

UNIVERSITE DE SAAD DAHLEB DE BLIDA

Faculté des Sciences de l'Ingénieur

Département de Génie Mécanique

MEMOIRE DE MAGISTERE

Spécialité : Construction Mécanique

PROGRAMMATION DES ROBOTS INDUSTRIELS ET APPLICATION SUR LE ROBOT MANIPULATEUR

[ALG.-M. O.-1] (*)

Par

BENMISRA Abdelkader

Devant le jury composé de

M. OUALI	Professeur, U.S.D. de Blida	Président
A. NOUR	Professeur, U.M.B.de Boumerdès	Examineur
A. BELAIDI	Maître de conférences, U.M.B.de Boumerdès	Examineur
A. ALLALI	Maître de conférences, U.S.D.de Blida	Rapporteur
A. BRAHIMI	Chargé de cours, U.S.D. de Blida	Co-Rapporteur

Blida, le 29 Octobre 2007

(*) : MODELISATION DES ROBOTS INDUSTRIELS ET APPLICATION SUR LE ROBOT MANIPULATEUR

[ALGERIE-MACHINES OUTILS- 1].

REMERCIEMENTS

Louange à DIEU Seigneur des mondes .«Notre Seigneur ! Nous avons entendu un crieur criant pour nous appeler à foi, " Croyez en notre Seigneur !", et nous avons cru; notre Seigneur ! Pardonne nous nos péchés ! Efface nos mauvaises actions ! Rappelle-nous à toi avec ceux qui sont bons 193 . Notre Seigneur ! Accorde nous ce que tu nous as promis par la voix de tes Prophètes; ne nous affiche pas le jour de la Résurrection ;-tu ne manques jamais à ta promesse- 194 . Leur Seigneur les a exaucés : "Je ne laisse pas perdre l'action de celui qui, parmi vous, homme ou femme, agit bien. Vous dépendez les uns des autres. J'effacerai les mauvaises actions de ceux qui ont émigré de ceux qui ont été expulsés de leurs maisons de ceux qui on souffert dans mon chemin de ceux qui ont combattu et qui ont été tués je les ferai certainement entrer dans des Jardins où coulent les ruisseaux; ce sera une récompense de la part de DIEU". DIEU ! La plus belle des récompenses se trouve auprès de lui ! 195 ».

LA FAMILLE DE IMRAN 193, 194,195

DEDICACES

Je dédie mon travail.

À mes très chers parents.

À mes frères et mes sœurs.

À toute la famille.

À tous mes amis.

À tous les Enseignants de Département de Mécanique.

ملخص

العمل المقدم من خلال هذه المذكرة هو مساهمة في ميدان النمذجة للروبوتات، و اقتراح بنك معطيات، استخرجت النماذج الهندسية، الحركية و الديناميكية المباشرة والعكسية، اقترحت طريقة لاجرونج دالوميرت مع استعمال معاملات دونافي هاتنبرج، كما أثرت انتباه الباحثين لدراسة الأخطاء السكونية والديناميكية، تحصلت على براءة اختراع للروبوت النموذج لدى المعهد الوطني الجزائري للملكية الصناعية برقم 060205.

طرحتم نموذج أولي، ثم درست النموذج وتمكنت من وضع الأبعاد والقياسات بدقة لهذه الآلية غير أن اختيار هذه البنية كان تحت تأثير بعض العوامل كالفعالية والإمكانات المتوفرة في مخابر البحث العلمي، استخرجت مختلف النماذج في ميدان البرمجة و المحاكاة لاحظت استعمال مختلف المقاربات من خلال دراسة الأعمال المنجزة في الجامعات، الإشكالية العامة لعملي هي اقتراح نظام حساب رقمي على شكل بنك معطيات بهدف تجميع مقاربات النمذجة الجديدة المستعملة في الميدان الصناعي.

ABSTRACT

The work presented in this memory, has the aim of contributing a share in the domain of robotic modelling.

I proposed a data bank, and determined the models geometrical, kinematics and dynamic (direct and opposite).

I applied the formalism of Lagrange Dalember; and used the system of location of Denavit and Hartenberg, as well as a contributing on the study of the errors static and dynamic.

My standard robot is patented (I.N.A.P.I.) under the number 060205, and proposed a prototype, and then, I defined the design of my manipulator, I can dimension, rigorously the robot.

In other, the choice of this structure is made under constraints of feasibility and the existing means at the scientific research laboratories.

A variety of model were elaborate; in the field of the programming and the simulation of the manipulators, I noted that the various approaches of the mechanical command of the poly articulated systems were elaborate through the study of work completed in various Universities.

The general problematical of my memory is to propose a data bank end to count the new approaches of design in the industry.

RÉSUMÉ

Le travail présenté dans ce mémoire, a pour objet d'apporter une contribution dans le domaine de la modélisation.

J'ai proposé une banque des données, et déterminé les modèles géométriques, cinématiques et dynamiques (direct et inverse).

J'ai appliqué le formalisme de Lagrange D'Alembert, et utilisé le système de repérage de Denavit et Hartenberg, ainsi qu'une réflexion sur l'étude des erreurs statique et dynamique. Mon robot type est breveté (I.N.A.P.I.) sous le numéro 060205.

J'ai proposé un prototype, et puis définie la conception de mon manipulateur, j'ai su dimensionner, rigoureusement le robot.

En outre, le choix de ma structure est fait sous contraintes de la faisabilité et les moyens existants aux laboratoires de recherches scientifiques.

Les différentes modélisations ont été élaborées ; dans le domaine de la programmation et la simulation des manipulateurs, à travers l'étude de travaux réalisés dans diverses universités, j'ai constaté que les différentes approches de la commande des systèmes mécanique poly articulés ont été élaborées.

La problématique générale de mon travail est de proposer un calculateur sous forme de banque des données à fin de recenser les nouvelles approches de conception dans le domaine industriel.

TABLE DES MATIERES

REMERCIEMENTS	
RÉSUMÉS	
TABLE DES MATIÈRES	
LISTES DES ILLUSTRATIONS GRAPHIQUES ET TABLEAUX.....	
INTRODUCTION	9
1. GENERALITES	13
1.1. Historique	13
1.2. Définitions générales	13
1.3. Structure et morphologie des robots	15
1.4. Classification des robots	17
1.5. Les porteurs (bras)	18
1.6. Caractéristiques générales des robots	21
1.7. Conclusion	21
2. MODELISATIONS GEOMETRIQUE CINEMATIQUE ET DYNAMIQUE (DIRECTE ET INVERS) DES SYSTEMES MECANIQUE POLY ARTICULE AUX ELEMENTS RIGIDES	23
2.1. Modèle géométrique	23
2.2. Les variables articulaires	24
2.3. Le système de paramétrage de Denavit-Hartenberg	24
2.4. Modèle géométrique inverse	28
2.5. Résolvabilité des S.M.P.A	30
2.6. Calcul du modèle inverse	31
2.7. Etude cinématique	31
2.8. Détermination des matrices de changements de repères.....	33
2.9. Modélisation dynamique.....	35
2.10. Méthode d'obtention du modèle dynamique	35
2.11. L'utilisation de la méthode de Runge-Kutta	38
2.12. Conclusion	39
3. LES CAUSES STRUCTURELLES.....	40
3.1. Modèles dynamique des S.M.P.A. aux joints déformable.....	40
3.2. Les erreurs statique et dynamique.....	44
4. APPLICATIONS.....	58
4.1 Introduction	58
4.2 Exemple d'application et de validation	58
4.3 Fichier de résultats	64
4.4 Interprétation des résultats	66
CONCLUSION GENERALE.....	68
APPENDICES	
REFERENCES	

Listes des figures, tableaux, programmes, organigrammes, algorithmes appendices, symboles, abréviations, et acronymes

- Figure 1.1 : Structure à chaînes simples ouvertes [21, 38,71]
Figure 1.2 : Structure à chaînes, arborescentes [21, 38,71]
Figure 1.3 : Structure à chaînes complexes [21, 38,71]
Figure 1.4 : Les douze porteurs de robots [144]
Figure 1.5 : La composition du robot industriel [7][144].
Figure 1.6 : Robots Manipulateurs Industriel à Poste Fixe (d'après I.S.O 8373) [14]
Figure 2.1 : Structure arborescente [12,21]
Figure 2.2 : Système de coordonnées et paramètres de D.-H. [12,21, 27, 29, 30, 31,71]
Figure 2.3 : Représentation des paramètres de D.-H. [, 8, 10,12, 35, 38,85]
Figure 2.4. : Le robot R.P.[10,12,14,21,38]
Figure 2.4 : Repère avec deux articulations. [38]
Figure 2.6 : La structure R.P. [38]
Figure 2.7 : Représentation d'un poignet. [10, 12,21]
Figure 3.1 : L'exactitude de positionnement statique d'un manipulateur. [10, 55, 65,38]
Figure 3.2 : La répétitivité de positionnement statique d'un manipulateur. [10, 55, 65,38]
Figure 3.3 : Exactitude de trajectoire [10, 55, 65,70]
Figure 3.4 : Flexibilité d'un segment du manipulateur [55]
Figure 3.5 : La transformation définissant le repère $\{R_j\}$ dans le repère $\{R_{j-1}\}$ [, 7, 10,12]
Figure 3.6 : Compensation des erreurs des déformations [12, 10,55]
Figure 3.7 : Organigramme d'un modèle d'évaluation et de compensations des erreurs [55]
Figure 4.1 : Position 1, (l'état initial).
Figure 4.2 : Phase de fraisage (fraiseuse verticale).
Figure 4.3 : Phase de fraisage (fraiseuse universelle).
Figure 4.4 : Phase de perçage.
Figure 4.5 : Phase d'alésage.
Figure 4.6 : L'interface de l'utilisateur pour l'état initial
Figure 4.7 : L'interface de l'utilisateur pour la fraiseuse verticale
Figure 4.8 : L'interface de l'utilisateur pour la fraiseuse universelle
Figure 4.9 : L'interface de l'utilisateur pour la perceuse
Figure 4.10 : L'interface de l'utilisateur pour l'aléseuse
Figure 4.11: Positions
Figure 4.12: Vitesses
Figure 4.13: Accélération
Figure 4.14: Forces, couples
Figure A.1 : Vue de la station de fraisage alésage [21]
Figure A.2 : Schéma cinématique du robot industriel [21]
Figure A.3 : Vue en élévation en coupe d'un robot manipulateur [21]
Figure A.4 : Vue en coupe du module de levage [21, 142,144]
Figure A.5 : Module de rotation à deux cylindres [21, 142,144]
Figure A.6 : Module de déplacement longitudinal [21, 142,144]
Figure A.7 : Module de l'organe terminale [21, 142,144]
Figure L.1 : Le problème de type 2 [27]
Figure L.2 : Le problème de type 3 [27]

Figure L.3 : Schéma représentatif d'une trajectoire dans le plan [27]
 Figure L.4 : Variation de l'angle φ en fonction de α . [27]
 Figure L.5 : Génération du deuxième tronçon [27]
 Figure L.6 : Génération d'une trajectoire tridimensionnelle [27]
 Figure L.7 : Position articulaire [27,28]
 Figure L.8 : Vitesse angulaire [27,28]
 Figure L.9 : Accélération articulaire [27,28]
 Figure R.1 : Convergence d'une technique de Monte-Carlo [71]
 Figure R.2 : Meilleures solutions trouvées [71]
 Figure R.3 : Architecture du système N.N.S. [71]
 Tableau 1.1 : Représentation des liaisons mécaniques entre deux corps [14]
 Tableau 2.1 : Les paramètres géométriques d'un robot type. [21, 140, 141, 142,144]
 Tableau R.1 : Les principales formes d'usure [138]
 Tableau R.2 : Liste des systèmes de CAO ayant des applications en robotique [64]
 Tableau R.3 : Avantages des programmations par CAO [64]
 Programme : (Appendice B) : Le programme de l'espace de travail. [21, 71,143]
 Programme : (Appendice C) : Le programme principal. [21, 71,143]
 Programme : (Appendice D) : Le programme de synchronisation.[21,71,143]
 Programme : (Appendice E) : Programme de calcul des forces des moments.[21,71,143]
 Programme : (Appendice F) : Programme de calcul des forces actives.[21,71,143]
 Programme : (Appendice G) : Programme de calcul des forces actives.[21,71,143]
 Programme : (Appendice H) : Programme de modèle géométrique direct.[21,71,143]
 Programme : (Appendice R) : Programmation et contrôle d'exécution [64,104]
 Organigramme :(Figure 3.7.) : Organigramme d'un modèle d'évaluation [55]
 Organigramme :(Appendice I) : Organigramme général
 Organigramme :(Appendice R) : Organigramme d'un processus d'optimisation [71]
 Algorithme 2.1.: Algorithme simplifié du programme [143]
 Algorithme R.1. :(Appendice R): Algorithme de la technique de Monte-Carlo [71]
 Algorithme R.2. :(Appendice R): Algorithme de la technique de Hill-Climbing [71]
 Algorithme R.3. :(Appendice R): Algorithme, Monte-Carlo avec réduction d'intervalle [71]
 Algorithme R.4. :(Appendice R): Algorithme de la recherche Tabou [71]
 Algorithme R.5. :(Appendice R): Algorithme Génétique [71]
 Algorithme R.6. : [Processus](Appendice R): Processus d'assemblage[64,104]
 Appendice A : Le brevet d'invention [I.N.A.P.I.]
 Appendice B : Le programme de l'espace de travail
 Appendice C : Le programme principal
 Appendice D : Le programme de synchronisation
 Appendice E : Programme de calcul des forces des moments articulaires
 Appendice F : Programme de calcul des forces actives
 Appendice G. : Programme de simulation
 Appendice H. : Programme de modèle géométrique direct
 Appendice I : Organigramme général
 Appendice J : Le système d'équation non linéaire [21,38 ,71]
 Appendice K : Dérivation de la matrice de transformation homogène [21,38]
 Appendice L : Exemple de trajectoires (Génération de trajectoires)[27,28]
 Appendice M: Dérivée de la matrice de transformation homogène [21, 38,71]
 Appendice N : Expression de l'énergie cinétique [21, 38,71]
 Appendice O : Expression des coefficients dynamiques [21, 38,71]
 Appendice P : Propriété des coefficients dynamiques [21,71]
 Appendice Q : Formalisme de NEWTON-EULER [71]

INTRODUCTION

Jusqu'à présent, les robots industriels sont surtout employés pour servir que de générateur de positions et d'orientations d'un outil dans l'espace. Pour cela, ils ne sont modélisés que par trois modèles: **Le modèle géométrique**, **Le modèle cinématique** et **Le modèle dynamique**. Un robot industriel peut-être vu comme un système mécanique poly articulé [44]. Pour pouvoir faire exécuter une action où une tâche au robot manipulateur, il est nécessaire de modéliser ce système multi variable pour positionner les différents axes par contre un modèle de déformation est une nécessité.

Le but de la modélisation dans la robotique est de fournir les équations mathématiques qui permettent de programmer, commander et exécuter les tâches désirées mais, j'ai constaté la nécessité de fournir les consignes des positions qui permettent d'exécuter les tâches désirées ; de plus, il faut disposer de langages et d'environnement de programmation spécifique ce qui complique le travail des programmeurs non initiés en robotique. Admettant que la recherche scientifique dans le domaine des banques des données sur les paramètres de la mécanique articulée est l'un des outils de développement dans la plus part des pays. Par contre cette branche n'existe pas en Algérie ; la problématique générale consiste à élaborer une banque des données sur les paramètres de la mécanique articulée et des articulateurs type pour pouvoir faire un choix judicieux des paramètres et des régimes de fonctionnement d'un manipulateur automatique. La recherche bibliographique si dessous donnerait quelque rapprochement ; B.MADANI [1] : a contribué à la conception mécanique des robots. La référence [2] est un cours de JAVA, les auteurs [3, 4] : ont pris en considération les contraintes de l'espace dans le quel agit le robot , [5] : ont insisté sur la nécessité de disposition d'une interface d'apprentissage ,N. SEGUY [6] : a spécifié de plus le type d'interpolation que le robot utilise ,la vitesse le long de la trajectoire,et le temps de décélération sur le point d'arrivée,Y. KOREN : [7] a fait appel à une modélisation de l'environnement du robot sous forme symbolique . P.COIFFET [8] ; L.PERROTTE [9] et M.GIORDANO, J.LOTTIN [10] : ont donné la modélisation, la

commande, la description et le fonctionnement des robots industriels sous forme de cours. J.GRANT, F.BARA [11] : ont introduit à la robotique, W.KHALIL [12] : a basé sur la génération de trajectoires. J.DENAVIT, R.S. HARTENBERG [13] : ont communiqué «Kinématic rotation for lower pair mechanisms based on matrices » en1955.

P.COIFFET [14, 16]: a reflété à l'intention des roboticiens l'ambiguïté de définition du mot robot. L. KHEMICI [15] a optimisé des trajectoires coordonnées, V.DUPOURQUE [18,143]: A constaté que dans le cas ou le dispositif utilisé est une boite à boutons, Les trajectoires restituées peuvent être classées. Leur réalisation nécessite un générateur de trajectoires.

UNIMATION® : [19] Le langage VAL est à la fois un langage et un système de commande spécialement conçu en 1975 pour les manipulateurs de la société UNIMATION®.Du point de vue du langage VAL est proche du BASIC classique.Le système d'exploitation permet de créer deux types de fichiers. L'un contient le programme proprement dit, l'autre, les valeurs des données de type emplacement.C.MELIN H.HAMDI : [20] Un langage de programmation de niveau objet (programmation mixte) est une extension d'un langage structuré de programmation explicite. On peut le considérer comme étant une interface pour l'utilisateur. M.M.HATTALI [21] : a modélisé un robot de soudage, E. DOMBRE [22] :Les modèles des objets peuvent également être créés par un moyen graphique interfacique (station graphique et systèmes de conception assistée par ordinateur C.A.O 3D; R. P.POPPLESTONE ,P. AMBLER ,I. BELLOS [23] : Un modèle géométrique du manipulateur utilisé permet également d'engendrer un programme de niveau explicite ou directement un programme de niveau articulaire, dans lequel l'accessibilité de toutes les destinations du robot a été testé. Les références [24 ,25 ,26] : des contributions à l'étude des manipulateurs.

J. J. CRAIG , R. J. SCHILING [27 , 28] : Une approche analytique a été utilisée pour le robot de type série, elle consiste à éliminer à chaque étape une des coordonnées généralisées (articulaires) par la multiplication de la matrice de transfert finale T_6^0 et les matrices de transformation intermédiaires.

Les références [29, 30, 31, 32,33] : Modélisations, aspects fondamentaux de la robotique; P. BORREL: [34] a proposé une méthode originale de détermination de volume de travail des robots, basée sur l'utilisation de leurs modèles cinématiques. Cette méthode étudie ,par décomposition du volume de travail, un aspect représentant le domaine accessible par le robot pour une configuration articulaire.

ANGELES [35]: Il est possible d'homogénéiser les dimensions par la " longueur caractéristique " du robot étudié.

M. MINOUX : [36] la résolution d'un problème avec contraintes peut se ramener, moyennant l'introduction de termes de pénalités ou bien d'un lagrangien augmenté, à un problème d'optimisation sans contraintes.

Z. MICHALWICZ : [37] : Une méthode nouvelle a été proposée récemment pour résoudre de manière simultanée la synthèse dimensionnelle. Les coordonnées généralisées (articulaires) par la multiplication de la matrice de transfert final T_6^0 les matrices de transformation intermédiaires .A PRUSKI [38]: En connaissant les contraintes d'orientation et de position du point terminal, il est possible de déduire la position de l'extrémité du porteur. Les expressions sont simplifiées et la solution du système devient plus aisée. De [39] à [54] : Définitions générales de la robotique.

M.HADDAD : [55] a présenté une nouvelle méthode de description géométrique des bras manipulateurs, qui répond pleinement à l'objectif fixé, maintien du caractère systématique des calculs mis en œuvre dans la formulation récurrente des modèles géométrique, cinématique et dynamique. Il a aussi présenté les nouvelles formulations des modèles géométriques, cinématique de second ordre. Comparativement avec le paramétrage de Denavit Hartenberg. De[56]à[63] : Définitions générales ;J. D. BOISSONNAT , B. FAYERJON , J. P. MERLET : [64] ont proposé des solutions permettant d'une part de synthétiser automatiquement des programmes de mouvements définis, et d'autre part de procéder à une phase vérification / correction sur les plans. De [65] à [70] : Définitions générales dans la robotique, A. BOUGUERRA[71] a contribué à la planification optimale des robots coopérants .De [72]à[98] : communications, revus, et journaux sur la robotique , ARTOBOLEVSKI[99] : Théorie détaillée sur les mécanismes et les machines , © Editions Mir, Moscou, 1977;De[100]à[130] : Articles et ouvrages sur la robotique.

DONNEES TECHNIQUES [131,132]: « Machines outils, perceuses radiales, TGL 42 1995 », d'origines République Démocratique Allemande, Août 1985 ; « Appareil de fraisage vertical Ap. FS 250/2 », Caractéristiques techniques « DZFG 200 x 500 » ; Atelier de fabrication mécanique, Département de Génie Mécanique, Université Saad-Dahleb de Blida; [133] : Etude dynamique et avant projet de conception d'un Robot de soudage, [134]: Programmation d'un robot manipulateur des machines-outils, [135]: Etude et conception d'un Robot manipulateur.

[136]: Communication aux JMA 2000 sur un model mathématique d'un robot de soudage par résistance par points, [137]: Communication aux NCMES'07 sur un model mathématique d'un système mécanique articulé, [138] : Brevet d'invention sur un robot manipulateur des machines outils.

A.ALLALI, D.TOUMI, A.BRAHIMI, M.M.HATTALI [139]: Brevet d'invention, sur un robot de soudage par résistance ; R. CHATILA [140] : mobile robot navigation space modeling and decisional processes : Robotics Research ; E. LOPEZ MELLADO [141] : Le control d'exécution dans les cellules flexibles d'assemblages ; R.S. SMITH, M. GINI [142] : Robot tracking and control issues in an intelligent error recovery system .

V. DUPOURQUE [18,143] : Architecture matérielles des contrôleurs de robots Techniques de la robotique ; D.A. BOURNE, M.S. FOX [145] : Autonomous manufacturing automating the job- shop ; A.R. SANDERSON , G. PERRY [146] : Sensor-based robotic assembly systems; research and application in electronic manufacturing ; R. CASSINIS [147] : Resource allocation in industrial multirobot systems ; A. CORNET, J.-P. DEVILLE [148] : Physique et ingénierie des surfaces ; D. DRIBINE [149] : Modélisation cinématique et dynamique de bras manipulateur .

Mon mémoire est composé de quatre chapitres, **le premier chapitre** traite une recherche historique de la robotique, une description des différents types de robots et le vocabulaire utilisé dans la robotique. Ainsi **qu'une définition détaillée** des robots manipulateurs.

Le deuxième chapitre décrit les modélisations géométriques cinématique et dynamique direct et inverse en utilisant le système de paramétrage de Denavit Hartenberg, le model cinématique consiste à établir la matrice jacobéenne associée au model géométrique, l'approche utilisée dans la formulation du problème est le formalisme de Lagrange.

Le troisième chapitre traite les causes structurelles telles que la modélisation dynamique des systèmes mécaniques poly articulés à joints déformables. De même les erreurs statique et dynamique. Sous forme d'une étude de contribution au recensement du problème de la plasticité dont le but de refléter à l'attention des chercheurs un problème crucial dans le domaine de la robotique industrielle, qui nécessite une solution adéquate afin de le résoudre. Enfin **Le quatrième chapitre** contient l'application sur mon robot manipulateur je l'ai conçu et déposé a l'I.N.A.P.I un Brevet d'Invention ,Un modèle-type a été proposé, une banque des données sur les paramètres de la mécanique articulée et des articulateurs types est proposée. Un programme de simulation a été proposé qui nécessite des améliorations dans l'avenir pour les futurs Masters. Une station d'usinage dont les données pratiques connus a été proposée par l'S.N.V.I.

CHAPITRE 1 GENERALITES

1.1 Historique : [7, 55, 56] :

Le terme robot vient du Tchèque, il signifie travail forcé le mot sous son acception moderne, fut introduit par l'auteur tchèque Karel Capek dans son ouvrage R.U.R .qui date de 1921 en anglais Rossum's Universel Robots.

Le concept de Robot industriel fut breveté en 1954, par G.C. Devol (Brevet U.S. N°=2988237) ; ce brevet décrit la réalisation d'un bras mécanique asservi, capable d'effectuer des tâches du caractère industriel.

1.1 Définitions générales: [14, 55, 56, 57, 58, 59] :

Le **dictionnaire Robert** propose comme définition : Mécanisme automatique pouvant se substituer à l'homme pour effectuer certaines opérations, et capable d'en modifier de lui même le cycle et d'exercer un certain choix, (par détection photoélectrique des "cerveaux" électroniques de servomoteurs etc.).

La J.I.R.A. (Japan industrial Robot Association), quand à elle met plus l'accent sur la notion d'automorphisme. En donnant comme définition (1980), le dispositif versatile et flexible offrant des fonctions de déplacement similaires à celle des membres humains ou dont les fonctions de déplacement sont commandés par ses capteurs et ses moyens de reconnaissance.

La R.I.A. (Robotic Industries Association) aux Etats Unis ,insiste sur l'utilisation en proposant (1979) "(manipulateur multifonction reprogrammable conçu pour déplacer des matériaux, des pièces, des outils ou des dispositifs spécialisés de déplacements variables et programmables pour accomplir diverses tâches)".

L'A.F.R.I. (Association Française de Robotique Industrielle) a de son coté renoncé à donner une définition générale. Pour se focaliser sur la définition de deux grandes classes (définition enregistrée par **L'A.F.N.O.R NF.E61-100**) on y trouve ainsi définis :

"**Le manipulateur** mécanisme généralement composé d'éléments en série articulés ou coulissants l'un par rapport à l'autre, dont le but est la saisie et le déplacement d'objets suivant plusieurs degrés de liberté. Il est multifonctionnel et peut-être commandé directement par un opérateur humain ou par tout système logique (système à cames, logique pneumatique, logique électrique câblée ou programmée), et **le Robot industriel** : "**Manipulateur automatique**, asservi en position, reprogrammable, polyvalent, capable de positionner et d'orienter des matériaux pièces, outils ou dispositif spécialisé au cours de mouvement variables et programmé pour l'exécution de tâches variées, il se présente souvent sous la forme d'un ou plusieurs bras se terminant par un poignet.

Son unité de commande utilise, notamment, un dispositif de mémoire et éventuellement de perception de l'environnement.

Ces machines polyvalentes sont généralement étudiées pour effectuer la même fonction de façon cyclique et peuvent être adaptées à d'autres fonctions sans modification permanente du matériel".

1.2.1 Mobilité du robot [12, 8]:

C'est le nombre de variables indépendantes qui définissent la position du robot à un instant donné, la mobilité est également le nombre d'actionneurs du robot, c'est ce qu'on appelle le nombre d'axes de chaque actionneur commandant un axe, (m) est un nombre entier (en général $3 \leq m \leq 7$).

1.2.2 Degré de la tâche Dr.[8]:

C'est le nombre de paramètres indépendants qui permettent de définir la tâche c'est à dire la situation possible de l'organe terminal (position et orientation).

1.2.3 Espace généralisé [15, 32, 33]:

En général la configuration de la $i^{\text{ème}}$ articulation d'un robot manipulateur ne possède qu'un degré de liberté correspondant à un mouvement de translation rectiligne ou de rotation entre deux corps successifs des robots, situés sur une chaîne qui joint la base de l'organe terminal. A chacun de ces mouvements est associé une variable q_i ou $i = 1, 2, \dots, n$ dont l'ensemble constitue le vecteur $q = [q_1, \dots, q_n]^T$ des coordonnées articulaires (généralisées) qui est un élément de l'ensemble de configuration du robot.

1.2.4 Espace opérationnel (géométrique):

Afin de caractériser et de mesurer la position de l'organe terminal, on définit un espace appelé espace opérationnel qui a pour référence un repère (R_{op}) orthonormé (o_1, x_1, y_1, z_1) lié généralement à la base du robot manipulateur, de même il est nécessaire d'introduire un autre repère (R_n) orthonormé (o_n, x_n, y_n, z_n) lié à l'organe terminal afin de mieux définir la situation (position, orientation de l'organe terminal). Les coordonnées associées sont donc appelées coordonnées opérationnelles.

Du point de vue théorique, six coordonnées opérationnelles suffisent dans le cas général pour définir la situation du repère (R_n) par rapport au repère (R_{op}): Dont trois définissent la position de l'organe terminal (O_n) et trois définissent son orientation.

1.2.5 Redondance [12, 32, 33]:

Un robot est redondant lorsque le nombre de degrés de liberté de l'organe terminal est inférieur au nombre de degrés de liberté de l'espace articulaire (nombre d'articulations motorisées). Cette propriété permet d'augmenter le volume du domaine accessible et de préserver les capacités de déplacement de l'organe terminal en présences d'obstacles.

1-2-6 Configuration singulière [12]:

Pour tous les robots, qu'ils soient redondants ou non, il se peut que dans certaines configurations dites singulières, le nombre de degrés de liberté de l'organe terminal soit inférieur à la dimension de l'espace opérationnel.

La configuration singulière ou singularités se traduisent, physiquement, par la nullité du déterminant de la matrice Jacobéenne.

1-3 Structure et morphologie des robots: [12, 21, 55, 71]:

La partie mécanique d'un robot est généralement constituée de deux sous ensembles distincts.

Un (ou plusieurs) organe terminal considéré comme une interface permettant au robot d'interagir avec son environnement.

Une structure mécanique articulée qui sert à amener l'organe terminal dans une position et orientation désirées cette structure est composée du : bras, base, poignet coupleur et axe mécanique. Les robots peuvent être de:

- Structure à chaînes simples ouvertes figure 1.1.
- Structure à chaînes arborescentes figure 1.2.

-Structure à chaînes complexes figure 1.3.

Terminologie :

C_0 : Corps de référence.

C_1, C_2, C_3, C_4, C_5 : Corps.

O.T. : Organe terminal.

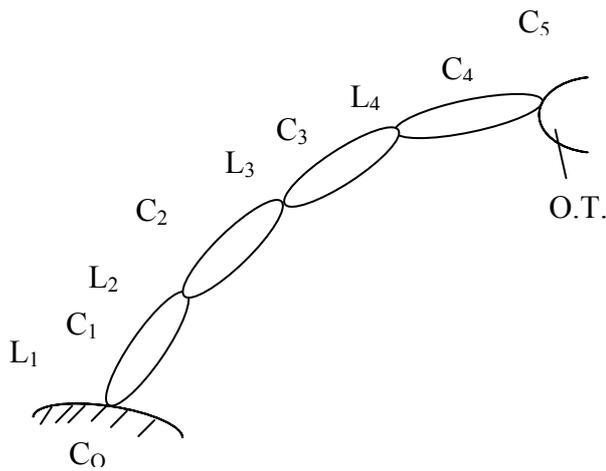


Figure 1.1 : Structure à Chaînes Simples Ouvertes [21, 71].

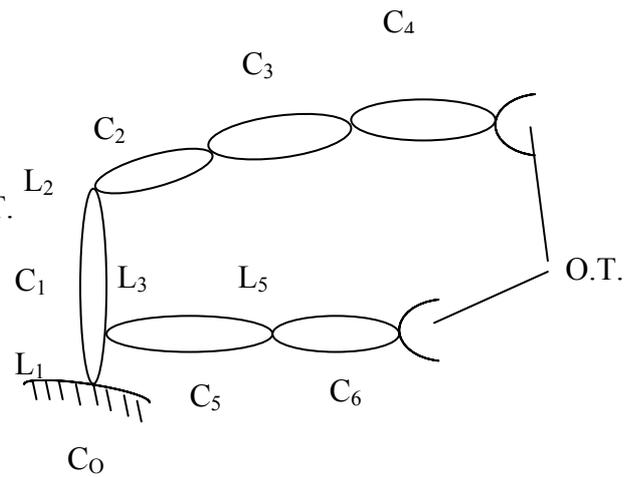


Figure 1.2 : Structure à Chaînes Arborescentes [21, 71].

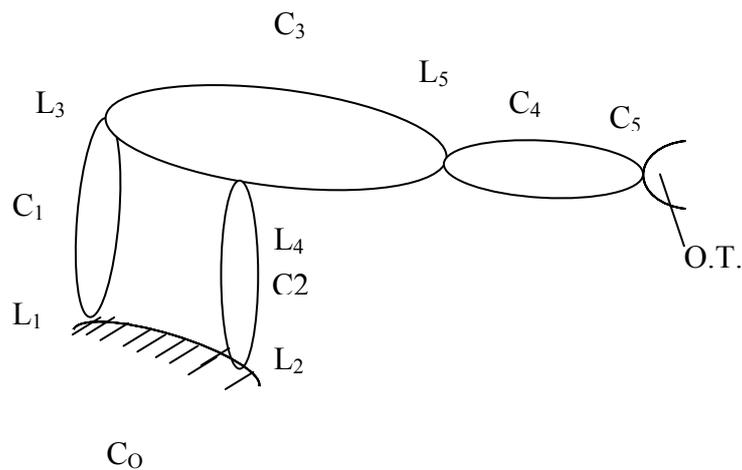


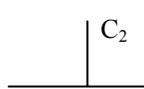
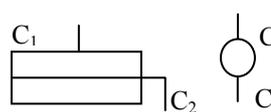
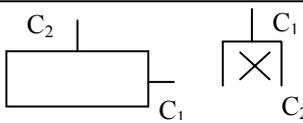
Figure 1.3 : Structure à Chaînes Complexes [21, 71].

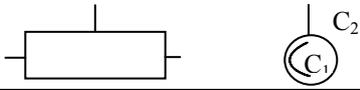
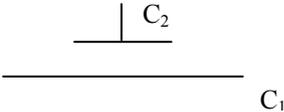
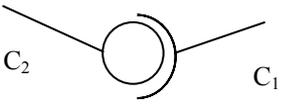
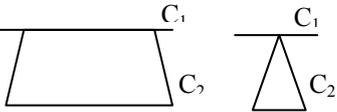
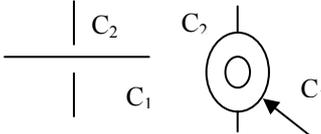
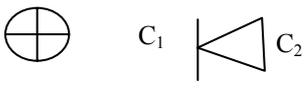
1-4 Classification des robots : [14] :

On peut aborder des classifications qui ne s'attachent qu'à un aspect particulier du robot. C'est le cas du classement morphologique ou cinématique reconnu par la norme (I.S.O. 8373) qui ne porte attention là encore, qu'aux robots manipulateurs industriels à poste fixe. Cette norme propose le découpage du tableau 1.1.

- on classe structurellement les robots en fonction des systèmes de coordonnées dans lesquels ils travaillent [7,26] :
 - Cartésien : Trois axes de translation.
 - Cylindrique : Deux axes de translation, un axe de rotation.
 - Sphérique : Un axe de translation, deux axes de rotation.
 - Articulé : Trois axes de rotation.
- Quand on commande un robot, on a intérêt à pouvoir contrôler individuellement chaque articulation ou axe pour être bien maître de la trajectoire. C'est pourquoi on utilise principalement des liaisons pivot (articulations cylindriques). Elles ont par ailleurs l'avantage d'une réalisation pratique peu onéreuse comparée aux articulations à plusieurs degrés de liberté. On comprend qu'une combinaison d'articulations cylindriques et prismatiques permet de réaliser toutes les liaisons mécaniques autour de trois axes normaux concourants donnant une rotule à l'articulation rotoïde. [14].

Tableau 1.1 : [14] : Représentation des liaisons mécaniques entre deux corps
(D'après N.F. EO4-O15).

Nom de Liaison	Mouvement relatifs	Nombre de Degré de liberté	Symbole
Encastrement	0: rotation 0: translation	0	 C1: corps 1 C2: corps 2
Pivot Articulation Cylindrique	1: rotation 0: translation	1	
Glissière Articulation Prismatique	0: rotation 1: translation	1	
A	B	C	D

A	B	C	D
Glissière Hélicoïdale	1: rotation 1: translation conjuguées	1	
Appui Plan	1: rotation 2: translation	3	
Rotule Articulation Rotoïde	0: rotation 3: translation	3	
Linéaire Rectiligne	2: rotation 2: translation	4	
Linéaire Annulaire	3: rotation 1: translation	4	
Ponctuelle	3: rotation 2: translation	5	
Libre	3: rotation 3: translation	6	Pas de symbole, pas de contact entre les deux corps

1.5 Les porteurs (bras) : [9, 14, 140] :

Cette configuration permet de classifier les robots par type de porteurs :

- structure cartésienne (T.T.T) (3.d.d.l.).
- structure sphérique (ou polaire).
- structure angulaire (3R).
- structure SCARA.

Les bras manipulateurs à chaîne continue ouverte et à chaîne arborescente sont les plus nombreux, ces bras manipulateurs sont simples sur le plan conception car sans risque d'hyper statisme largement exploitées, ces structures ont donné naissance à plusieurs générations de robots pratiquement chez tous les constructeurs.

Les bras manipulateurs conçus sur la base d'une structure mécanique de type chaîne complexe ont pour avantage essentiel d'augmenter la rigidité et par conséquent la précision en mode programmé. Ces chaînes permettent en outre un meilleur équilibrage statique.

Leur exploitation industrielle, bien que nécessaire dans certain cas, revient trop chère et

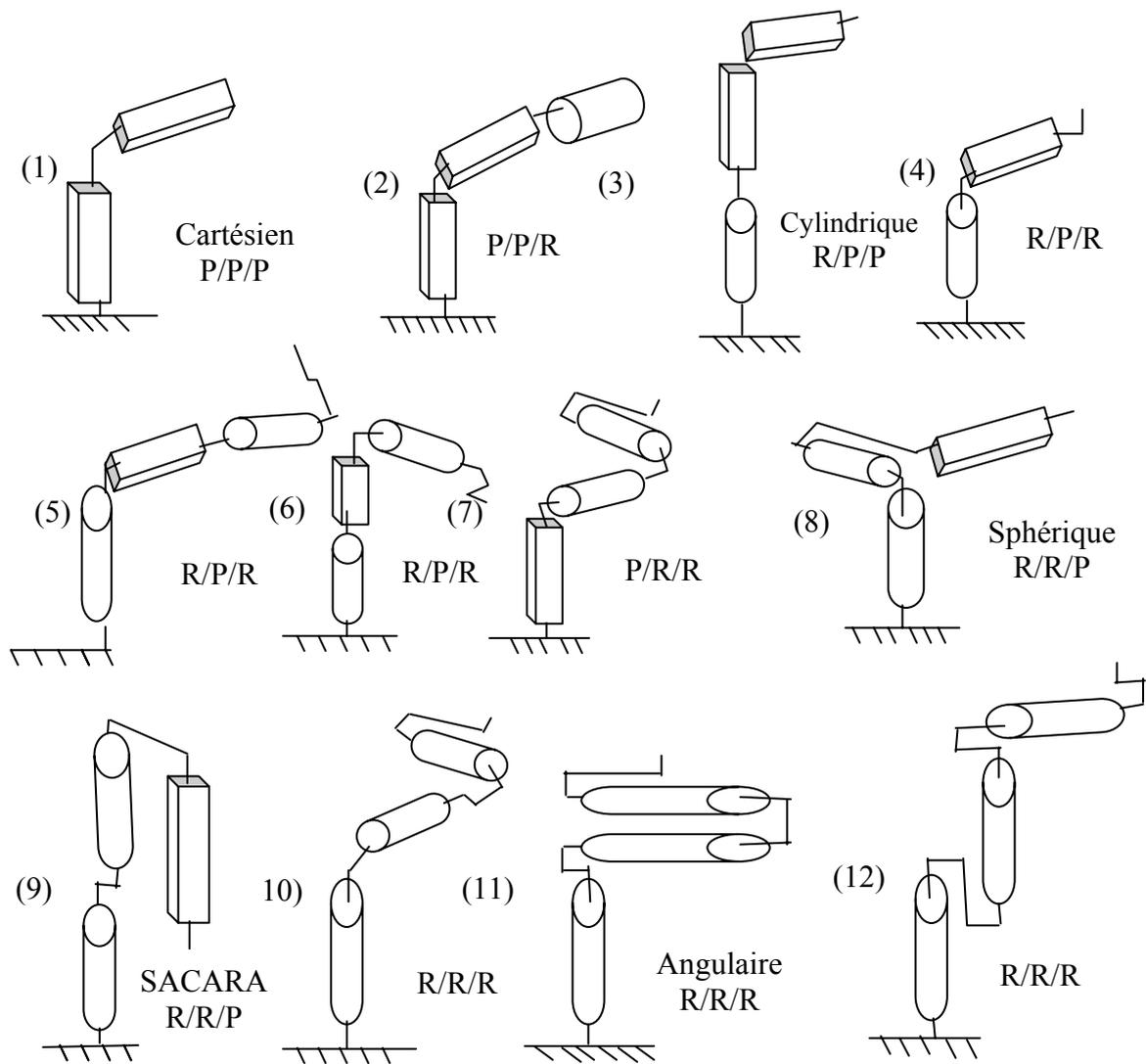


Figure 1.4 : les douze porteurs de robots

Le nombre total des possibilités des combinaisons de trois articulations en série est de trente six, dont seulement douze remplissent la fonction du porteur spatial et sont mathématiquement distinctes.

Dans la pratique il semble dans une étude portant sur cent quinze robots que quatre ou cinq arrangements soient utilisés pour des raisons géométriques, la figure 1.6 : reproduit ces cinq architectures avec leur partage d'utilisation.

Tous des robots sont de type robot série signifiant par là qu'en partant de la base et en allant vers l'extrémité on rencontre les articulations les unes après les autres sur la même chaîne cinématique. [14].

Les télémanipulateurs comprennent un bras maître et bras esclave qui répète les mouvements du bras maître.

Les manipulateurs assistés sont dépourvus du bras- maître qui est remplacé par une poignée de manœuvre située sur l'organe de préhension.

En milieu industriel, on trouve des télémanipulateurs lorsque la sécurité de l'homme est menacée (en génie nucléaire. On manipule à distance les produits radioactifs). [55].

Dans le cadre d'une définition restrictive fréquente d'un robot industriel, l'unité centrale est constituée mécaniquement par un bras manipulateur caractérisé par: plusieurs articulations successives à partir d'une base fixe, de façon à donner l'outil terminal les mobilités nécessaires (jusqu'à 3 degrés de liberté en déplacements et 3 en orientation). [60].

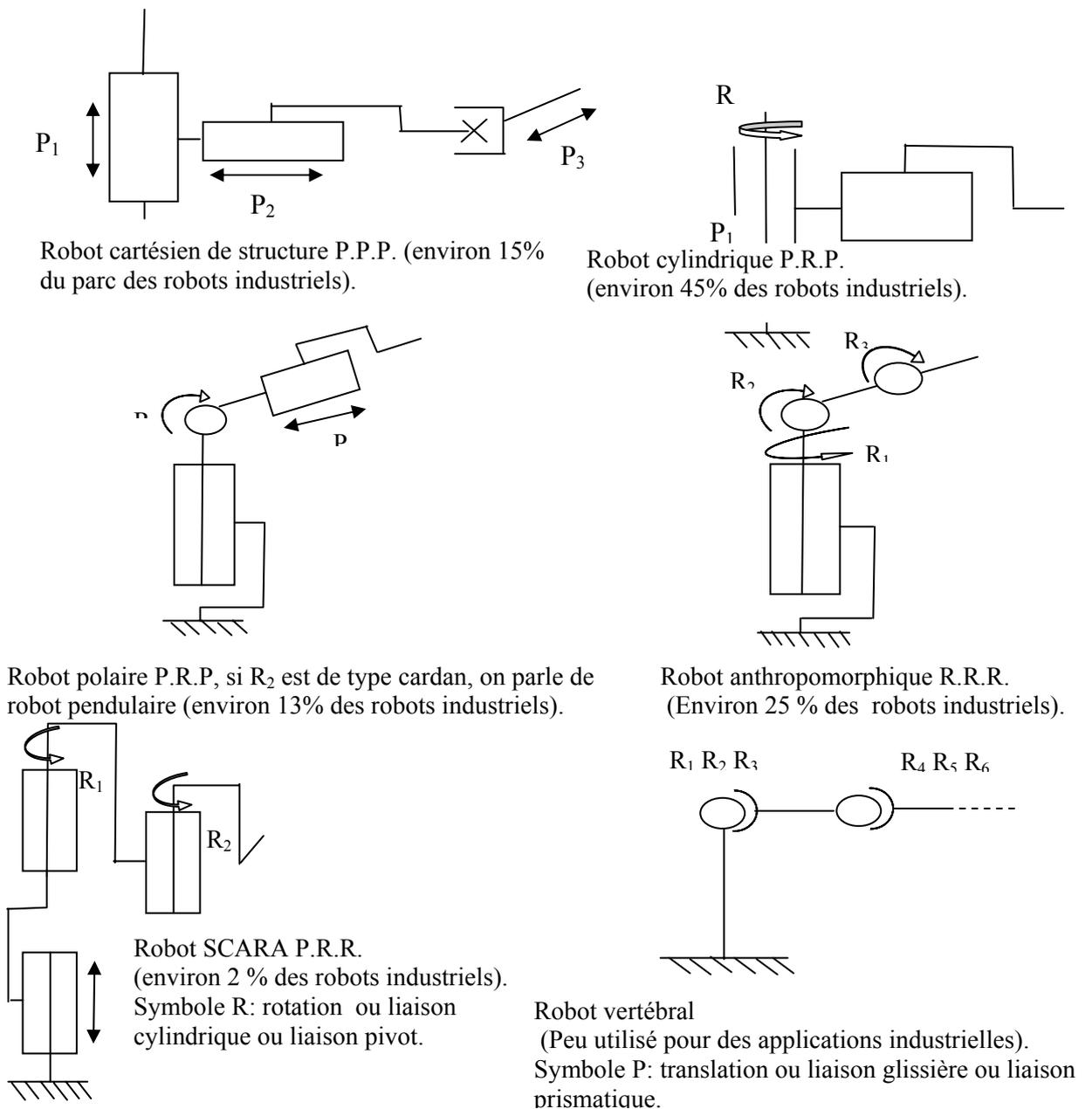


Figure 1.5. : Robots Manipulateurs Industriel à Poste Fixe (d'après I.S.O 8373). [14].

1-5 Caractéristiques générales des robots : [21] :

Un robot est caractérisé par une structure arborescente articulée simple ou multiple dont les segments sont mobiles les uns par rapport aux autres.

Cet ensemble a pour objectif de mener l'organe terminal vers un lieu géométrique imposé par la tâche.

Les actions directes ne posent à priori pas de problèmes mathématiques spécifiques puisque les informations liées aux divers points de constitution des trajectoires sont mémorisées au fur et à mesure d'un apprentissage.

Dans le second cas, la description du but, ou sens géométrique par un système de décision est effectuée selon une méthode de représentation particulière par rapport à des références connues.

L'exécution de l'ordre se fait par des transformations successives dans divers espaces, reliés au robot vers l'espace de la tâche ou espace opérationnel par rapport à d'autres références.

- Description de la tâche en termes de position géométrique avec éventuellement des contraintes.
- Transformation de la description initiale dans l'espace des variables articulaires propres à chaque robot, chaque variable représente une mobilité du système mécanique.
- Déduction des couples moteurs à générer pour déplacer les articulations.
- Mouvement du système mécanique vers le but dans l'espace réel de la tâche.

1-6 Conclusion :

La structure mécanique d'un robot manipulateur est composée de plusieurs corps connectés les uns aux autres par des liaisons appelées articulations ou joints, à un seul d.d.l. de translation ou rotation, cette structure mécanique peut constituer une chaîne cinématique continue ouverte simple, une chaîne arborescente ou une chaîne complexe.

Un système de commande d'un robot peut être vu comme un ensemble de processus communicants, ces processus interdépendants sont alternativement producteur (s) ou consommateurs de données, ils sont classés en deux catégories :

Des processus opératifs qui agissent sur l'environnement du manipulateur

[Autres(s) manipulateur(s), autre (s) équipement(s) automatisé(s)].

Des processus sensoriels, liés aux capteurs extéroceptifs qui acquièrent des informations sur l'état de l'environnement et engendrent des messages pouvant être reçus et traités par les processus opératifs, l'exécution de ces processus est en général confiée à des processus que celui qui sert à l'exécution [18].

Pour réaliser la commande automatique par ordinateur d'un système mécanique articulé il faut être en possession d'un modèle de ce dernier. Ce modèle est élaboré à partir de certains paramètres caractéristiques du système (l'agencement des D.D.L. les longueurs masses inerties. etc...), ces paramètres sont plus ou moins nombreux suivant qu'on s'intéresse au modèle cinématique ou géométrique. Il est évident que ces modèles vont s'éloigner de la réalité au fur et à mesure que les vitesses augmentent et que, d'une part, des forces inertielles, centrifuges et de couplage vont apparaître, et d'autre part les jeux frottements et élasticités de toutes origines ne vont plus pouvoir être négligés. Il est nécessaire de revoir la modélisation [21].

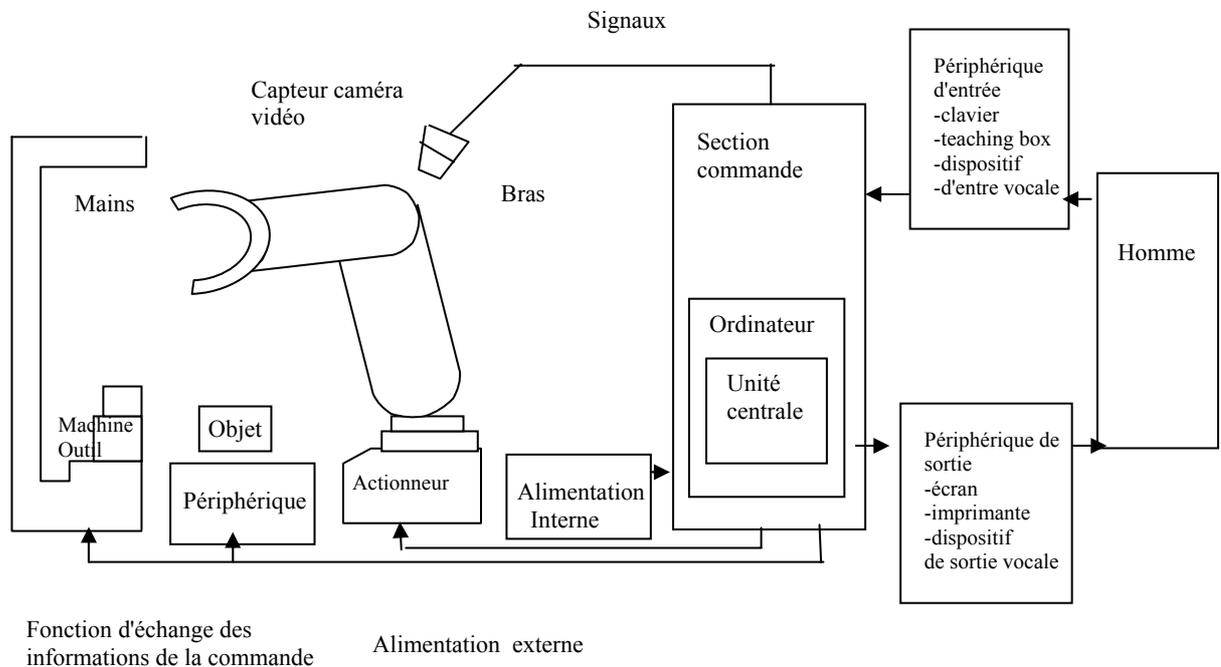


Figure 1.6 : la composition du robot industriel [7]

CHAPITRE 2

**MODELISATIONS GEOMETRIQUE, CINEMATIQUE ET DYNAMIQUE
DIRECT ET INVERS DES SYSTEMES MECANQUES POLY ARTICULE AUX
ELEMENTS RIGIDES**

2.1 Modèle géométrique [9, 27] :

Le modèle géométrique d'un robot constitue une représentation mathématique, en le considérant comme une chaîne simple, ouverte, de (n) solides (corps), rigides, sans boucles ni branchement, sans masses, articulés entre eux; chaque corps est réduit à sa plus simple expression, c'est-à-dire à son orientation et à sa position par rapport au corps précédent.

$C_i = (P_i, O_i) = (\text{position } i, \text{orientation } i)$.

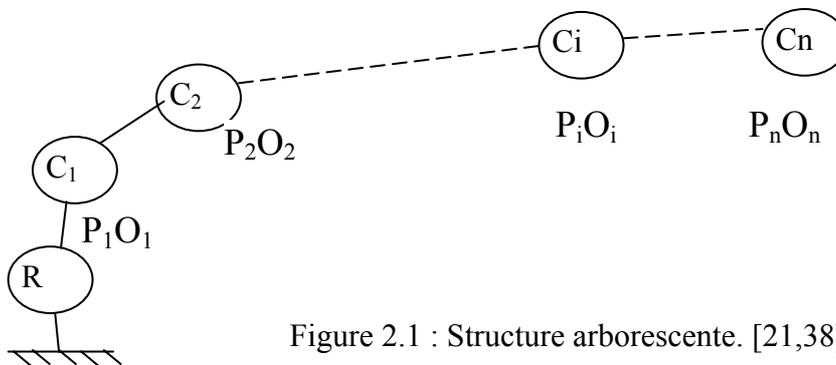


Figure 2.1 : Structure arborescente. [21,38]

Chaque configuration géométrique possible est définie par un ensemble de variables articulaires qui traduisent les déplacements relatifs d'un corps par rapport au précédent. Les variables articulaires (ou coordonnées articulaires ou coordonnées généralisées) font le lien entre la position et l'orientation de l'organe terminal et les consignes définies dans un repère de base.

Nous l'exprimons par :

$$\underbrace{X_i / R_i}_{\text{Variables opérationnelles}} = F(\underbrace{q_1, q_2, \dots, q_n}_{\text{Variables articulaires}}) \dots \dots \dots (2.1)$$

2.2 Les variables articulaires :

Les variables articulaires expriment l'ensemble des possibilités de mouvement entre deux articulations, le modèle géométrique est composé tel que sur les six possibilités de mouvements élémentaires d'un corps par rapport à un autre (3 rotations et 3 translations) une seule est retenue [8,10, 11,12]. Ainsi les possibilités de mouvements multiples aux niveaux technologiques sont décomposées en mouvements élémentaires, au niveau du modèle, l'orientation entre deux repères peut être traitée selon plusieurs techniques classiques ; Les matrices de rotation, les cosinus directeurs, les angles de Bryant, les angles d'Euler [12, 10, 11]...

2.3 Le système de paramètres de Denavit-Hartenberg : [11, 12, 13, 24, 25, 26, 27, 28] :

La méthode est la plus couramment utilisée en robotique pour la définition de l'orientation et de la position des différents éléments d'un système mécanique articulé.

Dans le domaine de la robotique, l'élaboration du modèle nécessite une étude détaillée et approfondie de la structure du robot. Dans la littérature il existe plusieurs méthodes pour la modélisation des mécanismes à structures de chaînes simples ou complexes, ouvertes ou fermées [24].

Les plus utilisées sont les méthodes de Denavit-Hartenberg et Sheth-Uiker, La première est très bien adaptée pour les mécanismes à structures de chaînes simples où toutes les liaisons sont élémentaires, mais, elle présente des difficultés lorsqu'il s'agit de mécanismes à structures de chaînes complexes; en effet les corps possédant plusieurs liaisons élémentaires (rotoïdes et / ou prismatiques) en aval, doivent être dotés d'autant de repères, ce qui entraîne des lourdeurs. La deuxième méthode vient palier les inconvénients cités précédemment, mais elle présente des redondances pour les mécanismes à structures de chaînes simples.

L'utilisation de la transformée de Denavit-Hartenberg (D.-H.), facilite la description géométrique du manipulateur ; cette dernière nous permet d'aboutir au modèle cinématique et géométrique direct et inverse du robot.

La même transformation offre une souplesse dans le calcul du modèle dynamique direct en utilisant le formalisme d'Euler Lagrange [25, 26, 27, 28].

Denavit et Hartenberg ont proposé une méthode qui repose sur l'assignation d'un repère unique pour chaque lieu, cette convention est une méthode systématique, elle permet le

passage entre articulations adjacentes d'un système robotique, elle concerne les chaînes cinématiques ouvertes ou l'articulation possède uniquement un degré de liberté, les surfaces adjacentes restent en contact. Le choix adéquat des repères dans les liaisons, facilite le calcul des matrices homogènes de Denavit-Hartenberg et permet d'arriver rapidement aux informations de l'élément terminal dans la base et vice versa; la figure 2.2 représente l'utilisation de cette notation pour deux liens successifs l'axe Z_i du repère est concourant avec l'axe de l'articulation i , quand à l'axe X_i , il est sur la droite perpendiculaire aux axes Z_{i-1} et Z_i .

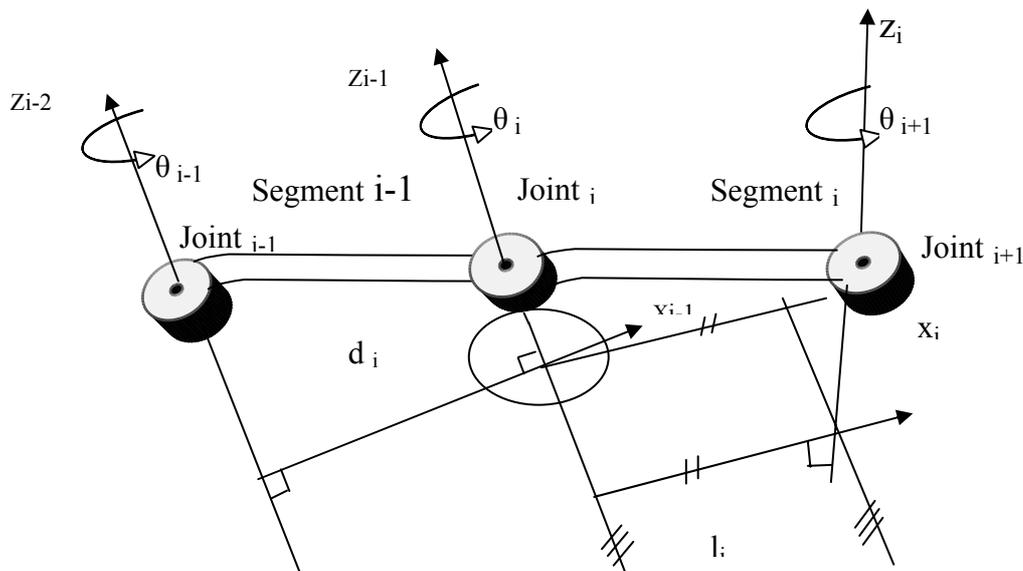


Figure 2.2 : Système de Coordonnées et Paramètres de Denavit-Hartenberg [12, 21, 27, 29, 30,31, 38,71].

Quatre paramètres sont alors utilisés pour décrire la forme géométrique d'un lien et sa position par rapport au lien précédent, la notation de Denavit-Hartenberg ne fonctionne que pour des chaînes cinématiques sérielles, (pour des chaînes arborescentes des ambiguïtés apparaissent),

Cette notation, apparue très tôt dans le domaine de la robotique est encore largement utilisée par la communauté scientifique pour décrire les robots en vue de leur analyse et/ ou modélisation; quelques variantes relativement proches sont aussi courantes, comme par exemple la notation Paul, cette notation se différencie, essentiellement, de celle de Denavit-Hartenberg, par l'assignation des paramètres, relativement aux liens, (décalage des indices).

Les étapes à suivre pour cette technique sont les suivantes :

1^{ère}. Numérotation des segments constitutifs du bras manipulateur de la base vers l'élément terminal, on associe le référentiel "zéro" à la base de celui-ci, et l'ordre "n" à l'élément terminal (effecteur).

2^{ème}. Définition des axes principaux de chaque segment :

- Si z_i et z_{i-1} ne se coupent pas, on choisit x_i de manière à être en parallèle avec l'axe perpendiculaire à z_i et z_{i-1} .
- Si z_i et z_{i-1} sont colinéaires, on choisit x_i dans le plan perpendiculaire à z_{i-1} .

3^{ème}. Fixer les quatre paramètres géométriques : d_i , θ_i , a_i et α_i , (voir la figure 2.2) pour chaque articulation tels que :

- d_i est une coordonnée de l'origine o_i sur l'axe z_{i-1} pour une glissière d_i est une variable et pour une charnière d_i est une constante.
- θ_i est l'angle que l'on obtient par vissage de x_{i-1} vers x_i autour de l'axe z_{i-1} pour une glissière θ_i c'est une constante et pour une charnière θ_i c'est une variable.
- l_i est la distance entre les axes z_i et z_{i-1} mesuré sur l'axe x_i négatif à partir de son origine jusqu'à l'intersection avec l'axe z_{i-1} .
- α_i est l'angle entre l'axe z_i et z_{i-1} obtenu en vissant z_{i-1} vers z_i autour de x_i .

4^{ème}. On forme enfin, la matrice homogène de Denavit-Hartenberg de déplacement qui lie la rotation et la translation, la partie supérieure gauche définit la matrice de rotation R_{i-1} et le vecteur de translation à droite d_{i-1}^i .

Par la suite on aboutit à la matrice de transformation de Denavit-Hartenberg suivante :

$$T_i^{i-1} = \begin{pmatrix} \cos \theta_i & -\sin \theta_i & 0 & a_{i-1} \\ \sin \theta_i \cos \alpha_{i-1} & \cos \theta_i \cos \alpha_{i-1} & -\sin \alpha_i & -\sin \alpha_{i-1} d_i \\ \sin \theta_i \sin \alpha_{i-1} & \cos \theta_i \sin \alpha_{i-1} & \cos \alpha_{i-1} & -\cos \alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \dots\dots\dots (2.2)$$

Finalement on peut écrire le modèle géométrique direct sous la forme :

$$X = f(q) \dots\dots\dots (2.3)$$

Avec $X \in \mathbb{R}^6$ les coordonnées cartésiennes, et $q \in \mathbb{R}^6$, les coordonnées articulaires, on va admettre quelques hypothèses [12, 27, 29, 30, 31] dans le but de simplifier la modélisation des robots, ces hypothèses sont les suivantes :

- Les liaisons du manipulateur sont rigides.
- Les jeux dans les articulations sont négligeables.
- Les capteurs ont un gain unitaire et de dynamique négligeable

L'orientation d'un repère est donnée par une matrice 3x3, représentant les 3 vecteurs unitaires. On a cependant indiqué que dans ces 9 valeurs, plusieurs sont redondantes et qu'en fait, il est possible de donner l'orientation en donnant simplement 3 valeurs. Supposons, par exemple, qu'on désire avoir un robot qui suit une certaine trajectoire dans l'espace en lui donnant un certain nombre de points intermédiaires à passer. Si chacun des points est donné par 3 positions et une matrice d'orientation 3x3, cela risque de prendre assez de mémoire, il est préférable de ne prendre que 3 chiffres pour l'orientation, en fait il y a une infinité de choix possibles pour définir une orientation d'un repère par rapport à un autre, l'idée est de trouver 3 transformations qui vont faire passer le premier dans le deuxième, on peut adopter la paramétrisation d'Euler ZYZ, chaque liaison d'un manipulateur fait des rotations ou des translations par rapport au référentiel d'inertie fixe, par exemple, un repère fixé à la base du robot.

Le calcul des coordonnées des liaisons du manipulateur exprimées dans le référentiel d'inertie de la base est relativement difficile, cette difficulté augmente suivant l'ordre de la liaison jusqu'à l'élément terminal; pour ne pas alourdir les calculs et ramener toutes les informations géométriques au repère d'inertie de la base, il est judicieux de localiser les articulations correspondantes et situer chaque liaison à son propre référentiel.

Le passage d'un référentiel à un autre est garanti par les transformations homogènes.

Lorsqu'on a uniquement des rotations, on se satisfait d'une matrice de transformation R de troisième ordre, et lorsqu'il existe une translation autour d'un point, on est obligé de passer à une matrice de quatrième ordre pour permettre au référentiel de transformer, dans ce cas le vecteur de position ${}^a p$ sera augmenté par une quatrième composante pour avoir un vecteur de position ${}^a p'$ exprimé par ses coordonnées homogènes [27, 30] :

$${}^a p = \begin{pmatrix} P_x \\ P_y \\ P_z \end{pmatrix} \dots\dots\dots(2.4)$$

Le vecteur homogène correspondant est :

$${}^a p' = \begin{pmatrix} P_x \\ P_y \\ P_z \\ 1 \end{pmatrix} \dots\dots\dots(2.5)$$

La matrice augmentée de transformation aura la forme suivante [27, 30] :

$$T = \begin{pmatrix} r_{11} & r_{12} & r_{13} & d_x \\ r_{21} & r_{22} & r_{23} & d_y \\ r_{31} & r_{32} & r_{33} & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \dots\dots\dots(2.6)$$

Avec r_{ij} les composantes de la matrice de rotation R_1 et d_x , d_y et d_z sont les composantes du vecteur de translation qui comportent les coordonnées du repère de destination dans le repère source. Si on appelle T la matrice de transformation du référentiel (x_1, y_1, z_1) vers le référentiel (x_2, y_2, z_2) alors $P' \begin{matrix} x_2 \\ y_2 \\ z_2 \end{matrix} = T P' \begin{matrix} x_1 \\ y_1 \\ z_1 \end{matrix} \dots\dots\dots (2.7)$

2.4 Modèle géométrique inverse : [27, 28] :

Le modèle géométrique direct d'un robot permet de calculer les coordonnées opérationnelles en donnant la situation de l'organe terminal en fonction des coordonnées articulaires; le problème inverse consiste à calculer les coordonnées articulaires correspondant à une situation donnée de l'organe terminal.

Lorsqu'elle existe, la forme explicite qui donne toutes les solutions possibles -il y a rarement unicité de solution-constitue ce qu'on appelle le modèle géométrique inverse (M.G.I.).on présentera trois méthodes pour le calcul du M.G.I.

La méthode de Paul : Elle traite séparément chaque cas particulier et convient pour la plupart des robots industriels.

La méthode de Pieper : Elle permet de résoudre le problème pour les robots à six degrés de liberté possédant trois articulations rotoïdes ,d'axes concourants ou trois articulations prismatiques.

La méthode générale de Raghavan et Roth : on donnera la solution générale des robots à six articulations à partir d'un polynôme de degrés au plus égal à 16.

Le modèle géométrique inverse consiste à trouver l'angle de chaque articulation pour une position et orientation données de l'effecteur (l'élément terminal).

Le problème posé est le suivant : étant donné la position et l'orientation de l'outil par rapport à la situation de travail, comment calculer l'ensemble des angles articulaires qui accomplissent cet objectif ? La réponse à cette question constitue le modèle géométrique inverse du robot manipulateur.

La situation du problème, concernant la recherche des angles articulaires nécessaires, pour positionner le repère de l'outil, par rapport au repère de la station de travail est décomposée en deux parties. En premier lieu, on détermine les transformations nécessaires pour trouver

le repère du poignet, par rapport au repère de la base, et après, le modèle géométrique inverse est utilisé pour trouver les angles des articulations.

Une approche analytique a été utilisée pour un robot série, (Craig), elle consiste à éliminer à chaque étape une des coordonnées généralisées (articulaires) par la multiplication de la matrice de transfert finale T_6^0 par les matrices de transformation intermédiaires pour le cas d'un robot à six degrés de liberté. La matrice de transformation T_6^0 s'écrit sous la forme

$$T_6^0 = \begin{pmatrix} r_{11} & r_{12} & r_{13} & d_x \\ r_{21} & r_{22} & r_{23} & d_y \\ r_{31} & r_{32} & r_{33} & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = T_1^0(\theta_1) \cdot T_2^1(\theta_2) \cdot T_3^2(\theta_3) \cdot T_4^3(\theta_4) \cdot T_5^4(\theta_5) \cdot T_6^5(\theta_6) \dots \dots \dots (2.8)$$

Par exemple la première étape sera comme suit : On multiplie de part et d'autre

la matrice $T_1^0(\theta_1)$, le résultat sera : $[T_1^0]^{-1} \cdot T_1^0 = T_6^1 \dots \dots \dots (2.9)$

On s'intéresse toujours à la dernière colonne de la matrice, qui contient à chaque étape les équations découplées qui permettent de résoudre le problème du modèle géométrique inverse, toutefois le modèle géométrique comporte aussi des inconvénients :

La non unicité du modèle géométrique inverse implique qu'il existe plusieurs "chemins" pour se rendre d'un point à un autre. Le traitement par incrément peut amener à des imprécisions. Des singularités mécaniques et / ou mathématiques apparaissent.. Une haute précision de solutions obtenues n'est pas nécessaire puisqu'il suffit de fournir à l'utilisateur une vision globale, le calcul des accroissements est à chaque fois effectué à partir d'une nouvelle configuration exacte du robot. Quant au problème des singularités, il existe plusieurs méthodes mathématiques pour les traiter et les éviter.

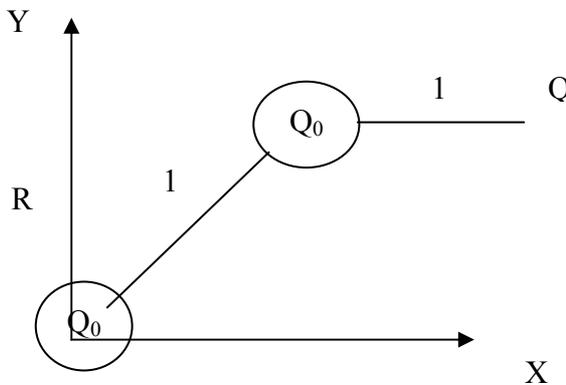


Figure 2.3 : Repère avec deux articulations. [38]

2.5 Résolvabilité des Systèmes Mécanique Poly Articulés

2.5.1 Absence de solution :

2.5.1.1 Origine mécanique :

les mouvements du mécanisme tiennent compte des limites des rotations et translations.

Des butées empêchent le robot d'atteindre les points en dehors de volume du travail malgré l'existence de solutions mathématiques.

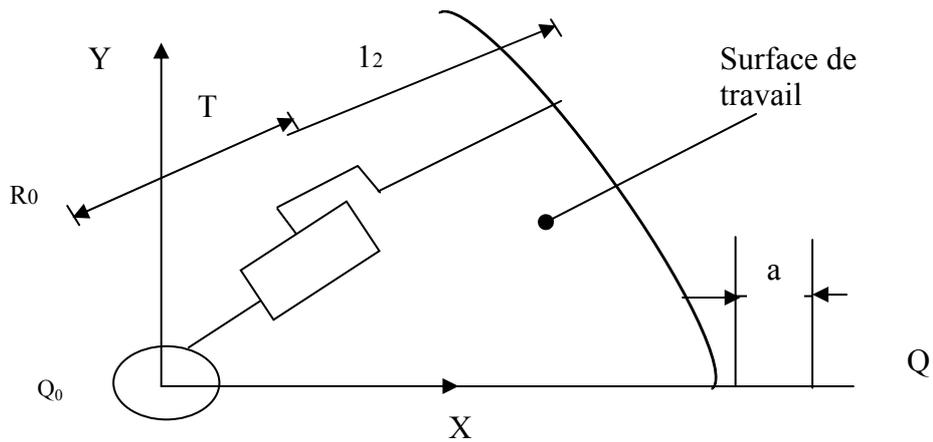


Figure. 2.4 : Le robot R.P. [38]

2-5-1-2 Origine mathématique :

un système dont le nombre de variables est inférieur au nombre d'équations ne donne pas de solutions mathématiques. Cela revient, en robotique, à imposer plus de contraintes qu'il n'existe de degrés de liberté.

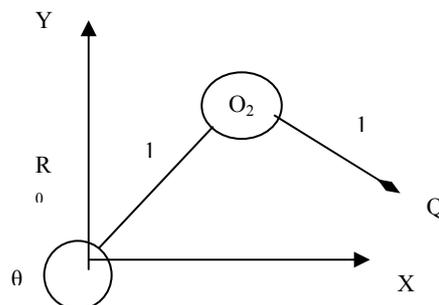


Figure 2.5 : La structure R.P. [38]

2-5-2 Infinité de solution :

la résolution du système d'équation peut aboutir à une infinité de solutions dans deux cas :

lorsque le nombre de contraintes est inférieur au nombre de d.d.l .du robot, on se trouve en face de surabondance de potentialités en fonction de la tâche demandée. La solution consiste à réduire le nombre de variables articulaires en leur imposant une valeur. Dans le second cas, le robot se trouve en position de singularité.

Cette configuration particulière est créée par exemple lors de la mise en parallèle de deux axes.

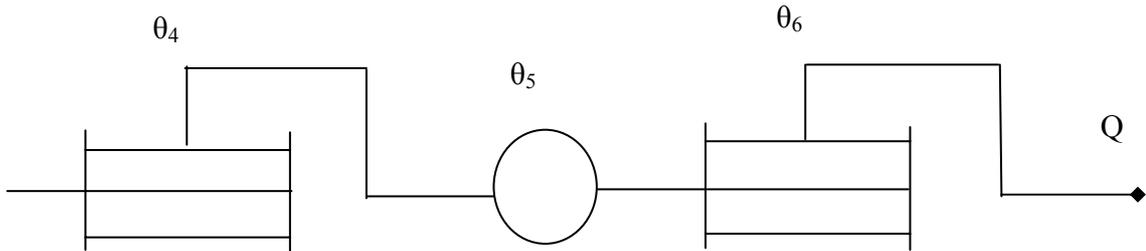


Figure 2.6 : Représentation d'un poignet. [38]

2-5-3 Nombre fini de solutions:

Si le nombre de contraintes est égal au nombre de d.d.l .du mécanisme et si le robot ne se trouve pas dans un des cas décrits plus haut, alors il existe une ou plusieurs solutions au problème.

2-6 Calcul du modèle inverse :

Pour la résolution des problèmes inverse c'est-à-dire résoudre le système d'équations non linéaires, on utilisera les méthodes de Newton- Raphson,Range kutta,loi de Bang-Bang et le formalisme de Newton-Euler (voir appendices).

2-7 Analyse cinématique : [4, 21,38,39 , 40, 71]:

La position de l'organe terminal dans l'espace opérationnel peut être écrite en termes de variables articulaires comme suit :

$$x = f(q) \dots\dots\dots *$$

Le modèle cinématique direct permet d'obtenir la vitesse de l'organe terminal dans l'espace opérationnel en fonction des vitesses articulaires.

En différenciant l'équation * on obtient:

$$\dot{X}_j = \sum_j \frac{\partial f_j}{\partial q_j} \dot{q}_j \dots\dots\dots (2.10)$$

Qui s'écrit aussi sous forme matricielle :

$$[\dot{X}] = [J][\dot{q}] \dots\dots\dots (2.11)$$

La Jacobinne du système est définie par :

$$[J] = \frac{\partial f_j}{\partial q_j} \dots\dots\dots (2.12)$$

Les accélérations sont données par :

$$\ddot{X} = \sum_j \frac{\partial f_i}{\partial q_j} \ddot{q}_j + \sum_j \sum_k \frac{\partial^2 f_i}{\partial q_j \partial q_k} \dot{q}_j \dot{q}_k \dots\dots\dots (2.13)$$

Ou bien sous forme matricielle par :

$$\ddot{x} = [J]\ddot{q} + [A]\dot{q}^2 \dots\dots\dots (2.14)$$

$$\text{ou : } q = [q^1, q^2, \dots\dots\dots]^t \dots\dots\dots (2.15)$$

$$\dot{q}^2 = [\dot{q}_1 \dot{q}_1, \dot{q}_1 \dot{q}_2, \dots\dots\dots]^t \dots\dots\dots (2.16)$$

j (q) est la matrice Jacobienne donnée par :

$$j(q) = \frac{\partial f_j}{\partial q} \dots\dots\dots (2.17)$$

$$\text{Et : } [A] = \left(\frac{\partial^2 f_j}{\partial q_j \partial q_k} \right) \dots\dots\dots (2.18)$$

A partir du modèle cinématique (2.18) on peut écrire le modèle différentielle (2.19).

Supposons que les variables q_i soient maintenant non les variables articulaires de Denavit-Hartenberg, mais les variables associées aux déplacements des moteurs rotatifs ou linéaires et que le robot présente une chaîne cinématique directe passant par ces moteurs. Il existe alors un modèle différentiel du type (2.24). Chaque actionneur peut associer en statique la force ou le couple δ_i qu'il exerce et forme le vecteur :

$$\Gamma = [\delta_1, \dots\dots\dots, \delta_i, \dots\dots\dots, \delta_n]^T \dots\dots\dots (2.19)$$

Des forces articulaires sous l'effet de ces forces combinées, l'organe terminal exerce sur l'environnement des forces qui peuvent être réduites à leur torseur résultant (force et mouvement) noté F , qui a donc six composantes. En utilisant alors la relation :

$$\begin{pmatrix} P_N \\ \omega_N \end{pmatrix} = J_0(q)q \dots\dots\dots (2.20)$$

Ou : (P_N) : La vitesse du point de référence par rapport au repère Fixe.

(ω_N) : La vitesse de rotation instantanée et le principe des travaux virtuels pour des déplacements infinitésimaux de type (2.24) ou à alors :

$$\Gamma = J_0^T f \dots\dots\dots (2.21)$$

Qui permet de calculer les forces matrices nécessaires pour exercer sur l'environnement des forces données. L'équation (2.22) constitue donc plutôt un modèle inverse au sens habituel du terme. Le modèle direct ne peut s'obtenir que si la matrice J est régulière.

Dans le cas d'un robot non redondant (n=6) et en dehors des singularités, on a alors le modèle direct :

$$F = (J_0^T)^{-1} \Gamma \dots\dots\dots (2.22)$$

Tableau 2.1 : Les paramètres géométriques du robot type.[140,141]

Indice	0	1	2	3	4	5
d_i	0	h_1	h_2	h_3	h_4	
α_i		0	$\pi/2$	$\pi/2$	0	
σ		0	1	1	1	1
r_i		0	Z_2	Z_3	Z_4	Z_5
θ_i		γ	0	0	0	0

2-8 Détermination des matrices de changements de repères <i-1/i> pour le robot type (ALG-M.O.1).

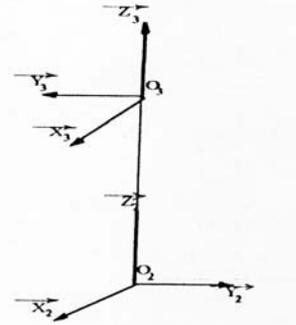
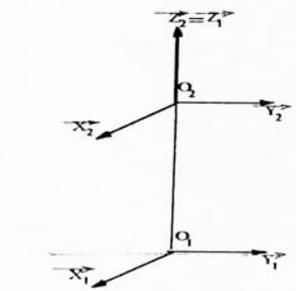
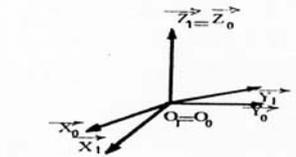
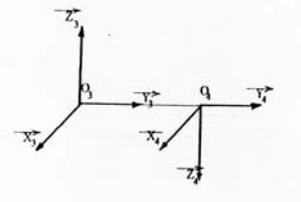
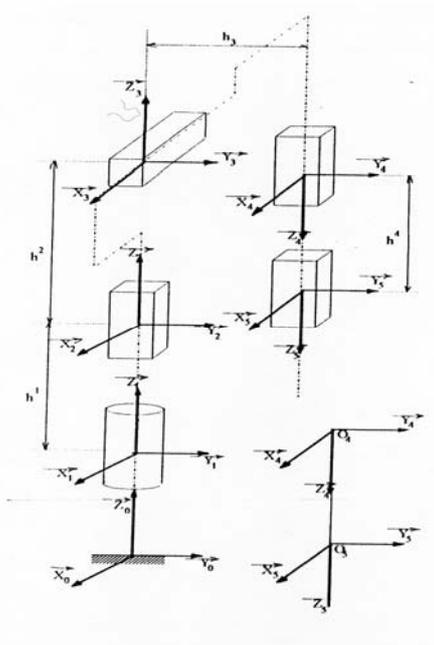


Figure 2.7 : Les différents repères liés au corps du robot.

2-8-1 Espace de travail :

L'espace de travail est l'ensemble des positions et/ou orientations accessible par l'organe terminal du robot.

Le volume ou l'espace de travail d'un robot dépend généralement de trois facteurs :

- De la géométrie du robot,
- De la longueur des segments,
- Du débattement des articulations (limité par des butées)

2-8-2 Analyse de l'espace de travail du robot type (ALGERIE-MACHINES OUTILS-1) :

L'analyse de l'espace de travail des robots trouve de nombreuses applications. Notamment dans le domaine de la C.A.O.- Robotique pour la conception optimale des robots, des sites robotisés, et pour la programmation hors ligne.

Soit $q = [q_1, \dots, q_n]$ un élément de \mathbb{R}^n représentant une configuration articulaire donnée et soit $x = [x_1, \dots, x_n]$ l'élément de l'espace opérationnel \mathbb{R}^n correspondant, tel que :

$$X = f(q) \dots\dots\dots (2.23)$$

On note Q l'ensemble des configurations accessibles compte tenu des butées articulaires.

Par conséquent, Q sera appelé domaine articulaire :

L'image de Q par le modèle géométrique direct f définit l'espace de travail W du robot :

$$W = f(Q) \dots\dots\dots (2.24)$$

2-8-3 Calcul de l'espace de travail du robot choisi :

Comme on l'a définie précédemment, la position de l'organe terminal dans le repère atelier est donnée par le vecteur position dans la matrice de transformation qui exprime le repère R_5 dans R_0 noté :

$$P = \begin{Bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{Bmatrix}; \text{ avec } \begin{cases} p_x = (z_3 - z_4)c\gamma_1 \\ p_y = (z_3 - z_4)s\gamma_1 \\ p_z = h_1 + h_2 + h_3 + h_4 + z_2 - z_5 \end{cases} \dots\dots\dots (2.25)$$

En développant un programme qui a comme paramètres entrées les limites articulaires de chaque articulation et sortie toutes les configuration possibles de l'organe terminal.

2-8-4 Algorithme simplifié du programme :

Algorithme 2.1: Algorithme simplifié du programme :

Début programme

Entrer h_{\min} et h_{\max} pour chaque articulation ;

Début Do

Incrémentation de h_1 et entre $h_{1\min}$ et $h_{1\max}$

Incrémentation de h_2 et entre $h_{2\min}$ et $h_{2\max}$

Incrémentation de h_3 et entre $h_{3\min}$ et $h_{3\max}$

$$\begin{cases} p_x = (z_3 - z_4)c\gamma_1 \\ p_y = (z_3 - z_4)s\gamma_1 \\ p_z = h_1 + h_2 + h_3 + h_4 + z_2 - z_5 \end{cases}$$

Fin Do

Sortie (P_x, P_y, P_z)

2-9 Modélisation dynamique des systèmes mécanique articulé aux éléments rigides : [8, 9, 21, 25, 40, 41, 42, 43, 71] :

Les modèles dynamiques des bras manipulateurs sont décrits par un ensemble d'équations mathématiques qui portent la dynamique de ceux-ci et peuvent être simulées sur ordinateur dans le but de synthétiser une commande conditionnée par des performances désirées, les équations différentielles qui décrivent le comportement d'un mécanisme à plusieurs corps articulés peuvent être déterminés par des lois mécaniques classiques Newtoniennes (théorèmes généraux de la mécanique classiques) et Lagrangiennes.

Les approches d' Euler- Lagrange et Newton- Euler permettent d'aboutir aux équations du mouvement des robots.

2-10 Méthodes d'obtention du modèle dynamique : [8, 9, 21, 41] :

Les principales méthodes actuelles d'obtention du modèle dynamique sont basées sur l'un des quatre formalismes suivants :

- La notion d'énergie d'accélération ou fonction de Gibbs.
- Les équations de Newton et d' Euler.
- Le principe du travail virtuel de D'Alembert.
- Les équations de Lagrange.

Ces dernières semblent les plus utilisées et peuvent être les plus faciles à manipuler.

2-10-1 Obtention du modèle dynamique : [42, 43]

L'énergie cinétique du système est une forme quadratique des vitesses articulaires :

$$E_c = 1/2 \dot{q}' [A] \dot{q} \dots\dots\dots (2.26)$$

Tel que :

[A] : matrice (n x n) symétrique définie positive d'éléments génériques : $A_{ij}(q)$ dépendant du variable articulaires q .

$\dot{q} = (\dot{q}_1, \dot{q}_2, \dot{q}_3, \dots, \dot{q}_n)'$ Matrice uni colonne des vitesses généralisées.

L'énergie potentielle est due aux champs de pesanteur, alors l'effort généralisé par le champ de pesanteur sur l'articulation i s'écrit :

$$G_i = -\frac{\delta E_p}{\delta q} \dots \dots \dots (2.27)$$

E_p : Représente l'énergie potentielle externe du système.

Le Principe des puissances virtuelles donne les équations suivantes : $A_i = F_i \dots \dots \dots (2.28)$

A_i : Désigne la quantité d'accélération généralisée.

F_i : Désigne les forces généralisées.

Tel que : $A_i = \delta_i (E_c) \dots \dots \dots (2.29)$

$$\partial_i = \frac{d}{dt} \left[\frac{\partial}{\partial \dot{q}_i} \right] - \left[\frac{\partial}{\partial q_i} \right] \dots \dots \dots (2.30)$$

$$F_i = \frac{\partial E_D}{\partial q_i} - \frac{\partial E_p}{\partial q_i} + \Gamma_i \dots \dots \dots (2.31)$$

E_D : Energie de dissipation par effet du frottement visqueux.

Γ_i : Forces généralisées non conservatives.

Les équations scalaires de Lagrange peuvent se mettre sous la forme suivante :

$$\Gamma_i = \sum_{j=1}^n \left[A_{ij} \ddot{q}_j + \sum_{k=j+1}^n \left(\frac{\partial A_{ij}}{\partial q_k} + \frac{\partial A_{ik}}{\partial q_j} - \frac{\partial A_{jk}}{\partial q_i} \right) \dot{q}_j \dot{q}_k + \left(\frac{\partial A_{ij}}{\partial q_j} - \frac{1}{2} - \frac{\partial A_{ij}}{\partial q_j} \right) \dot{q}_j^2 \right] - G_i \dots \dots \dots (2.32)$$

Avec :

$$B_{ijk} = \frac{\partial A_{ij}}{\partial q_k} + \frac{\partial A_{ik}}{\partial q_j} - \frac{\partial A_{jk}}{\partial q_i} \dots \dots \dots (2.33)$$

$$C_{ij} = \left(\frac{\partial A_{ij}}{\partial q_j} - \frac{1}{2} - \frac{\partial A_{ij}}{\partial q_j} \right) \dots \dots \dots (2.34)$$

$$G_i = -\frac{\partial E_p}{\partial q_i} \dots \dots \dots (2.35)$$

Γ_i : Matrice uni colonne des forces généralisées.

[A] : Matrice carrée de dimension (n x n) symétrique définie positive. C'est la matrice de masse du système, elle intervient dans le calcul du couple / force d'inertie exprimé par le produit $[A] \cdot \ddot{q}$.

[B] : Matrice de dimension (n x (n-1) n/2), appelée matrice des termes de Coriolis.

[C] : Matrice de dimension (n x n), appelée matrice des termes centrifuges.

G : Matrice colonne de dimension (n x 1), représentant les forces généralisées aux champs de pesanteur.

$\dot{q} = (\dot{q}_1, \dot{q}_2, \dot{q}_3, \dots, \dot{q}_n)^t$ Matrice uni colonne des accélérations généralisées.

$\dot{q}\dot{q}_v = (\dot{q}_1\dot{q}_2, \dot{q}_1\dot{q}_3, \dots, \dot{q}_1\dot{q}_n, \dot{q}_2\dot{q}_3, \dots, \dot{q}_{n-1}\dot{q}_n)^t$

$\dot{q}^2 = (\dot{q}_1^2, \dot{q}_2^2, \dot{q}_3^2, \dots, \dot{q}_n^2)^t$

Les équations peuvent être regroupées sous la forme matricielle suivante:

$$\Gamma = [A]\ddot{q} + [B]\dot{q}\dot{q} + [C]\dot{q}^2 - G \dots\dots\dots(2.36)$$

Les éléments des A, B, C et G s'appellent les coefficients dynamiques du système. Ils sont des paramètres géométriques d'inertie du mécanisme.

2-10-3 Modèle dynamique générale :

Le modèle dynamique générale d'un robot manipulateur rigide à n degrés de liberté peut être représenté par un système d'équation différentielle non linéaire de second ordre à n entrée formant le vecteur de forces ou couples généralisées T, et n sorties qui forment le vecteur position q, les équations de ce système à n liaisons, décrites dans l'espace des coordonnées articulaires, sont données sous forme matricielle comme suit :

$$M(q) \ddot{q} + B(q, \dot{q})\dot{q} + K(q, \dot{q})q + G(q) + H(\dot{q}) = T_p(t) + T(t) \dots\dots\dots(2.37)$$

Avec Avec $q \in R_n$; $\dot{q} \in R_n$; $\ddot{q} \in R_n$ représente respectivement les positions, les vitesses et les accélérations articulaires et :

$M(q) \in R_{n \times n}$: Matrice d'inerte ;

$B(q, \dot{q}) + K(q, \dot{q})q \in R_n$: Englobe les couples dus aux forces de Coriolis et centrifuges ;

$G(q) \in R_n$: Vecteur de forces ou couples dus aux forces de gravitation ;

$T_p \in R_n$: Le vecteur de forces ou couples de perturbation externe ;

$T(t) \in R_n$: Vecteur de forces ou couples moteurs .

Les éléments de M, B, K, G et H sont généralement des fonctions très compliqués et non linéaires par rapport aux coordonnées généralisées de manipulateur.

Le modèle précédant du robot est complexe mais vérifie certaines propriétés fondamentales qui peuvent être exploitées pour l'analyse du comportement du système et le calcul de commande.

Propriété 1 : La matrice $M(q)$ est symétrique définie positive (S. D. P.) , par conséquent tous les éléments diagonaux de cette matrice sont positifs $\{ M_{ij}(q) > 0 \quad i = 1, \dots, n \}$

Propriété 2 : Les matrices M, B, K et les vecteurs G, H sont uniformément bornés.

Propriété 3 : L'entrée de commande est indépendante pour chaque articulation du manipulateur.

Propriété 4 : Le vecteur des frottements visqueux et secs $H(\dot{q})$ est caractérisé par les n éléments.

$$\{ H_i(\dot{q}_i) \quad i = 1, \dots, n \}, \text{ tel que : } H_i(\dot{q}_i) = f_{vi} \dot{q}_i + f_{si} \operatorname{sgn}(\dot{q}_i) \dots \dots \dots (2.38)$$

Avec f_{vi} et f_{si} sont respectivement les coefficients des frottements visqueux et secs de la i ème articulation.

Les propriétés 1-4 découlent de la nature physique du robot manipulateur. La propriété 3 est due au fait que les flexibilités des articulations et des structures n'ont pas été prises en compte.

Dans ce cas de figure chaque degré de liberté (D.D.L.) est piloté par un actionneur (moteur à courant continu).

2-10- 4 Modèle dynamique directe : Ce modèle consiste à déterminer les variables articulaires en fonction des forces (ou / et couples) généralisés.

Le calcul de ces variables se fait en résolvant le système d'équations différentielles non linéaires suivant :

$$\Gamma = [A] \ddot{q} + [B] \dot{q} \cdot \dot{q} + [C] \dot{q}^2 - G \dots \dots \dots (2.39)$$

$$\text{À } t=t_0 \quad q(t_0) = q_0, \quad \dot{q}(t_0) = 0$$

La résolution peut se faire par plusieurs méthodes numériques dans notre cas nous utilisons la méthode de Runge-Kutta à 4 approximations.

2-11 L'utilisation de la méthode de Runge -Kutta [21, 45, 46] :

$$dt [y] = f(t, y) \text{ avec } [y] = [y_0] \text{ pour } t = t_0$$

$$[y]_{t+dt} = [y]_t + 1/6 ([k_1] + 4[k_2] + [k_3] + [k_4]) \dots \dots \dots (2.40)$$

$$[K_1] = f(t, [y]) dt \dots \dots \dots (2.41)$$

$$[K_2] = f(t + dt/2, [y] + k_1/2) dt \dots \dots \dots (2.42)$$

$$[K_3] = f(t + dt/2, [y] + k_2/2) dt \dots \dots \dots (2.43)$$

$$[k_4] = f(t + dt, [y] + [k_3]) dt \dots \dots \dots (2.44)$$

Le système d'équation peut se mettre sous la forme suivante :

$$\ddot{q} = - [A]^{-1} [[B](\dot{q} \dot{q}) + [c] \dot{q}^2 - g - \Gamma] \dots \dots \dots (2.45)$$

$$T = t_0, \quad q(t_0) = q_0, \quad (\dot{q}(t_0) = 0)$$

$$[y] = \begin{pmatrix} q \\ \dot{q} \end{pmatrix} \quad f(t, y) = d[y]/dt = \begin{pmatrix} \dot{q} \\ \ddot{q} \end{pmatrix} = \begin{pmatrix} - [A]^{-1} [B] (\dot{q} \dot{q}) + [c] \dot{q}^2 - G - \Gamma \end{pmatrix} \dots (2.46)$$

2-12 Conclusion :

Pour modélisé un système, c'est-à-dire gouverner ses sorties, il faut prévoir le comportement du système, en réponse aux différentes excitations d'entrer qui pourront lui être appliquées; la démarche est de représenter le comportement du système sous la forme d'un modèle, une telle démarche s'appelle **la modélisation**; d'une manière générale, on recherche toujours le modèle le plus simple qui permet d'expliquer, de manière satisfaisante, le comportement du processus dans son domaine d'application; les modèles de transformation entre l'espace opérationnel (dans lequel est définie la situation de l'organe terminal) et l'espace articulaire. (Dans lequel est définie la configuration du robot), On distingue :

- Les modèles géométriques qui expriment la situation de l'organe terminal en fonction de la configuration du mécanisme.
- Les modèles cinématiques permettent de contrôler la vitesse de déplacement du robot afin de connaître la durée d'exécution d'une tâche.
- Les modèles dynamiques définissent les " équations du mouvement du robot qui permettent d'établir les relations entre les couples ou forces exercés par les actionneurs et positions, vitesses et accélérations.

CHAPITRE 3

LES CAUSES STRUCTURELLES

3.1 Modèles dynamiques des systèmes mécaniques poly articulés aux joints déformables

3.1.1 Descriptions d'une liaison déformable : [47, 48, 49] :

Nous supposons que la déformation du joint est supposée localisée à la sortie réductrice.

Pour chaque articulation nous prendrons deux variables articulaires q_{2i-1} et q_{2i} ou :

Q_{2i-1} : Variable articulaire délivrée par l'actionneur;

Q_{2i} : Variable articulaire prise par le segment ;

3.1.2 Hypothèses du travail:

- Les segments sont supposés parfaitement rigides;
- Les liaisons sont prismatiques ou rotoides élastiques linéaires;
- La dissipation d'énergie dans les liaisons est du type visqueux.

3.1.3 Mise en équation : [49, 50, 51, 52, 53, 54] :

Pour la mise en équation du modèle dynamique nous avons utilisées le formalisme de Lagrange associé à la méthode Uiker (même démarche que dans le chapitre 2), ce qui conduit au calcul de:

L'énergie de dissipation par frottement visqueux dans les joints $E_p = E_p(q_{2i-1}, q_{2i})$.

3.1.4 Energie cinétique, potentiel et de dissipation du système :

Pour un système mécanique articulé à joints flexibles, l'énergie cinétique est calculée en considérant ; la structure comme une chaîne ouverte simple à $2n$ éléments les n segments et les n actionneurs qu'on peut mettre sous la forme quadratique suivante :

$$E_c = \frac{1}{2} \dot{q}_{2i}^t [A] \dot{q}_{2i} + \frac{1}{2} \dot{q}_{2i-1}^t [I_a] \dot{q}_{2i-1} \dots \dots \dots \quad (3.1)$$

[A] : Représente la matrice de masse relative aux coordonnées généralisées q_{2i} . dimension $(n \times n)$ symétrique définie positive.

$[I_a]$: Matrice des inerties des actionneurs de dimension $(n*n)$ diagonal, construite par les éléments de type $N_i^2 I_a$ tel que:

N_i : Rapport de transmission du $i^{\text{ème}}$ actionneur $N_i \geq 1$

I_a : Inertie du rotor et du premier étage du réducteur du $i^{\text{ème}}$ actionneur.

Q_{2i} : Vitesse généralisée relative au $i^{\text{ème}}$ segment .

Q_{2i-1} : Vitesse généralisée relative au $i^{\text{ème}}$ segment.

L'énergie potentielle est calculée de la même façon que dans le cas rigide, mais tout considérons la structure comme une chaîne cinétique simple à $2n$ éléments, les n segments et les n actionneurs :

$$E_p = E_p(\text{ext}) + E_p(\text{int}) \dots \dots \dots (3.2)$$

Avec $E_p(\text{ext}) = E_p(\text{pesanteur})$.

$E_p(\text{int}) = E_p(\text{élastique})$.

L'énergie potentielle s'écrit sous la forme :

$$E_{p\text{int}} = \sum_{i=1}^n E_{pi} (q_{2i-1} - q_{2i}) \dots \dots \dots (3.3)$$

$$E_{p\text{int}} = \sum_{i=1}^n \frac{1}{2} K_i (q_{2i-1} - q_{2i})^2 \dots \dots \dots (3.4)$$

En posant:

$$E_{p\text{int}} = \varphi_i = q_{2i-1} - q_{2i} \dots \dots \dots (3.5)$$

$$E_{p\text{int}} = \sum_{i=1}^n \frac{1}{2} K_i (\varphi_i)^2 \dots \dots \dots (3.6)$$

K_i : Représente la constante de rigidité du $i^{\text{ème}}$ joint élastique.

φ_i : Représente le déplacement angulaire relatif au niveau du $i^{\text{ème}}$ joint :

$$E_{p\text{int}} = \frac{1}{2} \{ \varphi \}^T [T] \{ \varphi \} \dots \dots \dots (3.7)$$

$\{ \varphi \}$: Matrice uni colonne des déplacements angulaires.

$[K]$: Matrice de rigidité des joints de dimension $(n*n)$. diagonale :

$$E_{p\text{ext}} = E_{pg} = \sum_{i=1}^n E_{pg} (q_{2i}) \dots \dots \dots (3.8)$$

C'est une forme quadratique des vitesses angulaires relatives aux joints, on aura:

$$E_D = \sum_{i=1}^n E_D (\dot{q}_{2i-1}, \dot{q}_{2i-1}) \dots \dots \dots (3.9)$$

$$E_D = \sum_{i=1}^n \frac{1}{2} b_i (\dot{q}_{2i-1}, \dot{q}_{2i})^2 \dots \dots \dots (3.10)$$

$$E_D = \sum_{i=1}^n \frac{1}{2} b_i (\dot{\varphi})^2 \dots \dots \dots (3.11)$$

$$E_D = \frac{1}{2} \{ \dot{\varphi} \}^T [D] \{ \dot{\varphi} \} \dots \dots \dots (3.12)$$

{\varphi} : Matrice uni colonne constituée des vitesses angulaires.

[D]: Matrice des coefficients d'amortissement dimension (n*n) diagonale.

Si nous appliquons le principe des puissances virtuelles et le formalisme de Lagrange conduisent aux 2n équations suivantes:

$$\frac{d}{dt} \left[\frac{\partial E_c}{\partial \dot{q}_j} \right] - \frac{\partial E_c}{\partial q_j} = - \frac{\partial E_D}{\partial q_j} - \frac{\partial E_p}{\partial q_j} + \Gamma_j \quad j = 1, 2, \dots, n \dots \dots \dots (3.13)$$

$$\left\{ \begin{array}{l} \frac{d}{dt} \left[\frac{\partial E_c}{\partial \dot{q}_i} \right] - \frac{\partial E_c}{\partial q_i} + \frac{\partial E_{ps}}{\partial q_i} - K_i (q_{2i} - q_{2i-1}) - b_i (q_{2i} - q_{2i-1}) = \Gamma_i \quad \text{si } i = 2i-1 \\ \frac{d}{dt} \left[\frac{\partial E_c}{\partial \dot{q}_i} \right] - \frac{\partial E_c}{\partial q_i} + \frac{\partial E_{ps}}{\partial q_i} + K_i (q_{2i} - q_{2i-1}) - b_i (2q_i - q_{2i-1}) = 0 \quad \text{si } j = 2i \dots \dots \dots \end{array} \right. (3.14)$$

On posant la variable articulaire [q]= telle que :

[q_i]=[q_{2i-1}] et [q₂]=[q_{2i}] i=1,n.

On aura le système d'équation suivant :

$$\left\{ \begin{array}{l} [A] \ddot{q}_1 + [B] \dot{q}_1 \dot{q}_1 + [C] \dot{q}_1^2 - G + [K] (q_1 - q_2) + [D] (\dot{q}_1 - \dot{q}_2) = 0 \\ [I_a] \ddot{q}_2 - [K] (q_1 - q_2) - [D] (\dot{q}_1 - \dot{q}_2) = \Gamma \end{array} \right. \dots \dots \dots (3.15)$$

[A] : Matrice de masse.

[B] : Matrice de Coriolis.

G : Matrice uni colonne des termes de gravité.

Si le système mécanique est à joints parfaitement rigides.

Le coefficient de rigidité K → ∞

B → 0, q₁ → q₂ et q₁-q₂ → 0.

Le système d'équation devient alors :

$$\left\{ \begin{array}{l} [A_s] \ddot{q}_1 + [B] \dot{q}_1 \dot{q}_1 + [C] \dot{q}_1^2 - G = 0 \\ [I_a] \ddot{q}_1 = \Gamma \end{array} \right. \dots \dots \dots (3.16)$$

Tel que $[A_s]$ est la matrice d'inertie de la partie segments et $[I_a]$ est la matrice d'inertie de la partie actionneurs.

Si on additionne membre à membre les équations (1) et (2) on retrouve les équations du système poly articulés indéformables données sous la forme matricielle suivante :

$$\Gamma = [A] \ddot{q} + [B] \dot{q} \cdot \dot{q} + [C] \dot{q}^2 - G \quad \text{Sachant que } [A] = [A_s] + [I_a].$$

3.1.5 Résolution des équations: [4 , 49, 68, 69] :

Pour la résolution du système d'équations on utilise la méthode de Runge - Kutta à quatre approximations pour cela on arrange le système comme suit :

$$\begin{cases} \ddot{q}_1 = - [A]^{-1} [B] \dot{q}_1 \dot{q}_1 + [C] \dot{q}_1^2 - G + [K] (q_1 - q_2) + [D] (\dot{q}_1 - \dot{q}_2) \\ \ddot{q}_2 = [I_a]^{-1} [\Gamma + [K] (q_1 - q_2) + [D] (\dot{q}_1 - \dot{q}_2)] \end{cases} \dots\dots\dots(3.17)$$

Les conditions initiales sont :

$$\begin{aligned} q_1(t_0) = q_{1_0} & \quad \dot{q}_1(t_0) = \dot{q}_{1_0} = 0 \\ q_2(t_0) = q_{2_0} & \quad \dot{q}_2(t_0) = \dot{q}_{2_0} = 0 \end{aligned}$$

on pose

$$[Y] = \begin{bmatrix} q_1 \\ q_2 \\ \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad \text{Et donc nous avons :} \quad \frac{d}{dt} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_2 \end{bmatrix}$$

$$\frac{d[Y]}{dt} = \frac{d}{dt} \begin{bmatrix} q_1 \\ \dot{q}_1 \\ q_2 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} \dot{q}_1 \\ -[A]^{-1} \dot{q}_1 + [B] \dot{q}_1 \dot{q}_1 + [C] \dot{q}_1^2 - G + [K](q_1 - q_2) + [D](\dot{q}_1 - \dot{q}_2) \\ \dot{q}_2 \\ [I_a]^{-1} [\Gamma + [K](q_1 - q_2) + [D](\dot{q}_1 - \dot{q}_2)] \end{bmatrix} \dots\dots\dots(3.18)$$

Qu'on peut poser sous la forme suivante :

$$\frac{d}{dt} [Y] = f(t, [Y])$$

Sachant que $[Y(t_0)] = [Y_0]$ on obtient le système d'équation différentielle suivant :

$$\begin{cases} \frac{d}{dt} [Y] = f(t, [Y]) \\ [Y] = [Y_0] \text{ pour } t = t_0 \end{cases}$$

Qu'on peut résoudre par la méthode de Runge -Kutta .

3.2 Les erreurs statique et dynamique

3.2.1 Les erreurs de positionnement d'un manipulateur [55, 65] :

Le positionnement réel d'un manipulateur, que ce soit en statique ou en dynamique (poursuit des trajectoires) s'écarte inévitablement et pour différentes causes, de sa position désirée. De même, un manipulateur ne se positionne jamais au même endroit lorsque la même trajectoire est répétée plusieurs fois ».

Ces erreurs de positionnement sont d'origines diverses, peuvent être classées en quatre catégories principales :

- Erreurs de quantification et de calcul.
- Erreurs cinématique de type aléatoire dominant.
- Erreurs d'étalonnage et d'identification.
- Erreurs cinématique de type systématique dominant.

Les erreurs de quantification et de calcul :

sont liées à :

- L'arrondi dans les calculs effectués par le calculateur.
- La quantification des codeurs incrémentaux ou absolus utilisés pour la détermination des coordonnées articulaires.
- La performance du calcul numérique et des algorithmes employés.
- La troncature des valeurs numériques dans les calculs trigonométriques

Les erreurs cinématique de type aléatoire dominant:

Chacune des articulations d'un manipulateur à motorisation électrique peut être déplacée à l'intérieur d'une petite zone, sans entraîner de signale d'erreur à l'intérieur du système d'asservissement. Cette zone morte est due aux défauts géométriques non systématiques des organes mécaniques (les jeux mécaniques) à la résolution des capteurs et aux performances des asservissements (non linéarités dues aux frottements mécaniques par exemple) [10, 65].

Les erreurs cinématiques de types systématique dominante sont liées à :

- La déformation des segments sous l'effet des charges statiques et dynamiques.
- La déformation des articulations sous l'effet des charges statiques et dynamique.
- La dilatation des pièces mécaniques de la structure sous l'effet de la température.

3.2.2 Identification des paramètres d'un manipulateur :

L'identification consiste à déterminer, suite à une série de mesures et à l'aide des méthodes de masse du manipulateur [55,66].

Les erreurs de positionnement statique d'un manipulateur sont de deux natures : Géométrique et non géométrique.

Les erreurs géométriques : Elles regroupent les imprécisions de fabrication dans les corps et les liaisons et les erreurs d'initialisations des offsets codeurs (les valeurs des offsets codeurs correspondants à la configuration initiale géométrique dans laquelle les variables articulaires sont nulles).

Les erreurs non géométriques : Elles regroupent les déformations, les jeux dans les chaînes cinématiques, les erreurs liées à la résolution des capteurs et aux performances des asservissements. Elles ne sont pas accessibles à la calibration géométrique.

La procédure d'identification géométrique distingue trois niveaux de complexité :

Niveaux 1 : Ou « calibration des articulations », l'objectif est d'établir la relation la plus exacte entre le signal produit par les capteurs de position et les déplacements articulaires. Ceci, implique généralement la calibration de la cinématique des organes d'entraînement (réducteurs, courroies,.....etc.), les mécanismes des valeurs des offsets codeurs.

Niveaux 2 : Ou « calibration géométrique globale » dans ce niveau on doit identifier tous les paramètres géométriques de description du manipulateur. L'objectif de ce niveau est de déterminer le modèle géométrique de base, qui lie les coordonnées opérationnelles aux coordonnées articulaires (ou valeurs de commande des actionneurs).

Niveaux 3 : Ou « calibration non géométrique », il porte sur les possibilités de compenser les erreurs d'ordre non géométriques à savoir les déformations des articulations et des segments et des frottements.

Pour utiliser le modèle dynamique, il faut connaître les valeurs numériques des paramètres de masse (masse, centre de masse, et matrice d'inertie) relatifs aux différents corps et qui interviennent dans le modèle dynamique. Plusieurs techniques peuvent être envisagées pour estimer ces paramètres : Soit par calcul au moment de la conception, surtout si on utilise un logiciel de C.A.O. performant, soit par mesure corps par corps avant le montage ou par identification.

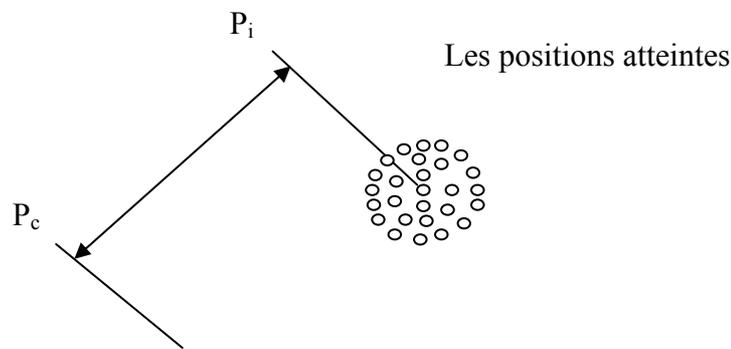
La technique d'identification des paramètres, consiste à exploiter le caractère linéaire des actions dynamiques relativement à ces paramètres pour les identifier, en utilisant la méthode d'optimisation des moindres carrés[56] .

La précision d'un manipulateur est généralement définie en terme de précision statique (les caractéristiques de pose) et de précision dynamique (les caractéristiques de trajectoire), ces deux caractéristiques, quantifient la différence entre la situation désirée et celle réellement atteinte [10, 65, 67].

Pour les manipulateurs utilisés dans les tâches d'assemblages, d'insertion de composants, de soudage par point..., les caractéristiques de composants sont les caractéristiques de pose. Ces caractéristiques sont définies par trois termes:

- Exactitude statique.
- Répétitivité statique.
- Temps de stabilité et dépassement statique.

Selon la définition ISO, l'exactitude statique est représentée par l'écart entre une pose commandée (P_c) et la moyenne des poses réellement atteintes (figures 3.1), lorsqu'on demande au manipulateur de se positionner plusieurs fois en P_c en suivant toujours la même trajectoire (exactitude statique unidirectionnelle) soit P_c le point de la position commandée de coordonnées X_c , Y_c , Z_c dans le référentiel de base $\{R_0\}$; P_i une des positions réellement atteintes ($i=1, \dots, k$), $P_c = [X_c, Y_c, Z_c]$ et $P_i = [X_i, Y_i, Z_i]^T$



La position programmées

Figure 3.1: L'exactitude de positionnement statique d'un manipulateur [55].

L'exactitude locale de position E_s est la distance entre le point P_c et le point P_g barycentre de tous les points atteints P_i donc:

$$E_s = \sqrt{\left(\sum_{i=1}^k x_i - x_c\right)^2 + \left(\sum_{i=1}^k y_i - y_c\right)^2 + \left(\sum_{i=1}^k z_i - z_c\right)^2} \dots\dots\dots(3.19)$$

$$\text{avec: } X_g = \frac{\sum_{i=1}^k x_i}{k}; \quad Y_g = \frac{\sum_{i=1}^k y_i}{k}; \quad Z_g = \frac{\sum_{i=1}^k z_i}{k}$$

Pour les rotations, on peut définir de la même manière la variation entre la moyenne des positions angulaires atteintes et la valeur commandée.

Par définition la répétitivité statique unidirectionnelle est la distance maximale entre le point moyen P_g et les points réellement atteints P_i , $i= 1 \dots\dots\dots k$ (Figure 3.2)

Les positions atteintes.

R_s

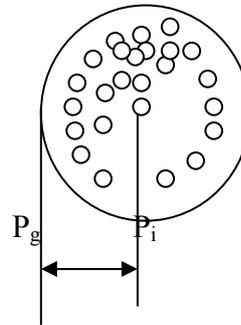


Figure 3.2: La répétitivité de positionnement statique d'un manipulateur [55].

$$R_s = \text{Max}_{i=1 \dots\dots\dots k} \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2 + (z_k - z_i)^2} \dots\dots\dots (3.20)$$

La norme (NF-E 61-103) définit également une répétitivité statique à partir de la moyenne de distances entre P_g et les points atteints P_i .

Si les trajectoires pour atteindre la position désirée P_c sont différentes (on partira de différentes positions vers la même destination). L'exactitude et la répétitivité vont avoir des valeurs différentes appelées exactitudes et répétitivité multidirectionnelles.

Dans le cas où la position commandée serait définie par apprentissage, la répétitivité est la même, par contre l'exactitude est très différente, pour les manipulateurs très précis la répétitivité peut être de l'ordre du centième de mm, alors que l'exactitude de position programmée peut être de plusieurs mm.

Remarque: Au terme répétitivité on associe souvent la notion de réversibilité qui caractérise la précision statique quant le point est atteint selon plusieurs directions ; elle est plus mauvaise que la répétitivité.

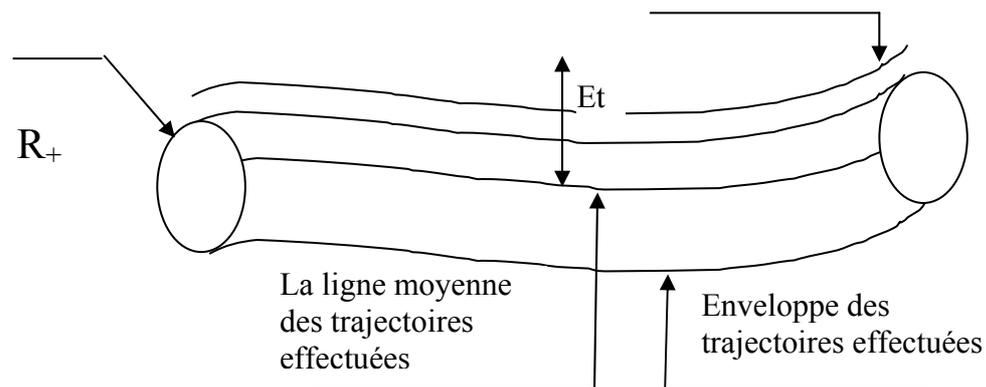
Il est bien intéressant de connaître le comportement du robot lorsqu'il approche une pose commandée. Suivant le réglage des asservissements des actionneurs et le niveau de déformations des segments, le manipulateur peut osciller, dépasser la situation commandée ou au contraire s'en approcher sans oscillation, ce comportement se traduit par deux caractéristiques : Dépassement et temps de stabilisation ces caractéristiques peuvent être utiles pour régler une temporisation du manipulateur avant d'effectuer une tâche ; la

connaissance du dépassement permet de s'assurer que l'espace dégagé autour du point d'arrêt est suffisant pour éviter les collisions de l'outil avec l'environnement [65].

En robotique, de nombreuses tâches sont réalisées en utilisant un mode de commande de type interpolation linéaire ou circulaire afin de quantifier les défauts et les écarts entre la trajectoire réellement parcourue et la trajectoire commandée, les caractéristiques essentielles pour ces applications sont définies par les notions d'exactitudes et de répétitivité de trajectoire.

Elle caractérise l'aptitude d'un manipulateur à faire suivre à l'interface mécanique (l'effecteur) une trajectoire désirée le fois dans la même direction et le fois dans la direction opposée [10].

L'exactitude de trajectoire est définie par la distance maximale entre la ligne moyenne des trajectoires réellement atteintes et la ligne programmée (Figure 3.3), mesurée dans un plan orthogonal à la trajectoire.



Trajectoire programmée

Figure 3.3: Exactitude de trajectoire [55]

L'exactitude pour des positions ou des trajectoires programmées dépend surtout des erreurs d'étalonnage et des erreurs dues aux déformations de la structures mécanique du manipulateur [10, 65, 70].

La répétitivité de trajectoire (ou dynamique) est la distance moyenne (ligne des barycentres des trajectoires effectuées) et une trajectoire effectuée; sera donc le rayon du tore qui contiendrait toutes les trajectoires effectuées (Figure 3.3).

Les erreurs de répétitivité sont dues aux défauts géométriques non systématiques des organes mécaniques, à la résolution des codeurs de positions et aux performances des asservissements (non linéarités dues aux frottements mécaniques par exemple)

[10, 65] avec le développement de l'électronique, de l'informatique et des moyens de fabrication (machine à commande numérique, CFAO) la précision géométrique en terme de répétitivité tend à devenir largement suffisant pour la majorité des applications industrielles.

Les applications industrielles de la robotique font appel à des modes d'emploi principaux des manipulateurs:

De point d'arrêt acquis par apprentissage d'une suite discrète de configuration articulaires.

- De trajectoires acquises par l'enregistrement échantillonné d'une suite continue de configurations articulaires correspondants aux mouvements que doit reproduire le manipulateur.

Les points d'arrêt sont seuls fonctionnels (travail en cours mouvement).

Dans ces quatre types d'emploi, le problème de précision prend des dimensions tout à fait différentes:

- Il peut dépendre de la répétitivité de la réversibilité, de l'exactitude statique ou dynamique (§ 3.3) ou de l'un de ces critères seulement.

Les déformations de la structure mécanique peuvent être tout à fait transparentes à l'utilisateur comme elles peuvent s'imposer comme un aspect du comportement dont la modélisation et la prise en charge est vital pour l'application.

3-2-3 Evaluation des erreurs de positionnement ou de poursuite de trajectoire [55] :

La simulation de l'effecteur dans l'espace opérationnel est donnée par le vecteur x . Si on désigne par:

X_D : la situation instantanée de l'effecteur correspond à la configuration déformée du manipulateur.

X_R : La situation de l'effecteur correspond à la configuration non déformée du manipulateur.

L'erreur de positionnement ou de poursuite de trajectoire instantané sera donnée par:

$$\delta X = X_D - X_R \dots\dots\dots(3.21)$$

La situation X_R de l'effecteur est donnée par le modèle géométrique direct, dans l'hypothèse rigide obtenu par le produit des matrices de transformations homogènes

Ces matrices sont calculées sur la base des paramètres de descriptions géométrique de la topologie du manipulateur.

Les déformations d'un segment (segment C_{j-1}) engendre un torseur des déplacements au point de l'articulation en aval (point O_i) du segment et se traduisent par une variation de la

transformation relative entre deux repères consécutifs (le repère $\{R_{j-1}\}$ et le repère $\{R_j\}$) le modèle d'évaluation de la situation X_D de l'effecteur peut être donc obtenu par un modèle géométrique modifié basé sur des matrices de transformations homogènes corrigées pour prendre en compte l'effet des déformations sur la situation de l'effecteur .

3-2-4 Le modèle géométrique direct corrigé [55]

La figure 3.4 représente le corps C_{j-1} du manipulateur dans la configuration rigide puis en position déformée et les deux référentiels $\{R_{j-1}\}$ et $\{R_j\}$ associés aux articulateurs A_{i-1} et A_i respectivement.

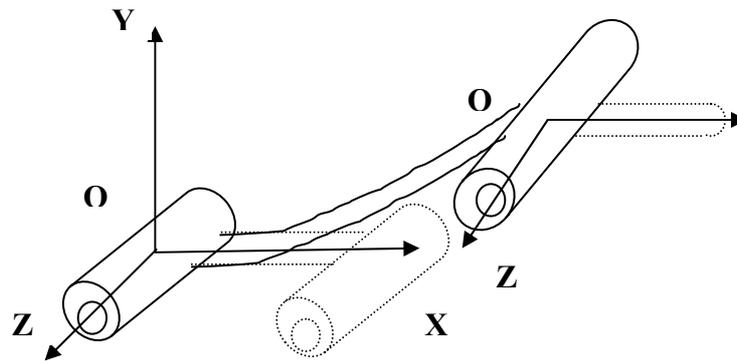


Figure : 3.4 Flexibilité d'un segment du manipulateur [55]

Le passage de $\{R_{j-1}\}$ à $\{R_j\}$ s'exprime en fonction des composants du vecteur des déplacements dus aux déformations du corps C_{j-1} et des trois paramètres de descriptions α_j , L_j , et θ_j .

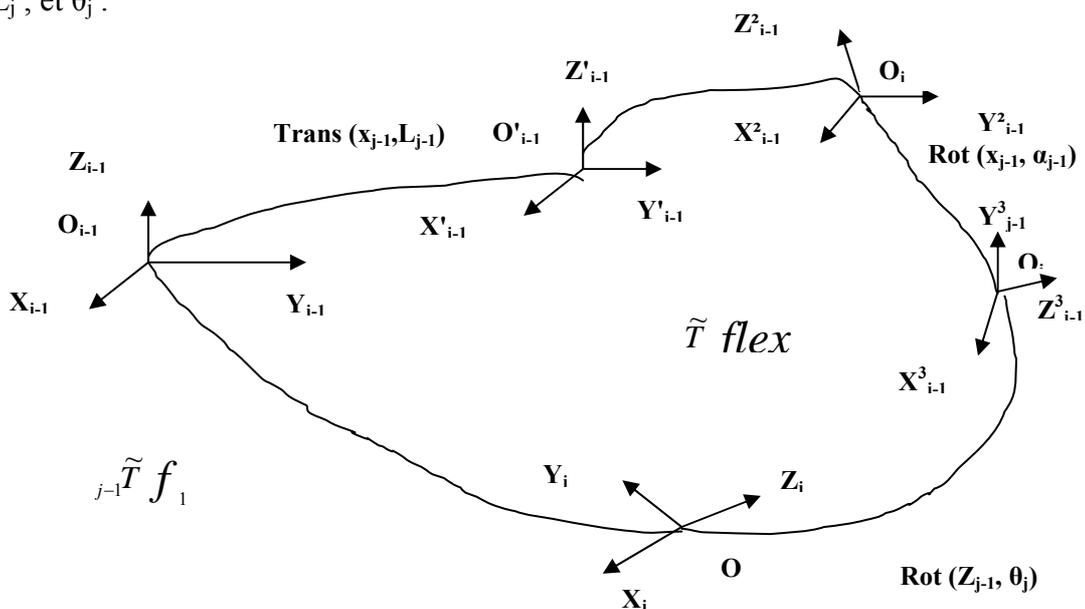


Figure : 3.5 : La transformation définissant le repère $\{R_j\}$ dans le repère $\{R_{j-1}\}$ [55]

La manière de transformation définissant le repère $\{R_j\}$ dans le repère $\{R_{j-1}\}$ Figure 3.5 est donnée par:

$${}_{j-1}\tilde{T} f_j = Trans(X_{j-1}, L_{j-1}) * \tilde{T} flex_{j-4} * Rot(x_{j-1}, \alpha_{j-1}) * Rot(z_j, \theta_j) \dots \dots \dots (3.22)$$

Où:

$$Rot(x_{j-1}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_{j-1} & -S\alpha_{j-1} & 0 \\ 0 & S\alpha_{j-1} & C\alpha_{j-1} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \dots \dots \dots (3.23)$$

$$Trans(x_{j-1}, L_{j-1}) = \begin{pmatrix} 1 & 0 & 0 & L_{j-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \dots \dots \dots (3.24)$$

$$Rot(z_j, \theta_j) = \begin{pmatrix} C\theta_j & -S\theta_j & 0 & 0 \\ S\theta_j & C\theta_j & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \dots \dots \dots (3.25)$$

La matrice de transformation homogène associée aux déplacements $\tilde{T}flex_{j-1}$ (Translations et rotations) dus aux déformations du corps C_{j-1} . Cette matrice peut être décomposée en deux matrices:

Une matrice de transformation pure qui représente les translations dus aux déformations $dx_{j-1}, dy_{j-1}, dz_{j-1}$ donnée par :

$$\tilde{T}flex_i = \begin{pmatrix} 1 & 0 & 0 & dx_{j-1} \\ 0 & 1 & 0 & dy_{j-1} \\ 0 & 0 & 1 & dz_{j-1} \\ 0 & 0 & 0 & 1 \end{pmatrix} \dots \dots \dots (3.26)$$

Représente les rotations dues aux déformations $\tilde{T}flexr$ et une matrice de rotation pure $r_{x_{j-1}}, r_{y_{j-1}}, r_{z_{j-1}}$ de la section droite d'abscisse b_{i-1} au corps C_{i-1} point O_{i-1} par rapport à l'extrémité (point O_{j-1}), autour des trois axes $x_{j-1}, y_{j-1}, z_{j-1}$ respectivement les matrices de transformation homogène associées à ces rotations, sont donnée par:

$$\text{Rot}(r_{x_{j-1}}, x_{j-1}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & C_x & -S_x & 0 \\ 0 & S_x & C_x & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \dots\dots\dots(3.27)$$

$$\text{Rot}(r_{y_{j-1}}, y_{j-1}) = \begin{pmatrix} C_y & 0 & S_y & 0 \\ 0 & 1 & 0 & 0 \\ -S_y & 0 & C_y & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \dots\dots\dots (3.28)$$

$$\text{Rot}(r_{z_{j-1}}, z_{j-1}) = \begin{pmatrix} C_z & -S_z & 0 & 0 \\ S_z & C_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \dots\dots\dots(3.29)$$

avec: $C_x = \cos(r_{x_{j-1}})$, $C_y = \cos(r_{y_{j-1}})$, $C_z = \cos(r_{z_{j-1}})$

$S_x = \sin(r_{x_{j-1}})$, $S_y = \sin(r_{y_{j-1}})$, $S_z = \sin(r_{z_{j-1}})$

Les rotations rx_{j-1} , ry_{j-1} , et rz_{j-1} sont des rotations infinitésimales, on peut écrire:

$$C_x \cong 1 \quad S_y \cong ry_{j-1} \quad S_x \cong rx_{j-1} \quad C_z \cong 1, \quad C_y \cong 1,$$

$$S_z \cong rz_{j-1}$$

En remplaçant $C_x, C_y, C_z, S_x, S_y, S_z$ par leurs expressions et en négligeant les termes d'ordre deux et d'ordre trois devant les termes d'ordre un [62]. On trouve:

$$\text{Rot}(r_{x_{j-1}}, x_{j-1}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -rx_{j-1} & 0 \\ 0 & rx_{j-1} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \dots\dots\dots(3.30)$$

$$\text{Rot}(ry_{j-1}, y_{j-1}) = \begin{pmatrix} 1 & 0 & ry_{j-1} & 0 \\ 0 & 1 & 0 & 0 \\ -ry_{j-1} & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \dots\dots\dots(3.31)$$

$$\text{Rot}(rz_{j-1}, z_{j-1}) = \begin{pmatrix} 1 & -rz_{j-1} & 0 & 0 \\ rz_{j-1} & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \dots\dots\dots(3.32)$$

La matrice homogène est obtenue par le produit des matrices de transformation $\tilde{T}flex_{i-1}$ $\text{Rot}(rx_{j-1}, x_{j-1})$, $\text{Rot}(ry_{j-1}, y_{j-1})$, et $\text{Rot}(rz_{j-1}, z_{j-1})$, on peut effectuer ce produit dans n'importe quel ordre, En faisant abstraction des valeurs infinitésimales de deuxième ordre de type $r_{x_{j-1}} \times r_{y_{j-1}}$, et $r_{y_{j-1}} \times r_{z_{j-1}}$ on obtient :

$$\tilde{T}flex = \begin{pmatrix} 1 & r_{z_{j-1}} & r_{y_{j-1}} & 0 \\ r_{z_{j-1}} & 1 & -r_{x_{j-1}} & 0 \\ -r_{y_{j-1}} & r_{x_{j-1}} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \dots\dots\dots(3.33)$$

La matrice de transformation homogène associée aux déplacements dus aux déformations du corps C_{j-1} (la matrice de flexibilité) est donnée donc par:

$$\Rightarrow \tilde{T}flex_{j-1} * \tilde{T}flex_{j-1} = \tilde{T}flex_{j-1}$$

$$\tilde{T}flex_{j-1} = \begin{pmatrix} 1 & r_{z_{j-1}} & r_{y_{j-1}} & d_{x_{j-1}} \\ r_{z_{j-1}} & 1 & -r_{x_{j-1}} & d_{y_{j-1}} \\ -r_{y_{j-1}} & r_{x_{j-1}} & 1 & d_{z_{j-1}} \\ 0 & 0 & 0 & 1 \end{pmatrix} \dots\dots\dots(3.34)$$

Soit $R_E (O_E, x_E, y_E, z_E)$ un repère lié à l'effecteur, la position et l'orientation de l'effecteur (l'outil) correspondant à la configuration déformée du manipulateur, sont définies dans le repère $\{R_o\}$ par la matrice de transformation :

$${}^n\tilde{T}_E \quad {}^{n-1}\tilde{T}_n \quad \dots \quad {}^1\tilde{T}_2 \quad {}^0\tilde{T}_1 = {}^0\tilde{T}_E \quad \dots \quad (3.35)$$

Le modèle géométrique direct corrigé du manipulateur est l'ensemble des relations qui permettent d'exprimer la situation de l'effecteur qui correspond à la configuration déformée du manipulateur X_d en fonction des coordonnées articulaires q et des vecteurs des déplacements dus aux déformations des différents segments: [63]

$$X_d = F_d(q, u) \quad \dots \quad (3.36 a)$$

Les trois premières composantes de X_j fixent la position du point O_E de l'effecteur par rapport au repère ($\{R_E\}/\{R_o\}$).

3-2-4-1 Procédure et modèle de compensation :

Supposons que le manipulateur est dans une configuration q bien déterminée, les erreurs dues aux déformations des segments correspondant à cette configuration sont représentées par le vecteur δ_X , la correction ou la compensation de ces erreurs port de l'idée schématisée ci-après:

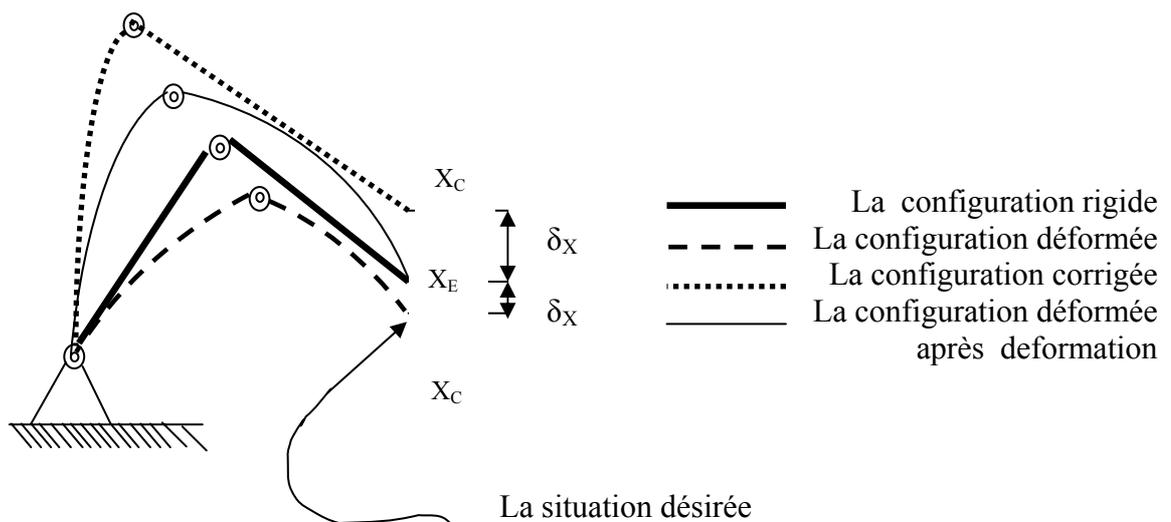


Figure 3.6: compensation des erreurs des déformations [55]

On commande le déplacement du manipulateur sur une situation $X_C = X_R - \delta_X$
(La situation de l'effecteur qui correspond à la configuration corrigée et non déformée du manipulateur) dans l'espace de travail de telle sorte que:

$$X_{ed} \approx X_C + \delta_X = X_E - \delta_X + \delta_X = X_E \dots\dots\dots(3.36 \text{ b})$$

La situation que doit atteindre le manipulateur rigide permet de calculer; si elles existent les solutions articulaires possibles.

Parmi ces solutions, il existe une solution définie dans l'espace articulaire par le vecteur des coordonnées généralisées $(q + \delta_q)$ qui est la plus indiquée pour réaliser la tâche car elle se traduit par de légères corrections sur les variables de commande relativement à l'hypothèse rigide.

Ainsi au lieu de traiter le problème par la résolution du modèle géométrique inverse, il est plus pratique d'utiliser le modèle différentiel:

$$-\delta X = \tilde{J}(q) \delta q \Rightarrow \delta q = -\tilde{J}^{-1}(q) \delta X \dots\dots\dots(3.37)$$

$$q_c = q + \delta q \dots\dots\dots(3.38)$$

δ_q : Représente le mouvement correctif à effectuer au niveau des variables articulaires, à partir de la configuration rigide.

q_c : Les coordonnées articulaires corrigées permettant la compensation des erreurs dues aux déformations élastiques des segments.

L'organigramme figure 3.7 présente, de manière chronologique les étapes de calcul nécessaire à l'évaluation et à la compensation des erreurs de positionnement ou de poursuite dues aux déformations quasi-statiques des segments d'un manipulateur en fonction des paramètres cinématiques (coordonnées, vitesses, et accélérations généralisées) et de charge.

REMARQUES:

Pour un manipulateur à moins de six degrés de liberté, plan par exemple, il est bien évident que les mouvements correctifs $-\delta_q$ ne pourront jamais compenser une erreur qui se produirait hors plan. Donc dans la relation (3.37), on ne tiendra pas compte les composantes $d_e - \delta_X$ qui ne peuvent pas être engendrées par $-\delta_q$.

Si l'écart $-\delta_X$ est important, on ne peut pas compenser l'erreur de positionnement d'un seul coup, mais il faut faire la compensation par plusieurs itérations.

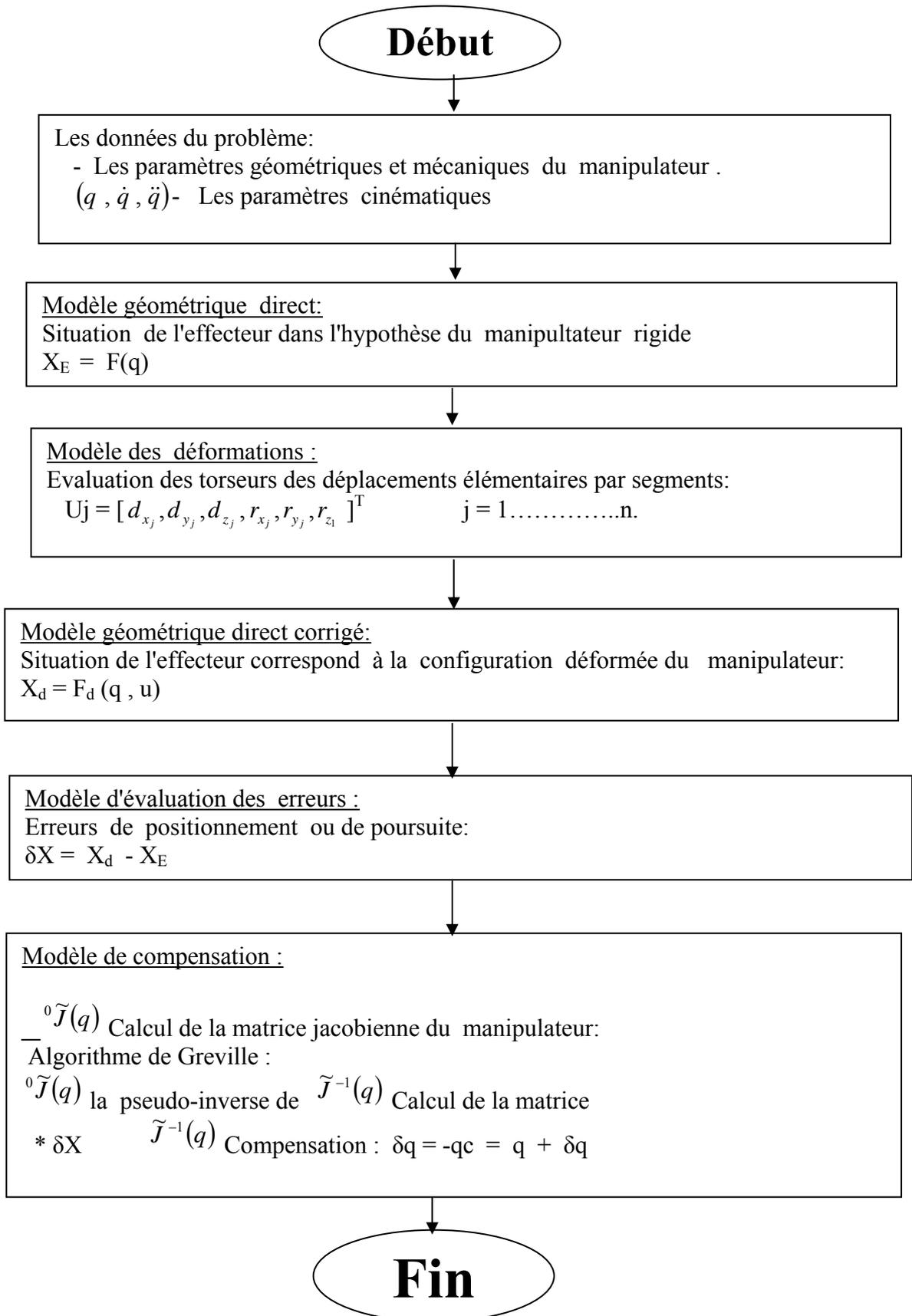


Figure 3.7 : Organigramme d'un modèle d'évaluation et de compensations des erreurs dues aux déformations élastiques des segments [55].

3-2-4-2 Calibration et déformation des manipulateurs :[56,66] :

L'étalonnage ou l'identification géométrique consiste à déterminer, suite à une série de mesures et à l'aide d'un modèle mathématique basée sur un modèle géométrique de description du manipulateur et des offsets codeurs.

Les valeurs des paramètres géométriques et des offsets codeurs identifiés au cours de l'étalonnage sont des valeurs optimisées, l'optimisation est itérative est basé sur la méthode des moindres carrés. Les paramètres géométriques identifiés, selon les procédures classiques de calibration, intégrant moyennement les effets des déformations.

Manipulateur non calibré, l'écart de positionnement entre la situation mesurée et la situation désirée pour une configuration q donné, dépend surtout des erreurs dues aux déformations :

$$X_{\text{mes}}(q) - X_{\text{thé}}(q) = \tilde{H}(q) * (P_{g \text{ réel}} - P_{g \text{ nom}}) + \Delta X_{\text{def}}(q) \dots\dots\dots(3.39)$$

Avec : $X_{\text{mes}}(q)$ = la situation mesurée.

$X_{\text{thé}}(q)$ = la situation théorique

$\tilde{H}(q)$ = la matrice d'identification

$P_{g \text{ réel}}$ = les paramètres géométriques réels du manipulateur

$P_{g \text{ nom}}$ = les paramètres géométriques nominaux du manipulateur

Il est suffisant d'intégrer les erreurs dues aux déformations dans les valeurs identifiées des paramètres géométriques de description, la procédure d'identification est celle qui est classiquement utilisée:

$$X_{\text{mes}}(q) - X_{\text{thé}}(q) = \tilde{\pi}(q) \times (P_{g \text{ thé}} - P_{g \text{ nom}}) \dots\dots\dots(3.40)$$

Avec : $P_{g \text{ thé}}$: les valeurs des paramètres géométriques et offsets codeurs identifiés.

Dans les cas, des manipulateurs flexibles ou applications exigeantes en précision. La calibration géométrique préalable n'est pas en mesure de rendre compte, à elle seule du problème des déformations. La compensation est incontournable et un modèle des déformations est donc nécessaire.

CHAPITRE 4

APPLICATIONS

4.1 Introduction [2]:

Java est un langage de programmation, fortement inspiré des langages C et C++, et fait partie des langages orientés objets.

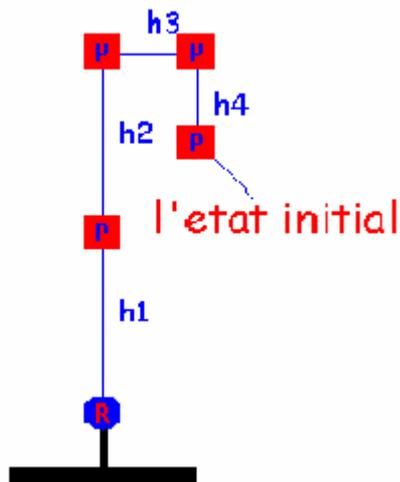
Dans ce chapitre je vais présenter des modélisations développés en java simulent la géométrie, cinématique et la dynamique directe et inverse d'un manipulateur [Algérie-Machines Outils -1].

Le programme est un système développé en créant le projet robot avec les interfaces (figure 4.6), (figure 4.7), (figure 4.8), (figure 4.9) et (figure 4.10) montrent un exécutable dans sa forme la plus simple ; c'est un modélisateur pour robots mécaniques, la modélisation présente une interface graphique qui permet avec les paramètres géométriques de manipuler chaque lien du robot. Le but principal de ce modélisateur est la simulation des robots industriels employés sur le marché. Il a été développé à l'université de Saad Dahleb de Blida faculté des sciences de l'ingénieur département de génie mécanique, il est basé sur le projet de recherche nationale: ROBOT, on peut obtenir une visualisation qui exprimait le circuit travail du robot type [ALG.- M. O.-1] dans une station d'usinage dans un environnement deux dimensions.

4.2 Exemple d'application et de validation:

Lors du développement de ce système je suis passé par plusieurs étapes, chacune d'elles m'a donné un résultat que je l'ai évalué selon mes besoins. Le système spatial du robot mis en application possède cinq articulations, une rotoïde et quatre prismatique dont le schéma cinématique est définit dans le corps du mémoire , les paramètres géométrique et inertiels sont résumés dans les appendices, j'ai déterminé les matrices de passage associées à chaque repère. J'ai considéré les éléments du système comme étant rigides que le mouvement s'effectue sans frottements.

Dans cette section je vais exposer un ensemble de résultats obtenus pendant le développement de ce système. Dans la première position, le but (besoin) était la localisation des chaînons en deux dimensions (2D) sur l'écran, pour cela la première position conçue était l'état initial. La figure 4.1 montre la position du robot au repos.

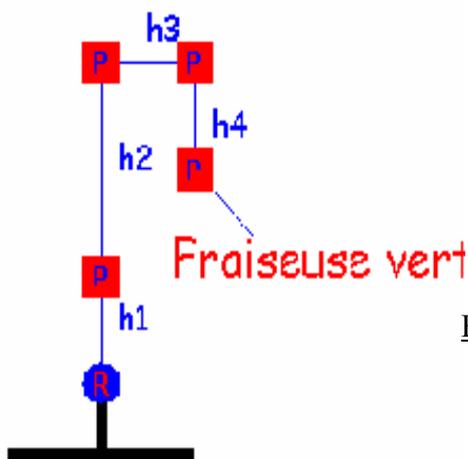


$$x_{p_1} = 200, x_{p_2} = 200, x_{p_3} = 250, x_{p_4} = 250$$

$$y_{p_1} = 250, y_{p_2} = 150, y_{p_3} = 150, y_{p_4} = 200$$

Figure 4.1 : Position 1, (l'état initial)(Dimension en centimètre).

Dans la deuxième position, le but (besoin) était la localisation des chaînons en deux dimensions (2D) sur l'écran, pour cela la deuxième position conçue était la phase de fraisage (fraiseuse verticale), La figure 4.2 montre la position du robot sur la fraiseuse verticale.



$$x_{p_1} = 200, x_{p_2} = 200, x_{p_3} = 250, x_{p_4} = 250$$

$$y_{p_1} = 300, y_{p_2} = 200, y_{p_3} = 200, y_{p_4} = 250$$

Figure 4.2 : Phase de fraisage

Dans la troisième position, le but (besoin) était la localisation des chaînons en deux dimensions (2D) sur l'écran, pour cela la troisième position conçue était la phase de fraisage (fraiseuse universel), La figure 4.3 montre la position du robot sur la fraiseuse universel.

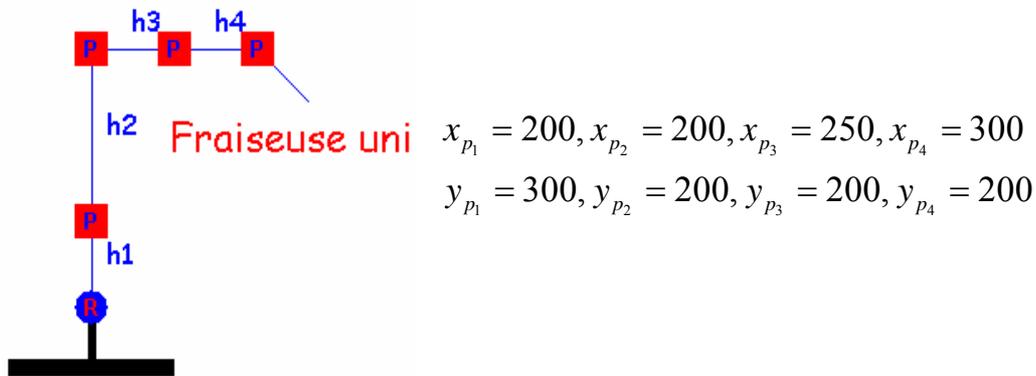


Figure 4.3 : Phase de fraisage (fraiseuse universelle) (Dimensions en centimètre).

Dans la quatrième position, le but (besoin) était la localisation des chaînons en deux dimensions (2D) sur l'écran, pour cela la quatrième position conçue était la phase de perçage La figure 4.4 montre la position du robot sur la perceuse.

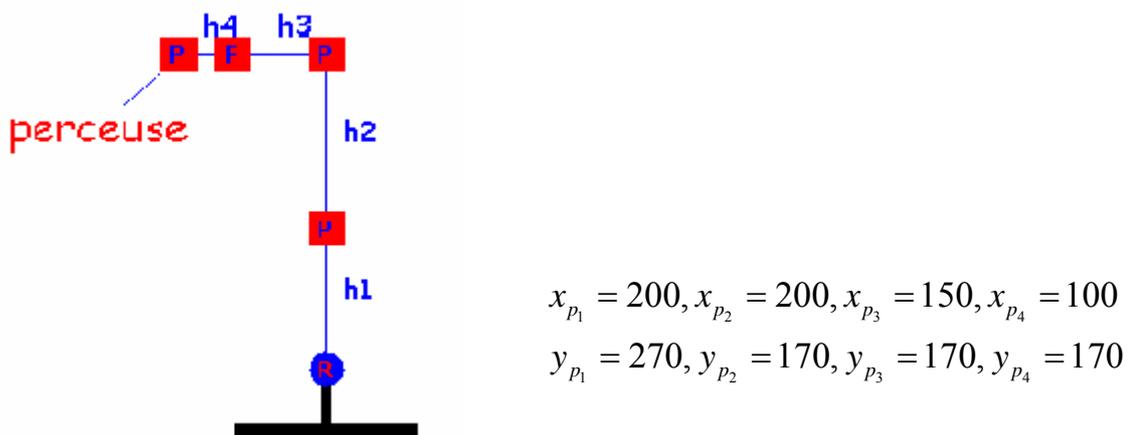


Figure 4.4 : Phase de perçage (Dimensions en centimètre).

Dans la cinquième position, le but (besoin) était la localisation des chaînons en deux dimensions (2D) sur l'écran, pour cela la cinquième position conçue était la phase d'alésage La figure 4.5 montre la position du robot sur l'aléreuse.

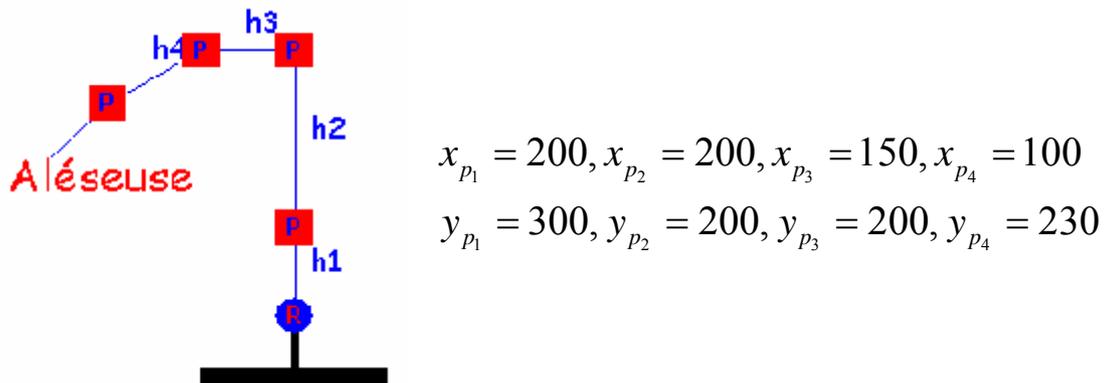


Figure 4.5 : Phase d'alésage (Dimensions en centimètre).

Ayant exposé les différents logiciels et bibliothèques utilisés pour le développement de l'interface, je passe maintenant, à la description de cette dernière. Dans mon interface, je propose un espace graphique, c'est-à-dire que j'offre à l'utilisateur un espace bidimensionnel, où il peut projeter cinq états du robot mais aussi, faire animer ces derniers et obtenir les différentes possibilités des modèles géométriques cinématique et dynamique directes et inverses de ce robot. Les figures (4.6),(4.7),(4.8),(4.9)et(4.10),permettent de connaître les valeurs des coordonnées généralisées de chaque chaînon pour une position donnée,ainsi que la position de l'effecteur dans l'espace.

On précise que l'utilisateur dispose dans l'interface une palette, cette dernière comporte les différents types de curseurs, alors l'utilisateur peut déplacer, faire tourner, pivoter le robot dans tous les sens.

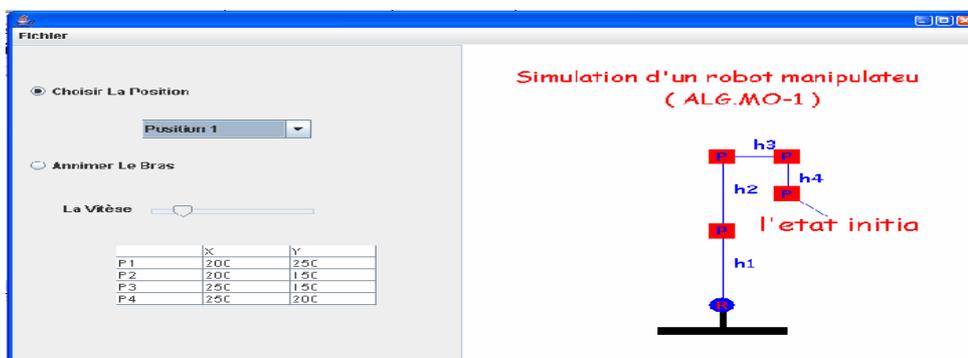


Figure 4.6 : Interface de l'utilisateur pour l'état initial

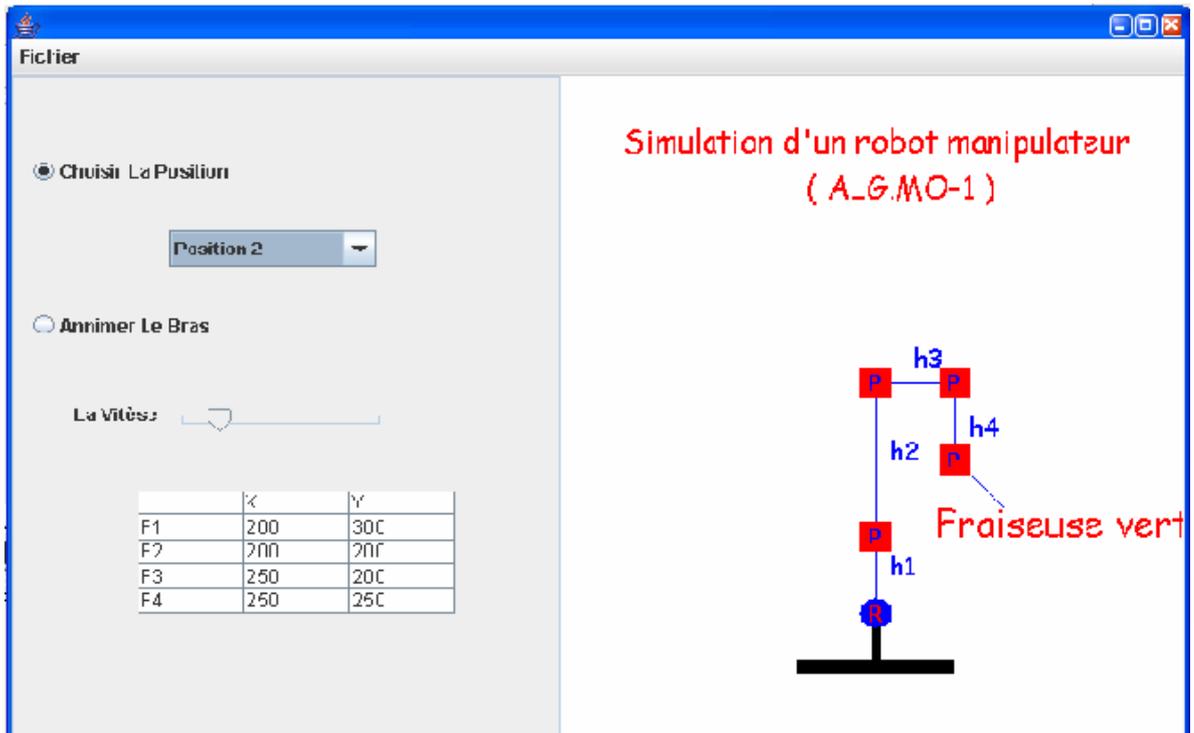


Figure 4.7 : Interface de l'utilisateur pour la fraiseuse verticale

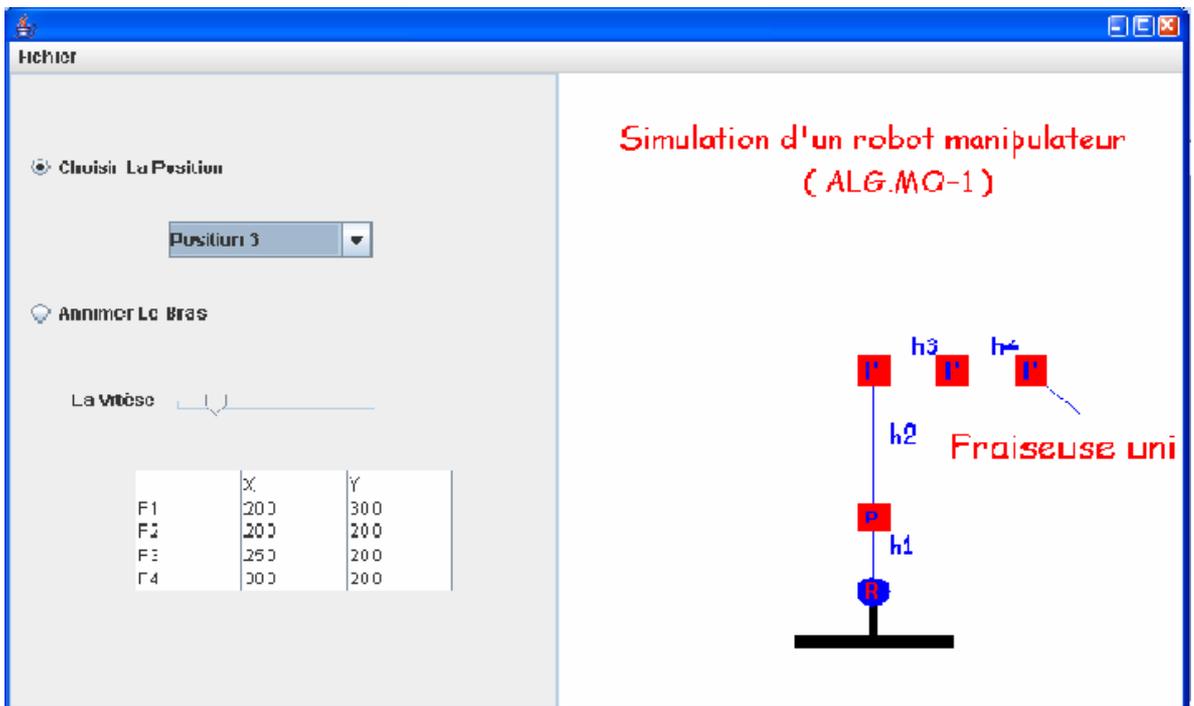


Figure 4.8 : Interface de l'utilisateur pour la fraiseuse universelle

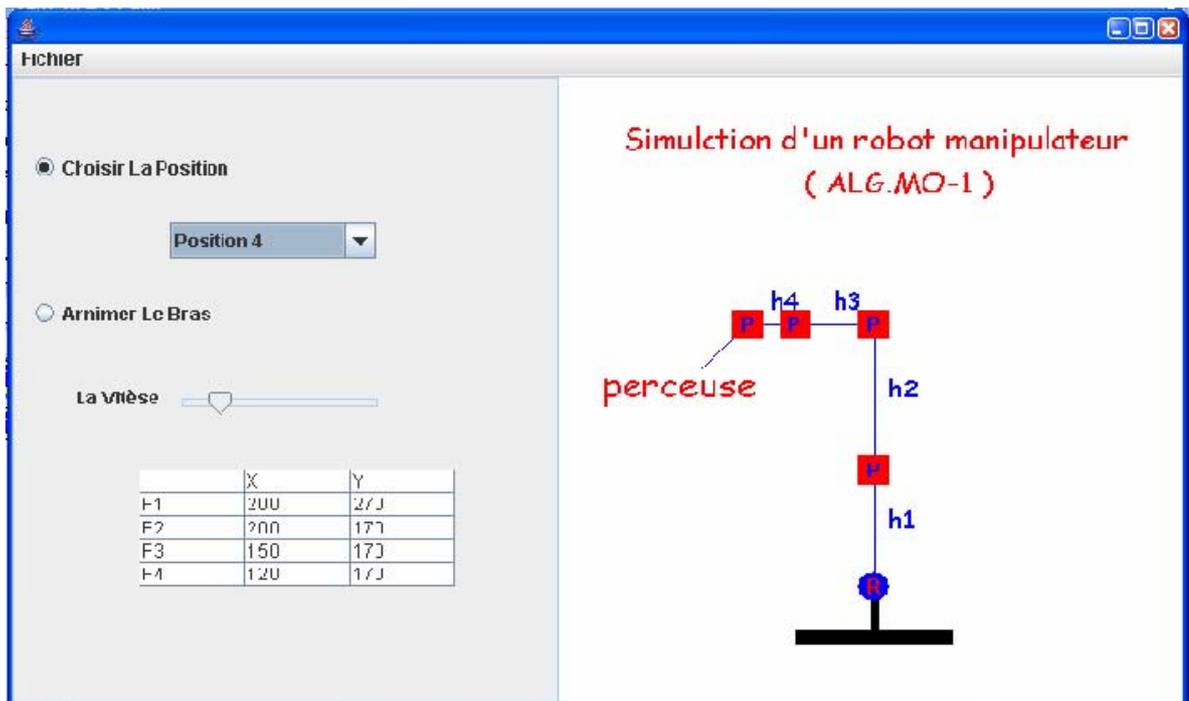


Figure 4.9 : Interface de l'utilisateur pour la perceuse

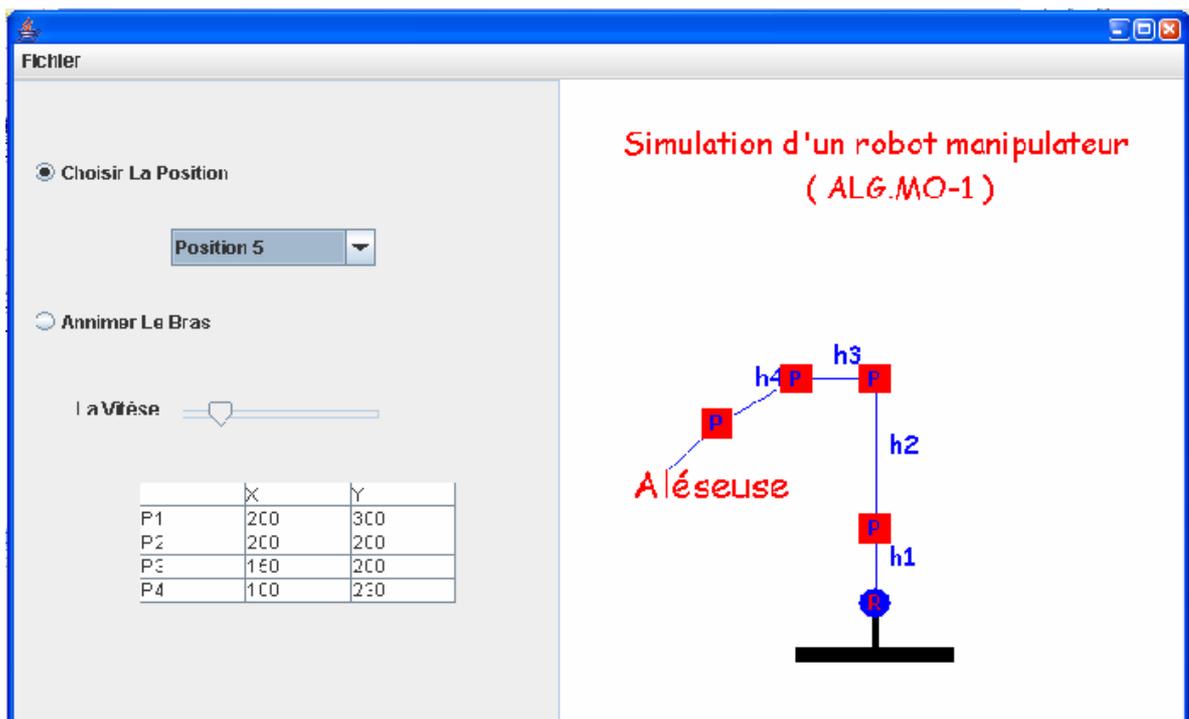


Figure 4.10 : Interface de l'utilisateur pour l'aléseuse

4.3 Fichier de résultats :

Afin de mieux apprécier les caractéristiques géométrique cinématique et dynamique de robot type, j'ai déterminé ces derniers pour un seul temps de mouvement T nécessaire. Les énergies cinétiques potentielles et de dissipation sont données, les équations de Lagrange sont établies. Pour effectuer un cycle de travail. J'obtiendrais sur les graphes ci-dessous des figures (4.11),(4.12),(4.13) et (4.14) l'évolution en fonction du temps des positions, des vitesses. Des accélérations ainsi que des couples et forces pour un temps T égal à 5, secondes.

Dans ce mémoire j'ai présenté une interface en exposant les différentes palettes qu'elle comporte, et en expliquant les différentes possibilités qu'elle offre à l'utilisateur.

Le logiciel Fortran consiste en un langage interprété qui s'exécute dans une fenêtre dite d'exécution. L'intérêt de Fortran tient, d'une part, à sa simplicité d'utilisation, et à sa richesse fonctionnelle arithmétique matricielle et nombreuses fonctions, analyse numérique, graphique.

En ce qui me concerne, j'ai utilisé la version Fortran 6.1, pour deux raisons la première sa puissance en calcul matriciel, la deuxième raison, est l'utilisation des boucles pour l'obtention des exécutables.

En considérant les systèmes d'équations différentielles de la géométrie cinématique et dynamique et après récupération des variables et appel aux fonctions et affectation de coordonnées et rotations on aboutit à l'affichage des résultats sous forme de blocs de matrices, on peut illustrer ces résultats numériques en utilisant l'outil Microsoft Excel.

Dans le cas où les causes structurelles seraient considérées les paramètres pris en considération seront l'inertie du rotor, les inerties de l'arbre entrée réducteur, rapport de réduction et coefficients du frottement visqueux, en ce qui concerne les termes de Coriolis sont représentés sous forme matricielle de dimension $(n \times (n-1) n/2)$.

Les graphes:

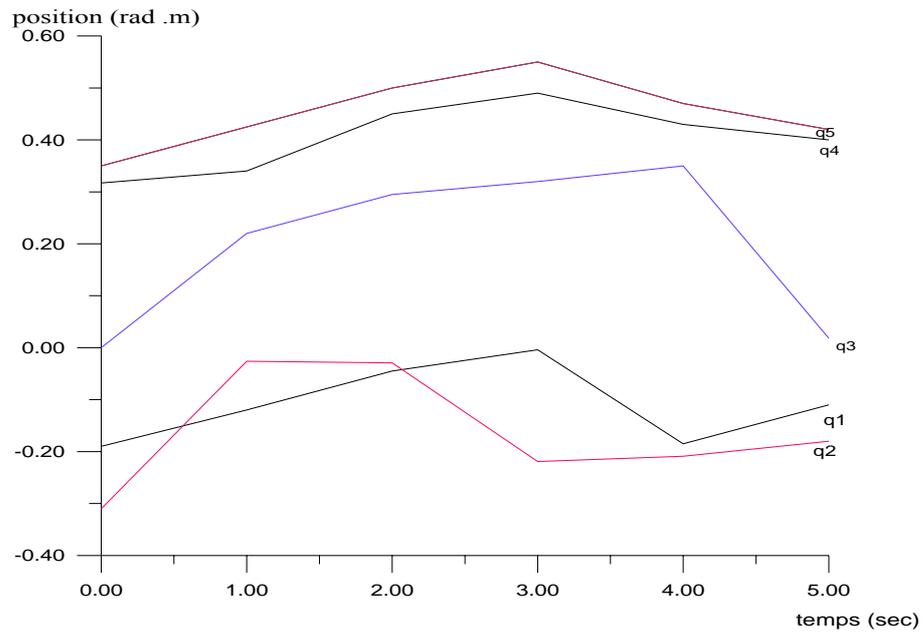


Figure 4.11: Positions

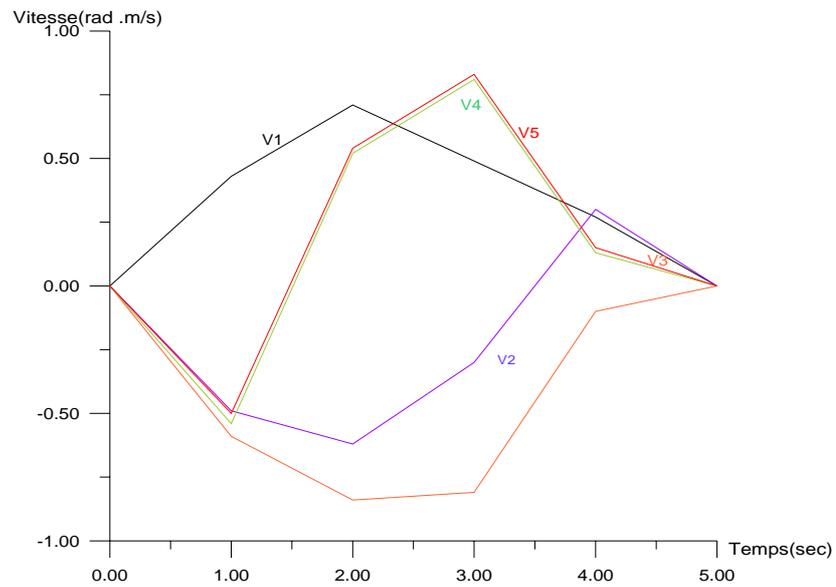


Figure 4.12: Vitesses

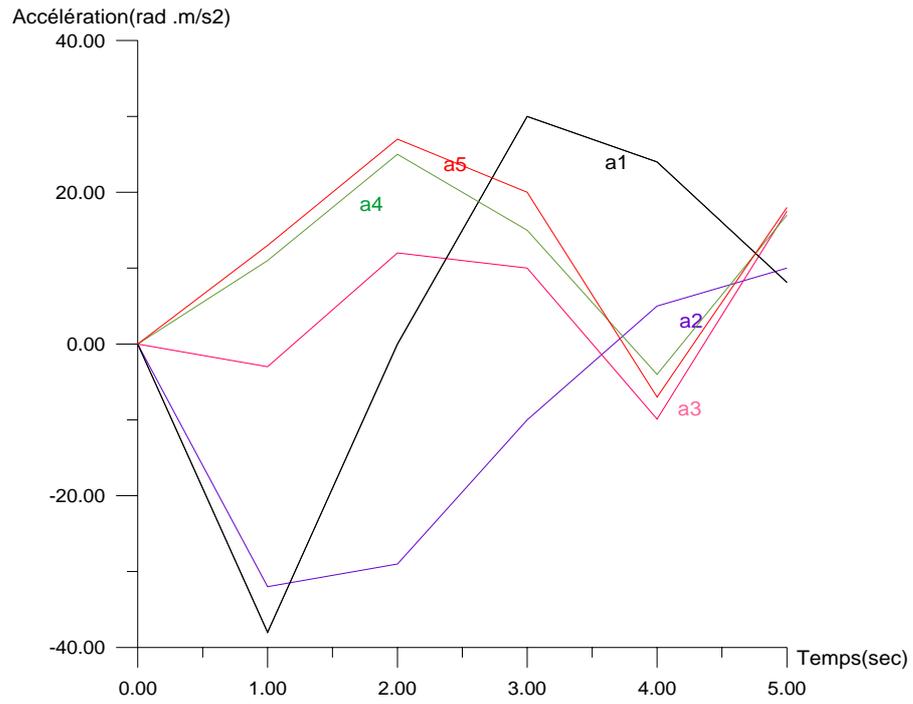


Figure 4.13: Accélérations

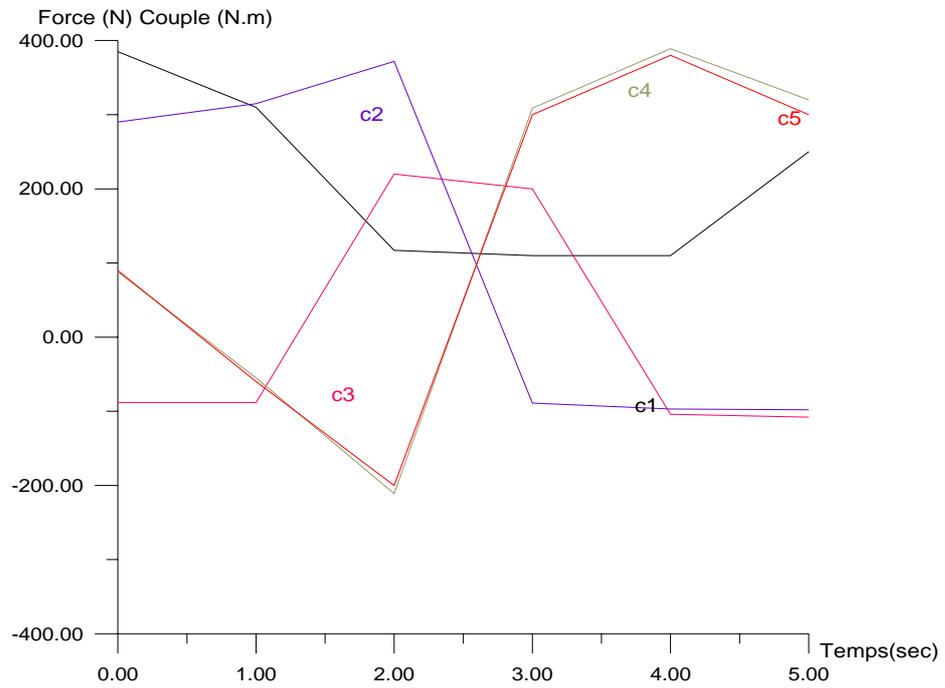


Figure 4.14: Forces, couples

4.4 Interprétation des résultats :

Un programme de simulation a été finalisé qui englobe les différentes étapes de l'étude, dont la partie géométrique illustrée par les différentes courbes de position de ce robot manipulateur (Figure 4.11). Un régime adéquat de fonctionnement du robot a été indiqué, dont la partie cinématique (Figure 4.12) et (Figure 4.13). Et à la fin les courbes dynamiques (Figure 4.14). Dans mon système, l'utilisateur peut choisir le type d'actionneur -moteur asynchrones- autre que les actionneur de type hydraulique, et ainsi faire rentrer les caractéristiques spécifiques, ceci va faire intervenir la dynamique des moteurs asynchrones dans la dynamique des différentes articulations. On peut voir une série de simulations appliquée au robot en passant par un interface de code source du microcontrôleur du servomoteur PIC, et une série de capteurs cartes de ENTRE SORTIE connectées par l'intermédiaire d'un automate et armoire de commandes, avec les actionneurs du robot situées dans le circuit hydraulique du robot.

Ce programme nous permet d'étudier n'importe quels robots industriels pour les différentes stations d'usinage, dont le but d'augmenter la productivité de n'importe quel atelier d'usinage flexible.

Si les vitesses augmentent, il est évident que les modèles calculés sont très loin de la réalité, parce que d'une part des forces inertielles, centrifuges et de couplage apparaissent et d'autre part les jeux dus aux frottements et élasticités de toutes origines ne sont plus négligés. Il est donc nécessaire de revoir la modélisation en tenant compte des ces phénomènes dynamiques. Le calcul effectué détermine les variables articulaires en fonction des forces (ou / et couples) généralisés. Pour le calcul j'ai utilisé la méthode de Runge-Kutta d'ordre 4 à pas variable pour la résolution des systèmes différentiels. Un modèle des déformations est nécessaire, quant à la validation et comparaison, mes résultats présente une co-ressemblance avec les résultats [21], [55] et [71], malgré les différences de performances entre nos robots.

Des paramètres comme la vitesse, la température ou le temps vont influencer et certaines formes d'usure sont relativement irrégulières, d'autres au contraire provoquent des sauts très brutaux, dont des rapports pouvant aller parfois de 1 à 100 000 ou plus, lorsque certaines valeurs critiques sont franchies.

CONCLUSION GENERALE

Tout pays rêve d'évoluer à chaque instant par l'amélioration de sa production, ces produits, son service pour arriver à la plate-forme d'un atelier, ce dernier est un rêve dans notre pays cette majeure automation (atelier flexible de fabrication), nécessite une large présence des robots industriels.

On abordait la modélisation en appliquant la convention de Denavit–Hartenberg, le calcul du modèle géométrique inverse nous a permis la localisation de l'effecteur à n'importe quel point de l'espace de travail, les inconvénients sont évités ou contournés de la manière suivante : Le problème de multiples solutions du modèle inverse n'intervient pas. En effet, le robot possède une configuration initiale connue, et se rend à une autre solution à partir de celle-ci, il ne sert à rien de calculer des configurations du robot qui seraient impossibles à atteindre depuis sa position ; Une haute précision de solutions obtenues n'est pas nécessaire puisqu'il suffit de fournir à l'utilisateur une vision globale, le calcul des accroissements est à chaque fois effectué à partir d'une nouvelle configuration exacte du robot. Quant au problème des singularités, il existe plusieurs méthodes mathématiques pour les traiter et les éviter.

La géométrie de la mécanique rend impossible la détermination d'une solution car les contraintes imposées ne sont pas atteintes. La modélisation cinématique directe ou inverse me permet de déterminer la vitesse de l'organe terminal. Par contre des butées empêchent le robot d'atteindre les points en dehors de volume du travail malgré l'existence de solutions mathématiques. Pour la mise en équation du modèle dynamique j'ai utilisé le formalisme de Lagrange, ce qui conduit au calcul de l'énergie de dissipation par frottement visqueux dans les joints. La modélisation dynamique m'a permis de mettre en évidence un énorme calcul. J'ai constaté que les erreurs sont dues aux défauts géométriques des organes mécaniques, (non linéarités dues aux frottements mécaniques par exemple). Et avec le développement de l'électronique, de l'informatique et des moyens de fabrication (machine à commande numérique, CFAO) la précision tend à devenir largement suffisante pour la majorité des applications industrielles, j'ai conçu un programme à partir de la géométrie, afin de mieux

apprécier les caractéristiques cinématique et dynamique de mon robot type, pour effectuer un cycle de travail, en exploitant les modèles géométriques cinématiques et dynamiques et en tenant compte des lois de mouvements utilisées en robotique dans le but d'évaluer les couples articulaires pour mon manipulateur qui n'est alors qu'une composante de la cellule robotisée. J'obtiens sur les graphes de l'évolution en fonction du temps des positions, des vitesses, des accélérations ainsi que des couples et forces. Le travail effectué me permet d'étudier n'importe quels robots industriels pour les différentes stations d'usinage dont le but est d'augmenter la productivité de n'importe quel atelier d'usinage flexible.

J'ai constaté que dans tous les cas il est nécessaire de bien comprendre les mécanismes du frottement de manière à adapter au mieux les résultats. On peut augmenter l'espace de travail, ainsi que l'autorisation d'une charge utile plus importante, on peut donc parler d'une famille de robots (ALG.-M.O. -1). (ALG.-M.O. -2).

Pour une valorisation de mon travail, j'ai élaboré comme première application, une simulation, sous diverses configurations, permettant de visualiser le fonctionnement du robot dans son environnement de travail.

Un brevet d'invention a été déposé à l'I.N.A.P.I. et également une conception d'un modèle-type a été proposée, une banque des données sur les paramètres de la mécanique articulée et des articulateurs types sera proposée. Et en fin mon travail de recherche aura pour perspective une communication sur le modèle mathématique des systèmes mécanique poly articulés. Ce travail, est sujet à des améliorations. En ce qui concerne ses améliorations ultérieures pour un même travail, il est recommandé aux personnes devant pour suivre ce travail de : Faire une étude bien détaillée sur la modélisation dynamique en tenant compte les caractéristiques dynamiques et les erreurs statiques et dynamiques ainsi que la précision.

Je souhaite que les promotions à venir compléteront mon travail pour réaliser le manipulateur après calcul et acquisition des organes constitutifs.

Il est à noter que le rapprochement entre l'université et l'industrie est seul et unique moyen pour le développement industriel et avec le soutien de DIEU on peut résoudre les problèmes existants dans cet univers ou dans la vie.

APPENDICES

Appendice A :

Titre de l'invention :

Robot manipulateur des machines - outils (Algérie- M.O.-1)

Domaine technique auquel se rapporte l'invention :

La présente invention concerne les robots industriels utilisés dans les ateliers flexibles de fabrication mécanique, qui forment une nouvelle série de fabrication (Brevets d'invention: WO 9507799623622 AL 1996088, 9622856 AL 19960801, 9620818 AL 19960711. . .).

But de l'invention :

La nouvelle structure très simple du robot industriel type (ALG. -M.O.-1) est une nouveauté en Algérie et a été adaptée aux données d'une station d'usinage composée de quatre phases d'usinage différentes, et en vue de transformer cette unité de fabrication mécanique en une cellule flexible.

Etat de la technique antérieure :

L'opération après amélioration des caractéristiques techniques y compris le centre de fraisage alésage, pour la fabrication d'une grande variété des pièces constructives des véhicules industriels se caractérise par une trajectoire fermée d'une suite de déplacement et d'arrêt avec une livraison évacuation des pièces à l'aide de la main et cela après un choix optimal de toutes les caractéristiques.

Les différentes opérations qui comportent la descente, la fermeture et la montée de la main ainsi que le déchargement à l'aide des circuits et des modules judicieux réalisés.

Les phases intermédiaires sont celles du déclenchement de l'usinage dans les différentes phases et de l'évacuation vers les transporteurs à palettes, après fabrication.

Les modules utilisés pour la conception, le fonctionnement et suivant le cahier des charges de ce robot industriel sont composés de cylindres, de vérins hydrauliques, des éléments auxiliaires qui constituent les composantes du circuit hydraulique, des

éléments d'assemblage, la main de chargement et déchargement et de moteurs électriques.

Les blocs ci-dessus sont liés entre eux et équipés par des capteurs qui permettent de produire les positionnements et l'ordre d'orientation des objets à traiter à l'entrée des installations automatiques lors de l'exécution des opérations et le déchargement de la dernière phase, ainsi que la synchronisation pendant les rotations d'angles (90° , 17° , 56° et 90°) pour effectuer les différentes phases. La chaîne de production existante sur la figure 1 pour quatre postes de travail, deux palettes (a_0 et b_0) et deux chariots de transport de pièce (a et b), le chariot (a) livre les pièces, tandis que le chariot (b) les évacue.

Le robot est situé au centre des quatre postes, le centre est bien la position repère.

Le cycle des mouvements est :

La rotation initiale du robot industriel vers le chariot (a) (angle = 135°), la descente de la main dépend des caractéristiques des machines-outils utilisées synchronisées par les capteurs.

La fermeture de la main sur la pièce répond aux poids des pièces usinées et qui ne doivent pas dépasser deux tonnes. Le régime de fonctionnement proposé, ainsi que la montée et la rotation vers (a_0) (angle = 90°) et le déchargement dans la deuxième phase de fabrication d'un repère (angle = 45°).

Il en est de même pour la troisième phase (b_0) après avoir exécuté la quatrième phase (b).

Un déchargement correct de la pièce est effectué à l'aide d'un mouvement de rotation d'un angle ($\gamma = 270^\circ$), le robot est positionné devant le chariot (a) pour exécuter une nouvelle gamme d'usinage.

Enoncé des figures :

De toutes façons, l'invention sera bien comprise à l'aide de la description qui suit, en références au dessins annexés, représentant, à titre d'exemple non limitatifs, plusieurs formes d'exécution de ce robot manipulateur :

-figure A-1 est une vue de la station de fraisage –alésage ;

-figure A-2 représente le schéma cinématique du robot industriel, avec la convention du cycle des mouvements.

-figure A-3 est une vue en élévation et partiellement en coupe d'un robot manipulateur, avec ses modules et la partie du circuit hydraulique.

-figure A-4 est une vue en coupe du module de levage à un cylindre à double effet.

-figure A-5 a présenté un module de rotation à deux cylindres.

-figure A-6 est une vue du module de déplacement longitudinal vers le bas de l'organe terminal.

-figure A-7 a présenté le module de la main sous forme de mâchoire pour garantir une bonne adhérence.

Présentation de l'essence de l'invention et son mode de réalisation :

Le robot industriel (fig. 3) est solidaire d'une base (1), maintenue à sa partie inférieure par un corps (2), qui sert aussi de support à la colonne du module de déplacement vertical (fig. 4) et à son intérieur sont encoché les éléments du circuit hydraulique. Au dessus du cylindre (côté frontale), on a monté le module de rotation (fig. 5) et au dessus (cylindre) les modules de déplacement transversal (fig. 6) de la fermeture, d'ouverture et de la main.

Le bâti du robot est composé d'une base (1) et le corps (2), ce dernier est lié à la base par quatre supports (3) (fig. 3). Les boulons (4) servent pour la fixation au sol.

Dans la direction verticale, on a placé une colonne (5) fixée au-dessus du bras par soudage à la plate-forme (6), celle-ci est fixée par des boulons (7) avec le module de déplacement transversal. Et ce bâti est limité par des couvercles - avant, latéraux et arrières.

Le module de rotation (fig. 5) est constitué de deux cylindres (1) et (2) installé dans le corps (3) à l'aide des éléments de fixation (4) et (5).

La transmission par crémaillère et la roue dentée (6) et (7) transforme le mouvement rectiligne des vérins plongeurs (à double effet) en un mouvement de rotation de la colonne.

C'est un système réversible ayant les avantages d'une réalisation simple pouvant supporter des grandes charges.

La pièce (8) sert en même temps pour le guidage de la crémaillère et à la fixation du vérin avec le corps.

La partie saillante du couvercle (9) permet l'amortissement du mouvement du piston (10) en formant un lit de l'huile une fois engagée dans le creux aménagé dans le piston.

Le module de levage fonctionne à l'aide d'un cylindre à double effet (fig. 4).

Dans le cylindre (6) ou se trouve le piston (7) qui est garni par deux joints, la tige (8) du piston est connectée avec la plate forme de dessus (9), la partie inférieure du carré, qui fixe la superficie extérieure du cylindre par un jeu, où sont encochés des billes (10).

Le déplacement rectiligne de l'ensemble (7, 8 et 9) s'effectue grâce à un système de guidage protégé par un cache en caoutchouc (14) le long du cylindre (1), écrou (2) et une tige (3).

Le déplacement vers le haut s'effectue par une transmission de l'huile dans la conduite (13) et en bas à l'aide de la fente (12) au dessus sur le méplat (4) sont placés des capteurs de positions, qui confirment le passage des chaînons mobiles dans les points des différentes positions à l'aide des aimants (5).

La transmission de l'huile dans la conduite inférieure (13) pousse le piston (7) vers le haut et par la suite fait monter le bras horizontal.

Pour un bon guidage lors de la montée (ou de la descente) on utilise des billes.

Le mécanisme sert comme plate-forme pour le module de déplacement transversal et repose (en bas) sur le module de rotation.

Le module de déplacement longitudinal vers le bas de l'organe terminal (fig. 6) sert à rapprocher la main de la zone du travail, c'est un vérin à double effet, constitué d'un cylindre (1) à l'intérieur duquel se déplace le piston creux (2) muni d'une garniture d'étanchéité aux deux extrémités (3).

Les tiges creuses (4) et (5) assurent le guidage parfait du piston. Aux extrémités du cylindre (1) sont installées des douilles, à gauche douille (6) avec les joints d'étanchéité qui sont fixés par des vis, à droite la douille (7) avec un filetage contre-écrou (9), et la douille (8) qui est liée par un filetage à la tige (4). A droite, le piston est connecté à la bride (10) avec la douille (11) auquel est fixée l'installation de l'organe terminal.

Le cylindre est muni des conduites (12) et (13) pour la circulation de l'huile assurant le déplacement en va et vient des pistons.

Le module de la main (fig. 7) est composé par une structure dont le bout est constitué de leviers articulés (1, 2, 3 et 4) et de doigts, qui sont au nombre de deux sous forme de mâchoire, inclinés, changeables pour garantir une bonne adhérence. Le robot industriel peut adapter plusieurs variétés de mains en cas de nécessité.

La main possède un système vis écrou (5) pour le réglage de la hauteur en cas de besoin.

Le bout supérieur du cylindre (10) hydraulique d'attrape est lié par goujon au bout inférieur du piston (12) qui sert à déplacer la main vers le bas pour une opération de chargement ou vers le haut pour véhiculer la pièce.

Le piston active dans un cylindre hydraulique (13) lié au module de déplacement transversal par deux équerres fixées par quatre boulons pour chacun, ce dispositif permet la fixation des diverses mains sur le robot.

Les deux cylindres sont munis des conduites (6) pour celui de dessus afin d'assurer la descente et la montée du piston (7). Et celui du bas pour attraper la pièce (8) et la décharger.

Le développement de l'effort nécessaire est pris en considération en cas de contact.

La main peut avoir un mouvement de rotation actionné par le circuit hydraulique ainsi que le positionnement de l'organe terminal sur la pièce à soulever est réalisé par un contact de fin de course.

Revendications :

1. Le principe de montage, de déchargement sur le dispositif d'usinage des différentes stations de fabrication mécanique s'effectue par un procédé d'une cellule flexible, en améliorant quelques caractéristiques techniques, en augmentant la cadence, la précision et la simplicité des phases d'usinage selon les données de l'unité de production.

2. Robot selon la revendication 1, ces structures des installations proposées pour Le robot industriel manipulateur est construit suivant le principe d'approvisionnement des éléments de base par blocs pour les différents schémas technologiques proposés et déchargement.

3. Robot selon la revendication 1, la construction et le principe d'action des modules et des installations complémentaires doivent permettre d'accomplir la manipulation et l'évacuation de produits usinés, les opérations de base sont réalisées grâce aux passages inventés et leur simplicité d'utilisation.

4. Robot selon la revendication 1, la production en série, ou en grande série augmente par ce procédé de robotisation. Le remplacement du processus manuel par un autre plus précis et robuste permet l'optimisation des indices technico-économiques.

5. Robot selon la revendication 1, le choix des nuances répondant aux constructions des éléments, l'utilisation des pièces universelles dans la construction du manipulateur, la simplicité des commandes directes ou indirectes, l'amélioration de la fiabilité de l'ensemble permettent d'avoir un produit de qualité et moins coûteux.

Abrégé descriptif :

Titre de l'invention :

Robot manipulateur des machines – outils (Algérie -M.O.-1).

Le robot industriel (ALG. -M.O.-1) proposé pour les différentes stations d'usinage est adapté aux différentes données de fabrication mécanique. Il est composé de modules utilisés pour la conception et cela suivant le cahier des charges proposé à l'étude.

Le robot est solidaire à une base maintenue par un corps, qui sert aussi de support à la colonne du module de déplacement vertical.

A l'intérieur de ce corps se trouvent les éléments du circuit hydraulique, juste à la partie inférieure du cylindre est fixé le module de rotation, au dessus à l'aide de supports est maintenu le module de déplacement longitudinal, à l'extrémité de ce dernier est monté le module de la main avec ses deux cylindres hydrauliques qui servent pour la montée et la descente, ainsi que le serrage et le déchargement de la pièce dans les différentes phases d'usinage.

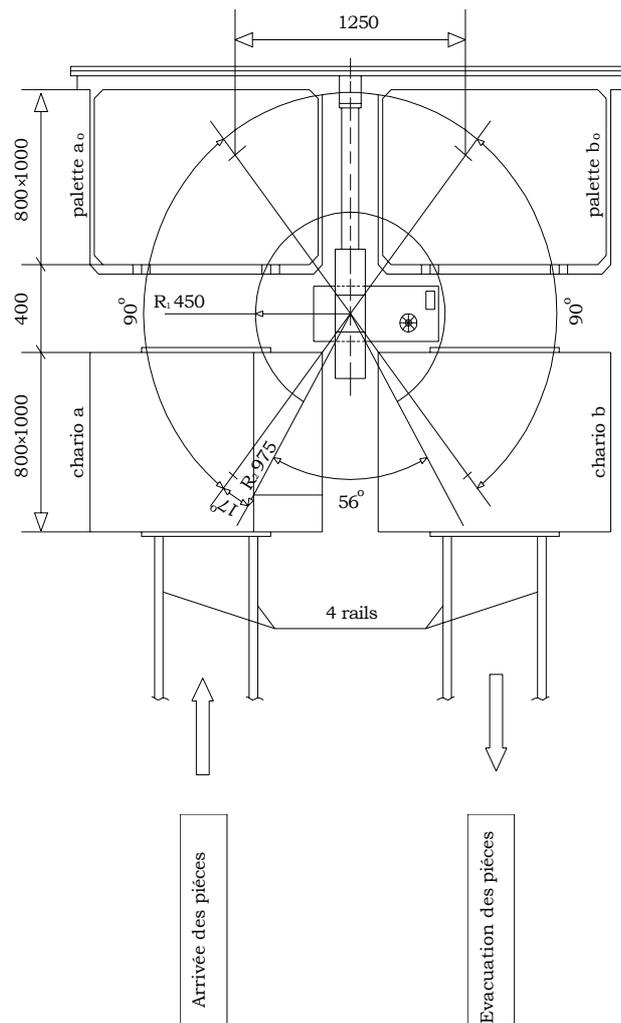


Figure A-1 : Vue de la station de fraisage –alésage ;

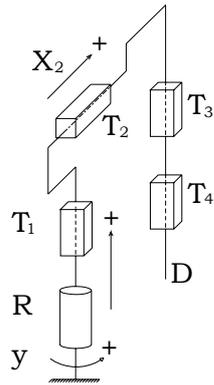


Figure A-2 : Schéma cinématique du robot industriel, avec la convention du cycle des mouvements.

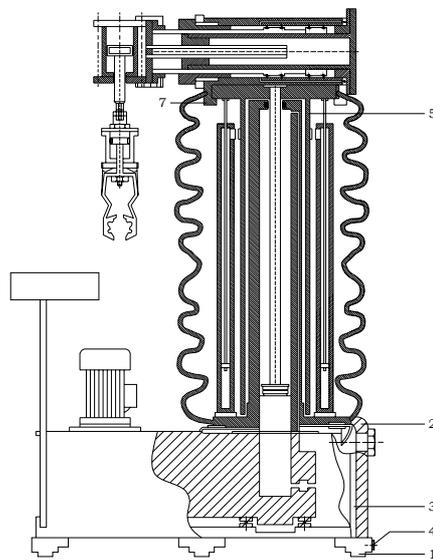


Figure A-3 : Vue en élévation et partiellement en coupe d'un robot manipulateur, avec ses modules et la partie du circuit hydraulique.

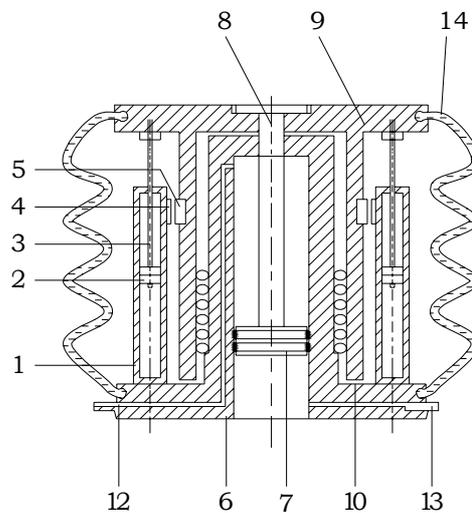


Figure A-4 : Vue en coupe du module de levage à un cylindre à double effet.

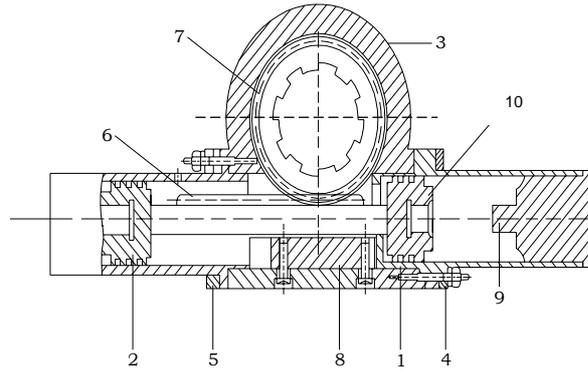


Figure A-5 : Module de rotation à deux cylindres.

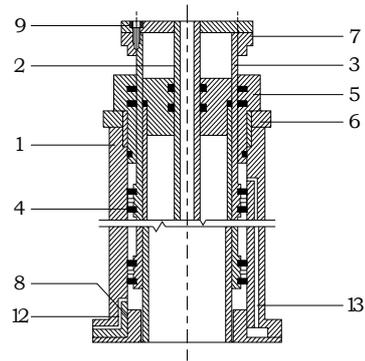


Figure A-6 : Module de déplacement longitudinal vers le bas de l'organe terminal.

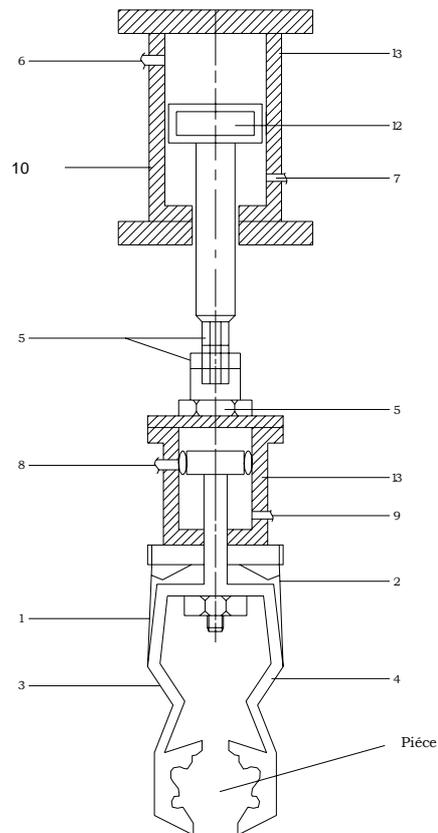


Figure A-7 : Module de la main sous forme de mâchoire pour garantir une bonne adhérence.

المعهد الوطني الجزائري للملكية الصناعية

INSTITUT NATIONAL ALGERIEN
DE LA PROPRIETE INDUSTRIELLE



الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE
DEMOCRATIQUE ET POPULAIRE

DEMANDE DE PROTECTION D'INVENTION

Imp. Officielle - Alger

(12) NATURE DE LA DEMANDE *	CADRE RESERVE A L'INAPI	
<input checked="" type="checkbox"/> BREVET D'INVENTION (A1)	(21) N° de dépôt :	060205
<input type="checkbox"/> CERTIFICAT D'ADDITION (A2)	(22) Date de dépôt :	24 AVR 2006
Signature du préposé à la réception	Heure	14H00
	Visa	

(74) - MANDATAIRE : <i>Nom, Prénom, Adresse</i>	Date du pouvoir
	Nationalité du ou des déposants ALGERIENNE

(71) - DEPOSANT (S) : *Nom, Prénom, (dénomination), Adresse*

ALLALI ABDERRAZAK cedex B.P. 47 Beni-Mered Blida
BRAHIMI ABDELHALIM 01 Rue Sadok Kouider El-Affroun Blida
BENMISRA ABDELKADER Cité du 24 février B.T E₄ N°11 Blida
HALAIMIA MUSTAPHA KAMEL Cité du 1^{er} Mai Bt n° 17 Cage B appt N° 07 Ouled Yaich Blida

Tél. : Fax : E-mail : Yasminallali@Yahoo.fr

(54) - TITRE DE L'INVENTION :

**ROBOT MANIPULATEUR DES MACHINES OUTILS
(ALGERIE.-M.O.-1)**

(30) - REVENDICATION DE PRIORITE (S)

(31) - N° (s) de dépôt	(32) - date (s) :	(33) - pays d'origine	Nature de la demande

Demande de certificat d'addition rattaché au brevet principal N° du

Fait à **BLIDA** le **24 AVR 2006** Signature et cachet
ALLALI .A
BRAHIMI .A
BENMISRA .A
HALAIMIA .MK

Appendice B :

Program CALCUL DES ROBOTS INDUSTRIELES MODELISATION GEOMETRIQUE CINEMATIQUE ET DYNAMIQUE ELABORE PAR A.ALLALI, M.M.HATTALI, A.BENMISRA, M.MEGHERBI, N.BEDHIAF, R.MAZARI, K.AIMEUR, N.MOHAMMEDI, K.TOUMADJ, A.BRAHIMI, M.K.HALAIMIA.

En collaboration : Départements d'Informatique, Mathématique, Electronique, Aéronautique et Génie mécanique de l'Université de Saad Dahleb de Blida.

- Bibliothèque de l'Ecole Nationale Polytechnique d'El-Harrache, Alger.
- Département de Génie Mécanique, Laboratoire de Vibroacoustique et Bibliothèque centrale de l'Université de M'hamed Bouguarra de Boumerdès.
- Centre de Développement des Technologies Avancé, C.D.T.A., de Baba Hssene, Alger.
- Le Centre de Recherche sur l'Informatique Scientifique et Technique, C.R.I.S.T., Ben Aknoun, Alger.

Université de Saad Dahleb de Blida

19 Mai 2006

22 :54

Nbre. d'articulations, coord. Point initial, coord. Point final, temps T, ΔT , paramètres de D.H. ; Positions, Vitesses, Accélérations, Forces et Couples, Variable angulaire.

Programme d'un espace de travail.

```
c PROGRAM: ESPACE DE TRAVAIL.
c PURPOSE: Entry point for the console application.
c program ESPACEDETRAVAIL
c Déclaration des variables
c Variables
integer n,i,j,k
real a,b,c,e,f,d,r,u,q,p,z2,z3,z4,z5,h1,h2,h3,h4
real ydmax, ydmin,ymin, ymax,ydmaxi,ydmini, ymini,ymaxi, pi
real y1,y2,y6, y3,y4,px,py,pz ,hmin,hmax
real d3,d4
```

dimension T(3),ymin(1),ymax(1) ,hmin(1) ,hmax(1)

C

c introduction de la dimension des matrices de passages

pi=3.1415926535897932384626433832795

c déclaration de PI pour la transformation des degrés en radians

n=5

d3=0.6

d4=0.6

z2=600

z3=635

z4=700

z5=500

c Entrer des paramètres de Dénavit-Hertenberg

c write(*,*)'chargement des limites artiuulaires '

! do k=1,n

c incrémentation chaque matrice de passage

write(*,*)'donner les limites angulaires des butées mécaniques

+ de 1 articulation k'

write(*,*)'theta min'

read(*,*) theta min

ymini=(theta min*pi)/180

write(*,*)'ymini'

write(*,*)'theta max'

read(*,*)theta max

ymaxi=(theta max*pi)/180

write(*,*)'ymaxi'

write(*,*)'theta min','=', ymini, 'theta max','=',ymaxi

c ymin(k)=ymini

ymax(k)=ymaxi

write(*,*)'(ymax(i)* 180/pi,i=1,5)'

write(*,*)'(ymin(i)* 180/pi,i1,5)'

do k=1,n

write(*,*)'donner les limites de la distance des butées mécaniques

```

+   de K articulation k'
    write(*,*)'zmin'
read(*,*) zmin
write(*,*)'zmax'
    read(*,*) zmax
write(*,*)'zmin',k, '=', zmin , 'zmax',k, '=',zmax
    write(*,*)'hmin'
read(*,*) hmin
write(*,*)'hmax'
    read(*,*) hmax
write(*,*)'hmin',k, '=', hmin , 'hmax',k, '=',hmax
c   Entrer des paramètre de Dénavit-Hertenberg
c   write(*,*)'chargement des limites artiulaires '
c   incrimmentation pour chaque matrice de passage
write(*,*)'donner les limites angulaires des butés mécanique
+   de 1 articuiation k'
write(*,*)'theta min'
    read(*,*) ydmin
    ymini=(ydmin*pi)/1 80
    write(*,*)'theta max'
    read(*,*) ydmax
    ymaxi=(ydmax*pi)/1 80
write(*,*)'theta min',k, '=', ydmin, 'theta max',k, '=',ydmax
c   ymin(k)=ymini
    ymax(k)=ymaxi
    end do
write(*,*)'(ymax(i)* 180/pi,i=1,5)'
write(*,*)'(ymin(i)* 180/pi,i=1,5)'
c   calcul des matrice de transformation
a=ymin(1)
b=ymax(1)
c=ymin(2)
e=ymax(2)
f=ymin(3)

```

```

    p=ymax(3)
    write(*,*)'calcul du vecteur P(Px,Py,Pz)Position des
+   coordonnées articulaires'
    do 11 y1=a,b,0.05
    do 11 y2=c,e,1
    do 11 y3=f,p,0.1
c   calcul des élément de T(m,m) et son remplissage
write(*,*)' calcul des élément de T(m,m) et son remplissage '
    T(1)=(z3-z4)*cos(y1)
    T(2)=(z3-z4)*sin(y1)
    T(3)= (h1)+(h2)+(h3)+(h4)+z2-z5
    px=T(1)
    py=T(2)
    pz=T(3)
    !write(*, *) (T(i),i=1,3)
    !write(*,*) (T(i),i=1,3),';'
    write(*,*) T(1),T(2),T(3)
10   format (3f)
11   write(*,*)
c   Body of Espace de travail
    end .

```

Appendice C :

Programme principale

```
c PROGRAM
```

```
implicit none
```

```
real,dimension(100):: qi,qf,kv,ka,q,a,c,v,st,tf
```

```
real alpha1,alpha2,alpha3,alpha4,alpha5,teta1,teta2,teta3,  
+teta4,teta15
```

```
integer i,j,ii,jj,t,dt,tfm,w
```

```
WRITE(*,*)
```

```
write(*,*)
```

```
write(*,*)'nombre articulation w'
```

```
read(*,*)w
```

```
write(*,*)'les parametres de Denavit Hartenberg'
```

```
write(*,*)'alpha1,alpha2,alpha3,alpha4,alpha5'
```

```
read(*,*)alpha1,alpha2,alpha3,alpha4,alpha5
```

```
write(*,*)'teta1,teta2,teta3,teta4,teta15'
```

```
read(*,*)teta1,teta2,teta3,teta4,teta15
```

```
write(*,*)'LES PARAMETRES DU CHAINON'
```

```
do 122 j=1,5
```

```
write(*,*)'coordone de point initial=qi(j)'
```

```
read(*,*) qi(j)
```

```
write(*,*)'coordone de point final=qf'
```

```
read(*,*) qf(j)
```

```
write(*,*)'kv(j)'
```

```
read(*,*)kv(j)
```

```
write(*,*)'ka(j)'
```

```
read(*,*)ka(j)
```

```
read(*,*) qi(j),qf(j),kv(j),ka(j)
```

```
122 continue
```

```
st (1)=a cos (-1.) /180
```

```
st (2) =1
```

```
st (3)=1
```

```
st (4) =1
```

```
st (5) =1
```

```

dt=30
tfm=5
t=0
40 continue
call synch(qi,qf,kv,ka,tfm,st,tf)
print*, 'tfm=',tfm
do 41 i=1,5
call para (qi(i),qf(i),kv(i),ka(i) ,q(i),v(i) ,a(i) ,st(i),tfm,t)
41 continue
print*, '(q(i) ,i=1,5) ,t,q'
print*, '(v(i) ,i=1,5), v'
write(*,*)t      q(2)      q(3)      q(4)      q(5)'
write(*,*)t,q(2),q(3),q(4),q(5)
write(*,*)t      v(2)      v(3)      v(4)      v(5)'
write(*,*)t,v(2),v(3),v(4),v(5)
write(*,*)t      a(2)      a(3)      a(4)      a(5)'
write(*,*)t,a(1),a(2),a(3) ,a(4),a(5)
call dynami (q,v, a, c)
write(*,*)t      c(2)      c(3)      c(4)      c(5)'
write(*,*)t,c(1),c(8),c(15),c(24)
write(*,*)'le vecteur <c(ii)> '
write(*,*)'calcul des forces et les momoment articulaires'
write(*,*)t,(c(ii),ii=1,5)
write(*,*)t,(c(jj),jj=6,11)
write(*,*)t,(c(jj),jj=12,17)
write(*,*)t,(c(jj),jj=18,24)
t=t+1
if(t.le.6)goto 40
end .

```

Appendice D :

Programme Synchronisation

```
C*****
C   Synchronisation
C*****

subroutine synch(qi,qf,kv,ka,tfm,st,tf)
real qi(5),qf(5),kv(5),ka(5),tf(5),tfm,st(5),lamda(5),nu(5),s(5),
+s1, s2,p(5)
integer i,J
do 4 j=1,5
s1=abs(qf(j) - qi (j) )
s2=kv(j) **2/ka (j)
if (s1.gt.s2) go to 4
print*,'La condition du palier vitesse pour 1 articulation'
4   continue
tfm =0
print*,'qi=' , (qi(i),i=1,5)
print*,'qf=',(qf(i),i=1,5)
print*,'kv= ',(kv(i),i=1,5)
print*,'ka=',(ka(i),i=1,5)
print*,'st=',(st(i),i=1,5)
pause
do 15 j=1,5
tf(j)=kv(j) /ka(j) +abs ( (qf(j)-qi(j) ) *st(j) ) /kv(j)
print*,'tf(j)',tf(j)
pause
if (tf(j).ge.tfm) go to 15
tfm=tf(j)
15  continue
print*,'tfm-synhronisation ,tfm '
pause
do 2 j=1,5
s1=tfm-kv(j)/ka(j)
lamda(j)=abs(qf(j) -qi (j)) *st(j) / (kv(j) *s1)
```

```

kv(j) =lamda(j) *kv(j)
ka(j)=lamda(j)*ka(j)
print*,kv(j),ka(j),kv,ka, tps mini
pause 'kv ,ka'
2   continUe
return end.

```

Appendice E :

Programme E.1 : Programme de calcul des coordonnées; vitesses et accélérations articulaires

```

C *****
c   calcul des coordonnees; vitesses et accélérations
C                               articulaires
c*****
subroutine para (qi,qf,kv,ka,q,v,a,st,tfm,t)
real qi, qf, kv, ka, q, v,a, t, t0, si, tfm
c***** Loi de Bang-Bang avec palier de vitesse *****
t0=kv/ ka
if (t.le.t0) then
q=qi+ ( (ka*t*t/2)/st) *abs (qf-qi) / (qf-qi)
v=ka*t*abs (qf-qi) / (qf-qi)
a=ka*abs (qf-qi) / (qf-qi)
return
else
if (t.le.tfrn-t0) then
q=qi+( (t-t0/2)*Kv/st) *abs(qf-qi) / (qf-qi)
v=kv*abs (qf-qi) / (qf-qi)
a=0
return
else
if (t.gt.t0) then
q=qf- ((tfm-t) **2/2*ka) /st*abs (qf-qi) / (qf-qi)
v=(tfm-t)*ka*abs (qf-qi) / (qf-qi)
a=-ka
endif

```

```
endif
endif
return
end
```

Programme E.2 : Programme de calcul des forces et les moment articulaires

```
C *****
c   calcul des forces et les moment articulaires
c *****

subroutine dynami (q,v, a,c)
real v(4),a(4),q(4),c(24),m(4),x(3),z1,z2,z3,z4,z5,z6,z7,z8,z9,z10,
+z11,j(4)
j(1)=.0319
j(2)=0.0508
j(3)=129.14
j (4)=6.283
m(1)=25.549
m(2)=40.70
m(3)=241.779
m(4)=39.049
X(2)=0.02
x(3)=0.02
l2=0.6
z1=a(4)+v(1)**2*q(4)
z2=a(1)*q(4)+2*v(1)*v(4)
z3=m(4)*9.81+m(4)* (a(2)+a (3))
z4=a( 1)*q(3)+2*v(1)*v(3)
z5=m(4)+m(3)
z6=x(2)+x(3)+12
z7=a(3)-v(1)**2*q(3)
z8=m(2)+m(3)+m(4)
z9=l2/2
z10=q (2) +12/2
z11=m( 1)+m(2)+m( 3)+m(4)
```

```

C(1)=m(4)*z1
C(2)=m(4)*z2
C(3)=z3
C(4)=q(3)*z3
C(5)=-q(4)*z3
C(6)=q(4)*m(4)*z2-q(3)*m(4)*z1+J(1)*a(1)
C(7)=-m(3)*z4+z4*z1
C(8)=m(4)*z2+m(3)*z7
C(9)=z5*9.81+z5*a(2)+m(4)*a(3)
C(10)=z5*9.81+z5*a(2)+m(4)*a(3)-z6*(m(4)*z2+m(3)*z7)
C(11)=q(4)*z2+z6+(-m(3)*z4+m(4)*z1)
C(12)=m(4)*q(4)*z2-3*q(3)*m(4)*z1+m(3)*q(3)*z4+(j(3)+j(4))*a(1)
C(13)=-m(3)*z4+m(4)*z1
C(14)=m(4)*z2+m(3)*z7
C(15)=m8*(9.81+a(2))
C(16)=q(3)*z5*(9.81)+z5*a(2)+m(4)*a(3)-(z6+z9+z10)*(m(4)*z2+m(3)*z7)
C(17)=q(4)*z3+(-z9-z10+z6)*(m(4)*z1-m(3)*z4)
C(18)=-3*m(4)*q(3)*z1+m(4)*q(4)*z2+m(3)*q(3)*z4
C(19)=-m(3)*z4+m(4)*z1
C(20)=m(4)*z2+m(3)*z7
C(21)=z11*9.81+z8*a(2)
C(22)=q(3)*(z5*(9.81+a(2))+m(4)*a(3))-(z6+z9+z10)*(m(4)*z2+m(3)*z7)
C(23)=q(4)*z3+(-m(3)*z4+m(4)*z1)*(-z9+z6+z10)
C(24)=m(4)*q(4)*z2-m(3)*q(3)*z4+m(4)*q(3)*z1
return
end

```

Appendice F :

Programme de calcul des forces actives

c*****

c CALCUL DES FORCES ACTIVES

c*****

subroutine dynami (q,v,a,c1,c2,c3, c4)

real v(5),a(5),q(5),c1,c2,c3,c4,m(5),x(5),b,

+s1, s2, s3, s4, s5, s6, s7, s8, s9, s10, s11, s12, s13, s14, s15,
+ s16, s17, s18, s19 , s20, s21, s22, s23, s24, s25, s26, s27, s28,
+s29, s30, r (5)

$$m(1)=25.549$$

$$m(2) =40.70$$

$$m(3) =241.779$$

$$m(4)=39.049$$

$$x(1)=0.6$$

$$x(2) =0.6$$

$$x(3)=0$$

$$x(4)=0.4$$

$$r(1)=0.05$$

$$r(2)=0.05$$

$$r(3)=0$$

$$r(4)=0.06$$

$$b=0.7$$

$$a1=0.2$$

$$s10=m(3) +m(4)$$

$$s11=m(3)*(b*\cos(2*q(1))+2*q(3))$$

$$s12=m(4)*q(3)*(sin(q(1)))**2$$

$$s13=m(2)+m(3)$$

$$s14=m(2) +m(3) +m(4)$$

$$s15=m(4)$$

$$s16=1+(sin(q(1)))**2$$

$$s17=m(1)*x(1)**2$$

$$s18=m(2) * (3*r(2) **2+4*x(2) **2) /12$$

$$s19=m(3)*((a1**2+13*b**2)/12+b*q(3)*\cos(2*q(1))+q(3)**2)$$

$$s20=(3*r(4)**2+4*r(4)**2)/12$$

$$s21=x(4) *q(4) +2*q(3) **2$$

$$s22=2* (q(4) **2+q(3) **2)$$

$$s23=b*\cos(2*q(1))+2*q(3)$$

$$s24=2*b*q(3)*v(1)*sin(2*q(1))$$

$$s25=x(4)*v(4) +4*q(3) *v(3)$$

$$s26 =q(4) *v(4) +q(3)*v(3)$$

```

s27=q(4) **2+q(3)**2
s28=v(1)*sin(q(1))*cos(q(1))
s29=m(3)*b*q(3)*sin(2*q(1))
s30=(q(4)**2+q(3)**2)*cos(q(1))*sin(q(1))
c1=((s17/12)+s18+s19+s15*(s20s21+s22+s16))*a(1)+(m(3)*(s23*v(3)-
+s24)+s15*(4*s26*s16-s25+4*s27*s28))*v(1)+(s29-2*s15*s30)*v(1)-
+q(2) *1000
c2=s13*q(2) +9.81*s14-1000
c3=a(3)*s10- (s11+4*s12) * (v(1) **2)
c4=s15*(a(4)+(x(4)-4*s16*q(4))*v(1)**2)
return
end

```

Appendice G :

Programme de simulation

```

/*
 * JFrame.java
 *
 * Created on 19 mai 2006, 22:54
 */

/**
 *
 * @author Redha
 */
import java.awt.*;
public class JFrame extends javax.swing.JFrame {

    /** Creates new form JFrame */
    public JFrame() {
        initComponents();
        ann= new Animation(this);
        setBounds(100,100,750,480);
        jSlider1.setEnabled(false);

```

```

    setVisible(true);
}

/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is
 * always regenerated by the Form Editor.
 */
private void initComponents() {
    buttonGroup1 = new javax.swing.ButtonGroup();
    jPanel1 = new javax.swing.JPanel();
    jSlider1 = new javax.swing.JSlider();
    jComboBox1 = new javax.swing.JComboBox();
    jRadioButton1 = new javax.swing.JRadioButton();
    jRadioButton2 = new javax.swing.JRadioButton();
    jLabel1 = new javax.swing.JLabel();
    jPanel2 = new javax.swing.JPanel();
    canvas1 = new java.awt.Canvas();
    jMenuBar2 = new javax.swing.JMenuBar();
    jMenu2 = new javax.swing.JMenu();
    jMenuItem2 = new javax.swing.JMenuItem();
    jTable1 = new javax.swing.JTable();

    getContentPane().setLayout(new
javax.swing.BoxLayout(getContentPane(), javax.swing.BoxLayout.X_AXIS));

    addWindowListener(new java.awt.event.WindowAdapter() {
        public void windowClosing(java.awt.event.WindowEvent evt) {
            exitForm(evt);
        }
    });

    jPanel1.setLayout(null);

```

```
jPanel1.setBorder(new javax.swing.border.TitledBorder(""));
jSlider1.setMinimum(1);
jSlider1.setValue(20);
jSlider1.addChangeListener(new
javax.swing.event.ChangeListener() {
    public void stateChanged(javax.swing.event.ChangeEvent evt)
{
    jSlider1StateChanged(evt);
}
});
```

```
jPanel1.add(jSlider1);
jSlider1.setBounds(100, 200, 140, 50);
```

```
jComboBox1.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "Position 1", "Position 2", "Position 3", "Position 4",
"Position 5" }));
```

```
jComboBox1.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt)
{
    jComboBox1ActionPerformed(evt);
}
});
```

```
jPanel1.add(jComboBox1);
jComboBox1.setBounds(100, 100, 130, 25);
```

```
jRadioButton1.setSelected(true);
jRadioButton1.setText("Choisir La Position");
buttonGroup1.add(jRadioButton1);
jRadioButton1.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt)
```

```

{
    jRadioButton1ActionPerformed(evt);
}
});

jPanel1.add(jRadioButton1);
jRadioButton1.setBounds(10, 50, 131, 24);

jRadioButton2.setText("Annimer Le Bras");
buttonGroup1.add(jRadioButton2);
jRadioButton2.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt)
    {
        jRadioButton2ActionPerformed(evt);
    }
});
jTable1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, "X", "Y"},
        {"P1", null, null},
        {"P2", null, null},
        {"P3", null, null},
        {"P4", null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3"
    }
));
jPanel1.add(jTable1);
jTable1.setBounds(80, 270, 200, 80);
jPanel1.add(jRadioButton2);
jRadioButton2.setBounds(10, 150, 130, 24);

```

```
jLabel1.setText("La Vit\u00e8se ");
jPanel1.add(jLabel1);
jLabel1.setBounds(40, 210, 60, 20);

getContentPane().add(jPanel1);

jPanel2.setLayout(new java.awt.GridLayout());

jPanel2.setBackground(new java.awt.Color(255, 255, 255));

jPanel2.setDebugGraphicsOptions(javax.swing.DebugGraphics.NONE_OPTION);
jPanel2.add(canvas1);

getContentPane().add(jPanel2);

jMenu2.setText("Fichier");
jMenuItem2.setText("Quiter");
jMenuItem2.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt)
{
    jMenuItem2ActionPerformed(evt);
    }
});

jMenu2.add(jMenuItem2);

jMenuBar2.add(jMenu2);

setJMenuBar(jMenuBar2);

pack();
}
```

```
private void
jRadioButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // Add your handling code here:
    ann= new Animation(this);
    if(jRadioButton2.isSelected()){
        ann.stop();
        jSlider1.setEnabled(true);
        jComboBox1.setEnabled(false);
        ann.start();
    }
}
```

```
private void
jRadioButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // Add your handling code here:
    if(jRadioButton1.isSelected()){
        ann.stop();
        jSlider1.setEnabled(false);
        jComboBox1.setEnabled(true);
    }
}
```

```
private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent
evt) {
    // Add your handling code here:
    ann.stop();
    System.exit(0);
}
```

```
private void jComboBox1ActionPerformed(java.awt.event.ActionEvent
evt) {
    // Add your handling code here:
    ann= new Animation(this);
    switch(jComboBox1.getSelectedIndex()){
```

```

    case(0):Exemple1();
        position="Position 1";
        break;
    case(1):Exemple3();
        position="Position 2";
        break;
    case(2):Exemple4();
        position="Position 3";
        break;
    case(3):Exemple2();
        position="Position 4";
        break;
    case(4):Exemple5();
        position="Position 5";
        break;
}
ann.start();

}

void jSlider1StateChanged(javax.swing.event.ChangeEvent evt) {
    // Add your handling code here:
}

/** Exit the Application */
private void exitForm(java.awt.event.WindowEvent evt) {
    ann.stop();
    System.exit(0);
}

public Graphics Mypaint(Graphics g){
    for(int jcas=1; jcas<=NbrCas; jcas++){
        XA[jcas][Nbr+1]=(cas*30)+XA[jcas][Nbr];
    }
}

```

```

YA[jcas][Nbr+1]=30+YA[jcas][Nbr];
int x1=150, y1=380, x2=250, y2=350;
int x3=(x1+x2)/2-2;
int y3=y1-y2;
int Xh=0,Yh=0,px=0,py=0;      // Longueur du bras vertical
g.setColor(Color.black);
g.fillRect(x1,y1,100,10);// Base horizontale imposée
g.fillRect(x3,y2,5,y3); // Bras vertical imposé
int j1,j2;
g.setColor(Color.red);
Font f = new Font("Comic Sans MS",14,20);
g.setFont(f);
g.drawString("Simulation d'un robot manipulateur",40,50);
g.drawString(" ( ALG.MO-1 )",150,80);
f = new Font("Comic Sans MS",14,24);
g.setFont(f);
g.drawString(" "+position,Xpos,Ypos);
f = new Font("Comic Sans MS",25,16);
g.setFont(f);
// Trace Bras
for(j1=2; j1<=Nbr+1; j1++){
    g.setColor(Color.black);
    if(j1<=Nbr){
        if(j1<4){ px=10;py=0;}
        if(j1>=4){ py =10;px=0;}
        if((pos == 1||pos==3) && j1==Nbr){ py=-10;px =10;}
        Xh=XA[jcas][j1] + ((XA[jcas][j1-1]- XA[jcas][j1])/2)+px;
        Yh=YA[jcas][j1] + ((YA[jcas][j1-1]- YA[jcas][j1])/2)-py;
        g.drawString("h"+String.valueOf(j1-1),Xh,Yh);
    }
    g.drawLine(XA[jcas][j1-1], YA[jcas][j1-1], XA[jcas][j1],
YA[jcas][j1]);
    System.out.println("J= "+j1+" XA= "+XA[jcas][j1]+" YA=
"+YA[jcas][j1]);

```

```

    }
// Trace Articulations

for(j2=1; j2<=Nbr; j2++){
    if(Type[j2]==1) /* Articulation Rotoide */
    {
        g.setColor(Color.blue);
        g.fillOval(XA[jcas][j2]-R/2, YA[jcas][j2]-R/2, R, R);
        g.setColor(Color.red);
        Font Dialog=new Font("Dialog", Font.BOLD, 13);
        g.setFont(Dialog);
        g.drawString("R", XA[jcas][j2]-R/4, YA[jcas][j2]+R/4);
    }
    else if(Type[j2]==2)/* Articulation Prismatique*/
    {
        g.setColor(Color.red);
        g.fillRect(XA[jcas][j2]-R/2, YA[jcas][j2]-R/2, R, R);
        g.setColor(Color.blue);
        Font Dialog=new Font("Dialog", Font.BOLD, 13);
        g.setFont(Dialog);
        g.drawString("P", XA[jcas][j2]-R/4, YA[jcas][j2]+R/4);
    }
}
jTable1.setValueAt(new Integer(XA[1][2]), 1,1);
jTable1.setValueAt(new Integer(XA[1][3]), 2,1);
jTable1.setValueAt(new Integer(XA[1][4]), 3,1);
jTable1.setValueAt(new Integer(XA[1][5]), 4,1);
jTable1.setValueAt(new Integer(YA[1][2]), 1,2);
jTable1.setValueAt(new Integer(YA[1][3]), 2,2);
jTable1.setValueAt(new Integer(YA[1][4]), 3,2);
jTable1.setValueAt(new Integer(YA[1][5]), 4,2);
return g;
}

```



```
//===== Exemple n°1=====
```

```
public static void Exemple1(){
```

```
// Cas de 1 bras de 5 articulations
```

```
    NbrCas=1;    // Nombre de cas
```

```
    Nbr=5;      // Nombre d'articulation
```

```
    cas =1;
```

```
    pos = 1;
```

```
//Sinq articulations de type RPRP
```

```
    Type[1]=1;  // Articulation Rotoide
```

```
    Type[2]=2;  // Articulation Prismatique
```

```
    Type[3]=2;  // Articulation Prismatique
```

```
    Type[4]=2;  // Articulation Prismatique
```

```
    Type[5]=2;  //Articulation Prismatique
```

```
// Premier cas (premier bras):
```

```
    XA[1][1]=200; // X du premier cas, première articulation
```

```
    YA[1][1]=350; // Y du premier cas, Première articulation
```

```
    XA[1][2]=200; //X du premier cas, 2end articulation
```

```
    YA[1][2]=250; //Y du premier cas, 2end articulation
```

```
    XA[1][3]=200; //X du premier cas, 3ème articulation
```

```
    YA[1][3]=150; //Y du premier cas, 3ème articulation
```

```
    XA[1][4]=250; //X du premier cas, 4ème articulation
```

```
    YA[1][4]=150; //Y du premier cas, 4ème articulation
```

```
    XA[1][5]=250; //X du premier cas, 5ème articulation
```

```
    YA[1][5]=200; //Y du premier cas, 5ème articulation
```

```
    Xpos=30+XA[1][5];
```

```
    Ypos=30+YA[1][5];
```

```
}
```

//===== Exemple n°5 =====

```
public static void Exemple5(){
```

```
// Cas de 1 bras de 5 articulations
```

```
    NbrCas=1;    // Nombre de cas  
    Nbr=5;  
    cas=-1;     // Nombre d'articulation  
    pos =5;
```

```
//Sinq articulations de type RPRP
```

```
    Type[1]=1;  // Articulation Rotoide  
    Type[2]=2;  // Articulation Prismatique  
    Type[3]=2;  // Articulation Prismatique  
    Type[4]=2;  // Articulation Prismatique  
    Type[5]=2;  //Articulation Prismatique
```

```
// Premier cas (premier bras):
```

```
    XA[1][1]=200; // X du premier cas, première articulation  
    YA[1][1]=350; // Y du premier cas, Première articulation
```

```
    XA[1][2]=200; //X du premier cas, 2end articulation  
    YA[1][2]=300; //Y du premier cas, 2end articulation
```

```
    XA[1][3]=200; //X du premier cas, 3ème articulation  
    YA[1][3]=200; //Y du premier cas, 3ème articulation
```

```
    XA[1][4]=150; //X du premier cas, 4ème articulation  
    YA[1][4]=200; //Y du premier cas, 4ème articulation
```

```
    XA[1][5]=60;  //X du premier cas, 5ème articulation  
    YA[1][5]=280; //Y du premier cas, 5ème articulation
```

```
    Xpos=-50+XA[1][5];
```

```
    Ypos=50+YA[1][5];
```

```
// Second cas (2 bras):
```

```
    XA[2][1]=200; // X du 2end cas, première articulation
```

YA[2][1]=300; // Y du 2end cas, Première articulation

XA[2][2]=210; //X du 2end cas, 2end articulation

YA[2][2]=300; //Y du 2end cas, 2end articulation

XA[2][3]=250; //X du 2end cas, 3ème articulation

YA[2][3]=200; //Y du 2end cas, 3ème articulation

XA[2][4]=300; //X du 2end cas, 4ème articulation

YA[2][4]=100; //Y du 2end cas, 4ème articulation

XA[2][5]=200; //X du second cas, 5ème articulation

YA[2][5]=100; //Y du second cas, 5ème articulation

// Troisième cas (3ème bras):

XA[3][1]=200; // X du 3ème cas, première articulation

YA[3][1]=350; // Y du 3ème cas, Première articulation

XA[3][2]=250; //X du 3ème cas, 2end articulation

YA[3][2]=300; //Y du 3ème cas, 2end articulation

XA[3][3]=300; //X du 3ème cas, 3ème articulation

YA[3][3]=200; //Y du 3ème cas, 3ème articulation

XA[3][4]=-10; //X du 3ème cas, 4ème articulation

YA[3][4]=100; //Y du 3ème cas, 4ème articulation

XA[3][5]=-50; //X du troisième cas, 5ème articulation

YA[3][5]=100; //Y du troisième cas, 5ème articulation

}

//===== Exemple n°3=====

public static void Exemple3(){

// Cas de 1 bras de 5 articulations

NbrCas=1; // Nombre de cas
Nbr=5;
cas =1; // Nombre d'articulation
pos=3;

//Sinq articulations de type RPRP

Type[1]=1; // Articulation Rotoide
Type[2]=2; // Articulation Prismatique
Type[3]=2; // Articulation Prismatique
Type[4]=2; // Articulation Prismatique
Type[5]=2; //Articulation Prismatique

// Premier cas (premier bras):

XA[1][1]=200; // X du premier cas, première articulation
YA[1][1]=350; // Y du premier cas, Première articulation

XA[1][2]=200; //X du premier cas, 2end articulation
YA[1][2]=300; //Y du premier cas, 2end articulation

XA[1][3]=200; //X du premier cas, 3ème articulation
YA[1][3]=200; //Y du premier cas, 3ème articulation

XA[1][4]=250; //X du premier cas, 4ème articulation
YA[1][4]=200; //Y du premier cas, 4ème articulation

XA[1][5]=250; //X du premier cas, 5ème articulation
YA[1][5]=250; //Y du premier cas, 5ème articulation

Xpos=30+XA[1][5];
Ypos=30+YA[1][5];

// Second cas (2 bras):

XA[2][1]=200; // X du 2end cas, première articulation
YA[2][1]=350; // Y du 2end cas, Première articulation

XA[2][2]=150; //X du 2end cas, 2end articulation

YA[2][2]=300; //Y du 2end cas, 2end articulation

XA[2][3]=250; //X du 2end cas, 3ème articulation

YA[2][3]=200; //Y du 2end cas, 3ème articulation

XA[2][4]=300; //X du 2end cas, 4ème articulation

YA[2][4]=100; //Y du 2end cas, 4ème articulation

XA[2][5]=200; //X du second cas, 5ème articulation

YA[2][5]=100; //Y du second cas, 5ème articulation

// Troisième cas (3ème bras):

XA[3][1]=200; // X du 3ème cas, première articulation

YA[3][1]=350; // Y du 3ème cas, Première articulation

XA[3][2]=250; //X du 3ème cas, 2end articulation

YA[3][2]=300; //Y du 3ème cas, 2end articulation

XA[3][3]=300; //X du 3ème cas, 3ème articulation

YA[3][3]=200; //Y du 3ème cas, 3ème articulation

XA[3][4]=-10; //X du 3ème cas, 4ème articulation

YA[3][4]=100; //Y du 3ème cas, 4ème articulation

XA[3][5]=-50; //X du troisième cas, 5ème articulation

YA[3][5]=100; //Y du troisième cas, 5ème articulation

}

//===== Exemple n°4 =====

public static void Exemple4(){

// Cas de 1 bras de 5 articulations

NbrCas=1; // Nombre de cas

Nbr=5;

cas=1; // Nombre d'articulation

pos =4;

//Sinq articulations de type RPRP

Type[1]=1; // Articulation Rotoide

Type[2]=2; // Articulation Prismatique

Type[3]=2; // Articulation Prismatique

Type[4]=2; // Articulation Prismatique

Type[5]=2; //Articulation Prismatique

// Premier cas (premier bras):

XA[1][1]=200; // X du premier cas, première articulation

YA[1][1]=350; // Y du premier cas, Première articulation

XA[1][2]=200; //X du premier cas, 2end articulation

YA[1][2]=300; //Y du premier cas, 2end articulation

XA[1][3]=200; //X du premier cas, 3ème articulation

YA[1][3]=200; //Y du premier cas, 3ème articulation

XA[1][4]=250; //X du premier cas, 4ème articulation

YA[1][4]=200; //Y du premier cas, 4ème articulation

XA[1][5]=300; //X du premier cas, 5ème articulation

YA[1][5]=200; //Y du premier cas, 5ème articulation

Xpos=-30+XA[1][5];

Ypos=60+YA[1][5];

// Second cas (2 bras):

XA[2][1]=200; // X du 2end cas, première articulation

YA[2][1]=350; // Y du 2end cas, Première articulation

XA[2][2]=150; //X du 2end cas, 2end articulation

YA[2][2]=300; //Y du 2end cas, 2end articulation

XA[2][3]=250; //X du 2end cas, 3ème articulation

YA[2][3]=200; //Y du 2end cas, 3ème articulation

XA[2][4]=300; //X du 2end cas, 4ème articulation

YA[2][4]=100; //Y du 2end cas, 4ème articulation

XA[2][5]=200; //X du second cas, 5ème articulation

YA[2][5]=100; //Y du second cas, 5ème articulation

// Troisième cas (3ème bras):

XA[3][1]=200; // X du 3ème cas, première articulation

YA[3][1]=350; // Y du 3ème cas, Première articulation

XA[3][2]=250; //X du 3ème cas, 2end articulation

YA[3][2]=300; //Y du 3ème cas, 2end articulation

XA[3][3]=300; //X du 3ème cas, 3ème articulation

YA[3][3]=200; //Y du 3ème cas, 3ème articulation

XA[3][4]=-10; //X du 3ème cas, 4ème articulation

YA[3][4]=100; //Y du 3ème cas, 4ème articulation

XA[3][5]=-50; //X du troisième cas, 5ème articulation

YA[3][5]=100; //Y du troisième cas, 5ème articulation

}

//===== Exemple n°2=====

```
public static void Exemple2(){
```

```
// Cas de 1 bras de 2 articulations
```

```
    NbrCas=1;    // Nombre de cas
```

```
    Nbr=5;
```

```
    cas=-1;    // Nombre d'articulation
```

```
    pos= 2;
```

```
//Sinq articulations de type RPRP
```

```
    Type[1]=1;    // Articulation Rotoide
```

Type[2]=2; // Articulation Prismatique

Type[3]=2; // Articulation Prismatique

Type[4]=2; // Articulation Prismatique

Type[5]=2; //Articulation Prismatique

// Premier cas (premier bras):

XA[1][1]=200; // X du premier cas, première articulation

YA[1][1]=350; // Y du premier cas, Première articulation

XA[1][2]=200; //X du premier cas, 2end articulation

YA[1][2]=300; //Y du premier cas, 2end articulation

XA[1][3]=200; //X du premier cas, 3ème articulation

YA[1][3]=200; //Y du premier cas, 3ème articulation

XA[1][4]=150; //X du premier cas, 4ème articulation

YA[1][4]=200; //Y du premier cas, 4ème articulation

XA[1][5]=120; //X du premier cas, 5ème articulation

YA[1][5]=200; //Y du premier cas, 5ème articulation

Xpos=-100+XA[1][5];

Ypos=50+YA[1][5];

// Second cas (2 bras):

XA[2][1]=200; // X du 2end cas, première articulation

YA[2][1]=350; // Y du 2end cas, Première articulation

XA[2][2]=150; //X du 2end cas, 2end articulation

YA[2][2]=300; //Y du 2end cas, 2end articulation

XA[2][3]=250; //X du 2end cas, 3ème articulation

YA[2][3]=200; //Y du 2end cas, 3ème articulation

XA[2][4]=300; //X du 2end cas, 4ème articulation

YA[2][4]=100; //Y du 2end cas, 4ème articulation

```
XA[2][5]=200; //X du second cas, 5ème articulation
```

```
YA[2][5]=100; //Y du second cas, 5ème articulation
```

```
// Troisième cas (3ème bras):
```

```
XA[3][1]=200; // X du 3ème cas, première articulation
```

```
YA[3][1]=350; // Y du 3ème cas, Première articulation
```

```
XA[3][2]=250; //X du 3ème cas, 2end articulation
```

```
YA[3][2]=300; //Y du 3ème cas, 2end articulation
```

```
XA[3][3]=300; //X du 3ème cas, 3ème articulation
```

```
YA[3][3]=200; //Y du 3ème cas, 3ème articulation
```

```
XA[3][4]=-10; //X du 3ème cas, 4ème articulation
```

```
YA[3][4]=100; //Y du 3ème cas, 4ème articulation
```

```
XA[3][5]=-50; //X du troisième cas, 5ème articulation
```

```
YA[3][5]=100; //Y du troisième cas, 5ème articulation
```

```
}
```

```
/**
```

```
* @param args the command line arguments
```

```
*/
```

```
public static void main(String args[]) {
```

```
    new JFrame().show();
```

```
}
```

```
// Variables declaration - do not modify
```

```
private javax.swing.ButtonGroup buttonGroup1;
```

```
public java.awt.Canvas canvas1;
```

```
private javax.swing.JComboBox jComboBox1;
```

```
private javax.swing.JLabel jLabel1;
```

```

private javax.swing.JMenu jMenuItem2;
private javax.swing.JMenuBar jMenuItemBar2;
private javax.swing.JMenuItem jMenuItem2;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
public javax.swing.JRadioButton jRadioButton1;
private javax.swing.JRadioButton jRadioButton2;
public javax.swing.JSlider jSlider1;
private javax.swing.JTable jTable1;
public Animation ann;
public String position;
// End of variables declaration
static int R=20;
    static int Nbr; //Nombre d'articulation
    static int cas;
static int pos =0,Xpos,Ypos;
    static int NbrCas;//=Nombre de bras
    static int Type[]=new int[50]; //Type d'articulation
    static int XA[][]=new int[50][50]; //Coordonnées suivant X
    static int YA[][]=new int[50][50]; //Coordonnées suivant Y

// La Calss Animation Pour géré l'animation du Bras
class Animation extends Thread {
    JFrame MyForm;
    Animation(JFrame MyForm){
        this.MyForm = MyForm;
    }

    public void run() {
        if(MyForm.jRadioButton1.isSelected()){
            MyForm.canvas1.getGraphics().clearRect(0,0,500,500);
            MyForm.Mypaint(MyForm.canvas1.getGraphics());
        }
        else{

```


}

////////////////////////////////////**Fin du programme**////////////////////////////////////

Appendice H :

Programme de model géométrique direct

C PROGRAM: MGD

c PURPOSE: Entry point for the console application.

c program MGD

c Variables

c*****

implicit none

integer i,j

real theta1 ,h1,h2,h3,h4,z2,z5,z3,z4

real c1,s1

real d3,d4,Pi

real T(4,4)

d4=0.6

C selon le cahier de charge

d3=d4

Pi=3.1415926535897932384626433832795

write(*,*) 'donner la valeur de theta1'

read(*,*) theta1

theta1=(theta1 *Pi)/180.

write(*,*) 'theta1r=',theta1

write(*,*) 'donner la valeur de h1=,h2=,h3=,h4=,z2=,z5=,z3=,z4='

read(*,*) h1,h2,h3,h4,z2,z5,z3,z4

c

write (*,*)'calcul des éléments de la matrice de transformation

+ homogène T<0,5>'

c

c1 =cos(theta1)

s1=sin(theta1)

write(*,*) 'c1 =cos(theta1)=',c1 ',s 1 =sin(theta1)=', s1

```
write (*,*)'la matrice t(i,j)'
```

```
    T(1,1)=c1-s1
```

```
T(1,2)=0
```

```
T(1,3)=0
```

```
T(1,4)=(z3-z4)*c1
```

```
T(2,1)=c1+s1
```

```
T(2,2)=0
```

```
T(2,3)=0
```

```
T(2,4)=(z3-z4)*s1
```

```
T(3, 1)=0
```

```
T(3,2)=0
```

```
T(3,3)=1
```

```
T(3,4)=h1+h2+h3+h4+z2-z5
```

```
T(4, 1)=0
```

```
T(4,2)=0
```

```
T(4,3)=0
```

```
T(4,4)=1
```

```
write(*,*)' ((T(i,j)j=1,4),i=1,4)'
```

```
5  format (4F)
```

```
do 2 i=1,4
```

```
write( *,*)(T(i,j),j=1,4)
```

```
2  continue
```

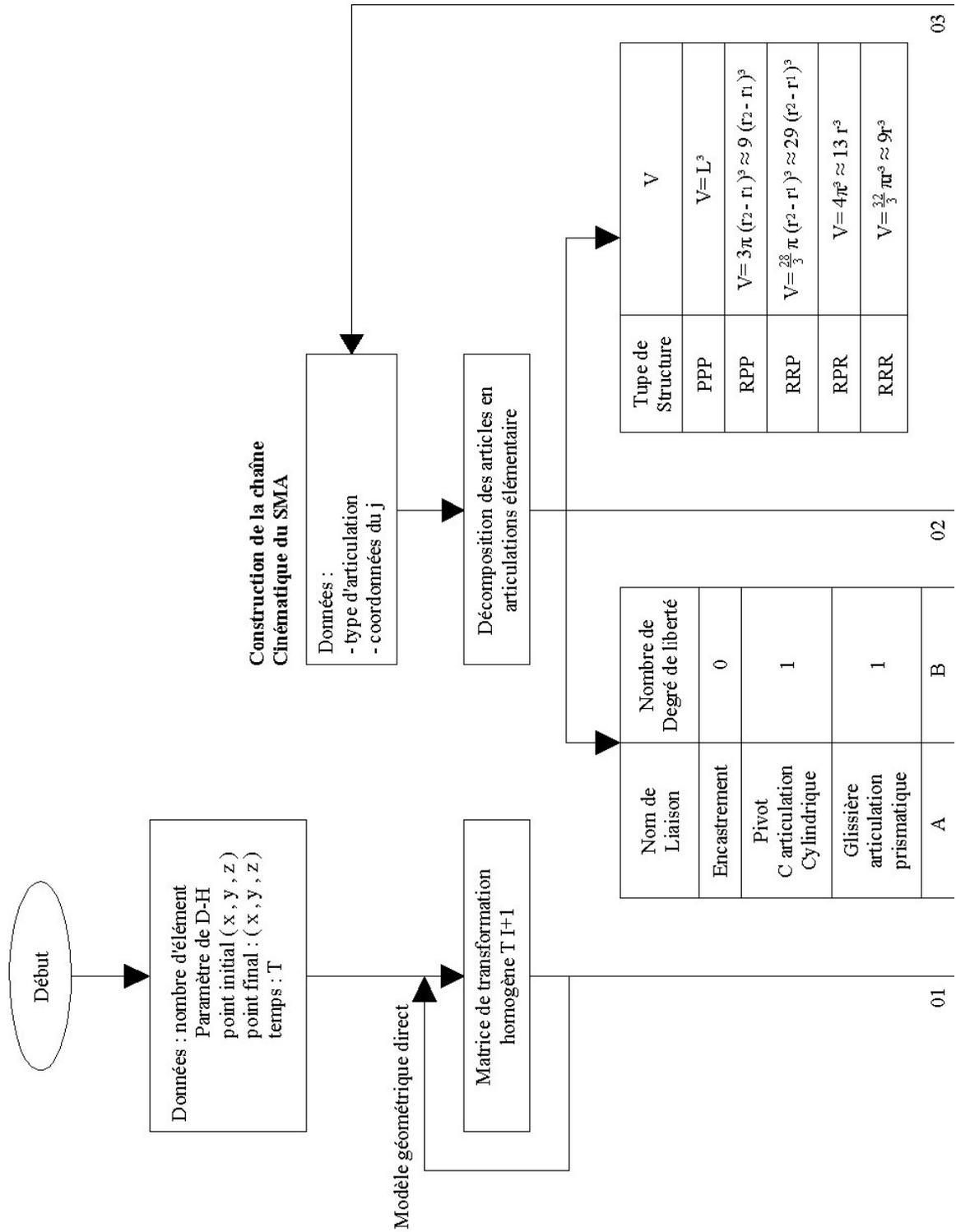
```
write(*,*) 'CE PROGRAMME MARCHE TRES BIEN '
```

```
c  Body ofMGD
```

```
end.
```

APPENDICE I :

Organigramme général : (Proposé comme une banque des données) :



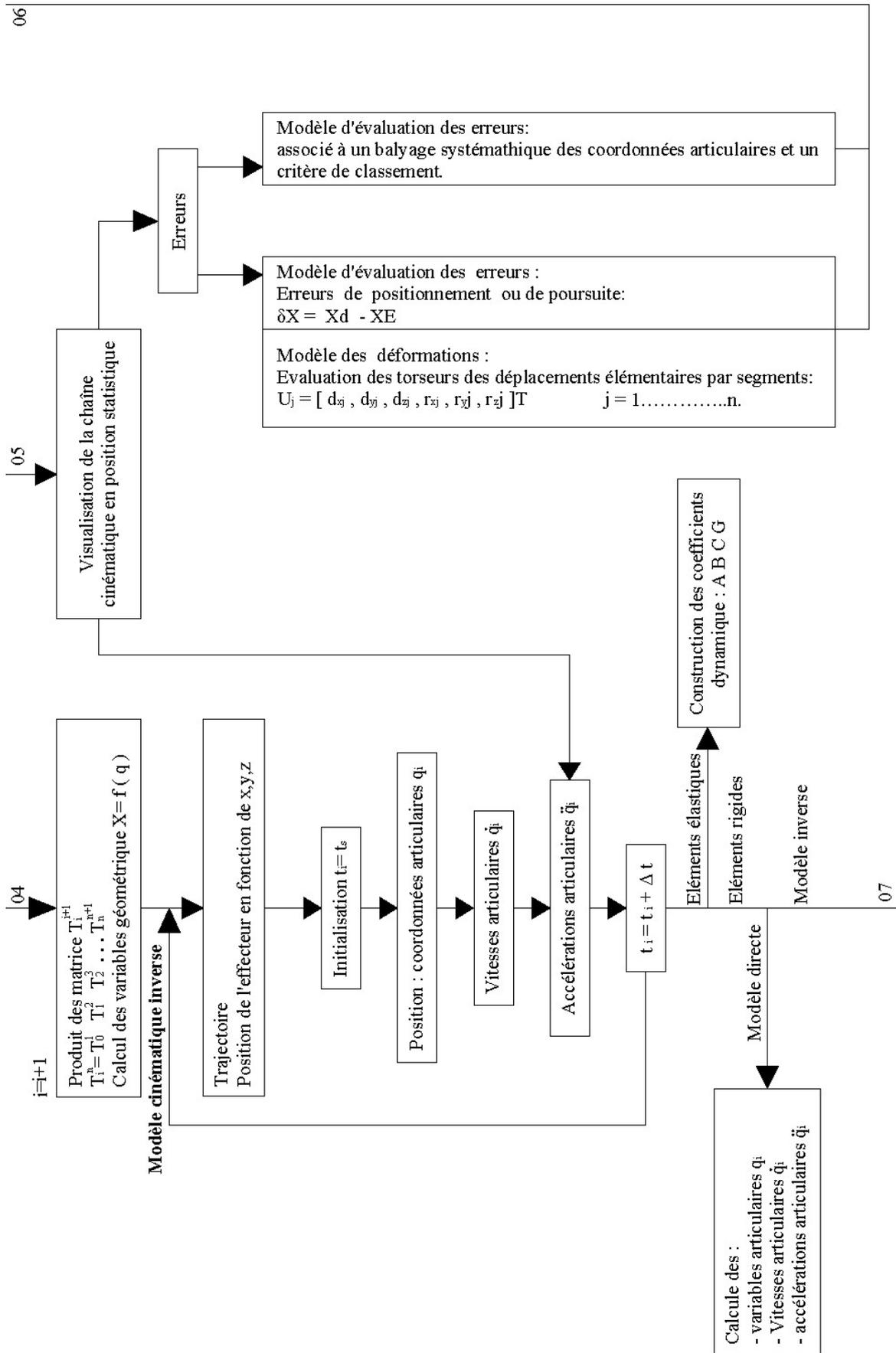
A	B
Glissière hélicoïdale	1
Pivot Glissant	2
Appui Plan	3
Rotule Articulation rotoïde	3
Linéaire rectiligne	4
Linéaire annulaire	4
Ponctuelle	5
Libre	6

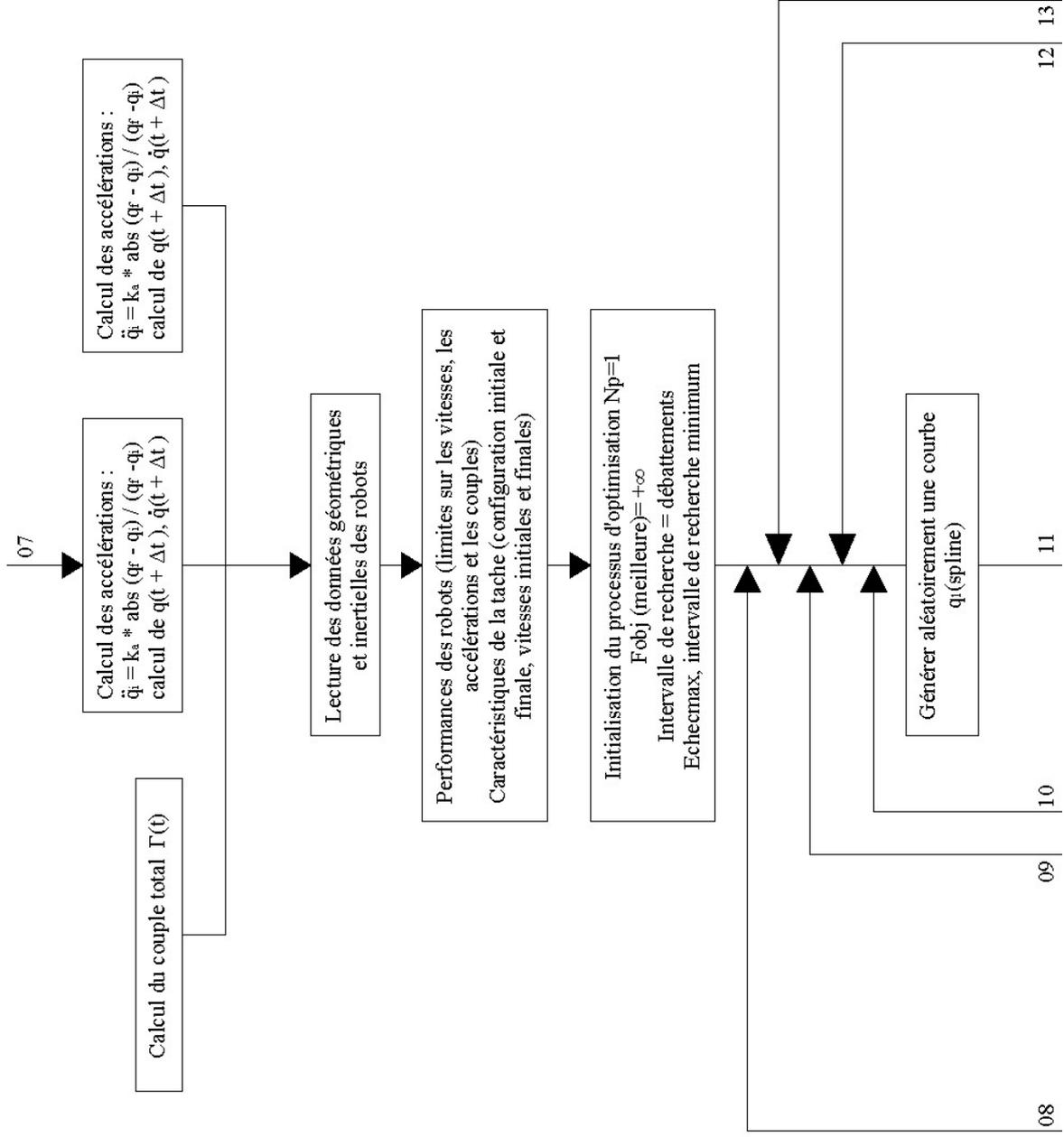


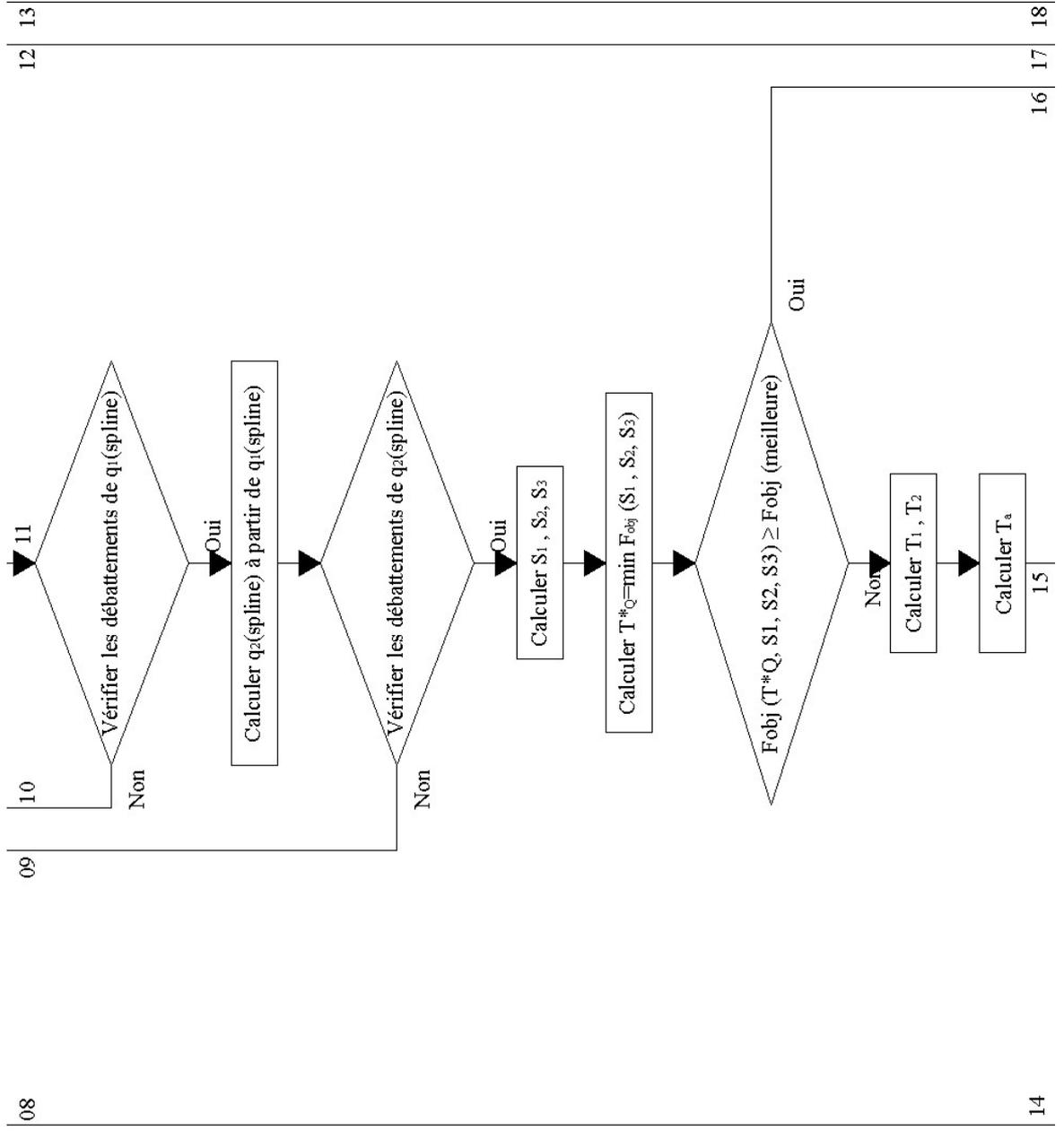
type de porteurs :

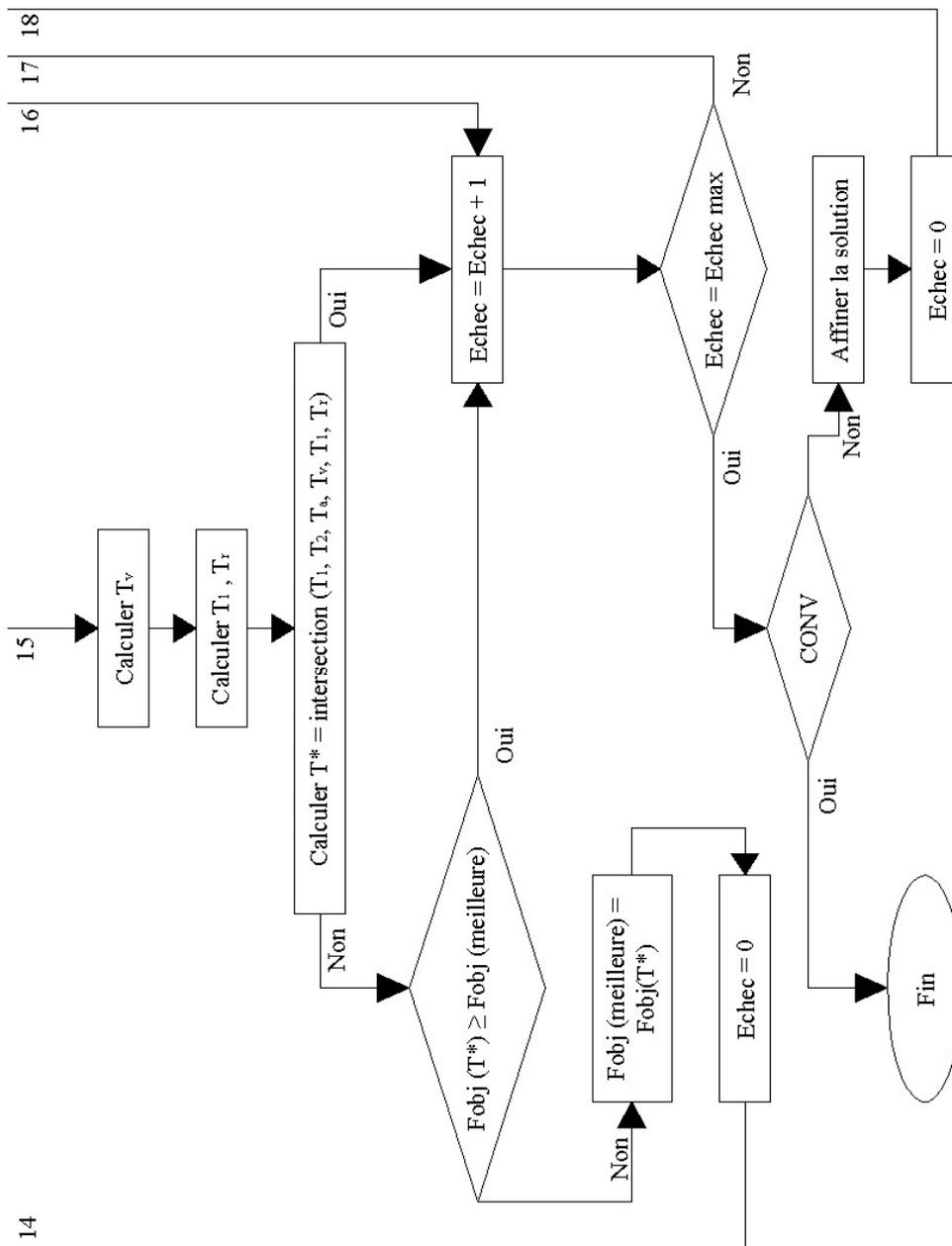
- structure cartésienne (T.T.T) (3.d.d.l.)
- structure sphérique (ou polaire).
- structure angulaire (3R)
- structure SCARA

(1) Cartésien P/P/P	(2) P/P/P	(3) Cylindrique P/P/P	(4) R/P/R
(5) R/P/R	(6) R/P/R	(7) P/R/R	(8) sphérique R/R/P
(9) SACARA R/R/P	(10) R/R/R	(11) Angulaire R/R/R	(12) R/R/R









Organigramme I.1 : Organigramme général

Appendice K [21] :

Dérivation de la matrice de transformation homogène :

Soit T_{ij} la matrice de transformation du repère R_i au repère r_j qu'on peut mettre sous forme :

$$T_{ij} = T_i(i+1).T_{i+1}(i+1).T_{(i+1)(i+2)} \dots T_{(j-1)j} \text{ elle dépend des variables } q_{i+1}, \dots, q_j.$$

Pour calculer la dérivée partielle de T_{ij} par rapport à la variable q_1 .

On distingue les deux cas suivants :

1^{er} cas : $i \geq 1 > j$ Les éléments de la matrice T_{ij} ne dépend pas de q_1 alors :

$$\frac{\partial T_{ij}}{\partial q_1} = [0]$$

2^{ème} cas : $i < 1 \leq j$

$$\frac{\partial T_{ij}}{\partial q_1} = \frac{\partial}{\partial q_1} [T_{i(i+1)} T_{(i+1)(i+2)} \dots T_{(j-1)j}] = T_{i(i+1)} T_{(i+1)(i+2)} \dots T_{(1-2)(1-1)} \left[\frac{\partial T_{(1-1)1}}{\partial q_1} \right] T_{(1-1)1} \dots T_{(j-1)j}$$

Sachant que :

$$\left[\frac{\partial T_{(i-1)i}}{\partial q_i} \right] = T_{(i-1)i} \cdot \Omega_{i1}$$

$$\Omega_{i1} = \begin{bmatrix} 0 & -(1-\sigma_1) & 0 & 0 \\ (1-\sigma_1) & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$\sigma_j = 0$ pour une liaison rotoïde

$\sigma_j = 1$ pour une liaison prismatique.

$$\frac{\partial T_{ij}}{\partial q_1} = T_{i(i-1)} T_{(1-1)1} \Omega_{11} T_{ij} = T_{i1} \Omega_{11} T_{1j}$$

$$\frac{\partial T_{ij}}{\partial q_1} = T_{i1} \Omega_{11} T_{1j}$$

On pose $\Omega_{j1} = T_{j1} \cdot \Omega_{11} \cdot T_{ij}$ et on peut écrire :

$$\frac{\partial T_{ij}}{\partial q_1} = T_{i1} \Omega_{11} T_{1j} = T_{ij} (T_{j1} \Omega_{11} T_{1j}) = T_{ij} \Omega_{j1}$$

Notons ici que :

$$\begin{aligned}\Omega_{i1} T_{ij} &= (T_{i1} \Omega_{11} T_{1i}) T_{ij} = T_{ij} \Omega_{j1} \\ &= T_{ij} (T_{j1} \Omega_{11} T_{1j}) = T_{ij} \Omega_{j1}\end{aligned}$$

Donc on a : $\Omega_{ij} T_{ij} = T_{ij} \Omega_{j1}$

Finalement on peut écrire :

$$\frac{\partial T_{ij}}{\partial q_1} = \begin{cases} [0] si, i > 1 > j \\ T_{ij} \cdot \Omega_{j1} si, i < 1 \leq j \end{cases}$$

Matrice d'un système constitué de n corps :

Soit T_{0n} la matrice de transformation homogène du repère R_0 au repère R_n alors :

$$\frac{\partial T_{0n}}{\partial q_1} = \begin{cases} [0] si, 0 \geq 1 > n \\ T_{0n} \cdot \Omega_{n1} si, 0 < 1 \leq n \end{cases}$$

Appendice L

L-1 La trajectoire : [12, 3, 4, 21, 27, 71, 90, 91] ; Selon la tâche que doit accomplir le robot, on peut classer les différents mouvements selon deux catégories de trajectoires l'une libre l'autre imposée, [90, 91], (voir figure L.7, L.8, L.9).

La génération de trajectoire est une étape très importante dans la commande des robots manipulateurs. Elle consiste à calculer les consignes de référence en position, en vitesse, et en accélération, qui décrivent leurs mouvements désirés

Cependant, la trajectoire est l'évolution de la position, et ses dérivées temporelles en fonction du temps pour chacune des articulations; le mouvement le plus simple est d'aller d'un point initial à un point final, ainsi le robot est commandé pour changer sa configuration initiale vers une configuration finale, ce type de mouvement convient aux tâches de transfert d'objets quand l'espace de travail ne comporte aucun obstacle ; le chemin à suivre par l'élément terminal peut être contraint par l'addition de points intermédiaires aux configurations initiales et finales.

Pour assurer le fonctionnement normal du mécanisme, on choisira des mouvements continus, pour cela on définit une fonction lisse, (dérivées première et secondaire continues), afin d'éviter les risques d'usure (voir chapitre des causes structurales) et de vibrations pouvant exciter les modes propres du manipulateur.

En résumé, le parcours peut être planifié de différentes manières dont on distingue :

Le mouvement entre les deux points avec une trajectoire libre entre les points.

Le mouvement entre deux points via des points intermédiaires avec une trajectoire libre entre les points intermédiaires.

Le mouvement entre deux points avec une trajectoire contrainte entre les points (trajectoire rectiligne).

Le mouvement entre deux points via des points intermédiaires avec une trajectoire contrainte entre les points intermédiaires.

Dans les deux premiers cas, la génération de la trajectoire peut se faire directement dans l'espace articulaire. Quant aux deux derniers, la trajectoire étant décrite dans l'espace opérationnelle, il est préférable de raisonner dans cet espace.

L-1-1 Génération de la trajectoire dans l'espace articulaire : [12, 3, 4, 21, 27] : La génération de trajectoire dans l'espace articulaire donne au résultat un ensemble de données : Positions, vitesses et accélérations articulaires ($\theta_d, \dot{\theta}_d, \ddot{\theta}_d$) qui sont utilisées comme un signal de référence, pour une position initiale et finale données dans l'espace de travail, on utilise la géométrie inverse pour

déterminer les angles articulaires correspondant à cette position et même pour l'orientation. La position initiale du manipulateur devient un ensemble d'angles articulaires d'arrivée, ce qui est demandé pour la planification de la trajectoire est de trouver une fonction lisse pour chaque articulation dont la valeur à l'instant t_0 est la position initiale de l'articulation et dont la valeur à l'instant t_f est la position désirée de cette même articulation. Il y a beaucoup de fonctions lisses $\theta(t)$, qui pourraient être utilisées pour interpoler les valeurs d'angles articulaires. On cite :

L-1-1-1 La fonction polynomiale cubique [26, 27] : Il s'agit d'une interpolation simple en effectuant un simple mouvement continu, au moins quatre contraintes sur $\theta(t)$ sont évidentes pour avoir une fonction polynomiale cubique de la forme :

$$\theta(t) = a_0 + a_1t + a_2t^2 + a_3t^3 \dots\dots\dots(L.1)$$

Ainsi on obtient la vitesse articulaire :

$$\dot{\theta}(t) = a_1 + 2a_2t + 3a_3t^2 \dots\dots\dots(L.2)$$

et aussi l'accélération articulaire :

$$\ddot{\theta}(t) = 2a_2 + 6a_3t \dots\dots\dots(L.3)$$

Sachant qu'un polynôme du 3^{ème} degré, admet quatre coefficients, il peut être donc construit à partir de quatre contraintes, deux sont obtenues à partir du choix des valeurs initiales et finales de la position :

$\theta(0) = \theta_0$, $\theta_d(t_f) = \theta_f$ quant aux deux autres contraintes, elles proviennent du fait que l'articulation démarre et arrive avec une vitesse nulle $\dot{\theta}(0) = 0$, $\dot{\theta}_d(t_f) = 0$ et en combinant les deux fonctions :

$\theta(t)$ et $\dot{\theta}(t)$ avec les quatre contraintes, on obtient quatre équations à quatre inconnues :

$$\theta_0 = a_0 \dots\dots\dots(L.4)$$

$$\theta_f = a_0 + a_1t_f + a_2t_f^2 + a_3t_f^3 \dots\dots\dots(L.5)$$

$$0 = a_1 \dots\dots\dots(L.6)$$

$$0 = a_1 + 2a_2t_f + 6a_3t_f^2 \dots\dots\dots(L.7)$$

En relevant ce système d'équations on obtient :

$$a_0 = \theta_0 \dots\dots\dots(L.8)$$

$$a_1 = 0 \dots\dots\dots(L.9)$$

$$a_2 = \frac{3}{t_f^2}(\theta_f - \theta_0) \dots\dots\dots(L.10)$$

$$a_3 = -\frac{2}{t_f^3}(\theta_f - \theta_0) \dots\dots\dots(L.11)$$

Avec ces quatre coefficients, on peut calculer le polynôme cubique qui connecte n'importe quelle position désirée.

Exemple de génération d'une trajectoire articulaire (voir figure L.7, L.8, L.9).

L-1-1-2 Polynôme de degré cinq : [12, 26] : On peut avoir des polynômes d'ordre supérieur pour la planification de la trajectoire, ils sont quelquefois utilisés dans le cas où on voudrait indiquer la position, la vitesse, et l'accélération au début et à la fin de la trajectoire. Ils sont aussi utilisés quand les robots manipulateurs fonctionnent à une grande vitesse. Il est nécessaire d'assurer la continuité des accélérations afin d'éviter l'excitation du mécanisme. Il faut avoir un polynôme d'ordre cinq de la forme :

$$\theta(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \dots\dots\dots(L.12)$$

Pour déterminer les coefficients de ce polynôme, il faut avoir au moins six contraintes, deux sur la vitesse et deux autres sur l'accélération :

$$\theta_0 = a_0 \dots\dots\dots(L.13)$$

$$\dot{\theta}_0 = a_1 \dots\dots\dots(L.14)$$

$$\ddot{\theta}_0 = 2a_2 \dots\dots\dots(L.15)$$

$$\theta_f = a_0 + a_1t_f + a_2t_f^2 + a_3t_f^3 + a_4t_f^4 + a_5t_f^5 \dots\dots\dots(L.16)$$

$$\ddot{\theta}_f = 2a_2 + 6a_3t_f + 12a_4t_f^2 + 20a_5t_f^3 \dots\dots\dots(L.17)$$

En résolvant le système d'équation, on trouve :

$$a_0 = \theta_0 \dots\dots\dots(L.18)$$

$$a_1 = \dot{\theta}_0 \dots\dots\dots(L.19)$$

$$a_2 = \ddot{\theta}_0 \dots\dots\dots(L.20)$$

$$a_3 = \frac{20\theta_f - 20\theta_0 - (8\dot{\theta}_f + 12\dot{\theta}_0)t_f + (3\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^3} \dots\dots\dots(L.21)$$

$$a_4 = \frac{30\theta_0 - 30\theta_f + (14\dot{\theta}_f + 16\dot{\theta}_0)t_f - (3\ddot{\theta}_0 - 2\ddot{\theta}_f)t_f^2}{2t_f^4} \dots\dots\dots(L.22)$$

$$a_5 = \frac{12\theta_f - 12\theta_0 - (6\dot{\theta}_f + 6\dot{\theta}_0)t_f - (\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^5} \dots\dots\dots(L.23)$$

L-1-1-3 Loi Bang-Bang avec palier de vitesse (loi trapèze) : [12, 26] : Une autre voie pour générer les consignes est ce qu'on appelle segment linéaire avec portions paraboliques ou bien loi Bang-Bang avec palier de vitesse. Cette trajectoire est telle que la vitesse est rampée en haut à sa valeur spécifique initialement et puis rampée en bas au début (position désirée). Pour accomplir tout. Cela on spécifie la position désirée en trois phases.

La première phase de l'instant t_0 à l'instant t_b est polynomiale quadratique.

A l'instant t_b la position change de forme et devient linéaire cela correspond à une vitesse constante.

Enfin à l'instant t_f la trajectoire de position redevient polynomiale quadratique de sorte que la vitesse soit linéaire.

On choisit l'instant t_b de façon à ce que la courbure de position soit symétrique, par convention, on suppose que $t_0 = 0$ et $q(t_f) = q(0) = 0$ puis entre les instants 0 et t_b on a :

$$q_d(t) = a_1 + a_1 t + a_2 t^2 \dots\dots\dots (L.24)$$

de façon que la vitesse soit :

$$q_d(t) = a_1 + a_2 t \dots\dots\dots (L.25)$$

Les contraintes $q(t_f) = 0$ et $q(0) = 0$ impliquent que :

$$a_0 = q_0 \dots\dots\dots (L.26)$$

$$a_1 = 0 \dots\dots\dots (L.27)$$

Puisque à l'instant t_b on veut que la vitesse soit égale à une constante donnée on aura :

$$q_d(t_b) = 2a_2 t_b = V \dots\dots\dots (L.28)$$

$$\text{ce qui implique que : } a_2 = V / 2t_b \dots\dots\dots (L.29)$$

Par conséquent, la trajectoire requise entre les instants 0 et t_b est donnée par :

$$q_d(t) = q_0 + V / 2t_b \cdot t^2 = q_0 + a / 2 \cdot t^2 \dots\dots\dots (L.30)$$

$$q_d(t) = q_0 + V / t_b \cdot t = at \dots\dots\dots (L.31)$$

$$q_d = V/t_b \cdot t = a \dots\dots\dots (L.32)$$

Où a est l'accélération.

Maintenant et entre les instants t_f et $t_f - t_b$, la trajectoire est un segment linéaire qui correspond à une vitesse constante V :

$$q_d(t) = \alpha_0 + \alpha_1 t = \alpha_0 + V t \dots\dots\dots (L.33)$$

Par symétrie : $q_d(t) = \frac{q_0 + q_1}{2} \dots\dots\dots (L.34)$

On aura :
$$\frac{q_0 + q_1}{2} = \alpha_0 + \frac{V_{tf}}{2} \dots\dots\dots (L.35)$$

Ce qui implique que :

$$\alpha_0 = \frac{q_0 + q_1 - V_{tf}}{2} \dots\dots\dots(L.36)$$

Comme les deux segments coïncident à l'instant t_b on obtient :

$$q_0 + \frac{Vt_b}{2} = \frac{q_0 + q_1 V_{t_2} + V_{t_b}}{2} \dots\dots\dots(L.37)$$

Ce qui donne l'expression :

$$t_b = \frac{q_0 + q_1 l - V}{2} \dots\dots\dots (L.38)$$

Il est à noter qu'on a la contrainte : $0 < t_b < \frac{t_f}{2} \dots\dots\dots(L.39)$

Ceci mène à l'inéquation :

$$\frac{q_f - q_0}{V} < t_f \leq \frac{2(q_f - q_0)}{V} \dots\dots\dots(L.40)$$

Autrement dit :
$$\frac{q_f + q_0}{t_f} < V \leq \frac{2(q_f - q_0)}{t_f} \dots\dots\dots (L.41)$$

Par conséquent, la vitesse spécifiée doit être entre ces deux limites ou bien le mouvement ne sera pas possible, la portion de la trajectoire entre $t_f - t_b$ et t_f est maintenant déterminée par des considérations de symétrie. La trajectoire complète est donnée par :

$$q_d(t) = \begin{cases} \frac{q_0 + a.t^2}{2} \dots\dots\dots 0 < t_d \leq t_f \\ \frac{q_f + q_0 - V_{tf} + V_t}{2} \dots\dots\dots t_d < t < t_f - t_b \\ \frac{q_f - a.t_f^2}{2} + \frac{a.t_f t - 2t^2}{2} \dots\dots\dots t_f - t_b < t \leq t_f \end{cases} \dots\dots\dots(L.42)$$

L-1-2 Génération de la trajectoire dans l'espace cartésien : Les trajectoires dans l'espace articulaire sont plus faciles à calculer et elles sont plus simples, cependant elles deviennent complexes si elles sont décrites dans l'espace cartésien puisque dans le premier cas on ne fait aucune correspondance continue entre l'espace articulaire et l'espace cartésien; il n'y a aucun problème avec les singularités du mécanisme.

La trajectoire dans l'espace cartésien donne comme résultat la position, la vitesse et l'accélération cartésienne (x, \dot{x}, \ddot{x}) . Pour certaines structures de commande spécifique, ces données doivent être converties en grandeurs équivalentes dans l'espace articulaire. Une analyse complète devrait être utilisée comme un signal de référence pour le régulateur synthétisé dans l'espace articulaire.

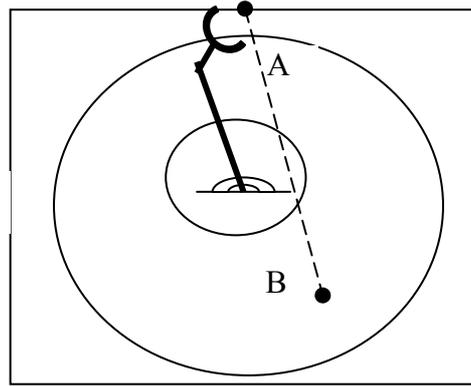
Dans l'espace de travail, la forme du parcours prise par l'élément terminal n'est pas simple, mais plutôt une forme compliquée qui dépend de la géométrie du manipulateur. Les formes du parcours sont décrites en terme des fonctions du temps qui déterminent la position et l'orientation cartésiennes, la forme spatiale du parcours entre ces points peut être : Une ligne droite, circulaire, sinusoïdale, ellicoïdale ou d'autres formes.

Pour la planification de trajectoire basée sur l'espace cartésien, les fonctions qui forment une trajectoire sont des fonctions du temps qui représentent les variables cartésiennes. Ces parcours peuvent être planifiés directement à partir de la définition, par l'utilisation des points de parcours qui sont les spécifications sur le parcours désiré.

L-1-3 Problème géométrique dans un parcours cartésien : [27] : Plusieurs problèmes sont posés dans la génération des trajectoires cartésiennes à cause de la correspondance continue qui se trouve entre la forme de la trajectoire décrite dans l'espace cartésien et celle décrite dans l'espace articulaire. On en cite trois problèmes majeurs :

Problème du type 1 : C'est le problème des points intermédiaires inaccessibles, comme il est présenté dans la figure L.1, si on veut que le robot démarre du point A pour arriver au point B en passant par une trajectoire désirée qui est une ligne droite par exemple, bien que les points initiaux et finals du parcours soient compris dans l'espace de travail du robot, il y a certains points qui appartiennent à cette trajectoire et qui n'appartiennent pas à l'espace de travail du manipulateur. Ces points deviennent inaccessibles, ce qui rend impossible d'effectuer cette tâche dans l'espace opérationnel. Néanmoins il n'y aurait pas ce type de problème dans l'espace articulaire.

Figure L.1 : Problème de type 2



Problème de type 2 : C'est le problème de vitesse articulaire qui est trop élevée en passant près de la singularité. Certains parcours dans l'espace cartésien sont impossibles à exécuter par le robot manipulateur surtout dans le cas où il devrait suivre une trajectoire en s'approchant d'une singularité du mécanisme, (par exemple une ligne).

Une ou plusieurs vitesses articulaires peuvent augmenter vers l'infini. Puisque les vitesses des articulations sont bornées supérieurement, cette situation a pour conséquence la déviation du robot manipulateur de son parcours désiré. La figure L.1 montre un robot à deux articulations, ayant la même longueur, se déplaçant le long du parcours du point A au point B, la trajectoire désirée correspond au mouvement de l'élément terminal du manipulateur à vitesse linéaire constante le long d'un parcours rectiligne. Sur la figure, on montre plusieurs positions intermédiaires du manipulateur qui ont été dessinées pour mieux voir le mouvement, et tous les points de la trajectoire sont accessibles, mais dès que la vitesse de l'articulation 01 devient très élevée, plus le parcours s'approche étroitement de l'axe articulaire 01, plus la vitesse sera grande.

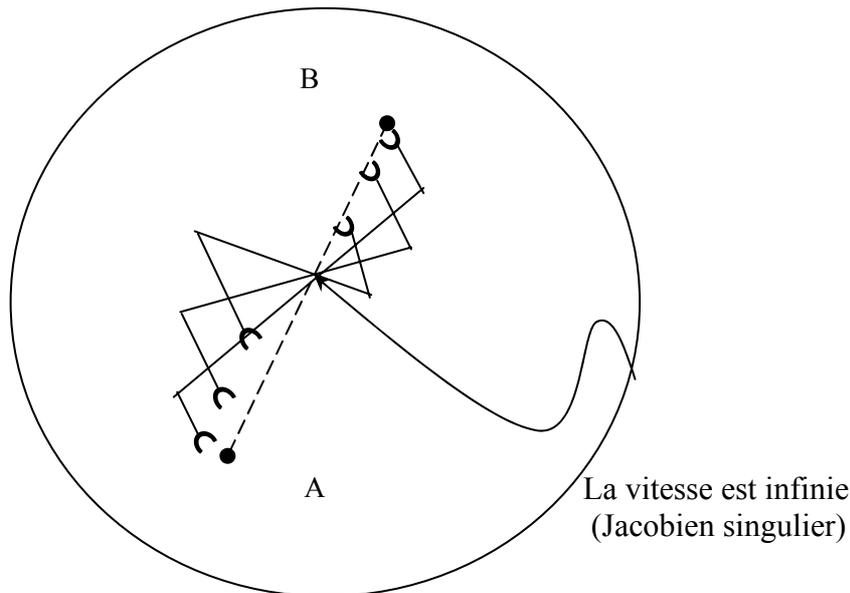


Figure L.2 : Problème de type 3.

Problème de type 3 : Le troisième problème se produit quand il y a plusieurs solutions pour atteindre un point donné dans la trajectoire, la figure L.2 montre un robot manipulateur à deux liaisons de même longueur, ayant des butées au niveau des articulations qui diminue le nombre de solutions avec lesquelles il peut atteindre un point donné dans l'espace. En particulier, un problème surviendrait si le point final ne peut pas être atteint dans la même solution physique quand le robot est au point de départ.

Comme il se voit dans la figure L.1, le manipulateur peut atteindre tous les points du parcours pour une certaine solution, mais pas pour n'importe quelle solution. Dans cette situation le système ne peut pas se déplacer à cause des butées mécaniques.

L-1-4 Étude comparative et discussion du choix d'une trajectoire : [12, 27] : Les deux approches étudiées précédemment présentent plusieurs avantages et inconvénients. La génération du mouvement dans l'espace articulaire présente les avantages suivants :

Elle nécessite moins de calculs en ligne puisqu'il n'y a pas d'appel au modèle géométrique ou cinématique inverse. Le mouvement n'est pas affecté par le message, par les configurations singulières. Les contraintes de vitesse et de couples maximaux sont directement déduites des limites physiques des actionneurs. En contre partie, la géométrie de la trajectoire de l'organe terminal dans l'espace opérationnel est imprévisible bien qu'elle soit répétitive : Il y a donc un risque de collision, lorsque le robot évolue dans un environnement encombré. Ce type de mouvement est par conséquent approprié pour réaliser des déplacements rapides dans un espace libre. La génération des mouvements dans l'espace opérationnel permet de contrôler la géométrie de la trajectoire. En revanche:

Elle exige la transformation en coordonnées articulaires de chaque point de la trajectoire. Elle peut être mise en échec lorsque la trajectoire calculée passe par une position singulière.

Elle est mise en échec chaque fois que les points de la trajectoire engendrée ne sont pas dans l'espace accessible par le robot ou chaque fois que la trajectoire impose une reconfiguration du mécanisme (changement d'aspect en cours de trajectoire); les limites en vitesse et en couple dans l'espace opérationnel varient selon la configuration du robot. On exprime alors ces limites par des valeurs traduisant des performances moyennes, satisfaisantes quelle que soit la configuration du robot. On impose donc au robot de travailler au-delà de ses capacités réelles.

Le choix d'une méthode de génération de mouvement dépend de l'application considérée. Chaque approche a ses propres limites, inhérente au fait que les contraintes sont exprimées soit dans l'espace articulaire (vitesse, couples, butées) soit dans l'espace opérationnel (précision, obstacles).

L-1-5 Génération de trajectoire dans l'espace cartésien :

L-1-5-1 Description d'une trajectoire planaire : On veut générer une trajectoire dans l'espace cartésien, pour imposer une position et une orientation au manipulateur à suivre, on veut que l'élément terminal du bras manipulateur suive un demi cercle qui se situe dans le plan OXZ, le centre du cercle est repéré par les coordonnées suivantes (x', z') et dont le rayon est R. on impose une orientation à l'outil terminal de façon à ce qu'il soit radial au sens rentrant et en arrivant à l'extrémité de l'arc il conserve son orientation perpendiculairement au tronçon rectiligne.

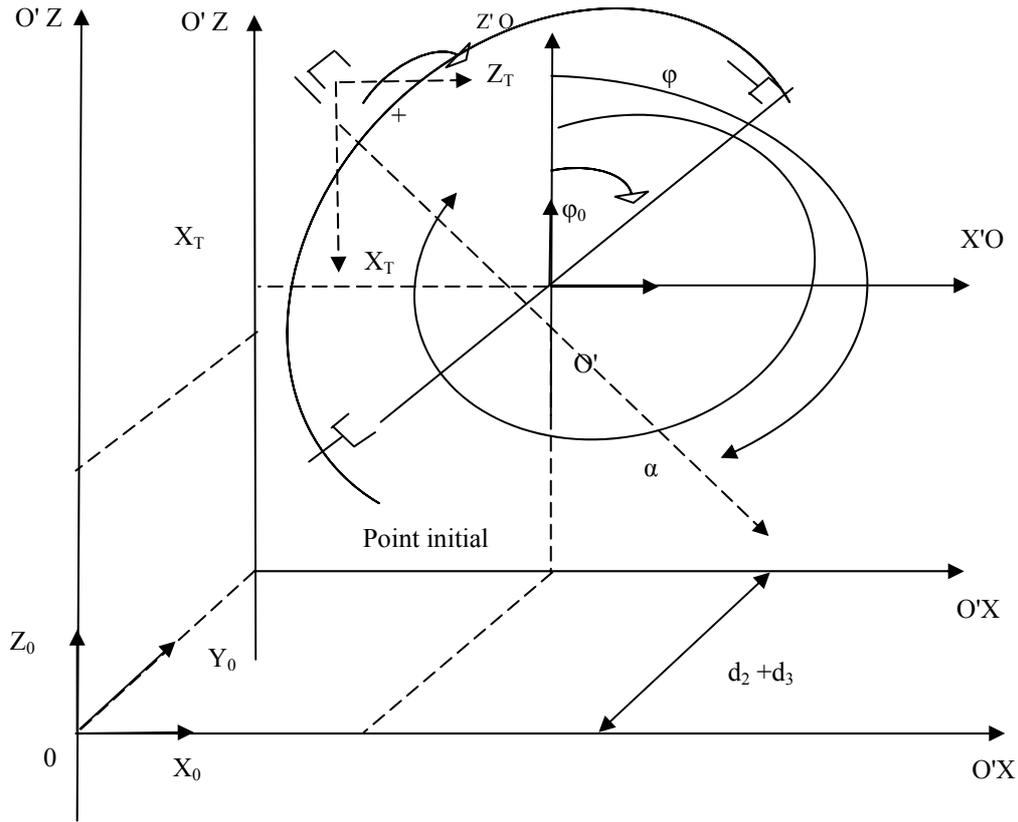


Figure L.3 : Schéma représentatif d'une trajectoire dans le plan.

On désire que l'outil parcoure le demi cercle dans le sens de la flèche. L'orientation de l'outil terminal est représentée par l'angle $\varphi(t)$ pour lequel le point concerné sur le demi cercle est repéré par l'angle $\alpha(t)$.

Ainsi nous obtenons les équations suivantes :

$$Z_t = \begin{cases} x_t = ox' + r \sin(\alpha) \\ z_t = oz' + R \cos(\alpha) \\ y_t = d_2 + d_3 = oy' \end{cases} \dots\dots\dots (L.43)$$

On impose à l'angle φ une variation linéaire en fonction de α :

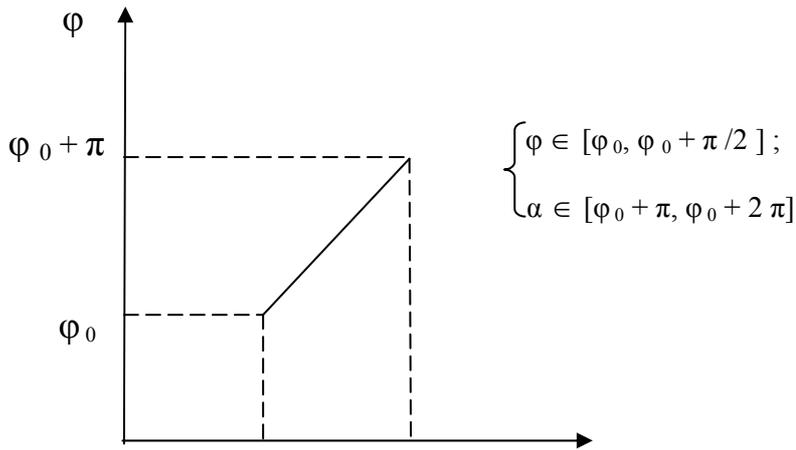


Figure L.4 : Variation de l'angle φ en fonction du α .

D'où $\varphi(t) = 1/2\alpha(t) + 1/2\alpha_0 - \pi/2$ (L.44)

L-1-5-1-1 Génération de trajectoire sur le 1er tronçon: L'évolution de l'angle $\alpha(t)$ est imposée telle qu'il suive une loi Bang –Bang avec palier de vitesse.

En résumé :

Position :

$$Z_t = \begin{cases} x_t = ox' + r \sin(\alpha) \\ \alpha z' + R \cos(\alpha) \\ y_t = d_2 + d_3 = oy \\ \alpha = \alpha(t) \\ \varphi(t) = 1/2\alpha(t) + 1/2\alpha_0 - \pi/2 \end{cases} \dots\dots\dots(L.45)$$

Vitesse :

$$\begin{cases} \dot{x}_t = R \cos(\alpha) \\ \dot{z}_t = -R \sin(\alpha) \\ \dot{y}_t = 0 \\ \alpha(t) = \alpha(t)/2 \end{cases} \dots\dots\dots(L.46).$$

La vitesse :

$$V_{TX} = V_{Z''} \sin(\varphi_0) \begin{cases} V_{TY} = 0 \\ V_{TZ} = V_{Z''} \cos(\varphi_0) \\ \varphi = 0 \end{cases} \dots\dots\dots(L.49).$$

L'accélération :

$$a_{TX} = a_{Z''} \sin(\varphi_0) \begin{cases} a_{TY} = 0 \\ a_{TZ} = a_{Z''} \cos(\varphi_0) \\ \varphi = 0 \end{cases} \dots\dots\dots(L.50).$$

L-1-5-2 Description d'une trajectoire spatiale : On veut que l'outil terminal suit la trajectoire décrite ci-dessus (glissement sur la face d'un huitième de sphère de rayon R et de centre O' (x'_0, y'_0, z'_0) donné. L'orientation est calculée pour assurer la direction radiale entrante de l'outil afin qu'il puisse réaliser la tâche demandée (polissage par exemple) sachant que nous avons adopté la représentation d'Euler concernant les orientations.

Les angles φ et ψ sont des paramètres auxiliaires qui servent à déterminer la position et l'orientation de l'outil terminal dans les repères lié à la sphère, et par une simple translation du vecteur [x'_0, y'_0, z'_0] on retrouvera les coordonnées cartésiennes de l'outil dans le repère de la base.

En fonction des paramètre φ et ψ on peut exprimer le vecteur des positions et orientations comme suit :

$$\begin{cases} x = R \sin \varphi \cos \psi + x'_0 \\ y = R \sin \varphi \sin \psi + y'_0 \\ z = R \cos \varphi + z'_0 \\ \alpha = \psi \\ \beta = \varphi - \pi \\ \delta = \psi \end{cases} \dots\dots\dots(L.51)$$

Avec $\begin{cases} \varphi \in [-\pi/2, 0] \\ \psi \in [0, \pi/2] \end{cases} \dots\dots\dots(L.52)$

Ou : x , y et z représentent la position de l'outil et α , β et δ sont les angles d'Euler Correspondants à l'orientation de l'outil terminal exprimées dans le repère de base .

Les angles φ et ψ obéissent à une loi d'évaluation dans le temps du type Bang Bang avec palier de vitesse comme il est schématisé ci-dessous.

Les vitesses :

$$\begin{cases} \dot{x} = R(\cos \varphi \dot{\varphi} \cos \psi + \sin \varphi \dot{\varphi} \sin \psi) \\ \dot{y} = R(\cos \varphi \dot{\varphi} \sin \psi + \sin \varphi \dot{\varphi} \cos \psi) \\ \dot{z} = -R \dot{\varphi} \sin \varphi \\ \alpha = \psi \\ \beta = \varphi \\ \delta = \psi \end{cases} \dots\dots\dots(L.53)$$

Les accélérations :

$$\begin{cases} \ddot{x} = R(-\sin(\varphi)\dot{\varphi}^2 \cos \psi + \cos(\varphi)\ddot{\varphi} \cos \psi + -\cos(\varphi)\dot{\varphi} \sin(\psi)\dot{\psi} - \\ \cos(\varphi)\dot{\varphi} \sin(\psi)\dot{\psi} - \sin \varphi \cos(\psi)\dot{\psi}^2 - \sin \varphi \sin(\psi)\ddot{\psi} \\ \ddot{y} = R(-\sin(\varphi)\dot{\varphi}^2 \sin \psi + \cos(\varphi)\ddot{\varphi} \sin \psi + \cos(\varphi)\dot{\varphi} \cos(\psi)\dot{\psi} + \\ \cos(\varphi)\dot{\varphi} \sin(\psi)\dot{\psi} + \sin \varphi \cos(\psi)\dot{\psi}^2 + \sin \varphi \sin(\psi)\ddot{\psi} \\ \ddot{z} = -R(\cos(\varphi)\dot{\varphi}^2 + \sin(\varphi)\ddot{\varphi}) \\ \alpha = \psi \\ \beta = \varphi \\ \delta = \psi \end{cases} \dots\dots\dots(L.54)$$

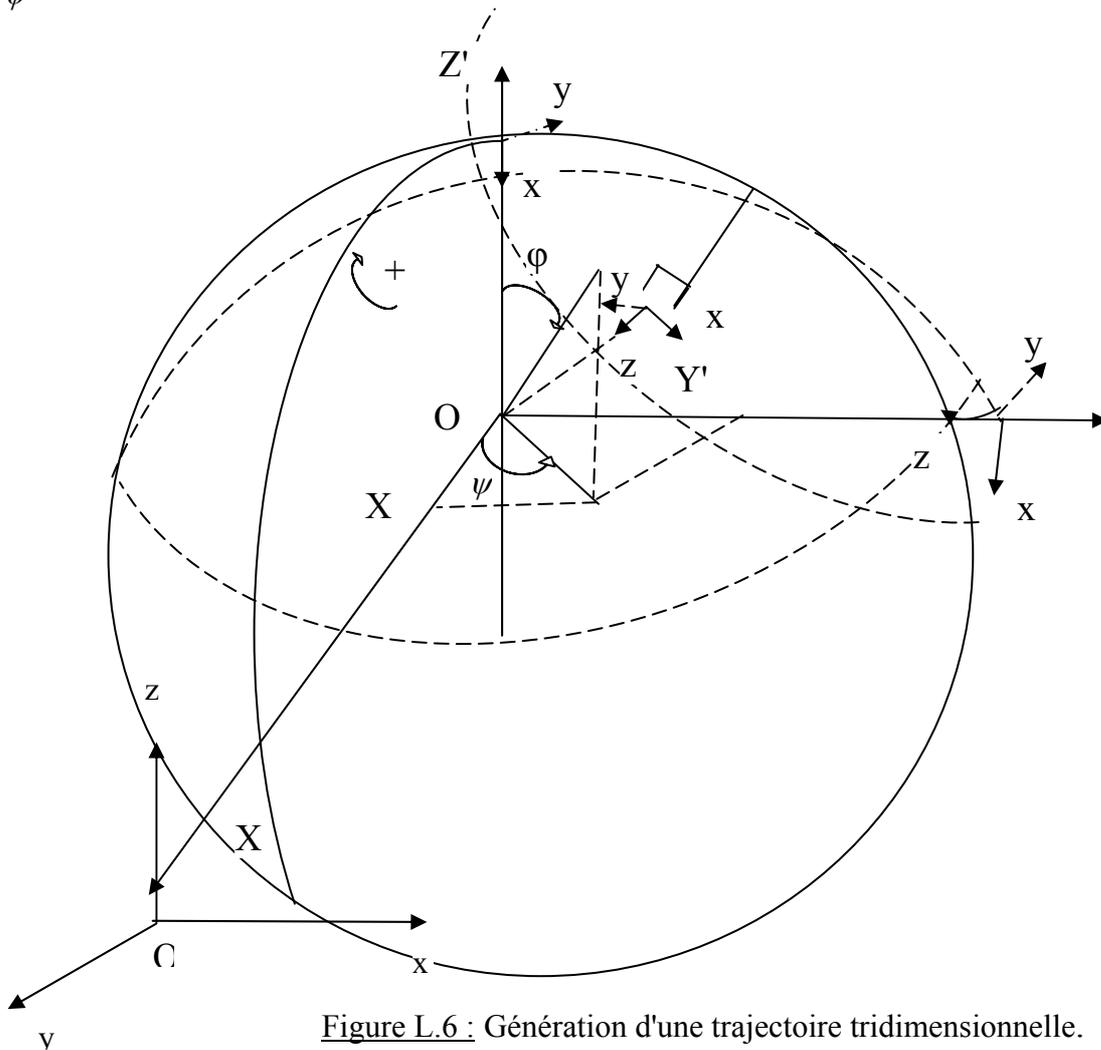


Figure L.6 : Génération d'une trajectoire tridimensionnelle.

Exemple de trajectoires de la position articulaires, vitesse angulaire et accélération articulaire.

On veut générer la trajectoire articulaire pour passer d'un angle initial de 5° à un angle final de 65° dans 4 secondes [27,28].

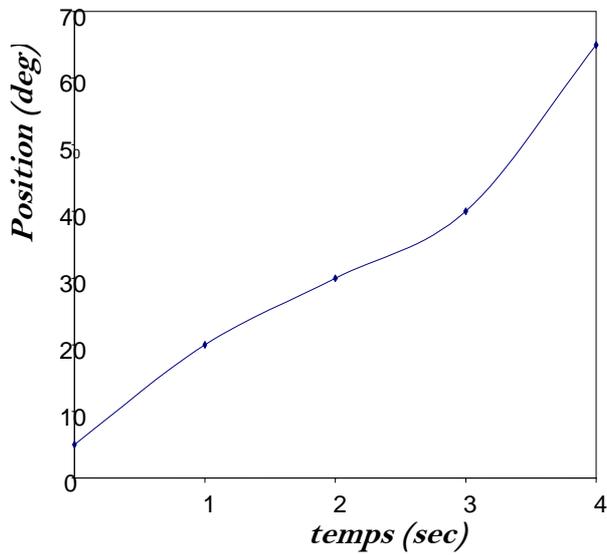


Figure L.7 : Position articulaire

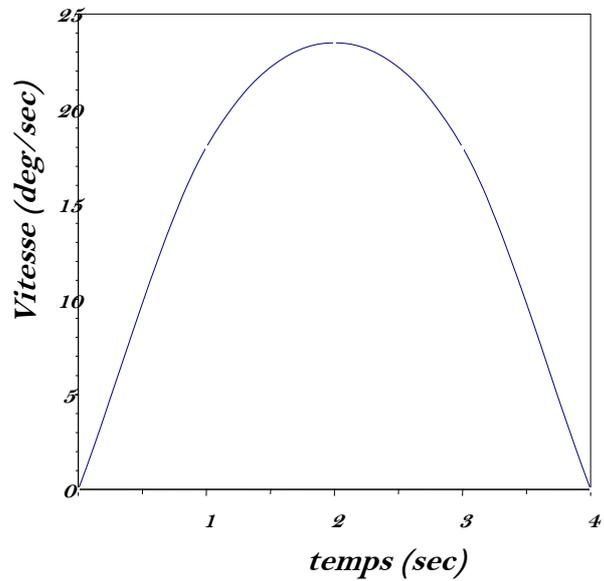


Figure L.8 : Vitesse Angulaire

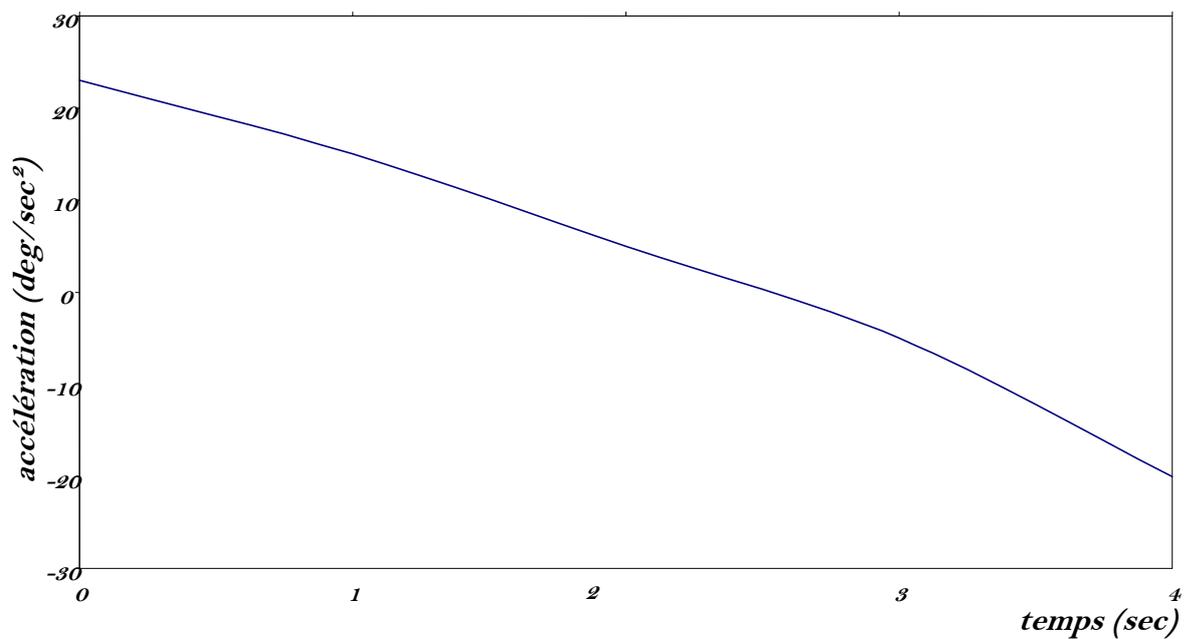


Figure L.9 : accélération articulaire

Planification des mouvements

Introduction sur la planification :

L'utilisation courante manipulateurs dans l'exécution des tâches répétitives fait que les recherches se sont naturellement orientées vers la planification de mouvements minimisant un coût relatif essentiellement le rendement du robot et son fonctionnement. Cette idée générale a conduit à la formulation du problème de planification dans lesquels des grandeurs physiques telles que la durée de parcours, les efforts actionneurs ou la puissance consommée soient optimisés individuellement ou globalement.

Définition :

Trajectoire libre : appelée aussi mouvement point à point ou mouvement de transfert. Seules les configurations initiale et finale doivent être respectées en plus des obstacles à éviter. La trajectoire qui relie ces deux configurations est alors libre. Ceci peut être le cas de la manutention d'objets ou de soudure point à point.

Trajectoire imposée : de tel mouvement sont rencontrés lorsque l'outil en fin de chaîne sur son environnement sans interruption et selon un parcours déterminé.

Il est nécessaire de spécifier la trajectoire de l'effecteur. On citera par exemple les travaux de découpage ou de soudure en continu. Dans ces deux cas, il existe généralement plusieurs trajectoires possibles. Il faut profiter de cette multiplicité de choix pour adapter la meilleure solution afin d'accomplir la tâche. Pour les mouvements libres, l'optimisation a pour but de rechercher la trajectoire à suivre de ainsi que les modalités pour la parcourir tandis que pour les mouvements imposés, l'optimisation ne porte que sur les modalités de la parcourir.

Fonction objectif :

Le choix du critère de performance est déterminant quant à la qualification du mouvement et conditionne l'efficacité des méthodes de résolution du problème. Le critère à minimiser peut se présenter sous 3 formes distinctes : une fonctionnelle de type intégrale, de type terminal ou de type ou de type mixte. Bien que formulés différemment, ces 3 problèmes sont équivalents, la fonctionnelle intégrales est le milieu adaptée pour les mouvements de transfert. Les critères à optimiser dans ce cas se mettent sous la forme générale suivante [71, 90,91] :

$$F_{(u,t)} = \int_0^T L(x(t), u) dt \dots\dots\dots(L.56)$$

Ou :

L : représente le lagrangien qui prend la forme selon l'objectif visé.

X(t) : vecteur de variables d'état à l'instant $t \in [0, T]$

U : vecteur qui représente la commande à optimiser, considère comme une inconnue du problème.

T : temps de transfert total entre les configurations initiale et finale.

Pour le cas d'une planification de mouvements libres, u représente les couples moteurs, x(t) les

variables articulaires de position q(t), de vitesse $\dot{q}(t)$ et d'accélération $\ddot{q}(t)$

Dans ce cas, le critère s'écrit de la manière suivante :

$$F(\Gamma, T) = \int_0^T L(q(t), \dot{q}(t), \ddot{q}(t), \Gamma) dt \dots\dots\dots(L.57)$$

Pour générer des mouvements de qualité, on peut citer parmi les critères les plus significatifs la durée de parcours, mixte temps – efforts quadratiques et mixte avec puissance quadrique .

Critère durée de parcours :

Il suscite l'intérêt d'un bon nombre de chercheurs roboticiens [92] car pour les robots qui travaillent d'une manière cyclique, la rapidité d'exécution (rentabilité) est un critère très significatif. La dynamique du robot n'est pas représenté directement ans cette formulation. Ceci conduit à des commandes optimales discontinues.

Les couples articulaires ne sont pas représentés dans le lagrangien d'où une forme simple comparée aux autres.

$$L(q(t), \dot{q}(t), \ddot{q}(t), \Gamma) = 1 \dots\dots\dots(L.58)$$

$$\text{Alors : } F(T, \Gamma) = F(T) = \int_0^T dt = T \dots\dots\dots(L.59)$$

Critère mixte temps efforts quadratiques :

Comparativement au critère précédant implique des discontinuités lors des transferts de charges, donc, nuisible au bon fonctionnement du manipulateur, ce critère est un moyen efficace pour régularisé les vitesses et les couples, en introduisant un facteur de pondération μ , on peut favoriser soit le temps, soit les efforts (couples) [93], le lagrangien prend la forme suivante :

$$L(q(t), \dot{q}(t), \ddot{q}(t), \Gamma) = \mu + \frac{1-\mu}{2} \sum_{i=1}^n \left(\frac{\Gamma_i(t)}{\Gamma_{i \max}} \right)^2 \quad \text{avec } \mu \in]0,1[\dots\dots\dots(L.60)$$

$\Gamma_{i, \max}$ représente le couple maximum délivré par la $i^{\text{ème}}$ articulation. Quand μ se rapproche de 0, l'optimisation des couples moteurs tandis que quand μ est au voisinage de 1, on privilégie plutôt

l'optimisation du temps. Le choix DE μ d2termine alors le compromis entre ces deux quantités. Ceci conduit à une certaine souplesse d'exécution en tenant compte de la durée de la tâche. Néanmoins, on rencontre des difficultés dans la formation de ce critère vu l'existence des efforts dans le lagrangien.

Les contraintes :

Pour que la planification du mouvement soit possible, on tient compte des capacités technologiques du robot et des spécificités de la tâche à effectuer. Ceci indique une restriction des solutions. Pour planifier un mouvement, on s'intéresse aux contraintes suivantes :

Contraintes sur les débattements :

Du fait de la conception de la liaison et que les mouvements sont limités entre des butées mécaniques, les mouvements ne doivent pas dépasser les capacités de la structure. Tout ceci se traduit par la formule suivante :

$$\forall t \in [0, T], |q_i(t)| \leq q_{i\max} \quad i=1, \dots, n \dots \dots \dots (L.61)$$

T : représente le temps

$q_i(t)$: représente les coordonnées généralisées de la $i^{\text{ème}}$ articulation.

$q_{i\max}$: représente la valeur maximale de la $i^{\text{ème}}$ articulation (valeur intrinsèque du robot)

Contraintes cinématiques :

La conception des articulations et la technologie des actionneurs impliquent une limitation des caractéristiques cinématiques notamment pour les objets manipulés sensibles aux survitesse, accélération et freinages (par exemple des liquides qui peuvent se déverser).

Les formulations se traduisent par :

Vitesse :

$$\forall t \in [0, T], \left| \dot{q}_i(t) \right| \leq \dot{q}_{i\max} \quad i=1, \dots, n \dots \dots \dots (L.62)$$

Accélérations :

$$\forall t \in [0, T], \left| \ddot{q}_i(t) \right| \leq \ddot{q}_{i\max} \quad i=1, \dots, n \dots \dots \dots (L.63)$$

$\dot{q}_{i\max}$ et $\ddot{q}_{i\max}$ sont imposées par les caractéristiques techniques et la nature de la tâche à effectuer .

Contraintes sur les couples moteurs:

Les capacités maximales des actionneurs impliquent la formule suivante [94]

$$\forall t \in [0, T], |\Gamma_i(t)| \leq \Gamma_{i\max} \quad i=1, \dots, n \dots \dots \dots (L.64)$$

Contraintes dues aux obstacles dans l'espace opérationnel :

L'encombrement dû à la présence de robots fait que la planification tient compte de ces obstacles. Ces contraintes rendent le problème très complexe. Pour éviter les collisions et si g traduit la distance entre le robot et l'obstacle, la formulation mathématique est la suivante [95,96,97] :

En plus des dysfonctionnements déjà cités, la génération des mouvements peut aussi entraîner les phénomènes de survitesses au niveau des articulations [98]. La puissance instantanée est telle que :

$$P(t) = \Gamma(t) \cdot \dot{q}(t) \dots\dots\dots(L.65)$$

Le lagrangien s'écrit dans ce cas :

$$L(q(t), \dot{q}(t), \ddot{q}(t), \Gamma) = \mu + \frac{1-\mu}{2} \left[\alpha \sum_{i=1}^n \left(\frac{\Gamma_i(t)}{\Gamma_{i \max}} \right)^2 + (1-\alpha) \sum_{i=1}^n \left(\frac{p_i(t)}{p_{i \max}} \right)^2 \right] \dots\dots\dots(L.66)$$

Où : $p_{i \max} = \Gamma_{i \max} \cdot \dot{q}_{i \max}$

$\dot{q}_{i \max}$ représente la vitesse articulaire maximale tolérée de la $i^{\text{ème}}$ articulation.

Quant à α , il joue le même rôle que μ mais sur le terme des puissances quadratiques si veut accorder plus ou moins d'importance à la minimisation de ces dernières .

$$G(q(t)) \leq 0.$$

Appendice M :

Dérivée de la matrice de transformation homogène par rapport au temps :

$$dT_y \sum_{k=i+1}^l \frac{\partial T_y}{\partial q_k} dq_k$$

$$\frac{dT_y}{dt} = \sum_{k=i+1}^l \frac{\partial T_y}{\partial q_k} \cdot \frac{dq_k}{dt}$$

Il existe deux formes pour la dérivée de T_y par rapport au temps :

$$\frac{dT_y}{dt} = \sum_{k=i+1}^l \Omega_{ik} \cdot T_y q_k = \left(\sum_{k=i+1}^l \Omega_{ik} \cdot q_k \right) T_y$$

$$\frac{dT_y}{dt} = \sum_{k=i+1}^l T_y \Omega_{ik} \cdot q_k = T_y \left(\sum_{k=i+1}^l \Omega_{ik} \cdot q_k \right)$$

$$\text{Donc } \frac{dT_y}{dt} = \left(\sum_{k=i+1}^l \Omega_{ik} \cdot q_k \right) T_y = T_y \left(\sum_{k=i+1}^l \Omega_{ik} \cdot q_k \right)$$

Appendice N :

Expression de l'énergie cinétique

Expression de la matrice d'inertie A :

Soit M_k un point appartenant au solide S_k tel que :

$$O_0 M_k = T_{0k}(\overrightarrow{O_k M_k}) \quad \|\overrightarrow{O_k M_k}\| = cte \quad \text{cas des rigides}$$

$$V^0(M_k) = \frac{d^0}{dt}(O_0 M_k) = \frac{d^0}{dt}(T_{0k})[O_k M_k]$$

Soit deux vecteurs \vec{a} .et . \vec{b} alors $\vec{a} \cdot \vec{b} = \sum_{i=1}^n a_i b_i = Trace([a][b]^T)$

$$[V^0(M_k)]^2 = Trace([V^0(M_k)][V^0(M_k)]^T)$$

$$[V^0(M_k)]^T = [O_k M_k]^T \left(\sum_{i=1}^k \frac{\partial T_{0k}}{\partial q_i} \cdot q_i \right)^T = [O_k M_k]^T \left(\sum_{i=1}^k \frac{\partial T_{0k}}{\partial q_i} \right)$$

L'énergie cinétique du solide S_k est donnée par :

$$E_{0k} = \frac{1}{2} \int [V^0(M_k)]^2 dm$$

$$E_{0k} = \frac{1}{2} \int Trace \left[\sum_{i=1}^k \left[\frac{\partial T_{0k}}{\partial q_i} \right] q_i [O_k M_k] [O_k M_k]^T \sum_{i=1}^k \left[\frac{\partial T_{0k}}{\partial q_i} \right] q_i \cdot dm \right]$$

$$E_{0k} = \frac{1}{2} \sum_{i=1}^k \sum_{l=1}^k Trace \left[\sum_{i=1}^k \left[\frac{\partial T_{0k}}{\partial q_i} \right] \int [O_k M_k] [O_k M_k]^T dm \left[\frac{\partial T_{0k}}{\partial q_i} \right] q_i \cdot q_l \right]$$

$$[J_K] = \int [O_k M_k] [O_k M_k]^T dm$$

$[J_k]$ c'est la pseudo matrice d'inertie de dimension (4) relative au corps S_k de la chaîne dans le repère R_k .

Elle est constitué a partir :

Du moment d'ordre zéro $d_v S_k$: m_k masse de S_k .

Des moments d'ordre un de S_k : G_k centre de masse de S_k .

Des moments d'ordre deux S_k qui représente la matrice d'inertie $[I_k]$ en O_k dans le repère R_k .

$$[I_k] = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}$$

$$[J_k]_k = \begin{bmatrix} (-I_{xx} + I_{yy} + I_{zz})/2 & -I_{xy} & -I_{xz} & \overline{mx} \\ -I_{xy} & (I_{xx} - I_{yy} + I_{zz})/2 & -I_{yz} & \overline{my} \\ -I_{xz} & -I_{yz} & (I_{xx} + I_{yy} - I_{zz})/2 & \overline{mz} \\ \overline{mx} & \overline{my} & \overline{mz} & m \end{bmatrix}$$

$$E_{0k} = \frac{1}{2} \sum_{j=1}^k \sum_{i=1}^k \text{Trace} \left[\left[\frac{\partial T_{0k}}{\partial q_j} \right] [J_k] \left[\frac{\partial T_{0k}}{\partial q_i} \right] \right] \dot{q}_j \cdot \dot{q}_i$$

L'énergie cinétique total du système est donnée par :

$$E_c = \sum_{i=1}^n E_{0k}$$

$$E_c = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^k \sum_{l=1}^k \text{Trace} \left[\left[\frac{\partial T_{0k}}{\partial q_j} \right] [J_k] \left[\frac{\partial T_{0k}}{\partial q_l} \right] \right] \dot{q}_j \cdot \dot{q}_l$$

En tenant compte de l'annexe 3 on a :

$$\frac{\partial T_{0k}}{\partial q_l} = \begin{cases} [0] & \text{si } j > k \\ T_{0k} \cdot \Omega_{0j} & \text{si } j \leq k \end{cases}$$

tenant compte de l'annexe 3 on a :

$$\frac{\partial T_{0k}}{\partial q_l} = \begin{cases} [0] & \text{si } l > k \\ T_{0k} \cdot \Omega_{0l} & \text{si } l \leq k \end{cases}$$

$$\frac{\partial T_{0k}}{\partial q_l} = \begin{cases} [0] & \text{si } l > k \\ T_{0k} \cdot \Omega_{0l} & \text{si } l \leq k \end{cases}$$

Puisque $\frac{\partial T_{0k}}{\partial q_j} = [0]$ si $j > k$ et $\frac{\partial T_{0k}}{\partial q_l} = [0]$ si $l > k$ alors on peut écrire :

$$E_c = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^k \sum_{l=1}^k \text{Trace} \left[\left[\frac{\partial T_{0k}}{\partial q_j} \right] [J_k] \left[\frac{\partial T_{0k}}{\partial q_l} \right] \right] \dot{q}_j \cdot \dot{q}_l$$

$$E_c = \frac{1}{2} \sum_{j=1}^k \sum_{i=1}^n \sum_{l=1}^k \text{Trace} \left[\left[\frac{\partial T_{0k}}{\partial q_j} \right] [J_k] \left[\frac{\partial T_{0k}}{\partial q_l} \right] \right] \dot{q}_j \cdot \dot{q}_l$$

Pour cela :

$$\left[\frac{\partial T_{0k}}{\partial q_j} \right] [J_k] \left[\frac{\partial T_{0k}}{\partial q_l} \right] = \begin{cases} [0] & \text{Si } k < \max(l, j) \\ \left[\frac{\partial T_{0k}}{\partial q_j} \right] [J_k] \left[\frac{\partial T_{0k}}{\partial q_l} \right] & \text{Si } k \geq \max(l, j) \end{cases}$$

$$E_c = \frac{1}{2} \sum_{j=1}^n \sum_{l=1}^n \left[\sum_{k=\max(l, j)}^n \text{Trace} \left[\frac{\partial T_{0k}}{\partial q_j} \right] [J_k] \left[\frac{\partial T_{0k}}{\partial q_l} \right] \right] \dot{q}_j \dot{q}_l$$

$$\text{On pose } A_{ij} = \sum_{k=\max(l, j)}^n \text{Trace} \left[\frac{\partial T_{0k}}{\partial q_j} \right] [J_k] \left[\frac{\partial T_{0k}}{\partial q_l} \right]$$

$$E_c = \frac{1}{2} \sum \sum A_{jl} \dot{q}_j \dot{q}_l \quad \Rightarrow E_c = \frac{1}{2} [\dot{q}] [A] [\dot{q}]$$

Les éléments de la matrice d'inertie [A] sont donnés par la relation suivante :

$$A_{ij} = \sum_{k=\max(i, j)}^n \text{Trace} \left[\frac{\partial T_{0k}}{\partial q_i} \right] [J_k] \left[\frac{\partial T_{0k}}{\partial q_j} \right]$$

Appendice O :

Expression des coefficients dynamiques

Expression des éléments de la matrice des termes de Coriolis

$$B_{i,jl} = \frac{\partial A_{ij}}{\partial q_l} + \frac{\partial A_{il}}{\partial q_j} + \frac{\partial A_{jl}}{\partial q_i}$$

$$B_{i,jl} = 2 \sum_{k=\max(i,j,l)}^n \text{Trace} \left[\left[\frac{\partial^2 T_{0k}}{\partial q_j \partial q_l} \right] [J_k] \left[\frac{\partial T_{0k}}{\partial q_i} \right] \right]$$

$$B_{i,jl} = 2 \sum_{k=\max(i,j,l)}^n \text{Trace} \left[[\Omega_{0j} \cdot \Omega_{0l} T_{0k}] [J_k]_k [\Omega_{0j} T_{0k}]^l \right]$$

Expression des éléments de la matrice des forces centrifuges :

$$C_{ij} = \frac{\partial A_{ij}}{\partial q_i} - \frac{1}{2} \cdot \frac{\partial A_n}{\partial q_i}$$

$$C_{i,j} = 2 \sum_{k=\max(i,j)}^n \text{Trace} \left[\left[\frac{\partial^2 T_{0k}}{\partial q_j \partial q_l} \right] [J_k] \left[\frac{\partial T_{0k}}{\partial q_i} \right] \right]$$

$$C_{i,j} = 2 \sum_{k=\max(i,j,l)}^n \text{Trace} \left[[\Omega_{0j} \cdot \Omega_{0l} T_{0k}] [J_k]_k [\Omega_{0j} T_{0k}]^l \right]$$

Expression des éléments du vecteur force de gravité :

$$E_p = \sum_{i=1}^n (-m_i \cdot g \cdot T_i \cdot u_i) \quad \text{donc} \quad G_i = -\frac{\partial E_p}{\partial q_i}$$

$$G_i = \sum_{i=1}^n -m_i \cdot g \cdot \left(\frac{\partial T_{0i}}{\partial q_i} \right)$$

$$G_i = \sum_{i=1}^n -m_i \cdot g \cdot T_{0i} \cdot \Omega_{0i}$$

Appendice P [71] :

Propriété des coefficients dynamiques:

Termes de Coriolis :

$$B_i^{jk} = \frac{\partial A_{ij}}{\partial q_k} + \frac{\partial A_{ik}}{\partial q_j} + \frac{\partial A_{jk}}{\partial q_i}$$

Avec : $B_i^{jl} = - B_l^{ij}$ si $j \leq i$ et $j \leq l$

$$B_i^{jj} = - B_j^{ij} \quad \text{si } j \leq i$$

$$B_i^{jk} = 0 \quad \text{si } j > i$$

Termes centrifuges :

$$C_{ij} = \frac{\partial A_{ij}}{\partial q_k} - 1/2 \frac{\partial A_{jj}}{\partial q_i}$$

Avec : $C_{ij} = 1/2 B_j^{ij}$ si $j > i$

$$C_{ii} = 0$$

Appendice Q [71] :

Formalisme de NEWTON – EULER

Il est adapté à la construction du modèle dynamique inverse. Il permet le dimensionnement de la structure et des actionneurs. Le caractère itératif de ce formalisme réduit le temps de calcul par rapport au formalisme de LAGRANGE [12,31,88,89], il est basé sur une double récurrence : une récurrence avant de la base du robot vers l'effecteur en utilisant la formule de composition pour calculer les vitesses et accélérations donc, le torseur dynamique est une récurrence arrière de l'effecteur vers la base pour calculer les couples des actionneurs en exprimant le bilan des efforts pour chaque corps. La composition des vitesses donne :

$$\dot{\omega}_j = \dot{\omega}_{j-1} + \bar{\sigma}_j \dot{q}_{j\text{aj}} \quad (\text{Q} . 1)$$

$$\dot{V}_j = \dot{V}_{j-1} \Lambda L_j + \sigma_j \dot{q}_{j\text{aj}} \quad (\text{Q} . 2)$$

On dérive ces 2 expressions par rapport au temps pour obtenir la composition des accélérations :

$$\ddot{\omega}_j = \ddot{\omega}_{j-1} + \bar{\sigma}_j (\ddot{q}_{j\text{aj}} + \dot{\omega}_{j-1} \Lambda \dot{q}_{j\text{aj}}) \quad (\text{Q} . 3)$$

$$\ddot{V}_j = \ddot{V}_{j-1} + \ddot{\omega}_{j-1} \Lambda L_j + \dot{\omega}_{j-1} \Lambda (\dot{\omega}_{j-1} \Lambda L_j + \sigma_j \dot{q}_{j\text{aj}}) + \sigma_j (\ddot{q}_{j\text{aj}} + \dot{\omega}_{j-1} \Lambda \dot{q}_{j\text{aj}}) \quad (\text{Q} . 4)$$

On arrange l'expression (Q . 4) comme suit :

$$\ddot{V}_j = \ddot{V}_{j-1} + \ddot{\omega}_{j-1} \Lambda L_j + \dot{\omega}_{j-1} \Lambda (\dot{\omega}_{j-1} \Lambda L_j) + \sigma_j (\ddot{q}_{j\text{aj}} + 2 \dot{\omega}_{j-1} \Lambda \dot{q}_{j\text{aj}}) \quad (\text{Q} . 5)$$

On peut utiliser cette dernière expression pour déterminer la vitesse du centre de gravité, à savoir :

$$\dot{V}_{Gj} = \dot{V}_j + \omega_j \Lambda (\omega_j \Lambda S_j) \quad (\text{Q} . 6)$$

Ce qui permet d'obtenir le torseur dynamique :

$$F_j = m_j \dot{V}_{Gj} \quad (\text{Q} . 7)$$

$$N_j = C_j \dot{\omega}_j + \omega_j \Lambda (C_j \Lambda \omega_j) \quad (\text{Q} . 8)$$

F_j et N_j représente respectivement la somme totale des forces extérieures et la somme totale des moments extérieures. On initialise la récurrence par :

$$\dot{\omega}_0 = 0 \quad , \quad \omega_0 = 0 \quad , \quad \dot{V}_0 = 0$$

Appendice R : [41]

Rappel des notions de théorie des matrices la matrice est un tableau rectangulaire de nombres disposés par ligne et par colonnes. Le tableau ci-dessous

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

qui contient m lignes et n colonnes est une matrice d'ordre m x n. Les matrices sont généralement désignées par des lettres majuscules (A,B,.....), les éléments des matrices, par des lettres minuscules (a_{ik} , b_{ik},.....), chacun des éléments de la matrice est affecté d'un double indice. Le premier indice i désigne le numéro de la ligne. Si la matrice contient autant de lignes que de colonnes (m=n), on l'appelle la matrice carrée.

Par exemple la matrice de rotation (3x3) est une matrice carrée.

Il y a des cas particuliers où la matrice ne contient qu'une seule colonne ou une seule ligne, par exemple :

$$A = \begin{pmatrix} a_1 \\ a_2 \\ \dots \\ a_n \end{pmatrix} ; b = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{pmatrix}$$

La matrice colonne et la matrice ligne sont désignées par une lettre minuscule (a₁b₁,.....) les éléments de telles matrices unidimensionnelles sont désignés par la même lettre affectée d'un indice qui indique le numéro de l'élément.

Les opérations fondamentales effectuées sur la matrice sont les suivantes :

Transposition : la transposée A' de la matrice A s'obtient en substituant à chaque ligne de la matrice A une colonne de même numéro.

Par exemple étant donnée une matrice carrée (3 x 3) A .

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Sa transposé sera

$$A' = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Multiplication par un scalaire le produit de multiplication de la matrice A par le scalaire λ est la matrice

$$\lambda A = \begin{pmatrix} \lambda a_{11} & \lambda a_{12} & \lambda a_{13} \\ \lambda a_{21} & \lambda a_{22} & \lambda a_{23} \\ \lambda a_{31} & \lambda a_{32} & \lambda a_{33} \end{pmatrix}$$

Obtenu par multiplication de chaque élément de la matrice A par la quantité λ

Les produits λA et $A \lambda$ définissent une seule et même matrice $\lambda A = A \lambda$.

Addition : deux matrices A et B sont additionnables si elles ont un même nombre de ligne et même nombres de colonnes.

Par addition des matrices A et B on obtient une matrice C,

$$C = A + B$$

Dont les éléments

$$C_{ik} = a_{ik} + b_{ik} \text{ (pour tout } i,k \text{)}$$

Sont les sommes correspondants des matrices A et B

L'addition des matrices est commutative et associative.

$$A + B = B + A$$

$$A + (B + C) = (A + B) + C = A + B + C$$

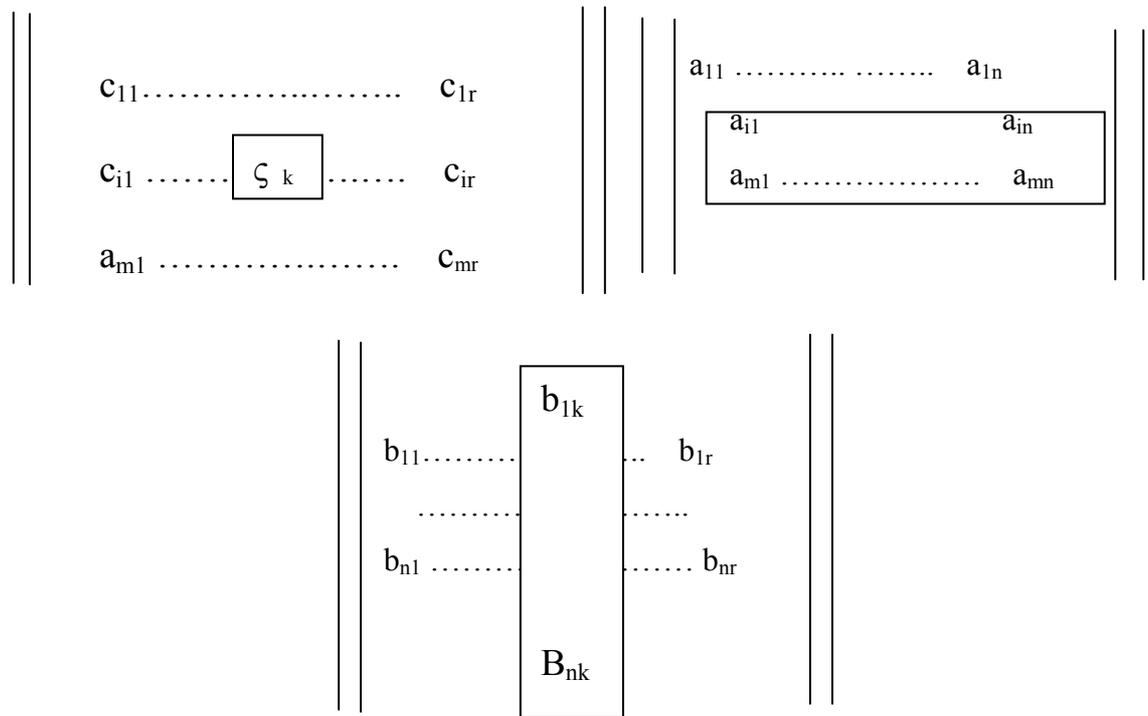
Il en est de même pour la soustraction de deux matrices, car $A - B = A + (-1) B$.

En cinématique des mécanismes, les opérations d'addition des matrices et de leur multiplication par un scalaire s'effectuent sur des matrices colonnes.

Multiplication matricielle. On ne peut multiplier deux matrices A et B si la première a autant de colonnes que la deuxième a de lignes.

Si A est une matrice d'ordre (m x n) et b est une matrice d'ordre (n x r).

Illustrons la règle de multiplication de ces matrices par un schéma :



On considère les lignes de la matrice A et les colonnes de la matrice B comme des vecteurs dimensionnels qui ont pour projections les éléments des lignes et des colonnes correspondantes. C'est ainsi que dans la matrice A les éléments de l' i -^{ème} ligne $a_{i1}, a_{i2}, \dots, a_{in}$ sont considérés comme les projections du vecteur a_i de même, la k -^{ème} colonne de la matrice B est considérée comme un vecteur b_k qui a comme projection $b_{1k}, b_{2k}, \dots, b_{nk}$ (remarquons que k est le second indice) conformément à la règle de multiplication matricielle, l'élément c_{ik} de la matrice C se trouve à l'intersection de l' i -^{ème} ligne et de la k -^{ème} colonne est produit scalaire $c_{ik} = a_i \cdot b_k$ de l' i -^{ème} ligne de la matrice A par la k -^{ème} colonne de la matrice B,

$$c_{ik} = a_{i1} b_{1k} + a_{i2} b_{2k} + \dots + a_{in} b_{nk} = \sum_{s=1}^n a_{is} b_{sk}$$

(i = 1, 2, ..., m ; k = 1, 2, ..., r) (a)

Au cas où m=n=r=3 c'est -à-dire lorsque A, B et C sont des matrices carrées (3 x 3)

(tel est le cas des matrices de rotation), la formule (a) devient

$$c_{ik} = \sum_{s=1}^3 a_{is} b_{sk} \quad (i, k = 1, 2, 3)$$

Citons un exemple de multiplication de deux matrices :

$$\begin{aligned}
 c = & \begin{vmatrix} 3 & 0 \\ 1 & 1 \\ 5 & 2 \end{vmatrix} = \begin{vmatrix} 4 & 7 \\ 6 & 8 \end{vmatrix} \begin{vmatrix} 3.4+0.6 & 3.7+0.8 \\ 1.4+1.6 & 1.7+1.8 \\ 5.4+2.6 & 5.7+2.8 \end{vmatrix} \\
 = & \begin{vmatrix} 12 & 21 \\ 10 & 15 \\ 32 & 51 \end{vmatrix}
 \end{aligned}$$

Considérons encore un cas particulier: C=Ab

On multiplie ici la matrice (m x n) A par la matrice colonne (n x 1) b ; le produit en est la matrice colonne (m x 1) c.

En vertu de la règle de multiplication des matrices

$$\begin{array}{|c|} \hline c_1 \\ \hline \boxed{c_i} \\ \hline c_m \\ \hline \end{array} = \begin{array}{|ccc|} \hline a_{11} & \dots & a_{1n} \\ \hline \boxed{a_{i1}} & & \boxed{a_{in}} \\ \hline a_{m1} & \dots & a_{mn} \\ \hline \end{array} \times \begin{array}{|c|} \hline b_1 \\ \hline \\ \hline b_m \\ \hline \end{array}$$

$$c_i = a_{i1} b_1 + a_{i2} b_2 + \dots + a_{in} b_n = \sum_{s=1}^n a_{is} b_s \quad (i=1,2,\dots,m)$$

pour $m=n=3$ cette formule traduit l'opération de multiplication de la matrice (3x3) A par une colonne b à trois éléments. Ce cas a lieu pendant la transformation matricielle des projections d'un vecteur.

De façon générale, la multiplication matricielle est non commutative.

$$AB \neq BA$$

Mais par contre, elle est associative et distributive :

$$(AB)C = A(BC) = ABC \quad (\text{associativité})$$

$$A(B+C) = AB+AC \quad (\text{Distributivité})$$

Techniques stochastiques d'optimisation

Résoudre un problème d'optimisation, c'est trouver parmi un ensemble Ω de solutions possible S, la solution S^* minimisant (ou maximisant) une fonction objectif $F(S)$ donnée. Les techniques déterministes sont efficaces à condition que $F(S)$ soit continue et comporte très peu d'extremums (minimums locaux) sinon les développements mathématiques deviennent très lourds et difficiles à établir. De plus, elles se basent sur la construction d'une direction de recherche privilégiée dans Ω donc, le risque de manquer l'optimum global S^* est d'autant plus grand quand $F(S)$ comporte plusieurs optimums locaux. Dans ce cas, les techniques stochastiques qui se basent sur une recherche aléatoire de S^* uniformément distribuée dans l'espace Ω sont préférées. Seules des informations sur les valeurs de $F(S)$ sont nécessaires. Les techniques stochastiques les plus répandues sont :

La technique de Monté-Carlo.

Méthode de Hill-Climbing

- Méthode de Monte-Carlo.avec réduction d'intervalle.

Le recuit simulé.

Les algorithmes génétiques.

La recherche taboue.

Technique de Monte-Carlo :

Elle consiste à générer aléatoirement plusieurs solutions S , de les comparer en retenant à chaque fois la meilleure solution [51]. L'algorithme du cycle de base présente comme suit :

Algorithme R.1 Algorithme de la technique de Monte-Carlo

Générer aléatoirement une solution initiale S_0 dans Ω

Prendre $S^* = S_0$

Générer aléatoirement une solution S dans Ω

Evaluer $F(S)$

Si $F(S) < F(S_0)$ alors $S^* = S$

Fin

Répéter

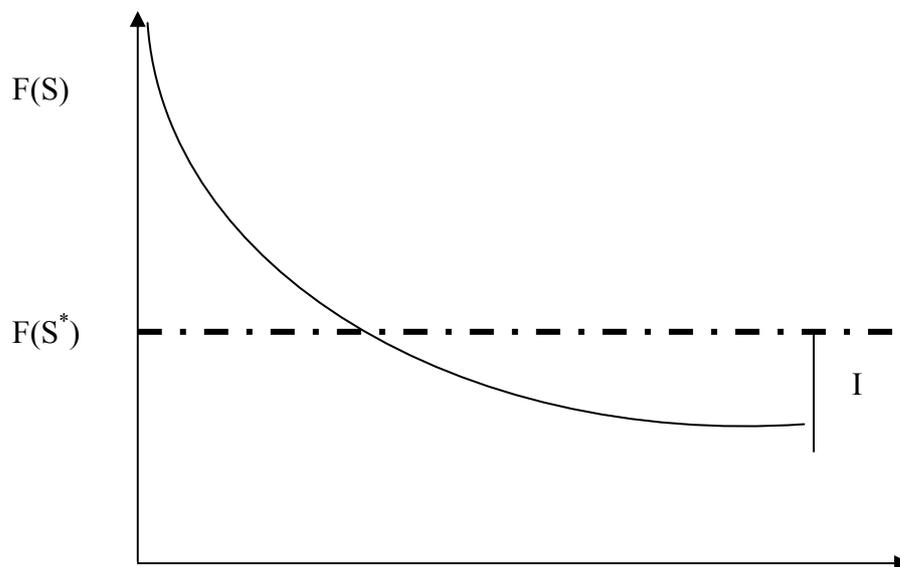


Figure R.1 : Convergence d'une technique de Monte-Carlo

L'application de cette méthode semble aisée. Il suffit d'évaluer la fonction F à chaque tirage aléatoire de S . le temps de calcul est infini. Pour des raisons pratiques, le

processus de base est arrêté après un nombre d'itérations 1 suffisant pour que la solution recherchée soit acceptable. Pour réduire l'effort de calcul et accélérer la convergence, on introduit la notion de voisinage $V(S_i)$ défini par son étendue L_i de la meilleure solution S_i obtenue à l'étape i . cette idée est exploitée dans les techniques de Hill-Climbing, Monte-Carlo, recuit simulé et recherche Tabou.

Méthode de Hill-Climbing :

Elle propose d'effectuer une recherche dans un voisinage $V(S)$ d'étendue fixe L centrée autour de la meilleure solution retenue [52]

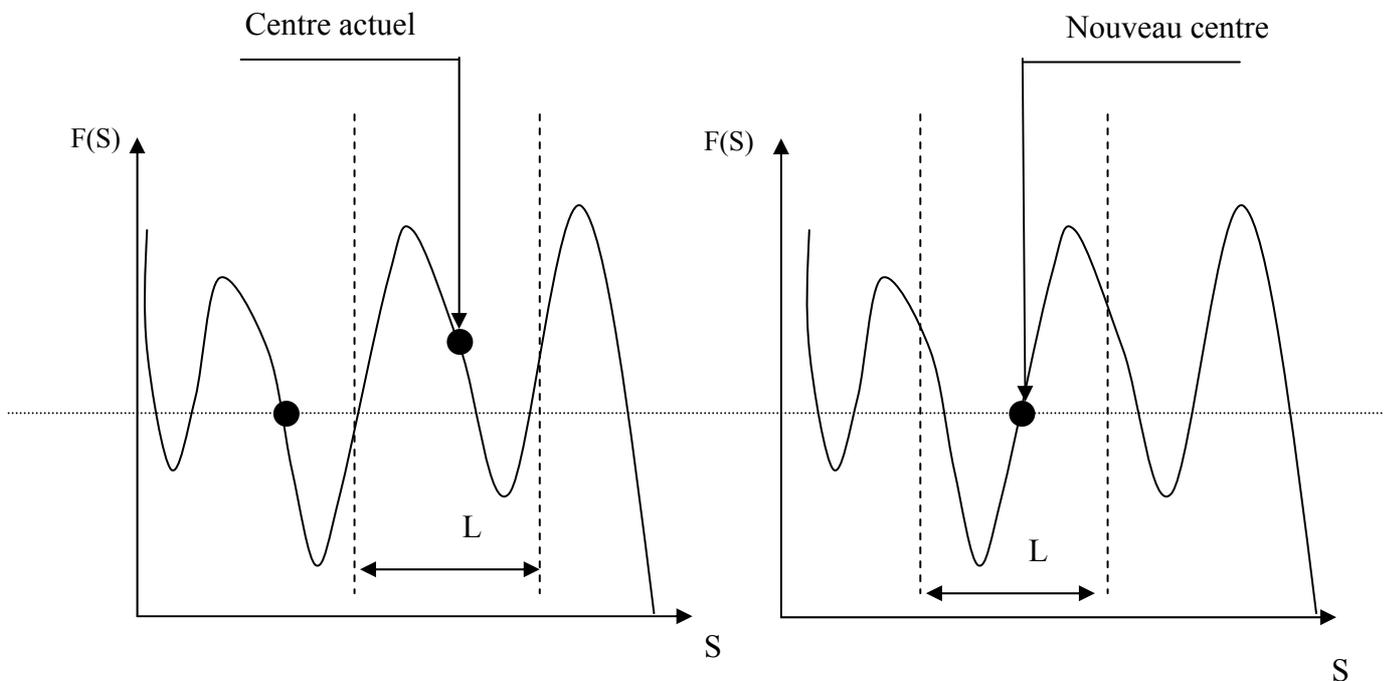


Figure R.2 : Meilleure solution trouvée.

Changement du centre dans la technique de Hill-Climbing

La seule difficulté réside dans le choix de la taille de L . un mauvais choix induit à :

Piégeage de la technique dans un minimum local (L trop petit).

Augmentation du temps de calcul (L trop grands)

L'algorithme correspondant est comme suit :

Algorithme R.2 Algorithme de la technique de Hill-Climbing

Générer aléatoirement une solution initiale S_0

$S^* = S_0$

Centre de voisinage = S^*

Générer aléatoirement S dans le voisinage $V(S^*)$

Evaluer $F(S)$

Si $F(S) < F(S^*)$ alors $S=S^*$

Centre du voisinage S^*

Jusqu'à convergence

Méthode de Monté-Carlo avec réduction d'intervalle :

C'est une version améliorée de la méthode précédente. Elle introduit un facteur de réduction de l'intervalle L . lors du tirage initial, L est choisi large. Au fur et à mesure que le nombre d'itérations i augmente, la taille de L est réduite afin d'affiner la solution. Soit E le nombre d'échecs successifs comptabilisés depuis la dernière amélioration (on considère un échec quand le choix de S fait augmenter $F(S)$). Quand E prend des valeurs significatives et pour rentabiliser le calcul, on réduit l'intervalle L . pour cela, on se fixe un seuil E_{max} pour une prise de décision.

L'algorithme correspondant sera alors :

Algorithme R.3 Algorithme de la technique de Monte-Carlo réduction d'intervalle

Générer initiale S_0 aléatoirement

$S^* = S_0$

Générer une solution S dans $V(S^*)$, évaluer $F(S^*)$

Si $F(S) < F(S^*)$, alors $S^*=S$

Sinon compter E

Si $E = E_{max}$

Réduire $V(S)$

Jusqu'à convergence

L'efficacité de cette méthode dépend du choix du facteur de réduction de L et de E_{max}

Recuit simulé :

Le fait, à tout prix, de diminuer la fonction objectif $F(S)$ peut conduire à une situation de piégeage de la recherche autour d'un optimum local. L'idée de cette méthode est de

faire échapper $F(S)$ de cette situation (permettre l'augmentation de $F(S)$ pour pouvoir se déplacer vers l'optimum global) [53, 54] .

En métallurgie, le recuit est un traitement thermique qui consiste à chauffer un métal à un niveau tel qu'il permet l'équilibre physico-chimique et structurel d'un matériau et de le refroidir par palier afin que les atomes s'organisent de manière à le faire passer (le matériau) d'une configuration de haute énergie à celle d'énergie minimale. Ce passage obéit à la loi de Boltzman :

$$P = \exp \dots\dots\dots [dE/KT]$$

Ou :

K : constante de Boltzman

dE : différence d'énergie entre les niveaux initial et final.

F : température.

P : probabilité de passage d'un groupe d'atomes d'un niveau d'énergie E_1 à un niveau d'énergie supérieure E_2

Par analogie, les configurations stables correspondent aux solutions qui font diminuer la fonction objectif, les configurations instables font augmenter la fonction objectif. La loi de Boltzman adaptée s'écrit alors :

$$P = \exp[\Delta F/T]$$

e) Recherche Tabou :

Cette technique consiste à se déplacer d'une solution à une autre en s'interdisant de revenir à une solution déjà rencontrée. On définit un voisinage $V(S)$ pour chaque solution S en supposant qu'on dispose d'une liste Tab de toutes les solutions rencontrées depuis le début de l'exécution de la méthode. A partir de la solution courante S_i on passe à une solution $S_{i+1} \dots V(S)$ minimisant la fonction objectif et en ajoutant S_i à la liste Tab. En pratique, ce processus utilise trop de place mémoire ainsi que le temps de calcul (on doit comparer chaque solution choisie avec tous les éléments de la liste Tab). Pour éviter cet inconvénient, la méthode Tabou préconise de conserver en mémoire la transformation élémentaire qui permet de passer de la solution courante à la suivante en s'interdisant d'appliquer son inverse (la solution courante n'est pas

ajoutée à la liste Tab). La recherche tabou se prête bien au problème d'optimisation discret (optimisation dans les réseaux) [13, 14, 15, 55]

Algorithme R.4 Algorithme de la recherche Tabou

L'algorithme correspondant sera :

Générer aléatoirement une solution initiale S_0

$S^* = S_0$

Tab = \emptyset

Répéter

Générer une solution aléatoire S dans $V(S^*)$

Si $S \dots \text{Tab}$ alors évaluer $F(S)$

Si $F(S) < F(S^*)$ alors $S^* = S$

Centre du voisinage S^*

Sinon ajouter S à la liste Tab

Jusqu'à convergence

Algorithme génétique :

Ils sont basés sur des mécanismes de sélection naturelle et sur la génétique. A partir d'une population de solutions potentielles (chromosomes) initiale arbitrairement choisie, on évalue leurs performances en utilisant des opérateurs simples : la sélection, le croisement et la mutation et on recommence le cycle [16, 17, 56]. Ces algorithmes diffèrent des méthodes précédentes par les deux points suivants :

On utilise un codage des paramètres (solutions).

On travaille sur une population de points au lieu d'un point unique.

Algorithme R.5 Algorithme génétique

L'algorithme sera le suivant :

Générer une population initiale.

Répéter

Evaluer les performances de chaque individu

Les sélectionner et les regroupe par paires selon leurs

Performances

Générer une nouvelle population (appliquer les opérations de croisements et de mutations)

Jusqu'à convergence

1

2
Oui

3

4

Calculer S_1, S_2, S_3

Calculer $T^*_Q = \min F_{obj}(S_1, S_2, S_3)$

$F_{obj}(T^*_Q, S_1, S_2, S_3) \geq F_{obj}(\text{meilleure})$
Oui

Non

Calculer T_1, T_2

Calculer T_a

Calculer T_v

Calculer T_l, T_r

Calculer $T^* = \text{intersection}(T_1, T_2, T_a, T_v, T_l, T_r)$

Non

Oui

$F_{obj}(T^*) \geq F_{obj}(\text{meilleure})$
Oui

Echec = Echec + 1

Non

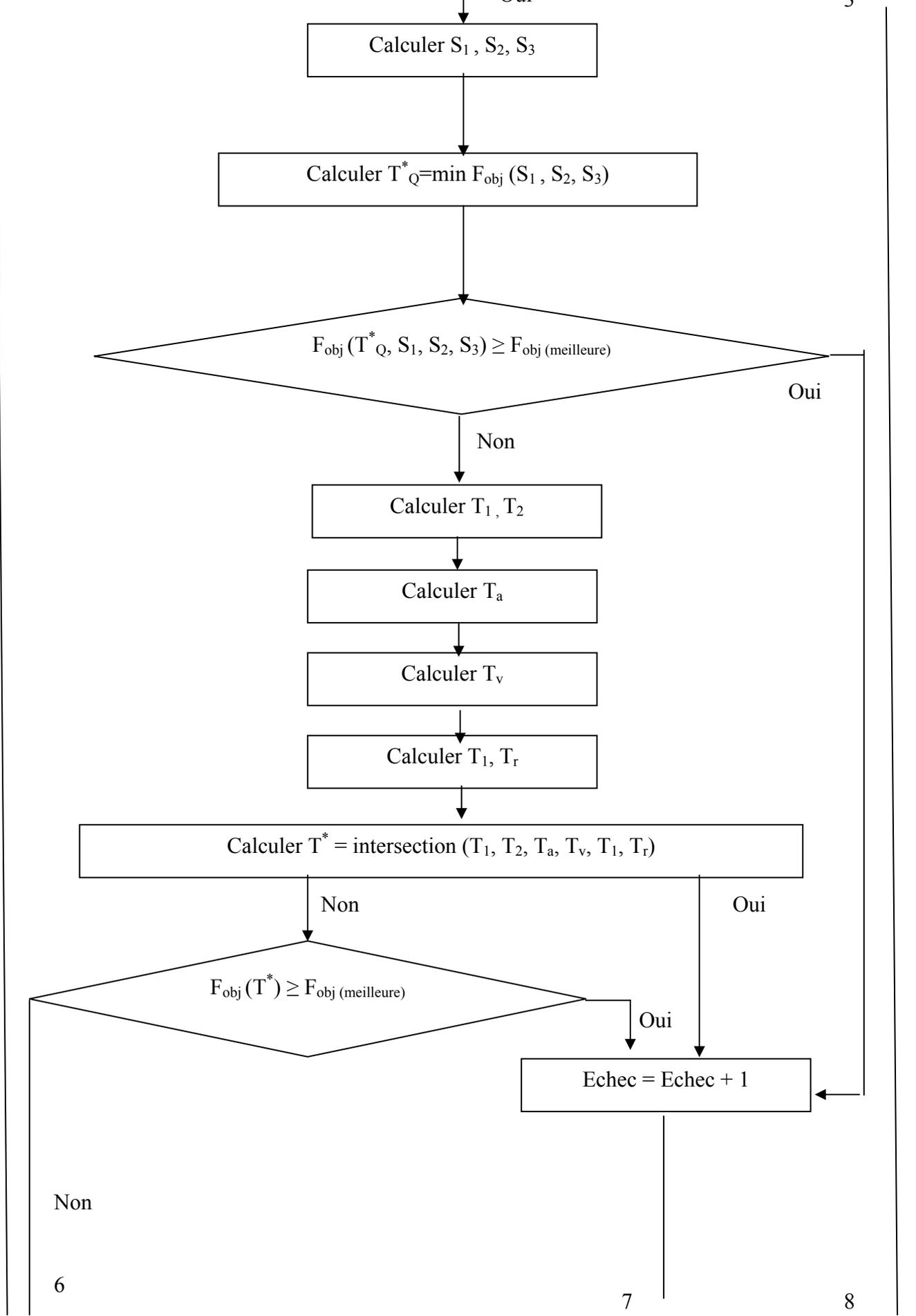
5

6

7

8

9



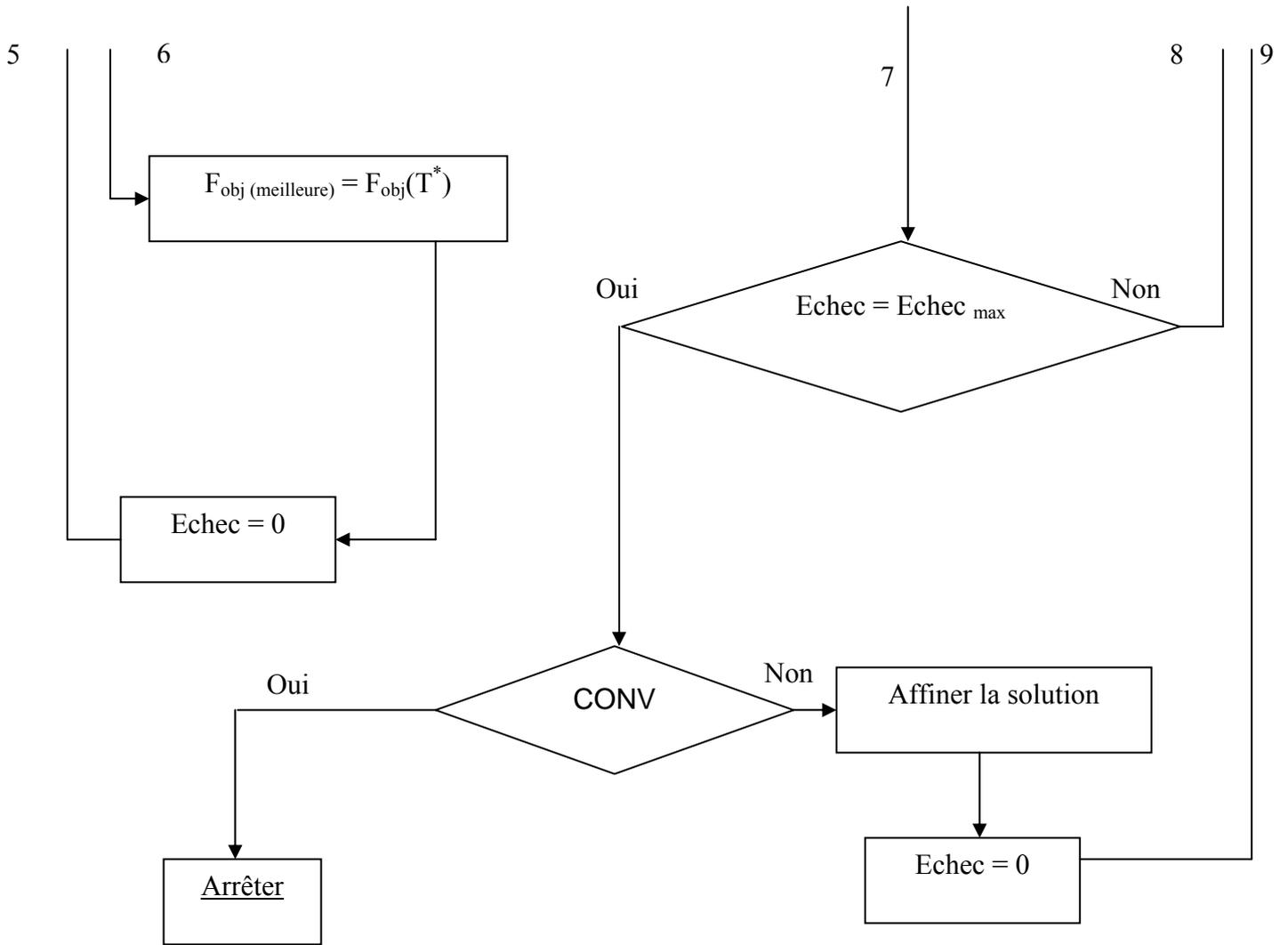


Tableau R.2 : Liste des systèmes de CAO ayant des applications dédiées à la robotique [64] .

Fournisseur	Produit	Caractéristiques	Matériel
COMPUTERVISION Tout Gallieni 2 36 av. Gallieni 175 BAGIVOLET Tel 43 60 01 51	ROBOGRAPHIX	Modules spécifiques ROBOT KINEMATIC MODEL et AOUTPUT PROCESSOR représentation Filaire.	Stations CDS 400 Et Designer v-x
DASSAULTSYSTEMES 40 bd. H.Sellies 92150 SURENES Tel 47 28 00 44	CATIA ROBOTICS	Jusqu'à 20 ddl efforts statiques description minimal des tâches. visualisation de la trace.	IBM 30 XX .43XX
GENERAL ELECTRIC /CALM 31 Bd. Bouvets 92000 NANTERRE Tel 47 76 44 31	ROBT-SIM	Représentation filaire. Langage CRL. Modèles dynamiques.	Famille Appollo Et vax
INTEGRAPH 101 Rue des Slets SILIC 578 94653 Rungis cedex Tel 46 87 15 62	INTEGRAPH ROBOT PROGRAMMING	Bibliothèque de trois robots. Pas de chronométrage sur l'image	Vax et micro-vax integraph
IRISA Compus de Beaulieu 35042 RENNES CEDEX Tel 99 36 20 00	SIEL (disponible sous certaines conditions)	Simulation de capteurs locaux et des lois de commande associées.	Vax et Tektronix

Fournisseur	Produit	Caractéristiques	Matériel
<p>Mc DONNEL DOUGLAS 106 Breaux de la Colline 22213 St Cloud Tel 46 02 31 01</p>	<p>ROBOTICS : BUILD PLACE COMMAND ADJUST</p>	<p>Représentations filaire. Boucles Complexes. Parallélisme. Coopérations de robots. ADJUST pour calibrations.</p>	<p>Gamme Dec station de travail Evans et Sutherland</p>
<p>SILMA 2111 Grand Road Los ALTOS CA 94029, USA Tel 415 967 5878</p>	<p>CIMSTATION (ex ROBOCAM)</p>	<p>Représentation filaire. Plusieurs lois de comande parallélisme langage SIL. Aide à l'implantation de sites .</p>	<p>Famille Appollo</p>
<p>TECNOMATIX Herntalesebeai 55 B.2100 ANTWEPEN (DEURNE), Belgique Tel .3 332 38 90</p>	<p>ROBCAD</p>	<p>Recherche automatique de trajectoires optimales en temps langage TDL . Collision parallélisme.</p>	<p>Stations ROBCAD 50 Et ROBCAD 100</p>

Tableau R.3 : Avantages comparés des programmations par CAO et par langages textuels [64].

	CAO	LANGAGES
Raisonnement dans un univers 3D (trajectoire sans collision, apprentissage de pointes étude des contacts...)	Naturel	Difficile
Utilisation des structures logiques	Peu naturelle	Naturelle
Intégration des informations sensorielles	Nécessite des modèles de capteurs	Possible
Evaluation du temps de cycle	Fonctions Intégrées	Difficile
Vérification des programmes, apprentissage.	Hors –ligne (disponibilité totale du matériel)	Impossible
Optimisation des taches.	Selon différents Critères	Sur le site (immobilisation du matériel)
Indépendances robot / programme	Total	Partielle
Mise en œuvre	Transparence par rapport au robot: ergonomie élevée.	Apprendre autant de langage que de robots.
Ecart entre trajectoire programmé et trajectoire réelle	Erreur de modélisation + même sources d'erreurs que les langages	Due au changeur de coordonnées et à la précision de répétition
Coût	Elevé	Raisonnable

Développement d'un système de CAO sur micro-ordinateur:

Le système est organisé autour d'une base de donnée géométrique tridimensionnelle de type réseau. Un logiciel de modélisation permet de définir des objets par assemblage de primitives élémentaires. Une interface de dialogue opérateur assure une manipulation simple des modèles, essentiellement grâce à une tablette à digitaliser. Enfin, un logiciel graphique soumet les informations vectorielles 3D et les informations de type texte à un contrôleur spécialisé.

Base e données : afin de répondre au exigences ce temps de réponse et de place mémoire d'un micro- ordinateur. La base a été spécifiée autour des concepts suivants : Les entités sont stockées sur deux supports différents ; les modèles actifs, c'est-à-dire sollicités par l'application, restant en mémoire vive; les entités activables sont stockées en mémoire de masse et sont accessibles grâce à un répertoire. On minimise ainsi le nombre d'accès disque.

Les modèles activés plusieurs fois dans une même scène ne sont pas dupliqués, ceci pour éviter toute redondance de l'information, la mémoire est gérée dynamiquement.

Les trois types d'entités fondamentales représentées en base de données sont les objets, les robots

(Chaînes cinématiques simples) et les outils. A ces entités de base, collectionnées par type dans des bibliothèques, il convient d'ajouter scène, qui rassemblent plusieurs entités ainsi qu'un certain nombre d'élément géométriques de conception (point, droite, plan, repère...) logiciel de modélisation tridimensionnelle :

Les primitives polyédriques sont de type simple (prismatique ou de révolution), ou de type complexe (bielle, chape, équerre). A chaque primitive, est associée une description volumique paramétrée de type quadrique (suspensoïde) , qui constitue une enveloppe convexe englobante de al description. A partir de cette description, les calculs d'interférence entre solides peuvent être rapides.

La modélisation des robots se fait au moyen d'un éditeur graphique interactif permettant de définir, de modifier ou de supprimer, des articulations de type primastique ou rotoïde non couplées. Pour chaque liaison, on peut spécifier les limites articulaires, ainsi que les performances en vitesse et en accélération.

Interface de dialogue :

Il met à la disposition de l'opérateur plusieurs fonctionnalités :

La trace d'une évolution d'une session de modélisation dans l'arbre hiérarchique des menus est toujours présente à l'opérateur;

Un ensemble de fonction d'analyse permet d'accéder à la description des entités à tous les niveaux de conception,

La gestion de la tablette à digitaliser permet de décrire et de manipuler les modèles en limitant au strict minimum les saisies au clavier,

Les fonctions graphiques classiques (point observé, distance et direction d'observation, changement d'échelle, zoom) peuvent être utilisées mode incrémental, procurant ainsi une pression des déplacement continu. Ces fonctions sont accessibles en permanence.

Logiciel graphique : il réalise le prétraitement des données géométriques destinées au contrôleur graphique 3D qui assure les transformations, les projections 2D/3D et le clipping. Le logiciel effectue aussi une élimination des lignes cachées, partielle en ce sens que seules les lignes non visibles de chaque solide pris indépendamment des autres solides de la scène ne sont pas visualisées. On arrive ainsi à un compromis réalisme de la scène/temps de traitement satisfaisant.

L'édition d'un fichier graphique intermédiaire 3D permet en outre d'envisager un couplage avec les systèmes de DAO en vue de constituer des dossiers techniques (cotation, sortie graphiques...).

Nous avons développé autour de ce système une application interactive pour la programmation de robots de type SCARA, à partir d'une tablette à digitaliser. Celle-ci offre un moyen simple et efficace de définition de trajectoires 2D, l'altitude et l'orientation de l'outil sont gérées dans des zones spécifiques de la tablette. La trajectoire cartésienne peut être programmée soit en pointant, sur un schéma de la cellule posé sur la tablette, les points de passage désirés, soit en faisant référence à des entités géométriques de la scène représentant la cellule en base de données le choix d'une configuration de départ (coude à droite ou coude à gauche) est spécifié par l'utilisateur [64,101,102,130].

Programmation et contrôle d'exécution d'une cellule flexible d'assemblage :

Introduction :

Une cellule flexible d'assemblage (C.F.A) est le lieu, dans un atelier flexible, où les tâches d'assemblage et les opérations associées sont effectivement réalisées. Elle comprend un ensemble d'éléments fortement liés tels que manipulateurs, capteurs et dispositifs de péri-robotique. Ces éléments coopèrent étroitement pour réaliser une tâche (ou un ensemble de tâches) d'assemblage.

A l'inverse de la problématique des robots dits " d'intervention" (robots mobiles autonomes par exemple) dans laquelle le système évolue dans un environnement peut ou pas structuré (et changeant), et acquiert par lui-même l'essentiel des informations nécessaires, la problématique de la CFA est celle de la robotique dite " à poste fixe ". Elle s'intéresse à l'exécution, par un système robotisé, d'une tâche pré-établie dans un environnement bien structuré et connu à priori, les aspects essentiels sont, dans ce cas, la programmation, ou plus généralement la " préparation des travaux " ainsi que l'ensemble des moyens permettant une exécution efficace et "robuste" de la tâche, d'où la mise en avant des notions de flexibilité et d'autonomie.

La flexibilité : une CFA doit être conçue de manière à permettre la réalisation non pas d'une seule tâche mais d'un ensemble de tâches appartenant à une même application. Il doit être possible de la programmer (et de la reprogrammer) facilement, de changer ou d'ajouter un composant, d'en modifier la configuration spatiale, de construire de nouveaux programmes traitant de nouvelles variantes de pièces. Bien qu'un certain niveau de flexibilité existe désormais au niveau des composants (manipulateurs généraux programmables dans l'espace cartésien, système de vision programmables ...), ceci ne suffit pas pour obtenir une flexibilité raisonnable au niveau de l'ensemble.

Des méthodes et outils doivent être développés de manière à aider l'utilisateur à réaliser des tâches d'assemblage. En effet, une telle programmation nécessite l'intégration et l'utilisation d'un grand nombre d'informations et de connaissances provenant de sources très diverses et il n'est pas simple de les combiner dans le cadre d'une approche structurée bien définie.

L'autonomie : à l'exécution, une CFA doit se comporter comme un système autonome qui délivre des pièces assemblées selon un processus préalablement défini. Elle doit être capable de gérer ses propres ressources, d'adapter son fonctionnement conformément aux informations les plus récentes, de traiter les événements qui interviennent au niveau de la cellule de manière " Transparente". On ne fera appel au système de gestion de plus

haut niveau (atelier, opérateur) que si les tâches qui sont allouées à la cellule ne peuvent être réalisées conformément aux spécifications [64,104].

le système NNS :

Le système NNS comporte un environnement de programmation (environnement hors-ligne) et un système de conduite (système en ligne) (figure R-3)

L'environnement de programmation inclut essentiellement des Modules Décisionnels Spécialisés à aider le programmeur à spécifier et à raffiner progressivement une tâche d'assemblage pour une cellule donnée.

A partir de cette description, le compilateur de tâche d'assemblage produit un ensemble structuré d'informations appelé modèle d'exécution.

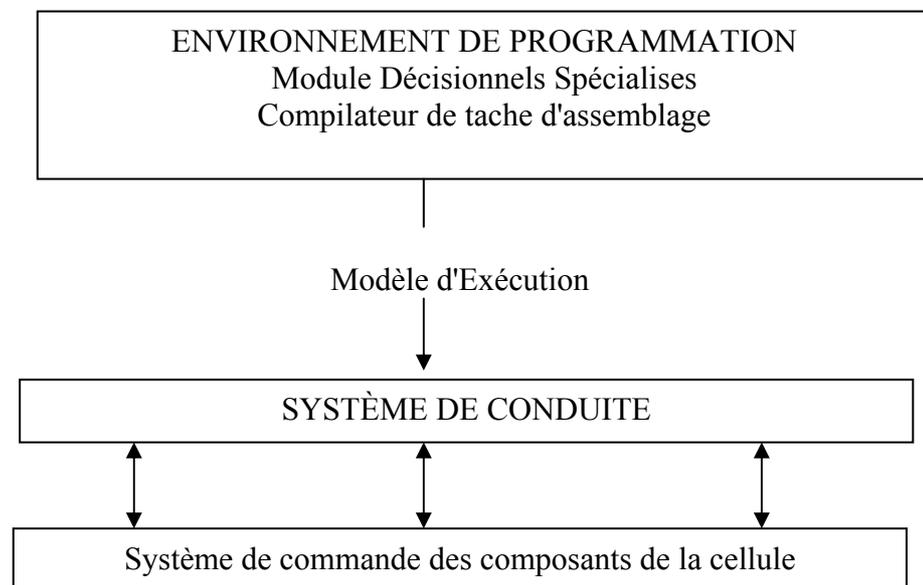


Figure R-3 : Architecture du système

Le système de conduite est un système basé sur la connaissance (Knowledge – Based system); il assure l'exécution de la tâche. Il utilise le modèle d'exécution pour générer, au fur et à mesure, les consignes nécessaires à la réalisation de la tâche, à partir des informations provenant de l'environnement (arrivée d'une pièce, fin d'une action, détection d'une anomalie...)

Cette architecture est fondée sur les idées suivantes.

Des capacités décisionnelles doivent être mises en œuvre à la fois hors – ligne et en-ligne afin de répondre aux objectifs de flexibilité et d'autonomie visés.

Toutefois, il est important de pouvoir distinguer clairement les informations provenant de la phase de conception de la cellule et de définition de la tâche, les problèmes qui

peuvent être résolus à la programmation ainsi que les décisions qui doivent être prises au cours de l'exécution. Il est également important de noter que la plupart des problèmes nécessitant des processus décisionnels " lourds" peuvent être traités hors- ligne.

Les aspects modélisation et représentation de la connaissance sont ici de première importance. En effet, tant à la programmation qu'à l'exécution, le système doit être capable de raisonner sur la tâche, sur la cellule et sur ses interactions avec (environnement à différents niveaux d'abstraction.

La modélisation :

L'ensemble du système est basé sur une modélisation de la tâche à différents niveaux d'abstraction:

Le niveau PROCESSUS D'ASSEMBLAGE spécifie les différentes opérations qui doivent être réalisées sur les pièces: alimentation, montage, inspection, etc..

Le niveau TACHE conserve l'évolution des pièces dans la cellule, il définit les différentes actions qui permettent à la cellule considérée de réaliser un processus d'assemblage donné.

Le niveau ACTION explicite la manière de faire évoluer les pièces dans la cellule : grands mouvements d'approche, prise d'objets.

Stratégies d'inscription... ce niveau établit notamment une modélisation fonctionnelle des éléments de la cellule à partir des fonctions qui leur sont allouées dans le cadre des tâches d'assemblage (effecteurs, support, capteur, machine spécialisée, opérateur atelier) ainsi qu'une structuration de l'espace de la cellule (en vue de sa gestion pour l'exécution des différentes opérations).

Le niveau FONCTION " encapsule" les primitives fournies par les systèmes de commande des différents composants de la cellule dans des fonctions génériques manipulées par les niveaux supérieurs.

Un processus d'assemblage consiste en un ensemble d'opérations partiellement ordonnées. Les composants primaires sont introduits dans la cellule et les produits fins repartent sur des convoyeurs.

Une pièce représente l'état d'avancement (pièce primaire, sous – assemblage ...).

Chaque pièce est caractérisée par son identité.

Exemple :

L'identité " pièce A " indique que la pièce correspondante est du type " pièce A " ;

l'identité "(pièce A₁ pièce A₂pièce A_n) " indique que la pièce correspondante est du type " pièce A", ou " pièce A₂" ... c'est le cas typique de pièces

fournisseurs à la cellule dans un ordre quelque ou après une opération d'indentification qui ne discrimine pas complétement l'identité d'une pièce;

L'identité "INCONNU" ou "!" est associée à toute pièce qui ne correspond à aucune étape du processus d'assemblage.

Les opérations d'un processus d'assemblage font apparaître, disparaître ou modifient l'identité des pièces dans la cellule. Nous définissons les opérations

Opération d'assemblage

Pièce X & pièce Y \longrightarrow pièce X Y

Opération de démontage

Pièce X \longrightarrow pièce X₁ & pièce X₂

Opération technologique

Pièce X \longrightarrow pièce X*

Opération d'inscription :

pièce X \longrightarrow pièce X+ / pièce X-

Opération d'identification

? (pièce X₁.....pièce X_n) pièce X₁ |.....| pièces X_n

Opération de déchargement

pièce x ... \longrightarrow opération d'alimentation

..... \longrightarrow Pièce X

Un processus d'assemblage est spécifié par la donnée de l'ensemble des pièces concernées ainsi que les différents opérations à réaliser sur ces pièces.

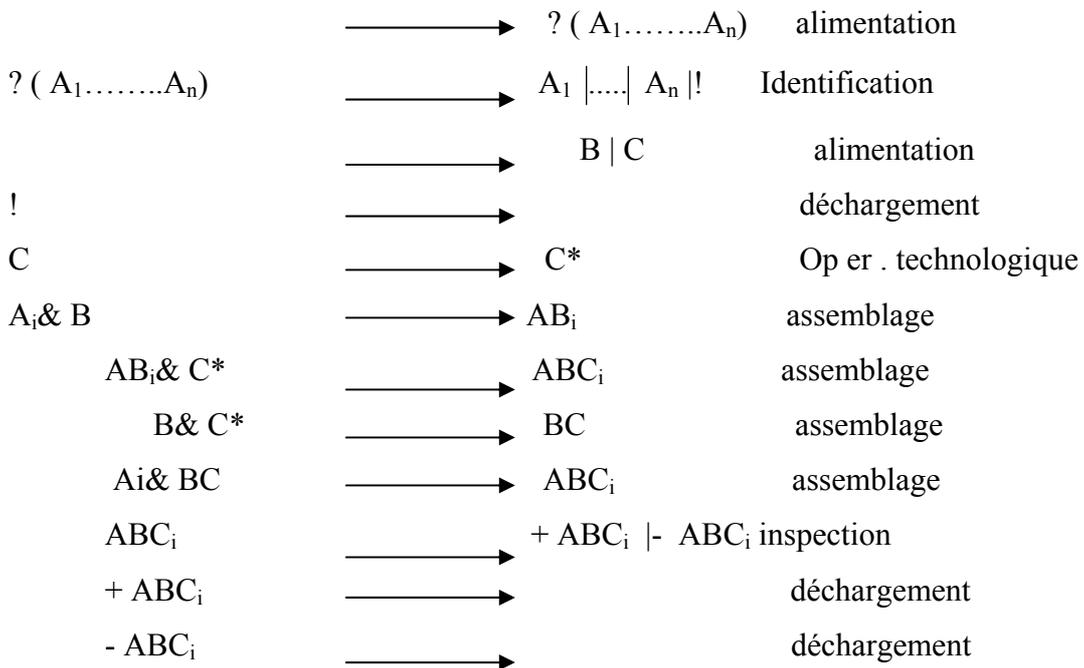
Le processus peut comporter plusieurs ordonnancements possibles des opérations.

Exemple :

" Trois pièces A,B et C sont assemblées puis inspectées avant d'être déchargées. Il existe N variantes de la pièce A : (A₁.....A_n); elles sont alimentées sur un convoyeur et doivent être identifiées. Les pièces B et C sont fournies par un autre convoyeur. La pièce B peut être présentée dans deux positions d'équilibre différentes (à l'endroit ou l'univers). C doit subir une opération de vissage avant d'être assemblée. Les pièces ABC. Peuvent être obtenus soit par l'assemblage de A; et de B puis de C, soit par l'assemblage de B et de C puis A_i"

L'ensemble du processus peut être représenté par la liste des opérations suivantes :

Algorithme [Processus] R-6 :



L'état d'un processus d'assemblage peut être représenté, à chaque instant par l'identité de toutes les pièces présentes dans la cellule :

< état. Processus > :: (<pièce> *)

Cette représentation est indépendante de toute cellule particulière remarquons qu'elle peut être utilisée en- ligne au niveau de la gestion de l'atelier pour connaître l'état d'avancement d'un processus dans une cellule ainsi que les capacités fonctionnelles d'une cellule après une panne ou un incident (opérations qui ne peuvent plus être réalisées).

L'environnement de programmation :

L'objectif du système d programmation est de préparer et de structurer l'ensemble des connaissances qui sont nécessaire pour gérer et contrôler l'exécution.

Outils pour la programmation avancée des taches de manipulation:

Le but de ce paragraphe est montré, sans aucune prétention à l'exhaustivité, la grande variété des problèmes soulevés par la programmation et la conduite de taches d'assemblage robotisées.

Au cours des dix dernières années des efforts importantes ont été consacrés à la programmation avancée des robots [107,108] de nombreux aspects ont été abordés.

la programmation automatique essentiellement fondée sur les techniques de raisonnement géométrique.

Des résultats significatifs ont été obtenus concernant notamment l'évitement d'obstacles

[64,109,110,111,112] la détermination des positions de prise des objets [113], la génération des mouvements fins [114,115,116], l'analyse des incertitudes de positionnement [117,118]. Ces différentes techniques constituent incontestablement les "briques de base" des systèmes de programmation à venir. Cependant deux difficultés demeurent. D'une part, la plupart de ces systèmes sont encore en cours de développement et ne peuvent traiter de manière complètement automatique que des cas relativement simples. D'autre part, l'intégration de ces modules décisionnels pose de problèmes difficiles.

En effets, dans le cas général, une opération d'assemblage ne peut se décomposer aisément en sous- problèmes résolus séparément (grands mouvement; mouvement d'approche, détermination des positions de prise; synthèse des stratégies d'insertion...)

Des travaux récents dans ce domaine sont en cours notamment dans le cadre des projets ATALAS- HANDEY [119,120,121]

le développement d'outils d'aide à la programmation au moyen de logiciels de type CAO [123,124] ce sont généralement des outils interactifs. Les principales fonctions fournies concernant l'aide à la configuration des cellules robotisés (choix de manipulateurs, emplacements ...), l'aide à la construction de trajectoires, simulation et la vérification des choix opérés par l'utilisateur.

Le développement de logiciels pour la programmation de tâches de montage complexe nécessitant notamment une interprétation des efforts et l'adoption de stratégies locales exprimées directement dans l'espace de la tâche [123,125,126,127,128] .

La modélisation des processus d'assemblage pour la génération automatique de séquences opératoires [129] ou la mise en œuvre d'un ordonnancement opportuniste des actions [105,106] .

Le contrôle d'exécution: ces recherches visent à doter le robot de moyens lui permettant de contrôler l'exécution d'un plan ou d'un programme [130,131,132,] l' système doit être capable de vérifier en permanence de conformité de l'état réel de la tâche avec l'état planifié.

Au plus bas niveau, la surveillance est effectuée principalement au moyen de capteurs.

Le système de conduite doit être capable de déterminer, en fonction de contexte, quelle

sont les surveillance qu'il doit activer. Par ailleurs, il doit être capable de réagir, avec un temps de réponse compatible avec la dynamique de l'environnement, à une grande variété d'évènements généralement asynchrone (échec d'une opération, détection d'une situation non prévue dans le plan courant, ordre en provenance de l'opérateur ou de l'atelier...). Les réactions du système peuvent entraîner des conséquences plus ou moins importantes sur l'exécution en cours: correction locale d'une anomalie de fonctionnement, amendement, planification totale, arrêt partiel.

Tableau R-4 : Principales formes d'usure [138].

Usure abrasive	à trois corps (low stress) à deux corps (high stress) coupe sous l'effet de chocs (gouging) meulage (grinding) polissage (polishing)
Usure adhésive (scuffing)	douce (oxidative wear) sévère avec grippage (seizure) galling
Corrosion induite par petits débattements (UIP- usure PEDEBA)	corrosion de contact ou corrosion de frottement (fretting wear, fretting corrosion) fatigue induite par la corrosion de contact (fretting fatigue) faux effet Brinell (fa/se brinelling)
Usure par fatigue superficielle	piqûration (pitting), écaillage (spalling) formation de taches grises (frosting)
Usure par fatigue thermique	
Usure par érosion et cavitation	érosion par un fluide (impingement érosion) érosion par un fluide chargé (slurry érosion) érosion par gouttelettes liquides cavitation (cavitation érosion)
Corrosion sous frottement	dépassivation

RÉFÉRENCES

1. B.MADANI ; « Dynamique de système multi corps ; application à la conception mécanique des robots », Mémoire de Magistère, I.N.G.M. Boumerdès, 07 Juin 1998.
2. E. AURELIEN ; « Cours de JAVA 2 », ENSERB Informatique.
3. C.BARTHLEY. D WILLIS ; « Conception mécanique cinématique et dynamique des robots », Revue Française de Mécanique N° 1995-4, Paris, Automne 1989.
4. M.VUKOBRATOVIK, M. KIRCANSKI ; « Kinematics and trajectory synthesis of manipulation robots », Springer Verlag, Berlin, 1986.
5. P.CHEDMAIL, E. DOMBRE, P. WANGER ; « CAO en robotique outils et méthodologies » Hermès, Paris, 1998.
6. N. SEGUY ; « Système de commande d'un manipulateur S7732 Traité Informatique Industrielle », © Technique de l'Ingénieur, France, Décembre 2005.
7. Y. KOREN ; « La robotique pour ingénieurs », Mc Graw Hill, Paris, 1986.
8. P. COIFFET ; « Les robots, Tome 1 : Modélisation et commande » Hermès, Paris, 1981.
9. L. PERROTTE ; « Modélisation et commande », Hermès, Paris, 1994.
10. M. GIORDANO, J.LOTTIN ; « Cours de robotique description et fonctionnement des robots industriels », Armand Colin, Paris, 1990.
11. J.GRANT, F.BARA ; « Introduction à la robotique » Dalloz, 1994.
12. W. KHALIL, E.DOMBRE ; « Modélisation identification et commande des robots », Hermès Sciences Publications, Paris, 1999.
13. J.DENAVIT, R.S.HARTENBERG ; « Kinematic rotation for lower-pair mechanisms based on matrices », Jour Appl Mech ASME 22.215-221, 1955.
14. P. COIFFET ; « Robots : définitions et classifications R7700 traité Mesures et Contrôle », © Technique de l'Ingénieur, France, Décembre 2005.

15. L. KHEMICI ; « Génération de trajectoires optimales coordonnées et sans collision en vue de la supervision multi-robots », Mémoire de Magister, Département d'Electronique, Université de Saad Dahlab de Blida, 2003.
16. P.COIFFET ; « La robotique : principes et applications », Hermès, Paris, 1986.
17. M. PARENT, C.LAURGEAU ; « Les robots : langages et méthodes de programmation », Hermès, Paris, 1983.
18. V.DUPOURQUE ; « Les contrôleurs de robots », collection Novotique INRIA/AFRI/ADI, 1986.
19. UNIMATION[®] : « Unimate Puma Mark II Robot, User's Guide de Val version 560.18.T.A », Unimation Inc[®] .,1983.
20. C. MELIN, H. HAMDI ; « Automatique Robotique R8 langages de programmation des robots R7720 », © Techniques de l'Ingénieur, France, Octobre 1991.
21. M.M. HATTALI ; « Logiciel de calcul de robots industriels application au robot de soudage ALG.SOUD.1 », Mémoire de Magister, Département de Mécanique, Université de Saad Dahlab de Blida, 2001.
22. E.DOMBRE ; « Environnement de programmation par CAO que faut-il entendre ? », 16p. 36 ref Actes séminaire SYSCOROB Collection A.F.R.I/A.D.I, 1986
23. R. POPPESTONE.P.AMBLER, I.BELLOS ; « An interpreter for a language for describing assemblies robot motion : planning and control », 585 p.317 réf, MIT Press, Cambridge Massachusetts, 1983.
24. M.AIT AHMED ; « Contribution à la modélisation géométrique et dynamique des robots parallèle », thèse de Doctorat, Toulouse, 1993.
25. K.JABELLA B.LASSAMI ; « Développement d'un logiciel d'animation et de commande pour Bras Manipulateurs », Projet de Fin d'études, Département Génie Electrique, Ecole National Poly Technique, El Harrach, Alger, 2002.
26. MARK W.SPDNG AND M.VIDYASAGAR ; « Robot Dynamics and control », Quinnwoodbine, USA, 1981.
27. JOHN J.CRAIG ; « Introduction to robotics : Mechanics and Control », 2nd ed, Addison Wesley Publishing Company, Canada, 1989.
28. R.J. SCHILING.; « Fundamentals of robotics: analysis and control», Prentice Hall, 1990.

29. B. ARMSTRONG, O. KHATIB AND J. BURDICK ; « The explicit Dynamique Model and Inertial parameters of the Puma 560 Arm », In ProcIEEE Int. Conf. On Robotics and Automation, San Francisco, C.A. ,1986.
30. T.MADANI ; « Différentes Approches de Commande Décentralisée à Structure Variables Appliquées en Robotique », Mémoire de Magister, Ecole National Polytechnique, Alger, 2000.
31. J.P LALLEMAND, S.ZEGHLOUL ; « Robotique ; Aspects Fondamentaux, Modélisation Mécanique C.A.O. Robotique Commande », Masson, Paris, Milan, Barcelone, 1994.
32. S.CHARENTUS ; « Modélisation et commande d'un manipulateur redondant composé de plusieurs plate formes de Stewart », Thèse de Doctorat, Université Paul Sabatier, Toulouse, France, Avril 1990.
33. B. GORLA, M.RENAUD ; « Modèles des robots manipulateurs : application à leur commande », Cepadues Edition, 1984.
34. P. BORREL; « contribution à la modélisation géométrique des robots manipulateurs : application à la conception assistée par ordinateur », Thèse d'État, U.S.T.L. Montpellier, Juillet 1986.
35. ANGELES; « Fundamentals of Robotic Mechanical Systems : theory methods and algorithms Mechanical Engineering Series», Springer, Verlag, 1997.
36. M .MINOUX ; « Programmation mathématique : théorie et algorithmes, tome 1 », Éditions Dunod, Bordas et C.E.N.T E.N.S.T., Paris, 1983.
37. Z. MICHALEWICZ ; « Genetic algorithms +data structures = evolution programs », Springer, Verlag, 1992.
38. A. PRUSKI ; « Robotique générale », Ellipses, Paris, 1988.
39. R.J HOOKER; J-PEREIRA ; « an integrated robot analysis procedure vol , N° 10 , PP1069 », IEEE, 1997.
40. A. LIÉGEOIS ; « Modélisation et commande des robots manipulateurs R7730, automatique robotique R8 © Techniques de l'ingénieur », France, Octobre 1991.
41. M. RENAUD ; « Contribution à la modélisation et à la commande dynamique des robots manipulateurs », Thèse Doctorat d'Etat Université Paul Sabatier de Toulouse, Septembre 1980.
42. H. ASADA J.J. SLOTINE ; « Robot analysis and control », John Willey & sons, New York, 1990.

43. M.J. ALDON ; « Elaboration automatique des méthodes dynamiques des robots en vue de leur commande », Thèse de Doctorat d'État, Université de Langue de Doc. Montpellier, 1982.
44. A.ALAPETITE, P.COHADE, J.PETTRE ; « B E de vision en robotique sur robot COGNEX-SANKYO », <http://alexandre.alapetite.net/dess-irr/robotique/sankyo/index.html>. Alexandre Alapetite, Pierre Cohade, Julien Pettre 16-01-2003.
45. A. CASSANO, A, CARDANO ; « A comparison between three variable step algorithms for the integration of motion in structural dynamics », Latin American Research, 1991.
46. COMPUTER SIMULATION OF MANIPULATOR DYNAMICS USING DIFFERENT CONTROL LAWS : “Third International Conference on Advanced Robotics”, Versailles, 1987.
47. DYNAMIQUE BEHAVIOUR OF DEFORMABLE AND RIGID ARTICULATED SYSTEMS : “Structural Mechanics in Reactor Technology 10th Conf.”, Los Angels, August 1989.
48. SHYING HER LIN , SABRI TOSUNOGLI et DELBERT TESAR ; « Control of a six degree-of- freedom flexible industrial manipulator », I.E.E.E. Journal of Robotics Research, 1995.
49. P.ANDER , J.N KAUFMAN ; LHOTE ; JP TAIL LARD ; « Le robots, tome 4 : les constituants technologiques », Hermès, Paris, 1983.
50. K.A TABOUB ; P.C MULLER ; « A new control method applied to robot with joint elasticity », PP 565-570, I.E.E.E., 1994.
51. K.P JAUKOWSKY ; H.A EL MAGHY ; « Dynamic decoupling for hybride control of rigid/flexible joint robots interacting with the environment vol 22 » , N°4 , PP 736-747 I.E.E.E., 1992.
52. KHOSMAN ; « Adaptive control of flexible joint robots vol 8 », N° PP 250-267 I.E.E.E., 1992.
53. M.C READMAN ; P.R BELANGER ; « Analysis and control of flexible joint robot », I.E.E.E., 1990.
54. P.RUSSEL ; « Procedure of control for flexible joint robots », I.E.E.E., 1997.
55. M. HADDAD; « Modélisation des déformations des bras manipulateurs par les concepts de base de la théorie des poutres évaluation et compensation des erreurs » Mémoire de Magister, © E. M. P. Bordj El Bahri, Alger, Septembre 1999.

56. A. YOUSNADJI; « La robotique et son environnement rétrospective et aperçu général sur les bras manipulateurs actes des journées d'études sur la robotique et son environnement », © E.N.I.T.A.ROB'95, du 16 au 18 Septembre 1995.
57. K.TOUMADJ « étude et conception d'un robot type utilisé pour le soudage par résistance a Alger SOUD 1, partie cinématique », Projet fin d'étude, Université Saad Dahlab de BLIDA, Département d'Aéronautique , 1988.
58. A.BENSAFIA ; « Etude du comportement dynamique des système polyarticulé plans flexibles », Mémoire de Magistère, I.N.G.M. Boumerdès, 14 Juin 1996.
59. M. BENZAOUI; « Les techniques de commande sous contraintes d'un robot manipulateur redondant », Mémoire de Magister, Automatique, systèmes intelligents de commande et robotique Ecole National Polytechnique, El Harrache, Alger, 2006.
60. GERARD ROORYCK ; « Aspects généraux de la robotique », R 7700, © Techniques de l'Ingénieur, France, Octobre 1991.
61. SAIDOUNI TARIK ; « Etude critique et classification des méthodes de description des robots manipulateurs », Thèse de D.E.A. F.N.S.A.M., Paris, 19 Septembre 1990.
62. G.GOYU ; P. COIFFET ; A.BARRACO ; « Mathématique pour la robotique : Représentation des déplacements des robots », Hermès, Paris 1997.
63. J.P I ALLEMAND ; S.ZEGHLOUL ; « Robotique aspects fondamentaux : Modélisation mécanique, C.A.O. Robotique, Commande », Masson, Paris, Barcelone, 1994.
64. J.D. BOISSONNAT B. FAVERJON J.P. MERLET ; « Techniques de la robotique tome 2 perception et planification » Hermès, Paris, 1988.
65. PARC PRIFL ; « Les robots industriels : Caractéristiques performances et choix » A.F.NOR.
66. Z.ROTH. BENJAMIN ; W. MOORING. BAHRAM RAVANI ; « An overlew of robot calibration », I.E.E.E., Journal of Robotics and Automation vol.R.A3 N° 5, October 1987.
67. A. LIEGOIS ; « Les robots : Analyse des performances et C.A.O. Tome7 » Hermès, Paris, 1984.
68. M.C READMAN ; P.R BELANGER ; « Analysis and control of flexible joint robot », I.E.E.E., 1990.
69. B.RUSSEL ; « Procedure of control for flexible joint robots », I.E.E.E., 1997.
70. A.BELAID , A. KHOUKIH ; « Compensation des erreurs systématiques de positionnement des robots industriels », C.N.P. 1^{er} Colloque National sur la Productique, Tizi ouzou, Algérie, 1998.

71. A . BOUGUERRA ; « Contribution à la planification optimale des robots cooperants », Mémoire de Magister, Département de Mécanique, Université de Saad Dahlab de Blida, Janvier 2005.
72. P . ATHANASIOS ; « Probability random variables and stochastic processes », W.C.B., 1991.
73. S . KRIK PATRIK, C.D GELATT and P.M VECCHI ; «Optimization by simulated annealing science » 220 p 671.681, 1983 .
74. J . LAM and J. M . DELOSME ; « Logic minimization using simulated annealing Proc.», International Conference on Computer Aided Design I.C.C.A.D.86, Santa Clara C.A. p.348-351, Novembre 1986.
75. J.LAM and J.M DELOSME « Simulated annealing ; a fast heversitic for some generatic layout problems »
- 76 . R. BATTITI, G TECCHIOLI ; « The reactive tabu search », O.R.S.A. Journal on Compuling p.126-140, 1994.
77. F . GLOVER ; « Tabu search » port 2 ORSA J. on computing 1(3) p. 190-206. 1989.
78. F. GLOVER ; « Tabu search » port 2 ORSA J. on computing 2(1) p. 04.32. 1990.
79. T . CRAINIC M. GENDERAU P. SORIANO and M. TOULOUSE ; « A tabu search procedure for multicommodity location all ocation with balancing requirement », Annals of Operations Research, 42 (1- 4) p.359-383, 1993.
80. D. DAVIS Cd ; « Genetic algorithms and simulated annealing », Morgan Kaufmann.
81. R. DORSEY and W.MAYER ; « Genetic algorithm for estimation problems with multiple optimal: Nondiffe rentability and other irregular features », Journal of Business and Economic Static's, 13 (1) p. 53-60, 1995.
82. K. KRISHNAKUMAR and D.GOLDBERG ; « Control system optimization using genetic algorithm », Journal of Guidance, Control and Dynamics, 15(3) p.735.740, 1992.
83. M. KIRKANSKI O. TIMENKO ; « A geometric approach to manipulator path planning in 3d space in the presence of obstacles avoidance », Robotica, vol, 10p. 321-328, 1992.
84. P. BORNE. G.D TANGUY, J.P RICHARD, F. ROTELLA I.ZAMBETTAKIS ; «Commande et optimisation des processus », Editions Techni, 1990.
85. J.J. GRAIG ; « Introduction to robotics, mechanics and control », Addison Wesley.
86. W KHALIL, J.F KLEIN FINGER ; « A new geometric notation for open and close loop robots », Proceeding of the I.E.E.E. int. Conf. on Robotics and Automation, p 1174.1180, San Francisco , 1986.

87. R.P.PAUL ; « Robot manipulators, mathematics, programming and control », The M.I.T. Press., Cambridge, 1981.
88. A.A. KOBRINSKI , A.E .KOBINSKI ; « Bras manipulateurs des robots : Architecture et théorie », édition Mir, Moscou, 1989.
89. V .ARNOLD ; « Les méthodes mathématiques de la mécanique classique », Edition Mir, Moscou, 1976.
90. L. LANDAU , E.LIFCHITZ ; « Mécanique », édition Mir, Moscou, 1988.
91. R.L. FOX « Optimization methods for engineering design », Addison-Wesley, 1971.
92. T. LALIBERTE , C.M. GOSSELIN « Efficient algorithm for the trajectory planning of redundant manipulators with obstacles avoidance », I.E.E.E., International Conference on Robotics and Automatics, vol.3, p. 2044,2049, May 1994.
93. Z. SHILLER « Time optimal control of articulated systems with geometric path constraints », I.E.E.E. int. Conf. on Rob and Aut., vol 4p, 2680.2685, May 1994.
94. T. CHETTIBI , H. E. LEHTIHET ; « A new approach for point to point optimal motion planning probles of robotic manipulators » E.SD.A.2002 A.P.M.-10 A.S.M.E. conf. 2002.
95. K. GLASS, R.COLBAUGH, D. LIM, H. SERADJ ; « Real time collision avoidance for redundant manipulators », I.E.E.E. Transaction on Rob and Aut. N° 11,vol 3p. 448-457, 1995.
96. S.MITSI , K.D BOUZAKIS , G. MANSOUR ; « Optimization of robot links motion in inverse kinematics solution considering collision avoidance and joints limits », Mach.and Mec. Theory, N°30 vol 5.p 653.663, 1995.
97. R.V. MA YORGA ; « A frame work for the path planning of robot manipulator », Lasted Thiol Int. Conf. on Rob. and Manufacturing, p61.66, June 1995.
98. F.DANES ; « Critères et contraintes pour la synthèse optimale de robots manipulateurs : Applications à l'évitement d'obstacles » Thèse de Doctorat d'Etat, Université de Poitiers, 1998.
99. ARTOBOLEVSKI « Théorie des mécanismes et des machines », © Editions Mir, Moscou, 1977.
100. D. PAYANNET ; « Modélisation et correction des erreurs statiques des robots manipulateurs », Thèse de Docteur Ingénieur, Montpellier, U.S.T.L., Octobre 1985.
101. M. ABDELHADI ; « Etude et réalisation d'un système de C.A.O. pour la robotique; application à la modélisation l'évolution et la simulation des robots manipulateurs », Thèse de Docteur Ingénieur, U.ST.L., Montpellier, Juillet 1988.

102. C.QUARO ; « Etude et réalisation d'un système graphique interactif pour la robotique », Thèse de Docteur Ingénieur, U.S.T.L., Montpellier, Octobre 1985.
103. E. SCRIVE ; « Etude et réalisation d'un système de C.A.O. pour la robotique, application à la programmation graphique des robots de type SCARA », Thèse de 3^{ème} Cycle, U.S.T.L., Montpellier, Juillet 1988.
104. G. GIRALT ; « Research trends in decisional and multisensory aspects of third generation robots », 2nd I.S.R.R., Kyoto, Japan, Août 1984.
105. B.R. FOX .K.G. KEMPE; « Opportunistic scheduling for robotic assembly », I.E.E.E., International Conférence en Robotics and Automation, Saint-Louis, Avril 1985.
106. LUIZ S.HOMEM DE MELLO , A.C SANDERSON ; « AND./OR. graph. représentation of assembly plans », C.M.U-R.I-T.R-86-8, Carnegie- Mellon University, 1986.
107. J.C LATOMBE ; « Une analyse structure d'outils de programmation pour la robotique industrielle », Séminaire International Sur Les Méthodes de Langages de Programmation de Robots Industriels I.R.I.A., Juin 1979.
108. T. LOZANO-PEREZ ; « Robot programming », Al. Memo Artificial Intelligence Lab. M.I.T, 1982.
109. T. LOZANO-PEREZ ; « Automatic planning of manipulator transfer movements » , I.E.E.E., Transaction on System, Man and Cybernetics, S.M.C. 11.10, 1981.
110. L.GOUZENES ; « Strategies for solving collision-free trajectories problems for mobile or manipulator robot international », Journal of Robotics Research, vol3, N°4, Winter 1984.
111. B.FAVERJON ; « Obstacle avoidance using an octree in the configuration space of a manipulator », I.E.E.E. International Conference on Robotics and Automation, Atlanta, Mars 1984.
112. B. FAVERJON, P. TOURNASOUD ; « Planification de trajectoire et systèmes multi-robots : Technique de la robotique », tome 1, Hermès, 1988.
113. J. TROCCAZ ; « Modélisation du raisonnement géométrique pour la programmation de robots », Thèse I.N.P.G., France, Juillet 1986.
114. M. T .MASON; « Manipulator grasping and pushing operations », P.H.D. Thesis MIT. Al. Lab., 1982.
115. J.M VALADE ; « Geometric reasoning and synthesis fo assembly trajectory 85», I.C.A.R, Japon, Septembre 1985.

116. C. LAUGIER, PUGET, P. THEVENEAU ; « Traitement des incertitudes en programmation automatique des robots : Techniques de la robotique », Hermès, tome 1, Paris, 1986.
117. R.A. BROOKS ; « Symbolic error analysis and robot planning », the International Journal of Robotics Research, vol 1 N°4, Winter 1983.
118. TROCCAZ. P PUGET ; « Dealing with uncertainties using program proving method », Uth. I.S.R.R. Santa Cruz, USA, Aout 1987.
119. T. LOZANO-PEREZ, R A BROOKS ; « A approach to automatic robot programming » , Al. Memo 842, Al. lab. MIT., 1984.
120. T. LOZANO-PEREZ , J.L. JONES, E. MAZER et AL. HANDY; « A robot system that recognizes : Plans and manipulates » I.E.E.E., International Conference on robotics and automation, Raleigh, USA, Avril 1987.
121. E. MAZER , T. LAZANO-PEREZ ; « The structure of an interpreter for task-level robot programs » I.C.A.R. 87, Versailles, Octobre 1987.
122. C. LAUGIER. J. TROCCAZ; « A system for automatic programming of manipulation robots » 3rd International Symposium of Robotics Research, Gouvieux, France, Octobre 1985.
123. JOURNÉES ANNUELLES DU PROGRAMME A.R.A. ; Toulouse, Septembre 1984.
124. E. DOMBRE, P BOREL, A LIÉGEOIS ; « A CAD. system for industrial automation », vol 2, 1984.
125. A. GIRAUD ; « Generalized active compliance for part mating with ensemble robots », Isrt. I.S.R.R., Britton Woods, Septembre 1983.
126. C. REBOULET .A ROBERT ; « Hybrid control of a manipulator equipped with an active compliant wrist : Robotics research 3», Mit press., USA, 1986.
127. O. KHATIB , I BURDICK ; « Motion and force control or robot manipulators », I.E.E.E. International Conference on Robotics and Automation, San Francisco, USA, Avril 1986.
128. J. IRIGOYEN ; « Commande en position et en force d'un robot manipulateur d'assemblage », Doctorat de l'Université Paul Sabatier, Toulouse, Octobre 1986.
129. A. BOURJAULT ; « Contribution a une approche méthodologique de l'assemblage robotisé; l'élaboration automatique des séquences opératoires », Thèse d'Etat, Université de Franch-Comté, Besançon, Novembre 1984.

130. M. GHALLAB ; « Task execution monitoring by compiled production rules in an advanced multi- sensor robot », 2nd Intentional Symposium or Robotics Research, Kyoto, Japon, Août 1984.
131. DONNEES TECHNIQUES : « Machines outils, perceuses radiales, TGL 42 995 », République démocratique allemande, Aout 1985.
132. DONNEES TECHNIQUES : « Appareil de fraisage vertical ApFS 250/2 » Caractéristiques techniques « DZFG 200 x 500 » ; Atelier de fabrication mécanique, Département de Génie Mécanique, Université Saad-Dahleb de Blida.
133. R. MAZARI, N. MOHAMMEDI « Etude dynamique et devant projet de conception d'un Robot de soudage [ALG.SOUD.1] », Projet de fin d'étude, Département de Génie Mécanique, Université Saad-Dahleb de Blida, 1998.
134. N.MEGHERBI, N.BEDIAF : « Programmation d'un robot manipulateur des machines-outils (ALGERIE-M.O.1) », Projet de Fin d'Etude Département d'Aéronautique Université de Saad-Dahleb, 2005.
135. A.TIAIBA : « Etude et conception d'un Robot manipulateur type (ALGERIE-M.O.1) », Mémoire de fin d'étude, Département de Génie Mécanique, Centre Universitaire de M'sila, 1995.
136. A.ALLALI, A.BRAHIMI, M.M.HATTALI : «Model mathématique d'un robot de soudage par résistance par points [type : ALG.Soud.-1] » Communication aux JMA 2000.
137. A.ALLALI, A.BRAHIMI, A.BENMISRA ,M.K HALAIMIA : «Model mathématique d'un système mécanique articulé» Communication aux NCMES'07, May 26-27,2007.
138. A.ALLALI, A.BRAHIMI, A.BENMISRA, M.K HALAIMIA : «Robot manipulateur des machines outils», Brevet d'invention, I.N.A.P.I. N° de dépôt 060205, Alger, Mai 2007.
139. A.ALLALI, D.TOUMI, A.BRAHIMI, M.M.HATTALI : «Robot de soudage par résistance», Brevet d'invention, I.N.A.P.I. N° de dépôt 990204, Alger, Octobre 1999.
140. R. CHATILA ; « mobile robot navigation space modeling and decisional processes : Robotics Research 3», MIT. Press., USA, 1986.
141. E. LOPEZ MELLADO ; « Le control d'exécution dans les cellules flexibles d'assemblages », Thèse de Docteur- Ingénieur U.P.S. L.A.A.S., Toulouse, Décembre 1986.
142. R.S SMITH, M. GINI ; « Robot tracking and control issues in an intelligent error recovery system », I.E.E.E. International Conference on Robotics and Automation, San Francisco, USA, Avril 1986.

143. V. DUPOURQUE ; « Architecture matérielles des contrôleurs de robots : Techniques de la robotique », tome 1, Hermès, 1988.
145. D.A BOURNE, MS FOX ; « Autonomous manufacturing automating the job- shop » I.E.E.E. Computer vol. N° 4, Winter 1983.
146. A.R SANDERSON , G. PERRY ; « Sensor- based robotic assembly systems; research and application in electronic manufacturing », proceedings of the I.E.E.E., Vol 71, N° 7, Juillet 1983.
147. R. CASSINIS ; « Resource allocation in industrial multirobot systems », I.E.E.E. International Conference on Robotics and Automation, San Francisco, USA, Avril 1986.
148. A. CORNET, J.-P. DEVILLE ; « Physique et ingénierie des surfaces », ©EDP Sciences, Paris, 1998.
149. D. DRIBINE ; « Modélisation cinétique et dynamique », Thèse de Fin d'Etude Electronique, Ecole Nationale Polytechnique, El Harrache, Alger.