



Université de Ghardaïa

Faculté des Sciences et Technologies

Département d'automatique et électromécanique

Mémoire présenté en vue de l'obtention du diplôme de

MASTER

Domaine : *Sciences et Technologies*

Filière : Automatique

Spécialité : Automatique et systèmes

Par : LABED AHMED OUSSAMA

Thème

Commande non linéaire par Backstepping d'un system non linéaire
complexe application au bras manipulateur

Devant le jury :

BECHOUAT MOHCEN	M.C.A	Université Ghardaïa	Examineur
BEKKAR BELGACEM	M.C.B	Université Ghardaïa	Examineur
ACHOUR BEN CHAABANE	M.A.A	Université Ghardaïa	Encadreur

Année universitaire 2020/2021

Dédicace

Je dédie mon travail de recherche :

À ceux qui m'ont donné la vie, mes anges et mon bonheur, ...ma très chère
maman et mon très cher papa

À mes chers frère et sœurs.

À toute ma Famille.

À mes chères amies.

Remerciements

Je remercie Dieu qui m'a donné le courage et la force, de réaliser et de terminer ce modeste travail.

Mes plus sincères remerciements et ma plus profonde gratitude vont à :

Mes parents, ma famille pour leurs soutiens et encouragements. Mon encadreur Mr Achour Ben Chaabane pour sa disponibilité, ses efforts, sa patience, ses orientations, ses conseils durant tout le long du travail.

A l'ensemble des enseignants de notre département, et aux étudiants de notre promotion.

Je vous remercie tous.

المخلص

تمثل الانظمة الغير خطية حصة كبيرة من الانظمة المدروسة في المجال الصناعي من بينها نجد الروبوتات الالية او الاذرع الالية. من اجل هذا نجد ان البحث في هذا المجال ادى الى تطوير سبل التحكم وجعلها عاملا مهما في تحسين خصائص هذه الانظمة .

في هذا العمل الحالي الذي يتعلق بشكل اساسي بالتحكم بال (باكستيبينغ) لروبوت المناور بدرجتين من الحرية ، نحن مهتمون بالنموذج الديناميكي للروبوت المناور الذي يعبر عن عزم الدوران الذي تمارسه المشغلات كدالة لمواقع المفاصل من خلال التطبيق عليها التحكم بال (باكستيبينغ) القائمة على وظيفة (ليانينوف) التي توفر الرقابة الدائمة للسعي وراء بعض المسارات المرغوبة.

يتم إجراء اختبار المتانة من أجل الحصول على تقدير أفضل للنتائج التي تم الحصول عليها من خلال النظام الذي تمت دراسته من خلال مراعاة الاضطرابات في نظامنا ، والنتائج التي تم الحصول عليها تؤكد بوضوح الأداء الديناميكي وقوة تطبيق التحكم لمتابعة المسارات

كلمات مفتاحية : الانظمة الغير خطية , النموذج الديناميكي للروبوت , التحكم بال (باكستيبينغ)

ABSTRACT

Nonlinear systems represent a large part of the systems studied in industry among which we find manipulators robots. It is for this reason that the development of nonlinear control laws has become very interesting for improved performance.

In this present work which mainly concerns the Backstepping control of the manipulator robot with 2 degrees of freedom, we are interested in the dynamic model of the manipulator robot which expresses the torques exerted by the actuators as a function of the positions of the joints by applying to it Backstepping control laws based on a Lyapunov function for a pursuit of certain desired trajectories.

The Robustness test is carried out in order to have a better appreciation of the results obtained through the system studied by considering disturbances on our system, the results obtained clearly confirm the dynamic performances and the robustness of the control applied for the pursuit of the trajectories

Key words: Nonlinear systems, dynamic model of the manipulator robot , Backstepping control

Table de matière

INTRODUCTION GENERAL	12
CHAPITRE I	16
BRAS MANIPULATEURS	16
1 Généralité sur les bras manipulateurs	17
1.1 Introduction	17
1.2 Origine des termes	17
1.3 Historique :	18
1.4 Définition d'un robot	18
1.5 Classification	19
1.6 Anatomie d'un bras manipulateur :	21
1.7 Caractéristiques d'un robot	23
2 Modélisation des bras manipulateur	25
2.1 Introduction	25
2.2 Modèle géométrique	25
2.3 Modèle cinématique	26
2.4 Modèle dynamique :	27
3 Conclusion:	30
Chapitre II Modélisation dynamique d'un bras manipulateur a deux DDL	31
1. Définition et propriété des systèmes non linéaires:	32
1.1 Introduction	32
1.2 Définition :	32
1.3 Principe de superposition :	33
2. De la non-linéarité à la linéarité :	33
2.1 Points d'équilibre :	33
2.2 La linéarisation autour d'un point d'équilibre :	33
3. Modélisation dynamique du bras manipulateur :	35
3.1 Introduction :	35
3.2 modelés géométrique et cinématique directes	35
3.3 Formalisme de Lagrange :	36
3.4 La représentation d'état du robot:	37

3.5 Application numérique:	38
3.6 La linéarisation du model :	38
4. conclusion :	40
CHAPITRE III COMMANDE PAR RETOUR D'ETAT ET COMMANDE PAR BOUCLAGE	
LINEARISANT	41
1. Introduction :	42
2. La commande par retour d'état:	42
2.1 Calcul de la matrice des gains de retour :	42
2.2 Réglage du régime permanent:	44
3. Application de la commandes par retour état	48
3.1 Commande du système linéarisé par retour d'état :	48
3.2 Implantation de la commande dans le modèle non linéaire :	51
4. Commande par bouclage linéarisant entrée-sortie :	54
4.1 Systèmes mono-entrée mono-sortie	54
4.2 Systèmes multi-entrées multi-sorties	58
4.3 Application Commande par bouclage linéarisant entrée-sortie:	61
5. Conclusion	67
CHAPITRE IV COMMANDE PAR BACKSTEPPING	68
1. Introduction :	69
2 Théorie du Backstepping	69
2.1 Stabilité au sens de Lyapunov	69
2.2 Commande par Backstepping	72
3. Application du commande BACKSTEPPING :	77
4. Conclusion :	82
Conclusion générale	83
Table de référence	85

Liste de figures

Figure 1: exemple de telemanipulateurs	20
Figure 2: les constiutants d'un bras manipulateur	21
Figure 3: Robot planaire à 2 deux degre de liberte	35
Figure 4: Schéma de la commande par retour d'état avec précompensateur pour le système linéarisé	49
<i>Figure 5: Les 2 angles et leurs valeurs désirées du système linéarisé commandé</i>	<i>49</i>
<i>Figure 6: Les 2 angles et leurs valeurs désirées du système linéarisé commandé</i>	<i>50</i>
<i>Figure 7: Les couples générés par le retour d'état avec précompensateur</i>	<i>50</i>
Figure 8: Montage de commande du robot par retour d'état avec action intégrale	52
<i>Figure 9: Les 2 positions et leurs valeurs désirées du robot commandé par retour</i>	<i>52</i>
<i>Figure 10: Les 2 positions et leurs valeurs désirées du robot commandé par retour</i>	<i>53</i>
<i>Figure 11: Les couples générés par le retour d'état avec action intégrale</i>	<i>53</i>
Figure 12: Schéma de commande par linéarisation entrée-sortie.	57
Figure 13: Architecture de la commande par linéarisation et découplage entrées-sorties[12]	61
Figure 14: Montage de commande du robot par bouclage linéarisant entrée-sortie	64
Figure 15: Les 2 positions et leurs valeurs désirées du robot commandé par bouclage linéarisant sans présence de perturbation	65
Figure 16: Les 2 positions et leurs valeurs désirées du robot commandé par bouclage linéarisant en présence de perturbation	65

<i>Figure 17: Les couples générés pour les 2 actionneurs du robot commandé par bouclage linéarisant</i>	66
Figure 18: Stabilité au sens large et stabilité asymptotique[13]	71
Figure 19: Montage de commande du robot par Backstepping	79
Figure 20: Montage de contrôleur du robot par Backstepping	79
Figure 21: Les 2 positions et leurs valeurs désirées du robot commandé par Backstepping sans présence de perturbation	80
Figure 22: Les 2 positions et leurs valeurs désirées du robot commandé par Backstepping avec présence de perturbation	80
Figure 23: Les couples générés par Backstepping	81

INTRODUCTION

GENERAL

Quand on parle de la robotique, plusieurs idées viennent à l'esprit de chacun de nous. Historiquement, nous pourrions nous référer aux premiers concepts et automates de l'antiquité ou aux premiers robots comme à des personnages de la mythologie. Même le mot 'robot' a sa propre histoire. Séparer la science-fiction n'est pas une chose aisée, surtout lorsque en robotique nous cherchons parfois à faire réaliser la fiction. Un exemple de l'influence des fictions nous est donné par les lois de la robotique.

C'est au siècle dernier que la robotique industrielle a amorcé l'explosion des thèmes de recherche. A cette époque les robots étaient conçus en respectant les contraintes imposées par le milieu industriel, comme la répétitivité, la précision dans la réalisation des tâches, le respect des cadences de production, ..., etc. Les premiers robots à avoir été conçus dans ce domaine sont *les bras manipulateurs*, dont leur utilisation s'est peu à peu banalisée.

C'est avec les développements scientifiques, spécifiquement de l'électronique et de l'informatique, l'automatique, mathématique, mécanique et matériaux, que la technologie robotique a progressé. Les robots actuels sont dotés d'une 'intelligence' qui leur permet d'accomplir des missions complexes de manière autonome.

La robotique autonome n'est que la partie apparente. De nombreux autres domaines d'application sont naturellement concernés par de telles recherches. Cela permet la diffusion des robots dans de nouveaux domaines plus diversifiés et plus riches. Nous parlerons maintenant de robot d'exploration, de robots d'intervention, de robots de service, de robots ludique, ..., etc.

Généralement, nous pouvons définir un système robotique comme étant un système artificiel, équipé de capteurs et d'actionneurs, conçu pour agir sur le monde qui l'entoure. Il existe de nombreux types de systèmes robotiques, à titre d'exemple bras manipulateur, main articulée, véhicule terrestre ou aérien, robot humanoïde, robot parallèle, ..., etc.

L'élaboration d'une loi de commande pour un système non linéaire tel que le robot manipulateur nécessite la prise en compte de certains paramètres tels que le suivi de la consigne, le rejet de la perturbation, une marge de robustesse vis-à-vis de certains paramètres de procédé à piloter. La littérature propose une multitude de structure de commande. Chacune d'elle possède son application et également des propriétés (cas

linéaire, cas non linéaire, procédé stable, consigne d'un type donné... etc.). Commander un processus, c'est déterminer les commandes à lui appliquer, de manière à assurer aux variables d'états ou aux sorties qui nous intéressent un comportement précisé par un cahier des charges.

Le problème de la stabilité possède, en matière de commande, une importance fondamentale. La recherche d'une loi de commande garantissant la stabilité d'un système asservi peut se faire à l'aide des fonctions de Lyapunov.

Les théorèmes de Lyapunov ont été pendant longtemps des outils importants dans la théorie de commande non linéaire. Cependant, leur utilisation a été entravée par les difficultés de trouver une fonction de Lyapunov pour un système donné. La méthode de trouver une telle fonction a été souvent laissée à l'imagination et à l'expérience du concepteur[1]. En outre, plus la dynamique non linéaire de système est plus compliquée, plus cette fonction sera difficile à élaborer.

Au cours des deux dernières décennies, on a assisté à un développement extraordinaire dans l'élaboration de lois de commande non linéaires robustes telle que la commande par Backstepping, commande passive, ...etc.

La méthode de Backstepping est une procédure récursive utilisant la théorie de Lyapunov dans la recherche de la loi de commande et dans l'étude de la stabilité. Dans la technique du Backstepping, il s'agit de choisir une fonction de l'état comme étant l'entrée d'un sous-système et de procéder de la même manière récursivement jusqu'à obtenir la commande à appliquer au système global. Cette procédure de conception comme son nomen anglais "Backstepping" l'indique, veut dire marche arrière ou étape arrière ; puisque la procédure commence à la sortie du système et fait des pas en arrière à travers des intégrateurs du système en sélectionnant les valeurs désirées des composantes de l'état jusqu'à ce que l'entrée actuelle de la commande soit atteinte.

Dans la technique du Backstepping, il s'agit de trouver une fonction de Lyapunov qui permet de déduire une loi de commande pour le système tout en montrant la stabilité globale. Un problème majeur existe dans le développement de cette approche dite Backstepping. La difficulté réside dans la détermination des matrices de régression et du nombre des paramètres inconnus et augmente avec chaque étape du processus de

développement de la méthode du Backstepping. De plus, la supposition de la linéarité des paramètres ne peut pas être toujours vraie en pratique.

Outre l'introduction et la conclusion générales, ce mémoire est organisé en trois chapitres répartis comme suit :

- Le premier chapitre est dédié à une présentation des généralités sur bras manipulateurs et leurs modélisations
- Le deuxième chapitre est consacré à la modélisation dynamique de notre exemple un bras manipulateur à deux degrés de liberté et donner sa représentation d'état comme un système non linéaire complexe.
- Le troisième chapitre présente une étude théorique sur des commandes appliquées à la commande de robot manipulateur nous sommes intéressés à la commande par retour d'état et la commande par bouclage linéarisant (La linéarisation entrée-sortie) avec leurs applications sur notre système, On exposera les différentes étapes nécessaires à la mise au point de chaque commande, les résultats de simulation obtenus.
- Le quatrième chapitre est consacré à la commande par Backstepping en donnant tous les aspects théoriques avec leurs principes de développement nécessaires et l'application sur le robot manipulateur.

CHAPITRE I

BRAS

MANIPULATEURS

1 Généralité sur les bras manipulateurs

1.1 Introduction

Dans le monde industriel, l'automatisation est l'un des éléments les plus importants pour le développement. Elle aide à réduire le besoin d'humains et à augmenter l'efficacité et la productivité. Le domaine de l'automatisation occupe de vastes zones, principalement dans l'industrie de fabrication et en plus de cela ; l'automatisation est appliquée pour construire beaucoup de dispositifs sophistiqués qui sont utilisés quotidiennement tels que des équipements médicaux (radiographie machines, radiographie, etc.), réfrigérateurs, automobiles, etc. Résultats, le bras robotique est l'un d'entre eux, qui est largement utilisé dans l'industrie fine. Un bras robotique peut être comparé à une main humaine. Il a un joint tournant libre (rotation) et une articulation de translation (déplacement) pour le mouvement du bras. Ce mouvement du bras est généralement entraîné par un conducteur électrique (moteur) ou un pneumatique et un système hydraulique (pistons). Ces actionneurs sont contrôlés par un microcontrôleur (CPU), généralement programmable et conçu pour effectuer un ensemble de tâches séquentielles. La plupart de ces bras robotiques sont conçus pour être utilisés à des fins industrielles pour des performances fiables, aidant pour les productions de masse.

1.2 Origine des termes

☞ Le terme "**robot**" apparaît pour la première fois dans une pièce de Karel Capek en 1920: *Rossum's Universal Robots*. Il vient du tchèque "robota" (servitude) et présente une vision des robots comme serviteurs dociles et efficaces pour réaliser les tâches pénibles mais qui déjà vont se rebeller contre leurs créateurs[2].

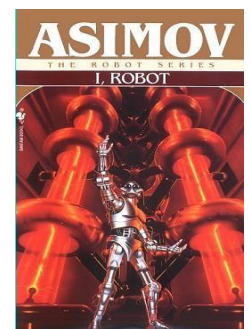


Karel Capek
1890-1938

☞ Le terme "**robotique**" a été employé pour la première fois par Asimov en 1941. Il recadre les robots en tant que machine servant l'homme et non dangereuse.

Asimov a proposé les trois lois de la robotique (I robot):

1. A robot may not injure a human being, or through inaction, allow a human being to come to harm.



2. A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.

3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Law [2].

1.3 Historique :

Le mot robot est dérivé du mot tchèque *robota* qui signifie travail ou corvée, il a été introduit pour la première fois en 1920 par l'écrivain tchèque Karel Capek dans sa pièce d'anticipation « Les Robots Universels de Rossem ». En 1947, le premier manipulateur électrique téléopéré a vu le jour. En 1954 on a assisté au premier robot manipulateur programmable, sept ans plus tard la société Animation a commercialisé le premier robot industriel mis en service dans une chaîne de montage de la société General Motors, et dans la même année le premier robot avec contrôle en effort est apparu. En 1963, on a assisté à l'utilisation de la vision pour commander un robot. Dès 1971 les grands laboratoires de recherche commencèrent à se développer tels que la JIRA (Association Japonaise de Robotique Industrielle), la RIA aux USA en 1975 [3]

1.4 Définition d'un robot

- La définition américaine par RIA (Robot Institut of America) : Un robot industriel est un manipulateur reprogrammable à fonctions multiples. Il est conçu pour déplacer des matériaux, des pièces, des outils ou des instruments spécialisés suivant des trajectoires variables programmées, en vue d'accomplir des tâches très diverses.

-L'Association Française de Normalisation (A.F.N.O.R.) définit un robot comme étant un système mécanique de type manipulateur commandé en position, reprogrammable, polyvalent (i.e., à usages multiples), à plusieurs degrés de liberté, capable de manipuler des matériaux, des pièces, des outils et des dispositifs spécialisés, au cours de mouvements variables et programmés pour l'exécution d'une variété de tâches. Il a souvent l'apparence d'un, ou plusieurs, bras se terminant par un poignet. Son unité de commande utilise, notamment, un dispositif de mémoire et éventuellement de perception et d'adaptation à l'environnement et aux circonstances. Ces machines polyvalentes sont généralement étudiées pour effectuer la même fonction de façon cyclique et peuvent être adaptées à d'autres fonctions sans modification permanente du matériel [2].

En général, il n'existe pas de définition particulière du robot qui satisfasse tout le monde. Mais elles tournent en général autour de celle-ci : *Un robot est une machine équipée de capacités de perception, de décision et d'action qui lui permettent d'agir de manière autonome dans son environnement en fonction de la perception qu'il en a.*

1.5 Classification

L'AFRI (Association Française de Robotique Industrielle) distingue quatre classes de robots :

-1- Les Télémanteurs ou manipulateurs à commande manuelle

Ils sont commandés à distance et "en temps réel" par un opérateur humain. Cette télécommande se fait à plus ou moins longue distance par signaux mécaniques, hydrauliques, ou le plus souvent électriques. Ces manipulateurs sont employés en forge, fonderie, meulage ébarbage, milieux "hostiles", etc...., mais nécessitent toujours la présence et l'intervention constante d'un opérateur.

-2- Les Manipulateurs automatiques à cycles prééglés

Leurs mouvements sont limités par des butées et cames réglables à la main. Ils sont commandés à l'aide de logiques à relais ou pneumatiques (séquences fixes), ou par automates programmables et cartes à microprocesseurs (séquences variables). Généralement modulaires, ces appareils sont conçus pour une application déterminée.

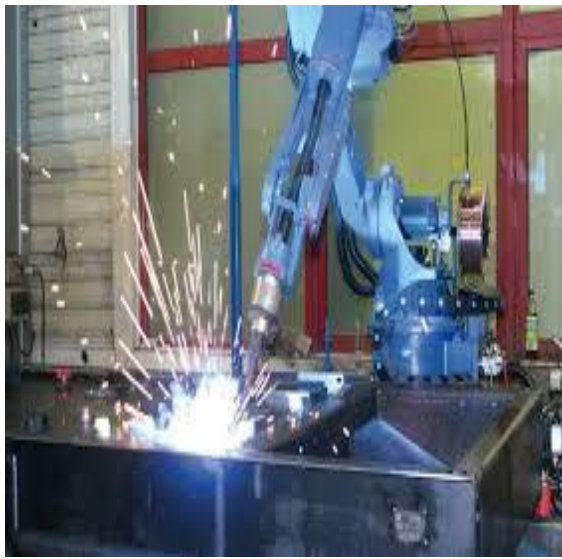
-3- Les robots programmables

Ils sont pilotés par des ordinateurs ou des armoires de commande numérique.

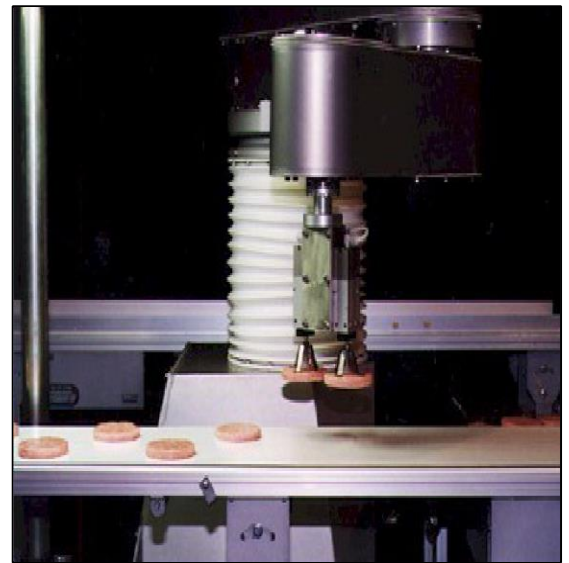
Leurs mouvements continus dans l'espace sont alors programmés par apprentissage ou en langage symbolique par l'intermédiaire d'un clavier, ou encore sur l'écran d'un poste de CAO. Ils assurent des manipulations complexes, des opérations de soudage, usinage, découpe, peinture et pulvérisation, etc...

-4- Les robots dits "intelligents"

Equipés de capteurs (par exemple un système de vision artificielle ou de suivi de joint en soudage), ils peuvent analyser les modifications de leur environnement ou de leur trajectoire et réagir en conséquence. Ces machines appelées robots de "deuxième génération" commencent à être répandus dans l'industrie. La "troisième génération" disposant de capacités de raisonnement grâce à l'intelligence artificielle fait aujourd'hui l'objet de recherches approfondies.



(a)



(b)



(c)



(d)

Figure 1: exemple de telemanipulateurs

1.6 Anatomie d'un bras manipulateur :

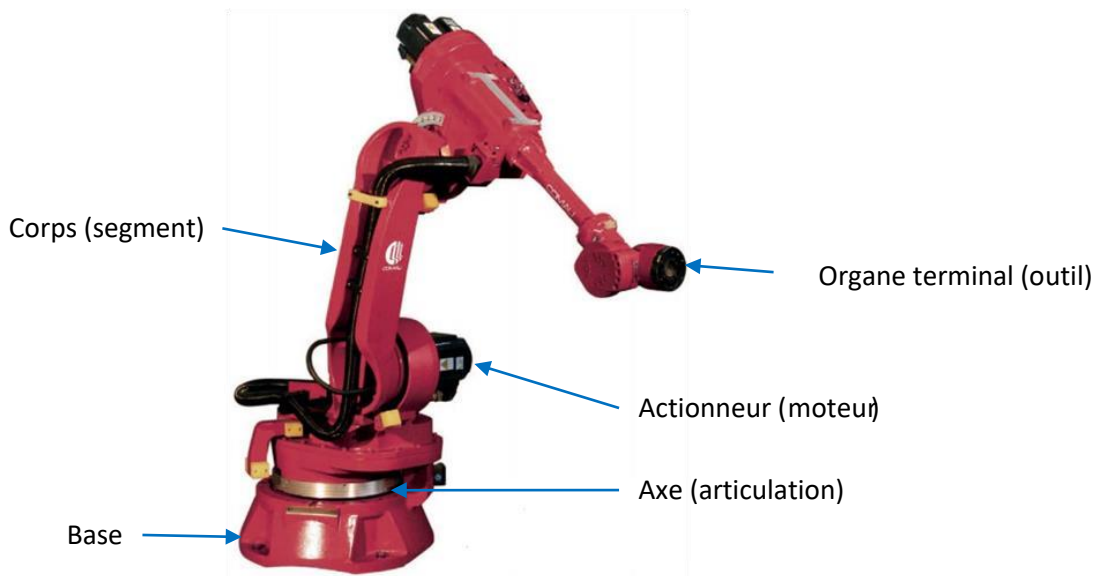


Figure 2: les constituants d'un bras manipulateur

a) L'organe terminal

L'organe terminal, regroupe tout dispositif destiné à manipuler des objets (dispositifs de serrage, dispositifs magnétiques, à dépression, ...), ou à les transformer (outils, torche de soudage, pistolet de peinture, ...). En d'autres termes, il s'agit d'une interface permettant au robot d'interagir avec son environnement. Un organe terminal peut être multi-fonctionnel, au sens où il peut être équipé de plusieurs dispositifs ayant des fonctionnalités différentes. Il peut aussi être mono-fonctionnel, mais interchangeable. Un robot, enfin, peut-être *multi-bras*, chacun des bras portant un organe terminal différent. On utilisera indifféremment le terme *organe terminal*, *préhenseur*, *outil* ou *effecteur* pour nommer le dispositif d'interaction fixé à l'extrémité mobile de la structure mécanique.

b) L'articulation

L'articulation lie deux corps successifs en limitant le nombre de degré de liberté de l'un par rapport à l'autre. Soit m le nombre de degré de liberté résultant, encore appelé *mobilité de l'articulation*. La mobilité d'une articulation est telle que : $0 \leq m \leq 6$. Lorsque $m = 1$; ce qui est fréquemment le cas en robotique, l'*articulation* est dite *simple* : soit *rotoïde*, soit *prismatique*.

c) Le système mécanique articulé (SMA)

Il s'agit d'une structure mécanique articulée dotée de plusieurs degrés de liberté (généralement six pour un robot destiné à la manutention d'objets), et équipée de moteurs qu'on appelle aussi *actuateurs*. Ces derniers peuvent être électriques ou pneumatiques pour les faibles charges (jusqu'à quelques décanewtons), ou hydrauliques pour les fortes charges (quelques centaines de newtons). Ils entraînent les articulations du manipulateur à travers des transmissions, elles aussi de natures diverses (câbles, courroies crantées, engrenages, bielles etc...). Pour connaître à chaque instant la position des articulations, on fait appel à des capteurs (potentiomètres, codeurs optiques ou incrémentaux, "resolvers", etc...) appelés « proprioceptifs » par analogie avec le système humain. Le système mécanique articulé peut être considéré à la fois comme un générateur de déplacements de l'outil dans l'espace géométrique, et un générateur de forces. Cette première entité correspond à ce qui est généralement montré dans une photographie de robot, et ce à quoi on pense dans la conversation courante quand on parle de robot. Pour notre part, nous emploierons indifféremment le terme de manipulateur ou de système mécanique articulé (SMA).

d) Les actionneurs

Le S.M.A. comporte des moteurs le plus souvent avec des transmissions (courroies crantées), l'ensemble constitue les *actionneurs*. Les actionneurs utilisent fréquemment des moteurs *électriques* à aimant permanent, à courant continu, à commande par l'induit (la tension n'est continue qu'en moyenne car en général l'alimentation est un hacheur de tension à fréquence élevée ; bien souvent la vitesse de régime élevée du moteur fait qu'il est suivi d'un réducteur, ce qui permet d'amplifier le couple moteur). On trouve de plus en plus de moteurs à commutation électronique (sans balais), ou, pour de petits robots, des moteurs pas à pas. Pour les robots devant manipuler de très lourdes charges (par exemple, une pelle mécanique), les actionneurs sont le plus souvent *hydrauliques*, agissant en translation (vérin hydraulique) ou en rotation (moteur hydraulique). Les actionneurs *pneumatiques* sont d'un usage général pour les *manipulateurs à cycles (robots tout ou rien)*. Un manipulateur à cycles est un S.M.A. avec un nombre limité de degrés de liberté permettant une succession de mouvements contrôlés uniquement par des capteurs de fin de course réglables manuellement à la course désirée (asservissement en position difficile dû à la compressibilité de l'air). (Positions et vitesses des articulations) et *extéroceptifs*⁵ lorsqu'ils recueillent des informations sur l'environnement (détection de présence, de contact, mesure de distance, vision artificielle).

e) Système de commande

Le système de commande synthétise les consignes des asservissements pilotant les actionneurs, à partir de la fonction de perception et des ordres de l'utilisateur.

f) Les capteurs

La perception permet de gérer les relations entre le robot et son environnement. Les organes de perception sont des *capteurs*. On peut distinguer entre deux classes :

- **Les capteurs proprioceptifs** : ils mesurent l'état interne du robot (positions et vitesses désarticulations).
- **Les capteurs extéroceptifs** : ils recueillent des informations sur l'environnement (détection de présence, de contact, mesure de distance, vision artificielle) [2][4] .

1.7 Caractéristiques d'un robot

Un robot doit être choisi en fonction de l'application qu'on lui réserve. Voici quelques paramètres à prendre, éventuellement, en compte :

- **La charge maximum transportable** (de quelques kilos à quelques tonnes), à déterminer dans les conditions les plus défavorables (en élongation maximum).
- **L'architecture du S.M.A.** : le choix est guidé par la tâche à réaliser (quelle est la rigidité de la structure ?).
- **L'espace de travail (*work space*)** : Défini comme l'ensemble des points atteignables par l'organe terminal. Tous les mouvements ne sont pas possibles en tout point du volume de travail [2].
- **L'espace de travail maximal (*reachable workspace*)** : est le volume de l'espace que le robot peut atteindre *via* au moins une orientation.
- **L'espace de travail dextre (*dextrous workspace*)**: est le volume de l'espace que le robot peut atteindre avec toutes les orientations possibles de l'effecteur (organe terminal). Cet espace de travail est un sous-ensemble de l'espace de travail maximal.
- **La précision** : ou l'erreur de positionnement absolu, correspondant à l'erreur entre un point souhaité (réel) – défini par une position et une orientation dans l'espace cartésien et le point

atteint et calculé *via* le modèle géométrique inverse du robot. Cette erreur est due au modèle utilisé, à la quantification de la mesure de position, à la flexibilité du système mécanique. En général, l'erreur de positionnement absolu est de l'ordre de 1 *mm* [2][5].

- **La répétabilité** : ce paramètre caractérise la capacité que le robot a à retourner vers un point (position, orientation) donné. La répétabilité correspond à l'erreur maximum de positionnement sur un point prédéfini dans le cas de trajectoires répétitives. En général, la répétabilité est de l'ordre de 0,1 *mm*
- **La vitesse de déplacement** (vitesse maximum en élongation maximum), accélération.
- **La masse du robot.**
- **Le coût du robot.**
- **La maintenance** [2][5].

2 Modélisation des bras manipulateur

2.1 Introduction

La conception et le contrôle des robots nécessitent certains modèles mathématiques, tels que :

- des modèles de transformation entre l'espace opérationnel (dans lequel la position l'effecteur terminal est défini) et l'espace articulaire (dans lequel la configuration du robot est défini). On distingue ce qui suit :

- modèles géométriques directs et inverses donnant l'emplacement de l'effecteur terminal (ou l'outil) en fonction des coordonnées conjointes du mécanisme.

- modèles cinématiques directs et inverses donnant la vitesse de l'effecteur terminal en termes des vitesses communes.

- des modèles dynamiques donnant les relations entre les couples ou les forces de l'actionneur, et les positions, vitesses et accélérations des articulations, Les erreurs de modélisation, les incertitudes sur l'estimation des paramètres physiques ainsi que les différentes perturbations externes influent beaucoup sur la qualité du contrôle. Tous ces facteurs doivent être pris en compte lors de la modélisation pour préserver la qualité et la précision des tâches effectuées par le robot, en effet la majorité des tâches confiées aux robots sont délicates et exigent une très grande précision.

2.2 Modèle géométrique

a) Modèle géométrique direct

Le modèle géométrique direct permet de déterminer la position et l'orientation de l'organe terminal du manipulateur par rapport à un repère de référence en fonction des variables articulaires, le modèle s'écrit :

$$X = f(q) \quad (\text{I.1})$$

Où

$q = [q_1, q_2, \dots, q_n]^T \in R^n$: Vecteur des variables articulaires.

$x = [x_1, x_2, \dots, x_m]^T \in R^m$: Vecteur des variables opérationnelles, $m \leq n$.

Par exemple, si le manipulateur se déplace dans l'espace on pose $m=6$ (3 coordonnées pour la position et 3 coordonnées pour la rotation). S'il se déplace dans un plan on pose $m=2$ et si en plus on est concerné par la rotation on pose $m=3$. La position de l'organe terminal peut être définie par des coordonnées cartésiennes, cylindriques ou sphériques. Le choix d'une structure particulière est guidé par les caractéristiques du robot, ainsi que par celle de la tâche à réaliser [6].

b) Modèle géométrique inverse

Le modèle géométrique inverse permet de déterminer le vecteur des variables articulaires à partir du vecteur de coordonnées opérationnelles, le modèle s'écrit :

$$q = f^{-1}(X) \quad (I.2)$$

Parmi les méthodes utilisées pour déterminer le modèle géométrique inverse on cite :

- **Les méthodes géométriques** : permettent de déterminer le vecteur q par utilisation des transformations géométriques en prenant avantage de la structure particulière du manipulateur considéré.
- **Les méthodes algébriques** : permettent de déterminer le vecteur q en effectuant des transformations algébriques sur l'équation (I.1). Parmi les méthodes utilisées on cite la méthode de Paul [6] qui consiste à multiplier successivement les deux membres du modèle géométrique direct par les matrices homogènes permettant ainsi d'isoler et d'identifier les variables articulaires l'une après l'autre. Lors de la résolution du problème géométrique inverse on rencontre pratiquement les situations suivantes[6][7].
 - Solutions en nombre fini, lorsqu'elles peuvent être calculées sans ambiguïté.
 - Aucune solution possible lorsque la position désirée ne peut être atteinte par le manipulateur.
 - Plusieurs solutions possibles lorsque le manipulateur est redondant ou lorsqu'il passe par une configuration singulière.

2.3 Modèle cinématique

a) Modèle cinématique direct

Le modèle cinématique directe permet de déterminer la vitesse de l'organe terminal dans l'espace opérationnel en fonction de la vitesse des variables articulaires.

Le modèle est décrit par l'équation :

$$\dot{X} = J(q)\dot{q} \quad (\text{I.3})$$

où $J(q) \in \mathbb{R}^{n \times m}$ est la matrice jacobienne.

L'une des méthodes utilisées pour le calcul de la matrice jacobienne est la dérivation du modèle géométrique direct :

$$J(q) = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \dots & \frac{\partial f_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial q_1} & \dots & \frac{\partial f_m}{\partial q_n} \end{bmatrix} \quad (\text{I.4})$$

b) Modèle cinématique inverse

Le modèle cinématique inverse permet de déterminer la vitesse des variables articulaires en fonction de la vitesse des variables opérationnelles. Pour les manipulateurs non redondants ($n=m$), le modèle s'écrit :

$$\dot{q} = J^{-1}(q)\dot{X} \quad (\text{I.5})$$

La solution de l'équation (I.5) existe si J est de rang plein, cela est valable tant que le manipulateur ne passe pas par une configuration singulière. Pour les manipulateurs redondants, le modèle cinématique inverse admet plusieurs solutions possibles. Le choix d'une parmi plusieurs est guidé par l'optimisation d'une fonction objective. Les méthodes employées pour résoudre un tel problème peuvent être classées en deux groupes [7][6].

- Méthodes d'optimisation cinématique.
- Méthodes d'optimisation dynamique.

2.4 Modèle dynamique :

On convient d'appeler *modèle dynamique inverse*, ou tout simplement *modèle dynamique* la relation entre les couples (et/ou force) appliqués aux actionneurs et les positions, vitesses et accélérations articulaires. On le représente par une relation de la forme : $\Gamma = f(q, \dot{q}, \ddot{q}, F_e)$

Γ : vecteur des couples/forces des actionneurs (articulation rotoïde ou prismatique) ;

q : vecteur des positions articulaires ;

\dot{q} : vecteur des vitesses articulaires ;

\ddot{q} : vecteur des accélérations articulaires ;

F_e : vecteur représentant l'effort extérieur qu'exerce le robot sur l'environnement.

Le modèle dynamique direct est celui qui exprime les accélérations articulaires en fonction des positions, vitesses et couples des articulations. Il est présenté par :

$$\ddot{q} = f(q, \dot{q}, \Gamma, F_e).$$

Le modèle dynamique est utilisé dans plusieurs buts parmi lesquels on peut citer :

- la simulation, qui utilise le modèle dynamique directe ;
- la commande qui utilise le modèle dynamique directe ;
- l'identification des paramètres inertiels et des paramètres de frottement du robot.

La détermination du modèle dynamique se fait avec plusieurs formalismes dont les plus utilisés sont :

✓ **Formalisme de Lagrange** : permet de décrire les équations du mouvement des mécanismes par l'équation suivante :

$$\Gamma_j = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_j} - \frac{\partial L}{\partial q_j}$$

Γ_j : le couple généralisé associé à $j^{\text{ème}}$ articulation. Il correspond à la somme des forces externes (couple moteur, force appliquée à l'organe terminale,) et des forces dissipatives (frottement, amortissements internes).

q_j : la coordonnée généralisée correspondante la $j^{\text{ème}}$ articulation ;

L : le Lagrangien du système se définit comme la différence entre l'énergie cinétique totale K du système et l'énergie potentielle totale U du système :

$$L = K - U$$

L'énergie cinétique du système est donnée par la relation :

$$K = \sum_{j=1}^n k_j$$

, où :

k_j est l'énergie cinétique du corps C_j qui s'exprime par :

$$k_j = \frac{1}{2} (\omega_j^T I_{G_j} \omega_j + m_j V_{G_j}^T V_{G_j})$$

ω_j : vitesse de rotation du corps C_j ;

I_{G_j} : inertie de C_j par rapport à un repère parallèle à R_j et d'origine son centre de gravité ;

m_j : masse du corps C_j ;

v_{G_j} : vitesse linéaire du centre de gravité du corps C_j ;

L'énergie potentielle du système est donnée par : $U = \sum_{j=1}^n U_j$

U_j : est l'énergie cinétique du corps C_j qui s'exprime par : $u_j = m_j \cdot g^T \cdot T_{G_j} + U_{ref}$

g : vecteur de gravité;

◦ T_{G_j} : transformation homogène localisant le centre de gravité du corps C_j ;

u_{ref} : constante choisie de sorte que la valeur minimale de u_j soit nulle .

✓ **Formalisme de Newton-Euler** : exprime le torseur dynamique des efforts extérieurs sur un corps C_j en son centre de gravité par les équations :

$$F_j = m_j \dot{v}_{G_j} \quad \text{et} \quad M_{G_j} = I_{G_j} \dot{\omega}_j + \dot{\omega}_j (I_{G_j} \omega_j)$$

F_j : résultante des forces extérieures sur le corps C_j .

\dot{v}_{G_j} : accélération linéaire du centre de gravité du corps C_j .

M_{G_j} : moment des efforts extérieurs exercés sur C_j autour de son centre de gravité.

$\dot{\omega}_j$: accélération de rotation du corps C_j .

✓ Comparaison entre les deux formalismes

Le formalisme de Lagrange est le plus adapté pour le calcul du modèle dynamique direct, il est souvent employé pour la modélisation et la simulation du système. Par contre le formalisme de Newton-Euler est utilisé pour la commande en temps réel car il offre une réduction considérable dans le temps de calcul du modèle dynamique inverse comme le montre le tableau I.1.

Tableau 1.1. Nombre d'opérations effectuées par chaque formalisme [6]

Formalisme	Multiplication	Addition
Lagrange	$32\frac{1}{2}n^4 + 86\frac{5}{12}n^3 + 171\frac{1}{4}n^2 + 53\frac{1}{3}n - 128$	$25n^4 + 66\frac{1}{3}n^3 + 129\frac{1}{2}n^2 + 42\frac{1}{3}n - 96$
Newton-Euler	150n-48	131n-48

3 Conclusion:

Dans ce chapitre nous avons exposé des généralités sur les robots manipulateurs et les différents modèles utilisés pour décrire les mouvements des articulations d'un manipulateur. Nous avons montré que :

- ✓ Le choix du modèle géométrique est fait selon le type de la tâche à réaliser ainsi que l'architecture du manipulateur.
- ✓ Le calcul du modèle cinématique inverse pour les manipulateurs redondants est réalisé en définissant un objectif secondaire en plus de l'objectif principal, afin de choisir une solution parmi plusieurs. Cet objectif consiste à optimiser une fonction de nature cinématique ou dynamique.
- ✓ Le calcul du modèle dynamique peut être effectué soit par la méthode de Newton-Euler ou par celle de Lagrange. La première nécessite moins de calcul, elle est destinée pour des applications en temps réel. Tandis que la deuxième s'avère intéressante pour la simulation.

Chapitre II

Modélisation dynamique
d'un bras manipulateur a
deux DDL

1. Définition et propriété des systèmes non linéaires:

1.1 Introduction

La plupart des systèmes physiques (procédés) qui nous entourent sont non linéaires. Bien souvent, ces non-linéarités sont faibles ou ne sont pas visibles sur la plage d'opérations de ces procédés. Le souci constant d'améliorer les performances des systèmes commandés conduit à des modélisations de plus en plus précises qui permettent de répondre sur une plus large plage d'opérations. C'est à ce moment que les non-linéarités se font sentir et rendent les outils d'analyse et/ou de synthèse des lois de commande, utilisés dans le domaine linéaire, caduques et absolument incapables de rendre compte de certains phénomènes. C'est pourquoi, depuis quelques années, beaucoup de recherche ont été effectuées dans le domaine de la commande des systèmes non linéaires.

1.2 Définition :

Tout système obéissant au principe de superposition est un système linéaire.

Par conséquent, tout système qui n'obéit pas au principe de superposition est un système non linéaire [8].

Un système non linéaire est un ensemble d'équations mathématiques non linéaires qui représentent une réalité physique provenant de plusieurs disciplines : biologique, génie mécanique, électrique, chimique, ...etc.

L'étude des systèmes non linéaires se confronte à de plusieurs difficultés :

- L'analyse par les fonctions de transfert est impossible.
- L'absence de la notion des pôles.
- L'existence de plusieurs points d'équilibre et la complexité de l'étude de leur stabilité.

Généralement, les systèmes non linéaires sont présentés sous la forme suivante :

$$\begin{cases} \dot{x} = \begin{bmatrix} x_1 \\ \cdot \\ x_n \end{bmatrix} \\ y = f(x(t), u(t), t) = \begin{bmatrix} f_1(x(t), u(t), t) \\ \vdots \\ f_n(x(t), u(t), t) \end{bmatrix} \\ h(x), u(t), t \end{cases} \quad \text{II.1}$$

Où:

t est le temps

$x(t) \in \mathbb{R}^n$ est le vecteur d'état

$u(t) \in \mathbb{R}^m$ est le vecteur d'entrée (ou de commande)

f et h : sont des fonctions vectorielles et $f: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}_+ \rightarrow \mathbb{R}^n$ est non linéaire.

1.3 Principe de superposition

Soit un deux signaux d'entrées u_1 et u_2 engendrant deux signaux de sorties y_1 et y_2 .

La réponse à la somme des entrées $u = u_1 + u_2$ est la somme des réponses individuelles

$y = y_1 + y_2$.

2. De la non-linéarité à la linéarité

2.1 Points d'équilibre

Un système se trouve dans un état d'équilibre (par état, l'on entend un point de l'espace d'état c-à-d une instance du vecteur d'état) si cet état n'est pas modifié lorsque le système est abandonné à lui-même.

Considérons un système non linéaire représenté par $\dot{x} = f(x)$

On dit qu'un état x^* est un état d'équilibre (ou un point d'équilibre) du système :

si une fois $x(t_0) = x^*$, alors pour tout $t \geq t_0$: $x(t) = x^*$. Cela signifie mathématiquement que le vecteur x^* satisfait l'équation : $f(x^*) = 0$ (II.2)

Donc la recherche des points d'équilibre d'un système décrit par (II.1) revient à poser $u = 0$ et à résoudre (II.2). [9], [10]

2.2 La linéarisation autour d'un point d'équilibre

Les modèles non linéaires sont par essence difficiles à manipuler. Cela signifie en pratique qu'ils rendent ardues l'analyse du comportement du système et, plus encore, sa commande.

Par conséquent, même si c'est une entorse au principe de description fidèle de la dynamique du système, l'on décide bien souvent de travailler dans une gamme de valeurs des grandeurs se situant autour de

valeurs centrales constituant ce qu'il est convenu d'appeler un point de fonctionnement (ou point d'équilibre). [9]

Considérons le système non linéaire (II.1), et supposons que le point d'équilibre est

caractérisé par les valeurs x_e et u_e , donc on peut considérer la variation d'état

$\xi(t) = x(t) - x_e$, celle de l'entrée $v(t) = u(t) - u_e$ et celle de sortie

$z(t) = y(t) - g(x_e, u_e, t)$.

Si les valeurs de $v(t)$ et des $\xi_i(t)$ restent faibles (donc assimilables aux différentielles correspondantes), l'on peut calculer les matrices Jacobiennes suivantes:

$$A(t) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(x_e, u_e, t) & \dots & \frac{\partial f_1}{\partial x_n}(x_e, u_e, t) \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1}(x_e, u_e, t) & \dots & \frac{\partial f_n}{\partial x_n}(x_e, u_e, t) \end{bmatrix}; B(t) = \begin{bmatrix} \frac{\partial f_1}{\partial u_1}(x_e, u_e, t) & \dots & \frac{\partial f_1}{\partial u_m}(x_e, u_e, t) \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial u_1}(x_e, u_e, t) & \dots & \frac{\partial f_n}{\partial u_m}(x_e, u_e, t) \end{bmatrix}$$

$$C(t) = \left[\frac{\partial h}{\partial x_1}(x_e, u_e, t) \dots \frac{\partial h}{\partial x_n}(x_e, u_e, t) \right]; D(t) = \frac{\partial h}{\partial u}(x_e, u_e, t)$$

Ainsi, si l'on considère que $\xi(t)$ est le nouveau vecteur d'état et $v(t)$ le nouveau vecteur d'entrée du modèle " linéarisé tangent » autour du point d'équilibre, il vient :

$$\begin{cases} \dot{\xi} = A\xi + Bv \\ z = C\xi + Dv \end{cases} \quad \text{II.3}$$

L'approximation linéaire au premier ordre consiste donc à supposer que $\xi(t)$ et $v(t)$ restent faibles et à décrire le comportement du système par (II.3). Comme $x(t)$, $u(t)$ et $y(t)$ sont les notations habituelles de l'état, de l'entrée et de la sortie, l'on écrira plutôt :

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \quad \text{II.4}$$

3. Modélisation dynamique du bras manipulateur :

3.1 Introduction :

Comme nous l'avons énoncé dans le premier chapitre, qu'afin de synthétiser une commande adéquate pour un bras manipulateur, il est nécessaire d'établir un modèle mathématique précis qui décrit d'une façon exacte le comportement du robot, une chose qui n'est pas évidente. En effet, les bras manipulateurs sont des systèmes fortement couplés et non linéaires et sont soumis à plusieurs perturbations externes ce qui augmente la complexité du modèle et par conséquent, quelques approximations doivent être prises.

Dans la suite, on négligera les frottements secs et visqueux et les couples de perturbation externes et on utilisera le formalisme de Lagrange pour établir le modèle dynamique d'un bras manipulateur qui est un robot planaire à deux degrés de liberté constitué de deux tiges rigides de masses m_1 , m_2 et de longueurs l_1 , l_2 avec deux articulations rotoïdes $q_1 = q_1$ et $q_2 = q_2$ dont la rotation de chacune est assurée par un servomoteur à entraînement direct comme il est représenté dans la figure suivante[11] :

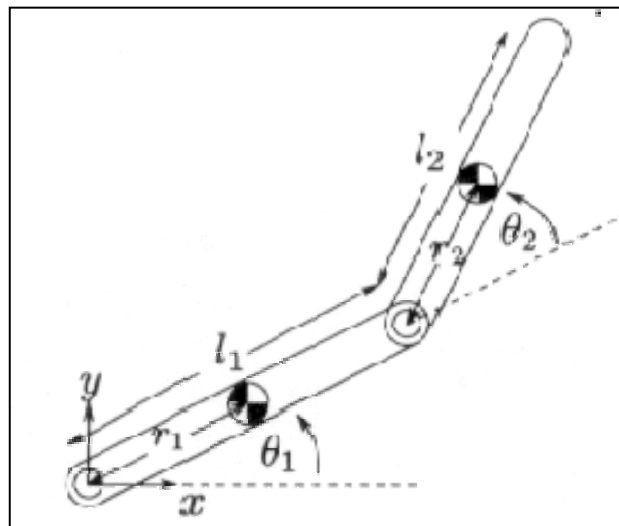


Figure 3: Robot planaire à 2 deux degré de liberté

3.2 modèle géométrique et cinématique directes

A partir de la figure 1, on peut déterminer facilement la relation entre les variables articulaires q_1, q_2 et les coordonnées opérationnelles x, y (le MGD) des centres de gravité de chaque tige comme :

$$\left\{ \begin{array}{l} x_1 = r_1 \cos \theta_1 \\ y_1 = r_1 \sin \theta_1 \end{array} \right. \quad \text{et} \quad \left\{ \begin{array}{l} x_2 = l_1 \cos \theta_1 + r_2 \cos(\theta_1 + \theta_2) \\ y_2 = l_1 \sin \theta_1 + r_2 \sin(\theta_1 + \theta_2) \end{array} \right\} \quad (\text{III.1})$$

A partir de (III.1), on peut calculer les vitesses linéaires des centres de gravité en fonction des vitesses articulaires (le MCD) ce qui donne :

$$\left\{ \begin{array}{l} \dot{x}_1 = -r_1 \dot{\theta}_1 \sin \theta_1 \\ \dot{y}_1 = r_1 \dot{\theta}_1 \cos \theta_1 \end{array} \right. \quad \text{et} \quad \left\{ \begin{array}{l} \dot{x}_2 = -l_1 \dot{\theta}_1 \sin \theta_1 - r_2 (\dot{\theta}_1 + \dot{\theta}_2) \sin(\theta_1 + \theta_2) \\ \dot{y}_2 = l_1 \dot{\theta}_1 \cos \theta_1 + r_2 (\dot{\theta}_1 + \dot{\theta}_2) \cos(\theta_1 + \theta_2) \end{array} \right\} \quad (\text{III.2})$$

Sachant que $v_2^2 = \dot{x}_2^2 + \dot{y}_2^2$ et d'après (III.2) on obtient :

$$\left\{ \begin{array}{l} v_1^2 = r_1^2 \dot{\theta}_1^2 (\sin^2 \theta_1 + \cos^2 \theta_1) = r_1^2 \dot{\theta}_1^2 \\ v_2^2 = l_1^2 \dot{\theta}_1^2 + r_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + 2l_1 r_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \cos \theta_2 \end{array} \right. \quad (\text{III.3})$$

3.3 Formalisme de Lagrange :

3.3.1 L'énergie cinétique du système :

$$\left\{ \begin{array}{l} K_1 = \frac{1}{2} m_1 v_1^2 + \frac{1}{2} I_1 \dot{\theta}_1^2 = \frac{1}{2} m_1 r_1^2 \dot{\theta}_1^2 + \frac{1}{2} I_1 \dot{\theta}_1^2 \\ K_2 = \frac{1}{2} m_2 v_2^2 + \frac{1}{2} I_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 = \frac{1}{2} m_2 (l_1^2 \dot{\theta}_1^2 + r_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + 2l_1 r_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \cos \theta_2) + \frac{1}{2} I_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 \end{array} \right.$$

d'où :

$$K = \frac{1}{2} m_1 r_1^2 \dot{\theta}_1^2 + \frac{1}{2} m_2 (l_1^2 \dot{\theta}_1^2 + r_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + 2l_1 r_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \cos \theta_2) + \frac{1}{2} I_1 \dot{\theta}_1^2 + \frac{1}{2} I_2 (\dot{\theta}_1 + \dot{\theta}_2)^2$$

3.3.2 L'énergie potentielle du système :

$$\left\{ \begin{array}{l} U_1 = m_1 g r_1 \sin \theta_1 \\ U_2 = m_2 g (l_1 \sin \theta_1 + r_2 \sin(\theta_1 + \theta_2)) \end{array} \right.$$

d'où : $U = m_1 g r_1 \sin \theta_1 + m_2 g (l_1 \sin \theta_1 + r_2 \sin(\theta_1 + \theta_2))$

3.3.3 Les équations de Lagrange:

Le Lagrangien du système est: $L = K - U$, et les équations de Lagrange sont:

$$\text{-) pour la première articulation : } \tau_1 = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_1} \right) - \frac{\partial L}{\partial \theta_1} \quad (\text{III.4})$$

$$\frac{\partial L}{\partial \dot{\theta}_1} = (m_1 r_1^2 + m_2 l_1^2 + m_2 l_1 r_2 \cos(\theta_2) + I_1) \dot{\theta}_1 + (m_2 r_2^2 + m_2 l_1 r_2 \cos \theta_2 + I_2) (\dot{\theta}_1 + \dot{\theta}_2)$$

$$\text{d'où : } \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_1} \right) = M_{11} \ddot{\theta}_1 + M_{12} \ddot{\theta}_2 - m_2 l_1 r_2 (2\dot{\theta}_2 \dot{\theta}_1 + \dot{\theta}_2^2) \sin(\theta_2) \quad (\text{III.5})$$

$$\text{avec: } \begin{cases} M_{11} = m_1 r_1^2 + m_2 (l_1^2 + r_2^2) + 2m_2 l_1 r_2 \cos(\theta_2) + I_1 + I_2 \\ M_{12} = m_2 r_2^2 + m_2 l_1 r_2 \cos \theta_2 + I_2 \end{cases} \quad (\text{III.6})$$

$$-\frac{\partial L}{\partial \theta_1} = (m_1 r_1 + m_2 l_1) g \cos(\theta_1) + m_2 r_2 g \cos(\theta_1 + \theta_2) \quad (\text{III.7})$$

En faisant la sommation de (III.5) et (III.7), et en tenant compte de (III.4) on conclue que:

$$\tau_1 = M_{11} \ddot{\theta}_1 + M_{12} \ddot{\theta}_2 + C_1 \quad (\text{III.8})$$

$$\text{Avec: } C_1 = -m_2 l_1 r_2 (2\dot{\theta}_2 \dot{\theta}_1 + \dot{\theta}_2^2) \sin(\theta_2) + (m_1 r_1 + m_2 l_1) g \cos(\theta_1) + m_2 r_2 g \cos(\theta_1 + \theta_2) \quad (\text{III.9})$$

$$\text{-) pour la deuxième articulation : } t_2 = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_2} \right) - \frac{\partial L}{\partial \theta_2} \quad (\text{III.10})$$

$$\frac{\partial L}{\partial \dot{\theta}_2} = (m_2 r_2^2 + I_2) (\dot{\theta}_1 + \dot{\theta}_2) + m_2 l_1 r_2 \cos(\theta_2) \dot{\theta}_1$$

$$\text{d'où : } \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_2} \right) = M_{21} \ddot{\theta}_1 + M_{22} \ddot{\theta}_2 + m_2 l_1 r_2 \sin(\theta_2) \dot{\theta}_2 \dot{\theta}_1 \quad (\text{III.11})$$

$$\text{avec : } \begin{cases} M_{21} = m_2 r_2^2 + m_2 l_1 r_2 \cos(\theta_2) + I_2 \\ M_{22} = m_2 r_2^2 + I_2 \end{cases} \quad (\text{III.12})$$

$$-\frac{\partial L}{\partial \theta_2} = m_2 l_1 r_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \sin(\theta_2) + m_2 r_2 g \cos(\theta_1 + \theta_2) \quad (\text{III.13})$$

En faisant la sommation de (III.11) et (III.13), et en tenant compte de (III.10) on conclue que:

$$\tau_2 = M_{21} \ddot{\theta}_1 + M_{22} \ddot{\theta}_2 + C_2 \quad (\text{III.14})$$

$$\text{Avec: } C_2 = m_2 l_1 r_2 \dot{\theta}_1^2 \sin(\theta_2) + m_2 r_2 g \cos(\theta_1 + \theta_2) \quad (\text{III.15})$$

3.4 La représentation d'état du robot

En regroupant les équations (III.8), (III.14) et en utilisant la notation matricielle on trouve:

$$\begin{bmatrix} r_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}$$

d'où :

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}^{-1} \begin{bmatrix} \tau_1 - C_1 \\ \tau_2 - C_2 \end{bmatrix}$$

Alors:

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \frac{1}{\det(M)} \begin{bmatrix} M_{22} & -M_{12} \\ -M_{21} & M_{11} \end{bmatrix} \begin{bmatrix} \tau_1 - C_1 \\ \tau_2 - C_2 \end{bmatrix} \quad (\text{III.16})$$

Avec:

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$$

Pour avoir la représentation d'état, choisissons :
$$\begin{cases} x_1 = \theta_1 \\ x_2 = \dot{\theta}_1 \\ x_3 = \theta_2 \\ x_4 = \dot{\theta}_2 \end{cases} \text{ d'où : } \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \ddot{\theta}_1 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \ddot{\theta}_2 \end{cases} \quad (\text{III.17})$$

Et considérons que les angles θ_1, θ_2 sont les sorties du système et les couples t_1, r_2 sont les entrées. D'après (III. 16) on peut écrire la représentation (III.17) sous la forme :

$$\begin{cases} \dot{x} = f(x, u) \\ y = h(x) \end{cases} \quad (\text{III.18})$$

$$\text{avec: } f(x, u) = \begin{cases} \frac{-M_{22}C_1 + M_{12}C_2}{\det(M)} x_2 + \frac{M_{22}u_1 - M_{12}u_2}{\det(M)} \\ \frac{M_{21}C_1 - M_{11}C_2}{\det(M)} x_4 + \frac{-M_{21}u_1 + M_{11}u_2}{\det(M)} \end{cases} \text{ et } h(x) = \begin{cases} x_1 \\ x_3 \end{cases}$$

3.5 Application numérique

Fin remplaçant dans (III.6), (III.9), (III.12) et (III. 15), les valeurs numériques suivantes:

$$m_1 = 1 \text{ kg}, m_2 = 0.5 \text{ kg}, l_1 = 0.5 \text{ m}, l_2 = 0.25 \text{ m}, g = 9.8 \text{ ms}^{-2}$$

$$\text{On aura: } \begin{cases} M_{11} = 0.289 + 0.0625 \cos(x_3) \\ M_{12} = M_{21} = 0.01823 + 0.03125 \cos(x_3) \\ M_{13} = 0.01823 \\ \det(M) = 0.0060755 - 0.0011394 \cos(x_3) - 0.00098 \cos^2(x_3) \end{cases}$$

$$\text{Et: } \begin{cases} C_1 = -0.03125(x_2^2 + 2x_2x_4) \sin(x_3) + 4.905 \cos(x_1) + 0.613 \cos(x_1 + x_3) \\ C_2 = 0.03125x_2^2 \sin(x_3) + 0.613 \cos(x_1 + x_3) \end{cases}$$

3.6 La linéarisation du modèle

3.6.1 Point d'équilibre

Notre objectif dans cette section c'est de linéariser le modèle du robot au voisinage d'un point d'équilibre qu'on appelle souvent point de fonctionnement.

Choisissant le point $x_0 = \left[\frac{\pi}{2} \quad 0 \quad 0 \quad 0 \right]^T$ et $u_0 = [0 \quad 0]^T$ on trouve : $f(x_0, u_0) = \begin{cases} 0 \\ 0 \\ 0 \\ 0 \end{cases}$

Alors la condition (II 2) est bien satisfaite,

3.6.2 La linéarisation locale

Pour trouver le modèle linéarisé du robot autour du point (x_0, u_0) , on a calculé les matrices

Jacobiennes de (II.3) et on a obtenu le système linéaire : $\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases}$ (III. 19)

Avec:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 17.4925 & 0 & -4.843 & 0 \\ 0 & 0 & 0 & 1 \\ -14.5423 & 0 & 46.8058 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 4.6082 & -12.5075 \\ 0 & 0 \\ -12.5075 & 88.8524 \end{bmatrix} \text{ et } \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

4. conclusion :

Dans ce chapitre, nous avons fait un petit rappel sur les systèmes non linéaires ensuite on a appliqué la modélisation dynamique sur un bras manipulateur a deux DDL qui résulte un système non linéaire complexe a deux entrés et quatre sorties.

La commande de ce bras manipulateur est notre objective dans les deux chapitres suivants.

CHAPITRE III

**COMMANDE PAR RETOUR
D'ETAT**

ET

**COMMANDE PAR BOUCLAGE
LINEARISANT**

1. Introduction :

Le problème de la linéarisation exacte et découplage entrées-sorties, consiste à transformer de manière exacte un système non linéaire en un système linéaire découplé et commandable, à l'aide d'un bouclage d'état et d'un changement de coordonnées sur l'état du système. Lorsque cette transformation existe, elle permet de stabiliser le système en utilisant les méthodes de commande des systèmes linéaires (PID, placement de pôles, synthèse quadratique...). Cette technique de commande est basée sur la théorie de la géométrie différentielle et nécessite la connaissance du modèle du système avec une précision suffisante. Les propriétés de robustesse sont peu garanties face aux incertitudes paramétriques. Il est possible, à partir de la théorie de Lypunov, d'utiliser une commande par linéarisation adaptative afin de contrôler un système ayant des paramètres inconnus ainsi que des perturbations extérieures.

2. La commande par retour d'état:

Considérons dans ce qui suit un système non linéaire où sa linéarisation autour d'un point de fonctionnement est définie par la représentation d'état :

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases} \quad (\text{II.5})$$

Où : $x(t) \in \mathbb{R}^n$ représente vecteur de l'état du système

$u(t) \in \mathbb{R}^m$ est le vecteur de l'entrée (la commande)

$y(t) \in \mathbb{R}^p$ est le vecteur des sorties mesurées du système.

On désire asservir ce système à une valeur $y_{ref}(t)$ tout en imposant les dynamiques du régime transitoire et en maintenant une erreur petite ou nulle en régime permanent.

Le fait que la commande par retour d'état est une commande élaborée pour les systèmes linéaires n'empêche pas à l'implanter pour des systèmes non linéaires à condition d'assurer un fonctionnement proche du point d'équilibre [11].

2.1 Calcul de la matrice des gains de retour :

Modifier le régime transitoire du système (II.5), c'est modifier les pôles de la matrice dynamique A .

Nous pouvons modifier tous les pôles du système si et seulement si le système est commandable,

c'est-à-dire si et seulement si $\text{rang}(\mathcal{E}) = n$ où :

$$\mathcal{E} = \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix} \text{ est appelée la matrice de commandabilité.}$$

On implante ainsi une loi de commande par retour d'état qui prend en compte les valeurs de

$$\text{l'état à l'instant } t : \begin{bmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{bmatrix}$$

$$\text{La loi de commande s'écrit alors : } u(t) = -Kx(t) + v(t) \quad (\text{II.6})$$

où : $K \in \mathbb{R}^{n \times m}$ est appelée la matrice des gains du retour d'état

$v(t)$ est une nouvelle entrée pour le système en boucle fermée (elle représente éventuellement la consigne).

C'est une commande en boucle fermée car elle dépend des signaux internes du système même si elle ne prend pas en compte directement la sortie du système $y(t)$ comme le montre la figure suivante :

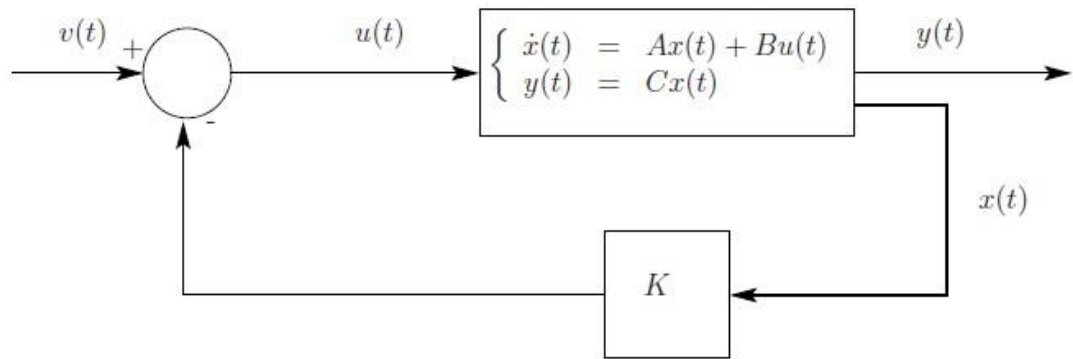


Figure 11 : Le schéma bloc du commande par retour d'état

D'après les relations (II.5) et (II.6), Le système en boucle fermée s'écrit :

$$\begin{cases} \dot{x}(t) = (A - BK)x(t) + Bv(t) \\ y(t) = Cx(t) \end{cases} \quad (\text{II.7})$$

Donc le polynôme caractéristique sera noté :

$$P_{A-BK}(\lambda) = \det((A - BK) - I\lambda) = \lambda^n + a_{n-1}\lambda^{n-1} + \dots + a_0$$

les coefficients $a_{n-1}; \dots; a_0$ du polynôme caractéristique sont en fonction des gains.

En choisissant les pôles $p_n; p_{n-1}; \dots; p_0$, le polynôme caractéristique désiré sera :

$$P_{dési}(\lambda) = (\lambda - p_n)(\lambda - p_{n-1}) \dots (\lambda - p_0) = \lambda^n + b_{n-1}\lambda^{n-1} + \dots + b_0$$

les coefficients $b_{n-1}; \dots; b_0$ du polynôme désiré sont en fonction des pôles.

Alors une fois le système d'équations
$$\begin{cases} a_{n-1} = b_{n-1} \\ \vdots \\ a_0 = b_0 \end{cases}$$
 est résolu, on aura déterminé les valeurs

des gains de la matrice K qui force le système à suivre la dynamique imposée par les pôles placés et donc avoir un régime transitoire convenable et assurer la stabilité du système par le choix des pôles.

2.2 Réglage du régime permanent :

Le choix des pôles du système permet de régler sa dynamique. Par contre, cette technique ne permet pas de régler le problème de la précision. Nous ne pouvons pas choisir le régime permanent du système en boucle fermée par le choix de la matrice K .

Une première structure de commande permettant d'assurer une erreur de position nulle en régime permanent est : $u(t) = Kx(t) + Ny_{ref}(t)$ (II.8)

où : N est un gain matriciel appelé gain précompensateur pour régler le gain statique du système en boucle fermée.

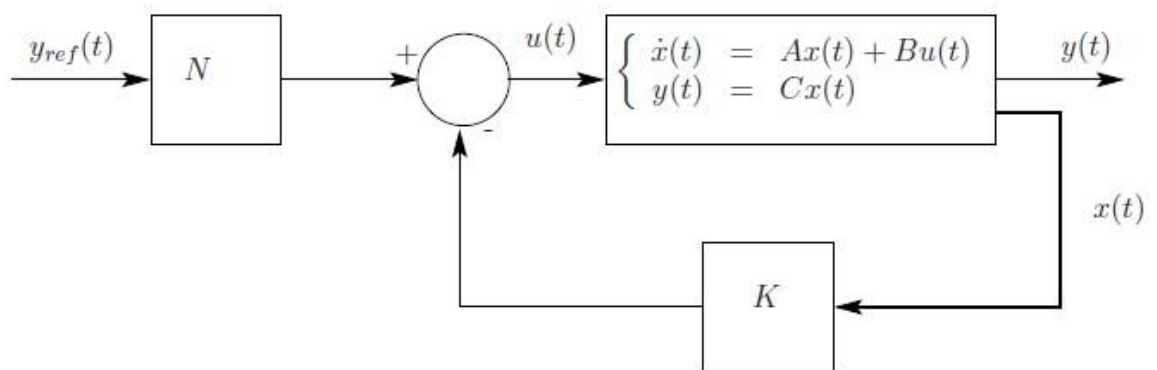


Figure 12 : Commande par retour d'état et gain précompensateur

Considérons pour des raisons de simplicité que le système est SISO (qui signifie mono entrée mono sortie) :

D'après (II.5) et (II.8), la fonction de transfert associé du système en boucle fermée s'écrit :

$$G_{BF}(p) = C(pI - A + BK)^{-1}BN \quad (\text{II.9})$$

Supposons que $y_{ref}(t) = y_{ref}$ est une constante. L'objectif de la commande est d'avoir un régime permanent avec une erreur qui converge vers le 0 donc $\lim_{t \rightarrow \infty} y_{ref}(t) = y_{ref}$, et pour que cela aura lieu, il est nécessaire que $G_{BF}(0) = 1$ d'où :

$$N = \frac{1}{C(-A + BK)^{-1}B}$$

Pour les système MIMO(multi entrées multi sorties) , le calcul de la matrice N se fait par la relation :

$$N = \left(C(-A + BK)^{-1}B \right)^{-1} \tag{II.10}$$

La deuxième idée pour assurer un régime permanent sans erreur réside dans l'ajout d'un intégrateur dans la chaîne directe. Pour montrer l'intérêt d'une telle commande, choisissons un système SISO modélisé à l'aide de la forme commandable et qui ne possède pas de zéros.

Dans ce cas, les états du système sont la sortie et ses dérivées successives $\begin{cases} x_1 = y(t) \\ \vdots \\ x_n = y^{n+1}(t) \end{cases}$

Le correcteur implanté est défini par le retour d'état suivant: $u(t) = -Kx(t) + e(t)$

Où $e(t)$ est la consigne. Ce retour d'état peut donc s'exprimer à l'aide du formalisme de Laplace :

$$U(p) = E(p) - (k_1 + k_2p + \dots + k_np^{n-1})Y(p)$$

On obtient alors le schéma bloc du système bouclé suivant :

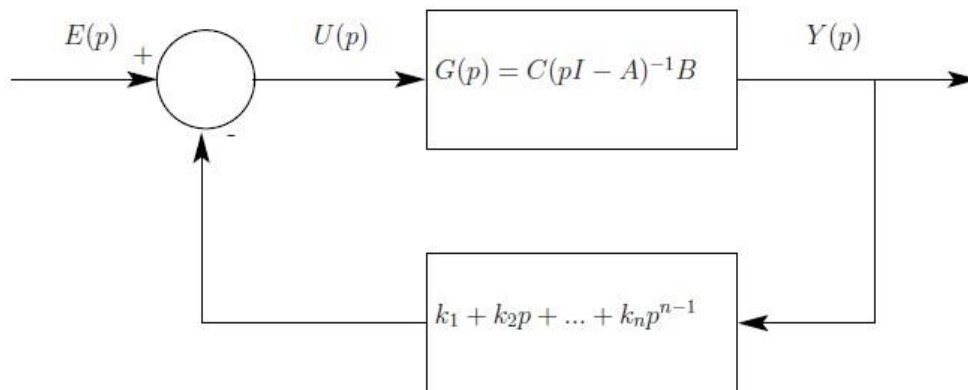


Figure 13 : Retour d'état en utilisant le formalisme de Laplace

Il apparait clairement que le retour d'état est une généralisation de la commande proportionnelle dérivateur. En notant $Q(p) = k_1 + k_2p + \dots + k_n p^{n-1}$, on peut écrire la fonction de transfert du système en boucle fermée entre $Y(p)$ et $E(p)$:

$$Y(p) = \frac{G(p)}{1 + G(p)Q(p)} E(p)$$

Cette fonction de transfert n'a pas un gain statique unitaire ce qui ne permet pas d'obtenir une erreur de position nulle en boucle fermée. Si on analyse le système bouclée en terme de réponse fréquentielle, il apparaît que le système bouclé ne possède pas d'effet intégrale dans la chaîne directe. La solution donc est d'ajouter un intégrateur dans la chaîne directe comme le montre la figure :

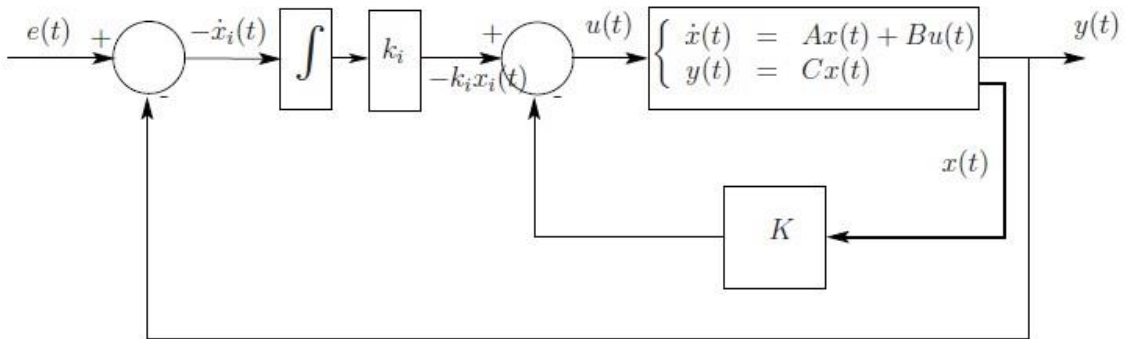


Figure 14 : Commande par retour d'état et action intégrale

Remarquons que pour des raisons d'homogénéité, on note le signal d'erreur $-\dot{x}(t)$ Les équations de la boucle fermée s'écrivent :

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \\ \dot{x}_i(t) = y(t) - e(t) \end{cases}$$

La loi de commande implantée est définie par: $u(t) = -k_i x_i(t) - Kx(t) = -[K \quad k_i] \begin{bmatrix} x(t) \\ x_i(t) \end{bmatrix}$

C'est donc une commande par retour d'état pour le système augmenté suivant:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{x}_i(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ x_i(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ -1 \end{bmatrix} e(t)$$

En posant : $z(t) = \begin{bmatrix} x(t) \\ x_i(t) \end{bmatrix}$, le système de (II.11) s'écrit :

$$\begin{cases} \dot{z}(t) = \hat{A}z(t) + \hat{B}u(t) + \hat{D}e(t) \\ u(t) = -\tilde{K}z(t) \end{cases}$$

$$\text{Où : } \hat{A} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix}, \hat{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}, \tilde{K} = [K \quad k_i].$$

Afin de prouver que la dynamique de l'erreur est stable et tend vers zéro pour des temps

suffisamment grands, il faut choisir \tilde{K} tel que $\hat{A} - \hat{B}\tilde{K}$ soit stable. Par ailleurs, le choix des pôles de $\hat{A} - \hat{B}\tilde{K}$ permet d'imposer la vitesse de convergence de l'erreur vers zéro.

Le problème de la synthèse d'une commande par retour d'état avec action intégrale revient donc au calcul classique d'un retour d'état sur un système augmenté $(\hat{A}, \hat{B}, \hat{C}, 0)$.

3. Application de la commande par retour état

3.1 Commande du système linéarisé par retour d'état :

Dans cette première approche, on va utiliser le modèle linéarisé (III.19) pour établir une loi de commande qu'on implantera sur le modèle non linéaire et on verra les performances de ce type de commande sur le fonctionnement du robot.

3.1.1 Commande par retour d'état avec précompensateur:

Pour réaliser une commande par retour d'état on doit calculer le rang de la matrice de commandabilité $\sigma = [B \ AB \ A^2B \ \dots \ A^{n-1}B]$, qui peut être réalisé par le logiciel MATLAB avec l'instruction : `rank(ctr b(A,B))`, le résultat de calcul est: `ans = 4`. étant égale à l'ordre du système, le système est alors complètement commandable et le placement de pôles est réalisable.

La deuxième étape consiste en la détermination de la matrice des gains K du retour d'état qui assure la stabilité du système (régler la dynamique du système) et pour ce faire, les pôles doivent être choisis à partie réelle strictement négative, par exemple:

$$p = [-7 \quad -7 \quad -10 \quad -10]$$

L'instruction `place(A,B,p)` du logiciel MATLAB donne le résultat suivants :

$K =$

$$30.1017 \quad 5.9703 \quad 4.0737 \quad 0.8404$$

$$4.0737 \quad 0.8404 \quad 1.8881 \quad 0.3096$$

Pour éliminer l'erreur statique (régler le régime permanent), on ajoute la matrice des gains pré compensateur N calculé par la relation (II.10). Le résultat de calcul est :

$$N = \begin{matrix} & 24.5834 & 3.4606 \\ & 3.4606 & 1.2750 \end{matrix}$$

Pour voir l'effet de cette commande, un montage de simulation sur SUMILINK du logiciel MATLAB est réalisé comme le montre la figure suivante :

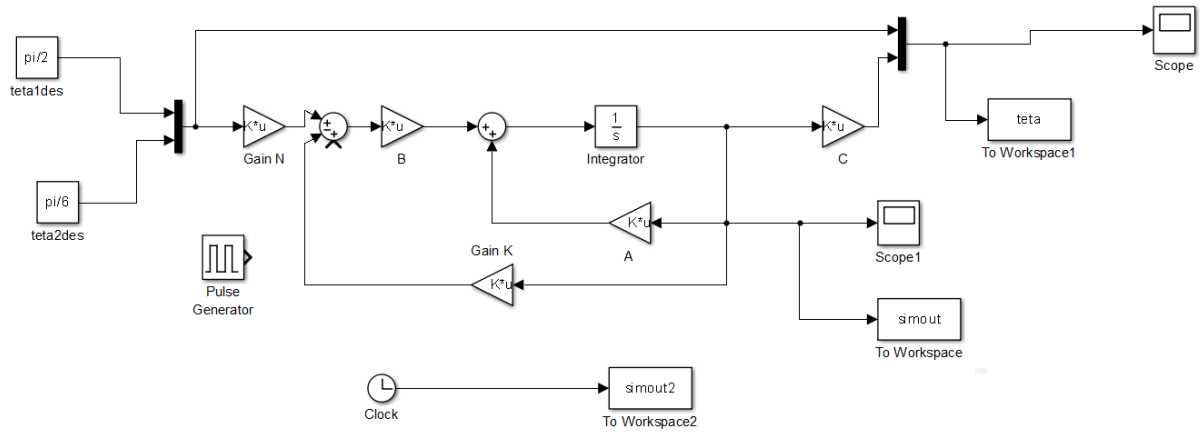


Figure 4: Schéma de la commande par retour d'état avec précompensateur pour le système linéarisé

Le modèle doit avoir en sortie les valeurs désirées suivantes : $\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} \frac{\pi}{2} \\ \frac{\pi}{6} \end{bmatrix}$ qui sont proches du point de fonctionnement. Les résultats de simulation sont satisfaisants même avec la présence d'une perturbation sous forme d'impulsion à $t = 7s$ comme le montre les figures suivantes :

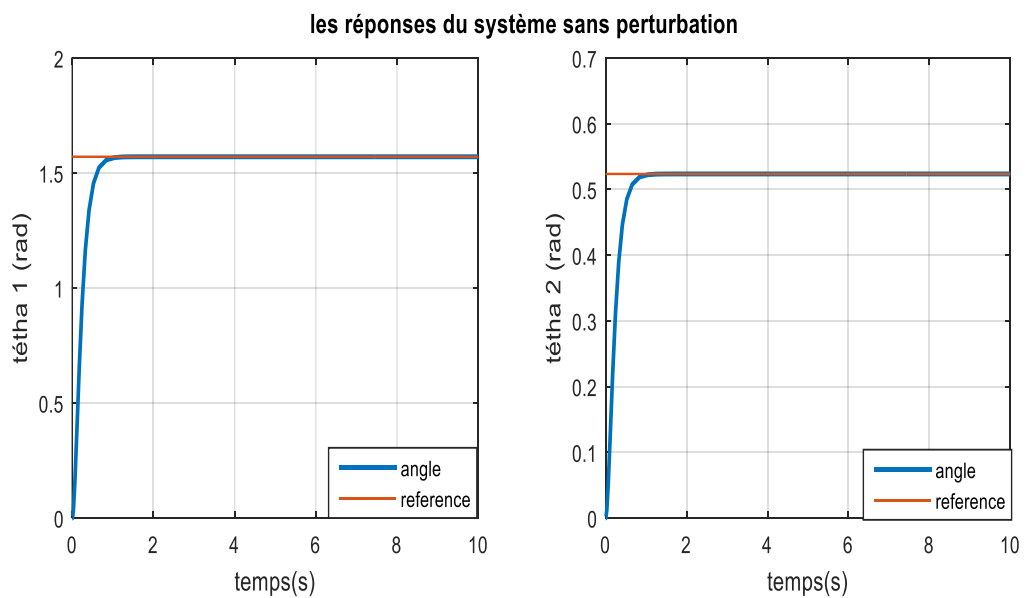


Figure 5: Les 2 angles et leurs valeurs désirées du système linéarisé commandé

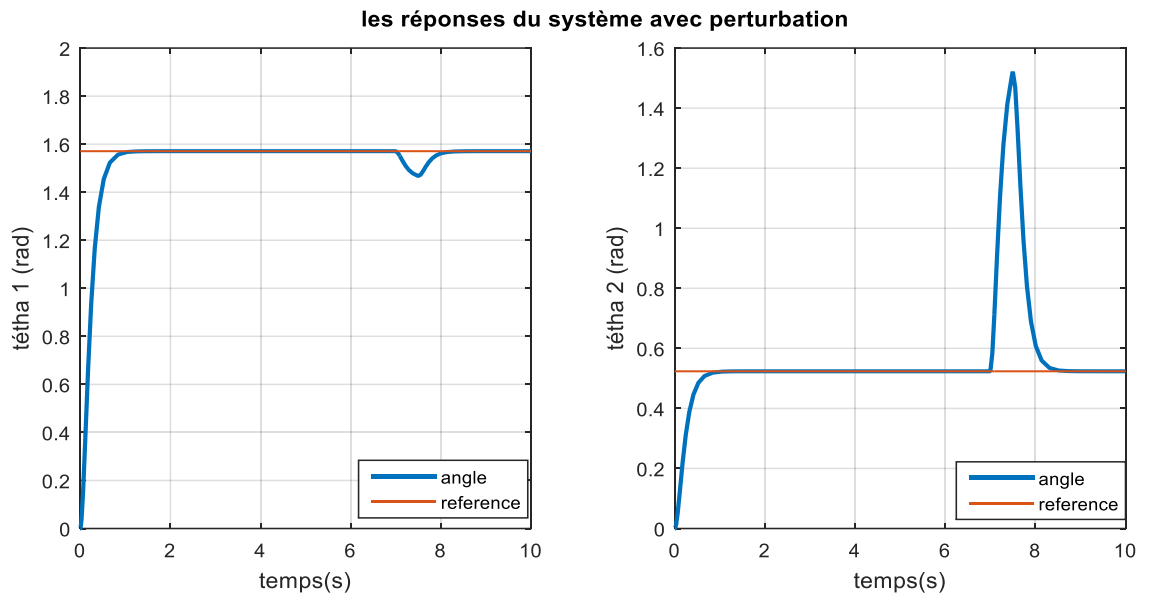


Figure 6: Les 2 angles et leurs valeurs désirées du système linéarisé commandé

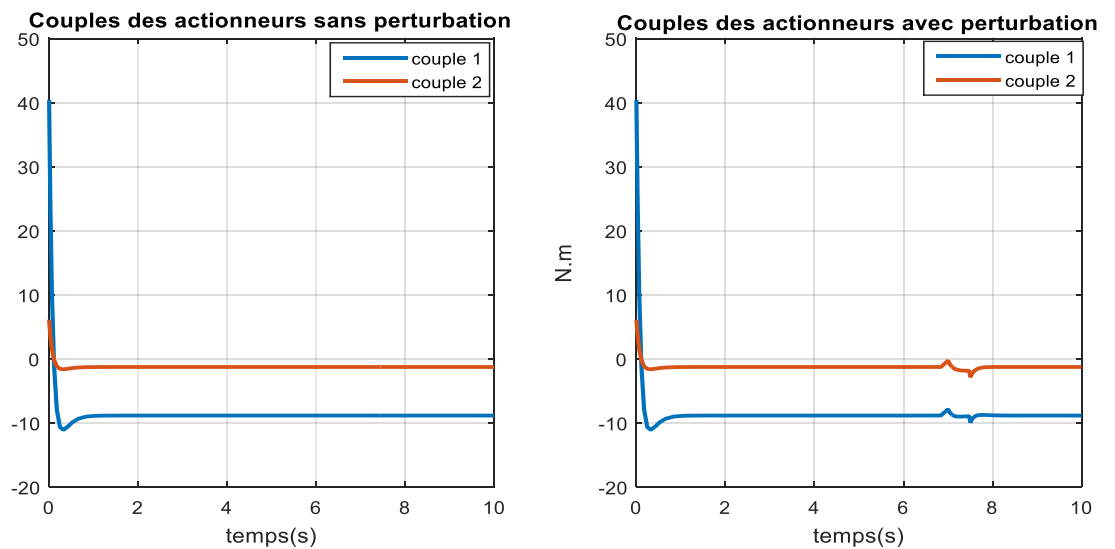


Figure 7: Les couples générés par le retour d'état avec précompensateur

Dans les figures 5 et 6 on voit bien que les valeurs désirées sont atteintes et les perturbations sont rejetés dans un temps court et même les pics provoqués par la perturbation sont réduits d'une façon remarquable.

3.1.2 Commande par retour d'état avec action intégrale :

Pour éliminer l'erreur statique et améliorer la robustesse de la commande on ajoute une action intégrale tel que la loi de commande sera :

$$u = -K(X) - Ki \int (y - w) dt$$

pour le système augmenté suivant :

$$\begin{bmatrix} \dot{x}(t) \\ \dot{x}_i(t) \end{bmatrix} = \hat{A} \begin{bmatrix} x(t) \\ x_i(t) \end{bmatrix} + \hat{B}u(t) + \hat{D}w(t)$$

$$\text{où : } \dot{x}_i(t) = y(t) - w(t); \hat{A} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix}; \hat{B} = \begin{bmatrix} B \\ 0 \end{bmatrix} \text{ et } \hat{D} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

En choisissant les pôles: $p = [-7 - 7 - 2 - 2 - 10 - 10]$; le calcul des gains sur

MATLAB donne :

La matrice des gains du retour d'état: $K = 65.9233 \ 8.0774 \ 9.1162 \ 1.1370$

La matrice des gains de la chaîne d'intégration:

$$K_i = 147.5007 \ 20.7634 \ 20.7634 \ 7.6499$$

$$9.1162 \ 1.1370 \ 3.7459 \ 0.4189$$

3.2 Implantation de la commande dans le modèle non linéaire :

La commande par retour d'état est maintenant établie et est prête à être utilisée pour contrôler le robot. Reste à voir l'effet de cette commande sur le fonctionnement du robot, et pour cela on remplace sur SIMULINK du logiciel MATLAB le modèle linéarisé par le modèle non linéaire comme le montre la figure :

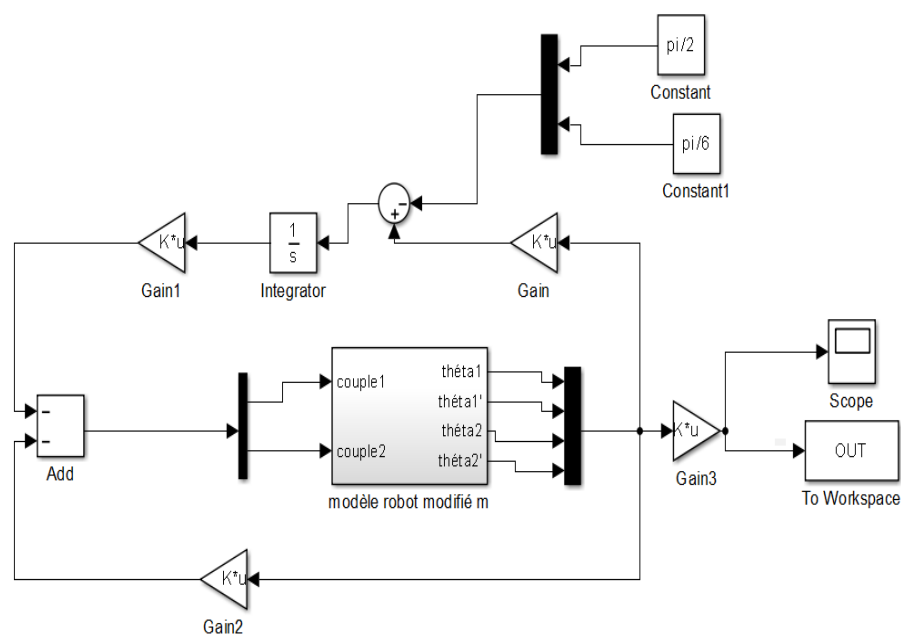


Figure 8: Montage de commande du robot par retour d'état avec action intégrale

Les réponses du robot sans et avec perturbation à $t = 7s$ sont illustrées dans les 3 figures :

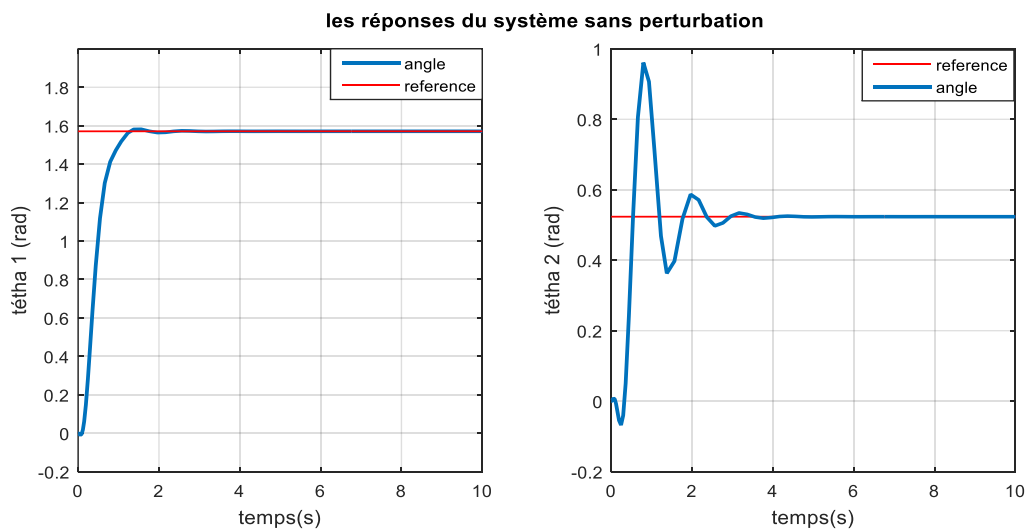


Figure 9: Les 2 positions et leurs valeurs désirées du robot commandé par retour d'état avec action intégrale

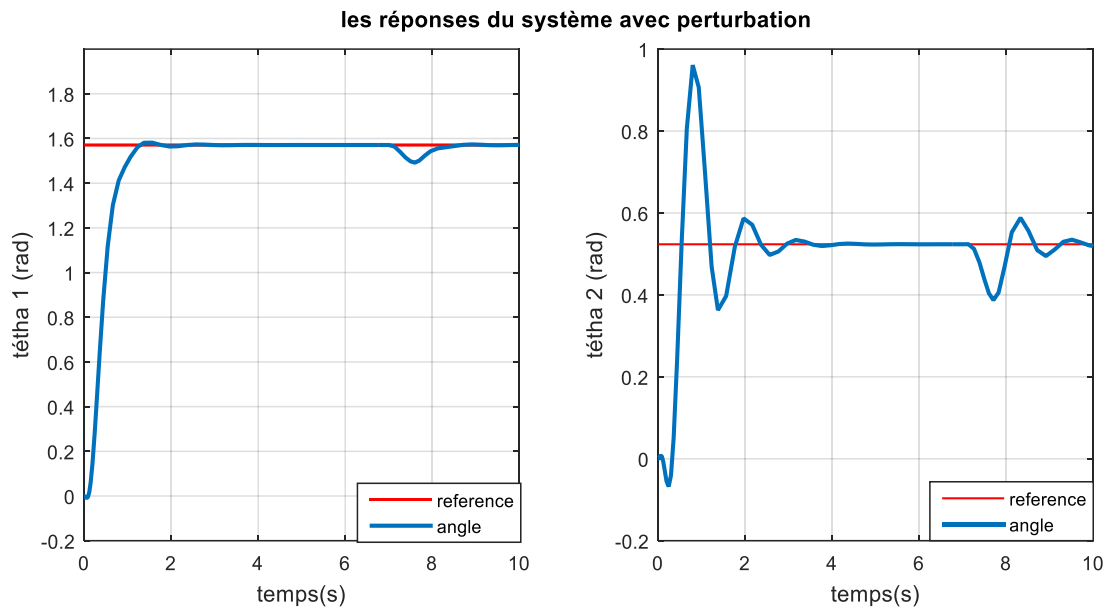


Figure 10: Les 2 positions et leurs valeurs désirées du robot commandé par retour d'état avec action intégrale

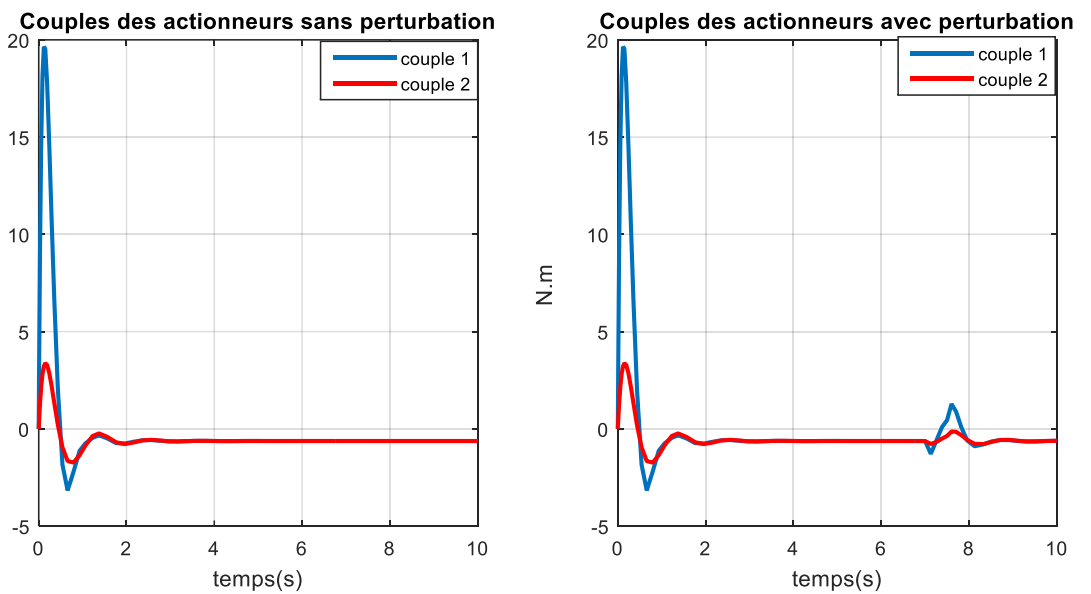


Figure 11: Les couples générés par le retour d'état avec action intégrale

D'après les figures 9 ; 10 et 11, on conclut que la commande adoptée est cohérente, robuste et exacte même avec le modèle non linéaire du robot.

4. Commande par bouclage linéarisant entrée-sortie :

4.1 Systèmes mono-entrée mono-sortie

Soit le système non-linéaire, représenté sous forme affine en la commande :

$$\begin{cases} \dot{x} = f(x) + g(x)u \\ y = h(x) \end{cases}$$

Où $x = [x_1, \dots, x_n]^T \in \mathbb{R}^n$ est le vecteur d'état, $u \in \mathbb{R}$ est le signal de commande, $y \in \mathbb{R}$ est la sortie à contrôler. Les fonctions $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$; $g: \mathbb{R}^n \rightarrow \mathbb{R}^n$ et $h: \mathbb{R}^n \rightarrow \mathbb{R}$ sont analytiques.

L'objectif est d'essayer de synthétiser[12][13]:

- Un retour d'état statique : $u = \alpha(x) + \beta(x)v$, où v une nouvelle entrée de commande externe ;
- Un changement de coordonnées $z = \phi(x)$, où ϕ un difféomorphisme (le Jacobien de $\phi(x)$ n'est singulier : $\det \left(\frac{\partial \phi(x)}{\partial x} \right) \neq 0$); pour transformer un système non linéaire en un système linéaire commandable :

$$\begin{cases} \dot{z} = Az + Bv \\ y = Cz \end{cases}$$

Pendant, afin de faciliter la compréhension, il est préférable de rappeler certaines définitions et montrer les procédures à suivre pour synthétiser la commande.

On définit la dérivée de Lie de la fonction $h(x)$ dans la direction du champ de vecteur $f(x)$:

$$L_f h(x) = \frac{\partial h(x)}{\partial x} f(x) = \left(\frac{\partial h(x)}{\partial x_1} \dots \frac{\partial h(x)}{\partial x_n} \right) \begin{pmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{pmatrix} = \sum_{i=1}^n f_i(x) \frac{\partial h(x)}{\partial x_i}$$

- Le degré relatif r du système par rapport à la sortie y est exactement le nombre de fois qu'il faut dériver la sortie de manière à faire apparaître explicitement la commande u . Par conséquent, on définit de manière récursive:

$$\begin{aligned} L_f^0 h(x) &= h(x) \\ L_f^1 h(x) &= \frac{\partial h(x)}{\partial x} f(x) \dots \\ L_f^i h(x) &= L_f \left(L_f^{i-1} h(x) \right); i = 1, 2, 3, \dots \end{aligned}$$

En effet, si nous dérivons la sortie du système par rapport au temps, nous obtenons:

$$\dot{y} = \frac{\partial h(x)}{\partial x} \dot{x} = \frac{\partial h(x)}{\partial x} (f(x) + g(x)u) = L_f h(x) + L_g h(x)u$$

Lorsque le degré relatif du système est supérieur à 1, nous avons : $L_g h(x) = 0$ et $\dot{y} = L_f h(x)$ La deuxième dérivée de y conduit à :

$$\ddot{y} = \frac{\partial L_f h(x)}{\partial x} \dot{x} = \frac{\partial L_f h(x)}{\partial x} (f(x) + g(x)u) = L_f^2 h(x) + L_g L_f h(x)u$$

Encore une fois, si le degré relatif est supérieur à deux, alors: $L_g L_f h(x) = 0$ et $\ddot{y} = L_f^2 h(x)$

En étendant le raisonnement, on obtient :

$$L_g L_f^i h(x) = 0; \text{ pour } 0 \leq i < r - 1$$

$$L_g L_f^{r-1} h(x) \neq 0$$

La dérivée de la sortie qui fait apparaître la commande est :

$$y^{(r)} = L_f^r h(x) + L_g L_f^{r-1} h(x)u = \Delta_0(x) + \Delta(x)u$$

On introduit une entrée externe telle que :

$$y^{(r)} = v$$

Le retour d'état linéarisant permettant de compenser les non linéarités et d'établir une relation linéaire entre l'entrée v et la sortie y est:

$$u = \alpha(x) + \beta(x)v = \frac{1}{L_g L_f^{r-1} h(x)} (-L_f^r h(x) + v); L_g L_f^{r-1} h(x) \neq 0$$

$$\text{avec : } \alpha(x) = -\frac{\Delta_0(x)}{\Delta(x)} = -\frac{L_f^r h(x)}{L_g L_f^{r-1} h(x)}; \beta(x) = \frac{1}{\Delta(x)} = \frac{1}{L_g L_f^{r-1} h(x)}$$

Le comportement entré-sortie du système bouclé est linéaire décrit par une chaîne d'intégrateurs :

$$\frac{Y(p)}{V(p)} = \frac{1}{s^r}$$

Il existe également un changement de coordonnées $z = \phi(x)$ tel que dans le nouveau système de coordonnées, le système bouclé s'écrit partiellement ou complètement sous forme canonique. Deux cas sont à envisager:

- 1^{er} cas : $r = n$; n étant la dimension de l'espace d'état, le changement de coordonnées est complètement défini par

$$z = \begin{pmatrix} z_1 \\ \vdots \\ z_n \end{pmatrix} = \begin{pmatrix} y \\ \vdots \\ y^{(r-1)} \end{pmatrix} = \begin{pmatrix} \phi_1(x) \\ \vdots \\ \phi_n(x) \end{pmatrix} = \begin{pmatrix} h(x) \\ \vdots \\ L_f^{r-1}h(x) \end{pmatrix}$$

Forme canonique commandable et observable :

$$\begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = z_3 \\ \dots \\ \dot{z}_{n-1} = z_n \\ \dot{z}_n = v \end{cases} \Rightarrow \begin{cases} \dot{z} = Az + Bv \\ y = Cz \end{cases}$$

Avec:

$$A(r \times r) = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 1 \\ 0 & \dots & \dots & \dots & 0 \end{pmatrix}; B(r \times 1) = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}; C(1 \times r) = (1 \quad 0 \quad \dots \quad 0)$$

- 2^{ème} cas: $r < n$; il est toujours possible de compléter l'état du système par les $n - r$ fonctions supplémentaires $\phi_{r+1}(x), \dots, \phi_n(x)$; afin de pouvoir définir un changement de coordonnées $z = \phi(x) = [\phi_1(x), \dots, \phi_r(x), \phi_{r+1}(x), \dots, \phi_n(x)]^T$. Les fonctions

$\phi_{r+1}(x), \dots, \phi_n(x)$ peuvent être fixées arbitrairement, mais il est toujours possible de choisir ces fonctions de la façon suivante :

$$\begin{cases} \dot{z}_{r+1} = \frac{\partial \phi_{r+1}(x)}{\partial x} (f(x) + g(x)u) = L_f \phi_{r+1}(x) + \underbrace{L_g \phi_{r+1}(x)}_{=0} u \\ \dot{z}_n = \frac{\partial \phi_n(x)}{\partial x} (f(x) + g(x)u) = L_f \phi_n(x) + \underbrace{L_g \phi_n(x)}_{=0} u \end{cases}$$

En utilisant les notations suivantes:

$$\xi = \begin{pmatrix} z_1 \\ \vdots \\ z_r \end{pmatrix}; \eta = \begin{pmatrix} z_{r+1} \\ \vdots \\ z_n \end{pmatrix}; z = \begin{pmatrix} \xi \\ \eta \end{pmatrix}; \psi(\xi, \eta) = \begin{pmatrix} L_f \phi_{r+1}(x) \\ \vdots \\ L_f \phi_n(x) \end{pmatrix} = \begin{pmatrix} L_f \phi_{r+1}(\phi^{-1}(z)) \\ \vdots \\ L_f \phi_n(\phi^{-1}(z)) \end{pmatrix}$$

Après transformation et bouclage, le système peut s'écrire sous la forme condensée suivante:

$$\begin{cases} \dot{\xi} = A\xi + Bv \\ \eta = \psi(\xi, \eta) \\ y = Cz \end{cases}$$

Dans ce cas, le système bouclé possède bien un comportement entrées-sorties linéaire mais l'inconvénient de rendre certaines dynamiques inobservables. Le système bouclé, dans les nouvelles coordonnées, est constitué d'un système linéaire sous forme canonique et d'un système non linéaire. Pour garantir la stabilité, il ne suffit pas de regarder le comportement entrée-sortie du système : on doit aussi vérifier que les états rendus inobservables sont stables. Pour cela, la dynamique des zéros doit être analysée. Elle est donnée par $S = \mathcal{f}(\xi, 5)$ en supposant les sorties et toutes les dérivées des sorties nulles. La dynamique des zéros correspond au comportement interne du système.[12][13]

Si on suppose que l'intérêt de la commande est de maintenir la sortie du système égale à zéro $y = 0 \Rightarrow \xi = 0 \Rightarrow \dot{\xi} = 0$.

La dynamique des zéros du système non linéaire est donnée par :

$$\dot{\eta} = \psi(0, \eta)$$

Si la dynamique des zéros est stable, il existe une loi de commande v qui stabilise le système.

Figure 1.1 : Schéma de commande par linéarisation entrée-sortie.

Le problème de synthèse de la commande v qui permet la poursuite de trajectoires de référence ou le suivi de consigne y_d peut être traité comme un problème linéaire :

Soit $e = y - y_d$ l'erreur de poursuite. On définit $\varepsilon_1 = e$; $\varepsilon_2 = \dot{e} \dots$; $\varepsilon_r = e^{(r-1)}$. On obtient:

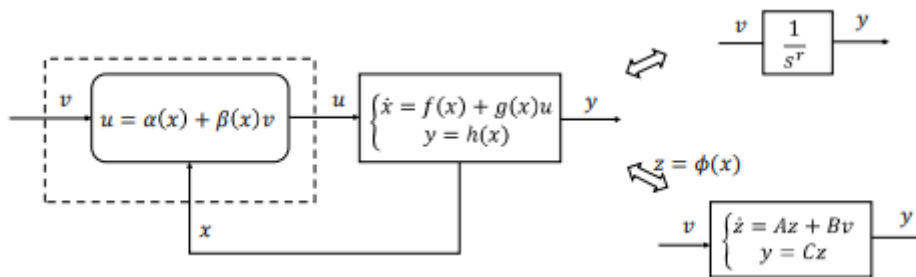


Figure 12: Schéma de commande par linéarisation entrée-sortie.

Le problème de synthèse de la commande v qui permet la poursuite de trajectoires de référence ou le suivi de consigne y_d peut être traité comme un problème linéaire :

Soit $e = y - y_d$ l'erreur de poursuite. On définit $s_1 = e$; $s_2 = \dot{e} \dots$; $s_r = e^{(r-1)}$.

On obtient :

$$\left\{ \begin{array}{l} \dot{\varepsilon}_1 = \varepsilon_2 \\ \dot{\varepsilon}_2 = \varepsilon_3 \\ \vdots \\ \dot{\varepsilon}_{r-1} = \varepsilon_r \\ \dot{\varepsilon}_r = v - y_d^{(r)} = \bar{v} \end{array} \right. \Rightarrow \{\varepsilon = A\varepsilon + B\bar{v} \Rightarrow$$

Avec la commande v doit être choisie de telle sorte que la stabilité du système soit garantie, la commande v sera choisie telle que :

$$\bar{v} = -K\varepsilon \Rightarrow v = -K\varepsilon + y_d^{(r)}$$

On obtient $\dot{\varepsilon} = (A - BK)\varepsilon$; alors $\varepsilon(t) = e^{(A-BK)t}\varepsilon(0)$. La convergence vers zéro de l'erreur de poursuite est garantie si $\text{Re}(vp(A - BK)) < 0$. Alors:

$$v = -K \begin{pmatrix} y - y_d \\ \dot{y} - \dot{y}_d \\ \dots \\ y^{(r-1)} - y_d^{(r-1)} \end{pmatrix} + y_d^{(r)}$$

$$v = -k_1(y - y_d) - k_2(\dot{y} - \dot{y}_d) - \dots - k_r(y^{(r-1)} - y_d^{(r-1)}) + y_d^{(r)}$$

Le vecteur $K = [k_1 k_2 \dots k_r]$ (gain de la commande) est choisi tel que les valeurs propres de la matrice $(A - BK)$ sont à partie réelle strictement négative. Ou le polynôme caractéristique $P(s) = s^r + k_r s^{r-1} + \dots + k_2 s + k_1$ soit d'Hurwitz.

4.2 Systèmes multi-entrées multi-sorties

On considère un système non linéaire de la forme suivante[12]:

$$\left\{ \begin{array}{l} \dot{x} = f(x) + g(x)u = f(x) + \sum_{i=1}^m g_i(x)u_i \\ y = h(x) \end{array} \right.$$

Où $x = [x_1, \dots, x_n]^T \in \mathbb{R}^n$ est le vecteur des états, $u = [u_1, \dots, u_m]^T \in \mathbb{R}^m$ est vecteur des commandes, $y = [y_1, \dots, y_p]^T \in \mathbb{R}^p$ est le vecteur des sorties à contrôler. Les fonctions $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$; $g_i: \mathbb{R}^n \rightarrow \mathbb{R}^n$ et $h: \mathbb{R}^n \rightarrow \mathbb{R}^p$ sont analytiques. On prend le nombre de sorties égal au nombre d'entrées $p = m$.

Le problème consiste à trouver :

Une loi de commande par retour d'état statique: $u = \alpha(x) + \beta(x)v$, où v une nouvelle entrée de commande externe; et un changement de coordonnées $z = \phi(x)$, tel que le système bouclé soit découplé et linéaire.

Le système découplé se présente comme un ensemble de p sous systèmes mono-entrée monosortie, dans le sens où chaque entrée v_j n'affecte que la sortie y_j ; $j = 1, 2, \dots, p$. Soit r_j le degré relatif du système par rapport à la sortie y_j , c'est le nombre de fois qu'il faut dériver la j^{mn} sortie de manière à faire apparaître explicitement au moins une composante du vecteur de commande u .

$$\begin{aligned} \dot{y}_j &= L_f h_j(x) \\ \ddot{y}_j &= L_f^2 h_j(x), \dots \\ y_j^{(r_j)} &= L_f^{r_j} h_j(x) + L_g L_f^{r_j-1} h_j(x) u = L_f^{r_j} h_j(x) + \sum_{i=1}^m L_{g_i} L_f^{r_j-1} h_j(x) u_i \end{aligned}$$

Ecriture sous forme matricielle, pour chaque sortie y_j ; $j = 1, 2, \dots, m$.

$$\begin{pmatrix} y_1^{(r_1)} \\ \vdots \\ y_m^{(r_m)} \end{pmatrix} = \underbrace{\begin{pmatrix} L_f^{r_1} h_1(x) \\ \vdots \\ L_f^{r_m} h_m(x) \end{pmatrix}}_{=\Delta_0(x)} + \underbrace{\begin{pmatrix} L_{g_1} L_f^{r_1-1} h_1(x) & \dots & L_{g_m} L_f^{r_1-1} h_1(x) \\ \vdots & \dots & \vdots \\ L_{g_1} L_f^{r_m-1} h_m(x) & \dots & L_{g_m} L_f^{r_m-1} h_m(x) \end{pmatrix}}_{=\Delta(x)} \begin{pmatrix} u_1 \\ \vdots \\ u_m \end{pmatrix}$$

La matrice $\Delta(x)$ est appelée matrice de découplage. Si cette matrice est inversible ($\det(\Delta(x)) \neq 0$), alors le système est découplable par retour d'état statique :

$$u = \alpha(x) + \beta(x)v = \Delta^{-1}(x) \cdot (-\Delta_0(x) + v) ; \alpha(x) = -\Delta^{-1}(x) \cdot \Delta_0(x) ; \beta(x) = \Delta^{-1}(x)$$

Le comportement entrée-sortie du système bouclé est linéaire décrit par : $y_j^{(r_j)} = v_j$; $j = 1, 2, \dots, m$. Le comportement dynamique du système découplé est composé de chaînes d'intégrateurs en cascade n'est pas admissible. Il est donc nécessaire de le modifier par un second bouclage linéaire v_j .

Il existe également un changement de coordonnées $z = \phi(x)$ tel que dans le nouveau système de coordonnées, le système bouclé s'écrive partiellement ou complètement sous forme canonique

Le degré relatif global du système est $r = \sum_{i=1}^m r_i$. Deux cas à envisager :

Si $r = n$, le changement de coordonnées est complètement défini par :

$$z = \phi(x) = \begin{pmatrix} z_1 = \phi_1(x) \\ \vdots \\ z_m = \phi_m(x) \end{pmatrix} ; \underbrace{z_j = \phi_j(x)}_{j=1,2,\dots,m} = \begin{pmatrix} z_{j1} = y_j = h_j(x) = \phi_{j1}(x) \\ z_{j2} = \dot{y}_j = L_f h_j(x) = \phi_{j2}(x) \\ \vdots \\ z_{jr_j} = y_j^{(r_j-1)} = L_f^{r_j-1} h_j(x) = \phi_{jr_j}(x) \end{pmatrix}$$

Forme canonique :

$$\begin{cases} \dot{z}_{j1} = z_{j2} \\ \dot{z}_{j2} = z_{j3} \\ \dots \\ \dot{z}_{j(r_j-1)} = z_{jr_j} \\ \dot{z}_{jr_j} = v_j \end{cases} \Rightarrow \begin{cases} \dot{z}_j = A_j z_j + B_j v_j \\ y_j = C_j z_j \end{cases}$$

Avec:

$$A_j(r_j \times r_j) = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 1 \\ 0 & \dots & \dots & \dots & 0 \end{pmatrix}; B_j(r_j \times 1) = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}; C_j(1 \times r_j) = (1 \quad 0 \quad \dots \quad 0 \quad 0)$$

Sous forme condensée:

$$\begin{cases} \dot{z} = Az + Bv \\ y = Cz \end{cases}$$

avec : $A = \text{diag}(A_1, A_2, \dots, A_m)$; $B = \text{diag}(B_1, B_2, \dots, B_m)$; $C = \text{diag}(C_1, C_2, \dots, C_m)$

Si $r < n$, le changement de coordonnées doit être complété par $n - r$ fonctions supplémentaires, $\phi_{r+1}(x), \dots, \phi_n(x)$ telles que,

$$\phi(x) = [\phi_{11}(x), \dots, \phi_{1r_1}(x), \dots, \phi_{m1}(x), \dots, \phi_{mr_m}(x), \phi_{r+1}(x), \dots, \phi_n(x)]^T$$

Soit un difféomorphisme. Le calcul des fonctions $\eta = [\phi_{r+1}(x), \dots, \phi_n(x)]^T$ est tel que :

$$\begin{cases} \dot{z}_{r+1} = L_f \phi_{r+1}(x) + \underbrace{L_g \phi_{r+1}(x) u}_{=0} \\ \vdots \\ \dot{z}_n = L_f \phi_n(x) + \underbrace{L_g \phi_n(x) u}_{=0} \end{cases}$$

Avec les notations suivantes:

$$\xi = \begin{pmatrix} z_1 \\ \vdots \\ z_m \end{pmatrix} \in \mathbb{R}^r; \eta = \begin{pmatrix} z_{r+1} \\ \vdots \\ z_n \end{pmatrix} \in \mathbb{R}^{n-r}; z = \begin{pmatrix} \xi \\ \eta \end{pmatrix} = \phi(x) \in \mathbb{R}^n; \psi(\xi, \eta) = \begin{pmatrix} L_f \phi_{r+1}(x) \\ \vdots \\ L_f \phi_n(x) \end{pmatrix}$$

On obtient:

$$\begin{cases} \dot{\xi} = Az + Bv \\ \dot{\eta} = \psi(\xi, \eta) \\ y = Cz \end{cases}$$

Si la dynamique des zéros $\dot{\eta} = \psi(0, \eta)$ est stable, alors le problème de poursuite de trajectoire désirée

$y_d = [y_{1d}, \dots, y_{pd}]^T$ est résolu par les lois de commande linéaire :

$$v_j = -K_j \begin{pmatrix} y_j - y_{jd} \\ \dot{y}_j - \dot{y}_{ja} \\ \dots \\ y_j^{(r_j-1)} - y_{jd}^{(r_j-1)} \end{pmatrix} + y_{ja}^{(r_j)} \\ v = -k_{j1} (y_j - y_{jd}) - k_{j2} (\dot{y}_j - \dot{y}_{ja}) - \dots - k_{jr_j} (y_j^{(r_j-1)} - y_{jd}^{(r_j-1)}) + y_{jd}^{(r_j)}$$

Le vecteur K_j est tel que $\text{Re} (vp(A_j - B_jK_j)) < 0$

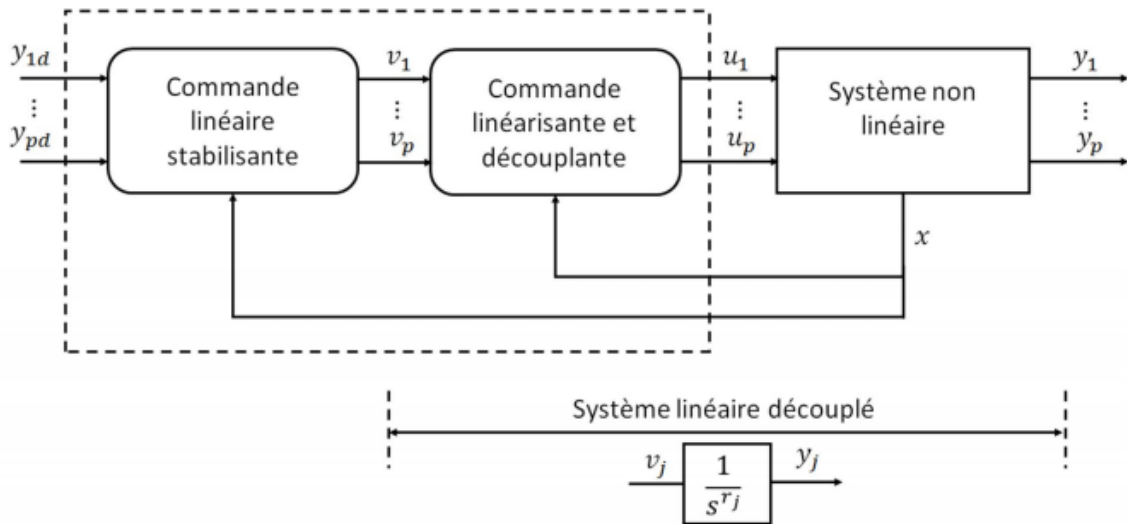


Figure 13: Architecture de la commande par linéarisation et découplage entrées-sorties[12]

4.3 Application Commande par bouclage linéarisant entrée-sortie :

4.3.1 Introduction:

La commande par bouclage linéarisant est une méthode de commande qui se ramène au problème de commander un système composé de plusieurs chaînes d'intégrateurs découplées les unes des autres.

On rappelle que d'après (III.18), le modèle non linéaire du robot peut être représenté en espace d'état sous la forme :

$$\begin{cases} \dot{x} = F(x) + G(x)u \\ y = Cx \end{cases}$$

Ou :

$$F(x) = \begin{bmatrix} f1 \\ f2 \\ f3 \\ f4 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{-M_{21}C_1 + M_{12}C_2}{\det(M)} \\ x_2 \\ \frac{M_{21}C_1 - M_{11}C_2}{\det(M)} \end{bmatrix}$$

$$G(x) = [g1 \ g2] = \begin{bmatrix} 0 & 0 \\ \frac{M_{22}}{\det(M)} & \frac{-M_{12}}{\det(M)} \\ 0 & 0 \\ \frac{-M_{21}}{\det(M)} & \frac{M_{11}}{\det(M)} \end{bmatrix} \text{ et } h(x) = \begin{cases} x_1 \\ x_3 \end{cases}$$

4.3.2 Les degrés relatifs:

Le degré relatif r d'une sortie est le nombre de fois qu'il faut dériver cette sortie pour faire apparaître

l'entrée. En utilisant la dérivée de Lie on a : $y_j^{(r_j)} = L_f^{r_j} h_j + \sum (L_{g_i} L_f^{r_j-1} h_j) u_i$

-) Pour la première sortie $y_1 = h_1$:

$$\dot{y}_1 = L_f h_1 + L_{g_1} h_1 u_1 + L_{g_2} h_1 u_2 = f_1$$

$$L_f h_1 = \frac{\partial h_1}{\partial x} f = [1 \ 0 \ 0 \ 0] f = f_1 = x_2; L_{g_1} h_1 = 0; L_{g_2} h_1 = 0$$

$$\ddot{y}_1 = L_f^2 h_1 + L_{g_1} L_f h_1 u_1 + L_{g_2} L_f h_1 u_2 = f_2 + g_{21} u_1 + g_{22} u_2$$

$$L_f^2 h_1 = \frac{\partial(L_f h_1)}{\partial x} f = \frac{\partial x_2}{\partial x} f = [0 \ 1 \ 0 \ 0] f = f_2; L_{g_1} L_f h_1 = g_{21} \neq 0; L_{g_2} L_f h_1 = g_{22} \neq 0$$

-) Pour la deuxième sortie $y_2 = h_2$:

$$\dot{y}_2 = L_f h_2 + L_{g_1} h_2 u_1 + L_{g_2} h_2 u_2 = f_3$$

$$L_f h_2 = \frac{\partial h_2}{\partial x} f = [0 \ 0 \ 1 \ 0] f = f_3 = x_4; L_{g_1} h_2 = 0; L_{g_2} h_2 = 0$$

$$\ddot{y}_2 = L_f^2 h_2 + L_{g_1} L_f h_2 u_1 + L_{g_2} L_f h_2 u_2 = f_4 + g_{41} u_1 + g_{42} u_2$$

$$L_f^2 h_2 = \frac{\partial(L_f h_2)}{\partial x} f = \frac{\partial x_4}{\partial x} f = [0 \ 0 \ 0 \ 1] f = f_4; L_{g_1} L_f h_2 = g_{41} \neq 0; L_{g_2} L_f h_2 = g_{42} \neq 0.$$

-) Conclusion : le degré relatif de la première et la deuxième sortie sont $r_1 = r_2 = 2$, donc le degré relatif global $r = r_1 + r_2 = 4$ étant égale à l'ordre du système en déduit que le système est exactement linéarisable par: retour d'état non linéaire et difféomorphisme. (III.20)

4.3.3 Calcul des entrées du système (les couples du robot):

D'après (II.15) et les dérivées de Lie calculées précédemment on trouve:

$$\begin{aligned} \begin{bmatrix} \ddot{y}_1 \\ \ddot{y}_2 \end{bmatrix} &= \begin{bmatrix} f_2 \\ f_4 \end{bmatrix} + M^{-1} \cdot \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\ \ddot{y} &= A(x) + D(x)u \end{aligned} \quad (\text{III.21}) \quad (\text{III.22})$$

Qui est de la forme :

Si on considère un nouveau vecteur d'entrées μ définit par $\mu = A(x) + D(x)u$

D'où :

$$u = D^{-1}(x)(-A(x) + \mu) \quad (\text{III.23})$$

On aboutit d'après (III.22) et (III.23) à : $\ddot{y} = \mu$ (III.24)

l'égalité (III.24) montre bien que notre système est devenu linéaire et découplé de point de vue entrée/sortie, et équivaut à une chaîne de 2 intégrateurs. Mais tous ça dans la réserve que la matrice $D(x) = M^{-1}$ est réversible (c'est le cas de notre système car: $\det(M) \neq 0$).

La commande par retour d'état non linéaire est définit d'après (III.21) et (III.24) par:

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = M \cdot \begin{bmatrix} \mu_1 - f_2 \\ \mu_2 - f_4 \end{bmatrix} \quad (\text{III.25})$$

En remplaçant f_2, f_4 et la matrice M dans la relation (III.25) on trouve :

$$\begin{cases} u_1 = C_1 + M_{11}\mu_1 + M_{12}\mu_2 \\ u_2 = C_2 + M_{21}\mu_1 + M_{22}\mu_2 \end{cases} \quad (\text{III.26})$$

4.3.4 La dynamique de l'erreur de suivi des trajectoires de référence:

On considère le vecteur des positions de référence w et donc le vecteur des erreurs de suivie des trajectoires est: $w - y = e$. Pour assurer le suivi des trajectoires de référence et donc l'annulation de l'erreur, on décrit Le nouveau vecteur d'entrées μ par :

$$\mu = \alpha_0(w - y) + \alpha_1(\dot{w} - \dot{y}) + \ddot{w}$$

En tenant compte de l'égalité (II.24) et en remplaçant dans (III.27) on aboutira à l'équation différentielle de l'erreur : $\ddot{e} + \alpha_1 \dot{e} + \alpha_0 e = 0$ avec α_1, α_0 sont les coefficients du polynôme caractéristique qui peuvent être fixés en choisissant des pôles stabilisants. En choisissant les pôles : $p = [-7 -7]$ on trouve les coefficients : $\alpha_0 = 49, \alpha_1 = 14$. Sur la figure suivante, il est illustré le montage sur SUMILINK de commande par bouclage linéarisant entrée-sortie:

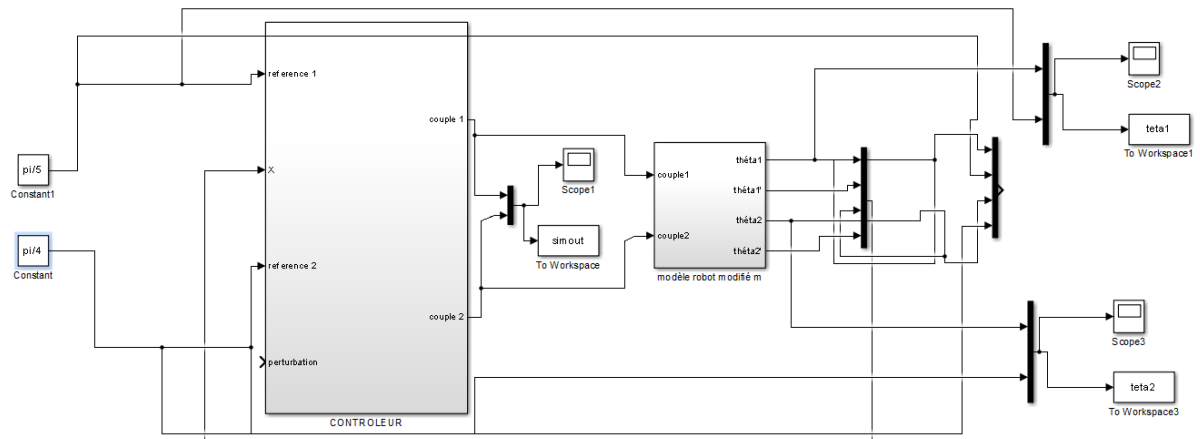


Figure 14: Montage de commande du robot par bouclage linéarisant entrée-sortie

Les résultats de simulation pour les valeurs de référence $\theta_{1ref} = \frac{\pi}{8}$ et $\theta_{2ref} = \frac{\pi}{5}$ ref sont satisfaisants comme l'indiquent ces figures :

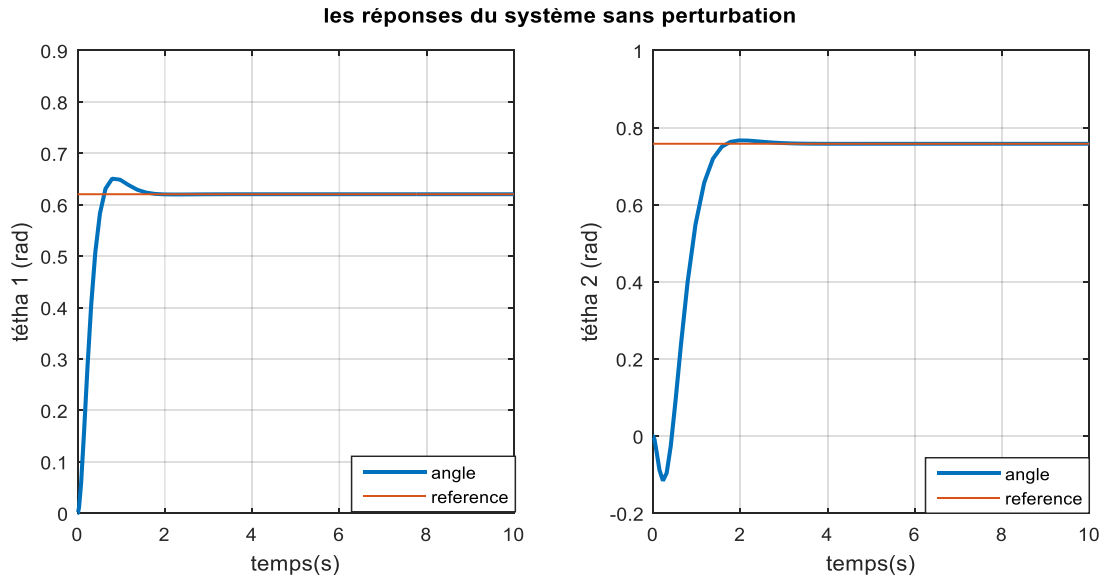


Figure 15: Les 2 positions et leurs valeurs désirées du robot commandé par bouclage linéarisant sans présence de perturbation

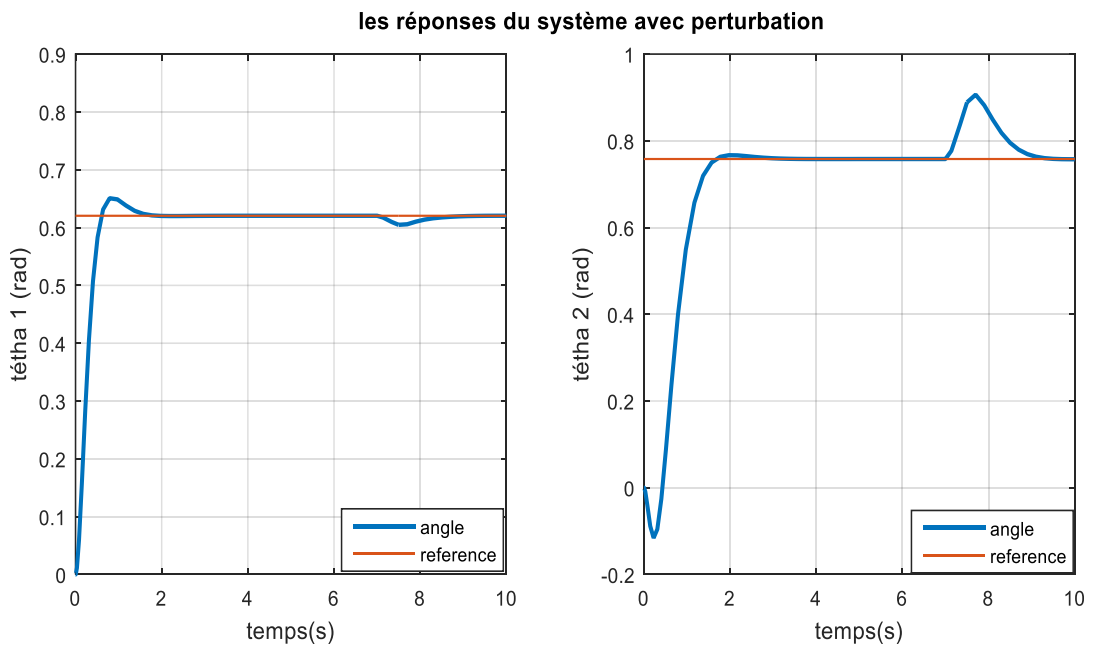


Figure 16: Les 2 positions et leurs valeurs désirées du robot commandé par bouclage linéarisant en présence de perturbation

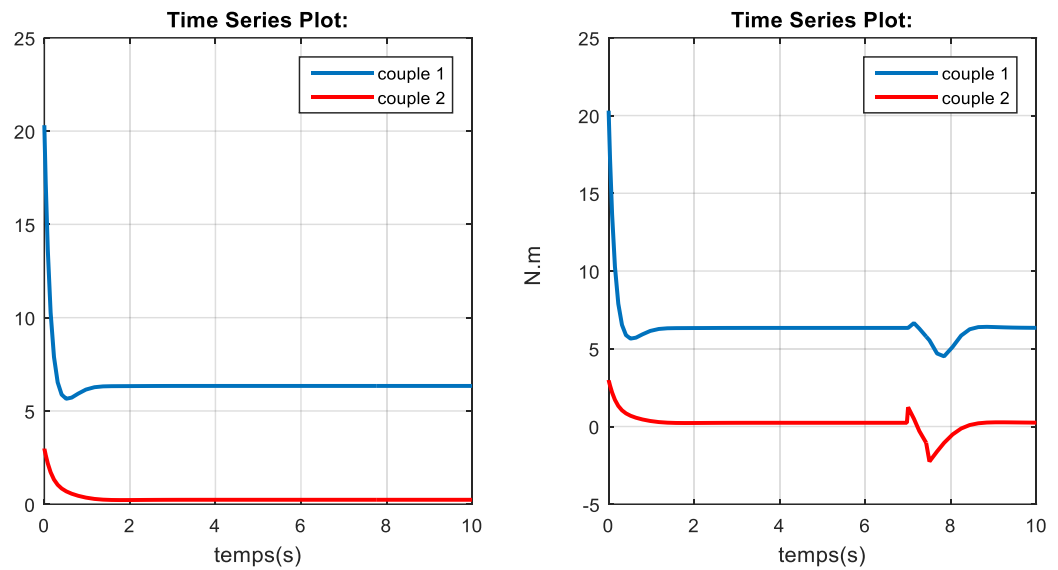


Figure 17: Les couples générés pour les 2 actionneurs du robot commandé par bouclage linéarisant

Les résultats de simulation montrent que la commande du robot par bouclage linéarisant entrée-sortie est parfaite, en effet le suivi de la trajectoire désirée est assuré à n'importe quel point et non pas au voisinage d'un point d'équilibre, en plus la commande garantit la stabilité de fonctionnement du système, même dans le cas de présence des perturbations, elles sont rapidement rejetées.

5. Conclusion

Ce chapitre nous a permis de présenter une étude théorique des lois de commande non linéaire, on a fait deux commandes classiques, telle que la commande par retour d'état et la commande par bouclage linéarisant.

Ensuite pour valider ces deux commandes nous l'avons appliqué sur un bras manipulateur à deux DDL, les résultats obtenus sont encourageants.

CHAPITRE IV

COMMANDE PAR BACKSTEPPING

1. Introduction :

La technique de backstepping est une méthode systématique et récursive de synthèse de lois de commande non linéaires à partir des fonctions de Lyapunov qui assurent pas à pas la stabilisation de chaque étape de synthèse. A chaque étape du processus, une commande virtuelle est ainsi générée pour assurer la convergence du système vers son état d'équilibre. Cette technique permet la synthèse des lois de commande en tenant compte éventuellement des perturbations ou de la méconnaissance des paramètres du système. L'idée de base de la commande par backstepping est de rendre les systèmes bouclés équivalents à des sous-systèmes d'ordre un en cascade stable au sens de Lyapunov, ce qui lui confère des qualités de robustesse et une stabilité globale asymptotique.

De plus, et contrairement à la méthode de linéarisation entrée-sortie qui exige des modèles définis et compense souvent des non-linéarités utiles, la méthode backstepping offre un choix d'outils de synthèse permettant de s'accommoder d'incertitudes et peut éviter des éliminations des non-linéarités utiles pour la performance et la robustesse de la commande.

2. Théorie du Backstepping

2.1. Stabilité au sens de Lyapunov

Notions fondamentales de la stabilité :

La notion de stabilité d'un système dynamique constitue une problématique centrale de la théorie du contrôle. Elle caractérise l'évolution de sa trajectoire d'état initialisée autour d'un point d'équilibre[12][14].

On considère un système non linéaire autonome (libre) de la forme suivante :

$$\dot{x} = f(x) \quad (1)$$

où $x \in \mathbb{R}^n$; $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ est continue.

On désigne par $x(t, t_0, x_0)$ la solution de l'équation différentielle (2.1) à l'instant $t \geq t_0$ initialisée en $x(t_0) = x_0$.

Point d'équilibre :

Physiquement, un système est en équilibre lorsqu'il conserve son état en absence de forces

externes. Mathématiquement, cela équivaut à dire que la dérivée de son vecteur d'état est nulle. L'état x_e est un point d'équilibre pour le système (2.1) si $f(x_e) = 0$; on suppose dans la suite que $x_e = 0$ (qui peut être l'erreur d'asservissement ou d'estimation d'état).

Définition 2.1 : (Notion intuitive de la stabilité)

Si toute évolution d'un système issu d'un voisinage suffisamment petit d'un point d'équilibre x_e demeure au voisinage de ce point, alors x_e est dit stable au sens de Lyapunov[12].

Définition 2.2 : (Stabilité au sens de Lyapunov)

Le point d'équilibre $x_e = 0$ est stable au sens de Lyapunov, si pour tout $R > 0$, il existe $r > 0$ tel que :

$$\|x_0\| < r \implies \|x(t, t_0, x_0)\| < R ; \forall t \geq t_0$$

Dans le cas contraire, on dit que l'équilibre est instable.

En d'autres termes, pour tout $t \geq t_0$, une petite perturbation de la condition initiale x_0 autour de l'origine $x_e = 0$, donne naissance à une solution $x(t, t_0, x_0)$ qui reste proche de l'origine. La stabilité du système n'implique pas la convergence des solutions vers l'origine[12].

Définition 2.2 : (Attractivité)

On dit que l'origine $x_e = 0$ est[13] :

- Un point d'équilibre attractif, s'il existe $r_0 > 0$, tel que :

$$\|x_0\| < r_0 \implies \|x(t, t_0, x_0)\| \rightarrow 0 ; \text{ quand } t \rightarrow \infty$$

- Un point d'équilibre globalement attractif si :

$$\forall x_0 \in \mathbb{R}^n \implies \|x(t, t_0, x_0)\| \rightarrow 0 ; \text{ quand } t \rightarrow \infty$$

Définition 2.3 : (Stabilité asymptotique)

Le point d'équilibre $x_e = 0$ est [12]:

- Un point d'équilibre asymptotiquement stable, s'il est stable et attractif.
- Un point d'équilibre globalement asymptotiquement stable, s'il est stable et globalement attractif.

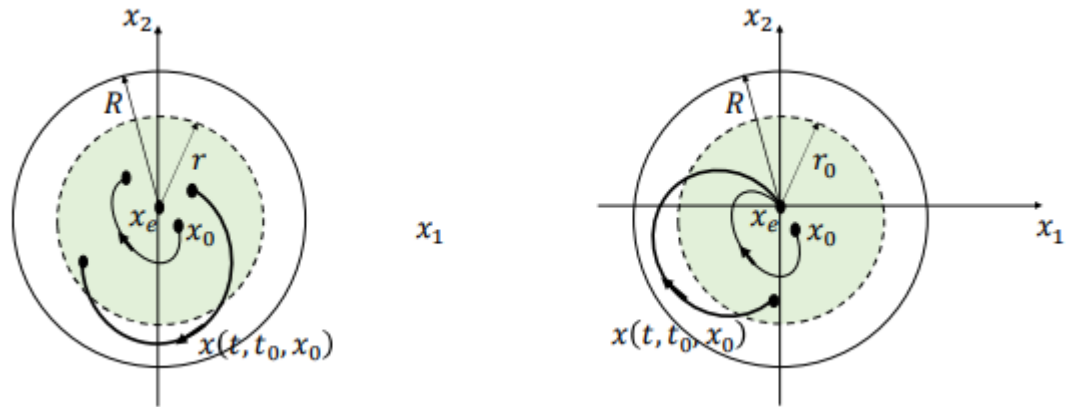


Figure 18: Stabilité au sens large et stabilité asymptotique [13]

Définition 2.4 : (Stabilité exponentielle)

Le point d'équilibre $x_e = 0$ est exponentiellement stable, s'il existe des constantes positives r, λ_1, λ_2 telles que [12][13]:

$$\|x_0\| < r \implies \|x(t, t_0, x_0)\| \leq \lambda_1 \|x_0\| e^{-\lambda_2(t-t_0)}; \quad \forall t \geq t_0$$

La constante λ_2 est appelée le taux ou aussi la vitesse de convergence.

Si en plus $r \rightarrow \infty (\forall x_0 \in \mathbb{R}^n)$, $x_e = 0$ est un point d'équilibre globalement exponentiellement stable.

Remarque 2.1 : La propriété de la stabilité exponentielle du système entraîne nécessairement la stabilité asymptotique.

Méthode directe de Lyapunov :

La méthode directe ou deuxième méthode de Lyapunov permet d'analyser la stabilité d'un système au voisinage de son point d'équilibre sans calculer explicitement la solution $x(t, t_0, x_0)$ de l'équation différentielle du système.

Cette méthode découle du concept d'énergie d'un système. Pour un système physique, l'énergie est une fonction définie positive de son état. Si le système est conservatif, l'énergie reste constante (système stable) ; elle décroît pour un système dissipatif (système asymptotiquement stable). Si l'énergie croît, le système est instable [14].

Définition 2.5 : (Fonction de Lyapunov)

On appelle fonction de Lyapunov pour le système (2.1), une fonction réelle $V(x)$ continue et

différentiable qui vérifie les propriétés suivantes [14]:

- $V(x) > 0; \forall x \neq 0$
- $V(x) = 0 \Rightarrow x = 0;$
- $\dot{V}(x) = \frac{\partial V(x)}{\partial x} f(x) \leq 0; \forall x \neq 0$

Théorème 2.1 : (Méthode directe de Lyapunov)

Si le système admet une fonction de Lyapunov, alors l'origine $x_e = 0$ est un point d'équilibre stable. Si de plus $\dot{V}(x) = \frac{\partial V(x)}{\partial x} f(x) \leq 0; \forall x \neq 0$ alors $x_e = 0$ est un point d'équilibre asymptotiquement stable[14].

Théorème 2.2 : (Stabilité exponentielle)

Le point d'équilibre $x_e = 0$ est exponentiellement stable si le système admet une fonction de Lyapunov et s'il est existé des constantes positives γ , ρ_1 et ρ_2 , telles que [14]:

- $\rho_1 \|x\|^2 \leq V(x) \leq \rho_2 \|x\|^2$
- $\dot{V}(x) \leq -\gamma V(x)$

2.2 Commande par Backstepping

Système d'ordre 3 :

Le Backstepping n'a aucune contrainte au niveau du type de non-linéarité. Cependant, pour que la technique puisse s'appliquer, le système non linéaire doit être sous forme « strict feedback » (rétroaction stricte), la dérivée de chaque composante du vecteur d'état doit être une fonction des composantes précédentes et dépend additivement de la composante suivante. Les équations d'un tel système sont données par [12][14]:

$$\begin{cases} \dot{x}_1 = f_1(x_1) + g_1(x_1)x_2 \\ \dot{x}_2 = f_2(x_1, x_2) + g_2(x_1, x_2)x_3 \\ \dot{x}_3 = f_3(x_1, x_2, x_3) + g_3(x_1, x_2, x_3)u \\ y = x_1 \end{cases}; g_i(x) \neq 0$$

On désire faire suivre à la sortie y le signal de référence y_r , les signaux \dot{y}_r , \ddot{y}_r et $\ddot{\ddot{y}}_r$ sont supposées connues et uniformément bornées. Le système étant du troisième ordre, la conception s'effectue en trois étapes. Dans chaque étape on introduit un écart entre une variable d'état et sa valeur désirée appelée commande virtuelle et une fonction de Lyapunov stabilisante :

La procédure de conception d'une commande par backstepping est telle que :

- La commande réelle u stabilise l'état x_3 ;
- La commande virtuelle $(x_3)_{des}$ (valeur désirée de x_3) stabilise l'état x_2 ;
- La commande virtuelle $(x_2)_{des}$ (valeur désirée de x_2) stabilise l'état x_1 et donc la sortie y .

Étape 1 :

On définit une première variable d'erreur $\varepsilon_1 = y - y_r$, c'est l'erreur d'asservissement. Sa dérivée est:

$$\dot{\varepsilon}_1 = f_1(x_1) + g_1(x_1)x_2 - \dot{y}_r$$

La variable d'état x_2 est traitée comme une commande virtuelle permettant de stabiliser ε_1 . On introduit la première fonction de Lyapunov candidate $V_1(\varepsilon_1) = \frac{1}{2}\varepsilon_1^2$. Sa dérivée est donnée par:

$$\dot{V}_1(\varepsilon_1) = \varepsilon_1 \dot{\varepsilon}_1 = \varepsilon_1(f_1(x_1) + g_1(x_1)x_2 - \dot{y}_r)$$

Un choix judicieux de x_2 et qui rends \dot{V}_1 négative et assure la stabilité de l'origine du soussystème (2.2). Prenons comme commande virtuelle $\alpha_1 = (x_2)_d$, c'est la valeur désirée de x_2 . Le choix de α_1 est telle que :

$$f_1(x_1) + g_1(x_1)\alpha_1 - \dot{y}_r = -k_1\varepsilon_1$$

Où $k_1 > 0$ est un paramètre de réglage. Cela donne:

$$\alpha_1 = \frac{1}{g_1(x_1)}(-k_1\varepsilon_1 - f_1(x_1) + \dot{y}_r)$$

Avec ce choix : $\dot{V}_1(\varepsilon_1) = -k_1\varepsilon_1^2 \leq 0$. D'où la stabilité asymptotique de ε_1 vers zéro.

Etape 2:

Comme x_2 n'est une vraie commande. On définit une deuxième variable d'erreur entre la variable d'état x_2 est sa valeur désirée α_1 : $\varepsilon_2 = x_2 - \alpha_1$. La variable x_2 ne peut être forcée instantanément vers sa valeur désirée, alors l'erreur ε_2 n'est pas instantanément nulle. La conception dans cette étape consiste à converger ε_2 vers zéro avec une dynamique choisie. La dynamique des erreurs ε_1 et ε_2 est:

$$\begin{cases} \dot{\varepsilon}_1 = f_1(x_1) + g_1(x_1)(\varepsilon_2 + \alpha_1) - \dot{y}_r = -k_1\varepsilon_1 + g_1(x_1)\varepsilon_2 \\ \dot{\varepsilon}_2 = f_2(x_1, x_2) + g_2(x_1, x_2)x_3 - \dot{\alpha}_1 \end{cases}$$

Soit $\alpha_2 = (x_3)_d$ la commande virtuelle permettant de stabiliser le sous système (2,3) On définit pour ce sous-système une deuxième fonction de Lyapunov:

$$V_2(\varepsilon_1, \varepsilon_2) = \frac{1}{2}\varepsilon_1^2 + \frac{1}{2}\varepsilon_2^2 = V_1 + \frac{1}{2}\varepsilon_2^2$$

Sa dérivée est :

$$\dot{V}_2(\varepsilon_1, \varepsilon_2) = \dot{V}_1 + \varepsilon_2\dot{\varepsilon}_2 = -k_1\varepsilon_1^2 + \varepsilon_2(g_1(x_1)\varepsilon_1 + f_2(x_1, x_2) + g_2(x_1, x_2)x_3 - \dot{\alpha}_1)$$

Le choix de la valeur désirée de x_3 est donné par:

$$g_1(x_1)\varepsilon_1 + f_2(x_1, x_2) + g_2(x_1, x_2)\alpha_2 - \dot{\alpha}_1 = -k_2\varepsilon_2; k_2 > 0$$

On déduit :

$$\alpha_2 = \frac{1}{g_2(x_1, x_2)}(-k_2\varepsilon_2 - g_1(x_1)\varepsilon_1 - f_2(x_1, x_2) + \dot{\alpha}_1)$$

La fonction $\dot{\alpha}_1$ peut être calculée analytiquement:

$$\dot{\alpha}_1 = \frac{\partial \alpha_1}{\partial x_1} \dot{x}_1 + \frac{\partial \alpha_1}{\partial y_r} \dot{y}_r + \frac{\partial \alpha_1}{\partial \dot{y}_r} \ddot{y}_r$$

Un tel choix, conduit à : $\dot{V}_2(\varepsilon_1, \varepsilon_2) = -k_1\varepsilon_1^2 - k_2\varepsilon_2^2 \leq 0$, ce qui assure la stabilité asymptotique de $(\varepsilon_1, \varepsilon_2)$ vers $(0,0)$.

Etape 3:

On définit la dernière variable d'erreur : $\varepsilon_3 = x_3 - \alpha_3$. La dynamique des erreurs $\varepsilon_1, \varepsilon_2$ et ε_3 est :

$$\begin{cases} \dot{\varepsilon}_1 = -k_1 \varepsilon_1 + g_1(x_1) \varepsilon_2 \\ \dot{\varepsilon}_2 = -g_1(x_1) \varepsilon_1 - k_2 \varepsilon_2 + g_2(x_1, x_2) \varepsilon_3 \\ \dot{\varepsilon}_3 = f_3(x_1, x_2, x_3) + g_3(x_1, x_2, x_3) u - \dot{\alpha}_2 \end{cases}$$

La troisième fonction de Lyapunov est augmentée comme suit :

$$V_3(\varepsilon_1, \varepsilon_2, \varepsilon_3) = \frac{1}{2} \varepsilon_1^2 + \frac{1}{2} \varepsilon_2^2 + \frac{1}{2} \varepsilon_3^2 = V_2 + \frac{1}{2} \varepsilon_3^2$$

Sa dérivée est :

$$\begin{aligned} \dot{V}_3(\varepsilon_1, \varepsilon_2, \varepsilon_3) &= \dot{V}_2 + \varepsilon_3 \dot{\varepsilon}_3 \\ \dot{V}_3 &= -k_1 \varepsilon_1^2 - k_2 \varepsilon_2^2 + \varepsilon_3 (g_2(x_1, x_2) \varepsilon_2 + f_3(x_1, x_2, x_3) + g_3(x_1, x_2, x_3) u - \dot{\alpha}_2) \end{aligned}$$

La vraie commande u apparaît dans cette étape. La conception de la loi de commande par la méthode de backstepping est terminée. Le choix de u est telle que :

$$g_2(x_1, x_2) \varepsilon_2 + f_3(x_1, x_2, x_3) + g_3(x_1, x_2, x_3) u - \dot{\alpha}_2 = -k_3 \varepsilon_3; \quad k_3 > 0$$

Cela donne l'expression de la commande :

$$u = \frac{1}{g_3(x_1, x_2, x_3)} (-k_3 \varepsilon_3 - g_2(x_1, x_2) \varepsilon_2 - f_3(x_1, x_2, x_3) + \dot{\alpha}_2)$$

L'expression analytique de $\dot{\alpha}_2$ est :

$$\dot{\alpha}_2 = \frac{\partial \alpha_2}{\partial x_1} \dot{x}_1 + \frac{\partial \alpha_2}{\partial x_2} \dot{x}_2 + \frac{\partial \alpha_2}{\partial y_r} \dot{y}_r + \frac{\partial \alpha_2}{\partial \dot{y}_r} \ddot{y}_r + \frac{\partial \alpha_2}{\partial \ddot{y}_r} \dddot{y}_r$$

Avec ce choix, on obtient :

$$\dot{V}_3(\varepsilon_1, \varepsilon_2, \varepsilon_3) = -k_1 \varepsilon_1^2 - k_2 \varepsilon_2^2 - k_3 \varepsilon_3^2 \leq 0$$

La fonction V_3 apparaît maintenant comme une fonction de Lyapunov pour le système (2.4), ce qui prouve la stabilité asymptotique de (s_1, s_2, s_3) vers l'origine. Ceci se traduit par la stabilité, en boucle fermée, du système (2.2) et la régulation à zéro de l'erreur de poursuite $s_1 = y - y_r$. Les performances de l'asservissement en boucle fermée sont données par les

paramètres de réglage k_1 , k_2 et k_3 .

Système d'ordre n

L'application récursive du backstepping permet l'extension de la procédure de synthèse aux systèmes de la forme :

$$\begin{cases} \dot{x}_1 = f_1(x_1) + g_1(x_1)x_2 \\ \dot{x}_2 = f_2(x_1, x_2) + g_2(x_1, x_2)x_3 \\ \vdots \\ \dot{x}_{n-1} = f_{n-1}(x_1, \dots, x_{n-1}) + g_{n-1}(x_1, \dots, x_{n-1})x_n; g_i(x) \neq 0 \\ \dot{x}_n = f_n(x_1, \dots, x_n) + g_n(x_1, \dots, x_n)u \\ y = x_1 \end{cases}$$

Algorithme de commande :

$$(x_1)_d = \alpha_0 = y_r$$

$$(x_{i+1})_d = \alpha_i = \frac{1}{g_i} \left[-k_i \varepsilon_i - g_{i-1} \varepsilon_{i-1} - f_i + \sum_{k=1}^{i-1} \left(\frac{\partial \alpha_{i-1}}{\partial x_i} f_k + \frac{\partial \alpha_{i-1}}{\partial x_i} g_k x_{k+1} \right) + \right.$$

$$\left. \sum_{k=1}^i \frac{\partial \alpha_{i-1}}{\partial y_r^{(k-1)}} y_r^{(k)} \right]$$

$$\varepsilon_i = x_i - \alpha_{i-1}; i = 1, 2, \dots, n$$

$$V_i = \frac{1}{2} \sum_{i=1}^n (x_i - \alpha_{i-1})^2$$

La commande qui permet d'atteindre la stabilité asymptotique du système global est donnée par la dernière commande virtuelle $u = \alpha_n$.

3. Application de la commande BACKSTEPPING :

On peut écrire le système de la forme suivante :

$$\Sigma_{NL} = \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f_1(x, u) + g_{11}(x)\tau_1 + g_{12}(x)\tau_2 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = f_2(x, u) + g_{21}(x)\tau_1 + g_{22}(x)\tau_2 \end{cases}$$

$$f(x, u) = \begin{cases} \frac{-M_{22}C_1 + M_{12}C_2}{\det(M)} + \frac{x_2}{\det(M)}\tau_1 + \frac{-M_{12}}{\det(M)}\tau_2 \\ \frac{M_{21}C_1 - M_{11}C_2}{\det(M)} + \frac{x_4}{\det(M)}\tau_1 + \frac{M_{11}}{\det(M)}\tau_2 \end{cases}$$

Avec:

$$g_{11} = \frac{M_{22}}{\det(M)}$$

$$g_{12} = \frac{-M_{12}}{\det(M)}$$

$$g_{21} = \frac{-M_{21}}{\det(M)}$$

$$g_{22} = \frac{M_{11}}{\det(M)}$$

$$f_1 = \frac{-M_{22}C_1 + M_{12}C_2}{\det(M)}$$

$$f_2 = \frac{M_{21}C_1 - M_{11}C_2}{\det(M)}$$

Cette forme n'est pas une forme cascade, donc la méthode du Backstepping n'est pas applicable, on doit réécrire le système dans une autre base où une partie du vecteur d'état apparaît comme une commande virtuelle sur l'autre partie du vecteur d'état.

$$\text{Soit : } Z_1 = \begin{pmatrix} x_1 \\ x_3 \end{pmatrix} \text{ et } Z_2 = \begin{pmatrix} x_2 \\ x_4 \end{pmatrix} \text{ donc le système devient : } \begin{cases} \dot{Z}_1 = Z_2 \\ \dot{Z}_2 = F_z + G_z u \end{cases}$$

Tel que :

$$F_z = \begin{pmatrix} f_1(x, u) \\ f_2(x, u) \end{pmatrix} \quad G_z = \begin{pmatrix} g_{11}(x) & g_{12}(x) \\ g_{21}(x) & g_{22}(x) \end{pmatrix} \quad u = \begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix}$$

Le principe de cette méthode est le suivant :

Etape 1 :

Considérer dans un premier temps Z_2 comme étant une commande pour Z_1 . Il ne reste plus qu'à synthétiser la nouvelle commande pour obtenir la convergence de l'erreur vers zéro.

On définit alors la fonction candidate de Lyapunov V_1 définie et positive :

$$V_1 = \frac{1}{2} (Z_1 - Z_r)^2 = (Z_1 - Z_r)^T (Z_1 - Z_r) \text{ ou } Z_r \text{ est la consigne}$$

$$\text{Sa dérivée est donnée par } \dot{V}_1 = (Z_1 - Z_r)^T \dot{Z}_1 = (Z_1 - Z_r)^T Z_2$$

$$\text{Si } Z_2 = -\varphi_2 = -\lambda_2 (Z_1 - Z_r)^T \text{ avec } \lambda_2 > 0$$

$$\text{Donc } \dot{V}_1 = -\lambda_2 (Z_1 - Z_r)^2 \text{ est une fonction définie négative sur } \mathcal{R}^2$$

Etape 2 :

Calculer la commande u permettant de faire tendre Z_2 vers $-\varphi_2$

Pour cela, on définit la fonction de Lyapunov suivante :

$$V_2 = \frac{1}{2} (Z_1 - Z_r)^2 + \frac{1}{2} (Z_2 - (-\varphi_2))^2$$

Sa dérivée est donnée par :

$$\begin{aligned} \dot{V}_2 &= (Z_1 - Z_r)^T Z_2 + (Z_2 + \varphi_2)^T (\dot{Z}_2 + \dot{\varphi}_2) \\ &= -(Z_1 - Z_r)^T \varphi_2 + (Z_1 - Z_r)^T (Z_2 + \varphi_2) + (Z_2 + \varphi_2)^T (Z_2 + \varphi_2) \\ &= V_1 + (Z_1 - Z_r)^T (Z_2 + \varphi_2) + (Z_2 + \varphi_2)^T (F_z + G_z u + \varphi_2) \end{aligned}$$

$$\text{Soit } G_z u = -(Z_1 - Z_r) - F_z - \dot{\varphi}_2 - \lambda_2 (Z_2 + \varphi_2)$$

$$\text{Donc } u = G_z^{-1} [-(Z_1 - Z_r) - F_z - \lambda_1 Z_2 - \lambda_2 (Z_2 + \lambda_1 (Z_1 + Z_r))]$$

Nous obtenons finalement la fonction définie négative suivante :

$$\dot{V}_2 = -\lambda_1 (Z_1 - Z_r)^2 - \lambda_2 (Z_2 + \varphi_2)^2$$

Pour voir l'effet de cette commande, une simulation sur SUMILINK du logiciel MATLAB est réalisé comme le montre la figure suivante:

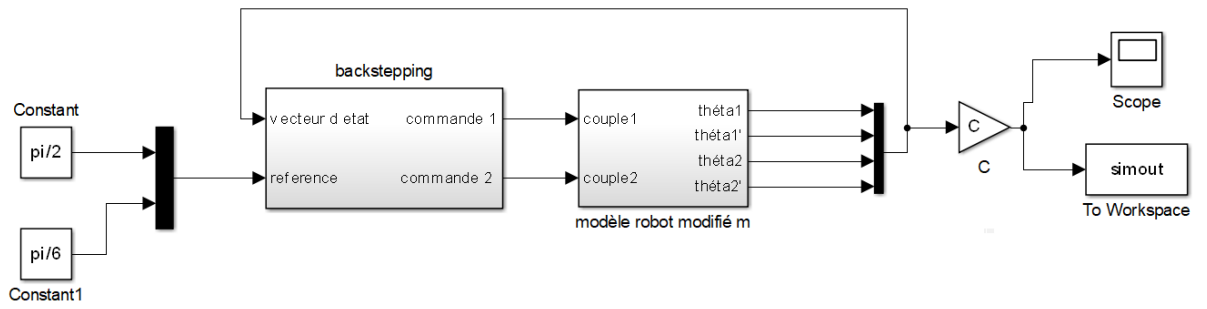


Figure 19:: Montage de commande du robot par Backstepping

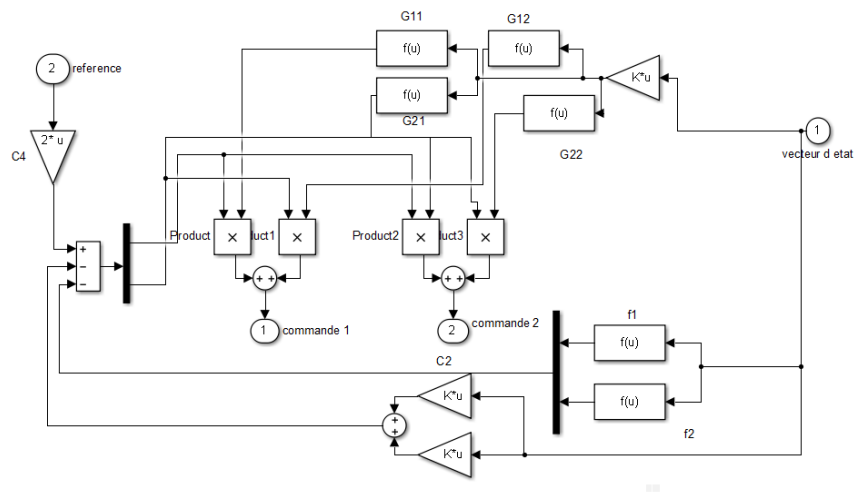


Figure 20:Montage de contrôleur du robot par Backstepping

Les résultats de simulation pour les valeurs de référence $\theta_{1ref} = \frac{\pi}{2}$ et $\theta_{2ref} = \frac{\pi}{6}$ ref sont satisfaisants comme l'indiquent les 3 figures :

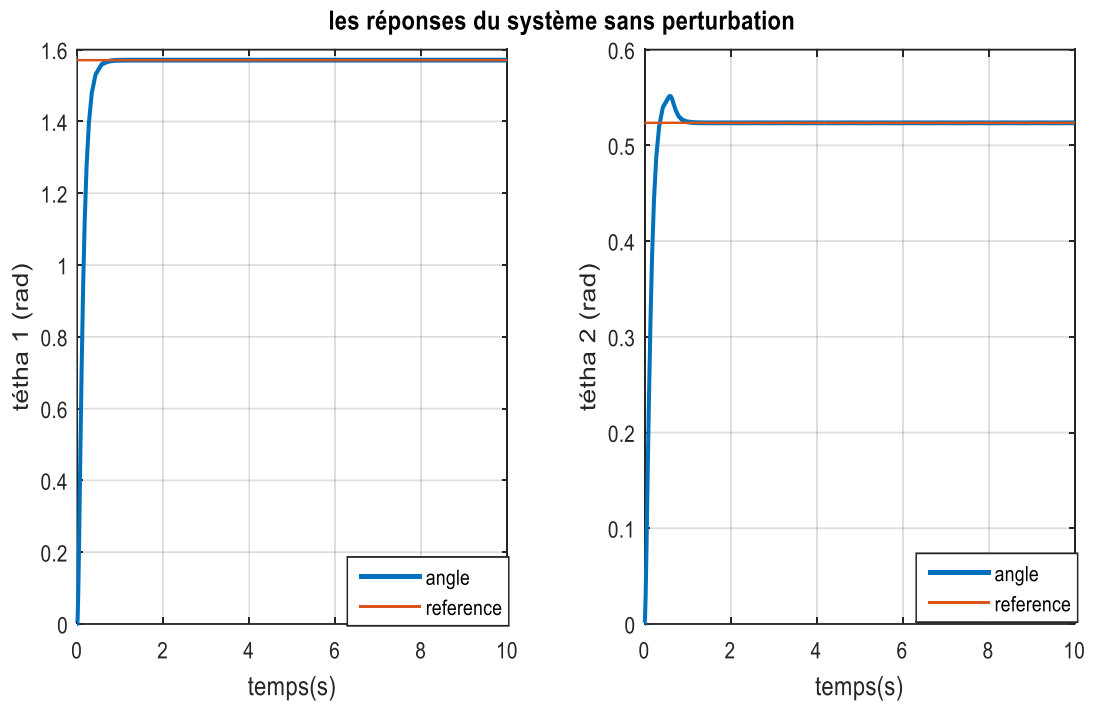


Figure 21: Les 2 positions et leurs valeurs désirées du robot commandé par Backstepping sans présence de perturbation

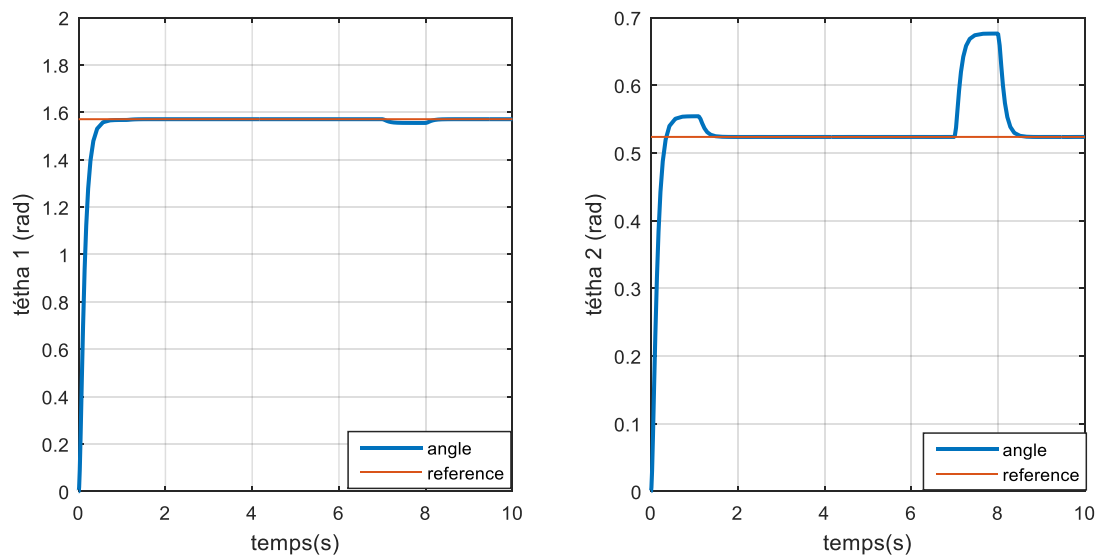


Figure 22: Les 2 positions et leurs valeurs désirées du robot commandé par Backstepping avec présence de perturbation

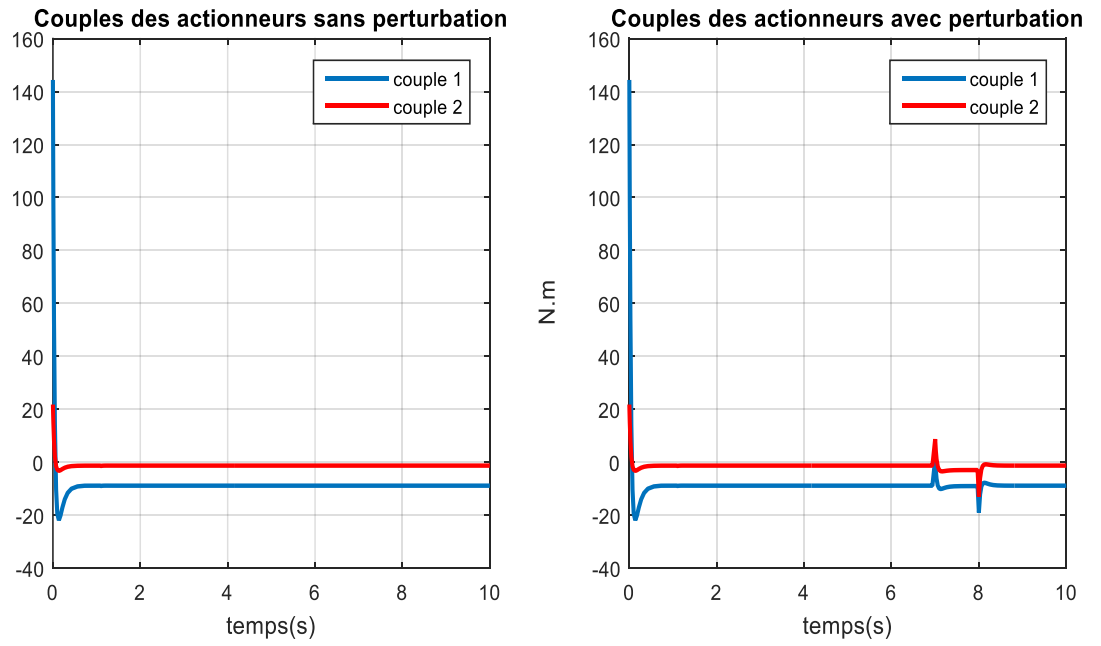


Figure 23: Les couples générés par Backstepping

Les résultats obtenus montrent que le robot atteint sa position désirée et avec erreur statique nulle, et avec une commande amissible.

4. Conclusion :

Dans ce chapitre, nous avons fait une description de la commande par « Backstepping ». Cette commande permet d'obtenir de façon constructive une loi de commande qui assure la stabilité du système en boucle fermée via une fonction de Lyapunov.

Pour valider cette technique nous l'avons appliquée sur un robot à deux degrés de liberté, les résultats obtenus sont encourageants, les erreurs de suivi sont excellentes. La commande est relativement acceptable pour une proposition expérimentale.

Par contre nous avons des faibles oscillations ,on peuvent être diminués par élimination de quelques non linéarités dans la commande du système

Conclusion générale

Conclusion générale

L'objectif assigné à ce travail était d'élaborer une commande non linéaire Backstepping dédiée à la commande de position d'un bras manipulateur à couple électrique. Une telle commande se base sur une approche analytique.

La première étape du travail a porté des définitions nécessaires pour bien familiarisé des robots manipulateurs puis on a présenté la modélisation des robots en trois modèles (géométrique, cinématique, dynamique), en deuxième chapitre on a appliqué une modélisation dynamique sur notre modèle ce qui nous donne un système non linéaire complexe interconnecté qu'on a déjà donné une initiation au début de ce chapitre

En effet, ces travaux ont eu pour but de proposer des lois de commande qui répondent impérativement aux objectifs précis, tel que l'évolution de robot manipulateur sous contraintes de performance, rapidité et de précision, pour cette raison une étude théorique concernant les commandes classiques tel que la commande par retour d'état et la commande par linéarisation exacte, et une commande non linéaire tel que la commande par Backstepping.

Pour la commande par retour d'état nous avons montré des bonnes performances, mais il faut toujours garder les références au voisinage du point d'équilibre pour que le système soit stable

Pour la commande par linéarisation exact nous avons montré des bonnes performances, mais la réponse possède des oscillations qui peuvent provoquent des commandes inacceptables pour une commande réelle

Enfin pour la commande par Backstepping nous avons montré des bonnes performances par rapport au derniers commandes.

Nous avons essayé au cours de ce travail de trouver une solution robuste pour améliorer les Performances des commandes non linéaires.

Comme perspectives nous espérons d'appliquer ces méthodes expérimentalement sur le robot manipulateur.

Référence

- [1] A. LOTFAZAR, M. EGHTEHAD, "Application of Passivity Based and Integrator Backstepping Control Methods on a 5 DOF Robot Manipulator Incorporating Actuator Dynamics", Proc. of the International Conference on Control and Cybernetics, pp 147-154, New York, 2007.
- [2] N.HACENE « Commande de robots de manipulation » Université de Ghardaïa
- [3] Craig, J. J. "Introduction to Robotics: mechanics and Control". Pearson Education, Inc, troisième édition, 2005, USA.
- [4] Gu, E. Y. L. . "A Journey from Robot to Digital Human". Springer-Verlag Berlin Heidelberg, 2013.
- [5] Selig, G. M. "Introductory Robotics". Prentice Hall International (UK) Ltd, 1992, UK.
- [7] Abdellah Ghoul « Modélisation des robots manipulateurs » Université Tahri Mouhammed Béchar 2016.
- [6] Y. BAAZI, "Etude d'un Manipulateur à 3 Degrés de Liberté Application à la Fonction de Préhension", thèse de magister, Institut d'électronique, Université de Batna, 2000.
- [7] Abdellah Ghoul « Modélisation des robots manipulateurs » Université Tahri Mouhammed Béchar 2016.
- [8] Philippe Mühlaupt « Introduction à l'analyse et à la commande des systèmes non linéaires » juin 2007.
- [9] Olivier Bachelier « Représentations d'état linéaires des systèmes monovariabiles » cours d'automatique . Ecole nationale supérieure d'ingénieurs de Poitiers juin 2017.
- [10] Jean-Jacques E Slotine, Weiping Li « Applied nonlinear control » Prentice Hall Englewood Cliffs New Jersey.
- [11] Boukamel Y. « commande des systèmes non linéaires » Université de Ghardaïa 2019.
- [12] M'hammed GUISSER, Support de Cours, « Commande et Observateurs d'état des Systèmes non Linéaires », Université HASSAN 1^{er}
- [13] J. J. Slotine et W. Li, Applied Nonlinear Control. Prentice Hall, 1991
- [14] H. K. Khalil, Nonlinear Systems (Third Edition). Prentice Hall, 2001.

