie entre 0.097 et 0.103 m, donc on a un bon rejet des bruits appliqués au système; en ce qui s'agit de la réponse de l'angle θ , la variation est située entre + et $- 0.2$ degrés cela est très petit comme variation.

Quant à la commande u (figure 3.44) on remarque bien qu'elle est instable, dans son régime permanent, par sa variation brutale dans un intervalle très petit allant de -0.5 volts à +0.6 volts.

ii. Le deuxième choix des matrices de pondération

Les matrices de pondération sont :

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}, \quad R = 0.05$$

Le gain correspondant à ce choix de matrice est comme suit :

$$K_c = [13.8384 \quad 24.9002 \quad 56.0167 \quad 9.4357]$$

Pour cette commande on illustre si après la réponse du système dans la figure 3.45 :

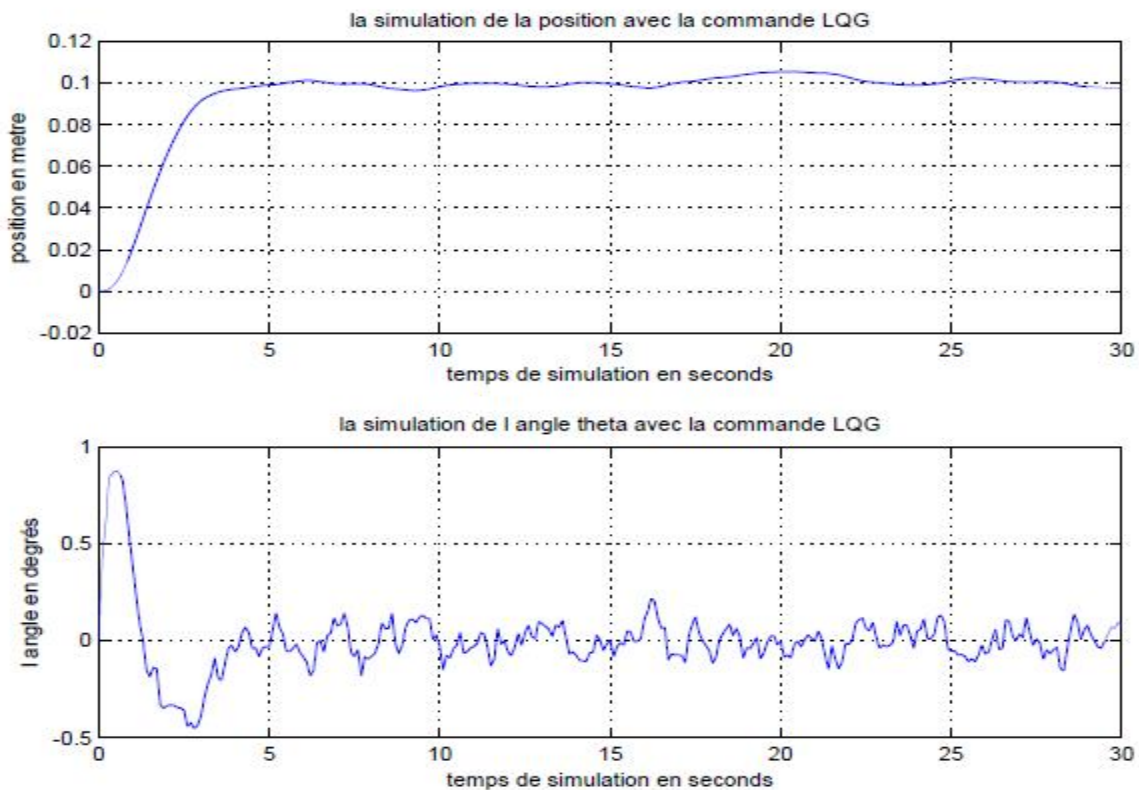


Figure 3.45: Réponse du système à la commande LQG avec observateur réduit.

Le signal de commande est représenté quant à lui dans la figure 3.46 :

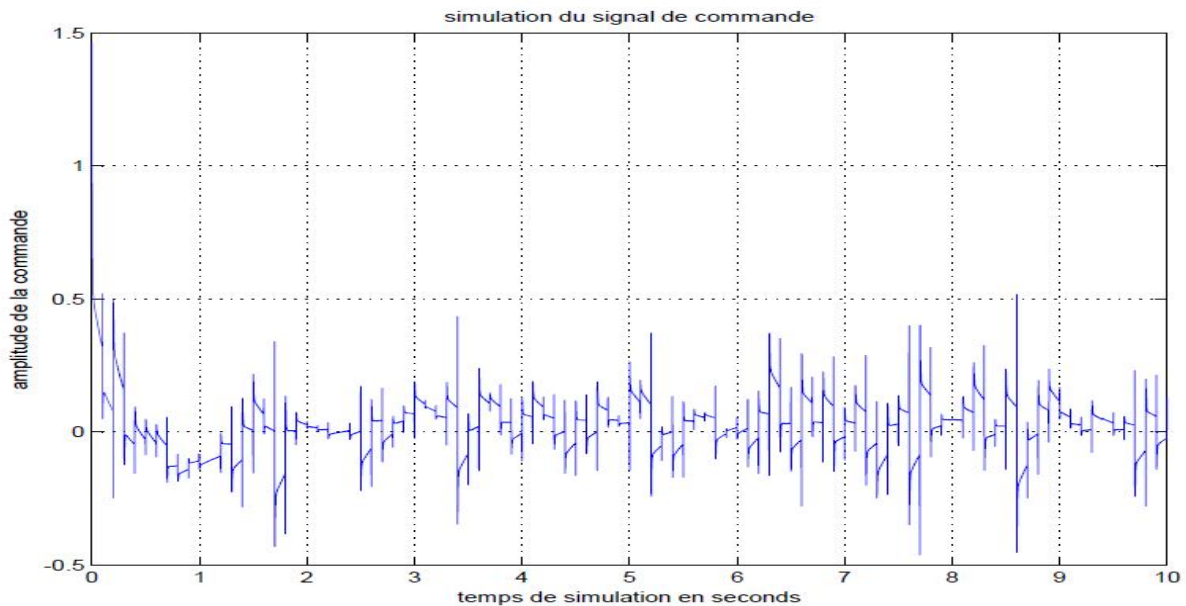


Figure 3.46: Signal de commande u après application de la commande LQG avec un observateur réduit.

On constate par rapporte à la réponse de la position, figure 3.45, dans son régime permanent une très petite variation entre 0.096 et 0.106 et pas de dépassement cela prouve l'efficacité de la commande à rejeter les bruits qui ont été appliqués au système; en ce qui s'agit de la réponse de l'angle θ , la variation est située entre + et $- 0.2$ degré cela est très minime comme variation. Quant à la commande u , illustré dans la figure 3.46, on remarque bien qu'elle est très instable de par sa variation top brutale dans son régime permanent entre les valeurs maximal +0.6 volts et minimal -0.5 volts.

3.5 Conclusion

Si on devait classer les commandes appliquées précédemment par leurs performances la première place sera attribuée à la commande LQR pour son excellent rejet des bruits en deuxième position ça sera la commande par placement de pôles qui fourni de bon résultat qui sont en général acceptable. Quand à la commande LQG, qui sera classé bon troisième on remarque bien la perte de robustesse de cette commande et ce qui fait que les bruits ne sont pas aussi bien rejeté que pour les commande précédente et surtout pour le signal de commande u qui est très brutale dans ces variations avec une grande rapidité.

L'introduction des observateurs, que se soit l'observateur d'état introduit à la commande par placement de pôles ou bien le filtre de Kalman combiné à la commande LQR, détériore les performances du système. Ceci est du aux bruits affectant les observateurs en même temps que le modèle d'état.

CHAPITRE 4 : Etude des performances

4.1 Introduction

Dans le chapitre précédent on a visualisé les différentes sorties du modèle ainsi que la commande u délivrée au moteur, dans ce chapitre on va étudier la stabilité et les performances de toutes les commandes étudié précédemment. Ceci se fera en traçant les lieux de Nyquist de la boucle ouverte du modèle corrigé, les fonctions de transfert en boucle fermée seront données pour faire part des pôles et des zéros du modèle.

4.2 Fonctions de transfert en boucles fermée du modèle corrigé

Dans ce qui suit, nous allons donner les fonctions de transferts du modèle corrigé, ceci se fera pour chacune des commandes appliquées, nous donnerons aussi leurs pôles et zéros dans des tableaux pour une comparaison de ces derniers.

4.2.1 Commande par placement de pôles

Pour cette commande nous obtenons les fonctions de transfert en boucle fermée du modèle d'état corrigé comme suit :

- Pour le premier choix des pôles où nous avons obtenu le retour d'état suivant :

$$K_p = [4.4118 \quad 5.3480 \quad 7.7746 \quad -5.0456] \quad (4.1)$$

La fonction de transfert en boucle fermée correspondante est:

$$H(s) = \begin{bmatrix} \frac{5.029 * 10^{-14} s^3 + 5.365 * 10^{-13} s^2 + 1.878 * 10^{-12} s + 300}{s^4 + 20 s^3 + 131 s^2 + 340 s + 300} \\ \frac{7.105 * 10^{-15} s^3 + 16.85 s^2 + 3.979 * 10^{-13} s + 1.137 * 10^{-13}}{s^4 + 20 s^3 + 131 s^2 + 340 s + 300} \end{bmatrix} \quad (4.2)$$

- Pour le deuxième choix des pôles nous donnant le retour d'état suivant :

$$K_p = [62.6106 \quad 25.1671 \quad 14.8371 \quad -4.8675] \quad (4.3)$$

La fonction de transfert en boucle fermée correspondante est comme suit:

$$H(s) = \left[\begin{array}{c} \frac{4.471 * 10^{-13} s^3 + 1.252 * 10^{-11} s^2 + 2.861 * 10^{-11} s + 4000}{s^4 + 23 s^3 + 250 s^2 + 1600 s + 4000} \\ \frac{7.105 * 10^{-15} s^3 + 16.85 s^2 + 3.638e * 10^{-12} s + 1.137 * 10^{-11}}{s^4 + 23 s^3 + 250 s^2 + 1600 s + 4000} \end{array} \right] \quad (4.4)$$

Les zéros des deux fonctions de transfert données par les expressions (4.2) et (4.4) sont donnés dans le tableau 4.1 suivant :

<i>1^{er} choix des pôles</i>	<i>2^{ème} choix des pôles</i>
$Z_{\text{position}} = 10^5 * \begin{bmatrix} -1.8136 \\ 0.9068 + 1.5706i \\ 0.9068 - 1.5706i \end{bmatrix}$	$Z_{\text{position}} = 10^5 * \begin{bmatrix} -2.0761 \\ 1.0379 + 1.7979i \\ 1.0379 - 1.7979i \end{bmatrix}$
$Z_{\text{angle}} = 10^{15} * \begin{bmatrix} -2.3714 \\ -0.0000 + 0.0000i \\ -0.0000 - 0.0000i \end{bmatrix}$	$Z_{\text{angle}} = 10^{15} * \begin{bmatrix} -2.3714 \\ -0.0000 + 0.0000i \\ -0.0000 - 0.0000i \end{bmatrix}$

Tableau 4.1 : Zéros des fonctions de transfert du modèle d'état corrigé par placement de pôles

On remarque bien que dans les deux cas les zéros sont presque identiques. Cela est dû au fait que les vecteurs des pôles ont été choisis de manière à ce qu'ils soient proche.

La réponse de la position en boucle fermée est à non minimum de phase dû à l'apparition de zéros « instables ».

4.2.2 Commande par placement de pôles utilisée avec un observateur d'état

La fonction de transfert du correcteur composée du gain de retour Kp et de la fonction de transfert de l'observateur d'état est comme suit :

- Pour le premier choix des pôles, nous obtenons la fonction de transfert du correcteur comme suit :

$$K_p(s)^T = \begin{bmatrix} \frac{5494 s^3 + 8.493 * 10^5 s^2 + 2.723 * 10^7 s + 2.187 * 10^7}{s^4 + 135 s^3 + 6328 s^2 + 1.2 * 10^5 s + 7.808 * 10^5} \\ \frac{1935 s^3 + 1.685 * 10^5 s^2 + 4.643 * 10^6 s + 4.295 * 10^7}{s^4 + 135 s^3 + 6328 s^2 + 1.2 * 10^5 s + 7.808 * 10^5} \end{bmatrix} \quad (4.5)$$

La fonction de transfert en boucle fermée correspondante à cette commande est :

$$H(s) = \begin{bmatrix} \frac{8.527 * 10^{-14} s^7 + 1.455 * 10^{-11} s^6 + 6.985 * 10^{-10} s^5 + 63.58 s^4 + 8582 s^3 + 4.023 * 10^5 s^2 + 7.63 * 10^6 s + 4.964 * 10^7}{s^8 + 240 s^7 + 2.05 * 10^4 s^6 + 8.171 * 10^5 s^5 + 1.622 * 10^7 s^4 + 1.606 * 10^8 s^3 + 7.778 * 10^8 s^2 + 1.733 * 10^9 s + 1.406 * 10^9} \\ \frac{2.558 * 10^{-13} s^7 + 16.85 s^6 + 2274 s^5 + 1.066 * 10^5 s^4 + 2.022 * 10^6 s^3 + 1.316 * 10^7 s^2 + 8.821 * 10^{-6} s + 6.127 * 10^{-5}}{s^8 + 240 s^7 + 2.05 * 10^4 s^6 + 8.171 * 10^5 s^5 + 1.622 * 10^7 s^4 + 1.606 * 10^8 s^3 + 7.778 * 10^8 s^2 + 1.733 * 10^9 s + 1.406 * 10^9} \end{bmatrix} \quad (4.6)$$

- Pour le deuxième choix des pôles, nous obtenons la fonction de transfert du correcteur suivante :

$$K_p(s)^T = \begin{bmatrix} \frac{2.852 * 10^4 s^3 + 4.452 * 10^6 s^2 + 1.473 * 10^8 s + 2.945 * 10^8}{s^4 + 138 s^3 + 6791 s^2 + 1.405 * 10^5 s + 1.038 * 10^6} \\ \frac{2.852 * 10^4 s^3 + 4.452 * 10^6 s^2 + 1.473 * 10^8 s + 2.945 * 10^8}{s^4 + 138 s^3 + 6791 s^2 + 1.405 * 10^5 s + 1.038 * 10^6} \end{bmatrix} \quad (4.7)$$

La fonction de transfert de la boucle fermée correspondante est représentée ci-dessous :

$$H(s) = \begin{bmatrix} \frac{8.527 * 10^{-14} s^7 + 3.638 * 10^{-12} s^6 + 1.164 * 10^{-10} s^5 + 63.58 s^4 + 8772 s^3 + 4.318 * 10^5 s^2 + 8.93 * 10^6 s + 6.601 * 10^7}{s^8 + 243 s^7 + 2.128 * 10^4 s^6 + 8.925 * 10^5 s^5 + 1.981 * 10^7 s^4 + 2.513 * 10^8 s^3 + 1.985 * 10^9 s^2 + 9.37 * 10^9 s + 1.874 * 10^{10}} \\ \frac{-5.684 * 10^{-14} s^7 + 16.85 s^6 + 2325 s^5 + 1.144 * 10^5 s^4 + 2.367 * 10^6 s^3 + 1.749 * 10^7 s^2 + 5.722 * 10^{-6} s + 0.0001984}{s^8 + 243 s^7 + 2.128 * 10^4 s^6 + 8.925 * 10^5 s^5 + 1.981 * 10^7 s^4 + 2.513 * 10^8 s^3 + 1.985 * 10^9 s^2 + 9.37 * 10^9 s + 1.874 * 10^{10}} \end{bmatrix} \quad (4.8)$$

Les pôles et zéros des fonctions de transfert données par les expressions (4.6) et (4.8) sont donnés dans le tableau 4.2 suivant :

<i>1^{er} choix des pôles</i>	<i>2^{ème} choix des pôles</i>
$P = \begin{bmatrix} -110 \\ -44 \\ -44 \\ -22 \\ -10 \\ -5 \\ -3 \\ -2 \end{bmatrix}$	$P = \begin{bmatrix} -110 \\ -44 \\ -44 \\ -22 \\ -10 \\ -5 \\ -4 \pm 8i \\ -4 - 8i \end{bmatrix}$
$Z_{position} = 10^4 * \begin{bmatrix} -9.0691 \\ 4.5327 + 7.8530i \\ 4.5327 - 7.8530i \\ -0.0054 \\ -0.0044 \\ -0.0022 \\ -0.0015 \end{bmatrix}$	$Z_{position} = 10^4 * \begin{bmatrix} 4.5371 + 7.8530i \\ 4.5371 - 7.8530i \\ -9.0647 \\ -0.0051 \\ -0.0044 \\ -0.0022 \\ -0.0021 \end{bmatrix}$
$Z_{angle} = 10^{13} * \begin{bmatrix} -6.5872 \\ -0.0000 \\ -0.0000 \\ -0.0000 \\ -0.0000 \\ 0.0000 + 0.0000i \\ 0.0000 - 0.0000i \end{bmatrix}$	$Z_{angle} = 10^{14} * \begin{bmatrix} 2.9642 \\ -0.0000 \\ -0.0000 \\ -0.0000 \\ -0.0000 \\ 0.0000 + 0.0000i \\ 0.0000 - 0.0000i \end{bmatrix}$

Tableau 4.2: Pôles et zéros des fonctions de transfert du modèle d'état corrigé par placement de pôles en présence de l'observateur d'état

On remarque bien, dans le tableau 4.2, que les pôles choisis pour l'observateur et le modèle d'état constituent les pôles de la fonction de transfert en boucle fermée.

Dans ce cas ci même la fonction de transfert en boucle fermée de l'angle θ possède des zéros dans la partie droite du plan complexe.

4.2.3 Commande LQR

La fonction de transfert en boucle fermée du modèle d'état corrigé par la commande LQR se présente comme suit :

- Pour le premier choix des matrices de pondération qui nous donnent le retour d'état suivant :

$$K_c = [13.9819 \quad 16.8488 \quad 34.7857 \quad 1.7287] \quad (4.9)$$

La fonction de transfert en boucle fermée correspondante est :

$$H(s) = \begin{bmatrix} \frac{-2.437 * 10^{-12} s^3 - 8.122e10^{-12} s^2 - 1.624e10^{-11} s + 908.4}{s^4 + 134.1 s^3 + 586.1 s^2 + 1071 s + 908.4} \\ \frac{8.527 * 10^{-14} s^3 + 16.85 s^2 - 2.274 * 10^{-12} s - 2.501 * 10^{-12}}{s^4 + 134.1 s^3 + 586.1 s^2 + 1071 s + 908.4} \end{bmatrix} \quad (4.10)$$

- Pour le deuxième choix des matrices de pondération qui nous donnent le retour d'état suivant :

$$K_c = [13.8384 \quad 24.9002 \quad 56.0167 \quad 9.4357] \quad (4.11)$$

La fonction de transfert en boucle fermée dans ce cas ci est :

$$H(s) = \begin{bmatrix} \frac{8.041 * 10^{-13} s^3 - 1.769 * 10^{-11} s^2 - 4.503 * 10^{-11} s + 899.3}{s^4 + 264 s^3 + 943.9 s^2 + 1583 s + 899.3} \\ \frac{2.842 * 10^{-13} s^3 + 16.85 s^2 + 1.819 * 10^{-12} s + 1.705 * 10^{-12}}{s^4 + 264 s^3 + 943.9 s^2 + 1583 s + 899.3} \end{bmatrix} \quad (4.12)$$

Les pôles et zéros des fonctions de transfert présentées par les expressions (4.10) et (4.12) sont donnés dans le tableau 4.3 suivant :

1^{er} choix des matrices de pondération	2^{ème} choix des matrices pondération
$P = 10^2 * \begin{bmatrix} -1.2969 \\ -0.0115 + 0.0139i \\ -0.0115 - 0.0139i \\ -0.0216 \end{bmatrix}$	$P = 10^2 * \begin{bmatrix} -2.6040 \\ -0.0130 + 0.0133i \\ -0.0130 - 0.0133i \\ -0.0099 \end{bmatrix}$
$Z_{position} = 10^4 * \begin{bmatrix} -3.5987 + 6.2329i \\ -3.5987 - 6.2329i \\ 7.1971 \end{bmatrix}$	$Z_{position} = 10^5 * \begin{bmatrix} 0.5191 + 0.8989i \\ 0.5191 - 0.8989i \\ -1.0379 \end{bmatrix}$
$Z_{angle} = 10^{14} * \begin{bmatrix} -1.9762 \\ 0.0000 \\ 0.0000 \end{bmatrix}$	$Z_{angle} = 10^{13} * \begin{bmatrix} -5.9285 \\ -0.0000 + 0.0000i \\ -0.0000 - 0.0000i \end{bmatrix}$

Tableau 4.3: Pôles et zéros des fonctions de transfert du modèle d'état corrigé par la commande LQR.

4.2.4 Commande LQG

La fonction de transfert du correcteur est la combinaison entre la fonction de transfert du filtre de KALMAN et le gain du retour d'état de la commande LQR, ainsi on obtient l'expression suivante du correcteur :

$$K(s) = -K_c(sI - A + BK_c + K_f C)^{-1} K_f \quad (4.13)$$

- Pour le premier choix des pondérations on obtient la fonction de transfert du correcteur comme suit :
-

$$K(s)^T = \begin{bmatrix} \frac{985.7 s^3 + 2.178 * 10^5 s^2 + 1.205 * 10^7 s + 2.315 * 10^6}{s^4 + 1.014 * 10^4 s^3 + 1.35 * 10^6 s^2 - 1.56 * 10^6 s + 7.055 * 10^5} \\ \frac{4.872 * 10^5 s^3 + 5.19 * 10^7 s^2 + 8.566 * 10^7 s + 9.898 * 10^7}{s^4 + 1.014 * 10^4 s^3 + 1.35 * 10^6 s^2 - 1.56 * 10^6 s + 7.055 * 10^5} \end{bmatrix} \quad (4.14)$$

La fonction de transfert en boucle fermée du modèle pour cette première commande est comme suit :

$$H(s) = \left[\begin{array}{l} \frac{-9.095 * 10^{-12} s^7 + 2.794 * 10^{-9} s^6 + 7.749 * 10^{-7} s^5 + 63.58 s^4 + 6.45 * 10^5 s^3 + 8.58 * 10^7 s^2 - 9.919 * 10^7 s + 4.485 * 10^7}{s^8 + 1.025 * 10^4 s^7 + 2.415 * 10^6 s^6 + 1.484 * 10^8 s^5 + 7.114 * 10^8 s^4 + 1.518 * 10^9 s^3 + 1.708 * 10^9 s^2 + 7.355 * 10^8 s + 1.61 * 10^8} \\ \frac{-3.638 * 10^{-12} s^7 + 16.85 s^6 + 1.709 * 10^5 s^5 + 2.274 * 10^7 s^4 - 2.629 * 10^7 s^3 + 1.189 * 10^7 s^2 - 0.0006485 s - 0.0001725}{s^8 + 1.025 * 10^4 s^7 + 2.415 * 10^6 s^6 + 1.484 * 10^8 s^5 + 7.114 * 10^8 s^4 + 1.518 * 10^9 s^3 + 1.708 * 10^9 s^2 + 7.355 * 10^8 s + 1.61 * 10^8} \end{array} \right] \quad (4.15)$$

- Pour ce second choix des matrices de pondération on obtient la fonction de transfert du correcteur correspondante est comme suit :

$$K(s)^T = \left[\begin{array}{l} \frac{1751 s^3 + 3.129 * 10^5 s^2 + 1.368 * 10^7 s + 2.153 * 10^6}{s^4 + 1.028 * 10^4 s^3 + 2.683 * 10^6 s^2 - 7.225 * 10^5 s + 1.233 * 10^6} \\ \frac{8.689 * 10^5 s^3 + 9.22 * 10^7 s^2 + 1.493 * 10^8 s + 1.28 * 10^8}{s^4 + 1.028 * 10^4 s^3 + 2.683 * 10^6 s^2 - 7.225 * 10^5 s + 1.233 * 10^6} \end{array} \right] \quad (4.16)$$

Quant à la fonction de transfert en boucle fermée qui correspond au correcteur décrit ci-dessus, elle est mentionnée ci après :

$$H(s) = \left[\begin{array}{l} \frac{-1.273 * 10^{-11} s^7 - 4.657 * 10^{-10} s^6 + 7.153 * 10^{-7} s^5 + 63.58 s^4 + 6.534 * 10^5 s^3 + 1.706 * 10^8 s^2 - 4.594 * 10^7 s + 7.842 * 10^7}{s^8 + 1.038 * 10^4 s^7 + 3.762 * 10^6 s^6 + 2.957 * 10^8 s^5 + 1.479 * 10^9 s^4 + 2.645 * 10^9 s^3 + 2.229 * 10^9 s^2 + 8.556 * 10^8 s + 1.61 * 10^8} \\ \frac{-1.637 * 10^{-11} s^7 + 16.85 s^6 + 1.732 * 10^5 s^5 + 4.521 * 10^7 s^4 - 1.217 * 10^7 s^3 + 2.078 * 10^7 s^2 + 0.0003695 s - 0.000134}{s^8 + 1.038 * 10^4 s^7 + 3.762 * 10^6 s^6 + 2.957 * 10^8 s^5 + 1.479 * 10^9 s^4 + 2.645 * 10^9 s^3 + 2.229 * 10^9 s^2 + 8.556 * 10^8 s + 1.61 * 10^8} \end{array} \right] \quad (4.17)$$

Les pôles et zéros des fonctions de transfert données par les expressions (4.15) et (4.17) sont mentionnés dans le tableau 4.2 suivant :

1^{er} choix des matrices de pondération	2^{ème} choix des matrices de pondération
$P = 10^3 * \begin{bmatrix} -1.7321 \\ -0.1297 \\ -0.1050 \\ -0.0022 \\ -0.0005 + 0.0017i \\ -0.0005 - 0.0017i \\ -0.0011 + 0.0014i \\ -0.0011 - 0.0014i \end{bmatrix}$	$P = 10^3 * \begin{bmatrix} -1.7321 \\ -0.2604 \\ -0.1050 \\ -0.0005 + 0.0017i \\ -0.0005 - 0.0017i \\ -0.0013 + 0.0013i \\ -0.0013 - 0.0013i \\ -0.0010 \end{bmatrix}$
$Z_{position} = 10^4 * \begin{bmatrix} 2.5040 \\ -1.1681 + 2.1247i \\ -1.1681 - 2.1247i \\ -0.1731 \\ -0.0132 \\ -0.0002 + 0.0003i \\ -0.0002 - 0.0003i \end{bmatrix}$	$Z_{position} = 10^4 * \begin{bmatrix} 3.3194 \\ -1.5822 + 2.8333i \\ -1.5822 - 2.8333i \\ -0.1732 \\ -0.0262 \\ -0.0002 + 0.0002i \\ -0.0002 - 0.0002i \end{bmatrix}$
$Z_{angle} = 10^{12} * \begin{bmatrix} 3.7053 \\ -0.0000 \\ -0.0000 \\ -0.0000 + 0.0000i \\ -0.0000 - 0.0000i \\ -0.0000 + 0.0000i \\ -0.0000 - 0.0000i \end{bmatrix}$	$Z_{angle} = 10^{13} * \begin{bmatrix} -3.7053 \\ -0.0000 \\ -0.0000 \\ -0.0000 + 0.0000i \\ -0.0000 - 0.0000i \\ -0.0000 \\ 0.0000 \end{bmatrix}$

Tableau 4.4 Pôles et zéros de la fonction de transfert en boucle fermée du modèle corrigé par la commande LQG.

4.3 Critère de Nyquist

Enoncé

Un système en boucle fermée est asymptotiquement stable à la condition nécessaire et suffisante que le diagramme de Nyquist de sa transmittance en boucle ouverte $H_{BO}(s)$ effectue autour du point -1 et dans le sens trigonométrique, un nombre de tours T égal au nombre de pôles instables P_{in} de $H_{BO}(s)$.

Remarques

Les pôles instables et les zéros à parties réelles positives sont comptés avec leur ordre de multiplicité.

L'intérêt du critère de Nyquist est d'étudier les conditions de stabilité du système asservi à partir du diagramme de Nyquist du système en boucle ouverte.

Application

Soit P le nombre de pôles instables de $1 + H_{BO}(s)$ (donc de $H_{BO}(s)$)

Soit Z le nombre de zéros à parties réelles positives de $1 + H_{BO}(s)$



Figure 4.1: Sens trigonométrique.

Théorème

Le diagramme de Nyquist de $1 + H_{BO}(s)$ effectue $T = P_{in} - Z$ tours dans le sens trigonométrique autour de l'origine.

Remarques

Les zéros de $1 + H_{BO}(s)$ étant les pôles de $H_{BF}(s)$, la condition de stabilité du système en BF impose que $1 + H_{BO}(s)$ ne possède pas de zéros à parties réelles positives c.à.d.

$$Z = 0 \Rightarrow T = P_{in}$$

En pratique, étudier la position du diagramme de Nyquist de $1 + H_{BO}(s)$ par rapport à l'origine équivaut à analyser la position du diagramme de $H_{BO}(s)$ par rapport au point -1 appelé point critique.

4.4 Lieu de Nyquist des fonctions de transfert en boucle ouverte $L(s)$

Afin de tester la stabilité des asservissements précédents nous allons tracer le diagramme de Nyquist des fonctions de transfert en boucle ouverte $1 + L(s)$ avec :

$$L(s) = K(s) * G(s) \quad (4.18)$$

Où :

$K(s)$ est la fonction de transfert du correcteur

$G(s)$ est la fonction de transfert en boucle ouverte du modèle.

Nous allons calculer le nombre de tour qu'effectue ce transfert autour de l'origine du plan et le comparer avec le nombre de pôles instables

Sachant que les zéros de la fonction de transfert " $1 + L(s)$ " sont les pôles de la fonction de transfert en boucle fermée du modèle corrigée, et que ceux-ci sont stables d'après le paragraphe 4.2 précédent, nous comparerons le nombre de pôles instables P_{in} de $L(s)$ avec le nombre de tour T effectué autour de l'origine du plan complexe.

4.4.1 Stabilité de l'asservissement par placement de pôles

- Pour le premier choix des pôles

$$L_1(s) = \frac{-85.02 s^3 + 131 s^2 + 340 s + 280.5}{s^4 + 105 s^3 - 6.75 * 10^{-14} s^2 + 4.974 * 10^{-14} s + 19.52} \quad (4.19)$$

Les pôles de cette fonction de transfert sont :

$$P = 10^2 * \begin{bmatrix} -1.0502 \\ -0.0057 \\ 0.0029 + 0.0049i \\ 0.0029 - 0.0049i \end{bmatrix} \quad (4.20)$$

Nous remarquons bien la présence de 2 pôles instables.

Le lieu de Nyquist de la fonction de transfert $1 + L_1(s)$ est représenté dans la figure 4.2 :

Le vecteur P représente aussi les valeurs propres de la matrice A vu que la fonction de transfert $L(s)$ se calcule comme suit pour un retour d'état:

$$L(s) = K * (sI - A)^{-1} * B \quad (4.21)$$

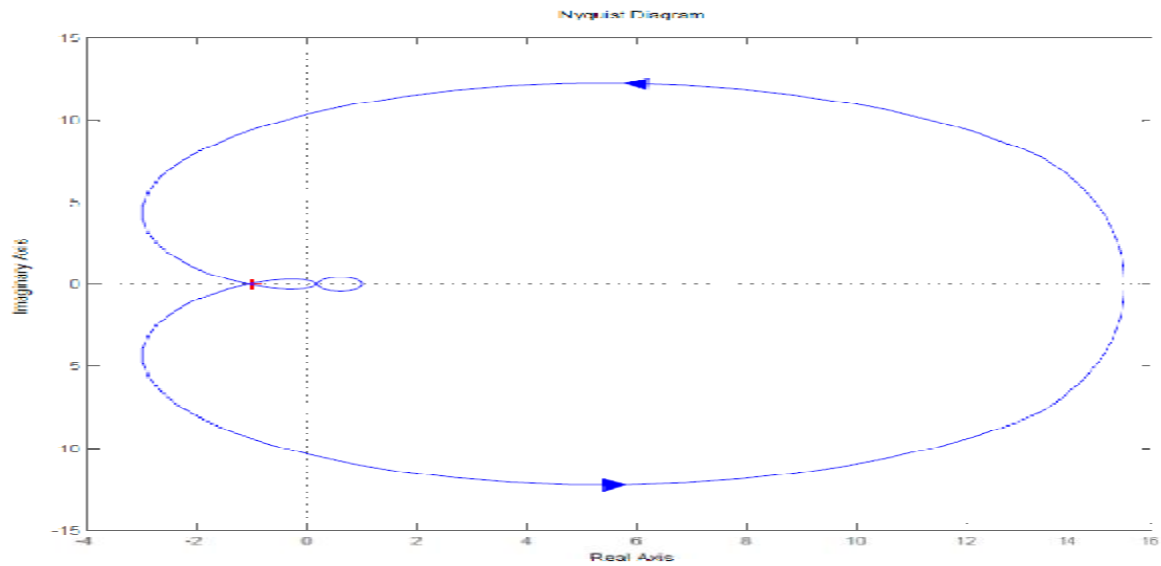


Figure 4.2: Lieu de Nyquist de la fonction de transfert $1 + L_1(s)$

Le nombre de tour qu'effectue le tracé de Nyquist dans le sens trigonométrique est égal à 2 tours. Cela coïncide bien avec le nombre de pôles instables de la fonction de transfert $L_1(s)$ donnée par l'expression (4.19).

$$P_{in} = T$$

Dans ce cas là, l'asservissement est stable.

- Pour le second choix des pôles

$$L_2(s) = \frac{-82.02 s^3 + 250 s^2 + 1600 s + 3980}{s^4 + 105 s^3 - 6.75 * 10^{-14} s^2 + 4.974 * 10^{-14} s + 19.52} \quad (4.22)$$

La fonction de transfert $L_1(s)$ et $L_2(s)$ ont le même dénominateur ce qui veut dire que les pôles de la fonction de transfert $L_2(s)$ sont ceux donnés par l'expression (4.20)

Le lieu de Nyquist de la fonction de transfert $1 + L_2(s)$ est illustré dans la figure 4.3 :

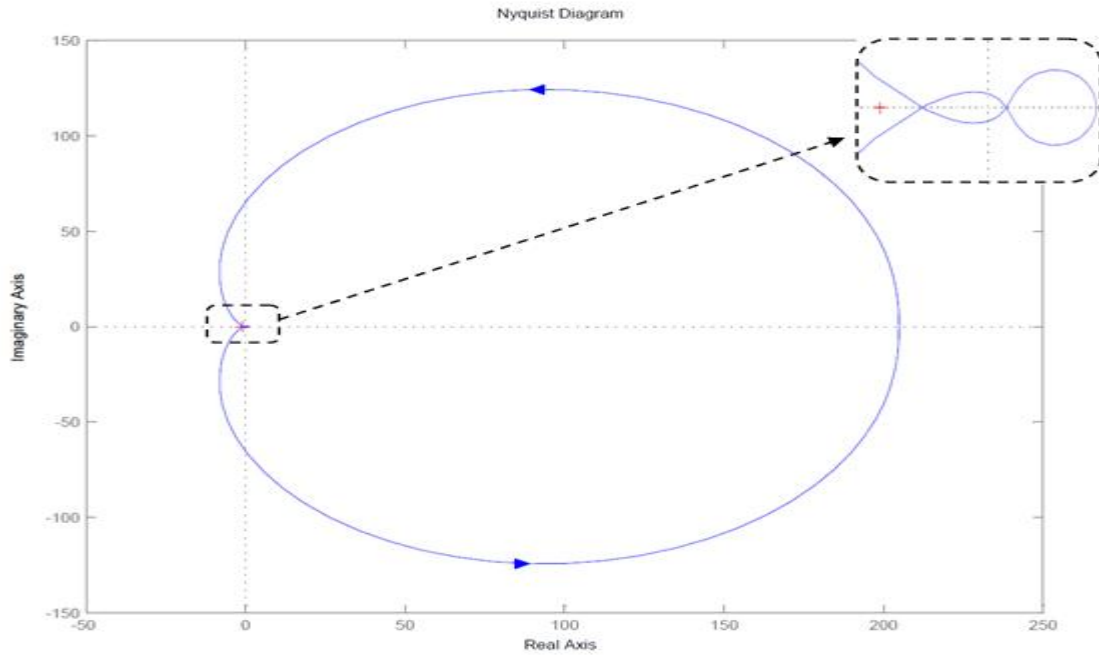


Figure 4.3 : Lieu de Nyquist de la fonction de transfert $1 + L_2(s)$

Le nombre de tour T qu'effectue le tracé de Nyquist dans le sens trigonométrique est égal à 2 tours. Et le nombre de pôles instables P_{in} de la fonction de transfert $L_2(s)$ donnée par l'expression (4.22) est aussi égal à 2.

$$P_{in} = T$$

Donc le deuxième asservissement est stable aussi.

4.4.2 Stabilité de l'asservissement par placement de pôles en présence de l'observateur d'état

- Pour le premier choix des pôles

$$L_3(s) = \frac{4.38 * 10^{-10} s^6 + 3.878 * 10^4 s^5 + 4.02 * 10^6 s^4 + 1.423 * 10^8 s^3 + 1.985 * 10^9 s^2 + 9.367 * 10^9 s + 1.872 * 10^{10}}{s^8 + 240 s^7 + 2.05 * 10^4 s^6 + 7.845 * 10^5 s^5 + 1.338 * 10^7 s^4 + 8.2 * 10^7 s^3 + 1.235 * 10^5 s^2 + 2.343 * 10^6 s + 1.524 * 10^7} \quad (4.23)$$

Les pôles de la fonction de transfert donnée par l'expression (4.23) sont :

$$P = 10^2 * \begin{bmatrix} -1.0502 \\ -0.5406 \\ -0.4400 \\ -0.2200 \\ -0.1492 \\ -0.0057 \\ 0.0029 + 0.0049i \\ 0.0029 - 0.0049i \end{bmatrix}$$

Le nombre de pôles instable est, dans ce cas ci, $P_{in} = 2$

Le lieu de Nyquist de la fonction de transfert $1 + L_3(s)$ est représenté dans la figure 4.4 :

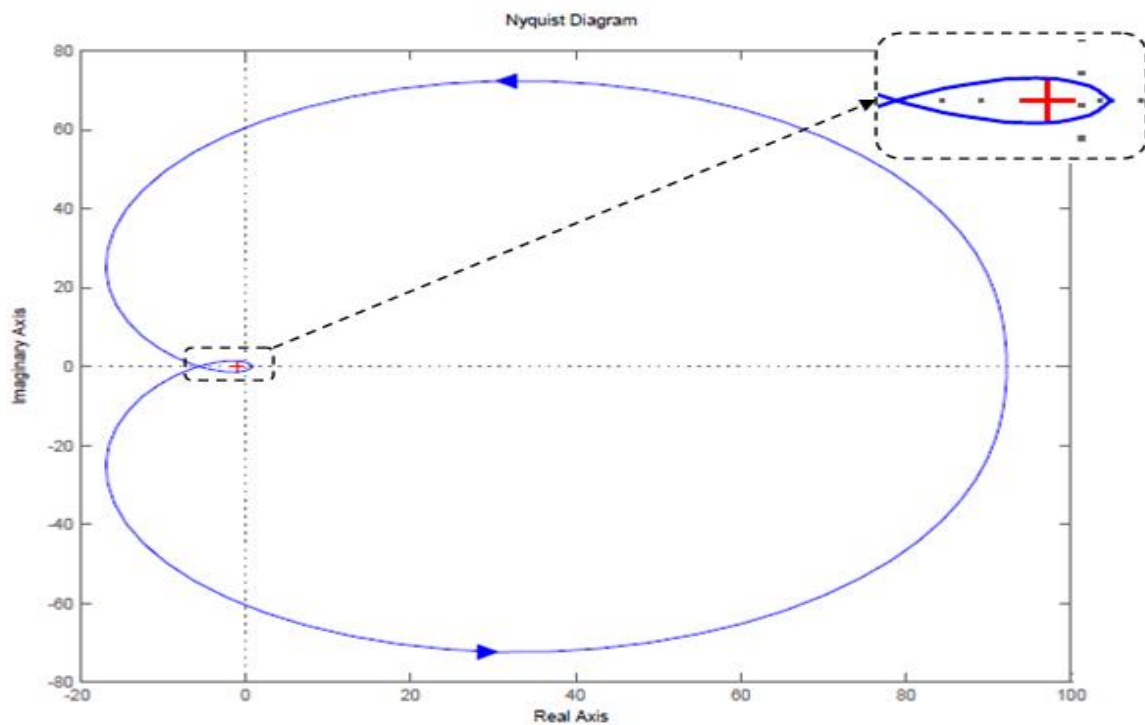


Figure 4.4 : Lieu de Nyquist de la fonction de transfert $1 + L_3(s)$

Le nombre de tour qu'effectue le tracé autour de l'origine est de $T = 2$

$P_{in} = T \Leftrightarrow$ L'asservissement est stable

- Pour le deuxième choix des pôles, la fonction de transfert s'écrit comme suit

$$L_4(s) = \frac{4.38 * 10^{-10} s^6 + 3.878 * 10^4 s^5 + 4.02 * 10^6 s^4 + 1.423 * 10^8 s^3 + 1.985 * 10^9 s^2 + 9.367 * 10^9 s + 1.872 * 10^{10}}{s^8 + 243 s^7 + 2.128 * 10^4 s^6 + 8.537 * 10^5 s^5 + 1.579 * 10^7 s^4 + 1.09 * 10^8 s^3 + 1.326 * 10^5 s^2 + 2.742 * 10^6 s + 2.027 * 10^7} \quad (4.24)$$

Les pôles de la fonction de transfert donnée par l'expression (4.24) sont :

$$P = 10^2 * \begin{bmatrix} -1.0502 \\ -0.5092 \\ -0.4400 \\ -0.2200 \\ -0.2106 \\ -0.0057 \\ 0.0029 + 0.0049i \\ 0.0029 - 0.0049i \end{bmatrix}$$

Dans ce cas ci, nous remarquons que :

$$P_{in} = 2$$

Le lieu de Nyquist de la fonction de transfert $1 + L_4(s)$ est représenté dans la figure 4.5 :

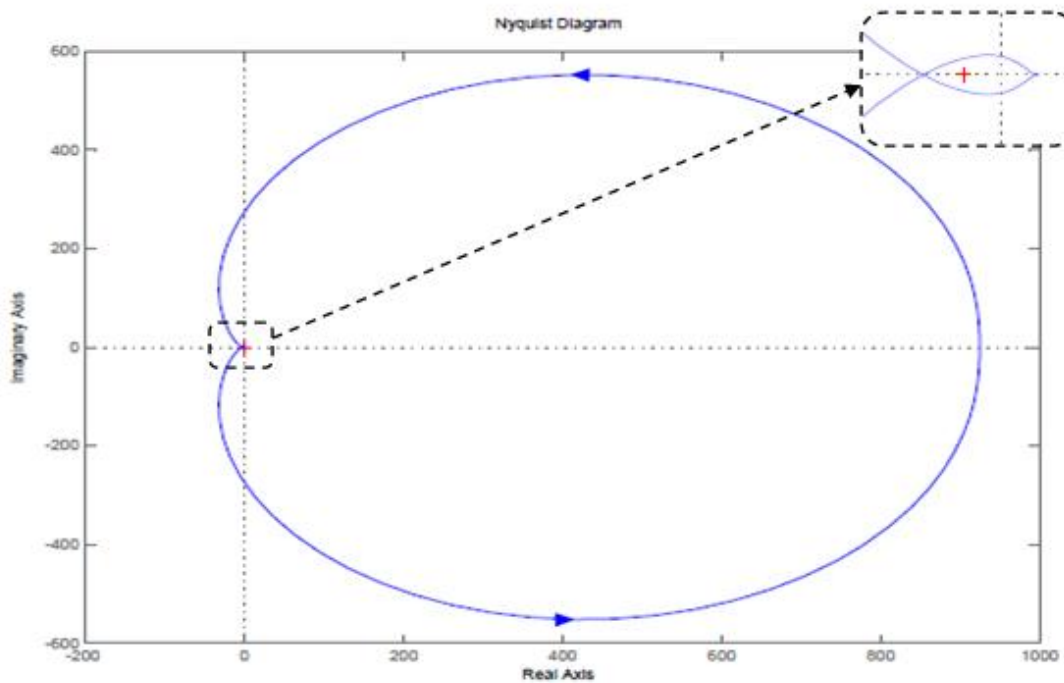


Figure 4.5 : Lieu de Nyquist de la fonction de transfert $1 + L_4(s)$

4.4.3 Stabilité de l'asservissement par la commande LQR

- Pour le premier choix des pondérations, la fonction de transfert est :

$$L_5(s) = \frac{29.13 s^3 + 586.1 s^2 + 1071 s + 888.9}{s^4 + 105 s^3 - 6.75 * 10^{-14} s^2 + 4.974 * 10^{-14} s + 19.52} \quad (4.25)$$

Les pôles de la fonction de transfert $L_5(s)$ sont donnés par l'expression (4.20)

Le lieu de Nyquist de la fonction de transfert $1 + L_5(s)$ est illustré dans la figure 4.6

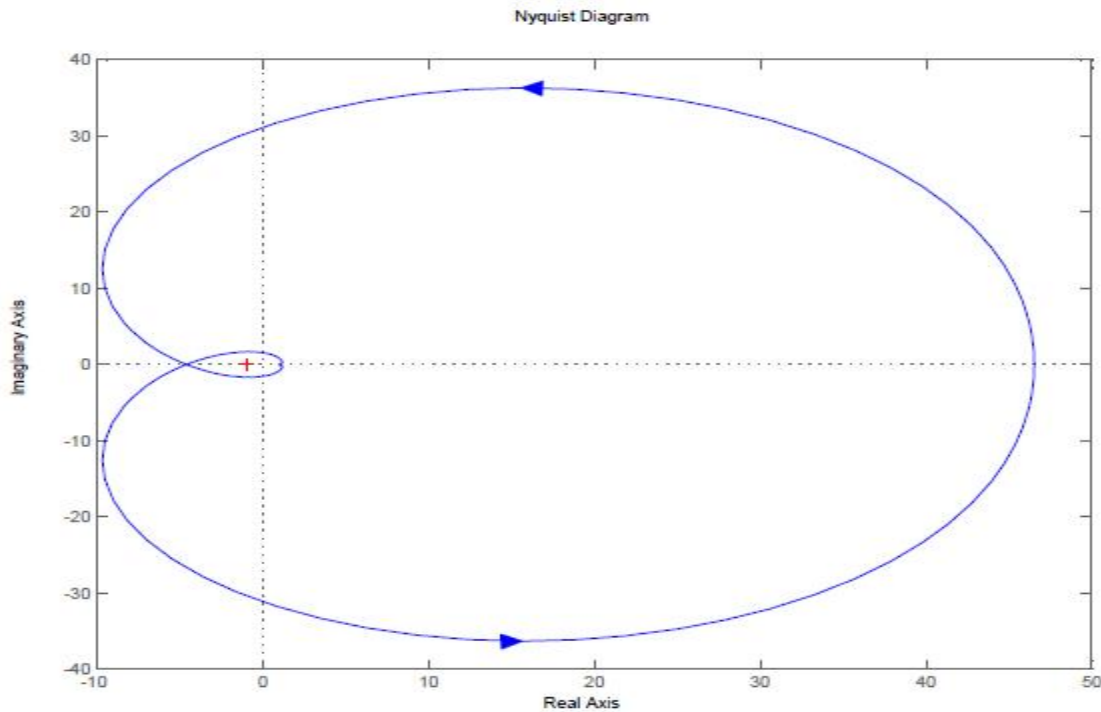


Figure 4.6 : Lieu de Nyquist de la fonction de transfert $1 + L_5(s)$

Le tracé fait un double tour sur l'origine du plan et nous avons deux pôles instables

Donc

$$P_{in} = T \Leftrightarrow \text{L'asservissement est stable.}$$

- Pour le deuxième choix des matrices de pondération :

$$L_6(s) = \frac{159 s^3 + 943.9 s^2 + 1583 s + 879.8}{s^4 + 105 s^3 - 6.75 * 10^{-14} s^2 + 4.974 * 10^{-14} s + 19.52} \quad (4.26)$$

Les pôles de la fonction de transfert $L_6(s)$ sont donnés par l'expression (4.20)

Le lieu de Nyquist de la fonction de transfert $1 + L_6(s)$ est illustré dans la figure 4.7 :

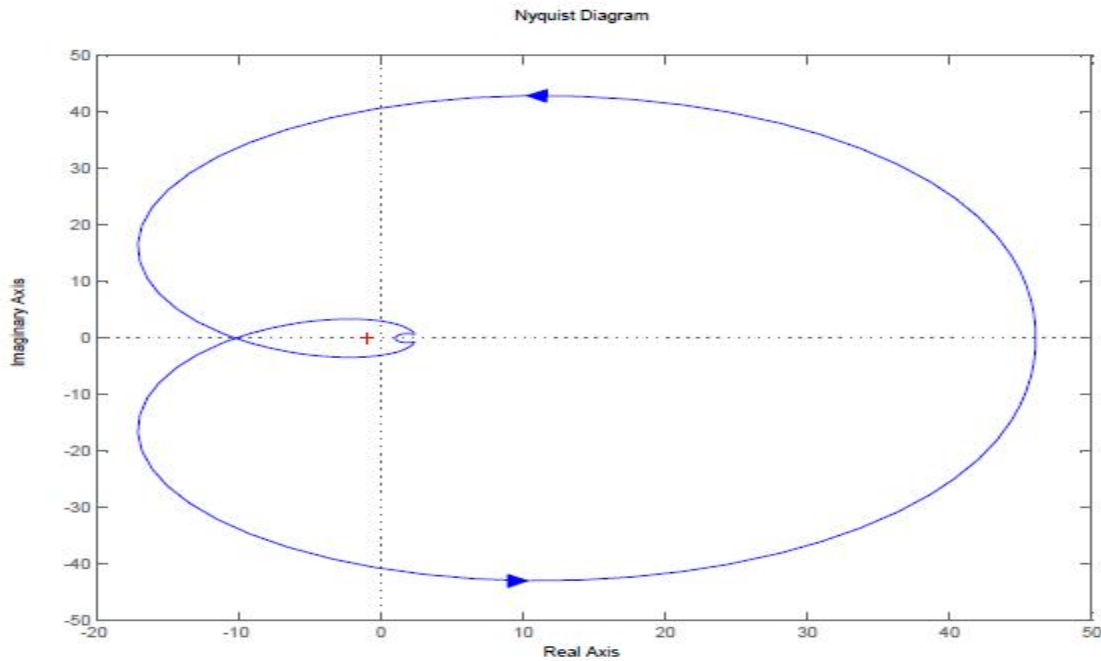


Figure 4.7 : Lieu de Nyquist de la fonction de transfert $1 + L_6(s)$

Le tracé fait un double tour sur l'origine du plan et nous avons deux pôles instables. Alors :

$$P_{in} = T \Leftrightarrow \text{L'asservissement est stable.}$$

4.4.4 Stabilité de l'asservissement pour la commande LQG

Les fonctions de transfert en boucle ouverte du modèle corrigé sont comme suit :

- Pour le premier choix des matrices de pondérations :

$$L_7(s) = \frac{1.201 * 10^{-9} s^6 + 4.749 * 10^5 s^5 + 5.123 * 10^7 s^4 + 1.56 * 10^8 s^3 + 7.043 * 10^8 s^2 + 7.657 * 10^8 s + 4.655 * 10^8}{s^8 + 1972 s^7 + 4.31 * 10^5 s^6 + 2.547 * 10^7 s^5 + 8.58 * 10^7 s^4 + 2.346 * 10^8 s^3 + 4.587 * 10^6 s^2 + 1.553 * 10^7 s + 4.36 * 10^7} \quad (4.27)$$

Les pôles de cette fonction de transfert sont :

$$P = 10^3 * \begin{bmatrix} -1.7319 \\ -0.1320 \\ -0.1050 \\ -0.0017 + 0.0026i \\ -0.0017 - 0.0026i \\ 0.0003 + 0.0005i \\ 0.0003 - 0.0005i \\ -0.0006 \end{bmatrix}$$

Le nombre de pôles instables est de :

$$P_{in} = 2$$

Le diagramme de Nyquist correspondant est représenté dans la figure 4.8 suivante :

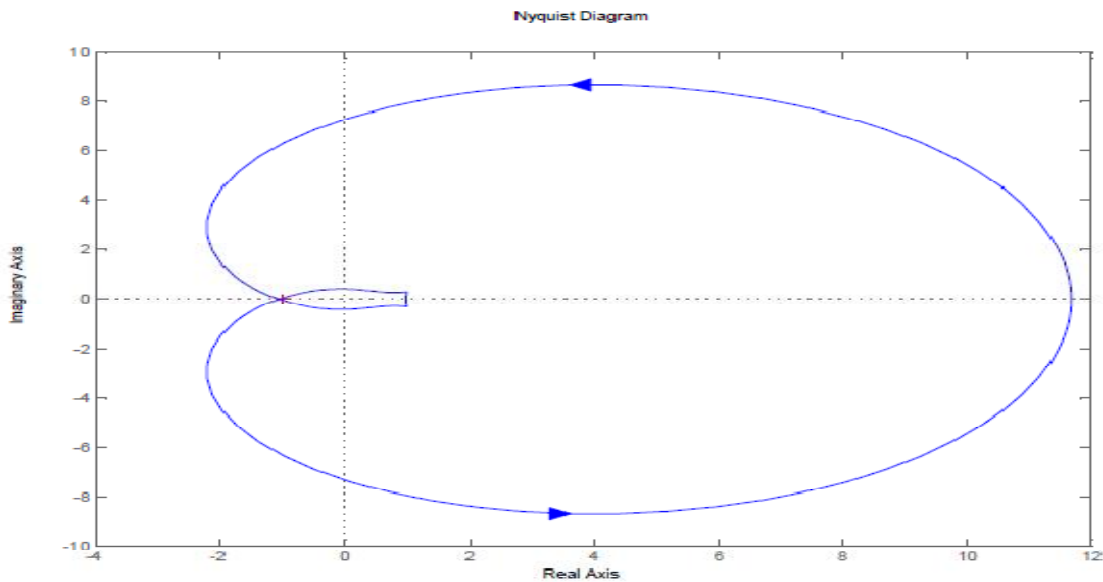


Figure 4.8 : Lieu de Nyquist de la fonction de transfert $1 + L_7(s)$

Le nombre de tour qu'effectue le tracé étant égal au nombre de pôles instable alors l'asservissement par la commande LQG, pour ce premier choix des matrices de pondération, est stable.

- Pour le second choix des matrices de pondération :

La fonction de transfert en boucle ouverte est donnée dans l'expression (4.28)

$$L_8(s) = \frac{1.72 * 10^{-9} s^6 + 6.802 * 10^5 s^5 + 7.326 * 10^7 s^4 + 2.453 * 10^8 s^3 + 1.001 * 10^9 s^2 + 1.038 * 10^9 s + 4.327 * 10^8}{s^8 + 2102 s^7 + 6.701 * 10^5 s^6 + 4.98 * 10^7 s^5 + 1.564 * 10^8 s^4 + 3.835 * 10^8 s^3 + 8.987 * 10^6 s^2 + 2.839 * 10^7 s + 7.128 * 10^7} \quad (4.28)$$

Les pôles de cette fonction de transfert sont :

$$P = 10^3 * \begin{bmatrix} -1.7318 \\ -0.2622 \\ -0.1050 \\ -0.0016 + 0.0024i \\ -0.0016 - 0.0024i \\ 0.0003 + 0.0005i \\ 0.0003 - 0.0005i \\ -0.0006 \end{bmatrix}$$

Le nombre de pôles instables est de :

$$P_{in} = 2$$

Le diagramme de Nyquist correspondant est représenté dans la figure **4.9** suivante :

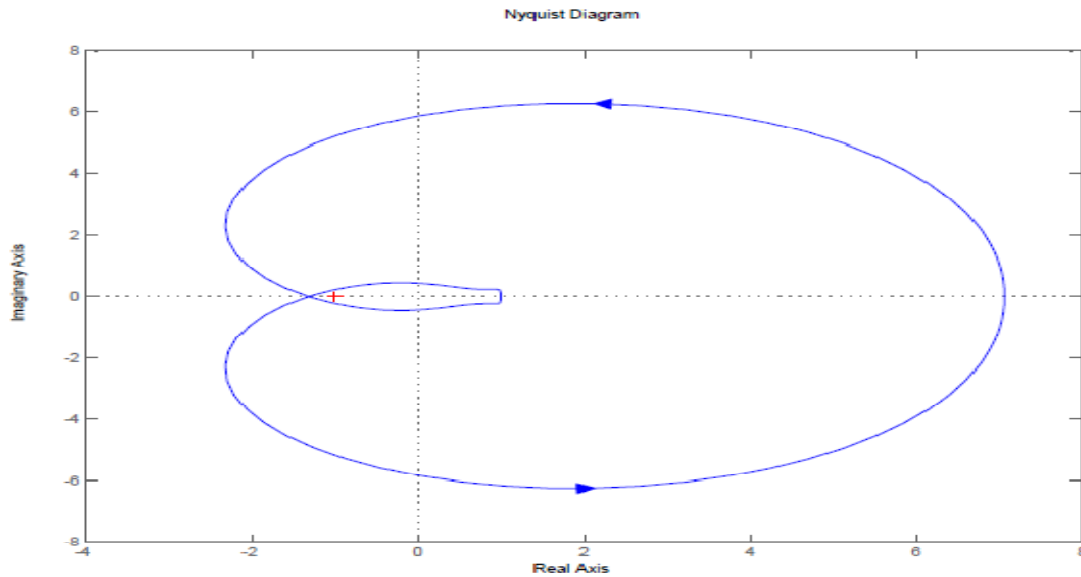


Figure 4.9: Lieu de Nyquist de la fonction de transfert $1 + L_8(s)$

Le tracé de Nyquist effectuée, dans ce cas ci aussi, deux tours autour de l'origine du plan complexe.

$$P_{in} = T = 2$$

L'asservissement par la commande LQG, pour le second choix des pondérations, est stable.

4.5 Conclusion :

Les retours d'état qui ont été appliqués au modèle sont tous stables, sauf qu'aucun d'eux ne parvient à remédier au fait que le système soit à non minimum de phase (il y'a toujours des zéros dans la partie droite du plan complexe que se soit pour la fonction de transfert en boucle fermée de la position ou de l'angle θ)

Le fait que le système soit à non minimum de phase, ne nous a pas permis d'étudier les marges de stabilité (marge de gain et marge de phase).

CHAPITRE 5 : Présentation de l'interface du modèle sous MATLAB

5.1 Introduction

Les IHM (Interfaces Homme Machine), sont appelées GUI (Graphical User Interfaces) dans MATLAB. Elles permettent à l'utilisateur, grâce à des objets graphiques (boutons, menus, cases à cocher, ...) d'interagir avec un programme informatique.

Depuis la version 5.0 (1997), MATLAB possède un outil IDE dédié à la création des interfaces graphiques. Cet outil, appelé GUIDE (Graphical User Interface Development Environment), permet de concevoir intuitivement ces interfaces graphiques.[Bri]

5.2 Objets graphiques Matlab

5.2.1 Hiérarchie des objets graphiques

Sous Matlab, les objets graphiques sont disposés selon une hiérarchie pyramidale parent-enfant [Bri]:

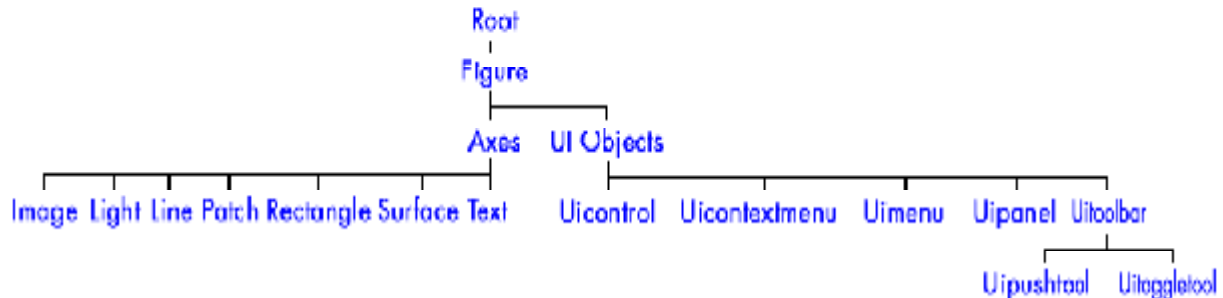


Figure 5.1: Hiérarchie des objets graphiques sous Matlab

Au sommet de la hiérarchie se trouve l'objet **Root**. Cet objet est invisible (on peut se le représenter comme étant l'écran de l'ordinateur). L'utilisateur n'interagit que très rarement avec cet objet.

Ensuite, on trouve les objets de type **Figure**. Ce sont les conteneurs visibles où sont disposés tous les autres objets enfants. Plusieurs objets **Figure** peuvent être ouverts simultanément et peuvent éventuellement communiquer entre eux.

Les objets de type **Axes** sont les zones de traçage des graphiques (2D ou 3D). Ces objets ont pour enfants, tous les objets représentant des résultats mathématiques (courbes, surfaces, images, maillages, etc.).

On trouve également au même niveau, les objets **UI (User Interface)** tels que des boutons, des menus, des cases à cocher, ... Ces objets permettent à l'utilisateur d'interagir dynamiquement à la souris avec le GUI.

5.2.2 Propriétés des objets graphiques

Chaque objet graphique possède des propriétés (position, couleur, action, etc.) qui sont définies à sa création et qui peuvent être modifiées dynamiquement au cours du fonctionnement du GUI. Ces propriétés peuvent être récupérées et modifiées en utilisant l'identifiant de l'objet et les fonctions GET et SET.

5.3 Programmation des interfaces

Les interfaces graphiques peuvent être programmées de deux manières différentes :
La première se fait en utilisant l'outil GUIDE de Matlab illustré dans la figure 5.2. [Lien3]

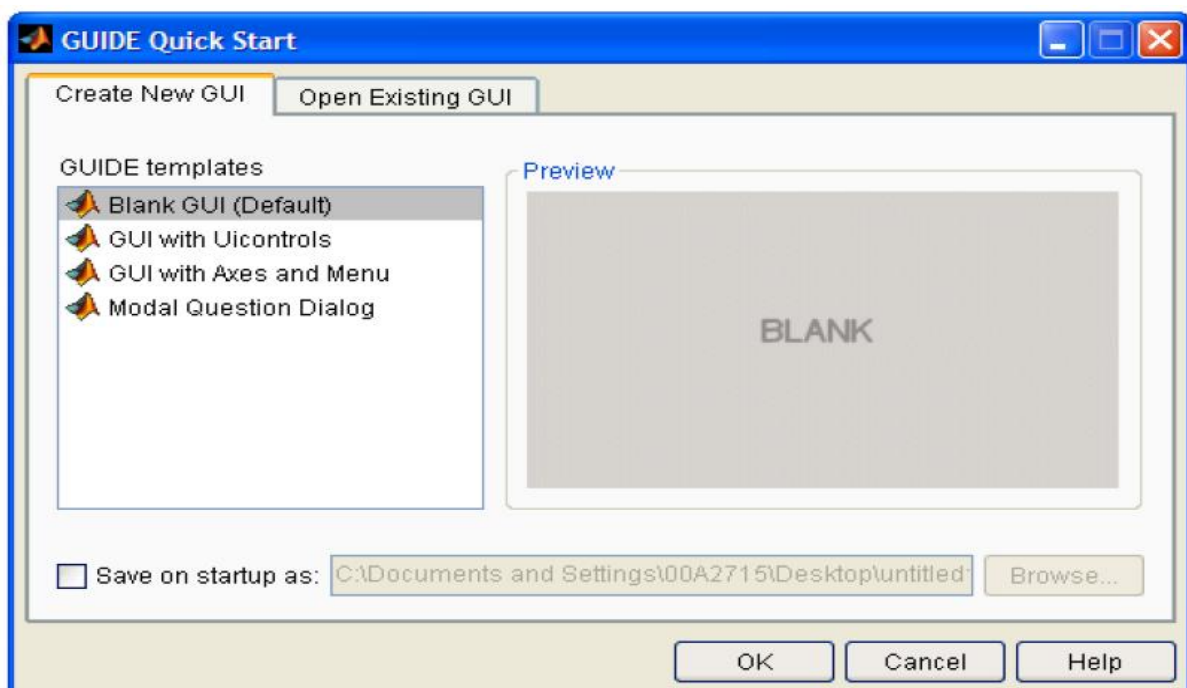


Figure 5.2: Fenêtre de l'outil GUIDE

Celle-ci apparaît en tapant GUIDE dans la fenêtre de commande de MATLAB

Comme le démontre la figure 5.3 cet outil permet la programmation des interfaces avec des objets prédéfinis [Lien3]

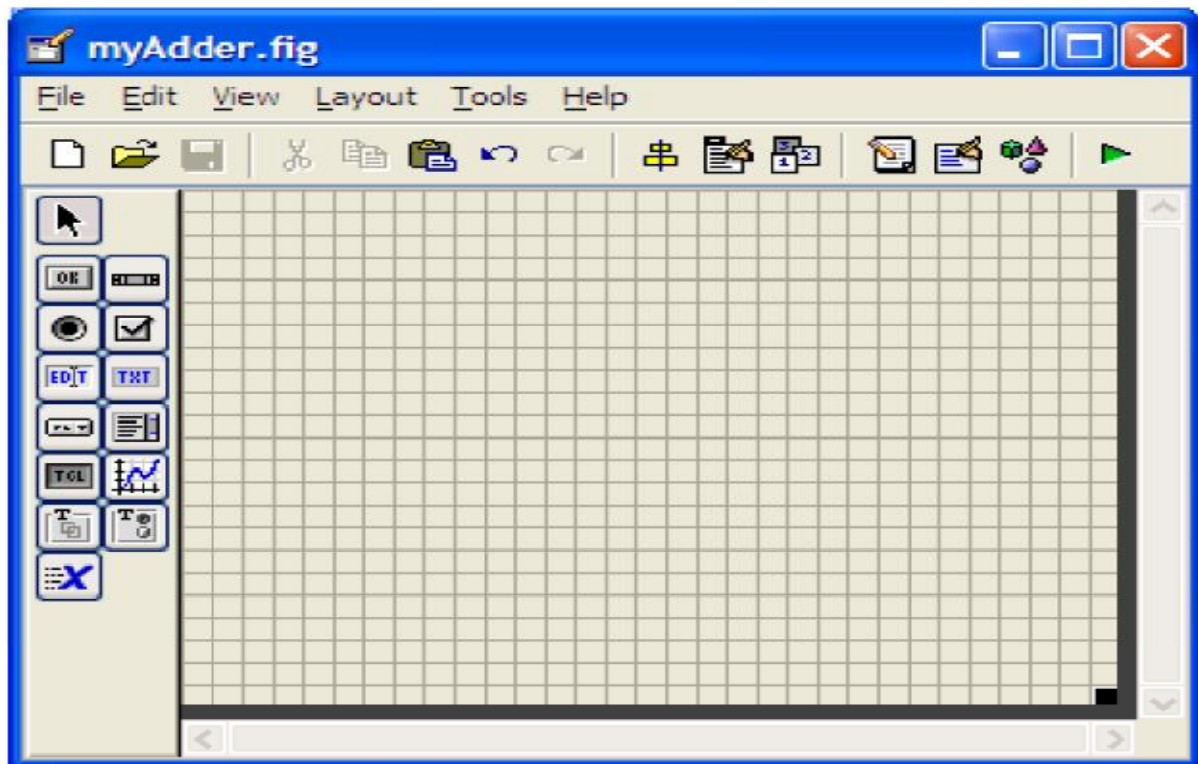


Figure 5.3 : l'espace de programmation GUI

Après avoir mis en place tous les objets et ajusté toutes les propriétés, le GUIDE génère deux fichiers. Un fichier *.fig* (non éditable) contenant les objets graphiques (Figure, Axes et Boutons poussoirs...) et un fichier *.m* contenant les lignes de code du programme.

Bien que la génération automatique du code permette d'éviter les erreurs de syntaxe (généralement périlleuses à corriger pour le débutant), l'inconvénient de cette méthode de programmation est la difficulté de lire ou exploiter les codes générés dans le fichier *.m* même pour un programmeur averti. Son évolution et sa maintenance sont donc également très difficiles (qui plus est sans l'utilisation du GUIDE). Mais la principale limitation vient du fait que la programmation des objets graphiques n'apparaît nulle part (fichier *.fig* crypté). [Bri]

La seconde manière permettant de réaliser une interface est la méthode par programmation dans un fichier *.m* de Matlab.

Les commandes utilisées à cet effet sont comme suit [Dav]:

- La commande *figure* permet la création d'une fenêtre.
- Les commandes *set* et *get* permettent l'accès à toutes les propriétés des graphes insérés dans l'interface.
- La commande *uicontrol* permet d'incérer à notre figure les objets présentés dans le tableau 5.1.[Dav][Hel]

5.4 Interface du modèle ball and beam

Cette interface va nous permettre de rendre concret le déplacement de la balle sur la poutre suivant la réponse du modèle corrigé avec un retour d'état (commande par placement de pôles).

Les objectifs de cette interface sont :

- Visualiser la réponse de la position du modèle ;
- Animation du déplacement de la balle sur la barre ;
- Possibilité du choix des pôles, qui peuvent être réels ou complexes ;
- Variation de la consigne ;
- Effacement et répétition des graphes.

Le programme d'origine, permettant de réaliser l'animation du modèle ball and beam, est donné dans l'annexe. Il a été modifié pour mieux répondre à notre modèle d'état qui a été étudié dans ce projet.

Ainsi l'interface finale, illustrant notre modèle, va se présenter comme l'illustre la figure 5.4 :

Cette interface ne simule que les réponses apériodiques (sans dépassement) du modèle Ball and Beam c'est pour cela que des améliorations sont à apporter.

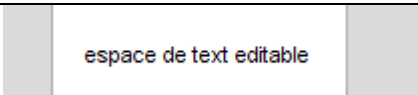
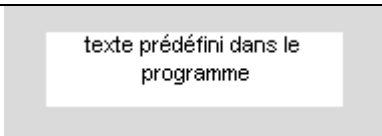
Objets de l'uicontrol	Objets correspondants dans le GUIDE
Check boxes	
Editable text fields	
Frames	
List boxes	
Pop-up menus	Menu instantané
Push buttons	
Radio buttons	
Sliders	
Static text labels	
Toggle buttons	 « bouton on/off »

Tableau 5.1: Objets de l'uicontrol

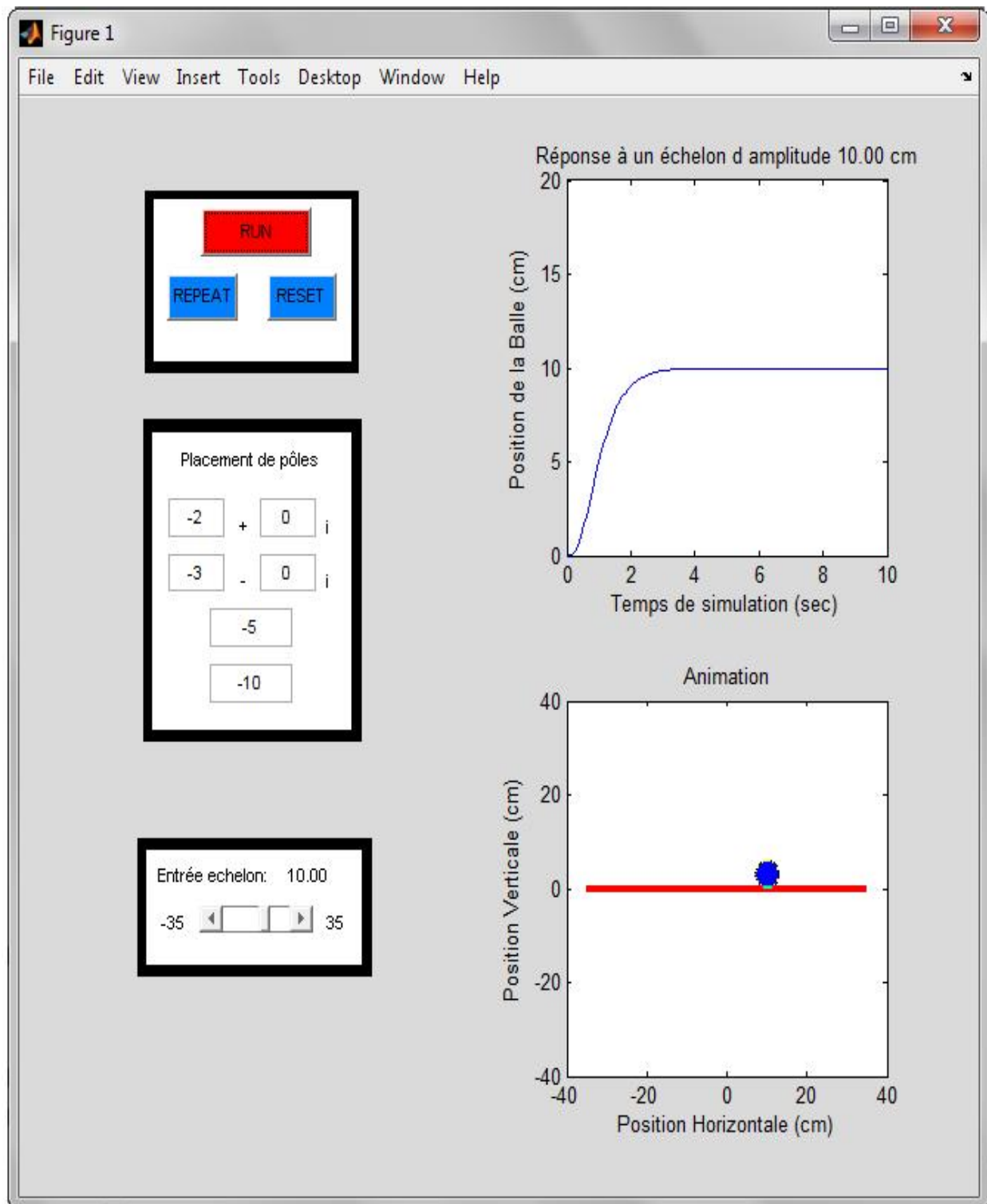


Figure 5.4: Intreface du modèle Ball and Beam.

5.5 Conclusion

L'animation est très utile et intéressante pour une utilisation au laboratoire dans le cas d'absence d'un banc de travaux pratiques expérimental pour une meilleure appréciation des réponses du modèle obtenu sous Simulink et du fonctionnement du système commandé par les différentes méthodes.

Conclusion générale

Conclusion générale :

Le système Ball and Beam a été modélisé et linéarisé autour de son origine, la commandabilité et observabilité ont été améliorées après la réduction de l'ordre du modèle. Nous avons aussi négligé le frottement entre la balle et la barre du fait de sa très faible valeur afin de simplifier le modèle.

Des simulations du modèle ont été conduites pour étudier plusieurs structures de commande, afin de comparer les résultats. De ce fait on a bien remarqué que la commande LQR donne de meilleurs résultats par rapport aux autres commandes. La commande LQG fut la plus sensible aux bruits.

La commande LQG/LTR n'a pas pu être étudiée pour ce modèle du fait que ce dernier ne satisfait pas les hypothèses d'application de la commande. L'introduction des observateurs influence négativement les performances. Cela est dû au fait que les bruits agissent à la fois sur l'observateur en détériorant son pronostique et en même temps sur le modèle d'état.

Les lieux de Nyquist des fonctions de transfert en boucle ouverte du modèle corrigé ont été tracés pour chaque application de commande afin d'étudier la stabilité de ces commandes.

Dans le cas de système à déphasage non minimal (c'est-à-dire avec des zéros "instables"), comme l'est notre système, on ne peut pas "pousser" indéfiniment les performances même si l'on s'autorise des commandes sans aucune limitation ; ceci est tout à fait logique puisque les zéros (du système comme du correcteur) attirent les pôles en boucle fermée quand le gain de boucle augmente. [Alaz2]

Recommandation pour les travaux ultérieurs :

D'autres méthodes de commande pourraient s'appliquer au système comme la commande non linéaire, commande robuste ou encore commande par la logique floue.

La commande non linéaire donnera de meilleurs résultats du fait de la non linéarité du système.

Des commandes mieux adaptées aux systèmes à non minimum de phase sont intéressantes à étudier aussi.

L'interface développée pour ce modèle ne simule que les réponses sans dépassement, donc une amélioration de celle-ci ou même une application d'un autre type de commande et une comparaison sont à effectuer.

Bibliographie

Bibliographie :

[Alaz1] D.ALAZARD, Ch. CUMER, P.APKARIAN, M.GAUVRIR, G.FERRERES.
«*Robustesse et commande optimale* » Edition Cépadués.1999

[Alaz2] D.ALAZARD. « *régulation LQ/LQG* » notes de cours. Edition SUPAERO

[Ata] Armin Ataei-Esfahani “*Ball-on-Beam Balancer*”
www.personal.psu.edu/users/a/x/axa943/ball_on_beam_balancer.htm

[Berg] Benoît Bergeon. « *Commande linéaire des systèmes multivariables* ». 1Master
ISAS. Université Bordeaux 1.

[Bhat] Shankar P. Bhattacharyya, aniruddha Datta, L. H. Keel: “*Linear control theory
Structure, Robustness, and Optimization*” CRC Press 2009

[Blan] Gérard Blanchet, “*Optimisation quadratique en automatique*” Version provisoire.
2007.

[Bri] Jérôme Briot, “*Introduction à la programmation des interfaces graphiques*”
<http://briot-jerome.developpez.com/> Dernière mise à jour : 17/03/2009.

[Dav] Timothy A. Davis, Kermit Sigmon “*Matlab Primer Seventh Edition*” Chapman and
Hall/CRC. 2005

[Gasp] Oleg N. Gasparyan. “*Linear and Nonlinear Multivariable Feedback Control: A
Classical Approach*”. John Wiley & Sons, Ltd. 2008

[Gou] « retour d'état et observateur »
<http://fredgouaisbaut.free.fr/IMG/pdf/observateur4.pdf>

[Hau] J. Hauser, Sh. Sastry, P. Kokotovic. “*Nonlinear Control Via Approximate Input-
Output Linearization: the ball and beam example*”. IEEE TRANSACTION ON
AUTOMATIC CONTROL, Vol. 37, NO.3, MARCH 1992.

[Help] Help de Matlab version R2007b

[Hesp] Joao P. Hespanha « *Linear systems theory* » copyright by Princeton University Press Published by Princeton University Press 2009.

[Hirs] Robert Hirsch EDUMECH “*Mechatronic Instructional Systems Ball on Beam System*”. Shandor Motion Systems. (1999)

[Lar] Edouard Laroche. “Commande Optimale”, ENSPS - 3A ISAV 2009/2010.
<http://eavr.u-strasbg.fr/perso/edouard/Student/>

[Lieb] Jeff Lieberman “*A Robotic Ball Balancing Beam*” February 10, 2004

[Lien1] <http://www-hadoc.lag.ensieg.inpg.fr/hadoc/continuu/n24/d24.htm> ; 06/2010.

[Lien2] <http://sciences-physiques.ac-dijon.fr/archives/documents/PhysiqueAppliquee/Tournus/MoteurCC/CoursMoteurCC.htm>

[Lien3] <http://blinkdagger.com/matlab/matlab-gui-graphical-user-interface-tutorial-for-beginn> . Dernière consultation 05/05/2010

[Lien4] <http://www.tn.refer.org/hebergement/robuste/stab.html>

[Lien5] <http://www.engin.umich.edu/group/ctm/gui/ball>

[Man] Simone Mannori , Ramine Nikoukhah , Serge Steer « *Commande des procédés industriels avec des logiciels libres* » www.scicos.org/ScicosHIL/angers2006.pdf

[Oste] Eric OSTERTAG. “*Commande et estimation multivariables Méthodes linéaires et optimisation quadratique*”, édition ellipses, 2006.

[Proc] Hynek Procházka, Ioan Doré Landau « *Synthèse de régulateurs robustes et performants par placement de pôles et calibrage des sensibilités* » LAG
Laboratoire d’automatique de Grenoble 2004/2005.

[Sko] Sigurd Skogestad, Ian Postlethwaite “*Multivariable feedback control Analysis and Design*” John Wiley and Sons

[Vac] Corinne VACHIER. « *REPRÉSENTATION D’ÉTAT ET COMMANDE DANS L’ESPACE D’ÉTAT* » UNIVERSITÉ PARIS XII-VAL DE MARNE UFR Sciences et

Technologie Master 1 Mention Sciences de l'Ingénieur. *Dernière mise à jour septembre 2007*

[Vez] Simon Vézina « *Le moment d'inertie et l'énergie cinétique de rotation* »
http://profs.cmaisonneuve.qc.ca/svezina/nya/note_nya/NYA_XXI_Chap%204.4.pdf

[Von] J. Vondřich, E. Thóndel “*MODELLING OF LQR CONTROL WITH MATLAB*”.
Department of Mechanics and Materials Science, Faculty of Electrical Engineering, CTU
in Prague

[Wan] Wei Wang, “*Control of a Ball and Beam System*”, School of Mechanical
Engineering, The University of Adelaide South Australia 5005. AUSTRALIA.

[Well] Peter Wellstead “Ball and Beam 1: Basics” www.control-systems-principles.co.uk

[Well2] Peter E. Wellstead “*Introduction to physical system modeling*”. Electronically
published by: www.control-systems-principles.co.uk

[Wiki] http://fr.wikipedia.org/wiki/Observateur_d%C3%A9tat

[Wiki2]
http://fr.wikipedia.org/wiki/A%C3%A9ronef_%C3%A0_d%C3%A9collage_et_atterrissage_verticaux

Annexe

Cette annexe contient les programmes (fonction) d'origine, téléchargeable sur internet.

A.1.

Ce programme permet la simulation du graphe de la position et la réalisation de l'animation correspondante :[Lien 5]

%*****%

% *Début du programme* %

%*****%

```
function bbFUN(action)
%The Callback for the Run Button%
if action == 1
    g=-9.8;
    A=[0 1 0 0;0 0 -5/7*g 0;0 0 0 1;0 0 0 0];
    B=[0;0;0;1];
    C=[1 0 1 0];
    D=[0];

%Get the poles from the editable text in the GUI%
    realhandle = findobj('Tag','realtext');
    real = eval(get(realhandle,'String'));
    imaghandle = findobj('Tag','imagtext');
    imag = eval(get(imaghandle,'String'));
    p1=real+imag*i;
    p2=real-imag*i;
    p3handle = findobj('Tag','pole3');
    p3=eval(get(p3handle,'String'));
    p4handle = findobj('Tag','pole4');
    p4=eval(get(p4handle,'String'));
```

```

%Place the poles and create the K-matrix%
K=place(A,B,[p1,p2,p3,p4]);

%Store the K-matrix in the GUI.
Khandle = findobj('Tag','Kframe');
set(Khandle,'Value',K);

%Get the value of the step input from the step slider%
stephandle = findobj('Tag','stepslider');
stepval=get(stephandle,'Value');
stepaxis=stepval;

%The following Lines represent the rscale function to find Nbar%
%The reference input checkbox is checked for its value, also%
Nbarhandle = findobj('Tag','Checkbox3');
Nbarval = get(Nbarhandle,'Value');
if Nbarval == 0
    Nbar = 1
    Nbarfhandle = findobj('Tag','Nbarframe');
    set(Nbarfhandle,'Value',Nbar);
    stepaxis=stepval/1000;
elseif Nbarval == 1
    s = size(A,1);
    Z = [zeros([1,s] 1);
    N = inv([A,B;C,D])*Z';
    Nx = N(1:s);
    Nu = N(1+s);
    Nbar=Nu + K*Nx;
    Nbarfhandle = findobj('Tag','Nbarframe');
    set(Nbarfhandle,'Value',Nbar);
end

%Check if the system is linear or non-linear from checkbox%
syshandle = findobj('Tag','syscheckbox');

```

```
sysval = get(syshandle,'Value');
```

```
%Linear System Simulation%
```

```
if sysval == 0
```

```
    T = 0:0.05:6;
```

```
    U = stepval*ones(size(T));
```

```
    [Y,X]=lsim(A-B*K,B*Nbar,C,D,U,T);
```

```
    ball=X(:,1);
```

```
    theta=X(:,3);
```

```
%Non-Linear System Simulation%
```

```
else
```

```
    t0=0;
```

```
    tfinal=6;
```

```
    x0=[0 0 0 0];
```

```
        v=version;
```

```
        if eval(v(1))>=5
```

```
            'Please wait while simulation is running'
```

```
            [T,X]=ode23('bbFUNODE',[t0 tfinal],x0);
```

```
        else
```

```
            tol=1e-5;
```

```
            'Please wait while simulation is running'
```

```
            [T,X]=ode45('bbFUNODE',t0,tfinal,x0,tol);
```

```
        end
```

```
    ball=X(:,1);
```

```
    theta=X(:,3);
```

```
end
```

```
%Store the values of ballx anf theta%
```

```
ballhandle = findobj('Tag','ballxframe');
```

```
set(ballhandle,'Value',ball);
```

```
thetahandle = findobj('Tag','thetaframe');
```

```
set(thetahandle,'Value',theta);
```

```
timehandle = findobj('Tag','timeframe');
```

```

set(timehandle,'Value',T);

%Set ball and beam characteristics for animation%
ball=ball*100;
beam_length = 100;
bl2 = beam_length/2;
radius = 3;          %Ball radius%
arcstep = 36;
ballx=ball.*cos(theta);
bally = - ballx .* tan(theta) + radius;
j = 0:arcstep:(360-arcstep);
arcx = radius * cos((j+arcstep) * pi/180);
arcy = radius * sin((j+arcstep) * pi/180);
beamx1 = -bl2 * cos(theta);
beamx2 = bl2 * cos(theta);
beamy1 = bl2 * sin(theta);
beamy2 = -bl2 * sin(theta);

%Check if the step response is to be plotted seperately%
handle2 = findobj('Tag','Checkbox2');
value2=get(handle2,'Value');
if value2 == 1
    subplot(2,2,2)
    plot(T,ball)
else
    subplot(2,2,2)
    plot(T(1),ball(1), 'EraseMode', 'none')
end

%Assign values for the axis based on the value of stepval%
if stepval > 0
    axis([0 6 0 stepaxis*200])
elseif stepval < 0
    axis([0 6 stepaxis*200 0])

```

```

else
    axis([0 6 -50 50])
end

%Add title, xlabel, and ylabel to the step response plot%
title(sprintf('Step Response to %2.2f cm input',stepval*100))
ylabel('Ball Position (cm)')
xlabel('Time (sec)')

hold on

%Plot the first element of the ball and beam with title,etc...%
subplot(2,2,4)
L = plot([beamx1(1) beamx2(1)], [beamy1(1) beamy2(1)], 'r', 'EraseMode',...
'xor','LineWidth',[3]);
axis([-55 55 -55 55]);
title('Animation')
xlabel('Horizontal Position (cm)')
ylabel('Vertical Position (cm)')
H = patch(arcx+ballx(1), arcy+bally(1), 'b', 'EraseMode', 'xor');

%Check if the graphs should advance manuallly%
handle = findobj('Tag','Checkbox');
value=get(handle,'Value');
if value == 1
    'Press any key to advance the animation'
    pause

end

%For-loop runs the animation%
ltheta=length(theta);
for k = 2:ltheta-1,

```

```

%Manual advance%
    if value == 1
        pause
    end

%Set ball and beam data and draw the new coordinates%
    set(L, 'XData', [beamx1(k) beamx2(k)]);
    set(L, 'YData', [beamy1(k) beamy2(k)]);
    set(H, 'XData', arcx+ballx(k));
    set(H, 'YData', arcy+bally(k));
    drawnow;

%Check if ball is still on the beam%
    if ball(k) > 50
        subplot(2,2,4)
        text(-25,25,'Ball has fallen off beam!')
        return
    end

%Plot response with animation%
    if value2 == 0
        subplot(2,2,2)
        plot([T(k),T(k+1)],[ball(k),ball(k+1)], 'EraseMode', 'none')
    end

end

%The Callback for the Reset Button%
elseif action == 2

%Clears the step response axis%
    subplot(2,2,2)

```

```

cla
axis([0 5 0 50])
ylabel('Ball Position (cm)')
xlabel('Time (sec)')
title('Step Response')

%Returns ball and beam to their zero conditions%
beam_length = 100;
bl2 = beam_length/2;
radius = 3.0;
arcstep = 36;
ballx = 0;
bally = - ballx .* tan(0) + radius;
j = 0:arcstep:(360-arcstep);
arcx = radius * cos((j+arcstep) * pi/180);
arcy = radius * sin((j+arcstep) * pi/180);
beamx1 = -bl2 * cos(0);
beamx2 = bl2 * cos(0);
beamy1 = bl2 * sin(0);
beamy2 = -bl2 * sin(0);
subplot(2,2,4)
L = plot([beamx1(1) beamx2(1)], [beamy1(1) beamy2(1)], 'r', 'EraseMode', ...
    'xor','LineWidth',[3]);
axis([-55 55 -55 55]);
title('Animation')
xlabel('Horizontal Position (cm)')
ylabel('Vertical Position (cm)')
H = patch(arcx+ballx(1), arcy+bally(1), 'b', 'EraseMode', 'xor');

%Callback for viewing the current value of the step-slider%
elseif action == 3
    stephandle = findobj('Tag','stepslider');
    stepval=get(stephandle,'Value')*100;
    curhandle = findobj('Tag','curtext');

```

```

set(curhandle,'String',sprintf('%6.2f',stepval));

%Callback for the Repeat button%
elseif action == 4
%Get the stored values of ballx, theta, and time%
    ballhandle = findobj('Tag','ballxframe');
    ball=get(ballhandle,'Value');
    thetahandle = findobj('Tag','thetaframe');
    theta=get(thetahandle,'Value');
    timehandle = findobj('Tag','timeframe');
    T=get(timehandle,'Value');

%Set ball and beam characteristics for animation%
    ball=ball*100;
    beam_length = 100;
    bl2 = beam_length/2;
    radius = 3;          %Ball radius%
    arcstep = 36;
    ballx=ball.*cos(theta);
    bally = - ballx .* tan(theta) + radius;
    j = 0:arcstep:(360-arcstep);
    arcx = radius * cos((j+arcstep) * pi/180);
    arcy = radius * sin((j+arcstep) * pi/180);
    beamx1 = -bl2 * cos(theta);
    beamx2 = bl2 * cos(theta);
    beamy1 = bl2 * sin(theta);
    beamy2 = -bl2 * sin(theta);

%Check if the step response is to be plotted seperately%
    handle2 = findobj('Tag','Checkbox2');
    value2=get(handle2,'Value');
    if value2 == 1
        subplot(2,2,2)

```

```

    plot(T,ball)
else
    subplot(2,2,2)
    plot(T(1),ball(1), 'EraseMode', 'none')
end

%Assign values for the axis based on the value of stepval%
stephandle = findobj('Tag','stepslider');
stepval=get(stephandle,'Value');

Nbarhandle = findobj('Tag','Checkbox3');
Nbarval = get(Nbarhandle,'Value');
if Nbarval == 1
    stepaxis=stepval;
else
    stepaxis=stepval/1000;
end

if stepval > 0
    axis([0 6 0 stepaxis*200])
elseif stepval < 0
    axis([0 6 stepaxis*200 0])
else
    axis([0 6 -50 50])
end

%Add title, xlabel, and ylabel to the step response plot%
title(sprintf('Step Response to %2.2f cm input',stepval*100))
ylabel('Ball Position (cm)')
xlabel('Time (sec)')

hold on

%Plot the first element of the ball and beam with title,etc...%

```

```

subplot(2,2,4)
L = plot([beamx1(1) beamx2(1)], [beamy1(1) beamy2(1)], 'r', 'EraseMode',...
'xor','LineWidth',[3]);
axis([-55 55 -55 55]);
title('Animation')
xlabel('Horizontal Position (cm)')
ylabel('Vertical Position (cm)')
H = patch(arcx+ballx(1), arcy+bally(1), 'b', 'EraseMode', 'xor');

%Check if the graphs should advance manuallly%
handle = findobj('Tag','Checkbox');
value=get(handle,'Value');

%For-loop runs the animation%
ltheta=length(theta);
for k = 2:ltheta-1,

%Manual advance%
    if value == 1
        pause
    end

%Set ball and beam data and draw the new coordinates%
    set(L, 'XData', [beamx1(k) beamx2(k)]);
    set(L, 'YData', [beamy1(k) beamy2(k)]);
    set(H, 'XData', arcx+ballx(k));
    set(H, 'YData', arcy+bally(k));
    drawnow;

%Check if ball is still on the beam%
    if ball(k) > 50
        subplot(2,2,4)
        text(-25,25,'Ball has fallen off beam!')
        return
    end
end

```

```

end

%Plot response with animation%
if value2 == 0
    subplot(2,2,2)
    plot([T(k),T(k+1)],[ball(k),ball(k+1)], 'EraseMode', 'none')
end
end
end
%*****%

%                               Fin du programme                               %

%*****%

```

A.2.

Cette fonction permet la construction de l'interface permettant la simulation et l'animation de la réponse du système.[Lien 5]

```

%*****%

%                               Début du programme                               %

%*****%

function bbGUI()
a = figure('Color',[1 1 1], ...
    'Position',[440 124 700 700], ...
    'Tag','Fig1');
whitebg
whitebg
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[0 0 0], ...
    'Position',[0.128571 0.514286 0.228571 0.192857], ...
    'Style','frame', ...

```

```

        'Tag','Nbarframe');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.137143 0.522857 0.211429 0.177143], ...
    'Style','frame', ...
    'Tag','Kframe');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.181429 0.672857 0.115714 0.02], ...
    'String','Place Poles', ...
    'Style','text', ...
    'Tag','placetext');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.155714 0.627143 0.0585714 0.035], ...
    'String','-2', ...
    'Style','edit', ...
    'Tag','realtext', ...
    'Value',1);

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.25 0.627143 0.0585714 0.035], ...
    'String','5', ...
    'Style','edit', ...
    'Tag','imagtext', ...
    'Value',1);

```

```
b = uicontrol('Parent',a, ...  
    'Units','normalized', ...  
    'BackgroundColor',[1 1 1], ...  
    'Position',[0.225 0.64 0.017 0.020], ...  
    'String','+', ...  
    'Style','text', ...  
    'Tag','plustext');
```

```
b = uicontrol('Parent',a, ...  
    'Units','normalized', ...  
    'BackgroundColor',[1 1 1], ...  
    'Position',[0.225 0.624 0.017 0.017], ...  
    'String','- ', ...  
    'Style','text', ...  
    'Tag','minustext');
```

```
b = uicontrol('Parent',a, ...  
    'Units','normalized', ...  
    'BackgroundColor',[1 1 1], ...  
    'Position',[0.311429 0.631429 0.0185714 0.0185714], ...  
    'String','i', ...  
    'Style','text', ...  
    'Tag','itext');
```

```
b = uicontrol('Parent',a, ...  
    'Units','normalized', ...  
    'BackgroundColor',[1 1 1], ...  
    'Position',[0.198571 0.58 0.0857143 0.035], ...  
    'String','-20', ...  
    'Style','edit', ...  
    'Tag','pole3');
```

```
b = uicontrol('Parent',a, ...
```

```

        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Position',[0.198571 0.538571 0.0857143 0.035], ...
        'String','-80', ...
        'Style','edit', ...
        'Tag','pole4');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[0 0 0], ...
    'Position',[0.122857 0.365714 0.244286 0.125714], ...
    'Style','frame', ...
    'Tag','Frame1');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.13 0.374286 0.228571 0.108571], ...
    'Style','frame', ...
    'Tag','timeframe');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','bbFUN(3)', ...
    'Max',0.5, ...
    'Min',-0.5, ...
    'Position',[0.187143 0.395714 0.118571 0.0242857], ...
    'Style','slider', ...
    'Tag','stepslider', ...
    'Value',0.25);

b = uicontrol('Parent',a, ...
    'Units','normalized', ...

```

```

'BackgroundColor',[1 1 1], ...
'Position',[0.137143 0.398571 0.0442857 0.0185714], ...
'String','-50', ...
'Style','text', ...
'Tag','slid_min');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.307143 0.398571 0.0357143 0.0185714], ...
    'String',' 50', ...
    'Style','text', ...
    'Tag','slid_max');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.151429 0.444286 0.0928571 0.0214286], ...
    'String','Step Input:', ...
    'Style','text', ...
    'Tag','step_text');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.238571 0.444286 0.0842857 0.0214286], ...
    'String','25.00', ...
    'Style','text', ...
    'Tag','curtext');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[0 0 0], ...
    'Position',[0.13 0.737143 0.224286 0.165714], ...

```

```

        'Style','frame', ...
        'Tag','ballxframe');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.138571 0.744286 0.208571 0.15], ...
    'Style','frame', ...
    'Tag','thetaframe');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[1 0 0], ...
    'Callback','bbFUN(1)', ...
    'Position',[0.188571 0.825714 0.115714 0.0442857], ...
    'String','RUN', ...
    'Tag','Runbutton', ...
    'Value',1);

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[0 1 0], ...
    'Callback','bbFUN(2)', ...
    'Position',[0.258571 0.767143 0.0728571 0.0428571], ...
    'String','RESET', ...
    'Tag','Restbutton');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[0 1 0], ...
    'Callback','bbFUN(4)', ...
    'Position',[0.154286 0.767143 0.0728571 0.0428571], ...
    'String','REPEAT', ...
    'Tag','repbutton');

```

```

b = axes('Parent',a, ...
        'Box','on', ...
        'Color',[1 1 1], ...
        'NextPlot','add', ...
        'Position',[0.577977 0.581098 0.327023 0.343902], ...
        'Tag','anim_axes', ...
        'XColor',[0 0 0], ...
        'XLim',[0 6], ...
        'XLimMode','manual', ...
        'YColor',[0 0 0], ...
        'YLim',[0 50], ...
        'YLimMode','manual');

```

```

c = text('Parent',b, ...
        'Color',[0 0 0], ...
        'HorizontalAlignment','center', ...
        'Position',[2.98684 51.3542 0], ...
        'String','Step Response', ...
        'Tag','Text1', ...
        'VerticalAlignment','bottom');
set(get(c,'Parent'),'Title',c);

```

```

c = text('Parent',b, ...
        'Color',[0 0 0], ...
        'HorizontalAlignment','center', ...
        'Position',[-0.539474 25.1042 0], ...
        'Rotation',90, ...
        'String','Ball Position (cm)', ...
        'Tag','Text2', ...
        'VerticalAlignment','baseline');
set(get(c,'Parent'),'YLabel',c);

```

```

c = text('Parent',b, ...

```

```

'Color',[0 0 0], ...
'HorizontalAlignment','center', ...
'Position',[2.98684 -3.64583 0], ...
'String','Time (sec)', ...
'Tag','Text3', ...
'VerticalAlignment','cap');
set(get(c,'Parent'),'XLabel',c);

b = axes('Parent',a, ...
'Box','on', ...
'Color',[1 1 1], ...
'Position',[0.577977 0.11 0.327023 0.343902], ...
'Tag','step_axes', ...
'XColor',[0 0 0], ...
'XLim',[-55 55], ...
'XLimMode','manual', ...
'YColor',[0 0 0], ...
'YLim',[-55 55], ...
'YLimMode','manual');

c = line('Parent',b, ...
'Color',[1 0 0], ...
'EraseMode','xor', ...
'LineWidth',3, ...
'Tag','Line2', ...
'XData',[-50 50], ...
'YData',[0 -0]);

c = text('Parent',b, ...
'Color',[0 0 0], ...
'HorizontalAlignment','center', ...
'Position',[-0.241228 57.9792 0], ...
'String','Animation', ...
'Tag','Text4', ...

```

```
        'VerticalAlignment','bottom');  
set(get(c,'Parent'),'Title',c);
```

```
c = text('Parent',b, ...  
        'Color',[0 0 0], ...  
        'HorizontalAlignment','center', ...  
        'Position',[-0.241228 -63.0208 0], ...  
        'String','Horizontal Position (cm)', ...  
        'Tag','Text5', ...  
        'VerticalAlignment','cap');  
set(get(c,'Parent'),'XLabel',c);
```

```
c = text('Parent',b, ...  
        'Color',[0 0 0], ...  
        'HorizontalAlignment','center', ...  
        'Position',[-68.2675 0.229167 0], ...  
        'Rotation',90, ...  
        'String','Vertical Position (cm)', ...  
        'Tag','Text6', ...  
        'VerticalAlignment','baseline');  
set(get(c,'Parent'),'YLabel',c);
```

```
b = uicontrol('Parent',a, ...  
             'Units','normalized', ...  
             'BackgroundColor',[0 0 0], ...  
             'Position',[0.118571 0.105714 0.252857 0.232857], ...  
             'Style','frame', ...  
             'Tag','stepframe');
```

```
b = uicontrol('Parent',a, ...  
             'Units','normalized', ...  
             'BackgroundColor',[1 1 1], ...  
             'Position',[0.127143 0.28 0.235714 0.0485714], ...  
             'String','Manual Advance', ...
```

```

        'Style','checkbox', ...
        'Tag','Checkbox');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.127143 0.224286 0.235714 0.0485714], ...
    'String','Plot Separately', ...
    'Style','checkbox', ...
    'Tag','Checkbox2');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.127143 0.17 0.235714 0.0485714], ...
    'String','Reference Input', ...
    'Style','checkbox', ...
    'Tag','Checkbox3', ...
    'Value',1);

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Position',[0.127143 0.114286 0.235714 0.0485714], ...
    'String','Non-Linear System', ...
    'Style','checkbox', ...
    'Tag','syscheckbox');

%*****%

%                               Fin du programme                               %

%*****%

```

A.3.

Cette fonction permet la simulation de la réponse du modèle non linéaire du système ball and beam.[Lien5]

```
%*****%

%                               Début du programme                               %

%*****%

function dx = bbFUNODE(t,x)

%Retrieve the value of the K matrix.
Khandle = findobj('Tag','Kframe');
K = get(Khandle,'Value');

%Retrieve the value of Nbar.
Nbarhandle = findobj('Tag','Nbarframe');
Nbar = get(Nbarhandle,'Value');

%Retrieve the step input value.
stephandle = findobj('Tag','stepslider');
stepval=get(stephandle,'Value');

%Representation of the non-linear system.
dx=zeros(4,1);
dx(1)=x(2);
dx(2)=-5/7*(-9.8)*sin(x(3))+5/7*x(1)*x(4)^2;
dx(3)=x(4);
dx(4)=-K*[x(1) x(2) x(3) x(4)]'+Nbar*stepval;

%*****%

%                               fin                               %

%*****%
```