

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبي بكر بلقايد -

تلمس -

Université Aboubakr Belkaïd – Tlemcen –

Faculté de TECHNOLOGIE



THESE

Présentée pour l'obtention du grade de **DOCTORAT en sciences**

En : Automatique

Spécialité : Automatique

Par : SAIDI Khayreddine

Sujet

Commandes non linéaires optimales appliquées à un système mécanique articulé

Soutenue publiquement, le 09/12/2021, devant le jury composé de :

Mme CHOUKCHOU BRAHAM Amal	Professeur	Univ. Tlemcen	Président
Mr BOUMEDIENE Abdelmadjid	Professeur	Univ. Tlemcen	Directeur de thèse
Mr GUENFAF Lakhdar	Professeur	USTHB d'Alger	Examineur
Mr MANSOURI Abdellah	Professeur	ENSP d'Oran	Examineur

Remerciements

Mon premier remerciement va tout d'abord à Allah, le miséricordieux, pour m'avoir donné la foi, le courage et la patience qui m'ont permis de réaliser et de finir ce travail.

Je suis très heureux de pouvoir exprimer mes remerciements à tous ceux qui ont contribué à l'aboutissement de ce travail de doctorat.

Ce travail a été effectué sous la direction de **Mr Abdelmadjid BOUMEDIENE** Professeur à l'Université Abou Bekr Belkaïd Tlemcen. Je le remercie très sincèrement pour la confiance qu'il a manifestée à mon égard, pour son soutien, et pour la patience inouïe qu'il m'a témoignée tout au long de l'élaboration de ce travail. Nos discussions stimulantes ont contribué à la clarification de beaucoup de notions. Qu'il reçoive l'expression de ma très grande reconnaissance.

Je remercie très sincèrement **Mme CHOUKCHOU BRAHAM Amal**, Professeur à l'Université Abou Bekr Belkaïd Tlemcen, de m'avoir honoré par sa présence en qualité de présidente de jury.

J'adresse mes sincères remerciements aux examinateurs, **Mr GUENFAF Lakhdar**, Professeur à l'Université des Sciences et de la Technologie Houari Boumediene d'Alger et **Mr MANSOURI Abdellah** Professeur à l'Ecole Nationale Supérieure Polytechnique d'Oran, d'avoir acceptés d'examiner ce travail.

Mes remerciements les plus chaleureux vont également à mes chers parents et ma femme qui m'ont soutenu tout le long de ces années de travail. Qu'ils trouvent ici l'expression de ma profonde reconnaissance.

Un grand merci à Mme BOUBEKEUR Djamila pour ses encouragements et son soutien et à Mr MASSOUM Ahmed pour sa précieuse aide.

Enfin dans le souci de n'oublier personne, que tous ceux qui m'ont aidé, de près ou de loin, que ce soit par leurs amitiés, leurs conseils ou leurs soutiens trouvent dans ces quelques aimables et sincères mots l'expression de ma profonde gratitude.

Dédicaces

*À mes chers parents,
À ma femme,
À toute ma famille et mes amis.*

Résumé

La conception d'un contrôleur consiste un défi pour n'importe quel automaticien, lorsqu'on ne dispose pas de modèle mathématique précis du processus à commander ou lorsque ce dernier présente de très fortes non linéarités, des couplages ou même des imprécisions. La dynamique des robots manipulateurs peut être décrite par un ensemble d'équations différentielles non linéaires couplées du second ordre. En raison de la présence d'un couplage dans le système, toutes les composantes du vecteur de commande apparaissent dans chaque équation du système. Par conséquent, l'application d'une simple commande conçue pour les systèmes SISO est impossible. Les algorithmes de commande existants sont complexes à mettre en œuvre. Une combinaison entre l'intelligence artificielle et les contrôleurs non linéaires est proposée pour commander un robot manipulateur. Les contrôleurs associent des techniques d'optimisation aux approches dites de commandes robustes non linéaires, SMC et Backstepping. La motivation derrière l'utilisation de la commande par modes glissants et la commande Backstepping dans les problèmes de contrôle en robotique repose principalement sur leurs caractéristiques appréciables, comme la simplicité de conception et la robustesse. Les algorithmes d'optimisation basés sur la métaheuristique sont une approche très intéressante pour résoudre des problèmes difficiles d'optimisation, parmi les métaheuristicues les plus connues, bien adéquates à l'optimisation dans les applications d'ingénierie, nous pouvons mentionner optimisation par Colonie de fourmis (Dorigo, 1992), Particle Swarm Optimization (Kennedy et Eberhart, 1995) Colonie d'abeilles artificielle (Karaboga, 2005) et algorithmes génétiques (Holland, 1975). Par rapport à d'autres méthodes d'optimisation, les méthodes de l'algorithme génétique (GA) et PSO en tant que méthodes de recherche globale sont réputés d'être plus efficace pour trouver l'optimum global. A travers Cette thèse nous abordons la faisabilité et l'applicabilité des algorithmes génétiques et PSO en tant qu'outil d'optimisation dans les applications de commande des processus. L'objectif ici est de trouver une structure de commande optimal basée sur les algorithmes génétiques et le PSO et ensuite de la testée sur un problème de commande. Les résultats indiquent que les techniques d'optimisation sont une alternative intéressante et efficace dans de nombreux problèmes de commande. Des simulations numériques utilisant le modèle dynamique d'un robot manipulateur à 2 ddl montrent l'efficacité de la stratégie de commande optimale proposée basée sur les approches SMC et Backstepping d'une part et GA et PSO d'autre part dans les problèmes de régulation et de suivi de trajectoire. Une deuxième partie consiste à introduire la dynamique des actionneurs. Le choix d'utiliser les moteurs de type MSAP pour cette application est justifié par ses avantages tels qu'un rendement élevé, couple important ainsi que l'absence du mécanisme balais-collecteur. Les lois de commande par mode glissant classique et par l'algorithme du Super-Twisting ont été appliquées sur le système global (robot + actionneur). Les résultats de simulation obtenus sont satisfaisants.

Mots clés : Robot manipulateur, système non linéaire, commande par modes glissants, Backstepping, Optimisation, Algorithmes Génétiques, Population, PSO, Particule, Essaim, Super-Twisting, MSAP.

Abstract

Control of robot manipulators includes uncertainties, nonlinearities and external perturbations that should be considered in the design of control laws. The dynamics of the robot manipulators can be described by a set of coupled nonlinear second-order differential equations. Because of the presence of inertial coupling in the system, all the components of the control vector appear in every equation of the system. Therefore, the straightforward application of the design of a control for single-input single-output systems is impossible. The existing control algorithms are complex and costly for implementation. A combination between the artificial intelligence and the nonlinear controllers is proposed for the control of robot manipulators with model uncertainties. The controllers combines an optimization techniques with the so-called nonlinear robust controllers, SMC and Backstepping, approaches. The motivation for using the sliding mode and Backstepping control in robotic control problems mainly relies on its appreciable features, like design simplicity and robustness. Metaheuristics based optimization algorithms are an approaches to solve hard optimization problems, among the well-known population-based metaheuristics, adequate to optimization in engineering applications, we can also mention Ant Colony Optimization (Dorigo, 1992), Particle Swarm Optimization (Kennedy and Eberhart, 1995) Artificial Bee Colony (Karaboga, 2005) and genetic algorithms (Holland, 1975). As compared to other optimization methods, Genetic Algorithm method (GA) and PSO as an auto-adapted global searching system by simulating biological evolution, natural behavior and the fittest element principle in natural environment seeks best solution more effectively. This thesis discusses the workability and applicability of genetic algorithms and PSO for process control applications as well as the concept and design procedure of Genetic Algorithm and PSO as an optimization tool. Here, the genetic algorithm (GA) and PSO, perspective is to find the optimum control structure for a time horizon is presented and afterwards is tested on a control problem. The results indicate that the optimization techniques are a valid and efficient alternative to the classical optimization methods in many problems in control. Numerical simulations using the dynamic model of a two-link planar rigid robot manipulator shows the effectiveness of the proposed optimal control strategy based on SMC and Backstepping approaches on the one hand and GA and PSO on the other side in regulation and trajectory tracking problems. A second part consists in introducing the dynamics of the actuators. The choice to use PMSM type motors for this application is justified by its advantages such as high efficiency, high torque and the absence of the brush-collector mechanism. The classic sliding mode control laws and the Super-Twisting algorithm were applied to the overall system (robot + actuator). The simulation results obtained are satisfactory.

Keywords: Robot manipulator, Nonlinear, Sliding Mode Control, Backstepping control, optimization, Genetic Algorithm, Population, PSO, Particle, Swarm, Super-Twisting, PMSM.

ملخص

يمثل تصميم وحدة التحكم تحديًا لأي مهندس آلية، خاصة عندما لا يكون هناك نموذج رياضي دقيق للنظام المراد التحكم فيه أو عندما يكون هذا الأخير ذو طبيعة غير خطية أو هنالك تداخل بين متغيراته أو حتى عدم الدقة. يمكن وصف ديناميكية الروبوتات الآلية أو الأذرع الآلية باستعمال مجموعة من المعادلات التفاضلية غير الخطية المقترنة من الدرجة الثانية. نظرًا لوجود تداخل بين متغيرات النظام، تظهر جميع مكونات شعاع التحكم في كل معادلة من معادلاته. لذلك، لا يمكن تطبيق نظام تحكم بسيط مصمم للأنظمة ذات مدخل واحد ومخرج واحد. كما أن خوارزميات التحكم الحالية معقدة للتنفيذ. تم اقتراح مزيج من الذكاء الاصطناعي وأنظمة تحكم غير خطية للتحكم في الذراع الآلي. تجمع وحدات التحكم المقترحة بين تقنيات التحسين مع ما يسمى بعناصر التحكم غير الخطية القوية كالتحكم ذو نمط انزلاقي والتحكم التراجعي. الدافع وراء استخدام التحكم ذو نمط انزلاقي والتحكم التراجعي في مشاكل التحكم في الروبوتات يعتمد بشكل أساسي على ميزات القيمة، مثل بساطة التصميم والمتانة. تعد خوارزميات التحسين المستندة إلى طرق البحث العشوائي نهجًا مثيرًا للاهتمام لحل مشكلات التحسين الصعبة، ومن بين أكثر علميات التحليل شهرة، والمناسب تمامًا في التطبيقات الهندسية، يمكننا أن نذكر التحسين بواسطة مستعمرة النمل، سرب الجسيمات، مستعمرة النحل الاصطناعي والخوارزميات الجينية. مقارنة مع طرق التحسين الأخرى، من المعروف أن طرق الخوارزمية الجينية وطرق سرب الجسيمات تعتبر الأكثر كفاءة في العثور على المستوى الأمثل بشكل عام. من خلال هذه الأطروحة نتناول جدوى وإمكانية تطبيق الخوارزميات الجينية وسرب الجسيمات كأداة تحسين في تطبيقات التحكم في الأنظمة. الهدف هنا هو إيجاد هيكل تحكم مثالي يعتمد على الخوارزميات الجينية وسرب الجسيمات ثم اختياره على مشكلة التحكم. تشير النتائج إلى أن تقنيات التحسين هي بديل جذاب وفعال في العديد من مشاكل التحكم. تظهر عمليات المحاكاة الرقمية باستخدام النموذج الديناميكي لذراع آلي ذو درجتى حرية فعالية استراتيجية التحكم المثلى المقترحة بناءً على نهج التحكم ذو نمط انزلاقي والتحكم التراجعي من ناحية وطرق الخوارزمية الجينية وطرق سرب الجسيمات من ناحية أخرى في مشاكل التحكم وتتبع المسار. من خلال الجزء الثاني سوف نحاول ادخال ديناميكية المحركات. إن اختيار استخدام محركات من نوع المحرك المتزامنة لهذا التطبيق له ما يبرره من خلال مزاياه مثل الكفاءة العالية وعزم الدوران العالي وعدة مزايا أخرى. تم تطبيق قوانين التحكم ذو نمط انزلاقي كلاسيكي وخوارزمية الالتواء الفائق على النظام العام المكون من (ذراع آلي + محرك). نتائج المحاكاة التي تم الحصول عليها مرضية.

الكلمات المفتاحية: الذراع الآلي، نظام غير خطي، التحكم ذو نمط انزلاقي، التحكم التراجعي، طرق التحسين، الخوارزمية الجينية، سرب الجسيمات، الالتواء الفائق، المحرك المتزامنة.

Table des matières

Liste des figures.....	vi
Liste des tableaux.....	ix
Notations.....	x
Abréviations.....	xi
INTRODUCTION GÉNÉRALE	1
CHAPITRE 1	
ELÉMENTS DES SYSTÈMES MÉCANIQUES ARTICULÉS	
1.1. Introduction	5
1.2. Terminologie générale.....	5
1.2.1. Articulations	6
1.2.1.1. Articulation de type rotoïde.....	6
1.2.1.2. Articulation prismatique	6
1.2.2. Organe terminal	7
1.2.3. Variable articulaire	7
1.2.4. Espace articulaire	7
1.2.5. Espace articulaire	7
1.2.6. Redondance d'un robot manipulateur	7
1.2.7. Les actionneurs.....	8
1.2.7.1. Actionneurs électriques (Les moteurs électriques).....	8
1.2.7.2. Actionneurs hydrauliques et pneumatiques (les Vérins)	8
1.2.8. Les capteurs	8
1.2.8.1. Les capteurs proprioceptifs.....	9
1.2.8.2. Les capteurs extéroceptifs	9
1.3. Modélisation des robots manipulateurs	10
1.3.1. Les coordonnées homogènes	10
1.3.1.1. Représentation d'un point.....	10
1.3.1.2. Représentation d'un vecteur libre.....	11
1.3.1.3. Représentation d'un plan.....	11
1.3.2. Transformation homogène.....	11
1.3.2.1. Transformation d'un repère à un autre	11
1.3.2.2. Transformation des vecteurs.....	12
1.3.2.3. Matrice de transformation dans le cas d'une translation pure	12

1.3.2.4. Matrice de transformation dans le cas d'une rotation pure.....	13
1.3.3. Modèle géométrique d'une structure ouverte simple	14
1.3.3.1. Convention de Denavit-Hartenberg.....	14
1.3.3.2. Convention de Denavit-Hartenberg modifiée.....	16
1.3.3.3. Expression du modèle géométrique direct (MGD).....	17
1.3.3.4. Modèle géométrique inverse (MGI).....	17
1.3.4. Modélisation cinématique	18
1.3.4.1. Modèle cinématique direct (MCD).....	18
1.3.4.2. Modèle cinématique inverse (MCI).....	19
1.3.5. Modélisation dynamique	19
1.3.5.1. Formalisme de Newton-Euler.....	20
1.3.5.2. Formalisme d'Euler-Lagrange.....	21
1.4. Conclusion.....	27

CHAPITRE 2

QUELQUES LOIS DE COMMANDE CLASSIQUE DES ROBOTS MANIPULATEURS À CHAÎNE OUVERTE SIMPLE

2.1. Introduction	29
2.2. Lois de commande dans le cas du problème de régulation	30
2.2.1. La loi de commande de type PID	30
2.2.2. La loi de commande proportionnelle dérivée (PD) avec compensation de l'effet de gravité (point à point).....	32
2.3. Lois de commande dans le cas du problème de poursuite de trajectoire	34
2.3.1. Générateur de trajectoire	34
2.3.1.1. Interpolation polynomiale	35
2.3.1.2. Interpolation Bang-Bang	38
2.3.2. La loi de commande du couple calculé (computed torque control).....	39
2.4. Résultats de simulation des lois de commande classique	41
2.4.1. Bras manipulateur à 2ddl.....	42
2.4.1.1. Application de la loi de commande PID.....	42
2.4.1.2. Application de la loi de commande point à point	43
2.4.1.3. Application de la loi de commande du couple calculé	45
2.5. Conclusion.....	46

CHAPITRE 3

CONCEPTS THÉORIQUES SUR LES COMMANDES ROBUSTES PAR MODES GLISSANTS ET BACKSTEPPING

3.1. Introduction	47
-------------------------	----

3.2. Commande par modes glissants	47
3.2.1. Concepts de base de la commande par modes glissants	48
3.2.1.1. Définitions	48
3.2.1.2. Choix de la surface de glissement et degré relatif	51
3.2.1.3. Conditions d'existence du mode glissant	52
3.2.1.4. Synthèse de la commande par modes glissants (SMC)	55
3.2.2. Phénomène de chattering.....	57
3.2.2. Robustesse des modes glissants vis-à-vis des perturbations	59
3.3. La commande de Backstepping.....	60
3.3.1. Principe de base de la commande par Backstepping.....	60
3.3.1.1. Commande stabilisante par Backstepping.....	60
3.4. Conclusion.....	65

CHAPITRE 4

MÉTHODES D'OPTIMISATION MÉTAHEURISTIQUES

4.1. Introduction	66
4.2. Optimisation	67
4.2.1. Définition.....	67
4.2.2. Problème d'optimisation	68
4.2.3. Méthodes d'optimisation.....	68
4.2.3.1. Méthodes déterministes (classiques)	68
4.2.3.2. Méthodes non-déterministes (Métaheuristiques).....	69
4.3. Les algorithmes génétiques	71
4.3.1. Terminologie et concepts de base des algorithmes génétiques	71
4.3.1.1. Génotype et phénotype	72
4.3.1.2. La population.....	72
4.3.1.3. Les individus	72
4.3.1.4. Les chromosomes	73
4.3.1.5. Les gènes	73
4.3.2. Méthodologie adoptée	73
4.3.2.1. Le codage	75
4.3.2.2. La fonction objectif (Fitness)	76
4.3.2.3. Générer une population initiale	76
4.3.2.4. L'opérateur de sélection	77
4.3.2.5. L'opérateur de croisement.....	79
4.3.2.6. L'opérateur de mutation	80

4.3.2.7. L'opérateur d'élitisme	80
4.3.2.8. Test d'arrêt	81
4.3.3. Paramètres de l'algorithme génétique	81
4.3.4. Exemple de résolution à base d'algorithme génétique	81
4.4. Optimisation par Essaim de particules (PSO)	85
4.4.1. Origine et inspiration du PSO.....	86
4.4.2. Topologies de voisinage	86
4.4.3. Principe de fonctionnement de l'algorithme du PSO	87
4.4.3.1. Formalisation.....	88
4.4.3.2. Choix des paramètres de l'algorithme	91
4.5. Conclusion.....	92

CHAPITRE 5

COMMANDES NON LINÉAIRES ROBUSTES OPTIMALES APPLIQUÉES AU ROBOT MANIPULATEUR

5.1. Introduction	94
5.2. Synthèse de la loi de commande par mode de glissement	94
5.2.1. Calcul du degré relatif	95
5.2.2. Choix de la surface de glissement	96
5.2.3. Conception de la loi de commande	96
5.2.3.1. La loi de commande discontinue	97
5.2.3.2. La loi de commande équivalente	97
5.2.3.3. La loi de commande globale.....	97
5.3. Synthèse de la loi de commande par backstepping	98
5.4. Résultats de simulation et interprétation	101
5.4.1. Simulations de la loi de commande par mode glissant.....	101
5.4.2. Simulations de la loi de commande par Backstepping	108
5.5. Optimisation des paramètres des lois de commande.....	113
5.5.1. Optimisation par la méthode des AGs de la commande par modes glissants	113
5.5.2. Optimisation par la méthode PSO de la commande par modes glissants.....	116
5.5.3. Optimisation par la méthode des AGs de la commande Backstepping	124
5.5.4. Optimisation par la méthode PSO de la commande Backstepping	125
5.6. Conclusion.....	132

CHAPITRE 6

COMMANDE NON LINÉAIRE D'UN ROBOT MANIPULATEUR ACTIONNÉ PAR DES MSAP

6.1. Introduction	133
6.2. Description du MSAP	133
6.2.1. Moteur synchrone à aimants permanents	133
6.2.2. Modélisation du MSAP	134
6.2.2.1. Equations électriques.....	134
6.2.2.2. Equations magnétiques.....	134
6.2.2.3. Equations mécaniques	136
6.2.2.4. Transformation de PARK.....	136
6.2.2.5. Mise sous forme d'équation d'état	138
6.2.2.6. Autopilotage	138
6.2.2.7. Principe de la commande vectorielle.....	139
6.3. Modèle du système global.....	140
6.4. Application de la commande non linéaire par modes glissants au modèle global	141
6.4.1. La loi de commande discontinue.....	142
6.4.2. La loi de commande équivalente.....	142
6.4.3. Résultats de simulation.....	143
6.5. La commande par modes glissants d'ordre supérieur	148
6.5.1. Concepts de base des MGOS	149
6.5.2. Commande par mode glissant d'ordre deux.....	149
6.5.2.1. Algorithme du Twisting	151
6.5.2.2. Algorithme du Super-Twisting.....	151
6.5.2.3. Algorithme prescribed convergence law (PCL)	152
6.5.2. Commande par mode glissant d'ordre supérieur.....	153
6.6. Application de la commande par l'algorithme du Super-Twisting au modèle global.....	155
6.6.1. Résultats de simulation.....	156
6.7. Conclusion.....	161
CONCLUSION GÉNÉRALE ET PERSPECTIVES	162
ANNEXE A	164
ANNEXE B	168
RÉFÉRENCES BIBLIOGRAPHIQUES.....	172

Liste des figures

Figure 1.1. Structures des chaînes cinématiques	6
Figure 1.2. Articulations rotoïde et prismatique.....	7
Figure 1.3. Classification des capteurs utilisés en robotique.....	9
Figure 1.4. Coordonnées homogènes d'un point.....	10
Figure 1.5. Transformation entre deux repères	12
Figure 1.6. Transformation d'un vecteur.....	12
Figure 1.7. Translation pure entre deux repères	13
Figure 1.8. Rotation pure entre deux repères	14
Figure 1.9. Système mécanique à n+1 corps	15
Figure 1.10. Paramètres géométriques de Denavit-Hartenberg.....	15
Figure 1.11. Paramètres géométriques pour une structure ouverte simple.....	16
Figure 1.12. Bras manipulateur à 2ddl	24
Figure 2.1. Schéma fonctionnel de la commande PID	30
Figure 2.2. Schéma fonctionnel de la commande PD avec compensation de l'effet de gravité.....	32
Figure 2.3. Position, vitesse et accélération pour une interpolation polynômiale linéaire	36
Figure 2.4. Position, vitesse et accélération pour une interpolation polynômiale de degré 3.....	37
Figure 2.5. Position, vitesse et accélération pour une interpolation polynômiale de degré 5.....	38
Figure 2.6. Position, vitesse et accélération pour une interpolation Bang-Bang.....	39
Figure 2.7. Schéma fonctionnel de la commande du couple calculé.....	40
Figure 2.8 : Robot manipulateur à deux degrés de liberté.....	42
Figure 2.9. Réponses avec la commande PID	43
Figure 2.10. Réponses avec la commande point à point.	44
Figure 2.11. Réponses avec la commande du couple calculé.....	46
Figure 3.1. Différents comportements en dehors de la surface de discontinuité.....	49
Figure 3.2. Mode de glissement idéal.....	50
Figure 3.3. Différents modes des trajectoires d'état dans un plan de phase.....	51
Figure 3.4. Comportement de la trajectoire de l'état par rapport à la surface de commutation	55
Figure 3.5. Grandeur de commande équivalente ueq	56
Figure 3.6. Forme de la commande discontinue avec la fonction $sign$	57
Figure 3.7. Phénomène de réticence.....	58
Figure 3.8. Fonction saturation.....	58
Figure 3.9. Fonction tangente hyperbolique.....	59
Figure 4.1. Minimum local vs minimum global.....	67
Figure 4.2. Le génotype.....	72
Figure 4.3. Les niveaux d'organisation d'un Algorithme Génétique.....	73
Figure 4.4. Schéma fonctionnel d'un Algorithme Génétique	74
Figure 4.5. Codage binaire	75
Figure 4.6. Codage réel	76
Figure 4.7. Sélection par roulette	78
Figure 4.8. Sélection par tournoi	78
Figure 4.9. Croisement en un seul point.....	79
Figure 4.10. Croisement en deux points.....	80
Figure 4.11. Une mutation dans un chromosome.....	80

Figure 4.12. Représentation de la fonction $f(x)$	82
Figure 4.12. Le croisement.....	83
Figure 4.13. La mutation	83
Figure 4.14. Le croisement.....	84
Figure 4.15. La mutation	84
Figure 4.16. Exemple d'essaim de poissons.....	85
Figure 4.17. Différents types de topologie de voisinage	87
Figure 4.18. Principe de déplacement d'une particule	89
Figure 4.19. Organigramme d'un algorithme PSO.....	91
Figure 5.1. Commande par Modes glissants en utilisant la fonction sign	103
Figure 5.2. Commande par Modes glissants en utilisant la fonction tanh	104
Figure 5.3. Commande par Modes glissants en poursuite de trajectoire avec conditions initiales [0.2 , 0.1].....	105
Figure 5.4. Commande par Modes glissants en poursuite de trajectoire avec conditions initiales [0.2 , 0.1], une perturbation et variation paramétrique	107
Figure 5.5. Commande par Backstepping.	109
Figure 5.6. Commande par Backstepping en poursuite de trajectoire avec conditions initiales [0.2 , 0.1]	110
Figure 5.7. Commande par Backstepping en poursuite de trajectoire avec conditions initiales [0.2 , 0.1] une perturbation et variation paramétrique.....	112
Figure 5.8. Schéma de principe de l'optimisation des régulateurs.....	113
Figure 5.9. Structure du chromosome	114
Figure 5.10. Principe d'optimisation par les algorithmes génétiques.....	115
Figure 5.11. Organigramme de l'algorithme du PSO.....	117
Figure 5.12. Principe d'optimisation par les algorithmes génétiques.....	117
Figure 5.13. Commande par Modes glissants optimisée par AG et PSO	120
Figure 5.14. Commande par Modes glissants optimisée par AG et PSO en poursuite de trajectoire avec conditions initiales [0.2 , 0.1]	123
Figure 5.15. La structure du chromosome.....	124
Figure 5.16. Principe d'optimisation de la commande Backstepping par les AGs	125
Figure 5.17. Principe d'optimisation de la commande Backstepping par PSO.....	126
Figure 5.18. Commande par Backstepping optimisée par AG et PSO	128
Figure 5.19. Commande par Backstepping optimisée par AG et PSO en poursuite de trajectoire avec conditions initiales [0.2 , 0.1].....	131
Figure 6.1. Schéma équivalent de la MS dans le référentiel de Park	136
Figure 6.2. Schéma block de MSAP	138
Figure 6.3. Schéma de principe d'autopilotage	139
Figure 6.4. Modèle de la MSAP après découplage	140
Figure 6.5. Schéma synoptique du modèle global (Robot manipulateur + MSAP)	140
Figure 6.6. Commande par Modes glissants du modèle global avec consigne constante	145
Figure 6.7. Commande par Modes glissants en poursuite de trajectoire avec conditions initiales [0.2 , 0.1].....	146
Figure 6.8. Commande par Modes glissants en poursuite de trajectoire avec variation paramétrique et perturbation externe, les conditions initiales sont [0.2 , 0.1].....	147
Figure 6.9. Un espace d'état de dimension 3 : l'intersection des 3 surfaces de commutation définit le point de référence ξ_{ref}	149
Figure 6.10. Trajectoire du mode glissant d'ordre 2	150
Figure 6.11. Convergence de l'algorithme Twisting dans le plan (s, \dot{s}).....	151

Figure 6.12. Convergence de l'algorithme PCL dans le plan (s, \dot{s})	153
Figure 6.13. Schéma de principe de la commande par Super-Twisting	156
Figure 6.14. Commande par l'algorithme du Super-Twisting du modèle global avec consigne constante	158
Figure 6.15. Commande par l'algorithme du Super-Twisting en poursuite de trajectoire avec conditions initiales $[0.2, 0.1]$	159
Figure 6.16. Commande par l'algorithme du Super-Twisting en poursuite de trajectoire avec variation paramétrique et perturbation externe, les conditions initiales sont $[0.2, 0.1]$	160
Figure A.1. Notion de stabilité	166
Figure A.2. Stabilité asymptotique.....	166
Figure B.1. Prototype réel du 1er modèle.....	169
Figure B.2. Robot manipulateur avec dimensions.....	170
Figure B.1. Prototype réel du 2 ^{ème} modèle	171

Liste des tableaux

Tableau 4.1. Choix aléatoire des individus	82
Tableau 4.2. Résultats obtenus d'une sélection par tournoi	82
Tableau 4.3. Individus de la première génération	83
Tableau 4.4. Re-sélection par tournoi.....	84
Tableau 4.5. Sélection par tournoi.....	85
Tableau 5.1. Les paramètres de l'algorithme génétique.....	116
Tableau 5.2. Les paramètres obtenus	116
Tableau 5.3. Les paramètres de l'algorithme PSO.....	118
Tableau 5.4. Les paramètres obtenus par PSO	118
Tableau 5.5. Tableau de comparaison entre les techniques d'optimisation appliquées au MG	123
Tableau 5.6. Les paramètres de l'algorithme génétique.....	125
Tableau 5.7. Les paramètres obtenus par AG.....	125
Tableau 5.8. Les paramètres de l'algorithme PSO.....	126
Tableau 5.9. Les paramètres obtenus par PSO	126
Tableau 5.10. Tableau de comparaison entre les techniques d'optimisation appliquées au Backstepping	131

Notations

q_i	Position angulaire du segment i .
\dot{q}_i	Vitesse angulaire du segment i .
\ddot{q}_i	Accélération angulaire du segment i .
τ_i	Couple moteur de l'articulation i .
m_i	Masse du segment i (kg).
l_i	Longueur du segment i (m).
l_{ci}	Longueur du centre de gravité (m).
G	Accélération de gravité du segment i (m/s ²).
I_i	Inertie du segment i (kg.m ²).
${}^{i-1}T_i$	La matrice de transformation homogène.
$J(q)$	Matrice jacobienne
I	Matrice identité.
L	Lagrangien du mécanisme.
E_{cin}	Energie cinétique totale du mécanisme.
E_{pot}	Energie potentielle totale du mécanisme.
$M(q)$	Matrice d'inertie.
$C(q, \dot{q})$	Matrice de Coriolis et Centrifuge.
$g(q)$	Vecteur de gravité.
t_{fin}	Temps final.
q_i^{init}	Position angulaire initiale de l'articulation i .
q_i^{fin}	Position angulaire finale de l'articulation i .
$S(\xi)$	Surface de glissement.
$sign(S)$	Fonction <i>sign</i> .
$sat(S)$	Fonction saturation.
$tanh(S)$	Fonction tangente hyperbolique.
p_m	Probabilité de l'opérateur de mutation.
P_c	Probabilité de l'opérateur de croisement.
P_{best}	Meilleure position trouvée par la particule
G_{best}	Meilleure position trouvée par l'essaim

Abréviations

Ddl	Degrés de liberté.
D-H	Denavit-Hartenberg.
MGD	Modèle géométrique direct
MGI	Modèle géométrique inverse
MCD	Modèle cinématique direct
MCI	Modèle cinématique inverse
PID	Proportionnel intégral dérivée.
PD	Proportionnelle dérivée
MG	Mode de glissement
SMC	Commande par modes glissants (Sliding Mode Control)
AG	Algorithme génétique
AGs	Algorithmes génétiques
PSO	Optimisation par Essaim de particules (Particle Swarm Optimization)
MSE	Mean of the Square of the Error
MAE	Mean of Absolute Magnitude of the Error
MSAP	Moteur Synchrone à Aimant Permanent

INTRODUCTION GÉNÉRALE

L'objectif du contrôle est de concevoir un système automatique pouvant fonctionner dans un environnement réel. En présence des bruits et des perturbations cet environnement peut être variable.

La plupart des systèmes réels sont généralement de nature non linéaire. Le recours à une approximation linéaire autour d'un point d'équilibre sur de tel processus peut s'avérer insuffisant ou même complètement inadapté. Vers la fin des années soixante les ingénieurs de l'époque ont ressenti la nécessité de considérer des modèles non linéaires puisqu'ils savaient que la plupart des processus ne présentaient pas un comportement linéaire, mais ils ne disposaient que des méthodes valables pour le cas linéaire[1] [2].

Les lois de commande sont généralement conçues sur la base d'un modèle obtenu par modélisation ou par identification et après usage de beaucoup de simplifications, tel que la négligence de certaines dynamiques. En conséquence, les lois de commandes conçues sur la base des modèles simplifiées peuvent ne pas fonctionner sur les systèmes réels dans un environnement réel, où les systèmes sont dans la plupart du temps non linéaires, mal définis, à paramètres variables et soumis à des perturbations externes. Utiliser une approche d'analyse dans le domaine temporel et fréquentiel n'est plus possible puisqu'il n'existe généralement pas de solution directe aux équations représentant ces types de systèmes. D'où le recours au développement des méthodes d'analyse qui traitent directement les non linéarités[3] [4].

Dans une situation pareille on cherche à concevoir une approche de commande qui s'adapte à cet environnement, en assurant le bon fonctionnement du système pour toutes les situations rencontrées, on parle alors de commande robuste.

Depuis leurs apparitions au début des années soixante, où leurs interventions étaient limités à accomplir des tâches répétitives et dangereuses pour l'être humain, les robots industriels n'ont pas cessé de progresser et de gagner de l'espace dans le secteur industriel, grâce aux développements de l'électronique, l'automatique, l'informatique et de la mécanique.

Certains secteurs industriels ont largement bénéficié du développement de la robotique, plus particulièrement l'industrie automobile, où les robots sont devenus un outil indispensable tout en augmentant de manière importante le rendement et conditionnant ainsi leur survie économique.

Plus récemment, la robotique connaît une grande évolution dans plusieurs secteurs afin de pouvoir répondre aux attentes croissantes des entreprises et des utilisateurs. Les robots sont aujourd'hui dopés par l'intelligence artificielle. Cobotique, robotique de service, machine Learning, maintenance prédictive et surtout avec l'essor de l'industrie 4.0 qui fait la part belle à de nouvelles technologies appliquées dans le domaine de la robotique, lesquelles offrent une toute nouvelle efficacité à l'industrie et aux collaborateurs, en même temps qu'elle pousse au développement de nouvelles compétences.

Une des principales caractéristiques des robots modernes est la capacité d'exploiter l'intelligence artificielle, afin d'améliorer leurs performances en continu. Une entreprise américaine de conseil et de recherche dans le domaine des techniques avancées nommée

Gartner a mené une étude dans 89 pays du monde entier auprès de 3000 industriels, entre 2015 et 2018, selon cette étude l'introduction de l'intelligence artificielle dans les entreprises a connu une augmentation de 270%.

Résoudre un problème d'optimisation consiste à trouver la ou les meilleures solutions vérifiant un ensemble de contraintes et d'objectifs définis par l'utilisateur.

Pour déterminer si une solution est meilleure qu'une autre, les métas heuristiques offrent un outil d'analyse. Parmi ces outils, nous citons la famille des algorithmes évolutionnaires, qui ont connu un grand succès dans l'optimisation multi objectifs.

Les algorithmes génétiques et le PSO sont des algorithmes d'optimisation stochastique dont le but est d'obtenir une solution convenable dans un temps acceptable. Dans ce contexte, le chercheur s'intéressera à l'utilisation de ce type d'algorithmes d'optimisation pour la détermination d'une structure optimale des contrôleurs, afin d'avoir des performances meilleures.

Contexte du travail et motivations

Les robots manipulateurs sont des systèmes multi-entrée multi-sortie ayant une dynamique couplée non linéaire et des variations paramétriques. Généralement, dans les applications industrielles, de tels systèmes complexes sont commandés à l'aide des contrôleurs traditionnels de type PID par exemple, qui sont faciles à mettre en œuvre, ne nécessitant la détermination que de trois paramètres et des performances de suivi relativement acceptables du robot [5] [6] [7]. Mais l'inconvénient majeur du contrôle PID est de ne pas atteindre les performances de suivi souhaitées du bras du robot lorsque la dynamique du système non linéaire domine la dynamique linéaire. Par exemple, dans les systèmes robotiques, utiliser pour des opérations d'assemblage avec des pièces lourdes portées par la pince du manipulateur, un robot à fonctionnement rapide, etc. plusieurs travaux de recherche ont été menés afin d'éviter les lacunes des contrôleurs classiques en utilisant un contrôleur non linéaire et différentes approches ont été proposées pour concevoir le contrôle non linéaire, tel que la commande par modes glissants et la commande par backstepping. Cependant, ces techniques de commande qualifiées de robustes, nécessite la détermination de certains paramètres afin d'avoir un contrôle optimal.

Actuellement, les problèmes d'optimisation occupent un espace important au sein de la communauté des scientifiques. En effet, la résolution de ce genre de problème intervient dans divers domaines d'activité telle que le traitement d'images, la conception mécanique, l'électronique, la recherche opérationnelle, la planification de trajectoire ou la commande des systèmes.

Le problème d'optimisation peut être défini par la spécification d'un ensemble de variables, d'une fonction objectif, d'un ensemble de contraintes et d'un espace de recherche qui représente l'ensemble des solutions possibles au problème. La résolution du problème d'optimisation revient à chercher la ou les meilleures solutions possibles par minimisation et/ou maximisation de la fonction objectif, tout en tenant compte des contraintes. Il existe des problèmes d'optimisations qui sont qualifiés de difficiles, et leur résolution en un temps raisonnable fait appel à l'utilisation d'algorithmes plus sophistiqués, tel que les méthodes métaheuristiques. Parmi les méthodes basées sur les métaheuristiques qui sont destinées à résoudre ce genre de

problèmes on trouve l'optimisation par les Algorithmes génétiques (AGs) [8] et l'optimisation par essaim particulaire (PSO) [9].

Objectifs et contribution

Sachant que les commandes classiques telles que, le PID peuvent s'avérer suffisantes si les exigences recherchées en termes de précision et performances ne sont pas trop élevés. Dans le cas contraire, il est recommandé de concevoir des techniques de commande assurant une certaine robustesse du comportement du processus envers les incertitudes paramétriques et leurs variations. Donc, l'objectif recherché en premier lieu à travers ce travail consiste à concevoir des lois de commandes non linéaire et robuste.

La mise en œuvre de ces différentes lois de commande, nécessite la détermination des différents paramètres de commande. Chose qui n'est pas évidente puisqu'il n'existe pas de méthode directe pour déterminer ces paramètres. Notre première contribution est d'introduire deux méthodes d'optimisation métaheuristiques à savoir les Algorithmes génétiques et le PSO afin de trouver les paramètres optimaux des contrôleurs non linéaires.

En deuxième lieu nous allons prendre en considération l'introduction de la dynamique de l'actionneur sur le modèle global du robot manipulateur. La majorité des travaux de recherche sur la commande des robots manipulateurs avec introduction de la dynamique de l'actionneur (électrique) utilisent un moteur à courant continu. Ce dernier souffre de la présence du balais-collecteur, nécessitant ainsi un entretien permanent avec un coût plus élevé et le risque de l'utiliser dans un milieu explosif. Notre deuxième contribution est d'utiliser comme actionneur un moteur synchrone à aimant permanent (MSAP), ainsi que la conception des lois de commande non linéaire robuste pour le système global (robot +actionneur MSAP).

A la fin des résultats sur la stabilité, les performances et la robustesse des différentes lois de commande sont présentés.

Organisation de la thèse

Cette thèse est organisée en six chapitres.

Le premier chapitre sera axé sur la présentation des généralités sur les systèmes mécaniques articulés, avec une description de la modélisation des robots manipulateurs à chaîne ouverte simple. Un exemple de calcul d'un modèle dynamique pour un robot manipulateur à 2ddl est exposé à la fin.

Le deuxième chapitre présente d'abord quelques lois de commande classique existant dans la littérature, appliquées au robot manipulateur, nous exposons ensuite quelques techniques de génération de mouvement des robots manipulateurs par interpolation. Quelques lois de commande classique seront appliquées sur un bras manipulateur.

Une présentation sur les aspects théoriques des lois de commande non linéaire et robuste : le mode glissant et le backstepping, est donnée dans le troisième chapitre.

Dans le quatrième chapitre, nous nous concentrons d'abord sur l'introduction du concept d'optimisation, ainsi que les différentes classes de méthodes d'optimisation. Deux techniques d'optimisation qui vont être utilisées par la suite ont été détaillées en deuxième partie de ce chapitre. La première méthode inspirée des phénomènes naturels est celle des Algorithmes génétiques, la deuxième méthode inspirée en analogie avec le comportement des animaux

lorsqu'ils se déplacent en collectif est la méthode d'optimisation par l'essaim de particules (PSO).

Le cinquième chapitre sera consacré à l'application des différentes lois de commande traitées dans le troisième chapitre. Ces lois de commande seront ensuite optimisées en utilisant les techniques d'optimisation traitées dans le chapitre quatre. Plusieurs résultats de simulation obtenus seront présentés.

Dans le sixième chapitre, la modélisation du moteur synchrone à aimants permanents sera présentée, ensuite des commandes non linéaires robustes appliquées sur le modèle global du robot manipulateur en tenant compte de la dynamique de l'actionneur seront développées. Les résultats de simulations vont être présentés par la suite.

Enfin, nous achèverons ce travail par une conclusion générale faisant la synthèse des principaux résultats obtenus et donnant quelques perspectives.

CHAPITRE 1

ELÉMENTS DES SYSTÈMES MÉCANIQUES ARTICULÉS

1.1. Introduction

Les premiers concepts de la robotique datent de plusieurs centaines d'années, mais l'apparition du mot robot pour la première fois fut en 1920 dans une pièce théâtrale (RUR « Rossum's Universal Robots »), écrite par le tchèque Karel Čapek. Le verbe tchèque « robota » qui signifie travail pénible ou forcé est à l'origine du mot robot[10].

Il a fallu patienter jusqu'au début des années soixante pour que la robotique soit plus une réalité qu'un thème de science-fiction. La société américaine General Motors sera le premier fabricant à avoir installé un robot dans une chaîne de production en 1961, Ce bras manipulateur est nommé « Unimate »[11].

Depuis, L'évolution de la robotique a connu trois générations grâce au développement technologique, la première génération est définie par les robots industriels classiques avec des mouvements répétitifs appris ou programmés, sans avoir d'informations sur son environnement. Lorsque les robots ont une capacité à acquérir certaines informations sur leurs environnement, ce qui leur permet de localiser les pièces, nous parlerons dans ce cas de la deuxième génération. Actuellement, dans la troisième génération on parle de robots évolués ayant des capacités de vision, toucher, entendre et une certaine autonomie dans la prise de décision.

La connaissance d'un modèle qui décrit le comportement dynamique du robot est primordial, afin de pouvoir appliquer les différentes techniques de commande, plus précisément la représentation par un modèle mathématique ainsi que ses paramètres.

1.2. Terminologie générale

Un robot manipulateur est constitué d'une succession de corps rigides en chaîne cinématique qui peut être ouverte, arborescente ou composée. Les liaisons entre les différents corps de la chaîne sont appelées **articulations**.

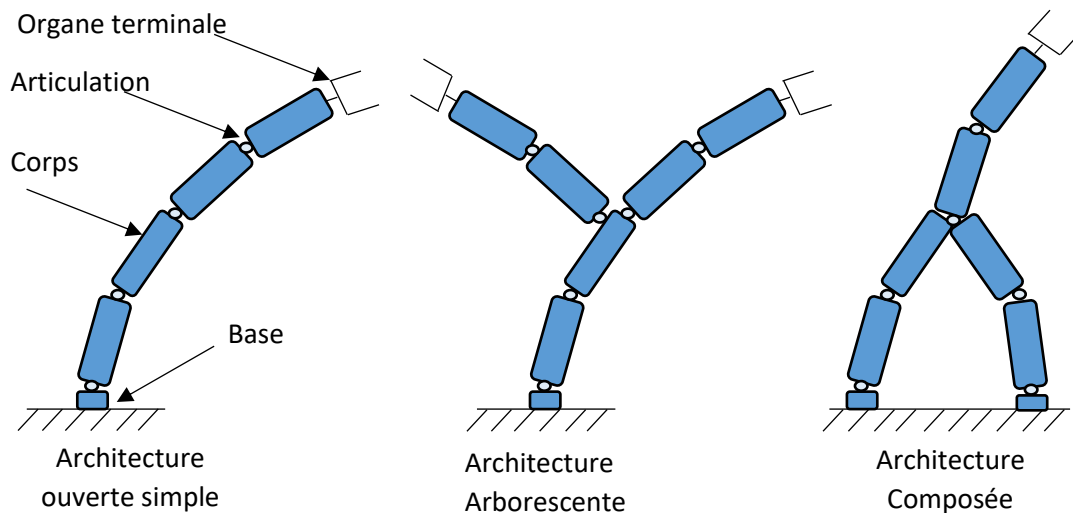


Figure 1.1. Structures des chaînes cinématiques

Cette structure mécanique articulée a pour rôle d'amener l'organe terminal vers une position et orientation désirées, en respectant certaines caractéristiques de vitesse et d'accélération.

Un organe terminal est l'élément qui permet au robot d'interagir avec son environnement (poignet, préhenseur, effecteur, ...).

1.2.1. Articulations

La liaison mécanique entre deux segments successifs limitant le nombre de degrés de liberté (d.d.l) de l'un par rapport à l'autre est dit articulation [6]. Le nombre de degré de liberté est noté m , appelé aussi mobilité de l'articulation. Cette mobilité remplit la condition suivante $0 \leq m \leq 6$ [6].

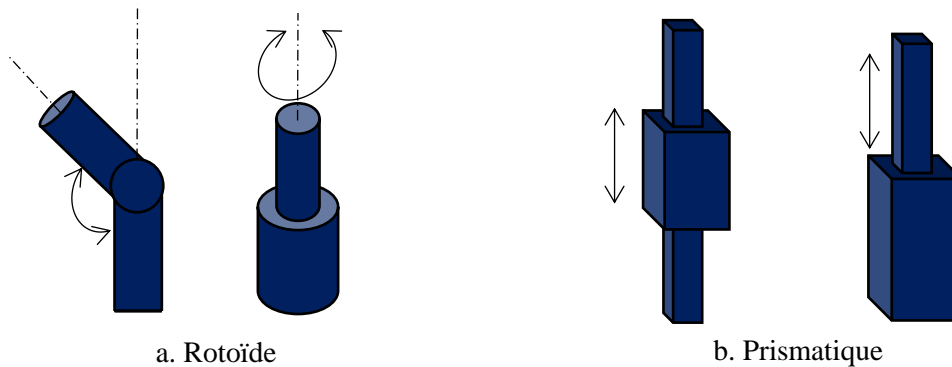
En robotique dans la plupart des cas $m=1$, si c'est le cas alors cette articulation elle est soit de type *rotoïde* et notée R, soit de type *prismatique* et notée P.

1.2.1.1. Articulation de type rotoïde

Ce type d'articulation réduit le mouvement relatif entre deux corps qui se succèdent à une rotation autour d'un axe commun, c'est une articulation de type pivot tel qu'il est montré sur la figure 1.2a.

1.2.1.2. Articulation prismatique

C'est une articulation de type glissière, le mouvement est réduit entre deux corps qui se succèdent à une translation le long d'un axe commun, comme le montre la figure 1.2b.



a. Rotoïde

b. Prismatique

Figure 1.2. Articulations rotoïde et prismatique.

Des liaisons avec un degré de liberté supérieure à 1 peuvent être constituées en combinant plusieurs articulations simples. Comme pour l'exemple de la rotule, elle est obtenue en utilisant trois articulations rotoïdes d'axes concourants [6].

1.2.2. Organe terminal

C'est la désignation des dispositifs fixé sur l'extrémité du dernier corps du robot. Il représente l'interface du robot avec son environnement. Son rôle est de manipuler ou de transformer des pièces ou des objets, selon sa nature. Ça peut être une mèche, une pince, un pistolet de peinture, etc.

1.2.3. Variable articulaire

C'est la désignation pour chaque articulation, de la position angulaire pour l'articulation rotoïde, ou la position linéaire pour l'articulation prismatique du corps correspondant dans un repère attaché à ce dernier [12].

1.2.4. Espace articulaire

Appelé aussi espace de configuration est celui dans lequel est représentée la situation de tous les corps du robot en utilisant les variables articulaires. Sa dimension notée N est égal au nombre de degré de liberté.

1.2.5. Espace opérationnel

L'espace opérationnel est utilisé pour représenter la situation de l'organe terminal. Un ensemble de deux espaces noté R^x est considéré, en utilisant pour la position les coordonnées cartésiennes dans R^3 et pour l'orientation des rotations propres de R^3 . La dimension de R^x notée M qui constitue le nombre de ddl au maximum qu'un organe terminal peut avoir, est égale au nombre de paramètres indépendants indispensables pour décrire dans l'espace la situation de l'organe terminal [6]. Cette dimension vaut six en espace tridimensionnel, tel que, trois pour une orientation quelconque d'un corps et trois pour placer un point de ce corps dans une position quelconque [6]. Et par conséquent $M \leq 6$ et aussi $M \leq N$.

1.2.6. Redondance d'un robot manipulateur

Dans le cas où le nombre d'articulations motorisées (nombre de d.d.l de l'espace articulaire) est supérieur au nombre de d.d.l de l'organe terminal, le robot est dit redondant. Cela permet

d'accroître le volume de l'espace accessible et de maintenir les capacités de déplacement de l'organe terminal en présence d'obstacles [13].

A titre d'exemple, pour les chaînes ouvertes simples, une structure redondante est obtenue par les combinaisons suivantes [6]:

- ❖ Plus de six articulations motorisées.
- ❖ Plus de trois articulations rotoïdes d'axes concourants.
- ❖ Plus de trois articulations rotoïdes d'axes parallèles.
- ❖ Plus de trois articulations prismatiques.
- ❖ Deux axes d'articulations prismatiques parallèles.
- ❖ Deux axes d'articulations rotoïdes confondus.

1.2.7. Les actionneurs

Les actionneurs sont indispensables pour assurer le mouvement des différents corps du robot manipulateur, en dépit de la gravité, l'inertie et les forces externes [14]. Ils sont placés au niveau de chaque articulation. Les actionneurs des robots manipulateurs peuvent être électriques, hydrauliques ou pneumatiques [10].

1.2.7.1. Actionneurs électriques (Les moteurs électriques)

Une grande partie des robots manipulateurs sont équipées des moteurs électriques comme actionneur. L'utilisation d'un moteur ou d'un autre dépend du domaine d'application. Le moteur à courant continu est le moteur le plus répandu en robotique par rapport aux moteurs pas à pas ou encore les moteurs asynchrone et synchrone. Ces dernières années on voit de plus en plus l'introduction des moteurs synchrones à aimants permanents [15] [16].

1.2.7.2. Actionneurs hydrauliques et pneumatiques (les Vérins)

Les actionneurs hydrauliques et pneumatiques convertissent l'énergie hydraulique ou l'énergie pneumatique respectivement en une énergie mécanique de translation principalement et de rotation. Ces types d'actionneurs sont simples à mettre en œuvre, de conception robuste, et permettent de reproduire des tâches manuelles telles que : pousser, serrer, soulever, tirer, etc. on distingue plusieurs types de vérins, les plus connus sont les vérins simples effets et les vérins doubles effets [7].

1.2.8. Les capteurs

Les capteurs sont les éléments qui servent à détecter et collecter les informations sur les états internes et en environnementaux de la structure mécanique, les robots manipulateurs sont équipés de capteurs placés à des endroits différents du robot pour mesurer différentes grandeurs pertinentes, permettant à l'utilisateur de se renseigner sur l'état du robot à chaque moment [12]. La position, la vitesse, l'accélération et la force sont les informations les plus importantes à détecter en robotique. Des capteurs, intégrés au robot, envoient des informations sur chaque liaison et joint à l'unité de contrôle, et l'unité de contrôle détermine la configuration du robot pour permettre un contrôle convenable du robot [14].

Placés sur le robot, Ces capteurs sont de deux types différents. Soit ils informent le robot sur son état interne, alors on parle de capteurs *proprioceptifs*, Soit ils fournissent au robot des informations sur la nature de l'environnement, alors ce seront des capteurs *extéroceptifs*.

1.2.8.1. Les capteurs proprioceptifs

Les capteurs proprioceptifs permettent de localiser l'état mécanique interne du robot. La position, la vitesse et l'accélération font partie des grandeurs mesurées par ce type de capteur [17].

Selon la nature de l'information fournie par le capteur, plusieurs types de capteurs proprioceptifs existent [17]:

- Ceux qui fournissent une information relative à la position entre chaque segment

(Codeur incrémental, potentiomètres rotatifs mono –tour...);

- D'autres qui signalent les positions extrêmes des articulations, appelé butées fin de course.

1.2.8.2. Les capteurs extéroceptifs

Ces capteurs recueillent des informations relatives à l'environnement (tel que la mesure de distance, la détection de présence...etc.), il faut avoir une connaissance exacte de leurs caractéristiques pour pouvoir interpréter les informations reçues et les décoder. En plus il faut connaître leurs plages d'utilisation afin de pouvoir juger si l'information reçue est valable ou non.

Il existe plusieurs types de capteurs extéroceptifs utilisés en robotique [17]:

- Des capteurs donnant l'inclinaison par rapport à la verticale (inclinomètre)
- Des capteurs informant sur la présence ou non d'obstacles (Capteur photoélectriques, à ultrasons);
- D'autres qui indiquent la vitesse de rotation (gyroscopes).

D'une manière générale en robotique, une classification des capteurs peut être Présentée par la figure 1.3.

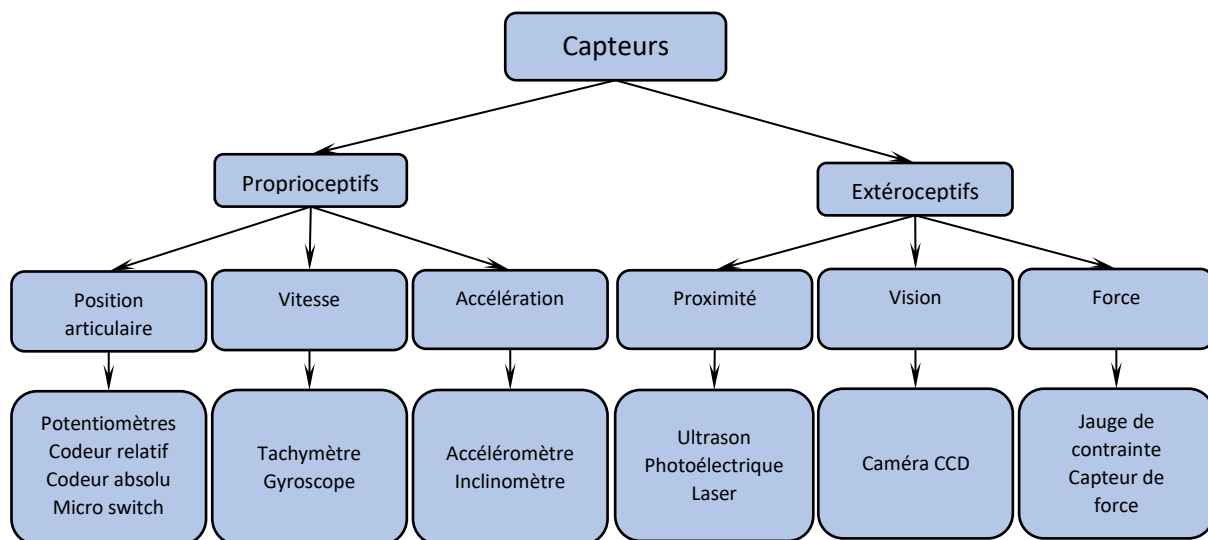


Figure 1.3. Classification des capteurs utilisés en robotique

1.3. Modélisation des robots manipulateurs

Selon un objectif fixé préalablement, de conception, de simulation ou de commande d'un système mécanique articulé, il est nécessaire de disposer de modèles du mécanisme à traiter. Ce sont des modèles de transformation entre l'espace articulaire et l'espace opérationnel (dans lequel où est définie la position de l'organe terminal) [18]. Suivant un cahier des charges spécifiques à une application envisagée, différents niveaux de modélisation existent : modèles *géométriques*, modèles *cinématiques* et modèles *dynamiques* à partir desquels peuvent être engendrés les mouvements du robot, ou bien des modèles statiques qui décrivent les interactions du mécanisme avec son environnement [11]. L'obtention de ces différents modèles n'est pas une tâche aisée, dont la difficulté peut varier suivant plusieurs facteurs tels que la structure de la chaîne cinématique qui peut être ouverte simple, fermée ou arborescente, la nature des articulations, le nombre de d.d.l et aussi la complexité de la configuration de la chaîne articulée [11].

1.3.1. Les coordonnées homogènes

En robotique, tous les éléments du poste de travail sont associés à un ou plusieurs repères pour représenter les positions et les orientations des différents corps, ainsi que les différentes transformations entre ces systèmes de coordonnées. En général, ces repères sont définis d'une manière à ce que leurs origines coïncident avec des points privilégiés et leurs axes correspondent à des directions ayant un rôle utile lorsque le robot exécute une tâche [6]. En effet, la géométrie dans l'espace tridimensionnel et les mouvements rigides jouent un rôle important dans tous les aspects des robots manipulateurs [19]. Nous allons donc commencer à examiner les représentations de points et de vecteurs dans un espace euclidien, afin de pouvoir introduire les différentes transformations.

1.3.1.1. Représentation d'un point

Tout point M de l'espace peut être représenté par des coordonnées cartésiennes (M_x , M_y , M_z), les termes $w.M_x$, $w.M_y$, $w.M_z$ et w sont appelés coordonnées homogènes du point P, où w qui vaut 1 en robotique est un facteur d'échelle. Alors le vecteur suivant représente les coordonnées homogènes d'un point :

$$M = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (1.1)$$

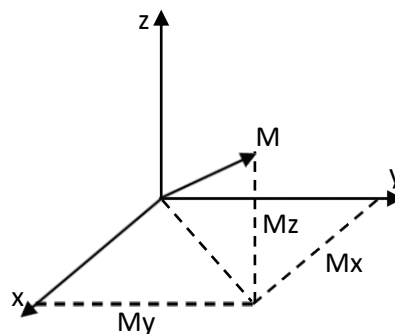


Figure 1.4. Coordonnées homogènes d'un point

1.3.1.2. Représentation d'un vecteur libre

La représentation d'un vecteur dans l'espace se fait par quatre composantes, les trois premières pour les coordonnées cartésiennes et la dernière vaut zéro [6]. En notant U_x , U_y , U_z les coordonnées cartésiennes d'un vecteur unitaire U , en coordonnées homogènes on écrit :

$$U = \begin{bmatrix} U_x \\ U_y \\ U_z \\ 0 \end{bmatrix} \quad (1.2)$$

1.3.1.3. Représentation d'un plan

Le plan $\alpha x + \beta y + \gamma z + \delta = 0$, est représenté par un vecteur Q , tel que :

$$Q = [\alpha \quad \beta \quad \gamma \quad \delta] \quad (1.3)$$

Pour tout point M appartenant au plan Q , le produit matriciel $Q M$ est nul.

$$Q M = [\alpha \quad \beta \quad \gamma \quad \delta] \begin{bmatrix} M_x \\ M_y \\ M_z \\ 1 \end{bmatrix} = 0 \quad (1.4)$$

1.3.2. Transformation homogène

Pour plusieurs raisons, la transformation de repère est une notion fondamentale dans la modélisation des robots, elle permet [6] :

- De situer les positions des différents corps du robot les uns par rapport aux autres ;
- De définir la situation que doit atteindre le repère associé à l'organe terminal, ainsi que la vitesse correspondante afin de réaliser une tâche donnée ;
- De spécifier et de commander les efforts lors de l'interaction du robot avec son environnement ;
- D'intégrer les informations issues des capteurs à la commande.

1.3.2.1. Transformation d'un repère à un autre

Soit la matrice jT_k dite matrice de transformation homogène de dimension (4×4) . Cette matrice définit toute transformation de rotation et/ou translation d'un repère R_j l'amenant sur le repère R_k , comme le montre la figure 1.5, telle que [20] :

$${}^jT_k = \begin{bmatrix} {}^j s_k & {}^j n_k & {}^j a_k & {}^j m_k \\ s_x & n_x & a_x & m_x \\ s_y & n_y & a_y & m_y \\ s_z & n_z & a_z & m_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.5)$$

Avec :

- ${}^j s_k$, ${}^j n_k$ et ${}^j a_k$ représentent les vecteurs unitaires du repère R_k exprimés dans le repère R_j suivant les trois axes x , y et z respectivement.
- ${}^j m_k$ exprime l'origine du repère R_k dans le repère R_j .

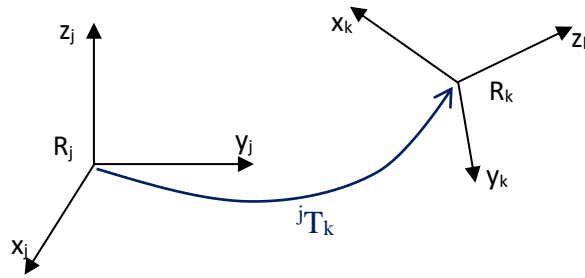


Figure 1.5. Transformation entre deux repères

La matrice jT_k peut être représentée sous la forme suivante [20]:

$${}^jT_k = \begin{bmatrix} {}^jA_k & {}^jM_k \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^jS_k & {}^jn_k & {}^ja_k & {}^jM_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.6)$$

Où jA_k est la matrice d'orientation, et jM_k est le vecteur de translation.

1.3.2.2. Transformation des vecteurs

Un point M_1 est défini par un vecteur ${}^i m_1$ dans un repère R_k , comme il est montré sur la figure 1.6. Les coordonnées du point M_1 dans le repère R_j sont calculées en utilisant la formule suivante [6] :

$${}^j p_1 = {}^j(O_i M_1) = {}^jS_k^k m_{1x} + {}^jn_k^k m_{1y} + {}^ja_k^k m_{1z} + {}^jM_k = {}^jT_k^k p_1 \quad (1.7)$$

Donc la matrice jT_k va permettre d'exprimer dans le repère R_j les coordonnées d'un point données dans le repère R_k [6].

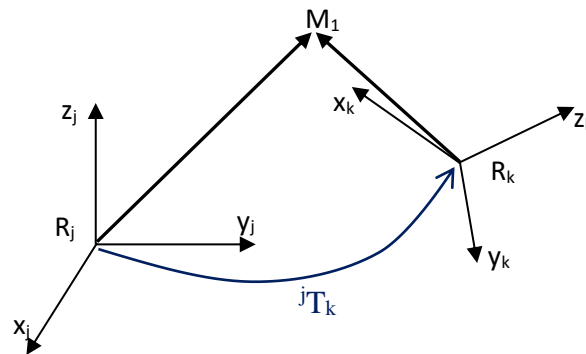


Figure 1.6. Transformation d'un vecteur

1.3.2.3. Matrice de transformation dans le cas d'une translation pure

Dans le cas où la liaison entre deux repères se résume à une translation, le passage d'un repère à un autre peut se faire à l'aide d'une matrice de transformation homogène de translation pure, notée $\text{Trans}(a,b,c)$, qui signifie une translation de valeur a selon l'axe x , b selon l'axe y et c selon l'axe z .

Les notations $\text{Trans}(x,a)$, $\text{Trans}(y,b)$ et $\text{Trans}(z,c)$ indiquent une translation d'une longueur a , b , et c selon l'axe x , y et z respectivement.

En se basant sur l'exemple montré sur la figure 1.7, la translation d'un repère R_j vers un repère R_k est représentée par la relation suivante [21]:

$${}^jT_k = \text{Trans}(x, a) \cdot \text{Trans}(y, b) \cdot \text{Trans}(z, c) = \text{Trans}(a, b, c) \quad (1.8)$$

Nous pouvons alors exprimer la matrice de transformation homogène de translation pure jT_k par :

$${}^jT_k = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.9)$$

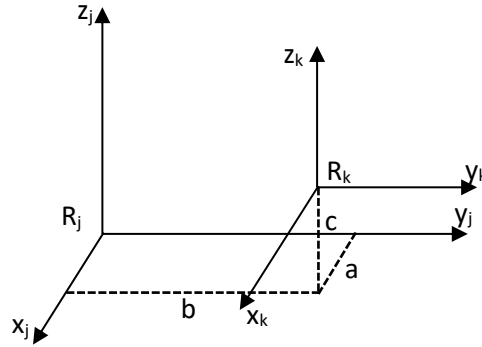


Figure 1.7. Translation pure entre deux repères

1.3.2.4. Matrice de transformation dans le cas d'une rotation pure

Lorsque la liaison entre deux repères se résume à une rotation, le passage d'un repère à un autre est possible en utilisant une matrice de transformation homogène de rotation pure, trois cas de matrice de rotation sont définis comme suit [6]:

- Matrice de rotation selon l'axe x d'un angle ϑ_x :

$${}^jT_k = \text{Rot}(x, \vartheta_x) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \vartheta_x & -\sin \vartheta_x & 0 \\ 0 & \sin \vartheta_x & \cos \vartheta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.10)$$

- Matrice de rotation selon l'axe y d'un angle ϑ_y :

$${}^jT_k = \text{Rot}(y, \vartheta_y) = \begin{bmatrix} \cos \vartheta_y & 0 & \sin \vartheta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \vartheta_y & 0 & \cos \vartheta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.11)$$

- Matrice de rotation selon l'axe z d'un angle ϑ_z :

$${}^jT_k = \text{Rot}(z, \vartheta_z) = \begin{bmatrix} \cos \vartheta_z & -\sin \vartheta_z & 0 & 0 \\ \sin \vartheta_z & \cos \vartheta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.12)$$

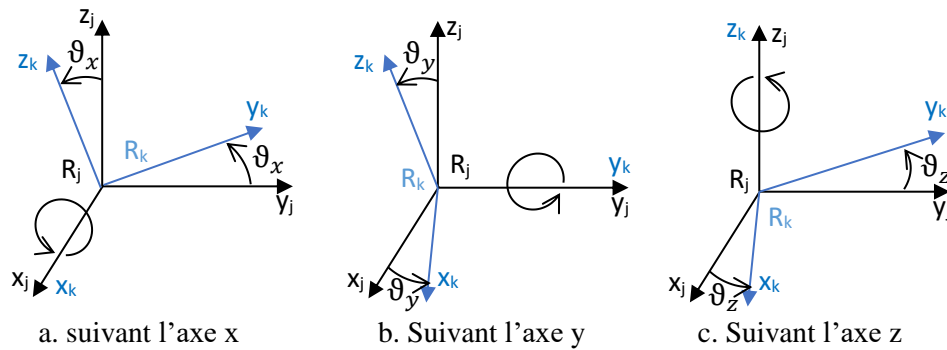


Figure 1.8. Rotation pure entre deux repères

Donc la matrice de transformation homogène T peut exprimer la position et l'orientation d'un solide dans l'espace :

$$T = \begin{bmatrix} s_x & n_x & a_x & m_x \\ s_y & n_y & a_y & m_y \\ s_z & n_z & a_z & m_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R & M \\ 0 & 1 \end{bmatrix} \quad (1.13)$$

Avec :

- R matrice (3×3) des rotations décrivant l'orientation du solide dans un repère fixe R_0 ,
- M vecteur (3×1) des translations décrivant la position du solide dans un repère fixe R_0 .

1.3.3. Modèle géométrique d'une structure ouverte simple

La modélisation des robots manipulateurs nécessite l'utilisation de certains concepts qui aident à décrire leur morphologie géométrique. La méthode la plus célèbre et la plus utilisée est celle de Denavit-Hartenberg [22], qui a été développée principalement pour des architectures ouvertes simples. Les paramètres de cette convention donnent la possibilité de calculer la position ainsi que l'orientation d'un repère R_i dans un repère R_{i-1} . Cependant lorsqu'elle est utilisée pour une architecture fermée ou arborescente, cette méthode présente des ambiguïtés. Afin de lever ces ambiguïtés, et permettre une description homogène avec un nombre réduit de paramètres pour représenter n'importe quelle architecture, Khalil et Kleinfinger [23] [20] ont proposé une méthode nommée Denavit-Hartenberg modifiée.

1.3.3.1. Convention de Denavit-Hartenberg

Supposant un système mécanique composée de $n+1$ corps $C_0, C_1, C_2 \dots C_n$ supposés rigides (voir figure 1.9), et liés entre eux par des articulations simples (n articulations) qui peuvent être soit de type rotoïde ou prismatique. Les articulations sont supposées idéales, c à d qu'il n'y a pas de jeu mécanique ni d'élasticité. Les corps C_0 et C_n désignent la base du robot et le corps chargé de porter l'organe terminal respectivement.

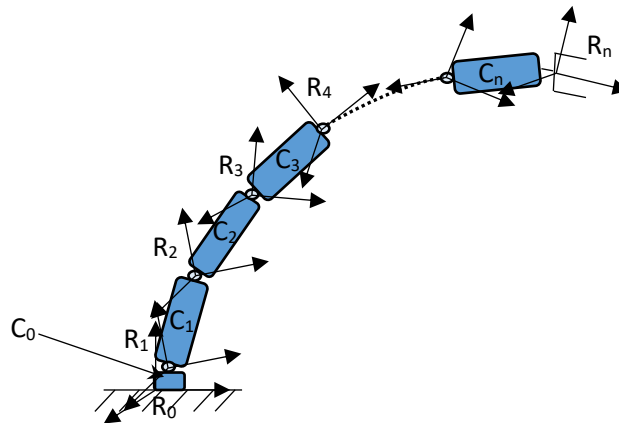


Figure 1.9. Système mécanique à n+1 corps

On associe à tout corps C_i un repère R_i d'origine O_i et d'axes x_i , y_i et z_i , de base orthonormée. Les repères sont définis de telle sorte que [22] :

- L'axe z_i est fixé sur l'axe de l'articulation $i+1$,
- Afin que x_0 et y_0 forment un repère orthonormé direct, O_0 sera choisi a un point quelconque de z_0 ,
- O_i est placé à l'intersection de l'axe z_i avec la droite perpendiculaire commune à z_{i-1} et z_i . Si les deux axes se coupent, l'origine est le point d'intersection. Si par contre, les axes sont parallèles l'origine est l'origine du repère de l'articulation $i+1$,
- L'axe x_i est la droite perpendiculaire commune à z_{i-1} et z_i , et l'axe y_i est choisi de sorte que le repère soit orthonormé direct.

La représentation du repère R_i dans le repère R_{i-1} est exprimée par les paramètres suivants :

- θ_i correspond à une rotation autour de l'axe z_{i-1} .
- d_i correspond à une translation le long de l'axe z_{i-1} .
- r_i correspond à une translation le long de l'axe x_i .
- α_i correspond à une rotation autour de l'axe x_i .

La figure 1.10 présente une configuration géométrique des paramètres de Denavit-Hartenberg.

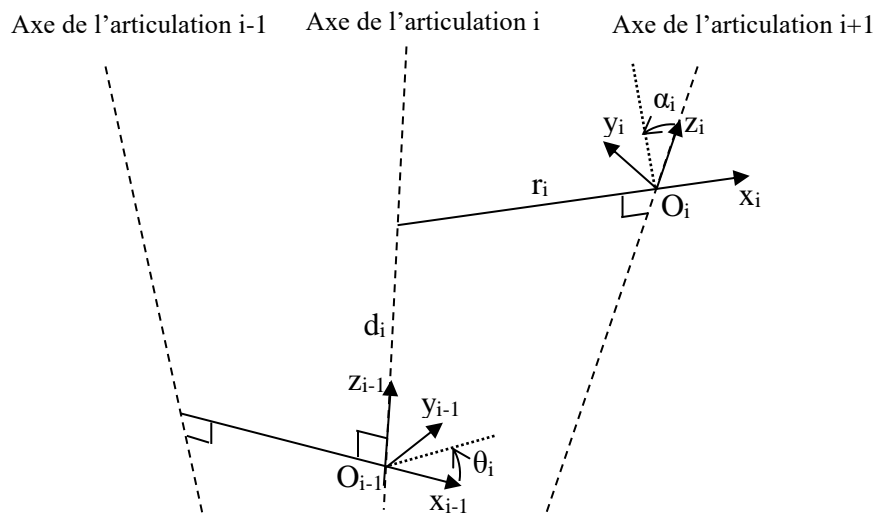


Figure 1.10. Paramètres géométriques de Denavit-Hartenberg

La matrice de transformation homogène permettant le passage entre les repères R_{i-1} et R_i , selon la convention de Denavit-Hartenberg est obtenue par la multiplication des matrices de transformation homogène élémentaires [21]:

$${}^{i-1}T_i = Rot(z_{i-1}, \theta_i). Trans(z_{i-1}, d_i). Trans(x_i, r_i). Rot(x_i, \alpha_i) \quad (1.14)$$

$$= \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \cdot \sin \theta_i & \sin \alpha_i \cdot \sin \theta_i & r_i \cdot \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cdot \cos \theta_i & -\sin \alpha_i \cdot \cos \theta_i & -r_i \cdot \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.15)$$

1.3.3.2. Convention de Denavit-Hartenberg modifiée [6]

En gardant la même configuration de la figure 1.9, le positionnement des repères se fait selon les règles suivantes :

- L'axe z_i est porté par l'axe de l'articulation i ,
- L'axe x_{i-1} est porté par la perpendiculaire commune aux axes z_{i-1} et z_i . Si z_{i-1} et z_i sont parallèle ou colinéaire, alors le choix de x_i n'est pas unique.

Le passage du repère R_{i-1} au repère R_i , s'exprime en fonction de quatre paramètres géométriques, et donnés par :

- α_i correspond à l'angle entre les axes z_{i-1} et z_i pour une rotation autour de x_{i-1} .
- d_i correspond à la distance entre les axes z_{i-1} et z_i le long de x_{i-1} .
- θ_i correspond à l'angle entre les axes x_{i-1} et x_i pour une rotation autour de z_i .
- r_i correspond à la distance entre les axes x_{i-1} et x_i le long de z_i .

Les paramètres de la convention de Denavit-Hartenberg modifiée sont définis par la figure 1.11.

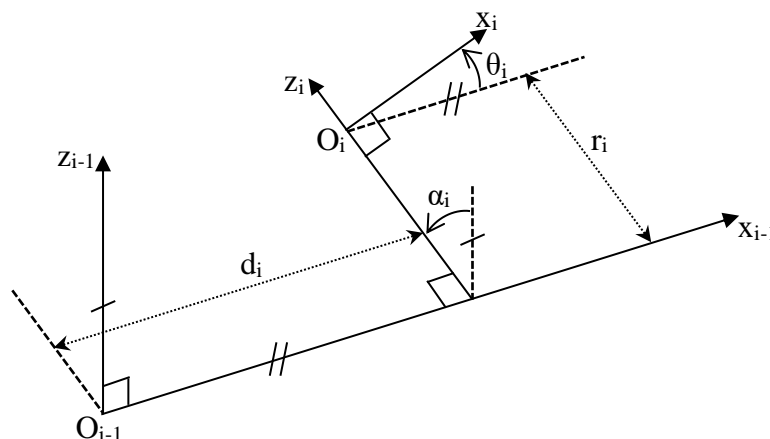


Figure 1.11. Paramètres géométriques pour une structure ouverte simple

On note q_i la variable articulaire associée à la $i^{\text{ème}}$ articulation, selon le type de l'articulation nous avons :

$$q_i = \begin{cases} \theta_i, & \text{lorsque l'articulation est de type rotoïde} \\ r_i, & \text{lorsque l'articulation est de type prismatique} \end{cases} \quad (1.16)$$

Selon la convention de Denavit-Hartenberg modifiée, la matrice de transformation homogène permettant de définir le repère R_i dans le repère R_{i-1} est donnée par l'expression suivante [23] :

$${}^{i-1}T_i = Rot(x_{i-1}, \alpha_i).Trans(x_{i-1}, d_i).Rot(z_i, \theta_i).Trans(z_i, r_i) \quad (1.17)$$

$$= \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & d_i \\ \cos \alpha_i \cdot \sin \theta_i & \cos \alpha_i \cdot \cos \theta_i & -\sin \alpha_i & -r_i \cdot \sin \alpha_i \\ \sin \alpha_i \cdot \sin \theta_i & \sin \alpha_i \cdot \cos \theta_i & \cos \alpha_i & r_i \cdot \cos \alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.18)$$

1.3.3.3. Expression du modèle géométrique direct (MGD)

On appelle l'ensemble des relations permettant d'exprimer la position et l'orientation de l'organe terminal en fonction des variables articulaires un modèle géométrique direct, représenté par l'expression suivante [6]:

$$\vec{X} = f(\vec{q}) \quad (1.19)$$

Avec :

\vec{X} est un vecteur qui représente la situation de l'organe terminal, et défini par l coordonnées opérationnelles.

$$\vec{X} = [x_1 \ x_2 \ x_3 \ \dots \ x_l] \quad (1.20)$$

\vec{q} est le vecteur des variables articulaires de dimension n .

$$\vec{q} = [q_1 \ q_2 \ q_3 \ \dots \ q_n] \quad (1.21)$$

f une fonction vectorielle statique.

Sachant que la position et l'orientation d'un repère R_i dans un repère de base R_0 sont déduite de la multiplication des matrices de transformation homogène selon l'expression (1.20) [6]:

$${}^0T_i = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \dots {}^{i-1}T_i \quad (1.22)$$

Le modèle géométrique direct permet d'exprimer la situation du repère lié à l'organe terminal R_n dans le repère de base R_0 , à l'aide des matrices de transformation homogène en partant de la base jusqu'à l'organe terminal :

$${}^0T_n = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \dots {}^{n-1}T_n \quad (1.23)$$

1.3.3.4. Modèle géométrique inverse (MGI)

Le modèle géométrique direct tel qu'il est cité dans le précédent paragraphe, permet d'exprimer les coordonnées opérationnelles qui définissent la position et l'orientation de l'organe terminal en fonction des variables articulaires. A l'inverse du MGD, le modèle géométrique inverse permet de calculer les coordonnées articulaires qui correspondent à une position désirée de l'organe terminal, ce qui peut être exprimé par la relation suivante [6]:

$$\vec{q} = g(\vec{X}) \quad (1.24)$$

Lorsqu'elle existe, la forme explicite qui donne toutes les solutions possibles (Il y a rarement unicité de solution) constitue ce que l'on appelle le modèle géométrique inverse noté MGI [6].

Il n'y a pas de méthode spécifique pour la détermination du modèle géométrique inverse, mais une multitude de méthodes adaptées à des robots ayant des classes de cinématiques particulières, Parmi ces méthodes nous citons :

- Les méthodes géométriques [24] [25] : détermination directe de chaque coordonnée articulaire à partir de considérations géométriques. Ces méthodes dépendent de l'architecture du robot.
- La méthode de Pieper [26] : utilisée pour trouver les variables articulaire pour les robots possédant six ddl, avec trois articulations de type rotoïdes d'axes concourants ou trois articulations prismatiques.
- La méthode de Paul [27] : Cette méthode traite chaque cas particulier séparément, et convient pour une grande partie des robots industriels.

Des méthodes numériques peuvent être utilisées dans le cas où une forme explicite du modèle géométrique est inexistante, pour trouver une solution particulière qui sera locale et qui dépendra des conditions initiales [6]. Nous pouvons citer à titre d'exemple :

- Les méthodes de type Newton RAPHSON : procédant par linéarisation de la matrice de passage du mécanisme [26].
- Les méthodes établies sur la transposée de la matrice jacobienne [28].
- Les méthodes conçues sur la base du modèle différentiel inverse [25].

1.3.4. Modélisation cinématique

1.3.4.1. Modèle cinématique direct (MCD) [20]

Le modèle cinématique direct des robots manipulateurs exprime les vitesses des coordonnées opérationnelles \dot{X} en fonction des vitesses des variables articulaire \dot{q} , il est exprimé par :

$$\dot{X} = J(q) \cdot \dot{q} \quad (1.25)$$

Où $J(q)$ est une matrice de dimension $(l \times n)$ nommée matrice jacobienne du mécanisme, qui dépend des coordonnées articulaires q telle que :

$$J(q) = \frac{\partial X}{\partial q} \quad (1.26)$$

Cette même matrice sert à calculer le modèle différentiel direct d'un robot, qui exprime la relation entre les variations des coordonnées opérationnelles dX et les variations des coordonnées articulaires dq , tel que :

$$dX = J(q) \cdot dq \quad (1.27)$$

1.3.4.1.1. Calcul indirect de la matrice jacobienne

La façon la plus simple et la plus naturelle de calculer le jacobien est d'utiliser la dérivée du modèle géométrique direct $X = f(q)$. Ainsi, les éléments de la matrice jacobienne sont calculés en utilisant la relation suivante [20] :

$$j_{ij} = \frac{df_i(q)}{dq_j} \quad \text{pour } i = 1, \dots, l; j = 1, \dots, n \quad (1.28)$$

Où j_{ij} représente l'élément (i, j) de la matrice jacobienne J .

Pour des robots ayant deux ou trois degré de liberté, cette méthode s'avère très pratique et de mise en œuvre facile.

1.3.4.1.2. Calcul direct de la matrice jacobienne

Une autre approche directe pour calculer la matrice jacobienne est possible, sans avoir recours au calcul de la dérivée du modèle géométrique direct. Cette approche est plus pratique pour des robots ayant n degré de liberté. La matrice jacobienne obtenue dans ce cas est nommée jacobienne de base [6]. Elle est obtenue par un calcul direct utilisant la relation entre le torseur cinématique et le vecteur des vitesses articulaires \dot{q} . Sachant que, les éléments du torseur cinématique du repère R_n sont les vitesses de rotation w_n et de translation v_n du repère R_n .

$$\begin{aligned} \begin{bmatrix} v \\ w \end{bmatrix} &= J(q) \cdot \dot{q} \\ &= \begin{bmatrix} J_v(q) \\ J_w(q) \end{bmatrix} \cdot \dot{q} \end{aligned} \quad (1.29)$$

Où w et v sont les vecteurs de vitesse angulaire et linéaire de l'organe terminal représenté dans l'espace cartésien.

Il faut noter que v est obtenu en dérivant par rapport au temps le vecteur M . Par contre, w ne représente pas la dérivée d'une représentation quelconque de l'orientation [6].

Dans [20] [6] [19] différentes méthodes sont présentées pour le calcul de la matrice jacobienne.

1.3.4.2. Modèle cinématique inverse (MCI)

Le modèle cinématique inverse d'un robot manipulateur défini à partir d'une configuration des variables articulaires q donnée les vitesses articulaires \dot{q} qui permettent au repère associé à l'organe terminal d'avoir une vitesse opérationnelle \dot{X} imposée. Le calcul du MCI revient à l'inversion de la matrice jacobienne du robot :

$$\dot{q} = J(q)^{-1} \cdot \dot{X} \quad (1.30)$$

L'inversion de la matrice jacobienne est relativement simple lorsque celle-ci est carrée, d'ordre n et son déterminant est différent de zéro (cas régulier). Lorsqu'on a une configuration singulière, le calcul de J^{-1} n'est pas possible. Généralement la matrice pseudo inverse J^+ est alors utilisée, avec :

$$J^+ = (J^T \cdot J)^{-1} \cdot J^T \quad (1.31)$$

1.3.5. Modélisation dynamique

Tandis que le modèle cinématique décrit le mouvement du robot sans tenir compte des forces et des couples produisant le mouvement, le modèle dynamique décrit explicitement la relation entre les force (et/ou couples) exercés par les actionneurs et les positions, vitesses et accélérations des articulations. Les équations du mouvement issues du modèle dynamique sont importantes à considérer dans la conception de robots, dans la simulation et l'animation du mouvement d'un robot, et dans la conception d'algorithmes de commande [19]. Ce modèle est représenté par une relation de la forme [6] :

$$\tau = f(q, \dot{q}, \ddot{q}, f_e) \quad (1.32)$$

Avec :

- τ : le vecteur des forces/couples des actionneurs, selon que l'articulation est prismatique ou rotoïde (pivot),
- q : le vecteur des positions articulaires,
- \dot{q} : le vecteur des vitesses articulaires,
- \ddot{q} : le vecteur des accélérations articulaires,
- f_e : le vecteur représentant les efforts extérieurs (forces et moments) que le robot exerce sur l'environnement.

La relation (1.32) est couramment appelée modèle dynamique inverse, ou modèle dynamique tout court.

La relation qui exprime les accélérations articulaires en fonction des positions, vitesses et forces et/ou couples est appelée modèle dynamique direct, et est représenté par l'expression :

$$\ddot{q} = f(q, \dot{q}, \tau, f_e) \quad (1.33)$$

Pour déterminer le modèle dynamique d'un robot, Plusieurs formalismes ont été développés [6] [29]. Les deux formalismes les plus utilisés sont celui de **Lagrange** [27] [30] [31], qui exprime le comportement dynamique du robot en termes d'énergies, et celui de **Newton-Euler** [32] [33] [34], se basant sur les forces et les moments.

1.3.5.1. Formalisme de Newton-Euler

Le principe de la formulation de Newton-Euler des équations du mouvement est d'abord d'écrire les équations définissant les vitesses et accélérations linéaires et angulaires de chaque corps, puis d'écrire les équations qui définissent les forces et les moments qui s'exercent sur les corps successifs en cours de mouvement [35]. Il est basé donc sur les théorèmes généraux de la mécanique.

Une fois qu'un repère a été fixé à chaque corps, on définit m_j la masse du corps j , g_j la position de son centre de gravité dans son repère, et I_{g_j} sa matrice d'inertie par rapport à son centre de gravité. En faisant appel aux lois fondamentales de la dynamique, Les équations de Newton-Euler sont exprimées par :

$$F_j = m_j \dot{V}_j \quad (1.34)$$

$$M_j = I_{g_j} \dot{w}_j + w_j \times (I_{g_j} w_j) \quad (1.35)$$

Où F_j est la résultante des forces extérieures appliquées sur le corps C_j , \dot{V}_j vitesse du centre de gravité du corps C_j , M_j est la résultante des couples extérieures appliqués sur C_j par rapport à son centre de gravité G_j , w_j est la vitesse de rotation du corps C_j par rapport au centre de gravité et I_{g_j} est sa matrice d'inertie calculée dans un repère dont l'origine est le centre de gravité g_j du corps C_j .

Le calcul du modèle dynamique peut se faire en ligne par la méthode de Luh, décrite dans [33], sur double récurrence et en utilisant les équations (1.34) et (1.35). La première récurrence de la base vers l'organe terminal du robot pour calculer les vitesses et accélérations des corps puis du torseur dynamique. Une seconde récurrence de l'organe terminal vers la base du robot pour calculer les couples des actionneurs.

1.3.5.2. Formalisme d'Euler-Lagrange

Le formalisme d'Euler-Lagrange est basé sur la description des équations de mouvement en termes d'énergies. Une application des équations d'Euler-Lagrange pour représenter le mouvement d'un robot manipulateur à n degrés de liberté conduit à un système de n équations différentielles ordinaires non linéaires de second ordre couplées de la forme :

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_j} - \frac{\partial L}{\partial q_j} = \tau_j \quad j = 1, 2, \dots, n \quad (1.36)$$

Avec :

$$L = E_{cin} - E_{pot} \quad (1.37)$$

Où :

L : Le lagrangien du mécanisme.

E_{cin} : L'énergie cinétique totale du mécanisme.

E_{pot} : L'énergie potentielle totale du mécanisme.

q : Vecteur des positions articulaires, tel que $q = [q_1, q_2, \dots, q_n]^T$

\dot{q} : Vecteur des vitesses articulaires, tel que $\dot{q} = [\dot{q}_1, \dot{q}_2, \dots, \dot{q}_n]^T$

τ : Forces et couples généralisés appliqués au mécanisme.

1.3.5.2.1. Calcul de l'énergie cinétique

Pour un robot manipulateur à n articulations, l'énergie cinétique totale est la somme des énergies cinétiques de rotation et de translation de tous les corps constituant le robot. Elle est donnée par :

$$E_{cin} = \sum_{j=1}^n \frac{1}{2} (w_j^T I_{gj} w_j + m_j V_j^T V_j) \quad (1.38)$$

Où m_j est la masse du corps j , I_{gj} est la matrice du tenseur d'inertie autour du centre de masse g_i .

Nous avons vu précédemment que les vitesses linéaires et angulaires de n'importe quel point sur n'importe quel corps peuvent être exprimées en termes de matrice jacobienne et de dérivée des variables articulaire :

$$v_j = J_{vj}(q) \cdot \dot{q}, \quad w_j = J_{wj}(q) \cdot \dot{q} \quad (1.39)$$

En remplaçant l'équation (1.39) dans (1.38), il vient :

$$E_{cin} = \frac{1}{2} \dot{q}^T \sum_{j=1}^n [J_{wj}(q)^T {}^0R_j(q) I_{gj} {}^0R_j(q)^T J_{wj}(q) + m_j J_{vj}(q)^T J_{vj}(q)] \dot{q} \quad (1.40)$$

Tel que 0R_j est la matrice de rotation du repère R_j dans le repère R_0 .

En d'autre terme, l'énergie cinétique du robot peut prendre la forme compacte suivante :

$$E_{cin} = \frac{1}{2} \dot{q}^T M(q) \dot{q} \quad (1.41)$$

Où $M(q)$ appelée matrice d'inertie, est une matrice symétrique et définie positive qui dépend généralement de la configuration.

1.3.5.2.2. Calcul de l'énergie potentielle

Considérons maintenant le terme de l'énergie potentielle. Dans le cas des dynamiques rigides, la seule source d'énergie potentielle est la gravité. L'énergie potentielle du j -ème corps peut être calculée en supposant que la masse de l'objet entier est concentrée en son centre de masse et est donnée par [19]:

$$E_{pot_j} = g^T r_{c_j} m_j \quad (1.42)$$

Où g est le vecteur donnant la direction de la gravité dans le repère inertiel et le vecteur r_{c_j} donne les coordonnées du centre de masse du corps j . L'énergie potentielle totale d'un robot à n articulations est donc :

$$E_{pot} = \sum_{j=1}^n E_{pot_j} = \sum_{j=1}^n g^T r_{c_j} m_j \quad (1.43)$$

L'énergie potentielle est fonction uniquement des coordonnées généralisées et non de leurs dérivées, c'est-à-dire que l'énergie potentielle dépend de la configuration du robot mais pas de sa vitesse [19].

1.3.5.2.3. Equations du mouvement [19]

L'énergie cinétique est une fonction quadratique du vecteur \dot{q} de la forme :

$$E_{cin} = \frac{1}{2} \dot{q}^T M(q) \dot{q} := \frac{1}{2} \sum_{i,j} m_{ij}(q) \dot{q}_i \dot{q}_j \quad (1.44)$$

Où $M(q)$ la matrice d'inertie de dimension $(n \times n)$ est symétrique et définie positive pour chaque $q \in R^n$.

Le Lagrangien pour un tel système est représenté par l'expression suivante :

$$L = E_{cin} - E_{pot} = \frac{1}{2} \sum_{i,j} m_{ij}(q) \dot{q}_i \dot{q}_j - E_{pot}(q) \quad (1.45)$$

Par la suite, nous allons procéder au calcul de l'expression du premier terme de l'équation (1.36), en commençant d'abord par les expressions de la dérivée partielle du lagrangien par rapport aux vitesses articulaires. Puisque l'énergie potentielle $E_{pot} = E_{pot}(q)$ est indépendante des vitesses articulaires \dot{q} , nous avons :

$$\frac{\partial L}{\partial \dot{q}_k} = \sum_j m_{kj} \dot{q}_j \quad (1.46)$$

En calculant la dérivée de $\frac{\partial L}{\partial \dot{q}_k}$ par rapport au temps. Il vient :

$$\begin{aligned} \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_k} &= \sum_i m_{kj} \ddot{q}_j + \sum_j \frac{d}{dt} m_{kj} \dot{q}_j \\ &= \sum_j m_{kj} \ddot{q}_j + \sum_{i,j} \frac{\partial m_{kj}}{\partial q_i} \dot{q}_i \dot{q}_j \end{aligned} \quad (1.47)$$

En dérivant le lagrangien par rapport à q_k Nous avons l'expression suivante :

$$\frac{\partial L}{\partial q_k} = \frac{1}{2} \sum_{i,j} \frac{\partial m_{ij}}{\partial q_k} \dot{q}_i \dot{q}_j - \frac{\partial E_{pot}}{\partial q_k} \quad (1.48)$$

En remplaçant les équations (1.47) et (1.48) dans (1.36), les équations d'Euler-Lagrange s'écrivent de la forme suivante :

$$\sum_j m_{kj} \ddot{q}_j + \sum_{i,j} \left(\frac{\partial m_{kj}}{\partial q_i} - \frac{1}{2} \frac{\partial m_{ij}}{\partial q_k} \right) \dot{q}_i \dot{q}_j - \frac{\partial E_{pot}}{\partial q_k} = \tau_k \quad (1.49)$$

En échangeant l'ordre de sommation et en profitant de la symétrie, nous pouvons montrer que :

$$\sum_{i,j} \left(\frac{\partial m_{kj}}{\partial q_i} \right) \dot{q}_i \dot{q}_j = \frac{1}{2} \sum_{i,j} \left(\frac{\partial m_{kj}}{\partial q_i} - \frac{\partial m_{ki}}{\partial q_j} \right) \dot{q}_i \dot{q}_j \quad (1.50)$$

Par conséquent

$$\sum_{i,j} \left(\frac{\partial m_{kj}}{\partial q_i} - \frac{1}{2} \frac{\partial m_{ij}}{\partial q_k} \right) \dot{q}_i \dot{q}_j = \sum_{i,j} \frac{1}{2} \left(\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j} - \frac{\partial m_{ij}}{\partial q_k} \right) \dot{q}_i \dot{q}_j \quad (1.51)$$

Posons :

$$c_{ijk} := \frac{1}{2} \left(\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j} - \frac{\partial m_{ij}}{\partial q_k} \right) \quad (1.52)$$

Les termes c_{ijk} sont connus sous l'appellation *symboles de Christoffel*. Il faut noter que pour une valeur fixe de k nous avons $c_{ijk} = c_{jik}$ ce qui facilite les calculs.

Finalement en tenant compte de la notation suivante :

$$Q_k = \frac{\partial E_{pot}}{\partial q_k} \quad (1.53)$$

L'équation d'Euler-Lagrange prend la forme ci-dessous [19]:

$$\sum_i m_{kj} \ddot{q}_j + \sum_{i,j} c_{ijk}(q) \dot{q}_i \dot{q}_j + Q_k(q) = \tau_k \quad k = 1, \dots, n \quad (1.54)$$

Dans l'équation (1.54), il existe trois types de termes. Le premier implique la deuxième dérivée des coordonnées généralisées. Les seconds sont des termes quadratiques de \dot{q} , où les coefficients peuvent dépendre de q . Ceux-ci sont en outre classés en deux types. Les termes impliquant un produit de type \dot{q}_i^2 sont appelés centrifuges, tandis que ceux impliquant un produit de type $\dot{q}_i \dot{q}_j$ où $i \neq j$ sont appelés termes de Coriolis. Le troisième type de termes est celui impliquant seulement q mais pas ses dérivés. Il est clair que ces derniers résultent de la différenciation de l'énergie potentielle. L'équation (1.54) est couramment écrite sous forme matricielle de la forme [19]:

$$M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + g(q) = \tau \quad (1.55)$$

On peut calculer les éléments de la matrice des termes centrifuges et Coriolis $C(q, \dot{q})$ à partir du symbole de *Christoffel*, où le $k, j^{\text{ième}}$ élément est défini par [19]:

$$\begin{aligned} c_{kj} &= \sum_{i=1}^n c_{ijk}(q) \dot{q}_i \\ &= \sum_{i=1}^n \frac{1}{2} \left(\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j} - \frac{\partial m_{ij}}{\partial q_k} \right) \dot{q}_i \end{aligned} \quad (1.56)$$

1.3.5.2.4. Exemple de modélisation d'un robot à 2ddl

Considérons un robot manipulateur à 2ddl, illustré par la figure (1.12), dont les deux articulations sont de type rotoïde. Nous avons donc : pour $j=1, 2$, q_j désigne l'angle de

l'articulation, qui sert également de coordonnée généralisée ; l_j désigne la longueur du corps j ; l_{c_j} désigne la distance entre l'articulation précédente et le centre de masse du corps j ; m_j désigne la masse du corps j ; et I_j le moment d'inertie du corps j autour d'un axe sortant de la page, passant par le centre de masse.

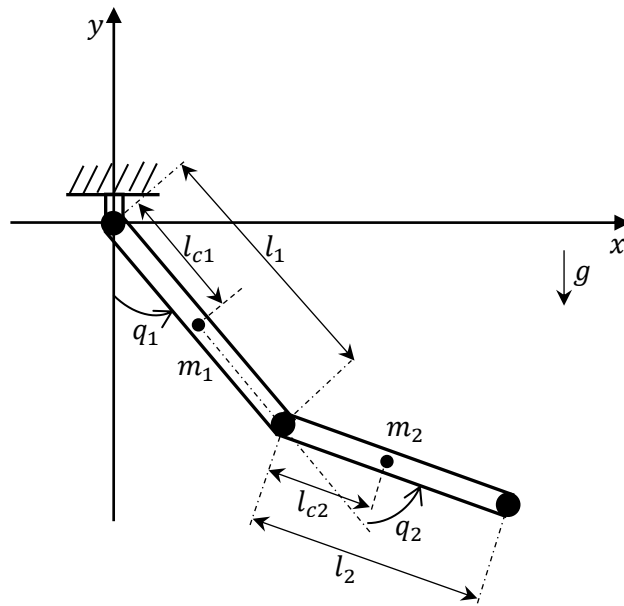


Figure 1.12. Bras manipulateur à 2ddl

Avec : $q = [q_1 \quad q_2]$ et $\tau = [\tau_1 \quad \tau_2]$ les couples fournis par les actionneurs.

Les coordonnées cartésiennes du robot manipulateur sont données par :

$$\begin{cases} x_1 = l_{c1} \sin q_1 \\ y_1 = -l_{c1} \cos q_1 \end{cases} \quad (1.57)$$

$$\begin{cases} x_2 = l_1 \sin q_1 + l_{c2} \sin(q_1 + q_2) \\ y_2 = -l_1 \cos q_1 - l_{c2} \cos(q_1 + q_2) \end{cases} \quad (1.58)$$

Pour calculer l'énergie cinétique nous aurons besoin des matrices jacobiniennes :

$$v_1 = J_{v1} \dot{q} \quad (1.59)$$

Avec

$$J_{v1} = \begin{bmatrix} l_{c1} \cos q_1 & 0 \\ l_{c1} \sin q_1 & 0 \\ 0 & 0 \end{bmatrix}$$

Et de même pour la deuxième vitesse :

$$v_2 = J_{v2} \dot{q} \quad (1.60)$$

Avec

$$J_{v2} = \begin{bmatrix} l_1 \cos q_1 + l_{c2} \cos(q_1 + q_2) & l_{c2} \cos(q_1 + q_2) \\ l_1 \sin q_1 + l_{c2} \sin(q_1 + q_2) & l_{c2} \sin(q_1 + q_2) \\ 0 & 0 \end{bmatrix}$$

Par conséquent, la première partie de l'énergie cinétique qui dépend des vitesses de translation est :

$$\begin{aligned} E_{cin1} &= \frac{1}{2} m_1 v_1^T v_1 + \frac{1}{2} m_2 v_2^T v_2 \\ &= \frac{1}{2} \dot{q}^T (m_1 J_{v1}^T J_{v1} + m_2 J_{v2}^T J_{v2}) \dot{q} \end{aligned} \quad (1.61)$$

Ensuite, nous traitons des termes de vitesse angulaire. En raison de la nature particulièrement simple de ce robot, de nombreuses difficultés potentielles ne se posent pas. Il est clair que :

$$w_1 = \dot{q}_1 \quad \text{et} \quad w_2 = (\dot{q}_1 + \dot{q}_2) \quad (1.62)$$

La deuxième partie de l'énergie cinétique qui dépend des vitesses de rotation est donc exprimée par :

$$E_{cin2} = \frac{1}{2} \dot{q}^T (I_1 \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + I_2 \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}) \dot{q} \quad (1.63)$$

L'énergie cinétique totale du mécanisme est alors :

$$E_{cin} = E_{cin1} + E_{cin2} = \frac{1}{2} \dot{q}^T (m_1 J_{v1}^T J_{v1} + m_2 J_{v2}^T J_{v2} + \begin{bmatrix} I_1 + I_2 & I_2 \\ I_2 & I_2 \end{bmatrix}) \dot{q} \quad (1.64)$$

En se basant sur les équations (1.44) et (1.64), Nous sommes prêts maintenant à former la matrice d'inertie $M(q)$:

$$M(q) = m_1 J_{v1}^T J_{v1} + m_2 J_{v2}^T J_{v2} + \begin{bmatrix} I_1 + I_2 & I_2 \\ I_2 & I_2 \end{bmatrix} \quad (1.65)$$

En développant les multiplications ci-dessus et en utilisant quelques lois de la trigonométrie, les éléments obtenus de la matrice $M(q)$ sont :

$$\begin{cases} m_{11} = m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos q_2) + I_1 + I_2 \\ m_{12} = m_{21} = m_2 (l_{c2}^2 + l_1 l_{c2} \cos q_2) + I_2 \\ m_{22} = m_2 l_{c2}^2 + I_2 \end{cases} \quad (1.66)$$

Nous pouvons maintenant calculer les symboles de Christoffel donnés par (1.56). Ce qui donne :

$$\begin{cases} c_{111} = \frac{1}{2} \frac{\partial m_{11}}{\partial q_1} = 0 \\ c_{121} = c_{211} = \frac{1}{2} \frac{\partial m_{11}}{\partial q_2} = -m_2 l_1 l_{c2} \sin q_2 \\ c_{221} = \frac{\partial m_{12}}{\partial q_2} - \frac{1}{2} \frac{\partial m_{22}}{\partial q_1} = -m_2 l_1 l_{c2} \sin q_2 \\ c_{112} = \frac{\partial m_{21}}{\partial q_1} - \frac{1}{2} \frac{\partial m_{11}}{\partial q_2} = m_2 l_1 l_{c2} \sin q_2 \\ c_{122} = c_{212} = \frac{1}{2} \frac{\partial m_{22}}{\partial q_1} = 0 \\ c_{222} = \frac{1}{2} \frac{\partial m_{22}}{\partial q_2} = 0 \end{cases} \quad (1.67)$$

Pour calculer l'énergie potentielle totale du robot, il suffit d'additionner les deux énergies potentielles des deux segments. Nous aurons donc :

$$E_{pot} = m_1 g h_1 + m_2 g h_2 \quad (1.68)$$

Avec h_1 et h_2 les hauteurs des deux centres de masse des deux segments c_1 et c_2 respectivement.

$$\begin{cases} h_1 = y_1 = -l_{c1} \cos q_1 \\ h_2 = y_2 = -l_1 \cos q_1 - l_{c2} \cos(q_1 + q_2) \end{cases}$$

L'équation (1.68) devient :

$$E_{pot} = (m_1 l_{c1} + m_2 l_1) g \cos q_1 + m_2 l_{c2} g \cos(q_1 + q_2) \quad (1.69)$$

Par conséquent, les fonctions Q_k définies dans (1.53) deviennent :

$$Q_1 = \frac{\partial E_{pot}}{\partial q_1} = (m_1 l_{c1} + m_2 l_1) g \sin q_1 + m_2 l_{c2} g \sin(q_1 + q_2) \quad (1.70)$$

$$Q_2 = \frac{\partial E_{pot}}{\partial q_2} = m_2 l_{c2} g \sin(q_1 + q_2) \quad (1.71)$$

Finalement, en se référant à (1.54) nous pouvons écrire les équations dynamiques du mécanisme. Ce qui conduit à :

$$\begin{cases} \tau_1 = m_{11} \ddot{q}_1 + m_{12} \ddot{q}_2 + c_{121} \dot{q}_1 \dot{q}_2 + c_{211} \dot{q}_2 \dot{q}_1 + c_{221} \dot{q}_2^2 + Q_1 \\ \tau_2 = m_{21} \ddot{q}_1 + m_{22} \ddot{q}_2 + c_{112} \dot{q}_1^2 + Q_2 \end{cases} \quad (1.72)$$

Reprenons la forme générale du modèle dynamique d'un robot manipulateur (1.55), et par identification avec (1.72), nous obtenons :

$$\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} c_{121} \dot{q}_2 & c_{211} \dot{q}_1 + c_{221} \dot{q}_2 \\ c_{112} \dot{q}_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (1.73)$$

La matrice d'inertie $M(q)$ est alors égale à :

$$M(q) = \begin{bmatrix} m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos q_2) + I_1 + I_2 & m_2 (l_{c2}^2 + l_1 l_{c2} \cos q_2) + I_2 \\ m_2 (l_{c2}^2 + l_1 l_{c2} \cos q_2) + I_2 & m_2 l_{c2}^2 + I_2 \end{bmatrix} \quad (1.74)$$

La matrice des termes Coriolis et centrifuges $C(q, \dot{q})$ est donnée par :

$$C(q, \dot{q}) = \begin{bmatrix} -m_2 l_1 l_{c2} \sin q_2 \dot{q}_2 & -m_2 l_1 l_{c2} \sin q_2 \dot{q}_1 - m_2 l_1 l_{c2} \sin q_2 \dot{q}_2 \\ m_2 l_1 l_{c2} \sin q_2 \dot{q}_1 & 0 \end{bmatrix} \quad (1.75)$$

Le vecteur des termes des forces de gravité $g(q)$ est exprimé par :

$$g(q) = \begin{bmatrix} (m_1 l_{c1} + m_2 l_1) g \sin q_1 + m_2 l_{c2} g \sin(q_1 + q_2) \\ m_2 l_{c2} g \sin(q_1 + q_2) \end{bmatrix} \quad (1.76)$$

1.3.5.2.5. Propriétés du modèle dynamique

Les équations de mouvement d'un bras à n corps peuvent être assez redoutables surtout si le robot contient une ou plusieurs articulations rotoïde [19]. Heureusement, ces équations ont des propriétés structurelles importantes, qui peuvent être exploitées à bon escient notamment pour développer des algorithmes de commande [19]. Nous allons voir certaines de ces propriétés, dont les plus importantes dépendent de la matrice d'inertie, qui joue un rôle important dans le calcul du modèle dynamique ainsi que la conception des lois de commande. Cette matrice qui ne dépend que de q est définie positive et symétrique.

Propriété 1 :

Pour une valeur fixe des variables articulaires q , soit $0 < \lambda_1(q) \leq \dots \leq \lambda_n(q)$ désignant les n valeurs propres de $M(q)$. Du fait que, $M(q)$ est définie positive, ces valeurs propres sont strictement positives. Il peut facilement être démontré que :

$$\lambda_1(q)I_{n \times n} \leq M(q) \leq \lambda_n(q)I_{n \times n} \quad (1.77)$$

Où $I_{n \times n}$ est la matrice identité de dimension $(n \times n)$.

Propriété 2 :

Soit un nombre positif α tel que :

$$M(q) \leq \alpha I_{n \times n} \quad \forall q \in R^n \quad (1.78)$$

La matrice $M(q)^{-1}$ existe et elle est définie positive.

Propriété 3 :

La matrice $\dot{M}(q) - 2C(q, \dot{q})$ est antisymétrique, ce qui permet d'écrire :

$$x^T [\dot{M}(q) - 2C(q, \dot{q})] x = 0 \quad \forall q, \dot{q}, x \in R^n \quad (1.79)$$

Cette relation entre la matrice d'inertie $M(q)$ et la matrice $C(q, \dot{q})$ est d'une importance fondamentale surtout pour les problèmes de commande des robots manipulateurs.

Propriété 4 :

Dans le cas où le robot ne contient que des articulations de type rotoïde, le vecteur des termes des forces de gravité $g(q)$ de dimension $(n \times 1)$, est lipschitzien. C'est-à-dire qu'il existe une constante strictement positive α_g tel que :

$$\|g(x) - g(y)\| \leq \alpha_g \|x - y\| \quad \forall x, y \in R^n \quad (1.80)$$

Propriété 5 :

Pour les robots ne possédant que des articulations rotoïdes, il existe une constante $\acute{\alpha}$ tel que

$$\|g(q)\| \leq \acute{\alpha} \quad \forall q \in R^n \quad (1.81)$$

1.4. Conclusion

Nous avons exposé dans ce chapitre les méthodes usuelles pour calculer les différents modèles représentant les robots manipulateurs et plus particulièrement ceux ayant une structure ouverte simple. Dans ce contexte nous pouvons citer quelques points :

- Le calcul du modèle géométrique qui dépend de l'architecture du robot ainsi que la tâche qu'il doit réaliser, repose sur l'utilisation des matrices de transformation homogène, suivant certaines règles imposées par la convention de Denavit-Hartenberg ou celle modifiée (Khalil-Kleinfinger). Il faut noter aussi que pour calculer le modèle géométrique inverse la méthode de PAUL s'avère la plus utilisée.
- Le calcul de la matrice jacobienne et de son inverse permet d'établir le modèle cinématique direct et inverse.

- Le calcul du modèle dynamique peut être réalisé à l'aide de deux formalismes. Le premier nécessitant moins de calcul est le formalisme de Newton-Euler. Tandis que le deuxième intéressant pour la simulation est celui d'Euler-Lagrange. Ce dernier nous l'avons utilisé pour calculer le modèle dynamique d'un robot manipulateur à 2ddl.

CHAPITRE 2

QUELQUES LOIS DE COMMANDE CLASSIQUE DES ROBOTS MANIPULATEURS Á CHAINE OUVERTE SIMPLE

2.1. Introduction

Plusieurs lois de commande classiques ont été proposées pour la commande des robots manipulateurs. Suivant leur objectif, on peut classer ces lois de commande en deux catégories. Celles qui traitent le problème de régulation qui consiste à faire déplacer l'organe terminale d'un point initial vers un point final d'une façon libre sans aucune imposition de trajectoire à suivre et celles qui traitent le problème de poursuite de trajectoire qui consiste dans ce cas à imposer à l'organe terminale de suivre une trajectoire prédéfinie en allant du point initial vers le point final, ceci devient intéressant lorsqu'il y a des obstacles dans l'espace de travail du manipulateur [10] [19] [17].

Nous allons voir dans ce chapitre en premier lieu quelques lois de commandes classiques dans le cas d'un problème de régulation [10] [36] [6] [37] [38]. Nous commencerons par la commande PID qui assure au robot en boucle fermée une stabilité asymptotique tout en gardant l'erreur de position nulle. Cette commande est très employée dans le secteur industriel, bien que le terme intégral dans cette commande puisse provoquer de sérieux problèmes de saturation. Nous allons voir aussi la commande PD avec compensation de l'effet de gravité appelée point à point qui permet d'assurer une stabilité asymptotique globale et de contourner le problème d'utilisation de grands gains dans la commande PD ce qui est indésirable en pratique [39].

En deuxième lieu nous nous intéresserons aux commandes classiques dans le cas d'un problème de poursuite de trajectoire, où nous allons exposer d'abord les différentes méthodes de génération de trajectoire [6] [38] [37], telle que les méthodes qui sont basées sur l'interpolation linéaire, l'interpolation polynomiale de degré trois, de degré cinq et l'interpolation Bang-Bang. Dans ce contexte, nous allons traiter la commande du couple calculé (computed torque control) [10] [37] qui permet une linéarisation et un découplage de la dynamique du robot manipulateur.

Nous ne pouvons pas dans le contexte de ce travail présenter en détail toutes les techniques de commande des robots manipulateurs. Pour avoir plus de détail, le lecteur pourra bien sûr se

référer à [40], [6], [19], [10], [17] et [7]. Un aperçu général sur les différentes techniques de commande rapportées dans la littérature est donc présenté dans ce manuscrit.

2.2. Lois de commande dans le cas du problème de régulation

Les problèmes de commande des robots manipulateurs peuvent être formulés comme étant la détermination de l'évolution des forces généralisées (forces dans le cas d'une translation ou couples dans le cas d'une rotation) que les actionneurs doivent exercer tout en satisfaisant certains critères de performance pour garantir l'exécution d'une tâche.

Considérant le modèle dynamique des robots manipulateurs sans frottement et sans perturbation décrivant un système à n équations différentielles du second ordre non linéaires et couplées, n étant le nombre d'articulations. Ce modèle est décrit (dans le chapitre 1 eq 1.55) par l'équation suivante :

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) \quad (2.1)$$

2.2.1. La loi de commande de type PID

La commande PID a été proposée pour les problèmes liés à l'utilisation de grands gains en utilisant la commande PD. L'ajout de l'action intégrale agit donc sur les forces de gravité qui sont considérées dans certains cas comme des perturbations externes. En introduisant l'action intégrale l'erreur s'annule en régime permanent tout en gardant les gains faibles.

La forme de la loi de commande par PID est donnée par (voir figure 2.1) :

$$\tau = k_p(q_d - q) + k_d(\dot{q}_d - \dot{q}) + k_i \int (q_d - q) dt \quad (2.2)$$

Avec $k_p, k_d, k_i \in R^{n \times n}$ sont des matrices diagonales et définies positives qui représentent respectivement les matrices des gains proportionnels, dérivés et intégral.

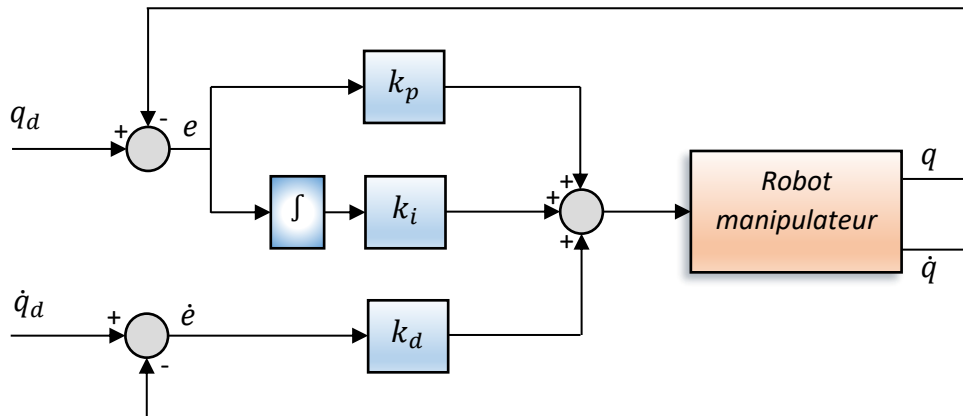


Figure 2.1. Schéma fonctionnel de la commande PID

La dynamique du robot manipulateur représentée par l'équation (2.1) peut être décrite en utilisant la représentation d'état, en posant $\chi_1 = q$, $\chi_2 = \dot{q}$ et $\chi = (\chi_1^T \ \chi_2^T)^T$, nous aurons :

$$\begin{cases} \dot{\chi}_1 = \chi_2 \\ \dot{\chi}_2 = M^{-1}(\chi_1)(-C(\chi_1, \chi_2)\chi_2 - g(\chi_1) + \tau) \end{cases} \quad (2.3)$$

Un vecteur des erreurs d'état est défini comme suit :

$$\begin{cases} \tilde{\chi}_1 = \chi_1 - \chi_{1d} \\ \tilde{\chi}_2 = \chi_2 \end{cases} \quad (2.4)$$

Avec χ_{1d} qui représente le vecteur des positions de référence q_d . Un troisième état est introduit composé de l'intégral de l'erreur de position et d'un terme de biais $-k_i^{-1}g(\chi_{1d})$.

$$\tilde{\chi}_3 = \int_0^t \tilde{\chi}_1 d\theta - k_i^{-1}g(\chi_{1d}) \quad (2.5)$$

En substituant l'équation (2.2) dans l'équation (2.3) et en tenant compte des équations (2.4) et (2.5), nous aurons :

$$\begin{cases} \dot{\tilde{\chi}}_1 = \tilde{\chi}_2 \\ \dot{\tilde{\chi}}_2 = M^{-1}(\chi_1)(-C(\chi_1, \tilde{\chi}_2)\tilde{\chi}_2 - g(\chi_1) - k_p \tilde{\chi}_1 - k_d \tilde{\chi}_2 - k_i(\tilde{\chi}_3 + g(\chi_{1d}))) \\ \dot{\tilde{\chi}}_3 = \tilde{\chi}_1 \end{cases} \quad (2.6)$$

Afin de pouvoir étudier la stabilité locale du système d'équation d'erreurs (2.6), on considère une approximation autour de l'origine de sa tangente linéaire.

$$\dot{\tilde{\chi}} = A \tilde{\chi} + h(\tilde{\chi}) \quad (2.7)$$

Avec

$$A = \begin{pmatrix} 0 & I & 0 \\ (-B_0 k_p + A_0) & -B_0 k_d & -k_i \\ I & 0 & 0 \end{pmatrix}, \tilde{\chi} = \begin{pmatrix} \tilde{\chi}_1 \\ \tilde{\chi}_2 \\ \tilde{\chi}_3 \end{pmatrix} \quad (2.8)$$

Où $h(\tilde{\chi})$ décrit les termes d'ordre supérieur de l'expansion de Taylor. Il faut noter que, compte tenu des propriétés du modèle dynamique du robot manipulateur, $h(\tilde{\chi})$ satisfait : $\|h(\tilde{\chi})\| \leq c_1 \|\tilde{\chi}\|^2$, où c_1 est une constante positive bornée. Les matrices A_0 et B_0 sont exprimées par :

$$A_0 = -\left. \frac{\partial(M^{-1}(\chi_1)g(\chi_1))}{\partial \chi_1} \right|_{\chi_1 = \chi_{1d}}, B_0 = M^{-1}(\chi_{1d}) \quad (2.9)$$

Pour un choix judicieux des matrices k_p , k_d et k_i permettant de rendre le système linéaire et autonome (2.7) asymptotiquement stable, les variables d'état du système en boucle fermée augmenté (2.7) tendent alors vers zéro. puisque $\tilde{\chi}_1 \rightarrow 0$, l'erreur statique peut être annulée sans aucune connaissance des perturbations.

La fonction V peut être définie si la matrice A est stable, par :

$$V = \tilde{\chi}^T P \tilde{\chi} \quad (2.10)$$

Où la matrice P est symétrique et définie positive qui satisfait $A^T P + P A = -Q$, pour une matrice Q symétrique et définie positive. Nous aurons alors :

$$\begin{aligned} V &= -\tilde{\chi}^T Q \tilde{\chi} + 2\tilde{\chi}^T P h(\tilde{\chi}) \\ &\leq -\lambda_{\min}(Q) \|\tilde{\chi}\|^2 + 2c_1 p_0 \|\tilde{\chi}\|^3 \\ &= -\|\tilde{\chi}\|^2 (\lambda_{\min}(Q) - 2c_1 p_0 \|\tilde{\chi}\|) \end{aligned} \quad (2.11)$$

Avec p_0 qui représente la borne supérieure de la norme de P .

Au voisinage de l'origine, le terme cubique qui est positif est dominé par le terme carré qui est négatif. Donc le point d'équilibre $\tilde{\chi} = 0$ est localement stable au sens de Lyapunov. Il est relativement simple maintenant de prouver la convergence vers zéro du vecteur de l'erreur, en invoquant le théorème de La Salle (voir ANNEXE A).

2.2.2. La loi de commande proportionnelle dérivée (PD) avec compensation de l'effet de gravité (point à point)

En appliquant une loi de commande de type PD, nous pouvons constater que la dynamique du robot manipulateur n'est pas utilisée. Par conséquent, les non linéarités du système ne sont pas compensées. L'idée alors est d'inclure dans la commande PD des termes dynamiques non linéaires. La compensation de l'effet de la gravité peut agir comme une correction qui compense l'ensemble des efforts qui provoquent le dépassement et un comportement transitoire asymétrique. Afin d'en profiter de cet avantage la commande PD est substituée par une commande PD avec ajout d'un terme des forces de gravité. Cette loi de commande est nommée aussi commande point à point.

Cette loi de commande permet donc un positionnement des différentes articulations du robot manipulateur autour d'une position désirée constante, de plus elle permet de lever la contrainte relative à l'utilisation de grand gain dans d'autre lois de commande, et ceci par la compensation de l'effet du vecteur de gravité [41].

La loi de commande est définie par [42] [7] (voir figure 2.2) :

$$\tau = g(q) - k_p(q - q_d) - k_d(\dot{q} - \dot{q}_d) \quad (2.12)$$

Où $k_p \in R^{n \times n}$ et $k_d \in R^{n \times n}$ sont respectivement les matrices des gains proportionnels et dérivés, ces deux matrices sont diagonales et définies positives.

Soit $\tilde{q} = q - q_d$ l'écart de position, avec q (mesurée) qui représente la position angulaire, et q_d représente la position désirée. Alors l'écart en vitesse est $\dot{\tilde{q}} = \dot{q} - \dot{q}_d = \dot{q}$.

En combinant les équations (2.1) et (2.12) la dynamique du système en boucle fermée est donnée après calcul par la relation suivante :

$$\ddot{\tilde{q}} = -M^{-1}(q)[C(q, \dot{q}) + k_d]\dot{\tilde{q}} + k_p\tilde{q} \quad (2.13)$$

Où $\ddot{\tilde{q}} = \ddot{q} - \ddot{q}_d = \ddot{q}$ est l'écart en accélération.

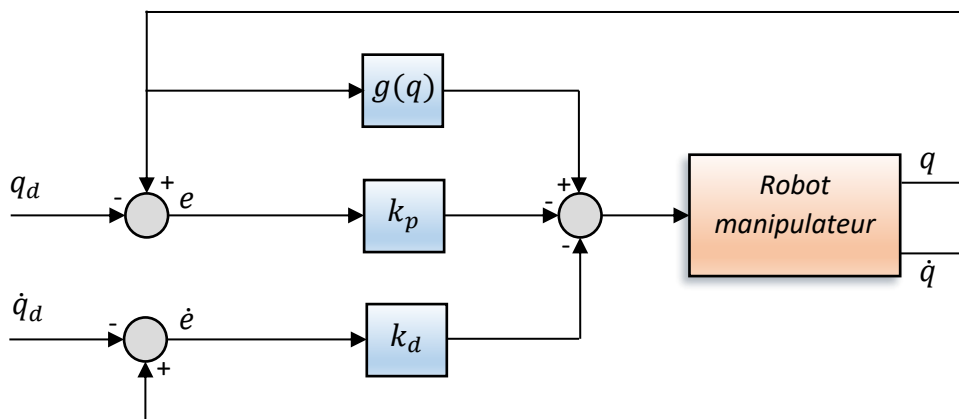


Figure 2.2. Schéma fonctionnel de la commande PD avec compensation de l'effet de gravité

Considérons le vecteur d'état $[\tilde{q}^T \ \dot{\tilde{q}}^T]^T$, nous pouvons écrire alors :

$$\frac{d}{dt} \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix} = \begin{bmatrix} \dot{\tilde{q}} \\ -M^{-1}(q)[C(q, \dot{q})\dot{\tilde{q}} + k_d\dot{\tilde{q}} + k_p\tilde{q}] \end{bmatrix} \quad (2.14)$$

Compte tenu du fait que la position désirée q_d est constante et de l'équation différentielle autonome. Cette équation admet l'origine $[\tilde{q}^T \ \dot{\tilde{q}}^T]^T = 0$ comme le seul point d'équilibre.

Pour étudier la stabilité de l'origine comme point d'équilibre, nous utilisons la méthode directe de Lyapunov. Considérons la fonction de Lyapunov candidate suivante :

$$V(\tilde{q}, \dot{\tilde{q}}) = \frac{1}{2} \dot{\tilde{q}}^T M(q) \dot{\tilde{q}} + \frac{1}{2} \tilde{q}^T k_p \tilde{q} \quad (2.15)$$

Où le terme $\frac{1}{2} \dot{\tilde{q}}^T M(q) \dot{\tilde{q}}$ représente l'énergie cinétique du robot. La fonction $V(\tilde{q}, \dot{\tilde{q}})$ est globalement définie positive puisque l'énergie cinétique $\frac{1}{2} \dot{\tilde{q}}^T M(q) \dot{\tilde{q}}$ est définie positive et d'autre part le terme quadratique $\frac{1}{2} \tilde{q}^T k_p \tilde{q}$ est défini positif, sachant que k_p est une matrice définie positive.

En dérivant cette fonction par rapport au temps nous obtenons :

$$\dot{V}(\tilde{q}, \dot{\tilde{q}}) = \dot{\tilde{q}}^T M(q) \ddot{\tilde{q}} + \frac{1}{2} \dot{\tilde{q}}^T \dot{M}(q) \dot{\tilde{q}} + \tilde{q}^T k_p \dot{\tilde{q}} \quad (2.16)$$

En substituant le terme $M(q) \ddot{\tilde{q}}$ par l'équation de la boucle fermée (2.14) et en éliminant le terme $\dot{\tilde{q}}^T [\dot{M}(q) - 2C(q, \dot{q})] \dot{\tilde{q}}$ en faisant appel à la propriété 3 du modèle dynamique des robots manipulateurs citée dans le premier chapitre, la dérivée de la fonction de Lyapunov devient :

$$\dot{V}(\tilde{q}, \dot{\tilde{q}}) = -\dot{\tilde{q}}^T k_d \dot{\tilde{q}} \leq 0 \quad (2.17)$$

Donc $\dot{V}(\tilde{q}, \dot{\tilde{q}}) \leq 0$ pour tout \tilde{q} et $\dot{\tilde{q}}$, l'origine est stable et toutes les solutions $\tilde{q}(t)$ et $\dot{\tilde{q}}(t)$ sont bornées selon le théorème A.1 (Voir Annexe A). Nous pouvons utiliser le théorème de LaSalle (voir Annexe A) afin d'analyser la stabilité asymptotique globale de l'origine puisque l'équation en boucle fermée (2.14) est explicitement indépendante du temps.

Pour cela, considérons l'ensemble Ω donné par :

$$\begin{aligned} \Omega &= \{[\tilde{q} \ \dot{\tilde{q}}]^T \in R^{2n} : \dot{V}(\tilde{q}, \dot{\tilde{q}}) = 0\} \\ &= \{\tilde{q} \in R^n : \dot{\tilde{q}} = 0 \in R^n\} \end{aligned} \quad (2.18)$$

On remarque que $\dot{V}(\tilde{q}, \dot{\tilde{q}}) = 0$ si et seulement si $\dot{\tilde{q}} = 0$. Pour qu'une solution $[\tilde{q} \ \dot{\tilde{q}}]^T$ appartienne à Ω pour tout $t \geq 0$, $\dot{\tilde{q}}(t) = 0$ est une condition nécessaire et suffisante pour tout $t \geq 0$. Et par conséquent $\ddot{\tilde{q}}(t) = 0$ pour tout $t \geq 0$. En tenant compte de cela, nous pouvons conclure de l'équation en boucle fermée (2.14) que si $[\tilde{q}^T \ \dot{\tilde{q}}^T]^T \in \Omega$ pour tout $t \geq 0$, alors :

$$0 = -M^{-1}(q)k_p\tilde{q}(t) \quad (2.19)$$

Ce qui signifie que $\tilde{q}(t) = 0$ pour tout $t \geq 0$. Donc, $[\tilde{q}(0)^T \ \dot{\tilde{q}}(0)^T]^T = 0$ est la seule condition initiale dans Ω pour laquelle $[\tilde{q} \ \dot{\tilde{q}}]^T \in \Omega$ pour tout $t \geq 0$. Par conséquent, selon le théorème de LaSalle (Voir Annexe A), cela suffit pour garantir la stabilité asymptotique globale de l'origine $[\tilde{q}^T \ \dot{\tilde{q}}^T]^T = 0$.

En conséquence, nous avons :

$$\begin{aligned} \lim_{t \rightarrow \infty} \tilde{q}(t) &= 0 \\ \lim_{t \rightarrow \infty} \dot{q}(t) &= 0 \end{aligned} \quad (2.20)$$

2.3. Lois de commande dans le cas du problème de poursuite de trajectoire

Le problème de poursuite dans l'espace articulaire consiste à suivre une trajectoire prédéfinie variable dans le temps $q_d(t)$ et ses dérivées successives $\dot{q}_d(t)$ et $\ddot{q}_d(t)$ qui décrivent respectivement la vitesse et l'accélération désirées.

2.3.1. Générateur de trajectoire

Afin d'étudier les lois de commandes des robots manipulateurs il est nécessaire d'étudier les différentes trajectoires de référence utilisées en robotique, ce qui nous permet une meilleure analyse de la validité d'une loi de commande.

D'une manière générale une tâche de mouvement d'un robot manipulateur est spécifiée en définissant un chemin le long duquel le robot doit se déplacer. Ce chemin peut être défini par une courbe géométrique ou une séquence de points définis soit en espace opérationnel soit en espace articulaire. Le problème de la génération de trajectoire est de définir pour chaque articulation les trajectoires de référence en position, vitesse et accélération, ces trajectoires sont fonction du temps, assurant ainsi le passage du robot par une trajectoire désirée d'une configuration initiale à une configuration finale désirée [20].

Les trajectoires d'un robot peuvent être classées comme suit [20] :

- Trajectoire entre deux points avec trajectoire libre ;
- Trajectoire entre deux points via des points intermédiaires, spécifiés notamment pour éviter les obstacles, avec trajectoire libre ;
- Trajectoire entre deux points avec un chemin contraint entre eux ;
- Trajectoire entre deux points via des points intermédiaires avec trajectoire spécifiée entre les points intermédiaires.

Pour les deux premières classes, la trajectoire est généralement créée dans l'espace articulaire. Pour les deux dernières classes, il est généralement préférable de générer la trajectoire dans l'espace opérationnel [20].

Considérons un robot à n degré de liberté où :

Le vecteur des positions initiales est représenté par $q^{init} = [q_1^{init} \dots q_n^{init}]^T$, et le vecteur des positions finales par $q^{fin} = [q_1^{fin} \dots q_n^{fin}]^T$.

La trajectoire pour passer de q^{init} à q^{fin} est décrite par l'équation suivante [20] :

$$q_d(t) = q^{init} + r(t) D \quad \text{pour } 0 \leq t \leq t_{fin} \quad (2.21)$$

Avec $D = q^{fin} - q^{init}$, $r(t)$ est le polynôme d'interpolation et t_{fin} est la durée du mouvement.

Les valeurs aux limites pour $r(t)$ sont données par :

$$\begin{cases} r(0) = 0 \\ r(t_{fin}) = 1 \end{cases} \quad (2.22)$$

Il existe plusieurs fonctions permettant de satisfaire le passage de q^{init} à $t = 0$ vers q^{fin} à $t = t_{fin}$, comme l'interpolation polynomiale et l'interpolation bang-bang.

2.3.1.1. Interpolation polynomiale [6]

Il existe plusieurs modes d'interpolation polynomiale parmi lesquels l'interpolation linéaire, l'interpolation par des polynômes de degrés trois et de degré cinq.

A. Interpolation linéaire

Elle est considérée comme étant l'interpolation la plus simple, où la trajectoire de chaque articulation est décrite par une équation linéaire en temps. L'équation du mouvement s'écrit :

$$q_d(t) = q^{init} + r(t) D \quad \text{avec } r(t) = \frac{t}{t_{fin}} \quad (2.23)$$

Nous aurons alors :

$$\dot{q}_d(t) = \dot{r}(t) D \quad \text{avec } \dot{r}(t) = \frac{1}{t_{fin}} \quad (2.24)$$

Et

$$\ddot{q}_d(t) = \ddot{r}(t) D \quad \text{avec } \ddot{r}(t) = 0 \quad (2.25)$$

Dans ce type d'interpolation les trajectoires sont continues en position mais discontinues en vitesse tel qu'il est montré sur la figure 2.3 pour une articulation quelconque j . en réalité ce type de mouvement est inacceptable pour les robots à cause de ces discontinuités.

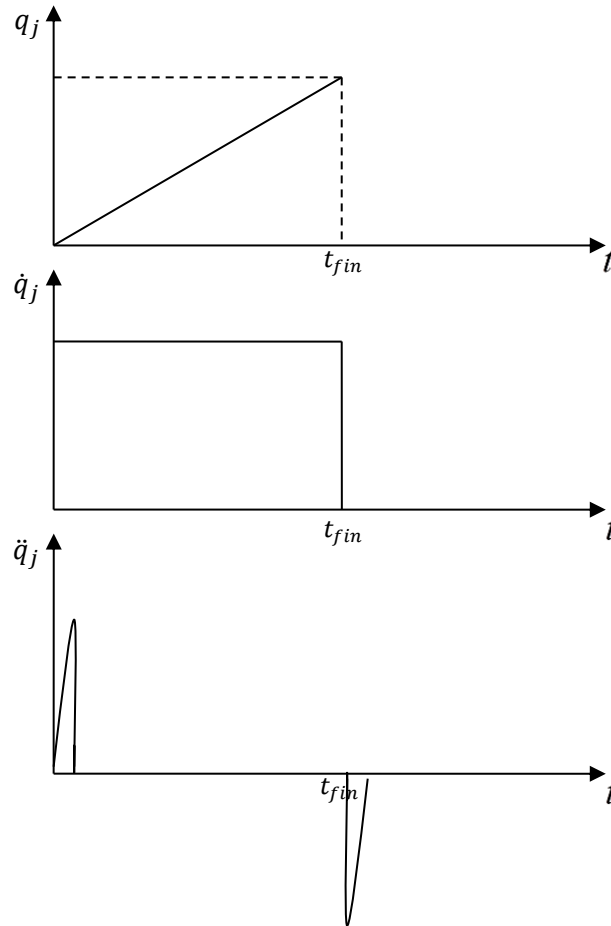


Figure 2.3. Position, vitesse et accélération pour une interpolation polynômiale linéaire

B. Interpolation polynomiale de degré 3

Si aux points de départ et d'arrivée on impose une vitesse nulle, on ajoute deux contraintes aux deux contraintes de position. Le degré minimal du polynôme qui satisfait ces quatre contraintes est de degré trois et est représenté par la formule générale suivante :

$$q_d(t) = q^{init} + \left[3 \left(\frac{t}{t_{fin}} \right)^2 - 2 \left(\frac{t}{t_{fin}} \right)^3 \right] D \quad (2.26)$$

Ainsi nous aurons :

$$\dot{q}_d(t) = \left[6 \left(\frac{t}{t_{fin}} \right) - 6 \left(\frac{t^2}{t_{fin}^2} \right) \right] D \quad (2.27)$$

Et l'accélération est exprimée par :

$$\ddot{q}_d(t) = \left[\left(\frac{6}{t_{fin}^2} \right) - 12 \left(\frac{t}{t_{fin}^3} \right) \right] D \quad (2.28)$$

L'évolution des positions vitesses et accélérations pour n'importe quelle articulation j est montrée sur la figure 2.4. Ce type d'interpolation assure la continuité des trajectoires en positions et vitesses, mais pas celle des accélérations.

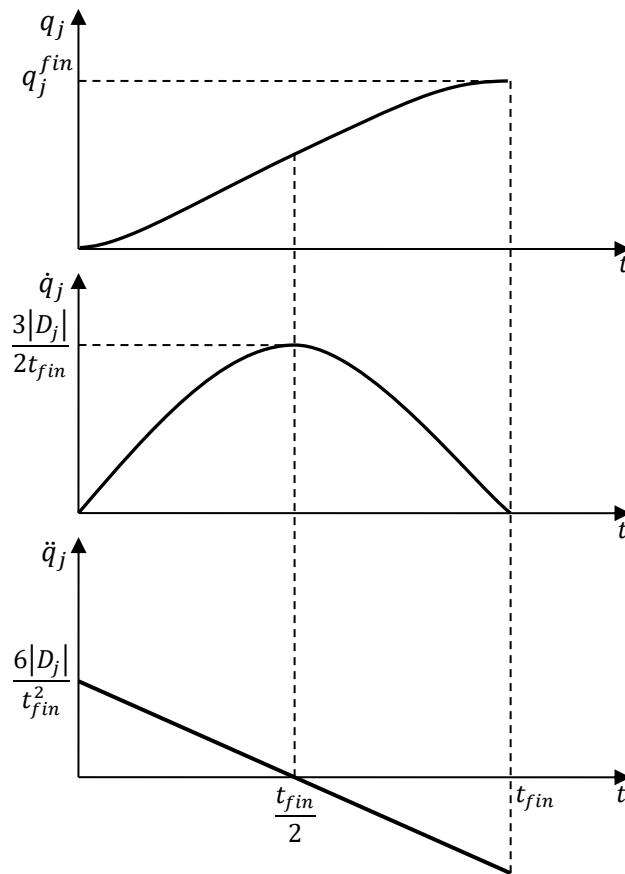


Figure 2.4. Position, vitesse et accélération pour une interpolation polynômiale de degré 3

La vitesse atteint sa valeur maximale lorsque $t = \frac{t_{fin}}{2}$, Elle vaut donc :

$$\dot{q}_{j \max} = \frac{3|D_j|}{2t_{fin}} \quad \text{avec } |D_j| = |q_j^{fin} - q_j^{init}| \quad (2.29)$$

La valeur maximale de l'accélération est atteinte à $t = 0$ et $t = t_{fin}$, elle a pour valeur:

$$\ddot{q}_{j \max} = \frac{6|D_j|}{t_{fin}^2} \quad (2.30)$$

C. Interpolation polynômiale de degré 5

Pour les robots manipulateurs transportant des charges importantes ou les robots à grande vitesse, il est nécessaire de garantir la continuité des accélérations afin d'éviter l'excitation de la mécanique.

L'interpolation polynômiale de degré cinq assure des trajectoires continues en positions, vitesses et accélérations. Le polynôme est obtenu en employant les conditions aux limites suivantes :

$$q(0) = q^{init}, q(t_{fin}) = q^{fin}, \dot{q}(0) = 0, \dot{q}(t_{fin}) = 0, \ddot{q}(0) = 0, \ddot{q}(t_{fin}) = 0 \quad (2.31)$$

Pour satisfaire les six contraintes, le polynôme d'interpolation doit être de degré cinq :

$$q_d(t) = q^{init} + \left[10 \left(\frac{t}{t_{fin}} \right)^3 - 15 \left(\frac{t}{t_{fin}} \right)^4 + 6 \left(\frac{t}{t_{fin}} \right)^5 \right] D \quad (2.32)$$

Les trajectoires des positions, vitesses et accélérations pour l'articulation j sont représentées à la figure 2.5. Les valeurs maximales des vitesses et des accélérations ont pour expressions :

$$\dot{q}_j \max = \frac{15|D_j|}{8t_{fin}} \quad (2.33)$$

$$\ddot{q}_j \max = \frac{10|D_j|}{\sqrt{3} t_{fin}^2} \quad (2.34)$$

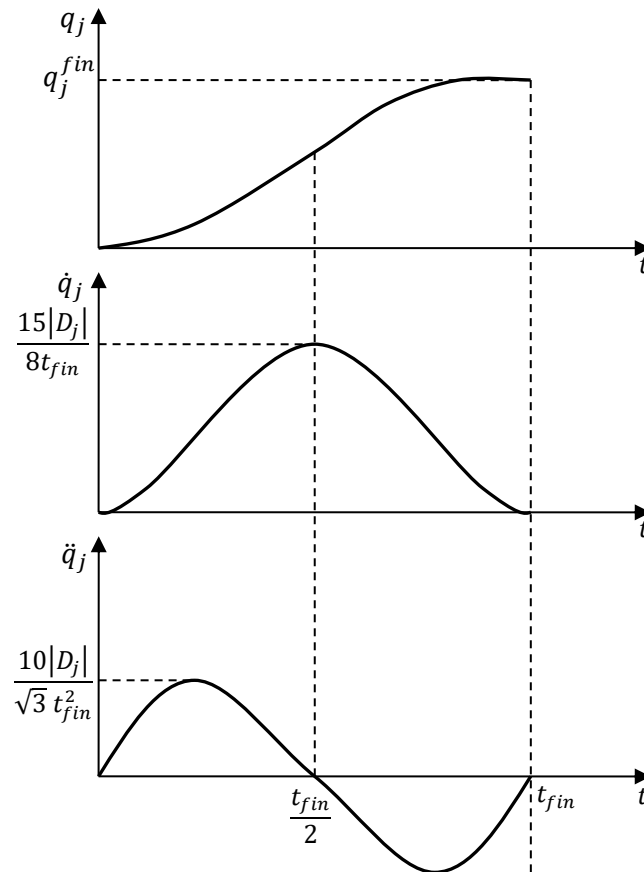


Figure 2.5. Position, vitesse et accélération pour une interpolation polynômiale de degré 5

2.3.1.2. Interpolation Bang-Bang [6]

La trajectoire est représenté par une phase d'accélération jusqu'à $\frac{t_{fin}}{2}$ et d'une phase de décélération de $\frac{t_{fin}}{2}$ à t_{fin} . L'équation de mouvement dans ce cas est représentée par la formule suivante :

$$\begin{cases} q_d(t) = q^{init} + 2 \left(\frac{t}{t_{fin}} \right)^2 D & \text{pour } 0 \leq t \leq \frac{t_{fin}}{2} \\ q_d(t) = q^{init} + \left[-1 + 4 \left(\frac{t}{t_{fin}} \right) - 2 \left(\frac{t}{t_{fin}} \right)^2 \right] D & \text{pour } \frac{t_{fin}}{2} \leq t \leq t_{fin} \end{cases} \quad (2.35)$$

Les vitesses initiale et finale sont nulles. Il y a une continuité dans le mouvement de la position et de la vitesse. L'allure des positions, vitesses et accélérations est montré dans la figure 2.6.

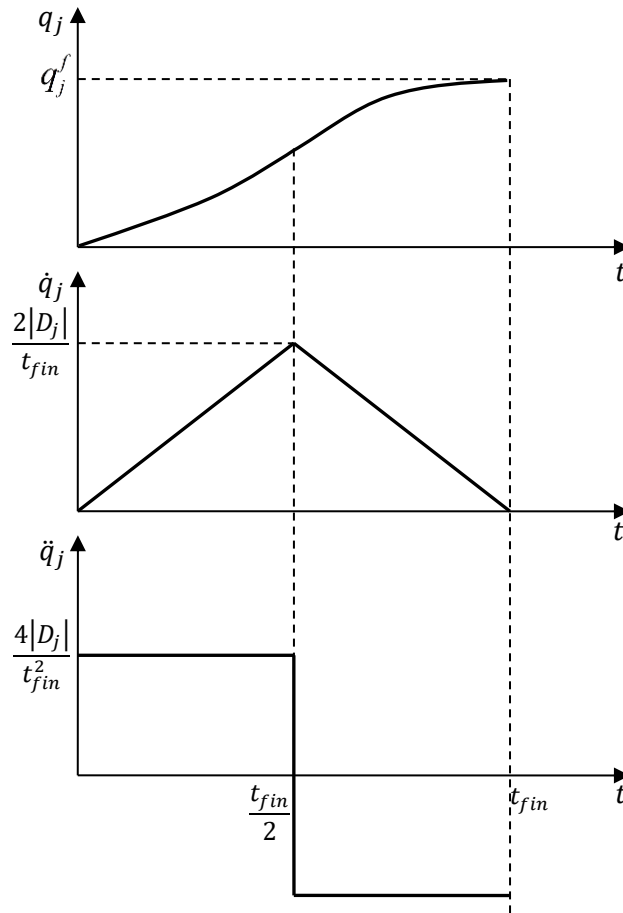


Figure 2.6. Position, vitesse et accélération pour une interpolation Bang-Bang

Les valeurs maximales des vitesses et des accélérations sont données par les deux expressions suivantes :

$$\dot{q}_{j \max} = \frac{2|D_j|}{t_{fin}} \quad (2.36)$$

$$\ddot{q}_{j \max} = \frac{4|D_j|}{t_{fin}^2} \quad (2.37)$$

2.3.2. La loi de commande du couple calculé (computed torque control) [39]

La loi de commande du couple calculé ressemble à la classe des lois de commande par découplage non linéaire [6]. Elle consiste à poursuivre une trajectoire prédéfinie par $q_d(t)$, $\dot{q}_d(t)$ et $\ddot{q}_d(t)$ reposant sur l'idée d'annuler les termes non linéaire, permettant ainsi l'obtention en boucle fermée d'un comportement linéaire et découplé. Ce type de loi de commande sera proche de la loi de commande par découplage non linéaire mais avec un objectif différent.

En se basant sur le modèle dynamique donné par l'équation (2.1), la loi de commande du couple calculé est exprimée par la relation suivante [39] :

$$\tau = M(q)[\ddot{q}_d + k_p \tilde{q} + k_v \dot{\tilde{q}}] + C(q, \dot{q})\dot{q} + g(q) \quad (2.38)$$

Où k_p et k_v sont des matrices symétriques et définies positives ; et $\tilde{q} = q_d - q$ qui représente l'erreur de poursuite en position. Ainsi, la dynamique en boucle fermée du robot est obtenue en remplaçant le couple du modèle dynamique (2.1) par la loi de commande (2.38) et sachant que la matrice $M(q)$ est définie positive et inversible selon la propriété 2 du modèle dynamique des robots manipulateurs citée dans le premier chapitre :

$$\ddot{\tilde{q}}_d + k_p \tilde{q} + k_v \dot{\tilde{q}} = 0 \quad (2.39)$$

Le choix d'une matrice diagonale pour k_p et k_v nous permet d'assurer un découplage et ainsi le système est qualifié de linéaire découplé par feedback. Cette loi de commande est représentée par le schéma bloc donné à la figure 2.7.

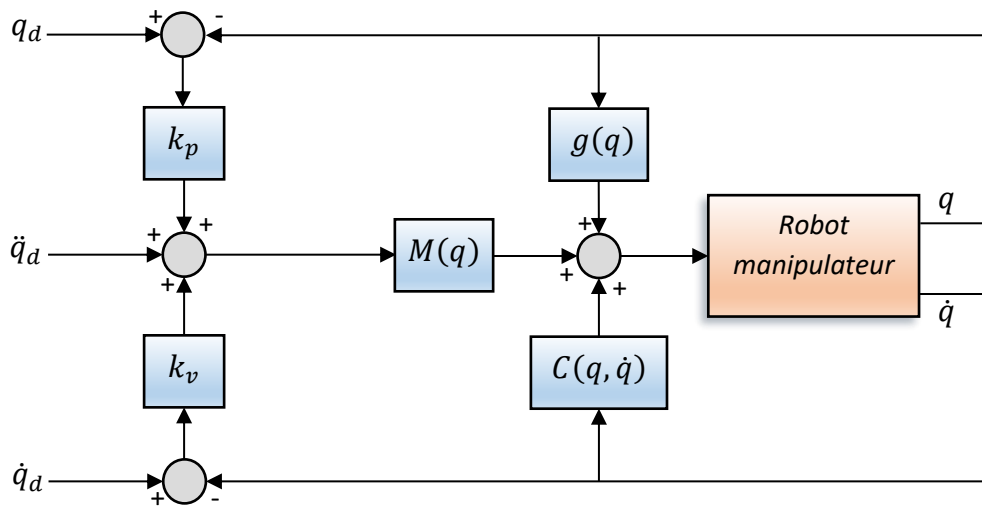


Figure 2.7. Schéma fonctionnel de la commande du couple calculé

L'équation d'erreur (2.39) peut être exprimée sous forme de représentation d'état ayant comme vecteur d'état $[\tilde{q}^T \ \dot{\tilde{q}}^T]^T$:

$$\frac{d}{dt} \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -k_p & -k_v \end{bmatrix} \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix} = A_{ct} \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix} \quad (2.40)$$

Où I est la matrice identité de dimension n .

Sachant que les matrices k_p et k_v sont définies positives, la matrice A_{ct} est donc stable. Il est important de noter que l'équation (2.40) est une équation différentielle linéaire autonome, et son unique point d'équilibre est $[\tilde{q}^T \ \dot{\tilde{q}}^T]^T = 0$. L'unicité du point d'équilibre provient du fait que la matrice k_p est conçue pour être définie positive et donc non singulière.

Afin de démontrer que le point d'équilibre $[\tilde{q}^T \ \dot{\tilde{q}}^T]^T = 0$ est stable, nous allons utiliser la méthode directe de Lyapunov. Pour cela nous commençons par introduire une constante η satisfaisant :

$$\lambda_{\min}\{k_v\} > \eta > 0 \quad (2.41)$$

En multipliant (2.41) par $\xi^T \xi$ avec $\xi \in \mathbb{R}^n$ est un vecteur quelconque, nous aurons :

$$\lambda_{\min}\{k_v\}\xi^T\xi > \eta\xi^T\xi \quad (2.42)$$

Sachant que k_v est une matrice symétrique, alors $\xi^T k_v \xi \geq \lambda_{\min}\{k_v\}\xi^T\xi$, nous avons donc :

$$\xi^T[k_v - \eta I]\xi > 0 \quad \forall \xi \neq 0 \quad (2.43)$$

L'inégalité (2.43) implique que la matrice $k_v - \eta I$ est définie positive, *i.e.*

$$k_v - \eta I > 0 \quad (2.44)$$

En se basant sur le fait que la constante η et la matrice k_p sont définies positives, nous concluons que :

$$k_p + \eta k_v - \eta^2 I > 0 \quad (2.45)$$

Considérons la fonction de Lyapunov candidate suivante :

$$\begin{aligned} V(\tilde{q}, \dot{\tilde{q}}) &= \frac{1}{2} \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix}^T \begin{bmatrix} k_p + \eta k_v & \eta I \\ \eta I & I \end{bmatrix} \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix} \\ &= \frac{1}{2} [\dot{\tilde{q}} + \eta \tilde{q}]^T [\dot{\tilde{q}} + \eta \tilde{q}] + \frac{1}{2} \tilde{q}^T [k_p + \eta k_v - \eta^2 I] \tilde{q} \end{aligned} \quad (2.46)$$

Où la constante η satisfait les deux inégalités (2.44) et (2.45). À partir de cela, il est clair que la fonction $V(\tilde{q}, \dot{\tilde{q}})$ est globalement définie positive. Elle peut être réécrite aussi de la forme suivante :

$$V(\tilde{q}, \dot{\tilde{q}}) = \frac{1}{2} \dot{\tilde{q}}^T \dot{\tilde{q}} + \frac{1}{2} \tilde{q}^T [k_p + \eta k_v] \tilde{q} + \eta \tilde{q}^T \dot{\tilde{q}} \quad (2.47)$$

En dérivant cette fonction nous aurons :

$$\dot{V}(\tilde{q}, \dot{\tilde{q}}) = \ddot{\tilde{q}}^T \dot{\tilde{q}} + \dot{\tilde{q}}^T [k_p + \eta k_v] \dot{\tilde{q}} + \eta \dot{\tilde{q}}^T \dot{\tilde{q}} + \eta \tilde{q}^T \ddot{\tilde{q}} \quad (2.48)$$

En remplaçant $\ddot{\tilde{q}}$ de l'équation d'erreur en boucle fermée (2.40) dans l'expression (2.48) et en faisant quelques simplifications, nous obtenons :

$$\begin{aligned} \dot{V}(\tilde{q}, \dot{\tilde{q}}) &= -\dot{\tilde{q}}^T [k_v - \eta I] \dot{\tilde{q}} - \eta \tilde{q}^T k_p \tilde{q} \\ &= - \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix}^T \begin{bmatrix} \eta k_p & 0 \\ 0 & k_v - \eta I \end{bmatrix} \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix} \end{aligned} \quad (2.49)$$

Si η est choisi tel que la condition (2.44) est vérifiée, et la matrice k_p est définie positive, alors la fonction $\dot{V}(\tilde{q}, \dot{\tilde{q}})$ dans (2.49) est globalement définie négative. Suivant le théorème A.1 (voir Annexe A), le point d'équilibre $[\tilde{q}^T \ \dot{\tilde{q}}^T]^T = 0$ de l'équation en boucle fermée est globalement asymptotiquement stable, donc :

$$\begin{aligned} \lim_{t \rightarrow \infty} \tilde{q}(t) &= 0 \\ \lim_{t \rightarrow \infty} \dot{\tilde{q}}(t) &= 0 \end{aligned} \quad (2.50)$$

2.4. Résultats de simulation des lois de commande classique

Afin de tester l'efficacité des lois de commande citées tout au long de ce chapitre, plusieurs simulations ont été effectuées avec le modèle dynamique du robot décrit par l'équation (2.1) avec trois types de lois de commande, la loi de commande PID, la loi de commande point à

point et la loi de commande du couple calculé. Nous avons deux objectifs différents : l'objectif de régulation pour les deux premières et de poursuite de trajectoire pour la troisième. Nous utilisons le modèle du robot manipulateur à deux degrés de liberté.

2.4.1. Bras manipulateur à 2ddl

Le système contrôlé dans cette étude est un bras manipulateur plan à deux degrés de liberté (voir figure 2.8), dont le modèle de l'équation (2.1) est donné par les trois matrices $M(q)$, $C(q, \dot{q})$ et $g(q)$ (voir section B.1 dans l'Annexe B)

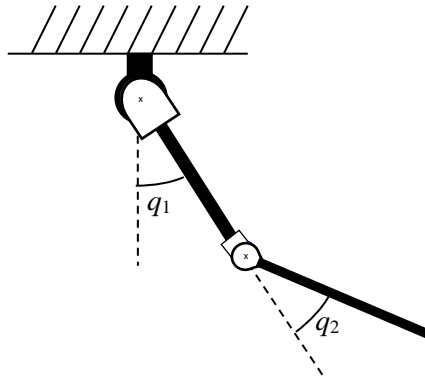


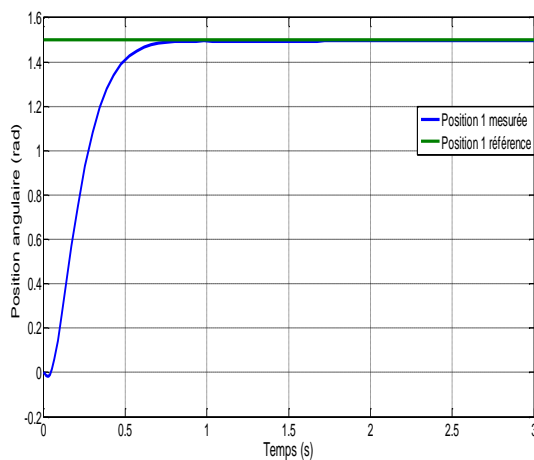
Figure 2.8 : Robot manipulateur à deux degrés de liberté

2.4.1.1. Application de la loi de commande PID

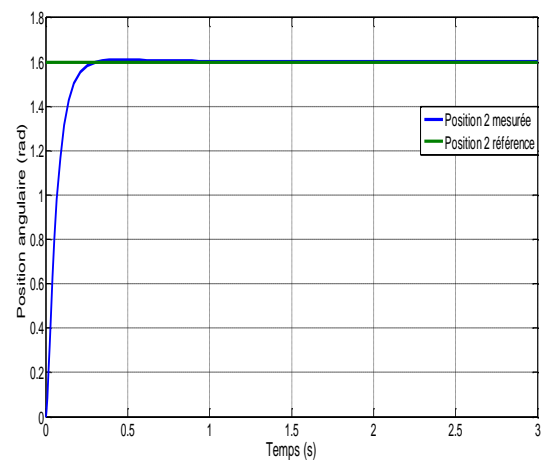
Les figures (2.9.A) au (2.9.F) présentent les résultats de simulation, pour des positions désirées $q_1=1.5 \text{ rad}$ et $q_2=1.6 \text{ rad}$, les matrices des paramètres k_p , k_d et k_i sont données par :

$$k_p = \begin{bmatrix} 500 & 0 \\ 0 & 800 \end{bmatrix}, k_d = \begin{bmatrix} 120 & 0 \\ 0 & 50 \end{bmatrix}, k_i = \begin{bmatrix} 200 & 0 \\ 0 & 30 \end{bmatrix}$$

Nous remarquons d'après les figures 2.9.A et 2.9.B que les deux articulations 1 et 2 atteignent les valeurs désirées 1.5 rad et 1.6 rad respectivement. Nous remarquons aussi que cette commande permet de résoudre le problème de régulation avec des erreurs de position nulles pour les deux articulations, ce qui montré sur les figures 2.9.E et 2.9.F. Le temps de réponse est de 0.6 s pour la première articulation et 0.25 s pour la deuxième articulation.



(A)



(B)

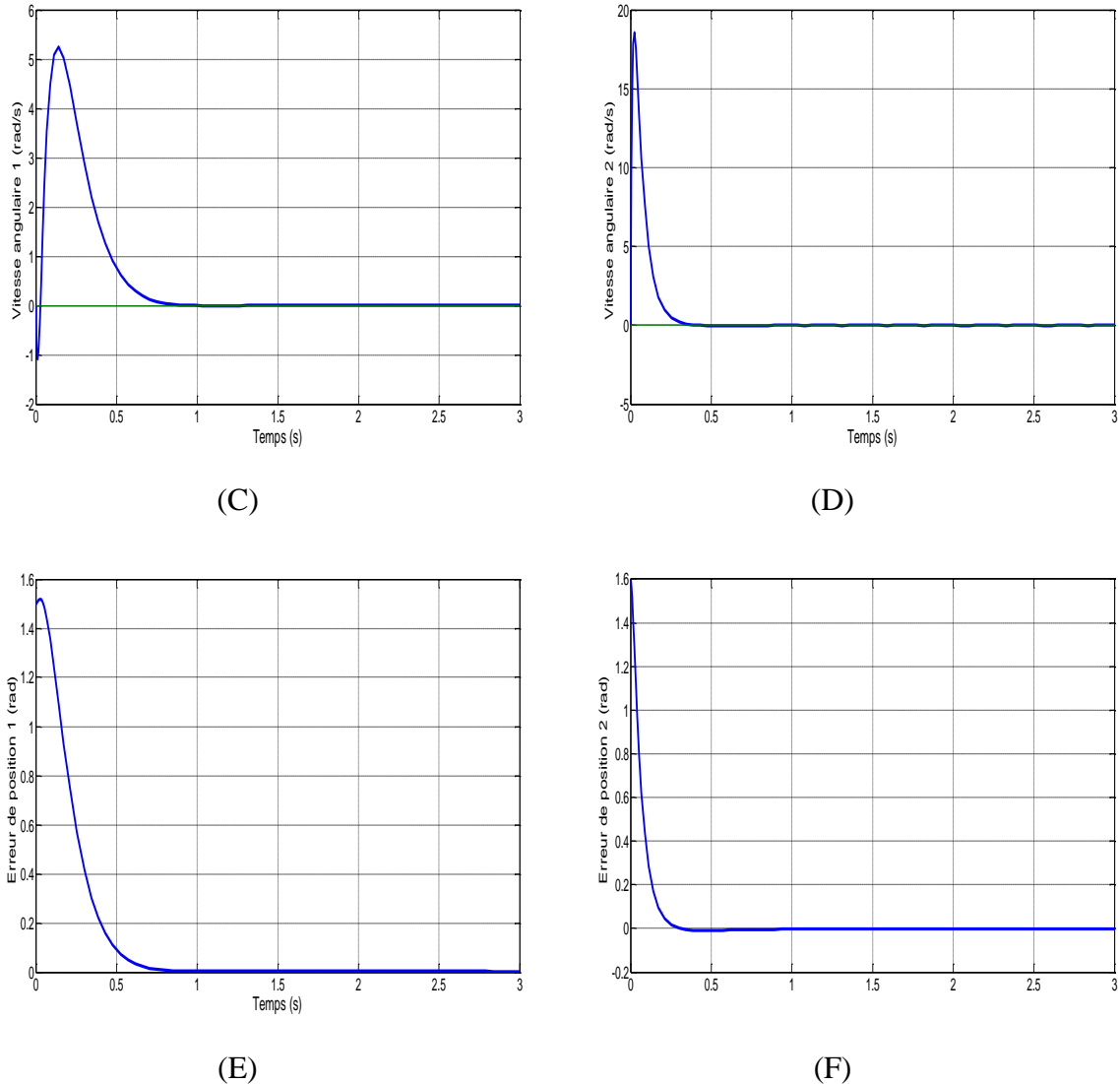


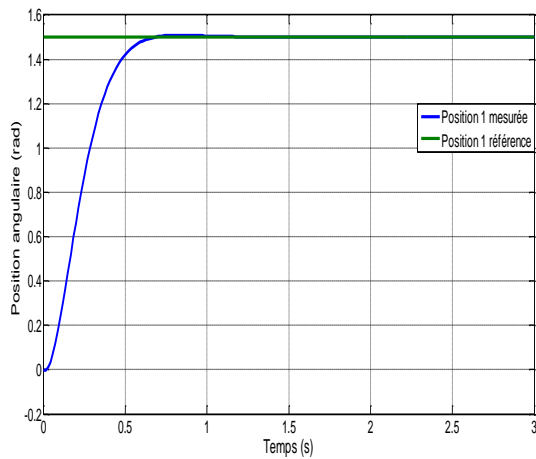
Figure 2.9. Réponses avec la commande PID : (A) et (B) Allures des positions angulaires, (C) et (D) Allures des vitesses angulaires, (E) et (F) Allures des erreurs des positions angulaires.

2.4.1.2. Application de la loi de commande point à point

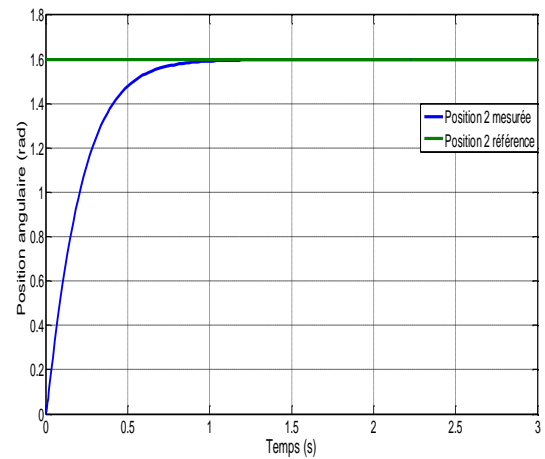
Les figures (2.10.A) au (2.10.F) présentent les résultats de simulation, pour des positions désirées $q_1=1.5 \text{ rad}$ et $q_2=1.6 \text{ rad}$, les matrices des paramètres k_p et k_d sont données par :

$$k_p = \begin{bmatrix} 500 & 0 \\ 0 & 500 \end{bmatrix}, k_d = \begin{bmatrix} 120 & 0 \\ 0 & 100 \end{bmatrix}$$

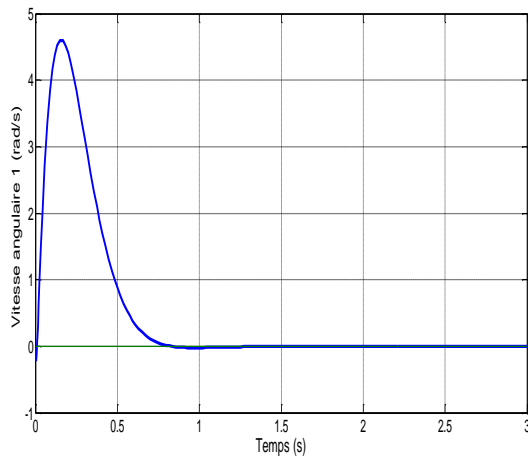
Nous remarquons d'après les figures 2.10.A et 2.10.B que les deux articulations 1 et 2 atteignent les valeurs désirées 1.5 rad et 1.6 rad respectivement. Nous remarquons aussi que le temps de réponse des deux articulations diffère, puisque la première articulation atteint la valeur désirée plus rapidement que la deuxième. Les erreurs sur les positions 1 et 2 ont une convergence exponentielle, telle qu'il est montré dans les figures 2.10.E et 2.10.F. Cette commande assure une erreur de position nulle avec des gains relativement petits, ceci est dû à la compensation de l'effet de gravité.



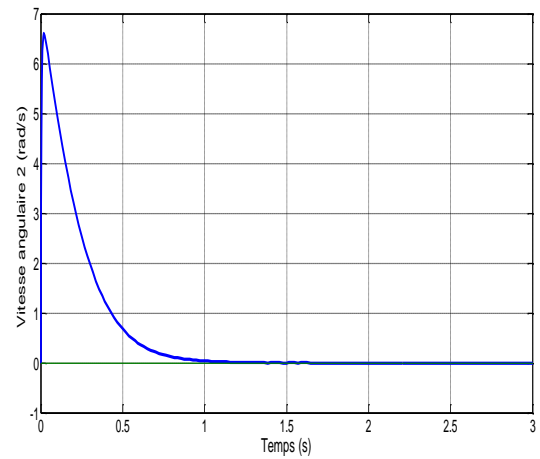
(A)



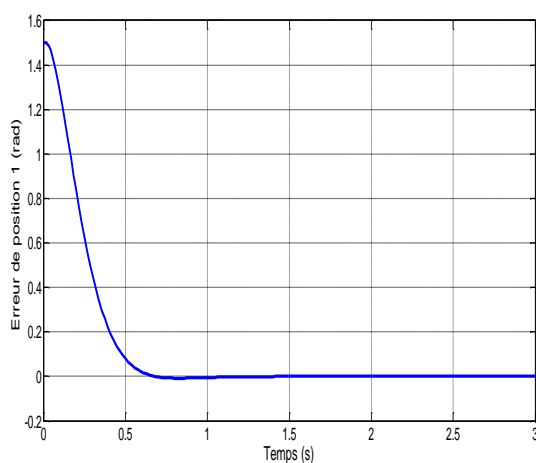
(B)



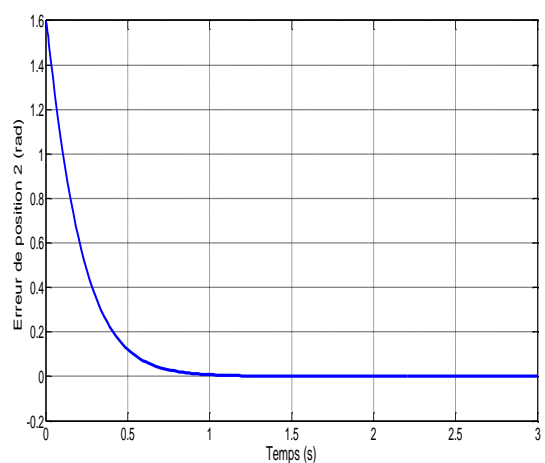
(C)



(D)



(E)



(F)

Figure 2.10. Réponses avec la commande point à point : (A) et (B) Allures des positions angulaires, (C) et (D) Allures des vitesses angulaires, (E) et (F) Allures des erreurs des positions angulaires.

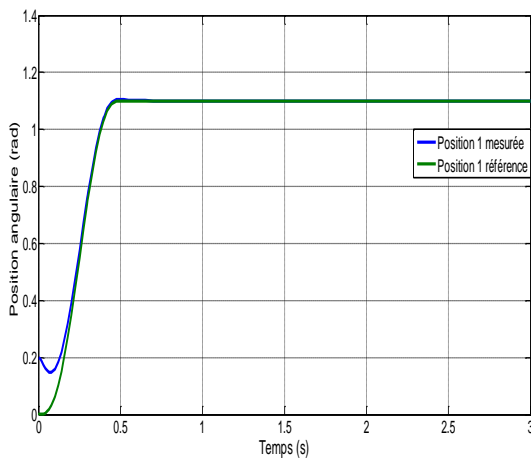
2.4.1.3. Application de la loi de commande du couple calculé

Le comportement de notre système après l'application de cette loi de commande, est illustré par les figures (2.11.A) au (2.11.F) pour les matrices des gains k_p et k_v suivantes :

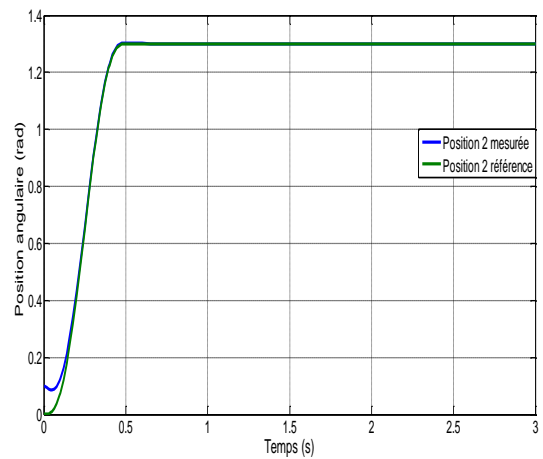
$$k_p = \begin{bmatrix} 650 & 0 \\ 0 & 600 \end{bmatrix}, k_v = \begin{bmatrix} 85 & 0 \\ 0 & 80 \end{bmatrix}$$

La génération de trajectoire a été réalisée par une interpolation polynomiale de degré cinq. Nous avons procédé à un changement des conditions initiales pour les deux articulations, afin de mieux vérifier si les deux articulations arrivent à suivre leurs trajectoires de référence. Les conditions initiales sont $q_{1\text{init}}=0.1 \text{ rad}$ et $q_{2\text{init}}=-0.1 \text{ rad}$.

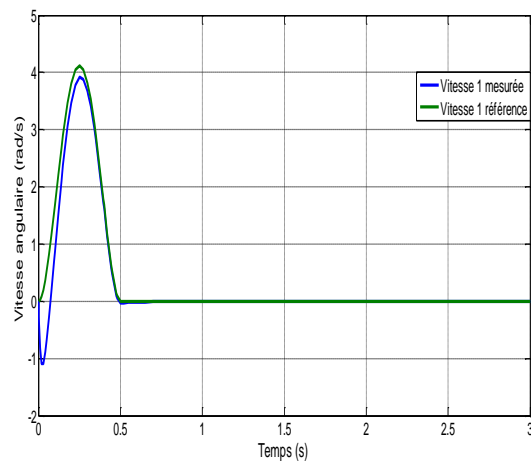
Pour ce type de commande en poursuite de trajectoire, les résultats de simulation montrent qu'il y a une bonne poursuite des trajectoires, ce qui garantit que les erreurs de poursuite convergent vers zéro. Le choix des conditions initiales non nulles nous a permis de confirmer la bonne poursuite des deux positions de leurs trajectoires de références. La commande du couple calculée permet une linéarisation et un découplage parfait du système ce qui est vérifié à travers les équations (2.39) et (2.40), en le réduisant à un système composé de n équations différentielles de deuxième ordre linéaires et indépendantes.



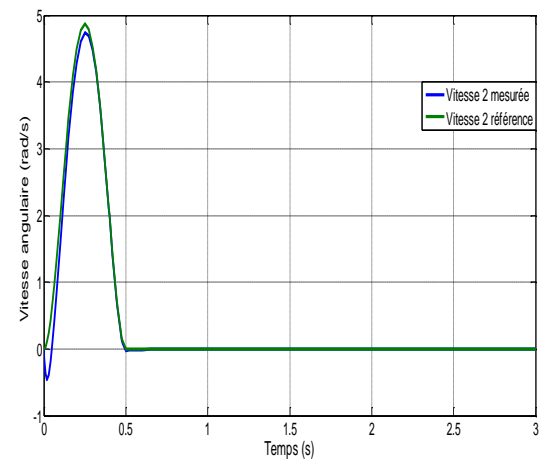
(A)



(B)



(C)



(D)

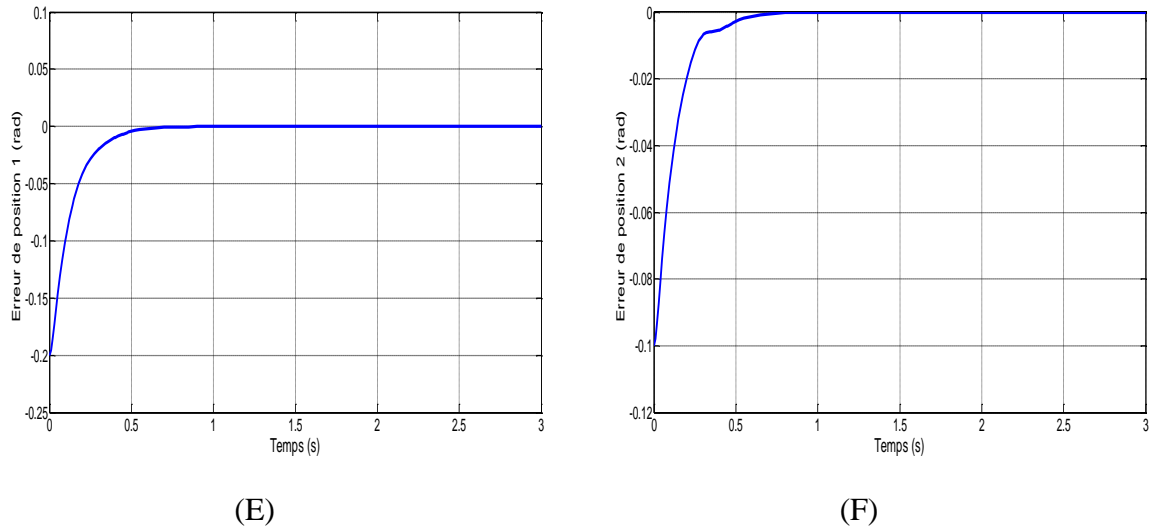


Figure 2.11. Réponses avec la commande du couple calculé: (A) et (B) Allures des positions angulaires, (C) et (D) Allures des vitesses angulaires, (E) et (F) Allures des erreurs des positions angulaires.

2.5. Conclusion

A travers ce chapitre nous avons présenté quelques lois de commandes dynamiques classiques les plus utilisées pour contrôler les robots manipulateurs rigides. Ayant comme objectif de traiter le problème de régulation, la commande PID et la commande PD avec compensation de l'effet de gravité ont été exhibées en premier, par la suite une commande traitant l'objectif de poursuite de trajectoire a été présentée à savoir la commande du couple calculé. En poursuite de trajectoire plusieurs méthodes de génération de trajectoire communément utilisées en robotique existent. Parmi ces dernières nous avons présenté celles qui traitent le mouvement entre deux points, notamment les modes d'interpolation. La loi d'interpolation du polynôme de degré cinq assure une continuité de la position, la vitesse et aussi de l'accélération. Afin de valider notre étude théorique, nous avons appliqué les lois de commande étudiées au modèle dynamique d'un robot manipulateur à 2 ddl.

CHAPITRE 3

CONCEPTS THÉORIQUES SUR LES COMMANDES ROBUSTES PAR MODES GLISSANTS ET BACKSTEPPING

3.1. Introduction

La commande des robots manipulateurs a suscité un intérêt grandissant en raison de la large gamme de leurs applications, et de leurs natures non linéaires, couplés et multi-entrées multi-sorties (MIMO). Le principal objectif des systèmes de commande en robotique est de suivre une consigne ou une trajectoire de référence, ce qui implique la génération d'un signal de commande permettant d'éliminer l'erreur entre la position du robot et la référence. La conception des lois de commande, exploitées dans le domaine linéaire pour les systèmes non linéaires est souvent caduque et ne tient pas compte de certains phénomènes. Plusieurs techniques ont été développées ces dernières décennies pour la commande des systèmes non linéaires, parmi elles la théorie de commande non linéaire qui a connu un développement intense. Les techniques de commande non linéaire sont plus générales car elles peuvent être utilisées dans le cas des systèmes linéaire et non linéaires. Ces contrôleurs ont la capacité de résoudre différents problèmes tels que : l'invariance aux incertitudes du système et la résistance aux perturbations externes. Les commandes non linéaires les plus courantes qui ont été proposées pour résoudre les problèmes de contrôle sont la linéarisation par retour d'état ou linéarisation par retour de sortie, la commande adaptative, la commande basée sur l'intelligence artificielle, la technique de commande à structure variable dite par modes glissants [43] [1] [44] et la commande récursive basée sur la fonction de Lyapunov par backstepping [3] [45]. Ces deux derniers sont considérés parmi les meilleurs contrôleurs robustes non linéaires, et ont été analysés par de nombreux chercheurs en particulier pour les systèmes non linéaires ; leurs concepts théoriques vont faire l'objet de ce chapitre.

3.2. Commande par modes glissants

Les premiers travaux sur la théorie des systèmes à structure variable ont été lancés dans l'ex-Union soviétique à la fin des années cinquante et début des années soixante, avec les travaux de Tzypkin [46], d'Anosov [47], de Fillipov [48] et d'Emelyanov [49] [50]. Ces travaux de recherche ont reçu beaucoup plus d'attention à la fin des années soixante-dix, avec les travaux

d'Itkis [51] et d'Utkin [52], ce dernier qui introduisit la théorie des modes glissants. Cette technique connaît ces dernières années une grande plage d'applications dans une variété de domaines, pas seulement autant que technique de commande mais aussi pour l'observation [43] [53] [54] [55] [56].

La commande par modes glissants est considérée comme une commande à structure variable, dont sa structure change d'une façon discontinue. Dans [57] trois configurations de base des systèmes de réglage à structure variable peuvent être distinguées :

- Structure par commutation d'une contre-réaction d'état variable ;
- Structure par commutation au niveau de l'organe de commande ;
- Structure par commutation au niveau de l'organe de commande avec ajout de la commande équivalente.

Pour les trois structures une commutation entre deux grandeurs est définie par une loi de commutation. Cette commutation se fait sous certaines conditions à une fréquence très élevée (théoriquement infinie).

Nous allons présenter dans la partie suivante une description générale de la commande par modes glissants.

3.2.1. Concepts de base de la commande par modes glissants

Une loi de commande est généralement construite à base d'un modèle mathématique souvent donnant une description proche de la réalité physique. Cette loi de commande devra être donc robuste pour garantir une faible sensibilité aux variations et incertitudes paramétriques, aux erreurs de modélisation et aux perturbations. La commande par modes glissants est une technique de commande non linéaire et robuste qui modifie la dynamique d'un système non linéaire par les multiples structures de commande qui sont conçues de manière à garantir que les trajectoires se déplacent toujours vers une condition de commutation. La loi de commande par retour d'état n'est donc pas une fonction continue du temps. Au lieu de cela, elle passe d'une structure continue à une autre en fonction de la position actuelle dans l'espace d'états. Par conséquent, la commande par modes glissants est une technique de commande à structure variable. D'une manière générale les trajectoires ultimes vont glisser le long des limites des structures de commande. Le mouvement du système qui glisse le long de ces limites est appelé un *mode glissant* et le lieu géométrique constitué des limites est nommé *la (hyper) surface de glissement* [57] [1].

3.2.1.1. Définitions [58] [59]

Considérons le système non linéaire suivant :

$$\begin{cases} \dot{\xi} = f(\xi, t, u) \\ y = g(\xi) \end{cases} \quad \text{avec} \quad \xi(0) = \xi_0 \quad (3.1)$$

Ainsi que l'hypersurface : $s(\xi) = 0$.

Où f est un champ de vecteur suffisamment différentiable.

$$u(\xi, t) = \begin{cases} u^+ & \text{si } s(\xi) > 0 \\ u^- & \text{si } s(\xi) < 0 \end{cases} \quad (3.2)$$

$$u(\xi, t) \in [-1, 1] \quad \text{si } s(\xi) = 0$$

$u \in \Omega \subset R$: représente la commande du système.

Où ξ est l'état du système de dimension n . Le système décrit par l'équation (3.1) avec la loi de commande donnée par (3.2) est à structure variable [52]. Le choix d'une commande discontinue u rend le système discontinu, il est dit alors à discontinuité artificielle.

Lorsque le système (3.1) est piloté par la loi de commande (3.2), il peut être représenté par l'écriture suivante :

$$\dot{\xi} = f(\xi, t, u) = \begin{cases} f^+(\xi, t) & \text{si } s(\xi, t) > 0 \\ f^-(\xi, t) & \text{si } s(\xi, t) < 0 \end{cases} \quad (3.3)$$

Où $f^+(\xi, t)$ et $f^-(\xi, t)$ représentent des champs de vecteurs complets dans R^n .

$s(\xi, t)$: est une surface dans R^n qui divise l'espace en deux parties disjointes $s(\xi, t) > 0$ et $s(\xi, t) < 0$ qu'on notera respectivement φ^+ et φ^- .

Il existe aussi des systèmes qui sont naturellement conçus discontinus, comme les systèmes mécaniques possédant un frottement sec et les circuits électroniques avec commutateurs [52] [60]. Tout système à structure variable décrit par (3.1) peut être ramené à la forme (3.3) quel que soit le type de sa discontinuité naturelle ou artificielle.

Les vecteurs vitesse f^+ et f^- se comportent de différentes manières à l'extérieur de la surface de discontinuité comme il est montré par la figure 3.1.

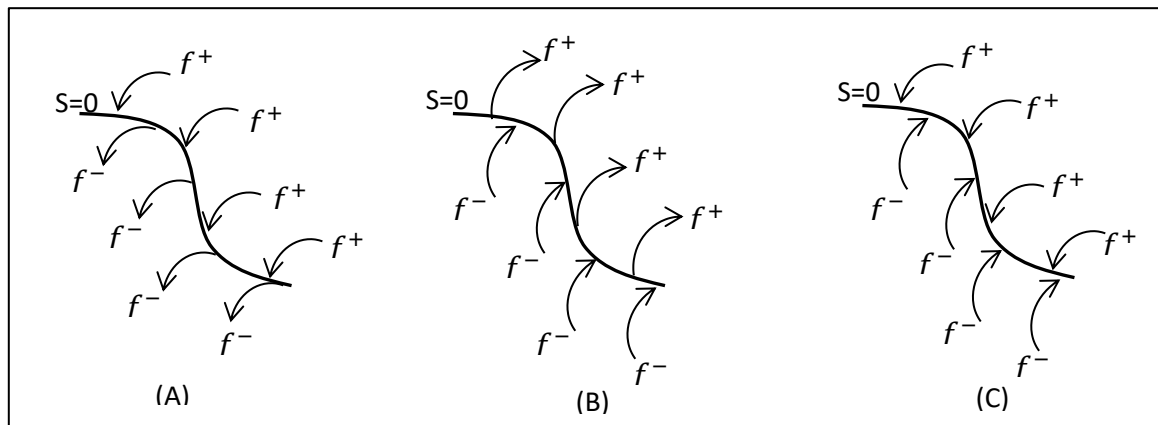


Figure 3.1. Différents comportements en dehors de la surface de discontinuité

- Le premier cas est représenté par Figure 3.1.A et Figure 3.1.B où les vecteurs vitesse f^+ et f^- traversent la surface d'un côté vers un autre.
- Le deuxième cas, celui qui nous intéresse où les deux vecteurs f^+ et f^- sont pointés chacun vers la surface, tel qu'il est montré dans la figure 3.1.C. dans ce cas la surface est dite attractive.

Définition 3.1.

Pour un domaine de convergence donné, la surface $S=0$ est dite attractive si toute trajectoire évoluant dans le domaine d'attraction est dirigée vers cette surface.

Définition 3.2.

La surface $S=0$ est dite invariante, dans le cas où toutes les trajectoires qui débutent dans cette surface ou atteignent cette surface, ne pourront en sortir et évoluent que sur cette surface.

Si l'état du système est du côté φ^+ de l'espace d'état (ou du côté φ^-), il rejoindra forcément la surface $S=0$. S'il dépasse de l'autre côté φ^- (ou du côté φ^+), il se ramènera vers $S=0$ (figure 3.2). Cette surface $S=0$ est appelée donc surface glissante et le mouvement sur cette surface est un mode glissant dont l'équation détermine la dynamique désirée du système.

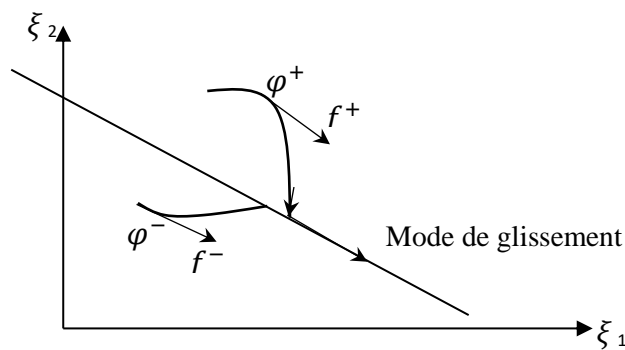


Figure 3.2. Mode de glissement idéal

Selon la position de l'état du système, s'il est sur φ^+ ou sur φ^- , il atteint la surface de discontinuité $S=0$, avec les vitesses f^+ ou f^- respectivement.

Les solutions obtenues à partir des équations différentielles à second membre continu f^+ et f^- donnent l'équation du mouvement, tel que :

$$\dot{\xi} = \begin{cases} f^+(\xi, t) & \text{si } \xi \in \varphi^+ \\ f^-(\xi, t) & \text{si } \xi \in \varphi^- \end{cases} \quad (3.4)$$

Par hypothèse, ces deux équations différentielles ordinaires admettent des solutions.

Généralement, d'une façon théorique l'étude des systèmes qui comportent des discontinuités ne se fait pas d'une manière classique. En effet, les conditions d'existence et d'unicité de la solution ne sont pas satisfaites pour les équations différentielles, et ceci à cause des termes discontinus non Lipchitziens, et non défini généralement sur la surface de discontinuité.

Définition 3.3.

Lorsqu'un système est soumis à une loi de commande qui le force à rester sur une surface donnée, son évolution dépendra alors uniquement des propriétés de cette surface, et non plus de lui-même ou des perturbations auxquelles il est soumis. La trajectoire dans un plan de phase est constituée de trois phases distinctes, comme il est montré dans la figure 3.3 pour un cas d'un système d'ordre deux [57] [1] [61] :

- **Le mode de convergence (MC)** : c'est un mode durant lequel la variable à régler converge vers la surface de commutation $S=0$, en partant de n'importe quel point initial dans le plan de phase en un temps fini. La loi de commande et le critère de convergence caractérise ce mode.
- **Le mode de glissement (MG)** : lorsque la variable d'état rejoint la surface de glissement et tend vers l'origine du plan de phase, on parle alors de mode de glissement. Le choix de la surface de glissement S caractérise la dynamique de ce mode.
- **Le mode de régime permanent (MRP)** : permettant l'étude autour du point d'équilibre (origine du plan de phase) la réponse du système. Ce mode caractérise la qualité et les performances de la commande.

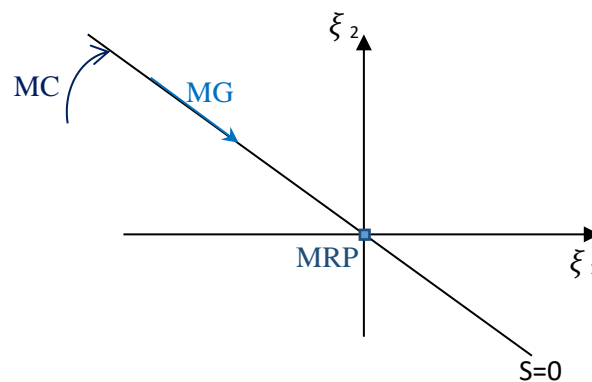


Figure 3.3. Différents modes des trajectoires d'état dans un plan de phase

3.2.1.2. Choix de la surface de glissement et degré relatif

Le choix de la surface de glissement dépend généralement de l'erreur de poursuite et du degré relatif du système à contrôler. Afin de montrer la relation entre la surface de glissement et le degré relatif, considérons un système non linéaire défini par :

$$\dot{\xi} = f(\xi, t) + g(\xi, t)u \quad (3.5)$$

Où : $\xi(t)$ est le vecteur d'état, $u(t)$ est le vecteur d'entrée, et $f(\xi, t)$ et $g(\xi, t)$ sont des fonctions non linéaires.

Soit $e(t) = y(t) - y_{ref}(t)$, l'erreur de poursuite du système, où $y(t)$ est la sortie du système et $y_{ref}(t)$ est le signal de référence.

Le degré relatif d'un système est défini par le nombre minimum de fois qu'il faut dériver la sortie $y(t)$ par rapport au temps pour faire apparaître l'entrée d'une manière explicite [62].

Théorème 3.1 [63]

Un régime glissant d'ordre un existe sur la surface S si et seulement si le système (3.5) est de degré relatif un par rapport à $s(\xi, t)$.

Le choix de la surface de glissement est alors fait selon la remarque suivante :

Remarque 3.1

Posons $s(z^0, \dots, z^{v-1}) = z^{v-1} + \lambda_{v-2}z^{v-2} + \dots + \lambda_1z^1 + \lambda_0z^0$ où le choix des coefficients λ_i est tel que l'équation caractéristique : $z^{v-1} + \lambda_{v-2}z^{v-2} + \dots + \lambda_1z + \lambda_0$ possède ses racines strictement dans le demi plan complexe gauche.

Alors $s(e, \dot{e}, \dots, e^{v-1}) = 0 \Rightarrow e^{v-1} + \lambda_{v-2}e^{v-2} + \dots + \lambda_1\dot{e} + \lambda_0 = 0$ est une équation linéaire ordinaire stable.

L'erreur de poursuite $e(t) = y(t) - y_{ref}(t)$ converge alors asymptotiquement vers zéro en un temps fini s'il existe une loi de commande qui assure que $s(e, \dot{e}, \dots, e^{v-1}) = 0 \quad \forall t \geq t_f$ avec $t_f > t_0$.

J.J. Slotine propose une forme générale pour la surface de glissement assurant la convergence de la variable à régler vers sa consigne [1].

$$S(\xi) = \left(\frac{d}{dt} + \lambda\right)^{r-1} e(\xi) \quad (3.6)$$

Où ξ est la variable à réguler, $e(\xi)$ est l'écart de la variable à réguler, r est le degré relatif et λ une constante positive. Donc nous pouvons écrire :

$$\text{Pour } r = 1 \Rightarrow S(\xi) = e(\xi)$$

$$\text{Pour } r = 2 \Rightarrow S(\xi) = \dot{e}(\xi) + \lambda e(\xi)$$

$$\text{Pour } r = 3 \Rightarrow S(\xi) = \ddot{e}(\xi) + 2\lambda\dot{e}(\xi) + \lambda^2 e(\xi)$$

Jusqu'à la v ème valeur de $r = v$ la surface de glissement est alors de la forme :

$$S(\xi) = e^{v-1}(\xi) + \binom{v-1}{1} \lambda e^{v-2}(\xi) + \binom{v-1}{2} \lambda^2 e^{v-3}(\xi) + \dots + \lambda^k e^{v-1}(\xi) \quad (3.7)$$

Pour un choix judicieux de λ , $S(x) = 0$ est une équation différentielle linéaire autonome dont la réponse $e(\xi)$ converge vers zéro. L'utilisation de cette surface est très répandue et très pratique parce qu'elle réduit le nombre des paramètres à synthétiser de la surface en un seul coefficient.

3.2.1.3. Conditions d'existence du mode glissant

Un régime glissant est existant sur une surface de glissement si et seulement si au voisinage de cette surface toutes les trajectoires du système se dirigent vers celle-là. Etudier l'existence des modes glissants, revient au même principe de l'étude de la stabilité d'un point d'équilibre, qui est basé sur la deuxième méthode de Lyapunov (voir **Annexe A**). Afin de garantir l'attractivité de la surface de glissement $S(\xi, t) = 0$, nous considérons une fonction de Lyapunov V définie positive, dont sa dérivée par rapport au temps sera définie négative tout au long des trajectoires du système en boucle fermée.

Le glissement peut ne pas avoir lieu que sur une partie de la surface de commutation dans certains cas, à cause de l'attractivité de cette surface qui n'est assuré que dans un domaine restreint S_g c'est le domaine de glissement. Ce domaine est défini par [63] comme suit :

Définition 3.4 [63]

Un domaine S_g dans la variété $S(\xi, t) = 0$ est un domaine de mode glissant si pour tout $\varphi > 0$, il existe $\delta > 0$, tel que tout mouvement commençant dans le δ -voisinage de S_g peut quitter le φ -voisinage de S_g uniquement à travers le φ -voisinage des frontières du domaine S_g .

3.2.1.3.1. Conditions d'existence et d'attractivité

La condition pour que le mode de glissement existe est que la surface de commutation $S(\xi) = 0$ soit attractive. En terme de géométrie, les vecteurs vitesses f^+ et f^- vont être dirigés vers la surface de commutation. Cependant, dans certains cas l'attractivité de la surface de commutation ne peut être assurée que dans un domaine restreint S_g appelé domaine de glissement, donc le glissement n'aura pas lieu sur n'importe quel point de cette surface. Le théorème ci-dessous, fournit les conditions d'existence du mode de glissement selon Filippov [48].

Théorème 3.2 [48]

Considérons le système décrit par (3.1) qui satisfait la condition :

$$\frac{\partial f_i}{\partial \xi_j} \leq K \quad i, j = 1, \dots, n \tag{3.8}$$

Avec K une constante. Cette condition étant vérifiée pour tout ξ dans le domaine $X = S^- \cup S^+$. Soit une fonction S deux fois différentiable, telle que chacune des fonctions f_N^+ et f_N^- est continue par rapport à ξ et t , pour ξ solution de $S(\xi) = 0$, et le vecteur $h = f_N^+ - f_N^-$ est continument différentiable. Si en chaque point de la surface $S(\xi) = 0$, les inégalités $f_N^+ < 0$ et $f_N^- > 0$ sont vérifiées, il existe alors dans le domaine X , une solution unique $\xi(t)$ du système (3.1) qui dépend des conditions initiales de façon continue.

Nous pouvons remarquer que les inégalités $f_N^+ < 0$ et $f_N^- > 0$, signifient qu'au moins dans un voisinage S est attractive.

Un autre théorème, proposé par Utkin [63] fournit les conditions d'existence du mode de glissement en se basant sur les fonctions de Lyapunov.

Théorème 3.3 [63] [60]

Pour que le domaine S_g de dimension $(n-1)$ soit un domaine de glissement pour le système (3.1) et (3.2), il suffit que dans un domaine $\Omega \in \mathcal{D}$ contenant S_g , il existe une fonction de Lyapunov $V(S, \xi, t)$ continument différentiable par rapport à tous ses arguments et satisfaisant les conditions suivantes :

- A- $V(S, \xi, t)$ est définie positive par rapport à S , i.e. $V(S, \xi, t) > 0$ pour $S \neq 0$ avec (ξ, t) arbitraire et $V(S, \xi, t) = 0$ pour $S = 0$.
- B- Sur la sphère $\|S\| = R, \quad \forall \xi \in \Omega$
 - i- $\inf V(S, \xi, t) = h_R, \quad h_R > 0$
 - ii- $\sup V(S, \xi, t) = H_R, \quad H_R > 0$

Avec h_R et H_R dépendant de R et $h_R \neq 0$ si $R \neq 0$.

- C- La dérivée totale de $V(S, \xi, t)$ le long des trajectoires du système a un maximum négatif pour tout ξ de Ω excluant la surface de commutation pour laquelle la commande u n'est pas définie et la dérivée de $V(S, \xi, t)$ n'existe pas.

Théorème 3.4 [60]

Pour obtenir un mouvement de modes glissants stable le long de l'intersection des surfaces de discontinuité $S=0$, il suffit que pour tout ξ et t , il existe une fonction continument différentiable $V(S, \xi, t)$ avec les conditions A, B et C du théorème 3.3 et la condition suivante doit être satisfaite :

$$\lim_{R \rightarrow \infty} h_R \rightarrow \infty \quad (3.9)$$

Une classe de fonctions de Lyapunov classique pour déterminer la condition d'attractivité est celle des fonctions quadratiques du type :

$$V = \frac{1}{2} S^2 \quad (3.10)$$

Alors pour s'assurer de l'attractivité de la surface $S=0$ sur tout le domaine de fonctionnement ϑ , il suffit que la dérivée par rapport au temps de V soit négative :

$$S\dot{S} < 0 \quad \forall \xi \in \vartheta \quad (3.11)$$

Cette inégalité est appelée condition d'attractivité qui n'est pas suffisante pour assurer une convergence en temps fini vers la surface.

Pour une convergence en temps fini, la condition (3.11) qui ne garantit qu'une convergence asymptotique vers la surface de glissement est remplacée par une condition plus restrictive appelée η -attractivité et donnée dans [1].

$$S\dot{S} < \eta|S| \quad \forall \xi \in \vartheta \quad (3.12)$$

Avec η une constante positive. Dans ce cas on peut s'assurer que la surface $S=0$ va être rejointe avec un temps de convergence finie t_c tel que :

$$t_c < \frac{|S(t=0)|}{\eta} \quad (3.13)$$

Il n'est pas suffisant que la dynamique de glissement soit asymptotiquement stable, il faut aussi que le domaine de glissement passe par le point d'équilibre désiré.

Remarque 3.2

À travers les équations (3.10) et (3.11) que les trajectoires du système sont contraintes à converger vers la surface de des deux côtés, car la fonction quadratique de la distance vers cette surface est décroissante, voir figure 3.4.

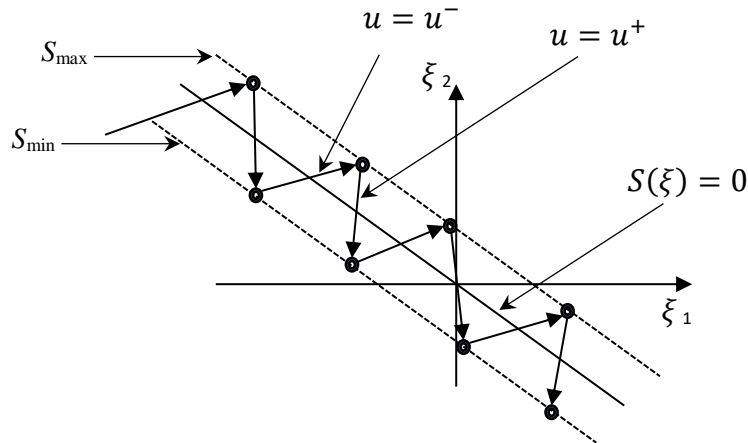


Figure 3.4. Comportement de la trajectoire de l'état par rapport à la surface de commutation

3.2.1.4. Synthèse de la commande par modes glissants (SMC)

Le principe de base de la commande par modes glissants est premièrement d'attirer les états du système vers une région convenablement choisie, puis de maintenir le système dans cette région à l'aide d'une loi de commande conçue pour cela. Donc la commande par modes glissants est constituée de deux parties. Une première partie continue assurant une linéarisation exacte u_{eq} , et l'autre partie discontinue u_d pour garantir l'attractivité de la surface de commutation.

$$u = u_{eq} + u_d \quad (3.14)$$

3.2.1.4.1. La commande équivalente

La commande équivalente u_{eq} sert à maintenir l'état sur la surface de commutation, elle a été proposée par Filippov [48], et reprise par Utkin [60] et Slotine [1] pour décrire le mouvement de glissement sur la surface $S=0$. Car pendant le régime glissant, la fréquence de commutation de la commande est théoriquement infinie, ce qui veut dire que la commande est discontinue à chaque instant. Dans ce cas de figure la théorie des équations différentielles ordinaires n'est plus applicable parceque le système (3.4) ne vérifie plus les conditions classiques d'existence et d'unicité de solutions du théorème de Cauchy-Lipshitz. En effet l'existence et l'unicité de la solution de l'équation (3.4) sont vérifiées si la condition de Lipshitz est satisfaite :

$$\forall t > 0, \exists L > 0 / \forall \xi_1, \xi_2 \in \mathcal{D} \quad \|f(\xi_1, t) - f(\xi_2, t)\| \leq L \|\xi_1 - \xi_2\| \quad (3.15)$$

Il est évident que cette condition n'est pas vérifiée au voisinage de la surface de commutation à cause de la discontinuité de du champ f .

En supposant l'existence du comportement en mode glissant sur la surface de glissement $S(\xi, t)$, la commande équivalente est calculée en utilisant la dérivée de la surface sachant que $\dot{S}(\xi, t) = 0$.

Pour calculer l'expression de cette commande continue, nous allons considérer le système défini par l'équation (3.5). En mode de glissement, la surface de glissement s'annule, par conséquent sa dérivée s'annule aussi, ainsi que la composante discontinue, nous avons alors :

$$\dot{S}(\xi, t) = \frac{dS}{dt} = \frac{\partial S}{\partial \xi} \frac{\partial \xi}{\partial t} = 0 \quad (3.16)$$

En substituant $\dot{\xi}$ donnée par l'équation (3.5) dans (3.16), nous aurons :

$$\frac{\partial S}{\partial \xi} (f(\xi, t) + g(\xi, t)u) = 0 \quad (3.17)$$

En remplaçant u par u_{eq} , et en supposant de $\frac{\partial S}{\partial \xi} g(\xi, t)$ est inversible, nous trouvons alors :

$$u_{eq} = -\left(\frac{\partial S}{\partial \xi} g(\xi, t)\right)^{-1} \left(\frac{\partial S}{\partial \xi} f(\xi, t)\right) \quad (3.18)$$

Par substitution de u_{eq} dans l'équation (3.5), nous obtenons la trajectoire d'état en mode de glissement :

$$\dot{\xi} = f(\xi, t) - g(\xi, t) \left(\frac{\partial S}{\partial \xi} g(\xi, t)\right)^{-1} \left(\frac{\partial S}{\partial \xi} f(\xi, t)\right) \quad (3.19)$$

La grandeur de commande équivalente peut être interprétée comme étant la valeur moyenne que prend la grandeur de commande discontinue lors de la commutation rapide entre u_{max} et u_{min} , comme il est montré schématiquement sur la figure 3.5 [57].

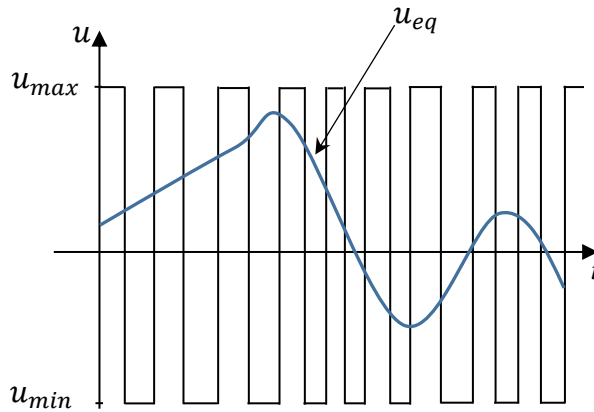


Figure 3.5. Grandeur de commande équivalente u_{eq}

3.2.1.4.2. La commande discontinue

La commande discontinue u_d est déterminée dans le but de garantir l'attractivité de la surface de glissement, et ainsi satisfaire les conditions de convergence de la dynamique du système vers cette surface.

Durant le mode de convergence, en remplaçant l'expression de u_{eq} donnée par (3.18), nous obtenons une nouvelle expression de la dérivée de la surface de commutation :

$$\dot{S}(\xi, t) = \frac{\partial S}{\partial \xi} g(\xi, t) u_d \quad (3.20)$$

La condition d'attractivité énoncée par (3.11) devient alors :

$$S\dot{S} = S \cdot \left[\frac{\partial S}{\partial \xi} g(\xi, t) u_d\right] < 0 \quad (3.21)$$

Dans ce cas il suffit que u_d et $(S \frac{\partial S}{\partial \xi} g(\xi, t))$ soient de signe opposé, plusieurs formes pour la commande discontinue u_d peuvent être proposées, la plus simple et la plus utilisée est exprimée par :

$$u_d = -k \operatorname{sign}(S) \quad (3.22)$$

Où k est une constante positive, et sign représente la fonction classique signe, tel que :

$$\operatorname{sign}(S) = \begin{cases} +1 & \text{si } S(\xi, t) > 0 \\ 0 & \text{si } S(\xi, t) = 0 \\ -1 & \text{si } S(\xi, t) < 0 \end{cases} \quad (3.23)$$

La figure 3.6 montre la forme de la commande discontinue qui correspond à l'expression (3.22).

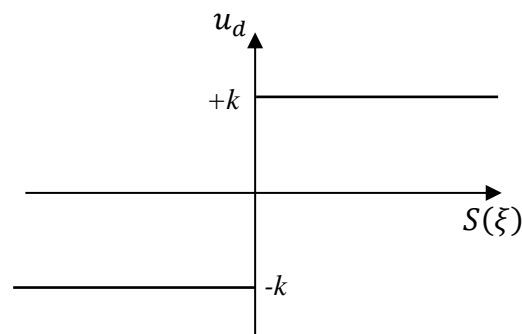


Figure 3.6. Forme de la commande discontinue avec la fonction sign

Remarque 3.3

Le choix de la constante k est très important, si elle est choisie très petite alors le temps de réponse sera long, si dans le cas contraire elle est choisie très grande nous aurons un temps de réponse plus rapide mais il y a risque que des oscillations apparaissent sur la réponse en régime permanent, ce phénomène est appelé *broutement* (ou *chattering* en anglais).

3.2.2. Phénomène de chattering

Un régime glissant idéal est pratiquement inexistant, étant donné que cela signifie que la commande puisse commuter à une fréquence infinie or aucun organe de commande ne peut réaliser cette opération. En présence d'imperfections de commutation telles que des retards au niveau des commutations ou la présence aussi des limites physiques et technologique ou de petites constantes de temps des actionneurs, la discontinuité de la commande provoque au voisinage de la surface de commutation un comportement dynamique particulier qu'on appelle *réticence* ou *broutement* ou encore *chattering* en anglais [44]. Ce phénomène se caractérise par de fortes oscillations autour de la surface, comme il est montré à la figure 3.7 [44].

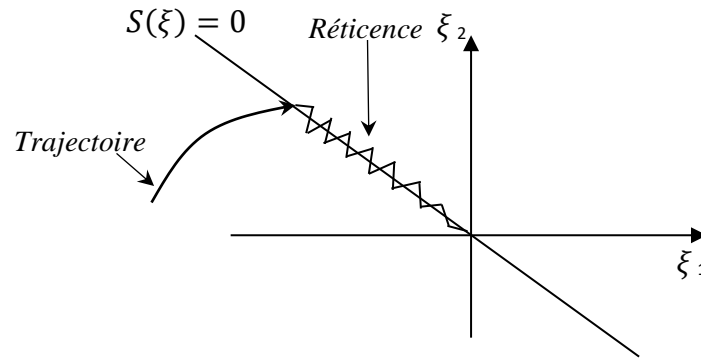


Figure 3.7. Phénomène de réticence

Ce phénomène risque de provoquer la détérioration anticipée de l'organe de commande, ou même l'excitation des dynamiques hautes fréquences non considérées dans la modélisation du système, ce qui peut provoquer la dégradation des performances et même l'instabilité [44]. Dans ce cas les actionneurs aussi sont soumis à de fortes sollicitations mécaniques, ce qui peut engendrer leurs usures rapides.

Pour y remédier à ce problème, différentes solutions ont été proposées dans la littérature. Parmi ces solutions, remplacer la fonction signe de la loi de commande par une fonction à variation plus douce, telle que la fonction de saturation et la tangente hyperbolique...etc [1], cette solution qui a été proposée par Slotine et Sastry [64] [43] est connue sous le nom de couche limite « boundary layer » [1].

Parmi les fonctions les plus utilisées nous avons :

- **La fonction saturation**

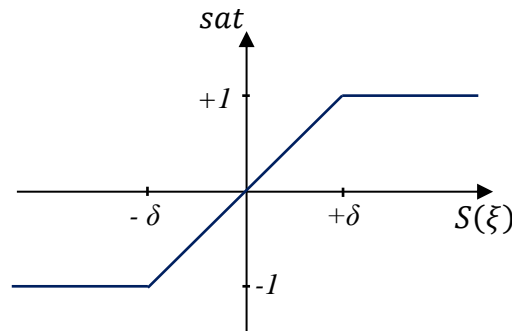


Figure 3.8. Fonction saturation

Son expression est donnée par :

$$sat(S) = \begin{cases} \frac{S}{\delta} & si |S| < \delta \\ sign(S) & si |S| > \delta \end{cases} \quad (2.24)$$

- **La fonction tangente hyperbolique**

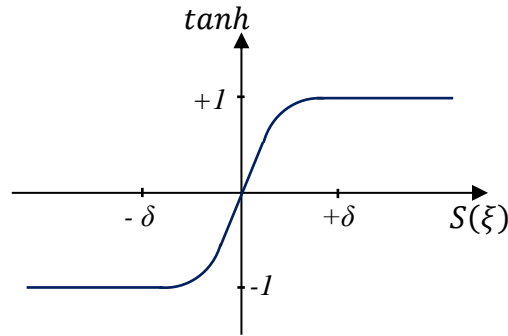


Figure 3.9. Fonction tangente hyperbolique

L'expression de cette fonction est :

$$v(S) = \tanh\left(\frac{S}{\delta}\right) \quad (3.25)$$

D'autres auteurs ont proposés comme solution au phénomène de réticence l'utilisation des contrôleurs flous adaptatifs pour faire varier le gain de la commande discontinue [65], ou pour remplacer carrément la partie discontinue de la commande, ayant une seule entrée la surface de glissement [66], [67], [68], ou deux entrées [69].

[70] a également proposé une méthode pour diminuer le broutement basée sur l'utilisation des observateurs. Le principe de base de cette méthode consiste à introduire une deuxième boucle contenant l'observateur pour générer un régime glissant au lieu de le générer dans la boucle du système. Un mode glissant proche de l'idéal est obtenu dans la seconde boucle, vu que l'observateur est indépendant des dynamiques non modélisées. Mais il peut être troublé par la fréquence d'échantillonnage finie [2].

Il existe également d'autres techniques pour réduire le phénomène de réticence, comme la technique des modes glissants d'ordre supérieur. Introduite par Emelyanov [71] et Levantovsky [72], cette technique permet d'avoir de bonnes propriétés de robustesse, une simplicité de mise en œuvre en plus de la réduction du chattering, tout en conservant les performances du système [58].

3.2.2. Robustesse des modes glissants vis-à-vis des perturbations

Nous allons maintenant considérer le système avec perturbation suivant :

$$\dot{\xi} = f(\xi, t) + g(\xi, t)u + v(\xi, t) \quad (3.26)$$

Où v est l'effet des perturbations externes ou des incertitudes paramétriques sur le modèle.

Lorsque la surface de glissement est atteinte la robustesse du régime glissant vis-à-vis des perturbations est énoncée par le théorème ci-dessous, en généralisant la condition classique d'invariance vis-à-vis des perturbations externes proposée par Drazenovic [73] dans le cas des systèmes linéaires [58].

Théorème 3.5 [74]

Le régime glissant sur la surface de glissement S , du système perturbé (3.26), est invariant vis-à-vis du vecteur perturbation v , si et seulement si v est engendré par $g(\xi)$.

Cette condition est nommée condition de recouvrement connue sous l'appellation « matching condition ».

En calculant la commande équivalente pour le système perturbé (3.26), nous avons :

$$u_{eq} = - \left(\frac{\partial S}{\partial \xi} g(\xi, t) \right)^{-1} \left(\left(\frac{\partial S}{\partial \xi} f(\xi, t) + \frac{\partial S}{\partial t} \right) + \frac{\partial S}{\partial \xi} v \right) \quad (3.27)$$

Par substitution de l'expression (3.27) de u_{eq} dans (3.26), nous avons :

$$\dot{\xi} = f(\xi, t) - g(\xi, t) \left[\left(\frac{\partial S}{\partial \xi} g(\xi, t) \right)^{-1} \left(\left(\frac{\partial S}{\partial \xi} f(\xi, t) + \frac{\partial S}{\partial t} \right) + \frac{\partial S}{\partial \xi} v \right) \right] + v \quad (3.28)$$

Avec $v(\xi, t) = g(\xi, t)\sigma(\xi, t)$, on a :

$$\dot{\xi} = f(\xi, t) - g(\xi, t) \left[\left(\frac{\partial S}{\partial \xi} g(\xi, t) \right)^{-1} \left(\left(\frac{\partial S}{\partial \xi} f(\xi, t) + \frac{\partial S}{\partial t} \right) \right) \right] \quad (3.29)$$

Donc la trajectoire de $\xi(t)$ est invariante par rapport à v .

3.3. La commande de Backstepping

Depuis les premiers travaux sur la commande non linéaire, notamment ceux d'Isidori [75] au milieu des années 80, la théorie de Lyapunov a longtemps été un outil important dans la conception des commandes non linéaires aussi bien que des commandes linéaires. Cependant, l'utilisation de cette théorie en commande non linéaire a été contrainte par des difficultés liées aux choix de la fonction de Lyapunov pour un système donné. Le système sera stable si cette fonction est trouvée, mais par contre la tâche pour trouver une telle fonction a souvent été laissée à l'imagination et à l'expérience du concepteur. Ces dernières années, la mise au point des lois de commande pour les systèmes non linéaires basées sur des procédures récursives a suscité l'intérêt d'une grande partie de la communauté scientifique, tel que la commande stabilisante non linéaire connue sous le nom de backstepping. Cette technique de commande a été développée pour la première fois au début des années 90 par le professeur Kokotovic et ses collaborateurs [76] [77] [78] [79] [80] [81] [82] [3] [45] [4]. Par la suite, elle a été appliquée à divers procédés tel qu'il est présent dans la littérature [83] [84] [85].

3.3.1. Principe de base de la commande par Backstepping

L'idée de base du backstepping consiste à considérer un ensemble de sous-systèmes comme fragment du système global, pour lesquels des lois de commandes virtuelles sont conçues d'une manière récursive, jusqu'à la construction de la loi de commande finale du système global. A chaque étape une fonction de Lyapunov est successivement construite pour garantir une stabilité globale du système en tout temps.

Vu que c'est une technique de commande systématique et récursive se basant sur le principe de stabilité de Lyapunov, cela lui confère l'avantage d'être appliquée à un très grand nombre de systèmes non linéaires [61].

3.3.1.1. Commande stabilisante par Backstepping

Le backstepping est généralement appliqué pour des systèmes de forme triangulaire inférieure, d'une manière récursive en construisant des lois de commande globalement stabilisantes. Cela lui permet de conserver les non linéarités utiles et de dominer les mauvaises [61] [86]. Cependant, le modèle du système doit être écrit sous la forme dite paramétrique pure. La forme d'un tel modèle est donnée par :

$$\begin{cases} \dot{\xi}_1 = f_1(\xi_1, t) + g_1(\xi_1, t)\xi_2 \\ \dot{\xi}_2 = f_2(\xi_1, \xi_2, t) + g_2(\xi_1, \xi_2, t)\xi_3 \\ \vdots \\ \dot{\xi}_j = f_j(\xi_1, \dots, \xi_j, t) + g_j(\xi_1, \dots, \xi_j, t)\xi_{j+1} \\ \vdots \\ \dot{\xi}_n = f_n(\xi_1, \dots, \xi_n, t) + g_n(\xi_1, \dots, \xi_n, t)u \\ y = \xi_1 \end{cases} \quad (3.30)$$

Où $\xi = [\xi_1, \dots, \xi_j, \dots, \xi_n] \in R^n$ est le vecteur d'état, $u \in R$ est le vecteur des commandes et $y \in R$ est le vecteur des sorties. $f_j(\cdot)$ et $g_j(\cdot)$, $j = 1, \dots, n$ sont des fonctions non linéaires connues

L'objectif est de trouver une loi commande à travers la contre réaction du système (3.30) qui garantit une stabilité globale et force le système à converger vers son point d'équilibre. La procédure contient n étapes. Une loi de commande virtuelle α_j appelée aussi commande intermédiaire est déterminée pour les $n - 1$ premières étapes, en utilisant des fonctions de Lyapunov appropriées V_j .

On considère d'abord un premier sous-système de l'équation (3.30) pour $j = 1$, dont sa sortie est z_1 et l'entrée correspondante est ξ_2 . Une commande virtuelle $\alpha_1(\xi_1)$ va rendre ce premier sous-système passif. De la même façon, un deuxième sous-système sera formé pour $j = 2$ avec une entrée ξ_3 , une sortie $z_2 = \xi_2 - \alpha_1$ et une commande virtuelle $\alpha_2(\xi_1, \xi_2)$. Cette démarche sera appliquée jusqu'au $n^{\text{ième}}$ sous-système avec une entrée u et une sortie $z_n = \xi_n - \alpha_{n-1}$. La loi de commande finale u rend la sortie z_n passive et Globalement Asymptotiquement Stable (GAS) [11]. Une fonction de Lyapunov est construite à chaque étape, elle est observée aussi comme une fonction de stockage de type quadratique [11].

1^{ère} étape

Un premier sous-système est considéré :

$$\dot{\xi}_1 = f_1(\xi_1) + g_1(\xi_1)\xi_2 \quad (3.31)$$

Une première variable du backstepping z_1 est définie comme sortie virtuelle de ce premier sous-système tel que $z_1 = \xi_1$.

La loi de commande virtuelle α_1 qui est une fonction stabilisante est alors définie par :

$$z_2 = \xi_2 - \alpha_1 \quad (3.32)$$

Avec z_2 est une nouvelle variable du deuxième sous-système.

Le premier sous-système (3.31) devient alors :

$$\dot{z}_1 = \dot{\xi}_1 = f_1 + g_1 z_2 + g_1 \alpha_1 \quad (3.33)$$

Pour un tel sous-système, une fonction de Lyapunov quadratique constitue un bon choix :

$$V_1(z_1) = \frac{1}{2} z_1^2 \quad (3.34)$$

Sa dérivée est donnée par :

$$\dot{V}_1 = z_1 \dot{z}_1 = g_1 z_1 z_2 + z_1 (f_1 + g_1 \alpha_1) \quad (3.35)$$

Posons

$$(f_1 + g_1 \alpha_1) = -k_1 z_1 \quad (3.36)$$

Où k_1 est une constante positive.

Le choix de α_1 doit garantir que \dot{V}_1 soit définie négative, alors un contrôle linéaire remplace la non linéarité. L'équation (3.36) donne :

$$\alpha_1 = \frac{1}{g_1} (-k_1 z_1 - f_1) \quad (3.37)$$

En substituant (3.37) dans (3.35), nous aurons :

$$\dot{V}_1 = -k_1 z_1^2 + g_1 z_1 z_2 \quad (3.38)$$

Le terme $g_1 z_1 z_2$ de \dot{V}_1 sera éliminé dans l'étape suivante, afin de s'assurer que \dot{V}_1 sera négative. Ce qui garantira la convergence asymptotiquement de z_1 vers 0.

2^{ème} étape

Nous considérons, dans ce cas, un deuxième sous-système en plus du premier :

$$\dot{\xi}_2 = f_2(\xi_1, \xi_2, t) + g_2(\xi_1, \xi_2, t) \xi_3 \quad (3.39)$$

Une nouvelle variable de sortie est choisie dans cette étape, $z_3 = \xi_3 - \alpha_2$ où ξ_3 est l'entrée du deuxième sous-système, ce qui donne :

$$\dot{z}_2 = \dot{\xi}_2 - \dot{\alpha}_1 = f_2 + g_2 z_3 + g_2 \alpha_2 - \dot{\alpha}_1 \quad (3.40)$$

La dérivée de la fonction stabilisante α_1 est donnée par :

$$\dot{\alpha}_1 = \frac{\partial \alpha_1}{\partial t} = \frac{\partial \alpha_1}{\partial \xi_1} \dot{\xi}_1 = \frac{\partial \alpha_1}{\partial \xi_1} (g_1 \xi_2 + f_1) \quad (3.41)$$

Par substitution de (3.41) dans (3.40), nous obtenons :

$$\dot{z}_2 = f_2 + g_2 z_3 + g_2 \alpha_2 - \frac{\partial \alpha_1}{\partial \xi_1} (g_1 \xi_2 + f_1) \quad (3.42)$$

Une deuxième fonction de Lyapunov V_2 est construite et prend la forme suivante :

$$V_2(z_1, z_2) = V_1(z_1) + \frac{1}{2} z_2^2 \quad (3.43)$$

En dérivant V_2 nous aurons :

$$\dot{V}_2 = \dot{V}_1 + z_2 \dot{z}_2 \quad (3.44)$$

Ce qui donne :

$$\dot{V}_2 = -k_1 z_1^2 + g_2 z_2 z_3 + z_2 (f_2 + g_2 z_3 + g_2 \alpha_2 - \frac{\partial \alpha_1}{\partial \xi_1} (g_1 \xi_2 + f_1)) \quad (3.45)$$

Posons

$$\left(f_2 + g_2 z_3 + g_2 \alpha_2 - \frac{\partial \alpha_1}{\partial \xi_1} (g_1 \xi_2 + f_1) \right) = -k_2 z_2 \quad (3.46)$$

Avec k_2 est une constante positive.

Le choix de la nouvelle commande virtuelle est fait de tel sorte que \dot{V}_2 soit négative. Une conception est possible :

$$\alpha_2 = \frac{1}{g_2}(-k_2 z_2 - f_2 - g_2 z_3 + \frac{\partial \alpha_1}{\partial \xi_1}(g_1 \xi_2 + f_1)) \quad (3.47)$$

Par substitution de (3.47) dans (3.45), nous obtiendrons :

$$\dot{V}_2 = -k_1 z_1^2 - k_2 z_2^2 + g_2 z_2 z_3 \quad (3.48)$$

Avec :

$$\dot{z}_2 = -k_2 z_2 + g_2 z_3 + g_1 z_1 \quad (3.49)$$

Pour s'assurer que la fonction de Lyapunov \dot{V}_2 soit négative, le terme $g_2 z_2 z_3$ sera éliminé dans l'étape suivante. Ceci garantira la convergence asymptotiquement de z_2 vers 0.

j^{ième} étape

Nous allons considérer dans ce cas d'une façon générale le $j^{\text{ième}}$ sous-système :

$$\dot{\xi}_j = f_j(\xi_1, \dots, \xi_j, t) + g_j(\xi_1, \dots, \xi_j, t)\xi_{j+1} \quad (3.50)$$

La nouvelle variable de sortie virtuelle est choisie dans cette étape, $z_j = \xi_j - \alpha_{j-1}$, ce qui donne :

$$\dot{z}_j = \dot{\xi}_j - \dot{\alpha}_{j-1} = -k_j z_j + g_j z_{j+1} + f_{j^{\text{ième}}} \quad (3.51)$$

Où $z_{j+1} = \xi_{j+1} - \alpha_j$, avec α_j est la commande virtuelle a déterminée qui correspond à l'étape j .

La loi de commande virtuelle est conçue de telle sorte que la fonction de Lyapunov soit définie négative :

$$V_j = V_{j-1} + \frac{1}{2} z_j^2 \quad (3.52)$$

La dérivée de cette fonction par rapport au temps est donnée par :

$$\dot{V}_j = \dot{V}_{j-1} + z_j \dot{z}_j \quad (3.53)$$

Posons :

$$\sum_{k=1}^{j-1} (k_k z_k^2) + z_j (f_{j^{\text{ième}}} + g_j z_{j+1} + g_j \alpha_j + g_{j-1} z_{j-1}) = -k_j z_j \quad (3.54)$$

La commande virtuelle est donnée par :

$$\alpha_j = \frac{1}{g_j} (-k_j z_j - g_{j-1} z_{j-1} - f_{j^{\text{ième}}}) \quad (3.55)$$

Avec :

$$f_{j^{\text{ième}}} = f_j - \sum_{k=1}^{j-1} \frac{\partial \alpha_{j-1}}{\partial \xi_k} (f_k + g_k \xi_{k+1}) \quad (3.56)$$

En substituant l'équation (3.55) dans (3.51) et (3.53), nous aurons :

$$\dot{z}_j = -k_j z_j + g_j z_{j+1} - g_{j-1} z_{j-1} \quad (3.57)$$

$$\dot{V}_j = -\sum_{k=1}^{j-1} (k_k z_k^2) + g_j z_j z_{j+1} \quad (3.58)$$

Afin de s'assurer que la fonction de Lyapunov \dot{V}_j soit définie négative, le terme $g_j z_j z_{j+1}$ sera éliminé dans l'étape qui suit. Ceci garantira la convergence asymptotiquement de z_j vers 0.

n^{ième} étape

C'est la dernière étape pour concevoir la commande par backstepping, puisque en calculant \dot{z}_n la commande finale u apparaîtra :

$$\dot{z}_n = \dot{\xi}_n - \dot{\alpha}_{n-1} = g_n u + f_{n^{\text{ième}}} \quad (3.59)$$

Avec :

$$f_{n^{\text{ième}}} = f_n - \sum_{k=1}^{n-1} \frac{\partial \alpha_{n-1}}{\partial \xi_k} (f_k + g_k \xi_{k+1}) \quad (3.60)$$

C'est la loi de commande u pour cette étape qui est conçu pour rendre la fonction globale de Lyapunov définie positive.

$$V_n = V_{n-1} + \frac{1}{2} z_n^2 \quad (3.61)$$

Sa dérivée par rapport au temps est donnée par :

$$\dot{V}_n = \dot{V}_{n-1} + z_n \dot{z}_n \quad (3.62)$$

De la même façon que dans la $j^{\text{ième}}$ étape \dot{V}_n peut s'écrire sous la forme suivante :

$$\dot{V}_n = -\sum_{k=1}^{n-1} (k_k z_k^2) + z_n (g_{n-1} z_{n-1} + g_n u + f_{n^{\text{ième}}}) \quad (3.63)$$

La loi de commande u choisie est donnée par :

$$u = \frac{1}{g_n} (-k_n z_n - g_{n-1} z_{n-1} - f_{n^{\text{ième}}}) \quad (3.64)$$

Ainsi la loi de commande conçue stabilisera le système (3.30) d'ordre n . cette commande u existe et elle est considérée dans la boucle de retour comme une fonction de stabilisation régulière.

Par substitution de (3.64) dans (3.59) et (3.63), nous obtenons :

$$\dot{z}_n = -k_n z_n - g_{n-1} z_{n-1} \quad (3.65)$$

$$\dot{V}_n = -\sum_{k=1}^n (k_k z_k^2) \quad (3.66)$$

La dérivée de la fonction de Lyapunov globale est donc définie négative $\dot{V}(z) < 0, \forall z \neq 0$. Le point d'équilibre $\xi = \xi_{eq}$ de (3.30) est asymptotiquement stable puisque la fonction $V(z)$ est continument différentiable et ayant les propriétés citées dans le théorème de LaSalle (Voir Annexe A), donc le système global ayant comme coordonnées (z_1, \dots, z_n) est globalement asymptotiquement stable (GAS) [11]:

$$\begin{cases} \dot{z}_1 = -k_1 z_1 - g_1 z_2 \\ \vdots \\ \dot{z}_j = -k_j z_j - g_{j-1} z_{j-1} + g_j z_{j+1} \\ \vdots \\ \dot{z}_n = -k_n z_n - g_{n-1} z_{n-1} \end{cases} \quad (3.67)$$

3.4. Conclusion

Dans ce chapitre, nous avons présenté quelques aspects théoriques de deux types de commandes non linéaires robustes, à savoir la commande par mode glissant et la commande par backstepping. Cette dernière est une procédure récursive qui divise un problème de conception pour le système non linéaire complet en une séquence de problèmes pour des systèmes d'ordre réduits, ceci permet d'avoir d'une manière récursive une loi de commande permettant d'assurer la stabilité en boucle fermée du système usant la fonction de Lyapunov. La commande par mode glissant force le système à évoluer sur une surface de glissement, ce qui rend sa dynamique insensible aux erreurs de modélisation et aux variations paramétriques. Cependant le phénomène de broutement représente un handicap pour cette technique de commande.

Il faut noter que ces deux approches sont simples à mettre en œuvre et robustes vis-à-vis de perturbations extérieures et des incertitudes. Par la suite nous allons appliquer ces deux techniques de commande pour contrôler un robot manipulateur.

CHAPITRE 4

MÉTHODES D'OPTIMISATION MÉTAHEURISTIQUES

4.1. Introduction

L'être humain est de nature exigeant, il adore la perfection, il cherche toujours à améliorer son quotidien, tout en essayant bien sûr de minimiser ses charges, sa facture d'électricité ou la consommation du carburant de sa voiture, etc. donc il cherche toujours à maximiser ses gains et minimiser ses dépenses à vrai dire optimiser.

Accroître le taux de production tout en réduisant le coût est une requête très sollicitée dans le milieu industriel, ce qui a fortement contribué à l'émergence de l'optimisation qui est considérée comme une branche très importante des mathématiques. Les travaux faisant appel aux méthodes d'optimisations ont connus une croissance très rapide ces dernières années dans plusieurs domaines de la science tels que l'économie, les sciences sociales, la classification, la théorie de la décision et bien sûr l'engineering [87] [88] [89].

Utiliser une méthode pour la résolution d'un problème d'optimisation dépend de plusieurs facteurs tels que la nature du problème, de son degré de complexité et aussi des outils mathématiques et numériques disponibles. L'usage des méthodes dites classiques est limité aux cas où la modélisation du problème à résoudre peut être analytiquement et/ou convexe, ce qui n'est pas le cas malheureusement pour de nombreuses situations [89].

Les méthodes stochastiques connues sous le nom de méta-heuristiques sont des outils puissants et efficaces pour résoudre le problème d'optimisation dans de telles situations. Ces algorithmes méta-heuristiques sont formellement définis comme des processus itératifs qui combinent intelligemment différents concepts pour explorer et exploiter l'espace de recherche. Des stratégies d'apprentissage sont utilisées pour structurer l'information afin de trouver efficacement des solutions quasi optimales. Parmi ces algorithmes nous pouvons citer les colonies de fourmis [90] [87], les algorithmes génétiques [8], le recuit simulé [91], les essaims de particules (PSO) [9], etc.

Les algorithmes génétiques (AGs) sont une méthode d'optimisation s'appuyant sur des techniques dérivées de la génétique et de l'évolution naturelle, inspirés de la théorie darwinienne à savoir les mutations, les croisements, la sélection...etc. [92]. Ces algorithmes sont basés sur des mécanismes très simples, ils sont robustes car ils peuvent résoudre des problèmes fortement non linéaires et discontinus, et efficaces car ils font évoluer non pas une solution mais toute une population de solutions potentielles et donc ils bénéficient d'un parallélisme puissant.

A l'image des AGs, l'optimisation par les PSO a connu un grand succès ces dernières années dans divers domaines. Cette technique inspirée du comportement social des animaux a été largement utilisée en automatique pour optimiser les paramètres des contrôleurs comme le PID [93] [94] [95].

Dans ce chapitre nous aborderons quelques définitions générales des méthodes d'optimisation. Nous donnons aussi quelques concepts théoriques de deux méthodes d'optimisation qui sont les AGs et le PSO.

4.2. Optimisation

4.2.1. Définition

L'optimisation consiste à trouver les meilleures solutions pour un problème donné dans des conditions préétablies. Le mathématicien vient dans ce cas pour concrétiser le vœu de l'ingénieur qui cherche à maximiser le bénéfice et/ou minimiser l'effort requis, en modélisant les différents problèmes sous des fonctions coûts dépendantes de certaines variables. Donc l'optimisation peut être définie comme étant le processus qui sert à trouver les variables donnant le minimum ou le maximum de cette fonction [96] [89].

D'un point de vue mathématique, optimiser une fonction $f(\xi)$ où $\xi \in \delta$ (avec δ est un ensemble de solutions) revient à trouver une solution $\xi_{opt} \in \delta$, dite solution optimale, tel que $\forall \xi \in \delta$ [89]:

- Dans un problème de maximisation $f(\xi_{opt}) \geq f(\xi)$;
- Dans un problème de minimisation $f(\xi_{opt}) \leq f(\xi)$;

La fonction coût f reflète la qualité des solutions, elle est appelée aussi fitness ou fonction objectif.

Chaque problème de maximisation peut se transformer sous la forme d'un problème de minimisation et vis-versa.

Deux types de minima peuvent être distingués dans le domaine de l'optimisation [89]:

- **Minimum local** : S'il existe un ensemble $N \subset \delta$, contenant ξ_{loc} , tel que $\forall \xi \in N$, $f(\xi_{loc}) < f(\xi)$, avec $\xi \neq \xi_{loc}$, alors ξ_{loc} représente un minimum local.
- **Minimum global** : Si on peut prouver que $\forall \xi \in \delta$, $f(\xi_{glob}) < f(\xi)$, alors ξ_{glob} représente un minimum global.

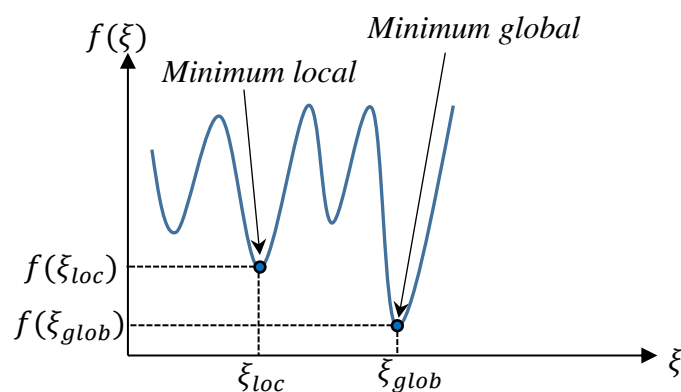


Figure 4.1. Minimum local vs minimum global

4.2.2. Problème d'optimisation

Nous pouvons définir un problème d'optimisation par un ensemble de variables, avec un ensemble de contraintes si elles existent et une fonction objectif à minimiser (ou maximiser). La résolution de ce problème revient donc à trouver la ou les meilleures solutions avec satisfaction de l'ensemble des contraintes. Le problème peut être formulé de la façon suivante [96]:

$$\begin{aligned} & - \textit{Problème sans contrainte} : \\ \min_{\xi \in R^n} J(\xi) \end{aligned} \quad (4.1)$$

$$\begin{aligned} & - \textit{Problème avec contrainte} : \\ \min_{\xi \in V} J(\xi) \end{aligned} \quad (4.2)$$

Où $\xi = [\xi_1, \xi_2, \dots, \xi_n]^T$ représente le vecteur des n variables de décision, et $J(\xi)$ est la fonction objectif. Minimiser $J(\xi)$ est équivalent à maximiser $-J(\xi)$.

Les contraintes peuvent être de deux types :

$$\begin{aligned} & - \textit{Type inégalités} : \\ V = \{ \xi \in R \text{ tels que } : w_j(\xi) \leq 0, j = 1, \dots, m \} \end{aligned} \quad (4.3)$$

Où m représente le nombre de contrainte de type inégalité.

$$\begin{aligned} & - \textit{Type égalités} : \\ V = \{ \xi \in R \text{ tels que } : h_j(\xi) = 0, j = 1, \dots, k \} \end{aligned} \quad (4.4)$$

Où k représente le nombre de contrainte de type égalité.

Un problème d'optimisation doit posséder un espace de recherche qui représente l'ensemble des solutions possibles du problème [89].

Nous pouvons classer un problème d'optimisation de plusieurs façons [89]:

- Existe-t-il des contraintes ou pas.
- Statique ou dynamique (un changement avec le temps de la fonction objectif est possible : les bourses économiques, le routage dans les réseaux, etc.).
- Mono-objectif ou multi-objectif (plusieurs fonctions objectifs à optimiser).

Actuellement les problèmes d'optimisations interviennent dans divers domaines et représentent des axes de recherche importants, en automatique, l'électronique, la mécanique, l'économie, le traitement d'image, la recherche opérationnelle, l'informatique, etc [97] [95] [88] [89] .

4.2.3. Méthodes d'optimisation

En dépit de l'abondance des méthodes d'optimisation proposées, nous pouvons suivant leurs approches leurs discerner des traits caractéristiques. Ainsi nous proposons la classification suivante : Les méthodes déterministes (ou classiques) et les méthodes non-déterministes (méta heuristiques).

4.2.3.1. Méthodes déterministes (classiques)

Lorsque la méthode de résolution évolue d'une manière prévisible ne laissant aucune place au hasard, alors celle-ci est qualifiée de *déterministe*. Tout d'abord ces méthodes ont été conçues

pour résoudre d'une façon exacte des problèmes particuliers tels que les problèmes continus et linéaires avec des contraintes linéaires. Les pionniers à qui revient l'existence des méthodes d'optimisation classiques sont Newton, Lagrange, Euler Leibnitz et Cauchy. Malgré leurs contributions, il n'y a pas eu de très grands progrès, ce n'est qu'à partir de la deuxième moitié du vingtième siècle avec l'utilisation des ordinateurs numériques, permettant ainsi la mise en œuvre des procédures d'optimisation qu'on a pu voir des progrès spectaculaires. Mais de l'autre côté les problèmes à optimiser sont devenus plus complexes [88].

Généralement les méthodes déterministes sont adoptées lorsque la fonction objectif du problème possède certaines propriétés telles que la dérivabilité, la continuité ou la convexité [88]. La qualité principale de ces méthodes est qu'elles n'ont pas besoin de point de départ et permettent d'avoir, à la convergence, une solution exacte du problème d'optimisation traité avec une garantie absolue, ainsi qu'une bonne gestion des contraintes. Cependant, il faut noter que ces méthodes restent d'usage tant que le nombre de variables n'est pas important (une vingtaine maximum).

Parmi les méthodes déterministes les plus réputées, nous pouvons trouver : la méthode du simplexe de Dantzig [98], la méthode du gradient [99] et aussi la méthode de Newton [100].

4.2.3.2. Méthodes non-déterministes (Métaheuristiques)

Les méthodes déterministes peuvent être limitées dans la résolution de certains problèmes d'optimisation complexes, Parmi ces limites, nous avons la non-dérivabilité, la discontinuité, l'absence de convexité ou même des difficultés pour définir avec précision la fonction objectif. De plus, la plupart des problèmes d'optimisation sont NP-difficile, et souffre d'un temps de résolution trop long. Donc leur utilisation est limitée à certains problèmes de petite taille.

Les approches métaheuristiques paraissent comme dans ce cas comme un moyen intéressant pour résoudre les problèmes d'optimisation. Ces méthodes non déterministes appelées aussi stochastiques font appel à des tirages de nombre aléatoires. Elles permettent aussi d'explorer plus efficacement l'espace de recherche.

Le mot métaheuristique introduit pour la première fois par Fred Glover [101] en 1986, est composé de deux mots d'origine Grec :

- Méta : un suffixe qui signifie « au-delà ».
- heuristique : dérivé du verbe « heuriskein » qui signifie « trouver ».

Plusieurs définitions ont été proposées dans la littérature pour les métaheuristiques. Nous proposons celle donnée par [97] : Une métaheuristique est formellement définie comme un processus de génération itératif qui guide une heuristique subordonnée en combinant intelligemment différents concepts pour explorer et exploiter l'espace de recherche, des stratégies d'apprentissage sont utilisées pour structurer l'information afin de trouver efficacement des solutions quasi optimales.

Les métaheuristiques sont caractérisées par le caractère stochastique, c.à.d. que la recherche de des solutions est partiellement conduite d'une manière aléatoire, et sont généralement itératives, permettant d'améliorer successivement les solutions obtenues. Elles sont inspirées de processus naturels, tel que la biologie (les algorithmes génétiques, recherche tabou...), l'éthologie (colonies de fourmis, PSO...) ou la physique (recuit simulé...) [102] [90] [9] [91].

Selon Blum et Roli [102] les métaheuristiques ont en commun les propriétés suivantes [89] :

- les approches métaheuristiques vont de procédures simples de recherche locale à des procédures très complexe.
- L'objectif visé est d'explorer efficacement tous l'espace de recherche dans le but de trouver des solutions (presque) optimales.
- Les métaheuristiques sont des procédures permettant de guider le processus de recherche.
- Peuvent utiliser des mécanismes pour éviter la stagnation dans des optimaux locaux.
- Les métaheuristiques sont généralement approximatives et non-déterministes.
- Les métaheuristiques peuvent utiliser l'expérience accumulée pendant la recherche de l'optimum, pour mieux orienter la suite du processus de recherche.

4.2.3.2.1. Classification des métaheuristiques

Plusieurs classifications des métaheuristiques sont possibles, selon la nature de l'algorithme en lui-même, on peut distinguer [102] :

a- Inspirées de phénomènes naturels vs non-inspirées de phénomènes naturels

D'une manière intuitive une classification des métaheuristiques est possible en séparant les méthodes inspirées de la nature, des autres qui ne le sont pas. A titre d'exemple, les algorithmes des colonies de fourmis et les algorithmes génétiques sont inspirées respectivement du comportement naturel des fourmis en cherchant la nourriture et la de théorie de l'évolution. En contrepartie, l'algorithme de recherche tabou n'est pas inspiré d'un phénomène naturel.

Cette classification est parfois difficile à réaliser, et ne semble pas significative. En effet, il y a de nombreuses métaheuristiques hybrides récentes qui sont difficiles à classer dans l'une des deux classes. Par exemple, l'usage d'une mémoire dans le cas de l'algorithme de recherche tabou pourrait le faire classer dans la catégorie des méthodes inspiré de la nature.

b- Méthodes de trajectoire vs Méthodes basées sur une population de points

On peut également séparer les métaheuristiques travaillant avec une population de solutions de celles à solution unique. Ces dernières qui ne manipulent qu'une seule solution à la fois sont appelées aussi méthodes de trajectoires. L'algorithme de recherche tabou, la Recherche à Voisinages Variables et le Recuit Simulé sont des exemples typiques de méthodes de recherche locale à solution unique. Par contre, les métaheuristiques à population de solutions au fur et à mesure des itérations, améliorent une population des solutions. Ces méthodes utilisent la population comme un facteur de diversité. Les algorithmes génétiques, l'optimisation par essaim particules et la recherche par dispersion sont des exemples de méthodes appartenant à cette catégorie.

c- Fonction objectif statique vs fonction objectif dynamique

La manière d'utiliser la fonction objectif peut être considérée comme un critère pour classer les métaheuristiques. Lorsqu'elles travaillent directement sur la fonction à optimiser f , ces métaheuristiques sont dites statiques. Par contre si elles utilisent une fonction g obtenue à partir de f avec l'ajout de quelques composantes permettant ainsi de modifier la topologie de l'espace des solutions, ces composantes additionnelles ont la possibilité de varier pendant le processus de recherche.

d- Usage de mémoire à court terme vs mémoire à long terme

L'utilisation de l'historique de la recherche peut être considérée comme un critère de classification. Certains algorithmes utilisent l'historique, alors que d'autres n'ont aucune

mémoire du passé. Ces derniers sont des processus markoviens puisque la situation courante c'est elle qui détermine totalement l'action à réaliser. L'utilisation de l'historique de la recherche peut être effectuée de différentes manières. On peut faire généralement la différence entre les méthodes ayant une mémoire à long terme de celles qui ont une mémoire à court terme.

e- Diversification vs intensification

Un autre critère de classification selon l'utilisation de la diversification ou de l'intensification. Le terme diversification, signifie une exploration assez large de l'espace de recherche, alors que l'intensification veut dire l'exploitation de l'information accumulée durant la recherche.

Dans ce qui suit, nous nous sommes intéressés particulièrement à deux méthodes basées sur des populations de solutions et qui ont trouvées un large domaine d'utilisation en engineering. Ces méthodes sont « les algorithmes génétiques » et « les essaims de particules PSO ».

4.3. Les algorithmes génétiques

L'objectif visé par tous les algorithmes d'optimisation c'est de trouver une solution optimale tout en respectant un ensemble de contraintes.

Inspirées des phénomènes naturels, les algorithmes génétiques sont des algorithmes informatiques qui reposent dans leurs conceptions sur la théorie de Darwin [103] qui date des années 1800. Cette théorie repose sur deux principes. Le premier est qu'au sein d'un environnement, il n'y a que les espèces qui sont les mieux adaptées qui ont la possibilité de perdurer au cours du temps, pour le reste elles sont menacées de disparaître. Le deuxième principe de cette théorie repose sur le fait que dans chaque espèce, la population est renouvelée à l'aide des meilleurs individus, c'est ce qu'on appelle l'évolution et elle est opérée selon Darwin [103] sur les chromosomes contenus dans leurs ADN.

Mais il fallait attendre les années cinquante du vingtième siècle pour voir les premiers travaux sur les algorithmes génétiques, lorsque des biologistes américains arrivent à simuler des structures biologiques sur ordinateur. Ensuite le professeur John Holland de l'Université du Michigan commença à s'intéresser à ce qui allait devenir les algorithmes génétiques au début des années soixante. Un premier aboutissement de ses travaux fut en 1975 en publiant son livre intitulé *Adaptation in Natural and Artificial Systems* [104]. Malheureusement à cette époque les ordinateurs n'étaient pas assez puissants pour pouvoir envisager une application sur des problèmes concrets de grande taille.

La publication de l'ouvrage de Goldberg [8] qui décrit d'une manière concrète l'utilisation des algorithmes génétiques dans le cadre de la résolution des problèmes réels a permis de mieux faire connaître ces algorithmes et a signé le début d'un nouvel intérêt pour cette technique d'optimisation.

Pratiquement la performance des algorithmes génétiques est basée sur une approche empirique du problème et repose sur le savoir-faire de l'utilisateur.

4.3.1. Terminologie et concepts de base des algorithmes génétiques

Les algorithmes génétiques sont des méthodes d'optimisation s'appuyant sur des techniques issues de la génétique et de l'évolution naturelle selon les règles de Mendel et la théorie de Darwin, à savoir le croisement, la mutation, la sélection, etc. [92]. Durant l'évolution, les individus ou les organismes qui y participent interagissent et vivent ensemble dans un

environnement où ces ils partagent les mêmes ressources et se reproduisent pour créer de nouvelles générations. Les individus les plus puissants possèdent plus de chance de se reproduire et de subsister que les individus les plus faibles. De ce fait, les gènes des individus qui se sont bien adaptés se transmettront à d'autres individus au cours des générations et l'espèce s'adaptera à son environnement sans cesse.

Puisque ces algorithmes emploient une terminologie empruntée à celle de la génétique, nous devons définir les concepts suivants :

4.3.1.1. Génotype et phénotype

En termes de la génétique, le génotype est défini comme étant l'ensemble des valeurs des gènes situés sur des chromosomes tandis que le phénotype indique la solution réelle après la modification du chromosome en d'autre terme le phénotype représente l'ensemble des caractères d'un individu. Ces caractères ont la possibilité d'être acquis d'une manière héréditaire (c.à.d. transmission d'une génération à une autre) ou modifiés par l'environnement (cicatrice, musculature, etc.). Pour générer une nouvelle population, plusieurs opérateurs génétiques interviennent tels que le croisement, la mutation, et la sélection pour manipuler les chromosomes [105].

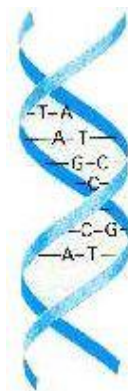


Figure 4.2. Le génotype

4.3.1.2. La population

Dans un problème d'optimisation les algorithmes génétiques vont faire évoluer un ensemble de solutions candidates, celles-ci est appelées population. Donc la population est l'ensemble des individus (chromosomes) qui représentent chacun une solution possible d'un problème donnée. À tout instant t , cette population est nommée génération.

4.3.1.3. Les individus

Tout point de l'espace d'évolution est appelé individu (une solution). Ces individus sont caractérisées par une architecture génétique "génotype du chromosome ou individu" (codage des solutions du problème) contenant ainsi toute information nécessaire pour évaluer la fonctionnelle a optimisée. Classiquement un chromosome possède N gènes qui sont dans un problème binaire N bits de valeur 0 ou 1.

4.3.1.4. Les chromosomes

Un chromosome est un ensemble de gènes. Dans le langage de la génétique c'est la structure qui donne les caractéristiques d'un individu.

4.3.1.5. Les gènes

L'élément de base constituant les chromosomes est appelé un gène (en génétique il donne le caractère de la chaîne ADN). Chaque chromosome est constitué d'une suite de gènes permettant de coder les fonctionnalités de l'organisme.

La composition d'une population dans un algorithme génétique est donnée par la figure 4.3.

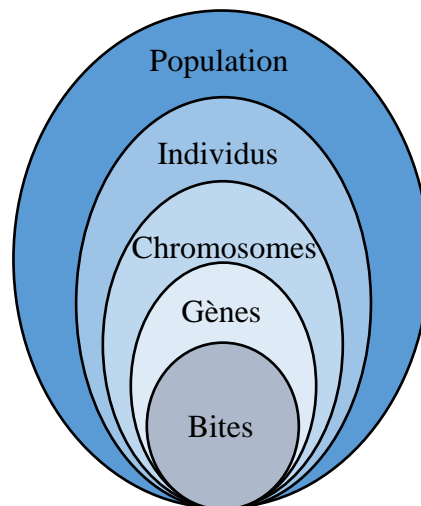


Figure 4.3. Les niveaux d'organisation d'un Algorithme Génétique [106]

4.3.2. Méthodologie adoptée

D'une façon générale, pour optimiser une fonction $f(\xi)$ quelconque, il suffit de calculer sa dérivée par rapport à la variable ξ et ensuite calculer la valeur de cette variable pour laquelle l'expression résultante s'annule. Par la suite pour vérifier si cette solution optimale correspond à un minimum ou un maximum, la dérivée seconde est requise.

$$\max f(\xi) \rightarrow \frac{df}{d\xi} = 0, \quad \frac{d^2f}{d\xi^2} < 0 \quad (4.5)$$

Pratiquement toutes les techniques d'optimisation conventionnelles se basent sur ce genre de procédure. Malgré la simplicité dans la conception de cette méthode, elle présente des limites en termes de pratique. Plusieurs problèmes existent malheureusement, dont il est quasiment impossible de calculer d'une manière analytique une solution pour ξ . Les algorithmes génétiques apparaissent comme une alternative pour résoudre tous ces problèmes.

La recherche du ou des extrema d'une fonction définie sur un espace de données en utilisant les algorithmes génétiques se fait d'une manière itérative (génération), jusqu'à ce qu'un test d'arrêt soit vérifié. Pour pouvoir utiliser un algorithme génétique pour résoudre un problème d'optimisation, plusieurs étapes doivent être établies [107]:

- Le codage des différents éléments de la population ;
- La génération d'une population initiale avec une taille fixe ;
- Définir la fonction objectif (fitness) pour évaluer les solutions ;

- Appliquer les opérateurs génétiques (croisement, mutation, etc.) pour générer de nouvelles solutions ;
- Un mécanisme pour choisir les solutions qui doivent rester et celles qui doivent disparaître, avec un test d'arrêt.

Le principe de fonctionnement des algorithmes génétiques est montré sur la figure 4.4. L'algorithme débute avec un ensemble de solutions possibles du problème (les individus), pour constituer une population. Des éléments variables forment les individus, qui représentent dans un dispositif les paramètres à ajuster. La conception de cette population se fait d'une façon aléatoire dans un intervalle prédéfini (les limites dictées par l'aspect pratique du problème d'optimisation). Par la suite, certaines solutions sont sélectionnées à partir de la première population pour former une nouvelle population, en utilisant les opérateurs génétiques (la sélection, le croisement et la mutation). Le but escompté à travers ceci est que la nouvelle population soit nettement meilleure que la population qui la précède. Une sélection aléatoire permet de choisir les solutions qui vont servir à créer de nouvelles solutions, en se basant sur leurs fonctions objectifs construites en concordance avec le problème d'optimisation. Les meilleurs individus auront plus de chance de se reproduire (c'est-à-dire, la probabilité d'être choisi pour subir les différents opérateurs génétiques sera plus grande). Cette procédure sera répétée jusqu'à la satisfaction d'un critère de convergence (un nombre prédéfini de générations, un temps de simulation, pas d'amélioration sur la performance des solutions, etc.).

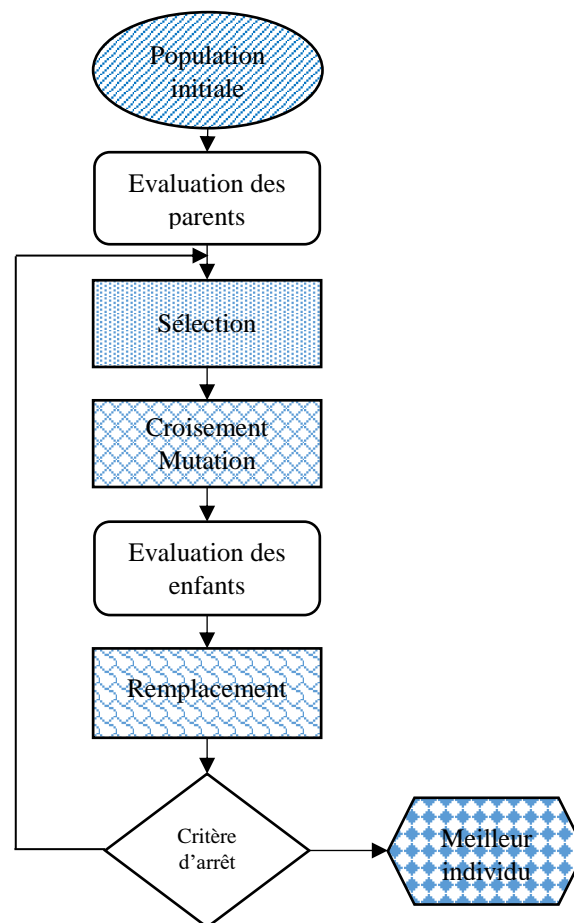


Figure 4.4. Schéma fonctionnel d'un Algorithme Génétique

La structure d'un algorithme génétique peut aussi être présentée sous forme de l'algorithme suivant :

Algorithme 4.1 : Algorithme génétique

Début

Initialiser la population

Evaluer les individus de la population

Répéter

Sélection

Reproduction des individus sélectionnés (croisement et mutation)

Evaluer les enfants

Remplacer certains parents par les enfants

Jusqu'à condition d'arrêt

Fin

4.3.2.1. Le codage

La première étape consiste à définir et à coder d'une manière convenable le problème. La représentation d'un individu sous une forme de chromosome est appelée codage [108]. Ce chromosome est formé d'un ensemble de gènes, à qui sont attribués des valeurs dans un alphabet qui dépend de la nature du codage en lui-même. Ceci permettra d'établir une liaison entre les individus de la population et la valeur de la variable, de façon à imiter la version génotype-phénotype existante dans le monde réel. Cette étape demande un grand soin, car l'efficacité des algorithmes génétiques dépend du choix du codage d'un chromosome. C'est la spécificité du problème traité qui conditionne le choix du codage des données. Différentes manières existent pour coder un chromosome qui dépend de l'alphabet utilisé. Les plus connus sont le codage binaire, le codage réel et un peu moins le codage de Gray.

4.3.2.1.1. Le codage binaire

Dans ce type de codage les gènes sont codés par des caractères binaires 0 ou 1 [108]. C'est le premier codage qui a été appliqué sur les algorithmes génétique, et c'est le plus couramment utilisé. Le codage binaire peut facilement être utilisé pour coder différents sortes d'objets (les entiers, les réels, les chaînes de caractères, etc.), il suffit juste d'utiliser des fonctions de codage et de décodage pour assurer le passage d'une représentation à une autre.

Chromosome A

1	0	1	1	0	1	0	1
---	---	---	---	---	---	---	---

Chromosome B

0	0	1	0	0	1	1	0
---	---	---	---	---	---	---	---

Figure 4.5. Codage binaire

4.3.2.1.2. Le codage réel

Dans certain cas, il est plus efficace de coder les chromosomes en utilisant un codage réel. Ainsi les gènes sont représentés par des nombres réels au lieu d'avoir une étape de transcodage (du réel vers le binaire et vice versa). Ce type de codage permet une plus grande efficacité et une évaluation plus rapide des chromosomes. En effet, les chromosomes codés en réel sont plus courts que ceux codés en binaires.

Chromosome			
7.9	1.6	0.5	2.3

Figure 4.6. Codage réel

4.3.2.2. La fonction objectif (Fitness)

Pour concrétiser au mieux le processus d'évolution, il est essentiel de pouvoir faire la distinction entre les chromosomes les moins adaptés et les plus adaptés. Une valeur d'adaptation est assignée à chaque chromosome pour distinguer les mieux adaptés à leur environnement des autres. Ceci permet de faire la comparaison entre les différents individus.

Souvent, la formulation de cette fonction spécifique à un problème est simple lorsque le nombre des paramètres n'est pas élevé. Par contre, définir cette fonction devient difficile lorsque le nombre de paramètre est élevé ou dans le cas où ils sont corrélés. La fonction peut devenir dans ce cas une somme avec pondération de plusieurs fonctions.

Mettre au point une bonne fonction objectif (fitness function) impose un respect de plusieurs critères qui se ramènent à satisfaire les contraintes du problème ainsi que sa complexité. S'il y a des contraintes à satisfaire et que les chromosomes issus des opérateurs de recherche codent les individus invalides, une solution parmi d'autres consiste à attribuer une mauvaise fitness à l'individu qui a violé les contraintes afin de favoriser la reproduction des individus valides.

4.3.2.3. Générer une population initiale

Une fois le codage établi, nous devons déterminer une population initiale formée d'un ensemble de solutions admissibles du problème d'optimisation. Cette étape conditionne fortement la rapidité de l'algorithme [109]. Cette population est générée d'une manière aléatoire dans le cas où la position du minimum (ou maximum) est inconnue dans l'espace de recherche, en effectuant des tirages uniformes dans tous les domaines associés aux composantes de l'espace et en respectant les contraintes sur les individus. Si par contre, il existe des informations a priori sur le problème, il paraît évident de générer les individus dans un sous domaine particulier, ce qui permettra d'accélérer la convergence.

Disposant d'une population non homogène, la diversité de la population doit être entretenue au cours des générations pour pouvoir parcourir le plus largement possible l'espace de recherche. C'est le rôle des opérateurs de reproduction.

Par ailleurs, à cette étape le choix de la taille de la population pose problème. En effet, une population trop petite conduit probablement à obtenir un optimum local peu intéressant, alors qu'une population trop grande augmentera excessivement le temps de calcul ainsi que l'espace

mémoire requis. Le choix de la taille de la population doit être effectué de telle sorte à avoir un bon compromis entre qualité des solutions et temps de calcul.

Mais il faut savoir aussi que la puissance de calcul dont nous disposons, les méthodes utilisées dans les opérateurs de reproduction, la fonction objectif et le nombre de variables considérées, affecte grandement la taille de la population. Si à titre d'exemple, une fonction que l'on cherche à optimiser comporte un optimum global clair et peu d'optimums locaux, une petite taille de la population sera suffisante, ce qui n'est pas le cas pour une fonction plus compliquée qui comporte un nombre élevé d'optimums locaux.

4.3.2.4. L'opérateur de sélection

L'opérateur de sélection permet l'identification statistique des individus d'une population courante qui auront l'autorisation à se reproduire. Ce qui offre aux individus sélectionnés une capacité à se diffuser et à persister dans la population. Cet opérateur est fondé sur la performance de chaque individu au sein de la population, estimée en utilisant la fitness. Cela interprète d'une manière approchée le concept de la sélection naturelle, où les gènes qui ont une performance relativement faible ont tendance à disparaître tandis que les gènes les plus performants tendent à se diffuser. Ce procédé va permettre de donner aux meilleurs individus, une probabilité plus élevée de contribuer aux générations futures. Donc la sélection est opérée à partir de la fonction objectif, après avoir effectué l'évaluation d'une génération de la population à un instant donné t .

De nombreuses techniques de sélection existent dans la littérature, les plus courantes vont être présentées dans les sous paragraphes suivants [110] :

4.3.2.4.1. Sélection par la roulette

Ce type de sélection est basé sur le principe de la roulette utilisée dans les casinos. On associe à chaque individu de la population un segment dans cette roulette. La fitness de l'individu détermine la largeur du segment associé d'une manière proportionnelle, donc la probabilité d'être sélectionné est proportionnelle à sa fitness. En tournant la roue et en faisant lancer la boule dans cette roulette, la case où la boule tombera correspond à l'individu sélectionné [111]. Les individus qui ont une meilleur fitness possèdent donc plus de chance d'être tirés au sort au déroulement de ce jeu. Ce système favorisera les grands segments, c'est-à-dire les bons individus seront plus souvent sollicités que les moins bons. Cependant, les individus les moins bons gardent toujours une chance d'être sélectionnés, pour maintenir une certaine diversité.

Si le nombre d'individus participants dans la population est égal à N , alors un individu ξ_j a une probabilité de sélection $P(\xi_j)$ égale à :

$$P(\xi_j) = \frac{F(\xi_j)}{\sum_{k=1}^N F(\xi_k)} \quad (4.6)$$

Avec $F(\xi_j)$ est la fonction objectif de l'individu ξ_j .

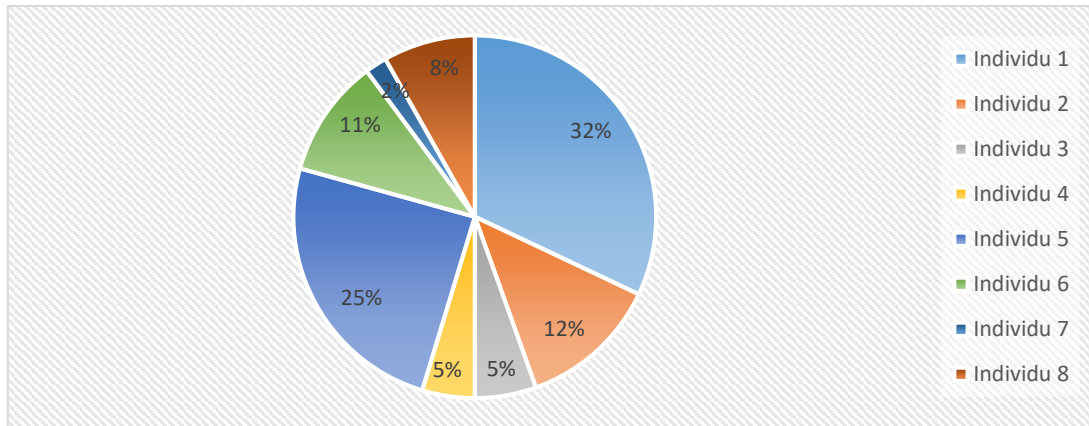


Figure 4.7. Sélection par roulette

4.3.2.4.2. Sélection par tournoi

Dans ce mode de sélection, la procédure se fait par un choix de deux ou de plusieurs individus au hasard qui vont se combattre (comparaison de leurs fitness) pour participer à la prochaine étape. Les individus de mauvaise qualité vis-à-vis à leurs fitness ont plus de chance de participer dans l'amélioration de la population.

En réalité c'est une compétition qui opposera les individus d'une sous-population prise au hasard dans la population, de taille égale ou inférieure à la taille de cette population et qui est fixée à priori par l'utilisateur. Le vainqueur dans cette compétition est considéré comme le meilleur individu dans la sous-population et sera donc sélectionné pour lui appliquer l'opérateur de croisement.

Plusieurs possibilités existent pour faire varier le mode de fonctionnement de la compétition, comme la variation du nombre d'individus qui devront s'affronter au départ, ou encore la permission ou non à un même individu d'être éligible plusieurs fois dans un même tournoi.

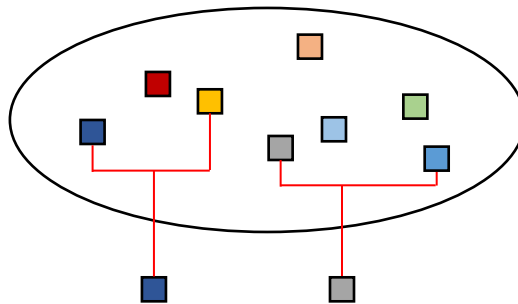


Figure 4.8. Sélection par tournoi

4.3.2.4.3. Sélection par rang

Dans cette sélection les individus de la population sont rangés soit en ordre croissant ou décroissant suivant l'objectif et par la suite ne retenir qu'un nombre fixe de chromosomes. Dans ce cas on conserve les individus les plus forts. Ce qui parfois peut représenter un inconvénient majeur pour cette méthode, parce qu'il est nécessaire parfois de garder quelques individus faibles pour avoir une diversité dans la population. Un autre problème dans la limite à fixer à

la sélection, ce qui peut empêcher parfois de garder de bons individus pour les prochaines générations.

4.3.2.5. L'opérateur de croisement

Pour créer de nouveaux individus, il est nécessaire de prendre aléatoirement une partie des chromosomes de chacun des deux parents et de les mélanger. Ce phénomène, inspiré de la nature est appelé croisement (crossover en anglais). Le but du croisement est d'enrichir la diversité de la population à travers la manipulation des structures des chromosomes. D'une manière classique, le croisement est appliqué en utilisant deux parents et génère deux enfants comme le montre la figure 4.9. Les couples d'individus sélectionnés subissent un croisement avec une probabilité P_c , un nombre aléatoire α est généré dans un intervalle $[0,1]$, et les individus subissent un croisement si et seulement si $\alpha \leq P_c$, sinon le couple procède sans croisement. Les valeurs typiques de P_c sont de 0,4 à 0,9. Si par exemple, $P_c=0,5$, alors la moitié de la population sera formée par sélection et croisement, et l'autre moitié par sélection seule [112] [109].

Initialement, le croisement à découpage de chromosome est associé au codage par chaînes de bits. Ce type de croisement est appliqué sur des chromosomes formés de M gènes, en tirant d'une façon aléatoire une position dans chacun des parents. Par la suite, on effectue un changement des deux sous-chaînes terminales par rapport à cette position de chacun des deux chromosomes, produisant ainsi deux enfants. Ce principe peut être étendu non seulement à un découpage du chromosome en deux sous-chaînes mais en trois, quatre, etc. le croisement à découpage de chromosome est très pratique pour les problèmes discrets.

En présence d'un problème continu, un croisement « barycentrique » est souvent sollicité. Dans ce cas deux gènes $P_1(j)$ et $P_2(j)$ sont sélectionnés à la même position j dans chacun des parents. Ils vont définir deux nouveaux gènes par pondération, en créant deux nouveaux points sur la droite reliant ces derniers. Les enfants $E_1(j)$ et $E_2(j)$ sont créés de la manière suivante [92]:

$$\begin{cases} E_1(j) = \alpha P_1(j) + (1 - \alpha)P_2(j) \\ E_2(j) = (1 - \alpha)P_1(j) + \alpha P_2(j) \end{cases} \quad (4.7)$$

Avec α un coefficient de pondération aléatoire adapté au domaine d'extension des gènes (pas forcément compris entre 0 et 1).

Ce type de croisement permet de générer des points à l'extérieur, ou entre les deux gènes considérés.

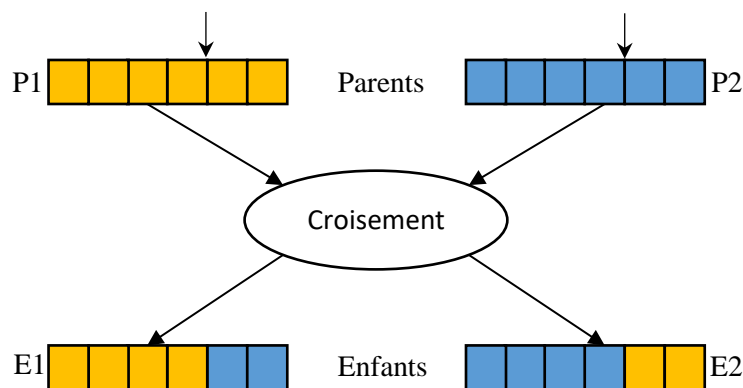


Figure 4.9. Croisement en un seul point

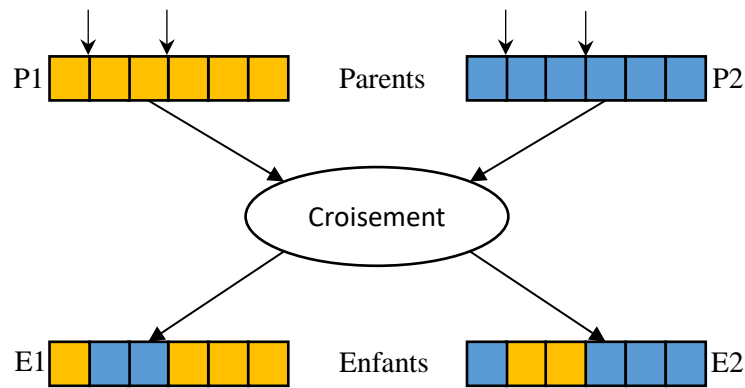


Figure 4.10. Croisement en deux points

4.3.2.6. L'opérateur de mutation

La mutation est un autre opérateur utilisé dans les algorithmes génétiques. Il s'agit de l'altération occasionnelle de la valeur d'un chromosome avec une certaine probabilité p_m [111]. Même si d'autres types de mutations existent dans la littérature, la plus couramment utilisée est la mutation de bits. Dans la mutation de bits, chaque bit de la chaîne est muté avec une probabilité p_m . Pour une chaîne codée en binaire, la mutation est définie comme la conversion de 1 en 0, et vice versa, tel qu'il est montré sur la figure 4.11.

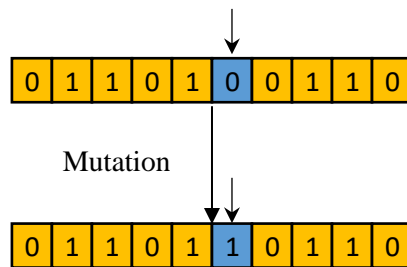


Figure 4.11. Une mutation dans un chromosome

La mutation est un opérateur très important, car elle apporte aux algorithmes génétiques l'aléa essentiel à une exploration efficace. Elle garantit aussi que l'algorithme génétique sera capable d'atteindre la majorité des points du domaine de recherche. C'est aussi une assurance contre une convergence prématurée, en évitant les optimums locaux par la diversification qu'elle donne. Sans mutation, des chaînes similaires sont traitées à chaque génération, ce qui peut provoquer une convergence vers des optimums locaux.

L'opérateur de mutation intervient généralement avec une probabilité fixée assez faible (de l'ordre de 1%) [110]. Si la probabilité de mutation est notée p_m , et r_m un nombre tiré au hasard tel que $r_m \in [0,1]$, et que $r_m \leq p_m$ alors l'individu subira une mutation. D'autre variante existe aussi où la mutation peut être appliquée sur plusieurs bits dans un même chromosome.

4.3.2.7. L'opérateur d'élitisme

En créant une nouvelle population, il y a un grand risque que les meilleurs individus soient perdus après l'application des deux opérateurs de reproduction. La méthode d'élitisme permet d'éviter cela, en copiant un ou plusieurs des meilleurs individus dans la nouvelle génération. Par la suite le reste de la population sera généré suivant l'algorithme de reproduction. Cet

opérateur permet d'améliorer considérablement l'algorithme génétique, car les meilleures solutions seront gardées [110].

4.3.2.8. Test d'arrêt

Le test d'arrêt joue un rôle primordial dans le jugement de la qualité des individus. Son but est de nous assurer l'optimalité, de la solution finale obtenue par l'algorithme génétique.

Les critères d'arrêts sont de deux natures :

1. Arrêt après un nombre fixé a priori de générations. C'est la solution retenue lorsqu'une durée maximale de temps de calcul est imposée.
2. Arrêt lorsque la population cesse d'évoluer ou n'évolue plus suffisamment. Nous sommes alors en présence d'une population homogène dont on peut penser qu'elle se situe à la proximité de l'optimum. Ce test d'arrêt reste le plus objectif et le plus utilisé.

Il est à noter qu'aucune certitude concernant la bonne convergence de l'algorithme n'est assurée. Comme dans toute procédure d'optimisation l'arrêt est arbitraire, et la solution en temps fini ne constitue qu'une approximation de l'optimum.

4.3.3. Paramètres de l'algorithme génétique

Plusieurs paramètres peuvent conditionner la convergence des algorithmes génétiques :

- La taille de la population ;
- Le nombre max de génération que doit faire l'algorithme ;
- La probabilité de croisement ;
- La probabilité de mutation.

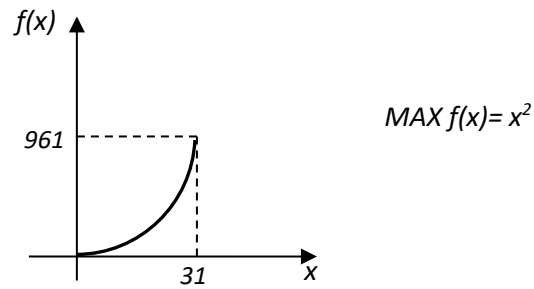
L'ajustement de ces paramètres dépend fortement du problème étudié. Donc il n'y a pas de paramètres valables pour résoudre tous les problèmes posés. Cependant, il existe des valeurs qui sont souvent employées (définies par la littérature) et peuvent être utilisés comme un point départ pour lancer la recherche de solutions.

- La valeur de la probabilité de croisement est choisie dans un intervalle [0.4,0.9] ;
- La valeur de la probabilité de mutation est choisie dans un intervalle [0.001,0.1]

Trouver des valeurs à ces paramètres qui donnent de bons résultats est parfois une tâche délicate.

4.3.4. Exemple de résolution à base d'algorithme génétique [113]

Afin de présenter les étapes à suivre dans un algorithme génétique classique, nous allons reprendre l'exemple cité par Goldberg [8]. Il consiste à trouver le maximum de la fonction $f(x) = x^2$, pour une plage de $x = 0$ jusqu'à $x = 31$.

Figure 4.12. Représentation de la fonction $f(x)$

Comme première étape nous allons générer au hasard une population de n chromosome. Nous avons choisi de coder l'information en utilisant cinq gènes de 0 ou 1. Nous procédons ensuite à l'évaluation des performances de chaque chromosome en calculant la valeur de $f(x)$. La taille de la population est choisie, $n=6$. Le résultat obtenu de la première génération est montré au tableau suivant (tableau 4.1).

chromosome					x	$f(x)=x^2$	n
0	1	1	0	0	12	144	1
1	0	0	1	1	19	361	2
1	0	0	1	1	19	361	3
0	1	0	0	0	8	64	4
1	0	0	0	1	17	289	5
1	1	0	0	0	24	576	6

Tableau 4.1. Choix aléatoire des individus

Deux chromosomes sont sélectionnés d'une façon aléatoire, pour comparer leurs performances. Nous conservons l'individu le plus performant des deux. Nous répétons cette sélection par compétition dans le but de créer la nouvelle population qui participera à la prochaine génération (Tableau 4.2).

chromosome					x	$f(x)=x^2$	comparaison	n
1	0	0	1	1	19	361	2-1	1
1	1	0	0	0	24	576	6-5	2
0	1	1	0	0	12	144	1-4	3
1	0	0	1	1	19	361	3-4	4
1	0	0	0	1	17	289	5-4	5
1	0	0	0	1	17	289	5-1	6

Tableau 4.2. Résultats obtenus d'une sélection par tournoi

Alors deux chromosomes vont subir un croisement. Nous tirons aléatoirement deux parents $Pn1$ et $Pn2$. Au hasard, nous choisissons les points de croisement, ainsi les groupes de gènes à croiser et nous les accouplons. Ceci produit deux nouveaux enfants, $En1$ et $En2$.

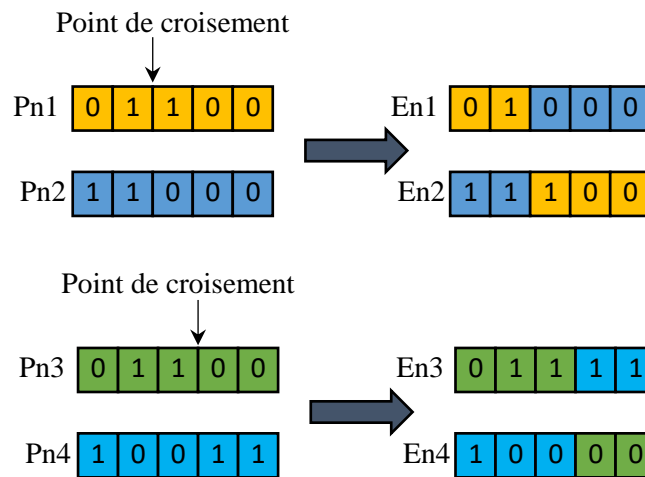


Figure 4.12. Le croisement

La mutation implique un changement aléatoire d'un gène d'un chromosome. Ceci implique un changement de 0 à 1 d'un gène choisi de façon aléatoire.

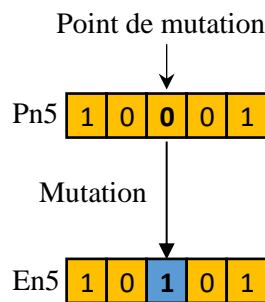


Figure 4.13. La mutation

L'élitisme correspond à transmettre l'individu le plus performant à la prochaine génération. Dans cet exemple, l'individu à n=2 est le plus performant alors, il est transmis directement à la prochaine génération. Maintenant, nous devons refaire une nouvelle sélection par compétition avec la nouvelle génération de solution. Les deux prochains tableaux démontrent respectivement le résultat de la première génération et le résultat d'une re-sélection par compétition.

chromosome					x	$f(x)=x^2$	n
0	1	0	0	0	8	64	1
1	1	1	0	0	28	784	2
0	1	1	1	1	15	225	3
0	1	0	1	1	11	121	4
1	0	0	0	0	16	256	5
1	1	0	0	0	24	576	6

Tableau 4.3. Individus de la première génération

chromosome					x	$f(x)=x^2$	comparaison	n
1	1	1	0	0	28	784	2-1	1
1	1	0	0	0	24	576	6-4	2
1	1	1	0	0	28	784	2-4	3
0	1	1	1	1	15	225	3-4	4
1	1	1	0	0	28	784	2-3	5
1	0	0	0	0	16	256	5-1	6

Tableau 4.4. Re-sélection par tournoi

Nous accouplons aléatoirement les chromosomes Pn4 à Pn6 et Pn2 à Pn5.

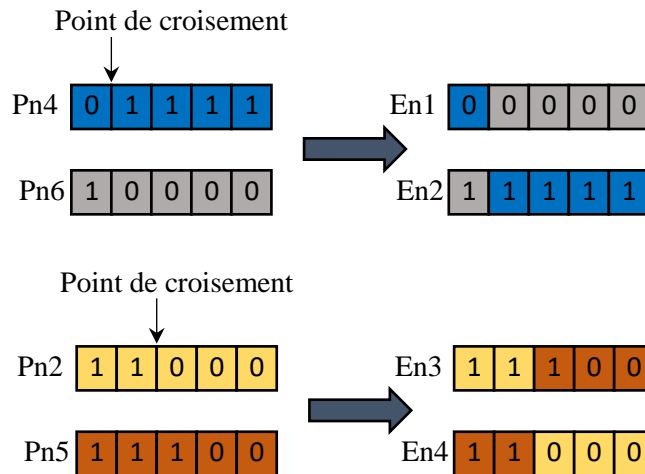


Figure 4.14. Le croisement

Nous effectuons alors la mutation au parent Pn3.

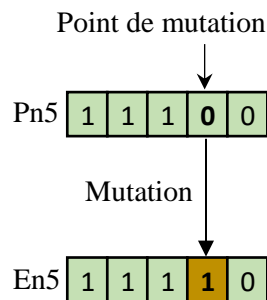


Figure 4.15. La mutation

L'individu le plus performant est Pn1, ce chromosome élite sera directement transmis à la prochaine génération. Le tableau 4.5 démontre le résultat de la sélection par tournoi effectuée pour générer la deuxième génération. Nous pouvons voir que les performances de chaque individu augmentent. C'est-à-dire, les plus faibles seront éliminés. Nous nous approchons de plus en plus vers la valeur maximale de x . Les chromosomes de la prochaine itération auront sûrement tous convergé vers la valeur maximale de 31.

chromosome					x	$f(x)=x^2$	comparaison	n
1	1	1	1	1	31	961	2-1	1
1	1	1	0	0	28	784	6-4	2
1	1	1	1	1	31	961	2-4	3
1	1	1	0	0	28	784	3-4	4
1	1	1	1	1	31	961	2-3	5
1	1	1	1	0	30	900	5-4	6

Tableau 4.5. Sélection par tournoi

Cet exemple nous a permis de montrer le concept de base des différentes étapes, notamment la sélection par tournoi, le croisement, la mutation et l'élitisme, de l'algorithme génétique classique.

4.4. Optimisation par Essaim de particules (PSO)

L'algorithme d'optimisation par l'essaim de particules (en anglais PSO pour Particle Swarm Optimization), est une technique d'optimisation stochastique inspirée en analogie avec le comportement des animaux lorsqu'ils se déplacent en collectif. Elle a été proposée pour la première fois par l'ingénieur en Electricité Russell Eberhart et le socio-psychologue James Kennedy en 1995 [9]. Au début Eberhart et Kennedy cherchaient à simuler le comportement des oiseaux et leurs capacités à voler d'une manière synchrone ainsi que leurs aptitudes à varier brusquement de direction tout en gardant une formation optimale, c'est ce modèle qui fut étendu par la suite à un algorithme d'optimisation.

Le PSO est basé sur le comportement social des animaux lorsqu'ils évoluent en essaim, tels que les vols des nuées d'oiseaux et les bancs de poissons. Les insectes, les poissons, les animaux, en particulier les oiseaux, etc., voyagent toujours en groupe de membres en ajustant leurs positions et vitesses à leurs informations de groupe (voir figure 4.16). Cette méthode réduit leur effort individuel pour rechercher la nourriture, un abri, ou même éviter un prédateur, etc. [114] [115]. Dans cet algorithme, la population est appelée l'essaim et chacun de ses éléments la particule.

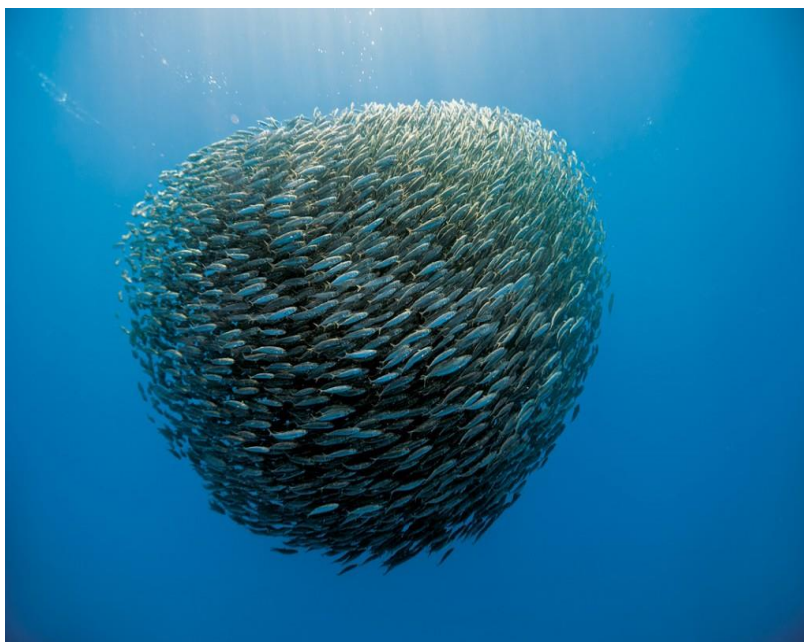


Figure 4.16. Exemple d'essaim de poissons

4.4.1. Origine et inspiration du PSO

L'idée principale de cet algorithme, s'appuie sur les travaux de Reynolds [116] et Heppner et Grenander [117], qui ont présentés des simulations de vol d'oiseaux. Reynolds était intrigué par l'esthétique de la chorégraphie d'une nuée d'oiseaux, et Heppner, zoologiste, souhaitait découvrir les règles sous-jacentes qui permettaient à un grand nombre d'oiseaux de se rassembler de manière synchrone, changeant souvent d'une manière soudaine de direction, se dispersant et se regroupant, etc. [9]. Les modèles proposés dans ces travaux reposaient fortement sur la manipulation des distances interindividuelles; c'est-à-dire que la synchronisation du comportement des essaims était considérée comme une fonction des efforts des oiseaux pour maintenir une distance optimale entre eux et leurs voisins [9].

Dans une nuée d'oiseaux en vol, pour garantir la stabilité de l'essaim, les différents agents obéissent à trois règles locales [89]:

- Cohésion : se déplacer vers le centre des voisins pour se maintenir dans l'essaim.
- Séparation : si les voisins sont trop proches, il faut s'éloigner.
- Alignement : le déplacement se fait dans la même direction que l'essaim en moyennant les directions et les vitesses des voisins.

Les trois règles citées ci-dessus, permettent la répulsion et l'attraction de chacun des individus et ainsi maintenir la cohésion de l'essaim.

Le PSO correspond donc à une population constituée d'agents simples, qu'on appelle particules. Toute particule appartenant à cette population est considérée comme une solution possible du problème, possédant une position et une vitesse. De plus, toutes les particules possèdent une mémoire, permettant ainsi à chacune de se souvenir de la meilleure performance atteinte (en position et en valeur) et de la meilleure performance réalisée par les autres particules (voisines) «informatrices» : en effet, chaque particule dispose d'un groupe de particules informatrices, appelé historiquement son voisinage.

4.4.2. Topologies de voisinage

Le voisinage représente la structure du réseau social. La topologie du voisinage définit le réseau de communication de chaque particule. Plusieurs topologies de voisinage ont été proposées [118] et sont considérées en fonction des identificateurs des particules et non des informations topologiques comme les distances euclidiennes dans l'espace de recherche, les plus connus sont (voir figure 4.17) :

- **Topologie en anneau ou *lbest***: cette topologie est basée sur un voisinage local. Dans cette approche, chaque particule ne partage les informations qu'avec les n voisins directs. Chaque particule tend dans son voisinage local à se déplacer vers la meilleure solution notée *lbest*. La probabilité de localiser l'optimum global est plus importante, mais la convergence de l'algorithme est lente [89].
- **Topologie en étoile ou *gbest*** : les particules dans cette topologie partagent les informations en utilisant une structure entièrement connectée. Ainsi, chaque particule peut recevoir les informations par la totalité des particules. Un mécanisme de voisinage global est utilisé dans ce cas, où la meilleure position trouvée par n'importe quelle particule de la population va influencer la trajectoire de recherche de chaque membre [89]. Avec cette topologie, toutes les particules sont attirées au même temps vers la meilleure solution ce qui conduit à une convergence rapide du PSO. Par contre, si

l'optimum global est loin de la meilleure particule, l'exploration n'est pas suffisante et l'algorithme risque de stagner dans un optimum local [88].

- **Topologie en rayon:** toutes les particules sont connectées à une seule particule centrale. Cette particule centrale peut ajuster sa position vers la meilleure, si cela donne une amélioration, l'information est propagée aux autres.

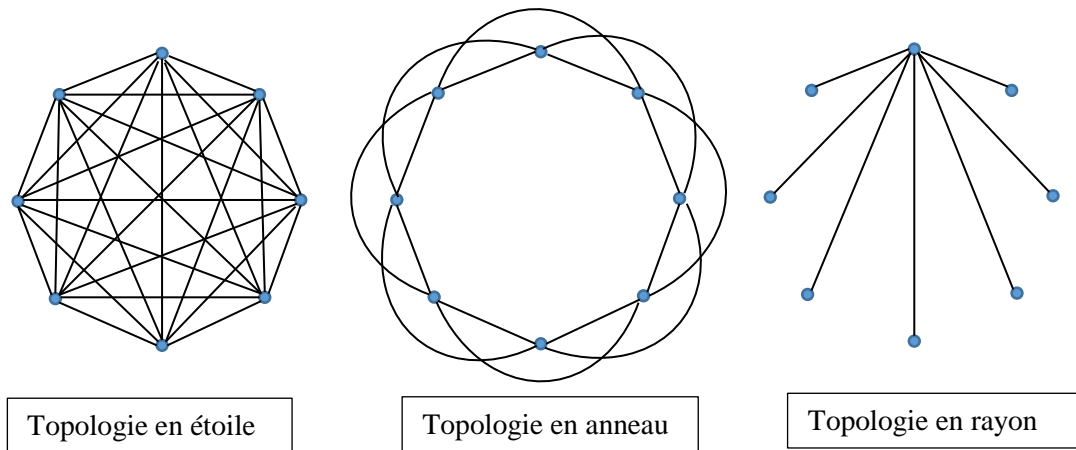


Figure 4.17. Différents types de topologie de voisinage

D'autres topologies ont été proposées dans la littérature, afin d'augmenter les performances du PSO [118] [88]. Ces topologies peuvent être caractérisées par rapport à deux facteurs, le premier est le degré de connectivité K représentant le nombre de voisins d'une particule, et le deuxième est le nombre de voisins d'une particule faisant partie également à d'autres voisinages. Parmi ces topologies nous pouvons citer : Von Neumann, Four-clusters, arbre, etc. [88].

Le voisinage géographique n'est pas nécessairement pertinent, malgré que ce soit le premier auquel nous sommes amenés à penser, parce que d'une part c'est un voisinage trop local, et d'autre part parce que la socialisation des particules rend tout voisinage social en voisinage géographique. En plus, en matière de calcul c'est un voisinage très lourd car à chaque itération on doit recalculer le voisinage de chaque particule.

4.4.3. Principe de fonctionnement de l'algorithme du PSO

Le principe sur lequel l'algorithme du PSO est fondé ressemble à celui d'un essaim d'oiseaux, d'une façon particulière du fait qu'un oiseau possédant une certaine capacité de mémorisation et d'analyse, lorsqu'il capte un site intéressant (source d'eau, source de nourriture, etc.), il passe cette information à certain de ces voisins qui vont la prendre en considération pour leur prochain déplacement. En effet, chaque oiseau possède individuellement une connaissance locale et une intelligence limitée. Donc, les différentes interactions entre les oiseaux qui forment l'essaim vont créer une intelligence globale. La somme des performances de l'ensemble des oiseaux est inférieure à la performance de l'essaim dont ils font partie [119].

Les particules dans un essaim qui représentent potentiellement des solutions au problème d'optimisation, vont parcourir l'espace de recherche afin de trouver l'optimum global. Au départ de l'algorithme, pour chaque particule la position est fixée au hasard dans l'espace de recherche, la vitesse prend une valeur aléatoire, elle est dotée d'une mémoire qui va lui

permettre de mémoriser le meilleur point par lequel elle est déjà passée, et également d'un ensemble de voisines informatrices (le voisinage) pour être en courant du meilleur point atteint de son voisinage. À chaque pas, chaque particule peut faire:

- L'évaluation de la qualité sa position actuelle (fitness) qui interprète la distance à l'optimum global et mémoriser la meilleure performance atteinte jusqu'ici.
- Recevoir des informations des particules voisines et ainsi obtenir la meilleure performance de chacune d'entre elles.
- Le choix de la meilleure parmi les meilleures performances dont elle a connaissance.

Donc trois composantes peuvent influencer le déplacement d'une particule [88]:

1. *Une composante d'inertie* : la particule à tendance à suivre sa propre direction de déplacement.
2. *Une composante cognitive* : la particule est attirée par sa propre expérience et à tendance à se diriger vers la meilleure position par laquelle elle est déjà passée.
3. *Une composante sociale* : la particule est attirée par l'expérience de ses paires et à tendance à se diriger vers la meilleure position atteinte par ses voisins.

4.4.3.1. Formalisation

Dans le cas d'un problème d'optimisation la performance de chaque particule appartenant à l'essaim est mesurée par une fonction objectif notée dans notre cas f .

Dans un espace de recherche de dimension N , chaque particule P_k est exprimée en termes des caractéristiques suivantes [89] :

- Vecteur position $\vec{x}_k = [x_{k1}, x_{k2}, \dots, x_{kN}]$
- Vecteur vitesse $\vec{v}_k = [v_{k1}, v_{k2}, \dots, v_{kN}]$
- La meilleure position visitée par la particule k

$$\begin{aligned} \overrightarrow{Pbest}_k &= [Pbest_{k1}, Pbest_{k2}, \dots, Pbest_{kN}], \text{ à l'itération } i \text{ elle est calculée comme suit :} \\ \overrightarrow{Pbest}_k(i) &= \begin{cases} \overrightarrow{Pbest}_k(i-1) & \text{si } f(\vec{x}_k(i)) \geq f(\overrightarrow{Pbest}_k(i-1)) \\ \vec{x}_k(i) & \text{sinon} \end{cases} \end{aligned} \quad (4.8)$$

- La meilleure position du voisinage de la particule k

$$\begin{aligned} \overrightarrow{Gbest}_k &= [Gbest_{k1}, Gbest_{k2}, \dots, Gbest_{kN}], \text{ et calculée comme suit :} \\ \overrightarrow{Gbest}_k(i) &= \arg \min_{\overrightarrow{Pbest}_k} f(\overrightarrow{Pbest}_k(i)), \quad 1 \leq k \leq L \end{aligned} \quad (4.9)$$

Avec L qui représente le nombre de particule dans l'essaim.

Au cours du processus d'optimisation chaque particule de l'essaim change sa position dans l'espace de recherche en fonction de sa dernière position, sa dernière vitesse, sa meilleure position trouvée au cours des itérations passées $Pbest$ et la meilleure position trouvée par l'essaim $Gbest$, à chaque itération la vitesse et la position sont mises à jour en utilisant les équations suivantes [120] [115] :

$$v_k(i) = w * v_k(i-1) + r_1 \varphi_1 * (Pbest_k - x_k(i-1)) + r_2 \varphi_2 * (Gbest - x_k(i-1)) \quad (4.10)$$

$$x_k(i) = x_k(i-1) + v_k(i) \quad (4.11)$$

Où w est une constante, nommée coefficient d'inertie, r_1 et r_2 sont des nombres aléatoires compris entre 0 et 1. Les coefficients d'accélération sont notés φ_1 et φ_2 et sont des constantes

positives, généralement comprises entre 0 et 2. En se basant sur ces relations le principe de déplacement d'une particule est illustré par la figure 4.18.

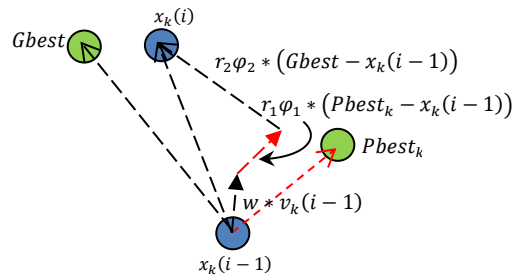


Figure 4.18. Principe de déplacement d'une particule

$w * v_k(i-1)$ correspond à la composante d'inertie introduite par Shi et Eberhart [121]. Le paramètre w contrôle l'impact des directions de déplacement sur les déplacements futurs. Une petite valeur de w favorise une recherche locale (l'exploitation), tandis qu'une grande valeur favorise une recherche globale (l'exploration).

$r_1 \varphi_1 * (Pbest_k - x_k(i-1))$ correspond à la composante cognitive, qui représente l'attraction linéaire de la particule k vers la meilleure position déjà trouvée.

$r_2 \varphi_2 * (Gbest - x_k(i-1))$ correspond à la composante sociale, qui représente l'attraction linéaire de la particule k vers la meilleure position trouvée par le voisinage de cette particule.

La structure d'un algorithme PSO présentée ci-dessous peut aussi être représentée sous forme de l'organigramme présenté dans la figure 4.19.

Algorithme 4.2 : PSO

Début

Initialiser aléatoirement les positions et les vitesses de chaque particule

Pour chaque particule P_k , $Pbest_k = x_k$

Tant que le critère d'arrêt n'est pas satisfait **faire**

Pour $k=1$ à L **faire**

Déplacement de la particule P_k selon eq 4.10 et eq 4.11

Evaluer les position

Si $f(x_k) < f(Pbest_k)$ alors

 | $Pbest_k = x_k$

Fin si

Si $f(Pbest_k) < f(Gbest)$ alors

 | $Gbest = Pbest_k$

Fin si

Fin pour

Fin Tant que

Fin

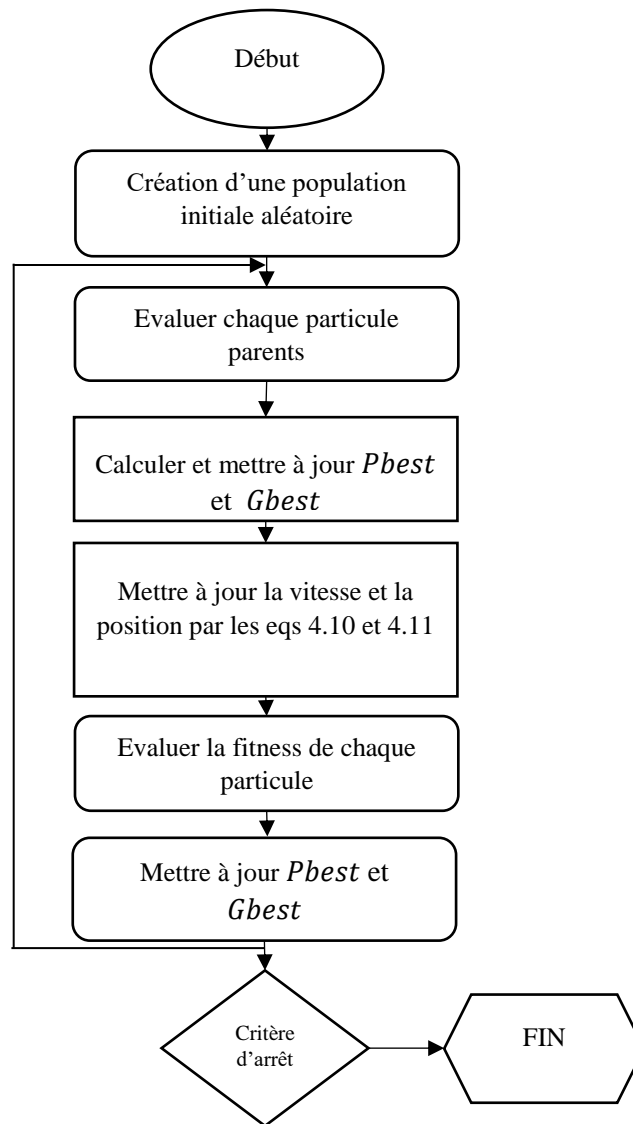


Figure 4.19. Organigramme d'un algorithme PSO [115]

4.4.3.2. Choix des paramètres de l'algorithme

Les paramètres de l'algorithme PSO ont une grande influence sur la conduite des particules et donc sur la convergence de l'algorithme ; et même si des résultats satisfaisants sont obtenus par cette méthode, un bon choix des paramètres de la méthode demeure comme un point critique pour s'assurer du succès de l'algorithme PSO.

4.4.3.2.1. Nombre de particules L

Le nombre de particules réservées pour résoudre un problème d'optimisation est dépendant principalement de la taille de l'espace de recherche. Il n'existe pas de règle fixe pour la détermination de ce paramètre, pour avoir une bonne valeur il faut faire plusieurs essais.

Généralement, augmenter le nombre de particules va augmenter la diversité, ce qui minimise le risque d'être orienter vers des minimums locaux [89]. La valeur attribuée à L ne doit pas être trop petite selon [121].

4.4.3.2.2. L'initialisation de l'algorithme

L'initialisation des positions des particules est aléatoire, mais d'une manière à couvrir tout l'espace de recherche. Il faut noter que l'algorithme du PSO peut trouver des difficultés pour localiser l'optimum, si les positions des particules ne couvrent pas initialement tout l'espace de recherche, de risque que cet optimum se trouve à l'extérieur de la zone initiale.

4.4.3.2.3. Coefficient d'inertie w

Le coefficient d'inertie w a été introduit par Shi et Eberhart [121] pour contrôler l'influence des directions actuelles des particules de l'essaim sur les déplacements futur de ces derniers. Ce paramètre permet d'avoir un compromis entre l'exploration (la recherche globale) et l'exploitation (la recherche locale). Une petite valeur du coefficient d'inertie mène à une recherche locale sur un espace réduit tandis qu'une grande valeur provoque une exploration étendue de l'espace de recherche [88].

Les meilleures valeurs de w pour avoir une meilleure convergence selon l'étude de [122] sont dans l'intervalle $[0.8, 1.2]$. Au-delà des valeurs indiquées dans l'intervalle l'algorithme peut avoir des difficultés pour converger.

4.4.3.2.4. Coefficients d'accélération φ_1 et φ_2

φ_1 et φ_2 sont deux constantes positives, qui contrôlent l'attraction vers la meilleure position de soi-même et l'attraction vers la meilleure position globale respectivement. D'une manière empirique elles sont déterminées suivant la relation $\varphi_1 + \varphi_2 \leq 4$. Combiner les trois coefficients w , φ_1 et φ_2 permet de régler la balance entre une recherche globale et locale.

4.4.3.2.5. Critère d'arrêt

Différents critères d'arrêts existent pour terminer l'exécution de l'algorithme PSO [89] :

- **Convergence vers la solution recherchée** : un seuil ε est souvent utilisé, si la valeur de ε est trop petite la recherche est lente, si par contre elle est trop grande les solutions trouvées sont mauvaises.
- **Nombre maximum d'itération** : fixer un nombre maximum d'itération à l'avance. Si ce nombre est petit les solutions générées par l'algorithme ne sont pas optimales.
- **Aucune amélioration/nombre d'itération** : plusieurs façons existent pour savoir si la recherche s'améliore ou pas, à titre d'exemple la variation des positions des particules est trop petite ou la variation de la vitesse est proche de zéro.

4.5. Conclusion

A travers ce chapitre, quelques méthodes d'optimisation ont été présentées en se basant sur les caractéristiques fondamentales des métaheuristiques. Ces dernières sont réputées d'être efficace dans la résolution des problèmes d'optimisation difficiles sans avoir recours à la modification de l'architecture de base de l'algorithme utilisé. Ces dernières années elles sont devenues très sollicitées par les scientifiques à cause de leurs simplicités d'emploi dans divers domaines. Malgré leur succès remarquable, les métaheuristiques souffrent néanmoins de quelques difficultés auxquelles sont confrontés les utilisateurs dans le cas d'un problème concret, tels que le choix d'une approche efficace pour trouver la solution optimale et le réglage des paramètres. Parmi les métaheuristiques qui existent dans la littérature notre intérêt s'est orienter

vers deux méthodes à savoir les algorithmes génétiques et l'optimisation par l'essaim de particule (PSO).

L'algorithme génétique est une technique d'intelligence artificielle permettant la résolution des problèmes d'optimisation par une approche itérative qui impose une série de solutions aléatoires se rapprochant d'une manière progressive de la solution réelle du problème. Cette technique est basée sur le principe de l'évolution naturelle énoncé par Darwin, c'est-à-dire la capacité des espèces à survivre et à s'adapter à leur environnement par des processus naturels comme le croisement et la mutation. Mais les algorithmes génétiques ont un désavantage du point de vue temps de calcul pour les problèmes ayant des fonctions objectifs complexes à plusieurs variables. Il est parfois nécessaire d'utiliser une population d'individu assez grande et un nombre important de générations afin de s'assurer que la solution trouvée est optimale à cause de la nature aléatoire de l'algorithme.

Malgré sa récente apparition, l'algorithme PSO inspiré particulièrement du comportement social des animaux qui évoluent en essaim, a connu un intérêt croissant ces dernières années. Ceci est dû aux avantages qu'offre cet algorithme : une excellente robustesse, une facilité dans la réalisation d'un calcul parallèle, une convergence rapide vers la valeur optimale et peut s'hybrider avec d'autres techniques pour accroître ses performances.

CHAPITRE 5

COMMANDES NON LINÉAIRES ROBUSTES OPTIMALES APPLIQUÉES AU ROBOT MANIPULATEUR

5.1. Introduction

Les robots manipulateurs sont caractérisés par un modèle mathématique non linéaire fortement couplé avec une sensibilité aux variations paramétriques. Le choix d'une structure de commande qui pourrait être construite à base de ce modèle, devra garantir une certaine robustesse vis-à-vis des perturbations et une faible sensibilité aux erreurs de modélisation et aux incertitudes paramétriques ainsi que leurs variations [10] [11] [19].

Une synthèse bibliographique a été présentée dans le chapitre précédent sur deux lois de commande qualifiées de non linéaires et robustes vis-à-vis les perturbations, à savoir le mode glissant et le backstepping [1] [45]. Le développement de ces deux dernières commandes et leurs applications par la suite au robot manipulateur fera l'objet de ce chapitre, tout en présentant les différents résultats de simulation obtenus.

5.2. Synthèse de la loi de commande par mode de glissement

Le principal objectif de cette approche est d'utiliser la commande par mode de glissement dans le but de proposer une loi de commande permettant de forcer les différentes variables articulaires à suivre leurs trajectoires de références, tout en assurant une robustesse vis-à-vis des perturbations.

La conception d'une telle commande est basée sur le choix d'une surface de glissement dont les racines du polynôme caractéristique sont situées à gauche du demi-plan dans le plan complexe.

En considérant un robot manipulateur à deux degrés de libertés (2ddl). Le modèle dérivé en utilisant le formalisme d'Euler-Lagrange tel qu'il est montré dans le premier chapitre, est donné par :

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (5.1)$$

Avec : q, \dot{q} et \ddot{q} représentent respectivement les vecteurs (de dimension 2) des positions articulaires, vitesses articulaires et accélérations articulaires. $M(q)$ est la matrice d'inertie de dimension (2×2) , $C(q, \dot{q})$ matrice de dimension (2×2) exprimant les forces Centrifuges et Coriolis, $g(q)$ le vecteur de gravité et τ représente le vecteur des couples appliqués sur chacune des articulations.

Le modèle dynamique du robot manipulateur donné par l'équation (5.1) peut être représenté par la forme suivante :

$$\ddot{q} = M(q)^{-1}(-C(q, \dot{q})\dot{q} - g(q) + \tau) \quad (5.2)$$

5.2.1. Calcul du degré relatif

Le modèle du robot manipulateur peut être représenté par un modèle d'état non linéaire de la forme :

$$\begin{cases} \dot{\chi} = f(\chi) + g_l(\chi) u \\ y = h(\chi) \end{cases} \quad (5.3)$$

En posant $\chi_1 = q$, $\chi_2 = \dot{q}$ et $\chi = (\chi_1^T \ \chi_2^T)^T$, nous avons :

$$\begin{cases} \dot{\chi}_1 = \chi_2 \\ \dot{\chi}_2 = \ddot{q} \end{cases} \quad (5.4)$$

En substituant (5.2) dans (5.4), nous obtenons :

$$\begin{cases} \dot{\chi} = \begin{pmatrix} \chi_2 \\ M^{-1}(\chi_1) \left(-C(\chi_1, \chi_2)\chi_2 - g(\chi_1) \right) \end{pmatrix} + \begin{pmatrix} 0 \\ M^{-1}(\chi_1) \end{pmatrix} u \\ y = \chi_1 \end{cases} \quad (5.5)$$

Par identification avec l'équation (5.3), nous aurons :

$$\begin{aligned} f(\chi) &= \begin{pmatrix} \chi_2 \\ M^{-1}(\chi_1) \left(-C(\chi_1, \chi_2)\chi_2 - g(\chi_1) \right) \end{pmatrix} \\ g_l(\chi) &= \begin{pmatrix} 0 \\ M^{-1}(\chi_1) \end{pmatrix} \\ h(\chi) &= \chi_1 \end{aligned} \quad (5.6)$$

Pour un système non linéaire le degré relatif sur un domaine U est définie comme étant le plus petit entier noté généralement r pour lequel :

$$L_g L_f^{r-1} h(\chi) \neq 0 \quad \text{pour tout } \chi \in U \quad (5.7)$$

Sachant que $L_g h(\chi)$ représente la dérivée de Lie de $h(\chi)$ par rapport à $g(\chi)$. Dans le cas des systèmes linéaires la dérivée de Lie devient alors $L_g L_f^{r-1} h(\chi) = C A^{r-1} B \neq 0$.

En considérant le système non linéaire décrit par l'équation (5.3), appelé système affine en entrée.

- *Dérivées de Lie* : la dérivée de Lie de la fonction scalaire $h(\chi) : \mathbb{R}^n \rightarrow \mathbb{R}$, le long du champ de vecteur $f(\chi)$, notée $L_f h(\chi)$ est donnée par :

$$L_f h(\chi) = \frac{\partial h}{\partial \chi} \cdot f(\chi) = \nabla h(\chi) \cdot f(\chi) \quad (5.8)$$

La dérivée de Lie d'ordre supérieur :

$$\begin{cases} L_f^0 h(\chi) = h(\chi) \\ L_f^r h(\chi) = L_f \left(L_f^{r-1} h(\chi) \right) \neq 0, \quad r = 1, 2 \dots n \end{cases} \quad (5.9)$$

En utilisant la dérivée de Lie, nous avons :

$$L_g L_f^i h(\chi) = 0 \quad \text{avec} \quad i=0 \dots r-1 \quad (5.10)$$

- Pour $i=0$

$$L_g h(\chi) = \left[\frac{\partial h(\chi)}{\partial \chi_1} \quad \frac{\partial h(\chi)}{\partial \chi_2} \right] g_l(\chi) = 0 \quad (5.11)$$

- Pour $i=1$

$$L_g L_f^1 h(\chi) = L_g \left(\left[\frac{\partial h(\chi)}{\partial \chi_1} \quad \frac{\partial h(\chi)}{\partial \chi_2} \right] f(\chi) \right) = M^{-1}(\chi_1) \quad (5.12)$$

Avec $M^{-1}(\chi_1) \neq 0$, d'où :

$$r-1=1 \Rightarrow r=2$$

Donc le degré relatif du système considéré est $r=2$.

5.2.2. Choix de la surface de glissement

Sachant que le degré relatif du système étudié est $r=2$, et en se basant sur le théorème 3.1 ainsi que l'équation (3.6), la surface de glissement choisie est de la forme suivante :

$$\begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} \dot{e}_1 + \lambda_1 e_1 \\ \dot{e}_2 + \lambda_2 e_2 \end{pmatrix} \quad (5.13)$$

Où : $e_1 = q_1 - q_{1d}$, $e_2 = q_2 - q_{2d}$, $\dot{e}_1 = \dot{q}_1 - \dot{q}_{1d}$ et $\dot{e}_2 = \dot{q}_2 - \dot{q}_{2d}$.

Avec : q_{1d} et q_{2d} sont les positions angulaires de référence du premier et du deuxième segment respectivement. \dot{q}_{1d} et \dot{q}_{2d} sont les vitesses angulaires de référence du premier et du deuxième segment respectivement.

λ_1 et λ_2 sont des constantes positives.

5.2.3. Conception de la loi de commande

En exploitant la surface de glissement donnée par l'équation (5.13), le principal objectif de cette technique est l'utilisation de la fonction de Lyapunov dans le but de déterminer une loi de commande par mode de glissement qui contraint l'équation de glissement à converger vers zéro en un temps fini.

Théorème 5.1:

Pour un modèle du système défini par l'équation (5.2), la loi de commande par mode glissant qui contraint les erreurs de poursuite à converger vers zéro en un temps fini est exprimée par la relation suivante :

$$u = u_d + u_{eq} \quad (5.14)$$

Sachant que : $u_d = \begin{bmatrix} u_{d1} \\ u_{d2} \end{bmatrix}$ et $u_{eq} = \begin{bmatrix} u_{eq1} \\ u_{eq2} \end{bmatrix}$.

Avec u_d est la commande discontinue appelée aussi commande robuste et u_{eq} est la commande équivalente.

Nous allons maintenant déterminer la loi de commande par mode glissant globale u .

5.2.3.1. La loi de commande discontinue

La loi de commande discontinue est conçue suivant l'équation (3.22), nous aurons alors :

$$\begin{pmatrix} u_{d1} \\ u_{d2} \end{pmatrix} = \begin{pmatrix} -k_1 \text{sign}(s_1) \\ -k_2 \text{sign}(s_2) \end{pmatrix} \quad (5.15)$$

Où k_1 et k_2 sont des constantes positives.

Par substitution de l'équation (5.13) dans (5.15), nous aurons :

$$\begin{pmatrix} u_{d1} \\ u_{d2} \end{pmatrix} = \begin{pmatrix} -k_1 \text{sign}(\dot{e}_1 + \lambda_1 e_1) \\ -k_2 \text{sign}(\dot{e}_2 + \lambda_2 e_2) \end{pmatrix} \quad (5.16)$$

5.2.3.2. La loi de commande équivalente

La loi de commande équivalente est extraite suivant l'équation (3.18), qui est déduite à partir de $\dot{s} = 0$. En utilisant la surface considérée, nous obtenons :

$$\begin{pmatrix} \dot{s}_1 \\ \dot{s}_2 \end{pmatrix} = \begin{pmatrix} \ddot{e}_1 + \lambda_1 \dot{e}_1 \\ \ddot{e}_2 + \lambda_2 \dot{e}_2 \end{pmatrix} \quad (5.17)$$

Avec :

$$\begin{pmatrix} \ddot{e}_1 \\ \ddot{e}_2 \end{pmatrix} = \begin{pmatrix} \ddot{q}_1 - \ddot{q}_{1d} \\ \ddot{q}_2 - \ddot{q}_{2d} \end{pmatrix} \quad (5.18)$$

Donc, Au régime glissant :

$$\begin{pmatrix} \dot{s}_1 \\ \dot{s}_2 \end{pmatrix} = \begin{pmatrix} (\ddot{q}_1 - \ddot{q}_{1d}) + \lambda_1 \dot{e}_1 \\ (\ddot{q}_2 - \ddot{q}_{2d}) + \lambda_2 \dot{e}_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (5.19)$$

Nous aurons alors :

$$\begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{pmatrix} = \begin{pmatrix} \ddot{q}_{1d} - \lambda_1 \dot{e}_1 \\ \ddot{q}_{2d} - \lambda_2 \dot{e}_2 \end{pmatrix} \quad (5.20)$$

En substituant l'équation (5.2) dans (5.20), et en faisant quelques calculs nous obtenons la commande équivalente donnée par la formule suivante :

$$\begin{pmatrix} u_{eq1} \\ u_{eq2} \end{pmatrix} = M(q) \begin{pmatrix} \ddot{q}_{1d} - \lambda_1 \dot{e}_1 \\ \ddot{q}_{2d} - \lambda_2 \dot{e}_2 \end{pmatrix} + C(q, \dot{q})\dot{q} + g(q) \quad (5.21)$$

5.2.3.3. La loi de commande globale

Suivant l'équation (5.14), et en utilisant les équations (5.16) et (5.21), la loi de commande globale par mode glissant est exprimée par :

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = M(q) \begin{pmatrix} \ddot{q}_{1d} - \lambda_1 \dot{e}_1 \\ \ddot{q}_{2d} - \lambda_2 \dot{e}_2 \end{pmatrix} + C(q, \dot{q})\dot{q} + g(q) + \begin{pmatrix} -k_1 \text{sign}(\dot{e}_1 + \lambda_1 e_1) \\ -k_2 \text{sign}(\dot{e}_2 + \lambda_2 e_2) \end{pmatrix} \quad (5.22)$$

Les deux commande u_1 et u_2 représentent respectivement, les couples τ_1 et τ_2 dans notre cas.

Démonstration :

Considérons la fonction de Lyapunov candidate suivante :

$$V(s) = \frac{1}{2} s^T M(q) s \quad (5.23)$$

Afin d'atteindre le régime glissant, les trajectoires au voisinage de $s=0$ doivent être orientées vers le mode $\dot{V} \leq 0$. Soit \dot{q}_r une variable de référence définie par $\dot{q}_r = \dot{q}_d - \lambda e$. En dérivant la fonction de Lyapunov donnée par l'équation (5.23) par rapport au temps, nous obtenons :

$$\dot{V}(s) = s^T M(q) \dot{s} + \frac{1}{2} s^T \dot{M}(q) s \quad (5.24)$$

$$\dot{V}(s) = s^T [-M(q) \ddot{q}_r - C(q, \dot{q}) \dot{q} - g(q) + u] + \frac{1}{2} s^T \dot{M}(q) s \quad (5.25)$$

En utilisant la troisième propriété du modèle dynamique citée dans le premier chapitre qui stipule que la matrice $\dot{M}(q) - 2C(q, \dot{q})$ est antisymétrique, nous aurons :

$$\dot{V}(s) = s^T [-M(q) \ddot{q}_r - C(q, \dot{q}) \dot{q}_r - g(q) + u] \quad (5.26)$$

En utilisant la loi de commande donnée par :

$$u = M(q) \ddot{q}_r + C(q, \dot{q}) \dot{q}_r + g(q) - K \text{sign}(s) \quad (5.27)$$

Par substitution de (5.27) dans (5.26) nous obtenons :

$$\dot{V}(s) = -s^T K \text{sign}(s) \quad (5.28)$$

Nous trouverons donc :

$$\dot{V}(s) = -|s|K \leq 0 \quad (5.29)$$

L'expression (5.29) montre que la surface $s=0$ est atteinte en un temps fini, et que l'erreur converge exponentiellement vers 0.

5.3. Synthèse de la loi de commande par backstepping

La loi de commande par backstepping est une technique de commande non linéaire basée sur le théorème de Lyapunov. Il s'agit d'une technique de conception systématique appliquée à des systèmes de rétroaction stricts. La commande par backstepping procède d'une manière systématique en construisant la fonction de Lyapunov tout au long des étapes de conception de la loi de commande [61].

Cette stratégie de commande est utilisée notamment dans le domaine de la robotique et elle repose sur les points suivants :

- i- Définition de la variable d'erreur
- ii- Définition de la fonction de Lyapunov
- iii- Définition et dérivation de la fonction de Lyapunov
- iv- Détermination d'une commande virtuelle
- v- Aller vers l'étape suivante jusqu'à l'apparition des vraies variables de contrôle

Le principal objectif de la commande par backstepping est de rendre les systèmes bouclés équivalents à un système d'ordre un qui vérifie la stabilité au sens de Lyapunov.

Considérons le vecteur d'état suivant :

$$x = [x_1 \ x_2 \ x_3 \ x_4]^T = [q_1 \ \dot{q}_1 \ q_2 \ \dot{q}_2]^T \quad (5.30)$$

En considérant le modèle du robot donné par l'équation (1.73). Le modèle du système est donc subdivisé en deux sous-systèmes. Le premier sous-système est décrit par :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{\tau_1}{m_{11}} - \frac{m_{12}}{m_{11}} \dot{x}_4 - \frac{c_{11}}{m_{11}} x_2 - \frac{c_{12}}{m_{11}} x_4 - \frac{Q_1}{m_{11}} \end{cases} \quad (5.31)$$

Le deuxième sous-système est décrit par :

$$\begin{cases} \dot{x}_3 = x_4 \\ \dot{x}_4 = \frac{\tau_2}{m_{22}} - \frac{m_{21}}{m_{22}} \dot{x}_2 - \frac{c_{21}}{m_{22}} x_2 - \frac{Q_2}{m_{22}} \end{cases} \quad (5.32)$$

❖ Le premier sous-système (premier segment)

Soit $[x_1 \ x_2]^T = [q_1 \ \dot{q}_1]^T$ le vecteur d'état du premier sous-système. Cela correspond à la représentation d'état donnée par (5.31).

Étape 1

Sachant que x_{1d} représente la trajectoire de référence de x_1 . Nous définissons l'erreur de poursuite de q_1 comme suit :

$$z_1 = x_{1d} - x_1 \quad (5.33)$$

Sa dérivée est exprimée par :

$$\dot{z}_1 = \dot{x}_{1d} - \dot{x}_1 = \dot{x}_{1d} - x_2 \quad (5.34)$$

Considérons la fonction de Lyapunov candidate suivante :

$$V_1 = \frac{1}{2} z_1^2 \quad (5.35)$$

Cette fonction de Lyapunov est définie positive. Afin de garantir la stabilité de ce premier sous système, la dérivée de cette fonction doit être définie négative, elle est donnée par :

$$\dot{V}_1 = z_1 \dot{z}_1 \quad (5.36)$$

En substituant (5.34) dans (5.36), nous aurons :

$$\dot{V}_1 = z_1 (\dot{x}_{1d} - x_2) \quad (5.37)$$

Afin de rendre le sous-système stable une commande virtuelle α_1 est choisie tel que :

$$\alpha_1 = k_1 z_1 + \dot{x}_{1d} \quad (5.38)$$

Où k_1 est une constante positive.

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

$$\dot{V}_1 = -k_1 z_1^2 \quad (5.39)$$

Par identification des deux équations (5.37) et (5.39), nous obtiendrons :

$$\dot{x}_{1d} - x_2 = -k_1 z_1 \quad (5.40)$$

Nous trouverons donc :

$$x_2 = k_1 z_1 + \dot{x}_{1d} \quad (5.41)$$

Étape 2

Sachant maintenant que x_{2d} est la trajectoire de référence de x_2 . L'erreur de poursuite de q_2 est définie par :

$$z_2 = x_{2d} - x_2 \quad (5.42)$$

En utilisant l'équation (5.41), cette relation devient :

$$z_2 = k_1 z_1 + \dot{x}_{1d} - x_2 \quad (5.43)$$

Ou encore :

$$z_2 = k_1 z_1 + \dot{z}_1 \Rightarrow \dot{z}_1 = z_2 - k_1 z_1 \quad (5.44)$$

En dérivant l'équation (5.43), nous aurons :

$$\dot{z}_2 = k_1 \dot{z}_1 + \ddot{x}_{1d} - \dot{x}_2 = k_1(\dot{x}_{1d} - \dot{x}_1) + \ddot{x}_{1d} - \dot{x}_2 \quad (5.45)$$

La fonction de Lyapunov dans ce cas est augmentée par un terme afin d'assurer la stabilité du système (z_1, z_2) , ce qui donnera :

$$V_2 = V_1 + \frac{1}{2} z_2^2 = \frac{1}{2} z_1^2 + \frac{1}{2} z_2^2 \quad (5.46)$$

En dérivant cette fonction, nous obtenons :

$$\dot{V}_2 = z_1 \dot{z}_1 + z_2 \dot{z}_2 \quad (5.47)$$

Le développement de la dérivée de la fonction de Lyapunov, en substituant les équations (5.44) et (5.45) dans (5.47) donnera :

$$\dot{V}_2 = -k_1 z_1^2 + z_2(k_1 \dot{z}_1 + \ddot{x}_{1d} - \dot{x}_2 + z_1) \quad (5.48)$$

Afin de garantir la stabilité du premier sous-système, la dérivée de la deuxième fonction de Lyapunov \dot{V}_2 doit prendre une valeur négative ($\dot{V}_2 < 0$). Pour cela, pour k_2 une constante positive, nous posons :

$$\dot{V}_2 = -k_1 z_1^2 - k_2 z_2^2 \quad (5.49)$$

Par identification entre (5.48) et (5.49), nous obtenons :

$$\dot{x}_2 = k_1 \dot{z}_1 + \ddot{x}_{1d} + z_1 + k_2 z_2 \quad (5.50)$$

Par substitution de l'équation (5.31) dans (5.50), la commande du premier sous système est enfin extraite et est donnée par :

$$\tau_1 = m_{11}((k_1 + k_2)\dot{z}_1 + \ddot{x}_{1d} + (1 + k_1 k_2)z_1) + m_{12}\dot{x}_4 + c_{11}x_2 + c_{12}x_4 + Q_1 \quad (5.51)$$

❖ Le deuxième sous-système (deuxième segment)

Soit $[x_3 \ x_4]^T = [q_2 \ \dot{q}_2]^T$ le vecteur d'état du deuxième sous-système. Cela correspond à la représentation d'état donnée par (5.32).

Étant donné k_3 et k_4 des constantes positives. La détermination de la loi de commande du deuxième sous-système (segment) a été développée en procédant de la même façon que lors de la détermination de la loi de commande du premier sous-système. Nous aurons alors :

$$\tau_2 = m_{22}((k_3 + k_4)\dot{z}_3 + \ddot{x}_{3d} + (1 + k_3k_4)z_3) + m_{21}\dot{x}_2 + c_{21}x_2 + Q_2 \quad (5.52)$$

5.4. Résultats de simulation et interprétation

Les simulations des différentes lois de commandes non linéaires et robustes développées ci-dessus et appliquées au robot manipulateur ont été effectuées sous l'environnement Matlab/Simulink et en utilisant le modèle du robot manipulateur mentionnées dans l'Annexe B.

Ces commandes ont été testées sur le même modèle du robot manipulateur afin de considérer leurs performances, en tenant compte des cas suivants :

- La poursuite de trajectoire de position et de vitesse désirées des différentes articulations.
- La réaction des différentes lois de commande vis-à-vis des perturbations extérieures, qui peuvent représenter un choc au niveau de chaque articulation, on parle dans ce cas de la régulation. L'influence de ce choc est ressentie sur les deux articulations à travers les deux couples de commande. Ce choc sera introduit dans les simulations par une impulsion d'une durée et d'une amplitude bien définies.
- La réaction des différentes lois de commande vis-à-vis des incertitudes paramétriques, plus précisément le changement de son inertie. Ceci permet de tester la robustesse des lois de commandes appliquées au robot manipulateur.

5.4.1. Simulations de la loi de commande par mode glissant

En premier lieu nous allons appliquer au robot manipulateur la commande robuste par mode glissant développée précédemment dans ce chapitre avec et sans présence de perturbations et d'incertitudes. Les coefficients de la loi de commande sont choisis d'une manière convenable afin de garantir la stabilité du système ainsi que la robustesse de la commande.

Le modèle dynamique du robot manipulateur est exprimé par l'équation (1.55), dont les matrices représentatives sont données dans la section B.1 de l'Annexe B [123].

Les positions désirées ont pour valeurs 1.5 rad et 1.6 rad pour la première et la deuxième articulation respectivement.

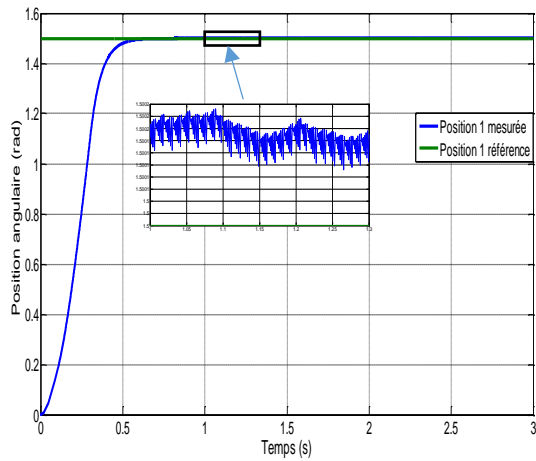
En poursuite de trajectoire les trajectoires de références sont générées en utilisant une interpolation polynomiale de degré cinq assurant des trajectoires continues en positions, vitesses et accélérations. Avec des valeurs désirées $q_1^{fin} = 1.1 \text{ rad}$, $q_2^{fin} = 1.3 \text{ rad}$ et $q_1^{init} = q_2^{init} = 0 \text{ rad}$ pour la première et la deuxième articulation respectivement.

La perturbation représentant un choc d'impact intervenant à un instant donné, et qui influencera les deux articulations sous forme des couples tranchées des deux commandes. Cette perturbation est simulée par une impulsion d'une durée de 30 ms et d'une amplitude de 200 N.m pour les deux articulations à l'instant 1.5 s. afin de tester la robustesse de la commande appliquée une variation de 80% de l'inertie du robot de la valeur initiale est considérée.

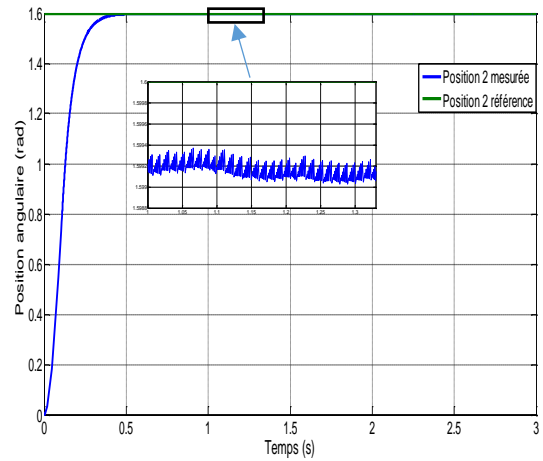
Paramètres de la commande

$\lambda_1 = 15$, $\lambda_2 = 15$, $k_1 = 800$, $k_2 = 200$.

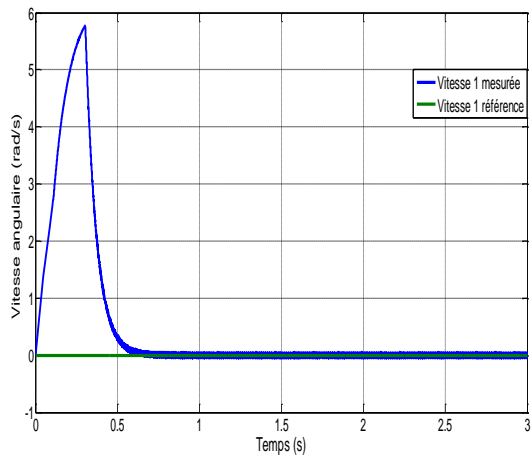
Les résultats de simulation obtenus sont montrés sur les figures (5.1), (5.2), (5.3) et (5.4).



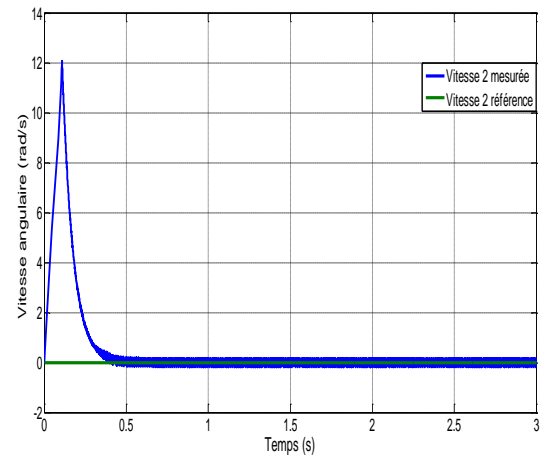
(A)



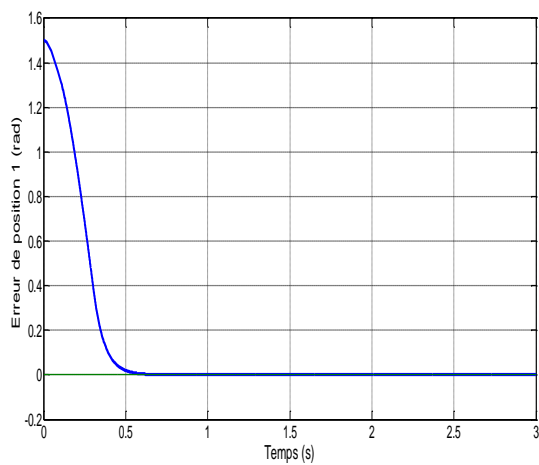
(B)



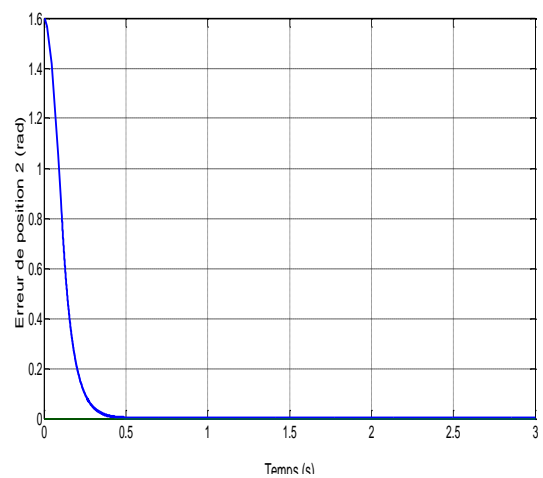
(C)



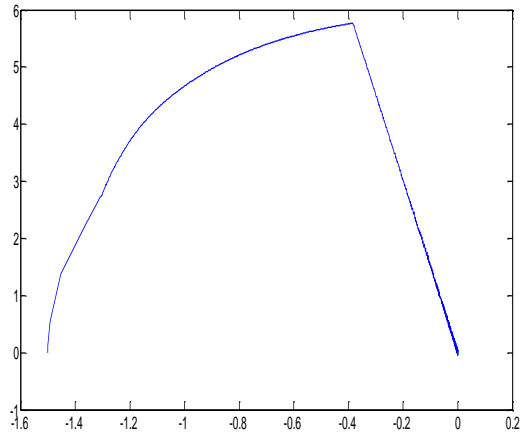
(D)



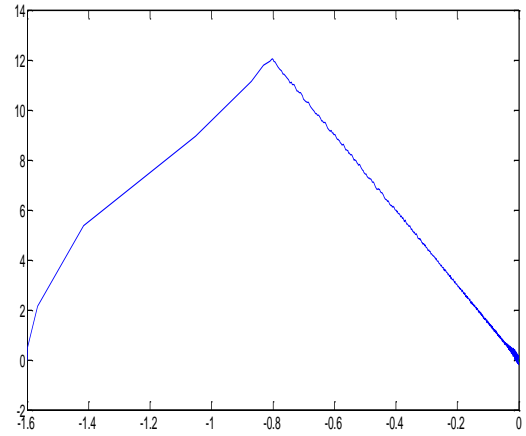
(E)



(F)

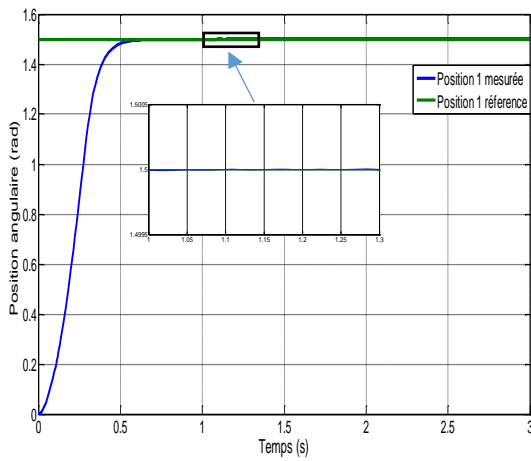


(G)

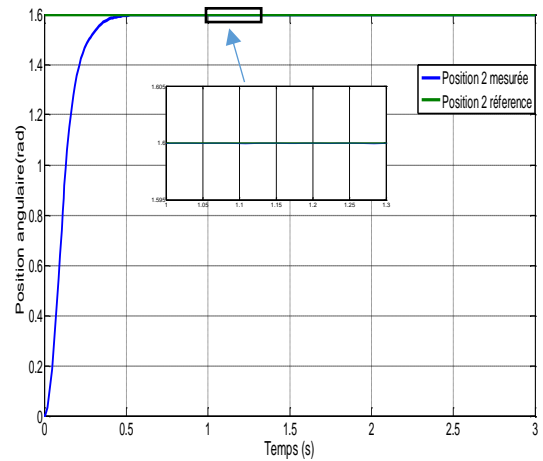


(H)

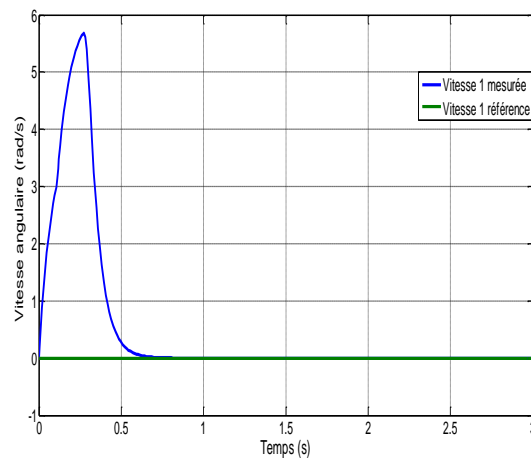
Figure 5.1. Commande par Modes glissants en utilisant la fonction sign : (A) et (B) positions angulaires, (C) et (D) vitesses angulaires, (E) et (F) erreurs des positions angulaires, (G) et (H) plans de phase.



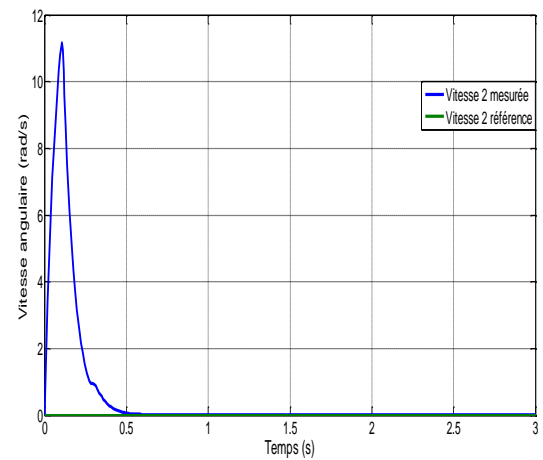
(A)



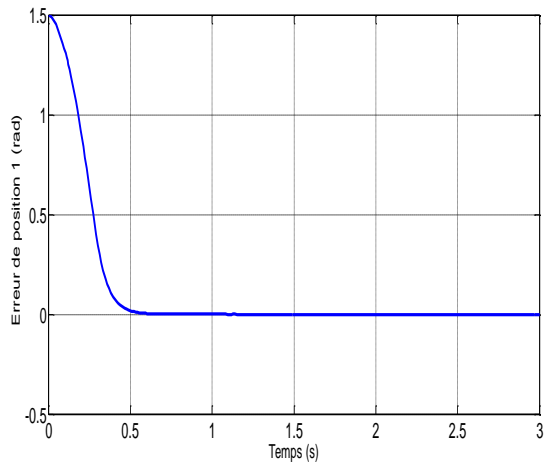
(B)



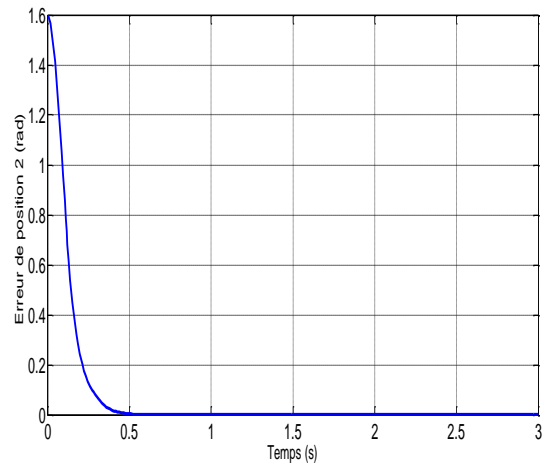
(C)



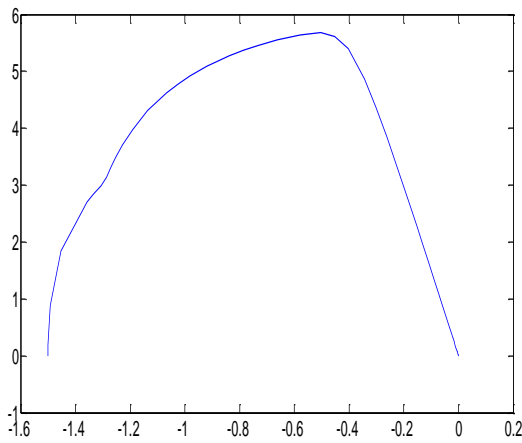
(D)



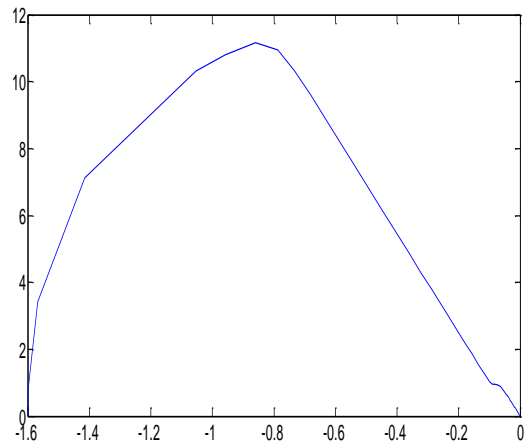
(E)



(F)

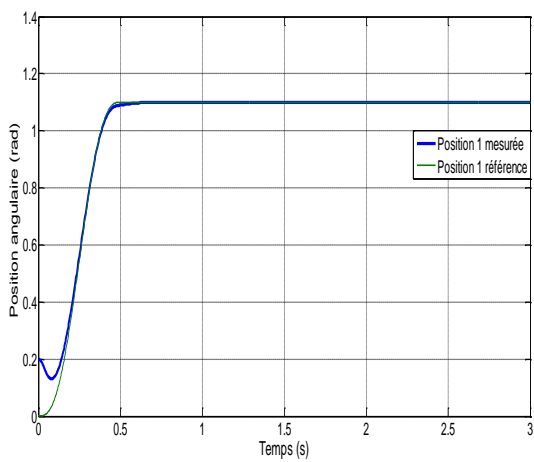


(G)

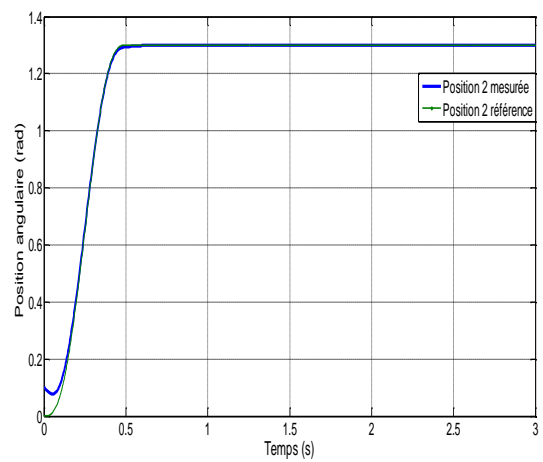


(H)

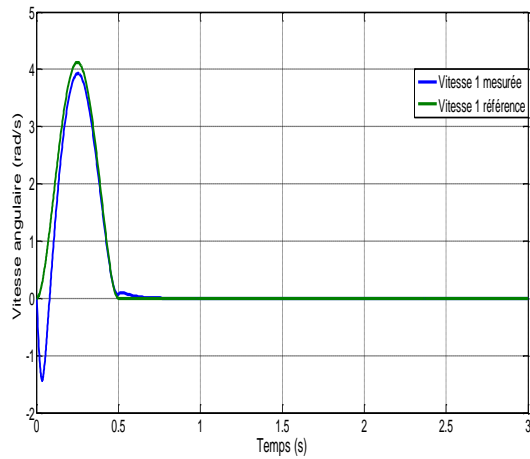
Figure 5.2. Commande par Modes glissants en utilisant la fonction tanh : (A) et (B) positions angulaires, (C) et (D) vitesses angulaires, (E) et (F) erreurs des positions angulaires, (G) et (H) plans de phase.



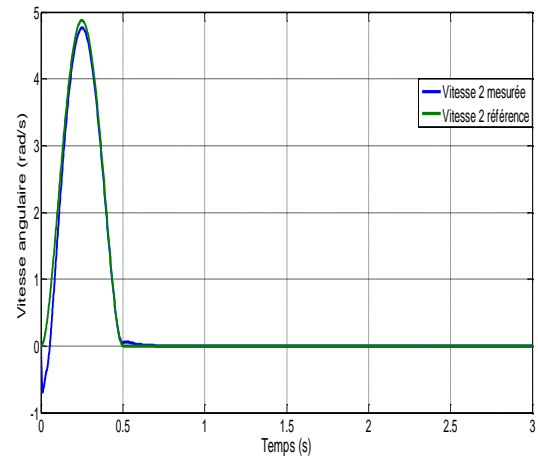
(A)



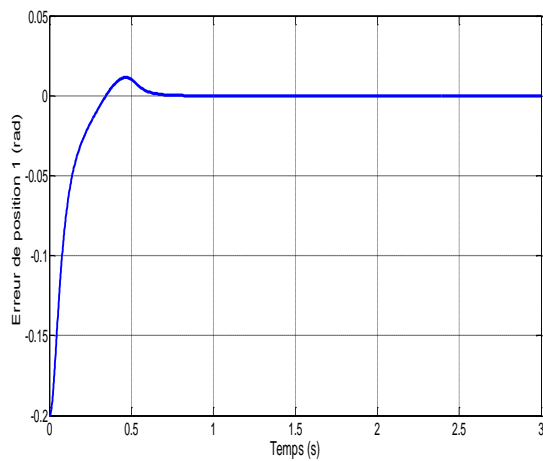
(B)



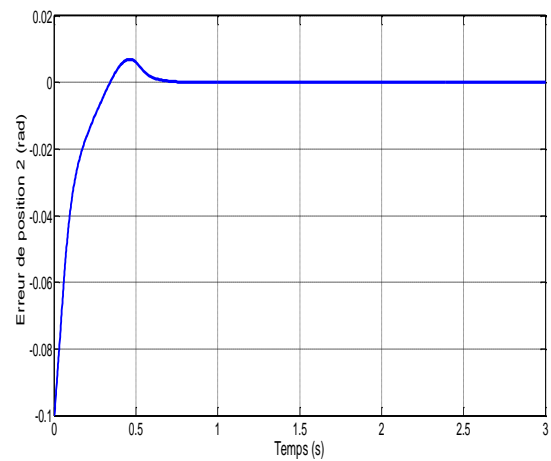
(C)



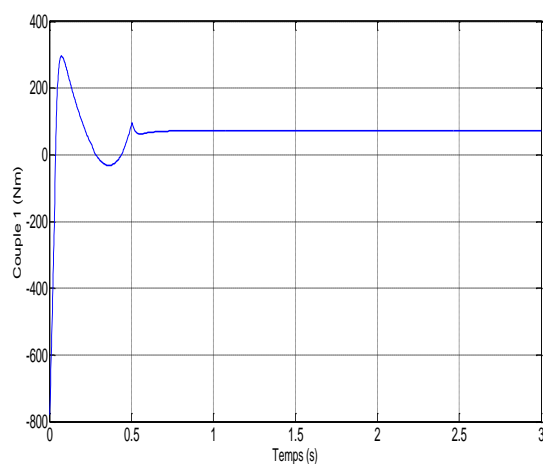
(D)



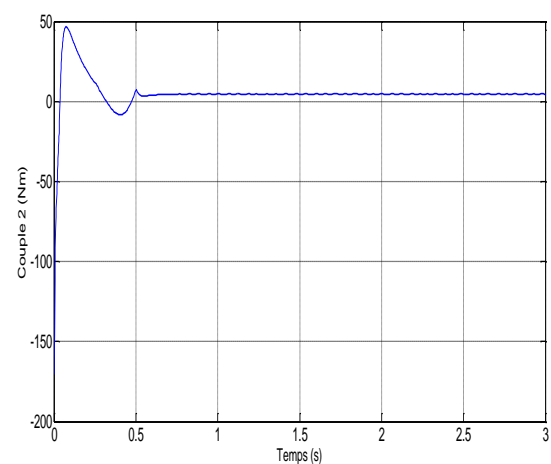
(E)



(F)

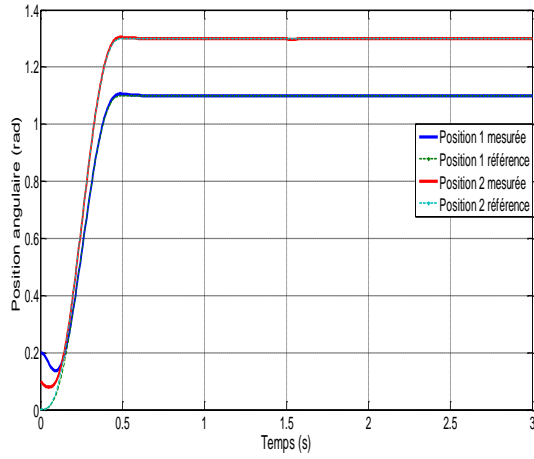


(G)

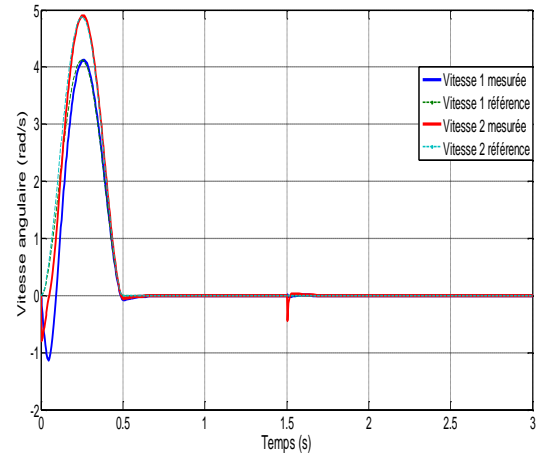


(H)

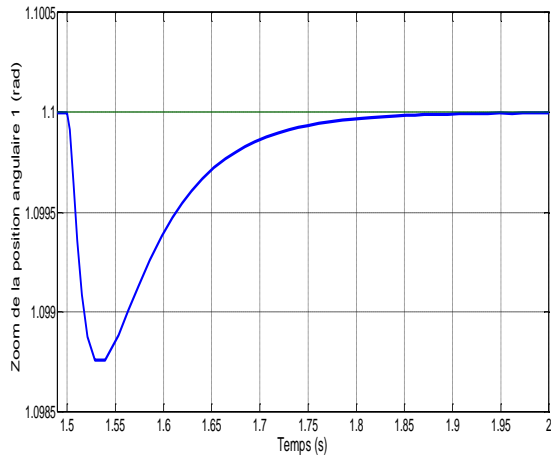
Figure 5.3. Commande par Modes glissants en poursuite de trajectoire avec conditions initiales $[0.2, 0.1]$: (A) et (B) positions angulaires, (C) et (D) vitesses angulaires, (E) et (F) erreurs des positions angulaires, (G) et (H) couples de commande..



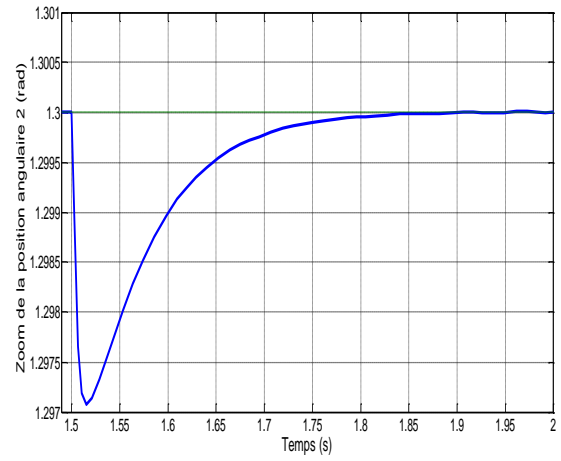
(A)



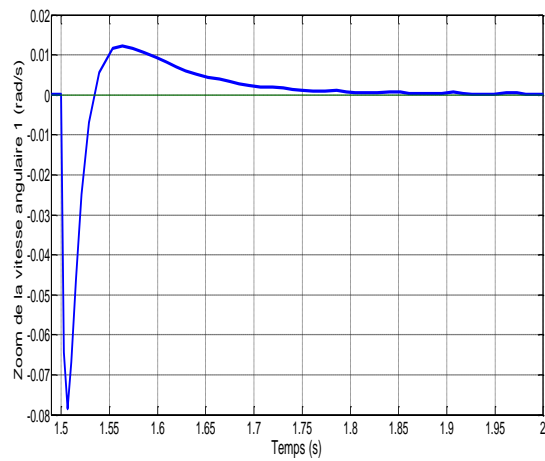
(B)



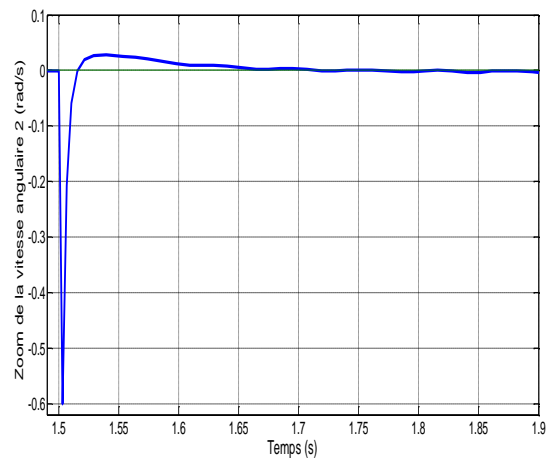
(C)



(D)



(E)



(F)

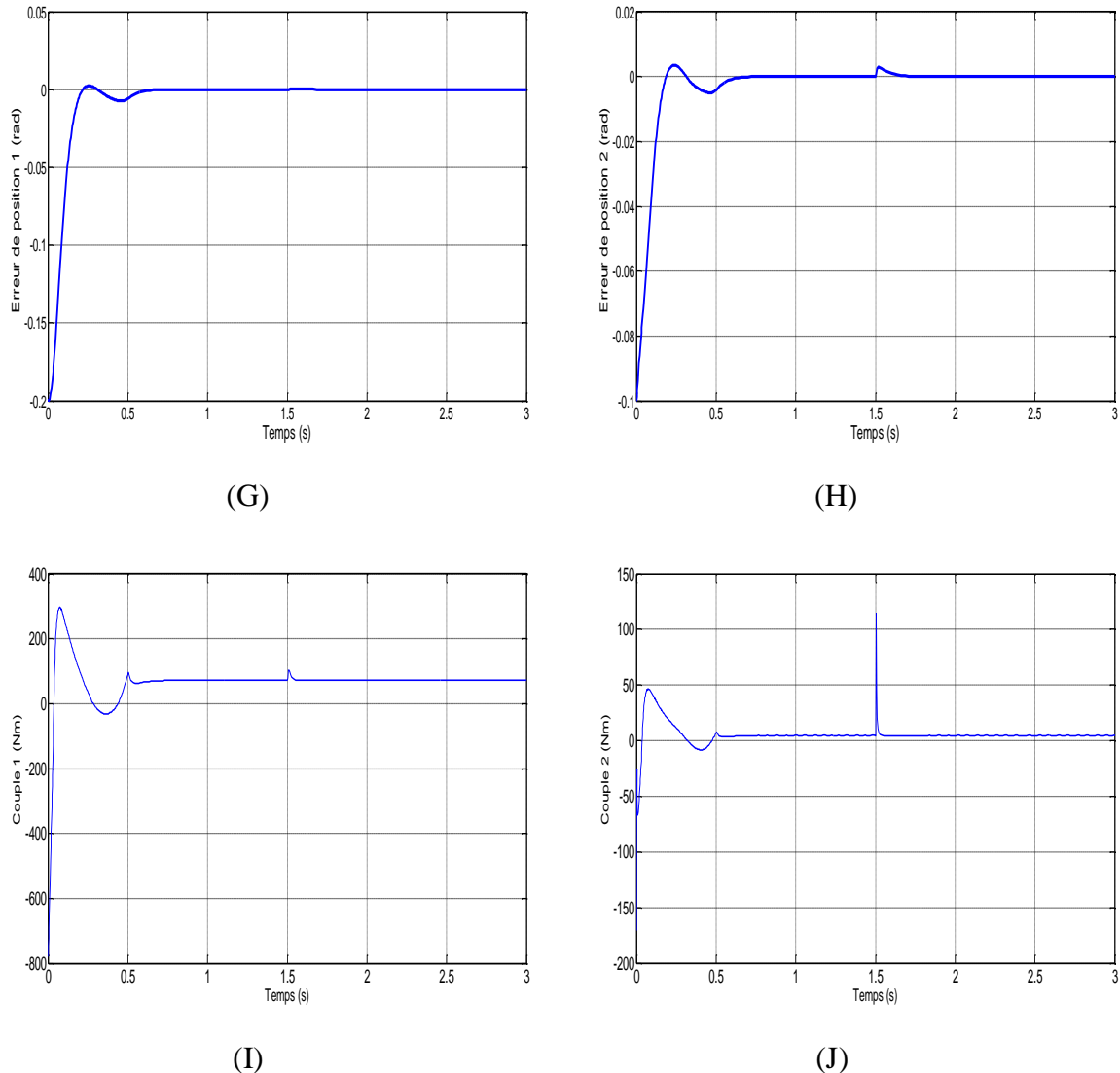


Figure 5.4. Commande par Modes glissants en poursuite de trajectoire avec conditions initiales $[0.2, 0.1]$, une perturbation et variation paramétrique: (A) et (B) positions angulaires, (C) et (D) vitesses angulaires, (E) et (F) erreurs des positions angulaires, (I) et (J) couples de commande.

Synthèse 1

A travers les figures (5.1.A) et (5.1.B) nous remarquons que les deux positions articulaires 1 et 2 rejoignent leurs positions désirées avec un temps de réponse de 0.41s et 0.295s respectivement. Les figures (5.1.E) et (5.1.F) représentant les erreurs de positions confirment ce résultat puisque les deux erreurs convergent exponentiellement vers zéro. Les figures (5.1.C) et (5.1.D) montrent que les vitesses articulaires vont atteindre des valeurs maximales lors du démarrage puis elles diminuent vers des valeurs nulles à l'instant où les positions articulaires atteignent leurs positions désirées. Il faut noter aussi qu'il y a présence des oscillations autour des valeurs désirées (phénomène de broutement « chattering ») qui peut être associé à l'utilisation de la fonction « sign » au niveau de commande discontinue. Les plans de phases montrent que la dynamique du système est attirée vers la surface de glissement puis glisse sur celle-ci jusqu'au point d'équilibre telle qu'il est montré sur les figures (5.1.G) et (5.1.H).

D'après les figures (5.2.A) au (5.2. H), nous remarquons que l'utilisation de la fonction Tanh à la place de la fonction sign au niveau de la commande discontinue a permis de limiter le phénomène de chattering tout en gardant pratiquement les mêmes performances.

En se référant aux figures (5.3.A) et (5.3.B), nous remarquons un bon suivi de trajectoire de référence des deux positions articulaires, malgré le changement des conditions initiales, 0.2 rad pour la première articulation et 0.1rad pour la deuxième. Nous avons aussi un bon suivi des vitesses articulaires tel qu'il est montré sur les figures (5.3.C) et (5.3.D) avec une phase de freinage au démarrage due au fait que les conditions initiales des deux articulations sont différentes de celles des trajectoires désirées, ensuite, les deux vitesses articulaires ont une allure croissante jusqu'à une valeur maximale $\dot{q}_{max} = \frac{15(q^f - q^i)}{8 t_f}$, puis elles décroissent jusqu'à zéro. Nous pouvons confirmer ces résultats à travers des figures (5.3.E) et (5.3.F) représentant les erreurs de poursuite. Afin de mettre en évidence la robustesse de cette loi de commande, les figures (5.4.A) au (5.4.H) illustrent les courbes de poursuite de position en présence d'une variation paramétrique et d'une perturbation externe. Nous constatons à partir des figures (5.4.A) et (5.4.B), que la variation paramétrique a un léger effet sur la poursuite de trajectoire en position et en vitesse ce qui est confirmé en observant les figures (5.4.G) et (5.4.H) des erreurs de positions. Nous remarquons aussi à partir des figures (5.4.C) et (5.4.D), que les erreurs augmentent légèrement au moment de l'application du choc (à $t=1.5s$) pour s'annuler par la suite.

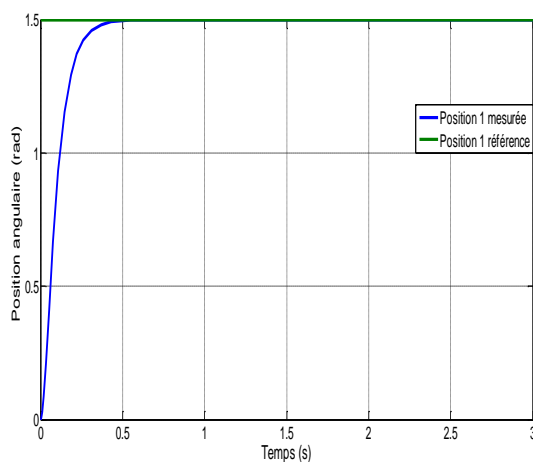
5.4.2. Simulations de la loi de commande par Backstepping

Nous allons maintenant appliquer au robot manipulateur la commande backstepping. Nous considérons toujours les mêmes conditions utilisées dans l'application précédente.

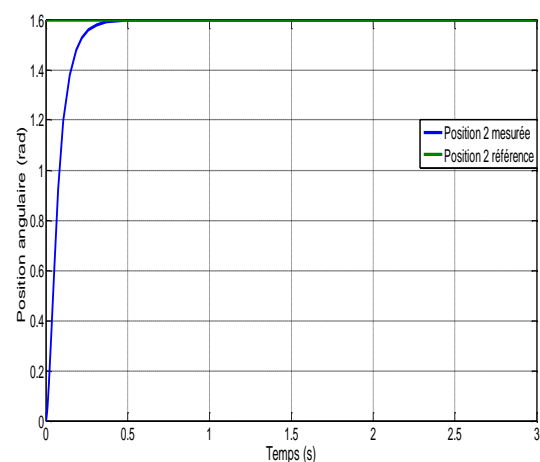
Paramètres de la commande

$$k_1= 47, k_2= 50, k_3= 55, k_4= 50.$$

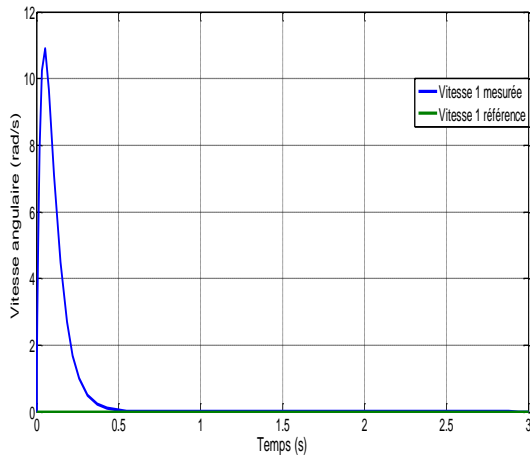
Les figures (5.5), (5.6) et (5.7) montrent les résultats de simulation obtenus.



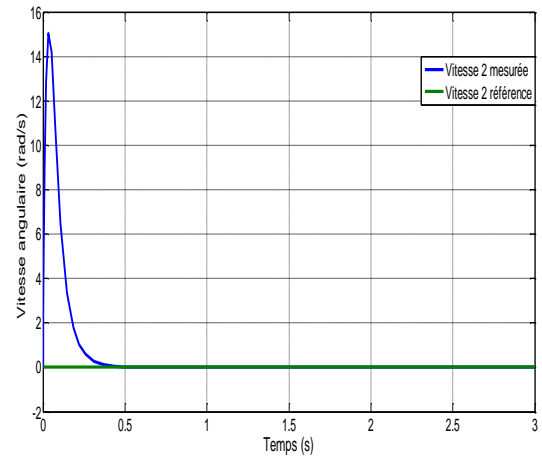
(A)



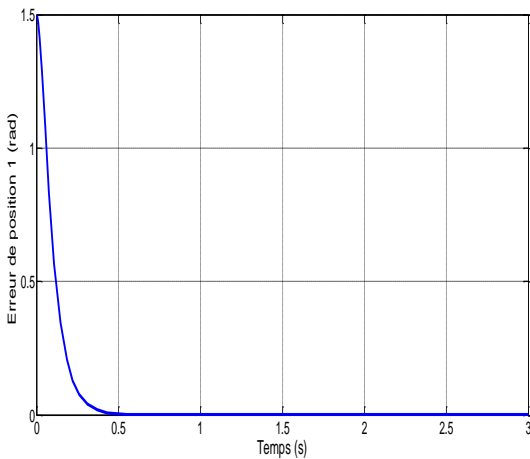
(B)



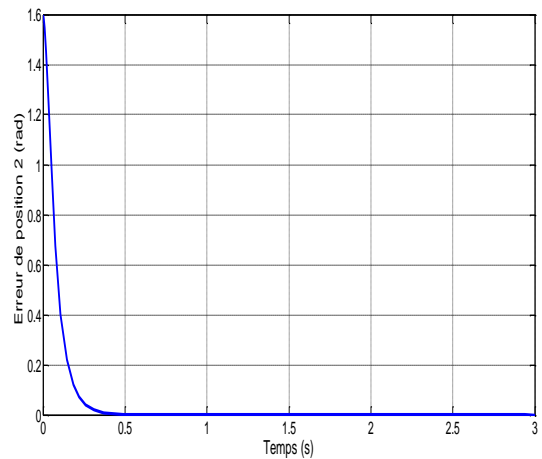
(C)



(D)

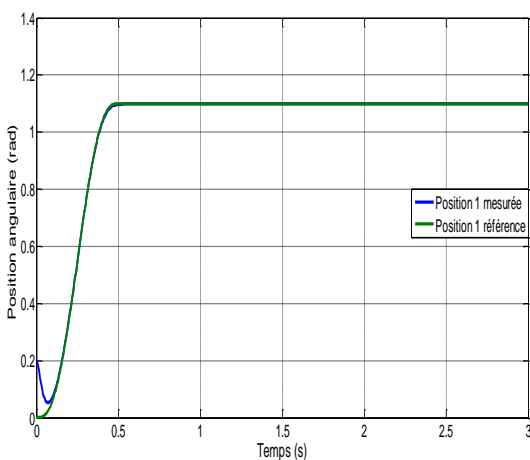


(E)

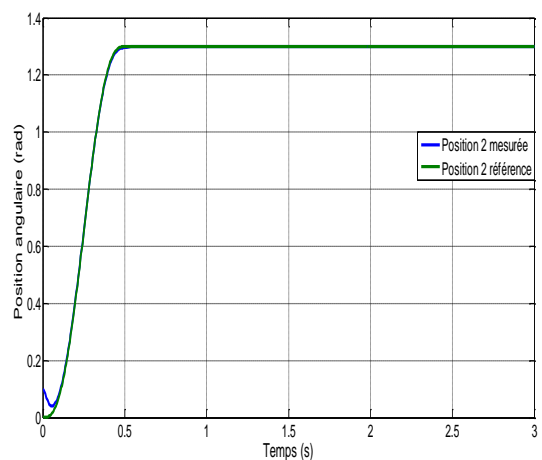


(F)

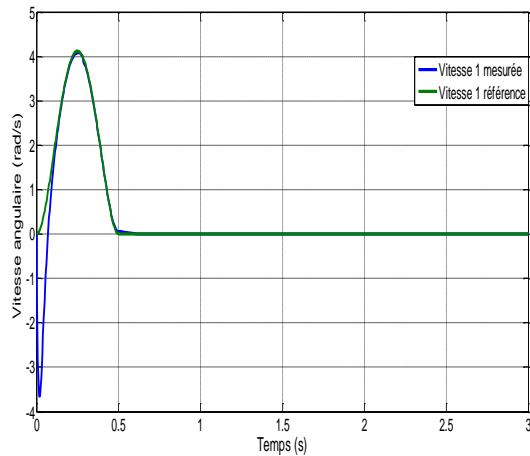
Figure 5.5. Commande par Backstepping: (A) et (B) positions angulaires, (C) et (D) vitesses angulaires, (E) et (F) erreurs des positions angulaires.



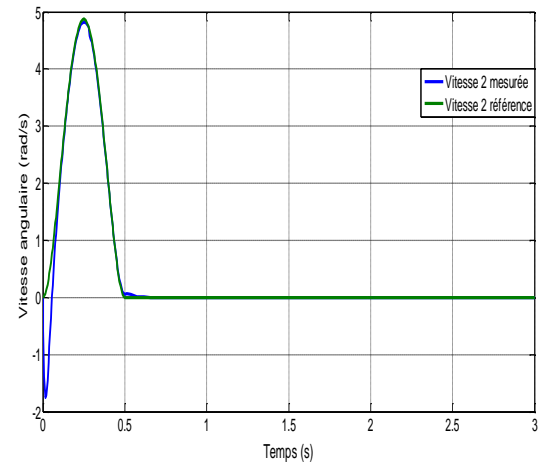
(A)



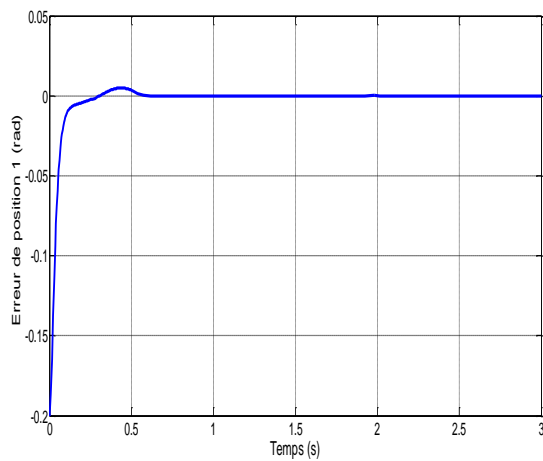
(B)



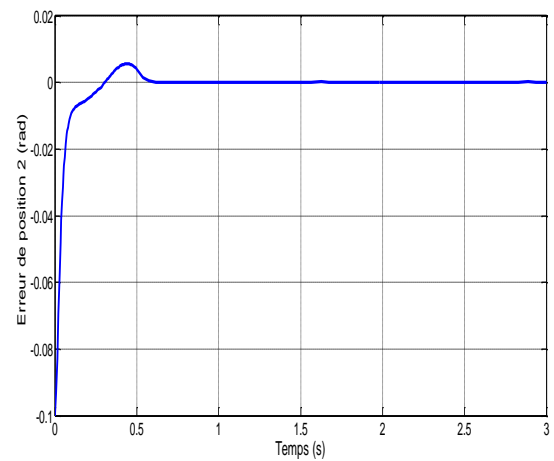
(C)



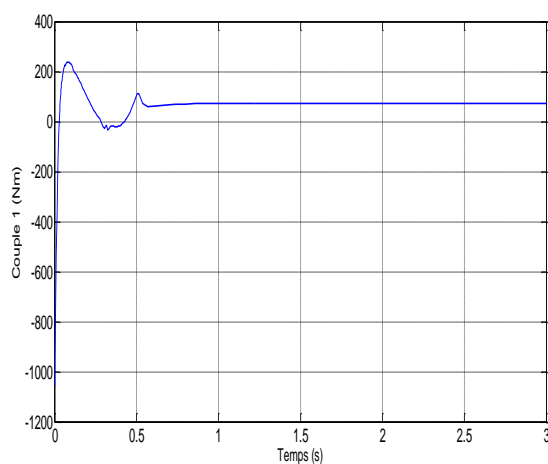
(D)



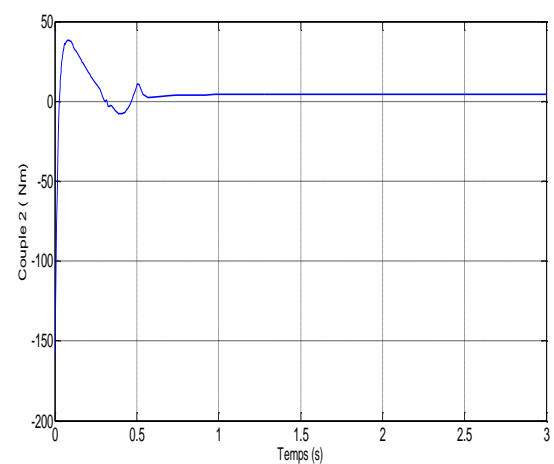
(E)



(F)

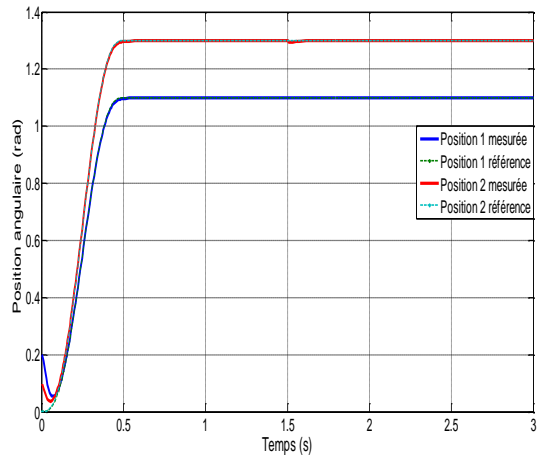


(G)

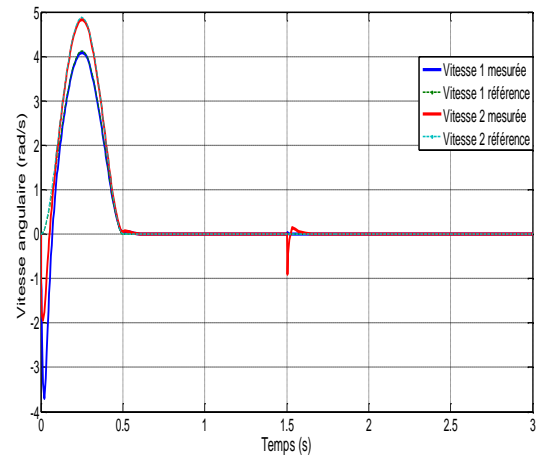


(H)

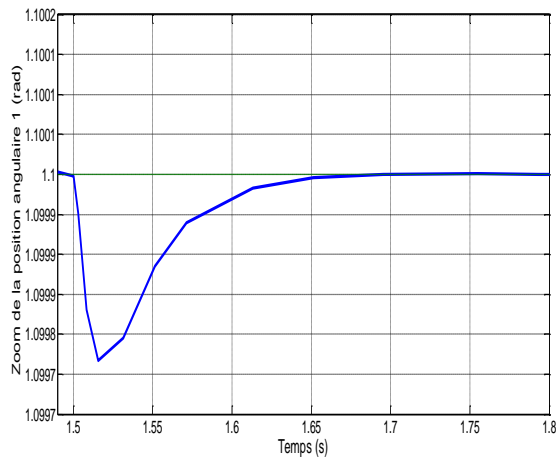
Figure 5.6. Commande par Backstepping en poursuite de trajectoire avec conditions initiales $[0.2, 0.1]$: (A) et (B) positions angulaires, (C) et (D) vitesses angulaires, (E) et (F) erreurs des positions angulaires, (G) et (H) couples de commande.



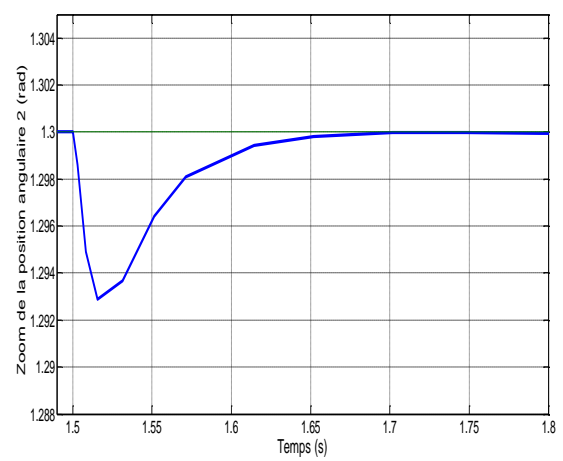
(A)



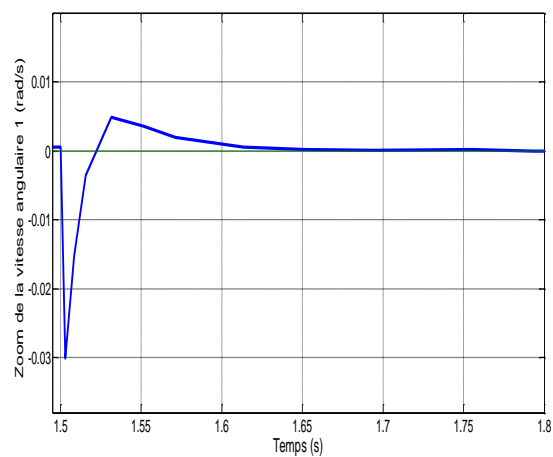
(B)



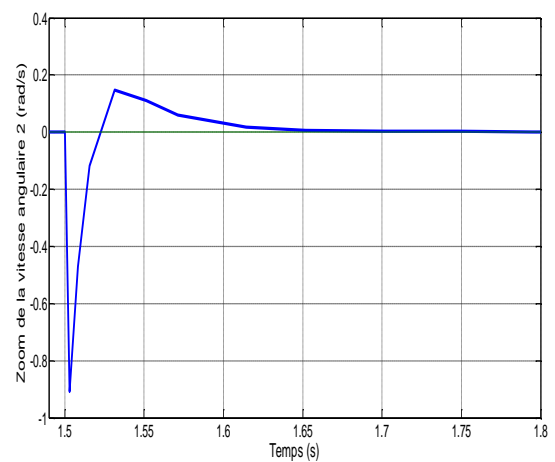
(C)



(D)



(E)



(F)

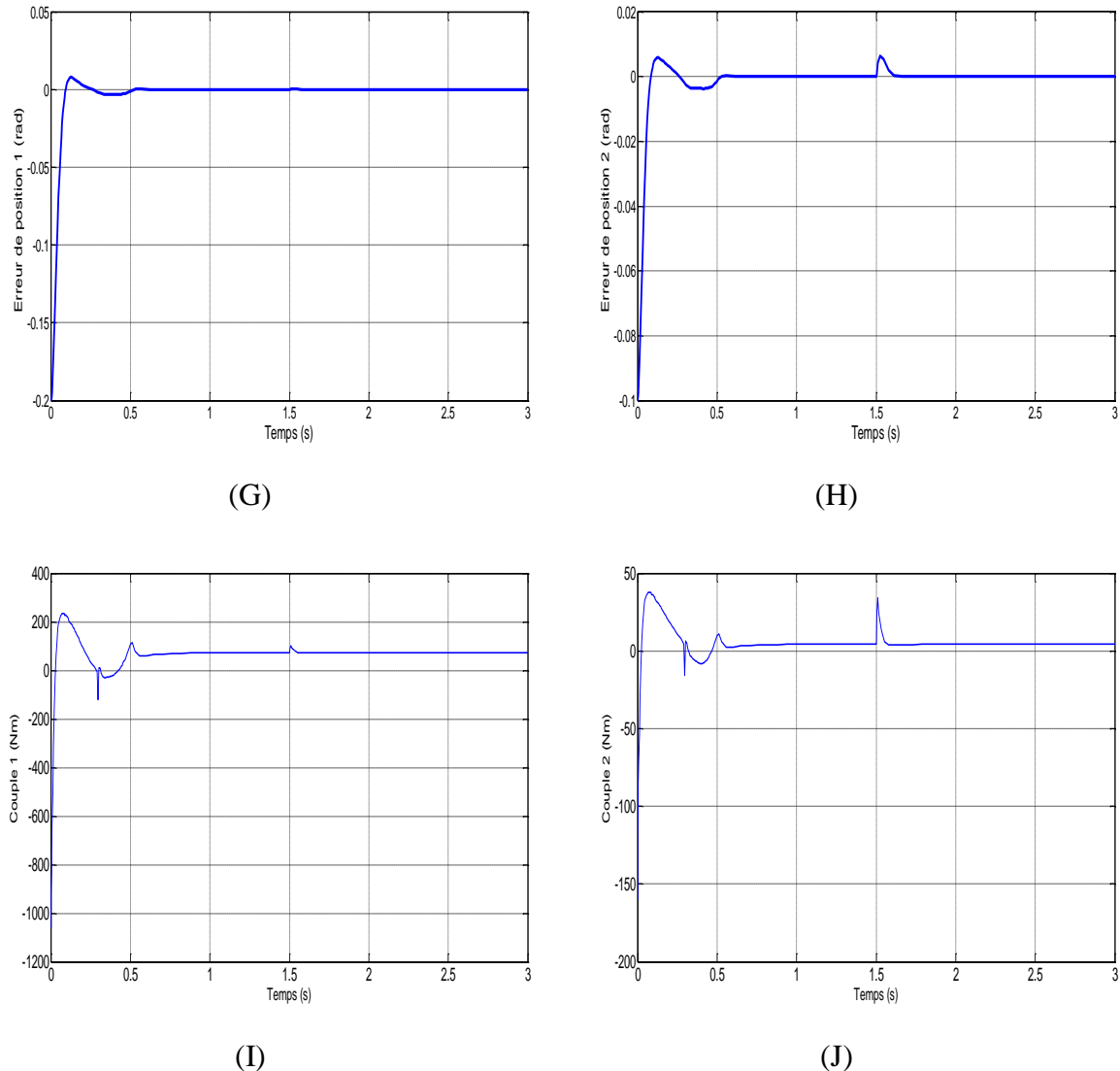


Figure 5.7. Commande par Backstepping en poursuite de trajectoire avec conditions initiales $[0.2, 0.1]$ une perturbation et variation paramétrique: (A) et (B) positions angulaires, (C) et (D) vitesses angulaires, (E) et (F) erreurs des positions angulaires, (I) et (J) couples de commande..

Synthèse 2

Les figures (5.5.A) et (5.5.B) montrent que les deux positions articulaires rejoignent leurs positions désirées avec un temps de réponse de 0.26s et 0.21s pour la première et la deuxième articulation respectivement. Les deux erreurs de position convergent vers zéro tel qu'il montré sur les figures (5.5.E) et (5.5.F).

Le changement des conditions initiales $[0.2, 0.1]$ n'empêche pas les deux positions articulaires à rejoindre leurs références comme il est montré sur les figures (5.6.A) et (5.6.B). Nous remarquons aussi que ce changement des conditions initiales provoque une phase de décélération pour que les positions articulaires atteignent leurs références ce qui est clairement montré sur les figures (5.6.C) et (5.6.D). Nous constatons aussi à travers les figures (5.7.C) et (5.7.D), que les erreurs de position augmentent légèrement tout autour du moment de l'introduction de la perturbation (à $t=1.5$ s) ceci est dû au choc appliqué à cet instant. Les figures

(5.7.E) et (5.7.F) représentant les erreurs de position montrent un effet presque négligeable de la variation paramétrique.

5.5. Optimisation des paramètres des lois de commande

La conception d'un contrôleur classique simple tel qu'il soit, exige la spécification de ses paramètres. Si nous considérons un contrôleur de type PI considéré comme le plus simple des régulateurs, deux paramètres doivent être spécifiés : un gain proportionnel et un gain intégral. En effet, plusieurs techniques existent pour la détermination de ces gains. Les plus répandues sont :

- La méthode de Ziegler-Nichols.
- La méthode de Naslin.
- La méthode basée sur le lieu des racines, etc.

La particularité de toutes ces méthodes est que les paramètres du régulateur sont obtenus pour un point de fonctionnement où le modèle peut être considéré linéaire. Ce qui implique que lorsque le système a réglé fonctionne hors de la zone valide du modèle linéaire, il y a un réglage sous optimal.

Ce problème devient de plus en plus compliqué pour d'autres types de contrôleurs. Pour remédier à ce problème, les méthodes d'optimisation apparaissent comme un outil puissant pour la détermination des paramètres optimaux des régulateurs, tel que les algorithmes génétiques et le PSO.

Le principe de l'optimisation des contrôleurs par les méthodes d'optimisation est illustré dans la figure 5.8.

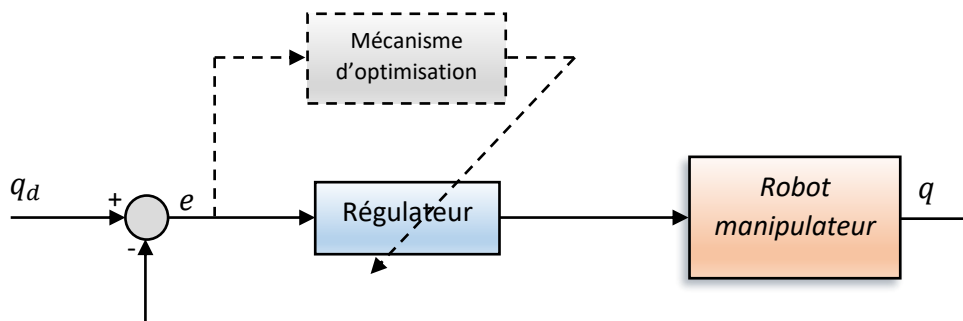


Figure 5.8. Schéma de principe de l'optimisation des régulateurs

5.5.1. Optimisation par la méthode des AGs de la commande par modes glissants

La conception du contrôleur décrit par l'équation (5.22) nécessite la spécification de quatre paramètres: λ_1 et λ_2 qui sont les pentes des deux surfaces de glissement, k_1 et k_2 les gains du contrôleur. Ces paramètres doivent être choisis pour assurer la convergence des positions articulaire vers les positions souhaitées. Malheureusement, nous n'avons pas de méthode directe pour trouver ces paramètres, à cause des non-linéarités et des couplages des systèmes robotiques. Les AGs peuvent être appliquées pour obtenir une régulation optimale, en considérant les performances et les caractéristiques du processus, et en tenant compte de toutes les réponses du système à régler. Dans cette partie, le contrôleur par modes glissants sera optimisé par un algorithme génétique.

Le codage

La première phase consiste à coder le problème dans les chromosomes appropriés de l'AG, et puis construire une population. Les paramètres à optimiser du régulateur sont codés en binaire avec des longueurs finies. Les chaînes de caractères représentant ces paramètres sont concaténées et juxtaposées pour construire un chromosome (figure 5.9). Chaque chromosome de la population peut présenter une solution possible du problème.

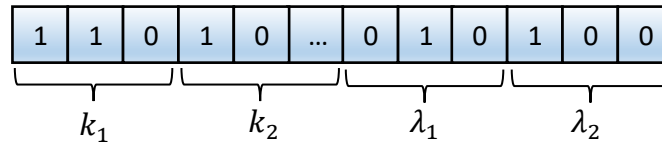


Figure 5.9. Structure du chromosome

La taille de la population initiale

La taille de la population est choisie de telle sorte à réaliser un bon compromis entre qualité du résultat et temps de calcul. Il est recommandé d'utiliser 20 à 100 chromosomes dans une population. Cependant, plus le nombre de chromosome est grand, plus la chance d'obtenir des résultats optimaux est meilleure. Par contre, le temps de calcul risque d'être considérable. Pour cela nous employons 80 ou 100 chromosomes par génération.

La fonction objectif

Le choix de la fonction objectif est l'une des étapes les plus difficiles dans l'application des algorithmes génétiques, elle permet d'évaluer les individus de la population. Dans la littérature, nous trouvons diverses fonctions objectives basées sur l'erreur. Ainsi, plusieurs indices de performances peuvent être définis :

- *L'intégrale de la valeur absolue de l'erreur (IAE):*

$$J_{IAE} = \int_0^{t_f} |e(t)| dt \quad (5.53)$$

IAE: Integral of Absolute Magnitude of the Error

- *L'intégrale du carré de l'erreur (ISE):*

$$J_{ISE} = \int_0^{t_f} (e(t))^2 dt \quad (5.54)$$

ISE: Integral of the Square of the Error

- *L'intégrale de la valeur absolue de l'erreur pondérée par le temps (ITAE):*

$$J_{ITAE} = \int_0^{t_f} t|e(t)| dt \quad (5.55)$$

ITAE: Integral of Time-Weighted Absolute Error

- *L'intégrale du carré de l'erreur pondérée par le temps (ITSE):*

$$J_{ITSE} = \int_0^{t_f} t(e(t))^2 dt \quad (5.56)$$

ITSE: Integral of Time-Weighted Square Error

Choix de la méthode de sélection

La toolbox des AGs dans MATLAB propose plusieurs techniques de sélection. Deux d'entre elles sont les plus connues et les plus utilisées : à savoir la sélection par roulette et la sélection par tournoi. Cette dernière nécessite un temps de calcul plus élevé que la première, ainsi nous avons choisi la technique de sélection par roulette.

Les opérateurs de reproduction

Après avoir sélectionné les parents, vient l'application des opérateurs de reproduction. Le premier est l'opérateur de croisement. La toolbox de MATLAB dispose aussi de plusieurs types de croisements : dont le croisement à un seul point, le croisement en deux points et le croisement dispersé (crossover scattered). Dans le croisement à un seul point ou à deux points, les génomes qui sont près les uns des autres tendent à survivre ensemble, tandis que des génomes qui sont distants tendent à être séparés. L'utilisation de la technique de croisement dispersé élimine cet effet. Chaque gène a une chance égale de venir de l'un ou l'autre parent. Ceci s'appelle parfois croisement uniforme ou aléatoire. La probabilité de croisement est fixée à 0.8.

Le deuxième opérateur qui intervient est l'opérateur de mutation, il sera appliqué avec une probabilité autour de 0.01.

Test d'arrêt

Nous possédons deux possibilités d'arrêt, la première consiste à fixer à priori le nombre de générations que doit effectuer l'algorithme, et la deuxième comporte à fixer le temps de calcul. Afin d'avoir un résultat optimal, nous avons essayé de combiner les deux critères d'arrêt, pour ne pas avoir un temps de calcul très grand et au même temps s'assurer que les résultats obtenus sont optimaux. Le nombre de générations est fixé généralement entre 100 à 300 générations.

Le rôle de l'AG est de déterminer les valeurs optimales des paramètres, afin de pouvoir contrôler notre robot manipulateur. Tel qu'il est montré dans la figure 5.10.

Les mêmes conditions de simulation utilisées précédemment sont valables pour cette partie.

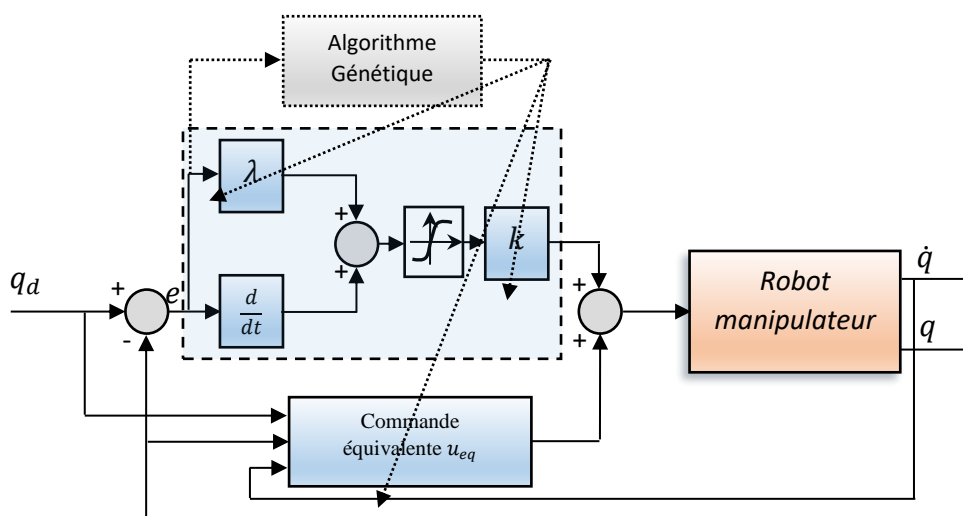


Figure 5.10. Principe d'optimisation par les algorithmes génétiques

Les fonctions objectives utilisées dans l'optimisation des paramètres sont basées sur les critères J_{MAE} et J_{MSE} donnés par les expressions suivantes :

$$J_{MSE} = \frac{1}{N} \sum_{i=1}^N (e(i))^2 \quad (5.57)$$

MSE: Mean of the Square of the Error

$$J_{MAE} = \frac{1}{N} \sum_{i=1}^N |e(i)| \quad (5.58)$$

MAE: Mean of Absolute Magnitude of the Error

Où $e(i)$ représente l'erreur à l'instant i et N est le nombre de mesure.

Les paramètres de l'algorithme génétique choisis sont :

Paramètres	Valeur
Taille de la population	100
Nombre maximal de génération	120
Probabilité de croisement	0.8
Probabilité de mutation	0.01
Nombre de variable	4
Fonction de sélection	Mécanisme de sélection par roulette

Tableau 5.1. Les paramètres de l'algorithme génétique

En utilisant les paramètres suscités de l'AG, l'optimisation du contrôleur nous a permis d'avoir les valeurs optimales des paramètres.

	k_1	k_2	λ_1	λ_2
MSE	860.9233	285.8091	17.5198	17.5198
MAE (IAE)	872.8699	203.7020	12.6459	12.6459

Tableau 5.2. Les paramètres obtenus

Les résultats de simulation sont illustrés par les figures (5. 13) et (5. 14).

5.5.2. Optimisation par la méthode PSO de la commande par modes glissants

Les paramètres du contrôleur par mode glissant sont maintenant optimisés en utilisant la méthode d'optimisation basée sur le PSO. En considérant toujours les caractéristiques et les performances du système à contrôlé.

La première étape consiste à coder le problème pour construire la population dans l'algorithme PSO. Tous les paramètres à optimiser du contrôleur peuvent présenter une solution possible du problème. La convergence vers les paramètres optimaux est dirigée par une fonction de fitness (une fonction objectif ou fonction de coût). Par conséquent, cette fonction, qui est basée sur l'erreur dans notre cas, doit être correctement définie avant l'exécution de l'algorithme PSO. Dans ce travail, deux fonctions de fitness basées sur l'erreur quadratique J_{MSE} et l'erreur absolue J_{MAE} sont considérées. Et sont données par les équations suivantes :

$$fitness(MSE) = \frac{1}{N} \sum_{i=1}^N (e(i))^2 = \frac{1}{N} \sum_{i=1}^N (erreur)^2 \quad (5.59)$$

$$fitness(MAE) = \frac{1}{N} \sum_{i=1}^N |e(i)| = \frac{1}{N} \sum_{i=1}^N |erreur| \quad (5.60)$$

Avec $e(i)$ représente l'erreur de poursuite à la $i^{ème}$ période, N est le nombre de période.

L'organigramme de l'algorithme PSO est donné dans la figure (5.11). Cet organigramme nous a servi pour la réalisation du programme de l'algorithme PSO.

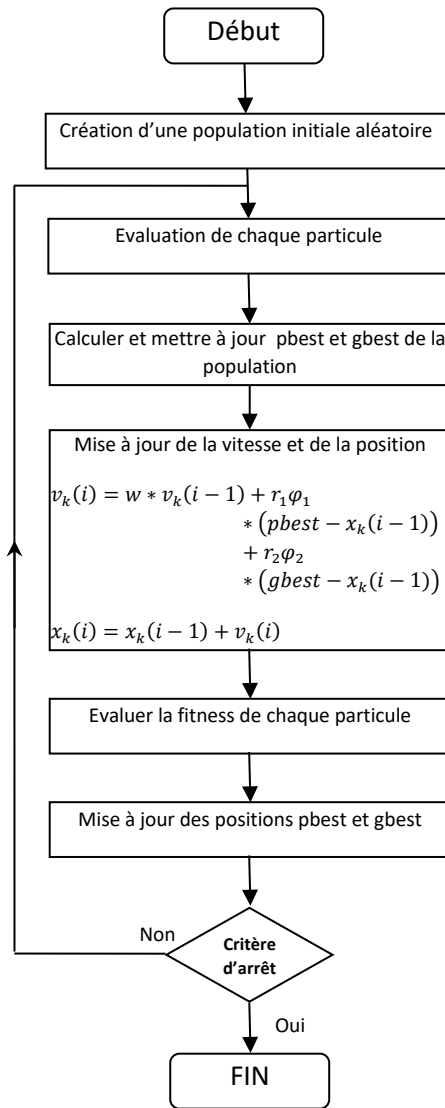


Figure 5.11. Organigramme de l’algorithme du PSO

La figure (5.12) montre la structure du principe d’optimisation du contrôleur par mode glissant à l’aide de l’algorithme PSO.

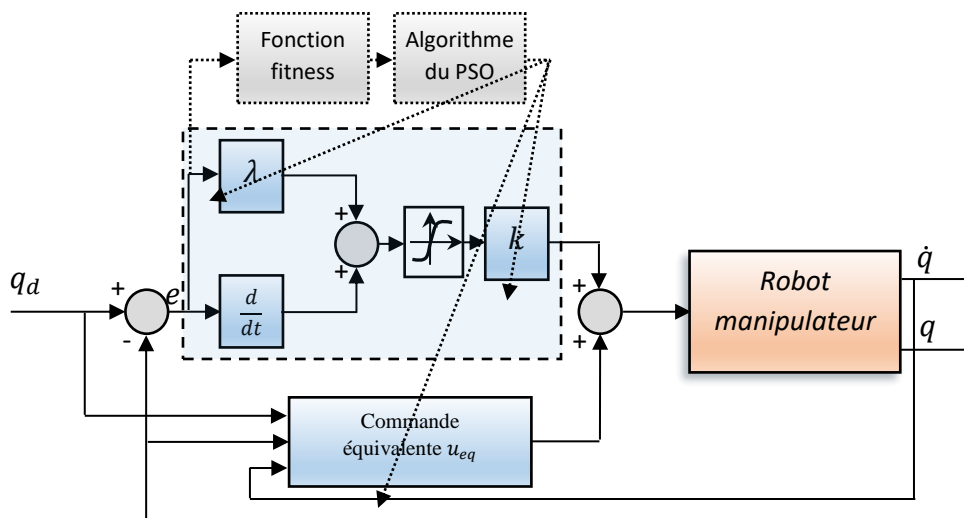


Figure 5.12. Principe d’optimisation par les algorithmes génétiques

Pour générer les trajectoires de références, les valeurs désirées sont toujours égales à $q_1^{fin} = 1.1 \text{ rad}$, $q_2^{fin} = 1.3 \text{ rad}$ et $q_1^{init} = q_2^{init} = 0 \text{ rad}$

Les valeurs des paramètres PSO sont données par le tableau 5.3:

Paramètres	Valeur
Coefficient d'accélération vers la meilleure position de soi-même (constante cognitive φ_1)	2
Coefficient d'accélération vers la meilleure position globale (constante du groupe φ_2)	2
Le coefficient d'inertie (w)	1
Nombre de particules (n_p)	80
Nombre d'itération (n_{it})	150

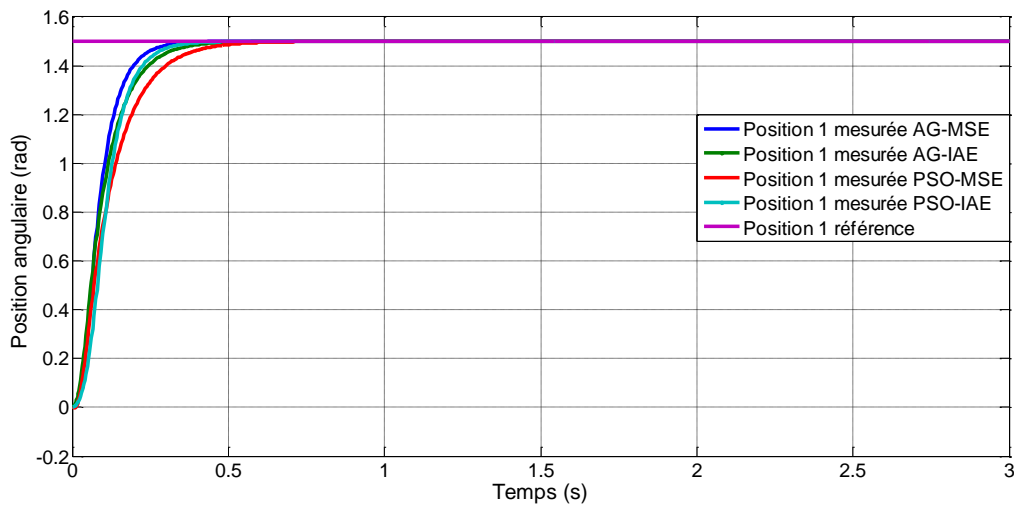
Tableau 5.3. Les paramètres de l'algorithme PSO

En utilisant les paramètres suscités du PSO, l'optimisation du contrôleur nous a permis d'avoir les valeurs optimales des paramètres.

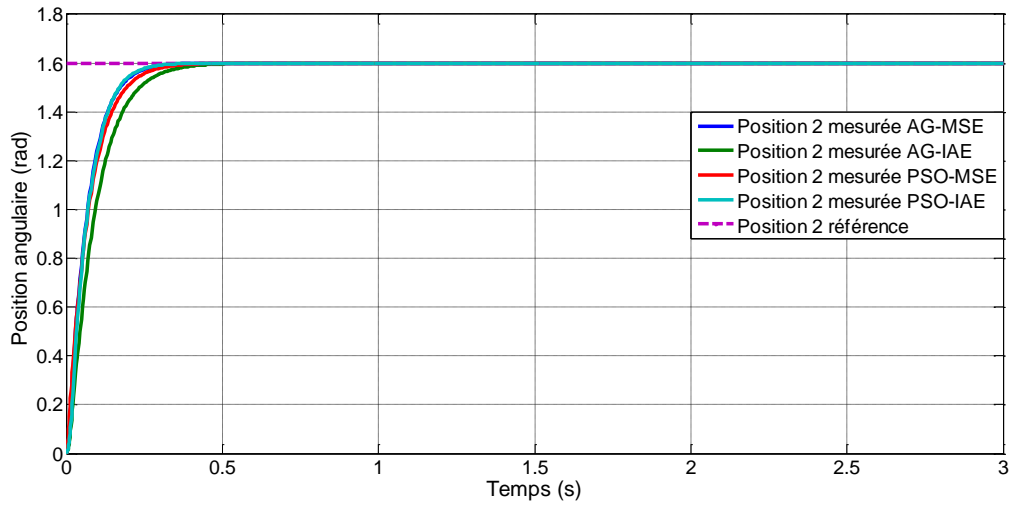
	k_1	k_2	λ_1	λ_2
MSE	756.59	445.1520	10.0703	14.8274
MAE (IAE)	571.58	183.4991	16.6846	19.0656

Tableau 5.4. Les paramètres obtenus par PSO

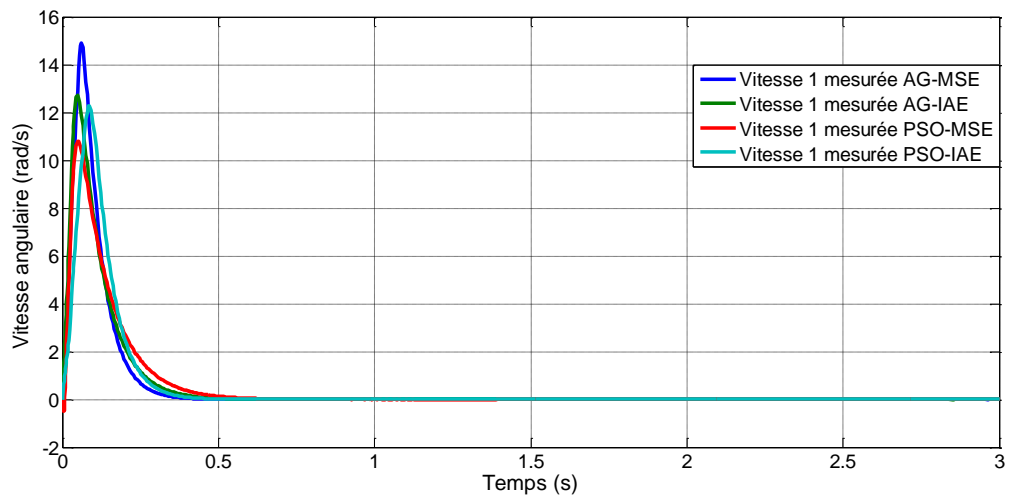
Les figures (5. 13) et (5. 14) illustrent Les résultats de simulation.



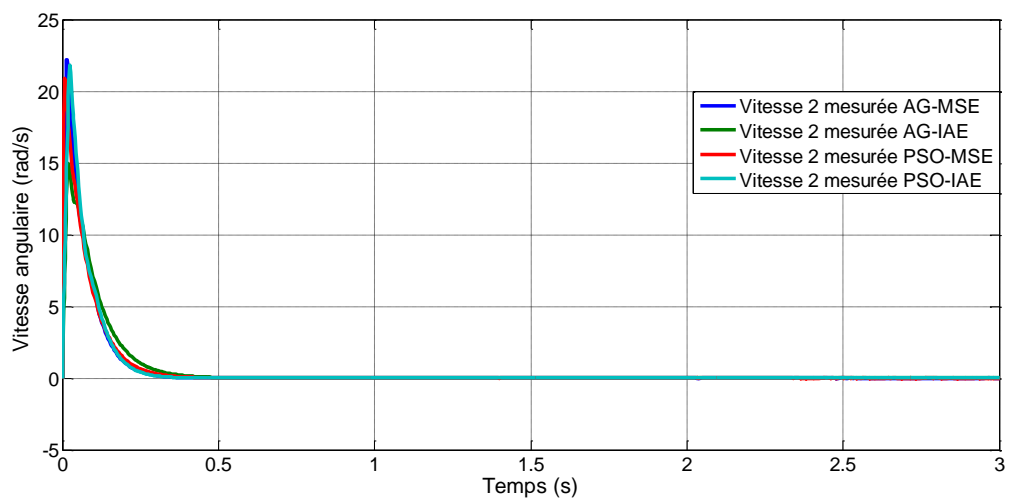
(A)



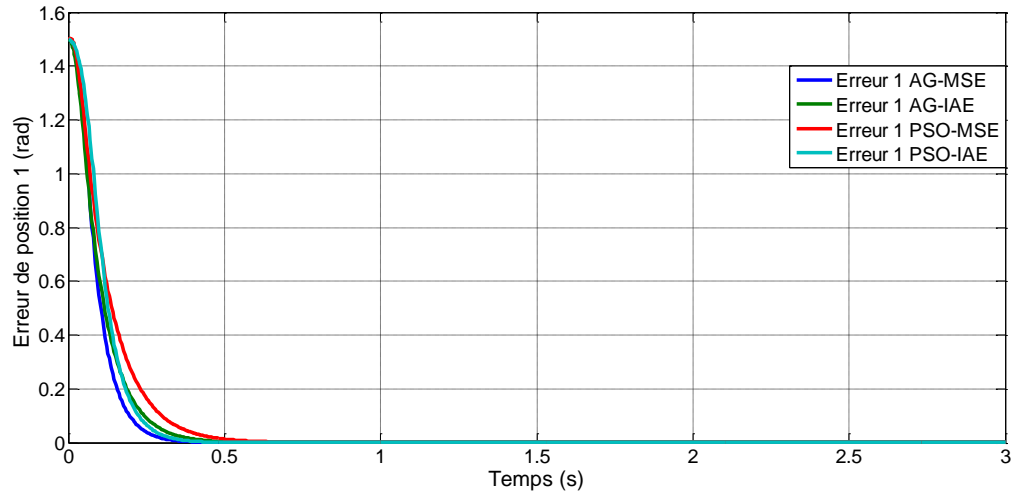
(B)



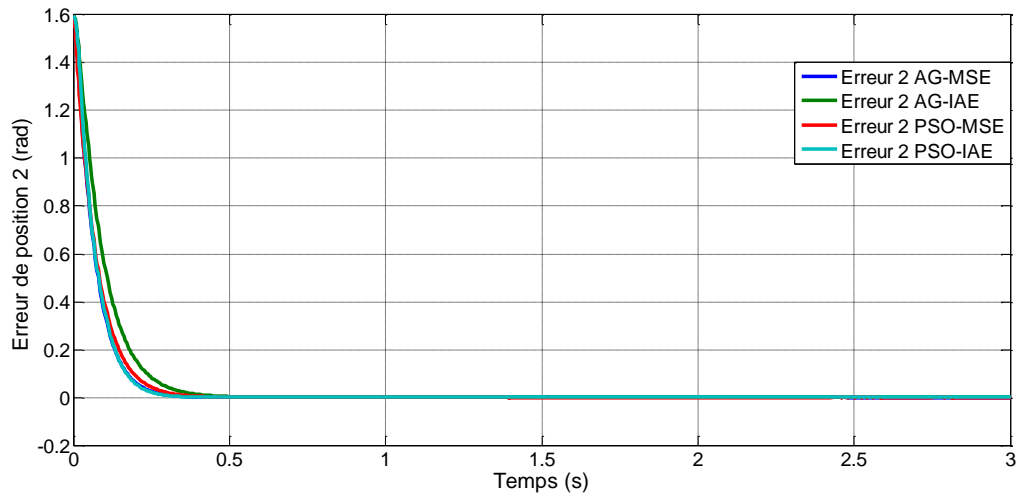
(C)



(D)

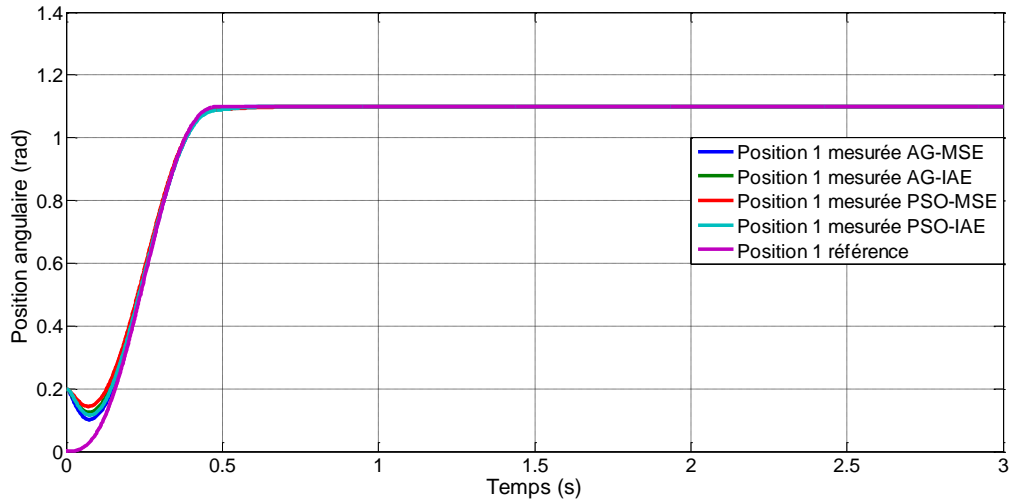


(E)

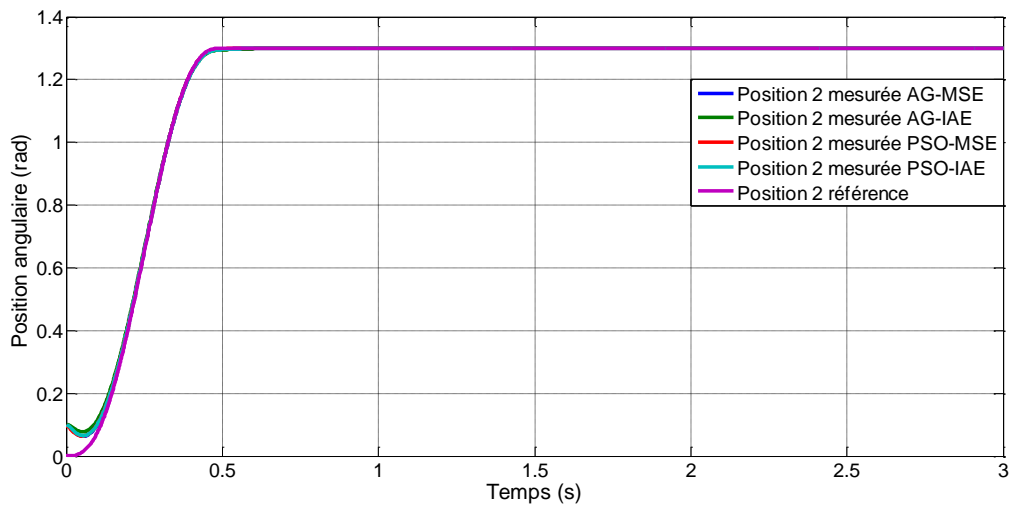


(F)

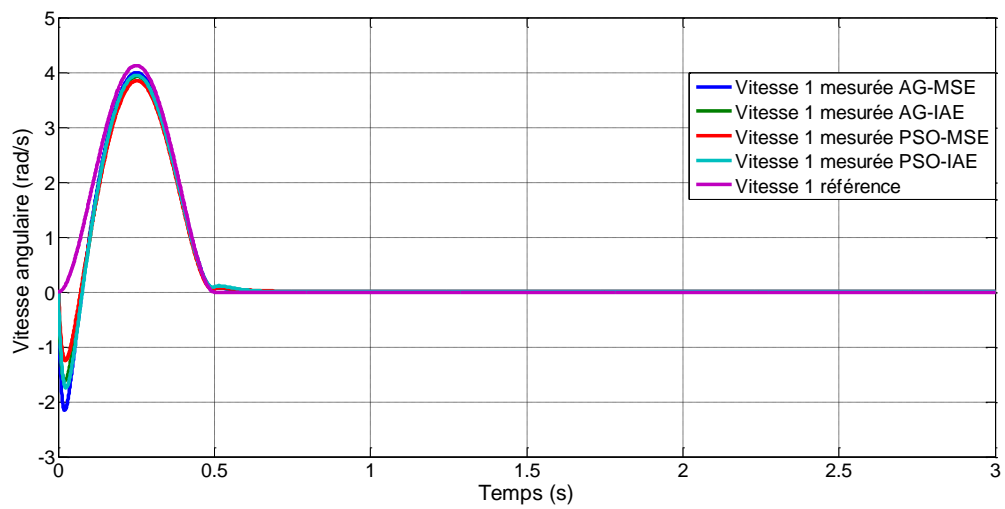
Figure 5.13. Commande par Modes glissants optimisée par AG et PSO : (A) et (B) positions angulaires, (C) et (D) vitesses angulaires, (E) et (F) erreurs des positions angulaires.



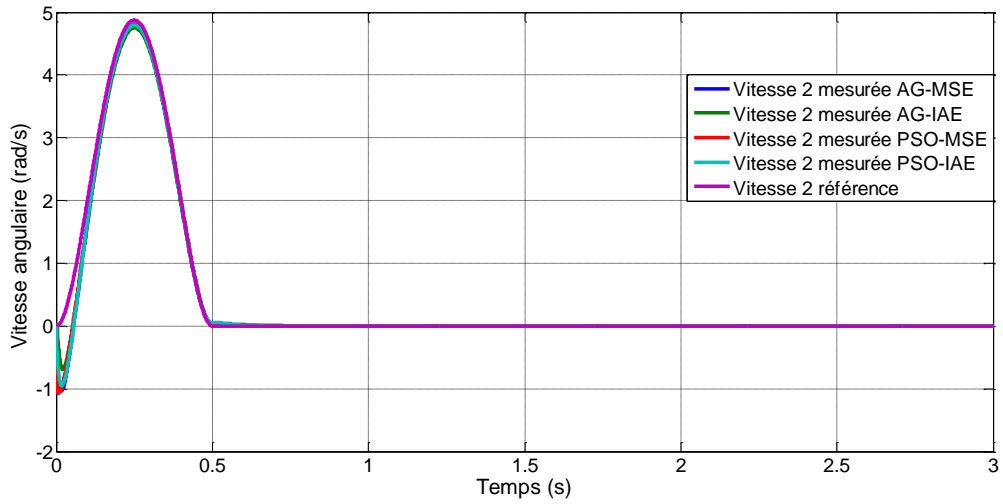
(A)



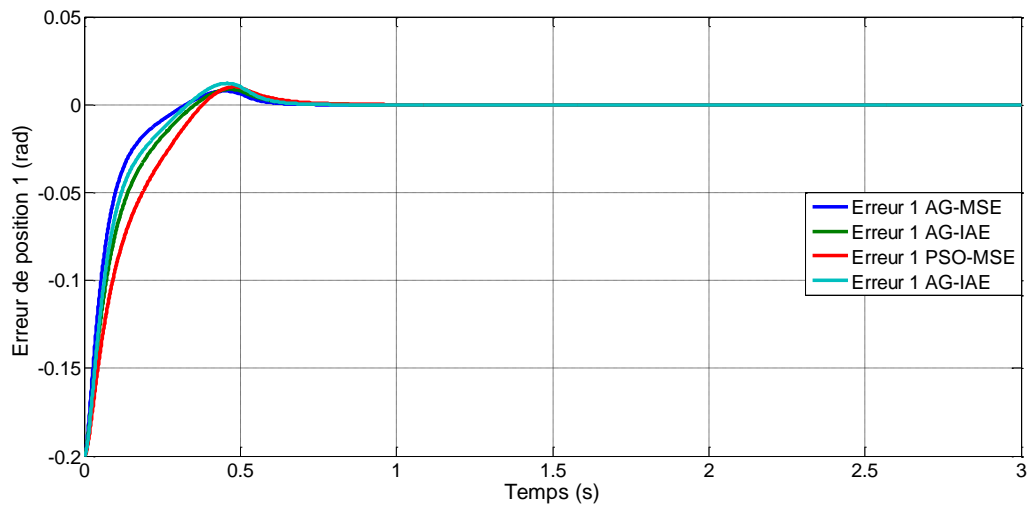
(B)



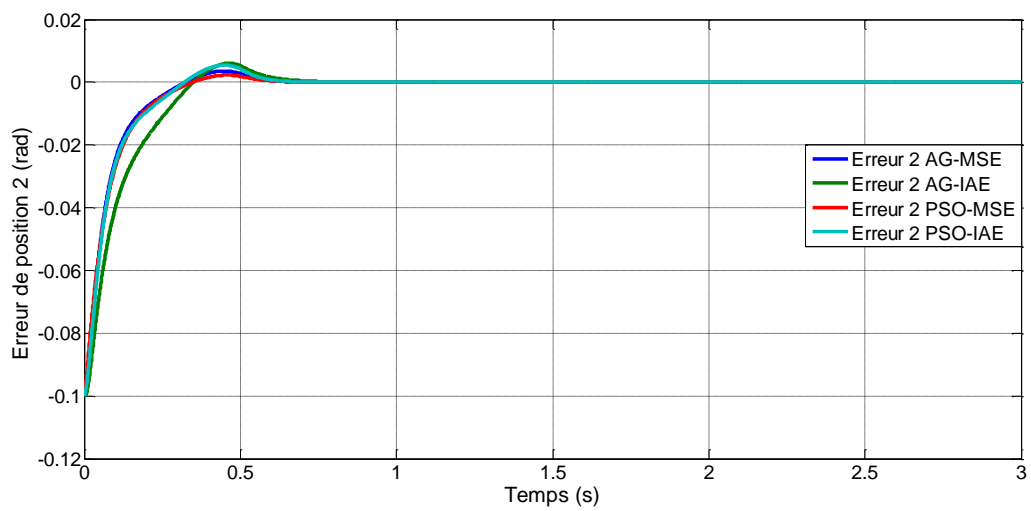
(C)



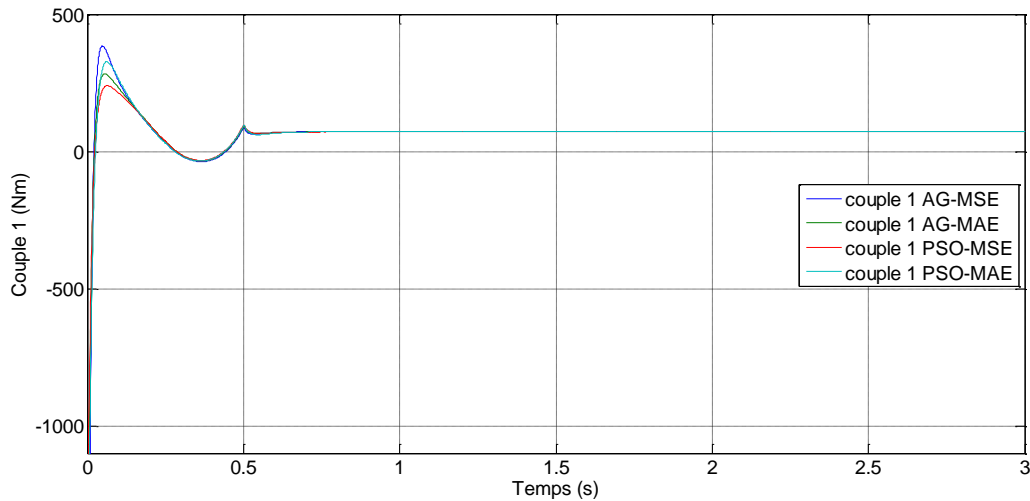
(D)



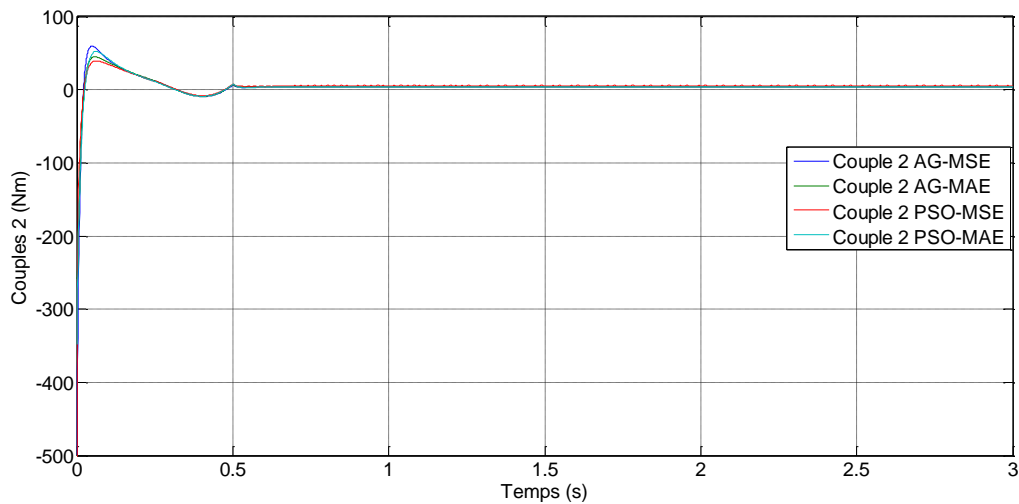
(E)



(F)



(G)



(H)

Figure 5.14. Commande par Modes glissants optimisée par AG et PSO en poursuite de trajectoire avec conditions initiales $[0.2, 0.1]$: (A) et (B) positions angulaires, (C) et (D) vitesses angulaires, (E) et (F) erreurs des positions angulaires, (G) et (H) couples de commande.

	AG-MSE		AG-MAE (IAE)		PSO-MSE		PSO-MAE (IAE)	
	1 ^{er}	2 ^{ème}	1 ^{er}	2 ^{ème}	1 ^{er}	2 ^{ème}	1 ^{er}	2 ^{ème}
Temps de réponse	0.21s	0.19s	0.265s	0.25s	0.325	0.21	0.24	0.19
Temps de calcul	Très grand		grand		moyen		moyen	

Tableau 5.5. Tableau de comparaison entre les techniques d'optimisation appliquées au MG

Synthèse 3

Les figures (5.13.A) et (5.13.B) montrent que les positions articulaires dans les quatre cas rejoignent leurs positions désirées pour la première et la deuxième articulation respectivement. Ceci est confirmé à travers les erreurs des positions qui convergent tous vers zéro, ce qui est montré sur les figures (5.13.E) et (5.13.F). Nous avons une phase d'accélération qui correspond

au démarrage puis une phase de décélération lorsque les positions articulaires se rapprochent des positions désirées, ce qui peut être remarqué dans les figures (5.13.C) et (5.13.D) montrant les vitesses articulaires.

En poursuite de trajectoire nous avons opté pour un changement des conditions initiales pour les quatre cas, 0.2 rad pour la première articulation et 0.1 rad pour la deuxième articulation, malgré cela, les positions rejoignent leurs trajectoires de références ce qui est visible sur les figures (5.14.A) et (5.14.B).

Comme nous pouvons le remarquer en introduisant les deux méthodes d'optimisation les AGs et PSO, il y a une nette amélioration du temps de réponse par rapport aux résultats obtenus par la méthode essai-erreur données dans la 4^{ème} partie de ce chapitre, tel qu'il montré dans les figures (5.13.A) et (5.13.B), et ceci pour les deux articulations.

Tel qu'il montré sur le Tableau comparatif 5.5 et les figures (5.13.A) et (5.13.B), l'application de la méthode AG avec le critère MSE comme fonction objectif a donné le meilleur résultat en terme de temps de réponse pour les deux articulations 0.21s et 0.19s respectivement, mais en termes de temps de calcul cette technique est très couteuse. Par contre l'utilisation de la méthode PSO est moins couteuse en termes de temps de calcul avec des résultats satisfaisants, tel que le PSO-MAE. Il faut noter aussi que en augmentant la taille de la population dans l'AG qui est égale à 100 dans notre cas, le temps de calcul devient encore plus important, ce qui est moins important lorsqu'on augmente le nombre de particules dans le PSO.

Vu la nature du système a commandé, qui est non linéaire et fortement couplé, ceci rend la tâche d'optimisation très délicate à cause du couplage entre les deux articulations, parce qu'on risque de gagner sur une articulation et perdre sur l'autre et vis-versa.

5.5.3. Optimisation par la méthode des AGs de la commande Backstepping

L'application de la commande Backstepping décrite à travers les équations (5.51) et (5.52) nécessite la détermination de quatre paramètres : k_1 , k_2 , k_3 et k_4 . Le choix de ces paramètres est effectué en utilisant la méthode d'optimisation par les AGs en tenant compte des réponses de notre système.

Les paramètres à optimiser k_1 , k_2 , k_3 et k_4 du régulateur sont maintenant codés en binaire avec des longueurs finies. Les chaînes de caractères représentant ces paramètres sont juxtaposées afin de construire un chromosome (figure 5.15). Ces chromosomes représentent les solutions possibles du problème.

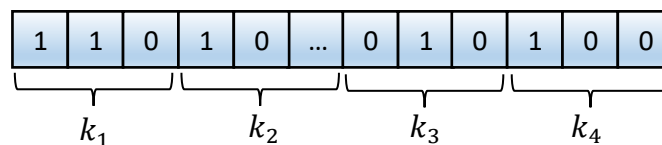


Figure 5.15. La structure du chromosome

Les mêmes étapes de l'AG vu précédemment vont être appliquées dans cette partie. Le principe de l'application des AGs afin de trouver les valeurs optimales du contrôleur est illustré dans la figure (5.16).

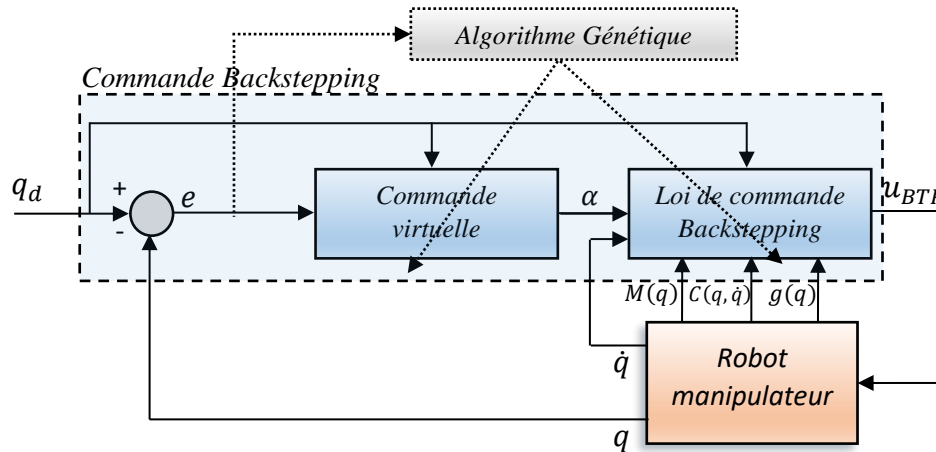


Figure 5.16. Principe d'optimisation de la commande Backstepping par les AGs

Nous avons utilisé comme fonction objectif deux critères de performances J_{MSE} et J_{IAE} basée sur l'erreur quadratique et l'erreur absolue.

Les paramètres de l'algorithme génétique considérés dans ce cas sont :

Paramètres	Valeur
Taille de la population	80
Nombre maximal de génération	120
Probabilité de croisement	0.8
Probabilité de mutation	0.01
Nombre de variable	4
Fonction de sélection	Mécanisme de sélection par roulette

Tableau 5.6. Les paramètres de l'algorithme génétique

En utilisant les paramètres de l'AG cités dans le tableau 5.6, nous avons pu avoir les valeurs optimales des paramètres, donnés dans le tableau suivant :

	k_1	k_2	k_3	k_4
MSE	31.1761	74.9924	29.5065	17.8122
MAE (IAE)	24.5468	28.4141	34.3054	49.7622

Tableau 5.7. Les paramètres obtenus par AG

Les résultats de simulation sont illustrés par les figures (5. 18) et (5. 19).

5.5.4. Optimisation par la méthode PSO de la commande Backstepping

La méthode d'optimisation PSO est utilisée dans ce cas pour trouver les paramètres optimaux de la commande Backstepping. En tenant compte des performances et des caractéristiques du système.

Afin de construire la population des particules intervenant dans l'algorithme du PSO nous avons codé le problème. Chaque particule de cette population peut représenter une solution au problème d'optimisation. La fonction fitness qui conditionne la convergence des solutions possibles vers les paramètres optimaux sont définies par les critères de performances J_{MSE} et J_{IAE} .

La structure du principe d'optimisation par PSO donné par l'organigramme illustré par la figure (5.11), appliquée à la commande Backstepping est représentée sur la figure (5.17).

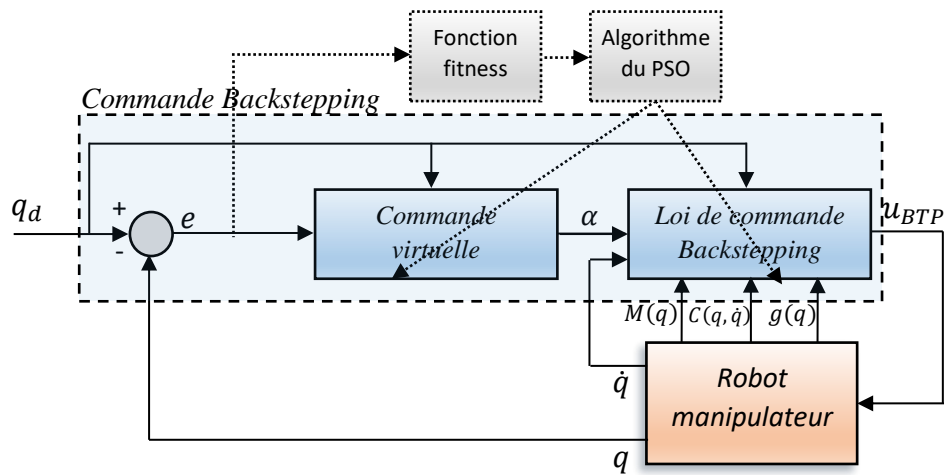


Figure 5.17. Principe d'optimisation de la commande Backstepping par PSO

Les paramètres de l'algorithme du PSO choisis sont donnés par le tableau 5.8 :

Paramètres	Valeur
Coefficient d'accélération vers la meilleure position de soi-même (constante cognitive φ_1)	2
Coefficient d'accélération vers la meilleure position globale (constante du groupe φ_2)	2
Le coefficient d'inertie (w)	1
Nombre de particules (n_p)	80
Nombre d'itération (n_{it})	120

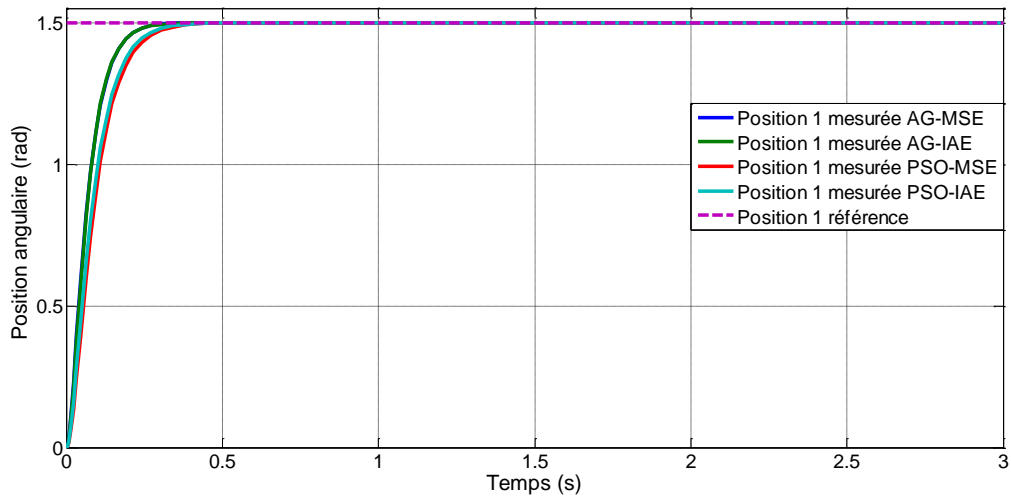
Tableau 5.8. Les paramètres de l'algorithme PSO

Les valeurs optimales des paramètres trouvés en utilisant les paramètres du PSO cités dans le tableau 5.8, sont données dans le tableau suivant :

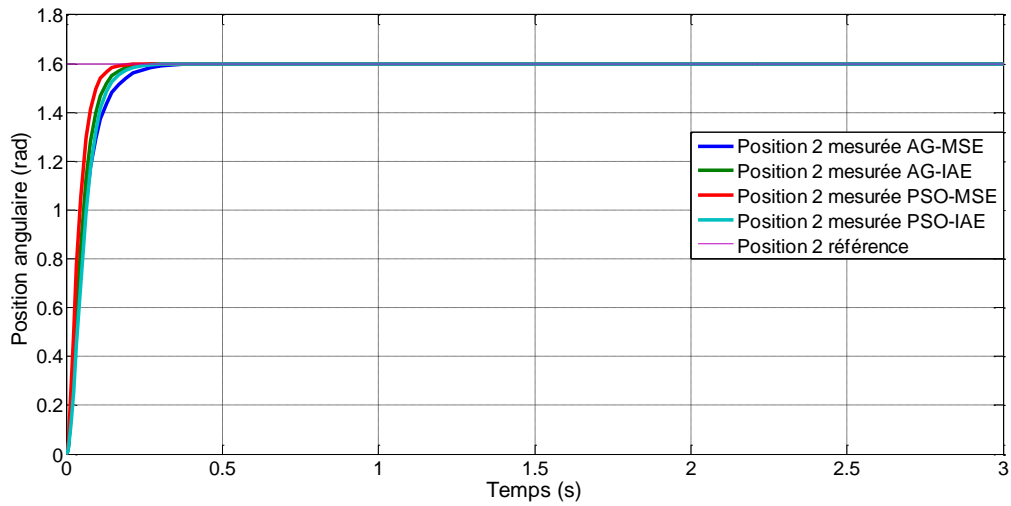
	k_1	k_2	k_3	k_4
MSE	17.3145	35.9781	28.1796	66.4746
MAE (IAE)	28.7320	44.6633	20.2356	21.0396

Tableau 5.9. Les paramètres obtenus par PSO

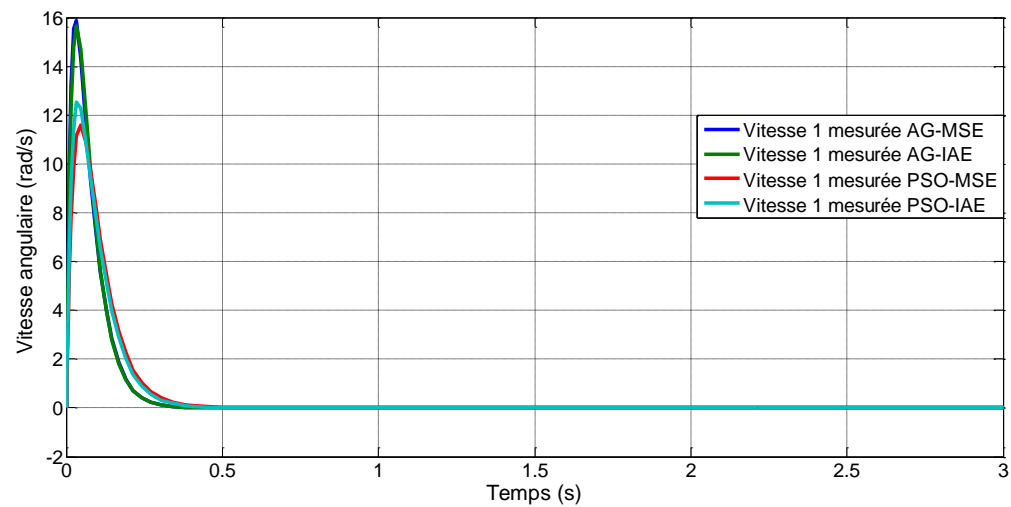
Les figures (5. 18) et (5. 19) montrent Les résultats de simulation.



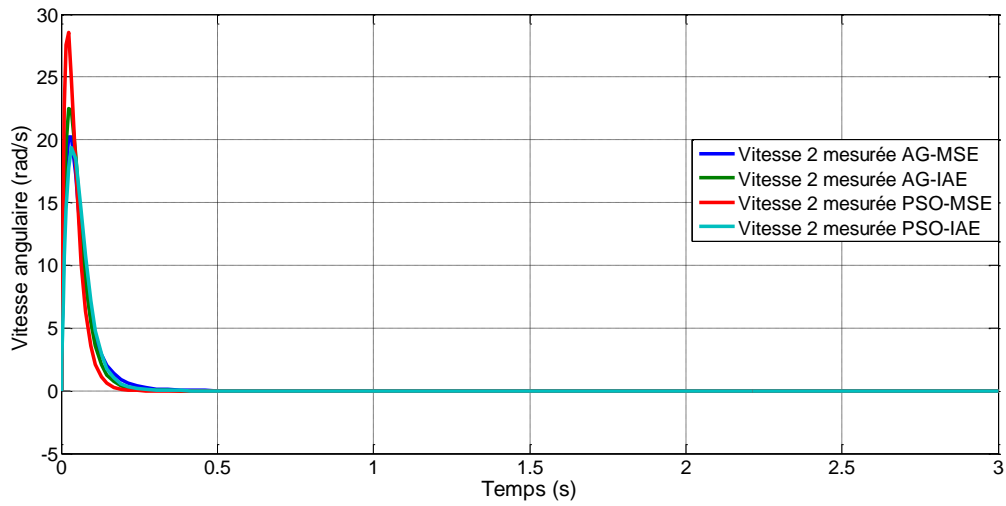
(A)



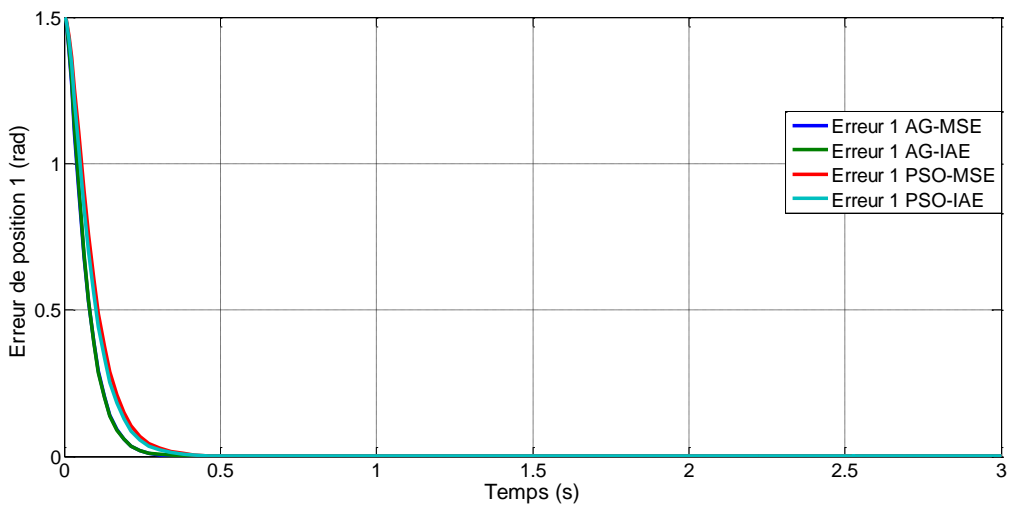
(B)



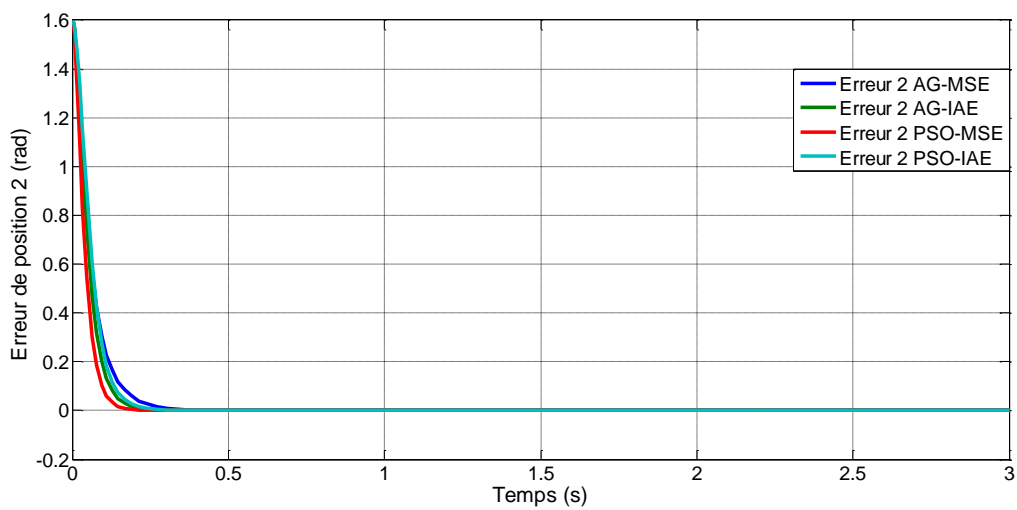
(C)



(D)

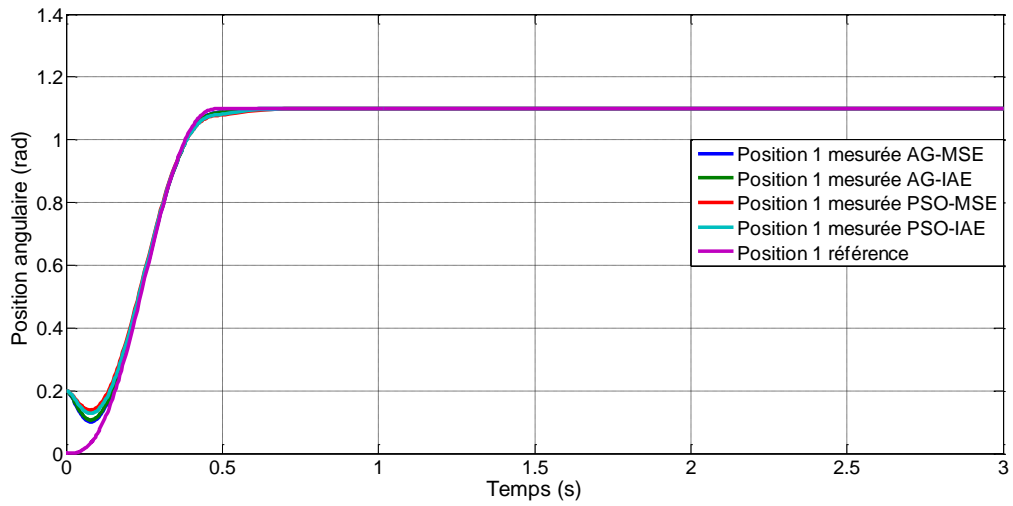


(E)

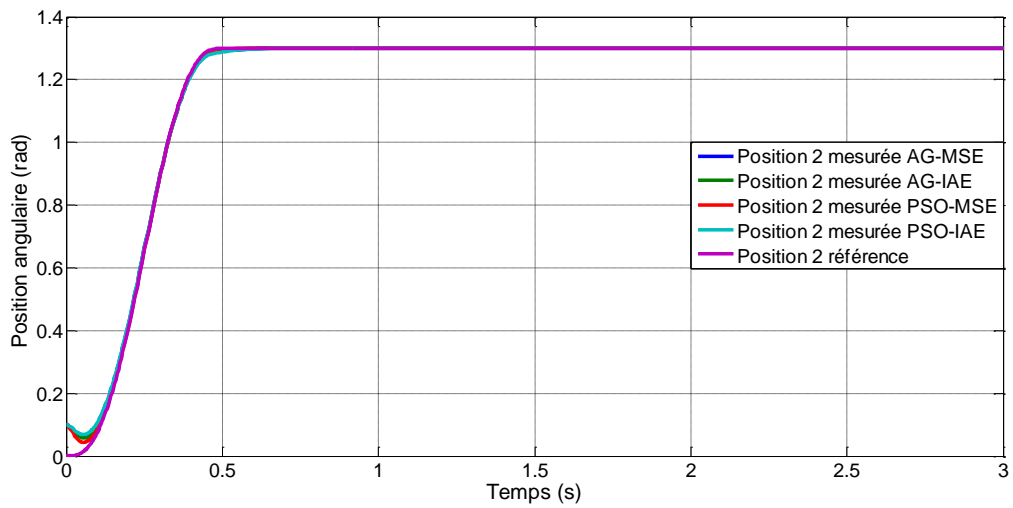


(F)

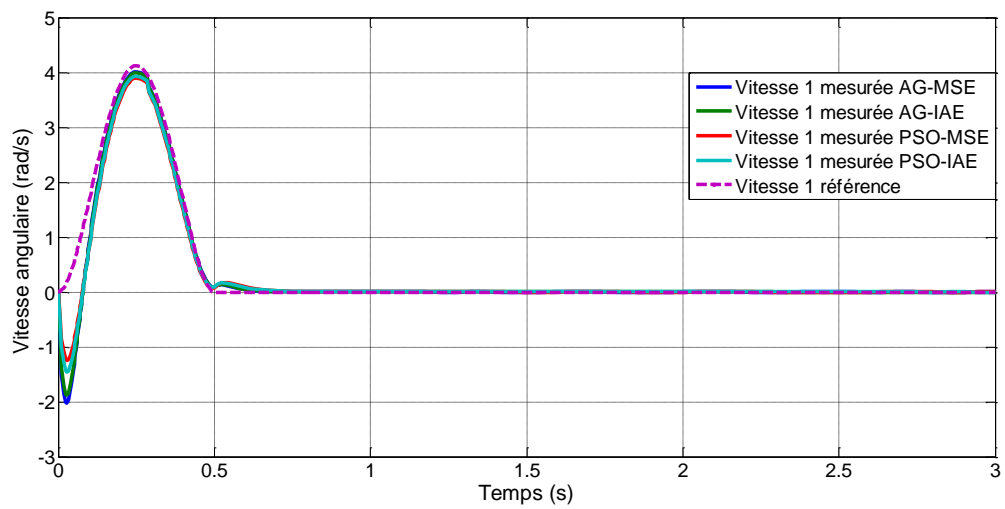
Figure 5.18. Commande par Backstepping optimisée par AG et PSO : (A) et (B) positions angulaires, (C) et (D) vitesses angulaires, (E) et (F) erreurs des positions angulaires.



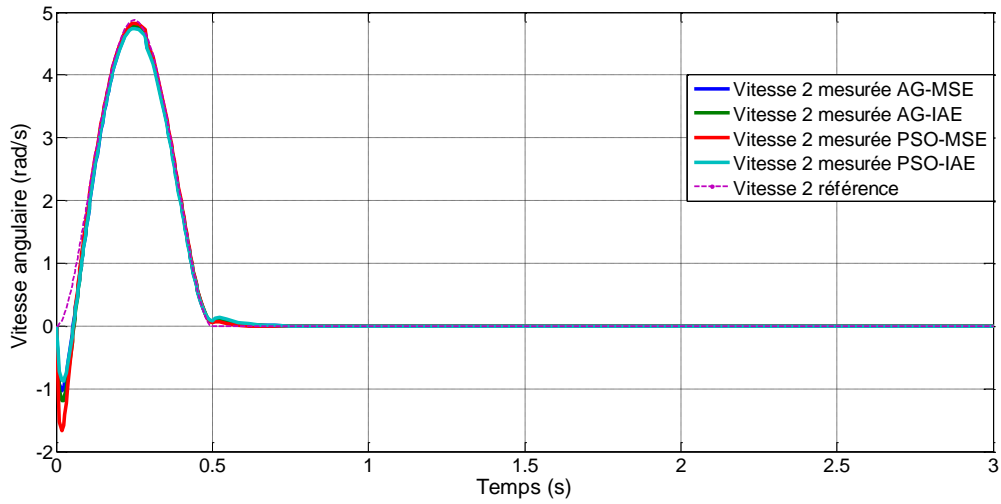
(A)



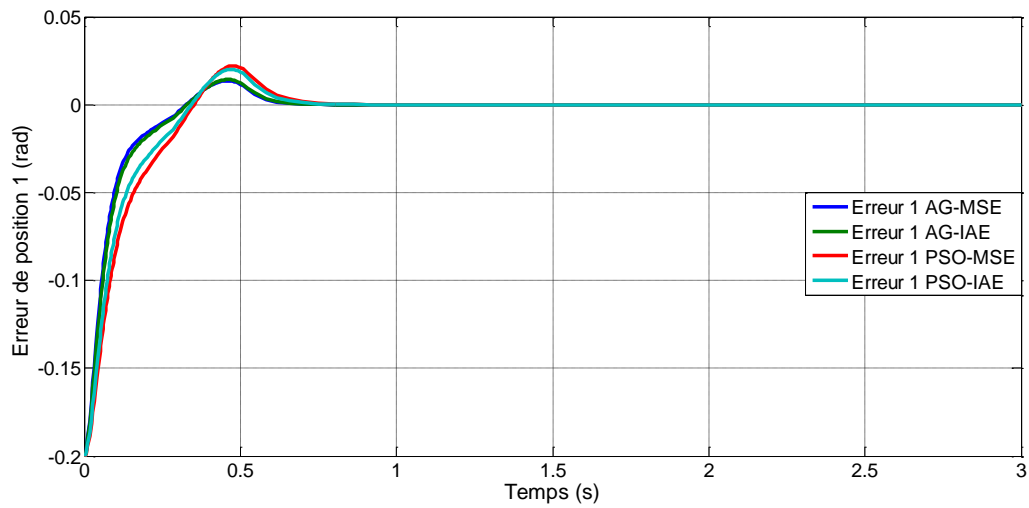
(B)



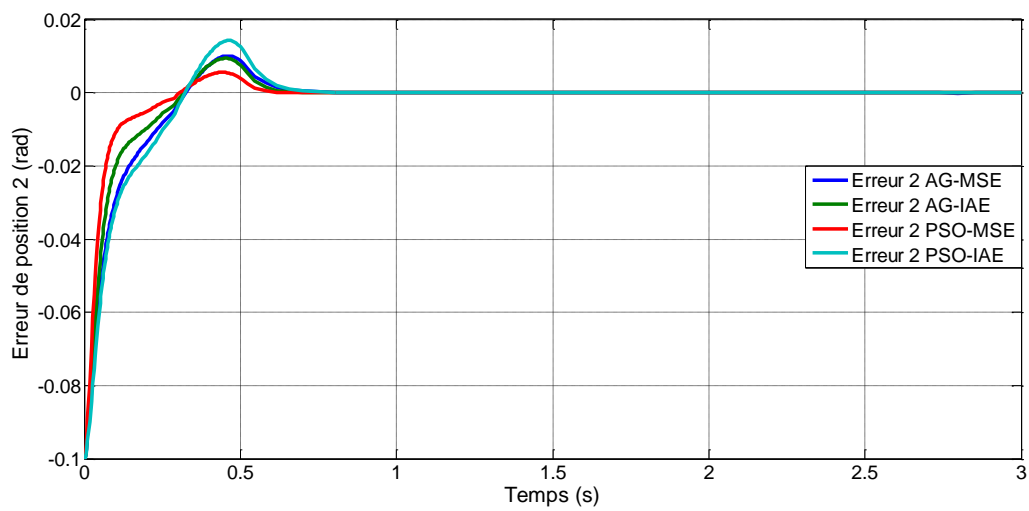
(C)



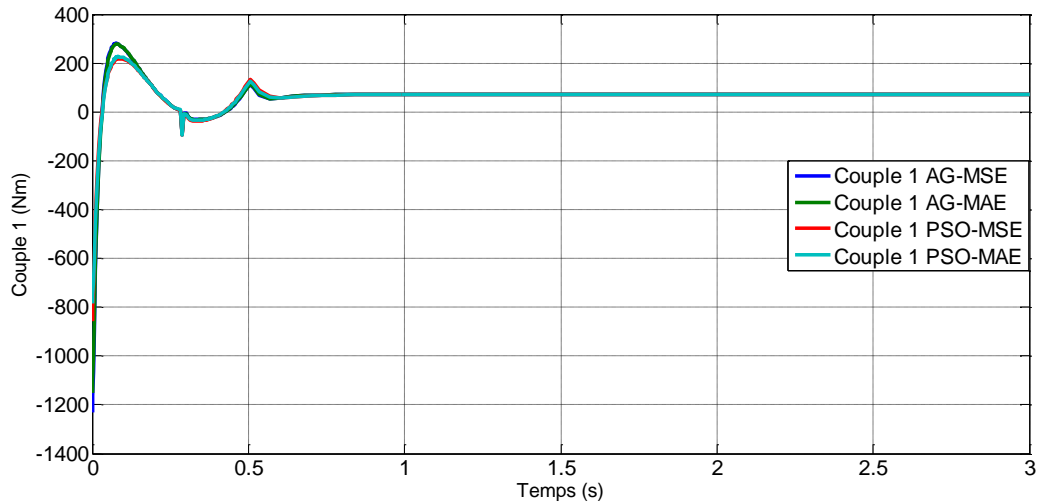
(D)



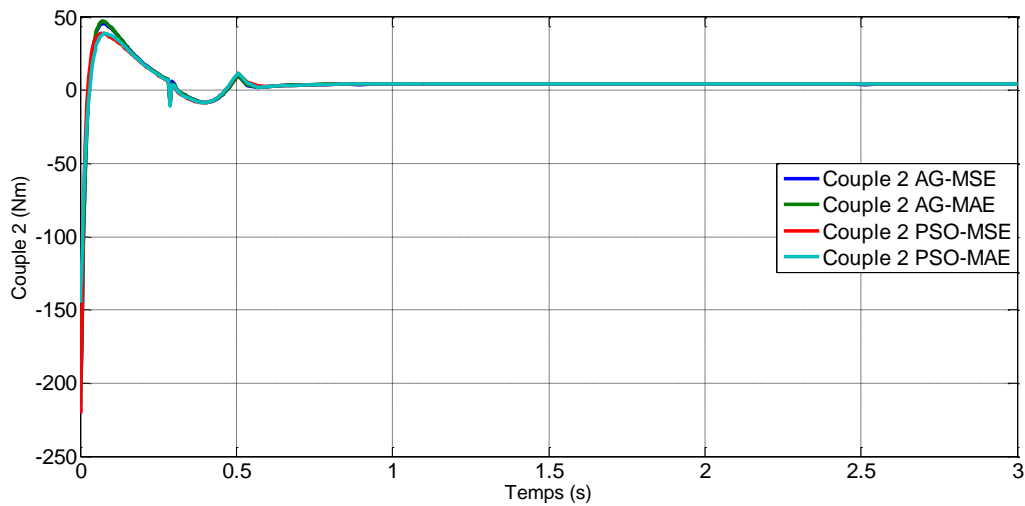
(E)



(F)



(G)



(H)

Figure 5.19. Commande par Backstepping optimisée par AG et PSO en poursuite de trajectoire avec conditions initiales [0.2 , 0.1] : (A) et (B) positions angulaires, (C) et (D) vitesses angulaires, (E) et (F) erreurs des positions angulaires, (G) et (H) couples de commande.

	AG-MSE		AG-MAE (IAE)		PSO-MSE		PSO-MAE (IAE)	
	1er	2ème	1er	2ème	1er	2ème	1er	2ème
Temps de réponse	0.18s	0.17s	0.18s	0.13s	0.23	0.11	0.22	0.145
Temps de calcul	grand		grand		moyen		moyen	

Tableau 5.10. Tableau de comparaison entre les techniques d’optimisation appliquées au Backstepping

Synthèse 4

Les résultats de simulations illustrés dans les figures (5.18.A) et (5.18.B) montrent une légère amélioration du temps de réponse par rapport à l’utilisation de la commande par mode glissant, ce qui peut être justifié par l’utilisation de la fonction « tanh » dans cette dernière. Nous

pouvons confirmer ce résultat à partir des courbes des erreurs et des vitesses (5.18.C) au (5.18.F). Nous constatons aussi une bonne poursuite de trajectoire des positions et des vitesses articulaires (voir les figures (5.19.A) au (5.19.D)), avec une légère augmentation de l'erreur de poursuite tel qu'il est montré sur les figures (5.19.E) et (5.19.F).

L'optimisation en utilisant la méthode des AGs a donnée des résultats légèrement meilleurs que celles obtenus en utilisant l'algorithme du PSO, ce qui peut être constaté à partir du tableau de comparaison 5.10, en tenant compte de la nature non linéaire et fortement couplé du système à contrôlé.

Le temps de calcul consommé, en utilisant la méthode des AGs est beaucoup plus important que celui en utilisant le PSO. L'augmentation du nombre de particule dans l'algorithme du PSO n'a pas un grand effet sur le temps de calcul, par contre dans les AGs en augmentant la taille de la population le temps de calcul devient plus important.

Tel qu'il est illustré sur le Tableau de comparaison 5.10 ainsi que les figures (5.18.A) et (5.18.B), l'utilisation du critère MSE ou du critère MAE n'a pas vraiment d'influence sur la qualité du résultat, ce qui peut être remarqué sur la première articulation.

5.6. Conclusion

Les lois de commandes développées dans ce chapitre, en utilisant les techniques de commandes robustes par modes glissants et par Backstepping, appliqués au robot manipulateur ont donnés des résultats satisfaisant en terme de régulation et en terme de poursuite des trajectoires de références des positions et des vitesses angulaires. Même en présence de perturbations extérieures et des variations paramétriques nous avons constaté que le système garde pratiquement les mêmes performances. Les résultats de simulation montrent que le phénomène de broutement « chattering » rencontré lors de l'utilisation de la fonction « sign » dans la commande par modes glissants a pu être réduit en la remplaçant par la fonction « tanh ».

Cependant la conception de ces contrôleurs nécessite la détermination de certains gains, d'où l'utilisation de quelques méthodes d'optimisation tels que les algorithmes génétiques et le PSO. La majorité des applications qui utilisent ces techniques d'optimisation sont de nature hors lignes. Il est à noter que l'utilisation des algorithmes génétiques présente plusieurs difficultés, notamment pour le choix de la taille de la population et le nombre de génération à effectuer, ce qui conditionne la rapidité d'exécution et la qualité des résultats. Nous avons remarqué que plus la taille de la population est grande plus le temps d'exécution de l'algorithme est long, ce qui nécessite un calculateur puissant. Par contre si cette taille est inférieure à 20 nous avons obtenu de mauvais résultats, ce qui signifie que nous ne sommes pas tombés sur un optimum global. En tant que technique apparaissant il y a peu de temps, l'algorithme PSO a attiré l'attention ces dernières années. C'est essentiellement un algorithme évolutif en essaim. Nous avons remarqué que sa convergence est relativement plus rapide vers la valeur optimale. En comparant les résultats des tests des deux fonctions objectifs utilisées MSE et MAE sur les deux commandes, nous pouvons constater que l'utilisation de la fonction fitness basée sur l'erreur quadratique donne de meilleurs résultats. Il faut noter aussi que la nature du système joue un rôle important dans l'efficacité et la rapidité de convergence des algorithmes d'optimisation.

CHAPITRE 6

COMMANDE NON LINÉAIRE D'UN ROBOT MANIPULATEUR ACTIONNÉ PAR DES MSAP

6.1. Introduction

Dans un objectif de s'approcher encore plus du modèle réel, nous devons prendre en considération les modèles des actionneurs qui génèrent les couples d'entrées du robot manipulateur.

Plusieurs types d'actionneurs électriques sont courants dans les applications robotisées, tels que les moteurs à courant continu, les moteurs asynchrones, les moteurs Brushless DC, etc [17] [124] [5] [15]. Dans notre travail nous nous intéressons principalement à un type particulier d'actionneur de par ses avantages le moteur synchrone à aimant permanent (MSAP) dans la motorisation des articulations du robot manipulateur. Ces types d'actionneurs permettent d'atteindre des performances satisfaisantes en raison de la puissance massique importante et de l'absence des limitations propres aux moteurs à collecteurs [16].

Donc, le modèle du système non linéaire global ou la dynamique du robot manipulateur avec la dynamique de l'actionneur seront développées. Mais avant cela nous donnerons quelques notions de base ainsi que le modèle du moteur synchrone à aimants permanents.

6.2. Description du MSAP

Les moteurs à aimants permanents (PM) sont de loin les plus gros utilisateurs de matériaux à aimants permanents, capturant 60% du marché des aimants permanents. L'utilisation de stratégies de contrôle modernes a également montré une nouvelle perspective dans l'utilisation des moteurs PM dans des applications de haute performance, y compris l'automatisation d'usine, la robotique, l'aérospatiale, etc. Les moteurs synchrones à aimants permanents (MSAP) sont des dispositifs de mouvement électromécanique de haute performance qui remplacent les servomoteurs à courant continu traditionnels et les machines asynchrones en raison de leur capacité de haute performance. La nécessité de hautes performances dans les systèmes avec MSAP augmente avec la demande des commandes de précision [125] [126].

6.2.1. Moteur synchrone à aimants permanents

Durant plusieurs décennies, le moteur à courant continu a dominé les entraînements en industrie, grâce à la facilité de commande qu'il offre qui est dû au découplage naturel du couple et du flux. Cependant, le grand inconvénient de ce type de moteur parmi d'autres est la présence du système balais collecteur, limitant ainsi de plus en plus son utilisation.

Le moteur synchrone à aimants permanents bénéficie en revanche d'un avantage déterminant par rapport au moteur à courant continu, est l'absence du contact glissant (système ballais collecteur). Ceci, permet de réduire les opérations de maintenance, l'augmentation de la vitesse, la fiabilité et la robustesse de l'actionneur. De plus, il y a absence d'étincelles, augmentant ainsi les domaines d'utilisation. Puisque c'est les aimants permanents qui produisent le flux d'excitation, les circuits de commande et de réglage sont plus simples que ceux du moteur asynchrone. On peut donc également atteindre des performances dynamiques élevées [127] [126].

6.2.2. Modélisation du MSAP

Le stator du MSAP et le moteur synchrone à rotor bobiné sont similaires. De plus, il n'y a pas de différence entre la force électromotrice arrière produite par l'aimant permanent et celle produite par une bobine excitée. Par conséquent, le modèle mathématique d'un MSAP est similaire à celui du moteur synchrone à rotor bobiné.

Les modèles présentés pour la machine tiennent compte d'hypothèses simplificatrices présentées ci-dessous [128] [129] [130] [16].

- Caractéristique magnétique de la machine linéaire (circuit magnétique non saturé).
- La variation des résistances des enroulements en fonction de la température négligeable.
- Distribution spatiale sinusoïdale des forces magnétomotrices d'entrefer. Cela revient à considérer la variation sinusoïdale des inductances mutuelle entre les enroulements statoriques en fonction de l'angle de leurs axes magnétiques.
- La symétrie de la machine
- L'effet de peau est négligé.
- Les pertes fer sont négligées.
- Les effets des encoches ne sont pas pris en compte

6.2.2.1. Equations électriques

Les équations électriques dans un repère fixe lié au stator sont décrites par :

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = R_s \begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \Phi_a \\ \Phi_b \\ \Phi_c \end{bmatrix} \quad (6.1)$$

Où :

R_s : est la résistance des enroulements statorique.

$[V_a \ V_b \ V_c]^T$ Les tensions des phases statoriques.

$[I_a \ I_b \ I_c]^T$: Les courants dans les phases statoriques.

$[\Phi_a \ \Phi_b \ \Phi_c]^T$: Les flux totaux à travers les bobines statoriques.

6.2.2.2. Equations magnétiques

Les relations entre les flux et les courants peuvent s'écrire sous une forme matricielle suivante:

$$[\Phi_{abc}] = [l_{ss}][I_{abc}] + [\Phi_f] \quad (6.2)$$

Dans un cas général, c'est-à-dire, les machines à pôles saillants (sans amortisseurs), la matrice $[l_{ss}]$ est une matrice d'inductances statoriques. Elle contient des termes constants que nous regroupons dans $[l_{s0}]$, et des termes variables dépendant de θ que nous regroupons dans $[l_{s1}(\theta)]$.

Elle peut s'écrire :

$$[l_{ss}] = [l_{s0}] + [l_{s1}] \quad (6.3)$$

Où

$$[l_{s0}] = \begin{bmatrix} l_{s0} & M_{s0} & M_{s0} \\ M_{s0} & l_{s0} & M_{s0} \\ M_{s0} & M_{s0} & l_{s0} \end{bmatrix} \quad (6.4)$$

$$[l_{s1}] = l_{s0} \begin{bmatrix} \cos(2\theta) & \cos 2\left(\theta - \frac{2\pi}{3}\right) & \cos 2\left(\theta - \frac{4\pi}{3}\right) \\ \cos 2\left(\theta - \frac{2\pi}{3}\right) & \cos 2\left(\theta - \frac{4\pi}{3}\right) & \cos(2\theta) \\ \cos 2\left(\theta - \frac{4\pi}{3}\right) & \cos(2\theta) & \cos 2\left(\theta - \frac{2\pi}{3}\right) \end{bmatrix} \quad (6.5)$$

Avec

M_{s0} : Inductance mutuelle entre deux phases statoriques

l_{s0} : Inductance propre d'une phase statorique

θ : caractérise la position angulaire du rotor par rapport au stator

$[\Phi_f]$: Est le flux crée par les aimants permanents supposés à répartition sinusoïdale le long de l'entrefer.

Par conséquent, les expressions des flux mutuels rotor phases statoriques s'écrivent:

$$[\Phi_f] = \phi_f \begin{bmatrix} \cos(\theta) \\ \cos\left(\theta - \frac{2\pi}{3}\right) \\ \cos\left(\theta - \frac{4\pi}{3}\right) \end{bmatrix} \quad (6.6)$$

ϕ_f : Est la valeur crête constante du flux crée par l'aimant lisse à travers les enroulements statorique,

La substitution de (6.2) dans (6.1) donne :

$$[V_{abc}] = [R][I_{abc}] + \frac{d}{dt}([l_{ss}][I_{abc}] + [\Phi_f]) \quad (6.7)$$

On remarque que le système (6.7) engendre des équations fortement non linéaires et couplées

6.2.2.3. Equations mécaniques

L'équation mécanique du moteur s'écrit :

$$C_{em} - C_r = J \frac{d\Omega}{dt} + f\Omega \quad (6.8)$$

La connaissance du couple électromagnétique est essentielle pour le dimensionnement de la commande et son expression définie par:

$$C_{em} = \frac{p}{2} [I]^T \left(\frac{\partial [L]}{\partial \theta} \right) [I] \quad (6.9)$$

$\Omega = \frac{w}{p}$: représente la vitesse mécanique de rotation de la machine, w est la pulsation électrique de rotation.

6.2.2.4. Transformation de PARK [61]

La majorité des travaux exploitent la transformée de Park, afin de supprimer les non linéarités d'induction dans les équations du modèle précédent. Le principe de la transformation de Park consiste à remplacer les enroulements réels par des enroulements fictifs (d, q). A travers la transformation de Park les équations dynamiques des moteurs à courant alternatif sont plus simples à représentées ce qui facilite leurs utilisation et leurs étude.

La transformation de Park est représentée comme suit :

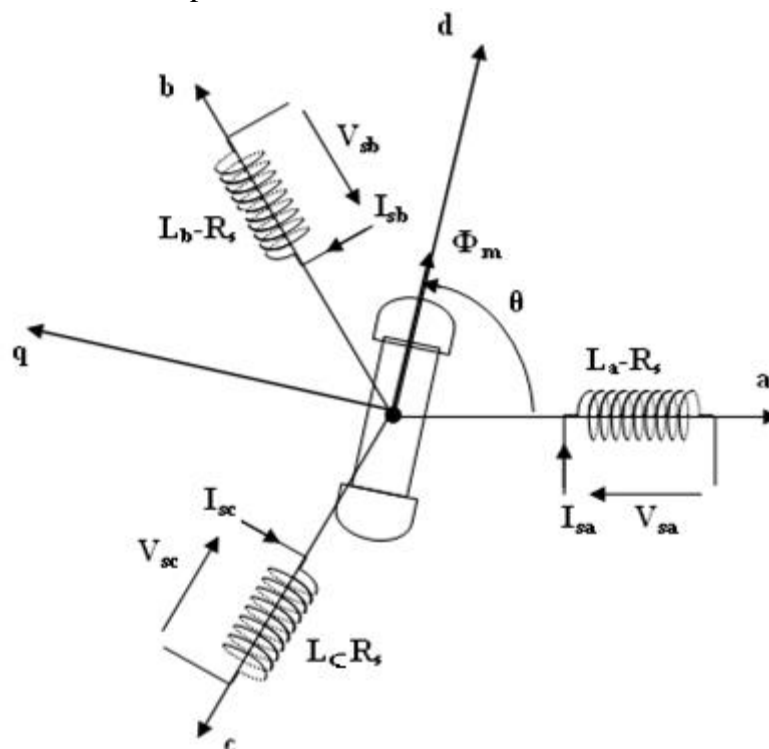


Figure 6.1. Schéma équivalent de la MS dans le référentiel de Park

$$\begin{bmatrix} x_{dq0} \end{bmatrix} = [T_\theta] \begin{bmatrix} x_{abc} \end{bmatrix} \quad (6.10)$$

Avec:

$[x]$ Peut représenter une tension, un courant ou un flux et θ est la position du rotor. Les termes x_d, x_q représentent les composantes directes et quadratures des variables statoriques (tensions, courants et flux).

La matrice de transformation $[T_\theta]$ est exprimée par :

$$[T_\theta] = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\theta) & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta - \frac{4\pi}{3}\right) \\ -\sin(\theta) & -\sin\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta - \frac{4\pi}{3}\right) \\ \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \end{bmatrix} \quad (6.11)$$

La matrice inverse dans le cas où T est orthogonal, $T^{-1}(\theta) = T^t(\theta)$

$$[T_\theta]^{-1} = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\theta) & -\sin(\theta) & \sqrt{\frac{1}{2}} \\ \cos\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta - \frac{2\pi}{3}\right) & \sqrt{\frac{1}{2}} \\ \cos\left(\theta - \frac{4\pi}{3}\right) & -\sin\left(\theta - \frac{4\pi}{3}\right) & \sqrt{\frac{1}{2}} \end{bmatrix} \quad (6.12)$$

Le passage du système triphasé réel au système (d, q) lié au rotor se fait en utilisant les relations suivantes :

$$\begin{aligned} [V_d \ V_q \ V_0] &= [T_\theta][V_a \ V_b \ V_c] \\ [I_d \ I_q \ I_0] &= [T_\theta][I_a \ I_b \ I_c] \\ [\Phi_a \ \Phi_b \ \Phi_0] &= [T_\theta][\Phi_a \ \Phi_b \ \Phi_c] \end{aligned} \quad (6.13)$$

Alors, le modèle de la machine synchrone à aimants permanents après la transformation de Park est donné par :

$$\begin{bmatrix} V_d \\ V_q \end{bmatrix} = \begin{bmatrix} R_s + L_d S & -p\Omega L_q \\ p\Omega L_d & R_s + L_q S \end{bmatrix} \begin{bmatrix} I_d \\ I_q \end{bmatrix} + \begin{bmatrix} 0 \\ p\Omega \phi_f \end{bmatrix} \quad (6.14)$$

Avec la relation entre flux et courant est définie par :

$$\begin{cases} \Phi_d = L_d I_d + \phi_f \\ \Phi_q = L_q I_q \end{cases} \quad (6.15)$$

6.2.2.5. Mise sous forme d'équation d'état

En considérant les tensions (V_d, V_q) ainsi que le flux d'excitation (ϕ_f) comme grandeurs de commande, les courants statorique (I_d, I_q) comme variables d'état et le couple C_r comme perturbation. A partir des équations (6.14), on peut écrire le système d'équations comme suit :

$$\begin{bmatrix} \frac{dI_d}{dt} \\ \frac{dI_q}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R_s}{L_d} & p\Omega \frac{L_q}{L_d} \\ -p\Omega \frac{L_d}{L_q} & -\frac{R_s}{L_d} \end{bmatrix} \begin{bmatrix} I_d \\ I_q \end{bmatrix} + \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \end{bmatrix} \begin{bmatrix} V_d \\ V_q \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{p\Omega}{L_q} \end{bmatrix} \phi_f \quad (6.16)$$

$$J \frac{d\Omega}{dt} = C_{em} - C_r - f\Omega$$

Où :

$$C_{em} = \frac{3}{2} p [(L_d - L_q) I_d I_q + \phi_f I_q] \quad (6.17)$$

La MSAP peut être représentée par le schéma bloc illustré par la figure (6.2) :

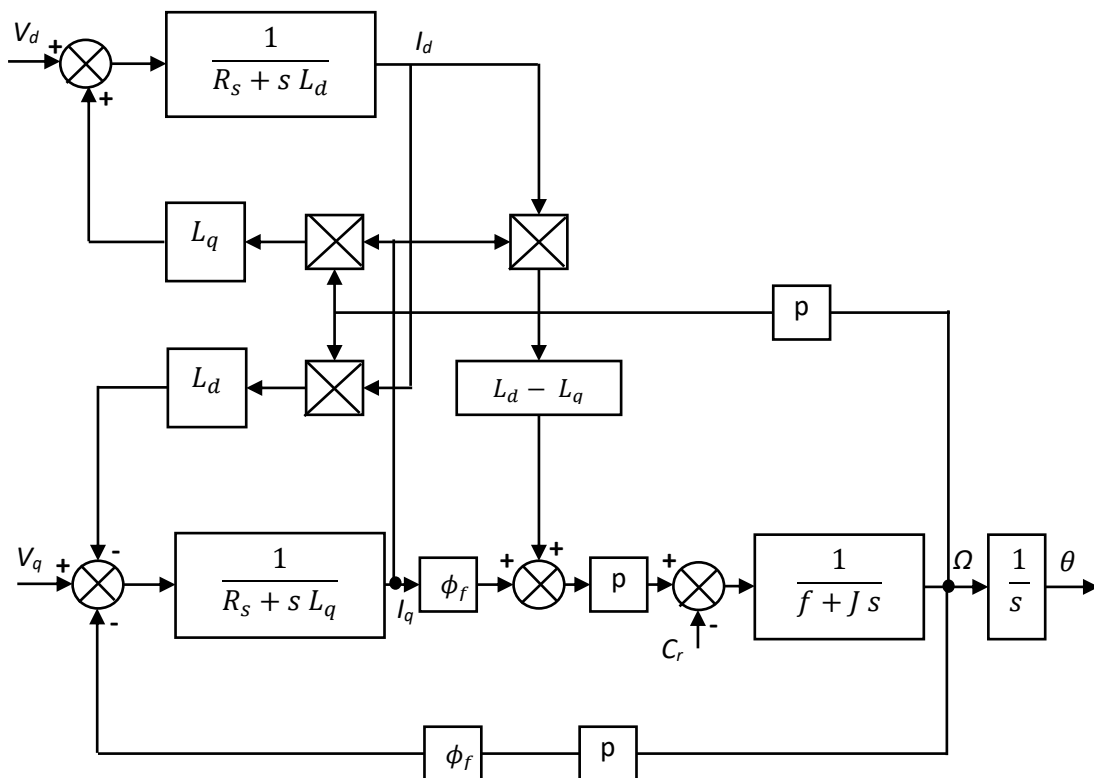


Figure 6.2. Schéma block de MSAP

6.2.2.6. Autopilotage

Afin de s'assurer un bon fonctionnement des MSAP, les courants de phase et la rotation du rotor doivent évoluer en synchronisme. Les convertisseurs doivent être commandés alors à l'aide d'un capteur de position (autopilotage)

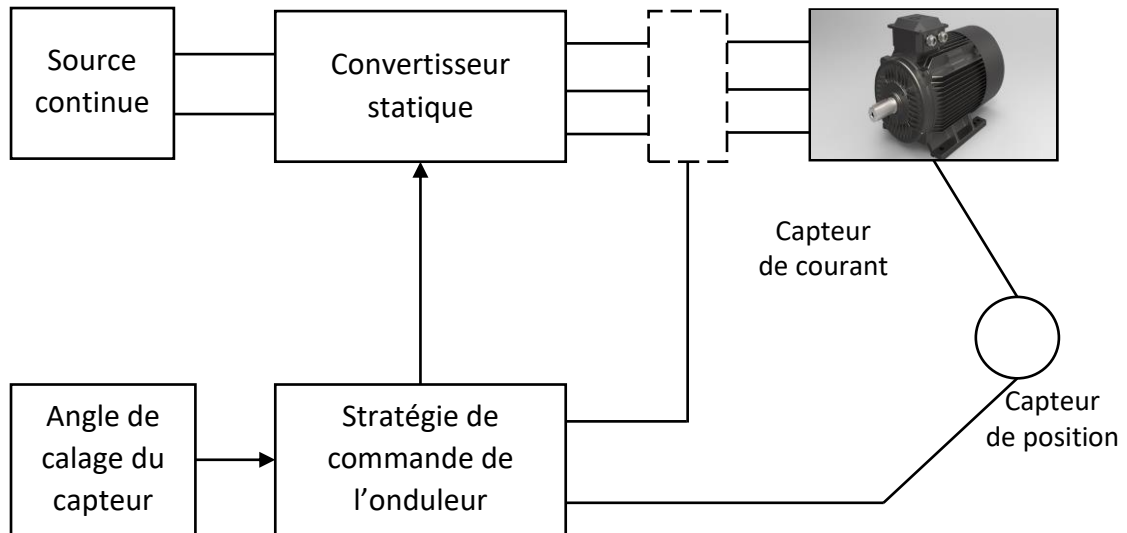


Figure 6.3. Schéma de principe d'autopilotage

L'autopilotage d'une machine synchrone consiste à maintenir constant ou peu variable le décalage angulaire entre les Fem de celle-ci et les courants statoriques. Avec cette condition le couple électromagnétique développé par la machine peut être contrôlé et une boucle d'asservissement de position ou de vitesse peut être réalisée autour de la boucle de commande du couple de la machine. Pour réaliser cette tâche, le synchronisme de la machine doit être contrôlé par un capteur de position lié au rotor. Cela permet d'imposer la fréquence du courant ou de la tension.

Avec l'utilisation de ce principe, plusieurs variantes existent, dans lesquelles le type de la machine et du convertisseur est pris en compte. [131] [132]

6.2.2.7. Principe de la commande vectorielle

Le principe de la commande vectorielle des machines à courant alternatif est de créer un modèle identique à celui d'une machine à courant continu à excitation séparée. Il faut cependant se placer dans un repère particulier où le couple électromagnétique s'exprime simplement en fonction des composantes des courants suivant les deux axes (axe d et axe q).

Habituellement, la composante d'axe d du courant statorique joue le rôle de l'excitation et permet de régler la valeur du flux dans la machine. La composante d'axe q joue le rôle du courant d'induit et permet de contrôler le couple. [125] [126] [133]

L'application de la commande vectorielle aux machines synchrones à aimants permanents consiste à maintenir le courant I_d nul [133].

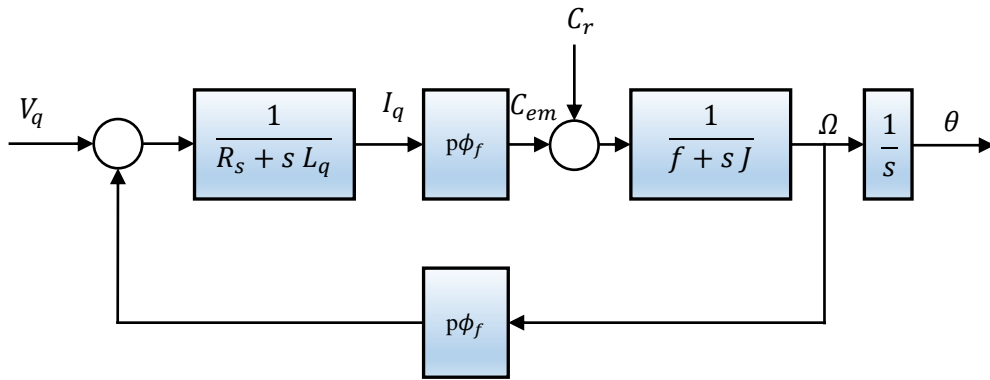


Figure 6.4. Modèle de la MSAP après découplage

Après avoir présenté et modélisé le moteur synchrone à aimants permanents, nous allons l'introduire dans notre système de façon à rajouter les parties électriques qui génèrent les couples qui vont être injectés dans la partie mécanique commune (Robot manipulateur + actionneur).

6.3. Modèle du système global

Dans l'approche proposée, chaque articulation du manipulateur est entraînée par un moteur synchrone à aimant permanent dans le système de commande, comme il est représenté dans la figure 6.5.

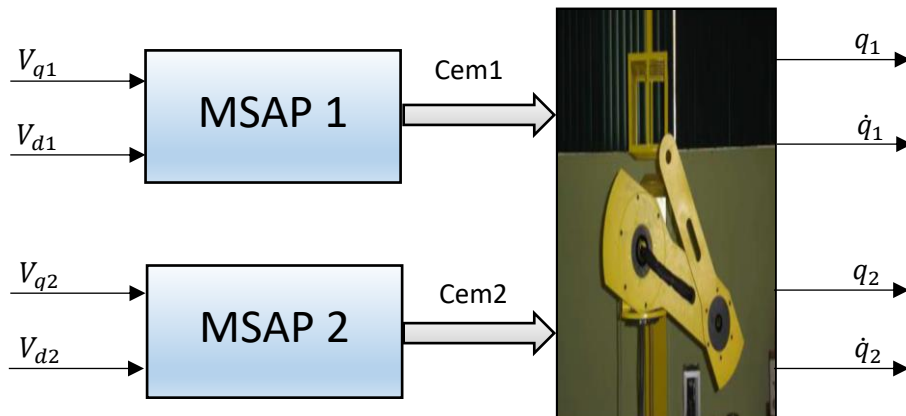


Figure 6.5. Schéma synoptique du modèle global (Robot manipulateur + MSAP)

Le couple inséré sur l'articulation pour entraîner le manipulateur est le couple de charge du moteur, qui est considéré dans une équation dynamique formée comme :

$$C_{em} = J \frac{d\Omega}{dt} + f\Omega + C_r \tag{6.18}$$

Sachant que :

$$\Omega = \frac{d\theta}{dt} \text{ est la vitesse angulaire du moteur.}$$

L'équation (6.17) peut être écrite sous forme :

$$C_{em} = J \frac{d\Omega}{dt} + f\Omega + C_r \Rightarrow C_{em} = J\ddot{\theta} + f\dot{\theta} + C_r \tag{6.19}$$

C'est à partir de l'équation (6.17) que le couple produit est composé de deux mécanismes distincts. Le premier terme correspond au «couple de réaction mutuel» intervenant entre I_q et l'aimant permanent. Alors que le second terme correspond au «couple de réluctance» en raison des différences de réluctance sur l'axe d et sur l'axe q ou (inductance) pour un fonctionnement à flux constant lorsque I_d est égal à zéro, le couple électrique C_{em} est aussi défini par :

$$C_{em} = \frac{3}{2} p \phi_f I_q = K_i I_q \quad (6.20)$$

Où K_i est la constante de couple du moteur. Cette équation de couple ressemble à celle du moteur à courant continu ordinaire et offre donc une facilité de contrôle.[134]

C_{em} et C_r représentent respectivement le couple moteur généré et le couple de charge. Dans le but d'augmenter la vitesse de déplacement du robot manipulateur, les moteurs sont équipés d'engrenages à réduction (réducteur rotatif-rotatif). Le réducteur relie la position du moteur à la position de l'articulation comme suit :

$$\theta = \eta q \text{ et } \tau = \eta C_r \quad (6.21)$$

Où

η est la matrice diagonale du rapport de réduction. Dans ce qui suit, une contrainte pratique est considérée. L'équation dynamique d'un robot manipulateur à 2ddl se présente sous la forme standard et est réécrite comme suit:

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + \tau_d \quad (6.22)$$

Où τ_d représente la perturbation externe.

Il convient de noter que l'entrée de commande appliquée au robot manipulateur est la tension des moteurs. Ainsi, en utilisant les équations (6.16) - (6.22) l'équation suivante est obtenue.

$$\begin{cases} \dot{\xi}_1 = \xi_2 \\ \dot{\xi}_2 = \left(\frac{M}{\eta} + \eta J\right)^{-1} \left[-\left(\frac{C}{\eta} + \eta f\right) \xi_2 + K_i \xi_3 - \frac{g}{\eta} - \frac{\tau_d}{\eta} \right] \\ \dot{\xi}_3 = -\frac{p \phi_f \eta}{L_q} \xi_2 - \frac{R_s}{L_q} \xi_3 + \frac{1}{L_q} V_q \end{cases} \quad (6.23)$$

Avec :

$$\xi_1 = q = [q_1 \ q_2]^T, \quad \xi_2 = \dot{q} = [\dot{q}_1 \ \dot{q}_2]^T \text{ et } \xi_3 = I_q = [I_{q1} \ I_{q2}]^T$$

Sachant que :

$$L_q = \begin{bmatrix} L_{q1} & 0 \\ 0 & L_{q2} \end{bmatrix}, \quad \eta = \begin{bmatrix} \eta_1 & 0 \\ 0 & \eta_2 \end{bmatrix}, \quad f = \begin{bmatrix} f_1 & 0 \\ 0 & f_2 \end{bmatrix}, \quad \tau_d = \begin{bmatrix} \tau_{d1} & 0 \\ 0 & \tau_{d2} \end{bmatrix}, \quad p = \begin{bmatrix} p_1 & 0 \\ 0 & p_2 \end{bmatrix}, \quad R_s = \begin{bmatrix} R_{s1} & 0 \\ 0 & R_{s2} \end{bmatrix} \text{ et } \phi_f = \begin{bmatrix} \phi_{f1} & 0 \\ 0 & \phi_{f2} \end{bmatrix}.$$

6.4. Application de la commande non linéaire par modes glissants au modèle global

Après avoir introduit le modèle de l'actionneur dans le modèle global, nous allons maintenant lui appliquer la commande par modes glissants pour le contrôle de la position et la vitesse du robot manipulateur.

La commande du modèle global se fait maintenant en agissant sur les nouvelles grandeurs de commandes qui sont les tensions de chaque actionneur électrique pour réduire le nombre de grandeurs de commandes et donc simplifié la synthèse du régulateur, nous allons appliquer la commande vectorielle par régulateur PI afin de maintenir le courant statorique dans le référentiel d,q selon un seul axe. Ce qui conduit à lier l'entrée V_d des deux moteurs à la sortie du PI assurant la commande vectorielle.

La surface de glissement dans ce cas est donnée par la relation suivante :

$$S = \left(\frac{d}{dt} + \lambda \right)^{r-1} e(t) \quad (6.24)$$

Sachant que $e(t) = \xi_1 - \xi_{1d}$ est l'erreur de poursuite. Et λ est une matrice diagonale définie positive.

Le degré relatif r dans ce cas est égale à 3, nous aurons donc :

$$S = \ddot{e}(t) + 2\lambda\dot{e}(t) + \lambda^2 e(t) \quad (6.25)$$

La loi de commande par mode glissant qui contraint les erreurs à converger vers zéro en un temps fini est exprimée par la relation suivante :

$$V_q = V_{qd} + V_{qeq} \quad (6.26)$$

Sachant que : $V_{qd} = \begin{bmatrix} V_{qd1} \\ V_{qd2} \end{bmatrix}$ et $V_{qeq} = \begin{bmatrix} V_{qeq1} \\ V_{qeq2} \end{bmatrix}$.

Avec V_{qd} est la commande discontinue appelée aussi commande robuste et V_{qeq} est la commande équivalente.

Nous allons maintenant déterminer la loi de commande par mode glissant globale V_q .

6.4.1. La loi de commande discontinue

La loi de commande discontinue est conçue suivant l'équation (6.25), nous aurons alors :

$$\begin{pmatrix} V_{qd1} \\ V_{qd2} \end{pmatrix} = \begin{pmatrix} -k_1 \text{sign}(S_1) \\ -k_2 \text{sign}(S_2) \end{pmatrix} \quad (6.27)$$

Où k_1 et k_2 sont des constantes positives.

Par substitution de l'équation (6.25) dans (6.27), nous aurons :

$$\begin{pmatrix} V_{qd1} \\ V_{qd2} \end{pmatrix} = \begin{pmatrix} -k_1 \text{sign}(\ddot{e}_1(t) + 2\lambda\dot{e}_1(t) + \lambda^2 e_1(t)) \\ -k_2 \text{sign}(\ddot{e}_2(t) + 2\lambda\dot{e}_2(t) + \lambda^2 e_2(t)) \end{pmatrix} \quad (6.28)$$

6.4.2. La loi de commande équivalente

La loi de commande équivalente est extraite suivant $\dot{S} = 0$. En utilisant la surface considérée, nous obtenons :

$$\dot{S} = \ddot{e}(t) + 2\lambda\dot{e}(t) + \lambda^2 e(t) \quad (6.29)$$

Donc nous aurons :

$$\dot{S} = \ddot{\xi}_1 - \ddot{\xi}_{1d} + 2\lambda(\dot{\xi}_1 - \dot{\xi}_{1d}) + \lambda^2(\xi_1 - \xi_{1d}) \quad (6.30)$$

Par substitution de (6.23) dans (6.30), nous obtenons :

$$V_{qeq} = \left(\frac{L_q}{K_i \rho_1} \right) \left[(\dot{\rho}_1 \rho_2 + \rho_1 \dot{\rho}_2 + 2\lambda \rho_1 \rho_2 + K_i \rho_1 \frac{p\phi_f}{L_q} \eta - \rho_1^2 \rho_2^2 - \lambda^2) \xi_2 + \left(K_i \dot{\rho}_1 + 2\lambda K_i \rho_1 - K_i \rho_1 \frac{R_s}{L_q} - \rho_1^2 \rho_2 K_i \right) \xi_3 + \frac{1}{\eta} (\rho_1 + \dot{\rho}_1 - \rho_1^2 \rho_2) g + \frac{1}{\eta} (\rho_1 + \dot{\rho}_1 - \rho_1^2 \rho_2) \tau_d + \frac{1}{\eta} \rho_1 \dot{g} + \frac{1}{\eta} \rho_1 \dot{\tau}_d + \ddot{\xi}_{1d} + 2\lambda \dot{\xi}_{1d} + \lambda^2 \xi_{1d} \right] \quad (6.31)$$

$$\text{Avec : } \rho_1 = \left(\frac{M}{\eta} + \eta J \right)^{-1} \text{ et } \rho_2 = \left(\frac{C}{\eta} + \eta f \right)$$

L'expression (6.31) est réduite vu l'utilisation de l'écriture matricielle, cependant, l'utilisation de la forme développée conduit à une expression gigantesque de la commande.

6.4.3. Résultats de simulation

Dans cette partie, nous allons présenter les résultats de simulation afin d'apprécier les performances obtenues par le contrôleur introduit ci-dessus. Nous utilisons les valeurs numériques du robot manipulateur rapportées par la section B.2 dans l'ANNEXE B. À des fins de simulation, nous supposons que ce robot est équipé d'un MSAP à entraînement direct à chaque articulation.

Les paramètres du MSAP sont indiqués dans le tableau ci-dessous [15] [135]:

Paramètres	Valeur	
	MSAP 1	MSAP 2
Moment d'inertie (J) kg m ²	0.0125	0.0025
Résistance (R _s) ohm	1	1.9
Inductance (l _q) mH	0.01	0.00672
Inductance (l _d) mH	0.01	0.00636
Coefficient de frottement visqueux (f) N.m.s	0.006	0.203
Flux (φ _f) v.s	0.08627	0.0106
Constante du couple (K _i)	8.5683	1.8695

Tableau 6.1. Paramètres des MSAP

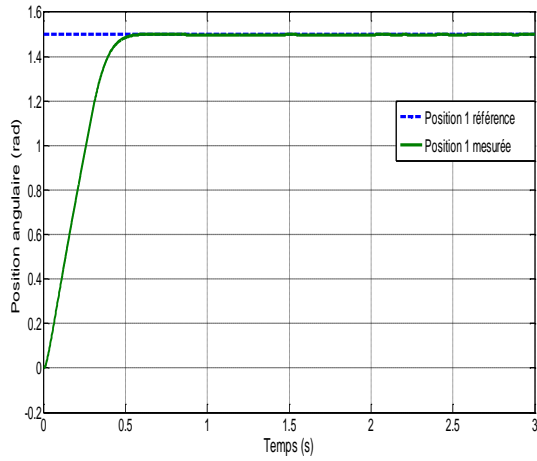
Les paramètres les plus importants qui affectent les performances de contrôle des robots manipulateurs sont la perturbation externe et la variation paramétrique de la masse du 2^{ème} segment m₂. Dans les simulations numériques, deux circonstances, dont la situation nominale (m₂ = 2.0458 kg et τ_d=0) au départ, la variation des paramètres va être introduite à l'instant 1.5 s, ensuite une perturbation à l'instant 2s, est considérée. La variation des paramètres est considéré par l'ajout d'un poids de 1.5 kg à la masse m₂, c'est-à-dire, m₂ = 3.5458 kg. La perturbation externe est exprimée par des forces externes qui sont appliquées sur le bras manipulateur, leurs formes s'expriment par τ_d = [5 sin(10t) 2 sin(10t)].

Les mêmes trajectoires du chapitre précédent sont considérées dans ce chapitre.

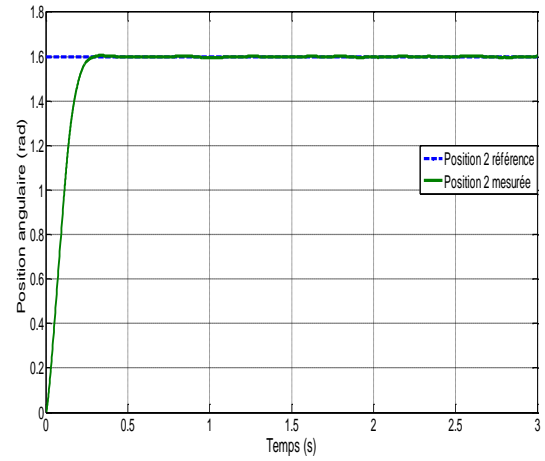
Paramètres de la commande

$$\lambda_1 = 30, \lambda_2 = 50, k_1 = 25, k_2 = 35.$$

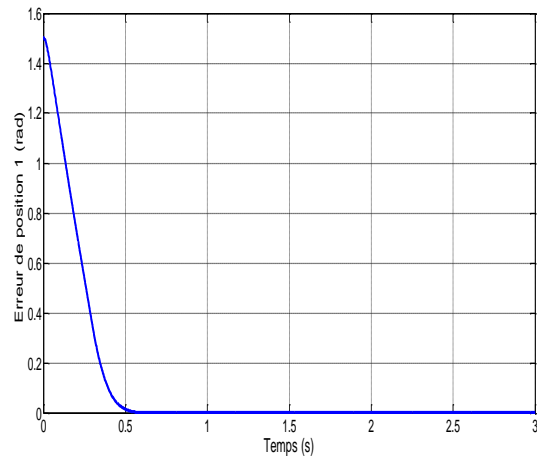
Les résultats de simulation obtenus sont montrés sur les figures (6.6), (6.7) et (6.8).



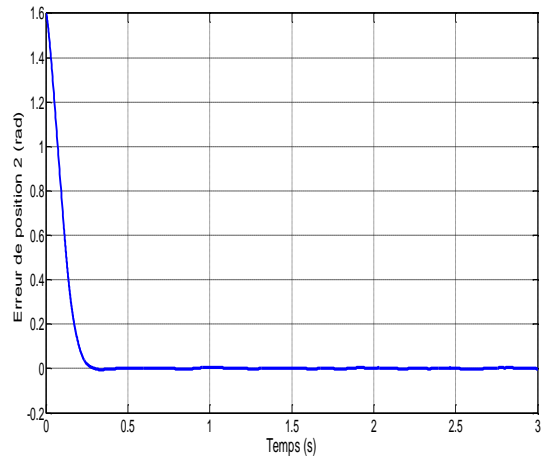
(A)



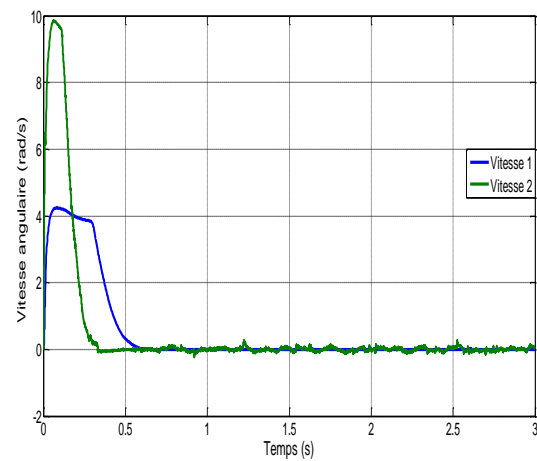
(B)



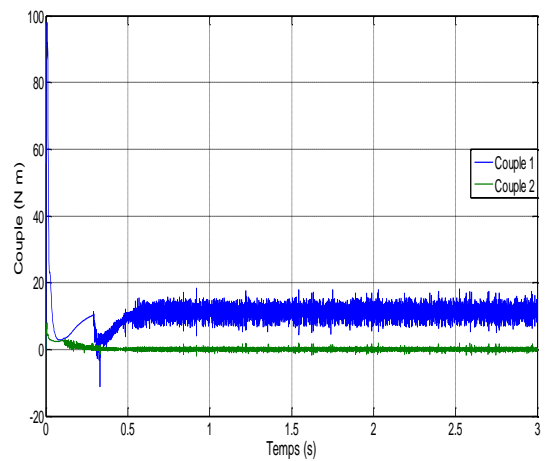
(C)



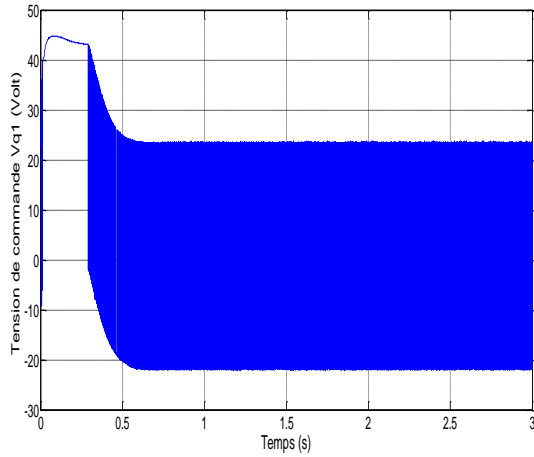
(D)



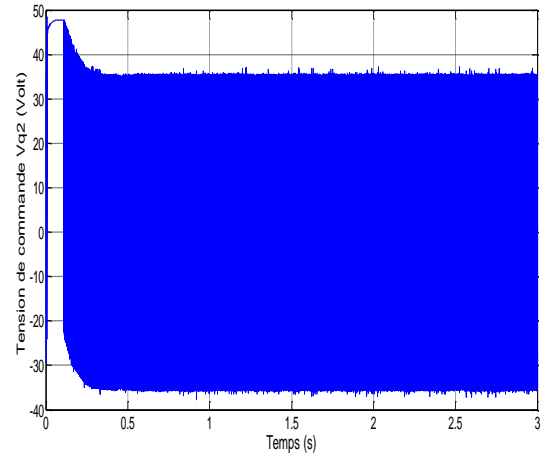
(E)



(F)

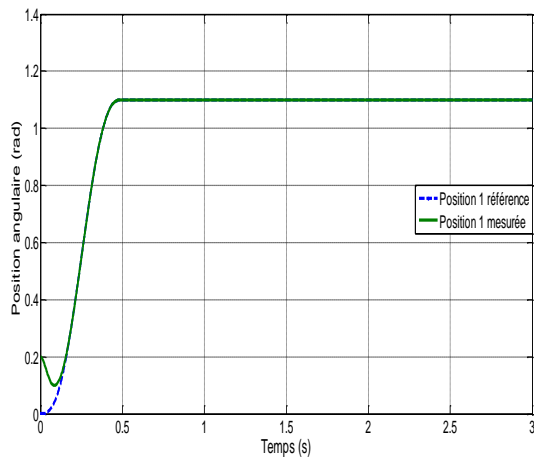


(G)

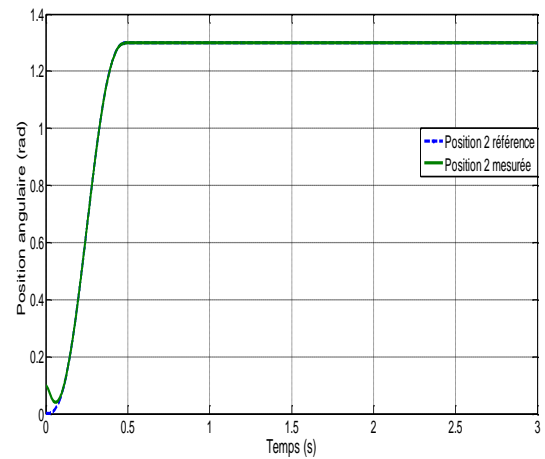


(H)

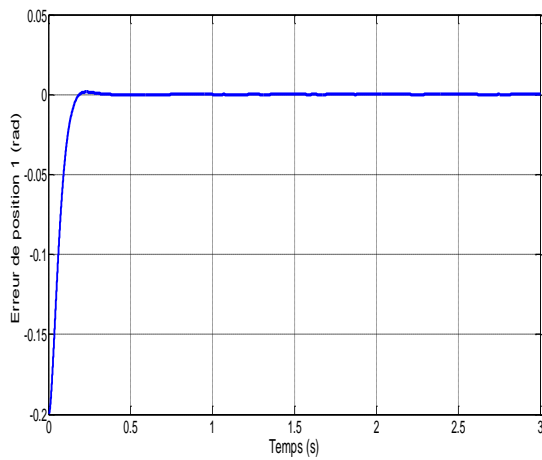
Figure 6.6. Commande par Modes glissants du modèle global avec consigne constante : (A) et (B) positions angulaires, (C) et (D) erreurs des positions, (E) vitesses angulaires et (F) les couples moteur, (G) et (H) tensions de commande.



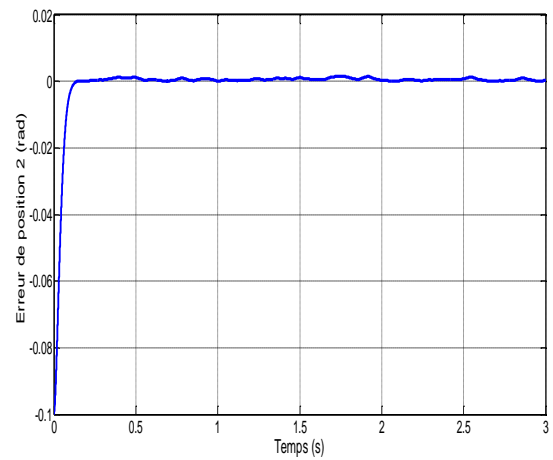
(A)



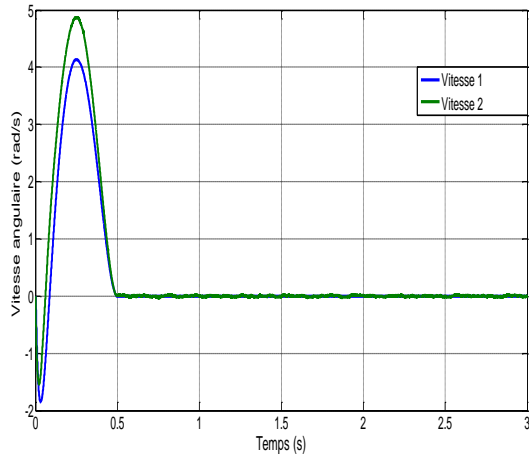
(B)



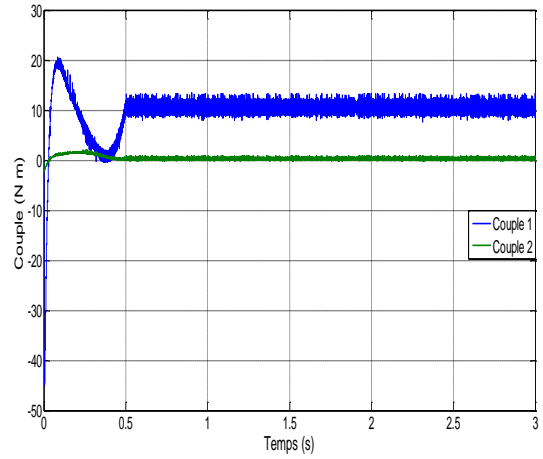
(C)



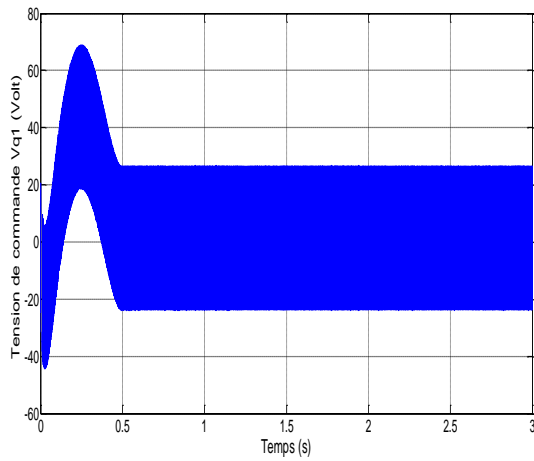
(D)



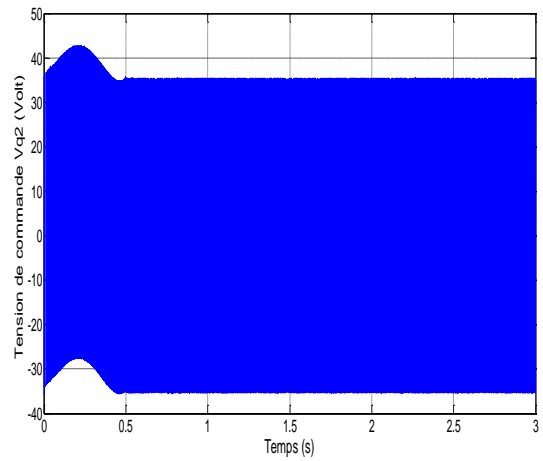
(E)



(F)

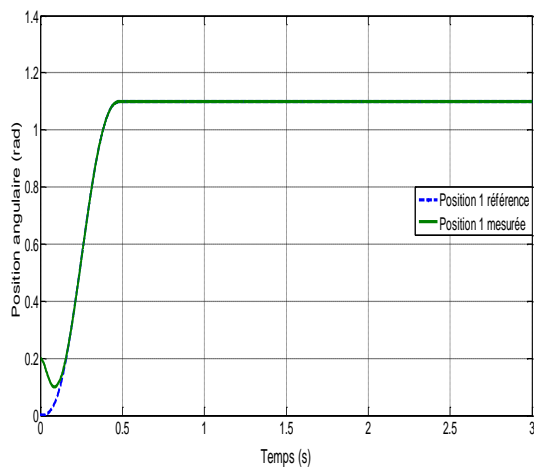


(G)

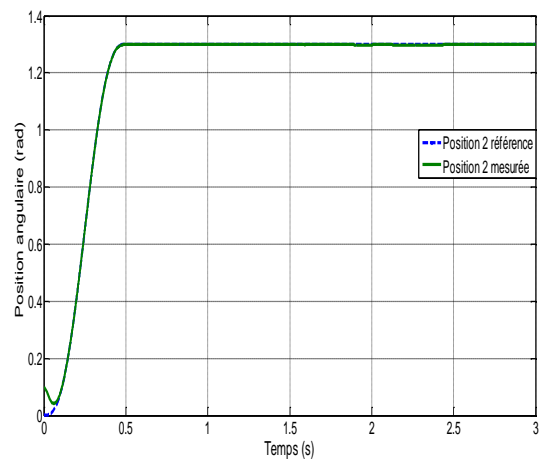


(H)

Figure 6.7. Commande par Modes glissants en poursuite de trajectoire avec conditions initiales $[0.2, 0.1]$: (A) et (B) positions angulaires, (C) et (D) erreurs des positions, (E) vitesses angulaires et (F) les couples moteur, (G) et (H) tensions de commande.



(A)



(B)

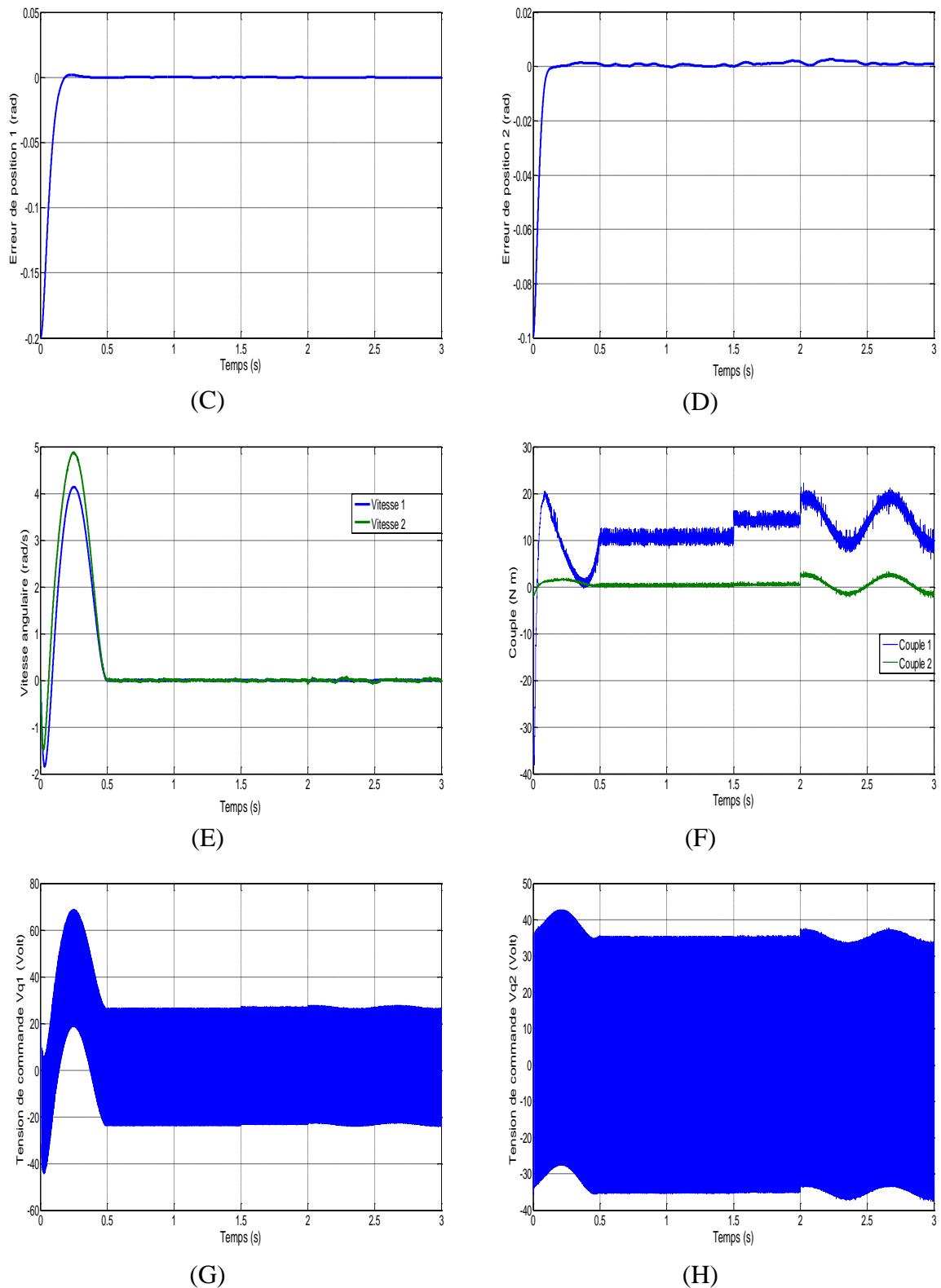


Figure 6.8. Commande par Modes glissants en poursuite de trajectoire avec variation paramétrique et perturbation externe, les conditions initiales sont $[0.2, 0.1]$: (A) et (B) positions angulaires, (C) et (D) erreurs des positions, (E) vitesses angulaires et (F) les couples moteur, (G) et (H) tensions de commande.

Synthèse 1

A travers les figures (6.6.A) et (6.6.B) nous remarquons que les deux positions articulaires 1 et 2 rejoignent leurs positions désirées avec un temps de réponse de 0.47s et 0.32s respectivement. Dans les (6.6.C) et (6.6.D), nous pouvons voir aussi que les erreurs de position convergent vers zéro avec des oscillatoires dûs au phénomène de chattering qui apparaissent sur la deuxième erreur. Tels qu'il est montré sur la figure (6.6.E) les vitesses articulaires atteignent des valeurs maximales lors du démarrage puis elles diminuent vers des valeurs nulles à l'instant où les positions articulaires atteignent leurs positions désirées. Nous réalisons également sur la figure (6.6.F) que le couple appliqué à la première articulation a une valeur maximale d'environ 90 Nm. Cela nécessite qu'un courant électrique maximal à travers les phases q du moteur soit d'environ 5A. Il faut noter aussi qu'il y a présence des oscillations dûes au phénomène de chattering. De plus, les couples générés restent dans des plages admissibles rapportées dans [124], c'est-à-dire $\pm 15\text{Nm}$ et $\pm 4\text{Nm}$ pour les articulations 1 et 2, respectivement. Bien que l'on observe sur les figures (6.6.G) et (6.6.H) que de grandes valeurs de crête apparaissent sur les deux signaux de tension au début, La raison en est que les valeurs des tensions élevées sont calculées par le contrôleur lorsque les erreurs des positions sont grandes. Nous constatons aussi qu'il y a des commutations à une fréquence très élevée, ceci est dû au phénomène de chattering.

En se référant aux figures (6.7.A) et (6.7.B), nous remarquons un bon suivi de trajectoire de référence des deux positions articulaires, malgré le changement des conditions initiales, 0.2 rad pour la première articulation et 0.1rad pour la deuxième. Nous pouvons confirmer ces résultats à travers des figures (6.7.C) et (6.7.D) représentant les erreurs de poursuite. Nous avons aussi un bon suivi des vitesses articulaires tel qu'il montré sur la figure (6.7.E) avec une phase de freinage au démarrage due au fait que les conditions initiales des deux articulations sont différentes de celles des trajectoires désirées, ensuite, les deux vitesses articulaires ont une allure croissante jusqu'à une valeur maximale, puis elles décroissent jusqu'à zéro.

Le couple appliqué à la première articulation à une valeur maximale d'environ -38 Nm tels qu'il est montré sur la figure (6.7.F) ceci est expliqué par le fait que les conditions initiales sont différentes de zéro et que l'articulation cherche à atteindre la position de référence. Des oscillations dûes au phénomène de chattering sont visibles sur les couples. Les couples générés restent toujours dans les plages admissibles. Nous observons sur les figures (6.7.G) et (6.7.H) que les deux signaux de tension commutent à une fréquence très élevée.

Afin de mettre en évidence la robustesse de cette loi de commande, les figures (6.8) illustrent les courbes de poursuite de position en présence de variation paramétrique et d'une perturbation externe. Nous constatons à partir des figures (6.8.A) et (6.7.B) que nous avons toujours une bonne poursuite de trajectoire. A travers la figure (6.8.D) nous remarquons une petite influence de l'introduction d'une masse supplémentaire sur m_2 à l'instant 1.5s et aussi d'une perturbation externe à l'instant 2s. Cette influence est nettement visible sur la figure (6.8.F) où nous remarquons la réaction des couples appliqués.

6.5. La commande par modes glissants d'ordre supérieur

La théorie de la commande par modes glissants d'ordre supérieur MGOS [57] a été introduite comme une alternative au problème des modes glissants classiques, tout en préservant les principaux avantages de la précédente approche, telles que la robustesse et la convergence en temps fini. Elle permet aussi d'améliorer la précision asymptotique.

Les premiers concepts de cette approche ont été introduits au milieu des années 80 par Levantovsky [136] et Emelyanov [137]. Celle-ci est caractérisé par le fait, que le terme discontinu n'apparaît plus directement dans la commande synthétisée mais dans une de ses dérivées supérieures ce qui a le mérite de réduire le chattering.

6.5.1. Concepts de base des MGOS

Nous allons considérer le système non linéaire suivant :

$$\xi^{(n)} = f(\xi, t) + g(\xi, t)u \quad (6.32)$$

$$s = s(\xi, t)$$

Où $\xi = [\xi_1, \dots, \xi_n]^T \in X \subset R^n$ qui représente l'état du système. La commande $u \in X \subset R$ est une fonction discontinue et bornée qui dépend de l'état et du temps. f et g sont des champs de vecteurs suffisamment différentiables mais connus de façon incertaine.

s est une variable de glissement choisie afin d'assurer la convergence en temps fini vers l'ensemble de glissement d'ordre n .

L'ensemble de glissement d'ordre n par rapport à $s(\xi, t)$ est défini par

$$s_n = \{\xi \in X : s = \dot{s} = \dots = s^{(n-1)} = 0\} \quad (6.33)$$

Par abus de langage, cet ensemble est souvent appelé surface de glissement d'ordre n .

Si le système est de degré relatif $n > 1$ par rapport à la variable de glissement, une commande en mode glissant d'ordre n permettra d'obtenir une convergence en temps fini vers la surface, en forçant les trajectoires d'état du système à être confinées dans l'ensemble de glissement comme le montre la figure suivante[138]:

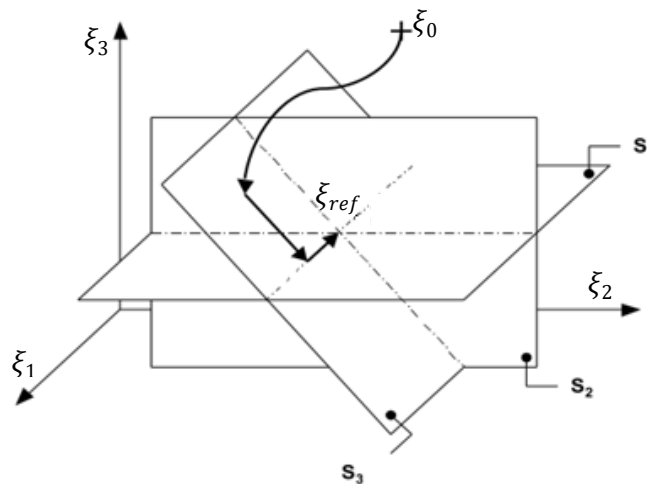


Figure 6.9. Un espace d'état de dimension 3 : l'intersection des 3 surfaces de commutation définit le point de référence ξ_{ref}

6.5.2. Commande par mode glissant d'ordre deux

L'objectif de la commande par mode glissant d'ordre deux est de forcer la variable de glissement et sa première dérivée à zéro. Dans le cas où le degré relatif du système par rapport à la variable de glissement est 1 ou 2, s et \dot{s} convergent vers zéro en temps fini.

$$S_2 = \{\xi : s = \dot{s} = 0\} \quad (6.34)$$

Ceci peut être réalisé par une commande agissant sur la dérivée seconde de la variable de glissement qui, de manière générale, peut s'écrire sous la forme :

$$\ddot{S} = \Psi(\xi, t) + \Phi(\xi, t).v \quad (6.35)$$

Deux cas sont possibles, en fonction du degré relatif du système par rapport à la variable de glissement [138]:

- $v = \dot{u}$ dans le cas où le système (6.32) est de degré relatif $n=1$ par rapport à s , c.à.d.

$$\frac{\partial \dot{s}}{\partial u} \neq 0.$$

- $v = u$ dans le cas où le système (6.32) est de degré relatif $n=2$ par rapport à s , c.à.d.

$$\frac{\partial \ddot{s}}{\partial u} \neq 0.$$

Afin de pouvoir réaliser des algorithmes par modes glissants d'ordre deux, les hypothèses de travail suivantes doivent être vérifiées afin de valider l'atteignabilité de la surface de glissement et la bornitude de la variable \dot{s} [57].

- Les fonctions incertaines $\Psi(\xi, t)$ et $\Phi(\xi, t)$ sont bornées.
- il existe quatre constantes positives S_0 , C_0 , K_m et K_M telles que dans un voisinage $|s(\xi, t)| < S_0$, les inégalités suivantes soient vérifiées :

$$|\Psi(\xi, t)| < C_0 \quad \text{et} \quad 0 < K_m \leq \Phi(\xi, t) \leq K_M \quad (6.36)$$

Ces hypothèses impliquent que la dérivée seconde de la fonction de commutation est uniformément bornée dans un certain domaine (E_I) pour l'entrée considérée.

La Figure 6.10 fait apparaître la trajectoire de convergence du système vers la surface S_2

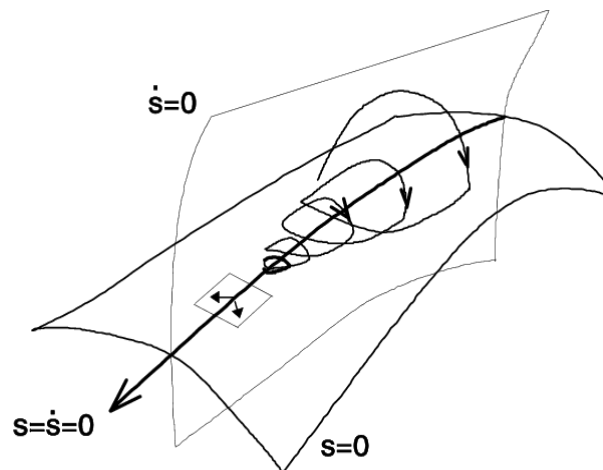


Figure 6.10. Trajectoire du mode glissant d'ordre 2

Plusieurs algorithmes des modes glissants d'ordre deux ont été proposées dans la littérature [44], parmi ces algorithmes on peut citer :

- Algorithme du Twisting.
- Algorithme du Super-Twisting.

- Algorithme du prescribed convergence law (Algorithme avec une loi de convergence prédéfinie).

6.5.2.1. Algorithme du Twisting

Cet algorithme est connu historiquement pour être le premier contrôleur par mode glissant d'ordre supérieur. Son principe de fonctionnement repose sur la commutation de l'amplitude de la commande entre deux valeurs en fonction du quadrant dans lequel se trouve l'état du système [139] [140]. Cet algorithme *twiste* autour de l'origine du plan de glissement d'ordre deux, et est nommé Twisting à cause de sa convergence en temps fini vers l'origine du plan de phase (s, \dot{s}) , avec un nombre infini de rotation, tel qu'il est montré sur la figure 6.11 [141].

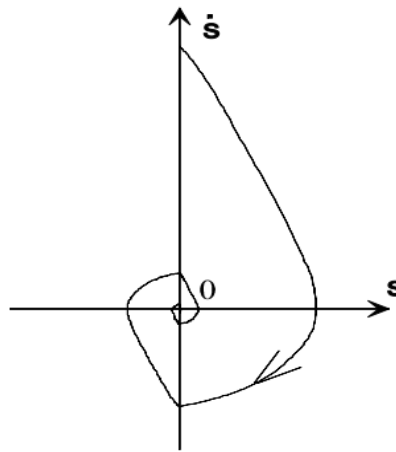


Figure 6.11. Convergence de l'algorithme Twisting dans le plan (s, \dot{s})

L'expression de cet algorithme pour un système de degré relatif 2 est donnée par [139]:

$$u = -r_1 \text{sign}(s) - r_2 \text{sign}(\dot{s}) \quad \text{avec } r_1 > r_2 > 0 \quad (6.37)$$

La trajectoire du système différentiel converge au point d'équilibre $s = \dot{s} = 0$ en un temps fini sous les conditions suivantes :

$$\begin{cases} (r_1 + r_2)K_m - C_0 > (r_1 - r_2)K_m + C_0 \\ (r_1 - r_2)K_m > C_0 \end{cases} \quad (6.38)$$

6.5.2.2. Algorithme du Super-Twisting

Cette loi de commande a été développée au début des années 90 par Emylanov, ensuite par Levant [139], pour l'asservissement des systèmes à degré relatif égal à 1 par rapport à la surface de glissement. Cet algorithme est considéré comme étant l'algorithme le plus utilisé parmi les autres algorithmes de modes glissants d'ordre deux. Le plus grand intérêt de cette technique réside dans sa capacité à réduire le chattering, grâce à la continuité du signal de commande.

Le Super-Twisting est composé de deux parties, La première est définie par sa dérivée par rapport au temps u_1 , tandis que la deuxième est donnée par la fonction continue de la variable de glissement u_2 [44]:

$$u(t) = u_1(t) + u_2(t) \quad (6.39)$$

$$\dot{u}_1 = \begin{cases} -u & \text{si } |u| > 1 \\ -W \operatorname{sign}(s) & \text{si } |u| \leq 1 \end{cases} \quad (6.40)$$

$$u_2 = \begin{cases} -\gamma |s_0|^\rho \operatorname{sign}(s) & \text{si } |s| > s_0 \\ -\gamma |s|^\rho \operatorname{sign}(s) & \text{si } |s| \leq s_0 \end{cases} \quad (6.41)$$

Avec les conditions suffisantes de convergence en temps fini suivantes [44] :

$$\begin{aligned} W &> \frac{c_0}{K_m} \\ \gamma^2 &\geq \frac{4 c_0 K_M (W + c_0)}{K_m^2 K_m (W - c_0)} \\ 0 &< \rho \leq 0.5 \end{aligned} \quad (6.42)$$

Cette loi de commande peut être simplifiée si le système de contrôle est linéairement dépendant de la commande, dans ce cas u n'a pas besoin d'être borné et $s_0 = \infty$, ainsi la loi de commande est donnée par [141] [139]:

$$\begin{cases} u = -\gamma |s|^{1/2} \operatorname{sign}(s) + u_1 \\ \dot{u}_1 = -W \operatorname{sign}(s) \end{cases} \quad (6.43)$$

L'algorithme du Super-Twisting n'a pas besoin d'information sur la dérivée de s . L'intérêt de ce dernier est qu'il très robuste. Généralement, ρ est fixé à 0.5.

6.5.2.3. Algorithme prescribed convergence law (PCL)

Cette classe de contrôleurs par modes glissants offre la possibilité de choisir une trajectoire transitoire du processus : la commutation de u dépend d'une fonction appropriée de s . La formulation générale d'une telle classe d'algorithmes de commande par modes glissants est la suivante [44]:

$$u(t) = -\alpha \operatorname{sign}(\dot{s} - \varphi(s)) \quad (6.44)$$

Où α est une constante positive et φ une fonction continue lisse (figure 6.12) définie par :

$$\varphi(s) = \beta |s|^{\frac{1}{2}} \operatorname{sign}(s) \quad (6.45)$$

Avec β est une constante positive.

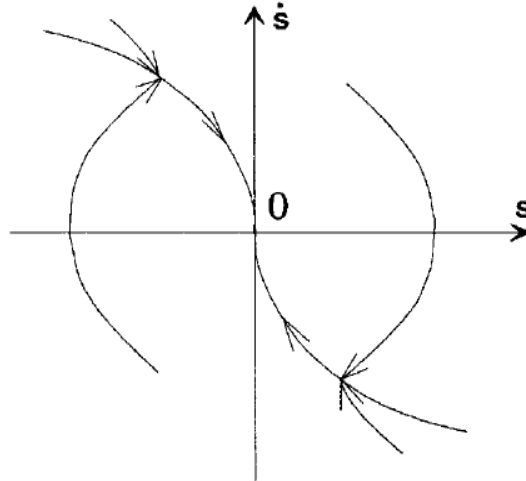


Figure 6.12. Convergence de l'algorithme PCL dans le plan (s, \dot{s})

Une condition suffisante pour une convergence en temps fini vers la surface de glissement est définie par la relation suivante :

$$\alpha K_m - C_0 > \frac{\beta^2}{2} \quad (6.46)$$

Où K_m et C_0 sont des constantes positives.

6.5.2. Commande par mode glissant d'ordre supérieur

La commande mode glissant d'ordre supérieur représente une extension du mode glissant d'ordre 1 à un degré plus élevé. Cette généralisation conserve la principale caractéristique de robustesse que celle du mode glissant classique, permettant ainsi de réduire leur principal inconvénient : l'effet de chattering au voisinage de la surface de glissement.

L'extension des modes glissants d'ordre un aux modes glissants d'ordre supérieur est caractérisée par le choix d'une commande discontinue agissant non pas sur la surface de glissement mais sur ses dérivées supérieures[140].

Il est nécessaire de connaître les variables d'états ainsi que leurs dérivées dans la commande par modes glissants d'ordre supérieur. Il en résulte que les fonctions $(s, \dot{s}, \ddot{s}, \dots, s^{(r-1)})$ sont nécessaires pour synthétiser une commande par modes glissants d'ordre r .

Considérons le système non linéaire suivant :

$$\begin{aligned} \dot{\xi} &= f(\xi, t) + g(\xi, t)u \\ s &= s(\xi, t) \end{aligned} \quad (6.47)$$

Avec $\xi \in R^n$ représentant l'état du système, $u \in R$ est l'entrée du système. f et g sont des fonctions suffisamment différentiables.

L'idée de base d'une commande par modes glissants d'ordre supérieur est que la fonction discontinue agit sur un ordre supérieur (dérivées $s^{(i)}$ ($i = 1, 2, \dots, r - 1$)) de la surface de glissement :

$$s(\xi, t) = \dot{s}(\xi, t) = \ddot{s}(\xi, t) = \dots = s^{(r-1)}(\xi, t) = 0 \quad (6.48)$$

Supposons que le degré relatif du système r est connu et qu'il est constant. Cela signifie que la commande u n'apparaît que dans la $r^{\text{ième}}$ dérivée de s et que $\frac{\partial s^{(r)}}{\partial u} \neq 0$ en ce point. Il est possible de définir un nouveau système :

$$y = (y_1, y_2, \dots, y_r) = (s, \dot{s}, \dots, s^{(r-1)}) \quad (6.49)$$

Donc la formule suivante est obtenue :

$$s^{(r)} = \Psi(y, t) + \Phi(y, t).u \quad \text{avec } \Phi(y, t) \neq 0 \quad (6.50)$$

Avec $\Psi(y, t)$ et $\Phi(y, t)$ des fonctions bornées telle que :

$$\begin{aligned} 0 < K_m \leq \Phi(\xi, t) \leq K_M \\ |\Psi(\xi, t)| < C_0 \end{aligned} \quad (6.51)$$

En posant :

$$\begin{aligned} N_{1,r} &= |s|^{(r-1)/r} \\ N_{i,r} &= (|s|^{p/r} + |\dot{s}|^{p/r-1} + \dots + |s^{(i-1)}|^{p/r-i+1})^{r-1/p} \quad i = 1, \dots, r-1 \\ N_{r-1,r} &= (|s|^{p/r} + |\dot{s}|^{p/(r-1)} + \dots + |s^{(r-2)}|^{p/2})^{1/p} \end{aligned} \quad (6.52)$$

La commande pour un système d'ordre r s'écrit [44]:

$$u = -\alpha \text{sign}[\psi_{r-1,r}(s, \dot{s}, \dots, s^{(r-1)})] \quad (6.53)$$

Avec,

$$\begin{aligned} \psi_{0,r} &= s \\ \psi_{1,r} &= \dot{s} + \beta_1 N_{1,r} \text{sign}(s) \quad i = 1, \dots, r-1 \\ \psi_{i,r} &= s^{(i)} + \beta_i N_{i,r} \text{sign}(\psi_{i-1,r}) \end{aligned} \quad (6.54)$$

Où $\beta_1, \dots, \beta_{r-1}$ sont des nombres positifs.

La loi de commande pour $r \leq 4$ est choisie respectivement comme suit :

$$\begin{aligned} \text{pour } r = 1 &\Rightarrow u = -\alpha \text{sign}(s) \\ \text{pour } r = 2 &\Rightarrow u = -\alpha \text{sign}\left(\dot{s} + |s|^{1/2} \text{sign}(s)\right) \\ \text{pour } r = 3 &\Rightarrow u = -\alpha \text{sign}\left(\ddot{s} + 2(|\dot{s}|^3 + |s|^2)^{1/6} \text{sign}(\dot{s} + |s|^{2/3} \text{sign}(s))\right) \\ \text{pour } r = 4 &\Rightarrow u = -\alpha \text{sign}\left\{\begin{aligned} &\ddot{s} + 3(\ddot{s}^6 + \dot{s}^4 + |s|^3)^{1/12} \text{sign}[\ddot{s} + (\dot{s}^4 + |s|^3)^{1/6} + \\ &\text{sign}(\dot{s} + \frac{1}{2}|s|^{3/4} \text{sign}(s))] \end{aligned}\right\} \end{aligned} \quad (6.55)$$

La généralisation des modes glissants d'ordre 2 aux modes glissants d'ordre supérieur, est caractérisée par le choix d'une commande discontinue agissant non pas sur la surface de glissement et sa 1ère dérivée mais sur ses dérivées supérieures, malgré les performances qu'ils manifestent, ils sont pénalisés par l'information élevée dont ils ont besoin sur les variables du système, ainsi que par le choix de la valeur des gains de commande.

6.6. Application de la commande par l'algorithme du Super-Twisting au modèle global

La commande par modes glissants classique est l'une des techniques de commande non linéaire les plus intéressantes qui existent dans la littérature. Néanmoins, sa mise en œuvre en pratique est accompagnée généralement de quelques inconvénients, tels que le phénomène de chattering et les efforts mécaniques indésirables. La commande par modes glissants d'ordre supérieur se présente alors comme une solution très intéressante dans le but de réduire les effets de ces phénomènes.

Dans notre travail nous nous sommes limités au mode glissant d'ordre deux, afin de conserver la caractéristique de réduction du phénomène de chattering tout en limitant le nombre de capteurs nécessaires pour la mise en œuvre de la commande. En outre des autres propriétés de robustesse et de précision de convergence. L'objectif à travers l'utilisation de ce type de commande est donc de générer un régime glissant du second ordre sur une surface S tout en l'annulant ainsi que sa dérivée en un temps fini ($S = \dot{S} = 0$). Dans cette stratégie de commande la partie discontinue apparaît sur la dérivée de la commande. En calculant ainsi la commande du système $u = \int \dot{u}$, celle-là devient continue, ce qui conduit à la limitation du phénomène de chattering [141].

La stratégie de commande adoptée dans cette partie est basée sur l'algorithme du Super-Twisting qui est considéré parmi les algorithmes des modes glissants d'ordre deux les plus utilisés. Son plus grand intérêt réside dans la réduction du phénomène chattering, grâce à la continuité du signal de commande. L'avantage de cet algorithme est d'utiliser uniquement l'information sur S de ne pas avoir besoin de connaître le signe de sa dérivée \dot{S} , entraînant l'annulation de S et \dot{S} en un temps fini. En fait, la mesure du signe de la dérivée de S , en temps réel est très difficile à cause des bruits [62].

La loi de commande du Super-Twisting est constituée de deux termes. Le premier terme est sa dérivée temporelle discontinue et le second est une fonction continue de la variable de glissement. Cet algorithme est considéré donc comme une généralisation non linéaire d'un PI[142].

Considérant la surface de glissement suivante :

$$S = \ddot{e}(t) + 2\lambda\dot{e}(t) + \lambda^2 e(t) \quad (6.56)$$

Sachant que $e(t) = \xi_1 - \xi_{1d}$ est l'erreur. Et λ est une matrice diagonale définie positive.

On dérive S par rapport au temps jusqu'à l'apparition des entrées V_q .

$$\dot{S} = \ddot{\ddot{e}}(t) + 2\lambda\ddot{e}(t) + \lambda^2 \dot{e}(t) \quad (6.57)$$

$$\dot{S} = \ddot{\xi}_1 - \ddot{\xi}_{1d} + 2\lambda(\ddot{\xi}_1 - \ddot{\xi}_{1d}) + \lambda^2(\dot{\xi}_1 - \dot{\xi}_{1d}) \quad (6.58)$$

Par substitution de (6.23) dans (6.58), nous obtenons :

$$\begin{aligned} \dot{S} = & (-\dot{\rho}_1\rho_2 - \rho_1\dot{\rho}_2 - 2\lambda\rho_1\rho_2 - K_i\rho_1 \frac{p\phi_f}{L_q}\eta + \rho_1^2\rho_2^2 + \lambda^2)\xi_2 + \left(K_i\dot{\rho}_1 + 2\lambda K_i\rho_1 - K_i\rho_1 \frac{R_s}{L_q} - \right. \\ & \left. \rho_1^2\rho_2 K_i\right)\xi_3 - \frac{1}{\eta}(\rho_1 + \dot{\rho}_1 - \rho_1^2\rho_2)g - \frac{1}{\eta}(\rho_1 + \dot{\rho}_1 - \rho_1^2\rho_2)F - \frac{1}{\eta}\rho_1\dot{g} - \frac{1}{\eta}\rho_1\dot{F} - \ddot{\xi}_{1d} - 2\lambda\dot{\xi}_{1d} - \\ & \lambda^2\xi_{1d} + \left(\frac{K_i\rho_1}{L_q}\right)V_q \end{aligned} \quad (6.59)$$

Avec : $\rho_1 = \left(\frac{M}{\eta} + \eta J\right)^{-1}$ et $\rho_2 = \left(\frac{C}{\eta} + \eta f\right)$

Le degré relatif de S par rapport à l'entrée V_q est égal à 1. La loi de commande par Super-Twisting est exprimée par la relation suivante :

$$\begin{cases} \dot{V}_q = -\gamma |S|^{1/2} \text{sign}(S) + V_1 \\ \dot{V}_1 = -W \text{sign}(S) \end{cases} \quad (6.60)$$

Le principe de l'application de cette loi de commande est présenté sur la figure (6.13).

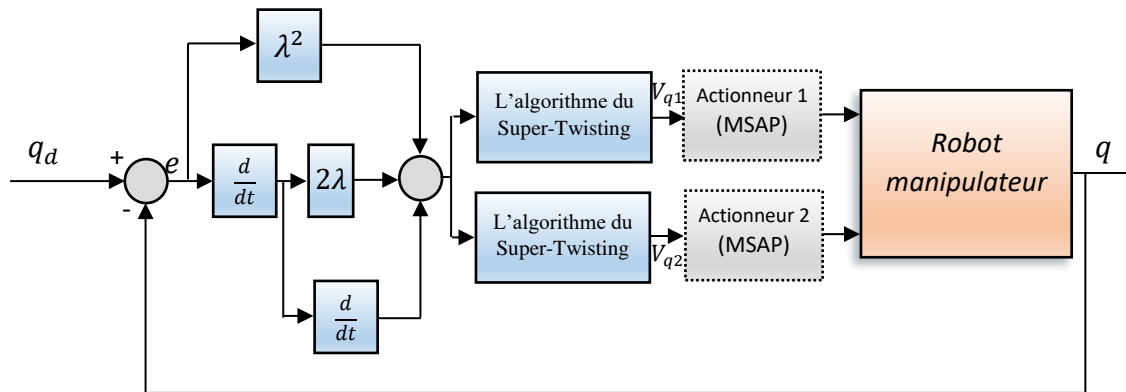


Figure 6.13. Schéma de principe de la commande par Super-Twisting

6.6.1. Résultats de simulation

Afin de tester les performances et la robustesse de la commande par mode glissant d'ordre 2 basé sur l'algorithme de Super-Twisting du modèle global du bras manipulateur, des séries de simulations numériques similaires à celles réalisées dans la partie précédente ont été établies.

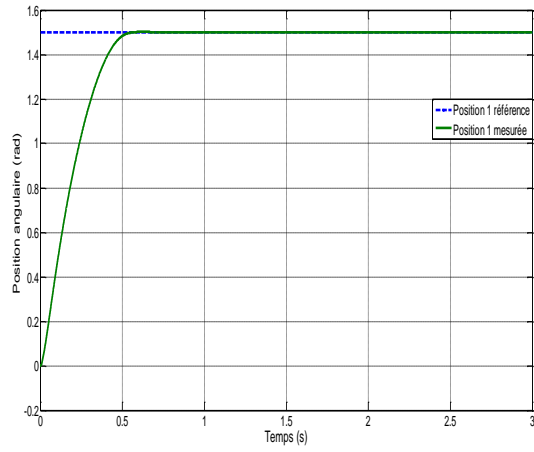
Le même modèle dynamique du robot manipulateur est considéré (voir section B.2 dans l'ANNEXE B), les paramètres du robot manipulateur utilisés sont mentionnés aussi dans l'ANNEXE B. Les paramètres des actionneurs sont indiqués dans le tableau 6.1.

Les mêmes trajectoires de la section précédente sont considérées dans cette partie.

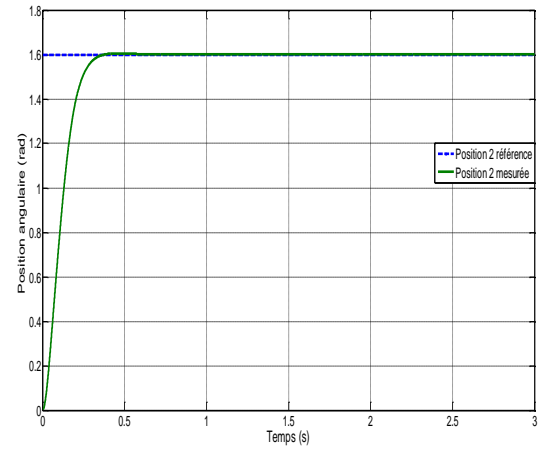
Paramètres de la commande

$\lambda_1 = 30$, $\lambda_2 = 50$, $W_1 = 200$, $W_2 = 2$, $\gamma_1 = 50$, $\gamma_2 = 5$.

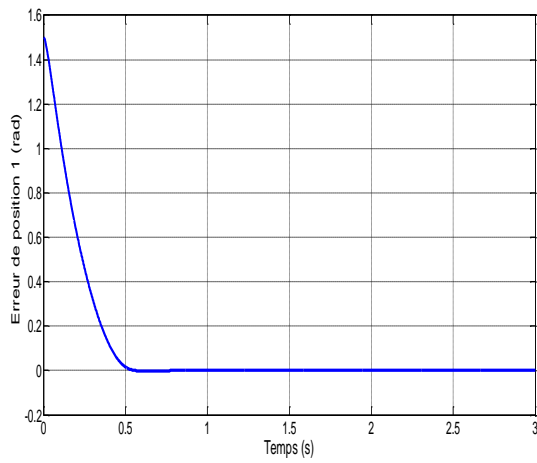
Dans les figures (6.14), (6.15) et (6.16), nous présentons les résultats de simulation que nous avons obtenus.



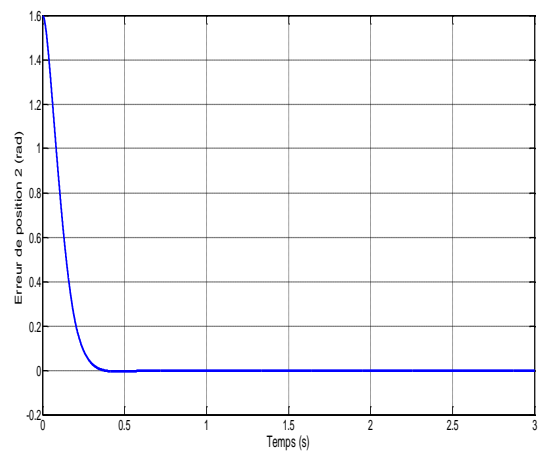
(A)



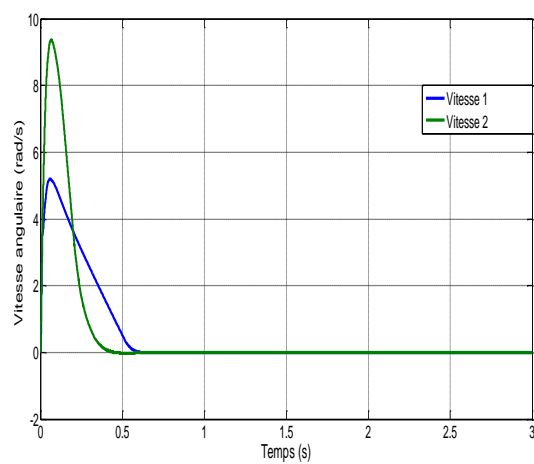
(B)



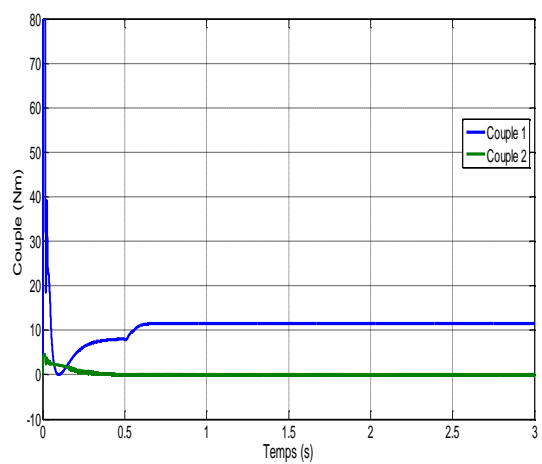
(C)



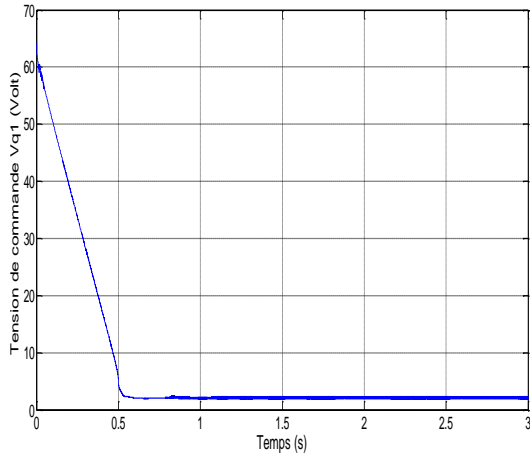
(D)



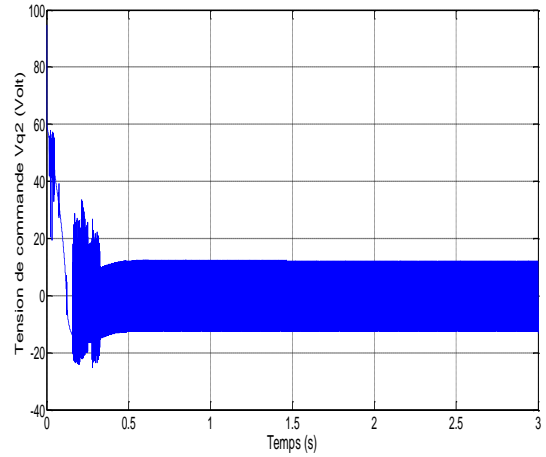
(E)



(F)

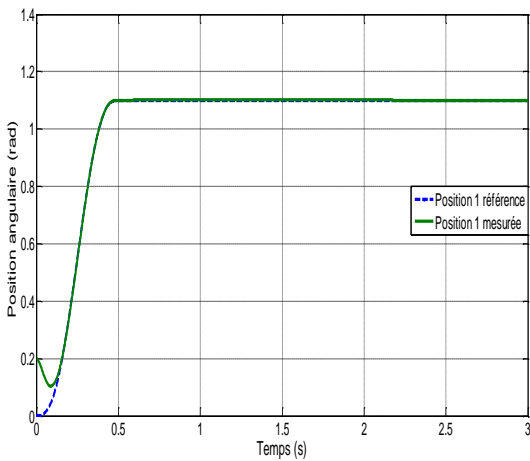


(G)

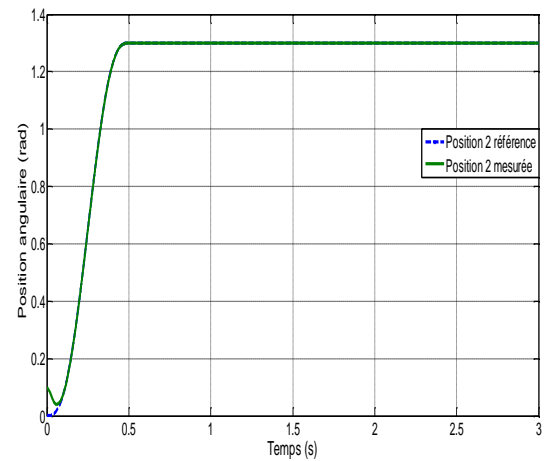


(H)

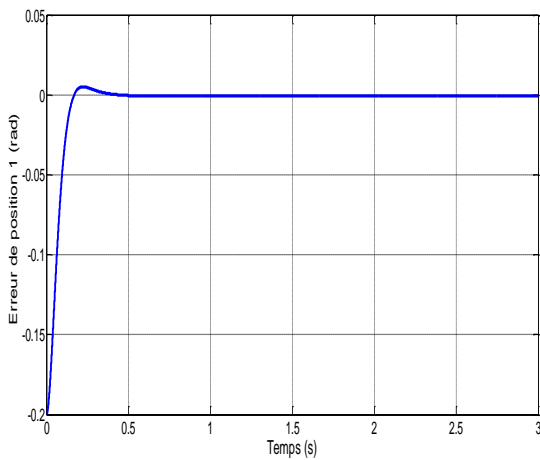
Figure 6.14. Commande par l'algorithme du Super-Twisting du modèle global avec consigne constante : (A) et (B) positions angulaires, (C) et (D) erreurs des positions, (E) vitesses angulaires et (F) les couples moteur, (G) et (H) tensions de commande.



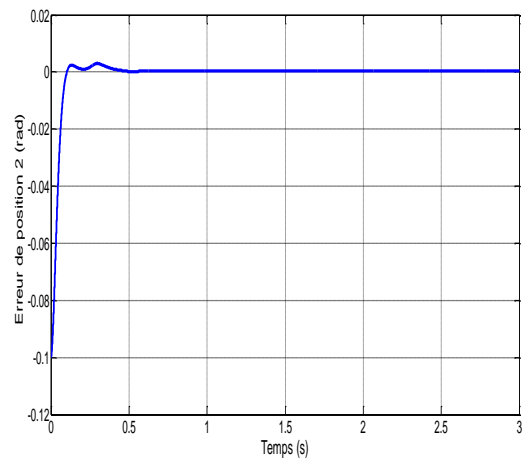
(A)



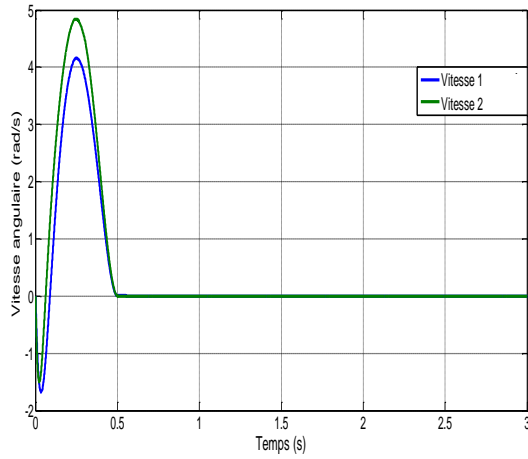
(B)



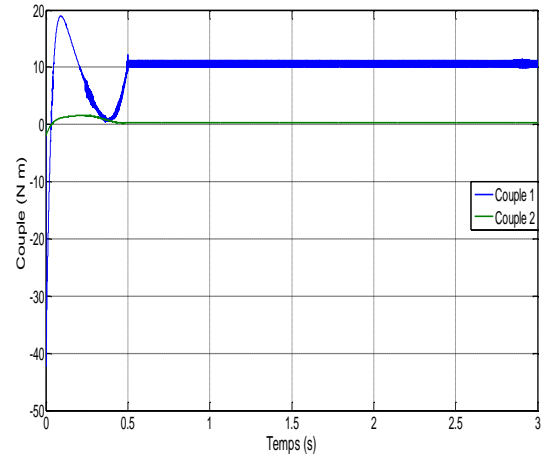
(C)



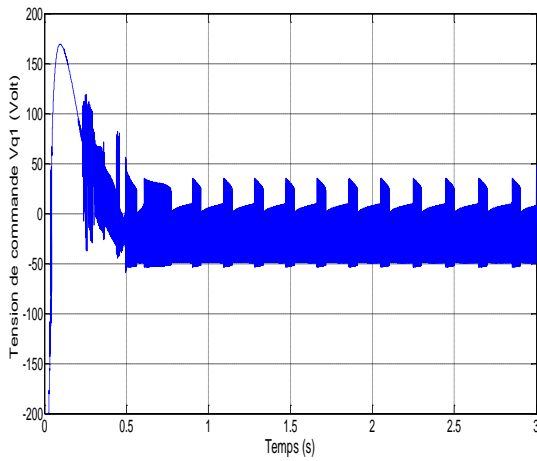
(D)



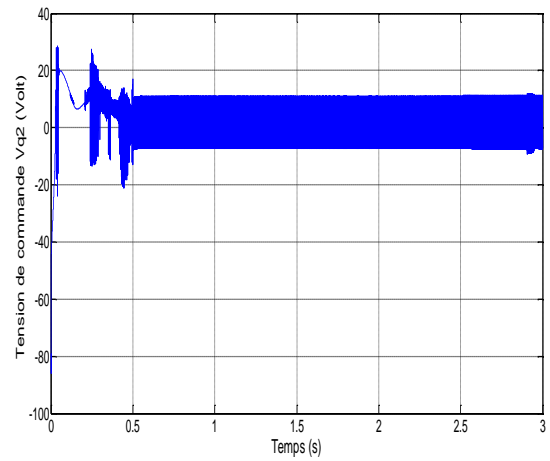
(E)



(F)

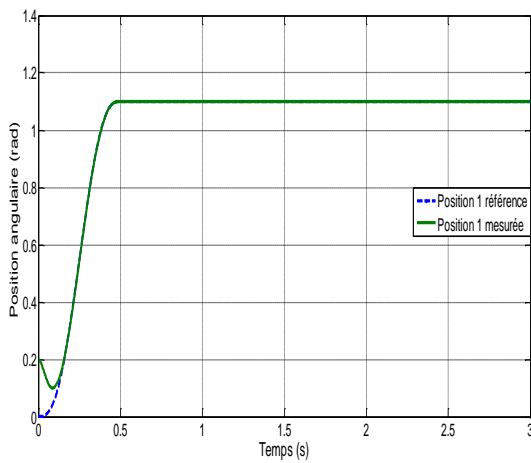


(G)

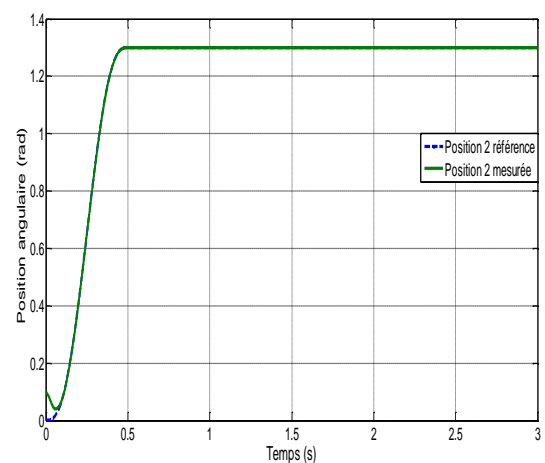


(H)

Figure 6.15. Commande par l'algorithme du Super-Twisting en poursuite de trajectoire avec conditions initiales $[0.2, 0.1]$: (A) et (B) positions angulaires, (C) et (D) erreurs des positions, (E) vitesses angulaires et (F) les couples moteur, (G) et (H) tensions de commande.



(A)



(B)

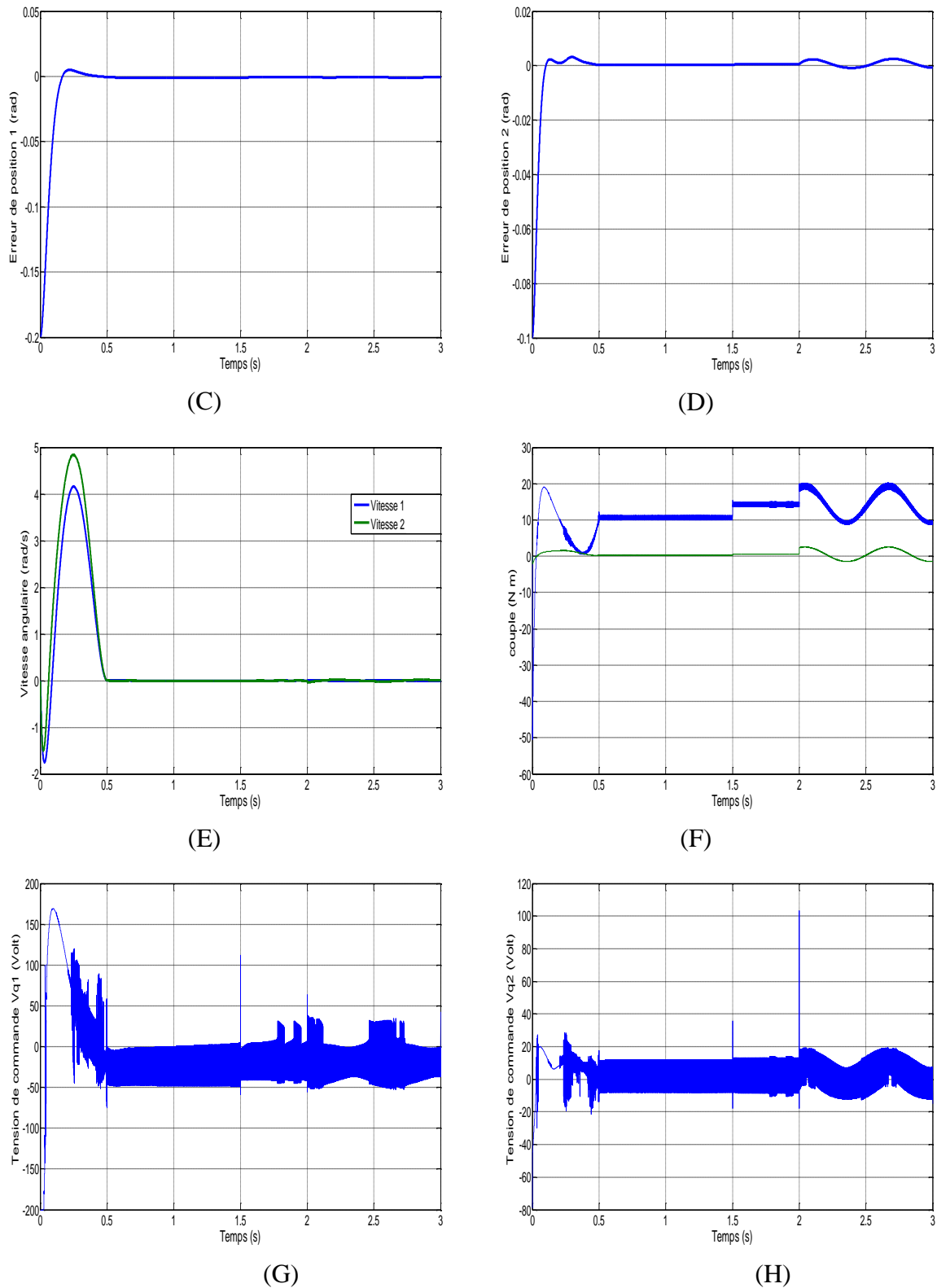


Figure 6.16. Commande par l'algorithme du Super-Twisting en poursuite de trajectoire avec variation paramétrique et perturbation externe, les conditions initiales sont $[0.2, 0.1]$: (A) et (B) positions angulaires, (C) et (D) erreurs des positions, (E) vitesses angulaires et (F) les couples moteur, (G) et (H) tensions de commande.

Synthèse 2

Les résultats de simulation sont obtenus en appliquant la commande basée sur l'algorithme du Super-Twisting. Des figures (6.14.A), (6.14.B), (6.14.C) et (6.14.D), nous pouvons voir que les deux positions articulaires 1 et 2 rejoignent leurs positions désirées avec un temps de réponse presque similaire à celui de la commande par modes glissants classique. Les erreurs de position mesurées dans l'état d'équilibre sont égales à zéro pour les deux axes. Nous pouvons remarquer aussi que les couples générés ne dépassent pas les $\pm 15\text{Nm}$ et $\pm 4\text{Nm}$ pour la première et la deuxième articulation, respectivement qui sont les plages admissibles. Le couple maximale appliqué à la première articulation atteint une valeur 80 Nm durant le démarrage. Les figures (6.14.G) et (6.14.H) montrent le signal de commande appliqué en tension. On remarque l'absence de toute variation brusque et que les sollicitations de commande sont raisonnables. Nous pouvons constater à travers ces résultats qu'il y a une baisse considérable du phénomène de chattering.

La commande Super-Twisting nous a permis d'avoir un bon suivi de trajectoire en position et en vitesse pour les deux articulations ce qui est illustré à travers les figures (5.15) avec un temps de réponse de l'ordre de 0.1 s, pour des conditions initiales de [0.2, 0.1]. Nous remarquons aussi une diminution considérable du phénomène de chattering par rapport à la commande par modes glissants classique. Les signaux commande commutent à une fréquence très élevée avec une amplitude moins importante.

Les figures (6.16) illustrent les courbes de poursuite de position et vitesse avec l'introduction d'une variation paramétrique et d'une perturbation externe aux instants 1.5 s et 2 s, respectivement dans le but de mettre en évidence la robustesse de cette loi de commande. Nous avons toujours une bonne poursuite de trajectoire en position et en vitesse, ce qui est illustré à travers les figures (6.16.A), (6.16.B) et (6.16.E). De la figure (6.16.D), nous pouvons voir une petite influence de la variation paramétrique à partir de l'instant 1.5 s ainsi que la perturbation externe à l'instant 2 s, sur les erreurs de poursuite, spécialement sur la deuxième articulation. Une variation brusque des couples intervienne aux instants d'application de la variation paramétrique et des perturbations tels qu'il est remarqué sur la figure (6.16.F). Nous constatons enfin que l'utilisation de la commande par l'algorithme du Super-Twisting a permis de réduire considérablement le phénomène de chattering tout en gardant presque les mêmes performances obtenues en utilisant une commande par modes glissants classique.

6.7. Conclusion

Nous avons présenté dans ce chapitre, en premier lieu le moteur synchrone à aimant permanent réputé d'être robuste et facile à contrôler. Ce qui nous a permis d'élaborer un modèle global associant le modèle du robot manipulateur avec le modèle de la MSAP. Ce modèle global est jugé être plus réaliste pour représenter un robot manipulateur.

En se basant sur le modèle global nous avons pu concevoir une commande par mode glissant classique et une autre commande basée sur l'algorithme du Super-Twisting assurant la stabilité du système en boucle fermé vers le point d'équilibre. Donc, l'erreur de poursuite tend vers zéro, et nous avons une bonne poursuite de trajectoire dans les deux cas.

Nous avons testé les commandes appliquées à notre système par des simulations numériques, ce qui nous a permis de constater que les performances sont satisfaisantes à savoir la poursuite

de la trajectoire, le dépassement, le temps de réponse, la robustesse pour le réglage de position et de vitesse.

L'utilisation de la commande par l'algorithme du Super-Twisting nous a permis de réduire considérablement le phénomène de chattering par rapport à la commande par modes glissants d'ordre 1. Tout en gardant presque les mêmes performances.

CONCLUSION GÉNÉRALE ET PERSPECTIVES

Dans ce travail, nous avons abordé la problématique de commande des robots manipulateurs, en optant pour des commandes non linéaires robustes et optimisées par des approches métaheuristiques, et en tenant compte de la dynamique de l'actionneur, de type MSAP dans notre cas.

Afin d'élaborer cette thèse, nous avons commencé par des généralités sur la robotique dans le premier chapitre. Nous avons exposé, ensuite les différents types de modélisation existant dans le cas des robots manipulateurs rigides. Nous nous sommes intéressés particulièrement au modèle dynamique vu son importance pour la suite de notre travail, un exemple de modélisation d'un bras manipulateur à 2ddl a été développé. Nous avons noté la nature du modèle qui est non linéaire, multivariable et couplé.

Dans le deuxième chapitre, en se basant sur le modèle développé dans le chapitre 1, nous avons appliqué plusieurs commandes classiques telles que le PID, la commande point à point et la commande du couple calculé pour commander notre robot. Plusieurs résultats de simulation ont été donnés. Les résultats obtenus avec ces lois de commande classique sont acceptables. Néanmoins, ce type de loi de commande souffre en termes de robustesse.

Le troisième chapitre, a été consacré à la présentation de quelques lois de commande non linéaires robustes, en se limitant à deux d'entre elles, la commande par modes glissants et la commande Backstepping.

Dans le quatrième chapitre, nous avons présenté le concept théorique de l'optimisation ainsi que les différents types de méthodes utilisées. Nous nous sommes focalisé sur deux approches métaheuristiques, très réputés pour leurs efficacités de trouver des solutions optimales, la première inspirée de l'évolution naturelle est la méthode des algorithmes génétiques. Et la deuxième qui s'inspire du comportement naturel des animaux est la méthode d'optimisation par essaim de particule PSO.

Dans le cinquième chapitre, nous avons appliqué au robot manipulateur les différentes lois de commande non linéaire robuste, à savoir la commande par modes glissants et la commande Backstepping. Des tests de simulation ont été faits sans et avec perturbations affectant le système tels que un choc d'impact et la variation paramétrique. Les résultats obtenus montrent que les commandes assurent de meilleures performances en termes de régulation et de poursuite de trajectoire de référence. Par contre, le souci qui se posait est de trouver les paramètres des deux commandes, nous avons opté à l'utilisation des deux méthodes d'optimisation citées auparavant avec l'utilisation de diverses fonctions objectifs. A travers les simulations, nous avons remarqué que la méthode des algorithmes génétiques donne relativement de meilleurs résultats que le PSO, mais le temps consommé pour trouver la solution optimale était plus important. Nous avons constaté aussi que la taille de la population choisie joue un rôle important dans la qualité des résultats, Plus le nombre d'individus ou de particules est grand, plus la chance de tomber sur un optimum global est meilleure. Par contre, le temps de calcul risque

d'être plus important. Nous avons remarqué aussi que l'utilisation de la fonction objectif basée sur l'erreur quadratique (MSE) donne relativement de meilleurs résultats. L'utilisation de la commande par Backstepping nous a permis de contourner le problème du phénomène de chattering typique à l'utilisation de la commande par modes glissants, avec une amélioration des performances requises.

Le sixième chapitre a été réservé à l'introduction de la dynamique de l'actionneur. L'actionneur que nous avons choisi dans notre cas est le moteur synchrone à aimant permanent, pour sa simplicité, sa facilité de commande et sa robustesse par rapport à d'autres types d'actionneurs.

Un modèle global a été établi, en lui appliquant deux lois de commande non linéaires robustes par modes glissants classique et la commande basée sur l'algorithme du Super-Twisting. Les résultats de simulation montrent que nous avons obtenus de bonnes performances, un temps de réponse acceptable pour les deux articulations, et une bonne poursuite de trajectoire. L'application d'une variation paramétrique et d'une perturbation a permis de confirmer la robustesse des deux lois de commande. Nous avons constaté que la commande par l'algorithme du Super-Twisting en plus d'être facile à concevoir, elle nous a permis de réduire considérablement le phénomène de chattering. En plus de la non-nécessité du modèle du système a contrôlé. Chose que nous avons remarquée lors de la conception de la commande par modes glissants d'ordre un et de la commande par Backstepping en utilisant une forme explicite du modèle global afin de faire apparaître toutes les variables du système. Ce qui a abouti à des formules gigantesques, impossible à appliquer.

Bien entendu, notre travail est loin d'être achevé, un certain nombre de points restent à explorer. A titre d'exemple :

- Valider les commandes non linéaires robustes développées dans ce travail sur un prototype réel.
- Utilisation des méthodes d'optimisation vues dans cette thèse pour la génération des trajectoires optimales.
- Tester d'autres algorithmes de commande en associant les techniques d'intelligence artificielle.
- Etendre ce travail sur un bras manipulateur flexible.
- Il serait plus intéressant d'utiliser des calculateurs plus puissants dans l'étape d'optimisation, ce qui permettrait de réduire le temps de calcul.
- La mise en œuvre des lois de commande pour les robots manipulateurs est basée sur la connaissance de tous les états du système à chaque instant, chose qui n'est pas vraie en réalité, comme le vecteur vitesse qui peut ne pas être disponible généralement pour plusieurs raisons. Il serait intéressant dans ce cas de concevoir des observateurs non linéaires permettant la reconstruction du signal manquant.

ANNEXE A

STABILITÉ AU SENS DE LYAPUNOV

A.1. Notion de stabilité

La théorie de la stabilité joue un rôle central dans la théorie et l'ingénierie des systèmes. Il existe différents types de problèmes de stabilité qui se posent dans l'étude des systèmes dynamiques. Dans cette partie nous nous intéressons à la stabilité des points d'équilibre, mais il existe d'autres types de stabilité, tel que la stabilité des orbites périodiques et la stabilité entrée-sortie. La stabilité des points d'équilibre est généralement caractérisée dans le sens de Lyapunov, un mathématicien et ingénieur russe qui a énoncé les bases de la théorie, qui porte maintenant son nom [143]. Par définition, si un système est dans un état d'équilibre, il restera dans cet état pour t variant dans le temps. L'étude de la stabilité au sens de Lyapunov consiste en l'étude des trajectoires du système quand l'état initial est "près" d'un état d'équilibre. Cela reflète la possibilité de perturbations affectant le système, sous forme de conditions initiales non nulles [143]. Un système est qualifié de stable si le démarrage du système quelque part près de son point d'équilibre souhaité implique qu'il restera autour de ce point pour toujours.

Définition A.1 : point d'équilibre

Considérons le système autonome :

$$\dot{x} = f(x) \tag{A.1}$$

Où $x \in \mathbb{R}^n$ est le vecteur d'état et $f: D \rightarrow \mathbb{R}^n$ une fonction localement lipchitzienne et continue sur un domaine D de \mathbb{R}^n pour cette équation.

Un état x^* est un état d'équilibre (ou point d'équilibre) du système si une fois que $x(t)$ est égal à x^* , il reste toujours égal à x^* . Mathématiquement, cela signifie que le vecteur constant x^* satisfait

$$f(x^*) = 0 \tag{A.2}$$

Définition A.2 : Boule dans \mathbb{R}^n

On définit une boule fermée dans \mathbb{R}^n comme l'ensemble :

$$B_r = \{x \in \mathbb{R}^n / \|x\| \leq r\} \tag{A.3}$$

où la norme $\| \cdot \|$ est une norme sur \mathbb{R}^n .

Définition A.3 : stabilité du point d'équilibre

Considérons le système donné par l'équation (A.1).

L'état d'équilibre x_e est dit **stable** si $\forall \varepsilon > 0, \exists \delta > 0$ tel que si $\|x(0) - x_e\| < \delta$ alors $\|x(t) - x_e\| < \varepsilon \quad \forall t \geq 0$. Dans le cas contraire, x_e est dit **instable**.

$$(\forall \varepsilon > 0)(\exists \delta > 0)(\|x(0) - x_e\| < \delta \Rightarrow \|x(t) - x_e\| < \varepsilon, \forall t \geq 0) \quad (\text{A.4})$$

La stabilité au sens de Lyapunov signifie que la trajectoire d'état peut être gardée arbitrairement près de x_e , si l'on prend une condition initiale suffisamment proche de x_e .

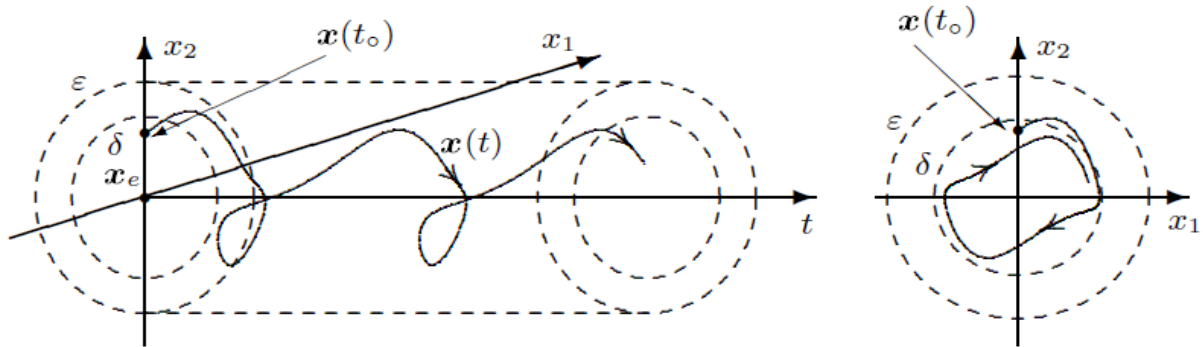


Figure A.1. Notion de stabilité

Définition A.4 : *stabilité asymptotique*

Un point d'équilibre x_e est **asymptotiquement stable** s'il est stable et s'il existe $\delta > 0$ tels que :

$$\|x(0) - x_e\| < \delta \Rightarrow \lim_{t \rightarrow +\infty} x(t) = x_e \quad (\text{A.5})$$

La stabilité asymptotique signifie que non seulement l'équilibre est stable mais que de plus on est capable de déterminer un voisinage du point d'équilibre tel que n'importe quelle trajectoire, issue d'un $x(0)$ appartenant à un voisinage de x_e , tend vers x_e quand $t \rightarrow +\infty$, tels qu'il est montré sur la figure A.2.

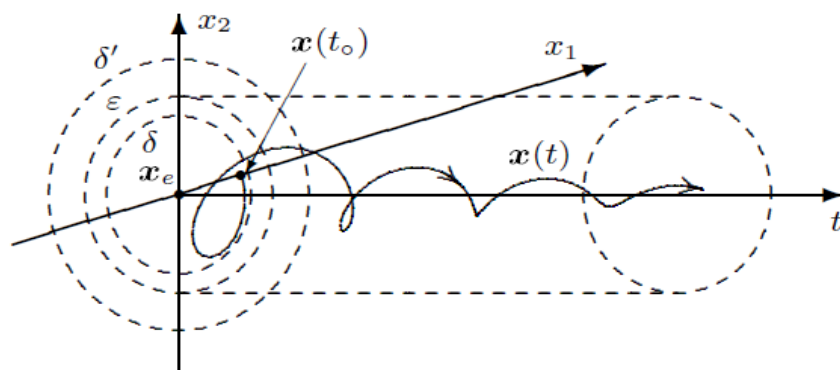


Figure A.2. Stabilité asymptotique

Définition A.5 : *stabilité exponentielle*

Un point d'équilibre x_e est **exponentiellement stable** s'il existe deux nombre strictement positifs $\alpha > 0$ et $\beta > 0$ tels que :

$$\forall t > 0, \exists B_r(x_e, r), \forall x(0) \in B_r, \|x(t) - x_e\| \leq \alpha \|x(0) - x_e\| e^{-\beta t} \quad (\text{A.6})$$

Définition A.6 : stabilité globale

Si la propriété de stabilité asymptotique, (exponentielle) est vérifiée quelque soit l'état initial, le point d'équilibre est **globalement asymptotiquement (exponentiellement) stable**.

A.2. Seconde méthode de Lyapunov ou méthode directe

Les définitions citées ci-dessus concernent $x(t)$, la solution explicite de (A.1), qui généralement ne peut pas être trouvée analytiquement. Lyapunov a eu l'idée de proposer une autre manière de prouver la stabilité en incorporant le vecteur d'état $x(t)$ dans une fonction scalaire $V(x)$ afin de quantifier la distance entre $x(t)$ et le point d'équilibre du système. La philosophie de la méthode réside dans l'extension mathématique d'une observation fondamentale de la physique : "Si l'énergie totale d'un système est dissipée de manière continue alors le système, (qu'il soit linéaire ou non linéaire), devra rejoindre finalement un point d'équilibre". On pourra donc conclure à la stabilité d'un système par l'examen d'une seule fonction scalaire, ici l'énergie totale. $V(x)$ peut donc être considérée comme une représentation de l'énergie contenue dans le système. S'il est possible de montrer que $V(x)$ est continue et strictement décroissante selon t , alors le système lui-même doit se déplacer vers l'équilibre. Cette approche démontrant la stabilité s'appelle la méthode directe de Lyapunov. Définissons d'abord quelques concepts utiles [11].

Définition A.8 : Fonction définie positive

Une fonction définie positive est une fonction $V(x): R^n \rightarrow R$ telle que $V(x) > 0, \forall x \neq 0$ et $V(0) = 0$.

Définition A.7 : Fonction semi-définie positive

Une fonction semi-définie positive est une fonction $V(x): R^n \rightarrow R$ telle que $V(x) \geq 0, \forall x \neq 0$ et $V(0) = 0$.

Le théorème de stabilité fondamental de la théorie de Lyapunov peut maintenant être énoncé. Considérons le système (A.1) et $f(0) = 0$.

Théorème A.1 : Seconde méthode de Lyapunov, dite aussi méthode directe

Soit $V(x)$ est une fonction scalaire semi-définie positive, bornée, continue et différentiable dans le temps, si :

$$\dot{V}(x) = \frac{d}{dt} V(x) = \frac{\partial V}{\partial x} \frac{dx}{dt} = \frac{\partial V}{\partial x} \dot{x} = \frac{\partial V(x)}{\partial x} f(x) < 0; \forall x \neq 0 \quad (\text{A.7})$$

Alors $x = x_e$ est un point d'équilibre **globalement asymptotiquement stable**. Une fonction scalaire $V(x)$ qui satisfait la condition $\dot{V}(x) < 0$ est appelée alors, la fonction de Lyapunov du système.

Théorème A.2 :

Soit $V(x)$ est une fonction scalaire Semi définie positive, bornée, continue et différentiable dans le temps, si :

$$\dot{V}(x) = V(x)f(x) < 0; \forall x \neq 0, S = \{x: \dot{V}(x) = 0\} \quad (\text{A.8})$$

Et supposons que le système (A.1) admet une solution unique $x(t) = x_e$. Si $x(t)$ reste au voisinage de S , alors $x(t) = x_e$ est un point d'équilibre globalement asymptotiquement stable.

Théorème A.3 :

Le point d'équilibre $x(t) = x_e$ du système d'équation (A.1) est dit uniformément Asymptotiquement Stable si toute trajectoire solution du système comprise dans une boule de rayon R de grandeur quelconque et initialisée en x_0 dans une boule de rayon R_0 dépendant de R converge vers le point d'équilibre quand t tend vers ∞ .

$$\forall R > 0, \exists r > 0 \text{ tel que } \|x_0\| < r \Rightarrow \|x_t\| < R \quad (\text{A.9})$$

En théorie du contrôle, c'est-à-dire lorsque nous devons analyser la stabilité d'un système particulier, trouver une fonction de Lyapunov avec une dérivée définie négative est en général très difficile. Néanmoins, si malgré des efforts laborieux nous ne parvenons pas à trouver une fonction de Lyapunov, nous ne devons pas conclure que l'origine du système analysé est instable ; au contraire, aucune conclusion ne peut être tirée. Heureusement, pour les systèmes autonomes, il existe des méthodes basées sur des conditions plus restrictives mais considérablement plus faciles à vérifier. Un exemple notable est le principe d'invariance de La Salle qui est largement utilisé dans l'analyse des systèmes de commande de robot [10].

Théorème A.4 : La Salle [10]

Soit $\Omega \subset D$ un ensemble compact et invariant par rapport au système (A.1). Soit $V: D \rightarrow \mathbb{R}$ une fonction continue différentiable et telle que $\dot{V}(x) \leq 0$ sur Ω . Soit E un ensemble de point de Ω tel que $\dot{V}(x) = 0$. Soit M le plus grand ensemble de E , alors chaque solution qui commence dans Ω approche M quand $t \rightarrow +\infty$.

Lemme A.1 : Lemme de Barbalat

Soit $\phi: \mathbb{R} \rightarrow \mathbb{R}$ une fonction uniformément continue sur $[0, +\infty[$. On suppose que $\lim_{t \rightarrow +\infty} \int_0^t \phi(\tau) d\tau$ existe et finie, alors : $\lim_{t \rightarrow +\infty} \phi(t) = 0$.

ANNEXE B

PARAMÈTRES DES ROBOTS MANIPULATEURS

La description des robots manipulateurs ainsi que les modèles utilisés pour la simulation des lois de commande dans notre travail sont montrés ci-dessous :

B.1. 1^{er} modèle :

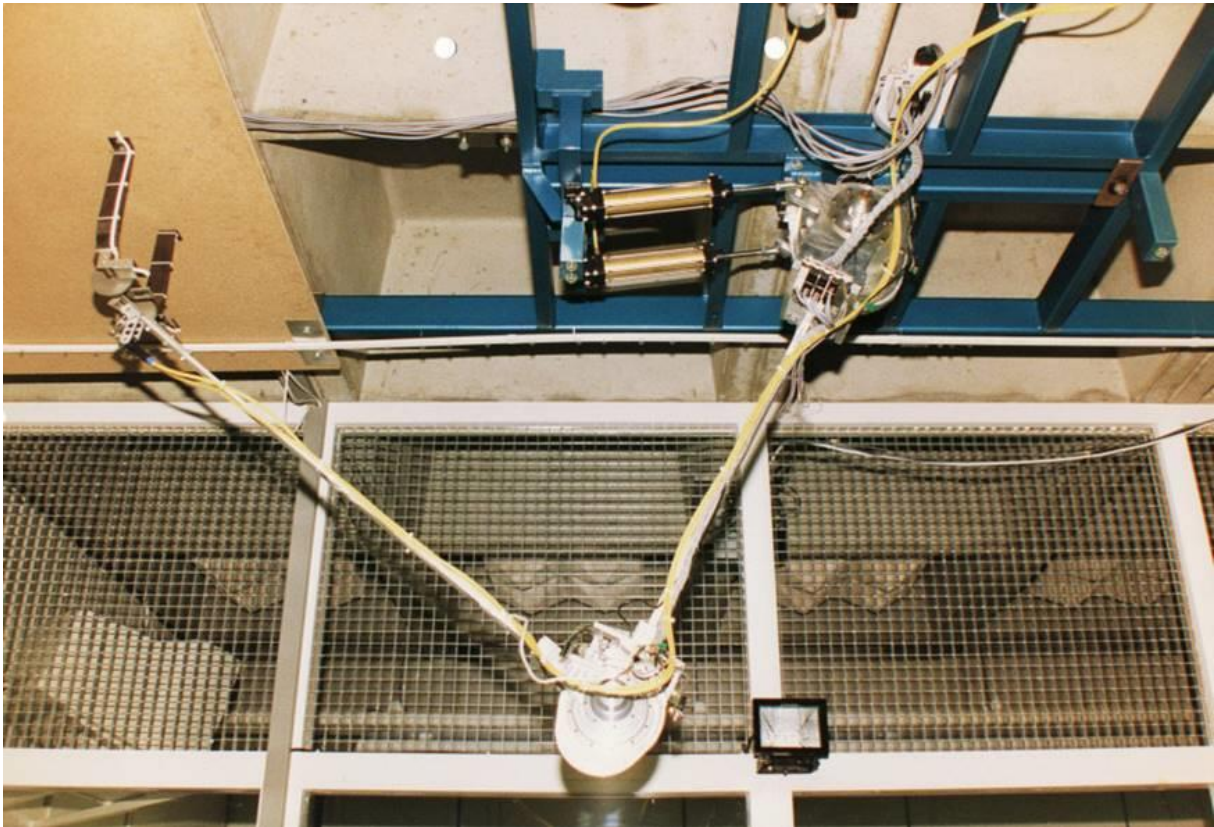


Figure B.1. Prototype réel du 1er modèle

B.1.1. Paramètres du robot [144]

m_1 : masse du premier segment ($\approx 1\text{kg}$),

m_2 : masse du deuxième segment ($\approx 1,5\text{kg}$),

m_a : masse du coude de l'articulation ($\approx 5,5\text{kg}$),

l_1 : longueur du premier segment ($\approx 1\text{m}$),

l_2 : longueur du deuxième segment ($\approx 1\text{m}$).

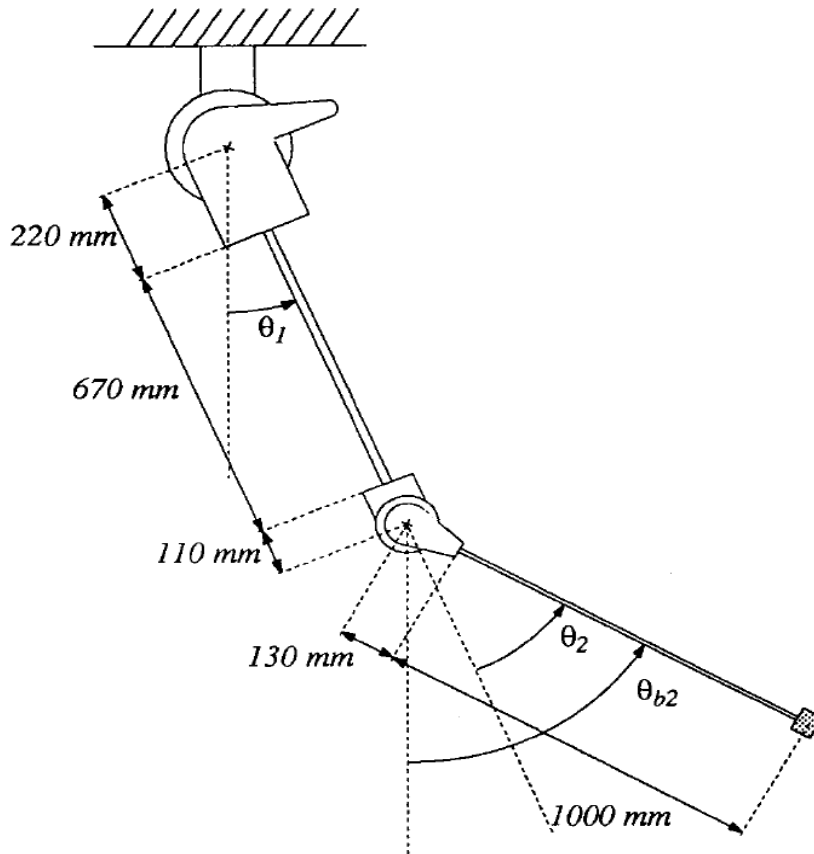


Figure B.2. Robot manipulateur avec dimensions

B.1.2. Modèle du robot manipulateur [144]

Le modèle est décrit par l'équation suivante :

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) \quad (\text{B.1})$$

Avec :

$$M(q) = \begin{bmatrix} 8.77 + 1.02 \cos(q_2) & 0.76 + 0.51 \cos(q_2) \\ 0.76 + 0.51 \cos(q_2) & 0.62 \end{bmatrix}$$

$$C(q, \dot{q}) = \begin{bmatrix} -0.5 \sin(q_2) \dot{q}_2 & -0.5 \sin(q_2) (\dot{q}_1 + \dot{q}_2) \\ 0.5 \sin(q_2) \dot{q}_1 & 0 \end{bmatrix}$$

$$\text{Et } G(q) = 10 \begin{bmatrix} 7.6 \sin(q_1) + 0.63 \sin(q_1 + q_2) \\ 0.63 \sin(q_1 + q_2) \end{bmatrix}$$

B.2. 2^{ème} modèle [39], [124]

Le mécanisme est constitué de segments rigides et actionnés par des servo-actionneurs à entraînement direct pour entraîner les articulations sans démultiplication. Ces actionneurs (de Parker Compumotor) fonctionnent en mode couple, ils agissent donc comme des sources de couple et acceptent une tension analogique comme signal de référence. Le bras réel utilise les modèles DM1015B (15 Nm) et DM1004C (4 Nm).

Figure B.1. Prototypé réel du 2^{ème} modèle

Le modèle est décrit toujours par l'équation (B.1) dont:

La matrice d'inertie $M(q)$:

$$M(q) = \begin{bmatrix} m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos q_2) + I_1 + I_2 & m_2 (l_{c2}^2 + l_1 l_{c2} \cos q_2) + I_2 \\ m_2 (l_{c2}^2 + l_1 l_{c2} \cos q_2) + I_2 & m_2 l_{c2}^2 + I_2 \end{bmatrix} \quad (\text{B.2})$$

La matrice des termes Coriolis et centrifuges $C(q, \dot{q})$:

$$C(q, \dot{q}) = \begin{bmatrix} -m_2 l_1 l_{c2} \sin q_2 \dot{q}_2 & -m_2 l_1 l_{c2} \sin q_2 \dot{q}_1 - m_2 l_1 l_{c2} \sin q_2 \dot{q}_2 \\ m_2 l_1 l_{c2} \sin q_2 \dot{q}_1 & 0 \end{bmatrix} \quad (\text{B.3})$$

Le vecteur des termes des forces de gravité $g(q)$:

$$g(q) = \begin{bmatrix} (m_1 l_{c1} + m_2 l_1) g \sin q_1 + m_2 l_{c2} g \sin(q_1 + q_2) \\ m_2 l_{c2} g \sin(q_1 + q_2) \end{bmatrix} \quad (\text{B.4})$$

Paramètres du 2^{ème} modèle

Paramètres	Valeur
Longueur du 1 ^{er} segment (l_1)	0.26 m
Longueur du 2 ^{ème} segment (l_2)	0.26 m
Centre de masse 1 ^{er} segment (l_{c1})	0.0983 m
Centre de masse 2 ^{ème} segment (l_{c2})	0.0229 m
masse 1 ^{er} segment (m_1)	6.5225 kg
masse du 2 ^{ème} segment (m_2)	2.0458 kg
Moment d'inertie 1 ^{er} segment (I_1)	0.1213 kg m ²
Moment d'inertie 2 ^{ème} segment (I_2)	0.0116 kg m ²
Gravité (g)	9.81 m/s ²

Travaux scientifiques

Publications Internationales

Saidi, K., Boumédiène, A. and Boubekour, D. (2021). Genetic Algorithm Optimization of sliding Mode Controller Parameters for Robot Manipulator. *International Journal on Emerging Technologies*, 12(2): 119–127.

Saidi K, Boumediene A, and Massoum S, « An optimal PSO-based sliding-mode control scheme for the robot manipulator », *Elektrotehniski Vestnik/Electrotechnical Review*, vol. 87, no 1-2, p. 53-59, 2020.

F. Bouriachi, B. Tolbi, K. Saidi, and O. Kada Kloucha. « Efficient Traffic Signal Control for Multi-phase Intersections. In *Proceedings of the 2019 International Conference on Artificial Intelligence, Robotics and Control (AIRC '19)*. Association for Computing Machinery, New York, NY, USA, 75–79, 2019.

Communications Internationales

K. Saidi and M. Allad, « Fuzzy controller parameters optimization by using genetic algorithm for the control of inverted pendulum », in *2015 3rd International Conference on Control, Engineering & Information Technology (CEIT)*, p. 1-6, 2015.

RÉFÉRENCES BIBLIOGRAPHIQUES

- [1] J.-J. E. (Jean-J. E.) Slotine et W. Li, *Applied nonlinear control*. EngleWood Cliffs: Prentice Hall, 1991.
- [2] V. Bregeault, F. Plestan, et Institut de recherche en communications et cybernétique, « Quelques contributions à la théorie de la commande par modes glissants ». Ecole Centrale de Nantes, France, 2010.
- [3] M. Kristic, I. Kanellakopoulos, P. V. Kokotovic, et D. Mayne, « Nonlinear and adaptive control design », *IEEE Trans. Autom. Control*, vol. 41, n° 12, p. 1849-1852, 1996.
- [4] R. Sepulchre, M. Janković, et P. V. Kokotović, *Constructive Nonlinear Control*. Springer-Verlag, 1997.
- [5] R. Kelly, V. S. Davila, et J. A. L. Perez, *Control of robot manipulators in joint space*. Springer Science & Business Media, 2006.
- [6] W. Khalil et E. Dombre, *Modélisation identification et commande des robots*. Paris: Hermès, 1999.
- [7] B. Siciliano, O. Khatib, et Springer International Publishing AG, *Springer handbook of robotics*, 2nd edition. Springer Publishing Company, 2016.
- [8] D. E. Goldberg, *Genetic Algorithms in Search Optimizations & Machine Learning*. Addison-Wesley Publishing Company, Inc, 1989.
- [9] J. Kennedy et R. Eberhart, « Particle swarm optimization », in *Proceedings of ICNN'95-international conference on neural networks*, 1995, vol. 4, p. 1942-1948.
- [10] R. Kelly, V. S. Davila, et J. A. L. Perez, *Control of robot manipulators in joint space*. Springer Science & Business Media, 2006.
- [11] S. Aboura, « Génération de mouvement optimal et commande d'un bras manipulateur hyper dynamique », PhD Thesis, Ph. D. Department of Automation. University of Sciences and Technology MB ..., 2016.
- [12] A. Bahloul, « Sur la commande des robots manipulateurs industriels en co-manipulation robotique », PhD Thesis, Université Paris-Saclay (ComUE), 2018.
- [13] A. Hassam, « Calcul de trajectoire et contrôle de déplacement d'un robot », PhD Thesis, Université ferhat abbas sétif, ALGERIE, 2007.
- [14] R. N. Jazar, *Theory of applied robotics: kinematics, dynamics, and control*, Second Edition. Springer Science & Business Media, 2010.
- [15] V. M. Hernández-Guzmán et J. Orrante-Sakanassi, « Global PID control of robot manipulators equipped with PMSMs », *Asian J. Control*, vol. 20, n° 1, p. 236-249, 2018.
- [16] A. Boumédiène, M. S. Boucherit, et F. Boudjema, « A robustness comparative study of several types of speed controllers with the kind based on backstepping for PMSM control fed by a three-level voltage source inverter », *Int. Rev. Electr. Eng.-IREE*, vol. 2, n° 4, p. 619-628, 2007.
- [17] B. Siciliano, L. Sciavicco, L. Villani, et G. Oriolo, *Robotics Modelling, Planning and Control*. 2009. Consulté le: mai 17, 2021. [En ligne]. Disponible sur: <https://doi.org/10.1007/978-1-84628-642-1>
- [18] S. Jlassi, « Formulation et études des problèmes de commande en co-manipulation robotique », PhD Thesis, Université Paris Sud-Paris XI, 2013.
- [19] M. W. Spong, S. Hutchinson, et M. Vidyasagar, *Robot modeling and control*, vol. 3. wiley New York, 2006.
- [20] E. Dombre et W. Khalil, *Modeling, performance analysis and control of robot manipulators*. Wiley Online Library, 2007.
- [21] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 3^e éd. Upper Saddle River, NJ: Pearson Education International, 2005.
- [22] J. Denavit et R. S. Hartenberg, « A kinematic notation for lower-pair mechanisms based on matrices », *J. Appl. Mech.*, vol. 22, n° 2, p. 215-221, 1955.

- [23] W. Khalil, J. Kleinfinger, et 1986 IEEE International Conference on Robotics and Automation, « A new geometric notation for open and closed-loop robots », *IEEE Int. Conf. Robot. Autom.*, vol. 3, p. 1174-1179, 1986.
- [24] C. S. G. Lee et M. Ziegler, « Geometric approach in solving inverse kinematics of PUMA robots », *IEEE Trans. Aerosp. Electron. Syst.*, n° 6, p. 695-706, 1984.
- [25] A. Goldenberg, B. Benhabib, et R. Fenton, « A complete generalized solution to the inverse kinematics of robots », *IEEE J. Robot. Autom.*, vol. 1, n° 1, p. 14-20, 1985.
- [26] D. L. Peiper, « The kinematics of manipulators under computer control », Stanford univ ca dept of computer science, 1968.
- [27] R. P. Paul, *Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators*. Richard Paul, 1981.
- [28] W. A. Wolovich et H. Elliott, « A computational technique for inverse kinematics », in *The 23rd IEEE Conference on Decision and Control*, 1984, p. 1359-1363.
- [29] P. Coiffet, *Les robots. Tome 1, Tome 1*. Neuilly-sur-Seine: Hermes publishing France, 1981.
- [30] J. M. Hollerbach, « A recursive lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity », *IEEE Trans. Syst. Man Cybern.*, vol. 10, n° 11, p. 730-736, 1980.
- [31] M. Renaud, « Contribution à la modélisation et à la commande dynamique des robots manipulateurs », PhD Thesis, Univ Paul Sabatier de Toulouse, 1980.
- [32] W. W. Hooker, « The dynamical attitude equations for n-body satellite », *J Astronaut. Sci*, vol. 12, p. 123-128, 1965.
- [33] J. Y. Luh, M. W. Walker, et R. P. Paul, « On-line computational scheme for mechanical manipulators », *J. Dyn. Syst. Meas. Control*, vol. 102, n° 8, 1980.
- [34] D. E. Orin, R. B. McGhee, M. Vukobratović, et G. Hartoch, « Kinematic and kinetic analysis of open-chain linkages utilizing Newton-Euler methods », *Math. Biosci.*, vol. 43, n° 1-2, p. 107-130, 1979.
- [35] V. Dupourqué, B. Faverjon, et H. Guiot, *Les contrôleurs de robots*. INRIA, 1986.
- [36] M. F. Khelfi, M. Zasadzinski, M. Darouach, H. Rafaralahy, et E. Richard, « Reduced-order observer-based point-to-point and trajectory controllers for robot manipulators », *IFAC Proc. Vol.*, vol. 28, n° 8, p. 341-347, 1995.
- [37] F. L. Lewis, C. T. Abdallah, D. M. Dawson, et F. L. Lewis, *Robot manipulator control: theory and practice*. New York: Marcel Dekker, 2004.
- [38] M. W. Spong, S. Hutchinson, et M. Vidyasagar, *Robot modeling and control*. 2020.
- [39] R. Kelly, V. Santibáñez, et A. Loría, *Control of robot manipulators in joint space*. London: Springer, 2005.
- [40] C. A. Canudas de Wit, B. Siciliano, et G. Bastin, *Theory of robot control*, 1ère édition. London: Springer, 1996.
- [41] H. F. Ho, Y.-K. Wong, et A. B. Rad, « Robust fuzzy tracking control for robotic manipulators. », *Simul Model Pr. Theory*, vol. 15, n° 7, p. 801-816, 2007.
- [42] M. F. Khelfi, M. Zasadzinski, H. Rafaralahy, E. Richard, et M. Darouach, « Reduced-order observer-based point-to-point and trajectory controllers for robot manipulators », *Control Eng. Pract.*, vol. 4, n° 7, p. 991-1000, 1996.
- [43] J.-J. E. Slotine, « Sliding controller design for non-linear systems », *Int. J. Control*, vol. 40, n° 2, p. 421-434, 1984.
- [44] W. Perruquetti et J. P. Barbot, *Sliding mode control in engineering*. New York: M. Dekker, 2002.
- [45] M. Krstic et P. V. Kokotovic, « Nonlinear backstepping designs and applications: adaptive, robust, and optimal », 2002.
- [46] Y. Z. Tzypkin, « Theory of control relay systems », *Mosc. Gostekhizdat*, p. 212-222, 1955.
- [47] D. V. Anosov, « On stability of equilibrium points of relay systems », *Autom. Telemekh. Autom. Remote Control*, vol. 2, p. 135-149, 1959.
- [48] A. F. Filippov, « Differential equations with discontinuous right-hand side », *Mat. Sb.*, vol. 93, n° 1, p. 99-128, 1960.
- [49] S. V. Emelyanov, « On peculiarities of variables structure control systems with discontinuous switching functions », in *Doklady ANSSR*, 1963, vol. 153, p. 776-778.
- [50] S. V. Emelyanov, « Variable structure control systems », *Mosc. Nouka*, 1967.

- [51] U. Itkis, *Control systems of variable structure*. New York: John Wiley & Sons, 1976.
- [52] V. I. Utkin, *Sliding modes and their application in variable structure systems*. Moscow: Mir Publishers, 1978.
- [53] J. Slotine, J. Hedrick, E. Misawa, et 1986 25th IEEE Conference on Decision and Control, « Nonlinear state estimation using sliding observers », p. 332-339, 1986.
- [54] C. C. De Wit et J.-J. Slotine, « Sliding observers for robot manipulators », *Automatica*, vol. 27, n° 5, p. 859-864, 1991.
- [55] H. Sira-Ramirez, « On the dynamical sliding mode control of nonlinear systems », *Int. J. Control*, vol. 57, n° 5, p. 1039-1061, 1993.
- [56] T. Boukhobza, M. Djemai, et J. P. Barbot, « Nonlinear Sliding Observer for Systems in Output and Output Derivative Injection Form », *IFAC Proc. Vol. IFAC Proc. Vol.*, vol. 29, n° 1, p. 2179-2184, 1996.
- [57] H. Bühler, *Réglage par mode de glissement*. Lausanne, Suisse: Presses polytechniques romandes, 1986.
- [58] M. C. S. Alaoui, « Commande et Observateur par Modes glissants d'un système de pompage et d'un bras manipulateur », PhD Thesis, thèse de doctorat. Université Sidi Mohammed Ben Abdellah. Maroc, 2009.
- [59] V. I. Utkin, J. Guldner, et J. Shi, *Sliding mode control in electromechanical systems*. Boca Raton, FL; London: CRC Press ; Taylor & Francis, 2009.
- [60] V. I. Utkin, *Sliding modes in control and optimization*. Berlin; New York: Springer-Verlag, 1992.
- [61] D. Boubekeur, « Sur la commande d'un fauteuil roulant électrique », PhD thesis, Faculté de TECHNOLOGIE, UABT, Tlemcen, Tlemcen ALGERIE, 2016.
- [62] A. Isidori, *Nonlinear control systems*. Berlin: Springer, 2002.
- [63] V. Utkin, « Variable structure systems with sliding modes », *IEEE Trans Autom. Contr IEEE Trans. Autom. Control*, vol. 22, n° 2, p. 212-222, 1977.
- [64] J.-J. Slotine et S. S. Sastry, « Tracking control of non-linear systems using sliding surfaces, with application to robot manipulators », *Int. J. Control*, vol. 38, n° 2, p. 465-492, 1983.
- [65] Yagiz N et Hacioglu Y, « Robust control of a spatial robot using fuzzy sliding modes », *Math Comput Model Math. Comput. Model.*, vol. 49, n° 1-2, p. 114-127, 2009.
- [66] A. Hamzaoui, N. Essounbouli, et J. Zaytoon, « Fuzzy sliding mode control for uncertain SISO systems », Faro, Portugal, 2003.
- [67] H. F. Ho, Y.-K. Wong, et A. B. Rad, « Adaptive fuzzy sliding mode control with chattering elimination for nonlinear SISO systems. », *Simul Model Pr. Theory*, vol. 17, n° 7, p. 1199-1210, 2009.
- [68] N. Noroozi, M. Roopaei, et M. Z. Jahromi, « Adaptive fuzzy sliding mode control scheme for uncertain systems », *CNSNS Commun. Nonlinear Sci. Numer. Simul.*, vol. 14, n° 11, p. 3978-3992, 2009.
- [69] A. F. Amer, E. A. Sallam, et W. M. Elawady, « Adaptive fuzzy sliding mode control using supervisory fuzzy control for 3 DOF planar robot manipulators », *Appl. SOFT Comput.*, vol. 11, n° 8, p. 4943-4953, 2011.
- [70] A. G. Bondarev, S. A. Bondarev, N. E. Kostyleva, et V. I. Utkin, « Sliding modes in systems with asymptotic state observers », *Avtom. Telemekhanika*, n° 6, p. 5-11, 1985.
- [71] S. V. Emelyanov, S. K. Korovin, et L. V. Levantovskiy, « A drift algorithm in control of uncertain processes. », *PROB CONTROL INFO THEORY*, vol. 15, n° 6, p. 425-438, 1986.
- [72] L. V. Levantovsky, « Second order sliding algorithms: their realization », *Dyn. Heterog. Syst.*, p. 32-43, 1985.
- [73] B. Draženović, « The invariance conditions in variable structure systems », *Automatica*, vol. 5, n° 3, p. 287-295, 1969.
- [74] H. SIRA-RAMIREZ, « Differential geometric methods in variable-structure control », *Int. J. Control Int. J. Control*, vol. 48, n° 4, p. 1359-1390, 1988.
- [75] A. Isidori, *Nonlinear Control Systems: An Introduction*. Berlin Heidelberg: Springer-Verlag, 1985. doi: 10.1007/BFb0006368.
- [76] I. Kanellakopoulos, P. V. Kokotovic, et A. S. Morse, « A toolkit for nonlinear feedback design », *SCL Syst. Control Lett.*, vol. 18, n° 2, p. 83-92, 1992.

- [77] P. V. Kokotovic, « The joy of feedback: nonlinear and adaptive », *IEEE Control Syst IEEE Control Syst.*, vol. 12, n° 3, p. 7-17, 1992.
- [78] R. A. Freeman et P. V. Kokotovic, « A New Lyapunov Function for the Backstepping Design of Softer Robust Nonlinear Control Laws », *Automatica*, p. 307-312, 1992.
- [79] R. Freeman et P. V. Kokotovic, *Robust Nonlinear Control Design: State-Space and Lyapunov Techniques*. Birkhäuser Basel, 1996. doi: 10.1007/978-0-8176-4759-9.
- [80] M. Krstić, I. Kanellakopoulos, et P. V. Kokotović, « Adaptive nonlinear control without overparametrization », *Syst. Control Lett.*, vol. 19, n° 3, p. 177-185, 1992.
- [81] M. Krstic, I. Kanellakopoulos, et P. Kokotovic, « Nonlinear Design of Adaptive Controllers for Linear Systems », *IEEE Trans. Autom. CONTROL AC*, vol. 39, n° 4, p. 738, 1994.
- [82] M. Krstic, I. Kanellakopoulos, et P. V. Kokotovic, *Nonlinear and adaptive control design*. New York: John Wiley & Sons, 1995.
- [83] Z.-H. Li et M. Krstić, « Optimal design of adaptive tracking controllers for non-linear systems », *Automatica*, vol. 33, n° 8, p. 1459-1473, 1997.
- [84] C. Hua et X. Guan, « Adaptive control for chaotic systems », *Chaos Solitons Fractals*, vol. 22, n° 1, p. 55-60, 2004.
- [85] M. Horch, A. Boumediene, et L. Baghli, « Nonlinear Integral Backstepping Control for Induction Motor drive with Adaptive Speed Observer using Super Twisting Strategy. », *Electroteh. Electron. Autom.*, vol. 64, n° 1, 2016.
- [86] A. R. Benaskeur, « Aspects de l'application du backstepping adaptatif à la commande décentralisée des systèmes non linéaires. », la Faculté des Études Supérieures de l'université Laval, Québec, 2000.
- [87] M. Dorigo, « Optimization, learning and natural algorithms », *Ph Thesis Politec. Milano*, 1992.
- [88] A. E. Dor, « Improvement of particle swarm optimization algorithms: applications in image segmentation and electronics », PhD Thesis, Dissertation, University Paris-Est, 2012.
- [89] M. Semchedine, « Contribution à la segmentation d'images médicales par les algorithmes bio-inspirés », PhD Thesis, Université Ferhat Abbas Sétif 1, ALGERIE, 2018.
- [90] A. Colorni, M. Dorigo, et V. Maniezzo, « Distributed optimization by ant colonies », in *Proceedings of the first European conference on artificial life*, 1991, vol. 142, p. 134-142.
- [91] S. Kirkpatrick, C. D. Gelatt, et M. P. Vecchi, « Optimization by simulated annealing », *science*, vol. 220, n° 4598, p. 671-680, 1983.
- [92] R. L. Haupt et S. E. Haupt, *Practical genetic algorithms*. Hoboken (N.J.): Wiley-interscience, 2004.
- [93] Z.-L. Gaing, « A particle swarm optimization approach for optimum design of PID controller in AVR system », *IEEE Trans. Energy Convers.*, vol. 19, n° 2, p. 384-391, 2004.
- [94] M. Nasri, H. Nezamabadi-Pour, et M. Maghfoori, « A PSO-based optimum design of PID controller for a linear brushless DC motor », *World Acad. Sci. Eng. Technol.*, vol. 26, n° 40, p. 211-215, 2007.
- [95] S. M. GirirajKumar, D. Jayaraj, et A. R. Kishan, « PSO based tuning of a PID controller for a high performance drilling machine », *Int. J. Comput. Appl.*, vol. 1, n° 19, p. 12-18, 2010.
- [96] R. Bachir Bouiadjra, « Commande robuste des systèmes non linéaires », PhD Thesis, Paris-Est, Université d'Oran 1 Ahmed Ben Bela, ALGERIE, 2015.
- [97] I. H. Osman et G. Laporte, *Metaheuristics: A bibliography*. Springer, 1996.
- [98] G. B. Dantzig et P. Wolfe, « The decomposition algorithm for linear programs », *Econom. J. Econom. Soc.*, p. 767-778, 1961.
- [99] M. Avriel, *Nonlinear programming: analysis and methods*. NJ: Prentice-Hall, Englewood Cliffs, 2003.
- [100] D. P. Bertsekas, « Projected Newton methods for optimization problems with simple constraints », *SIAM J. Control Optim.*, vol. 20, n° 2, p. 221-246, 1982.
- [101] F. Glover, « Future paths for integer programming and links to artificial intelligence », *Comput. Oper. Res.*, vol. 13, n° 5, p. 533-549, 1986.
- [102] C. Blum et A. Roli, « Metaheuristics in combinatorial optimization: Overview and conceptual comparison », *ACM Comput. Surv. CSUR*, vol. 35, n° 3, p. 268-308, 2003.
- [103] C. Darwin, *On the origin of species*, [1st edition]. London: John Murray., 1859.

- [104] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press, 1975.
- [105] Y. Harrath, « Contribution à l'ordonnancement conjoint de la production et de la maintenance: Application au cas d'un Job Shop. », PhD Thesis, Université de Franche-Comté, 2003.
- [106] C.-C. Lin et Y.-J. Lee, « Stacking sequence optimization of laminated composite structures using genetic algorithm with local improvement », *Compos. Struct.*, vol. 63, n° 3-4, p. 339-345, 2004.
- [107] S. Bourazza, « Variantes d'algorithmes génétiques appliquées aux problèmes d'ordonnancement », PhD Thesis, Université du Havre, 2006.
- [108] K. Saidi et M. Allad, « Fuzzy controller parameters optimization by using genetic algorithm for the control of inverted pendulum », in *2015 3rd International Conference on Control, Engineering & Information Technology (CEIT)*, 2015, p. 1-6.
- [109] A. Poursamad et M. Montazeri, « Design of genetic-fuzzy control strategy for parallel hybrid electric vehicles », *Control Eng. Pract.*, vol. 16, n° 7, p. 861-873, 2008.
- [110] U. Maulik, S. Bandyopadhyay, et A. Mukhopadhyay, *Multiobjective genetic algorithms for clustering: applications in data mining and bioinformatics*. Springer Science & Business Media, 2011.
- [111] D. A. Coley, *An introduction to genetic algorithms for scientists and engineers*. World Scientific Publishing Company, 1999.
- [112] D. A. Coley, *An introduction to genetic algorithms for scientists and engineers*. Singapore: World Scientific, 2010.
- [113] E. Poirier, « Optimisation énergétique et entraînement sans capteur de position des machines à courant alternatif », PhD Thesis, Département du génie électrique, Faculté d'ingénierie, Université de Moncton, 2001.
- [114] S. Banerjee, A. Ghosh, et N. Rana, « An improved interleaved boost converter with PSO-based optimal type-III controller », *IEEE J. Emerg. Sel. Top. Power Electron.*, vol. 5, n° 1, p. 323-337, 2016.
- [115] Saidi K, Boumediene A, Saidi K, et Massoum S, « An optimal PSO-based sliding-mode control scheme for the robot manipulator », *Elektrotehniski VestnikElectrotechnical Rev.*, vol. 87, n° 1-2, p. 53-59, 2020.
- [116] C. W. Reynolds, « Flocks, herds and schools: A distributed behavioral model », in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, 1987, p. 25-34.
- [117] F. Heppner et U. Grenander, « A stochastic nonlinear model for coordinated bird flocks », *Ubiquity Chaos*, vol. 233, p. 238, 1990.
- [118] J. Kennedy, « Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance », in *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)*, 1999, vol. 3, p. 1931-1938.
- [119] Y. Cooren, « Perfectionnement d'un algorithme adaptatif d'Optimisation par Essaim Particulaire: application en génie médical et en électronique », PhD Thesis, Université Paris-Est, 2008.
- [120] J. Yu, Z. Wu, M. Wang, et M. Tan, « CPG network optimization for a biomimetic robotic fish via PSO », *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, n° 9, p. 1962-1968, 2015.
- [121] Y. Shi et R. C. Eberhart, « Parameter selection in particle swarm optimization », in *International conference on evolutionary programming*, 1998, p. 591-600.
- [122] R. C. Eberhart et Y. Shi, « Comparing inertia weights and constriction factors in particle swarm optimization », in *Proceedings of the 2000 congress on evolutionary computation. CEC00 (Cat. No. 00TH8512)*, 2000, vol. 1, p. 84-88.
- [123] H. Berghuis et H. Nijmeijer, « Global regulation of robots using only position measurements », *Syst. Control Lett.*, vol. 21, n° 4, p. 289-293, oct. 1993, doi: 10.1016/0167-6911(93)90071-D.
- [124] R. Campa, R. Kelly, et V. Santibanez, « Windows-based real-time control of direct-drive mechanisms: platform description and experiments », *Mechatronics*, vol. 14, n° 9, p. 1021-1036, 2004.
- [125] A. AMEUR, « Commande sans capteur de vitesse par DTC d'une machine synchrone à aimants permanents dotée d'un observateur d'ordre complet à modes glissants », Magister, Université de Batna 2, 2005.

- [126] D. Lahouel, « Commande Non Linéaire Adaptative D'une Machine Synchrone à Aimants Permanents », Magister, Université de Batna 2, 2009.
- [127] M. A. Ferreira de Sousa, « Contribution à la conception de lois de commande à caractère robuste pour une machine synchrone entraînant une charge élastique à inertie variable », PhD Thesis, Institut National Polytechnique de Toulouse, France, 2007.
- [128] P. Pillay et R. Krishnan, « Modeling of permanent magnet motor drives », *IEEE Trans. Ind. Electron.*, vol. 35, n° 4, p. 537-541, 1988.
- [129] A. Kaddouri, « Etude d'une commande non-linéaire adaptative d'une machine synchrone à aimants permanents », Université Laval, QUÉBEC, Canada, 2000.
- [130] R. Lateb, « Modélisation des machines asynchrones et synchrones a aimants avec prise en compte des harmoniques d'espace et de temps: application à la propulsion marine par POD », PhD Thesis, Institut National Polytechnique de Lorraine-INPL, 2006.
- [131] B. Hacéne, « Contribution à la Commande des Machines Synchrones a Aimants Permanents », PhD Thesis, Thèse de Doctorat, Institut National Polytechnique de Lorraine, 1995.
- [132] N. CHOUG, « Etude de la robustesse des contrôleurs flous d'une machine synchrone à aimants permanents avec pilotage vectoriel », PhD Thesis, Université de Batna 2, 2011.
- [133] M. Ezzat, « Commande non linéaire sans capteur de la machine synchrone à aimants permanents », PhD Thesis, Ecole Centrale de Nantes (ECN), France, 2011.
- [134] L. S. Mey et K. S. Kalyan, « Analysis of Permanent Magnet Synchronous Motor under Different Operating Condition Using Vector Controlled in MATLAB », présenté à 2012 IEEE Electrical Power and Energy Conference, 2012.
- [135] R. V. Carrillo-Serrano, V. M. Hernández-Guzmán, et V. Santibáñez, « PD control with feedforward compensation for rigid robots actuated by brushless DC motors », *Robotica*, vol. 29, n° 4, p. 507-514, 2011.
- [136] L. V. Levantovsky, « Second order sliding algorithms: Their realization, dynamics of heterogeneous systems [J] », *Inst. Syst. Stud.*, vol. 12, n° 7, 1985.
- [137] S. V. Emel'Yanov, S. K. Korovin, et L. V. Levantovskii, « Higher-order sliding modes in binary control systems », in *Soviet Physics Doklady*, 1986, vol. 31, p. 291.
- [138] I. Deghboudj, « Commande des systemes non lineaires par mode glissant d'ordre superieur », mémoire de Magister, UNIVERSITE CONSTANTINE 1, Constantine, Algerie, 2013.
- [139] A. Levant, « Sliding order and sliding accuracy in sliding mode control », *Int. J. Control*, vol. 58, n° 6, p. 1247-1263, 1993.
- [140] A. BOUDISSA, « Commande Non Linéaire en Impédance d'un Robot Manipulateur Industriel », mémoire de Magister, Université de Batna 2, 2016.
- [141] N. Bounasla, « Commande par mode de glissement d'ordre supérieur de la machine synchrone à aimants permanents », mémoire de Magister, Université Ferhat ABBAS Sétif -1, Sétif, ALGERIE, 2018.
- [142] A. Bouyekni, R. Taleb, Z. Boudjema, et M. Moumna, « Commande par mode glissant d'ordre 2 pour une capture maximale d'énergie d'une turbine éolienne », *J. Renew. Energ.*, vol. 21, n° 1, p. 19-26, 2018.
- [143] H. K. Khalil, *Nonlinear Systems*. Prentice Hall, 2002.
- [144] L. Kruise, « Modelling and control of a flexible beam and robot arm », PhD Thesis, University of Twente, Holland, 1990.