

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane MIRA Bejaia
Faculté de Technologie
Département de génie électrique



Mémoire de fin d'études
En vue de l'obtention du diplôme de Master en Automatique et
Informatique Industriel

Thème :

Commande floue dynamique d'un robot manipulateur

Réalisé par :

Mr. ACHERCHOUR Allaoua

Mr. ARDJOUNE Nassim

Encadré par :

Mr. Boubekur MENDIL

Promotion: 2017/2018

Remerciement

Nous tenons à remercier, tout d'abord, le dieu le tout puissant qui nous a donné la sante courage et la patience pour achever ce travail.

Nous tenons à remercier aussi notre promoteur, Mr B.MENDIL pour l'aide qu'il a apportée, ses conseils précieux, sa disponibilité, pour la confiance qu'il nous' a accordée et surtout pour son grand savoir qu'il nous a transmis tout au long de notre formation.

Nous remercions les membres de jury pour l'honneur qu'ils nous font en participant au jugement de notre travail.

Nous remercions tous les enseignant de département (**A.T.E**) de l'université de A-MIRA de Bejaia, qui ont participé à notre formation.

Nous remercions aussi tous nos amis, amies et tous ceux qui ont nous à aider de prés ou de loin, et pour leurs encouragement, conseils et soutiens.

Dédicaces

Je dédie ce modeste travail ;

À mes chers parents.

*À mes sœurs, à mes frères, mon binôme A. Nassim et sa famille, et
tous mes amis.*

*Et à tous ceux qui ont contribué de près ou de loin pour que ce projet
soit possible, je vous dis merci.*

A. Allaoua

Dédicaces

Je dédie ce modeste travail à :

A mes parents .Aucun hommage ne pourrait être à la hauteur de l'amour Dont ils ne cessent de me combler. Que dieu leur procure bonne santé et longue vie

A mon cher frère Yanis et a mes chères sœurs Yasmin et Meriem , je vous souhaite une vie pleine de bonheur et de succès et que Dieu, le tout puissant, vous protège et vous garde.

A mon binôme A. Alloua et sa famille.

A tous mes amis (es).

Et à tous ceux qui ont contribué de près ou de loin pour que ce projet soit possible, je vous dis merci.

A. Nassim .

Liste des figures

Chapitre I

Figure I.1: Structure fonctionnelle d'un robot.	3
Figure I.2: Structure sériel.	4
Figure I.3: Structure fermée.	5
Figure I.4: Architecture d'un bras de robot.	5
Figure I.5: Type d'articulation	6
Figure I.6: Structure d'un moteur électrique.	7
Figure I.7: Moteur a courant continu	9
Figure I.8: Moteur pas-à-pas.	10
Figure I.9: Actionneur hydraulique.	11
Figure I.10: Transmetteurs.	12
Figure I.11: Les capteurs.	12
Figure I.12: Robot cartésien.	13
Figure I.13: Robot cylindrique.	13
Figure I.14: Robot sphérique.	14
Figure I.15: Robot 3R.	14

Chapitre II

Figure II.1: Représentation cartésienne.	17
Figure II.2: Représentation d'une rotation autour l'axe Ox.	19
Figure II.3: Représentation d'une rotation autour de l'axe Oy.	20
Figure II.4: Représentation d'une rotation.	20
Figure II.5: Paramètres géométrique.	24
Figure II.6: Robot à structure ouverte simple.	25
Figure II.7: Plusieurs formes de chemin possible pour une seule articulation.	31

Chapitre III

Figure III.1: Structure du bras de robot étudié.	34
Figure III.2: Schéma bloc d'un système asservi.	36
Figure III.3: Schéma bloc d'un système asservi avec correcteur.	36
Figure III.4: Schéma-bloc du régulateur PD.	37
Figure III.5: Formes des variables articulaires et les vitesses angulaires.	40
Figure III.6: Formes des variables articulaires et les vitesses angulaires.	41
Figure III.7: Schéma-bloc du régulateur PD avec compensation de gravité.	42

Figure III.8: Formes des variables articulaires et les vitesses angulaires.	43
Figure III.9: Schéma-bloc du régulateur PID.	44
Figure III.10: Formes des variables articulaires et les vitesses angulaires.	46
Figure III.11: Schéma-bloc du régulateur PID avec compensation de gravité.	47
Figure III.12: Formes des variables articulaires et les vitesses angulaires.	48
Figure III.13: Structure d'un système flou.	49
Figure III.14: Structure générale de la commande floue.	51
Figure III.15: Formes des fonctions d'appartenance.	51
Figure III.16: Résultats de la commande floue.	54
Figure III.17: Résultats de commande floue avec compensation de gravité.	55
Figure III.18: Schéma synoptique de la commande PD en poursuite du bras de robot.	57
Figure III.19: Formes des variables articulaires et les vitesses angulaires.	58
Figure III.20: La forme de la trajectoire de position et de vitesse.	59
Figure III.21: La forme de la trajectoire de position et de vitesse.	60
Figure III.22: La forme de la trajectoire de position et de vitesse pour la commande PID en poursuite.	62
Figure III.23: La forme de la trajectoire de position et de vitesse.	63
Figure III.24: La forme de la trajectoire de position et de vitesse avec la commande floue en poursuite.	64

Liste des tableaux

Tableau III.1: Table de décision.	52
Tableau III.2: Table de décision réduite.	53

SOMMAIRE

Introduction générale.....	1
----------------------------	---

CHAPITRE I

GENERALITES SUR LES BRAS DE ROBOT

I.1 Introduction.....	2
I.2 Généralités	2
I.2.1 Définition d'un bras manipulateur	2
I.2.2 Eléments constitutifs d'un robot	3
I.2.2.1 Unité informationnelle	3
I.2.2.2 Unité opérationnelle.....	4
I.2.2.3 La structure mécanique articulée	4
I.3 Architecture d'un bras de robot	5
I.3.1 La base	5
I.3.2 Le porteur.....	6
I.3.3 Actionneurs de la robotique	7
I.3.3.1 Particularité des moteurs utilisés en robotique	7
I.3.3.2 Actionneurs électriques.....	7
I.3.3.2.1 Avant ages des actionneurs électriques	8
I.3.3.2.2 Types de moteurs utilises	8
I.3.3.3 Actionneur hydraulique	11
I.3.3.4 Actionneur pneumatique	11
I.3.4 Les transmetteurs	11
I.3.5 L'organe terminal	12
I.3.6 les capteurs.....	12
I.4 Classification des robots	13
I.4.1 Structure cartésienne (PPP).....	13
I.4.2 Structure cylindrique (RPP ou PRP).....	13
I.4.3 La structure sphérique (RRT ou RRP).....	14
I.4.4 La structure anthropomorphe (RRR)	14

I.4.5 LA structure SCARA	14
I.5 Caractéristiques des robots	15
I.6 Applications des robots manipulateurs	16
I.6.1 Robots industriels de soudage.....	16
I.6.2 Robots manipulateurs de service aux humains	16
I.6.3 Robots manipulateurs de service aux équipements	16
I.7 Conclusion	16

CHAPITRE II

MODELISATION DES BRAS DE ROBOTS

II.1 Introduction	17
II.2. Modélisation géométrique	17
II.2.1 La transformation homogène.....	17
II.2.1.1 Système de coordonnées décrivant la position d'un solide dans l'espace.....	17
II.2.1.2 Matrice de transformation homogène.....	18
II.2.2 Description géométrique.....	21
II.2.3 Modèle géométrique direct (MGD).....	21
II.2.4 Modèle géométrique inverse(MGI)	21
II.2.4.1 La méthode de Paul	23
II.2.5 Convention de Denavit-Hartenberg modifiée pour le calcul du MGD.....	24
II.3 Modélisation cinématique.....	26
II.3.1 Modèle cinématique direct	26
II.3.2 Modèle cinématique inverse	26
II.3.3 Forme générale du modèle cinématique	27
II.4 Modélisation dynamique	27
II.4.1 Modèle dynamique inverse	27
II.4.2 Modèle dynamique direct	28
II.4.3 L'approche de $E-L$	28
II.4.4 Formalisme de Newton-Euler.....	29
III.4.5 Vitesses des articulations d'un bras manipulateur.....	29
II.5 Génération de trajectoires	30
II.5.1 Méthode des polynômes cubique	30
II.6 Conclusion	32

CHAPITRE III

COMMANDE FLOUE D'UN BRAS DE ROBOT

III.1 Introduction	33
III.2 Technique de commande des robots	33
III.3 Modèle dynamique du robot utilisé.....	33
III.4 Linéarisation du système par compensation des non linéarités.....	34
III.5 Commande classique.....	36
III.5.1. Régulation linéaire	37
III.5.1.1 Régulation Proportionnel Dérivé(PD).....	37
III.5.1.2 Régulation Proportionnel Intégral Dérivé(PID).....	44
III.6 Commande floue en régulation	49
III.6.1 Rappels	49
III.6.2 La commande par la logique floue à un bras de deux degrés de liberté.....	51
III.7 Commande en poursuite.....	56
III.7.1 Commande PD en poursuite.....	56
III.7.2 La commande PID en poursuite	61
III.7.3 La commande floue en poursuite	63
III.8 Conclusion.....	65

Introduction générale

Un robot est une machine capable d'effectuer des tâches et de manipuler des objets selon un programme de façon automatique. Ils sont généralement utilisés pour remplacer les humains dans des situations où ces derniers sont incapables d'effectuer le travail, des situations plus dangereuses, de haute précision ou répétitives. Ils sont de plus en plus utilisés en industrie, car ils sont très rapides et précis.

Les robots industriels se distinguent souvent par un système articulé qui ressemble à un bras humain. Ce bras peut se déplacer, dans certains cas, sur plusieurs axes et est muni d'un outil propre à l'opération à effectuer : l'organe effecteur. Les applications typiques de ces robots en industrie sont le soudage, la peinture, la manipulation, l'assemblage et l'inspection et, généralement, des opérations (tâches) répétitives définies par l'utilisateur. Donc, à ce dernier de définir et de commander les mouvements des différentes articulations selon la tâche à réaliser.

Ce travail représente une introduction au domaine de la robotique et principalement la modélisation et la commande des robots manipulateurs. Il s'agit de l'application de la logique floue à la commande d'un bras à deux articulations, en régulation et en poursuite de trajectoires. Les résultats sont confrontés à ceux de certaines techniques classiques.

Le mémoire est constitué de trois chapitres. Dans le premier chapitre on présentera une introduction aux robots industriels et les éléments constitutifs d'un robot manipulateur tels que la structure mécanique, les actionneurs ou les servomoteurs et les capteurs.

Le deuxième chapitre expose les modélisations géométriques, cinématique et dynamique des robots, après avoir rappelé les principes et les méthodes de la modélisation.

Le dernier chapitre traite quelques techniques classiques de commande des robots manipulateurs telle que la commande PID et la commande dynamique qui sont comparées à la commande floue. On y trouve principalement l'exposé des méthodes et leur simulation pour un robot manipulateur rigide à deux degrés de liberté.

On terminera par une conclusion générale sur le contenu des trois chapitres et les résultats obtenus.

I.1 Introduction

Les outils et les machines ont souvent été pensés depuis longtemps, mais leur réalisation physique ne pouvait alors aboutir en raison notamment de l'absence de techniques ou matériaux disponibles pour les fabriquer ou pour dépasser le stade de la maquette. Il en est ainsi de l'idée de robot qui n'a trouvé son nom que dans les années 1920 (comme conséquence du succès de la pièce de théâtre du tchèque Karel Chapek : R.U.R. ou *Rossum's Universal Robots*), pour désigner alors de petits êtres artificiels anthropomorphes répondant parfaitement aux ordres de leurs maîtres, et sa première réalisation industrielle vers 1960.

La robotique peut être définie comme l'ensemble des techniques et études tendant à concevoir des systèmes mécaniques, informatiques ou mixtes, capables de se substituer à l'homme dans ses fonctions motrices et sensorielles.

I.2 Généralités

I.2.1 Définition d'un bras manipulateur

Un bras manipulateur est le bras d'un robot généralement programmable, avec des fonctions similaires à un bras humain. Les éléments (corps) de ce manipulateur sont reliés par des articulations permettant, soit des mouvements de rotation ou des translations (déplacements linéaires). Il peut être autonome ou contrôlé manuellement et peut être utilisé pour effectuer une variété de tâches avec une grande précision. Les bras manipulateurs peuvent être fixes ou mobiles (c'est-à-dire à roues) et peuvent être conçus pour des applications industrielles. Généralement, un robot manipulateur est considéré comme un système articulé rigide. Dans la littérature, on trouve différentes définitions, telle que celle donnée par JIRA (Japan Industriel Robot Association) : « Un robot est un système versatile doté d'une mémoire et pouvant effectuer des mouvements comme ceux d'un opérateur humain », ou celle donnée par RIA (Robot Institute of America) : « Un robot est un manipulateur à fonction multiple programmé pour réaliser automatiquement des tâches variées éventuellement répétitives ».

I.2.2 Eléments constitutifs d'un robot

De même qu'un trou n'est défini que par ce qui l'entoure, un robot n'a de sens que dans un environnement qu'il modifie [1]. C'est pourquoi on peut distinguer les ensembles Interactifs dans un robot en fonctionnement [2].

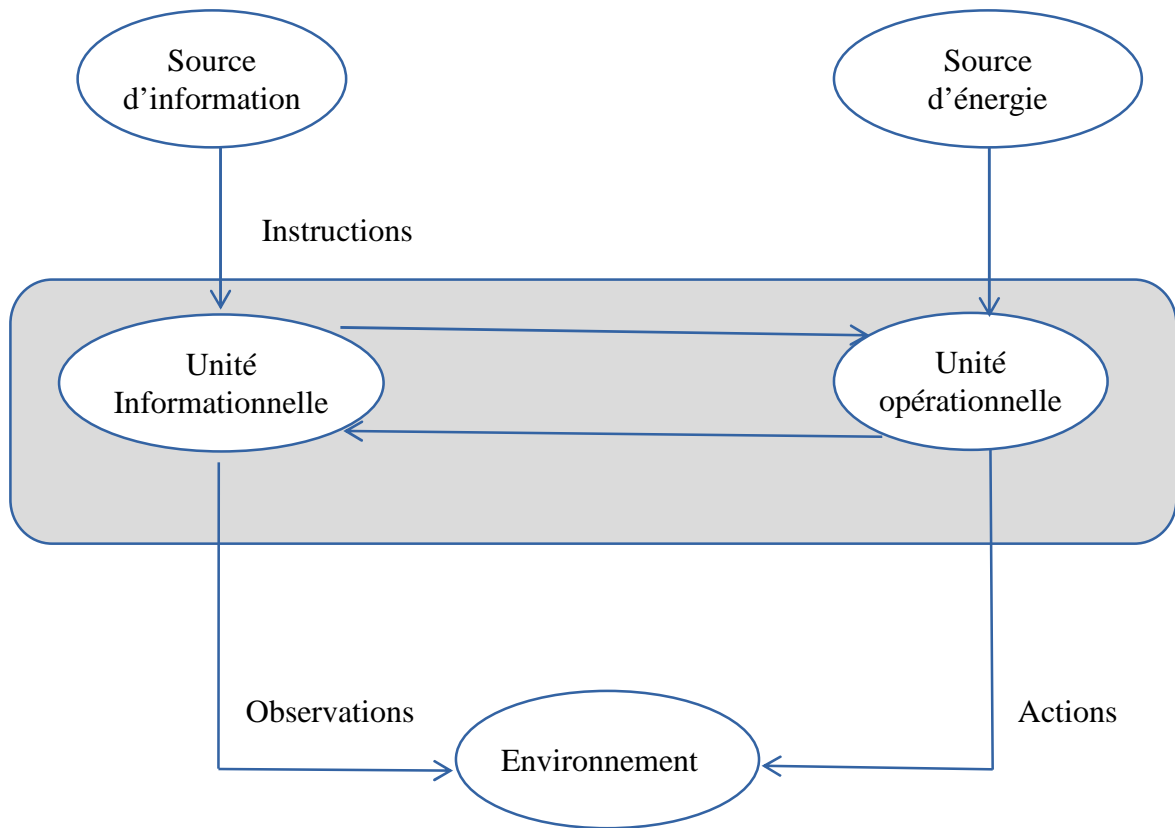


Figure I.1: Structure fonctionnelle d'un robot.

I.2.2.1 Unité informationnelle

Elle 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 ses tâches. Les systèmes actuels fonctionnent en interaction permanente selon le cycle information-décision-action [3].

I.2.2.2 Unité opérationnelle

Elle 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,...), les modules 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 permanence, leur positions et leur vitesses, et enfin l'effecteur ou organe terminal qui est en interaction avec l'environnement [3].

I.2.2.3 La structure mécanique articulée

Un robot manipulateur est constitué généralement de deux sous-ensembles distincts : un organe terminal qui est le dispositif destiné à manipuler des objets et une structure mécanique articulée (SMA), constituée d'un ensemble de solides reliés entre eux, généralement les uns à la suite des autres où chaque solide est mobile par rapport au précédent. Cette mobilité s'exprime en termes de degrés de liberté qui est par définition le nombre de mouvements indépendants possibles d'un solide C_1 par rapport au solide qui lui est directement relié C_2 .

Une structure mécanique articulée peut être représentée par une architecture composée de plusieurs chaînes de corps rigides assemblés par des liaisons appelées articulations. Les chaînes peuvent être dites soit *ouvertes* ou en série dans les quelles tous les corps ont de plus deux liaisons, ou bien *arborescentes* où au moins l'un des corps a plus de deux liaisons. Les chaînes peuvent aussi être fermées dans lesquelles l'organe terminal est relié à la base du mécanisme par l'intermédiaire de plusieurs chaînes [4].

- **Structure mécanique articulée à chaîne cinématique simple**

C'est une chaîne cinématique dont chaque membre possède un degré de connexion (nombre de liaisons mécaniques) inférieur ou égal à deux. Un robot sériel est formé d'une chaîne cinématique simple dont la base et l'organe effecteur possèdent un degré de connexion de un (c'est-à-dire qu'il n'est relié qu'à un seul corps) et les autres éléments un degré de connexion de deux.



Figure I.2: Structure sériel.

- **Structure mécanique articulée à chaîne cinématiques fermée**

C'est une chaîne cinématique dont l'un des membres, différent de la base, possède un degré de connexion supérieur ou égal à trois.



Figure I.3: *Structure fermée.*

I.3 Architecture d'un bras de robot

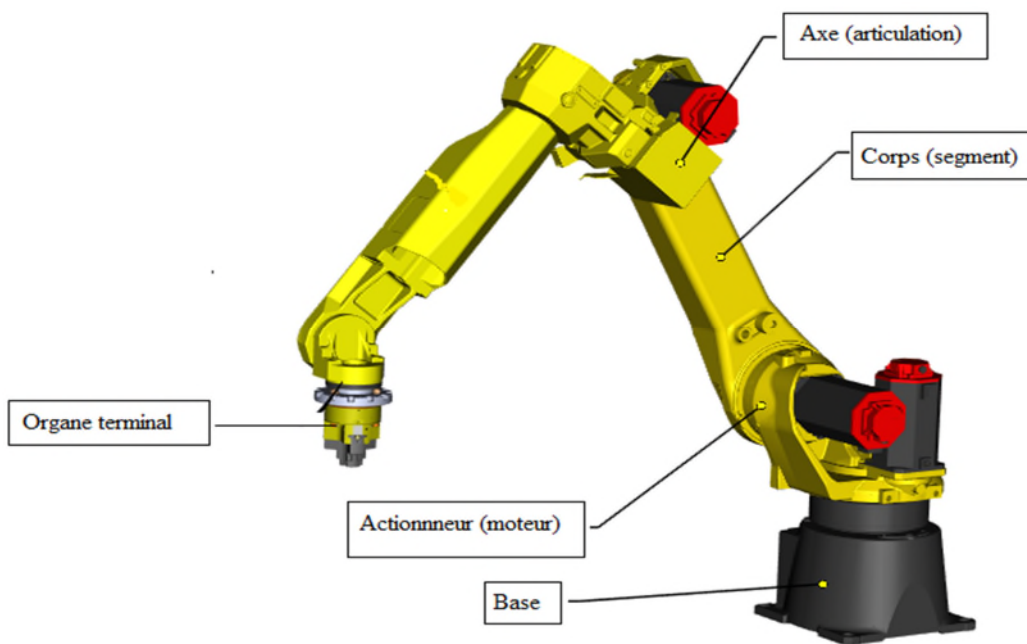


Figure I.4: *Architecture d'un bras de robot.*

I.3.1 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 [5].

I.3.2 Le porteur

Le porteur représente l'essentiel du système mécanique articulé. Il a pour rôle d'amener l'organe terminal dans une situation donnée imposée par la tâche. La situation d'un corps peut être définie comme la position et l'orientation d'un repère attaché à ce corps par rapport à un repère de référence.

Le porteur est constitué de :

- **Segments**

Ils sont des corps solides rigides susceptibles d'être en mouvement par rapport à la base du porteur et les uns par rapport aux autres,

- **Articulations**

Une articulation lie deux corps successifs en limitant le nombre de degrés de liberté, 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 [6].

-**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 commun. La situation relative entre les deux corps est donnée par l'angle autour de cet axe.

-**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.

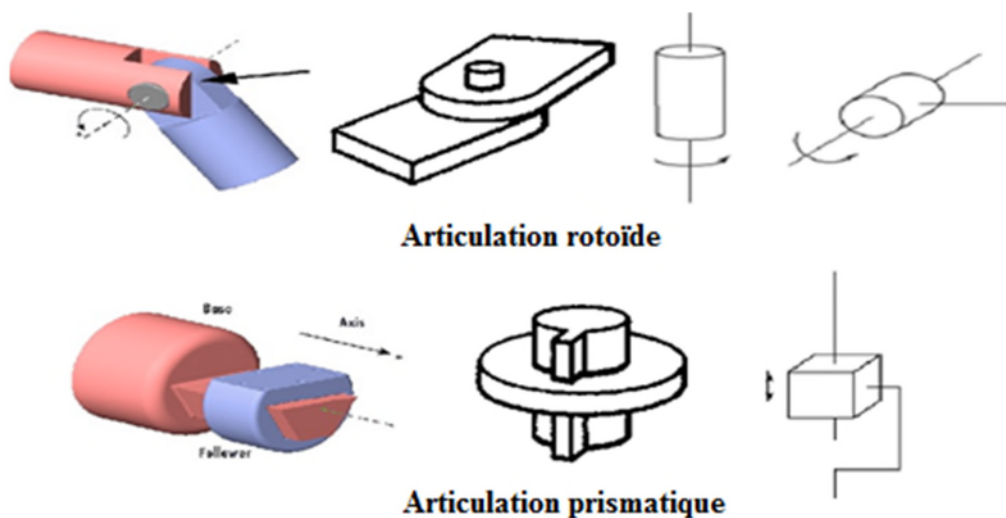


Figure I.5: Type d'articulation

I.3.3 Actionneurs de la robotique

I.3.3.1 Particularité des moteurs utilisés en robotique

Un robot réalise des fonctions mécaniques telles qu'un déplacement ou un positionnement. Pour cela, il a besoin d'un actionneur qui va réaliser à la fois une transmission d'information et une conversion d'énergie. Plusieurs technologies existent. On rencontre des actionneurs électriques, pneumatiques et hydrauliques. Les premiers sont le plus souvent des moteurs en rotation et un mécanisme permettant éventuellement de modifier la nature du mouvement. Les moteurs employés font appel aux mêmes principes que ceux qu'on utilise en électrotechnique classique, mais leurs caractéristiques et leurs technologies sont différentes. Tout d'abord, la puissance en général modeste des machines qu'on fait souvent appel à des aimants permanents plutôt qu'à des inducteurs bobinés. Ce qui simplifie la réalisation et élimine les pertes par effet Joule correspondantes.

Ensuite, on ne cherche pas à optimiser les mêmes paramètres. Les grandeurs liées à la conversion d'énergie (puissance, rendement) sont certes importantes, mais les grandeurs liées à l'information (précision, rapidité de réponse) sont primordiales. Ainsi, on minimise l'inertie des parties tournantes en adoptant des structures particulières ou une géométrie adaptée pour réaliser des moteurs à réponse rapide [7].

I 3.3.2 Actionneurs électriques

Un actionneur ou machine électrique est un dispositif électromécanique basé sur l'électromagnétisme permettant la conversion d'énergie électrique en énergie mécanique.

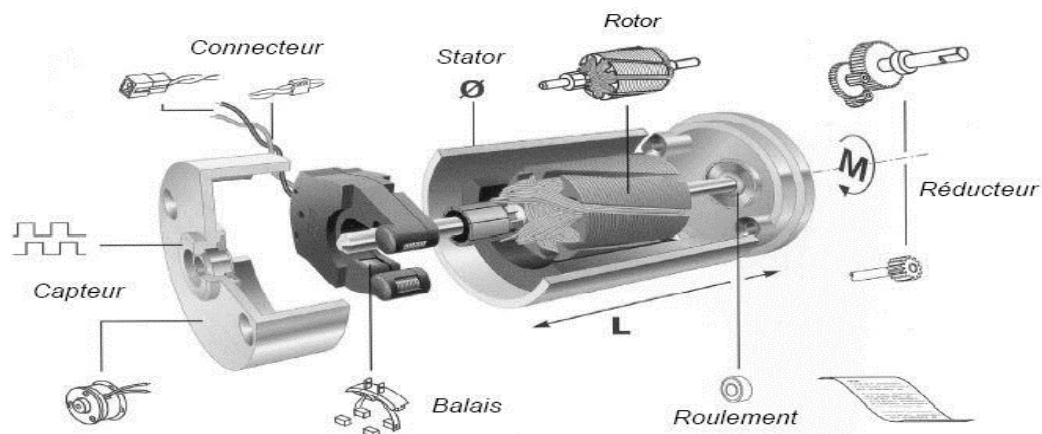


Figure I.6: Structure d'un moteur électrique.

I 3.3.2.1 Avantages des actionneurs électriques

Par rapport à leurs concurrents hydrauliques et pneumatiques, les actionneurs électriques présentent un certain nombre d'avantages parmi lesquels :

- une énergie facilement disponible, soit à partir du secteur, soit à partir de batterie pour les engins autonomes.
- une adaptation aisée de l'actionneur et de sa commande du fait de la nature électrique de l'ensemble des grandeurs.

Les progrès récents ont permis un élargissement du domaine d'emploi des actionneurs électriques. On peut citer :

- L'amélioration des performances des moteurs grâce en particulier à la disponibilité d'aimants plus efficaces et l'optimisation obtenue par l'utilisation de logiciels de simulation.
- la simplification de l'électronique associée grâce aux avancées dans le domaine des composants (circuits intégrés, transistors de puissance ...) [7].

I 3.3.2.2 Types de moteurs utilisés :

Les machines rencontrées dans le domaine de la robotique sont essentiellement de trois types :

- le moteur à courant continu représente la solution traditionnelle lorsqu'on a besoin de commander une vitesse ou une position, mais nécessite un asservissement.
- le moteur à courant continu sans balais (autopiloté) est en fait une machine alternative associée à un commutateur électronique qui peut remplacer le moteur à courant continu classique avec des caractéristiques similaires.
- le moteur pas-à-pas est à la base d'un actionneur de positionnement ne nécessitant pas d'asservissement, mais peut être aussi utilisé pour une commande de déplacement [7].

1. Moteur à courant continu: Le principe de la machine à CC repose sur l'application de force de Laplace sur des conducteurs solidaires du rotor est baignés dans une induction magnétique. La particularité de la machine à CC est qu'elle pourvue d'un système appelée «association balais/collecteur» qui permet de répartir les courant dans les conducteur du rotor suivant une disposition fixe qui ne dépend pas de rotation du rotor. En conséquence, cette peut produire un couple sur son rotor indépendant de la vitesse de rotation de ce dernière (théoriquement du moins). Cette particularité lui vaut, si elle est la machine posant le plus problèmes technologique de complications et d'usure, d'être celle qui propose le fonctionnement le plus simple et le plus linéaire [8].

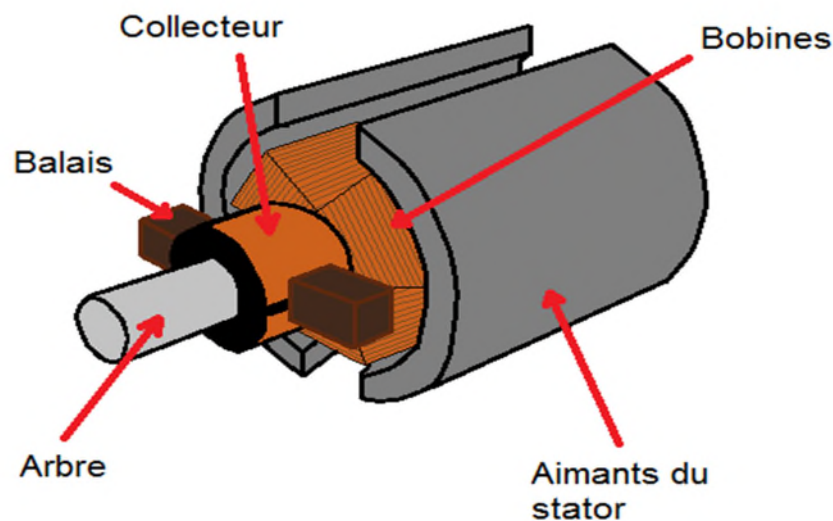


Figure I.7: *Moteur à courant continu*

2. Moteur synchrone autopilote: L'appellation de ces moteurs (héritée de l'anglais DC brushless motors) est trompeuse. Il ne s'agit en effet pas de moteurs à courant continu, mais de moteurs à courant alternatif : des moteurs synchrones auto-pilotés. Ils furent développés et commercialisés à partir de 1962, notamment pour palier les défauts des moteurs à courant continu, principalement dus au frottement des balais sur le collecteur. Les moteurs à courant continu sans balais ont ainsi un meilleur rendement, et des propriétés mécaniques meilleures. Ils offrent notamment de meilleurs couples massiques et des vitesses de rotation maximales plus grandes. Leur fonctionnement, sans frottements entre rotor et stator, leur confère une plus grande fiabilité et un temps de vie supérieur. Ils occasionnent moins de bruit de commutation et aucune étincelle. L'absence de bobinage au rotor permet souvent un refroidissement par simple convection, sans circulation forcée d'air vers l'extérieur. Ceci peut donc permettre une meilleure étanchéité. La vitesse de rotation reste limitée mécaniquement par les liaisons pivot entre bâti et rotor, et par l'échauffement qui a un effet sur l'efficacité des aimants. Ces moteurs présentent toutefois certains défauts (outre le prix), comme des effets d'ondulation de couple aux basses vitesses. Ces meilleures caractéristiques sont en revanche obtenues au prix d'un fonctionnement plus délicat à modéliser, et d'une électronique de commande plus complexe et plus coûteuse. Un moteur à courant continu sans balais est généralement équipé au rotor d'aimants placés dans le champ tournant produit par le bobinage triphasé du stator. De manière très grossière, on peut dire que le dispositif électronique de

commande du moteur sans balais se substitue au convertisseur électromécanique constitué de l'ensemble

balais+collecteur du moteur à courant continu. Pour cela, il est nécessaire de disposer de capteur à effet Hall pour repérer la position du rotor afin de réaliser l'auto-pilotage du moteur [9].

3. Moteur pas à pas : Un moteur pas-a-pas est une machine tournante dont le rotor se déplace d'un angle élémentaire α_p appelé pas chaque fois que son circuit de commande effectue une commutation de courant dans un ou plusieurs de ses enroulements. Il s'agit donc avant tout d'un actionneur de positionnement. Toutefois, une succession rapprochée de commutations permet d'obtenir une rotation continue. Compte tenu de son principe, la commande de la position ou de la vitesse d'un moteur pas-a-pas peut se faire sans asservissement: il n'est pas nécessaire de contrôler le résultat qui correspond exactement aux ordres donnés à condition de respecter certaines limites de fonctionnement [7].

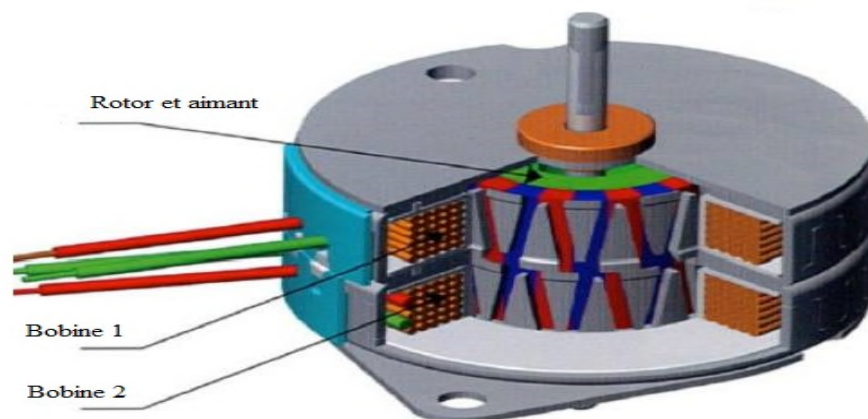


Figure I.8: Moteur pas-à-pas

- **Avantage**

Le moteur pas-a-pas constitue une solution simple et bon marché pour les commandes en position. Il est parfaitement adapté à une commande par un signal numérique. Par rapport aux moteurs à courant continu avec ou sans balais, l'avantage déterminant est que l'asservissement de la position n'est pas nécessaire.

- **Inconvénient**

Le rendement du moteur pas-a-pas est en général déplorable. Dans certaines conditions, il y a un risque de perte de pas. Pour les modes de commande élémentaires, les vibrations et les bruits acoustiques sont importants. Pour s'approcher d'une qualité de fonctionnement comparable à celle obtenue par un moteur à courant continu avec ou sans balais, il faut utiliser une commande plus

complexe ou un asservissement, mais les avantages de la simplicité et du faible coût disparaissent alors.

1.3.3.3 Actionneur hydraulique

Ce type d'actionneur est réservé pour des manipulations de très lourdes charges (par exemple, une pelle mécanique), agissant en translation (vérin hydraulique) ou en rotation (moteur hydraulique).

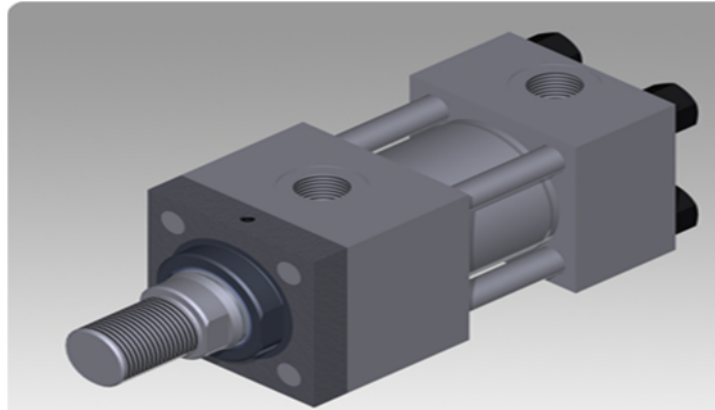


Figure I.9: *Actionneur hydraulique.*

1.3.3.4 Actionneur pneumatique

Les actionneurs pneumatiques sont d'un usage général pour les manipulateurs à cycles (robots tout ou rien). Un manipulateur à cycles est une structure mécanique articulée avec un nombre limité de degrés de liberté permettant une succession de mouvements contrôlés uniquement par des capteurs de fin de course réglables manuellement à la course désirée (asservissement en position difficile dû à la compressibilité de l'air) [6].

1.3.4 Les transmetteurs

Un transmetteur est un dispositif mécanique permettant de transmettre un mouvement d'une pièce à une autre. Cet élément de la chaîne d'énergie a pour fonction l'adaptation du couple et de la vitesse entre l'organe moteur et l'organe entraîné. Selon les mécanismes, la transmission est dimensionnée suivant des considérations concernant :

- la position d'une partie du mécanisme.
- le mouvement souhaité.
- la force ou le couple recherché.
- la puissance.



Figure I.10: *Transmetteurs.*

I.3.5 L'organe terminal

Il s'agit d'une interface permettant au robot d'interagir avec son environnement. Il peut être multifonctionnel, au sens où il peut être équipé de plusieurs dispositifs ayant des fonctionnalités différentes. Il peut aussi être monofonctionnel, mais interchangeable.

I.3.6 Les capteurs

Les capteurs sont des éléments qui ont pour objet de fournir des informations sur l'état de système. En robotique, on distingue deux types de capteurs [10][11]:

- **Les capteurs proprioceptifs:** destinés à donner les grandeurs réelles de la position, de la vitesse et de l'accélération du couple développé à chaque articulation, afin d'assurer, par asservissement, le bon suivi de trajectoires imposées.
- **Les capteurs extéroceptifs:** l'objectif est le prélèvement d'information sur l'environnement afin d'exécuter la tâche demandée (détection de présence, de contact, mesure de distance, vision artificielle). Suivent la tâche projetée et l'environnement, la nature et la quantité d'information à prélever varient largement.



Figure I.11: *Les capteurs.*

I.4 Classification des robots

On peut classer les robots d'un point de vue fonctionnel ou d'après leur structure géométrique.

I.4.1 Structure cartésienne (PPP)

Cette structure est constituée de trois axes qui sont animés d'un mouvement de translation. Elle est relativement peu utilisée, sauf dans quelques applications particulières telles que les robots portiques et les robots de magasinage.

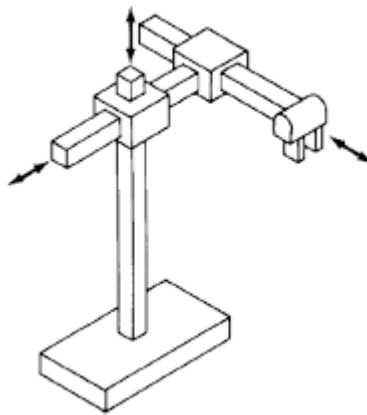


Figure I.12: Robot cartésien.

I.4.2 Structure cylindrique (RPP ou PRP)

Cette structure associe une rotation et une translation axiale, complétées par une translation radiale.

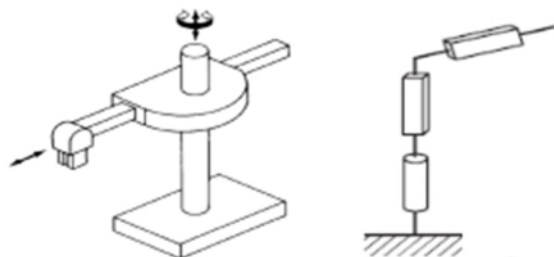


Figure I.13: Robot cylindrique.

I.4.3 La structure sphérique (RRT ou RRP)

Cette structure associe deux rotations (longitude et latitude) autour d'axes orthogonaux, complétées par une translation radiale.

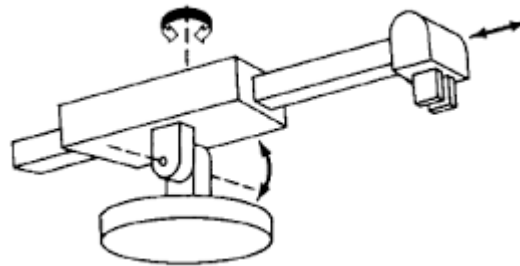


Figure I.14: Robot sphérique.

I.4.4 La structure anthropomorphe (RRR)

Le porteur est en coordonnées universelles, appelé aussi configuration poly articulée, trois rotations dont les deux dernières se font autour d'axes parallèles orthogonaux au premier. Les trois articulations correspondant respectivement au tronc (base), à l'épaule et au coude d'un être humain.



Figure I.15: Robot 3R.

I.4.5 La structure SCARA (Selective Compliance Arm for Robotic Assembly)

Cette architecture comprend deux rotations autour de deux axes parallèles, précédées ou suivies d'une translation dans la même direction (éventuellement celle-ci peut être reportée au niveau du poignet, qui peut aussi tourner autour du même axe, soit au total 4 ddm). Cette structure est utilisée dans des opérations d'assemblage.

I.5 Caractéristiques des robots

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

- *La charge maximum transportable* : de quelques kilos à quelques tonnes, à déterminer dans les conditions les plus défavorables (en élongation maximum).
- *L'architecture du S.M.A* : le choix est guidé par la tâche à réaliser.
- *Le volume de travail* : défini comme l'ensemble des points atteignables par l'organe terminal. Tous les mouvements ne sont pas possibles en tout point du volume de travail. 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, 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)
- *L'accélération.*
- *La masse du robot.*
- *Le coût du robot.*
- *La maintenance.*

I.6 Applications des robots manipulateurs

Dans les entreprises manufacturières, des tâches pénibles, répétitives réalisées par des opérateurs humains, peuvent être avantageusement confiées à des systèmes mécaniques articulés (les manipulateurs) dont la dextérité est sans égaler celle de l'homme, suffisamment proches de celui-ci pour exécuter des mouvements complexes à l'image de ceux d'un bras humain.

On peut citer quelques exemples des domaines d'application :

I.6.1 Robots industriels de soudage

Le soudage robotisé des châssis de voiture améliore la sécurité. Car, un robot ne manque jamais son point de soudure et le réalise toujours de la même manière tout au long de la journée.

I.6.2 Robots manipulateurs de service aux humains

Le manipulateur médical (MKM) produit par CARL ZEISS en Allemagne, consiste en un bras manipulateur à 6 ddl servo-contrôlés pour un contre-balancement du poids. Un ordinateur de contrôle et une station de travail graphique sont utilisés pour la visualisation et la programmation.

I.6.3 Robots manipulateurs de service aux équipements

Les robots peuvent également s'avérer utiles dans le domaine de l'aviation. En effet, le robot "Skywash" (Putzmeister Werke, Allemagne) peut diminuer par un facteur de 2 le temps de lavage d'un avion. Skywash intègre toutes les composantes d'un système robotique avancé : pré-programmation des mouvements à partir d'un modèle CAO de l'avion, localisation automatique des objets par des capteurs 3-D, un asservissement du mouvement par des capteurs tactiles, une architecture fortement redondante (11 ddl) installée sur une base mobile et une sécurité de fonctionnement maximale. Le manipulateur agit sous la supervision d'un être humain.

I.7 Conclusion

Dans ce chapitre, nous avons donné une idée générale sur la robotique, l'historique des robots, leurs structures, leurs utilisations et les différents types de robots ainsi que leurs Classifications et leurs domaines d'applications et cela va nous servir pour la construction de notre bras manipulateur qui va respecter les classifications et les définitions donner dans ce chapitre.

II.1 Introduction

La conception et la commande des robots nécessitent le calcul de certains modèles mathématiques, tels que :

- les modèles de transformation entre l'espace opérationnel (dans lequel est définie la situation de l'organe terminal) et l'espace articulaire (dans lequel est définie la configuration du robot). On distingue :
 - les modèles géométriques direct et inverse qui expriment la situation de l'organe terminal en fonction des variables articulaires du mécanisme et inversement.
 - les modèles cinématiques direct et inverse qui expriment la vitesse de l'organe terminal en fonction des vitesses articulaires et inversement.
- les modèles dynamiques définissant les équations du mouvement du robot, qui permettent d'établir les relations entre les couples ou forces exercés par les actionneurs et les positions, vitesses et accélérations des articulations [12].

II.2. Modélisation géométrique

II.2.1 La transformation homogène

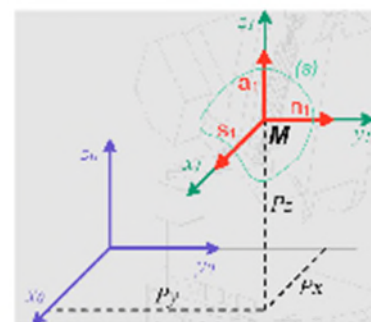
II.2.1.1 Système de coordonnées décrivant la position d'un solide dans l'espace

On choisit souvent le système de coordonnées qui convient le mieux à l'architecture du robot et la nature de la tâche.

a) Système de coordonnées cartésiennes :

Il permet d'obtenir d'une manière directe les composantes du vecteur de position P.

$$P = \begin{bmatrix} Px \\ Py \\ Pz \end{bmatrix}$$



Position d'un point

Figure II.1: Représentation cartésienne

b) Système de coordonnées cylindriques :

Cette représentation donne les composantes du vecteur de position tel que :

$$P = [r \cos \theta \quad r \sin \theta \quad z]^T$$

La relation qui lie les coordonnées cylindrique et cartésiennes est donnée par :

$$\begin{cases} r = \sqrt{Px^2 + Py^2} \\ \theta = \arctg \frac{Py}{Px} \\ z = Pz \end{cases}$$

c) Système de coordonnées sphériques :

Cette représentation donne le vecteur de position tel que :

$$P = [r \cos \theta \sin \phi \quad r \sin \theta \sin \phi \quad r \cos \phi]^T$$

La relation qui lie les coordonnées sphériques et cartésiennes est tel que :

$$\begin{cases} r = \sqrt{Px^2 + Py^2 + Pz^2} \\ \theta = \arctg \frac{Py}{Px} \\ \phi = \arctg \frac{\sqrt{Px^2 + Py^2}}{Pz} \end{cases}$$

II.2.1.2 Matrice de transformation homogène

Elle est utilisée pour décrire le lien géométrique entre les objets dans l'espace. Elle permet d'associer à un point, une position et une orientation dans l'espace.

Une matrice de transformation ${}^nT_m(4X4)$ permet le passage d'un repère R_m à un repère R_n par rotation et translation :

$${}^nT_m = \begin{bmatrix} S_x & N_x & A_x & P_x \\ S_y & N_y & A_y & P_y \\ S_z & N_z & A_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ avec } P = \begin{bmatrix} Px \\ Py \\ Pz \\ 1 \end{bmatrix} \text{ et } \text{Rot}(r,\theta) = \begin{bmatrix} S_x & N_x & A_x \\ S_y & N_y & A_y \\ S_z & N_z & A_z \end{bmatrix}$$

$\text{Rot}(r,\theta)$, est dite matrice d'orientation du référentiel R_m relative au référentiel R_n .

Le vecteur P est dit vecteur de position, les trois premières composantes représentent les coordonnées cartésiennes de l'origine du référentiel R_m par rapport référentiel R_n et la quatrième composante mise à 1 et dite facteur d'échelle.

Dans ce qui suit, on considère le repère fixe associé à la base du robot.

- **Matrice de translation**

On définit une matrice de translation qui permet de représenter le repère R_1 par rapport au repère R_0 comme suit :

$$\text{Trans}(P_x, P_y, P_z) = \begin{bmatrix} 1 & 0 & 0 & P_x \\ 0 & 1 & 0 & P_y \\ 0 & 0 & 1 & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Avec P_x, P_y, P_z les coordonnées de l'origine du repère R_1 dans le repère R_0 .

- **Matrice de rotation**

Matrice de rotation autour de l'axe OX : on définit une matrice de rotation, d'un angle α autour de l'axe OX_0 , permettant de représenter le repère R_1 par rapport au repère R_0 comme suit :

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

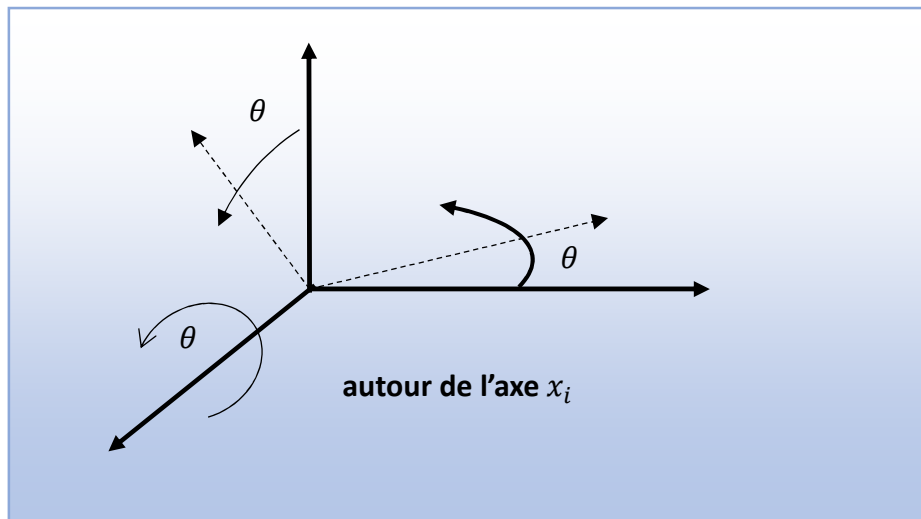


Figure II.2: Représentation d'une rotation autour l'axe Ox .

Matrice de rotation autour de l'axe OY : on définit la matrice de transformation de rotation, d'un angle φ autour de l'axe OY_0 , permettant l'expression du repère R_1 dans le repère R_0 par :

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

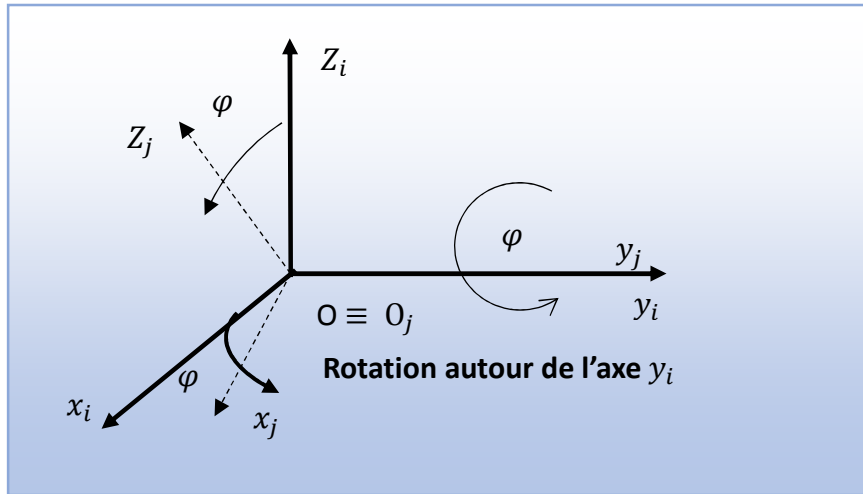


Figure II.3: Représentation d'une rotation autour de l'axe Oy .

Matrice de rotation autour de l'axe OZ : on définit la matrice de transformation de rotation, d'un angle ψ autour de l'axe OY_0 , permettant l'expression du repère R_1 dans le repère R_0 par :

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

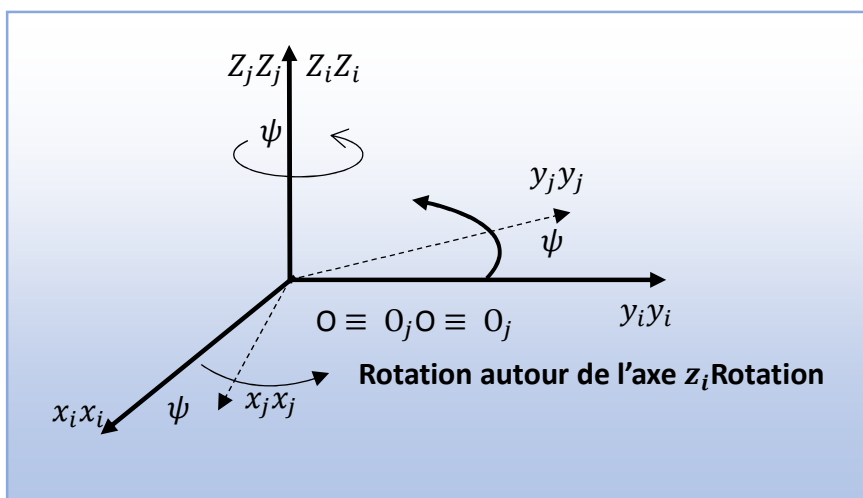


Figure II.4: Représentation d'une rotation.

II.2.2 Description géométrique

La modélisation des robots de façon systématique et automatique exige une méthode adéquate pour la description de leur morphologie. Plusieurs méthodes et notations ont été proposées. La plus répandue est celle de *Denavit-Hartenberg* [12]. Dans les années 1950, les messieurs *Jacques Denavit* et *Richard Hartenberg* ont eu l'excellente idée de proposer une méthode simple et systématique pour placer des référentiels sur chaque lien d'un mécanisme sériel qui facilite énormément le calcul des matrices de transformation [13].

II.2.3 Modèle géométrique direct (MGD)

Le modèle géométrique direct (MGD) est l'ensemble des relations qui permettent d'exprimer la situation de l'organe terminal, c'est-à-dire les coordonnées opérationnelles du robot, en fonction de ses coordonnées articulaires. Dans le cas d'une chaîne ouverte simple, il peut être représenté par la matrice de transformation 0T_n :

$${}^0T_n = {}^0T_1(q_1) {}^1T_2(q_2) \cdots {}^{n-1}T_n(q_n) \quad (\text{II.1})$$

Le modèle géométrique direct du robot peut aussi être représenté par la relation :

$$X = F(q) \quad (\text{II.2})$$

q étant le vecteur des variables articulaires tel que :

$$q = [q_1 \ q_2 \ \cdots \ q_n]^T \quad (\text{II.3})$$

Les coordonnées opérationnelles sont définies par :

$$X = [X_1 X_2 \cdots X_m]^T \quad (\text{II.4})$$

Plusieurs possibilités existent pour la définition du vecteur X . Par exemple, avec les éléments de la matrice 0T_n :

$$X = [n_x n_y n_z \ o_x o_y o_z \ a_x a_y a_z \ p_x p_y p_z] \quad (\text{II.5})$$

Enfin, si nous avons un référentiel outil, et un référentiel atelier, la pose du premier par rapport au dernier est définie par l'équation suivante:

$${}^{atelier}T_{outil} = {}^{atelier}T_{outil} {}^0T_{nl} {}^nT_{outil} \quad (\text{II.6})$$

II.2.4 Modèle géométrique inverse(MGI)

On a vu que le modèle géométrique direct d'un robot permettait 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. Lorsqu'elle existe,

la forme explicite qui donne toutes les solutions possibles (il y a rarement unicité de solution) constitue ce que l'on appelle le modèle géométrique inverse. On peut distinguer trois méthodes de calcul du MGI [12] :

- La méthode de *Paul* qui traite séparément chaque cas particulier et convient pour la plupart des robots industriels.
- La méthode de *Pieper* qui permet de résoudre le problème pour les robots à six degrés de liberté possédant trois articulations rotoïdes d'axes concourants ou trois articulations prismatiques.
- La méthode générale de *Raghavan et Roth*, donnant la solution générale des robots à six articulations à partir d'un polynôme de degré au plus égal à 16.

Quand le modèle géométrique inverse ne peut pas être obtenu, des techniques numériques peuvent être employées. Ces techniques emploient la méthode de *Newton-Raphson* ou des méthodes fondées sur la transposée de la matrice Jacobéenne.

Soit ${}^{i-1}T_E^d$ la matrice de transformation homogène représente la position désirée du repère outil R_E par rapport au repère atelier R_f . En général, on peut exprimer ${}^fT_E^d$ sous la forme :

$${}^fT_E^d = Z {}^0T_n(q)E \tag{II.7}$$

Tels que :

- Z est la matrice de transformation définissant le repère de base du robot dans le repère atelier R_f .
- 0T_n est la matrice de transformation du repère terminal dans le repère R_0 .
- E est la matrice de transformation du repère outil R_E dans le repère terminal R_n .

Mettant tous les termes connus dans le membre gauche, on obtient :

$$U_0 = {}^0T_n(q) \tag{II.8}$$

Avec :

$$U_0 = Z^{-1} {}^fT_E^d E^{-1}$$

Le MGD donnant $X = F(q), q = [q_1 q_2 \dots q_n]^T$ $X = [X_1 X_2 \dots X_m]^T$, où n est le nombre de coordonnées opérationnelles et m le nombre de coordonnées articulaires, le problème au-dessus (II.10) s'agit de résoudre un système de m équations à n inconnues, ce système étant non linéaire.

Le nombre de solutions dépend de l'architecture du robot manipulateur et de l'amplitude des articulations.

Trois cas se présentent pour calculer le MGI :

1. Solutions en nombre fini.
2. Absence de solution, lorsque la position de l'organe terminal désirée est en dehors de la zone accessible du robot.
3. Infinité de solutions lorsque :
 - Le robot est redondant vis-à-vis la tâche.
 - Le robot se trouve dans certaines configurations singulières.

Lorsqu'il est possible de calculer toutes les configurations permettant d'atteindre une situation donnée d'un robot manipulateur ce dernier est dit résoluble.

II.2.4.1 La méthode de Paul

Soit la matrice de transformation homogène d'un robot manipulateur (II.1) [12] :

$${}^0T_n = {}^0T_1(q_1) {}^1T_2(q_2) \dots {}^{n-1}T_n(q_n)$$

Soit U_0 la situation désirée telle que :

$$U_0 = \begin{bmatrix} S_x & N_x & A_x & P_x \\ S_y & N_y & A_y & P_y \\ S_z & N_z & A_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{II.9}$$

On cherche à résoudre le système d'équations suivant :

$$U_0 = {}^0T_1(q_1) {}^1T_2(q_2) \dots {}^{n-1}T_n(q_n) \tag{II.10}$$

Pour résoudre le système (II.10), Paul a proposé une méthode (Méthode de Paul) qui consiste à pré multiplier successivement les deux membres de l'équation par les matrices ${}^iT_{i-1}$ pour i allant de 1 à $n-1$. Ces opérations permettent d'isoler les variables d'articulations l'une après l'autre.

$$\begin{aligned} U_0 &= {}^0T_1 {}^1T_2 {}^2T_3 \dots {}^{n-1}T_n \\ {}^1T_0 U_0 &= {}^0T_1 {}^1T_2 {}^2T_3 \dots {}^{n-1}T_n \\ {}^2T_1 U_0 &= {}^0T_1 {}^1T_2 {}^2T_3 \dots {}^{n-1}T_n \\ {}^3T_2 U_0 &= {}^0T_1 {}^1T_2 {}^2T_3 \dots {}^{n-1}T_n \\ &\vdots \\ {}^{n-1}T_{n-2} U_{n-2} &= {}^{n-1}T_n U_{j+1} = {}^{j+1}T_n = {}^{j+1}T_n U_j, \text{ pour } j = 0, \dots, n. \end{aligned}$$

Pour un robot à six degrés de liberté, on procède comme suit :

$$\begin{aligned}
 U_0 &= {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 \\
 {}^1T_0 U_0 &= {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 \\
 {}^2T_1 {}^1T_0 U_0 &= {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 \\
 {}^3T_2 {}^2T_1 {}^1T_0 U_0 &= {}^3T_4 {}^4T_5 {}^5T_6 \\
 {}^4T_3 {}^3T_2 {}^2T_1 {}^1T_0 U_0 &= {}^4T_5 {}^5T_6 \\
 {}^5T_4 {}^4T_3 {}^3T_2 {}^2T_1 {}^1T_0 U_0 &= {}^5T_6
 \end{aligned}$$

II.2.5 Convention de Denavit-Hartenberg modifiée pour le calcul du MGD

Les paramètres de la convention *Denavit-Hartenberg* modifiée permettent de disposer d'un paramétrage des liaisons tel que les matrices de passage aient toutes la même forme littérale, ce qui facilite les calculs.

La méthode qui suit s'applique lorsque le robot correspond à une chaîne simple ouverte et que ses articulations sont rotoïdes, ou prismatiques (ce qui est le cas en général). Les corps constituant le robot sont supposés parfaitement rigides et connectés par des articulations idéales (pas de jeu mécanique, pas d'élasticité) [6].

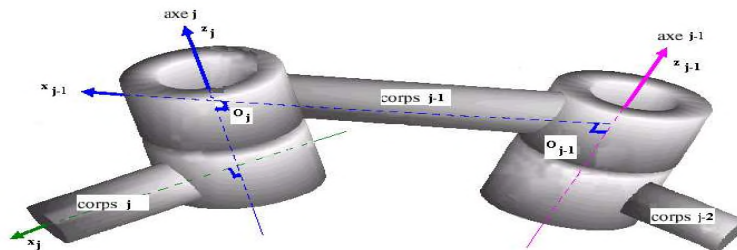


Figure II.5: Paramètres géométrique

Le passage du repère R_{j-1} au repère R_j se fait à travers 4 étapes :

- Une rotation d'angle α_j autour de l'axe $\overrightarrow{x_{j-1}}$.
- Une translation de distance d_j mesurée le long de l'axe $\overrightarrow{x_{j-1}}$.
- Une rotation d'angle θ_j autour de l'axe $\overrightarrow{z_j}$.
- Une translation de distance r_j mesurée le long de l'axe $\overrightarrow{z_j}$.

Avec :

- α_j l'angle de rotation entre les axes $\overrightarrow{z_{j-1}}$ et $\overrightarrow{z_j}$ autour de l'axe $\overrightarrow{x_{j-1}}$.
- d_j la distance entre $\overrightarrow{z_{j-1}}$ et $\overrightarrow{z_j}$ mesurée le long de l'axe $\overrightarrow{x_{j-1}}$.
- θ_j l'angle de rotation entre les axes $\overrightarrow{x_{j-1}}$ et $\overrightarrow{x_j}$ autour de l'axe $\overrightarrow{z_j}$.
- r_j la distance entre $\overrightarrow{x_{j-1}}$ et $\overrightarrow{x_j}$ mesurée le long de l'axe $\overrightarrow{z_j}$.

La variable articulaire q_j , associée à l'articulation j , est soit θ_j , soit, selon que cette articulation est respectivement de type rotoïde ou prismatique, ce qui se traduit par la relation :

$$q_j = (1 - \sigma_j)\theta_j + \sigma_j r_j$$

La matrice de transformation homogène définissant le R_j dans le repère R_{j-1} est donnée par :

$$T_{j-1,j} = Rot(x_j, \alpha_j). Trans(x_{j-1}, d_j). Rot(z_j, \theta_j). Trans(z_j, r_j)$$

$$T_{j-1,j} = \begin{bmatrix} C\theta_j & -S\theta_j & 0 & d_j \\ C\alpha_j S\theta_j & C\alpha_j C\theta_j & -S\alpha_j & -r_j S\alpha_j \\ S\alpha_j S\theta_j & S\alpha_j C\theta_j & C\alpha_j & -r_j C\alpha_j \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

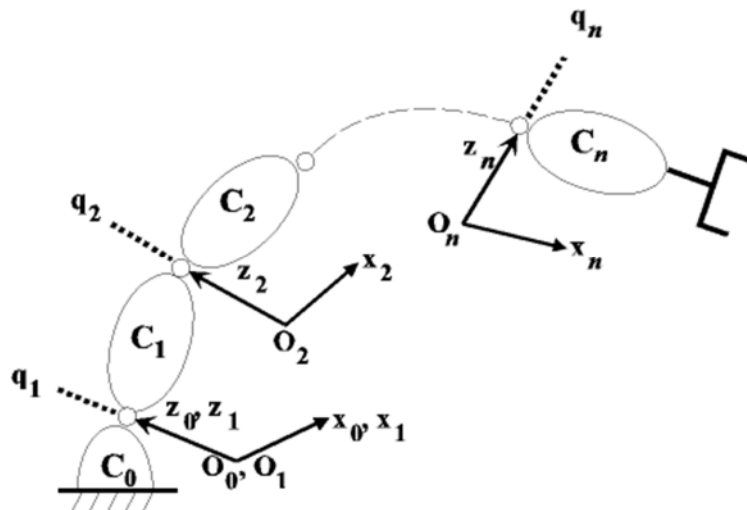


Figure II.6: Robot à structure ouverte simple.

II.3 Modélisation cinématique

II.3.1 Modèle cinématique direct

Le modèle cinématique direct d'un robot manipulateur décrit les vitesses des coordonnées opérationnelles en fonction des vitesses articulaires. Il est noté [12] :

$$\dot{X} = J(q)\dot{q} \tag{II.11}$$

Où $J(q)$ désigne la matrice jacobéenne de dimension $(m \times n)$ du mécanisme, égale à $\frac{\partial X}{\partial q}$ et fonction de la configuration articulaire q . 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 élémentaires des coordonnées articulaires $d\mathbf{q}$, soit :

$$d\mathbf{X} = J(\mathbf{q}) d\mathbf{q}$$

L'intérêt de la matrice jacobéenne est multiple :

- elle est à la base du modèle différentiel inverse, permettant de calculer une solution locale des variables articulaires \mathbf{q} connaissant les coordonnées opérationnelles \mathbf{X} .
- en statique, on utilise le jacobéen pour établir la relation liant les efforts exercés par l'organe terminal sur l'environnement aux forces et couples des actionneurs.
- elle facilite le calcul des singularités et de la dimension de l'espace opérationnel accessible du robot.

II.3.2 Modèle cinématique inverse

L'objectif du modèle cinématique inverse 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. Cette définition est analogue à celle du modèle différentiel inverse. Ce dernier permet de déterminer la différentielle articulaire $d\mathbf{q}$ correspondant à une différentielle des coordonnées opérationnelles $d\mathbf{X}$ spécifiée. Pour obtenir le modèle cinématique inverse, on inverse le modèle cinématique direct en résolvant un système d'équations linéaires. La mise en œuvre peut être faite de façon analytique ou numérique :

1. la solution analytique a pour avantage de diminuer considérablement le nombre d'opérations, mais on doit traiter séparément tous les cas singuliers.
2. les méthodes numériques sont plus générales, la plus répandue étant fondée sur la notion de pseudo-inverse.
3. les algorithmes traitent de façon unifiée les cas réguliers, singuliers et redondants. Ils nécessitent un temps de calcul relativement important.

Nous présentons dans ce paragraphe les techniques à mettre en œuvre pour établir un modèle cinématique inverse dans les cas réguliers, singuliers et redondants.

II.3.3 Forme générale du modèle cinématique

Soit $\mathbf{X} = [\mathbf{X}_p^T \ \mathbf{X}_r^T]^T$ une représentation quelconque dans le repère R_0 de la situation du repère R_n fixé à un solide, les éléments \mathbf{X}_p et \mathbf{X}_r désignant respectivement la position et l'orientation opérationnelles du solide. Les relations entre les vitesses $\dot{\mathbf{X}}_p$ et $\dot{\mathbf{X}}_r$ et les vecteurs vitesses V_n^0 et ω_n^0 du repère R_n sont telles que :

$$\begin{bmatrix} \dot{\mathbf{X}}_p \\ \dot{\mathbf{X}}_r \end{bmatrix} = \begin{bmatrix} \Omega & 0_3 \\ 0_3 & \Omega \end{bmatrix} \begin{bmatrix} V_n^0 \\ \omega_n^0 \end{bmatrix} = \Omega \begin{bmatrix} V_n^0 \\ \omega_n^0 \end{bmatrix} \quad (\text{II.12})$$

Les matrices Ω_p et Ω_r dépendant de la représentation choisie respectivement pour la position et pour l'orientation. Le modèle cinématique direct a pour forme générale :

$$\dot{\mathbf{X}} = \mathbf{J} \dot{\mathbf{q}}$$

Dans le cas régulier la matrice jacobéenne \mathbf{J} est carrée d'ordre n et son déterminant est non nul. La méthode la plus générale consiste à calculer \mathbf{J}^{-1} , la matrice inverse de \mathbf{J} , qui permet de déterminer les vitesses articulaires $\dot{\mathbf{q}}$ grâce à la relation :

$$\dot{\mathbf{q}} = \mathbf{J}^{-1} \dot{\mathbf{X}} \quad (\text{II.13})$$

II.4 Modélisation dynamique

II.4.1 Modèle dynamique inverse

Les équations dynamiques du mouvement d'un robot manipulateur sont l'ensemble d'équations mathématiques décrivant le comportement dynamique des robots. De telles équations sont utilisables pour la simulation du mouvement du robot manipulateur.

Le modèle dynamique est la relation entre les couples (et/ou forces) appliquée aux actionneurs et les positions, vitesse et accélération articulaires. On représente donc le modèle dynamique par une relation de la forme:

$$\tau = f(q, \dot{q}, \ddot{q}, f_e) \quad (\text{II.14})$$

- τ : Vecteur des couples/ forces des actionneurs, selon que l'articulation est rotoïde ou prismatique.
- q : Vecteur de position articulaire.
- \dot{q} : Vecteur de vitesse articulaire.
- \ddot{q} : Vecteur d'accélération articulaire.
- f_e : Vecteur représentant l'effort extérieur (force et moment) qu'exerce le robot sur l'environnement.

II.4.2 Modèle dynamique direct

Le modèle dynamique direct exprime la position, la vitesse et l'accélération des articulations en fonction du couple appliqué. Il est obtenu par inversion du modèle précédent. Il s'écrit [14] :

$$\ddot{q} = f(q, \dot{q}, \tau, f_e) \quad (\text{II.15})$$

Le modèle dynamique des robots peut être obtenu par des lois mécaniques de *Newton* et de *Lagrange*. Le formalisme d'E-L nous permet d'aboutir aux équations du mouvement du robot.

II.4.3 L'approche d'E-L

Le formalisme d'E-L est utilisé pour modéliser le comportement dynamique d'un robot. Cette approche particulière est assez simple à mettre en œuvre et elle est bien adaptée de calcul manuel ainsi qu'aux méthodes de calcul assistées par ordinateur.

La dérivation des équations dynamiques d'un robot manipulateur à n degrés de liberté est basée sur la compréhension de:

1. La matrice de transformation de coordonnées homogènes ${}^{i-1}T_i$ qui décrit la relation spatiale entre le i^{eme} et le $(i-1)^{eme}$ système de coordonnées. Elle relie un point fixe dans le corps i exprimé en coordonnées homogène par rapport au i^{eme} système de coordonnées au $(i-1)^{eme}$ système de coordonnées.

2. Formulation LE

Elle est basée sur l'équation du mouvement de Lagrange-Euler :

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \tau_i \quad i = 1, 2, \dots, n \quad (\text{II.16})$$

Avec

n : Nombre de d.d.l du bras.

$L = E - U$: fonction de Lagrange.

E : énergie cinétique totale du bras.

U : énergie potentielle totale du bras.

τ_i : force généralisée appliquée au système à l'articulation i .

q_i : Variable articulaire généralisée.

II.4.4 Formalisme de Newton-Euler

Cette formulation est basée sur des équations récursives et décrit le mouvement du bras en appliquant le principe de d'Alembert pour de chaque articulation. Ce dernier considère les conditions d'équilibre statique dans des situations dynamiques en tenant compte des forces externes appliquées et celles induites par réaction. Le lien entre les forces, les inerties et les accélérations est exprimé par l'équation de Newton et celle d'Euler [15].

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

$$F_i = m_i \cdot \vec{a}_i \tag{II.17}$$

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

$$N_i = I_i \dot{\omega}_i + \omega_i \times I_i \omega_i \tag{II.18}$$

Les calculs s'effectuent en deux étapes :

- Propagation directe des vitesses et des accélérations de la base vers le poignet.
- Rétro propagation des forces et des couples exercés, du poignet vers la base.

III.4.5 Vitesses des articulations d'un bras manipulateur

La formulation de Lagrange-Euler requière la détermination de l'énergie cinétique du système physique, ce qui revient à déterminer les vitesses des différentes articulations.

Nous définissons les points suivants :

r_i^i : Point fixe qui repose sur le segment i de coordonnées homogènes définies par rapport au repère R_i .

r_i^0 : définit le même point r_i^i de coordonnées homogènes définies par rapport au repère R_0 .

Tel que :

$$r_i^i = \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = (x_i, y_i, z_i, 1)^T$$

Ce qui nous permet d'écrire : $r_i^0 = T_i^0 r_i^i$

La vitesse du point r_i^i est donnée par :

$$v_i^0 = v_i = \frac{d}{dt} r_i^0 = \left(\sum_{j=0}^i \frac{\partial T_i^0}{\partial q_j} \dot{q}_j \right) r_i^i = \sum_{j=1}^i U_{ij} \dot{q}_j r_i^i$$

Par définition, on a :

$$U_{ij} = \frac{\partial T_i^0}{\partial q_j} = \begin{cases} T_{j-1}^0 Q_j T_i^{j-1} & \text{pour } j \leq i. \\ 0 & \text{pour } j > i. \end{cases}$$

Avec :

$$Q_j = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad i = 1, 2, \dots, n.$$

II.5 Génération de trajectoires

La génération des trajectoires dans l'espace cartésien, permet au robot d'aller d'une position actuelle P_a vers une position finale P_f appartenant à l'espace de travail. C'est connu que l'espace de travail est bien limité. Cette limitation est due principalement aux contraintes mécaniques du robot, telles que les marges de variation de chaque articulation et les longueurs des segments [16].

II.5.1 Méthode des polynômes cubique

Considérez le problème de déplacer l'outil de sa position initiale à une position finale et ce durant un certain temps t_f . La cinématique 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, 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 valeur initiale et la valeur finale [5].

$$\begin{aligned} \theta(0) &= \theta_0 \\ \theta(t_f) &= \theta_f \end{aligned} \quad (\text{II.19})$$

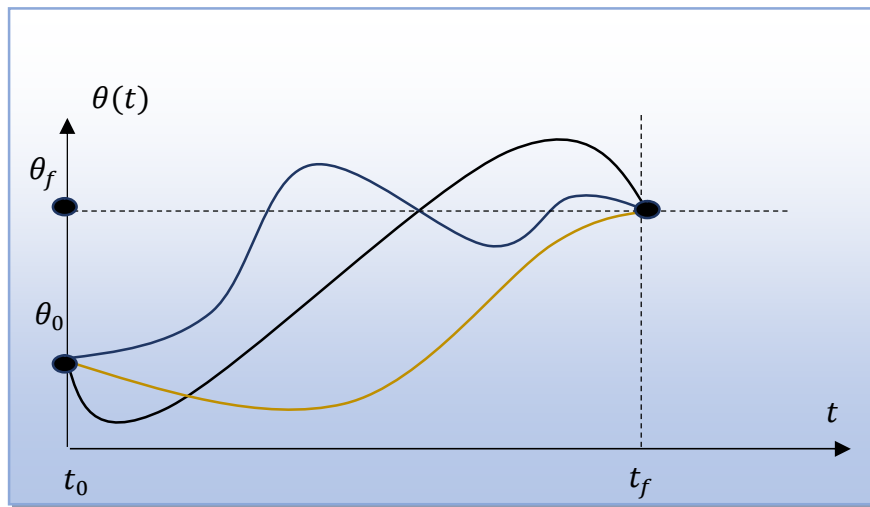


Figure II.7: Plusieurs formes de chemin possible pour une seule articulation.

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

$$\theta(0) = 0, \theta(t_f) = 0 \tag{II.20}$$

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

$$\theta(t) = a_0 + a_1 t^1 + a_2 t^2 + a_3 t^3 \tag{II.21}$$

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

$$\begin{aligned} \dot{\theta}(t) &= a_1 + 2a_2 t^1 + 3a_3 t^2 \\ \ddot{\theta}(t) &= 2a_2 + 6a_3 t \end{aligned} \tag{II.22}$$

La combinaison (II.22) et (II.21) avec les contraintes de quatre désirs donne quatre équations en quatre inconnues

$$\begin{aligned} \theta(0) &= 0, 0 = a_1, \\ \theta_f &= a_0 + a_1 t_f^1 + a_2 t_f^2 + a_3 t_f^3 \\ 0 &= a_1 + 2a_2 t_f^1 + 3a_3 t_f^2 \end{aligned} \tag{II.23}$$

Résoudre les équations de a_i , pour l'obtention

$$\begin{aligned} a_0 &= \theta(0), \quad a_1 = 0 \\ a_2 &= \frac{3}{t_f^2}(\theta_f - \theta_0), \quad a_3 = -\frac{3}{t_f^3}(\theta_f - \theta_0) \end{aligned} \tag{II.24}$$

En utilisant (II.24), on peut calculer le polynôme cubique qui relie toute position d'angle d'articulation initiale avec une position finale souhaitée. Cette solution est pour les cas où l'articulation commence et se termine à une vitesse nulle.

II.6 Conclusion

Dans ce chapitre nous avons étudié les concepts de base de la robotique et la modélisation du robot à "n" degrés de liberté, en basant sur les paramètres de D-H. Cette formulation nous a permis d'exploiter le formalisme d'Euler- Lagrange, afin d'établir le modèle géométrique, cinématique et dynamique que nous utilisons pour le calcul des commandes. Nous avons développé le calcul du modèle dynamique pour un robot à deux degrés liberté.

III.1 Introduction

La commande des robots manipulateurs a pour but de contrôler le déplacement des actionneurs suivant une trajectoire programmée, donc le problème de la commande des robots manipulateurs peut être formulé comme la détermination de l'évolution des forces généralisées (forces ou couples) que les actionneurs doivent exercer pour garantir l'exécution de la tâche tout en satisfaisant certains critères de performance. Durant ces trois dernières décennies, en vue d'améliorer les performances des manipulateurs, des recherches avancées ont permis de faire émerger de nouvelles techniques de commande non linéaire pour les applications aux robots manipulateurs. Le but de cette partie est d'introduire la commande classique et la commande floue d'un bras manipulateur à 2 d.d.l.

III.2 Technique de commande des robots

Dans le cas où le modèle exact du robot est parfaitement connu, plusieurs stratégies de commande peuvent être appliquées. Cependant, en pratique, cette condition idéale n'est jamais tout à fait remplie, vue les différentes perturbations agissant sur le robot manipulateur et les incertitudes du modèle, d'où la nécessité d'adapter la commande.

Différentes techniques sont utilisées pour la commande des bras manipulateurs. La conception mécanique du bras manipulateur a une influence sur le choix de schéma de commande. Un robot manipulateur est une structure mécanique complexe dont les inerties par rapport aux axes des articulations varient non seulement en fonction de la charge mais aussi en fonction de la configuration, des vitesses et des accélérations [17].

Dans ce travail, on s'intéresse à la PID classique, la commande dynamique et la commande floue,

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 ($m_1=4.7Kg$ et $m_2=2.3Kg$) et deux actionneurs [18].

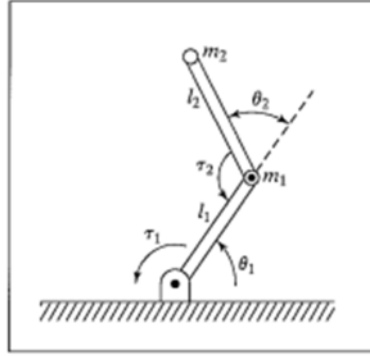


Figure III.1: Structure du bras de robot étudié.

Modèle dynamique du bras est donné par :

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} m_2 l_2^2 + 2m_2 l_1 l_2 C_2 + (m_1 + m_2) l_1^2 & m_2 l_2^2 + m_2 l_1 l_2 C_2 \\ m_2 l_1 l_2 C_2 + m_2 l_2^2 & m_2 l_2^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} m_2 l_1 l_2 S_2 \dot{\theta}_2^2 - 2m_2 l_1 l_2 S_2 \dot{\theta}_1 \dot{\theta}_2 \\ 2m_2 l_1 l_2 S_2 \dot{\theta}_1 \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} m_2 l_2 C_{12} g + (m_1 + m_2) l_1 C_1 g \\ m_2 l_2 C_{12} g \end{bmatrix} \quad (\text{III.1})$$

Avec $C_2 = \text{Cos}(\theta_2)$, $C_{12} = \text{Cos}(\theta_1 + \theta_2)$ et $S_2 = \text{Sin}(\theta_2)$.

On remplace m_2, l_1 et l_2 par leurs valeurs, on aboutit à :

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} 2.3 + 1.5C_2 & 0.575 + 0.575C_2 \\ 0.575C_2 & 0.575 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 0.575S_2 \dot{\theta}_2^2 - 1.15S_2 \dot{\theta}_1 \dot{\theta}_2 \\ 1.15S_2 \dot{\theta}_1 \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 11.5C_{12} + 34.5C_1 \\ 11.5C_{12} \end{bmatrix} \quad (\text{III.2})$$

L'équation (III.2) correspond au modèle dynamique inverse du bras, du fait qu'elle donne le vecteur de commande en fonction de l'accélération. Le modèle direct est donné par :

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} 2.3 + 1.15C_2 & 0.575 + 0.575C_2 \\ 0.575C_2 + 0.575 & 0.575 \end{bmatrix}^{-1} * \left(\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} - \begin{bmatrix} 0.575S_2 \dot{\theta}_2^2 - 1.15S_2 \dot{\theta}_1 \dot{\theta}_2 \\ 1.15S_2 \dot{\theta}_1 \dot{\theta}_2 \end{bmatrix} - \begin{bmatrix} 11.5C_{12} + 34.5C_1 \\ 11.5C_{12} \end{bmatrix} \right) \quad (\text{II.3})$$

III.4 Linéarisation du système par compensation des non linéarités

La méthode par linéarisation est une méthode qui n'est valable que localement autour d'un point de fonctionnement et, par conséquent, cette méthode ne peut pas être utilisée pour

définir un comportement global. De plus, lors de la linéarisation les effets non linéaires sont alors considérés comme perturbateurs et, de ce fait, négligés. Or, la dynamique apportée par ces effets non linéaires est plus riche que les systèmes linéaires.

Dans ce qui suit, on s'intéressera à la linéarisation par compensation des termes non linéaires. Le modèle linéaire reste ainsi valable quelle que soit la configuration du robot, à condition que l'estimation et la compensation des termes non linéaires est bien faite.

Dans le cas d'un modèle non linéaire:

$$a\ddot{x} + h(\dot{x}) + g(x) = f \quad (\text{III.4})$$

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

$$f = \alpha f' + \beta \quad (\text{III.5})$$

On choisit les paramètres comme suit :

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

On aboutit à : $f' = \ddot{x}$

On choisit :

$$f' = -k_p x - k_v \dot{x} \quad (\text{III.6})$$

On remplace (III.5) dans (III.4)

$$a\ddot{x} + b\dot{x} + cx = \alpha(-k_p x - k_v \dot{x}) + \beta$$

$$\ddot{x} + b\dot{x} + cx = \alpha(-k_p x - k_v \dot{x}) + b\dot{x} + cx$$

$$\ddot{x} = (-k_p x - k_v \dot{x}) \quad (\text{III.7})$$

$$\ddot{x} + k_p x + k_v \dot{x} = 0$$

On obtient enfin l'équation dynamique d'un système linéaire. Il suffit de bien choisir les paramètres k_p et k_v pour imposer les pôles et le comportement dynamique désirés.

III.5 Commande classique

La synthèse d'un asservissement doit toujours répondre à certaines exigences. Ces dernières sont appelées cahier des charges. Un cahier des charges d'une boucle de régulation, impose en boucle fermée: Stabilité, Précision, Rapidité, Robustesse, Dépassement, Rejet de perturbations, ...

Les systèmes peuvent présenter des défauts, par exemples: une précision insuffisante, une stabilité faible (voir une instabilité), un temps de réaction trop lent, un dépassement trop important, etc. Il est souvent nécessaire d'intégrer dans le système asservis un correcteur. L'objectif est alors d'améliorer un ou plusieurs de ces différents paramètres, sans bien évidemment le faire au détriment des autres.

On considère un système asservi défini par le schéma bloc suivant

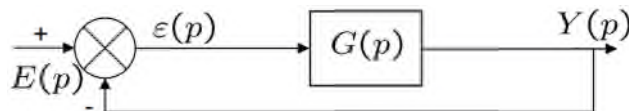


Figure III.2 : schéma bloc d'un système asservi.

L'idée de base consiste à introduire dans la chaîne directe, en amont du système un dispositif supplémentaire de fonction de transfert $C(p)$. Ce dispositif est appelé correcteur, contrôleur, compensateur ou encore régulateur. Le rôle du correcteur consiste à modifier les performances du système initial (précision, stabilité, rapidité, ...).

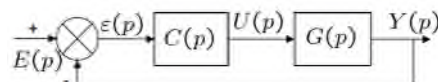


Figure III.3 : schéma bloc d'un système asservi avec correcteur.

L'objectif de la correction (le contrôle ou la commande) consiste à choisir la bonne fonction de transfert $C(p)$ du correcteur de manière à régler chaque performance sur sa valeur requise, sans perturber le fonctionnement du système.

En automatique, il existe trois actions correctives élémentaires qui permettent, individuellement, de corriger telle ou telle performance: l'action Proportionnelle (P), l'action Intégrale (I), l'action Dérivée (D). Ces actions sont relativement simples à réaliser mais,

généralement, elles dégradent d'autres performances. Quand le cahier des charges est peu exigeant, ces actions suffisent largement à satisfaire ce dernier. Dans le cas contraire, il faut envisager de combiner ces différentes actions au sein d'un correcteur plus complexe de type **PI**, **PD**, ou **PID**.

III.5.1. Régulation linéaire

III.5.1.1 Régulation Proportionnel Dérivé(PD)

Un régulateur Proportionnel Dérivé (PD) est un organe de contrôle qui permet d'effectuer une régulation en boucle fermée d'un procédé industriel. Le régulateur PD compare une valeur mesurée par le procédé (position de l'extrémité du bras flexible) avec une valeur de consigne, c.à.d. l'angle de référence. Le signal erreur, étant la différence entre les deux valeurs, est utilisé pour calculer une nouvelle valeur d'entrée qui tend à réduire au maximum l'écart entre la mesure et la consigne. Le régulateur proportionnel dérivée est régi par :

- **Régulateur proportionnel dérivée sans compensation de gravité :**

$$\tau = K_p \tilde{q}(t) - K_v \dot{\tilde{q}}(t) \quad (\text{III.8})$$

Avec : K_p et $K_v \in \mathbb{R}^{n \times n}$ sont des matrices de gains de retour symétriques définies positives et τ est l'entrée du système.

$\tilde{q}(t) = q_d - q$: est l'erreur de position.

$\dot{\tilde{q}}(t) = \dot{q}_d - \dot{q}$: est le vecteur des erreurs de vitesse.

$q_d \in \mathbb{R}^{n \times n}$: la position articulaire désirée.

$q \in \mathbb{R}^{n \times n}$: la position articulaire mesurée.

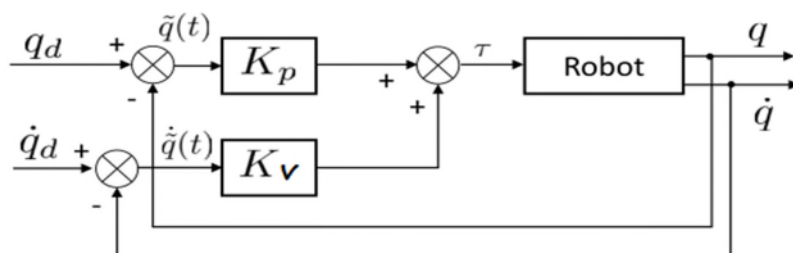


Figure III.4 : Schéma-bloc du régulateur PD

Le robot, étant modélisé par un modèle de second ordre linéaire, est équivalent à l'équation :

$$a\ddot{x} + b\dot{x} + cx = f \quad (\text{III.9})$$

Dans la commande des bras de robot, 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 apériodique. On choisit une loi de commande proportionnelle dérivée :

$$f = -k_p x - k_v \dot{x} \quad (\text{III.10})$$

On remplace (III.10) dans (III.9) :

$$\begin{aligned} a\ddot{x} + b\dot{x} + cx &= -k_p x - k_v \dot{x} \\ a\ddot{x} + (b+k_v)\dot{x} + (c+k_p)x &= 0 \\ a\ddot{x} + k'_v \dot{x} + k'_p x &= 0 \end{aligned} \quad (\text{III.11})$$

Cette équation peut être mise sous la forme standard :

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

Pour $\zeta = 1$, il reste à fixer la valeur de la pulsation naturelle, w_n , et de calculer k_v et k_p .

Application de la commande PD à un bras de deux degrés de liberté

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

$$\tau = M\ddot{\theta} \quad (\text{III.12})$$

Avec : M une matrice diagonale constante, τ et $\ddot{\theta}$ des vecteurs colonnes.

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

Structure de la commande

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

$$E = [\theta_d - \theta(t)] \quad (\text{III.13})$$

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

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, les trajectoires réelles du robot sont calculées en utilisant son modèle dynamique direct.

Le modèle de commande est défini par (III.12). En utilisant une loi de commande PD :

$$\tau = k_p e + k_v \dot{e} \quad (\text{III.14})$$

Avec τ , e , et \dot{e} des vecteurs colonnes et k_p, k_v des vecteurs lignes et

$$e = [\theta_d - \theta(t)].$$

Ce qui revient à :

$$\begin{aligned} \tau_1 &= k_{v1} \dot{e}_1 + k_{p1} e_1 \\ \tau_2 &= k_{v2} \dot{e}_2 + k_{p2} e_2 \end{aligned} \quad (\text{III.15})$$

Pour une $\zeta=1$ et on choisissant une pulsation $\omega_n = 25$, on calcule $k_v = 50$ et $k_p=625$.

$$\begin{aligned} \tau_1 &= 50(\dot{\theta}_{d1} - \dot{\theta}_1) + 625(\theta_{d1} - \theta_1) \\ \tau_2 &= 50(\dot{\theta}_{d2} - \dot{\theta}_2) + 625(\theta_{d2} - \theta_2) \end{aligned} \quad (\text{III.16})$$

Simulation et résultats

LE programme MATLAB (*fichier.m*) est donné en (annexe A .1).Les résultats de simulation sont illustrés par la figure III.5.

1. Pour une pulsation naturelle $w_n = 25$ rad/sec et $\zeta = 1$, on calcule :

$$k_p = 625 \quad \text{et} \quad k_v = 50$$

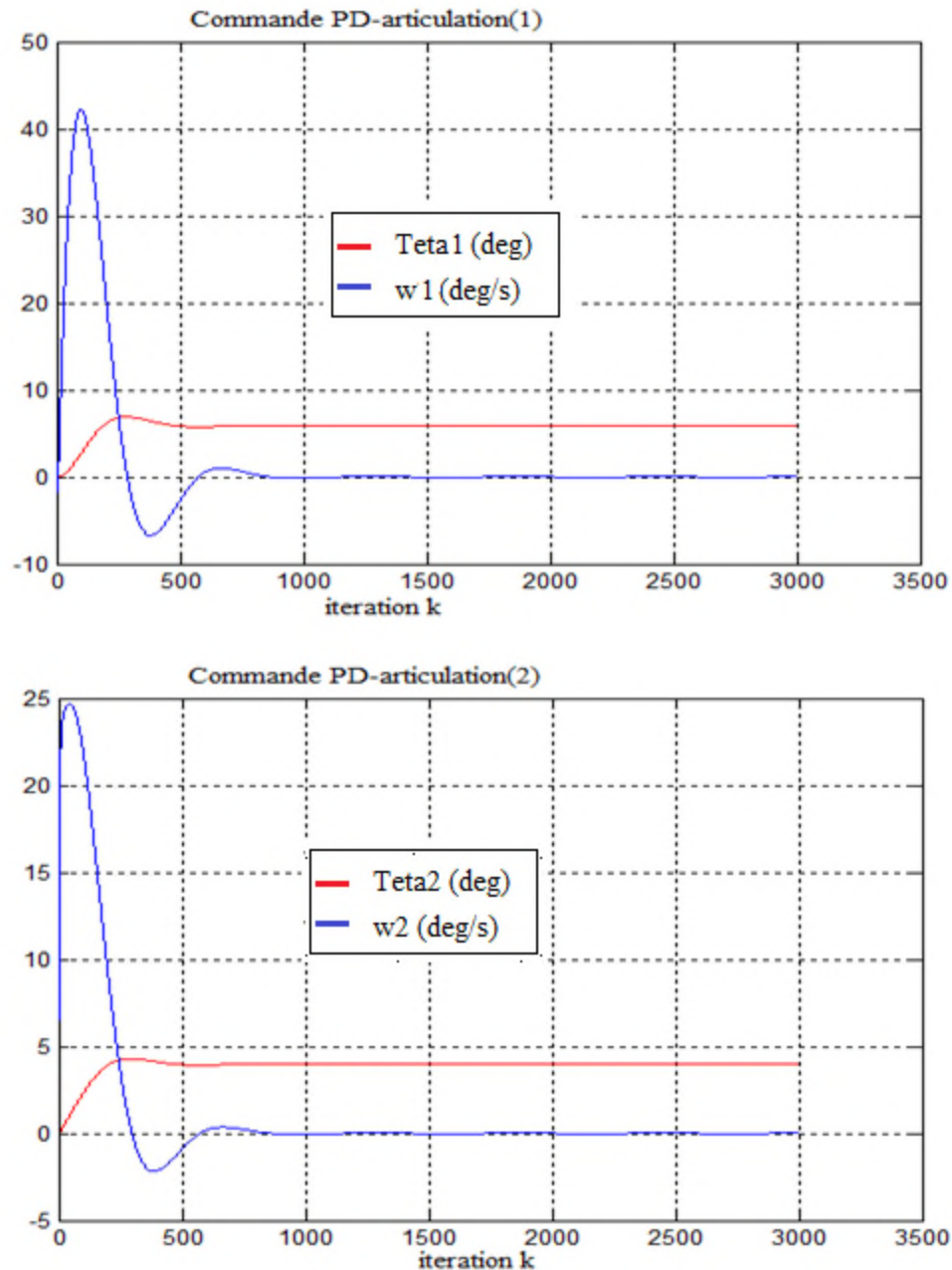


Figure III.5: Formes des variables articulaires et les vitesses angulaires.

D'après les résultats de la simulation, on constate que l'évolution de la 1^{ère} articulation $\theta_1(t)$ converge vers une valeur finale qui est un peu loin de celle désirée $\theta_{d1}(t) = 10^\circ$. Par contre, l'évolution de la 2^{ème} articulation $\theta_2(t)$ converge vers une valeur finale qui est proche de celle désirée $\theta_{d2}(t) = 5^\circ$.

Dans ce cas, l'ajustement du contrôleur est trivial, étant donné qu'il suffit de choisir les matrices k_p, k_v de gains de retour symétriques définies positives. Pour cela, on ajustant sur les paramètres k_p et k_v , on obtient le résultat suivant.

2. Pour une autre valeur de $w_n = 30$ rad/sec, on calcule :

$$k_p = 900 \text{ et } k_v = 60$$

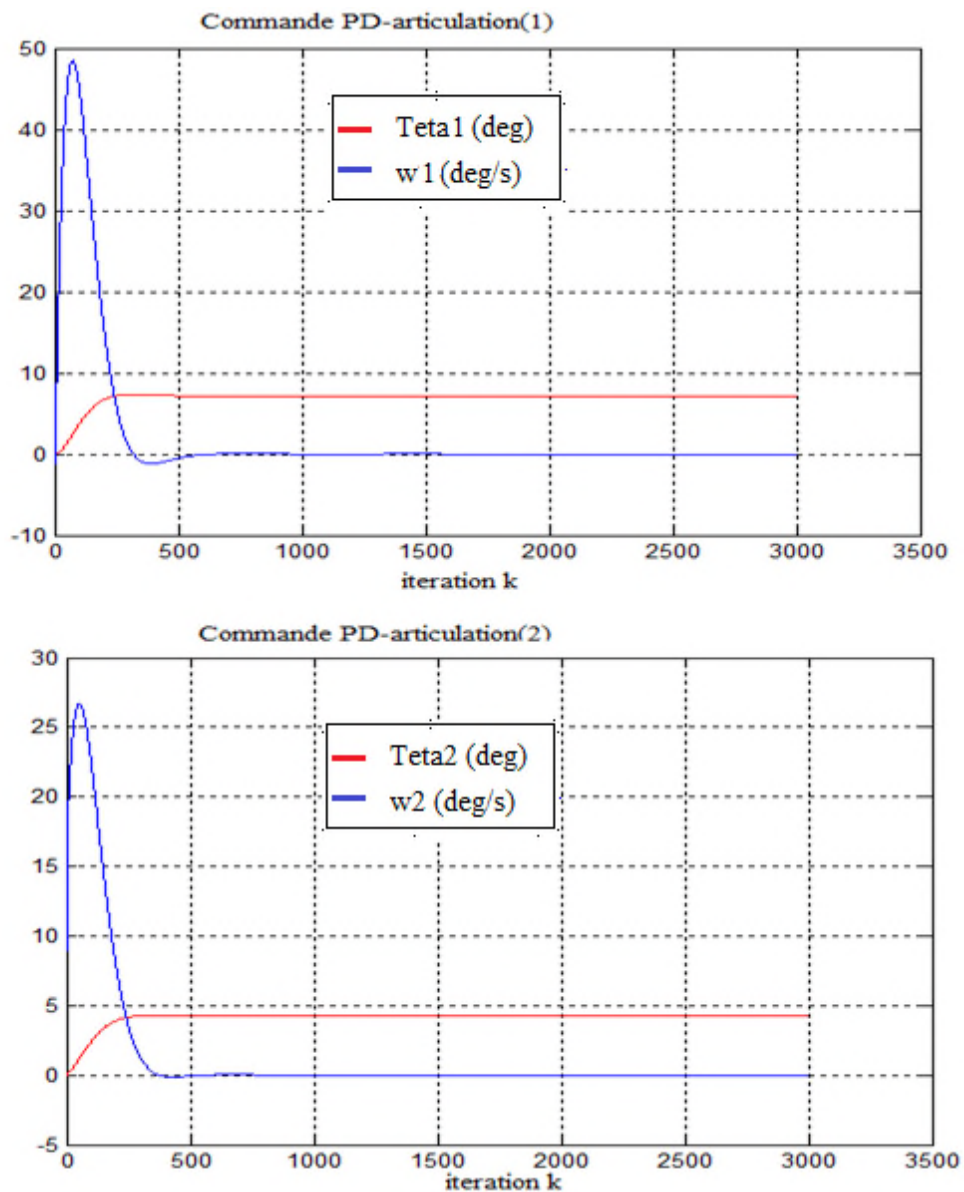


Figure III.6: Formes des variables articulaires et les vitesses angulaires.

On remarque que le dépassement (D%) et le temps de repense t_r ont diminuées, mais les erreurs statiques restent significatives. Néanmoins, la commande PD ne peut pas garantir un bon suivi en position pour les manipulateurs dont le modèle dynamique contient un terme de gravité non négligeable. Pour ce faire, on fait appel à d'autres méthodes, telles que la commande PD avec compensation de gravité qui est capable d'atteindre l'objectif de commande en position globalement pour des robots à n d.d.l.

➤ **Régulateur proportionnel-dérivée avec compensation de gravité**

De l'application d'une simple loi de commande de type PD, nous pouvons remarquer que la dynamique du système n'est pas utilisée. Donc, les non linéarités du système ne sont pas compensés. Alors, la compensation des non linéarités s'avère plus adéquat. Mais, dans notre travail, comme recommandé aussi dans la littérature, on s'est intéressé à la compensation du terme le plus influant, à savoir la gravitation.

Pour la commande PD avec compensation de la gravité, le couple τ appliqué aux actionneurs est donné par [5] :

$$\tau = K_p \tilde{q}(t) - K_v \dot{\tilde{q}}(t) + G(q) \quad (\text{III.17})$$

Avec $G(q)$ est la force gravitationnelle.

$$G(q) = \begin{bmatrix} m_2 l_2 g + (m_1 + m_2) l_1 g \\ m_2 l_2 g \end{bmatrix} \quad (\text{III.18})$$

En fait, au lieu d'ajuster en temps réel le terme $G(q)$, l'utilisation d'une estimation $G(0)$ est suffisante. Après calcul, on obtient la forme :

$$\tau = k_p e + k_v \dot{e} + G(0) \quad (\text{III.19})$$

$$\tau_1 = 50(\dot{\theta}_{d1} - \dot{\theta}_1) + 625(\theta_{d1} - \theta_1) + 45.08$$

$$\tau_2 = 50(\dot{\theta}_{d2} - \dot{\theta}_2) + 625(\theta_{d2} - \theta_2) + 11.27$$

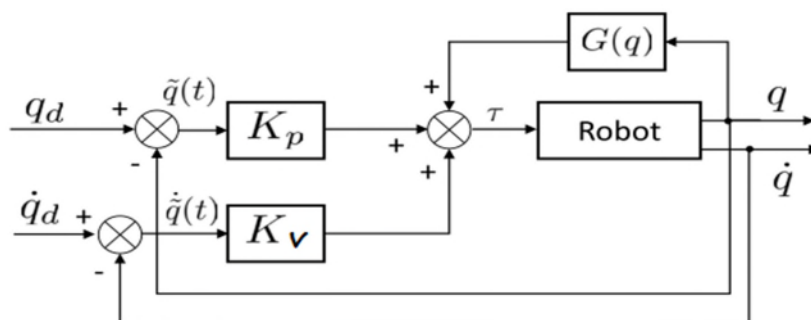


Figure III.7: Schéma-bloc du régulateur PD avec compensation de gravité.

Simulation et résultats

Les résultats de simulation sont illustrés par la figure III.8.

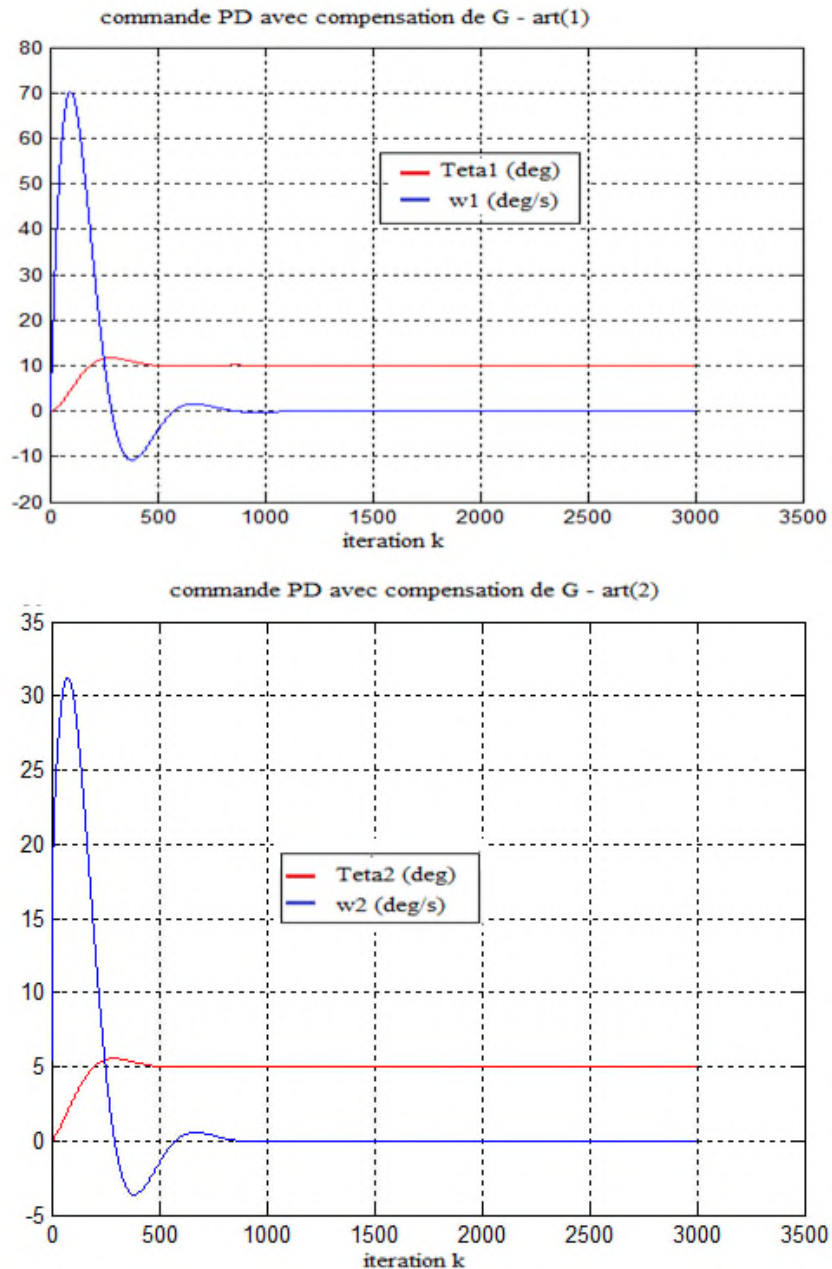


Figure III.8: Formes des variables articulaires et les vitesses angulaires.

On constate, cette fois-ci, que la commande PD avec compensation de gravité est capable d'atteindre l'objectif de commande en position globalement pour des robots à n ddl. De plus son réglage est trivial.

III.5.1.2 Régulation Proportionnel Intégral Dérivé(PID)

Le correcteur PID (**P**roportionnel-**I**ntégral-**D**érivé) est une combinaison d'un régulateur P, un régulateur I et un régulateur D. L'intérêt de ce correcteur est d'intégrer les effets positifs des trois correcteurs élémentaires. Il a la capacité d'éliminer la compensation de

l'état d'équilibre grâce à la l'action intégrale, comme il peut anticiper le futur grâce à une action dérivée [19].

Pour élaborer la commande PID classique, il faut considérer chaque articulation comme un mécanisme indépendant dont le modèle peut être linéarisé dans une zone de fonctionnement. La commande prend le caractère local. Avec sa simplicité et sa facilité d'implémentation, elle tient le monopole dans le domaine industriel [20].

La loi de commande $\tau(t)$ s'exprime en fonction de l'erreur comme suite :

$$\tau(t) = K_p \tilde{q}(t) - K_v \dot{\tilde{q}}(t) + k_I \int_0^t \tilde{q}(t) \quad (\text{III.20})$$

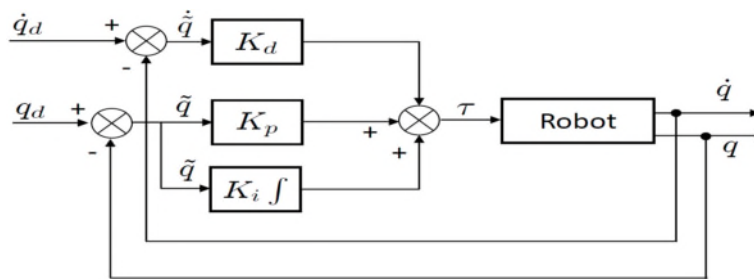


Figure III.9: Schéma-bloc du régulateur PID

En supposant que les articulations sont découplées et linéaires et en négligeant les forces centrifuges et Coriolis ainsi que les forces de gravitation et des frottements, chaque articulation peut être décrite par le modèle simplifié :

$$\tau = M\ddot{\theta}$$

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.

Application de la commande PID à bras de deux degrés de liberté :

➤ **régulateur proportionnel intégral dérivée sans compensation de gravité :**

La loi de commande PID est de la forme :

$$\tau_i = k_{vi} \dot{e}_i + k_{pi} e_i + k_{Ii} \int_0^t e_i(t) dt = M_i \ddot{\theta} \quad (\text{III.21})$$

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

$$M_i P^3 + k_{vi} P^2 + k_{pi} P + k_{li} = 0 \quad (\text{III.22})$$

On prend $M_i=1$ (les versions les plus simple de commande n'utilise aucun paramètre de système)

$$P^3 + k_{vi} P^2 + k_{pi} P + k_{li} = 0 \quad (\text{III.23})$$

L'équation (III.21) est une équation d'ordre 3. Pour éviter les réponses oscillatoires, on prend le facteur d'amortissement $\zeta = 1$:

$$(P-P_3)(P^2 + 2\zeta w_n P + w_n^2) \quad (\text{III.24})$$

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

$$P_1 = P_2 = -5$$

On choisit le troisième pôle (plus rapide) 5 fois plus grande en module : $P_3 = 5P_2$. Ce qui donne :

$$(P+25)(P+10P+25)=0$$

$$P^3 + 35P^2 + 275P + 625 = 0 \quad (\text{III.25})$$

Par comparaison avec (III.21) et identification : $k_v=35$, $k_p=275$, $k_l=625$

$$\tau_1 = 35\dot{e}_1 + 275e_1 + 625 \int_0^t e_1(t) dt$$

$$\tau_2 = 35\dot{e}_2 + 275e_2 + 625 \int_0^t e_2(t) dt \quad (\text{III.26})$$

Le modèle du 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 :

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \text{inv} \begin{bmatrix} 2.3 + 1.5C_2 & 0.575 + 0.575C_2 \\ 0.575C_2 & 0.575 \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} - \begin{bmatrix} 0.575S_2\dot{\theta}_2^2 - 1.15S_2\dot{\theta}_1\dot{\theta}_2 \\ 1.15S_2\dot{\theta}_1\dot{\theta}_2 \end{bmatrix} - \begin{bmatrix} 11.5C_{12} + 34.5C_1 \\ 11.5C_{12} \end{bmatrix} - \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \quad (\text{III.27})$$

Simulation et résultats

Dans cette simulation, on a un système avec des perturbations F (III.27) qui se génère avec la fonction `rand` de MATLAB, et la loi de commande (III.26), (voir l'annexe A.2).

Les résultats de simulation sont illustrés par la figure III.10.

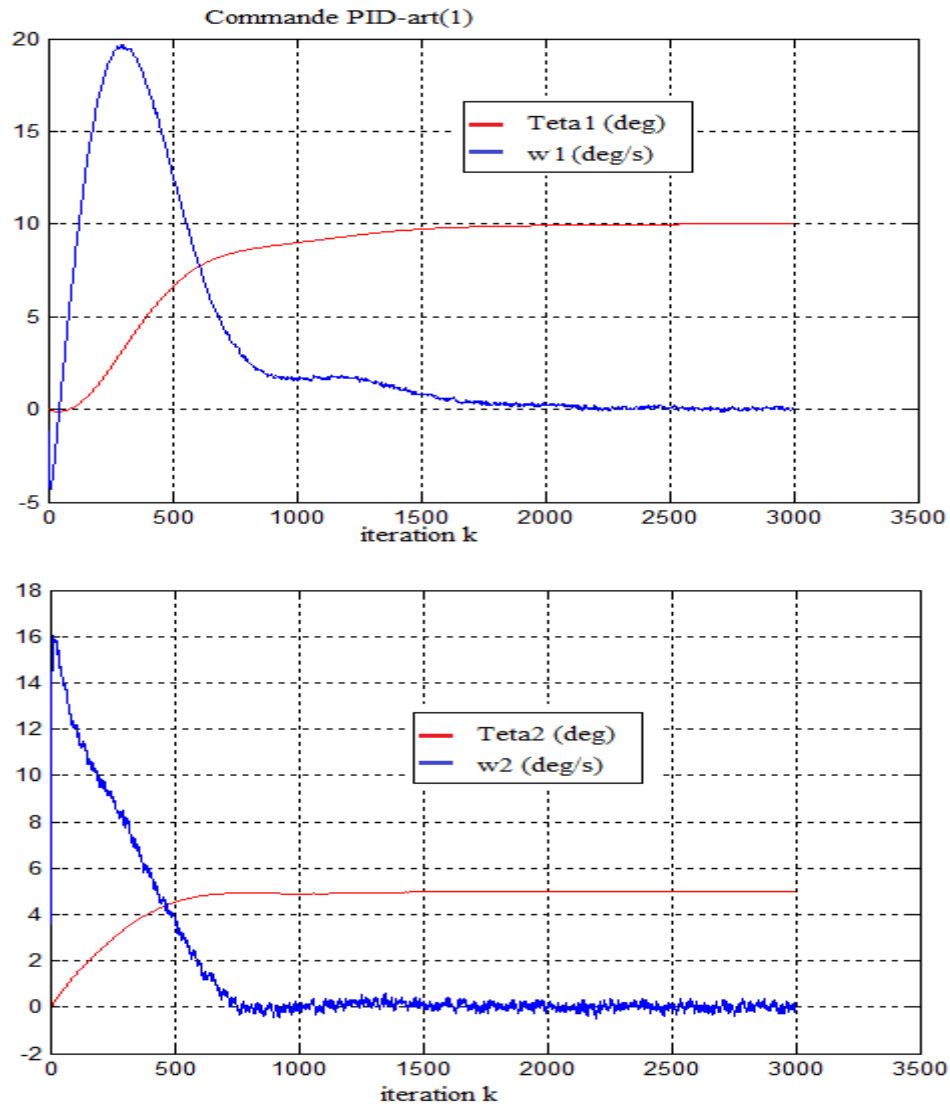


Figure III.10: *Formes des variables articulaires et les vitesses angulaires.*

Avec la comparaison à la loi de commande PD précédente, on constate que le terme intégral a assuré l'annulation de l'erreur statique ($\theta_f = \theta_{fd}$), ainsi que l'effet des perturbations n'affecte que les profils des vitesses.

➤ **Régulateur proportionnel intégral dérivée avec compensation de gravité :**

$$\tau = K_p \tilde{q}(t) - K_v \dot{\tilde{q}}(t) + k_I \int_0^t \tilde{q}(t) dt + G(0)$$

Après le calcul, on obtient cette forme :

$$\tau_1 = 35\dot{e}_1 + 275e_1 + 625 \int_0^t e_1(t) dt + 45.08$$

$$\tau_2 = 35\dot{e}_2 + 275e_2 + 625 \int_0^t e_2(t) dt + 11.27$$

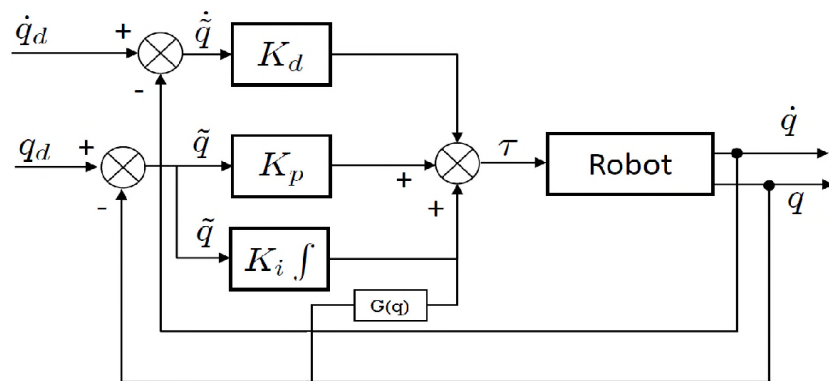


Figure III.11: Schéma-bloc du régulateur PID avec compensation de gravité.

Simulation et résultats

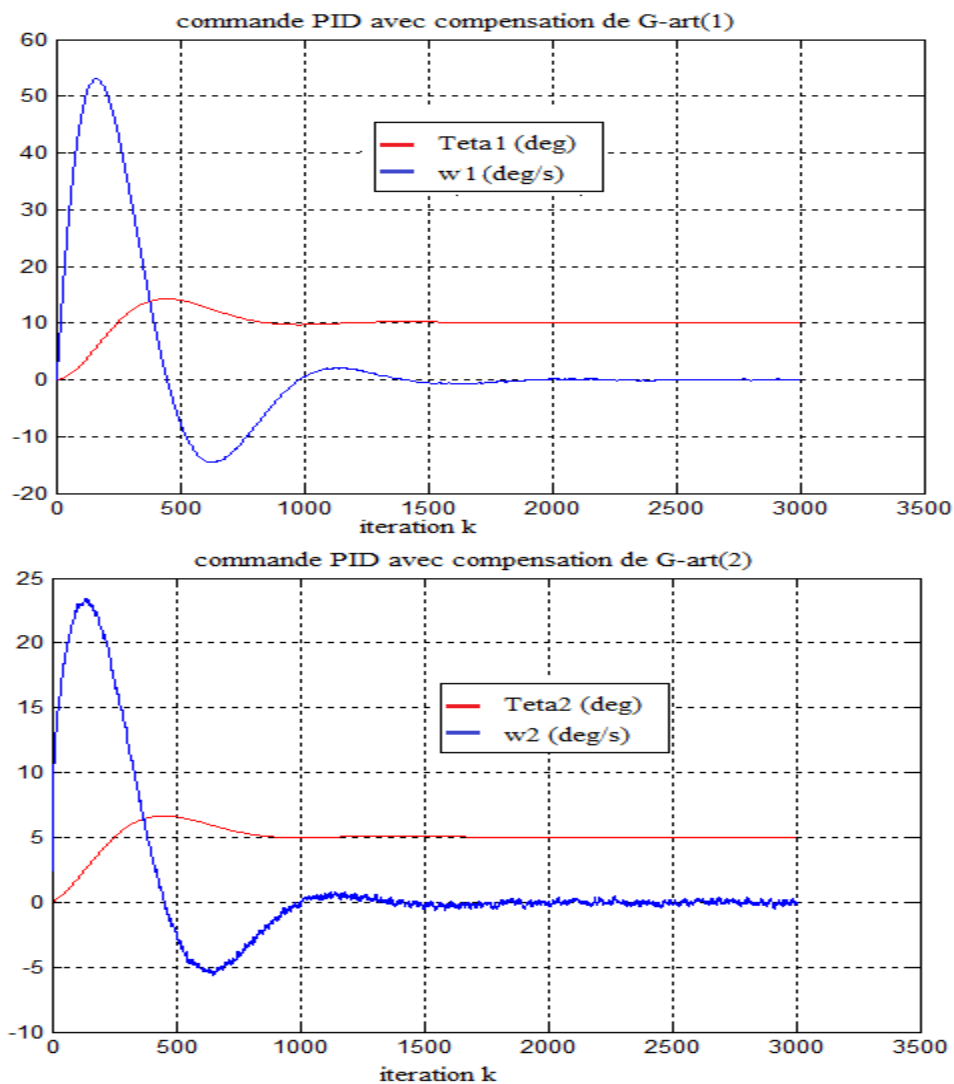


Figure III.12: Formes des variables articulaires et les vitesses angulaires.

Dans cette simulation, on constate que l'effet des perturbations sur le profil des vitesses a diminué dans la deuxième articulation et presque nulle dans la première et un temps d'établissement très court, mais le pic de dépassement important par rapport à la loi de commande PID sans compensation de gravité.

III.6 Commande floue en régulation

III.6.1 Rappels

La logique floue est une extension de la logique booléenne créée par Lotfi Zadeh en 1965 en se basant sur sa théorie mathématique des ensembles flous, qui est une généralisation de la théorie des ensembles classiques. En introduisant la notion de degré dans la vérification d'une condition, nous permettons à une condition d'être dans un autre état que vrai ou faux. La logique floue confère ainsi une flexibilité très appréciable aux raisonnements qui l'utilisent, ce qui rend possible la prise en compte des imprécisions et des incertitudes.

Un des intérêts de la logique floue pour formaliser le raisonnement humain est que les règles sont énoncées en langage naturel [21].

La structure de base d'un contrôleur flou est illustrée par figure III.16. Pour la simplicité, on considère le cas général d'un système flou ayant 02 entrées et une sortie. La première étape dans la conception est le choix des variables linguistique d'entrée/sortie, ainsi que leurs valeurs linguistique A_i et B_i . La construction et structure de la base de règles floues dépend de l'ensemble de ces valeurs. Les valeurs souvent utilisés sont : PG (positif grand), PM (positif moyen), PP (positif petit), ZR (zéro), NP (négatif petit), PM (négatif moyen), PG (négatif grand).

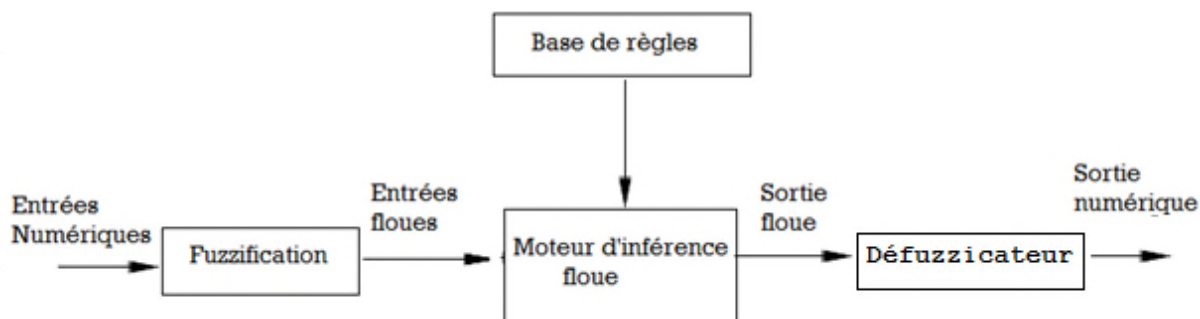


Figure III.13: Structure d'un système flou.

1. La base de règle

On appelle base de règles, l'ensemble des règles reliant les variables floues d'entrée aux variables floues de sortie à l'aide de différents opérateurs. Ces règles se présentent sous la forme : **Si** condition1 **et** condition2 (**et...**) **alors** action sur les sorties.

Elles forment le noyau des systèmes d'inférence utilisés en commande et retracent le raisonnement que peut faire un être humain sur une connaissance imparfaitement définie [22].

2. Inférence floue

Elle permet de calculer l'ensemble flou associé à la commande et se fait par les opérations d'inférence floue et l'agrégation des règles. L'inférence floue repose sur l'utilisation d'un opérateur d'implication floue pour chaque règle à analyser. Cet opérateur quantifie la force de liaison entre la prémisse et la conclusion de la règle [19].

3. Le fuzzificateur

Transforme les valeurs numériques d'entrée à des ensembles flous. Le fuzzificateur de type singleton est souvent utilisé [23].

4. Défuzzificateur

Permet d'extraire de l'ensemble flou de sortie, la valeur numérique qui peut être considérée comme l'élément le plus représentatif de cet ensemble. Plusieurs méthodes de défuzzification peuvent être utilisées. Les plus connues sont [23]:

- Méthode du maximum : la sortie correspond à l'élément ayant le degré d'appartenance le plus élevé. En d'autres termes, la sortie numérique correspond au centre de la conséquence de la règle ayant la valeur de vérité la plus élevée :

$$y_n = \bar{y}_i \quad (\text{III.28})$$

Où \bar{y}_i est le centre de la conséquence de la règle R_i ayant la valeur de vérité : $w_i = \text{Max}(w_1, \dots, w_m)$

- Méthode du centre de gravité(COG) : produit, comme sortie, le centre de gravité de l'ensemble flou de sortie

$$y^{COG} = \frac{\int_s y \cdot \mu_B(y) dy}{\int_s \mu_B(y) dy} \quad (\text{III.29})$$

Où s est le support de la fonction $\mu_B(y)$ caractérisant l'ensemble flou de sortie.

Une version simplifiée et très populaire de cette méthode est donnée par

$$y^{COG} = \frac{\sum_{i=1}^m \bar{y}_i \cdot \mu_B(\bar{y}_i)}{\sum_{i=1}^m \mu_B(\bar{y}_i)} = \frac{\sum_{i=1}^m \bar{y}_i \cdot w_i}{\sum_{i=1}^m w_i} \quad (\text{III.30})$$

Où \bar{y}_i : centre de gravité du label de sortie B_i proposé par la règle R_i , $\mu_B(\bar{y}_i) = w_i$: valeur de vérité la règle R_i .

m : nombre de règles dans la base.

III.6.2 La commande par la logique floue à un bras de deux degrés de liberté

Dans cette partie, nous établissons une commande floue qui permettra, à l'aide du modèle dynamique, de générer le couple nécessaire pour chaque articulation du bras pour passer d'une position à une autre (de θ vers θ_d).

Structure de contrôleur flou

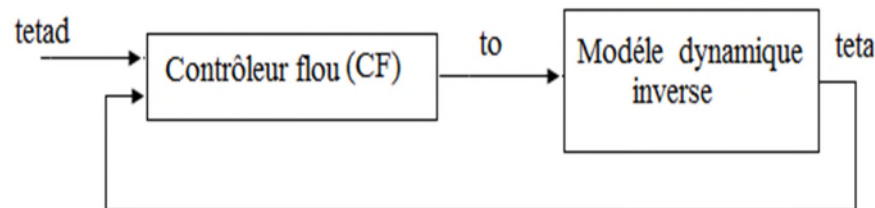


Figure III.14 Structure générale de la commande floue.

La première étape dans la conception du contrôleur flou est la partition floue des espaces d'entrée/sortie. Ceci est illustré par figure III.15, en associant 07 valeurs linguistiques pour chaque variable et en utilisant des fonctions d'appartenance triangulaires de plus en plus étroites en se rapprochant de l'état d'équilibre.

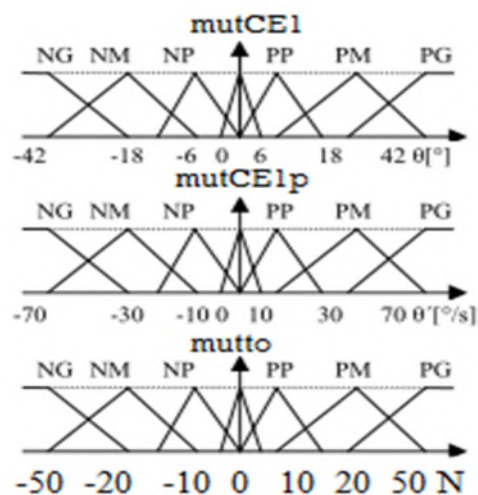


Figure III.15: Formes des fonctions d'appartenance.

Les 49 règles floues résultantes sont arrangées sous forme d'une table de décision. Chaque case correspond à une règle. Par exemple, on peut lire :

R_8 : **SI** E est NG **ET** \dot{E} est NM **Alors** τ est NG .

Avec : $E_i = \theta_{ai}(t) - \theta_i(t)$ et $\dot{E}_i = -\dot{\theta}_i(t) = -\omega_i$

La sortie numérique du contrôleur est calculée en utilisant la méthode du centre de gravité simplifiée. On note que, pour cet exemple, on peut utiliser moins de 07 valeurs linguistiques et surtout pour les variables d'entrée. Ce qui réduit le nombre de règles à concevoir.

Tableau III.1: Table de décision.

	E							
	τ	NG	NM	NP	ZR	PP	PM	PG
\dot{E}	NG	NG	NG	NG	NG	NP	NP	ZR
	NM	NG	NG	NG	NM	ZR	PP	PP
	NP	NG	NG	NG	NP	PP	PP	ZR
	ZR	NG	NM	NP	ZR	PP	PM	PG
	PP	NM	NP	NP	PP	PM	PG	PG
	PM	NP	NP	ZR	PM	PG	PG	PG
	PG	ZR	PP	PP	PG	PG	PG	PG

Du fait que les méthodes de défuzzification n'utilisent que les centres de gravité des conséquences, la table III.1 est réduite à la table III.2

Tableau III.2: Table de décision réduite.

	E							
	τ	NG	NM	NP	ZR	PP	PM	PG
NG		-50	-50	-50	-50	-10	-10	0
NM		-50	-50	-50	-20	0	10	10
NP		-50	-50	-50	-10	10	10	0
ZR		-50	-20	-10	0	10	20	50
PP		-20	-10	-10	10	20	50	50
PM		-10	-10	0	20	50	50	50
PG		0	10	10	50	50	50	50

Le calcul de chaque sortie se fait en évaluant d'abord la matrice des valeurs de vérité des règles (degrés d'activation des règles) :

$$\omega = \mu'_{E'} * \mu_E \quad (\text{III.31})$$

Puis, le couple de commande est calculé par la méthode de défuzzification du centre de gravité simplifiée :

$$\tau = \frac{\sum_j^n \sum_i^n \omega_{i,j} * C_{i,j}}{\sum_j^n \sum_i^n \omega_{i,j}} \quad (\text{III.32})$$

Avec : $n=7$, $i=j=1 \dots, n$. et C : la matrice de la table de décision.

Pour calculée $\mu'_{E'}$ et μ_E on a utilisé la fonction triangle sur MATLAB voir (annexe A.3).

Simulation et résultats

Après la simulation de l'organigramme donné dans la figure A.3 (annexe A) on a, aboutit aux résultats de la (figure III.16).

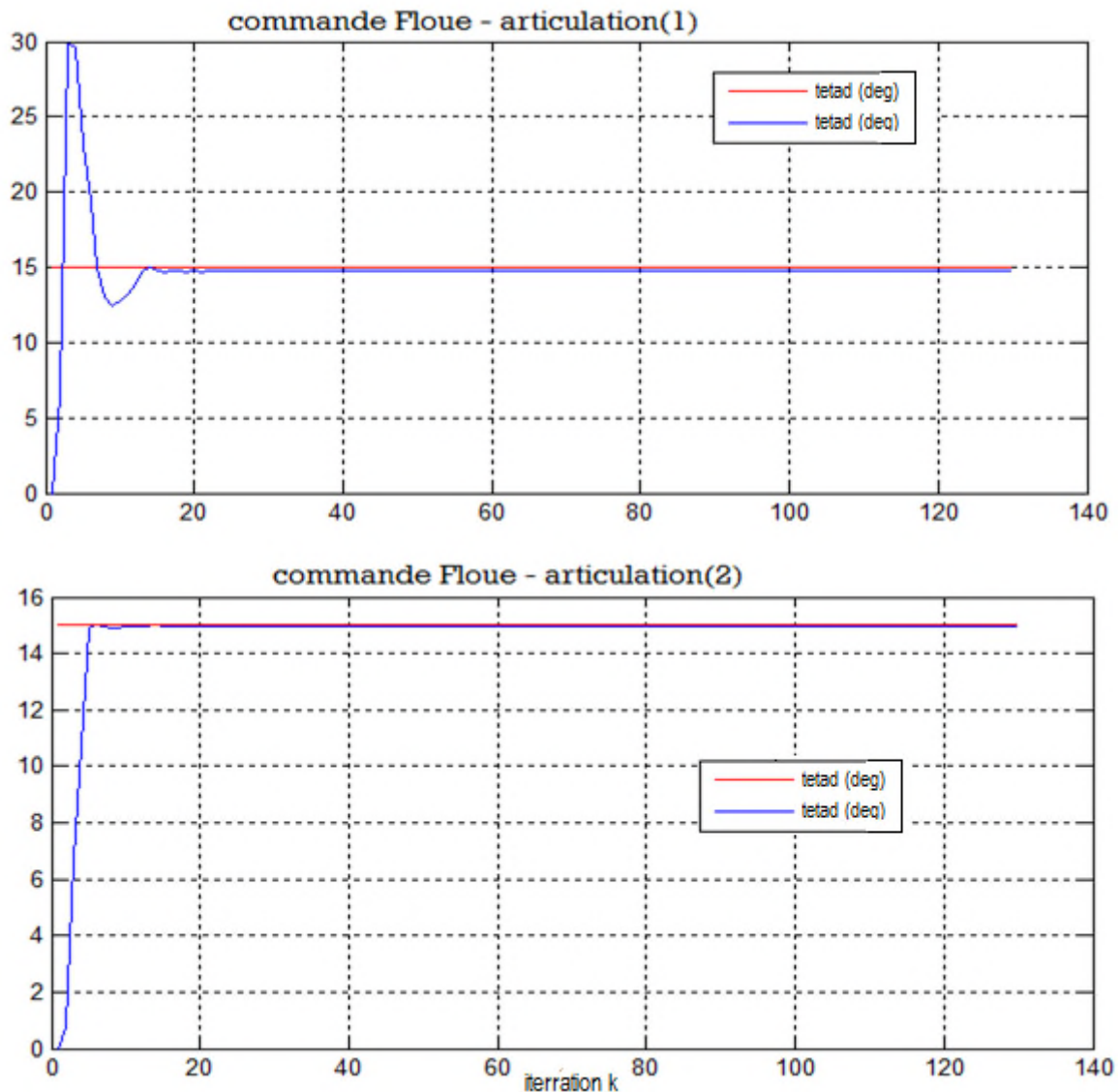


Figure III.16: Résultats de la commande floue

Au départ, on remarque que la première articulation a un dépassement très important. Car, elle atteint la valeur 30.28 degrés qui est loin de celle de θ_d ; qui est dûe au poids de la deuxième articulation, et aux forces centrifuges, Coriolis et les forces gravitationnelles, avant d'avoir une valeur qui est presque la même avec θ_d où $\theta_{f1} = 14.7244$ degrés par contre la deuxième articulation elle atteints presque la valeur désirée, $\theta_{f2} = 14.9345$ degré. Et ces erreurs due a les forces gravitationnels et de frottements.

Dans le cas où on aurait simulé cette commande avec compensation de gravité (programme de l'Annexe A.4) on a abouti aux résultats de la Figure III.17.

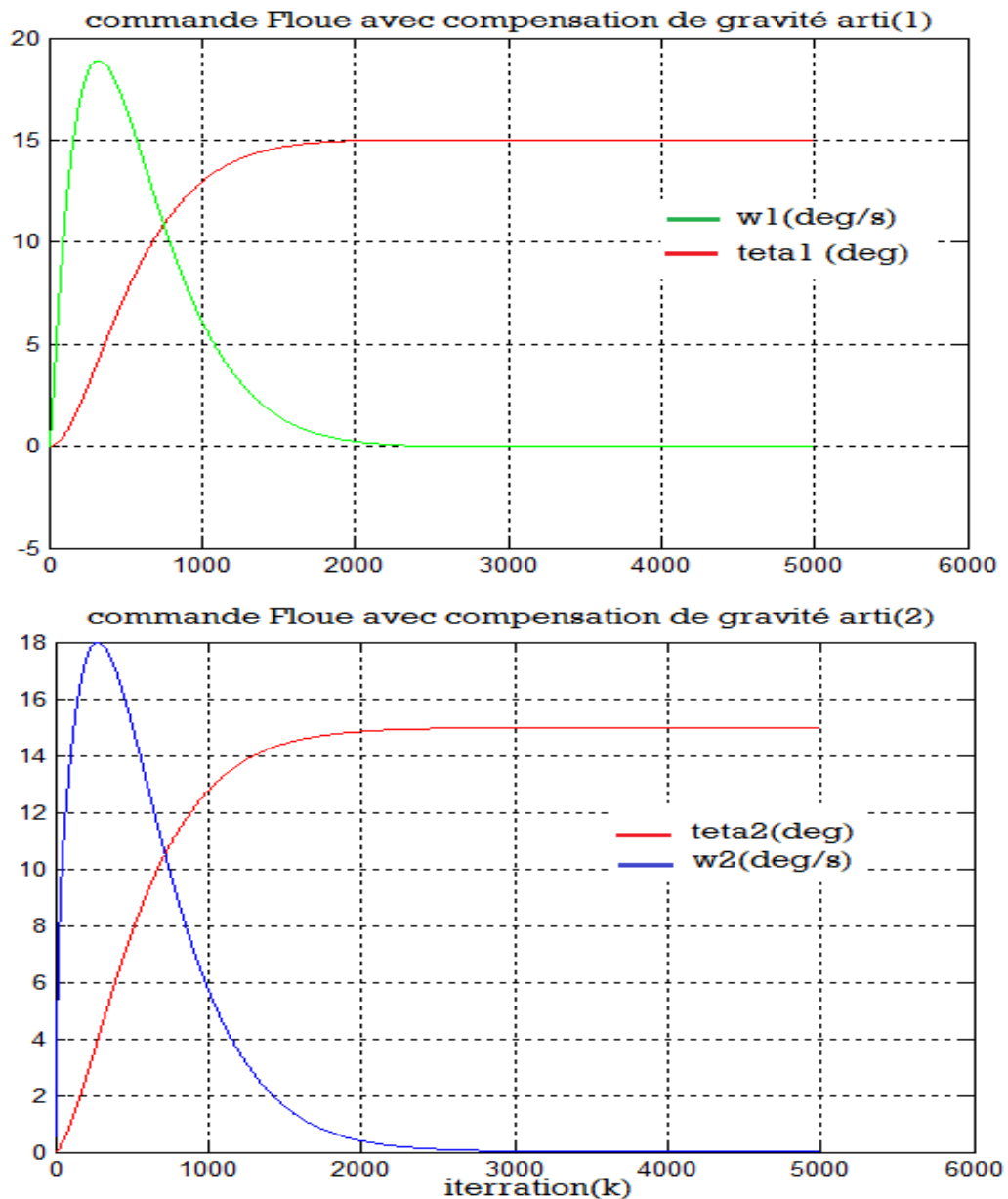


Figure III.17 : Résultats de commande floue avec compensation de gravité

D'après ces résultats on constate que les valeurs de θ_{f1} et θ_{f2} ont la même valeur de celle de θ_d et en un temps court sans avoir un dépassement, ce qui implique les bonnes performances de la commande ; avec un bon choix de la table de décision et les fonctions d'appartenance

III.7 Commande en poursuite

Le problème de poursuite de trajectoire dans l'espace articulaire, consiste à suivre une trajectoire donnée, $\theta_d(t)$ et ses dérivées successives $\dot{\theta}_d(t)$ et $\ddot{\theta}_d(t)$ qui décrivent la vitesse désirée et l'accélération désirée respectivement, générée par un module de génération (MGT), On cherche à minimiser l'erreur de poursuite définie comme suit :

$$E(t) = \theta_d(t) - \theta(t)$$

Reprenons le modèle non linéaire (III.4) et la loi de commande (III.5), on écrit alors :

$$\tau = \alpha \tau' + \beta A \text{vect}' = \ddot{\theta}_d + k_v \dot{e} + k_p e \quad (\text{III.32})$$

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

$$\begin{aligned} \ddot{\theta} &= \ddot{\theta}_d + k_v \dot{e} + k_p e \\ \ddot{e} + k_v \dot{e} + k_p e &= 0 \end{aligned} \quad (\text{III.33})$$

Qui peut être mise sous la forme standard

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

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

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 avec un module de génération de trajectoire.

Dans ce qui suit, on a une trajectoire désirée : générée avec MGT, 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 ($t_f = 15s$).

III.7.1 Commande PD en poursuite

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

$$E = [\theta_d(t) - \theta(t)]$$

$$\dot{E} = [\dot{\theta}_d(t) - \dot{\theta}(t)]$$

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 du robot.

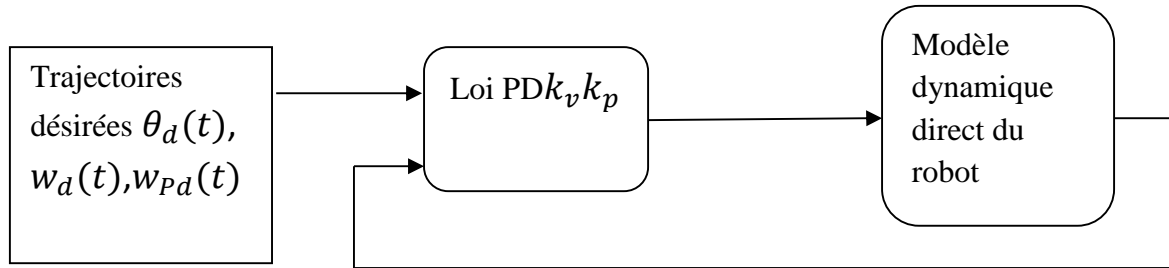


Figure III.18: Schéma synoptique de la commande PD en poursuite du bras de robot.

Simulation et résultat

Pour générer les trajectoires désirées, on a créé un (fichier.m) sur MATLAB (donnée en annexe A.6). À l'aide de la commande « savegeneplot123 » de MATLAB on sauvegarde les valeurs instantanées de ces trajectoires, et avec la commande « loadgeneplot123 », on fait l'appel à ces trajectoires. Les paramètres de cette commande sont les mêmes que la commande en régulation (III.16) sauf que θ_d est variable cette fois-ci. Le programme de la commande est donné en (Annexe A.5)

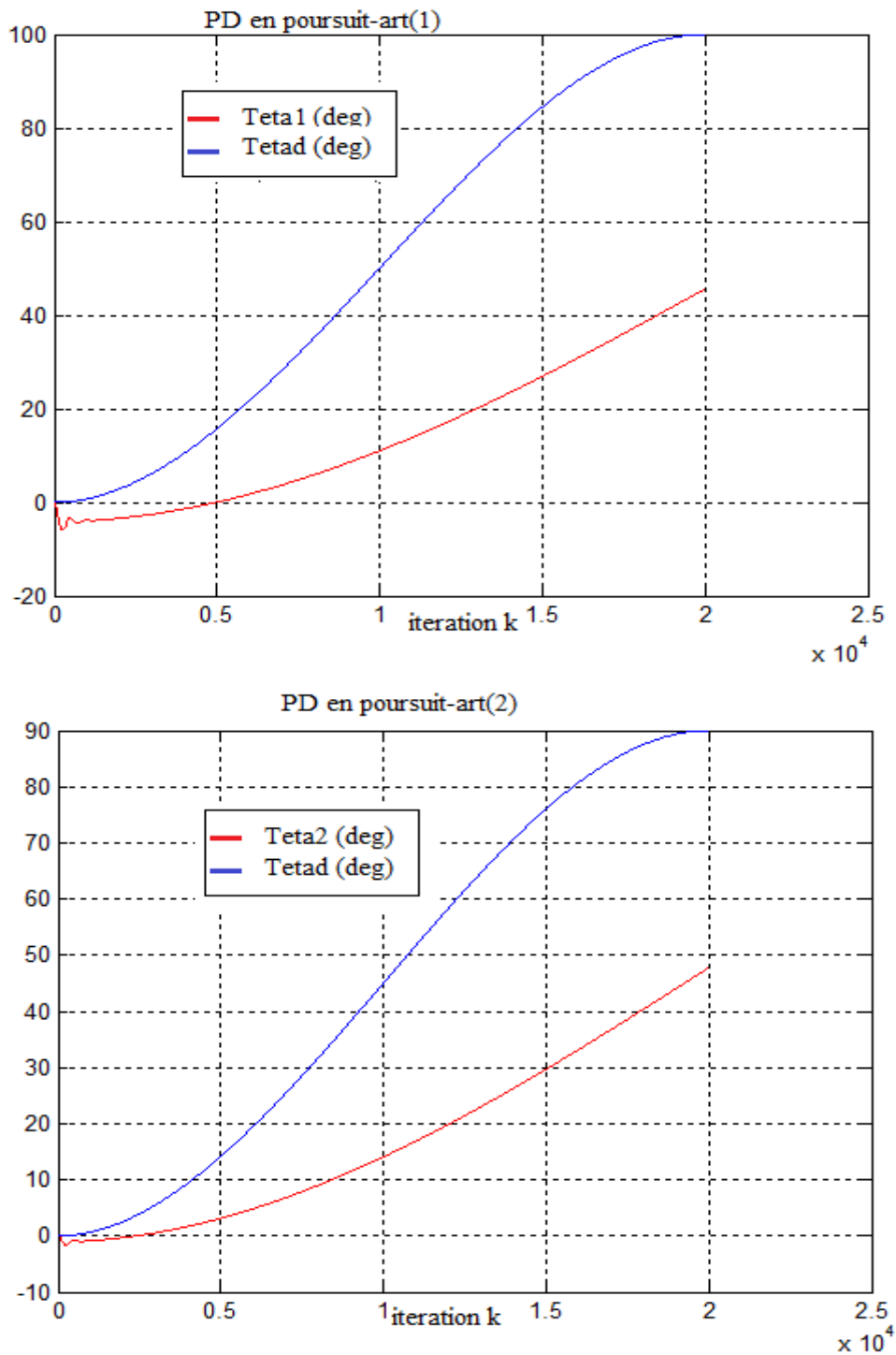


Figure III.19: Formes des variables articulaires et les vitesses angulaires

D'après les résultats, on constate la non convergence des trajectoires, qui est due à des erreurs d'intégration numérique pour avoir θ et $\dot{\theta}$ à partir de $\ddot{\theta}$. Pour cela, on était obligé de travailler avec les logigrammes de Simulink figure A.1 en (Annexe A). Les résultats obtenus sont donnés par la figure III.20.

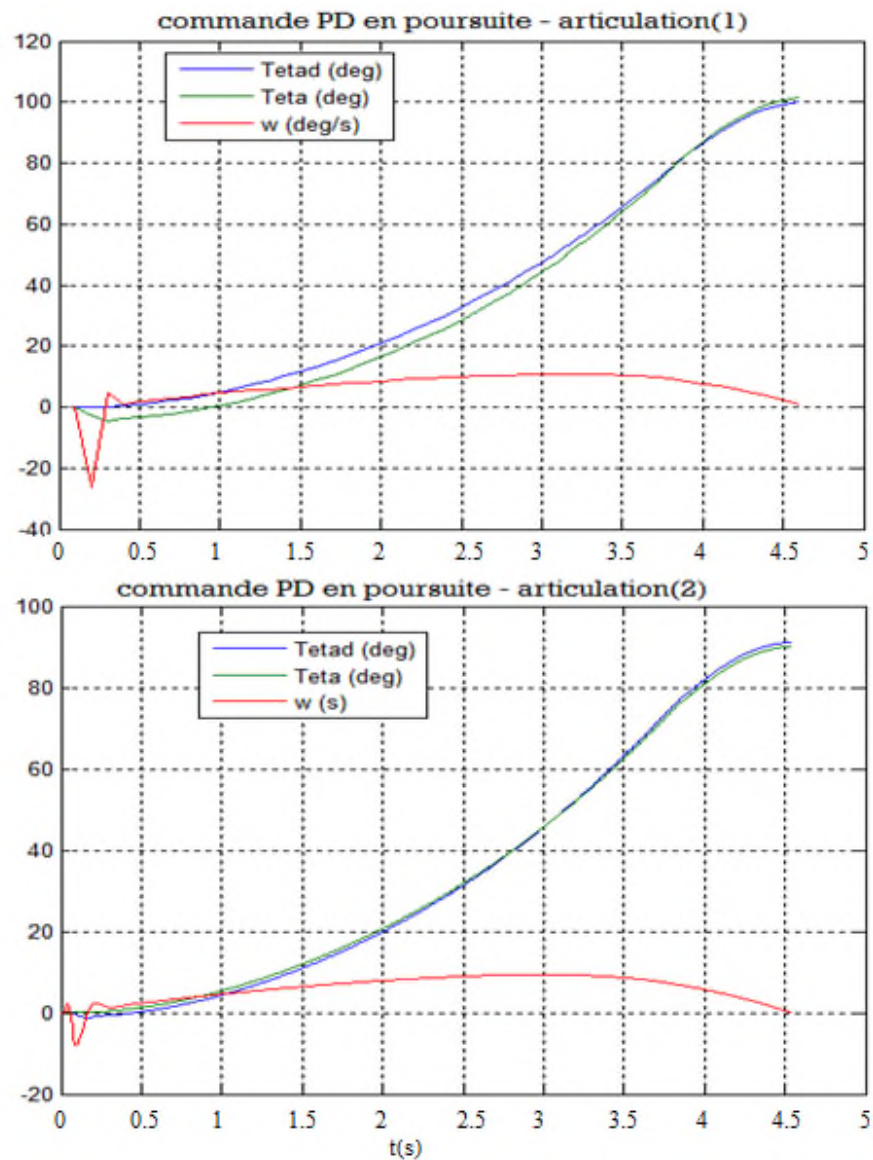


Figure III.20: La forme de la trajectoire de position et de vitesse.

On constate tout d'abord que la trajectoire de position et de vitesse sont négatives dans les premières secondes (due à la gravitation), ensuite on a une convergente presque parfaite de $\theta(t)$ à $\theta_d(t)$.

La figure nous montre donc que le système n'a pas atteint les performances désirées, cela est dû aux forces centrifuges et gravitationnelles.

Enfin, pour avoir les performances désirées, on a introduit la commande PD en poursuite avec compensation de gravité.

Après la simulation, on constate tout d'abord que les valeurs négatives de la trajectoire de position et de vitesse sont bien corrigées, et l'évolution de notre trajectoire suit correctement la trajectoire désirée (Figure III.21).

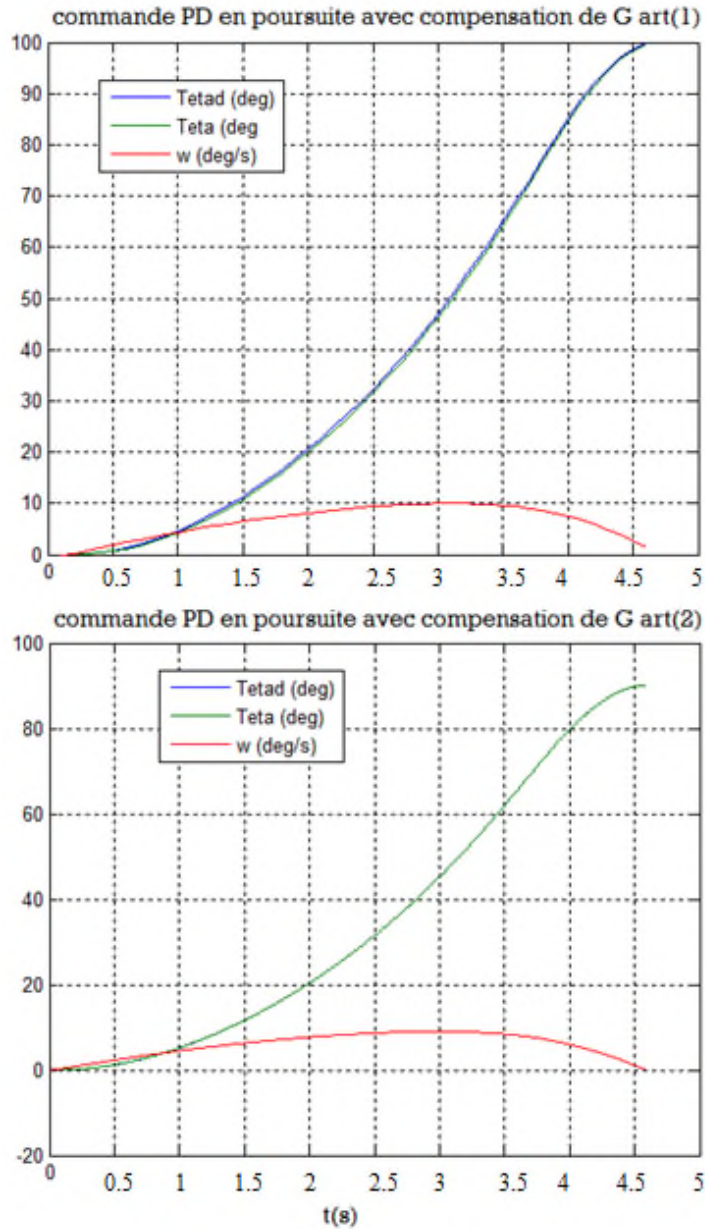


Figure III.21: La forme de la trajectoire de position et de vitesse.

III.7.2 La commande PID en poursuite

Pour élaborer cette commande, on suit le même modèle simplifié pour la commande (III.16), avec la variation de θ_d . Ce qui donne :

$$E(t) = [\theta_d(t) - \theta(t)]$$

$$E\dot{(t)} = [\dot{\theta}_d(t) - \dot{\theta}(t)]$$

$$\int E = \int [\theta_d(t) - \theta(t)] dt \quad (III.34)$$

La structure de cette commande est exactement la même que la commande PD en poursuite. 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 $\omega_n = 5 \text{ ras/sec}$. Ce qui donne les mêmes paramètres de (III.26)

$$\tau_1 = 35\dot{e}_1 + 275e_1 + 625 \int_0^t e_1(t) dt$$

$$\tau_2 = 35\dot{e}_2 + 275e_2 + 625 \int_0^t e_2(t) dt$$

Simulation et résultats

Après la simulation du système de la figure A.2 en l'Annexe A, on a abouti aux résultats de la figure III.22.

On constate tout d'abord que la trajectoire de position est négative dans les premières secondes, ensuite on voit que la trajectoire de la position suit correctement la trajectoire désirée. La figure nous montre donc que le système n'a pas atteint les performances désirées, cela est dû aux forces centrifuges et gravitationnelles.

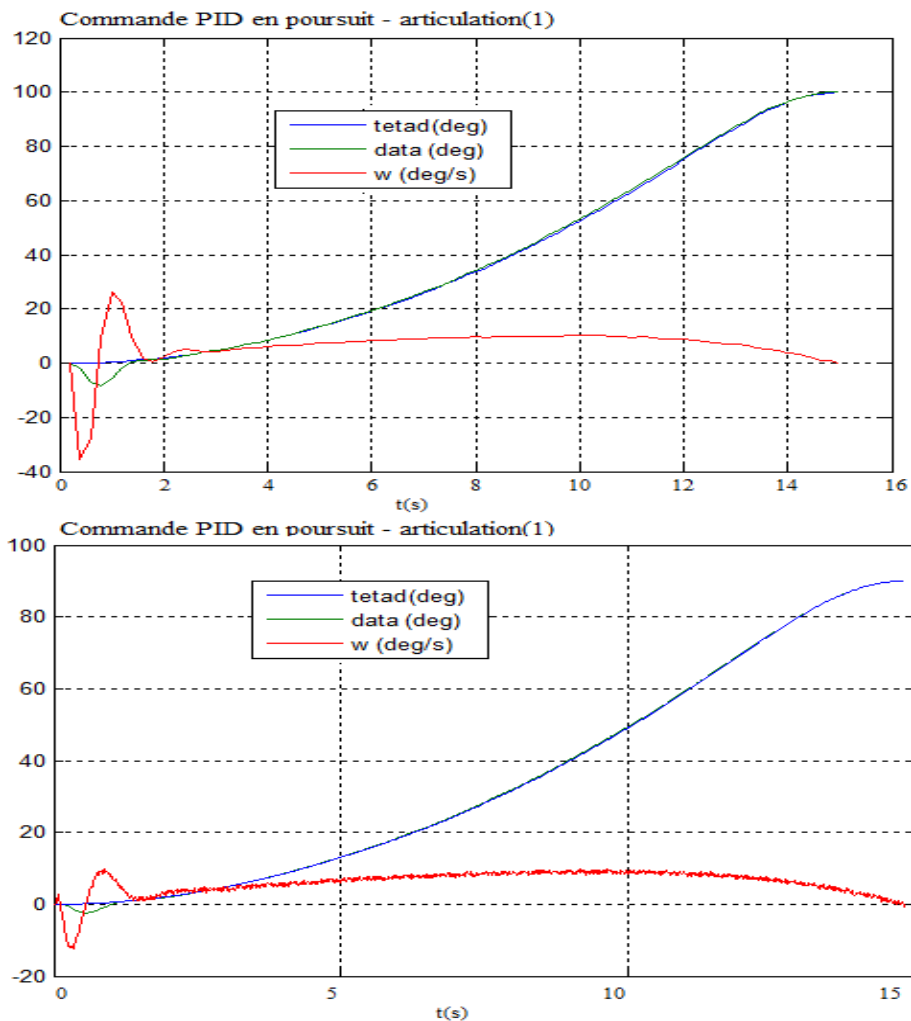


Figure III.22: La forme de la trajectoire de position et de vitesse pour la commande PID en poursuite

Enfin, pour avoir les performances désirées on a introduit la commande PID en poursuite avec compensation de gravité.

Après avoir simulé cet organigramme (figure III.23), on constate que l'erreur est bien corrigée, car l'évolution de notre trajectoire suit correctement la trajectoire désirée.

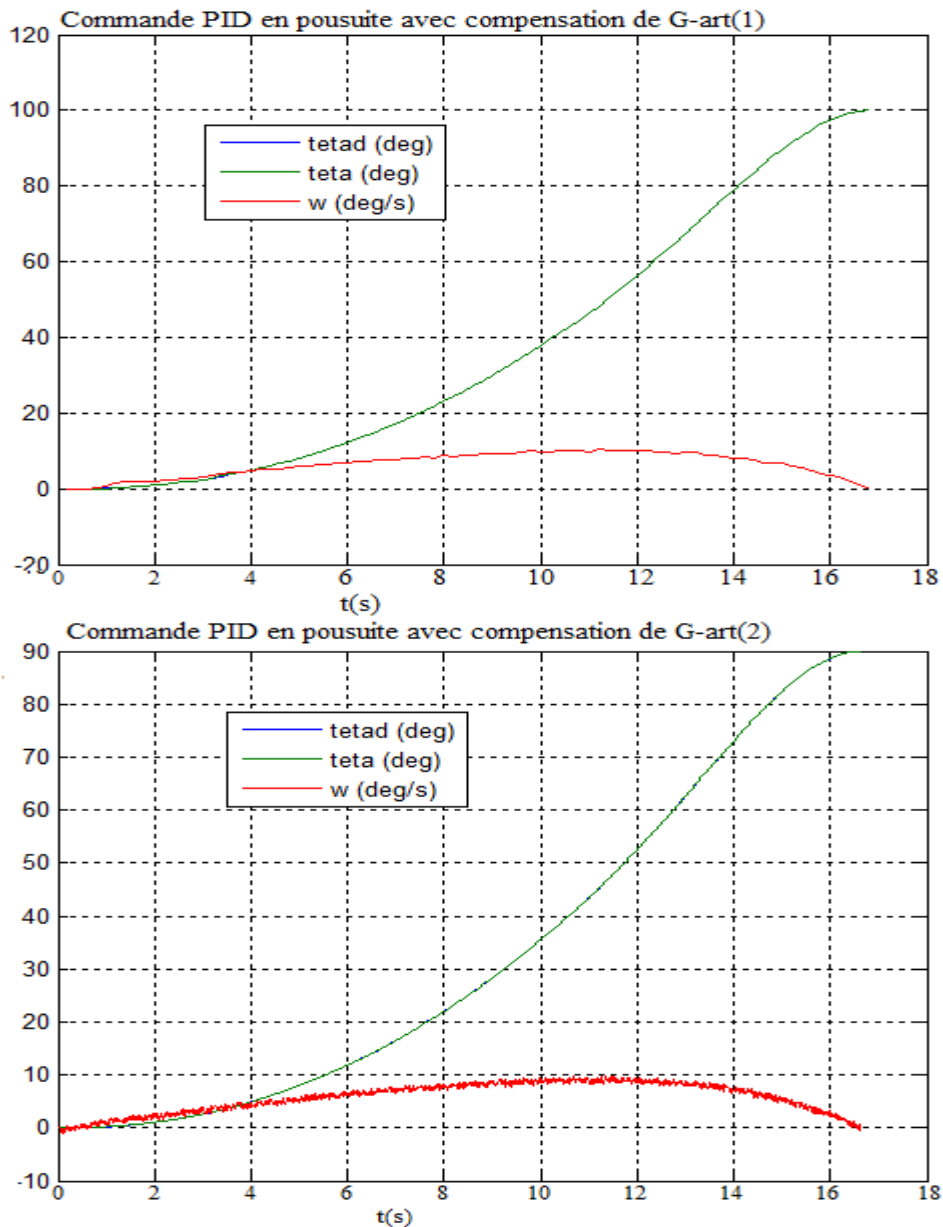


Figure III.23: La forme de la trajectoire de position et de vitesse.

III.7.3 La commande floue en poursuite

La structure de cette commande est exactement la même que la commande floue en régulation, mais avec la variation de θ_d . Dans ce cas, on a introduit l'organigramme de la figure A.4 (Annexe A) qui contient quatre blocs un pour la génération des trajectoires (MGT), les deux autres pour générer τ avec la commande floue et le dernier pour calculer $\ddot{\theta}$ avec l'équation de modèle dynamique direct (III.3).

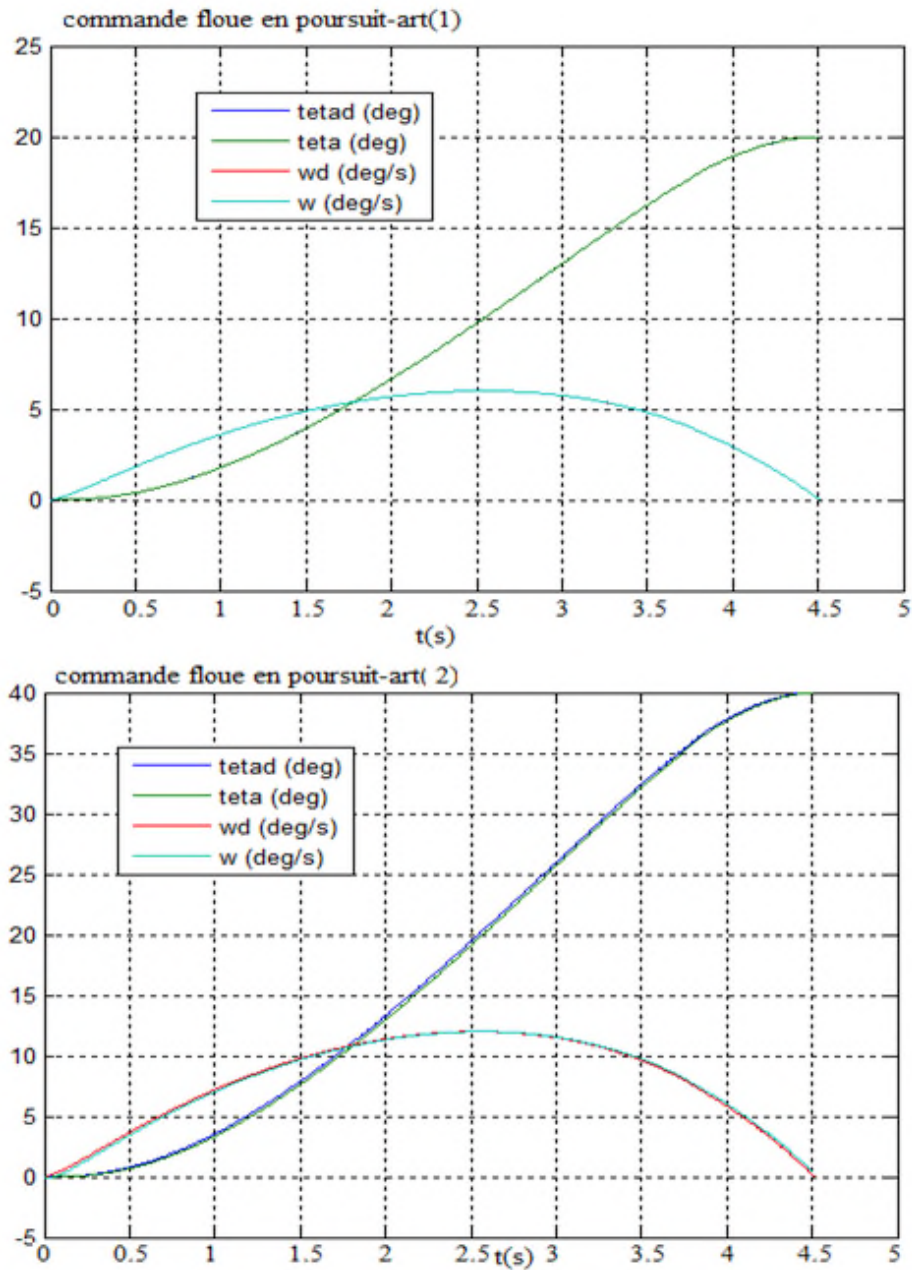


Figure III.24: La forme de la trajectoire de position et de vitesse avec la commande floue en poursuite.

Comme nous le constatons, les trajectoires obtenues (figure III.24) par le contrôleur flou suivent assez bien les trajectoires qui lui sont assignées.

III.8 Conclusion

Ce chapitre est consacré à la présentation des résultats de simulation suite à l'implémentation logicielle des différentes commandes. En fait, en connaissant le modèle du robot, certaines commandes basées modèle donnent des résultats satisfaisants, notamment la commande dynamique linéarisante. Les performances des commandes (PD et PID) restent valables autour du point de linéarisation du modèle. La commande floue à l'origine est conçue pour les systèmes mal définis ou inconnus. C'est approche en boîte noire dont les résultats dépendent des connaissances qualitatives du concepteur. Elle est efficace et plus indiquée dans les situations d'incertitudes et de difficultés de modélisation.

Conclusion générale

Notre travail a consisté à étudier deux types de commande, la commande classique et la commande floue en régulation et en poursuite de trajectoires dans le but de faire une comparaison de performances pour un bras manipulateur.

Dans le premier chapitre, on a donné une introduction aux robots industriels et leurs éléments constitutifs. Puis, on a exposé, dans le deuxième chapitre, la modélisation (géométrique, cinématique et dynamique) des bras de robots.

Le fond du travail est exposé dans le dernier chapitre. Quelques techniques classiques de commande des robots manipulateurs, telle que la commande PID et la commande dynamique, ont été étudiées et comparées à la commande floue qui fait l'objet de notre projet. Les résultats de simulation sur un bras à deux degrés de liberté nous permis d'avoir une idée claire sur les performances et les caractéristiques de chacune et de confirmer l'intérêt et l'efficacité que porte la commande floue dans ce domaine. Retenons que l'intérêt majeur de celle-ci en commande réside dans sa capacité à traduire une stratégie de contrôle d'un opérateur qualifié en un ensemble de règles linguistique facilement interprétables.

- [1] Philippe Coiffet, « *La Robotique, principes et applications*, » 3^{ème} Edition Hermes,1992.
- [2] H. Hamdi, « *Introduction à la Robotique*, » Les éditions de l'université Mentouré Constantine, 2002-2003.
- [3] E. Dombre,«*Analyse et Modélisation des Robots Manipulateurs*, » Hermès-Lavoisier, 2002.
- [4] A. Chrifa, « *Analyse Cinématique des Robots Parallèles*, » Thèse Docteur D'état. Université Mentouri Constantine, 2008.
- [5] Mr. *BENZOUAOUI AHMED*, « *Contrôle Robuste Des Robots Manipulateurs Rigides* » , mémoire Magister, Université des Sciences et de la Technologie d'Oran Mohamed BOUDIAF.2010.
- [6] Jean-Louis, Boimond, « *Robotique*, » Cours de robotique, ISTIA, Université Angers, 2008.
- [7] PierreMayé, « *Moteurs électriques pour la robotique* » - 3e édition,Dunod, 2016.
- [8] Luc Lasne, « *Exercice et problème d'électrotechnique*, » Dunod, Paris, 2005.
- [9] Bernard BAYLE, « *Modélisation et commande des robots manipulateurs*, » Cours de robotique, Université de Strasbourg,2009.
- [10] Philippe Coiffet et Michel Chirouze«*Eléments de Robotique*», Hermes, 1982.
- [11] R. MERZOUKI, « *Modélisation des systèmes articulés rigides*, » Cours de robotique, Master 1 polytechnique de Lille/France 2005.
- [12] Wisama KHALIL et Etienne DOMBRE ,« Bases de la modélisation et de la commande des robots-manipulateurs de type série », Université numérique ingénierie et technologie(UNIT),France, 08 mai 2012.
- [13] Ilian Bonev, Yanick Noiseux, Notes de cours , École de technologie supérieure(GPA), Montréal (Québec), 27 mars 2013.
- [14] Khalissa Behih Aida Cherif e, Djamila Zehar « modélisation et commande adaptative floue d'un rebot Manipulateur » Presse académique France (PAF) , 27/ 10 /2016.

- [15] Mr B. Mendil, « note de cour Modélisation Dynamique », université A.mira de Bejaia, 2016.
- [16] Azzaoui Ahlem Yasmine, Boudjema Nouria, La réalisation d'un bras palettiseur afin d'améliorer la productivité, mémoire master, Université Aboubakr Belkaïd – Tlemcen ,2016.
- [17] I.H. AGUILAR. Commande des bras manipulateurs et retour visuel des applications à la robotique de service. Thèse du doctorat de l'Université Toulouse III/France/2007 (spécialité robotique)
- [18] Jon Graig « Introduction to Robotics Mechanics and Control », Printice Hall, 3^{eme} edition 2005.
- [19] Moussaoui selma, « Commande floue d'un moteur a courant continu(MCC). Comparaison avec la commande PID » Université de A. Mira de Bejaia, 2017.
- [20] Mr B. Mendil, « note de cour, Commande des Bras Manipulateurs », », université A.mira de Bejaia, 2016.
- [21] Franck Deroncourt, « Introduction à la logique floue », www.openclassrooms.com,2012.
- [22] S.Benmeziane, K.Kouche, « Commande Géométrique et Dynamique de Bras manipulateur « IR50p » par logique Floue », Mémoire de fin d'études, Université A.Mira Bejaia, 20004.
- [23] Mr B. Mendil, « Cours technique intelligence artificiel », », université A.mira de Bejaia, 2016.

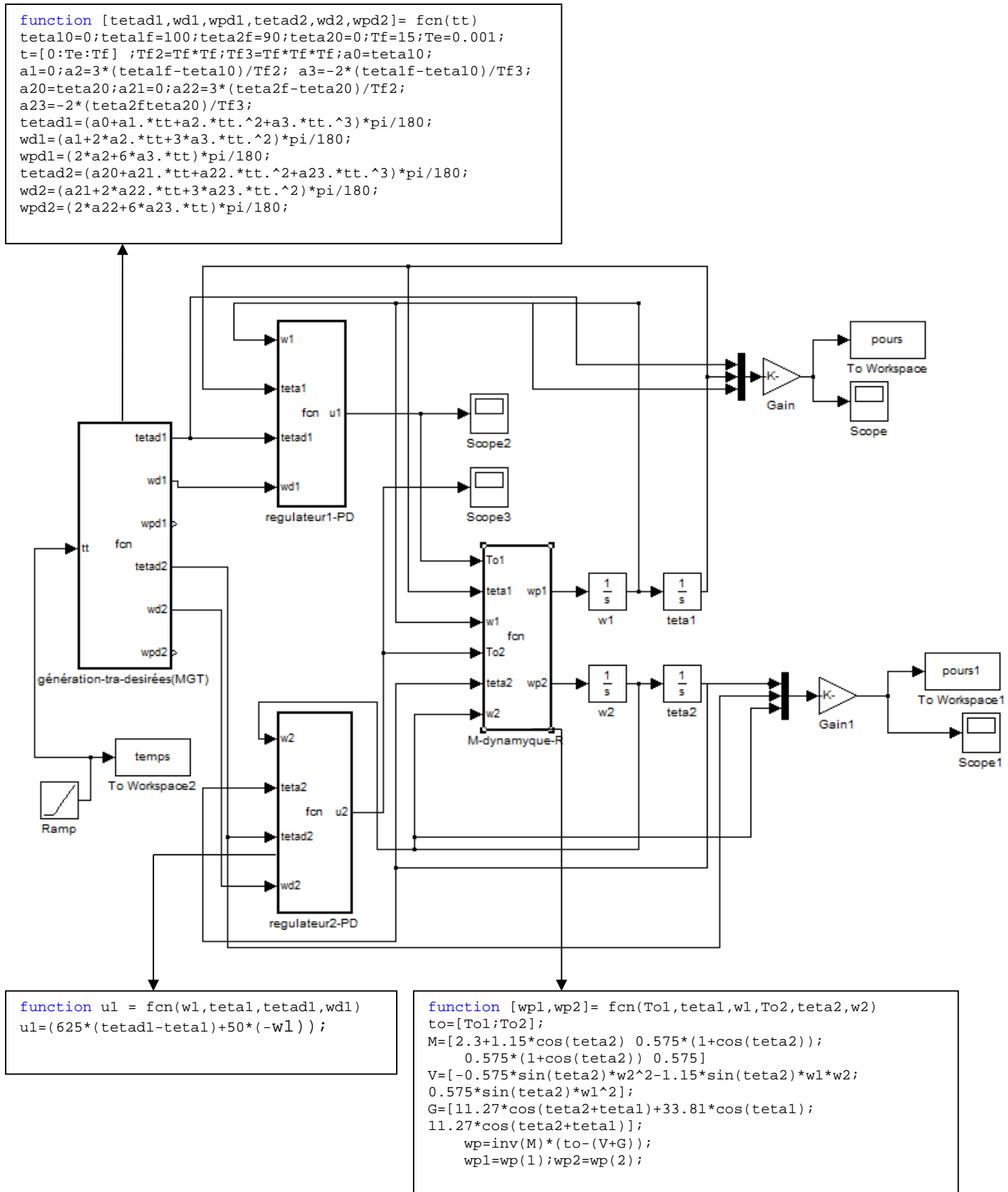


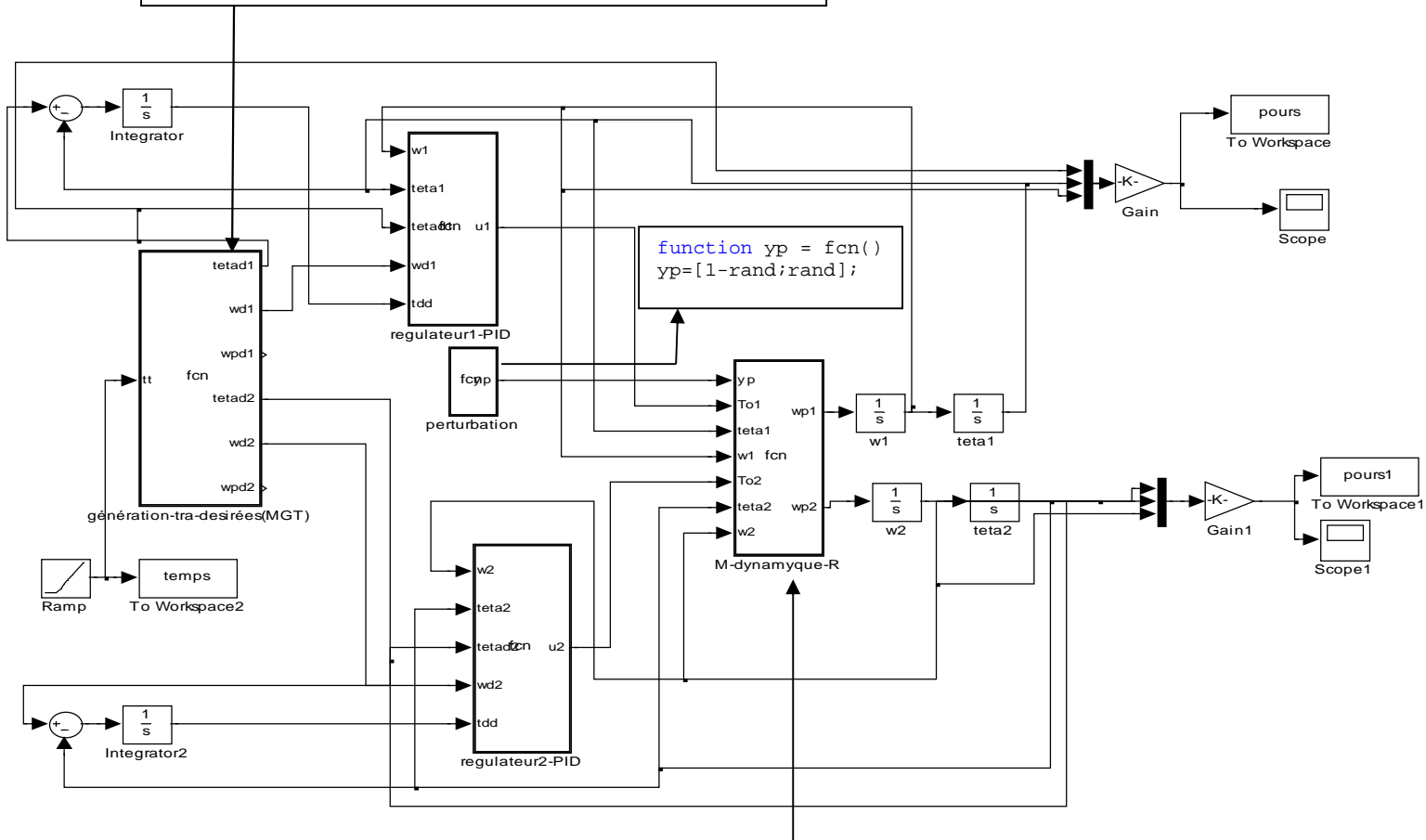
Figure A.1 : Organigramme de la commande PD en poursuite.

```

function [teta1,w1,wp1,teta2,w2,wp2]= fcn(tt)
teta10=0;teta1f=100;teta2f=90;teta20=0;Tf=15;Te=0.001;
t=[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;
teta1=(a0+a1.*tt+a2.*tt.^2+a3.*tt.^3)*pi/180;
w1=(a1+2*a2.*tt+3*a3.*tt.^2)*pi/180;
wp1=(2*a2+6*a3.*tt)*pi/180;

teta2=(a20+a21.*tt+a22.*tt.^2+a23.*tt.^3)*pi/180;
w2=(a21+2*a22.*tt+3*a23.*tt.^2)*pi/180;
wp2=(2*a22+6*a23.*tt)*pi/180;

```



```

function [wp1,wp2]= fcn(yp,To1,teta1,w1,To2,teta2,w2)
to=[To1;To2];
M=[2.3+1.15*cos(teta2) 0.575*(1+cos(teta2));
0.575*(1+cos(teta2)) 0.575];
V=[-0.575*sin(teta2)*w2^2-1.15*sin(teta2)*w1*w2; 0.575*sin(teta2)*w1^2];
G=[11.27*cos(teta2+teta1)+33.81*cos(teta1); 11.27*cos(teta2+teta1)];

wp=inv(M)*(to-(V+G+yp));
wp1=wp(1);wp2=wp(2);

```

Figure A.2 : Organigramme de la commande PID en poursuite.

```

function [wp1,wp2]= fcn(To1,teta1,w1,To2,teta2,w2)
to=[To1;To2];
M=[2.3+1.15*cos(teta2) 0.575*(1+cos(teta2));
   0.575*(1+cos(teta2)) 0.575];
V=[-0.575*sin(teta2)*w2^2-1.15*sin(teta2)*w1*w2 ;
   0.575*sin(teta2)*w1^2];
G=[11.27*cos(teta2+teta1)+33.81*cos(teta1);
   11.27*cos(teta2+teta1)];
wp=inv(M)*(to-V+G);
wp1=wp(1);wp2=wp(2);

```

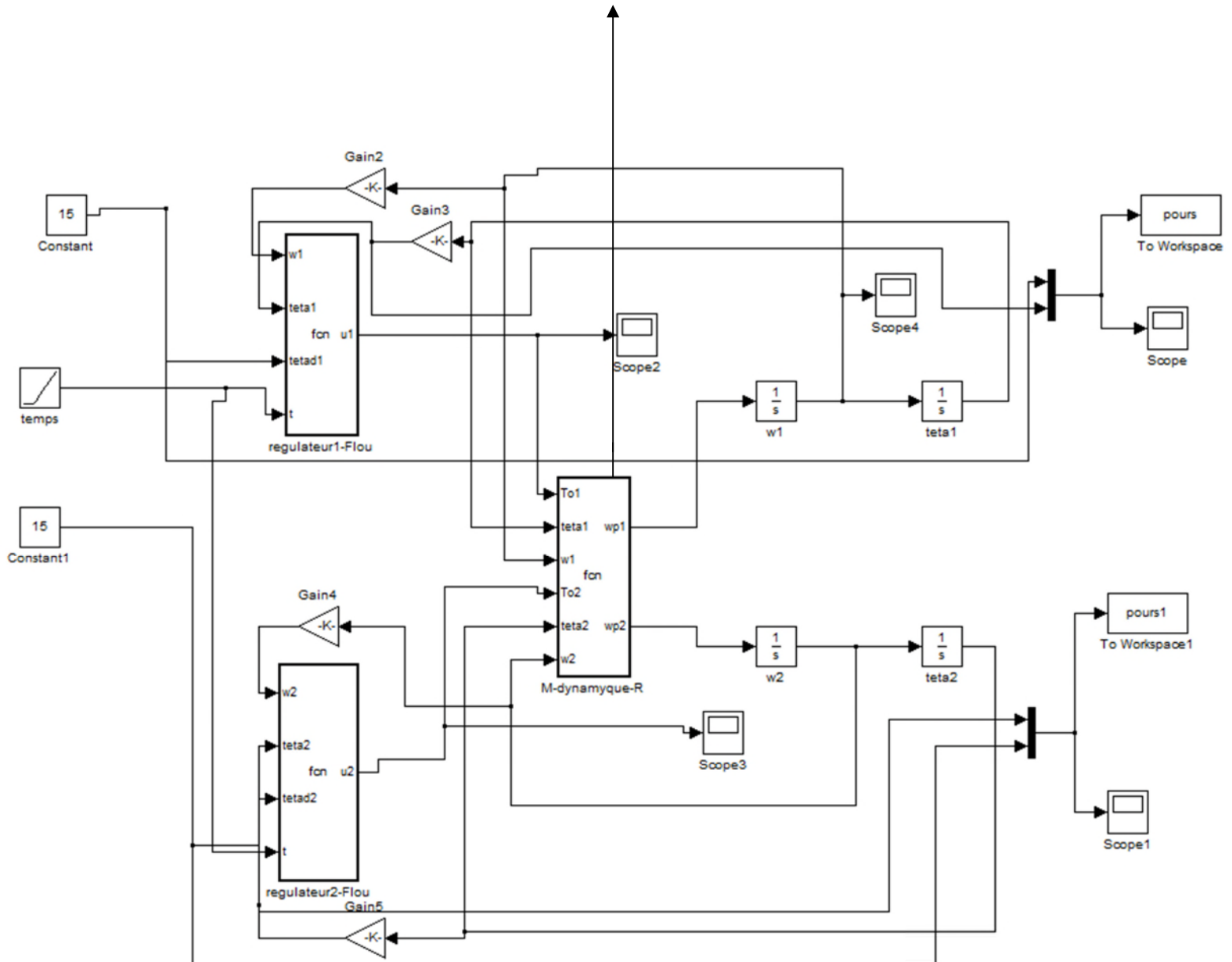


Figure A.3 Organigramme de la *commande floue en régulation*.

Régulateur Flou pour l'articulation 1 de la figure A.3

```

function u1 = fcn(w1,tetal,tetad1,t)
mut1=[0 0 0 0 0 0 0];mutp1=[0 0 0 0 0 0 0];ww1=mutp1'*mut1;
a=1; b=1;
if (t<0.2)tetad1=tetad1*(1-(0.2-t)/0.2)
    end;
E=10*(tetad1-tetal);
Ep=0.4*(-w1);
Ct=a*[-42 -18 -6 0 6 18 42];dCt=a*[24 12 6 3 6 12 24];g=Ct-dCt; d=Ct+dCt;
Ctp= b*[-70 -30 -10 0 10 30 70];dCtp=b*[40 20 10 5 10 20 40];gp=Ctp-dCtp;
dp=Ctp+dCtp;
c=[-50 -50 -50 -50 -10 -10 0;-50 -50 -50 -20 0 10 10;-50 -50 -50 -10 10 10
0;-50 -20 -10 0 10 20 50;-20 -10 -10 10 20 50 50;-10 -10 0 20 50 50 50;0 10
10 50 50 50 50];
%.....

for i=1:7
    if (E>g(i))&& (E<=Ct(i))
        mut1(i)=(E-g(i))/(Ct(i)-g(i));
    elseif (E>Ct(i)) && (E<=d(i))
        mut1(i)=(d(i)-E)/(d(i)-Ct(i));
    else
        mut1(i)=0;
    end;
end;
if E<Ct(1) mut1(1)=1; end;
if E>Ct(7) mut1(7)=1; end;

for i=1:7
    if (Ep>gp(i)) && (Ep<=Ctp(i))
        mutp1(i)=(Ep-gp(i))/(Ctp(i)-gp(i));
    elseif (Ep>Ctp(i)) && (Ep<=dp(i))
        mutp1(i)=(dp(i)-Ep)/(dp(i)-Ctp(i));
    else
        mutp1(i)=0;
    end;
end;

if Ep<Ctp(1) mutp1(1)=1; end;
if Ep>Ctp(7) mutp1(7)=1; end;

ww1=mutp1'*mut1;
tol=sum(sum(ww1.*c)/sum(sum(ww1)));
u1=5*tol;

```

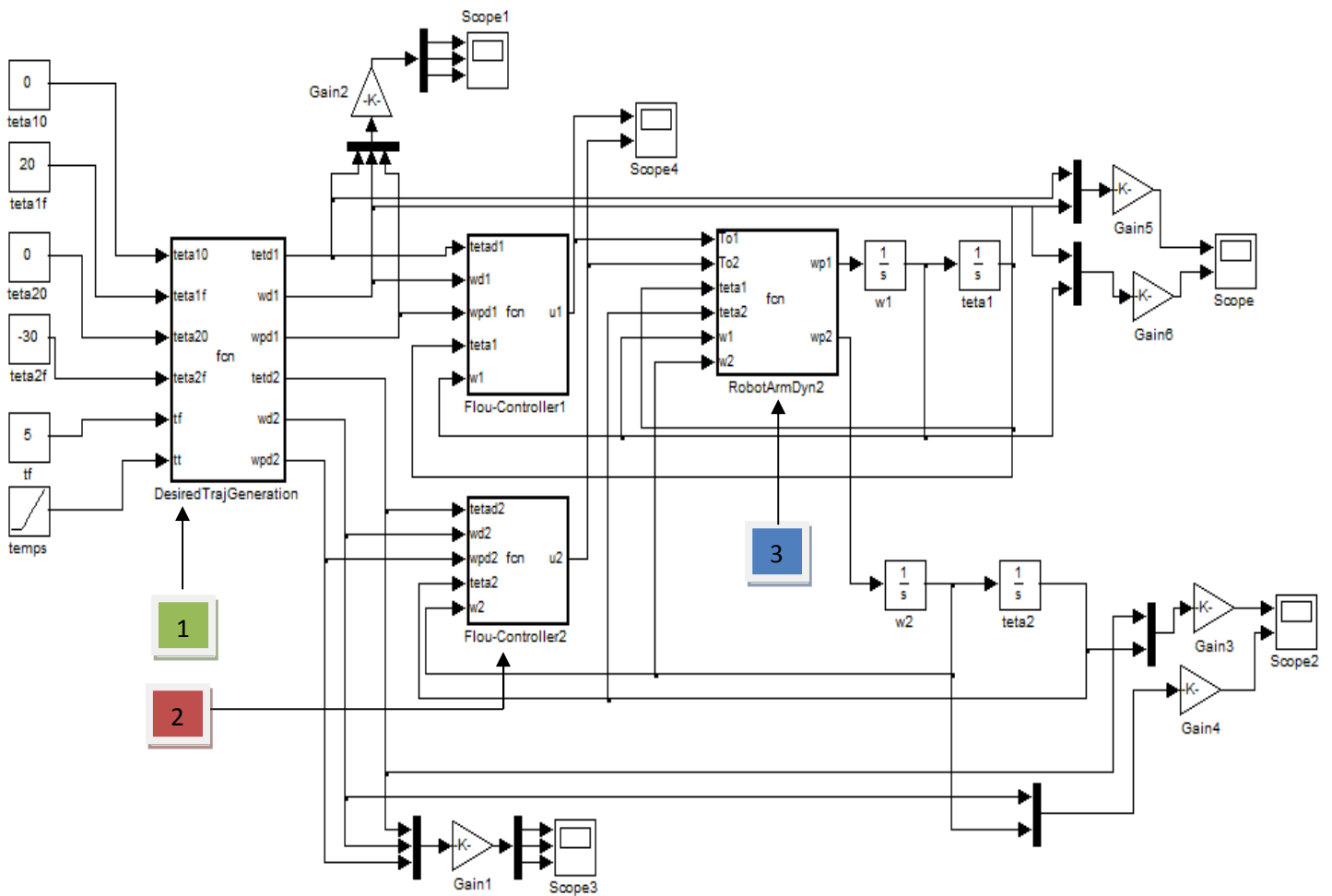


Figure A.4: Organigramme de la commande floue en poursuite.

1 Programme de génération de la trajectoire

```
function [tetd1,wd1,wpd1,tetd2,wd2,wpd2] =
fcn(teta10,teta1f,teta20,teta2f,tf,tt)
C=1
tf2=tf*tf; tf3=tf2*tf;
if C==1
    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
end
if C==2
    wpdmin1=4*(teta1f-teta10)/tf2; wa1=2*wpdmin1;
    tb=tf/2-sqrt(tf2/4-(teta1f-teta10)/wa1); t2=tf-tb
    tetb1=teta10+0.5*wa1*tb^2; wb1=wa1*tb;
    wpdmin2=4*(teta2f-teta20)/tf2; wa2=2*wpdmin2;
```

```

tetb2=teta20+0.5*wa2*tb^2;      wb2=wa2*tb;

if (tt<=tb)
    wpd1=wa1; wd1= wpd1*tt; tetd1=teta10+0.5*wpd1*tt^2;
    wpd2=wa2; wd2= wpd2*tt; tetd2=teta20+0.5*wpd2*tt^2
elseif (tb<tt) &&(tt<=t2)
    wpd1=0; wd1= wb1; tetd1=tetb1+wb1*(tt-tb);
    wpd2=0; wd2= wb2; tetd2=tetb2+wb2*(tt-tb);
else
    wpd1=-wa1; wd1= wb1-wa1*(tt-t2);
    tetd1= tetb1+wb1*(tt-tb)+0.5*wpd1*(tt-t2)^2;
    wpd2=-wa2; wd2= wb2-wa2*(tt-t2);
    tetd2= tetb2+wb2*(tt-tb)+0.5*wpd2*(tt-t2)^2;
end

tetd1=tetd1*pi/180; wd1=wd1*pi/180; wpd1=wpd1*pi/180;
tetd2=tetd2*pi/180; wd2=wd2*pi/180; wpd2=wpd2*pi/180;
end

```

2 Programme de régulateur flou

```

function u1 = fcn(tetad1,wd1,wpd1,teta1,w1)
a=0.01;b=0.2;
E=15*(tetad1-teta1);
Ep=0.5*(wd1-w1);

mut1=[0 0 0 0 0 0 0];
mutp1=[0 0 0 0 0 0 0];
ww1=mutp1'*mut1;
Ct=a*[-42 -18 -6 0 6 18 42];dCt=a*[24 12 6 3 6 12 24];g=Ct-dCt; d=Ct+dCt;
Ctp= b*[-70 -30 -10 0 10 30 70];dCtp=b*[40 20 10 5 10 20 40];gp=Ctp-dCtp;
dp=Ctp+dCtp;
c=[-50 -50 -50 -50 -10 -10 0;-50 -50 -50 -20 0 10 10;-50 -50 -50 -10 10 10
0;-50 -20 -10 0 10 20 50;-20 -10 -10 10 20 50 50;-10 -10 0 20 50 50 50;0 10
10 50 50 50 50];
%.....
for i=1:7
    if (E>g(i))&& (E<=Ct(i))
        mut1(i)=(E-g(i))/(Ct(i)-g(i));
    elseif (E>Ct(i)) && (E<=d(i))
        mut1(i)=(d(i)-E)/(d(i)-Ct(i));
    else
        mut1(i)=0;
    end;
end;
if E<Ct(1) mut1(1)=1; end;
if E>Ct(7) mut1(7)=1; end;
for i=1:7
    if (Ep>gp(i)) && (Ep<=Ctp(i))
        mutp1(i)=(Ep-gp(i))/(Ctp(i)-gp(i));
    elseif (Ep>Ctp(i)) && (Ep<=dp(i))
        mutp1(i)=(dp(i)-Ep)/(dp(i)-Ctp(i));
    else
        mutp1(i)=0;
    end;end;
if Ep<Ctp(1) mutp1(1)=1; end;
if Ep>Ctp(7) mutp1(7)=1; end;
ww1=mutp1'*mut1;
tol=sum(sum(ww1.*c)/sum(sum(ww1)));
u1=20*tol;

```

Programme de système dynamique

```

function [wp1,wp2] = fcn(To1,To2,teta1,teta2,w1,w2)
to=[To1;To2];
M=[2.3+1.15*cos(teta2)  0.575*(1+cos(teta2));
   0.575*(1+cos(teta2))  0.575];
V=[-0.575*sin(teta2)*w2^2-1.15*sin(teta2)*w1*w2
   0.575*sin(teta2)*w1^2];
G=[11.27*cos(teta1+teta2)+33.81*cos(teta1)
   11.27*cos(teta1+teta2)];
wp=inv(M)*(to-(V+G));
wp1=wp(1);wp2=wp(2);

```

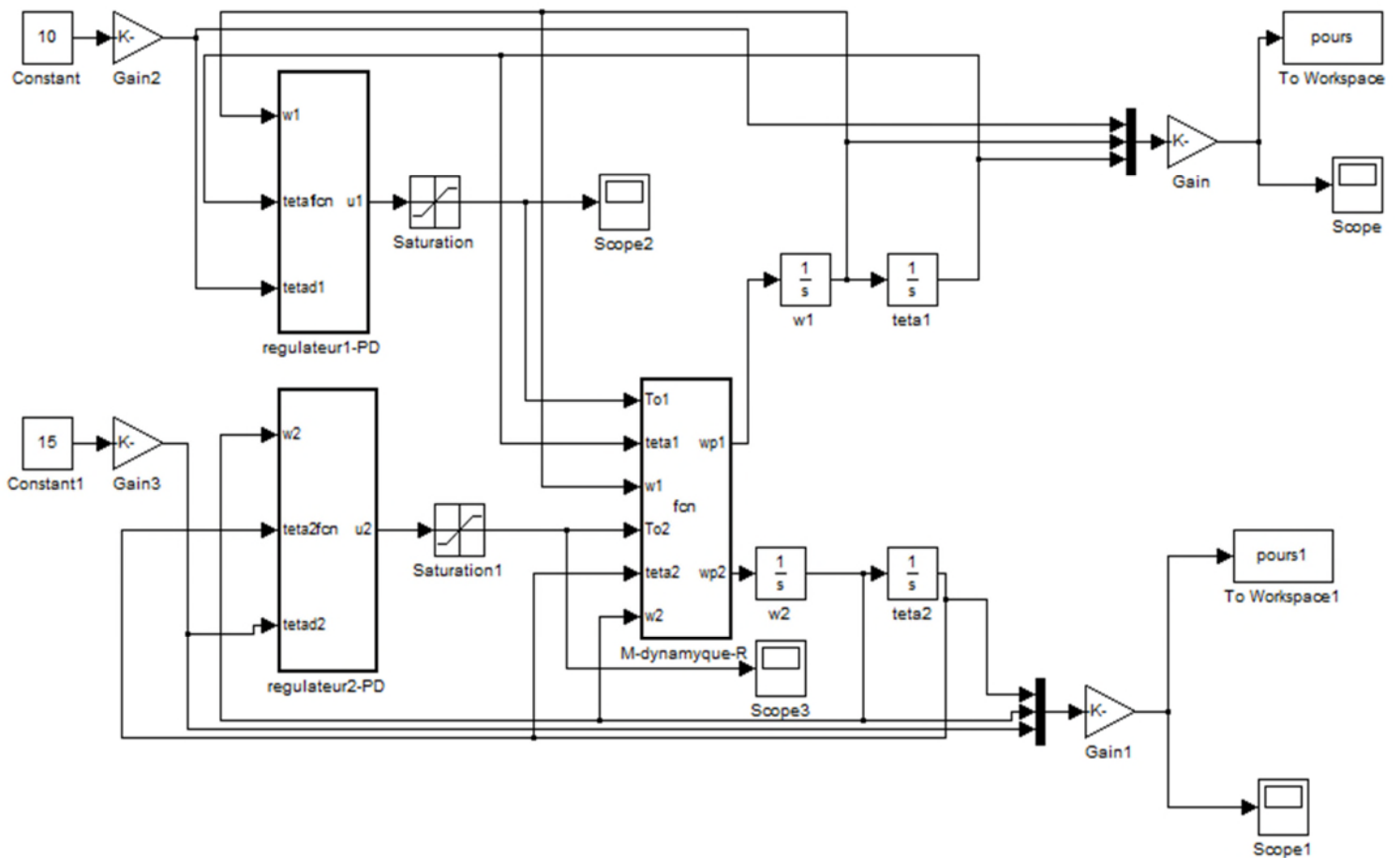


Figure A.5: organigramme de la commande PD

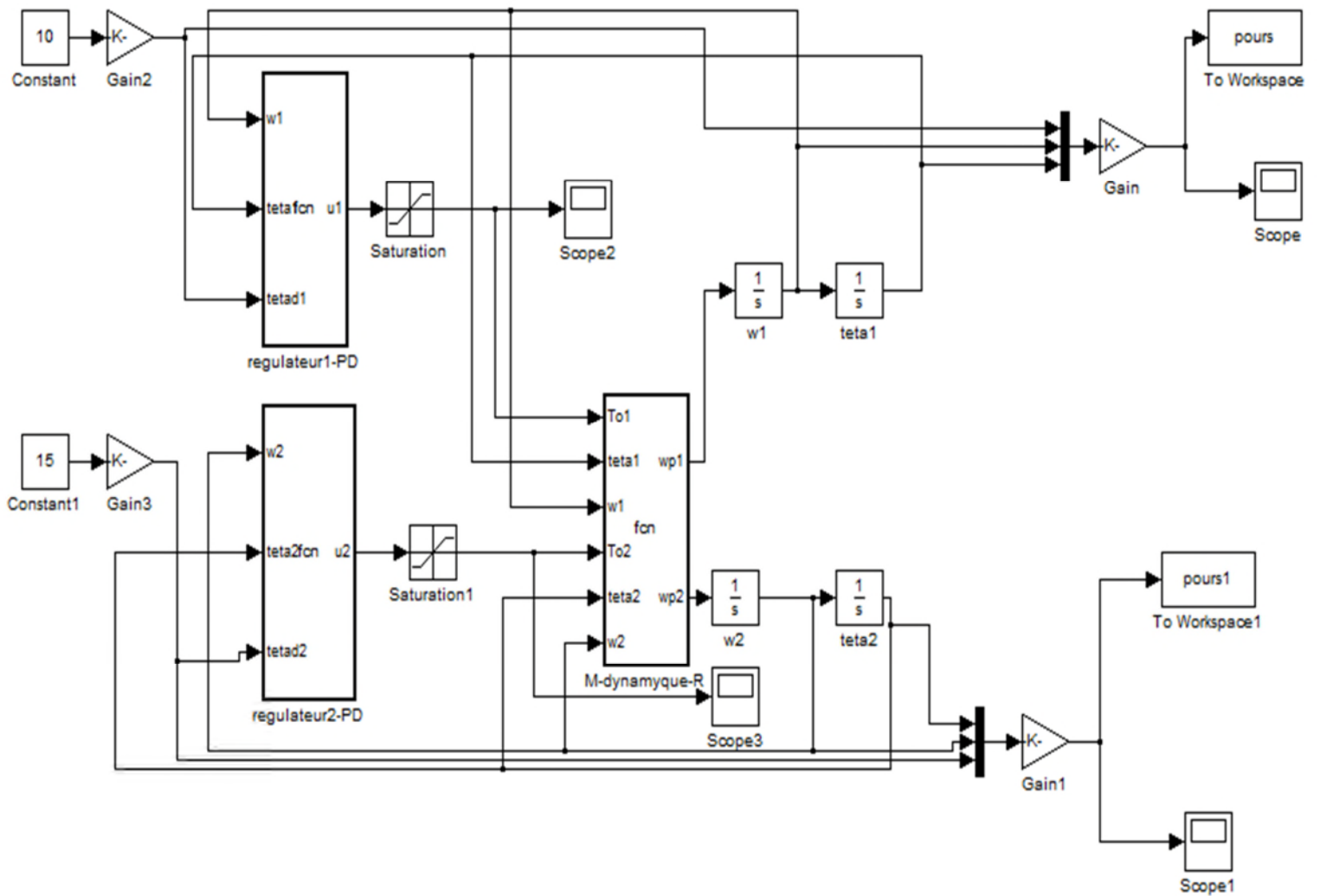


Figure A.6 : organigramme de la commande PID

1. Programme de la 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=[];tbb=[];
SI=[0;0];M1=[3.4499 0;0 0.575];
kvv=50;kpp=625;
JI=[1;1];Kp=kpp*JI;Kv=kvv*JI;
E=[],Ev=[];
ttd=[10;5]*pi/180;
k=1;G0=[m2*l2*g+(m1+m2)*l1*g;
        m2*l2*g];
for t=Te:Te:(tf+Te)
    E=ttd-teta;Ev=-w;
    SI=SI+E;
    to=Kv.*Ev+Kp.*E
    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*l2*l1*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)];
    wp=inv(M)*(to-(V+G));
    w=w+Te*wp;
    teta=teta+Te*w+0.5*wp*Te^2;
    bb=teta-ttd;
    tbb=[tbb E]
    a=[a teta];
    vi=[vi w];
    k=k+1
end;
figure
plot(a(1,:)*180/pi,'r')
hold on
plot(vi(1,:)*180/pi,'b')
grid
figure
plot(a(2,:)*180/pi,'r')
hold on
plot(vi(2,:)*180/pi)
grid

```

2. Programme de la 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;0 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]);
kvv=35;kpp=275;kii=625;
JI=[1;1];Kp=kpp*JI;Kv=kvv*JI;ki=kii*JI/1226
E=[],Ev=[];
ttd=[10;5]*pi/180;
k=1; G0=[m2*l2*g+(m1+m2)*l1*g;
        m2*l2*g];
for t=Te:Te:(tf+Te)
    E=ttd-teta;Ev=-w;
    SI=SI+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);l2^2*m2+l1*l2*m2*cos(teta2) l2^2*m2]
    V=[-m2*l2*l1*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
plot(a(2,:)*180/pi,'r')
hold on
plot(vi(2,:)*180/pi)
grid

```

3. Fonction triangulaire

```
function y=triang(x,g,c,d)
Nf=length(c);
for i=1:Nf
    y(i)=trimf(x,[g(i) c(i) d(i)]);
end;
if x<c(1)
    y(1)=1;
end;
if x>c(Nf)
    y(Nf)=1;
end;
return
```

4. Programme de la commande floue en régulation

```
clear all
clc
CE1=[-42 -18 -6 0 6 18 42];
CE1p= [-70 -30 -10 0 10 30 70];
dCE1t=[24 12 6 3 6 12 24];
dCE1tp=[40 20 10 5 10 20 40];
c=[-50 -50 -50 -50 -10 -10 0;-50 -50 -50 -20 0 10 10;-50 -50 -50 -10 10 10
0;-50 -20 -10 0 10 20 50;-20 -10 -10 10 20 50 50;-10 -10 0 20 50 50 50;0 10
10 50 50 50];
teta=[0*pi/180;0*pi/180]; w=[0;0]; wp=[0;0];Te=0.001; Te2=Te^2;tf=5;
l=0.5;l1=0.5; l2=0.5;m1=4.6;m2=2.3;m3=1;g=9.8;
a=[];vi=[];ac=[];tool=[];err=[]
SI=[0;0];M1=[3.4499 0;0 0.575];
kvv=50;kpp=625;
JI=[1;1];Kp=kpp*JI;Kv=kvv*JI;
EE=[],Ep=[];
ttd=[10;5]*pi/180;
k=1;
tetal=teta(1);teta2=teta(2);
G=[m2*l2*g+(m1+m2)*l1*g; m2*l2*g];

for t=Te:Te:(tf+Te)
    E=(ttd-teta);Ep=-w;
    mut1=triang(E(1),CE1-dCE1t,CE1,CE1+dCE1t)
    mutp1=triang(Ep(1),CE1p-dCE1tp,CE1p,CE1p+dCE1tp);
    w1=mutp1'*mut1
    to1=sum(sum(w1.*c)/sum(sum(w1)))*10;
    mut2=triang(E(2),CE1-dCE1t,CE1,CE1+dCE1t);
    mutp2=triang(Ep(2),CE1p-dCE1tp,CE1p,CE1p+dCE1tp);
    w2=mutp2'*mut2;
    to2=sum(sum(w2.*c)/sum(sum(w2)))*5;
    to=[to1;to2]; to=to +G;
    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*l2*l1*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-(V+G));
```



```

w=w+Te*wp;
teta=teta+Te*w+0.5*wp*Te^2;
bb=-teta+ttd;
err=[err bb];
a=[a teta];
vi=[vi w];
k=k+1;
end;
figure
plot(a(1,:)*180/pi,'r')
hold on
plot(vi(1,:)*180/pi,'g')
hold off
grid
figure
plot(a(2,:)*180/pi,'r')
hold on
plot(vi(2,:)*180/pi)
grid

```

5. Programme de la commande PD en poursuite

```

clear all
clc
load geneplot123
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=[];
SI=[0;0];M1=[3.4499 0;0 0.575];
kvv=50;kpp=625;
JI=[1;1];Kp=kpp*JI;Kv=kvv*JI;
E=[],Ev=[];TOO=[];
ttd=[100;50]*pi/180;
G0=[m2*l2*g+(m1+m2)*l1*g;
m2*l2*g];
k=1;
for t=Te:Te:(tf+Te)
E=tetad(k)-teta;Ev=tetadv(k)-w;
to=Kv.*Ev+Kp.*E
TOO=[TOO to];
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*l2*g*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-(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

```

```

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

```

6. La fonction MGT

```

clear all
clc
teta10=0;teta1f=100;teta2f=90;teta20=0;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;
tetad1=(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;
tetad2=(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;
tetad=[tetad1;tetad2];
tetadv=[wd1;wd2];
tetada=[wpd1;wpd2];
save geneplot123 tetad tetadv tetada

```

Fonction trimf de matlab

$y = \text{trimf}(x,[a \ b \ c])$

La courbe triangulaire est une fonction d'un vecteur, x , et dépend de trois paramètres scalaires a , b et c , donnés par

$$f(x; a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{b-c}, & b \leq x \leq c \\ 0, & c \leq x \end{cases}$$

Ou, de manière plus compacte, par

$$f(x; a, b, c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{b-c}\right), 0\right)$$

Les paramètres a et c localisent les "pieds" du triangle et le paramètre b localise le pic.

Remarque

Pour la compensation de gravité il suffit d'ajouter $G(t)$ à la loi de commande.

Chapitre I

GENERALITES SUR LES BRAS DE ROBOT

Chapitre II

MODELISATION DES BRAS DE ROBOTS

Chapitre III

COMMANDE FLOUE D'UN BRAS DE ROBOT

Bibliographie

Introduction Générale

Conclusion Générale

Table des matières

Annexe