

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

Université Abderrahmane Mira

Faculté de la Technologie



Département d'Automatique, Télécommunication et d'Electronique

Projet de Fin d'Etudes

Pour l'obtention du diplôme de Master

Filière : Automatique

Spécialité : Automatique et informatique industrielle

Thème

Planification du chemin et commande d'un bras de robot

Préparé par :

Mr. SAHI Abdelghani

Mr. SEMARA Mahmoud

Dirigé par :

Mr. MENDIL Boubekour

Examiné par :

Mr. LEHOUCHE Hocine

Mr. NAIT MOHAND Nacim

Année universitaire : 2021/2022

Remerciement

On tient tout d'abord à remercier en premier lieu Allah, le tout puissant, de nous avoir donné autant de courage, de patience et de volonté pour atteindre ce but.

On tient à exprimer à notre promoteur Mr. MENDIL Boubekour, notre profonde gratitude et toutes nos reconnaissances pour avoir accepté de diriger ce mémoire et de sa patience durant la période de l'encadrement.

Tous nos remerciements vont également aux membres de jury pour nous avoir fait l'honneur de juger ce modeste travail.

Nos remerciements vont également à toutes les personnes qui nous ont aidés et soutenus de près ou de loin dans la réalisation de ce travail.

Dédicace

Louanges à Dieu, le seul et unique.

Je dédie ce modeste travail à :

*À mes très chers parents qui ont tout fait pour me donner une bonne
éducation et me soutenir dans mes études, ainsi que leurs
compréhensions et encouragements ont pu me porter pour pouvoir
arriver au bout de mon travail, que Dieu leur procure bonne santé et
longue vie.*

Et bien sûr à mon frère Merouane.

Et ma sœur Mounia.

À toute ma famille SEMARA sans exception.

Et sans oublier la famille BENLOUNIS.

Mes très chers amis sans exception.

*Et à tous ceux qu'ont contribué de près ou de loin pour que ce projet
soit possible.*

Je vous dis merci.

SEMARA Mahmoud

Dédicace

A ma très chère mère

Quoi que je fasse ou que je dise, je ne saurai point te remercier comme il se doit. Ton affection me couvre, ta bienveillance me guide et ta présence à mes cotes a toujours été ma force pour affronter les différents obstacles.

A mon très cher père

Tu as toujours été à mes cotes pour me soutenir et m'encourager. Que ce travail traduit ma gratitude et mon affection.

A mon très chers frère Rayan, et ma très belle sœurs Chahrazed .

Puisse dieu vous donne santé, bonheur, courage et surtout réussite

SAHI Abdelghani

Liste des abréviations

DDL : degré de liberté.

AMR : Robots mobiles autonomes

AGV : Véhicules à guidage automatique

SMA : Système mécanique articulé

MGD : Modèle géométrique direct.

MGI : Modèle cinématique inverse.

MCD : Modèle cinématique direct.

MCI : Modèle cinématique inverse.

MDD : Modèle dynamique direct

MDI : Modèle dynamique inverse

DH : Denavit et Hartenberg

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

PID : Proportionnel Intégral et Dérivé.

PD : Proportionnel et Dérivé.

TL : Transformé de Laplace

CTC : Computed Torque Contrôle.

Sommaire

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

Chapitre I : Généralités sur les bras de robots

<i>I.1 Introduction.....</i>	<i>3</i>
<i>I.2 Définitions préliminaires.....</i>	<i>3</i>
1.2.1 Robotique	3
1.2.2 Robot	3
1.2.3 Articulation	3
1.2.4 Degré de liberté.....	4
1.2.5 Espace opérationnel	4
1.2.6 Espace articulaire	4
<i>I.3 Classification des Robots</i>	<i>4</i>
1.3.1 Robots mobiles autonomes (AMR)	4
1.3.2 Véhicules à guidage automatique (AGV).....	4
1.3.3 Humanoïdes.....	5
1.3.4 Cobots.....	5
1.3.5 Robots articules ou industriel.....	5
1.3.6 Robot Hybrides.....	5
<i>I.4 Éléments constitutifs des robots manipulateurs</i>	<i>6</i>
1.4.1 Système mécanique articulé	7
1.4.2 Organe terminal	7
1.4.3 Système de traitement.....	8
1.4.4 Actionneurs	8
1.4.5 Transmetteurs	9
1.4.6 Capteurs :	9
<i>I.5 Morphologie des bras manipulateurs :</i>	<i>10</i>
1.5.1 Structure cartésienne (PPP)	10
1.5.2 Structure cylindrique	10
1.5.3 Structure sphériques (polaires).....	10
1.5.4 Structure anthropomorphe	10
1.5.5 Structure SCARA	11

<i>I.6 Différentes générations de robots</i>	11
I.6.1 Les robots de première génération	11
I.6.2 Les robots de deuxième génération	11
I.6.3 Les robots de troisième génération	11
<i>I.7 Caractéristiques des Robots</i>	12
<i>I.8 Domaines d'application</i>	12
I.8.1 L'industrie	12
I.8.2 L'agriculture	12
I.8.3 L'Armée	13
I.8.4 La médecine	13
I.8.5 Autres Usages	13
<i>I.9 Conclusion</i>	13

Chapitre II: Modélisation des bras de robots

<i>II.1 introduction</i>	14
<i>II.2 Modélisation géométrique</i>	14
II.2.1 Description de la position et de l'orientation	14
II.2.2 Coordonnées homogènes d'un vecteur	15
II.2.3 Position et orientation d'un corps rigide	16
II.2.4 Transformation homogène	17
II.2.5 Transformation des repères	17
II.2.6 Matrices de rotations pures autour des axes principaux	17
II.2.7 Matrice de translations pures	18
II.2.8 Modèle géométrique direct	19
II.2.9 Modèle Géométrique Inverse	23
II.2.10 Model géométrique inverse, du robot PUMA560	24
<i>II.3 Modélisation cinématique</i>	24
II.3.1 Calcul de la matrice jacobienne par dérivation du MGD	25
II.3.2 Modèle cinématique inverse	27
<i>II.4 Le modèle dynamique</i>	28
II.4.1 Le modèle dynamique inverse	28
II.4.2 Le modèle dynamique direct	28
II.4.3 Modèle dynamique du robot PUMA 560	30
<i>II.5 Conclusion</i>	32

Chapitre III : Planification de Trajectoires

<i>III.1 Introduction</i>	33
<i>III.2 Planification de trajectoire</i>	33
III.2.1 Méthodes de Planification dans l'espace articulaire	33
III.2.2 Avantages et inconvénients	37
<i>III.3 Méthodes de Planification dans l'espace opérationnelle</i>	37
III.3.1 Polynôme cubique	38
III.3.2 Mouvement en points via	39
III.3.3 Avantages et inconvénients	39
<i>III.4 Planification de trajectoire pour le robot puma 560</i>	39
III.4.1 Planification dans l'espace articulaire	40
III.4.2 Planification dans l'espace opérationnel	44
<i>III.5 Conclusion</i>	47

Chapitre IV : Commande des bras de robots

<i>IV.1 Introduction</i>	48
<i>IV.2 Lois de commande de suivi de trajectoire</i>	48
IV.2.1 Commande classique par PID.....	48
IV.2.2 Commande PD.....	50
IV.2.3 Commande découplante linéarisante	51
<i>IV.3 Systèmes de contrôle cartésien</i>	52
<i>IV.4 Commande du robot PUMA560</i>	53
IV.4.1 Implémentation de la commande PD :	55
IV.4.2 Implémentation de la commande PD avec compensation de gravité	56
IV.4.3 Implémentation de la commande PID	58
IV.4.4 Implémentation de la commande CTC.....	60
<i>IV.5 Conclusion</i>	62
<i>Conclusion générale</i>	63
<i>Bibliographie</i>	64
<i>Annexes A</i>	66
<i>Annexes B</i>	72

Introduction générale

L'homme a depuis toujours été fasciné par la possibilité de pouvoir recréer la vie. Désir profond de jouer les créateurs, ou bien tentation de la mystification, ou encore simple jeu de copie de l'existant, les divers hommes d'airain, androïdes ou robots, créés dans la littérature, au cinéma ou dans la réalité, ont depuis la haute antiquité répondu à nos fantasmes.

Cependant même si le concept de robot date de plusieurs siècles, le terme robot lui fut inventé par le tchèque Karel Capek dans une pièce de théâtre écrite en 1920 : "RUR ou les robots universels de Rossum". Ce terme est dérivé du verbe tchèque "robota" signifiant travail forcé ou corvée. [1]

Mais, en fait, ce ne fut que vers la fin des années 1970 que les robots industriels de première génération ont vu le jour. Car, depuis fort longtemps, les réalisateurs d'automates ont cherché à pouvoir insuffler à leurs machines des comportements adaptés aux circonstances. Malheureusement, jusqu'au vingtième siècle, les techniques étaient trop primitives pour permettre de telles réalisations. Il a fallu attendre jusqu'en 1959 pour que Georges Devol invente une machine originale, polyvalente et reprogrammable, ce qui a permis au robot d'acquérir une réalité industrielle. [1]

La première problématique en robotique se réduit dans la plupart des cas à faire se déplacer un robot dans un environnement connu ou inconnu, tout en évitant d'éventuels obstacles fixes ou mobiles, pour réaliser une tâche telle qu'il en découle qu'il faut pouvoir définir une stratégie de mouvement (planification) [2]. Dans cette suite logique La deuxième problématique se réduit à la commande des robots qui permettra à ces derniers de suivre avec précision la trajectoire imposée.

L'objectif de ce mémoire trouve son intérêt dans ces deux problématiques à savoir la planification et le développement de lois de commande pour le suivi de trajectoire. Nous nous intéresserons au robot puma560 pour notre application. Une série de simulations selon les différentes lois de commande, sont faites pour conclure sur les performances de chacune d'elle.

Ce manuscrit se divise en quatre chapitres :

- Dans le premier, nous présentons des généralités sur les bras de robots, au travers de la description des différents constituants technologiques d'un système robotique articulé et les définitions des principaux termes du domaine.
- Le chapitre deux est consacré à la modélisation géométrique, cinématique et dynamique des bras de robots.
- Le chapitre trois présente les méthodes de planification de trajectoires dans les deux espaces articulaire et opérationnel pour robot PUMA560.
- Dans le quatrième chapitre, nous développons les schémas de commande linéaire et non linéaire adaptés au contrôle en position et en vitesse du robot PUMA560, que ce soit dans l'espace de tâche ou dans l'espace articulaire.

Chapitre I :

Généralités sur les

bras de robots

I.1 Introduction

De la fabrication à l'automobile en passant par l'agriculture, les bras robotiques industriels sont l'un des types de robots les plus couramment utilisés aujourd'hui. Les bras robotiques, également connus sous le nom de bras robotiques articulés, sont rapides, fiables et précis et peuvent être programmés pour effectuer un nombre infini de tâches dans une variété d'environnements. Ils sont utilisés dans les usines pour automatiser l'exécution de tâches répétitives, telles que l'application de peinture sur des équipements ou des pièces, dans les entrepôts pour ramasser sélectionner ou trier les marchandises des convoyeurs de distribution, ou dans un champ agricole pour cueillir et placer des fruits mûrs sur des plateaux de stockage, afin de répondre aux commandes des consommateurs. Et à mesure que les technologies robotiques se développent et que les environnements industriels deviennent plus connectés, les capacités des bras robotiques s'étendent pour permettre de nouveaux cas d'utilisation et modèles d'opérations commerciales.

I.2 Définitions préliminaires

I.2.1 Robotique

La robotique est la science et la technologie des robots, ainsi que leur conception, leur fabrication, leur application et leur disposition structurelle. Cela nécessite une connaissance pratique de l'électronique, de la mécanique et des logiciels.[3]

I.2.2 Robot

Un robot est un dispositif mécatronique qui allie la mécanique, l'électronique et l'informatique. Il a pour but d'accomplir automatiquement soit des tâches qui sont généralement dangereuses, pénibles, répétitives ou impossibles pour les humains, soit des tâches plus simples, mais en les réalisant mieux que ce que ferait un être humain.[3]

I.2.3 Articulation

Une articulation lie deux corps successifs en limitant le nombre de degrés de liberté, de l'un par rapport à l'autre [4].

Il existe principalement deux types d'articulation :

❖ 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 [4].

❖ 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 [4]

I.2.4 Degré de liberté

C'est le nombre de directions indépendantes ou d'articulations du robot, qui permettraient au robot de déplacer son organe terminal à travers la séquence de mouvements requise.[5]

Pour un positionnement arbitraire, 6 degrés de liberté sont nécessaires : 3 pour la position (gauche-droite, avant-arrière et haut-bas) et 3 pour l'orientation (lacet, tangage et roulis).[5]

I.2.5 Espace opérationnel

L'espace des tâches (ou espace cartésien) est défini par la position et l'orientation de l'effecteur terminal d'un robot.

I.2.6 Espace articulaire

L'espace articulaire est défini par un vecteur dont les composantes sont les déplacements en translation et angulaire de chaque articulation d'une liaison robotique.

I.3 Classification des Robots

Si les applications robotiques sont très variées (donner des instructions, approvisionner les rayons, souder du métal dans des environnements dangereux, etc.), les robots d'aujourd'hui peuvent généralement être regroupés en six catégories [6].

I.3.1 Robots mobiles autonomes (AMR)

Un robot mobile autonome est un type de robot qui peut se déplacer dans son environnement de manière indépendante.



Figure I.1 Robot MIR25

I.3.2 Véhicules à guidage automatique (AGV)

Les AGV s'appuient sur des pistes ou des chemins prédéfinis et nécessitent souvent la surveillance d'un opérateur. Ils sont généralement utilisés pour livrer des matériaux et déplacer des articles dans des environnements contrôlés tels que des entrepôts et des usines [6].

I.3.3 Humanoïdes

Ce sont les robots qui remplissent des fonctions centrées sur l'homme et prennent souvent des formes semblables à celles de l'homme. Ils utilisent bon nombre des mêmes composants technologiques que les AMR pour détecter, planifier et agir dans le cadre de tâches telles que l'indication d'un itinéraire ou la prestation de services de conciergerie [6].

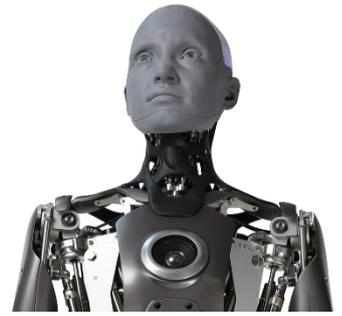


Figure I.2 humanoïde Ameca

I.3.4 Cobots

Est un raccourci pour robot collaboratif. Cobot fait référence à un type de robot qui a été créé pour travailler aux côtés d'employés humains. Alors que certains robots sont destinés à remplacer les tâches traditionnellement effectuées par les humains, les cobots sont conçus pour aider les humains dans les tâches en milieu de travail.



Figure I.3 Spot le chien robot

I.3.5 Robots articules ou industriel

Selon British Robots Association « Un robot industriel est un appareil reprogrammable conçu pour manipuler et transporter des pièces, des outils ou des outils de fabrication spécialisés au moyen de mouvements programmés variables afin d'exécuter des tâches de fabrication spécifiques ».



Figure I.4 Bras manipulateur à 6 ddl

- Dans ce travaille on se contentera d'étudier ce type de robot.

I.3.6 Robot Hybrides

Les différents types de robots sont souvent combinés pour créer des solutions hybrides capables de réaliser des tâches plus complexes. Ainsi, à mesure que les fonctionnalités sont combinées dans des solutions uniques, les capacités de calcul sont également consolidées.[6]



Figure I.5 Robot Mobile équipé d'un bras

I.4 Éléments constitutifs des robots manipulateurs

Les robots manipulateurs sont composés de liaisons rigides interconnectées par le moyen d'articulations, et un organe effecteur se trouvant à l'extrémité de la dernière liaison. Le mouvement de ces liaisons est assuré par des actionneurs et l'état du manipulateur est donné par des mesures issues des capteurs.

Le mouvement désiré du manipulateur est achevé en utilisant un système de contrôle qui fournit des commandes aux actionneurs des articulations dépendant sur la méthodologie de commande implémentée. La figure I.7 suivante illustre les différents éléments qui constituent un robot manipulateur :

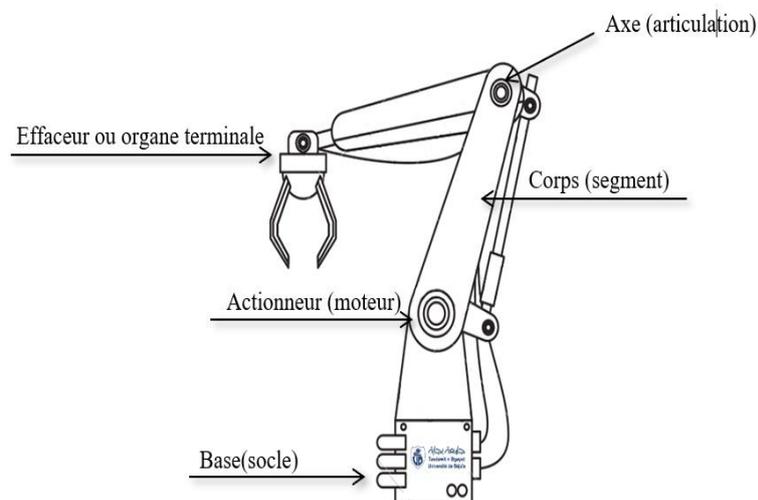


Figure I.6 Élément constitutif des robots manipulateurs

Il convient d'appeler les trois premiers ddl d'un robot (à partir de la base du robot) le **porteur** du robot. Les ddl résiduels forment le **poignet**, caractérisé par des dimensions beaucoup plus petites et une plus faible masse. Ce dernier il est destiné à l'orientation de l'organe terminal.

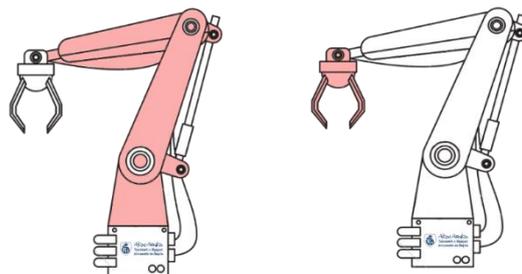


Figure I.7 Représentations du porteur et poignet

Dans son ensemble, on distingue généralement quatre éléments principaux dans un bras de robots.

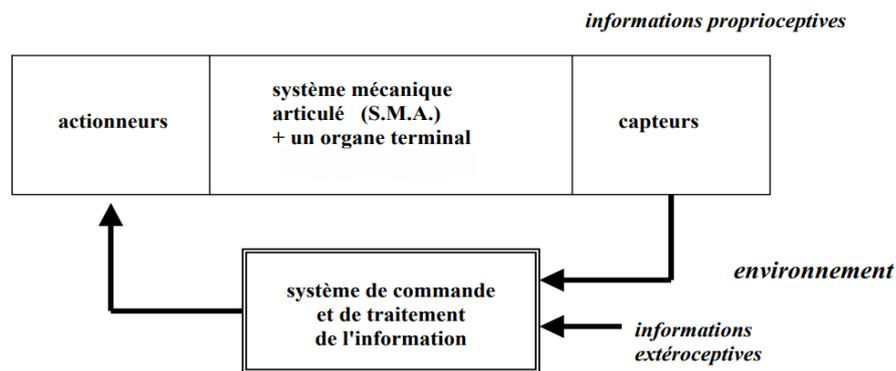


Figure I.8 Chaîne d'asservissement pour un robot manipulateur

I.4.1 Système mécanique articulé

Le SMA est un mécanisme ayant une structure plus ou moins proche de celle du bras humain. Son rôle est d'acheminer l'organe terminal dans une pose (position et orientation) donnée, selon des caractéristiques de vitesse et d'accélération données.

Sa motorisation est réalisée par des actionneurs électriques, pneumatiques ou hydrauliques qui transmettent leurs mouvements aux articulations par des systèmes appropriés.

Son architecture est une chaîne cinématique de corps, généralement rigides (ou supposés comme tels), assemblés par des liaisons appelées articulations.

❖ Il en existe trois types de chaîne :

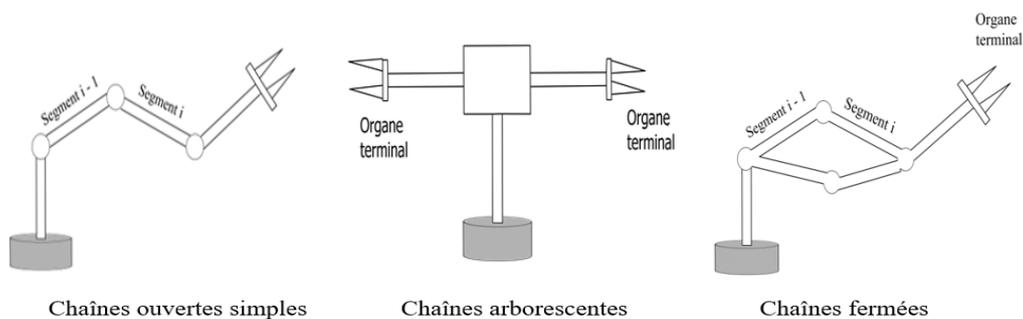


Figure I.9 différents types de Chaînes

I.4.2 Organe terminal

Il s'agit d'une interface permettant au robot d'interagir avec son environnement. Un organe terminal peut être monofonctionnel ou multifonctionnel, dans le sens où il peut être équipé de plusieurs dispositifs ayant des fonctionnalités différentes.

I.4.3 Système de traitement

C'est lui qui gère l'ensemble des tâches. Il admet trois rôles essentiels : [7]

1. Le rôle de l'information, qui consiste à collecter l'information venant des capteurs.
2. Le rôle de la décision : en partant d'une tâche définie et en tenant compte des données du système et de l'environnement, il établit les actions adéquates.
3. Le rôle de la communication.

I.4.4 Actionneurs

Un actionneur est la partie du robot lui permettant d'agir sur son environnement. C'est grâce à lui que le robot peut exécuter un travail. On peut aussi l'appeler "partie opérative".

Plusieurs technologies existent. On rencontre des actionneurs électriques, pneumatiques et hydrauliques.

I.4.4.1 Actionneurs électriques

Les robots de petite et moyenne taille sont généralement alimentés par des actionneurs électriques via des trains d'engrenages utilisant des servomoteurs et des moteurs pas à pas. Les moteurs à courant continu sont les plus couramment utilisés.

La puissance en général modeste de ces machines fait 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.

I.4.4.2 Actionneurs pneumatiques

Généralement utilisé pour les robots plus petits qui possèdent moins de degrés de liberté (2 à 4 mouvements articulaires). Ils sont limités aux tâches de sélection avec des cycles rapides. Le système d'entraînement pneumatique peut être appliqué à l'actionnement de dispositifs à piston pour fournir des mouvements linéaires. Les mouvements de rotation peuvent être obtenus par des actionneurs rotatifs [1].

I.4.4.3 Actionneurs hydrauliques :

Les plus gros robots sont généralement alimentés par des actionneurs hydrauliques. Le système d'entraînement hydraulique peut fournir un mouvement de rotation (actionneurs à palettes rotatives) et un mouvement linéaire (pistons hydrauliques). [1]

I.4.5 Transmetteurs

Les transmetteurs sont des mécanismes de transformation de mouvement qui peuvent permettre de passer d'un mouvement de rotation à un mouvement de rotation ou d'un mouvement de rotation à un mouvement de translation. Ils permettent ainsi d'adapter la vitesse de sortie et les efforts de sortie entre l'actionneur et l'effecteur. Bien évidemment il en existe une multitude de variété de transmetteur.

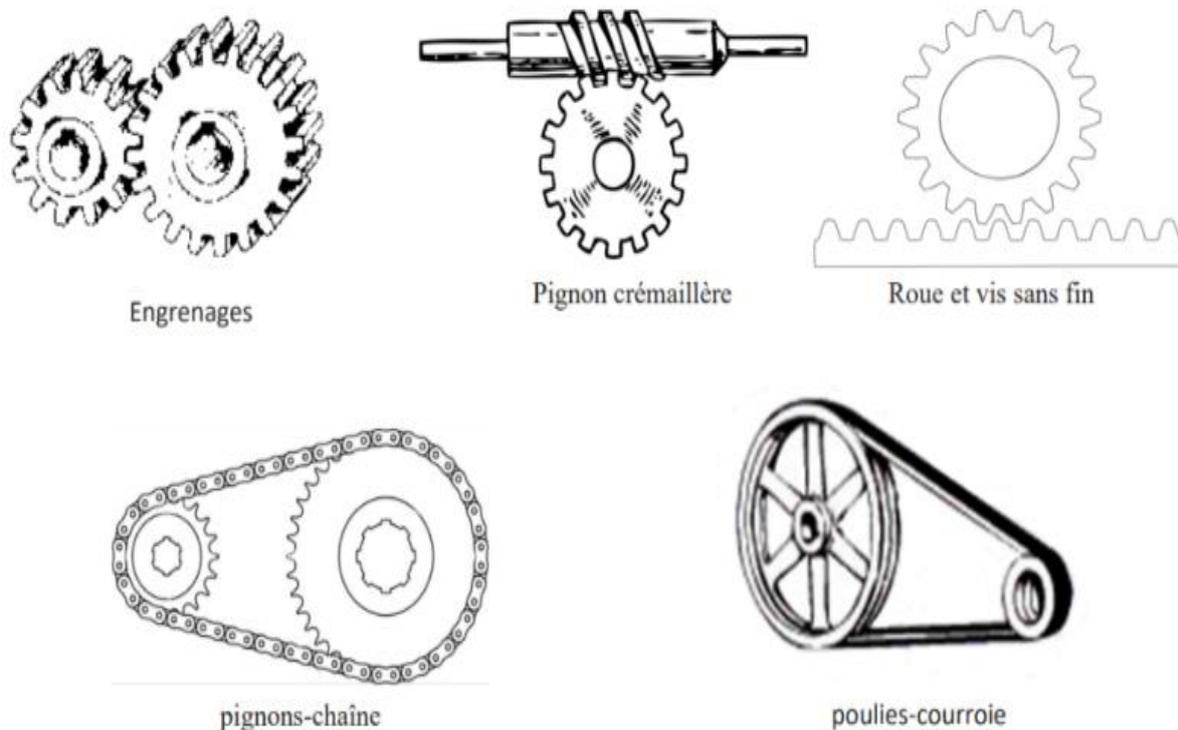


Figure I.10 différents types de Transmission d'un bras du robot

I.4.6 Capteurs :

Les capteurs et les transducteurs (qui impliquent la conversion d'énergie d'une forme à une autre et les données captées sont traitées par un contrôleur) sont des outils très importants dans la conception d'un robot. Grâce à eux, le robot peut percevoir la dimension physique de son environnement afin d'effectuer certains mouvements ou actions et prendre des mesures exactes. Un capteur est utilisé pour modifier une gamme de courants mécaniques, magnétiques, thermiques, chimiques et électriques. Les capteurs fournissent l'équivalent des yeux, des oreilles, du nez, et de la langue d'une machine et sont considérés comme le cerveau du système d'automatisation dans les usines [11].

I.5 Morphologie des bras manipulateurs :

Selon l'architecture du porteur, les robots manipulateurs peuvent être classés en différentes catégories :

I.5.1 Structure cartésienne (PPP)

Les robots cartésiens effectuent uniquement des déplacements sur des axes linéaires, ce qui les rend très rentables et faciles à programmer. Ils sont parfaitement adaptés pour effectuer des tâches répétitives, comme par exemple décharger des machines [8].

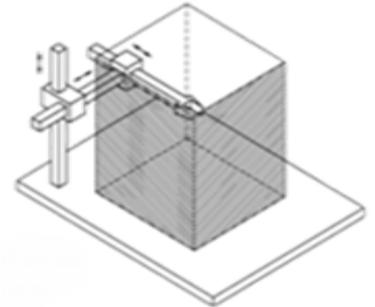


Figure I.11 Manipulator Cartesian

I.5.2 Structure cylindrique

Le porteur possède une articulation rotoïde et deux articulations prismatiques, ou deux articulations rotoïdes et une articulation prismatique, qui permettent d'effectuer des translations et des rotations selon des axes de mouvement définis par un système de coordonnées cylindriques, donnant ainsi la forme géométrique d'un cylindre à l'enveloppe de travail [9].

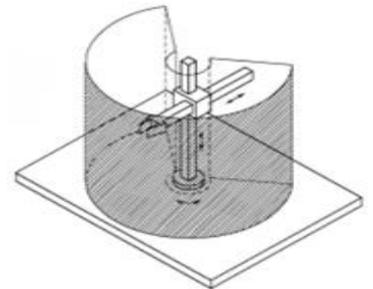


Figure I.12 Manipulator Cylindrique

I.5.3 Structure sphériques (polaires)

Le porteur possède deux articulations rotoïdes consécutives suivies d'une articulation prismatique. Les axes de mouvement des deux articulations rotoïdes étant concourants et l'ensemble des articulations permettant d'effectuer des rotations et des translations selon des axes de mouvement définis par un système de coordonnées sphériques, donnant ainsi à l'enveloppe de travail la forme d'une sphère [10].

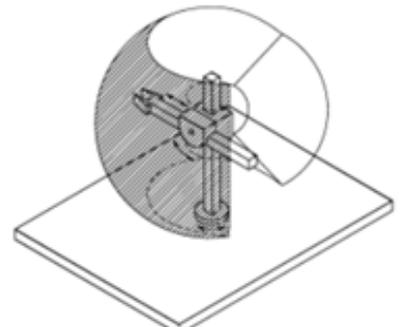


Figure I.13 Manipulateur sphérique

I.5.4 Structure anthropomorphe

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 correspondent respectivement au tronc (base), à l'épaule et au coude d'un être humain [11]

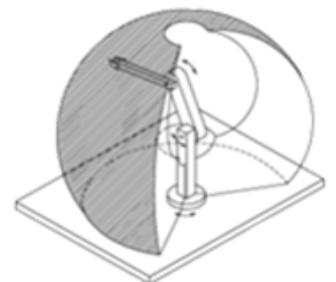


Figure I.14 Manipulateur Anthropomorphe

I.5.5 Structure SCARA

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 ddl). Cette structure est utilisée dans des opérations d'assemblage [11].

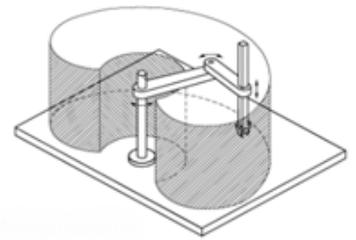


Figure I.15 Structure SCARA

I.6 Différentes générations de robots

Le monde industriel a vu naître plusieurs générations de robots, en fonction de l'évolution de la technologie et donc de l'intégration de ces nouvelles techniques dans les robots.

I.6.1 Les robots de première génération

Étaient destinés à des tâches simples, répétitives et lourdes, souvent dans un environnement hostile (manutention en milieu nucléaire etc...). Ce ne sont que des machines-outils à commande numérique très évoluées. Ces robots s'adaptent avec les modifications dans l'environnement, mais leur précision est faible.

I.6.2 Les robots de deuxième génération

Sont capables de reconstituer et d'exécuter des trajectoires et des tâches complexes par rapport aux précédentes (usinage, assemblage). Ils nécessitent donc l'utilisation de capteurs proprioceptifs. Par conséquent, ils possèdent un fonctionnement en boucle fermée au niveau de la structure interne mais en boucle ouverte au niveau de la tâche. Ce sont de véritables automates industriels.

I.6.3 Les robots de troisième génération

Ne sont pas fondamentalement différents des précédents, mais on enregistre le franchissement d'un pas supplémentaire dans le sens de l'amélioration de l'automatisme et la versatilité. Le concept de versatilité englobe deux notions : celle d'adaptativité et celle de polyvalence.

I.7 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 maximale transportable (de quelques kilos à quelques tonnes), à déterminer dans les conditions les plus défavorables (c'est-à-dire, en élongation maximale)
- L'architecture du S.M.A. : le choix est guidé par la tâche à réaliser. Par exemple, les robots à structure rigide vs. Robots avec segments et/ou articulations flexibles
- Le volume ou espace de travail ("workspace" en anglais), défini comme l'ensemble des points atteignables par l'organe terminal du robot

I.8 Domaines d'application

Les bras robotiques affectent nos vies à grande échelle. L'usage de la robotique se multiplie dans tous les secteurs importants de la vie et de la société. Quelques exemples notables sont énumérés ci-dessous :

I.8.1 L'industrie

Les robots manipulateurs d'usine ont pris en charge la plupart des tâches qui exigent un niveau de précision plus élevé, une vitesse supérieure et de la patience. Les usines automobiles utilisent des bras de robots pour couper et assembler des pièces. Dans les chaînes d'assemblage, on retrouve des robots soudeurs, manipulateurs, peintres etc.



Figure I.16 Ligne d'assemblage de robot dans l'usine de voiture

I.8.2 L'agriculture

La robotique permet désormais aux agriculteurs d'être plus efficaces et plus productifs dans la gestion des cultures, car ils peuvent effectuer des tâches humaines à répétition de manière efficace et plus efficiente.



Figure I.17 robot Agricole

I.8.3 L'Armée

Robots manipulateurs est de plus en plus utilisée par les militaires pour les missions de recherche et de sauvetage.



Figure I.18 robot militaire

I.8.4 La médecine

Dans le domaine médical, un robot manipulateur peut être utilisé pour effectuer des opérations trop délicates pour les mains d'un chirurgien ou comme aide lors d'opérations régulières telles que les pontages coronariens. On parle de chirurgie assistée (mot né de l'anglais « surgery » : chirurgie) c'est-à-dire tout ce qui consiste à introduire les derniers outils des technologies informatiques et robotiques dans la pratique médico-chirurgicale. Cette pratique de « chirurgie assistée » est émergente donc bien que peu répandue, elle est en phase de devenir la chirurgie du futur.



Figure I.19 robots chirurgicaux

I.8.5 Autres Usages

- Les scientifiques envoient des robots pour explorer les surfaces de la lune ou de planètes comme Mars, tandis que d'autres robots vont dans l'espace pour réparer des équipements spatiaux.
- La robotique domestique met également à la disposition de tout un chacun des appareils d'aide aux ménages, de la surveillance...

I.9 Conclusion

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

Chapitre II :

Modélisation des bras

de robots

II.1 introduction

La complexité croissante des tâches assignées aux robots manipulateurs implique une étude mathématique plus rigoureuse des différents mouvements lors de la conception. Il est alors possible de dimensionner les actionneurs en fonction des couples maximums requis aux articulations. Ces derniers sont fonction des performances prédéfinies : vitesse de déplacement et charge, ainsi que de la constitution du bras : inertie, masse, frottement.

L'étude de la morphologie d'un bras permet une définition rapide de l'encombrement spatial et des éventuelles collisions lors d'un déplacement programmé.

Plusieurs types de modèles sont nécessaires pour décrire le comportement du robot, tel que :

Le modèle géométrique, le modèle cinématique et le modèle dynamique.

- ❖ Le modèle géométrique direct et inverse exprime la situation de l'organe terminal en fonction des variables articulaires du mécanisme et inversement ;
- ❖ Le modèles cinématique directe et inverse exprime la vitesse de l'organe terminal en fonction des vitesses articulaires et inversement ;
- ❖ Le modèle dynamique implique les vitesses, accélérations et les forces généralisées.

II.2 Modélisation géométrique

II.2.1 Description de la position et de l'orientation

Dans l'étude de la robotique, la localisation des objets dans l'espace tridimensionnel est fondamentale et nécessaire. Ces objets sont décrits par seulement deux attributs la position et l'orientation qui dépendent de la représentation des équations mathématiques. Pour le développement des équations du robot, il est nécessaire d'établir plusieurs systèmes de coordonnées pour représenter les positions et les orientations de corps rigides. Il est également nécessaire de connaître les transformations des coordonnées entre ces systèmes, afin que les vecteurs représentant des positions, des vitesses et des accélérations, donnés dans un système de coordonnées donné, puissent être représentés dans d'autres systèmes des coordonnées [11].

Le problème fondamental est la formulation des équations permettant d'exprimer :

- Les coordonnées opérationnelles à partir des coordonnées articulaires. On établit alors le modèle géométrique.
- Les coordonnées articulaires à partir des coordonner opérationnelles. On parle alors du modelé géométrique inverse.

❖ L'espace articulaire

Il est décrit par des coordonnées articulaires comme les angles et les distances en fonction des liaisons mécaniques liant les segments du robot [13].

$$\theta = [\theta_1, \theta_2 \dots, \theta_m]^T \quad (\text{II.1})$$

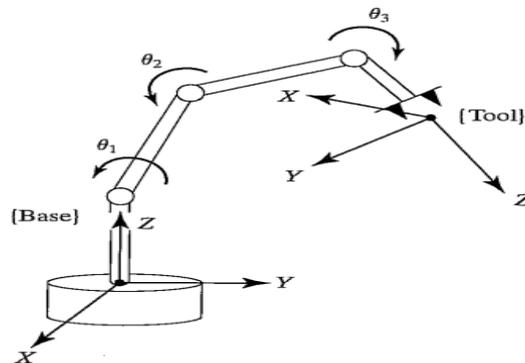


Figure II.1 Représentation de l'espace articulaire

❖ L'espace opérationnelle

Il est décrit par les positions et les orientations de l'effecteur dans l'espace tâche [11].

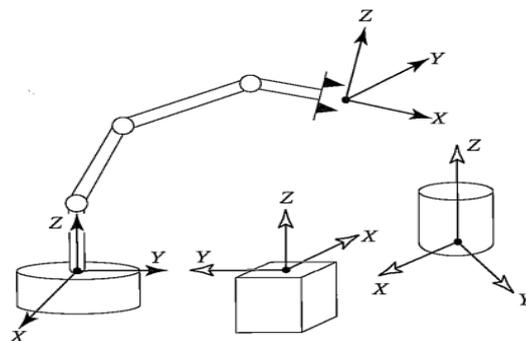


Figure II.2 Représentation de L'espace opérationnelle

II.2.2 Coordonnées homogènes d'un vecteur

Chaque liaison d'un manipulateur fait des rotations ou des translations par rapport au référentiel d'inertie fixe (par exemple un repère fixé à la base du robot). Le calcul des coordonnées des liaisons du manipulateur exprimées dans le référentiel de base est relativement difficile. Cette difficulté augmente suivant l'ordre de la liaison (numéro de la liaison) jusqu'à l'élément terminal. Pour ne pas alourdir les calculs et ramener toutes les informations géométriques au repère d'inertie de la base, il est judicieux de les localiser à leurs articulations correspondantes et situer chaque liaison à son propre référentiel.

Le passage d'un référentiel à un autre est garanti par des transformations. Lorsqu'on a uniquement des rotations, on se satisfait à une matrice de transformation R de troisième ordre. En présence de translations, il est plus judicieux d'utiliser des matrices de quatrième ordre pour exprimer, à la fois, la position et l'orientation d'une manière compacte, avec les matrices de transformations homogènes. Dans ce cas, le vecteur de position p sera augmenté par une quatrième composante [12].

II.2.3 Position et orientation d'un corps rigide

II.2.3.1 Représentation d'un point

Considérons un repère fixe R, d'axes X Y Z et un corps rigide dans l'espace représenté par un point P. La position du point P est décrite dans le repère fixe par le vecteur colonne 3x1 [18].

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

Le vecteur homogène correspondant du point p est :

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

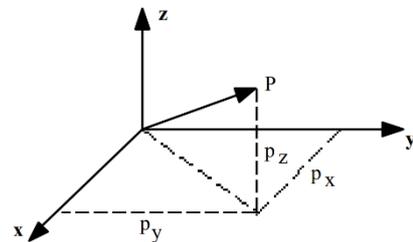


Figure II.3 Représentation d'un point

w : est un facteur d'échelle, égale à 1 en robotique, ce qui donne :

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

II.2.3.2 Position et orientation du corps rigide

On associe au corps rigide un repère orthonormé, R_p, d'axes (X_p, Y_p, Z_p) centré au point O_p, qui permet de définir sa position et son orientation par rapport au repère fixe R₀. La rotation du corps rigide est décrite par les neuf cosinus directeurs qui forment la matrice de rotation R [18].

$$R = \begin{bmatrix} n \cdot i & o \cdot i & a \cdot i \\ n \cdot j & o \cdot j & a \cdot j \\ n \cdot k & o \cdot k & a \cdot k \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}$$

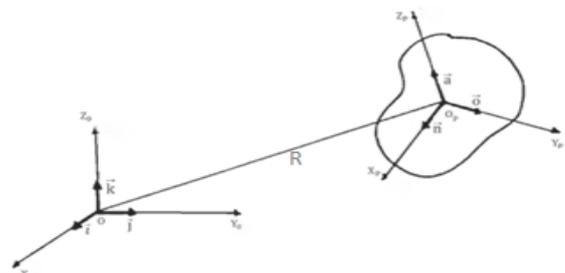


Figure II.4 position et orientation du corps rigide

II.2.4 Transformation homogène

- La matrice augmentée de transformation a la forme suivante [11].

$$\mathbf{T} = \begin{bmatrix} \mathbf{R}_{3 \times 3} & \mathbf{P}_{3 \times 1} \\ \mathbf{N}_{1 \times 1} & \mathbf{1} \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & P_X \\ R_{21} & R_{22} & R_{23} & P_Y \\ R_{31} & R_{32} & R_{33} & P_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Avec :

- $\mathbf{N} = (\mathbf{0}, \mathbf{0}, \mathbf{0})$: représente la transformation de perspective (utilisée en vision).
- $\mathbf{P}(\mathbf{3x1})$: est le vecteur de position.
- $\mathbf{R}(\mathbf{3x3})$: est la matrice de rotation.
- $\mathbf{1}$: est le facteur d'échelle.
- \mathbf{T} : est la matrice de transformation.

II.2.5 Transformation des repères

Pour transformer les coordonnées (translation et/ou rotation) d'un repère R_i et l'amener dans un autre repère R_j , on fait appel à une matrice de transformation homogène ou bien matrice de passage homogène notée T_j^i .

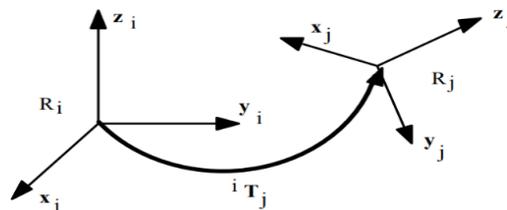


Figure II.5 Représentation d'un Transformation des repères

II.2.6 Matrices de rotations pures autour des axes principaux

Les matrices de rotation du repère R_j , autour des axes (OX, OY, OZ) du repère R_i [15] sont données par :

- Rotation autour axe X

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

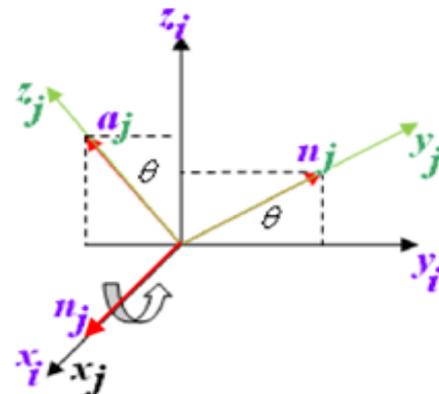


Figure II.6 Rotation autour l'axe X [15]

- **Rotation autour axe Y**

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

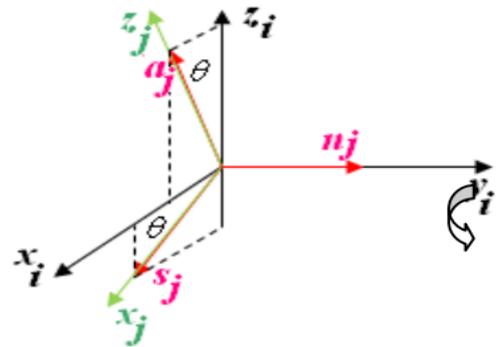


Figure II.7 Rotation autour l'axe Y [15]

- **Rotation autour axe Z**

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

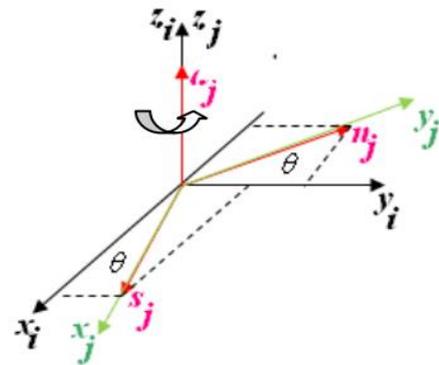


Figure II.8 Rotation autour l'axe Z [15]

II.2.7 Matrice de translations pures

La matrice de translation de l'origine de R_j des distances (a, b, c) le long des axes

(OX, OY, OZ) du repère R_i est donnée par :

$$T_j^i = \text{Trans}(a, b, c) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

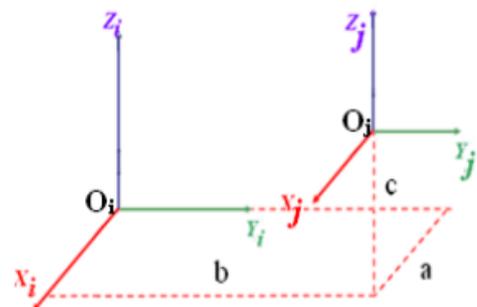


Figure II.9 Translations pures [15]

II.2.8 Modèle géométrique direct

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 T_0^n : [11]

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

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

$$\mathbf{X}(t) = \mathbf{f}[\boldsymbol{\theta}(t)] \quad (\text{II.3})$$

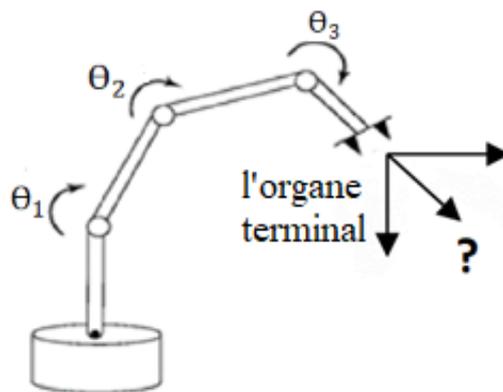


Figure II.10 Représentation du MGD

- $\mathbf{X} = [x_1, x_2, \dots, x_n]^T$: vecteur défini dans l'espace opératoire et sa dimension n dépend de la tâche à réaliser.
- $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_m]^T$: vecteur défini dans l'espace articulaire et sa dimension m est égale au nombre de *ddl* du robot.

Pour réaliser une tâche, il faut : $m \geq n$. Mais, quand $m > n$, on a une redondance qui pose quelques problèmes de calcul.

II.2.8.1 Méthodes de Représentation des Robots

La modélisation des robots de façon systématique exige une méthode adéquate pour la description de leur structure. Plusieurs méthodes et notations ont été proposées, telles que : la convention de Denavit-Hartenberg (DH) [12].

II.2.8.1.1 Convention DH

La convention de Denavit et Hartenberg (DH 1955) est une méthode systématique. Elle permet le passage entre articulations adjacentes d'un système robotique. Elle concerne les chaînes cinématiques ouvertes où l'articulation possède uniquement un degré de liberté et les surfaces adjacentes restent en contact. Le choix adéquat des repères dans les liaisons facilite le calcul des matrices homogènes de DH et permet d'arriver à exprimer rapidement des informations de l'élément terminal vers la base ou l'inverse [12].

➤ Pour passer de R_{i-1} à R_i , on peut suivre le chemin suivant :

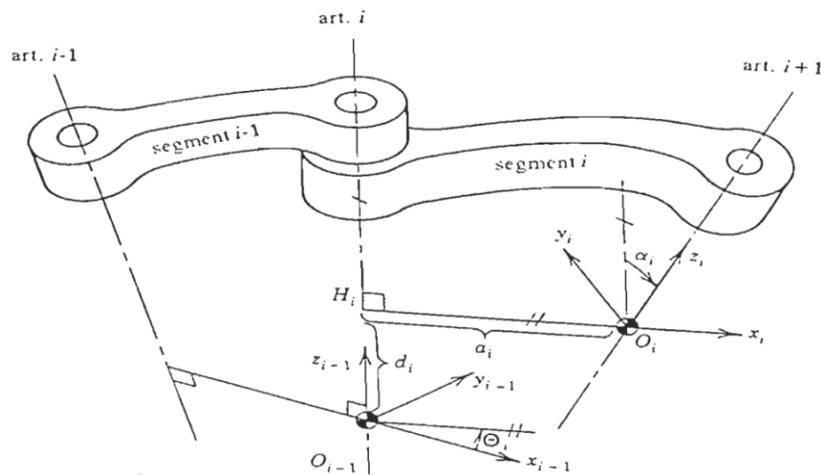


Figure II.11 Notation de Denavit-Hartenberg pour une paire de segments adjacents [9]

- 1- Rotation (Z_{i-1}, θ_i) avec θ_i : c'est l'angle entre l'axe X_{i-1} , et X_i , autour de Z_{i-1} .
- 2- Translation (Z_{i-1}, d_i) d_i : c'est la distance entre l'origine O_{i-1} à et l'intersection de l'axe Z_{i-1} avec le prolongement de l'axe X_i , elle est mesurée le long de Z_{i-1}
- 3- Translation (X_i, a_i) a_i : il décrit la distance entre les axes Z_{i-1} et Z_i , mesurée le long de l'axe X_i et correspondant à la longueur du segment i
- 4- Rotation (X_i, α_i) α_i : c'est l'angle que fait l'axe Z_{i-1} avec Z_i , autour de X_i .

Alors, la matrice de passage d'une articulation à la suivante est :

$$T = T_{i-1}^i = T_{z\theta_i} \cdot T_{zd_i} T_{xa_i} T_{x\alpha_i} = T_z \cdot T_x = \begin{bmatrix} c\theta_i & -c\alpha_i s\theta_i & s\alpha_i s\theta_i & a_i \cdot c\theta_i \\ s\theta_i & c\alpha_i c\theta_i & -s\alpha_i c\theta_i & a_i \cdot s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (II.4)$$

Le MGD d'un bras de robot est obtenu en utilisant le produit :

$$T_{MGD} = T_0^n = T_0^1 T_1^2 \dots T_{n-1}^n \quad (II.5)$$

II.2.8.2 Description du robot puma 560

Le robot PUMA 560 est un bras robotisé à 6 axes, il offre une charge utile de 2,5 kg et une portée de 864 mm. La répétabilité du robot PUMA 560 est de 0,1 mm. Les applications courantes du robot PUMA 560 incluent : le soudage à l'arc, la distribution, le tcp à distance et le soudage par points. Le PUMA 560 est fabriqué par l'entreprise qui porte le même nom PUMA.

II.2.8.3 Modèle géométrique du robot PUMA 560

Dans le présent travail, la convention D-H est adoptée pour l'établissement du MGD du robot PUMA 560. La Figure II.11 représente la disposition des repères associés aux différentes articulations.

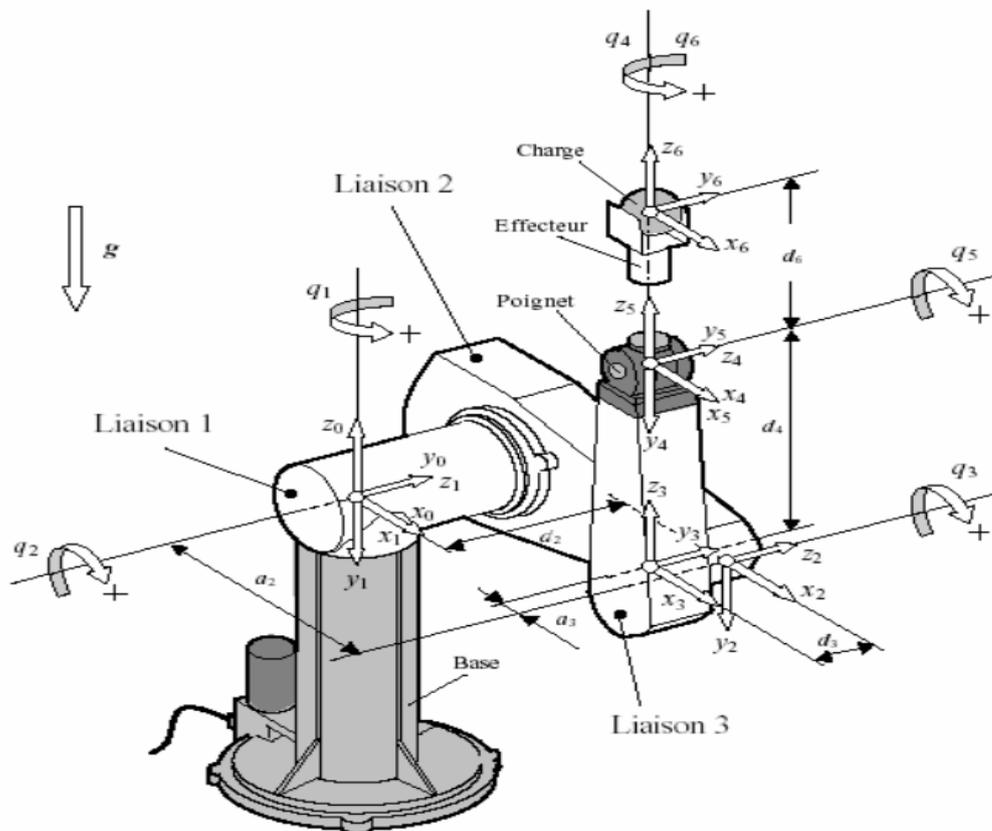


Figure II.12 Association des repères et les paramètres DH pour La représentation du bras PUMA 560 [12].

En respectant la position de l'origine du repère de base, on peut établir les paramètres DH.

Tableau II-1 Paramètres de DH du bras de Robot PUMA 560

i	α_i	θ_i	a_i	d_i
Numéro de la liaison	Degrés	Variable	Mètres	Mètres
1	-90	θ_1	0	0
2	0	θ_2	0.4318	0.2435
3	+90	θ_3	-0.0203	-0.0934
4	-90	θ_4	0	0.4331
5	+90	θ_5	0	0
6	0	θ_6	0	-

Dans ce travail, la dynamique des trois dernières articulations est considérée comme négligeable par rapport à celle des trois premières. Par conséquent, nous nous sommes intéressés à étudier le mouvement des trois premières articulations du bras de robot PUMA 560, en fixant le poignet à la position d'origine ($\theta_4=0$, $\theta_5=0$ et $\theta_6=0$).

En utilisant les trois premières lignes de la table des paramètres, on peut établir les matrices de transformations homogènes permettant de passer du repère R_3 à R_0 :

$$T_0^1 = \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.6})$$

$$T_1^2 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2 c_2 \\ s_1 c_2 & c_2 & 0 & a_2 c_2 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.7})$$

$$T_2^3 = \begin{bmatrix} c_3 & 0 & -s_3 & a_2 c_2 \\ s_3 & 0 & c_3 & a_2 c_2 \\ 0 & 1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.8})$$

$$T_{MGD} = T_0^3 = T_0^1 T_1^2 T_2^3 = \begin{bmatrix} c_1 c_{23} & -s_1 & c_1 s_{23} & c_1 (a_2 c_2 + a_3 c_{23}) - (d_2 + d_3) s_1 \\ s_1 c_{23} & c_1 & s_1 s_{23} & s_1 (a_2 c_2 + a_3 c_{23}) + (d_2 + d_3) c_1 \\ -s_{23} & 0 & c_{23} & -(a_2 s_2 + a_3 s_{23}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.9})$$

Avec $C_i = \cos \theta_i$; $S_i = \sin \theta_i$; $C_{ij} = \cos(\theta_i + \theta_j)$; $S_{ij} = \sin(\theta_i + \theta_j)$

II.2.9 Modèle Géométrique Inverse

Le modèle géométrique direct d'un robot permet de calculer les coordonnées opérationnelles donnant la situation de l'organe terminal en fonction des coordonnées articulaires. Le problème inverse consiste à calculer les coordonnées articulaires correspondant à une situation donnée de l'organe terminal [11] . Le modèle s'écrit : $\Theta = f^{-1}(x)$

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

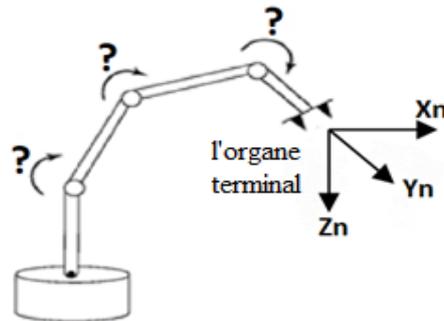


Figure II.13 Représentation du MGI

Il n'existe pas une méthode analytique générale pour trouver le MGI, mais un certain nombre de méthodes, adaptées à des classes de cinématiques particulières. Certains problèmes peuvent également être distingués MGI :

- leurs solutions ne sont pas simples et quelques fois impossibles.
- problème d'existence ou la non-existence d'un solution inverse définit l'espace de travail du robot.
- Le manque d'une solution signifie que le robot ne peut atteindre certaine situation, car elles se trouvent à l'extérieur de l'espace de travail du robot.

II.2.10 Model géométrique inverse, du robot PUMA560

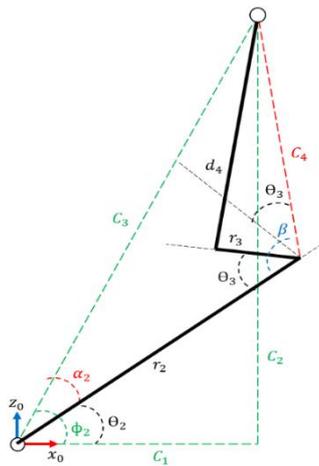


Figure II.15 plan $x_0 - z_0$ (vue latérale du PUMA560)

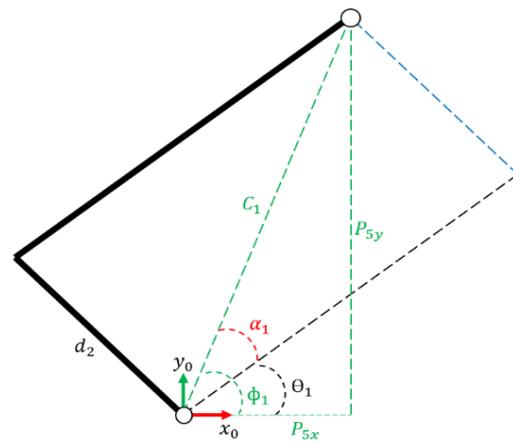


Figure II.14 plan $x_0 - y_0$ (vue de dessus du PUMA560)

À partir des figure II.14 et II.15 nous obtiendrons les résultats suivants :

$$\begin{aligned}
 \theta_1 &= \text{atan2}(P_{5y}, P_{5x}) - \text{atan2}(D_1, \sqrt{1 - D_1^2}). \\
 \theta_2 &= \text{atan2}(C_2, C_1) - \text{atan2}(\sqrt{1 - D_2^2}, D_2). \\
 \theta_3 &= \text{atan2}(\sqrt{1 - D_3^2}, D_3) - 90. \\
 \theta_4 &= \text{atan2}(T_6^3(2,3), T_6^3(1,3)) \\
 \theta_5 &= \text{atan2}(\sqrt{1 - T_6^3(3,3)^2}, T_6^3(3,3)) \\
 \theta_6 &= \text{atan2}(T_6^3(3,2) - T_6^3(3,1)).
 \end{aligned}
 \tag{II.10}$$

II.3 Modélisation cinématique

Le modèle cinématique directe d'un robot manipulateur décrit les vitesses des coordonnées opérationnelles en fonction des vitesses articulaires. Il est noté :[11]

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

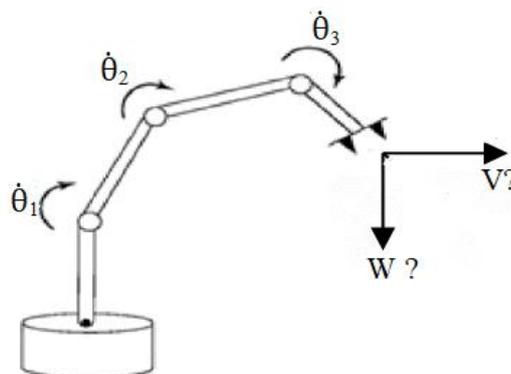


Figure II.16 Représentation du MCD

Avec :

$$\dot{\mathbf{X}} = \begin{bmatrix} v_n \\ w_n \end{bmatrix}, \dot{\Theta} = [\dot{\theta}_1, \dot{\theta}_2, \dots, \dot{\theta}_n]$$

v_n : vitesse de translation de l'organe terminal par rapport à R_0 .

w_n : vitesse de rotation de l'organe terminal par rapport à R_0 .

Où $\mathbf{J}(\Theta)$ désigne la matrice jacobienne de dimension $(\mathbf{m} \times \mathbf{n})$ du mécanisme, en fonction de la configuration articulaire Θ . La même matrice jacobienne 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 $d\Theta$, soit :

$$d\mathbf{X} = \mathbf{J}(\Theta) d\Theta \quad (\text{II.12})$$

II.3.1 Calcul de la matrice jacobienne par dérivation du MGD

Le calcul de la matrice jacobienne peut se faire par la dérivation du MGD $\mathbf{X} = \mathbf{f}(\Theta)$

à partir de la relation suivante [16] :

$$J_{ij} = \frac{\partial f_i(q)}{\partial q_j} \quad i = 1 \dots n; j = 1 \dots m ; \quad (\text{II.13})$$

Où J_{ij} est l'élément (i, j) de la matrice jacobienne \mathbf{J} .

Cette méthode est facile à mettre en œuvre pour des robots à deux ou trois degrés de liberté comme le montre l'exemple suivant. Matrice jacobienne par dérivation du MGD du bras PUMA 560.

On va étudier les des trois premières articulations (trois degrés de liberté) du bras de robot PUMA 560 représenté sur la figure (II- 11).

La notation simplificatrice r_3^6 signifie le vecteur des coordonnées homogènes de l'élément terminal exprimées dans le repère R_3 , sous l'hypothèse du poignet fixe à la position d'origine ($\theta_4 = 0, \theta_5 = 0, \theta_6 = 0$), le vecteur r_3^6 est donné par l'équation suivante :

$$r_3^6 = \begin{bmatrix} 0 \\ 0 \\ d_4 + d_6 \\ 1 \end{bmatrix} \quad (\text{II.14})$$

Par la transformation de DH, T_0^3 , on peut calculer les coordonnées de l'élément terminal exprimées dans le repère R_0 par :

$$P = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = T_0^3 r_3^6 \quad (\text{II.15})$$

Donc :

$$\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} c_1 c_{23} & -s_1 & c_1 s_{23} & c_1(a_2 c_2 + a_3 c_{23}) - (d_2 + d_3)s_1 \\ s_1 c_{23} & c_1 & s_1 s_{23} & s_1(a_2 c_2 + a_3 c_{23}) + (d_2 + d_3)c_1 \\ -s_{23} & 0 & c_{23} & -(a_2 s_2 + a_3 s_{23}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ d_4 + d_6 \\ 1 \end{bmatrix} \quad (\text{II.16})$$

Enfin, nous définissons le vecteur position de l'élément terminal exprimé dans le repère de base comme suit :

$$\begin{aligned} P_x &= c_1(a_2 c_2 + a_3 c_{23}) - (d_2 + d_3)s_1 + (d_4 + d_6)c_1 s_{23} \\ P_y &= s_1(a_2 c_2 + a_3 c_{23}) + (d_2 + d_3)c_1 + (d_4 + d_6)s_1 s_{23} \\ P_z &= -(a_2 s_2 + a_3 s_{23}) + (d_4 + d_6)c_{23} \end{aligned} \quad (\text{II.17})$$

La matrice jacobienne est calculée en dérivant ces trois relations par rapport à θ_1 , θ_2 et θ_3 :

$$J = \begin{bmatrix} j_{11} & j_{12} & j_{13} \\ j_{21} & j_{22} & j_{23} \\ j_{31} & j_{32} & j_{33} \end{bmatrix} \quad (\text{II.18})$$

Avec :

$$\begin{aligned} j_{11} &= -s_1(a_2 c_2 + a_3 c_{23}) - (d_2 + d_3)c_1 - (d_4 + d_6)s_1 s_{23} \\ j_{12} &= -c_1(a_2 c_2 + a_3 c_{23}) + (d_4 + d_6)c_1 c_{23} \\ j_{13} &= -a_3 c_1 s_{23} - (d_4 + d_6)c_1 s_{23} \\ j_{21} &= c_1(a_2 c_2 + a_3 c_{23}) - (d_2 + d_3)s_1 + (d_4 + d_6)c_1 c_{23} \\ j_{22} &= -s_1(a_2 c_2 + a_3 c_{23}) + (d_4 + d_6)s_1 c_{23} \\ j_{23} &= -a_3 s_1 s_{23} + (d_4 + d_6)s_1 c_{23} \\ j_{31} &= 0 \\ j_{32} &= -(a_2 c_2 + a_3 c_{23}) - (d_2 + d_3)s_{23} \\ j_{33} &= -a_3 c_{23} - (d_4 + d_6)s_{23} \end{aligned} \quad (\text{II.19})$$

II.3.2 Modèle cinématique inverse

L'objectif du MCI est de calculer, à partir d'une configuration donnée, les vitesses articulaires $\dot{\theta}$ qui assurent une vitesse opérationnelle \dot{X} imposée. Pour obtenir le modèle cinématique inverse, on inverse le modèle cinématique directe en résolvant un système d'équations. La mise en œuvre peut être faite de façon analytique ou numérique :

- La solution analytique a pour avantage de diminuer considérablement le nombre d'opérations. Mais, on doit traiter séparément tous les cas singuliers.
- Les méthodes numériques sont plus générales. La plus répandue étant fondée sur la notion de pseudo-inverse : les algorithmes traitent de façon unifiée les cas réguliers, singuliers et redondants. Elles nécessitent un temps de calcul relativement important [12].

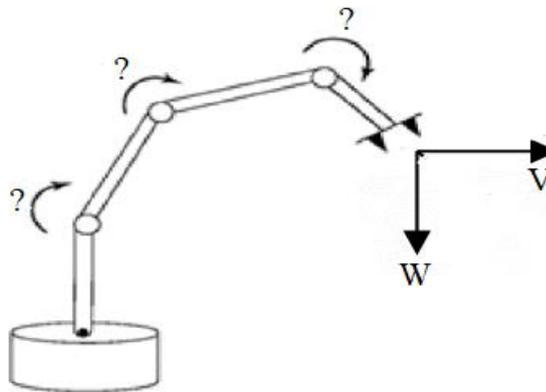


Figure II.17 Représentation du MCI

II.3.2.1 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 [11], les éléments X_p et X_r désignant respectivement la position et l'orientation opérationnelles du solide. Les relations entre les vitesses \dot{X}_p et \dot{X}_r et les vecteurs vitesses v_0^n et ω_0^n on du repère R_n sont telles que :

$$\begin{bmatrix} \dot{X}_p \\ \dot{X}_r \end{bmatrix} = \begin{bmatrix} \Omega_p & 0_3 \\ 0_3 & \Omega_r \end{bmatrix} \begin{bmatrix} V_n^0 \\ \Omega_n^0 \end{bmatrix} = \Omega \begin{bmatrix} V_n^0 \\ \Omega_n^0 \end{bmatrix} \quad (\text{II.20})$$

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

$$\dot{\theta} = \mathbf{J}^{-1} \dot{X} \quad (\text{II.21})$$

Dans le cas régulier où la matrice jacobéenne \mathbf{J} est carrée d'ordre n et son déterminant est non nul, la méthode la plus générale consiste matrice inverse de \mathbf{J} .

II.4 Le modèle dynamique

La modélisation dynamique d'un robot est l'opération permettant de construire un modèle de son évolution dans le temps au travers une relation entre les forces/couples appliquées et le mouvement résultant du robot (positions, vitesses et accélérations articulaires).

Cette modélisation est utile dans le cas de la commande et de la simulation. Le but étant l'obtention de l'évolution des différents angles, vitesses et accélérations angulaires en fonction des couples appliqués aux différentes articulations, ou l'inverse [5].

II.4.1 Le modèle dynamique inverse

On représente le modèle dynamique inverse par une relation [5]:

$$\boldsymbol{\tau} = \mathbf{f}(\boldsymbol{\Theta}, \dot{\boldsymbol{\Theta}}, \ddot{\boldsymbol{\Theta}}, \mathbf{F}_e) \quad (\text{II.22})$$

Avec :

$\boldsymbol{\tau}$: Vecteur des couples / forces articulaires ;

$\boldsymbol{\Theta}$: Vecteur des positions articulaires ;

$\dot{\boldsymbol{\Theta}}$: Vecteur des vitesses articulaires ;

$\ddot{\boldsymbol{\Theta}}$: Vecteur des accélérations articulaires ;

\mathbf{F}_e : Vecteurs de l'effort extérieur qu'exerce le robot sur l'environnement.

II.4.2 Le modèle dynamique direct

Le modèle dynamique direct (ou modèle de simulation) exprime les accélérations articulaires en fonction des positions, vitesses et couples des articulaires. Il est représenté par la relation suivante [13]:

$$\ddot{\boldsymbol{\Theta}} = \mathbf{g}(\boldsymbol{\Theta}, \dot{\boldsymbol{\Theta}}, \boldsymbol{\tau}, \mathbf{F}_e) \quad (\text{II.23})$$

Plusieurs formalismes ont été utilisés pour obtenir le modèle dynamique des robots manipulateurs, les plus souvent utilisés sont :

- ❖ Le Formalisme de Lagrange – Euler, et l'une des plus utilise ;
- ❖ Le formalisme de Newton- Euler.

II.4.2.1 Formalisme de Lagrange

L'approche de Lagrange-Euler sert à modéliser et à présenter la dynamique des robots à travers les équations du mouvement et permettent d'obtenir directement les relations entre les efforts des moteurs aux articulations et les mouvements. Cette méthode est basée sur le calcul des énergies cinétiques et potentielles du système. Ce qui se traduit, lorsque l'effort extérieur sur l'organe terminal est supposé nul, par l'équation suivante [16] :

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \tau_i \quad I = 1, 2, \dots, n \quad (\text{II.24})$$

Avec :

τ_i : Couple généralisé correspondant à l'articulation i

n : nombre de ddl du bras.

$L = \mathbf{K} - \mathbf{P}$: fonction de Lagrange.

\mathbf{K} : énergie cinétique totale du bras.

\mathbf{P} : énergie potentielle totale du bras.

q_i : variable articulaire généralisée.

II.4.2.2 Forme générale des équations dynamiques

L'énergie cinétique du système est une fonction quadratique des vitesses articulaires [16]:

$$\mathbf{E} = \frac{1}{2} \dot{\Theta}^T \mathbf{A} \dot{\Theta} \quad (\text{II.25})$$

Avec :

\mathbf{M} : est la matrice ($n \times n$) de l'énergie cinétique d'éléments génériques M_{ij} , appelée aussi matrice d'inertie du robot, qui est symétrique et définie positive. Ses éléments sont fonctions des variables articulaires Θ . L'énergie potentielle étant elle aussi fonction des variables articulaires Θ . Le couple τ_i peut se mettre, à partir de la fonction de Lagrange et l'équation de mouvement d'Euler (II-24), sous la forme :

$$\Gamma = \mathbf{M}(\Theta)\ddot{\Theta} + \mathbf{V}(\Theta, \dot{\Theta})\dot{\Theta} + \mathbf{G}(\Theta) \quad (\text{II.26})$$

Avec : $\mathbf{V}(\Theta, \dot{\Theta})\dot{\Theta}$: vecteur de dimension ($n \times 1$) représentant les couples/forces de Coriolis et

des forces centrifuges : $\mathbf{V}\dot{\Theta} = \dot{\mathbf{M}}\dot{\Theta} - \frac{\partial \mathbf{E}}{\partial \Theta}$

$\mathbf{G} = [Q_1 \dots Q_n]^T$: vecteur des couples/forces de gravité

II.4.2.3 Formalisme de Newton-Euler

Nous examinerons chaque maillon d'un manipulateur comme un corps rigide. Si nous connaissons l'emplacement du centre de masse et le tenseur d'inertie de la liaison, sa répartition des masses est complètement caractérisée. Afin de déplacer les liens, nous devons soit l'accélérer, soit les décélérer. Les forces nécessaires pour un tel mouvement sont fonction de l'accélération souhaitée et de la distribution de masse des liaisons. L'équation de Newton pour les mouvements de translation, ainsi que son analogue en rotation (équation d'Euler) décrivent comment les forces inerties et accélérations agissent.[16]

II.4.3 Modèle dynamique du robot PUMA 560

Ce paragraphe présente le modèle dynamique du bras de robot PUMA 560. Les trois dernières articulations de ce manipulateur qui constituent le poignet (Wrist) sont fixées à la position zéro (c'est-à-dire) :

- Les positions $\theta_4 = 0, \theta_5 = 0, \theta_6 = 0$
- Les vitesses $\dot{\theta}_4 = 0, \dot{\theta}_5 = 0, \dot{\theta}_6 = 0$
- Les accélérations $\ddot{\theta}_4 = 0, \ddot{\theta}_5 = 0, \ddot{\theta}_6 = 0$

Nous avons adopté l'abréviation des fonctions trigonométriques des coordonnées articulaires du robot en écrivant, $c_i, s_i, c_{ij}, s_{ij}, c_{ijk}$ pour signifier $\cos(\theta_i), \sin(\theta_i), \cos(\theta_i + \theta_j), \cos(\theta_i + \theta_j + \theta_k)$ respectivement. Le modèle dynamique pour les trois premières articulations du bras de robot PUMA 560 est le suivant :

$$M(\theta) \ddot{\theta} + V(\theta, \dot{\theta}) \dot{\theta} + G(\theta) = \Gamma \quad (\text{II.27})$$

Avec :

- $\theta = [\theta_1, \theta_2, \theta_3]^T$: le vecteur des coordonnées généralisées,
- \mathbf{m} : est le poids de la charge à porter,
- $\mathbf{J}(\theta)$: la matrice Jacobienne.

Les expressions des éléments des matrices M, V et le vecteur G sont les suivants [15]:

Les éléments de la matrice d'inertie $M(\theta)$:

$$\begin{aligned}
 m_{11} &= 2.57 + 1.38 c_2^2 + 0.30 s_2 s_3 + 0.744 c_2 s_{23} & (II.28) \\
 m_{12} = m_{21} &= 0.69 s_2 - 0.134 c_{23} + 0.0238 c_2 \\
 m_{13} = m_{31} &= -0.134 c_{23} - 0.00397 s_{23} \\
 m_{22} &= 6.79 + 0.744 s_3 \\
 m_{23} = m_{32} &= 0.333 + 0.372 s_3 - 0.011 c_3 \\
 m_{33} &= 1.16
 \end{aligned}$$

Les éléments de la matrice des forces centrifuge et Coriolis $V(\theta, \dot{\theta})$:

$$\begin{aligned}
 v_{11} &\approx -2.76 s_2 c_2 + 0.744 c_{223} + 0.6 s_2 c_3 - 0.0213(1-2 s_2 s_3) \dot{\theta}_2 & (II.29) \\
 v_{12} &\approx (0.69 c_2 + 0.134 s_{23} - 0.0238 s_2) \dot{\theta}_2 + (0.267 s_{23} - 0.00758 c_{23}) \dot{\theta}_3 \\
 v_{13} &\approx (0.744 c_2 c_{23} + 0.6 s_2 c_3 + 0.022 c_2 s_{23} - 0.0213(1-2 s_2 s_3)) \dot{\theta}_1 + 0.5(0.267 \\
 & s_{23} - 0.00758 c_{23}) \dot{\theta}_3 \\
 v_{21} &\approx -0.5(-2.76 s_2 c_2 + 0.744 c_{223} + 0.6 s_2 c_3 - 0.0213(1-2 s_2 s_3)) \dot{\theta}_1 \\
 v_{22} &\approx (0.022 s_3 + 0.744 c_3) \dot{\theta}_3 \\
 v_{23} &\approx 0.5*(0.022 s_3 + 0.744 c_3) \dot{\theta}_3 \\
 v_{31} &\approx -0.5(0.744 c_2 c_{23} + 0.6 s_2 c_3 + 0.022 c_2 s_{23} - 0.0213(1-2 s_2 s_3)) \dot{\theta}_1 \\
 v_{32} &\approx -0.5(0.022 s_3 + 0.744 c_3) \dot{\theta}_2 \\
 v_{33} &= 0
 \end{aligned}$$

Les éléments du vecteur des forces gravitationnelles $G(\theta)$:

$$\begin{aligned}
 g_1 &= 0 & (II.30) \\
 g_2 &= -37.2 c_2 - 8.4 s_{23} + 1.02 s_2 \\
 g_3 &= -8.4 s_{23} + 0.25 c_{23}
 \end{aligned}$$

La démonstration et le calcul des éléments des matrices M , V et le vecteur G du modèle dynamique de ce bras PUMA 560 est présenté dans [15].

II.5 Conclusion

Ce chapitre permis la découverte des notions de base de la modélisation des robots manipulateurs en partant de la modélisation géométrique jusqu'à la modélisation dynamique, ou il est représenté le formalisme de Lagrange pour l'obtention de l'équation de mouvement des robots manipulateurs.

En parallèle à cela une application de calcul des différents modèles du robot PUMA 560 est donné afin de les utilisés par la suite dans les chapitres suivants qui seront consacrés aux méthodes de planification et de commandes des bras de robots.

Chapitre III :

Planification de

Trajectoires

III.1 Introduction

En robotique, on est souvent amené à devoir planifier des trajectoires pour permettre à un robot de se déplacer d'un point initial à un point final. Plusieurs algorithmes permettant de réaliser une telle tâche. Dans ce chapitre, on présentera certaines de ces méthodes en expliquant brièvement leur principe ainsi que leurs avantages et leurs inconvénients.

III.2 Planification de trajectoire

Dans l'espace articulaire, on désigne par trajectoire, les variations dans le temps de la position, de la vitesse et de l'accélération de chaque articulation. Ces trajectoires sont calculées sous forme d'un ensemble de points, selon une certaine fréquence appelée taux de mise à jour du chemin. Dans les systèmes de manipulateur typiques, ce taux se situe entre 60 et 2000 Hz.

Dans l'espace opérationnel, une trajectoire décrit le mouvement (position et orientation) du repère associé à l'outil, par rapport au repère associé à la station de travail. Le problème de génération du mouvement revient donc au calcul des consignes de référence en position, vitesse et accélération.

On peut distinguer deux classes de mouvements :

- Mouvement entre deux points avec une trajectoire libre entre les deux extrémités,
- Mouvement entre des points via (intermédiaires), spécifiés notamment pour éviter les obstacles, avec une trajectoire libre entre les points.

III.2.1 Méthodes de Planification dans l'espace articulaire

Chaque point de la trajectoire généré représente ici l'angle de rotation désirée de chaque articulation. Une fonction lisse est ainsi trouvée pour chacune de ces dernières (pour garantir des mouvements doux, et éviter de provoquer des vibrations). Celle-ci passe par les points intermédiaires et se termine au point final. Le temps exigé pour chaque segment est le même pour toutes les articulations, afin d'atteindre le point via en même temps. Pour passer d'un point à l'autre, plusieurs fonctions peuvent être envisagées. Mais, on doit tenir compte de certaines contraintes.

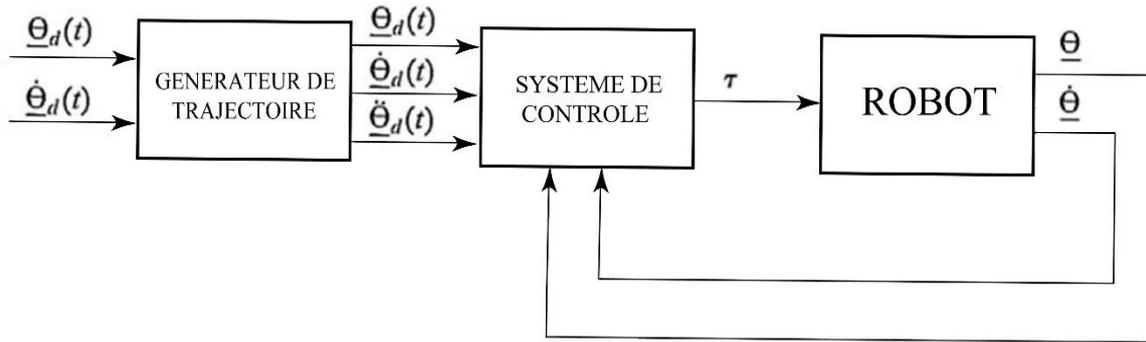


Figure III.1 Génération de mouvement dans l'espace articulaire

III.2.1.1 Trajectoires polynomiales cubiques

Comme nous avons déjà les positions articulaire θ_0 et θ_f , nous allons chercher une fonction permettant de relier la position θ_0 à l'instant t_0 à la position θ_f à l'instant t_f .

Comme indiqué sur Figure III.2, de nombreuses fonctions lisses, $\theta(t)$, qui pourraient être utilisées pour interpoler la valeur de l'articulation.[5]

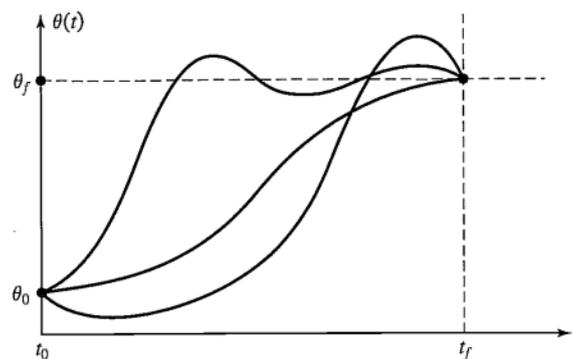


Figure III.2 Plusieurs formes de chemin possible pour une seule articulation

Ainsi En faisant un seul mouvement régulier, au moins quatre contraintes sur $\theta(t)$ sont évidentes. Deux contraintes sur la valeur de la fonction proviennent de la sélection des valeurs initiale et valeurs finales [5].

$$\theta(0) = \theta_0 \quad (\text{III.1})$$

$$\theta(t_f) = \theta_f$$

Et deux autres contraintes sur la continuité de la vitesse :

$$\dot{\theta}(0) = 0 \quad (\text{III.2})$$

$$\dot{\theta}(t_f) = 0$$

Pour satisfaire ces contraintes, il faut un polynôme de degré ≥ 3 ; d'où l'usage de polynômes cubiques (de degré 3) ayant quatre paramètres [5].

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

$$\dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2 \quad (\text{III.4})$$

$$\ddot{\theta}(t) = 2a_2 + 6a_3 t \quad (\text{III.5})$$

En combinant les formules (III 3, III 4, III 5) avec les quatre contraintes souhaitées cela nous donnera quatre équations dont quatre inconnues :

$$\theta_0 = a_0 \quad (\text{III.6})$$

$$\theta_f = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 \quad (\text{III.7})$$

$$0 = a_1 \quad (\text{III.8})$$

$$0 = a_1 2a_2 t_f + 3a_3 t_f^2 \quad (\text{III.9})$$

Après résolution cela nous donnera :

$$a_0 = \theta_0 \quad (\text{III.10})$$

$$a_1 = 0 \quad (\text{III.11})$$

$$a_2 = \frac{3}{t_f^2} (\theta_f - \theta_0) \quad (\text{III.12})$$

$$a_3 = -\frac{2}{t_f^3} (\theta_f - \theta_0) \quad (\text{III.13})$$

- Cette solution ne concerne que le cas où le bras démarre et finit à vitesse nulle.

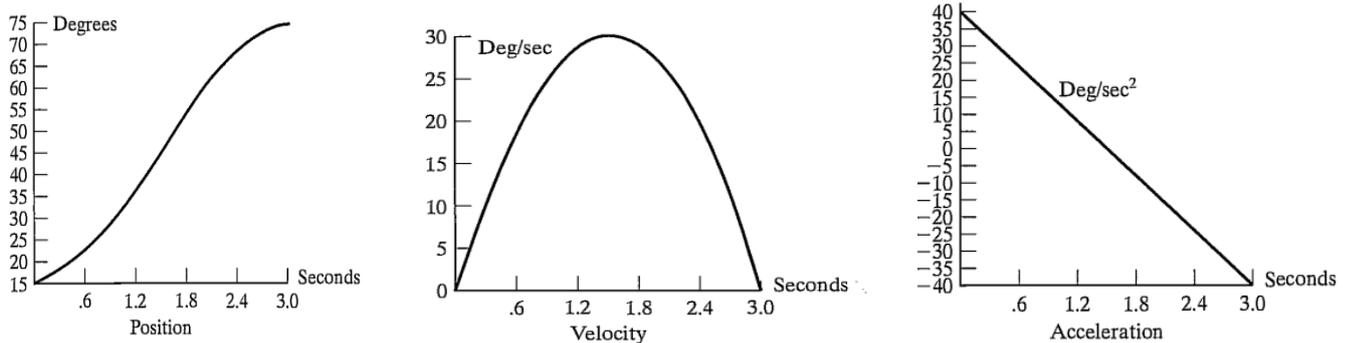


Figure III.3 Allures de la position, de la vitesse et de l'accélération

pour $\theta_0 = 15^\circ$, $\theta_f = 75^\circ$ et $t_f = 3$ s. [10]

III.2.1.2 Polynômes cubiques pour une trajectoire avec des points via

En général nous souhaitons déterminer un chemin incluant des points intermédiaires. Si le manipulateur doit s'immobiliser à chaque point de passage, nous pouvons utiliser la solution qu'on vient de présenter dans la section précédente. Cependant, il est impératif de pouvoir passer par un point via sans s'arrêter, pour cela nous devons donc généraliser la manière dont nous adaptons les cubiques aux contraintes de chemin.[5]

Si les vitesses souhaitées des articulations aux points via sont connues, alors nous pouvons construire des polynômes cubiques comme précédemment, cependant les contraintes de vitesse à chaque fin ne sont pas nulles, mais plutôt une vitesse connue. Les contraintes de (3) deviennent alors. [5]

$$\dot{\theta}(0) = \dot{\theta}_0 \quad (\text{III.14})$$

$$\dot{\theta}(t_f) = \dot{\theta}_f \quad (\text{III.15})$$

On écrit alors :

$$\theta_0 = a_0 \quad (\text{III.16})$$

$$\theta_f = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 \quad (\text{III.17})$$

$$\dot{\theta}_0 = a_1 \quad (\text{III.18})$$

$$\dot{\theta}_f = a_1 + 2a_2 t_f + 3a_3 t_f^2 \quad (\text{III.19})$$

Après résolution des a_i , nous obtiendrons

$$a_0 = \theta_0 \quad (\text{III.20})$$

$$a_1 = \dot{\theta}_0 \quad (\text{III.21})$$

$$a_2 = \frac{3}{t_f^2} (\theta_f - \theta_0) - \frac{2}{t_f} \dot{\theta}_0 - \frac{1}{t_f} \dot{\theta}_f \quad (\text{III.22})$$

$$a_3 = -\frac{2}{t_f^3} (\theta_f - \theta_0) + \frac{2}{t_f} (\dot{\theta}_f + \dot{\theta}_0) \quad (\text{III.23})$$

De cette façon, on calcule le polynôme cubique entre deux points avec des vitesses initiales et finales quelconques.

Plusieurs options peuvent se présenter pour la spécification des vitesses aux points via (si celles-ci ne sont pas données) :

- 1- L'utilisateur spécifie les vitesses désirées aux points via, en termes des vitesses linéaires et angulaires de l'outil dans l'espace Cartésien.
- 2- Le système choisit automatiquement les vitesses aux points via, par l'application de règles heuristiques.

- 3- Le système choisit automatiquement les vitesses, de telle sorte que l'accélération reste continue aux points via.

Une méthode heuristique simple consiste à :

- Choisir une vitesse nulle si la pente des segments change de signe,
- Si les pentes des segments ne changent pas de signe, choisir la pente moyenne comme vitesse au point via.

III.2.2 Avantages et inconvénients

La génération de mouvement dans l'espace articulaire présente plusieurs avantages :

- Elle nécessite moins de calculs en ligne, puisqu'il n'y a pas d'appel au modèle géométrique ou cinématique inverse ;
- Le mouvement n'est pas affecté par le passage sur les configurations singulières ;
- Les contraintes de vitesses et de couples maximaux sont directement déduites des limites physiques des actionneurs.

En contrepartie, la géométrie de la trajectoire de l'outil dans l'espace opérationnel est imprévisible. Ce type de mouvement est par conséquent approprié pour réaliser des déplacements rapides dans un espace dégagé. La génération de mouvement dans l'espace opérationnel permet de contrôler la géométrie de la trajectoire [14].

III.3 Méthodes de Planification dans l'espace opérationnelle

Dans cette section, nous considérons des méthodes de génération de trajectoires dans laquelle la forme du chemin généré est décrite en termes de points cartésiens (x,y,z) exprimant ainsi La position souhaitée de l'effecteur.

Les positions, vitesses et accélération générées par le MGT sont "converties" en un ensemble d'angles, vitesses et accélérations articulaires, par la baie du modèle géométrique inverse, le modèle différentiel inverse et la dérivée du modèle différentiel inverse.

Ensuite une fonction lisse est trouvée pour chacune des n articulations qui passe par les points intermédiaires et se terminer au point final. Comme dans le cas précédant, le temps nécessaire pour chaque segment est le même pour chaque articulation de sorte que toutes les articulations atteignent le point de passage en même temps, résultant ainsi la position cartésienne souhaitée de l'effecteur à chaque point de passage [5].

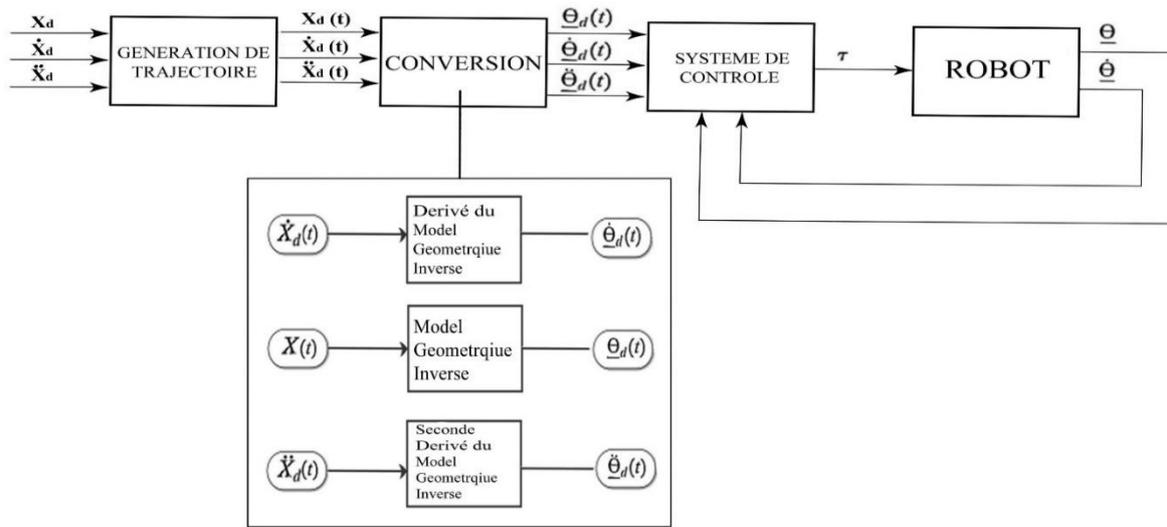


Figure III.4 Génération de mouvement dans l'espace opérationnel

$$\Theta_d = MGI(X_d) \tag{III.24}$$

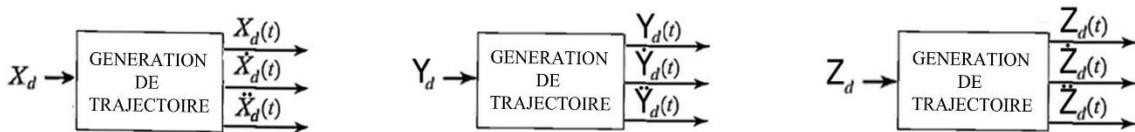
$$\dot{\Theta}_d = J^{-1}(\Theta)\dot{X}_d \tag{III.25}$$

$$\ddot{\Theta}_d = j^{-1}(\Theta)\ddot{X}_d + J^{-1}(\Theta)\dot{X}_d \tag{III.26}$$

III.3.1 Polynôme cubique

Nous utiliserons le même principe que nous avons vu dans la section espace articulaire, sauf qu'ici au lieu d'utiliser un seul module de génération de trajectoire nous utiliserons trois. On illustre la méthode ainsi [5].

❖ Module de génération de trajectoire x, y et z



$$X_d(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

$$\dot{X}_d(t) = a_1 + 2 a_2 t + 3 a_3 t^2$$

$$\ddot{X}_d(t) = 2 a_2 + 6 a_3 t$$

$$a_0 = X_0$$

$$a_1 = 0$$

$$a_2 = \frac{3}{t_f^2} (X_f - X_0)$$

$$a_3 = -\frac{2}{t_f^3} (X_f - X_0)$$

$$Y_d(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

$$\dot{Y}_d(t) = a_1 + 2 a_2 t + 3 a_3 t^2$$

$$\ddot{Y}_d(t) = 2 a_2 + 6 a_3 t$$

$$a_0 = Y_0$$

$$a_1 = 0$$

$$a_2 = \frac{3}{t_f^2} (Y_f - Y_0)$$

$$a_3 = -\frac{2}{t_f^3} (Y_f - Y_0)$$

$$Z_d(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

$$\dot{Z}_d(t) = a_1 + 2 a_2 t + 3 a_3 t^2$$

$$\ddot{Z}_d(t) = 2 a_2 + 6 a_3 t$$

$$a_0 = Z_0$$

$$a_1 = 0$$

$$a_2 = \frac{3}{t_f^2} (Z_f - Z_0)$$

$$a_3 = -\frac{2}{t_f^3} (Z_f - Z_0)$$

III.3.2 Mouvement en points via

$$\begin{aligned}
 X_d(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 & Y_d(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 & Z_d(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 \\
 \dot{X}_d(t) &= a_1 + 2 a_2 t + 3 a_3 t^2 & \dot{Y}_d(t) &= a_1 + 2 a_2 t + 3 a_3 t^2 & \dot{Z}_d(t) &= a_1 + 2 a_2 t + 3 a_3 t^2 \\
 \ddot{X}_d(t) &= 2 a_2 + 6 a_3 t & \ddot{Y}_d(t) &= 2 a_2 + 6 a_3 t & \ddot{Z}_d(t) &= 2 a_2 + 6 a_3 t \\
 a_0 &= X_0 & a_0 &= Y_0 & a_0 &= Z_0 \\
 a_1 &= \dot{X}_0 & a_1 &= \dot{Y}_0 & a_1 &= \dot{Z}_0 \\
 a_2 &= \frac{3}{t_f^2} (X_f - X_0) - \frac{2}{t_f} \dot{X}_0 - \frac{1}{t_f} \dot{X}_f & a_2 &= \frac{3}{t_f^2} (Y_f - Y_0) - \frac{2}{t_f} \dot{Y}_0 - \frac{1}{t_f} \dot{Y}_f & a_2 &= \frac{3}{t_f^2} (Z_f - Z_0) - \frac{2}{t_f} \dot{Z}_0 - \frac{1}{t_f} \dot{Z}_f \\
 a_3 &= -\frac{2}{t_f^3} (X_f - X_0) + \frac{2}{t_f} (\dot{X}_f + \dot{X}_0) & a_3 &= -\frac{2}{t_f^3} (Y_f - Y_0) + \frac{2}{t_f} (\dot{Y}_f + \dot{Y}_0) & a_3 &= -\frac{2}{t_f^3} (Z_f - Z_0) + \frac{2}{t_f} (\dot{Z}_f + \dot{Z}_0)
 \end{aligned}$$

III.3.3 Avantages et inconvénients

La génération de mouvement dans l'espace opérationnel permet de contrôler la géométrie de trajectoire de l'effecteur et vérifier l'absence de collisions. En revanche :

- Elle implique la transformation en coordonnées articulaires de chaque point de la trajectoire.
- Elle peut être mise en échec lorsque la trajectoire calculée passe par une position singulière.
- Elle peut être mise en échec lorsque la trajectoire calculée passe par des points qui sont inaccessible pour l'effecteur.
- Les limites en vitesse et en couple dans l'espace opérationnel varient selon la configuration du robot. On doit alors exprimer ces limites par des valeurs traduisant des performances moyennes, satisfaites quelle que soit la configuration du robot. On impose donc au robot de travailler au-delà de ses capacités réelles.
- Si le manipulateur peut atteindre tous les points du chemin avec des solutions multiples.

III.4 Planification de trajectoire pour le robot puma 560

Dans cette section nous présentons quelques résultats de simulation illustrant l'applicabilité de notre démarche de planification de chemin dans les deux espaces (articulaire et opérationnel) pour le robot PUMA 560. On introduira ainsi les consignes désirées à ce dernier puis en visualisera les résultats.

Une animation sera rajoutée afin de mieux appréhender les choses. Elle est réalisée en utilisant le toolbox robotic system de Peter Corke [20].

III.4.1 Planification dans l'espace articulaire

Dans ce qui suit nous allons planifier les trajectoires des trois premières articulations du robot. Les trois dernières articulations qui constituent le poignet (Wrist) seront fixées à la position zéro, ($\theta_4 = 0, \theta_5 = 0, \theta_6 = 0$).

III.4.1.1 Mouvement entre deux points

Les valeurs initiales et finales des variables articulaires sont données par Tableau III-1. La durée du segment est choisie $T_f = 2 \text{ sec}$.

Tableau III-1 mouvement des articulations entre deux points

	Valeur d'initiale θ_i	Valeur de finale θ_f
Première articulation θ_1	0°	100°
Deuxième articulation θ_2	0°	90°
Troisième articulation θ_3	0°	60°

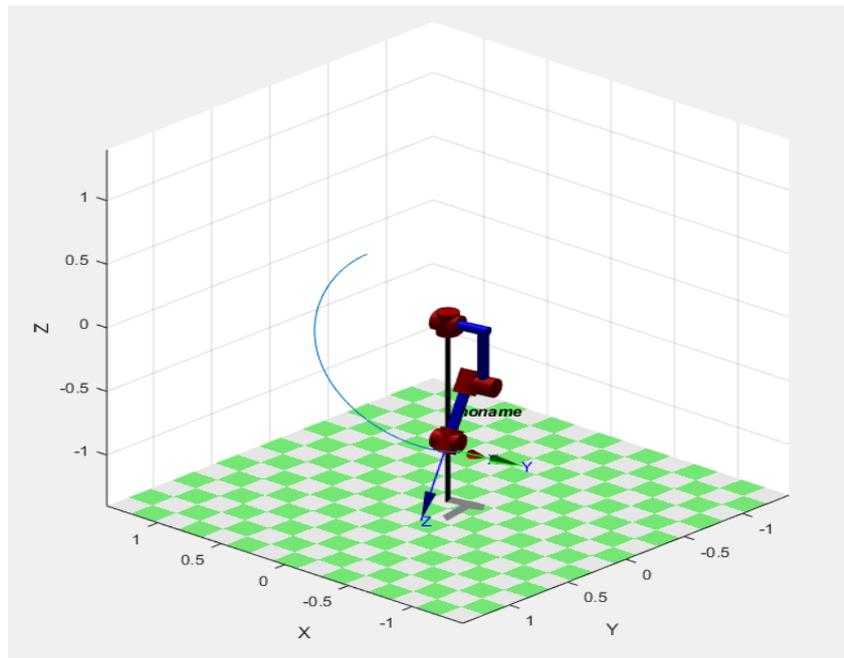
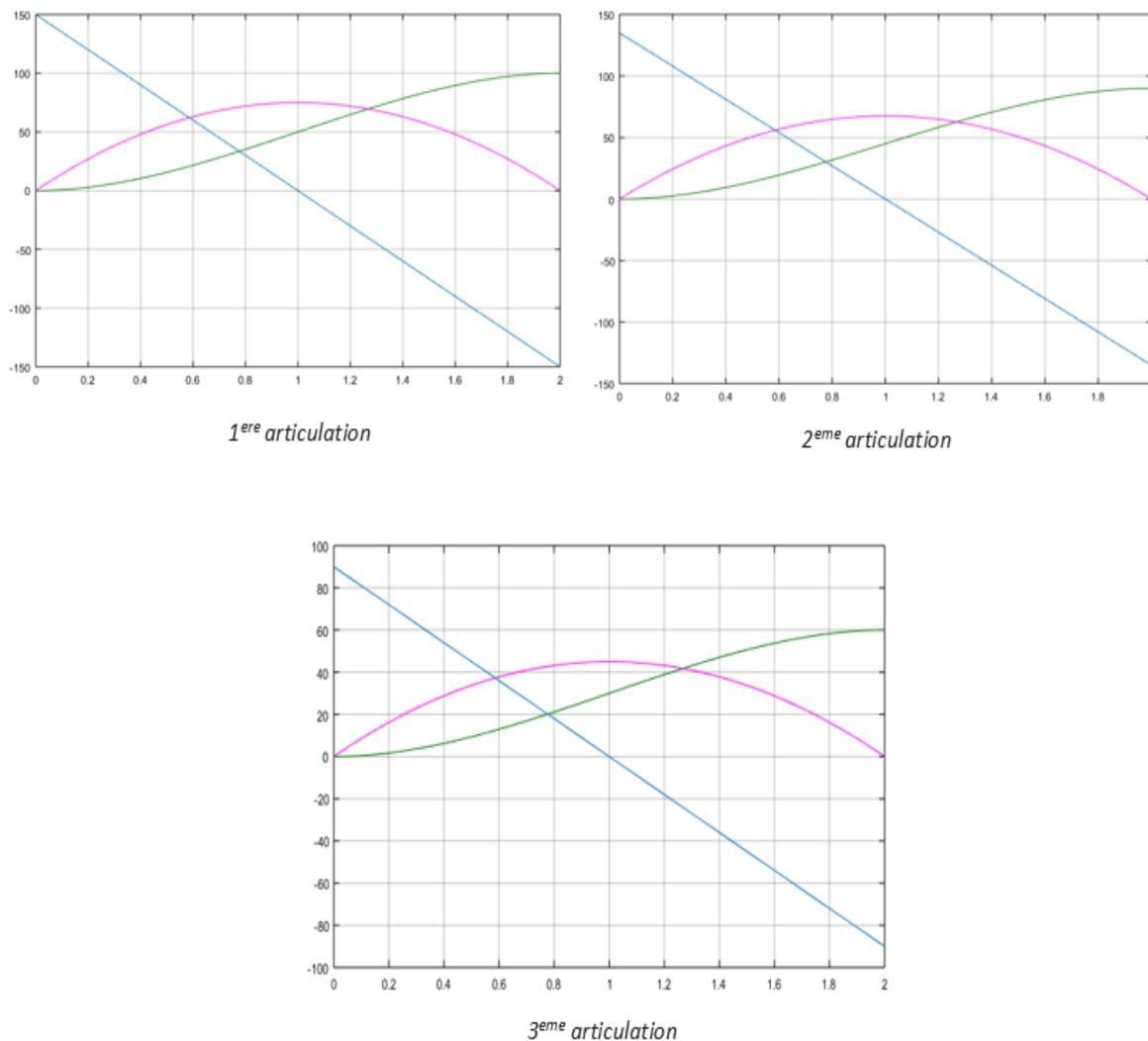


Figure III.5 Illustration en 3D de la trajectoire générée dans l'espace articulaire, (cas de mouvement entre deux points)

La trajectoire 3D que le robot doit suivre est illustrée par la Figure (III.5). Elle est obtenue en effectuant une transformation des trajectoires de référence de l'espace articulaire en trajectoires équivalentes dans l'espace opérationnel, et ce en utilisant le modèle géométrique direct.

- ⇒ Le polynôme cubique sera calculé par la fonction Matlab présentée dans l'annexe A.0.
- ⇒ Le programme de l'animation du robot est donné dans l'annexe A.3.

L'évolution des différentes grandeurs articulaires sont données par la figure ci-dessous :



III.6 Evolution des variables articulaires de référence: position (en vert), vitesses (en violet) et accélérations (en bleu)

III.4.1.2 Mouvement entre points via

Trois points-via sont considérés. Les valeurs des variables articulaires des différents points sont données par Tableau (III-2). La durée des quatre segments est considérée constante (2 s), pour toutes les articulations, (le temps global est à 8s).

La trajectoire que prendra le robot est illustrée par la Figure III.7, obtenue à travers le modèle géométrique direct.

Tableau III-2 Valeurs des variables articulaires pour les différents points

	X_1	X_2	X_3	X_4	X_5
1^{ere} ARTICULATION θ_1	10°	50°	100°	200°	250°
2^{eme} ARTICULATION θ_2	-50°	-100°	-150°	-200°	-200°
3^{eme} ARTICULATION θ_3	0°	50°	50°	100°	100°

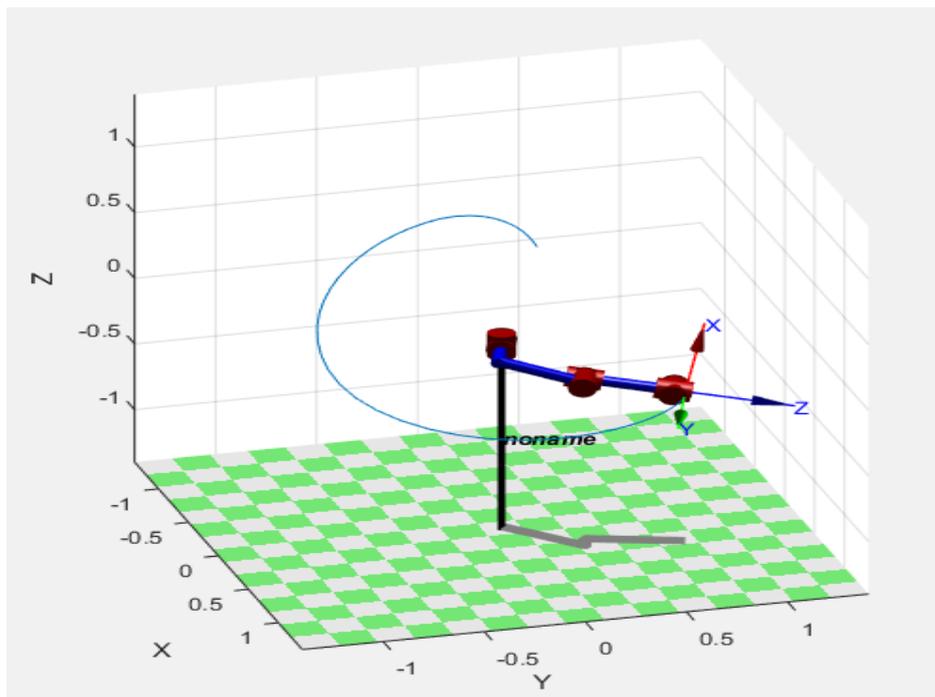


Figure III.7 Illustration de la trajectoire générée dans l'espace 3D, avec points-via.

- ⇒ La fonction qui calcule le polynôme cubique en point via est donné dans l'annexe A.1.
- ⇒ Le programme de l'animation du robot est donné dans l'annexe A.3.

L'évolution des différentes grandeurs articulaires sont données par Figure suivant :

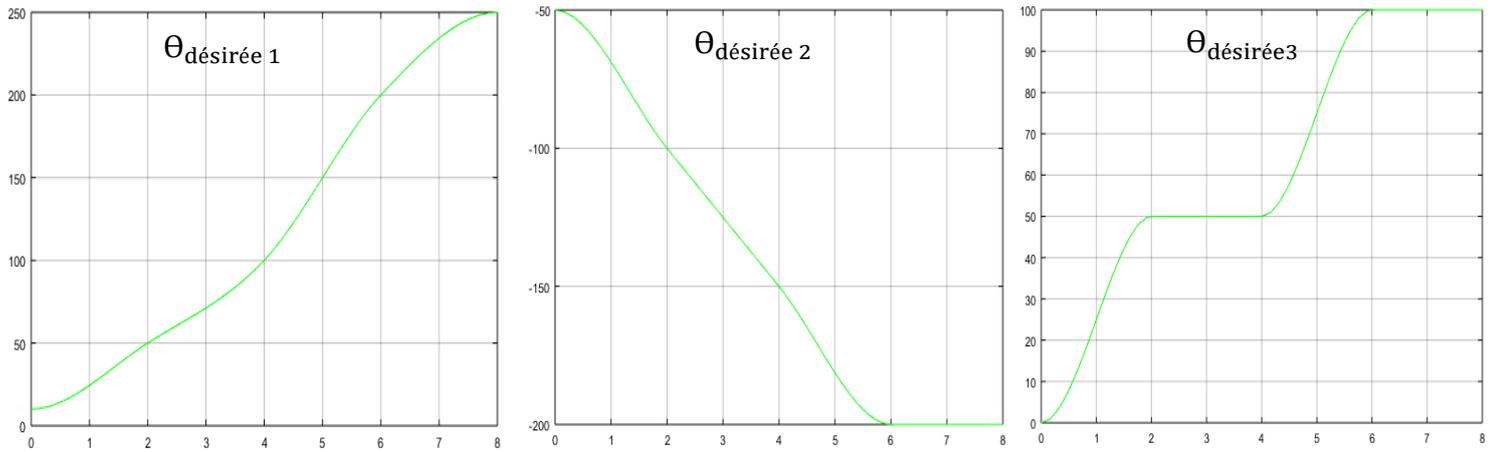


Figure III.8 Évolution des différents variables articulaires *désirées* pour les articulations de référence (1,2 et 3)

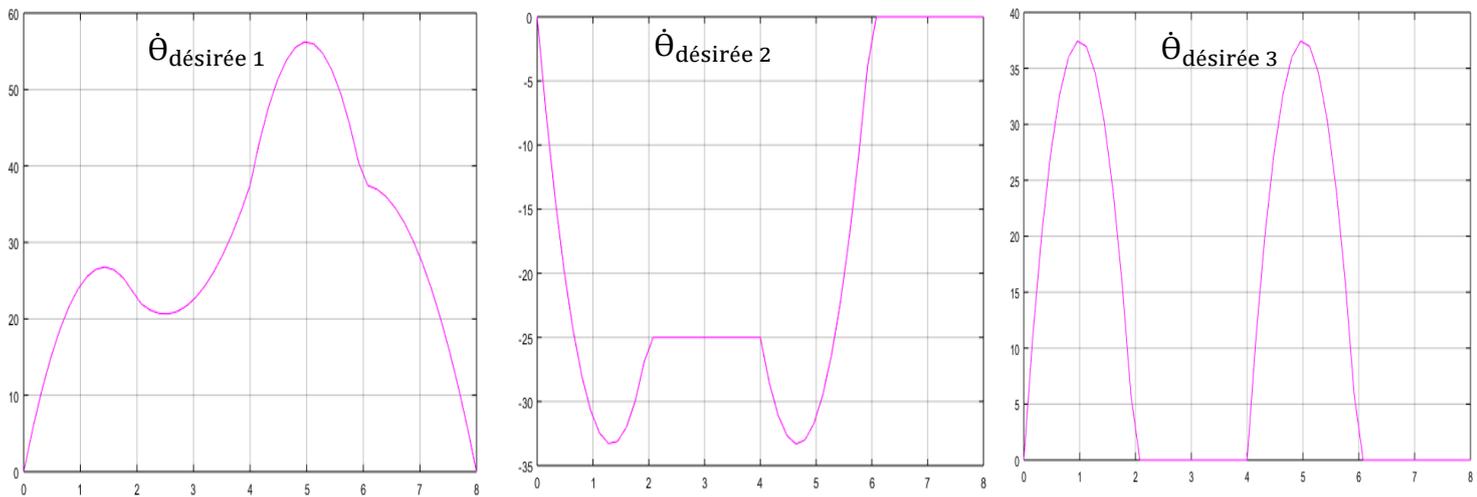


Figure III.9 Évolution des différentes variables de vitesses *désirées* pour les articulations de référence (1,2 et 3)

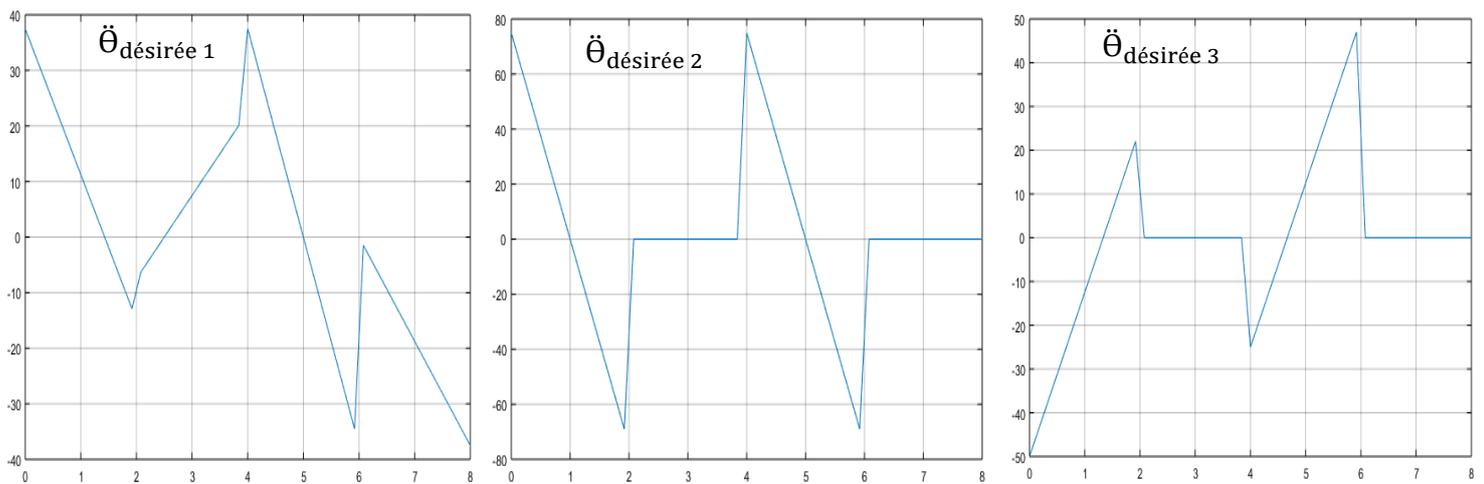


Figure III.10 Évolution des différentes variables d'accélération *désirées* pour les articulations de référence (1,2 et 3)

III.4.2 Planification dans l'espace opérationnel

Dans cette section, nous allons planifier le chemin que prendra l'effecteur à travers des points cartésiens que nous injecterons dans le programme.

Après avoir mis les données de trajectoire le MGT (module de générations de trajectoire) va générer le mouvement que va effectuer le robot. Ensuite, on utilise les modules de conversion (modèle géométrique inverse, modèle différentielle inverse, et la dérivée du modèle différentielle inverse) qui vont convertir les trajectoires de l'espace cartésien à l'espace articulaire qui vont servir aux boucles de régulation du robot.

L'organigramme (simulink) qui illustre ce fonctionnement est donné en annexe A.2. Les trajectoires articulaires sont enregistrées dans le Workspace pour simuler l'animation du robot.

III.4.2.1 Simulation

Pour cette simulation nous avons planifié une trajectoire entre deux stations (station A et station B), le robot passera donc par les points suivants :

Tableau III-3 les points de trajectoire entre A et B

	X1	X2	X3	X4	X5	X6
X	-0.5	-0.1	0.2	0.4	0.6	0.3
Y	0.7	0.7	0.6	0.4	-0.08	-0.6
Z	0.2	0.2	0.2	0.2	-0.4	-0.5

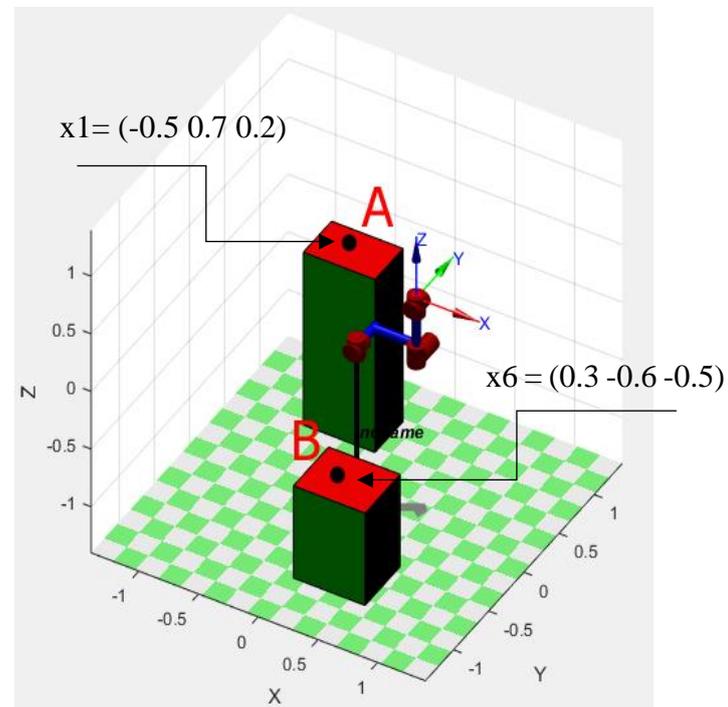


Figure III.11 Illustration 3D des stations A, B

On fixera les durées des segments $t = [1 \ 1 \ 1 \ 2 \ 1]$, le temps global étant égal 6s.

- ⇒ Les fonctions qui calculent les polynômes cubiques, le modèle géométrique inverse ainsi que le dérivé du modèle géométrique inverse et ses seconds dérivés sont données dans l'annexe A.2.
- ⇒ Le programme de l'animation du robot puma 560 dans l'espace opérationnel est donné dans l'annexe A.4.

Résultats :

La trajectoire générée par le MGT est donnée dans la Figure III.12 Les marques rouges et vertes représentent les points via que nous avons introduit dans le programme.

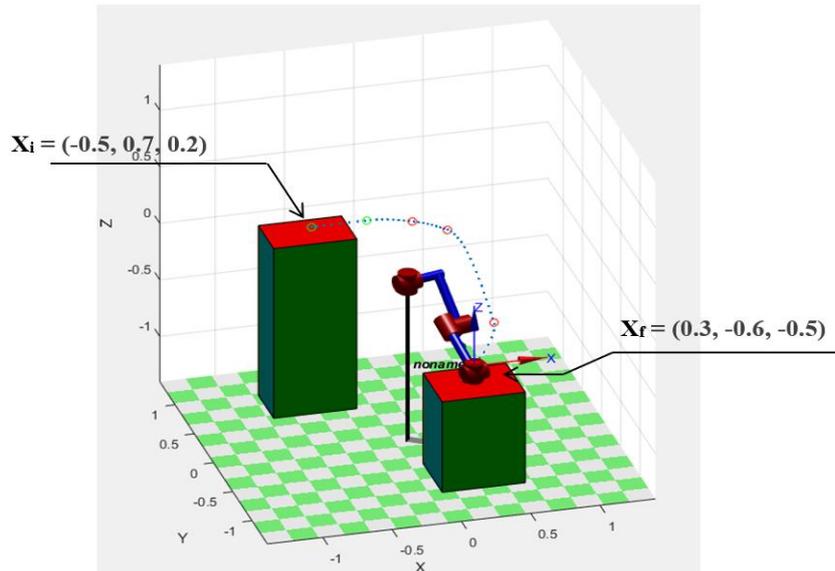


Figure III.12 Visualisation en 3D de la trajectoire générée dans l'espace opérationnel.

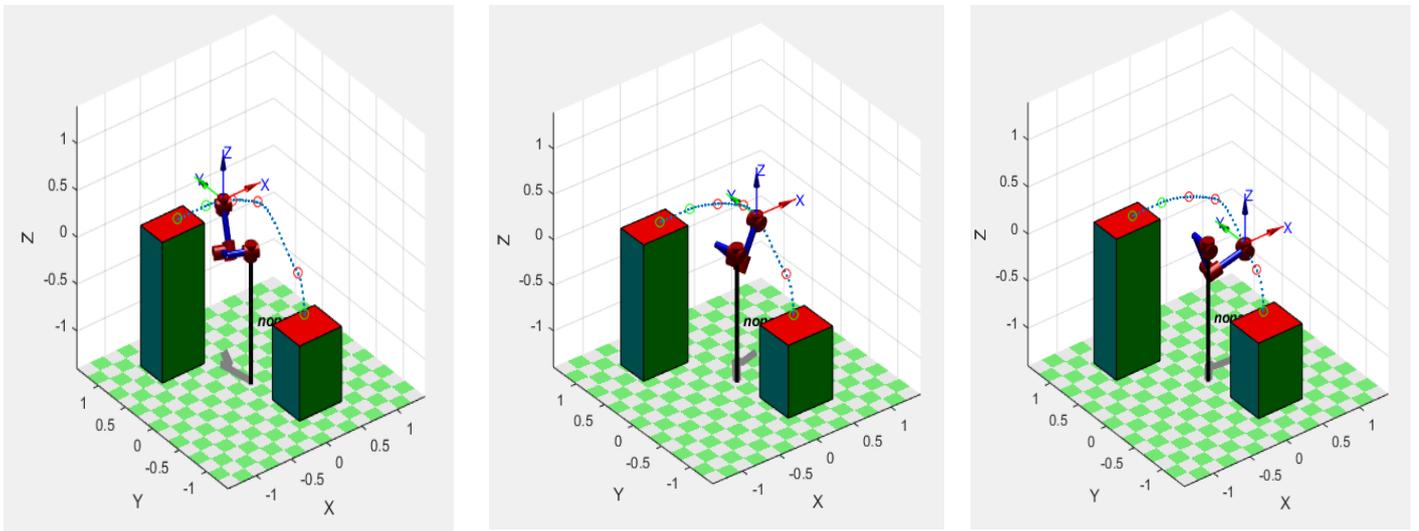


Figure III.13 Animation 3D du Robot

L'évolution des différentes grandeurs articulaires sont données par la figure ci-dessous :

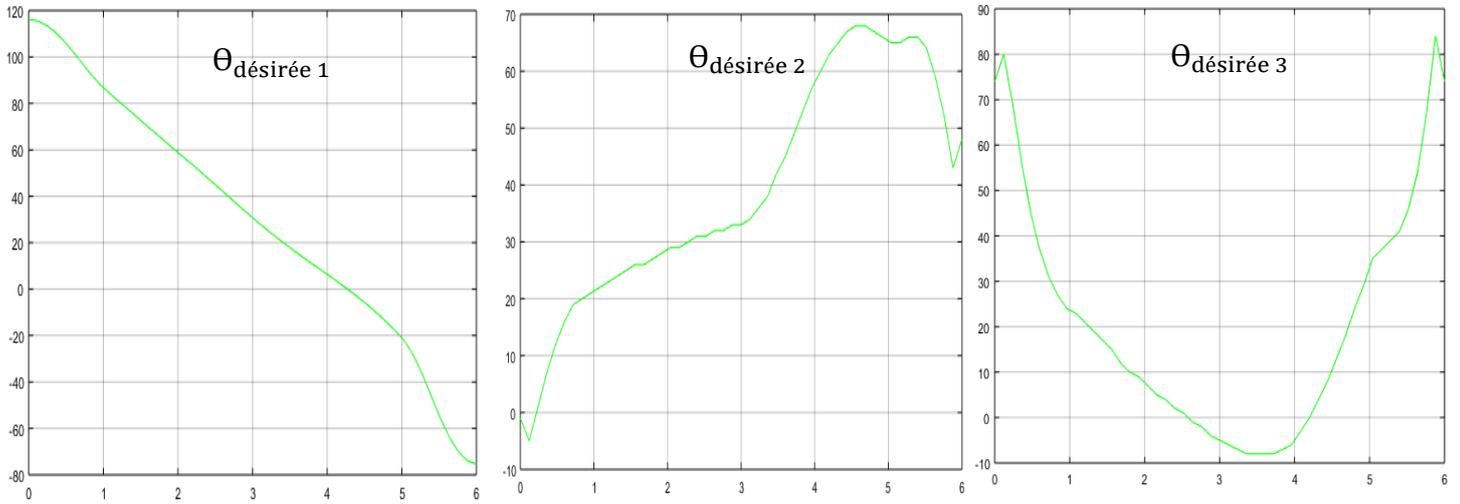


Figure III.14 Evolution des variables articulaires *désirées* dans l'espace opérationnel pour les articulations de référence (1,2 et 3)

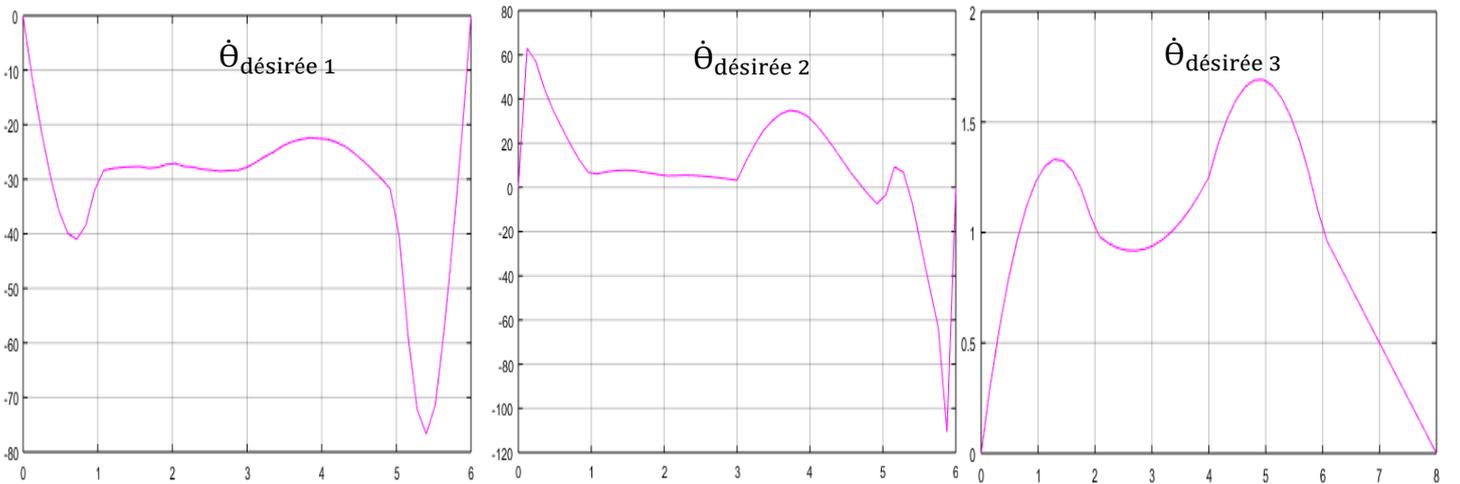


Figure III.15 Evolution des vitesses *désirées* dans l'espace opérationnel pour les articulations de référence (1,2 et 3)

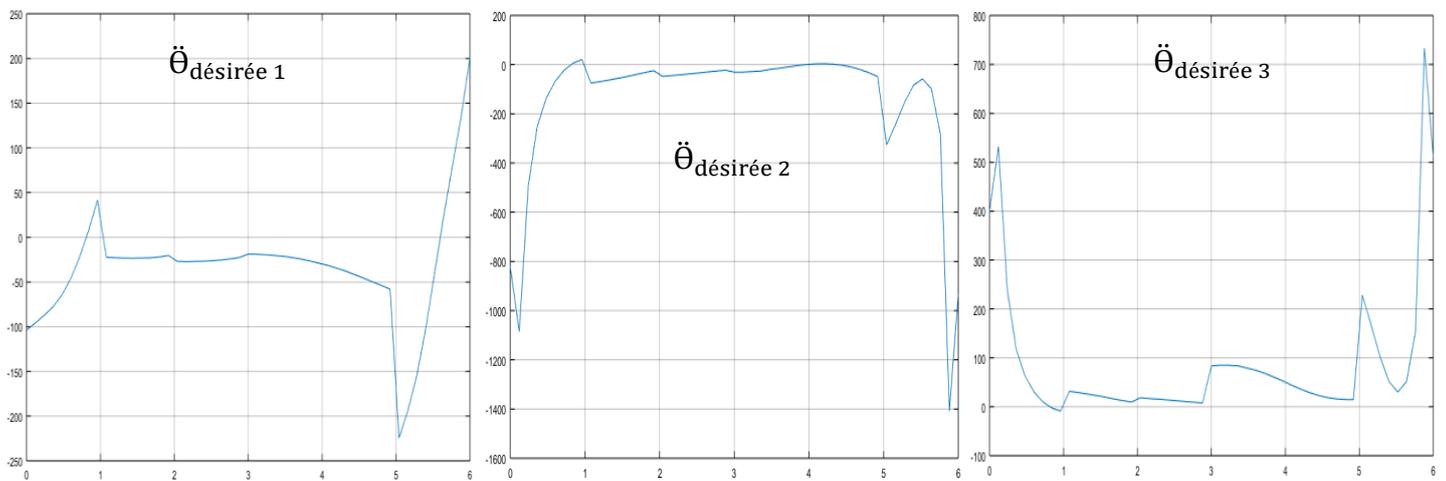


Figure III.16 Evolution de l'accélération *désirées*, dans l'espace opérationnel pour les articulations de référence (1,2 et 3)

III.5 Conclusion

Dans ce chapitre, nous avons exposé les méthodes de planification de trajectoires dans les espaces articulaire et opérationnel, ainsi que les avantages et inconvénient de la planification dans chacun de ces espaces. Puis on a fini par le calcul et la simulation de ces méthodes sur le robot PUMA560, en visualisant les différents résultats en 3D.

Chapitre IV :

Commande des bras

de robots

IV.1 Introduction

La commande du mouvement d'un manipulateur est un sujet qui peut être formulé à partir de problématiques plus générales issues de l'automatique. Dans le domaine de la commande de systèmes dynamiques, on peut distinguer trois problématiques essentielles pour notre sujet

- La stabilisation.
- Le suivi de trajectoires.
- Le rejet de perturbations.

La condition nécessaire à la mise en œuvre d'un système dynamique comme un robot est de garantir sa stabilité. La stabilisation d'un robot manipulateur autour d'une consigne de position constante au cours du temps est donc le premier objectif à atteindre. Mais, pour les applications industrielles de robots manipulateurs, la stabilisation n'est pas suffisante. En effet, pour effectuer les tâches décrites précédemment, un robot manipulateur doit suivre des trajectoires avec la précision attendue. Les trajectoires de référence prédéfinies pour générer le mouvement souhaité constituent, les consignes à suivre pour chaque articulation qui évoluent au cours du temps de manière synchronisée. Ce suivi de trajectoire de référence occupe une place centrale en commande de robot manipulateur. Et il est bien connu que le problème de suivi de trajectoire est plus difficile que celui de la stabilisation.

Enfin, le rejet de perturbation consiste à minimiser l'effet de perturbations sur la sortie, comme des phénomènes "mal modélisés" au sein du système en raison de leur caractère incertain ou trop complexe [1].

La commande du mouvement d'un robot manipulateur inclue donc ces trois problématiques. Les travaux présentés dans ce mémoire poursuivront l'objectif principal du suivi de trajectoire, en définissant les lois de commande les plus couramment utilisés en robotique.

IV.2 Lois de commande de suivi de trajectoire

Aujourd'hui encore, la plupart des robots industriels sont commandés avec des lois linéaires décentralisées (articulation par articulation) de type PID. Néanmoins, depuis plusieurs décennies, d'autres lois de commande, basées sur des approches non linéaires, ont été proposées pour améliorer les performances. Nous les passons en revue dans cette partie.

IV.2.1 Commande classique par PID

La commande des robots manipulateurs industriels est basée le plus souvent sur un schéma de type commande linéaire PID en raison d'une structure décentralisée simple et adaptée

à une implémentation articulation par articulation (un correcteur linéaire implémenté par carte d'axe ou variateur numérique). La commande PID est donnée par l'expression en temps continu

$$\Gamma = k_p e + k_v \dot{e} + k_i \int_0^t e(t) dt \quad (\text{IV.1})$$

Avec ($e = \theta(t)_d - \theta(t)$), l'erreur de position et k_p , k_v , k_i les matrices diagonales de gains proportionnels, dérivés et intégraux, respectivement. La figure IV.1 présente le schéma de principe de la commande PID pour les robots manipulateurs.

D'un point de vue pratique, des règles ont également été proposées pour le choix des différents gains du correcteur. On pourra utiliser un réglage par placement de pôles comme proposé dans [1]. Schéma de principe de la commande PID.

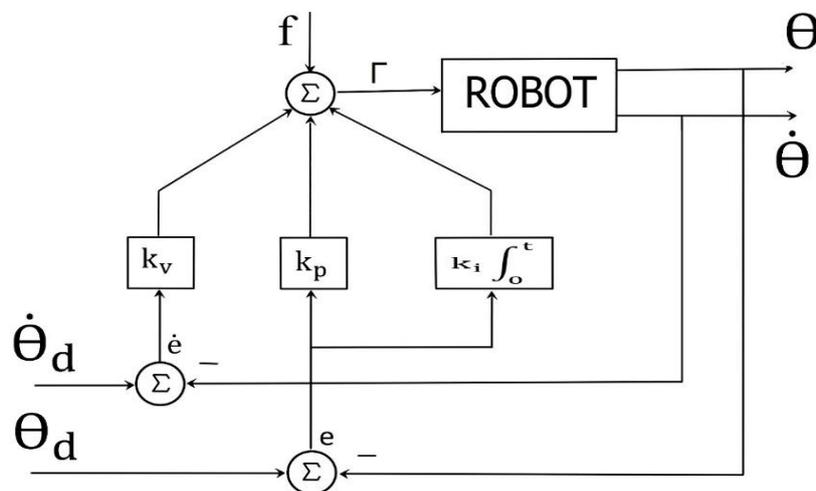


Figure IV.1 schéma de principe de la commande PID

Remarque

Il est clair que l'action intégrale permet d'améliorer la précision statique. Cependant cette action s'avère très dangereuse pour les robots pouvant présenter une structure fermée (auto-collision, ou quelque chose qui vient bloquer la structure mécanique), cela conduira à une erreur statique constante entre l'angle désiré et l'angle réelle, ce qui va provoquer par la suite l'augmentation du coefficient intégrale et par conséquent une augmentation du couple jusqu'à ce que le moteur soit endommager, Voir même abimer le système de commande.

Il existe plusieurs méthodes pour remédier à ça, mais en générale pour éviter cela il existe une alternative qui consiste à utiliser une commande linéaire PD, on utilisera ainsi un grand gain proportionnel pour réduire l'erreur statique.

IV.2.2 Commande PD

Comme énoncé ce peut le régulateur PD est présenté comment étant une alternative aux régulateurs PID, il est régi par :

$$\Gamma = k_p e + k_v \dot{e} \quad (\text{IV.2})$$

Avec ($e = \theta_d(t) - \theta(t)$), l'erreur de position et k_p et k_v sont des matrices diagonales de gains de retour symétriques définies positives, et τ est l'entrée du système. La figure (IV.2) présente le schéma de principe de la commande PD pour les robots manipulateurs.

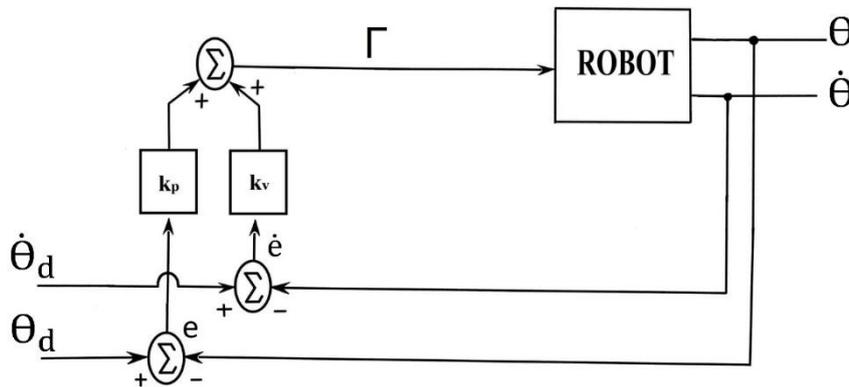


Figure IV.2 schéma de principe de la commande PD

L'application d'une simple loi de commande de type PD est parfois insuffisante. La compensation des non linéarités, notamment le terme le plus influant, à savoir la gravitation, s'avère nécessaire (16). La figure (IV.3) illustre la commande,

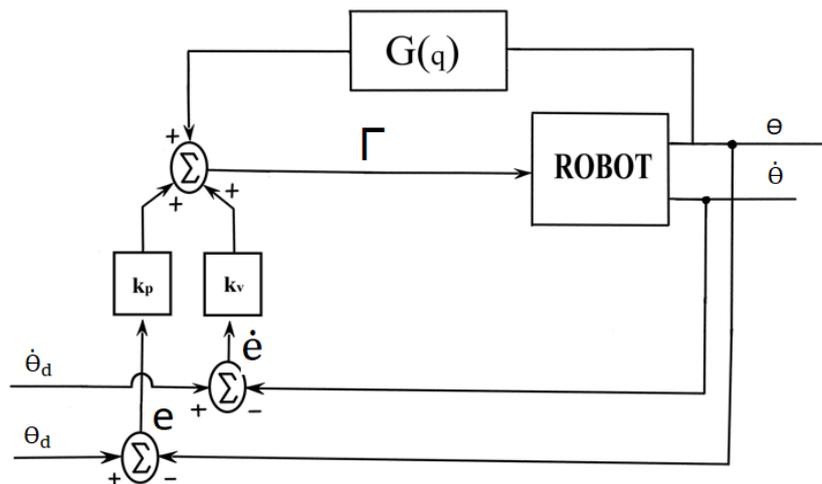


Figure IV.3 Schéma de principe de la commande PD avec compensation de gravité

Bien que ce schéma ne soit plus linéaire pour faire la transition avec les commandes non linéaires. Le terme de compensation est issu de la partie statique du modèle dynamique et peut être calculé à partir des mesures de positions :

$$\Gamma = k_p e + k_v \dot{e} + G(\Theta) \quad (\text{IV.3})$$

Ou à partir des positions articulaires de référence :

$$\Gamma = k_p e + k_v \dot{e} + G(\Theta_d) \quad (\text{IV.4})$$

IV.2.3 Commande découplante linéarisante

Dès lors que les vitesses de parcours de trajectoires sont élevées et que le comportement dynamique du manipulateur ne peut plus être considéré comme quasi-linéaire et découplé entre ses articulations, des lois de commande non linéaires permettent d'améliorer les performances en suivi de trajectoires. Beaucoup d'approches ont été développées depuis quelques décennies. [1], La plupart de ces approches sont basées sur le modèle dynamique du manipulateur.

La commande directement basée sur le modèle dynamique inverse est la commande découplante linéarisant ou encore CTC (Computed Torque Control) qui permet de linéariser et découpler le comportement dynamique du robot par rétroaction. Cette loi de commande s'écrit à partir du modèle dynamique inverse :

$$\Gamma = M(\Theta)v + V(\Theta, \dot{\Theta})\dot{\Theta} + G(\Theta) \quad (\text{IV.5})$$

Cette loi est définie à partir de l'équation suivant :

$$\Gamma = \alpha v + \beta \quad (\text{IV.6})$$

$$\alpha = M(\Theta) \quad , \quad \beta = V(\Theta, \dot{\Theta})\dot{\Theta} + G(\Theta) \quad (\text{IV.7})$$

En remplaçant (IV.6) dans la dynamique du système (IV.5) on aura :

$$\begin{aligned} \Gamma = M(\Theta)v + V(\Theta, \dot{\Theta})\dot{\Theta} + G(\Theta) &= M(\Theta)\ddot{\Theta} + V(\Theta, \dot{\Theta})\dot{\Theta} + G(\Theta) \\ v &= \ddot{\Theta} \end{aligned} \quad (\text{IV.8})$$

Donc l'accélération articulaire est remplacée par une nouvelle entrée de commande, v , telle que le système résultant est un ensemble de n_q systèmes linéaires découplés de type double intégrateur : $v = \ddot{\Theta}$. On peut alors appliquer à la nouvelle entrée v un correcteur linéaire de type PD ou PID avec une anticipation en accélération [1].

Par exemple :

$$v = \ddot{\theta} + k_p e + k_v \dot{e} + k_i \int_0^t e(t) dt \quad (\text{IV.9})$$

Il en résulte l'équation suivante régissant la dynamique de l'erreur pour un modèle dynamique exact et en l'absence de perturbation :

$$\ddot{e} + k_p \dot{e} + k_v e + k_i \int_0^t e(t) dt = 0 \quad (\text{IV.10})$$

Avec : $e = (\theta_d(t) - \theta(t))$, $\dot{e} = (\dot{\theta}_d(t) - \dot{\theta}(t))$, $\ddot{e} = (\ddot{\theta}_d(t) - \ddot{\theta}(t))$,

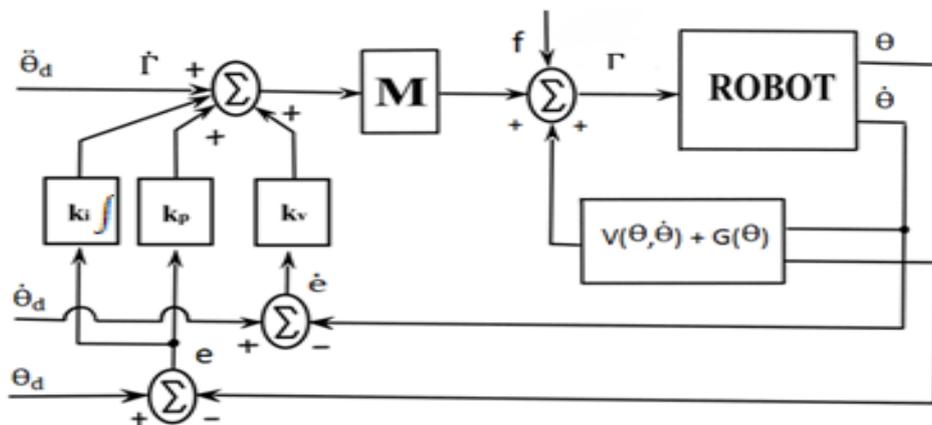


Figure IV.4 Commande dynamique par découplage non linéaire d'un robot manipulateur

IV.3 Systèmes de contrôle cartésien

Une caractéristique fondamentale de l'approche est le processus de conversion de trajectoire, pour calculer les trajectoires articulaires. Le système d'asservissement est basé sur les articulations, comme nous l'avons vu en amont dans ce chapitre.

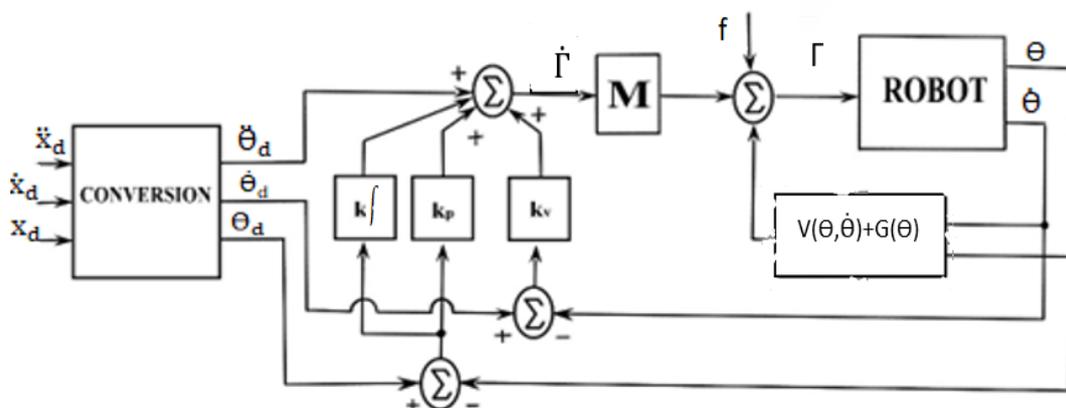


Figure IV.5 Schéma de contrôle basé sur les articulations avec une entrée de chemin cartésien

Le processus de conversion de trajectoire est assez difficile, si cela doit être fait analytiquement [5]. Les calculs qui seraient nécessaires sont :

$$\Theta_d = \text{MGI}(X_d) \quad (\text{IV.11})$$

$$\dot{\Theta}_d = J^{-1}(\Theta)\dot{X}_d \quad (\text{IV.12})$$

$$\ddot{\Theta}_d = j^{-1}(\Theta)\ddot{X}_d + J^{-1}(\Theta)\dot{X}_d \quad (\text{IV.13})$$

IV.4 Commande du robot PUMA560

Dans cette partie nous réalisons des programmes pour simuler les différentes lois de commandes sur le robot PUMA560,

Le robot PUMA 560 est largement utilisé dans les applications académiques et industrielles, il a été modélisé avec plusieurs représentations D-H et des paramètres mesurés légèrement différentes [15][16][117][18]. Ici, nous adoptons le schéma et les paramètres de B. Armstrong [15], avec la configuration et les paramètres indiqués ci-dessous :

Tableau IV-2 masses des liaisons [Kg]

LIAISON	MASSE
m_1	-
m_2	17.40
m_3	4.80
m_4	0.82
m_5	0.35
m_6	0.09

Tableau IV-1 centre de gravité des liaisons [mm]

	G_x	G_y	G_z
LES LIAISON	-	-	-
1	0.068	0.006	-0.016
2	0	-0.070	0.014
3	0	-0.143	0.014
4	0	0	-0.019
5	0	0	0
6	0	0	0.032

Tableau IV-3 moments d'inertie.

LES LIAISONS	I_{xx}	I_{yy}	I_{zz}
1	-	-	0.35
2	0.130	0.524	0.539
3	0.066	0.0125	0.086
4	1.80×10^{-3}	1.80×10^{-3}	1.30×10^{-3}
5	0.3×10^{-3}	0.3×10^{-3}	0.40×10^{-3}
6	0.15×10^{-3}	0.15×10^{-3}	0.04×10^{-3}

Tableau IV-4 Paramètres D-H

I	A_i	θ_i	a_i	d_i
NUMERO DE LA LIAISON	DEGRES	VARIABLE	METRES	METRES
1	0	θ_1	0	0
2	-90	θ_2	0	0
3	0	θ_3	0.4318	-0.0934
4	+90	θ_4	-0.0203	0.4331
5	-90	θ_5	0	0

Les différentes modélisations du robot puma560 sont donnée dans le chapitre 2. La démonstration et le calcul des éléments des matrices M , V et le vecteur G du modèle dynamique de ce bras de robot manipulateur est présenté dans la référence [19].

Simulation :

Dans ce qui suit nous allons tester les performances des méthodes de contrôles que nous avons vus précédemment en poursuite de trajectoire dans l'espace opérationnelle. Cette trajectoire sera générée à partir du MGT suivant les points via que nous avons injecté au préalable à ce dernier. Ces points sont répertoriés dans Tableau IV.5.

Le robot sera ainsi mis dans un environnement dans laquelle il doit faire un aller du point A vers le point B puis revenir sans qu'il touche l'obstacle.

Figure IV.6 illustre le principe.

Une animation du robot sera introduite pour visualiser le suivie de trajectoire. Le programme de ce dernier est présenté en annexe (B.5).

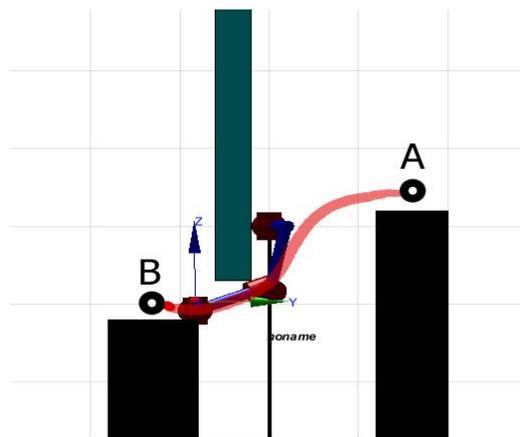


Figure IV.6 illustration 3D du plan de travail

Tableau IV-5 les consignes introduites dans le programme.

X_d	-0.5	-0.1	0.2	0.4	0.6	0.3	0.3	0.6	0.4	0.2	-0.1	-0.5
Y_D	0.7	0.7	0.6	0.4	-0.08	-0.6	-0.6	-0.08	0.4	0.6	0.7	0.7
Z_d	0.2	0.2	0.2	0.2	-0.4	-0.5	-0.5	-0.4	0.2	0.2	0.2	0.2
Temps de passage Entre chaque point	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5

(Nb, que ce soit dans l'espace articulaire ou dans l'espace opérationnelle, les lois de commandes restent les mêmes (voir la figure IV.6). Cependant nous avons choisie de travailler sur l'espace opérationnelle pour voir les performances de ces commandes dans un espace de travail restreint).

IV.4.1 Implémentation de la commande PD :

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

$$\tau = M\ddot{\theta} \quad (IV.14)$$

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 force gravitationnelles $G(\theta)$ sont complètement négligés.

Structure de la commande

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

$$e = (\theta_d(t) - \theta(t)) \quad \dot{e} = (\dot{\theta}_d(t) - \dot{\theta}(t)) \quad (IV.15)$$

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 l'équation (IV-11) En utilisant une loi de commande PD :

$$\Gamma = k_p e + k_v \dot{e} \quad (IV.16)$$

$$\text{Ce qui revient à : } \Gamma_1 = k_{p1} e_1 + k_{v1} \dot{e}_1 \quad \Gamma_2 = k_{p2} e_2 + k_{v2} \dot{e}_2 \quad \Gamma_3 = k_{p3} e_3 + k_{v3} \dot{e}_3 \quad (IV.17)$$

Pour avoir la dynamique désirée, il suffit de bien placer les pôles en ajustant les valeurs des paramètres (k_p , k_v) du régulateur. En robotique, on évite les réponses oscillatoires, et pour cela on choisit généralement la valeur critique du rapport d'amortissement $\xi=1$. Ce qui donne un pôle double [13]. En choisissant une pulsation $w_n = 25$ rad/sec, on calcule :

$$k_v = 50 \text{ et } k_p = 625. \quad \text{Avec : } k_v = 2\xi w_n, \quad k_p = w_n^2$$

La pulsation w_n est choisie le plus grand possible, toutefois elle ne devra pas être supérieure à la pulsation de résonance w_{rn} correspondant aux modes de vibration mécanique, afin de ne pas déstabiliser le système. Une valeur $w_n = w_{rn}/2$ représente un bon compromis. [14].

Nous obtiendrons ainsi les lois de commandes suivantes :

$$\Gamma_1 = 625 e_1 + 50 \dot{e}_1 \quad \Gamma_2 = 625 e_2 + 50 \dot{e}_2 \quad \Gamma_3 = 625 e_3 + 50 \dot{e}_3 \quad (IV.18)$$

$$\text{Avec : } e = (\theta_d(t) - \theta(t)) \quad \text{et} \quad \dot{e} = \dot{\theta}_d(t) - \dot{\theta}(t)$$

Simulation et résultats :

- ⇒ L'organigramme de commande est donné dans l'annexe (B.0).
- ⇒ Les résultats de simulation sont illustrés par la Figure (IV.7).

Interprétation des résultats :

Les figures nous montrent bien un suivi de trajectoire mais qui présente quand-même des erreurs de suivi non négligeable entre les consignes désirée et réelles, si on prend ainsi exemple sur la figure qui représente une illustration 3d du robot, on voit bien que celui-ci ne suit pas exactement le chemin imposé (en Blue). Cela est dû, a la proche du régulateur PD qui ne prend pas en compte le model dynamique du robot.

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

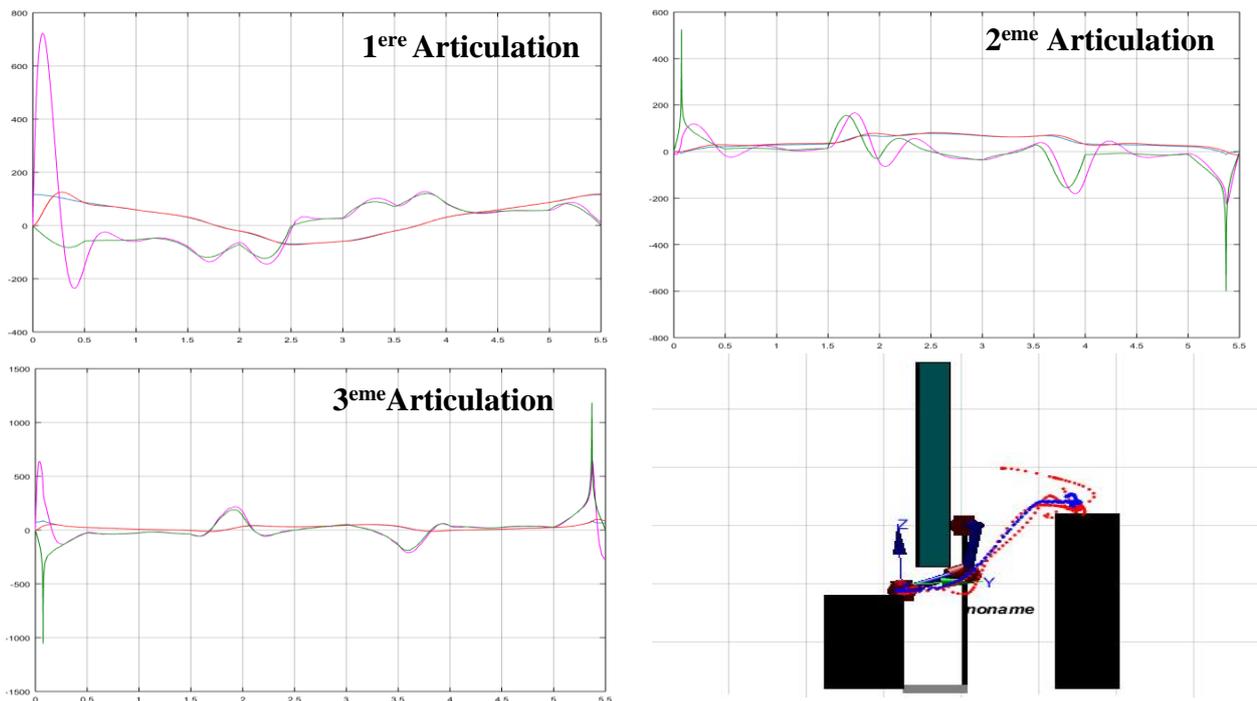


Figure IV.7 résultats de simulation de la commande PD

(Positions désirées (en bleu), réelles (en rouge), vitesses désirées (en vert), réelles (en violet))

IV.4.2 Implémentation de la commande PD avec compensation de gravité

La loi de commande est donnée par :

$$\Gamma = k_p e + k_v \dot{e} + G(\theta) \quad (\text{IV.19})$$

Avec $G(\theta)$ est la force gravitationnelle

$$g_1 = 0 \quad (\text{IV.20})$$

$$g_2 = -37.2 c_2 - 8.4 s_{23} + 1.02 s_2$$

$$g_3 = -8.4 s_{23} + 0.25 c_{23}$$

On aura donc :

$$\begin{aligned}\Gamma_1 &= 625 e_1 + 50 \dot{e}_1 + G(\theta_1) & \Gamma_2 &= 625 e_2 + 50 \dot{e}_2 + G(\theta_2) \\ \Gamma_3 &= 625 e_3 + 50 \dot{e}_3 + G(\theta_3)\end{aligned}\quad (\text{IV.21})$$

Simulation et résultats

- ⇒ L'organigramme de commande est donné dans l'annexe (B.1).
- ⇒ Les résultats de simulation sont illustrés par la Figure IV.8.

Interprétation des résultats :

Grâce à la compensation des effets perturbateurs, à savoir la gravité (l'effet le plus influent), le régulateur PD nous présente ici une meilleure performance que la précédente, avec un suivi de trajectoire qui est quasi en superposition avec celle désirée.

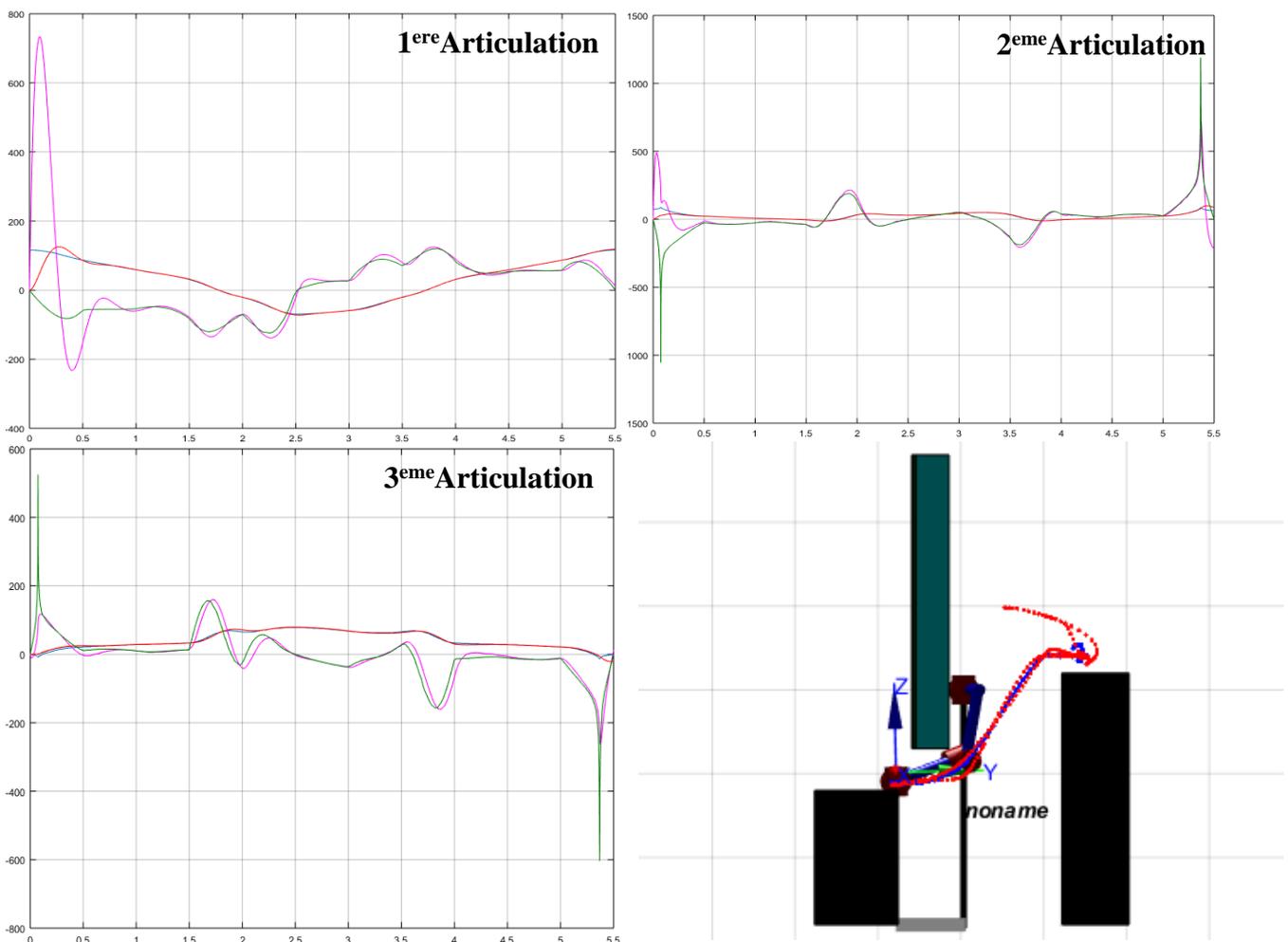


Figure IV.8 résultats de simulation de la commande PD avec compensation de gravité
(Positions désirées (en bleu), réelles (en rouge), vitesses désirées (en vert), réelles (en violet))

IV.4.3 Implémentation de la commande PID

Lors de la partie précédente, on a supposé que le robot ne soit 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.

Structure de la commande :

La loi de commande PID est de la forme

$$\tau_i = k_{vi}\dot{e}_{pi} + k_{pi}e_i + k_{Ii} \int_0^t e_i(t) dt = M_i \ddot{\theta} \quad (IV.22)$$

Par application de la transformée de Laplace on a :

$$M_i p^3 + k_{vi}p^2 + k_{pi}p + k_{Ii} = 0 \quad (IV.23)$$

On prend $M_i = 1$ (les versions les plus simples de commande n'utilisent aucun paramètre de système, approche non basée model) :

$$p^3 + k_{vi}p^2 + k_{pi}p + k_{Ii} = 0 \quad (IV.24)$$

L'équation (IV.24) est une équation d'ordre 3. Pour éviter les réponses oscillatoires, on prend le facteur d'amortissement $\xi=1$:

$$(p - p_3)(p^2 + 2\xi w_n p + w_n^2) \quad (IV.25)$$

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

$$p_1 = p_2 = -5 \quad (IV.26)$$

(Nb, en choisissant une pulsation de 25 rad/s la réponse de système a été instable, donc on a opté pour une pulsation de 5 rad/s).

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

$$(P+25)(P+10P+25) = 0 \quad (IV.27)$$

$$p^3 + 35p^2 + 275p + 625 = 0$$

Par comparaison avec (IV.22) et identification : $k_v = 35$, $k_p = 275$, $k_i = 625$, cependant après simulation de ces paramètres la réponse du système présentait quelques oscillations. Donc on a décidé d'augmenter le gain k_v à 100 afin de réduire leurs effets :

$$\begin{aligned} \tau_1 &= 100\dot{e}_1 + 275e_1 + 625 \int_0^t e_1(t) dt & \tau_2 &= 100\dot{e}_2 + 275e_2 + 625 \int_0^t e_2(t) dt \\ \tau_3 &= 100\dot{e}_3 + 275e_3 + 625 \int_0^t e_3(t) dt \end{aligned} \quad (IV.28)$$

Avec : $e = (\theta_d(t) - \theta(t))$, $\dot{e} = (\dot{\theta}_d(t) - \dot{\theta}(t))$

Simulation et résultats :

- ⇒ L'organigramme de commande est donné dans l'annexe (B.2).
- ⇒ Les résultats de simulation sont illustrés par la Figure IV.9.

Interprétation des résultats :

D'après les résultats de simulation, on constate de bien meilleures performances qu'à celle donné par le régulateur PD, car l'erreur de suivi du contrôleur PID est plus petite. Cependant cette réponse n'est pas tous a fait satisfaisante compte tenu du contexte dans laquelle le robot doit être le plus précis possible afin d'éviter les collisions avec les obstacles.

Mais on pourra toujours améliorer cette réponse en ajustant les matrices k_p , k_v et k_i

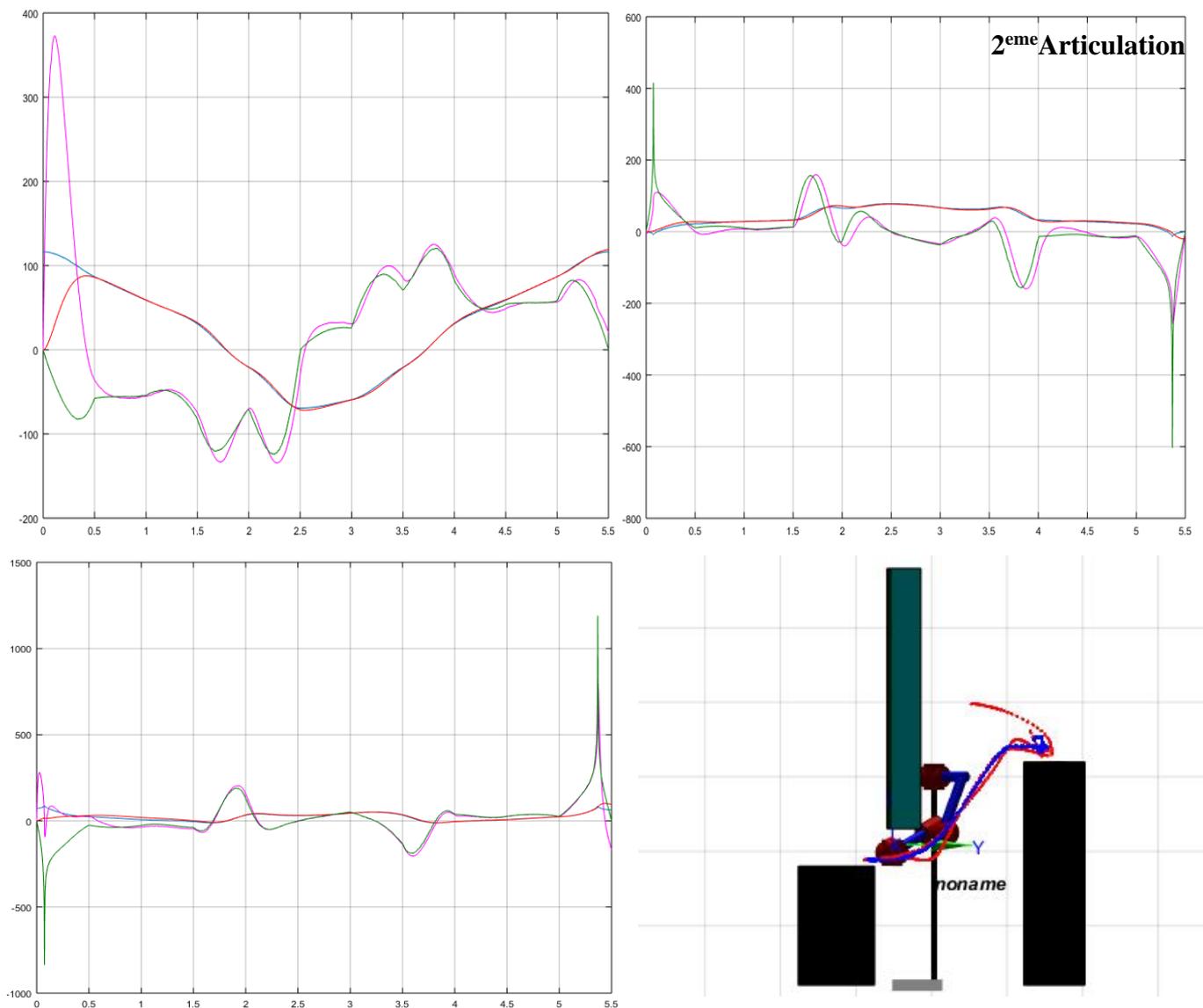


Figure IV.9 résultats de simulation de la commande PID

(Positions désirées (en bleu), réelles (en rouge), des vitesses désirées (en vert), réelles (en violet))

IV.4.4 Implémentation de la commande CTC

La commande CTC, prend en compte le modèle complet (réel) du robot, sa méthode est basée sur la linéarisation et le découplage du système. La linéarisation est assurée par le choix de la loi de commande, en utilisant une commande par partition afin de commander chaque articulation séparément. La figure (IV.4) illustre la structure de la commande.

Structure de la commande :

Sois la commande linéarisant suivante :

$$\Gamma = M(\theta)v + V(\theta, \dot{\theta})\dot{\theta} + G(\theta) \quad (IV.29)$$

Remplacé dans la dynamique du robot l'équation (II.26) donne le système linéarisé suivant :

$$\ddot{\theta} = v \quad (IV.30)$$

On considère maintenant le choix suivant de v :

$$v = \ddot{\theta}_d + k_d \dot{e} + k_p e \quad (IV.31)$$

Où

$$v = \ddot{\theta}_d + k_d \dot{e} + k_p e + k_i \int_0^t e dt$$

La dynamique résultante en boucle fermée s'écrit donc :

$$\ddot{\theta} = v = \ddot{\theta}_d + k_d \dot{e} + k_p e \quad (IV.32)$$

$$\ddot{\theta} = v = \ddot{\theta}_d + k_d \dot{e} + k_p e + k_i \int_0^t e dt$$

Régulation PD :

On choisira la même pulsation $\omega_n = 25 \text{ rad/sec}$, Ce qui donne :

$$\Gamma_1 = \ddot{\theta}_{d1} + 625 e_1 + 50 \dot{e}_1 \quad (IV.33)$$

$$\Gamma_2 = \ddot{\theta}_{d2} + 625 e_2 + 50 \dot{e}_2$$

$$\Gamma_3 = \ddot{\theta}_{d3} + 625 e_3 + 50 \dot{e}_3$$

Régulation PID :

Pareille ici on reprendra les mêmes paramètres utilisés dans la section PID :

$$\tau_1 = \ddot{\theta}_{d1} + 275 e_1 + 100 \dot{e}_2 + 625 \int_0^t e_1(t) dt \quad (IV.34)$$

$$\tau_2 = \ddot{\theta}_{d1} + 275 e_2 + 100 \dot{e}_2 + 625 \int_0^t e_2(t) dt$$

$$\tau_3 = \ddot{\theta}_{d3} + 275 e_3 + 100 \dot{e}_3 + 625 \int_0^t e_2(t) dt$$

Simulation et résultats :

⇒ La commande PID_CTC sera simulée sous l'effet des perturbations.

⇒ L'organigramme de commande PD_CTC est donné dans l'annexe (B.3).

⇒ L'organigramme de commande PID_CTC est donné dans l'annexe (B.4).

⇒ Les résultats de simulation sont illustrés par Figures (IV.10 et 11)

Interprétation des résultats :

Les deux approches PD_CTC et PID_CTC ont donné pratiquement les mêmes résultats, avec un suivi de trajectoires en parfaite superposition avec les trajectoires désirées.

PD_CTC :

PID_CTC :

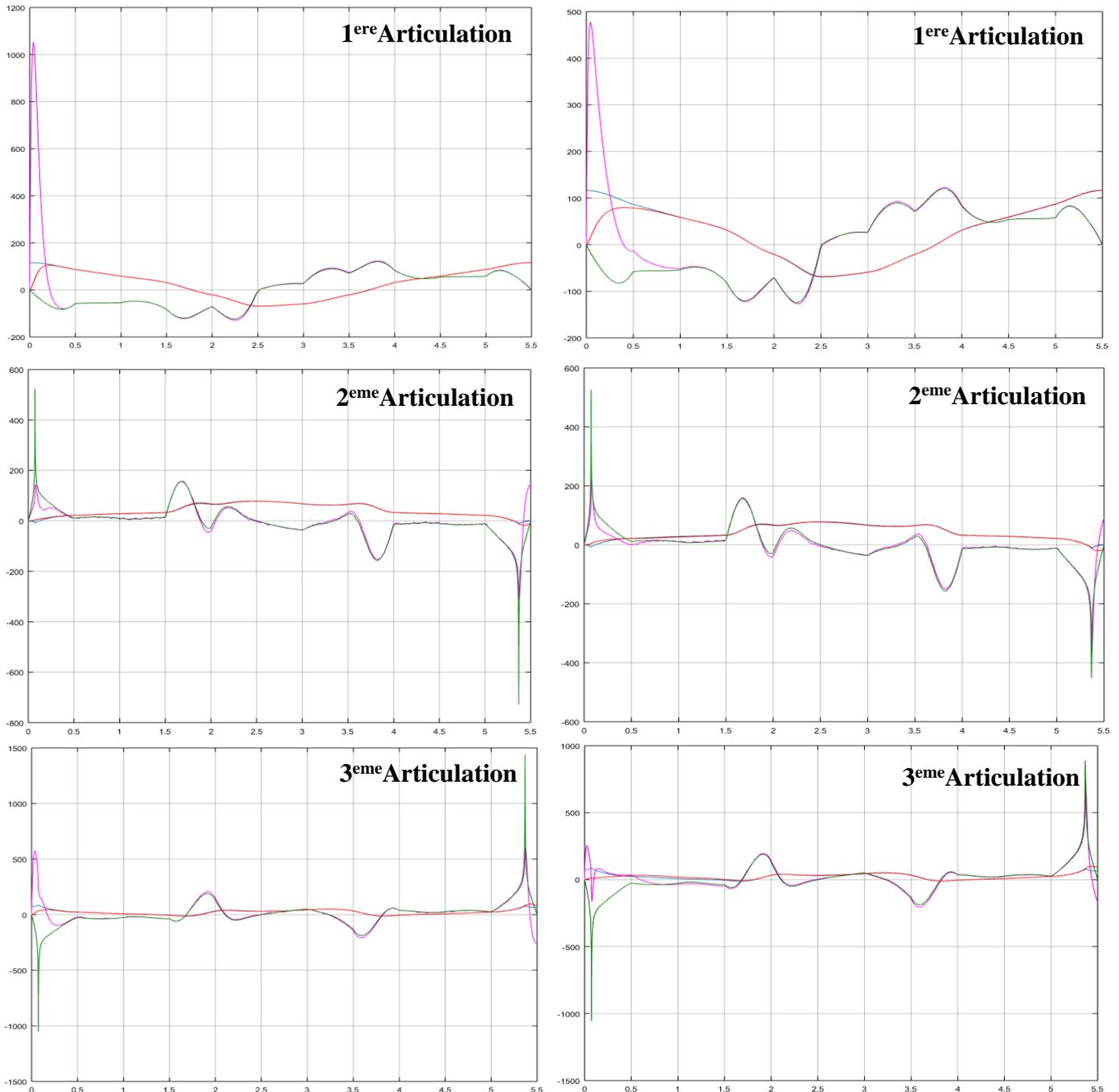


Figure IV.10 Forme des trajectoires pour les trois articulations (PD_CTC) et (PID_CTC): positions désirées (en bleu), réelles en (rouge), vitesses désirées (vert), réelles (en violet).

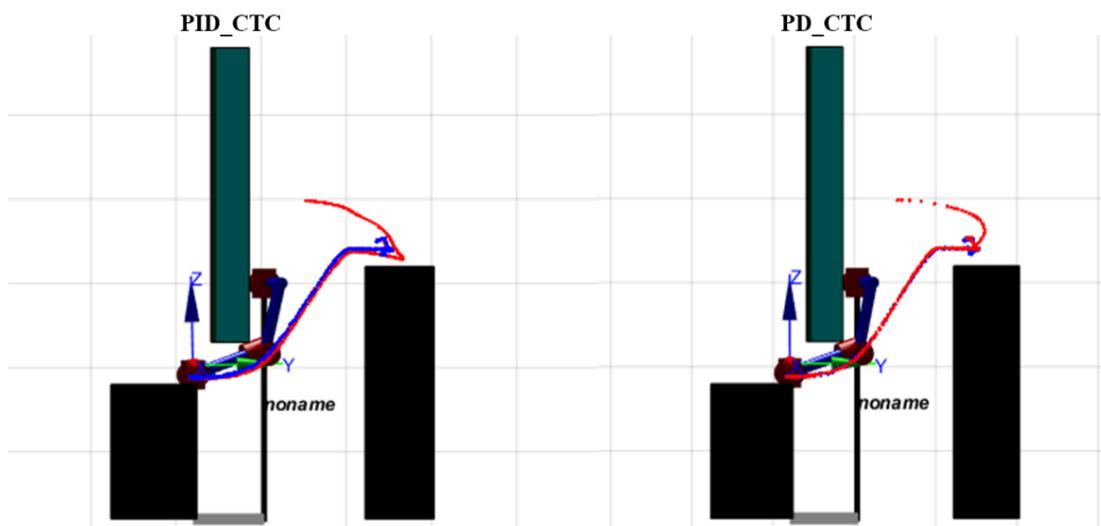


Figure IV.11 illustration 3D de la trajectoire désiré (en Blue) et de la trajectoire réelle (en rouge)

IV.5 Conclusion

Ce chapitre est consacré à la présentation des lois de commande linéaires et non linéaires. On en dénombrera la commande PD et la commande PID pour celles qui ont une approche linéaire (qui ne sont pas basées sur le modèle non linéaire) et commande PD avec compensation de gravité et les commandes à couple calculé pour les commandes non linéaires (basées sur le modèle dynamique non linéaire).

Des simulations ont été présentées pour tester les performances de ces commandes. Les résultats montrent la supériorité de la commande CTC. Cependant nos simulations ont été effectuées dans le cadre où les paramètres du système étaient bien définis et parfaitement connus ce qui est rarement le cas dans la réalité.

En résumé, on pourra dire que les performances des commandes CTC et PD avec compensation de gravité, dépendent fortement du modèle dynamique du robot. Tant dis que les performances des commandes PD et PID restent valables autour du point de linéarisation du modèle.

Conclusion générale

Ce mémoire présente notre travail concernant la modélisation et la commande en suivi de trajectoires des robots manipulateurs. Dans un premier temps, on a donné un aperçu sur les robots manipulateurs, leurs éléments constitutifs, leurs utilisations, ainsi que quelques définitions, afin de mieux appréhender la suite du la suite du mémoire.

Le deuxième chapitre illustre les notions de base de la modélisation pour les robots manipulateurs. Notamment les modèles géométriques, cinématiques et dynamiques. Parallèlement, une étude de modélisation sur le robot PUMA 560 a été ainsi faite. Le but étant de disposer d'un modèle de simulation le plus fidèle possible du point de vue du comportement mécanique, et de l'utiliser par la suite pour lala commande de ce dernier.

Dans le chapitre trois, les méthodes de planification à savoir la méthode du polynôme cubique a été ainsi exposée, que ce soit dans l'espace articulaire ou dans l'espace opérationnel, les deux ont été implémentés. Une simulation de ces méthodes a été effectuée sur le robot PUMA 560, afin de visualiser les différents résultats.

La problématique du suivi de trajectoire ainsi qu'une revue de littérature succincte sur les principales méthodes de commande des robots manipulateurs ont été exposées dans le chapitre quatre. Parmi ces méthodes, on a vu les commandes classiques (à savoir la commande PID et PD) et les commandes non linéaires telles que la commande CTC et la commande PD avec compensation de gravité. Les résultats de simulation de ces commandes sur le PUMA560 ont permis d'avoir une idée claire sur leurs performances et leurs caractéristiques, afin de confirmer par la suite leurs intérêts et leurs efficacités sur le domaine de la commande des robots.

Ce projet nous a permis de consolider les connaissances acquises dans le domaine de la robotique et de la commande. On espère que ce travail sera poursuivi en étudiant les autres aspects de ce sujet, tels que : l'implémentation sur un bras réelle, l'exploration des autres techniques de commande plus récentes basées sur l'intelligence artificielle.

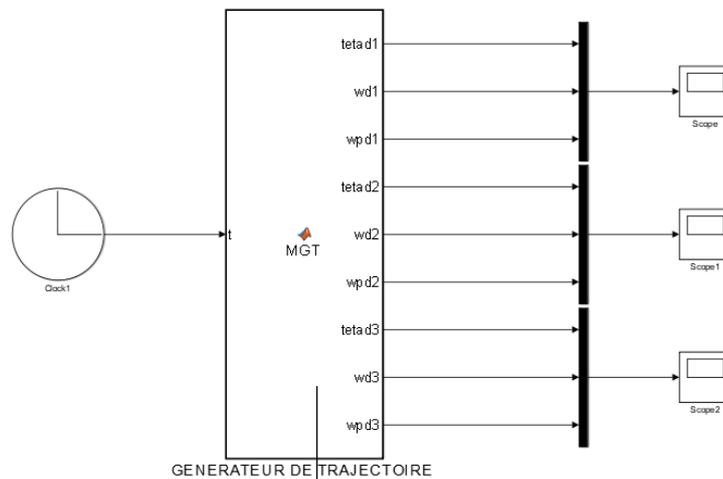
Bibliographie

- [1]. Hocine HAMDI, " *INTRODUCTION A LA ROBOTIQUE* ", Les éditions de l'université Mentouri_Constantine 2002-2000.
- [2]. Thi Van Anh Nguyen, " *Commande de robots manipulateurs basée sur le modèle de Takagi-Sugeno : nouvelle approche pour le suivi de trajectoire* ", Thèse de doctorat, université polytechnique HAUTS-DE-France, Valenciennes le 04/10/2019,
- [3]. M. S. BOUTAANI, " *Robotique Industrielle* " support de cours Université Abderrahmane mira Bejaia, 2018.
- [4]. E. DOMBRE et W. KHALIL, " *Bases de la modélisation et de la commande des robots-manipulateurs de type série* " Université numérique ingénierie et technologie(UNIT), France , 08 mai 2012.
- [5]. Graig, Jon. " *Introduction to Robotics Mechanics and Control. s.l. : Pearson.Printice Hall, 2005*". 3eme edition.
- [6]. Documentation Corporation, Intel, " *Comment la robotique façonne le monde d'aujourd'hui*" <https://www.intel.fr/content/www/fr/fr/robotics/types-and-applications.html>. [consulté le 04/05/2022.]
- [7]. P. Mayé, " *Moteurs électriques pour la robotique*", DUNOD, 3e édition. 2016.
- [8]. " *Robot à structure cartésienne* " dans Le Grand Dictionnaire terminologique. <https://gdt.oqlf.gouv.qc.ca>. [consulté le 29/ 04/2022]
- [9]. " *Robot à structure cylindrique* " dans Le Grand Dictionnaire terminologique. <https://gdt.oqlf.gouv.qc.ca>. [consulté le 29/04/2022]
- [10]. " *Robot à structure sphériques* " dans Le Grand Dictionnaire terminologique. <https://gdt.oqlf.gouv.qc.ca>. [consulté le 29/04/2022]]
- [11]. N ACHERCHOUR et A. ARDJOUNE, " *Commande floue dynamique d'un robot manipulateur*". Mémoire de Master, Université de Bejaia, 2017.
- [12]. R. Merzouki, " *Robotique Industrielle. Robotique de Manipulation* " , support de cours École polytechnique universitaire de Lille , 2010.
- [13]. B. Mendil, " *Modélisation dynamique complète d'un bras de robot*", Notes de cours, Université de Bejaia, 2016.

-
- [14]. W.Khalil. " *Modélisation et commande des robots. Paris* " Édition Hermès, 1988.
- [15]. B. Armstrong, O. Khatib, & J. Burdick, " *The Explicit Dynamic Model and Inertial Parameters of the PUMA 566 Arm*", in Proc. International Conference of Robotics and Automation, vol. 1, Washington, USA, pp. 510—18, 1986.
- [16]. R. P. Paul, M. Rongand, and H. Zhang, " *The dynamics of the PUMA manipulator,* " Proc. American Control Conference, San Fransisco, pp 491 - 496, June 22 - 24, 1983
- [17]. P. I. Corke and B Armstrong-Helouvry, " *A meta-study of PUMA 560 dynamics: A critical appraisal of literature data,* " Robotica, Vol. 13, pp253-258, 1995
- [18]. J. M. Hollebrach, " *RecursiveLagrangianFormulationof Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity,* " IEEETRANSACTIONSON SYSTEMS,MAN, AND CYBERNETICS,VOL. SMC-10, pp730-736,NO. 1,NOVEMBER 1980
- [19]. John J. Murray et Charles P. Neuman, " *AN ALGEBRAIC ROBOT DYNAMIC MODELING PROGRAM* ", L'université de Carnegie Mellon, 1987.
- [20]. Peter Cork , " *ROBOTICS TOOLBOX* ", [En ligne], Disponible sur <https://petercorke.com/toolboxes/robotics-toolbox>, (Consulté le 15/05/2022).

Annexes A

A.0 fonction polynôme cubique pour un mouvement entre deux points



```
Function [tetad1, wd1, wpd1, tetad2, wd2, wpd2, tetad3, wd3, wpd3]= MGT(t)
```

```
T1_zero=0; T1_finale=100; T2_zero=0; T2_finale=90; T3_zero=0;
T3_finale=60;
```

```
%=====
```

```
Tf=2; Tf2=Tf*Tf; Tf3=Tf*Tf*Tf;
```

```
%=====
```

```
a0=T1_zero; a1=0;a2=3*(T1_finale-T1_zero)/Tf2;
a3=-2*(T1_finale-T1_zero)/Tf3;
```

```
a20=T2_zero;a21=0;a22=3*(T2_finale-T2_zero)/Tf2;
a23=-2*(T2_finale-T2_zero)/Tf3;
```

```
a30=T3_zero;a31=0;a32=3*(T3_finale-T3_zero)/Tf2;
a33=-2*(T3_finale-T2_zero)/Tf3;
```

```
%=====
```

```
tetad1= (a0+a1.*t+a2.*t.^2+a3.*t.^3)*pi/180;
```

```
wd1= (a1+2*a2.*t+3*a3.*t.^2)*pi/180;
```

```
wpd1= (2*a2+6*a3.*t)*pi/180;
```

```
tetad2= (a20+a21.*t+a22.*t.^2+a23.*t.^3)*pi/180;
```

```
wd2= (a21+2*a22.*t+3*a23.*t.^2)*pi/180;
```

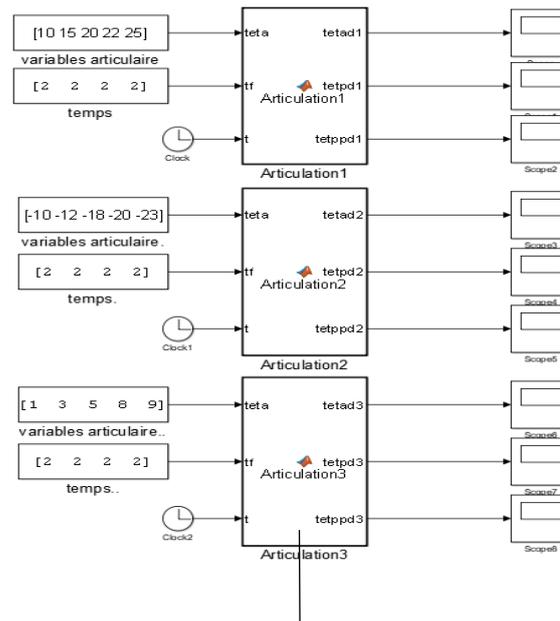
```
wpd2= (2*a22+6*a23.*t)*pi/180;
```

```
tetad3= (a30+a31.*t+a32.*t.^2+a33.*t.^3)*pi/180;
```

```
wd3= (a31+2*a32.*t+3*a33.*t.^2)*pi/180;
```

```
wpd3= (2*a32+6*a33.*t)*pi/180;
```

A.1 fonction polynôme cubique pour un mouvement avec points via



```

function [tetad3, tetpd3, tetppd3]= Articulation3(teta,tf,t)

Np=length (teta); Ns=Np-1;
tetaD= teta(2:Np);tetaG=teta(1:Np-1);
Pnt= (tetaD-tetaG). /tf;
Tetp=zeros (1, Np);

tfs= 0*tf;
tfs(1)= tf(1);
for i=2:Ns tfs(i)=tfs(i-1)+tf(i);
end
for i=2:(Ns)
    if sign(Pnt(i-1))==sign(Pnt(i)) tetp(i)=(Pnt(i)+Pnt(i-1))/2;
    end
end

tetpD=tetp(2:Np);    tetpG=tetp(1:Np-1);

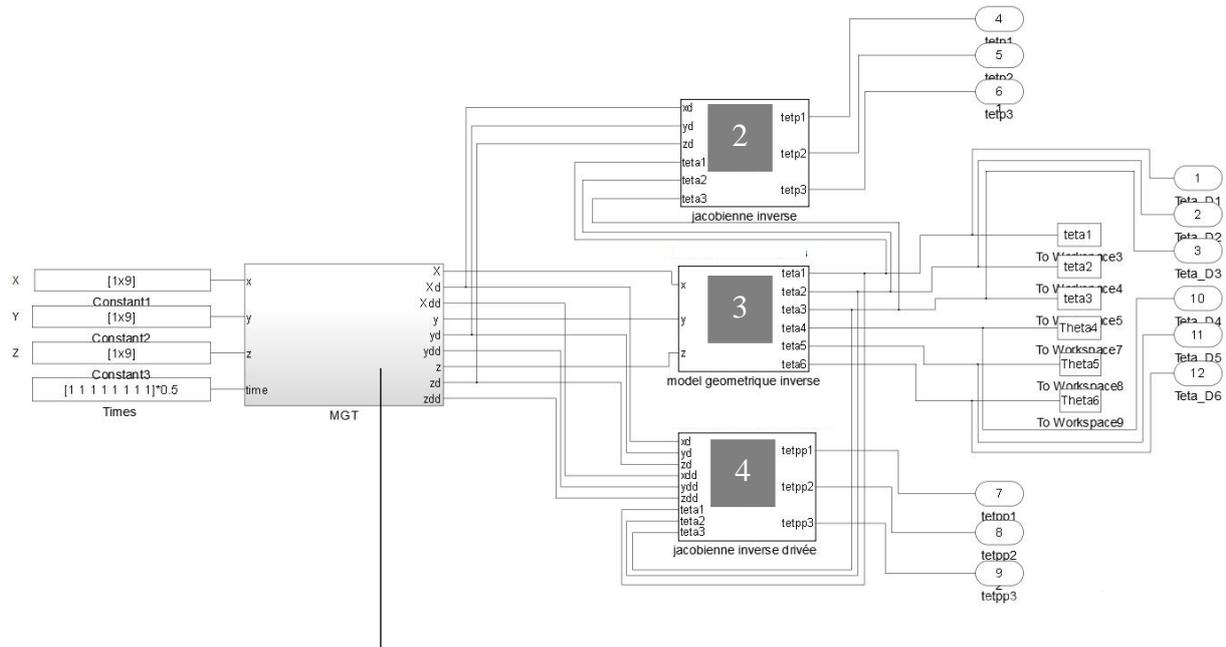
Prm=zeros (4, Ns);
Prm(1,:)= teta(1:Ns); Prm (2,:) = tetp(1:Ns);
tf2= tf.^2;tf3= tf.^3;
Prm(3,:)= 3*(tetaD-tetaG)./tf2 - 2*tetpG'./tf - tetpD'./tf;
Prm (4,:)=-2*(tetaD-tetaG)./tf3 + (tetpG + tetpD)'./tf2;

for j=2:Ns
    if (( tfs(j-1)<t)&& (t<= tfs(j)))
        s=j;
    end
end
tsG= [0 tfs(1:Ns-1)'];
tetad3= Prm(1,s)+Prm(2,s)*(t-tsG(s))+Prm(3,s)*(t-
tsG(s))^2+Prm(4,s)*(t-tsG(s))^3;
tetpd3= Prm(2,s)+2*Prm(3,s)*(t-tsG(s))+3*Prm(4,s)*(t-tsG(s))^2;
tetppd3= 2*Prm(3,s)+6*Prm(4,s)*(t-tsG(s));

end

```

A.2 Génération de mouvement dans l'espace opérationnel

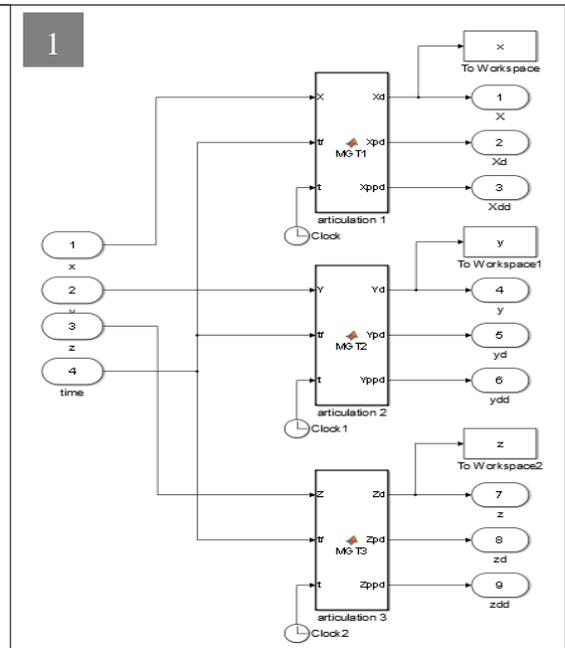


1 Génération de trajectoires

```

function [Xd,Xpd,Xppd]= MGT1(X,tf,t)
Np=length(X);Ns=Np-1;
X_D=X(2:Np);X_G=X(1:Np-1);
Pnt=(X_D-X_G)./tf;
Xp=zeros(1,Np);
tfs=0*tf;
tfs(1)=tf(1);
for i=2:Ns tfs(i)=tfs(i-1)+tf(i);
end
for i=2:(Ns)
    if sign(Pnt(i-1))==sign(Pnt(i))
        Xp(i)=(Pnt(i)+Pnt(i-1))/2;
    end
end
Xp_D=Xp(2:Np);    Xp_G=Xp(1:Np-1);
Prm=zeros(4,Ns);
Prm(1,:)=X(1:Ns); Prm(2,:)=Xp(1:Ns);
tf2=tf.^2;tf3=tf.^3;
Prm(3,:)=3*(X_D-X_G)./tf2 - 2*Xp_G'./tf - Xp_D'./tf;
Prm(4,:)=-2*(X_D-X_G)./tf3 + (Xp_G + Xp_D)'./tf2;
for j=2:Ns
    if (( tfs(j-1)<t)&& (t<= tfs(j)))
        s=j;
    end
end
tsG= [0 tfs(1:Ns-1)'];
Xd= Prm(1,s)+Prm(2,s)*(t-tsG(s))+Prm(3,s)*(t-tsG(s))^2+Prm(4,s)*(t-tsG(s))^3;
Xpd= Prm(2,s)+2*Prm(3,s)*(t-tsG(s))+3*Prm(4,s)*(t-tsG(s))^2;
Xppd= 2*Prm(3,s)+6*Prm(4,s)*(t-tsG(s));
end

```



2 Modèle cinématique inverse

```

function [tetp1,tetp2,tetp3]= fcn(xd,yd,zd,teta1,teta2,teta3)

Xd = [xd,yd,zd]';

a2 = 0.4318 ; a3 = -0.0203 ;
d2 = 0.2435 ; d3 = -0.0934 ; d4 = 0.4331 ; d6 = 0 ;
b2 = 0.2435 ; b3 = -0.0934 ; b4 = 0.4331 ; b6 = 0 ;

J =[-sin(teta1)*(a2*cos(teta2)+a3*cos(teta2+teta3))-(d2+d3)*cos(teta1)-
(d4+d6)*sin(teta1)*sin(teta2+teta3),-
cos(teta1)*(a2*sin(teta2)+a3*sin(teta2+teta3))+(d4+d6)*cos(teta1)*cos(teta2
+teta3),-
a3*cos(teta1)*sin(teta2+teta3)+(d4+d6)*cos(teta1)*cos(teta2+teta3);
cos(teta1)*(a2*cos(teta2)+a3*cos(teta2+teta3))-
(d2+d3)*sin(teta1)+(d4+d6)*cos(teta1)*sin(teta2+teta3),-
sin(teta1)*(a2*sin(teta2)+a3*sin(teta2+teta3))+(d4+d6)*sin(teta1)*cos(teta2
+teta3),-
a3*sin(teta1)*sin(teta2+teta3)+(d4+d6)*sin(teta1)*cos(teta2+teta3);
0,-(a2*cos(teta2)+a3*cos(teta2+teta3))-(d4+d6)*sin(teta2+teta3),-
a3*cos(teta2+teta3)-(d4+d6)*sin(teta2+teta3)];

Qp = (J)^-1 * Xd;

tetp1 = Qp(1);
tetp2 = Qp(2);
tetp3 = Qp(3);

```

3 Modèle géométrique inverse

```

function [teta1,teta2,teta3,teta4,teta5,teta6]= fcn(x,y,z)

A = [-90 0 90 -90 90 0] ;
r = [0 0.431 -0.020 0 0 0];
d = [0 0.139 -0.0934 0.431 0 0.056];

xx = x ; yy = y ; zz = z ;

T0_6 = [1 0 0 xx;
0 1 0 yy;
0 0 1 zz;
0 0 0 1];

P = [xx-0.056*0; yy-0.056*0; zz-0.056*1];

C1 = sqrt(P(1)^2+P(2)^2);
C2 = P(3)-d(1);
C3 = sqrt(C1^2+C2^2);
C4 = sqrt(r(3)^2+d(4)^2);
D1 = d(2)/C1;
D2 = (C3^2+r(2)^2-C4^2)/(2*r(2)*C3);
D3 = (r(2)^2+C4^2-C3^2)/(2*r(2)*C4);

a1 = atan2d(D1,sqrt(abs(1-D1^2)));

```

```

a2 = atan2d(sqrt(abs(1-D2^2)),D2);
b = atan2d(sqrt(abs(1-D3^2)),D3);
p1 = atan2d(P(2),P(1));
p2 = atan2d(C2,C1);

J = [p1-a1 round(a2-p2) round(b-90)];

teta1 = J(1)*(pi/180);
teta2 = J(2)*(pi/180);
teta3 = J(3)*(pi/180);

T01 = [cos(teta1) 0 -sin(teta1) 0 ; sin(teta1) 0 cos(teta1) 0 ; 0 -1 0
0 ; 0 0 0 1];
T12 = [cos(teta2) -sin(teta2) 0 a2*cos(teta2) ; sin(teta2) cos(teta2) 0
a2*sin(teta2) ; 0 0 1 d(2) ; 0 0 0 1];
T23 = [cos(teta3) 0 sin(teta3) r(3)*cos(teta3) ;
sin(teta3) 0 -cos(teta3) r(3)*sin(teta3) ; 0 1 0 0 ; 0 0 0 1];

T0_3 = T01 * T12 * T23 ;

T3_6 = inv(T0_3)*T0_6;

teta4 = (round(atan2d(T3_6(2,3),T3_6(1,3))))*(pi/180);
teta5 = (round(atan2d(sqrt(abs(1-T3_6(3,3)^2)),T3_6(3,3))))*(pi/180);
teta6 = (atan2d(T3_6(3,2),-T3_6(3,1)))*(pi/180);

```

4

Dérivée du modèle cinématique inverse

```

function [tetpp1,tetpp2,tetpp3]=
fcn(xd,yd,zd,xdd,ydd,zdd,teta1,teta2,teta3)

Xd = [xd,yd,zd]';
Xdd = [xdd,ydd,zdd]';

a2 = 0.4318 ; a3 = -0.0203 ;
d2 = 0.2435 ; d3 = -0.0934 ; d4 = 0.4331 ; d6 = 0 ;
b2 = 0.2435 ; b3 = -0.0934 ; b4 = 0.4331 ; b6 = 0 ;

J = [-sin(teta1)*(a2*cos(teta2)+a3*cos(teta2+teta3))-(d2+d3)*cos(teta1)-
(d4+d6)*sin(teta1)*sin(teta2+teta3),-
cos(teta1)*(a2*sin(teta2)+a3*sin(teta2+teta3))+(d4+d6)*cos(teta1)*cos(teta2
+teta3),-
a3*cos(teta1)*sin(teta2+teta3)+(d4+d6)*cos(teta1)*cos(teta2+teta3);
cos(teta1)*(a2*cos(teta2)+a3*cos(teta2+teta3))-
(d2+d3)*sin(teta1)+(d4+d6)*cos(teta1)*sin(teta2+teta3),-
sin(teta1)*(a2*sin(teta2)+a3*sin(teta2+teta3))+(d4+d6)*sin(teta1)*cos(teta2
+teta3),-
a3*sin(teta1)*sin(teta2+teta3)+(d4+d6)*sin(teta1)*cos(teta2+teta3);
0,-(a2*cos(teta2)+a3*cos(teta2+teta3))-(d4+d6)*sin(teta2+teta3),-
a3*cos(teta2+teta3)-(d4+d6)*sin(teta2+teta3)];
Qpp = (J)^-1 * Xd + (J)^-1 * Xdd;
tetpp1 = Qpp(1);
tetpp2 = Qpp(2);
tetpp3 = Qpp(3);

```

A.3 programme de l'animation du robot 560 dans l'espace articulaire

```

l(1)=Link([0,      0,      0,      -pi/2,      0]);
l(2)=Link([0,      0.2435,      0.431,      0,      0]);
l(3)=Link([0,      -0.0934,      -0.020,      pi/2,      0]);
l(4)=Link([0,      0.431,      0,      -pi/2,      0]);
l(5)=Link([0,      0,      0,      +pi/2,      0]);
l(6)=Link([0,      0.056,      0,      0,      0]);

puma = SerialLink(l)

figure(1)
plot3(x,y,z)
puma.plot([Theta1,Theta2,Theta3,Theta4,Theta5,Theta6])

```

A.4 programme de l'animation du robot 560 dans l'espace opérationnel

```

l(1)=Link([0,      0,      0,      -pi/2,      0]);
l(2)=Link([0,      0.2435,      0.431,      0,      0]);
l(3)=Link([0,      -0.0934,      -0.020,      pi/2,      0]);
l(4)=Link([0,      0.431,      0,      -pi/2,      0]);
l(5)=Link([0,      0,      0,      +pi/2,      0]);
l(6)=Link([0,      0.056,      0,      0,      0]);

puma = SerialLink(l)

figure(1)

plot3(x,y,z, '.')
puma.teach()
hold on
plot3(-0.5,0.7,0.2,'go')
plot3(-0.1,0.7,0.2,'go')
plot3(0.2,0.6,0.2,'ro')
plot3(0.4,0.4,0.2,'ro')
plot3(0.6,-0.08,-0.4,'ro')
plot3(0.3,-0.6,-0.5,'go')

patch([-0.8,-0.2,-0.2,-0.8],[0.6,0.6,1,1],[-1.4,-1.4,-1.4,-1.4],'blue');
patch([-0.8,-0.2,-0.2,-0.8],[0.6,0.6,1,1],[0.1,0.1,0.1,0.1],'red');
patch([-0.8,-0.2,-0.2,-0.8],[0.6,0.6,0.6,0.6],[0.1,0.1,-1.4,-1.4],'green');
patch([-0.8,-0.2,-0.2,-0.8],[1,1,1,1],[0.1,0.1,-1.4,-1.4],'yellow');
patch([-0.8,-0.8,-0.8,-0.8],[1,0.6,0.6,1],[0.1,0.1,-1.4,-1.4],'cyan');
patch([-0.2,-0.2,-0.2,-0.2],[1,0.6,0.6,1],[0.1,0.1,-1.4,-1.4],'black');

patch([0,0.6,0.6,0],[-0.9,-0.9,-0.4,-0.4],[-1.4,-1.4,-1.4,-1.4],'blue');
patch([0,0.6,0.6,0],[-0.9,-0.9,-0.4,-0.4],[-0.6,-0.6,-0.6,-0.6],'red');
patch([0,0.6,0.6,0],[-0.9,-0.9,-0.9,-0.9],[-0.6,-0.6,-1.4,-1.4],'green');
patch([0,0.6,0.6,0],[-0.4,-0.4,-0.4,-0.4],[-0.6,-0.6,-1.4,-1.4],'yellow');
patch([0,0,0,0],[-0.4,-0.9,-0.9,-0.4],[-0.6,-0.6,-1.4,-1.4],'cyan');
patch([0.6,0.6,0.6,0.6],[-0.4,-0.9,-0.9,-0.4],[-0.6,-0.6,-1.4,1.4],'black');

puma.plot([Theta1,Theta2,Theta3,Theta4,Theta5,Theta6])

```

Annexe B

B.0 Commande PD en poursuite :

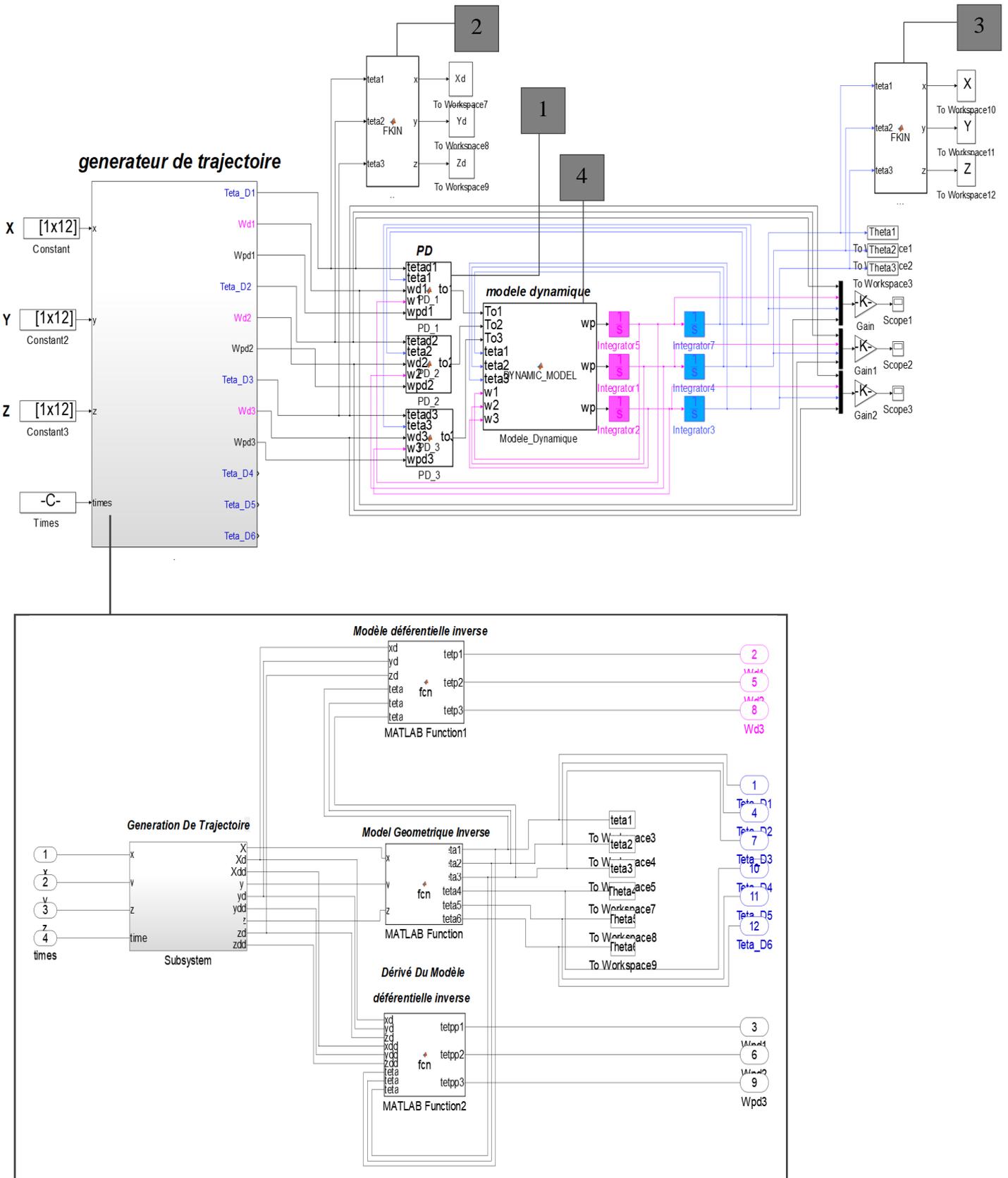


Figure B.0 : application de la commande PD en poursuite au Robot PUMA560.

1 Contrôleur PD

```
function to1 = PD_1(tetad1,teta1,wd1,w1,wpd1)
to1=(625*(tetad1-teta1)+50*(wd1-w1));
```

```
function to2 = PD_2(tetad2,teta2,wd2,w2,wpd2)
to2=(625*(tetad2-teta2)+50*(wd2-w2));
```

```
function to3 = PD_3(tetad3,teta3,wd3,w3,wpd3)
to3=(625*(tetad3-teta3)+50*(wd3-w3));
```

2 Transformation des trajectoires désirées vers l'espace opérationnel :

```
function [x,y,z] = FKIN(teta1,teta2,teta3)
teta4 = 0;
teta5 = 0;
teta6 = 0;
d1 = 0.6718 ; d2 = 0.2435 ; d3 = 0 ; d4 = 0.4331 ; d5 = 0 ; d6 = 0.0565 ;
a2 = 0.4318 ; a3 = -0.0203 ;

T01 = [cos(teta1) 0 -sin(teta1) 0 ; sin(teta1) 0 cos(teta1) 0 ; 0 -1 0
0 ; 0 0 0 1];
T12 = [cos(teta2) -sin(teta2) 0 a2*cos(teta2) ; sin(teta2) cos(teta2) 0
a2*sin(teta2) ; 0 0 1 d2 ; 0 0 0 1];
T23 = [cos(teta3) 0 sin(teta3) a3*cos(teta3) ;
sin(teta3) 0 -cos(teta3) a3*sin(teta3) ; 0 1 0 0 ; 0 0 0 1];
T34 = [cos(teta4) 0 -sin(teta4) 0 ; sin(teta4) 0 cos(teta4) 0 ; 0 -1 0
d4 ; 0 0 0 1];
T45 = [cos(teta5) 0 sin(teta5) 0 ; sin(teta5) 0 -cos(teta5) 0 ; 0 1 0
0 ; 0 0 0 1];
T56 = [cos(teta6) -sin(teta6) 0 0 ; sin(teta6) cos(teta6) 0 0 ; 0 0 1
d6 ; 0 0 0 1];

MGD = T01*T12*T23*T34*T45*T56;

x = MGD(1,4);
y = MGD(2,4);
z = MGD(3,4);
```

3 Transformation des trajectoires réelles vers l'espace opérationnel :

```
function [x,y,z] = FKIN(teta1,teta2,teta3)
teta4 = 0;
teta5 = 0;
teta6 = 0;
d1 = 0.6718 ; d2 = 0.2435 ; d3 = 0 ; d4 = 0.4331 ; d5 = 0 ; d6 = 0.0565 ;
a2 = 0.4318 ; a3 = -0.0203 ;
T01 = [cos(teta1) 0 -sin(teta1) 0 ; sin(teta1) 0 cos(teta1) 0 ; 0 -1 0 0 ; 0 0 0 1];
T12 = [cos(teta2) -sin(teta2) 0 a2*cos(teta2) ; sin(teta2) cos(teta2) 0 a2*sin(teta2) ; 0 0 1 d2 ; 0 0 0 1];
T23 = [cos(teta3) 0 sin(teta3) a3*cos(teta3) ; sin(teta3) 0 -cos(teta3) a3*sin(teta3) ;
0 1 0 0 ; 0 0 0 1];
T34 = [cos(teta4) 0 -sin(teta4) 0 ; sin(teta4) 0 cos(teta4) 0 ; 0 -1 0 d4 ; 0 0 0 1];
T45 = [cos(teta5) 0 sin(teta5) 0 ; sin(teta5) 0 -cos(teta5) 0 ; 0 1 0 0 ; 0 0 0 1];
T56 = [cos(teta6) -sin(teta6) 0 0 ; sin(teta6) cos(teta6) 0 0 ; 0 0 1 d6 ; 0 0 0 1];
MGD = T01*T12*T23*T34*T45*T56;
x = MGD(1,4);
y = MGD(2,4);
z = MGD(3,4);
```

4 Modèle dynamique du robot PUMA560

```
function [wp1,wp2,wp3]= DYNAMIC_MODEL(To1,To2,To3,teta1,teta2,teta3,w1,w2,w3)

to=[To1;To2;To3];

M=[2.57+1.38*cos(teta2)^2+0.30*sin(teta2)*sin(teta3)+7.44*10^-1*cos(teta2)*sin(teta2+teta3),6.90*10^-1*sin(teta2)-1.34*10^-1*cos(teta2+teta3)+2.38*10^-2*cos(teta2),-1.34*10^-1*cos(teta2+teta3)-3.97*10^-3*sin(teta2+teta3);
6.90*10^-1*sin(teta2)-1.34*10^-1*cos(teta2+teta3)+2.38*10^-2*cos(teta2),6.79+7.44*10^-1*sin(teta3),0.333+3.72*10^-1*sin(teta3)-1.10*10^-2*cos(teta3);
-1.34*10^-1*cos(teta2+teta3)-3.97*10^-3*sin(teta2+teta3),0.333+3.72*10^-1*sin(teta3)-1.10*10^-2*cos(teta3),1.16];

V=[((-2.76*sin(teta2)*cos(teta2)+7.44*10^-1*cos(teta2+teta2+teta3)+0.60*sin(teta2)*cos(teta3)-2.13*10^-2*(1-2*sin(teta2)*sin(teta3)))*w2,((6.90*10^-1*cos(teta2)+1.34*10^-1*sin(teta2+teta3)-2.38*10^-2*sin(teta2))*w2+(2.67*10^-1*sin(teta2+teta3)-7.58*10^-3*cos(teta2+teta3))*w3),(7.44*10^-1*cos(teta2)*cos(teta2+teta3)+0.60*sin(teta2)*cos(teta3)+2.20*10^-2*cos(teta2)*sin(teta2+teta3)-2.13*10^-2*(1-2*sin(teta2)*sin(teta3)))*w1+0.5*(2.67*10^-1*sin(teta2+teta3)-7.58*10^-3*cos(teta2+teta3))*w3;
(-0.5*(-2.76*sin(teta2)*cos(teta2)+7.44*10^-1*cos(teta2+teta2+teta3)+0.60*sin(teta2)*cos(teta3)-2.13*10^-2*(1-2*sin(teta2)*sin(teta3)))*w1,((2.20*10^-2*sin(teta3)+7.44*10^-1*cos(teta3))*w3),(0.5*(2.20*10^-2*sin(teta3)+7.44*10^-1*cos(teta3))*w3);
(-0.5*(7.44*10^-1*cos(teta2)*cos(teta2+teta3)+0.60*sin(teta2)*cos(teta3)+2.20*10^-2*cos(teta2)*sin(teta2+teta3)-2.13*10^-2*(1-2*sin(teta2)*sin(teta3)))*w1,(-0.5*(2.20*10^-2*sin(teta3)+7.44*10^-1*cos(teta3))*w2),0];

V=V*[w1; w2; w3];

G=[0;-37.2*cos(teta2)-8.4*sin(teta2+teta3)+1.02*sin(teta2);-8.4*sin(teta2+teta3)+0.25*cos(teta2+teta3)];

wp=inv(M)*(to-(V+G));
wp1=wp(1);wp2=wp(2);wp3=wp(3);
```

B.1 Commandes PD avec compensation de gravité :

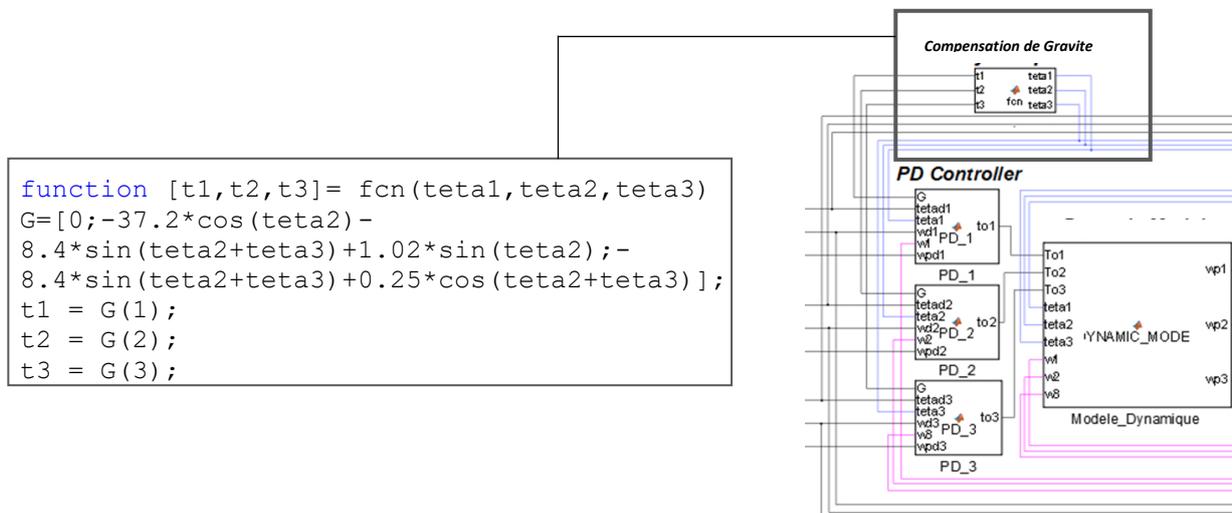


Figure B1 : application de la commande PD avec compensation de gravité en poursuite, au Robot PUMA560.

B.2 Commande PID en poursuite :

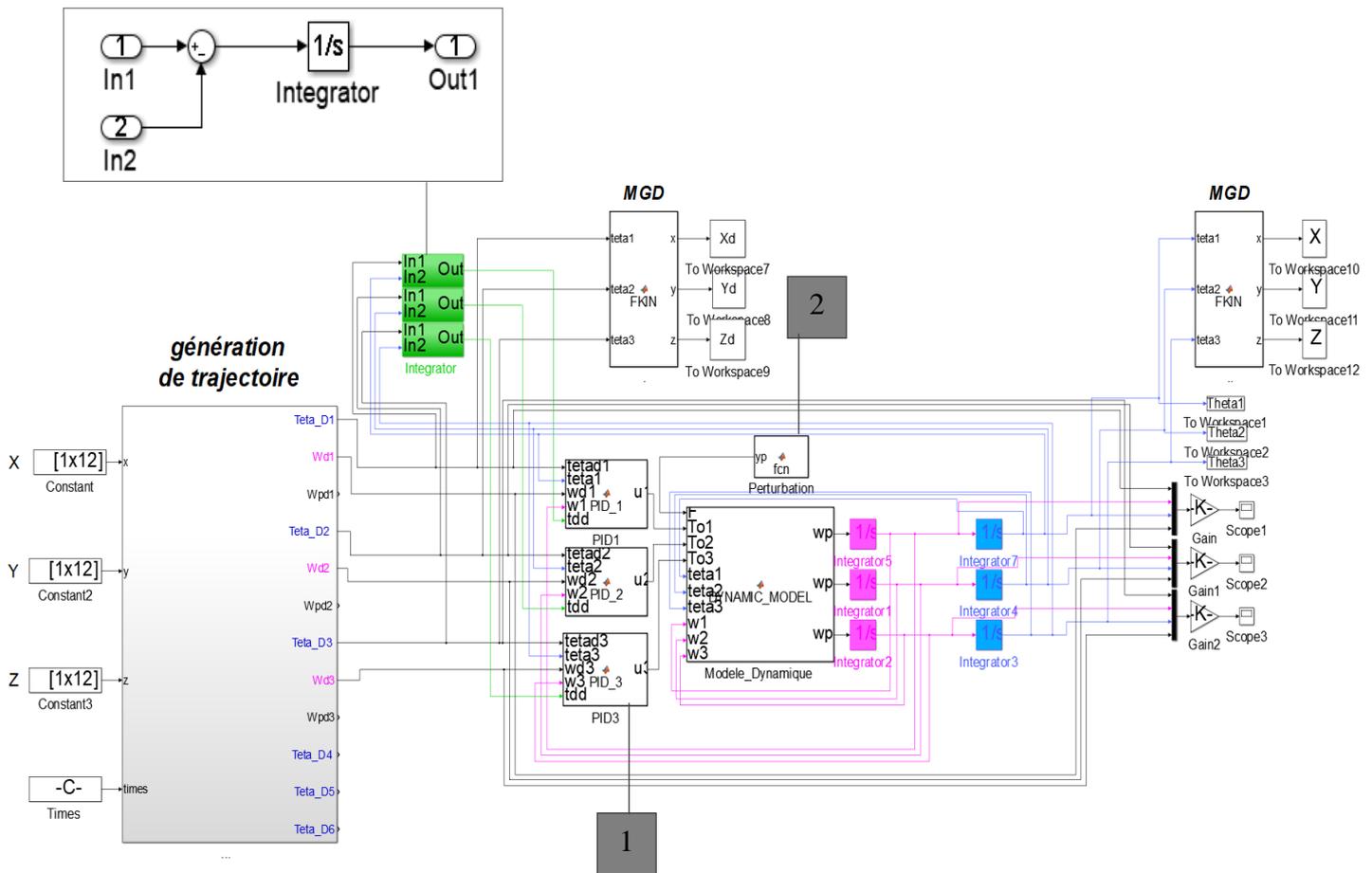


Figure C.2 : Application de la commande PID en poursuite au Robot PUMA560.

1 Contrôleur PID

```
function [u1,y]= PID_1(tetad1,teta1,wd1,w1,tdd)
u1=(300*(tetad1-teta1)+60*(wd1-w1)+5*tdd);
y = tetad1-teta1 ;
```

```
function [u2,y]= PID_2(tetad2,teta2,wd2,w2,tdd)
u2=(500 *(tetad2-teta2)+100*(wd2-w2)+625*tdd);
y = tetad2-teta2 ;
```

```
function [u3,y]= PID_3(tetad3,teta3,wd3,w3,tdd)
u3=(500*(tetad3-teta3)+100*(wd3-w3)+625*tdd);
y = tetad3-teta3;
```

2 Perturbation

```
function yp = fcn()
yp=[randn;randn;randn];
```

B.3 Commandes PD_CTC en poursuite :

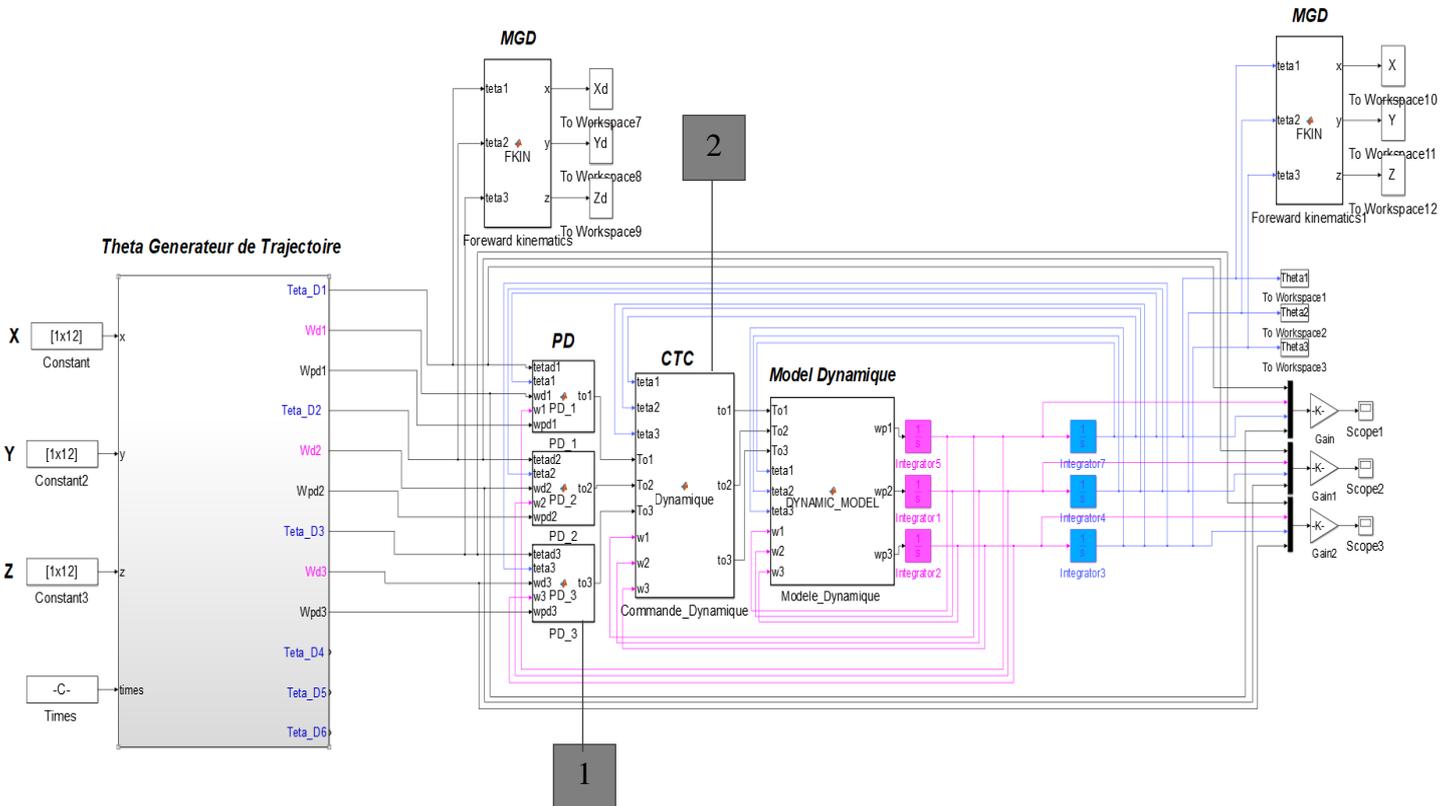


Figure B.3 : application de la commande PD_CTC en poursuite, au Robot PUMA560.

2 Découplage et linearisation

```
function [to1,to2,to3] = CTC (teta1,teta2,teta3,To1,To2,To3,w1,w2,w3)
```

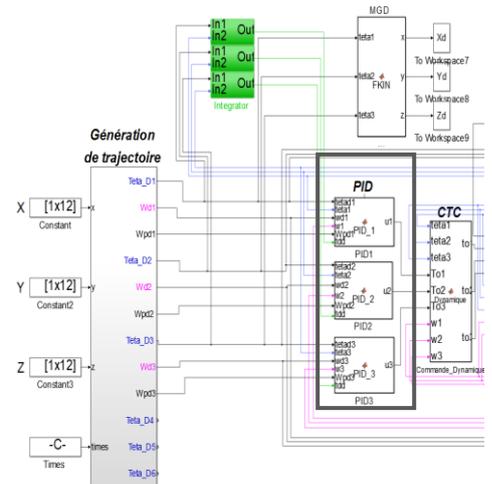
```
Top=[To1;To2;To3];
M = [ . . . ]; V = [ . . . ]; G = [ . . . ];
V=V*[w1;w2;w3];
to=M*Top+(V+G);
to1=to(1);to2=to(2);to3=to(3);
```

B.4 Commandes PID_CTC en poursuite :

```
function u1 = PID_1(tetad1,teta1,wd1,w1,Wpd1,tdd)
u1=(300*(tetad1-teta1)+60*(wd1-w1)+5*tdd+Wpd1);

function u2 = PID_2(tetad2,teta2,wd2,w2,Wpd2,tdd)
u2=(500*(tetad2-teta2)+100*(wd2-w2)+625*tdd+Wpd2);

function u3 = PID_3(tetad3,teta3,wd3,w3,Wpd3,tdd)
u3=(500*(tetad3-teta3)+100*(wd3-w3)+625*tdd+Wpd3);
```



B.5 Programme animation PUMA560:

```

l(1)=Link([0,      0,      0,      -pi/2,      0]);
l(2)=Link([0,      0.2435,      0.431,      0,      0]);
l(3)=Link([0,      -0.0934,      -0.020,      pi/2,      0]);
l(4)=Link([0,      0.431,      0,      -pi/2,      0]);
l(5)=Link([0,      0,      0,      +pi/2,      0]);
l(6)=Link([0,      0.056,      0,      0,      0]);

puma = SerialLink(l)

figure(1)

%Xd Yd Zd generation de la MGT //
plot3(X,Y,Z,'r.','linewidth',1.5)
%X Y Z resultats du model geometrique directe //
hold on
plot3(Xd,Yd,Zd,'b.','linewidth',1.5)

patch([-0.8,-0.2,-0.2,-0.8],[0.6,0.6,1,1],[-1.4,-1.4,-1.4,-1.4],'blue');
patch([-0.8,-0.2,-0.2,-0.8],[0.6,0.6,1,1],[0.1,0.1,0.1,0.1],'red');
patch([-0.8,-0.2,-0.2,-0.8],[0.6,0.6,0.6,0.6],[0.1,0.1,-1.4,-1.4],'green');
patch([-0.8,-0.2,-0.2,-0.8],[1,1,1,1],[0.1,0.1,-1.4,-1.4],'yellow');
patch([-0.8,-0.8,-0.8,-0.8],[1,0.6,0.6,1],[0.1,0.1,-1.4,-1.4],'cyan');
patch([-0.2,-0.2,-0.2,-0.2],[1,0.6,0.6,1],[0.1,0.1,-1.4,-1.4],'black');

patch([1.4,-1.4,-1.4,1.4],[-0.3,-0.3,-0.1,-0.1],[-0.35,-0.35,-0.35,-
0.35],'blue');
patch([1.4,-1.4,-1.4,1.4],[-0.3,-0.3,-0.1,-0.1],[1.4,1.4,1.4,1.4],'red');
patch([1.4,-1.4,-1.4,1.4],[-0.3,-0.3,-0.3,-0.3],[1.4,1.4,-0.35,-
0.35],'green');
patch([1.4,-1.4,-1.4,1.4],[-0.1,-0.1,-0.1,-0.1],[1.4,1.4,-0.35,-
0.35],'yellow');
patch([1.4,1.4,1.4,1.4],[-0.1,-0.3,-0.3,-0.1],[1.4,1.4,-0.35,-
0.35],'cyan');
patch([-1.4,-1.4,-1.4,-1.4],[-0.1,-0.3,-0.3,-0.1],[1.4,1.4,-0.35,-
0.35],'black');

patch([0,0.6,0.6,0],[-0.9,-0.9,-0.4,-0.4],[-1.4,-1.4,-1.4,-1.4],'blue');
patch([0,0.6,0.6,0],[-0.9,-0.9,-0.4,-0.4],[-0.6,-0.6,-0.6,-0.6],'red');
patch([0,0.6,0.6,0],[-0.9,-0.9,-0.9,-0.9],[-0.6,-0.6,-1.4,-1.4],'green');
patch([0,0.6,0.6,0],[-0.4,-0.4,-0.4,-0.4],[-0.6,-0.6,-1.4,-1.4],'yellow');
patch([0,0,0,0],[-0.4,-0.9,-0.9,-0.4],[-0.6,-0.6,-1.4,-1.4],'cyan');
patch([0.6,0.6,0.6,0.6],[-0.4,-0.9,-0.9,-0.4],[-0.6,-0.6,-1.4,-
1.4],'black')

%Theta1 .... Theta6 représente les tetas généré par le modèle dynamique //
puma.plot([Theta1,Theta2,Theta3,Theta4,Theta5,Theta6])

```

Résumé- Le travail réalisé concerne la planification du chemin et la commande des bras manipulateurs. En guise d'application, le choix s'est porté sur le robot PUMA560.

Après un aperçu sur les robots manipulateurs, on s'est intéressé à leur description mathématique et les différents modèles mis en jeu. La suite de travail concerne la planification de trajectoires notamment en présence d'obstacles et le cas de points via. Puis s'ensuivra par la problématique du suivie de trajectoires en utilisant deux types de commandes, linéaire (PD et PID) et non linéaire (PD avec compensation de gravite et la commande linéarisant découplante).

Mots clés : Bras de robots, planification de trajectoires, commande des bras de robots, commande en poursuite.

Abstract- The work carried out concerns path planning and robot manipulators' control. The PUMA560 robot is used as benchmark.

After robotics overview, different models involved in manipulators description are studied. Then, the trajectory planning problem is tackled, especially for the case of multiple via-points corresponding to the presence of obstacles. Finally, linear (PD and PID) and non-linear (PD with gravity compensation and computed torque) control laws are used to deal with the trajectory tracking problem.

Keywords : robot arms, trajectory planning, robot arms control, tracking control.

ملخص يتعلق العمل المقدم بالتخطيط للمسارات ومراقبة ذراع المتلاعبين. وكتطبيق، تم اختيار التطبيق على الروبوت بوما 560 بعد نظرة عامة على المتلاعبين الروبوتية، نظرنا إلى وصفهم الرياضي والنماذج المختلفة المعنية. يتعلق مجمل العمل على تخطيط مع تعويض الجاذبية (المسارات خاصة في وجود عقبات وحالة النقاط العبور. ثم يتبع ذلك بمشكلة تتبع المسار باستخدام نوعين من الضوابط، الخطية (PD و PID) وغير الخطية PD والتحكم الخطي الفاصل

كلمات مفتاحية ذراع الروبوت مخطط المسار التحكم في الروبوت التحكم المتتالي

