

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université 8 Mai 1945 Guelma



Faculté des Sciences et de la Technologie
Département de Génie Electrotechnique et d'Automatique

THÈSE

EN VUE DE L'OBTENTION DU DIPLOME DE
DOCTORAT EN SCIENCE

Filière : GENIE ELECTRIQUE

Présentée par

LOUCIF FATIHA

Intitulée

**Contribution à la commande non linéaire des robots
manipulateurs**

Soutenue le : 06/10/2021

Devant le Jury composé de :

Pr BOULOUH	Messaoud	Univ. de Guelma	Président
Pr KECHIDA	Sihem	Univ. de Guelma	Encadreur
Dr SEBBAGH	Abdennour	Univ. de Guelma	Co-encadreur
Dr KELAIAIA	Ridha	Univ. de Skikda	Examineur
Dr YAKHELEF	Yassine	Univ. de Boumerdes	Examineur

Année Universitaire : 2020/2021



Table des matières

Remerciements.....	i
Dédicaces.....	ii
Résumés.....	iii
Productions scientifiques.....	v
Liste des figures.....	vi
Liste des tableaux.....	ix
Abréviations et nomenclatures.....	xi
Introduction Générale.....	1
Chapitre1 : Commande des robots manipulateurs	
1.1 Introduction	5
1.2 Stratégies de contrôle des robots manipulateurs	6
1.2.1 Contrôle linéaire de position.....	6
1.2.2 Contrôle non linéaire de position.....	6
1.2.3 Contrôle de force.....	7
1.3 Commande des bras manipulateurs	7
1.3.1 Commande par articulation.....	8
1.3.2 Commande jacobéenne.....	8
1.3.3 Commande par découplage non linéaire.....	8
1.3.4 Commande adaptative.....	9
1.3.5 Commande fondée sur une fonction de Lyapunov.....	9
1.3.6 Commande passive.....	9
1.3.7 Commande prédictive.....	10
1.3.8 Commande robuste.....	10
1.3.9 Commande optimale.....	10
1.4 Conclusion.....	10
Références.....	11
Chapitre 2 : Modélisation des robots manipulateurs	
2.1 Introduction.....	15
2.2 Conception du manipulateur et capteurs.....	15

2.3 Modélisation	16
2.3.1. Modélisation géométrique.....	16
2.3.1.1 <i>Modèle géométrique direct (MGD)</i>	17
➤ <i>Calcul du modèle géométrique direct</i>	17
2.3.1.2 <i>Modèle géométrique indirect (MGI)</i>	18
➤ <i>Calcul du modèle géométrique indirect</i>	19
2.3.2 Modélisation cinématique.....	20
2.3.2 .1 <i>Modèle cinématique direct</i>	20
2.3.2.2 <i>Modèle cinématique inverse</i>	21
2.3.3 Modélisation Dynamique	22
2.3.3.1 <i>Modèle dynamique inverse</i>	22
i. <i>Formalisme de Lagrange</i>	23
ii. <i>Calcul de l'énergie cinétique et potentielle d'un bras manipulateur</i> <i>à 2ddl</i>	24
2.3.3.2 <i>Modèle dynamique directe</i>	26
2.4 Conclusion.....	26
Références.....	27

Chapitre 3 : Les algorithmes d'optimisation

3.1 Introduction	28
3.2 Les algorithmes d'optimisation	31
3.2.1 Optimiseur d'essaim de particules PSO	31
3.2.2 Optimiseur des loups gris (GWO)	31
3.2.2.1 <i>Le comportement d'encerclement</i>	31
3.2.2.2 <i>Le modèle de chasse</i>	32
3.2.3 Algorithme d'optimisation des baleines (WOA).....	32
3.2.4 Algorithme de sinus cosinus (SCA).....	33
3.2.5 Optimiseur du lion de fourmi (ALO)	34
3.2.5.1 <i>Marches aléatoires de fourmis</i>	36
3.2.5.2 <i>Piégeage dans les fosses des lions des fourmis</i>	36
3.2.5.3 <i>Construction de piège</i>	36
3.2.5.4 <i>Glissement des fourmis vers lion de fourmi</i>	37
3.2.5.5 <i>Attraper des proies et reconstruire la fosse</i>	37

3.2.5.6	<i>Élitisme</i>	38
3.2.6	Algorithme du moule visqueux (SMA)	38
3.2.6.1	<i>Concept et élicitation</i>	38
3.2.6.2	<i>Modèle mathématique</i>	39
i.	<i>Approche de la nourriture (Approach food)</i>	39
ii.	<i>Enveloppe de la nourriture (Wrap food)</i>	40
iii.	<i>Nourriture de grable (Grable food)</i>	40
3.3	Conclusion.....	41
	Références.....	42
Chapitre 4 : Commandes classiques des robots manipulateurs		
4.1	Introduction	43
4.2	Commande PID	45
4.2.1	Résultats de simulation	46
4.2.2	Commande PID du robot manipulateur	52
4.2.3	Tests de Robustesse.....	55
4.3	Commande à Couple Calculé.	57
4.3.1	Conception du contrôleur à couple calculé	58
4.3.2	Conception de boucle extérieure type PD (commande à couple calculé type PD)	59
4.3.2.1	<i>Choix des gains PD</i>	60
4.3.2.2	<i>Résultats de simulation</i>	61
i.	<i>Pour une commande sinusoïdale</i>	61
ii.	<i>Pour une commande constante</i>	63
4.3.3	Conception de boucle extérieure type PID (commande à couple calculé type PID)	64
4.3.3.1	<i>Résultats de simulation</i>	66
i.	<i>Pour une commande sinusoïdale</i>	66
ii.	<i>Pour une commande constante</i>	69
4.4	Contrôleur de type PD-Plus gravité.....	70
4.4.1	Résultats de simulation.....	72
i.	<i>Pour une commande sinusoïdale</i>	72
ii.	<i>Pour une commande constante</i>	77
4.5	Conclusion	79

Références.....	79
-----------------	----

Chapitre 5: Commande par mode glissant des robots manipulateurs

5.1 Introduction	84
5.2 Commande par mode glissant	85
5.3 Formulation du problème	86
5.4 Existence d'un mode glissant.....	87
5.5 Conception de la surface de glissement	88
5.6 Modes glissants du second ordre : première génération.....	89
5.6.1 Algorithme de torsion (Twisting Algorithme).....	89
5.6.2 Algorithme de super-torsion (Super-Twisting Algorithm)	90
5.6.3 Algorithme sous-optimal (Sub-Optimal Algorithm)	90
5.6.4 Deuxième génération : Contrôleurs mode glissant d'ordre arbitraire	90
5.7 Contrôle par mode de glissement terminal (Terminal Sliding Mode Control)	92
5.8 Troisième génération : HOSMs non homogènes (Non-Homogeneous HOSMs).....	93
5.9 Contrôle par intégral du mode glissant (Integral Sliding Mode Control).....	93
5.10 Suppression et atténuation des broutements dans les systèmes mode glissant.....	95
5.11 Commande par mode glissant d'un robot manipulateur.....	95
5.12 Commande par mode glissant avec surface proportionnelle intégrale dérivée (SMC_PID) d'un robot manipulateur.....	96
5.13 Résultats de simulation	98
i. <i>Optimisation SMC_PID sans perturbations</i>	99
ii. <i>Optimisation SMC_PID avec perturbations de couple</i>	103
5.14 Conclusion.....	106
Références.....	107

Chapitre 6 : Commande par backstepping des robots manipulateurs

6.1 Introduction	112
6.2 Commande par backstepping	113
6.3 Synthèse d'une loi de commande par backstepping.....	113
6.3.1 Loi de commande backstepping pour le 1 ^{er} segment (1 ^{er} sous-système)	115

6.3.2 Loi de commande backstepping pour le 2 ^{ème} segment (2 ^{ème} sous-système)	116
6.4 Résultats de simulation	117
6.4.1 Optimisation du contrôle backstepping sans perturbations	117
6.4.2 Optimisation du contrôle backstepping avec perturbations de couple	120
6.5 Conclusion	123
Références	124
Conclusion Générale	127
Annexe	131

Remerciements

Au nom de Dieu, le Clément, le Miséricordieux.

Après avoir terminé cette thèse, après près de trois ans, je voudrais tout d'abord remercier Le Bon Dieu, qui m'a aidé et m'a donné la force pour poursuivre et le courage pour surmonter toutes les difficultés.

Le travail présenté dans cette thèse a été réalisé au sein du laboratoire d'Automatique et Informatique de Guelma (LAIG) sous la direction du Pr Sihem Kechida qu'elle trouve ici toute ma gratitude.

Mes vifs remerciements vont à mon encadreur Pr. Kechida sihem qui m'a dirigé et suivie tous le long de cette période. J'aimerais aussi la remercier pour sa disponibilité et ses précieux conseils qui m'ont permis de mener à bien ce travail.

Je tiens à exprimer toute ma reconnaissance à **Pr. Boulouh Messaoud** d'avoir accepté de présider mon jury de soutenance, je remercie également **Dr. KELAIAIA Ridha** de l'université de Skikda et **Dr. YAKHLEF Yassine** de l'université de Boumerdes d'avoir accepté d'examiner et de critiquer ce modeste travail. Leurs remarques et leurs suggestions constructives vont contribuer qu'à la rendre meilleure.

Enfin, un immense merci à mon mari Nacer pour ses encouragements et son soutien quotidien, et à mes enfants qui ont été ma source d'énergie et de motivation. J'associe ces remerciements à mes chers parents, pour leurs prières qui éclairent mon chemin.

Dieu merci, pour ce résultat, fruit des années, que je l'espère être utile.

وماتوفيقي إلا بالله



Dédicaces

A ma petite famille : *Nacer, Ayoub, loudjayne et Rimes.*

A ma grande famille : *mes parents, mes sœurs et frères.*

A ma sœur : *Wassila Que Dieu vous protège, vous guérisse, et faite de
Vous la prunelle de vos enfants et faite-en la prunelle de vos yeux.*

Résumé : Cette thèse présente la commande d'un robot manipulateur pour le suivi de trajectoire. Différentes techniques ont été utilisées pour la commande d'un bras à 2 degrés de libertés telles que : la commande PD, PID classique, la commande à couple calculé type PD et PID, la commande PD par compensation de gravité, la commande par mode glissant et la commande par backstepping. La conception des différents contrôleurs a été faite par l'investigation des algorithmes d'optimisation récents tels que : PSO, GWO, WOA, SCA, ALO et SMA. Une étude comparative entre ces algorithmes a prouvé en premier temps la supériorité de WOA par rapport au GWO et PSO en terme de vitesse de convergence, taille de population et meilleur optimum trouvé pour la commande PID classique. Ensuite, l'application de l'ALO pour la commande par mode glissant a montré une amélioration en temps de calcul et en précision pour le choix de l'optimum global par rapport aux GWO, WOA et SCA. D'autre part, les résultats de simulation ont prouvés la robustesse de la commande par mode glissant en tant qu'une commande non linéaire robuste vis à vis les perturbations tandis que les commandes PIDs classiques se comportent bien pour des vitesses faibles du robot mais présentent ces limites lorsque le robot nécessite des vitesses élevés ou bien soumis à des perturbations. La commande à couple calculé nécessite l'exactitude du modèle du robot et ne tient pas en compte les perturbations, et la commande par compensation de gravité présente la simplicité de construction par rapport aux autres techniques précitées. L'algorithme SMA a dominé les autres algorithmes dans la commande par backstepping, cette dernière a surmonté la commande par mode glissant en terme de poursuite de trajectoire et rejet des perturbations en assurant une stabilité globale du robot à cause de la fonction de Lyapunov utilisée dans la conception du contrôleur. Chacune des techniques de commande précitée présente des avantages et des inconvénients, le choix du type de commande dépend du cahier de charge et des critères dominants à titre d'exemple : la stabilité, la rapidité, la précision et le rejet des perturbations.

Mots-clés : Robot manipulateur, Commande non-linéaire, commande PID, commande à couple calculé, commande par compensation de gravité, commande par mode glissant, commande par backstepping, Algorithmes d'optimisations.

المخلص : تتحور هذه الأطروحة حول التحكم في ذراع روبوتية (مناول ربوتي) ذات درجتين (2) حرية من أجل تتبع المسار. حيث تم استخدام تقنيات مختلفة للتحكم ك: المتحكم PD، PID التقليدي، نوع التحكم المحسوب في عزم الدوران PD و PID، التحكم PD عن طريق تعويض الجاذبية، التحكم عن طريق وضع الانزلاق و التحكم عن طريق backstepping ولقد تم تصميم وحدات التحكم أو المتحكمات المختلفة باستعمال خوارزميات الأمثل الحديثة ك: PSO، GWO، WOA، SCA، ALO و SMA. حيث أثبتت نتائج الأولى لدراسة مقارنة بين هذه الخوارزميات تفوق WOA على PSO و GWO من حيث سرعة التقارب، تعداد المجتمعات وأفضل حل امثل للمتحكم الكلاسيكي PID. بعد ذلك، أظهر تطبيق ALO للتحكم في الوضع الانزلاقي تحسناً في وقت الحساب والدقة لاختيار قيمة الملائم العام مقارنة بـ GWO و WOA و SCA. من ناحية أخرى، أثبتت نتائج المحاكاة متانة التحكم في الوضع المنزلق، وهو تحكم غير خطي، تجاه الاضطرابات فيما أبدي المتحكم PID الكلاسيكي، نتائج مقبولة عند سرعات منخفضة لروبوت الا انه اظهر محدوديته عند

سرعات مرتفعة لروبوت او عند تعرضه للاضطرابات. بينما يتطلب التحكم في عزم الدوران المحسوب دقة نموذج الروبوت ولا يأخذ بعين الاعتبار الاضطرابات الموجودة، فان التحكم في تعويض الجاذبية يمتاز ببساطته مقارنة بالتقنيات الأخرى المذكورة أعلاه. بالنسبة للخوارزميات، فقد تميزت خوارزمية SMA في التحكم backstepping على الخوارزميات الأخرى، والتي تغلبت على التحكم في الوضع الانزلاقي من حيث تتبع المسار وطرده الاضطرابات مع ضمان الاستقرار الكلي للروبوت وذلك بسبب دالة Lyapunov المستخدمة أثناء تصميم المتحكم. كل من تقنيات التحكم المذكورة أعلاه لها مزايا وعيوب، واختيار نوع التحكم يعتمد على دفتر المواصفات والمعايير السائدة على سبيل المثال: الاستقرار والسرعة والدقة ورفض الاضطرابات.

الكلمات الدالة: روبوت مناوول، تحكم غير خطي، تحكم PID، تحكم عزم الدوران المحسوب، التحكم بتعويض الجاذبية، تحكم الوضع الانزلاقي، التحكم الخلفي، خوارزميات الامثل.

Abstract: This thesis presents the control of a manipulator robot for trajectory tracking. Different techniques have been used to control an arm with 2 degrees of freedom such as: conventional PD, PID controls, computed torque control type PD and PID, PD gravity compensation control, the sliding mode control and the backstepping control. The design of different controllers was made using recent optimization algorithms such as: PSO, GWO, WOA, SCA, ALO and SMA. Firstly, a comparative study between these algorithms proved the superiority of WOA over GWO and PSO in terms of convergence speed, population size and best optimum for classical PID controller. Then, the application of ALO for sliding mode control showed an improvement in computing time and in precision for the choice of the global optimum compared to GWO, WOA and SCA. On the other hand, the simulation results have proven the robustness of the sliding mode control as a non-linear control against disturbances whereas the conventional PID controls behave well at low speeds of robot but present these limitations when the robot requires high speeds or subjected to disturbances. The computed torque control requires the accuracy of the robot model and does not take into account disturbances and the gravity compensation control presents the simplicity of construction compared to the other techniques mentioned above. The SMA algorithm dominated the previous algorithms in the backstepping control, the latter overcomes the sliding mode control in terms of trajectory tracking and rejection of disturbances ensuring overall stability of the robot thanks to the Lyapunov function used in the controller design. Each of the aforementioned control techniques has advantages and disadvantages, the choice of the type of control depends on specifications and the dominant criteria for example: stability, speed, precision and rejection of disturbances.

Keywords: Robot Manipulator, Non-linear control, PID control, computed torque control, gravity compensation control, sliding mode control, backstepping control, optimization algorithms.

1. **Loucif. F, Kechida. S, Sebbagh. A.** Whale optimizer algorithm to tune PID controller for the trajectory tracking control of robot manipulator. Journal of the Brazilian Society of Mechanical Sciences and Engineering, volume 42, pp: 1-11, 2020. DOI: 10.1007/s40430-019-2074-3.
2. **Loucif. F, Kechida. S.** Sliding Mode Control with PID Surface for Robot Manipulator Optimized by Evolutionary Algorithms. Recent Advances in Engineering Mathematics and Physics, pp : 19-32, Springer Nature Switzerland AG, 04, August 2020. DOI: 10.1007/978-3-030-39847-7.
3. **Loucif. F, Kechida. S.** Optimization of Sliding mode control with PID surface for robot manipulator by Evolutionary Algorithms. Open Computer sciences. 4 Novembre 2020. DOI: <https://doi.org/10.1515/comp-2020-0144>.
4. **Loucif. F, Kechida. S.** Design of optimal backstepping controller for a robot manipulator using Slime Mould Algorithm. Pour Soumission .

Articles de conférences internationales

1. **Loucif. F, Kechida. S.** Grey Wolf optimizer to tune PID controller for the trajectory tracking control of robot manipulator. 3rd International Conference on Technological Advances in Electrical Engineering ICATEE2018, Skikda, Algeria, Décembre 2018.
2. **Loucif. F, Kechida. S, Sebbagh. A.** Heuristic methods for optimization of a PID controller for Trajectory tracking of a manipulator robot. The 3rd International Conference on Electromechanical Engineering ICEE'2018, Skikda, Algeria, November 21-22, 2018.

Article de conférence nationale

1. **Loucif. F, Kechida. S, Sebbagh. A.** Whale optimizer algorithm to tune PID controller for the trajectory tracking control of robot manipulator. Journée 2019 sur les Signaux et les Systèmes JSS'19, 13 Novembre 2019, Laboratoire d'Automatique et Informatique de Guelma, université 8 mai 1945 de Guelma, Algeria.

Liste des figures

Figure 2.1	Modèle géométrique directe et inverse.....	16
Figure 2.2	Les paramètres géométrique d'un robot planaire à 2 ddl.....	17
Figure 3.1	Classification des méthodes évolutionnaires et IS.....	30
Figure 3.2	Comportement de chasse des loups gris : (A) chasser, approcher et suivre les proies. (B-D) poursuivre, harceler et encercler. (E) situation stationnaire et attaque.....	32
Figure 3.3	Comportement au filet à bulles des baleines.....	33
Figure 3.4	Effets de Sine et Cosine dans les équations (10) et (11) sur la position suivante.....	34
Figure 3.5	Pièges coniques et comportement de chasse des lions des fourmis.....	37
Figure 3.6	Promenade aléatoire d'une fourmi à l'intérieur d'un piège du lion des fourmis.....	38
Figure 3.7	Morphologie de recherche du moule visqueux.....	39
Figure 3.8	Tendances de \vec{vb} et \vec{vc}	41
Figure 4.1	Mécanisme de contrôle PID.....	45
Figure 4.2	Schéma Simulink de la stratégie de contrôle PID.....	46
Figure 4.3	Position actuel du robot manipulateur contrôlé par PID-PSO.....	50
Figure 4.4	Position actuel du robot manipulateur contrôlé par PID-GWO.....	51
Figure 4.5	Position actuel du robot manipulateur contrôlé par PID-WOA.....	52
Figure 4.6	Position actuel du robot manipulateur contrôlé par PID optimisé par les trois algorithmes.....	53
Figure 4.7	(a) Signal d'erreur pour la première et la deuxième articulation. (b) Courbe de Convergence de WOA algorithme.....	54
Figure 4.8	Fonction de coût de l'algorithme WOA.....	55
Figure 4.9	(a) Position de la première et la deuxième articulation sans perturbation, (b) Couple de contrôle.....	56
Figure4.10	(a) Position de la première et la deuxième articulation avec perturbation (variance 0.8), (b) Couple de contrôle.....	57
Figure 4.11	Mécanisme de Couple calculé type PD.....	60

Figure 4.12	(a), (b) Positions des deux articulations commandées par couple calculé type PD pour une entrée sinusoïdale.....	62
Figure 4.13	Erreur de position des deux articulations.....	63
Figure 4.14	Commande par calcul de couple type PD d'un robot à 2 ddl.....	63
Figure 4.15	Position des deux articulations commandé par couple calculé type PD pour une entrée constante.....	64
Figure 4.16	Mécanisme de couple calculé type PID.....	65
Figure 4.17	Position des deux articulations commandé par couple calculé type PID pour une entrée sinusoïdale.....	67
Figure 4.18	Erreur de position des deux articulations.....	68
Figure 4.19	Commande par calcul de couple -PID d'un robot à 2 ddl.....	69
Figure 4.20	Position des deux articulations commandé par couple calculé type PID pour une entrée constante.....	70
Figure 4.21	Position des deux articulations commandé par PD plus gravité pour une entrée sinusoïdale.....	73
Figure 4.22	Erreur de position des deux articulations.....	74
Figure 4.23	Commande par PD plus gravité d'un robot à 2 ddl.....	74
Figure 4.24	Position des deux articulations commandé par PD plus gravité pour une entrée sinusoïdale avec perturbation.....	75
Figure 4.25	Erreur de position des deux articulations avec perturbation.....	76
Figure 4.26	Commande par PD plus gravité d'un robot à 2 ddl avec perturbation.....	76
Figure 4.27	Position des deux articulations commandé par PD plus gravité pour une entrée constante.....	78
Figure 5.1	Broutement en raison d'une commutation de commande imparfaite.....	86
Figure 5.2	Attractivité du collecteur glissant.....	87
Figure 5.3	Le couple du contrôle pour la première et la deuxième articulation du robot manipulateur.....	100
Figure 5.4	Suivi de trajectoire des liaisons 1 et 2 avec contrôleur SMC_PID optimisé sans perturbations.....	100
Figure 5.5	Courbes de convergence des critères ISTE.....	101
Figure 5.6	Position X de l'organe terminal du robot à 2ddl contrôlé par le contrôleur ALO_SMC_PID.....	101

Figure 5.7	Position Y de l'organe terminal du robot à 2ddl contrôlé par le contrôleur ALO_SMC_PID.....	102
Figure 5.8	Position de l'organe terminal dans le plan XY du robot à 2ddl contrôlé par le contrôleur ALO_SMC_PID.....	102
Figure 5.9	Commande SMC_PID pour la première et la deuxième articulation du robot manipulateur.....	103
Figure 5.10	Suivi de position avec perturbations des deux articulations avec SMC_PID optimisé par ALO.....	105
Figure 6.1	Courbe de convergence des algorithmes d'optimisation.....	118
Figure 6.2	Suivi de trajectoire du robot manipulateur.....	119
Figure 6.3	Erreur de suivi de trajectoire du robot manipulateur.....	119
Figure 6.4	Couple de contrôle des deux articulations du robot manipulateur.....	120
Figure 6.5	Suivi de trajectoire du robot manipulateur optimisé par l'algorithme SMA avec une perturbation de couple de 0,8 de puissance.....	122
Figure 6.6	Couple de contrôle des deux articulations du robot manipulateur optimisé par l'algorithme SMA avec une perturbation de couple de 0,8 de puissance.....	122
Figure 1.A	Robot à structure ouverte simple.....	131
Figure 2.A	Paramètres géométriques dans le cas d'une structure ouverte simple.....	132

Liste des tableaux

Tableau 2.1	Configuration de Denavit-Hartenberg du robot à 2ddl.....	18
Tableau 4.1	Résultats de comparaison des caractéristiques de réponse de la première et la deuxième d'articulation contrôlées par PID-PSO.....	48
Tableau 4.2	Résultats de comparaison des caractéristiques de réponse de la première et la deuxième d'articulation contrôlées par PID-GWO.....	49
Tableau 4.3	Résultats de comparaison des caractéristiques de réponse de la première et la deuxième d'articulation contrôlées par PID-WOA.....	49
Tableau 4.4	Paramètres des contrôleurs PID-PSO, PID-GWO et PID-WOA.....	50
Tableau 4.5	Temps de convergence des trois algorithmes.....	50
Tableau 4.6	Caractéristique de la réponse du robot à la commande couple calculé type PD.....	61
Tableau 4.7	Caractéristique de la réponse du robot à la commande couple calculé type PD.....	63
Tableau 4.8	Caractéristique de la réponse du robot à la commande couple calculé type PID.....	66
Tableau 4.9	Caractéristique de la réponse du robot à la commande couple calculé type PID.....	69
Tableau 4.10	Caractéristique de la réponse du robot à la commande PD plus gravité.....	73
Tableau 4.11	Caractéristique de la réponse du robot à la commande PD plus gravité avec couple perturbateur.....	75
Tableau 4.12	Caractéristique de la réponse du robot à la commande PD plus gravité.....	77
Tableau 5.1	Fonction de coût pour la première et la deuxième articulation sans perturbations....	99
Tableau 5.2	Paramètres du SMC_PID pour la première et la deuxième articulation sans perturbations	99
Tableau 5.3	Fonction de coût ISTE en présence de perturbations de couple pour l'optimisation GWO.....	103
Tableau 5.4	Fonction de coût ISTE en présence de perturbations de couple pour l'optimisation WOA	104
Tableau 5.5	Fonction de coût ISTE en présence de perturbations de couple pour l'optimisation SCA.....	104
Tableau 5.6	Fonction de coût ISTE en présence de perturbations de couple pour l'optimisation ALO.....	104
Tableau 6.1	Fonction de coût pour la première et la deuxième articulation sans perturbations..	118

Tableau 6.2	Paramètres du contrôleur par backstepping pour la première et la deuxième articulation sans perturbations	118
Tableau 6.3	Paramètres du contrôleur par backstepping pour la première et la deuxième articulation avec perturbations de couple.....	121
Tableau 6.4	Comparaison entre la commande par backstepping et mode glissant.....	121
Tableau 6.5	Paramètres d'algorithmes d'optimisation pour le contrôle par backstepping.....	121
Tableau 6.6	Temps de calcul total des algorithmes d'optimisation.....	121

Abréviations et nomenclature

<i>Abréviation</i>	<i>Anglais</i>	<i>Français</i>
ABC	Artificial Bee Colony	Colonie d'abeilles artificielle
ACO	Ant Colony Optimization	Optimisation des colonies de fourmis
BA	Bat-inspired Algorithm	Algorithme inspiré des chauves-souris
BBO	Biogeography-based optimization	Optimisation basée sur la biogéographie
CFO	Central Force Optimization	Optimisation de la force centrale
CSS	Charged System Search	Recherche du système chargé
EA	Evolutionary algorithm	Algorithme évolutif
FOA	Fruit Fly Optimization	Optimisation de la mouche des fruits
GA	Genetic Algorithm	Algorithme Génétique
gbest	Global best	L'optimum globale
GLSA	Gravitational Local Search Algorithm	Algorithme de recherche locale gravitationnelle
GSA	Gravitational Search Algorithm	Algorithme de recherche gravitationnelle
GWO	Grey Wolf Optimizer	Optimiseur des loups gris
HHO	Harris Hawk Optimizer	Optimiseur du faucon de Harris
HOSM	High Order Sliding Mode	Mode de glissement d'ordre supérieur
ISM	Integral Sliding Mode	Mode de glissement intégral
MFO	Moth Flame Optimization	Optimisation de la flamme des mites
MVO	Multi-Verse Optimizer	Optimiseur multi-verse
pbest	Partial best	L'optimum partiel
PSO	Particule swarm optimizer	Optimiseur d'essaim de particules
SA	Simulated Annealing	Recuit simulé
SCA	Sine Cosine Algorithm	Algorithme de Sinus Cosinus
SMA	Slime mould Algorithm	Algorithme du Moule visqueuse
SMC	Sliding mode Controller	Contrôleur par mode glissant
SOA	Sub-Optimal Algorithm	Algorithme sous-optimal

SOSM	second order sliding mode	mode glissant du second ordre
SI	Swarm Intelligence	Intelligence en essaim
SISO	Single input single output	seul entrée seul sortie
STA	Super-Twisting Algorithm	Algorithme de super-torsion
TA	Twisting Algorithme	Algorithme de torsion
TLBO	Teaching-Learning-Based Optimization	Optimisation basée sur l'enseignement et l'apprentissage
TS	Tabu Search	Recherche de Tabu
TSM	Terminal sliding mode	Mode de glissement terminal
WOA	Whale Optimizer Algorithm	Optimiseur des baleines
WSO	Wasp Swarm Optimization	Optimisation de l'essaim de guêpes

Introduction générale

Introduction générale

La robotique, est une discipline scientifique, intersection de l'ingénierie et de la technologie, permettant la conception et la réalisation des machines automatiques appelées robots. C'est un domaine multidimensionnel impliquant diverses domaines technologiques telles que la physique, électronique, instrumentations, le contrôle, l'intelligence artificielle (IA)...Un robot est un dispositif mécatronique conçu pour accomplir automatiquement des tâches imitant ou reproduisant, dans un domaine précis, des actions humaines. Le terme robot apparaît pour la première fois dans la pièce de théâtre (science-fiction) R. U. R. (Rossum's Universal Robots), écrite en 1920 par l'auteur Karel Capek. Le mot a été créé par son frère Josef à partir du mot tchèque « robota » qui signifie travail, besogne, corvée. Les premiers robots industriels apparaissent, malgré leur coût élevé, au début des années 1970. Ils sont destinés à exécuter certaines tâches répétitives, éprouvantes ou toxiques pour un opérateur humain : peinture ou soudage des carrosseries automobiles. Industriels, militaires ou spécialistes chirurgicaux rivalisent d'inventivité pour mettre au point des robots assistants les aidant dans la réalisation de tâches délicates ou dangereuses. Aujourd'hui, l'évolution de l'électronique et de l'informatique permet de développer des robots plus précis, plus rapides avec une meilleure autonomie.

Généralement, un système robotique est considéré comme un système artificiel équipé de capteurs et d'actionneurs, conçu pour agir avec le monde qui l'entoure. Il existe de nombreux types de systèmes robotiques, à titre d'exemple main articulée, véhicule terrestre ou aérien, robot humanoïde, robot parallèle et les bras manipulateurs, etc. Ces derniers sont devenus des outils nécessaires dans l'environnement industriel permettant à automatiser des tâches s'étendant de plus répétitif au plus complexe. Les capacités de l'agent robotique sont déterminées par des capacités mécaniques et d'effort, la précision du mouvement, le nombre de degrés de liberté du bras, le degré de manipulabilité de la pince, la technologie des capteurs, la sophistication et la fiabilité des contrôleurs. Le robot manipulateur et son système de contrôle sont considérés comme une entité unique ou 'agent' pour l'interaction avec le monde extérieur c'est-à-dire qu'en absence d'un système de contrôle adéquat, un appareil robotique devient inutile. L'élaboration d'une loi de commande pour le robot manipulateur, vu comme un système non linéaire, nécessite la prise en compte de certains points (ou

hypothèses) tels que le suivi de la consigne, le rejet de la perturbation, une marge de robustesse vis-à-vis de certains paramètres de procédé à piloter. La littérature présente une multitude de structure de commande, chacune d'elle possède son contexte d'application et également des propriétés (cas linéaire, cas non linéaire, procédé stable, consigne d'un type donné, ...). Le problème de la stabilité possède en matière de commande une importance fondamentale ; la recherche d'une loi de commande garantissant la stabilité d'un système asservi peut se faire à l'aide des fonctions de Lyapunov. Ces dernières ont été depuis longtemps des outils importants dans la théorie de commande non linéaire. Cependant, la commande des robots manipulateurs présente un champ très intéressant pour la conception des contrôleurs robuste répondant aux exigences telles que la rapidité et la précision tout en assurant la stabilité du robot. Les lois de commandes classiques résolvent le problème de régulation ou de poursuite de trajectoire, parmi lesquelles on trouve la loi de commande PD qui assure la stabilité asymptotique globale de la dynamique du robot en boucle fermée, mais avec une erreur statique non nulle pour des gains proportionnels très grands ce qui est indésirables dans la pratique. Quant à la commande PID, elle garantit la stabilité asymptotique locale du robot en boucle fermée avec une erreur de position nulle. Bien que la présence du terme intégral puisse causer de sérieux problèmes de saturation, ce type de commande reste efficace dans le domaine industriel. Plusieurs applications utilisent des contrôleurs non-linéaire telle que la commande PD avec compensation de l'effet de gravité et la commande par découplage non linéaire, cette dernière présente des réponses très satisfaisante lorsque le système est connu avec exactitude, ce qui n'est pas toujours le cas à cause des frottements du robot, des perturbations internes ou externes, ..., non prise en compte dans la loi de commande.

Au cours des deux dernières décennies, on a assisté à un développement extraordinaire dans l'élaboration de lois de commande non linéaires robustes telle que la commande par backstepping, commande passive, commande par mode glissant, etc. La commande non linéaire avec linéarisation entrée-états ou entrée-sortie mène à l'annulation des non linéarités qui pourraient s'avérer utiles. Le backstepping est moins restrictif et n'oblige pas le système à devenir linéaire ce qui le rend une technique de commande puissante assurant une stabilité globale du robot. Malgré le succès que présente la commande par mode glissant en tant qu'une commande non linéaire simple à mettre en œuvre cette dernière souffre d'un inconvénient majeur qui est le broutement, ce qui rend la commande par backstepping une bonne alternative.

Contribution et organisation de la thèse

Cette thèse est principalement dédiée à la commande des robots manipulateurs où un bras à 2 degrés de liberté est choisi comme cas d'étude. La commande des robots manipulateurs est un champ très intéressant dans le domaine des systèmes non linéaire. Néanmoins, le choix d'un contrôleur qui répond à un cahier des charges représente un grand challenge, soit pour le suivi de trajectoire ou le rejet des perturbations, vu le couplage et les non linéarités du robot. Une fois le type de commande est bien choisi, le réglage des paramètres du contrôleur s'impose comme un autre défi. La majorité des techniques de commandes non linéaire ne décrit pas la méthodologie de choix des paramètres du contrôleur. Afin de répondre à cet objectif, nous faisons appel aux algorithmes d'optimisation pour la détermination des meilleurs paramètres, vu le grand succès que présentent récemment les algorithmes évolutionnaires. Ces derniers ont été introduits pour l'optimisation des paramètres des contrôleurs du robot. On se limite dans le cadre de ce travail aux techniques de commandes des robots manipulateurs et on se concentre sur les techniques précitées.

Cette thèse est organisée en six chapitres encadrée par une **introduction générale**, qui présente un aperçu global sur la commande des robots manipulateurs, et une **conclusion** qui récapitule les points importants et les observations concernant chaque technique de commande aussi bien les différents algorithmes d'optimisation utilisés, suivi de quelques perspectives et directives pour des futurs travaux.

Le chapitre 1 regroupe quelques techniques de commande des robots manipulateurs avec une description succincte de chaque type de commande.

L'approximation des phénomènes physiques mis en jeu du robot manipulateur, par une modélisation mathématique, est présentée dans **le chapitre 2**. C'est à partir de cette représentation que l'on peut synthétiser le système de commande pour le robot manipulateur.

Le chapitre 3 est consacré à la description du principe des différents algorithmes d'optimisation utilisé dans le cadre de cette étude.

Le chapitre suivant est dédié à la commande classique du robot telles que la commande PID (proportionnel, intégrale et dérivés), la commande par couple calculé, la commande par compensation de gravité, etc. Les résultats de simulation sont obtenus sous l'environnement de MATLAB/Simulink.

Le chapitre 5 dévoile les points forts de la commande par mode glissant, et détaille les différents types de cette technique ainsi que la description mathématique de chacune d'elles. La commande par mode glissant avec une surface PID est ensuite appliquée pour le suivi de trajectoire du robot et le rejet de perturbation. Le chapitre est enrichi par des discussions des résultats obtenus.

Une autre technique de commande non linéaire appelée la commande par backstepping est appliquée sur le robot manipulateur. Les différents tests sont présentés et discutés dans le **chapitre 6**.

Chapitre 1

Commande des robots manipulateurs

Commande des robots manipulateurs

1.1 Introduction

Les robots manipulateurs sont actuellement d'une très large utilisation dans les applications industrielles et spatiales. Ils sont d'une importance majeure, surtout dans les travaux dangereux, fastidieux et monotones [1]. Le contrôle de robot est un domaine de recherche et de développement mature et productif. Par conséquent, un robot utile est celui qui est capable de contrôler son mouvement, les forces et les couples interactifs entre le robot et son environnement [2]. Le domaine de la commande des robots manipulateur est très vaste, il est encore en pleine d'évolution. Plusieurs algorithmes de commande ont été développés pour satisfaire les objectifs exigés. Pour le control d'un robot, nécessite généralement la disponibilité d'un modèle mathématique et d'une sorte d'intelligence pour agir sur le modèle. Ce modèle est obtenu à partir des lois physiques de base régissant son mouvement. En revanche, l'intelligence nécessite des capacités sensorielles et des moyens pour agir et réagir aux variables détectées. Ces actions et les réactions du robot sont le résultat de la conception du contrôleur. Dans ce chapitre, nous passons en revue les concepts de la théorie du contrôle nécessaires utilisés dans le contrôle des robots [2].

De nombreuses techniques et méthodologies de contrôle peuvent être appliquées au contrôle des manipulateurs. La méthode de contrôle particulière choisie ainsi que la manière dont elle est mise en œuvre peuvent avoir un impact significatif sur les performances du manipulateur et par conséquent sur la gamme de ses applications possibles. Par exemple, le suivi de chemin continu nécessite une architecture de contrôle différente de celle du contrôle point à point. De plus, la conception mécanique du manipulateur lui-même influencera le type de schéma de contrôle requis. Par exemple, les problèmes de contrôle rencontrés avec un manipulateur cartésien sont fondamentalement différents de ceux rencontrés avec un manipulateur de type coude. Cela crée un compromis dit logiciel / logiciel entre la structure mécanique du système et l'architecture / la programmation du contrôleur [3].

Il est impossible, de faire une synthèse, même partielle, des techniques de commande des bras manipulateurs, tant ce domaine recouvre de problématiques différentes. Outre les travaux liés à la modélisation et l'identification [4-6], et à la planification de mouvements [7-9], le traitement de la redondance [10-14], des singularités géométriques, la commande dynamique [15, 16], La commande par retour d'efforts [17- 19], et plus généralement la commande référencée capteurs [20, 21], sont autant de domaines de recherche qui mériteraient à eux seuls une synthèse bibliographique. En dehors des quelques références précitées, nous renvoyons le lecteur aux ouvrages et articles de synthèse suivants : [22] pour la modélisation, la commande dynamique et adaptative, [23] pour le cadre méthodologique de synthèse de retours d'état par l'approche par fonctions de tâches, [24] pour un état de l'art sur la modélisation et les techniques de commande des bras manipulateurs, [25, 26] pour une synthèse des techniques de commande de manipulateurs redondants et [27] pour des synthèses de commandes dynamiques.

1.2 Stratégies de contrôle des robots manipulateurs

A partir de la littérature de commande des robots manipulateurs, nous pouvons discriminer les stratégies de contrôle suivantes :

1.2.1 Contrôle linéaire de position

Certains manipulateurs sont équipés de moteurs pas à pas ou d'actionneurs pouvant exécuter directement une trajectoire souhaitée. Cependant, la grande majorité des manipulateurs sont entraînés par des actionneurs qui fournissent une force ou un couple pour provoquer le mouvement des liaisons. Dans ce cas, un algorithme est nécessaire pour calculer les couples qui provoqueront le mouvement. Le problème de la dynamique est au cœur de la conception de tels algorithmes, mais ne constitue pas en soi une solution. Une préoccupation première d'un contrôle de position, le système doit compenser automatiquement les erreurs de connaissance des paramètres d'un système et de supprimer les perturbations qui tendent à perturber la trajectoire souhaitée du système. Pour ce faire, les capteurs de position et de vitesse sont surveillés par l'algorithme de contrôle, qui calcule les couples de commandes pour les actionneurs [29].

1.2.2 Contrôle non linéaire de position

Bien que les systèmes de contrôle basés sur des modèles linéaires approximatifs soient populaires dans les robots industriels, il est important de considérer la dynamique non linéaire complète du manipulateur lors de la synthèse des algorithmes de contrôle. Certains robots industriels qui utilisent des algorithmes de contrôle non linéaires dans leurs contrôleurs sont

récemment en cours d'introduction. Ces techniques non linéaires de contrôle d'un manipulateur promettent une meilleure performance que ne le font des schémas linéaires plus simples [29].

1.2.3 Contrôle de force

La capacité d'un manipulateur à contrôler les forces de contact lorsqu'il touche des pièces, les outils ou les surfaces de travail semblent avoir une grande importance dans l'application des manipulateurs à de nombreuses tâches du monde réel. Le contrôle de la force est complémentaire au contrôle de la position, que nous pensons généralement que l'un ou l'autre est applicable dans une situation donnée. Lorsqu'un manipulateur se déplace dans l'espace libre, seul le contrôle de la position a du sens, parce qu'il n'y a pas de surface contre laquelle réagir. Quand un manipulateur touche une surface rigide, cependant, les systèmes de contrôle de position peuvent entraîner des forces excessives pour construire au contact, ou faire perdre le contact avec la surface au moment souhaité pour certaines applications. Les manipulateurs sont rarement contraints par les surfaces de réaction dans toutes les directions simultanément. Donc, un contrôle mixte ou hybride est nécessaire, avec une certaine direction contrôlée par une loi de contrôle de position et les directions restantes contrôlées par une loi de contrôle de la force [29]. Dans les paragraphes suivants, nous nous limiterons à une description simple des différentes techniques rapportées dans la littérature.

1.3 Commande des bras manipulateurs

D'après [30], le problème de la commande d'un robot manipulateur peut être formulé comme la détermination de l'évolution des forces généralisées (forces ou couples) que les actionneurs doivent exercer pour garantir l'exécution de la tâche tout en satisfaisant certains critères de performance. Différentes techniques sont utilisées pour la commande des bras manipulateurs. La conception mécanique du bras manipulateur a une influence sur le choix du schéma de commande [31]. Un robot manipulateur est une structure mécanique complexe dont les inerties par rapport aux axes des articulations varient non seulement en fonction de la charge mais aussi en fonction de la configuration, des vitesses et des accélérations. Deux types de mouvements apparaissent quand on parle de commande du bras manipulateur. Un premier type considère que les mouvements nécessaires pour la réalisation de la tâche sont exécutés dans l'espace libre. Le deuxième type considère des mouvements spécifiques avec des forces de contact pour l'organe terminal qui se déplace dans un espace contraint. Toute tâche de robotique est réalisée par une combinaison de ces deux types de mouvement. Pour simplifier la commande, les deux types de mouvement sont abordés

séparément. Nous ne pouvons pas, dans le cadre de cette thèse, traiter en détail l'ensemble des techniques de commande des bras manipulateurs. Pour une étude plus détaillée, le lecteur pourra se référer à [22], [30], [32-34]. Nous présentons juste un panorama succinct sur les différentes techniques rapportées dans la littérature. Dans les paragraphes suivants nous nous limiterons à une description simple des différentes techniques [28].

1.3.1 Commande par articulation

Cette technique est utilisée par des robots manipulateurs qui utilisent des servomoteurs avec de forts rapports de réduction. Lorsque le système présente un comportement linéaire, l'asservissement du mouvement peut être réalisé par des techniques de commande classiques. Nous parlons alors d'une commande décentralisée de type PID. Dans [33] et [34], le schéma classique est amélioré avec des signaux d'anticipation pour corriger les effets de la force de gravité et de couplage.

1.3.2 Commande jacobéenne

Cette technique est utilisée depuis les travaux de Whitney [35] et elle est appelée de cette façon puisqu'elle utilise la matrice jacobéenne inverse du bras manipulateur pour calculer les vitesses de consigne aux articulations. Elle est aussi connue sous le nom de commande à mouvement résolu. Les approches les plus courantes sont de type : mouvement à vitesse résolu, mouvement à accélération résolu [16] et mouvement à force résolu [36]. La technique de mouvement résolu commande la position de l'organe terminal du manipulateur dans l'espace cartésien, par combinaison des mouvements de plusieurs articulations. Ce type de schémas peut se présenter sous la forme d'un schéma cinématique, quand les vitesses calculées aux articulations sont utilisées directement comme des consignes pour les boucles de commande de chaque articulation, ou sous la forme d'un schéma dynamique quand le schéma utilise le modèle dynamique pour découpler les articulations. Son principal inconvénient est l'utilisation de l'inverse de la matrice jacobéenne, qui peut devenir singulière sous certaines conditions.

1.3.3 Commande par découplage non linéaire

Cette technique est aussi connue sous les noms de commande dynamique ou commande par couple calculé. Lorsque l'application exige des évolutions rapides avec des contraintes dynamiques, la commande doit prendre en compte les forces d'interaction. Ce type de technique considère l'ensemble des articulations et, pour les découpler, utilise la théorie du

découplage non linéaire. Cette théorie utilise le modèle dynamique du robot pour le calcul de la loi de commande, ce qui conduit à des lois de commande centralisées non linéaires. Des signaux d'anticipation peuvent être utilisés pour minimiser des effets non linéaires [37]. Ce type de technique permet la commande dans l'espace des articulations ou dans l'espace cartésien, avec l'avantage que les articulations sont découplées et peuvent évoluer à grandes vitesses avec de fortes inerties. Cette méthode dépend fortement du modèle du système, elle est très sensible aux imprécisions du modèle qui entraînent un découplage imparfait. Ceci constitue son principal inconvénient.

1.3.4 Commande adaptative

Ce type de techniques vise à corriger les déficiences de la commande par découplage non linéaire, comme la connaissance approximative des paramètres du modèle du robot ou pour s'adapter aux différentes conditions opératoires. Ce type de schémas cherche à estimer ou ajuster en ligne les valeurs des paramètres utilisées dans le calcul de la loi de commande. Un travail plus intéressants sur ce sujet est la commande proposé par [38] appelée commande de Slotine-Li ou commande adaptative passive. Plusieurs travaux sur la commande adaptative sont présentés dans [39]. Les avantages de ce type de techniques sont évidents, malheureusement la puissance de calcul demandée au système constitue un inconvénient important.

1.3.5 Commande fondée sur une fonction de Lyapunov

Des méthodes basées sur une fonction de Lyapunov ont été utilisées pour la commande des bras manipulateurs de façon satisfaisante pour des tâches de suivi. Particulièrement lorsqu'on cherche à garantir la convergence asymptotique et non à linéariser le système ou à obtenir le découplage [40].

1.3.6 Commande passive

Cette technique considère le robot comme un système passif, c'est à dire un système qui dissipe de l'énergie. De telles lois de commande permettent de modifier l'énergie naturelle du robot pour qu'il réalise la tâche. En utilisant le formalisme de Hamilton, la commande cherche à minimiser l'énergie du système en utilisant un bloc non linéaire passif dans la boucle de retour. La commande passive tend à être plus robuste que le découplage non

linéaire, lorsque la technique ne recherche pas à annuler les non linéarités [30].

1.3.7 Commande prédictive

Ce type de commande, en utilisant le modèle du système et les consignes, est capable de prédire son évolution, de telle manière qu'il est capable d'agir en fonction de l'erreur de prédiction. Trois schémas différents sont proposés dans [41] : point final fixe, horizon fini et une combinaison des deux précédents. Un grand avantage de ce type d'approche est lié au fait que l'erreur de prédiction n'est pas contaminée par les bruits de mesure mais la dépendance au modèle reste forte.

1.3.8 Commande robuste

Dans le cas de paramètres fixes, il est connu que la technique de découplage non linéaire peut devenir instable en présence d'incertitudes. Si les paramètres du modèle ne sont pas connus de façon précise et si l'incertitude sur les paramètres admet des bornes connues, alors les techniques de commande robuste peuvent être utilisées. Par exemple, les travaux de Slotine [42] considèrent la technique de mode glissant appelé aussi commande de structure variable. Cette technique utilise une surface de glissement où la stabilité du système est garantie.

1.3.9 Commande optimale

Pour réaliser une tâche, il peut exister un grand nombre de solutions. Dans ce cas, il peut être souhaitable de choisir une solution qui satisfasse un certain critère. La littérature présente différents types de critères pour la commande optimale : la commande en temps minimal [43-45], du domaine des neurosciences la minimisation du jerk pour maximiser la souplesse du mouvement [46], modélisation et commande des robots manipulateurs par les outils de l'intelligence artificielle [47-49] et la minimisation du couple [50, 51], entre autres. La complexité du problème de commande optimale a motivé les chercheurs pour diviser la tâche en deux étapes : la première étape est la planification de trajectoire suivi d'une étape d'asservissement de la trajectoire [52-54].

1.4 Conclusion

La commande robuste des systèmes non linéaires, notamment les robots manipulateurs, est un sujet d'étude d'actualité et les méthodes développées dans la littérature sont nombreuses. Bien que, la plupart des systèmes réels sont essentiellement non linéaires et ces non linéarités

sont, bien souvent, faibles ou ne sont pas visibles sur une plage de fonctionnement de ces procédés. Alors, les méthodes linéaires conventionnelles deviennent impuissantes, incapable de rendre compte de certains phénomènes et aboutissent souvent à des résultats moins performants. Pour surmonter ce problème, la tendance des recherches actuelles ont été effectuées dans le domaine des commandes des systèmes non linéaires robustes qui donnent des résultats acceptables dans des larges domaines de fonctionnement. Il faut donc faire appel à des régulateurs plus sophistiqués qui soient à la fois robustes et nécessitant que peu de calculs, étant donné la rapidité du système. C'était l'objectif de ce chapitre qui représente, de manière non-exhaustive, quelques techniques de commande pour des procédés industriels.

Références du chapitre 1 :

1. **Khalil HK:** *Nonlinear Systems*. 3^{ème} édition. 2002.
2. **Munro N, Lewis F:** *Robot manipulator control theory*. 2^{ème} Edition, par Marcel Dekker, 2004.
3. **Spong MW, Vidyasagar M:** *Robotdynamics and control*. John Wiley & Sons, New York. 1989.
4. **Khalil W :** *Contribution à la commande automatique des manipulateurs avec l'aide d'un modèle mathématique des mécanismes*. Thèse de Doctorat, université de Montpellier 1978.
5. **Gorla B, Renaud M :** *Modèle des robots manipulateurs. Application à leur commande*. Cepadues Toulouse, 1984.
6. **Khalil W, Dombre E :** *Modélisation, identification et commande des robots*. Hermes Science Publications, Paris, 1999.
7. **Brooks R:** *Planning collision-free motions for pick-and-place operations*. International Journal of Robotics Research (IJRR), 2(4).1983.
8. **Faverjon B:** *Obstacle avoidance using an octree in the configuration space of a manipulator*. Dans IEEE Conf. on Robotics and Automation (ICRA), Atlanta, USA, 1984.
9. **Craig J:** *Introduction to robotics: mechanisms and control*. Addison, Wesley publishing company, Reading, 1986.
10. **Liégeois A:** *Automatic supervisory control of the configuration and behavior of multibody mechanisms*. IEEE Transactions Systems Man Cybernetics, 7:868–871, 1977.

11. **Vukobratovic M, Kircanski M:** *A dynamic approach to nominal trajectory synthesis for redundant manipulators.* IEEE Trans. Systems Man Cyber. 14:580–586, 1984.
12. **Bailleul J, Hollerbach J, Brockwett R:** *Programming and control of kinematically redundant manipulators.* Dans IEEE Conf. On Decision and Control (CDC), Pages 768–774, 1984.
13. **Yoshikawa T:** *Manipulability of robotic mechanisms.* International Journal of Robotics Research (IJRR), 4(2):3–9, 1985.
14. **Sciavicco L, Siciliano B:** *A solution algorithm to the inverse kinematic problem for redundant manipulators.* IEEE Journal Robotics and Automation, 4:403–410, 1988.
15. **Khalil W, Liegeois A, Fournier A:** *Commande dynamique des robots.* RAIRO automatique / Systems analysis and control, 13(2) :189–201, 1979.
16. **Luh J, Walker M, Paul R:** *Resolved acceleration control of mechanical manipulators.* IEEE Trans. on Automatic Control (ITAC), 25(3):468–474, 1980.
17. **Withney D:** *Force feedback control of manipulator fine motions.* Journal of Dynamic Systems, Measures and Control, pages 91–97, 1977.
18. **Withney D:** *Historical perspective and state of the art in robot force control.* Dans IEEE Conf. On Robotics and Automation (ICRA), pages 262–268, 1985.
19. **Boissonnat JD, Faverjon B, Merlet :** *Techniques de la robotique. Perception et planification.* J.-P. Hermès, 1988.
20. **Espiau B, Chaumette F, Rives P:** *A new approach to visual servoing in robotics.* IEEE Trans. On Automatic Control (ITAC), 8(3) :313–326, 1992.
21. **Samson C, Ait-Abderrahim K:** *Feedback control of a non-holonomic wheeled cart in cartesian space.* Dans IEEE Conf. On Robotics and Automation (ICRA), pages 1136–1141, 1991.
22. **Dombre E, Khalil W :** *Modélisation et commande des robots.* Hermès, 1988.
23. **Samson C, Ait-Abderrahim K:** *Feedback control of a non-holonomic wheeled cart in cartesian space.* Dans IEEE Conf. On Robotics and Automation (ICRA), pages 1136–1141, 1991.
24. **Canudas de Wit C, Siciliano B, Bastin G:** *Theory of robot control.* Éditeurs Springer Verlag, 1996.
25. **Siciliano, B:** *Kinematic control of redundant robot manipulators: a tutorial.* Journal of Intelligent and Robotic Systems, 3(3): 201–212, 1990.

26. **Yoshikawa T:** *Foundations of robotics: Analysis and control*. MIT Press, 1990.
27. **Asada H, Slotine J:** *Robot analysis and control*. Wiley, New York. 1986.
28. **Makhloufi F:** *Modélisation et commande des robots manipulateurs par les outils de l'intelligence artificielle. Thèse de doctorat, université de Annaba, 2015.*
29. **John J. Graic:** *Introduction to robotic*. 3^{ème} Edition. Pearson Education International 2005.
30. **Sciavicco L, Siciliano B:** *Modelling and Control of Robot Manipulators*. Springer, 2000.
31. **Herrera-Aguilar I :** *Commande des bras manipulateurs et retour visuel pour des applications à la robotique de service*. Ecole doctorale systèmes, version1-18 Dec 2007.
32. **Paul RP:** *Manipulator cartesian path control*. IEEE Trans. Syst., Man Cybern. 9 :702–711, 1979.
33. **Gorla B, Renaud M :** *Modèle des robots manipulateurs. Application à leur commande*. Cepadues Toulouse, 1984.
34. **Canudas C, Siciliano B, Bastin G:** *Theory of Robot Control*. Springer, 1997.
35. **Whitney DE:** *Resolved motion rate control of manipulators and human prostheses*. IEEE Trans, on Man Machine Systems, Vol. MMS-10(2), p. 47-53, juin 1969.
36. **Wu CH, Paul RP:** *Resolved motion force control of robot manipulator*. IEEE Trans. Syst., Man Cybern., pages 266–275, 1982.
37. **Er MJ:** *Recent developments and futuristic trends in robot manipulator control*. Dans Asia-Pacific Workshop on Advances in Motion Control Proceedings, Pages 106–111, 1993.
38. **Slotine JJE, Li W :** *Adaptive strategies in constrained manipulation*. Dans Proc. IEEE Int. Conf. On Robotics and Automation, volume Raleigh, N.C., pages 595–601, 1987.
39. **Lozano R, Taoutaou D :** *Commande adaptative et applications*. Hermes Science Publications, Paris, 2001.
40. **Canudas C, Siciliano B, Bastin G:** *Theory of Robot Control*. Springer, 1997.
41. **Hedjar R, Toumi R, Boucher P, et Dumur D:** *Feedback nonlinear predictive control of rigid link robot manipulators*. Dans Proceedings of the American Control Conference, pages 3594–3599, Anchorage, 2002.
42. **Slotine JJE:** *Robust control of robot manipulators*. The International Journal of Robotics Research, 4:49–64, 1985.

43. **Kahn ME, Roth BE**: *The near-minimum-time control of open-loop articulated kinematic chains*. Trans. ASME J. Dynamic, Systems, Meas., Cont., 93:164–172, 1971.
44. **Bobrow JE, Dubowsky S, Gibson JS**: *Time optimal control of robotic manipulators along specified paths*. The Int. J. of Robotics Research, 4 :3–17. 1985.
45. **Chen Y, Desrochers A**: *Minimum-time control laws for robotic manipulators*. Dans Proc. of the 28th Conference on Decision and Control, pages 2494–2499, Tampa, FL. 1989.
46. **Flash T, Hogan N**: *The coordination of arm movements: An experimentally confirmed mathematical model*. The Journal of Neuroscience, 5 :1688–1703, 1985.
47. **Kyriakopoulos KJ, Saridis GN**: *Minimum jerk path generation*. Dans Proc. IEEE International Conference on Robotics and Automation, pages 364–369, 1988.
48. **Seki K, Tadakuma S**: *Minimum jerk control of power assisting robot based on human arm behavior characteristic*. Dans International Conference on Systems, Man and Cybernetics, pages 722–721, 2004.
49. **Amirabdollahian F, Loureiro R, Harwin W**: *Minimum jerk trajectory control for rehabilitation and haptic applications*. Dans IEEE International Conference on Robotics and Automation, pages 3380–3385, May 2002.
50. **Egeland O, Lunde E**: *Trajectory generation for manipulators using linear quadratic optimal tracking*. Dans IEEE, pages 376–381, 1988.
51. **Martin BJ, Bobrow JE**: *Minimum-effort motions for open-chain manipulators with task-dependant end-effector constraints*". The International Journal of Robotics Research, 18:213–224, 1999.
52. **Lin CS, Chang PR, Luh JYS**: *Formulation and optimization of cubic polynomial joint trajectories for industrial robots*. IEEE Transactions on Automatic Control, AC- 28:1066–1074, 1983.
53. **Kim BK, Shin KG**: *Minimum-time path planning for robot arms and their dynamics*. IEEE Trans. on Systems, Man and Cybernetics, SMC-15:213–223, 1985.
54. **Shin KG, McKay ND**: *Minimum-time control of robotic manipulators with geometric path constraints*. IEEE Transactions on Automatic Control, AC- 30:531–541, 1985.

Chapitre 2

Modélisation des robots manipulateurs

Modélisation des robots manipulateurs

2.1 Introduction

Pour développer une stratégie de commande performante pour un robot manipulateur, il est impératif de connaître la cinématique et la dynamique du robot considéré. Et comme les robots manipulateurs sont des systèmes mécaniques poly articulés, fortement non linéaires, d'une dynamique qui peut être mal définie, ils ont des paramètres variables et sont soumis à des perturbations externes. Pour pouvoir faire exécuter une action ou une tâche au robot manipulateur, il est nécessaire de modéliser ce système multi variable pour positionner les différents axes. La modélisation de ces systèmes relève très souvent d'une approximation des phénomènes physiques mis en jeu, c'est à partir de cette représentation approximative que l'on souhaite construire une commande pour le système réel, cette commande doit être alors robuste dans le sens où elle devra assurer une faible sensibilité aux incertitudes des paramètres, à leurs variations et aux perturbations externes [1].

2.2 Conception du manipulateur et capteurs

Bien que les manipulateurs soient, en théorie, des dispositifs universels applicables à de nombreuses situations, les considérations économiques dictent généralement que le domaine de tâche prévu influence la conception mécanique du manipulateur. Avec des problèmes tels que la taille, la vitesse et capacité de charge, le concepteur doit également prendre en compte le nombre d'articulations et leur arrangement géométrique. Ces considérations affectent l'espace de travail du manipulateur en terme de taille et qualité, la rigidité de la structure du manipulateur et d'autres attributs. Plus un bras de robot contient d'articulations, plus il sera capable, Bien sûr il sera également plus difficile à construire et plus coûteux. Afin de construire un robot utile, cela peut prendre deux approches : construire un robot spécialisé pour une tâche, ou construire un robot universel capable d'exécuter une grande variété de tâches [2].

2.3 Modélisation

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

- Les modèles de transformation entre l'espace opérationnel (dans lequel est définie la situation de l'organe terminal) et l'espace articulaire (dans lequel est définie la configuration du robot). On distingue :
 - ✓ Les modèles géométriques direct et inverse qui expriment la situation de l'organe terminal en fonction des variables articulaires du mécanisme et inversement ;
 - ✓ Les modèles cinématiques direct et inverse qui expriment la vitesse de l'organe terminal en fonction des vitesses articulaires et inversement ;
- Les modèles dynamiques définissant les équations du mouvement du robot, qui permettent d'établir les relations entre les couples ou forces exercés par les actionneurs et les positions, vitesses et accélérations des articulations. On présente dans ce chapitre quelques méthodes permettant d'établir ces modèles. On se limitera au cas des robots à structure ouverte simple.

2.3.1 Modélisation géométrique

La modélisation des robots de façon systématique et automatique exige une méthode adéquate pour la description de leur morphologie. Plusieurs méthodes et notations ont été proposées [3- 8]. La plus répandue est celle de Denavit-Hartenberg [3]. Mais cette méthode, développée pour des structures ouvertes simples, présente des ambiguïtés lorsqu'elle est appliquée sur des robots ayant des structures fermées ou arborescentes [9].

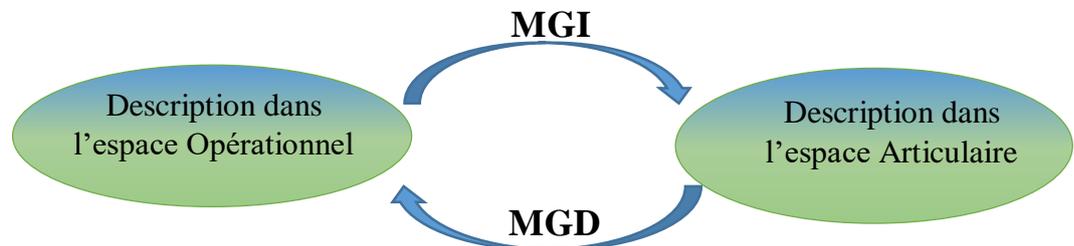


Figure 2.1 : Modèle géométrique directe et inverse.

2.3.1.1 Modèle géométrique direct (MGD) : Permet d'exprimer la situation de l'organe terminal, c'est-à-dire les coordonnées opérationnelles du robot, en fonction de ses coordonnées articulaires [9]. Le modèle géométrique direct est unique et est exprimé sous forme d'équations explicites.

$$X = f(q) \quad (2.1)$$

Où $\begin{cases} q = [q_1, q_2, \dots, q_n] \in R^n & \text{: vecteurs des variables articulaires.} \\ X = [x_1, x_2, \dots, x_n] \in R^n & \text{: vecteurs des variables opérationnelles.} \end{cases}$

Il existe aussi la notation de Khalil et Kleinfinger qui permet la description homogène, et avec un nombre minimum de paramètres, des architectures ouvertes simple et complexe [6].

- **Calcul du modèle géométrique direct :** Dans ce qui suit, on élaborera le modèle d'un robot planaire (RR) à deux degrés de liberté (2ddl), représenté par la figure 2.2, à travers la mise en forme des équations mathématiques régissant le mouvement de ce bras.

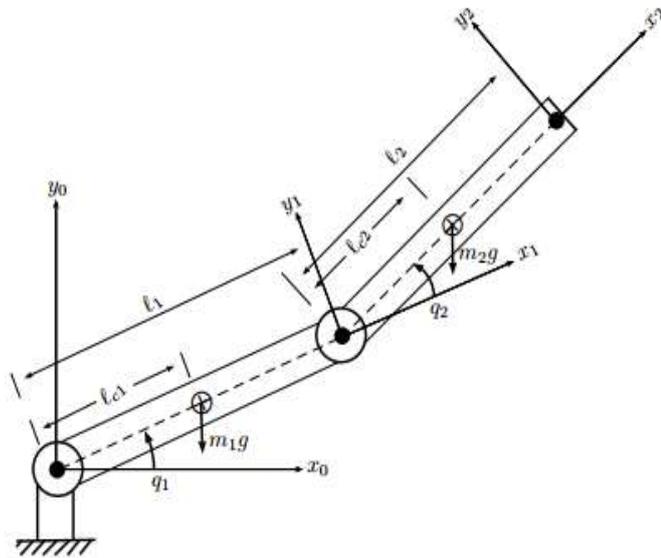


Figure 2.2: Les paramètres géométrique d'un Robot planaire à 2 ddl.

Avec :

l_i : Longueur du segment i .

l_{ci} : Longueur du centre de gravité du segment i .

La configuration de Denavit-Hartenberg (Annexe) du robot à 2ddl est donnée par le tableau 2.1 :

Tableau 2.1 : Configuration de Denavit-Hartenberg du robot à 2ddl.

Articulation	Paramètres				
	σ_i	d_i	α_i	θ_i	r_i
1	0	l_1	0	q_1	0
2	0	l_2	0	q_2	0

Les matrices de transformation homogène des deux segments sont données par :

$${}^0T_1 = \begin{bmatrix} C_1 & -S_1 & 0 & l_1 C_1 \\ S_1 & C_1 & 0 & l_1 S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

$${}^1T_2 = \begin{bmatrix} C_2 & -S_2 & 0 & l_2 C_2 \\ S_2 & C_2 & 0 & l_2 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

La matrice de transformations finale 0T_2 qui sert à déterminer le modèle géométrique direct, après multiplication des matrices élémentaires, est la suivante :

$${}^0T_2 = \begin{bmatrix} C_{12} & -S_{12} & 0 & l_2 C_{12} + l_1 C_1 \\ S_{12} & C_{12} & 0 & l_2 S_{12} + l_1 S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

Avec

$$\begin{aligned} \cos(q_i) &= C_i, \sin(q_i) = S_i \\ \cos(q_i + q_j) &= C_{ij}, \sin(q_i + q_j) = S_{ij} \end{aligned}$$

$${}^0T_2 = \begin{bmatrix} x_x & y_x & z_x & p_x \\ x_y & y_y & z_y & p_y \\ x_z & y_z & z_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

A partir des équations (2.4) et (2.5) nous obtenons les coordonnées cartésiennes représentant le modèle géométrique direct du robot :

$$\begin{cases} p_x = l_2 C_{12} + l_1 C_1 \\ p_y = l_2 S_{12} + l_1 S_1 \\ p_z = 0 \end{cases} \quad (2.6)$$

2.3.1.2 Modèle géométrique indirect (MGI)

Le Modèle géométrique inverse consiste à calculer les coordonnées articulaires correspondant à une situation donnée de l'organe terminal. Le modèle s'écrit :

$$q = f^{-1}(X) \quad (2.7)$$

On peut distinguer trois méthodes de calcul du MGI :

- La méthode de Paul [10] qui traite séparément chaque cas particulier et convient pour la plupart des robots industriels.
- La méthode de Pieper [11] qui permet de résoudre le problème pour les robots à six degrés de liberté possédant trois articulations rotoïdes d'axes concourants ou trois articulations prismatiques.
- La méthode générale de Raghavan et Roth [12], donnant la solution générale des Robots à six articulations à partir d'un polynôme de degré au plus égal à 16.

➤ Calcul du modèle géométrique indirect :

- ⊗ Calcul de q_2 : D'après les équations déduites de l'analyse cinématique directe basée sur la géométrie [makhloufi,] nous obtenons :

$$p_x^2 + p_y^2 = l_2^2 C_{12}^2 + l_1^2 C_1^2 + 2l_2 C_{12} l_1 C_1 + l_2^2 S_{12}^2 + l_1^2 S_1^2 + 2l_1 l_2 S_{12} S_1 \quad (2.8)$$

$$= l_1^2 + l_2^2 + 2l_1 l_2 (C_1 C_{12} + S_1 S_{12}) \quad (2.9)$$

Alors $(C_1 C_{12} + S_1 S_{12}) = \frac{p_x^2 + p_y^2 - l_1^2 - l_2^2}{2l_1 l_2} \quad (2.10)$

En utilisant les relations trigonométriques suivantes :

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

Et en posant :

$$D = \frac{p_x^2 + p_y^2 - l_1^2 - l_2^2}{2l_1 l_2} \quad \text{nous obtenons : } C_2 = D \text{ et } S_2 = \pm \sqrt{1 - D^2}$$

Posant $\varepsilon = \mp 1$.

Ce qui implique :

$$q_2 = ATAN2(\varepsilon\sqrt{1 - D^2}, D) \quad (2.11)$$

La fonction $ATAN2(\sin\alpha, \cos\alpha)$ est presque similaire au $Arctg$ [6], elle donne l'angle α en fonction des signes des sinus et cosinus.

✎ Calcul de q_1 :

En procédant de la même manière que pour q_2 , nous pouvons encore écrire en développant C_{12} , S_{12} :

$$\begin{cases} p_x = l_1 C_1 + l_2 (C_1 C_2 - S_1 S_2) \\ p_y = l_1 S_1 + l_2 (S_1 C_2 + S_2 C_1) \end{cases} \quad (2.12)$$

En regroupant suivant les inconnues C_1 , S_1 nous trouvons :

$$\begin{cases} p_x = (l_1 + l_2 C_2) C_1 - l_2 S_2 S_1 \\ p_y = l_2 S_2 C_1 + (l_1 + l_2 C_2) S_1 \end{cases} \quad (2.13)$$

On résout le système (2.12) par rapport aux deux inconnues C_1 et S_1 , le déterminant vaut $p_x^2 + p_y^2$ sont donnés par:

$$C_1 = \frac{p_x(l_1 + l_2 C_2) + p_y l_2 S_2}{p_x^2 + p_y^2}$$

$$S_1 = \frac{p_y(l_1 + l_2 C_2) + p_x l_2 S_2}{p_x^2 + p_y^2}$$

D'où

$$q_1 = ATAN2(S_1, C_1) \quad (2.14)$$

2.3.2 Modélisation cinématique

2.3.2.1 Modèle cinématique direct

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

$$\dot{X} = J_a \dot{q} \quad (2.15)$$

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

La même matrice jacobienne intervient dans le calcul du modèle différentiel direct qui donne les variations élémentaires dX des coordonnées opérationnelles en fonction des variations élémentaires des coordonnées articulaires dq , soit :

$$dX = J_a(q).dq \quad (2.16)$$

L'intérêt de la matrice jacobienne est multiple :

- ✓ Elle est à la base du modèle différentiel inverse, permettant de calculer une solution locale des variables articulaires q connaissant les coordonnées opérationnelles X .
- ✓ En statique, on utilise le jacobien pour établir la relation liant les efforts exercés par l'organe terminal sur l'environnement aux forces et couples des actionneurs.
- ✓ Elle facilite le calcul des singularités et de la dimension de l'espace opérationnel accessible du robot [6].

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

$$J_{a_{ij}} = \frac{\partial f_i(q)}{\partial q_j} \quad i = 1, \dots, m; j = 1, \dots, n \quad (2.17)$$

Où $J_{a_{ij}}$ est l'élément (i, j) de la matrice jacobienne J_a .

Cette méthode est facile à mettre en œuvre pour des robots à deux ou trois degrés de liberté.

2.3.2.2 *Modèle cinématique inverse*

L'objectif du modèle cinématique inverse est de calculer, à partir d'une configuration q donnée, les vitesses articulaires \dot{q} , qui assurent au repère terminal une vitesse opérationnelle \dot{x} imposée. Cette définition est analogue à celle du modèle différentiel inverse : ce dernier permet de déterminer la différentielle articulaire $d\dot{q}$ correspondant à une différentielle des coordonnées opérationnelles $d\dot{x}$ spécifiée. Pour obtenir le modèle cinématique inverse, on inverse le modèle cinématique direct en résolvant un système d'équations linéaires [6]. La mise en œuvre peut être faite de façon analytique ou numérique :

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

Pour beaucoup de détails sur le calcul du modèle cinématique inverse, le lecteur peut se référer à [9].

2.3.3 Modélisation Dynamique

Pour la conception des contrôles, il est nécessaire de disposer d'un modèle mathématique cela révèle le comportement dynamique d'un système. Par conséquent, dans cette section, nous dérivons les équations dynamiques du mouvement pour un robot manipulateur. L'approche consiste à dériver l'énergie cinétique et potentielle du manipulateur et ensuite utilise les équations du mouvement de Lagrange [13].

2.3.3.1 Modèle dynamique inverse

Le modèle dynamique est la relation entre les couples (et/ou forces) appliqués aux actionneurs et les positions, vitesses et accélérations articulaires. On représente le modèle dynamique inverse, ou tout simplement le modèle dynamique, par une relation de la forme [6] :

$$\Gamma = f(q, \dot{q}, \ddot{q}, f_e) \quad (2.18)$$

Avec

- Γ : vecteur des couples/forces des actionneurs, selon que l'articulation est rotoïde ou prismatique. Dans la suite, on écrira tout simplement *couples*.
- q : vecteur des positions articulaires.
- \dot{q} : vecteur des vitesses articulaires.
- \ddot{q} : vecteur des accélérations articulaires.
- f_e : vecteur représentant l'effort extérieur (forces et moments) qu'exerce le robot sur l'environnement.

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

- ✎ Le formalisme de Lagrange : La méthode n'est pas celle qui donne le modèle le plus performant du point de vue du nombre d'opérations, mais c'est la méthode la plus simple compte tenu de ces objectifs.

C'est la méthode utilisée ultérieurement pour la modélisation du robot manipulateur à 2 ddl.

- ✎ Le formalisme de Newton-Euler : Cette méthode permet d'obtenir directement le modèle dynamique inverse sans avoir à calculer explicitement les matrices A , C et Q .

i. Formalisme de Lagrange

Le formalisme de Lagrange décrit les équations du mouvement d'un robot manipulateur constitué de n degrés de liberté et de n articulations par l'équation suivante :

$$\Gamma_i = \frac{d}{dt} \left(\frac{\partial l}{\partial \dot{q}_i} \right) - \frac{\partial l}{\partial q_i} \quad (2.19)$$

Avec $i = 1, 2, \dots, n$.

L : Fonction lagrangienne de robot égale à : $L = E_c - E_p$.

E_c : Energie cinétique totale du robot manipulateur.

E_p : Energie potentielle totale du robot manipulateur.

Nous passons en revue certains concepts de base de la physique qui nous permettront de mieux comprendre la dynamique des armes.

La force centripète F_{cent} d'une masse tournant autour d'un point de rayon r et la vitesse angulaire w est donnée par [13] :

$$F_{cent} = \frac{m v^2}{r} = m w^2 r = m \dot{q}^2 r \quad (2.20)$$

La vitesse linéaire est donnée par :

$$v = w \times r \quad (2.21)$$

L'énergie cinétique d'une masse se déplaçant avec une vitesse linéaire v est :

$$k = \frac{1}{2} m v^2 \quad (2.22)$$

L'énergie cinétique de rotation de la masse m est donnée par :

$$k_{rot} = \frac{1}{2} I w^2 \quad (2.23)$$

Où le moment d'inertie est

$$I = \int_{vol} \rho(r) r^2 dr \quad (2.24)$$

Avec $\rho(r)$ la distribution de masse au rayon r dans un volume. Dans le cas simple où m est une masse ponctuelle, cela devient :

$$I = m r^2 \quad (2.25)$$

Donc

$$k_{rot} = \frac{1}{2} m r^2 \dot{q}^2 \quad (2.26)$$

L'énergie potentielle d'une masse m de hauteur h dans un champ gravitationnel avec la constante g est donné par :

$$I = mgh \quad (2.27)$$

ii. Calcul de l'énergie cinétique et potentielle d'un bras manipulateur à 2ddl

Dans notre travail, nous avons considéré un robot manipulateur à 2ddl, dont le mouvement des articulations est rotoïde. Le variable articulation est $q = [q_1 \ q_2]^T$, le vecteur de force généralisé $\tau = [\tau_1 \ \tau_2]^T$ avec τ_1 et τ_2 sont les torques fournis par les actionneurs.

Pour le premier segment, l'énergie cinétique et potentielle sont :

$$K_1 = \frac{1}{2} m_1 l_1^2 \dot{q}_1^2 \quad (2.28)$$

$$P_1 = m_1 g l_1 \sin q_1 \quad (2.29)$$

Pour le deuxième segment, on a :

$$x_2 = l_1 \cos q_1 + l_2 \cos(q_1 + q_2) \quad (2.30)$$

$$y_2 = l_1 \sin q_1 + l_2 \sin(q_1 + q_2) \quad (2.31)$$

$$\dot{x}_2 = -l_1 \dot{q}_1 \sin q_1 - l_2 (\dot{q}_1 + \dot{q}_2) \sin(q_1 + q_2) \quad (2.32)$$

$$\dot{y}_2 = l_1 \dot{q}_1 \cos q_1 + l_2 (\dot{q}_1 + \dot{q}_2) \cos(q_1 + q_2) \quad (2.33)$$

De sorte que la vitesse au carré est :

$$v_2^2 = \dot{x}_2^2 + \dot{y}_2^2 = l_1^2 \dot{q}_1^2 + l_2^2 (\dot{q}_1 + \dot{q}_2)^2 + 2l_1 l_2 (\dot{q}_1^2 + \dot{q}_1 \dot{q}_2) \cos q_2 \quad (2.34)$$

Par conséquent, l'énergie cinétique du 2^{ème} segment est :

$$K_2 = \frac{1}{2} m_2 v_2^2 = \frac{1}{2} m_2 l_1^2 \dot{q}_1^2 + \frac{1}{2} m_2 l_2^2 (\dot{q}_1 + \dot{q}_2)^2 + m_2 l_1 l_2 (\dot{q}_1^2 + \dot{q}_1 \dot{q}_2) \cos q_2 \quad (2.35)$$

L'énergie potentielle pour le 2^{ème} segment est :

$$P_2 = m_2 g y_2 = m_2 g [l_1 \sin q_1 + l_2 \sin(q_1 + q_2)] \quad (2.36)$$

Le lagrangien pour tout le bras est :

$$\begin{aligned} L &= K - P = K_1 + K_2 - P_1 - P_2 \\ &= \frac{1}{2} (m_1 + m_2) l_1^2 \dot{q}_1^2 + \frac{1}{2} m_2 l_2^2 (\dot{q}_1 + \dot{q}_2)^2 + m_2 l_1 l_2 (\dot{q}_1^2 + \dot{q}_1 \dot{q}_2) \cos q_2 - (m_1 + \\ & m_2) g l_1 \sin q_1 - m_2 g l_2 \sin(q_1 + q_2) \end{aligned} \quad (2.37)$$

Alors

$$\frac{\partial L}{\partial \dot{q}_1} = (m_1 + m_2)l_1^2 \dot{q}_1 + m_2 l_2^2 (\dot{q}_1 + \dot{q}_2) + m_2 l_1 a_2 (2\dot{q}_1 + \dot{q}_2) \cos q_2 \quad (2.38)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_1} = (m_1 + m_2)l_1^2 \ddot{q}_1 + m_2 l_2^2 (\ddot{q}_1 + \ddot{q}_2) + m_2 l_1 l_2 (2\ddot{q}_1 + \ddot{q}_2) \cos q_2 - m_2 l_1 l_2 (2\dot{q}_1 \dot{q}_2 + \dot{q}_2^2) \sin q_2 \quad (2.39)$$

$$\frac{\partial L}{\partial q_1} = -(m_1 + m_2)gl_1 \cos q_1 - m_2 gl_2 \cos(q_1 + q_2) \quad (2.40)$$

$$\frac{\partial L}{\partial \dot{q}_2} = m_2 l_2^2 (\dot{q}_1 + \dot{q}_2) + m_2 l_1 l_2 \dot{q}_1 \cos q_2 \quad (2.41)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_2} = m_2 l_2^2 (\ddot{q}_1 + \ddot{q}_2) + m_2 l_1 l_2 \ddot{q}_1 \cos q_2 - m_2 l_1 l_2 \dot{q}_1 \dot{q}_2 \sin q_2 \quad (2.42)$$

$$\frac{\partial L}{\partial q_2} = -m_2 l_1 l_2 (\dot{q}_1^2 + \dot{q}_1 \dot{q}_2) \sin q_2 - m_2 gl_2 \cos(q_1 + q_2) \quad (2.43)$$

Finalement, selon l'équation de Lagrange, la dynamique du bras manipulateur est donnée par les deux équations différentielles non linéaires couplées suivantes :

$$\tau_1 = [(m_1 + m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos q_2] \ddot{q}_1 + [m_2 l_2^2 + m_2 l_1 l_2 \cos q_2] \ddot{q}_2 - m_2 l_1 l_2 (2\dot{q}_1 \dot{q}_2 + \dot{q}_2^2) \sin q_2 + (m_1 + m_2)gl_1 \cos q_1 + m_2 gl_2 \cos(q_1 + q_2) \quad (2.44)$$

$$\tau_2 = [m_2 l_2^2 + m_2 l_1 l_2 \cos q_2] \ddot{q}_1 + m_2 l_2^2 \ddot{q}_2 + m_2 l_1 l_2 \dot{q}_1^2 \sin q_2 + m_2 gl_2 \cos(q_1 + q_2) \quad (2.45)$$

La dynamique des bras en forme vectoriel est donné par :

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = M(q) \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} -m_2 l_1 l_2 (2\dot{q}_1 \dot{q}_2 + \dot{q}_2^2) \sin q_2 \\ m_2 l_1 l_2 \dot{q}_1^2 \sin q_2 \end{bmatrix} + \begin{bmatrix} (m_1 + m_2)gl_1 \cos q_1 + m_2 gl_2 \cos(q_1 + q_2) \\ m_2 gl_2 \cos(q_1 + q_2) \end{bmatrix} \quad (2.46)$$

Où

$$M(q) = \begin{bmatrix} (m_1 + m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos q_2 & m_2 l_2^2 + m_2 l_1 l_2 \cos q_2 \\ m_2 l_2^2 + m_2 l_1 l_2 \cos q_2 & m_2 l_2^2 \end{bmatrix}$$

Alors la forme standard de la dynamique du manipulateur est :

$$M(q)\ddot{q} + H(q, \dot{q}) + G(q) = \tau \quad (2.47)$$

Avec $M(q)$ la matrice d'inertie, $H(q, \dot{q})$ le vecteur de Coriolis / centrifuge et $G(q)$ le vecteur de gravité. Notez que $M(q)$ est symétrique.

En général, un bras du robot manipulateur est toujours affecté de frottement et de perturbation. Par conséquent, nous devons généraliser le modèle du bras exprimé par l'équation (2.47) en écrivant le modèle dynamique du manipulateur ainsi [14] :

$$\Gamma = M(q)\ddot{q} + H(q, \dot{q}) + G(q) + F(\dot{q}) + \Gamma_d \quad (2.48)$$

2.3.3.2 Modèle dynamique direct

Le modèle dynamique direct est celui qui exprime les accélérations articulaires en fonction des positions, vitesses et couples des articulations. Pour simuler le comportement du robot et de sa boucle de commande, on utilise le modèle dynamique direct. Il est obtenu par inversion du modèle précédent, il s'écrit :

$$\ddot{q} = M^{-1}(q)(\Gamma - H(q, \dot{q}) - G(q) - F(\dot{q}) - \Gamma_d) \quad (2.50)$$

Parmi les applications du modèle dynamique, aussi bien que la commande, 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 [6].

2.4 Conclusion

Dans ce chapitre, nous avons présenté la description des différents modèles utilisés pour décrire les mouvements des articulations d'un bras manipulateur. Ceci est réalisé en fonction du modèle géométrique, cinématique et dynamique, dont le but est de fournir les équations mathématiques qui permettent de programmer, commander et exécuter les tâches désirées. Dans ce qui suit, la théorie de commande du robot manipulateur sera discutée et appliquée. Pour la construction des différents contrôleurs, les algorithmes d'optimisation utilisés sont décrits dans le chapitre suivant.

Références du chapitre 2 :

1. **Chekakta Z** : *Modélisation et commande d'un robot industriel de type brasmanipulateur à 2ddl*. Mémoire de master en Electronique, École nationale Polytechnique d'Oran département de Génie Electrique, 2016.
2. **John J Graic**: *Introduction to robotic*. 3^{ème} Edition. Pearson Education International 2005.
3. **Denavit J, Hartenberg RS**: *A kinematic notation for lower pair mechanism based on matrices*, Trans. Of ASME, J. of Applied Mechanics, Vol. 22, p. 215-221, juin 1955.
4. **Sheth PN, Uicker JJ**: *A generalized symbolic notation for mechanism*, Trans. of ASME, J. of Engineering for Industry, Vol. 93, 1971, p. 102-112.
5. **Renaud M** : *Contribution à l'étude de la modélisation et de la commande des systèmes mécaniques articulés*. Thèse de Docteur-Ingénieur, UPS, Toulouse, déc. 1975.
6. **Khalil W** : *Modélisation et commande par ordinateur du manipulateur MA-23 ; extension à la conception par ordinateur des manipulateurs*. Thèse de Docteur-Ingénieur, USTL, Montpellier, sept. 1976.
7. **Borrel P** : *Modèle de comportement de manipulateurs ; application à l'analyse de leurs performances et à leur commande automatique*. Thèse de Troisième Cycle, USTL, Montpellier, déc. 1979.
8. **Craig JJ**: *Introduction to robotics: mechanics and control*. Addison Wesley Publishing Company, Reading, 1986.
9. **Khalil W, Dombre E** : *Modélisation, identification et commande des robots*. Hermes Science Publications, Paris, 1999.
10. **Paul RCP**: *Robot manipulators: mathematics, programming and control*. MIT Press, Cambridge, 1981.
11. **Pieper DL**: *The kinematics of manipulators under computer control*. Ph. D. Thesis, Stanford University, 1968.
12. **Raghavan M, Roth B**: *Inverse kinematics of the general 6R manipulator and related linkages*. Trans. of the ASME, J. of Mechanical Design, Vol. 115, p. 502-508, 1990.
13. **Munro N, Lewis F**: *Robot manipulator control theory and practice*. 2^{ème} Edition, par Marcel Dekker, 2004.
14. **Makhloufi F** : *Modélisation et commande des robots manipulateurs par les outils de l'intelligence artificielle*. Thèse de doctorat, université de Annaba, 2015.

Chapitre 3

Les algorithmes d'optimisation

Les algorithmes d'optimisation

3.1 Introduction

Les algorithmes méta-heuristiques (AM) sont devenus courants au cours des dernières décennies dans de nombreuses disciplines appliquées, en raison de la haute performance, de la faible capacité de calcul et du temps requis que les algorithmes déterministes dans divers problèmes d'optimisation. Des concepts simples sont nécessaires pour obtenir des résultats favorables et il est facile de transplanter dans différentes disciplines. De plus, le manque du caractère aléatoire à la dernière étape d'un algorithme déterministe le rend enclin à sombrer dans l'optimum local, les facteurs aléatoires dans les algorithmes méta-heuristiques peut faire l'algorithme recherche toutes les solutions optimales dans l'espace de recherche, évitant ainsi efficacement l'optimum local. Dans les problèmes linéaires, un certain algorithme de la descente du gradient tel que [1] sont plus efficaces que les algorithmes stochastiques pour l'utilisation de l'information du gradient. La vitesse de convergence des AM sera inférieure aux algorithmes de descente du gradient et peut être considéré comme un inconvénient. Dans les problèmes non linéaires, les AM commencent généralement le processus d'optimisation avec des solutions générées aléatoirement et ne demandent pas d'informations sur le gradient, ce qui rend l'algorithme parfaitement adapté aux problèmes pratiques lorsque l'information de la dérivée est inconnue.

Dans les scénarios du monde réel, l'espace de résolution de nombreux problèmes est souvent indéterminé ou infini. Il peut être impossible de trouver des solutions optimales en traversant l'espace de solution dans les circonstances actuelles. Les AM détectent la solution optimale immédiate du problème en échantillonnant au hasard un espace large de solution d'une certaine manière, pour trouver ou générer de meilleures solutions pour le problème d'optimisation, dans des circonstances ou une capacité de calcul limitées. Les AM sont généralement inspirés par des phénomènes du monde réel pour trouver de meilleures solutions heuristiques pour des problèmes d'optimisation en simulant des règles physiques ou des phénomènes biologiques.

Les AM peuvent être divisés en deux catégories principales : les méthodes basées sur la natation et les techniques évolutives [2]. Le premier type simule principalement des phénomènes physiques, applique des règles ou des méthodes mathématiques, y compris : Optimiseur Multi-Verse (MVO), Algorithme de recherche locale gravitationnelle (GLSA), recherche du système chargé (CSS), Algorithme de recherche gravitationnelle (GSA), Algorithme à sinus cosinus (SCA), Recuit Simulé (SA), Optimisation basée sur l'enseignement et l'apprentissage (TLBO), Optimisation de force centrale (CFO) et recherche de Tabu (TS).

Les méthodes inspirées de la nature comprennent principalement deux types : méthodes évolutives et techniques d'essaim intelligentes. L'inspiration d'algorithme évolutionniste (EA) provient du processus d'évolution biologique dans la nature. Par rapport au algorithme d'optimisation traditionnel, c'est une méthode d'optimisation globale avec une meilleure robustesse et applicabilité. Certains des algorithmes répandus dans la classe d'EA sont l'algorithme génétique (GA), Programmation Génétique (GP), Stratégie d'évolution (ES) Programmation évolutive (EP) et évolution différentielle (DE). L'application des ES et EP dans la recherche scientifique et les problèmes pratique deviennent également de plus en plus importants. L'intelligence en essaim (SI) comprend une intelligence collective ou sociale qui simule artificiellement la décentralisation des clusters biologiques dans la nature ou le comportement collectif des systèmes auto-organisés. Dans cette classe d'algorithmes, l'inspiration vient généralement de groupes biologiques dans la nature qui ont un comportement collectif et intelligent pour atteindre un certain but.

En général, les algorithmes d'intelligence en essaim (SI) sont plus avantageux que les algorithmes évolutifs parce que les algorithmes SI sont plus accessibles à l'appareil que les algorithmes évolutifs avec moins d'opérateurs à contrôler. De plus, l'algorithme SI a une plus forte capacité d'enregistrer et d'utiliser des informations historiques que EA. Les algorithmes récents établis dans cette classe sont : l'optimisation des essaims de particules (PSO), l'optimisation des essaims de guêpes (WSO), Algorithme inspiré des chauves-souris (BA), Optimisation du loup gris (GWO), Optimisation du baleine (WOA), Optimisation de la mouche des fruits (FOA), Optimisation de la flamme des mites (MFO), Optimisation de la colonie de fourmis (ACO), Optimiseur du faucon de Harris (HHO) et les colonies d'abeilles artificielles (ABC). Une conception schématique pour la classification des méthodes évolutives et SI sont illustrées à la figure 3.1. Bien que différentes MA aient une certaine distinction, elles ont toutes deux étapes identiques dans la gradation de recherche : exploration et exploitation. La phase d'exploration se réfère au processus de recherche l'espace de solution aussi largement, au hasard et globalement que possible [2].

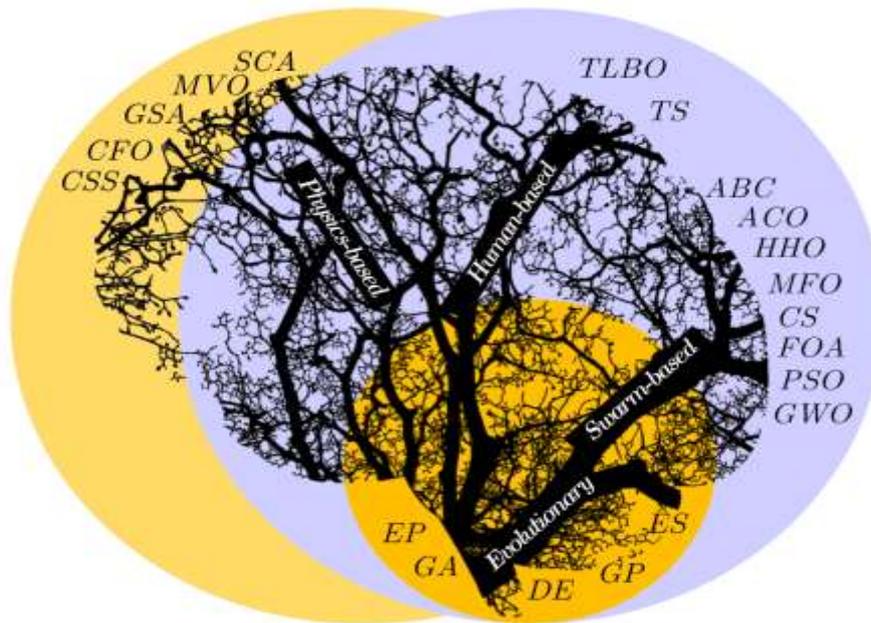


Figure 3.1: Classification des méthodes évolutionnaires et SI.

La phase d'exploitation se réfère à la compétence de l'algorithme pour rechercher plus précisément dans la zone acquise par la phase d'exploration, et son caractère aléatoire diminue tandis que sa précision augmente. Lorsque la capacité d'exploration de l'algorithme est dominante, il peut rechercher l'espace de solution de manière plus aléatoire et produire plusieurs solutions différenciées pour converger flottamment. Lorsque la capacité d'exploitation de l'algorithme est dominante, il recherche plus localement pour améliorer la qualité et la précision des ensembles de solutions. Cependant, lorsque la phase d'exploration sera améliorée, elle entraînera une réduction de la capacité d'exploitation, et vice versa. Un autre défi est que l'équilibre de ces deux capacités n'est pas nécessairement identique pour différents problèmes. Par conséquent, il est relativement difficile de parvenir à un équilibre approprié, entre les deux phases, efficace pour tous les problèmes d'optimisation.

Malgré le succès des AM conventionnels et récents, aucune d'entre eux ne peut garantir un optimum global pour tous les problèmes d'optimisation. Cela a été prouvé logiquement par la théorie de No-Free-Lunch (NFL) [3]. Ce théorème a motivé de nombreux chercheurs à concevoir un nouvel algorithme et résoudre plus efficacement de nouvelles classes de problèmes [2]. Dans ce chapitre, on expose quelques algorithmes utilisés, dans le cadre de cette étude, pour l'optimisation des contrôleurs du robot manipulateur à 2 ddl par différentes techniques de commandes présentés à partir du chapitre suivant.

3.2 Les algorithmes d'optimisation

3.2.1 Optimiseur d'essaim de particules (PSO)

PSO est une méthode de calcul qui optimise un problème en essayant, de manière itérative, d'améliorer une solution candidate par rapport à une mesure de qualité donnée [4]. Il résout un problème en ayant une population de solutions candidates et en déplaçant ces particules dans l'espace de recherche selon des formules mathématiques simples de la position et la vitesse des particules. Mais il est également guidé vers les positions les plus connues dans l'espace de recherche qui sont mises à jour en tant que meilleures positions trouvées par d'autres particules. Cela devrait faire avancer l'essaim vers les meilleures solutions en utilisant la meilleure position individuelle (pbest) et la meilleure position globale (gbest) exprimées par :

$$V_i(t) = \lambda(w.V_i(t-1) + c1.rand().(pbest_i - X_i(t-1)) + c2.rand().(gbest_i - X_i(t-1))) \quad (3.1)$$

$$X_i(t) = X_i(t-1) + V_i(t) \quad (3.2)$$

Où c_1 et c_2 sont deux constantes positives, appelées respectivement taux d'apprentissage cognitif et taux social, w est le facteur d'inertie et λ est le facteur de constriction.

3.2.2 Optimiseur des loups gris (GWO)

GWO est un optimiseur méta-heuristique récent inspiré des loups gris et proposé par [5]. Il imite la hiérarchie du leadership et le mécanisme de chasse des loups gris dans la nature. Comme décrit dans la littérature, l'algorithme GWO comprend deux modèles mathématiques : encerclement des proies et la chasse des proies.

3.2.2.1 Le comportement d'encerclement

Pendant la chasse, les loups gris encerclent leurs proies. Le modèle mathématique est présenté dans les équations suivantes :

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| \quad (3.3)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A}\vec{D} \quad (3.4)$$

Où D est la distance, $\vec{X}_p(t)$ est le vecteur de position de la proie et $\vec{X}(t)$ indique la position du loup gris. t indique l'itération actuelle. \vec{A} et \vec{C} sont des vecteurs de coefficient.



Figure 3.2: Comportement de chasse des loups gris : (A) chasser, approcher et suivre les proies. (B-D) poursuivre, harceler et encercler. (E) situation stationnaire et attaque.

3.2.2.2 Le modèle de chasse

L'algorithme d'optimisation est guidé par α , β et δ ; trois meilleures solutions obtenues à ce jour, les autres agents de recherche les suivent et mettent à jour leurs positions en fonction du meilleur agent de recherche. Comme suit :

$$\vec{D}_j = |\vec{C}_i \cdot \vec{X}_j(t) - \vec{X}(t)| \quad (3.5)$$

$$\vec{X}_i = \vec{X}_j(t) - \vec{A}_i \cdot \vec{D}_j \quad (3.6)$$

et
$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (3.7)$$

Avec $j=\alpha, \beta, \delta$ et $i=1,2,3$.

3.2.3 Algorithme d'optimisation des baleines (WOA)

Comme l'algorithme GWO, le WOA est un algorithme d'optimisation méta-heuristique récent imitant le comportement de chasse des baleines à bosse [6]. La particularité de cette approche est dans la manière du comportement de chasse simulé qui se fait aléatoirement, c'est-à-dire le meilleur agent de recherche pour chasser la proie et l'utilisation de la spirale pour simuler le mécanisme d'attaque au filet à bulles des baleines à bosse. La philosophie de la chasse peut être décrite en trois étapes : encerclant des proies, méthode d'attaque au filet à bulles (phase d'exploitation), recherche de proies (phase d'exploration). Une probabilité de 50% de choisir entre le mécanisme d'encerclement rétrécissant ou le modèle en spirale pour mettre à jour la

position des baleines pendant l'optimisation a été supposée. Le modèle mathématique est le suivant :

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A} \vec{D} & \text{if } p < 0.5 \\ \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) & \text{if } p \geq 0.5 \end{cases} \quad (3.8)$$

Où p est un nombre aléatoire dans $[0, 1]$. La même approche basée sur la variation du vecteur \vec{A} peut être utilisée pour rechercher des proies (exploitation). D' indique la distance de la $i^{\text{ème}}$ baleine à la proie (meilleure solution obtenue jusqu'à présent X^*). b est une constante pour définir la forme logarithmique de la spirale, l est un nombre aléatoire dans $[-1, 1]$.

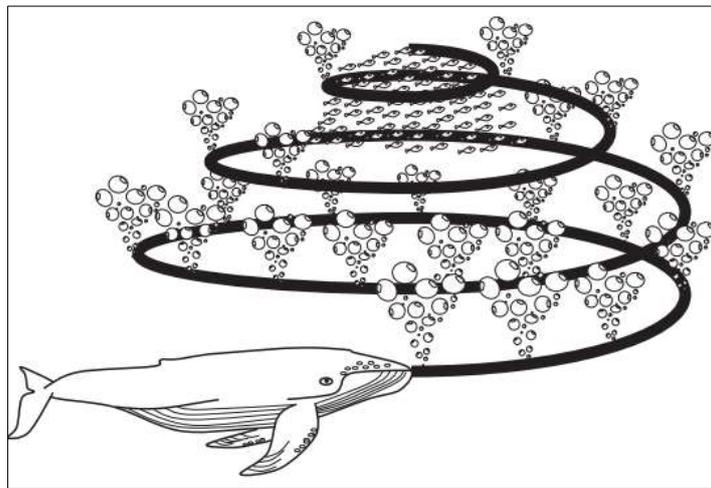


Figure 3.3: Comportement au filet à bulles des baleines.

3.2.4 Algorithme de sinus cosinus (SCA)

L'algorithme SCA appelé sine cosine algorithm est un nouvel algorithme d'optimisation basé sur la population pour résoudre les problèmes d'optimisation [7]. Le SCA crée plusieurs solutions candidates aléatoires initiales et les oblige à fluctuer vers l'extérieur ou vers la meilleure solution en utilisant un modèle mathématique basé sur les fonctions sinus et cosinus.

Pour mettre l'accent sur l'exploration et l'exploitation de l'espace de recherche dans différentes étapes d'optimisation, plusieurs variables aléatoires et adaptatives sont également intégrées à cet algorithme. Dans cet algorithme, les équations de mise à jour de position suivantes sont proposées pour les deux phases :

$$X_i^{t+1} = X_i^t + r_1 * \sin(r_2) * |r_3 * P_i^t - X_i^t| \quad (3.10)$$

$$X_i^{t+1} = X_i^t + r_1 * \cos(r_2) * |r_3 * P_i^t - X_i^t| \quad (3.11)$$

Où X_i^t est la position de la solution actuelle dans la $i^{\text{ème}}$ dimension à la $i^{\text{ème}}$ itération, $r_1 / r_2 / r_3$ sont des nombres aléatoires, P_i est la position du point de destination dans la $i^{\text{ème}}$ dimension, et $|\cdot|$ indique la valeur absolue. Ces deux équations sont combinées pour être utilisées comme suit :

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 * \sin(r_2) * |r_3 * P_i^t - X_i^t|, & r_4 < 0.5 \\ X_i^t + r_1 * \cos(r_2) * |r_3 * P_i^t - X_i^t|, & r_4 \geq 0.5 \end{cases} \quad (3.12)$$

Où r_4 est un nombre aléatoire dans $[0,1]$. Comme le montrent les équations ci-dessus, il existe quatre paramètres principaux dans SCA : r_1, r_2, r_3 et r_4 . Le paramètre r_1 dicte la région (ou le sens de déplacement) de la position suivante. Le paramètre r_2 définit dans quelle mesure le mouvement doit être vers ou vers l'extérieur de la destination. Le paramètre r_3 apporte un poids aléatoire pour la destination afin d'accentuer stochastiquement, pour ($r_3 > 1$), ou désaccentuer, pour ($r_3 < 1$), l'effet de destination dans la définition de la distance. Enfin, le paramètre r_4 bascule également entre les composantes sinus et cosinus dans l'équation. L'algorithme SCA démarre le processus d'optimisation par un ensemble de solutions aléatoires. L'algorithme enregistre ensuite les meilleures solutions obtenues jusqu'à présent, l'affecte comme point de destination et met à jour d'autres solutions par rapport à celui-ci. Pendant ce temps, les plages de fonctions sinus et cosinus sont mises à jour pour mettre l'accent sur l'exploitation de l'espace de recherche lorsque le compteur d'itérations augmente.

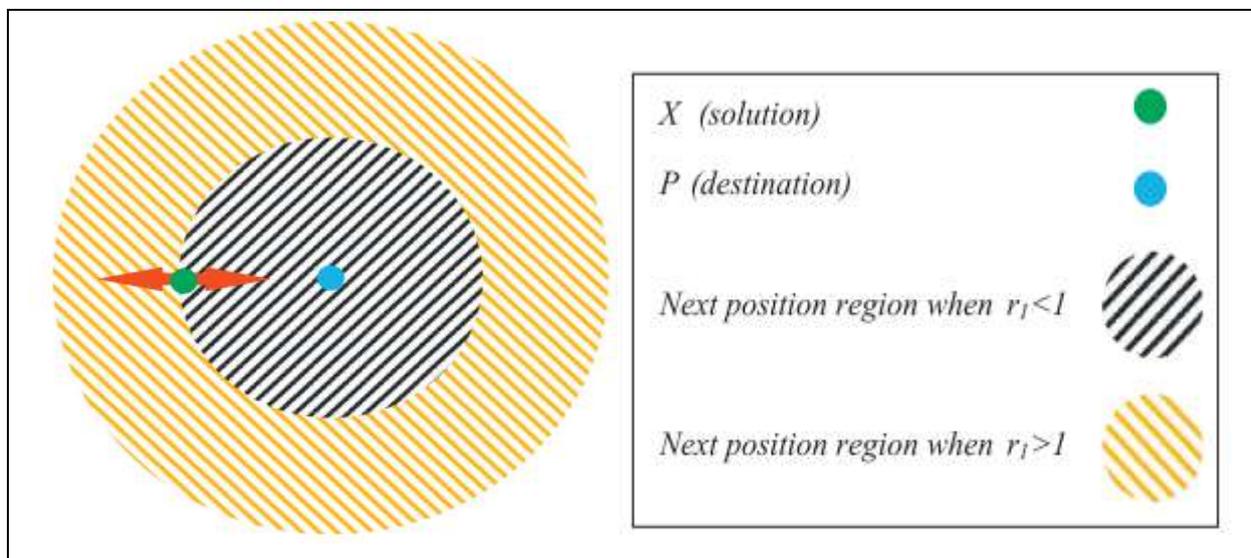


Figure 3.4: Effets de sine et cosine dans les équations (10) et (11) sur la position suivante.

3.2.5 Optimiseur du lion de fourmi (ALO)

Un autre nouvel algorithme inspiré de la nature appelé optimiseur du lion de fourmi (ALO) imite le mécanisme de chasse des lions des fourmis dans la nature [8]. Cinq étapes principales

de la chasse des proies telles que la marche aléatoire des fourmis, la construction des pièges, le piégeage des fourmis dans les pièges, la capture des proies et la reconstruction des pièges sont implémentées dans cet algorithme.

Étant donné que les fourmis se déplacent de manière stochastique dans la nature lors de la recherche de nourriture, une marche aléatoire est choisie pour modéliser le mouvement des fourmis comme suit :

$$X(t) = [0, \text{cumsum}(2r(t_1 - 1)), \text{cumsum}(2r(t_2 - 1)), \dots, \text{cumsum}(2r(t_n - 1))] \quad (3.13)$$

Où *cumsum* calcule la somme cumulée, *n* est le nombre maximal d'itérations, *t* représente le pas de marche ou d'itération aléatoire, et *r* (*t*) est une fonction stochastique définie comme suit :

$$r(t) = \begin{cases} 1 & \text{if } rand > 0.5 \\ 0 & \text{if } rand \leq 0.5 \end{cases} \quad (3.14)$$

Et *rand* est un nombre aléatoire généré avec une distribution uniforme dans l'intervalle de [0,1]. La position des fourmis est enregistrée et utilisée lors de l'optimisation dans la matrice suivante :

$$M_{Ant} = \begin{bmatrix} A_{1.1} & A_{1.2} & \dots & A_{1.d} \\ A_{2.1} & A_{2.2} & \dots & A_{2.d} \\ \vdots & & & \\ A_{n.1} & A_{n.2} & \dots & A_{n.d} \end{bmatrix} \quad (3.15)$$

Où $A_{i,j}$ montre la valeur de la $j^{ième}$ variable (dimension) de la $i^{ième}$ fourmi, *n* est le nombre de fourmis, et *d* est le nombre de variables. La position d'une fourmi fait référence aux paramètres d'une solution particulière. Une fonction de fitness (objectif) est utilisée pendant l'optimisation et la matrice suivante stocke la valeur de fitness de toutes les fourmis :

$$M_{OA} = \begin{bmatrix} f([A_{1.1}, A_{1.2}, \dots, A_{1.d}]) \\ f([A_{2.1}, A_{2.2}, \dots, A_{2.d}]) \\ \vdots \\ f([A_{n.1}, A_{n.2}, \dots, A_{n.d}]) \end{bmatrix} \quad (3.16)$$

Où *f* est la fonction objective. En plus des fourmis, nous supposons que les lions des fourmis se cachent également quelque part dans l'espace de recherche. Afin de sauvegarder leurs positions et les valeurs de fitness, les matrices suivantes sont utilisées :

$$M_{Antlion} = \begin{bmatrix} AL_{1.1} & AL_{1.2} & \dots & AL_{1.d} \\ AL_{2.1} & AL_{2.2} & \dots & AL_{2.d} \\ \vdots & & & \\ AL_{n.1} & AL_{n.2} & \dots & AL_{n.d} \end{bmatrix} \quad M_{OAL} = \begin{bmatrix} f([AL_{1.1}, AL_{1.2}, \dots, AL_{1.d}]) \\ f([AL_{2.1}, AL_{2.2}, \dots, AL_{2.d}]) \\ \vdots \\ f([AL_{n.1}, AL_{n.2}, \dots, AL_{n.d}]) \end{bmatrix} \quad (3.17)$$

$AL_{i,j}$ montre la valeur de la $j^{ième}$ dimension du $i^{ième}$ lion de fourmi, n est le nombre des lions des fourmis et d est le nombre de variables (dimension) où M_{OAL} est la matrice pour sauvegarder le fitness de chaque lion de fourmi $AL_{i,j}$.

3.2.5.1 Marches aléatoires de fourmis

Les marches aléatoires sont toutes basées sur l'équation (3.13). Les fourmis mettent à jour leur position X_i^t avec une marche aléatoire à chaque étape de l'optimisation, qui est normalisé à l'aide de l'équation suivante (normalisation min – max) afin de les garder à l'intérieur de l'espace de recherche:

$$X_i^t = \frac{(X_i^t - a_i) * (d_i - c_i^t)}{(d_i^t - a_i)} + c_i \quad (3.18)$$

Où a_i est le minimum de marche aléatoire, d_i est le maximum de marche aléatoire, c_i^t est le minimum, d_i^t indique le maximum de la $i^{ième}$ variable à la $t^{ième}$ itération.

3.2.5.2 Piégeage dans les fosses des lions des fourmis

Les fourmis marchent dans une hypersphère définie par les vecteurs c et d , autour d'un lion de fourmi sélectionné, affecté par les pièges des lions de fourmis. Afin de modéliser mathématiquement cette supposition, les équations suivantes sont proposées :

$$c_i^t = Antlion_j^t + c^t \quad (3.19)$$

$$d_i^t = Antlion_j^t + d^t \quad (3.20)$$

Où c^t est le minimum de toutes les variables à la $i^{ième}$ itération, d^t indique le vecteur incluant le maximum de toutes les variables à la $t^{ième}$ itération, c_i^t est le minimum de toutes les variables pour la $i^{ième}$ fourmi, d_i^t est le maximum de toutes les variables pour la $i^{ième}$ fourmi et $Antlion_j^t$ montre la position du $j^{ième}$ lion de fourmi sélectionné à $t^{ième}$ itération.

3.2.5.3 Construction de piège

Afin de modéliser la capacité de chasse des lions des fourmis, une roue de roulette est utilisée pour sélectionner les lions des fourmis en fonction de leur aptitude lors de l'optimisation. Les fourmis sont supposées être piégé dans un seul lion de fourmi sélectionné. Ce mécanisme donne de grandes chances aux lions des fourmis plus aptes d'attraper des fourmis.

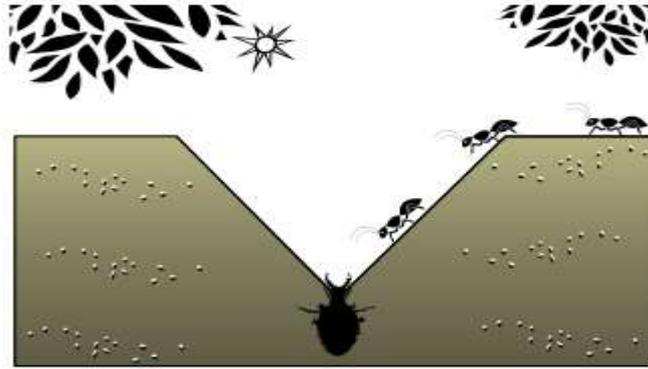


Figure 3.5: Pièges coniques et comportement de chasse des lions des fourmis.

3.2.5.4 Glissement des fourmis vers lion de fourmi

Une fois que les lions des fourmis se rendent compte qu'une fourmi est dans le piège, ils tirent des sables vers l'extérieur au centre de la fosse. Ce comportement glisse vers le bas la fourmi piégée qui tente de s'échapper. Pour modéliser mathématiquement ce comportement, le rayon de l'hyper-sphère des promenades aléatoires des fourmis est diminué de manière adaptative. Les équations suivantes sont proposées à cet égard :

$$c^t = \frac{c^t}{I} \quad (3.21)$$

$$d^t = \frac{d^t}{I} \quad (3.22)$$

Dans les équations (3.21) et (3.22), I est un rapport défini par $I = 10^{w \frac{t}{T}}$, où T est le nombre maximal d'itérations, et w est une constante définie en fonction de l'itération actuelle. Fondamentalement, la constante w peut ajuster le niveau de précision de l'exploitation.

3.2.5.5 Attraper des proies et reconstruire la fosse

Pour imiter la dernière étape du processus de chasse, il est supposé que la capture des proies se produit lorsque les fourmis deviennent plus en forme (pénètrent dans le sable) que son lion de fourmi correspondant, qui est nécessaire pour mettre à jour sa position à la dernière position de la fourmi chassée. Cela est défini par l'équation suivante :

$$Antlion_j^t = Ant_i^t \quad \text{if } f(Ant_j^t) > f(Antlion_j^t) \quad (3.23)$$

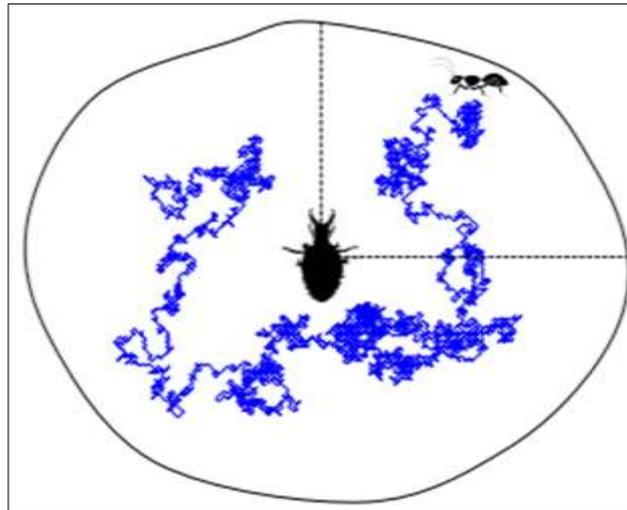


Figure 3.6 : Promenade aléatoire d'une fourmi à l'intérieur d'un piège du lion des fourmis.

3.2.5.6 Élitisme

Dans cet algorithme, le meilleur lion de fourmi obtenu jusqu'à présent à chaque itération est enregistré et considéré comme une élite. Étant donné que l'élite est le meilleur lion de fourmi le plus apte, il doit pouvoir affecter les mouvements de toutes les fourmis lors des itérations par conséquent, on suppose que chaque fourmi se promène au hasard autour d'un lion de fourmi sélectionné par la roulette et l'élite simultanément comme suit :

$$Ant_i^t = \frac{R_A^t + R_E^t}{2} \quad (3.24)$$

Où Ant_i^t indique la position de la $i^{\text{ème}}$ fourmi, R_A^t et R_E^t sont la marche aléatoire autour du lion de fourmi et de l'élite sélectionnée par la roulette à la $t^{\text{ème}}$ itération.

3.2.6 Algorithme du moule visqueux (SMA)

L'algorithme du moule visqueux (SMA) est un nouvel optimiseur stochastique proposé par Shimin Li [2]. Basé sur le mode d'oscillation de la moisissure visqueuse dans la nature il s'agit de *Physarum polycephalum*. Puisqu'il a été d'abord classé comme champignon, alors il a été nommé "Moule visqueux".

3.2.6.1 Concept et elicitation

Le moule visqueux est un eucaryote qui habite au frais et les endroits humides. Le stade nutritionnel principal est le Plasmodium, le stade actif et dynamique du moule visqueux. À ce stade, la matière organique dans le moule visqueux cherche de la nourriture, l'entoure et sécrète des enzymes pour la digérer. Pendant le processus de migration, le front extrémité se prolonge

en forme d'éventail, suivie d'un réseau veineux interconnecté qui permet au cytoplasme de se couler à l'intérieur, comme le montre la figure 3.7. En raison de leur motif et de leurs caractéristiques uniques, ils peuvent utiliser plusieurs sources de nourriture en même temps pour former un réseau veineux les reliant. S'il y a suffisamment de nourriture dans l'environnement, la moisissure visqueuse peut même atteindre plus de 900 centimètres carrés.

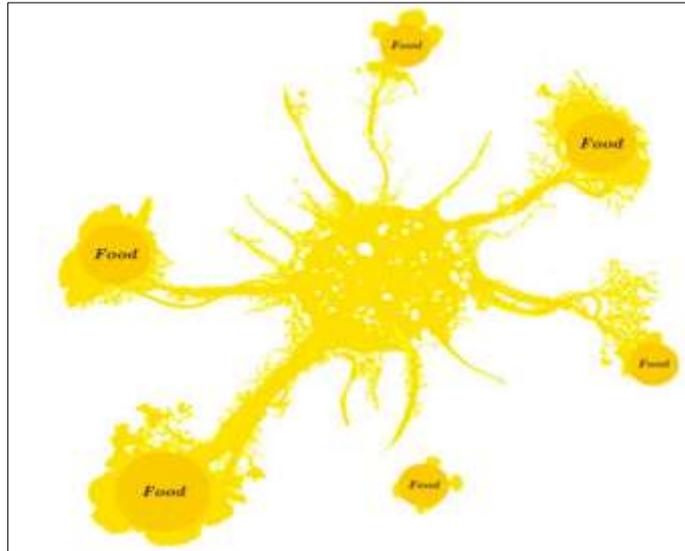


Figure 3.7: Morphologie de recherche du moule visqueux.

Nous savons maintenant que lorsqu'une veine s'approche d'une source de nourriture, le bio-oscillateur produit une onde qui augmente le flux cytoplasmique dans la veine, et plus le flux cytoplasmique est rapide, plus la veine est épaisse. Grâce à cette combinaison de rétroaction positive-négative, la boue peut établir le chemin optimal pour connecter les aliments d'une manière relativement supérieure.

3.2.6.2 *Modèle mathématique*

Certains des mécanismes et caractéristiques de la moisissure visqueuse sont modélisés mathématiquement. Dans cette partie, le modèle mathématique sera décrit comme suit :

i. Approche de la nourriture (Approach food)

Le moule visqueux peut s'approcher des aliments en fonction de l'odeur de l'air. Les formules suivantes expriment mathématiquement son comportement d'approche pour imiter le mode de contraction :

$$\overrightarrow{X(t+1)} = \begin{cases} \overrightarrow{X_b(t)} + \overrightarrow{vb} \cdot (\overrightarrow{W} \cdot \overrightarrow{X_A(t)} - \overrightarrow{X_B(t)}), & r < p \\ \overrightarrow{vc} \cdot \overrightarrow{X(t)}, & r \geq p \end{cases} \quad (3.25)$$

$$(3.26)$$

Où \vec{vb} est un paramètre avec une plage de $[-a, a]$, \vec{vc} décroît linéairement de un à zéro. t Représente l'itération actuelle. \vec{X}_b représente l'emplacement individuel avec la concentration d'odeur la plus élevée actuellement trouvée. \vec{X} représente l'emplacement du moule visqueux. \vec{X}_A et \vec{X}_B représentent deux individus choisis au hasard dans un moule visqueux et \vec{W} Représente le poids du moule visqueux.

Avec

$$p = \tanh|S(i) - DF| \quad (3.27)$$

$$a = \operatorname{arctanh}\left(-\left(\frac{t}{\max_t}\right) + 1\right) \quad (3.28)$$

$$\vec{W}(\operatorname{smellindex}(i)) = \begin{cases} 1 + r \cdot \log\left(\frac{bF-S(i)}{bF-wF} + 1\right), & \text{condition} \\ 1 - r \cdot \log\left(\frac{bF-S(i)}{bF-wF} + 1\right), & \text{others} \end{cases} \quad (3.29)$$

$$\operatorname{smellindex} = \operatorname{sort}(S) \quad (3.30)$$

Où $i \in 1, 2, \dots, n$, $S(i)$ représente l'aptitude de \vec{X} , DF désigne la meilleure aptitude obtenue dans toutes les itérations. r désigne la valeur aléatoire dans l'intervalle de $[0, 1]$.

ii. *Enveloppe de la nourriture (Wrap food)*

Cela simule mathématiquement le mode de contraction de la structure du tissu veineux du moule visqueux lors de la recherche. Plus l'onde générée par le bio-oscillateur est forte, plus le cytoplasme s'écoule rapidement et plus la veine est épaisse. La formule mathématique pour mettre à jour l'emplacement du moule visqueux est la suivante :

$$\vec{X}^* = \begin{cases} \operatorname{rand} \cdot (UB - LB) + LB, & \operatorname{rand} < z \\ \vec{X}_b(t) + \vec{vb} \cdot (\vec{W} \cdot \vec{X}_A(t) - \vec{X}_B(t)), & r < p \\ \vec{vc} \cdot \vec{X}(t), & r \geq p \end{cases} \quad (3.31)$$

Où LB et UB désignent les limites inférieure et supérieure de la plage de recherche. La valeur de z sera discutée dans l'expérience de paramétrage.

iii. *Nourriture de Grabble (Grable food)*

Le moule visqueux dépend principalement de l'onde de propagation produite par l'oscillateur biologique pour modifier le flux cytoplasmique dans les veines, de sorte qu'ils ont tendance à être dans une meilleure position de concentration des aliments. Dans le but de simuler les variations de la largeur veineuse du moule visqueux, \vec{W} , \vec{vb} et \vec{vc} ont été utilisés pour réaliser

ces variations. La valeur de \vec{vb} oscille de façon aléatoire entre $[-a, a]$, et s'approche progressivement de zéro à mesure que l'augmentation des itérations. La valeur de \vec{vc} oscille entre $[-1,1]$ et tend finalement vers zéro. La tendance des deux valeurs est illustrée à la figure 3.8.

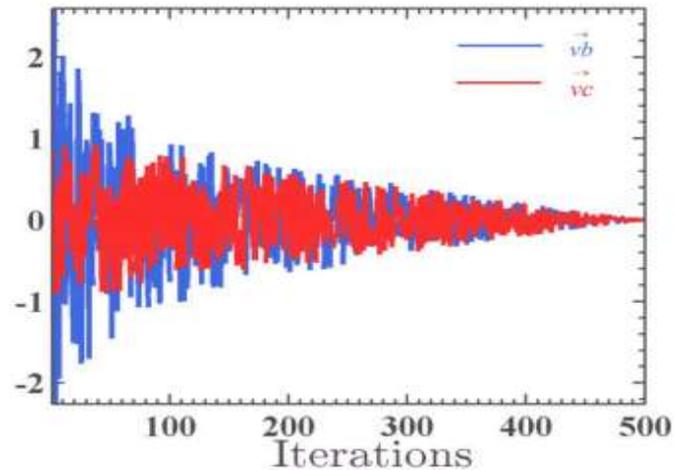


Figure 3.8: Tendances de \vec{vb} et \vec{vc} .

De plus, le processus d'oscillation de \vec{vb} simule l'état du moule visqueux en décidant s'il faut s'approcher de la source de nourriture ou trouver d'autres sources de nourriture. Pendant ce temps, le processus de sondage des aliments n'est pas lisse. Pendant cette période, il peut y avoir divers obstacles, tels qu'un environnement sec et léger, qui limitent la propagation du moule visqueux. Cependant, cela améliore également la possibilité de trouver des aliments de meilleure qualité et échappe au piègeage de l'optimum local.

3.3 Conclusion

Les algorithmes méta-heuristique ont connu une large diffusion dans la résolution des problèmes d'optimisation dans de nombreux domaines, en particulier dans le domaine de l'ingénierie et de la robotique. Dans ce chapitre, nous avons présenté quelques algorithmes d'optimisation qui seront utilisés dans la suite, pour la commande du robot manipulateur.

Références du chapitre 3 :

1. **Osher S, Wang B, Yin P, Luo X, Baretat F, Pham M, Lin A:** *Laplacian Smoothing Gradient Descent*. Mathematics, Computer Science, 2018.
2. **Shimin Li, Huiling Chen, Mingjing Wang, Ali Asghar Heidari, Seyedali Mirjalili:** *Slime mould algorithm: A new method for stochastic optimization*. Future Generation Computer Systems.1-40, April 2020.
3. **Wolpert D, Macready W:** *No Free Lunch Theorems for Optimization*. IEEE Transactions on evolutionary computation, Vol. 1. 67-82, 1997.
4. **Kennedy J, Eberhart R:** *Particles swarm optimization*. IEEE International Conference on Neural Networks. 1942-1948, 1995.
5. **Mirjalili S, Mirjalili MS, Lewis A:** *Grey wolf optimizer*. Advance in engineering software. 69:46-61, 2014.
6. **Mirjalili S, Lewis A:** *The whale optimization algorithm*. Advance in engineering software 95:51-67, 2016.
7. **Mirjalili S:** *SCA A Sine Cosine Algorithm for solving optimization problems*. Knowledge-Based Systems. pp. 1–14, 2016.
8. **Mirjalili S:** *The Ant Lion Optimizer*. Advances in Engineering Software, 83 80–98, 2015.

Chapitre 4

Commandes
classiques des robots
manipulateurs

Commandes classiques des robots manipulateurs

4.1 Introduction

Un problème fondamental dans le contrôle des robots manipulateur est la poursuite d'une trajectoire souhaitée pré-planifiée. Dans le but de réaliser sa tâche avec précision, il doit atteindre sa position finale au bon moment. Dans ce chapitre, nous examinons quelques schémas de contrôle simples pour les robots manipulateurs appartenant à la classe dite commande classique. Celles-ci fonctionnent généralement bien lorsque les paramètres du bras du robot sont connus assez exactement [1]. Nous abordons des connexions avec le contrôle classique du manipulateur en utilisant le contrôle PID. Ensuite nous montrons comment utiliser certaines techniques de conception en conjonction avec contrôle du couple calculé.

Le robot manipulateur est un système couplé à entrées multiples (MIMO), hautement non linéaire, conçu pour effectuer automatiquement des tâches émulant ou reproduisant, dans un domaine spécifique, des actions humaines. Les applications typiques incluent le soudage, la peinture, l'assemblage, etc. La robotique industrielle doit inspecter les produits rapidement et avec précision, de manière à améliorer leurs performances en insérant des contrôleurs efficaces. Par conséquent, concevoir un contrôleur efficace pour ce système est une tâche ardue [2]. Malgré le succès de la théorie du contrôle moderne, les contrôleurs du robot manipulateur utilisent encore couramment l'algorithme proportionnel-dérivé (PD) ou proportionnel-intégral-dérivé (PID) [3-6]. Pour améliorer leurs performances, la plupart de ces contrôleurs ont été conçus à l'aide de modèles linéaires ou linéarisés. Dans [7,8], des applications du contrôle PID non linéaire sont proposées soit en ajoutant un système proportionnel non linéaire et terme dérivé du contrôleur PID ou avec compensation de gravité pour les robots manipulateurs. Pour éliminer l'angle d'oscillation des effecteurs terminaux du robot manipulateur à articulation flexible et à liaison simple, la trajectoire de position est réalisée par le procédé de contrôle de retour d'état [9]. Un schéma de contrôle type PID généralisé avec un réglage simple pour la régulation globale du robot a également été présenté avec entrées contraintes [10]. En ce qui

concerne les approches de stabilité et / ou de contrôle telles que le contrôle prédictif, le contrôle par retour d'état, le contrôle décentralisé, les lecteurs peuvent se reporter à ces ouvrages [11-18]. Dans un contexte adaptatif, les auteurs de [19] ont présenté un contrôle PID adaptatif avec perturbation limitée. Ceux de [20] ont conçu le contrôleur en tenant compte des incertitudes non linéaires du système. La loi d'adaptation est motivée par le contrôle du mode glissant et dérivée pour accorder les gains PID afin de minimiser les conditions de glissement. Par la suite, d'autres applications basées sur des réseaux de neurones adaptatifs ont été introduites pour contrôler le robot manipulateur [21]. Récemment, certains chercheurs ont fait un effort pour développer de nouvelles stratégies de contrôle basées sur l'intelligence artificielle. Une approche basée sur le partitionnement de l'espace en segments décrit l'apprentissage du contrôle de rétroaction du manipulateur de robot où le comportement dynamique du robot est considéré comme un système linéaire découplé contrôlé par le contrôleur PID. Les segments sont représentés par des ensembles flous assurant la continuité des variables de contrôle [22]. Le contrôleur PID à réglage automatique par logique floue est introduit dans [23-25], les paramètres du contrôleur sont variés et calculés en ligne. Dans le contexte hybride, les automates d'apprentissage sont utilisés pour ajuster les paramètres du contrôleur PID flou pour un suivi optimal du robot [26]. Afin de minimiser l'erreur en régime permanent avec respect des incertitudes dans le contrôle du robot, le contrôleur PID nécessite un grand gain intégral lorsqu'un compensateur neuronal est ajouté à la commande PD classique avec un gain dérivé élevé [27]. Un moyen élégant d'améliorer les performances des contrôleurs PID consiste à trouver le réglage approprié des valeurs de paramètres (K_p , K_i et K_d) pour obtenir les performances optimales du robot. Récemment, plusieurs travaux ont été menés dans des approches d'optimisation. En littérature, certaines œuvres ont proposées des méthodes combinant l'algorithme génétique (GA) et logique floue pour optimiser les gains PID du Contrôle du robot avec comportement interne inconnu où GA est utilisé dans certaines applications comme estimateur principal de gain associé à la logique floue comme base de classement pour GA [28], et dans d'autres la logique floue est utilisée en tant qu'estimateur principal de gain associé à un analyseur de signal pour extraire du signal d'erreur du contrôleur, certains indices de performance, à savoir le dépassement, le temps de montée et l'erreur statique [29]. Une autre application de GA consiste à rechercher les fonctions d'appartenance optimales en entrée et en sortie d'un contrôleur PID à réglage automatique flou [30]. Afin de contrôler le robot manipulateur à deux degrés de liberté (2ddl), Elkhateeb et Badr ont appliqués l'algorithme de colonie d'abeilles pour régler le contrôleur PID [31]. Une version modifiée de l'algorithme d'optimisation de la mouche des fruits (MFOA) est également utilisée pour trouver les

paramètres PID optimaux dans lesquels la dynamique du robot non linéaire utilisée est linéarisée et découplée par une technique de contrôle de linéarisation à rétroaction non linéaire [32]. Dans le même contexte, un nouvel algorithme d'optimisation à base de biogéographie modifiée (BBO) est développé pour ajuster les paramètres du PID d'un robot à cinq barres [33]. D'autres travaux intéressants ont vu le jour grâce à l'algorithme PSO [34-40] tels que l'application de l'algorithme PSO dans la chaîne de contrôle du robot SCARA. D'autre part, nous avons constaté une attention accrue portée à l'enquête sur l'application de la GWO telle que l'optimisation de la planification de trajectoire de robot avec une approche multi-objectifs de GWO [41].

4.2 Commande PID

Un contrôleur PID est essentiellement un mécanisme de rétroaction en boucle fermée générique. Surveille l'erreur entre la sortie du système mesurée et le point de consigne souhaité. A partir de cette erreur, un signal de commande est calculé pour ajuster les performances du processus. L'équation différentielle du contrôleur PID est :

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^{t_f} e(t) dt + K_d \cdot \frac{de(t)}{dt} \quad (4.1)$$

Où K_p , K_i et K_d sont respectivement les gains proportionnel, intégral et dérivé. La superposition de ces trois actions constitue le mécanisme d'ajustement des performances des processus comme le montre la figure 4.1. La fonction de transfert continue du contrôleur PID est obtenue via la transformée de Laplace :

$$C(p) = K_p + \frac{K_i}{p} + K_d \cdot p \quad (4.2)$$

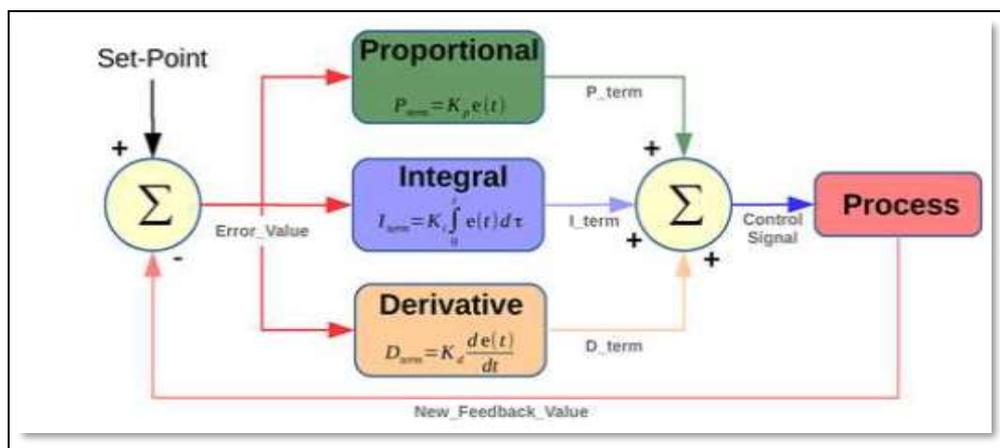


Figure 4.1 : Mécanisme de contrôle PID.

4.2.1 Résultats de simulation

Dans cette section, un contrôleur PID est conçu pour le contrôle de suivi de trajectoire d'un robot manipulateur à l'aide des techniques d'optimisation décrites dans le chapitre 3. L'idée est de déterminer les paramètres optimaux (K_p , K_i et K_d) garantissant les meilleures indices de performances (temps de montée, dépassement, temps de stabilisation et temps de pic) du système. L'objectif principal est d'appliquer le nouvel algorithme d'optimisation (WOA) pour régler les paramètres PID et démontrer ainsi son efficacité par rapport à celles obtenues des deux algorithmes les plus connus GWO et PSO. A partir des paramètres initialisés aléatoirement, l'algorithme d'optimisation minimise le critère d'intégral en temps d'erreur absolue (*ITAE*), mentionné dans l'équation 4.3. Pour chaque itération jusqu'à l'obtention d'un ensemble optimal des paramètres K_p , K_i et K_d . Tout d'abord, nous réglons le contrôleur PID avec trois algorithmes pour les différentes valeurs des agents ou taille de population de : 10, 20, 30, 50, 100 et 200. Nous choisissons ensuite la meilleure taille de population pour chaque algorithme.

$$J = ITAE = \int_0^{t_f} t \cdot |e(t)| dt \quad (4.3)$$

$e(t) = q_d(t) - q(t)$ est l'erreur entre la position de référence et la position réelle du robot.

Dans cette étude, les résultats de simulation sont obtenus en utilisant l'environnement MATLAB / Simulink. La figure 4.2 illustre le diagramme Simulink de la stratégie de contrôle de deux articulations du robot manipulateur. La dynamique du robot manipulateur est décrit par l'équation (2.47) du chapitre 2.

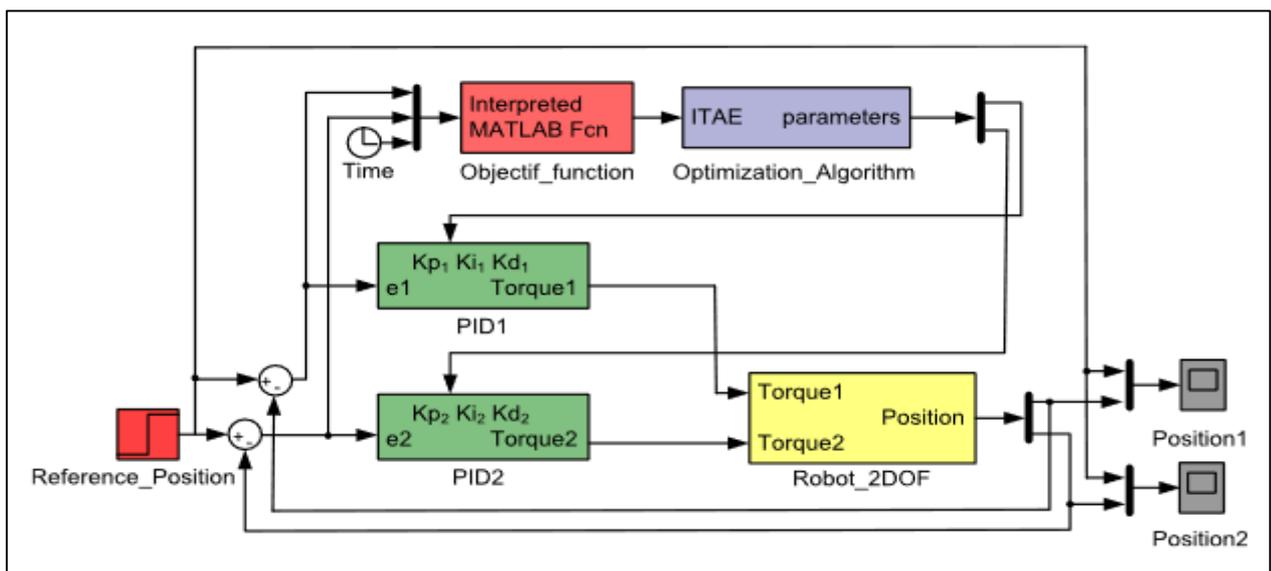


Figure 4. 2: Schéma Simulink de la stratégie de contrôle PID.

Les paramètres du robot considérés dans l'application sont les masses des deux liaisons $m_1 = 10$ kg, $m_2 = 5$ kg, et les longueurs $l_1 = 1$ m et $l_2 = 0,5$ m avec l'accélération gravitationnelle $g = 9,8$ m / s². Les positions désirées et réelles de la première et la deuxième articulation du robot manipulateur sont représentés sur les figures (4.4), (4.5) et (4.6) avec les conditions initiales $q_1(0) = 0.5$ deg, $q_2(0) = 1$ deg, $\dot{q}_1(0) = 0$ et $\dot{q}_2(0) = 0$. De plus, tous les contrôleurs étaient testés dans les mêmes conditions sur un horizon de simulation de 10 s, et le signal de référence est d'amplitude 10 degrés. Les tableaux (4.1), (4.2) et (4.3) illustrent les caractéristiques temporelles des réponses des deux articulations avec un nombre d'agents différent.

Le tableau (4.1) montre que la fonction objective inférieure à 7.94 est obtenue avec 200 agents pour l'optimisation PSO qui correspond au plus petit temps de montée pour les deux articulations et les autres caractéristiques sont en moyenne acceptables. Nous pouvons observer également, tableau (4.2), que la fonction objective d'optimisation GWO est faible avec une valeur de 6.464 pour seulement 20 agents. Outre la valeur de dépassement de la première articulation, GWO offre des améliorations par rapport au PSO en termes de temps de réponse pour la première articulation (0.453 / 1.296) et la deuxième articulation (0.585 / 2.8678), et améliore la fonction de coût en diminuant la valeur de (3.834 à 3.157) pour la première articulation et de (4.106 à 3.307) pour la deuxième articulation. Alors que dans le tableau (4.3), l'algorithme WOA a la plus petite fonction de coût (5.24), avec seulement 10 agents. Mais en raison du dépassement remarquable de la première articulation, il semble que les résultats obtenus avec 30 agents sont les meilleurs, avec une faible fonction de coût (6.97), par rapport aux autres valeurs.

En général, le WOA présente les meilleures performances pour le système contrôlé. Les différentes valeurs des paramètres du contrôleur déduit du processus d'optimisation avec la fonction objective (4.3) sont récapitulées dans le tableau (4.4). Afin de prouver l'efficacité de l'algorithme WOA en terme de temps de calcul, nous avons testé les trois algorithmes pour différents et même nombre d'agents, les résultats numériques dans le tableau (4.5) montrent que l'algorithme WOA est plus rapide par rapport aux algorithmes GWO et PSO, c'est-à-dire qu'il converge plus rapidement à l'optimum. Ceci présente alors un autre critère de choix de l'algorithme WOA et montre la supériorité de l'algorithme sur les autres algorithmes. À partir des figures (4.3), (4.4) et (4.5), nous pouvons voir que la première articulation a une forme oscillatoire pour les trois techniques d'optimisation et les différentes tailles de population, avec

un dépassement relatif remarquable pour le GWO-PID et WOA-PID entre [2.18–15.11%] et [0.43–20.78%] respectivement, où le dépassement maximum ne dépasse pas 6.4% pour le contrôle PID optimisé par l'algorithme PSO. Pour la deuxième articulation, la réponse est presque apériodique avec un petit dépassement de 3.87%, 4.53% et 2.774% maximum pour les Commandes de PSO-PID, GWO-PID et WOA-PID respectivement.

On peut observer aussi que la réponse de la première articulation converge vers la valeur finale, pour le contrôle PID optimisé par les algorithmes GWO et WOA, avec un temps de réponse entre [0.38–0.48 s] et [0.27–0.51 s] respectivement, et également avec un intervalle de [0.58–4.15 s] et [0.57–3.44 s] pour la seconde articulation, où l'optimisation par WOA présente les meilleurs résultats. Au contraire, le temps de réponse du robot contrôlé par un PSO-PID varie entre [0.44 – 1.6 s] pour la première articulation et entre [2.86–4.33 s] pour la deuxième articulation.

Tableau 4.1 : Résultats de comparaison des caractéristiques temporelles de la première et la deuxième d'articulation contrôlées par PID-PSO.

	N ^{bre} d'agents	Temps de monté	Temps de réponse	Dépassement	Pic	ITAE
Articulation 1	10	0.2847	1.6044	2.0626	10.261	5.519
	20	0.3706	0.6150	1.8813	10.264	6.694
	30	0.2845	0.4513	1.8375	10.242	5.487
	50	0.379	0.650	0.951	10.18	6.034
	100	0.135	0.445	6.408	10.68	2.958
	200	0.186	1.296	2.595	10.29	3.834
Articulation 2	10	0.428	3.379	2.911	10.34	6.487
	20	0.3150	2.9138	3.2203	10.344	3.813
	30	0.4307	3.545	3.0994	10.36	6.542
	50	0.5014	4.3387	3.8711	10.443	7.86
	100	0.395	3.430	3.177	10.36	6.002
	200	0.3138	2.8678	3.229	10.344	4.106
	Articulation 1 et 2					
N ^{bre} d'agents	10	20	30	50	100	200
Fonction objectif	12.006	10.925	12.029	13.894	8.96	7.94

Tableau 4.2 : Résultats de comparaison des caractéristiques temporelles de la première et la deuxième d'articulation contrôlées par PID-GWO.

	N ^{bre} d'agents	Temps de monté	Temps de réponse	Dépassement	Pic	ITAE
Articulation 1	10	0.080	0.468	15.11	11.54	2.243
	20	0.15	0.453	7.8466	10.828	3.157
	30	0.0863	0.3841	13.0525	11.3238	2.097
	50	0.175	0.4795	3.4604	10.3912	3.461
	100	0.090	0.4086	8.7769	10.8973	2.295
	200	0.1936	0.4288	2.1872	10.2687	3.113
Articulation 2	10	0.471	2.191	1.9771	10.277	7.197
	20	0.343	0.585	0.4101	10.09	3.307
	30	0.387	0.631	1.512	10.21	5.579
	50	0.449	3.236	2.658	10.32	6.932
	100	0.444	2.277	2.038	10.27	6.784
	200	0.4293	4.1512	4.5348	10.4862	6.426
Articulation 1 et 2						
N ^{bre} d'agents	10	20	30	50	100	200
Fonction objectif	9.44	6.464	7.676	10.396	9.079	9.539

Tableau 4.3 : Résultats de comparaison des caractéristiques temporelles de la première et la deuxième d'articulation contrôlées par PID-WOA.

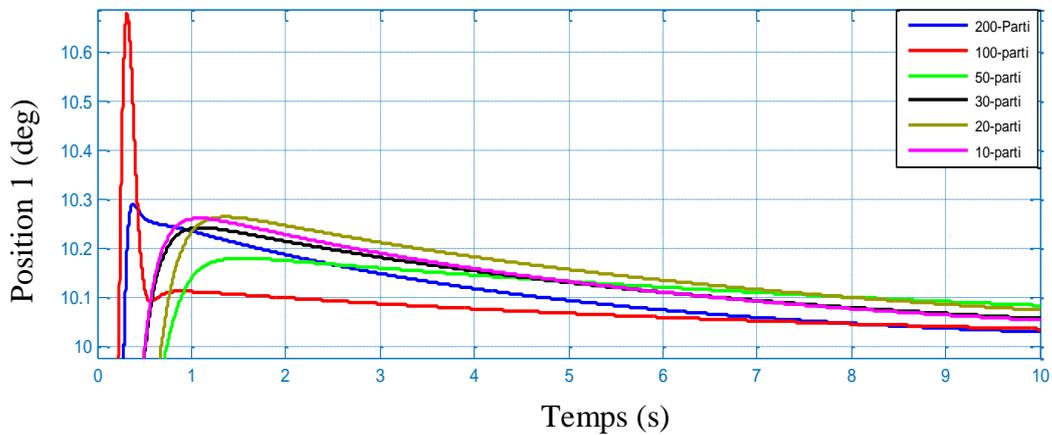
	N ^{bre} d'agents	Temps de monté	Temps de réponse	Dépassement	Pic	ITAE
Articulation 1	10	0.101	0.5104	20.784	12.117	2.033
	20	0.25	0.38	0.434	10.10	3.813
	30	0.1896	0.2728	1.3703	10.19	3.668
	50	0.1224	0.446	17.61	11.78	2.541
	100	0.1824	0.4402	3.058	10.35	3.669
	200	0.079	0.488	19.99	12.03	2.165
Articulation 2	10	0.336	0.577	0.392	10.08	3.207
	20	0.476	3.442	2.774	10.34	7.327
	30	0.342	0.5841	0.4066	10.089	3.301
	50	0.415	0.727	0.498	10.11	4.176
	100	0.364	0.591	1.302	10.18	4.928
	200	0.509	0.789	1.371	10.22	7.111
Articulation 1 et 2						
N ^{bre} d'agents	10	20	30	50	100	200
Fonction objectif	5.24	11.138	6.97	6.717	8.597	9.272

Tableau 4.4 : Paramètres des contrôleurs PID-PSO, PID-GWO et PID-WOA.

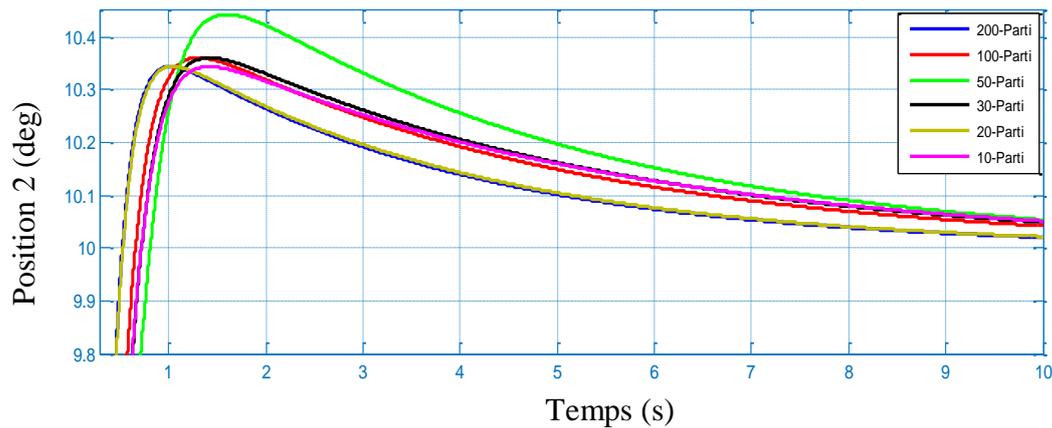
Algorithmes / Paramètres PID	K_p		K_i		K_d	
	Articul 1	Articul 2	Articul 1	Articul 2	Articul 1	Articul 2
PSO	4000	3000	900	900	462.87	450
GWO	4000	3000	639.45	210	346.79	457.03
WOA	4000	3000	350.27	210	453.69	450

Tableau 4.5 : Temps de convergence des trois algorithmes.

	PSO	GWO	WOA
Temps de convergence/N ^{bre} d'agents	09 min 53 s/200	2.1117 s/20	1.363 s/30
	53.392 s/20	1.882 s/20	0.928 s/20

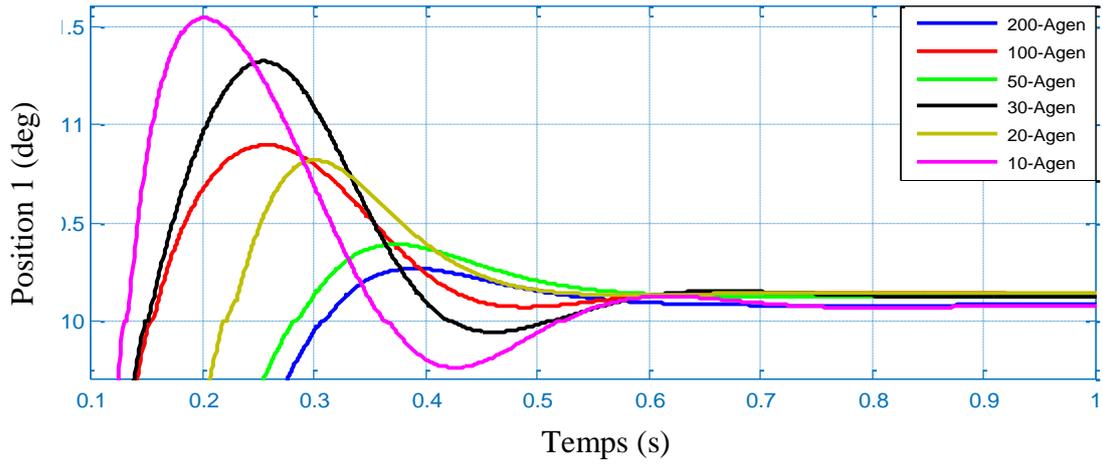


(a) Première articulation

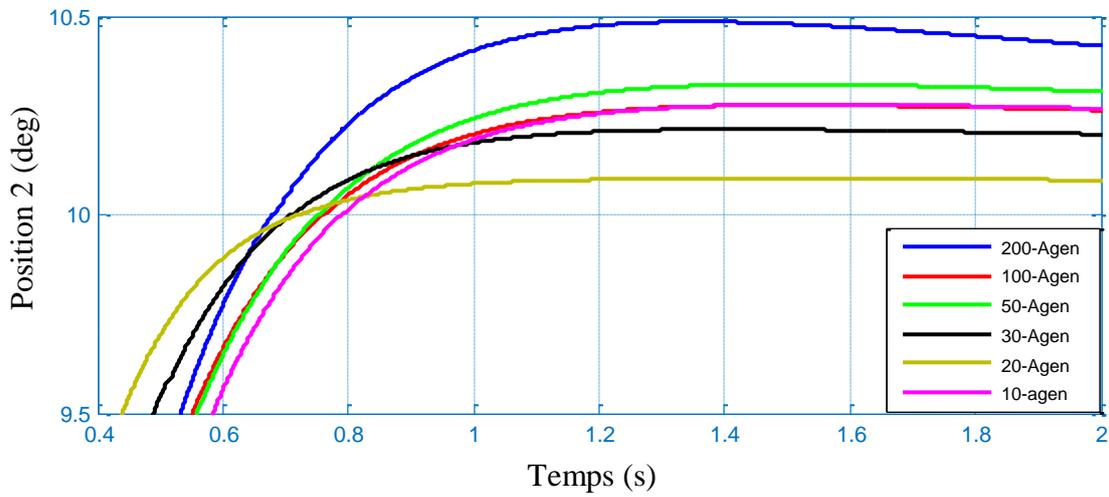


(b) Deuxième articulation

Figure 4.3 : Position actuelle du robot manipulateur contrôlé par PID-PSO.

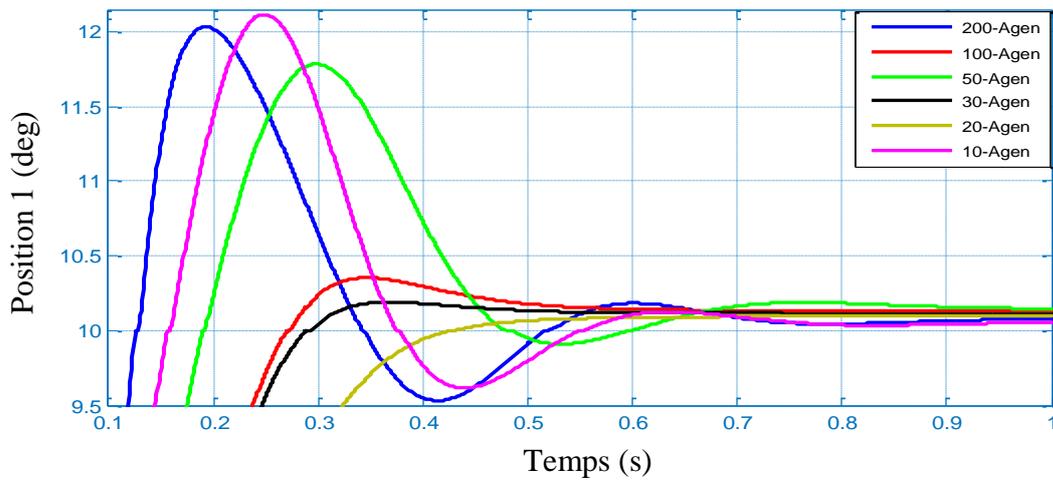


(a) Première articulation

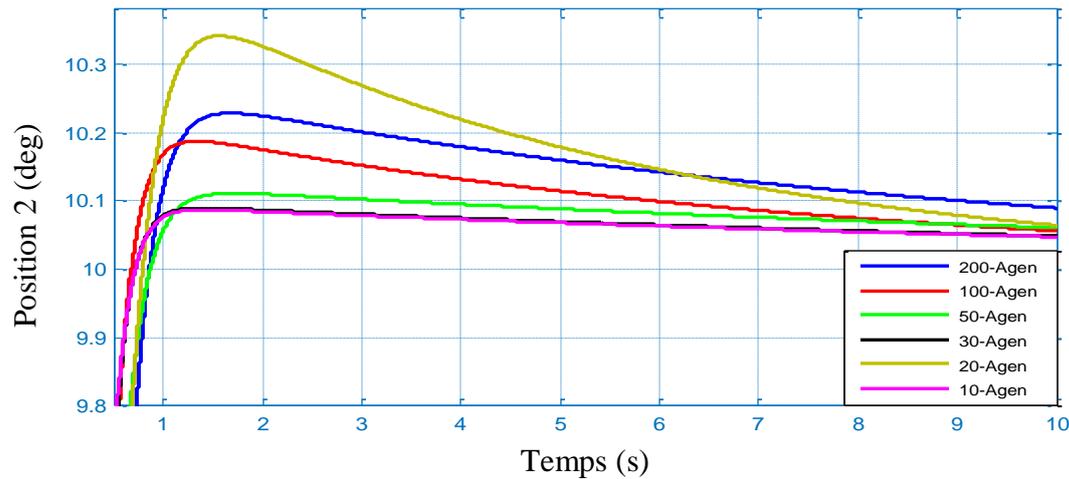


(b) Deuxième articulation

Figure 4.4 : Position actuelle du robot manipulateur contrôlé par PID-GWO.



(a) Première articulation.

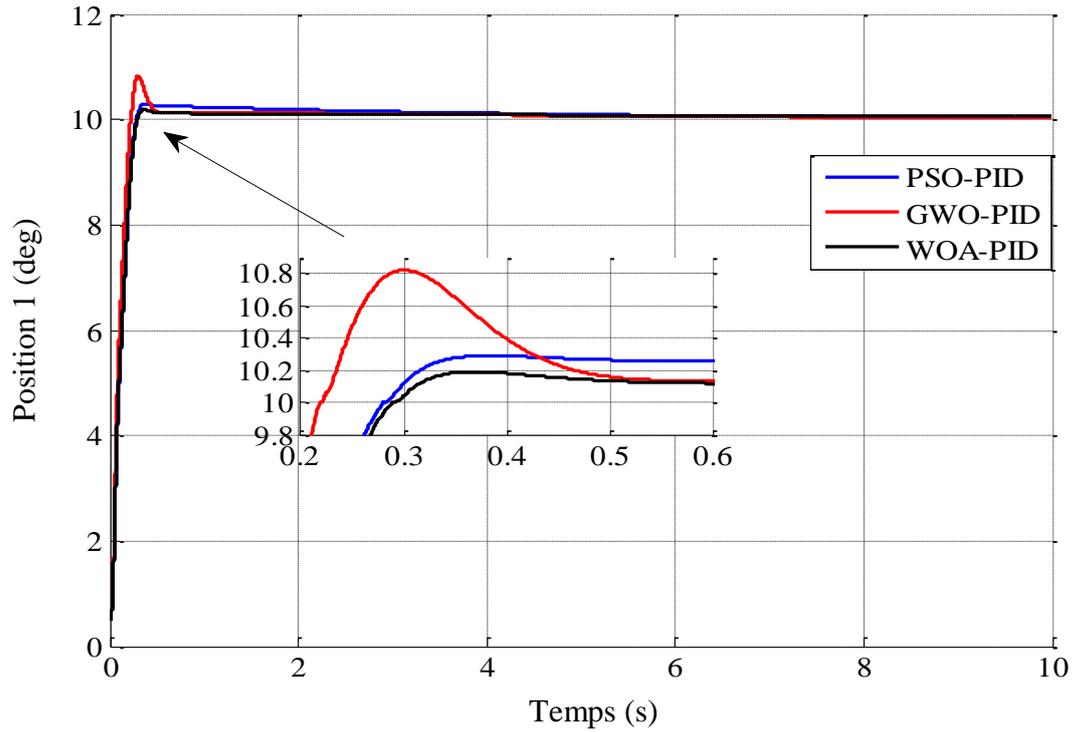


(b) Deuxième articulation.

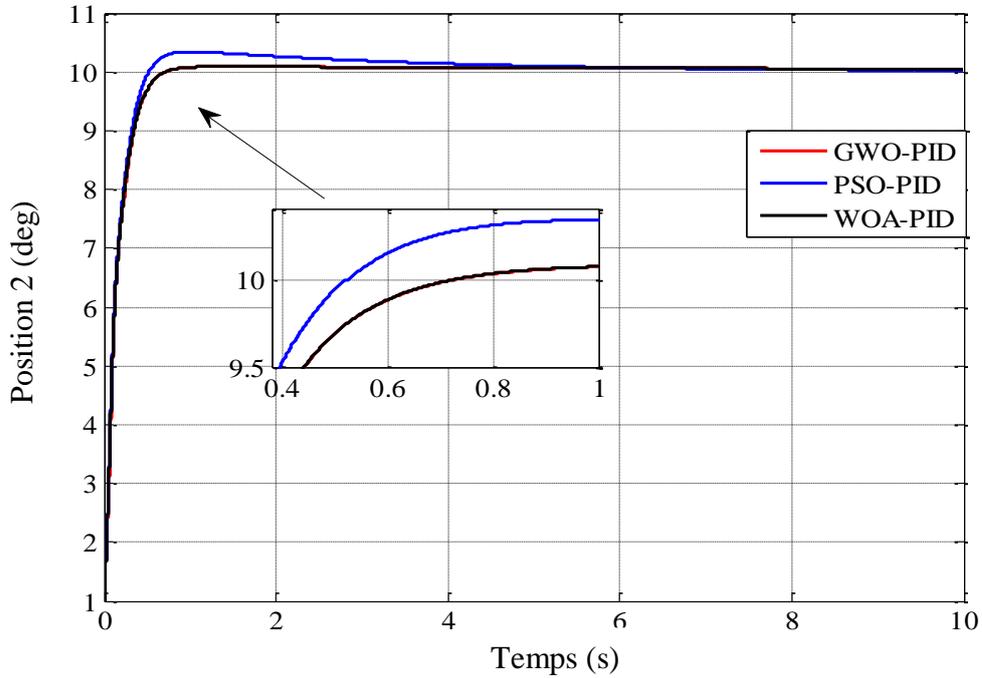
Figure 4.5 : Position actuelle du robot manipulateur contrôlé par PID-WOA.

4.2.2 Commande PID du robot manipulateur

Pour mieux voir la contribution de WOA pour un suivi de trajectoire rapide et précis du robot manipulateur. On a appliqué le contrôle PID réglé par les trois algorithmes, utilisé les meilleures performances des contrôleurs obtenues par 200 agents pour le PSO, 20 agents pour GWO et 30 agents pour WOA, respectivement. La figure 4.6 montre d'une part la supériorité des contrôleurs WOA-PID et GWO-PID par rapport au contrôleur PSO-PID. D'autre part nous pouvons constater une amélioration des performances de la première articulation du robot avec l'algorithme WOA. Ce dernier offre un temps de réponse optimal de la première articulation par rapport à ceux obtenu avec le réglage GWO-PID (0.2728 s / 0.453 s) et réduit le dépassement de (7.8466% à 1.3703%). On peut conclure que les performances du système sont satisfaites en termes de temps de réponse et les réponses du système sont plus rapides avec le contrôleur WOA-PID. La figure 4.7(a) montre la vitesse de convergence de l'erreur résultante de l'algorithme WOA qui confirme la compétitivité de cette méthodologie alors que la figure 4.7(b) montre la convergence de WOA vers l'optimum avec un nombre d'itérations plus petit.

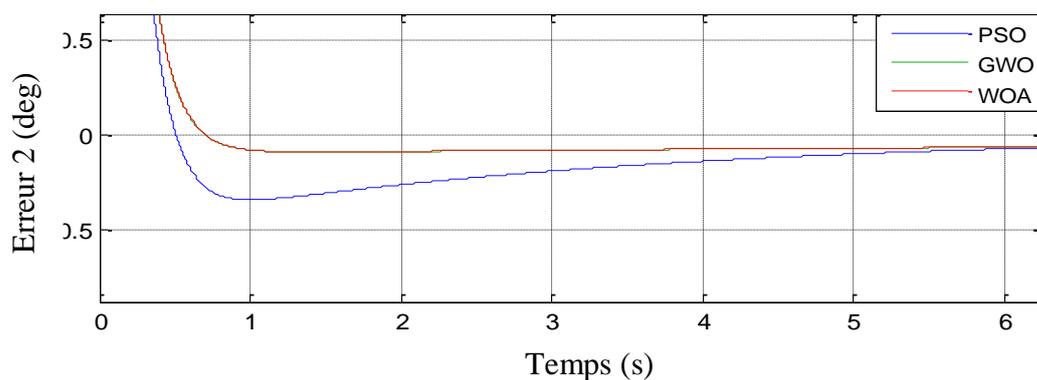
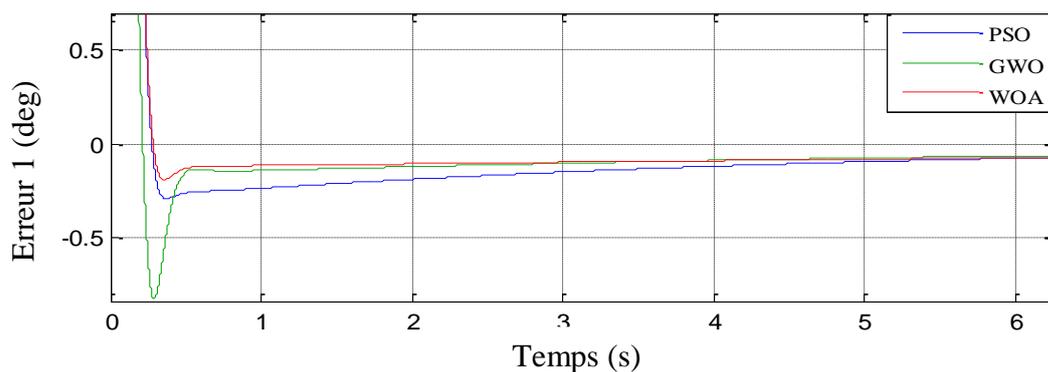


(a) Première articulation.

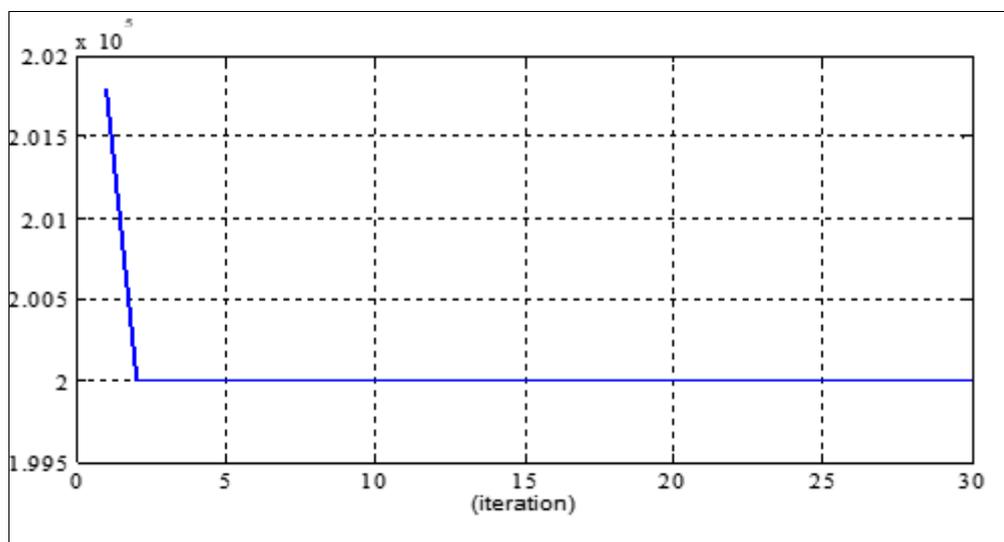


(b) Deuxième articulation.

Figure 4.6 : Position actuelle du robot manipulateur contrôlé par PID optimisé par les trois algorithmes.



(a)



(b)

Figure 4.7 : (a) Signal d'erreur pour la première et la deuxième articulation. (b) Courbe de Convergence de l'algorithme WOA.

4.2.3 Tests de Robustesses

Pour examiner la robustesse de l'algorithme d'optimisation, le contrôleur obtenu a été testé tout d'abord en changeant l'entrée du signal sinusoïdal puis en introduisant une perturbation de couple pour les deux articulations

- A. Suivi de l'entrée sinus : L'optimisation a été effectuée pour un signal d'entrée sinusoïdal, les gains obtenus du contrôleur PID permettent de poursuivre la trajectoire comme l'indique la figure 4.9(a) et le couple de commande généré sur la figure 4.9(b).
- B. Couple de perturbation : L'objectif est de tester la robustesse du contrôleur WOA-PID lorsque le robot est soumis à des perturbations. Un couple de bruit blanc est ajouté au signal de commande PID avec une variance de 0.8 représenté sur la figure 4.10(b). La figure 4.10(a) montre le suivi de la trajectoire souhaitée du robot contrôlé par le PID en présence des perturbations externes. La figure 4.8 révèle la fonction de coût (*ITAE*) pour les deux cas d'optimisations.

Le but de ce travail est de présenter une nouvelle application de l'algorithme d'optimiseur de baleine pour ajuster les paramètres PID pour le contrôle de suivi du robot manipulateur 2ddl. Les résultats obtenus témoignent l'efficacité de WOA-PID proposé en termes de temps de réponse, de précision et de temps de convergence ainsi que sa robustesse pour régler les paramètres du contrôleur PID pour le contrôle de suivi du robot avec ou sans perturbations.

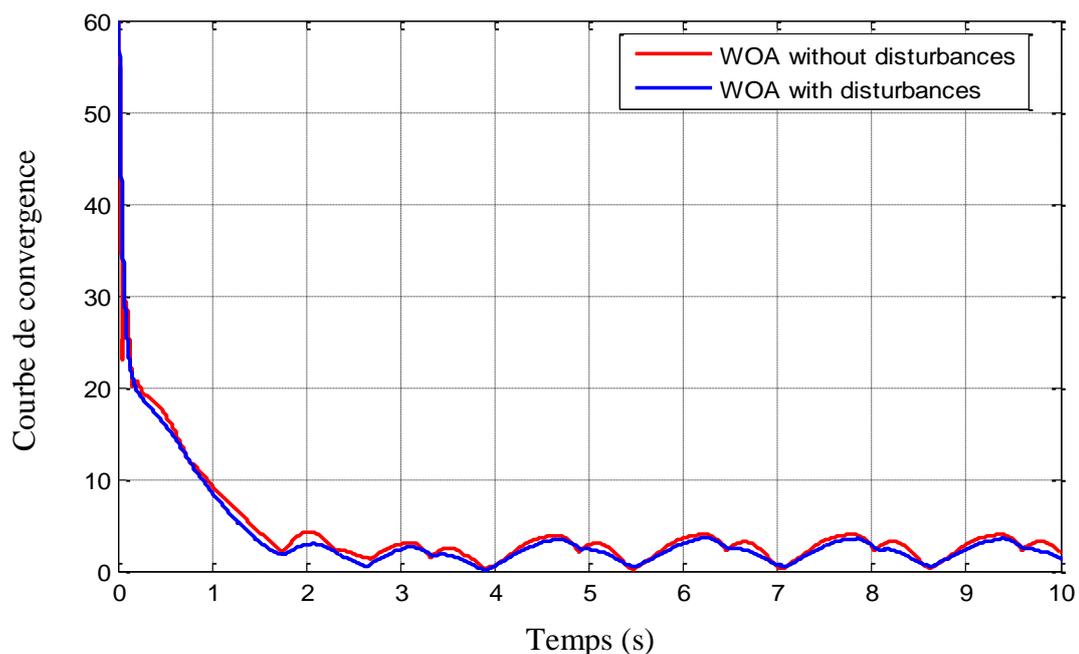


Figure 4.8 : Fonction de coût de l'algorithme WOA.

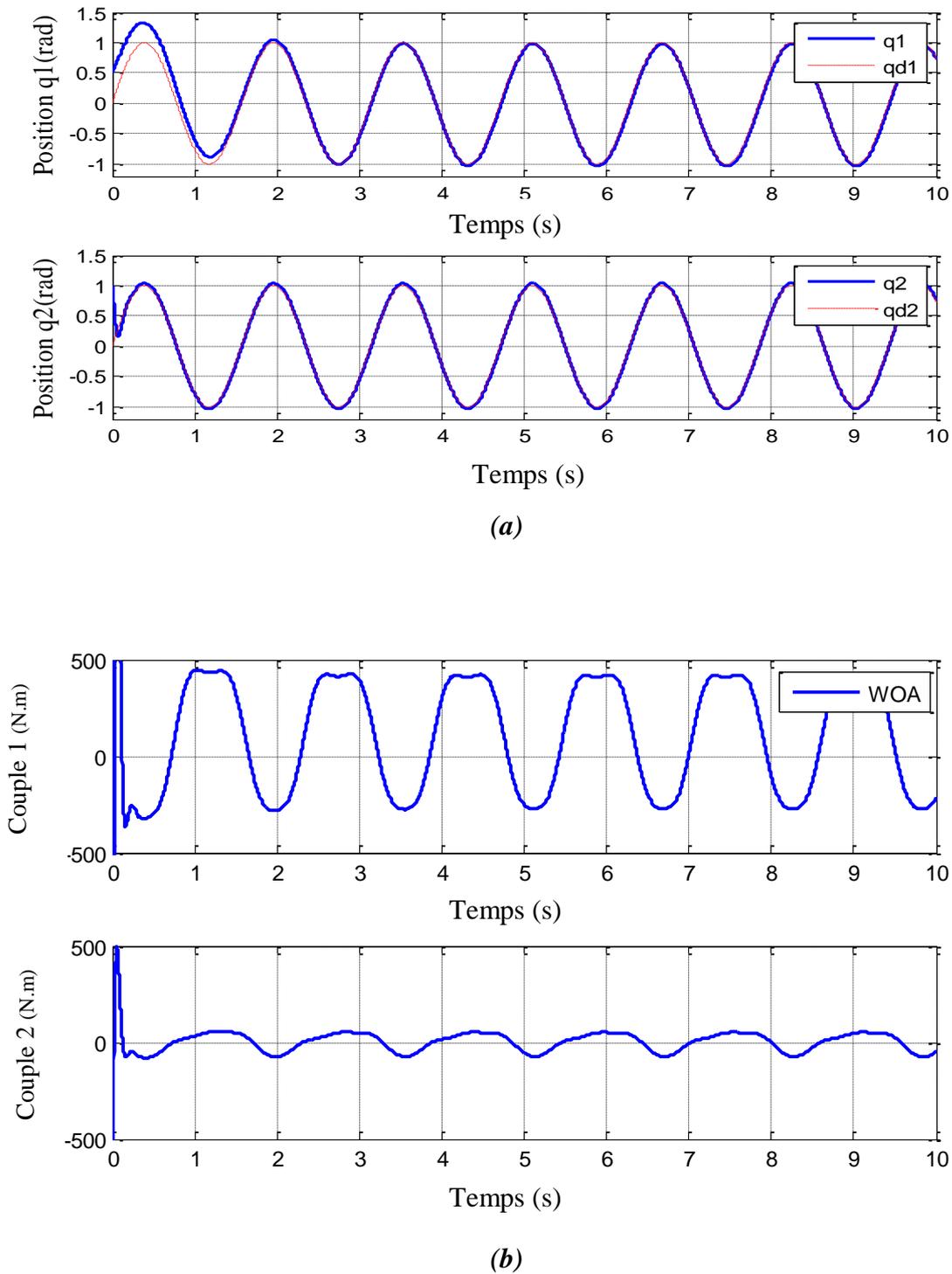


Figure 4.9 : (a) Position de la première et la deuxième articulation sans perturbation, (b) Couple de contrôle.

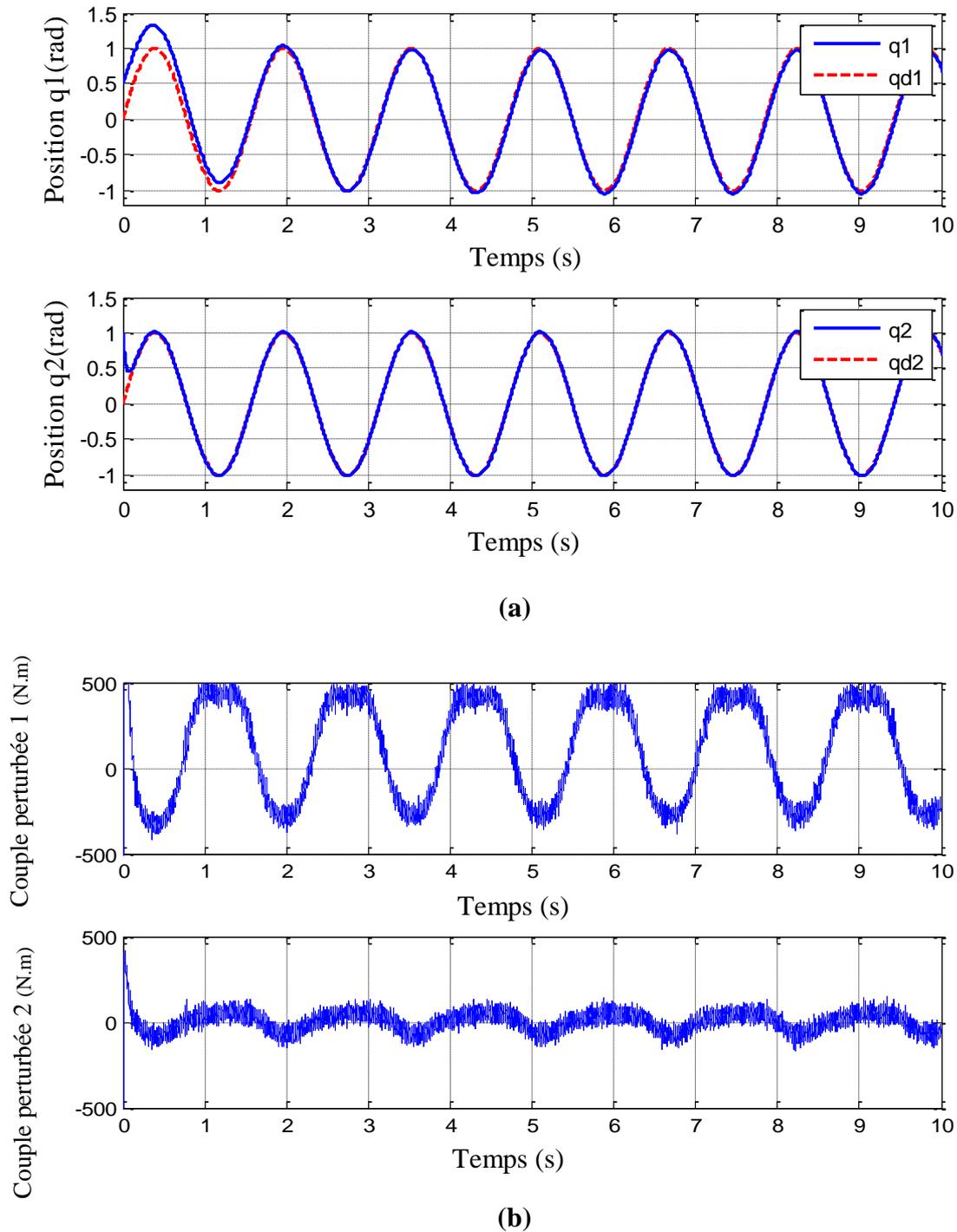


Figure 4.10 : (a) Position de la première et la deuxième articulation avec perturbation (variance 0.8), (b) Couple de contrôle.

4.3 Commande à Couple Calculé

Les méthodes conventionnelles du contrôle d'un système non linéaire est basé sur les modèles, et particulièrement dans le domaine du contrôle de robot. De nombreux systèmes de contrôle

robotique peuvent être considérés comme des cas particuliers de contrôle à modèles appelé approche du couple calculé, [42]. Le concept de base du couple calculé est de linéariser un système non linéaire, puis appliquer la théorie du contrôle linéaire. Certains robots manipulateurs sont contrôlés par des contrôleurs PID linéaires, mais la conception du contrôleur linéaire pour un robot manipulateur est extrêmement difficile parce que ce système est non linéaire et incertain. Réduire le défis ci-dessus, le contrôle des systèmes non linéaire par un contrôleur non linéaire robuste est nécessaire. Le contrôleur de couple calculé (CTC) est un contrôleur non linéaire puissant largement utilisé dans le contrôle du robot manipulateur. Ce contrôleur fonctionne très bien quand tous les paramètres dynamique et physiques sont connus. En pratique, la plupart des systèmes physiques tel que les robots manipulateurs par exemple sont incertain ou variant dans le temps, donc, un contrôleur à couple calculé est utilisé pour compenser l'équation dynamique du robot manipulateur [43]. Le contrôle des bras manipulateurs utilisant des contrôleurs à base de modèle fait appel au modèle dynamique du manipulateur. Ces contrôleurs ont souvent beaucoup de problèmes lors de la conception. Les contrôleurs conventionnels exigent des informations précises du modèle dynamique du robot manipulateur, mais la plupart du temps ces modèles sont MIMO, non linéaires et partiellement incertains et donc déterminer un modèle dynamique précis deviens une tache compliquée.

4.3.1 Conception du contrôleur à couple calculé

Le contrôleur à couple calculé est utilisé pour équilibrer l'équation non linéaire du mouvement du robot par annulation partielle ou totale, des termes non linéaires. Alors, le système soit découplé et linéarisé. Pour cela, le modèle dynamique du manipulateur est exploité.

$$M(q)\ddot{q} + N(q, \dot{q}) + \tau_d = \tau \quad (4.4)$$

τ_d présente un couple perturbateur et $N(q, \dot{q}) = H(q, \dot{q}) + F(\dot{q}) + G(q)$.

L'erreur de poursuite de position du robot manipulateur est :

$$e(t) = q_d(t) - q \quad (4.5)$$

et d'accélération:

$$\ddot{e}(t) = \ddot{q}_d(t) - \ddot{q}(t) \quad (4.6)$$

Résoudre maintenant $\ddot{q}(t)$ de (4.4) et substituer dans l'équation (4.6) on aboutit à :

$$\ddot{e}(t) = \ddot{q}_d(t) + M^{-1}(N(q, \dot{q}) + \tau_d - \tau) \quad (4.7)$$

$$u(t) = \ddot{q}_d(t) + M^{-1}(N(q, \dot{q}) - \tau) \quad (4.8)$$

$$w = M^{-1}\tau_d \quad (4.9)$$

Considérons le vecteur d'état :

$$x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \quad (4.10)$$

Un modèle d'état linéaire de la forme :

$$\dot{x} = Ax + Bu \quad (4.11)$$

peut être défini comme :

$$\dot{x} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ I \end{bmatrix} u \quad (4.12)$$

Ceci est connu sous le nom de « forme canonique de Brunousky ». Cette forme peut être réécrite sous :

$$\frac{d}{dt} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u \quad (\text{Sans perturbation}). \quad (4.13)$$

$$\frac{d}{dt} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u + \begin{bmatrix} 0 \\ I \end{bmatrix} w \quad (\text{Avec perturbation}). \quad (4.14)$$

La transformation linéarisée à rétroaction (4.8) peut être inversée pour donner :

$$\tau(t) = M(\ddot{q}_d(t) - u(t)) + N(q, \dot{q}) \quad (4.15)$$

Ceci est une loi de contrôle de rétroaction non linéaire qui garantit le suivi de trajectoire du robot manipulateur. Il est important de réaliser que le couple calculé dépend de l'inversion de la dynamique du robot, qu'en appelle parfois contrôle de la dynamique inverse.

4.3.2 Conception de boucle extérieure type PD (commande à couple calculé type PD)

Une façon de choisir le signal de commande auxiliaire $u(t)$ est la rétroaction proportionnelle dérivée négative (PD) [1] :

$$u(t) = -K_d \dot{e}(t) - K_p e(t) \quad (4.16)$$

Où k_p, k_d sont les gains du contrôleur.

La sélection de la rétroaction proportionnelle, dérivée (PD) pour $u(t)$ entraîne le PD-CTC [1] :

$$\tau = M \left(\ddot{q}_d(t) + K_d \dot{e}(t) + K_p e(t) \right) + N(q, \dot{q}) \quad (4.17)$$

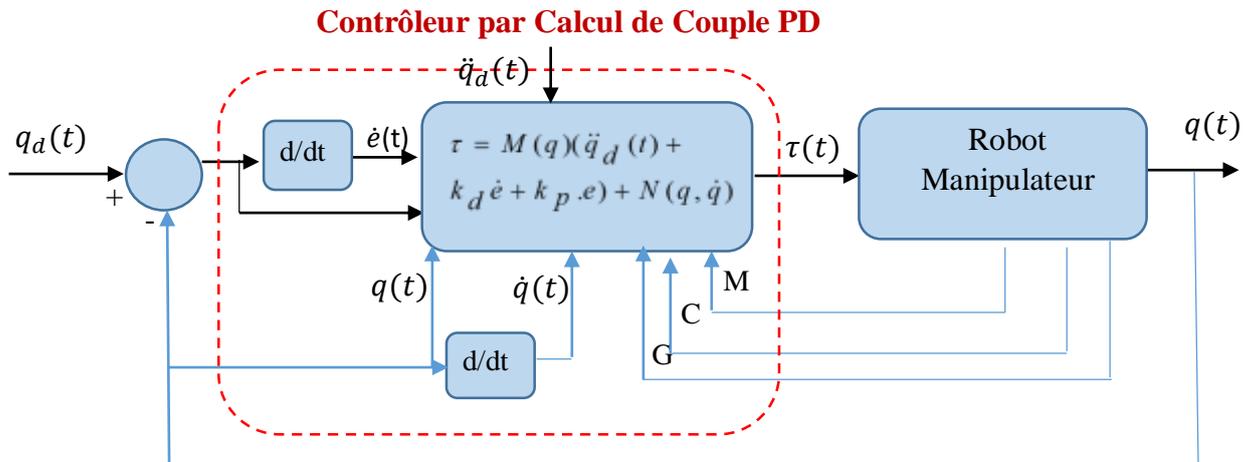


Figure 4.11 : Mécanisme de Couple calculé type PD.

Selon la théorie des systèmes linéaires, la convergence de l'erreur de suivi à zéro est garanti, et la dynamique de l'erreur en boucle fermée est régie par l'expression suivante :

$$\ddot{e}(t) + K_d \dot{e}(t) + K_p e(t) = 0 \quad (\text{Sans perturbation}). \quad (4.18)$$

$$\ddot{e}(t) + K_d \dot{e}(t) + K_p e(t) = w \quad (\text{Avec perturbation}). \quad (4.19)$$

Ou sous forme d'état :

$$\frac{d}{dt} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -K_p & -K_d \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} w \quad (4.20)$$

Le polynôme caractéristique en boucle fermée est :

$$\Delta_c(p) = |p^2 I + K_d p + K_p| \quad (4.21)$$

4.3.2.1 Choix des gains PD : Il est habituel de prendre la diagonale des matrices de gain $n \times n$

Alors :

$$K_d = \text{diag}\{k_{di}\}, \quad K_p = \text{diag}\{k_{pi}\},$$

$$\Delta_c(p) = \prod_{i=1}^n (p^2 I + k_{di} p + k_{pi})$$

Et le système d'erreur est asymptotiquement stable tant que le k_{di} et le k_{pi} sont tous positifs.

Par conséquent, tant que la perturbation $w(t)$ est limitée, il est de même pour l'erreur $e(t)$.

Dans le même cadre, examiner (4.9) et rappeler que M^{-1} a une borne supérieure. Ainsi, la limite

de $w(t)$ équivaut à limite de τ_d . Il est important de noter que bien que la sélection de la diagonale des matrices de gain PD entraîne un contrôle découplé au niveau de la boucle externe, il ne se traduit pas par une stratégie de contrôle conjoint découplée. En effet, la multiplication par $M(q)$ et l'ajout des termes à action directe non linéaire $N(q, \dot{q})$ dans la boucle interne brouille le signal $u(t)$ pour toutes les articulations. Ainsi, la connaissance de toutes les positions communes $q(t)$ et les vitesses $\dot{q}(t)$ sont généralement nécessaires pour calculer le contrôle $u(t)$ pour toute articulation donnée. Beaucoup de détails sur le choix des paramètres de la commande PD à couple calculé et d'autres techniques de commande du robot manipulateur ont été présenté dans [1].

4.3.2.2 Résultats de simulation

La commande par calcul de couple type PD est effectuée pour commander un robot à 2 ddl soumis à un couple perturbateur 0.8 de puissance. En premier temps, nous avons appliqué une entrée sinusoïdale d'amplitude 1 (rad) puis une entrée constante d'amplitude 1(deg). Pour obtenir les meilleurs paramètres du contrôleur, l'algorithme d'optimisation WOA a été utilisé en minimisant un critère de cout, ISTE, dont les résultats obtenus sont présentés ci-dessous :

i. Pour une commande sinusoïdale

Après l'étape d'optimisation les gains du contrôleur obtenus sont :

$$k_p = \begin{bmatrix} 3000 & 0 \\ 0 & 87.502879 \end{bmatrix} \text{ et } k_d = \begin{bmatrix} 5000 & 0 \\ 0 & 337.88196 \end{bmatrix}$$

Tableau 4.6 : Caractéristique de la réponse du robot à la commande couple calculée type PD.

ISTE1	0.0001504	Erreur statique 1	$[-2.75 * 10^{-3}, 1.5 * 10^{-3}]$
ISTE2	0.003449	Erreur statique 2	$[-1.25 * 10^{-2}, 0.875 * 10^{-2}]$

La figure 4.12 montre la position des deux articulations du robot ; et la figure 4.13 illustre les erreurs de poursuite. Les conditions initiales entraînent une erreur initiale qui s'annule presque dans 0.25 s pour la première articulation et 0.75 s pour la deuxième articulation, puis elle fluctue avec des valeurs faibles dans l'intervalle $[-2.75 * 10^{-3}, 1.5 * 10^{-3}]$ et $[-0.0125, 0.00875]$ pour la première et la deuxième articulation respectivement . La figure 4.14 montre le couple de commande engendrer par les moteurs des articulations. Le couple le plus élevé correspond au moteur intérieur, qui doit déplacer deux liens.

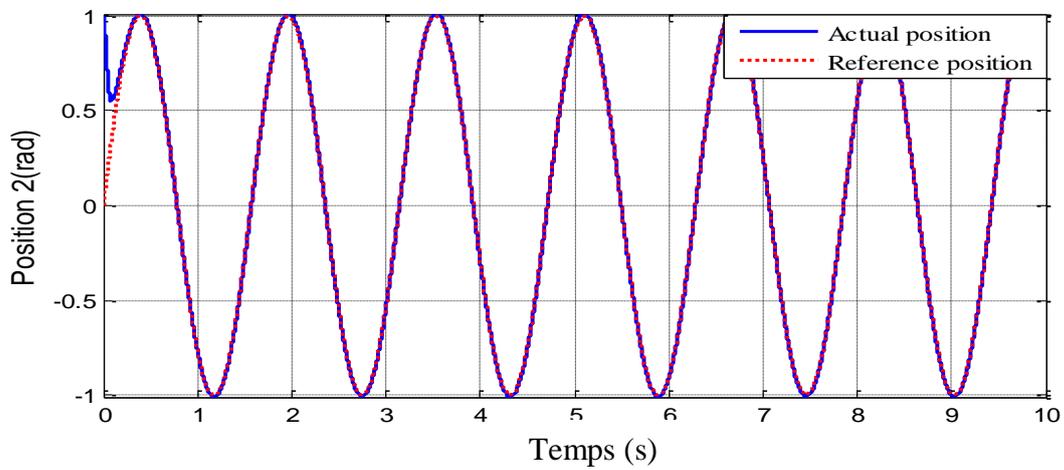
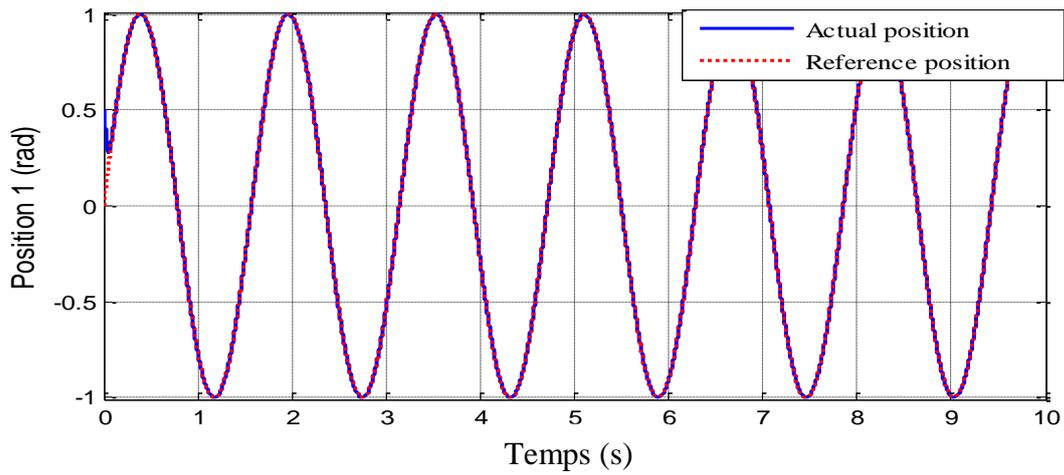
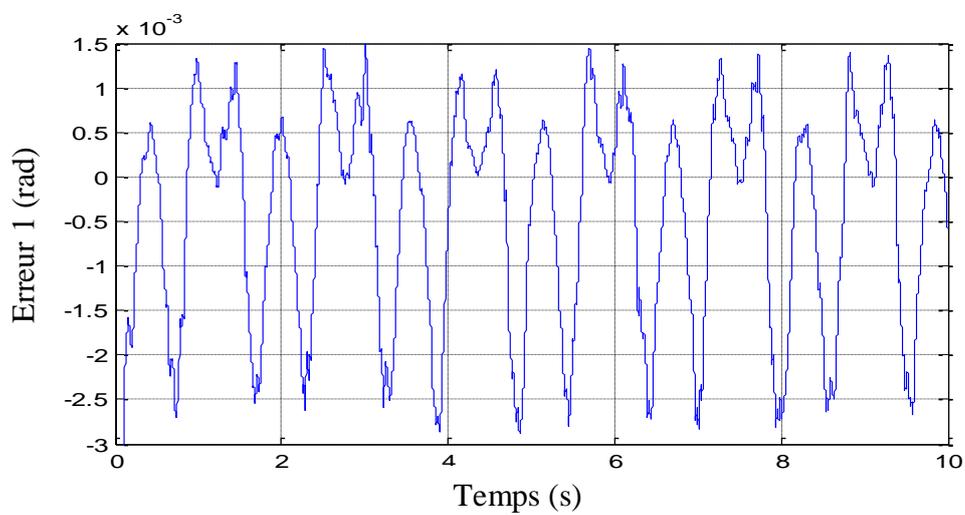


Figure 4.12 : Positions des deux articulations commandées par couple calculé type PD pour une entrée sinusoïdale.



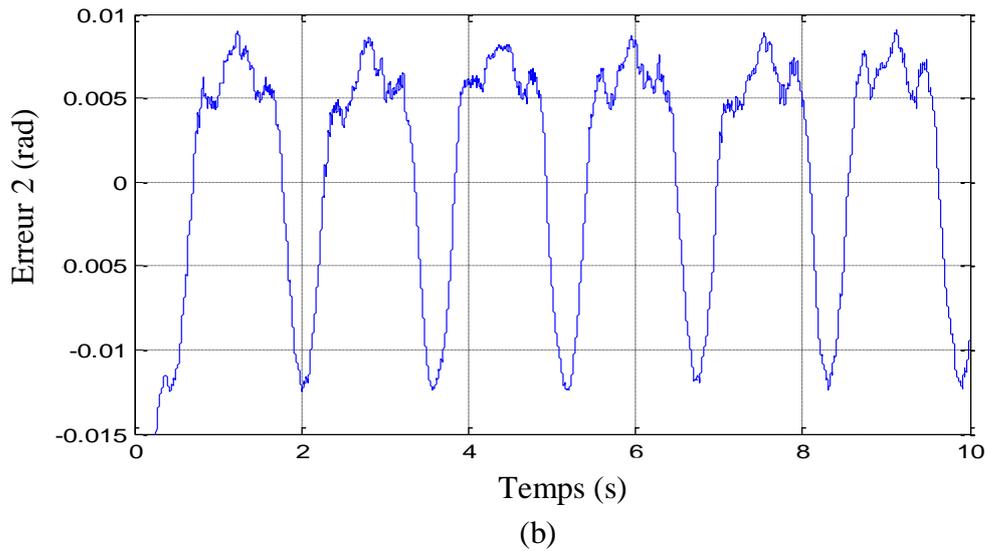


Figure 4.13 : Erreur de position des deux articulations.

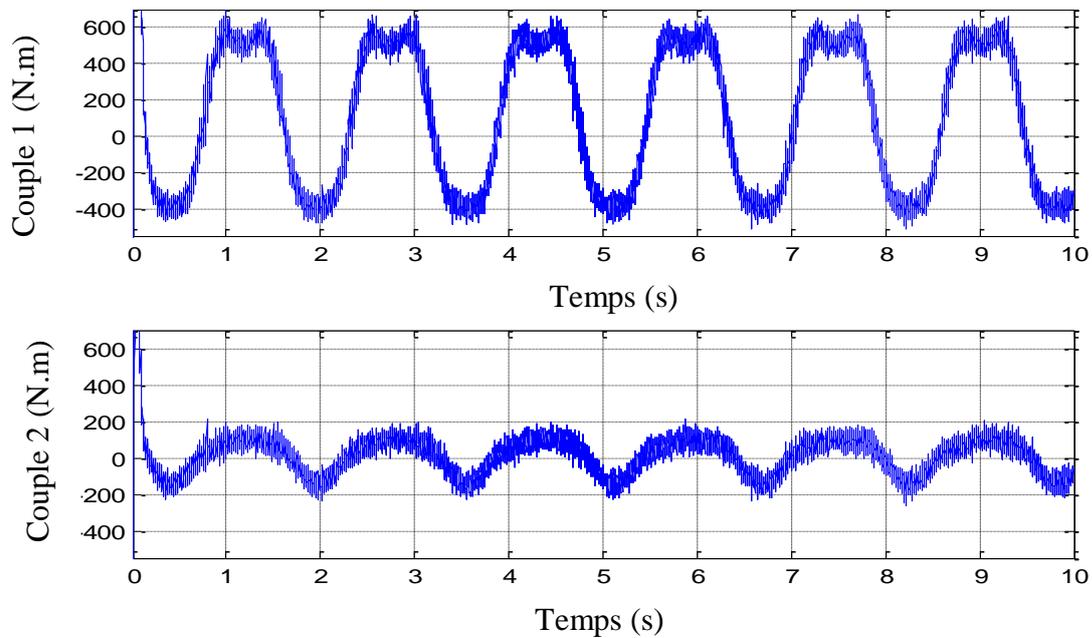


Figure 4.14 : Commande par Calcul de couple type PD d'un Robot a 2 ddl.

ii. Pour une commande constante

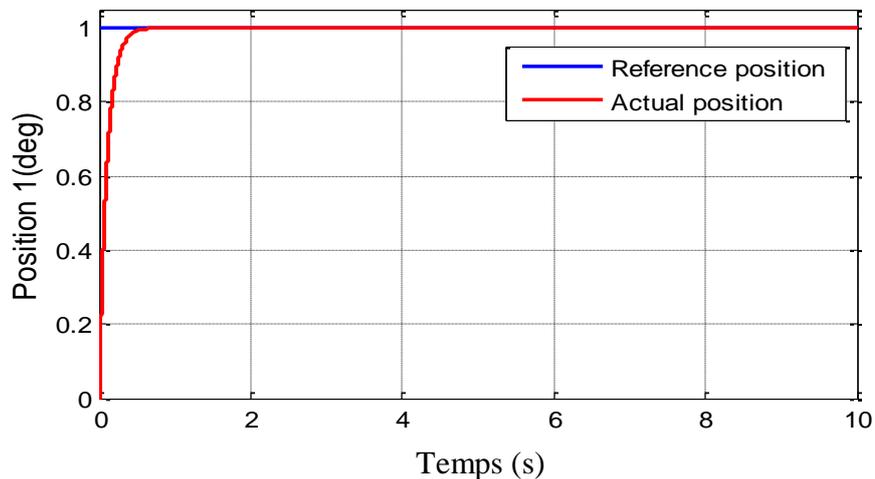
Les gains du contrôleur obtenus par optimisation sont :

$$k_p = \begin{bmatrix} 9000 & 0 \\ 0 & 9000 \end{bmatrix} \text{ et } k_d = \begin{bmatrix} 949.95598 & 0 \\ 0 & 511.24965 \end{bmatrix}$$

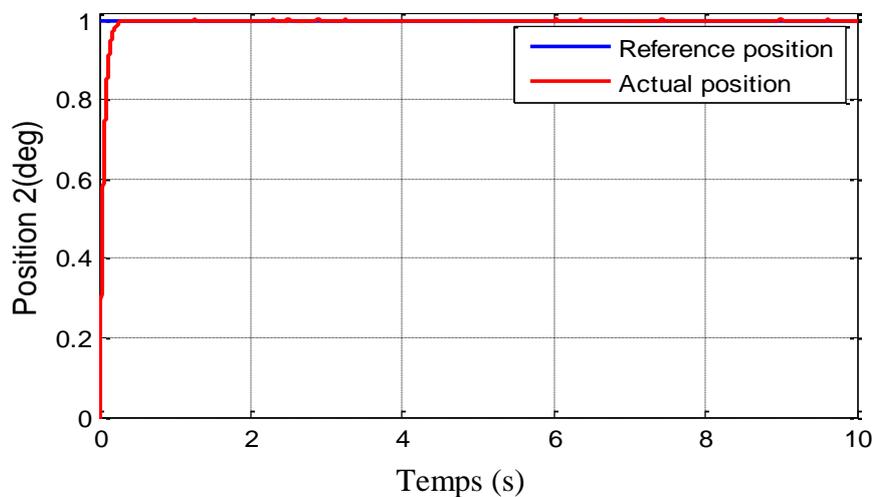
Tableau 4.7 : Caractéristique de la réponse du robot à la commande couple calculé type PD.

<i>tr1</i>	0.9999	<i>Dépassement1</i>	0.0058=0.58%	<i>ISTE1</i>	0.002729
<i>tr2</i>	1.0003	<i>Dépassement2</i>	0.0279=2.79%	<i>ISTE2</i>	0.0009634

Les résultats de simulation présentés par la figure 4.15 montrent le bon suivi de trajectoire du robot commandé par le couple calculé type PD, avec un temps de réponse de 0.9999 s pour la première articulation et 1.0003 s pour la deuxième articulation très surfaisant, et un dépassement relatif de 0.58% et 2.79% acceptable pour la première et la deuxième articulation respectivement, comme le montre le tableau 4.7.



(a)



(b)

Figure 4.15 : Position des deux articulations commandé par couple calculé type PD pour une entrée constante.

4.3.3 Conception de boucle extérieure type PID (commande à couple calculé type PID)

Nous venons de voir que le contrôle du couple calculé PD est très efficace si tous les paramètres du bras sont connus et en absence de perturbation τ_d . Cependant, de la théorie du contrôle classique, nous savons qu'en présence de perturbations constantes, le contrôle PD

présente une erreur en régime permanent non nulle. Par conséquent, nous sommes motivés à rendre le système de type 1 en incluant un intégrateur dans la boucle de rétroaction, cela peut être réalisé en utilisant le contrôleur de couple calculé PID [1].

Soit $\varepsilon(t)$ est l'intégrale de l'erreur de suivi $e(t)$:

$$\dot{\varepsilon}(t) = e(t) \tag{4.22}$$

$$u = -k_d \dot{e}(t) - k_p e(t) - k_i \varepsilon(t) \tag{4.23}$$

Ce qui donne le signal de contrôle du bras :

$$\tau = M(q) \left(\ddot{q}_d(t) + K_d \dot{e}(t) + K_p e(t) + K_i \varepsilon(t) \right) + N(q, \dot{q}) \tag{4.24}$$

Ainsi, des dynamiques supplémentaires ont été ajouté au compensateur de boucle externe linéaire. Cette loi de contrôle est décrite de manière pratique en définissant l'état comme :

$x = [\varepsilon^T \ e^T \ \dot{e}^T]^T \in \mathbb{R}^{3n}$ et la dynamique d'erreur (4.14) augmenté avec un intégrateur devient :

$$\frac{d}{dt} \begin{bmatrix} \varepsilon \\ e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \varepsilon \\ e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} w \tag{4.25}$$

Le schéma de principe du contrôleur de couple calculé PID est illustré sur la figure 4.16.

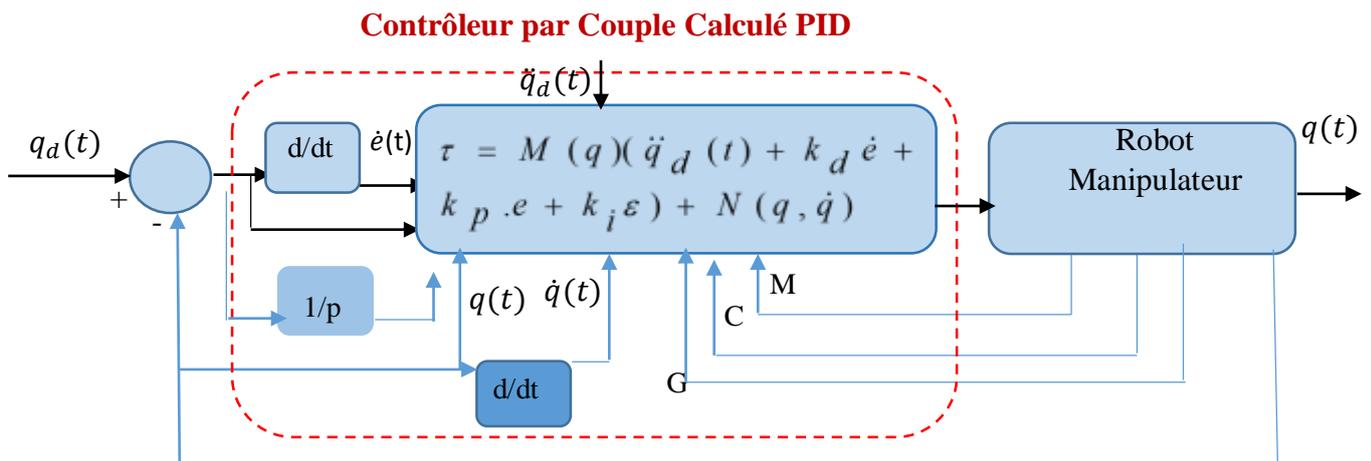


Figure 4.16 : Mécanisme de Couple calculé type PID.

Le système en boucle fermée est :

$$\frac{d}{dt} \begin{bmatrix} \varepsilon \\ e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I & 0 \\ 0 & 0 & I \\ -K_i & -K_p & -K_d \end{bmatrix} \begin{bmatrix} \varepsilon \\ e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} w \quad (4.26)$$

Le polynôme caractéristique en boucle fermée est :

$$\Delta_c(p) = |p^3 I + K_d p^2 + K_p p + K_i|$$

Choisissant les gains du contrôleur sous forme diagonal :

$K_d = \text{diag}\{k_{di}\}$, $K_p = \text{diag}\{k_{pi}\}$ et $K_i = \text{diag}\{k_{ii}\}$ donne :

$$\Delta_c(p) = \prod_{i=1}^n (p^3 I + K_d p^2 + K_p p + K_i) \quad (4.27)$$

En appliquant le test de Routh, on peut constater que le système en boucle fermée est stable si :

$$k_{ii} < k_{di} k_{pi} \quad (4.28)$$

C'est-à-dire que le gain intégral ne doit pas être trop grand.

4.3.3.1 Résultats de simulation

Dans la section précédente, nous avons simulé le contrôleur à couple calculé type PD pour un bras plan à 2ddl. Dans ce qui suit, nous ajoutons l'action intégrale au contrôleur et nous appliquons la commande par calcul de couple type PID dans les mêmes conditions. Ce contrôleur est optimisé par l'algorithme d'optimisation WOA dont les résultats obtenus sont détaillée ci-dessous :

i. Pour une commande sinusoïdale

Les gains optimisés du contrôleur sont :

$$k_p = \begin{bmatrix} 9000 & 0 \\ 0 & 9000 \end{bmatrix}, k_i = \begin{bmatrix} 300 & 0 \\ 0 & 364.27403 \end{bmatrix} \text{ et } k_d = \begin{bmatrix} 200 & 0 \\ 0 & 315.2231 \end{bmatrix}$$

Tableau 4.8 : Caractéristique de la réponse du robot à la commande couple calculé type PID.

ISTE1	4.677 e-05	Erreur statique 1	$[-6.4 * 10^{-4}, 8.8 * 10^{-4}]$
ISTE2	0.001196	Erreur statique 2	$[-6.3 * 10^{-3}, 6.3 * 10^{-3}]$

La position de chaque articulation du robot est configurée dans la figure 4.17 et la figure 4.18 illustre les erreurs de poursuite de position. Les conditions initiales entraîne une erreur initiale qui s'annule en presque 0.6 s pour la première articulation et 0.15 s pour la deuxième

articulation, et reste ultérieurement dans l'intervalle $[-6.4 * 10^{-4}, 8.8 * 10^{-4}]$ et $[-6.3 * 10^{-3}, 6.3 * 10^{-3}]$ pour la première et la deuxième articulation respectivement. On voit clairement la diminution de l'erreur de position des deux articulations ; et ainsi la commande par couple calculé type PID améliore la précision de la trajectoire du robot par rapport à la commande couple calculer type PD. Les couples de commande associés sont illustrés dans la figure 4.19, qui montre que les amplitudes de couple ne sont pas sensiblement augmentées en utilisant le terme intégral.

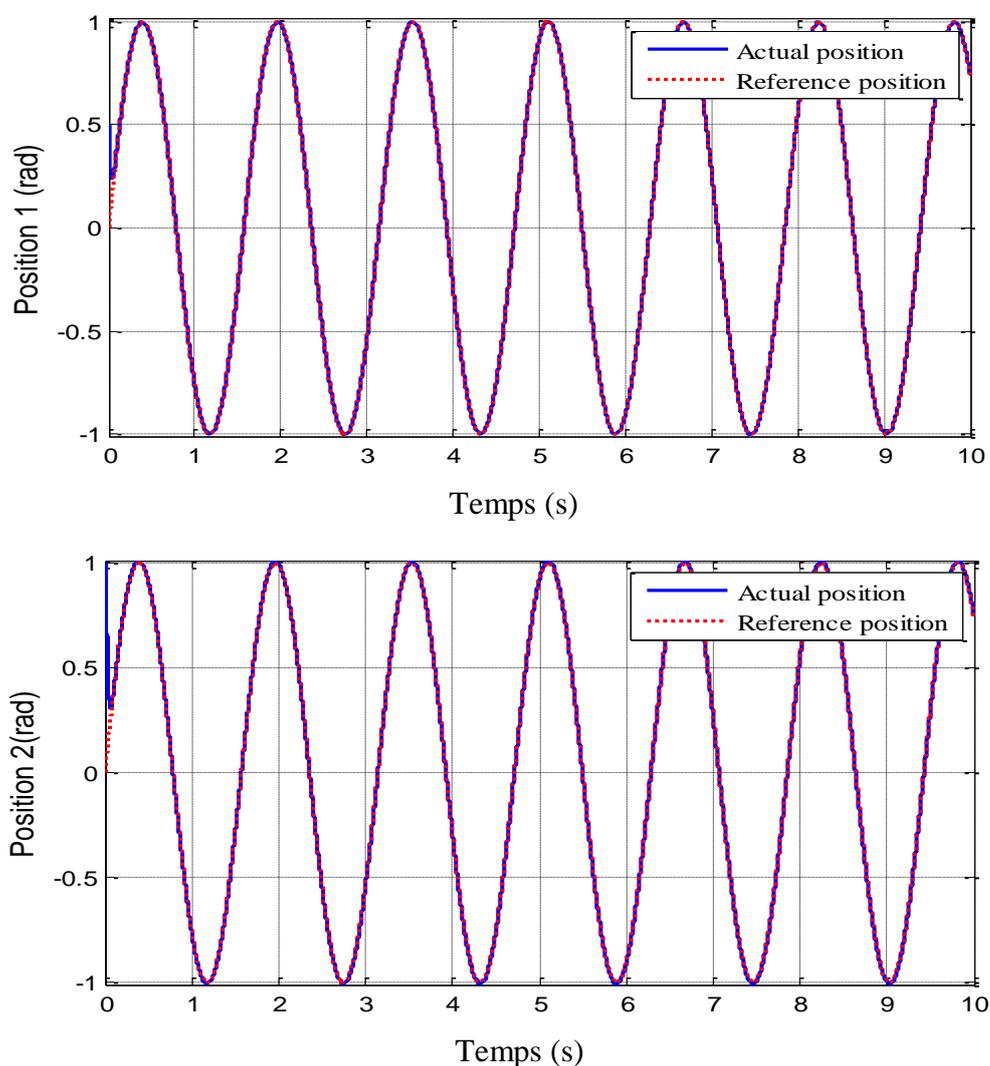


Figure 4.17 : Positions des deux articulations commandées par couple calculé type PID pour une entrée sinusoïdale.

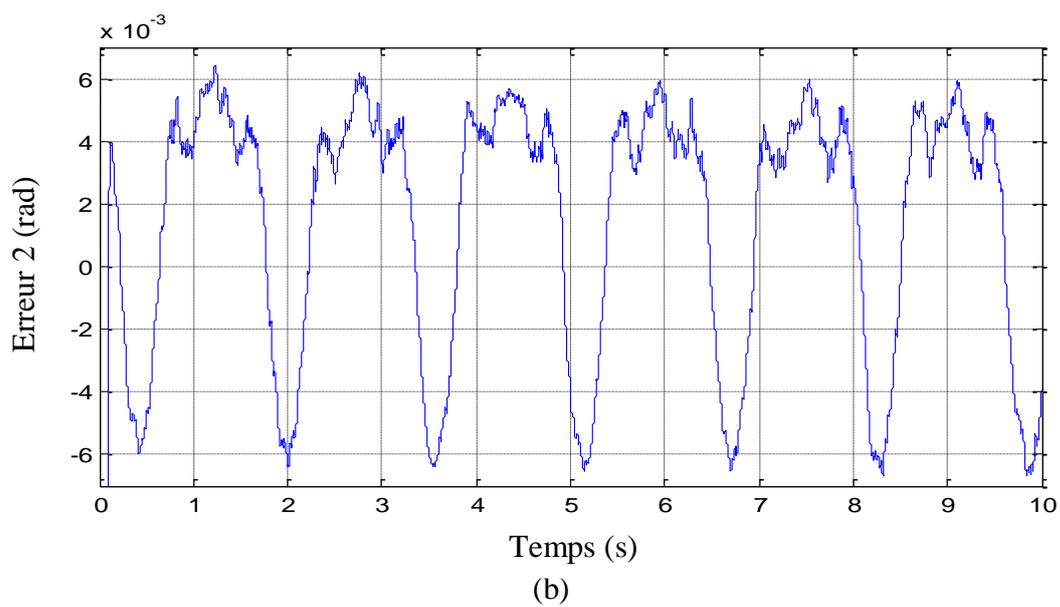
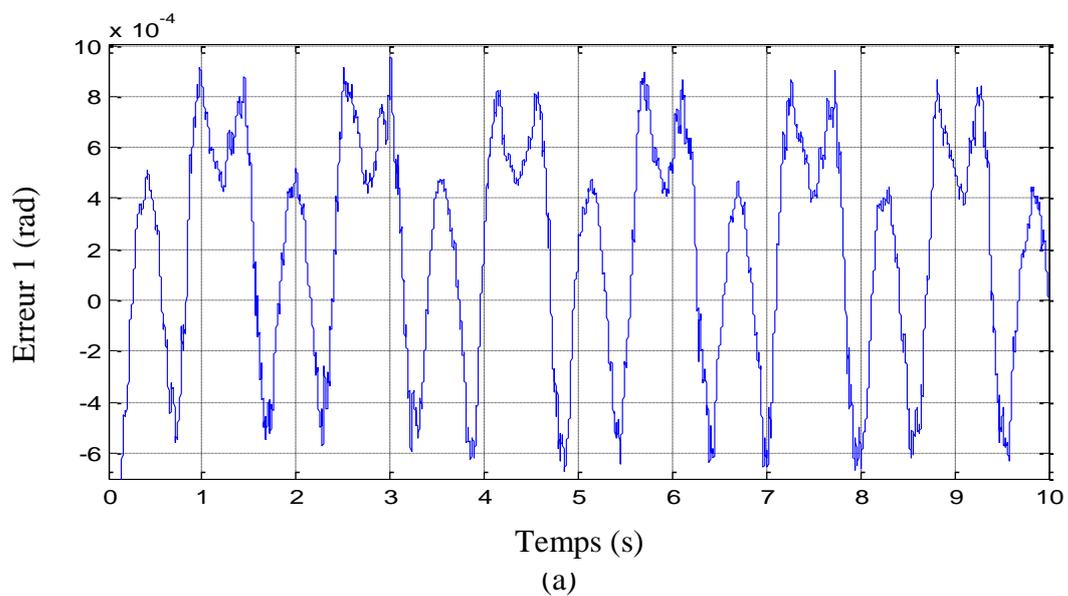


Figure 4.18 : Erreur de position des deux articulations.

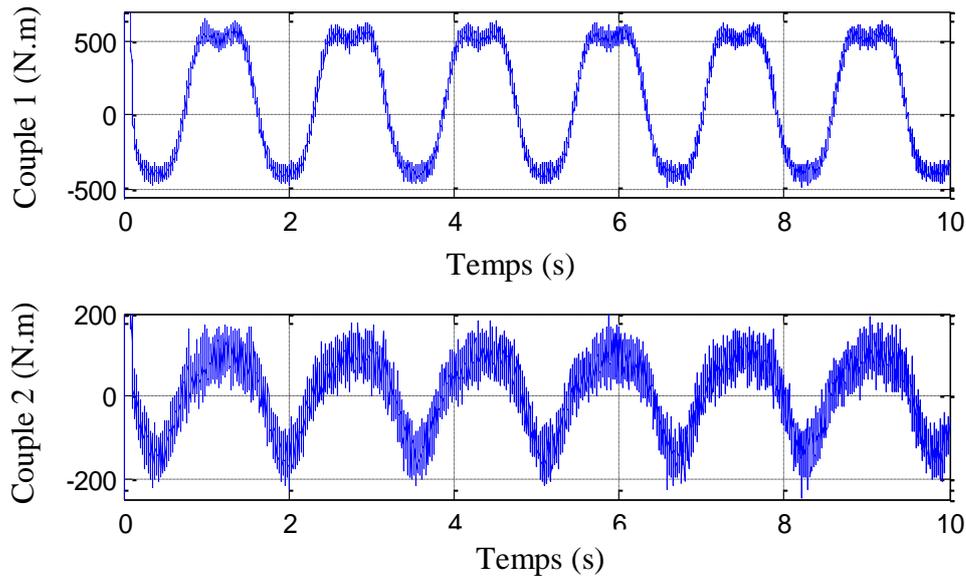


Figure 4.19 : Commande par Calcul de couple type PID d'un Robot à 2 ddl.

ii. Pour une commande constante

Après l'étape d'optimisation les gains du contrôleur obtenus sont :

$$k_p = \begin{bmatrix} 9000 & 0 \\ 0 & 9000 \end{bmatrix} k_i = \begin{bmatrix} 375.2418 & 0 \\ 0 & 325.25611 \end{bmatrix} \text{ et } k_d = \begin{bmatrix} 200 & 0 \\ 0 & 298.92233 \end{bmatrix}$$

Tableau 4.9 : Caractéristique de la réponse du robot à la commande couple calculé type PID.

tr1	1.0004 s	Dépassement1	0.0132=1.32%	ISTE1	0.0001334
tr2	1.0008 s	Dépassement2	0.0492=4.92%	ISTE2	0.000621

En utilisant le contrôleur couple calculer type PD, en paragraphe précédente, et à partir du graphique d'erreur de la figure 4.13, il apparaît un biais résiduel d'erreur de suivi en raison de la perturbation constante non modélisée, qui n'est pas prise en compte dans la loi de couple calculé. L'ajout d'un terme de pondération d'intégrale d'erreur donne un erreur permanent presque nulle qui est plutôt bien, avec un temps de réponse plus proche à celui de la commande type PD. Comme le montre le tableau 4.9, en vois que la commande couple calculer type PID provoque un dépassement de 1.32 % et de 4.92 % pour les deux articulations respectivement. Cela est due aux artefacts de l'intégrateur.

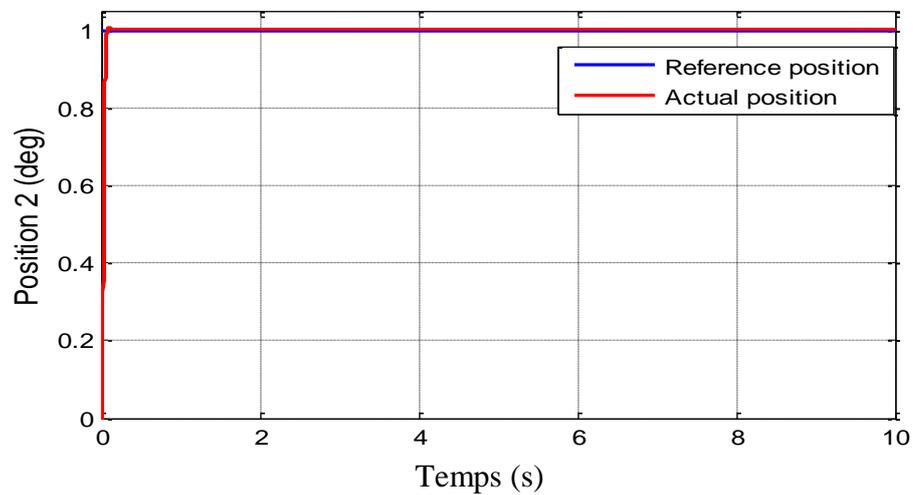
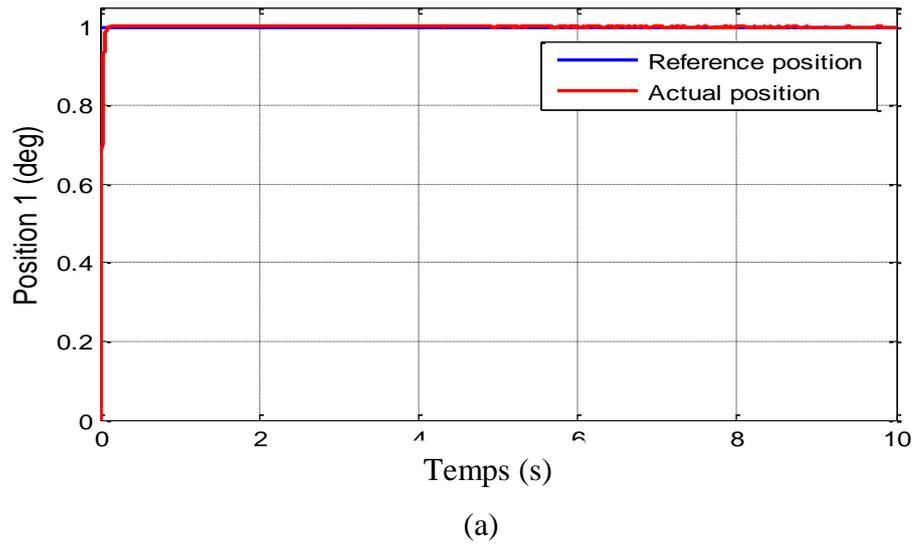


Figure 4.20 : Position des deux articulations commandé par couple calculé type PID pour une entrée constante.

4.4 Contrôleur de PD-Plus gravité

Un contrôleur utile dans la famille des couples calculés est le PD-plus-gravité avec $G(q)$ le terme de gravité de la dynamique du manipulateur. Ensuite, en choisissant la rétroaction PD pour $u(t)$ [1] :

$$\tau_c = K_v \dot{e} + K_p e + G(q) \quad (4.29)$$

Avec le terme de gravité pour un robot à 2ddl :

$$G(q) = \begin{bmatrix} (m_1 + m_2)gl_1 \cos q_1 + m_2 gl_2 \cos(q_1 + q_2) \\ m_2 gl_2 \cos(q_1 + q_2) \end{bmatrix}$$

Il y a des arguments forts même aujourd'hui par plusieurs chercheurs que ce système de contrôle est toujours approprié dans des implémentations pratique, et que les systèmes de contrôle non linéaires modernes sont trop compliqués pour des applications en robotiques industrielles. Cette loi de contrôle a été traitée par Arimoto et Miyazaki en 1984 [44], et Schilling en 1990, elle est beaucoup plus simple à mettre en œuvre que le contrôleur couple calculé exact. Lorsque le bras est au repos, les seuls termes non nuls dans la dynamique (4.4) sont la gravité $G(q)$, la perturbation τ_d et éventuellement le couple de commande τ .

Le contrôleur de gravité PD, comprend $G(q)$ de sorte que nous devrions nous attendre à une bonne performance pour le suivi du point de consigne, c'est-à-dire lorsqu'une constante q_d est donnée pour que $\dot{q}_d = 0$.

Théorème : Supposons que le contrôle de la gravité PD soit utilisé dans la dynamique du bras (4.4) et $\tau_d = 0$, $\dot{q}_d = 0$. Alors, l'erreur de suivi en régime permanent, $e = q_d - q$ est nul.

Preuve :

1. Système en boucle fermée : Ignorons le frottement, la dynamique du robot est donnée par :

$$\begin{aligned} M(q)\ddot{q} + H(q, \dot{q}) + G(q) &= \tau \\ M(q)\ddot{q} + C(q)\dot{q} + G(q) &= \tau \end{aligned} \quad (4.30)$$

Lorsque $\dot{q}_d = 0$, la loi de commande proposée (4.29) donne la dynamique en boucle fermée.

$$M\ddot{q} + C\dot{q} + K_v\dot{q} - K_p e = 0 \quad (4.31)$$

2. Fonction de Lyapunov : Considérons maintenant la fonction Lyapunov.

$$V = \frac{1}{2}(\dot{q}^T M \dot{q} + e^T K_p e) \quad (4.32)$$

Après dérivation, nous obtenons.

$$\dot{V} = \dot{q}^T \left(M\ddot{q} + \frac{1}{2}\dot{M}\dot{q} - K_p e \right) \quad (4.33)$$

La substitution de la dynamique en boucle fermée (4.31) aboutit à :

$$\dot{V} = \dot{q}^T \left(\frac{1}{2}\dot{M} - C \right) \dot{q} = \dot{q}^T K_v \dot{q} \quad (4.34)$$

Maintenant, l'antisymétrie du premier terme donne :

$$\dot{V} = -\dot{q}^T K_v \dot{q} \quad (4.35)$$

L'état est $x = [e^T \quad \dot{q}^T]^T$, de sorte que V est défini positif mais \dot{V} est semi-défini négatif.

Par conséquent, nous venons de démontrer la stabilité au sens de Lyapunov, c'est-à-dire que l'erreur et la vitesse conjointe sont toutes deux limitées.

3. Stabilité asymptotique par l'extension de LaSalle : Il est donc nécessaire de démontrer que le seul ensemble invariant contenu dans l'ensemble $V = 0$ est l'origine.

Puisque V a une borne inférieure de zéro et non positive, il s'ensuit que V s'approche d'une limite finie, qui peut être écrite

$$-\int_0^{\infty} \dot{V} dt = \text{const.} \quad (4.36)$$

Nous invoquons maintenant le Lemme de Barbalat pour montrer que V tend vers zéro. Pour cela, il est nécessaire de démontrer la continuité uniforme de V . Nous considérons que :

$$\dot{V} = -2\dot{q}^T K_v \ddot{q} \quad (4.37)$$

La stabilité démontrée montre que a et q sont bornés, d'où (4.31) et la limite de $M^{-1}(q)$ révéler que q est borné. Par conséquent, V est borné, d'où V est uniformément continu. Cela garantit par lemme de Barbalat que V va à zéro. Il est maintenant clair que \dot{q} va à zéro. Reste à montrer que l'erreur de suivi $e(t)$ disparaît.

Notez que lorsque $q = 0$, (4.4) révèle que :

$$\ddot{q} = M^{-1} K_p e \quad (4.38)$$

Par conséquent, un $e(t)$ non nulle aboutit à un \ddot{q} non nulle et donc à $q \neq 0$, une contradiction. Par conséquent, le seul ensemble invariant contenu dans $\{x(t) \mid V = 0\}$ est $x(t) = 0$. Ceci enfin démontre que $e(t)$ et disparaissent et conclut la preuve.

En général, si $q_d \neq 0$, alors le contrôleur de gravité PD garantit des erreurs de suivi limitées. La borne d'erreur diminue, c'est-à-dire les performances de suivi s'améliorent à mesure que les gains de PD augmentent.

4.4.1 Résultats de simulation

Pour illustrer les performances du contrôleur de gravité PD, ce dernier est appliqué pour le suivi de trajectoire d'un robot manipulateur à 2 ddl, décrite dans les sections précédentes. Les gains du contrôleur sont optimisés par l'algorithme d'optimisation WOA. Les résultats de simulation sont représentés ci-dessous :

i. Pour une commande sinusoïdale

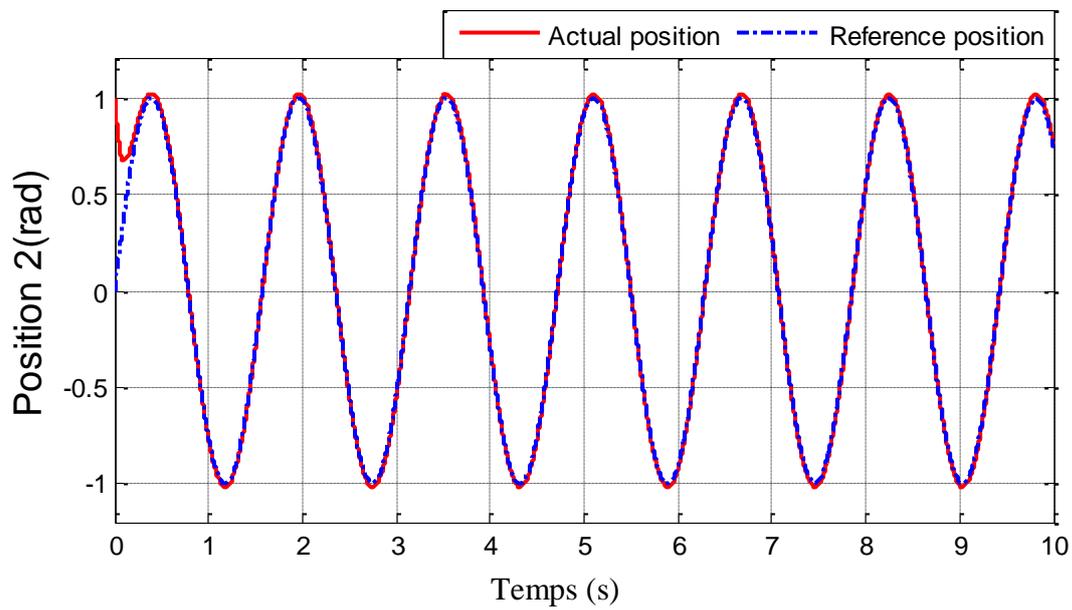
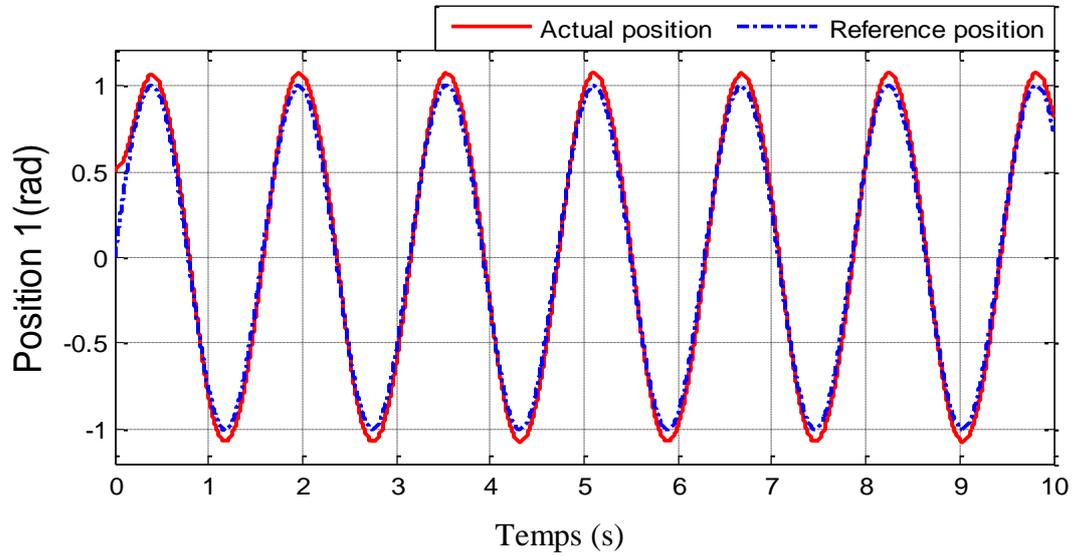
a) Sans perturbation :

Après l'étape d'optimisation les gains du contrôleur obtenus sont :

$$k_p = \begin{bmatrix} 6000 & 0 \\ 0 & 6000 \end{bmatrix} \text{ et } k_v = \begin{bmatrix} 751.13433 & 0 \\ 0 & 502.87301 \end{bmatrix}$$

Tableau 4.10 : Caractéristique de la réponse du robot à la commande PD plus gravité.

ISTE1	0.1844	Erreur statique 1	[-0.08, 0.08]
ISTE2	0.01167	Erreur statique 2	[-0.02, 0.0175]

**Figure 4.21** : Positions des deux articulations commandées par PD plus gravité pour une entrée sinusoïdale.

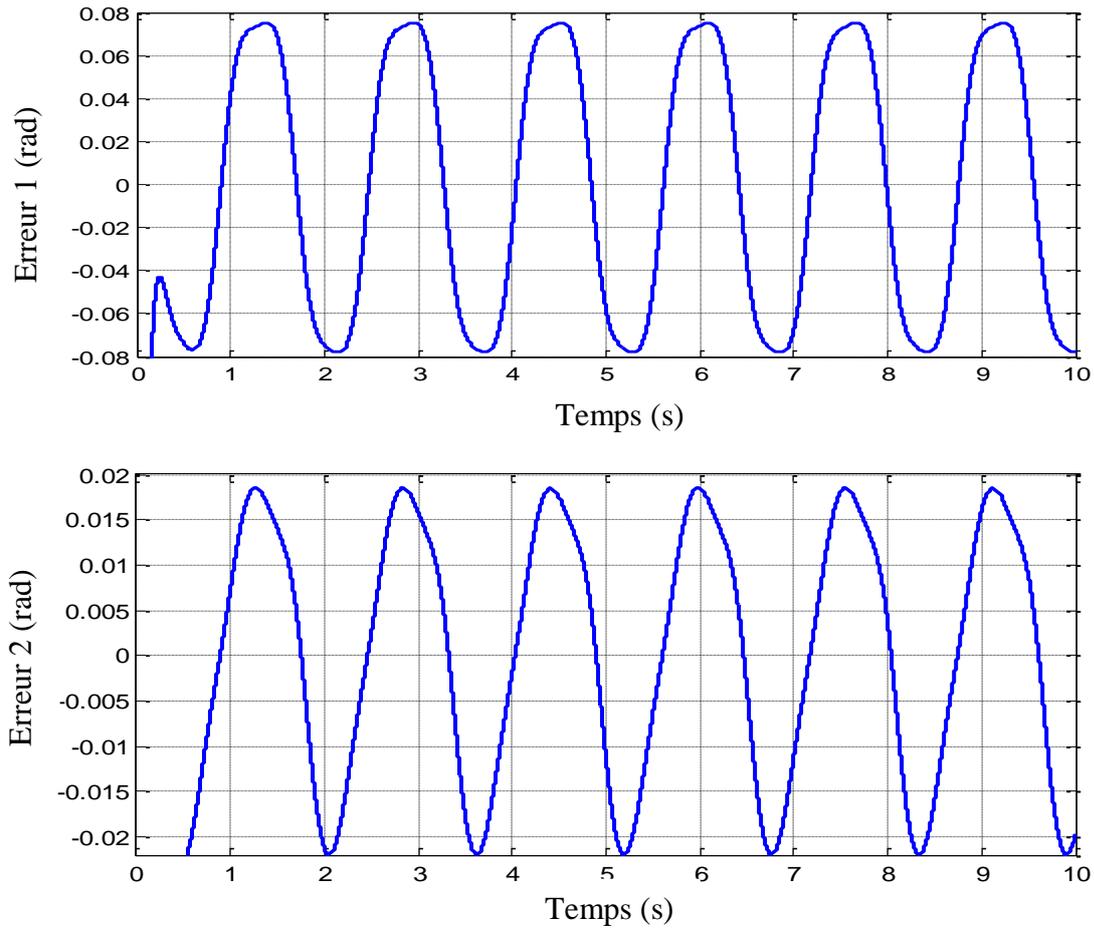


Figure 4.22 : Erreur de position des deux articulations.

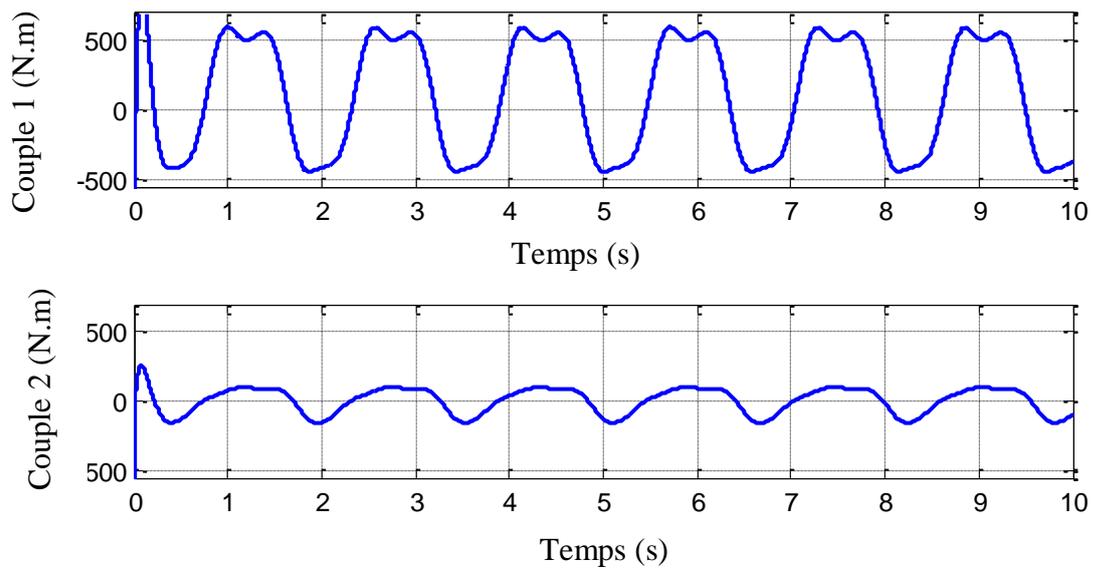


Figure 4.23 : Commande par PD plus gravité d'un Robot à 2 ddl.

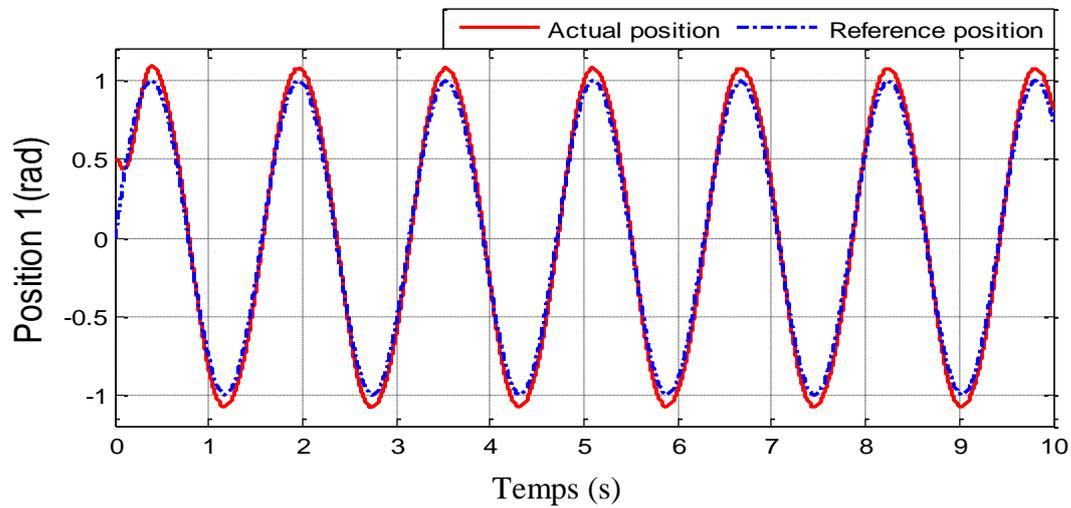
b) Avec perturbation :

Après l'étape d'optimisation les gains du contrôleur obtenus sont :

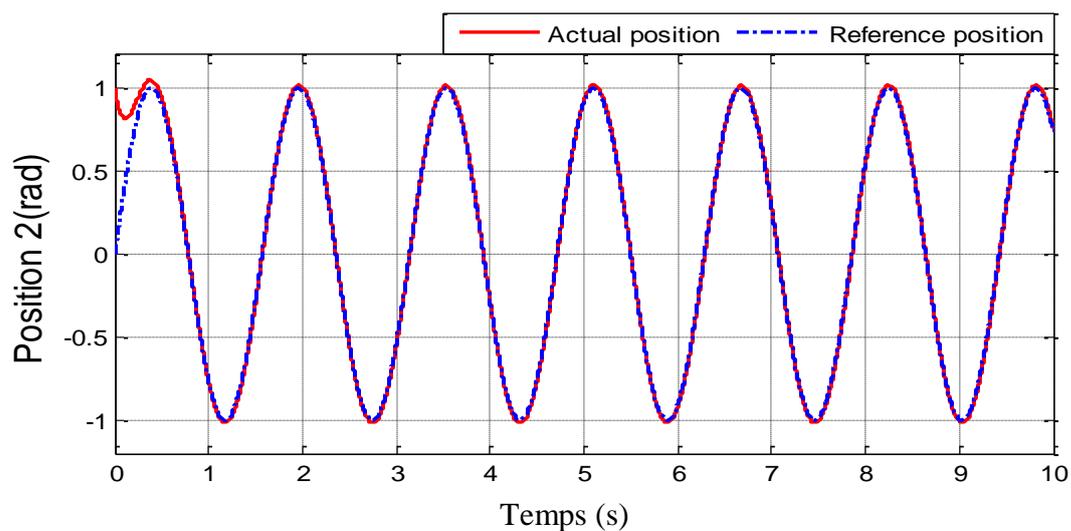
$$k_p = \begin{bmatrix} 6000 & 0 \\ 0 & 6000 \end{bmatrix} \text{ et } k_v = \begin{bmatrix} 369.70952 & 0 \\ 0 & 721.32335 \end{bmatrix}$$

Tableau 4.11 : Caractéristique de la réponse du robot à la commande PD plus gravité en présence du couple perturbateur.

ISTE1	0.226	Erreur statique 1	[-0.1, 0.1]
ISTE2	0.01282	Erreur statique 2	[-0.02, 0.0175]



(a)



(b)

Figure 4.24 : Positions des deux articulations commandées par PD plus gravité pour une entrée sinusoïdale avec perturbation.

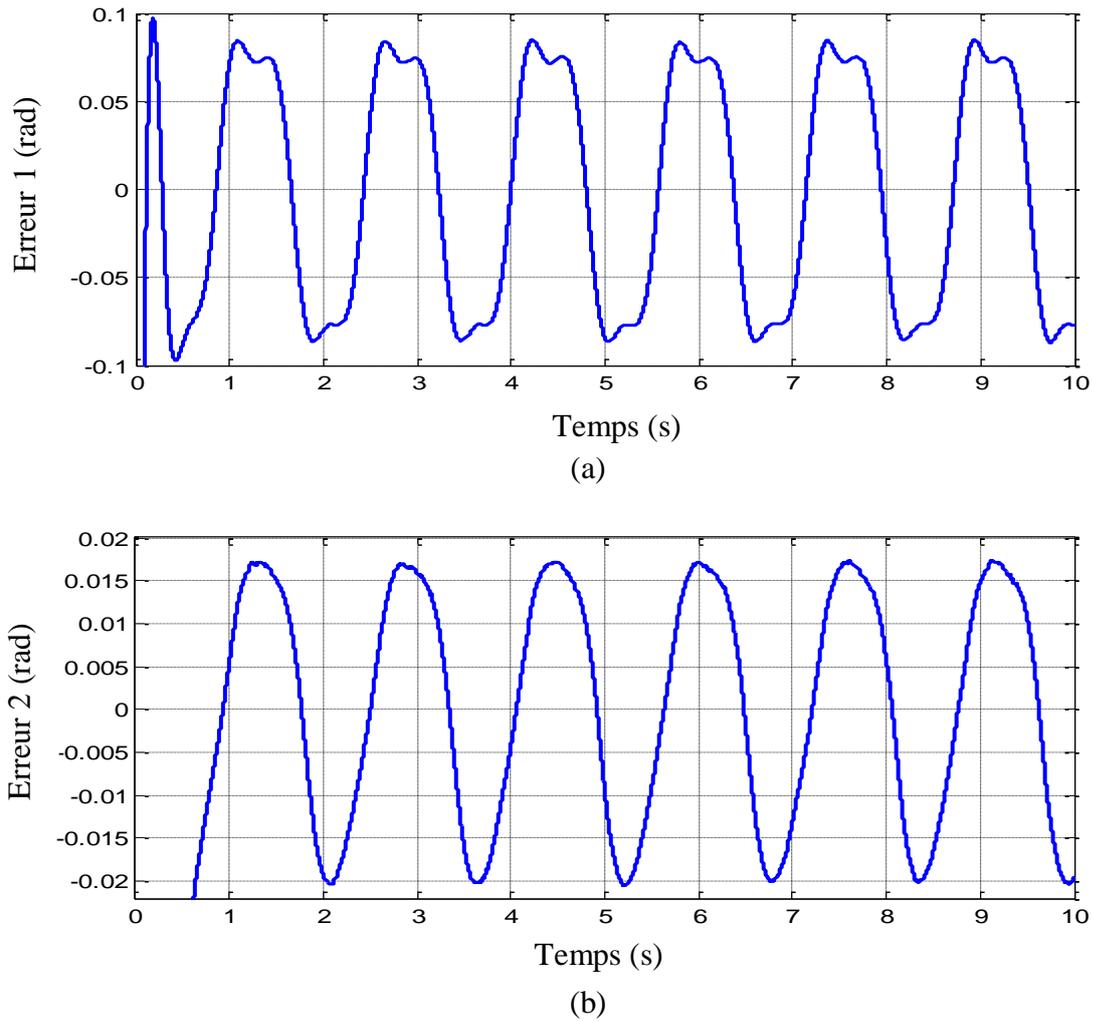


Figure 4.25 : Erreur de position des deux articulations avec perturbation.

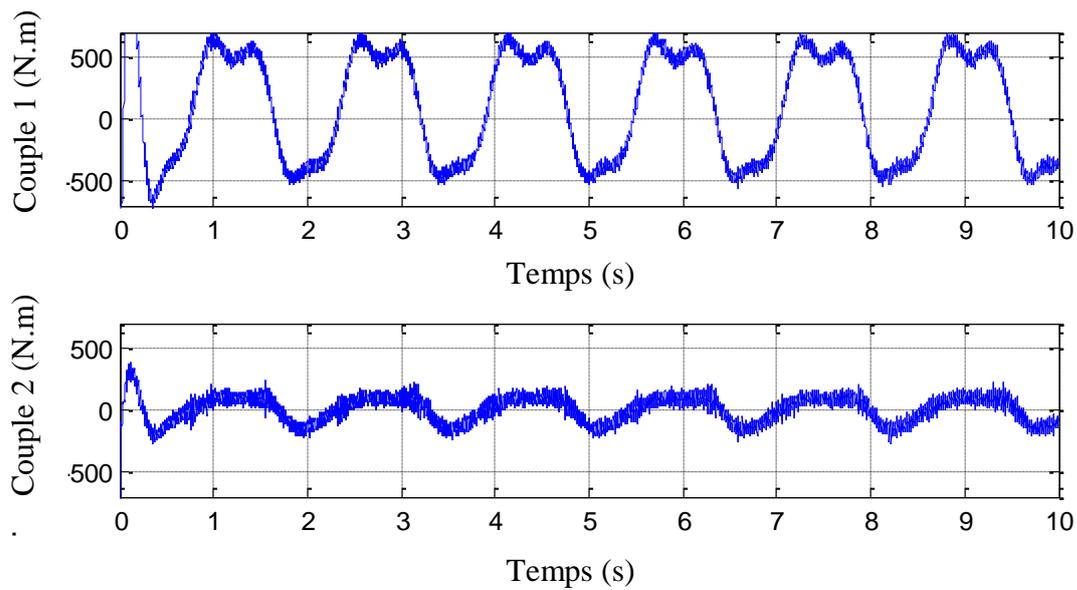


Figure 4.26 : Commande par PD plus gravité d'un Robot à 2 ddl avec perturbation.

ii. **Pour une commande constante**

Après l'étape d'optimisation les gains du contrôleur obtenus sont :

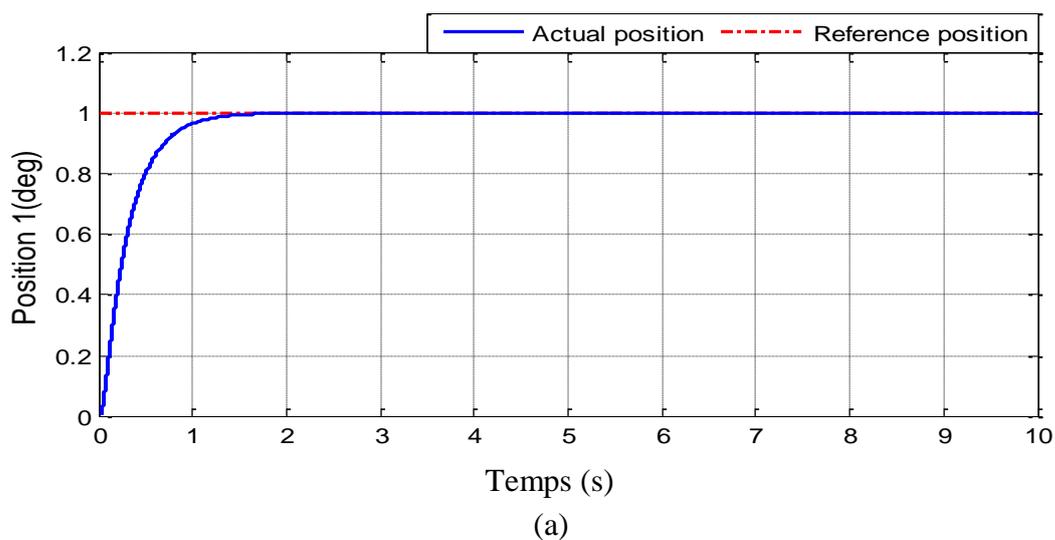
$$k_p = \begin{bmatrix} 2500 & 0 \\ 0 & 2500 \end{bmatrix} \text{ et } k_v = \begin{bmatrix} 813.8824 & 0 \\ 0 & 501.27642 \end{bmatrix}$$

Tableau 4.12 : Caractéristique de la réponse du robot à la commande PD plus gravité.

tr1	1.0039 s	Dépassement1	0%	ISTE1	0.028
tr2	1.0042 s	Dépassement2	8.801e-05%	ISTE2	0.01182

Les résultats obtenus de la commande du robot par le PD plus gravité contrôleur sont très acceptables en termes de l'erreur statique (ou le *ISTE*) et le temps de réponse, dans les trois cas de consigne : constante, sinus sans ou avec perturbation. Cependant, si on compare la commande PD plus gravité avec la commande couple calculer type PID appliquer au robot dans la section précédente on trouve que :

Premièrement, dans le cas d'une consigne sinusoïdale avec un couple perturbateur, à partir des tableaux 4.8 et 4.11, le PD plus gravité présente *0.226 de l'ISTE₁* de la première articulation par rapport à *4.677 e-05* de celui de la commande couple calculer type PID , et *0.01282 de l'ISTE₂* de la deuxième articulation par rapport à *0.001196* de celui de la commande couple calculer type PID.



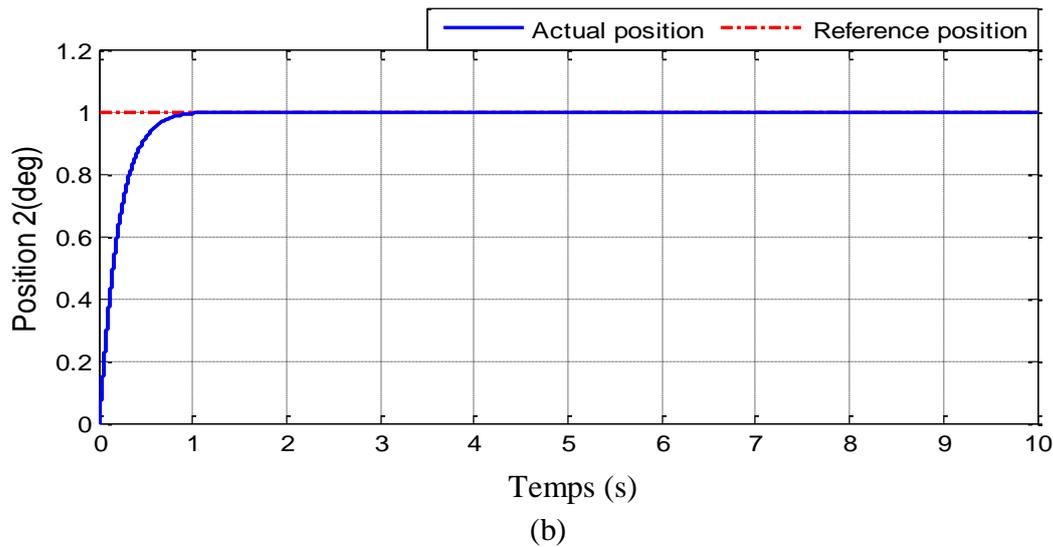


Figure 4.27 : Position des deux articulations commandé par PD plus gravité pour une entrée constante.

De plus, le PD plus gravité présente un intervalle de $[-0.1, 0.1]$ de l'erreur statique pour la première articulation par rapport à $[8.8 * 10^{-4}, 6.4 * 10^{-4}]$ de celui de la commande couple calculer type PID, et un intervalle de $[-0.02, 0.0175]$ de l'erreur statique pour la deuxième articulation par rapport à $[-6.3 * 10^{-3}, 6.3 * 10^{-3}]$ de celui de la commande couple calculer type PID.

Deuxièmement, et de même pour une excitation constante, à partir des tableaux 4.9 et 4.12 la commande couple calculer type PID présente une valeur de 0.0001334 de l' $ISTE_1$ mieux que celui de PD plus gravité par 0.028 , et de 0.000621 de l' $ISTE_2$ mieux que 0.01182 de celui de PD plus gravité.

Il est clair que la commande à couple calculé type PID est plus précis et peu sensible aux perturbations que la commande PD plus gravité.

Aussi, on peut voir à partir des résultats de simulation que les gains du PD augmentent, les performances de suivi s'améliorent. Cependant, quel que soit la valeur des gains du PD l'erreur de suivi ne tendra jamais vers zéro, mais est borné à zéro par une balle dont le rayon diminue à mesure que les gains augmentent. Il est alors intéressant de noter que les couples sont plus petits pour les gains élevés. C'est contraire à la croyance populaire, qui suppose que les couples de contrôle augmentent toujours avec l'augmentation des gains de PD. C'est dû au fait que des gains plus importants donnent des performance plus serré, et donc des erreurs de suivi plus petites.

4.5 Conclusion

Dans ce chapitre, nous avons présenté un tour d'horizon sur les techniques de commande classique des robots manipulateurs tels que la commande type PID. Le contrôle articulaire indépendant type PD et PID connu par leur popularité au début de la robotique, car il permet une analyse découplée du système en boucle fermée en utilisant des techniques classiques à entrée / sortie unique (SISO).

Ensuite nous couvrons une classe importante des contrôleurs de couple calculé, qui subsume de nombreux types d'algorithmes de contrôle de robot. Les algorithmes de contrôle à la fois classique et moderne sont décrits par cette classe, de sorte que le couple calculé fournit un pont entre des algorithmes plus anciens et plus modernes pour le contrôle de mouvement. En tant que types spéciaux d'algorithmes de couple calculé, nous avons mentionné le contrôle PD, Contrôle PID, PD-plus-gravité.

Références du chapitre 4 :

1. **Munro N, Lewis F:** *Robot manipulator control theory and practice*. 2^{ème} Edition, par Marcel Dekker, 2004.
2. **Kumar JS, Amuth EK:** Control and tracking of robot manipulator using PID controller and hardware in loop simulation. In: International conference on communication and network technologies (ICCNT), pp 1–3, 2014.
3. **Su Y, Muller PC, Zheng C:** Global asymptotic saturated PID control for robot manipulator. *IEEE Trans Control Syst Technol* 8(6):1280–1288, 2010.
4. **Abdelhedi F, Bouteraa Y, Chemori A, Derbel N:** Nonlinear PID and feedforward control of robotic manipulators. In: 15th International conference on sciences and techniques of automatic control and computer engineering (STA), pp 349–354, 2014.
5. **Bascetta L, Rocco P:** Revising the robust control design for rigid robot manipulator. In: International conference on robotics and automation. IEEE, pp 4478–4483, 2007.
6. **Kasac J, Novakovic B, Majetic D, Brezak D:** Global positioning of robot manipulator with mixed revolute and prismatic joints. *IEEE Trans Autom Control* 51:1035–1040, 2006.
7. **Su YX, Zheng CH:** PID control for global finite-time regulation of robotic manipulators. *Int J Syst Sci* 48:547–558, 2017.

8. **Mendoza M, Zavala-rio A, Santibanez V, Reyes F:** A generalized PID-type control scheme with simple tuning for the global regulation of robot manipulators with constrained inputs. *Int J Constr* 88:1995–2012, 2015.
9. **Akyuz IH, Yolacan E, Ertunc HM, Bingul Z:** PID and state feedback control of single-link flexible joint robot. In: *International conference on mechatronics*. IEEE, pp 409–414, 2011.
10. **Zavala-Rio A, Santibanez V:** Simple extensions of the PD with desired gravity compensation control law for robot manipulators with bounded inputs. *IEEE Trans Control Syst Technol* 14:958–965, 2006.
11. **Bouhajar S, Maherzi E, Khraief N, Besbes M, Belghith S:** Trajectory generation using predictive PID control for stable walking humanoid robot. In: *International conference on advanced wireless information and communication technology (AWICT2015) procedia computer science*, vol 73, pp 86–93, 2015.
12. **Leen G, Ray G:** A set of decentralized PID controllers for an n-link robot manipulator. *Indian Acad Sci Sadhana* 37:405–423, 2012.
13. **Su Y, Muller PC, Zheng CH:** A global asymptotic stable output feedback PID regulator for robot manipulator. In: *International conference on robotics and automation*. IEEE, pp 44489–44844, 2007.
14. **Su YX, Sun D, Ren L, Mills JK:** Integration of saturated PI synchronous control and PD feedback for control of parallel manipulators. *IEEE Trans Robot* 22:202–207, 2006.
15. **Kasac J, Novakovic B, Majetic D, Brezak D D:** Global positioning of robot manipulator with mixed revolute and prismatic joints. *IEEE Trans Autom Control* 51:1035–1040, 2006.
16. **Parra-Vega V, Arimoto S, Liu YH, Hirzinger G, Akella P:** Dynamic sliding PID control for tracking of robot manipulators: theory and experiments. *IEEE Trans Robot Autom* 19(6):967–976, 2003.
17. **Gorez R:** Globally stable PID-like control of mechanical systems. *Syst Control* 38(1):61–72, 1999.
18. **Kelly R:** Global positioning of robot manipulators via PD control plus a class of nonlinear integral actions. *IEEE Trans Autom Control* 43(7):934–938, 1998.
19. **Xu J, Qiao L:** Robust Adaptive PID control of robot manipulator with bounded disturbances. *Math Probl Eng* 2013:1–13, 2013.

20. **Wang H, Zhu S, Liu S:** Adaptive PID control of robot manipulators with Hinf tracking performance. In: International conference on advanced intelligence and mechatronics. IEEE/ASME, pp 1515–1520, 2009.
21. **Perez J, Perez JP, Soto R, Flores A, Rodriguez F, Meza JL:** Trajectory tracking error using PID control law for two link robot manipulator via adaptive neural networks. *Procedia Technol* 3:139–146, 2012.
22. **Cvejn J, Tvrđik J:** Learning control of robot manipulator based on a decentralized position dependent PID controller. In: 1st International conference on process control (PC), pp 167–172, 2017.
23. **Dehghani A, Khodadadi H:** Fuzzy logic self-tuning PID control for a single-link flexible joint robot manipulator in the presence of uncertainty. In: 15th international conference on control, automation and systems (ICCAS), pp 186–191, 2015.
24. **Du M, Fang J, Wang L:** A parameter self-tuning fuzzyPID control system for pneumatic manipulator of library robot. In: International conference on electronics, communications and control (ICECC), pp 4111–4115, 2011.
25. **Meza JL, Santibanez V, Soto R, Llama MA:** Stable fuzzy self-tuning PID control of robot manipulators. In: International Conference on Systems, Man, and Cybernetics. IEEE, pp 2624–2629, 2009.
26. **Norouzzadeh Ravari AR, Taghirad HD:** A novel hybrid fuzzy-PID controller for tracking control of robot manipulators. In: International conference in robotics and Biomimetic (ROBIO). IEEE, pp 16251–630, 2009.
27. **Yu W, Rosen J:** Neural PID control of robot manipulator with application to an upper limb exoskeleton. *IEEE Trans Cybern* 43(2):673–684, 2013.
28. **Nahapetian N, Jahed Motlagh MR, Analoui M:** PID gain tuning using Genetic algorithms and fuzzy logic for robot manipulator control. In: International conference on advanced computer control, pp 346–350, 2009.
29. **Nahapetian N, Jahed Motlagh MR, Analoui M:** Adaptive PID gain tuning using fuzzy logic and additional external performance index reference for controlling robot manipulator. In: International conference on advanced computer control, pp 448–452, 2009.
30. **Meza JL, Soto R, Arriaga J:** An optimal fuzzy self-tuning PID controller for robot manipulators via genetic algorithm. In: 8th Mescican international conference on artificial intelligence (MICAI), pp 21–26, 2009.

31. **Elkhateeb NA, Badr RI:** Novel PID tracking controller for 2DOF robotic manipulator system based on artificial bee colony algorithm. In: Electrical control and communication engineering, vol 13. Rig Technical University, pp 55–62, 2017.
32. **Mehdifar F, Gholami HS, Kharrati H, Menhaj MB:** A modified fruit fly optimization algorithm and its application to control of robot manipulator. In: 5th International conference on control, instrumentation and automation (ICCIA), pp 120–125, 2017.
33. **Kankashvar MR, Kharrati H, Asl RM, Sadeghiani AB:** Designing PID controllers for a five-bar linkage robot manipulator using BBO algorithm. In: 6th International conference on modeling, simulation and applied optimization (ICMSAO), pp1–6. 2015.
34. **Das MT, Dulger LC:** Control of SCARA robot: PSO–PID approach. Control Intell Syst 38(1) :24–31, 2010.
35. **Al-Saed F, Mohammed AH:** Design and implementation of PSO–PID controller for MA2000 robotic manipulator. Int J Comp Sci Eng Technol 2(9):1427–1433, 2012.
36. **Djaneye-Boundjou O, Xu X, Ordonez R:** Automated particle swarm optimization based PID tuning for control of robotic arm. In: IEEE national Aerospace and electronics conference (NECON) and Ohio innovation summit (OIS), pp 164–169, 2016.
37. **Thunyajarern S, Seeboonruang U, Kaitwanidvilai S:** PSO based adaptive force controller for 6DOF robot manipulators. In: Proceeding of the world congress on engineering and computer sciences (WCECS), San Francisco, USA, pp 691–695, 2017.
38. **Pano V, Ouyang PR:** Gain tuning of position domain PID control using particle swarm optimization. Robotica 34(6):1351–1366, 2014.
39. **Kapoor N, Ohr J:** Improved PSO tuned classical controllers (PID and SMC) for robotic manipulator. Int J Mod Educ Comp Sci 1:47–54, 2015.
40. **Elkaranshway HA, Bayoumi EHE, Soliman HM:** PSO based robust PID control for flexible manipulator systems. Int J Model Ident Constr 14(1–2):1–12, 2011.
41. **Tsai PW, Nguyen TT, Dao TK:** Robot path planning optimization based on multi-objective grey wolf optimizer. In: International conference on genetic and evolutionary computation. Springer, Berlin, pp 166–73, 2016.
42. **Spong MW, Hutchinson S, Vidyasagar M:** Robot modeling and control. John Wiley and Sons Inc., New York, 2006.
43. **Piltan F, Keshavarz M, Badri A, Zargari A:** Design Novel Nonlinear Controller Applied to Robot Manipulator: Design New Feedback Linearization Fuzzy Controller

with Minimum Rule Base Tuning Method. *International Journal of Robotics and Automation*,3 (1):1-12, 2012.

44. **Arimoto S, Miyazaki F:** Stability and robustness of PID feedback control for robot manipulators of sensory capability *Proc. First Int. Symp.*, pp 783–799, MIT, 1984.

Chapitre 5

Commande par mode
glissant des robots
manipulateurs

Commande par mode glissant d'un robot manipulateur

5.1 Introduction

Le contrôle des systèmes dynamiques en présence d'incertitudes du modèle et de perturbations, tels que les paramètres dynamiques (par exemple, l'inertie et conditions de charge utile), effets dynamiques (par exemple, frictions non linéaires complexes) et dynamique non modélisée, est devenu un problème courant à traiter lors de l'étude des installations industrielles. L'effet de ces incertitudes sur la dynamique du système doit être soigneusement pris en compte dans la phase de conception du contrôleur car il peut dégrader les performances ou même provoquer une instabilité du système, dans lequel les contrôleurs conventionnels ont de nombreuses difficultés à traiter ces incertitudes.

Pour cette raison, au cours des dernières années, le problème du contrôle des systèmes dynamiques en présence de fortes conditions d'incertitude est devenu un sujet de recherche important. En conséquence, des progrès considérables ont été atteints par l'application des techniques de contrôle robustes, tels que le contrôle adaptatif non linéaire, contrôle prédictif du modèle, contrôle par backstepping, contrôle par mode glissant et autres. Ces techniques sont capables de garantir les objectifs de contrôle malgré des erreurs de modélisation et / ou les incertitudes de paramètres affectant le système à contrôler. Parmi les méthodologies existantes, la commande par mode glissant (SMC), qui se caractérise par une grande simplicité et robustesse. Sa propriété la plus distinguée réside dans son insensibilité aux incertitudes dynamiques et perturbations externes, elle utilise essentiellement des lois de contrôle discontinues pour piloter la trajectoire d'état du système sur une surface spécifiée dans l'espace d'état, c-à-d la surface glissante ou de commutation, et pour garder l'état du système sur ce collecteur pour horizon de temps [1]. Cependant, cette approche présente des oscillations à haute fréquence appelé broutement lorsque l'état du système atteint la surface de glissement, ce

qui présente des effets négatifs sur les actionneurs contrôlés et excite la dynamique non modélisée indésirable.

Le contrôle à structure variable (VSC) avec le contrôle par mode glissant a d'abord été proposé et élaboré par plusieurs chercheurs de l'ancienne Russie, au cours des années soixante. (Emel'yanov and Taran, 1962 ; Emel'yanov, 1970 ; Utkin 1974). Les idées n'apparaissent pas en dehors de la Russie avant cette période quand un livre d'Itkis en 1976 [2] et un document d'enquête d'Utkin en 1977 [4] ont été publiés en anglais. En 1980, la partie principale de la théorie SMC classique était terminée et rapportée plus tard dans le Prof. Monographie d'Utkin en russe en 1981.

Depuis, le contrôle par mode glissant s'est développé en une méthode générale de conception de contrôle applicable à une large gamme de systèmes, y compris les systèmes non linéaires, les systèmes MIMO, les modèles à temps discret, à grande échelle et les systèmes à dimensions infinies. Les avantages d'obtenir un tel mouvement sont doubles : premièrement, le système se comporte comme un système d'ordre réduit par rapport au système d'origine ; et d'autre part le mouvement sur la surface de glissement du système est insensible à une classe de perturbation et aux incertitudes du modèle.

5.2 Commande par mode glissant

Le principe de ce type de contrôle consiste à apporter, quelles que soient les conditions initiales, le point représentatif de l'évolution du système sur une hyper surface dans le plan de phase représentant un ensemble de relations statiques entre les variables d'état. Le contrôle par mode glissant généralement comprend deux termes :

$$u(t) = u_{eq}(t) + u_n(t) \quad (5.1)$$

où $u_{eq}(t)$ un terme continu appelé contrôle équivalent et $u_n(t)$ un terme discontinu appelé commande de commutation.

- Contrôle équivalent : La méthode proposée par Utkin [4], consiste à admettre qu'en mode glissant tout se passe comme si le système était entraîné par une commande équivalente. Ce dernier correspond au régime de glissement idéal, pour lequel non seulement le point de fonctionnement reste en surface mais aussi la dérivée de la fonction de surface reste nulle $\dot{S} = 0$ (cela signifie, surface invariante dans le temps).

- Commande de commutation : La commande de commutation nécessite que le point de fonctionnement pour rester au voisinage de la surface. Le but principal de cette commande est de vérifier les conditions d'attractivité.

La conception du contrôleur par mode glissant offre une approche systématique du problème de maintenir la stabilité face aux imprécisions de modélisation.

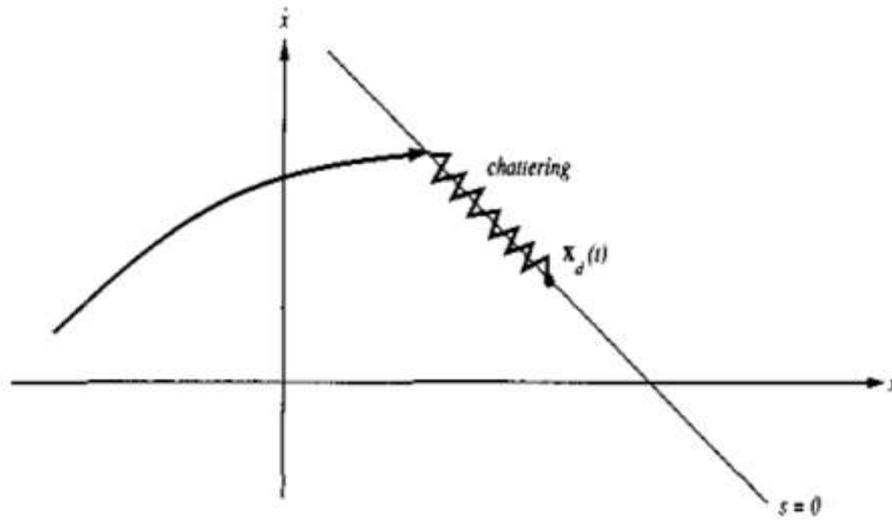


Figure 5.1: Broutement en raison d'une commutation de commande imparfaite.

5.3 Formulation du problème

Considérons le système non linéaire suivant affine dans le contrôle.

$$\dot{x}(t) = f(t, x) + g(t, x)u(t) \quad (5.2)$$

Avec $x(t) \in R^n$, $u(t) \in R^m$, $f(t, x) \in R^{n \times n}$ et $g(t, x) \in R^{n \times m}$.

Le composant de la rétroaction discontinu est donné par :

$$u_i = \begin{cases} u_i^+(t, x), & \text{si } \sigma_i(x) > 0 \\ u_i^-(t, x), & \text{si } \sigma_i(x) < 0 \end{cases} \quad i = 1, 2, \dots, m \quad (5.3)$$

Où $\sigma_i(x) = 0$ est la $i^{\text{ème}}$ surface de glissement, avec

$$\sigma(x) = [\sigma_1(x), \sigma_2(x), \dots, \sigma_m(x)]^T = 0 \quad (5.4)$$

est le $(n-m)$ collecteur glissant dimensionnel.

Le problème de contrôle consiste à développer la fonction continue u_i^+ , u_i^- , et la surface de glissement $\sigma(x) = 0$ de sorte que le système en boucle fermée (5.2) présente un mode de glissement sur le $(n - m)$ collecteur de glissement dimensionnel $\sigma(x) = 0$.

La conception de la loi de commande du mode glissant peut-être divisée en deux phases :

- **Phase 1** : consiste à construire une surface de glissement adaptée pour que la dynamique du système lié à la rampe de glissement produise un comportement souhaité.
- **Phase 2** : implique la conception d'une loi de commande discontinue qui force la trajectoire du système jusqu'à la surface de glissement et la maintienne à cet endroit.

Un mode de glissement idéal n'existe que lorsque la trajectoire d'état $x(t)$ du système à contrôler satisfait $\sigma(x(t)) = 0$ à chaque $t \geq t_0 \forall t_0$. À partir de l'instant t_0 , l'état du système est contraint à la discontinuité de la surface, qui est un ensemble invariant, après que le mode de glissement a été établi. Cela nécessite une commutation infiniment rapide. Dans les systèmes réels, il existe des imperfections tels que le retard, l'hystérésis, etc., qui forcent la commutation à fréquence finie. L'état du système oscille alors dans un voisinage de la surface de commutation ; cette oscillation est appelée broutement. Si la fréquence de la commutation est très élevée par rapport à la réponse dynamique du système, les imperfections et les fréquences de commutation finies sont souvent (mais pas toujours) négligeable.

5.4 Existence d'un mode glissant

Selon Itkis [2], Utkin [4], Edwards et Spurgeon [3], l'existence d'un mode glissant requiert une stabilité de la trajectoire de l'état vers la surface de glissement $\sigma(x) = 0$ au moins dans un voisinage de $\{x/\sigma(x)\} = 0$, c'est-à-dire l'état du système doit s'approcher à la surface au moins asymptotiquement. La plus grande zone est appelée la région d'attraction. Le problème d'existence peut être vu comme un problème de stabilité généralisé, par conséquent la deuxième méthode de Lyapunov fournit un cadre réel pour l'analyse. Plus précisément, la stabilité de la surface de commutation nécessite un choix d'une fonction de Lyapunov généralisée $V(t, x)$ définie positive ayant une dérivée temporelle négative dans la région d'attraction [1]. Formellement déclaré :

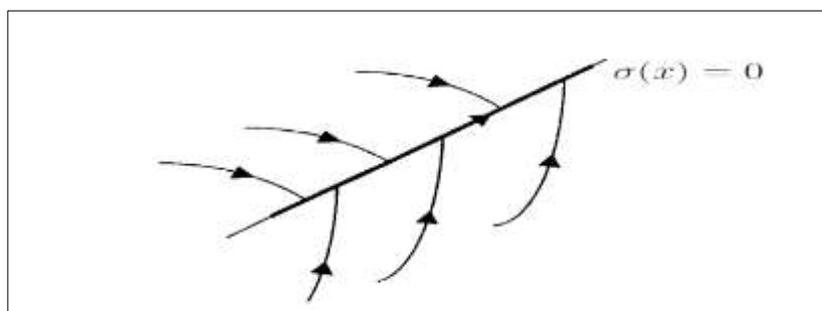


Figure 5.2: Attractivité du collecteur glissant

Preuve : d'après Utkin

Le domaine D est l'ensemble de x pour lequel l'origine du sous-espace ($\sigma_1 = 0, \sigma_2 = 0, \dots, \sigma_m = 0$) est un point d'équilibre asymptotiquement stable pour le système dynamique. Un mode glissant est globalement accessible si le domaine d'attraction est l'ensemble de l'espace statique. Sinon, le domaine d'attraction est un sous-ensemble de l'espace d'état. La structure de la fonction $V(t, x, \sigma)$ détermine la facilité avec laquelle on calcule les gains de rétroaction réels en mettant en œuvre la conception d'une commande par mode glissant. Malheureusement, il n'y a pas de méthode standard pour trouver une fonction Lyapunov pour les systèmes non linéaires arbitraires [4].

Notez que, pour tous systèmes à entrée unique, une fonction Lyapunov appropriée est :

$$V(t, x) = \frac{1}{2} \sigma^2(x) \quad (5.5)$$

est clairement (globalement) définie positive. Dans le contrôle du mode glissant, $\dot{\sigma} = 0$ dépend de la commande si des gains de rétroaction commutés peuvent être choisis pour que :

$$\dot{V}(t, x, \sigma) = \sigma \frac{\partial \sigma}{\partial t} < 0 \quad (5.6)$$

Alors, dans le domaine d'attraction, la trajectoire d'état converge vers la surface et limitée à la surface pendant tout le temps t . Cette dernière condition est appelée la condition d'atteinte ou d'accessibilité [3, 4] qui garantit que le collecteur glissant est atteint asymptotiquement.

5.5 Conception de la surface de glissement

La méthode de commande équivalente peut être utilisée pour déterminer le mouvement du système limité à la surface de commutation $\sigma(x) = 0$. La nature analytique de cette méthode est en fait un outil puissant à la fois pour l'analyse et la conception.

Considérons le système non linéaire (5.2), la première étape de l'approche de contrôle équivalente consiste à trouver l'entrée u_{eq} de telle sorte que la trajectoire d'état reste sur la surface de commutation $\sigma(x) = 0$. L'existence du mode glissant implique que $\sigma(x) = 0$, pour tout $t \geq t_0$, et, $\dot{\sigma} = 0$.

En différenciant $\sigma(x)$ par rapport au temps le long de la trajectoire de (5.2) donne :

$$\left[\frac{\partial \sigma}{\partial x} \right] \dot{x}(t) = \left[\frac{\partial \sigma}{\partial x} \right] [f(t, x) + g(t, x)u_{eq}] = 0 \quad (5.7)$$

A noter que sous l'action du contrôle équivalent u_{eq} toute trajectoire à partir de la variété $\sigma(x) = 0$ reste dessus, puisque $\dot{\sigma} = 0$. Par conséquent, la variété glissante $\sigma(x) = 0$ est un ensemble invariant.

Pour calculer u_{eq} , supposons que le produit matriciel $\left[\frac{\partial \sigma}{\partial x}\right] g(t, x)$ est non singulier pour tout t et x . Alors

$$u_{eq} = - \left[\frac{\partial \sigma}{\partial x} g(t, x) \right]^{-1} \frac{\partial \sigma}{\partial x} f(t, x) \quad (5.8)$$

Par conséquent, étant donné $\sigma [x(t_0)] = 0$, la dynamique du système sur la surface de commutation pour $t \geq t_0$, est obtenue en substituant (5.8) à (5.2), c'est-à-dire :

$$\dot{x}(t) = \left[I - g(t, x) \left[\frac{\partial \sigma}{\partial x} g(t, x) \right]^{-1} \frac{\partial \sigma}{\partial x} \right] f(t, x) \quad (5.9)$$

Dans le cas particulier d'une surface de commutation linéaire, $\sigma(x) = S(x(t))$, (5.9) devient :

$$\dot{x}(t) = [I - g(t, x)[S g(t, x)]^{-1} S] f(t, x) \quad (5.10)$$

Cette structure peut être avantageusement exploitée dans la conception de surfaces de commutation.

5.6 Modes glissants du second ordre : première génération

Au début des années 80, la communauté de contrôle avait compris que le principal inconvénient de la commande par modes glissants (SMC) est l'effet de « broutement ». Il a été démontré que cet effet est principalement causé par une dynamique de cascade non modélisée qui augmente le degré relatif du système et perturbe le mode de glissement idéal existant dans le système [5]. Afin de surmonter le problème du broutement en mode glissant, le concept du mode glissant du deuxième ordre (SOSM) a été introduit dans la thèse de doctorat de [6].

Définition : Le point $(x, \dot{x}) = (0, 0)$ est appelé point du mode glissant du second ordre s'il s'agit d'une solution du système.

$$\ddot{x}(t) = f(t, x, \dot{x}) + g(t, x, \dot{x})u \quad x, t, f(t), g(t) \in R \quad (5.11)$$

au sens de Filippov [7].

5.6.1 Algorithme de torsion (Twisting Algorithm)

Le premier et le plus simple algorithme (SOSM) est celui connu sous « l'algorithme de torsion » (TA). Pour un système de degré relatif égal à deux, le TA prend la forme :

$$u = -a \text{sign}(\dot{x}) - b \text{sign}(x), b > a \quad (5.12)$$

Sous l'hypothèse de bornes connues pour f et g et avec les paramètres a et b du contrôleur choisis de manière appropriée, l'algorithme de torsion assure la convergence exacte en temps fini de x et \dot{x} , c'est-à-dire qu'il existe $T > 0$ de telle sorte que pour tout $t > T$,

$x(t) = \dot{x}(t) = 0$. Ainsi, le TA est censé être un contrôle SOSM algorithme car il fournit l'existence d'un "mode glissant du second ordre" stable à l'origine $(x, \dot{x}) = (0,0)$. Selon des hypothèses supplémentaires concernant la fluidité du système, l'algorithme a été utilisé pour atténuer les broutements dans les systèmes de degré relatif 1 y compris un intégrateur dans l'entrée de commande.

$$\begin{cases} \dot{x}(t) = f(t, x) + g(t, x)u(t) \\ \dot{u} = v \\ v = -a \operatorname{sign}(\dot{x}) - b \operatorname{sign}(x) \end{cases} \quad (5.13)$$

et de cette façon, la commande réelle u est absolument continue.

5.6.2 L'algorithme de super-torsion (The Super-Twisting Algorithm)

Comme nous l'avons vu, l'utilisation de l'algorithme de torsion dans un système avec degré relatif 1 permet de compenser exactement une perturbation (absolument continue) au moyen d'un contrôle absolument continu. Cependant, cette approche requiert la dérivée \dot{x} .

L'algorithme de super torsion (STA) :

$$\begin{cases} \dot{x}(t) = f(t, x) + g(t, x)u \\ u = -k_1 |x|^{\frac{1}{2}} \operatorname{sign}(x) + v \\ \dot{v} = -k_2 \operatorname{sign}(x) \end{cases} \quad (5.14)$$

Pour tout f est Lipshitz d'incertitude / perturbation bornée pour certaines constantes k_1 et k_2 assure une convergence en temps fini exacte au deuxième ensemble de modes glissants $x(t) = \dot{x}(t) = 0, \forall t \geq T$ sans utilisation de \dot{x} .

5.6.3 L'algorithme sous-optimal (The Sub-Optimal Algorithm)

Depuis 1996, le nombre de publications sur la théorie et les applications des modes glissants du second ordre a augmenté de façon exponentielle, généralement à travers les efforts du professeur Bartolini et de son groupe. Les résultats du groupe du professeur Bartolini sont basés sur un autre contrôleur SOSM qui n'a pas (explicitement) besoin de dérivés, l'algorithme sous-optimal (SOA) :

$$\begin{cases} \ddot{x}(t) = f(t, x, \dot{x}) + g(t, x, \dot{x})u \\ u = -U_m \operatorname{sign}\left(x - \frac{1}{2} x^*\right) \end{cases} \quad (5.15)$$

Où x^* est le dernier extrema de la courbe $x(t)$. L'algorithme SOA est également capable de garantir la convergence exacte en temps fini $x(t) = \dot{x}(t) = 0, \forall t \geq T$ une fois le paramètre U_m a été réglé assez grand.

5.7 Contrôleurs mode glissant d'ordre arbitraire : Deuxième génération

En 2001, le premier contrôleur SM d'ordre arbitraire a été introduit en [8]. De tels contrôleurs ont permis de résoudre l'application en temps fini d'un mode glissant d'ordre $r > 2$ et une compensation exacte des perturbations / incertitudes. Étant donné le degré relatif r de la sortie, les contrôleurs HOSM sont construits en utilisant une forme de récursivité. Ce qui suit est la récursivité pour le premier type des contrôleurs HOSM signalé : dite « imbriqués » [9].

Prenons le système :

$$\begin{cases} \dot{x}(t) = f(t, x) + g(t, x)u \\ \sigma = \sigma(x, t) \end{cases} \quad x \in R^n \quad (5.16)$$

Où $\sigma \in R$ est une sortie à stabiliser exactement en temps fini à $\sigma = 0, u \in R$ est l'entrée de commande et $x \in R^n$ est l'état. Dans un tel cas, le problème de contrôle est traduit par la stabilisation en temps fini d'une équation différentielle incertaine ou, de manière équivalente, de l'inclusion différentielle suivante :

$$\sigma^{(r)} \in [-C, C] + [K_m, K_M]u \quad (5.17)$$

Où C, K_m et K_M sont des constantes connues paramétrant l'incertitude du système d'origine.

Si $r \leq 2$, les contrôleurs SM de premier ordre ou SOSM sont capables de résoudre le problème. Cependant, le cas où $r > 2$ est resté jusqu'à la fin des années 90 un problème ouvert.

Soit p le plus petit multiple commun de $1, 2, \dots, r$ et soit :

$$\begin{cases} \varphi_{0,r} = \sigma & , N_{1,r} = |\sigma|^{\frac{r-1}{r}} \\ \varphi_{i,r} = \sigma^{(i)} + \beta_i N_{i,r} \operatorname{sign}(\varphi_{i-1,r}), & N_{i,r} = (|\sigma|^{\frac{p}{r}} + \dots + |\sigma^{(i-1)}|^{\frac{p}{r-i+1}})^{\frac{r-i}{p}} \end{cases} \quad (5.18)$$

et le contrôleur par mode glissant d'ordre r :

$$u = -\alpha \operatorname{sign}(\varphi_{r-1,r}(\sigma, \dot{\sigma}, \dots, \sigma^{(r-1)})) \quad (5.19)$$

appliqué au système (5.14). Cet algorithme prévoit alors la stabilisation en temps fini de σ , $\dot{\sigma} = 0$, et donc de ses dérivées successives jusqu'à $\sigma^{(r-1)}$. Les paramètres β_i peuvent être sélectionnés à l'avance de telle sorte que seul le gain du contrôleur α doit être sélectionné suffisamment grand [5].

5.8 Contrôle par mode de glissement terminal (Terminal Sliding Mode Control)

L'idée du contrôle par mode glissant terminal a été rapportée pour la première fois dans [10-12] où une variable glissante terminale (vecteur) avec un terme non linéaire est proposée pour un système non linéaire du second ordre ainsi qu'un système du robot manipulateur rigide à n liens avec une dynamique incertaine. Le principe de base de la commande par mode glissant terminal est décrit comme suit.

Considérons le système linéaire ou non linéaire du deuxième ordre :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(x_1, x_2) + b(x_1, x_2)u(t) \end{cases} \quad (5.20)$$

Où x_1 et x_2 sont les états du système, $f()$ et $b() = 0$ sont des fonctions linéaires ou non linéaires de x_1 et x_2 , et u est l'entrée de commande. Afin d'atteindre le temps fini de convergence des variables d'états, la variable glissante terminale du premier ordre est défini par :

$$s = x_2 + \beta x_1^{\frac{q}{p}} \quad (5.21)$$

Où $\beta > 0$, p et q sont des entiers impairs positifs avec $p > q$.

Avec un contrôleur de mode glissant correctement conçu de la forme :

$$u(x) = \begin{cases} u^+(x) & \text{si } s > 0 \\ u^-(x) & \text{si } s < 0 \end{cases} \quad (5.22)$$

La variable de glissement terminale s peut être conduite vers la surface du mode de glissement terminal $s = 0$ dans un temps fini. Sur la surface du mode glissant terminal, la dynamique du système est ensuite déterminé par l'équation différentielle non linéaire suivante :

$$\dot{x}_1 = -\beta x_1^{\frac{q}{p}} \quad (5.23)$$

Bien que la technique du mode glissant terminal a été largement appliquée au contrôle des systèmes mécanique, systèmes électriques, systèmes aéronautiques et autres systèmes complexes, le développement de cette technique est encore à son stade initial, et de nombreuses recherches théoriques doivent être effectuées dans le futur proche.

5.9 HOSMs non homogènes (Non-Homogeneous HOSMs) : Troisième génération

Dans [13], Les contrôleurs HOSM ont été étendus pour inclure un gain variable et sont ainsi devenus non homogène. Dans ce cas, le contrôleur prend la forme :

$$u = -\alpha\Phi(t, x) \Psi_{r-1,r}(\sigma, \dot{\sigma}, \dots, \sigma^{(r-1)}) \quad (5.24)$$

Où la « fonction de gain » $\Phi(t, x) > 0$ peut être sélectionnée pour résoudre le problème où les limites de l'inclusion différentielle (5.17) ne sont pas constantes.

5.10 Contrôle par intégral du mode glissant (Integral Sliding Mode Control)

Les modes de glissement intégral ont été suggérés comme un outil pour atteindre les objectifs suivants :

- Compensation des perturbations appariées à partir du moment initial, c'est-à-dire assurer le démarrage en mode glissant à partir du moment initial. Les perturbations appariées correspondent au cas où les perturbations affectent indirectement les états du système robotique via l'entrée de commande. C'est-à-dire que les perturbations et l'entrée de commande se trouvent dans le même canal. En revanche, les perturbations dépareillées représentent le cas où les perturbations affectent directement les états du système robotique sans entrée de commande, c'est-à-dire que les perturbations et l'entrée de commande ne sont pas dans le même canal.
- Préservation de la dimension du système initial, c'est-à-dire sauvegarde du système dynamique précédemment conçue pour le cas idéal (sans perturbation).

Ces deux propriétés attrayantes de l'ISM leur permettent d'être utilisées avec succès dans différents types d'applications, par exemple la robotique et les problèmes de conception.

Supposons qu'une loi de contrôle $u = u_0(x, t)$ garantissant l'objectif de contrôle (par exemple pilotage, stabilisation ou suivi) est déjà disponible pour un système nominal idéal.

$$\dot{x} = f(x, t) + B(x)u \quad , x \in R^n, u \in R^m \quad (5.25)$$

De plus, supposons qu'au lieu du système idéal (5.25), on ait un système perturbé :

$$\dot{x} = f(x, t) + B(x)(u + \delta) + \Phi_{um} \quad (5.26)$$

Où δ est une perturbation appariée et ϕ_{um} est une perturbation dépareillées. Puis une loi de commande du mode glissant $u_1(x, t)$ peut être facilement introduite de telle sorte que la boucle fermée :

$$\dot{x} = f(x, t) + B(x)(u_0 + u_1 + \delta) + \phi_{um} \quad (5.27)$$

est insensible à δ . Commençons par construire la variable glissante :

$$s(x, t) = g(x) - z(t), \quad s(x, t) \in R^m \quad (5.28)$$

Où

$$z(t) = g(x_0) + \int_{t_0}^t G(x)[f(x, \tau) + B(x)u_0(x, \tau)] d\tau \quad G(x) = \frac{\partial g}{\partial x}(x) \quad (5.29)$$

$g(x)$ est n'importe quelle fonction telle que $G(x)B(x)$ est inversible. Notez d'abord que, à $t = t_0$, nous avons $s = 0$. Ainsi le système commence au surface de glissement (il n'y a pas de temps pour atteindre). Calculons maintenant la dérivée temporelle de s :

$$\begin{aligned} \dot{s} &= G(x)[f(x, t) + B(x)(u_0 + u_1 + \delta) + \phi_{um} - f(x, t) - B(x)u_0] \\ &= G(x)B(x)(u_1 + \delta) + G(x)\phi_{um} \end{aligned} \quad (5.30)$$

On peut facilement voir que si δ et ϕ_{um} sont bornés par des fonctions connues, alors il est possible de construire un contrôle unitaire u_1 assurant $\dot{s}=0$. Le contrôle équivalent est :

$$u_{eq} = -\delta - (G(x)B(x))^{-1}G(x)\phi_{um} \quad (5.31)$$

De sorte que les trajectoires du système à la surface de glissement sont données par :

$$\dot{x} = f(x, t) + B(x)u_0 + [I - B(x)(G(x)B(x))^{-1}G(x)]\phi_{um} \quad (5.32)$$

Ce qui montre l'insensibilité par rapport à δ . Il est clair que la matrice de projection $G(x)$ doit

- Ne pas amplifier la perturbation résiduelle :

$$\phi_{sm} = \left[\underbrace{I - B(x)(G(x)B(x))^{-1}G(x)}_{\Gamma(x)} \right] \phi_{um} \quad (5.33)$$

- Minimiser (5.33).

Si $B(x)$ est constant ($B(x) = B$), alors G peut être choisi de telle sorte qu'il minimise la norme de la matrice de projection :

$$\Gamma(x) = [I - B(GB)^{-1}G] \quad (5.34)$$

Dans le cas optimal, $G = B^+$. (B^+ est le pseudo-inverse de Moore-Penrose de B) [5].

5.11 Suppression et atténuation des broutements dans les systèmes mode glissant

Les trois principales approches d'atténuation et de suppression des broutements dans les systèmes mode glissant ont été proposées au milieu des années 80 :

- L'utilisation de la saturation de contrôle au lieu du contrôle discontinu, Cette approche permet la continuité du contrôle mais ne peut restreindre la dynamique du système sur la surface de commutation. Elle assure la convergence vers une couche limite du collecteur de glissement, dont la taille est définie par les caractéristiques de la pente de la saturation.
- L'approche basée sur l'observateur. Cette méthode permet de contourner la dynamique des systèmes par la boucle de broutement. Cette approche réduit le problème de la commande robuste à celui d'une estimation exacte et robuste.
- L'approche en mode glissant du second ordre (SOSM) permet une convergence en temps fini vers zéro non seulement de la variable glissante mais aussi de sa dérivée.

Tous les contrôleurs SOSM, à l'exception de l'algorithme de super torsion (STA), nécessitent la connaissance des valeurs des dérivés (ou le dispositif spécial identifiant la valeur maximale de la sortie en cas de l'algorithme sous-optimal), c'est-à-dire la connaissance de la perturbation si la surface de glissement à un degré relatif 1. L'algorithme de super-torsion (STA) est un algorithme en mode glissant absolument continu assurant toutes les propriétés principales du contrôle du mode glissant de premier ordre pour des systèmes avec des incertitudes / perturbations appariées continues de Lipschitz avec des gradients bornés [14].

5.12 Commande par mode glissant d'un robot manipulateur

Au cours des dernières décennies, la technique de contrôle du mode glissant (SMC) pour les robots manipulateurs a été étudié de manière approfondie par de nombreux chercheurs. différents types des contrôleurs mode glissant, présentés en section précédente, ont été utilisé pour la commande du robot manipulateurs. Le contrôleur mode glissant super-Tortion est

conçus et présenté de façon non exhaustive dans [15-19], le contrôleur mode glissant deuxième ordre [20-22] et le contrôleur mode glissant Terminal [23, 24]. Récemment, les contrôleurs mode glissant proportionnel intégral dérivé (SMC_PID) examinés comme un contrôleur puissant non linéaire [25–31]. Un contrôle adaptatif du mode glissant a été traité dans plusieurs articles [32-35]. L'intégration de la logique floue avec SMC a été réalisée dans de nombreux travaux [36–42]. Afin d'améliorer les performances du contrôle SMC, GA a été appliqué pour déterminer une fois les paramètres optimaux du contrôleur SMC conventionnel [43- 45] et encore une fois pour le contrôleur par mode glissant flou [45- 47]. Vijay et Jena ont conçu le contrôleur par backstepping mode glissant et un observateur basé sur le PSO pour les robots manipulateurs [48]. Les auteurs dans [49, 50] ont utilisé le PSO pour régler les paramètres de la surface de glissement de SMC couplé avec un système d'inférence neuro floue artificielle (ANFIS), et d'ajuster les fonctions d'appartenance du contrôleur SMC flou type Takagi Sugeno respectivement. Une nouvelle combinaison d'algorithme PSO avec un SMC super torsion pour éliminer les broutements est proposé dans [51]. Medjghou et al. ont présenté l'optimisation d'un filtre de Kalman étendu pour un contrôleur intégral SMC flou type 2 optimisé par l'algorithme (BBO) [52]. SMC conventionnel a été optimisé par l'algorithme colonie concurrentiel dans [53]. Oliveira et al. ont proposé l'utilisation de GWO avec une base chaotique pour régler une commande SMC plus robuste pour le contrôle de position du robot manipulateur [54]. Plus récemment, Mokeddem et Draïdi présentent le SMC_PID pour contrôler un système linéaire du second ordre [55], ils ont prouvé que l'optimisation des paramètres du contrôleur en utilisant ALO surpasse significativement les algorithmes GA et PSO. Visant le problème du contrôle des systèmes non linéaires, un nouveau contrôleur SMC_PID optimal pour le contrôle du suivi de trajectoire du robot manipulateur à 2ddl est proposé dans cette partie. Une nouvelle application de l'algorithme du lion de fourmis (ALO) pour le paramétrage du contrôleur SMC avec une surface PID, pour le contrôle de la position des articulations du robot. Afin de montrer l'efficacité de cet algorithme, une étude comparative basée sur l'application d'autres algorithmes d'optimisation a été réalisée.

5.13 Commande par mode glissant avec une surface proportionnelle intégrale dérivée (SMC_PID) d'un robot manipulateur

Le modèle dynamique d'un robot manipulateur à deux degrés de liberté (2ddl) présenté dans le chapitre 2 est donné par :

$$\tau = M(q)\ddot{q} + H(q, \dot{q}) + G(q) + F(\dot{q}) \quad (5.35)$$

En général, la surface de glissement du contrôle SMC peut être formé selon l'équation :

$$s(t) = \left(\lambda + \frac{d}{dt} \right)^{n-1} e(t) \quad (5.36)$$

Où n désigne l'ordre du système à contrôler.

La surface PID du contrôle par mode glissant peut-être indiquée en utilisant l'équation suivante:

$$s(t) = k_d \dot{e}(t) + k_p e(t) + k_i \int_0^{t_f} e(t) dt \quad (5.37)$$

Avec k_p , k_i et k_d les paramètres du contrôleur PID.

$$e(t) = q_d(t) - q(t) \rightarrow \dot{e}(t) = \dot{q}_d(t) - \dot{q}(t) \quad (5.38)$$

où q_d , q , \dot{q}_d et \dot{q} sont la position, la vitesse désirée et réelle des articulations du robot respectivement.

Pour calculer u_{eq} , il faut que $\dot{s}(t) = 0$. Ce qui implique que :

$$k_p \dot{e}(t) + k_i e(t) + k_d \ddot{e}(t) = 0.$$

$$\text{et} \quad \ddot{e}(t) = -k_d^{-1} (k_p \dot{e}(t) + k_i e(t)) = \ddot{q}_d(t) - \ddot{q}(t). \quad (5.39)$$

Alors

$$\ddot{q} = M(q)^{-1} (u_{eq}(t) - H(q, \dot{q}) - G(q) - F(\dot{q})) = \ddot{q}_d + k_d^{-1} (k_p \dot{e}(t) + k_i e(t)) \quad (5.40)$$

$$u_{eq} = M(q) (\ddot{q}_d + k_d^{-1} (k_p \dot{e}(t) + k_i e(t))) + H(q, \dot{q}) + G(q) + F(\dot{q}) \quad (5.41)$$

Pour obtenir le signal de commande de commutation u_n pour le contrôleur par mode glissant avec une surface PID (SMC_PID), la fonction de Lyapunov peut être choisi comme :

$$V = \frac{1}{2} s^T(t) s(t) \quad (5.42)$$

$$\dot{V} = s^T(t) \dot{s}(t) = s^T(t) [k_p \dot{e}(t) + k_i e(t) + k_d \ddot{e}(t)] \quad (5.43)$$

$$\dot{V} = s^T(t) \left[k_p \dot{e}(t) + k_i e(t) + k_d (\ddot{q}_d(t) - (M(q)^{-1} (u_{eq}(t) + u_n(t) - H(q, \dot{q}) - G(q) - F(\dot{q})))) \right] \quad (5.44)$$

le critère ITAE, l'ALO dépasse les autres algorithmes en terme de meilleure fonction de coût, ce qui signifie que l'algorithme ALO donne le meilleur optimum, par rapport au critère ISTE, les optimiseurs GWO et ALO fournissent des valeurs de coût proches considérées comme les meilleurs coûts obtenus. Les paramètres optimaux correspondants du SMC_PID sont récapitulés dans le tableau 5.2.

i. Optimisation SMC_PID sans perturbations

Tableau 5.1: Fonction de coût pour la première et la deuxième articulation sans perturbations

Fonction de coût	ITAE		ISTE	
	1 ^{ère}	2 ^{ème}	1 ^{ère}	2 ^{ème}
SMC_PID_ALO	9.669e-05	3.896e-05	1.149e-07	6.858e-08
SMC_PID_SCA	123.6e-05	17170e-05	1.302e-07	1.602e-07
SMC_PID_WOA	634.2e-05	61.82e-05	2.234e-07	6.042e-08
SMC_PID_GWO	82.2e-05	300.9e-05	2.095e-08	8.118e-08

Tableau 5.2: Paramètres du SMC_PID pour la première et la deuxième articulation sans perturbations

	k_{p1}	k_{i1}	k_{d1}	λ_1	k_{p2}	k_{i2}	k_{d2}	λ_2
SMC_PID_ALO	500	52.46	43.16	35.25	500	68.13	30.43	41.49
SMC_PID_SCA	500	1	58.32	62.34	500	31.28	53.55	14.02
SMC_PID_WOA	500	55.31	60.01	43.8	500	65.3	1	45.65
SMC_PID_GWO	500	49.89	15.75	43.01	603	37	46	48

La figure 5.3 montre le couple appliqué à la première et la deuxième articulation obtenue jusqu'à présent par les différents algorithmes d'optimisation. Il est clair que le signal de contrôle obtenu par l'optimisation ALO et SCA est très souple par rapport à ceux obtenus par les algorithmes GWO et WOA. Ceci est un autre critère sélectif pour confirmer la supériorité de l'algorithme ALO sur les autres algorithmes. La figure 5.4 illustre le suivi de trajectoire d'un robot manipulateur à 2 ddl contrôlé par un contrôleur SMC_PID optimisé par les algorithmes GWO, WOA, SCA et ALO respectivement minimisant le critère ISTE. On peut voir les bonnes performances du robot avec différentes méthodes d'optimisation. Cela prouve la robustesse des algorithmes évolutionnaires.

La figure 5.5 montre clairement la meilleure convergence d'ISTE obtenue par ALO par rapport aux autres algorithmes. La fonction "*sign*", qui est utilisée dans l'équation (5.48) génère plus d'effet de broutement sur le couple de commande. En pratique, ce broutement peut endommager les composants du système. En effet, afin d'éviter ces effets indésirables, la fonction *sign* est remplacée par la "*tanh*" tangente hyperbolique. La position finale de

l'effecteur du robot contrôlé par le contrôleur ALO_SMC_PID dans le plan XY est présenté dans les figures 5.6, 5.7 et 5.8 respectivement. L'entrée de commande des contrôleurs optimisés par différents algorithmes est décrite dans la figure 5.9.

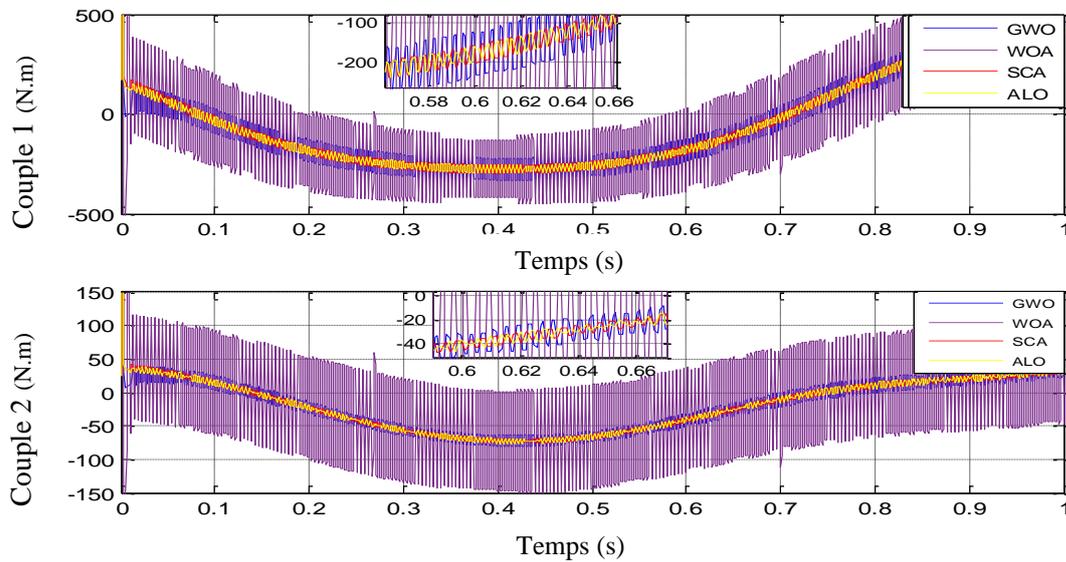


Figure 5.3: Le couple de contrôle pour la première et la deuxième articulation du robot manipulateur.

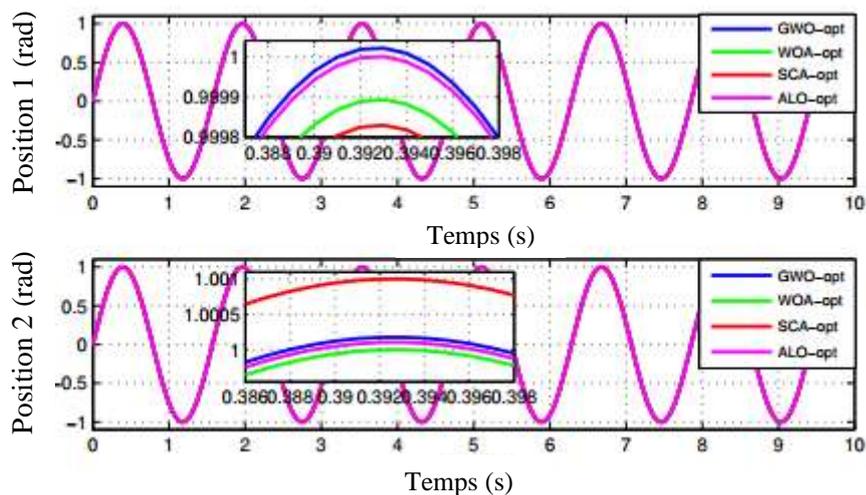


Figure 5.4 : Suivi de trajectoire des liaisons 1 et 2 avec contrôleur SMC_PID optimisé sans perturbations.

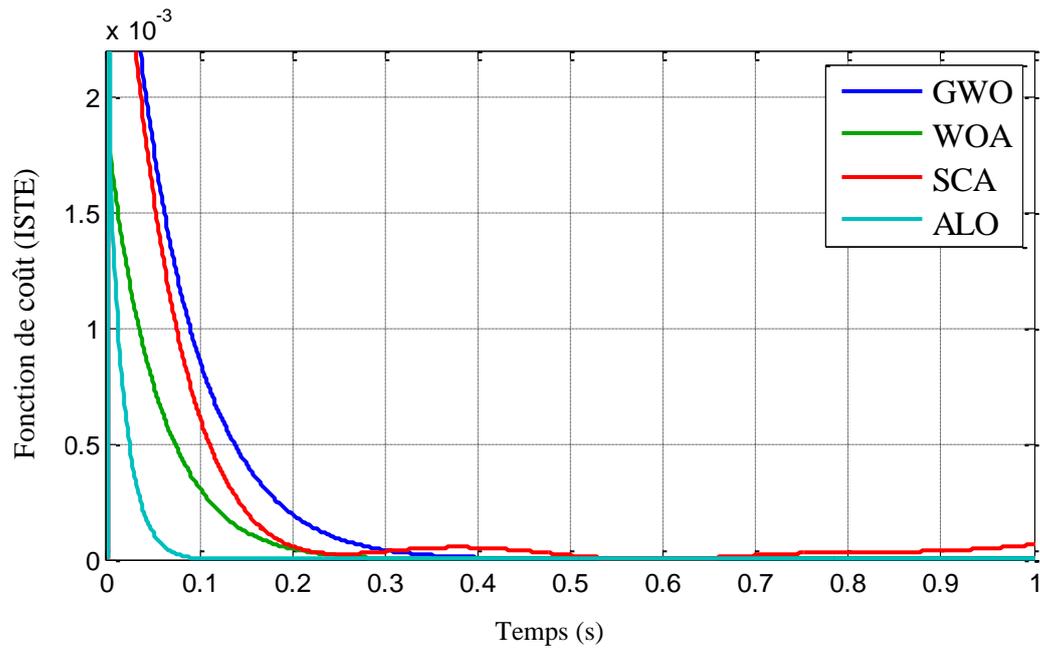


Figure 5.5: Courbes de convergence des critères ISTE.

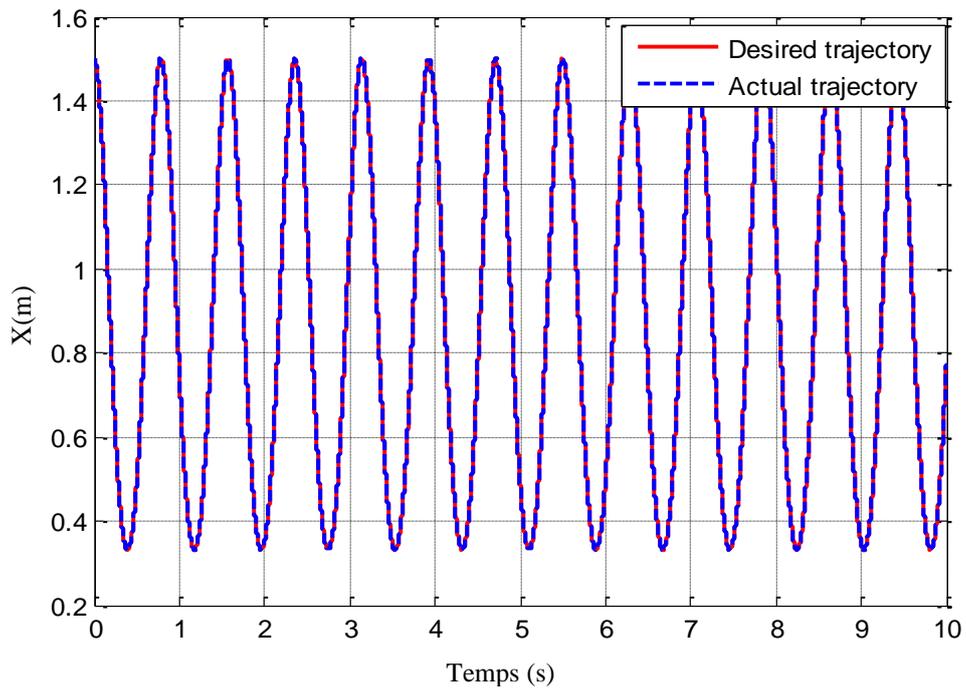


Figure 5.6 : Position X de l'organe terminal du robot à 2ddl contrôlé par le contrôleur ALO_SMC_PID.

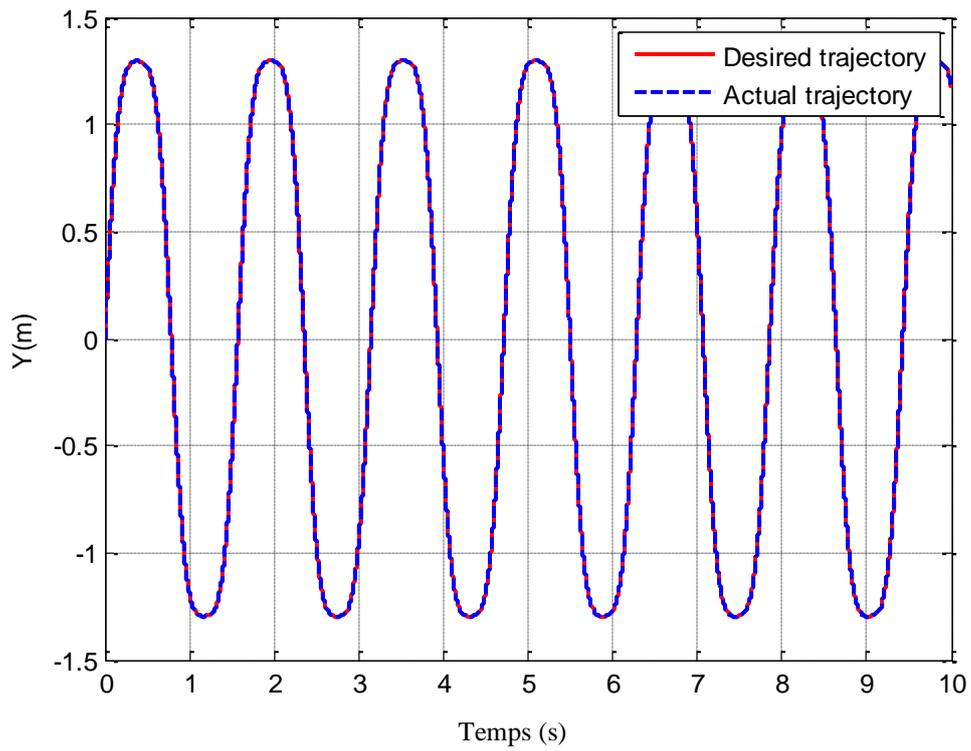


Figure 5.7 : Position Y de l'organe terminal du robot à 2ddl contrôlé par le contrôleur ALO_SMC_PID.

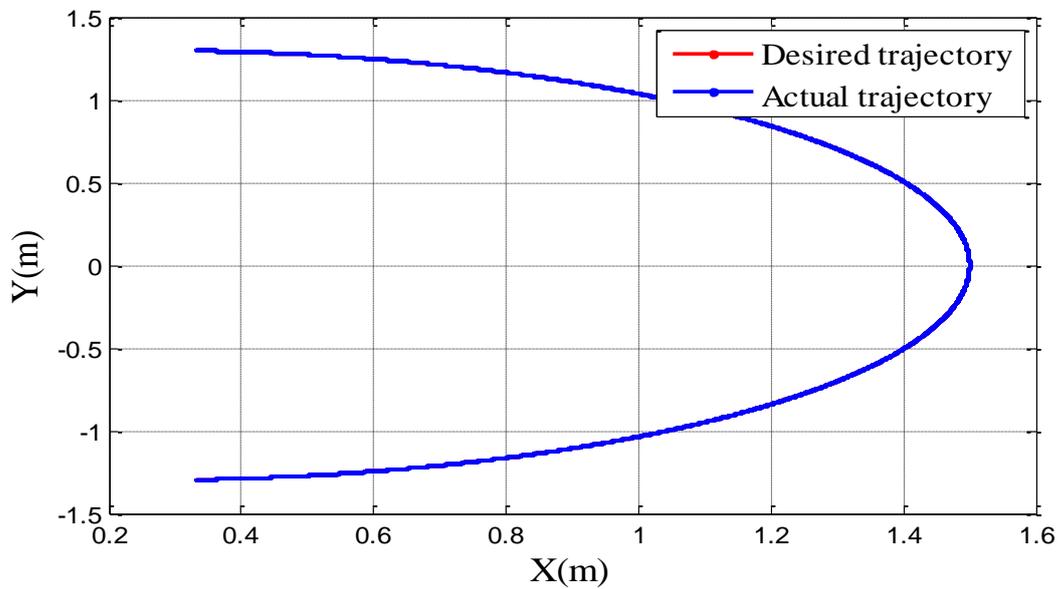


Figure 5.8 : Position de l'organe terminal dans le plan XY du robot à 2ddl contrôlé par le contrôleur ALO_SMC_PID.

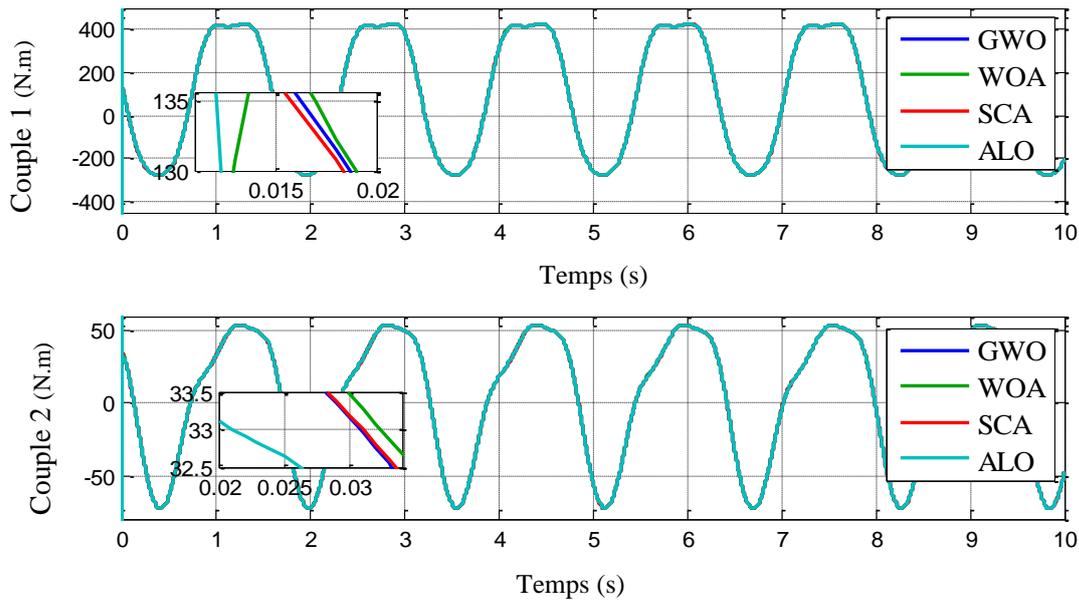


Figure 5.9: Commande SMC_PID pour la première et la deuxième articulation du manipulateur de robot.

ii. Optimisation SMC_PID avec perturbations de couple

Dans la suite, nous procédons à l'analyse de robustesse lorsque le bras du robot est soumis à une perturbation externe sous la forme d'un signal additif à la commande d'entrée. Cette perturbation est mise en œuvre avec un bruit blanc avec différentes variances. Les tableaux 5.3 à 5.6 répertorient les paramètres SMC_PID et les valeurs de la fonction de coût avec différentes puissances du bruit externe.

Tableau 5.3: Fonction de coût ISTE en présence de perturbations de couple pour l'optimisation GWO

δ	0.01	0.1	0.5	0.8
K_{p1}	500	500	500	500
K_{i1}	0	0	0	0
K_{d1}	52.39	92.83	17.33	37.34
λ_1	30.19	92.62	70.73	100
K_{p2}	511	680	530	522
K_{i2}	4	5	34	28
K_{d2}	14	64	9	16
λ_2	20	99	91	79
ISTE ₁	2.489e-06	2.752e-06	1.552e-06	4.393e-06
ISTE ₂	1.143e-05	0.001413	4.034e-05	0.001353
ISTE ₁₊	1.3919e-05	0.0014	4.1892e-05	0.0014
ISTE ₂				

Tableau 5.4: Fonction de coût ISTE en présence de perturbations de couple pour l'optimisation WOA.

δ	0.01	0.1	0.5	0.8
K_{p1}	500	500	500	500
K_{i1}	0	0	0	0
K_{d1}	64.24	75.13	78.74	67.9
λ_1	78.71	31.34	37.55	17.14
K_{p2}	500	500	500	500
K_{i2}	0	0	0	0
K_{d2}	11.16	97.71	30.44	36.93
λ_2	70.53	27.45	85.42	67.23
ISTE ₁	6.341e-07	2.119e-05	28.22e-05	45.36e-04
ISTE ₂	1.074e-06	0.3255	0.005707	0.0868
ISTE ₁₊	1.7081e-06	0.3255	0.0060	0.0913
ISTE ₂				

Tableau 5.5: Fonction de coût ISTE en présence de perturbations de couple pour l'optimisation SCA

δ	0.01	0.1	0.5	0.8
K_{p1}	500	500	500	500
K_{i1}	100	63.56	1	94.46
K_{d1}	89.48	100	11.61	55.81
λ_1	72.73	38.02	19.49	45.07
K_{p2}	500	500	500	500
K_{i2}	20.69	100	50.8	89.79
K_{d2}	29.24	1	1	80.07
λ_2	100	24.49	1.28	62.999
ISTE ₁	1.568e-06	3.1e-05	6.61e-06	0.0001073
ISTE ₂	2.258e-06	5.834e-07	0.0001134	1.875
ISTE ₁₊	3.826e-6	3.158e-5	1.2e-4	1.8751
ISTE ₂				

Tableau 5.6: Fonction de coût ISTE en présence de perturbations de couple pour l'optimisation ALO

δ	0.01	0.1	0.5	0.8
K_{p1}	500	500	500	500
K_{i1}	98.61	39.45	76.83	16.38
K_{d1}	72.98	25.57	16.19	36.23
λ_1	41.75	35.38	62.87	1
K_{p2}	500	500	500	500
K_{i2}	23.64	1	39.96	1
K_{d2}	1	1	1	1
λ_2	7.88	1	12.279	27.76
ISTE ₁	2.633e-06	2.191e-06	1.76e-06	0.0123
ISTE ₂	2.093e-07	2.164e-05	6.484e-06	4.224e-06
ISTE ₁₊	2.8423e-6	4.355e-6	8.244e-6	1.2304e-2
ISTE ₂				

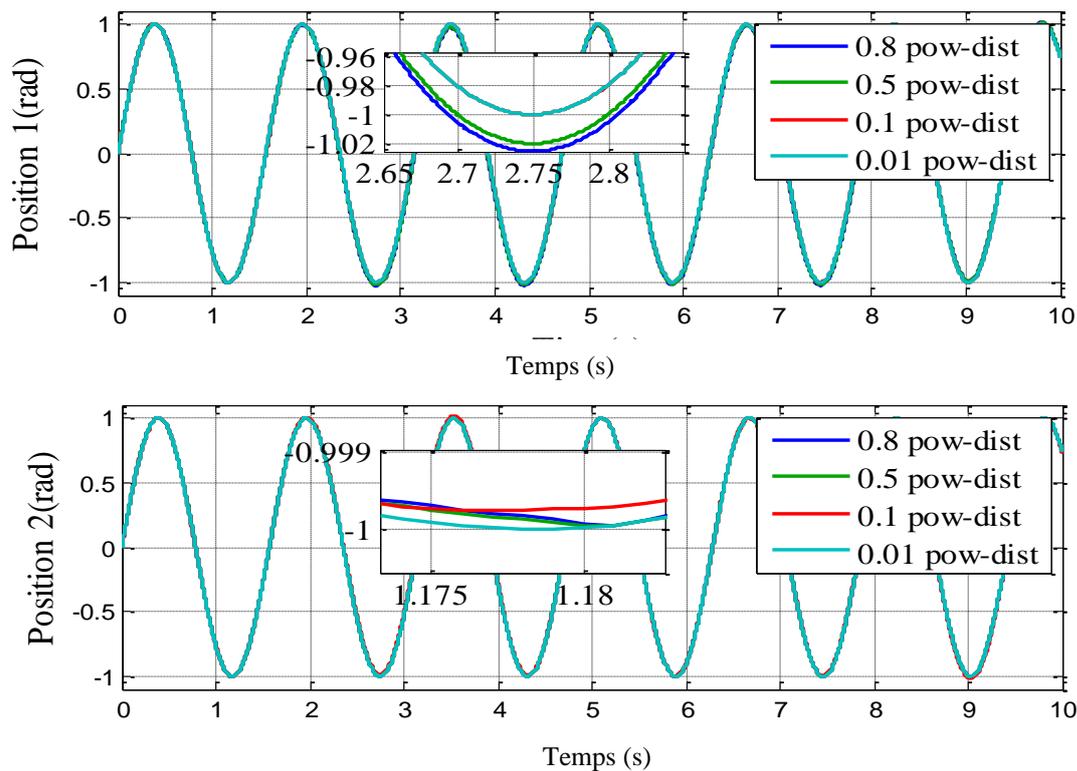


Figure 5.10: Suivi de position avec perturbations des deux articulations avec SMC_PID optimisé par ALO.

On peut voir d'après le test de robustesse du SMC_PID que la fonction de coût et les paramètres du contrôleur obtenus avec les algorithmes GWO, WOA, SCA et ALO changent relativement avec la puissance du bruit. Comme on peut le voir dans la figure 5.10, l'algorithme ALO est le plus robuste pour, éliminer l'effet des perturbations de couple. Par conséquent, le robot manipulateur contrôlé par SMC_PID a une très bonne performance et devient moins sensible au bruit avec un petit écart par rapport à la trajectoire de référence, se produit lorsque la puissance de bruit augmente.

5.15 Conclusion

Dans ce chapitre, La méthodologie de la commande par mode glissant est décrite. Cette technique est considérée comme l'une des techniques de contrôle les plus efficaces sous conditions d'incertitude élevées. Les objectifs de contrôle sont atteints en contraignant la dynamique du système sur une surface correctement choisie au moyen des lois de contrôle discontinues. Cette méthodologie offre une grande précision et robustesse par rapport à une

large gamme de perturbations et d'incertitudes. Cependant, en raison de la présence d'imperfections dans les actionneurs et les capteurs, telles que l'hystérésis, les retards, etc., et à la présence des bruits et / ou des perturbations exogènes, cette approche de contrôle peut produire des broutements, pour les éviter, différentes approches ont été proposées dans la littérature. L'idée principale de telles approches est de changer la dynamique dans un petit voisinage de la surface de discontinuité afin d'éviter une réelle discontinuité et, en même temps, de préserver les propriétés principales de l'ensemble du système. Cependant, la précision ultime et la robustesse du mode glissant est partiellement perdue. Au contraire, les modes de glissement d'ordre supérieur généralisent l'idée de base de mode glissant agissant directement sur les dérivées temporelles d'ordre supérieur de la variable glissante au lieu d'influencer sa première dérivée temporelle comme cela se produit en modes glissants standard. Le problème principal dans la mise en œuvre des modes glissants d'ordre supérieur est la demande croissante d'informations. D'une manière générale, tout contrôleur de glissement de $(r)^{ème}$ ordre nécessite la connaissance des dérivées temporelles de la variable de glissement jusqu'au $(r - 1)^{ème}$ ordre. Les seules exceptions sont données par le contrôleur de torsion, le contrôleur super torsion et l'algorithme sous-optimal qui sont des algorithmes de commande de mode glissant du second ordre. Pour cette raison, ces contrôleurs sont les plus largement utilisés dans la pratique parmi les contrôleurs de mode glissant d'ordre supérieur à cause de leur simplicité et de leur faible demande d'information.

Une commande par mode glissant avec une surface PID a été réalisée pour la commande d'un robot à 2ddl. Le paramétrage du contrôleur été faite avec des nouvelles techniques d'optimisation tel que l'ALO, SCA, GWO et WOA pour tester l'efficacité du technique de commande. D'après les résultats de simulations, le SMC_PID_ALO présente plus de robustesse dans le contrôle de suivi de trajectoire du robot manipulateur en présence ou sans perturbations. Également, le signal d'entrée appliquée aux articulations était très souple en comparaison avec d'autres stratégies d'optimisation, qui est un facteur très important pour la sécurité des actionneurs. On a vu aussi d'après les simulations la rapidité dynamique du contrôleur SMC à atteindre la consigne de référence.

Références du chapitre 5 :

1. **Vecchio C:** *Sliding Mode Control: Theoretical developments and applications to uncertain mechanical systems*. L'université de Pavie science informatique et systèmes.
2. **Itkis U:** *Control systems of variable structure*.1976.
3. **Edwards C, Spurgeon S:** *Sliding Mode Control: Theory and Applications*. Series in Systems and Control, 1998.
4. **Utkin:** *Variable structure systems with sliding modes*. IEEE Transactions on Automatic Control, Issue 2. pp: 212 – 222. 1977.
5. **fridman L, Moreno J, Rafael I:** *Sliding mode after the first decade of the 21st century*. Lecture note in control and information sciences 412, Springer 2011.
6. **Emel'yanov SV, Korovin SK, Levantovskii LV:** *Computational methods and algorithms: A family of new regulators based on second order sliding mode*. Matematicheskoe modelirovanie. issue 3. pp 89-100, 1990.
7. **Filippov AF:** *Differential equations with discontinuous righthand sides*. Mathematics and its applications. kluwer academic publisher,1988.
8. **levant A:** *Arbitrary order sliding modes with finite time convergence*. Theory and Practice of Control and Systems. pp. 349-354, 1999.
9. **levant A:** *Universal SISO sliding-mode controllers with finite-time. convergence*. IEEE Transactions on Automatic Control 46(9), 1447–1451,2001.
10. **Man Z, Paplinski, AP, Wu HR:** *A robust MIMO terminal sliding mode control scheme for rigid robotic manipulators*. IEEE Trans. Automatic Control 39, 2464–2469, 1994.
11. **Man Z, Yu X:** *Adaptive terminal sliding mode tracking control for rigid robotic manipulators with uncertain dynamics*. JSME Int. J. of Mechanical Systems, Machine Elements and Manufacturing 40(3), 493–502, 1997.
12. **Venkataraman ST, Gulati S:** *Control of nonlinear systems using terminal sliding modes*. In: Proc. American Control Conference, pp. 891–893, 1992.
13. **Levant A, Michael A:** *Adjustment of high-order sliding-mode controllers*. International Journal of Robust and Nonlinear Control 19(15), 1657–1672, 2009.
14. **Gonzalez T, Moreno JA, Fridman L:** *Variable Gain Super-Twisting Sliding Mode Control*. IEEE transactions on automatic control. VOL. 57, NO. 8. pp,2100-2105, 2012.
15. **Tayebi-Haghighi S, Piltan F, Kim JM:** *Robust Composite High-Order Super-Twisting Sliding Mode Control of Robot Manipulators*. Robotics 2018.

16. **Jeong CS, Kim JS, Han SI:** *Tracking Error Constrained Super-twisting Sliding Mode Control for Robotic Systems*. International Journal of Control Automation and Systems 16(6), 2018.
17. **Javed MU, Liu H, Nie J, Sun J:** *Robust tracking control for Robotic Manipulators based on Super-Twisting Algorithm*. IOP Conference Series: Materials Science and Engineering, 2019.
18. **Wang Y, Chen J, Yan F, Zhu k, Chen B:** *Adaptive super-twisting fractional-order nonsingular terminal sliding mode control of cable-driven manipulators*. ISA Transactions Volume 86, Pages 163-180, 2019.
19. **Shupeng zheng, Xingjian Niu, chenghui peng:** *Adaptive Super-Twisting-like sliding mode control with prescribed performance for robot manipulators*. Journal of mechanics in Medicine and Biology, vol19, N08, 2019.
20. **Capisani LM, Ferrara A, Magnani L:** *Design and experimental validation of a second-order sliding-mode motion controller for robot manipulators*. International Journal of Control. Volume 82, Issue 2, 2009.
21. **Parra-Vega V :** *Second Order Sliding Mode Control for Robot Arms with Time Base Generators for Finite-Time Tracking*. Dynamics and control, 11,2001,175-186.
22. **Ferrara A, Incremona GP:** *Design of an Integral Suboptimal Second-Order Sliding Mode Controller for the Robust Motion Control of Robot Manipulators*. IEEE Transactions on Control Systems Technology.2015.
23. **Hendel R, khaber F, Essounbouli N:** *Adaptive high order sliding mode controller/observer based terminal sliding mode for MIMO uncertain nonlinear system*. International Journal of Control, 2019.
24. **Jin M, Lee J, Chang PH, Choi C:** *Practical Nonsingular Terminal Sliding-Mode Control of Robot Manipulators for High-Accuracy Tracking Control*. IEEE Transactions on Industrial Electronics. Volume: 56, Issue: 9, 2009.
25. **Adhikary N, Mahanta C:** *Sliding mode control of position commanded robot manipulators*. Control Engineering Practice, pp. 183-198. December 2018.
26. **Jung S:** *Improvement of Tracking Control of a Sliding Mode Controller for Robot Manipulators by a Neural Network*. International Journal of Control. Automation and Systems, pp. 937–943, April 2018.

27. **Yoo DS:** *A Comparison of Sliding Mode and Integral Sliding Mode Controls for Robot Manipulators*. Transactions of the Korean Institute of Electrical Engineers. January 2009.
28. **Liu R, Li S:** *Optimal integral sliding mode control scheme based on pseudo spectral method for robotic manipulators*. International Journal of Control, 2014.
29. **Zhao Y, Sheng Y, Liu X:** *A Novel Finite Time Sliding Mode Control for Robotic Manipulators*. IFAC Proceedings Volumes, pp. 7336-7341, 2014.
30. **Azlan NZ, Osman JHS:** *Proportional Integral Sliding Mode Control of Hydraulic Robot Manipulators with Chattering Elimination*. First International Conference on Industrial and Information Systems. Sri Lanka, Aug 2006.
31. **Das M, Mahanta C:** *Optimal second order sliding mode control for nonlinear uncertain systems*. ISA Transactions, pp. 1191-1198, July 2014.
32. **Adelhedi F, Jribi A, Bouteraa Y, Derbe N:** *Adaptive Sliding Mode Control Design of a SCARA Robot Manipulator System Under Parametric Variations*. Journal of Engineering Science and Technology Review, pp. 117-123, 2015.
33. **Jiang S, Zhao J, Xie F, Fu J, Wang X, Li Z:** *A Novel Adaptive Sliding Mode Control for Manipulator with External Disturbance*. 37th Chinese Control Conference. 2008.
34. **Yi S, Zhai J:** *Adaptive second-order fast nonsingular terminal sliding mode control for robotic manipulators*. ISA Transactions. January 2019.
35. **Jing C, Xu H, Niu X:** *Adaptive sliding mode disturbance rejection control with prescribed performance for robotic manipulators*. ISA Transactions. February 2019.
36. **Mirshekaran M, Piltan F, Esmaeili Z, Khajeaian T, Kazeminasab M:** *Design Sliding Mode Modified Fuzzy Linear Controller with Application to Flexible Robot Manipulator*. International Journal of Modern Education and Computer Science, pp. 53-63, 2013.
37. **Tran MD, Kang HJ:** *Adaptive Fuzzy PID Sliding Mode Controller of Uncertain Robotic Manipulator*. International Conference on Intelligent Computing. Intelligent Computing Theories and Methodologies, pp. 92-103. ICIC, 2015.
38. **Mohammad A, Ehsan SS:** *Sliding mode PID controller design for robot manipulators by using fuzzy tuning approach*. 27th Chinese Control Conference. 2008.

39. **Esfahani HN, Azimirad V:** *A New Fuzzy Sliding Mode Controller with PID Sliding Surface for Underwater Manipulators*. International Journal of Mechatronics, Electrical and Computer Technology, pp. 224-249, October 2013.
40. **Kharabian B, Bolandi H, Ehyaei A F, Mousavi Mashhadi S K, Smailzadeh SM:** *Adaptive Tuning of Sliding Mode-PID Control in Free Floating Space Manipulator by Sliding Cloud Theory*. American Journal of Mechanical and Industrial Engineering, pp. 64-71, 2017.
41. **Medhaffar H, Derbel N, Damak T:** *A decoupled fuzzy indirect adaptive sliding mode controller with application to robot manipulator*. International journal of modeling, identification and control, pp. 23-29, 2006.
42. **Qureshi MS, Swarnkar P, Gupta S:** *Fuzzy PID Sliding Mode Control for Robotics: An Application to Surgical Robot*. Recent Advances in Electrical and Electronic Engineering, pp. 118 – 129, 2019.
43. **Firdaus AR, Rahman AS:** *Genetic Algorithm of Sliding Mode Control Design for Manipulator Robot*. TELKOMNIKA, pp. 645-654, 2012.
44. **Long Y, Du, ZJ, Wang WD, Dong W:** *Robust Sliding Mode Control Based on GA Optimization and CMAC Compensation for Lower Limb Exoskeleton*. Applied bionics and biomechanics, 13 pages, 2016.
45. **Zhou T, Liang XF:** *Position Sliding Mode Control of Manipulator Joint Based on Genetic Algorithm*. Advanced Materials Research, pp. 727-731, 2014.
46. **Rezapour J, Sharif M, Nariman-Zadeh N:** *Application of fuzzy sliding mode control to robotic manipulator using multi-objective genetic algorithm*. International Symposium on Innovations in Intelligent Systems and Applications, pp.455-459. 2011.
47. **Jalili-Kharaajoo M, Rouhani H:** *Fuzzy Sliding Mode Control of Robotic Manipulators Based on Genetic Algorithms*. Advances in Artificial Intelligence, pp.892-900, 2004.
48. **Vijay M, Jena D:** *PSO based backstepping sliding mode controller and observer for robot manipulators*. International Conference on Power, Instrumentation, Control and Computing. PICC, 2018.
49. **Vijay M, Jena D:** *PSO based neuro fuzzy sliding mode control for a robot manipulator*. Journal of Electrical Systems and Information Technology, pp. 243-256. May 2017.

50. **Khooban MH, Soltanpour M:** *A particle swarm optimization approach for fuzzy sliding mode control for tracking the robot manipulator.* Nonlinear Dynamics vol.74, pp. 467–478, 2013.
51. **Ibrahim AE, Gharghory S:** *Optimal Design of Super Twisting Control with PSO Algorithm for Robotic Manipulator.* International Journal of Intelligent Systems and Applications in Robotics, 15 pages, 2018.
52. **Medjghou A, Ghanai M, Chafaa K:** *BBO optimization of an EKF for interval type-2 fuzzy sliding mode control.* International Journal of Computational Intelligence Systems, pp. 770 – 789, 2018.
53. **Jalali A, Piltan F, Keshtgar M, Jalali M:** *Colonial Competitive Optimization Sliding Mode Controller with Application to Robot Manipulator.* I.J. Intelligent Systems and Applications, pp. 50-56, 2013.
54. **Oliveira J, Oliveira PM, Boaventura-Cunha J, Pinho T:** *Chaos-based grey wolf optimizer for higher order sliding mode position control of a robotic manipulator.* Nonlinear dynamics, pp. 1353–1362, 2017.
55. **Mokeddem D, Draïdi H:** *Optimization of PID Sliding Surface Using Antlion Optimizer.* International Symposium on Modelling and Implementation of Complex Systems MISC 2018, pp. 133-145, 2018.

Chapitre 6

Commande par
backstepping des
robots manipulateur

Commande par backstepping d'un robot manipulateur

6.1 Introduction

La plupart des systèmes physiques (procédés) qui nous entourent sont non linéaires. bien souvent, ces non-linéarités sont faibles ou ne sont pas visibles sur la plage d'opérations de ces procédés. Le souci constant d'améliorer les performances des systèmes commandés conduit à des modélisations de plus en plus précises qui permettent de répondre sur une plus large plage d'opérations. C'est à ce moment que les non-linéarités se font sentir et rendent les outils d'analyse et/ou de synthèse des lois de commande utilisés dans le domaine linéaire caduques et absolument incapables de rendre compte de certains phénomènes. C'est pourquoi, depuis quelques années, beaucoup de recherche ont été effectuées dans le domaine de la commande des systèmes non linéaires. Le backstepping fait partie de ces nouvelles méthodes de contrôle[1, 2] a été développé par Kanellako poulos et al. et inspiré par les travaux de Feurer et Morse d'une part et Tsinias et Kokotovii et Sussmann d'autre part. Elle offre une méthode systématique pour effectuer le design d'un contrôleur pour les systèmes non linéaires [3].

Au cours de la dernière décennie, le système de commande de backstepping a été largement exploité dans une variété d'applications pour les systèmes de commande robotiques [6-10]. Pour le contrôle adaptatif, les techniques intelligentes telles que la logique floue et les réseaux de neurones sont avérées d'être un bon candidat pour améliorer la capacité de la méthodologie de conception réursive de backstpping [11-18]. Pour assurer la stabilité, le rejet des perturbations et un bon suivi pour le manipulateur robotique, une stratégie de contrôle basée sur le couplage du Backstepping et mode glissant a été présentée dans [19-21].

Pourtant, il n'est pas facile d'obtenir des performances satisfaisantes car les paramètres du contrôleur obtenus par la méthode de backstepping sont arbitrairement choisis. Il est nécessaire de sélectionner les paramètres appropriés pour obtenir une bonne réponse, c'est pourquoi l'incorporation des algorithmes d'optimisation est le bon choix. Pour obtenir les paramètres optimaux du contrôleur de backstepping, les auteurs de [22] ont utilisé l'algorithme des abeilles

pour déterminer le contrôleur de backstepping optimal pour le contrôle du chaos dans un système de torche à plasma de type tige. Un contrôleur de backstepping intelligent réglé à l'aide d'une optimisation d'essaim de particules chaotiques (CPSO) est proposé par les mêmes auteurs pour contrôler le chaos dans le système chaotique de Genesio-Tesi dans [23].

6.2 Commande par backstepping

Le contrôle par Backstepping est un nouveau type de méthodologie de conception réursive et systématique pour le contrôle de rétroaction d'un système non linéaire incertain, en particulier pour le système avec des incertitudes appareillées. Le point le plus intéressant est d'utiliser la variable de contrôle virtuelle pour simplifier le système d'origine de haut niveau, ainsi les sorties finales peuvent être dérivées systématiquement grâce à des fonctions Lyapunov appropriées. Cette méthode élimine les difficultés d'obtention d'une fonction de contrôle Lyapunov à l'aide d'un certain nombre d'étapes récurives qui ne dépassent jamais l'ordre du système [5].

Cependant, le système doit se présenter sous la forme dite paramétrique pure. Les équations d'un tel système sont données par [2] :

$$\begin{aligned}
 \dot{x}_1 &= \varphi_1(x_1)^T \theta + \psi_1(x_1)x_2 \\
 \dot{x}_2 &= \varphi_2(x_1, x_2)^T \theta + \psi_2(x_1, x_2)x_3 \\
 &\vdots \\
 \dot{x}_{n-1} &= \varphi_{n-1}(x_1, \dots, x_{n-1})^T \theta + \psi_{n-1}(x_1, \dots, x_{n-1})x_n \\
 \dot{x}_n &= \varphi_n(x_1, \dots, x_{n-1}, x_n)^T \theta + \psi_n(x_1, \dots, x_{n-1}, x_n)u \\
 y &= x_1
 \end{aligned} \tag{6.1}$$

Où θ est le vecteur de paramètres constants. Les ψ_i et les φ_i sont des fonctions non linéaires connues, avec $\varphi(0) = 0$ et $\psi_n(x) \neq 0, \forall x \in R^n$. De plus, le backstepping permet de garder les non-linéarités utiles.

6.3 Synthèse d'une loi de commande par backstepping

L'idée de base du backstepping est de commencer avec un système qui est stabilisable avec une loi de rétroaction connue pour une fonction lyapunov connue, puis d'ajouter à son entrée un intégrateur. Pour le système augmenté, une nouvelle loi de rétroaction stabilisante est explicitement conçue et s'est avéré se stabiliser pour une nouvelle fonction lyapunov, et ainsi de suite [4].

Dans cette partie on présente l'application de la commande par backstepping pour le robot manipulateur à 2 ddl. La dynamique du robot manipulateur est décrit par l'équation (2.47) dans le chapitre 2.

6.3.1 Loi de commande backstepping pour le 1^{er} segment (1^{er} sous-système)

• Représentation d'état du 1^{er} segment

Soit $x = [x_1 \ x_2 \ x_3 \ x_4] = [q_1 \ \dot{q}_1 \ q_2 \ \dot{q}_2]$ vecteur d'état du système, la représentation d'état correspondante du premier sous-système est donnée par [3] :

$$\dot{x}_1 = \dot{q}_1 = x_2 \quad (6.2)$$

$$\tau_1 = M_{11}\ddot{q}_1 + M_{12}\ddot{q}_2 + h_{11}\dot{q}_1 + h_{12}\dot{q}_2 + g_1 + F_1$$

$$\dot{x}_2 = \ddot{q}_1 = M_{11}^{-1}(\tau_1 - M_{12}\dot{x}_4 - h_{11}x_2 - h_{12}x_4 - g_1 - F_1) \quad (6.3)$$

• Etape 1 :

Soit x_{1d} la trajectoire de référence de x_1 et l'erreur de position de q_1 est donc :

$$\dot{e}_1 = \dot{x}_{1d} - \dot{x}_1 \quad (6.4)$$

On a l'équation (6.4) devient :

$$\dot{e}_1 = \dot{x}_{1d} - x_2 \quad (6.5)$$

La fonction de Lyapunov candidate est :

$$v_1 = \frac{1}{2} e_1^2 \quad (6.6)$$

Il est clair que la fonction de Lyapunov donné par (6.6) est définie positive et il faut garantir que sa dérivée soit négative pour que le 1^{er} sous système soit stable. On a :

$$\dot{v}_1 = e_1 \dot{e}_1 \quad (6.7)$$

En substituant l'équation (6.4) dans (6.7), on trouve :

$$\dot{v}_1 = e_1(\dot{x}_{1d} - x_2) \quad (6.8)$$

La fonction de Lyapunov doit prendre la valeur suivante pour qu'elle soit négative :

$$\dot{v}_1 = -k_1 e_1^2 \quad (6.9)$$

Avec k_1 est une constante définit positive.

En utilisant (6.8) et (6.9) on obtient :

$$-k_1 e_1 = \dot{x}_{1d} - x_2 \quad (6.10)$$

On trouve alors :

$$x_2 = \dot{x}_{1d} + k_1 e_1 \quad (6.11)$$

L'équation (6.11) représente une commande virtuelle qui va être utilisée comme une trajectoire de référence dans la 2^{ème} étape, Soit $x_2 = x_{2d}$

• **Etape 2 :**

Etant donné l'erreur de poursuite de x_2 :

$$e_2 = x_{2d} - x_2 \quad (6.12)$$

En utilisant les équations développées dans l'étape 1 pour déterminer les relations suivantes :

$$e_2 = \dot{x}_{1d} + k_1 e_1 - x_2 \quad (6.13)$$

$$e_2 = \dot{x}_{1d} - \dot{x}_1 + k_1 e_1 \quad (6.14)$$

$$e_2 = \dot{e}_1 + k_1 e_1 \quad (6.15)$$

$$\dot{e}_1 = e_2 - k_1 e_1 \quad (6.16)$$

En dérivant l'équation (6.13), nous trouvons :

$$\dot{e}_2 = k_1 \dot{e}_1 + \ddot{x}_{1d} - \ddot{x}_2 \quad (6.17)$$

La fonction de Lyapunov candidate est augmentée par un terme, ce qui donne :

$$v_2 = \frac{1}{2} e_1^2 + \frac{1}{2} e_2^2 \quad (6.18)$$

Et sa dérivée \dot{v}_2 égale à :

$$\dot{v}_2 = e_1 \dot{e}_1 + e_2 \dot{e}_2 \quad (6.19)$$

Le développement de cette équation donne :

$$\dot{v}_2 = e_1(e_2 - k_1 e_1) + e_2(k_1 \dot{e}_1 + \ddot{x}_{1d} - \ddot{x}_2) \quad (6.20)$$

$$\dot{v}_2 = -k_1 e_1^2 + e_2(k_1 \dot{e}_1 + \ddot{x}_{1d} - \ddot{x}_2 + e_1) \quad (6.21)$$

Pour que le 1^{er} sous système soit stable il faut que $\dot{v}_2 < 0$. Pour cela il faut que $\dot{e}_2 = (-k_2 e_2)$. avec k_2 est une constante définit positive, ce qui donne :

$$\dot{v}_2 = -k_1 e_1^2 - k_2 e_2^2 \quad (6.22)$$

L'égalité entre (6.21) et (6.22) donne :

$$\dot{x}_2 = k_1 \dot{e}_1 + \ddot{x}_{1d} + e_1 + k_2 e_2 \quad (6.23)$$

L'égalité entre (6.3) et (6.23), en tenant en compte (6.15), donne le développement suivant qui va aboutir à la détermination de la commande (couple) τ_1 du 1^{er} segment :

$$M_{11}^{-1}(\tau_1 - M_{12}\dot{x}_4 - h_{11}x_2 - h_{12}x_4 - g_1 - F_1) = k_1 \dot{e}_1 + \ddot{x}_{1d} + e_1 + k_2 e_2 \quad (6.24)$$

$$\tau_1 = M_{11}(k_{p1}e_1 + k_{D1}\dot{e}_1 + \ddot{x}_{1d}) + M_{12}\dot{x}_4 + h_{11}x_2 + h_{12}x_4 + g_1 + F_1 \quad (6.25)$$

Avec

$$k_{D1} = (k_1 + k_2) \quad (6.26)$$

$$k_{p1} = (1 + k_1 k_2) \quad (6.27)$$

6.3.2 Loi de commande backstepping pour le 2^{ème} segment (2^{ème} sous-système)

• Représentation d'état du 2^{ème} segment

La représentation d'état correspondante du 2^{em} sous système, est donnée par :

$$\dot{x}_3 = \dot{q}_2 = x_4 \quad (6.28)$$

$$\tau_2 = M_{21}\ddot{q}_1 + M_{22}\ddot{q}_2 + h_{21}\dot{q}_1 + h_{22}\dot{q}_2 + g_2 + F_2 \quad (6.29)$$

La détermination de la loi de commande de ce 2^{ème} segment a été développée en suivant les mêmes étapes lors de la détermination de la commande du 1^{er} segment, ce qui aboutit à:

$$\dot{x}_4 = k_3 \dot{e}_3 + \ddot{x}_{3d} + e_3 + k_4 e_4$$

$$\tau_2 = M_{21}(k_{p2}e_3 + k_{D2}\dot{e}_3 + \ddot{x}_{3d}) + M_{22}(k_3 \dot{e}_3 + \ddot{x}_{3d} + e_3 + k_4 e_4) + h_{21}x_2 + h_{22}x_4 + g_2 + F_2 \quad (6.30)$$

Avec

$$k_{D2} = (k_3 + k_4) \quad (6.31)$$

$$k_{p2} = (1 + k_3 k_4) \quad (6.32)$$

Nous obtenons enfin la loi de commande backstepping du robot manipulateur donnée par :

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} k_{p1}e_1 + k_{D1}\dot{e}_1 + \ddot{x}_{1d} \\ k_{p2}e_3 + k_{D2}\dot{e}_3 + \ddot{x}_{3d} \end{bmatrix} + \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} + \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \quad (6.33)$$

6.4 Résultats de simulation

L'objectif principal de cette application est l'utilisation de l'algorithme Moissure visqueuse, SMA, pour l'optimisation des paramètres du contrôleur de backstepping pour le contrôle de suivi et le rejet des perturbations du robot manipulateur à 2 degrés de liberté. Cela a été fait en utilisant l'environnement MATLAB / Simulink basé sur la minimisation des fonctions objectives *ISTE* et *ITAE* mentionnées comme :

$$J_1 = ISTE = \int e^2 t. dt \quad (6.34)$$

$$J_2 = ITAE = \int |e| t. dt \quad (6.35)$$

Les paramètres du robot pris en application sont indiqués dans le chapitre 4. Nous choisissons une onde sinusoïdale d'amplitude (1 rad) comme entrée de référence. Premièrement, nous appliquons les algorithmes d'optimisation SMA, ALO, WOA et GWO pour régler le contrôleur de backstepping pour le contrôle de suivi du robot manipulateur sans perturbations puis avec perturbations de couple.

6.4.1 Optimisation du contrôle backstepping sans perturbations

Après la phase d'optimisation, les valeurs des fonctions objectives pour différents algorithmes d'optimisation obtenus avec les critères *ISTE* sont présentées dans le tableau 6.1. De plus, l'intégral d'erreur absolue de temps, *ITAE*, calculé lors de la simulation avec les paramètres optimaux obtenus jusqu'à présent sont données dans le même tableau. Les paramètres optimaux du contrôleur par backstepping optimisé sont récapitulés dans le tableau 6.2.

Il ressort du tableau 6.1 que l'algorithme SMA dépasse les autres algorithmes en termes de meilleure fonction de coût, c'est à dire que l'algorithme SMA donne le meilleur optimum minimisant le critère *ISTE*. L'application des paramètres obtenus conduit à un minimum de fonction *ITAE*. En outre, nous pouvons voir la supériorité de l'algorithme ALO par rapport aux algorithmes WOA et GWO. De même, la dominance de WOA sur l'algorithme GWO.

La courbe de convergence de la phase d'optimisation a été représentée sur la figure 6.1 où la figure 6.2 illustre le suivi de trajectoire du robot manipulateur. Il semble clairement la bonne poursuite des articulations pour le trajet de référence, en particulier la première articulation. Cela se traduit par l'erreur de suivi illustrée à la figure 6.3.

Le couple d'entrée fourni par les actionneurs pour la première et la deuxième articulation est illustré à la figure 6.4. Nous pouvons voir une énergie de commande élevée au démarrage. Cela est dû au temps de montée rapide et à la petite erreur d'état stationnaire, qui peut être

produite en augmentant les paramètres du contrôleur et donc l'augmentation du signal de commande.

Tableau 6.1: Fonction de coût pour la première et la deuxième articulation sans perturbations.

Fonction de coût	ISTE		ITAE	
	1 st	2 nd	1 st	2 nd
Backstepping_SMA	4.211e-09	1.953e-08	0.0003769	0.0008331
Backstepping_ALO	1.01e-08	1.615e-08	0.0006506	0.0007562
Backstepping_WOA	1.008e-08	1.739e-08	0.0006502	0.0007999
Backstepping_GWO	1.008e-08	1.904e-08	0.0006502	0.0008411

Tableau 6.2: Paramètres du contrôleur par backstepping pour la première et la deuxième articulation sans perturbations.

	K_{p1}	K_{D2}	K_{p2}	K_{D2}
Backstepping_SMA	200	8.771592	200	8.568219
Backstepping_ALO	200	5	252.57	7.134
Backstepping_WOA	200	5	228.56	7.44
Backstepping_GWO	200	5	300	5.38

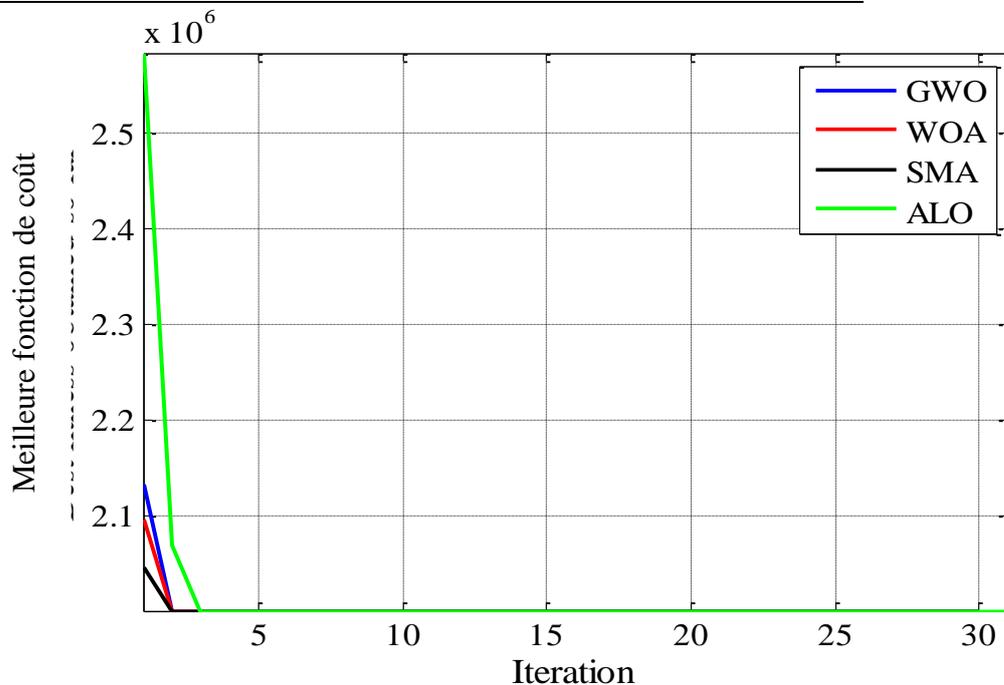


Figure 6.1: Courbe de convergence des algorithmes d'optimisation.

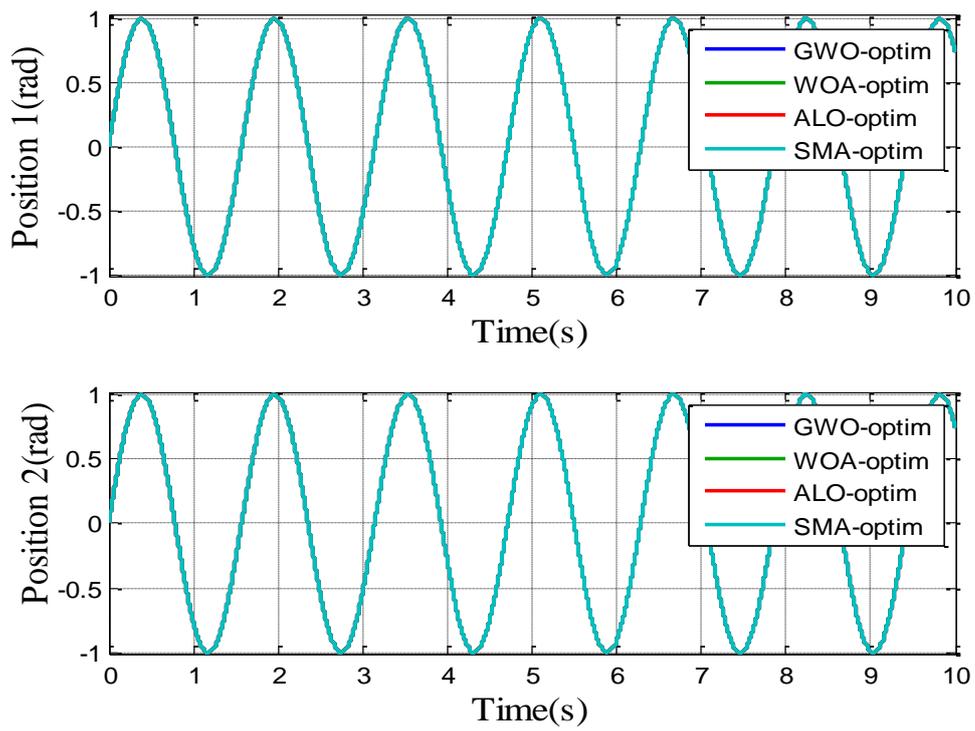


Figure 6.2: Suivi de trajectoire du robot manipulateur.

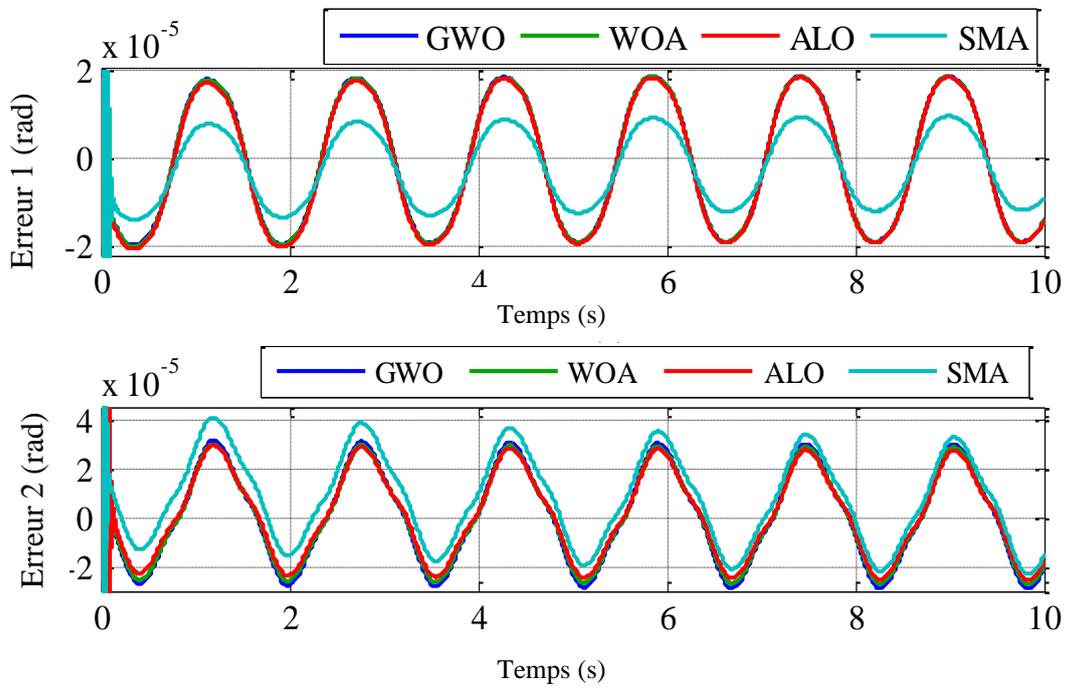


Figure 6.3: Erreur de suivi de trajectoire du robot manipulateur.

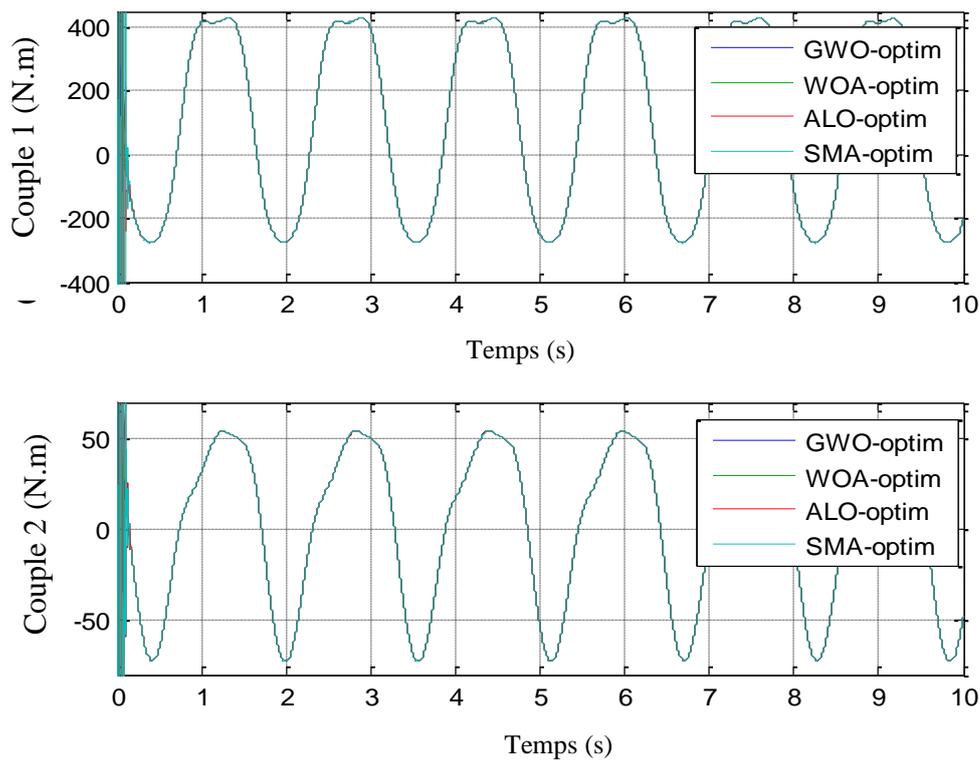


Figure 6.4: Couple de Contrôle des deux articulations du robot manipulateur.

6.4.2 Optimisation du contrôle Backstepping avec perturbations de couple

Dans cette partie, pour tester la robustesse du contrôleur avec des algorithmes d'optimisation, nous procédons lorsque le bras du robot est soumis à une perturbation de couple sous la forme d'un bruit blanc avec une variance de puissance de 0,8. Le tableau 6.3 répertorie les paramètres du contrôleur par backstepping et les valeurs de la fonction de coût avec perturbations.

Le contrôle par backstepping optimisé par l'algorithme SMA fournit la fonction de coût ISTE la plus petite de $4.93304e-05$ par rapport aux autres algorithmes. La position des deux articulations est représentée sur la figure 6.5 et le couple de commande correspondant est représenté sur la figure 6.6.

Nous pouvons voir d'après le test de robustesse que le contrôleur par backstepping optimisé par l'algorithme SMA est plus robuste pour éliminer l'effet des perturbations. En revanche, si l'on compare le contrôle backstepping avec le contrôle en mode glissant à surface PID optimisé par des algorithmes évolutifs réalisés dans un travail précédent par Loucif et al. [24], l'analyse comparative réalisée entre ces deux méthodes conduit aux résultats mentionnés dans le tableau 6.4 qui montre l'efficacité et l'utilité du contrôleur par backstepping par rapport

au contrôleur de mode glissant et sa robustesse en présence ou sans perturbations externes. Par conséquent, le robot manipulateur contrôlé par la commande de backstepping a de très bonnes performances en termes de rejet de perturbation et devient moins sensible aux bruits. Le tableau 6.5 récapitule les paramètres de tous les algorithmes utilisés dans ce travail.

Tableau 6.3 : Paramètres du contrôleur par backstepping pour la première et la deuxième articulation avec perturbations de couple.

	SMA	ALO	WOA	GWO
K_{p1}	200	200	200	200
K_{D1}	8.687602	7.886283	5	5
K_{p2}	200	200	200	212.5092
K_{D2}	9.751803	9.559624	8.599147	7.258901
$ISTE_1$	7.704e-07	9.445e-07	1.928e-06	1.93e-06
$ISTE_2$	4.856e-05	5.117e-05	6.017e-05	6.793e-05
$ITAE_1$	0.00548	0.006065	0.008721	0.008725
$ITAE_2$	0.04246	0.04357	0.04761	0.05116
$ISTE_{1+2}$	4.93304e-05	5.21145e-05	6.2098e-05	6.986e-05
$ITAE_{1+2}$	0.04794	0.049635	0.056331	0.059885

Tableau 6.4 : Comparaison entre la commande par backstepping et mode glissant.

Cost.Fun /contro	Sans perturbations		Avec perturbations	
	Backstepping_ALO	Sliding_PID_ALO	Backstepping_ALO	Sliding_PID_ALO
$ISTE_1$	1.01e-08	11.49e-07	9.445e-07	0.0123
$ISTE_2$	1.615e-08	6.858e-07	5.117e-05	4.224e-06
$ISTE_{1+2}$	2.625e-08	18.348e-07	5.21145e-05	1.2304e-2

Tableau 6.5 : Paramètres d'algorithmes d'optimisation pour le contrôle par backstepping.

SMA	ALO	WOA	GWO
N=300	N=3	Search agents N ^{bre} =30	Search agents Nbre=30
Max iterations T=30	Max iterations=30	Max iterations=30	Max iterations=30
dimsize=5			
Z=0.003			

Tableau 6.6: Temps de calcul total des algorithmes d'optimisation.

Algorithm	ALO	GWO	WOA	SMA
Temps(s)_Min	0.703	1.96	2.001	2.068
Temps(s)_Max	1.6	2.455	3.577	3.489

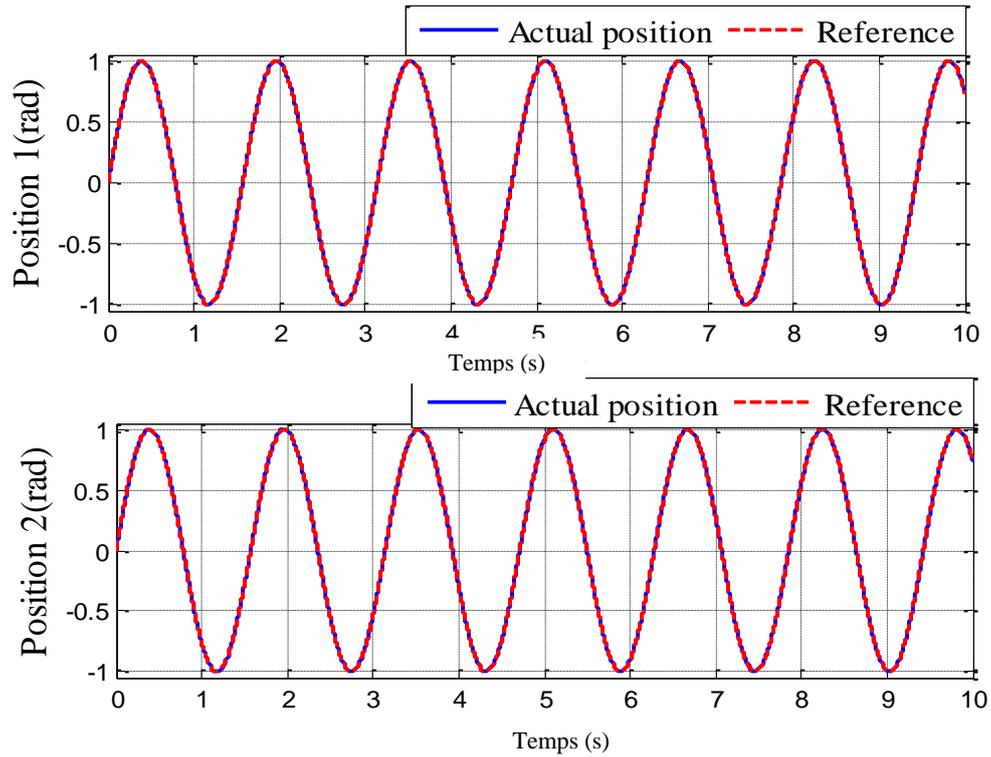


Figure 6.5: Suivi de trajectoire du robot manipulateur optimisé par l'algorithme SMA avec une perturbation de couple de 0,8 de puissance.

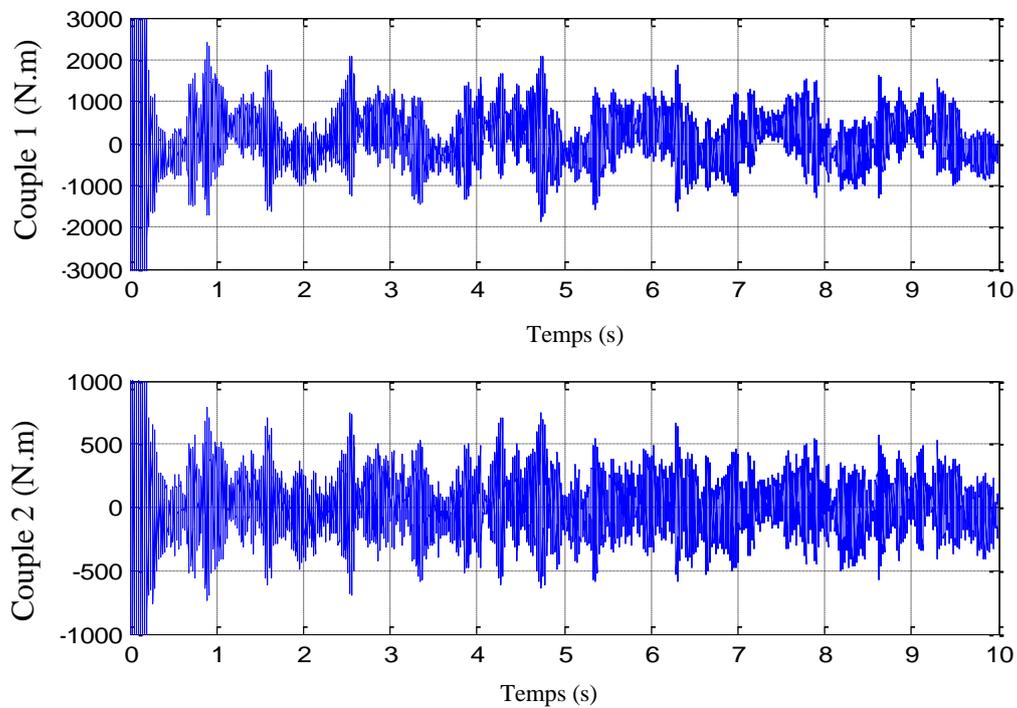


Figure 6.6: Couple de Contrôle des deux articulations du robot manipulateur optimisé par l'algorithme SMA avec une perturbation de couple de 0,8 de puissance.

Le temps total de calcul (cpu) d'optimisation des algorithmes utilisé dans cette étude est présenté dans le tableau 6.6. Le temps a été sélectionné après plus de 10 fois d'exécution. D'après le tableau, on voit que l'algorithme ALO fournit le temps de calcul le plus petit de 0.703 seconde et GWO de 1.96 seconde, précède les algorithmes SMA et WOA qui ont approximativement le même temps de calcul.

6.5 Conclusion

Dans ce chapitre, une étude analytique a été réalisée pour tester l'efficacité du contrôleur de backstepping optimisé par les techniques récentes d'optimisation comme les algorithmes SMA, ALO, WOA et GWO. Le contrôleur de backstepping optimisé par SMA présente plus de robustesse dans le contrôle de suivi de trajectoire du robot manipulateur à 2 ddl et le rejet des perturbations. Également à partir de l'étape de simulation, nous observons la rapidité de l'algorithme ALO pour le calcul de l'optimum des paramètres du contrôleur par rapport aux autres algorithmes, et la précision de l'algorithme SMA à obtenir l'optimum convenable. Cela nous conduit à l'idée de faire une hybridation entre ces deux algorithmes pour améliorer les performances de l'exploration et l'exploitation pour le calcul de l'optimum global.

Références du chapitre 6 :

1. **Krstic M, Kanellakopoulos I, Kokotovic P:** *Nonlinear and adaptive control design*. JOHN WILEY & SONS, INC,1995.
2. **Azoui N :** *Commande non linéaire d'un bras manipulateur*. Mémoire de magister en robotique. Université de Batna, 2008.
3. **Benadda H :** *Application des commandes non linéaires avec observateur d'un robot manipulateur*. Mémoire de magister en Automatique Robotique et Productique. Université des Sciences et de la Technologie d'Oran Mohamed boudiaf. 2013.
4. **kokotović PV:** *The joy of feedback: nonlinear and adaptive*. IEEE 1992.
5. **Karabacak M, Eskikurt HI:** *Design, modelling and simulation of a new nonlinear and full adaptive backstepping speed tracking controller for uncertain PMSM*. Applied Mathematical Modelling 36, 5199–5213, 2012.
6. **Shi H:** *A novel scheme for the design of backstepping control for a class of nonlinear systems*. Applied Mathematical Modelling. 35, 1893–1903, 2011.
7. **Haouari F, Bali N, Tadjine M, Boucherit MS:** *A CDM-backstepping control with nonlinear observer for electrically driven robot manipulator*. Automatic Control and Computer Sciences. 50. 332–346, 2016.
8. **Kali Y, Saad M, Kenné JP, K Benjelloun:** *Robot manipulator Control Using Backstepping with Lagrange's Extrapolation and PI Compensator*. New Developments and Advances in Robot Control. 137-161, 2019.
9. **Quynh NX, Nan WY, Yen VT:** *A novel Robust Adaptive Control Using RFWNNs and Backstepping for Industrial Robot Manipulators with Dead-Zone*. Journal of Intelligent & Robotic Systems, 2019.
10. **Xiong Gl, Xie Zw, Huang Jb, Liu H, Jiang Zn, Sun K:** *Dynamic surface control-backstepping based impedance control for 5-DOF flexible joint robots*. Journal of Central South University of Technology. 17. 807–815, 2010.
11. **Mai TL, Wang Y:** *Adaptive-backstepping force/motion control for mobile-manipulator robot based on fuzzy CMAC neural networks*. Control Theory and Technology, vol 12. 368–382, 2014.
12. **Maiti S, Das A:** *Controller Design for a Robot Manipulator System Using Adaptive Backstepping Method*. International Conference on Innovation in Modern Science and Technology ICIMSAT, 467-474, 2019.

13. **Zhao Y, Zhang S, Lee J:** *Adaptive finite-time backstepping control for a two-wheeled mobile manipulator.* Journal of Mechanical Science and Technology. 32. 5897–5906, 2018.
14. **Keighobadi J, Fateh MM, Xu B:** *Adaptive fuzzy voltage-based backstepping tracking control for uncertain robotic manipulators subject to partial state constraints and input delay.* Nonlinear Dynamics, 2020.
15. **Dian SY, Hu Y, Zhao T, Han JX:** *Adaptive backstepping control for flexible-joint manipulator using interval type-2 fuzzy neural network approximator.* Nonlinear Dynamics. 97. 1567–1580, 2019.
16. **Chang WM, Tong SC, Li YM:** *Adaptive fuzzy backstepping output constraint control of flexible manipulator with actuator saturation.* Neural Computing and Applications. 28. 1165–1175, 2017.
17. **Awan ZS, Ali K, Iqbal J, Mehmood A:** *Adaptive Backstepping Based Sensor and Actuator Fault Tolerant Control of a Manipulator.* Journal of Electrical Engineering & Technology, vol 14. 2497–2504, 2019.
18. **Su CY, stepanenko Y:** *Backstepping-based Hybride adaptive control of robot manipulators.* International of adaptive control and signal processing. 11. 141-153, 1997.
19. **Shi H:** *A novel scheme for the design of backstepping control for a class of nonlinear systems.* Applied Mathematical Modelling. 35, 1893–1903, 2011.
20. **Vijay M, Jena D:** *Optimal backstepping sliding mode control for robot manipulator.* IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES). 1-5, 2015.
21. **Huang Y, Cheng L, Li Z, Gao W, Lu H, Wei L:** *Backstepping Sliding Mode Control for Robot Manipulator via Nonlinear Disturbance Observer.* Chinese Control Conference (CCC). 3220-3224, 2019.
22. **Gholipour R, Khosravi A, Mojallali H:** *Multi-objective optimal backstepping controller design for chaos control in a rod-type plasma torch system using Bees Algorithm.* Applied Mathematical Modelling. 39. 4432-4444, 2015.
23. **Alinejad P, Keymanesh A, Nouri H, Gholipour R:** *Backstepping Control for Couillet Chaotic Nonlinear System via PSOPC Algorithm.* International journal of mechatronics, electrical and computer technology. vol 5. 2044-2055, 2015.
24. **Loucif F, Kechida S:** *Optimization of Sliding mode control with PID surface for robot manipulator by Evolutionary Algorithms.* Open computer science. Vol 10, 396–407, 4 Novembre 2020.

Conclusion générale

Conclusion générale

Au cours de ce travail de thèse, une étude analytique de la commande d'un robot manipulateur a été faite en appliquant différentes techniques de contrôle.

Dans le cadre de cette thèse, différentes techniques de commandes ont été appliqués pour la poursuite de trajectoire du robot manipulateur, nous rappelons : la commande PID, la commande par découplage non linéaire (à couple calculé) type PD et PID, la commande par compensation de gravité type PD, la commande par mode glissant et la commande par backstepping. Suite à cette étude, on peut clairement déduire que chaque technique a ses avantages et ses inconvénients et les conditions privilégiées pour que cette commande présente ses meilleures performances. Les commandes basées sur la fonction de Lyapunov tels que le mode glissant et le backstepping ont prouvés leur domination pour le suivi de trajectoire ou le rejet de perturbations par rapport aux techniques classiques précités. Avec le développement rapide de l'industrie moderne, l'efficacité et la qualité du travail du bras robotique sont devenus de plus en plus une exigence et les tâches à accomplir deviennent également de plus en plus compliqué. Cependant le contrôle classique (PD ou PID) ne répond pas aux exigences souhaitées, notamment un mouvement rapide et de haute précision. En effet, le contrôleur PID traditionnel a pour but d'augmenter le gain de contrôle mais le couple de l'actionneur est limité dans le système réel. Par conséquent, la synthèse d'un contrôleur précis et robuste est la tendance inévitable dans le développement de commande des robots manipulateurs.

La commande par mode glissant conventionnelle présente un phénomène de broutement qui peut dégrader les performances du système. De plus, elle ne peut garantir l'invariance de la performance du robot que pour les incertitudes et les perturbations appariées, et ne peut pas atténuer efficacement les incertitudes et les perturbations dépareillées. Quant à la commande backstepping est une stratégie de contrôle qui prend en compte la nature non linéaire du système pour gérer les incertitudes dépareillées et l'effet des perturbations environnementales de manière robuste. En revanche les performances de commande nominales sont compromises les effets de perturbations externes inconnues et les incertitudes du modèle dans les systèmes de commande sans détériorer le contrôleur existant.

Nous admettons que le contrôle du mode glissant et du backstepping sont des techniques efficaces et robustes. Néanmoins, la combinaison entre eux peut non seulement simplifier la conception du contrôle de backstepping, mais également augmenter la robustesse du système contre les perturbations externes. L'utilisation du SMC dans les étapes de backstepping peut rendre le SMC insensible aux incertitudes dépareillées.

Finalement, Le recours à une méthode ou à l'autre pour la commande des robots manipulateurs se fait normalement en fonction des contraintes du cahier des charges, auxquelles s'ajoutent parfois des nouvelles exigences de l'économie d'énergie et de l'économie du matériel qui devraient être prises en compte.

Au cours de ce travail, le choix des gains des contrôleurs pour les différentes techniques de commande est réalisé à l'aide des algorithmes d'optimisation en minimisant un critère de coût. Vu que l'objectif est le suivi de trajectoire de l'organe terminal du robot à une trajectoire de référence donnée par une commande articulaire, nous avons choisis deux critères de coût en fonction de l'erreur de position qui sont *l'ITAE* et *l'ISTE*. Pour plus de sécurité des moteurs de robot, il est mieux de prendre en compte le signal de commande, c.-à-d. le couple moteur à appliquer aux articulations, dans l'étape d'optimisation. Pour cela le critère de coût doit être une fonction quadratique de l'erreur de position et du couple moteur.

On a vu clairement l'efficacité et l'intérêt des algorithmes d'optimisation dans le domaine de contrôle des systèmes non linéaire notamment les robots manipulateurs. On a trouvé un compromis entre la population (nombre des particules ou les agents) et la vitesse de convergence vers l'optimum globale, notamment le PSO par rapport aux nouveaux algorithmes tels que le GWO et WOA. En outre, la présence du caractère aléatoire dans la dernière étape dans ces algorithmes peut amener l'algorithme à rechercher toutes les solutions optimales dans l'espace de recherche, évitant ainsi efficacement l'optimum local.

Les algorithmes métha-heuristique commencent généralement le processus d'optimisation avec des solutions générées aléatoirement et ne nécessitent pas d'informations du gradient, ce qui rend l'algorithme parfaitement adapté aux problèmes pratiques lorsque l'information de la dérivée est inconnue, surtout dans les problèmes non linéaires. En premier temps, les résultats de simulation obtenus ont montré la supériorité de l'algorithme de WOA par rapport aux algorithmes GWO et PSO en optimisant les paramètres du PID classique. Ensuite l'introduction de nouveaux algorithmes SCA et ALO pour l'optimisation du contrôleur par mode glissant montre l'efficacité des deux algorithmes par rapport aux algorithmes WOA, GWO et la

supériorité de ALO aux restes. Finalement, la construction d'un contrôleur par backstepping optimisé a prouvé la domination de l'algorithme SMA par rapport à WOA, GWO et ALO en terme de minimisation de l'erreur de poursuite du robot. L'optimisation des paramètres du contrôleur backstepping faite dans le dernier chapitre, nous a permis de constater les distinctions entre les algorithmes d'optimisations utilisés. Nous pouvons observer la vitesse de l'algorithme ALO par rapport aux autres algorithmes dans le calcul de l'optimum des paramètres et aussi la précision de l'algorithme SMA à obtenir l'optimum convenable. Cela nous conduit à concevoir un algorithme hybride à partir de ces deux algorithmes, ainsi que les autres algorithmes, pour améliorer les performances de l'exploration et l'exploitation de l'optimum global.

Le fruit de cette étude analytique des algorithmes métha-heuristique dans le domaine de la robotique ne juge pas le comportement de ces algorithmes dans d'autre domaine, disciplines ou d'autres applications. C'est ce que certains théorèmes l'indiquent, tel que : No free lunch (NFL) référencé au chapitre 3, qui ont été développées pour explorer le lien entre les algorithmes d'optimisation efficaces et les problèmes qu'ils résolvent. Ils établissent que pour tout algorithme, toute performance élevée sur une classe de problèmes est compensée par une performance sur une autre classe. Ces théorèmes aboutissent à une interprétation géométrique de ce que signifie pour un algorithme d'être bien adapté à un problème d'optimisation. Un autre défi est que l'équilibre des deux capacités, l'exploration et l'exploitation de l'algorithme d'optimisation, n'est pas nécessairement identique au différents problèmes. Par conséquent, il est relativement difficile de parvenir à un équilibre approprié entre les deux phases, efficaces pour tous les problèmes d'optimisation.

Perspectives pour de futurs travaux : Les perspectives envisagées peuvent être résumées comme suit :

- L'application et l'optimisation d'une technique hybride combinant la commande backstepping qui s'est avérée efficace pour les systèmes non linéaires, notamment les robots, et la commande par mode glissant pour éliminer le broutement et améliorer les performances.
- L'application des nouveaux algorithmes d'optimisations pour d'autres techniques de commandes du robot.
- L'hybridation entre les algorithmes métha-heuristiques pour avoir des algorithmes puissants.

- Loin de la robotique, nous pensons à appliquer ces algorithmes dans les procédures de diagnostic des symptômes des maladies complexes tel que le cancer.
- Outre, le champ de l'énergies renouvelable est un domaine intéressant pour la commande ou l'optimisation.

Annexe

Méthode de Denavit-Hartenberg

La modélisation des robots de façon systématique et automatique exige une méthode adéquate pour la description de leur morphologie. Plusieurs méthodes et notations ont été proposées. La plus répandue est celle de Denavit-Hartenberg. Mais cette méthode, développée pour des structures ouvertes simples, présente des ambiguïtés lorsqu'elle est appliquée sur des robots ayant des structures fermées ou arborescentes.

Une structure ouverte simple est composée de $n + 1$ corps notés C_0, \dots, C_n et de n articulations. Le corps C_0 désigne la base du robot et le corps C_n le corps qui porte l'organe terminal. L'articulation j connecte le corps C_j au corps C_{j-1} (figure 1). 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 idéales (pas de jeu mécanique, pas d'élasticité), soit rotoïdes, soit prismatiques :

- le repère R_j est lié au corps C_j .
- l'axe z_j est porté par l'axe de l'articulation j .
- l'axe x_j est porté par la perpendiculaire commune aux axes z_j et z_{j+1} . Si les axes z_j et z_{j+1} sont parallèles ou colinéaires, le choix de x_j n'est pas unique : des considérations de symétrie ou de simplicité permettent alors un choix rationnel.

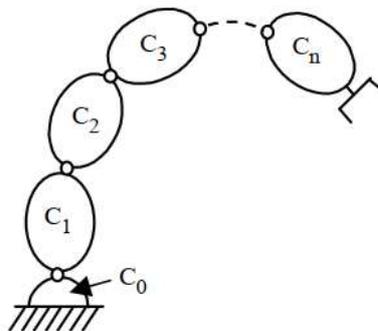


Figure 1.A : Robot à structure ouverte simple.

Le passage du repère R_{j-1} au repère R_j s'exprime en fonction des quatre paramètres géométriques suivants (figure 2) :

- α_j : angle entre les axes z_{j-1} et z_j correspondant à une rotation autour de x_{j-1} .
- d_j : distance entre z_{j-1} et z_j le long de x_{j-1} .
- θ_j : angle entre les axes x_{j-1} et x_j correspondant à une rotation autour de z_j .
- r_j : distance entre x_{j-1} et x_j le long de z_j .

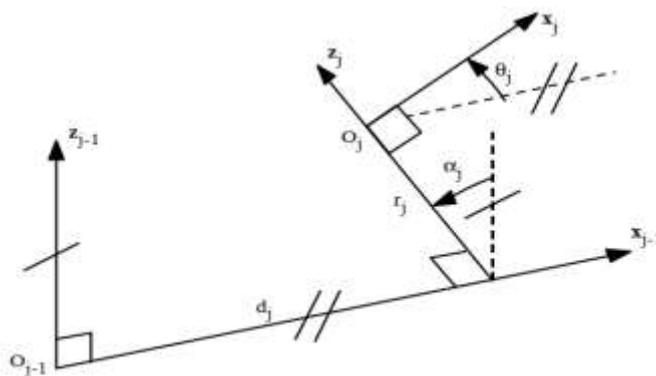


Figure 2.A : Paramètres géométriques dans le cas d'une structure ouverte simple.

La variable articulaire q_j associée à la $j^{\text{ème}}$ articulation est soit θ_j , soit r_j , selon que cette articulation est de type rotoïde ou prismatique, ce qui se traduit par la relation :

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

Avec :

- $\sigma_j = 0$ si l'articulation j est rotoïde ;
- $\sigma_j = 1$ si l'articulation j est prismatique ;
- $\bar{\sigma}_j = 1 - \sigma_j$.

La matrice de transformation définissant le repère R_j dans le repère R_{j-1} est donnée par (figure 2) :

$$\begin{aligned} {}^{j-1}T_j &= \mathbf{Rot}(x, \alpha_j) \mathbf{Trans}(x, d_j) \mathbf{Rot}(z, \theta_j) \mathbf{Trans}(z, r_j) \\ &= \begin{bmatrix} C\theta_j & -S\theta_j & 0 & d_j \\ C\alpha_j S\theta_j & C\alpha_j C\theta_j & -S\alpha_j & -r_j S\alpha_j \\ S\alpha_j S\theta_j & S\alpha_j C\theta_j & C\alpha_j & r_j C\alpha_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

Où $\mathbf{Rot}(u, \alpha)$ et $\mathbf{Trans}(u, d)$ sont des matrices de transformation homogène (4x4) représentant respectivement une rotation α autour de l'axe u et une translation d le long de u .

ففي الختام الحمد لله عز وجل الذي اعانني على اتمام هذا العمل واتمنى ان ينتفع به كل من يبحث في هذا المجال. وفي النهاية اذكر كل من يقرأه ونفسي بقول رسول الله صلى الله عليه وسلم قال اتاني جبريل فقال:

«يا محمد! عش ما شئت فإنك ميت، واحبب من شئت فإنك مفارقة، واعمل ما شئت فإنك مجزي به، واعلم ان شرف المؤمن قيامه بالليل، وعزه استغناؤه عن الناس»

En conclusion, je remercie Dieu Tout-Puissant qui m'a aidé à mener à bien ce travail, et j'espère que tous ceux qui cherchent dans ce domaine en profiteront. En fin de compte, je rappelle tous ceux qui le lisent et de moi-même en disant le Messenger de Dieu, que Dieu le bénisse et lui accorde la paix, me dit Gabriel :

"Ô Muhammad! Vivez ce que vous voulez, vous êtes mort, et aimez ce que vous voulez, parce que vous le laissez, et faites ce que vous voulez, vous en serez récompensé, et sachez que l'honneur du croyant est de prier la nuit, et sa fierté de se passer des gens ".