

République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la Recherche scientifique

Université A. MIRA - Bejaia



Faculté de Technologie

Département d'Automatique, Télécommunication et d'Electronique

Mémoire de fin de cycle
En vue de l'obtention du diplôme de Master en Electronique
Option : Automatique

Thème

Planification et contrôle de trajectoires pour un bras de robot

Réalisé par : CHABANE Smail

Encadré et dirigé par :

Mr . Boubekour MENDIL
Mr. Lyes TIGHZERT

Devant le jury composé de :

Mr.Hocine HADDAR
Mr.Mustapha SADJI

Année universitaire : 2016/2017

DEDICACE

Je dédie ce travail

A mes parents qui m'ont encouragé tout au long de mes études.

A mes frères

.

A mon cher cousin BOUKROUNA Laarbi.

A toute ma famille.

A mes collègues de l'université.

A mes enseignants et professeurs.

A tous mes amis.

Remerciements

Mes remerciements vont tout premièrement à Dieu le tout puissant pour la volonté, la santé et la patience qu'il m'a donné pour terminer mon travail.

Mes remerciements chaleureux à mon encadreur Monsieur Boubekour MENDIL, Professeur à l'université de Béjaia, pour avoir accepté de diriger ce mémoire et de sa patience durant la période de l'encadrement.

Je remercie mon Monsieur Lyes TIGHEZERT, pour l'aide qu'il nous a apporté.

Je tiens à remercier Monsieur HEDDAR et Monsieur SADJI, pour nous avoir fait l'honneur d'être membres de jury.

Liste des figures :

Chapitre I :

- Figure I.1 : Structure fonctionnelle d'un robot*
- Figure I.2 : Symbole de l'articulation rotoïde.*
- Figure I.3 : Symbole de l'articulation prismatique.*
- Figure I.4 : Anatomie d'un bras de robot*
- Figure I.5: Structure Cartésienne.*
- Figure I.6 : Structure cylindrique.*
- Figure I.7 :Structure sphérique.*
- Figure I.8 :Structure SCARA.*
- Figure I.9 :Structure anthropomorphe.*
- Figure I.10 :Structure d'un actionneur électrique*
- Figure I.11 : Moteur à aimants permanents*
- Figure I.12 :Moteur synchrone autopiloté.*
- Figure I.13 :Moteur pas-à-pas.*
- Figure I.14 :Actionneur hydraulique.*
- Figure I.15 : Types de transmetteurs.*
- Figure I.16: Robots industriels.*

Chapitre II :

- FigureII.1 :Représentation d'un point dans l'espace*
- Figure II.2: Paramètres géométriques*
- Figure II.3 : Robot à structure ouverte simple.*
- Figure II. 4: Modèle d'un bras de robot à deux ddl*
- FigureII.5:Une force F agissant au centre de la masse d'un corps*
- Figure II.6: Plusieurs formes de chemin possibles pour une seule articulation.*
- Figure II.7: Evolution des variables articulaires, de leurs vitesses et de leurs accélérations.*

Chapitre III :

- FigureIII.1:Schéma d'un régulateur PD*
- Figure III.2 : Système bouclé en présence de perturbations*
- Figure III.3 : Structure du bras de robot étudié.*
- Figure III.4 : Schéma synoptique de simulation de la commande PD du bras de robot.*
- Figure III.5 : Formes des variables articulaires (en rouge) et des vitesses angulaires (en bleu) pour la 1^{ere} et la 2^{eme} articulation.*

Figure III.6 : Schéma synoptique de simulation de la commande PID du bras de robot.

Figure III.7 : Formes des trajectoires de la 1^{ère} et la 2^{ème} articulation : positions réelles θ (en rouge) et vitesses réelles (en bleu)..

Figure III.8 : Structure de commande non linéaire d'un bras de robot

Figures III.9 : Formes des trajectoires de la 1^{ère} et la 2^{ème} articulation : positions réelles θ (en rouge) et vitesses réelles (en bleu).

,III.10 : Formes des trajectoires de la 1^{ère} et la 2^{ème} articulation : positions réelles θ (en rouge) et vitesses réelles (en bleu)..

Figure III.11 : Schéma synoptique de la commande PD en poursuite du bras de robot.

Figure III.12: Formes des trajectoires de la 1^{ère} et la 2^{ème} articulation : positions réelles θ (en bleu) et positions désirées (en rouge).

Figures III.13 : Formes des trajectoires de la 1^{ère} et la 2^{ème} articulation : positions réelles θ (en bleu), positions désirées (en rouge) et profils des vitesses réelles (en vert).

Figures III.14: Formes des trajectoires de la 1^{ère} et la 2^{ème} articulation : positions réelles θ (en bleu), positions désirées (en rouge) et profils des vitesses réelles (en vert).

Figures III.15 : Formes des trajectoires de la 1^{ère} et la 2^{ème} articulation : positions réelles θ (en bleu), positions désirées (en rouge) et profils des vitesses réelles (en vert).

Figures III.17 : Formes des trajectoires de la 1^{ère} et la 2^{ème} articulation : positions réelles θ (en bleu), positions désirées (en rouge) et profils des vitesses réelles (en vert).

Figure III.16 : structure de la commande non linéaire en poursuite d'un bras de robot.

Figures III.17 : Formes des trajectoires de la 1^{ère} et la 2^{ème} articulation : positions réelles θ (en bleu), positions désirées (en rouge) et profils des vitesses réelles (en vert).

Figures III.18 : Formes des trajectoires de la 1^{ère} et la 2^{ème} articulation : positions réelles θ (en bleu), positions désirées (en rouge) et profils des vitesses réelles (en vert).

Figures III.19 : Formes des trajectoires de la 1^{ère} et la 2^{ème} articulation : positions réelles θ (en bleu), positions désirées (en rouge) et profils des vitesses réelles (en vert).

III.20 : Formes des trajectoires de la 1^{ère} et la 2^{ème} articulation : positions réelles θ (en bleu), positions désirées (en rouge) et profils des vitesses réelles (en vert).

Liste des abréviations :

ddl : degré de liberté.

MGD : Modèle géométrique direct.

MGI : Modèle cinématique inverse.

DHM: Convention de Denavit-Hartenber modifié .

ATAN2 : En trigonométrie, atan2 à deux arguments est une variation d'ARC tangente.

ART : Articulation.

PID : Proportionnel Intégral et Dérivé.

PD : Proportionnel et Dérivé.

TL : Transformé de Laplace

diag : diagonale.

kg : kilogramme.

m : mètre.

s : seconde.

w : vitesse angulaire

Sommaire :

Introduction générale	1
Chapitre I : Généralités sur les bras de robots	
I.1 Introduction	2
I.2 Les systèmes robotiques.....	2
I.3 Définitions générales.....	4
I.3.1 Articulation.....	4
I.3.1.1 Articulation rotoïde :	4
I.3.1.2 Articulation prismatique :	4
I.3.2 Espace articulaire :	5
I.3.3 Espace opérationnelle :	5
I.3.4 Degré de liberté :	5
I.3.5 Le mécanisme :	5
I.3.6 La perception :	5
I.3.6 La commande	5
I.3.7 L'interface homme-machine	5
I.4 Anatomie d'un bras de robot	6
I.4.1 La base.....	6
I.4.2 Le porteur	6
I.4.2.1 La structure cartésienne PPP	7
I.4.2.2 La structure cylindrique RPP (ou PRP).....	7
I.4.2.3 La structure sphérique	7
I.4.2.4 La structure dite SCARA	8
I.4.2.5 La structure anthropomorphe à 3R.....	8
I.4.3 Les actionneurs.....	9
I.4.3.1 Motorisation des articulations	9
I.4.3.2 Actionneurs électriques	9
I.4.3.2.1 Moteurs à courant continu.....	10
I.4.3.2.2 Moteurs synchrones autopilotés	11
I.4.3.2.3 Les moteurs pas à pas	11
I.4.3.3 Actionneurs hydrauliques.....	12
I.4.3.4 Actionneurs pneumatique.....	12

I.4.4 Les transmetteurs.....	12
I.4.5 Les capteurs.....	13
I.4.5.1 Les capteurs proprioceptifs	13
I.5 Les capteurs extéroceptifs	13
I.4.6 Les outils	13
I.5 Différentes générations de robots.....	14
I.5.1 Les robots passifs	14
I.5.2 Les robots actifs :	14
I.5.3 Les robots intelligents	14
I.6 Caractéristiques des robots.....	14
I.7 Domaines d'application.....	15
I.7.1 L'industrie	15
I.7.2 Exploration spatiale.....	16
I.7.3 Militaire	16
I.8 Conclusion.....	16

Chapitre II : Modélisation des bras de robots.....

II.1 Introduction.....	17
II.2 Modélisation géométrique	17
II.2.1 Représentation d'un point dans l'espace:	17
II.2.2 Matrices de transformation homogènes	18
II.2.3 Convention de Denavit-Hartenberg modifiée pour le calcul du MGD:.....	19
II.2.4 Modèle géométrique inverse (MGI):	21
II.2.5 Calcul du modèle géométrique d'un bras de robot à deux d.d.l	21
II.3 Modélisation cinématique.....	23
II.3.1 Modèle cinématique direct.....	23
II.3.2 Modèle cinématique inverse	23
II.4 Modélisation dynamique.....	24
II.4.1 Formalisme de Lagrange:	24
II.4.2 Formalisme de Newton-Euler	25
II.4.3 Modèle dynamique d'un bras de robot à 2 d.d.l:	26
II.5 Génération de trajectoires	27
II.5.1 Méthode des polynômes cubiques	27

II.5.2 Planification de trajectoire pour un bras de robot a 2 ddl	29
II.6. Conclusion	30

Chapitre III : Commande des bras de robots.....

III.1 Introduction	31
III.2 Technique de commande des bras de robot :	31
III.2.1 Régulation linéaire.....	31
III.2.1.1 Régulateur proportionnel dérivé PD :.....	31
III.2.1.2 Régulateur PID	32
III.2.2 Commande et linéarisation	32
III.2.3 Commande par découplage non linéaire :	33
III.2.4 Poursuite de trajectoire	34
III.3 Modèle dynamique du robot utilisé.....	35
III.4 Commande en régulation du bras de robot.....	36
III.4.1 Commande PD	36
III.4.2 Commande PID	38
III.4.3 Commande découplante linéarisante	40
III.4.3.1 Régulation PD :	41
III.4.3.2 Régulation PID	42
III.5 Commande en poursuite	43
III.5.1 Commande PD	44
III.5.2 Commande PID:	47
III.5.3 Commande linéarisante découplante	49
III.5.3.1 Régulation PD :	49
III.5.3.2 Régulation PID	51
III.6 Conclusion	53

Conclusion générale54

Bibliographies

Annexes

Introduction générale

Introduction générale :

La compétition industrielle est très vive et ne laisse survivre que les industries capables de produire à faible coût ou de fournir rapidement des produits nouveaux et/ou de qualité supérieure et garantie. Ces exigences ont conduit au développement d'une automatisation flexible impliquant l'implantation de robots dans de nombreuses branches d'activités. C'est l'apparition des robots et leur association, à d'autres machines automatiques d'inspiration "classique" (par exemple, des machines-outils à commande numérique) qui permettent d'automatiser ou d'essayer d'automatiser des opérations considérées jusqu'ici comme non automatisables [10].

Les opérations complexes exigent non seulement la flexibilité mais aussi une information sur l'environnement. Suite à l'évolution de la robotique, les bras de robot se substituent aux êtres humains dans les chaînes industrielles d'une façon plus efficace. Ils peuvent travailler sans pause et encore sans absence et ne connaissent ni la fatigue ni l'ennui.

La plupart des opérations (tâches) réalisées par les bras de robots sont des tâches répétitives définies par l'utilisateur. Donc, c'est à ce dernier de définir et de commander les mouvements des différentes articulations selon la tâche à réaliser.

Le problème de la commande d'un bras de robot réside dans le calcul des forces (et/ou couples) requis que les actionneurs doivent générer afin de réaliser la tâche désirée. Néanmoins, les robots sont des systèmes multi-variables, non linéaires, et variant dans le temps. Ce qui exige des techniques de commande adaptées. Ceci fait l'objet de notre étude. Ainsi, des techniques de la commande classique linéaire et de la commande linéarisante découplante ont été étudiées et simulées pour un bras à deux degrés de liberté.

Ce mémoire est organisé en trois chapitres :

Le premier chapitre consacré pour la présentation générale des bras de robots.

Le deuxième chapitre expose la modélisation géométrique, cinématique et dynamique des bras de robots.

Enfin, le troisième chapitre présente notre travail. On y trouve un exposé sur chaque méthode utilisée, les calculs des régulateurs et l'implémentation et la simulation du système résultant.

Chapitre I :

Généralités sur les bras de robots

I.1 Introduction

L'étymologie du mot robot est de la langue tchèque du mot '*robota*' qui signifie travail. Le mot robot apparue pour la première fois dans une pièce théâtrale écrite par l'écrivain Karel Capek en 1920 et jouée pour la première fois en 1921.

Au début, les robots sont faits pour remplacer les ouvriers dans une chaîne industrielle. Actuellement, les applications sont diverses et la robotique est une discipline à part entière qui exploite plusieurs outils et fait appel à un ensemble de techniques pour la conception et la réalisation des machines automatiques de plus en plus intelligentes.

I.2 Les systèmes robotiques

Si les manipulateurs et automates existent depuis de nombreuses décennies, c'est seulement vers les années 1970 que sont véritablement apparues en industrie les premières machines que l'on puisse qualifier de robots. En effet, le robot est avant tout une machine industrielle poly-articulée, dotée de possibilités d'auto-adaptabilité et qui peut agir d'une manière plus ou moins autonome dans et sur son environnement. Depuis 1980 surtout, de très nombreux ouvrages ont été publiés, proposant de multiples définitions et points de vue sur la robotique, les robots, leur place dans la production industrielle et de notre environnement quotidien. Cette littérature scientifique, pseudo-scientifique ou de vulgarisation reflète les certitudes, les espoirs et les craintes de notre société vis-à-vis de machines auxquelles est confiée une part de nos capacités intellectuelles [1].

Un robot peut être défini comme une machine physique qui modifie son environnement pour atteindre le but qui lui est fixé (la tâche) et qui est dotée de flexibilité lui permettant d'exécuter une tâche donnée de plusieurs manières et/ou des tâches variés.

Un système robotique se situe par rapport à trois entités : un environnement de travail, une source d'énergie et une source d'informations (Figure 1.1). Cette dernière entité est fondamentale pour l'utilisation des robots dans une cellule de travail ou dans un atelier de production. Car, c'est à ce niveau que se situe le dialogue homme-machine (programmation, apprentissages, commande, maintenance, etc.).

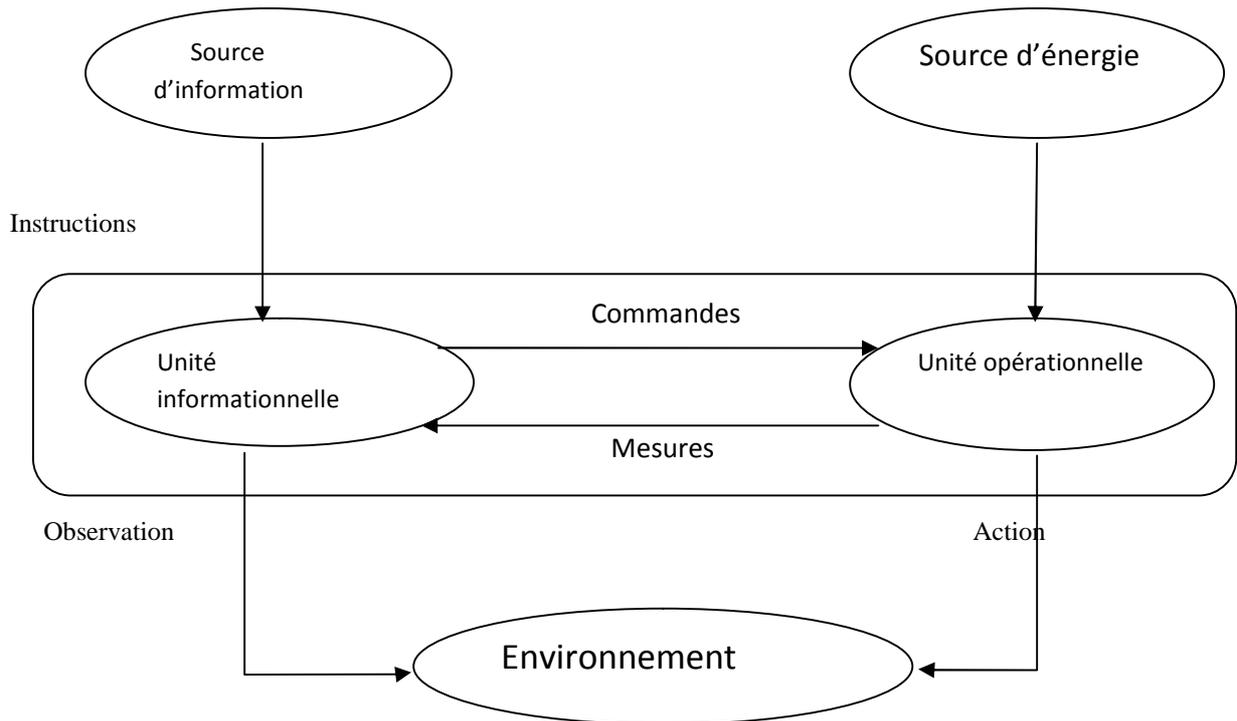


Figure I.1. Structure fonctionnelle d'un robot

Le système robotique lui-même se décompose en :

➤ **Une unité informationnelle** : qui reçoit les instructions décrivant la tâche à accomplir, les mesures relatives à l'état interne de la structure mécanique qui constitue le bras manipulateur et les observations concernant son environnement. Elle élabore en conséquence les commandes de ses différentes articulations en vue de l'exécution de la tâche. Les systèmes actuels fonctionnent en interaction permanente, selon le cycle « information – décision – action ».

➤ **Une unité opérationnelle** : qui exerce les actions commandées en empruntant la puissance nécessaire à la source d'énergie. Cette partie, qui constitue le robot physique, intègre la structure mécanique (segments, articulations, architecture, etc.), les modulateurs d'énergie (amplificateurs, variateurs, servovalves ..), les convertisseurs d'énergie (moteurs, vérins...), les chaînes cinématiques de transmission mécanique (réducteurs, vis à billes, courroies crantées...), les capteurs proprioceptifs placés sur chaque articulation pour mesurer en performance leurs positions et leurs vitesses, et enfin l'effecteur, ou organe terminal, qui est en interaction avec l'environnement.

I.3 Définitions générales

I.3.1 Articulation [3]

Une articulation lie deux corps successifs en limitant le nombre de degré de liberté (notion précisée au §2.2) de l'un par rapport à l'autre. Soit m le nombre de degrés 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.

I.3.1.1 Articulation rotoïde :

Il s'agit d'une articulation de type pivot, notée R , réduisant le mouvement entre deux corps à une rotation autour d'un axe qui leur est commun. La situation relative entre les deux corps est donnée par l'angle autour de cet axe (Figure I.2).



Figure I.2. Symbole de l'articulation rotoïde.

I.3.1.2 Articulation prismatique :

Il s'agit d'une articulation de type glissière, notée P , réduisant le mouvement entre deux corps à une translation le long d'un axe commun. La situation relative entre les deux corps est mesurée par la distance le long de cet axe (Figure I.3).



Figure I.3. Symbole de l'articulation prismatique.

I.3.2 Espace articulaire :

L'*espace articulaire* est l'espace qui a pour référence le repère lié à chaque articulation motorisée du robot. Les coordonnées associées sont appelées *coordonnées articulaires*.

I.3.3 Espace opérationnelle :

L'*espace opérationnel* est l'espace qui a pour référence le repère lié à l'organe terminal du robot. Les coordonnées associées sont appelées *coordonnées opérationnelles*.

I.3.4 Degré de liberté :

Le nombre m de degrés de liberté d'un bras de robot (m est un nombre naturelle positif) correspond au nombre d'articulations du bras de robot.

I.3.5 Le mécanisme :

La structure du mécanisme est plus au moins proche de celle du bras humain. On dit aussi manipulateur quand il ne s'agit pas d'un robot mobile. Sa motorisation est réalisée par des actionneurs électriques, pneumatiques ou hydrauliques qui transmettent leurs mouvements aux articulations par des systèmes appropriés.

I.3.6 La perception :

Permet de gérer les relations entre le robot et son environnement. Les organes de perception sont des capteurs dits « *proprioceptifs* » lorsqu'ils mesurent l'état interne du robot (position et vitesses des articulations) ou « *extéroceptifs* » lorsqu'ils recueillent des informations sur l'environnement (détection de présence, mesure de distance, vision artificielle).

I.3.7 La commande :

Qui synthétise les consignes des asservissements pilotant les actionneurs. A partir de la fonction de perception et des ordres de l'utilisateur, elle permet d'engendrer les actions du robot [11].

I.3.8 L'interface homme-machine :

C'est l'interface à travers laquelle l'utilisateur programme les tâches que le robot doit exécuter.

I.4 Anatomie d'un bras de robot :

Lorsqu'on parle d'anatomie on pense à l'être humain. Mais, dans notre cas, on s'intéresse au bras de robot. Ce dernier est composée de : la base, les segments, les articulations, les actionneurs, l'organe terminal et l'effecteur.

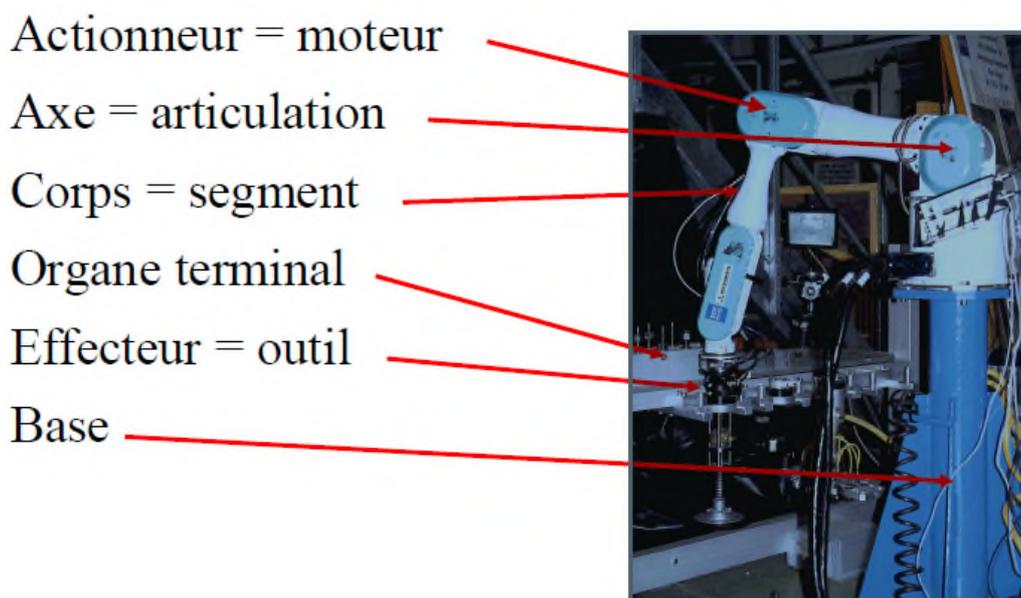


Figure I.4. Anatomie d'un bras de robot

I.4.1 La base :

La base est la partie qui permet de fixer le bras sur le lieu de travail. On la considère comme le premier repère pour localiser le robot et les objets qui l'entourent.

I.4.2 Le porteur :

Le porteur est une chaîne constituée d'éléments de translations et/ou de rotations animés les uns par rapport aux autres et qui permettent de déplacer et d'orienter l'organe terminal (ou effecteur)[2].

Les robots industriels peuvent être classifiés en fonction de leurs structures mécaniques et du type d'articulations. Les principales configurations des porteurs industriels sont :

I.4.2.1 La structure cartésienne PPP :

Cette structure est constituée de trois liaisons prismatiques. Elle est la plus ancienne. Historiquement, elle découle de la conception traditionnelle d'une machine-outil à trois axes.

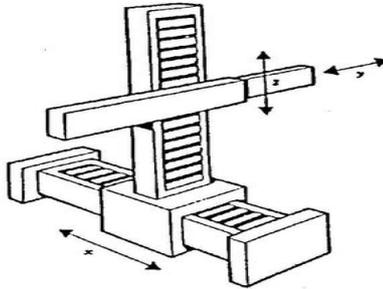


Figure I.5. Structure Cartésienne.

I.4.2.2 La structure cylindrique RPP (ou PRP) :

Elle associe une rotation et deux translations. Elle présente l'inconvénient d'offrir un volume de travail faible devant un encombrement total important.

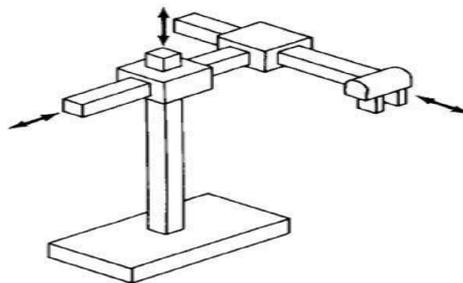


Figure I.6. Structure cylindrique.

I.4.2.3 La structure sphérique :

Il s'agit d'une structure qui est proche de la structure cylindrique et possède les mêmes inconvénients. Cette structure est abandonnée.

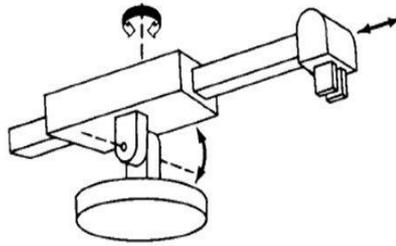


Figure I.7. Structure sphérique.

I.4.2.4 La structure dite SCARA:

La structure SCARA (*Selective Compliance Adaptive Robot Arm*), possède des axes de rotation parallèles. Elle est l'une des plus utilisées, en particulier pour les tâches de manutention ou d'assemblages très fréquentes dans l'industrie.

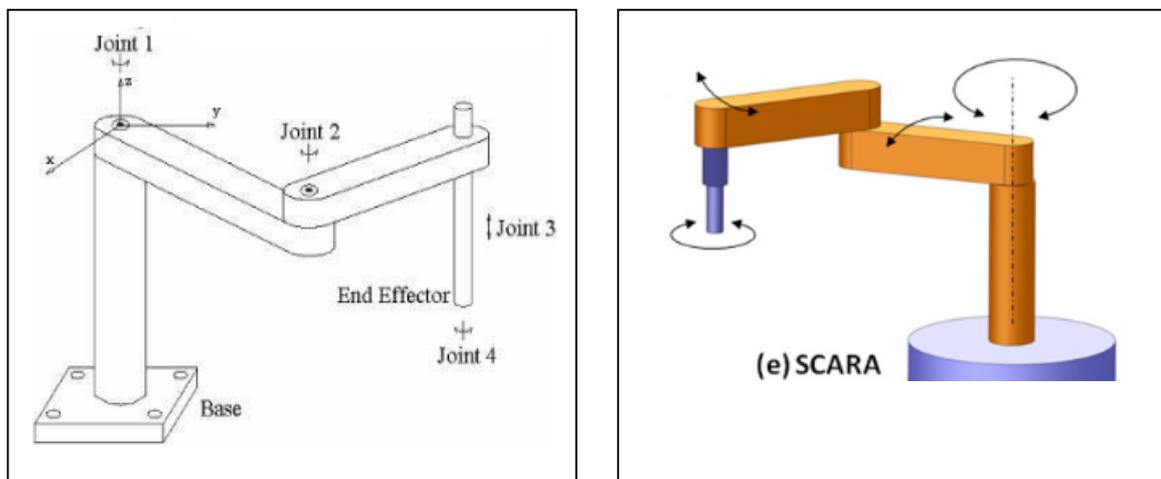


Figure I.8. Structure SCARA.

I.4.2.5 La structure anthropomorphe à 3R :

Elle est la plus fréquente. Elle reproduit la structure d'un bras humain (3 rotations, une correspond à l'épaule, les deux autres aux coude et le poignet).

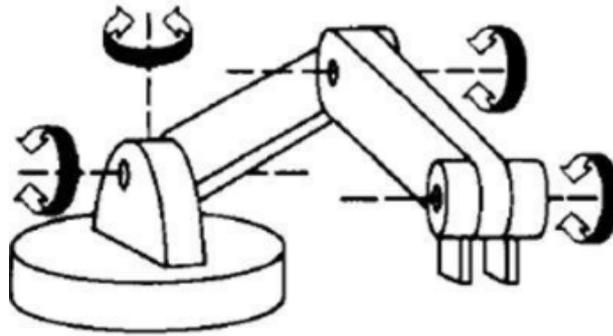


Figure I.9. Structure anthropomorphe.

I.4.3 Les actionneurs :

I.4.3.1 Motorisation des articulations :

La structure mécanique qui constitue le corps du robot possède un certain nombre de degrés de liberté, en générale de trois à six en considérant le poignet. Chacune de ces possibilités de mouvement correspond à une motorisation asservie [1] .

Si les trois types d'énergies (électrique, hydraulique et, dans une moindre mesure, pneumatique) sont présents sur le marché, force est de constater qu'actuellement la plupart des robots proposés par les constructeurs sont actionnement électrique.

I.4.3.2 Actionneurs électriques :

Les robots proposés par la plupart des constructeurs sont à actionnement électrique .Ils sont équipés essentiellement soit des moteurs classiques à courant continu, soit de plus en plus de moteurs autopilotés

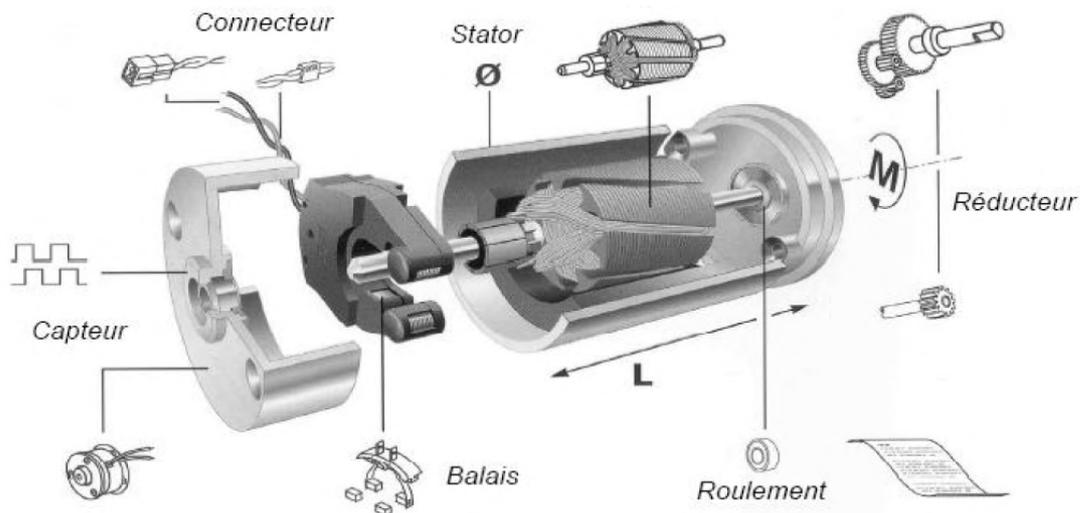


Figure I.10. Structure d'un actionneur électrique

I.4.3.2.1 Moteurs à courant continu :

Les moteurs utilisés sur les robots sont presque toujours à aimant permanents. Ils offrent le grand avantage de délivrer un couple proportionnel au courant qui traverse leur induit. La mesure et/ou le contrôle de ce courant peut servir de mesure et/ou de contrôle du couple moteur. D'autre part, la vitesse de rotation à vide est proportionnelle à la tension d'alimentation. Les moteurs ont rarement une puissance importante

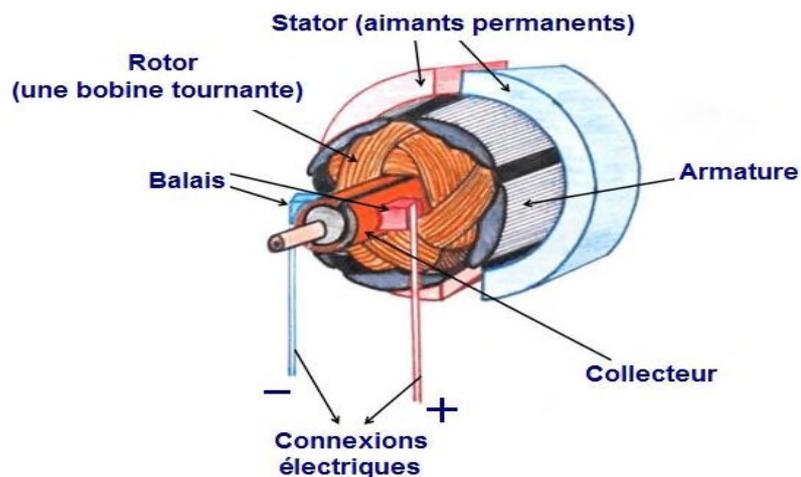


Figure I.11. Moteur à aimants permanents

I.4.3.2.2 Moteurs synchrones autopilotés :

Ces moteurs sont sans balais (moteurs brucheless) ou moteurs autopilotés. Ils présentent de nombreux avantages:

- Absence de collecteur, d'où une vitesse de rotation plus élevée ;
- Les enroulements supportés par le stator (échauffement réduit).

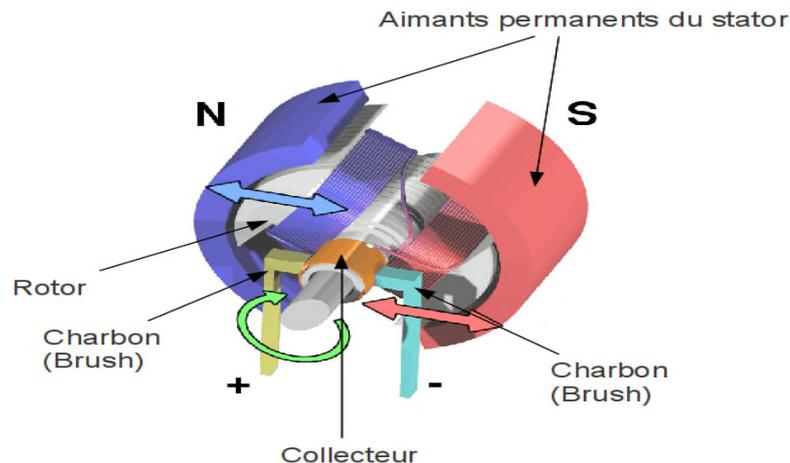


Figure I.12. Moteur synchrone autopiloté.

I.4.3.2.3 Les moteurs pas à pas :

les moteurs pas à pas, qui avaient suscité un grand enthousiasme lors de leur apparition sur le marché de fait de leur commande numérique sans asservissement, ont déçu les constructeurs de robots (difficulté de commande, masse prohibitive, imprécision...)et sont pratiquement abandonnés dans les applications robotiques manufacturières.

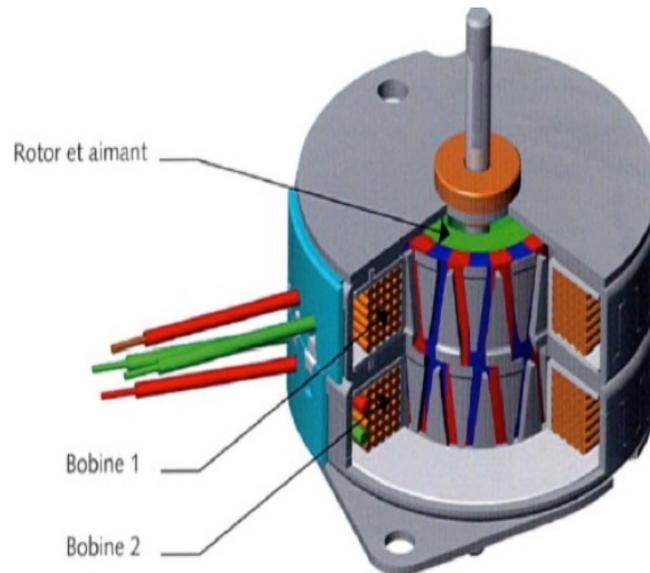


Figure I.13.Moteur pas-à-pas.

I.4.3.3 Actionneurs hydrauliques :

Ce type d'actionneurs est réservé à la robotique lourde. Notons toutefois, que certaines applications de la robotique de précision font à nouveau appel à ce type de technologie.

Parmi leurs avantages, on trouve un rapport puissance/poids élevé et des constantes de temps très faibles permettant d'excellentes performances en vitesse et accélération. Parmi ses inconvénients, on trouve le coût élevé, le danger de fuite de l'huile et l'encombrement.



Figure I.14.Actionneur hydraulique.

I.4.3.4 Actionneurs pneumatique :

Ils utilisent, comme énergie, l'air sous pression. Leur course est souvent limitée. Les outils ou les effecteurs pneumatiques utilisent la compressibilité de l'air pour adoucir la prise des objets et ajouter une certaine souplesse dans la manipulation.

I.4.4 Les transmetteurs :

Ils sont des dispositifs mécaniques permettant de transmettre un mouvement d'une pièce à une autre (actionneur vers articulations). Ils assurent aussi l'adaptation du mouvement tel que la réduction de la vitesse, le changement de direction et la transformation du mouvement.

Il existe beaucoup de modèles de transmissions, mais en robotique on s'intéresse aux transmissions à engrenages, à courroie et à roue de friction.



Figure I.15. Types de transmetteurs.

I.4.5 Les capteurs :

Un capteur est un dispositif transformant l'état d'une grandeur physique observée en une grandeur électrique. En robotique, on trouve deux types de capteurs :

I.4.5.1 Les capteurs proprioceptifs:

Ils permettent d'identifier l'état du robot à un instant donné (e.g., positions, vitesses et l'accélération des articulations).

I.4.5.2 Les capteurs extéroceptifs :

Ils permettent d'identifier l'environnement du robot à un instant donné (caméras, télémètres,...).

I.4.6 Les outils :

Selon leurs tailles, on peut les répartir en trois classes : lourds de grande taille (les pinces à soudeuse par point), de moyenne et petite taille (qui assurent quelques tâches comme le perçage de pièces et le découpage) et, en dernier, les outils à origine manuelle comme les visseuses.

I.5 Différentes générations de robots

Les robots évoluent avec les progrès qui s'opèrent dans la mécanique, l'informatique, l'énergétique et les capteurs-actionneurs. Il existe trois classes de robots :

I.5.1 Les robots passifs

Ils sont capables d'exécuter des tâches complexes, mais d'une manière répétitive et lourde. Ces robots n'adaptent avec les modifications dans l'environnement, mais leurs précision est faible.

I.5.2 Les robots actifs :

Ils sont capables de reconnaître leur environnement et d'éviter des obstacles. Leur auto-adaptative est très élevée.

I.5.3 Les robots intelligents

Ils exploitent les avancées réalisées dans le domaine de l'intelligence artificielle pour développer la capacité d'établir une stratégie afin d'exécuter une tâche ou effectuer un déplacement.

I.6 Caractéristiques des robots

Un robot doit être choisi en fonction de l'application qu'on lui réserve. Parmi les paramètres à prendre, éventuellement [11], 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 système mécanique articulé : le choix est guidé par la tâche à réaliser (quelle est la rigidité de la structure ?).

- Le volume de travail : défini comme l'ensemble des points atteignables par l'organe terminal. Tous les mouvements ne sont pas possibles en tous points du volume de travail. L'espace de travail (*reachable workspace*), également appelé espace de travail maximal, 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.

- Le 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 et à la flexibilité du système mécanique. En général, l'erreur de positionnement absolu, également appelée précision, est de l'ordre de 1 *mm*.

- La répétabilité : ce paramètre caractérise la capacité du robot 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

I.7 Domaines d'application

La robotique possède de nombreux domaines d'application. À l'origine, les robots ont été installés dans les industries pour réaliser des tâches répétitives avec une précision constante.

I.6.1 L'industrie :

Les robots sont intensivement utilisés dans l'industrie où ils effectuent sans relâche des tâches répétitives et avec rigueur. Dans les chaînes de montage de l'industrie automobile, ils y remplacent les ouvriers dans les tâches pénibles et dangereuses (peinture, soudage, etc.)



Figure I.16. Robots industriels.

I.6.2 Exploration spatiale

Les scientifiques ont pris le pari d'envoyer des robots pour explorer le système solaire, dans des environnements souvent mortels pour l'homme.

I.6.3 Militaire :

Un robot militaire est un robot autonome ou contrôlé à distance, conçu pour des applications militaires. Les drones sont une sous-classe des robots militaires.

Il existe d'autres domaines d'application des robots comme la médecine (les robots médicaux), dessin et pour l'utilisation domestique.

I.8 Conclusion :

Dans ce chapitre, nous avons présenté des généralités sur les bras de robot, leur utilisation, leurs constituants et quelques définitions de certains termes. Nous avons énuméré les différentes structures des bras de robots et les types d'actionneurs. Chaque structure possède des caractéristiques qui lui permettent de réaliser une tâche spécifiée mieux que d'autres. On a vu que l'utilisation des robots n'est pas limité seulement à l'industrie mais elle est étendue à d'autres domaines.

Chapitre II :

Modélisation des bras de robots

II.1 Introduction

La manipulation robotique, par définition, implique que les pièces et les outils soient déplacés dans l'espace par une sorte de mécanisme. Cela conduit naturellement à un besoin de représentation des positions et des orientations des parties, des outils et du mécanisme lui-même. Pour définir et manipuler des quantités mathématiques qui représentent la position et l'orientation, nous devons définir des systèmes de coordonnées et élaborer des conventions pour la représentation. Beaucoup d'idées développées dans le contexte de la position et de l'orientation serviront de base à l'étude des vitesses, des forces et des couples linéaires et rotatifs. Nous adoptons le principe d'un système de coordonnées universel auquel tout ce dont nous discutons peut être référencé[4].

II.2 Modélisation géométrique

II.2.1 Représentation d'un point dans l'espace:

Soit P un point de coordonnées cartésiennes p_x, p_y, p_z (Figure II.1). Les coordonnées homogènes du point P sont les termes $w \times p_x, w \times p_y$ et $w \times p_z$; où w est un facteur d'échelle souvent égal 1 en robotique [5]. On présente alors les coordonnées homogènes d'un point par le vecteur :

$$P = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$$

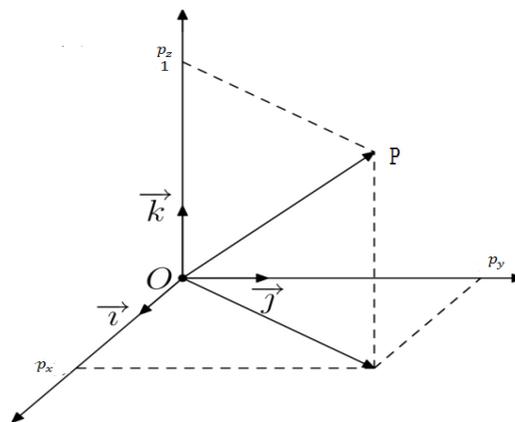


Figure II.1 : Représentation d'un point dans l'espace

II.2.2 Matrices de transformation homogènes

Le modèle géométrique directe (MGD) est l'ensemble des relations qui permettent d'exprimer la situation de l'organe terminal, c'est-à-dire les coordonnées opérationnelle du robot, en fonction de ses coordonnées articulaires[1].

La représentation de la position et de l'orientation à la fois dans un repère peut se faire avec les matrices de transformations homogènes. Toutes les manipulations, telle que le passage d'un repère à un autre, peut se faire par une simple multiplication de ces matrices. Une matrice de transformation homogène a la forme :

$$T = \begin{bmatrix} \mathbf{R}_{3 \times 3} & \mathbf{P}_{3 \times 1} \\ \mathbf{N}_{1 \times 3} & \mathbf{1} \end{bmatrix}$$

Avec $\mathbf{R}_{3 \times 3}$ est la matrice de rotation représentant l'orientation, \mathbf{P} est le vecteur colonne de position et $\mathbf{N} = [0 \ 0 \ 0]$ est un vecteur ligne généralement nul, mais qui peut être utilisé pour la perspective en vision artificielle.

Matrices de rotation :

Soient deux repère A et B dont les origines coïncident ($\mathbf{O}_0 = \mathbf{O}_1$). L'orientation du repère B par rapport au repère A peut être représentée par le produit de trois matrices élémentaires représentant les rotations autour des axes X, Y et Z [5]:

➤ Rotation d'un angle θ_1 autour de X :

$$\text{Rot}(X, \theta_1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta_1 & -\sin \theta_1 & 0 \\ 0 & \sin \theta_1 & \cos \theta_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

➤ Rotation d'un angle θ_2 autour de Y :

$$\text{Rot}(Y, \theta_2) = \begin{bmatrix} \cos \theta_2 & 0 & \sin \theta_2 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta_2 & 0 & \cos \theta_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

➤ Rotation d'un angle θ_3 autour de Z:

$$\text{Rot}(Z, \theta_3) = \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & 0 \\ \sin \theta_3 & \cos \theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matrice de translation :

Soient deux repère A et B et (p_x, p_y, p_z) les coordonnées de l'origine du repère B exprimées dans le repère A. Lorsque les axes des deux repères sont en parallèle (même orientation), alors la matrice A est appelée "matrice de translation" elle est donnée par :

$$T_{\text{Trans}} = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

II.2.3 Convention de Denavit-Hartenberg modifiée pour le calcul du MGD :

La technique la plus répandue pour décrire la géométrie d'un bras manipulateur Consiste à utiliser les paramètres de Denavit-hartenberg modifiée.

Le repère R_j est placé au début de chaque corps C_j .

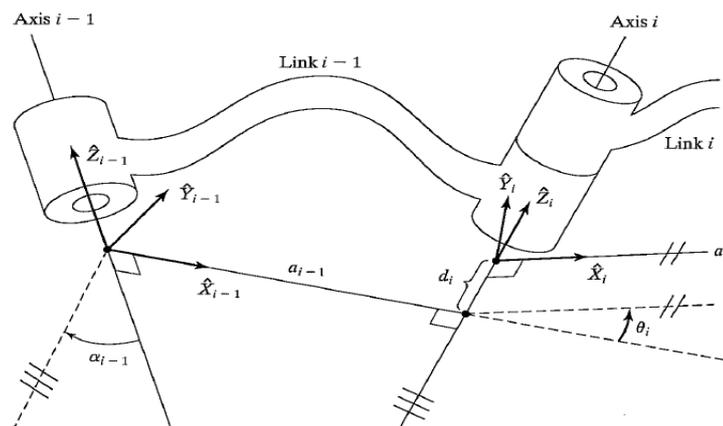


Figure II.2: Paramètres géométriques

Le passage du repère R_{j-1} au repère R_j , s'exprime en fonction de 4 transformations :

1. Rotation (x_{j-1}, α_{j-1})
2. Translation (x_{j-1}, a_{j-1})
3. Rotation (z_j, θ_j)
4. Translation (z_j, d_j)

Avec:

α_{j-1} : est l'angle entre les axes z_{j-1} et z_j , autour de x_{j-1}

a_{j-1} : est la distance entre les axes z_{j-1} et z_j le long de x_{j-1}

θ_j : est l'angle entre les axes x_{j-1} et x_j , autour de z_j

d_j : est la distance entre les axes x_{j-1} et x_j , le long de z_j

La variable articulaire q_j (associée à la $j^{\text{ème}}$ articulation) est soit :

- θ_j si l'articulation est de type rotoïde ($\sigma_j = 0$)
- r_j si l'articulation est de type prismatique ($\sigma_j = 1$)

On adonc $q_j = \bar{\sigma}_j \cdot \theta_j + \sigma_j \cdot r_j$

La matrice de transformation homogène définissant le repère R_j dans le repère

R_{j-1} est donnée par :

$$T_{j-1}^i = Rot(x, \alpha_{j-1}) \cdot Trans(x, a_{j-1}) \cdot Rot(z, \theta_j) \cdot Trans(z, d_j)$$

Soit

$$T_{j-1}^i = \begin{bmatrix} C\theta_j & -S\theta_j & 0 & -a_{j-1} \\ -C\alpha_{j-1} \cdot S\theta_j & C\alpha_{j-1} \cdot C\theta_j & -S\alpha_{j-1} & -d_j \cdot S\alpha_{j-1} \\ S\alpha_{j-1} \cdot S\theta_j & -S\alpha_{j-1} \cdot C\theta_j & C\alpha_{j-1} & d_j \cdot C\alpha_{j-1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

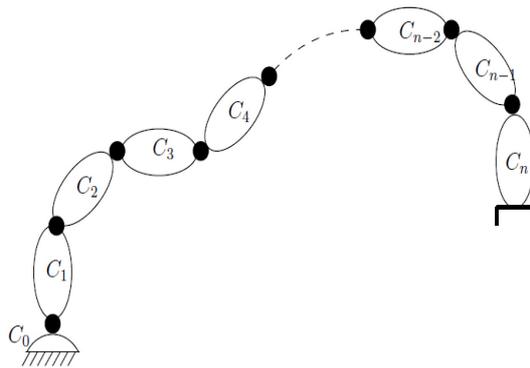


Figure II.3 : Robot à structure ouverte simple.

Le MGD d'un bras est obtenu par le produit des matrices :

$$T_0^n = T_0^1 \cdot T_1^2 \cdot \dots \cdot T_{n-1}^n \quad (\text{II.1})$$

II.2.4 Modèle géométrique inverse (MGI):

Le modèle géométrique direct d'un robot permet de calculer les coordonnées opérationnelles donnant la situation de l'organe terminal en fonction des coordonnées articulaires. Le problème inverse consiste à calculer les coordonnées articulaires correspondant à une situation donnée de l'organe terminal [6]. Le modèle s'écrit :

$$\theta = f^{-1}(x) \quad (\text{II.2})$$

La détermination du modèle géométrique inverse (MGI) est un problème complexe. Il y a rarement unicité de solution. On doit inverser un système d'équations non linéaires. Ce qui n'est pas trivial.

II.2.5 Calcul du modèle géométrique d'un bras de robot à deux d.d.l :

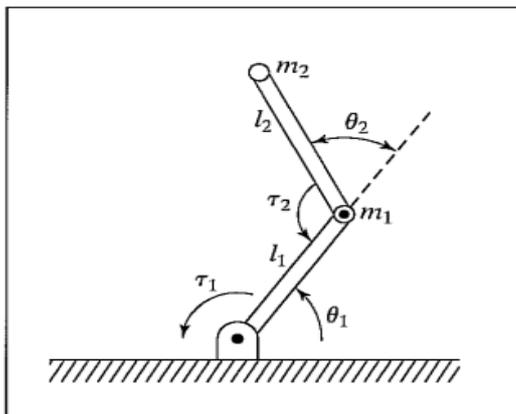


Figure II. 4: Modèle d'un bras de robot à deux ddl

Après l'implantation des repères au début de chaque corps, on calcule d'abord la table des paramètres de la convention de la convention de Denavit-hartenberg modifiée:

Tableau II.1 : Configuration de Denavit-Hartenberg modifiée du robot à 2ddl

Articulation	α_j	a_j	θ_j	d_j	σ_j
1	0	0	θ_1	0	0
2	0	L1	θ_2	0	0
E	0	L2	0	0	0

La matrice de transformation homogène de la première articulation est donné par :

$$T_0^1 = \begin{bmatrix} C\theta_1 & -S\theta_1 & 0 & 0 \\ S\theta_1 & C\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La matrice de transformation homogène de la deuxième articulation est donné par

$$T_1^2 = \begin{bmatrix} C\theta_2 & -S\theta_2 & 0 & L_1 \\ S\theta_2 & C\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Le MGD du robot est donné par :

$$T_0^E = T_0^1 T_1^2 T_2^E = T_0^2 T_2^E = \begin{bmatrix} C1C2 - S1S2 & -S1C1 - S1C2 & 0 & C1.L_1 \\ S1C2 + C1S2 & -S1S2 + C1C2 & 0 & S1.L_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & L_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

On utilise les relations trigonométriques on obtient :

$$T_0^E = \begin{bmatrix} C12 & -S1C1 - S1C2 & 0 & C1.L_1 + C12.L_2 \\ S12 & -S1S2 + C1C2 & 0 & S1.L_1 + S12.L_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Avec $C1 = \cos(\theta_1)$, $S1 = \sin(\theta_1)$ et $C12 = \cos(\theta_1 + \theta_2)$.

Après avoir calculé la matrice de transformation T_0^E on la utilise pour le chemin inverse afin de calculer les variables articulaires on la écrit sous la forme :

$$U = \begin{bmatrix} u_{11} & u_{12} & 0 & p_x \\ u_{21} & u_{22} & 0 & p_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

On calcule θ_1 et θ_2

$$\theta_1 = \arctang(\mathbf{p}_y - \mathbf{u}_{21} * l_2 / \mathbf{p}_x - \mathbf{u}_{11} * l_2)$$

$$\theta_2 = \arctan2(\mathbf{p}_y - \mathbf{S1} * l_2 / \mathbf{p}_x - \mathbf{C1} * l_2) - \theta_1$$

II.3. Modélisation cinématique

II.3.1 Modèle cinématique direct :

Le modèle cinématique direct (MDC) d'un robot manipulateur décrit les vitesses des coordonnées opérationnelles en fonction des vitesses articulaires [1]. Il est noté par :

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \quad (\text{II.3})$$

Où \mathbf{J} désigne la matrice jacobéenne de dimension $(\mathbf{m} \times \mathbf{n})$, égale à $\frac{\partial \mathbf{x}}{\partial \mathbf{q}}$ et fonction de la configuration articulaire. La même matrice jacobéenne intervient dans le calcul du modèle différentiel direct qui donne les variations élémentaires $d\mathbf{x}$ des coordonnées opérationnelles en fonction des variations $d\mathbf{q}$ soit :

$$d\mathbf{x} = \mathbf{J}(\mathbf{q})d\mathbf{q} \quad (\text{II.4})$$

II.3.2 Modèle cinématique inverse :

L'objectif du modèle cinématique inverse (MCI) est de calculer, à partir d'une configuration \mathbf{q} donnée, les vitesses articulaires $\dot{\mathbf{q}}$ qui assurent au repère terminal une vitesse opérationnelle $\dot{\mathbf{x}}$ imposée. On peut déterminer aussi la différentielle articulaire $d\mathbf{q}$ correspondant à une différentielle des coordonnées opérationnelles $d\mathbf{x}$. Dans le cas régulier d'une matrice Jacobéenne carrée et non singulière [5], le modèle cinématique inverse s'écrit :

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}\dot{\mathbf{x}} \quad (\text{II.5})$$

II.4 Modélisation dynamique

Le modèle dynamique est la relation entre les couples (et/ou force) appliquées aux actionneurs et les positions, vitesses et accélérations articulaires. On représente le modèle dynamique par une relation de la forme :

$$\boldsymbol{\tau} = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{f}_e) \quad (\text{II.6})$$

- $\boldsymbol{\tau}$: Vecteur des couples/forces articulaires ;
- \mathbf{q} : Vecteur des positions articulaires ;
- $\dot{\mathbf{q}}$: Vecteur des vitesses articulaires ;
- $\ddot{\mathbf{q}}$: Vecteur des accélérations articulaires ;
- \mathbf{f}_e : Vecteurs représentant l'effort extérieur (forces et moments) qu'exerce le robot sur l'environnement.

L'équation (II.6) correspond modèle dynamique inverse. Le modèle dynamique direct exprime les accélérations articulaires en fonction des positions, vitesses et couples des articulations. Il est alors représenté par la relation : $\boldsymbol{\tau}$

$$\ddot{\mathbf{q}} = \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\tau}, \mathbf{f}_e) \quad (\text{II.7})$$

Parmi les applications du modèle dynamique, on peut citer :

- La simulation qui utilise le modèle dynamique direct ;
- Le dimensionnement des actionneurs ;
- L'identification des paramètres inertiels et des paramètres de frottement du robot.

II.4.1. Formalisme de Lagrange [6]:

Nous considérerons un robot idéal sans frottement, sans élasticité et en subissant ou n'exerçant aucun effort extérieur. Le formalisme de Lagrange décrit les équations du mouvement, lorsque l'effort extérieur sur l'organe terminal est supposé nul :

$$\boldsymbol{\tau}_i = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} \quad (\text{II.8})$$

avec :

- L : lagrangien du système, $E = E - U$;
- E : énergie cinétique totale du système ;
- U : énergie potentielle totale du système.

II.4.2 Formalisme de Newton-Euler :

Afin de déplacer les corps, on a besoin de les accélérer et de les décélérer. Les forces requises pour de tels mouvements sont des fonctions des accélérations et la distribution de la masse (centre de masse, tenseur d'inertie et la masse). Le lien entre les forces, les inerties et les accélérations est exprimé par l'équation de Newton et celle d'Euler.

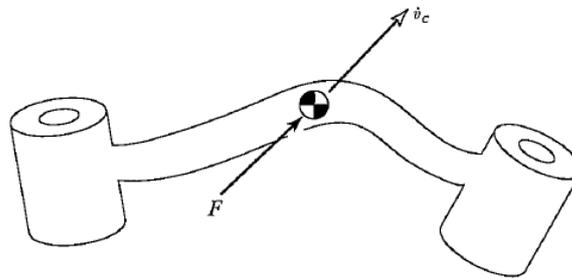


Figure II.5: Une force F agissant au centre de la masse d'un corps

- L'équation de Newton exprime la force appliquée au centre de masse d'un corps et l'accélération induite :

$$F_i = m \cdot \dot{v}_{Ci} \quad (\text{II.9})$$

- L'équation d'Euler exprime le moment qui doit être appliqué au corps pour le mettre en mouvement de rotation de vitesse w_i et d'accélération \dot{w}_i :

$$N_i = I_i \cdot \dot{w}_i + w_i * (I_i \cdot w_i) \quad (\text{II.10})$$

Pour ce faire, on calcule d'abord les accélérations \dot{v}_{Ci} et \dot{w}_i et les vitesses w_i d'une manière itérative, en commençant de la base et qui se termine au poignet. Ces itérations sont connus par les itérations extérieures (pour plus de détails, consulter l'annexe .1).

Le corps C_i est soumis aussi à des forces et des moments intérieurs (f_i^i, n_i^i) qu'on peut définir par :

$$f_i^i = R_{i+1}^i f_{i+1}^{i+1} + F_i^i \quad (\text{II.11})$$

$$n_i^i = N_i^i + R_{i+1}^i n_{i+1}^{i+1} + P_{C_i}^i F_i^i + P_{i+1}^i R_{i+1}^i f_{i+1}^{i+1} \quad (\text{II.12})$$

Avec :

R_{i+1}^i :Matrice de rotation ;

P_{i+1}^i :Vecteur aligné le long du corps C_i (rayon de rotation) indiquant l'origine du repère R_i par rapport au repère R_{i+1}

On calcule les forces et les moments en utilisant des itérations qui commencent du poignet vers la base. On les appelle les itérations intérieures.

On prend l'axe Z comme axe de chaque articulation et la fin chaque itération on calcule le moment

$$\tau_i = n_i^{iT} Z_i^i \quad (\text{II.13})$$

Pour une articulation prismatique : $\tau_i = f_i^{iT} Z_i^i \quad (\text{II.14})$

II.4.3. Modèle dynamique d'un bras de robot à 2 d.d.l :

On prend le bras de robot de *Figure II.4*, ce robot a deux corps c_1, c_2 avec m_1, m_2 les masses des corps, et l_1, l_2 les longueurs des corps.

Les vecteurs du centre de masse

$$\begin{aligned} P_{c1}^1 &= l_1 X_1 \\ P_{c2}^2 &= l_2 X_2 \end{aligned} \quad (\text{II.15})$$

En raison de l'hypothèse de la masse ponctuelle, le tenseur d'inertie au centre de masse pour chaque liaison est défini par une matrice nulle :

$$\begin{aligned} I_{c1}^1 &= 0 \\ I_{c2}^2 &= 0 \end{aligned} \quad (\text{II.16})$$

A l'effecteur, on n'a pas des forces et à la base juste la force de gravité (pas de vitesses de rotation). Ce qui donne :

$$\begin{aligned} w_0 &= 0, \dot{w}_0 = 0 \\ f_0 &= 0, n_0 = 0 \\ \dot{v}_0^0 &= g Y_0 \end{aligned} \quad (\text{II.17})$$

Après avoir passé par les itérations extérieures et intérieures on obtient (pour plus de détails consulter l'annexe.1)

$$\tau_1 = m_2 l_1^2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l_2 c_2 (2\ddot{\theta}_1 + \ddot{\theta}_2) + (m_1 + m_2) l_1^2 \ddot{\theta}_1 - m_2 l_1 l_2 c_2 \dot{\theta}_1^2$$

$$\begin{aligned}
& -2m_2l_1l_2s_2\dot{\theta}_1\dot{\theta}_2 + m_2l_2gc_{12} + (m_1 + m_2)l_1gc_1 \\
\tau_2 = & m_2l_1l_2c_2\ddot{\theta}_1 + m_2l_1l_2s_2\dot{\theta}_1^2 + m_2l_1^2(\ddot{\theta}_1 + \ddot{\theta}_2)
\end{aligned} \quad (\text{II.18})$$

On arrange ces deux équations sous forme vectorielle

$$\begin{aligned}
\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = & \begin{bmatrix} m_2l_2^2 + 2m_2l_1l_2C_2 + (m_1 + m_2)l_1^2 & m_2l_2^2 + m_2l_1l_2C_2 \\ m_2l_1l_2C_2 + m_2l_2^2 & m_2l_2^2 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \\
& \begin{bmatrix} -2m_2l_1l_2S_2\dot{\theta}_2 & m_2l_1l_2S_2\dot{\theta}_2 \\ 2m_2l_1l_2S_2\dot{\theta}_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} m_2l_2C_{12}g + (m_1 + m_2)l_1C_1g \\ m_2l_2C_{12}g \end{bmatrix} \quad (\text{II.19})
\end{aligned}$$

II.5. Génération de trajectoires

II.5.1 Méthode des polynômes cubiques [4] :

Considérez le problème de déplacer l'outil de sa position initiale à une position finale et ce durant un certain temps t_f . Le modèle géométrique inverse permet de calculer l'ensemble des angles articulaires qui correspondent à une posture désirée du poignet. La posture initiale du manipulateur est également connue sous la forme d'un ensemble d'angles des articulations. Ce qui est requis est une fonction pour chaque articulation dont la valeur à t_0 est la position initiale de l'articulation et dont la valeur à t_f est la position finale souhaitée. Comme le montre la *figure (II.6)*, il existe de nombreuses fonctions lisses, $\theta(t)$ qui peuvent être utilisées pour interpoler la valeur de l'articulation.

En faisant un seul mouvement lisse, au moins quatre contraintes sur $\theta(t)$ sont évidentes. Deux contraintes sur la valeur de la fonction proviennent de la sélection de la valeur initiale et valeur finale.

$$\begin{aligned}
\theta(0) &= \theta_0 \\
\theta(t_f) &= \theta_f
\end{aligned} \quad (\text{II.20})$$

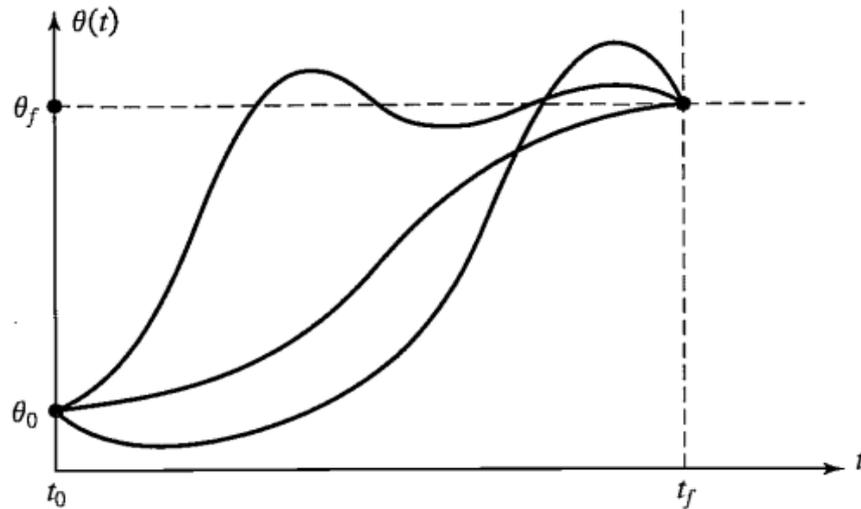


Figure II.6: Plusieurs formes de chemin possibles pour une seule articulation.

Deux autres contraintes sont que la fonction soit continue en vitesse, ce qui dans ce cas signifie que la vitesse initiale et finale sont nulles :

$$\dot{\theta}(0) = 0, \dot{\theta}(t_f) = 0 \quad (\text{II.21})$$

Ces quatre contraintes peuvent être satisfaites par un polynôme d'au moins un troisième degré (polynôme cubique à quatre paramètres) de la forme :

$$\theta(t) = a_0 + a_1 t^1 + a_2 t^2 + a_3 t^3 \quad (\text{II.22})$$

De sorte que la vitesse et l'accélération de l'articulation sur ce chemin sont claires

$$\begin{aligned} \dot{\theta}(t) &= a_1 + 2 a_2 t^1 + 3 a_3 t^2 \\ \ddot{\theta}(t) &= 2 a_2 + 6 a_3 t^2 \end{aligned} \quad (\text{II.23})$$

La combinaison (II.22) et (II.23) avec les contraintes de quatre désirs donne quatre équations en quatre inconnues

$$\begin{aligned} \theta(0) &= a_0, \quad 0 = a_1 \\ \theta_f &= a_0 + a_1 t_f^1 + a_2 t_f^2 + a_3 t_f^3 \\ 0 &= a_1 + 2 a_2 t_f + 3 a_3 t_f^2 \end{aligned} \quad (\text{II.24})$$

Résoudre les équations de a_i , pour l'obtention

$$\begin{aligned} a_0 &= \theta(0), \quad a_1 = 0 \\ a_2 &= \frac{3}{t_f^2} (\theta_f - \theta_0), \quad a_3 = -\frac{2}{t_f^3} (\theta_f - \theta_0) \end{aligned} \quad (\text{II.25})$$

En utilisant (II.25), on peut calculer le polynôme cubique qui relie toute position d'angle d'articulation initiale avec une position finale souhaitée. Cette solution est pour le cas où l'articulation commence et se termine à une vitesse nulle.

II.5.2 Planification de trajectoire pour un bras de robot a 2 ddl:

On planifie les trajectoires des corps d'un robot à deux ddl. La première articulation passe de la valeur initiale $\theta_{10}=0^\circ$ à la valeur finale $\theta_{1f}=10^\circ$ et le deuxième de $\theta_{20}=0^\circ$ à $\theta_{2f}=5^\circ$, et ce pendant $t_f=5\text{sec}$.

A fin de calculer les paramètres de polynôme cubique et de tracé les trajectoires on a crée une fonction Matlab (la fonction est donnée en annexe B.0)

Les paramètres des polynômes sont comme suit :

Première articulation : $a_0 = 0, a_1 = 0, a_2 = 0.133, a_3 = -0.0059$

Deuxième articulation : $a_0 = 0, a_1 = 0, a_2 = 0.0667, a_3 = -0.0030$

L'évolution des différentes grandeurs articulaires sont données par Figure II.7.

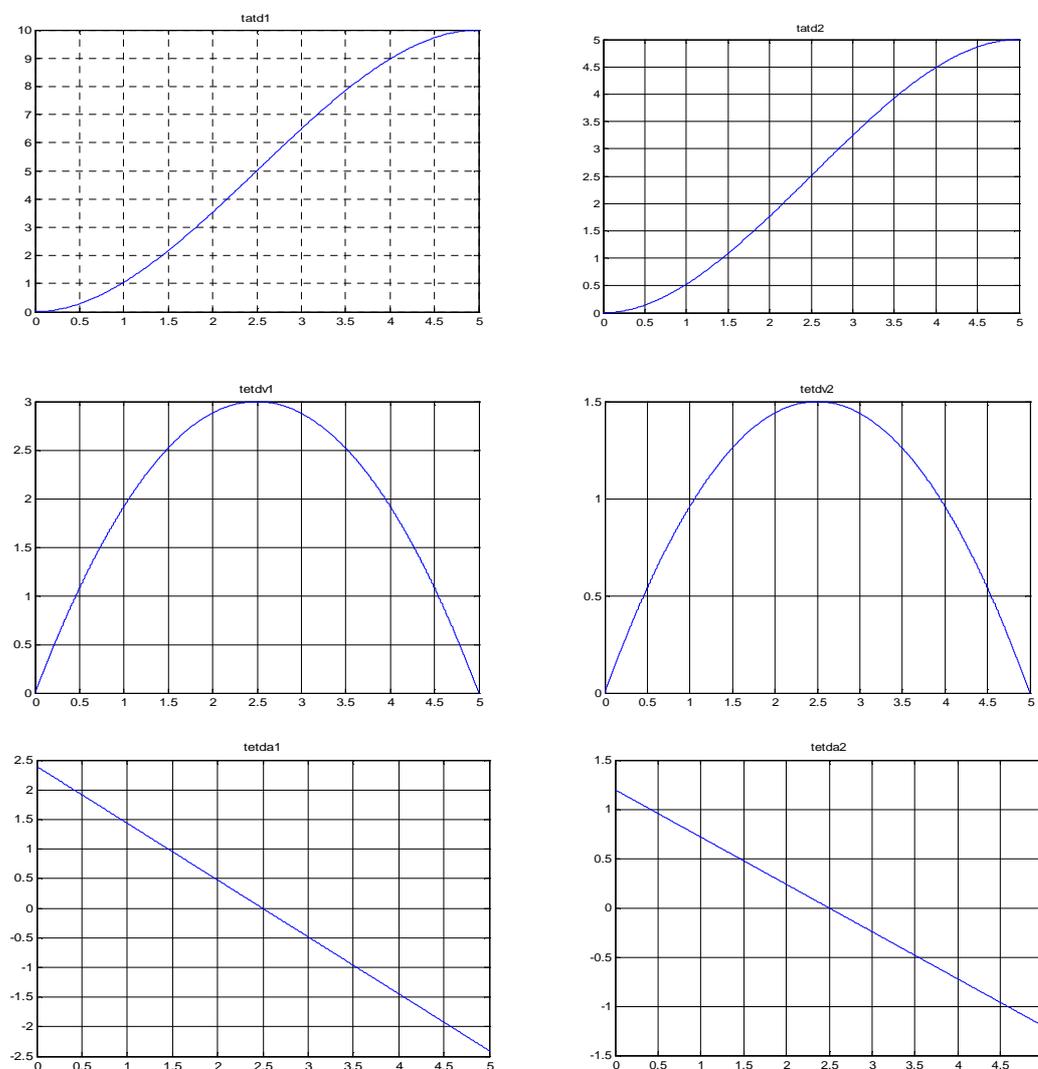


Figure II.7: Evolution des variables articulaires, de leurs vitesses et de leurs accélérations.

II.6. Conclusion

Dans ce chapitre, nous avons présenté la modélisation des bras de robots. Les trois modèles : géométrique, cinématique et dynamique (avec modèle direct et inverse pour tout les modèle), ont été exposés. Nous avons donné ensuite les modèles du robot étudié : le modèle géométrique directe, avec la convention *Denavit-Hartenberg modifiée*, et le modèle dynamique, en utilisant la méthode de *Newton-Euler*. Le problème de planification et de génération de trajectoires a été aussi évoqué.

Chapitre III :

Commande des bras de robots.

III.1 Introduction

Le but de la commande est de faire déplacer l'outil ou l'effectuer d'un point initial vers un point final, selon une trajectoire désirée. Dans ce qui suit, on utilise les techniques de commande linéaires et non linéaires, afin de commander un bras de robot à deux d.d.l.

III.2 Technique de commande des bras de robot :

III.2.1. Régulation linéaire

III.2.1.1 Régulateur proportionnel dérivé PD [7]:

Afin d'améliorer le fonctionnement d'un régulateur proportionnel, il peut élaborer une action non seulement en fonction de l'écart, mais aussi de la dérivée de cet écart par rapport au temps. Ainsi, une correction sur le système à commander est effectuée avant même que se manifeste un écart. Le régulateur **proportionnel dérivée** est régi par :

$$u(t) = Kp \left[e(t) + Td \frac{de}{dt}(t) \right] \quad (\text{III.1})$$

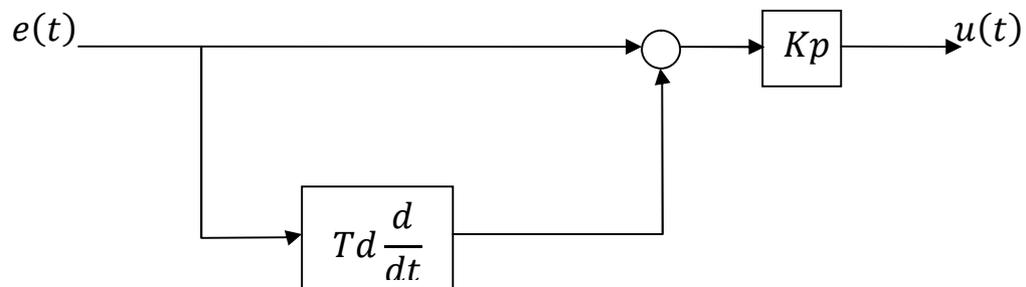


Figure III.1: Schéma d'un régulateur PD

Le robot peut être modélisé par un modèle de second ordre linéaire équivalent à l'équation :

$$a\ddot{x} + b\dot{x} + cx = f \quad (\text{III.2})$$

avec (a,b,c) des constantes et f est la force (ou couple) appliquée.

Dans la commande des bras de robots, on évite les réponses oscillatoires (pour éviter les chocs avec l'environnement). On utilise un facteur d'amortissement $\zeta=1$ qui correspond au début du régime aperiodique. On choisit une loi de commande proportionnelle -dérivée :

$$f = -k_p x - k_v \dot{x} \quad (\text{III.3})$$

On remplace (3.3) dans (3.2) :

$$\begin{aligned} a\ddot{x} + b\dot{x} + cx &= -k_p x - k_v \dot{x} \\ a\ddot{x} + (b + k_v)\dot{x} + (c + k_p)x &= 0 \\ a\ddot{x} + k'_v \dot{x} + k'_p x &= 0 \end{aligned} \quad (\text{III.4})$$

Cette équation peut être mise sous la forme standard :

$$\ddot{x} + 2\zeta\omega_n \dot{x} + \omega_n^2 x = 0$$

Pour $\zeta=1$, il reste à fixer la valeur de la pulsation naturelle, ω_n , et de calculer K_v et K_p .

III.2.1.2 Régulateur PID [8] :

La régulation P.I.D (proportionnelle, intégrale, dérivée) est sans doute la plus connue car la plus ancienne et la plus répandue dans l'industrie.

Les procédés sont souvent perturbés. La grandeur X sur Figure III.2 symbolise la perturbation principale que l'on veut prendre en compte : c'est souvent une difficulté dans la mesure où les perturbations sont rarement mesurables. Ce qui conduit à des erreurs statiques non nulles.

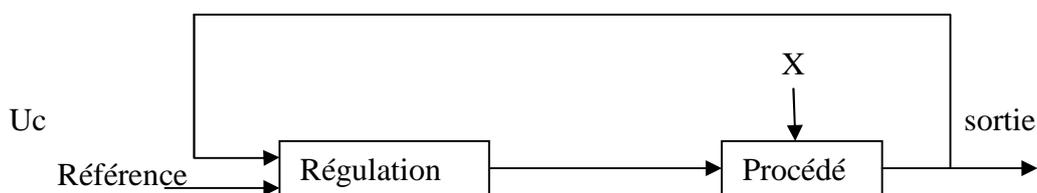


Figure III.2 : Système bouclé en présence de perturbations

De ce fait, une loi de commande PID est plus indiquée. L'ajout de l'action intégrale permet d'améliorer la précision statique. La loi de commande devient :

$$u(t) = Kp \left[e(t) + Td \frac{de}{dt}(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau \right] \quad (\text{III.5})$$

III.2.2 Commande et linéarisation :

Dans le cas d'un modèle non linéaire

$$a\ddot{x} + h(\dot{x}) + g(x) = f \quad (\text{III.6})$$

Une loi de commande linéarisante peut être écrite sous la forme :

$$f = \alpha f' + \beta \quad (\text{III.7})$$

On choisit les paramètres comme suit :

$$\alpha = a \quad \text{et} \quad \beta = h(\dot{x}) + g(x),$$

On aboutit à : $f' = \ddot{x}$

On choisit :

$$f' = -k_p x - k_v \dot{x} \quad (\text{III.8})$$

On remplace (III.7) dans (III.6):

$$\begin{aligned} a\ddot{x} + b\dot{x} + cx &= \alpha(-k_p x - k_v \dot{x}) + \beta \\ a\ddot{x} + b\dot{x} + cx &= a(-k_p x - k_v \dot{x}) + b\dot{x} + cx \\ \ddot{x} &= (-k_p x - k_v \dot{x}) \\ \ddot{x} + k_p x + k_v \dot{x} &= 0 \end{aligned} \quad (\text{III.9})$$

On parle de la linéarisation lorsqu'on a un système non linéaire. Son principe consiste à compenser les termes non linéaires par des termes équivalents.

III.2.3 Commande par découplage non linéaire[9] :

Ce type de commande, qui concerne les systèmes non linéaires et multi-variables, est aussi connu sous le nom de *commande dynamique* ou *couple calculé* (computed torque) dans la littérature anglo-saxonne, parce qu'elle est fondée sur l'utilisation du modèle dynamique. Théoriquement, elle assure **le découplage et la linéarisation** des équations du modèle, ayant pour effet une réponse uniforme quelle que soit la configuration du robot.

La commande par découplage non linéaire consiste à transformer par retour d'état, le problème de commande d'un système non linéaire en un problème de commande d'un système linéaire.

Dans le cas des bras de robots à plusieurs articulations, le principe consiste à commander chaque articulation séparément. Soit le modèle du bras décrit par :

$$\tau = M(q) \cdot \ddot{q} + V(q, \dot{q}) + G(q) \quad (\text{III.10})$$

On choisit la loi de commande :

$$\tau = \alpha \tau' + \beta \quad (\text{III.11})$$

Avec : $\alpha = \widehat{M}(q)$ et $\beta = \widehat{V}(q, \dot{q}) + \widehat{G}(q)$

et $(\widehat{M}, \widehat{V}, \widehat{G})$ sont les paramètres estimés du bras. Pour une régulation PID

$$\tau' = \ddot{q}_d + k_v[\dot{q}_d - \dot{q}] + k_p[q_d - q] + k_i \int [q_d(t) - q(t)] dt \quad (\text{III.12})$$

En remplaçant (III.12) et (III.11) dans (III.10), on aboutit à l'équation d'erreur suivante :

$$\ddot{E} + k_v \dot{E} + k_p E + k_i \int E(t) dt = 0 \quad (\text{III.13})$$

avec $E = [q_d - q]$. Le choix adéquat des paramètres (K_p, K_v, K_i) du régulateur permet d'assurer une bonne convergence de cette erreur vers zéro.

III.2.4 Poursuite de trajectoire :

Dans le cas où le système est appelé à poursuivre une trajectoire $(\theta_d, \dot{\theta}_d, \ddot{\theta}_d)$, générée par un module de génération de trajectoire (MGT), on cherche à minimiser l'erreur de poursuite définie comme suit :

$$e = \theta_d - \theta$$

Reprenons le modèle non linéaire (III.6) et la loi de commande (III.7), on écrit alors :

$$\tau = \alpha \tau' + \beta \quad \text{avec } \tau' = \ddot{\theta}_d + k_v \dot{e} + k_p e \quad (\text{III.14})$$

Ce qui conduit à l'équation d'erreur de poursuite :

$$\begin{aligned} \ddot{\theta} &= \ddot{\theta}_d + k_v \dot{e} + k_p e \\ \ddot{e} + k_v \dot{e} + k_p e &= 0 \end{aligned} \quad (\text{III.15})$$

Qui peut être mise sous la forme standard

$$\ddot{e} + 2\zeta \omega_n \dot{e} + \omega_n^2 e = 0$$

Pour $\zeta=1$ et choisissant la pulsation naturelle ω_n , on calcule : $k_v = 2\zeta \omega_n$ et $k_p = \omega_n^2$.

III.3 Modèle dynamique du robot utilisé

Le bras de robot étudié est à deux degrés de libertés avec des articulations rotoïdes. Il est composé d'une base, de deux corps de même longueur ($l=0.5m$) ayant deux masses différentes ($m_1=4.7k_g$ et $m_2=2.3k_g$) et deux actionneurs .

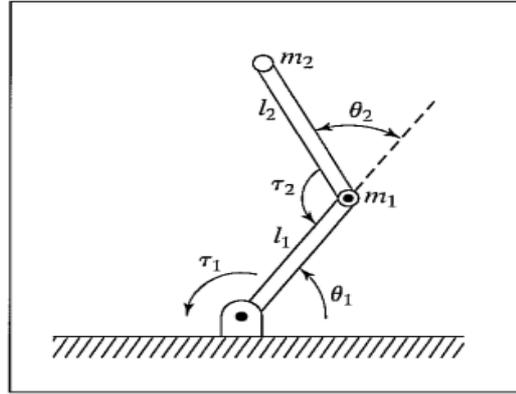


Figure III.3 : Structure du bras de robot étudié.

Modèle dynamique de bras :

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} m_2 l_2^2 + 2m_2 l_1 l_2 C_2 + (m_1 + m_2) l_1^2 & m_2 l_2^2 + m_2 l_1 l_2 C_2 \\ m_2 l_1 l_2 C_2 + m_2 l_2^2 & m_2 l_2^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} m_2 l_1 l_2 S_2 \dot{\theta}_2^2 - 2m_2 l_1 l_2 S_2 \dot{\theta}_2 \dot{\theta}_1 \\ 2m_2 l_1 l_2 S_2 \dot{\theta}_1 \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} m_2 l_2 C_{12} g + (m_1 + m_2) l_1 C_1 g \\ m_2 l_2 C_{12} g \end{bmatrix} \quad (III.16)$$

Avec $C_2 = \cos(\theta_2)$, $C_{12} = \cos(\theta_1 + \theta_2)$ et $S_2 = \sin(\theta_2)$. On remplace m_1, m_2, l_1 et l_2 par leurs valeurs, on aboutit à:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} 2.3 + 1.15C_2 & 0.575 + 0.575C_2 \\ 0.575C_2 + 0.575 & 0.575 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 0.575S_2 \dot{\theta}_2^2 - 1.15S_2 \dot{\theta}_2 \dot{\theta}_1 \\ 1.15S_2 \dot{\theta}_1 \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 11.5C_{12} + 34.5C_1 \\ 11.5C_{12} \end{bmatrix} \quad (III.17)$$

L'équation (III.17) correspond au modèle dynamique inverse du bras, du fait qu'il donne le vecteur de commande en fonction de l'accélération. Le modèle direct est donné par :

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} 2.3 + 1.15C_2 & 0.575 + 0.575C_2 \\ 0.575C_2 + 0.575 & 0.575 \end{bmatrix}^{-1} *$$

$$\begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} = \begin{bmatrix} 0.575S_2\dot{\theta}_2^2 - 1.15S_2\dot{\theta}_2\dot{\theta}_1 \\ 1.15S_2\dot{\theta}_1\dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 11.5C_{12} + 34.5C_1 \\ 11.5C_{12} \end{bmatrix} \quad (\text{III.18})$$

III.4 Commande en régulation du bras de robot

III.4.1 Commande PD :

Cette commande est basée sur l'hypothèse d'un bras de robot modélisé par un modèle simple. Dans certaines conditions de fonctionnement à faibles vitesses avec des constituants légers, le modèle de la commande peut être réduit à une forme très simple :

$$\boldsymbol{\tau} = \mathbf{M}\ddot{\boldsymbol{\theta}} \quad (\text{III.19})$$

Avec \mathbf{M} est une matrice diagonale constante, $\boldsymbol{\tau}$ et $\ddot{\boldsymbol{\theta}}$ des vecteurs colonnes .

Ce qui correspond à un modèle linéarisé autour de point d'un fonctionnement complètement découplé. Chaque articulation est supposée indépendante est les termes des forces centrifuges et Coriolis $\mathbf{V}(\dot{\boldsymbol{\theta}}, \boldsymbol{\theta})$ et les forces gravitationnelles $\mathbf{G}(\boldsymbol{\theta})$ sont complètement négligés.

Structure du système de commande :

Dans ce cas, le principe consiste à ramener le bras d'un état initial, θ_0 , vers un état final désiré, θ_d . La loi de commande est basée sur l'écart entre l'état instantané et l'état final (constant):

$$\mathbf{E} = [\theta_d - \theta(t)] \quad (\text{III.20})$$

Par conséquent, la variation d'erreur : $\dot{\mathbf{E}} = -\dot{\boldsymbol{\theta}}(t)$

Les profils des vitesses et des accélérations angulaires ne sont pas imposés. Ils dépendent de la loi de commande. Les couples requis pour les articulations sont calculés à base de la loi PD. Pour la simulation, les trajectoires réelles du robot sont calculées en utilisant son modèle dynamique direct.

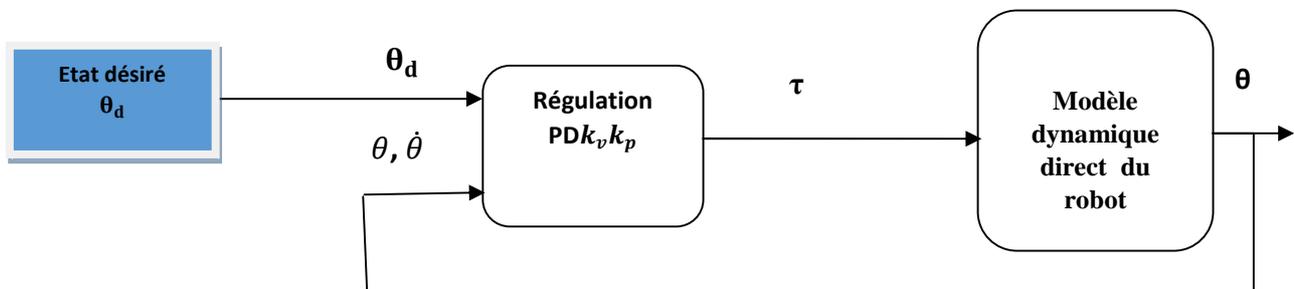


Figure III.4 : Schéma synoptique de simulation de la commande PD du bras de robot.

Le modèle de commande est défini par (III.19). En utilisant une loi de commande PD :

$$\boldsymbol{\tau} = \mathbf{kp} * \mathbf{e} + \mathbf{kv} * \dot{\mathbf{e}} \quad (\text{III.21})$$

Avec : $\boldsymbol{\tau}$, \mathbf{e} , et $\dot{\mathbf{e}}$ des vecteurs colonnes et \mathbf{kv} , \mathbf{kp} des vecteurs lignes et $\mathbf{e} = [\boldsymbol{\theta}_d - \boldsymbol{\theta}(t)]$.

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} kv_1 & kv_2 \end{bmatrix} \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \end{bmatrix} + \begin{bmatrix} kp_1 & kp_2 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \quad (\text{III.22})$$

Ce qui revient à :

$$\tau_1 = kv_1 \dot{e}_1 + kp_1 e_1$$

$$\tau_2 = kv_2 \dot{e}_2 + kp_2 e_2$$

Pour $\zeta=1$ et en choisissant une pulsation $\omega_n=25$, on calcule : $kv = 50$ et $kp = 625$.

$$\begin{aligned} \tau_1 &= 50(\dot{\theta}_{d1} - \dot{\theta}_1) + 625(\theta_{d1} - \theta_1) \\ \tau_2 &= 50(\dot{\theta}_{d2} - \dot{\theta}_2) + 625(\theta_{d2} - \theta_2) \end{aligned} \quad (\text{III.23})$$

Simulation et résultats :

Bien que la synthèse de la commande soit faite à base d'un modèle linéaire découplé simple (III.19), le robot a été simulé en utilisant le modèle non linéaire couplé et variant dans le temps de (III.18) qui est plus proche de la réalité.

Le robot est simulé en calculant l'accélération $\ddot{\boldsymbol{\theta}}$ avec le modèle dynamique direct (III.18), ensuite le calcul $\dot{\boldsymbol{\theta}}$ et $\boldsymbol{\theta}$ par intégration de $\ddot{\boldsymbol{\theta}}$ avec la méthode d'Euler. Selon la dynamique du bras de robot, on a choisi une période d'échantillonnage $T_e=0.001$.

Le programme Matlab(m.fille) donné en annexe.(B.1). Les résultats de simulation sont illustrés par Figure III.5

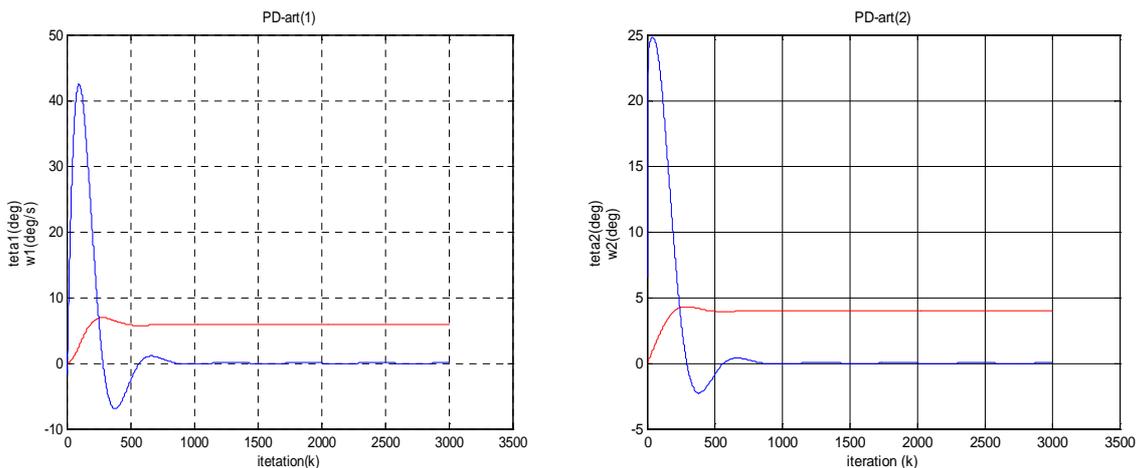


Figure III.5 : Formes des variables articulaires (en rouge) et des vitesses angulaires (en bleu) pour la 1^{ere} et la 2^{eme} articulation.

D'après les résultats de la deuxième articulation, on constate que l'évolution de l'angle $\theta_2(t)$ converge vers une valeur finale proche de la valeur finale désirée θ_{d2} en un temps court. Mais, pour la première articulation $\theta_1(t)$ converge vers une valeur finale qui n'est pas proche de celle désirée θ_{d1} . Ceci est dû à l'hypothèse très simplificatrice sur le modèle du robot utilisé pour la synthèse de la loi de commande PD (linéarité et forces gravitationnelles, centrifuges et Coriolis négligées). Un ajustement des paramètres de régulateur aboutit à la convergence vers l'angle désiré. Néanmoins le profil de vitesse n'est pas imposé. Ce qui peut conduire à des pics inadmissibles.

III.4.2 Commande PID :

Pour élaborer la commande PID classique, il on considère aussi l'hypothèse d'un modèle linéaire découplé. Chaque articulation peut être modélisée par :

$$\tau_i = m_{ii}\ddot{\theta}_i \quad (\text{III.24})$$

Lors de la partie précédente, on a supposé que le robot n'est pas soumis à des perturbations. Mais, dans la pratique, des perturbations telles que les frottements affectent le comportement dynamique du robot. Afin de réduire leur influence et d'améliorer la précision en régime statique, on utilise une commande PID.

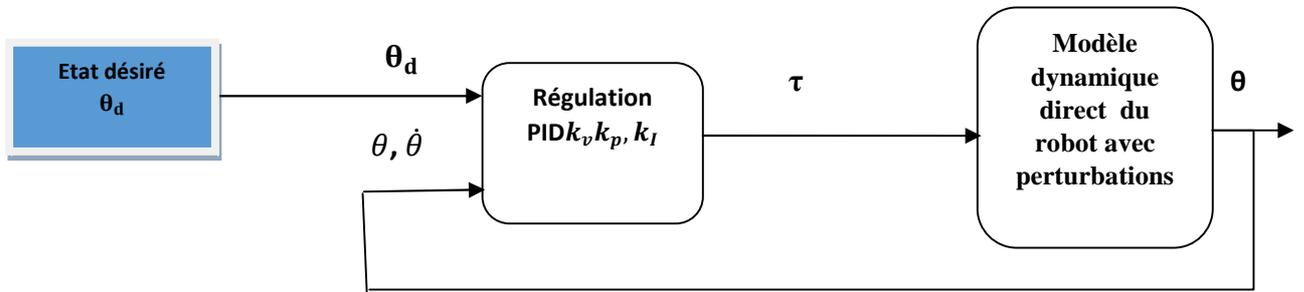


Figure III.6 : Schéma synoptique de simulation de la commande PID du bras de robot.

Le modèle du robot soumis à des perturbations peut être exprimé par la forme :

$$\ddot{\theta} = M^{-1}(\theta) * [\tau - V(\theta, \dot{\theta}) - G(\theta) - F]$$

avec : F les forces perturbatrices. Soit :

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \text{inv} \begin{bmatrix} 2.3 + 1.15C_2 & 0.575 + 0.575C_2 \\ 0.575C_2 + 0.575 & 0.575 \end{bmatrix} * \quad (\text{III.25})$$

$$\left(\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} - \begin{bmatrix} 0.575S_2\dot{\theta}_2^2 - 1.15S_2\dot{\theta}_2\dot{\theta}_1 \\ 1.15S_2\dot{\theta}_1\dot{\theta}_2 \end{bmatrix} - \begin{bmatrix} 11.5C_{12} + 34.5C_1 \\ 11.5C_{12} \end{bmatrix} - \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \right)$$

Calcul de la commande :

La loi de commande PID est de la forme :

$$\begin{aligned}\tau_i &= k_{vi}\dot{e}_i + k_{pi}e_i + k_{ii} \int_0^t e_1(\tau)d\tau \\ k_{vi}\dot{e}_i + k_{pi}e_i + k_{ii} \int_0^t e_1(\tau)d\tau &= m_i\ddot{\theta}\end{aligned}\quad (\text{III.26})$$

Par application de la *transformée de Laplace* on a :

$$m_i P^3 + k_{vi}P^2 + k_{pi}P + k_{ii} = 0 \quad (\text{III.27})$$

On prend $m_i = 1$ (les versions les plus simple de commande n'utilise aucun paramètres de système),

$$P^3 + k_{vi}P^2 + k_{pi}P + k_{ii} = 0 \quad (\text{III.28})$$

L'équation (III.26) est une équation d'ordre 3. Pour éviter les réponses oscillatoires, on prend le facteur d'amortissement $\zeta=1$:

$$(P - P_3)(P^2 + 2\zeta w_n P + w_n^2) = 0 \quad (\text{III.29})$$

On choisit deux pôles dominants à $\zeta=1$ et $w_n=5$ rad/sec. Ce qui donne

$$P_1 = P_2 = -5$$

On choisit le troisième pôle (plus rapide) 5 fois plus grand en module : $P_3 = 5P_2$.

Ce qui donne

$$\begin{aligned}(P+25)(P+10P+25)&=0 \\ P^3 + 35P^2 + 275P + 625 &= 0\end{aligned}\quad (\text{III.30})$$

Par comparaison avec (III.26) et identification : $kv=35$, $kp=275$, $ki=625$

$$\begin{aligned}\tau_1 &= 35\dot{e}_1 + 275e_1 + 625 \int_0^t e_1(\tau)d\tau \\ \tau_2 &= 35\dot{e}_2 + 275e_2 + 625 \int_0^t e_2(\tau)d\tau\end{aligned}\quad (\text{III.31})$$

Simulation et résultats :

On simule le système de la figure III.6 en tenant compte cette fois-ci du modèle avec perturbations (III.25) et la loi de commande (III.31). On génère les perturbations avec la fonction matlab *rand*(un nombre aléatoire entre 0 et 1).

Le programme matlab est donné en annexe (B.2). Les résultats de simulation sont illustrées par la figure (III.7).

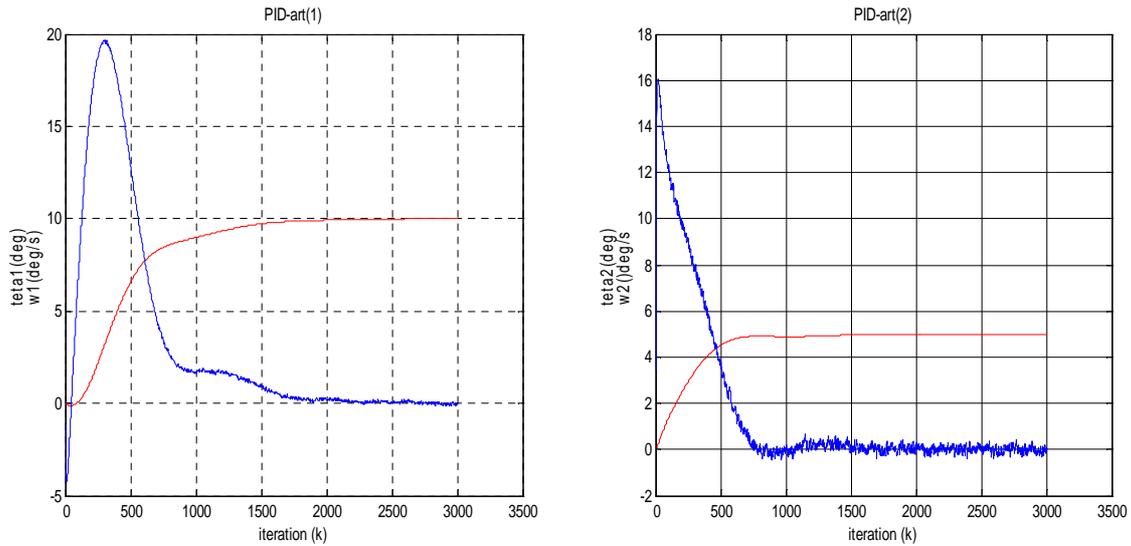


Figure III.7 : Formes des trajectoires de la 1^{ere} et la 2^{eme} articulation : positions réelles θ (en rouge) et vitesses réelles (en bleu)..

D’après la figure III.7, on constate que l’évolution de l’angle $\theta(t)$ converge vers la valeur finale désirée θ_d en un temps très court. On constate aussi que l’effet des perturbations n’est pas apparent sur les variables articulaires, mais affecte les profils des vitesses (plus claire dans la deuxième articulation).

III.4.3 Commande découplante linéarisante

Dans cette partie, on considère le modèle complet (réel) du robot, la méthode est basée sur la linéarisation et le découplage du système. La linéarisation est assurée par le choix de la loi de commande, en utilisant une commande par partition afin de commander chaque articulation séparément. La figure (III.8) illustre la structure de la commande. On fait la régulation à base d’un PD. Si le système est perturbé, on fait appel à la commande PID.

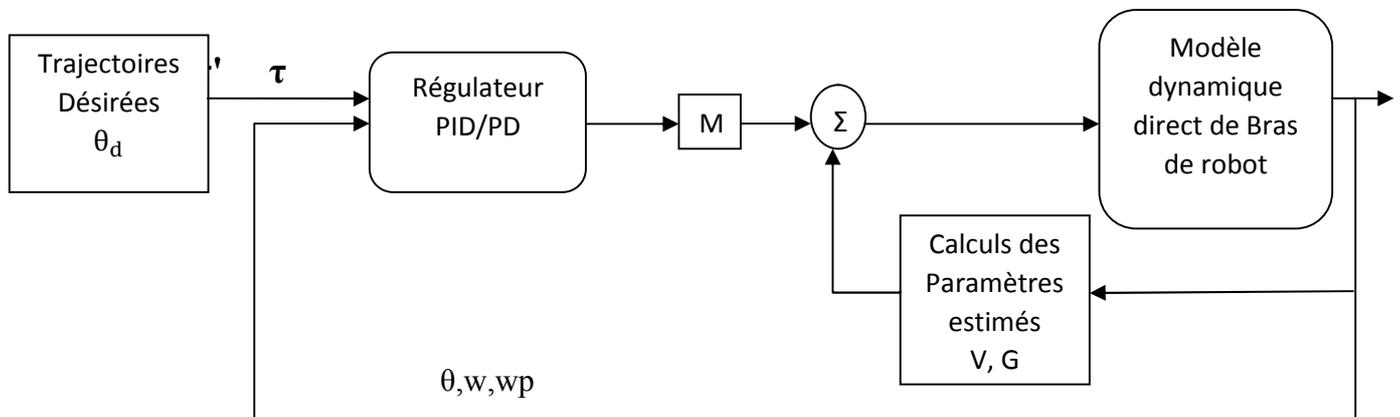


Figure III.8 : Structure de commande non linéaire d’un bras de robot

III.4.3.1 Régulation PD :

Par référence au modèle dynamique direct(III.18) et la loi commande (III.11),on prend :

$$\alpha = \widehat{M}(\theta)\beta = \widehat{V}(\dot{\theta}, \theta) + \widehat{G}(\theta)$$

On aura :

$$\tau' = \ddot{\theta} \quad (\text{III.32})$$

On choisit la loi de commande :

$$\tau' = \ddot{\theta}_d + k_v[\dot{\theta}_d - \dot{\theta}] + k_p[\theta_d - \theta] \quad (\text{III.33})$$

On remplace dans (III.32), on obtient :

$$\begin{aligned} \ddot{\theta} &= \ddot{\theta}_d + k_v[\dot{\theta}_d - \dot{\theta}] + k_p[\theta_d - \theta] \\ \ddot{E} + k_v\dot{E} + k_pE &= 0 \end{aligned} \quad (\text{III.34})$$

Par partition :

$$\ddot{e}_i + k_v\dot{e}_i + k_p e_i = 0 \quad (\text{III.35})$$

Pour le choix des pôles, pour les deux articulations, $\omega_n=5$ rad/sec. On calcule : $k_v=10, k_p=25$.

On remplace dans (III.33) :

$$\begin{aligned} \tau'_1 &= 10\dot{e}_1 + 25e_1 \\ \tau'_2 &= 10\dot{e}_2 + 25e_2 \\ \tau' &= [\tau'_1, \tau'_2] \\ \tau &= \alpha\tau' + \beta \end{aligned} \quad (\text{III.36})$$

Simulation et résultats :

Le robot a été simulé en utilisant le modèle non linéaire couplé et variant dans le temps sans perturbation (III.18). Le programme Matlab est donné en annexe (B.3).Les résultats de simulation sont illustrés par la figure (III.9).

D'après les résultats, on remarque l'évolution de l'angle $\theta(t)$ converge vers la position finale désirée en un temps court sans avoir un dépassement. Ce qui implique les bonnes performances de la commande et bon choix de la pulsation ω_n .

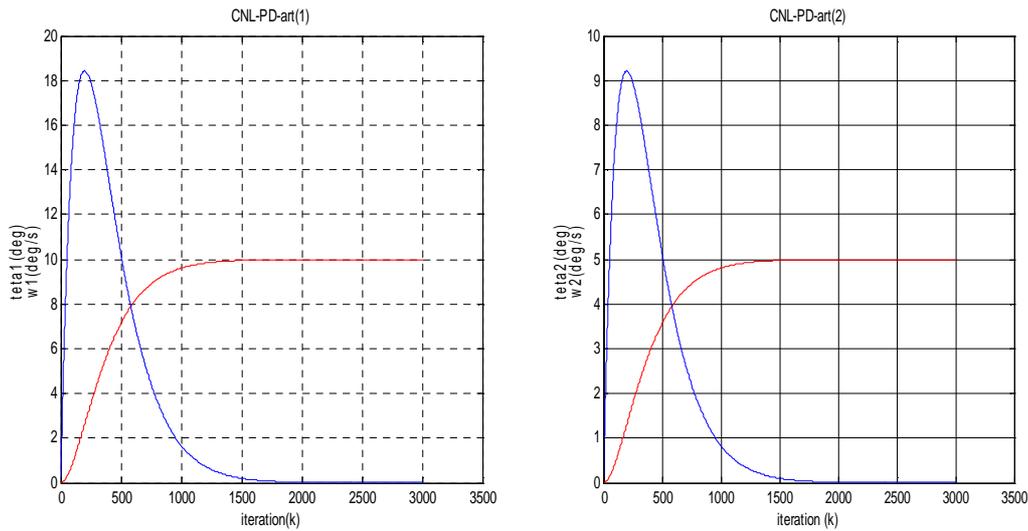


Figure III.9 : Formes de trajectoires de la 1^{ere} et la 2^{eme} articulation : positions réelles θ (en rouge) et vitesses réelles (en bleu).

III.4.3.2 Régulation PID

Pour la régulation PID, on prend directement (III.13). On applique la TL, on aura la même équation que (III.29) :

$$(P - P_3)(P^2 + 2\zeta w_n P + w_n^2) = 0 \quad (III.37)$$

Pour les deux articulations, on choisit $w_n=5$:

$$P^3 + 35P^2 + 275P + 625 = 0 \quad (III.38)$$

Par identification : $k_v = 35, k_p = 275, k_i = 625$

Ce qui donne la loi de commande :

$$\tau_i' = \ddot{\theta}_{di} + 35\dot{e}_i + 275e_i + 625 \int_0^t e_i(\tau) d\tau \quad (III.39)$$

$$\tau' = [\tau'_1, \tau'_2]$$

$$\tau = \alpha \tau' + \beta$$

Simulation et résultats

Le robot a été simulé en utilisant le modèle non linéaire couplé et variant dans le temps avec perturbations (III.25). Le programme Matlab est donné en annexe (B.4). Les résultats de simulation sont illustrés par Figure (III.10).

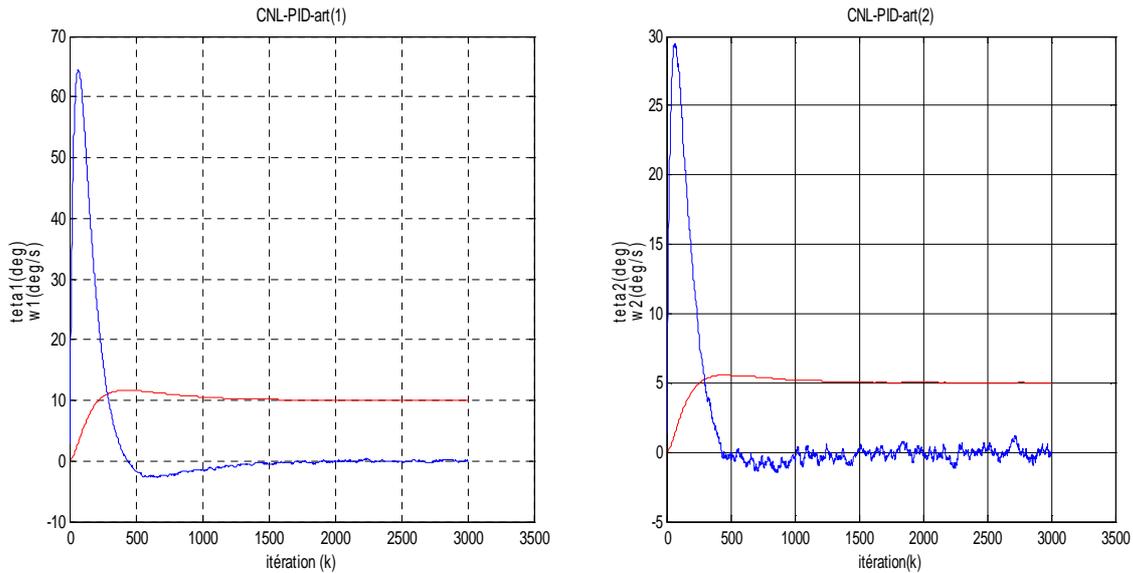


Figure III.10 : Formes des trajectoires de la 1^{ere} et la 2^{eme} articulation : positions réelles θ (en rouge) et vitesses réelles (en bleu).

D'après les résultats illustrés dans la figure (III.10), on constate que les variables articulaires convergent vers les positions désirées en un temps court, après avoir un dépassement. La trajectoire de la vitesse est perturbée. Ce signifie la possibilité d'avoir de légers mouvements autour de la position finale désirée.

Afin de réduire l'influence de ces perturbations sur la deuxième articulation on doit modifier les paramètres de régulation de la deuxième articulation.

III.5 Commande en poursuite

Le but de cette commande est de déplacer le robot, d'une position initiale vers la position finale, *selon une trajectoire désirée* qu'on génère soit avec un module de génération de trajectoire.

Dans ce qui suit, on a deux trajectoire désirées : la première générée avec MGT le d'une valeur initiale ($\theta_0=0$) vers une valeur finale désirée ($\theta_{fd}=100^\circ$) pour la première articulation et ($\theta_{fd}=90^\circ$) pour la deuxième, dans un temps spécifié ($t_f = 20s$). La deuxième générée avec une fonction sinusoïdale.

III.5.1 Commande PD

Cette commande est basée sur la même hypothèse que la commande PD en régulation, ce qui nous permet d'utiliser le même modèle simplifié de robot (III.19).

Structure de la commande :

La commande PD est basée sur l'erreur entre l'état instantané désiré et l'état instantané réelle:

$$\begin{aligned} E &= [\theta_d(t) - \theta(t)] \\ \dot{E} &= [\dot{\theta}_d(t) - \dot{\theta}(t)] \end{aligned} \quad (\text{III.40})$$

Les profils des vitesses et des accélérations angulaires sont imposés. Les couples requis pour les articulations sont calculés à base de régulateurs PD. Les trajectoires réelles sont calculées par le modèle dynamique direct du robot.

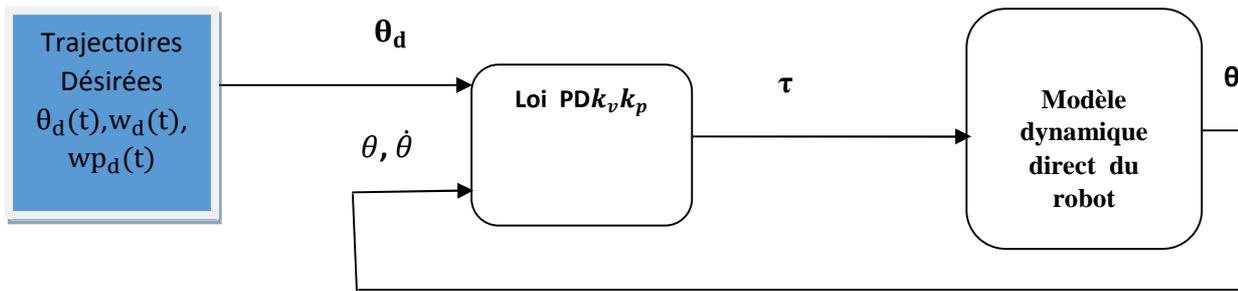


Figure III.11 : Schéma synoptique de la commande PD en poursuite du bras de robot.

Simulation et résultats :

On génère les trajectoires désirées dans un programme (donné en annexe B.5) en utilisant MGT dans lequel on sauvegarde les valeurs instantanées de ces trajectoires. Lors de la commande, on fait appel à ces trajectoires. Les paramètres de cette commande sont les mêmes que la commande en régulation (III.23), mais θ_d est variables cette fois-ci. Le programme de la commande est donné en annexe B.6. Les résultats de simulation sont illustrés par Figure III.12.

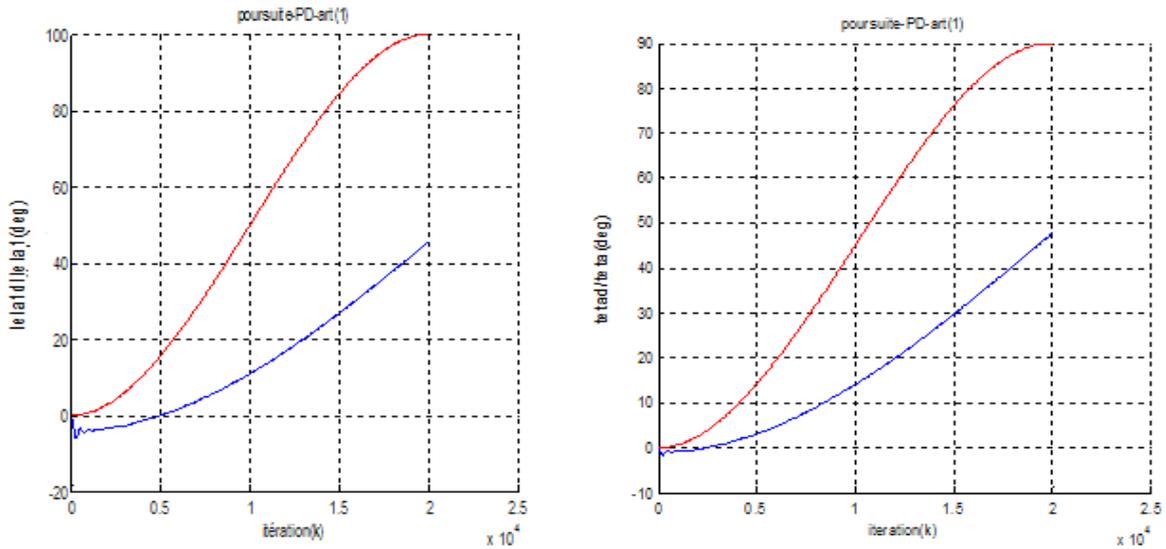


Figure III.12: Formes des trajectoires de la 1^{ère} et la 2^{ème} articulation : positions réelles θ (en bleu) et positions désirées (en rouge).

D'après les résultats, on remarque l'écart et la non convergence des trajectoires obtenues. Après plusieurs jours de recherche, on est arrivé à la conclusion que ces erreurs sont dues à la méthode d'intégration numérique pour avoir $\dot{\vartheta}$ et ϑ à partir de $\ddot{\vartheta}$. Ce qui nous a obligés de travailler avec les logigrammes de Simulink qui dispose d'un outil pour l'intégration. Le schéma de simulation est donné en annexe (figure B.1) et les résultats de simulation illustrés par Figure III.13.

L'organigramme (annexe , figure B.1) contient quatre blocs : un pour la génération des trajectoires désirées dans lequel on utilise le MGT , le deuxième pour la loi de commande et le dernier calcule $\ddot{\theta}$ avec l'équation de modèle dynamique directe (III.18).

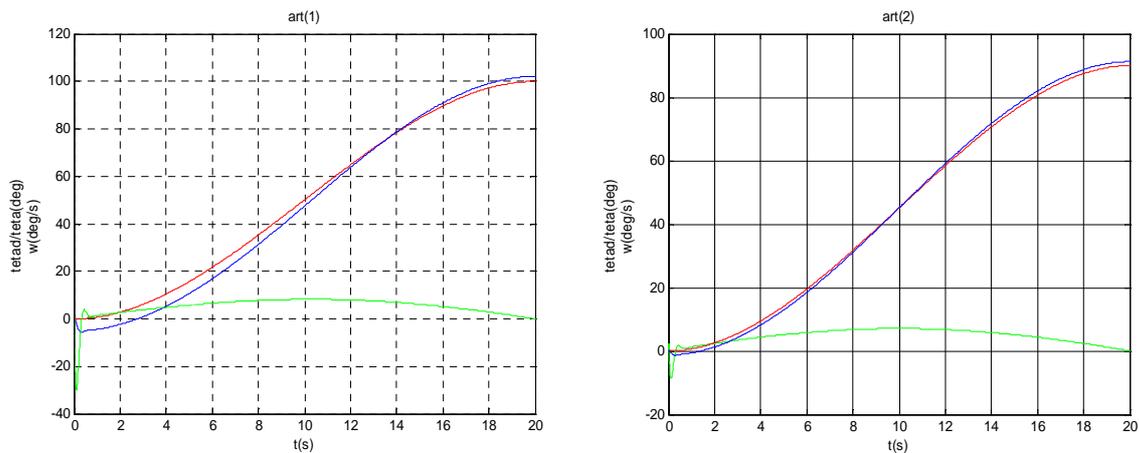
les résultats de simulation

Figure III.13 : Formes des trajectoires de la 1^{ère} et la 2^{ème} articulation : positions réelles θ (en bleu), positions désirées (en rouge) et profils des vitesses réelles (en vert).

On remarque au départ un écart clair entre les deux positions. Lors des premières secondes, les trajectoires des positions et celles de vitesses réelles sont négatives cela est due aux forces centrifuges, gravitationnelle et Coriolis. Si on néglige ces forces dans le modèle dynamique, on aura deux trajectoires de positions superposées et une vitesse positive aux premières secondes.

Pour une trajectoire sinusoïdale, il suffit de modifier le contenu de bloc de génération de trajectoire, dans lequel on introduit la fonction sinus (la figure B.2 en annexe présente l'organigramme utilisé avec le programme de génération de trajectoire), les résultats de simulation illustrés par Figure III.14.

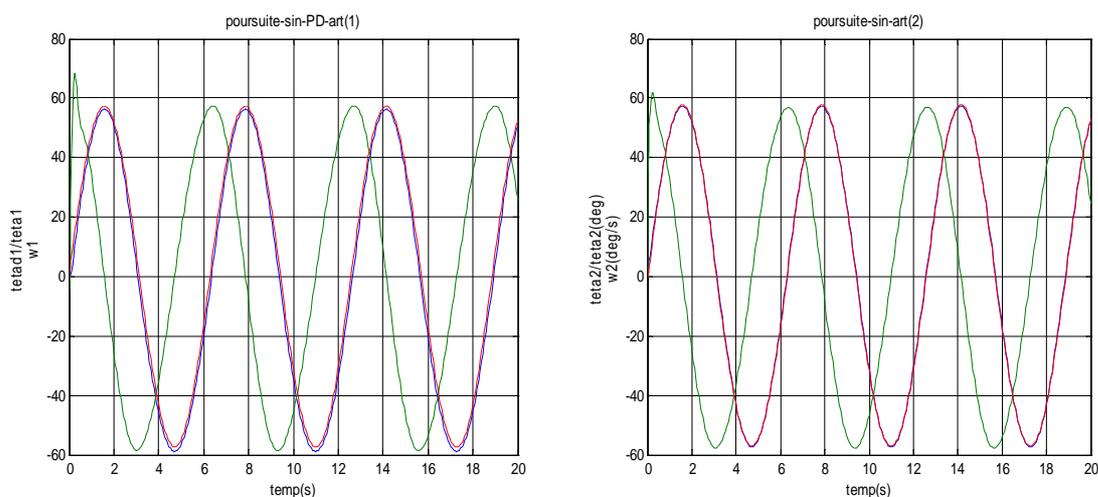


Figure III.14 : Formes des trajectoires de la 1^{ère} et la 2^{ème} articulation : positions réelles θ (en bleu), positions désirées (en rouge) et profils des vitesses réelles (en vert).

Pour la première articulation on constate une erreur entre la position désirée et réelle et pour la deuxième une superposition des deux trajectoires de position. Un ajustement des paramètres de la commande permet de réduire l'erreur de position de la première articulation.

III.5.2 Commande PID :

Pour la commande PID en poursuite, on se réfère à la commande PID en régulation. Donc on a le même modèle simplifié pour la commande (III.24), mais pour la poursuite l'angle désirée variable $\theta_d(t)$ (pour la commande en régulation θ_d fixe). Ce qui donne :

$$\begin{aligned} E &= [\theta_d(t) - \theta(t)] \\ \dot{E} &= [\dot{\theta}_d(t) - \dot{\theta}(t)] \\ \int E(t) dt &= \int [\theta_d(t) - \theta(t)] dt \end{aligned} \quad (\text{III.41})$$

La structure de cette commande est exactement la même que la précédente (commande en poursuite PD) mais cette fois on a les perturbations injectées dans le modèle et la loi de commande PID. Pour le calcul des paramètres, on choisit la même pulsation $\omega_n=5$ rad/sec. Ce qui donne les mêmes paramètres de (III.31)

$$\begin{aligned} \tau_1 &= 35\dot{e}_1 + 275e_1 + 625 \int_0^t e_1(\tau) d\tau \\ \tau_2 &= 35\dot{e}_2 + 275e_2 + 625 \int_0^t e_2(\tau) d\tau \end{aligned} \quad (\text{III.42})$$

Simulation et résultats :

On fait la simulation avec un organigramme (annexe, figure B.3) comme celui de la commande PD, mais on ajoute deux blocs pour l'intégration de l'erreur de chaque articulation. Les blocs de régulations contiennent l'équation (III.42). On ajoute aussi un bloc pour la génération des perturbations qui contient la fonction *rands* qui génère un nombre aléatoire entre 0 et 1. Deux trajectoires désirées sont générées par MGT ou par une simple fonction sinus. Les figures III.15 et III.16 illustrent les résultats des simulations.

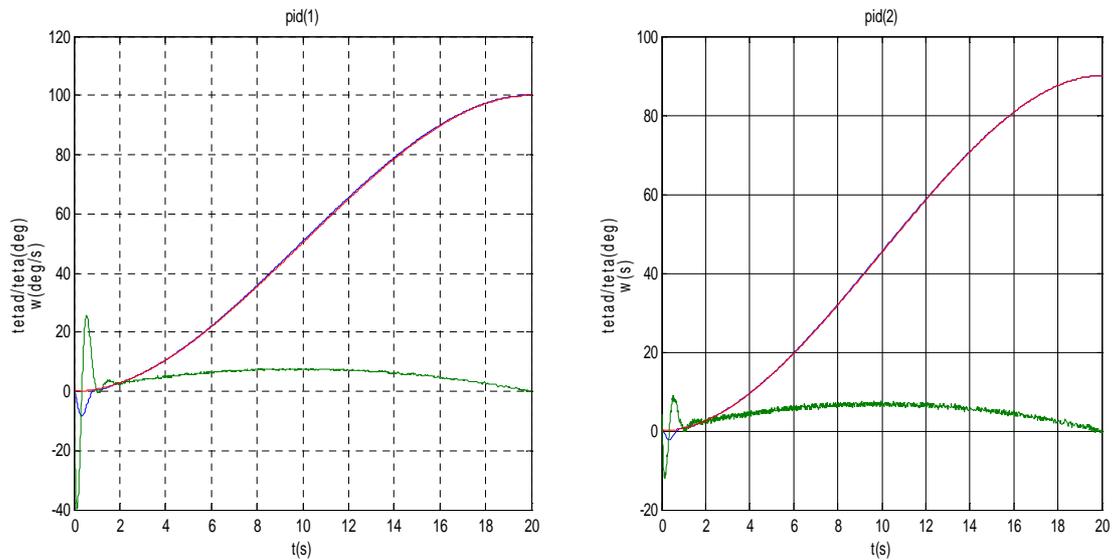
Résultats (MGT) :

Figure III.14 : Formes des trajectoires de la 1^{ère} et la 2^{ème} articulation : positions réelles θ (en bleu), positions désirées (en rouge) et profils des vitesses réelles (en vert).

Au départ, on remarque la chute de la trajectoire de la position réelle pour l'articulation 1 qui supporte le poids du bras. Ensuite l'erreur rapidement annulée. Ceci est moins apparent sur la deuxième articulation.

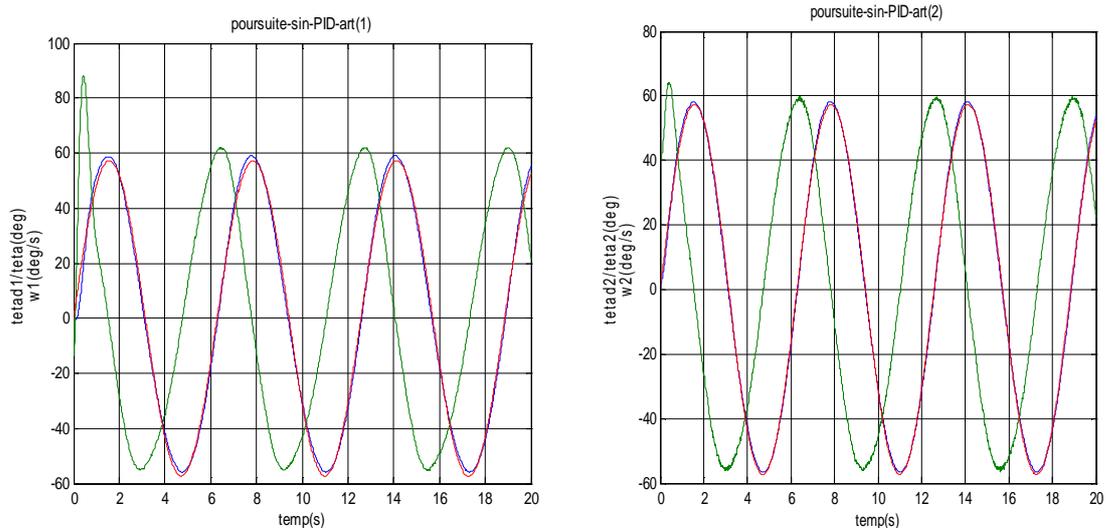
Résultats (sinus) :

Figure III.15 : Formes des trajectoires de la 1^{ère} et la 2^{ème} articulation : positions réelles θ (en bleu), positions désirées (en rouge) et profils des vitesses réelles (en vert).

On constate une erreur de poursuite de position. L'erreur est plus claire quand le robot change la direction. L'évolution de la trajectoire de la position réelle est en rapport avec l'évolution de la trajectoire de vitesse. A chaque fois que la trajectoire de position change de direction, on remarque une vitesse nulle.

III.5.3 Commande linéarisante découplante

Dans cette partie, on considère le modèle complet (réel) du robot, la commande non linéaire en poursuite a le même principe que la commande en régulation, mais on a à suivre une trajectoire instantanée de position et les profils de vitesse et d'accélération sont imposés.

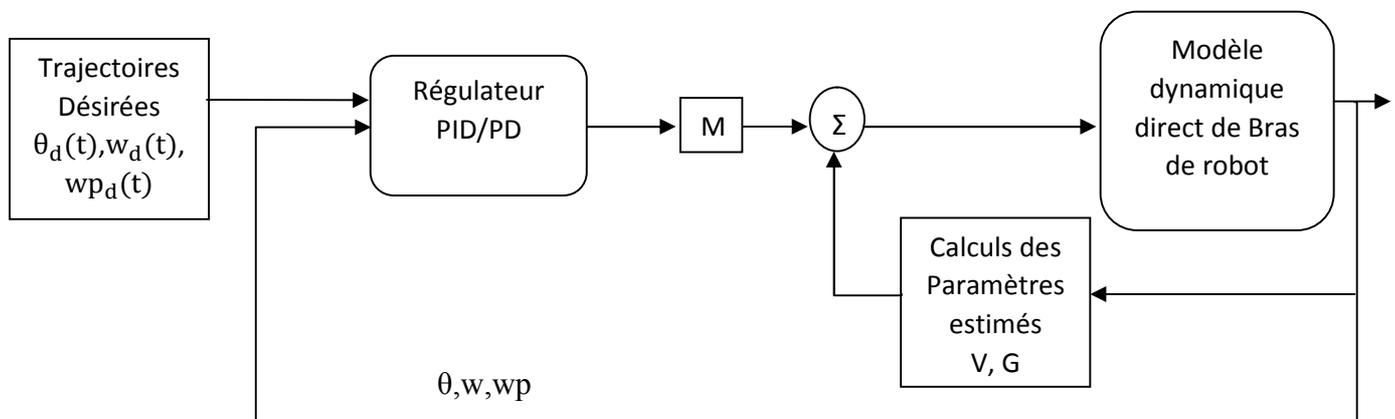


Figure III.16 : Structure de commande non linéaire en poursuite d'un bras de robot

III.5.3.1 Régulation PD :

Puisque si le même principe de commande, on a travaillé avec les mêmes équations et on choisit la même pulsation $\omega_n=5\text{ras/sec}$. Ce qui donne (III.36) mais avec l'accélération désirée $\theta_d(t) \neq 0$.

$$\begin{aligned}\tau'_1 &= \ddot{\theta}_{d1} + 10\dot{e}_1 + 25e_1 \\ \tau'_2 &= \ddot{\theta}_{d2} + 10\dot{e}_2 + 25e_2 \\ \tau' &= [\tau'_1, \tau'_2] \\ \tau &= \alpha\tau' + \beta\end{aligned}\quad (\text{III.43})$$

Pour cette partie, on désigne deux cas selon la trajectoire désirée. Le premier cas, la trajectoire est générée par un module de génération de trajectoire. Pour le deuxième cas, elle est générée par une fonction sinusoïdale.

Simulation et résultats du 1^{er} cas :

Le robot a été simulé en utilisant le modèle non linéaire couplé et variant dans le temps sans perturbations (III.18). Le schéma Simulink, donné en annexe (figure B.4), contient cinq bloc : un pour la génération de trajectoire (pour ce cas est faite un base d'un module de génération de trajectoire) ,deux blocs pour la régulation qui utilisent (III.36) pour le calcul de τ' , un bloc qui calcule τ en utilisant les paramètres estimés et enfin le modèle dynamique qui calcule $\ddot{\theta}_1$ et $\ddot{\theta}_2$. Les résultats de simulation sont illustrés par la *figure III.17*.

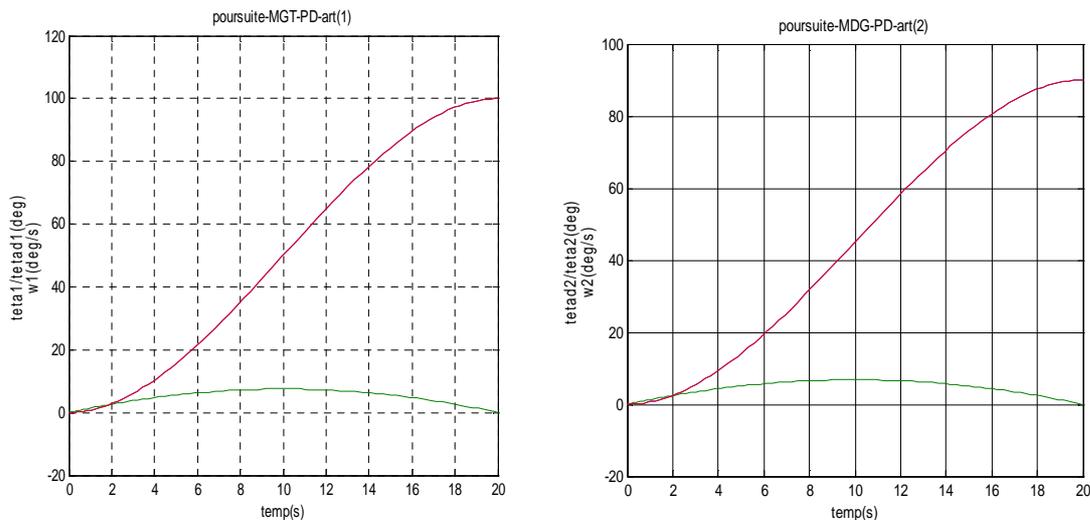


Figure III.17: Formes des trajectoires de la 1^{ere} et la 2^{eme} articulation : positions réelles θ (en bleu), positions désirées (en rouge) et profils des vitesses réelles (en vert).

D'après les résultats, on remarque la superposition des trajectoires de position (poursuite parfaite). Le comportement de vitesse est parfait : une augmentation graduelle (accélération) jusqu'à point milieu ($t=10$), à ce point la vitesse est maximale ensuite on remarque une décélération et à la fin la vitesse s'annule.

Simulation et résultats pour le 2^{ème} cas:

Pour ce cas, on refait exactement les mêmes étapes que le cas précédent, mais on change la méthode de génération de trajectoire(MGT) par une méthode qui utilise des fonctions sinusoïdales ($\theta = \sin(t)$). On modifie l'intitulé de bloc de génération de trajectoire de l'organigramme (on change le programme de l'annexe B.5 par B.6). Les résultats de simulation sont donnés par la *figure III.18*.

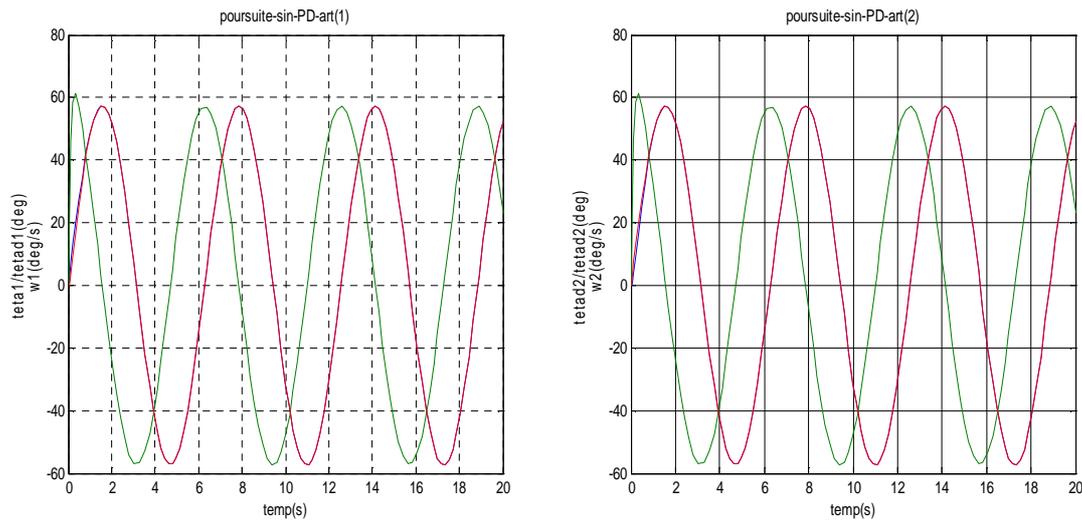


Figure III.18: Formes des trajectoires de la 1^{ère} et la 2^{ème} articulation : positions réelles θ (en bleu), positions désirées (en rouge) et profils des vitesses réelles (en vert).

D'après la figure III.18, on constate une erreur au premières secondes, due au forces gravitationnelle, centrifuges et Coriolis. Ensuite, cette erreur s'annule et on a une superposition des deux trajectoires.

III.5.3.2 Régulation PID :

Comme on a dit au paravent, le même principe de commande en régulation, juste on a trajectoire désirée variable. On applique les mêmes lois de commande et on choisit $\omega_n=5$ rad/sec. Ce qui donne les mêmes paramètres de (III.37) et remplace dans (III.12) :

$$\begin{aligned}\tau'_1 &= \ddot{\theta}_{d1} + 35\dot{e}_1 + 275e_1 + 625 \int_0^t e_1(\tau) d\tau \\ \tau'_2 &= \ddot{\theta}_{d2} + 35\dot{e}_2 + 275e_2 + 625 \int_0^t e_2(\tau) d\tau\end{aligned}\quad (\text{III.44})$$

Simulation et résultats

Le schéma de simulation est le même que celui de l'annexe (figure B.4) mais on ajoute un bloc pour la génération des perturbations et deux blocs pour l'intégration de l'erreur. Au départ, on simule pour une trajectoire générée par un MGT (annexe B.5) en suite avec une fonction sinusoïdale (annexe B.6).

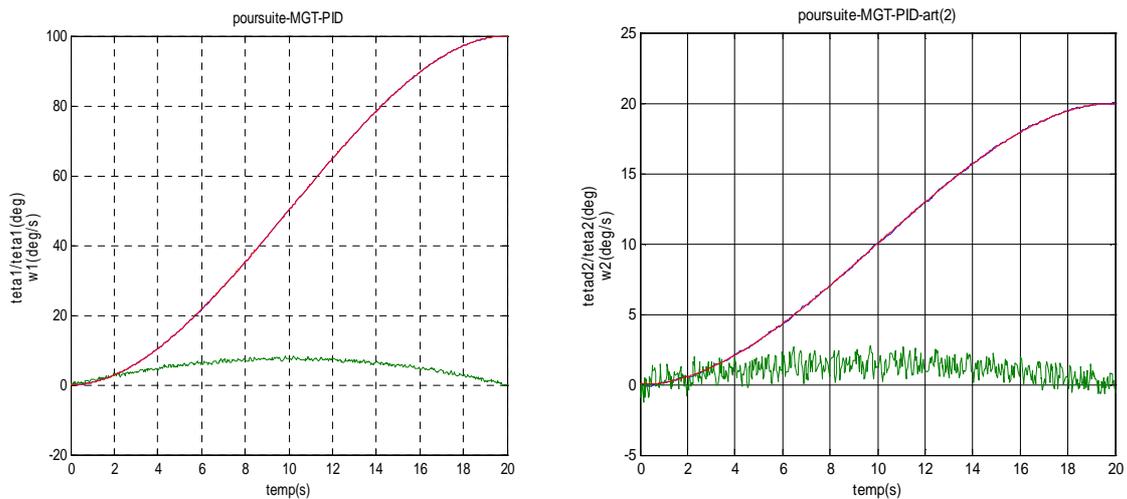
Résultats (MGT)

Figure III.19: Formes des trajectoires de la 1^{ere} et la 2^{eme} articulation : positions réelles θ (en bleu), positions désirées (en rouge) et profils des vitesses réelles (en vert).

Pour la première articulation, on remarque la superposition des trajectoires de position. Ce qui montre les performances de la méthode. Mais, dans la deuxième articulation on remarque une erreur alternante (pour un moment donné c'est la position désirée qui est la plus grande et pour un autre le contraire), la vitesse est complètement perturbée.

Afin de réduire l'influence de ces perturbations sur la deuxième articulation, on doit modifier les paramètres du régulateur de la deuxième articulation.

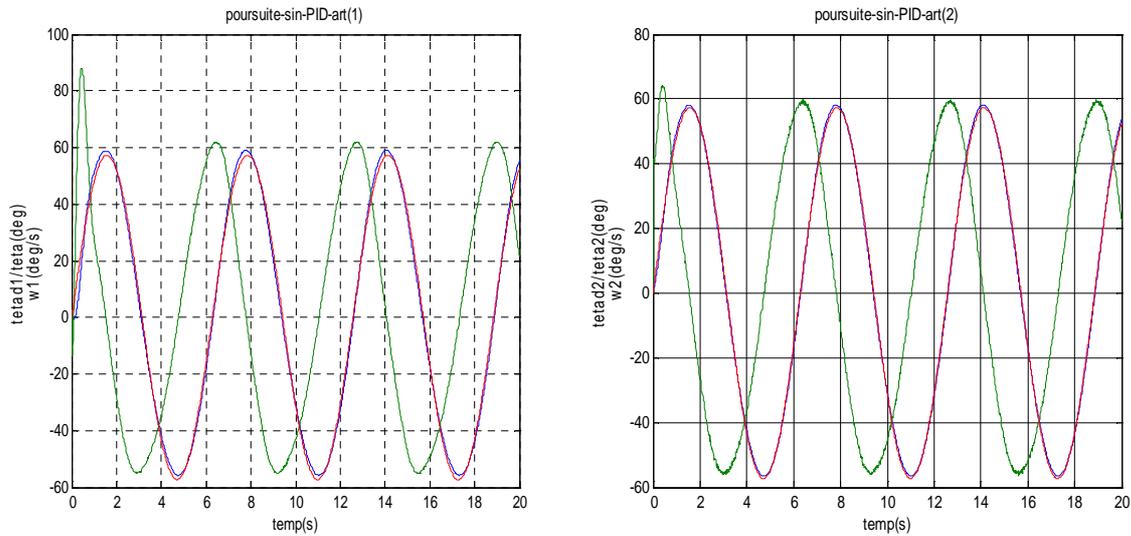
Résultats (sinus)

Figure III.20: Formes des trajectoires de la 1^{ère} et la 2^{ème} articulation : positions réelles θ (en bleu), positions désirées (en rouge) et profils des vitesses réelles (en vert).

D'après les résultats illustrés dans la figure III.20, on constate une erreur entre la trajectoire de position désirée et la trajectoire de position réelle, due aux perturbations. La vitesse atteint sa valeur maximale durant les premières secondes, ensuite elle se stabilise, mais influencée par les perturbations.

III.6 Conclusion

Dans ce chapitre, on a implémenté quelques techniques de commande des bras de robot. L'application de ces différentes techniques sur un modèle de bras de robot à deux degrés de liberté nous a permis d'évaluer les performances de chaque technique et les conditions de son meilleur usage. Ce qui permet, parfois, de commander un bras robot, considéré comme un système non linéaire multi-variable et variant dans le temps, par une simple commande linéaire découplé. C'est le cas de certaines applications industrielles. Mais, dans certaines situations, on a besoin des commandes à la fois performantes simples.

Conclusion générale

Conclusion générale :

Notre présent travail porte sur la planification et le contrôle de trajectoires pour un bras de robot à deux degrés de liberté.

On a abordé notre étude par la modélisation du bras. On a commencé par la modélisation géométrique qui nous a permis de localiser la position et l'orientation de l'effecteur à n'importe quel point dans l'espace de travail en utilisant la convention de Denavit-Hartenberg modifiée. Puis, On a étudié en profondeur le modèle dynamique avec le formalisme de Newton-Euler afin de calculer les couples requis pour effectuer les différents mouvements. La suite du travail concerne l'étude de techniques de commande des bras de robots (commande PD, PID et la commande dynamique linéarisante découplante) et le problème de planification de trajectoires. L'application de cette technique nous a permis de commander le bras de robot en régulation et en poursuite. L'implémentation est faite par simulation en utilisant le logiciel Matlab.

Ce projet nous a permis de consolider les connaissances acquises dans le domaine de la robotique et de la commande. On espère que ce travail sera poursuivi en étudiant les autres aspects de ce sujet, tels que : l'étude des bras à plus de deux degrés de liberté, la simulation 3D et implémentation sur un bras réel, exploration des autres techniques de commande, etc.

Bibliographies

BIBLIOGRAPHIE

- [1] Dombre, Etienne (sous la direction) «*Analyse et modélisation des robots manipulateurs* », Hermes science publication, Paris ,2001.
- [2] Guillaume Laurent et Nicolas « *Chaillet Robotique industrielle*», cours Université de Bourgogne Franche-Comte - UFC – ENSMM, 2017.
- [3] Jean-Louis Boimond « *ROBOTIQUE*» cours , ISTIA, Université Angers,2008.
- [4] John J. Craig « *Introduction to Robotics Mechanics and Control*» , Printice Hall ,3^{eme} edition 2005.
- [5]AIT SALAH Brahim, MEDJAHED Okba« *Commande intelligente d'un bras de robot manipulateur* » Mémoire d'ingénieur Université A.Mira Bejaia,2010.
- [6] Fateh MAKHLOUFI. «*Modélisation et commande des robots manipulateurs par les outils de l'intelligence artificielle* » , Thèse doctorat , Université Annaba , 2015.
- [7] Roland Longchamp «*Commande numérique de systèmes dynamique* » ,(volume 1 : Méthode de base),Presses polytechnique et universitaires romandes 3^{eme} édition , 2010.
- [8] Daneil Lequensne «*Réglage P.I.D*», Lavoisier 2009.
- [9] BOUDISSA Ahmed «*Commande Non Linéaire en Impédance d'un Robot Manipulateur Industriel*» Mémoire de magister , Université de Batna
- [10] P. Fisette, H. Buyse, J.C. Samin, « *Introduction à la Robotique*», cours Université Case Western, 2004.
- [11]AMOURI Ammar «*Modélisation dynamique d'un robot parallèle forme de plusieurs modules empilés* » Mémoire de magister Université L'Aarbi BEN M'HIDI Oum El Bouaghi, 2011.

Annexes

Annexe A

Itérations extérieures $0 \rightarrow n$:

$$w_{i+1}^{i+1} = R_i^{i+1} w_i^i + \dot{\theta}_{i+1} Z_{i+1}^{i+1}$$

$$\dot{w}_{i+1}^{i+1} = R_i^{i+1} \dot{w}_i^i + R_i^{i+1} w_i^i X \dot{\theta}_{i+1} Z_{i+1}^{i+1} + \ddot{\theta}_{i+1} Z_{i+1}^{i+1}$$

$$\dot{v}_{i+1}^{i+1} = R_i^{i+1} [\dot{w}_i^i X P_i^{i+1} + w_i^i X (w_i^i X P_i^{i+1}) + \dot{v}_i^i]$$

$$\dot{v}_{c_{i+1}}^{i+1} = \dot{w}_{i+1}^{i+1} X P_{c_{i+1}}^{i+1} + w_{i+1}^{i+1} X (w_{i+1}^{i+1} X P_{c_{i+1}}^{i+1}) + \dot{v}_{i+1}^{i+1}$$

$$F_{i+1}^{i+1} = m_{i+1} \cdot \dot{v}_{c_{i+1}}^{i+1}$$

$$N_{i+1}^{i+1} = I_{i+1}^{c_{i+1}} \dot{w}_{i+1}^{i+1} + w_{i+1}^{i+1} X I_{i+1}^{c_{i+1}} w_{i+1}^{i+1}$$

Itérations intérieures $n+1 \rightarrow 1$

$$f_i^i = R_{i+1}^i f_{i+1}^{i+1} + F_i^i$$

$$n_i^i = N_i^i + R_{i+1}^i n_{i+1}^{i+1} + P_{c_i}^i X F_i^i + P_{i+1}^i X R_{i+1}^i f_{i+1}^{i+1}$$

$$\tau_i^i = n_i^{iT} Z_i^i$$

Avec :

$I_{i+1}^{c_{i+1}}$: Matrice d'inertie du corps $i+1$ autour de ce centre de masse par rapport à R_{i+1}

$P_{c_{i+1}}^i$: Vecteur aligné le long du corps C_i (rayon de rotation) indiquant l'origine du repère R_i par rapport au centre de masse corps R_{i+1}

X : produit vectoriel

Application des itérations pour le calcul de modèle dynamique :

Après avoir défini des conditions initiales on calcule les matrices de rotation :

$$R_i^{i+1} = \begin{bmatrix} C_{i+1} & -S_{i+1} & \mathbf{0} \\ S_{i+1} & C_{i+1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}$$

$$R_{i+1}^i = \text{inv}(R_i^{i+1}) = \begin{bmatrix} C_{i+1} & S_{i+1} & \mathbf{0} \\ -S_{i+1} & C_{i+1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}$$

On commence les itérations extérieures par $i=1$:

$$w_1^1 = R_0^1 w_0^0 + \dot{\theta}_1 Z_1^1 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} \quad (\text{I.1})$$

$$\dot{w}_1^1 = R_1^1 \dot{w}_1^1 + R_0^1 w_0^0 * \dot{\theta}_1 Z_1^1 + \ddot{\theta}_1 Z_1^1 = \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_1 \end{bmatrix} \quad (\text{I.2})$$

$$\dot{v}_1^1 = R_0^1 [\dot{w}_0^0 X P_0^1 + w_1^1 X (w_1^1 X P_0^1) + \dot{v}_1^1] = \begin{bmatrix} C_1 & -S_1 & \mathbf{0} \\ S_1 & C_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} 0 \\ g \\ 0 \end{bmatrix} = \begin{bmatrix} -gS_1 \\ gC_1 \\ 0 \end{bmatrix} \quad (\text{I.3})$$

Annexe A

$$\dot{v}_{c1}^1 = \dot{w}_1^1 X P_{c1}^1 + w_1^1 X (w_1^1 X P_{c0}^1) + \dot{v}_1^1 = \begin{bmatrix} 0 \\ l1\ddot{\theta}_1 \\ 0 \end{bmatrix} + \begin{bmatrix} -l1\dot{\theta}_1^2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} gS1 \\ gc1 \\ 0 \end{bmatrix} = \begin{bmatrix} -l1\dot{\theta}_1^2 + gS1 \\ l1\ddot{\theta}_1 + gc1 \\ 0 \end{bmatrix}$$

$$F_1^1 = \begin{bmatrix} -m_1 l1 \dot{\theta}_1^2 + m_1 g S1 \\ m_1 l1 \ddot{\theta}_1 + m_1 g C1 \\ 0 \end{bmatrix} \quad (I.5)$$

$$N_1^1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (I.6)$$

Pour i=2

$$w_2^2 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix} \quad (I.7)$$

$$\dot{w}_2^2 = \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_1 + \ddot{\theta}_2 \end{bmatrix} \quad (I.8)$$

$$\dot{v}_2^2 = \begin{bmatrix} C_2 & -S_2 & 0 \\ S_2 & C_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -l1\dot{\theta}_1^2 + gS1 \\ l1\ddot{\theta}_1 + gc1 \\ 0 \end{bmatrix} = \begin{bmatrix} l1\ddot{\theta}_1 S2 - l1\dot{\theta}_1^2 C2 + gS12 \\ l1\ddot{\theta}_1 C1 + l1\dot{\theta}_1^2 S2 + gc12 \\ 0 \end{bmatrix} \quad (I.9)$$

$$\dot{v}_{c2}^2 = \begin{bmatrix} 0 \\ l2(\ddot{\theta}_1 + \ddot{\theta}_2) \\ 0 \end{bmatrix} + \begin{bmatrix} -l2(\dot{\theta}_1 + \dot{\theta}_2)^2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} l1\ddot{\theta}_1 S2 - l1\dot{\theta}_1^2 C2 + gS12 \\ l1\ddot{\theta}_1 C1 + l1\dot{\theta}_1^2 S2 + gc12 \\ 0 \end{bmatrix} =$$

$$\begin{bmatrix} l1\ddot{\theta}_1 S2 - l1\dot{\theta}_1^2 C2 + gS12 - l2(\dot{\theta}_1 + \dot{\theta}_2)^2 \\ l1\ddot{\theta}_1 C1 + l1\dot{\theta}_1^2 S2 + gc12 + l2(\ddot{\theta}_1 + \ddot{\theta}_2) \\ 0 \end{bmatrix} \quad (I.10)$$

$$F_2^2 = \begin{bmatrix} m_2 l1 \ddot{\theta}_1 S2 - m_2 l1 \dot{\theta}_1^2 C2 + m_2 g S12 - m_2 l2 (\dot{\theta}_1 + \dot{\theta}_2)^2 \\ m_2 l1 \ddot{\theta}_1 C1 + m_2 l1 \dot{\theta}_1^2 S2 + m_2 g c12 + m_2 l2 (\ddot{\theta}_1 + \ddot{\theta}_2) \\ 0 \end{bmatrix} \quad (I.11)$$

$$N_2^2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (I.12)$$

On passe aux itérations intérieures :

On commence par i=2

$$f_2^2 = F_2^2 \quad (I.13)$$

Annexe A

$$n_2^2 = \begin{bmatrix} l_2 \\ 0 \\ 0 \end{bmatrix} X \begin{bmatrix} m_2 l_1 \ddot{\theta}_1 S_2 - m_2 l_1 \dot{\theta}_1^2 C_2 + m_2 g S_{12} - m_2 l_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 \\ m_2 l_1 \ddot{\theta}_1 C_1 + m_2 l_1 \dot{\theta}_1^2 S_2 + m_2 g C_{12} + m_2 l_2 (\ddot{\theta}_1 + \ddot{\theta}_2) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ m_2 l_1 l_2 c_2 \ddot{\theta}_1 + m_2 l_1 l_2 s_2 \dot{\theta}_1^2 + m_2 l_1^2 (\ddot{\theta}_1 + \ddot{\theta}_2) \end{bmatrix} \quad (I.14)$$

Pour $i=1$,

$$f_1^1 = \begin{bmatrix} C_{i+1} & -S_{i+1} & \mathbf{0} \\ S_{i+1} & C_{i+1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} m_2 l_1 \ddot{\theta}_1 S_2 - m_2 l_1 \dot{\theta}_1^2 C_2 + m_2 g S_{12} - m_2 l_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 \\ m_2 l_1 \ddot{\theta}_1 C_1 + m_2 l_1 \dot{\theta}_1^2 S_2 + m_2 g C_{12} + m_2 l_2 (\ddot{\theta}_1 + \ddot{\theta}_2) \\ 0 \end{bmatrix} + \begin{bmatrix} -m_1 l_1 \dot{\theta}_1^2 + m_1 g S_1 \\ m_1 l_1 \ddot{\theta}_1 + m_1 g C_1 \\ 0 \end{bmatrix} \quad (I.15)$$

$$n_1^1 = \begin{bmatrix} 0 \\ 0 \\ m_2 l_1 l_2 c_2 \ddot{\theta}_1 + m_2 l_1 l_2 s_2 \dot{\theta}_1^2 + m_2 l_1^2 (\ddot{\theta}_1 + \ddot{\theta}_2) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ m_1 l_1^2 \ddot{\theta}_1 + m_1 l_1 g c_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ m_2 l_1^2 \ddot{\theta}_1 + m_2 l_1 l_2 s_2 (\dot{\theta}_2 + \dot{\theta}_1)^2 + m_2 l_1 g S_1 S_{12} + m_2 l_1 l_2 C (\ddot{\theta}_2 + \ddot{\theta}_1) + m_2 l_1 g c_2 c_{12} \end{bmatrix}$$

Ce qui donne

$$n_1^1 = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ nz \end{bmatrix} \text{ avec :}$$

$$nz = m_2 l_1^2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l_2 c_2 (2\ddot{\theta}_1 + \ddot{\theta}_2) + (m_1 + m_2) l_1^2 \ddot{\theta}_1 - m_2 l_1 l_2 c_2 \dot{\theta}_1^2 - 2m_2 l_1 l_2 s_2 \dot{\theta}_1 \dot{\theta}_2 + m_2 l_2 g c_{12} + (m_1 + m_2) l_1 g c_1 \quad (I.17)$$

La composante Z de vecteur n_i^i représente τ_i ce qui donne :

$$\begin{aligned} \tau_1 &= m_2 l_1^2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l_2 c_2 (2\ddot{\theta}_1 + \ddot{\theta}_2) + (m_1 + m_2) l_1^2 \ddot{\theta}_1 - m_2 l_1 l_2 c_2 \dot{\theta}_1^2 \\ &\quad - 2m_2 l_1 l_2 s_2 \dot{\theta}_1 \dot{\theta}_2 + m_2 l_2 g c_{12} + (m_1 + m_2) l_1 g c_1 \\ \tau_2 &= m_2 l_1 l_2 c_2 \ddot{\theta}_1 + m_2 l_1 l_2 s_2 \dot{\theta}_1^2 + m_2 l_1^2 (\ddot{\theta}_1 + \ddot{\theta}_2) \end{aligned} \quad (I.18)$$

Annexe B

B.0 La fonction de calcul des paramètres de polynôme cubique :

```
function [A1,A2] = fnc_polycub(teta10,teta1f,teta20,teta2f,tf)

tf2=tf*tf; tf3=tf2*tf;
    a0=teta10; a1=0; a2=3*(teta1f-teta10)/tf2; a3=-2*(teta1f-teta10)/tf3;
    a20=teta20; a21=0; a22=3*(teta2f-teta20)/tf2; a23=-2*(teta2f-teta20)/tf3;
    A1=[a0,a1,a2,a3];
    A2=[a20,a21,a22,a23];
```

B.1 Le programme de commande PD en régulation

```
clear all
clc
teta=[0*pi/180;0*pi/180];w=[0;0];wp=[0;0]; Te=0.001;Te2=Te^2;tf=3;
l=0.5;l1=0.5;l2=0.5;m1=4.6;m2=2.3;m3=1;g=9.8;
a=[];vi=[];ac=[];tool=[];
SI=[0;0];M=[3.4499 0;0 0.575];
%wn=25,
kvv=50;kpp=625;
    JI=[1;1];Kp=kpp*JI;Kv=kvv*JI;
EE=[];EV=[];
ttd=[10;5]*pi/180;
k=1;
for t=Te:Te:(tf+Te)
    E=ttd-teta; Ev=-w;
    SI=SI+E;
    to=Kv.*Ev+Kp.*E

        tol=to(1);
        to2=to(2);
        w1=w(1);w2=w(2);
        teta1=teta(1);teta2=teta(2);
        M=[l2^2*m2+2*l1*l2*m2*cos(teta2)+l1^2*(m1+m2)
12^2*m2+l1*l2*m2*cos(teta2);
        l2^2*m2+l1*l2*m2*cos(teta2) l2^2*m2];
        V=[-m2*l1*l2*sin(teta2)*w2^2-2*m2*l1*l2*sin(teta2)*w1*w2
        m2*l1*l2*sin(teta2)*w1^2];
        G=[m2*l2*g*cos(teta2+teta1)+(m1+m2)*l1*g*cos(teta1)
        m2*l2*g*cos(teta2+teta1)];
        yp=[rand,1-rand]';
        wp=inv(M)*(to-(V+G+0*yp));
        w=w+Te*wp;
        teta=teta+Te*w-0.5*wp*Te^2;
        a=[a teta];
        vi=[vi w];

    k=k+1
end
figure
plot(a(1,:)*180/pi,'r')
hold on
plot(vi(1,:)*180/pi)
hold off
```

Annexe B

```
grid
figure
plot(a(2,:)*180/pi, 'r')
hold on
plot(vi(2,:)*180/pi)
grid
```

B.2 Le programme de commande PID en régulation

```
clear all
clc
teta=[0*pi/180;0*pi/180];w=[0;0];wp=[0;0]; Te=0.001;Te2=Te^2;tf=3;
l=0.5;l1=0.5;l2=0.5;m1=4.6;m2=2.3;m3=1;g=9.8;
a=[];vi=[];ac=[];tool=[];
SI=[0;0];M1=[3.4499;0.575];
J1=l2^2*m2+2*l1*l2*m2*cos(2.5*pi/180)+l1^2*(m1+m2);
J2=l2^2*m2;
Mc=diag([J1 J2])
%wn=25,
kvv=35;kpp=275;kii=625
JI=[1;1];Kp=kpp*JI;Kv=kvv*JI;Ki=kii*JI/1226
EE=[];EV=[];
ttd=[10;5]*pi/180;
k=1;
for t=Te:Te:(tf+Te)
    E=ttd-teta; Ev=-w;
    SI=SI+E;
    to=Kv.*Ev+Kp.*E
    to=(Kv.*Ev+Kp.*E+Ki.*SI)
    to1=to(1);
    to2=to(2);
    w1=w(1);w2=w(2);
    teta1=teta(1);teta2=teta(2);
    M=[l2^2*m2+2*l1*l2*m2*cos(teta2)+l1^2*(m1+m2)
    l2^2*m2+l1*l2*m2*cos(teta2);
    -m2*l1*l2*sin(teta2)*w2^2-2*m2*l1*l2*sin(teta2)*w1*w2
    m2*l1*l2*sin(teta2)*w1^2];
    G=[m2*l2*g*cos(teta2+teta1)+(m1+m2)*l1*g*cos(teta1)
    m2*l2*g*cos(teta2+teta1)];
    yp=[rand,1-rand]'
    wp=inv(M)*(to-(V+G+yp));
    w=w+Te*wp;
    teta=teta+Te*w-0.5*wp*Te^2;
    a=[a teta];
    vi=[vi w];

    k=k+1
end
figure
plot(a(1,:)*180/pi, 'r')
hold on
plot(vi(1,:)*180/pi)
hold off
grid
figure
```

Annexe B

```
plot(a(2,:)*180/pi,'r')
hold on
plot(vi(2,:)*180/pi)
grid
```

B.3 Le programme de commande non linéaire en régulation sans perturbations.

```
clear all
clc
teta=[0*pi/180;0*pi/180];w=[0;0];wp=[0;0]; Te=0.001;Te2=Te^2;tf=3;
tetal=teta(1);teta2=teta(2);
l=0.5;l1=0.5;l2=0.5;m1=4.6;m2=2.3;m3=1;g=9.8;
a=[];vi=[];ac=[];tool=[];
SI=[0;0];M=[3.4499 0;0 0.575];
%wn=5,
kvv=5;kpp=25;
JI=[1;1];Kp=kpp*JI;Kv=kvv*JI;
EE=[];EV=[];
ttd=[10;5]*pi/180;
k=1;
for t=Te:Te:(tf+Te)
    E=ttd-teta; Ev=-w/w1=w(1);w2=w(2);
    SI=SI+E;
    top=Kv.*Ev+Kp.*E
    top1=top(1);
    top2=top(2);
% les paramètre estimé
Me=[2.3+1.15*cos(teta2) 0.575*(1+cos(teta2));
    0.575*(1+cos(teta2)) 0.575];
Ve=[-0.575*sin(teta2)*w2^2-1.15*sin(teta2)*w1*w2
    0.575*sin(teta2)*w1^2];
Ge=[11.27*cos(tetal+teta2)+33.81*cos(tetal)
    11.27*cos(tetal+teta2)];
% Me=diag(diag(Me));
alpha=Me;beta=Ve+Ge;
to=alpha*top+beta
tol=to(1);
to2=to(2);
w1=w(1);w2=w(2);
tetal=teta(1);teta2=teta(2);
M=[l2^2*m2+2*l1*l2*m2*cos(teta2)+l1^2*(m1+m2)
l2^2*m2+l1*l2*m2*cos(teta2);
l2^2*m2+l1*l2*m2*cos(teta2) l2^2*m2];
V=[-m2*l1*l2*sin(teta2)*w2^2-2*m2*l1*l2*sin(teta2)*w1*w2
m2*l1*l2*sin(teta2)*w1^2];
G=[m2*l2*g*cos(teta2+tetal)+(m1+m2)*l1*g*cos(tetal)
m2*l2*g*cos(teta2+tetal)];
yp=[rand,1-rand]';
wp=inv(M)*(to-(V+G+0*yp));
w=w+Te*wp;
teta=teta+Te*w-0.5*wp*Te^2;
a=[a teta];
vi=[vi w];
k=k+1
```

Annexe B

```
end
figure
plot(a(1,:)*180/pi,'r')
hold on
plot(vi(1,:)*180/pi)
hold off
grid
figure
plot(a(2,:)*180/pi,'r')
hold on
plot(vi(2,:)*180/pi)
grid
```

B.4 Le programme de la commande non linéaire en régulation avec perturbations.

```
clear all
clc
teta=[0*pi/180;0*pi/180];w=[0;0];wp=[0;0]; Te=0.001;Te2=Te^2;tf=3;
teta1=teta(1);teta2=teta(2);
l=0.5;l1=0.5;l2=0.5;m1=4.6;m2=2.3;m3=1;g=9.8;
a=[];vi=[];ac=[];too1=[];
SI=[0;0];M=[3.4499 0;0 0.575];
%wn=25,
kvv=35;kpp=275;kii=625;
JI=[1;1];Kp=kpp*JI;Kv=kvv*JI;Ki=kii*JI/1226
EE=[];EV=[];
ttd=[10;5]*pi/180;
k=1;
for t=Te:Te:(tf+Te)
    E=ttd-teta; Ev=-w/w1=w(1);w2=w(2);
    SI=SI+E;
    top=(Kv.*Ev+Kp.*E+Ki.*SI)
    top1=top(1);
    top2=top(2);
%application des paramètre estimé
Me=[2.3+1.15*cos(teta2) 0.575*(1+cos(teta2));
    0.575*(1+cos(teta2)) 0.575];
Ve=[-0.575*sin(teta2)*w2^2-1.15*sin(teta2)*w1*w2
    0.575*sin(teta2)*w1^2];
Ge=[11.27*cos(teta1+teta2)+33.81*cos(teta1)
    11.27*cos(teta1+teta2)];
% Me=diag(diag(Me));
alpha=Me;beta=Ve+Ge;
to=alpha*top+beta
to1=to(1);
to2=to(2);
w1=w(1);w2=w(2);
teta1=teta(1);teta2=teta(2);
M=[l2^2*m2+2*l1*l2*m2*cos(teta2)+l1^2*(m1+m2)
l2^2*m2+l1*l2*m2*cos(teta2);
l2^2*m2+l1*l2*m2*cos(teta2) l2^2*m2];
V=[-m2*l1*l2*sin(teta2)*w2^2-2*m2*l1*l2*sin(teta2)*w1*w2
m2*l1*l2*sin(teta2)*w1^2];
G=[m2*l2*g*cos(teta2+teta1)+(m1+m2)*l1*g*cos(teta1)
m2*l2*g*cos(teta2+teta1)];
yp=[rand,1-rand]'
```

Annexe B

```
wp=inv(M)*(to-(V+G+yp));
w=w+Te*wp;
teta=teta+Te*w-0.5*wp*Te^2;
a=[a teta];
vi=[vi w];

k=k+1
end
figure
plot(a(1,:)*180/pi, 'r')
hold on
plot(vi(1,:)*180/pi)
hold off
grid
figure
plot(a(2,:)*180/pi, 'r')
hold on
plot(vi(2,:)*180/pi)
grid
```

B.5 Programme de génération de trajectoire avec MGT :

```
clear all
clc
teta10=0;teta1f=100;teta20=0;teta2f=90;tf=20;Te=0.001;
tt=[0:Te:tf]
tf2=tf*tf; tf3=tf*tf*tf;
a0=teta10; a1=0; a2=3*(teta1f-teta10)/tf2; a3=-2*(teta1f-teta10)/tf3;
a20=teta20; a21=0; a22=3*(teta2f-teta20)/tf2; a23=-2*(teta2f-teta20)/tf3;
tetd1=(a0+a1.*tt+a2.*tt.^2+a3.*tt.^3)*pi/180;
wd1=(a1+2*a2.*tt+3*a3.*tt.^2)*pi/180;
wpd1=(2*a2+6*a3.*tt)*pi/180;
tetd2=(a20+a21.*tt+a22.*tt.^2+a23.*tt.^3)*pi/180;
wd2=(a21+2*a22.*tt+3*a23.*tt.^2)*pi/180;
wpd2=(2*a22+6*a23.*tt)*pi/180;

tetd=[tetd1;tetd2];
tetdv=[wd1;wd2];
tetda=[wpd1;wpd2];
save geneplot123 tetd tetdv tetda
```

B.6 Programme de génération de trajectoire avec la fonction sinus :

```
function [tetd1,wd1,wpd1,tetd2,wd2,wpd2] = fcn(t)

a=1
tetd1=sin(a*t);
wd1=a*cos(a*t);
wpd1=-(a^2)*sin(a*t);
tetd2=sin(a*t);
wd2=a*cos(a*t);
wpd2=-(a^2)*sin(a*t);
```

Annexe B

B.7 Programme de commande PD en poursuite :

```
clear all
clc
load geneplot123
% tetd=tetd; tetdv=tetdv*pi/180; tetda=tetda*pi/180;
teta=[0*pi/180;0*pi/180];w=[0;0];wp=[0;0]; Te=0.001;Te2=Te^2;tf=20;
l=0.5;l1=0.5;l2=0.5;m1=4.6;m2=2.3;m3=1;g=9.8;
a=[];vi=[];ac=[];tool=[];EE=[];EV=[];E=[0;0];
SI=[0;0];
    w1=w(1);w2=w(2); tetal=teta(1);teta2=teta(2);
kvv=25;kpp=625;
J11=[1;1];Kp=kpp*J11;Kv=kvv*J11;
k=1;
for t=Te:Te:(tf+Te)
    E=tetd(k)-teta; Ev=tetdv(k)-w;
    %    SI=SI+E;
    to=Kv.*Ev+Kp.*E
        to1=to(1);
        to2=to(2);

        w1=w(1);w2=w(2);
        tetal=teta(1);teta2=teta(2);
        M=[l2^2*m2+2*l1*l2*m2*cos(teta2)+l1^2*(m1+m2)
l2^2*m2+l1*l2*m2*cos(teta2);
        l2^2*m2+l1*l2*m2*cos(teta2) l2^2*m2];
        V=[-m2*l1*l2*sin(teta2)*w2^2-2*m2*l1*l2*sin(teta2)*w1*w2
m2*l1*l2*sin(teta2)*w1^2];
        G=[m2*l2*g*cos(teta2+tetal)+(m1+m2)*l1*g*cos(tetal)
m2*l2*g*cos(teta2+tetal)];
        wp=inv(M)*(to-1*(V+G));
        w=w+Te*wp;
        teta=teta+Te*w-0.5*wp*Te^2;
        a=[a teta];
        vi=[vi w];

    k=k+1
end

figure
hold on
plot(a(1,:)*(180/pi),'b');
plot(tetd(1,:)*180/pi,'r');
grid
figure
hold on
plot(a(2,:)*180/pi,'b')
plot(tetd(2,:)*180/pi,'r');

grid
```

Annexe B

```
function [tetd1,wd1,wpd1,tetd2,wd2,wpd2] = fcn(t)
teta10=0;teta1f=100;teta20=0;teta2f=90;tf=20
tf2=tf*tf; tf3=tf2*tf;
```

```

a0=teta10; a1=0; a2=3*(teta1f-teta10)/tf2;
a3=-2*(teta1f-teta10)/tf3;
a20=teta20; a21=0; a22=3*(teta2f-teta20)/tf2;
a23=-2*(teta2f-teta20)/tf3;
tetd1=(a0+a1*tt+a2*tt^2+a3*tt^3)*pi/180;
wd1=(a1+2*a2*tt+3*a3*tt^2)*pi/180;
wpd1=(2*a2+6*a3*tt)*pi/180;
tetd2=(a20+a21*tt+a22*tt^2+a23*tt^3)*pi/180;
wd2=(a21+2*a22*tt+3*a23*tt^2)*pi/180;
wpd2=(2*a22+6*a23*tt)*pi/180;
```

```
function u1 = fcn(w1,teta1,tetad1,wd1)
u1=(625*(tetad1-teta1)+50*(wd1-w1));
```

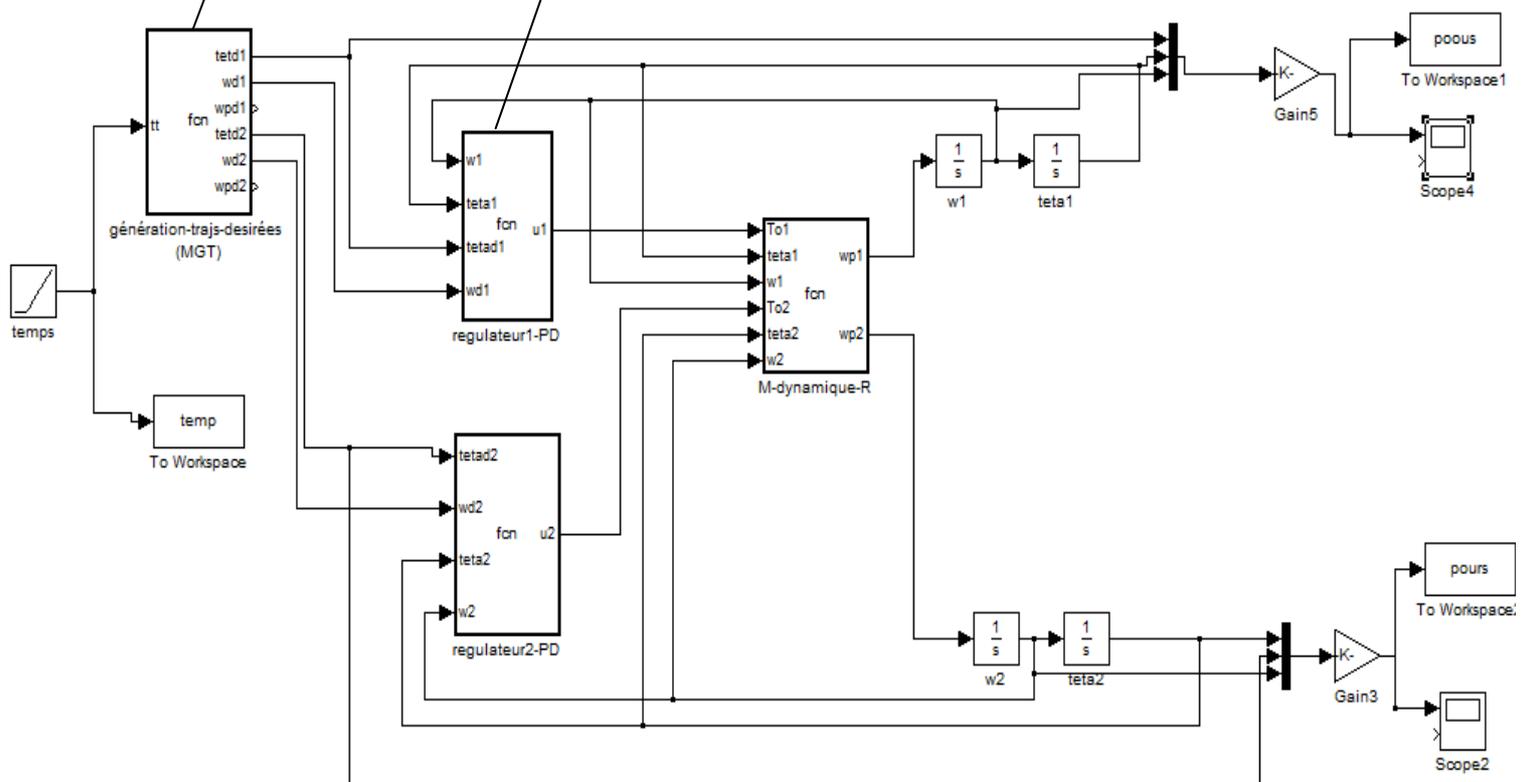


Figure B.1 : organigramme de la commande PD en poursuite pour une trajectoire générée par MGT

Annexe B

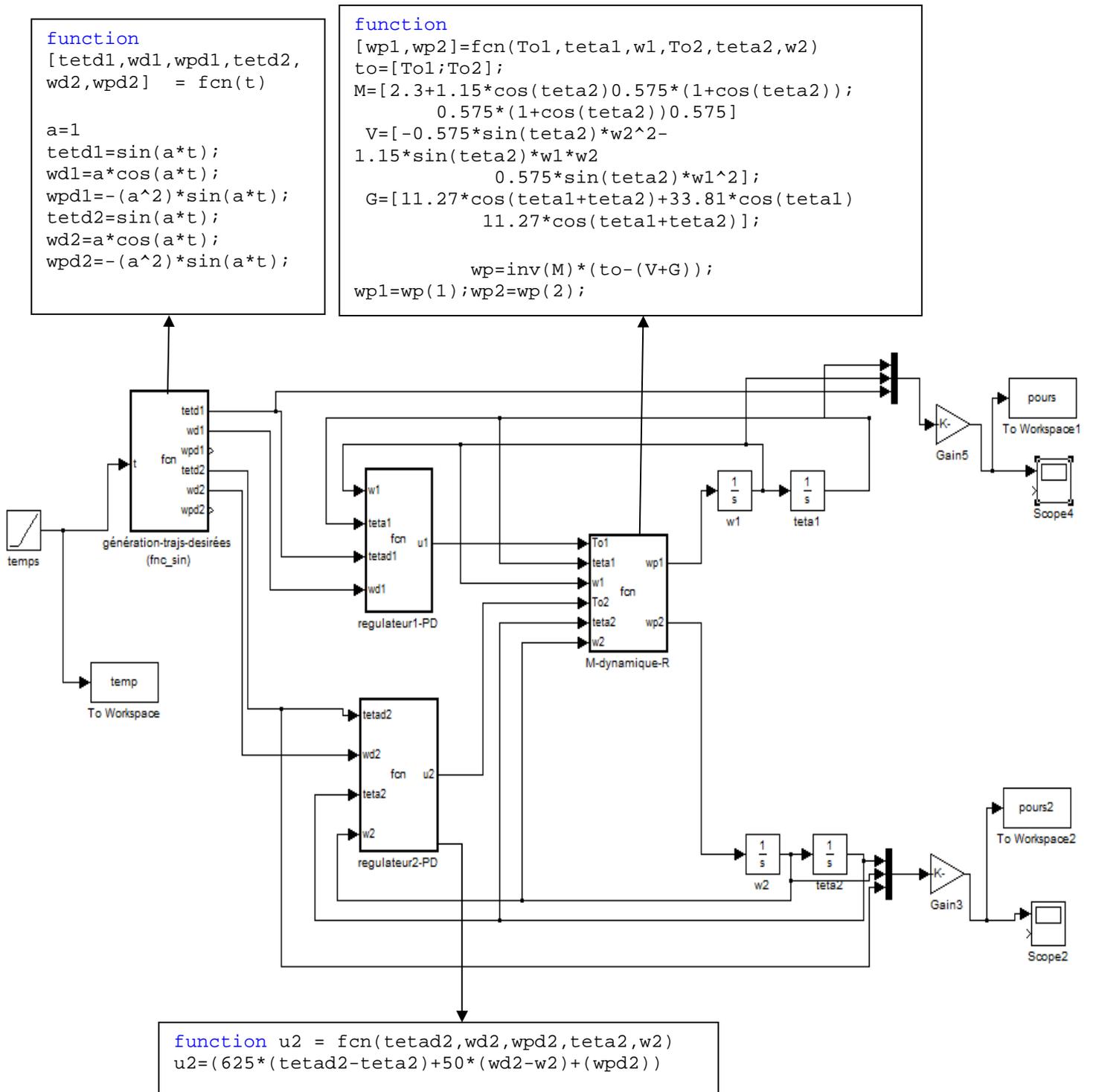


Figure B.2 : organigramme de la commande PD en poursuite pour une trajectoire générée par une fonction sinusoidale

Annexe B

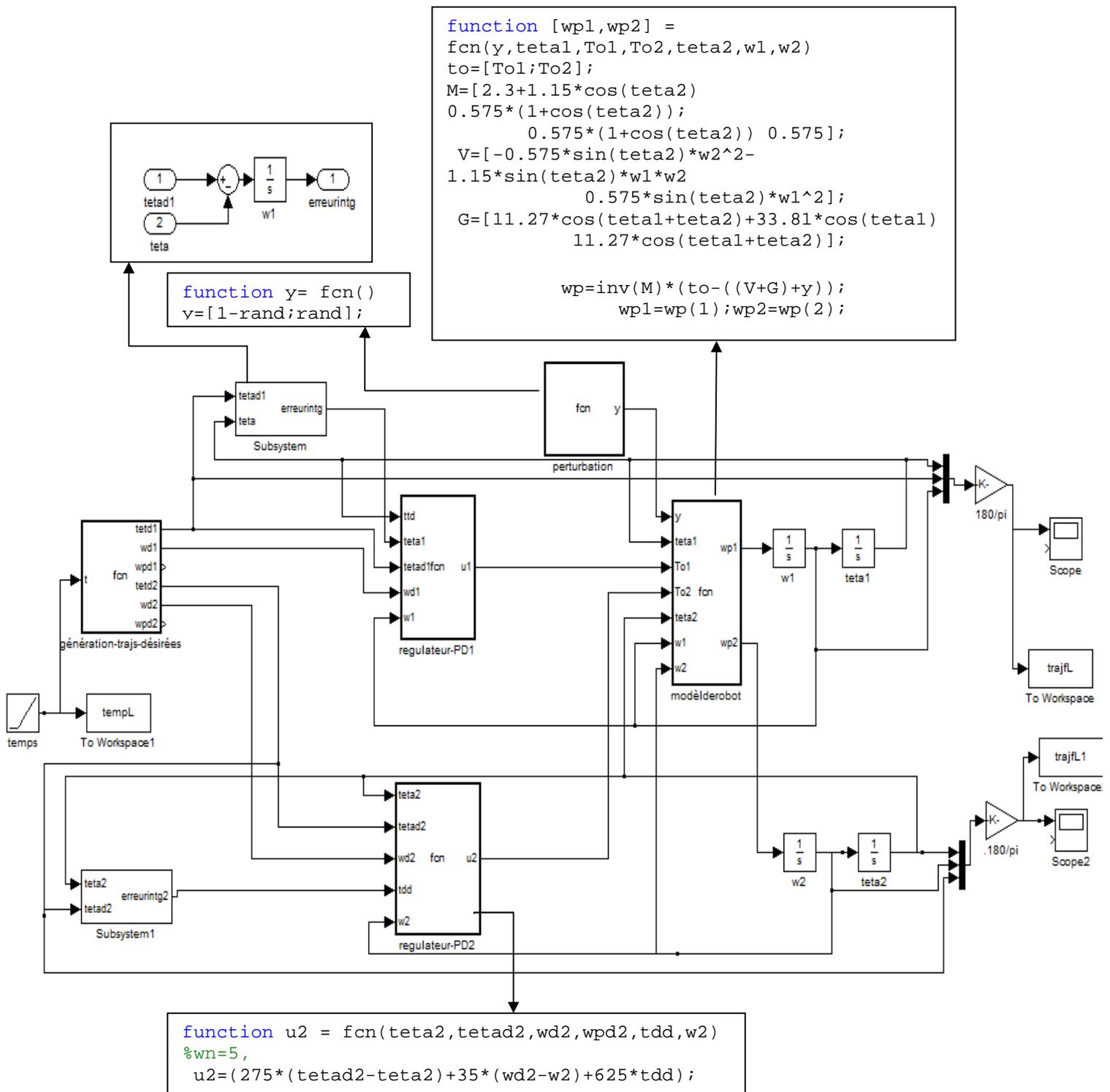


Figure B.3 : organigramme de la commande PID en poursuite

Annexe B

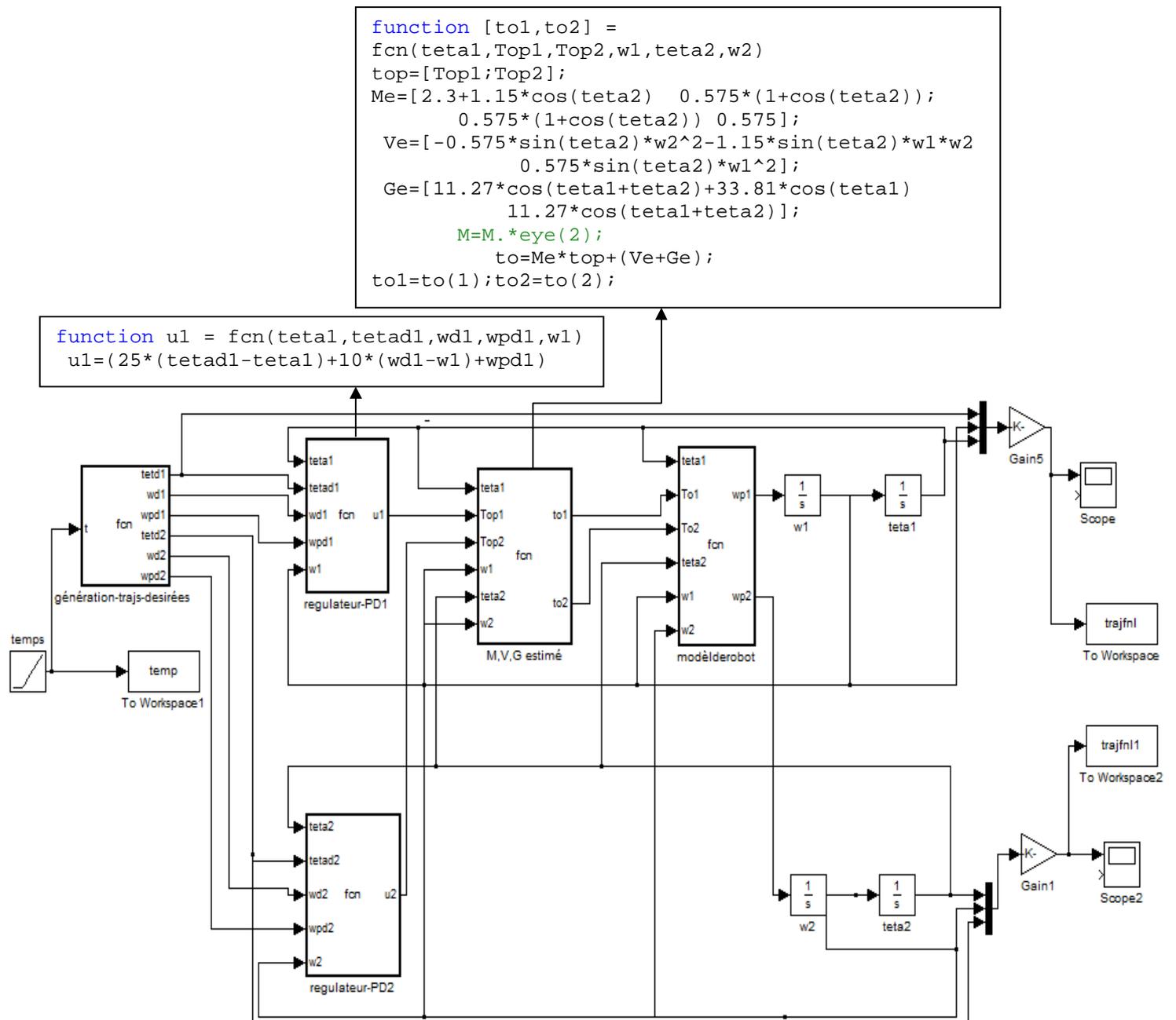


Figure B.4: organigramme de la commande non linéaire en poursuite