

UNIVERSITE DE SAAD DAHLAB DE BLIDA

Faculté des Sciences de l'ingénieur

Département de Génie Mécanique

MEMOIRE DE MAGISTER

Spécialité : construction mécanique

**SYNTHESE, ANALYSE ET MODELISATION DES ROBOTS
MANIPULATEURS**

Par

FERNINI Brahim

Devant le jury composé de :

A. GUESSOUM	Professeur, U. de Blida	Président
H. ZAHLOUL	Maitre de conférences A, U.de Chlef	Examineur
A. ALLALI	Maitre de conférences A, U. de Blida	Examineur
M. OUALI	Professeur, U.de Blida	Rapporteur
M. MEGHATRIA	Maitre assistant A, U.de Blida	Co-Rapporteur

Blida, Mai 2012

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

REMERCIEMENTS

Ce travail a été élaboré au département de mécanique de l'université de SAAD DAHLAB-BLIDA. Je saisis cette opportunité pour remercier tous mes enseignants d'avoir contribué à enrichir mon esprit par les connaissances scientifiques.

Que mes professeurs, Messieurs (M.MEGHATRIA, M.OUALI, M.GOUASMI), trouvent ici l'expression de mes meilleurs sentiments pour avoir dirigé ce travail.

Je remercie tous les membres du jury pour avoir accepté de juger ce mémoire.

Ma chaleureuse reconnaissance est adressée à tous ceux qui m'ont aidé de loin ou de près à réaliser ce travail.

Je dédie ce travail à mes parents, mes frères, Abdenour et Tasnim.

Enfin, j'adresse mes plus vifs remerciements à tous mes amis.

RÉSUMÉ

L'un des objectifs de ce travail est une initiation au domaine multidisciplinaire de la robotique.

L'étude du comportement dynamique d'un système robotisé mène à des équations de mouvement qui souvent sont délicates à résoudre. La modélisation d'une scène de manipulation d'un objet par des robots comporte, aussi bien pour l'objet que pour les robots les modèles suivants : le modèle géométrique, le modèle cinématique et le modèle dynamique. A cet effet, on aborde la modélisation d'un type de robot avec une application au robot Scara. Le but est de comparer entre deux postures du robot pour une même nature de trajectoire et une même durée et d'établir un code de calcul permettant l'obtention des paramètres cinématiques et dynamiques du robot.

On utilise deux logiciels SolidWorks(2010) et Matlab/Simulink(2009) pour vérifier la théorie et la simulation du mouvement du robot. La vérification des résultats obtenus par ces deux logiciels permet de valoriser qualitativement, de mettre en relief la pertinence du modèle étudié et d'apporter les conclusions adéquates. Les résultats obtenus sont confrontés à ceux de la littérature.

Mot-clés : Robotique; Systèmes robotisés; Robot SCARA ; Modélisation mathématique ; Modèle cinématique ; Comportement dynamique; trajectoire; manipulabilité; Simulation.

ABSTRACT

Initiation into the multidisciplinary field of robotics is the main aim of this work. Studying the dynamic behaviour of a robotic system often leads to very hard resolving kinematic equations.

Object staging modelisation using robots holds, whether for the object or the robot, the following models: The geometric one, the kinematics one and the dynamic one.

To do so, the modelisation of a robot type with a SCARA application is being implemented. Comparing between two robot postures with the same trajectory (path) and for the same length of time and establishing a computing code to obtain the kinematic and dynamic parameters are the main tasks.

SolidWorks(2010) and Matlab Simulink(2009) softwares are used to check the theory and the robot motion simulation.

The verification of the obtained results by both softwares allows us to, qualitatively evaluate, underline the rightness of the chosen model and to get the right conclusions. The results obtained are confronted with those in the robotic field.

Keywords : Robotics ; Multibody Systems ; Scara Robot ; Modelisation ; Kinematic Model ; Dynamic Behaviour ; Manipubality ; Simulation.

ملخص

من بين أهداف هذا العمل هو مقدمة لحقل متعدد التخصصات من الرجال الآليين. دراسة السلوك الديناميكي للنظام الآلي يؤدي إلى معادلات الحركة التي غالبا ما تكون صعبة الحل. نماذج من التلاعب مشهد كائن الروبوتات تشمل كلا من طرازات كائن لالروبوتات : نموذج هندسي، ونموذج ونموذج الحركية الديناميكية. ولهذه الغاية ، ونحن نقترح النمذجة نوع من الروبوت الروبوت SCARA مع التطبيق. الهدف هو مقارنة بين اثنين من مواقف للروبوت لمسار مماثل، ونفس المدة وإنشاء رمز الكمبيوتر للحصول على المعلمات الحركية والدينامية للروبوت. نستخدم اثنين برنامج (2010) SolidWorks ومطلب / Simulink (2009) للتحقق من نظرية المحاكاة والحركة الروبوت. التحقق من النتائج التي حققتها هذه البرامج يضيف قيمة نوعية لتسليط الضوء على أهمية النموذج دراستها وتقديم الاستنتاجات المناسبة. تتم مقارنة النتائج مع تلك التي تم الحصول عليها من الأدب.

الكلمات الرئيسية : الروبوتات والأنظمة الروبوتية، SCARA روبوت ، النمذجة الرياضية والحركية النموذجي manipulability سلوك مسار ديناميكي؛ المحاكاة.

SOMMAIRE

INTRODUCTION	1
CHAPITRE 1 : GENERALITES & ETAT DE L'ART.	
1.1 Historique	3
1.2 Justification technico-économique.....	5
1.2.1 L'automatisation quasi intégrale de la production	5
1.2.2. La flexibilité.....	5
1.3 Le caractère répétitif des tâches à accomplir.....	6
1.4. La production de séries moyennes.....	6
1.5 Accroissement de la capacité de production.....	7
1.6 Caractère pénible ou dangereux des opérations.....	7
1.7. Qualité du produit et soin apporté à sa fabrication.....	7
1.8. Manutentions.....	7
1.9 Classification des robots.....	7
1.9.1 Classification fonctionnelle.....	8
1.9.2 Classification géométrique.....	9
1.10 Cellule robotisée.....	11
1.10.1 <i>Le mécanisme</i>	11
1.10.2 <i>La perception</i>	11
1.10.3 <i>La commande</i>	11
1.10.4 <i>Interface homme machine</i>	11
1.10.5 <i>Le poste de travail</i>	11
1.11 Description des robots.....	11
1.11.1 <i>Définitions</i>	11
1.11.2 <i>La versatilité</i>	12
1.11.3 <i>L'auto-adaptabilité</i>	12
1.12 Constituants mécaniques d'un robot manipulateur.....	12
1.12.1 Structure mécanique.....	12
1.13 Organe terminal.....	13
1.14 Système mécanique articulé.....	13

1.15 Morphologie des robots manipulateurs.....	15
1.16 Définitions et terminologie.....	17
1.16.1 Redondance.....	17
1.16.2 Singularité.....	18
1.16.3 La parcourabilité.....	19
1.16.3.1 Une trajectoire discrète T_d	19
1.16.3.2 Les trajectoires continues T_c	20
1.17 Conclusion.....	21

CHAPITRE 2 : MODELISATION DES BRAS MANIPULATEURS.

2-1 Introduction.....	22
2-2 Structures en chaînes ouvertes arborescentes (multiple open-tree structure).....	22
2-3 Structures en chaîne articulaire ouverte simple (simple open-tree structure).....	22
2-4 Structure complexe (multiply connected structure).....	23
2-5 Qu'est-ce qu'un système multi corps ?.....	23
2-5-1 Modélisation des systèmes multicorps.....	25
2-5-2 Coordonnées.....	27
2-5-3 Coordonnées absolues.....	27
2-5-4 Formalismes.....	28
2-6 Structure géométrique.....	28
2-6-1 Modèle géométrique.....	29
2-6-1-1 Modèle géométrique direct.....	29
2-6-1-2 Modèle géométrique inverse.....	30
2-6-2 Modèle cinématique.....	30
2-6-2-1 Modèle cinématique direct.....	30
2-6-2-2 Modèle cinématique inverse.....	31
2-6-3 Modèle dynamique.....	31
2-6-3-1 Modèle dynamique inverse.....	31
2-6-3-1-1 Formalisme de Lagrange.....	31
2-6-3-1-2 Formalisme de Newton-Euler.....	34
2-6-3-1-3 Equation de NEWTON.....	34
2-6-3-1-4 L'équation d'EULER.....	34
2-6-3-1-5 Récurrence avant pour calculer les vitesses et les accélérations.....	34
2-6-3-1-6 L'algorithme itératif de dynamique de Newton-Euler.....	37

2-6-3-1-7 Inclusion des forces de pesanteur dans l'algorithme de dynamique.....	37
2-6-3-2 Modèle dynamique direct.....	37
2-6-3 Comparaison entre les deux formalismes.....	38
2-7 Les types des robots.....	38
2-7-1 Les robots sériels.....	38
2-7-1 -1 Rapport charge utile / masse du robot.....	39
2-7-1-2 Précision de positionnement.....	39
2-7-1-3 Masses importantes en mouvement.....	39
2-7-2 Les robots parallèles.....	39
2-7-3 Comparaison des architectures sérielles et parallèles.....	40
2-8 Evitement d'obstacle mobile.....	41
2-8-1 Méthode de fenêtre dynamique.....	41
2-8-1-1 Espace de recherche.....	41
2-8-2 Approche du V-Obstacle.....	42
2-8-3 Autres approches d'évitements d'obstacle mobile.....	43
2-9 Conclusion.....	44

CHAPITRE 3 : PLANIFICATION DE TRAJECTOIRES.

3-1 Introduction.....	45
3-2 La problématique générale.....	45
3-3 Loi d'orientation.....	47
3-5 Modélisation de l'espace de planification.....	47
3-5-1 Notions et définition.....	47
3-5-2 Espace de configuration.....	48
3-5-3 Espace libre.....	48
3-6 Construction de l'espace de planification.....	49
3-7 Génération de trajectoires.....	49
3-7-1 Introduction.....	49
3-7-1-1 Définition.....	49
3-7-2 Planification dans l'espace articulaire.....	50
3-7-3 Planification dans l'espace cartésien (opérationnel) X.....	50
3-7-3-1 Interpolation et régulation cartésienne.....	50

3-7-3-2 Interpolation et régulation articulaire.....	51
3-7-3-3 Trajectoires planifiées dans l'espace articulaire.....	51
3-8 Coordination des articulations.....	53
3-9 Trajectoire contrainte (points de passage intermédiaires).....	53
3-10 Interpolation cartésienne.....	54
3-11 Interpolation articulaire.....	54
3-12 Contraintes spatiales.....	54
3-12-1 Identification du problème.....	54
3-13 Les divers aspects de la planification de trajectoires.....	55
3-14 Contraintes temporelles.....	55
3-15 Synthèse des principaux travaux en planification de mouvements.....	56
3-15-1 Introduction.....	56
3-15-2 Méthodes par décomposition cellulaire.....	56
3-15-3 Méthodes exactes.....	57
3-15-4 Méthodes approchées.....	58
3-15-5 Méthodes de résolution de type rétraction.....	58
3-16 Autres méthodes de planification.....	59
3-16-1 Méthode du potentiel.....	59
3-17 Méthode des contraintes.....	60
3-18 conclusion.....	60

CHAPITRE 4 : APPLICATION.

4-1 Introduction.....	61
4-2 Modélisation des bras manipulateurs.....	61
4-3 Problème de la cinématique directe.....	61
4-3-1 Les paramètres de (D-H) du robot SCARA choisi.....	62
4-3-2 Calcul de la matrice de transformation homogène par deux méthodes.....	64
4-3-2-1 La première méthode.....	64
4-3-2-2 La deuxième méthode.....	65
4-4 Le modèle cinématique inverse.....	67
4-4-1 Le problème général de la cinématique inverse.....	67
4-4-2 Détermination des variables articulaires du robot SCARA.....	67
4-4-2-1 La position désirée.....	67
4-4-2-2 Calcul de la variable articulaire θ_2	68

4-4-2-3	Calcul de la variable articulaire θ_1 par deux méthodes.....	68
4-4-2-3-1	La première méthode.....	68
4-4-2-3-2	La deuxième méthode.....	69
4-4-3	Equation du coude haut (elbow up).....	70
4-4-4	Equation du coude bas (elbow down).....	70
4-4-5	Vérification des résultats obtenus.....	70
4-4-5-1	Application numérique.....	70
4-4-6	Calcul l'orientation de l'organe terminal θ_4	70
4-4-6-1	Calcul de la vitesse linéaire.....	71
4-4-6-2	La vitesse de translation.....	71
4-4-6-3	Calcul de l'accélération.....	71
4-5	Application.....	72
4-5-1	La position initiale (home position).....	72
4-5-2	L'équation du mouvement pour coude bas (elbow down) pour la position initiale et désirée.....	74
4-5-3	L'équation du mouvement pour coude haut (elbow up) pour la position initiale et désirée.....	75
4-5-4	Vérification de trajectoire.....	76
4-5-4-1	Vérification de la trajectoire a l'aide Solidworks.....	77
4-5-4-2	Vérification de la trajectoire à l'aide du logiciel Matlab.....	79
4-5-5	La variation des angles des deux postures.....	79
4-5-6	Vérification des résultats en utilisant.....	79
4-5-6-1	le logiciel Solidworks.....	79
4-5-6-2	MATLAB/simulink.....	83
4-5-7	Orientation de la matrice de transformation homogène pour les deux postures à $t=1s$	86
4-5-8	La position du robot durant la simulation.....	90
4-5-8-1	SolidWorks.....	90
4-5-8-2	Matlab/Simulink.....	94
4-5-9	les chémas des deux postures du robot SCARA pour la meme position désirée.....	96
4-6	Analyse de l'organe terminal.....	98
4-6-1	Précision dans la prise d'objet.....	98
4-6-2	Orientation de l'organe terminal.....	98
4-6-3	Déplacement en translation de l'organe terminal.....	97
4-6-3-1	Vérification des résultats obtenus a l'aide le logiciel Matlab.....	101

4-7	Calcul des vitesses dans l'espace opérationnel.....	102
4-7-1	La matrice jacobienne.....	102
4-7-2	La matrice jacobienne pour un point quelconque.....	102
4-7-3	La matrice jacobienne du robot SCARA.....	103
4-7-4	Les vitesses dans l'espace opérationnel.....	104
4-8	La singularité du robot SCARA.....	105
4-9	La manipulabilité.....	107
4-9-1	Application de la manipulabilité au robot SCARA.....	107
4-9-2	mesure de la manipulabilité du robot SCARA.....	109
4-9-3	Indice de longueur structurelle du robot SCARA.....	109
4-10	Calcul des couples d'articulations.....	110
4-10-1	Paramètre des liens.....	110
4-10-2	Calcul les couples des articulations.....	111
4-10-2-1	par la méthode de Lagrange.....	111
4-10-2-1-1	La matrice d'inertie.....	113
4-10-2-1-2	Matrice de Coriolis et de centrifuge.....	113
4-10-2-1-3	Vecteurs des forces de gravité.....	113
4-10-2-2	Vérification des résultats par l'algorithme de Newton-Euler.....	113
4-10-2-3	Vérification des résultats par la méthode de la mécanique classique.....	116
4-11	Comparaison entre les deux postures.....	119
4-12	Conclusion.....	120
	CONCLUSION	121

BIBLIOGRAPHIE

APPENDICE

LISTE DES FIGURES.

Figure 1.1 : Coûts de production en fonction du volume de production.....	7
Figure 1.2 : classe A : Télém manipulateur.....	9
Figure 1.3 : Classe B C D (robot intelligent).....	9
Figure 1.4 : Coordonnées cartésiennes et cylindriques.....	10
Figure 1.5 : Coordonnées polaires et universelles.....	10
Figure 1.6 : Robot SCARA.....	11
Figure 1.7 : Robot a chaine ouverte simple.....	12
Figure 1.8 : Symbole de <i>l'articulation rotoïde</i>	14
Figure 1.9 : Symbole de <i>l'articulation prismatique</i>	14
Figure 1.10 : Morphologie des robots manipulateurs.....	16
Figure 1.11 : les types de poignets.....	17
Figure 1.12 : le schéma d'un porteur.....	17
Figure 1.13 : Trajectoire discrète.....	19
Figure 1.14 : Trajectoire continue.....	20
Figure 2.1 : Une chaine arborescente.....	22
Figure 2.2 : Une chaine ouverte simple.....	23
Figure 2.3 : Une chaine fermée.....	23
Figure 2.4 : Structure arborescente ou bouclée.....	24
Figure 2.5 : Les paramètres de (D-H) (α_i, d_i, θ_i) : défini pour l'articulation i et lien i.....	29
Figure 2.6 : Représentation d'un point r^i de la chaîne cinématique.....	33
Figure 2.7 : La force exercée sur le lien i.....	34
Figure 2.8 : le couple exercé sur le lien i.....	35
Figure 2.9 : les forces agissant sur le lien i.....	36
Figure 2.10 : robot à architecture sérielle.....	38
Figure 2.11 : robot à architecture parallèle.....	40
Figure 2.12 : Modélisation des robots à 2 degrés de liberté utilisés pour la comparaison.....	41
Figure 2.13 : Espace de recherche de vitesses.....	42
Figure 2.14 : Représentation géométrique du concept de V-obstacle.....	43
Figure 3.1 : Génération de trajectoires.....	49
Figure 3.2 : Planification dans l'espace cartésien (opérationnel) X.....	51
Figure 3.3 : Loi bang-bang pour un axe : $\ddot{q} = k_a$ sur $(0, T/2)$, $\ddot{q} = -k_a$ sur $(T/2, T)$	52
Figure 3.4 : Loi bang-bang avec palier de vitesse.....	52

Figure 3.5 : Lois alternatives à la loi Bang-Bang.....	52
Figure 3.6 : Lois alternatives à la loi Bang-Bang.....	52
Figure 3.7 : la continuité en vitesse et accélération aux points intermédiaires de lift-off et set-down.....	53
Figure 3.8 : Décomposition en cellule génération du mouvement.....	57
Figure 3.9 : Champ de potentiel attractif et répulsif.....	60
Figure 4.1 : le lien i avec son articulation initiale $i-1$ et son articulation finale i	62
Figure 4.2 : Schéma du robot de type SCARA.....	62
Figure 4.3 : Le robot SCARA schématisé à l'aide logiciel Solidworks.....	63
Figure 4.4 : Schéma cinématique du robot SCARA.....	63
Figure 4.5 : la rotation des deux liens autour de son axe de rotation Z	64
Figure 4.6 : le schéma du robot SCARA dans cette étude.....	72
Figure 4.7 : La position initiale du robot SCARA pour les deux postures : logiciel Matlab/simulink.....	73
Figure 4.8 : La position initiale du robot SCARA pour les deux postures : logiciel Solidworks.....	73
Figure 4.9 : La trajectoire obtenue lors du déplacement de la <i>posture1</i>	77
Figure 4.10 : La trajectoire obtenue lors du déplacement de la <i>posture2</i>	77
Figure 4.11 : Trajectoires obtenues lors du déplacement des deux postures en même temps..	78
Figure 4.12 : la trajectoire obtenue lors du déplacement des deux postures.....	79
Figure 4.13: La variation des angles des deux postures.....	79
Figure 4.14 : La position de la <i>posture1</i> à l' instant $t=0.4s$	80
Figure 4.15 : La position de la <i>posture 2</i> à l' instant $t=0.4s$	80
Figure 4.16: La position de la <i>posture1</i> à l' instant $t=0.6s$	81
Figure 4.17: La position de la <i>posture2</i> à l' instant $t=0.6s$	81
Figure 4.18 : La position de la <i>posture1</i> à l' instant $t=1s$	82
Figure 4.19 : La position de la <i>posture2</i> à l' instant $t=1s$	82
Figure 4.20 : La position de la <i>posture1</i> à l' instant $t=0.4s$	83
Figure 4.21 : La position de <i>posture2</i> à l' instant $t=0.4s$	83
Figure 4.22 : La position de la <i>posture1</i> à l' instant $t=0.6s$	84
Figure 4.23 : La position de la <i>posture2</i> à l' instant $t=0.6s$	84
Figure 4.24 : La position de la <i>posture1</i> à l' instant $t=1s$	85
Figure 4.25 : La position de la <i>posture2</i> à l' instant $t=1s$	85
Figure 4.26 : L'orientation de la matrice de transformation homogène <i>posture1</i> à $t=1s$	86

Figure 4.27 : L'orientation de la matrice de transformation homogène <i>posture2</i> à $t=1s$	86
Figure 4.28 : Déplacement et vitesse du lien1 suivant x (<i>posture1</i>).....	87
Figure 4.29 : Déplacement et vitesse du lien1 suivant y (<i>posture1</i>).....	87
Figure 4.30 : déplacement et vitesse du lien2 suivant x (<i>posture1</i>).....	88
Figure 4.31 : Déplacement et vitesse du lien2 suivant y (<i>posture1</i>).....	88
Figure 4.32 : Déplacement et vitesse du lien1 suivant x (<i>posture2</i>).....	89
Figure 4.33 : Déplacement et vitesse du lien1 suivant y (<i>posture2</i>).....	89
Figure 4.34 : Déplacement et vitesse du lien2 suivant x (<i>posture2</i>).....	90
Figure 4.35 : Déplacement et vitesse du lien2 suivant y (<i>posture2</i>).....	90
Figure 4.36 : La position de <i>posture1</i> a l'instant $t=0s$ (home position) (SW)	91
Figure 4.37 : La position de <i>posture1</i> à l' instant $t=0.8s$ (SW).....	91
Figure 4.38 : La position de <i>posture1</i> à l'instant $t=1s$ (la position désirée) (SW)	92
Figure 4.39 : La position de <i>posture2</i> a l'instant $t=0s$ (home position) (SW).....	92
Figure 4.40 : La position de <i>posture2</i> à l' instant $t=0.8s$ (SW)	93
Figure 4.41 : La position de <i>posture2</i> à l'instant $t=1s$ (la position désirée) (SW).....	93
Figure 4.42 : La position de <i>posture1</i> a l'instant $t=0s$ (home position) (SW)	94
Figure 4.43 : La position de <i>posture1</i> à l' instant $t=0.8s$ (M/S).....	94
Figure 4.44 : La position de <i>posture1</i> à l'instant $t=1s$ (la position désirée) (M/S)	95
Figure 4.45 : La position de <i>posture2</i> a l'instant $t=0s$ (home position) (M/S).....	95
Figure 4.46 : La position de <i>posture2</i> à l' instant $t=0.8s$ (M/S)	96
Figure 4.47 : La position de <i>posture2</i> à l'instant $t=1s$ (la position désirée) (M/S).....	96
Figure 4.48: Les deux <i>postures</i> pour un robot SCARA.....	97
Figure 4.49 : Les deux <i>postures</i> à l'aide logiciel Solidworks pour notre cas.....	97
Figure 4.50 : Les deux <i>postures</i> à l'aide le logiciel Solidworks.....	97
Figure 4.51 : Organe terminal avec 4 tringleries de barre.....	98
Figure 4.52 : Schéma montrant l'orientation de l'organe terminal et l'objet à manipuler sur la scène de manipulation.....	99
Figure 4.53 : l'orientation de l'organe terminal et l'objet à manipuler sur la scène de manipulation.....	99
Figure 4.54: Déplacement et vitesse en translation de l'organe terminal (Solidworks).....	101
Figure 4.55 : Accélération en translation de l'organe terminal (Solidworks).....	101
Figure 4.56 : Déplacement et vitesse en translation de l'organe terminal (MATLAB).....	102
Figure 4.57 : Accélération en translation de l'organe terminal (MATLAB).....	102
Figure 4.58 : manipulateur plan a 3 bras.....	103

Figure 4.59 : espace opérationnel de <i>posture1</i>	105
Figure 4.60: espace opérationnel de <i>posture2</i>	105
Figure 4.61 : La singularité du robot SCARA.....	106
Figure 4.62 : La variation de $\det(J)$ en fct de θ_2 pour différentes valeurs de θ_1	107
Figure 4-63 : les variables articulaires dans le robot SCARA.....	108
Figure 4-64 : la manipulabilité d'un robot SCARA.....	109
Figure 4.65 : Mesure de la manipulabilité en fonction du temps.....	109
Figure 4.66 : Schéma dynamique du robot SCARA.....	110
Figure 4.67: la variation de l'énergie cinétique totale (<i>posture1</i>).....	117
Figure 4.68: la variation de l'énergie potentielle totale (<i>posture1</i>).....	117
Figure 4.69: le couple de l'articulation1 (<i>posture1</i>)	118
Figure 4.70 : le couple de l'articulation2 (<i>posture1</i>).....	118
Figure 4.71 : la variation de l'énergie cinétique totale (<i>posture2</i>).....	118
Figure 4.72 : la variation de l'énergie potentielle totale (<i>posture2</i>).....	119
Figure 4.73: le couple de l'articulation1 (<i>posture2</i>).....	119
Figure 4.74 : le couple de l'articulation2 (<i>posture2</i>)	119

LISTE DES TABLEAUX

Tableau 1.1 : la production des robots dans les années 80-90.....	4
Tableau 2.6.3 Nombre d'opérations effectuées pour chaque formalisme.....	38
Tableau 4.3.1 : DH paramètres du robot SCARA.....	64
Tableau 4.5.4 : la position des deux postures durant 1s.....	76
Tableau 4.10.1 : Paramètres des liens d'une structure SCARA.....	111

NOMENCLATURE

c : Le cout de production unitaire.

F : Le cout fixe.

v : Le cout variable.

n : Le nombre des pièces produites.

Td : Une trajectoire discrète.

Tc : Les trajectoires continues.

$H\mathbb{Q}$: ensemble de configuration

(q, \dot{q}, \ddot{q}) : la position, les vitesses et les accélérations des variables articulaires.

F_{ext} : les forces extérieures.

L_{ext} : les couples extérieurs.

g : les forces de gravité.

Q : les couples appliqués.

$J(q)$: la matrice jacobienne

λ : Le vecteur des multiplicateurs de Lagrange.

$h(q)$: Les équations de contraintes.

\ddot{R}^G : est l'accélération absolue du centre de masse du corps

m : est la masse du corps

F : est la résultante des forces appliquées sur le corps

I^G : est le tenseur d'inertie du corps, par rapport à son centre de masse

ω : le vecteur vitesse angulaire du corps

L^G : est la résultante des moments des forces appliquées sur le corps, par rapport à son centre

$M(q)$: La matrice de masse généralisée du système.

$c(q, \dot{q}, F_{ext}, L_{ext}, g)$; Les termes gyroscopiques de Coriolis et les forces couples extérieures appliquées sur le corps ainsi que la gravité.

a_i : (Link length).

α_i : (Link twist).

d_i : (Joint distance).

θ_i : (Joint angle)

m_i : Masse du corps i de la chaîne cinématique.

R_0 : Repère globale

R_i : Repère du lien i .

${}^i r$ Vecteur du centre d'inertie du corps i exprimé dans le repère R_i

$m^i r$: Premier moment du corps i autour de o_i exprimé dans R_i

${}^i S_i$: Position du centre de masse du corps i exprimée en R_i

H_i : Matrice des pseudo-inerties du corps i

g : Vecteur des accélérations de la gravité exprimé dans R_0

(x, y, z) : Les axes des liens dans un repère orthogonal.

l_1 : La longueur du lien1 (m)

l_2 : La longueur du lien 2 (m).

d_3 : La longueur de l'organe terminal.

\dot{d}_3 : La vitesse de l'organe terminal (m/sec)

\ddot{d}_3 : L'accélération de l'organe terminal (m/sec**2)

p_x : La position désirée selon l'axe (x)

p_y : La position désirée selon l'axe (y)

p_z : La position désirée selon l'axe (z)

θ_1 : Le déplacement angulaire du lien 1 (rad)

$\dot{\theta}_1$: La vitesse angulaire du lien 1 (rad/sec)

$\ddot{\theta}_1$: L'accélération angulaire du lien 1 (rad/sec**2)

θ_2 : Le déplacement angulaire du lien 2 (rad)

$\dot{\theta}_2$: La vitesse angulaire du lien 2 (rad/sec)

$\ddot{\theta}_2$: L'accélération angulaire du lien 2 (rad/sec**2)

θ_4 : L'orientation de l'organe terminal (rad)

$\dot{\theta}_4$: La vitesse angulaire de l'organe terminal (rad/sec)

$\ddot{\theta}_4$: L'accélération de l'organe terminal (rad/sec**2)

m_1 : La masse du lien 1 (Kg)

m_2 : La masse du lien 2 (Kg)

T_1^0 : La matrice qui représente la première transformation.

T_2^1 : La matrice qui représente la deuxième transformation.

T_3^2 : La matrice qui représente la troisième transformation.

T_4^3 : La matrice qui représente la quatrième transformation.

T_4^0 : La matrice finale qui représente le robot.

T_H^R : La matrice de la position désirée.

$J(q)$: La matrice jacobienne du robot SCARA.

\dot{X} : Les vitesses dans l'espace opérationnel (m/sec).

\dot{q} : Les vitesses des variables articulaires (rad/sec**2)

q_{B1} : La variable articulaire du lien 1 dans le cas de manipulabilité (rad)

q_{B2} : La variable articulaire du lien 2 dans le cas de manipulabilité (rad)

L : Le lagrangien du système

K : Énergie cinétique totale du robot (J)

V : Énergie potentielle totale du robot (J)

τ_1 : Le couple de l'articulation 1 (N.m)

τ_2 : Le couple de l'articulation 2 (N.m)

g : La force de pesanteur (9.81 m/sec**2)

f_3 : La force exercée de l'organe terminal sur le lien 2 (N)

n_3 : Le Couple exercé de l'organe terminal sur le lien 2 (N.m)

ω_0 : La rotation de la base du robot (rad)

$\dot{\omega}_0$: La vitesse angulaire de la base du robot (rad/sec)

${}^0\dot{V}_0$: La vitesse linéaire de la base du robot (m/sec)

ω_1 : La rotation du lien 1 selon l'axe z (rad)

${}^1\dot{\omega}_1$: La vitesse angulaire du lien 1 selon l'axe z (rad/sec)

${}^1\dot{V}_1$: La vitesse linéaire du lien 1 (m/sec)

${}^1\dot{V}_{C1}$: La vitesse du centre de masse du lien 1 (m/sec)

1F_1 : Les forces agissant sur le lien 1 (N)

1N_1 : Les couples agissant sur le lien 1 (N.m)

${}^2\omega_2$: La rotation du lien 2 selon l'axe z (rad)

${}^2\dot{\omega}_2$: La vitesse angulaire du lien 2 selon l'axe z (rad/sec)

${}^2\dot{V}_2$: La vitesse linéaire du lien 2 (m/sec)

${}^2\dot{V}_{C_2}$: La vitesse du centre de masse du lien 2 (m/sec)

2F_2 : Les forces agissant sur le lien 2 (N)

2N_2 : Les couples agissant sur le lien 2 (N.m)

K_1 : Énergie cinétique du lien 1 (J)

V_1 : Énergie potentielle du lien 1 (J)

K_2 : Énergie cinétique du lien 2 (J)

V_2 : Énergie potentielle du lien 2 (J)

INTRODUCTION

Représentant un incroyable défi, les robots interprètent un important aboutissement en matière d'évolution des technologies : << Que peut-il y avoir de plus difficile que de reproduire avec une machine, des caractéristiques du fonctionnement animal ou humain ? >>. Ce défi technologique mobilise depuis au moins un demi-siècle des énergies considérables, et chacune des étapes majeures dans l'évolution de la robotique constitue une victoire de l'intelligence humaine. Depuis le début des années 1970, les recherches dans le domaine de la robotique ont pris une allure croissante mettant en œuvre plusieurs résultats innovants dans d'autres disciplines connexes :

- Mathématiques : Développement des algorithmes efficaces pour la résolution des Systèmes complexes.
- Intelligence artificielle : Domaine connexe aux mathématiques qui consiste généralement en l'imitation des systèmes biologiques donnant parfois la possibilité de l'auto apprentissage comme les réseaux de neurones, ainsi que la diversité dans la création de nouveaux systèmes et la sélection pseudo naturelle comme les algorithmes génétiques.
- Electronique : Développement des systèmes <<intelligents>> pouvant même être embarqués sur le robot afin de le doter de la perception nécessaire pour pouvoir prendre certaines décisions concernant son comportement.
- Informatique : Développement des langages de programmation de plus en plus évolués.

Actuellement la présence des robots envahit les sphères d'activité humaine. Dans la vie domestique nous trouvons le robot subvenir en partie aux travaux ménagers et publiques (robot serveur, robot garde malade, hôtesse d'accueil, etc.) et dans l'industrie (soudure à l'arc, pistolage, assemblage, découpage, polissage, etc).

Une famille d'équipements industriels parmi la variété qui existe, est celle qui assure le mouvement exigé par un processus de fabrication comme la soudure à l'arc ou le découpage, d'où l'évolution vers la classe d'équipement de type de plus en plus populaire qu'est le robot industriel. Les différentes configurations de manipulateur sont disponibles et les plus utilisées sont : la configuration rotoïde et prismatique.

Un robot de configuration rotoïde, le bras du robot est articulé par conformité sélective (SCARA) à quatre degrés de libertés, ce robot est conçu au Japon, approprié généralement aux

tâches peu importantes d'insertion de pièces pour des chaînes de montage comme l'insertion de composant électronique [1]. Comme la théorie sert comme préalable à l'expérimentation, dans notre cas on a procédé à diverses simulations par logiciels interposés (Solidworks2010, Matlab/simulink2009).

Les possibilités technico-mathématiques qui sont offertes de nos jours pour effectuer des simulations en temps réel deviennent particulièrement incontournables dans les stades avancés du processus de conception. La conception finale peut être vérifiée avant de se lancer dans un processus coûteux de fabrication d'un prototype [2].

La modélisation consiste en une étape essentielle de tout sujet de robotique. Que le sujet ait l'aspect d' << analyse >> ou de << conception >>, un modèle mathématique s'avère indispensable. Ce modèle doit tenir compte de tous les paramètres qui peuvent influencer sur le comportement du robot.

L'objet de notre travail consiste à partir d'un centre d'intérêt qui est le robot SCARA à étudier dans un premier temps le modèle mathématique qui constitue son fondement intrinsèque puis procéder en une simulation des équations de mouvement à partir des deux logiciels Solidworks et Matlab/simulink afin de veiller à la validation, à chaque étape de l'étude, à la conformité des résultats obtenus avec le modèle mathématique du robot.

Dans ce mémoire, on abordera la modélisation du robot SCARA, robot sériel dans l'objectif d'étudier le modèle cinématique et dynamique. Ce mémoire est organisé comme suit : le premier chapitre est consacré à l'état de l'art et à la terminologie robotique. Dans le second chapitre on présentera la modélisation des bras manipulateurs sériels, les modèles : géométrique, cinématique et dynamique sont abordés en détail pour proposer un programme efficace et ayant l'aspect généraliste pouvant être exécuté pour tous types de robots SCARA. Dans le troisième chapitre on abordera la planification des trajectoires et la généralisation du mouvement ainsi qu'une synthèse des types des méthodes utilisées. Le dernier chapitre consiste en une application des modèles décrits pour le robot SCARA.

CHAPITRE 1

GENERALITES & ETAT DE L'ART.

1.1 Historique :

La robotique est un ensemble de disciplines techniques (mécanique, électronique, automatique, informatique) articulées autour d'un objectif et d'un objet commun. Cet objectif est l'automatisation flexible de nombreux secteurs de l'activité humaine réputés jusqu'à très récemment comme ne pouvant se passer de la présence de l'homme, et l'objet est le robot, sorte de machine universelle dont l'homme rêve depuis toujours (mais qui n'a pas encore la forme de ce rêve !).

D'une part, les progrès réalisés dans les différentes disciplines mentionnées plus haut permettent de construire des robots et autres systèmes "mécatroniques" performants et fiables. D'autre part, les situations économiques des pays industrialisés, conduisent à une compétition industrielle très vive, ne laissant survivre que les industries capables de produire à faible coût ou de fournir rapidement des produits nouveaux et/ou de qualité supérieure et garantie. Ces exigences ont conduit à l'implantation de robots dans de nombreuses branches d'activités. C'est l'apparition des robots et leur association à d'autres machines automatiques d'inspiration "classique" (par exemple, des machines-outils à commande numérique) qui permettent d'automatiser ou d'essayer d'automatiser des opérations considérées jusqu'ici comme non automatisables.

Il est à noter que, dans l'esprit du grand public, il existe deux domaines de la technologie qui sont étroitement liés à la science-fiction : les techniques spatiales et la robotique. Pour le premier de ces domaines, les médias audiovisuels ont largement contribué à le démystifier par une description objective et détaillée des conquêtes successives de l'homme dans l'espace. Pour le second, au contraire, les médias contribuent souvent à entretenir le mythe du robot en le présentant comme un être mécanique constitué de composants mystérieux, capable d'actions humaines, doué de la parole, mettant souvent une force surhumaine au service d'une intelligence primitive, et animé de sinistres intentions. Le mot robot trouve son origine dans une pièce de l'auteur tchèque Capek, écrite en 1923, intitulée "Les robots universels de Rossum". Le mot "robot" ou "robota" désigne en tchèque le travail ou un travailleur. Dans la pièce de Capek, les robots sont des humanoïdes créés par Rossum, dans l'espoir qu'ils se mettront au service de l'homme. Mais la réalité des robots est bien antérieure à l'œuvre de Capek. Nombreux sont les mécanismes automatiques qui ont été imaginés et proposés au cours des siècles. Leurs inventeurs ne disposaient malheureusement pas de la discipline extraordinairement

puissante que constitue l'électronique, pour le contrôle des processus qu'ils imaginaient et le traitement de l'information correspondante. Dans les associations que l'on établit entre la robotique et la science-fiction, il serait toutefois incomplet si on omet de faire référence à l'œuvre de Isaac Asimov († 6 avril 1992). Les robots imaginés par Asimov dans les années 40 sont des machines conçues de manière à réaliser leur travail en toute sécurité. Il cristallise cette notion dans trois lois de la robotique, dont la validité est largement reconnue [3] :

1.1.1. Un robot ne peut porter atteinte à un être humain. Le robot doit au contraire toujours agir en sorte que la sécurité de l'être humain soit assurée.

1.1.2. Un robot doit toujours obéir à l'homme, sauf s'il devait pour cela transgresser la première loi.

1.1.3. Un robot doit toujours veiller à sa propre sécurité, sauf s'il devait pour cela transgresser les deux lois précédentes.

Le concept d'un robot industriel a été breveté en 1954 par G.C. Devol (U.S. Patent N° 2988237).

Le brevet décrit comment construire un bras mécanique commandé capable d'accomplir certaines tâches dans l'industrie. En 1956, la société UNIMATION a été créée aux U.S.A., avec pour seul domaine d'activité la robotique ; 16 ans après elle produit de vrais robots industriels.

	1980	1981	1985	1990
<i>Japan</i>	6000	14000	85000	100 – 360000
<i>U.S</i>	3500	5000	25000	150 – 2500001
<i>Europe</i>	3700	7300	25000	62 – 200000
	13200	26300	135000	312 – 81000

Tableau 1.1 : la production des robots dans les années 80-90

A l'instar de ce que nous venons de vivre avec l'informatique, les projets technologiques en cours nous permettent d'assister à la succession sur une période de quelques années de plusieurs générations de robots industriels.

- Sur le plan technique, cette évolution se concrétise par l'intégration de la mécanique, l'électronique et l'informatique en une nouvelle discipline, la mécatronique, dont la robotique n'est qu'un des aspects au même titre que la conception et la fabrication assistées par ordinateur (CAO et FAO). Le mot "mécatronique" a été introduit par les japonais au début des années 80 pour signifier "l'harmonie des capacités mécaniques et électroniques".

- Sur le plan économique l'accroissement de la compétitivité oblige les pays industrialisés à produire toujours beaucoup plus.

- Sur le plan social le robot peut remplacer l'homme dans des postes de travail pénibles, insalubres ou dangereux. De plus, l'industrie de la robotique peut être elle-même génératrice d'emplois nouveaux plus nobles et plus qualifiés.

1.2 Justification technico-économique :

On peut considérer trois grands domaines d'application des robots.

Le domaine de la production : C'est dans ce domaine que l'effort est porté le plus par les industriels, qui voient de nombreux avantages à l'utilisation des robots, un accroissement de la productivité et une diminution de la main d'œuvre. En fait, l'association de robots entre eux et avec d'autres machines amène deux avantages fondamentaux par rapport aux modes de production traditionnels :

1.2.1 L'automatisation quasi intégrale de la production qui peut s'accompagner:

- d'une meilleure qualité du produit fini,
- d'une plus grande fiabilité dans le maintien de cette qualité,
- d'une meilleure adaptation de la quantité produite à la demande.

1.2.2. La flexibilité:

La rapidité de reconfiguration de l'unité de production quand on passe de la fabrication d'un produit à celle d'un produit voisin (exemple : fabriquer des modèles différents de voitures sur la même chaîne) ou bien quand un incident immobilise une machine de l'unité de production [4]

Le domaine de l'exploration : On veut exécuter des opérations dans un lieu :

- D'accès difficile : exploitation forestière, construction, maintenance, réparation ou nettoyage de bâtiments, lignes électriques, ...
- Dans un milieu hostile ou dangereux pour l'homme : les milieux sous-marin, spatial, irradié des centrales nucléaires, les sites de catastrophes ou d'accidents, incendies, les théâtres d'opérations militaires etc...

Le domaine de l'assistance individuelle : Pouvant permettre d'améliorer les conditions de vie des personnes handicapées, paralysées ou amputées. La robotique recouvre alors le domaine des :

- prothèses (mains et jambes artificielles par exemple) ;
- orthèses, structures rigides motorisées que l'on met autour d'un membre paralysé et qui entraînent ce dernier dans leurs mouvements ;
- téléthèses, destinées aux personnes paralysées des quatre membres (tétraplégiques)

Les raisons de la robotisation " automatisation " de la production sont technico- économiques.

1.3 Le caractère répétitif des tâches à accomplir:

S'il faut répéter inlassablement les mêmes gestes et opérations simples, une automatisation classique (dite "dure") sera sans doute préférable. Dès que le nombre de variations ou la complexité de la tâche augmente, il faut penser aux robots. A remarquer d'ailleurs que ceux-ci étant de moins en moins coûteux, une automatisation autrefois effectuées par une machine spécialement conçue à cet effet peut être aujourd'hui réalisée par le simple apprentissage de la tâche par un robot polyvalent. La combinaison de robots et de simples manipulateurs et/ou automates est également une solution extrêmement intéressante à de nombreux problèmes d'automatisation.

1.4. La production de séries moyennes:

La production de pièces dans l'industrie mécanique se répartit en :

- 1/4 de la production sous forme de grandes séries ;
- 3/4 de la production en petites et moyennes séries.

La production en grandes séries a fait depuis longtemps l'objet d'une automatisation poussée, par la mise en service de machines spéciales non convertibles "automatisation dure". De plus en plus, il faut aussi automatiser des productions en moyennes et même en petites séries, afin d'utiliser aux mieux les capacités de production et de diminuer les temps de passage sur machine et la quantité de pièces "en cours". Une telle automatisation doit être très flexible ; une robotisation bien conçue permet en effet de supprimer les temps morts nécessaires aux réglages entre des séries différentes, et même de remplacer la production par lots de pièces identiques par une production mélangeant différents produits sur une même chaîne de production. Il est à noter que cette flexibilité est permise essentiellement par le système informatique de gestion de la production ; on peut donc considérer l'automatisation flexible comme une solution logicielle, tandis que l'automatisation dure correspond à une réalisation matérielle.

Le choix entre automatisation dure, automatisation flexible et production manuelle résulte de l'importance des séries à produire.

Pour une production déterminée, le choix de l'une ou l'autre solution peut être dicté, par l'examen de courbes ou de tables donnant le coût de production unitaire en fonction du nombre d'unités produites. Les coûts de production se répartissent en coûts fixes F indépendants de la quantité de la production et en coûts variables V qui, en première approximation, sont proportionnels au nombre n de pièces produites : $V = v \times n$.

Le coût de production unitaire c est donc donné par la formule :

$$c = (F + v.n) / n$$

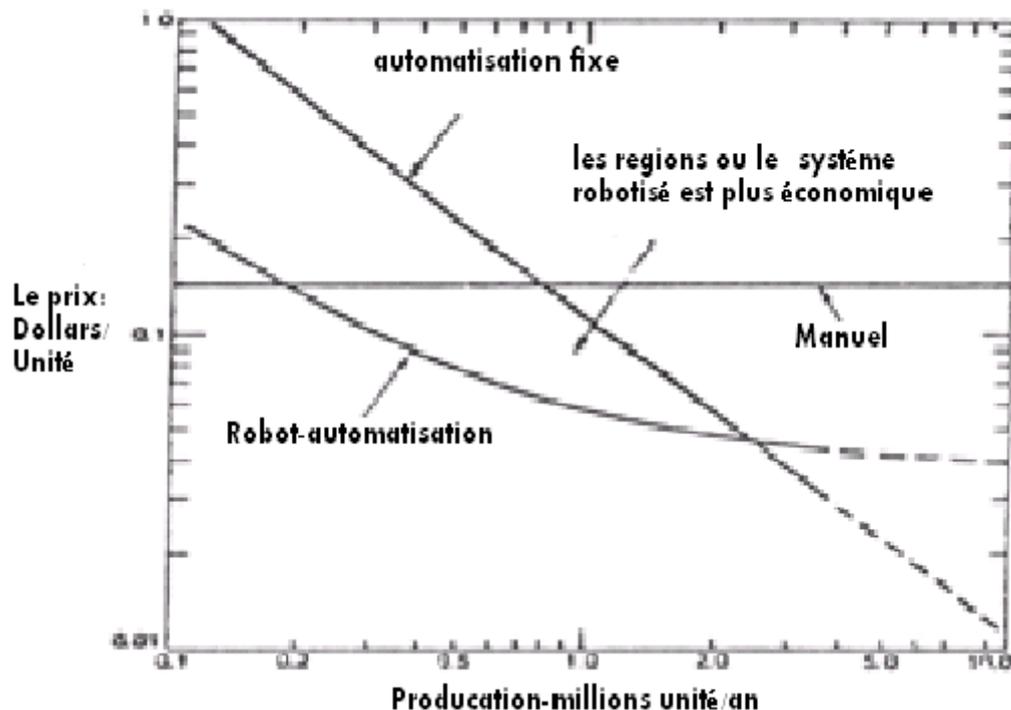


Figure 1.1 : Coûts de production en fonction du volume de production [3].

1.5 Accroissement de la capacité de production :

La robotisation de goulets d'étranglement d'une chaîne de production et la réduction des temps morts par robotisation permettent souvent un accroissement de 20 à 30 % de la capacité de production. Le travail en 2 ou 3 "pauses" (8 heures/jour) d'une chaîne robotisées permet de pratiquement doubler ou tripler la capacité de production sans demander un investissement supplémentaire pour les machines de base.

1.6 Caractère pénible ou dangereux des opérations :

L'exécution de ces tâches par des robots permet d'éviter l'installation de moyens de protection coûteux, contraignants et souvent peu efficaces, d'éviter l'organisation de longues (et coûteuses) périodes de repos pour les personnes exposées. La robotisation permet d'améliorer les conditions de travail et la productivité.

1.7. Qualité du produit et soin apporté à sa fabrication :

Le robot assure une qualité constante tout au long de son travail, alors qu'un ouvrier ne pourra généralement garder la même attention pendant toute sa période de travail et que la qualité de celui-ci risque donc de connaître des fluctuations importantes.

1.8. Manutentions :

Manutentions entre les opérations successives d'un processus de fabrication. Les robots sont bien adaptés aux manipulations d'objets (pièces, outils, ...) de masse moyenne ($\pm 0, 1 \text{ kg}$, $\pm 5 \text{ kg}$ ou $\pm 50 \text{ kg}$ suivant la catégorie de robot).

1.9 Classification des robots :

On peut classer les robots d'un point de vue fonctionnel ou d'après leur structure géométrique.

1.9.1 Classification fonctionnelle :

Le nombre de classe et les distinctions entre celles-ci varient d'un pays à un autre (6 classes au Japon, 4 en France). L'A.F.R.I. distingue 4 classes illustrées ci-dessous :

A : manipulateurs à commande manuelle ou télécommande.

B : manipulateurs automatiques à cycles pré-réglés (le réglage se fait mécaniquement par cames, butées, la commande peut se faire par automate programmable). On peut distinguer entre manipulateurs à cycle fixe et manipulateurs à cycle programmable.

C : robots programmables : c'est la première génération de robots industriels ; ils répètent les mouvements qu'on leur a appris ou programmés, sans informations sur l'environnement ou la tâche effectuée. On peut aussi faire la distinction entre robots "play-back" qui reproduisent la tâche apprise et robots à commande numérique qui peuvent être programmés hors-ligne. Pour de nombreux robots, l'apprentissage de la tâche se fait à l'aide d'un "syntaxeur" ("boite à boutons", "teach pendant") qui permet à un opérateur d'amener le robot en un certain nombre de points, qui sont ensuite mémorisés ; lors de l'exécution de la tâche, le robot suivra une trajectoire passant successivement par tous les points programmés, le passage d'un point au suivant se faisant suivant un profil de vitesse en fonction du temps qui est prédéfini (triangulaire ou trapézoïdal), l'opérateur n'ayant qu'à choisir la fraction de la vitesse maximum à laquelle il souhaite que le robot effectue la tâche. Pour certains robots, par exemple les robots de peinture, qui doivent suivre une trajectoire complexe qu'il est difficile d'exprimer mathématiquement, un opérateur humain spécialiste de la tâche effectue la trajectoire en guidant le bras du robot à l'aide d'un "pantin", et la trajectoire sera mémorisée par le robot. [5]

D : robots "intelligents". On trouve actuellement des robots de seconde génération qui sont capables d'acquérir et d'utiliser certaines informations sur leur environnement (systèmes de vision, détecteurs de proximité, capteurs d'efforts,...). On étudie des robots de troisième génération, capables de comprendre un langage oral proche du langage naturel et de se débrouiller de façon autonome dans un environnement complexe, grâce à l'utilisation de l'intelligence artificielle.

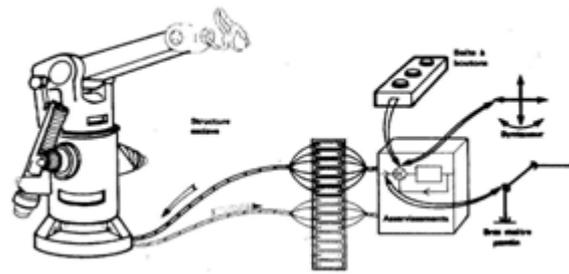
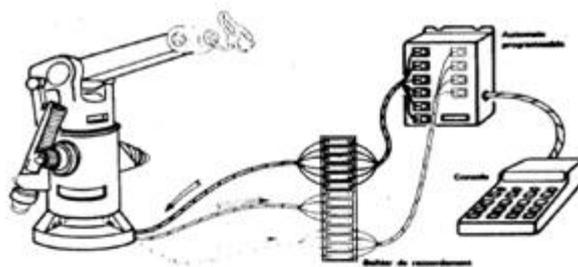
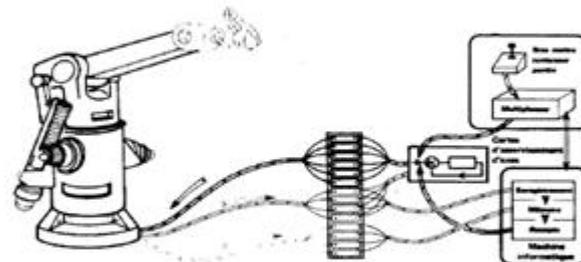


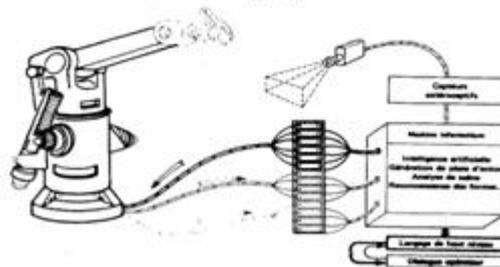
Figure 1.2 : classe A : Télém manipulateur [5]



classe B : Manipulateur à cycle pré-réglé



classe C : Robot programmable



classe D : Robot "intelligent"

Figure 1.3 : Classe B C D (robot intelligent). [5]

1.9.2 Classification géométrique :

On peut aussi classer les robots suivant leur configuration géométrique, autrement dit l'architecture de leur porteur. Les trois premiers degrés de mobilité d'un robot peuvent être réalisés en combinant les translations (max. 3T) et les rotations (max. 3R), autrement dit par des articulations prismatiques (P) ou rotoïdes (R). En pratique on n'utilise que 4 ou 5 d'entre elles :

- Porteur cartésien (TTT ou PPP) : les 3 axes sont animés d'un mouvement de translation.
- Porteur en coordonnées cylindriques (RTT ou RPP) : un mouvement de rotation et une translation axiale, complétées par une translation radiale
- Porteur en coordonnées polaires ou sphériques (RRT ou RRP) : deux rotations (longitude et latitude) autour d'axes orthogonaux, complétées par une translation radiale
- Porteur en coordonnées universelles, appelé aussi configuration polyarticulée ou anthropomorphe (RRR) : trois rotations dont les deux dernières se font autour d'axes parallèles orthogonaux au premier, les trois articulations correspondant respectivement au tronc (base), à l'épaule et au coude d'un être humain.

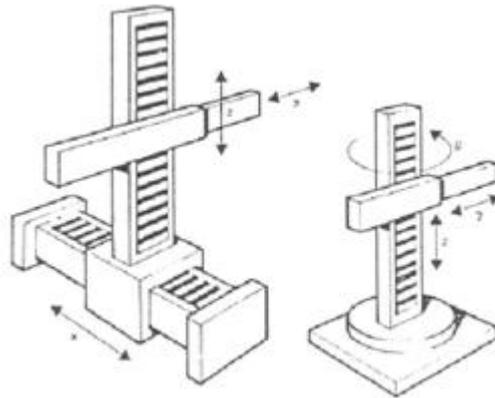


Figure 1.4 : Coordonnées cartésiennes et cylindriques. [5]

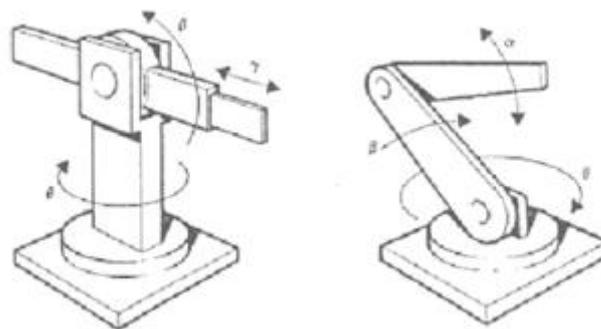


Figure 1.5 : Coordonnées polaires et universelles. [5]

Une cinquième architecture comprend deux rotations autour de deux axes parallèles, précédées ou suivies d'une translation dans la même direction (éventuellement celle-ci peut être reportée au niveau du poignet, qui peut aussi tourner autour du même axe, soit au total 4

ddm). Cette architecture est celle des robots SCARA (Selective Compliance Arm for Robotic Assembly) que l'on utilise dans des opérations d'assemblage.



Figure 1.6 : Robot SCARA. [5]

1.10 Cellule robotisée :

Généralement une cellule robotisée est constituée des composants suivants ;

1.10.1 Le mécanisme : ayant une structure proche de celle du bras humain, il permet de remplacer ou de prolonger son action, ses mouvements sont menés par des actionneurs électriques pneumatiques ou hydrauliques qui transmettent leurs mouvements aux articulations par des systèmes appropriés.

1.10.2 La perception : elle permet de gérer les relations entre le robot et son environnement les organes de perception sont des capteurs qui peuvent mesurer l'état interne du robot comme les positions et les vitesses des articulations, ou bien d'autres capteurs qui recueillent les informations sur l'environnement, détection d'objet, vision artificielle mesure de distance etc..

1.10.3 La commande : synthèse des consignes des asservissements pilotant les actionneurs.

1.10.4 Interface homme machine: dispositif à travers lequel l'utilisateur programme les tâches que le robot doit exécuter. Ce dispositif peut être une console de commande embarquée au robot ou un micro-ordinateur distant.

1.10.5 Le poste de travail: l'endroit ou l'environnement dans lequel évolue le robot.

1.11 Description des robots:

1.11.1 Définitions :

Le Petit Larousse définit un robot comme étant un appareil automatique capable de manipuler des objets, ou d'exécuter des opérations selon un programme fixe ou modifiable.

Pour "mériter" le nom de robot, un système doit posséder une certaine flexibilité, caractérisée par les propriétés suivantes : [6]

1.11.2 La versatilité : Un robot doit avoir la capacité de pouvoir exécuter une variété de tâches, ou la même tâche de différente manière.

1.11.3 L'auto-adaptabilité : Un robot doit pouvoir s'adapter à un environnement changeant au cours de l'exécution de ses tâches.

L'Association Française de Normalisation (A.F.N.O.R.) définit un robot comme étant un système mécanique de type manipulateur commandé en position, reprogrammable, polyvalent (i.e., à usages multiples), à plusieurs degrés de liberté, capable de manipuler des matériaux, des pièces, des outils et des dispositifs spécialisés, au cours de mouvements variables et programmés pour l'exécution d'une variété de tâches. Il a souvent l'apparence d'un, ou plusieurs, bras se terminant par un poignet. Son unité de commande utilise, notamment, un dispositif de mémoire et éventuellement de perception et d'adaptation à l'environnement et aux circonstances. 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.

1.12 Constituants mécaniques d'un robot manipulateur :

1.12.1 Structure mécanique :

La structure mécanique du robot manipulateur peut être divisée en deux parties distinctes comme le montre la figure 1-7.

- ♦ Organe terminal : les tâches qui sont dévolues aux robots sont très variées. Pour chaque opération ou travail spécifique, l'organe terminal prend un aspect particulier.
- ♦ Élément porteur : il est composé d'un ensemble de corps souples ou rigides liés par des articulations, servant à déplacer l'organe terminal d'une configuration à une autre.

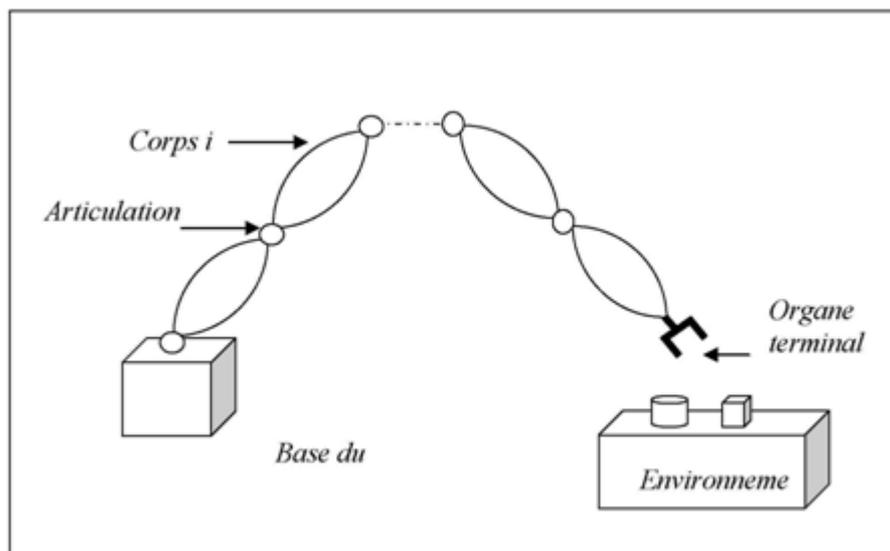


Figure 1.7 : Robot a chaîne ouverte simple. [6]

Avant de décrire les relations géométriques entre les différents corps du robot, on définit les notions suivantes :

- Degré de liberté : nombre de paramètres utilisés pour spécifier la configuration d'un élément de la chaîne cinématique par rapport à un autre.
- Degré de mobilité : toute articulation est caractérisée par son degré de mobilité m , c'est-à-dire le nombre de degrés de liberté entre deux corps successifs de la chaîne cinématique ($0 \leq m \leq 6$)
- Espace articulaire : représente l'état des corps composant le robot en fonction des variables articulaires, sa dimension n est égale au nombre de degrés de liberté du robot.
- Espace opérationnel : décrit la position et l'orientation de l'organe terminal du robot par rapport à un repère de référence, sa dimension m est inférieure ou égale à 6.

1.13 Organe terminal :

Sous le terme *organe terminal*, on regroupe tout dispositif destiné à manipuler des objets (dispositifs de serrage, dispositifs magnétiques, à dépression, ...), ou à les transformer (outils, torche de soudage, pistolet de peinture, ...). En d'autres termes, il s'agit d'une interface permettant au robot d'interagir avec son environnement. Un organe terminal peut être multifonctionnel, au sens où il peut être équipé de plusieurs dispositifs ayant des fonctionnalités différentes. Il peut aussi être monofonctionnel, mais interchangeable. Un robot, enfin, peut-être *multi-bras*, chacun des bras portant un organe terminal différent. On utilisera indifféremment le terme *organe terminal*, *préhenseur*, *outil* ou *effecteur* pour nommer le dispositif d'interaction fixé à l'extrémité mobile de la structure mécanique. [7]

1.14 Système mécanique articulé :

Le *système mécanique articulé* (S.M.A.) est un mécanisme ayant une structure plus ou moins proche de celle du bras humain. Il permet de remplacer, ou de prolonger, son action (le terme "manipulateur" exclut implicitement les robots mobiles autonomes). Son rôle est d'amener l'*organe terminal* dans une situation (position et orientation) donnée, selon des caractéristiques de vitesse et d'accélération données. Son architecture est une chaîne cinématique de corps, rigides (ou supposés comme tels), assemblés par des liaisons appelées *articulations*. Sa motorisation est réalisée par des *actionneurs* électriques, pneumatiques ou hydrauliques qui transmettent leurs mouvements aux *articulations* par des systèmes appropriés [8].

Précisons la notion d'*articulation* : Une *articulation* lie deux corps successifs en limitant le nombre de degré de liberté de l'un par rapport à l'autre. Soit m le nombre de degré de liberté résultant, encore appelé *mobilité de l'articulation*. La mobilité d'une articulation est telle que

$$0 \leq m \leq 6$$

Lorsque $m = 1$; ce qui est fréquemment le cas en robotique, l'*articulation* est dite *simple* : soit *rotoïde*, soit *prismatique*.

- Articulation rotoïde : Il s'agit d'une articulation de type pivot, notée R , réduisant le mouvement entre deux corps à une rotation autour d'un axe qui leur est commun. La situation relative entre les deux corps est donnée par l'angle autour de cet axe. Fig. 1.8



Figure 1.8 : Symbole de l'*articulation rotoïde*. [8].

- Articulation prismatique : Il s'agit d'une articulation de type glissière, notée P , réduisant le mouvement entre deux corps à une translation le long d'un axe commun. La situation relative entre les deux corps est mesurée par la distance le long de cet axe. Fig. 1.9



Figure 1.9 : Symbole de l'*articulation prismatique*. [8].

Remarque : Une articulation complexe, *i.e.*, avec une *mobilité* supérieure à 1, peut toujours se ramener à une combinaison d'articulations *prismatique* ou *rotoïde*. Par exemple, une rotule est obtenue avec trois articulations rotoïdes dont les axes sont concourants.

Pour être animé, le S.M.A. comporte des moteurs le plus souvent avec des transmissions (courroies crantées), l'ensemble constitue les *actionneurs*. Les actionneurs utilisent fréquemment des moteurs *électriques* à aimant permanent, à courant continu, à commande par l'induit (la tension n'est continue qu'en moyenne car en général l'alimentation est un hacheur de tension à fréquence élevée ; bien souvent la vitesse de élevée du moteur fait qu'il est suivi d'un réducteur, ce qui permet d'amplifier le couple moteur). On trouve de plus en plus de moteurs à commutation électronique (sans balais), ou, pour de petits robots, des moteurs pas à pas.

Pour les robots devant manipuler de très lourdes charges (par exemple, une pelle mécanique), les actionneurs sont le plus souvent *hydrauliques*, agissant en translation (vérin hydraulique) ou en rotation (moteur hydraulique).

Les actionneurs *pneumatiques* sont d'un usage général pour les *manipulateurs à cycles* (robots *tout ou rien*). Un manipulateur à cycles est un S.M.A. avec un nombre limité de degrés de liberté permettant une succession de mouvements contrôlés uniquement par des capteurs de fin de course réglables manuellement à la course désirée (asservissement en position difficile dû à la compressibilité de l'air) :

- La perception permet de gérer les relations entre le robot et son environnement. Les organes de perception sont des *capteurs* dits *proprioceptifs* lorsqu'ils mesurent l'état interne du robot (positions et vitesses des articulations) et *extéroceptifs*. Lorsqu'ils recueillent des informations sur l'environnement (détection de présence, de contact, mesure de distance, vision artificielle).
- La partie commande synthétise les consignes des asservissements pilotant les actionneurs, à partir de la fonction de perception et des ordres de l'utilisateur. [9]

S'ajoutent à cela :

- L'interface homme-machine à travers laquelle l'utilisateur programme les tâches que le robot doit exécuter.
- Le poste de travail, ou l'environnement dans lequel évolue le robot.

1.15 Morphologie des robots manipulateurs :

Ce paragraphe est relatif aux chaînes ouvertes simples. Afin de dénombrer les différentes architectures possibles, on ne considère que 2 paramètres : le type d'articulation (rotoïde (*R*) ou prismatique (*P*)) et l'angle que font deux axes articulaires successifs (0° ou 90° ; sauf cas très particulier, les axes consécutifs d'un robot sont soit parallèles, soit perpendiculaires).

On convient d'appeler les 3 premiers *d.d.l.* le *porteur du robot*. Les *d.d.l.* résiduels forment le *poignet*, caractérisé par des dimensions beaucoup plus petites et une plus faible masse. Sont schématisées dans la figure qui suit les 12 morphologies possibles de porteur (ces morphologies sont non redondantes (on élimine *a priori* les structures limitant les mouvements du porteur à des déplacements linéaires ou planaires 3 liaisons prismatiques d'axes parallèles, par exemple, ou 3 liaisons rotoïdes d'axes parallèles)). [10]

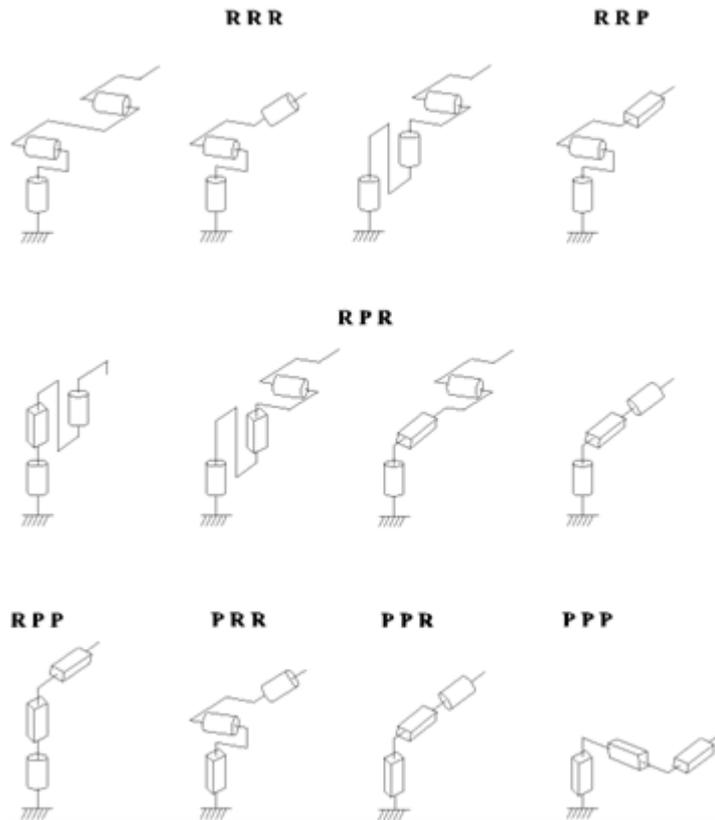


Figure 1.10 : Morphologie des robots manipulateurs. [10]

Dans la pratique, on trouve les 5 structures suivantes :

- Les porteurs anthropomorphes (RRR) : et plus précisément la première structure de cette classe (cf. figure précédente), comme par exemple les robots FANUC (LR, ARC), STÄUBLI RX, ACMA (V80 et SR400), UNIMATION (PUMA), SCEMI (6P-01), AID (V5), CINCINNATI (T3-7XX), AKR 3000, ASEA (IRB6 et 60), KUKA (IR600), AXEA (V08) ;
- Les porteurs sphériques (RRP) : comme par exemple les robots STANFORD, UNIMATION (1000, 2000, 4000), PSA (BARNABE) ;
- Les porteurs toriques (RPR) : et plus précisément la première structure de cette classe, comme par exemple les robots ACMA (H80), les robots de type SCARA (IBM, AXERA, ADEPT,...) ;
- Les porteurs cylindriques (RPP) : comme par exemple les robots ACMA (TH8), MANTEC (A, I et M), CINCINNATI (T3-363) ;
- Les porteurs cartésiens (PPP) : comme par exemple les robots ACMA (P80), IBM (7565), SORMEL (CADRATIC), OLIVETTI (SIGMA).

La structure *RRR* dont les 3 axes sont concourants forme ainsi une rotule et s'utilise plus généralement comme un *poignet*. D'autres types de poignets de un à trois axes sont représentés sur la figure suivante :

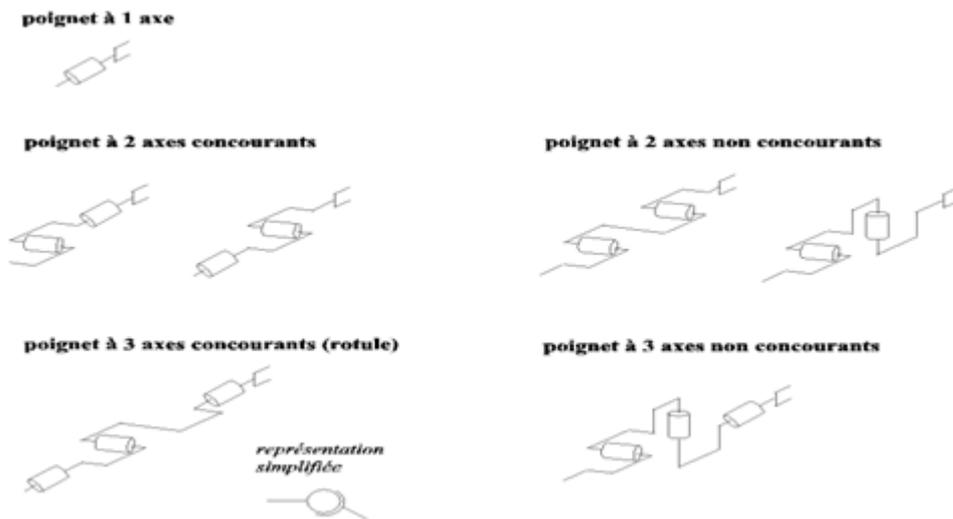


Figure 1.11 : les types de poignets. [10]

Dans la pratique, le *poignet* de type rotule est très répandu. Le robot, obtenu en lui associant un porteur à 3 *d.d.l.*, est la structure la plus classique à 6 *d.d.l.* Elle permet d'assurer un découplage entre la position et l'orientation de l'organe terminal :

- Le porteur a pour rôle de fixer la position du point d'intersection, noté *P*, des axes des 3 dernières articulations (centre du poignet) ; cette position (*P*) ne dépend que de la configuration des solides (corps) 1, 2 et 3 (*i.e.*, du porteur),
- Le poignet est destiné à l'orientation de l'organe terminal (pince, outil).

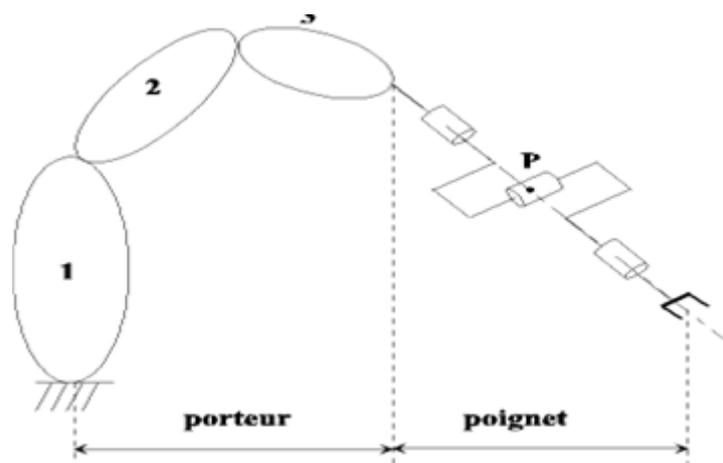


Figure 1.12 : le schéma d'un porteur.

1.16 Définitions et terminologie :

1.16.1 Redondance :

Les systèmes robotiques présentent généralement un degré de complexité qui impose de reconsidérer les techniques de modélisation et de commande de leur comportement.

L'une des dimensions particulières dans la complexité des systèmes est liée au nombre de leur degré de liberté qui peut induire une redondance/abondance vis-à-vis des tâches à réaliser.

Un robot est redondant lorsque le nombre de degrés de liberté de l'organe terminal est inférieur au nombre d'articulations motorisées.

1.16.2 Singularité :

Les mouvements d'un manipulateur sont observés dans l'espace cartésien. Ces mouvements sont donnés au manipulateur par les actionneurs, que l'on décrit dans l'espace articulaire. Donc, quand les actionneurs ont des vitesses $\dot{\theta}$. On doit pouvoir connaître les vitesses \dot{X} que cela impose à l'effecteur dans toutes les directions. La correspondance entre ces deux mesures doit donc être établie, en vue d'imposer aux actionneurs les vitesses articulaires requises pour amener l'effecteur à se déplacer de la façon prescrite dans l'espace cartésien. [11]

Cette relation prend la forme suivante, où A et B sont les matrices jacobiniennes du manipulateur

$$A\dot{X} = B\dot{\theta} \dots\dots\dots(1-1)$$

Dans certains cas, selon la position du manipulateur, cette relation n'amène plus de solution unique ou dégénère, c'est-à-dire qu'une entrée nulle amène une sortie non nulle ou vice-versa. Ces positions sont appelées les configurations singulières ou singularités du manipulateur. Les lieux de singularité regroupent toutes les configurations singulières d'un mécanisme.

Les singularités des manipulateurs parallèles se présentent sous trois formes distinctes. Les types rencontrés sont décrits par Gosselin et Angeles (1990). Essentiellement, les singularités sont classées de la manière suivante :

Type I : La matrice B est singulière. Ce type de singularité correspond généralement aux limites de l'espace atteignable. Dans cette situation, les vitesses cartésiennes peuvent être nulles pour des vitesses articulaires non nulles.

Type II : La matrice A est singulière. Dans ce cas, des vitesses cartésiennes non nulles peuvent être observées même si les actionneurs présentent des vitesses articulaires nulles.

Type III : Ce type de singularité peut se produire uniquement dans des cas particuliers, quand le manipulateur possède une architecture particulière.

Le type de la singularité qui sera étudié est celui attribué à la matrice jacobéenne A, le type II. En effet, celui-ci est le plus difficile à étudier et le plus susceptible de causer des problèmes mécaniques au manipulateur, en raison de l'augmentation des forces internes.

Pour les manipulateurs parallèles non-redondants, le calcul de ces lieux est connu.

Serfoui et Gosselin (1994 et 1995) ainsi que Mayer St-Onge et Gosselin(1997) en ont développé l'expression pour les manipulateurs à trois degrés de liberté. La représentation et l'expression des lieux de singularité des manipulateurs parallèles à six degrés de liberté ont été établies par Mayer St-Onge (1997).

1.16.3 La parcourabilité:

L'étude géométrique et cinématique des manipulateurs pleinement parallèles nous permet d'écrire une relation entre les entrées (q), c'est-à-dire les variables articulaires motorisées et les sorties (X), c'est-à-dire les configurations de la plate-forme mobile d'un manipulateur de la manière suivante : $F(X, q) = 0$. Par une dérivation au premier ordre de cette équation, nous obtenons l'équation suivante : $At + B$.

$q = 0$, où A et B sont deux matrices jacobéennes. [12]

L'analyse de la parcourabilité de l'espace de travail nous conduit à étudier les différentes trajectoires possibles dans l'espace opérationnel. Nous allons, dans les paragraphes suivants, définir géométriquement deux types de trajectoires et poser quelques hypothèses concernant leurs réalisations.

1.16.3.1 Une trajectoire discrète T_d : est une trajectoire définie par p configurations de l'espace opérationnel. Elle est définie à partir des éléments suivants :

- * Une origine X_0 .
- * Une extrémité X_E .

Un ensemble de configurations de la plate-forme mobile à atteindre

$$T_d = \{X_0, \dots, X_i, \dots, X_E\} / i \in [1, p], X_0 = X_1 \text{ et } X_E = X_p \dots \dots \dots (1.2)$$

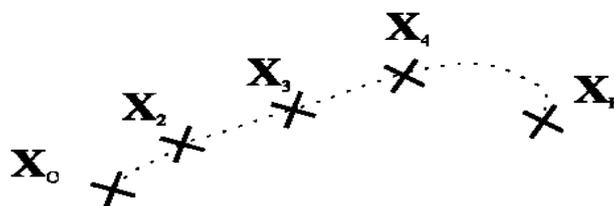


Figure1.13 : Trajectoire discrète. [12]

Entre chaque configuration X_i à atteindre, aucune trajectoire n'est définie.

Hypothèse 1 :

Lors d'une trajectoire discrète, le manipulateur ne peut pas franchir de singularité parallèle. En effet, la réalisation d'une trajectoire franchissant une singularité parallèle suppose l'emploi d'une commande dynamique car le manipulateur gagne un ou plusieurs degré(s) de liberté.

Pour ce faire, l'utilisation d'une telle commande suppose que le paramétrage utilisé assure la continuité du modèle géométrique direct, ce qui n'est pas toujours possible.

Notre étude étant purement cinématique, nous appliquerons donc l'hypothèse 1. On dira donc qu'une trajectoire discrète T_d est réalisable s'il existe un chemin continu de l'ensemble des configurations « $H\mathbb{Q}$ » qui ne rencontre pas de singularité parallèle telle que sa projection sur l'espace de travail passe par toutes les configurations de la plate-forme mobile définies par T_d .

Remarque :

Pour les manipulateurs possédant plusieurs solutions à leur modèle géométrique inverse, nous pouvons effectuer une trajectoire s'appelle «trajectoire de changement de mode de fonctionnement» entre chaque configuration. Ces trajectoires auront pour but d'éviter les configurations singulières présentes à l'intérieur de l'espace de travail. Pendant l'exécution de ces trajectoires, le manipulateur rencontre une singularité sérielle mais le contrôle de la plate-forme mobile n'est pas perdu.

1.16.3.2 Les trajectoires continues T_c :

Une trajectoire continue T_c est un arc paramétré de l'espace opérationnel E_{om} . Elle est définie à partir des éléments suivants.

* Une origine X_o .

* Une extrémité X_E .

$$\exists \lambda : [0,1] \rightarrow E_{O_m}, T_c = \lambda([0,1]) \dots\dots\dots(1.3)$$



Figure 1.14 : Trajectoire continue. [12]

Où λ est une fonction continue sur $[0,1]$ et dérivable par morceaux sur cet intervalle.

En d'autres termes, nous définissons une *trajectoire continue*, par une origine X_o et une extrémité X_E , et un arc paramétré T_c de longueur finie (l'origine et l'extrémité peuvent être confondues auquel cas la trajectoire est fermée).

Hypothèse 2 :

Lors d'une trajectoire continue, le manipulateur ne peut pas franchir ni une singularité parallèle ni une singularité sérielle.

En effet, nous souhaitons une continuité en position et orientation lors de la réalisation de trajectoires continues. Cette propriété ne peut pas être garantie pour n'importe quelle trajectoire franchissant une singularité parallèle ou sérielle.

Une trajectoire continue T_c sera dite réalisable s'il existe un chemin continu de l'ensemble des configurations « HQ » qui ne rencontre ni singularité parallèle et ni singularité sérielle et dont la projection sur l'espace de travail est la trajectoire continue T_c .

1.17 Conclusion :

Dans ce chapitre on a présenté d'une manière générale la terminologie utilisée en robotique ainsi que les différents constituants et structures mécaniques des robots industriels. Dans le chapitre suivant, on abordera d'une manière approfondie la modélisation mathématique des robots tenant compte du comportement cinématique et dynamique des bras manipulateurs.

CHAPITRE 2

MODELISATION DES BRAS MANIPULATEURS.

2-1 Introduction :

Pour développer une stratégie de commande performante d'un robot, il est impératif de connaître la cinématique et la dynamique du manipulateur considéré. Pour cela on est souvent amené à décrire les différentes relations mathématiques qui permettent de définir les mouvements de ce dernier dans l'espace.

Dans la pratique courante de robotique, la description du mouvement d'un robot manipulateur dans l'espace est réalisée en fonction du modèle géométrique, cinématique et dynamique.

Dans le présent chapitre, on présentera quelques définitions concernant ces modèles ainsi que la façon de leurs obtentions.

2-2 Structures en chaînes ouvertes arborescentes (multiple open-tree structure) :

- Certains corps sont connectés à plus de deux autres corps.
- Possibilité de numéroté les corps de manière séquentielle.
- Utilisation du concept de graphe.
- Possibilité d'avoir plusieurs effecteurs (comme le corps humain).
- Rare en robotique [13].

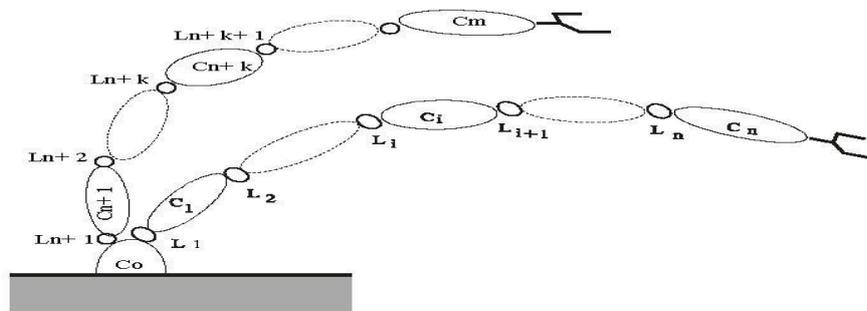


Figure2.1 : Une chaîne arborescente. [13].

2-3 Structures en chaîne articulaire ouverte simple (simple open-tree structure) :

- Chaque corps est attaché à seulement 2 voisins.
- Structure purement séquentielle.
- Architecture habituelle du robot manipulateur.

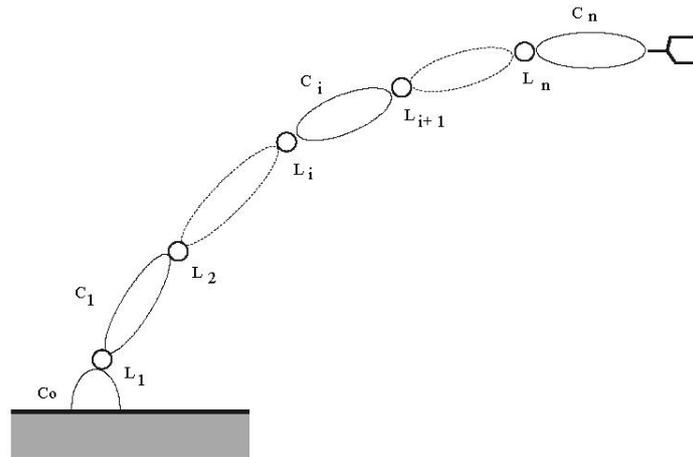


Figure 2.2 : Une chaîne ouverte simple. [13].

2-4 Structure complexe (multiply connected structure) :

- Présence de boucles fermées.
- Modélisation basée sur les structures simplement connectées en plus des contraintes liées à la fermeture des boucles.
- En robotique industrielle: grande raideur, grande précision, mais faible mobilité.

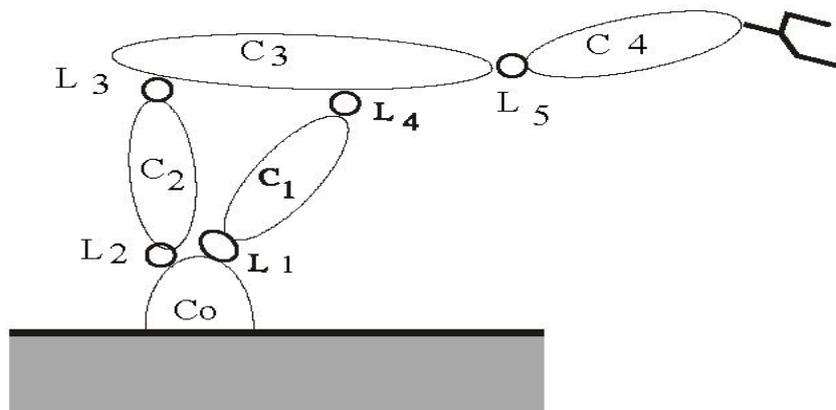


Figure 2.3 : Une chaîne fermée. [13].

2-5 Qu'est-ce qu'un système multi corps ?

Un vélo, une voiture, un camion, un hélicoptère, une grue, un moteur, un robot, un satellite, le corps humain : tous sont des exemples de système multi corps.

On les appelle également des systèmes mécaniques articulés.

Un système multi corps est un système composé de plusieurs parties rigides ou flexibles, appelées des corps, qui sont reliés entre eux par des articulations.

On distingue souvent ces systèmes selon leur structure. Celle-ci peut être de type arborescente ou ouverte, lorsque sa topologie peut être représentée par un arbre, ou bien de type bouclée ou

fermée, lorsque le graphe topologique contient des cycles. La figure1 illustre les deux types de topologie, ouverte ou fermée.[14]

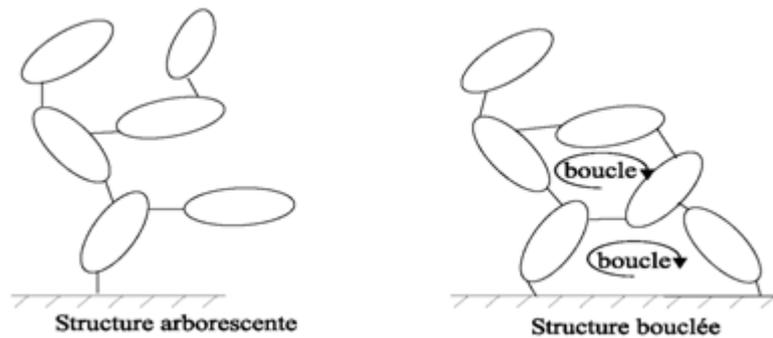


Figure 2.4 : Structure arborescente ou bouclée.[14]

Actuellement de nombreuses applications impliquent l'étude de tels systèmes. Ceux-ci sont modélisés et analysés à l'aide d'ordinateurs pour en comprendre et pour en améliorer le fonctionnement ou même pour mieux les concevoir.

Dans le domaine de l'automobile et des transports, on recherche continuellement à améliorer les performances, le confort et la sécurité des voitures, des camions et des trains. Dans le domaine aéronautique, les dispositifs de contrôle des avions et des hélicoptères sont constitués de mécanismes complexes dont le comportement dynamique est étudié pour en garantir le fonctionnement correct et performant.

Dans l'espace, des bras robotiques sont utilisés par les astronautes pour extraire les satellites de la navette ou pour l'assemblage et la construction de la station spatiale internationale. A quelques centaines de millions de kilomètres de la terre, les robots mobiles d'exploration Séjourner, Spirit et Opportunité, sont tour à tour partis à la découverte du sol martien.

Quel que soit le domaine d'application, l'étude de ces systèmes repose entre autres choses sur l'utilisation d'équations qui en décrivent le comportement.

Nous avons tous été confrontés à des équations au cours de nos études :

$$y = ax + b , \quad f = ma , \quad U = RI , \quad E = mc^2 \dots\dots\dots (2.1)$$

Quelques soient les domaines desquels ces équations proviennent, il s'agit toujours des expressions écrites à l'aide de symboles qui représentent des opérations arithmétiques + ; - ; / ou autres ainsi des variables x,y..... ou des paramètres a,b.....

Nous nous souvenons également que cela devenait fastidieux et difficile lorsque le problème était compliqué et qu'il n'y avait plus, une seule mais, plusieurs équations à traiter. Heureusement, pour ces situations difficiles qui correspondent bien souvent aux applications réelles, nous disposons actuellement des ordinateurs. Ces machines à calculer extraordinaires

sont bien utiles aux ingénieurs pour les aider à évaluer et à résoudre les équations lorsqu'elles sont nombreuses et compliquées.

Mais pour qu'ils exécutent un travail de calcul, les ordinateurs ont besoin d'instructions, de programmes. Naturellement, il y a plusieurs possibilités pour donner ces instructions à un ordinateur. L'une de ces possibilités est précisément d'écrire ces instructions sous la même forme que les équations que nous écrivons à la main. Dans ce cas, il est donc nécessaire d'écrire ces équations, ce qui constitue un travail fastidieux, comme nous l'avons déjà dit. L'idée intéressante est alors d'utiliser l'ordinateur pour écrire les équations. Cette première phase d'écriture des équations par l'ordinateur, sous la même forme symbolique que nous connaissons, s'appelle la génération symbolique.

Une fois ces équations générées, l'ordinateur va ensuite pouvoir les utiliser pour effectuer les calculs requis, par exemple pour l'étude du mouvement d'un système multicorps, puisque c'est le domaine qui nous intéresse. Lorsque le système devient complexe ou de grande taille, le nombre d'équations augmente, et le temps nécessaire pour exécuter les calculs également. Une solution pour réduire le temps de calcul consiste alors à utiliser plusieurs ordinateurs ou bien, un ordinateur qui contient plusieurs processeurs.

Un nouveau problème surgit alors : celui de la répartition du travail. Certaines règles classiques du travail en groupe sont d'application ici : l'efficacité de l'équipe sera meilleure si chacun peut effectuer sa partie du travail sans avoir besoin des autres, ou alors avec un minimum d'interactions. En ce qui nous concerne, il s'agit donc de séparer l'ensemble des équations à calculer et de constituer des sous-ensembles suffisamment indépendants afin qu'ils puissent être exécutés efficacement par les différents processeurs utilisés. Cette phase de division et de répartition des équations s'appellent la parallélisation.

Après ces quelques explications qui nous permettent de mieux comprendre le titre de ce travail, nous vous proposons de découvrir le domaine dans lequel ce travail s'inscrit, à savoir la modélisation et l'étude de la dynamique des systèmes multi corps.

2-5-1 Modélisation des systèmes multicorps :

Dans toute démarche de modélisation de système, l'objectif de l'ingénieur est d'obtenir des équations qui représentent le comportement du système étudié.

Afin d'écrire ces équations, il est tout d'abord nécessaire de se donner un jeu de variables qui permettent de décrire la configuration, ou l'état, du système : ce sont les coordonnées généralisées. Différents choix de coordonnées sont possibles. Selon les coordonnées choisies, l'ingénieur peut alors utiliser différentes méthodes, ou formalismes pour écrire les équations permettant d'étudier le comportement du système. Dans le cas des systèmes multicorps, ces

équations concernent les mouvements des corps qui sont soumis aux lois de la mécanique rationnelle.

On distingue deux problèmes classiques dans l'analyse dynamique des systèmes multicorps :

- 1- Le problème dynamique inverse consiste à déterminer les efforts Q à transmettre par les articulations du système afin que celui-ci évolue selon une trajectoire prescrite par les valeurs q des coordonnées généralisées ainsi que par les valeurs de leurs dérivées premières \dot{q} et secondes \ddot{q}

On peut l'exprimer par l'équation globale suivante :

$$Q = (q, \dot{q}, \ddot{q}) \dots\dots\dots(2.2)$$

- 2- Le problème dynamique direct consiste à déterminer les accélérations généralisées \ddot{q} lorsque le système est soumis aux forces articulaires Q dans une configuration connue en termes de valeurs des coordonnées généralisées

q et de leur dérivée première \dot{q}

On peut l'exprimer par l'équation globale suivante

$$\ddot{q} = \ddot{q}(q, \dot{q}, Q) \dots\dots\dots(2.3)$$

La dynamique inverse concerne le contrôle de robot et la dynamique directe concerne la simulation des systèmes multicorps, quels qu'ils soient.

Les accélérations généralisées \ddot{q} et les forces articulaires Q dépendent linéairement les unes des autres. Cette dépendance est exprimée par la relation suivante qui correspond aux équations du mouvement du système :

$$M(q)\ddot{q} + c(q, \dot{q}, F_{ext}, L_{ext}, g) = Q \dots\dots\dots (2.4)$$

$M(q)$: représente la matrice de masse généralisée du système.

$c(q, \dot{q}, F_{ext}, L_{ext}, g)$; regroupe les termes gyroscopiques de Coriolis et les forces couples extérieures appliquées sur le corps ainsi que la gravité.

Dans la plupart des cas, les équations du mouvement sont accompagnées d'un ensemble d'équations supplémentaires qui expriment des relations entre les coordonnées généralisées. Ce sont des équations de contraintes. Le nombre et la nature des équations de contraintes dépendent essentiellement du choix des coordonnées, de la structure du système et des conditions auxquelles le système est soumis. Ces équations de contraintes sont algébriques et généralement non linéaires. On les écrit souvent sous la forme implicite suivante :

$$h(q) = 0 \dots\dots\dots (2.5)$$

L'ensemble des équations nécessaires à l'étude du mouvement d'un système multicorps soumis à des contraintes est alors le suivant :

$$M(q)\ddot{q} + c(q, \dot{q}, F_{ext}, L_{ext}, g) = Q + J(q)^T \lambda \dots\dots\dots (2.6)$$

$$h(q) = 0 \dots\dots\dots (2.7)$$

$J(q) = \frac{\partial \dot{h}(q)}{\partial \dot{q}}$: est la matrice jacobienne associée aux contraintes.

λ : Le vecteur des multiplicateurs de Lagrange.

La présence simultanée d'équations différentielles et algébriques pose une difficulté. En effet leur intégration numérique nécessite l'utilisation de méthodes particulières. Il est toutefois possible de les transformer en équations purement différentielles. En pratique les dérivées premières et secondes des équations de contraintes jouent un rôle important pour l'élaboration de ces formulations purement différentielles.

$$\dot{h}(q, \dot{q}) = J(q)\dot{q} \dots\dots\dots(2.8)$$

$$\ddot{h}(q, \dot{q}, \ddot{q}) = J(q)\ddot{q} + \dot{J}\dot{q} \dots\dots\dots(2.9)$$

Nous présentons dans les paragraphes qui suivent les différentes options les plus classiques en ce qui concerne :

1. le choix des coordonnées pour représenter la configuration d'un système multi corps,
2. les formalismes utilisés pour obtenir les équations du mouvement du système,
3. les méthodes de traitements des équations de contraintes,
4. les différentes formulations des équations du mouvement en vue de leur utilisation pour l'analyse numérique du comportement du système.

2-5-2 Coordonnées :

Le choix des coordonnées a une influence directe sur le type de formalisme utilisable et le nombre d'équations obtenues pour la modélisation mathématique d'un système multicorps. Il existe différents types de coordonnées dont les plus connues sont les suivantes.

2-5-3 Coordonnées absolues :

Les coordonnées absolues expriment les positions et orientations des corps par rapport à un repère de référence unique. Six coordonnées, trois pour les translations et trois pour les rotations sont utilisées pour décrire la configuration individuelle de chaque corps du système. Le nombre de coordonnées est alors bien souvent très supérieur au nombre de degrés de liberté du système.

Elles ont l'avantage de rendre l'écriture des équations du mouvement des corps très directe et très facile. De plus la matrice de masse globale du système est constante et (bloc) diagonale,

et les équations du mouvement des corps sont très simples et peuvent être calculées indépendamment pour chaque corps.

En revanche, les restrictions imposées par les articulations au niveau des mouvements relatifs des corps les uns par rapport aux autres, nécessitent l'écriture d'équations de contraintes relativement complexes. De une à cinq équations de contraintes sont nécessaires selon le type d'articulation, ce qui compense largement la simplicité des équations du mouvement. Ainsi, que le système soit arborescent ou qu'il contient des boucles cinématiques, on obtient systématiquement un jeu d'équations différentielles et algébriques. Le nombre d'équations est alors bien plus grand que dans le cas de l'utilisation de coordonnées relatives.

2-5-4 Formalismes :

Dans le cas de l'utilisation de coordonnées absolues, l'écriture des équations du mouvement du système multicorps ne repose pas à proprement parler sur l'utilisation d'un formalisme particulier. Les équations de Newton (2.10) et d'Euler (2.11) peuvent être écrites directement pour chaque corps.

$$F = m\ddot{R}^G \dots\dots\dots (2.10)$$

$$I^G \dot{\omega} + \ddot{\omega} I^G \omega = L^G \dots\dots\dots (2.11)$$

\ddot{R}^G : est l'accélération absolue du centre de masse du corps

m : est la masse du corps

F : est la résultante des forces appliquées sur le corps

I^G : est le tenseur d'inertie du corps, par rapport à son centre de masse

ω : est le vecteur vitesse angulaire du corps

L^G : est la résultante des moments des forces appliquées sur le corps, par rapport à son centre de masse.

2-6 Structure géométrique :

Parmi les diverses méthodes utilisées pour déterminer la position et l'orientation de l'organe terminal par rapport au repère de référence, la plus répandue est celle de Denavit-hartenberg [15]. Un robot avec n articulations aura n+1 liens, la numérotation des liens commencent de 0 pour l'immobile fondée (base Link) et augmente séquentiellement à n pour l'organe terminal.

La numérotation des articulations commence de 1 pour l'articulation reliant le premier lien mobile vers le lien de base et augmente séquentiellement à n.

Le lien i est connecté avec son lien inférieur (i-1) à son extrémité proximale par l'articulation i et est relié à son lien supérieur (i + 1) à son extrémité distale par l'articulation i + 1.

Le schéma suivant montre les liens (i-1), i, (i+1) d'un robot, avec les articulations (i-1), (i+1). Chaque articulation est indiquée par son axe qui peut être de rotation ou de translation. Pour rapporter l'information cinématique des composants de robot, nous attachons rigidement un axe B_i de coordonnées locales à chaque lien (i) à l'articulation i+1 basée sur la méthode standard suivante, connue sous le nom de méthode (DH) de Denavit-Hartenberg.

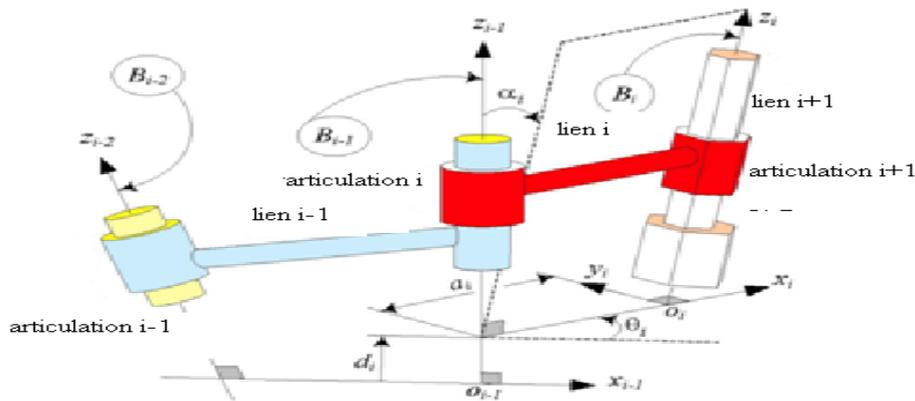


Figure 2.5 : Les paramètres de (D-H) ($a_i, \alpha_i, d_i, \theta_i$) : défini pour l'articulation i et lien i. [15]

a_i : (Link length) est la distance entre z_{i-1} et z_i le long de l'axe x_i . a_i est la longueur cinématique du lien i.

α_i : (Link twist) est la rotation exigée de l'axe z_{i-1} autour x_i pour devenir parallèle a z_i .

d_i : (Joint distance) est la distance entre x_{i-1} et x_i le long de l'axe z_{i-1} il est aussi appelé (Link offset).

θ_i : (Joint angle) est la rotation exigée de l'axe x_{i-1} autour de l'axe z_{i-1} pour devenir parallèle a l'axe x_i .

Méthode de (D-H) destinée à systématiser la modélisation de n'importe quel type de robot.

Ses principaux avantages sont :

- Simplification maximale du modèle géométrique
- Etablissement d'une norme reconnue par tous.

2-6-1 Modèle géométrique :

2-6-1-1 Modèle géométrique direct :

Le modèle géométrique direct permet de déterminer la position et l'orientation de l'organe terminal du manipulateur par rapport à un repère de référence en fonction des variables articulaires, le modèle s'écrit :

$$X = f(q) \dots\dots\dots(2-12)$$

$q = (q_1, q_2, \dots, q_n)^T \in R^n$: Vecteur des variables articulaires

$X = (x_1, x_2, \dots, x_n)^T \in R^m$: Vecteur des variables opérationnelles, $m \leq n$.

Par exemple, si le manipulateur se déplace dans l'espace on pose $m=6$ (3 coordonnées pour la position et 3 coordonnées pour la rotation). S'il se déplace dans un plan on pose $m=2$ et si en plus il est concerné par la rotation on pose $m=3$.

La position de l'organe terminal peut être définie par des coordonnées cartésiennes, cylindriques ou sphériques. Le choix d'une structure particulière est guidé par les caractéristiques du robot, ainsi que par celle de la tâche à réaliser. [16]

2-6-1-2 Modèle géométrique inverse :

Le modèle géométrique inverse permet de déterminer le vecteur des variables articulaires à partir du vecteur de coordonnées opérationnelles, le modèle s'écrit :

$$q = f^{-1}(x) \dots\dots\dots(2-13)$$

Parmi les méthodes utilisées pour déterminer le modèle géométrique inverse on cite :

- Les méthodes géométriques : permettent de déterminer le vecteur q par utilisation des transformations géométriques en prenant avantage de la structure particulière du manipulateur considéré.
- Les méthodes algébriques : permettent de déterminer le vecteur q en effectuant des transformations algébriques sur l'équation (2-13) parmi les méthodes utilisées on cite la méthode de Paul qui consiste à multiplier successivement les deux membres du modèle géométrique direct par les matrices homogènes ; T_i^{i-1} avec $(i=1\dots n-1)$ permettant ainsi d'isoler et d'identifier les variables articulaires l'une après l'autre.

Lors de la résolution du problème géométrique inverse on rencontre pratiquement les situations suivantes :

- Solutions en nombre fini, lorsqu'elles peuvent être calculées sans ambiguïté.
- Aucune solution possible lorsque la position désirée ne peut être atteinte par le manipulateur.
- Plusieurs solutions possibles lorsque le manipulateur est redondant ou lorsqu'il passe par une configuration singulière.

2-6-2 Modèle cinématique :

2-6-2-1 Modèle cinématique direct :

Le modèle cinématique direct permet de déterminer la vitesse de l'organe terminal dans l'espace opérationnel en fonction de la vitesse des variables articulaires. Le modèle est décrit par l'équation :

$$\dot{X} = J(q)\dot{q} \dots\dots\dots(2-14)$$

$J(q) \in R^{m \times n}$ est la matrice jacobéenne.

L'une des méthodes utilisées pour le calcul de la matrice jacobéenne est la dérivation du modèle géométrique direct.

$$J(q) = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_n} \\ \vdots & \vdots \\ \frac{\partial f_m}{\partial q_1} & \frac{\partial f_m}{\partial q_n} \end{bmatrix} \dots\dots\dots (2.15)$$

2-6-2-2 Modèle cinématique inverse :

Le modèle cinématique inverse permet de déterminer la vitesse des variables articulaires en fonction de la vitesse des variables opérationnelles. Pour les manipulateurs non redondants (n=m), le modèle s'écrit : [16]

$$\dot{q} = J^{-1}(q)\dot{X} \dots\dots\dots(2.16)$$

La solution de l'équation (2.16) existe si J est de rang plein, cela est valable tant que le manipulateur ne passe pas par une configuration singulière. Pour les manipulateurs redondants, le modèle cinématique inverse admet plusieurs solutions possibles. Le choix d'une solution parmi plusieurs est guidé par l'optimisation d'une fonction objective.

2-6-3 Modèle dynamique :

2-6-3-1 Modèle dynamique inverse :

Le modèle dynamique inverse exprime les couples exercés par les actionneurs en fonction des positions, vitesses et accélérations des articulations, les formalismes les plus utilisés pour les calculs du modèle dynamique inverse sont :

- Formalisme de Lagrange
- Formalisme de newton-Euler.

2-6-3-1-1 Formalisme de Lagrange :

Les équations de Lagrange opèrent à partir de l'énergie cinétique et l'énergie potentielle d'un système, le lagrangien s'écrit sous la forme :

$$L = K - V \dots\dots\dots(2.17)$$

Où :

K Et V représentent respectivement les énergies cinétique et potentielle du système.

Les équations de Lagrange sont définies par :

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \right) - \left(\frac{\partial L}{\partial x} \right) = \tau_i \dots\dots\dots (2.18)$$

Pour le calcul du modèle dynamique d'une chaîne cinématique (figure 2.6), on définit les paramètres suivants pour chaque corps i de la chaîne cinématique :

m_i : Masse du corps i de la chaîne cinématique.

${}^i r = [{}^i r_x, {}^i r_y, {}^i r_z]^T$ Vecteur du centre d'inertie du corps i exprimé dans le repère R_i .

$m^i r = [m^i r_x, m^i r_y, m^i r_z]^T$: Premier moment du corps i autour de o_i exprimé dans R_i

$${}^i I_i = \begin{bmatrix} {}^i I_{ixx} & -{}^i I_{ixy} & -{}^i I_{ixz} \\ -{}^i I_{ixy} & {}^i I_{iyy} & -{}^i I_{iyz} \\ -{}^i I_{ixz} & -{}^i I_{iyz} & {}^i I_{izz} \end{bmatrix} : \text{Tenseur d'inertie du corps } i \text{ en } o_i, \text{ exprimé dans } R_i$$

Où :

$${}^i I_{ixx} = \int ({}^i r_y^2 + {}^i r_z^2) dm_i : \text{Moment d'inertie.}$$

$${}^i I_{ixy} = \int {}^i r_x {}^i r_y dm_i : \text{Produit d'inertie.}$$

$${}^i S_i = [{}^i S_{ix}, {}^i S_{iy}, {}^i S_{iz}]^T : \text{Position du centre de masse du corps } i \text{ exprimée en } R_i$$

$$H_i = \int {}^i r^i r^T dm_i : \text{Matrice des pseudo-inerties du corps } i.$$

$$H_i = \begin{bmatrix} \frac{-{}^i I_{ixx} + {}^i I_{iyy} + {}^i I_{izz}}{2} & I_{ixy} & I_{ixz} & m_i S_{ix} \\ I_{ixy} & \frac{{}^i I_{ixx} - {}^i I_{iyy} + {}^i I_{izz}}{2} & I_{iyz} & m_i S_{iy} \\ I_{ixz} & I_{iyz} & \frac{{}^i I_{ixx} + {}^i I_{iyy} - {}^i I_{izz}}{2} & m_i S_{iz} \\ m_i S_{ix} & m_i S_{iy} & m_i S_{iz} & m_i \end{bmatrix} \dots\dots\dots(2.19)$$

L'énergie cinétique du corps i est donnée par :

$$K_i = \frac{1}{2} \int {}^0 \dot{r}^T {}^0 \dot{r} dm \dots\dots\dots(2.20)$$

$$K_i = \frac{1}{2} \sum_{j=1}^i \sum_{k=1}^i tr \left(\frac{\partial {}^0 T_i}{\partial q_i} H_i \frac{\partial {}^0 T_i^T}{\partial q_k} \right) \dot{q}_j \dot{q}_k \dots\dots\dots(2.21)$$

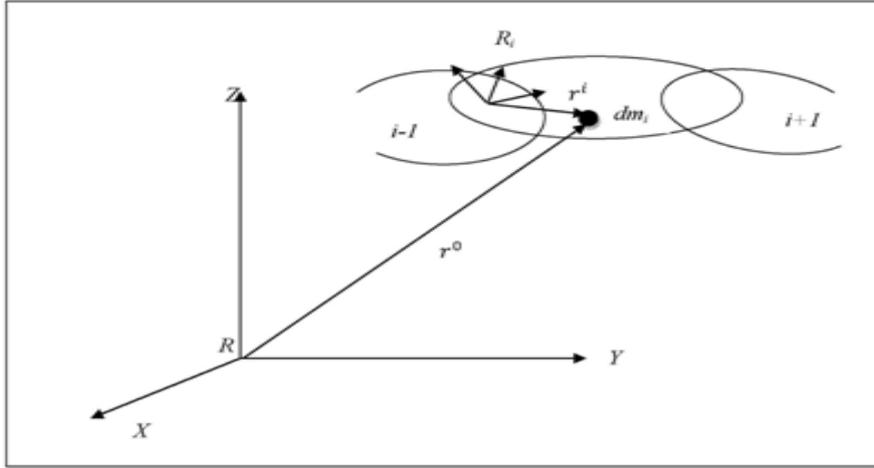


Figure 2.6 : Représentation d'un point r^i de la chaîne cinématique. [16]

tr : signifie trace de la matrice ; elle est égale à la somme des éléments de la diagonale.

L'énergie de potentiel du corps i est donnée par :

$$V_i = -m_i g^T T_i^i S_i \dots\dots\dots(2.22)$$

$g = [g_x, g_y, g_z, 0]^T$: Vecteur des accélérations de la gravité exprimé dans R_0

Le Lagrangien pour l'ensemble des corps du manipulateur est :

$$L = \sum_{i=1}^n (K_i - V_i) \dots\dots\dots(2.23)$$

$$\tau = \sum_{k=1}^n \sum_{j=1}^k tr \left(\frac{\partial^0 T_k}{\partial q_j} H_k \frac{\partial^0 T_k^T}{\partial q_i} \right) \ddot{q}_j + \sum_{k=i}^n \sum_{j=1}^k \sum_{m=1}^k tr \left(\frac{\partial^{20} T_k}{\partial q_j \partial q_m} H_k \frac{\partial^0 T_k^T}{\partial q_i} \right) \dot{q}_j \dot{q}_m - \sum_{j=1}^n m_j g^T \frac{\partial^0 T_j^j}{\partial q_i} S_j \dots\dots\dots (2.24)$$

Le modèle obtenu peut être mis sous la forme matricielle suivante :

$$\tau = M(q)\ddot{q} + N(q, \dot{q})\dot{q} + G(q) \dots\dots\dots(2.25)$$

Ou :

$$M_{ij} = \sum_{k=\max(i,j)}^n tr \left(\frac{\partial^0 T_k}{\partial q_j} H_k \frac{\partial^0 T_k^T}{\partial q_i} \right) \dots\dots\dots (2.26)$$

$$N(q, \dot{q}) = \sum_{k=\max(i,j,m)}^n \sum_{j=1}^k \sum_{m=1}^k tr \left(\frac{\partial^{20} T_k}{\partial q_j \partial q_m} H_k \frac{\partial^0 T_k^T}{\partial q_i} \right) \dot{q}_j \dot{q}_m \dots\dots\dots (2.27)$$

$$G(q) = \sum_{j=1}^n m_j g^T \frac{\partial^0 T_j^j}{\partial q_i} S_j \dots\dots\dots (2.28)$$

$M(q)$: Matrice d'inertie du manipulateur, symétrique et régulière.

$N(q, \dot{q})\dot{q}$: Vecteur des termes de Coriolis et centrifuges.

$G(q)$: Vecteur des actions de la gravité.

2-6-3-1-2 Formalisme de Newton-Euler :

Le formalisme de Newton-Euler est basé sur les théorèmes généraux de la mécanique.

Il ne nécessite pas le calcul extrinsèque des coefficients dynamiques du robot. Le modèle est obtenu par double récurrence. [17]

2-6-3-1-3 Equation de NEWTON :

Le schéma suivant montre un corps rigide dont le centre de masse accélère avec une accélération \dot{v}_c . Dans une telle situation, la force, F, agissant sur le centre de masse qui cause cette accélération est donnée par l'équation de NEWTON [17]

$$F = m\dot{v}_c \dots\dots\dots(2.29)$$

Où m est la masse totale du corps

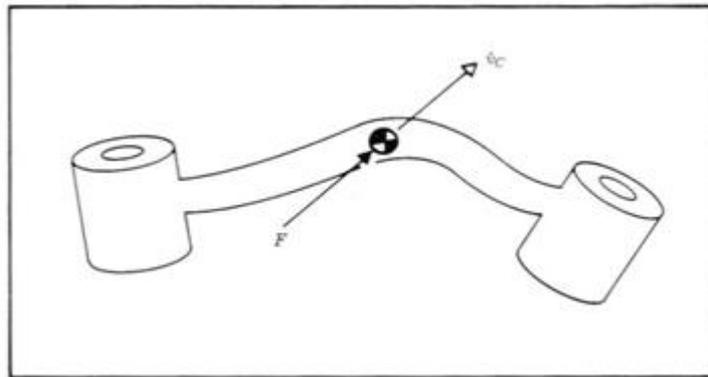


Figure 2.7 : La force exercée sur le lien i. [17]

2-6-3-1-4 L'équation d'EULER :

Le schéma suivant montre un corps rigide tournant avec une vitesse angulaire ω et avec une accélération angulaire $\dot{\omega}$. Dans une telle situation le moment N qui doit agir sur le corps pour causer ce mouvement est indiqué par l'équation d'Euler ;

$$N = {}^c I \dot{\omega} + \omega {}^c I \omega \dots\dots\dots(2.30)$$

${}^c I$: est le tenseur d'inertie du corps dont l'origine est située au centre de la masse.

Formulation dynamique itérative de Newton-Euler :

Le problème est de trouver les couples qui correspondent à une trajectoire donnée d'un manipulateur. Nous supposons connues la position, la vitesse, et l'accélération des articulations $(\theta, \dot{\theta}, \ddot{\theta})$. Cette hypothèse faite et avec la connaissance d'information de cinématique et de distribution de masse du robot, nous pouvons calculer les couples communs pris pour causer ce mouvement. L'algorithme présenté est basé sur la méthode éditée par Luh, Marcheur, et Paul. [15]

2-6-3-1-5 Récurrence avant pour calculer les vitesses et les accélérations :

Il est nécessaire de calculer les forces d'inertie agissant sur les liens, la vitesse de rotation et l'accélération linéaire et de rotation du centre de la masse de chaque lien du manipulateur à n'importe quel instant donné.

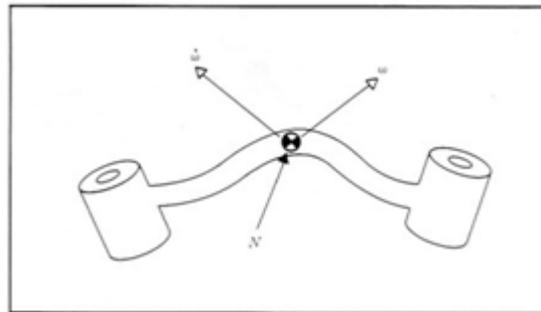


Figure 2.8 : le couple exercé sur le lien i. [17]

Ces calculs seront donnés dans une nature itérative commençant par le lien 1 et en déplaçant successivement lien par lien vers l'extérieur du lien n

La propagation de la vitesse de rotation d'un lien vers un autre est donnée par la relation suivante :

Pour (i+1) rotation :

$${}^{i+1}\dot{\omega}_{i+1} = {}^{i+1}\mathbb{R}^i \dot{\omega}_i + \dot{\theta}_{i+1} {}^{i+1}Z_{i+1} \dots \dots \dots (2.31)$$

Nous obtenons l'équation de transformation de l'accélération angulaire d'un lien à l'autre ;

$${}^{i+1}\dot{\omega}_{i+1} = {}^{i+1}\mathbb{R}^i \dot{\omega}_i + {}^{i+1}\mathbb{R}^i \omega_i \times \dot{\theta}_{i+1} {}^{i+1}Z_{i+1} + \dot{\theta}_{i+1} {}^{i+1}Z_{i+1} \dots \dots \dots (2.32)$$

Quand le lien (i+1) est prismatique :

$${}^{i+1}\dot{\omega}_{i+1} = {}^{i+1}\mathbb{R}^i \dot{\omega}_i \dots \dots \dots (2.33)$$

Accélération linéaire de chaque lien est obtenue par la relation suivante ;

$${}^{i+1}\dot{v}_{i+1} = {}^{i+1}\mathbb{R}^i [\dot{\omega}_i \times {}^i P_{i+1} + \omega_i \times ({}^i \omega_i \times {}^i P_{i+1}) + {}^i \dot{v}_i] \dots \dots \dots (2.34)$$

Pour le lien (i+1) est prismatique :

$${}^{i+1}\dot{v}_{i+1} = {}^{i+1}\mathbb{R}^i [\dot{\omega}_i \times {}^i P_{i+1} + \omega_i \times ({}^i \omega_i \times {}^i P_{i+1}) + {}^i \dot{v}_i] + 2{}^{i+1}\omega_{i+1} \times \dot{d}_{i+1} {}^{i+1}Z_{i+1} + \dot{d}_{i+1} {}^{i+1}Z_{i+1} \dots \dots \dots (2.35)$$

L'accélération linéaire du centre de masse pour chaque lien est :

$${}^i v_{Ci} = {}^i \dot{\omega}_i \times {}^i P_{Ci} + \omega_i \times ({}^i \omega_i \times {}^i P_{Ci}) + {}^i \dot{v}_i \dots \dots \dots (2.36)$$

Notez que l'application des équations de liaison est spécialement simple quand :

$${}^0 \omega_0 = {}^0 \dot{\omega}_0 = 0 \dots \dots \dots (2.37)$$

La force et le couple agissant sur un lien :

Après avoir calculé les accélérations linéaires et angulaires du centre de masse de chaque lien, on peut appliquer les équations de Newton-Euler pour calculer la force d'inertie et le couple agissant au centre de masse de chaque lien.

Récurrence arrière pour calculer les forces et les couples :

Après avoir calculé les forces et couples agissant sur chaque lien, maintenant il reste à calculer les couples de conjoints qui se traduiront par ces forces et les moments appliqués à chaque lien.

f_i : La force exercée sur le lien i par le lien i-1.

n_i : Le couple exercé sur le lien i par le lien i-1.

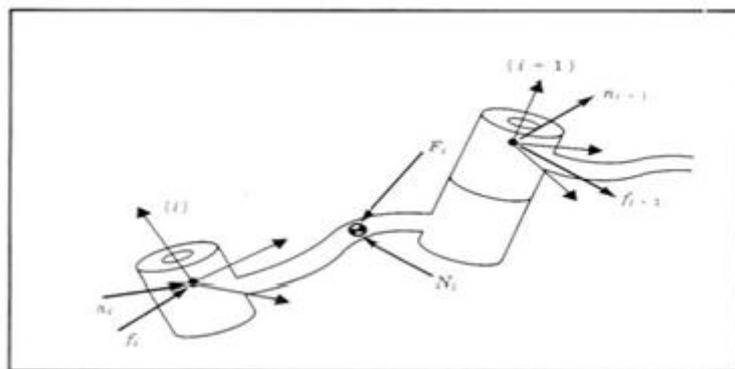


Figure 2.9 : les forces agissant sur le lien i. [17]

En additionnant les forces agissant sur le lien i, nous arrivons à une relation d'équilibre des forces,

$${}^i F_i = {}^i f_i - {}_i^{i+1} \mathbb{R}^{i+1} f_i \dots \dots \dots (2.38)$$

En additionnant les couples sur le centre de masse, nous arrivons à l'équation d'équilibre de couples:

$${}^i N_i = {}^i n_i - {}^i n_{i+1} + ({}^i P_{C_i}) \times {}^i f_i - ({}^i P_{i+1} - {}^i P_{C_i}) \times {}^i f_{i+1} \dots \dots \dots (2.39)$$

En utilisant le résultat de l'équilibre des forces et en ajoutant une matrice de rotation, on peut écrire l'équation d'équilibre de couple sous la forme suivante :

$${}^i N_i = {}^i n_i - {}_i^{i+1} \mathbb{R}^{i+1} n_{i+1} - {}^i P_{C_i} \times {}^i F_i - {}^i P_{i+1} \times {}_i^{i+1} \mathbb{R}^{i+1} f_{i+1} \dots \dots \dots (2.40)$$

Enfin, on peut réorganiser les équations de force et de couple sous la forme suivante ;

$${}^i f_i = {}_i^{i+1} \mathbb{R}^{i+1} f_{i+1} + {}^i F_i \dots \dots \dots (2.41)$$

$${}^i n_i = {}^i N_i + {}_i^{i+1} \mathbb{R}^{i+1} n_{i+1} + {}^i P_{C_i} \times {}^i F_i + {}^i P_{i+1} \times {}_i^{i+1} \mathbb{R}^{i+1} f_{i+1} \dots \dots \dots (2.42)$$

Comme dans le cas statique, les couples requis communs se trouvent en prenant la composante Z du couple appliqué par un lien sur son voisin :

$$\tau_i = {}^i n_i^T Z_i \dots \dots \dots (2.43)$$

Si on a (i+1) est prismatique on utilise :

$$\tau_i = {}^i f_i^T Z_i \dots\dots\dots(2.44)$$

2-6-3-1-6 L'algorithme itératif de dynamique de Newton-Euler :

L'algorithme complet pour calculer des couples communs du mouvement des joints se compose de deux parts. D'abord des vitesses de lien et les accélérations sont itérativement calculées du lien 1 dehors pour lien n et les équations de Newton-Euler sont appliquées à chaque lien. Second, forces et des couples de l'interaction et les couples de déclencheur de joint sont calculés périodiquement du lien n de nouveau au lien 1. Les équations sont récapitulées au-dessous de pour le cas de tous les liens de rotation.

Récurrance avant :

i=1.....5

$${}^{i+1} \omega_{i+1} = {}^i \mathbb{R}^i \omega_i + \dot{\theta}_{i+1} {}^{i+1} Z_{i+1} \dots\dots\dots(2.45)$$

$${}^{i+1} \dot{\omega}_{i+1} = {}^i \mathbb{R}^i \dot{\omega}_i + {}^i \mathbb{R}^i \omega_i \times \dot{\theta}_{i+1} {}^{i+1} Z_{i+1} + \dot{\theta}_{i+1} {}^{i+1} Z_{i+1} \dots\dots\dots(2.46)$$

$${}^{i+1} \dot{v}_{i+1} = {}^i \mathbb{R}^i [\dot{\omega}_i \times {}^i P_{i+1} + \omega_i \times ({}^i \omega_i \times {}^i P_{i+1}) + \dot{v}_i] + 2{}^{i+1} \omega_{i+1} \times \dot{d}_{i+1} {}^{i+1} Z_{i+1} + \ddot{d}_{i+1} {}^{i+1} Z_{i+1} \dots\dots\dots(2.47)$$

$${}^{i+1} \dot{v}_{Ci+1} = {}^{i+1} \dot{\omega}_{i+1} \times {}^{i+1} P_{Ci+1} + {}^{i+1} \omega_{i+1} \times ({}^{i+1} \omega_{i+1} \times {}^{i+1} P_{Ci+1}) + {}^{i+1} \dot{v}_{i+1} \dots\dots\dots(2.48)$$

$${}^{i+1} F_{i+1} = m_{i+1} {}^{i+1} \dot{v}_{Ci+1} \dots\dots\dots(2.49)$$

$${}^{i+1} N_{i+1} = {}^{Ci+1} I_{i+1} {}^{i+1} \dot{\omega}_{i+1} + {}^{i+1} \dot{\omega}_{i+1} \times {}^{Ci+1} I_{i+1} {}^{i+1} \omega_{i+1} \dots\dots\dots(2.50)$$

Récurrance arrière :

i=6.....1

$${}^i f_i = {}^i \mathbb{R}^{i+1} f_{i+1} + {}^i F_i \dots\dots\dots(2.51)$$

$${}^i n_i = {}^i N_i + {}^i \mathbb{R}^{i+1} n_{i+1} + {}^i P_{Ci} \times {}^i F_i + {}^i P_{i+1} \times {}^i \mathbb{R}^{i+1} f_{i+1} \dots\dots\dots(2.52)$$

$$\tau_i = {}^i n_i^T Z_i \dots\dots\dots(2.53)$$

2-6-3-1-7 Inclusion des forces de pesanteur dans l'algorithme de dynamique :

L'effet de la gravité du chargement sur les liens peuvent être inclus, tout simplement par la mise en ${}^0 \dot{v}_0 = G$, où G est le vecteur de gravité. Ceci est équivalent à dire que la base du robot est l'accélération ascendante de 1 G accélération. Cette accélération ascendante fictive cause exactement les mêmes effets sur les liens que la gravité ainsi, sans des dépenses informatiques supplémentaires, l'effet de pesanteur est calculé [17]

2-6-3-2 Modèle dynamique direct :

Le modèle dynamique direct exprime la position, la vitesse et l'accélération des articulations en fonction du couple appliqué, il est obtenu par inversion du modèle précédent, il s'écrit :

$$\ddot{q} = M^{-1}(q)(\tau - N(q, \dot{q}) - G(q) - \tau_f) \dots \dots \dots (2.54)$$

2-6-3 Comparaison entre les deux formalismes :

Le formalisme de Lagrange est le plus adapté pour le calcul du modèle dynamique direct. il est souvent employé pour la modélisation et la simulation du système. [16]

Par contre le formalisme de Newton-Euler est utilisé pour la commande en temps réel car il offre une réduction considérable du temps de calcul du modèle dynamique inverse comme le montre le tableau suivant : [16]

Formalisme	Multiplication	Addition
Lagrange	$32\frac{1}{2}n^4 + 86\frac{5}{12}n^3 + 171\frac{1}{4}n^2 + 53\frac{1}{3}n - 128$	$25n^4 + 66\frac{1}{3}n^3 + 129\frac{1}{2}n^2 + 42\frac{1}{3}n - 96$
Newton-Euler	150n-48	131n-48

Tableau 2.6.3 Nombre d'opérations effectuées pour chaque formalisme. [16]

2-7 Les types des robots :

2-7-1 Les robots sériels

Les robots sériels, qui forment la grande majorité des robots utilisés dans l'industrie, sont nommés ainsi en raison de leur chaîne cinématique ouverte, dont les différents segments sont placés en série. Ces derniers sont liés les uns aux autres par des liaisons à un degré de liberté (rotoïdes ou prismatiques) motorisées.

Le robot sériel le plus répandu est le robot SCARA. Selon la norme ISO/TR 8373, ce mécanisme est un "robot rotoïde dont les axes de rotation des articulations simples du bras sont verticaux, avec compliance horizontale". Ce robot possède quatre *ddl*, c'est-à-dire les trois translations *x, y, z* et la rotation autour de l'axe *z* [18]

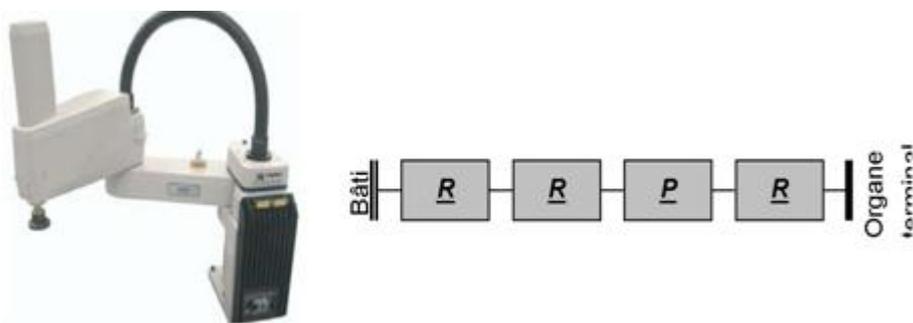


Figure 2.10 : robot à architecture sérielle. [18]

L'avantage incontesté des robots sériels est la taille de leur volume de travail ainsi que la relative simplicité des calculs liés à la commande [Clavel 1994a] [18]. Cependant, ils souffrent d'inconvénients qui peuvent être particulièrement pénalisants, en fonction de l'application :

2-7-1 -1 Rapport charge utile / masse du robot :

"Dans le cas d'un manipulateur sphérique [16] ce rapport n'est pas supérieur à 0,1. Pour une masse transportée de l'ordre de 500 kg, la masse du manipulateur atteindrait alors une valeur approximative de 5 tonnes [Merlet 1997] [18].

2-7-1-2 Précision de positionnement :

Il est communément admis que, dans la plupart des cas, la précision d'un robot série est médiocre [Merlet 1997].

2-7-1-3 Masses importantes en mouvement :

La disposition successive des segments ainsi que la nécessité de les rigidifier vont faire que la partie mobile du robot sera d'une masse appréciable. En conséquence, lors d'un mouvement à grande vitesse le manipulateur est soumis à des forces perturbatrices (inertie, forces centrifuge et Coriolis) qui vont rendre complexe la commande du robot." [Merlet 1997]

Notons que les problèmes de précision de positionnement n'ont pas lieu sur des robots de type cartésien. Dans le cas de mécanismes anthropomorphiques, les deux premières caractéristiques évoquées ci-dessus sont souvent contraignantes pour un grand nombre d'applications mais ne sont pas pénalisantes pour les applications de pick-and-place, en effet, ce type de tâche requiert peu de précision et les pièces à déplacer sont souvent de faible masse. En revanche, la contrainte induite par l'importance des masses en mouvement sera un point critique pour les tâches de prise-dépose à fortes cadences

2-7-2 Les robots parallèles :

Selon Merlet [Merlet 1997], un manipulateur parallèle généralisé est un mécanisme en chaîne cinématique fermée dont l'organe terminal est relié à la base par plusieurs chaînes cinématiques indépendantes. De plus, "un manipulateur pleinement parallèle est un manipulateur parallèle dont le nombre de chaînes est strictement égal au nombre de degrés de liberté de l'organe terminal."

Afin de faire l'analogie avec le robot sériel SCARA présenté précédemment, nous donnons ici l'exemple d'une architecture à deux *ddl* dont les mouvements en x et y sont réalisés par une

architecture répondant à la définition donnée ci-dessus. En effet, les deux *ddl* sont réalisés à l'aide de deux chaînes cinématiques fermées indépendantes [18]

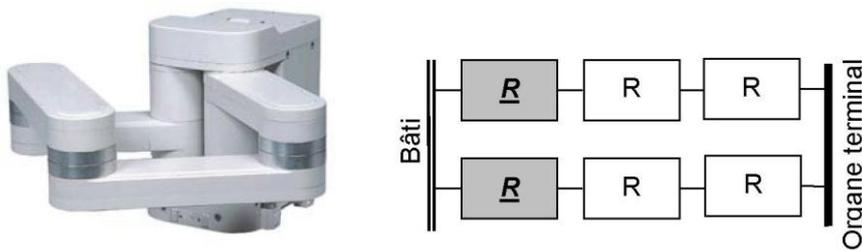


Figure 2.11 : robot à architecture parallèle. [18]

Les robots parallèles ont la réputation d'être précis et sont capables de manipuler des charges élevées tout en gardant une excellente rigidité.

Selon les mécanismes, les actionneurs de ces architectures peuvent être soit fixés sur le bâti, soit en mouvement (dans ce cas, le moteur reste malgré tout proche du bâti). Quoiqu'il en soit, les masses en mouvement sont très faibles en comparaison avec les architectures série ce qui leur confère de très bonnes performances dynamiques. Cependant leur défaut majeur est leur faible volume de travail par rapport à leur empreinte au sol. L'évolution récente de ce type d'architecture a connu deux étapes majeures : la création du premier hexapode dans les années 50 et la mise au point du premier "robot parallèle léger" dans les années 80 [Clavel 1988].

2-7-3 Comparaison des architectures sérielles et parallèles :

Il est communément admis que les robots parallèles ont un bien meilleur dynamique que les robots série s, bien que ces derniers aient un volume de travail plus important.

Nous proposons de comparer à l'aide de données concrètes ces deux familles de robots afin d'évaluer ces différences en termes de dynamique et de volume de travail. Pour cela, nous étudions deux mécanismes à deux *ddl*, l'un sériel et l'autre parallèle. Il est à noter que ces deux robots sont simulés avec des composants exactement identiques (longueurs l , masses m , inerties i_z suivant z au centre de gravité), comme présenté à la figure :

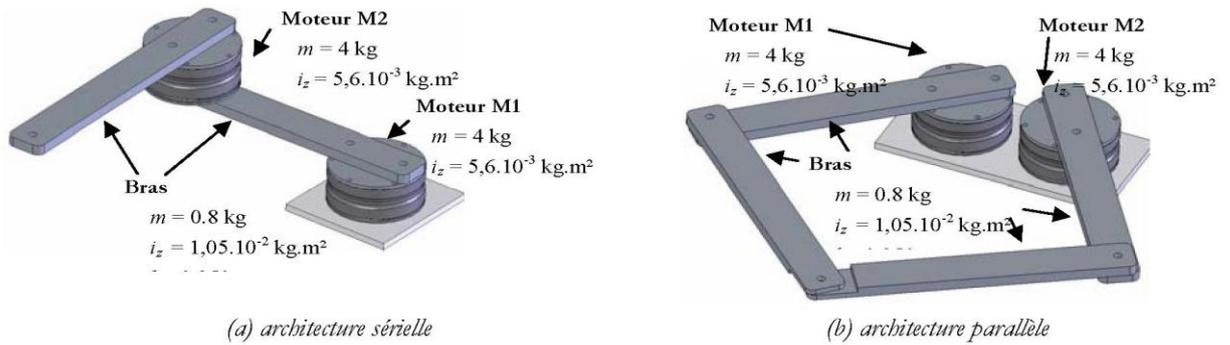


Figure 2.12 : Modélisation des robots à 2 degrés de liberté utilisés pour la comparaison. [18]

2-8 Evitement d'obstacle mobile :

Le problème de l'évitement d'obstacles fixes peut-être résolu en utilisant les méthodes de planification globale vues dans la partie suivante chapitre 3. Par contre, dans le cas de l'évitement d'obstacles mobiles, d'autres solutions prenant en compte le caractère dynamique des obstacles doivent être envisagées. De manière générale, ce problème est formulé de la façon suivante : on considère un espace de travail W dans lequel se déplacent plusieurs robots mobiles. L'objectif est d'éviter les collisions entre les robots ou avec les obstacles. Plusieurs techniques dédiées à ce problème ont été développées dans la littérature, les principales approches sont présentées dans la suite de cette section. [19]

2-8-1 Méthode de fenêtre dynamique :

Le principe de la méthode est de travailler dans l'espace des vitesses plutôt que dans l'espace des positions. La recherche de commandes à envoyer au robot pour éviter les obstacles mobiles est alors effectuée directement dans cet espace.

La dynamique du robot est introduite dans la méthode par la réduction de l'espace de recherche aux seules vitesses admissibles par le robot en prenant en compte les contraintes dynamiques telles que les valeurs minimales et maximales des vitesses linéaire et angulaire. Cette technique proposée par /Fox 1997/ est décrite par l'algorithme suivant :

2-8-1-1 Espace de recherche :

L'obtention de l'espace de recherche des vitesses admissibles pour le robot est décrite dans les quatre étapes suivantes :

- Détermination de l'espace de vitesses V_s du robot selon ces capacités (vitesses linéaire et angulaire minimale et maximale).
- Détermination de l'espace admissible de vitesses V_a , en éliminant les vitesses qui peuvent entraîner une collision avec les obstacles.

- Détermination de l'espace de vitesses V_d qui respectent les limites physiques des accélérations du robot mobile
- Détermination de l'espace de vitesse V_r à envoyer au robot mobile correspondant à l'intersection des espaces de vitesses précédents (V_s, V_a, V_d)

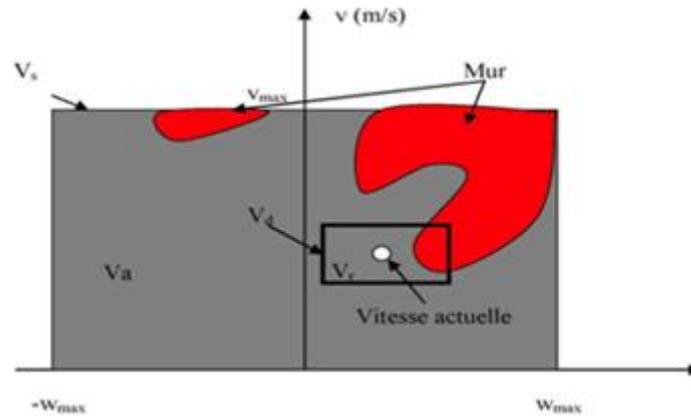


Figure 2.13 : Espace de recherche de vitesses. [19]

L'approche de fenêtre dynamique (Dynamic Window) a été étendue par /Brock, 1999/ où les auteurs ont proposé une approche de fenêtre dynamique globale, dont l'objectif est de permettre une exécution robuste dans le cas de mouvements rapides. Cette dernière est une combinaison entre la planification de mouvement et de l'évitement d'obstacle en temps réel. D'autres travaux (/Ögren 2005 et Ögren 2003/) utilisent une approche de fenêtre dynamique souple et convergente dont l'objectif est d'assurer la convergence de la fenêtre dynamique par une commande par fonction de Lyapunov (CLF) combinée avec un modèle prédictif. Les résultats cités précédemment portent principalement sur la modification de la fenêtre dynamique, il est possible de travailler sur l'algorithme d'optimisation comme c'est le cas dans /Seder 2007/ où un algorithme D* est développé pour améliorer l'évitement d'obstacles mobile.

2-8-2 Approche du V-Obstacle :

Le concept de V-Obstacle a été introduit par /Fiorini 1993/ pour le cas d'un obstacle se déplaçant en ligne droite avec une vitesse constante.

Définition 1.1 /Large 2003/: Soient un robot A, un obstacle B et un horizon de temps TH. Le V-obstacle de A associé à B est l'ensemble, noté VO, des vitesses linéaires constantes appliquées à A à l'instant t_0 , qui entraînent une collision future entre A et B sur l'intervalle de temps $[t_0, TH]$. Il est défini dans l'espace des vitesses V par :

$$V_0 = U \left\{ v_A^{\rightarrow} \in V, \exists t \in [t_0, TH], A(t) \cap B(t) \neq \emptyset \right\} \dots \dots \dots (2.55)$$

où $A(t)$ et $B(t)$ sont les trajectoires du robot et de l'obstacle respectivement.

En notant P_A et P_B les positions de A et B à l'instant t , le concept de V -obstacle peut être présenté géométriquement comme suit :

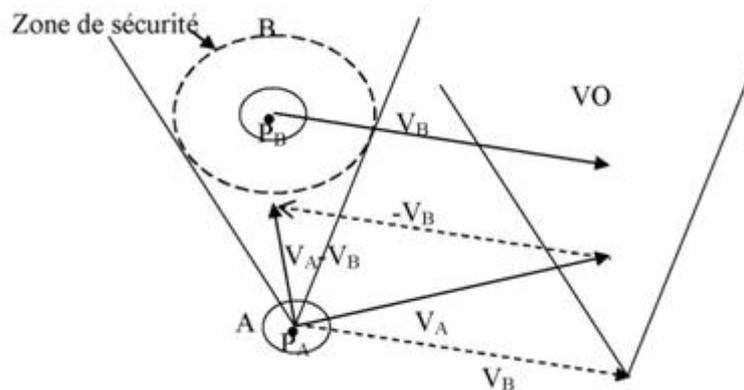


Figure 2.14 : Représentation géométrique du concept de V -obstacle. [Fiorini 1993]

où $V_A - V_B$ est la vitesse relative de A par rapport à B . La construction se fait en trois étapes

1. Construire le cône issu de P_A et dont les bords sont tangents à la zone de sécurité autour de B . Ce cône est appelé cône de collision.
2. Translater le cône de collision de V_B
3. Supprimer les vitesses qui entraînent des collisions après TH

Les deux inconvénients majeurs de cette approche sont, d'une part, que le domaine d'application est restreint aux obstacles qui roulent avec des vitesses linéaires constantes et, d'autre part, que l'efficacité de l'évitement d'obstacle dépend de l'horizon de temps choisi TH .

Ces problèmes ont été traités par /Shiller 2001, Large 2003 et Large 2005/ où une version étendue du concept de V -obstacle est proposée, incluant le cas où la vitesse de l'obstacle est variable et réduisant la dépendance à TH . Une autre amélioration du concept de V -obstacle a été introduite par /Berg 2008/ en développant un nouveau concept appelé V -obstacle réciproque dans lequel le problème d'oscillations de la trajectoire, qui peut se produire dans le cas classique, est résolu.

2-8-3 Autres approches d'évitements d'obstacle mobile :

Ils n'existent quasiment aucun résultat sur la planification analytique de trajectoire d'un robot mobile non-holonome évoluant dans un environnement dynamique et incertain.

Cependant, certaines approches traitent de ce problème dans un contexte restreint, par exemple, une approche analytique de génération de trajectoires pour un robot mobile en présence d'obstacles en mouvement a été introduite par /Qu 2004/. Dans cet article, un

modèle cinématique du robot mobile non-holonome est utilisé et la trajectoire générée pour ce dernier change en temps réel en fonction de l'environnement. D'autres approches d'évitement réactif d'obstacle pour des systèmes non-holonomes ont été proposées par /Lefebvre 2004, Lefebvre 2006 et Lamiroux 2004/. Le principe général de leur méthode est de perturber les entrées de commande définissant la trajectoire de référence calculée à l'instant initial de façon à ce que la trajectoire déformée s'éloigne des obstacles tout en respectant les contraintes cinématiques du système. Un algorithme temps réel d'évitement d'obstacles permet d'optimiser et de réduire le temps de calcul de la nouvelle trajectoire.

2-9 Conclusion :

Dans ce chapitre on a présenté les grandes lignes de la modélisation des robots en général avec un intérêt particulier qui a été porté aux bras manipulateurs à chaîne ouverte, commençant par la description géométrique directe qui a été décrite d'une manière générale en choisissant la notation de Denavit-Hartenberg pour obtenir des matrices de transformation homogènes régissant les différents liens d'un robot.

Quant à la géométrie inverse, la complexité du problème ne nous a pas été favorable pour donner une méthode générale pour décrire le modèle géométrique inverse, on s'est contenté de donner les définitions nécessaires et on a laissé la tâche de description du modèle pour le chapitre 4 en traitant un mécanisme spécifié.

Pour la modélisation dynamique on a présenté les deux formalismes (Lagrange) et (Newton-Euler) avec les deux récurrences avant et arrière.

L'étude faite dans ce chapitre nous a permis de préparer les différents modèles mathématiques régissant un robot ou bras manipulateurs pour le chapitre 4.

CHAPITRE 3

PLANIFICATION DE TRAJECTOIRES.

3-1 Introduction:

Un robot est une machine équipée de capacités de perception, de décision et d'action qui lui permettent d'agir de manière autonome dans son environnement en fonction de la perception qu'il en a. Les robots mobiles sont largement utilisés dans les environnements industriels pour le transport de produits. Le plus souvent ces tâches sont répétitives et suivent un chemin bien défini, parfois même bien matérialisé comme des lignes sur le sol. Il y a actuellement une forte tendance à élargir les milieux où évoluent les robots à des environnements de bureaux ou à des environnements domestiques (robots de service). Les types d'applications possibles sont innombrables. Cela peut être des tâches de nettoyage et d'entretien ou encore une assistance à une personne handicapée dans des tâches d'exploration et de préhension. On parle alors, de façon générale, de robotique d'intérieur.

Un tel cadre d'utilisation requiert que le système robotisé dispose d'un niveau minimum d'autonomie et de facilités de navigation. Pour ce faire, le système doit généralement accomplir trois tâches de base qui sont la localisation, la planification et la navigation. La robotique mobile vise à rendre un système autonome dans ses déplacements. Un robot, doté de capacités de perception et d'information sur son environnement, doit pouvoir se mouvoir en autonomie, sans se perdre tout en évitant les obstacles. Une des tâches que doit accomplir le robot est donc de planifier sa trajectoire dans l'environnement.

3-2 La problématique générale :

Pour un robot A évoluant dans un environnement W donné, le problème général de planification consiste à déterminer pour A un mouvement lui permettant de se déplacer entre deux configurations données tout en respectant un certain nombre de contraintes et de critères. Ceux-ci découlent de plusieurs facteurs de natures diverses et dépendent généralement des caractéristiques du robot, de l'environnement et du type de tâche à exécuter. En l'occurrence, les contraintes relatives au robot concernent sa géométrie, sa cinématique et sa dynamique et leur prise en compte peut être complexe selon l'architecture initiale considérée. Cette architecture pouvant correspondre à un système articulé d'objets rigides tel qu'un bras manipulateur, une main à plusieurs doigts ou un véhicule tractant des remorques, ou encore à plusieurs systèmes de robots à coordonner tels que des bras manipulateurs ou des robots mobiles de type voiture évoluant dans un réseau routier. Les contraintes émanant de l'environnement concernent essentiellement la non collision aux obstacles fixes encombrant

W et la prise en compte d'interactions de contact avec le robot. L'évitement d'obstacles dépend de la géométrie de l'environnement et est commun à toutes les tâches robotiques. De plus aucune hypothèse simplificatrice n'est faite sur la géométrie du robot ou sur celle de l'environnement. [20]

Les critères à satisfaire pendant la résolution du problème de planification concernent le fait qu'une solution doit optimiser une fonction de coût exprimée en terme de la distance parcourue par le robot entre les deux configurations extrêmes, de la durée ou de l'énergie nécessaires à l'exécution de son mouvement. D'autres critères peuvent être également considérés tels que la prise en compte de distance de sécurité aux obstacles pour un robot mobile ou manipulateur ou encore la "qualité" et la stabilité des prises pour une main articulée. Face à la nature aussi diverse de ces aspects et aux difficultés qu'elle peut induire sur un processus de résolution, la plupart des travaux proposés dans le domaine de la planification de mouvement ont porté sur la considération de certaines instances du problème général. Nous présentons dans la suite de ce chapitre les principaux concepts et approches développés à cet effet. Après ce bref exposé de la problématique de la planification de trajectoire, plusieurs questions restent en suspens :

- Comment exprimer les obstacles dans l'espace de configuration ?
- Comment trouver un chemin dans l'espace opérationnel libre ?
- Si plusieurs chemins sont possibles, y-a-t'il un chemin optimum ?
- En présence de contraintes de nature différente (géométrique, cinématique, dynamique) est-ce que le choix doit satisfaire un compromis entre toutes les contraintes ?
- Une fois que le chemin est désigné, comment générer un mouvement dans l'espace articulaire produisant la trajectoire donnée dans l'espace *opérationnel* ?

Le terme même de trajectoire mérite d'être précisé : on peut en effet distinguer au moins trois niveaux d'acception de ce mot .Un premier niveau, très général assimile tous les mouvements d'un robot à des trajectoires, sans distinction de nature (amplitude, vitesse etc.) ou d'objectif (mouvement de contact, d'évitement etc.). Pour des raisons de complexité et d'efficacité, le vaste problème de planification qui en découle est généralement décomposé d'un point de vue fonctionnel en sous problèmes autorisant ainsi le développement de planificateurs dédiés et d'algorithmes spécialisés .Enfin, au niveau le plus bas de la commande des actionneurs (des moteurs du robot), le terme désigne le déplacement réellement exécuté entre deux positions de consignes : on parle alors plutôt de génération de trajectoire.

3-3 Loi d'orientation :

On appelle une loi d'orientation ζ toute fonction qui associe à chaque position du point de référence une orientation : [21]

$$\zeta : (x, y) \in R^2 \rightarrow \mathcal{G} \in \{0, 2\pi\} \dots\dots\dots (3.1)$$

On impose toute loi d'orientation comme étant de classe au moins C^0

3-4 Interpolation linéaire :

Une loi d'orientation qui est souvent utilisée en robotique est l'interpolation linéaire

$$x_1 \neq x_2 \dots\dots\dots (3.2)$$

$$\zeta(M_1, M_2) : (x, y) \rightarrow \mathcal{G} = \mathcal{G}_1 + \frac{(x - x_1)(\mathcal{G}_2 - \mathcal{G}_1)}{(x_2 - x_1)} \dots\dots\dots (3.3)$$

Sinon :

$$\zeta(M_1, M_2) : (x, y) \rightarrow \mathcal{G} = \mathcal{G}_1 + \frac{(y - y_1)(\mathcal{G}_2 - \mathcal{G}_1)}{(y_2 - y_1)} \dots\dots\dots (3.4)$$

L'inconvénient de l'interpolation linéaire est que l'on ne sait pas déterminer pour un robot parallèle plan si une trajectoire constituée d'un segment de droite pour le point de référence et d'une interpolation linéaire pour l'orientation est entièrement dans l'espace de travail.

En effet, pour cette détermination il faut tester, si lors du mouvement la trajectoire des points B_1, B_2 coupent les cercles des couronnes annulaires associées aux segments, mais cette trajectoire est une partie cycloïde d'équation :

$$\text{Pour } \alpha \in \{0;1\} \dots\dots\dots (3.5)$$

$$x = x_1 + \alpha(x_2 - x_1) + r \cos(\mathcal{G}_1 + \alpha(\mathcal{G}_2 - \mathcal{G}_1)) \dots\dots\dots (3.6)$$

$$y = y_1 + \alpha(y_2 - y_1) + r \sin(\mathcal{G}_1 + \alpha(\mathcal{G}_2 - \mathcal{G}_1)) \dots\dots\dots (3.7)$$

r : La distance constante du point de référence.

M_1 et M_2 Les points de départ et arrivé respectivement.

3-5 Modélisation de l'espace de planification :

3-5-1 Notions et définition :

Les données du problème de planification de trajectoire sont donc disponibles sous la forme de représentations géométriques mal adaptées à sa résolution, nous définissons donc l'espace de planification E comme la représentation sur laquelle notre système doit raisonner afin de planifier une ou plusieurs trajectoires.

Plus généralement, cet espace représente l'ensemble des états du robot qu'il est possible de réaliser avec commande du système, et de différencier à l'aide des moyens perceptifs disponibles [22] [23] [24]. En pratique seuls nous intéressent les informations sensorielles de nature proprioceptive permettant de mesurer la dynamique du robot (position, vitesse, accélération).

Pour un robot à N degrés de liberté, l'espace de planification peut donc s'exprimer par un ensemble de 3N coordonnées généralisées :

$$E = \{(g_1, \dots, g_N, \dot{g}_1, \dots, \dot{g}_N, \ddot{g}_1, \dots, \ddot{g}_N)\} \dots \dots \dots (3.8)$$

3-5-2 Espace de configuration :

On considère à tout instant le robot comme étant statique. Ceci a pour effet de partitionner les états de manipulateurs en classes d'équivalences telles que sont associés tous les états atteignables de même position mais de vitesse ou d'accélération distincte. On appelle une telle classe d'états une configuration notée C, de manière générique.

Une configuration C est donc définie comme N variables généralisées (ou valeurs articulaires) de la structure cinématique (g_1, \dots, g_N) .

L'ensemble de toutes les classes d'états équivalents est donc appelé espace de configuration noté EC :

$$EC = \{(g_1, \dots, g_N)\} \dots \dots \dots (3.9)$$

3-5-3 Espace libre :

L'espace de configuration peut être clairement partitionné en deux sous espaces non connexes. Le premier qu'on appelle espace de contact EK, est l'ensemble de toutes les configurations du manipulateur qui impliquent une intersection matérielle entre deux objets de l'environnement, événement alors nommé collision.

On parle ici d'objets au sens général, incluant par conséquent les composants même du robot, les pièces transportées, les meubles.

Si la collision la plus fréquente est celle d'un élément de la structure articulée avec un volume solide présent dans l'univers, on doit également inclure celle, plus délicate, entre un solide fixe et un autre transporté par le robot. L'espace de contact est à son tour partitionné en un nombre fini de sous espaces connexes correspondant à chacun des obstacles de l'environnement.

Le complémentaire de l'espace EK, par rapport à l'espace EC définit donc l'ensemble de configurations valides du manipulateur. Soit celle qui n'engendre aucune collision ; on va la

noter donc espace libre EL: le but de la construction de l'espace de planification est précisément de déterminer l'espace libre du manipulateur.

Enfin la frontière entre l'espace de contact et l'espace libre constitue la frontière de contact EF, formé d'un ensemble de surfaces (de dimension N-1) usuellement appelées (C-surfaces).

$$EK \cup EL = EC, EK \cap EL = \phi, \text{ et } EK \cap EL = EF \dots\dots\dots (3.10)$$

3-6 Construction de l'espace de planification :

Le problème de la construction de l'espace libre est équivalent à celle de l'espace de contact, celui-ci traduisant les contraintes imposées par l'environnement, cette construction se heurte à deux difficultés majeurs: l'une d'ordre conceptuel, l'autre relative à la réalisabilité de la méthode conduisant à une solution. En effet, la complexité de l'espace de configuration ne permet pas de l'exprimer de manière analytique et sa grande dimension empêche en pratique la construction d'une représentation exacte de contraintes de l'environnement (d'une complexité exponentielle). De fait, l'ensemble de configurations, s'il est en pratique fini, reste d'une taille trop importante pour pouvoir opérer par énumération.

3-7 Génération de trajectoires :

3-7-1 Introduction :

3-7-1-1 Définition :

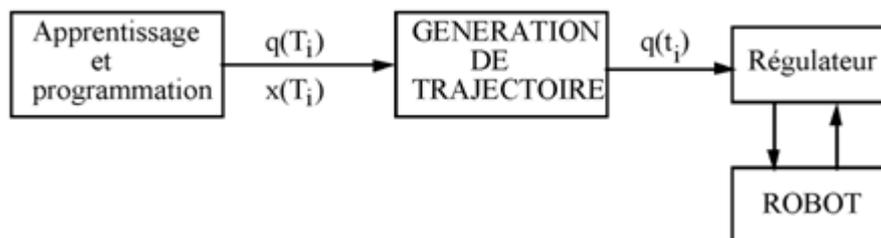


Figure 3.1 : Génération de trajectoires. [25]

La phase de génération de trajectoires se situe entre les phases d'apprentissage (ou programmation) et d'exécution du mouvement. Les points programmés ou appris pour planifier ces trajectoires ou mouvements peuvent être donnés : [25]

- soit dans l'espace articulaire $q(T_i)$
- soit dans l'espace cartésien $x(T_i)$

Dans le schéma ci-dessus :

$q(T_i)$ ou $x(T_i)$ sont les points appris ou programmés par l'opérateur pour spécifier le mouvement désiré pour le robot. Typiquement, les intervalles de temps $\Delta T = T_{i+1} - T_i \gg 10$ millisecondes (par exemple, la liaison computer-Link ALTER du robot Puma permet un transfert de données toutes les 28 millisecondes),

$q(t_i)$ sont les valeurs de consignes (en coordonnées articulaires) transmises au régulateur. Ici les intervalles de temps $\Delta t = t_{i+1} - t_i$ sont de l'ordre de la milliseconde.

Il est donc nécessaire d'interpoler d'une manière ou d'une autre entre les points qui ont été appris ou programmés.

On distingue différents types de mouvements suivant les applications (tâches) :

1. Trajectoire point à point (P.T.P.) : Pour la manutention (pick and place), pour l'assemblage, la soudure par points, ...
2. Trajectoire opérationnelle : idem que P.T.P. en plus, la possibilité d'éviter des obstacles.
3. Mouvement opérationnel : la vitesse doit être contrôlée pour les opérations de soudure en continu, peinture, dépose de joints, etc...

3-7-2 Planification dans l'espace articulaire :

Si les points programmés ou appris $q(T_i)$ sont donnés dans l'espace articulaire EC l'interpolation peut être tout naturellement calculée dans ce même espace :

Avantage : cet espace EC est celui utilisé normalement par les algorithmes de régulation (qui utilisent les informations proprioceptives); la génération de trajectoires peut donc être réalisée sous la forme d'un algorithme simple, calculable en temps réel.

Inconvénient : la planification des $q(T_i)$ ne se situe pas dans l'espace naturel de l'opérateur : celui-ci ne dispose d'aucun moyen de visualiser ce que sera la trajectoire, notamment pour éviter les risques de collisions.

Il faut s'assurer que les contraintes suivantes soient observées pour calculer les $q(t_i)$:

- l'algorithme ne peut pas être itératif car $q(t_i)$ doit être disponible à l'instant t_i
- les fonctions $q(t_i)$, $\dot{q}(t_i)$ (et $\ddot{q}(t_i)$) doivent être continues pour éviter des sauts trop importants des valeurs de consignes
- les valeurs $\dot{q}(t_i)$ (et $\ddot{q}(t_i)$) doivent tenir compte des limitations des actionneurs et des chaînes de transmissions :

$$|\dot{q}(t_i)| \leq k_v \text{ et } |\ddot{q}(t_i)| \leq k_a \dots\dots\dots (3.11)$$

Où k_v et k_a sont des constantes caractérisant les limites des actionneurs

- il faut éviter que l'interpolation ne donne lieu à des déviations (excursions) trop importantes dans l'espace articulaire.

3-7-3 Planification dans l'espace cartésien (opérationnel) X :

3-7-3-1 Interpolation et régulation cartésienne :

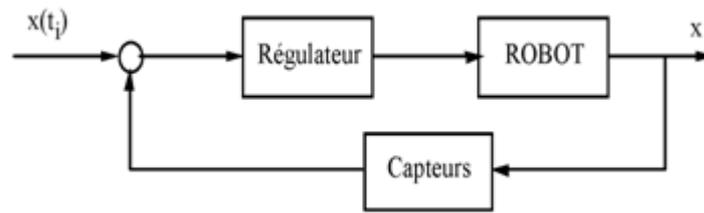


Figure 3.2 : Planification dans l'espace cartésien (opérationnel) X. [25]

Avantage :

- précision du suivi.

Inconvénient :

- peu ou pas de capteurs extéroceptifs disponibles pour le régulateur
- la transformation $x(t_i) \Leftrightarrow q(t_i)$ est non univoque et difficile à réaliser en temps réel
- les limitations k_v et k_a dues aux actionneurs ne sont pas prises en compte au cours de la génération de trajectoire.

3-7-3-2 Interpolation et régulation articulaire :

On effectue d'abord les transformations $x(T_i) \Rightarrow q(T_i)$, et ensuite on interpole par polynômes de degrés peu élevés pour obtenir les consignes $q(t_i)$.

Avantage :

- les calculs à faire en temps réel sont limités
- les limitations k_v et k_a peuvent être prises en compte lors de l'interpolation.

Inconvénient : risque d'imprécision du suivi sur les tronçons interpolés.

3-7-3-3 Trajectoires planifiées dans l'espace articulaire :

Trajectoires sans contraintes (mouvement P.T.P.)

La loi généralement utilisée pour chacun des axes du robot est du type bang-bang, ce qui correspond à la solution en temps minimum pour un déplacement Δq donné. Pour tenir compte si nécessaire, des limitations en vitesse, on introduit une phase de palier en vitesse. Cette loi présente pour inconvénient le fait d'introduire des discontinuités importantes en accélération ("JERK") provoquant des oscillations du régulateur qui se manifestent par des vibrations du robot.

D'autres lois, dérivées de cette loi BANG-BANG, sont proposées pour diminuer le "JERK" sans pour autant allonger outre mesure le temps d'exécution du mouvement.

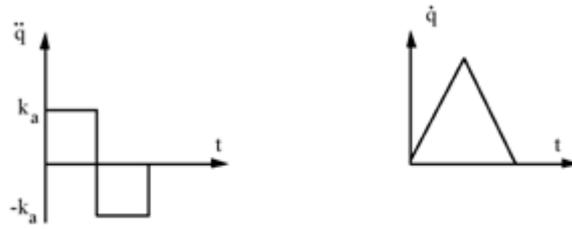


Figure 3.3 : Loi bang-bang pour un axe : $\ddot{q} = k_a$ sur $(0, T/2)$, $\ddot{q} = -k_a$ sur $(T/2, T)$ [25]

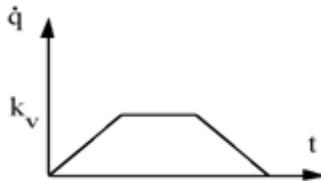


Figure 3.4 : Loi bang-bang avec palier de vitesse. [25]

	Loi "Bang-Bang"	Loi "Bang-Bang modifiée"	Loi polynomiale (2 polynômes degré 4)
Courbes Vitesse - Accélération			

Figure 3.5 : Lois alternatives à la loi Bang-Bang. [25]

Type du mouvement	Expression de la durée de parcours	Ecart en durée par rapport à la loi "Bang-Bang"
"Bang-Bang"	$2 \cdot \sqrt{\frac{\Delta q}{K_a}}$	-----
"Bang-Bang modifiée"	$\frac{2}{\sqrt{1-\alpha}} \cdot \sqrt{\frac{\Delta q}{K_a}}$	$\alpha = 0 \quad \dots \quad 0\%$ $\alpha = 0.5 \quad \dots \quad 35\%$
Polynomiale (1 polynôme de degré 5)	$\sqrt{\frac{10}{3}} \cdot \sqrt{\frac{\Delta q}{K_a}}$	20.1%
Polynomiale (2 polynômes de degré 4)	$\sqrt{6} \cdot \sqrt{\frac{\Delta q}{K_a}}$	22.5%
Cosinus	$\frac{\pi}{\sqrt{2}} \cdot \sqrt{\frac{\Delta q}{K_a}}$	11%

Figure 3.6 : Lois alternatives à la loi Bang-Bang. [25]

3-8 Coordination des articulations :

Le temps de parcours de chaque articulation prise isolément est une fonction de la distance à parcourir Δq et des contraintes cinématiques. Comme ces éléments peuvent varier d'une articulation à l'autre, on est confronté au problème de coordination entre les articulations pour que celles-ci finissent leur mouvement au même instant. Pour cela, on effectue la synchronisation des mouvements articulaires sur l'axe nécessitant la durée de parcours la plus longue pour le déplacement qui lui est imparti. Cet axe sera dit contraignant et il existe deux méthodes de synchronisation :

- la méthode proportionnelle sur l'axe contraignant,
- la méthode proportionnelle.

3-9 Trajectoire contrainte (points de passage intermédiaires) :

Ce type de trajectoire est contraint par des configurations intermédiaires imposées. Cela peut être le cas si la tâche du manipulateur est définie de manière à éviter des obstacles et/ou pour des opérations de "pick and place" simples si on veille à éloigner/approcher l'objet correctement des sites de prise et de dépose. Dans ce dernier cas, il suffira de spécifier une position intermédiaire sur la normale à la surface de pose qui passe par le point de prise : en imposant à la pince de passer par cette position intermédiaire, on obtiendra un mouvement admissible pour le départ (lift-off). On fera de même pour la phase d'approche (set-down) nécessaire à la dépose de l'objet manipulé. De plus, en imposant les temps de parcours relatifs aux phases de lift-off et set-down, il sera possible de contrôler la vitesse du manipulateur pour ces opérations délicates.

Considérons le cas suivant :

On exige la continuité en vitesse et en accélération aux points intermédiaires de lift-off et set-down.

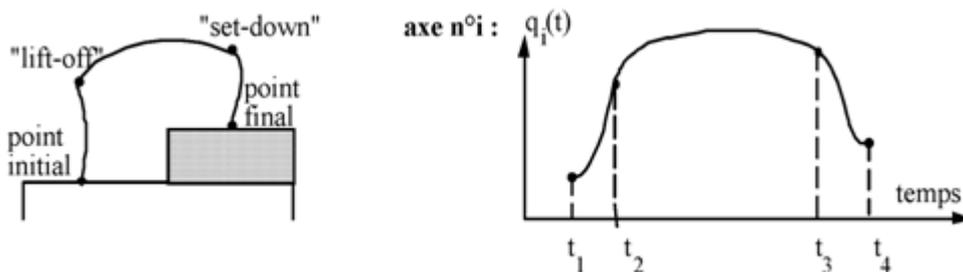


Figure 3.7 : la continuité en vitesse et accélération aux points intermédiaires de lift-off et set-down. [25]

Les durées $(t_2 - t_1)$ et $(t_4 - t_3)$ sont également imposées, tandis que la durée $(t_3 - t_2)$ pourra être optimisée compte tenu des limitations en vitesse k_v et accélérations k_a : on choisira pour tous les axes, une même durée dictée par l'articulation contraignante.

On notera que les deux méthodes [PAUL et LIN] peuvent se généraliser sans peine pour des trajectoires avec évitement d'obstacles qui comportent plus que 4 points de passage obligés. Dans ce cas, les points supplémentaires introduits par dans la méthode de LIN seront injectés sur les premier et dernier tronçons.

d_k : Le nombre total des contraintes.

Trajectoire planifiée dans l'espace cartésien

3-10 Interpolation cartésienne :

Réalisée dans l'espace naturel de l'opérateur, l'interpolation cartésienne rend plus aisée la programmation de trajectoires sans collisions. Deux difficultés apparaissent néanmoins :

1. si la notion d'interpolation (en ligne droite, en arc de cercle, etc ..) est évidente en ce qui concerne la position de l'outil, il n'en est pas de même pour son orientation ;
2. toute trajectoire comportant un tracé anguleux de l'outil nécessiterait des accélérations infinies, il sera nécessaire de prévoir des "raccords" pour le tracé de carrés, rectangles.

3-11 Interpolation articulaire :

L'interpolation cartésienne coûte cher en calcul si chaque point interpolé doit être transposé ($q = f^{-1}(x)$) dans l'espace articulaire. En outre, il est difficile de veiller aux respects des contraintes de limitations k_v et k_a .. D'où l'idée de ne calculer dans l'espace cartésien qu'un point sur n, de les transposer dans l'espace articulaire et ensuite d'interpoler plus fin dans cet espace pour subvenir aux besoins du régulateur.

Méthode de Taylor "bounded deviation joint path" L'interpolation articulaire est réalisée par segments de droite et raccords paraboliques. Supposons x_1 et x_2 donnés ($q_1 = f^{-1}(x_1)$) et ($q_2 = f^{-1}(x_2)$) (3.12)

3-12 Contraintes spatiales :

3-12-1 Identification du problème :

Les données du problème de planification de trajectoire sont relatives, en plus des contraintes de la tâche à exécuter, au robot et à son environnement connu sous la forme d'un modèle essentiellement géométrique. La clé de résolution de cette planification de trajectoires réside

donc à s'imprégner des modèles mathématiques au robot concerné et en déduire la solution adéquate à ce modèle géométrique.

3-13 Les divers aspects de la planification de trajectoires :

Un robot manipulateur est donc constitué d'une structure articulée dont la base est fixe et dont l'extrémité terminale est généralement munie d'un outil de préhension permettant de manipuler des objets. Le robot mobile est un véhicule progressant de façon autonome dans un environnement quelconque les deux peuvent être combinés : on obtient alors un robot autonome capable à la fois de déplacement, d'observation et d'action.

Le problème général de planification de trajectoire revient donc à déterminer le mouvement d'un système mécanique à N coordonnées généralisées. Sa complexité dépend de ce nombre et de la nature de ses liaisons extérieures.

Ainsi pour un robot manipulateur, on est confronté à un problème tridimensionnel dont la complexité dépend beaucoup du nombre d'articulation. Le problème est alors essentiellement de nature géométrique, et peut être formulé spécifiquement de la façon suivante :

Déplacer un solide dans l'espace (la pince du manipulateur par exemple) en évitant les objets présents. Avec les contraintes supplémentaires d'éviter toute collision à tout point de la structure articulée qui relie ce mobile à une base fixe.

Le problème de planification de trajectoire repose sur base géométrique celle du déplacement du solide au milieu d'autres solides. L'intérêt de l'espace de configuration est d'offrir la possibilité de caractériser l'existence d'une composante connexe contenant les positions initiale et finale.

3-14 Contraintes temporelles :

Si la planification dont nous parlons est à priori essentiellement spatiale, la prise en compte du temps peut y être faite à plusieurs niveaux.

L'hypothèse d'un environnement statique, figé n'est en effet pas réaliste dans le contexte de la robotique industrielle. Même en supposant que le robot soit le seul agent présent dans son domaine d'évolution, il est nécessaire de prendre en compte les modifications dues à ses propres actions. Dans ce cas l'univers du robot évolue suivant un temps qui lui est propre, et la planification d'une trajectoire dans un état donné se fait nécessairement en statique, puisque l'évolution ne se produira qu'avec le déplacement d'un objet, conséquence de cette planification même.

Par contre, si le domaine spatial atteignable par le manipulateur l'est également par un autre robot, l'hypothèse d'un état immuable n'est plus vide : d'une part les objets peuvent être déplacés par le robot étranger et, d'autre part, celui-ci constitue lui-même un objet de

l'environnement en déplacement constant. Dès lors, le modèle sur lequel opère le planificateur de trajectoires est amené à évoluer dans le temps au rythme des mouvements de l'autre robot. Entre deux de ces mouvements, on est ramené au cas d'un univers statique dans lequel le manipulateur surnuméraire n'est plus qu'un obstacle parmi d'autres. Le problème de la coopération de deux manipulateurs peut donc être vu comme un problème découplé de planification à un manipulateur augmenté de fortes contraintes temporelles. Les algorithmes utilisés doivent alors souvent être remis en question pour d'évidentes raisons de performance. Nous ne traitons pas dans ce mémoire cet aspect temporel de la planification de trajectoire. Aussi, dans la suite, et sauf mention contraire, le terme de robot désigne un manipulateur constitué d'un nombre quelconque de solides en liaison articulaire, évoluant dans un espace cartésien tridimensionnel partiellement encombré.

3-15 Synthèse des principaux travaux en planification de mouvements :

3-15-1 Introduction :

Historiquement, le problème de planification a été initialement abordé dans le cadre d'un système de robots se déplaçant dans un environnement contenant des obstacles fixes et soumis à la seule contrainte de non-collision. Cette instance du problème général de planification est connue sous le nom du *paradigme du déménageur de piano* et sa résolution a fait l'objet de plusieurs travaux dont nous présentons une brève synthèse dans la suite de cette section. La plupart de ces travaux sont basés sur le concept d'espace des configurations du robot introduit par [25]. Par configuration est désignée tout n-uplet (n-objets) de paramètres indépendants de position /orientation caractérisant d'une manière unique le robot dans son environnement ou espace de travail. Par ce concept, le problème de planification est formulé dans un espace *CSA* de dimension n , l'espace des configurations, où le robot est assimilé à un point et les obstacles à des sous-ensembles également de dimension n . Ainsi, un mouvement solution est donné par une courbe dans *CSA* correspondant à une séquence continue de configurations sans collision reliant les configurations initiale et finale considérées. Selon les cas, la solution à générer est contrainte à rester entièrement ou partiellement sans contact avec les obstacles. On parlera alors d'un mouvement dans l'espace des configurations libres ou au contact. De par sa généralité, le concept d'espace des configurations a permis d'apporter une réponse quant à l'existence ou non de chemins solutions au problème de planification et de caractériser cela par l'existence d'une composante connexe dans *CSA*.

3-15-2 Méthodes par décomposition cellulaire :

Cette méthode consiste à décomposer l'espace W d'évolution du robot (espace de configuration) en plusieurs régions adjacentes qui correspondent à une représentation discrète de l'environnement /Latombe 1991/. A partir de la décomposition obtenue, les barycentres de chaque cellule sont calculés et, en utilisant la théorie de graphe, la génération d'un graphe reliant les barycentres compris entre les configurations initiales et finales du robot est réalisée. Les points P_i et P_f sont respectivement les configurations initiale et finale du robot. Le graphe déterminé est alors exploré pour calculer le chemin permettant de minimiser un critère donné. Généralement dans le cas de robots mobiles, le critère peut-être le temps de parcours ou la détermination du plus court chemin par l'utilisation d'algorithmes d'optimisation : Dijkstra /Tournassoud 1992/, A^* /Stuart 1994/ ou D^* /Stentz 1995/.

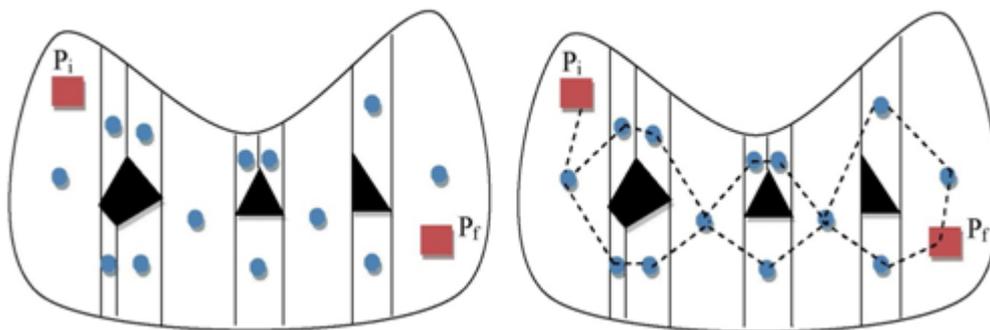


Figure 3.8 : Décomposition en cellule génération du mouvement. [Tournassoud 1992]

3-15-3 Méthodes exactes :

La première catégorie regroupe les méthodes de résolution dites exactes en ce sens que la décomposition effectuée se base sur un recouvrement exact de l'espace libre du robot de ses contacts en cellules convexes. La capacité de ce type de méthode à apporter une réponse quant à la résolution du problème général de planification a été démontrée par Schwartz et Sharir [J.T.Schwartz and M.sharir,] [26]. Le problème de planification est formulé sous forme algébrique en considérant une décomposition de l'espace libre du robot sous forme de composantes cylindriques semi-algébriques (cellules de Collins) [27].

Par cette formulation, les auteurs aboutissent à l'un des résultats importants sur la complexité de la planification, à savoir qu'elle est polynômial en la complexité de l'environnement et doublement exponentiels en la dimension de l'espace des configurations. La complexité de l'environnement est mesurée par le nombre et le degré des polynômes utilisés pour la description semi-algébrique de l'espace libre. A ce point une remarque s'impose, car le mot "cellule" peut prêter à confusion. Il est important en effet de noter que cette décomposition de l'espace libre est une méthode dite exacte, qui doit être distinguée des méthodes non-exactes. Au premier rang de ces dernières, on peut citer la discrétisation de

l'espace cartésien en cellules cubiques de tailles données, qui produit des solutions pour une résolution donnée.

3-15-4 Méthodes approchées :

Afin de remédier à cette complexité, des méthodes dites approchées sont généralement appliquées. Elles se distinguent des précédentes par la simplicité de la structure des cellules utilisées pour la décomposition et l'approximation faite de l'espace libre du robot. La structure de celui-ci est généralement capturée dans une représentation hiérarchique en cellules élémentaires identiques et permettant l'adaptation de la taille de celles-ci à la géométrie des zones à recouvrir. Plusieurs types de cellularisation de l'espace du robot sont généralement utilisés en pratique : représentation en octaèdres, découpage en tranches, ou encore en polyèdres [27]. En comparant les deux approches de décomposition, il ressort que les méthodes exactes sont plus complètes en théorie que les méthodes approchées. Toutefois, la complexité de leur mise en œuvre, les rend moins appropriées en pratique que les techniques approchées même si celles-ci restent parfois limitées à des espaces de dimension réduite. Cette constatation n'est pas générale puisqu'il apparaît dans [5] qu'une méthode exacte peut conduire dans le cas plan et pour une modélisation polygonale du robot et de son environnement à une implantation efficace.

3-15-5 Méthodes de résolution de type rétraction :

Une seconde grande approche pour la résolution du problème de planification consiste à ramener la recherche du mouvement du robot dans un espace de plus faible dimension que celle de l'espace admissible. Cela consiste à représenter la connectivité de l'espace libre du robot par un réseau de courbes unidimensionnelles pouvant être entièrement dans l'espace libre ou des contacts du robot. La planification de mouvement entre deux configurations données est alors résolue en trois étapes : détermination d'un chemin ramenant le robot de sa configuration initiale à un point situé sur l'une des courbes du réseau de connectivité, détermination d'un chemin entre la configuration finale et le réseau, et enfin exploration de celui-ci afin d'en extraire un chemin reliant les deux points connectés aux configurations initiale et finale. Une première méthode basée sur ce concept a été proposée par Nelson pour un espace de travail bidimensionnel encombré d'obstacles polygonaux [28]. Dans cette méthode, le réseau est décrit par un graphe, dit graphe de visibilité, où les nœuds correspondent aux sommets des obstacles et les arêtes à des segments de droite reliant ces sommets dans l'espace libre. L'inconvénient de cette méthode est que les solutions obtenues conduisent le robot à être en contact avec les obstacles en certains de leurs points (les sommets) même en présence d'une solution entièrement contenue dans l'espace libre. Afin

d'y remédier, une seconde méthode, connue sous le nom de rétraction et développée par Ô'Dùnlaing et Yap [29] consiste à utiliser un diagramme de Voronoi [30] pour capturer la connectivité de l'espace libre et générer des solutions éloignées le plus des obstacles. Une telle méthode, bien qu'elle soit générale en théorie, reste limitée à des espaces de dimension peu élevée. Une autre variante de cette méthode est développée par Brooks pour des espaces de travail polygonaux et se base sur la représentation de ceux-ci par des cylindres généralisés. Les chemins à suivre par le robot sont alors déterminés en considérant la connectivité entre les axes de ces cylindres. Enfin, une dernière méthode de type rétraction a été présentée par Canny dans le cadre de la résolution du problème général de la planification [P.Tournassoud, 1992]. L'algorithme proposé est exponentiel en la dimension de l'espace des configurations, et constitue l'un des résultats les plus importants sur la complexité du problème de planification de mouvement.

3-16 Autres méthodes de planification :

Les méthodes exposées dans les paragraphes précédents sont généralement dites globales sachant qu'elles se basent toutes sur la structuration de l'espace des configurations et la modélisation à priori de sa connectivité.

Afin d'éviter la complexité d'une telle étape de structuration, d'autres méthodes dites locales sont généralement utilisées. Leur principe consiste à déterminer les déplacements du robot en ne considérant qu'une représentation locale de l'environnement et à percevoir la planification de mouvement comme un problème d'optimisation.

3-16-1 Méthode du potentiel :

Dans cette approche, on considère que le robot roule dans un champ de forces virtuelles. Ce dernier est composé de deux champs : un champ de potentiel répulsif autour des obstacles et un champ de potentiel attractif dont le point d'attraction est la configuration finale du robot.

Sachant que q est la configuration courante du robot, la commande du robot mobile permettant d'éviter un obstacle noté 'O' et d'atteindre la configuration désirée (cible) qui est donnée par le champ de potentiel artificiel composé du champ de potentiel attractif et répulsif /Khatib 1986/ /Khalil 1999/. Dans le cas d'évitement de plusieurs obstacles 'O_i' ($i = 1, \dots, n$) où n est le nombre des obstacles, la commande à envoyer au robot est composée de la force résultante du champ de potentiel attractif et de la somme des forces qui résultent des champs de potentiel répulsifs des différents obstacles. Cette technique se met en oeuvre relativement facilement et peut être adaptée au cas d'obstacles mobiles. Par contre, le robot peut rester

bloqué en un minimum local ou osciller lors du franchissement d'un passage étroit entre des obstacles.

Plusieurs adaptations de cette technique ont été proposées. Par exemple, pour éviter que le robot ne soit bloqué (minima locaux), les auteurs /Koren 1991/, /Barraquand 1992/ et /Kavraki 1996/ proposent d'ajouter un point intermédiaire aléatoire afin de pouvoir sortir le robot de sa trappe.

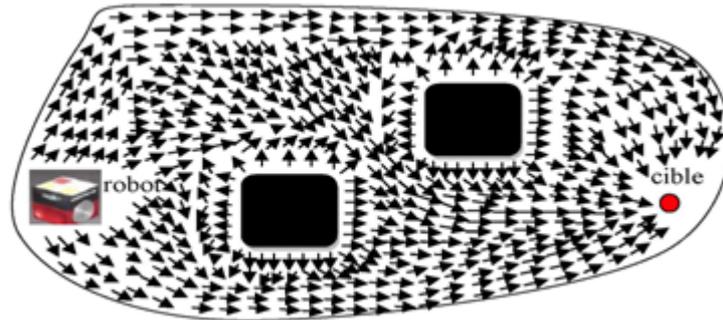


Figure 3.9 : Champ de potentiel attractif et répulsif. [Koren 1991]

3-17 Méthode des contraintes :

Dans le cadre d'une approche locale, Faverjon et Tournassoud proposent une méthode, dite des contraintes, qui aborde le problème de l'évitement d'obstacles en modélisant localement chacun de ceux-ci par l'ensemble de ses plans tangents [31]. La génération des mouvements du robot est obtenue par la minimisation d'un critère quadratique sur la tâche à effectuer en présence de contraintes linéaires associées aux équations des plans tangents aux obstacles. A la différence de la méthode du potentiel, les obstacles n'agissent sur le robot pendant le processus de minimisation que quand il en est très proche et a tendance à y pénétrer. Toutefois, la méthode reste sensible à la présence de minimums locaux. Ce problème est abordé dans [32] en combinant la méthode locale avec une technique d'apprentissage.

3-18 conclusions :

Dans ce chapitre on a présenté d'une manière générale les sujets de trajectoires en robotique, après avoir abordé les différents aspects du problème de planification, et avoir classé les types de méthodes utilisées.

On aborde le problème de suivi des trajectoires dans l'espace opérationnel et la génération de mouvements dans l'espace articulaire.

CHAPITRE 4

APPLICATION.

4-1 Introduction :

Partant des fondements théoriques, abordés dans les chapitres précédents, qui consistent en la modélisation en robotique avec tous ces axes d'études. La modélisation avec ses modèles : géométrique, cinématique et dynamique pour décrire le comportement du robot lors d'une trajectoire connue. Dans ce chapitre on procède à la modélisation du robot SCARA et confronter les résultats obtenus du modèle cinématique avec le logiciel Solidworks et MATLAB/Simulink, le modèle dynamique du robot nous permet de choisir la posture du robot SCARA du point de vue économie d'énergie.

4-2 Modélisation des bras manipulateurs :

La réalisation d'une simulation théorique permettant de décrire l'état et le comportement global d'une structure mécanique articulée ou tout autre système nécessite de combiner plusieurs concepts mathématiques.

Le souci principal dans la modélisation est de trouver une relation entre les consignes données dans l'espace opérationnel de la tâche et les postures des éléments du robot dans l'espace articulaire : cette relation permet de fournir une interface qui occulte le problème mathématique à l'utilisateur par le biais d'un programme informatique afin de manipuler un robot de manière interactive.

Vu la complexité et la non linéarité des problèmes, un modèle complet du comportement réel d'un robot n'est pas réalisable, L'étude cinématique et dynamique des robots se fait généralement sur des modèles simplifiés, obtenus en introduisant des hypothèses simplificatrices dans les équations qui décrivent le comportement du modèle.

Dans ce chapitre nous allons décrire la théorie sous-jacente aux problèmes de modélisation des robots de type SCARA tout en se basant sur les hypothèses suivantes :

- chaque corps composant le robot est assimilé à un solide indéformable.
- Chaque articulation ne possède qu'un seul degré de liberté

4-3 Problème de la cinématique directe :

Ayant un ensemble de caractéristiques géométriques du robot avec ses variables articulaires, nous pouvons déterminer la position et l'orientation de chaque lien du robot. Nous attachons les coordonnées de chaque lien et déterminons sa configuration dans les liens voisins suivant la méthode de mouvement rigide. Une telle analyse s'appelle la cinématique directe.[15]

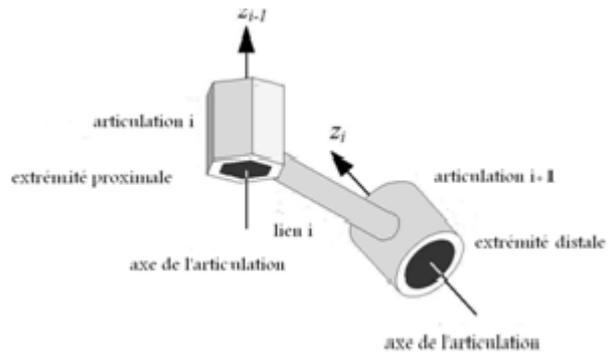


Figure 4.1 : le lien i avec son articulation initiale i-1 et son articulation finale i. [15]

4-3-1 Les paramètres de (D-H) du robot SCARA choisi :

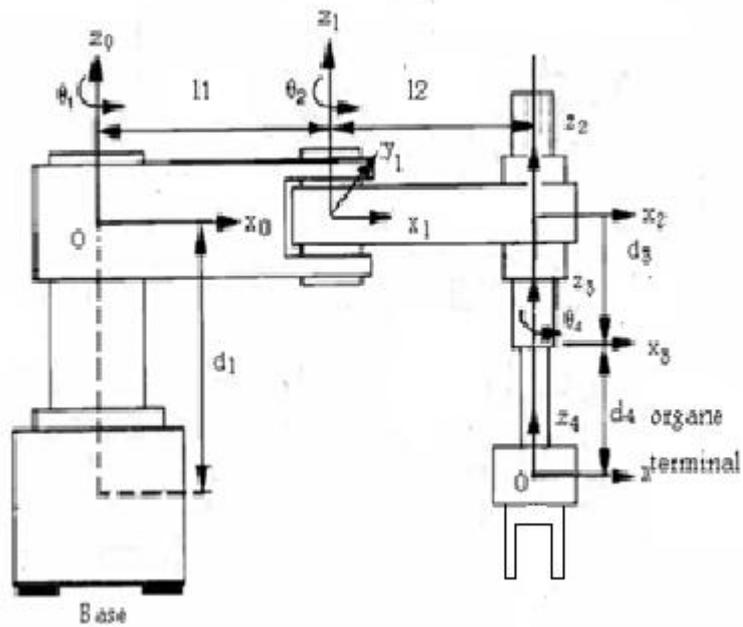


Figure 4.2 : Schéma du robot de type SCARA. [15]

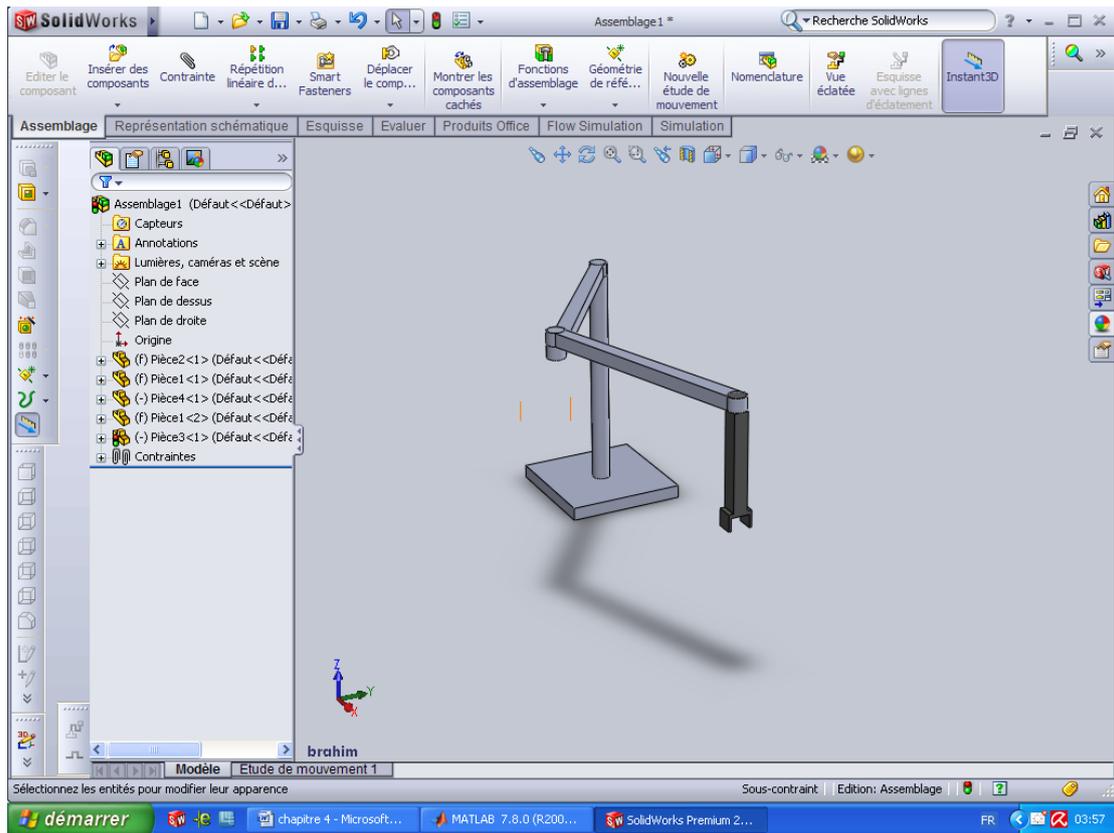


Figure 4.3 : Le robot SCARA schématisé à l'aide logiciel Solidworks.

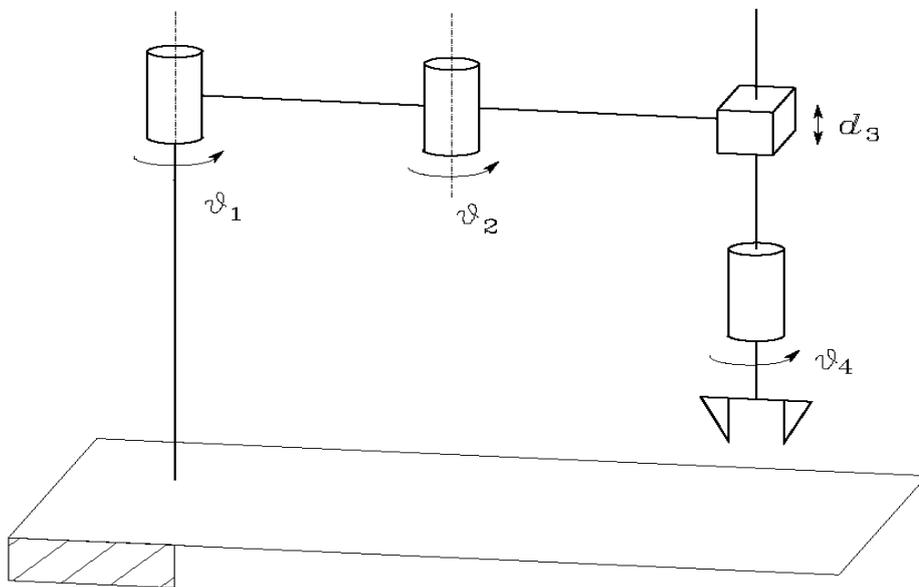


Figure 4.4 : Schéma cinématique du robot SCARA. [15]

Link	a_i	α_i	d_i	θ_i
1	l_1	0	0	θ_1^*
2	l_2	0	0	θ_2^*
3	0	0	d_3^*	0
4	0	0	d_4^*	θ_4^*

* les articulations variables.

Tableau 4.3.1 : DH paramètres du robot SCARA.

4-3-2 Calcul de la matrice de transformation homogène par deux méthodes :

4-3-2-1 La première méthode :

Cette méthode consiste à faire tourner les bras autour de l'axe de rotation(z) et la projection suivant l'axe(x) et l'axe(y) comme montre le schéma suivant : [15]

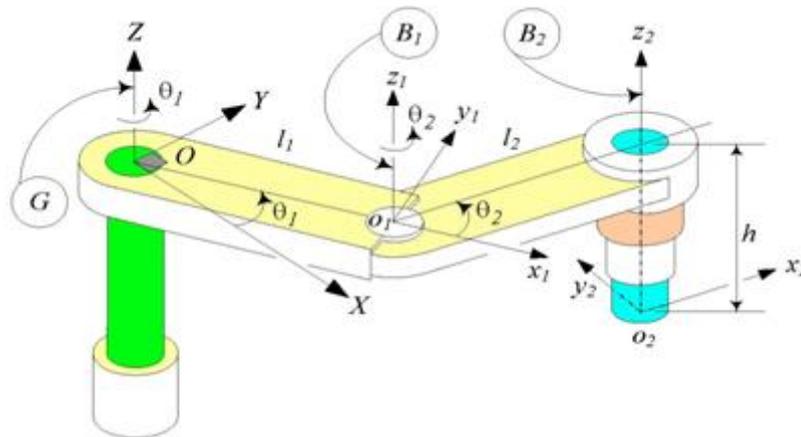


Figure 4.5 : la rotation des deux liens autour de son axe de rotation Z. [15]

On obtient les résultats suivant :

Sachant que :

$$\cos \theta_1 = c_1 \dots \dots \dots (4.1)$$

$$\sin \theta_1 = s_1 \dots \dots \dots (4.2)$$

$$\cos \theta_2 = c_2 \dots \dots \dots (4.3)$$

$$\sin \theta_2 = s_2 \dots \dots \dots (4.4)$$

$$\cos(\theta_1 + \theta_2) = c_{12} \dots \dots \dots (4.5)$$

$$\sin(\theta_1 + \theta_2) = s_{12} \dots \dots \dots (4.6)$$

$$T_1^0 = \begin{bmatrix} c_1 & -s_1 & 0 & l_1 c_1 \\ s_1 & c_1 & 0 & l_1 s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots (4.7)$$

$$T_2^1 = \begin{bmatrix} c_2 & -s_2 & 0 & l_2 c_2 \\ s_2 & c_2 & 0 & l_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots (4.8)$$

$$T_3^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots (4.9)$$

$$T_4^3 = \begin{bmatrix} c_4 & -s_4 & 0 & 0 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & -d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots (4.10)$$

Les équations du modèle cinématique directe sont données donc :

$$T_4^0 = T_1^0 T_2^1 T_3^2 T_4^3 \dots\dots\dots (4.11)$$

Pour L'obtention de la matrice T_4^0 voir (programme *appendice*)

$$T_4^0 = \begin{bmatrix} c_{124} & -s_{124} & 0 & l_1 c_1 + l_2 c_{12} \\ s_{124} & c_{124} & 0 & l_1 s_1 + l_2 s_{12} \\ 0 & 0 & 1 & -d_3 - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots (4.12)$$

4-3-2-2 La deuxième méthode :

Consiste à supposer que la position initiale du robot SCARA est quand les deux premiers bras sont dans le prolongement de l'axe x et toutes les articulations se faisant par rapport à l'axe Z, En supposant l'origine pour être sur la première articulation, la rotation est donné par

$$A_1(\theta_1) = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots (4.13)$$

Pour la deuxième rotation l'axe x est décalé selon la direction de x par la distance l_1 :

La rotation est donné par :

$$A_2(\theta_2) = \begin{bmatrix} c_2 & -s_2 & 0 & l_1(1-c_2) \\ s_2 & c_2 & 0 & -l_1s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots (4.14)$$

Même chose pour la troisieme rotation :

$$A_3(\theta_3) = \begin{bmatrix} c_4 & -s_4 & 0 & (l_1+l_2)(1-c_4) \\ s_4 & c_4 & 0 & -(l_1+l_2)s_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots (4.15)$$

L'articulation finale est prismatique le déplacement se fait selon l'axe Z :

$$A_4(d_3) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d_3-d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots (4.16)$$

$$T_4^0 = A_1(\theta_1)A_2(\theta_2)A_3(\theta_3)A_4(d_3) \dots\dots\dots (4.17)$$

$$T_4^0 = \begin{bmatrix} c_{124} & -s_{124} & 0 & k_x \\ s_{124} & c_{124} & 0 & k_y \\ 0 & 0 & 1 & k_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots (4.18)$$

$$k_x = l_1c_1 + l_2c_{12} - (l_1+l_2)c_{124} \dots\dots\dots (4.19)$$

$$k_y = l_1s_1 + l_2s_{12} - (l_1+l_2)s_{124} \dots\dots\dots (4.20)$$

$$k_z = -d_3 - d_4 \dots\dots\dots (4.21)$$

Le point $(l_1+l_2, 0, 0)^T$ après le mouvement nous donne $(x, y, z)^T$ donné par :

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{bmatrix} c_{124} & -s_{124} & 0 & k_x \\ s_{124} & c_{124} & 0 & k_y \\ 0 & 0 & 1 & k_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} l_1+l_2 \\ 0 \\ 0 \\ 1 \end{pmatrix} \dots\dots\dots (4.22)$$

$$x = l_1c_1 + l_2c_{12} \dots\dots\dots (4.23)$$

$$y = l_1s_1 + l_2s_{12} \dots\dots\dots (4.24)$$

$$z = -d_3 - d_4 \dots\dots\dots (4.25)$$

La matrice de transformation homogène s'écrit sous la forme :

$$T_4^0 = \begin{bmatrix} c_{124} & -s_{124} & 0 & l_1c_1 + l_2c_{12} \\ s_{124} & c_{124} & 0 & l_1s_1 + l_2s_{12} \\ 0 & 0 & 1 & -d_3 - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots (4.26)$$

4-4 Le modèle cinématique inverse :

Le problème de la cinématique inverse consiste en la détermination des variables articulaires correspondant à une position et à une orientation données de l'organe terminal. La solution à ce problème est d'importance fondamentale afin de transformer les caractéristiques de mouvement, assigné à l'organe terminal dans l'espace opérationnel, en mouvements d'espace communs correspondants, cela permet l'exécution du mouvement désiré.

Le problème de la cinématique inverse est beaucoup plus compliqué pour les raisons suivantes : [34]

Les équations à résoudre sont en général non linéaires, et il n'est pas ainsi toujours possible de trouver une solution exacte :

- Des solutions multiples peuvent exister.
- Les solutions infinies peuvent exister, par exemple, dans le cas d'un manipulateur cinématiquement redondant.
- Comme il ne pourrait y avoir aucune solution admissible, en raison notamment de la structure cinématique du manipulateur.

L'existence des solutions est garantie seulement si la position et l'orientation données de l'organe terminal appartiennent à la zone de travail adroite de manipulateur. [42]

4-4-1 Le problème général de la cinématique inverse :

Le problème général de la cinématique inverse peut être énoncé comme suite : Etant donné une transformation homogène de 4x4 :

$$T_H^R = \begin{bmatrix} R & O \\ 0 & 1 \end{bmatrix} \dots\dots\dots (4.27)$$

Trouver les solutions possibles de cette équation :

$$T_n^0(q_1, \dots, q_n) = T_H^R \dots\dots\dots (4.28)$$

On a :

$$T_n^0(q_1, \dots, q_n) = A_1(q_1) \dots\dots\dots A_n(q_n) \dots\dots\dots (4.29)$$

Sachant que H représente la position et l'orientation désirées de l'organe terminal le problème est de trouver les valeurs des articulations (q_1, \dots, q_n) de sorte que la relation

$$T_n^0(q_1, \dots, q_n) = T_H^R \text{ Soit vérifiée}$$

4-4-2 Détermination des variables articulaires du robot SCARA :

4-4-2-1 La position désirée :

La forme générale de la matrice de la position désirée est sous la forme [31] :

$$T_H^R = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \dots\dots\dots (4.30)$$

4-4-2-2 Calcul de la variable articulaire θ_2 :

D'après la définition du modèle cinématique inverse [41]

$$T_n^0(q_1, \dots, q_n) = T_H^R$$

On obtient :

$$p_x = l_1 c_1 + l_2 c_{12} \dots \dots \dots (4.31)$$

$$p_y = l_1 s_1 + l_2 s_{12} \dots \dots \dots (4.32)$$

$$p_x^2 = l_1^2 c_1^2 + l_2^2 c_{12}^2 + 2l_1 l_2 c_1 c_{12} \dots \dots \dots (4.33)$$

$$p_y^2 = l_1^2 s_1^2 + l_2^2 s_{12}^2 + 2l_1 l_2 s_1 s_{12} \dots \dots \dots (4.34)$$

$$p_x^2 + p_y^2 = l_1^2 + l_2^2 + 2l_1 l_2 (c_1 c_{12} + s_1 s_{12}) \dots \dots \dots (4.35)$$

Sachant que :

$$c_1^2 + s_1^2 = 1 \dots \dots \dots (4.36)$$

$$c_{12}^2 + s_{12}^2 = 1 \dots \dots \dots (4.37)$$

On utilise les relations trigonométriques :

$$c_{12} = c_1 c_2 - s_1 s_2 \dots \dots \dots (4.38)$$

$$s_{12} = s_1 c_2 + c_1 s_2 \dots \dots \dots (4.39)$$

On remplace (4.38) et (4.39) dans (4.35) on obtient :

$$p_x^2 + p_y^2 = l_1^2 + l_2^2 + 2l_1 l_2 c_2 \dots \dots \dots (4.40)$$

On peut tirer c_2 :

$$c_2 = \frac{1}{2l_1 l_2} (p_x^2 + p_y^2 - l_1^2 - l_2^2) \dots \dots \dots (4.41)$$

$$s_2 = \pm \sqrt{1 - c_2^2} \dots \dots \dots (4.42)$$

$$\theta_2 = \pm a \tan \frac{s_2}{c_2} \dots \dots \dots (4.43)$$

4-4-2-3 Calcul de la variable articulaire θ_1 par deux méthodes :

4-4-2-3-1 La première méthode :

En arrangeant les deux équations (4.31) et (4.32) et on applique les équations (4.38) et (4.39) on obtient :

$$p_x = (l_1 + l_2 c_2) c_1 - l_2 s_2 s_1 \dots \dots \dots (4.44)$$

$$p_y = l_2 s_2 c_1 + (l_1 + l_2 c_2) s_1 \dots \dots \dots (4.45)$$

On utilise la méthode de Kramer on obtient :

$$\Delta = \begin{bmatrix} l_1 + l_2 c_2 & -l_2 s_2 \\ l_2 s_2 & l_1 + l_2 c_2 \end{bmatrix} = (l_1 + l_2 c_2)^2 + (l_2 s_2)^2 \dots \dots \dots (4.46)$$

Et on a :

$$p_x^2 + p_y^2 = (l_1 + l_2 c_2)^2 + (l_2 s_2)^2 \dots\dots\dots (4.47)$$

On peut tirer :

$$\Delta s_1 = \begin{bmatrix} l_1 + l_2 c_2 & p_x \\ l_2 s_2 & p_y \end{bmatrix} \dots\dots\dots (4.48)$$

$$\Delta c_1 = \begin{bmatrix} p_x & -l_2 s_2 \\ p_y & l_1 + l_2 c_2 \end{bmatrix} \dots\dots\dots (4.49)$$

$$s_1 = \frac{\Delta s_1}{\Delta} = \frac{(l_1 + l_2 c_2)p_y - l_2 s_2 p_x}{p_x^2 + p_y^2} \dots\dots\dots (4.50)$$

$$c_1 = \frac{\Delta c_1}{\Delta} = \frac{(l_1 + l_2 c_2)p_x + l_2 s_2 p_y}{p_x^2 + p_y^2} \dots\dots\dots (4.51)$$

On peut tirer θ_1 :

On a $s_2 = \pm \sqrt{1 - c_2^2}$

$$\theta_1 = a \tan \frac{s_1}{c_1} = a \tan \frac{(l_1 + l_2 c_2)p_y \mp l_2 s_2 p_x}{(l_1 + l_2 c_2)p_x \pm l_2 s_2 p_y} \dots\dots\dots (4.52)$$

4-4-2-3-2 La deuxième méthode :

$$p_x c_1 + p_y s_1 = l_1 + l_2 c_2 \dots\dots\dots (4.53)$$

$$-p_x s_1 + p_y c_1 = \pm l_2 s_2 \dots\dots\dots (4.54)$$

D'après (4.53) on peut tirer :

$$c_1 = \frac{1}{p_x} (l_1 + l_2 c_2 - p_y s_1) \dots\dots\dots (4.55)$$

On remplace (4.55) dans (4.54) :

$$s_1 = \frac{\mp p_x l_2 s_2 + p_y (l_1 + l_2 c_2)}{(p_x^2 + p_y^2)} \dots\dots\dots (4.56)$$

$$c_1 = \frac{p_x (l_1 + l_2 c_2) \pm p_y l_2 s_2}{(p_x^2 + p_y^2)} \dots\dots\dots (4.57)$$

$$\theta_1 = a \tan \frac{\mp p_x l_2 s_2 + p_y (l_1 + l_2 c_2)}{p_x (l_1 + l_2 c_2) \pm p_y l_2 s_2} \dots\dots\dots (4.58)$$

L'équation (4.58) est vérifiée avec l'équation (4.52).

Le signe (\pm) indique coude haut et coude bas (elbow up and elbow down).

4-4-3 Equation du coude haut (elbow up) :

$$\theta_2 = -a \tan \frac{s_2}{c_2} \dots\dots\dots (4.59)$$

$$\theta_1 = a \tan \frac{+p_x l_2 s_2 + p_y (l_1 + l_2 c_2)}{p_x (l_1 + l_2 c_2) - (p_y l_2 s_2)} \dots\dots\dots (4.60)$$

4-4-4 Equation du coude bas (elbow down) :

$$\theta_2 = +a \tan \frac{s_2}{c_2} \dots\dots\dots (4.61)$$

$$\theta_1 = a \tan \frac{-p_x l_2 s_2 + p_y (l_1 + l_2 c_2)}{p_x (l_1 + l_2 c_2) + (p_y l_2 s_2)} \dots\dots\dots (4.62)$$

4-4-5 Vérification des résultats obtenus :

Pour une position désirée $p_x = 0.86$ et $p_y = 1.5$ on doit vérifier l'équation (4.31) et (4.32) :

avec $l_1 = l_2 = 1m$.

4-4-5-1 Application numérique :

Pour coude haut (elbow up) :

On applique l'équation (4.41) et (4.42) dans (4.59) et (4.60)

$$\theta_1 = 90^\circ$$

$$\theta_2 = -60^\circ$$

D'où l'équation (4.31) et (4.32) vérifié.

Pour coude bas (elbow down) :

On applique l'équation (4.41) et (4.42) dans (4.61) et (4.62)

$$\theta_1 = 30^\circ$$

$$\theta_2 = 60^\circ$$

D'où l'équation (4.31) et (4.32) vérifié.

Et d'après la matrice de transformation homogène (4.12) on tire :

$$d_3 = -p_z - d_4 \dots\dots\dots (4.63)$$

On a :

$$\theta_3 = 0^\circ \dots\dots\dots (4.64)$$

4-4-6 Calcul l'orientation de l'organe terminal θ_4 :

L'équation finale qui représente la position désirée du robot :

$$T_H^R = T_1^0 T_2^1 T_3^2 T_4^3 = T_4^0 \dots\dots\dots (4.65)$$

On doit mettre l'équation (4.10) sous la forme :

$$T_4^3 = (T_3^2)^{-1} (T_2^1)^{-1} (T_1^0)^{-1} (T_H^R)^{-1} \dots\dots\dots (4.66)$$

Inverse des matrices a été fait par logiciel Matlab (calcul formel) (voir appendice)

$$T_4^3 = \begin{bmatrix} n_x c_{12} + n_y s_{12} & o_x c_{12} + o_y s_{12} & a_x c_{12} + a_y s_{12} & p_x c_{12} + p_y s_{12} - l_1 c_2 - l_2 \\ -n_x s_{12} + n_y c_{12} & -o_x s_{12} + o_y c_{12} & -a_x s_{12} + a_y c_{12} & -p_x s_{12} + p_y c_{12} - l_1 s_2 \\ n_z & o_z & a_z & p_z + d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots (4.67)$$

Par identification (4.67) avec (4.10) on obtient :

$$c_4 = n_x c_{12} + n_y s_{12} \dots\dots\dots (4.68)$$

$$s_4 = -n_x s_{12} + n_y c_{12} \dots\dots\dots (4.69)$$

$$\theta_4 = a \tan \frac{-n_x s_{12} + n_y c_{12}}{n_x c_{12} + n_y s_{12}} \dots\dots\dots (4.70)$$

4-4-6-1 Calcul de la vitesse linéaire :

$$\dot{P}_x = -l_1 s_1 \dot{\theta}_1 - l_2 s_{12} \dot{\theta}_{12} \dots\dots\dots (4.71)$$

$$\dot{P}_y = l_1 c_1 \dot{\theta}_1 + l_2 c_{12} \dot{\theta}_{12} \dots\dots\dots (4.72)$$

On utilise la méthode de Kramer pour tirer $\dot{\theta}_1$:

$$\dot{\theta}_1 = \frac{\dot{P}_x c_{12} + \dot{P}_y s_{12}}{l_2 s_2} \dots\dots\dots (4.73)$$

$$\dot{\theta}_2 = \frac{-\dot{P}_x (l_1 c_1 + l_2 c_{12}) - \dot{P}_y (l_1 s_1 + l_2 s_{12})}{l_1 l_2 s_2} \dots\dots\dots (4.74)$$

4-4-6-2 La vitesse de translation :

$$\dot{d}_3 + \dot{d}_4 = -\dot{P}_z \dots\dots\dots (4.75)$$

$$\dot{\theta}_4 = \frac{-\dot{n}_x s_{12} + \dot{n}_y c_{12} - (-n_x c_{12} + n_y s_{12}) \dot{\theta}_{12}}{c_4} \dots\dots\dots (4.76)$$

4-4-6-3 Calcul de l'accélération :

$$\ddot{\theta}_1 = \frac{(-\dot{P}_x s_{12} + \dot{P}_y c_{12}) \dot{\theta}_{12} + (\ddot{P}_x c_{12} + \ddot{P}_y s_{12}) - l_1 c_2 \dot{\theta}_1 \dot{\theta}_2}{l_1 s_2} \dots\dots\dots (4.77)$$

$$\ddot{\theta}_2 = -\frac{[(\ddot{P}_y s_1 - \ddot{P}_x c_1) l_1 + (\ddot{P}_y s_{12} + \ddot{P}_x c_{12}) l_2 + (\dot{P}_y c_1 - \dot{P}_x s_1) l_1 \dot{\theta}_1 + (\dot{P}_x s_{12} + \dot{P}_y c_{12}) l_2 \dot{\theta}_{12} + l_1 l_2 c_2 \dot{\theta}_2^2]}{l_1 l_2 s_2} \dots\dots\dots (4.78)$$

$$\ddot{d}_3 + \ddot{d}_4 = -\ddot{P}_z \dots\dots\dots (4.79)$$

$$\ddot{\theta}_4 = \frac{-\ddot{n}_y c_{12} - \ddot{n}_y s_{12} - 2(\dot{n}_y s_{12} + \dot{n}_x c_{12})\dot{\theta}_{12} - (n_y c_{12} - n_x s_{12})\dot{\theta}_{12}^2 - (n_y s_{12} + n_x c_{12})\ddot{\theta}_{12} + s_4 \dot{\theta}_4^2}{c_4} \dots\dots\dots(4.80)$$

4-5 Application :

Cette application va porter sur la comparaison énergétique lors du mouvement du robot SCARA avec $l_1=1$ et $l_2=1$, pour les deux postures. A partir de l'étude du mouvement des deux postures « coude bas » et « coude haut » pour une même nature de la trajectoire, en atteignant une même position désirée au cours du même intervalle de temps de 1s.

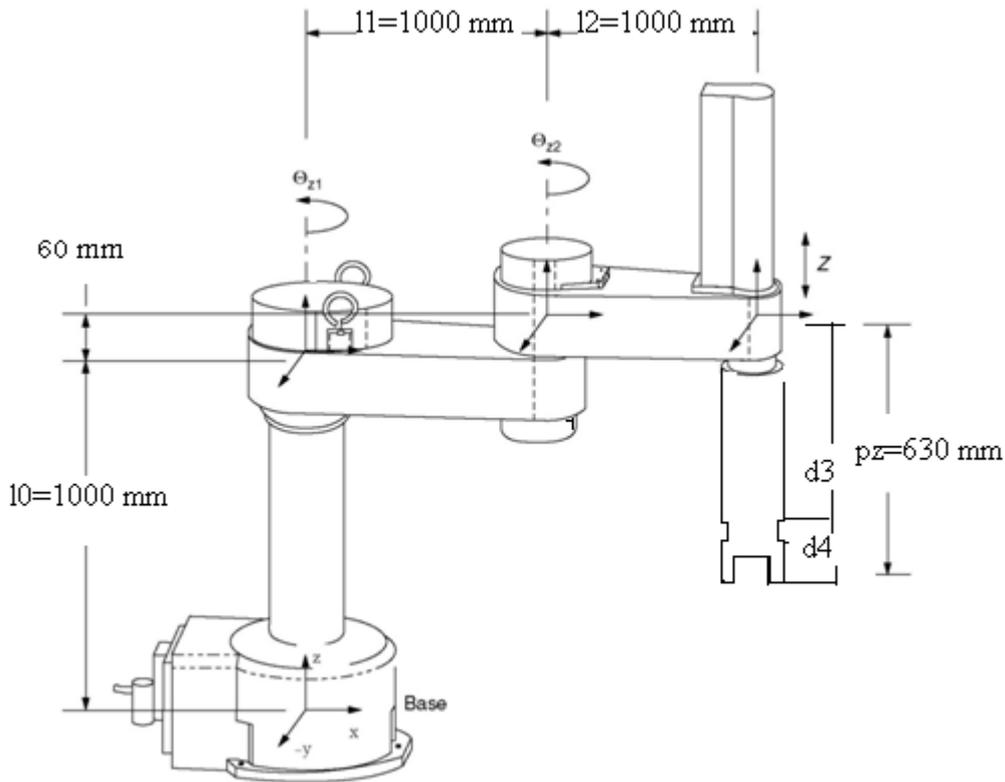


Figure 4.6 : le schéma du robot SCARA dans cette étude.

4-5-1 La position initiale (home position) :

La position initiale a été obtenue à partir de la matrice de transformation homogène (4.12) quand $\theta_1 = 0^\circ$ et $\theta_2 = 0^\circ$. come montre le schéma suivant :

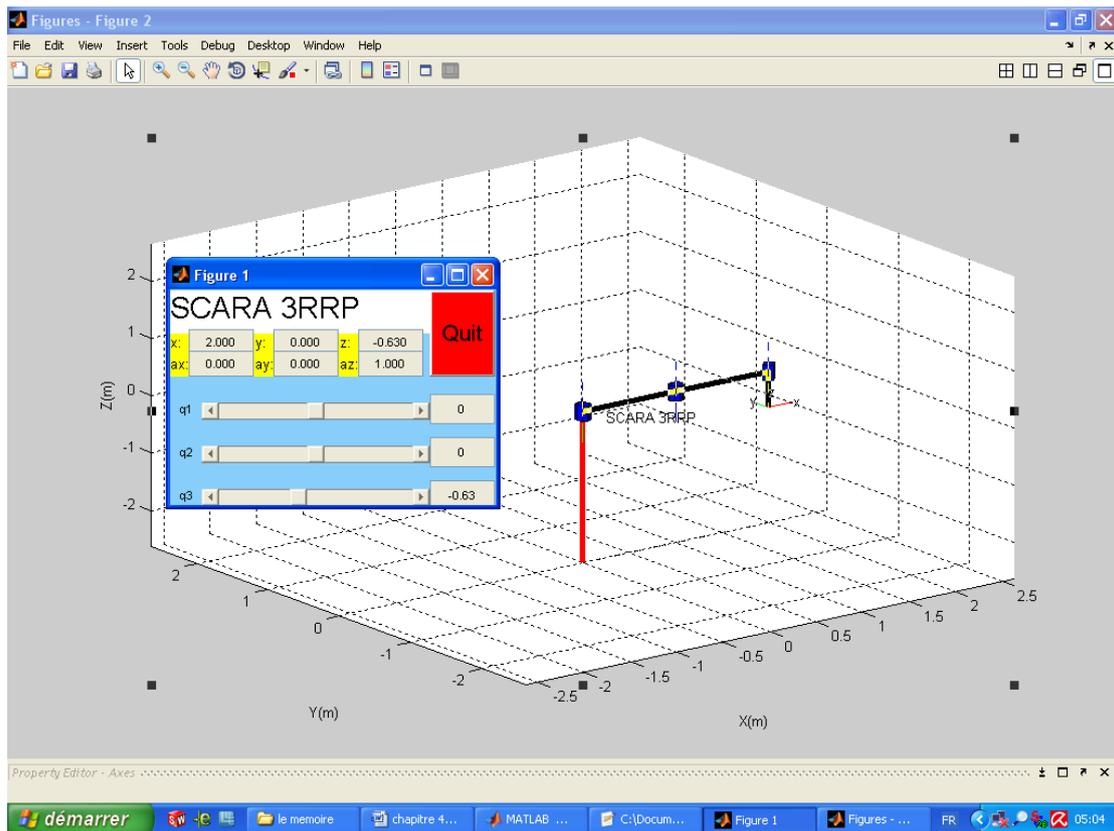


Figure 4.7 : La position initiale du robot SCARA pour les deux postures : logiciel Matlab/simulink.

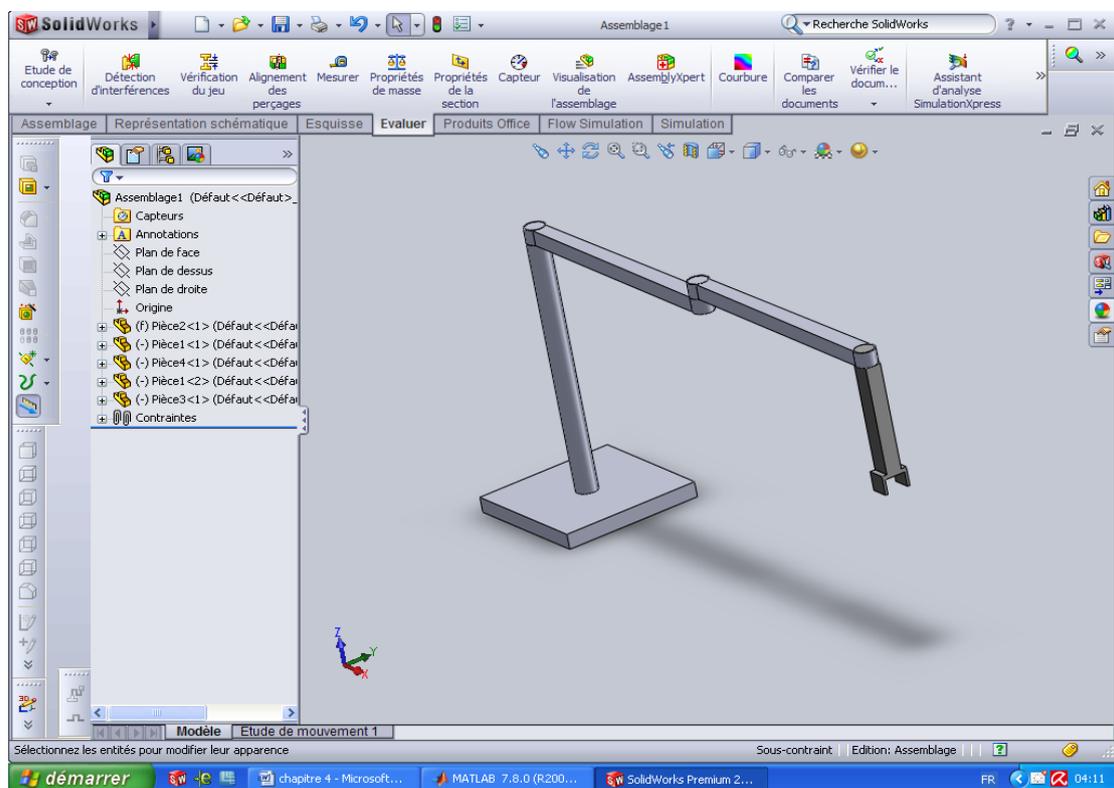


Figure 4.8 : La position initiale du robot SCARA pour les deux postures : logiciel Solidworks.

En partant de la position initiale :

$$p_x = 2 \quad p_y = 0 \quad p_z = -0.63$$

Vers la position désirée :

$$p_x = 0.86 \quad p_y = 1.5 \quad p_z = -0.63$$

4-5-2 L'équation du mouvement pour coude bas (elbow down) pour la position initiale et désirée :

L'équation du mouvement pour le chemin emprunté « coude bas » pour atteindre la position désirée est donnée par les relations : [45]

$$\theta_1 = (10t^4 + 10t^3 + 10t^2)^\circ \dots\dots\dots(4.81)$$

$$\theta_2 = (50t^4 + 10t^3)^\circ \dots\dots\dots (4.82)$$

Vérification du polynôme par deux méthodes :

La première méthode :

A t=0s

$$\theta_1 = 0^\circ$$

$$\theta_2 = 0^\circ$$

On remplace $\theta_1 = 0^\circ$ et $\theta_2 = 0^\circ$ dans (4.31) et (4.32) on trouve :

$$p_x = 2 \text{ et } p_y = 0 \quad p_z = -0.63 \text{ (la position initiale)}$$

La deuxième méthode :

Ou bien on remplace $p_x = 2$ et $p_y = 0$ dans (4.41) (4.42) (4.61) et (4.62) on trouve :

$$\theta_1 = 0^\circ \text{ et } \theta_2 = 0^\circ \text{ (la position initiale).}$$

A t=1s.

La première méthode :

$$\theta_1 = 30^\circ \text{ et } \theta_2 = 60^\circ$$

On remplace $\theta_1 = 30^\circ$ et $\theta_2 = 60^\circ$ dans (4.31) et (4.32) on trouve :

$$. p_x = 0.86 \text{ et } p_y = 1.5 \quad p_z = -0.63 \text{ (la position désirée)}$$

La deuxième méthode :

Ou bien on remplace $p_x = 0.86$ et $p_y = 1.5$ dans les équations (4.41) (4.42) (4.61) et (4.62) on trouve :

$$\theta_1 = 30^\circ \text{ et } \theta_2 = 60^\circ . \text{ (La position désirée)}$$

4-5-3 L'équation du mouvement pour coude haut (elbow up) pour la position initiale et désirée :

L'équation du mouvement pour le chemin emprunté « coude bas » pour atteindre la position désirée est donnée par les relations : [45]

$$\theta_1 = (60t^4 + 20t^3 + 10t^2)^\circ \dots\dots\dots(4.83)$$

$$\theta_2 = (-50t^4 - 10t^3)^\circ \dots\dots\dots(4.84)$$

Vérification du polynôme par deux méthodes :

La première méthode :

A t=0s

$$\theta_1 = 0^\circ$$

$$\theta_2 = 0^\circ$$

On remplace $\theta_1 = 0^\circ$ et $\theta_2 = 0^\circ$ dans (4.31) et (4.32) on trouve :

$$p_x = 2 \text{ et } p_y = 0 \text{ } p_z = -0.63 \text{ (la position initiale)}$$

La deuxième méthode :

Ou bien on remplace $p_x = 2$ et $p_y = 0$ dans (4.41) (4.42) (4.59) et (4.60) on trouve :

$$\theta_1 = 0^\circ \text{ et } \theta_2 = 0^\circ \text{ (la position initiale).}$$

A t=1s.

La première méthode :

$$\theta_1 = 90^\circ \text{ et } \theta_2 = -60^\circ$$

On remplace $\theta_1 = 90^\circ$ et $\theta_2 = -60^\circ$ dans (4.31) et (4.32) on trouve :

$$p_x = 0.86 \text{ et } p_y = 1.5 \text{ } p_z = -0.63 \text{ (la position désirée)}$$

La deuxième méthode :

Ou bien on remplace $p_x = 0.86$ et $p_y = 1.5$ dans les équations (4.41) (4.42) (4.59) et (4.60)

on trouve :

$$\theta_1 = 90 \text{ et } \theta_2 = -60^\circ \text{ (la position désirée)}$$

Le choix des expressions de θ_1 et θ_2 doit satisfaire les conditions suivantes [45]:

- Dans l'intervalle de temps doit vérifier la position initiale et la position désirée.
- A chaque instant les deux postures (coude bas et coude haut) doivent avoir les mêmes coordonnées p_x p_y p_z
- Les deux postures seront atteintes par des trajectoires de même nature.
- Les deux trajectoires ont la même longueur.
- Le degré 4 du polynôme est généralement utilisé en robotique.

Remarque :

Dans la suite de notre travail on utilisera le terme « posture 1 » le chemin coude bas (elbow down) et « posture 2 » le chemin coude haut (elbow up).

4-5-4 Vérification de trajectoire :

On applique les relations (4.81), (4.82), (4.83) et (4.84) dans (4.31) et (4.32) on obtient le tableau suivant :

Temps (sec)	Posture 1				Posture2			
	Tita1 (deg)	Tita2 (deg)	p_x (m)	p_y (m)	Tita1 (deg)	Tita2 (deg)	p_x (m)	p_y (m)
0	0	0	2	0	0	0	2	0
0.1	0.11	0.015	1.99	0.004	0.126	-0.015	1.99	0.004
0.2	0.49	0.16	1.99	0.02	0.65	-0.16	1.99	0.02
0.3	1.25	0.67	2	0.05	1.92	-0.67	2	0.05
0.4	2.49	1.92	1.996	0.12	4.41	-1.92	1.996	0.12
0.5	4.37	4.37	1.98	0.21	8.75	-4.37	1.98	0.21
0.6	7.05	8.64	1.95	0.39	15.69	-8.67	1.95	0.39
0.7	10.73	15.43	1.88	0.62	26.16	-15.43	1.88	0.62
0.8	15.61	25.6	1.71	0.927	41.21	-25.6	1.71	0.927
0.9	20	40.09	1.44	1.20	59.04	-40.09	1.44	1.20
1	30	60	0.86	1.5	90	-60	0.86	1.5

Tableau 4.5.4 : la position des deux postures durant 1s.

Dans ce tableau sont données les différentes positions de l'organe terminal pour les deux postures, en fonction du temps. Ces positions sont calculées suivant les variations des angles articulaires ; on a remplacé les valeurs des relations (4.60), (4.61), (4.75) et (4.76) dans (4.31) et (4.32).

4-5-4-1 Vérification de la trajectoire à l'aide Solidworks :

Posture1 :

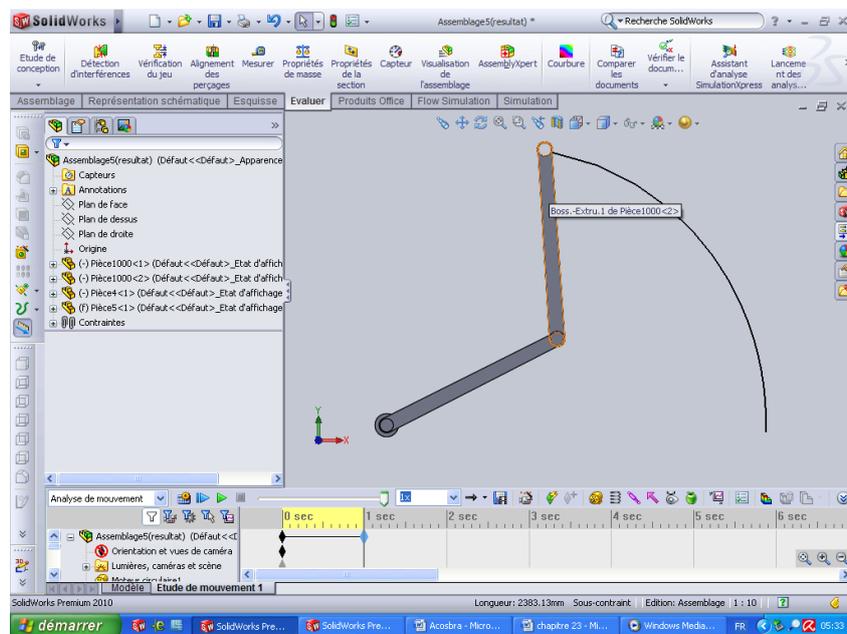


Figure 4.9 : La trajectoire obtenue lors du déplacement de la *posture1*.

Posture2 :

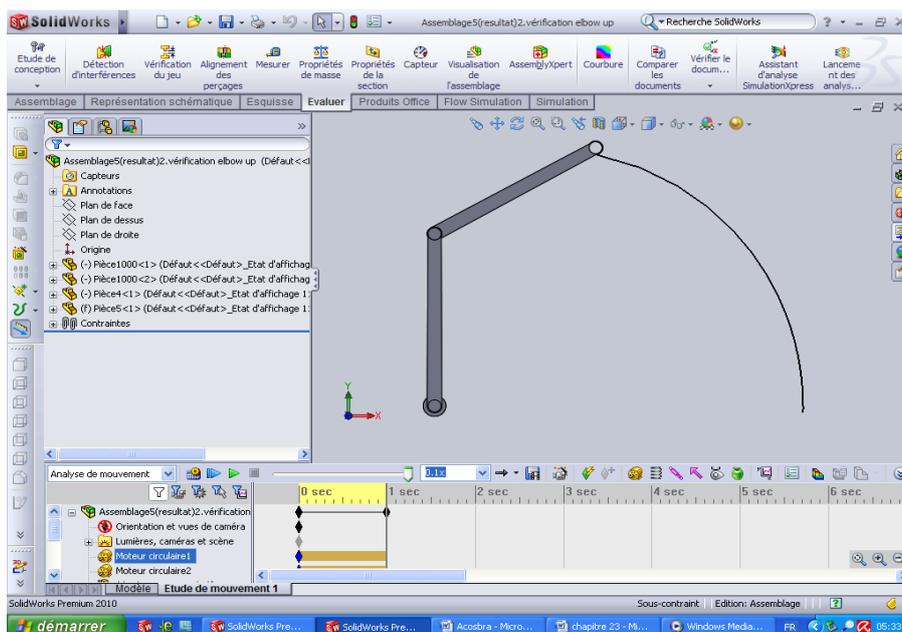


Figure 4.10 : La trajectoire obtenue lors du déplacement de la *posture2*.

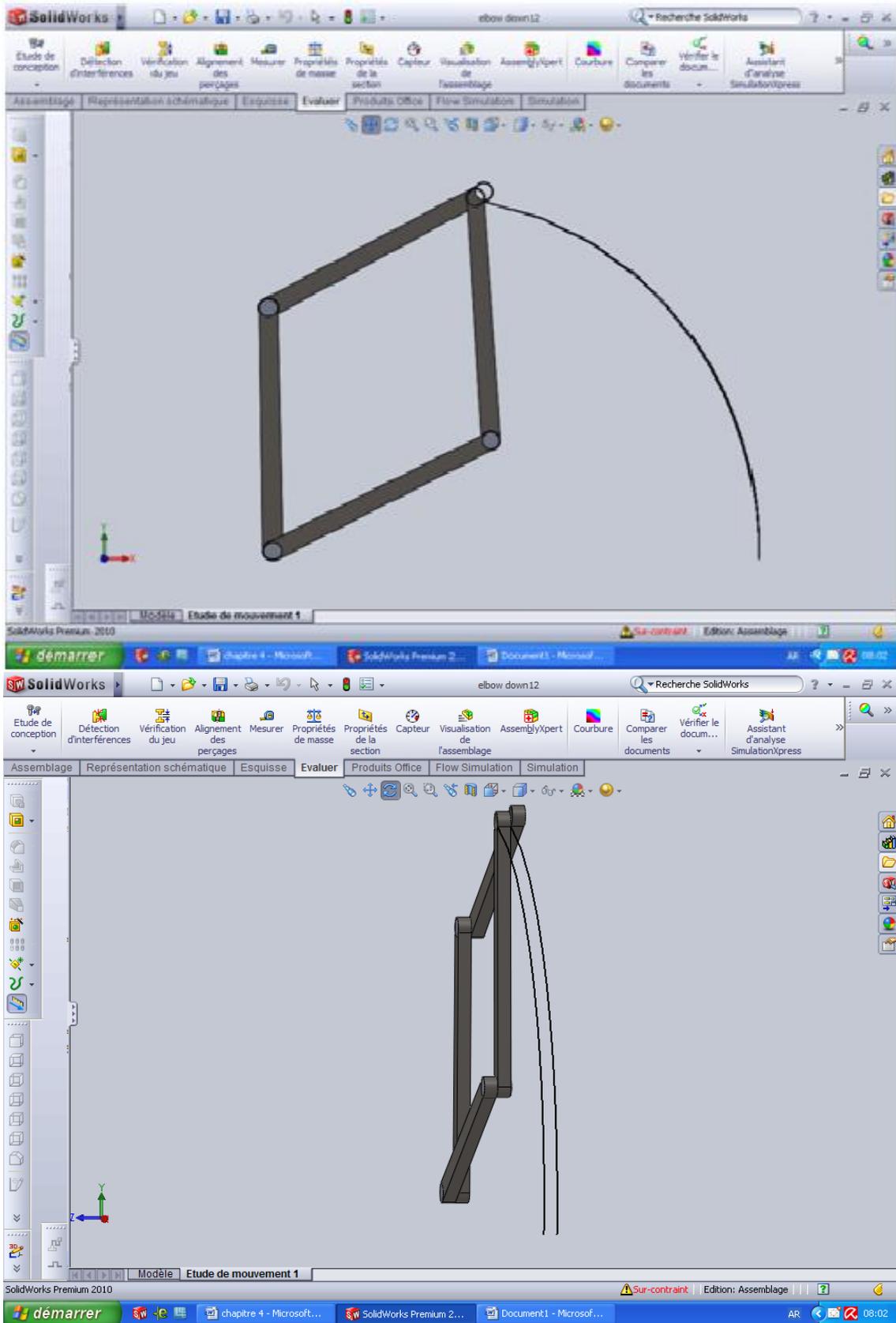


Figure 4.11 : Trajectoires obtenues lors du déplacement des deux postures en même temps.

4-5-4-2 Vérification de la trajectoire à l'aide du logiciel Matlab :

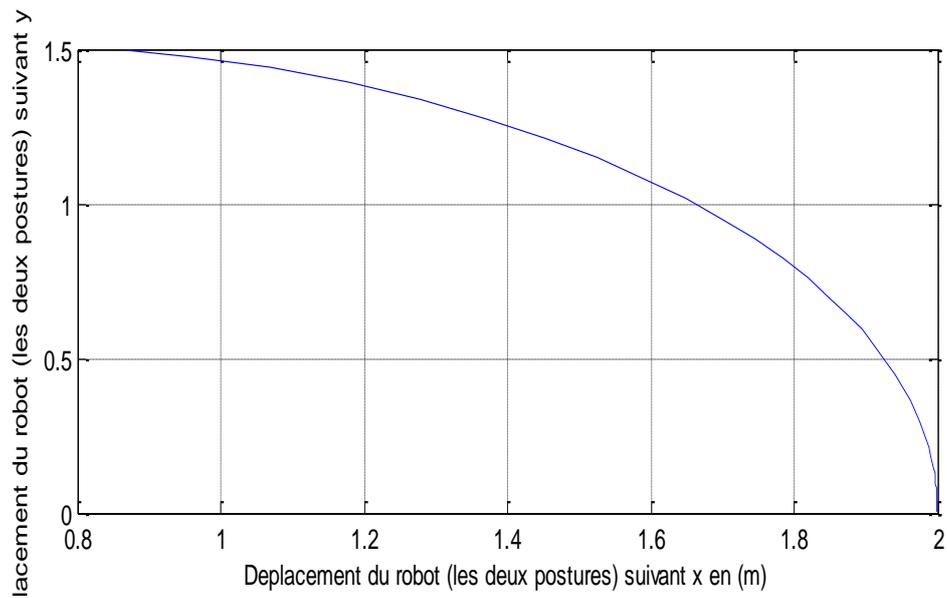


Figure 4.12 : la trajectoire obtenue lors du déplacement des deux postures.

4-5-5 : La variation des angles des deux postures :

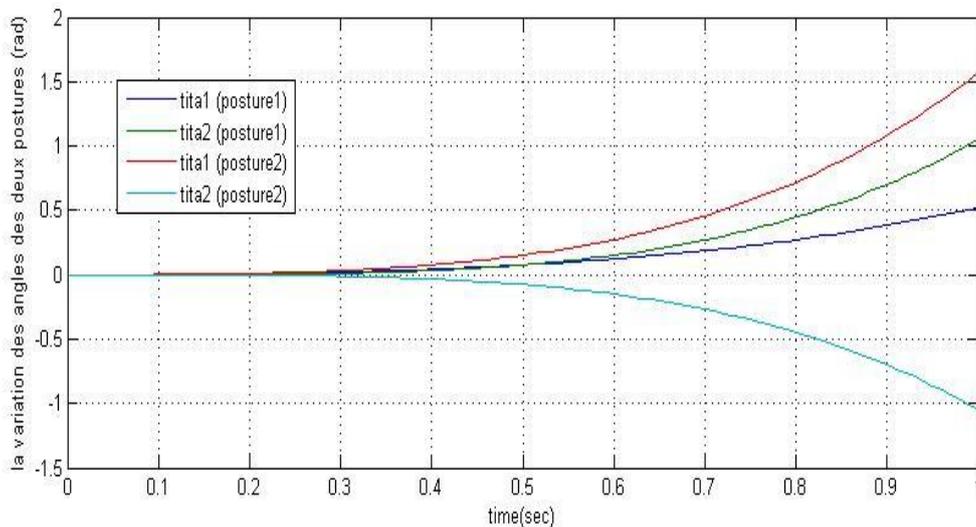


Figure 4.13: La variation des angles des deux postures.

4-5-6 Vérification des résultats en utilisant :

4-5-6-1 le logiciel Solidworks :

On va vérifier les résultats du tableau 4-5-4 en utilisant le logiciel Solidworks pour différentes valeurs de temps.

On prend par exemple les valeurs à $t=0.4s$, à $t=0.6s$ et $t=1s$:

A $t=0.4s$

Posture1 :

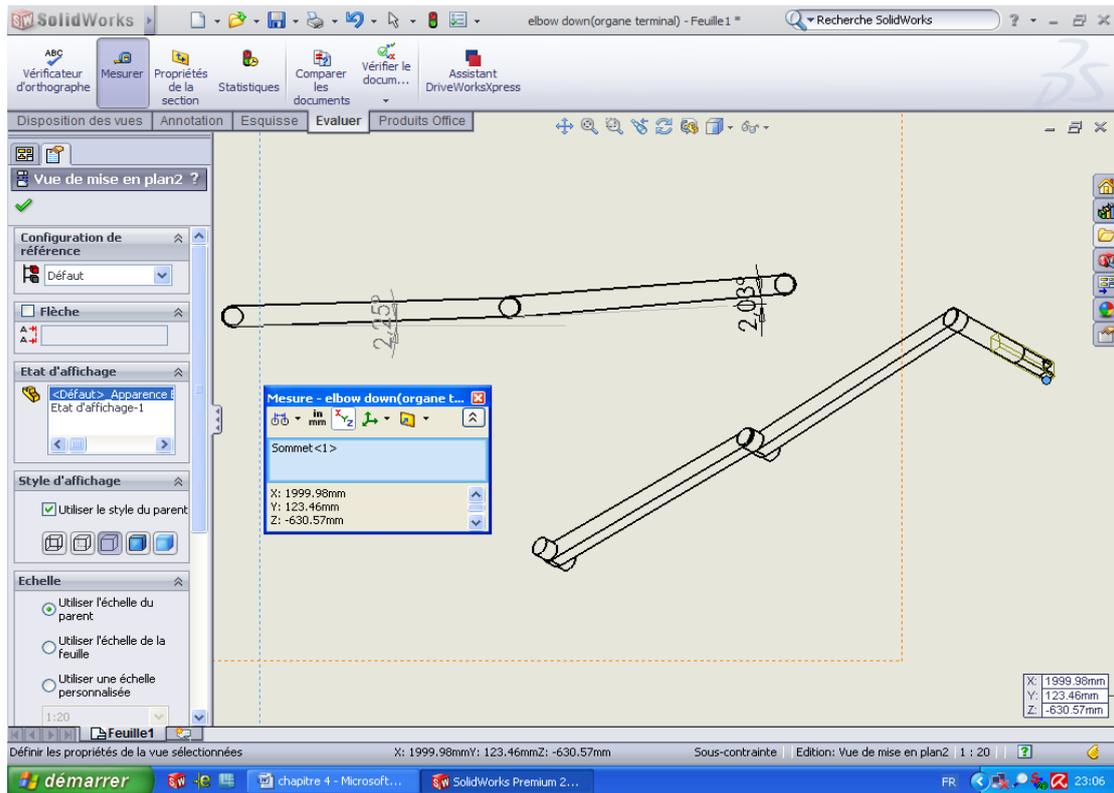


Figure 4.14 : La position de la posture1 à l' instant $t=0.4s$.

posture2 :

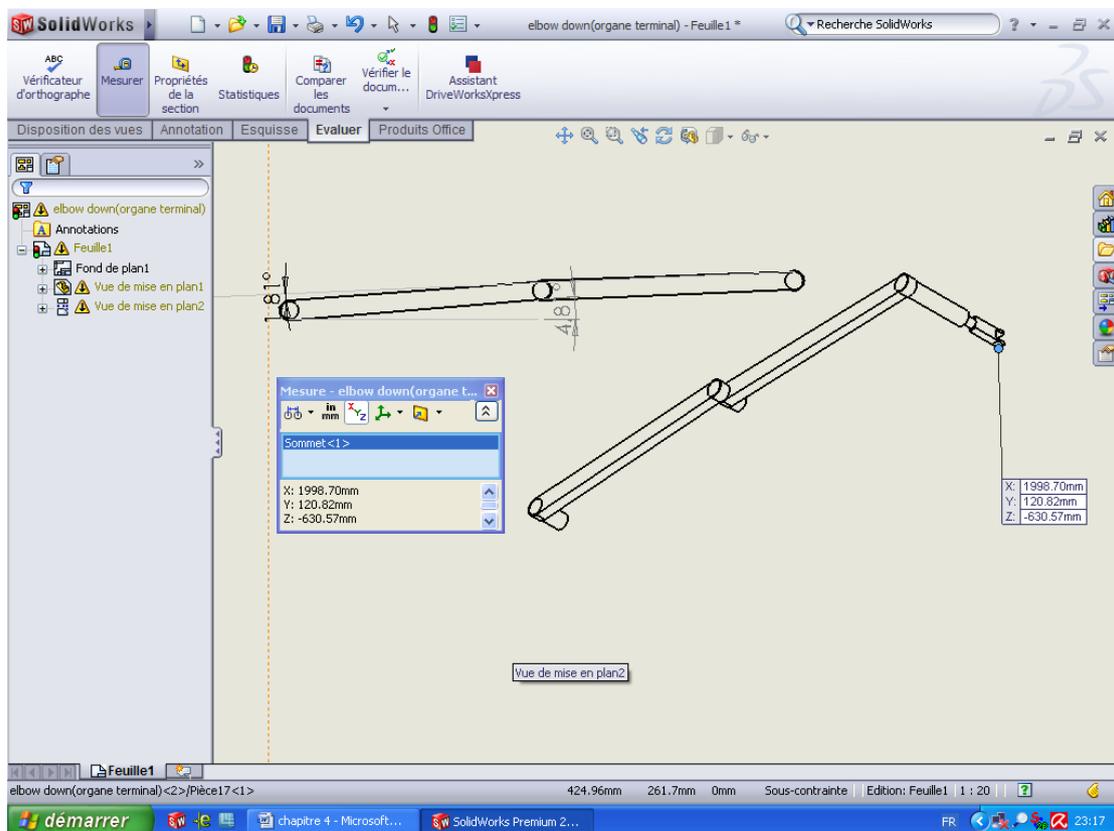


Figure 4.15 : La position de la posture 2 à l' instant $t=0.4s$.

A t=0.6s :
Posture1 :

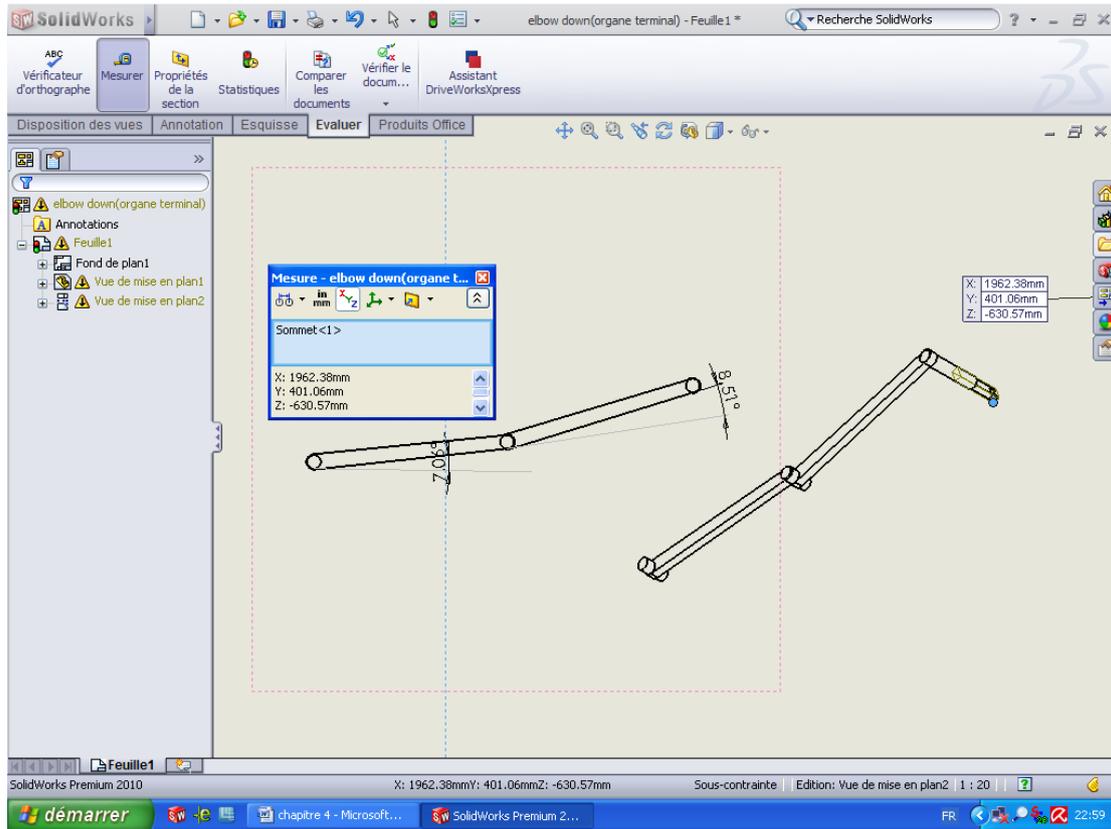


Figure 4.16: La position de la *posture1* à l' instant t=0.6s.

Posture2 :

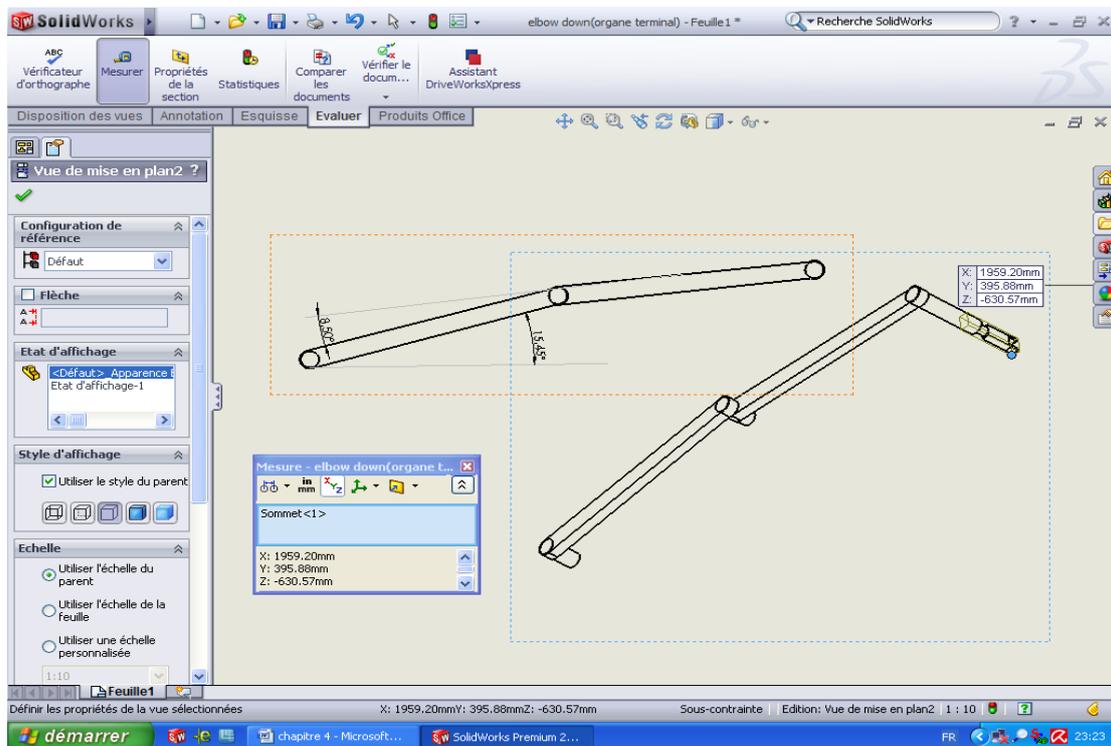


Figure 4.17: La position de la *posture2* à l' instant t=0.6s.

A t=1s

Posture1 :

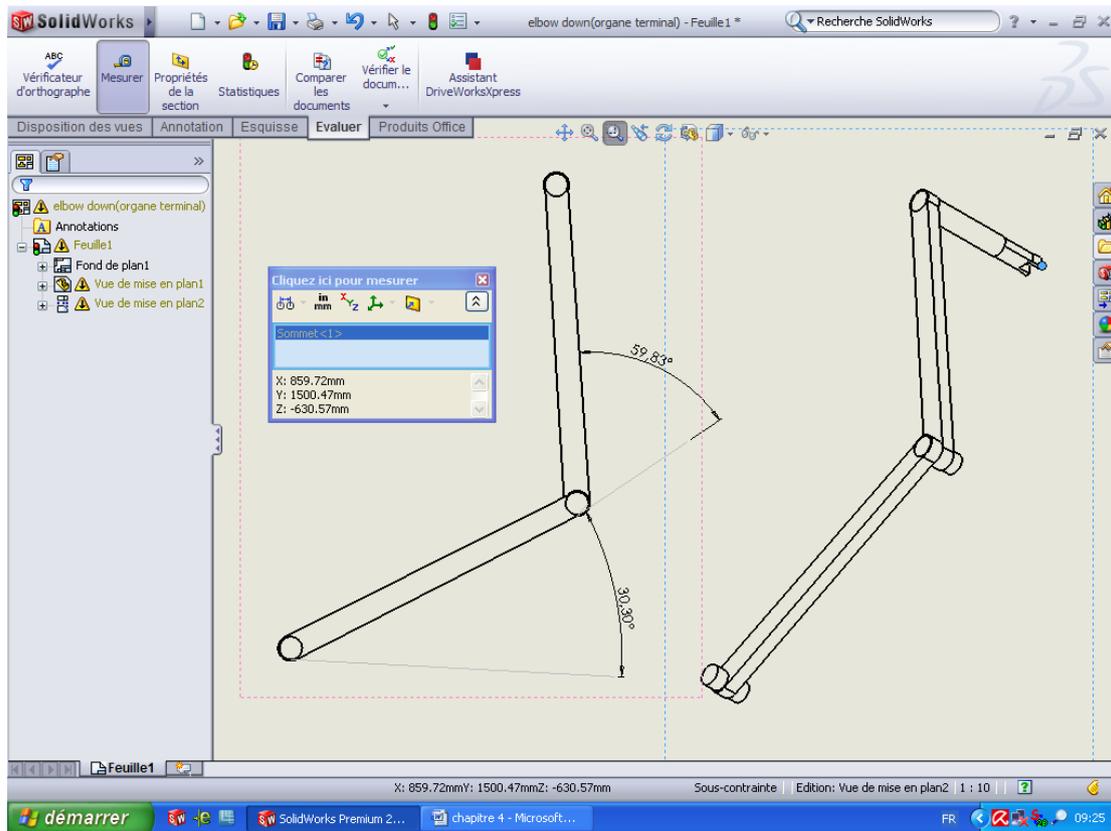


Figure 4.18 : La position de la *posture1* à l'instant t=1s.

Posture2 :

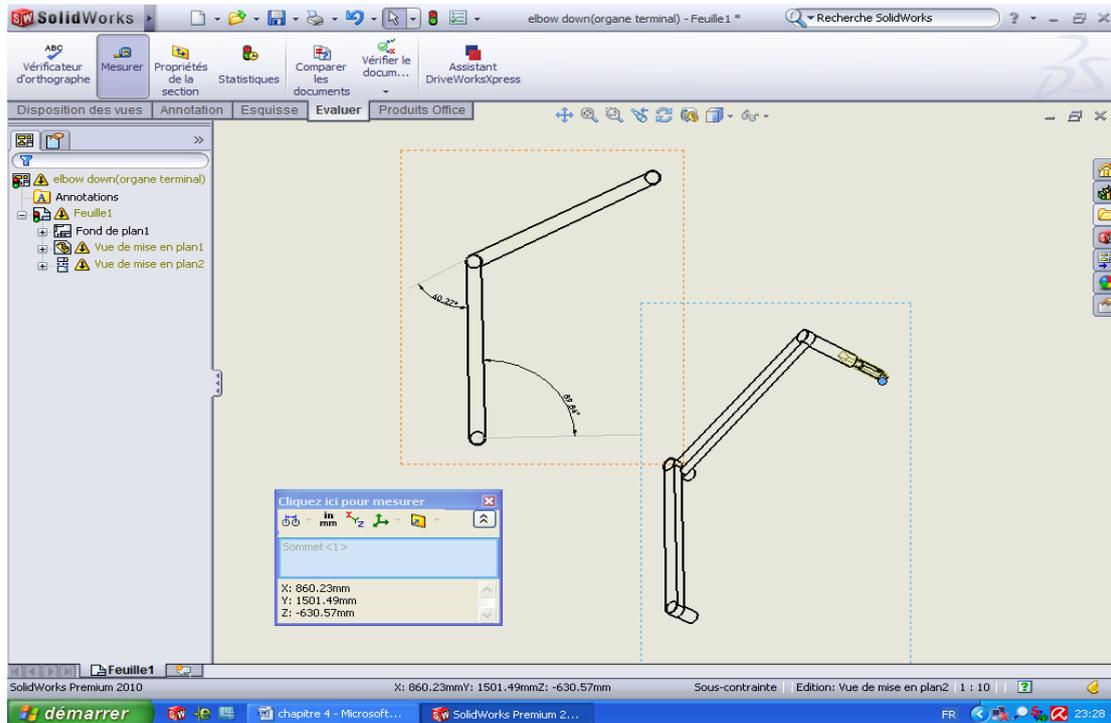


Figure 4.19 : La position de la *posture2* à l'instant t=1s.

4-5-6-2 MATLAB/simulink:

t=0.4s

Posture1:

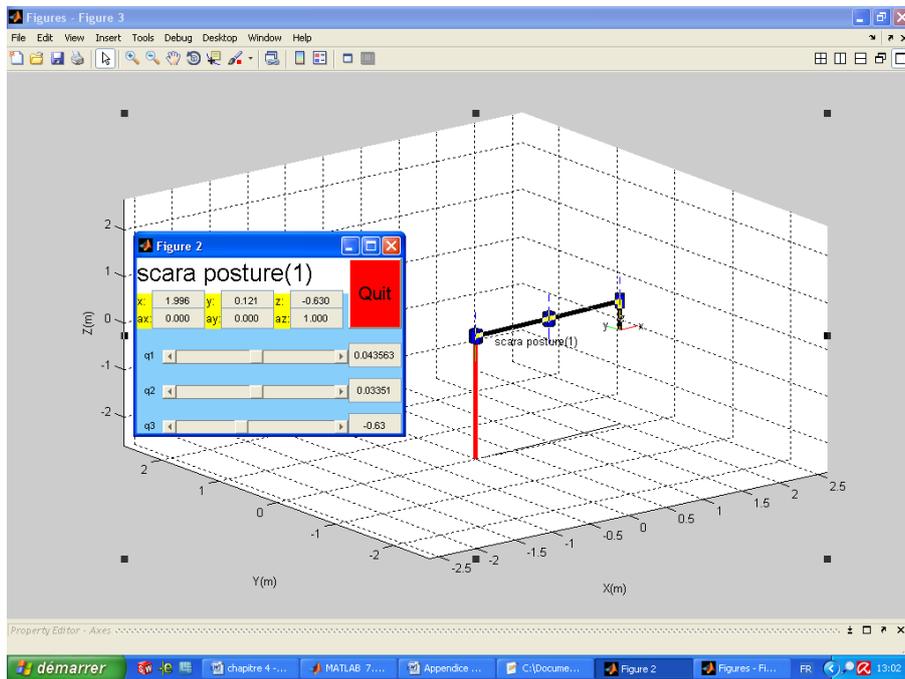


Figure 4.20 : La position de la *posture1* à l' instant t=0.4s.

Posture2

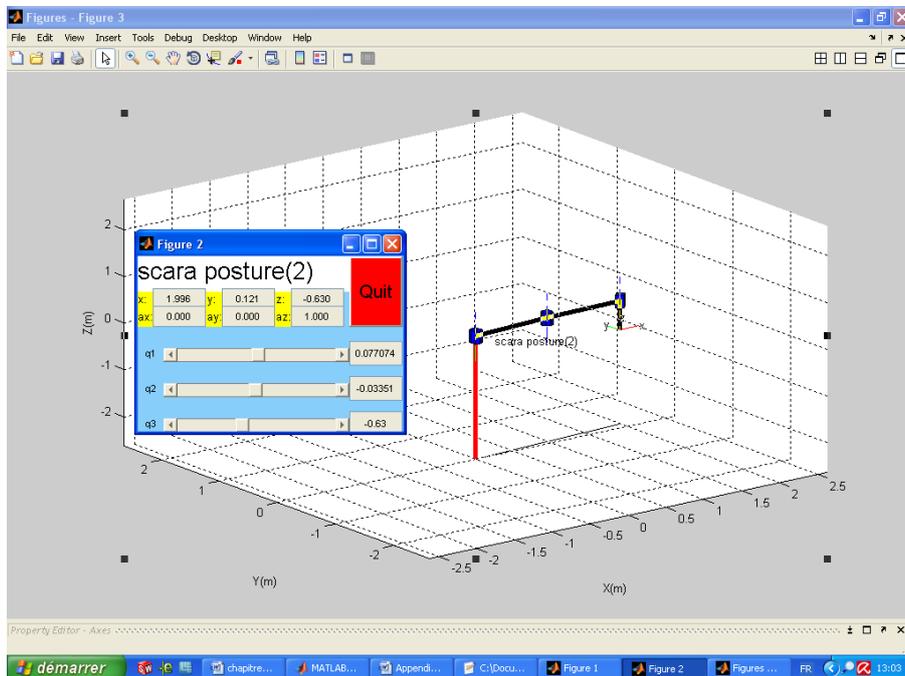


Figure 4.21 : Figure 4-17 : La position de *posture2* à l' instant t=0.4s.

t=0.6s
Posture1

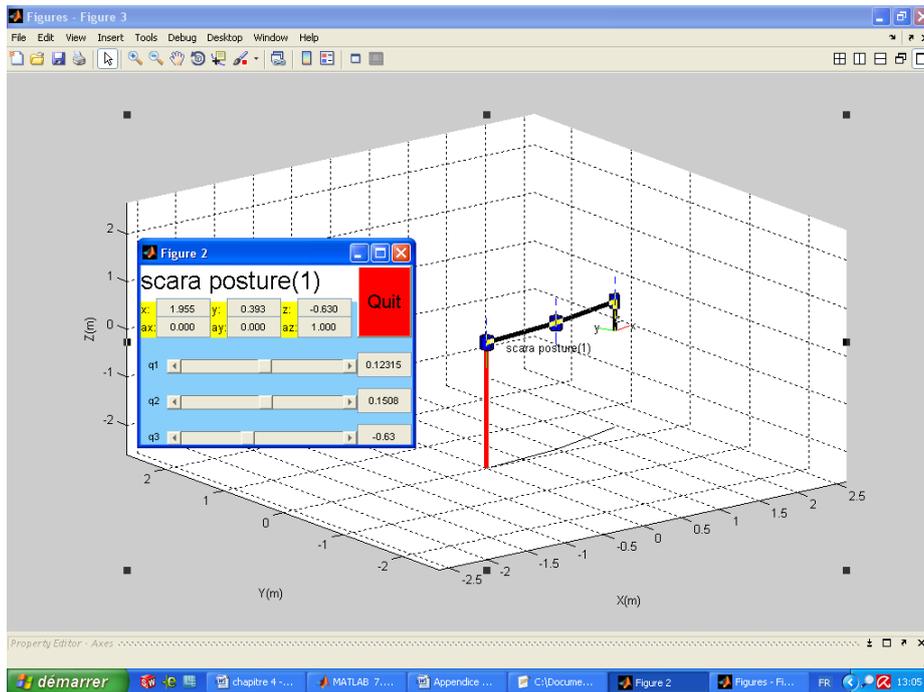


Figure 4.22 : Figure 4-18: La position de la *posture1* à l' instant t=0.6s.

Posture2

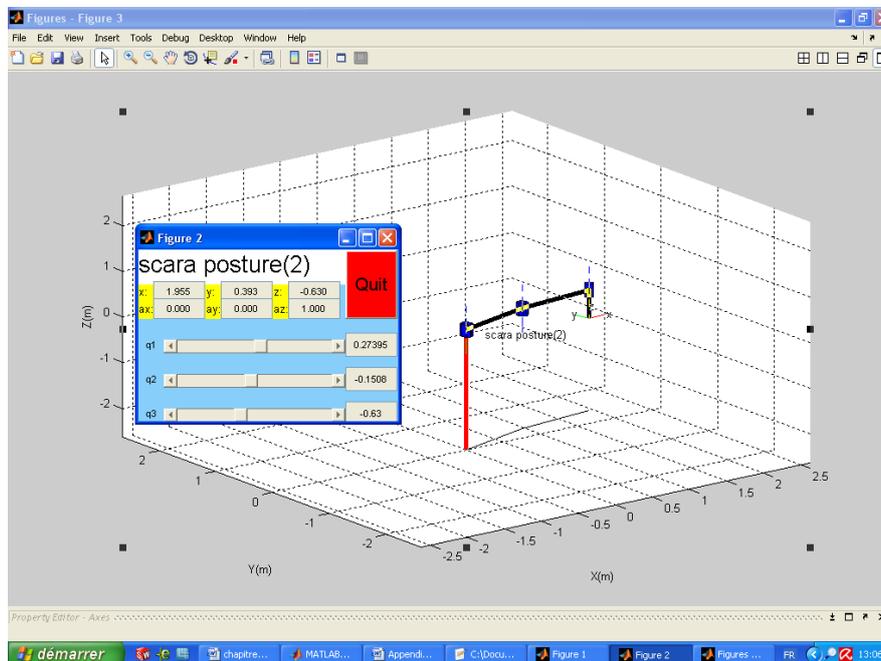


Figure 4.23 : Figure 4-19: La position de la *posture2* à l' instant t=0.6s.

t=1s
posture1

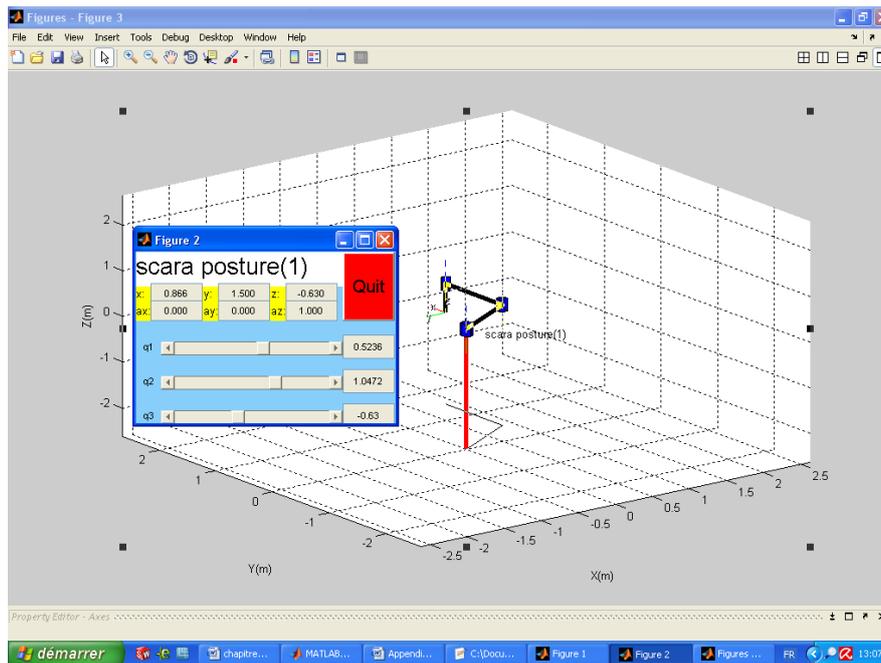


Figure 4.24 : La position de la *posture1* à l'instant $t=1s$.

Posture2

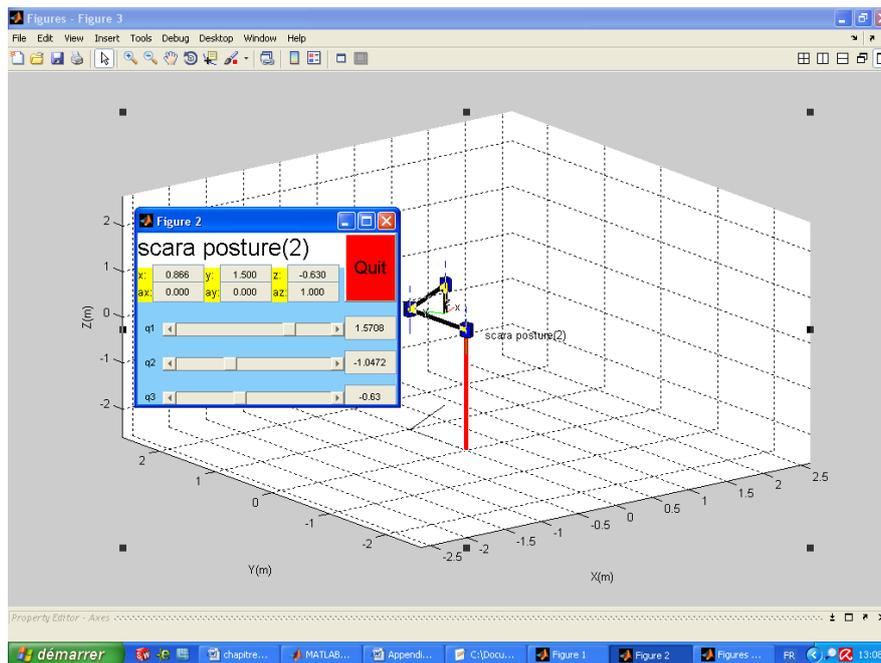


Figure 4.25 : La position de la *posture2* à l'instant $t=1s$.

4-5-7 Orientation de la matrice de transformation homogène pour les deux postures à t=1s :
Posture1 :

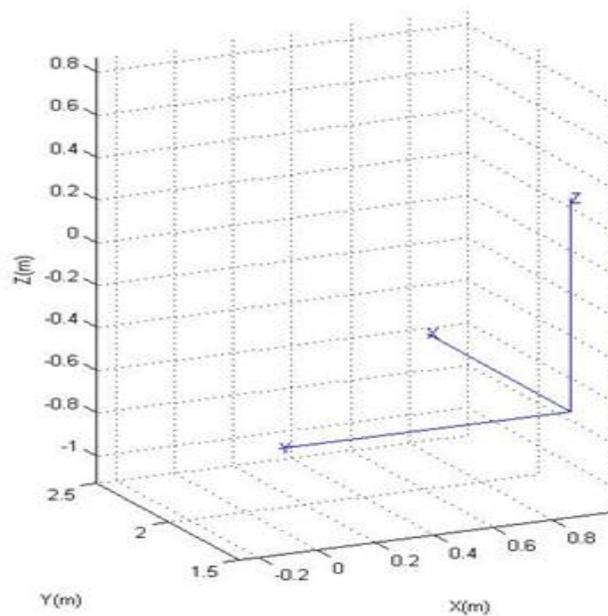


Figure 4.26 : L'orientation de la matrice de transformation homogène *posture1* à t=1s.

Posture2 :

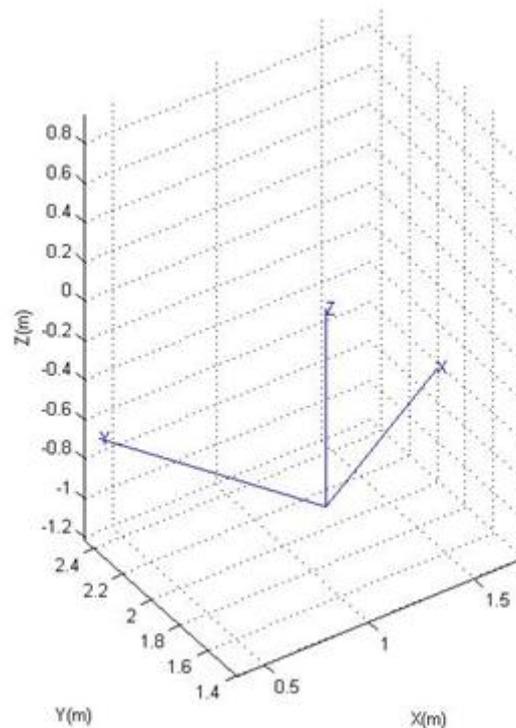


Figure 4.27 : L'orientation de la matrice de transformation homogène *posture2* à t=1s.

Remarque : les variables articulaires exprimées dans Matlab/simulink sont en radians

Les résultats obtenus soit à l'aide du logiciel SolidWorks et ou bien calculés par le logiciel MATLAB/simulink sont exactement les mêmes. Cette similitude des résultats conforte le degré de fiabilité du modèle cinématique.

On peut tirer profit de cette analyse pour étudier le comportement du robot dans les deux postures afin d'établir un bilan énergétique comparatif de ces deux postures.

A l'aide logiciel Solidworks on peut tirer :

posture1 : Le lien1 :

Déplacement suivant x :

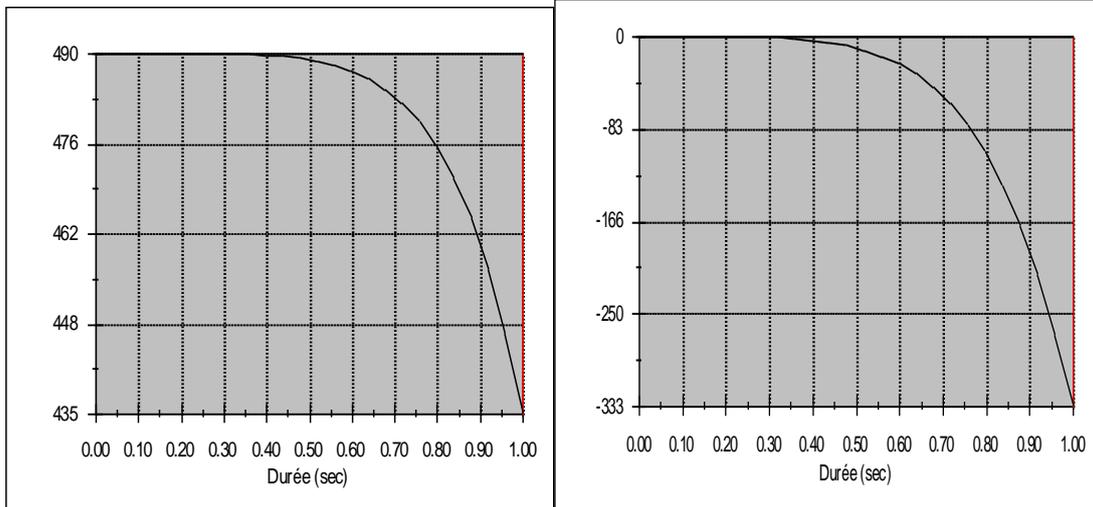


Figure 4.28 : Déplacement et vitesse du lien1 suivant x (*posture1*).

Déplacement suivant y :

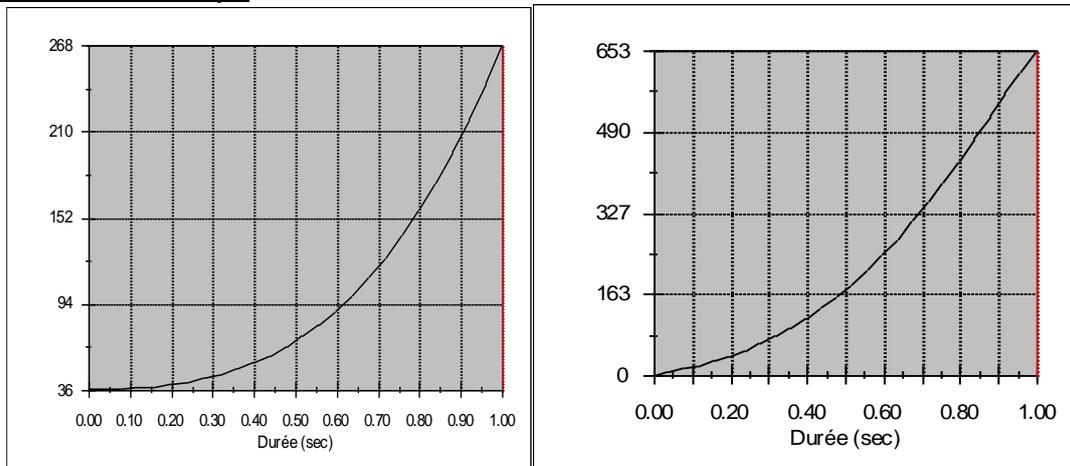


Figure 4.29 : Déplacement et vitesse du lien1 suivant y (*posture1*).

Le lien2 :

Déplacement suivant x :

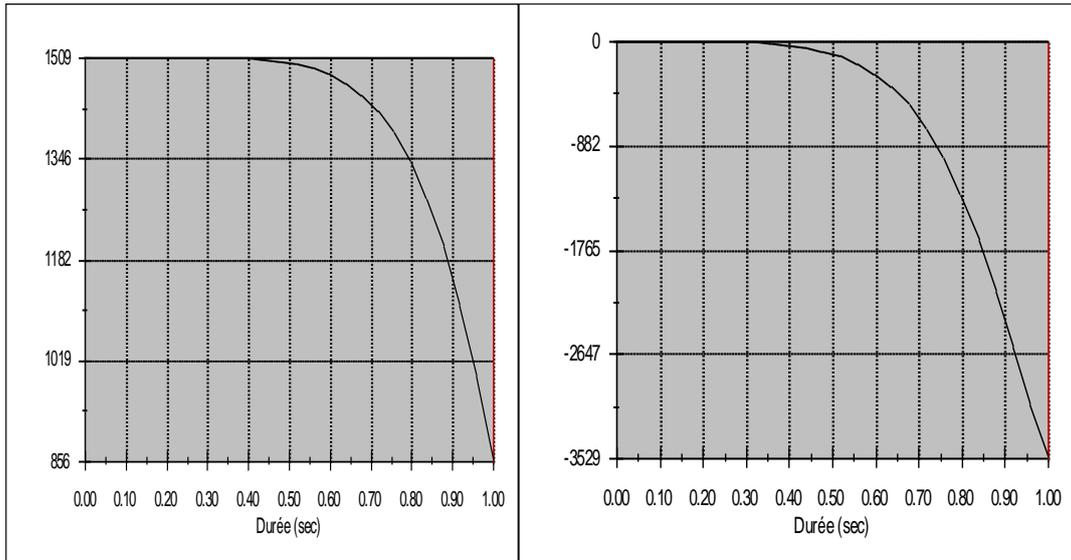


Figure 4.30 : déplacement et vitesse du lien2 suivant x (*posture1*).

Déplacement suivant y :

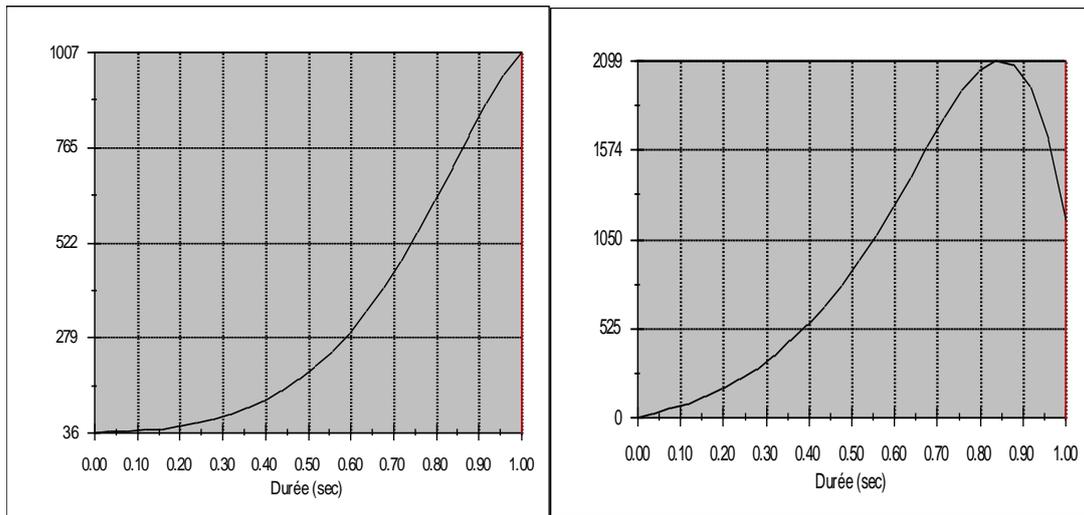


Figure 4.31 : Déplacement et vitesse du lien2 suivant y (*posture1*).

Posture2 : Le lien1 :

Déplacement suivant x :

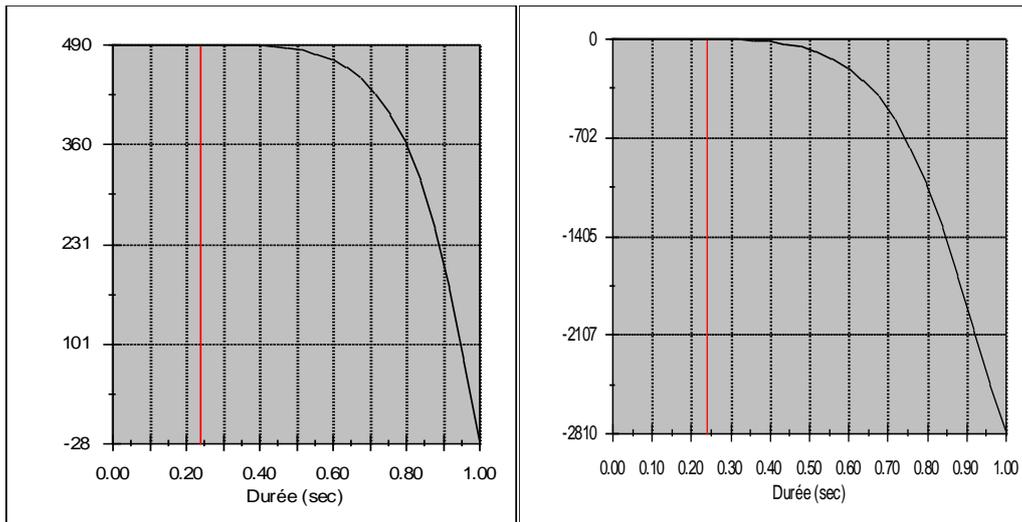


Figure 4.32 : Déplacement et vitesse du lien1 suivant x (*posture2*).

Déplacement suivant y :

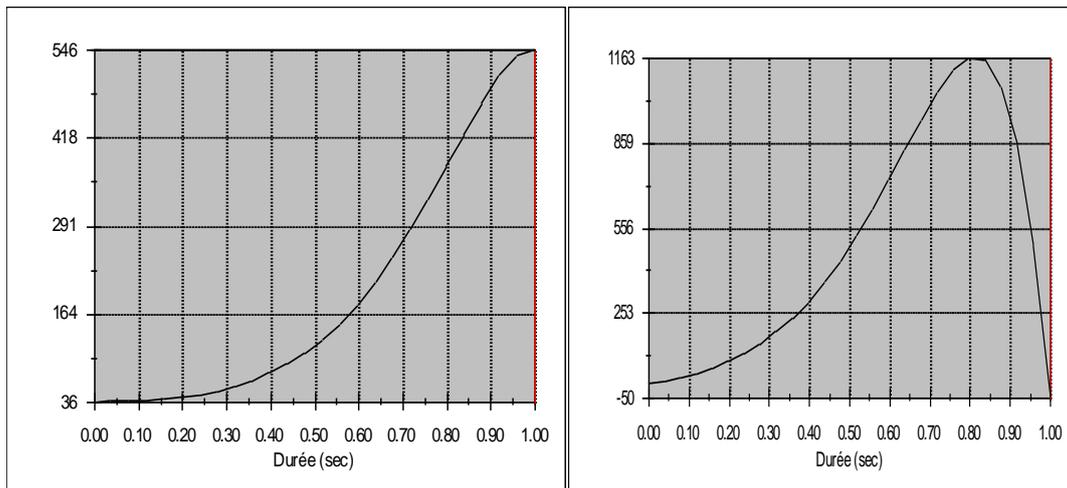


Figure 4.33 : Déplacement et vitesse du lien1 suivant y (*posture2*).

Le lien 2 :

Déplacement suivant x :

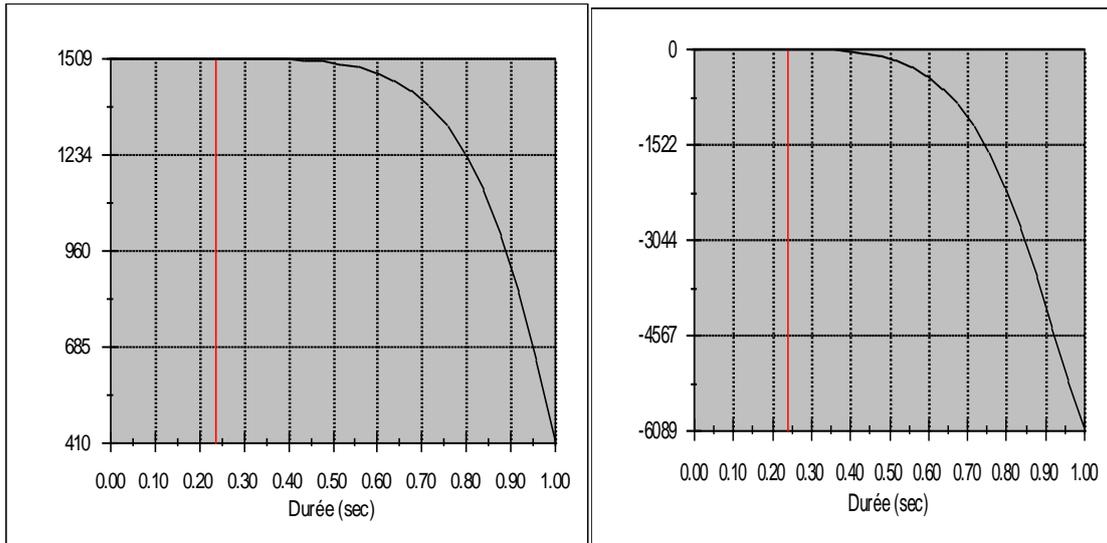


Figure 4.34 : Déplacement et vitesse du lien2 suivant x (*posture2*).

Déplacement suivant y :

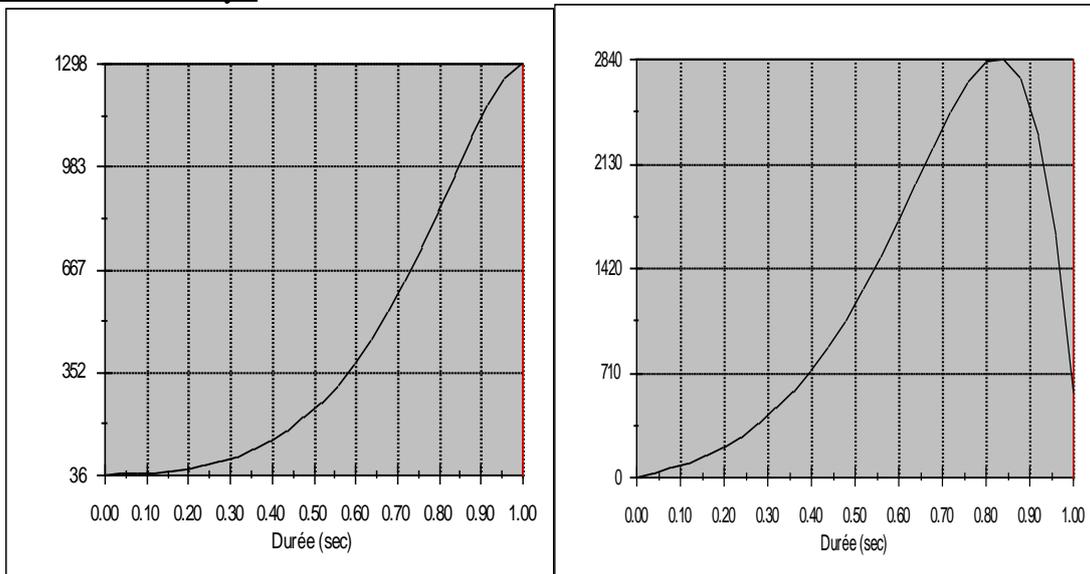


Figure 4.35 : Déplacement et vitesse du lien2 suivant y (*posture2*).

Conclusion :

On remarque d'après les résultats obtenus que le déplacement et la vitesse des liens 1et2de la posture2 sont supérieur au déplacement et la vitesse des liens1 et 2 de la posture1 .cette différence peut induire une différence des énergies consommées lors des deux postures, et ce pour une même trajectoire et une même position désirée.

4-5-8 La position du robot durant la simulation :

4-5-8-1 SolidWorks :

Posture1 :

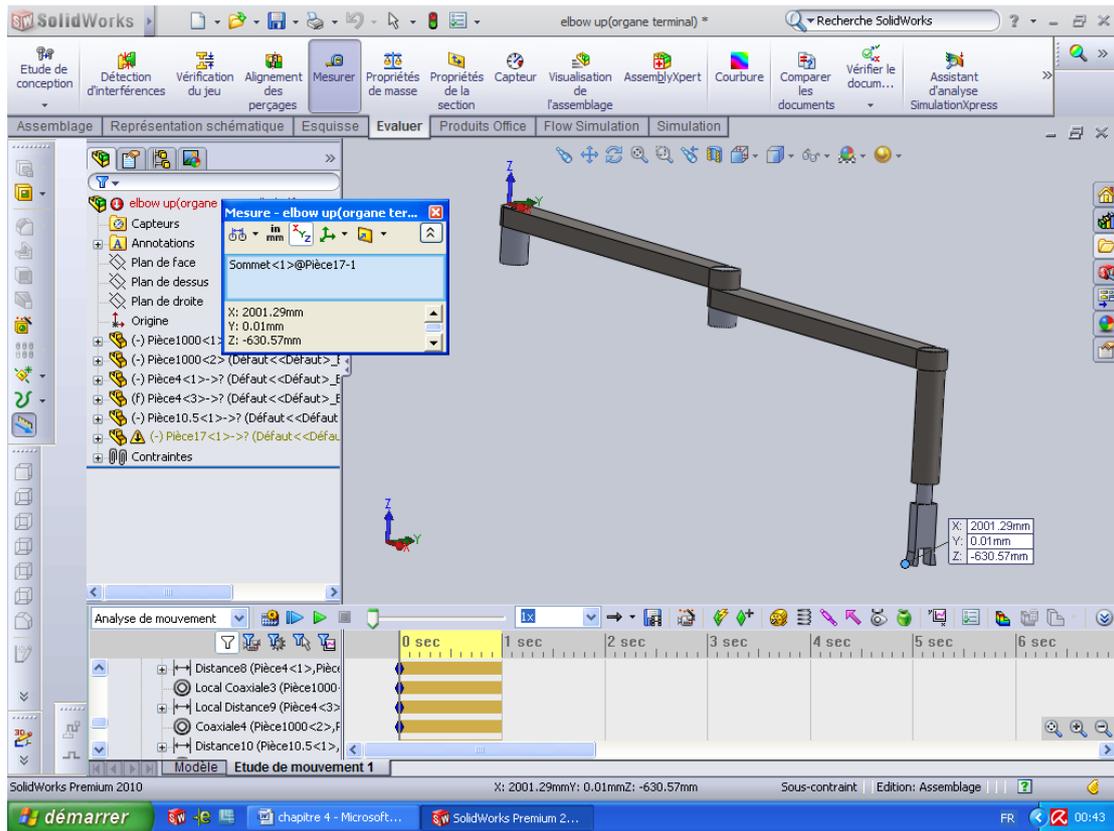


Figure 4.36 : La position de *posture1* à l'instant $t=0s$ (home position).(SW)

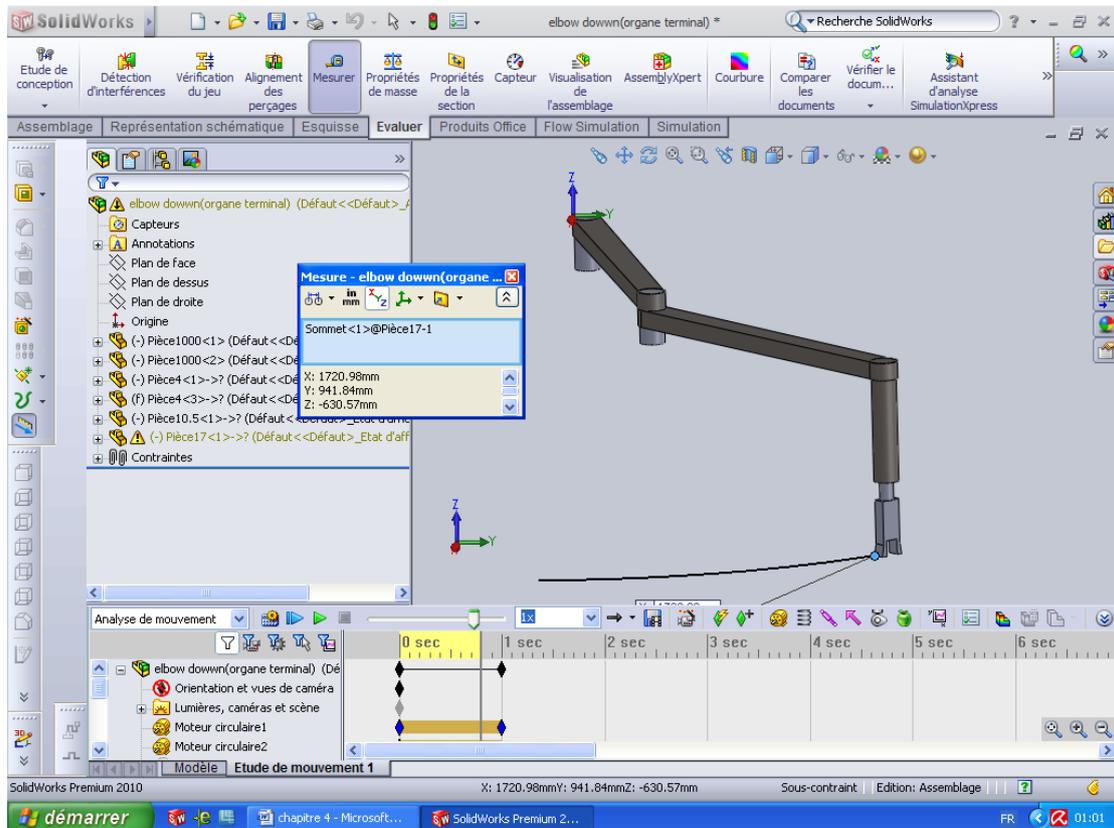


Figure 4.37 : La position de *posture1* à l' instant $t=0.8s$.(SW)

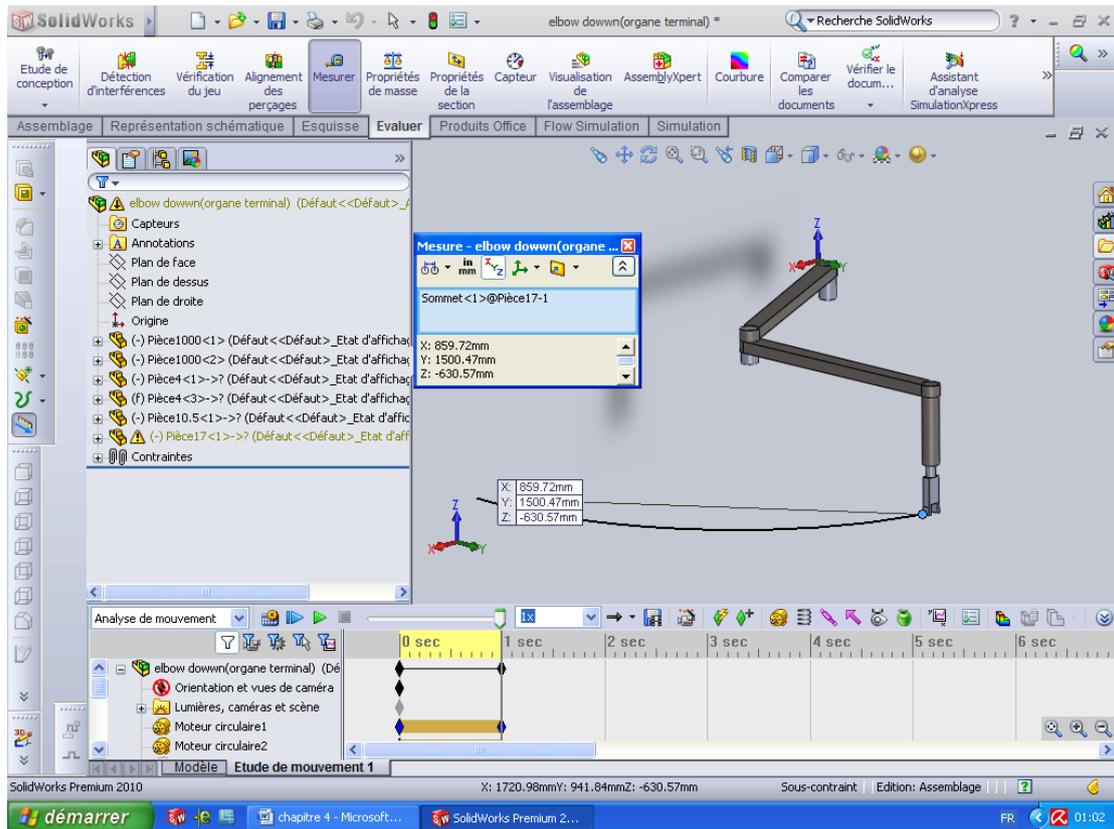


Figure 4.38 : La position de *posture1* à l'instant $t=1s$ (la position désirée).(SW)

Posture2 :

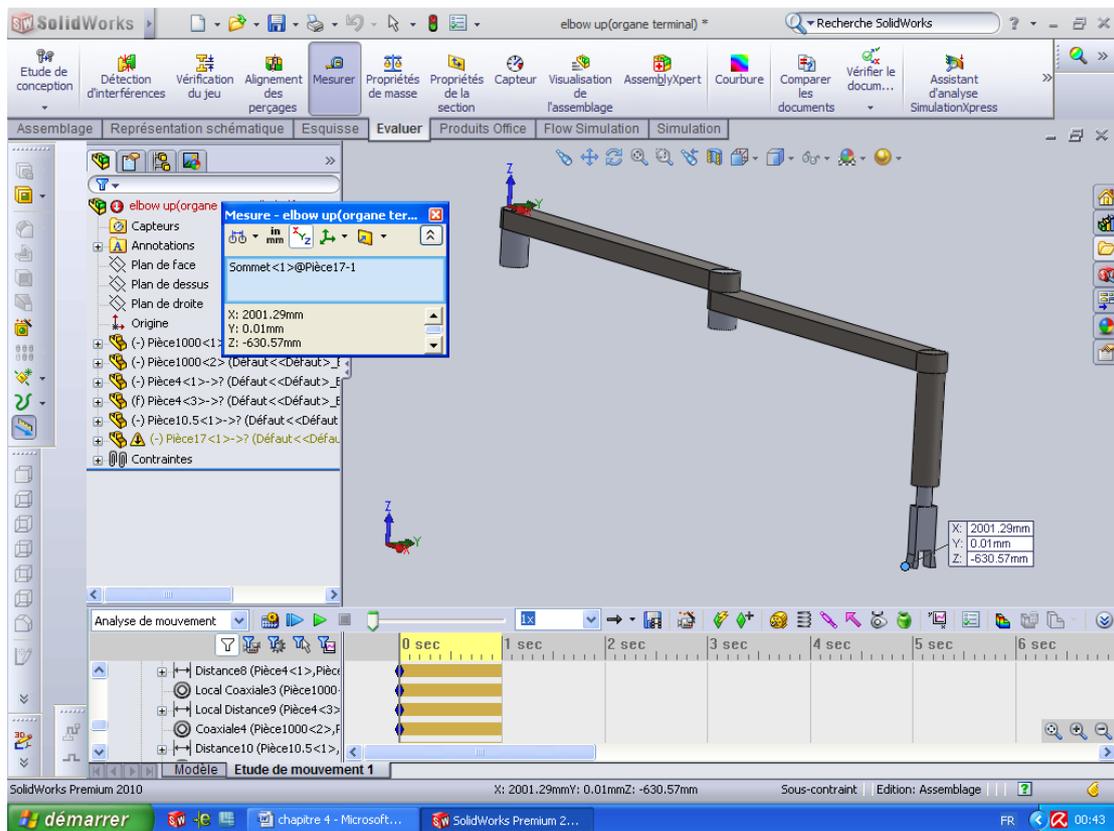


Figure 4.39 : La position de *posture2* a l'instant $t=0s$ (home position) (SW)

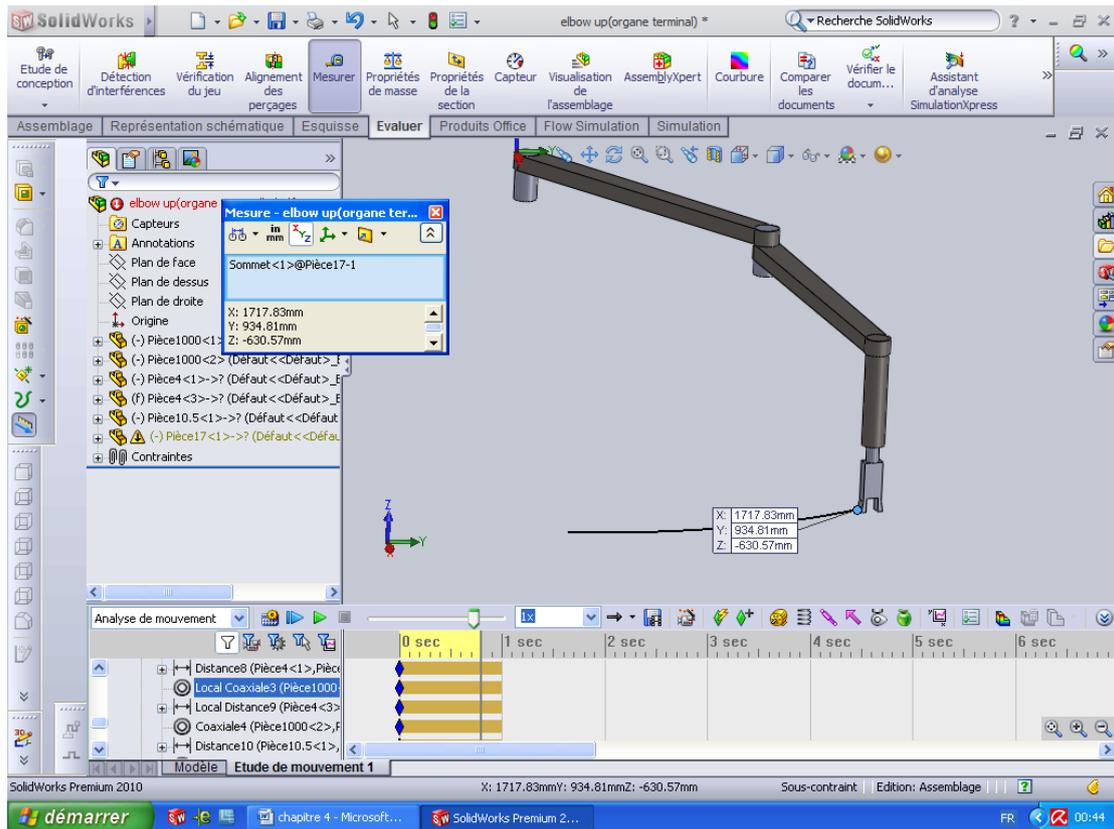


Figure 4.40 : La position de *posture2* à l' instant $t=0.8s$.(SW)

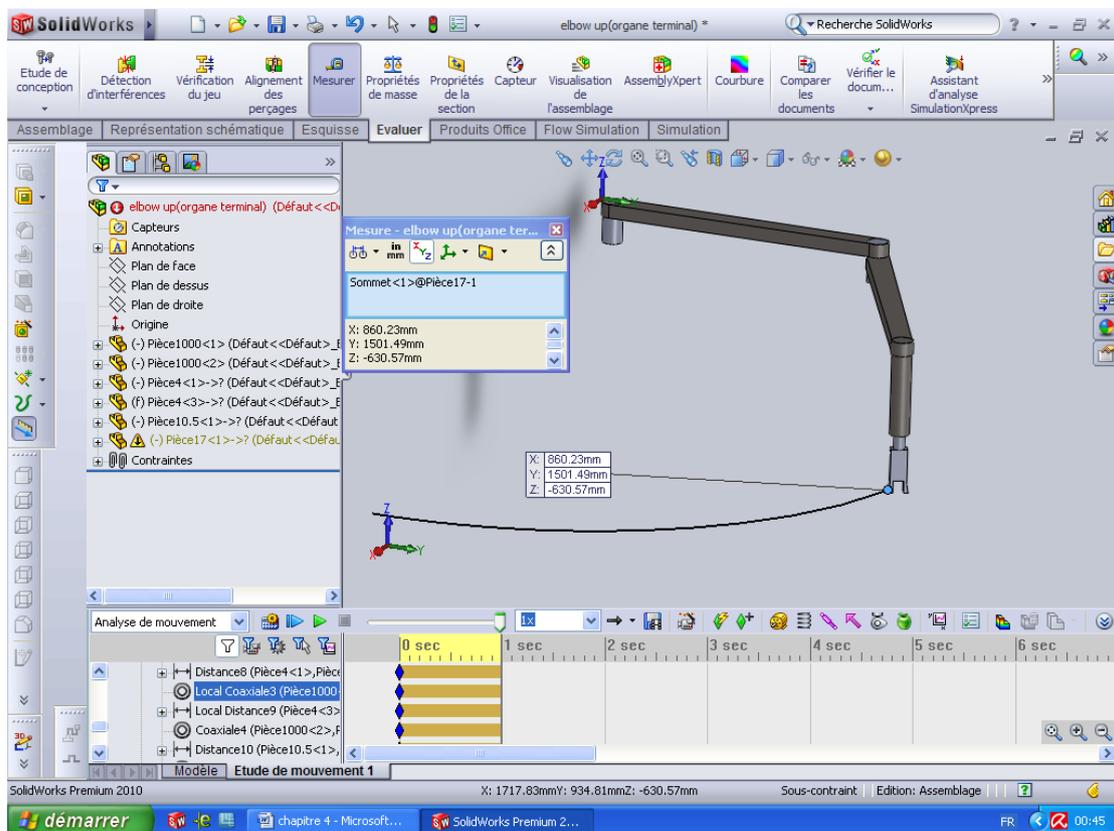


Figure 4.41 : La position de *posture2* à l' instant $t=1s$ (la position désirée).(SW)

4-5-8-2 Matlab/simulink :

Posture 1 :

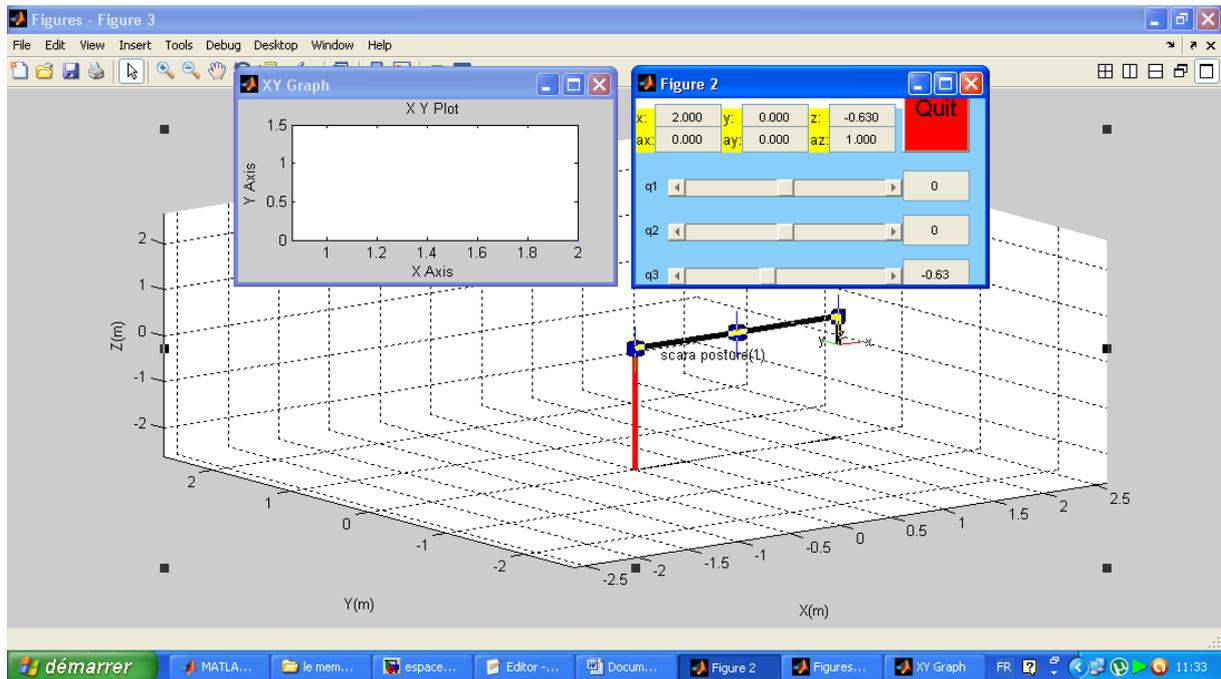


Figure 4.42 : La position de *posture 1* a l'instant $t=0s$ (home position).(M/S)

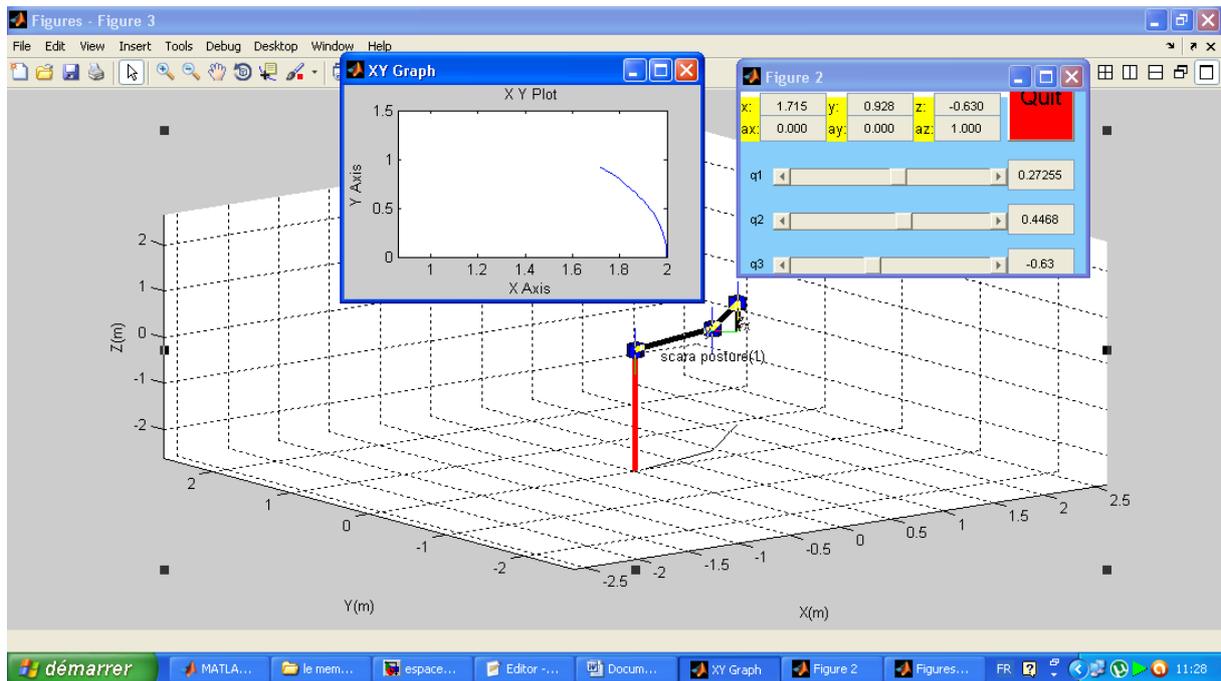


Figure 4.43 : La position de *posture 1* à l'instant $t=0.8s$.(M/S)

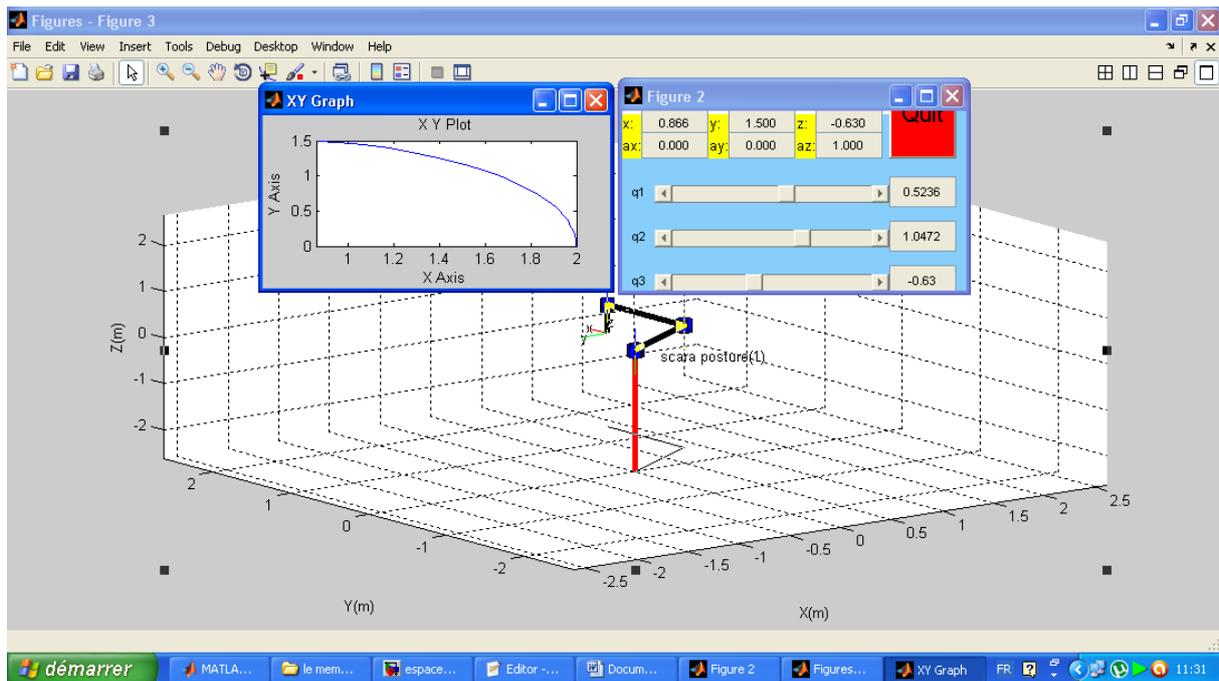


Figure 4.44 : La position de *posture1* à l'instant $t=1s$ (la position désirée).(M/S)

Posture 2 :

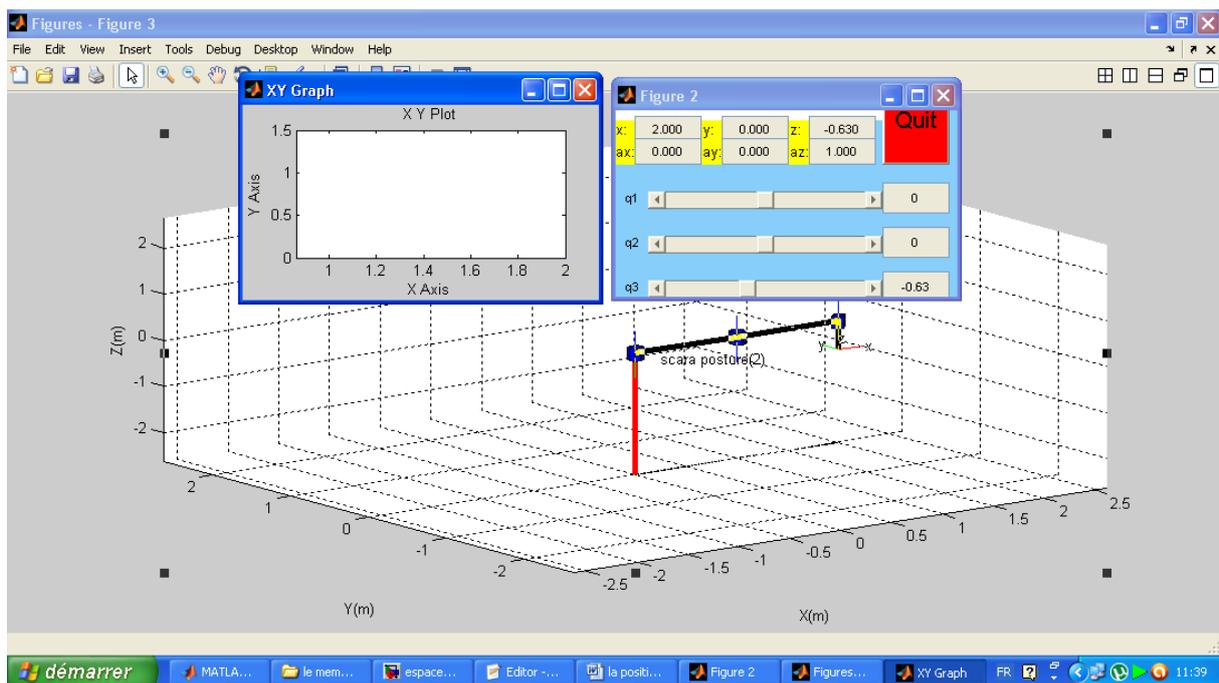


Figure 4.45 : La position de *posture2* a l'instant $t=0s$ (home position) (M/S)

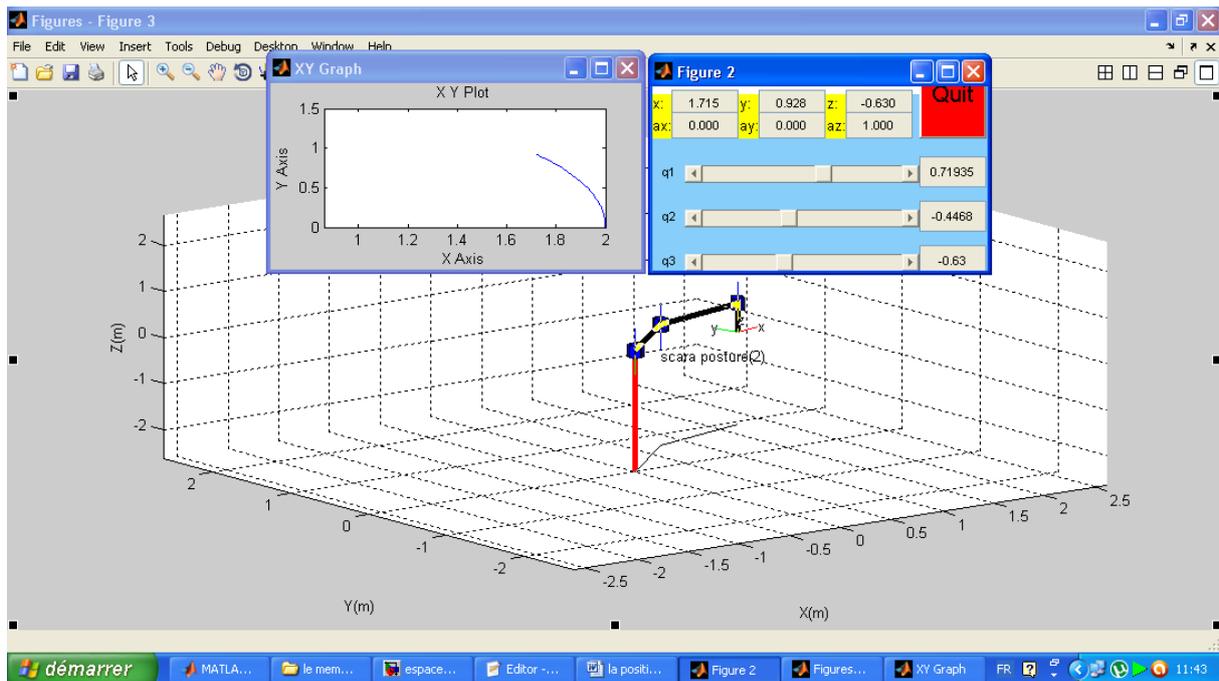


Figure 4.46 : La position de *posture2* à l'instant $t=0.8s$. (M/S)

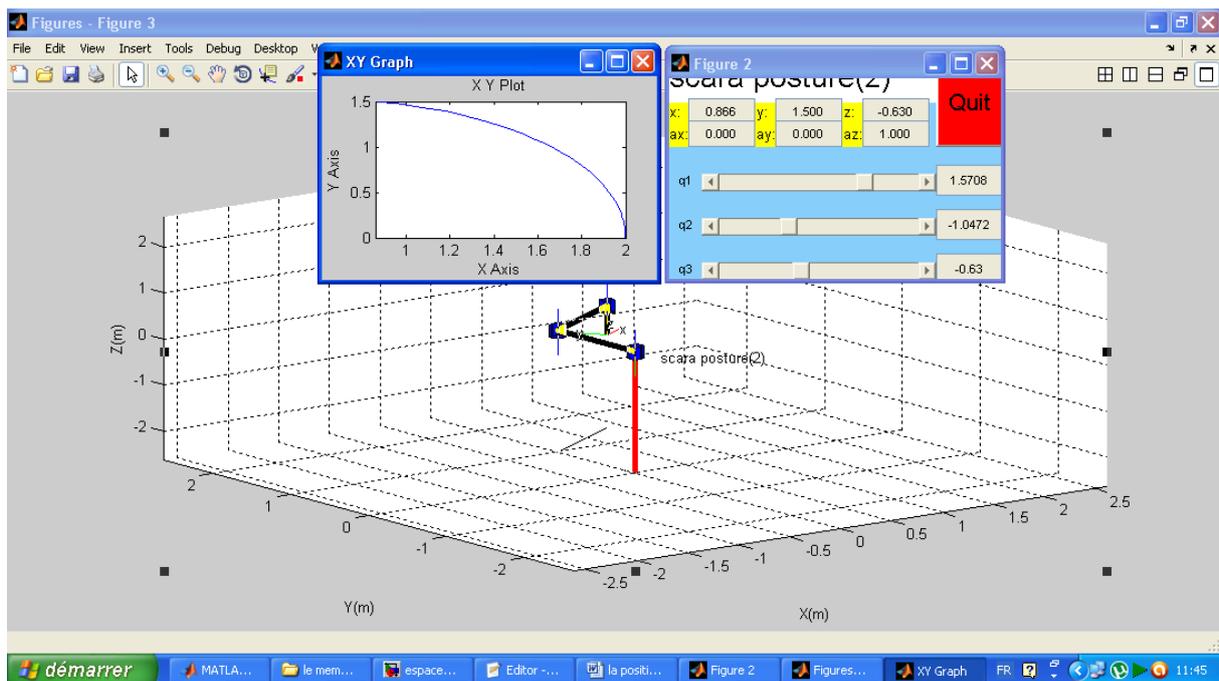


Figure 4.47 : La position de *posture2* à l'instant $t=1s$ (la position désirée). (M/S)

4-5-9 les schémas des deux postures du robot SCARA pour la même position désirée :

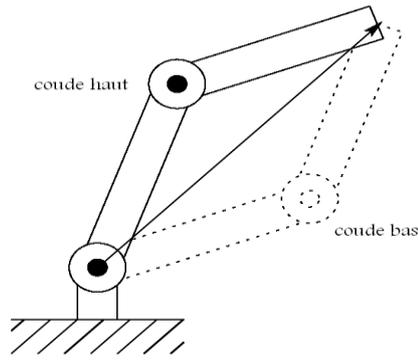


Figure 4.48: Les deux *postures* pour un robot SCARA.

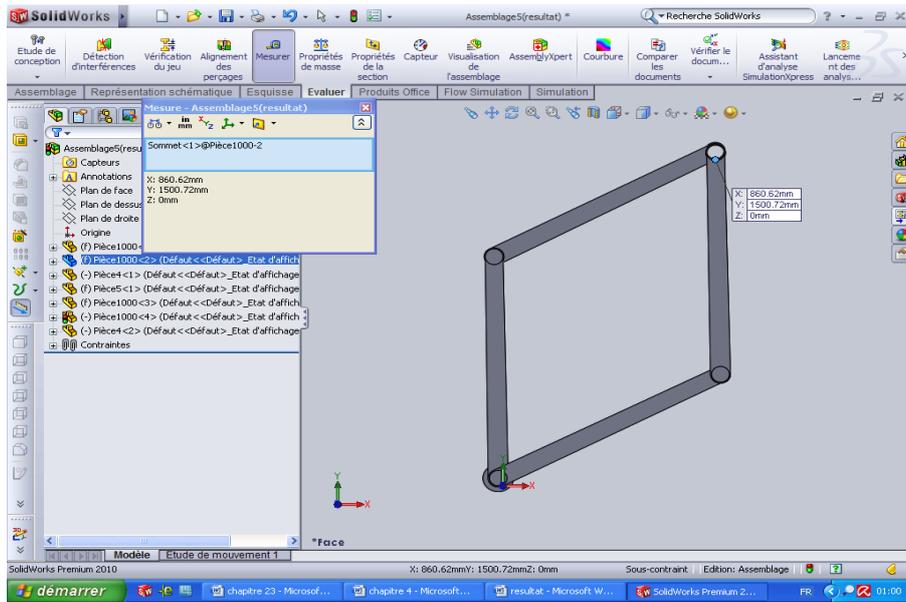


Figure 4.49 : Les deux *postures* à l'aide logiciel Solidworks pour notre cas.

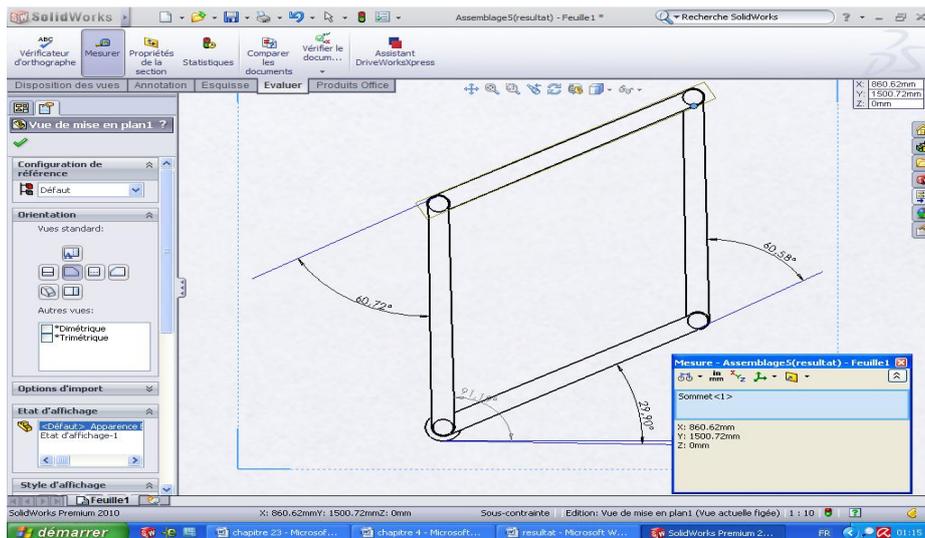


Figure 4.50 : Les deux *postures* à l'aide le logiciel Solidworks.

4-6 Analyse de l'organe terminal :

Dans notre cas l'organe terminal fait l'opération (pick and place), soit une translation suivant l'axe (z).

4-6-1 Précision dans la prise d'objet:

L'organe terminal agrippe l'objet et l'oriente avec précision selon une trajectoire préalablement établie.

Le schéma suivant montre le système le plus utilisé dans l'opération (pick and place) :

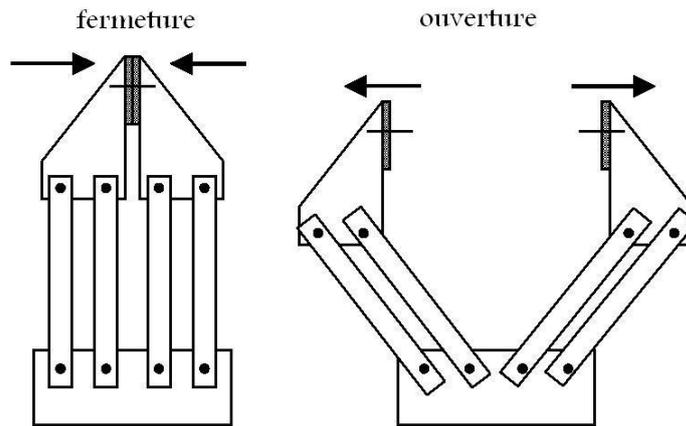


Figure 4.51 : Organe terminal avec 4 tringleries de barre. [46].

L'ouverture et la fermeture de ce système se font soit par commande électrique, hydraulique ou pneumatique.

4-6-2 Orientation de l'organe terminal :

L'orientation de l'organe terminal doit satisfaire la matrice de la position désirée, et l'axe de l'organe terminal doit être vertical, dans ce cas n'importe quelle valeur de θ_4 est acceptée. [34]

La position désirée dépend essentiellement de l'orientation de l'objet à manipuler sur la scène de manipulation.

Si on considère que l'orientation de l'organe terminal et l'objet à manipuler ont des axes parallèles; cela veut dire que l'organe terminal n'a pas besoin d'une rotation quand il arrive à la position désirée pour agripper l'objet $\theta_4 = 0^\circ$ on a sauf le déplacement linéaire. Comme montre le schéma suivant :

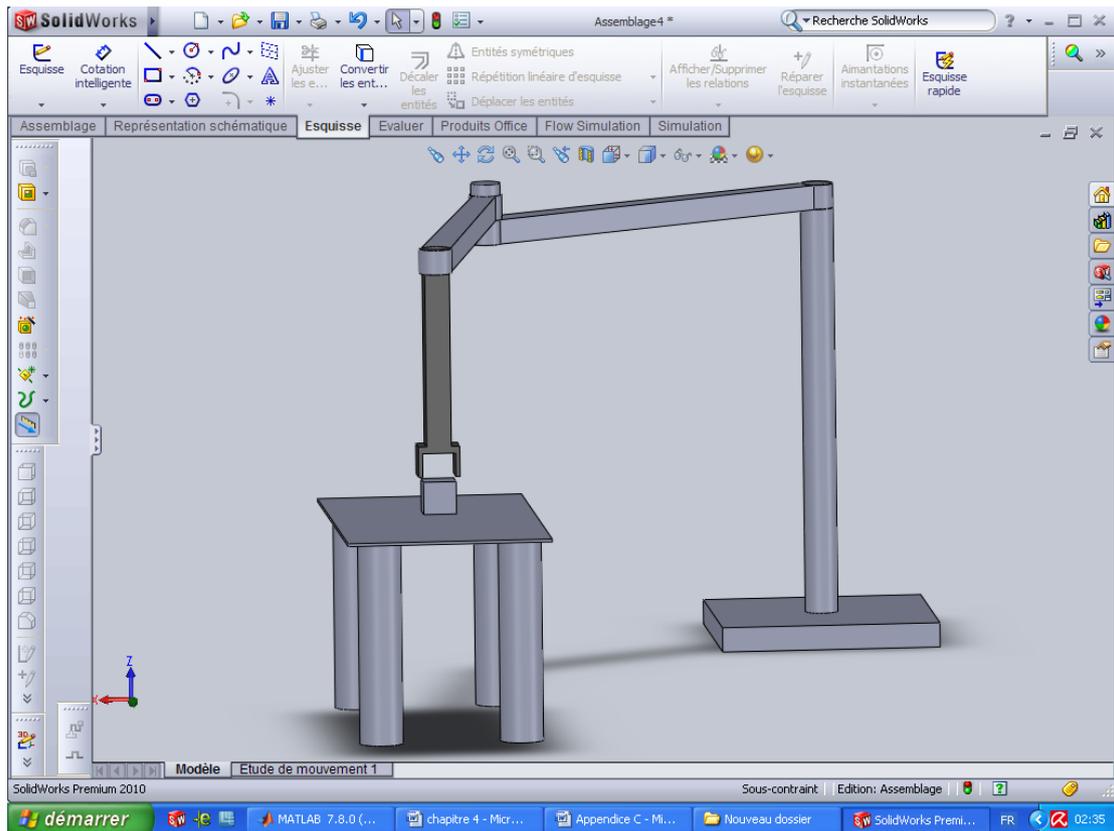


Figure 4.52 : Schéma montrant l'orientation de l'organe terminal et l'objet à manipuler sur la scène de manipulation.

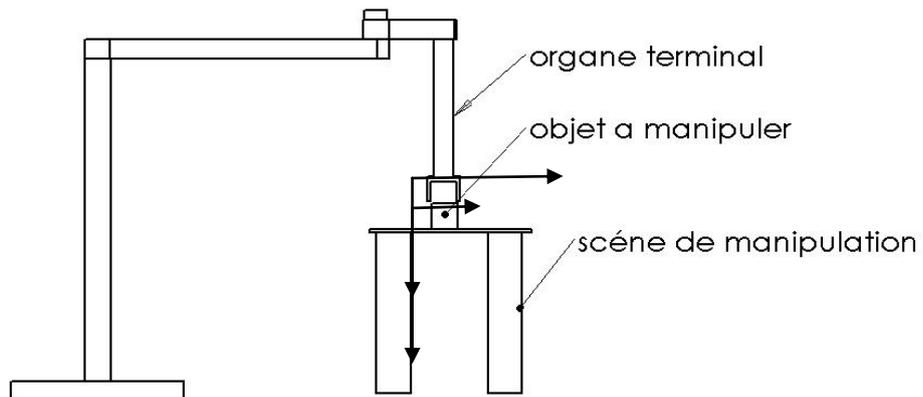


Figure 4.53 : l'orientation de l'organe terminal et l'objet à manipuler sur la scène de manipulation.

Dans ce cas La matrice de la position désirée est sous la forme :

Pour la posture 1 :

$$T_H^R = \begin{pmatrix} 0 & -1 & 0 & 0.86 \\ 1 & 0 & 0 & 1.5 \\ 0 & 0 & 1 & 0.63 \\ 0 & 0 & 0 & 1 \end{pmatrix} \dots\dots\dots (4-85)$$

Vérification :

On a une rotation suivant l'axe (z).

$$(4.85) = (4.12)$$

$$c_{124} = 0$$

$$\text{A } t=1s \theta_1 + \theta_2 = 90^\circ$$

$$\text{Donc : } \theta_4 = 0^\circ$$

Ou bien on applique la relation (4.70) on trouve :

Identification (4-85) avec (4-30) on peut tirer :

$$n_x = 0 \text{ et } n_y = 1$$

$$\text{donc } \theta_4 = 0^\circ .$$

Pour la posture2 :

$$T_H^R = \begin{pmatrix} 0.86 & -0.5 & 0 & 0.86 \\ 0.5 & 0.86 & 0 & 1.5 \\ 0 & 0 & 1 & 0.63 \\ 0 & 0 & 0 & 1 \end{pmatrix} \dots\dots\dots (4-86)$$

Vérification :

On a une rotation suivant l'axe (z).

$$(4.86) = (4.20)$$

$$c_{124} = 0.86$$

$$\text{A } t=1s \theta_1 + \theta_2 = 30^\circ$$

$$\text{Donc : } \theta_4 = 0^\circ$$

Ou bien on applique la relation (4.70) on trouve :

Identification (4-86) avec (4-30) on peut tirer

$$n_x = 0.86 \text{ et } n_y = 0.5$$

$$\text{donc } \theta_4 = 0^\circ .$$

4-6-3 Déplacement en translation de l'organe terminal :

On veut que l'organe terminal quand il arrive à la position désirée se déplace de 5mm à 1s

Soit l'équation de mouvement sous la forme [45]:

$$p_z = 0.63 + 0.005t^3 \dots\dots\dots (4.87)$$

Et on a : $0 \leq t \leq 1$

$$d_3 + d_4 = -p_z \dots\dots\dots (4.88)$$

Résultat de simulation :

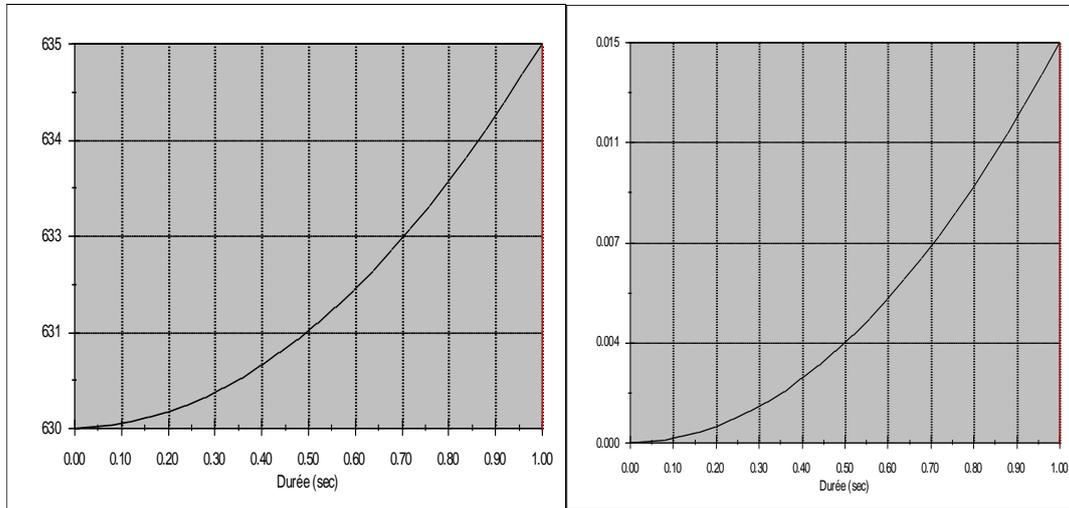


Figure 4.54: Déplacement et vitesse en translation de l'organe terminal (Solidworks).

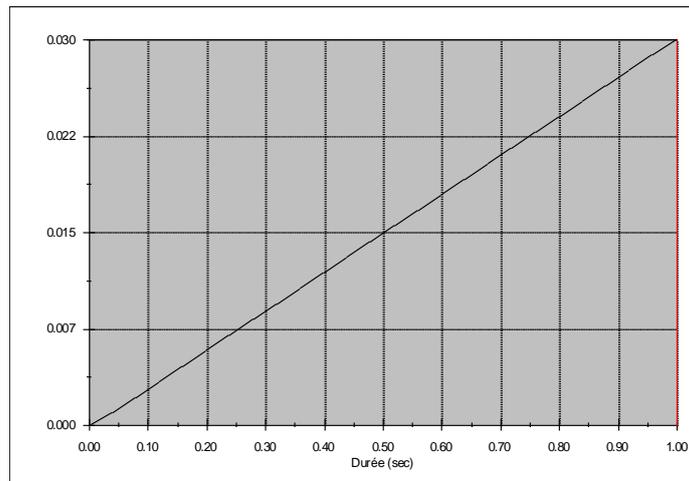


Figure 4.55 : Accélération en translation de l'organe terminal (Solidworks).

4-6-3-1 Vérification des résultats obtenus a l'aide le logiciel Matlab :

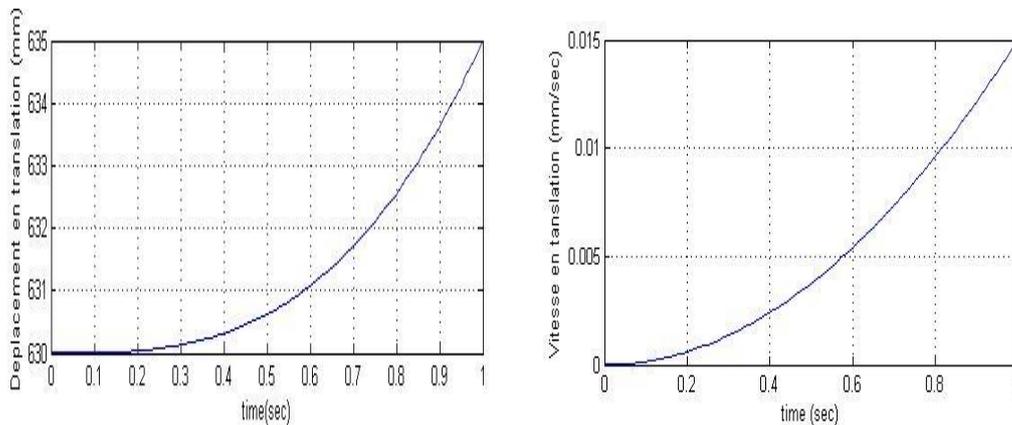


Figure 4.56 : Déplacement et vitesse en translation de l'organe terminal (MATLAB).

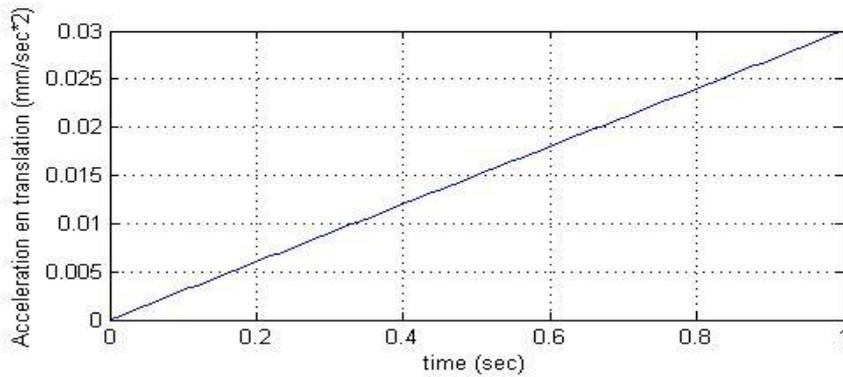


Figure 4.57 : Accélération en translation de l'organe terminal (MATLAB).

4-7 Calcul des vitesses dans l'espace opérationnel :

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

Le modèle est décrit par l'équation :

$$\dot{X} = J(q)\dot{q} \dots\dots\dots (4.89)$$

J est la matrice jacobienne ou le manipulateur jacobien.

Sachant que J s'appelle la matrice jacobienne ou le manipulateur jacobien.

4-7-1 La matrice jacobienne :

L'outil mathématique principalement utilisé pour traiter le problème de la cinématique, et particulièrement des modèles géométriques et cinématique des robots est la « la matrice jacobienne ». Elle représente un opérateur permettant de lier les vitesses des corps d'un robot exprimées dans différents espaces vectoriels.

En considérant une fonction f définie comme suite : [33]

$$f : Q \rightarrow w \dots\dots\dots (4.90)$$

$$q \rightarrow x = f(Q) \dots\dots\dots (4.91)$$

Ou Q et w sont deux espaces vectoriels de dimensions (respectivement) m et n

La différentiation de l'équation $x = f(Q)$ donne :

$$dx = \frac{\partial f}{\partial q} dq \dots\dots\dots (4.92)$$

On définit alors, la matrice jacobienne comme suit :

$$[J(q)] = \left[\frac{\partial f}{\partial q} \right] \dots\dots\dots (4.93)$$

4-7-2 La matrice jacobienne pour un point quelconque :

On considère un manipulateur de trois liens :

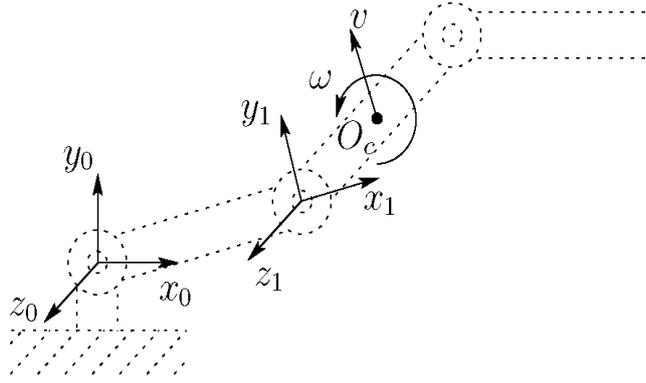


Figure 4.58 : manipulateur plan a 3 bras.

La matrice jacobienne est sous la forme :

$$J(q) = \begin{bmatrix} z_0(o_c - o_0) & z_1(o_c - o_1) & 0 \\ z_0 & z_1 & 0 \end{bmatrix} \dots \dots \dots (4.94)$$

Sachant que :

$$z_0 = z_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \dots \dots \dots (4.95)$$

La matrice jacobienne est sous la forme 6xn ; et n le nombre de liens.

4-7-3 La matrice jacobienne du robot SCARA :

Pour calculer le jacobien du robot SCARA selon [33] [35]; Les articulations 1,2 et 4 sont de type rotoïde et l'articulation 3 est prismatique, puisque $o_4 - o_3$ est parallèle à z_3 .

$$z_3(o_4 - o_3) = 0$$

Le jacobien est sous la forme :

$$J = \begin{bmatrix} z_0(o_4 - o_0) & z_1(o_4 - o_1) & z_2 & 0 \\ z_0 & z_1 & 0 & z_3 \end{bmatrix} \dots \dots \dots (4.96)$$

On a :

$$J = \begin{bmatrix} z_{i-1} \\ 0 \end{bmatrix} \dots \dots \dots (4.97)$$

z_{i-1} : Pour les joints rotoïdes

0 : pour les joints prismatique.

Sachant que :

$$o_1 = \begin{bmatrix} l_1 c_1 \\ l_1 s_1 \\ 0 \end{bmatrix} \dots \dots \dots (4.98)$$

$$o_2 = \begin{bmatrix} l_1 c_1 + l_2 c_{12} \\ l_1 s_1 + l_2 s_{12} \\ 0 \end{bmatrix} \dots\dots\dots (4.99)$$

$$o_4 = \begin{bmatrix} l_1 c_1 + l_2 c_{12} \\ l_1 s_1 + l_2 s_{12} \\ -d_3 - d_4 \end{bmatrix} \dots\dots\dots (4.100)$$

Et on a :

$$z_0 = z_1 = z_2 = z_3 \dots\dots\dots (4.101)$$

Finalemnt on obtient la matrice jacobienne suivante :

$$J = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} & 0 & 0 \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \dots\dots\dots (4.102)$$

4-7-4 Les vitesses dans l'espace opérationnel :

$$\dot{X} = J(q)\dot{q}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} & 0 & 0 \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{bmatrix} \dots\dots\dots (4.103)$$

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \\ 0 \\ 0 \\ V \end{bmatrix} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} & 0 & 0 \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_4 \\ 0 \\ 0 \\ \dot{d}_3 \end{bmatrix} \dots\dots\dots (4.104)$$

Résultat de simulation :

Posture1 :

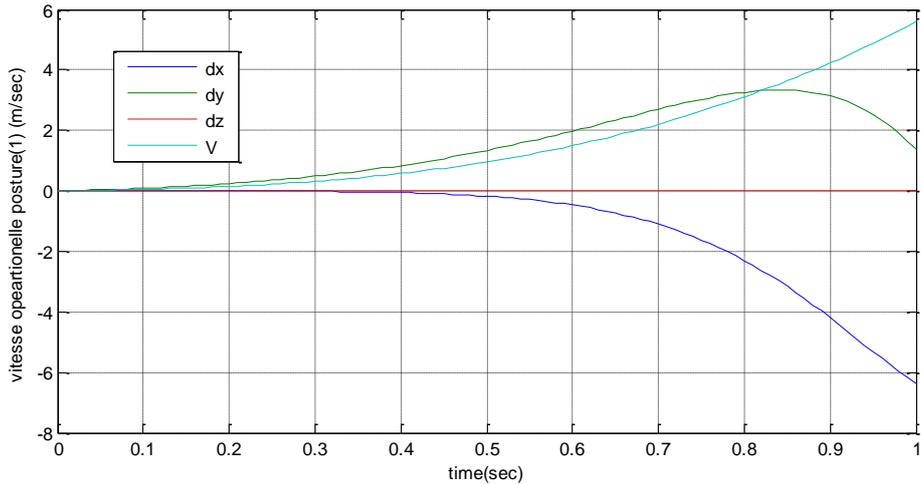


Figure 4.59 : espace opérationnel de *posture1*.

Posture2 :

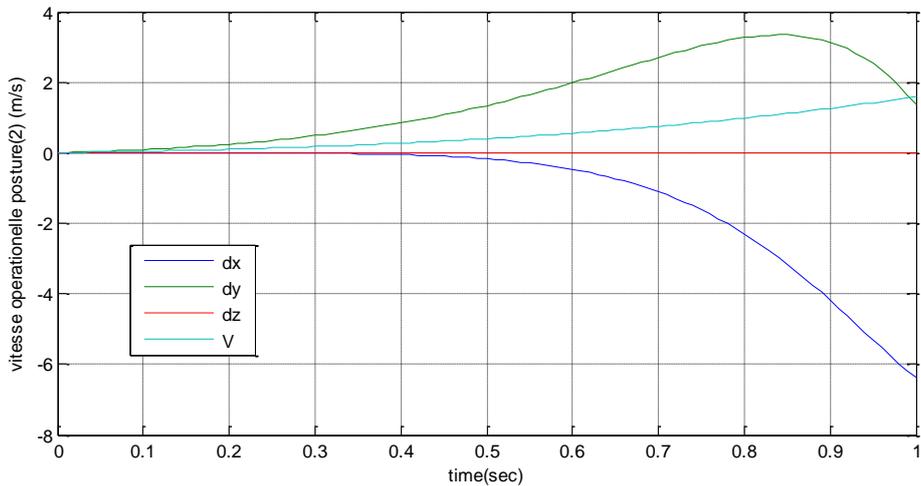


Figure 4.60: espace opérationnel de *posture2*.

4-8 La singularité du robot SCARA :

D'une façon générale, pour n'importe quel robot, redondant ou pas, il est possible de trouver quelques configurations, appelées « configurations singulières », dans lesquelles le nombre du degré de liberté de l'organe terminal est inférieur à la dimension dans laquelle il fonctionne généralement. Les configurations singulières se produisent quand :

- Deux axes des joints prismatiques deviennent parallèles.
- Deux axes des joints rotoides deviennent identiques.

Aux positions singulières, l'organe terminal perd un ou plusieurs degrés de liberté, puisque les équations cinématiques deviennent linéairement dépendantes, les solutions deviennent non définies. Les positions singulières doivent être évitées, car les vitesses exigées pour déplacer l'organe terminal deviennent théoriquement infinies.[43]

Les configurations singulières peuvent être déterminées de la matrice jacobienne J, elle rapporte les déplacements infinitésimaux de l'organe terminal. [35][36][37].

L'identification et l'action d'éviter les configurations de singularité sont très importantes en robotique pour les raisons principales suivantes :

1. Certaines directions de mouvement peuvent être inaccessibles.
2. Certaines vitesses communes sont infinies.
3. Certains couples communs deviennent infinis.
4. La non existence de l'unicité d'une solution unique au problème inverse de la cinématique.

La détection des configurations singulières qui utilise le déterminant du Jacobien peut être une tâche pénible pour les robots complexes. Cependant, pour des robots ayant un poignet sphérique, il est possible de classer le problème de détection de singularité en deux problèmes séparés :

1. singularités des coudes résultant du mouvement des bras de manipulateur.
2. Singularités de poignet résultant du mouvement du poignet.

Nous avons déjà calculé le Jacobien complet pour le manipulateur SCARA. Il est aisé de remarquer géométriquement que la seule singularité du bras du robot SCARA est quand le coude est entièrement prolongé ou entièrement rétracté.

On a la matrice jacobienne :

$$J = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} & 0 & 0 \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

Le déterminant de J est sous la forme :

$$\det(j) = [(-l_1 S_1 - l_2 S_{12})(l_2 C_{12}) - (l_1 C_1 + l_2 C_{12})(-l_2 S_{12})] \dots \dots \dots (4.105)$$

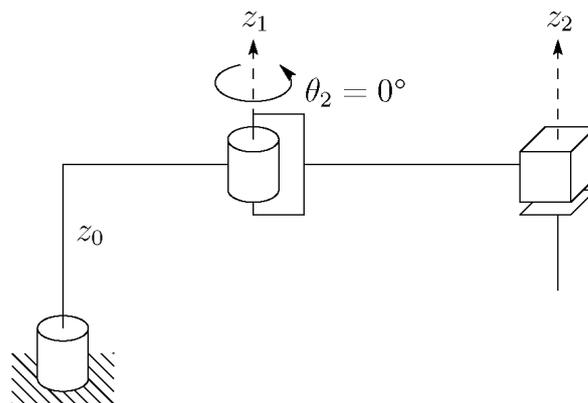


Figure 4.61 : La singularité du robot SCARA.

On évite $\theta_2 = 0, \pi$

On va étudier la variation de $\det(J)$ en fct de θ_2 pour différentes valeurs de θ_1

Résultat :

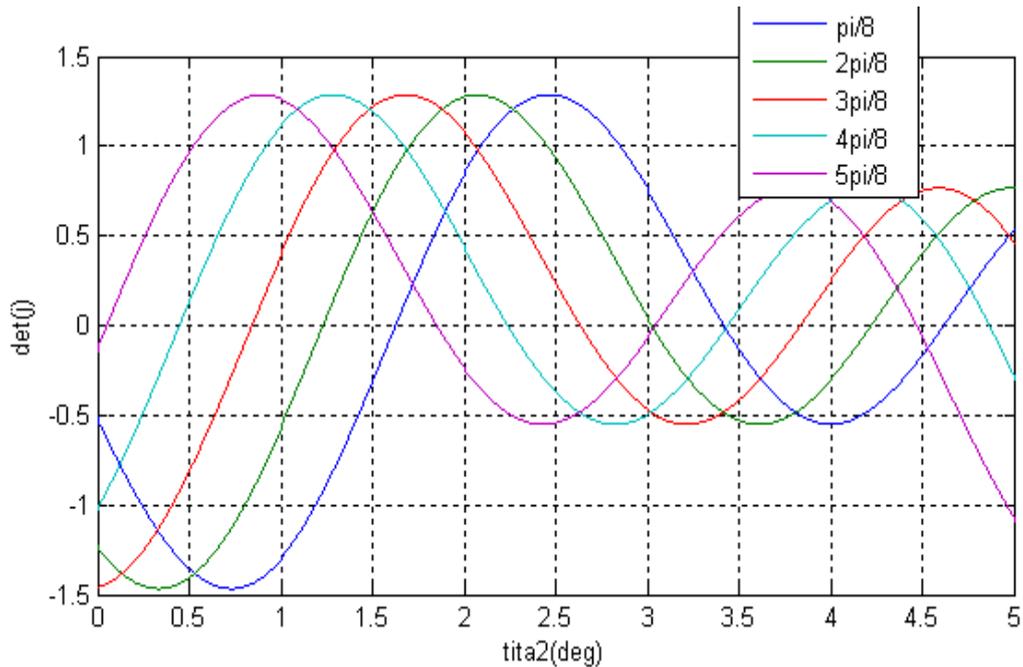


Figure 4.62 : La variation de $\det(J)$ en fct de θ_2 pour différentes valeurs de θ_1

Pour $\theta_1 = 22.5^\circ$ on évite que $\theta_2 = 3.42^\circ$ car le $\det(J)=0$.c'est la singularité du robot SCARA. [36].

4-9 La manipulabilité :

Plusieurs facteurs doivent être pris en compte lors du choix du mécanisme et de la taille du robot manipulateur durant la phase de conception, ou lors de la détermination de la posture du manipulateur dans l'espace de travail pour effectuer une tâche opérationnelle. Un facteur important parmi ceux ci est la facilité de changement arbitraire de la position et de l'orientation de l'organe terminal.

Yoshikawa a développé une approche pour l'évaluation quantitative de la capacité du manipulateur des points de vues cinématique et dynamique. L'auteur a introduit le concept de l'ellipsoïde de manipulabilité à partir duquel il a défini quelques mesures pour caractériser la capacité de manipulation [39].

La manipulabilité est une quantité qui représente la capacité du manipulateur à se déplacer autour d'une configuration donnée. Elle permet d'analyser la cinématique des systèmes mécaniques. La manipulabilité est utilisée dans diverses applications : lors de la conception des mécanismes, dans la planification de trajectoire ou lors de la commande des systèmes mécaniques.

4-9-1 Application de la manipulabilité au robot SCARA :

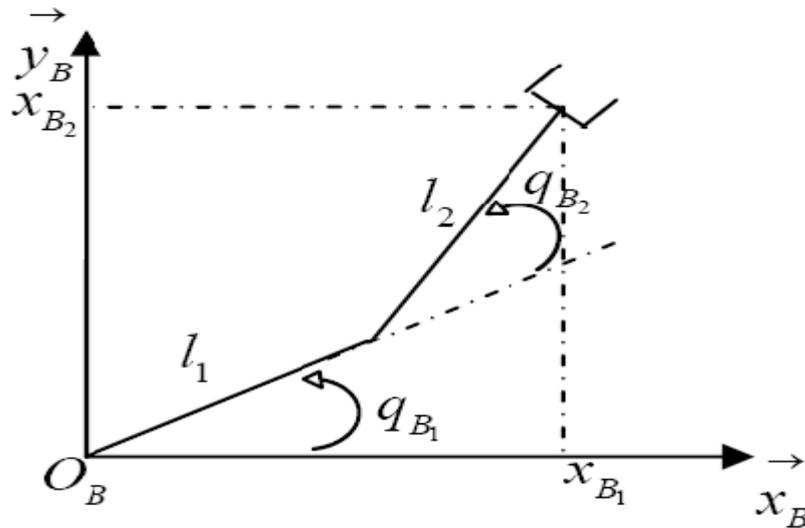


Figure 4-63 : les variables articulaires dans le robot SCARA.

Remarque :

On utilise la notation q_{B_1} et q_{B_2} au lieu θ_1 et θ_2 , car cette notation est plus utilisée lors de l'étude de la manipulabilité.

La configuration du bras manipulateur est donnée par les angles de rotation q_{B_1} et q_{B_2} des deux liaisons. Si l'on considère uniquement la position de l'organe terminal, (notée OT), donnée par :

$$\begin{cases} x_{B_1} = l_1 c_1 + l_2 c_{12} \\ x_{B_2} = l_1 s_1 + l_2 s_{12} \end{cases} \dots\dots\dots (4.106)$$

Avec :

$$c_1 = \cos(q_{B_1}) \dots\dots\dots (4.107)$$

$$s_1 = \sin(q_{B_1}) \dots\dots\dots (4.108)$$

$$c_{12} = \cos(q_{B_1} + q_{B_2}) \dots\dots\dots (4.109)$$

$$s_{12} = \sin(q_{B_1} + q_{B_2}) \dots\dots\dots (4.110)$$

Et l_1 et l_2 les longueurs des bras.

Dans ce cas la matrice jacobienne est sous la forme :

$$J_B = \begin{bmatrix} -(l_1 s_1 + l_2 s_{12}) & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix} \dots\dots\dots (4.111)$$

On examine l'évolution de la manipulabilité quand la position de l'OT suit une ligne droite.

Résultat :

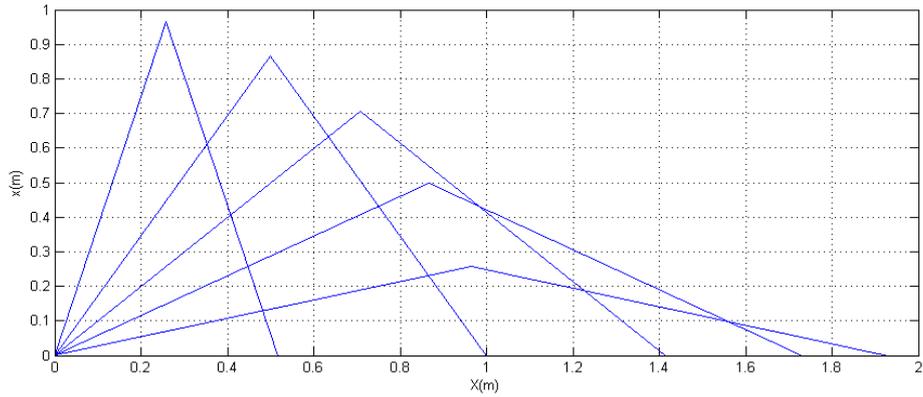


Figure 4-64 : la manipulabilité d'un robot SCARA.

Le graphe montre la facilité de changement arbitraire de la position et l'orientation de l'organe terminal. [39] [40]

La forme ellipsoïde caractérise les forces de l'organe terminal qui peuvent être produites par les couples des articulations, et cela pour les deux postures. [40] [41]

4-9-2 mesure de la manipulabilité du robot SCARA :

L'équation de la manipulabilité pour un robot SCARA redondant est donnée par la relation suivante [17] :

$$w = \sqrt{\det(J * J')} \dots\dots\dots(4.112)$$

Pour le calcul de w (voir *appendice*).

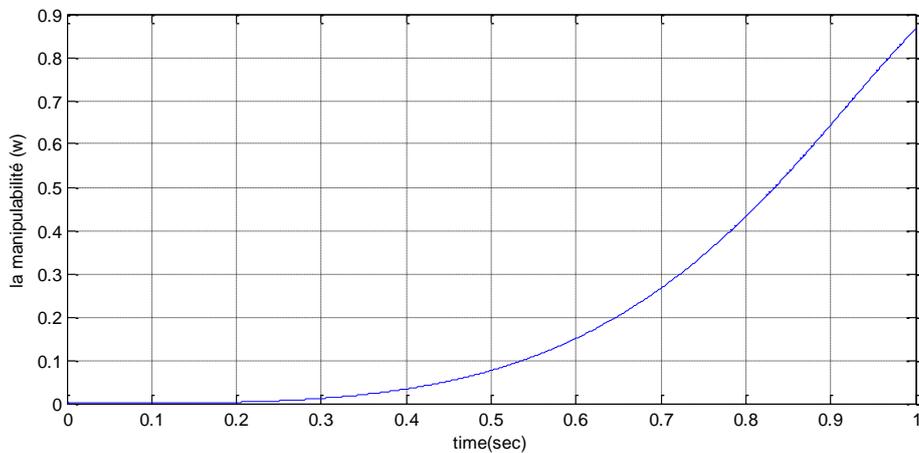


Figure 4.65 : Mesure de la manipulabilité en fonction du temps.

4-9-3 Indice de longueur structurelle du robot SCARA :

L'équation de l'indice de longueur structurelle pour un robot SCARA est donnée par la relation suivante [17] :

$$Q_L = \frac{l + d_3}{\sqrt[3]{\pi l^2 d_3}} \dots\dots\dots(4.113)$$

Sachant que :

$$l = l_1 + l_2 \dots\dots\dots(4.114)$$

$$d_3 = 0.635 \dots\dots\dots(4.115)$$

Donc :

$$Q_L = 1.31 \dots\dots\dots(4.116)$$

Une bonne conception d'un manipulateur se distingue par: un grand espace de travail (une valeur minimal de Q_L) caractérisée par une valeur élevée de w .[17]

Dans notre étude : $w = 0.86$ et $Q_L = 1.31$, de ce faite le robot est de bonne conception.

4-10 Calcul des couples d'articulations :

Le robot SCARA est supposé un robot plan dont la masse de l'organe terminal est négligeable par rapport aux deux masses des liens 1 et 2 [42] [43]

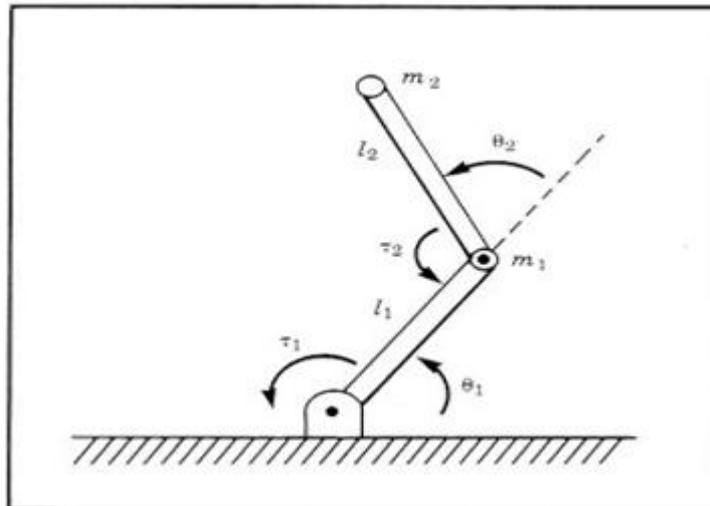


Figure 4.66 : Schéma dynamique du robot SCARA.

4-10-1 Paramètre des liens :

A l'aide du logiciel Solidworks (option : propriété de masse), on peut tirer les paramètres des liens du robot SCARA :

On considère que la base du robot possède une hauteur initiale l_0 :

N° du lien	Longueur du lien (m)	Masse du lien (Kg)	Position du centre de masse /repère du lien (m)	Tenseur d'inertie /repère du lien ($Kg.m^2$)
1	1	16.92	(0.5, 0,0)	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 5.29 & 0 \\ 0 & 0 & 5.29 \end{bmatrix}$
2	1	16.92	(0.5, 0,0)	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 5.29 & 0 \\ 0 & 0 & 5.29 \end{bmatrix}$
3	0.63	1.215	(0, 0,0.315)	$\begin{bmatrix} 0.1303 & 0 & 0 \\ 0 & 0.1303 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

Tableau 4.10.1 : Paramètres des liens d'une structure SCARA.

4-10-2 Calcul les couples des articulations :

4-10-2-1 par la méthode de Lagrange :

Les équations de Lagrange opèrent à partir de l'énergie cinétique et l'énergie potentielle d'un système, le lagrangien s'écrit sous la forme :

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} = \tau_i \dots \dots \dots (4.117)$$

$$L = K - V \dots \dots \dots (4.118)$$

K : Énergie cinétique de l'organe terminal.

V : Énergie potentielle de l'organe terminal.

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

$$K = \frac{1}{2} \dot{q}^T M(q) \dot{q} \dots \dots \dots (4.119)$$

$M(q)$: est la matrice d'inertie de dimensions (n*n) symétrique définie positive.

L'énergie potentielle du système est sous la forme :

$$V_i = -m_i g^T T_i^i S_i \dots \dots \dots (4.120)$$

$g = [g_x, g_y, g_z, 0]^T$: Vecteur des accélérations de la gravité exprimé dans R_0

Le modèle obtenu peut être mis sous la forme matricielle suivante

$$\tau = M(q)\ddot{q} + N(q, \dot{q})\dot{q} + G(q) \dots \dots \dots (4.121)$$

Ou

$$M_{ij} = \sum_{k=\max(i,j)}^n tr \left(\frac{\partial {}^0T_k}{\partial q_j} H_k \frac{\partial {}^0T_k^T}{\partial q_i} \right) \dots \dots \dots (4.122)$$

$$G(q) = \sum_{j=1}^n m_j g^T \frac{\partial {}^0T_j}{\partial q_i} S_j \dots \dots \dots (4.123)$$

$M(q)$: Matrice d'inertie du manipulateur, symétrique et régulière définie positive [41] [44]

$N(q, \dot{q})\dot{q}$: Vecteur des termes de Coriolis et centrifuges.

$G(q)$: Vecteur des actions de la gravité.

0T_k : La matrice de transformation homogène.

${}^i S_i = [{}^i S_{ix}, {}^i S_{iy}, {}^i S_{iz}]^T$: Position du centre de masse (m) du corps i exprimée en R_i

$H_i = \int {}^i r^i r^T dm_i$: Matrice des pseudo-inerties du corps i.

$$H_i = \begin{bmatrix} \frac{-{}^i I_{ixx} + {}^i I_{iyy} + {}^i I_{izz}}{2} & I_{ixy} & I_{ixz} & m_i S_{ix} \\ I_{ixy} & \frac{{}^i I_{ixx} - {}^i I_{iyy} + {}^i I_{izz}}{2} & I_{iyz} & m_i S_{iy} \\ I_{ixz} & I_{iyz} & \frac{{}^i I_{ixx} + {}^i I_{iyy} - {}^i I_{izz}}{2} & m_i S_{iz} \\ m_i S_{ix} & m_i S_{iy} & m_i S_{iz} & m_i \end{bmatrix} \dots \dots \dots (4.124)$$

Calcul la matrice des pseudo-inerties du corps i :

A l'aide du tableau 4-10-1 on peut tirer :

$$H_i = \begin{bmatrix} 5.29 & 0 & 0 & 8.46 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 8.46 & 0 & 0 & 16.92 \end{bmatrix} \dots \dots \dots (4.125)$$

$${}^i I_i = \begin{bmatrix} {}^i I_{ixx} & -{}^i I_{ixy} & -{}^i I_{ixz} \\ -{}^i I_{ixy} & {}^i I_{iyy} & -{}^i I_{iyz} \\ -{}^i I_{ixz} & -{}^i I_{iyz} & {}^i I_{izz} \end{bmatrix} : \text{Tenseur d'inertie du corps i en } o_i \text{ exprimé dans } R_i .$$

${}^i I_{ixx} = \int ({}^i r_y^2 + {}^i r_z^2) dm_i$: Moment d'inertie.

${}^i I_{ixy} = \int {}^i r_x {}^i r_y dm_i$: Produit d'inertie.

Application :

$q = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}$ Vecteur des variables articulaires.

$\tau = \begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix}$ Vecteur des couples généralisés.

4-10-2-1-1 La matrice d'inertie :

$$M(q) = \begin{pmatrix} m_2 l_2^2 + 2m_2 l_2 l_1 c_2 + (m_2 + m_1) l_1^2 & m_2 l_2^2 + m_2 l_2 l_1 c_2 \\ m_2 l_2^2 + m_2 l_2 l_1 c_2 & m_2 l_2^2 \end{pmatrix} \dots \dots \dots (4.126)$$

4-10-2-1-2 Matrice de Coriolis et de centrifuge :

$$N(q, \dot{q}) = \begin{pmatrix} -2m_2 l_2 l_1 s_2 \dot{\theta}_2 & -m_2 l_2 l_1 s_2 \dot{\theta}_2 \\ m_2 l_2 l_1 s_2 \dot{\theta}_1 & 0 \end{pmatrix} \dots \dots \dots (4.127)$$

4-10-2-1-3 Vecteurs des forces de gravité :

$$G(q) = \begin{pmatrix} m_2 l_2 g c_{12} + (m_2 + m_1) l_1 g c_1 \\ m_2 l_2 g c_{12} \end{pmatrix} \dots \dots \dots (4.128)$$

On applique l'équation (4.121) on trouve :

$$\tau_1 = m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l_2 c_2 (2\ddot{\theta}_1 + \ddot{\theta}_2) + (m_1 + m_2) l_1^2 \ddot{\theta}_1 - m_2 l_1 l_2 s_2 \dot{\theta}_2^2 - 2m_2 l_1 l_2 s_2 \dot{\theta}_1 \dot{\theta}_2 + m_2 l_2 g c_{12} + (m_1 + m_2) l_1 g c_1 \dots \dots \dots (4.129)$$

$$\tau_2 = m_2 l_1 l_2 c_2 \ddot{\theta}_1 + m_2 l_1 l_2 s_2 \dot{\theta}_1^2 + m_2 l_2 g c_{12} + m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) \dots \dots \dots (4.130)$$

4-10-2-2 Vérification des résultats par l'algorithme de Newton-Euler :

On applique l'algorithme de newton-Euler on obtient les résultats suivant :

L'organe terminal est de masse négligeable donc :

$$f_3 = 0 \dots \dots \dots (4.131)$$

$$n_3 = 0 \dots \dots \dots (4.132)$$

La rotation de la base du robot est nulle donc :

$$\omega_0 = 0 \dots\dots\dots(4.133)$$

$$\dot{\omega}_0 = 0 \dots\dots\dots(4.134)$$

$${}^0\dot{v}_0 = gY_0 \dots\dots\dots(4.135)$$

La rotation entre deux liens successives est donné par :

$${}^i_{i+1}R = \begin{bmatrix} c_{i+1} & -s_{i+1} & 0 \\ s_{i+1} & c_{i+1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \dots\dots\dots(4.136)$$

$${}^{i+1}_iR = \begin{bmatrix} c_{i+1} & s_{i+1} & 0 \\ -s_{i+1} & c_{i+1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \dots\dots\dots(4.137)$$

Réursive arrière du lien 1 :

$$\omega_1 = \dot{\theta}_1 {}^1Z_1 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} \dots\dots\dots(4.138)$$

$${}^1\dot{\omega}_1 = \ddot{\theta}_1 {}^1Z_1 = \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_1 \end{bmatrix} \dots\dots\dots(4.139)$$

$${}^1\dot{v}_1 = \begin{bmatrix} c_1 & -s_1 & 0 \\ s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ g \\ 0 \end{bmatrix} = \begin{bmatrix} gs_1 \\ gc_1 \\ 0 \end{bmatrix} \dots\dots\dots(4.140)$$

$${}^1\dot{v}_{C1} = \begin{bmatrix} 0 \\ l_1\ddot{\theta}_1 \\ 0 \end{bmatrix} + \begin{bmatrix} -l_1\dot{\theta}_1^2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} gs_1 \\ gc_1 \\ 0 \end{bmatrix} = \begin{bmatrix} -l_1\dot{\theta}_1^2 + gs_1 \\ l_1\ddot{\theta}_1 + gc_1 \\ 0 \end{bmatrix} \dots\dots\dots(4.141)$$

$${}^1F_1 = \begin{bmatrix} -m_1l_1\dot{\theta}_1^2 + m_1gs_1 \\ m_1l_1\ddot{\theta}_1 + m_1gc_1 \\ 0 \end{bmatrix} \dots\dots\dots(4.142)$$

$${}^1N_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \dots\dots\dots(4.143)$$

Réursive arrière du lien 2 :

$${}^2\omega_2 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix} \dots\dots\dots(4.144)$$

$${}^2\dot{\omega}_2 = \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_1 + \ddot{\theta}_2 \end{bmatrix} \dots\dots\dots (4.145)$$

$${}^2\dot{v}_2 = \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -l_1\dot{\theta}_1^2 + gs_1 \\ l_1\ddot{\theta}_1 + gc_1 \\ 0 \end{bmatrix} = \begin{bmatrix} l_1\ddot{\theta}_1 s_2 - l_1\dot{\theta}_1^2 c_2 + gs_{12} \\ l_1\ddot{\theta}_1 c_2 + l_1\dot{\theta}_1^2 s_2 + gc_{12} \\ 0 \end{bmatrix} \dots\dots\dots (4.146)$$

$${}^2\dot{v}_{c2} = \begin{bmatrix} 0 \\ l_2(\ddot{\theta}_1 + \ddot{\theta}_2) \\ 0 \end{bmatrix} + \begin{bmatrix} -l_2(\dot{\theta}_1 + \dot{\theta}_2)^2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} l_1\ddot{\theta}_1 s_2 - l_1\dot{\theta}_1^2 c_2 + gs_{12} \\ l_1\ddot{\theta}_1 c_2 + l_1\dot{\theta}_1^2 s_2 + gc_{12} \\ 0 \end{bmatrix} \dots\dots\dots (4.147)$$

$${}^2F_2 = \begin{bmatrix} m_2 l_1 \ddot{\theta}_1 s_2 - m_2 l_1 \dot{\theta}_1^2 c_2 + m_2 g s_{12} - m_2 l_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 \\ m_2 l_1 \ddot{\theta}_1 c_2 + m_2 l_1 \dot{\theta}_1^2 s_2 + m_2 g c_{12} + m_2 l_2 (\ddot{\theta}_1 + \ddot{\theta}_2) \\ 0 \end{bmatrix} \dots\dots\dots (4.148)$$

$${}^2N_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \dots\dots\dots (4.149)$$

Réursive avant du lien 2 :

$${}^2f_2 = {}^2F_2 \dots\dots\dots (4.150)$$

$${}^2n_2 = \begin{bmatrix} 0 \\ 0 \\ m_2 l_1 l_2 \ddot{\theta}_1 c_2 + m_2 l_1 l_2 \dot{\theta}_1^2 s_2 + m_2 l_2 g c_{12} + m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) \end{bmatrix} \dots\dots\dots (4.151)$$

Réursive avant du lien 1 :

$${}^1f_1 = \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} m_2 l_1 \ddot{\theta}_1 s_2 - m_2 l_1 \dot{\theta}_1^2 c_2 + m_2 g s_{12} - m_2 l_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 \\ m_2 l_1 \ddot{\theta}_1 c_2 + m_2 l_1 \dot{\theta}_1^2 s_2 + m_2 g c_{12} + m_2 l_2 (\ddot{\theta}_1 + \ddot{\theta}_2) \\ 0 \end{bmatrix} \begin{bmatrix} -m_1 l_1 \dot{\theta}_1^2 + m_1 g s_1 \\ m_1 l_1 \ddot{\theta}_1 + m_1 g c_1 \\ 0 \end{bmatrix} \dots\dots\dots (4.152)$$

$${}^1n_1 = \begin{bmatrix} 0 \\ 0 \\ m_2 l_1 l_2 \ddot{\theta}_1 c_2 + m_2 l_1 l_2 \dot{\theta}_1^2 s_2 + m_2 l_2 g c_{12} + m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) \end{bmatrix} \dots\dots\dots (4.153)$$

$$\begin{bmatrix} 0 \\ 0 \\ m_1 l_1^2 \ddot{\theta}_1 + m_1 l_1 g c_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ m_2 l_1^2 \ddot{\theta}_1 - m_2 l_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + m_2 l_1 g s_2 s_{12} + m_2 l_1 l_2 c_2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 g c_2 c_{12} \end{bmatrix}$$

$$\tau_i = {}^i n_i^T Z_i \dots\dots\dots (4.154)$$

$$\tau_1 = m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l_2 c_2 (2\ddot{\theta}_1 + \ddot{\theta}_2) + (m_1 + m_2) l_1^2 \ddot{\theta}_1 - m_2 l_1 l_2 s_2 \dot{\theta}_1^2 - 2m_2 l_1 l_2 s_2 \dot{\theta}_1 \dot{\theta}_2 + m_2 l_2 g c_{12} + (m_1 + m_2) l_1 g c_1 \dots\dots\dots (4.155)$$

$$\tau_2 = m_2 l_1 l_2 c_2 \ddot{\theta}_1 + m_2 l_1 l_2 s_2 \dot{\theta}_1^2 + m_2 l_2 g c_{12} + m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) \dots (4.156)$$

4-10-2-3 Vérification des résultats par la méthode de la mécanique classique :

$$K_1 = \frac{1}{2} m_1 l_1^2 \dot{\theta}_1^2 \dots (4.157)$$

$$V_1 = m_1 g l_1 \sin \theta_1 \dots (4.158)$$

$$x_2 = l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) \dots (4.159)$$

$$y_2 = l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) \dots (4.160)$$

$$\dot{x}_2 = -l_1 \dot{\theta}_1 \sin \theta_1 - l_2 (\dot{\theta}_1 + \dot{\theta}_2) \sin(\theta_1 + \theta_2) \dots (4.161)$$

$$\dot{y}_2 = l_1 \dot{\theta}_1 \cos \theta_1 + l_2 (\dot{\theta}_1 + \dot{\theta}_2) \cos(\theta_1 + \theta_2) \dots (4.162)$$

$$v_2^2 = \dot{x}_2^2 + \dot{y}_2^2 \dots (4.163)$$

$$v_2^2 = l_1^2 \dot{\theta}_1^2 + l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + 2l_1 l_2 (\dot{\theta}_1 + \dot{\theta}_1 \dot{\theta}_2) \cos \theta_2 \dots (4.164)$$

$$K_2 = \frac{1}{2} m_2 v_2^2 \dots (4.165)$$

$$K_2 = \frac{1}{2} m_2 l_1^2 \dot{\theta}_1^2 + \frac{1}{2} m_2 l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + m_2 l_1 l_2 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) \cos \theta_2 \dots (4.166)$$

$$V_2 = m_2 g y_2 = m_2 g [l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2)] \dots (4.167)$$

$$L = K - V = K_1 + K_2 - V_1 - V_2 \dots (4.168)$$

$$L = \frac{1}{2} (m_1 + m_2) l_1^2 \dot{\theta}_1^2 + \frac{1}{2} m_2 l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + m_2 l_1 l_2 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) \cos \theta_2 \dots (4.169)$$

$$- (m_1 + m_2) g l_1 \sin \theta_1 - m_2 g l_2 \sin(\theta_1 + \theta_2)$$

$$\frac{\partial L}{\partial \dot{\theta}_1} = (m_1 + m_2) l_1^2 \dot{\theta}_1 + m_2 l_2^2 (\dot{\theta}_1 + \dot{\theta}_2) + m_2 l_1 l_2 (2\dot{\theta}_1 + \dot{\theta}_2) \cos \theta_2 \dots (4.170)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_1} = (m_1 + m_2) l_1^2 \ddot{\theta}_1 + m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) \dots (4.171)$$

$$+ m_2 l_1 l_2 (2\ddot{\theta}_1 + \ddot{\theta}_2) \cos \theta_2 - m_2 l_1 l_2 (2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) \sin \theta_2$$

$$\frac{\partial L}{\partial \theta_1} = - (m_1 + m_2) g l_1 \cos \theta_1 - m_2 g l_2 \cos(\theta_1 + \theta_2) \dots (4.172)$$

$$\frac{\partial L}{\partial \dot{\theta}_2} = m_2 l_2^2 (\dot{\theta}_1 + \dot{\theta}_2) + m_2 l_1 l_2 \dot{\theta}_1 \cos \theta_2 \dots (4.173)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_2} = m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l_2 \ddot{\theta}_1 \cos \theta_2 - m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin \theta_2 \dots (4.174)$$

$$\frac{\partial L}{\partial \theta_2} = -m_2 l_1 l_2 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) \sin \theta_2 - m_2 g l_2 \cos(\theta_1 + \theta_2) \dots\dots\dots(4.175)$$

$$\tau_1 = m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l_2 c_2 (2\ddot{\theta}_1 + \ddot{\theta}_2) \dots\dots\dots(4.176)$$

$$+ (m_1 + m_2) l_1^2 \ddot{\theta}_1 - m_2 l_1 l_2 s_2 \dot{\theta}_2^2 - 2m_2 l_1 l_2 s_2 \dot{\theta}_1 \dot{\theta}_2 + m_2 l_2 g c_{12} + (m_1 + m_2) l_1 g c_1$$

$$\tau_2 = m_2 l_1 l_2 c_2 \ddot{\theta}_1 + m_2 l_1 l_2 s_2 \dot{\theta}_1^2 + m_2 l_2 g c_{12} + m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) \dots\dots\dots(4.177)$$

Résultats de simulation :

Posture1 :

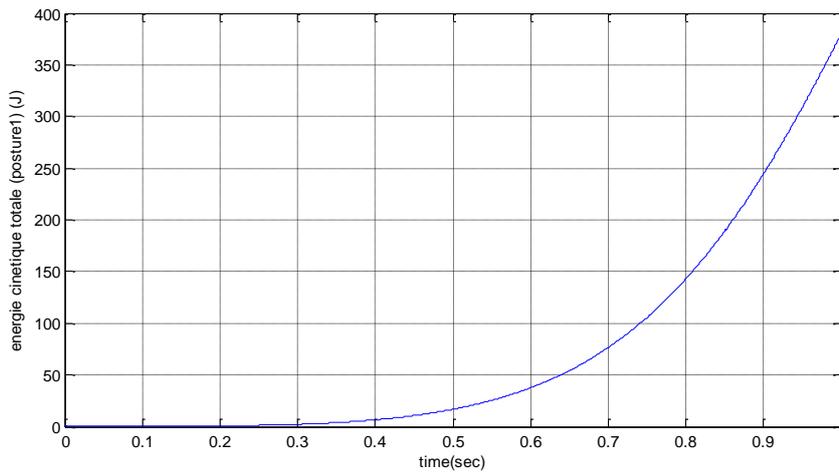


Figure 4.67: la variation de l'énergie cinétique totale (*posture1*).

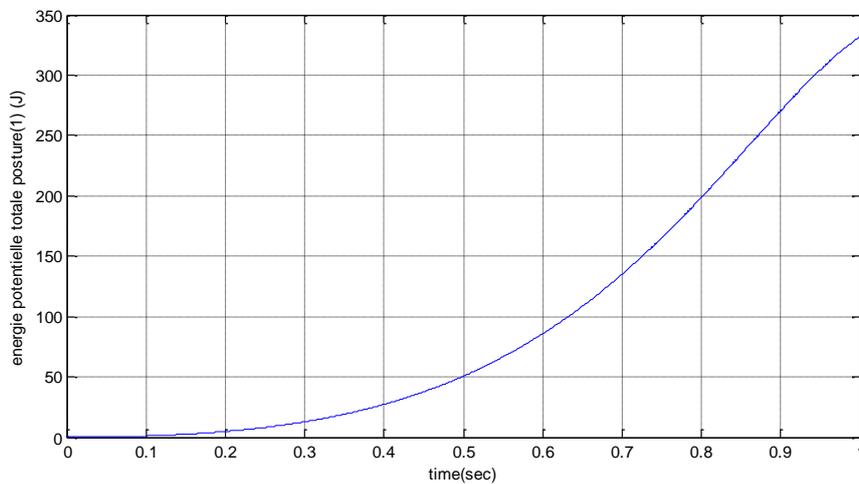


Figure 4.68: la variation de l'énergie potentielle totale (*posture1*).

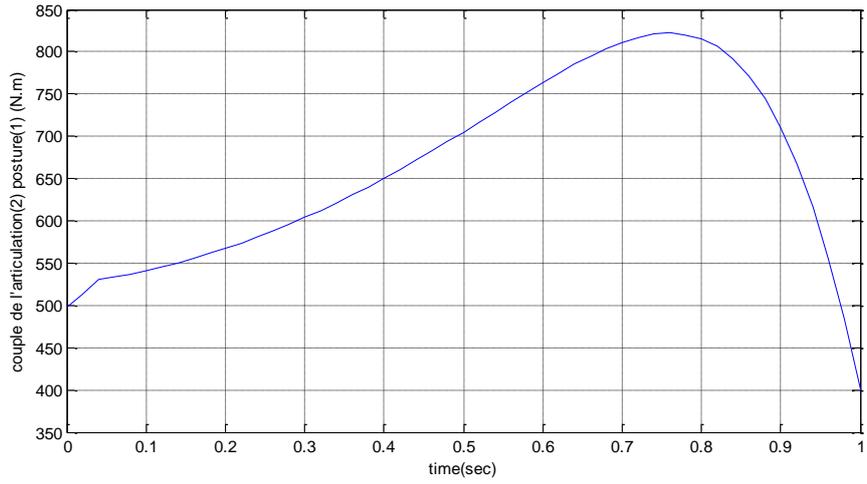


Figure 4.69: le couple de l'articulation1 (*posture1*)

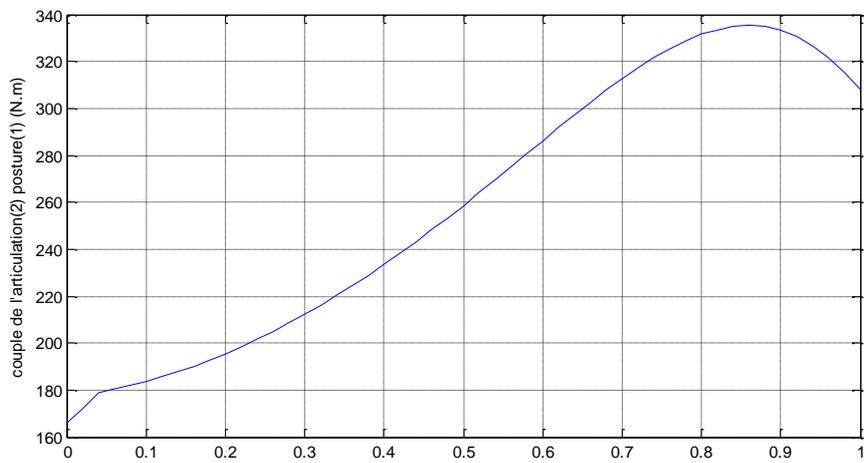


Figure 4.70 : le couple de l'articulation2 (*posture1*).

Posture2 :

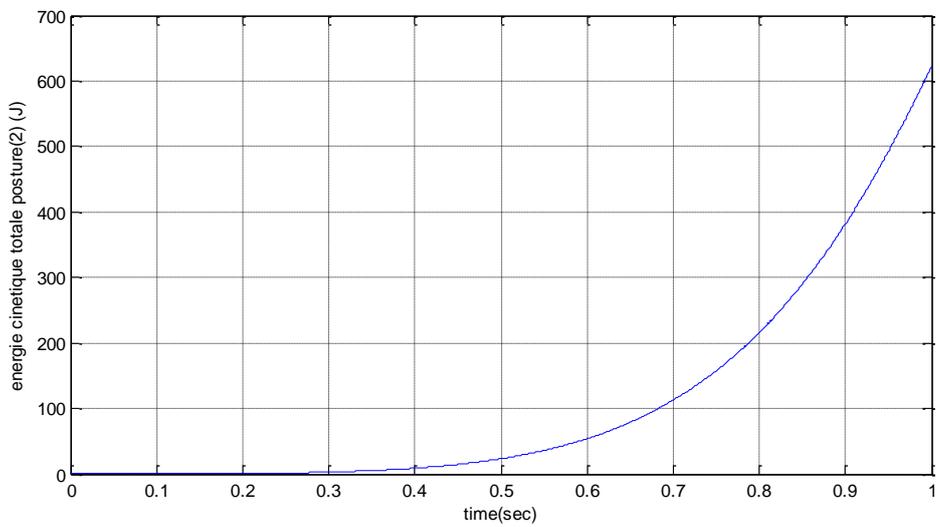


Figure 4.71 : la variation de l'énergie cinétique totale (*posture2*).

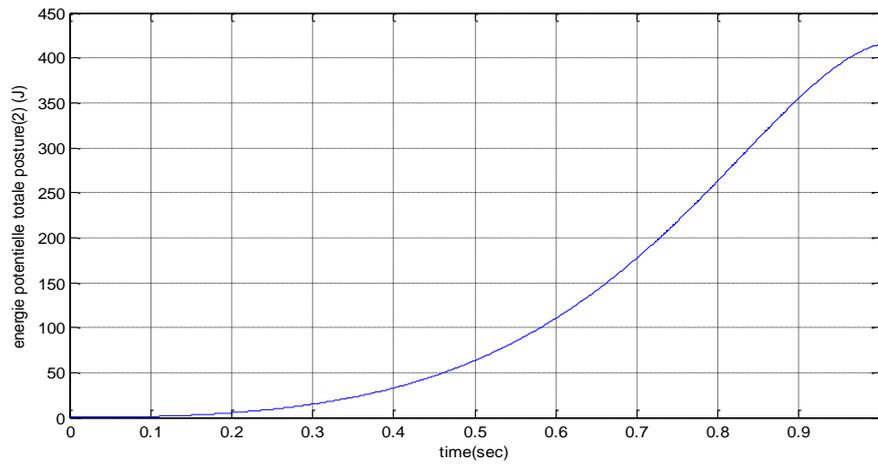


Figure 4.72 : la variation de l'énergie potentielle totale (*posture2*).

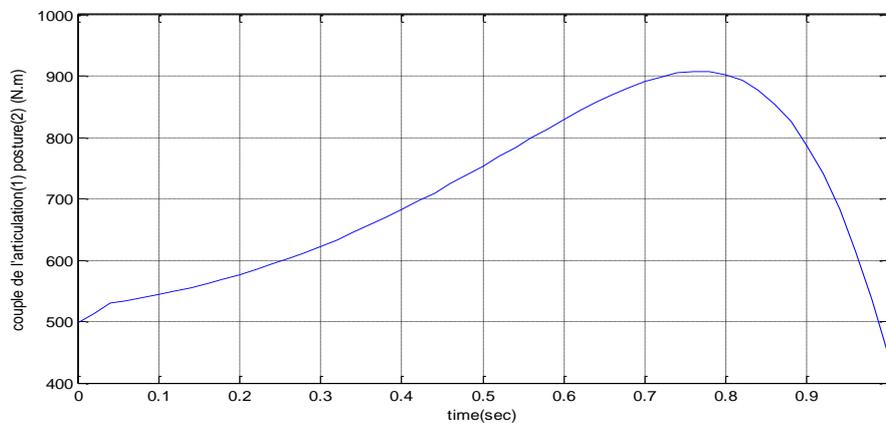


Figure 4.73: le couple de l'articulation1 (*posture2*)

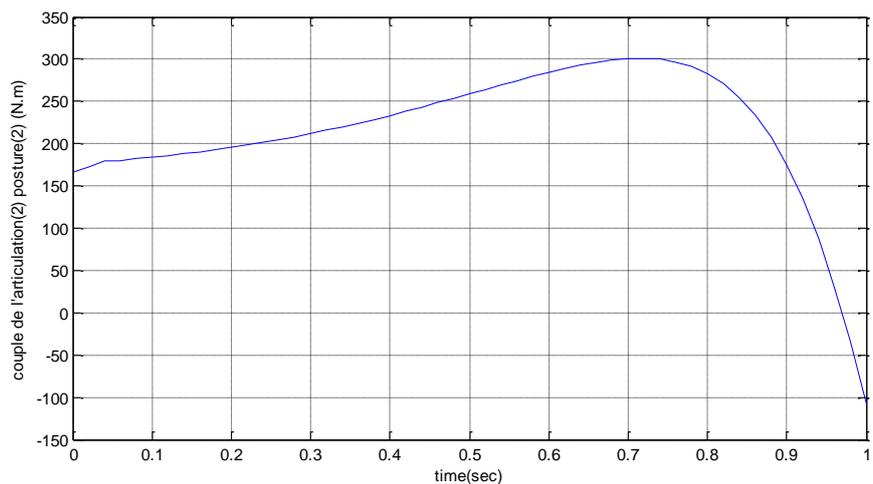


Figure 4.74 : le couple de l'articulation2 (*posture2*)

4-11 Comparaison entre les deux postures :

On remarque d'après les résultats de simulation obtenus que :

- la variation d'énergie cinétique totale calculée lors de *posture2* est supérieure à l'énergie cinétique totale calculée lors de *posture1* avec : $\Delta K = 243.02J$.

- la variation d'énergie potentielle totale calculée lors de *posture2* est supérieure a l'énergie potentielle totale calculée lors de *posture1* avec $\Delta V = 82.95J$.

Cette différence dans la variation des énergies, cinétique et potentielle, est due aux différences constatée dans la conclusion du paragraphe 4.5.7.

On remarque aussi, que le couple de l'articulation1 calculé lors de *posture2* a t=1s est $\tau_1 = 446.33N.m$ avec un max $\tau_1 = 900N.m$ par contre le couple de l'articulation1 calculé lors de *posture1* est $\tau_1 = 395.18N.m$ avec un max $\tau_1 = 820N.m$.

Par ailleurs, le couple de l'articulation2 calculé lors de *posture2* a t=1s est $\tau_2 = -127N.m$ avec un max $\tau_2 = 300N.m$ par contre le couple de l'articulation2 calculé lors de *posture1* est $\tau_2 = 319.65N.m$ avec un max $\tau_2 = 330N.m$.

Cette étude nous révèle que :

* Le robot est amené a consommer beaucoup plus d'énergie quand il est sollicité a travailler selon la *posture2*, comparativement a son usage, pour une même trajectoire et même position désirée, selon la *posture1*.

* Le choix de *posture1* est tout désigné dans l'éventualité d'une économie d'énergie. [41]

4-12 Conclusion :

Dans ce chapitre on a utilisé les deux logiciels SolidWorks et Matlab/Simulink pour vérifier la théorie du (tableau 4.5.4), et la simulation du mouvement du robot réalisée sous logiciels précités.

La vérification des résultats obtenus par deux logiciels SolidWorks et Matlab nous a permis de valoriser qualitativement et mettre en relief la pertinence du modèle étudié.

Dans notre cas, nous pouvons conclure au vue des résultats obtenus par l'analyse comparative des deux postures que les énergies dissipées lors du mouvement du robot selon la première posture sont très inférieures à celles qui seraient dissipées si le robot effectue le mouvement selon la deuxième posture et ce pour une même trajectoire et une même durée dans le temps selon les présentations énumérées ci-dessus.

Donc lors du travail du robot, le choix de la posture est tout désigné dans l'éventualité d'une économie d'énergie ce qui est toujours préconisé et recommandé.

CONCLUSION

La modélisation est une étape cruciale dans la robotique. Qu'il s'agit du dimensionnement des composants du robot ou de la planification d'une trajectoire de l'organe terminal, un modèle régissant le comportement du robot et tenant compte de tous ses paramètres s'avère indispensable. La modélisation d'une scène de manipulation d'un objet par des robots comporte, aussi bien pour l'objet que pour les robots les modèles suivants : le modèle géométrique, le modèle cinématique et le modèle dynamique. Ces modèles peuvent être directes, si les grandeurs cherchées sont exprimées dans l'espace opérationnel ou, inverse, si elles sont exprimées dans l'espace articulaire...

Dans une proposition de démarche à suivre lors de l'étude d'un sujet de robotique, notre travail porte particulièrement sur la modélisation du robot SCARA. Une recherche bibliographique approfondie nous a aidés à exposer les différents concepts théoriques et formalismes nécessaires à une compréhension assez large de ce domaine.

Notre thème dont l'intitulé est sus indiqué, porte sur : << la synthèse, analyse et modélisation des robots manipulateurs>>, ce travail est subdiviser en 3 axes différents et complémentaires.

Le modèle cinématique a fait l'objet de plusieurs travaux de recherches dont, pratiquement tous s'articulaient sur l'utilisation de l'un des deux modèles suivants : Le modèle cinématique direct et le modèle cinématique inverse. Une présentation des deux méthodes nous a menés à modéliser le robot et calculer toutes ses variables articulaires.

Pour la cinématique du robot on a utilisé le modèle cinématique direct afin de calculer la matrice de transformation homogène et le paramétrage de D-H qui seront utilisées pour le modèle cinématique inverse. Ce dernier nous permet de calculer toutes les variables articulaires et de ce fait modéliser le robot et calculer tous les paramètres nécessaires.

A la suite des approches théoriques du modèle présenté dans le chapitre 2, s'en est suivi une application dans le chapitre 4, consistant a la génération du mouvement par le robot traité.

Le modèle dynamique utilise les modèles : géométrique et cinématique pour calculer les efforts auxquels les différents liens sont assujettis. Le chapitre 2 a été présenté dans une chronologie favorisant la compréhension au lecteur. Commenant par la description des ces modèles, ensuit, une méthode algorithmique pour le calcul du jacobien utilisé pour le calcul des vitesses et accélérations.

Concernant la fiabilité de nos résultats, une confrontation avec les résultats donnés en littérature nous renseigne sur la justesse de notre étude.

Notre travail peut être considéré comme une base de départ pour une recherche approfondie dans le domaine de la robotique. Dans un travail futur, une étude complémentaire peut être envisagée en proposant des méthodes pour :

- La résolution du modèle cinématique direct et inverse pour les structures complexes.
- Utilisation des équations de mouvements a degré plus élevés pour une meilleure approche des résultats.
- Calculer la dynamique du robot SCARA a couple électrique.
- Tenir compte de la masse de l'organe terminal, du frottement, des inerties des actionneurs et les masses des articulations pour mieux approcher les résultats du modèle dynamique à la réalité.

BIBLIOGRAPHIE

- [1] Das, M.T. and L.C.Dulger, 2005. “Mathematical modelling, simulation and experimental verification of a scara robot”, *Simulation Modelling Practice and Theory* 13, pp.257–271
- [2] Kazi, A., G. Merk, M. Otter, and H. Fan, 2002. . “Design optimization of industrial robots using the Modelica multi-physics modeling language”, *Proceedings of the 33rd ISR (International Symposium on Robotics) October 7 – 11.*
- [3] *Génération et identification des lois du mouvement’*. Technical report.
- [4] R. Gorez : *Notes du cours d’Introduction à la robotique, 1991’*. Technical report, UCL,Louvain-la-Neuve
- [5] [http ://www.jautomatise.com](http://www.jautomatise.com)’. Technical report
- [6] *Robotique. Aspects fondamentaux*, J.-P. Lallemand, S. Zegloul, Masson 1994, 312 pages
- [7] C. VIBET, "Robots Principes et Contrôle", Editions Ellipses, 1988.
- [8] *Cours de robotique*, J.-L. Ferrier, DESS ASC
- [9] *Robots. Principes et contrôle*, C. Vibet, Ellipses 1987, 207 pages
- [10] *Introduction to Robotics Mechanics and Control*, 2 edition, John J. Craig, Addison-Wesley Publishing Company,, 450 pages.
- [11] Mayer St-Onge, B. et Gosselin, C.M. , 1996, *Singularity analysis and representation*
- [12] Chablat D., Wenger Ph., *Working Modes and Aspects in Fully-Parallel Manipulator*, *Proceedings IEEE International Conference on Robotics and Automation*
- [13] **ELEMENTS DE ROBOTIQUE ASPECTS MECANIQUES**
Pierre DUYSINX Ingénierie des Véhicules Terrestres
Département PROMETHE, (Productique, Mécanique et Thermodynamique)
- [14] Université Catholique de Louvain Faculté des Sciences Appliquées
Génération et Parallélisation des Équations du Mouvement de Systèmes Multicorps par l'Approche Symbolique.
Thèse présentée par Tony Postiau en vue de l'obtention du grade de Docteur en

Sciences Appliquées.

- [15] **REZA N.JAZAR: Theory of applied robotics Kinematics; Dynamics; and control Second edition (Springer) ISBN 978-1-4419-1749-2 e-ISBN 978-1-4419-1750-8 DOI 10.1007/978-1-4419-1750-8 Springer New York Dordrecht Heidelberg London**
- [16] **Y. BAAZI, "Etude d'un Manipulateur à 3 Degrés de Liberté Application à la Fonction de Préhenseur", thèse de magister, Institut d'électronique, Université de Batna, 2000.**
- [17] **introduction to robotics mechanics and control second edition JOHN J.Craig, John J., 1955- Introduction to robotics: mechanics and control / John J.Craig.- 2nd ed.**
- [18] **Robots parallèles à nacelle articulée Du concept à la solution industrielle pour le pick-and-place présenté par Vincent NABAT 2 mars 2007 docteur en Génie Informatique, Automatique et Traitement du Signal UNIVERSITE MONTPELLIER II**
- [19] **Suivi de trajectoires d'un robot mobile non holonome : approche par modèle flou de Takagi-Sugeno et prise en compte des retards -El-Hadi GUECHI juillet 2007 docteur en génie automatique université de UNIVERSITÉ DE VALENCIENNES ET DU HAINAUT CAMBRÉSIS**
- [20] **[Pérez, 1983] Pérez, T. (1983). Spatial planning : A configuration space approach. In IEEE transactions on computers, C-32 pages 108-120.**
- [21] **Jean- Pierre MERLET et Nicolas MOULY Rapport de recherche N 2291 février 1994 71 pages**
- [22] **C.laugier 'raisonnement géométrique et méthode de décision en robotique application a la programmation automatique des robots**
- [23] **T.lozano-Pérez An approach to automatic robot programming**
- [24] **T.lozano-Pérez J.L Jones A robot system that recognizes, plans and manipulates**
- [25] **Génération et identification des lois du mouvement'. Technical report.**
- [26] **Pérez, Special planning: A configuration space approach. In IEEE transactions on computers, C-32 pages 108-120**
- [27] **[J.T.Schwartz and M.sharir,] J.T.Schwartz and M.sharir (1983). On the piano movers' problem.**

- [28] **P.Tournassoud Géométrie et intelligences artificielle pour les robots, complexité du calcul de trajectoires, HERMES**
- [29] **[N.J.Nelsson, N.J.Nelsson. A mobile automaton : an application of artificial intelligence. Washington, DC(USA).**
- [30] **O’Dùnlain and N.Yap, O’Dùnlain and N.Yap . Journal of Algorithms**
- [31] **P.Tournassoud. Planification et controle en robotique, application aux robots Mobiles et manipulateurs. HERMES.**
- [32] **B.Faverjon and P.Tournassoud (1987). The mixed approach for path planning of manipulators. March.**
- [33] **robot modeling and control JOHN WILEY & SONS, INC. New York / Chichester / Weinheim / Brisbane / Singapore / Toronto page 97**
- [34] **Modeling, Performance Analysis and Control of Robot Manipulators Edited by Etienne Dombre Wisama Khalil © ISTE Ltd, 2007 © LAVOISIER, 2001, 2002**
- [35] **J.M. Hollerbach and S. Gideon. Wrist-partitioned inverse kinematic accelerations and manipulator dynamics. International Journal of Robotics Research.**
- [36] **J. Duffy. Analysis of Mechanisms and Robot Manipulators. Wiley, New York.**
- [37] **A. J. Critchlow. Introduction to Robotics. Macmillan, New York,**
- [38] **Etude d’un robot à 6ddl pour la télé-échographie : cinématique et performances Lama AL BASSIT**
- [39] **Exploitation de la redondance pour la commande coordonnée d’un manipulateur mobile d’assistance aux personnes handicapées DOCTEUR DE l’université d’EVRY Spécialité : Robotique présentée et soutenue publiquement par Khiair NAIT CHABANE le 30 novembre 2006**
- [40] **The power manipulability – A new homogeneous performance index of robot manipulators Imed MANSOURI, Mohammed OUALI Department of Mechanical Engineering, University of Science and Technology of Oran (USTO), Algeria Structural Mechanics Research Laboratory, Department of mechanical Engineering, University Saad Dahleb, Blida, Algeria Received 16 April 2009; revised 26 August 2010; Accepted 15 September 2010. Available online 14 October 2010**

- [41] **Bruno Siciliano • Lorenzo Sciavicco Luigi Villani • Giuseppe Oriolo**
Robotics Modelling, Planning and Control 123 ISBN 978-1-84628-641-4 e-ISBN
-1-84628-642-1 DOI 10.1007/978-1-84628-642-1
- [42] **Trajectory tracking for SCARA robots with compliant transmissions:**
A technique to improve the positioning precision G. Inceri_University Brescia -
Mechanical and Industrial Engineering Dept.Brescia, Italy.
- [43] **Position Control of 2-Link SCARA Robot by using Internal**
Model Control , Shinya AKAMATSU Okayama University
3-1-1, Tsushima-Naka Okayama, 700-8530 Okayama University
3-1-1, Tsushima-Naka Okayama, 700-8530 (Received November 22, 2008)
- [44] **cours de dynamique des structures Mr AIAd université de Blida.**
- [45] **Reza N Jazar theory of applied robotics (Kinematics,Dynamics,and contron)**
second edition springer (patrh planning) ISBN 978-1-4419-1749-2 e-ISBN 978-1-
4419-1750-8 DOI 10.1007/978-1-4419-1750-8
Springer New York Dordrecht Heidelberg London Springer Science+Business
Media, LLC 2006, 2010

APPENDICE

Listing des programmes sous MATLAB

1- programme pour calculer la matrice de transformation homogène et la matrice jacobienne et l'inverse de la matrice jacobienne :

.....la matrice de transformation homgene (homogenous transformation).....
.....fkine (forwardkinematic).....

```
clear all,clc
```

```
syms l1
```

```
syms l2
```

```
syms d3
```

```
syms d4
```

```
syms tita1
```

```
syms tita2
```

```
syms tita4
```

```
A1=[cos(tita1) -sin(tita1) 0 l1*cos(tita1)
     sin(tita1) cos(tita1) 0 l1*sin(tita1)
     0 0 1 0
     0 0 0 1]
```

```
A2=[cos(tita2) -sin(tita2) 0 l2*cos(tita2)
     sin(tita2) cos(tita2) 0 l2*sin(tita2)
     0 0 1 0
     0 0 0 1]
```

```
A3=[1 0 0 0
     0 1 0 0
     0 0 1 -d3
     0 0 0 1]
```

```
A4=[cos(tita4) -sin(tita4) 0 0
     sin(tita4) cos(tita4) 0 0
     0 0 1 -d4
     0 0 0 1]
```

```
fkine=A1*A2*A3*A4
```

```
simplify(fkine)
```

```
fkine=
```

```
[ cos(tita1 + tita2 + tita4), -sin(tita1 + tita2 + tita4), 0, l2*cos(tita1 + tita2) + l1*cos(tita1)]
```

```
[ sin(tita1 + tita2 + tita4), cos(tita1 + tita2 + tita4), 0, l2*sin(tita1 + tita2) + l1*sin(tita1)]
```

```
[ 0, 0, 1, -d3 - d4]
[ 0, 0, 0, 1]
```

.....calcul A4.....

```
syms nx
```

```

syms ox
syms ax
syms px
syms ny
syms oy
syms ay
syms py
syms nz
syms oz
syms az
syms pz

```

```
THR=[nx ox ax px;ny oy ay py;nz oz az pz;0 0 0 1]
```

```
THR =
```

```

[ nx, ox, ax, px]
[ ny, oy, ay, py]
[ nz, oz, az, pz]
[ 0, 0, 0, 1]

```

```
A4=inv(A3)*inv(A2)*inv(A1)*inv(THR)
```

```
A4 =
```

```
simplify(A4)
```

```
A4 =
```

```

[ nx*cos(tital + tita2) + ny*sin(tital + tita2), ox*cos(tital + tita2) +
oy*sin(tital + tita2), ax*cos(tital + tita2) + ay*sin(tital + tita2),
px*cos(tital + tita2) - l2 + py*sin(tital + tita2) - l1*cos(tita2)]
[ ny*cos(tital + tita2) - nx*sin(tital + tita2), oy*cos(tital + tita2) -
ox*sin(tital + tita2), ay*cos(tital + tita2) - ax*sin(tital + tita2),
py*cos(tital + tita2) - px*sin(tital + tita2) - l1*sin(tita2)]
[          nz,          oz,          az,          d3 + pz]
[          0,          0,          0,          1]

```

```

...programme pour calculer le jacobien et l'inverse du jacobien.....
.....inverse kinematic(cinematique inverse).....

```

```
syms tital
```

```
syms tita2
```

```
c12=cos(tital+tita2);
```

```
s12=sin(tital+tita2);
```

```
c1=cos(tital);
```

```
c2=cos(tita2);
```

```
s1=sin(tital);
```

```
s2=sin(tita2);
```

```
jacob0=le jacobien
```

```
va=vitesse articulaire
```

```
vo=vitesse operationelle
```

```
syms l1
```

```
syms l2
```

```

jacob0=[-l1*s1-l2*s12      -l2*s12  0  0
         l1*c1+l2*c12     l2*c12  0  0
                0          0  1  0
                1          1  0  1]

```

```
jacob0 =
```

```

[ - 12*sin(tital1 + tita2) - 11*sin(tital1), -12*sin(tital1 + tita2), 0, 0]
[ 12*cos(tital1 + tita2) + 11*cos(tital1), 12*cos(tital1 + tita2), 0, 0]
[ 0, 0, 1, 0]
[ 0, 0, 0, 0]
[ 0, 0, 0, 0]
[ 1, 1, 0, 1]

```

.....calcul les vitesses dans l'espace operationel.....

```

dt1=diff(tital1)
dt2=diff(tita2)
dt4=diff(tita4)
dd3=diff(d3)
syms dx
syms dy
syms dz
syms V
syms dt1
syms dt2
syms dt4
syms dd3
v0=[dx dy dz V]
va=[dt1 dt2 dt4 dd3]
v0'=jacob0*va'
v0 =

- (dt1)*(12*sin(tital1 + tita2) + 11*sin(tital1)) - 12*sin(tital1 +
tita2)*(dt2)
(dt1)*(12*cos(tital1 + tita2) + 11*cos(tital1)) + 12*cos(tital1 + tita2)*(dt2)
(dt4)
(dd3) +(dt1) +(dt2)

```

.....inverse kinematic(le cinematique inverse).....

```

inv(jacob0)

ans =

[cos(tital1 + tita2)/(11*sin(tita2)),sin(tital1 + tita2)/(11*sin(tita2)), 0,
0]
[-(12*cos(tital1 + tita2) + 11*cos(tital1))/(11*12*sin(tita2)), -
(12*sin(tital1 + tita2) + 11*sin(tital1))/(11*12*sin(tita2)), 0, 0]
[ 0, 0, 1, 0]
[cos(tital1)/(12*sin(tita2)), sin(tital1)/(12*sin(tita2)), 0, 1]

ikine=inv(jacob0)
ikine =

[cos(tital1 + tita2)/(11*sin(tita2)),sin(tital1 + tita2)/(11*sin(tita2)), 0,
0]
[-(12*cos(tital1 + tita2) + 11*cos(tital1))/(11*12*sin(tita2)), -
(12*sin(tital1 + tita2) + 11*sin(tital1))/(11*12*sin(tita2)), 0, 0]
[0, 0, 1, 0]
[cos(tital1)/(12*sin(tita2)), sin(tital1)/(12*sin(tita2)), 0, 1]

```

2- programme pour calculer la manipulabilité et la singularité et la trajectoire

.....programme pour calculer la manipulabilité du robot SCARA.....

```
clear all;clc
a=1;
%v=[xa ya xb yb]
v(:,4)=0;
for i=2:6
    v(i,1)=a*cos((i-1)*pi/12);
    v(i,2)=a*sin((i-1)*pi/12);
    v(i,3)=2*v(i,1);
end
v(7,:)= [0 a 0 0];
v(1,:)= [a 0 2*a 0];
x(:,2)=v(:,1)
x(:,3)=v(:,3)
y(:,2)=v(:,2)
y(:,3)=v(:,4)
for i=1:7
    plot(x(i,:),y(i,:))
    hold on
end
```

.....mesure de la manipulabilité(w).....

```
w=sqrt(det(jacob0*jacob0'))
w =
(cos((pi*t^4)/3 + (pi*t^3)/9 + (pi*t^2)/18)*sin((pi*t^4)/18 + (pi*t^3)/18 +
(pi*t^2)/18)*cos((pi*conj(t)^4)/3 + (pi*conj(t)^3)/9 +
(pi*conj(t)^2)/18)*sin((pi*conj(t)^4)/18 + (pi*conj(t)^3)/18 +
(pi*conj(t)^2)/18) - cos((pi*t^4)/3 + (pi*t^3)/9 +
(pi*t^2)/18)*sin((pi*t^4)/18 + (pi*t^3)/18 +
(pi*t^2)/18)*cos((pi*conj(t)^4)/18 + (pi*conj(t)^3)/18 +
(pi*conj(t)^2)/18)*sin((pi*conj(t)^4)/3 + (pi*conj(t)^3)/9 +
(pi*conj(t)^2)/18) - cos((pi*t^4)/18 + (pi*t^3)/18 +
(pi*t^2)/18)*sin((pi*t^4)/3 + (pi*t^3)/9 +
(pi*t^2)/18)*cos((pi*conj(t)^4)/3 + (pi*conj(t)^3)/9 +
(pi*conj(t)^2)/18)*sin((pi*conj(t)^4)/18 + (pi*conj(t)^3)/18 +
(pi*conj(t)^2)/18) + cos((pi*t^4)/18 + (pi*t^3)/18 +
(pi*t^2)/18)*sin((pi*t^4)/3 + (pi*t^3)/9 +
(pi*t^2)/18)*cos((pi*conj(t)^4)/18 + (pi*conj(t)^3)/18 +
(pi*conj(t)^2)/18)*sin((pi*conj(t)^4)/3 + (pi*conj(t)^3)/9 +
(pi*conj(t)^2)/18))^(1/2)
```

.....programme pour calculer la singularité du robot SCARA en

```
fct(tita2) pour différente valeur de tital
tital=[0;pi/8;2*pi/8;3*pi/8;4*pi/8;5*pi/8;6*pi/8];
tita2=0:0.01:15;
px=1;
c2=(px^2+1.5^2-2)/2;
s2=sqrt(1-c2^2);
s1=((1+c2)*1.5-s2*px)/(px^2+1.5^2);
c1=((1+c2)*px-s2*1.5)/(px^2+1.5^2);
for i=1:5
    s12=sin(tital(i)+tita2);
    c12=cos(tital(i)+tita2);
    detj(:,i)=- (s1+s12).*c12-s12.*(c1+c12)
    plot(tita2,detj)
    hold on
end
```

.....programme pour calculer la trajectoir...

```
t=0:0.001:1;
tita1=(10*t.^4+10*t.^3+10*t.^2)*pi/180;
dtita1=(40*t.^3+30*t.^2+20*t)*pi/180;
tita2=(50*t.^4+10*t.^3)*pi/180;
dtita2=(200*t.^3+30*t.^2)*pi/180;
tita3=(60*t.^4+20*t.^3+10*t.^2)*pi/180;
dtita3=(240*t.^3+60*t.^2+20*t)*pi/180;
tita4=(-50*t.^4-10*t.^3)*pi/180;
dtita4=(-200*t.^3-30*t.^2)*pi/180;
dd3=0.015*t.^2;
l1=1;
l2=1;
c1=cos(tita1);
s1=sin(tita1);
c2=cos(tita2);
s2=sin(tita2);
c12=cos(tita1+tita2);
s12=sin(tita1+tita2);
c3=cos(tita3);
s3=sin(tita3);
c4=cos(tita4);
s4=sin(tita4);
c34=cos(tita3+tita4);
s34=sin(tita3+tita4);
x1=l1*c1+l2*c12;
y1=l1*s1+l2*s12;
x2=l1*c3+l2*c34;
y2=l1*s3+l2*s34;
plot(x1,y1,x2,y2)
```

3- Le programme sous MATLAB pour calculer l'énergie cinétique et potentielle et le couple appliqué:

...calcul le couple et l'énergie cinétique et potentielle totale de posture(1)...

```
t=0:0.001:1;
t1=(10*t.^4+10*t.^3+10*t.^2)*pi/180;
t2=(50*t.^4+10*t.^3)*pi/180;
dt1=(40*t.^3+30*t.^2+20*t)*pi/180;
dt2=(200*t.^3+30*t.^2)*pi/180;
ddt1=(120*t.^2+60*t+20)*pi/180;
ddt2=(600*t.^2+60*t)*pi/180;
c1=cos(t1);
s1=sin(t1);
c12=cos(t1+t2);
s12=sin(t1+t2);
c2=cos(t2);
s2=sin(t2);
```

```

m1=16.92;
m2=16.92;
g=9.81;
l1=1;
l2=1;
.....énergie cinétique .....
K1=0.5*m1*l1.^2*dt1.^2;
K2=0.5*m2*l1.^2*dt1.^2+0.5*m2*l2.^2*(dt1+dt2).^2+m2*l1*l2*(dt1.^2+dt1.*dt2)
.*c2;
K=K1+K2;
figure(1)
plot(t,K)
.....énergie potentielle.....
V1=m1*g*l1*s1;
V2=m2*g*(l1*s1+l2*s12)
V=V1+V2;
figure(2)
plot(t,V)
.....couple de l'articulaion(1).....
t01=m2*l2.^2*(ddt1+ddt2)+m2*l1*l2*c2.*(2*ddt1+ddt2)+(m1+m2)*l1.^2*ddt1-
m2*l1*l2*s2.*dt2.^2-2*m2*l1*l2*s2.*dt1.*dt2+m2*l2*g*c12+(m1+m2)*g*c1;
figure(3);
plot(t,t01)
.....couple de l'articulation(2).....
t02=m2*l1*l2*c2.*ddt1+m2*l1*l2*s2.*dt1.^2+m2*l2*g*c12+m2*l2.^2*(ddt1+ddt2);
figure(4)
plot(t,t02);
.calcul le couple et l'energie cinetque et potentielle totale de posture(2)
t=0:0.001:1;
t1=(60*t.^4+20*t.^3+10*t.^2)*pi/180;
t2=(-50*t.^4-10*t.^3)*pi/180;
dt1=(240*t.^3+60*t.^2+20*t)*pi/180;
dt2=(-200*t.^3-30*t.^2)*pi/180;
ddt1=(720*t.^2+120*t+20)*pi/180;
ddt2=(-600*t.^2-60*t)*pi/180;
c1=cos(t1);
s1=sin(t1);
c12=cos(t1+t2);
s12=sin(t1+t2);

```

```

c2=cos(t2);
s2=sin(t2);
m1=16.92;
m2=16.92;
g=9.81;
l1=1;
l2=1;

.....energie cinetique .....
K1=0.5*m1*l1.^2*dt1.^2;
K2=0.5*m2*l1.^2*dt1.^2+0.5*m2*l2.^2*(dt1+dt2).^2+m2*l1*l2*(dt1.^2+dt1.*dt2)
.*c2;
K=K1+K2;
figure(1)
plot(t,K)

.....energie potentiell.....
V1=m1*g*l1*s1;
V2=m2*g*(l1*s1+l2*s12)
V=V1+V2;
figure(2)
plot(t,P)

.....couple de l'articulaion(1)..
t01=m2*l2.^2*(ddt1+ddt2)+m2*l1*l2*c2.*(2*ddt1+ddt2)+(m1+m2)*l1.^2*ddt1-
m2*l1*l2*s2.*dt2.^2-2*m2*l1*l2*s2.*dt1.*dt2+m2*l2*g*c12+(m1+m2)*l1*g*c1;
figure(3);
plot(t,t01)

.....couple de l'articulation(2).....
t02=m2*l1*l2*c2.*ddt1+m2*l1*l2*s2.*dt1.^2+m2*l2*g*c12+m2*l2.^2*(ddt1+ddt2);
figure(4)
plot(t,t02);

```

4- le code de calcul pour le robot SCARA pour les deux postures sans orientation de

l'organe terminal :

les mots clés:

```

R:articulation rotoide
P:articulation prismatique
fkine:forward kinematic
jacob0 :le jacobian du robot
ikine:inverse kinematic
A:link length
alpha:link twist
D:link offset
theta:joint angle
grav : [0.00 0.00 9.81]

```

```

t1:tita1
t2:tita2
dt1:diff(t,t1)
dt2:diff(t,t2)
l(i):link offset
drivebot:dessiner le robot a la position mentionée
tita4:orientation de lorgane terminal
dd3:vitesse de lorgane terminal
w : la manipulabilité

```

```

.....posture(1).....
L1=link([0 1 0 0 0], 'standard')
L2=link([0 1 0 0 0], 'standard')
L3=link([0 0 0 0.63 1], 'standard')
r=robot({L1 L2 L3})
r.name='scara posture(1)'
syms t
t1=(10*t.^4+10*t.^3+10*t.^2)*pi/180;
t2=(50*t.^4+10*t.^3)*pi/180;
tita4=0;
dt1=(40*t.^3+30*t.^2+20*t)*pi/180;
dt2=(200*t.^3+30*t.^2)*pi/180;
ddt1=(120*t.^2+60*t+20)*pi/180;
ddt2=(600*t.^2+60*t)*pi/180;
dt4=0
dd3=0.015*t.^2;
l1=1;
l2=1;
c1=cos(t1);
s1=sin(t1);
c2=cos(t2);
s2=sin(t2);
c12=cos(t1+t2);
s12=sin(t2+t1);
q=[t1 t2 -0.63]
T=fkine(r, q)
trplot(T)
drivebot(r, q)
jacob0=[-l1*s1-l2*s12 -l2*s12 0 0; l2*c1+l2*c12 l2*c12 0 0; 0 0 1 0; 1 1 0 1]
w=sqrt(det(jacob0*jacob0'))
va=[dt1 dt2 dt4 dd3]
vo=jacob0*va'
inv(jacob0);
ikine=inv(jacob0)
va=ikine*vo
m1=16.92;
m2=16.92;
g=9.81;
K1=0.5*m1*l1.^2*dt1.^2
K2=0.5*m2*l1.^2*dt1.^2+0.5*m2*l2.^2*(dt1+dt2).^2+m2*l1*l2*(dt1.^2+dt1.*dt2)
.*c2
K=K1+K2
V1=m1*g*l1*s1
V2=m2*g*(l1*s1+l2*s12)
v=V1+V2
t01=m2*l2.^2*(ddt1+ddt2)+m2*l1*l2*c2.*(2*ddt1+ddt2)+(m1+m2)*l1.^2*ddt1-
m2*l1*l2*s2.*dt2.^2-2*m2*l1*l2*s2.*dt1.*dt2+m2*l2*g*c12+(m1+m2)*l1*g*c1
t02=m2*l1*l2*c2.*ddt1+m2*l1*l2*s2.*dt1.^2+m2*l2*g*c12+m2*l2.^2*(ddt1+ddt2)

```

```

.....posture (2) .....
L1=link([0 1 0 0 0], 'standard')
L2=link([0 1 0 0 0], 'standard')
L3=link([0 0 0 0.63 1], 'standard')
r=robot({L1 L2 L3})
r.name='scara posture (2) '
syms t
t1=(60*t.^4+20*t.^3+10*t.^2)*pi/180;
t2=(-50*t.^4-10*t.^3)*pi/180;
dt1=(240*t.^3+60*t.^2+20*t)*pi/180;
dt2=(-200*t.^3-30*t.^2)*pi/180;
ddt1=(720*t.^2+120*t+20)*pi/180;
ddt2=(-600*t.^2-60*t)*pi/180;
dt4=0
dd3=0.015*t.^2;
l1=1;
l2=1;
c1=cos(t1);
s1=sin(t1);
c2=cos(t2);
s2=sin(t2);
c12=cos(t1+t2);
s12=sin(t2+t1);
q=[t1 t2 -0.63]
T=fkine(r, q)
trplot(T)
drivebot(r, q)
jacob0=[-l1*s1-l2*s12 -l2*s12 0 0; l2*c1+l2*c12 l2*c12 0 0; 0 0 1 0; 1 1 0 1]
w=sqrt(det(jacob0*jacob0'))
va=[dt1 dt2 dt4 dd3]
vo=jacob0*va'
inv(jacob0);
ikine=inv(jacob0)
va=ikine*vo
m1=16.92;
m2=16.92;
g=9.81;
K1=0.5*m1*l1.^2*dt1.^2
K2=0.5*m2*l1.^2*dt1.^2+0.5*m2*l2.^2*(dt1+dt2).^2+m2*l1*l2*(dt1.^2+dt1.*dt2)
.*c2
K=K1+K2
V1=m1*g*l1*s1
V2=m2*g*(l1*s1+l2*s12)
V=V1+V2
t01=m2*l2.^2*(ddt1+ddt2)+m2*l1*l2*c2.*(2*ddt1+ddt2)+(m1+m2)*l1.^2*ddt1-
m2*l1*l2*s2.*dt2.^2-2*m2*l1*l2*s2.*dt1.*dt2+m2*l2*g*c12+(m1+m2)*l1*g*c1
t02=m2*l1*l2*c2.*ddt1+m2*l1*l2*s2.*dt1.^2+m2*l2*g*c12+m2*l2.^2*(ddt1+ddt2)

```

exemple d'application :

Posture (1) :

L1 =
0.000000 1.000000 0.000000 0.000000 R (std)

L2 =

```

0.000000  1.000000  0.000000  0.000000  R  (std)

L3 =
0.000000  0.000000  0.000000  0.630000  P  (std)

r =
noname (3 axis, RRP)
      grav = [0.00 0.00 9.81]          standard D&H parameters

      alpha      A      theta      D      R/P
0.000000  1.000000  0.000000  0.000000  R  (std)
0.000000  1.000000  0.000000  0.000000  R  (std)
0.000000  0.000000  0.000000  0.630000  P  (std)

r =
scara posture(1) (3 axis, RRP)
      grav = [0.00 0.00 9.81]          standard D&H parameters

      alpha      A      theta      D      R/P
0.000000  1.000000  0.000000  0.000000  R  (std)
0.000000  1.000000  0.000000  0.000000  R  (std)
0.000000  0.000000  0.000000  0.630000  P  (std)

t =
  1

dt4 =
  0

q =
0.5236  1.0472  -0.6300

T =

0.0000  -1.0000  0  0.8660
1.0000  0.0000  0  1.5000
0  0  1.0000  -0.6300
0  0  0  1.0000

ans =
,

No tag
set tag

jacob0 =
-1.5000  -1.0000  0  0
0.8660  0.0000  0  0
0  0  1.0000  0
1.0000  1.0000  0  1.0000

w =
0.8660

va =
1.5708  4.0143  0  0.0150
vo =
-6.3705

```

```

1.3603
0
5.6001
ikine =
0.0000 1.1547 0 0
-1.0000 -1.7321 0 0
0 0 1.0000 0
1.0000 0.5774 0 1.0000
va =
1.5708
4.0143
0
0.0150
K1 =
20.8742
K2 =
358.9849
K =
379.8591
V1 =
82.9926
V2 =
248.9778
V =
331.9704
tol =
395.1813
t02 =
319.6525

```

Posture (2):

```

L1 =
0.000000 1.000000 0.000000 0.000000 R (std)

L2 =
0.000000 1.000000 0.000000 0.000000 R (std)

L3 =
0.000000 0.000000 0.000000 0.630000 P (std)

r =

```

```

noname (3 axis, RRP)
grav = [0.00 0.00 9.81] standard D&H parameters

```

alpha	A	theta	D	R/P
0.000000	1.000000	0.000000	0.000000	R (std)
0.000000	1.000000	0.000000	0.000000	R (std)
0.000000	0.000000	0.000000	0.630000	P (std)

```

r =
scara posture(2) (3 axis, RRP)
grav = [0.00 0.00 9.81] standard D&H parameters

```

alpha	A	theta	D	R/P
-------	---	-------	---	-----

```
0.000000 1.000000 0.000000 0.000000 R (std)
0.000000 1.000000 0.000000 0.000000 R (std)
0.000000 0.000000 0.000000 0.630000 P (std)
```

```
t =
```

```
1
```

```
dt4 =
```

```
0
```

```
q =
```

```
1.5708 -1.0472 -0.6300
```

```
T =
```

```
0.8660 -0.5000 0 0.8660
```

```
0.5000 0.8660 0 1.5000
```

```
0 0 1.0000 -0.6300
```

```
0 0 0 1.0000
```

```
ans =
```

```
trplot
```

```
jacob0 =
```

```
-1.5000 -0.5000 0 0
```

```
0.8660 0.8660 0 0
```

```
0 0 1.0000 0
```

```
1.0000 1.0000 0 1.0000
```

```
w =
```

```
0.8660
```

```
va =
```

```
5.5851 -4.0143 0 0.0150
```

```
vo =
```

```
-6.3705
```

```
1.3603
```

```
0
```

```
1.5858
```

```
ikine =
```

```
-1.0000 -0.5774 0 0
```

```
1.0000 1.7321 0 0.0000
```

```
0 0 1.0000 0
```

```
-0.0000 -1.1547 0 1.0000
```

```
va =
```

```
5.5851
```

```
-4.0143
```

```
0
```

```
0.0150
```

```
K1 =
```

```
263.8913
```

```
K2 =
```

```
358.9849
```

```
K =
```

```
622.876
```

```
V1 =
```

```
165.9852
```

```
V2 =
```

```
248.9778
```

```
v =
```

```

414.9630
t01 =
446.3383
t02 =
-127.2806

```

5- le code de calcul pour le robot SCARA pour les deux postures avec orientation de l'organe terminal :

```

.....posture(1).....
L1=link([0 1 0 0 0], 'standard')
L2=link([0 1 0 0 0], 'standard')
L3=link([0 0 0 0 0], 'standard')
L4=link([0 0 0 -0.63 1], 'standard')
r=robot({L1 L2 L3 L4})
r.name='scara posture(1)'
syms t
t1=(10*t.^4+10*t.^3+10*t.^2)*pi/180;
t2=(50*t.^4+10*t.^3)*pi/180;
t4=pi/3*t.^2;.....exemple
dt1=(40*t.^3+30*t.^2+20*t)*pi/180;
dt2=(200*t.^3+30*t.^2)*pi/180;
dt4=2*pi/3*t;.....exemple
ddt1=(120*t.^2+60*t+20)*pi/180;
ddt2=(600*t.^2+60*t)*pi/180;
dd3=0.015*t.^2;
l1=1;
l2=1;
c1=cos(t1);
s1=sin(t1);
c2=cos(t2);
s2=sin(t2);
c12=cos(t1+t2);
s12=sin(t2+t1);
q=[t1 t2 t4 -0.63]
T=fkine(r, q)
trplot(T)
drivebot(r, q)
jacob0=[-l1*s1-l2*s12 -l2*s12 0 0; l2*c1+l2*c12 l2*c12 0 0; 0 0 1 0; 1 1 0 1]
w=abs(det(jacob0))
va=[dt1 dt2 dt4 dd3]
vo=jacob0*va'
inv(jacob0);
ikine=inv(jacob0)
va=ikine*vo
m1=16.92;
m2=16.92;
g=9.81;
K1=0.5*m1*l1.^2*dt1.^2
K2=0.5*m2*l1.^2*dt1.^2+0.5*m2*l2.^2*(dt1+dt2).^2+m2*l1*l2*(dt1.^2+dt1.*dt2)
.*c2
K=K1+K2
V1=m1*g*l1*s1
V2=m2*g*(l1*s1+l2*s12)
V=V1+V2
t01=m2*l2.^2*(ddt1+ddt2)+m2*l1*l2*c2.*(2*ddt1+ddt2)+(m1+m2)*l1.^2*ddt1-
m2*l1*l2*s2.*dt2.^2-2*m2*l1*l2*s2.*dt1.*dt2+m2*l2*g*c12+(m1+m2)*g*c1

```

$t02=m2*l1*l2*c2.*ddt1+m2*l1*l2*s2.*dt1.^2+m2*l2*g*c12+m2*l2.^2*(ddt1+ddt2)$

.....posture (2)

```
L1=link([0 1 0 0 0], 'standard')
L2=link([0 1 0 0 0], 'standard')
L3=link([0 0 0 0 0], 'standard')
L4=link([0 0 0 -0.63 1], 'standard')
r=robot({L1 L2 L3 L4})
r.name='scara posture(2) '
syms t
t1=(60*t.^4+20*t.^3+10*t.^2)*pi/180;
t2=(-50*t.^4-10*t.^3)*pi/180;
t4=pi/3*t.^2;.....exemple
dt1=(240*t.^3+60*t.^2+20*t)*pi/180;
dt2=(-200*t.^3-30*t.^2)*pi/180;
ddt1=(720*t.^2+120*t+20)*pi/180;
ddt2=(-600*t.^2-60*t)*pi/180;
dt4=2*pi/3*t;.....exemple
dd3=0.015*t.^2;
l1=1;
l2=1;
c1=cos(t1);
s1=sin(t1);
c2=cos(t2);
s2=sin(t2);
c12=cos(t1+t2);
s12=sin(t2+t1);
q=[t1 t2 t4 -0.63]
T=fkine(r, q)
trplot(T)
drivebot(r, q)
jacob0=[-l1*s1-l2*s12 -l2*s12 0 0; l2*c1+l2*c12 l2*c12 0 0; 0 0 1 0; 1 1 0 1]
w=abs(det(jacob0))
va=[dt1 dt2 dt4 dd3]
vo=jacob0*va'
inv(jacob0);
ikine=inv(jacob0)
va=ikine*vo
m1=16.92;
m2=16.92;
g=9.81;
K1=0.5*m1*l1.^2*dt1.^2
K2=0.5*m2*l1.^2*dt1.^2+0.5*m2*l2.^2*(dt1+dt2).^2+m2*l1*l2*(dt1.^2+dt1.*dt2)
.*c2
K=K1+K2
V1=m1*g*l1*s1
V2=m2*g*(l1*s1+l2*s12)
V=V1+V2
t01=m2*l2.^2*(ddt1+ddt2)+m2*l1*l2*c2.*(2*ddt1+ddt2)+(m1+m2)*l1.^2*ddt1-
m2*l1*l2*s2.*dt2.^2-2*m2*l1*l2*s2.*dt1.*dt2+m2*l2*g*c12+(m1+m2)*g*c1
t02=m2*l1*l2*c2.*ddt1+m2*l1*l2*s2.*dt1.^2+m2*l2*g*c12+m2*l2.^2*(ddt1+ddt2)
```

Exemple d'application :

Posture (1) :

```

L1 =
  0.000000  1.000000  0.000000  0.000000  R  (std)
L2 =
  0.000000  1.000000  0.000000  0.000000  R  (std)
L3 =
  0.000000  0.000000  0.000000  0.000000  R  (std)
L4 =
  0.000000  0.000000  0.000000  -0.630000  P  (std)

```

r =

noname (4 axis, RRRP)

```

          grav = [0.00 0.00 9.81]          standard D&H parameter
alpha      A      theta      D      R/P
0.000000  1.000000  0.000000  0.000000  R  (std)
0.000000  1.000000  0.000000  0.000000  R  (std)
0.000000  0.000000  0.000000  0.000000  R  (std)
0.000000  0.000000  0.000000  -0.630000  P  (std)

```

r =

scara posture(1) (4 axis, RRRP)

```

          grav = [0.00 0.00 9.81]          standard D&H parameters
alpha      A      theta      D      R/P
0.000000  1.000000  0.000000  0.000000  R  (std)
0.000000  1.000000  0.000000  0.000000  R  (std)
0.000000  0.000000  0.000000  0.000000  R  (std)
0.000000  0.000000  0.000000  -0.630000  P  (std)

```

t =

1

q =

0.5236 1.0472 1.0472 -0.6300

T =

```

-0.8660  -0.5000  0  0.8660
 0.5000  -0.8660  0  1.5000
 0  0  1.0000  -0.6300
 0  0  0  1.0000

```

ans =

''

No tag

set tag

jacob0 =

-1.5000	-1.0000	0	0
0.8660	0.0000	0	0
0	0	1.0000	0
1.0000	1.0000	0	1.0000

w =

0.8660

va =

1.5708	4.0143	2.0944	0.0150
--------	--------	--------	--------

vo =

-6.3705

1.3603

2.0944

5.6001

ikine =

0.0000	1.1547	0	0
--------	--------	---	---

-1.0000	-1.7321	0	0
---------	---------	---	---

0	0	1.0000	0
---	---	--------	---

1.0000	0.5774	0	1.0000
--------	--------	---	--------

va =

1.5708

4.0143

2.0944

0.0150

K1 =

20.8742

K2 =

358.9849

K =

379.8591

V1 =

82.9926

V2 =

248.9778

V =

331.9704

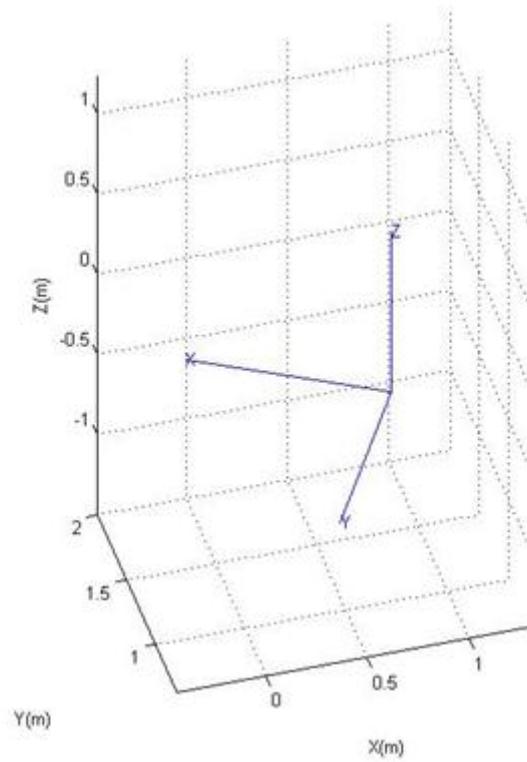
to1 =

395.1813

t02 =

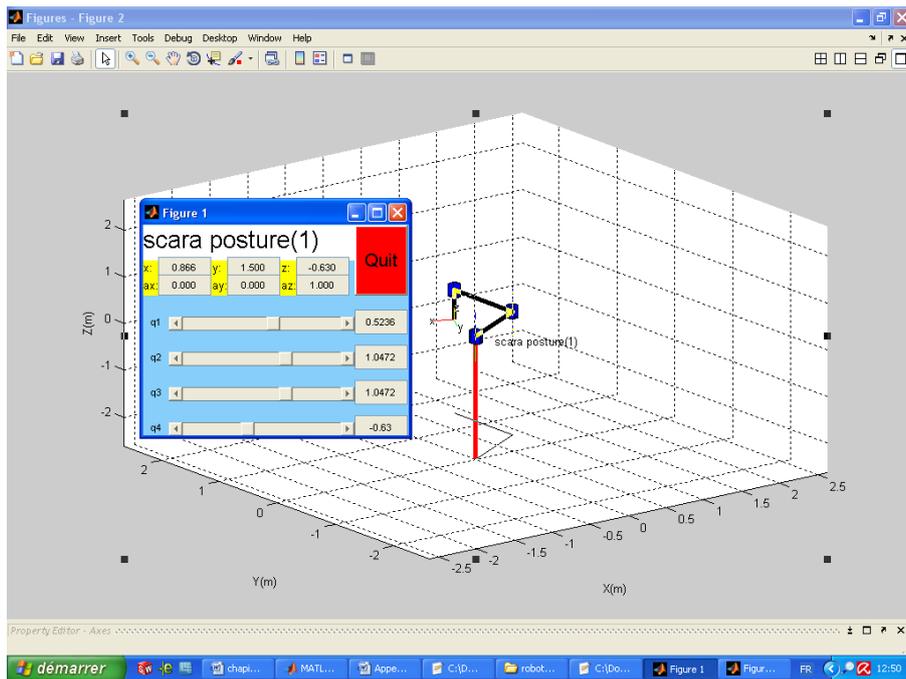
319.6525

L'orientation de la matrice de transformation homogène :

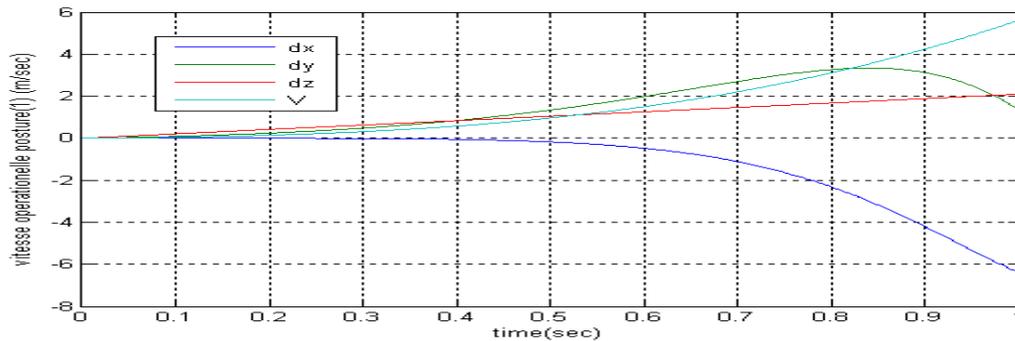


On remarque il ya une différence entre l'orientation de la matrice de transformation homogène pour les deux cas (sans orientation de l'organe terminal et avec orientation de l'organe terminal) pour un même temps voir figure (4.26)

La position du robot :



Espace operationnel :



posture(2) :

```

L1 =
    0.000000  1.000000  0.000000  0.000000  R  (std)

L2 =
    0.000000  1.000000  0.000000  0.000000  R  (std)

L3 =
    0.000000  0.000000  0.000000  0.000000  R  (std)

L4 =
    0.000000  0.000000  0.000000  -0.630000  P  (std)
r =
noname (4 axis, RRRP)
      grav = [0.00 0.00 9.81]          standard D&H parameters

      alpha      A      theta      D      R/P
0.000000  1.000000  0.000000  0.000000  R  (std)

```

```

0.000000  1.000000  0.000000  0.000000  R  (std)
0.000000  0.000000  0.000000  0.000000  R  (std)
0.000000  0.000000  0.000000  -0.630000 P  (std)

```

r =

scara posture(2) (4 axis, RRRP)

grav = [0.00 0.00 9.81]

standard D&H parameters

alpha	A	theta		D	R/P
0.000000	1.000000	0.000000	0.000000	R	(std)
0.000000	1.000000	0.000000	0.000000	R	(std)
0.000000	0.000000	0.000000	0.000000	R	(std)
0.000000	0.000000	0.000000	-0.630000	P	(std)

t =

1

q =

1.5708 -1.0472 1.0472 -0.6300

T =

0	-1.0000	0	0.8660
1.0000	0	0	1.5000
0	0	1.0000	-0.6300
0	0	0	1.0000

ans =

trplot

jacob0 =

-1.5000	-0.5000	0	0
0.8660	0.8660	0	0
0	0	1.0000	0
1.0000	1.0000	0	1.0000

w =

0.8660

va =

5.5851 -4.0143 2.0944 0.0150

vo =

-6.3705
1.3603
2.0944
1.5858

ikine =

-1.0000	-0.5774	0	0
1.0000	1.7321	0	0.0000
0	0	1.0000	0
-0.0000	-1.1547	0	1.0000

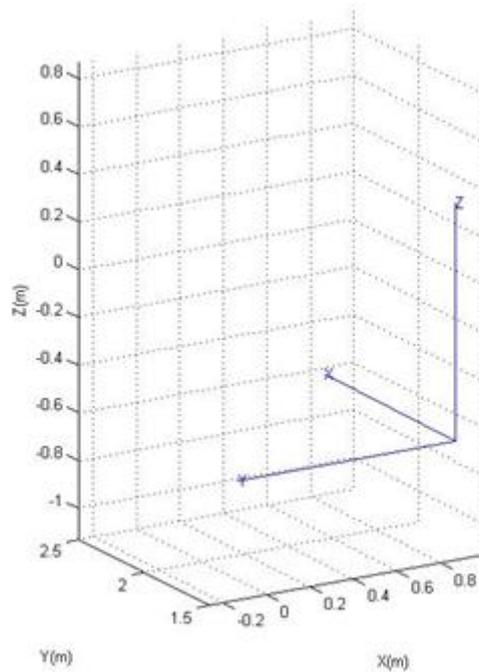
va =

```

5.5851
-4.0143
2.0944
0.0150
K1 =
263.8913
K2 =
358.984
K =
622.8762
V1 =
165.9852
V2 =
248.9778
V =
414.9630
to1 =
446.3383
t02 =
-127.2806

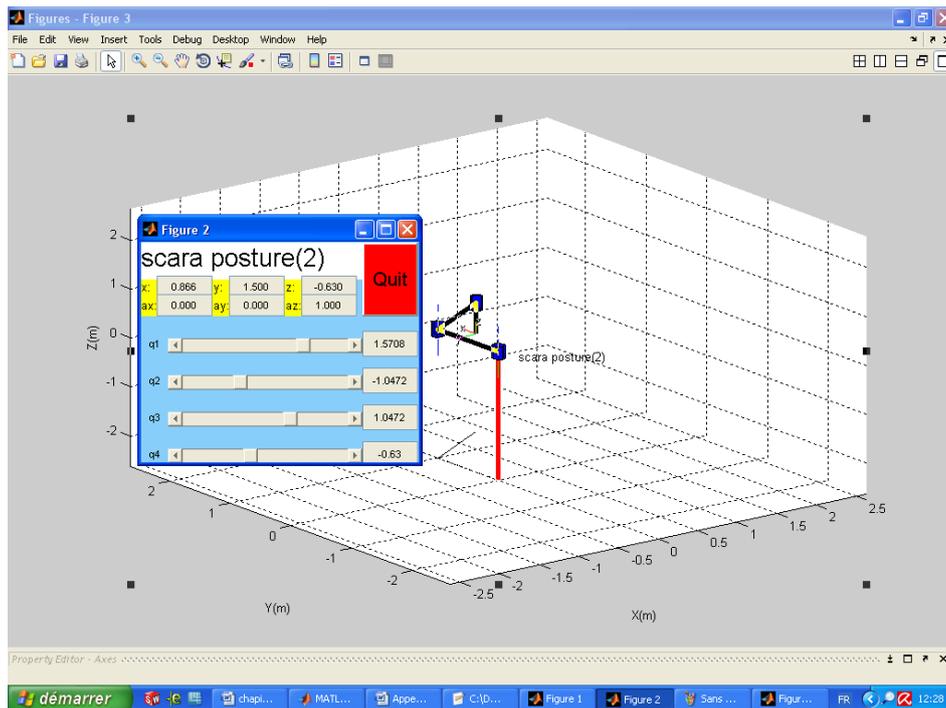
```

L'orientation de la matrice de transformation homogène :



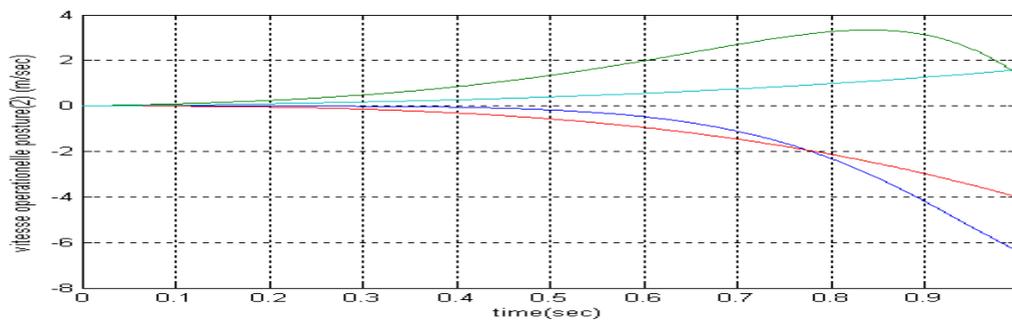
On remarque il ya une différence entre l'orientation de la matrice de transformation homogène pour les deux cas (sans orientation de l'organe terminal et avec orientation de l'organe terminal) pour un même temps voir figure (4.27)

La position du robot :



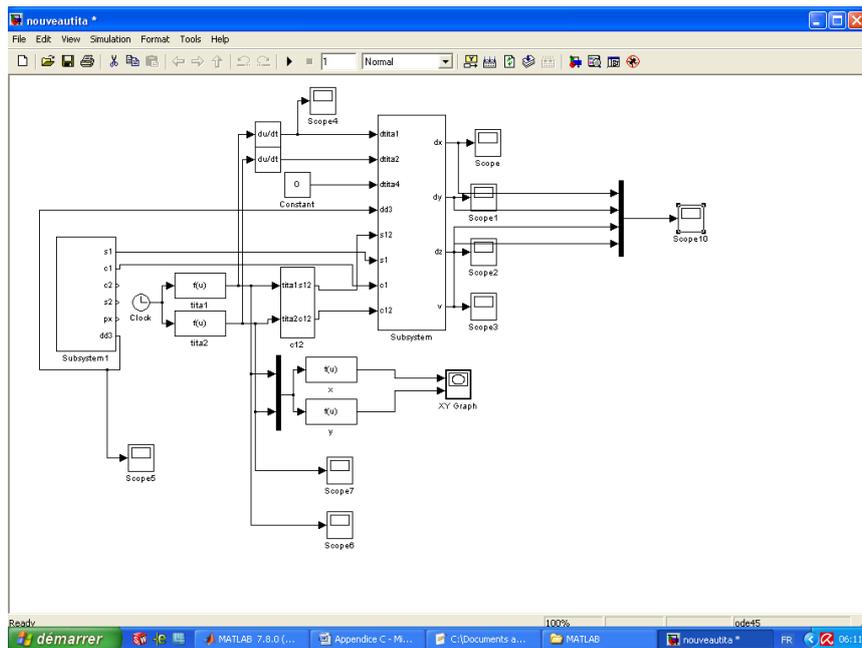
On remarque ainsi que la manipulabilité pour les robots redondants et non redondants est la même.

Espace operationnel :

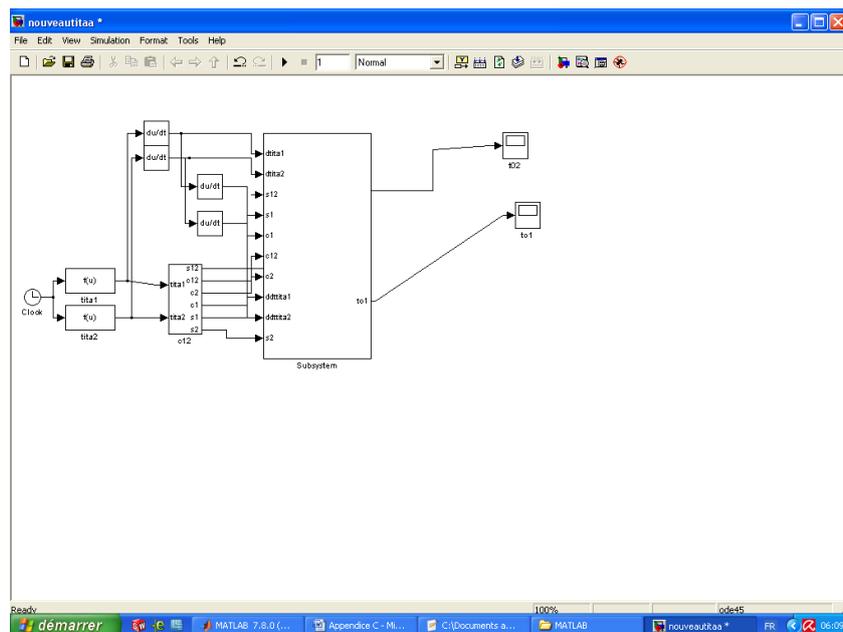


6- les blocs de simulation :

6-1 : bloc de simulation pour calculer l'espace opérationnel et la trajectoire :



6-2 : *bloc de simulation pour calculer le couple des articulations :*



7- Quelques trajectoires obtenus lors du déplacement des deux postures avec logiciel Solidworks et vérification avec Matlab pour la même position désirée :

Les équations du mouvement pour les deux postures sont données par les relations suivantes [45] :

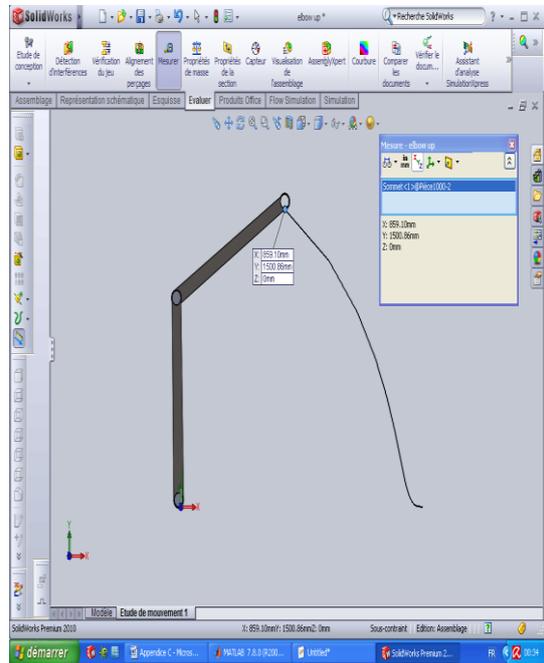
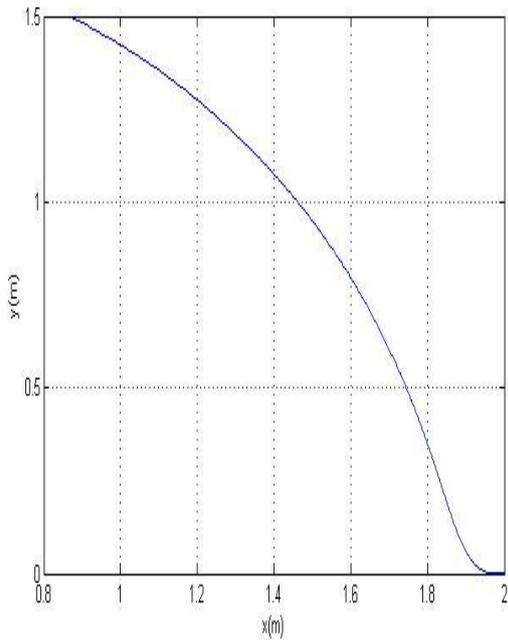
Posture(2) :

Equation du mouvement :

$$t=0:0.001:1;$$

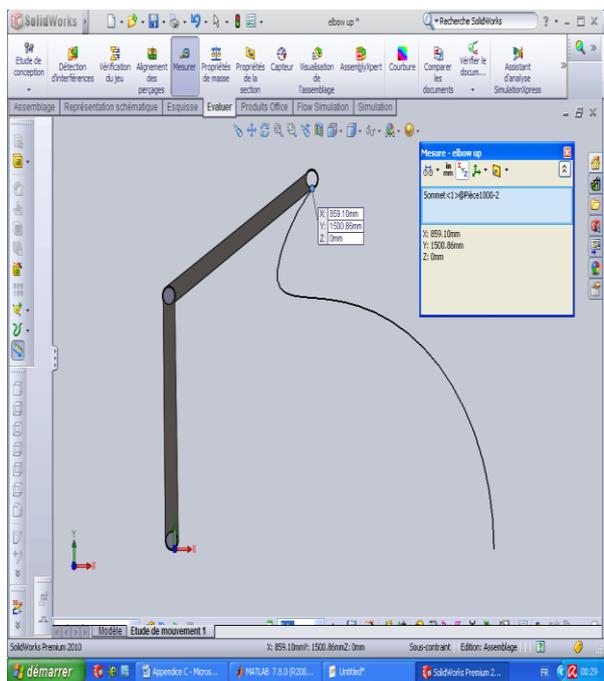
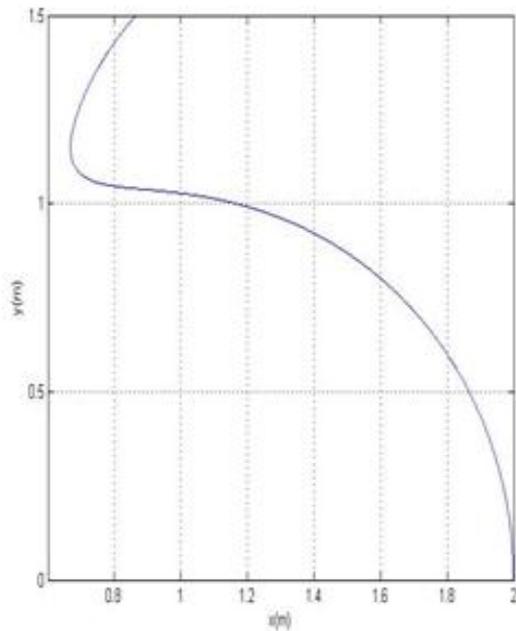
$$tita1= (20*t+10*t+60*t.^7)*pi/180;$$

```
tita2= (-20*t-40*t)*pi/180;
l1=1;
l2=1;
```



Equation du mouvement :

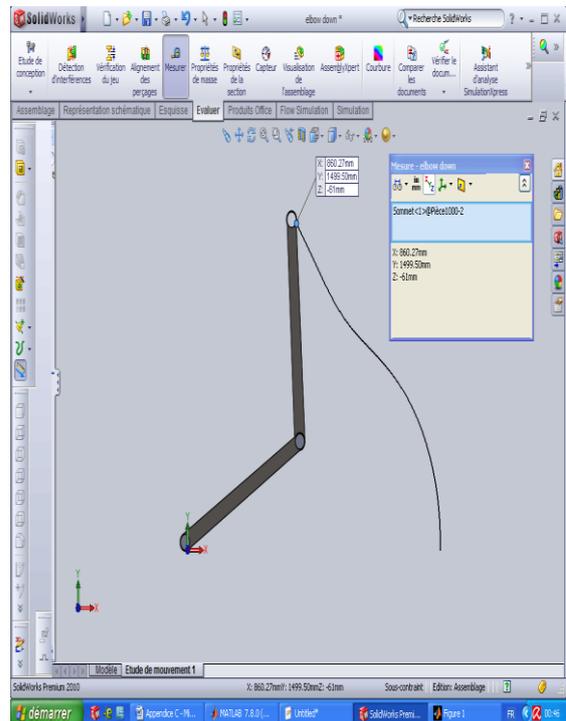
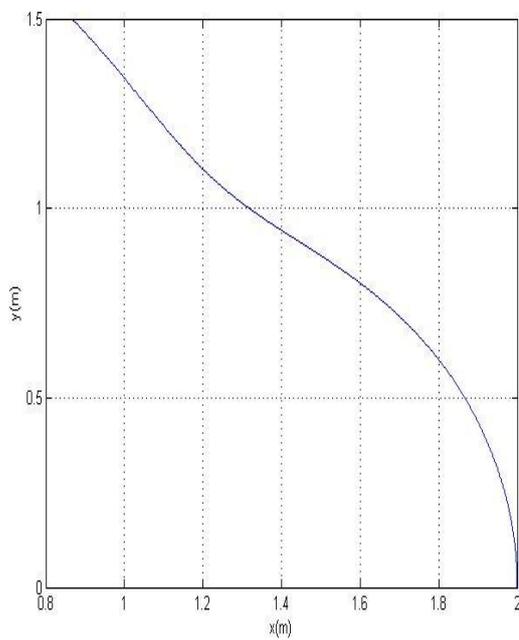
```
t=0:0.001:1;
tital= (180*t-20*t.^2-70*t.^5)*pi/180;
tita2= (-180*t+20*t.^2+100*t.^5)*pi/180;
l1=1;
l2=1;
```



Posture(1) :

Equation du mouvement :

```
t=0:0.001:1;  
tita1= (30*t.^10)*pi/180;  
tita2= (90*t-30*t.^10)*pi/180;  
l1=1;  
l2=1;
```



Equation du mouvement :

```
t=0:0.001:1;  
tita1= (90*t.^10-60*t.^2)*pi/180;  
tita2= (-90*t.^10+150*t.^2)*pi/180;  
l1=1;  
l2=1;
```

