

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA
RECHERCHE SCIENTIFIQUE

جامعة سعد دحطب البليدة 1

Université SAAD DAHLEB de BLIDA 1



كلية التكنولوجيا

Faculté de Technologies

قسم الالكترونك

Département d'Électronique

Mémoire de projet de fin d'études

Filière Automatique

Spécialité Automatique et Systèmes

Conception, réalisation et commande optimisée d'un bras manipulateur SCARA

Présenté par :

OUNNAS Hakim

SEGHIR ABDELLAH Siham

proposé par :

Pr. KARA Kamel

Mr. BENRABAH Mohamed

Année Universitaire : 2018/2019

Remerciements

En tout premier lieu, nous tenons à remercier le bon Dieu, tout puissant, de nous avoir donnés la force pour survivre, ainsi que l'audace pour dépasser toutes les difficultés.

Ce travail a été effectué au laboratoire LABSET (Laboratoire des Systèmes Electrique et Télécommande) de l'Université Saad Dahleb de Blida, sous la direction du Professeur K. KARA.

Notre plus grande gratitude va à nos encadreur : le professeur K. Kara et Mr. M. Benrabah pour leurs efforts et leurs innombrables conseils riches en expérience, ce qui nous a aidé à apprendre la rigueur dans le travail.

Nous adressons particulièrement nos vifs remerciements à Monsieur Seghir Abdellah Mazaar, pour sa gentillesse, son écoute, ses conseils techniques et scientifiques et pour les nombreux conseils qu'il nous a prodigués tout au long de notre travail, il nous a aidés à franchir les difficultés que nous avons pu rencontrer.

Nos sincères remerciements s'adressent à tous nos camarades, aux professeurs ainsi qu'aux autorités académiques de l'Université Saad Dahleb de Blida. Le soutien indéfectible de nos amis et de nos familles nous ont permis de ne pas nous écarter dans les moments difficiles de l'objectif c'est pour cela que nous tenons à les remercier amplement et à leur exprimer notre gratitude.

Que tous ceux qui d'une manière ou d'une autre, ont rendu possible la réalisation de ce travail trouvent ici l'expression de notre profonde gratitude.



Dédicace

Je dédie ce mémoire de fin d'étude :

A mes très chers parents,

A mon père, qui m'a aidé à construire l'homme et la personne que je suis aujourd'hui, sans lui toute ces années n'auraient jamais pu aboutir, grâce à son soutien moral et à l'éducation qu'il m'a donnée. Tu n'es plus de ce monde mais les valeurs que tu m'as apprises et les mots que tu me disais m'accompagneront éternellement. Je prie Dieu de t'accorder sa sainte miséricorde et de t'accueillir en son vaste paradis.

A ma mère, l'incarnation de la douceur et de la bienveillance, celle qui m'a appris à aimer le travail, aimer les autres et aimer la vie, dans les moments difficiles tu as été la lumière qui m'a guidé et que j'ai suivie sans me poser de questions, je te dédis toute ces années de dur labeur et ce mémoire que j'ai façonné à ton image.

A mon frère HALIM et à mes sœurs WAFIA et YASMINA, SABAH et SAMIRA pour leurs encouragements permanents, et leur soutien moral.

A mes amis, qui m'ont accompagné à chaque pas que je marchais, leurs réconforts inconditionnels et leurs bonnes idées.

A tous mes camarades de l'université et du département d'électronique ainsi qu'aux professeurs qui m'ont enseigné tout au long de mon parcours.

Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infaillible,

Merci d'être toujours là pour moi.

OUNNAS HAKIM



Dédicace

Je dédie ce modeste travail

A mon source de bonheur mon chère papa et ma chère maman Pour son grand amour et son encouragement.

Qui n'ont jamais cessés de formuler des prières à mon égard , de me soutenir et de m'encourager pour que je puisse atteindre mes objectifs.

A mes chers frères , mes très chères sœurs (Hassiba et Fella) et mon beau frère, je leurs souhaite beaucoup de réussite dans leurs chemin.

*A tous la famille **seghir abdellah** et **said azza** .*

*Mon cher ami **Bellaiifa islam** .*

*Aux personnes qui m'ont toujours aidé et encouragé, qui étaient toujours à mes côtés, et qui m'ont accompagné durant mon chemin d'études supérieures, mes aimables amis, collègues d'étude, et sœurs de cœur, **Ichraf, Hadjer, Rima, Katia, Meriem, Samah, Linda, Maroua, Hamida et Naghem.***

*Je le dédie à mes enseignants (**K, KARA, M. BERABAH...**) qui nous formés dans nôtres vie professionnelle en particulier.*

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

Siham



ملخص

العمل المقدم في هذا المشروع يندمج ضمن إطار التصميم والتحكم في ذراع آلية باستخدام عنصر التحكم PID المحسن عن طريق الخوارزمية TLBO. في الجزء الأول من هذا العمل، وبعد القيام بنموذج للذراع، استخدمنا خوارزمية التحسين TLBO لاستخراج معاملات عنصر التحكم PID. بعد ذلك، أجرينا العديد من عمليات المحاكات لتسليط الضوء على أداء الذراع الآلية. استخدمنا برنامج Matlab للقيام بهذه المحاكات. في الجزء الثاني من هذا العمل، عملنا على تطوير الكود الذي سيتم تثبيته في DSP، وأيضاً قمنا بتحسين مختلف الدوائر الكهربائية، وأجرينا العديد من الاختبارات التجريبية وفي الأخير أخذنا النتائج التي تم الحصول عليها.

Résumé

Le travail présenté dans ce projet de fin d'étude consiste à la conception, réalisation et la commande d'un bras manipulateur en utilisant un régulateur PID optimisé par l'algorithme TLBO. En effet, dans la première partie de ce travail et après avoir procédé à la modélisation du bras, nous avons utilisé l'algorithme d'optimisation TLBO pour déterminer les paramètres du régulateur PID. Ensuite, nous avons effectué plusieurs simulations pour mettre en évidence les performances du bras. Nous avons utilisé le logiciel Matlab pour faire ces simulations. La deuxième partie, consiste à développer, le code qui va être implémenté dans le DSP, mettre au point les différents circuits et effectuer les différents essais expérimentaux et prend les résultats obtenus.

Abstract

The work presented in this project consists in the design, realization and the control of a manipulator arm by using a PID regulator optimized by TLBO algorithm. Indeed, in the first part of this work and after modeling the arm, we used the TLBO algorithm to determine the parameters of the PID regulator. Then, we performed several simulations to highlight the performance of the arm. We used the Matlab software to do these simulations. The second part of this work consists in developing the code that will be implemented in the DSP, developing the different circuits and performing the various experimental tests and taking the results obtained.



Table des matières

Remerciements

Dédicace

ملخص

Résumé

Abstract

Liste des acronymes et abréviations

Liste des figures

Liste des tableaux

Introduction générale.....1

Chapitre 1 : Modélisation des Robots

1.1 Introduction.....3

1.2 Modélisation géométrique.....4

1.2.1 Modèle géométrique direct.....7

1.2.2 Modèle géométrique inverse.....10

1.3 Modélisation cinématique.....13

1.3.1 Modèle cinématique direct.....13

1.3.2 Modèle cinématique inverse.....16

1.4 Modélisation dynamique.....17

1.4.1 Modèle dynamique inverse.....17

1.4.2 Modèle dynamique direct.....20

1.5 Conclusion.....21



Chapitre 2 :

Méthode d'optimisation: Teaching-Learning Based Optimization

2.1 Introduction.....	22
2.2 Problème d'optimisation.....	22
2.3 Optimisation métaheuristique.....	22
2.3.1 Métaheuristique.....	22
2.3.2 Domaine d'application.....	23
2.4 Teaching-Learning Based Optimization Algorithm.....	23
2.4.1 Principe de l'algorithme TLBO.....	24
2.5 Exemples d'optimisation en utilisant la méthode TLBO.....	27
2.5.1 Comparaison entre TLBO et PSO.....	28
2.6 Conclusion.....	29

Chapitre 3 :

Simulation de la commande d'un bras manipulateur à deux degrés de liberté en utilisant des régulateurs PID

3.1 Introduction.....	30
3.2 Correction des systèmes.....	30
3.2.1 Correction en boucle ouverte.....	30
3.2.2 Correction en boucle fermée.....	31
3.3 Correcteurs classiques.....	31
3.3.1 Correcteur proportionnel (P).....	31
3.3.2 Correcteur proportionnel intégrateur (PI).....	32
3.3.3 Correcteur proportionnel dérivateur (PD).....	32
3.3.4 Correcteur proportionnel intégral dérivé (PID)	32



3.4 Résultats de simulation.....	33
3.4.1 Modèle dynamique du bras.....	33
3.4.2 Paramètres des régulateurs PID utilisés.....	34
3.4.3 Résultats obtenus pour des références en échelon	35
3.4.4 Résultats obtenus pour des références sinusoïdales.....	38
3.4.5 Résultats obtenus pour des références multi échelon.....	39
3.4.6 Poursuite des trajectoires dans l'espace opérationnel.....	41
3.5 Conclusion.....	42

Chapitre 4 :

Conception, réalisation et commande pratique d'un bras manipulateur

4.1 Introduction.....	43
4.2 Conception du bras manipulateur.....	43
4.3 Constituant du banc d'expérimentation.....	48
4.4 Description des éléments du banc d'expérimentation.....	48
4.4.1 Le kit de développement <i>eZdsp F28335</i>	48
4.4.2 Circuit de commande (carte mère).....	52
4.4.3 Moteur à courant continu (MCC).....	56
4.5 Identification par les réseaux de neurones.....	58
4.6 Commande pratique du bras en utilisant deux régulateurs PID optimisés à l'aide de l'algorithme TLBO.....	62
4.6.1 Structure et mise en œuvre de la commande.....	62
4.6.2 Résultats pratiques de commande.....	64
4.7 Conclusion.....	69
Conclusion générale.....	70
Références	
Annexe	



Liste des acronymes et abréviations

SCARA : Selective Compliance Articulated Robot Arm.

MGD : Modèle géométrique direct.

MGI : Modèle géométrique inverse.

2ddl : Deux degré de liberté.

DH : Denavit-Hartenberg.

Rot : Rotation.

Trans : Translation.

θ : Vecteur des positions articulaires.

$\dot{\theta}$: Vecteur des vitesses articulaires.

$\ddot{\theta}$: Vecteur des accélérations articulaires.

F : Vecteur représentant l'effort extérieur.

Γ : Vecteur des couples/forces.

$J(q)$: Matrice jacobienne.

L : Lagrangienne.

E_c : Energie cinétique.

E_p : Energie potentielle.

$M(\theta)$: matrice d'inertie.

$C(\theta, \dot{\theta})$: vecteur de Coriolis/centrifuges.

$f(\dot{\theta})$: vecteur de frottement.

$G(\theta)$: vecteur de gravité.

TLBO : Teaching and Learning Based Optimization.

PSO : Particle Swarm optimization.



P : Correcteur proportionnel.

PI : Correcteur proportionnel intégrateur.

PD : Correcteur proportionnel dérivateur.

PID : Correcteur proportionnel intégral et dérivé.

K_P : Gain proportionnel.

T_I : Constante d'intégration.

T_D : Constante de dérivation.

$u(t)$: Signal de commande.

$y(t)$: Sortie du processus.

$r(t)$: La référence.

$e(t)$: L'erreur.

PWM : Pulse Width Modulation.

DSP : Digital Signal Processor.

GPIO : General purpose input output.

ALU : Unité arithmétique et logique.

ePWM : Enhanced Pulse Width Modulator.

SCI : Serial Communications Interface.

UART : Universal Asynchronous Receiver-Transmitter.



Liste des figures

Figures	Chapitre 1	Page
Figure.1.1	Robot à structure ouverte simple.	4
Figure.1.2	Paramètres de DH dans le cas d'une structure ouverte simple.	5
Figure.1.3	Modèle géométrique direct et inverse.	6
Figure.1.4	Structure du Robot SCARA à 3ddl.	7
Figure.1.5	Description géométrique d'un robot du type RRP.	8
Figure.1.6	Position du point P dans les deux cas.	12
Figure.1.7	Les deux postures du bras.	13
Figure.1.8	Vue de dessus du bras manipulateur à 3 ddl.	14
Figure.1.9	Détermination des aspects dans l'espace articulaire.	15
Figure.1.10	Image des aspects dans l'espace opérationnel.	16
Figure.1.11	Paramètres dynamiques d'un bras manipulateur.	19
Chapitre 2		
Figure.2.1	Organigramme d'algorithme TLBO.	26
Figure.2.2	Le minimum de la fonction f .	27
Figure.2.3	Le minimum de la fonction g .	27
Chapitre 3		
Figure.3.1	Correction en boucle ouverte.	30
Figure.3.2	Correction en boucle fermée.	31
Figure.3.3	Performances de commande pour la première articulation dans le cas d'une trajectoire en échelon.	36
Figure.3.4	Performances de commande pour la deuxième articulation dans le cas d'une trajectoire en échelon.	36



Figure.3.5	Erreur de poursuite pour la première articulation dans le cas d'une trajectoire en échelon.	37
Figure.3.6	Erreur de poursuite pour la première articulation dans le cas d'une trajectoire en échelon.	37
Figure.3.7	Performances de commande pour la première articulation dans le cas d'une trajectoire sinusoïdale.	38
Figure.3.8	Performances de commande pour la deuxième articulation dans le cas d'une trajectoire sinusoïdale.	39
Figure.3.9	Performances de commande pour la première articulation dans le cas d'une trajectoire en échelon multiple.	40
Figure.3.10	Performances de commande pour la deuxième articulation dans le cas d'une trajectoire en échelon multiple.	40
Figure.3.11	Poursuite d'un cercle dans l'espace opérationnel.	41
Figure.3.12	Poursuite d'une ligne droite dans l'espace opérationnel.	42
Chapitre 4		
Figure.4.1	Structure du bras manipulateur à 2ddl.	44
Figure.4.2	Eléments du banc d'expérimentation.	48
Figure.4.3	La carte eZdsp F28335.	49
Figure.4.4	Architecture du <i>TMS320F28335</i> .	50
Figure.4.5	Circuit de commande (carte mère).	52
Figure.4.6	Schéma électrique du circuit de commande.	53
Figure.4.7	Schéma électrique du circuit d'isolation avec un optocoupleur (6N137).	53
Figure.4.8	Schéma électrique du circuit de régulation de tension.	54
Figure.4.9	Schéma électrique du circuit de puissance.	54
Figure.4.10	Principe de fonctionnement d'un hacheur à 4 quadrants.	55
Figure.4.11	Commande en tension de l'hacheur.	55



Figure.4.12	Moteur à courant continu.	56
Figure.4.13	Schéma bloc de l'identification.	58
Figure.4.14	Architecture du réseau de neurone pour chaque modèle ($\theta(k) = \theta_1(k)$ ou $\theta_2(k)$).	59
Figure.4.15	Données d'apprentissage et de test pour la première articulation.	60
Figure.4.16	Données d'apprentissage et de test pour la deuxième articulation.	60
Figure.4.17	Résultat de test pour le modèle de la première articulation.	61
Figure.4.18	Résultat de test pour le modèle de la deuxième articulation.	61
Figure.4.19	Adaptateur USB vers TTL UART.	62
Figure.4.20	Structure de commande du bras manipulateur.	63
Figure.4.21	Résultat de commande du bras pour le dessin de deux demi-cercles de rayon différents.	64
Figure.4.22	Poursuite de la trajectoire de référence pour l'articulation θ_1 dans le cas du demi-cercle $R=0.28m$.	65
Figure.4.23	Erreur entre la référence et la sortie θ_1 dans le cas du demi-cercle $R=0.28m$.	65
Figure.4.24	Poursuite de la trajectoire de référence pour l'articulation θ_2 dans le cas du demi-cercle $R=0.28m$.	66
Figure.4.25	Erreur entre la référence et la sortie θ_2 dans le cas du demi-cercle $R=0.28 m$.	66
Figure.4.26	Poursuite de la trajectoire de référence pour l'articulation θ_1 dans le cas du demi-cercle $R=0.33 m$.	67
Figure.4.27	Erreur entre la référence et la sortie θ_1 dans le cas du demi-cercle $R=0.33 m$.	67
Figure.4.28	Poursuite de la trajectoire de référence pour l'articulation θ_2 dans le cas du demi-cercle $R=0.33 m$.	68
Figure.4.29	Erreur entre la référence et la sortie θ_2 dans le cas du demi-cercle $R=0.33 m$.	68



Liste des tableaux

Figures	Chapitre 1	Page
Tableau.1.1	Paramètres de DH du robot SCARA à 2ddl.	8
	Chapitre 2	
Tableau.2.1	Le minimum global avec TLBO pour plusieurs valeurs de population.	28
Tableau.2.2	Le minimum global avec TLBO pour plusieurs nombres d'itérations.	28
Tableau.2.3	Résultats comparatifs entre l'algorithme TLBO et l'algorithme PSO.	29
	Chapitre 3	
Tableau.3.1	Paramètres du système SCARA.	34
Tableau.3.2	Valeurs des paramètres des régulateurs PID obtenues en utilisant l'algorithme TLBO.	35
	Chapitre 4	
Tableau.4.1	Pièces du bras manipulateur avec leurs dimensions.	47
Tableau.4.2	Valeurs des paramètres des deux régulateurs PID.	63



Introduction générale

Introduction générale

La robotique est considérée comme l'un des plus grands axes de progrès scientifiques et techniques, elle consiste en l'automatisation de nombreux secteurs de l'activité humaine, l'augmentation de la productivité dans les entreprises, la stimulation, la compétitivité entre eux et surtout l'aptitude de travailler dans des zones à haut risque (espace, démantèlement nucléaire, déminage ...etc.).

Le secteur d'activité des systèmes robotiques s'est élargi de façon importante en particulier dans le domaine médical, spatial, industriel, militaire et dans le domaine de l'agriculture.

La robotique est un ensemble de disciplines (mécanique, électronique, automatique, informatique), elle se subdivise en deux types : robots mobiles et robots manipulateurs.

Un robot mobile est un système mécanique, électronique et informatique agissant physiquement sur son environnement en vue d'atteindre un objectif qui lui a été assigné, Cette machine est polyvalente et capable de s'adapter à certaines variations de ses conditions de fonctionnement. Robots à poste mobiles capables de se déplacer dans un environnement et peuvent être équipés de manipulateur suivant leurs utilisations.

Un robot manipulateur est en forme d'un bras et se compose d'un certain nombre de segments qui est conçu pour manipuler ou déplacer des matériaux, outils et pièces sans contact humain direct. Ils sont des dispositifs qui permettent aux humains d'interagir avec des objets dans un environnement en toute sécurité. Les robots manipulateurs sont utilisés dans des applications industrielles, et ils sont des robots installés sur le lieu de travail afin d'exécuter des travaux programmés [8].

Dans ce travail de mémoire, nous nous sommes intéressés aux robots manipulateurs pour des applications en robotique industrielle. L'objectif de notre travail est de réaliser et de commander un bras manipulateur à 2 degrés de liberté en développant une carte de commande DSP (*TMS320F335*), une commande PID en utilisant l'algorithme de Teaching and Learning Based Optimization (TLBO) et identification par les réseaux de neurones.

Ce mémoire est structuré en quatre chapitres :

Le premier chapitre est consacré à la modélisation des robots à savoir la modélisation géométrique, la modélisation cinématique et la modélisation dynamique.

Dans le deuxième chapitre nous représentons l'algorithme d'optimisation Teaching and Learning Based Optimization ainsi que les résultats de simulation de l'optimisation de quelques fonctions mathématiques du test.

Le troisième chapitre est consacré à la mise en œuvre de l'algorithme de commande et la présentation des résultats de simulation.

Nous présentons dans le dernier chapitre, les différents circuits réalisés et les différents résultats obtenus expérimentalement.

Enfin, nous terminons par une conclusion générale sur le travail réalisé.

Chapitre 1
Modélisation des Robots

1.1 Introduction

Pour commander un robot ou pour simuler son comportement on doit disposer de modèles. Plusieurs niveaux de modélisation sont possibles selon les objectifs, les contraintes de la tâche et les performances recherchées ; on distingue :

- Les modèles géométriques direct et inverse qui expriment la situation de l'organe terminal en fonction de la configuration du mécanisme et inversement,
- Les modèles cinématiques direct et inverse qui expriment la vitesse de l'organe terminal en fonction des vitesses articulaires et inversement,
- Les modèles dynamiques définissant les équations du mouvement du robot, qui permettent d'établir les relations entre les couples ou forces exercés par les actionneurs et les positions, vitesses et accélérations des articulations [1].

Dans ce chapitre, nous présentons dans un premier temps ces modèles. On se limitera au cas des robots à structure ouverte simple de type SCARA.

➤ Matrice de transformation homogène

Le formalisme mathématique fait appel aux matrices de transformation homogènes de dimension (4x4). La matrice de transformation homogène ${}^i\mathbf{T}_j$ représente la transformation permettant de passer du repère \mathbf{R}_i au repère \mathbf{R}_j :

$${}^i\mathbf{T}_j = \begin{bmatrix} {}^i\mathbf{A}_j & {}^i\mathbf{P}_j \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^i\mathbf{s}_j & {}^i\mathbf{n}_j & {}^i\mathbf{a}_j & {}^i\mathbf{p}_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.1)$$

Où ${}^i\mathbf{s}_j$, ${}^i\mathbf{n}_j$ et ${}^i\mathbf{a}_j$ désignent respectivement les vecteurs unitaires suivant les axes \mathbf{x}_j , \mathbf{y}_j et \mathbf{z}_j du repère \mathbf{R}_j exprimés dans le repère \mathbf{R}_i , et où ${}^i\mathbf{p}_j$ est le vecteur exprimant l'origine du repère \mathbf{R}_j dans le repère \mathbf{R}_i .

1.2 Modélisation géométrique

La modélisation géométrique permet de déterminer la configuration (position, orientation) de l'effecteur d'un robot en fonction de la configuration de ses liaisons. La modélisation des robots de façon systématique et automatique exige une méthode adéquate pour la description de leur morphologie. Plusieurs méthodes et notions ont été proposées. La plus répandue est celle de Denavit-Hartenberg [2]. Cette méthode développée pour des structures ouvertes simples. Une structure ouverte simple est composée de $n+1$ corps noté C_0, \dots, C_n , et de n articulation ($n \geq 1$).

Le corps C_0 désigne la base du robot et le corps C_n est le corps qui porte l'organe terminal. L'articulation i connecte le corps C_i au corps C_{i-1} (figure 1.1).

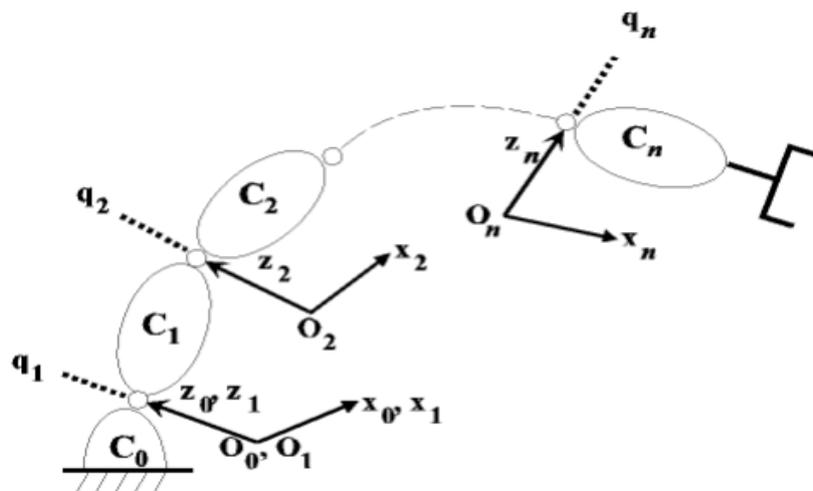


Figure 1.1. Robot à structure ouverte simple.

La méthode de description est fondée sur les règles et conventions suivantes :

- Les corps sont supposés parfaitement rigides. Ils sont connectés par des articulations considérées comme idéal (pas de jeu mécanique, pas d'élasticité), soit rotoïdes, soit prismatique,
- Le repère \mathbf{R}_i est lié au corps C_i ,
- L'axe z_i est porté par l'axe de l'articulation i ,

- L'axe \mathbf{x}_i est porté par la perpendiculaire commune aux axes \mathbf{z}_i et \mathbf{z}_{i-1} . Si les axes \mathbf{z}_i et \mathbf{z}_{i-1} sont parallèles ou colinéaires, le choix de \mathbf{x}_i n'est pas unique : des considérations de symétrie ou de simplicité permettent alors un choix rationnel [3].

Le passage du repère \mathbf{R}_{i-1} au repère \mathbf{R}_i s'exprime en fonction des quatre paramètres de DH Denavit-Hartenberg [2] suivants (figure 1.2) :

- α_i : angle entre les deux axes \mathbf{z}_{i-1} et \mathbf{z}_i correspondant à une rotation autour de l'axe \mathbf{x}_i .
- d_i : distance entre les deux origines \mathbf{O}_{i-1} et \mathbf{O}_i le long de l'axe \mathbf{z}_{i-1} .
- θ_i : angle entre les deux axes \mathbf{x}_{i-1} et \mathbf{x}_i correspondant à une rotation autour de l'axe \mathbf{z}_{i-1} .
- a_i : distance entre les deux origines \mathbf{O}_{i-1} et \mathbf{O}_i le long de l'axe \mathbf{x}_i .

Trois paramètres limitent les degrés de liberté du repère \mathbf{R}_i par rapport au repère \mathbf{R}_{i-1} :

- l'axe \mathbf{x}_i de \mathbf{R}_i est perpendiculaire à l'axe \mathbf{z}_{i-1} de \mathbf{R}_{i-1} ,
- l'axe \mathbf{x}_i coupe l'axe \mathbf{z}_{i-1} ,
- Les mouvements de l'articulation (rotoïde ou prismatique) sont toujours sur l'axe \mathbf{z} .

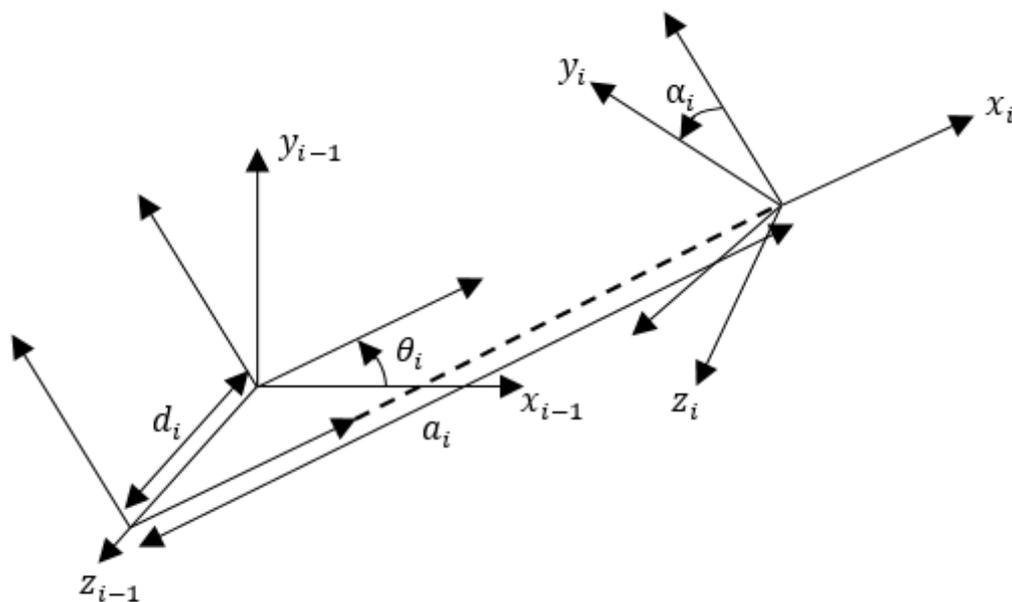


Figure 1.2. Paramètres de DH dans le cas d'une structure ouverte simple.

Le passage du repère \mathbf{R}_{i-1} au repère \mathbf{R}_i se fait à travers 4 étapes :

- une rotation d'angle θ_i autour de l'axe \mathbf{z}_{i-1} ,
- une translation de distance \mathbf{d}_i mesurée le long de l'axe \mathbf{z}_{i-1} ,
- une translation de distance \mathbf{a}_i mesurée le long de l'axe \mathbf{x}_i ,
- une rotation d'angle α_i autour de l'axe \mathbf{x}_i .

La matrice de transformation homogène définissant le repère \mathbf{R}_i dans le repère \mathbf{R}_{i-1} est donnée par (figure 1.2) :

$${}^{i-1}\mathbf{T}_i = \mathbf{Rot}(\mathbf{z}_{i-1}, \theta_i) \mathbf{Trans}(\mathbf{z}_{i-1}, \mathbf{d}_i) \mathbf{Trans}(\mathbf{x}_i, \mathbf{a}_i) \mathbf{Rot}(\mathbf{x}_i, \alpha_i)$$

$$= \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \mathbf{d}_i \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & \mathbf{a}_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & \mathbf{a}_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & \mathbf{a}_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & \mathbf{d}_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.2}$$

La modélisation géométrique est composée de deux parties (figure 1.3) :

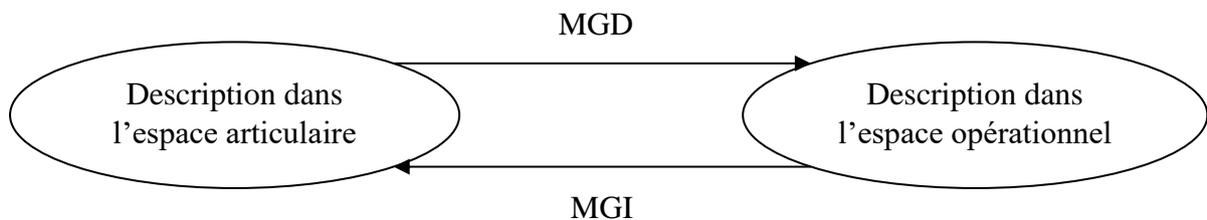


Figure 1.3. Modèle géométrique direct et inverse.

1.2.1 Modèle géométrique direct (MGD)

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

$${}^0T_n = {}^0T_1(q_1) {}^1T_2(q_2) \dots {}^{n-1}T_n(q_n) \quad (1.3)$$

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

$$\mathbf{X} = \mathbf{f}(\mathbf{q}) \quad (1.4)$$

Tel que :

$\mathbf{Q} = [q_1 \ q_2 \ \dots \ q_n]^T \in \mathbb{R}^n$: Vecteur des variables articulaires.

$\mathbf{X} = [x_1 \ x_2 \ \dots \ x_n]^T \in \mathbb{R}^n$: Vecteur des coordonnées opérationnelles.

Dans le cas d'une structure ouverte simple ou arborescente, le calcul de \mathbf{X} implique donc de calculer les matrices de transformations de l'organe terminal [3].

On se propose d'établir le modèle géométrique direct du robot SCARA à 2 degrés de libertés représenté dans sa configuration initiale dans la figure suivante (figure 1.4) :



Figure 1.4. Structure du Robot SCARA à 2ddl.

Afin de se faciliter la tâche, les différents repères permettant d'établir les paramètres de Denavit-Hartenberg sont représentés (figure 1.5).

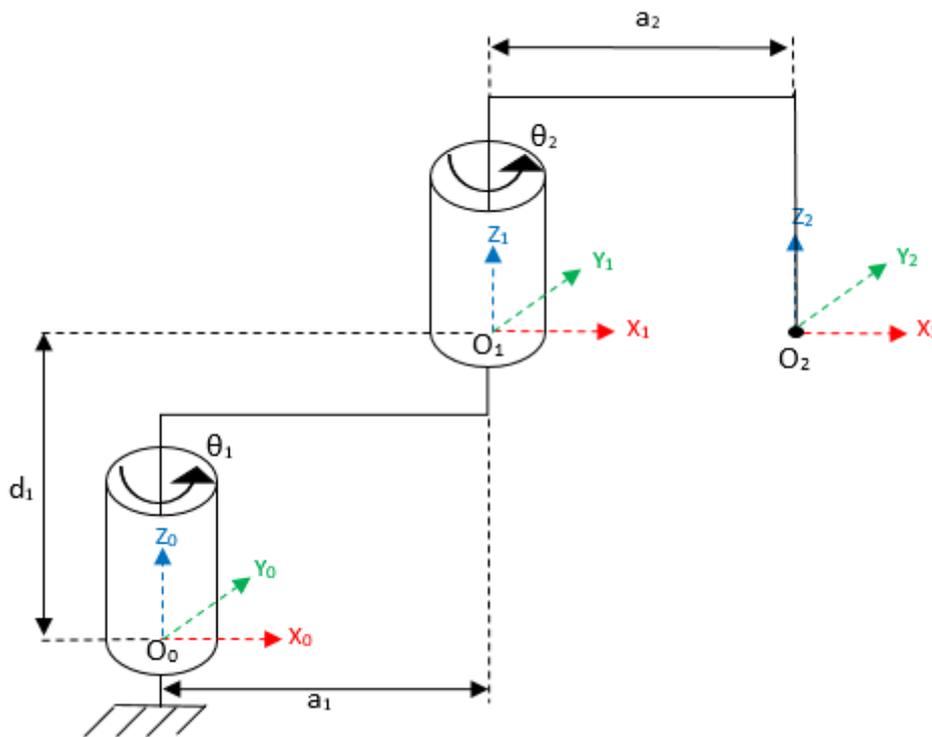


Figure 1.5. Description géométrique d'un robot du type RR.

a_1 : distance entre les deux origines O_0 et O_1 le long de l'axe x_1 .

a_2 : distance entre les deux origines O_1 et O_2 le long de l'axe x_2 .

d_1 : distance entre les deux origines O_0 et O_1 le long de l'axe z_0 .

A partir de la convention de Denavit-Hartenberg on obtient le tableau suivant :

Articulations	Paramètres d'articulation			
	a_i	θ_i	α_i	d_i
1	a_1	θ_1	0	d_1
2	a_2	θ_2	0	0

Tableau 1.1. Paramètres de DH du robot SCARA à 2ddl.

Les matrices de transformations homogènes des trois segments sont données par :

$${}^0\mathbf{T}_1 = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & a_1 \cos \theta_1 \\ \sin \theta_1 & \cos \theta_1 & 0 & a_1 \sin \theta_1 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.5)$$

$${}^1\mathbf{T}_2 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & a_2 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & a_2 \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.6)$$

Après la multiplication des matrices de transformations élémentaires, nous obtenons la matrice finale ${}^0\mathbf{T}_2$ qui sert à déterminer le modèle géométrique direct :

$${}^0\mathbf{T}_2 = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & 0 & a_1 \cos \theta_1 + a_2 \cos(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & a_1 \sin \theta_1 + a_2 \sin(\theta_1 + \theta_2) \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.7)$$

La matrice de transformation homogène peut se mettre sous sa forme générale :

$${}^0\mathbf{T}_2 = \begin{bmatrix} s_x & n_x & a_x & P_x \\ s_y & n_y & a_y & P_y \\ s_z & n_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.8)$$

D'après les deux expressions (1.7) et (1.8), nous obtenons les équations suivantes :

$$\begin{cases} s_x = \cos(\theta_1 + \theta_2) \\ s_y = \sin(\theta_1 + \theta_2) \\ s_z = 0 \end{cases} \quad (1.9)$$

$$\begin{cases} n_x = -\sin(\theta_1 + \theta_2) \\ n_y = \cos(\theta_1 + \theta_2) \\ n_z = 0 \end{cases} \quad (1.10)$$

$$\begin{cases} a_x = 0 \\ a_y = 0 \\ a_z = 1 \end{cases} \quad (1.11)$$

La situation de l'organe terminal est définie par les coordonnées de position :

$$\begin{cases} P_x = a_1 \cos \theta_1 + a_2 \cos(\theta_1 + \theta_2) \\ P_y = a_1 \sin \theta_1 + a_2 \sin(\theta_1 + \theta_2) \\ P_z = d_1 \end{cases} \quad (1.12)$$

1.2.2 Modèle géométrique inverse (MGI)

Le problème inverse consiste à calculer les coordonnées articulaires qui amènent l'organe terminal dans une situation désirée, spécifiée par ses coordonnées opérationnelles. Cette opération est très souvent appelée transformation de coordonnées (ou changement de coordonnées). Lorsqu'elle existe, la forme explicite qui donne toutes les solutions possibles au problème inverse constitue ce que l'on appelle le modèle géométrique inverse (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 articulaires. Des méthodes algébriques sont fournies dans cette section pour résoudre le MGI du bras SCARA. Ces méthodes sont utilisées pour effectuer une inversion du géométrique direct [4].

D'après les équations (1.12), nous pouvons calculer les deux coordonnées articulaires θ_1 et θ_2 :

➤ **Calcul de θ_1 :**

En reprenant les équations (1.12) du modèle géométrique directe et en utilisant les relations trigonométriques :

$$\cos(\alpha + \beta) = \cos\alpha\cos\beta - \sin\alpha\sin\beta \quad \text{Et} \quad \sin(\alpha + \beta) = \sin\alpha\cos\beta + \cos\alpha\sin\beta$$

Nous pouvons encore écrire en développant C_{12} , S_{12} :

$$\begin{cases} P_x = a_1 C_1 + a_2 (C_1 C_2 - S_1 S_2) \\ P_y = a_1 S_1 + a_2 (S_1 C_2 + S_2 C_1) \end{cases} \quad (1.13)$$

Avec : $\cos \theta_i = C_i$, $\sin \theta_i = S_i$

$$\cos(\theta_i + \theta_j) = C_{ij} , \sin(\theta_i + \theta_j) = S_{ij}$$

En regroupant suivant les inconnues C_1 et S_1 nous trouvons :

$$\begin{cases} P_x = (a_1 + a_2 C_2)C_1 - a_2 S_2 S_1 \\ P_y = a_2 S_2 C_1 + (a_1 + a_2 C_2)S_1 \end{cases} \quad (1.14)$$

On résout le système (1.14) par rapport aux deux inconnues C_1 et S_1 , on obtient :

$$\begin{cases} C_1 = \frac{P_x(a_1+a_2 C_2)+P_y a_2 S_2}{P_x^2+P_y^2} \\ S_1 = \frac{P_y(a_1+a_2 C_2)-P_x a_2 S_2}{P_x^2+P_y^2} \end{cases} \quad (1.15)$$

Avec :

$$P_x^2 + P_y^2 = a_1^2 + a_2^2 + 2a_1 a_2 C_2$$

D'où :

$$\theta_1 = \arctg\left(\frac{P_y(a_1+a_2 C_2)-P_x a_2 S_2}{P_x(a_1+a_2 C_2)+P_y a_2 S_2}\right) \quad (1.16)$$

➤ **Calcul de θ_2 :**

On a :

$$P_x^2 + P_y^2 = a_1^2 C_1^2 + a_2^2 C_{12}^2 + 2a_1 a_2 C_1 C_{12} + a_1^2 S_1^2 + a_2^2 S_{12}^2 + 2a_1 a_2 S_1 S_{12} \quad (1.17)$$

On obtient :

$$P_x^2 + P_y^2 = a_1^2 + a_2^2 + 2a_1 a_2 (C_1 C_{12} + S_1 S_{12}) \quad (1.18)$$

En utilisant la relation trigonométrique suivante :

$$\cos(\alpha - \beta) = \cos\alpha\cos\beta + \sin\alpha\sin\beta$$

On trouve :

$$C_2 = \frac{P_x^2 + P_y^2 - a_1^2 - a_2^2}{2a_1 a_2} \quad (1.19)$$

Ce qui implique :

$$\theta_2 = \pm \arccos\left(\frac{P_x^2 + P_y^2 - a_1^2 - a_2^2}{2a_1 a_2}\right) \quad (1.20)$$

Lorsque θ_2 est positive, le robot a une posture coude bas, si θ_2 est négatif le robot a une posture coude haut, voir (figure 1.6).

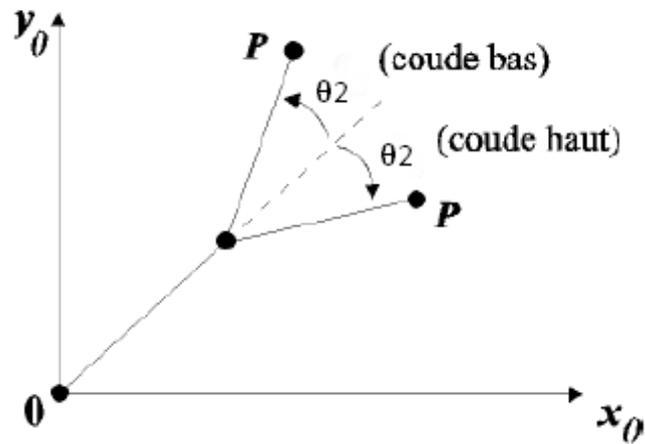


Figure 1.6. Position du point P dans les deux cas.

Ainsi, on aboutit au modèle géométrique inverse suivant :

$$\begin{cases} \theta_1 = \arctg\left(\frac{P_y(a_1+a_2 C_2)-P_x a_2 S_2}{P_x(a_1+a_2 C_2)+P_y a_2 S_2}\right) \\ \theta_2 = \pm \arccos\left(\frac{P_x^2 + P_y^2 - a_1^2 - a_2^2}{2a_1 a_2}\right) \end{cases} \quad (1.21)$$

On remarque qu'il y a 2 solutions, correspondant à 2 postures différentes du bras (En choisissant le signe, on détermine la configuration du coude) : L'une est dite « coude haut », l'autre « coude bas » (figure 1.7).

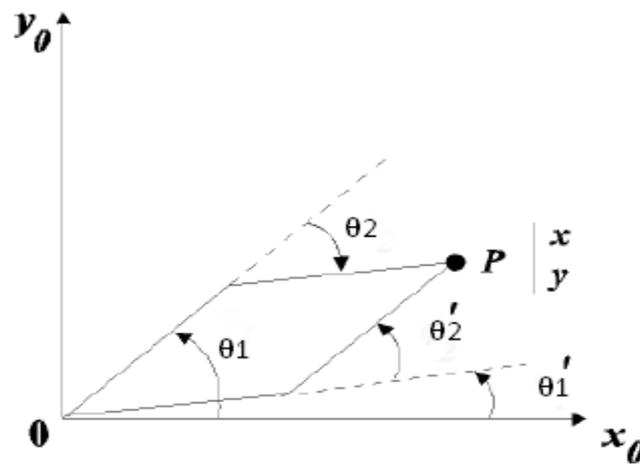


Figure 1.7. Les deux postures du bras.

Le modèle géométrique inverse est obtenu à partir des paramètres géométriques du tableau 1.1 qui représente un robot à deux degrés de liberté, en utilisant les résultats intermédiaires de la matrice de transformation 0T_2 .

1.3 Modélisation cinématique

La modélisation cinématique permet de trouver une solution au problème géométrique inverse quelle que soit la structure du robot. Elle permet aussi d'introduire la notion de vitesse de déplacement du mécanisme, pour calculer les vitesses de translation et de rotation du corps terminal en fonction des vitesses articulaires. Elle est donc à la base de leur commande en vitesse. La propriété intéressante de la modélisation cinématique est sa linéarité par rapport aux vitesses. Ainsi elle est plus simple à manipuler que la modélisation géométrique et permet d'utiliser des propriétés qui découlent de la résolution des systèmes linéaires.

Dans ce modèle on utilise la matrice jacobienne de base. Cette matrice peut être décomposée en deux ou trois matrices contenant des termes plus simples.

1.3.1 Modèle cinématique direct (MCD)

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

$$\dot{\mathbf{X}} = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}} \quad (1.22)$$

Où $\mathbf{J}(\mathbf{q})$ désigne la matrice jacobienne de dimension $(m \times n)$ du mécanisme, égale à $\frac{\partial \mathbf{X}}{\partial \mathbf{q}}$ et fonction de la configuration articulaire \mathbf{q} .

Le calcul de la matrice jacobienne peut se faire en dérivant le MGD, $\mathbf{X} = \mathbf{f}(\mathbf{q})$ à partir de la relation suivante :

$$\mathbf{J}_{ij} = \frac{\partial f_i(\mathbf{q})}{\partial q_j} \quad i = 1, \dots, m ; j = 1, \dots, n \quad (1.23)$$

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

Cette méthode est facile à mettre en œuvre pour des robots à deux ou trois degrés de liberté. En appliquant le MCD sur notre robot représenté dans la figure 1.8.

On note $\mathbf{a}_1, \mathbf{a}_2$ les longueurs des segments. D'après la relation (1.8), la matrice de transformation homogène de MGD est donnée par :

$${}^0\mathbf{T}_2 = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & 0 & a_1 \cos \theta_1 + a_2 \cos(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & a_1 \sin \theta_1 + a_2 \sin(\theta_1 + \theta_2) \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

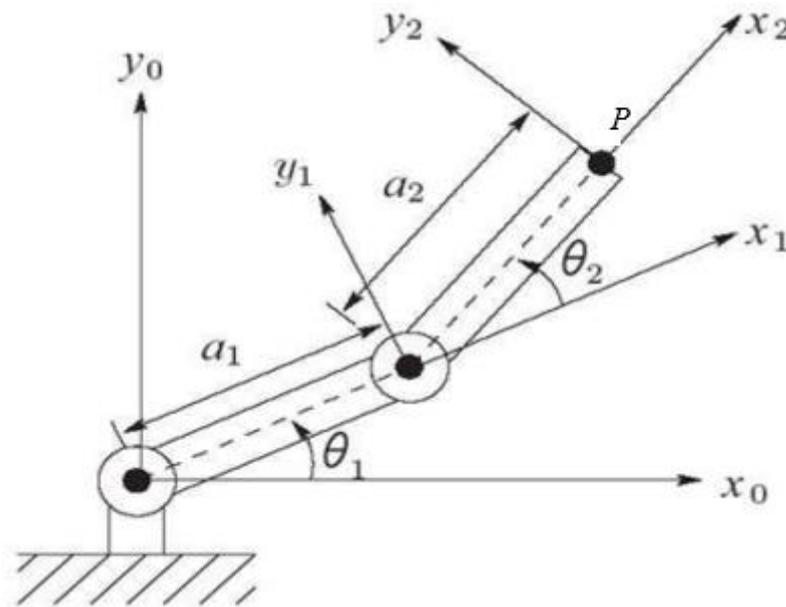


Figure 1.8. Vue de dessus du bras manipulateur à 2 ddl.

A partir de l'expression (1.12), on choisit comme coordonnées opérationnelles du point P les coordonnées (P_x, P_y) dans le plan (x_0, y_0) et les coordonnées articulaires $(\theta_1(t), \theta_2(t))$:

$$P_x = a_1 \cos \theta_1 + a_2 \cos(\theta_1 + \theta_2)$$

$$P_y = a_1 \sin \theta_1 + a_2 \sin(\theta_1 + \theta_2)$$

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

$$\begin{bmatrix} \dot{P}_x \\ \dot{P}_y \end{bmatrix} = \begin{bmatrix} -a_1 S_1 - a_2 S_{12} & -a_2 S_{12} \\ a_1 C_1 + a_2 C_{12} & a_2 C_{12} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \quad (1.24)$$

D'où :

$$J(q) = \begin{bmatrix} -a_1 S_1 - a_2 S_{12} & -a_2 S_{12} \\ a_1 C_1 + a_2 C_{12} & a_2 C_{12} \end{bmatrix} \quad (1.25)$$

Son déterminant vaut :

$$\mathbf{det}(J) = a_1 a_2 \sin \theta_2$$

La droite $\sin \theta_2 = 0$ décompose l'espace articulaire accessible en deux aspects (figure 1.9) correspondant aux deux solutions $\theta_2 > 0$ et $\theta_2 < 0$.

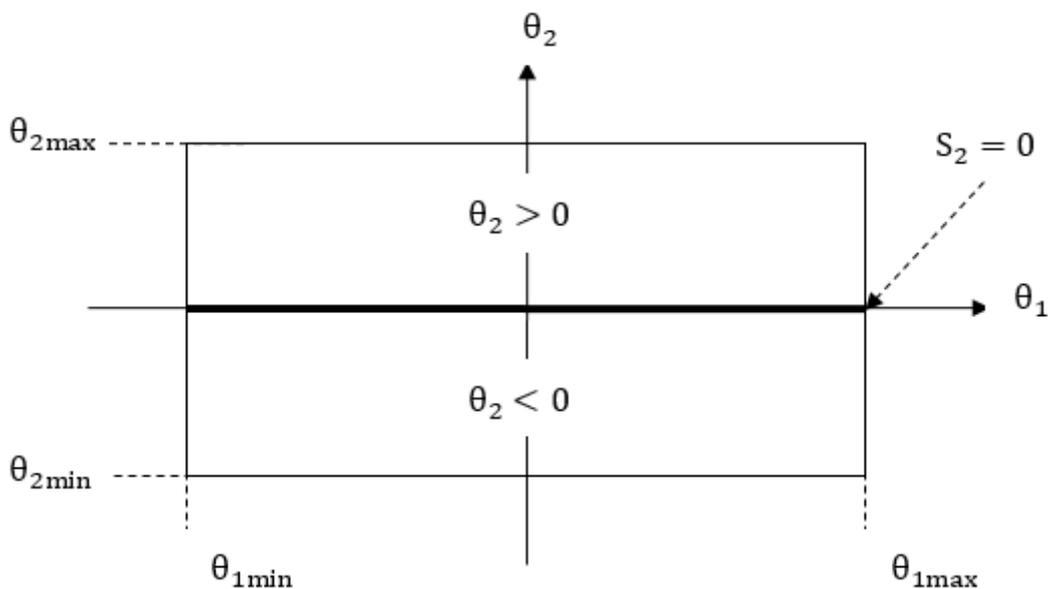


Figure 1.9. Détermination des aspects dans l'espace articulaire.

L'image de ces deux aspects dans l'espace opérationnel, pour des valeurs $\theta_{i\max}$ et $\theta_{i\min}$ arbitraires, est schématisée sur la figure 1.10, les zones hachurées et non hachurées représentent respectivement les lieux des points accessibles pour $\theta_2 > 0$ (posture coude bas), et $\theta_2 < 0$ (posture coude haut).

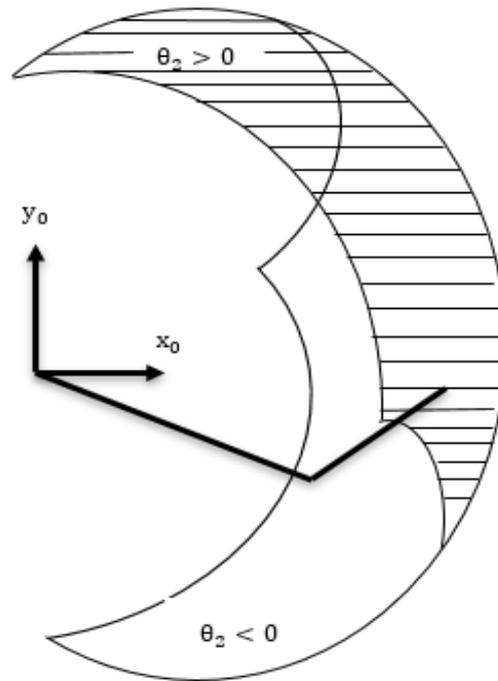


Figure 1.10. Image des aspects dans l'espace opérationnel.

1.3.2 Modèle cinématique inverse (MCI)

L'objectif d'un modèle cinématique inverse est de calculer, à partir d'une configuration \mathbf{q} donnée, les vitesses articulaires $\dot{\mathbf{q}}$ qui assurent au repère terminal une vitesse optimale $\dot{\mathbf{X}}$ imposée. Pour obtenir le MCI, on inverse le MCD en résolvant un système d'équations linéaires. Dans les cas réguliers, la matrice jacobienne \mathbf{J} est carrée d'ordre n et son déterminant est non nul. La méthode la plus générale consiste à calculer \mathbf{J}^{-1} la matrice inverse de \mathbf{J} , qui permet de déterminer les vitesses articulaires $\dot{\mathbf{q}}$ grâce à la relation :

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q}) \dot{\mathbf{X}} \quad (1.26)$$

La résolution du problème se ramène donc à l'inversion. D'après la relation (1.25), la matrice inverse de \mathbf{J} , est la suivante :

$$\mathbf{J}^{-1}(\mathbf{q}) = \frac{1}{a_2 a_2 s_2} \begin{bmatrix} a_2 C_{12} & a_2 S_{12} \\ -a_1 C_1 - a_2 C_{12} & -a_1 S_1 - a_2 S_{12} \end{bmatrix} \quad (1.27)$$

1.4 Modélisation dynamique

La modélisation dynamique est la relation entre les couples (et/ou forces) appliqués aux actionneurs et les positions, vitesses et accélérations articulaires [1]. On représente le modèle dynamique par une relation de la forme :

$$\mathbf{\Gamma} = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{F}) \quad (1.28)$$

Où :

$\mathbf{\Gamma}$: vecteur des couples/forces des actionneurs, selon que l'articulation est rotoïde ou prismatique. Dans la suite, on écrira tout simplement les couples ;

\mathbf{q} : vecteur des positions articulaires ;

$\dot{\mathbf{q}}$: vecteur des vitesses articulaires ;

$\ddot{\mathbf{q}}$: vecteur des accélérations articulaires ;

\mathbf{F} : vecteur représentant l'effort extérieur (forces et moments) qu'exerce le robot sur l'environnement.

Parmi les applications du modèle dynamique, on peut citer :

- La simulation, qui utilise le modèle dynamique direct ;
- Le dimensionnement des actionneurs ;
- L'identification des paramètres inertiels et des paramètres de frottement du robot ;
- La commande, qui utilise le modèle dynamique inverse [1].

1.4.1 Modèle dynamique inverse (MDI)

Le modèle dynamique inverse exprime les couples exercés par les actionneurs en fonction des positions, vitesses et accélérations des articulations. On convient d'appeler MDI, ou tout simplement modèle dynamique, la relation de la forme (1.28), tel que ($q = \theta$) :

$$\mathbf{\Gamma} = \mathbf{f}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}}, \mathbf{F}) \quad (1.29)$$

Plusieurs formalismes ont été utilisés pour obtenir le modèle dynamique des robots. Les formalismes les plus utilisés sont :

- Le formalisme de Lagrange ;
- Le formalisme de Newton-Euler [1].

Nous allons voir dans ce que suit la modélisation dynamique d'un bras manipulateur rigide à chaîne ouverte simple à deux degrés de liberté, on va utiliser le formalisme de Lagrange pour déterminer les équations du mouvement.

➤ **Formalisme de Lagrange :**

Le formalisme de Lagrange décrit les équations du mouvement en termes de travail et d'énergie du système. La forme générale représentant le mouvement d'un robot manipulateur de n degrés de liberté et de n articulations, s'écrit suivant Lagrange comme suit :

$$\Gamma_i = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_i} \right) - \frac{\partial L}{\partial \theta_i} \quad i = 1, 2, \dots, n. \quad (1.30)$$

Avec :

L : lagrangienne du système égal à $E_c - E_p$;

E_c : énergie cinétique total du système ;

E_p : énergie potentielle totale du système [3].

Dans le cas d'une articulation rotoïde $q_i = \theta_i$, tandis que pour une articulation prismatique $q_i = d_i$.

Dans notre travail, nous avons considéré d'abord un bras manipulateur à 2ddl, donné par (figure 1.11), dont le mouvement des articulations est rotoïde, alors $q_i = \theta_i$:

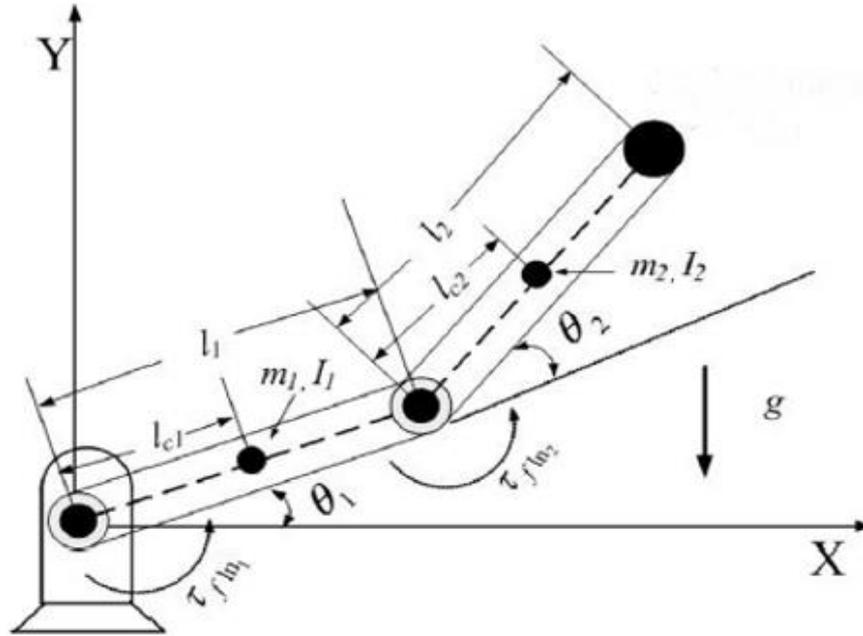


Figure 1.11. Paramètres dynamiques d'un bras manipulateur.

L'expression du couple peut être exprimé sous la forme matricielle suivante (1.31), ce qui correspondant au modèle dynamique du robot [5] :

$$\Gamma_i = \mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \mathbf{f}(\dot{\boldsymbol{\theta}}) + \mathbf{G}(\boldsymbol{\theta}) \quad (1.31)$$

Avec :

$\mathbf{M}(\boldsymbol{\theta})$: matrice d'inertie de dimension (n x n) ;

$\mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$: vecteur de Coriolis/centrifuges de dimension (n x 1) ;

$\mathbf{f}(\dot{\boldsymbol{\theta}})$: vecteur de frottement de dimension (n x 1) ;

$\mathbf{G}(\boldsymbol{\theta})$: vecteur de gravité.

Soit $\Gamma_i = [\tau_{fln1} \quad \tau_{fln2}]^T$, on obtient :

$$\begin{bmatrix} \tau_{fln1} \\ \tau_{fln2} \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} C_{11} \\ C_{21} \end{bmatrix} + \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} \quad (1.32)$$

Où :

$$M_{11} = I_1 + I_2 + m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos \theta_2)$$

$$M_{12} = M_{21} = I_2 + m_2 (l_{c2}^2 + l_1 l_{c2} \cos \theta_2)$$

$$M_{22} = I_2 + m_2 l_{c2}^2$$

$$C_{11} = -m_2 l_1 l_{c2} (2\dot{\theta}_1 + \dot{\theta}_2) \dot{\theta}_2 \sin \theta_2$$

$$C_{21} = m_2 l_1 l_{c2} \dot{\theta}_1^2 \sin \theta_2$$

$$f_1 = b_1 \dot{\theta}_1$$

$$f_2 = b_2 \dot{\theta}_2$$

$$G_1 = m_1 l_{c1} g \cos \theta_1 + m_2 g (l_{c2} \cos(\theta_1 + \theta_2) + l_1 \cos \theta_1)$$

$$G_2 = m_2 l_{c2} g \cos(\theta_1 + \theta_2)$$

Avec :

m_1, m_2 : masses des liens 1 et 2 ;

l_1, l_2 : longueurs des liens 1 et 2 ;

l_{c1}, l_{c2} : les distances des articulations à leur centre de gravité ;

I_1, I_2 : moments d'inertie des liens 1 et 2 par rapport à leur centre de masses ;

b_1, b_2 : coefficients de frottement des liens 1 et 2 [5].

1.4.2 Modèle dynamique direct (MDD)

Pour simuler le comportement du robot et de sa boucle de commande, on utilise le modèle dynamique direct. Où le modèle dynamique direct exprime la position, la vitesse et l'accélération des articulations en fonction du couple appliqué [7]. Il est obtenu par inversion du modèle précédent, il s'écrit :

$$\ddot{\theta} = M^{-1}(\theta) (\Gamma - C(\theta, \dot{\theta})\dot{\theta} - f(\dot{\theta}) - G(\theta)) \quad (1.33)$$

1.5 Conclusion

Nous avons, dans ce chapitre montré comment calculer la matrice de transformation homogène pour trouver la situation d'un solide dans l'espace. Nous avons aussi exposé les différents modèles pour décrire les mouvements d'un bras manipulateur, tels que les modèles géométriques direct et inverse pour exprimer la situation de l'organe terminal en fonction de la configuration du mécanisme et inversement. Les modèles cinématiques direct et inverse pour exprimer la vitesse de l'organe terminal en fonction de vitesse articulaire et inversement. Les modèles dynamiques qui permettent d'établir les relations entre les couples ou forces exercés et les positions, vitesses et accélérations des articulations pour la commande du bras.

Chapitre 2

Méthode d'optimisation:

Teaching-Learning Based Optimization

2.1 Introduction

Un algorithme d'optimisation métaheuristique appelé « Teaching-Learning Based Optimization (TLBO) » est présenté dans ce chapitre. Cet algorithme permet d'obtenir des solutions globales aux problèmes d'optimisations avec une bonne précision et moins de calcul. L'algorithme (TLBO) simule le phénomène d'enseignement-apprentissage afin de résoudre des problèmes multidimensionnels, linéaires et non linéaires avec une efficacité appréciable. Dans notre travail, nous allons utiliser cet algorithme pour déterminer les paramètres du correcteur PID.

2.2 Problème d'optimisation

Un problème d'optimisation au sens général est défini par un ensemble de variables, une fonction objectif f et un ensemble de contraintes d'égalité (ou d'inégalité) que les variables doivent satisfaire. L'ensemble des solutions possibles du problème forme l'espace de recherche, où chaque dimension correspond à une variable. Suivant le problème posé, nous cherchons à minimiser ou maximiser la fonction objectif f . Un problème d'optimisation peut être statique ou dynamique (la fonction objectif change avec le temps), mono-objectif ou multi-objectif (plusieurs fonctions objectifs doivent être optimisées) et avec ou sans contraintes. Ainsi, un problème d'optimisation peut être à la fois continu et dynamique [8].

Il existe nombreuses méthodes déterministes (ou exactes) permettant de résoudre certains types de problèmes d'optimisation et d'obtenir la solution optimale du problème, en un temps raisonnable. Ces méthodes nécessitent que la fonction objectif présente un certain nombre de caractéristiques telles que la convexité ou la dérivabilité. Nous pouvons citer, parmi les méthodes les plus connues, les méthodes de programmation linéaire, quadratique et/ou dynamique, la méthode de Newton, la méthode du simplexe ou encore la méthode du gradient.

2.3 Optimisation métaheuristique

2.3.1 Métaheuristique

Le mot métaheuristique est dérivé de la composition de deux mots grecs : méta, signifiant « au-delà » (ou « à un plus haut niveau ») et heuristique qui signifie « je trouve », il a été inventé par Fred Glover en 1986 [10]. Les métaheuristicques sont des algorithmes génériques, souvent inspirés de la nature, conçues pour résoudre des problèmes d'optimisation complexes sans nécessiter de changements profonds dans l'algorithme employé. La résolution satisfaisante d'un

problème d'optimisation difficile, qui comporte un grand nombre de solutions sous-optimales, justifie souvent le recours à une métaheuristique puissante. La majorité des algorithmes utilisés pour résoudre ces problèmes d'optimisation sont les métaheuristices à base de population. On distingue généralement deux grandes familles de métaheuristices : celles qui manipulent en parallèle toute une population de solutions (on peut citer les algorithmes génétiques, la méthode des colonies de fourmis, l'optimisation par essaim particulaire (PSO), etc.) et les autres qui se basent sur l'évolution itérative d'une solution unique (la méthode Tabou et le Recuit Simulé sont des exemples typiques de ces méthodes), qui sont plutôt axées sur l'exploitation de l'espace de recherche, on n'est donc jamais sûr d'obtenir l'optimum. Au contraire, les métaheuristices à base de population sont plutôt exploratoires et permettent une meilleure diversification de l'espace de recherche. Les métaheuristices sont en général non déterministes, ils donnent des résultats qui peuvent varier d'une exécution à l'autre [11].

2.3.2 Domaine d'application

Les méthodes d'optimisation métaheuristique sont utilisées dans beaucoup de domaines, comme la conception de circuits électroniques, le contrôle aérien, la recherche opérationnelle, la biologie, le fonctionnement d'une machine, gestion des lignes routières ou ferroviaires, choix des investissements et aussi pour répondre aux besoins croissants des secteurs économique et industriel.

2.4 Teaching-Learning Based Optimization Algorithm

Enseignement-apprentissage (Teaching-Learning) est un processus basé sur l'effet de l'influence d'un professeur sur le rendement des apprenants. Rao et al [6] ont proposé un algorithme appelé Teaching-Learning-Based Optimization (TLBO) qui simule le phénomène d'apprentissage traditionnel dans une classe. L'algorithme décrit deux modes d'apprentissage : (i) par l'intermédiaire de professeur, connu sous le nom de phase de professeur et (ii) par l'interaction avec les autres apprenants, connu sous le nom de phase d'apprenants. Dans cet algorithme d'optimisation, un groupe d'apprenants est considéré comme population, les différentes matières proposées aux apprenants sont considérées comme étant les variables du problème d'optimisation et le résultat d'un apprenant est considéré comme étant la valeur de la fonction fitness (fonction d'évaluation). Le professeur est considéré comme étant la meilleure solution dans l'ensemble de la population. Les variables de conception sont les paramètres

impliqués dans la fonction objectif du problème d'optimisation où la meilleure solution est la meilleure valeur de la fonction objectif.

2.4.1 Principe de l'algorithme TLBO

Le fonctionnement de l'algorithme TLBO est divisé en deux parties, 'phase professeur' et 'phase apprenants'. Le fonctionnement des deux phases est expliqué ci-dessous.

Phase professeur :

C'est la première partie de l'algorithme où les apprenants apprennent à travers le professeur. Pendant cette phase, un professeur essaie d'augmenter le résultat moyen de la classe dans la matière enseignée par lui. A n'importe quelle itération i , supposons qu'il existe m nombre de matières (variables d'optimisation), n nombre d'apprenants (taille de la population, $k = 1, 2, \dots, n$) et $M_{j,i}$ être le résultat moyen des apprenants dans une matière particulière j ($j = 1, 2, \dots, m$) Le meilleur résultat global $X_{total-kbest,i}$ considérant toutes les matières obtenues dans l'ensemble de la population d'apprenants peut être considéré comme le résultat du meilleur apprenant $kbest$. Cependant, comme le professeur est généralement peut avoir le meilleur résultat, l'algorithme identifie le meilleur apprenant comme le professeur. La différence entre le résultat moyen de chaque matière et le résultat de professeur pour chaque matière est donnée par :

$$Difference_Mean_{j,k,i} = r_i (X_{j,kbest,i} - T_F M_{j,i}) \quad (2.1)$$

Où, $X_{j,kbest,i}$ est le résultat du meilleur apprenant dans la matière j . T_F est le facteur d'enseignement, et r_i est le nombre aléatoire dans l'intervalle $[0, 1]$. La valeur de T_F est obtenue aléatoirement avec une probabilité égale à :

$$T_F = round [1 + rand (0, 1) \{2-1\}] \quad (2.2)$$

En basant sur $Difference_Mean_{j,k,i}$, la solution est mise à jour dans la phase professeur selon l'expression suivante :

$$X'_{j,k,i} = X_{j,k,i} + Difference_Mean_{j,k,i} \quad (2.3)$$

$X'_{j,k,i}$ est la mise à jour de la valeur $X_{j,k,i}$. La solution $X'_{j,k,i}$ est accepté si elle donne une meilleur valeur de la fonction objectif. A la fin de la phase professeur, toutes les valeurs acceptées sont conservées et deviennent les entrées de la phase apprenants. La phase apprenants dépend de la phase professeur.

Phase apprenants :

C'est la deuxième partie de l'algorithme, les apprenants améliorent leurs connaissances en interagissant entre eux. On doit sélectionner au hasard deux apprenants P et Q tels que $X'_{total-P,i} \neq X'_{total-Q,i}$ ($X''_{total-P,i}$ et $X''_{total-Q,i}$ sont les mise à jour de $X'_{total-P,i}$ et $X'_{total-Q,i}$)

$$X''_{j-P,i} = X'_{j-P,i} + r_i (X'_{j-P,i} - X'_{j-Q,i}), \text{ Si } X'_{total-P,i} < X'_{total-Q,i} \quad (2.4)$$

$$X''_{j-P,i} = X'_{j-P,i} + r_i (X'_{j-Q,i} - X'_{j-P,i}), \text{ Si } X'_{total-Q,i} < X'_{total-P,i} \quad (2.5)$$

$X''_{j-P,i}$ est accepté si elle donne une meilleur valeur de la fonction objectif.

L'organigramme de l'algorithme TLBO est présenté dans la figure suivante (figure 2.1).

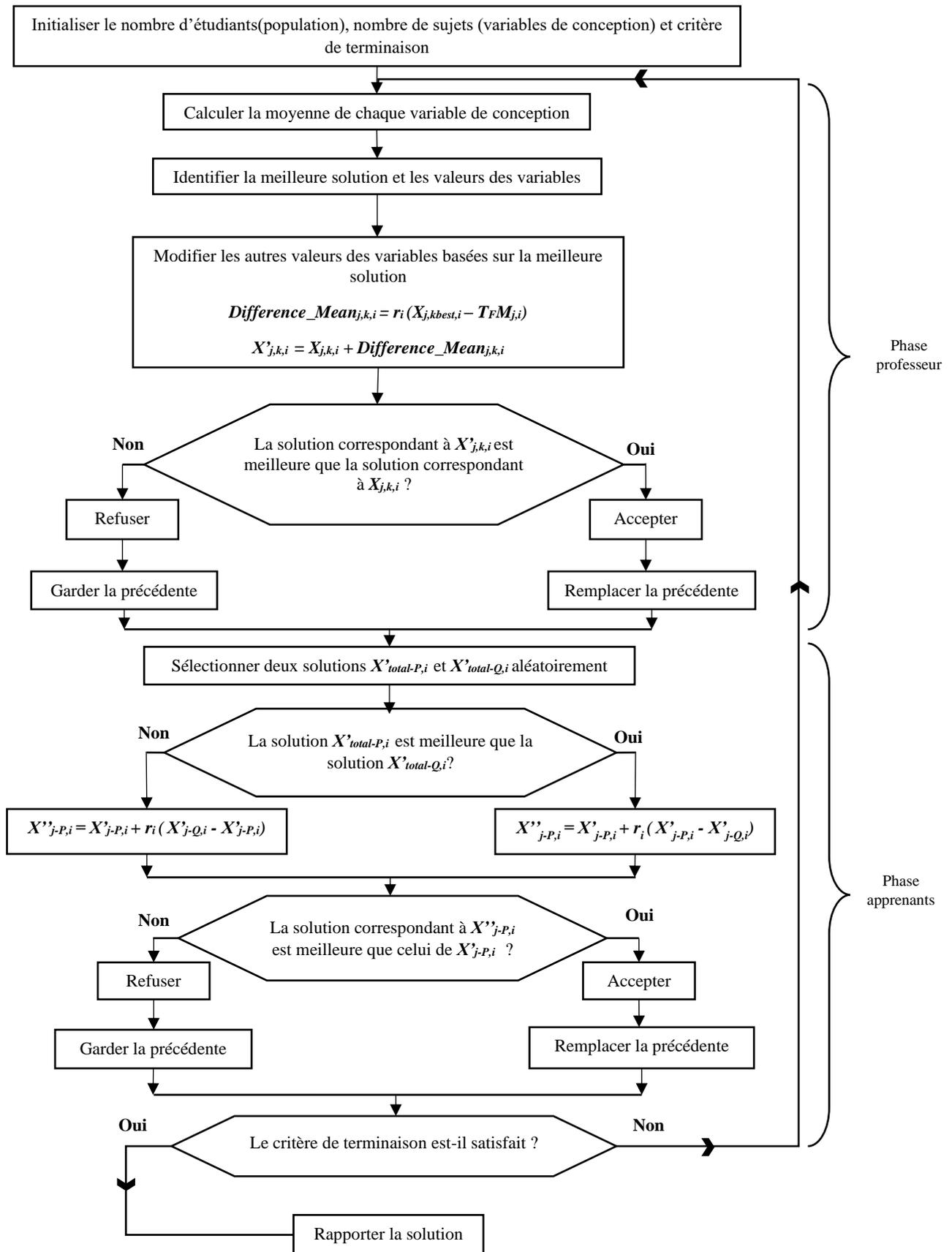


Figure 2.1. Organigramme d'algorithme TLBO.

2.5 Exemples d'optimisation en utilisant la méthode TLBO

Après avoir codé l'algorithme TLBO en utilisant le logiciel Matlab, nous l'utilisons pour trouver le minimum global des deux fonctions f et g . Les valeurs de x_i qui minimise ces deux fonctions de coût sont représentées sur les figures 2.2 et 2.3 :

$$f(x) = x^2$$

$$\text{Min}(f) = 0 \quad \text{Pour } x = 0$$

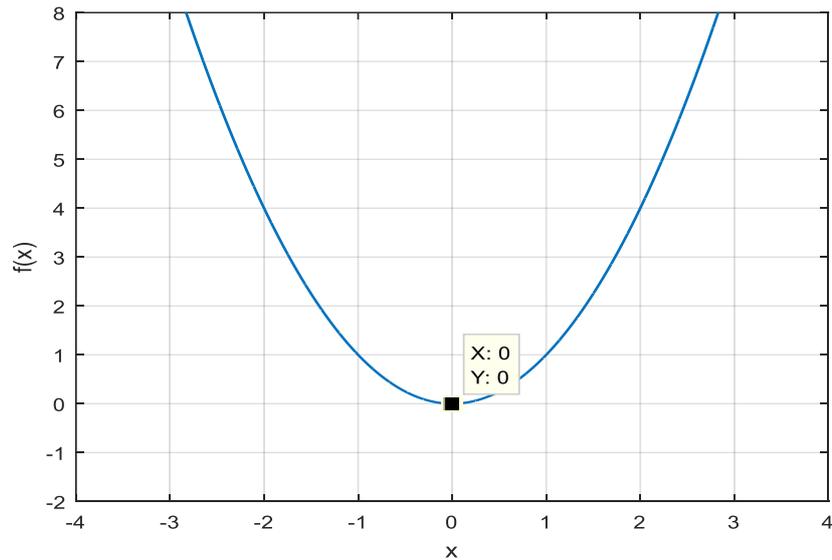


Figure 2.2. Le minimum de la fonction f

$$g(x) = 100(x^2 - x)^2 + (1 - x)^2$$

$$\text{Min}(g) = 0 \quad \text{Pour } x = 1$$

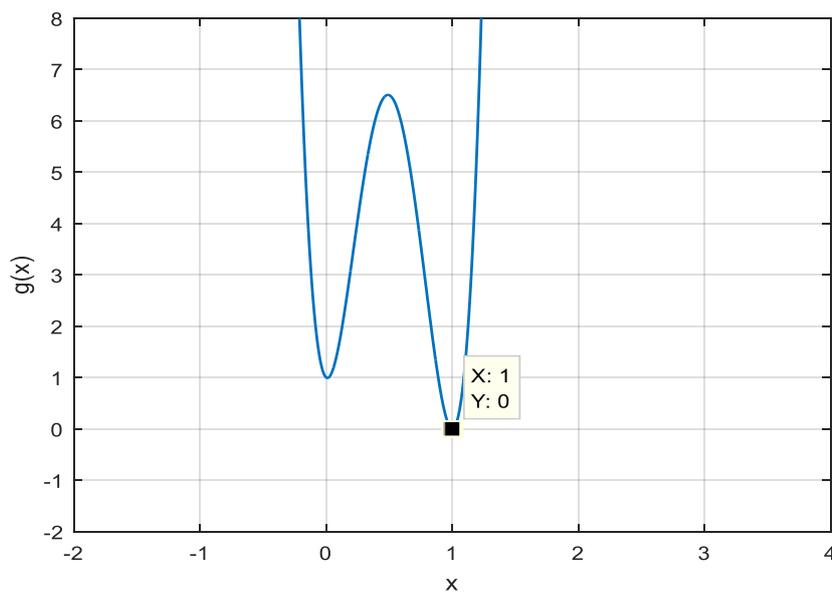


Figure 2.3. Le minimum de la fonction g

Dans l'exemple suivant nous supposons que l'intervalle de recherche est donné par $(-100 \leq x_i \leq 100)$ et la taille de la population est variée entre 15 et 30 apprenants et le nombre d'itérations égale à 20 (tableau 2.1). Ensuite, nous fixons la taille de la population à 20 apprenants et nous faisons varier le nombre d'itérations jusqu'à 100 itérations. Les résultats obtenus sont regroupés dans le tableau 2.2.

Population	15		25		30	
Fonction	f	g	f	g	f	g
Minimum Global	1.9189e-19	1.6341e-10	3.3883e-25	4.044e-17	4.0332e-30	8.6945e-20
Valeur de x	-7.2003e-06	0.89696	-4.2754e-07	0.91539	-1.2843e-07	0.98479

Tableau 2.1. Le minimum global avec TLBO pour plusieurs valeurs de population.

Itérations	30		50		100	
Fonction	f	g	f	g	f	g
Minimum Global	4.1386e-27	4.7147e-23	1.4881e-46	1.792e-28	7.1264e-90	4.9797e-30
Valeur de x	-1.0099e-08	0.99981	-4.5647e-17	0.99847	-1.0236e-40	1

Tableau 2.2. Le minimum global avec TLBO pour plusieurs nombres d'itérations.

Lorsque nous augmentons la taille de la population et le nombre d'itérations, la méthode TLBO converge vers un minimum global plus précis.

2.5.1 Comparaison entre TLBO et PSO

Pour mettre en évidence l'efficacité de l'algorithme TLBO, nous considérons la minimisation de six fonctions. Les résultats obtenus en utilisant l'algorithme TLBO sont comparés à ceux obtenus en utilisant l'algorithme PSO. Les détails des fonctions considérées sont donnés dans le tableau 2.3.

Fonctions	TLBO	PSO
Sphere : $f_{min} = \sum_{i=1}^D x_i^2$	$Min f = 0.00$ Pour $x = 0.00$	$Min f = 7.96 e-51$ Pour $x = 3.56 e-50$
Rosenbrock : $f_{min} = \sum_{i=1}^D [100 (x_i^2 - x_{i+1})^2 + (1 - x_i)^2]$	$Min f = 1.72 e+00$ Pour $x = 6.62 e-01$	$Min f = 3.08 e+00$ Pour $x = 7.69 e-01$
Ackley : $f_{min} = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^D \cos 2\pi x_i \right) + 20 + e$	$Min f = 3.55 e-15$ Pour $x = 8.32 e-31$	$Min f = 1.58 e-14$ Pour $x = 1.60 e-14$
Griewank : $f_{min} = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	$Min f = 0.00$ Pour $x = 0.00$	$Min f = 9.69 e-02$ Pour $x = 5.01 e-02$
Rastrigin : $f_{min} = \sum_{i=1}^D [x_i^2 - 10 \cos 2\pi x_i + 10]$	$Min f = 6.77 e-08$ Pour $x = 3.68 e-07$	$Min f = 5.82 e+00$ Pour $x = 2.96 e+00$
Schwefel : $f_{min} = -\sum_{i=1}^D (x_i \sin \sqrt{ x_i })$	$Min f = 2.94 e+02$ Pour $x = 2.68 e+02$	$Min f = 3.20 e+02$ Pour $x = 1.85 e+02$

Tableau 2.3. Résultats comparatifs entre l'algorithme TLBO et l'algorithme PSO [9].

2.6 Conclusion

Dans ce chapitre, nous avons étudié l'algorithme d'optimisation TLBO. Les résultats de comparaison obtenus montrent l'efficacité et la précision de l'algorithme TLBO.

Chapitre 3

Simulation de la commande d'un bras manipulateur à deux degrés de liberté en utilisant des régulateurs PID

3.1 Introduction

Le régulateur PID est bien connu dans l'industrie et convient en fait à tout type de système linéaire invariant dans le temps [17]. L'objectif de ce chapitre est de présenter les résultats de simulation de la commande d'un bras manipulateur à deux degrés de liberté en utilisant un régulateur PID. Les paramètres de ce régulateur sont obtenus en optimisant un critère de performance portant sur l'erreur de commande à l'aide de l'algorithme d'optimisation TLBO introduit dans le chapitre précédent. Dans ce chapitre, nous rappellerons brièvement la notion de correction de systèmes et les différents correcteurs classiques et nous présentons les résultats de commande obtenus pour différentes trajectoires de référence.

3.2 Correction des systèmes

La correction des systèmes peut être abordée selon deux approches; la première se fonde uniquement sur la valeur désirée à la sortie du processus (commande en boucle ouverte) et la seconde se base également sur la valeur désirée mais aussi sur la sortie du système réellement obtenue (commande en boucle fermée)

3.2.1 Correction en boucle ouverte

Quand la commande est élaborée sans l'aide de la connaissance des grandeurs de sorties, la boucle est dite ouverte, comme on peut voir dans la figure 3.1. Dans le cas d'une correction en boucle ouverte, la grandeur de commande u dépend que du signal de référence r , et le correcteur ne reçoit pas d'information sur la sortie réelle du système y .

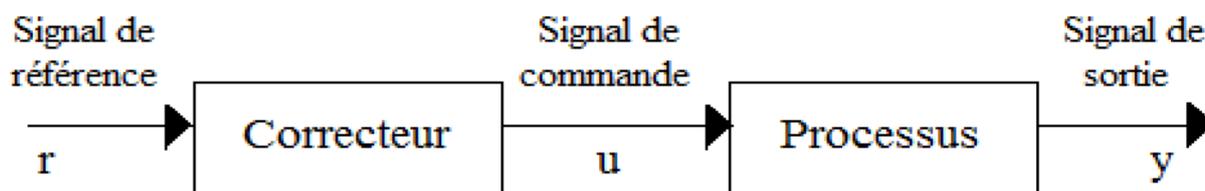


Figure 3.1. Correction en boucle ouverte.

3.2.2 Correction en boucle fermée

Dans ce cas, la sortie du procédé est comparée à la référence, et en fonction du résultat obtenu, le correcteur va agir sur le processus. Ce mode de correction est le plus robuste contre les perturbations. La figure 3.2 représente le schéma bloc d'un asservissement en boucle fermée.

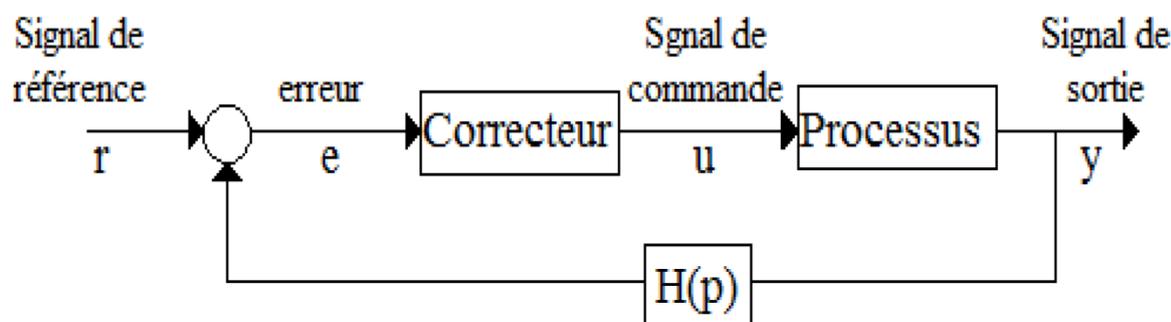


Figure 3.2. Correction en boucle fermée.

3.3 Correcteurs classiques

3.3.1 Correcteur proportionnel (P)

Le régulateur à action proportionnelle minimise l'écart entre la consigne et la mesure, il réduit le temps de monter et le temps de réponse. La fonction de transfert du correcteur est donnée par :

$$G(p) = K_p \quad (3.1)$$

L'action proportionnelle consiste à générer une action qui varie de façon proportionnelle au signal d'erreur :

$$u(t) = K_p e(t) = K_p [r(t) - y(t)] \quad (3.2)$$

Où :

$u(t)$: est la grandeur de commande (sortie du régulateur).

K_p : le gain du régulateur.

$r(t)$: la référence.

$y(t)$: la sortie du processus.

3.3.2 Correcteur proportionnel intégrateur (PI)

A la correction proportionnelle, nous pouvons ajouter l'intégration de l'erreur. Dans ce cas nous obtenons une régulation PI (proportionnel et intégral). La loi de commande est, dans ce cas, donnée par :

$$u(t) = K_p e(t) + K_I \int_0^t e(\tau) d\tau \quad (3.3)$$

Où : K_I représente le gain d'intégration

La fonction de transfert du correcteur est la suivante :

$$G(p) = K_p + \frac{K_I}{p} \quad (3.4)$$

Avec la présence du terme intégrateur, le problème de saturation apparaît. Si la commande de l'actionneur est limitée à une valeur maximale, la sortie du correcteur peut se saturer.

3.3.3 Correcteur proportionnel dérivateur (PD)

L'action de dérivation a un grand potentiel pour améliorer les performances du système, elle anticipe la tendance de l'erreur et essaye de corriger. Mais son implémentation pose quelques problèmes; il faut bien filtrer la sortie du processus pour éviter les composantes hautes fréquences (bruit) [13].

La loi de commande PD est donnée par :

$$u(t) = K_p \left(e(t) + T_D \frac{de(t)}{dt} \right) \quad (3.5)$$

Où : T_D représente le gain de dérivation

La fonction de transfert du régulateur PD est donnée par :

$$G(p) = K_p (1 + T_D p) \quad (3.6)$$

3.3.4 Correcteur proportionnel intégral dérivé PID)

Un régulateur PID est obtenu par l'association de ces trois actions (proportionnelle, intégrale et dérivée) et il remplit essentiellement les trois fonctions suivantes :

- Il fournit un signal de commande en tenant compte de l'évolution du signal de sortie par rapport à la consigne.
- Il élimine l'erreur statique grâce au terme intégrateur.

➤ Il anticipe les variations de la sortie grâce au terme dérivateur.

La commande $u(t)$ peut s'écrire, dans ce cas, sous la forme suivante :

$$u(t) = K_P[e(t) + \frac{1}{T_I} \int_0^t e(t)dt + T_D \frac{de(t)}{dt}] \quad (3.7)$$

Avec l'écart défini comme suit :

$$e(t) = r(t) - y(t) \quad (3.8)$$

Sa fonction de transfert s'écrit alors :

$$G(p) = \frac{u(p)}{e(p)} = K_P[1 + \frac{1}{T_I p} + T_D p] \quad (3.9)$$

Le PID classique a des limites malgré ses bonnes performances [14]. Les performances du système seront dégradées lorsque les perturbations varient d'une manière importante.

3.4 Résultats de simulation

Nous présentons, dans cette section, les résultats de commande d'un bras manipulateur à deux degrés de liberté pour plusieurs trajectoires de référence.

3.4.1 Modèle dynamique du bras

Le modèle dynamique de bras manipulateur SCARA considéré est donné par [5] :

$$\begin{bmatrix} \tau_{fn1} \\ \tau_{fn2} \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} C_{11} \\ C_{21} \end{bmatrix} + \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} \quad (3.10)$$

Où :

$$M_{11} = I_1 + I_2 + m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos\theta_2)$$

$$M_{12} = M_{21} = I_2 + m_2 (l_{c2}^2 + l_1 l_{c2} \cos\theta_2)$$

$$M_{22} = I_2 + m_2 l_{c2}^2$$

$$C_{11} = -m_2 l_1 l_{c2} (2\dot{\theta}_1 + \dot{\theta}_2) \dot{\theta}_2 \sin\theta_2$$

$$C_{21} = m_2 l_1 l_{c2} \dot{\theta}_1^2 \sin\theta_2$$

$$f_1 = b_1 \dot{\theta}_1$$

$$f_2 = b_2 \dot{\theta}_2$$

$$G_1 = m_1 l_{c1} g \cos\theta_1 + m_2 g (l_{c2} \cos(\theta_1 + \theta_2) + l_1 \cos\theta_1)$$

$$G_2 = m_2 l_{c2} g \cos(\theta_1 + \theta_2)$$

Chapitre 3 Simulation de la commande d'un bras manipulateur à 2 ddl en utilisant des régulateurs PID

Avec :

θ_1, θ_2 sont les positions articulaires des liens 1 et 2 ;

τ_{fn1}, τ_{fn2} sont les couples/forces des articulations 1 et 2 ;

m_1, m_2 sont les masses des liens 1 et 2 ;

l_1, l_2 sont les longueurs des liens 1 et 2 ;

l_{c1}, l_{c2} sont les distances des articulations à leur centre de gravité ;

I_1, I_2 sont les moments d'inertie des liens 1 et 2 par rapport à leur centre de masses ;

b_1, b_2 sont les coefficients de frottement des liens 1 et 2.

Les valeurs des différents paramètres du bras sont données dans le tableau 3.1.

Paramètres	Lien 1	Lien 2
Masse	0.392924 kg	0.094403 kg
Accélération due à la gravité	9.81 m/s ²	9.81 m/s ²
Longueur	0.2032 m	0.1524 m
Distance de l'articulation à son centre de gravité	0.104648 m	0.081788 m
Moment d'inertie	0.0011411kgm ²	0.0020247kgm ²
Coefficient de frottement	0.141231 N-m/radian/s	0.3530776 N-m/radian/s

Tableau 3.1. Paramètres du système SCARA [5].

3.4.2 Paramètres des régulateurs PID utilisés

Nous utilisons pour chacune des articulations du bras un régulateur PID. Les valeurs des paramètres K_p , T_I , et T_D de chaque régulateur sont obtenues en minimisant le critère de performances donné par l'équation (3.11) à l'aide de l'algorithme d'optimisation TLBO.

$$f_i(t) = \sum_{k=1}^N e_i^2(k), \quad i = 1, 2 \quad (3.11)$$

Où :

N est le nombre d'échantillons.

$$e_1 = r_1 - \theta_1 \text{ et } e_2 = r_2 - \theta_2$$

Les valeurs des paramètres de l'algorithme TLBO utilisées pour chacune des deux articulations sont :

$$N = 1000$$

Taille de la population : 20

Nombre d'itérations : 150

Nombre des variables d'optimisation : 3 (K_P , T_I , T_D).

Les valeurs obtenues des paramètres des deux régulateurs PID sont données dans le tableau 3.2 suivant :

Paramètres	K_P	T_I	T_D
Première articulation	0.2270	-0.0418	0.3833
Deuxième articulation	0.3052	0.0960	0.1931

Tableau 3.2. Valeurs des paramètres des régulateurs PID obtenues en utilisant l'algorithme TLBO.

3.4.3 Résultats obtenus pour des références en échelon

Les trajectoires de référence pour les deux positions angulaires θ_1 et θ_2 sont choisies des échelons unitaires. Les résultats de poursuites des trajectoires sont donnés par les figures 3.3 et 3.4.

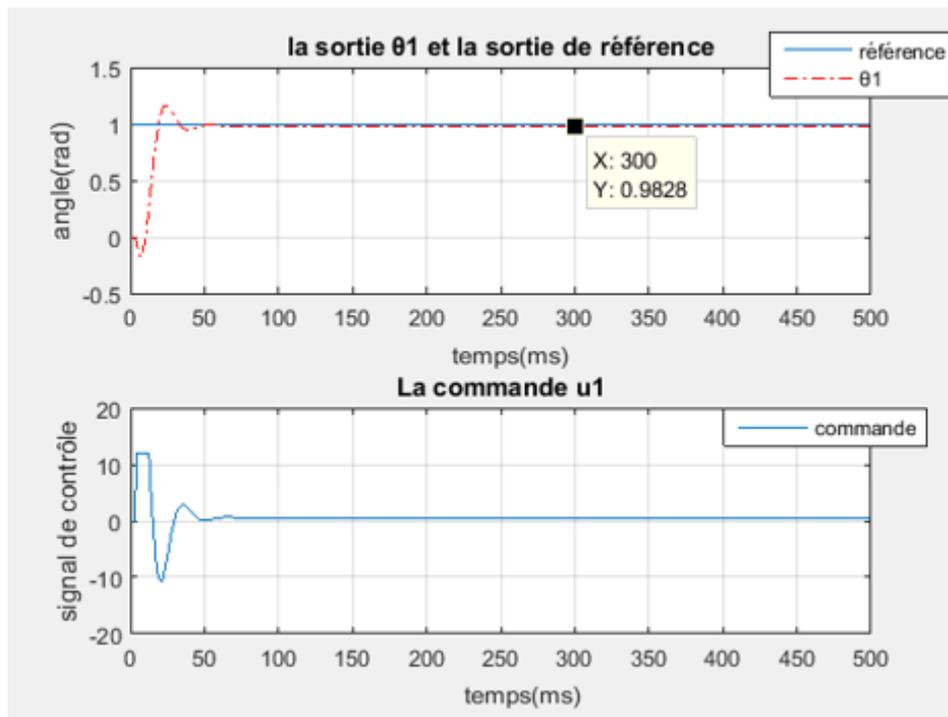


Figure 3.3. Performances de commande pour la première articulation dans le cas d'une trajectoire en échelon.

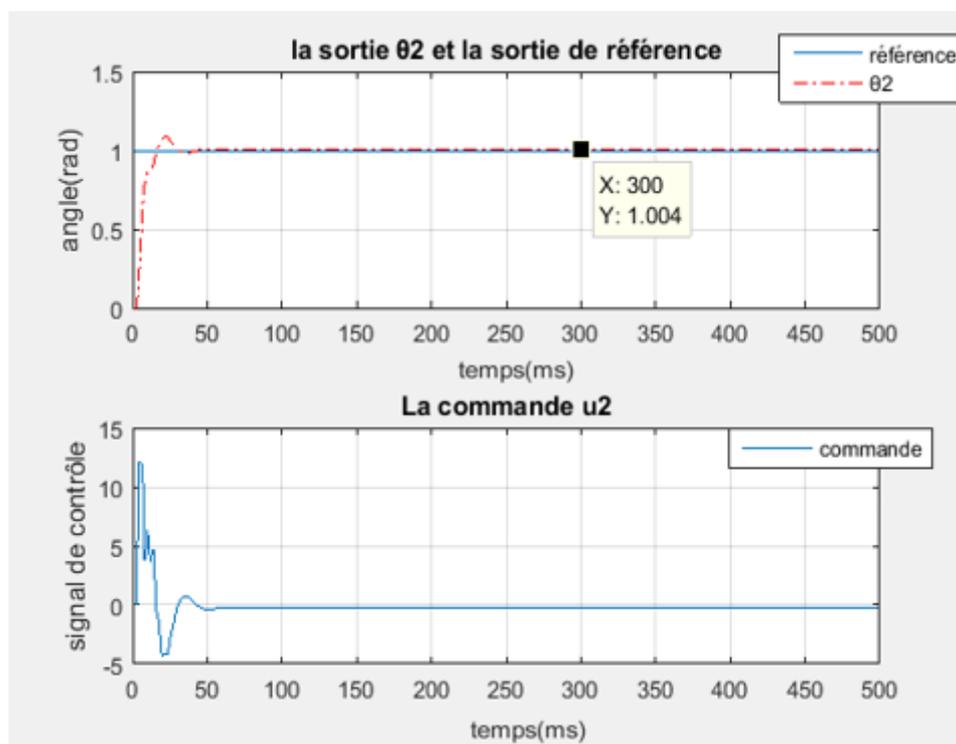


Figure 3.4. Performances de commande pour la deuxième articulation dans le cas d'une trajectoire en échelon.

L'écart entre la position angulaire réelle du bras et le signal de référence, pour les deux articulations, est représenté dans les figures 3.5 et 3.6.

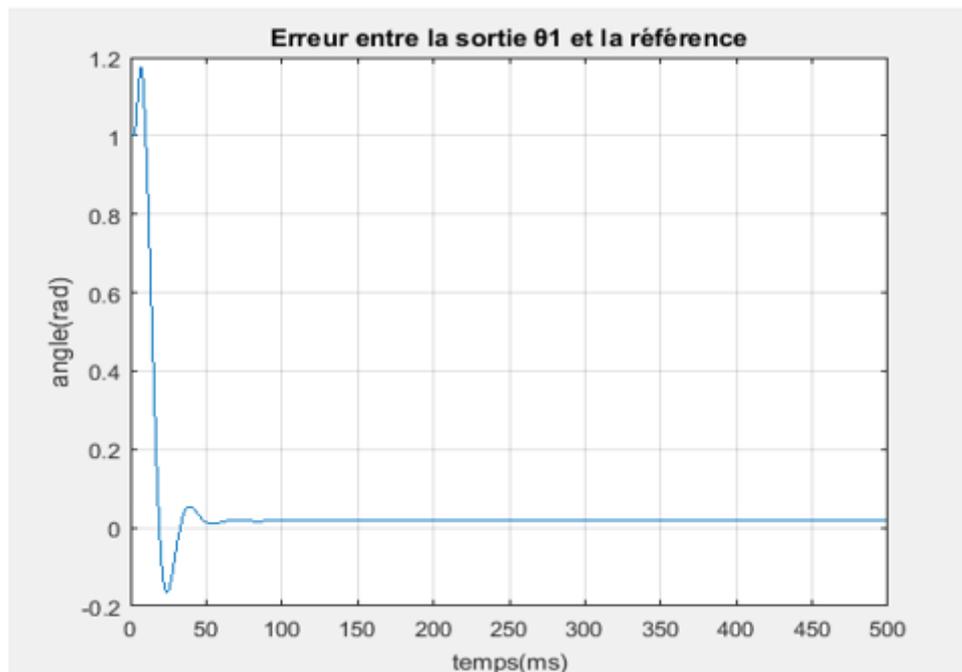


Figure 3.5. Erreur de poursuite pour la première articulation dans le cas d'une trajectoire en échelon.



Figure 3.6. Erreur de poursuite pour la deuxième articulation dans le cas d'une trajectoire en échelon.

A partir des figures précédentes nous remarquons qu'il y a une bonne poursuite des trajectoires de référence dès les premières secondes et avec des dépassements de faible amplitude, et nous remarquons aussi la convergence de l'erreur vers une valeur minimale de 0.02 rad pour la première articulation et de -0.004 rad pour la deuxième articulation.

3.4.4 Résultats obtenus pour des références sinusoïdales

Dans le but de mettre en évidence les performances des régulateurs obtenus, nous avons choisi dans ce cas des trajectoires de références sinusoïdales. Les résultats de poursuite obtenus sont donnés par les figures 3.7 et 3.8.

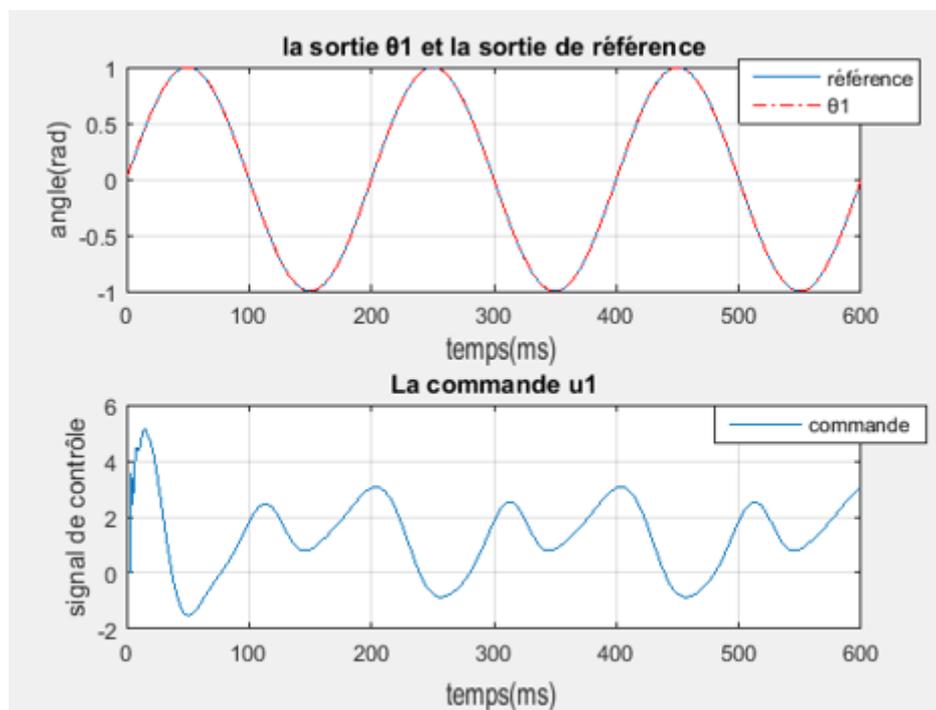


Figure 3.7. Performances de commande pour la première articulation dans le cas d'une trajectoire sinusoïdale.

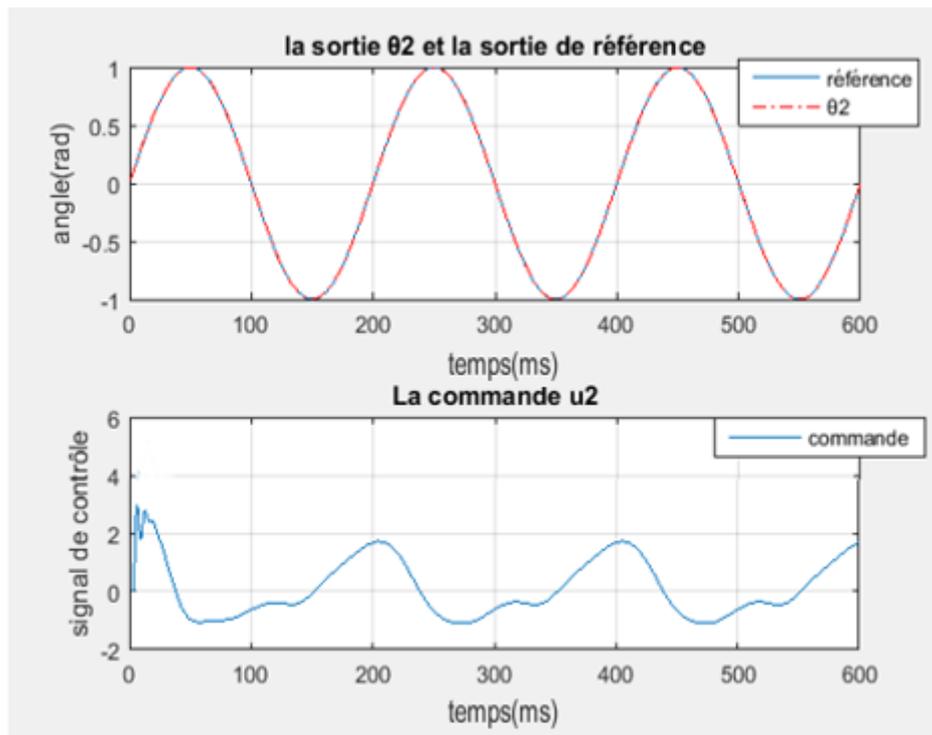


Figure 3.8. Performances de commande pour la deuxième articulation dans le cas d'une trajectoire sinusoïdale.

A partir de ces figures nous pouvons conclure qu'il y a une bonne poursuite des trajectoires de références choisies.

3.4.5 Résultats obtenus pour des références multi échelon

Les trajectoires de référence choisies dans ce cas sont constituées par des échelons d'amplitudes différentes. Les figures 3.9 et 3.10 illustrent les résultats obtenus.

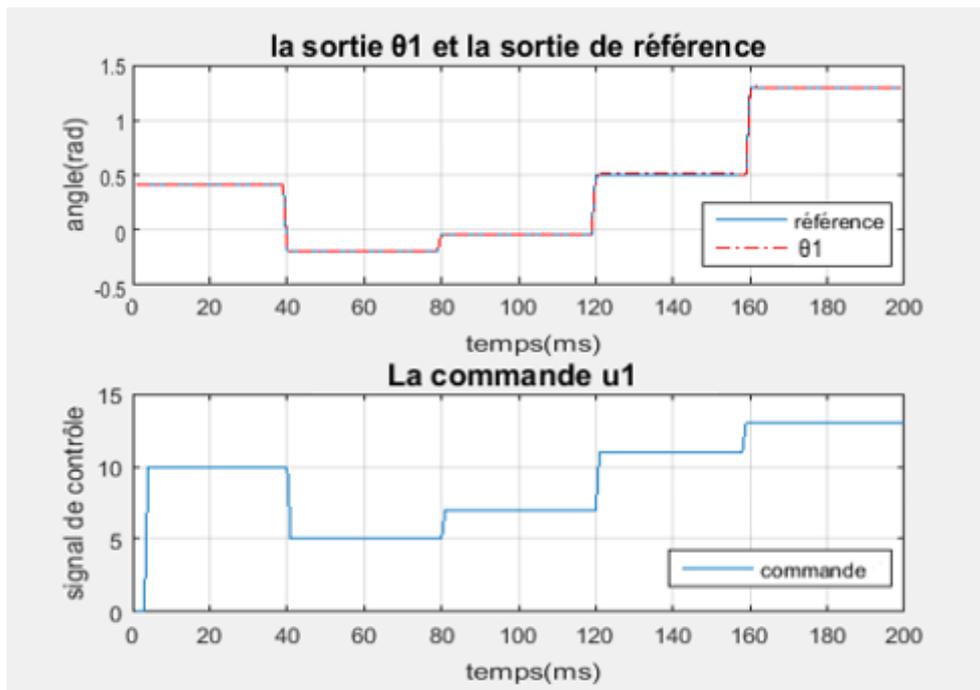


Figure 3.9. Performances de commande pour la première articulation dans le cas d'une trajectoire en échelon multiple.

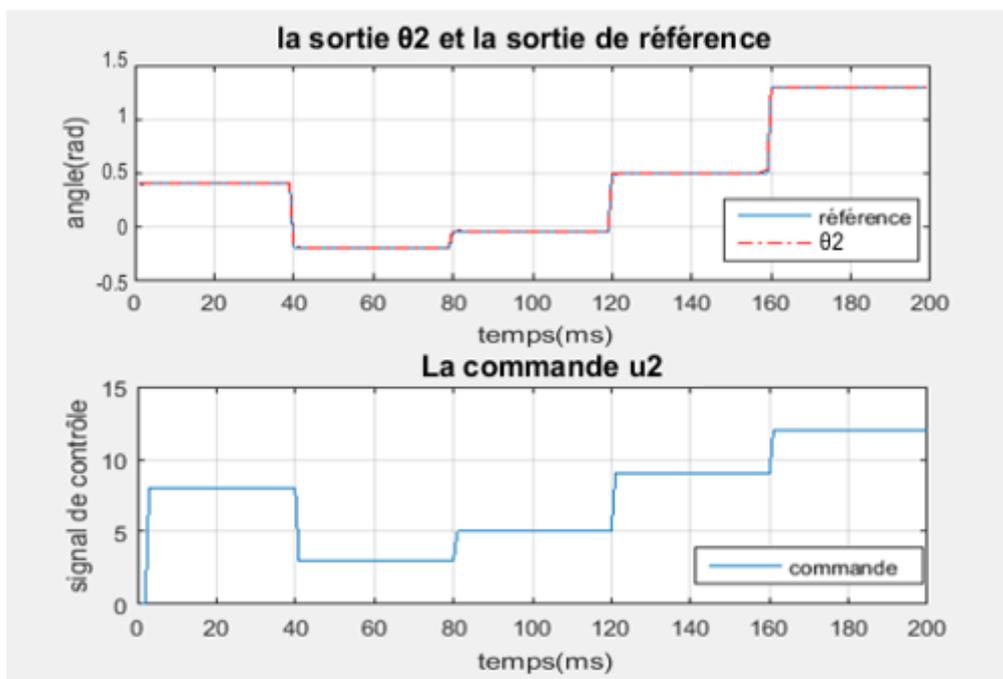


Figure 3.10. Performances de commande pour la deuxième articulation dans le cas d'une trajectoire en échelon multiple.

Dans ce cas nous remarquons qu'il y a aussi une bonne poursuite des trajectoires de référence choisies.

3.4.6 Poursuite des trajectoires dans l'espace opérationnel

Les figures 3.11 et 3.12 montrent la poursuite, dans l'espace opérationnel, d'un cercle et d'une ligne droite, respectivement.

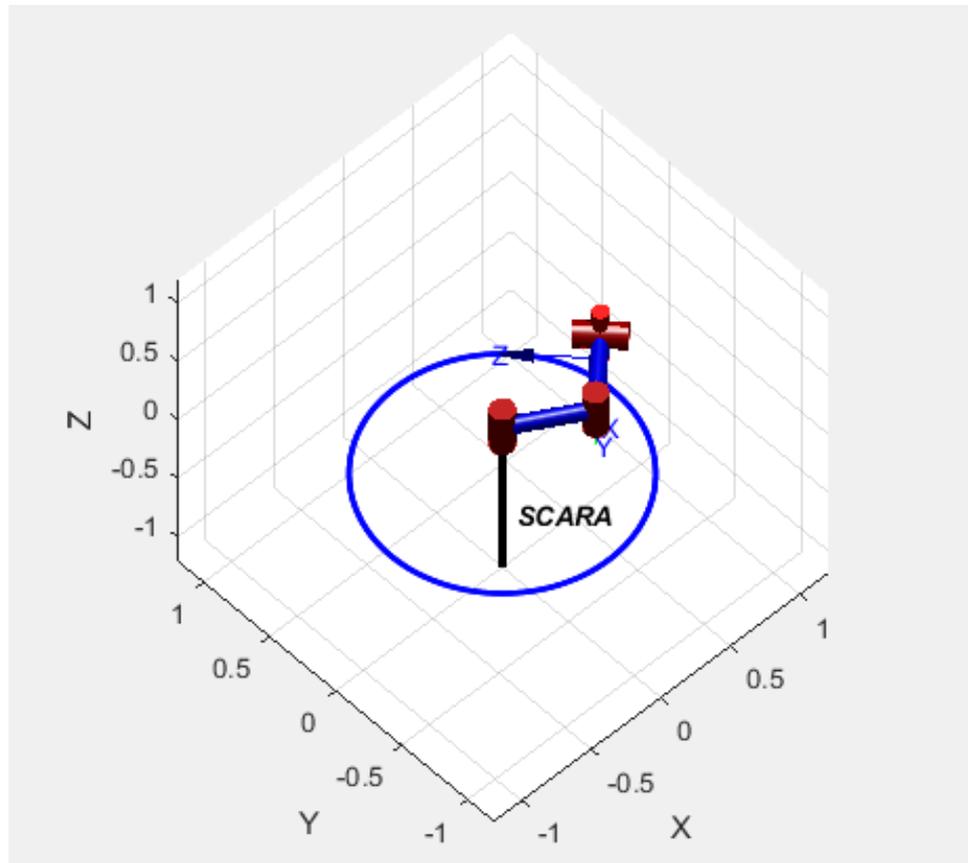


Figure 3.11. Poursuite d'un cercle dans l'espace opérationnel.

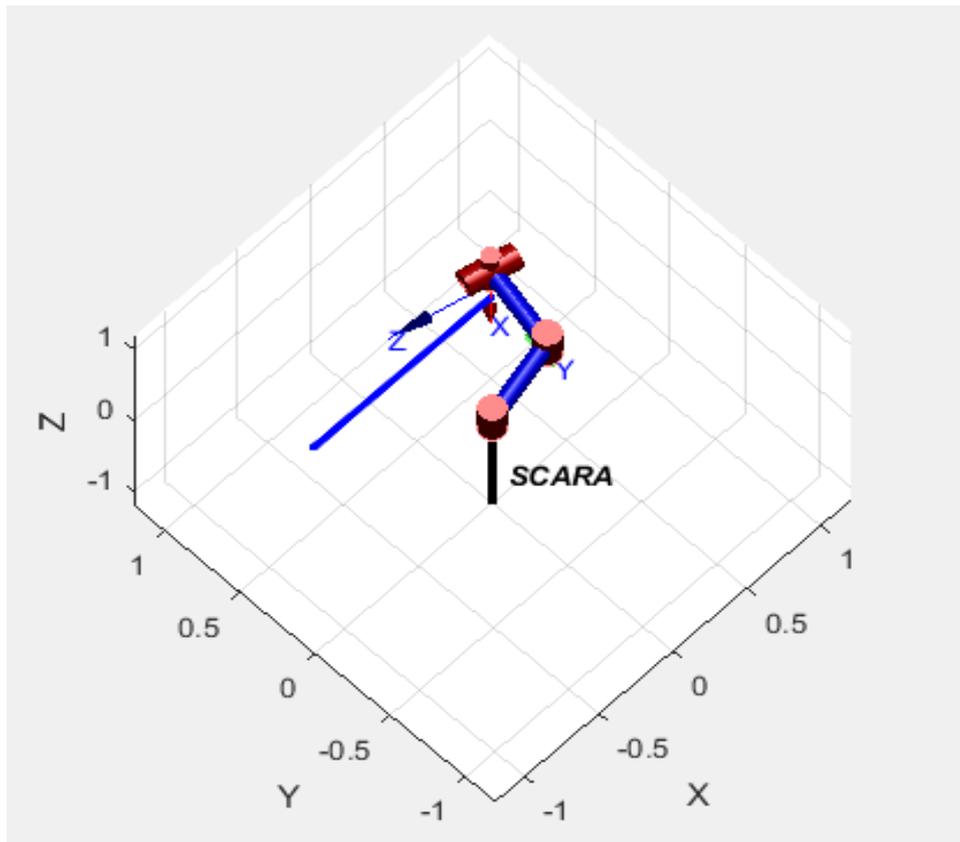


Figure 3.12. Poursuite d'une ligne droite dans l'espace opérationnel.

3.5 Conclusion

Dans ce chapitre, nous avons mis en évidence les performances de commande d'un bras manipulateurs en utilisant des régulateurs PID optimisés à l'aide de l'algorithme TLBO. Les résultats des différentes simulations montrent que les régulateurs obtenus possèdent des bonnes performances PID en termes de précision de poursuite, de rapidité et de valeurs de dépassement. Nous reprendrons dans le chapitre suivant cette étude d'une manière pratique en considérant la commande d'un bras manipulateur réel.

Chapitre 4

Conception, réalisation et commande pratique d'un bras manipulateur

4.1 Introduction

Nous considérons, dans ce chapitre, la conception et la réalisation d'un bras manipulateur à deux degrés de liberté de type SCARA ainsi que la mise en œuvre pratique de sa commande en utilisant des régulateurs PID optimisés à l'aide de l'algorithme TLBO. Les algorithmes de commande sont implémentés numériquement en utilisant la carte de développement *eZdsp TMS320F28335*. Nous présentons, dans ce chapitre les résultats pratiques de l'identification par réseaux de neurones des systèmes obtenu ainsi que ceux de la commande du bras manipulateurs réalisés.

4.2 Conception du bras manipulateur

La structure du robot SCARA à réaliser, donnée par la figure 4.1, est constituée de 2 articulations connectées les unes aux autres. Les liens 1 et 2 se déplacent dans le plan xy, tandis que l'effecteur final se déplacé le long de l'axe z. L'articulation 1, reliée à la base, s'appelle épaule en raison de son mouvement. Elle peut faire une rotation de 150 degrés de chaque côté gauche et droit. L'articulation 2 qui se raccorde à l'articulation 1 est appelée coude. Elle peut faire une rotation de 300 degrés comme l'articulation 1.

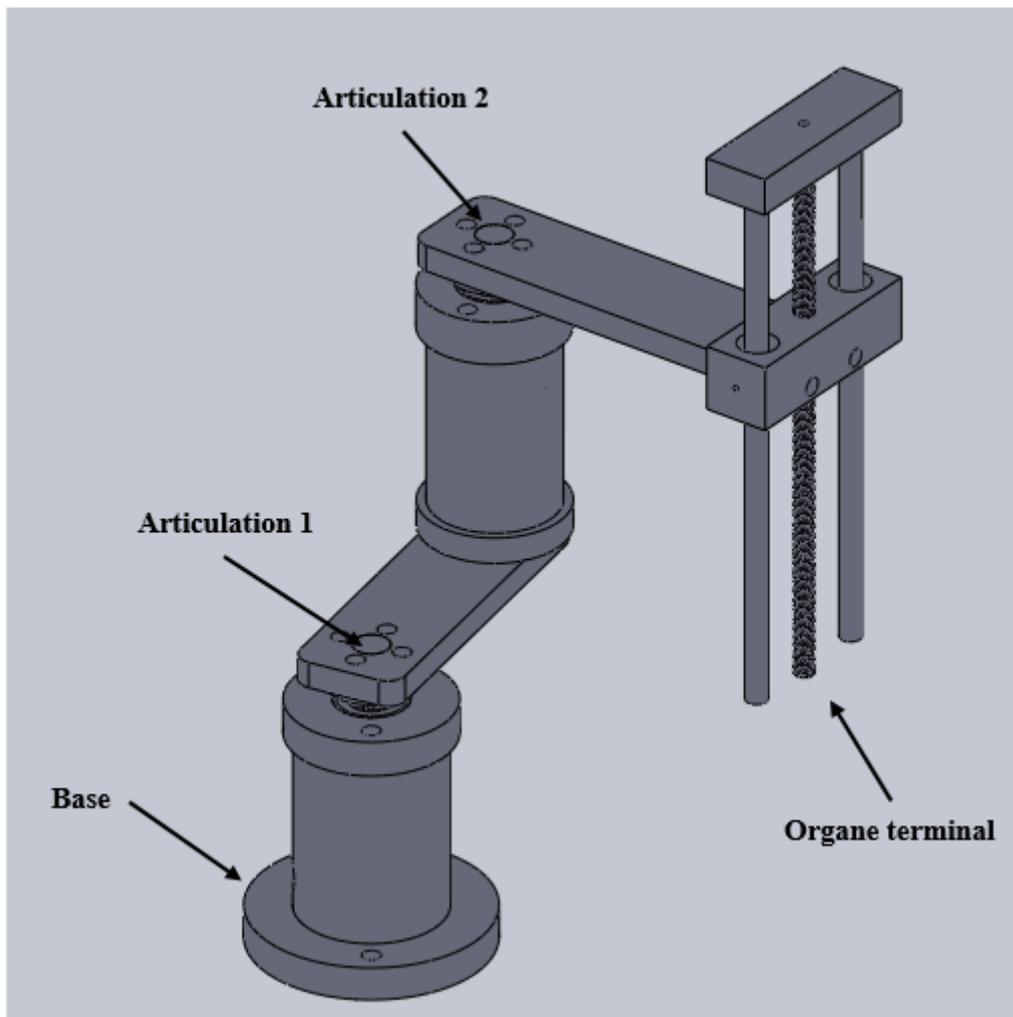
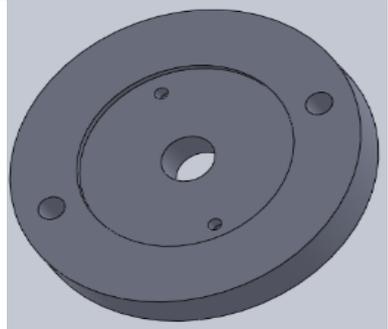
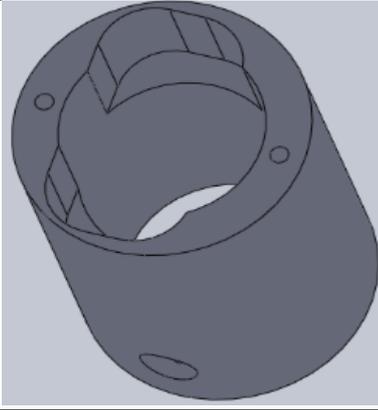
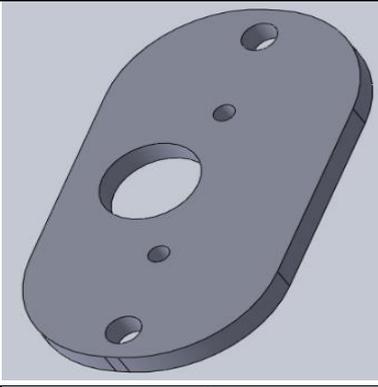
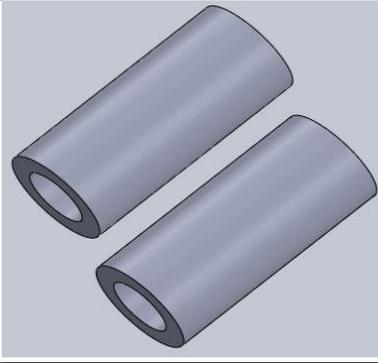
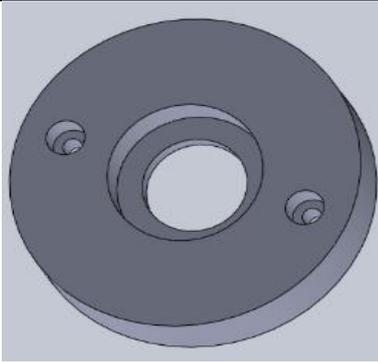
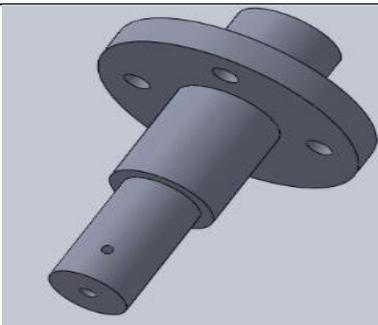
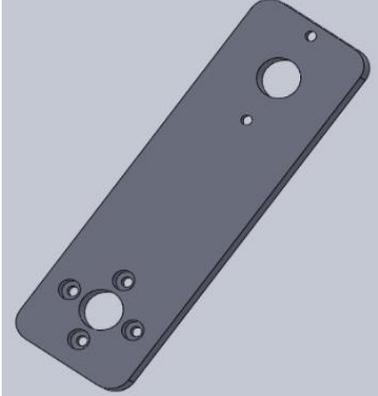
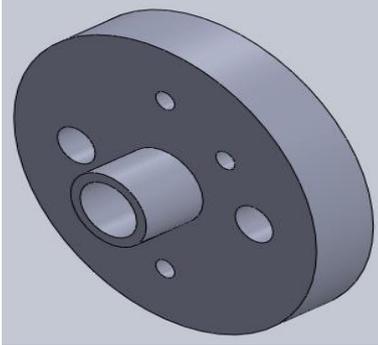
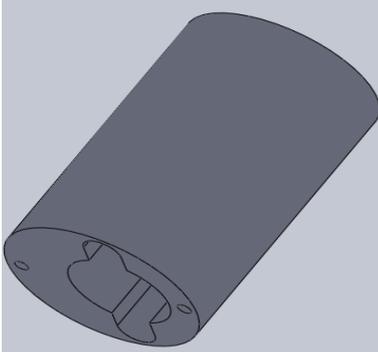
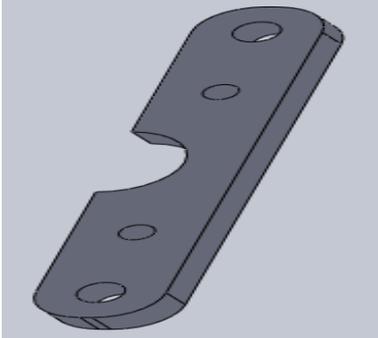
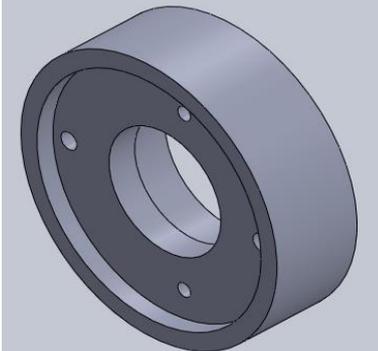


Figure 4.1. Structure du bras manipulateur à 2ddl.

Nous avons conçu les différentes pièces nécessaires à la réalisation de notre bras manipulateur à l'aide du logiciel SolidWorks. Ensuite, nous sommes allés chez un tourneur pour réaliser les différentes pièces conçues. Le tableau 4.1 montre ces pièces et leurs dimensions.

Nom de pièce	Pièces	Dimensions
Base du robot		Diamètre extérieur : 130 mm Diamètre intérieur : 80.50 mm Epaisseur : 19 mm Diamètre de petit cercle : 20 mm Diamètre de vis : 3 mm et 5 mm

<p>Cylindre de la base (colonne de la première articulation)</p>		<p>Diamètre extérieur : 80 mm Diamètre intérieur : 55.50 mm Hauteur : 108 mm Diamètre du cercle latéral : 15 mm Diamètre de vis : 5 mm</p>
<p>Support pour fixer le premier moteur</p>		<p>Longueur : 64 mm Largeur : 36 mm Epaisseur : 4 mm Diamètre de petit cercle : 14 mm Diamètre de vis : 3 mm et 5 mm</p>
<p>Deux cylindres pour fixer le support de moteur avec la couverture</p>		<p>Hauteur : 22 mm Diamètre extérieur : 10 mm Diamètre intérieur : 6 mm</p>
<p>Couverture de premier moteur</p>		<p>Diamètre extérieur : 90 mm Diamètre intérieur : 40 mm Epaisseur : 20 mm Diamètre de petit cercle : 27 mm Diamètre de vis : 5 mm</p>
<p>Support pour relier la base à la première articulation et cette dernière à la deuxième articulation</p>		<p>Diamètre extérieur : 50 mm Diamètre intérieur : 20 mm Diamètre de tige : 3 mm Epaisseur du cercle : 6.50 mm Hauteur : 71 mm Diamètre de vis : 5 mm 2 mm</p>

<p>Premier lien (lien interne)</p>		<p>Longueur : 200 mm Largeur : 60 mm Epaisseur : 15 mm Diamètre du cercle : 20 mm Diamètre de vis : 5 mm</p>
<p>La base de deuxième articulation</p>		<p>Diamètre extérieur : 80 mm Diamètre intérieur : 20 mm Diamètre du cercle central : 15 mm Epaisseur : 15 mm Epaisseur du cercle : 15 mm Diamètre de vis : 5 mm</p>
<p>Cylindre entre la première articulation et la deuxième articulation (colonne de la deuxième articulation)</p>		<p>Diamètre extérieur : 70 mm Diamètre intérieur : 37 mm Hauteur : 114 mm Diamètre de vis : 5 mm</p>
<p>Support pour fixer le deuxième moteur</p>		<p>Longueur : 62 mm Largeur : 11 mm Epaisseur : 4 mm Rayon de demi-cercle : 6 mm Diamètre de vis : 3 mm et 5 mm</p>
<p>Couverture de deuxième moteur</p>		<p>Diamètre extérieur : 80 mm Diamètre intérieur : 40 mm Epaisseur : 25 mm Diamètre de petit cercle : 35 mm Diamètre de vis : 5 mm</p>

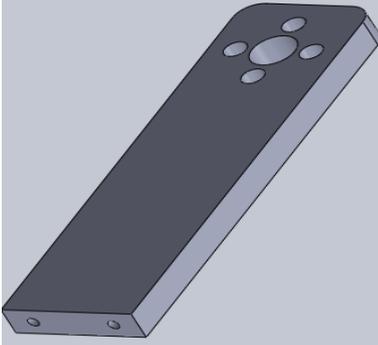
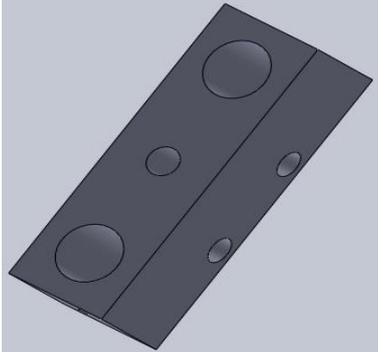
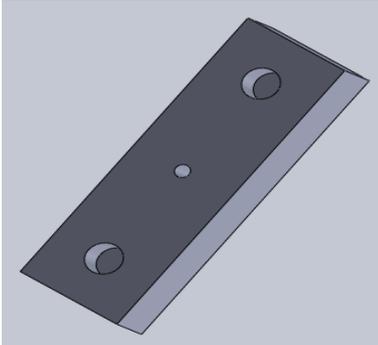
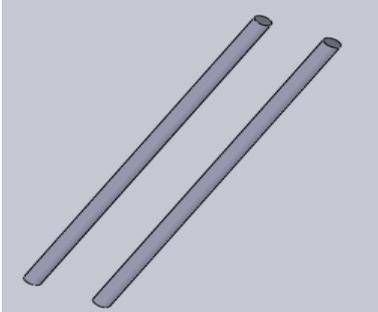
Deuxième lien (lien externe)		Longueur : 209 mm Largeur : 60 mm Epaisseur : 15 mm Diamètre du cercle : 20 mm Diamètre de vis : 5 mm
Support pour relier le lien externe à l'organe terminal		Longueur : 116 mm Largeur : 37 mm Hauteur : 37 mm Diamètre du cercle : 22 mm Diamètre du cercle central : 11 mm Diamètre de vis : 5 mm
Support pour fixer les tiges de l'organe terminal		Longueur : 116 mm Largeur : 37 mm Hauteur : 21 mm Diamètre du cercle : 12 mm Diamètre de vis : 5 mm
Tige filetée de l'organe terminal		Longueur de tige : 311 mm Diamètre de tige : 11 mm
Deux tiges pour supporter l'organe terminal		Longueur : 311 mm Diamètre : 12 mm

Tableau 4.1. Pièces du bras manipulateur avec leurs dimensions.

4.3 Constituant du banc d'expérimentation

Le banc d'expérimentation que nous avons réalisé autour de la carte de développement *eZdsp TMS320F28335* est constitué des éléments illustrés par la figure 4.2.

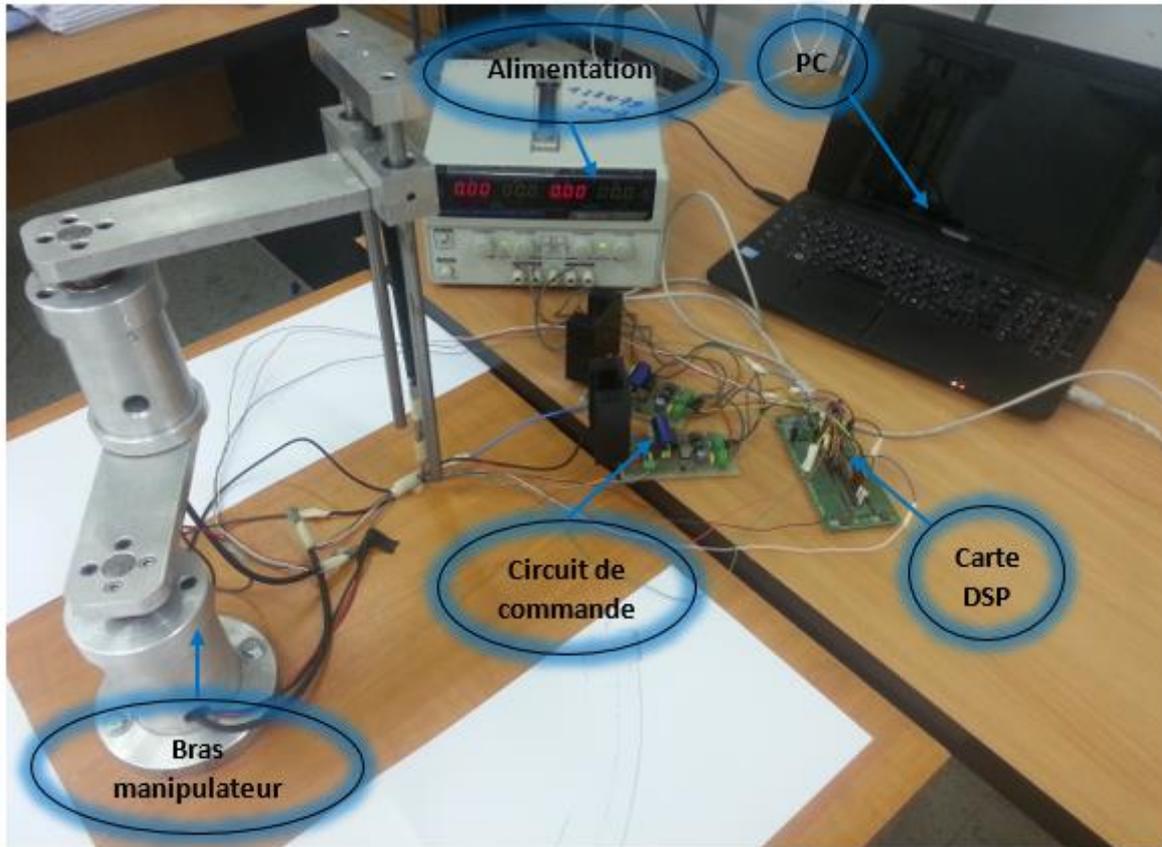


Figure 4.2. Eléments du banc d'expérimentation.

Notre banc d'expérimentation se compose des éléments suivants :

- Alimentation (12 v).
- Bras manipulateur à deux degrés de liberté (SCARA).
- Kit de développement *eZdsp TMS320F28335*.
- Micro-ordinateur.
- Deux circuits de commandes (carte mère).

4.4 Description des éléments du banc d'expérimentation

4.4.1 Le kit de développement *eZdsp F28335*

Le kit de développement *eZdsp F28335* (figure 4.3) est une carte qui nous permet d'implémenter nos programmes afin d'exécuter la tâche spécifiée ainsi de la vérifier plus rapidement par un processeur DSP TMS320F28335. Elle comprend toutes les extensions qui

permettent le dialogue entre le processeur et son entourage. Toutes les opérations se font à travers un micro-ordinateur en utilisant l'outil de développement code composer studio fourni avec le kit.

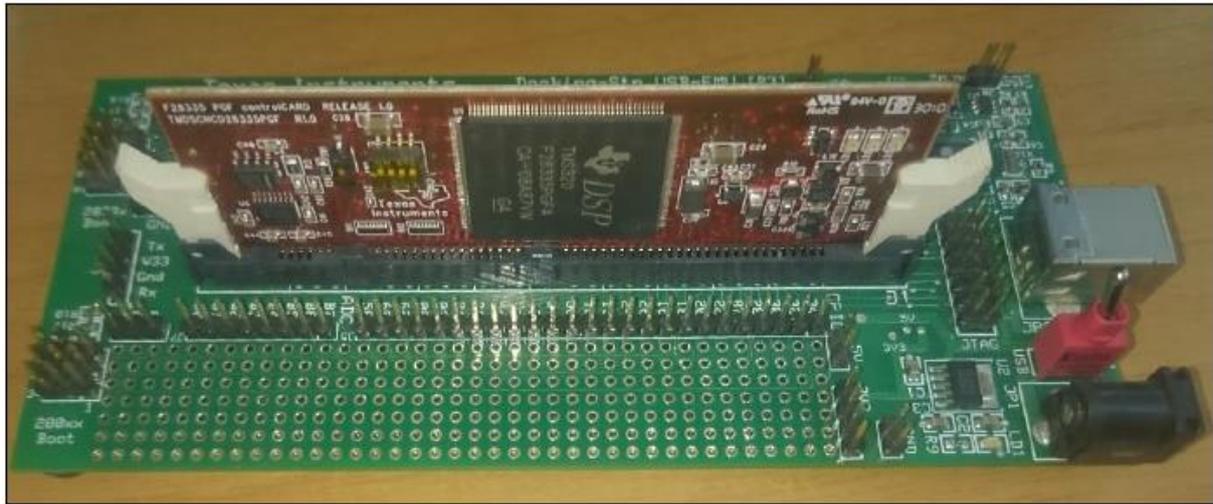


Figure 4.3. La carte eZdsp F28335.

a. Architecture du processeur *TMS320F28335*

Le *TMS320F28335* est un processeur de traitement des signaux qui fonctionne en virgule fixe et en virgule flottante [12]. Il se caractérise par ses performances élevées : un processeur 32 bits hautes performances, une horloge générée par un oscillateur extérieur de fréquence de « 30MHz » et de fréquence interne qui peut atteindre « 150MHz », L'espace mémoire du *TMS320F28335* est divisée en 1 K 16-bit * 4.DSP OTP de mémoire de données, et 4M mot en mémoire de programmation. Il est a noter aussi que ce processeur se distingue par son coût réduit. Les bus système qui lient les blocs fonctionnels du système se basent sur l'architecture « HARVARD » qui offre une meilleure flexibilité de fonctionnement [12]. Le processeur est compilé par un code appelé : « Code Composer Studio ». Il permet la saisie, la compilation, le chargement dans le DSP du programme. La figure 4.4 illustre l'architecture du *TMS320F28335*.

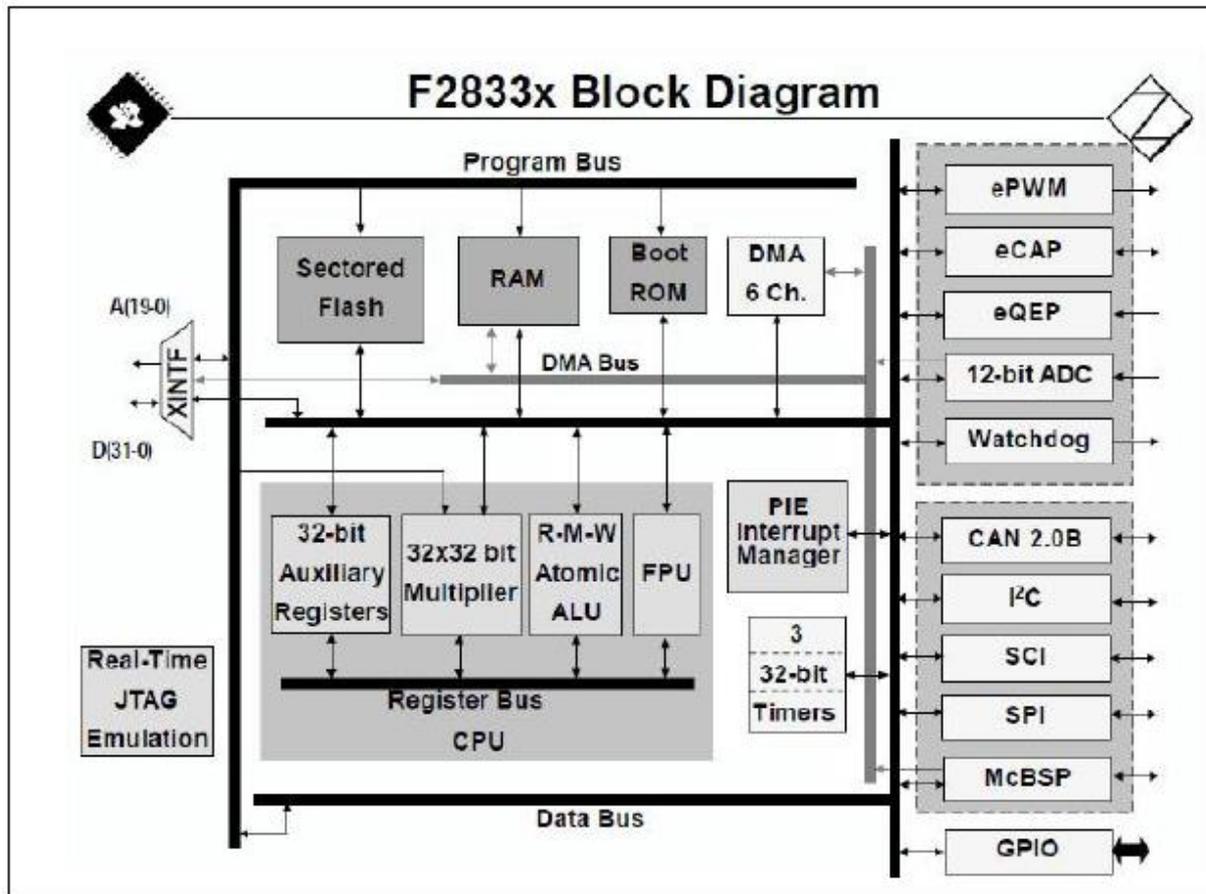


Figure 4.4. Architecture du *TMS320F28335* [12].

b. Modules principaux

- **GPIO (General Purpose Input Output)** : Ce sont les 88 pins multiplexés programmables individuellement avec filtrage d'entrée/sorties disponibles sur la carte. Etant généralement alimenté en 3.3Vcc, le connecteur GPIO ne peut émettre que des courants de faible intensité, allant de 3 mA à 50 mA.
- **Gestionnaire d'interruptions "PIE"** : Le bloc PIE sert à multiplexer de nombreuses sources d'interruption en un ensemble plus petit d'entrées d'interruption. Il peut prendre en charge jusqu'à 96 interruptions périphériques. Sur les 2833x / 2823x, 58 des 96 interruptions possibles sont utilisées par les périphériques. Les 96 interruptions sont regroupées en blocs de 8 et chaque groupe est alimenté dans l'une des 12 lignes d'interruption de CPU (INT1 à INT12). Chacune des 96 interruptions est supportée par son propre vecteur stocké dans un bloc de RAM dédié pouvant être écrasé par l'utilisateur. Le vecteur est automatiquement récupéré par le CPU lors du traitement de l'interruption. Il faut huit cycles d'horloge de CPU pour extraire le vecteur et

sauvegarder les registres critiques du CPU, cette dernière peut répondre rapidement aux événements d'interruption. La priorisation des interruptions est contrôlée dans le matériel et le logiciel. Chaque interruption individuelle peut être activée ou désactivée dans le bloc PIE.

- **Modulation à largeur d'impulsion, (Enhanced Pulse Width Modulator: ePWM) :**
Une des sorties est configurée sur la fonction ePWM. La fréquence des impulsions est réglée grâce au registre TBPRD du registre de contrôle EPwm2Regs.
- **Périphériques (E/S) :** Le TMS320F28335 contient 88 broches d'E/S multipoints réparties sur 3 multiplexeurs. Chaque broche peut être configurée pour 4 fonctions différentes à l'aide de registres de contrôle appelés GPAMUX, GPBMUX et GPCMUX.
- **Dispositif de sécurité (Watchdog) :** Un bug du programme peut avoir de lourdes conséquences sur un circuit externe, c'est pour cela on utilise le Watchdog pour but de surveiller tout plantage qui peut survenir tout le long d'un programme pour une raison quelconque. Il exige chaque 4.37 ms une exécution d'une certaine commande spécifique sauf si on est dans le cas contraire, à ce moment il initie un Reset du processeur remettant tous les sorties à zéro.
- **Module Serial Communications Interface (SCI) :** Il y a trois modules d'interface de communication série (SCI). Les modules SCI prennent en charge les communications numériques entre la CPU et les autres périphériques asynchrones utilisant le format NRZ (anti-retour à zéro) standard. Le récepteur et l'émetteur SCI sont dotés d'une mémoire tampon double et chacune a ses propres bits d'activation et d'interruption distincts. Les deux peuvent fonctionner indépendamment ou simultanément en mode fullduplex. Pour garantir l'intégrité des données, le SCI vérifie les données reçues pour détecter les erreurs de détection, de parité, de dépassement de temps et de cadrage. Le débit est programmable à plus de 65 000 vitesses différentes via un registre de sélection de bauds de 16 bits. Chaque module SCI comprend deux broches externes :
 - SCITXD : broche d'émission-sortie SCI.
 - SCIRXD : broche d'entrée de réception SCI.

4.4.2 Circuit de commande (carte mère)

Pour commander le robot nous avons réalisé un circuit de commande constitué de trois parties principales comme indiqué dans la figure 4.5.

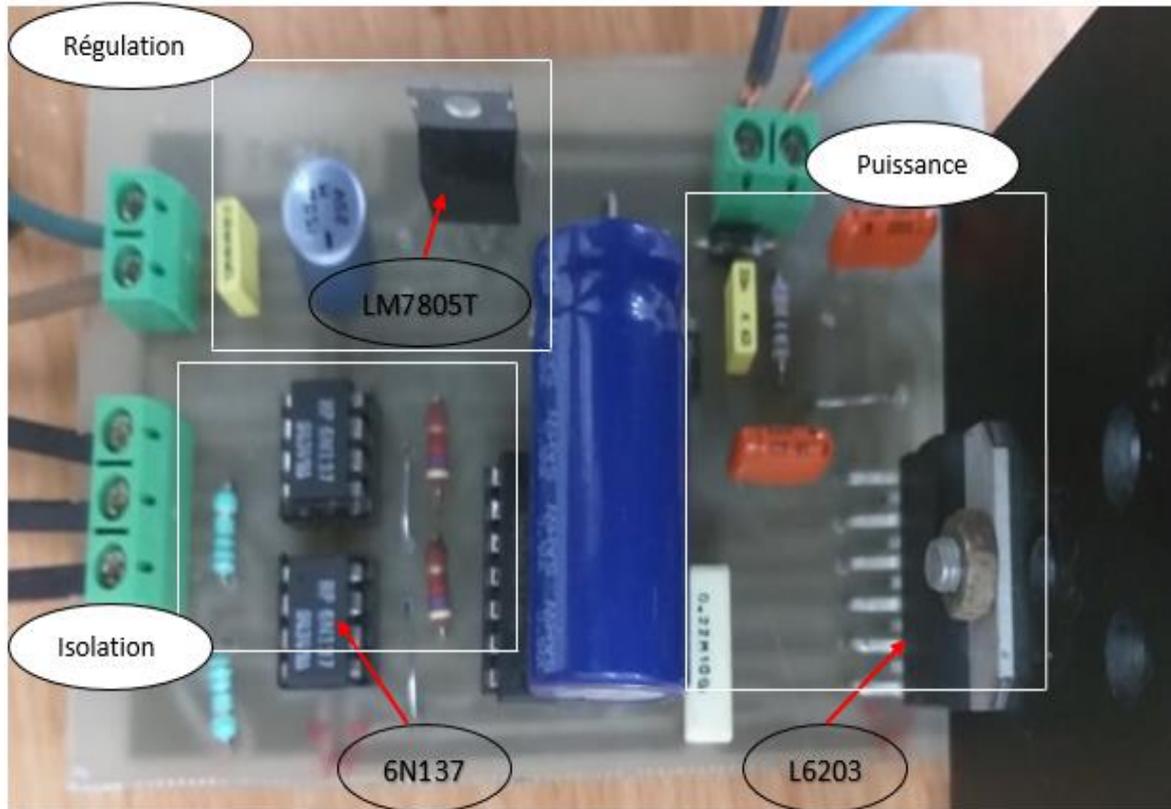


Figure 4.5. Circuit de commande (carte mère).

Nous avons utilisé le logiciel Eagle (Easily Applicable Graphical Layout Editor) pour réaliser notre circuit imprimé (carte mère). La figure 4.6 montre le schéma électrique global de la carte de commande avec les trois parties principales.

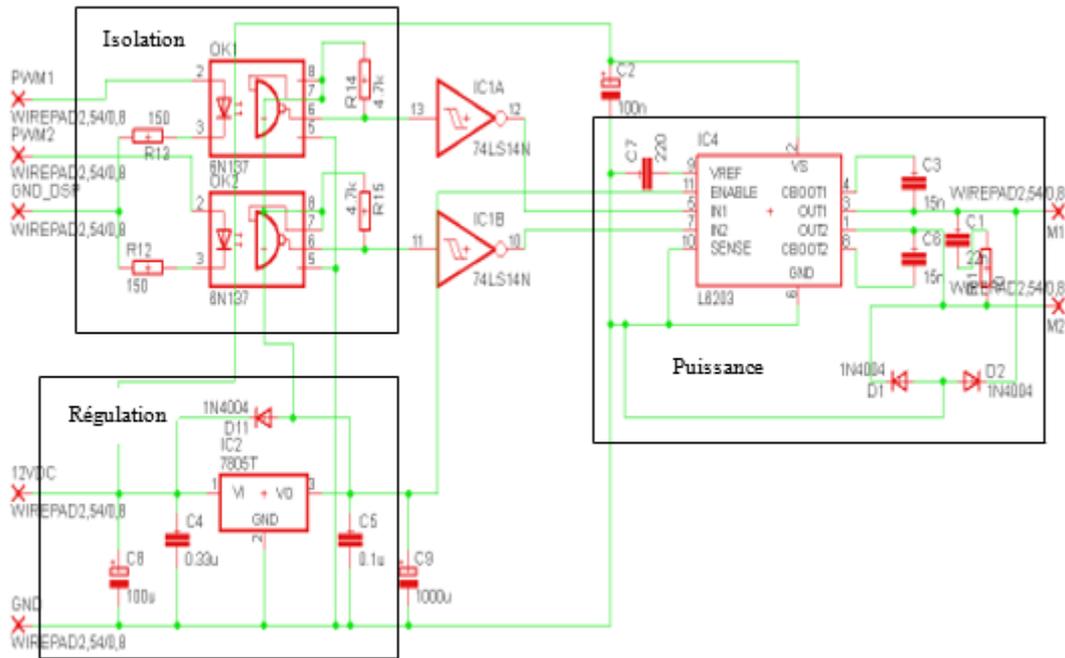


Figure 4.6. Schéma électrique du circuit de commande.

a. Partie d'isolation (isolation galvanique)

Pour protéger la partie commande, nous avons utilisé un optocoupleur de la série (6N137). Un optocoupleur est un composant électronique couplé optiquement à une porte logique intégrée à très haute vitesse capable de transmettre un signal d'un circuit électrique à un autre. La figure 4.7 illustre le circuit d'isolation galvanique.

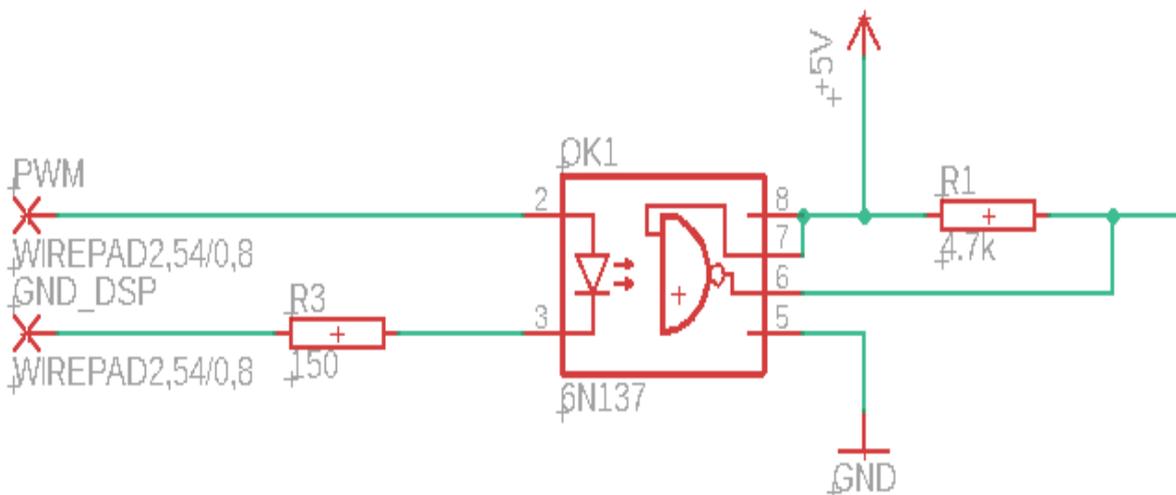


Figure 4.7. Schéma électrique du circuit d'isolation avec un optocoupleur (6N137).

b. Partie de régulation de tension

Un régulateur de tension, est un composant électronique qui maintien à sa sortie une tension constante indépendamment de la tension d'entrée, il utilise une limitation interne du courant, une coupure thermique et compensation de la zone de sécurité, ce qui le rend essentiellement indestructible. La figure 4.8 montre le régulateur de tension (LM7805) que nous avons utilisé dans notre circuit pour régler la tension d'entrée.

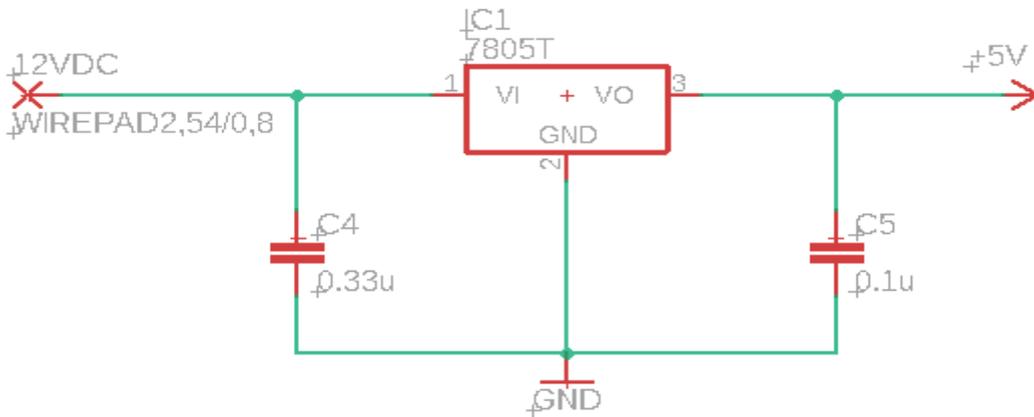


Figure 4.8. Schéma électrique du circuit de régulation de tension.

c. Partie de puissance

Le circuit de commande est constitué d'un hacheur à 4 quadrants (figure 4.9) qui combine des transistors de puissance. Les 4 transistors sont symbolisés par des interrupteurs T1, T2, T3, T4 (figure 4.10). Lorsque T1 et T4 sont fermés (saturés), le moteur tourne dans un sens (sens 1). Lorsque T2 et T3 sont fermés, le moteur va tourner dans l'autre sens (sens 2).

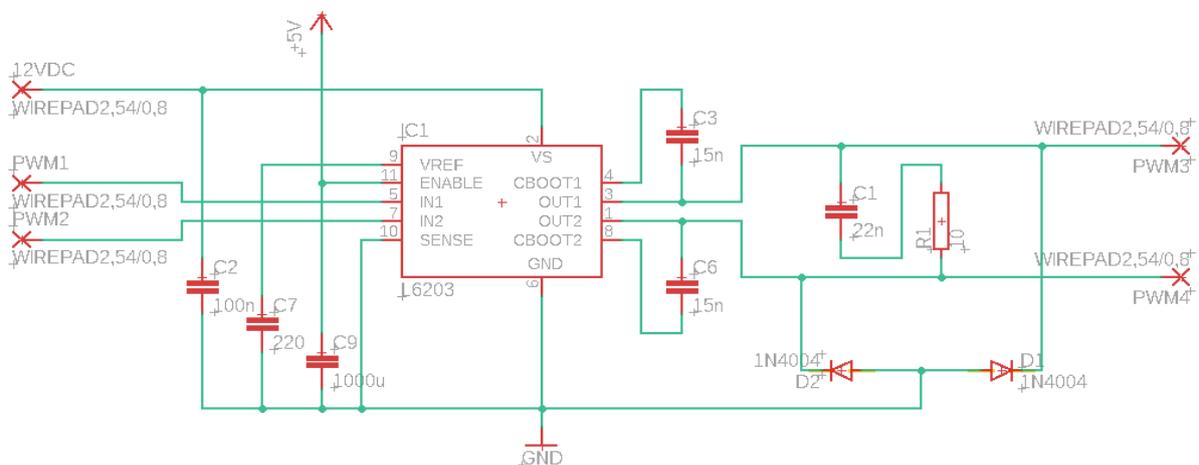


Figure 4.9. Schéma électrique du circuit de puissance.

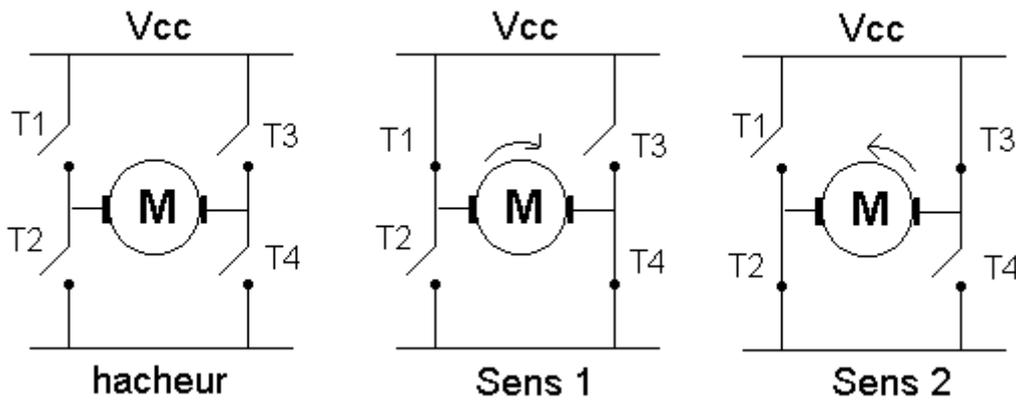


Figure 4.10. Principe de fonctionnement d'un hacheur à 4 quadrants.

Pour contrôler le sens de rotation du moteur, il suffit de varier le rapport cyclique selon le besoin, un rapport cyclique inférieur à 0.5 va résulter une tension moyenne positive, et un rapport cyclique supérieur à 0.5 va donner une tension moyenne négative. On peut toujours observer qu'un rapport cyclique de 0.5 va générer une tension moyenne nulle, donc pour arrêter le moteur il suffit de donner un rapport cyclique de 0.5 au signal de commande (PWM).

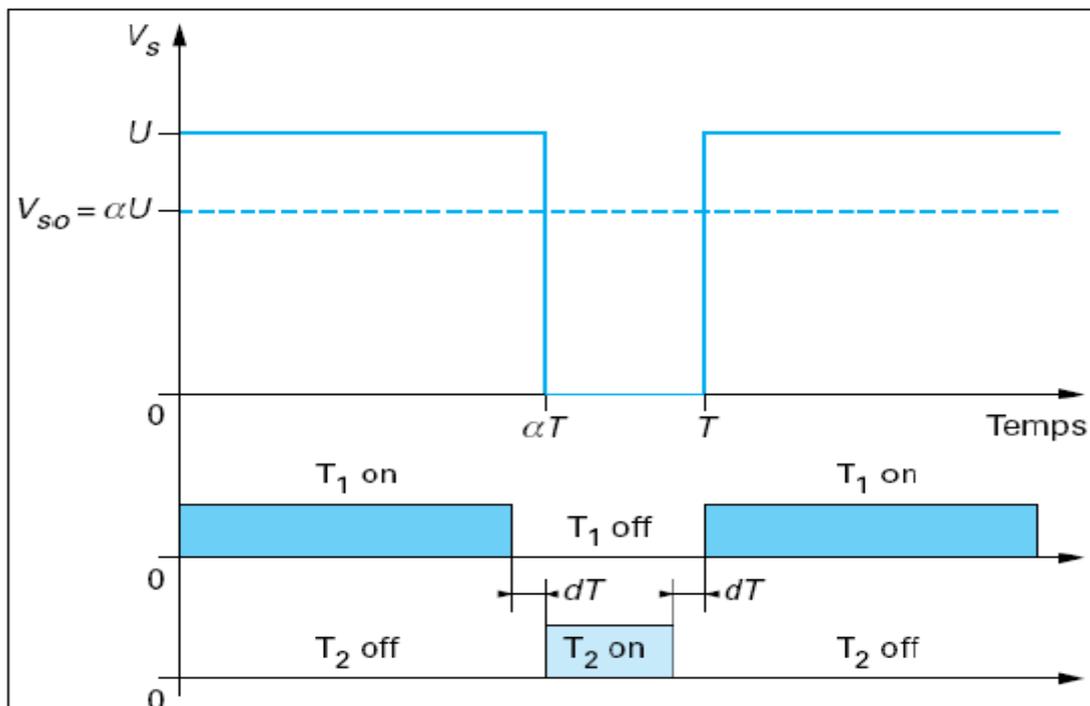


Figure 4.11. Commande en tension de l'hacheur.

4.4.3 Moteur à courant continu (MCC)

Pour les articulations du bras nous avons utilisé des moteurs à courant continu (MCC) (figure 4.12).

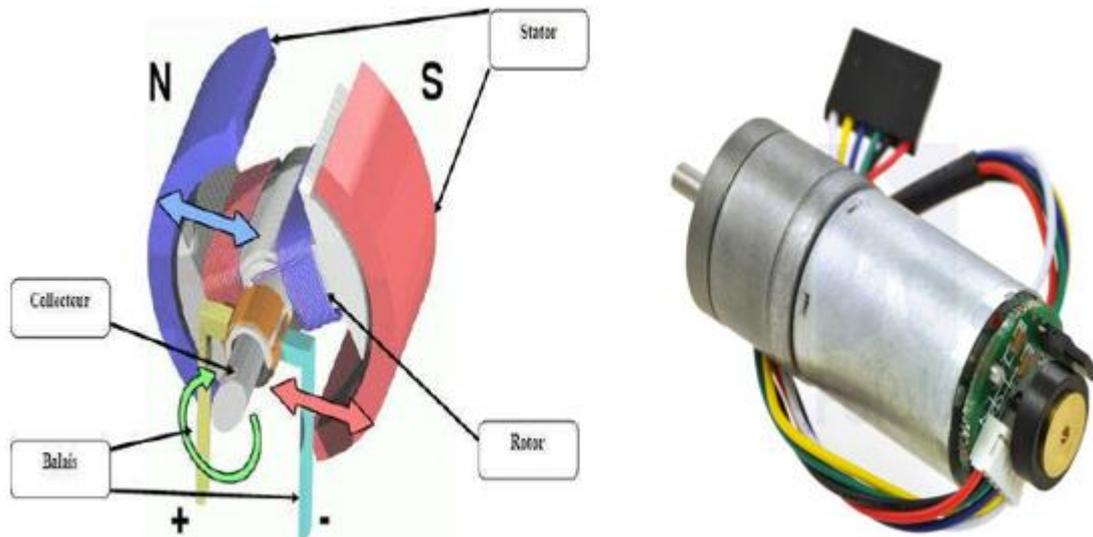


Figure 4.12. Moteur à courant continu.

Le moteur à courant continu a deux sources de flux distinctes qui interagissent pour développer un couple. Ce sont les circuits de champ et d'armature. En raison de l'action du commutateur, le couple développé est donné par :

$$T = K i_a \quad (4.1)$$

Où i_f et i_a sont respectivement les courants de champ et d'induit, et K est une constante reliant les dimensions du moteur et les paramètres des circuits magnétiques.

Les réponses dynamique et l'état stable du moteur et de la charge sont données par :

$$v_a = R_a i_a + L_a \frac{di_a}{dt} + e \quad v_a = R_a I_a + E \quad (4.2)$$

$$v_f = R_f i_f + L_f \frac{di_f}{dt} \quad V_f = R_f I_f \quad (4.3)$$

$$e = K i_f \omega \quad E = K I_f \omega \quad (4.4)$$

$$T = J \frac{d\omega}{dt} + D\omega + T_L \quad T = J\dot{\omega} + T_L \quad (4.5)$$

Où J , D et T_L sont le moment d'inertie, le facteur d'amortissement et le couple de charge, respectivement, par rapport au moteur, et les indices a et f se réfèrent aux circuits d'induit et de champ, respectivement. R , L , I et E se réfèrent à la résistance, à l'inductance, au courant et à la force contre-électromotrice du moteur dans les circuits associés désignés par les indices [14].

Les variables et paramètres suivants sont définis :

θ_m : position angulaire du moteur en (*rad*).

θ_l : position angulaire de charge en radians (*rad*).

ω : vitesse angulaire en (*rad/s*).

T_m : couple moteur développé en (N_m).

J_m : moment d'inertie du moteur en ($K_g m^2$).

J_l : moment d'inertie de la charge ($K_g m^2$).

E_m : tension appliquée en (*v*).

K_m : couple moteur constant en (N_m/A).

K_b : constante de force contre-électromotrice du moteur en (*v/s/rad*).

B_m : coefficient de frottement visqueux du moteur ($N_m / rad/s$).

B_l : coefficient de frottement visqueux de la charge ($N_m / rad/s$).

N : rapport charge-moteur à engrenages = θ_l / θ_m .

B_{me} : coefficient de frottement visqueux réfléchi par l'arbre du moteur = $B_m + N_2 B_l$.

J_{me} : inertie reflétée sur l'arbre du moteur = $K_m + N^2 J_l$.

R_a : résistance d'induit du moteur en (Ω).

L_a : inductance d'induit du moteur en (H).

La fonction de transfert entre le déplacement du moteur et la tension d'entrée est [15] :

$$\frac{\theta_m(s)}{E_m(s)} = \frac{K_m}{s(L_a J_{me} s^2 + (R_a J_{me} + B_{me} L_a) s + (K_b K_m + R_a B_{me}))} \quad (4.6)$$

4.5 Identification par les réseaux de neurones

Afin d'éviter l'intégration des équations différentielles définissant le modèle dynamique du bras (temps de calcul important, divergence) et résoudre le problème de couplage, nous avons préféré de représenter le modèle dynamique de chacune des deux articulations par deux modèles neuronaux découplés (figure 4.13). Chaque modèle, a un réseau de neurones de type MLP (Multi Layer Perceptron) (figure 4.14), est entraîné en utilisant l'apprentissage supervisé (en utilisant l'algorithme Levenberg-Marquardt) et un ensemble de données que nous avons généré en appliquant plusieurs commandes, dont les valeurs (3000 valeurs) sont générées aléatoirement, sur le bras et mesurant les positions angulaires correspondantes des articulations (figures 4.15 et 4.16). Les neurones de la couche cachée de chaque réseau utilisent comme fonction d'activation la fonction tangente hyperbolique. Le neurone de sortie de chaque réseau possède une fonction d'activation linéaire.

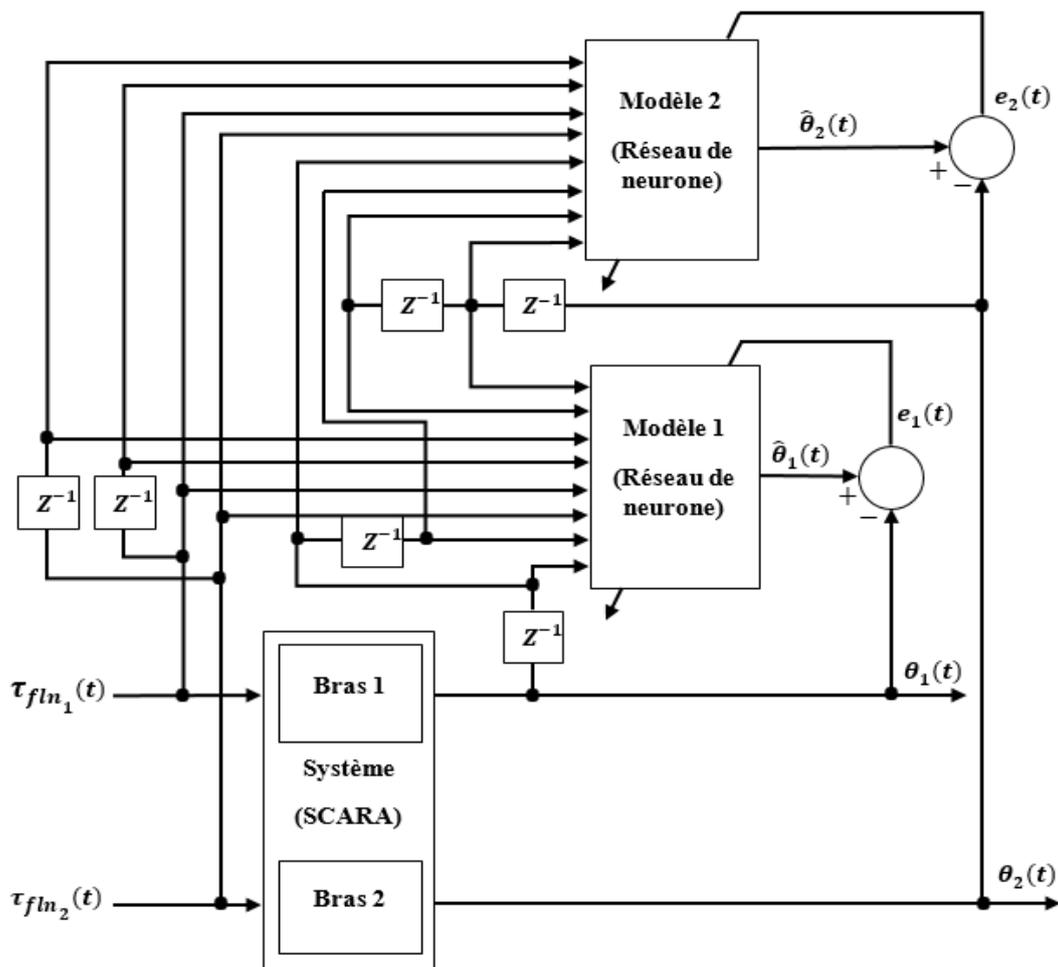


Figure 4.13. Schéma bloc de l'identification.

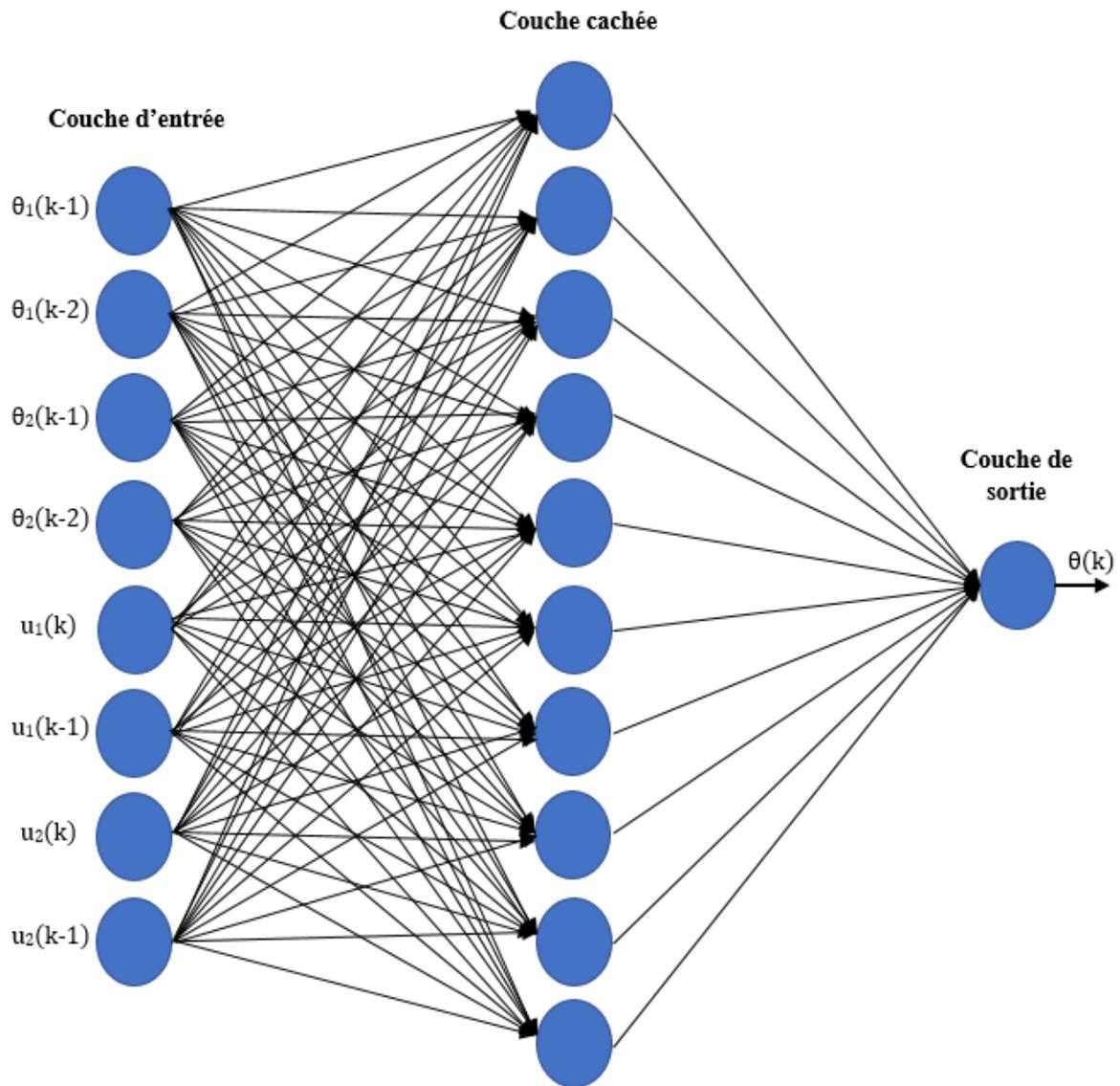


Figure 4.14. Architecture du réseau de neurone pour chaque modèle ($\theta(k) = \theta_1(k)$ ou $\theta_2(k)$).

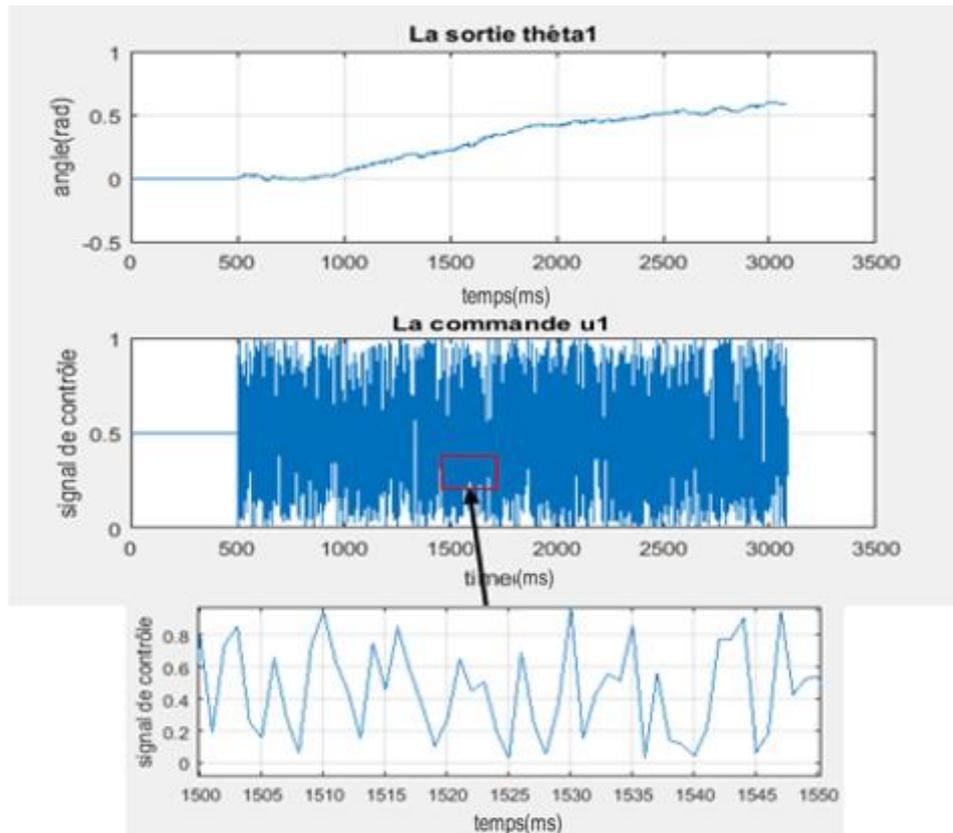


Figure 4.15. Données d'apprentissage et de test pour la première articulation.

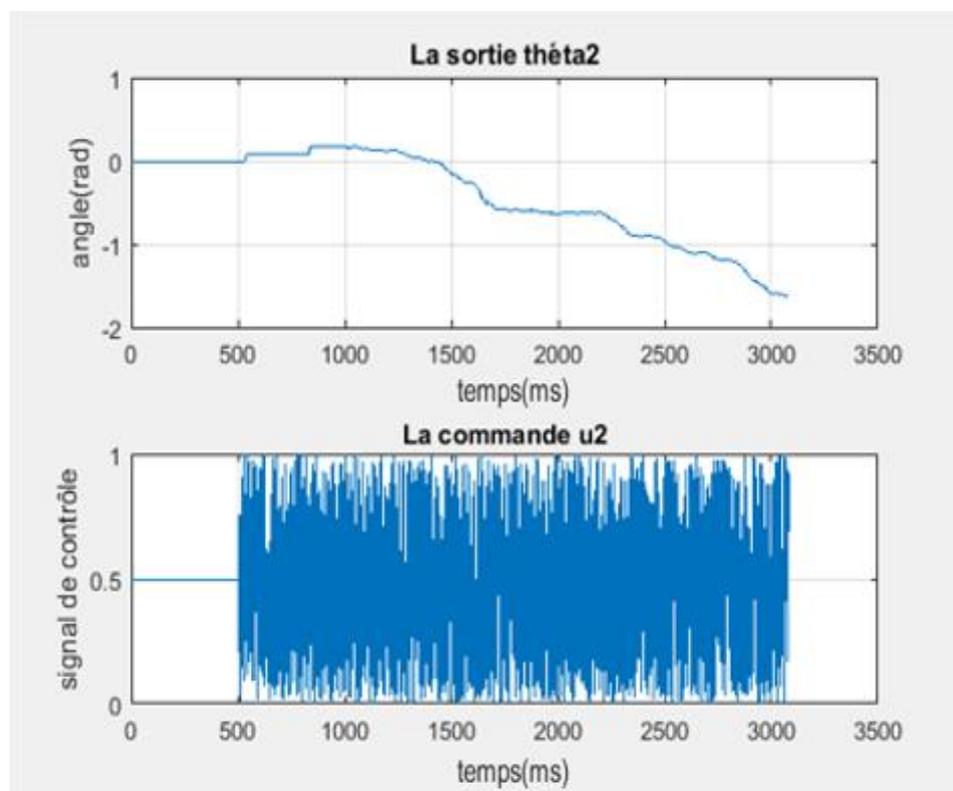


Figure 4.16. Données d'apprentissage et de test pour la deuxième articulation.

La base de données est divisée en deux parties, la première partie est utilisée pour faire l'apprentissage et la deuxième est utilisée pour tester le modèle obtenu. Les résultats de test des deux modèles obtenus (pour θ_1 et θ_2) sont donnés par les figures 4.17 et 4.18 où nous pouvons constater que les sorties des modèles et celles mesurées sont superposées.

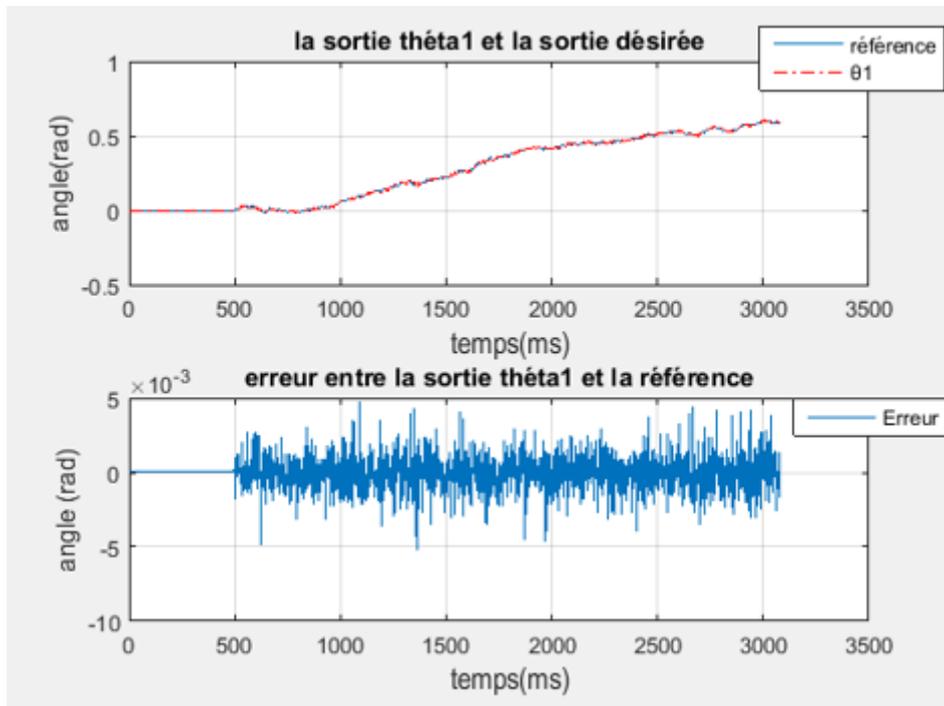


Figure 4.17. Résultat de test pour le modèle de la première articulation.

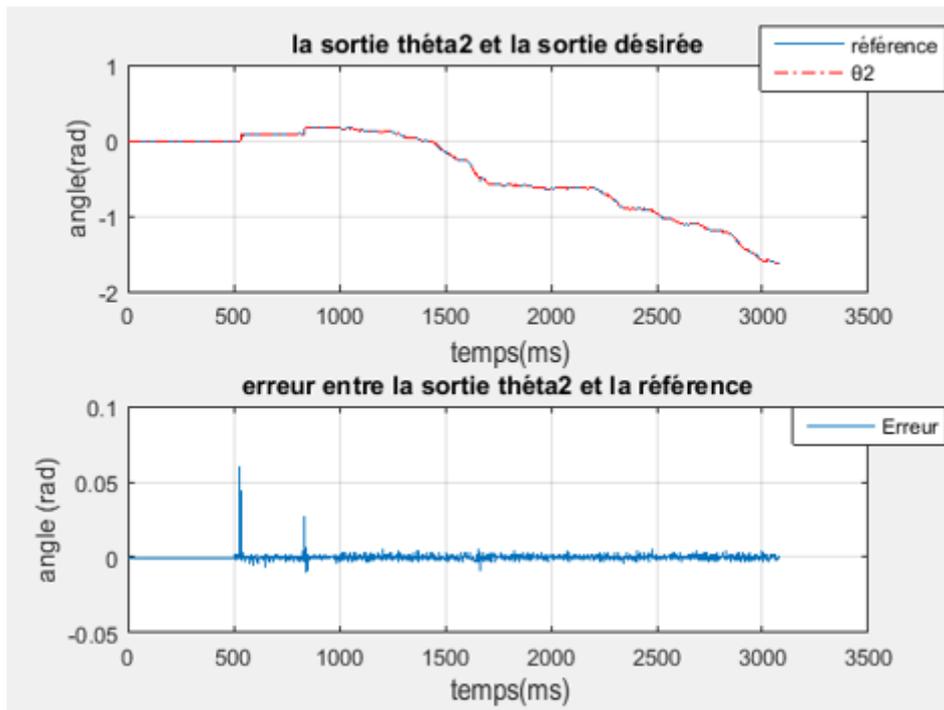


Figure 4.18. Résultat de test pour le modèle de la deuxième articulation.

Remarque :

Nous avons utilisé le protocole UART (Universal Asynchronous Receiver-Transmitter) pour la transmission des données aléatoires (les commandes u_1 et u_2 et les positions angulaires correspondantes θ_1 et θ_2) de la carte DSP à l'ordinateur. A cet effet, nous avons connecté le module SCITXD (SCI Transmit Data Register) qui se trouve dans la carte DSP avec le RXD de l'adaptateur série UART-USB (figure 4.19).

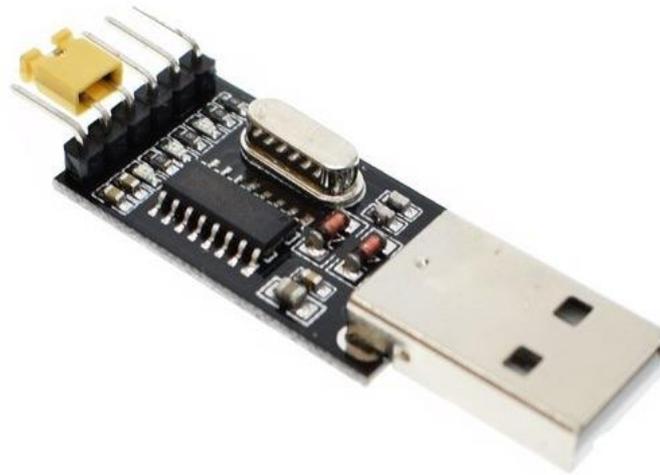


Figure 4.19. Adaptateur USB vers TTL UART.

4.6 Commande pratique du bras en utilisant deux régulateurs PID optimisés à l'aide de l'algorithme TLBO

4.6.1 Structure et mise en œuvre de la commande

Nous avons utilisé l'algorithme d'optimisation TLBO pour déterminer, hors ligne, les paramètres des deux régulateurs PID dans le but de commander les deux articulations du bras manipulateur. Le schéma de commande est donné par la figure 4.20.

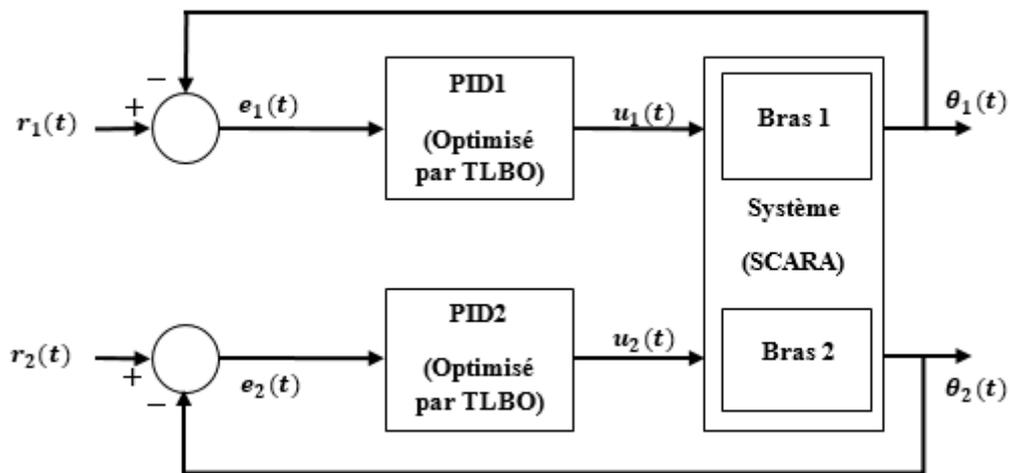


Figure 4.20. Structure de commande du bras manipulateur.

La fonction du coût à minimiser est donnée par :

$$f = \sum_{k=1}^N e_1^2(k) + e_2^2(k)$$

Avec : $e_i(k) = r_i(k) - \theta_i(k)$, $i = 1,2$, $N = 1000$

Les valeurs des paramètres de l'algorithme TLBO utilisées sont :

La taille de population : 20

Nombre d'itérations : 200

Nombre des variables de conception : 3

Nous avons obtenu pour les valeurs des paramètres des deux PID données dans le tableau 4.2.

Paramètres	K_P	T_I	T_D
Première articulation	-0.6046	0.0291	-0.0806
Deuxième articulation	0.2298	0.0937	0.2133

Tableau 4.2. Valeurs des paramètres des deux régulateurs PID.

Une fois les valeurs des paramètres des deux régulateurs PID sont obtenues, l'algorithme global de commande est implémenté dans le DSP.

4.6.2 Résultats pratiques de commande

Nous avons effectué plusieurs essais de commande du bras (commande des positions angulaires) pour dessiner des cercles de différents rayons (0.28 m et 0.33 m). La figure suivante (figure 4.21) montre les demi-cercles obtenus dans l'espace de travail du bras manipulateur.

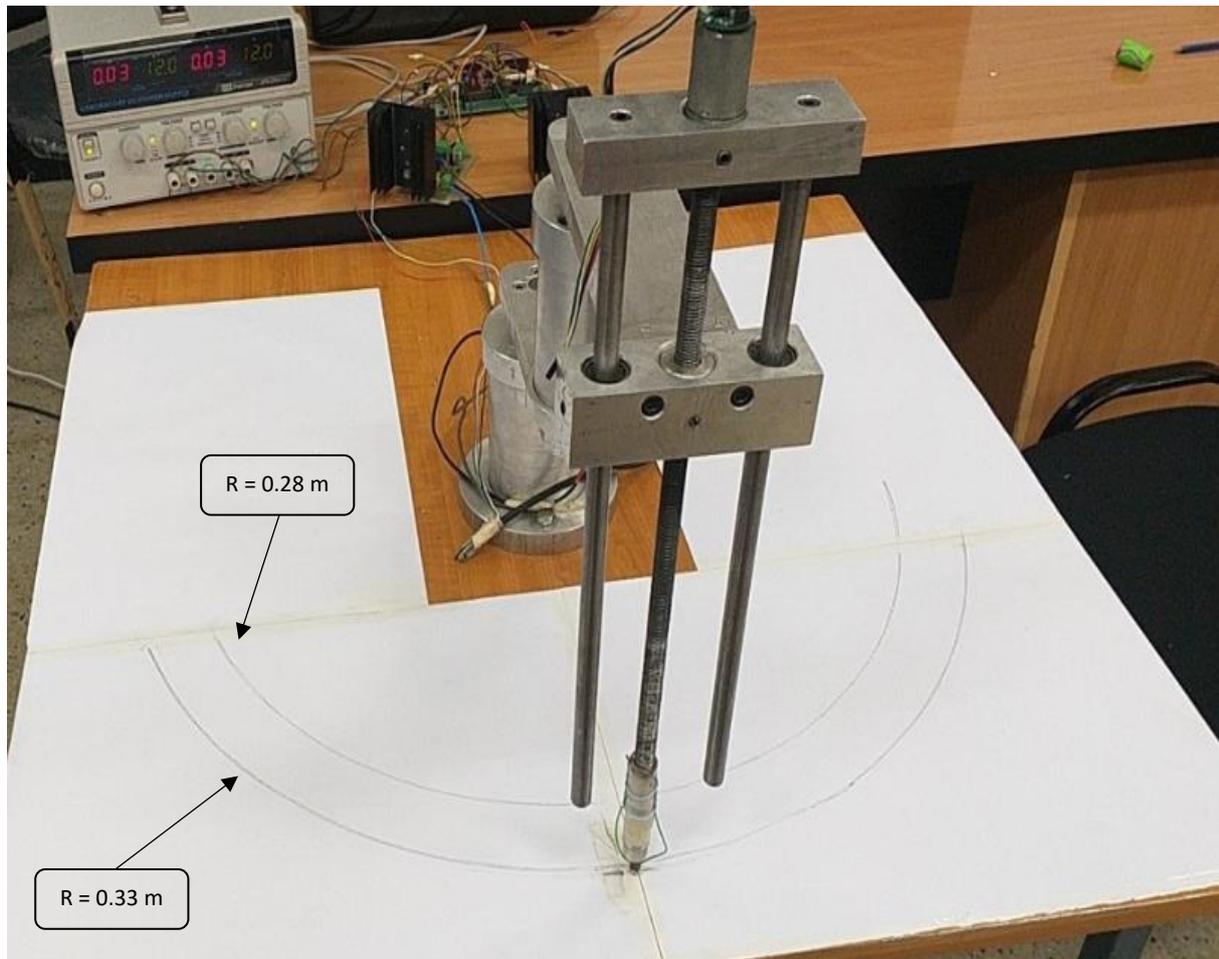


Figure 4.21. Résultat de commande du bras pour le dessin de deux demi-cercles de rayon différents.

Les figures 4.22 et 4.24 représentent les sorties θ_1 et θ_2 et les trajectoires de références correspondantes et l'erreur pour chaque articulation est représentée respectivement dans la figure 4.23 et la figure 4.25 dans le cas du demi-cercle de rayon 0.28 m.

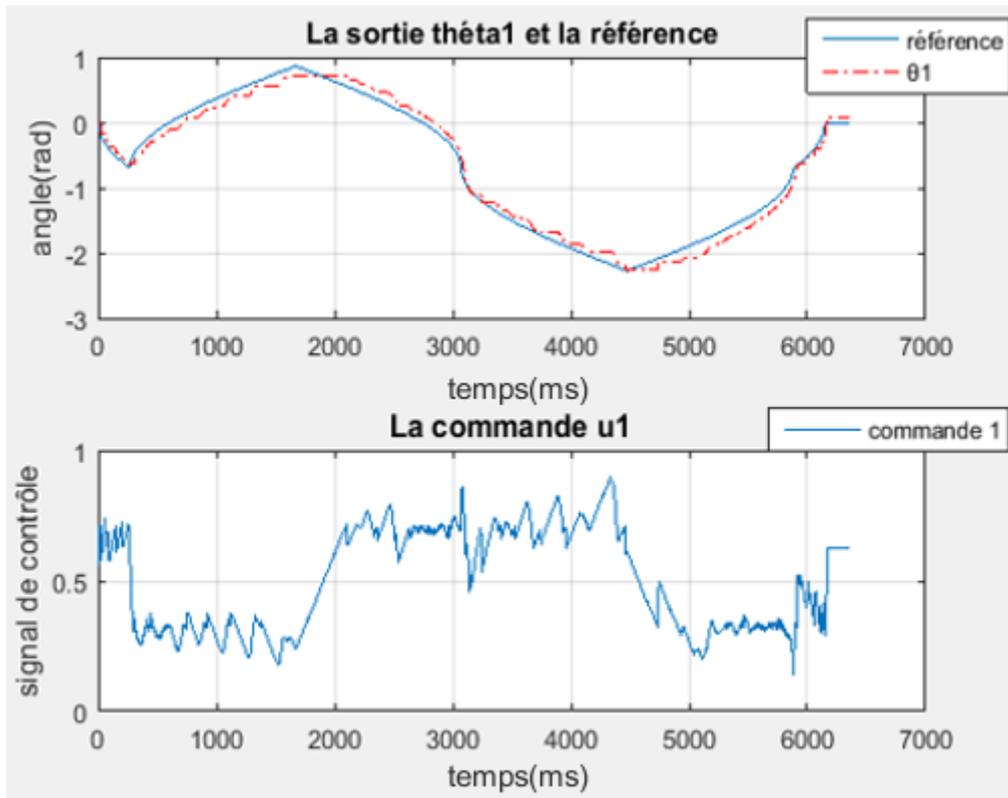


Figure 4.22. Poursuite de la trajectoire de référence pour l'articulation θ_1 dans le cas du demi-cercle $R=0.28m$.

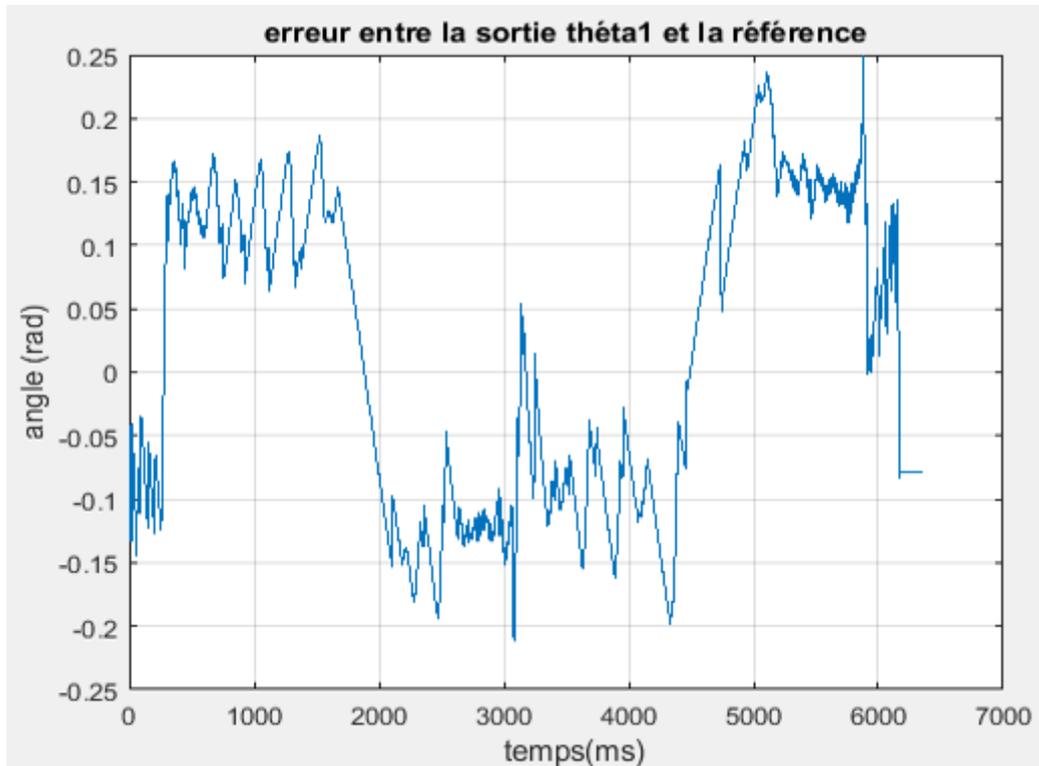


Figure 4.23. Erreur entre la référence et la sortie θ_1 dans le cas du demi-cercle $R=0.28m$.

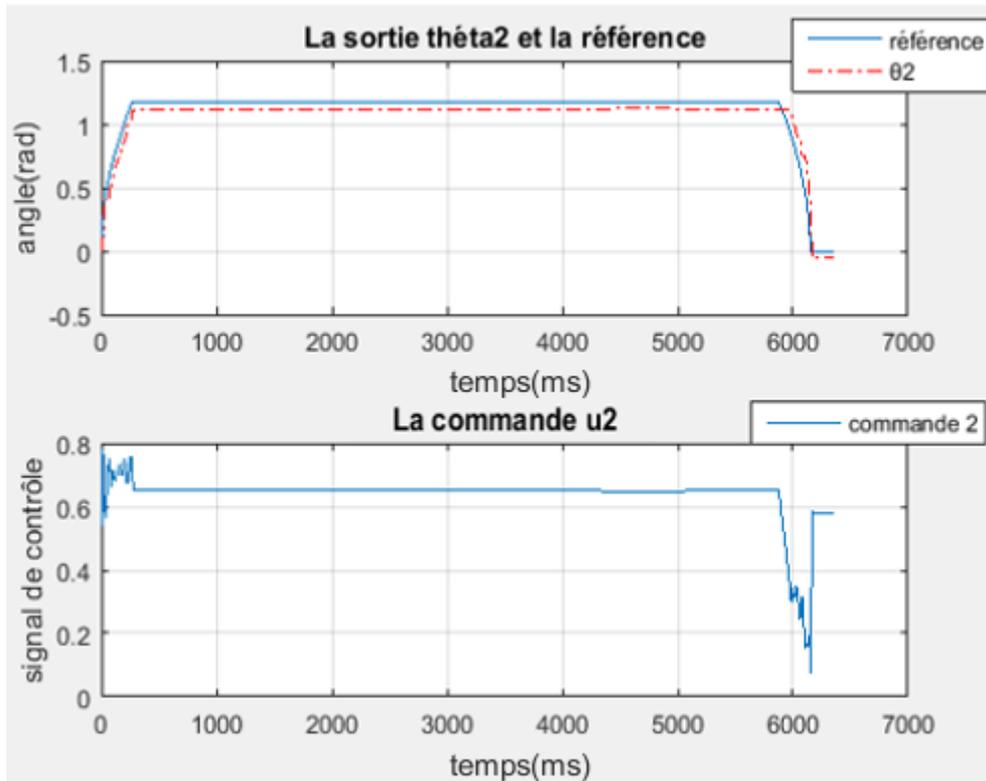


Figure 4.24. Poursuite de la trajectoire de référence pour l'articulation θ_2 dans le cas du demi-cercle $R=0.28m$.

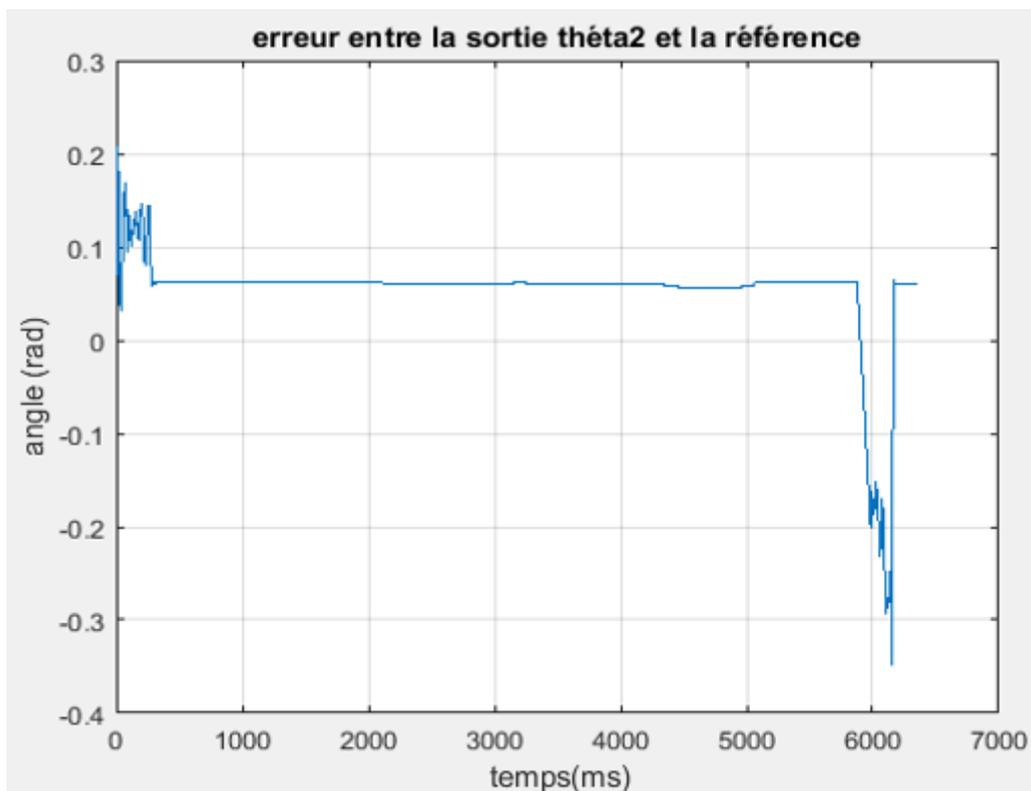


Figure 4.25. Erreur entre la référence et la sortie θ_2 dans le cas du demi-cercle $R=0.28 m$.

Les figures 4.26 et 4.28 représentent les sorties θ_1 et θ_2 et les trajectoires de références correspondantes et l'erreur pour chaque articulation est représentée respectivement dans la figure 4.27 et la figure 4.29 dans le cas du demi-cercle de rayon 0.33 m.

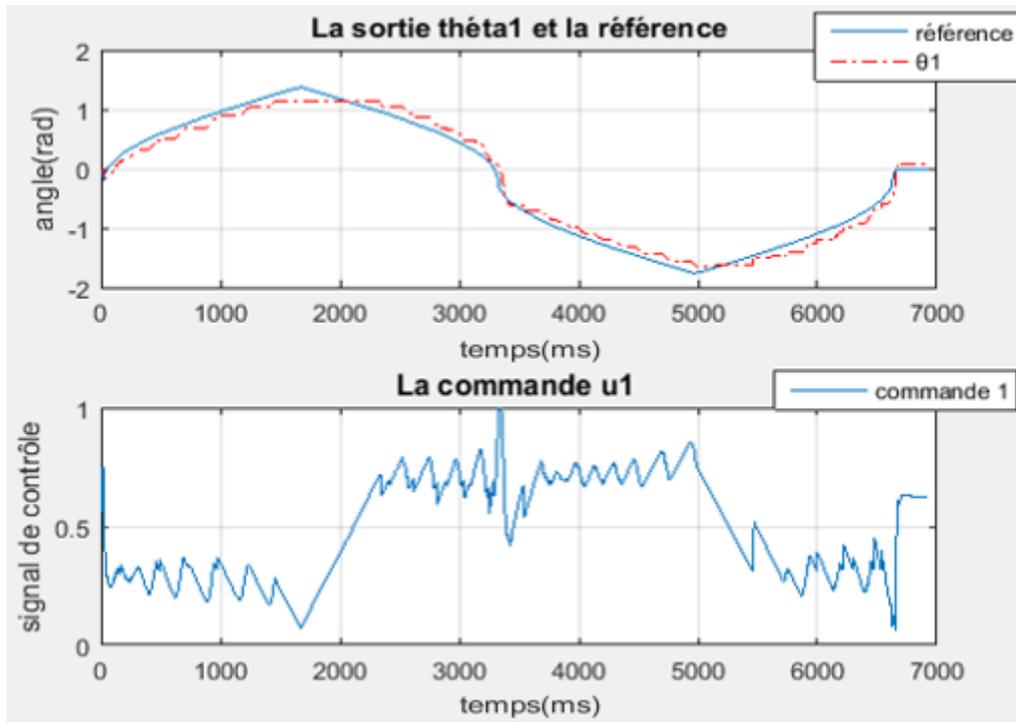


Figure 4.26. Poursuite de la trajectoire de référence pour l'articulation θ_1 dans le cas du demi-cercle $R=0.33$ m.

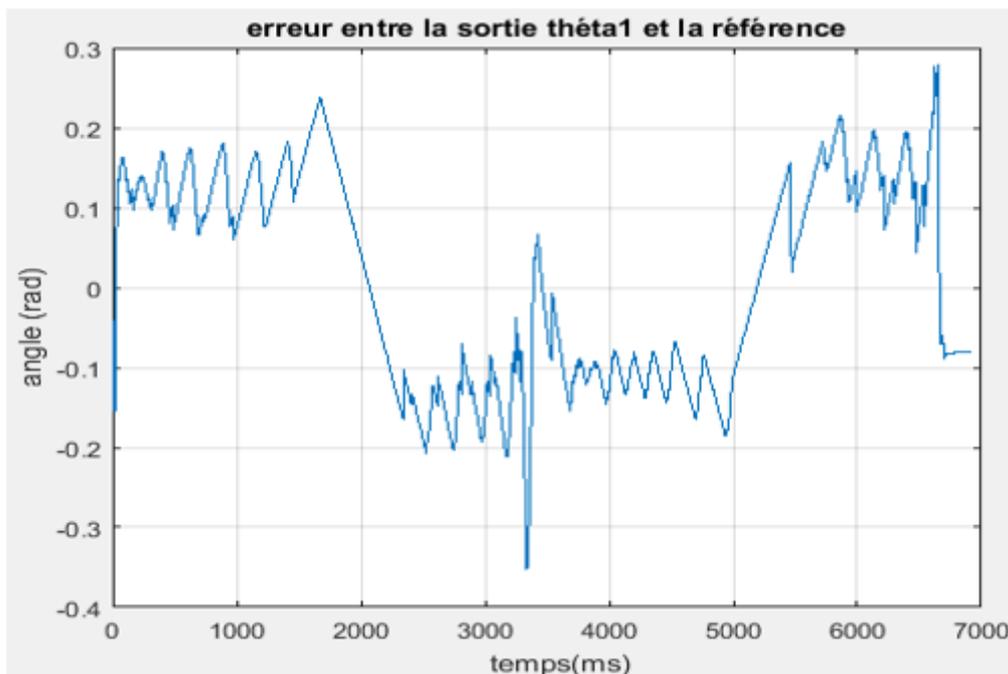


Figure 4.27. Erreur entre la référence et la sortie θ_1 dans le cas du demi-cercle $R=0.33$ m.

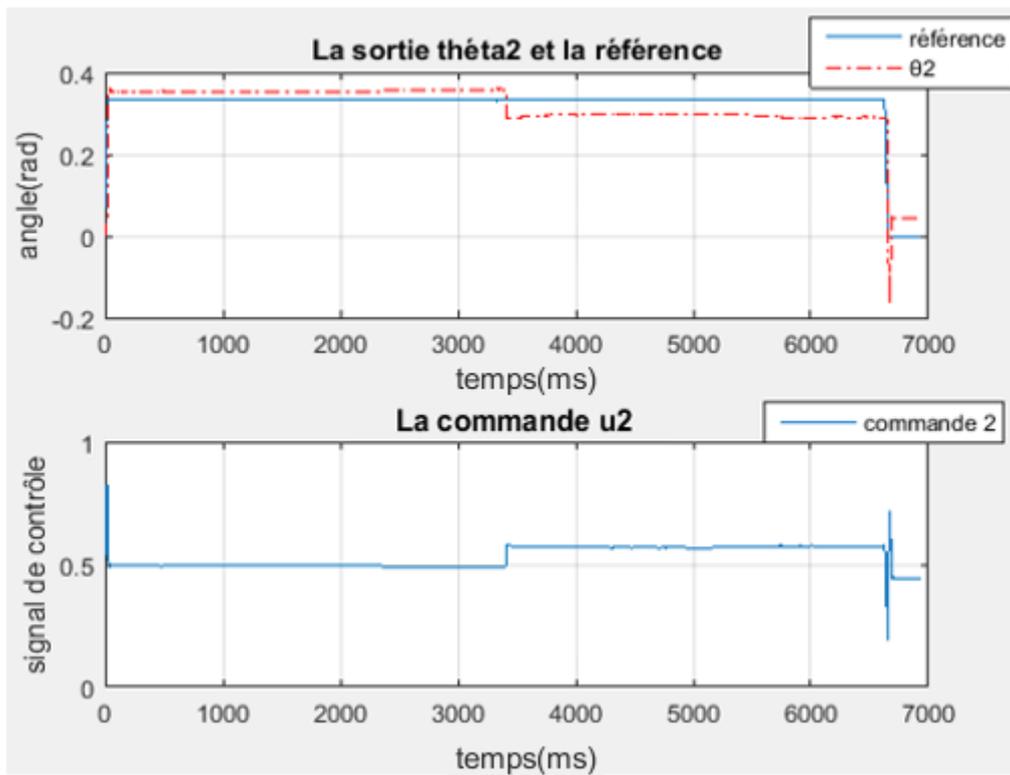


Figure 4.28. Poursuite de la trajectoire de référence pour l'articulation θ_2 dans le cas du demi-cercle $R=0.33$ m.

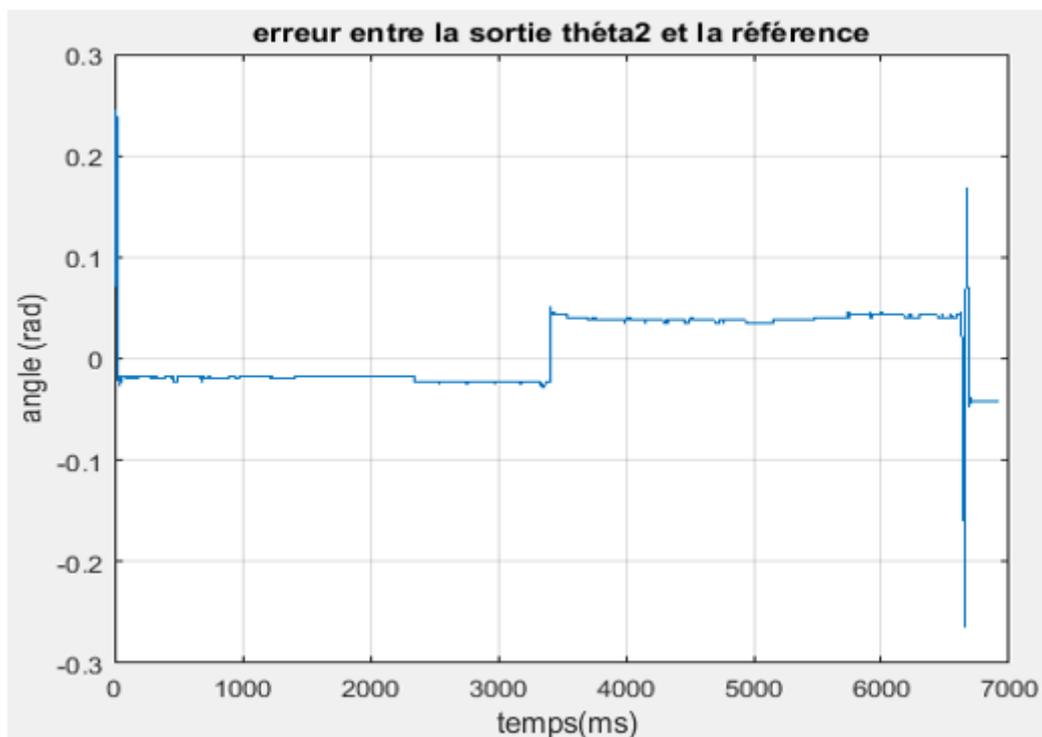


Figure 4.29. Erreur entre la référence et la sortie θ_2 dans le cas du demi-cercle $R=0.33$ m.

D'après les figures précédentes nous observons qu'il y a un petit écart entre la position angulaire et la référence. Cet écart est dû au couple moteur utilisé et à la mécanique du bras.

4.7 Conclusion

Nous avons introduit dans ce chapitre les différents éléments que nous avons utilisé telles que la conception avec SolidWorks de la forme finale du bras, et la carte DSP pour faciliter la tâche d'implémentation des algorithmes de commande développés. Ensuite, nous avons modélisé le système (les relations couples – positions angulaires) en utilisant les réseaux de neurones, et les résultats obtenus montrent les capacités d'apprentissage et d'approximation des réseaux de neurones. Enfin, pour mettre en évidence les performances des deux PID optimisés à l'aide de l'algorithme TLBO, nous avons effectué quelques essais de commande sur le bras réalisé. Les résultats obtenus sont satisfaisants.

Conclusion générale

Conclusion générale

L'objectif principal de notre projet, est la conception d'un bras manipulateur de type SCARA à deux degrés de liberté. Nous avons couvert tous les aspects de ce type de travail, dans un premier temps nous avons préparé et construit la partie mécanique qui est constituée des bras et des joints. Afin d'assurer leur mouvement nous les avons reliés par des moteurs à courant continu (MCC) qui sont dotés des encodeurs pour mesurer la position angulaire des différentes articulations. Ensuite, nous avons établi une modélisation géométrique, cinématique de la structure réalisée. Vu la difficulté d'obtenir les valeurs numériques du modèle dynamique, nous avons utilisé dans la partie simulation de la commande un modèle existant dans la bibliographie.

Le développement de l'algorithme de commande qui est basé sur la régulation PID nous a nécessité une phase de compréhension et de maîtrise de l'outil d'optimisation utilisée, à savoir l'algorithme TLBO. En fait, dans un premier lieu, nous avons utilisé le logiciel MatLab pour coder cet algorithme. Le code que nous avons développé a été testé sur des fonctions mathématiques. Une fois cet outil est maîtrisé et le programme d'optimisation est corrigé, nous avons abordé le problème de la détermination des paramètres des différents régulateurs PID utilisés tout en optimisant un critère de performances en utilisant l'algorithme d'optimisation TLBO. Pour mettre en évidence les performances des régulateurs PID obtenus, nous avons effectué plusieurs simulations.

Dans la phase de détermination des paramètres des régulateurs PID, au lieu d'utiliser le modèle dynamique du robot pour lequel nous n'avons pas les valeurs numériques de ses paramètres et l'intégration des équations différentielles qui pose des problèmes de convergence, nous avons développé des modèles neuronaux pour le bras considéré. Une partie de notre travail a été donc consacrée à la maîtrise de l'identification des systèmes en utilisant les réseaux de neurones. L'évaluation de la précision des modèles neuronaux obtenus a été faite en utilisant des données expérimentales.

Le système de commande que nous avons réalisé est composé principalement du kit de développement ezdsp F28335 dans lequel nous avons implémenté l'algorithme de commande développé, d'un circuit de puissance, d'un circuit d'isolation galvanique, une partie de régulation de tension, et des moteurs à courant continu.

Une fois les différents circuits ont été réalisés et l'algorithme de commande est développé, corrigé et implémenté dans le DSP, nous avons effectué plusieurs expériences pour mettre en évidence les performances de la commande développée. Les résultats obtenus en considérant plusieurs trajectoires de référence ont montré l'efficacité et les bonnes performances, en termes de rapidité, valeurs de dépassement et précision de poursuite, de ce type de commande.

La réalisation de ce projet nous a permis, plus particulièrement, de découvrir l'optimisation méta-heuristique, et d'une manière générale, de maîtriser plusieurs outils de travail à savoir : la programmation avec Matlab, les réseaux de neurones appliqués à l'identification, la programmation des DSP, la réalisation de cartes électroniques et de toucher de près à la régulation et la commande des systèmes. Au cours de la réalisation de ce projet, nous avons rencontré un certain nombre de difficultés liées à la non disponibilité des tous les composants électroniques et les moteurs à courant continu et à la fabrication des différentes pièces mécaniques constituant le bras manipulateur.

Notre travail peut faire l'objet d'autres améliorations dans les futurs projets, quant à sa structure mécanique ou l'algorithme de commande. Une fois la réalisation est complètement finalisée, elle peut servir dans la pédagogie comme un banc de travaux pratiques ou dans la recherche pour évaluer les performances des algorithmes de commande développés.

**Références
bibliographiques**



Références

- [1] W. Khalil and E. Dombre, "Bases de la modélisation et de la commande des robots-manipulateurs de type série" , Cours robotique, Université numérique ingénierie et technologie (UNIT), 08 mai 2012, France, Disponible sur : « http://cours-online.gdr-robotique.org/Khalil-Dombre_Modelisation/Khalil-Dombre_Modelisation.pdf ».
- [2] R. S. Hartenberg and J. Denavit, "A kinematic notation for lower pair mechanisms based on matrices", Trans. of ASME, J. of Applied Mechanics, juin 1955.
- [3] W. Khalil and E. Dombre, "Modélisation et commande des robots", Hermès, 1988, France.
- [4] Ph. Bidaud, " Cours de Modélisation des Systèmes Robotiques ", Master Sciences de L'Ingénieur Spécialités : " Systèmes Avancés et Robotiques ", " Systèmes Avancés et Robotiques ", Université Pierre et Marie Curie, France.
- [5] R. Sharma, V. Kumar, P. Gaur, and A. P. Mittal, "An adaptive PID like controller using mix locally recurrent neural network for robotic manipulator with variable payload," vol. 62, pp. 258-267, ISA Transactions, mai 2016, India.
- [6] R. V. Rao, "Teaching-learning-based optimization algorithm", Department of mechanical Engineering Sardar Vallabhbhai National Institute of Technology, Springer, 2016, pp. 9-39, India.
- [7] G. Gallot. " Modélisation Dynamique et Commande d'un robot Anguille", Automatique / Robotique. Ecole Centrale de Nantes (ECN), Décembre 2007, France.
- [8] A. El Dor, "Perfectionnement des algorithmes d'optimisation par essaim particulaire: applications en segmentation d'images et en électronique" Université Paris-Est, 2012, France.
- [9] R.V. Rao and V. Patel, "An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems", Sharif University of Technology, June 2013, India.
- [10] F. Glover, "Future paths for integer programming and links to artificial intelligence", Computers & operations research, vol 13, pp. 533-549, May 1986, USA.
- [11] I. Boussaid, "Perfectionnement de métaheuristiques pour l'optimisation continue", Université Paris Est, 2013, France.
- [12] Texas Instrument, DSPs the TMS320F2833x, architecture & peripherique, technical reference 2011 (41 43) p.
- [13] C. Bergmann, "Electronique et communication BTS industriels", Dunod, 2005, France.



- [14] G. Daniel, "Auto-ajustement de régulateurs PID robustes dans le domaine fréquentiel", École Polytechnique Fédérale de Lausanne, Thèse No 3520, 5 mai 2006, Suisse.
- [15] M. F. Rahman, D. Patterson, A. Cheok, and R. Betz, "30 - Motor Drives", in *Power Electronics Handbook (Fourth Edition)*, M. H. Rashid, Ed.: Butterworth-Heinemann, pp. 945-1021, 2018, England.
- [16] W. R. Perkins, "15 - Feedback Control Systems", in *Reference Data for Engineers (Ninth Edition)*, W. M. Middleton and M. E. Van Valkenburg, Eds. Woburn: Newnes, 225 Wildwood Avenue, pp. 15-1-15-33, 2002, USA.
- [17] S. Yadav, S. K. Verma, and S. K. Nagar, "Performance enhancement of magnetic levitation system using teaching learning based optimization", *Alexandria Engineering Journal*, vol. 57, no. 4, pp. 2427-2433, December 2018, India.
- [18] J. P. Lallemand, S. Zegloul, "Robotique", Aspects fondamentaux, Masson, 1994, France.



ANNEXE



Liste des composants

Circuit d'isolation :

6N137.

R4=R5 : 4.7 K Ω .

R13= R12 : 150 Ω .

Circuit de puissance :

C1 :22 nF.

C3 :15 nF.

C6 :15 nF.

C7 :220 nF.

IC4 : L6203.

D1=D2=D11 :1N4004. (Les diodes)

Circuit de régulation :

C2 :100 nF.

C4 :0.33 μ F.

C5 :0.1 μ F.

C8 :100 μ F.

C9 :1000 μ F.

R1 : 10 Ω

IC2 :7805T

IC1A=IC1B : 74LS14N. (Les inverseurs)

Datasheet 6N137 :

SINGLE-CHANNEL
6N137
HCPL-2601
HCPL-2611

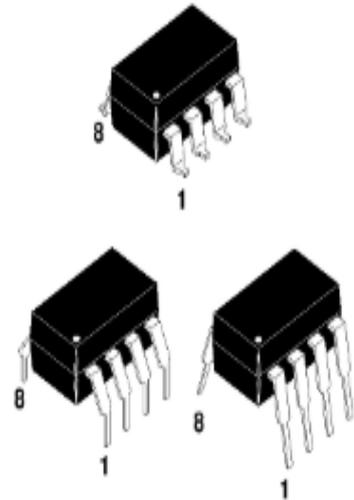
DUAL-CHANNEL
HCPL-2630
HCPL-2631

DESCRIPTION

The 6N137, HCPL-2601/2611 single-channel and HCPL-2630/2631 dual-channel optocouplers consist of a 850 nm AlGaAs LED, optically coupled to a very high speed integrated photodetector logic gate with a strobable output. This output features an open collector, thereby permitting wired OR outputs. The coupled parameters are guaranteed over the temperature range of -40°C to $+85^{\circ}\text{C}$. A maximum input signal of 5 mA will provide a minimum output sink current of 13 mA (fan out of 8).

An internal noise shield provides superior common mode rejection of typically 10 kV/ μs . The HCPL-2601 and HCPL-2631 has a minimum CMR of 5 kV/ μs .

The HCPL-2611 has a minimum CMR of 10 kV/ μs .

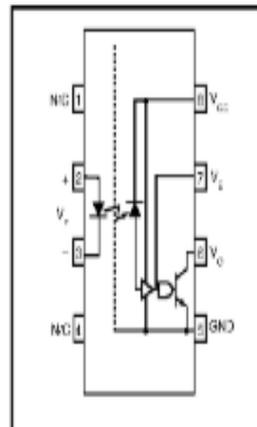


FEATURES

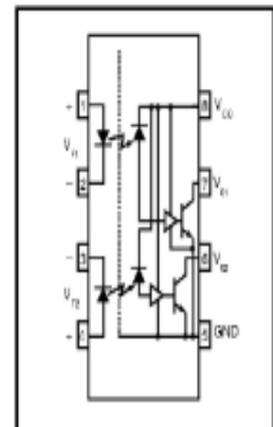
- Very high speed-10 MBit/s
- Superior CMR-10 kV/ μs
- Double working voltage-480V
- Fan-out of 8 over -40°C to $+85^{\circ}\text{C}$
- Logic gate output
- Storable output
- Wired OR-open collector
- U.L. recognized (File # E90700)

APPLICATIONS

- Ground loop elimination
- LSTTL to TTL, LSTTL or 5-volt CMOS
- Line receiver, data transmission
- Data multiplexing
- Switching power supplies
- Pulse transformer replacement
- Computer-peripheral interface



Single-channel
circuit drawing



Dual-channel
circuit drawing



SINGLE-CHANNEL
6N137
HCPL-2601
HCPL-2611

DUAL-CHANNEL
HCPL-2630
HCPL-2631

ABSOLUTE MAXIMUM RATINGS (No derating required up to 85°C)				
Parameter		Symbol	Value	Units
Storage Temperature		T_{STG}	-55 to +125	°C
Operating Temperature		T_{OPR}	-40 to +85	°C
Lead Solder Temperature		T_{SOL}	260 for 10 sec	°C
EMITTER				
DC/Average Forward Current	Single channel	I_F	50	mA
Input Current	Dual channel (Each channel)		30	
Enable Input Voltage	Single channel	V_E	5.5	V
Not to exceed V_{CC} by more than 500 mV				
Reverse Input Voltage	Each channel	V_R	5.0	V
Power Dissipation	Single channel	P_I	100	mW
	Dual channel (Each channel)		45	
DETECTOR				
Supply Voltage		V_{CC} (1 minute max)	7.0	V
Output Current	Single channel	I_O	50	mA
	Dual channel (Each channel)		50	
Output Voltage	Each channel	V_O	7.0	V
Collector Output	Single channel	P_O	85	mW
Power Dissipation	Dual channel (Each channel)		60	

RECOMMENDED OPERATING CONDITIONS				
Parameter	Symbol	Min	Max	Units
Input Current, Low Level	I_{FL}	0	250	μ A
Input Current, High Level	I_{FH}	*6.3	15	mA
Supply Voltage, Output	V_{CC}	4.5	5.5	V
Enable Voltage, Low Level	V_{EL}	0	0.8	V
Enable Voltage, High Level	V_{EH}	2.0	V_{CC}	V
Low Level Supply Current	T_A	-40	+85	°C
Fan Out (TTL load)	N		8	

* 6.3 mA is a guard banded value which allows for at least 20 % CTR degradation. Initial input current threshold value is 5.0 mA or less



Datasheet L6203 :



**L6201
L6202 - L6203**

DMOS FULL BRIDGE DRIVER

- SUPPLY VOLTAGE UP TO 48V
- 5A MAX PEAK CURRENT (2A max. for L6201)
- TOTAL RMS CURRENT UP TO
L6201: 1A; L6202: 1.5A; L6203/L6201PS:4A
- $R_{DS(ON)}$ 0.3 Ω (typical value at 25 °C)
- CROSS CONDUCTION PROTECTION
- TTL COMPATIBLE DRIVE
- OPERATING FREQUENCY UP TO 100 KHz
- THERMAL SHUTDOWN
- INTERNAL LOGIC SUPPLY
- HIGH EFFICIENCY

DESCRIPTION

The I.C. is a full bridge driver for motor control applications realized in Multipower-BCD technology which combines isolated DMOS power transistors with CMOS and Bipolar circuits on the same chip. By using mixed technology it has been possible to optimize the logic circuitry and the power stage to achieve the best possible performance. The DMOS output transistors can operate at supply voltages up to 42V and efficiently at high switch-

MULTIPOWER BCD TECHNOLOGY



Powerdip 12+3+3



SO20 (12+4+4)



Multiwatt11

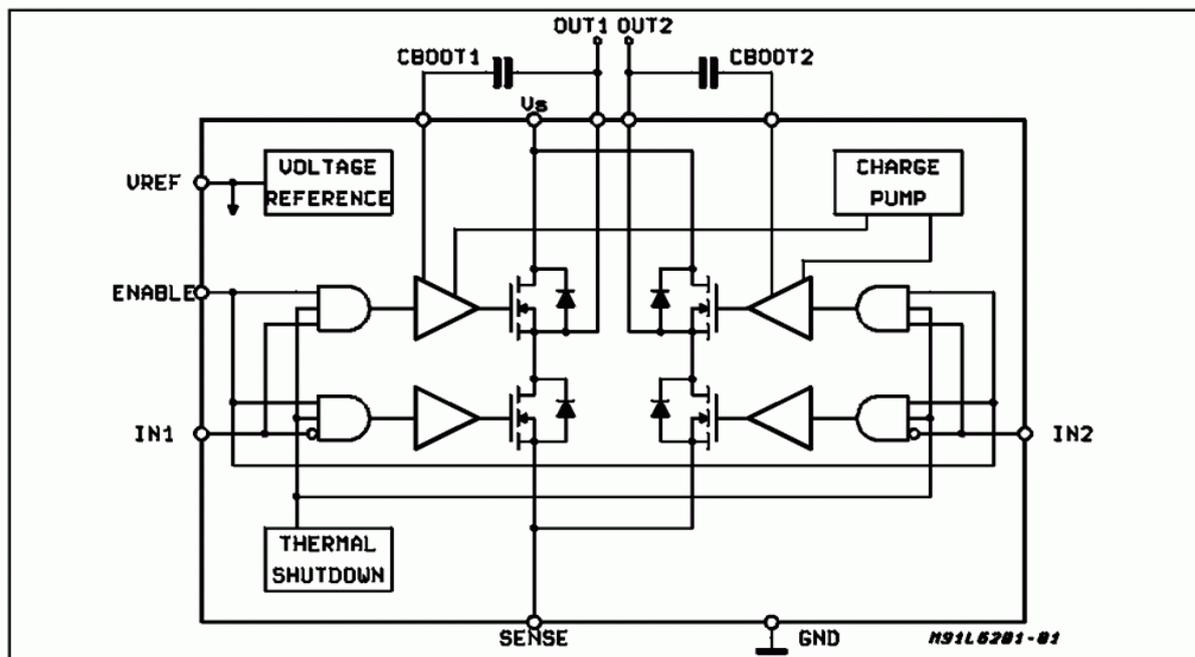


PowerSO20

ORDERING NUMBERS:
L6201 (SO20)
L6201PS (PowerSO20)
L6202 (Powerdip18)
L6203 (Multiwatt)

ing speeds. All the logic inputs are TTL, CMOS and μ C compatible. Each channel (half-bridge) of the device is controlled by a separate logic input, while a common enable controls both channels. The I.C. is mounted in three different packages.

BLOCK DIAGRAM



Datasheet7805T :

7805 • THREE-TERMINAL POSITIVE VOLTAGE REGULATOR IC

FEATURES:

- OUTPUT CURRENT IN EXCESS OF 1A;
- NO EXTERNAL COMPONENTS REQUIRED;
- INTERNAL SHORT CIRCUIT CURRENT LIMITING;
- INTERNAL THERMAL OVERLOAD PROTECTION;
- OUTPUT TRANSISTOR SAFE-AREA COMPENSATION;
- OUTPUT VOLTAGE OFFERED IN 4% TOLERANCE.

ABSOLUTE MAXIMUM RATINGS (Ta= 25° C)

Characteristic	Symbol	Norm	Unit
Input Voltage	Vin	V	35
Maximum Dissipated Power(with heat sink)	Ptot(max)	W	15
Maximum Dissipated Power(without heat sink)	Ptot(max)	W	1.5
Thermal Resistance Junction to Case	OjC	°C/W	5.0
Thermal Resistance, Junction to Air	OjA	°C/W	65
Junction Temperature	Tj	150	°C

Tc=-45+70°C

ELECTRICAL CHARACTERISTICS

(Vin=10V,Io=0.5A,Ci=0.33mkF,Co=0.1mkF,Tj=0+125°C, unless otherwise noted.)

Characteristic	Symbol	Norm			Unit
		Min	TYP	Max	
Output Voltage(Tj=25°C)	Vo	4.8		5.2	V
Output Voltage (5.0mA≤Io≤1.0A,Po≤15W) 7.0V≤Vin≤20V	Vo	4.75		5.25	V
Line Regulation(Tj=+25°C) 7.0V≤Vin≤25 V 8.0 V≤Vin≤12 V	ΔVv			100 50	mV
Load Regulation(Tj=+25°C) 5.0mA≤Io≤1.5A 0.25A≤Io≤0.75A	ΔVi			100 50	mV
Quiescent Current(Tj=+25°C)	Ib			8.0	mA
Quiescent Current Change 7.0 V≤Vin≤25 V 5.0mA≤Io≤1.0 A	ΔIb			1.3 0.5	mA
Dropout Voltage(Io=1.0A,Tj=+25°C)	Vi-Vo		2.0		V
Short Circuit Current Limit(Ta=-25°C),Vin=35V	Isc		0.4		A
Peak Output Current(Tj=+25°C)	I _{max}		2.2		A
Average Temperature Coefficient of Output Voltage	TCVo		0.3		mV/°C

