POLYPUBLIE

Mise au point d'un banc d'essai pour l'étude théorique et expérimentale de robots flexibles

François Germain

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

Date	1996
Type	Mémoire ou thèse / Dissertation or Thesis Mémoire de maîtrise/Master's thesis
URL de PolyPublie PolyPublie URL	https://publications.polymtl.ca/9004/
Conditions d'utilisation Terms of use	© Tous droits réservés / All rights reserved
Référence Citation	Germain, F. (1996). Mise au point d'un banc d'essai pour l'étude théorique et expérimentale de robots flexibles [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/9004/

Détails de la thèse
Thesis details

Directeurs de recherche Advisors	Richard Hurteau
Programme Program	Non spécifié

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal
https://publications.polymtl.ca

UNIVERSITÉ DE MONTRÉAL

MISE AU POINT D'UN BANC D'ESSAI POUR L'ÉTUDE THÉORIQUE ET EXPÉRIMENTALE DE ROBOTS FLEXIBLES

FRANÇOIS GERMAIN
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ET DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU GRADE DE MAÎTRE ÈS SCIENCES APPLIQUÉES (M.Sc.A.)
(GÉNIE ÉLECTRIQUE)
DÉCEMBRE 1996

© François Germain, 1996.

National Library of Canada
Acquisitions and Bibliographic Services
395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada
Acquisitions et services bibliographiques
395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-26475-0

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE

Ce mémoire intitulé:

MISE AU POINT D'UN BANC D'ESSAI POUR L'ÉTUDE THÉORIQUE ET EXPÉRIMENTALE DE ROBOTS FLEXIBLES

Présenté par : GERMAIN François

en vue de l'obtention du grade de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. DE SANTIS, Romano, Ph.D., président

M. HURTEAU, Richard, D.Ing., membre et directeur de recherche

M. PIEDBOEUF, Jean-Claude, Ph.D., membre et co-directeur de recherche

M. PELLETIER, Michel, Ph.D., membre

DÉDICACE

Je dédie ce mémoire de maîtrise à ma mère Marie-Reine Cyr et à mon père René Germain.

François

REMERCIEMENTS

J'aimerais exprimer ma reconnaissance à tous ceux qui ont contribué à la réalisation de ce travail.

Je remercie plus particulièrement mon directeur de recherche monsieur Richard Hurteau, de même que mon codirecteur monsieur Jean-Claude Piedboeuf, pour leur aide financière, leurs conseils et l'intérêt porté tout au long de la réalisation de ce mémoire.

Je tiens à remercier monsieur Laurent Juneau pour son oreille attentive et ses bons conseils.

Je profite aussi de l'occasion pour exprimer toute ma gratitude envers monsieur Romano DeSantis et monsieur Stéphane Goulet qui m'ont beaucoup aidé lors de la première session de mon programme d'étude de maîtrise.

De plus, j'aimerais souligner la très grande collaboration des techniciens de la section automatique du département de Génie électrique et génie informatique, messieurs Yves Léonard et Richard Grenier.

Pour terminer, je remercie monsieur Stéphane Blouin (premier utilisateur du montage et du logiciel développé au cours de ce projet de maîtrise) pour ses remarques qui m'ont permis de corriger ou d'améliorer le logiciel. Je lui souhaite bon succès pour sa maîtrise.

RÉSUMÉ

La modélisation et la commande de robots à membrures flexibles font l'objet de nombreux travaux de recherche. Cependant peu d'essais expérimentaux ont été réalisés sur des montages pour évaluer et valider les stratégies de commande proposées.

Le présent projet de recherche a pour objectif de combler cette lacune en mettant au point un robot flexible expérimental dont la conception tant au niveau mécanique qu'au niveau de l'unité de commande, permet d'analyser les stratégies de commande.

Le robot expérimental qui a été fabriqué est un robot planaire, monté sur une base verticale et possède trois articulations et deux membrures flexibles. Le robot peut également être opéré avec un nombre réduit d'articulations et être équipé de membrures rigides.

Les actionneurs sont des moteurs à courant continu avec réducteur harmonique. La position et l'orientation de l'extrémité du robot sont évaluées à partir de la connaissance des angles des articulations et de mesures de jauges de contraintes fixées sur les membrures flexibles.

L'unité de commande est implantée sur un micro-ordinateur sous le système d'opération Windows 95. La modélisation et la simulation du robot sont réalisées à partir des logiciels MATLAB. MAPLE et SIMULINK. Les programmes de commande en temps réel ont été développés en langage C et sont conçus pour être utilisés dans un contexte de recherche; ils sont modulaires et permettent l'implantation rapide de nouvelles stratégies de commande.

ABSTRACT

There are many research projects about the mathematical modeling and the command of robots with flexible arms. There are only a few existing experiments made to evaluate and validate command strategies for this type of robots. The design of the robot built in this project will facilitate the analysis of robot with flexible links.

The experimental robot considered here is planar and installed on a vertical base. It has tree joints and two flexible links. It is also used in a smaller configuration (less joints) and with either rigid or flexible links. The actuators are direct current motors with harmonics drives reducers. We can evaluate the position and orientation of the extremity of the robot with joints positions and measurements of the deformation of the flexible links by strain gauges.

A micro-computer with the Windows operating system is used for the command unit. The MATLAB, MAPLE and SIMULINK software are needed for the modeling and the simulation of the robot. The real time software have been developed in C language and designed with the purpose of versatility of application in a research context. The software is modular and has a general structure that enable easy implementation of new control strategies.

TABLE DES MATIÈRES

DÉDICACE iv

REMERCIEMENTS v

RÉSUMÉ vi

ABSTRACT vii

TABLE DES MATIÈRES viii

LISTE DES TABLEAUX xiii

LISTE DES FIGURES xvi

LISTE DES SIGLES ET ABRÉVIATIONS xx

1 INTRODUCTION 1

1.1 Objectifs du projet 1

1.2 Méthodologie 1

1.3 Caractéristiques du projet 2

1.3.1 Nouveautés et importance du projet 3

1.3.2 Particularités du projet 3

1.3.3 Impact potentiel dans le domaine de la recherche 5

1.4 Contributions 5

2 CADRE THÉORIQUE: DESCRIPTION DES ÉQUATIONS 7

2.1 Modèle du robot expérimental 7

2.2 Modèle de la friction 9

2.3 Système en boucle fermé 10

2.4 Calculs hors ligne 12

2.4.1 Génération de trajectoires 12

2.4.2 Transformation en coordonnées rigides 26
2.4.3 Calcul de la commande anticipative 28
2.5 Simulation d'un robot avec friction aux articulations 34
2.5.1 Calculs du vecteur d'accélération et du vecteur de friction 35
2.5.2 Transformation en coordonnées des capteurs de flexibilité (courbure) 37
2.5.3 Contrôleur PD avec ajout d'une commande anticipative 37
2.5.4 Contrôleur PD pour les capteurs de flexibilité 38
2.6 Conclusion 38

3 CONCEPTION DU LOGICIEL ROVASIMEX 39
3.1 Objectifs 39
3.2 Caractéristiques 40
3.2.1 Acquisitions de données et contrôle en temps réel 42
3.3 Environnement informatique 43
3.4 Structure du logiciel 44
3.4.1 Descriptions des répertoires utilisés 44
3.4.2 Relation entre les fichiers d'entrée et les fichiers de sortie des modules 51
3.4.3 Noms par défaut des fichiers de sorties 53
3.5 Conclusion 55

4 UTILISATION DU LOGICIEL ROVASIMEX 56
4.1 Démarrage du logiciel 56
4.1.1 Ré initialisation du logiciel avec la commande "rstrvse" 58
4.2 Gestionnaire de projets 58
4.3 Configuration d'un système et génération de modèles symboliques 60
4.3.1 Assigner les référentiels du système 60
4.3.2 Écrire un fichier de paramètres symboliques Maple (*.SYM) 61
4.3.3 Écrire les fichiers de composantes du système 62

4.3.4 Créer un fichier de configuration (*.CFG) 63
4.3.5 Générer un modèle symbolique Maple (*.M) 63
4.4 Initialisation des paramètres globaux et génération de modèles numériques 64
4.4.1 Créer un fichier de paramètres numériques (*.PAR) 65
4.4.2 Créer un modèle numérique (*.MEX) 65
4.4.3 Mettre à jour un fichier de paramètres et les modèles numériques correspondants 66
4.5 Génération de trajectoires 66
4.5.1 Générer une trajectoire cinématique(*.TC?) 66
4.5.2 Éditer un fichier de trajectoire 68
4.5.3 Afficher la description d'une trajectoire 68
4.6 Calculs des paramètres d'entrée pour les essais en simulation ou expérimentaux 69
4.6.1 Créer un fichier de consignes (paramètres d'entrée) pour un essai en boucle ouverte (*.C0?) 69
4.6.2 Éditer un fichier de consignes (paramètres d'entrée) pour un essai en boucle ouverte 70
4.6.3 Créer un fichier de consignes (paramètres d'entrée) pour un essai en boucle fermée (*.C_?) 71
4.6.4 Créer un fichier de gains (paramètres) du contrôleur (*.G??) 72
4.7 Simulation du système 73
4.7.1 Effectuer une simulation avec comme entrée les couples (forces) aux articulations (*.S??) 73
4.7.2 Effectuer une simulation avec comme entrée les coordonnées rigides (*.S?R) . 74
4.8 Essais sur le système expérimental 74
4.8.1 Réaliser un essai expérimental (*.X??) 74
4.8.2 Vérifier ou identifier des paramètres du système 75
4.9 Traitements des données 75
4.9.1 Visualisations infographiques 76

4.9.2 Graphiques temporels 77
4.9.3 Graphiques (histogrammes) pour les statistiques de différences entre deux signaux temporels 78
4.9.4 Options disponibles 79
4.10 Implantation de nouveaux contrôleurs 79
4.10.1 Simulation avec un nouveau contrôleur 80
4.10.2 Essai expérimental avec un nouveau contrôleur 81
4.11 Résumé de l'utilisation du logiciel 81

5 CONCEPTION ET CARACTÉRISATION DU MONTAGE EXPÉRIMENTAL 83
5.1 Description du montage 83
5.1.1 Caractéristiques du montage 83
5.1.2 Choix des composantes 84
5.2 Identification des paramètres 87
5.2.1 Cartes d'interface 87
5.2.2 Amplificateurs 89
5.2.3 Actionneurs 90
5.2.4 Capteurs 92
5.2.5 Articulations 95
5.2.6 Membrures 97
5.3 Conclusion 102

6 SIMULATION ET EXPÉRIMENTATION 104
6.1 Description des trajectoires désirées 104
6.1.1 Trajectoires dans l'espace des variables d'articulations 105
6.1.2 Trajectoires dans l'espace de la tâche (plan vertical XY) 106
6.1.3 Effet de la discontinuité de la dérivée d'accélération sur la déformation des membrures flexibles 109

6.2 Assignation des gains du contrôleur 111
6.3 Vérification de la commande anticipative à l'aide de simulation en boucle ouverte (vecteurs de gains nuls) 112
6.3.1 Effet de la discrétisation et de la friction sur le système à une seule articulation (ID_A1_.PAR) 113
6.4 Vérification de la position (statique) de l'extrémité à l'aide de mesures sur papier millimétrique 115
6.5 Robot à trois articulations et deux membrures rigides (V3A2MR_.PAR) 117
6.5.1 Visualisation de l'erreur de l'extrémité entre résultats expérimentaux et positions désirées 117
6.5.2 Vérification de l'erreur dynamique pour un suivi de trajectoire 119
6.6 Robot à trois articulations et deux membrures flexibles (V3A2MF_.PAR) 122
6.6.1 Comparaison entre les valeurs mesurées par les capteurs de flexibilité et les valeurs estimés 122
6.6.2 Vérification du contrôleur des capteurs de flexibilité 122
6.7 Comparaisons de résultats avec un robot à deux membrures rigides (V3A2MR_.PAR) et un robot à deux membrures flexibles (V3A2MF_.PAR) 124
6.7.1 Visualisation des trajectoires de l'extrémité 124
6.7.2 Vérification des erreurs dynamiques 125
6.8 Conclusion 127

7 CONCLUSION 128
7.1 Limitations du système 128
7.2 Suggestions pour la poursuite des travaux 129

RÉFÉRENCES 131

LISTE DES TABLEAUX

Tableau 3.1: Contenu des répertoires des fonctions de bases: \RVSE_FCT 45
Tableau 3.2: Contenue des répertoires pour l'acquisition de données et le contrôle en temps réel: \RVSE_CTR 46
Tableau 3.3: Variables servant à décrire les sous répertoires associés à un système 47
Tableau 3.4: Contenue et description des sous répertoires associés à un système: \SYSTEMS 47
Tableau 3.5: Exemple des noms de répertoires associés au robot de ce projet de recherche 48
Tableau 3.6: Variables servant à décrire les répertoires associés à un projet 49
Tableau 3.7: Contenue et description des répertoires associés à un projet: \RVSEPROJ. 49
Tableau 3.8:Exemple des noms de répertoires associés au projet relié à ce mémoire 50
Tableau 3.9: Fichiers d'entrées des modules 51
Tableau 3.10: Fichiers de sorties des modules 52
Tableau 3.11: Fichiers de données pour l'analyse des résultats (identifications et graphiques) 52
Tableau 3.12: Liste des choix lors de la génération des modèles symboliques et numériques 54
Tableau 3.13: Noms par défaut des fichiers de sorties 54
Tableau 3.14: Liste des premiers caractères de l'extension des fichiers de statistiques 55
Tableau 4.1: Options des fichiers de consignes pour les essais en boucle fermée (*.C_?) 72
Tableau 4.2: Liste des paramètres pouvant être vérifiés ou identifiés 75
Tableau 4.3: Liste des unités disponibles pour tracer les graphiques 79
Tableau 4.4: Exemple de fichiers créés lorsqu'on ajoute un contrôleur (ctrlA) 80
Tableau 5.1: Avantages et inconvénients des composantes du robot 86
Tableau 5.2: Temps d'exécution de la routine d'interruption en temps réel 89
Tableau 5.3: Paramètres des amplificateurs 89
Tableau 5.4: Paramètres des réducteurs harmoniques 90

Tableau 5.5: Inerties des actionneurs 90
Tableau 5.6: Paramètres identifiés de frictions actionneurs 91
Tableau 5.7: Gains des actionneurs, spécifications du manufacturier et paramètres identifiés 92
Tableau 5.8: Paramètres des capteurs de position 92
Tableau 5.9: Paramètres des articulations reliées aux pièces qui les précèdent 97
Tableau 5.10: Paramètres de la partie mobile des articulations 97
Tableau 5.11: Paramètres des membrures rigides 98
Tableau 5.12: Paramètres des membrures flexibles 98
Tableau 5.13: Paramètres des capteurs de flexibilité 98
Tableau 5.14: Erreur maximum entre résultats calculés avec fonctions de forme et avec formules de poutre encastrée 102
Tableau 6.1: Vitesses d'exécutions des trajectoires 105
Tableau 6.2: Paramètres limites des trois articulations 106
Tableau 6.3: Description des noms de fichiers de trajectoires dans l'espace des variables d'articulations 106
Tableau 6.4: Paramètres maximum des trajectoires dans l'espace de la tâche 107
Tableau 6.5: Description des noms de fichiers de trajectoires (extrémité) dans l'espace de la tâche 108
Tableau 6.6: Dimensions des formes géométriques tracés par l'extrémité du robot 109
Tableau 6.7: Gains pour le robot à trois articulations et deux membrures rigides 111
Tableau 6.8: Gains pour le robot à trois articulations et deux membrures flexibles 112
Tableau 6.9: Effet de la friction pour simulations discrètes ou continues (Légende supplémentaire) 113
Tableau 6.10: Effet de la discrétisation pour des simulations sans friction (Légende supplémentaire) 114
Tableau 6.11: Effet de la discrétisation pour des simulations avec friction (Légende supplémentaire) 114

Tableau 6.12: Erreurs (distances) moyennes de positions de l'extrémité pour robot à membrures rigides (V3A2MR.PAR) 115
Tableau 6.13: Erreurs (distances) moyennes de positions de l'extrémité pour robot à membrures flexibles (V3A2MF.PAR) 115

LISTE DES FIGURES

Figure 1.1: Illustration du banc d'essai de l'École Polytechnique de Montréal 4
Figure 2.1: Friction dynamique (polynôme du premier ordre) 10
Figure 2.2: Diagramme bloc du système en boucle fermée 11
Figure 2.3: Exemple de trajectoire avec dérivée d'accélération triangulaire 14
Figure 2.4: Exemple de trajectoire avec dérivée d'accélération échelon 18
Figure 2.5: Exemple de trajectoire avec un temps d'accélération défini et la dérivée d'accélération sans discontinuité 21
Figure 2.6: Exemple de trajectoire avec un temps d'accélération défini et la dérivée d'accélération discontinue 24
Figure 2.7: Cinématique inverse d'un robot planaire à trois articulations 27
Figure 2.8: Visualisation du système sans correction aux articulations 31
Figure 2.9: Visualisation de la correction sur la première articulation 32
Figure 2.10: Visualisation de la correction sur la deuxième articulation 32
Figure 2.11: Visualisation avant la correction sur la troisième articulation 33
Figure 2.12: Visualisation après la correction sur la troisième articulation 33
Figure 3.1: Fenêtre principale du logiciel ROVASIMEX 41
Figure 3.2: Arbre des répertoires contenant les fonctions de base (Matlab et Maple) du logiciel ROVASIMEX 46
Figure 3.3: Arbre des répertoires associés à un système 48
Figure 3.4: Arbre des répertoires associés à un projet 50
Figure 3.5: Structure hiérarchique des fichiers de données 53
Figure 4.1: Fenêtre d'interface pour ajouter un projet 57
Figure 4.2: Fenêtre d'interface de gestionnaire de projet 58
Figure 4.3: Fenêtre d'interface pour confirmer l'effacement d'un projet 59
Figure 4.4: Fenêtre d'interface pour enregistrer les résultats d un projet sur disquette 59
Figure 4.5: Visualisation des référentiels et des centres de masse 61

Figure 4.6: Fenêtre d'interface de génération de modèle symbolique 64
Figure 4.7: Fenêtre d'interface pour créer un fichier de consignes pour un essai en boucle ouverte 69
Figure 4.8: Fenêtre d'interface pour créer un fichier de consignes pour un essai en boucle fermée 71
Figure 4.9: Fenêtre d'interface pour ajouter un contrôleur 80
Figure 5.1: Erreur de lecture des encodeurs optiques 93
Figure 5.2: Acquisition de données sans filtrage 94
Figure 5.3: Résultats après filtrage des données 95
Figure 5.4: Déformations d'une membrure, calculées avec fonctions de formes et avec formule de poutre encastrée 100
Figure 5.5: Pentes d'une membrure, calculées avec fonctions de formes et avec formule de poutre encastrée 100
Figure 5.6: Courbures d'une membrure, calculées avec fonctions de formes et avec formule de poutre encastrée 101
Figure 5.7: Courbures d'une membrure, calculées avec trois fonctions de formes et avec formule de poutre encastrée 102
Figure 6.1: Effet de la discontinuité de la dérivée d'accélération (triangulaire : courbe en continu ; échelon : courbe en pointillée) sur la déformation de la membrure #1 110
Figure 6.2: Effet de la discontinuité de la dérivée d'accélération (triangulaire : courbe en continu ; échelon : courbe en pointillée) sur la déformation de la membrure #2 110
Figure 6.3: Effet de la friction pour des simulations discrètes ou continues 113
Figure 6.4: Effet de la discrétisation pour des simulations sans friction 114
Figure 6.5: Effet de la discrétisation pour des simulations avec friction 114
Figure 6.6: Décalage du tracé en fonction de la distance entre le panneau et le support du crayon 117
Figure 6.7: Cercle à vitesse moyenne (CMR) 118
Figure 6.8: Cercle à vitesse lente (CLR) 118

Figure 6.9: Cercle avec déplacement dans le sens horaire (CNR) 118
Figure 6.10: Cercle avec dérivée d'accélération. discontinue (CDR) 118
Figure 6.11: Cercle dans la région stable (y<0) (CSR) 118
Figure 6.12: Cercle dans la région instable (y>0) (CIR) 118
Figure 6.13: Comparaison coordonnée rigide #1 (CMR) 119
Figure 6.14: Erreur coordonnée rigide #1 (CMR) 119
Figure 6.15: Comparaison coordonnée rigide #2 (CMR) 119
Figure 6.16: Erreur coordonnée rigide #2 (CMR) 119
Figure 6.17: Comparaison coordonnée rigide #3 (CMR) 120
Figure 6.18: Erreur coordonnée rigide #3 (CMR) 120
Figure 6.19: Erreur horizontale de l'extrémité (CMR) 120
Figure 6.20: Erreur verticale de l'extrémité (CMR) 120
Figure 6.21: Comparaison de vitesse tangentielle de l'extrémité (CMR) 121
Figure 6.22: Comparaison d'orientation de l'extrémité (CMR) 121
Figure 6.23: Erreur de vitesse tangentielle de l'extrémité (CMR) 121
Figure 6.24: Erreur d'orientation de l'extrémité (CMR) 121
Figure 6.25: Comparaison capteurs de flexibilité pour la membrure #1 (CMR) 122
Figure 6.26: Comparaison capteurs de flexibilité pour la membrure #2 (CMR) 122
Figure 6.27: Effet du contrôleur des capteurs de flexibilité de la membrure #1 123
Figure 6.28: Effet du contrôleur des capteurs de flexibilité de la membrure #2 123
Figure 6.29: Carré avec déplacement dans le sens anti-horaire (PMR) 124
Figure 6.30: Carré avec déplacement dans le sens horaire (PNR) 124
Figure 6.31: Comparaison de cercles avec déplacement dans le sens anti horaire (CMR) 125
Figure 6.32: Comparaison de cercles avec déplacement dans le sens horaire (CNR) ... 125
Figure 6.33:Comparaison de cercles dans la région stable (CSR) 125
Figure 6.34:Comparaison de cercles dans la région instable (CIR) 125

Figure 6.35: Comparaison de position verticale de l'extrémité du robot à membrures rigides (PMR) 126
Figure 6.36: Erreur de position verticale de l'extrémité du robot à membrures rigides (PMR) 126
Figure 6.37: Comparaison de position verticale de l'extrémité du robot à membrures flexibles (PMR) 126
Figure 6.38: Erreur de position verticale de l'extrémité du robot à membrures flexibles (PMR) 126
Figure 6.39: Statistiques d'erreurs sur la coordonnée rigide #1 (PMR) 127
Figure 6.40: Statistiques d'erreurs sur la coordonnée rigide #2 (PMR) 127

LISTE DES SIGLES ET ABRÉVIATIONS

Sigle	Description
ROVASIMEX	Logiciel: RObotique VAlidation SIMulation EXpérimentation
SYMOFROS	Logiciel: SYmbolic MOdeling of Flexible RObots ans Simulation

Notation utilisée pour la rédaction de ce mémoire:

- les matrices et les vecteurs sont en caractères gras;
- les matrices sont en majuscules tandis que les vecteurs sont en minuscules;
- les variables sont en caractères italiques;
- les variables entre accolades indiquent les dimensions (nombre de lignes et nombre de colonnes) des vecteurs et des matrices.

Symboles	Description des symboles utilisés pour définir des paramètres
a	accélération
a	coefficients du polynômes de friction dynamique
b_{visq}	coefficient d'amortissement visqueux
$depl$	déplacement relatif
fr	friction dynamique
fr_{coul}	friction de Coulomb
fr_{stat}	friction statique
L	longueur d'un segment du robot (distance entre deux articulations)
n	nombre de colonnes ou de rangées d'un vecteur ou d'une matrice
p	position
t	temps
v	vitesse
x, y, θ	position et orientation de l'extrémité du robot
x_2, y_2	position de l'extrémité du deuxième segment
φ	angle du premier segment par rapport à l'axe X du référentiel observateur

Symbole	Description des symboles utilisés pour définir des vecteurs ou des matrices
$\mathbf{0}_n$	Vecteurs nuls de n rangés
$\mathbf{B}_l$	matrice d'amortissement du modèle linéarisé
$\mathbf{g}$	vecteur de force ou couple fonction de l'accélération gravitationnelle
$\mathbf{h}$	vecteur
$\mathbf{K}_l$	matrice de rigidité du modèle linéarisé
$\mathbf{M}_{nl}$	matrice de masse ou d'inertie
$\mathbf{p}$	vecteur de position et d'orientation de l'extrémité
$\mathbf{q}$	vecteurs de coordonnées du modèle
$\Delta\mathbf{q}$	vecteur d'écart par rapport au point de linéarisation
$\mathbf{u}$	vecteur d'entré du modèle
$\mathbf{v}$	vecteur de force de Coriollis et force centrifuge
Γ_{nl}	vecteur de force ou couple effectif
η	sous vecteurs de coordonnées flexibles

Indices	Description des indices utilisés
0	point de linéarisation
a	accélération
c	courbure
d	désirée
e	élastique
f	flexible
p	position
r	rigide
s	simulée
t	trajectoire cinématique
v	flexion
v	vitesse
w	flexion
α	torsion
Γ	couple ou force

1 INTRODUCTION

Pour bien situer ce projet de maîtrise dans le domaine de la robotique et avant d'aborder plus à fond le projet, le chapitre qui suit présente une description sommaire du projet. Les principaux objectifs du projet et la méthodologie préconisée sont décrits. Les principales caractéristiques du projet et un résumé du travail effectué (contributions) sont présentés.

1.1 Objectifs du projet

Le but premier de ce projet est de concevoir et fabriquer un robot expérimental à membrures flexibles. Le système expérimental comprend un robot opérant dans le plan vertical et un logiciel (ROVASIMEX: "RObotique, VAlidation, SIMulation, EXpérimentation) qui permet d'étudier le comportement du robot par simulation et par des essais avec acquisition de données et contrôle en temps réel. La conception de ce robot comprend la détermination de tous les paramètres mécaniques des actionneurs et des capteurs d'un robot pouvant posséder de une à trois articulations rotoïdes et de une à deux membrures qui peuvent être soient rigides ou flexibles dans le plan vertical. Le logiciel permettant de réaliser l'acquisition de données et le contrôle en temps réel est conçu pour être utilisé pour différents robots et permet l'implantation de plusieurs type de contrôleurs.

1.2 Méthodologie

Dans le cadre de ce projet, une stratégie de commande en temps réel a été développée et implantée pour un robot expérimental possédant trois articulations et deux membrures flexibles et travaillant dans un plan vertical.

Ce projet a été réalisé selon les étapes suivantes:

- conception mécanique et fabrication d'un robot expérimental facilitant l'analyse et la validation de modèles (membrures flexibles) de même que l'étude de stratégies de contrôle;
- développement des fonctions de base du logiciel temps réel;
- identification des paramètres du système: calculs des paramètres à partir des caractéristiques de conception, mesure et validation des paramètres, estimations des paramètres manquants;
- modélisation du robot et conception des programmes de simulation pour valider les modèles;
- conception et simulation d'un contrôleur anticipatif permettant un suivi de trajectoire désirée de l'extrémité du robot;
- implantation en temps réel du contrôleur anticipatif: simulation, mise au point expérimentale, comparaison de résultats;
- conclusions et recommandations pour la poursuite de travaux.

Il est à remarquer que le robot peut posséder des membrures rigides ou flexibles selon le montage mécanique retenu. L'ensemble des étapes décrites précédemment ont donc été réalisées à la fois pour le robot à membrures rigides et à membrures flexibles.

D'autre part, l'ensemble des logiciels développés à l'intérieur de ce projet, ont été conçus pour être utilisés dans un contexte de recherche, i.e. avec comme objectif de pouvoir mettre en oeuvre et de tester rapidement de nouvelles notions théoriques. Il a donc fallu établir une structure décrivant les paramètres d'un robot manipulateur et définir une structure pour les fonctions de base de façon à intégrer les divers éléments dans un logiciel.

1.3 Caractéristiques du projet

Pour compléter la description du projet, on en présente les principales caractéristiques.

1.3.1 Nouveautés et importance du projet

L'apport principal du projet est au niveau de la conception d'un robot expérimental (robot à trois articulations et deux membrures flexibles) et la conception d'un logiciel qui rend disponibles à l'usager les fonctions suivantes: configuration d'un robot; initialisation des paramètres et calcul des capacités d'un robot; génération de trajectoires; calcul de commandes anticipatives; simulation d'un système; contrôle en temps réel; traitement des données. Le montage expérimental permet de vérifier et de comparer des résultats de simulation avec des résultats expérimentaux. Ce système sera utilisé par la suite pour des travaux de recherche et pourra également servir de banc d'essai pour des travaux pratiques au niveau des cours de robotique.

1.3.2 Particularités du projet

La majorité des articles sur les robots flexibles rapportent des résultats sur des bancs d'essai ayant une seule articulation et une seule membrure flexible. Il existe quelques bancs d'essai pour des systèmes à deux membrures flexibles. Le robot sphérique de l'université de Munich (BREMER, H. et PFEIFFER, F., 1994) possède cinq articulations et deux membrures flexibles. Au Canada, un robot sphérique à trois articulations est utilisé à University of Western Ontario (ELMARAGHY, W.H. et ZAKI, A. S., 1994). L'université de Victoria en Colombie Britanique (DAMAREN, C., SHARF, I. et STANWAY, J., 1996) possède deux robots installés sur un table horizontale. Les deux articulations des robots sont supportées par un coussin d'air. L'université de Toronto (BUCHAN, K.S., CARUSONE, J. et D'ELEUTERIO, G.M.T., 1996) possède un robot (RADIUS) semblable aux robots de l'université de Victoria.

Le robot expérimental (Figure 1.1) de ce projet possède trois articulations permettant un mouvement dans le plan vertical, avec contrôle de la position et de l'orientation de

l'extrémité. Pour contrôler les déplacements, une commande anticipative est calculée en fonction de la trajectoire des articulations et de la flexibilité des membrures.

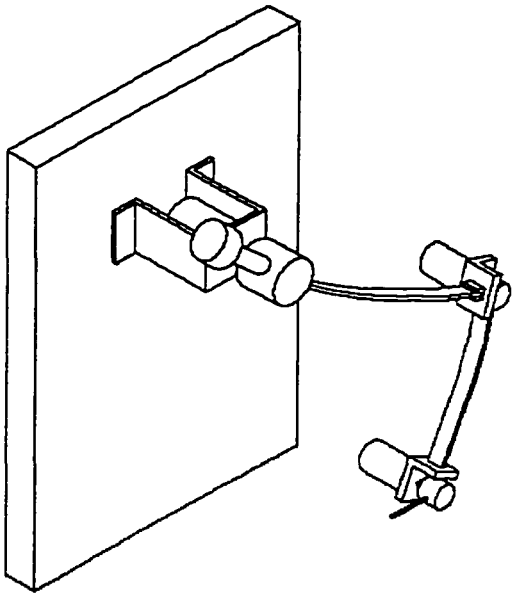


Figure 1.1: Illustration du banc d'essai de l'École Polytechnique de Montréal

Une des particularités du projet est que ROVASIMEX réuni dans un même logiciel plusieurs aspects de la robotique : la modélisation, la simulation et le contrôle en temps réel. D'autres logiciels tels que MECANO et la libraire de fonctions Matlab de Corke (1996) ne comprend pas l'aspect contrôle en temps réel. Le logiciel ROVASIMEX est développé dans un environnement Matlab. Il fait également appel aux fonctions de génération de modèles symboliques et numériques du logiciel SYMOFROS (Symbolic Modeling of Flexible Robots and Simulation) écrit par M. Jean-Claude Piedboeuf. Généralement, les trajectoires générées pour effectuer des simulations ne tiennent pas compte des capacités et des limitations du robot. Étant donnée que ce projet comporte des essais expérimentaux, les fonctions de générations de trajectoires qui ont été développé ont pour arguments les limites du système et respectent donc les capacités du robot.

1.3.3 Impact potentiel dans le domaine de la recherche

Le logiciel développé dans ce projet est un outil d'aide à la conception d'un robot. Il permet d'effectuer la conception de différents robots et de vérifier leurs comportements par simulation. Les résultats obtenus guideront le choix des composantes à acheter et la fabrication du robot. Ceci représente d'importantes économies d'argent. En plus d'être utile à la conception d'un robot, le logiciel permet de développer et vérifier les notions théoriques au niveau de la modélisation et des stratégies de contrôle. Les fonctions de base disponibles au niveau logiciel tel que la configuration du système, l'initialisation des paramètres, l'acquisition de données expérimentales, ont été conçues de façon à pouvoir être utilisés facilement par les usagers.

1.4 Contributions

La liste[1] qui suit présente un résumé du travail effectué dans le cadre de ce projet :

- mécanique :
    - participation à la conception de la deuxième et la troisième articulation du robot (dessins techniques) ;
    - sélection des membrures ;
    - utilisation d'un panneau vertical pour tracer l'extrémité du robot ;
- électrique :
    - participation à la conception du câblage du système ;
    - participation à la conception du système de protection du montage (enclenchement de relais pour les commandes aux actionneurs) ;
- identification et vérification des paramètres du système ;

[1] Certains éléments de cette liste ne sont pas décris dans ce document mais plutôt dans le document : Manuel technique du banc d'essai pour l'étude théorique et expérimentale de robots flexibles, École Polytechnique de Montréal.

- calculs (avec le logiciel Lotus) d'inertie et de centre de masse des articulations et des charges utilisées ;
- réalisation d'essais en boucle ouverte ;
- calibration et vérification des capteurs de flexibilité ;

- conception et réalisation logiciel :
  - ROVASIMEX ;
  - intégration (utilisation) du logiciel SYMOFROS ;
  - implantation d'un contrôleur PID avec commande anticipative ;
  - routines de base (driver) pour les cartes d'interfaces ;
  - routines en langage C pour extraire une seule variable d'un fichier de données Matlab et pour écrire les résultats expérimentaux des fichiers de données Matlab, ceci dans plusieurs formats de données (ex.: int, long, double, *char);
  - programme de commande en boucle ouverte avec acquisition de données en temps réel ;
  - programme de contrôle et d'acquisition de données en temps réel ;
- réalisation d'essais préliminaires (boucle fermée) ;
- documentation du système.

2 CADRE THÉORIQUE: DESCRIPTION DES ÉQUATIONS

Ce chapitre décrit les différents modèles utilisées pour représenter le robot expérimental fabriqué au cours de ce projet. Les modèles permettent de tenir compte de l'élasticité au niveau des articulations, de la flexibilité des membrures et des phénomènes de friction. Les algorithmes de génération de trajectoires qui sont actuellement disponibles dans le logiciel ROVASIMEX sont également décrits. Finalement, la stratégie de commande de type anticipative qui a été implantée sur le système est présentée en détail.

2.1 Modèle du robot expérimental

Le modèle du robot tient compte de l'élasticité des articulations et de la flexibilité des membrures.

Trois types de coordonnées sont utilisées pour modéliser le comportement d'un robot à membrures flexibles et possédant de l'élasticité aux articulations:

- $\mathbf{q}_r$: vecteur $\{n_r \times 1\}$ de coordonnées rigides;
- $\mathbf{q}_e$: vecteur $\{n_e \times 1\}$ de coordonnées élastiques;
- $\mathbf{q}_f$: vecteur $\{n_f \times 1\}$ de coordonnées flexibles.

Lorsqu'il n'y a pas de coordonnées élastiques, les coordonnées rigides représentent la position des articulations. Sinon la position des articulations est la somme des coordonnées rigides et des coordonnées élastiques: $\mathbf{q}_j = \mathbf{q}_r + \mathbf{q}_e$.

Le nombre de coordonnées flexibles correspond au nombre total de fonctions de forme utilisées dans la méthode des modes supposés. Les coordonnées sont regroupées par

membrure. Chaque groupe est subdivisé selon le type de déformation (les flexions v, w et la torsion α).

$$\mathbf{q}_f = \begin{bmatrix} \mathbf{q}_{f_1} \\ \mathbf{q}_{f_2} \\ \vdots \end{bmatrix}, \quad \mathbf{q}_{f_i} = \begin{bmatrix} {}^{v_i}\boldsymbol{\eta} \\ {}^{w_i}\boldsymbol{\eta} \\ {}^{\alpha_i}\boldsymbol{\eta} \end{bmatrix}$$

avec

- ${}^{v_i}\boldsymbol{\eta}$, ${}^{w_i}\boldsymbol{\eta}$, ${}^{\alpha_i}\boldsymbol{\eta}$: sous vecteurs $\{n_{v_i} \times 1\}, \{n_{w_i} \times 1\}, \{n_{\alpha_i} \times 1\}$ de coordonnées flexibles;
- $n_{v_i}, n_{w_i}, n_{\alpha_i}$: nombre de fonctions de formes (modes supposés) pour chacune des déformations de la membrure i .

Le vecteur de coordonnées généralisées est:

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_r \\ \mathbf{q}_e \\ \mathbf{q}_f \end{bmatrix}$$

et le modèle non-linéaire du robot s'écrit:

Éq. 2.1

$$\boxed{\begin{aligned} &\mathbf{M}_{nl}(\mathbf{q}, \mathbf{u}_p)\ddot{\mathbf{q}} + \boldsymbol{\Gamma}_{nl}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u}) = \mathbf{0}_n \\ &\boldsymbol{\Gamma}_{nl}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u}) = \mathbf{v}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u}_p, \mathbf{u}_v) + \mathbf{g}(\mathbf{q}, \mathbf{u}_p) - \mathbf{u}_\Gamma \end{aligned}}$$

avec

- $\mathbf{M}_{nl}$: matrice $\{n \times n\}$ de masse ou d'inertie;
- $\boldsymbol{\Gamma}_{nl}$: vecteur $\{n \times 1\}$ de force ou couple effectif;
- $\mathbf{v}$: vecteur $\{n \times 1\}$ de force de Coriolis et force centrifuge;
- $\mathbf{g}$: vecteur $\{n \times 1\}$ de force ou de couple fonction de l'accélération gravitationnelle;
- $\mathbf{u}$: vecteur $\{n_u \times 1\}$ d'entrée du modèle (force, couple, position, vitesse, accélération);

- $\mathbf{u}_\Gamma$: composantes force, couple du vecteur d'entrée $\mathbf{u}$;
- $\mathbf{u}_p$: composantes position du vecteur d'entrée $\mathbf{u}$;
- $\mathbf{u}_v$: composantes vitesse du vecteur d'entrée $\mathbf{u}$;
- $\mathbf{u}_a$: composantes accélération du vecteur d'entrée $\mathbf{u}$;

Le modèle linéarisé autour du point de fonctionnement $(\mathbf{q}_0, \dot{\mathbf{q}}_0, \ddot{\mathbf{q}}_0, \mathbf{u}_{p_0}, \mathbf{u}_{v_0}, \mathbf{u}_{a_0})$ s'écrit:

Éq. 2.2

$$\begin{aligned}&\mathbf{M}_{nl}\left(\mathbf{q}_0, \mathbf{u}_{p_0}\right)\Delta\ddot{\mathbf{q}} + \mathbf{B}_l\left(\mathbf{q}_0, \dot{\mathbf{q}}_0, \mathbf{u}_{p_0}, \mathbf{u}_{v_0}\right)\Delta\dot{\mathbf{q}} + \mathbf{K}_l\left(\mathbf{q}_0, \dot{\mathbf{q}}_0, \ddot{\mathbf{q}}_0, \mathbf{u}_{p_0}, \mathbf{u}_{v_0}, \mathbf{u}_{a_0}\right)\Delta\mathbf{q} \\ &+ \mathbf{B}_u\left(\mathbf{q}_0, \dot{\mathbf{q}}_0, \ddot{\mathbf{q}}_0, \mathbf{u}_{p_0}, \mathbf{u}_{v_0}, \mathbf{u}_{a_0}\right)\Delta\mathbf{u} \;=\; -\left(\mathbf{M}_{nl}\left(\mathbf{q}_0, \mathbf{u}_{p_0}\right)\ddot{\mathbf{q}}_0 + \Gamma_{nl}\left(\mathbf{q}_0, \dot{\mathbf{q}}_0, \mathbf{u}_{p_0}, \mathbf{u}_{v_0}, \mathbf{u}_\Gamma\right)\right)\end{aligned}$$

avec

- $\Delta\mathbf{q}, \Delta\dot{\mathbf{q}}, \Delta\ddot{\mathbf{q}}$: vecteurs $\{n \times 1\}$ d'écart par rapport au point de fonctionnement;
- $\mathbf{B}_l$: matrice $\{n \times n\}$ d'amortissement;
- $\mathbf{K}_l$: matrice $\{n \times n\}$ de rigidité.

2.2 Modèle de la friction

La friction est un phénomène complexe qui peut être décrit par un modèle non linéaire dépendant de la vitesse. Si la vitesse est nulle, alors il y a friction statique, sinon il y a friction dynamique. Le friction statique s'oppose à la force appliquée sans excéder une certaine valeur maximum $\left(fr_{stat_i}\right)$.

si $\dot{q}_{r_i} = 0, \quad -fr_{stat_i} \le fr_i \le fr_{stat_i}$

En général (KUO, B.C., 1987), la friction dynamique est modélisée sous forme d'un polynôme du premier ordre (somme de la friction de Coulomb et de la friction visqueuse).

si $\dot{q}_{r_i} \neq 0, \quad fr_i\left(\dot{q}_{r_i}\right) = \mathrm{sgn}\left(\dot{q}_{r_i}\right) fr_{coul_i} + b_{visq_i}\,\dot{q}_{r_i}$

avec

- sgn( ): fonction signe;
- fr_i: friction dynamique;
- fr_{coul_i}: friction de Coulomb;
- b_{visq_i}: coefficient d'amortissement visqueux.

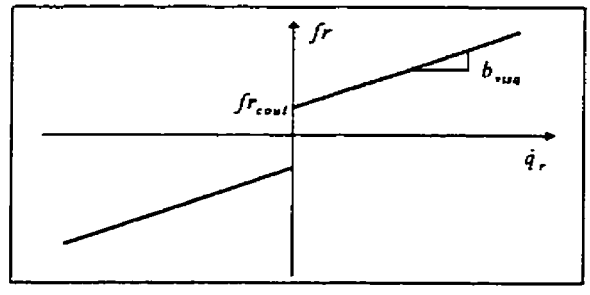


Figure 2.1: Friction dynamique (polynôme du premier ordre)

Dans le cas du robot décrit dans ce mémoire, la friction dynamique est représentée de façon plus exacte par un polynôme du deuxième ordre, soit:

Éq. 2.3

$$\text{si} \quad \dot{q}_{r_i} \neq 0, \quad fr_i\left(\dot{q}_{r_i}\right) = \text{sgn}\left(\dot{q}_{r_i}\right)\left(a_{2_i}\left|\dot{q}_{r_i}\right|^2 + a_{1_i}\left|\dot{q}_{r_i}\right| + a_{0_i}\right)$$

avec

- $a_{2_i}, a_{1_i}, a_{0_i}$: coefficients du polynôme de friction dynamique.

La friction statique et la friction de Coulomb (effets non-linéaires de discontinuité) sont rajoutées lors des simulations du système.

2.3 Système en boucle fermé

Tel qu'énoncé précédemment, un contrôleur anticipatif utilisant une commande calculée hors ligne qui tient compte de la flexibilité des membrures et de l'élasticité des articulations a été implanté. La Figure 2.2 présente le diagramme bloc du système en boucle fermée.

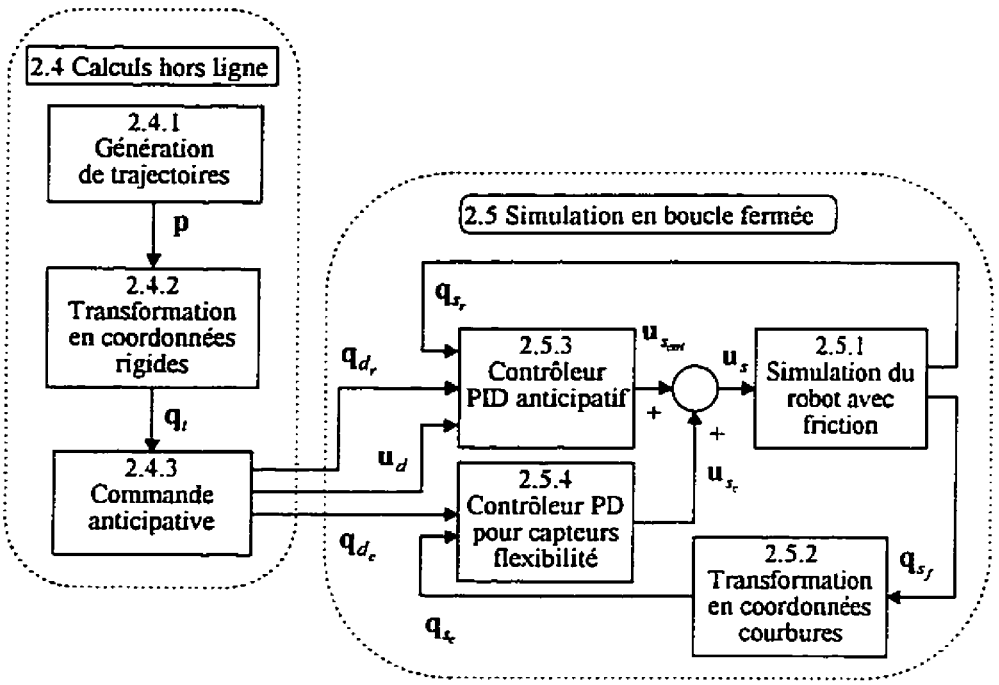


Figure 2.2: Diagramme bloc du système en boucle fermée

avec

- $\mathbf{p}$: vecteur des positions et d'orientations désirées l'extrémité du robot;
- $\mathbf{q}_t$: vecteur des coordonnées rigides calculé à partir des trajectoires (sans tenir compte de la flexibilité et de l'élasticité);
- $\mathbf{q}_{d_r}$: vecteur des coordonnées rigides désirées (avec correction due à la flexibilité et l'élasticité des articulations);
- $\mathbf{q}_{d_c}$: vecteur des coordonnées capteurs de flexibilité désirées (courbure);
- $\mathbf{u}_d$: vecteur de commandes anticipatives (désirées);
- $\mathbf{q}_{s_r}$: vecteur des coordonnées rigides simulées;
- $\mathbf{q}_{s_f}$: vecteur des coordonnées flexibles simulées;
- $\mathbf{q}_{s_c}$: vecteur des coordonnées capteurs de flexibilité simulées (courbure pour mesure en flexion, pente pour mesure en torsion);
- $\mathbf{u}_s$: vecteur de commandes simulées;

- $\mathbf{u}_{s_{ant}}$: vecteur de commandes provenant du contrôleur PD anticipatif;
- $\mathbf{u}_{s_c}$: vecteur de commandes provenant du contrôleur PD pour les capteurs de flexibilité.

2.4 Calculs hors ligne

2.4.1 Génération de trajectoires

Le déplacement d'un robot d'un point à un autre dans l'espace de travail est réalisé concrètement en générant des trajectoires désirées qui devront être suivies par le robot. Ces trajectoires sont générées pour les coordonnées rigides seulement et doivent tenir compte des capacités des actionneurs, c'est-à-dire des limites en vitesse et en couple. Les trajectoires (fonctions du temps) sont calculées à l'aide de fonctions de base[1] ayant comme arguments: un déplacement relatif (*depl*), une période d'échantillonnage et des paramètres limites (vitesse, accélération et dérivée d'accélération maximum). Les trajectoires calculées sont ensuite converties pour obtenir les trajectoires désirées des articulations. Par exemple, lorsqu'il s'agit d'une trajectoire dans l'espace de la tâche (pour notre robot dans le plan XY), la trajectoire est calculée pour un déplacement relatif avec comme point de départ une valeur nulle, par la suite le résultat est translaté vers le point de départ (dans le plan XY). Ce nouveau résultat est ensuite converti par cinématique inverse pour obtenir la trajectoire désirée des articulations.

2.4.1.1 Trajectoires désirées en fonction du temps et des paramètres limites

Il existe plusieurs livres traitant de la génération de trajectoires. Le livre de Craig (1989) et celui de BRADY (1984) en sont deux exemples.

---

[1] Pour voir le détails des calculs (équations) pour l'implantation des fonctions de génération de trajectoires consulter le document: Manuel technique du banc d'essai pour l'étude théorique et expérimentale de robots flexible, École Polytechnique de Montréal.

Pour ce projet quatre fonctions de bases de génération de trajectoires sont implantées:

1) avec la dérivée de l'accélération triangulaire;
2) avec la dérivée de l'accélération échelon;
3) avec un temps d'accélération défini et la dérivée d'accélération sans discontinuité;
4) avec un temps d'accélération défini et la dérivée d'accélération discontinue.

Les fonctions calculent la position, la vitesse et l'accélération en fonction du temps, avec une position initiale égale à zéro et avec une vitesse et une accélération initiales et finales nulles.

Les deux premières fonctions génèrent une trajectoire (la plus rapide possible) qui respecte les valeurs limites (ou valeurs de références) pour la vitesse, l'accélération et la dérivée d'accélération. Le temps de la trajectoire n'est pas fixé, il est fonction des valeurs limites de la trajectoire et du déplacement demandé. Pour vérifier si les paramètres limites sont atteints, il faut déterminer la position p_i, la vitesse v_i, l'accélération a_i et la dérivée d'accélération $\dot{a}_i$, pour chacun des temps t_i suivants:

- i=0 : point initial;
- i=1 : début d'accélération constante;
- i=2 : fin d'accélération constante;
- i=3 : fin de la phase d'accélération (début de vitesse constante).

D'autres points (temps t_i) sont utilisés pour compléter les calculs de génération du déplacement relatif (*depl*):

- i=4 : fin de la vitesse constante;
- i=5 : début de décélération constante;
- i=6 : fin de décélération constante.

Les fonctions de base #3 et #4 génèrent une trajectoire qui respecte seulement la valeur limite de vitesse. Les fonctions utilisent un paramètre temps d'accélération plutôt que la limite d'accélération pour calculer la trajectoire.

2.4.1.2 Trajectoires avec dérivée d'accélération triangulaire

La Figure 2.3 donne un exemple de trajectoire avec dérivée d'accélération triangulaire.

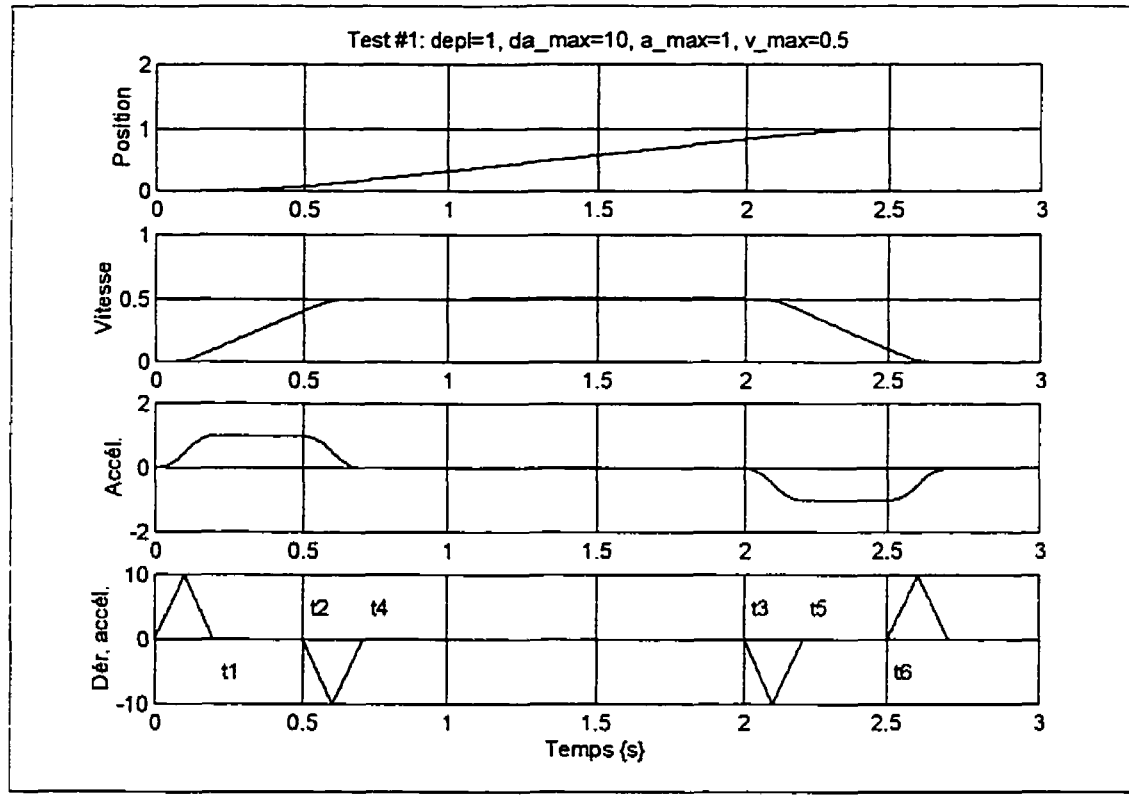


Figure 2.3: Exemple de trajectoire avec dérivée d'accélération triangulaire

La trajectoire avec dérivée d'accélération triangulaire est calculée par segment. Sachant que la seconde dérivée d'accélération est constante pendant un segment, elle peut prendre alors une des trois valeurs possibles:

$$\ddot{a}(t) = b = \left\{ \ddot{a}_d \quad ou \quad 0 \quad ou \quad -\ddot{a}_d \right\}$$

En utilisant comme conditions initiales:

$$p_i = p(t_i), v_i = v(t_i), a_i = a(t_i), \dot{a}_i = \dot{a}(t_i)$$

On obtient alors les équations de base servant à calculer les trajectoires en fonction du temps:

Éq. 2.4

$$\dot{a}(t) = b\Delta t + \dot{a}_i$$

$$a(t) = \frac{b\Delta t^2}{2} + \dot{a}_i \Delta t + a_i$$

$$v(t) = \frac{b\Delta t^3}{6} + \frac{\dot{a}_i \Delta t^2}{2} + a_i \Delta t + v_i$$

$$p(t) = \frac{b\Delta t^4}{24} + \frac{\dot{a}_i \Delta t^3}{6} + \frac{a_i \Delta t^2}{2} + v_i \Delta t + p_i$$

avec:

$$\Delta t = t - t_i$$

Pour générer une trajectoire avec dérivée d'accélération triangulaire, il faut d'abord calculer la dérivée seconde d'accélération $\left(\ddot{a}_d\right)$.

Sachant que la dérivée d'accélération maximum $\dot{a}_{max}$ est atteinte lorsque $t = \frac{t_1}{2}$ on obtient:

Éq. 2.5

$$\ddot{a}_d = \frac{2\dot{a}_{max}}{t_1}$$

donc au point #1 on a:

$$a_1 = a_{max} = -\frac{2\dot{a}_{max}}{t_1}\left(\frac{t_1}{2}\right)^2 + \dot{a}_{max}\left(\frac{t_1}{2}\right) + \dot{a}_{max}\left(\frac{t_1}{2}\right) = \dot{a}_{max}\left(\frac{t_1}{2}\right)$$

Si on isole t_1:

$$t_1 = \frac{2a_{max}}{\dot{a}_{max}}$$

En remplaçant t_1 dans Éq. 2.5 on obtient:

$$\ddot{a}_d = \frac{\dot{a}_{max}^{\ 2}}{a_{max}}$$

Après avoir calculé la seconde dérivée d'accélération, il faut vérifier si les paramètres limites seront atteints. Pour ce faire, la vitesse v_3 est calculée avec $t_3 = 2t_1$.

Si $v_3 > v_{max}$,

L'accélération maximum ne sera pas atteinte: $a_1 < a_{max}$
Le temps nécessaire pour atteindre la vitesse maximum (fin de la phase d'accélération) est:

$$t_3 = 4\left(\frac{v_{max}}{2\ddot{a}}\right)^{1/3}$$

avec

$$t_1 = \frac{t_3}{2}, \quad t_2 = t_1$$

sinon

L'accélération maximum est atteinte: $a_1 = a_{max}$
Le temps pour la fin d'accélération constante est:

$$t_2 = t_1 + \frac{v_{max} - \left(\frac{-\ddot{a}t_1^3}{8} + a_{max}t_1 - v_1\right)}{a_{max}}$$

avec

$$t_3 = t_2 + t_1$$

En plus de vérifier l'accélération, il faut vérifier si la vitesse maximum sera atteinte.

Si $p_3 > \frac{|depl|}{2}$,

La vitesse maximum ne pourra être atteinte: $v_3 < v_{max}$

La position p_3 est recalculée avec $t_3 = 2t_1$.

Si $p_3 > \frac{|depl|}{2}$ ou $a_1 < a_{max}$,

L'accélération maximum ne sera pas atteinte: $a_1 < a_{max}$

Le temps de fin de la phase d'accélération est:

$$t_3 = 4\left(\frac{|depl|}{8\ddot{a}}\right)^{1/4}$$

avec

$$t_1 = \frac{t_3}{2}, \quad t_2 = t_1$$

sinon

L'accélération maximum est atteinte: $a_1 = a_{max}$.

Le temps pour la fin d'accélération constante est:

$t_2 = t_1 + \Delta t$

avec Δt la valeur maximum de la solution de l'équation du seconde ordre:

$$0 = \frac{a_1 \Delta t^2}{2} + (a_1 t_1 + v_1)\Delta t + \left(\frac{-14\ddot{a}\left(t_1/2\right)^4}{24} + 2a_1\left(t_1/2\right)^2 + v_1 t_1 - \frac{|depl|}{2} + p_1\right)$$

et

$t_3 = t_2 + t_1$

En plus, des temps (t_1, t_2, t_3) utilisés pour vérifier les limites, il faut aussi assigner les temps de décélérations (t_4, t_5, t_6). À l'aide des équations de base (Éq. 2.4), la trajectoire est alors calculée morceau par morceau.

Si le déplacement demandé est négatif, il suffit de changer de signe du résultat final de la génération de la trajectoire: $p(t) = -p(t)$, $v(t) = -v(t)$, $a(t) = -a(t)$.

2.4.1.3 Trajectoires avec dérivée d'accélération échelon

La fonction de génération de trajectoires avec dérivée d'accélération échelon est semblable à la fonction avec dérivée d'accélération triangulaire décrite à la section précédente. Les tests à effectuer pour vérifier les limites sont les mêmes. Seules les équations pour

déterminer les temps sont différentes. La Figure 2.4 donne un exemple de trajectoire avec dérivée d'accélération échelon.

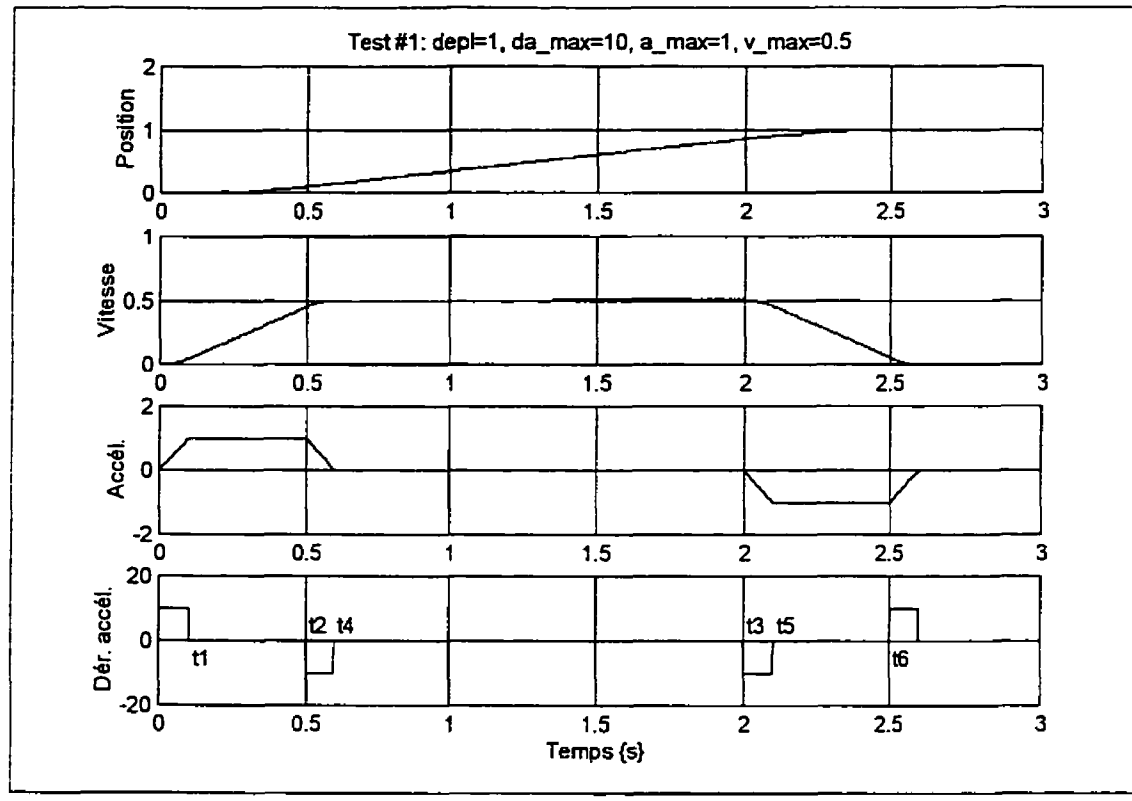


Figure 2.4: Exemple de trajectoire avec dérivée d'accélération échelon

La trajectoire avec dérivée d'accélération échelon est calculée par segment. Sachant que la dérivée d'accélération est constante pendant un segment, elle peut prendre alors une des trois valeurs possibles:

$$\dot{a}(t) = c = \left\{ \dot{a}_{\max} \quad ou \quad 0 \quad ou \quad -\dot{a}_{\max} \right\}$$

En utilisant comme conditions initiales:

$$p_i = p(t_i), v_i = v(t_i), a_i = a(t_i)$$

On obtient alors les équations de base servant à calculer les trajectoires en fonction du temps:

Éq. 2.6

$$a(t) = c\Delta t + a_i$$

$$v(t) = \frac{c\Delta t^2}{2} + a_i \Delta t + v_i$$

$$p(t) = \frac{c\Delta t^3}{6} + \frac{a_i \Delta t^2}{2} + v_i \Delta t + p_i$$

avec:

$$\Delta t = t - t_i$$

Le temps de début d'accélération constant $(a_1 = a_{max})$est:

$$t_1 = \frac{a_{max}}{\dot{a}_{max}}$$

Pour vérifier si la vitesse maximum sera atteinte, il faut calculer la vitesse v_3 avec $t_3 = 2t_1$.

Si $v_3 > v_{max}$,

L'accélération maximum ne sera pas atteinte: $a_1 < a_{max}$
Le temps nécessaire pour atteindre la vitesse maximum (fin de la phase d'accélération) est:

$$t_3 = 2\sqrt{\frac{v_{max}}{\dot{a}_{max}}}$$

avec

$$t_1 = \frac{t_3}{2}, \quad t_2 = t_1$$

sinon

L'accélération maximum est atteinte: $a_1 = a_{max}$
Le temps pour la fin d'accélération constante est:

$$t_2 = t_1 + 2\frac{\frac{v_{max}}{2} - v_1}{a_{max}}$$

avec

$$t_3 = t_2 + t_1$$

En plus de vérifier l'accélération, il faut vérifier si la vitesse maximum sera atteinte.

Si $p_3 > \frac{|depl|}{2}$,

La vitesse maximum ne pourra être atteinte: $v_3 < v_{max}$
La position p_3 est recalculée avec $t_3 = 2t_1$.
Si $p_3 > \frac{|depl|}{2}$ ou $a_1 < a_{max}$,
L'accélération maximum ne sera pas atteinte: $a_1 < a_{max}$
Le temps de fin de la phase d'accélération est:

$$t_3 = 2\left(\frac{|depl|}{2\dot{a}}\right)^{1/3}$$

avec

$$t_1 = \frac{t_3}{2}, \quad t_2 = t_1$$

sinon
L'accélération maximum est atteinte: $a_1 = a_{max}$.
Le temps pour la fin d'accélération constante est:
$t_2 = t_1 + \Delta t$
avec Δt la valeur maximum de la solution de l'équation du seconde ordre:

$$0 = \frac{a_1 \Delta t^2}{2} + v_1 \Delta t + \left(\frac{-\dot{a} t_1^3}{6} + \frac{a_1 t_1^2}{2} - \frac{|depl|}{2} + p_1\right)$$

et
$t_3 = t_2 + t_1$

En plus, des temps (t_1, t_2, t_3) utilisés pour vérifier les limites, il faut aussi assigner les temps de décélérations (t_4, t_5, t_6). À l'aide des équations de base (Éq. 2.6), la trajectoire est alors calculée morceau par morceau.

Si le déplacement demandé est négatif, il suffit de changer de signe du résultat final de la génération de la trajectoire: $p(t) = -p(t), \quad v(t) = -v(t), \quad a(t) = -a(t)$.

2.4.1.4 Trajectoires avec un temps d'accélération défini et la dérivée d'accélération sans discontinuité

La Figure 2.5 donne un exemple de trajectoire avec un temps d'accélération défini et la dérivée d'accélération sans discontinuité.

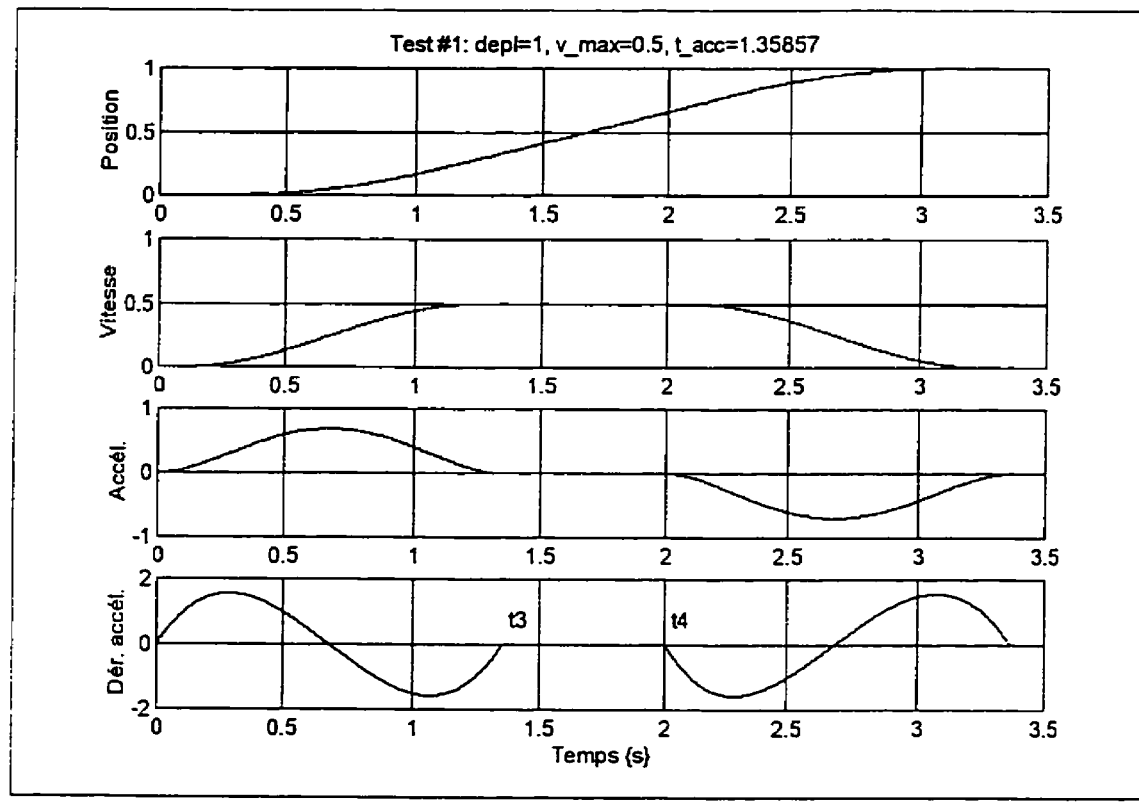


Figure 2.5: Exemple de trajectoire avec un temps d'accélération défini et la dérivée d'accélération sans discontinuité

La trajectoire est calculée à partir d'un polynôme d'ordre 7. Sachant les conditions initiales et finales on obtient alors les coefficients du polynôme:

Éq. 2.7

$$p(t) = -|depl|\left(\frac{20t^7}{t_f^7} + \frac{-70t^6}{t_f^6} + \frac{84t^5}{t_f^5} + \frac{-35t^4}{t_f^4}\right)$$

$$v(t) = -|depl|\left(\frac{140t^6}{t_f^7} + \frac{-420t^5}{t_f^6} + \frac{420t^4}{t_f^5} + \frac{-140t^3}{t_f^4}\right)$$

$$a(t) = -|depl|\left(\frac{840t^5}{t_f^7} + \frac{-2100t^4}{t_f^6} + \frac{1680t^3}{t_f^5} + \frac{-420t^2}{t_f^4}\right)$$

avec:

$$t_f = 2t_{acc}$$

Avant de calculer la trajectoire, il faut vérifier si la vitesse maximum sera atteinte. Pour ce faire, on calcule la vitesse à la fin de l'accélération $v(t_{acc})$.

Si $v(t_{acc}) \leq v_{max}$,

La vitesse maximum n'est pas atteinte. La trajectoire est alors calculée à l'aide des équations précédantes (Éq. 2.7)

sinon

La vitesse maximum est atteinte. Un polynôme d'ordre 6 est utilisé pour calculer un premier segment de trajectoire fin de l'accélération:

Éq. 2.8

$$p(t) = v_{max}\left(\frac{t^6}{t_{acc}^5} + \frac{-3t^5}{t_{acc}^4} + \frac{\frac{5}{2}t^4}{t_{acc}^3}\right)$$

$$v(t) = v_{max}\left(\frac{6t^5}{t_{acc}^5} + \frac{-15t^4}{t_{acc}^4} + \frac{10t^3}{t_{acc}^3}\right)$$

$$a(t) = v_{max}\left(\frac{30t^4}{t_{acc}^5} + \frac{-60t^3}{t_{acc}^4} + \frac{30t^2}{t_{acc}^3}\right)$$

pour

$$0 \leq t \leq t_3, \quad avec \quad t_3 = t_{acc}$$

Le temps de fin de vitesse constante est:

$$t_4 = \frac{|depl| - 2p(t_3)}{v_{max}}$$

Le segment de trajectoire à vitesse constante est:

Éq. 2.9

$$p(t) = v_{max}\left(t - t_3\right) + p\left(t_3\right)$$
$$v(t) = v_{max}$$
$$a(t) = 0$$
pour
$$t_3 < t < t_4$$

Le dernier segment de la trajectoire est:

Éq. 2.10

$$p(t) = v_{max}\left(\frac{-tt^6}{t_{acc}^5} + \frac{3tt^5}{t_{acc}^4} + \frac{-\frac{5}{2}tt^4}{t_{acc}^3}\right) + v_{max}\,tt + p\left(t_4\right)$$
$$v(t) = v_{max}\left(\frac{-6tt^5}{t_{acc}^5} + \frac{15tt^4}{t_{acc}^4} + \frac{-10tt^3}{t_{acc}^3}\right) + v_{max}$$
$$a(t) = v_{max}\left(\frac{-30tt^4}{t_{acc}^5} + \frac{60tt^3}{t_{acc}^4} + \frac{-30tt^2}{t_{acc}^3}\right)$$
pour
$$t_4 \le t \le \left(t_4 + t_{acc}\right), \quad avec \quad tt = t - t_4$$

Si le déplacement demandé est négatif, il suffit de changer de signe du résultat final de la génération de la trajectoire: $p(t) = -p(t)$, $v(t) = -v(t)$, $a(t) = -a(t)$.

2.4.1.5 Trajectoires avec un temps d'accélération défini et la dérivée d'accélération discontinue

La Figure 2.6 donne un exemple de trajectoire avec un temps d'accélération défini et la dérivée d'accélération discontinue.

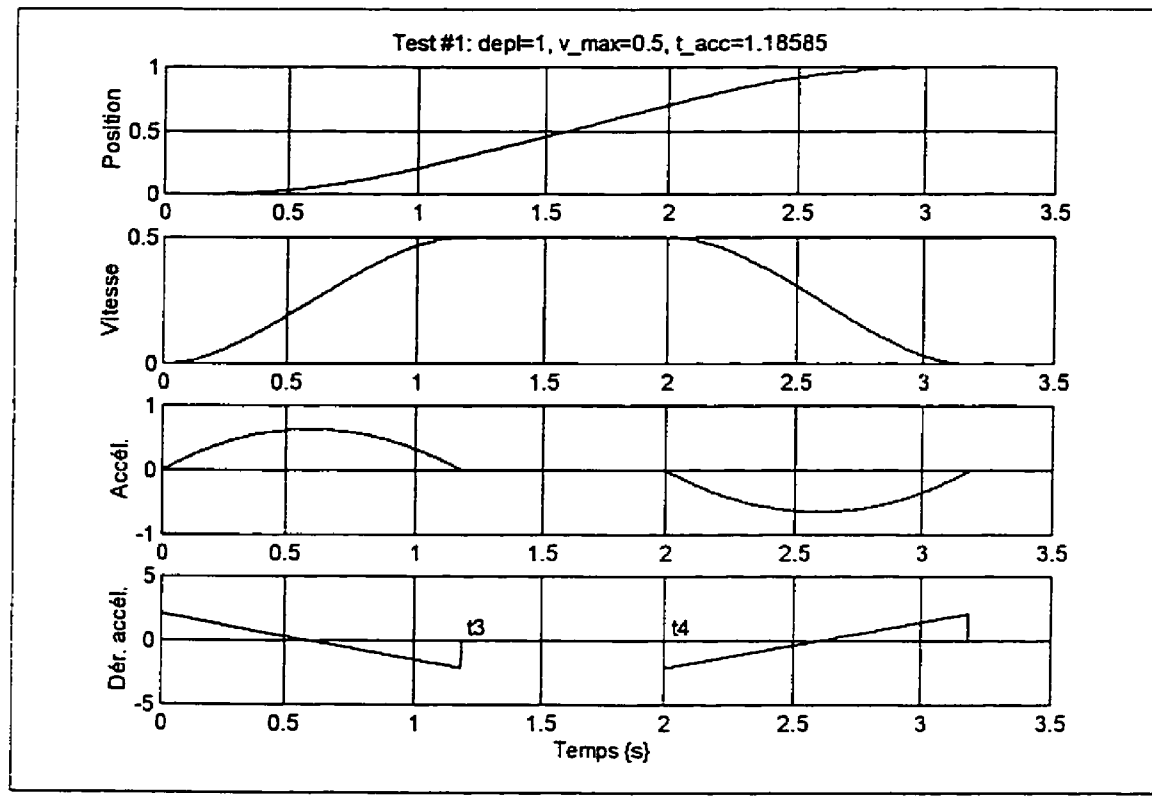


Figure 2.6: Exemple de trajectoire avec un temps d'accélération défini et la dérivée d'accélération discontinue

La trajectoire est calculée à partir d'un polynôme d'ordre 5. Sachant les conditions initiales et finales on obtient alors les coefficients du polynôme:

Éq. 2.11

$$p(t) = -|depl|\left(\frac{-6t^5}{t_f^5} + \frac{15t^4}{t_f^4} + \frac{-10t^3}{t_f^3}\right)$$

$$v(t) = -|depl|\left(\frac{-30t^4}{t_f^5} + \frac{60t^3}{t_f^4} + \frac{-30t^2}{t_f^3}\right)$$

$$a(t) = -|depl|\left(\frac{-120t^3}{t_f^5} + \frac{180t^2}{t_f^4} + \frac{-60t}{t_f^3}\right)$$

avec:

$$t_f = 2t_{acc}$$

Avant de calculer la trajectoire, il faut vérifier si la vitesse maximum sera atteinte. Pour ce faire, on calcule la vitesse à la fin de l'accélération $v(t_{acc})$.

Si $v(t_{acc}) \leq v_{max}$,

La vitesse maximum n'est pas atteinte. La trajectoire est alors calculée à l'aide des équations précédantes (Éq. 2.11)

sinon

La vitesse maximum est atteinte. Un polynôme d'ordre 4 est utilisé pour calculer un premier segment de trajectoire fin de l'accélération:

Éq. 2.12

$$p(t) = v_{max}\left(\frac{-\frac{1}{2}t^4}{t_{acc}^3} + \frac{t^3}{t_{acc}^2}\right)$$

$$v(t) = v_{max}\left(\frac{-2t^3}{t_{acc}^3} + \frac{3t^2}{t_{acc}^2}\right)$$

$$a(t) = v_{max}\left(\frac{-6t^2}{t_{acc}^3} + \frac{6t}{t_{acc}^2}\right)$$

pour

$$0 \leq t \leq t_3, \quad avec \quad t_3 = t_{acc}$$

Le temps de fin de vitesse constante est:

$$t_4 = \frac{|depl| - 2p(t_3)}{v_{max}}$$

Le segment de trajectoire à vitesse constante est:

Éq. 2.13

$$p(t) = v_{max}(t - t_3) + p(t_3)$$

$$v(t) = v_{max}$$

$$a(t) = 0$$

pour

$$t_3 < t < t_4$$

Le dernier segment de la trajectoire est:

Éq. 2.14

$$p(t) = -v_{\max}\left(\frac{-\frac{1}{2}tt^4}{t_{acc}^3} + \frac{tt^3}{t_{acc}^2}\right) + v_{\max}tt + p(t_4)$$

$$v(t) = -v_{\max}\left(\frac{-2tt^3}{t_{acc}^3} + \frac{3tt^2}{t_{acc}^2}\right) + v_{\max}$$

$$a(t) = -v_{\max}\left(\frac{-6tt^2}{t_{acc}^3} + \frac{6tt}{t_{acc}^2}\right)$$

pour

$$t_3 \le t \le (t_4 + t_{acc}), \quad avec \quad tt = t - t_4$$

Si le déplacement demandé est négatif, il suffit de changer de signe du résultat final de la génération de la trajectoire: $p(t) = -p(t)$, $v(t) = -v(t)$, $a(t) = -a(t)$.

2.4.2 Transformation en coordonnées rigides

La transformation en coordonnées rigides s'effectue par calculs de cinématique inverse. Les calculs, décrit dans cette section, sont directement reliés au robot utilisé (robot planaire). D'autres méthodes plus générales pourraient être implantées lors de futur travaux de recherches. Pour un robot planaire à trois articulations, une solution géométrique est retenue telle qu'illustrée à la figure suivante.

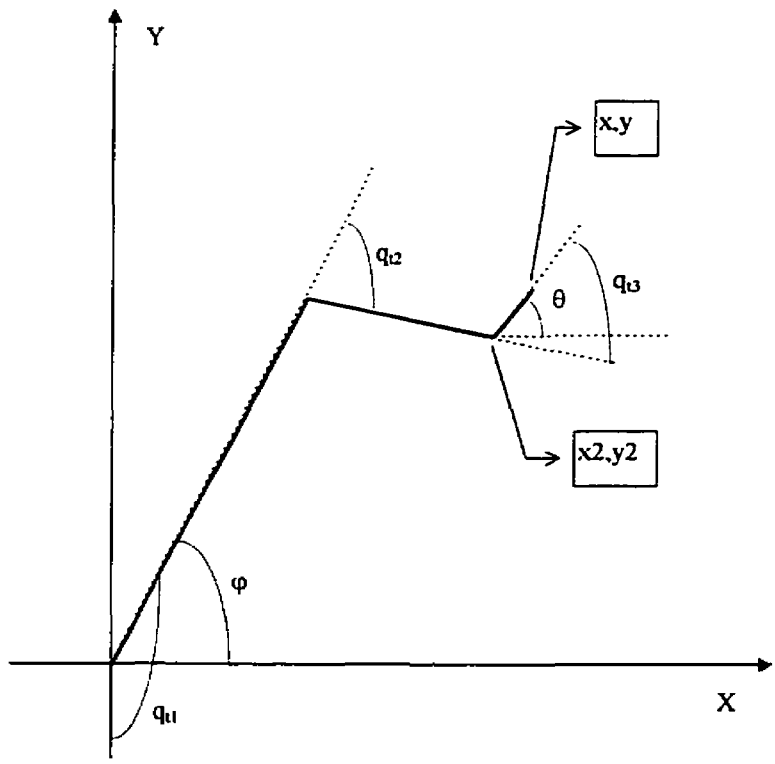


Figure 2.7: Cinématique inverse d'un robot planaire à trois articulations

avec

- L_i: la distance entre l'articulation i et l'articulation $i+1$ (ou l'extrémité du robot)
- x, y, θ : la position et l'orientation de l'extrémité du robot;
- x_2, y_2: la position de l'extrémité du deuxième segment;
- φ : angle du premier segment par rapport à l'axe X;
- q_{t_i}: angle de l'articulation i .

Pour obtenir la solution de cinématique inverse, il faut d'abord calculer la position de l'extrémité du deuxième segment: $x_2 = x - L_3 \cos(\theta), \quad y_2 = y - L_3 \sin(\theta)$.

Ensuite, il faut s'assurer que la position et l'orientation demandées sont valides (dans l'espace de travail du robot). Si $\sqrt{x_2^2+y_2^2} > (L_1+L_2)$ le point est trop près ou si $\left|\frac{x_2^2+y_2^2-L_1^2-L_2^2}{2L_1L_2}\right| > 1$ le point est trop près.

Les segments #1 et #2 forment deux cotés d'un triangle, par la loi des cosinus on obtient: $q_{t_2} = \cos^{-1}\left(\frac{x_2^2+y_2^2-L_1^2-L_2^2}{2L_1L_2}\right)$. L'angle calculé est positif: $0 \le q_{t_2} \le \pi$. Si $q_{t_2} \neq 0$, alors il y a deux solutions. Il faut changer de signe pour obtenir la deuxième solution.

Ayant calculé l'angle de la deuxième articulation, l'angle du premier segment du robot est:

$$\varphi = \tan^{-1}(y_2, x_2) - \tan^{-1}\left(L_2 \sin(q_{t_2}), L_1 + L_2 \cos(q_{t_2})\right)$$

L'angle de la troisième articulation est: $q_{t_3} = \theta - \varphi - q_{t_2}$. Si l'on veut que tous les angles initiaux des articulations soient nuls lorsque robot planaire vertical est au repos (commande nulle), l'angle de la première articulation est alors: $q_{t_1} = \varphi + \pi/2$

2.4.3 Calcul de la commande anticipative

La commande anticipative est fonction de la trajectoire désirée des articulations.

Le calcul de la commande s'effectue en plusieurs étapes:

1) calcul de la commande pour un système rigide;
2) estimation des coordonnées élastiques et des coordonnées de flexibilité et correction de positions des articulations;

3) calcul de la commande pour le système à membrures flexibles et élasticité aux articulations;
4) compensation de la friction ajoutée à la commande.

Certaines étapes sont omises selon la disponibilité des coordonnées du modèle utilisé. Si le modèle n'a que des coordonnées rigides, alors la deuxième et la troisième étape sont omises.

2.4.3.1 Calcul de la commande pour un système rigide

La première étape de calcul consiste à déterminer la commande du système $(\mathbf{u}_d)$ en négligeant la flexibilité des membrures et l'élasticité des articulations. Le modèle non-linéaire est utilisé pour calculer cette commande qui est fonction de la trajectoire désirée des articulations $(\mathbf{q}_t, \dot{\mathbf{q}}_t, \ddot{\mathbf{q}}_t)$.

$$\begin{bmatrix} \mathbf{u}_d \\ - \\ - \end{bmatrix} = \mathbf{M}_{nl_d}\left(\mathbf{q}_{d_t}\right)\ddot{\mathbf{q}}_{d_t} + \Gamma_{nl_d}\left(\mathbf{q}_{d_t}, \dot{\mathbf{q}}_{d_t}, \mathbf{u}\right)$$

avec: $\mathbf{q}_{d_t} = \begin{bmatrix} \mathbf{q}_t \\ \mathbf{0}_{n_e} \\ \mathbf{0}_{n_f} \end{bmatrix}$, $\dot{\mathbf{q}}_{d_t} = \begin{bmatrix} \dot{\mathbf{q}}_t \\ \mathbf{0}_{n_e} \\ \mathbf{0}_{n_f} \end{bmatrix}$, $\ddot{\mathbf{q}}_{d_t} = \begin{bmatrix} \ddot{\mathbf{q}}_t \\ \mathbf{0}_{n_e} \\ \mathbf{0}_{n_f} \end{bmatrix}$, $\mathbf{u} = \mathbf{0}_{n_u}$

2.4.3.2 Estimation des coordonnées élastiques et des coordonnées flexibles et correction des coordonnées rigides désirées

Pour estimer les coordonnées élastiques et flexibles, il faut obtenir la matrice de rigidité $(\mathbf{K}_t)$du modèle linéarisé et les matrices $(\mathbf{M}_{nl}, \Gamma_{nl})$ du modèle non-linéaire en utilisant la commande $(\mathbf{u}_d)$ pour le système rigide calculée à l'étape précédante.

En utilisant les vecteurs d'écart par rapport au point de linéarisation:

$$\Delta\ddot{\mathbf{q}} = 0_n, \quad \Delta\dot{\mathbf{q}} = 0_n, \quad ,\Delta\mathbf{q} = \begin{bmatrix} \mathbf{0}_{n_r} \\ \mathbf{q}_{fe} \end{bmatrix}, \quad \mathbf{q}_{fe} = \begin{bmatrix} \mathbf{q}_e \\ \mathbf{q}_f \end{bmatrix}$$

les coordonnées élastiques et flexibles sont alors estimées par l'équation:

$$\mathbf{q}_{fe} = \mathbf{K}_{l_{fe}}^{-1}\,\mathbf{h}_{fe}$$

avec

$$\mathbf{h} = -\left(\mathbf{M}_{nl}\left(\mathbf{q}_{d_t}\right)\ddot{\mathbf{q}}_{d_t} + \Gamma_{nl}\left(\mathbf{q}_{d_t}, \dot{\mathbf{q}}_{d_t}, \mathbf{u}_d\right)\right) = \begin{bmatrix} - \\ \mathbf{h}_{fe} \end{bmatrix} \quad , \quad \mathbf{K}_l\left(\mathbf{q}_{d_t}, \dot{\mathbf{q}}_{d_t}\right) = \begin{bmatrix} - & - \\ - & \mathbf{K}_{l_{fe}} \end{bmatrix}$$

La correction des coordonnées rigides dépend du type de robot (planaire, sphérique, etc.) et du nombre d'articulations. Le robot utilisé dans ce mémoire est un robot planaire à trois articulations rotoïdes. En utilisant les coordonnées rigides et les coordonnées flexibles, on peut déterminer les matrices de transformations homogènes entre les différents corps du robot. La correction de chacune des coordonnées rigides est calculée de façon récursive en commençant par la première. Pour chaque articulation, il faut déterminer la position (dans le plan XY) de la prochaine articulation (ou de l'extrémité du robot) par rapport au référentiel associé à la coordonnée rigide à corriger. La position désirée $(\mathbf{p}_r)$ est calculée en n'utilisant que les coordonnées rigides tandis que la position sans corrections $(\mathbf{p}_{fe})$ est obtenue en utilisant le vecteur de coordonnées élastiques et flexibles et les nouvelles coordonnées rigides désirées (avec corrections). La correction d'une coordonnée rigide est la valeur de l'angle qui fait en sorte de replacer la prochaine articulation (ou l'extrémité) à sa position désirée dans le plan XY.

$$q_{cor_i} = tg^{-1}\left(y_{fe}, x_{fe}\right) - tg^{-1}\left(y_r, x_r\right)$$

avec $\mathbf{p}_r = \begin{bmatrix} x_r \\ y_r \end{bmatrix}, \quad \mathbf{p}_{fe} = \begin{bmatrix} x_{fe} \\ y_{fe} \end{bmatrix}$

Le nouveau vecteur de coordonnée désirée est:

$$\mathbf{q}_d = \begin{bmatrix} \mathbf{q}_t \\ \mathbf{0}_{n_{fe}} \end{bmatrix} + \begin{bmatrix} \mathbf{q}_{r_{cor}} \\ \mathbf{q}_{fe} \end{bmatrix}$$

Les prochaines figures présentent les étapes de la correction. Les formes triangulaires représentent les articulations. On peut aussi y voir les référentiels (axe X en pointillé, axe Y en tiret point) associés aux différents corps du système.

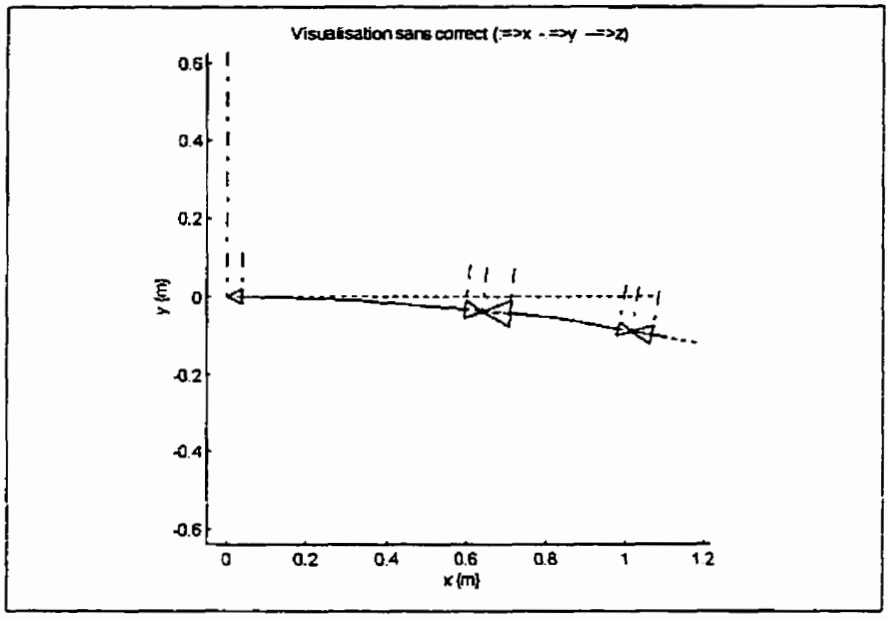


Figure 2.8: Visualisation du système sans correction aux articulations

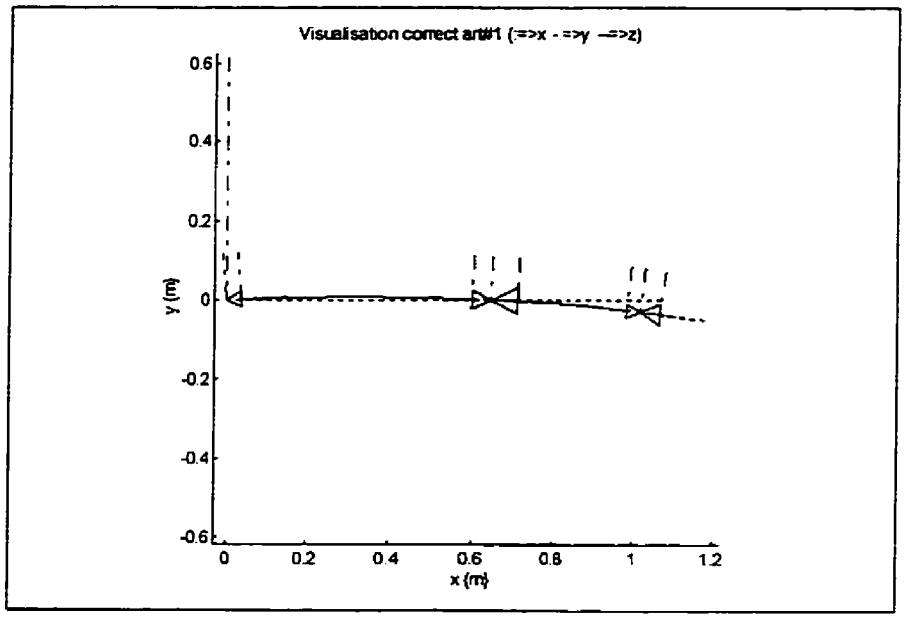


Figure 2.9: Visualisation de la correction sur la première articulation

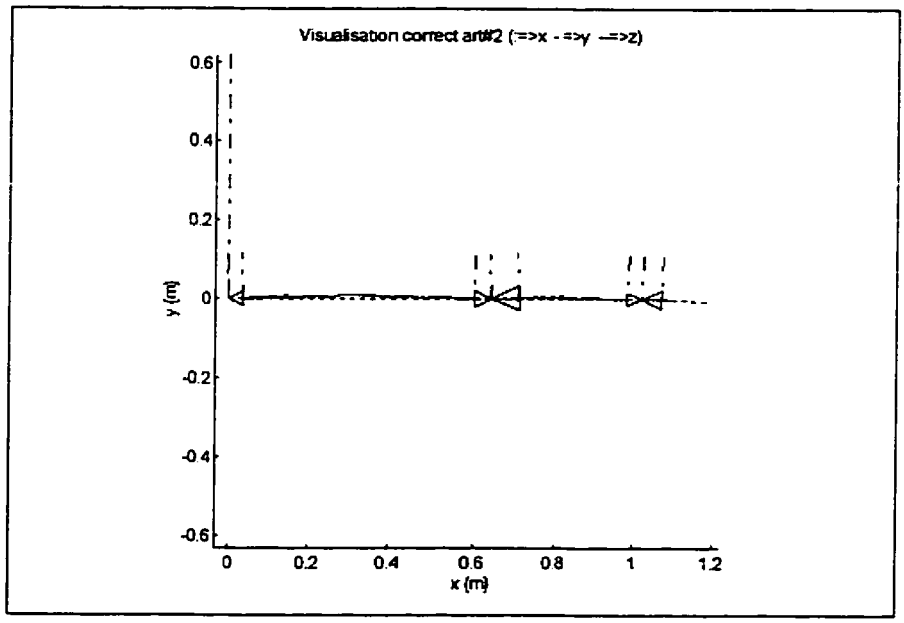


Figure 2.10: Visualisation de la correction sur la deuxième articulation

Les deux prochaines figures présentent la troisième articulation avant et après la correction.

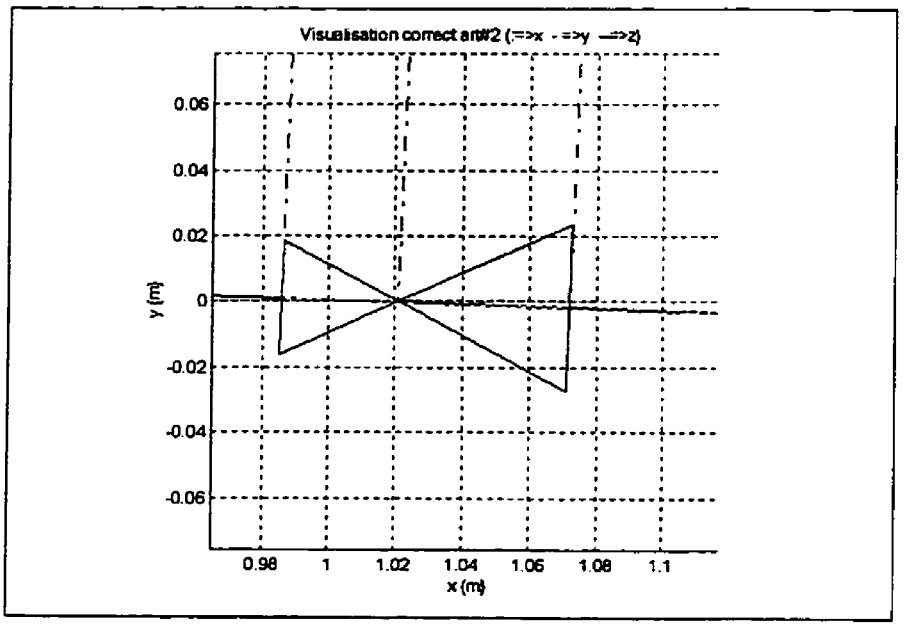


Figure 2.11: Visualisation avant la correction sur la troisième articulation

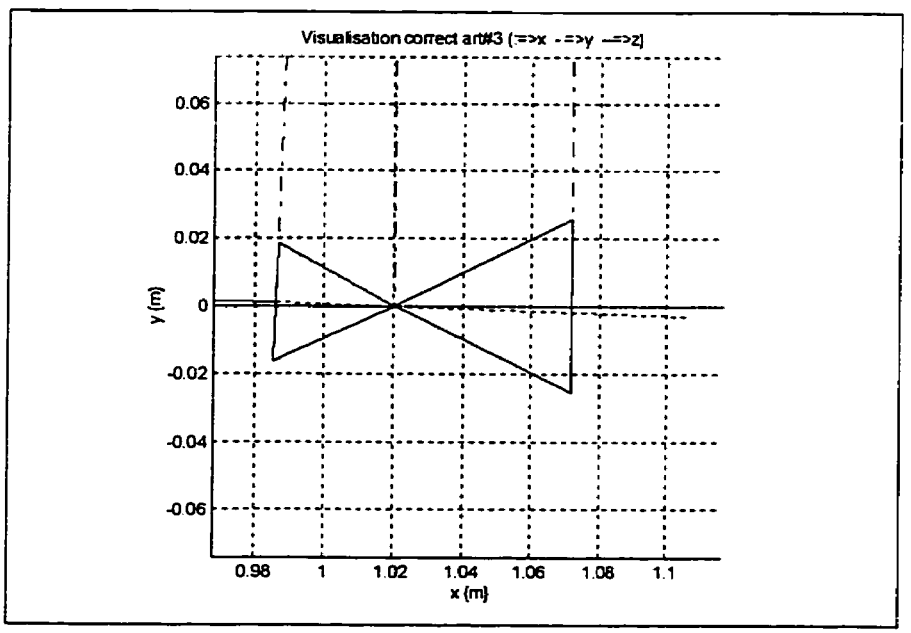


Figure 2.12: Visualisation après la correction sur la troisième articulation

2.4.3.3 Calcul de la commande pour le système complet, avec les coordonnées flexibles et élastiques estimées

Si le modèle contient des coordonnées flexibles ou des coordonnées élastiques, alors la commande $(\mathbf{u}_d)$ est recalculée avec le nouveau vecteur de coordonnées désirées $(\mathbf{q}_d)$.

$$\begin{bmatrix} \mathbf{u}_d \\ _ \\ _ \end{bmatrix} = \mathbf{M}_{nl_d}(\mathbf{q}_d)\ddot{\mathbf{q}}_d + \Gamma_{nl_d}(\mathbf{q}_d, \dot{\mathbf{q}}_d, \mathbf{u}), \quad \text{avec:} \quad \mathbf{u} = \mathbf{0}_{n_u}$$

2.4.3.4 Ajout de la friction à la commande

Tel que décrit précédemment (Éq. 2.3), la friction est fonction de la vitesse $\left(\mathbf{fr}\left(\dot{\mathbf{q}}_{r_d}\right)\right)$ et est calculée à l'aide d'un polynôme. Pour chaque articulation, il faut rechercher les points (k_i) de la trajectoire désirée où l'articulation (i) commence à se déplacer $\left(\dot{q}_{r_d}(k_i - 1) = 0, \quad \dot{q}_{r_d}(k_i) \neq 0\right)$. À chacun de ces points, la friction est alors égale à la friction statique de l'articulation: $\left(fr_i(k_i) = fr_{stat_i}\ \mathrm{sgn}\left(\dot{q}_{r_d}(k_i)\right)\right)$.

Le nouveau vecteur de commande anticipative est:

$\mathbf{u}_d = \mathbf{u}_d + \mathbf{fr}$

2.5 Simulation d'un robot avec friction aux articulations

Pour se rapprocher du comportement réel d'un robot manipulateur, il faut simuler le système en incluant au modèle l'effet de la friction aux articulations $(\mathbf{fr})$.

Éq. 2.15

$$\boxed{\begin{aligned}&\mathbf{M}_{nl}(\mathbf{q})\ddot{\mathbf{q}}+\Gamma_{nl}(\mathbf{q},\dot{\mathbf{q}},\mathbf{u})+\begin{bmatrix}\mathbf{fr}(\dot{\mathbf{q}}_r)\\ \mathbf{0}_{n-n_u}\end{bmatrix}=\mathbf{0}_n\\ &\Gamma_{nl}=\mathbf{v}(\mathbf{q},\dot{\mathbf{q}})+\mathbf{g}(\mathbf{q})-\begin{bmatrix}\mathbf{u}\\ \mathbf{0}_{n-n_u}\end{bmatrix}\end{aligned}}$$

Pour simuler le système, il faut calculer le vecteur d'accélération:

$$\ddot{\mathbf{q}}=-\mathbf{M}_{nl}^{-1}\left(\Gamma_{nl}-\begin{bmatrix}\mathbf{fr}\\ \mathbf{0}_{n-n_r}\end{bmatrix}\right)$$

Ensuite les vecteurs de vitesses et de positions sont obtenus par intégration numérique. Pour calculer le vecteur d'accélérations, il faut déterminer le vecteur de frictions **fr**.

2.5.1 Calculs du vecteur d'accélération et du vecteur de friction

Pour déterminer le vecteur de friction il faut d'abord vérifier quel est le type de friction qui agit sur les actionneurs. Si un actionneur est en mouvement, alors la friction dynamique est facilement calculable, elle est fonction de la vitesse. Par contre, si un actionneur est immobile, alors la friction n'est pas directement calculable à partir d'une fonction. La friction est alors égale au couple effectif (force) à l'actionneur mais ne dépassant pas la valeur maximum de friction statique pour cet actionneur. Puisque les articulations d'un robot manipulateur sont inter reliées, ceci engendre que chacun des actionneurs est couplé aux autres actionneurs. Par conséquent, pour déterminer la friction il faut résoudre le système d'équations suivant:

$$\mathbf{M}_{nl}\ddot{\mathbf{q}}+\Gamma_{nl}-\begin{bmatrix}\mathbf{fr}\\ \mathbf{0}_{n-n_r}\end{bmatrix}=\mathbf{0}_n$$

Ce système d'équations a trop d'inconnues. Pour le résoudre, on suppose que les actionneurs dont la friction statique est inconnue resteront immobiles, c'est-à-dire que leurs accélérations sont nulles. Connaissant le vecteur de couples et de forces du système

sans friction (Γ_{nl}) et en utilisant les valeurs de friction connues (friction dynamique ou friction statique maximum), les accélérations et les valeurs de friction inconnues peuvent être calculées. Les équations qui suivent, décrivent un exemple du système d'équations à résoudre pour un manipulateur ayant 3 degrés de liberté (à membrures rigides et sans élasticité) dont la friction (statique) du deuxième actionneur est inconnue.

$$\begin{bmatrix} M_{11} & X & M_{13} \\ M_{21} & X & M_{23} \\ M_{31} & X & M_{33} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ 0 \\ \ddot{q}_3 \end{bmatrix} + \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \Gamma_3 \end{bmatrix} + \begin{bmatrix} fr_1 \\ fr_2 \\ fr_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Si on réunit les variables inconnues (accélérations et frictions statiques) on obtient:

$$\begin{bmatrix} M_{11} & 0 & M_{13} \\ M_{21} & 1 & M_{23} \\ M_{31} & 0 & M_{33} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ fr_2 \\ \ddot{q}_3 \end{bmatrix} = - \begin{bmatrix} \Gamma_1 + fr_1 \\ \Gamma_2 \\ \Gamma_3 + fr_3 \end{bmatrix}$$

La solution du système d'équations est:

$$\begin{bmatrix} \ddot{q}_1 \\ fr_2 \\ \ddot{q}_3 \end{bmatrix} = - \begin{bmatrix} M_{11} & 0 & M_{13} \\ M_{21} & 1 & M_{23} \\ M_{31} & 0 & M_{33} \end{bmatrix}^{-1} \begin{bmatrix} \Gamma_1 + fr_1 \\ \Gamma_2 \\ \Gamma_3 + fr_3 \end{bmatrix}$$

Après avoir calculé la friction inconnue, il faut vérifier la validité de l'hypothèse de départ. Si la friction calculée est inférieure à la friction statique, alors l'hypothèse est valide, l'actionneur reste immobile. Par contre, si l'hypothèse de départ est invalide, l'actionneur accélère. La friction de l'actionneur est alors fixée comme étant égale à la friction statique maximum et il faudra recalculer l'effet de ce changement.

Une zone morte (vitesse minimum ou vitesse d'arrêt de l'actionneur) permet de détecter si la vitesse passe par zéro. Si un changement de signe de la vitesse d'un actionneur est détecté, la friction pour cet actionneur est alors inconnue. Si la friction calculée est

inférieure à la friction statique, on ajoute un terme d'amortissement numérique qui fera tendre la vitesse de l'actionneur vers zéro.

2.5.2 Transformation en coordonnées des capteurs de flexibilité (courbure)

Le système expérimental utilise des capteurs (jauges de déformation) sur les membrures dont le signal est proportionnel à la courbure (pour flexion) ou à la pente (pour torsion). Pour simuler le comportement du système, il faut convertir les coordonnées de flexibilité simulées en coordonnées de capteurs (courbure) à l'aide d'une matrice de transformation:

$$\mathbf{q}_c = \mathbf{T}_{q_c}^{q_f} \mathbf{q}_f$$

Les éléments de cette matrice $\{n_c \times n_f\}$ sont obtenues en évaluant les fonctions de formes aux points de mesure (normalisées par rapport à la longueur de la membrure) des capteurs de flexibilité. Pour les capteurs de flexibilité servant à détecter la flexion, on utilise la dérivée seconde des fonctions de forme, tandis que pour les capteurs de torsion, on utilise la dérivée première des fonctions de forme.

2.5.3 Contrôleur PD avec ajout d'une commande anticipative

$$u_i = k_{p_i}\left(q_{s_{\eta}} - q_{d_{\eta}}\right) + k_{d_i}\left(\dot{q}_{s_{\eta}} - \dot{q}_{d_{\eta}}\right) + u_{d_i}$$

avec

- $\mathbf{k}_p$: vecteur $\{n_u \times 1\}$ de gains proportionnels;
- $\mathbf{k}_d$: vecteur $\{n_u \times 1\}$ de gains dérivés.
- $\mathbf{u}_d$: vecteur $\{n_u \times 1\}$ de commande anticipative.

2.5.4 Contrôleur PD pour les capteurs de flexibilité

Dans le but diminuer les oscillations des membrures lorsqu'on atteint le régime permanent on ajoute une commande en fonction des capteurs de flexibilité.

$$u_i = \left(\mathbf{k}_{p_i} \cdot \left(\mathbf{q}_{s_q} - \mathbf{q}_{d_q}\right) + \mathbf{k}_{d_i} \cdot \left(\dot{\mathbf{q}}_{s_q} - \dot{\mathbf{q}}_{d_q}\right)\right)$$

avec

- $\mathbf{k}_{p_c}$: vecteur $\{n_c \times 1\}$ de gains proportionnels pour les capteurs de flexibilité;
- $\mathbf{k}_{d_c}$: vecteur $\{n_c \times 1\}$ de gains dérivés pour les capteurs de flexibilité.

2.6 Conclusion

Dans ce chapitre, nous avons présenté les différents modèles utilisé par le logiciel ROVASIMEX, les algorithmes de génération de trajectoires et la stratégie de commande anticipative d'un robot à membrures flexibles.

3 CONCEPTION DU LOGICIEL ROVASIMEX

Le logiciel **ROVASIMEX** pour "**RO**botique, **VA**lidation, **SIM**ulation, **EX**périmentation" permet de modéliser, simuler et réaliser des essais expérimentaux sur un robot. Ce chapitre est consacré à la conception et à la description de ce logiciel. Dans un premier temps, on rappelle les objectifs qui ont prévalus lors de la conception du logiciel et les principales caractéristiques qui ont été retenues. Par la suite, la structure du logiciel est décrite.

3.1 Objectifs

Puisque le logiciel sera utilisé dans un contexte de recherche, la **facilité d'implantation et de vérification** de différentes notions théoriques est un facteur important dans la conception du logiciel.

Le logiciel devra permettre de **tester rapidement** ces notions théoriques, i.e. implanter différents contrôleurs, vérifier différentes modélisations du système (avec ou sans membrure flexible, avec ou sans élasticité, etc.).

Puisque l'on veut comparer des résultats de simulation avec des résultats expérimentaux, il faut donc bien identifier les paramètres du système. Pour ce faire, on doit effectuer plusieurs **essais pour différentes configurations du système**.

Pour rendre le logiciel accessible à différents types d'utilisateurs, (chercheurs, étudiants 1er, 2ième, 3ième cycle) il serait avantageux que le logiciel soit **facile d'utilisation**.

3.2 Caractéristiques

À partir des objectifs qu'on a fixés, on peut maintenant établir les principales caractéristiques du logiciel.

La facilité d'implantation et de vérification de différentes notions théoriques fait en sorte que le logiciel est très versatile. Une **programmation modulaire** est donc de rigueur.

Les 7 principaux modules du logiciel ROVASIMEX sont:

1) [Graph] traitement des données;
2) [Config] configuration d'un système et génération de modèles symboliques;
3) [Initial] initialisation des paramètres et génération de modèles numériques;
4) [Traject] génération de trajectoires;
5) [Consignes] calcul des paramètres d'entrée d'un essai de simulation ou sur le système expérimental;
6) [Simul] simulation du système;
7) [Exper] essai expérimental.

La **présentation du logiciel sous forme de menus et fenêtres** augmente la facilité d'utilisation du logiciel. La Figure 3.1 présente la fenêtre principale du logiciel ROVASIMEX. Les 7 modules principaux apparaissent sur la ligne du haut.

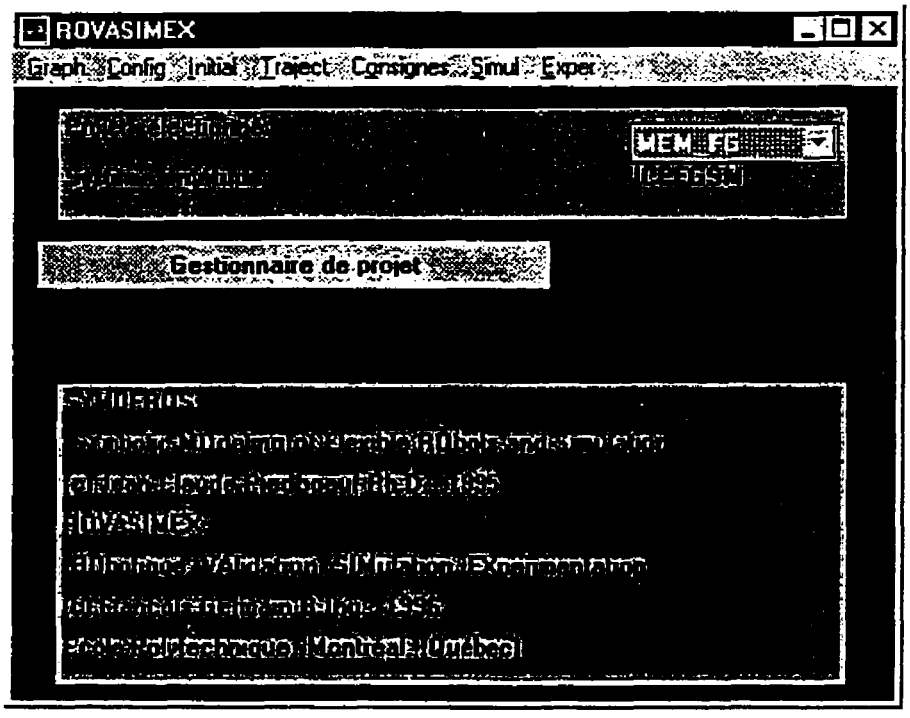


Figure 3.1 Fenêtre principale du logiciel ROVASIMEX

Pour diminuer le temps des tests et pour conserver l'intégrité des données, le **transfert d'informations** entre les différents modules du logiciel s'effectue **à l'aide de fichiers**. En plus de contenir les résultats (sorties) d'un module, les fichiers de données contiennent les noms des fichiers de données créés par les modules antérieurs.

Chacune des configurations du système est créée en sélectionnant les différentes composantes du système. Les **composantes du système sont regroupées selon leur type** (amplificateurs, actionneurs, articulations, capteurs, etc...). Lors de l'initialisation des paramètres numériques, on peut ajouter différentes charges à l'extrémité du système.

Puisque le temps pour générer les modèles est parfois assez long[1], la **génération des modèles numériques** des systèmes s'effectue **en deux étapes**. La première étape consiste à générer les modèles symboliques. À la seconde étape, on génère les modèles numériques en assignant les valeurs aux paramètres symboliques. Pour vérifier l'effet d'un paramètre, on peut le laisser libre lors de la génération d'un modèle numérique. Les valeurs par défauts des paramètres symboliques laissés libres sont alors proposées lors de l'utilisation du modèle.

3.2.1 Acquisitions de données et contrôle en temps réel

Pour facilement implanter différents types de contrôleurs, on **utilise un ordinateur** avec des cartes d'acquisitions plutôt que des micro-contrôleurs (ou DSP). Les micro-contrôleurs sont dédiés à un seul système expérimental, donc moins versatiles, par contre le temps d'exécution peut être plus rapide.

Pour s'assurer que l'acquisition de données s'effectue à période d'échantillonnage constante, un **signal de synchronisation est relié à une interruption matérielle de l'ordinateur**.

Les mesures de capteurs (nombres entiers) sont enregistrées dans un fichier binaire Matlab. La sauvegarde des données en nombres entiers a pour effet de réduire la dimension des fichiers de résultats d'essais expérimentaux et rend les données **indépendantes des facteurs de conversion**.

---

[1] Le temps de génération est de plusieurs minutes (60 à 120) pour modèles avec membrures flexibles et élasticité aux articulations.

3.3 Environnement informatique

Pour ce projet de recherche, un ordinateur IBM 486 (66 Mhz) est utilisé. Toutes les fonctions du logiciel ROVASIMEX sont écrites pour être exécutées avec les logiciels Matlab (version 4.2c) et Simulink (version 1.3). Les fonctions exécutables en temps réel sont écrites en langage "C" pour être ensuite compilées avec le logiciel Borland C (version 4.0) en format DOS (*.EXE). Les résultats obtenus par les différents modules du logiciel ROVASIMEX sont enregistrés dans des fichiers de données binaires de format Matlab. Les fonctions et les fichiers de données sont placés dans différents répertoires, selon qu'ils soient d'aspect général, en relation avec un système (ex. robot manipulateur) en particulier ou à un certain projet (utilisateur).

La modélisation d'un robot à membrures flexibles étant plus complexe que celle d'un robot à membrures rigides, on fait appel aux fonctions du logiciel SYMOFROS[2] qui génère différents modèles symboliques correspondant à une configuration du robot en étude. Par la suite, en utilisant un des modèles symboliques comme paramètre d'entrée, on génère des modèles numériques (routines écrites en langage "C") qui sont compilés (avec les logiciels Metaware High C/C++ V3.03 et Phar Lap TNT) pour donner une fonction (*.MEX) exécutable par le logiciel Matlab. Ces fonctions Matlab de modèles numériques sont utilisés pour calculer la commande anticipative, pour le traitement des données et lors des simulations avec le logiciel Simulink de Matlab. Le logiciel SYMOFROS implanté dans ROVASIMEX utilise la version 3.0 de Maple V.

La version 4.2 du logiciel Matlab utilise le système d'opération MS-Windows 3.1 ou Windows 95. En plus des libraires de fonctions pour le traitement mathématique (analyse, simulation), il y a une librairie de fonctions d'interface avec l'usagé. Ces fonctions

[2]SYMOFROS pour "SYmbolic MOdeling of Flexible RObots and Simulation".

permettent de créer facilement des menus et des fenêtres graphiques comprenant différents types d'éléments: bouton poussoirs, etc. . De plus, la version 4.2 du logiciel Matlab permet de tracer des graphiques à trois dimensions.

3.4 Structure du logiciel

Les prochaines sous-sections présentent la liste des répertoires des fonctions et des fichiers de données créés ou utilisés par chacun des modules du logiciel ROVASIMEX. La relation entre les fichiers de données est représenté sous forme de structure hiérarchique. Pour terminer, les noms par défaut des fichiers de données sont énumérés.

3.4.1 Descriptions des répertoires utilisés

Étant donnée le grand nombre[3] de fichiers utilisés par le logiciel, les fichiers sont regroupés dans différents répertoires. Le programme `chemins` (répertoire: `\ROVASIMEX`) défini les répertoires de base du logiciel. Ce programme est crée automatiquement lors de l'installation du logiciel.

Il y a cinq répertoires de base:

1) `\SRC_UTIL`, pour les fonctions utilitaires (Matlab et Maple);
2) `\RVSE_FCT`, pour les fonctions de bases (Matlab et Maple) du logiciel ROVASIMEX;
3) `\RVSE_CTR`, pour les fonctions de bases (langage C) pour l'acquisitions de données et le contrôle en temps réel.
4) `\SYSTEMS`, pour les sous répertoires des différents systèmes (robots);
5) `\RVSEPROJ`, pour les sous répertoires des différents projets (utilisateurs).

[3]Environ 520 fichiers: plus de 290 fichiers pour les fonctions Matlab et Maple du logiciel ROVASIMEX, près de 150 fichiers spécifiques au robot (paramètres et modèles symboliques), plus de 30 fichiers (en langage C) pour le contrôle en temps réel et près de 50 fichiers pour les fonctions utilitaires Matlab et Maple.

3.4.1.1 Répertoires des fonctions utilitaires: \SRC_UTIL

Pour simplifier et optimiser l'écriture des sous-programmes du logiciel, on utilise des fonctions utilitaires qui sont regroupées dans deux répertoires: \SRC_UTIL\MAT40 (pour Matlab version 4.0 et supérieur) et \SRC_UTIL\MAP30 (pour Maple version 3.0).

3.4.1.2 Répertoires des fonctions de bases: \RVSE_FCT

Le Tableau 3.1 indique le nom des sous répertoires des fonctions de base, le nom du logiciel avec lequel on l'utilise et ainsi qu'une courte description du contenu du répertoire.

Tableau 3.1: Contenu des répertoires des fonctions de bases: \RVSE_FCT

Répertoire	Logiciel	Contenu (type de programmes)
\CONFIG	Matlab	configuration d'un système
\INITIAL	Matlab	initialisation des paramètres d'un système
\GEN_MOD	Maple	génération des modèles symboliques et des modèles numériques
.\LIN_FORM	Maple	listes de linéarisation et fonctions de forme
\TRAJECT	Matlab	génération de trajectoires
\CONTROL	Matlab	calculs et assignations des paramètres des contrôleurs
\SIMUL	Matlab	simulation
\EXPER	Matlab	expérimentation
\TRT_DATA	Matlab	traitement des données

La Figure 3.2 présente sous forme d'arbre le lien entre les différents sous répertoires.

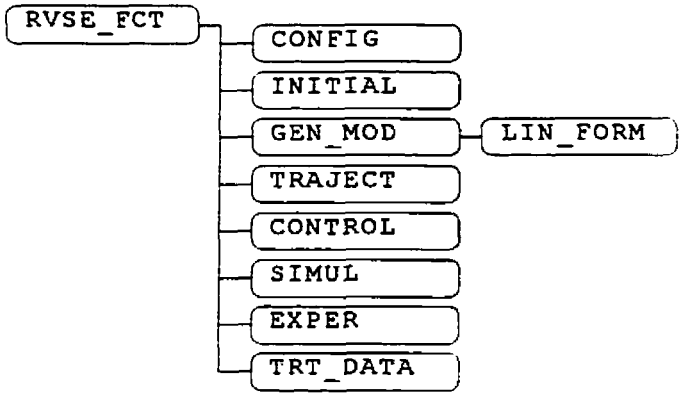


Figure 3.2: Arbre des répertoires contenant les fonctions de base (Matlab et Maple) du logiciel ROVASIMEX

3.4.1.3 Répertoires des fonctions pour l'acquisition et le contrôle en temps réel: \RVSE_CTR

Les fonctions de base pour créer les programmes d'acquisitions de données et de contrôle en temps réel sont rassemblées dans le répertoire de base \RVSE_CTR. Le Tableau 3.2 présente le contenue des répertoires et sous répertoires.

Tableau 3.2: Contenue des répertoires pour l'acquisition de données et le contrôle en temps réel: \RVSE_CTR

Répertoire	Contenu
\RVSE_CTR	fonctions de base du système
\RVSE_CTR\DRIVERS	fonctions pour les cartes d'interfaces
\RVSE_CTR\UTL_MLAB	fonctions utilitaires pour l'interface avec les fichiers Matlab
\RVSE_CTR\CONTROLS	fonctions pour le calcul de la commande (contrôleurs)

3.4.1.4 Répertoires des fonctions et des fichiers associés à un système: \SYSTEMS

Le logiciel pouvant être utilisé pour différents robots manipulateurs, les fichiers spécifiques à chacun des systèmes sont réunis dans le répertoire de base \SYSTEMS. Pour décrire les sous répertoires associés à un système en particulier, on utilise les variables du Tableau 3.3. À partir de ces variables, le Tableau 3.4 présente le contenu et une courte description des sous répertoires.

Tableau 3.3: Variables servant à décrire les sous répertoires associés à un système

Nom de variable	Définition
nom_system	nom d'un robot manipulateur
nom_mds	nom d'un fichier de modélisation symbolique du système

Tableau 3.4: Contenue et description des sous répertoires associés à un système: \SYSTEMS

Répertoire	Contenu et description
system_path	fonctions de base pour un robot =>\SYSTEMS*nom system*\
parts_path	fichiers de composantes du robot => *system path* PARTS\
mds_root_path	fichiers de définition des modèles symboliques => *system path* MOD_SYM\
opt_mds_path	options de modélisation symbolique => *mds root path* OPTIONS\
mod_sym_path	fichiers résultats de modélisation symbolique => *mds root path* nom_mds\

Le Tableau 3.5 présente un exemple des noms de répertoires associés au robot de ce projet de recherche (JCPFGSM) avec comme fichier de modélisation symbolique V3A2MF (robot Vertical à 3 Articulations et 2 Membrures Flexibles).

Tableau 3.5: Exemple des noms de répertoires associés au robot de ce projet de recherche

Nom de variable	Exemple
nom system	JCPFGSM
nom mds	V3A2MF
system path	\SYSTEMS\JCPFGSM\
parts path	\SYSTEMS\JCPFGSM\PARTS\
mds root path	\SYSTEMS\JCPFGSM\MOD_SYM\
opt mds path	\SYSTEMS\JCPFGSM\MOD_SYM\OPTIONS\
mod sym path	\SYSTEMS\JCPFGSM\MOD_SYM\V3A2MF\

La Figure 3.3 présente sous forme graphique le lien entre les différents sous répertoires associés à un système.

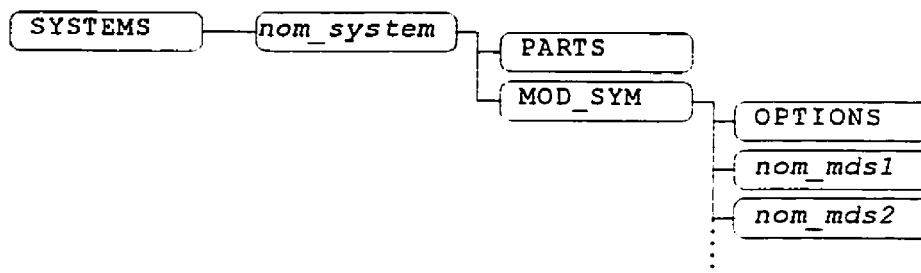


Figure 3.3: Arbre des répertoires associés à un système

3.4.1.5 Répertoires des fonctions et des fichiers de résultats associés à un projet (utilisateur): \RVSEPROJ

En plus, de permettre l'étude de différents systèmes, le logiciel peut être utilisé pour différents projets. C'est pourquoi tous les projets sont rassemblés dans le répertoire RVSEPROJ. Le Tableau 3.6 défini quelques variables qui sont utilisées pour décrire les sous répertoires des fonctions associés à un projet (Tableau 3.7).

Tableau 3.6: Variables servant à décrire les répertoires associés à un projet

Nom de variable	Définition
nom_projet	nom d'un projet (utilisateur)
nom_cfg	nom d'un fichier de configuration
nom_ext	nom d'un fichier de charge à l'extrémité du système

Tableau 3.7: Contenue et description des répertoires associés à un projet: \RVSEPROJ

Répertoire	Contenue et description
proj_path	programmes et fonctions de base pour un système => \RVSEPROJ*nom projet*\
cfg_par_path	fichiers de configuration et de paramètres => *proj path* CFG PAR\
options_path	fichiers d'options => *proj path* OPTIONS\
data_root_path	répertoire de base des fichiers de résultats => *proj path* DATA\
gain_path	fichiers de gains des contrôleurs => *data root path nom cfg*\
data_sub_path	sous répertoires des fichiers de résultats => *nom cfg**nom ext*\
data_path	fichiers de résultats => *data root path data sub path*
stat_path	fichiers de statistiques des erreurs => *data path* STAT\
mod_num_path	fichiers résultats de modélisation numérique => *system path* MOD NUM\ *data sub path*

Le Tableau 3.8 donne un exemple des noms de répertoires associés au projet relié à ce mémoire (MEM_FG), pour les essais d'identifications de la première membrure flexible (ID_MF1) sans charge à l'extrémité (NO_LOAD).

Tableau 3.8:Exemple des noms de répertoires associés au projet relié à ce mémoire

Nom de variable	Exemple
nom_projet	MEM_FG
nom_cfg	ID_MF1
nom_ext	NO_LOAD
proj_path	\RVSEPROJ\MEM_FG\
options_path	\RVSEPROJ\MEM_FG\OPTIONS\
cfg_par_path	\RVSEPROJ\MEM_FG\CFG_PAR\
data_root_path	\RVSEPROJ\MEM_FG\DATA\
data_sub_path	ID_MF1\NO_LOAD\
gain_path	\RVSEPROJ\MEM_FG\DATA\ID_MF1\
data_path	\RVSEPROJ\MEM_FG\DATA\ID_MF1\NO_LOAD\
stat_path	\RVSEPROJ\MEM_FG\DATA\ID_MF1\NO_LOAD\STAT\
mod_num_path	\RVSEPROJ\MEM_FG\MOD_NUM\ID_MF1\NO_LOAD\

La Figure 3.4 présente sous forme graphique le lien entre les différents sous répertoires associés à un projet.

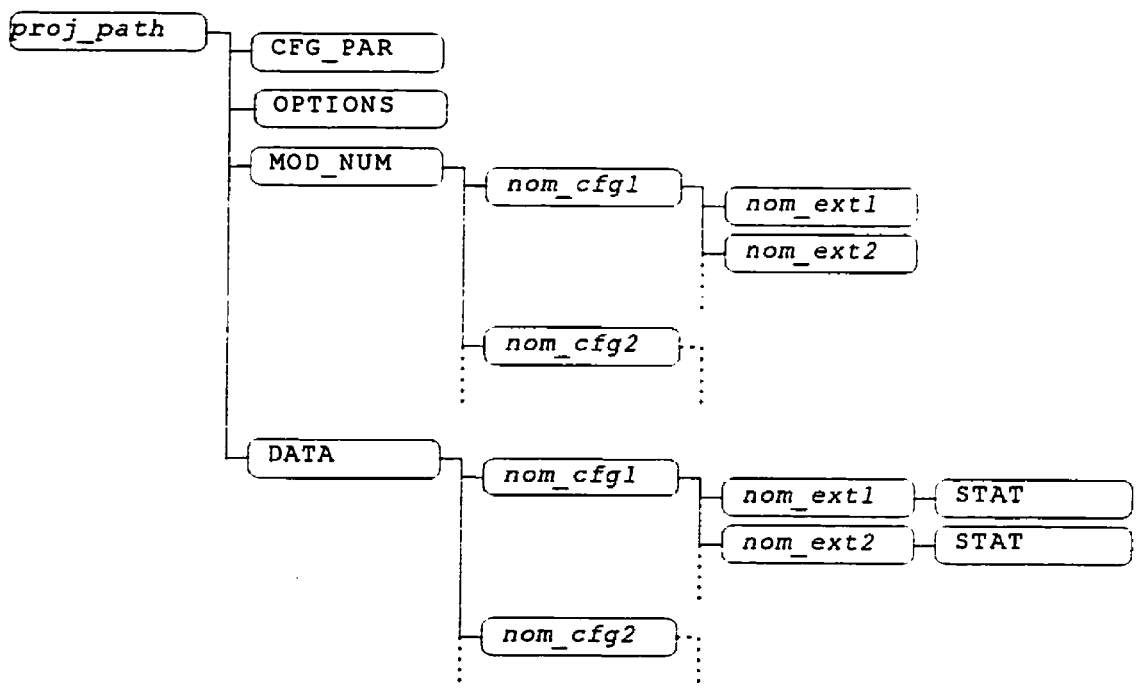


Figure 3.4: Arbre des répertoires associés à un projet

3.4.2 Relation entre les fichiers d'entrée et les fichiers de sortie des modules

Les fichiers de données correspondent aux paramètres d'entrée ou de sortie des différents modules du logiciel ROVASIMEX. Pour bien identifier les fichiers de données, on utilise différentes extensions. Les prochains tableaux indiquent le répertoire, l'extension des fichiers, le logiciel (Matlab ou Maple) avec lequel ils sont utilisés ainsi qu'une courte description.

Tableau 3.9: Fichiers d'entrées des modules

Répertoire	Extension	Logiciel	Description
parts_path	AMP	Matlab	paramètres d'un amplificateur
parts_path	ACT	Matlab	paramètres d'un actionneur
parts_path	POS	Matlab	paramètres d'un capteur de position
parts_path	TAC	Matlab	paramètres d'un capteur de vitesse
parts_path	FRC	Matlab	paramètres d'un capteur de force
parts_path	ART	Matlab	paramètres d'une articulation
parts_path	MEM	Matlab	paramètres d'une membrure
LIN_FORM	LIN F??	Maple	liste de linéarisation fonctions de forme
opt_mds_path	OPT	Maple	options de génération (*fn_opt*) de modèle symbolique
mds_root_path	SYM	Maple	paramètres symboliques (*fn_sym*)

Tableau 3.10: Fichiers de sorties des modules

Répertoire	Variable	Ext.	Logiciel	Description
cfg_par_path	*fn_cfg*	CFG	Matlab	configuration d'un système
mod_sym_path	*fn_mds*	M	Maple	modèle symbolique
cfg_par_path	*fn_par*	PAR	Matlab	paramètres globaux du système
mod_num_path	*fn_dat* *fn_mdn*	DAT MEX	Maple Matlab	paramètres numériques modèle numérique
data_path	*fn_trj*	TC? TD?	Matlab Matlab	trajectoire cinématique trajectoire dynamique
data_path	*fn_cmd*	C0? C_?	Matlab Matlab	essai en boucle ouverte essai en boucle fermée
gain_path	*fn_ctr*	G??	Matlab	paramètres d'un contrôleur (gains)
data_path	*fn_sim*	S??	Matlab	résultats d'une simulation
data_path	*fn_exp*	X??	Matlab	résultats d'un essai expérimental

Tableau 3.11: Fichiers de données pour l'analyse des résultats (identifications et graphiques)

Répertoire	Variable	Ext.	Logiciel	Description
data_path	*fn_data*	T*;C*; S*;X*	Matlab	trajectoires, entrées d'essais simulations, expérimentations
stat_path	*fn_stat*	*	Matlab	statistiques d'erreurs

Tel qu'énoncé précédemment, les fichiers de données sont dépendants les uns des autres. Le diagramme de la Figure 3.5 présente la relation entre les différents fichiers.

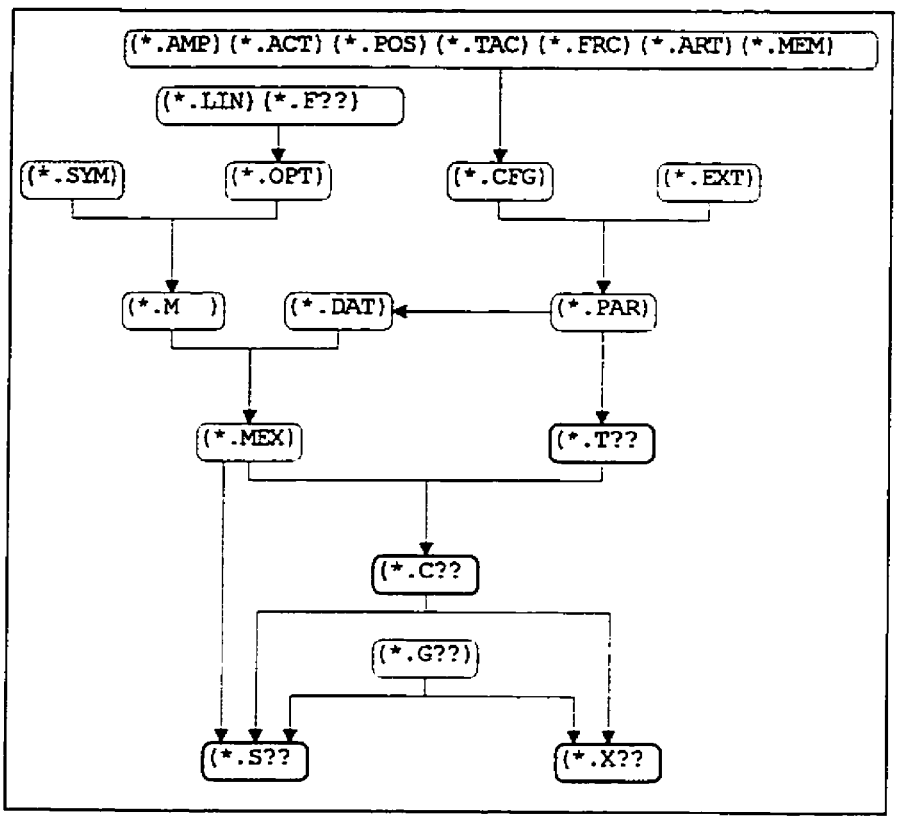


Figure 3.5: Structure hiérarchique des fichiers de données

3.4.3 Noms par défaut des fichiers de sorties

Encore une fois, pour faciliter l'utilisation, on propose des noms lors de la sauvegarde des fichiers de sorties des différents modules. Les noms des fichiers de sorties sont établis en fonction des choix effectués dans le module. Le Tableau 3.12 dresse la liste des variables correspondant aux choix disponibles lors de la génération de modèles symboliques.

Tableau 3.12: Liste des choix lors de la génération des modèles symboliques et numériques

Variable	description	valeurs possibles
chx_entree_mod	choix du type d'entrées des modèles symboliques	U: entrées forces, couples R: entrées coordonnées rigides
chx_principe	choix du principe de calculs des modèles symboliques	J: méthode de Jourdain L: méthode de Lagrange
chx_mod_elast	choix d'inclure l'élasticité aux modèles symboliques	_: sans élasticité E: avec élasticité
chx_methode	choix de la méthode de calculs des modèles numériques	G: globale R: récursif

Les noms des fichiers de sorties sont principalement composés à partir des noms des fichiers d'entrées des modules (voir le Tableau 3.13). Par exemple, lors de la création d'un fichier de paramètres, on utilise le nom du fichier de configuration en y ajoutant un caractère identifiant la charge à l'extrémité du système (*ident_charge*).

Tableau 3.13: Noms par défaut des fichiers de sorties

Variable nom de fichier	Caractères provenant de	Caractères modifiables par l'utilisateur
fn_par	*fn_cfg*(1à7)+*ident_charge*	
fn_mds	*chx_entree_mod*+*chx_principe*+ *chx_mod_elast*+*fn_opt*(1à4)	
fn_mdn	*fn_mds*(1à7)+*chx_methode*	*fn_mdn*(3à7)
fn_dat	*fn_mdn*(1à8)	
fn_cmd	*fn_trj*(1à3)+*fn_mdn*(3à6)+	*fn_cmd*(4à7)
fn_sim	*fn_cmd*(1à7)+*fn_ctr*(1)	*fn_sim*(4à7)
fn_exp	*fn_cmd*(1à7)+*fn_ctr*(1)	*fn_exp*(4à7)

Les fichiers de statistiques d'erreurs (différences entre le résultat de deux fichiers de données) utilisent le nom du premier fichier de données sélectionné. Les trois caractères de l'extension des fichiers de statistiques dépendent du type de variable utilisée et des types de fichiers sélectionnés. Le Tableau 3.14 dresse la liste des premiers caractères de l'extension des fichiers de statistiques et des types de variables utilisées. Les deux autres caractères

utilisent le premier caractère de l'extension de chacun des deux fichiers de données sélectionnés.

Tableau 3.14: Liste des premiers caractères de l'extension des fichiers de statistiques

Caractères	Types de variables utilisées
A	variables d'articulations
X	variables positions, orientations extrémité
R	coordonnées rigides
E	coordonnées élastiques
F	coordonnées flexibles
N	déformation des membrures
C	courbures et torsions des membrures
U	commandes, corrections, friction {aux articulations}
T	forces et couples à l'extrémité[4]

3.5 Conclusion

Le logiciel ROVASIMEX est exécuté à partir du logiciel Matlab et Simulink. Il permet, en faisant appel au logiciel SYMOFROS et à Maple, de générer des modèles symboliques de robots. Par la suite les modèles numériques sont générés et peuvent être utilisé pour la simulation et pour la commande d'un robot. Le logiciel ROVASIMEX a donc été conçu en relation avec les logiciels commerciaux Matlab, Simulink et Maple. Le logiciel est modulaire et facilement portable d'un ordinateur à un autre. De plus, différentes configurations de robots peuvent être modélisées et contrôlées.

[4]Les statistiques pour le type de variables: forces et couples à l'extrémité, ne sont pas encore disponibles.

4 UTILISATION DU LOGICIEL ROVASIMEX

Le logiciel ROVASIMEX donne accès dans sa fenêtre d'interface à 7 modules principaux. Ces modules du logiciel correspondent aux principales étapes à suivre pour être en mesure de simuler et réaliser des essais expérimentaux pour analyser un robot. Tout d'abord, la configuration du système est établie et un modèle symbolique correspondant à cette configuration doit être généré. Après avoir initialisé les paramètres numériques et avoir calculé les capacités du système, un modèle numérique est généré. Des essais en boucle ouverte permettent de vérifier ou d'identifier les paramètres du système. Lorsque les paramètres sont bien identifiés, on peut générer une tâche à réaliser qui consiste à un suivi de trajectoires au niveau des articulations. Afin de pouvoir suivre les trajectoires, il faut choisir un contrôleur et ajuster ses paramètres. En utilisant les paramètres du contrôleur, on effectue des simulations ou on réalise des essais avec le montage expérimental. Pour terminer, les graphiques obtenus à partir des résultats permettent d'analyser le comportement du système. Dans ce chapitre, nous décrivons l'utilisation du logiciel à partir du démarrage jusqu'à la visualisation des résultats.

4.1 Démarrage du logiciel

Le programme `startup` (`\RVSE_FCT`) dans la fenêtre de commande Matlab démarre le logiciel. Ce programme utilise un sous programme (`rvse_cfg`) qui permet d'ajouter, d'initialiser ou de supprimer un projet.

La fenêtre d'interface (Figure 4.1) ajouter un projet permet:

- d'inscrire le nom du projet (utilisateur);
- de sélectionner un système (robot manipulateur);

- de choisir si l'on veut inclure les fichiers de démonstrations correspondant au système sélectionné.

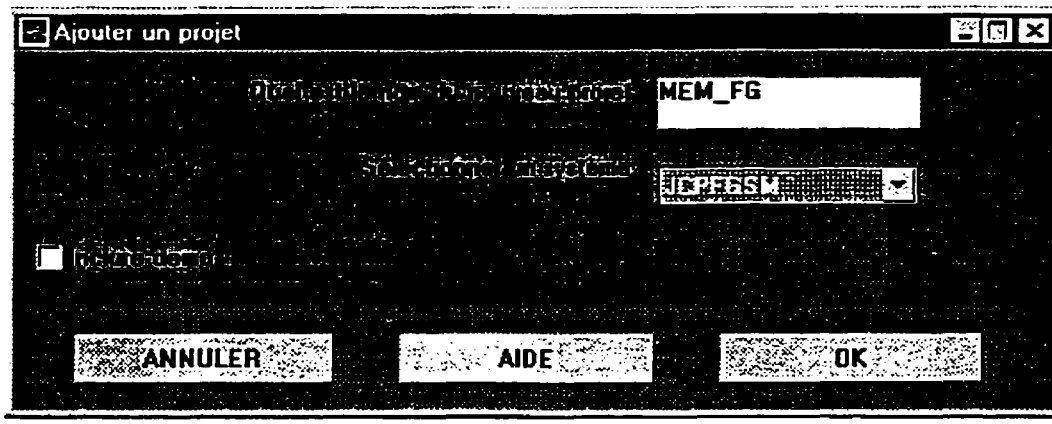


Figure 4.1 Fenêtre d'interface pour ajouter un projet

Les fichiers de démonstrations contiennent les fichiers correspondant aux principales configurations du système. Les fichiers de démonstrations[1] permet d'accéder directement à l'étape "4.5 Génération de trajectoires" (p.66) sans avoir à configurer le système et à générer les modèles numériques.

Lors de l'initialisation d'un projet, le sous-programme assigne les chemins de recherche des fonctions Matlab.

Voici l'ordre dans lequel les chemins sont parcourus:

1) répertoire associé au projet (`\RVSEPROJ\`*`nom_projet`*);
2) répertoires associés au système sélectionné (`\SYSTEMS\`*`nom_system`*; `\SYSTEMS\`*`nom_system`*`\cfg_par`; `\SYSTEMS\`*`nom_system`*`\parts`);
3) répertoires des fonctions et programmes du logiciel ROVASIMEX (`\RVSE_FCT` et tous ses sous répertoires);
4) répertoire des fonctions utilitaires Matlab (`\SRC_UTIL\MAT40`);
5) répertoires des librairies de fonctions Matlab;

[1] Pour obtenir la liste des fichiers de démonstrations consulter le document : Manuel technique du banc d'essai pour l'étude théorique et expérimentale de robots flexibles, École Polytechnique de Montréal.

Pour terminer, le programme `startup` crée le menu principal et tous ces sous menus.

4.1.1 Ré initialisation du logiciel avec la commande "rstrvse"

Tout au long du déroulement du logiciel, la fenêtre de menu principal est souvent rendue invisible. Lors de la mise au point du logiciel, une erreur dans un sous-programme laisse la fenêtre invisible. La commande `rstRVSE` ré initialise le logiciel ROVASIMEX. En plus de ré afficher la fenêtre de menu principal, les variables et les fonctions sont effacées de la mémoire du logiciel Matlab.

4.2 Gestionnaire de projets

À partir de la fenêtre principale du logiciel, on peut sélectionner un projet ou activer la fenêtre d'interface de gestionnaire de projet (Figure 4.2) qui permet d'ajouter, de supprimer, d'enregistrer sur disquette ou de récupérer à partir d'une disquette un fichier projet (`*.PRJ`).

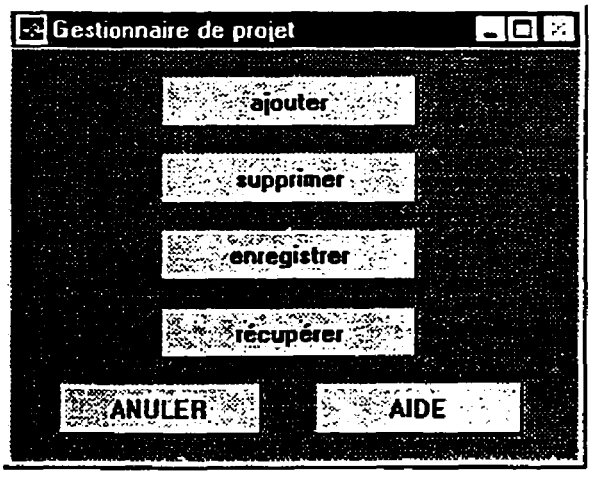


Figure 4.2: Fenêtre d'interface de gestionnaire de projet

Avant de supprimer le projet actif, il faut confirmer (Figure 4.3) son effacement de la liste des projets disponibles.

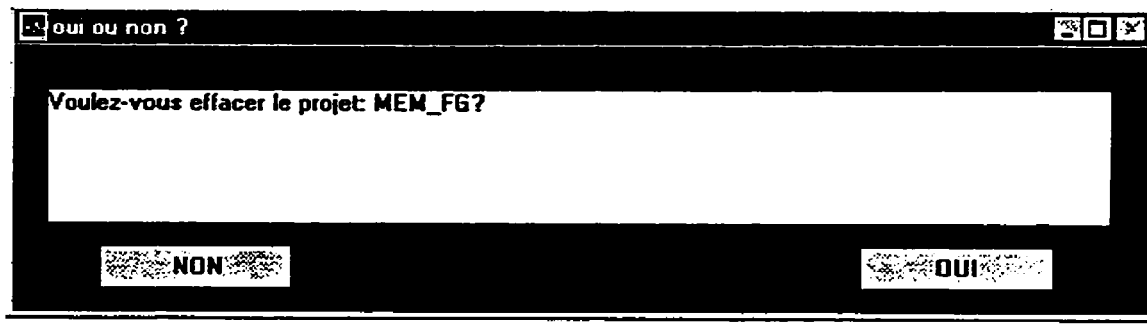


Figure 4.3: Fenêtre d'interface pour confirmer l'effacement d'un projet

Lors de la sauvegarde, il faut sélectionner (Figure 4.4) les catégories de résultats que l'on veut enregistrer sur la disquette. Si le projet ne contient pas beaucoup de données, on peut choisir d'enregistrer tout les résultats, sinon on mettra sur différentes disquettes chaque catégorie de résultats.

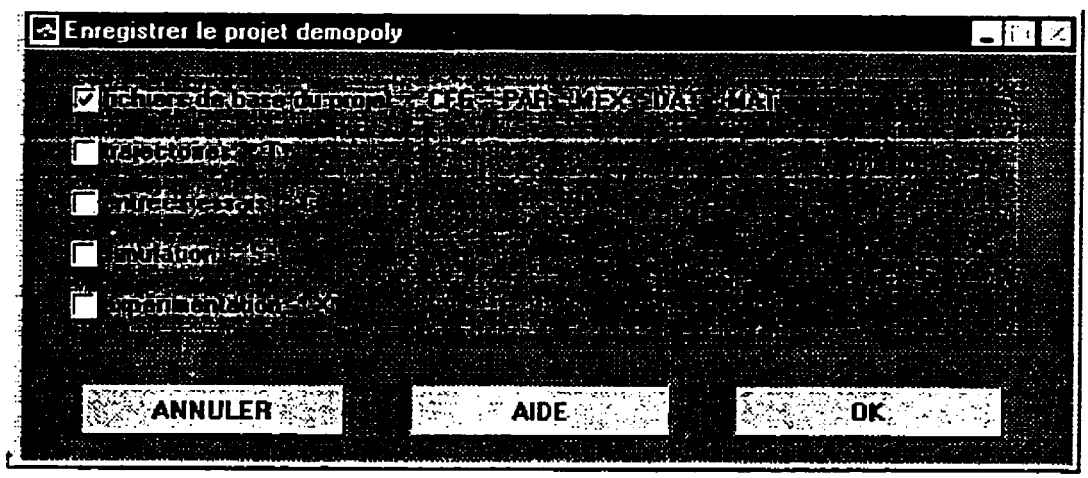


Figure 4.4: Fenêtre d'interface pour enregistrer les résultats d un projet sur disquette

Pour récupérer un fichier de projet, il suffit de sélectionner un fichier qui a été préalablement enregistrer par le logiciel.

4.3 Configuration d'un système et génération de modèles symboliques

Afin de pouvoir générer des modèles symboliques, il faut d'abord assigner les référentiels aux différentes composantes mécaniques du système (paragraphe 4.3.1) et définir les paramètres symboliques (paragraphe 4.3.2). Les modèles numériques seront générés en remplaçant les paramètres symboliques par leurs valeurs numériques. Ces paramètres numériques sont définis à l'intérieur des fichiers de composantes du système (paragraphe 4.3.3). C'est à partir des noms de fichiers de composantes que la configuration du système est définie (paragraphe 4.3.4).

4.3.1 Assigner les référentiels du système

La Figure 4.5 présente les référentiels du montage expérimental de ce projet. On y voit aussi les centres de masse de chacun des corps. Les formes triangulaires représentent les articulations. Les axes X sont des lignes pointillés, tandis que les lignes tiret point sont pour les axes Y.

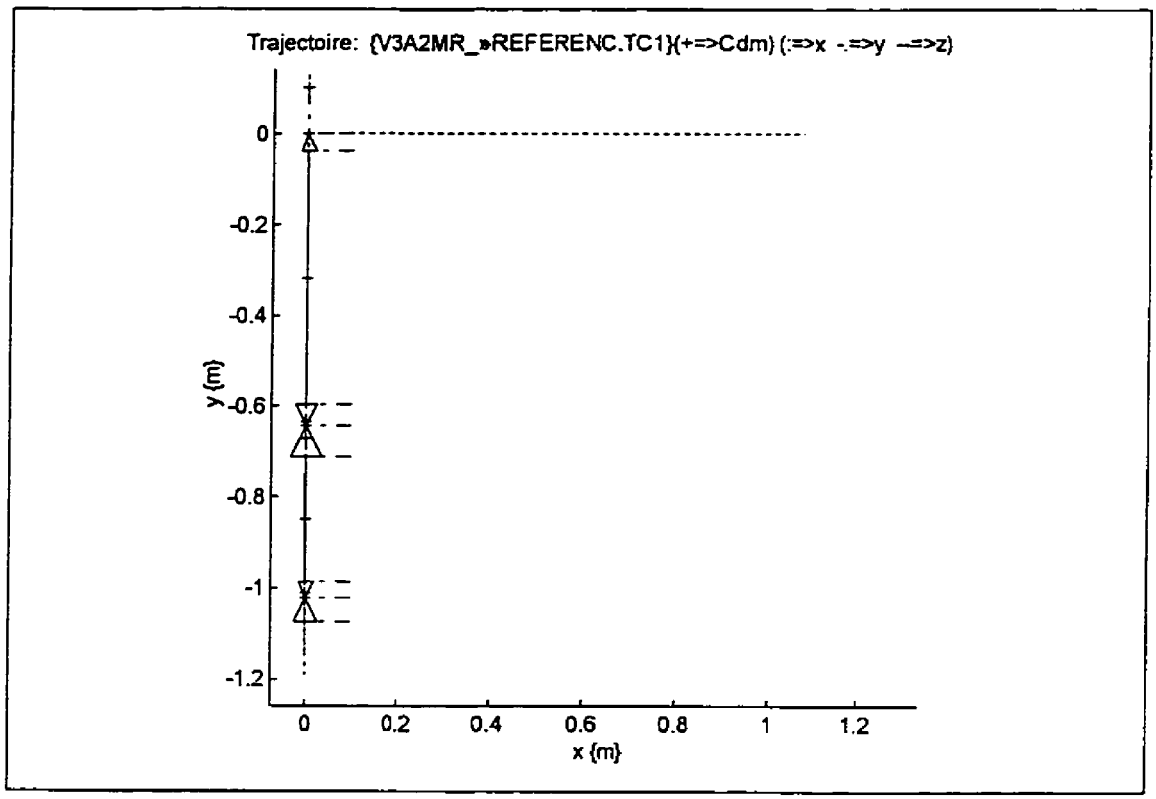


Figure 4.5: Visualisation des référentiels et des centres de masse

4.3.2 Écrire un fichier de paramètres symboliques Maple (*.SYM)

Puisque l'on doit faire correspondre les paramètres symboliques aux paramètres numériques, il est important de respecter une certaine notation. Le fichier SYMBOL.NOT[2] (\RVSE_FCT\GEN_MOD) décrit la notation utilisée.

L'assignation des référentiels des corps (composantes mécaniques) du système est totalement libre sauf pour le référentiel de la base (référentiel observateur) et pour les référentiels des membrures. Le référentiel de la base dépend du type de système (robots

[2] Le document « Manuel technique du banc d'essai pour l'étude théorique et expérimentale de robots flexibles » présente le fichier SYMBOL.NOT et les fichiers de paramètres symboliques correspondants aux différentes possibilités de configurations du système expérimental de l'École Polytechnique de Montréal.

planaires et robots à déplacement dans l'espace tridimensionnel). Les robots planaires se déplacent dans le plan XY. Si le plan est vertical, alors l'accélération gravitationnelle pointe vers le bas et l'axe Y du référentiel observateur pointe vers le haut. Sinon la gravité est selon l'axe Z (négatif). Pour les référentiels observateurs des robots tridimensionnels, c'est toujours l'axe Z qui pointe vers le haut. Cette assignation de référentiel correspond aux axes utilisés par le logiciel Matlab et évite une transformation supplémentaire des données lors de la visualisation infographique du robot. Pour ce qui est des membrures, l'axe X de leur référentiel en direction de la longueur de la membrure. Par conséquent, la torsion de la membrure s'effectue autour de son axe X. Autant que possible, la déformation principale de la membrure devra être selon l'axe Y (inertie autour de Z).

L'inertie de chacun des corps (composantes) du système peut être définie par rapport à leur centre de masse ou par rapport à leur référentiel. Même si l'inertie est définie par rapport au référentiel du corps, on a besoin du centre de masse du corps pour calculer l'effet de l'accélération gravitationnelle.

4.3.3 Écrire les fichiers de composantes du système

En plus d'écrire les fichiers de définition des paramètres numériques (distance, masse, centre de masse, inertie, etc.) pour chacune des composantes mécaniques du système (actionneurs, articulations, membrures, charges), il faut écrire les fichiers de définition des paramètres électriques (gains, facteurs de conversion des capteurs, etc.) des composantes du système permettant l'acquisition et le contrôle en temps réel (amplificateurs, actionneurs, capteurs).

Les paramètres des différentes composantes du système sont définis dans des programmes Matlab (*.M)[3]. A la fin de chacun de ces programmes, les variables sont sauvegardés dans des fichiers binaires. Lorsque l'on crée un nouveau fichier de définition de paramètres, il faut l'exécuter une première fois pour créer le fichier de données.

4.3.4 Créer un fichier de configuration (*.CFG)

Le logiciel ROVASIMEX permet de créer un fichier de configuration en sélectionnant les différentes composantes du système [Config=> Créer=][4]. Le fichier de configuration contient le nombre d'articulations, la liste des noms de fichiers de paramètres pour chacune des composantes du système et le nom du fichier de paramètres symboliques correspondant à cette configuration.

Le module de configuration permet de faire la mise à jour des fichiers de paramètres des composantes [Config=> Mettre à jour] ou encore d'afficher à l'écran le contenu d'un fichier de configuration [Config=> Afficher].

4.3.5 Générer un modèle symbolique Maple (*.M)

Pour générer un modèle symbolique [Config=> Générer un modèle symbolique], il faut sélectionner un fichier de configuration (*.CFG) et un fichier d'options de modélisation (*.OPT). Les fichiers d'options de modélisation assignent la liste de linéarisation du système (*.LIN) et définissent le nombre de modes des membrures flexibles ainsi que les fonctions de formes (*.F??). Les modèles symboliques peuvent inclure l'élasticité des articulations (ou des actionneurs).

---

[3] Le document « Manuel technique du banc d'essai pour l'étude théorique et expérimentale de robots flexibles » contient une courte description pour chacun des fichiers de composantes disponibles du montage expérimental.

[4] La séquence entre crochet[ ] décrit la suite des options à sélectionner, en commençant par l'option du menu principal.

Deux possibilités de modèles symboliques peuvent être générées:

1) avec comme entrée les coordonnées rigides (système à membrures flexibles);
2) avec comme entrée les couples ou les forces aux articulations.

Les modèles avec comme entrée les variables d'articulations sont principalement utilisés lors des simulations, tandis que les modèles avec comme entrée les couples sont nécessaires pour calculer la commande anticipative et pour effectuer des simulations du système en incluant la friction.

La Figure 4.6 présente la fenêtre d'interface de génération de modèle symbolique.

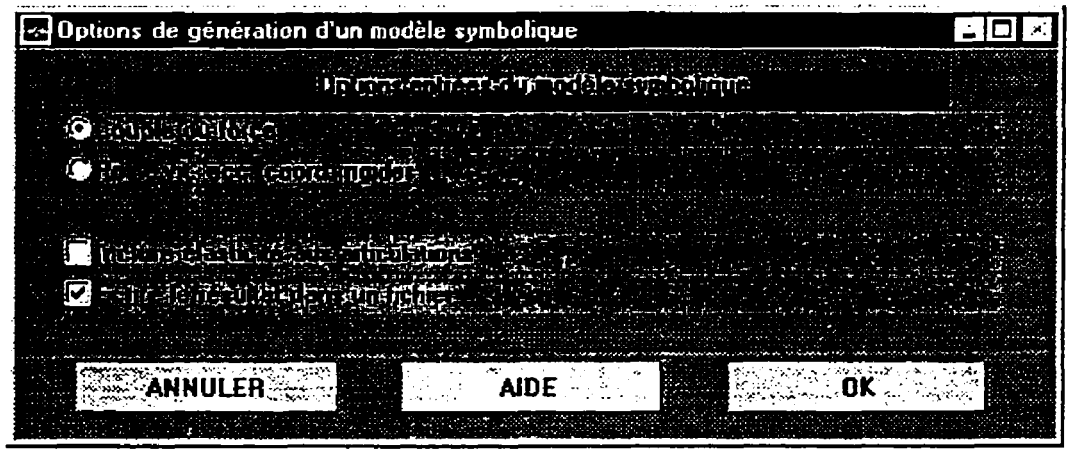


Figure 4.6 Fenêtre d'interface de génération de modèle symbolique

4.4 Initialisation des paramètres globaux et génération de modèles numériques

Après avoir créé un fichier de configuration, il faut initialiser les paramètres du système et les rassembler dans un seul fichier [`Initial=> Créer un fichier de paramètres`]. Par la suite, les modèles numériques pourront être générés [`Initial=>`

`Créer un modèle numérique`] en sélectionnant un fichier de paramètres et un fichier de modélisation symbolique.

4.4.1 Créer un fichier de paramètres numériques (*.PAR)

L'initialisation des paramètres s'effectue en deux phases. La première phase est l'initialisation générale des paramètres d'un système qui consiste à lire les fichiers de paramètres (mécaniques et électriques) des différentes composantes, à ajouter les paramètres de la charge à l'extrémité et à initialiser les vecteurs de paramètres globaux du système. En un deuxième temps, la position de repère ("home position") du système est fixée et ses capacités sont calculées (ex.: l'accélération maximum des actionneurs, les dimensions de l'espace de travail, déformation statique des membrures flexibles). C'est le sous-programme `init_rob` du répertoire associé au système (`\SYSTEMS\nom_systeme`) qui réalise cette deuxième phase.

4.4.2 Créer un modèle numérique (*.MEX)

Pour continuer, il faut générer un modèle numérique correspondant à un fichier de paramètres. Il y a deux catégories de modèles numériques: modèles récursifs ou modèles globaux ("close form"). Le fichier de paramètres numériques (*.DAT) à être associé aux paramètres du modèle symbolique (*.M) est créé à partir du fichier de paramètres globaux du système (*.PAR). Lors de la génération des modèles numériques, on peut laisser libres certains paramètres. Les valeurs nominales des paramètres sont alors proposées lors de l'initialisation du modèle.

4.4.3 Mettre à jour un fichier de paramètres et les modèles numériques correspondants

Dans l'éventualité où un fichier de définition de paramètres d'une composante du système a été modifié, il faut alors faire la mise à jour des fichiers de paramètres sans avoir à les recréer [`Initial=> Mettre à jour les paramètres`]. Le logiciel offre la possibilité de récupérer un fichier pour en examiner son contenu [`Initial=> Afficher les paramètres`]. Si l'un des paramètres modifiés est inclus dans les modèles symboliques, il faudra alors régénérer les modèles numériques [`Initial=> Générer un modèle numérique`].

4.5 Génération de trajectoires

La tâche à être exécutée par le manipulateur consiste à ce que l'extrémité du robot suive une trajectoire désirée. Il y a deux types de trajectoires: les trajectoires cinématiques (position, vitesse, accélération et orientation) et les trajectoires dynamiques (ajout de forces). Pour l'instant, seule la génération de trajectoires cinématiques (dans l'espace des variables d'articulations) qui est disponible. La génération de trajectoires dynamiques pourra être implantée lors de travaux de recherche subséquents.

4.5.1 Générer une trajectoire cinématique(`*.TC?`)

Les trajectoires cinématiques des articulations sont calculées en négligeant la flexibilité des membrures et l'effet d'élasticité des actionneurs [`Traject=> Trajectoires cinématiques`]. Toutes les trajectoires générées par le programme `traj_cin` (`\RVSE_FCT\TRAJECT\`) débutent et se terminent à la position de repère du manipulateur "home position", à vitesse et accélération nulles.

Chaque trajectoire est sectionnée en trois parties (départ, intermédiaire, retour):

1) la première partie est la trajectoire des articulations pour atteindre une position de départ;
2) si le nombre de degrés de liberté le permet, la partie intermédiaire peut être alors composée d'une suite d'éléments de base (lignes, polygones, arcs de cercle, cercles), sinon la partie intermédiaire de la trajectoire n'est fonction que des variables d'articulations;
3) la dernière partie est la trajectoire de retour à la position de repère du système.

Lors d'essais avec le système expérimental, la deuxième partie de la trajectoire pourra être répétée plusieurs fois, à la condition que la dernière position des articulations soit identique à la position de départ. Ceci permet de vérifier la répétabilité du système. Chaque partie ou élément de la trajectoire est calculé à l'aide d'une fonction de génération de trajectoire.

Voici la liste des choix de fonctions de génération de trajectoires:

- avec dérivée d'accélération triangulaire (`*.TC1`);
- avec dérivée d'accélération échelon (`*.TC2`);
- avec temps fixes et dérivée d'accélération sans discontinuité (`*.TC3`);
- avec temps fixes et dérivée d'accélération discontinue (`*.TC4`).

D'autres fonctions de génération de trajectoire pourraient être implantées dans le logiciel. Les arguments d'entrée des fonctions sont: la période d'échantillonnage, le déplacement désiré et les valeurs limites de vitesse, d'accélération et de dérivée d'accélération. Il est à noter que seul le déplacement désiré est assurément atteint, les autres arguments sont des paramètres limites à atteindre si c'est possible. Au début du module, les valeurs par défaut des arguments (sauf le déplacement) sont assignées aux valeurs limites (capacités du système) et peuvent être modifiées par l'utilisateur. De plus, lors de la génération de chaques éléments de trajectoires, la valeur de la vitesse désirée peut être encore modifiée.

Puisque les trajectoires sont dans l'espace des variables d'articulations, il faut donc convertir les trajectoires d'éléments de base (lignes, etc.). La fonction `c_inv` (`\SYSTEMS\nom_systeme`) permet de calculer la trajectoire des articulations par la cinématique inverse. Entre chaque élément de la trajectoire, le programme recalcule la position et l'orientation de l'extrémité à l'aide de la fonction de cinématique directe `posormod` (`\RVSE_FCT\TRT_DATA`). Cette fonction utilise une fonction de modélisation numérique (`*.MEX`) pour retourner la position et la matrice de rotation de l'extrémité. La fonction `rot2angl` (`\SYSTEMS\nom_systeme`) convertit la matrice de rotation en un vecteur d'angles d'orientation. Par conséquent, le choix du type d'orientation (angles d'Euler, quaternions) dépend du système assigné au projet. La cinématique inverse est alors en relation avec ce choix de type d'orientation.

4.5.2 Éditer un fichier de trajectoire

Le logiciel permet de sélectionner un fichier de trajectoire [`Traject=> Éditer trajectoire`] pour ensuite avoir la possibilité de modifier les paramètres demandés à l'usager à chacune des étapes de la génération de la trajectoire. La nouvelle trajectoire peut être enregistrée dans le même fichier ou sous un autre nom de fichier.

4.5.3 Afficher la description d'une trajectoire

Lors de la génération de trajectoire, un texte décrivant la trajectoire est créé. Ce texte présente la suite de type de trajectoires (variables d'articulations, lignes, etc.) et leurs paramètres correspondants (position, vitesse, longueur, angle, etc.). Cette option du menu permet de récupérer un fichier de trajectoire et d'afficher son texte descriptif [`Traject=> Afficher la description`].

4.6 Calculs des paramètres d'entrée pour les essais en simulation ou expérimentaux

Avant d'effectuer un essai en boucle ouverte ou en boucle fermée, il faut créer un fichier d'entrée pour les essais [`Consignes=> Créer un fichier de consignes`] qui contient tous les paramètres nécessaires pour simuler le système ou réaliser un essai avec le système expérimental.

4.6.1 Créer un fichier de consignes (paramètres d'entrée) pour un essai en boucle ouverte (`*.C0?`)

Pour créer un fichier d'entrée d'un essai en boucle ouverte [`Consignes=> Créer fichiers de consignes`], il faut sélectionner le contrôleur `0)Test en boucle ouverte` dans la fenêtre d'interface (Figure 4.7).

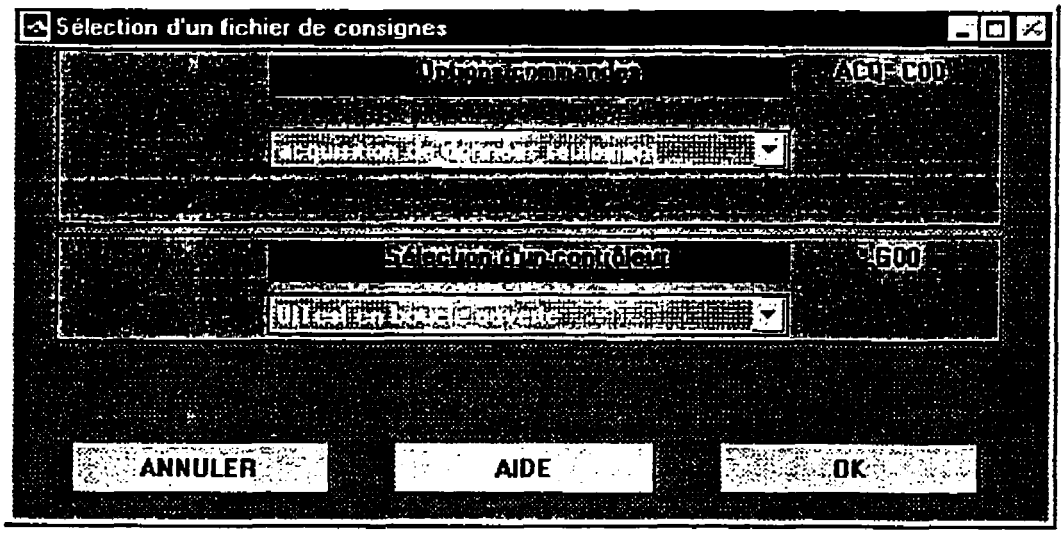


Figure 4.7 Fenêtre d'interface pour créer un fichier de consignes pour un essai en boucle ouverte

Le logiciel permet de générer différents types de signaux:

- acquisition de données seulement;
- échelon;
- rampe;
- onde carrée;
- onde triangulaire;
- onde sinusoïdale;
- onde AM FM;
- bruit blanc ;
- bruit blanc booléen.

Il est à noter que les essais en boucle ouverte ne se font que sur un système à une seule articulation. Pour faire des essais en boucle ouverte avec un système ayant plus d'une articulation, il faut réaliser un essai en boucle fermée avec commande anticipative et en utilisant des gains nuls pour le contrôleur (`0.G??`). Les paramètres contenus dans le fichier pour les tests en boucle ouverte sont: la période d'échantillonnage, le temps de l'essai et la commande appliquée à l'actionneur.

4.6.2 Éditer un fichier de consignes (paramètres d'entrée) pour un essai en boucle ouverte

Le logiciel permet aussi de sélectionner un fichier de consignes pour un test en boucle ouverte, pour ensuite modifier les paramètres du signal. Le nouveau signal peut être enregistré dans le même fichier ou sous un autre nom de fichier.

4.6.3 Créer un fichier de consignes (paramètres d'entrée) pour un essai en boucle fermée (`*.c_?`)

Pour créer un fichier de consignes pour un essai en boucle fermée [`Consignes=> Créer fichiers de consignes`], il suffit de sélectionner un contrôleur dans la fenêtre d'interface (Figure 4.8) et choisir les options.

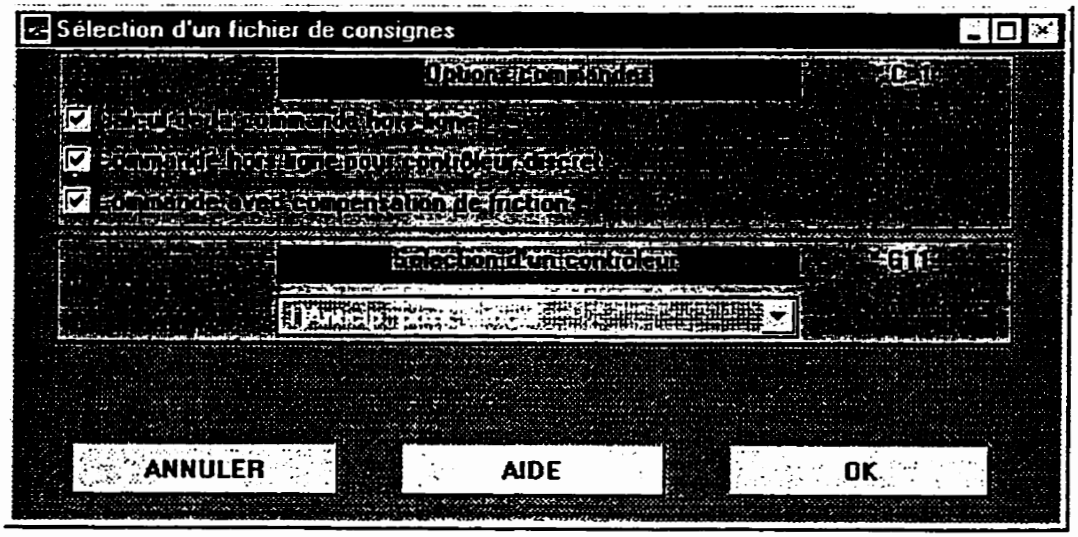


Figure 4.8 Fenêtre d'interface pour créer un fichier de consignes pour un essai en boucle fermée

Dans le cadre de ce projet, il y a seulement un type de contrôleur disponible[5]: un contrôleur anticipatif pour la position des articulations. Selon le choix des options de commandes, il y a 6 types de fichiers de consignes pour les essais en boucle fermée. Le Tableau 4.1 présente la relation entre l'extension du fichier et les options sélectionnées.

[5] Présentement, l'implantation d'un contrôleur à couple pré calculé est sur le point d'être complété par d'autres étudiants.

Tableau 4.1 Options des fichiers de consignes pour les essais en boucle fermée (*.C_?)

Extension du fichier	Commande anticipative hors ligne	Commande discrète	Commande avec compensation de friction
C_0	NON	-	NON
C_1	OUI	OUI	OUI
C_2	OUI	OUI	NON
C_3	OUI	NON	OUI
C_4	OUI	NON	NON
C_5	NON	-	OUI

La commande anticipative consiste à calculer le couple des actionneurs en fonction de la trajectoire (coordonnées rigides désirées) à suivre. De plus, la trajectoire est corrigée pour compenser la flexibilité des membrures et l'élasticité des articulations. Ayant identifié les paramètres de friction aux articulations, il est possible d'ajouter une compensation due à la friction. Les paramètres d'entrée des essais en boucle fermée sont: la période d'échantillonnage, la position et la vitesse désirée des coordonnées rigides de même que la commande anticipative.

4.6.4 Créer un fichier de gains (paramètres) du contrôleur (*.G??)

En plus de créer un fichier de paramètres d'entrée d'un essai en boucle fermée, il faut aussi déterminer les gains utilisés par le contrôleur [Consignes=> Paramètres (gains) de contrôleur]. Pour l'instant, le logiciel ROVASIMEX ne propose aucune méthode pour déterminer les gains du contrôleur. Les gains ont été obtenus de façon heuristique, par essais et erreurs. Par contre, le logiciel permet de vérifier la stabilité du contrôleur, en calculant les pôles (discrets) du système en boucle fermée (sans friction).

4.7 Simulation du système

Les programmes de simulation utilisent le logiciel Simulink. Pour effectuer une simulation, il faut sélectionner un contrôleur. A chaque type de contrôleur correspond un programme de simulation. Comme indiqué auparavant, si on simule un essai en boucle fermée, la trajectoire désirée est sectionnée en trois segments. La simulation des trois parties peut être effectuée automatiquement ou manuellement. La simulation automatique est plus rapide que la simulation manuelle, par contre on perd l'accès aux autres logiciels jusqu'à la fin de la simulation. Lors d'une simulation manuelle, une fenêtre Simulink est alors affichée. Pour chacune des trois parties, il faut d'abord initialiser une section de trajectoire en cliquant deux fois sur un bloc d'initialisation, ensuite on démarre la simulation à partir du menu de la fenêtre ou en appuyant en même temps sur les touches "Ctrl" et "T". Lorsque la simulation est terminée, on clique deux fois sur le bloc d'enregistrement des résultats.

Le logiciel permet d'effectuer deux catégories de simulation:

1) simulation utilisant un modèle avec comme entrée les couples ou les forces aux articulations;
2) simulation utilisant un modèle avec comme entrée les coordonnées rigides (système avec membrures flexibles).

4.7.1 Effectuer une simulation avec comme entrée les couples (forces) aux articulations (`*.s??`)

Pour cette catégorie de simulation [`Simul=> entrée:couples ou forces`], si le fichier d'entrée (`*.c??`) sélectionné correspond à un essai en boucle fermé, il faudra alors sélectionner un fichier de gains (paramètres) du contrôleur (`*.g??`). De plus, on peut inclure l'effet de la friction. Les paramètres de simulation (paramètres de friction, fonction

de modélisation numérique, paramètres libres du modèle) peuvent être différents de ceux utilisés lors de la création du fichier d'entrée de l'essai.

4.7.2 Effectuer une simulation avec comme entrée les coordonnées rigides (`*.S?R`)

La seconde catégorie de simulation [`Simul=> entrée:coordonnées rigides`] est utilisée pour vérifier ou comparer différentes modélisations d'un système à membrures flexibles ou à coordonnées élastiques.

La liste ci-dessous présente les avantages des simulations avec entrée coordonnées rigides:

- pas d'effet de friction aux articulations;
- pas de contrôleur (simulation en boucle ouverte);
- possibilité d'utiliser un fichier de résultats d'expérimentation comme fichier d'entrée pour la simulation;
- simulation plus rapide.

4.8 Essais sur le système expérimental

Pour identifier (vérifier) les paramètres du système ou pour valider les résultats de simulations, il faut réaliser des essais en temps réel avec le système expérimental.

4.8.1 Réaliser un essai expérimental (`*.X??`)

Pour réaliser un essai sur le système expérimental [`Exper=> Essai expérimental`], il faut sélectionner un fichier de paramètres d'entrée d'un essai (`*.C??`). Si le fichier sélectionné correspond à un essai en boucle fermée, alors on doit sélectionner un fichier de gains (paramètres) du contrôleur (`*.G??`). D'autre part, si le fichier correspond à un essai

en boucle ouverte (`*.C0?`), on peut alors utiliser les résultats de l'essai pour vérifier ou identifier différents paramètres.

4.8.2 Vérifier ou identifier des paramètres du système

Le logiciel ROVASIMEX permet d'identifier ou de vérifier différents paramètres [`Exper=> Identification`] à partir des résultats d'essai en boucle ouverte. Le Tableau 4.2 présente la liste des paramètres pouvant être vérifiés ou identifiés. Les essais en boucle ouverte utilisant un signal bruit blanc permettent de faire une analyse spectrale du système [`Exper=> Identification=> Analyse fréquentielle`].

Tableau 4.2: Liste des paramètres pouvant être vérifiés ou identifiés

Paramètres	Type d'essai en boucle ouverte
friction d'une articulation ou d'un actionneur	rampe
gain d'un actionneur	rampe
inertie d'une articulation ou d'un actionneur	signal AM FM
facteurs de conversion des capteurs de flexion v	acquisition de données
facteurs de conversion des capteurs de flexion v et w facteurs de conversion des capteurs de torsion a	acquisition de données

4.9 Traitements des données

Le module de traitements des données permet de vérifier ou d'analyser les résultats des différents modules du logiciel à l'aide de graphiques.

Voici la liste des différents types de fichiers de résultats:

- de génération de trajectoires (`*.T??`);
- de paramètres d'entrée d'essais en boucle ouverte (`*.C0?`) ou d'essais en boucle fermée (`*.C_?`);
- de simulations (`*.S??`);

- d'essais expérimentaux (`*.X??`).

Il y a trois catégories de graphiques pouvant être tracés:

1) visualisations infographiques;
2) graphiques temporels;
3) graphiques de statistiques (histogramme).

4.9.1 Visualisations infographiques

Pour mieux interpréter le comportement du système, le logiciel permet de visualiser les déplacements du système dans son espace de travail. On peut ralentir ou accélérer la visualisation selon un certain ratio: (temps de la visualisation / temps réel de la trajectoire). Plus on augmente ce rapport, plus le nombre de points de la visualisation augmente et se rapproche du nombre total de points de la trajectoire.

Les choix de visualisation sont:

- système dans son ensemble [`Graph=> Visualisation infographique=> Visualisation globale`];
- trajectoire de l'extrémité seulement [`Graph=> Visualisation infographique=> Traject. de l'extrémité`];
- comparaison de deux trajectoires (simultanées) de l'extrémité [`Graph=> Visualisation infographique=> Comparaison traject. extrémité`].

Les formes de chacune des pièces (corps) du système sont définies par une suite de points en coordonnées homogènes *`[x;y;z;1]`* et sont sauvegardées dans les fichiers de paramètres. Actuellement, la représentation des pièces n'utilise que des simples formes géométriques. Les articulations sont représentées par deux triangles, tandis que les membrures sont représentées par des lignes. À la fin de la visualisation globale du système, les référentiels et les centres de masse de chacune des pièces du système peuvent être

affichés. La visualisation tient compte de la déformation des membrures et cette déformation est calculée à partir des états de flexibilité ou des mesures de courbures.

Le logiciel permet de visualiser le déplacement du référentiel à l'extrémité du système. La dimension d'affichage est ajustée en fonction des trajectoires sélectionnées. Lors de la visualisation simultanée de deux trajectoires, on peut observer le retard (ou l'avance) entre les trajectoires. À la fin de la visualisation, les deux trajectoires sont retracées en utilisant tous leurs points. La trace correspondant au premier fichier de données est continue, tandis que la trace pour le second fichier est en pointillée.

4.9.2 Graphiques temporels

Il existe plusieurs types de variables temporelles pouvant être tracées par le logiciel [`Graph=> Variables en fonctions du temps`]:

- variables d'articulations: qa, qa', qa'';
- variables extrémité: position, orientation;
- coordonnées rigides: qr, qr', qr'';
- coordonnées élastiques: qe, qe';
- coordonnées flexibles: $qf(v,w,a)'$;
- déformation des membrures: $qn(v,w,a)$, qn', qnp;
- courbures: $qc(v,w)$ et torsions des membrures: $qc(a)$;
- commandes: u, corrections: unr, friction: ufr; {aux articulations};
- forces et couples à l'extrémité[6].

Le module de traitement des données offre la possibilité de sélectionner deux signaux temporels pour tracer des graphiques de différences [`Graph=> Signal d'erreurs`] ou des graphiques de comparaisons [`Graph=> Comparaison`].

---

[6]Les graphiques pour le type de variables: forces et couples à l'extrémité, ne sont pas encore disponibles.

Après avoir fait la différence entre deux signaux temporels, le logiciel calcule les statistiques suivantes:

- la médiane du signal d'erreur;
- l'écart type du signal d 'erreur;
- la moyenne de la valeur absolue du signal d 'erreur;
- le maximum du signal d 'erreur.

Les statistiques sont sauvegardés dans un fichier (répertoire: *data_path*\STAT). Sachant le temps de la trajectoire désirée, les statistiques sont séparées en deux catégories: les statistiques d'erreurs en régime transitoire et les statistiques d'erreurs en régime permanent. Le logiciel trace les signaux d'erreurs au complet, et affiche (légende axe X) une des deux catégories de statistiques.

Pour les graphiques de comparaisons, la trace en ligne continue correspond au premier fichier de données, tandis que la trace du second fichier est en pointillée.

4.9.3 Graphiques (histogrammes) pour les statistiques de différences entre deux signaux temporels

Pour aider à analyser les résultats, le module de traitement des données offre la possibilité de tracer des graphiques à barres [`Graph=> Statistiques signal d'erreurs`] utilisant les statistiques de signaux d'erreurs (régime transitoire ou régime permanent). Le logiciel permet de sélectionner jusqu'à cinq fichiers de statistiques. Chaque fichier est identifié par un numéro pour former la légende des graphiques.

4.9.4 Options disponibles

Il est à noter que tous les paramètres et les résultats utilisent les unités SI. Le logiciel ROVASIMEX permet de changer d'unités ou d'échelle pour tracer les graphiques. Le Tableau 4.3 dresse la liste des choix disponibles pour chacun des types d'unités. Cette liste de choix est définie dans le programme `sel_unit` (`\RVSE_FCT\TRT_DATA`).

Tableau 4.3: Liste des unités disponibles pour tracer les graphiques

Type	Unités
temps	s, ms
angle	rad, degré
distance	m, cm, mm, pouce
couple	N-m, N-cm
force	N
courbure	1/m, 1/cm, 1/mm, 1/po

De plus, pour faciliter la tâche de l'usager, le logiciel permet de fermer toutes les fenêtres graphiques.

4.10 Implantation de nouveaux contrôleurs

D'autres contrôleurs peuvent s'ajouter à la liste des contrôleurs disponibles [`Consigne=> Ajouter un nouveau contrôleur`]. La Figure 4.9 présente la fenêtre d'interface. Il suffit donc de sélectionner un contrôleur qui servira de base pour créer le nouveau contrôleur. Le nom du contrôleur (maximum 5 caractères) est utilisé pour créer deux nouveaux fichiers qui devront modifiés. Il y a un fichier qui assigne les gains (paramètres) du contrôleur et un fichier d'initialisation des paramètres pour la simulation avec ce contrôleur. Le Tableau 4.4 donne un exemple des fichiers créés avec comme nom de contrôleur: `ctrlA`.

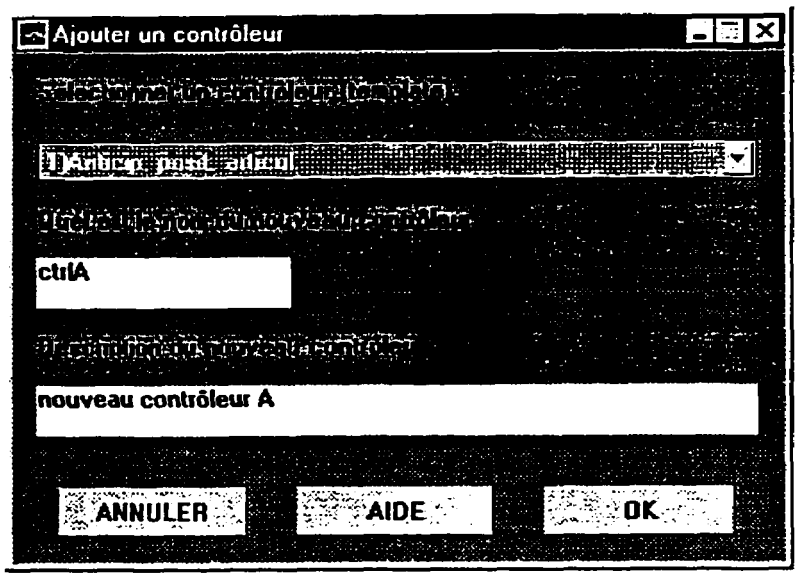


Figure 4.9: Fenêtre d'interface pour ajouter un contrôleur

Tableau 4.4: Exemple de fichiers créés lorsqu'on ajoute un contrôleur (ctrlA)

Nom	Répertoire	Description
`gctrlA`	`\RVSE_FCT\CONTROL`	Assignation des paramètres (gains) du contrôleur
`ictrlA`	`\RVSE_FCT\SIMUL`	Initialisation du contrôleur pour simulation

4.10.1 Simulation avec un nouveau contrôleur

Tous les programmes associés aux simulations (fenêtres Simulink) sont dans le sous répertoire `\RVSE_FCT\SIMUL\`. Les noms de programmes commencent par le nom du contrôleur suivie de trois caractères qui correspondent aux options de simulation et aux types de coordonnées disponibles. Si le programme n'existe pas, une nouvelle fenêtre Simulink est alors créée avec tous les éléments de base. L'utilisateur n'a qu'à compléter la fenêtre. Pour implanter un nouveau contrôleur, on peut utiliser les modules de bases fournis par le logiciel [`Simul=> blocks Simulink`] (fenêtre `rvseblck`).

4.10.2 Essai expérimental avec un nouveau contrôleur

Tel que spécifié dans le chapitre 3, les programmes de contrôle en temps réel sont en langage C. Les programmes sont créés à l'aide du logiciel Borland C. Si le programme exécutable n'existe pas, alors l'utilisateur à la possibilité de créer (générer) le nouveau programme de contrôle en temps réel. Deux fichiers sont alors créés: un fichier IDE (integrated development environment) et un fichier C qui contient les fonctions spécifiques au contrôleur. Avant de pouvoir générer le nouveau programme exécutable, il faut d'abord modifier les deux fichiers[7].

4.11 Résumé de l'utilisation du logiciel

L'usager du logiciel ROVASIMEX peut facilement faire la gestion de plusieurs projets (ajouter, supprimer, enregistrer, récupérer).

Chaque options de menu principal correspond à un module du logiciel ROVASIMEX :

1) [Graph]: Génération de graphiques à partir des fichiers de sorties (`*.TC?, *.C??, *.S??, *.X??`) ou de fichiers de statistiques;
2) [Config]: Création de fichier de configuration (`*.CFG`) d'un système (sélection de fichiers contenant les paramètres de chaque composante, et d'un fichier modèle symbolique (`*.SYM`) correspondant à la configuration) et création de fichiers de modèles symboliques;
3) [Initial]: Création et mise à jour de fichier de paramètres (`*.PAR`) (regroupement des paramètres du système dans un seul fichier) et création de fichiers de modèles numériques (`*.MEX`);
4) [Traject]: Génération et édition de fichier de trajectoire cinématique (`*.TC?`);
5) [Consignes]: Création de fichier de consignes pour effectuer des essais de simulations ou d'expérimentations, en boucle ouverte (`*.C0?`) ou en boucle fermée

[7] Les étapes à franchir (modifications à faire) sont précisées dans le document : Manuel technique du banc d'essai pour l'étude théorique et expérimentale de robots flexibles, École Polytechnique de Montréal.

(*.C_?) avec ou sans: calculs hors ligne de la commande, ajout à la commande d'une compensation de friction, ajustement de la commande pour un contrôleur discret.

6) [Simul]: Simulation à partir d'un fichier de consignes, les résultats de la simulation sont enregistrés dans un fichier (*.S??);
7) [Exper]: Expérimentation à partir d'un fichier de consignes, les résultats sont enregistrés dans un fichier (*.X??) et identification de paramètres à partir de fichiers de résultats;

Chaque projet peut contenir un ou plusieurs fichiers de configurations (*.CFG) d'un même système. À l'initialisation, l'usager ajoute un fichier contenant les paramètres de la charge reliée à l'extrémité (*.EXT), pour créer un fichier de paramètres numériques (*.PAR). À chaque fichier de paramètres correspond un ou plusieurs fichiers de modélisation numériques (*.MEX). Pour des essais en boucle ouverte, on peut directement passer au module de création de fichier consignes (*.C0?). Tandis que pour les essais en boucle fermée, il faut d'abord créer un fichier de trajectoire (*.TC?) avant de pouvoir créer un fichier de consignes (*.C_?). Pour effectuer une simulation du système (*.$??) ou pour réaliser un essai expérimental (*.X??), il faut sélectionner un fichier test. De plus, pour les essais en boucle fermée, il faut sélectionner un contrôleur et un fichier de gains (*.G??). À partir de résultats expérimentaux, on peut exécuter différents programmes d'identification de paramètres (ex.: paramètres de friction, inertie...). Le module graphique permet d'analyser les résultats à l'aide de graphiques (visualisation infographique, temporels, statistiques d'erreurs), Les fichiers de statistiques d'erreur sont crées lorsqu'on trace l'erreur (différence) entre deux signaux contenus dans deux fichiers différents.

5 CONCEPTION ET CARACTÉRISATION DU MONTAGE EXPÉRIMENTAL

Le système expérimental comprenant le robot à membrures flexibles, les capteurs, les actionneurs et les circuits d'interfaces sont décrits dans ce chapitre. Les différentes composantes du système sont choisies après avoir établi les caractéristiques désirées du montage expérimental. Par la suite, tous les paramètres des composantes (identifiés ou provenant des spécifications du manufacturier) sont enregistrés dans des fichiers de données pour être ensuite lus par le logiciel ROVASIMEX.

5.1 Description du montage

5.1.1 Caractéristiques du montage

La **versatilité** est une caractéristique importante du montage. Le manipulateur peut avoir différentes configurations selon qu'il s'agit d'un montage de démonstration ou d'un montage pour des travaux de recherches. De même, pour vérifier le bon fonctionnement du système, il est préférable d'utiliser des membrures rigides. Ceci permet aussi d'obtenir des données pour comparer avec les résultats d'un manipulateur possédant des membrures flexibles.

Le montage pouvant être utilisé dans différentes configurations, la **rapidité** et la **facilité** avec laquelle les **changements** peuvent être **effectués**, sont deux autres caractéristiques du montage.

Des jauges de contraintes sont utilisées pour mesurer la déformation des membrures. Le montage peut aussi être configuré pour identifier les paramètres (facteurs de conversions

des jauges) des différentes membrures flexibles. Par conséquent, la **facilité de calibration des capteurs de déformation** est une autre caractéristique du montage.

Enfin, le montage est **sécuritaire**, le système possède un bouton d'arrêt d'urgence pour limiter les dommages en cas de mauvais fonctionnement.

5.1.2 Choix des composantes

Les principaux avantages des composantes du montage expérimental sont énoncés pour en justifier le choix, tout en prenant en considération les inconvénients.

Le robot manipulateur conçu pour le projet de recherche possède deux membrures flexibles. Le choix des membrures a été réalisé de façon à bien mettre en évidence l'effet de la flexibilité. La déformation maximale (cas statique) des membrures est fixée à environ 5% lorsque les membrures du manipulateur sont en position horizontale. Les membrures sont des tiges en acier "low carbon" qui sert normalement de matériel de base pour la fabrication d'outils. Les tiges sont lisses (polies) et de masse uniforme. Le principal inconvénient est que la longueur disponible des tiges est limitée à environ 62 cm (2 pieds). Pour le montage à réaliser cette longueur est suffisante.

Des jauges de déformation collées sur les membrures sont utilisées pour mesurer les déformations des membrures. Le signal de sortie est proportionnel à la courbure. Ce signal étant de très faible intensité, il faut donc l'amplifier tout en filtrant le bruit. Des amplificateurs (conditionneurs) conçus pour ce genre de signal sont utilisés. L'influence de la température sur le signal est un autre inconvénient des jauges de déformation. Pour réduire l'influence de la température, on utilise deux jauges connectés en demi pont. Les jauges sont installées de chaque coté de la membrure, i.e. une jauge en compression l'autre en tension.

Il y a eu deux conceptions (versions) des articulations du robot. Pour la première version, les articulations étaient en acier et de forme cylindrique. Le poids (estimé) des articulations était trop grand. C'est pour cette raison que les trois articulations rotoïdes sont fabriquées en aluminium. Cependant, l'aluminium étant moins rigide que l'acier, il faut donc en tenir compte lors de la conception mécanique des articulations.

Les actionneurs du robot sont des moteurs à courant continu avec réducteur harmonique. Ce type de moteur est très compact. Les réducteurs harmoniques incorporés aux moteurs font en sorte qu'il n'y a plus de zone morte (pas de jeux d'engrenage), par contre il y a une augmentation de la force de friction. De plus, la modélisation d'une telle transmission doit inclure des effets d'élasticité (lien élastique).

En plus d'être muni de réducteurs harmoniques, les moteurs sélectionnés pour le montage ont des encodeurs optiques directement reliés à l'arbre. Le signal généré par les encodeurs optiques est composé de deux trains d'impulsions en quadrature. Pour déterminer la direction et la position des encodeurs, il faut un circuit de comptage d'impulsions[1].

Pour obtenir l'information des compteurs d'impulsions, des ports parallèles sont utilisés. Les autres cartes d'interface nécessaire au système sont des cartes avec des convertisseurs (A/N et N/A).

Pour contrôler les moteurs, les signaux provenant des convertisseurs N/A doivent être amplifiés. Avec un robot manipulateur, le contrôle du couple des actionneurs est plus pertinent que le contrôle de la vitesse. Des amplificateurs de courant sont utilisés plutôt que des amplificateurs de tension, puisque le courant d'un moteur est proportionnel au couple développé par le moteur. Les amplificateurs utilisés dans le montage sont de type

[1] Pour plus de détails sur le circuit de comptage d'impulsion consulter le mémoire de Steeve Miller

PWM (pulse width modulation) et ils ont un gain fixe. De plus, les amplificateurs PWM sont moins coûteux et moins massifs que des amplificateurs linéaires. Étant donné que les signaux de commande n'utilisent qu'une partie de la plage de fonctionnement des amplificateurs, des diviseurs de tensions à la sortie des convertisseurs N/A sont ajoutés, ce qui augmente la résolution du système.

Pour terminer, un panneau vertical parallèle aux déplacements des membrures du robot manipulateur est installé. En attachant à la dernière articulation un support avec deux crayons, ceci permet de tracer la position de l'extrémité du robot. De plus, l'écart entre les courbes tracées par les deux crayons permet de vérifier le contrôle de l'orientation de l'extrémité.

Le tableau ci-dessous résume les avantages et les inconvénients pour chaque type de composantes du montage expérimental.

Tableau 5.1: Avantages et inconvénients des composantes du robot

Composants	Avantages	Inconvénients
Amplificateurs de courant PWM	contrôle du couple	gain fixe
Actionneurs moteurs DC, avec réducteurs harmoniques	réducteur sans zone morte	lien élastique force de friction
Capteurs de position d'articulations encodeurs optiques	incorporés aux moteurs	circuit de comptage d'impulsions
Articulations rotoïdes en aluminium	faible masse, inertie	faible rigidité

Composants	Avantages	Inconvénients
Membrures flexibles barre rectangulaire en acier d'outils (low carbon)	lisse, masse uniforme	dimension limitée
Capteurs de flexibilité jauges de contraintes avec conditionneur de signal (ARIES)	mesures directes de courbure	signaux faibles, influence de la température
Cartes d'interfaces (N/A, A/N, ...): NI compteur d'impulsions ports parallèles: CIO	filtre incorporé	séquence de transfert un seul port 24bits
Capteur extéroceptif panneau vertical, parallèle au manipulateur	traçage de la position de l'extrémité et vérification du contrôle de l'orientation	ajout de frottement

5.2 Identification des paramètres

5.2.1 Cartes d'interface

Les programmes d'acquisitions de données et de contrôle en temps réel ont besoin de cartes d'interface.

Les cartes sont divisées en trois catégories:

1) avec entrées analogiques (AI);
2) avec ports parallèles 24 bits (DIO);
3) avec sorties analogiques.

Les fonctions de base des programmes en temps réel permettent d'utiliser deux cartes pour chacune des trois catégories. Une même carte peut avoir à la fois des entrées analogiques et des sorties analogiques. Les caractéristiques (type de cartes, fentes, l'adresse de base, etc.) des cartes d'interfaces sont définies dans le fichier `BCFGRMAN.H` (`\RVSE_CTR`). Pour le projet actuel, il y a quatre cartes d'interfaces installées dans l'ordinateur. Deux cartes d'entrées/sorties analogiques de la compagnie National Instrument (AT-MIO-16-F5, AT-MIO-L9) et deux cartes de ports parallèles de Computer Boards (CIO-DIO24). Les cartes de ports parallèles sont utilisées pour lire et contrôler les cartes de lecture d'encodeurs (compteurs d'impulsions).

Plusieurs versions de programme exécutable peuvent être crée lors de la compilation. Il suffit d'activer des constantes dans le fichier `CTRL_ACQ.C` (`\RVSE_CTR`). Par exemple, en définissant la constante *`TST_TEMPS_EXEC`* égale à 1, le programme en temps réel génère[2] un signal permettant de vérifier à l'aide d'un oscilloscope le temps d'exécution des trois principales étapes de la routine d'interruption. Les principales étapes sont: l'acquisitions de données, le calcul la commande aux actionneurs (contrôle) et le transfert des données dans des vecteurs (matrices). Au début de la routine, le signal de sortie passe de 0 à +5 volts, à la fin de l'acquisition de données le signal change de signe (-5 volts) et demeure constant jusqu'à la fin du calcul de la commande, ensuite le signal redevient +5 volts pendant le transfert des données. Lorsque la routine d'interruption est terminée, le signal de sortie devient nul (0 V).

Les temps d'exécution des deux dernières étapes varient selon l'état du système:

1) en attente de l'exécution d'un suivi de la trajectoire désirée;
2) pendant le suivi de trajectoire;

[2]Lors de ces essais, il faut éteindre l'alimentation ou déconnecter les actionneurs puisque la commande (sorties du contrôleur) est calculée mais n'est pas envoyée.

3) pendant le temps de stabilisation, i.e. le temps d'enregistrement de données supplémentaires à la fin du suivi de la trajectoire désirée.

Les temps les plus courts sont lorsque le système est en attente. Il n'y a pas de lecture de trajectoire désirée, ni de transfert des données. Le cas contraire est lorsque le système est en suivi de trajectoire. Les temps sont les plus longs car il y a lecture de la trajectoire et transfert des données. Le Tableau 5.2 présente les temps d'exécutions des trois étapes en utilisant le programme (CA_PA.EXE) de contrôle avec commande anticipative et PID sur l'erreur de la position des articulations. Les résultats correspondent à un essai avec un robot ayant trois articulations et deux membrures flexibles. Chaque membrure flexible a trois capteurs de flexibilité. Au total le système doit faire la lecture de trois encodeurs et de six entrées analogiques. Le temps total d'exécution est alors inférieur à 0,5 ms.

Tableau 5.2: Temps d'exécution de la routine d'interruption en temps réel

État du système	Temps d'exécution (μs)			
	Acquisition	Contrôle	Transfert	Total
en attente	195	150	3	348
suivi trajectoire	195	185	32	412
stabilisation	195	150	32	377

5.2.2 Amplificateurs

Trois amplificateurs de courant COPLEY 303 avec **gain fixe de 1.21 A/V** sont utilisés. La commande (entrées des amplificateurs) provient d'une carte d'interface ayant des convertisseurs Numériques/Analogiques. Pour augmenter la résolution de la commande, des diviseurs de tensions sont ajoutés à l'entrée des amplificateurs. Le Tableau 5.3 présente le nom des fichiers de définitions de paramètres des amplificateurs, le numéro des cartes d'interfaces (1 ou 2) et

Tableau 5.3: Paramètres des amplificateurs

Fichier	Canal # de la carte AO #	diviseur de tension
COP303_1	0, 1	0,425
COP303_2	1, 1	0,125
COP303_3	0, 2	0,05

du canal de sortie analogique (AO: Analogic Output) auquel l'amplificateur est connecté, ainsi que la valeur de chacun des diviseurs de tensions.

5.2.3 Actionneurs

Les actionneurs sont munis de réducteurs harmoniques. Pour inclure l'élasticité aux modèles, il faut définir les paramètres de rigidité et d'amortissement des réducteurs. De plus, le facteur de réduction permet d'inclure l'effet gyroscopique des actionneurs sur le système. Le Tableau 5.4 présente les paramètres (provenant du manufacturier) des réducteurs harmoniques.

Tableau 5.4: Paramètres des réducteurs harmoniques

Fichier	#	réducteur	rigidité	amortissement	inertie
var. Matlab		*n_r*	*act_rig*	*act_amort*	*act_Ir(3,3)*
var. Maple		*red#*	*Kr#*	*br#*	*Ir#zz*
unités			N-m/rad	N-m /rad/s	kg-m²
RH146002	1	50	1694,7	0,0140875	8,66223e-006
RH8 6006	2	50	395,43	0,0000637	1,99143e-006
RH5	3	80	66,6582	0,00001998	2,53527e-007

L'inertie des actionneurs (Tableau 5.5) comprend l'inertie du réducteur rapportée au niveau du moteur plus l'inertie du rotor, auquel on ajoute l'inertie de l'encodeur.

Tableau 5.5: Inerties des actionneurs

Fichier	#	inertie totale	inertie réducteur + rotor	inertie encodeur
var. Matlab		*act_Ir(3,3)*		
var. Maple		*Ir#zz*		
unités		kg-m²	kg-m²	kg-m²
RH146002	1	8.6622e-006	8,1619e-006	0,5003e-006
RH8 6006	2	1,9914e-006	1,4911e-006	0,5003e-006
RH5	3	2,5353e-007	2,4372e-007	0,0981e-007

Pour calculer la commande anticipative ou pour simuler le système avec friction, il faut déterminer les coefficients du polynôme de friction dynamique. Le logiciel ROVASIMEX permet d'identifier les coefficients à partir de fichiers de résultats d'un essai en boucle ouverte avec une onde rampe (`RMP*.C01`). Si l'accélération est très faible, alors l'effet d'inertie est négligeable et le couple appliqué compense la friction. La fonction Matlab *`polyfit`* est utilisée pour calculer les coefficients du polynôme (ordre deux). Sachant que le couple est proportionnel au courant, les données utilisées par la fonction *`polyfit`* sont le courant et la vitesse de l'actionneur. Par conséquent, les coefficients identifiés sont fonctions du courant d'entrée donc indépendants du gain de l'actionneur. Les paramètres de friction des actionneurs identifiés sont inscrits dans le Tableau 5.6. Il est à noter que les paramètres de friction utilisés pour calculer les commandes anticipatives et pour effectuer les simulations sont en fonction du couple. Par conséquent, dans les fichiers de définition de paramètres des actionneurs, les paramètres identifiés de friction sont multipliés par le gain des actionneurs.

Tableau 5.6: Paramètres identifiés de frictions actionneurs

Nom de fichier	#	friction statique	coefficient du polynôme fonction de la vitesse		
			a_2	a_1	a_0
unités		A	A/(rad/s)^2	A/(rad/s)	A
RH146006	1	0,2	-0,0044	0,115	0,17
RH8 6002	2	0,11	-0,00115	0,035	0,095
RH5	3	0,105	-0,00050	0,0180	0,0985

Le gain d'un actionneur (N-m/A) peut être identifié à l'aide du logiciel. Il suffit de réaliser des essais en boucle ouverte pour différentes charges à l'extrémité de l'actionneur. Si le centre de masse de la charge n'est pas sur l'axe de rotation de l'actionneur, alors le couple appliqué égale le couple de friction additionné au couple fonction de l'accélération gravitationnelle. Les essais en boucle ouverte utilisent des signaux de type rampe donnant une très faible accélération à l'actionneur. L'inertie est alors négligeable et le couple

dépend de la position (effet de la gravité) et de la vitesse (effet de la friction) de la charge. Sachant le courant appliqué à l'actionneur, il est alors possible de calculer le gain de l'actionneur (mesuré à la sortie du réducteur). Le Tableau 5.7 compare les gains obtenus des spécifications du manufacturier avec les gains identifiés par des essais en boucle ouverte.

Tableau 5.7: Gains des actionneurs, spécifications du manufacturier et paramètres identifiés

fichier	#	gain	
		manuf.	Identifié
unités		(N*m)/A	(N*m)/A
RH14006	1	2,92	2,45
RH8002	2	2,10	1,82
RH5	3	1,11	1,11

5.2.4 Capteurs

Des encodeurs optiques sont utilisés comme capteurs de position des articulations. Le Tableau 5.8 présente le nom des fichiers de paramètres, le numéro de carte d'interface DIO (1 ou 2) et le numéro de canal (port de 24 bits) ainsi que la résolution de chacun de ces capteurs. Pour lire la position des encodeurs, des compteurs d'impulsions (puce: HCTL 2020) sur un circuit d'interface sont utilisés. Comme on peut le voir à la Figure 5.1, il y a parfois des erreurs (glitch) de lectures. Ces erreurs ont une amplitude d'environ 255 impulsions. Le problème survient lorsque le nombre d'impulsions mesurées correspond à un changement de l'octet le plus significatif (HSB) du compteur d'impulsions (de 0x00FF à 0x0100, 0x0100 à 0x00FF, 0x01FF à 0x0200, etc.). Pour remédier à la situation, lorsque l'écart (en valeur absolue)

Tableau 5.8: Paramètres des capteurs de position

Nom de fichier	#	Canal # de la carte DIO #	impulsions/ tour
ENC_1000	1	0, 1	1000
ENC_500	2	1, 1	500
ENC_100	3	0, 2	100

entre deux mesures successives est supérieur à 200 impulsions, il faut compenser cette erreur en additionnant ou en soustrayant 256 à la valeur mesurée.

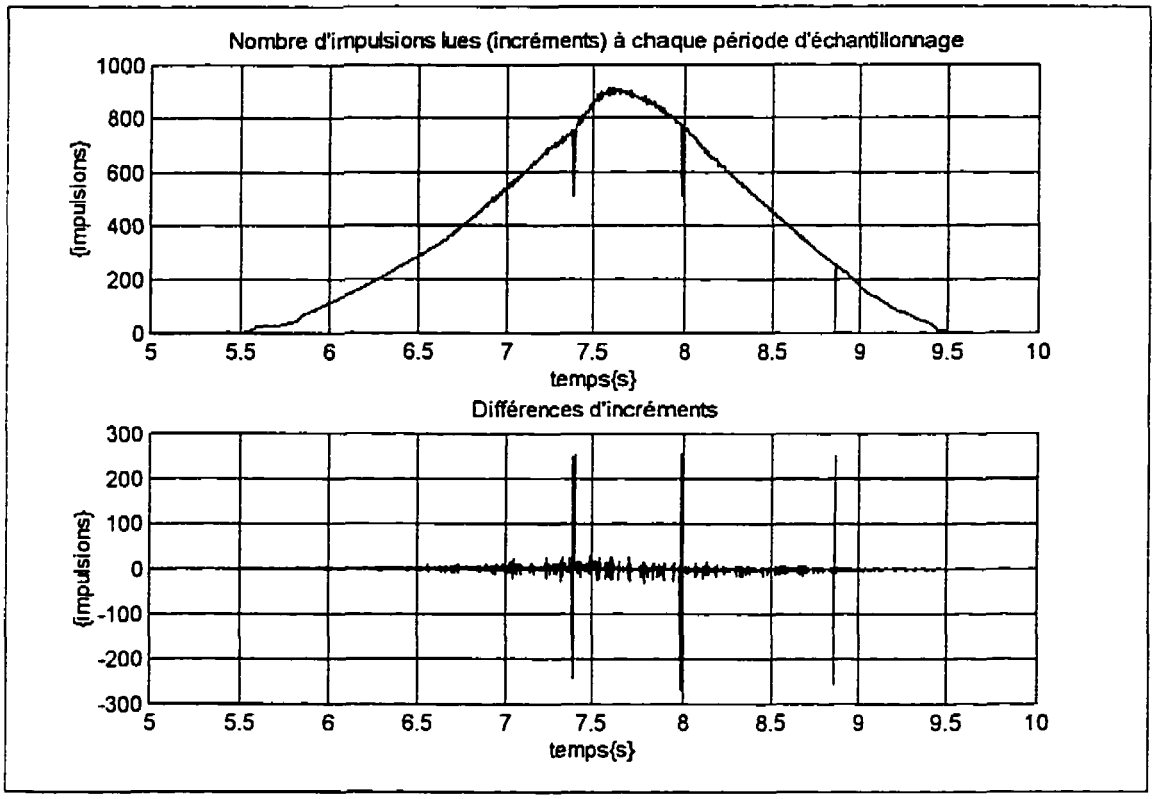


Figure 5.1: Erreur de lecture des encodeurs optiques

Le système expérimental n'a pas de capteur de vitesse et ni de capteur de force aux articulations. Lors de la création d'un fichier de configuration du système, il faut choisir les fichiers NO_TACH.TAC et NO_FRC.FRC. La vitesse des articulations est calculée à partir des mesures de positions et de la période d'échantillonnage. Deux filtres passe-bas (Butterworth d'ordre 1) sont utilisés pour améliorer la mesure de la vitesse. Un premier filtre sur la position et le second sur la vitesse. Tout comme la vitesse, l'accélération est calculée à partir de la vitesse. L'effet des filtres est plus évident lorsqu'on trace le graphique de l'accélération en fonction du temps. La Figure 5.2 présente les résultats de l'acquisition de données sans filtre.

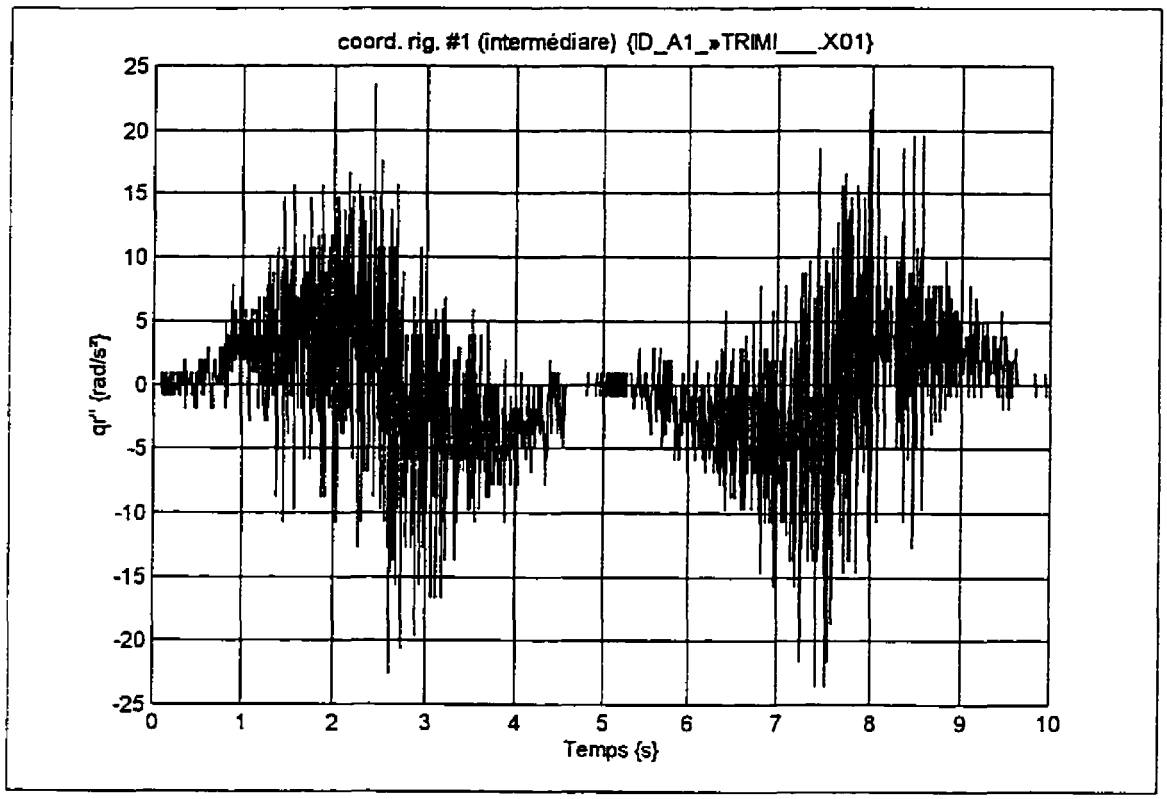


Figure 5.2: Acquisition de données sans filtrage

Tandis que la Figure 5.3 présente les résultats après filtrage. Les données de ces graphiques proviennent d'un essai en boucle ouverte avec une version modifiée du programme d'acquisitions dont les filtres passe-bas ont été désactivés. Les données sont filtrées par le logiciel avant de tracer les courbes. Les paramètres du filtre sont les mêmes que ceux utilisés dans le programme (normal) d'acquisition de données en temps réel.

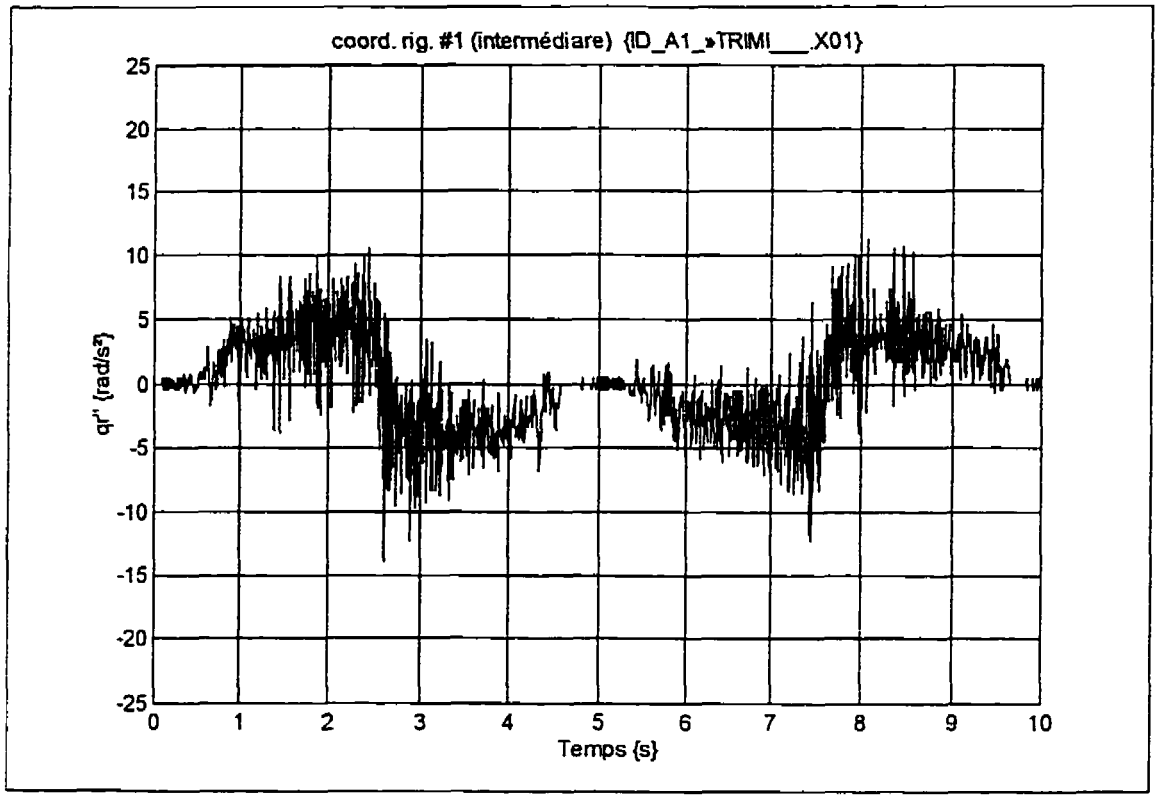


Figure 5.3: Résultats après filtrage des données

5.2.5 Articulations

Les trois articulations rotoïdes du robot ont été conçues[3] et fabriquées spécialement pour ce projet. Pour bien modéliser le robot, il faut déterminer plusieurs paramètres des articulations.

Les paramètres sont séparés en deux groupes (parties de l'articulation):

1) les paramètres de l'articulation reliés à la pièce précédente;
2) les paramètres de la partie mobile de l'articulation.

[3] Le document « Manuel technique du banc d'essai pour l'étude théorique et expérimentale de robots flexibles » contient les dessins techniques des articulations.

La pièce précédente peut être une membrure ou la partie mobile d'une autre articulation. Pour chacun des deux groupes, les paramètres des articulations sont: la distance entre le référentiel du groupe et le référentiel du prochain groupe (pièce suivante), la masse, le centre de masse et la matrice d'inertie. Le logiciel Lotus[4] a été utilisé pour calculer les paramètres des articulations (Tableau 5.9 et

Tableau 5.10). Chaque partie de l'articulation est décomposée en forme de base (prisme, cylindre, etc.). Les paramètres sont calculés dans les référentiels des formes de base pour ensuite être transférés dans le référentiel du groupe. Les centres de masse de chaque groupe sont alignés, la matrice d'inertie globale de l'articulation ne contient alors que l'inertie autour de l'axe Z. Les paramètres sont définis dans des fichiers Matlab (ART*.M) dont le quatrième caractère indique le numéro de l'articulation. Pour diminuer l'effort (statique) demandé au premier actionneur, la première articulation peut avoir un contre poids. Le contre poids est composé d'une vis (un pouce de diamètre) et de boulons s'y rattachants. Les caractères CP du nom de fichier indiquent que l'articulation a un contre poids et le caractère qui suit indique le nombre de boulons sur la vis de contre poids. Les paramètres des articulations inclus les paramètres du contre poids. Les articulations peuvent être assemblées pour un système à membrures flexibles (F) ou à membrures rigides (R).

[4] Les feuilles de calculs Lotus sont dans le document « Manuel technique du banc d'essai pour l'étude théorique et expérimentale de robots flexibles », École Polytechnique de Montréal

Tableau 5.9: Paramètres des articulations reliées aux pièces qui les précèdent

		Charge reliée à la pièce précédente			
fichier	#	distance	masse	cdm	inertie
var. Matlab		*P_art*	*M_prec*	*cdm_prec*	*I_prec*
var. Maple		*pc#x*	*mc#*	*ec#x*	*Ic#zz*
unités		m	kg	m	kg-m²
ART2FF	2	0,045	0,361	0,0356	0,000121
ART2RR	2	0,045	0,361	0,0356	0,000127
ART3F	3	0,035	0,138	0,0261	0,0000271
ART3R	3	0,035	0,138	0,0261	0,0000303

Tableau 5.10: Paramètres de la partie mobile des articulations

		Partie mobile de l'articulation			
fichier	#	distance	masse	cdm	inertie
var. Matlab		*P_suiv*	*M_suiv*	*cmd_suiv*	*I_suiv*
var. Maple		*pa#x*	*ma#*	*ea#x*	*Ia#zz*
unités		m	kg	m	kg-m²
ART1	1	0,0375125	1,242	0	0,00059
ART1CP	1	0,0375125	3,109	-0,0761	0,0212
ART1CP1	1	0,0375125	3,502	-0,0928	0,0291
ART1CP2	1	0,0375125	3,895	-0,1029	0,0328
ART2FF	2	0,067	0,192	0,0267	0,000129
ART2RR	2	0,067	0,192	0,0267	0,000132
ART3F	3	0,051	0,071	0	0,0000264
ART3R	3	0,051	0,071	0	0,0000264

5.2.6 Membrures

Les membrures du robot sont des tiges rectangulaires. Pour les essais avec membrures rigides, les membrures sont installées de façon à avoir le coté le plus large (rigidité la plus élevée) dans le plan du déplacement. Les mêmes membrures sont utilisées pour les essais avec des membrures flexibles. Il suffit simplement d'installer les membrures avec le coté le plus mince dans le plan du déplacement (rotation de pi/2 autour de l'axe neutre "X" des membrures). Les paramètres de modélisation des membrures sont toutefois différents selon le type de membrure (rigide ou flexible). Les deux prochains tableaux contiennent les paramètres des membrures rigides et des membrures flexibles.

Tableau 5.11: Paramètres des membrures rigides

fichier	#	masse	longueur	inertie
var. Matlab		*M_memb*	*L*	*I_memb(3,3)*
var. Maple		*mm#*	*L#*	*Im#zz*
unités		kg	m	kg-m²
R3205	1	0,4486	0,5620	0,0118
R3205 1F	1	0,4998	0,5620	0,0132
R3205 2F	1	0,5729	0,5620	0,0151
R2402	2	0,0671	0,2760	0,4278e-3
R2402 F	2	0,0970	0,2760	0,6186e-3

Tableau 5.12: Paramètres des membrures flexibles

Fichier	#	Densité linéaire	Longueur	Rigidité	Amortissement	Élément d'inertie
var. Matlab		*rho_l*	*L*	*Eiz*	*Ke*	*Idm(3,3)*
var. Maple		*rho#*	*L#*	*EI#zz*	*Ke#*	*Idm#zz*
unités		kg/m	m	N-m²	N-m²/s	kg-m
F0532	1	0,7982	0,5620	25,8017	0,0001	0,0104
F0532 1F	1	0,8892	0,5620	25,8017	0,0001	0,0116
F0532 2F	1	1,0194	0,5620	25,8017	0,0001	0,0133
F0224	2	0,2430	0,2747	1,2769	0,0001	0,0017
F0224 F	2	0,3519	0,2747	1,2769	0,0001	0,0024

Pour vérifier le comportement réel des membrures flexibles et pour déterminer la position et l'orientation de l'extrémité, il faut utiliser des capteurs de flexibilité. Tel que décrit dans le chapitre théorique, des jauges de déformation sont utilisées pour déterminer les déformations des membrures. À partir d'essais d'acquisitions de données (ACQ*.X00), le logiciel permet d'identifier ou de vérifier les facteurs de conversions des mesures de flexibilité. Les paramètres des capteurs de flexibilité sont inscrits dans le Tableau 5.13.

Tableau 5.13: Paramètres des capteurs de flexibilité

description	Positions normalisées			Facteurs de conversion			Canal # de la carte AI #			Tension excitation
unités				1/m						V
memb.#1	0,0091	0,4994	0,9884	2,86e-4	2,80e-4	2,85e-4	0,1	1,1	2,1	9
memb.#2	0,0189	0,5000	0,9814	7,75e-4	7,30e-4	7,70e-4	3,1	4,1	5,1	8

En plus de calculer les facteurs de conversion, le logiciel permet de comparer les résultats obtenus à l'aide de formules théoriques (poutre encastrée) avec les résultats obtenus avec les fonctions de forme d'un fichier de modélisation numérique[5]. Pour le robot planaire vertical, il n'y a que des déformations v (dans le plan XY).

Deux fonctions de formes sont utilisées pour chaque membrure:

$$^{v_i}\varphi_1(x) = 3\left(\frac{x}{L}\right)^2 - 2\left(\frac{x}{L}\right)^3, \quad ^{v_i}\varphi_2(x) = \frac{3}{2}\left(\frac{x}{L}\right)^2 - \frac{1}{2}\left(\frac{x}{L}\right)^3$$

Comme on peut le constater aux Figure 5.4 et Figure 5.5, la déformation et la pente à l'extrémité de la membrure calculée avec les fonctions de formes sont exacts, malgré le fait d'un écart le long de la membrure. Par contre, la courbure (Figure 5.6) à l'extrémité de la membrure qui est calculée avec fonctions de formes est différente de celle calculée avec la formule de poutre encastrée. Parce que les dérivées secondes des fonctions de forme (B-spline cubiques) deviennent des équations du premier ordre alors que l'équation de courbure pour poutre encastrée est du deuxième ordre. Par conséquent, lors du contrôle de la déformation des membrures (pour réduire les oscillations de la membrure en régime permanent), il est préférable d'utiliser des vecteurs de gains petits $\left(\mathbf{k}_{p_j}, \mathbf{k}_{d_j}\right)$ puisque les valeurs désirées de courbures (calculées avec les fonctions de formes) seront approximatives. De plus, pour diminuer le bruit de mesures des capteurs de flexibilité, les signaux sont filtrées à l'aide de filtres passe-bas Butterworth du premier ordre.

[5]Pour vérifier les résultats des essais avec différentes charges, le fichier de modélisation numérique doit être généré en laissant libre les paramètres de la charge à l'extrémité (masse et centre de masse).

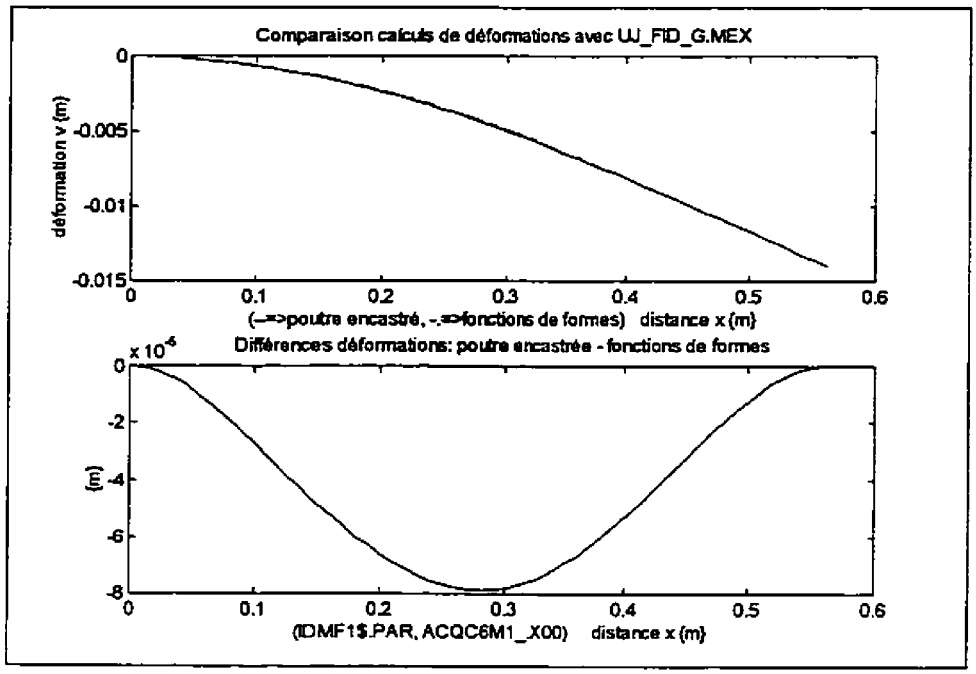


Figure 5.4: Déformations d'une membrure, calculées avec fonctions de formes et avec formule de poutre encastrée

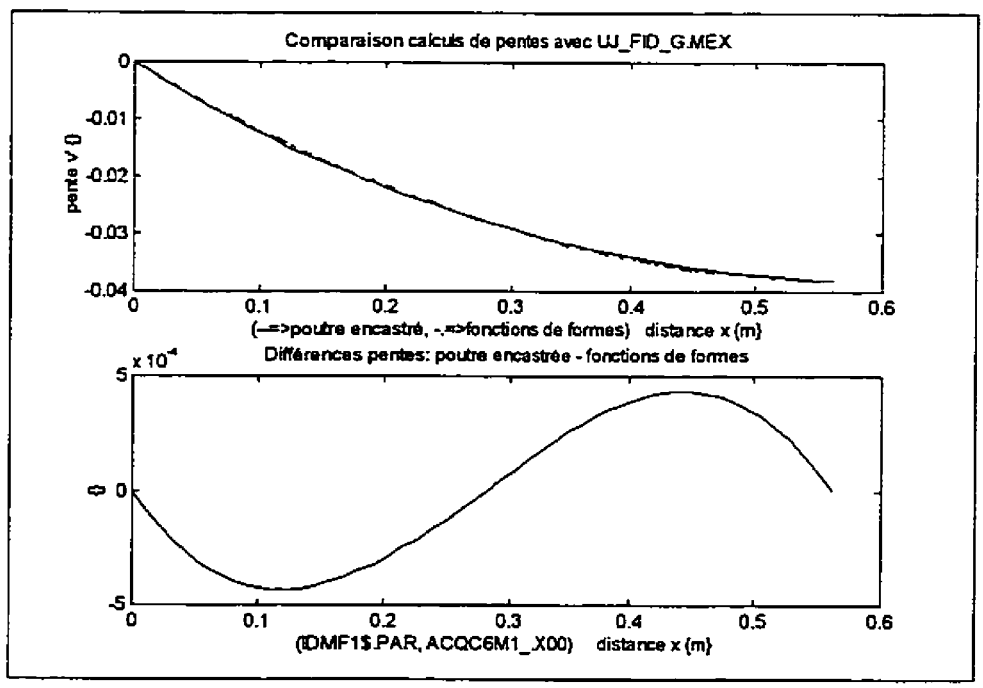


Figure 5.5: Pentes d'une membrure, calculées avec fonctions de formes et avec formule de poutre encastrée

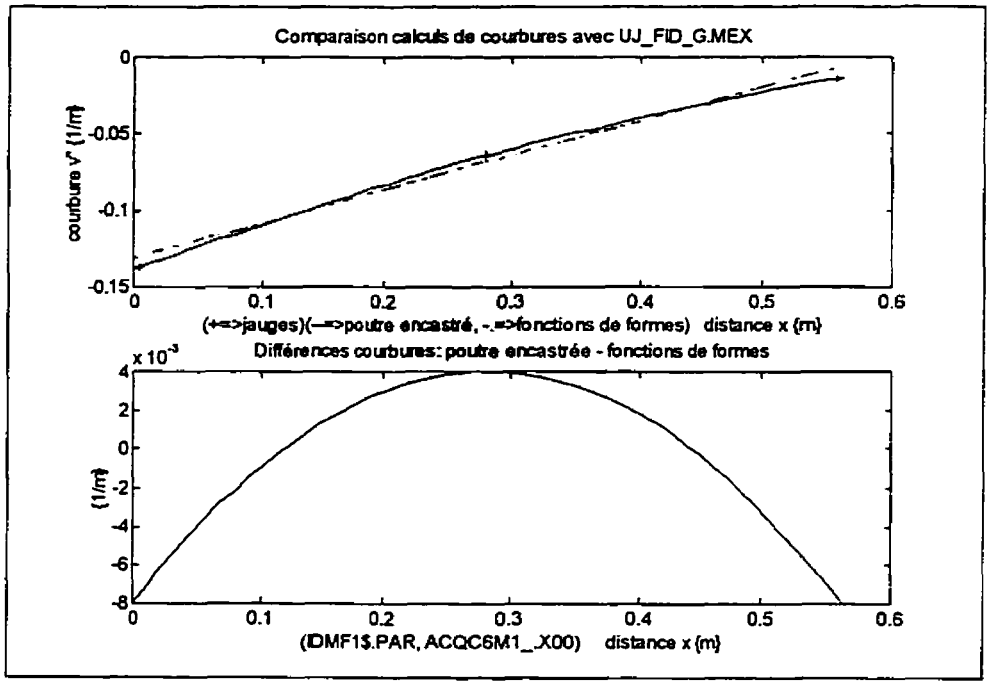


Figure 5.6: Courbures d'une membrure, calculées avec fonctions de formes et avec formule de poutre encastrée

Après plusieurs essais avec différentes charges, on remarque que les signaux d'erreurs (graphiques de différences) entre les résultats avec formules de poutre encastrée et les résultats des fonctions de formes sont indépendants de la charge (couple et force) à l'extrémité de la membrure.

Tel que démontré aux figures suivantes, si on augmente le nombre de modes (trois fonctions de forme: B-splines cubiques) l'erreur diminue. La courbure (Figure 5.7) est alors exprimée par deux segments de droite dont l'intersection est au milieu de la membrure.

Tableau 5.14: Erreur maximum entre résultats calculés avec fonctions de forme et avec formules de poutre encastrée

| Erreur |	2 fonctions de formes	3 fonctions de formes
déformation	8.0e-5	5.0e-6
pente	4.5e-4	5.5e-6
courbure	8.0e-3	2.0e-3

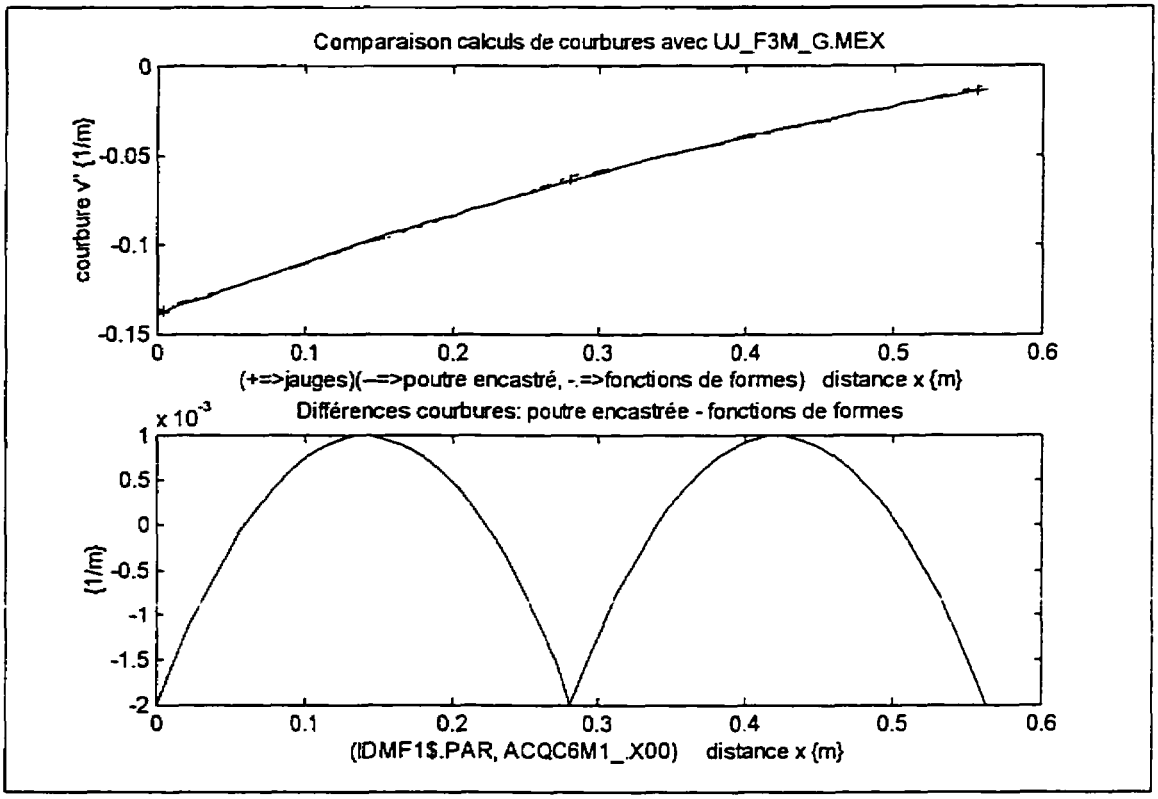


Figure 5.7: Courbures d'une membrure, calculées avec trois fonctions de formes et avec formule de poutre encastrée

5.3 Conclusion

Les principales caractéristiques du montage sont la versatilité, la facilité de changement de configuration du système.

Les composantes pour la configuration normale du robot sont:

- trois articulations en aluminium;
- trois actionneurs (moteur DC) avec réducteur harmonique;
- trois encodeurs optiques reliés directement aux actionneurs;
- trois amplificateurs de courant PWM;
- deux membrures flexibles ou deux membrures rigides;
- trois jauges de déformations collés sur chacune des deux membrures flexibles;
- deux cartes d'acquisitions de données (entrées/sorties analogiques);
- deux cartes de port parallèles;
- deux cartes de compteurs d'impulsions;
- un panneau vertical servant de table à dessin pour tracer la position de l'extrémité du robot;
- deux crayons feutres attaché à l'extrémité du robot.

6 SIMULATION ET EXPÉRIMENTATION

Dans ce chapitre, les résultats de simulations et d'essais expérimentaux sont présentés et brièvement analysés. Tout d'abord, les trajectoires désirées sont décrites, pour ensuite donner un résumé de la démarche utilisée pour déterminer les gains du contrôleur. Malgré la grande quantité d'essais intermédiaires effectués, on traite seulement des résultats pour deux configurations de robot: un robot planaire vertical à trois articulations possédant deux membrures rigides (V3A2MR_.PAR), et le même robot mais avec deux membrures flexibles (V3A2MF_.PAR). Pour terminer le chapitre, la comparaison des résultats des deux configurations de robot est présentée. Il est à noter que tous les essais ont été effectués en négligeant l'élasticité des actionneurs (avec réducteurs harmoniques). Lors de futur travaux de recherche, l'effet de l'élasticité devrait être incluse lors du calcul de la commande anticipative et des corrections pour obtenir les coordonnées désirées. Ceci pourrait probablement améliorer les résultats.

6.1 Description des trajectoires désirées

Deux catégories de trajectoires sont générées, des trajectoires dans l'espace des variables d'articulations et des trajectoires dans l'espace de la tâche (plan vertical XY). La liste complète des trajectoires est dans le document « Manuel technique du banc d'essai pour l'étude théorique et expérimentale de robots flexibles ». Les trajectoires dans l'espace des variables d'articulations sont utilisées pour ajuster les gains du contrôleur et pour vérifier sa robustesse. Les trajectoires dans l'espace de la tâche sont utilisées pour vérifier la position de l'extrémité du robot.

Le temps d'exécution d'une trajectoire dépend des paramètres maximum de vitesse et d'accélération. Pour les essais, trois vitesses d'exécution ont été utilisées (Tableau 6.1).

Puisqu'on ne peut pas dépasser la vitesse maximum des actionneurs, pour augmenter la vitesse d'exécution de vitesse moyenne à vitesse rapide, on augmente la valeur maximum de la dérivée d'accélération et de l'accélération. Le temps pour atteindre la vitesse maximum est alors réduit. La période d'échantillonnage de toutes les trajectoires a été fixé à 4 ms.

Tableau 6.1: Vitesses d'exécutions des trajectoires

Vitesses d'exécution	dérivée d'accélération	accélération	vitesse
rapide	16*a_max	2*a_max	v_max
moyenne	8*a_max	a_max	v_max
lente	2*a_max	a_max /4	v_max/4

6.1.1 Trajectoires dans l'espace des variables d'articulations

Les paramètres maximum des articulations sont fonctions de la configuration du système. Le Tableau 6.2 présente les paramètres limites des articulations[1] pour chacune des deux configurations du robot.

[1] Lorsque les trajectoires ont été générées (version préliminaire du logiciel), au lieu d'utiliser le calcul vectoriel, le calcul des accélérations maximum a été effectué de la dernière articulation en allant vers la première articulation. Présentement, la situation a été corrigé, le logiciel ROVASIMEX calcule maintenant les accélérations maximum en utilisant le modèle numérique avec les capacités maximum des actionneurs : $\ddot{\mathbf{q}}_{max} = \mathbf{M}_{nl}^{-1}\left(\Gamma_{nl}\left(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u}_{max}\right)\right)$.

Tableau 6.2: Paramètres limites des trois articulations

Configuration à membrures	# art.	Accélération maximum	vitesse maximum	position maximum
		a_max	*v_max*	
		(rad/s²)	(rad/s)	rad
rigides	1	6,23	6,98	
	2	9,28	11,87	±1,91986
	3	357,80	12,04	
flexibles	1	6,19	6,98	
	2	9,35	11,87	±2,44346
	3	357,80	12,04	

Le Tableau 6.3 décrit les noms de fichiers de trajectoires dans l'espace des variables d'articulations. Le premier caractère du nom de fichier indique si la trajectoire est dans la zone stable ou dans la zone instable (ZIARATI, K., 1993). Le deuxième caractère est fonction de la vitesse de la trajectoire, tandis que troisième caractère indique la configuration du robot.

Tableau 6.3: Description des noms de fichiers de trajectoires dans l'espace des variables d'articulations

#	Caract.	Description
1	V	dans la zone stable
	I	dans la zone Instable
2	R	à vitesse Rapide
	M	à vitesse Moyenne
	L	à vitesse Lente
3	R	avec membrures Rigides
	F	avec membrures Flexible

6.1.2 Trajectoires dans l'espace de la tâche (plan vertical XY)

Les angles limites de la deuxième articulation du robot dépendent du type de membrure. Par conséquent, l'espace de travail et les paramètres maximum sont fonctions du type de

membrure. Le Tableau 6.4 présente les paramètres maximum des trajectoires pour chacune des deux configurations de robot.

Tableau 6.4: Paramètres maximum des trajectoires dans l'espace de la tâche

Configuration à membrures	accélération tangentielle	vitesse tangentielle
	a_max	*v_max*
	(m/s^2)	(m/s)
rigides	3,2	0,6
flexibles	2,4	0,6

Quatre types de trajectoires sont générées: des lignes, des polygones, des cercles et des arcs de cercle. Pour ce projet, les polygones générés sont des carrés, les arcs de cercle sont des cercles dont on fait le tour deux fois. Les noms des fichiers de trajectoires sont composés de trois caractères dont ont peut voir la signification dans le Tableau 6.5. La première colonne du tableau indique la position (premier, deuxième, troisième) d'un caractère dans un nom de fichier.

Tableau 6.5: Description des noms de fichiers de trajectoires (extrémité) dans l'espace de la tâche

#	Caract.	Description
1	L	Ligne
	P	Polygone
	C	Cercle
	A	Arc de cercle
2	R	à vitesse Rapide
	M	à vitesse Moyenne
	L	à vitesse Lente
	N	déplacement Négatif (sens contraire)
	I	dans la zone Instable (Y>0)
	S	dans la zone Stable (Y<0)
	C	Coude vers le haut ($q_{r2} \leq 0$)
	H	ligne verticale vers le Haut
	B	ligne verticale vers le Bas
	G	dans le plan Gauche (X<0)
	D	dérivée d'accélération Discontinue (échelon)
3	R	avec membrures Rigides
	F	avec membrures Flexible

Puisque les angles limites de la deuxième articulation varie selon la configuration du robot, l'espace travail du robot à membrures rigides est plus petit que celui du robot à membrures flexible. Par conséquent, les formes géométriques sont plus petites. Les dimensions des formes géométriques sont inscrites dans le Tableau 6.6.

Tableau 6.6: Dimensions des formes géométriques tracés par l'extrémité du robot

Noms de fichier	Formes géométriques
P?R	Carré 0,20 m
P?F	Carré 0,30 m
LMR, LLR, LNR	Ligne horizontale 0,25 m
LMF, LLF, LNF, LCF, LGF	Ligne horizontale 0,45 m
LB?, LH?	Ligne verticale 0,5 m
C?R, A?R	Cercle ou arc de cercle de rayon 0,125 m
C?F, A?F	Cercle ou arc de cercle de rayon 0,25 m

Les cercles CS? et CI? permettent de vérifier l'influence de l'emplacement de la trajectoire (région stable en dessus de l'axe X et région instable au dessus). Les cercles sont identiques au cercle CM? il y a simplement une rotation de ±45 degrés autour du référentiel de la base. C'est seulement la position initiale de l'articulation #1 qui est différente.

6.1.3 Effet de la discontinuité de la dérivée d'accélération sur la déformation des membrures flexibles

L'utilisation de trajectoires sans discontinuité pour la dérivée d'accélération a pour but d'éviter les vibrations au niveau des membrures flexibles. Pour illustrer l'effet de discontinuité de la dérivée d'accélération les Figure 6.1 et Figure 6.2 présentent des graphiques de comparaison de déformation des membrures. L'effet de discontinuité est visible dans les graphiques de vitesse de déformation. Les courbes pour une trajectoire (CMR*.C_1) avec dérivée d'accélération triangulaire (sans discontinuité) sont plus lisses que les courbes pour une trajectoire (CDR*.C_1) avec dérivée d'accélération échelon.

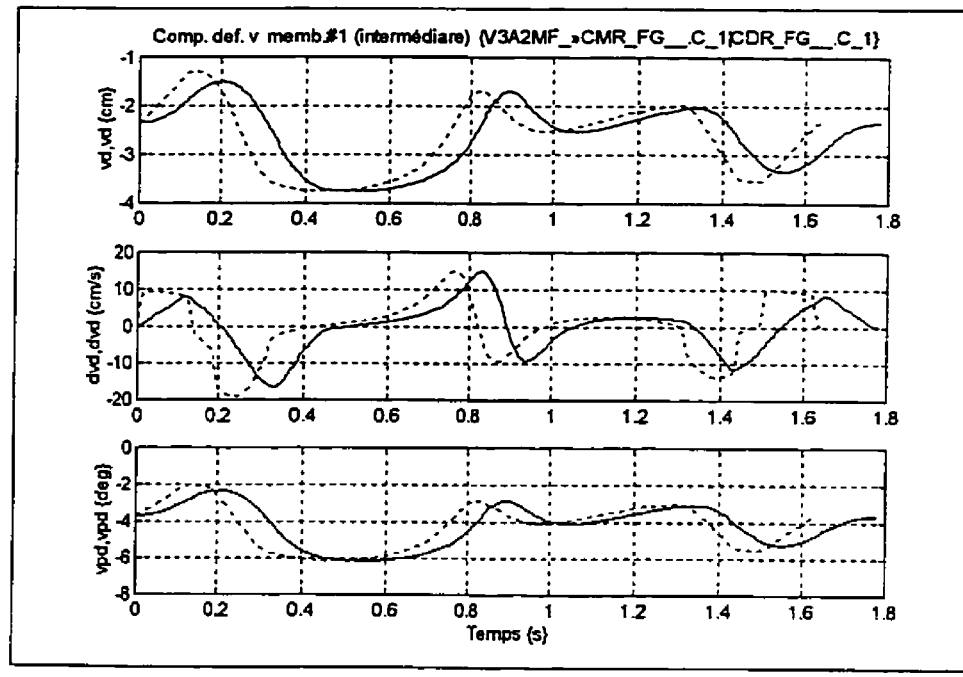


Figure 6.1: Effet de la discontinuité de la dérivée d'accélération (triangulaire : courbe en continu ; échelon : courbe en pointillée) sur la déformation de la membrure #1

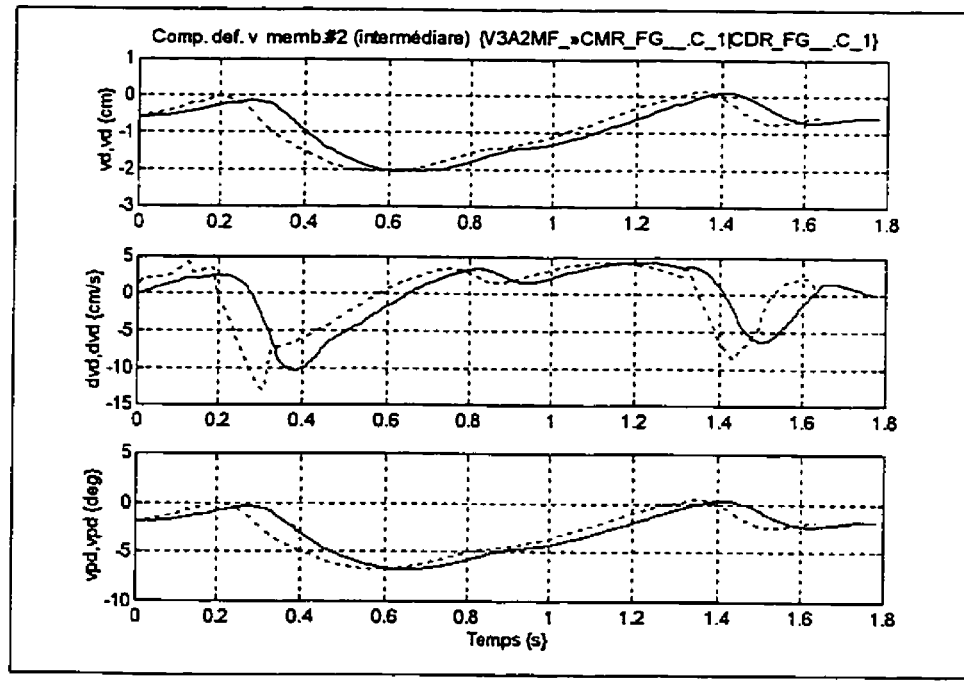


Figure 6.2: Effet de la discontinuité de la dérivée d'accélération (triangulaire : courbe en continu ; échelon : courbe en pointillée) sur la déformation de la membrure #2

6.2 Assignation des gains du contrôleur

Les gains du contrôleur PID ont été déterminés de façon heuristique (par essais et erreurs). Il est à noter que pour tous les essais en boucle fermée le vecteur de gains intégral n'a pas été utilisé (vecteur nul). Par conséquent, il s'agit en réalité d'un contrôleur PD sur la position des articulations. Les gains proportionnels et dérivés sont d'abord déterminés pour une seule articulation à la fois. Ces gains sont ensuite le point de départ pour déterminer les gains du système au complet. Tous les tableaux des gains utilisés pour déterminer les gains du contrôleur sont dans le document « Manuel technique du banc d'essai pour l'étude théorique et expérimentale de robots flexibles ». Les deux prochains tableaux donnent la liste des gains utilisées pour les essais avec chacune des deux configurations. Pour chaque série de vecteurs de gains, deux fichiers ont été crées (même si ils contiennent des gains identiques) afin de pouvoir différencier les essais avec ou sans friction à l'extrémité du robot. La friction provient du traçage de la position de l'extrémité sur le panneau vertical par des crayons feutres.

Tableau 6.7: Gains pour le robot à trois articulations et deux membrures rigides

Gains	#	TRAJ	S
Friction extrémité		oui	non
Proportionnel	1	60	60
	2	25	25
	3	7	7
Dérivée	1	1	1
	2	0,03	0,03
	3	0,12	0,12

Tableau 6.8: Gains pour le robot à trois articulations et deux membrures flexibles

Gains	#	TRAJ	OSCIL	FLEX	S
Friction extrémité		oui	non	non	oui
Proportionnel articulation.	1	40	40	40	40
	2	20	20	20	20
	3	7	7	7	7
Dérivée articulation.	1	0,65	0,65	0,65	0,65
	2	0,30	0,30	0,30	0,30
	2	0,12	0,12	0,12	0,12
Proportionnel jauges	1	0	0	-5,0	-5,0
	4	0	0	-0,5	-0,5
Dérivée jauges	1	0	0	0,3	0,3
	4	0	0	0,04	0,04

Les gains obtenues pour le contrôleur sont différents de ceux pour le système avec des membrures rigides. Si on utilise les mêmes gains lors des essais en boucle fermée, les membrures flexibles vibrent. Les essais utilisant les fichiers de gains `OSCIL.G11` et `FLEX.G11` permettent de vérifier le contrôle de stabilisation (diminution des oscillations en régime permanent) des membrures flexibles. Les essais utilisants les fichiers de gains `FLEX.G11` et `$.G11` permettent de constater l'effet de la friction de l'extrémité sur le système en boucle fermée.

6.3 Vérification de la commande anticipative à l'aide de simulation en boucle ouverte (vecteurs de gains nuls)

La simulation du système (contrôleur avec commande anticipative) est réalisé avec le même modèle que celui utilisé pour calculer la commande anticipative, par conséquent l'erreur est faible même en boucle ouverte (vecteur de gains nul). Lors du calcul de la commande anticipative, le logiciel offre la possibilité d'ajuster le signal pour un système discret. Le signal de commande devient alors la moyenne entre chaque pas

d'échantillonnage, sauf pour les points ou il y a friction statique (début de déplacement). De plus, la simulation s'effectue avec ou sans bloqueur d'ordre zéro à l'entrée du modèle de simulation du robot (commande simulée discrète ou continue). Les graphiques de cette section du chapitre représentent les statistiques d'erreurs de position des coordonnées rigides entre les valeurs désirées et les valeurs calculées (sortie de la simulation).

6.3.1 Effet de la discrétisation et de la friction sur le système à une seule articulation (ID_A1_.PAR)

La Figure 6.3 confirme bien que l'erreur entre les coordonnées désirées (pour une trajectoire de 0 à 90 degrés) et les coordonnées simulées est très faible. La différence d'erreur moyenne et d'erreur maximum pour la barre #1 par rapport aux autres barres, provient de la discrétisation. La commande initiale a l'amplitude de la friction statique. Dès que l'articulation se met en mouvement, la friction simulée est alors une friction dynamique plus faible que la friction statique. La commande initiale reste constante pendant le premier pas de discrétisation et par le fait même engendre une erreur plus grande que pour celles obtenues avec les autres simulations.

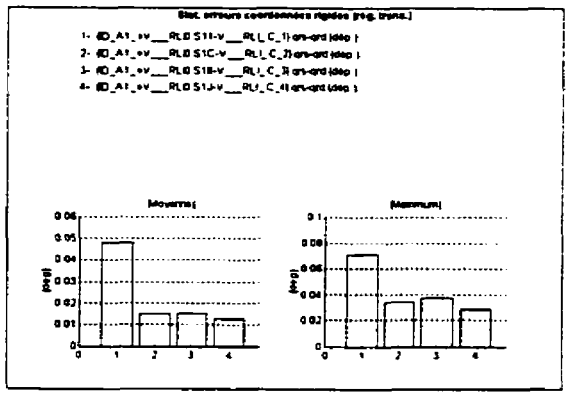


Figure 6.3: Effet de la friction pour des simulations discrètes ou continues

Tableau 6.9: Effet de la friction pour simulations discrètes ou continues (Légende supplémentaire)

	Commande		Simulation	
#	friction	discret	friction	BOZ
1	AVEC	OUI	AVEC	OUI
2	SANS	OUI	SANS	OUI
3	AVEC	NON	AVEC	NON
4	SANS	NON	SANS	NON

Les deux prochaines figures nous montre très bien l'influence de l'ajustement de la commande pour un système discret. La Figure 6.4 présente les statistiques d'erreurs pour des essais sans friction, tandis que pour les statistiques des essais avec friction sont données à la Figure 6.5.

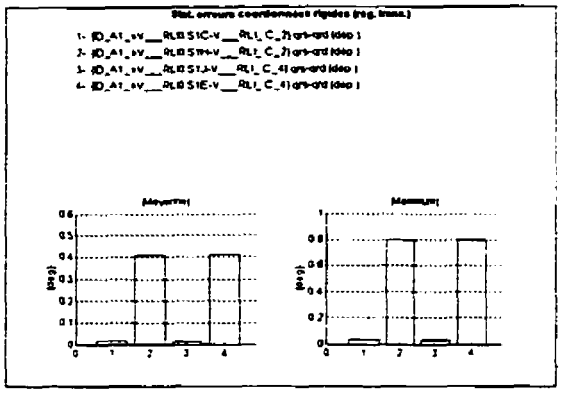


Figure 6.4: Effet de la discrétisation pour des simulations sans friction

Tableau 6.10: Effet de la discrétisation pour des simulations sans friction (Légende supplémentaire)

	Commande		**Simulation**	
#	friction	discret	friction	BOZ
1	SANS	OUI	SANS	OUI
2	SANS	OUI	SANS	NON
3	SANS	NON	SANS	NON
4	SANS	NON	SANS	OUI

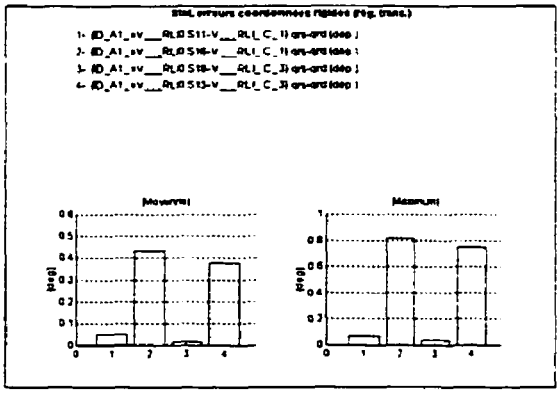


Figure 6.5: Effet de la discrétisation pour des simulations avec friction

Tableau 6.11: Effet de la discrétisation pour des simulations avec friction (Légende supplémentaire)

	Commande		**Simulation**	
#	friction	discret	friction	BOZ
1	AVEC	OUI	AVEC	OUI
2	AVEC	OUI	AVEC	NON
3	AVEC	NON	AVEC	NON
4	AVEC	NON	AVEC	OUI

6.4 Vérification de la position (statique) de l'extrémité à l'aide de mesures sur papier millimétrique

Pour vérifier la position de l'extrémité, des trajectoires (lignes horizontales et verticales) ont été tracées sur des feuilles de papier millimétrique. Le Tableau 6.12 et le Tableau 6.13 contiennent les moyennes[2] des erreurs de positions de départ et d'arrivée.

Tableau 6.12: Erreurs (distances) moyennes de positions de l'extrémité pour robot à membrures rigides (V3A2MR.PAR)

Erreur moyenne (distance)	pour les positions de départs	pour les positions d'arrivées
	(mm)	(mm)
Entre les positions tracées sur papier et les positions désirées	4,9	9,2
Entre les positions calculées à partir des capteurs et les positions désirées	1,4	8,5
Entre les positions tracées sur papier et les positions calculées à partir des capteurs	4,8	5,4

Tableau 6.13: Erreurs (distances) moyennes de positions de l'extrémité pour robot à membrures flexibles (V3A2MF.PAR)

Erreur moyenne (distance)	pour les positions de départs	pour les positions d'arrivées
	(mm)	(mm)
Entre les positions tracées sur papier et les positions désirées	12,7	14,4
Entre les positions calculées à partir des capteurs et les positions désirées	7,9	11,4
Entre les positions tracées sur papier et les positions calculées à partir des capteurs	7,3	5,9

[2] Les moyennes sont calculés à partir des valeurs inscrites dans les tableaux d'erreurs de positions de l'extrémité du document « Manuel technique du banc d'essai pour l'étude théorique et expérimentale de robots flexibles »

Voici la liste des sources d'erreurs possibles pour la position de l'extrémité:

1) Erreur entre les positions désirées des articulations et les positions atteintes;
2) Erreur de positionnement au point de repère (home) lors de l'initialisation du système avant de réaliser l'essais expérimental;
3) Précision du positionnement de la feuille de papier millimétrique par rapport au référentiel de la base (observateur) du robot;
4) Précision des dimensions des composantes du montage et des facteurs de conversion des capteurs de flexibilité (jauges de déformation);
5) Erreur (décalage) de tracé sur papier;
6) Précision des mesures sur le papier.

Une des principales sources d'erreurs de position de l'extrémité provient du contrôleur. Avec le contrôleur utilisé, il y a toujours une erreur en régime permanent entre les valeurs désirées (positions des articulations) et les valeurs atteintes. Puisque l'usager doit manuellement placer le robot à sa position de repère, une erreur peut donc s'introduire. La précision de positionnement du papier sur lequel on trace l'extrémité du robot est une autre source d'erreur. Malgré la minutie à laquelle les dimensions de chacune des composantes du montage sont mesurées, la précision des mesures engendre une erreur de position à l'extrémité du robot. De plus, pour un robot à membrures flexibles, la position de l'extrémité est fonction de la déformation des membrures. La position est calculée en utilisant les mesures des jauges de déformations collés sur les membrures. Les signaux provenant de ces capteurs sont très faibles et varient très peu. Par conséquent, la précision des facteurs de conversion des capteurs est une autre source d'erreur. Le support sur lequel est installé le crayon feutre fait en sorte que l'angle entre le crayon et la surface (Figure 6.6) varie selon la distance entre le panneau et le support de crayon et cause un décalage du tracé. La dernière source d'erreur est la précision des mesures effectuées sur le papier (± 0,05 cm).

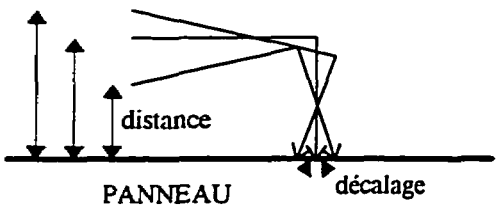


Figure 6.6: Décalage du tracé en fonction de la distance entre le panneau et le support du crayon

6.5 Robot à trois articulations et deux membrures rigides (V3A2MR_.PAR)

6.5.1 Visualisation de l'erreur de l'extrémité entre résultats expérimentaux et positions désirées

Les six prochaines figures permettent de comparer la trajectoire de l'extrémité par rapport à la trajectoire désirée pour un robot à membrures rigides. La Figure 6.7 peut servir de référence aux autres figures, il s'agit d'un suivie de cercle à vitesse moyenne (déplacement dans le sens anti-horaire). La Figure 6.8 est un cercle tracé à vitesse lente, l'erreur de suivi de trajectoire est plus petite. Par contre, lorsqu'on effectue le trajet dans le sens inverse, l'erreur devient plus grande (Figure 6.9). Pour un système à membrures rigides, une trajectoire avec discontinuité pour la dérivée d'accélération n'a pas (ou peu) d'influence sur le résultat (Figure 6.10). Les Figure 6.11 et Figure 6.12 permettent de constater que l'erreur de suivi de trajectoire n'est pas fonction de l'emplacement de la trajectoire, mais fonction du temps. Le "x" sur les graphiques indique le point de départ de la trajectoire. Les trajectoires désirées sont en trait pointillé tandis que les trajectoires calculées à partir des mesures expérimentales des articulations sont en trait continu.

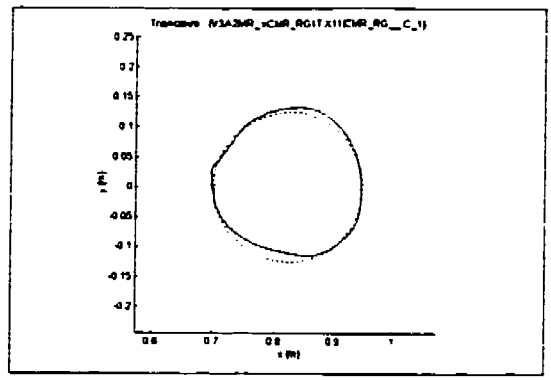


Figure 6.7: Cercle à vitesse moyenne (CMR)

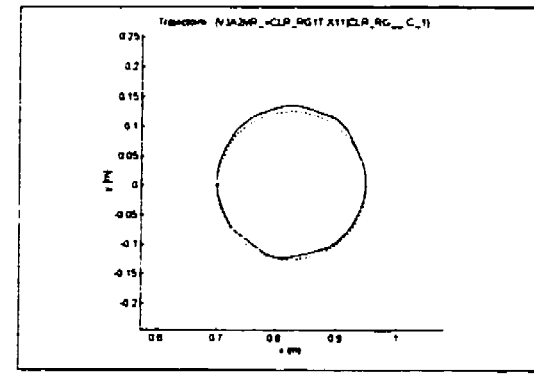


Figure 6.8: Cercle à vitesse lente (CLR)

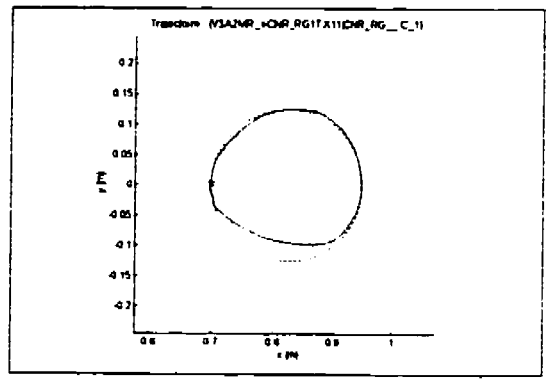


Figure 6.9: Cercle avec déplacement dans le sens horaire (CNR)

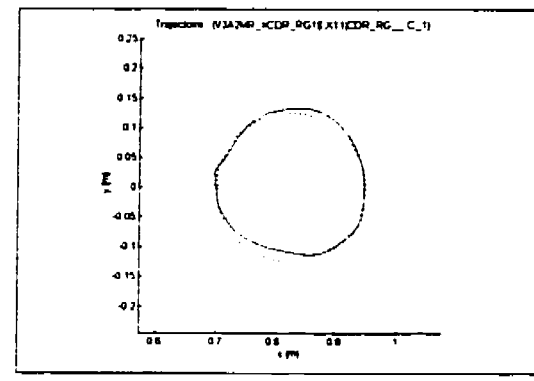


Figure 6.10: Cercle avec dérivée d'accélération. discontinue (CDR)

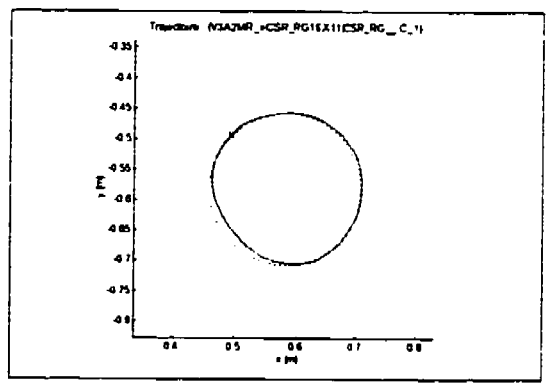


Figure 6.11: Cercle dans la région stable (y<0) (CSR)

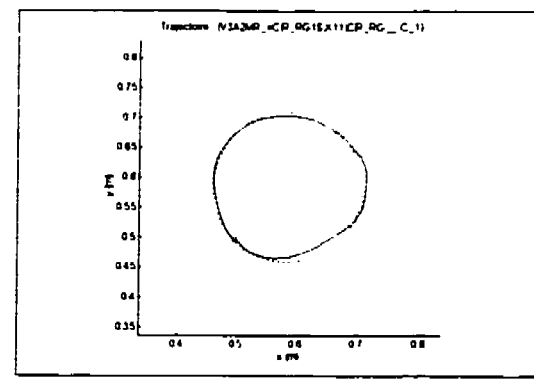


Figure 6.12: Cercle dans la région instable (y>0) (CIR)

6.5.2 Vérification de l'erreur dynamique pour un suivi de trajectoire

Comme on l'a dit précédemment, la source principale d'erreur de position de l'extrémité est fonction de l'erreur de suivi de trajectoire des articulations. Les six prochaines figures présentes les graphiques de comparaisons et d'erreur (différence) entre les coordonnées rigides mesurées et les coordonnées désirées pour chacune des trois articulations.

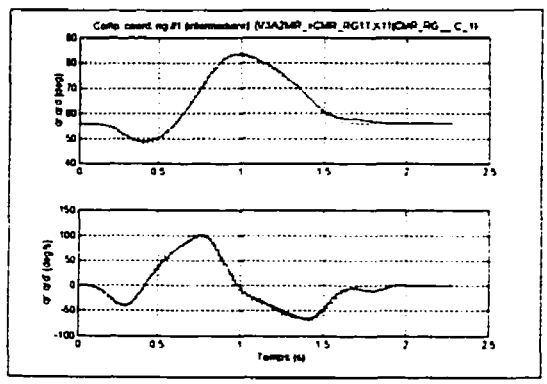


Figure 6.13: Comparaison coordonnée rigide #1 (CMR)

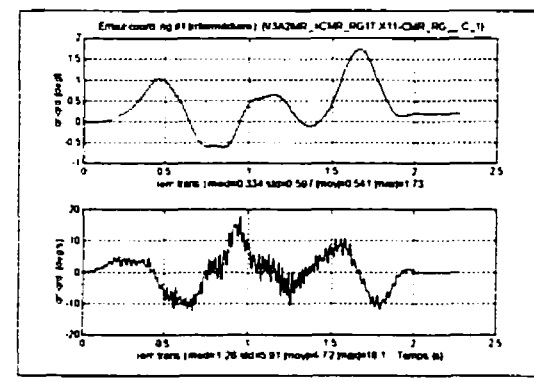


Figure 6.14: Erreur coordonnée rigide #1 (CMR)

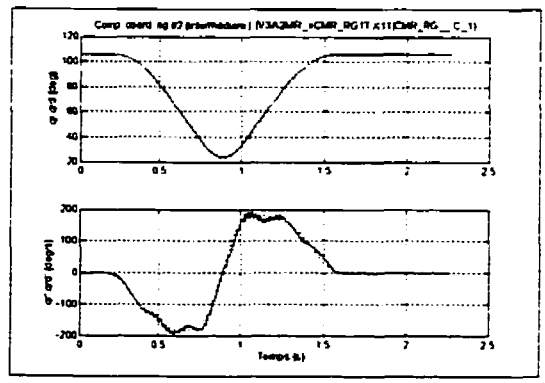


Figure 6.15: Comparaison coordonnée rigide #2 (CMR)

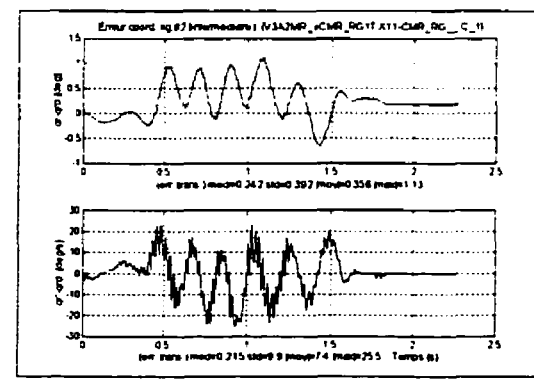


Figure 6.16: Erreur coordonnée rigide #2 (CMR)

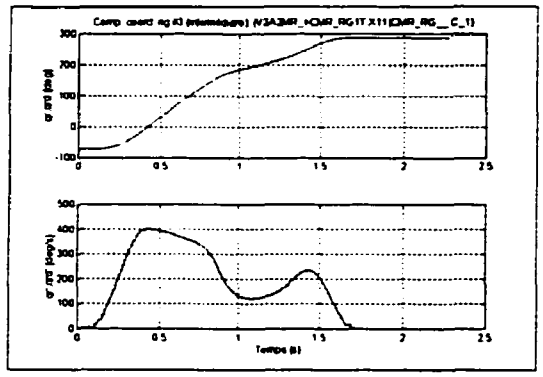


Figure 6.17: Comparaison coordonnée rigide #3 (CMR)

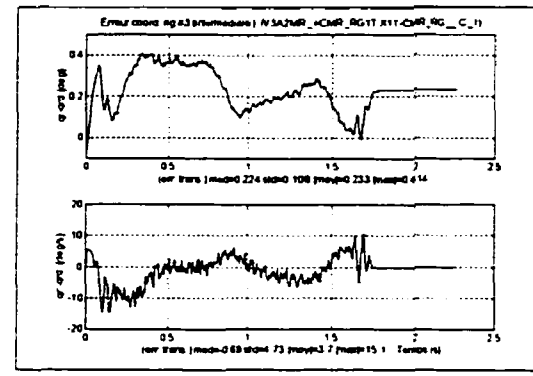


Figure 6.18: Erreur coordonnée rigide #3 (CMR)

En comparant les graphiques d'erreur horizontale (Figure 6.19) et d'erreur verticale (Figure 6.20) avec les graphiques d'erreur de coordonnées rigides, on constate que la trajectoire désirée (cercle avec son centre sur l'axe des X) fait en sorte que l'articulation #1 agit principalement sur le déplacement vertical et que l'articulation #2 agit principalement sur le déplacement horizontal.

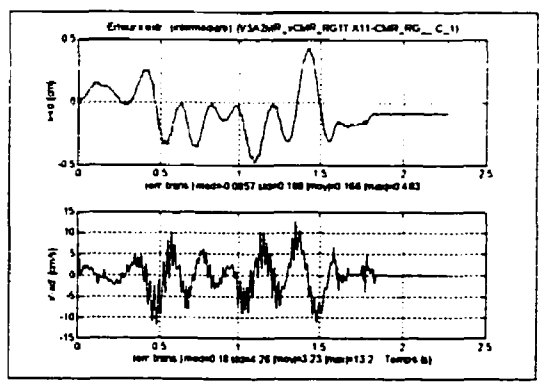


Figure 6.19: Erreur horizontale de l'extrémité (CMR)

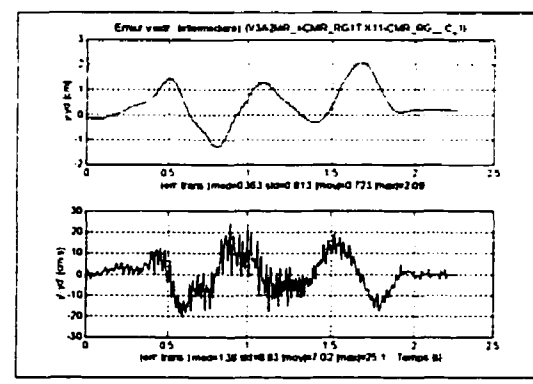


Figure 6.20: Erreur verticale de l'extrémité (CMR)

À partir des graphiques de comparaison et d'erreur de vitesse tangentielle (Figure 6.21, Figure 6.23) et de comparaison et d'erreur d'orientation (Figure 6.22, Figure 6.24), on remarque que l'oscillation de l'erreur débutent lorsque la trajectoire atteint le plateau de vitesse tangentielle et de vitesse de changement d'orientation de l'extrémité.

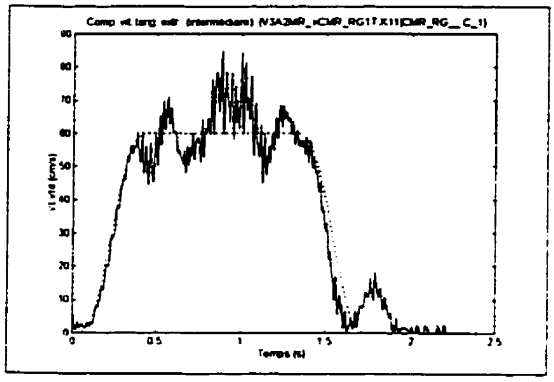


Figure 6.21: Comparaison de vitesse tangentielle de l'extrémité (CMR)

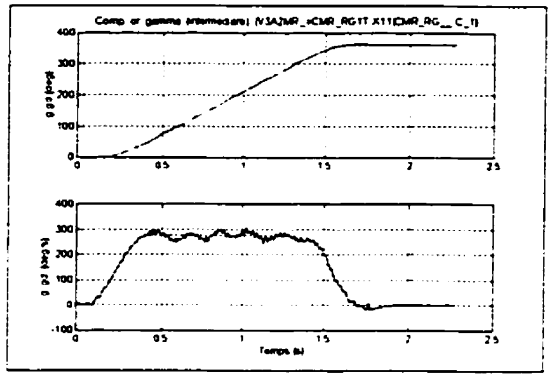


Figure 6.22: Comparaison d'orientation de l'extrémité (CMR)

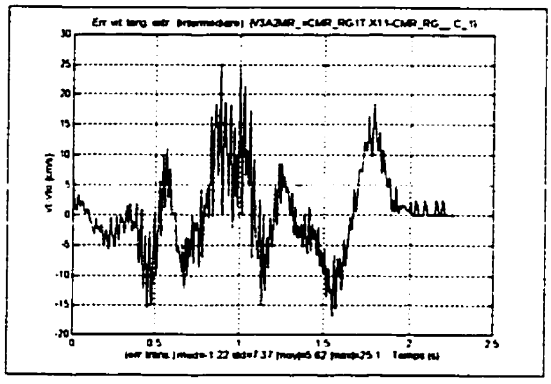


Figure 6.23: Erreur de vitesse tangentielle de l'extrémité (CMR)

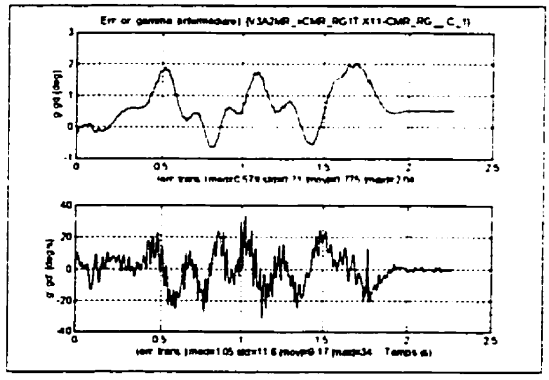


Figure 6.24: Erreur d'orientation de l'extrémité (CMR)

6.6 Robot à trois articulations et deux membrures flexibles (V3A2MF_.PAR)

6.6.1 Comparaison entre les valeurs mesurées par les capteurs de flexibilité et les valeurs estimés

On peut voir sur les graphiques de la Figure 6.25 et de la Figure 6.26 la comparaison des valeurs mesurées (trait continu) par les capteurs de flexibilité avec les valeurs estimées (trait pointillé) lors du calcul de commande anticipative.

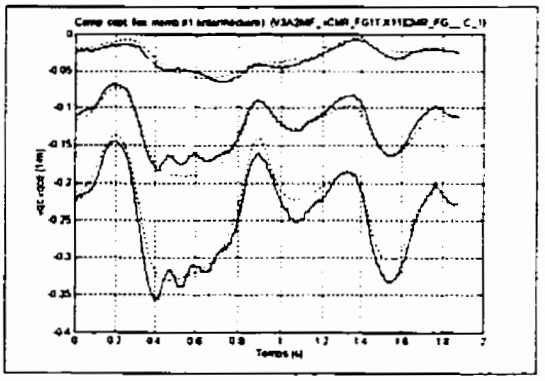


Figure 6.25: Comparaison capteurs de flexibilité pour la membrure #1 (CMR)

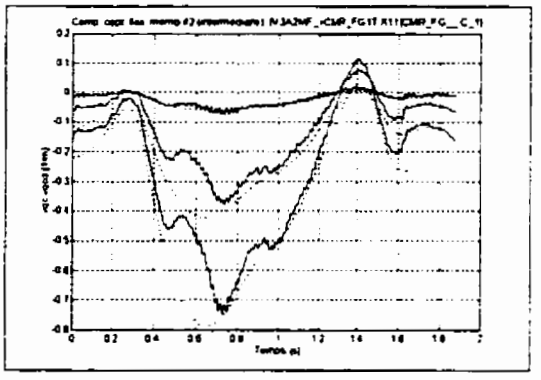


Figure 6.26: Comparaison capteurs de flexibilité pour la membrure #2 (CMR)

6.6.2 Vérification du contrôleur des capteurs de flexibilité

Pour vérifier l'effet du contrôleur sur les oscillations des membrures, deux essais (avec gains non nuls et avec gains nuls) ont été effectués sans frottement à l'extrémité (pas de traçage). Les courbes de la Figure 6.27 et de la Figure 6.28 permettent de comparer les résultats (avec contrôleur:trait continu, sans contrôleur: trait pointillé). Le contrôleur réduit effectivement les oscillations des membrures en régime permanent.

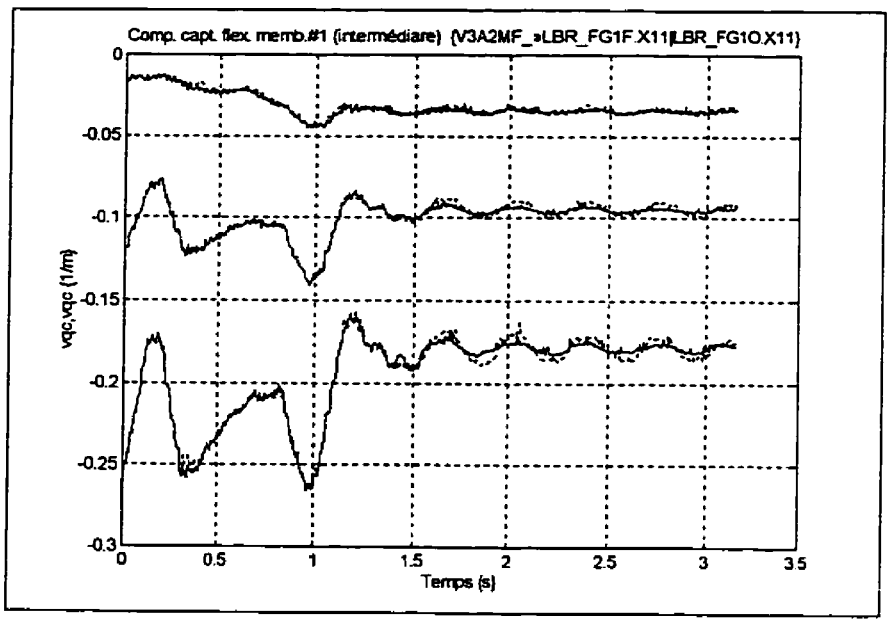


Figure 6.27: Effet du contrôleur des capteurs de flexibilité de la membrure #1

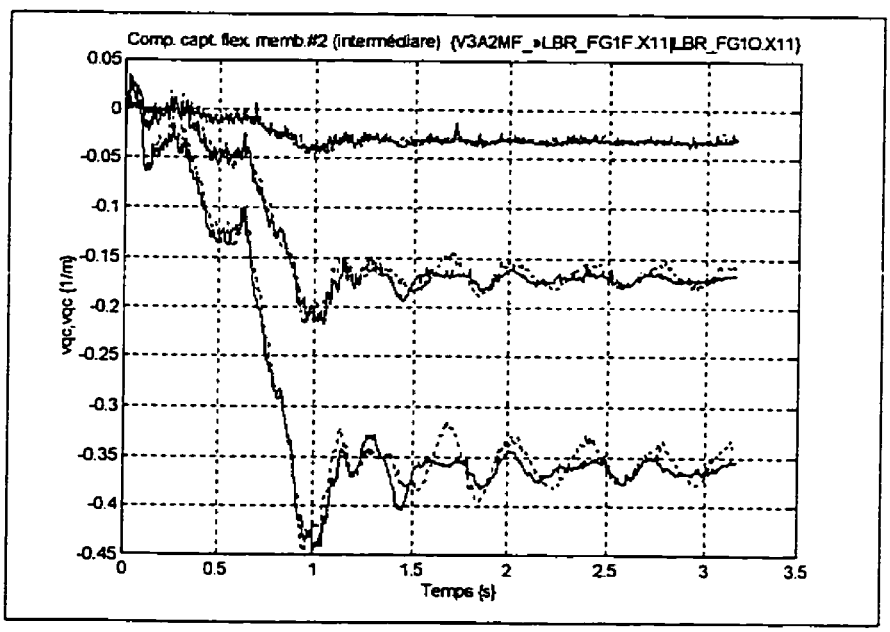


Figure 6.28: Effet du contrôleur des capteurs de flexibilité de la membrure #2

6.7 Comparaisons de résultats avec un robot à deux membrures rigides (V3A2MR_.PAR) et un robot à deux membrures flexibles (V3A2MF_.PAR)

6.7.1 Visualisation des trajectoires de l'extrémité

Les prochaines figures (tracés à partir de résultats expérimentaux avec contrôleur PD et commande anticipative) permettent de comparer les trajectoires (trait en continu) pour le robot à membrures flexibles avec les trajectoires obtenues (trait en pointillé) avec le robot ayant des membrures rigides. Les carrés (Figure 6.29, Figure 6.30) tracés avec le robot à membrures rigides sont un peu mieux que ceux tracés avec le robot à membrures flexibles. La différence se situe à chaque coin du carré, au moment ou il y a seulement un changement d'orientation de l'extrémité (la position désirée reste constante).

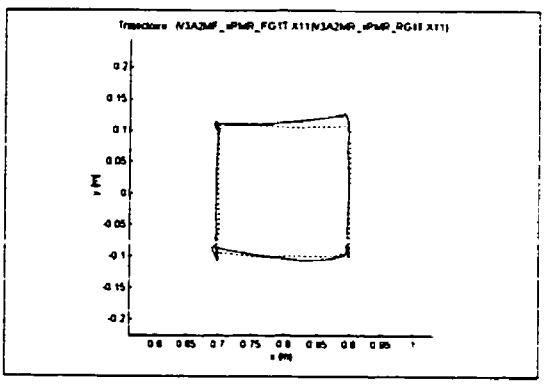


Figure 6.29: Carré avec déplacement dans le sens anti-horaire (PMR)

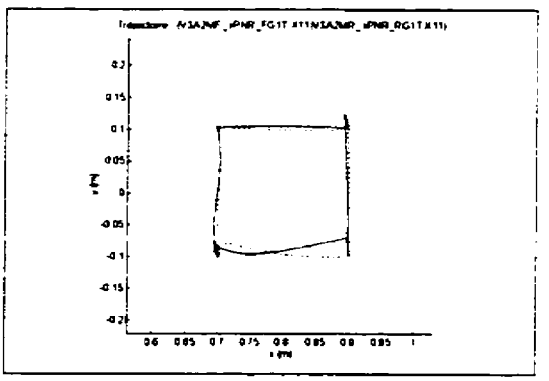


Figure 6.30: Carré avec déplacement dans le sens horaire (PNR)

Tout comme pour les carrés, les trajectoires (Figure 6.31 à Figure 6.34) circulaires effectués avec le robot à membrures rigides sont un peu mieux que celles effectuées avec le robot à membrures flexibles.

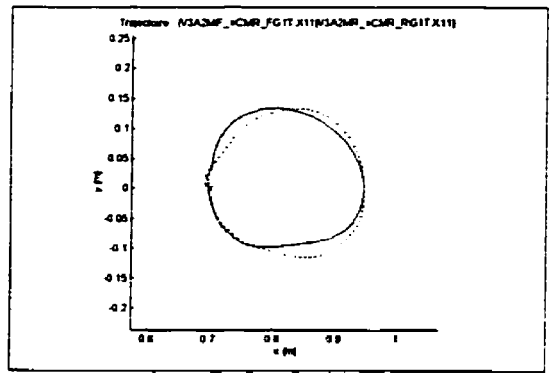


Figure 6.31: Comparaison de cercles avec déplacement dans le sens anti horaire (CMR)

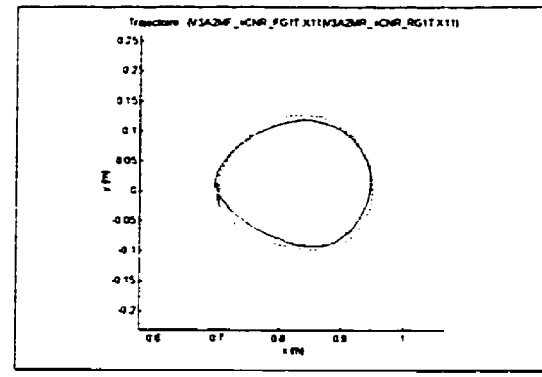


Figure 6.32: Comparaison de cercles avec déplacement dans le sens horaire (CNR)

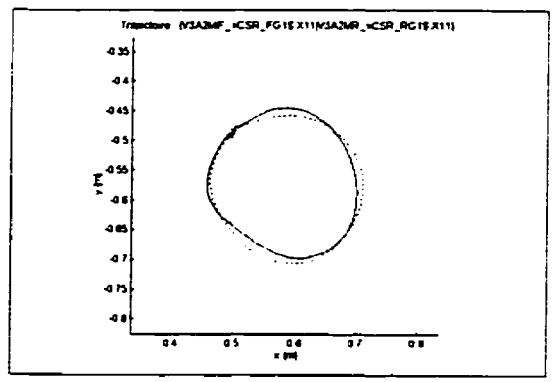


Figure 6.33:Comparaison de cercles dans la région stable (CSR)

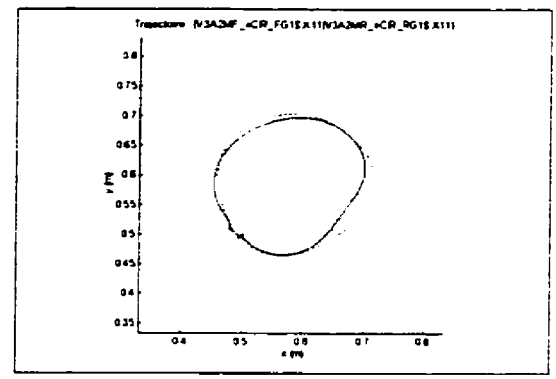


Figure 6.34:Comparaison de cercles dans la région instable (CIR)

6.7.2 Vérification des erreurs dynamiques

L'erreur de suivi des trajectoires carrées est plus grande selon l'axe vertical. Les prochaines figures montrent les graphiques de comparaison et d'erreur de position verticale de l'extrémité. On constate que l'erreur (oscillation) qui était présente pour le robot à membrures rigides (Figure 6.35, Figure 6.36) est accentuée pour le robot à membrures flexibles (Figure 6.37, Figure 6.38).

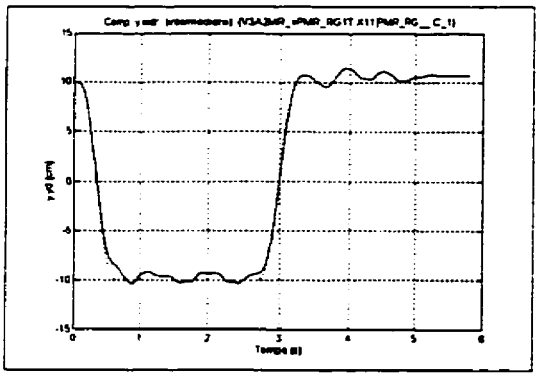


Figure 6.35: Comparaison de position verticale de l'extrémité du robot à membrures rigides (PMR)

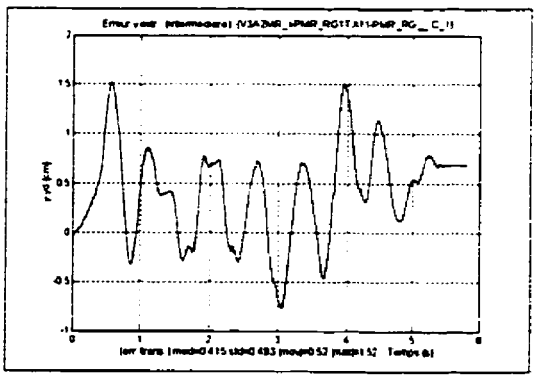


Figure 6.36: Erreur de position verticale de l'extrémité du robot à membrures rigides (PMR)

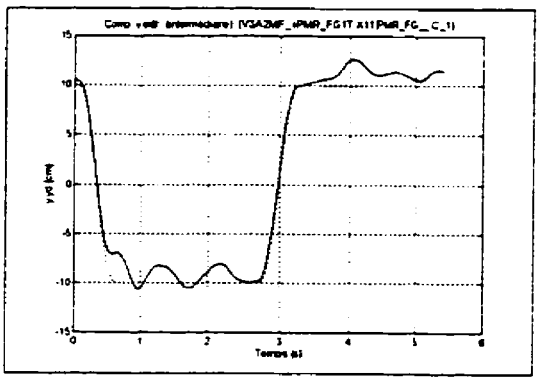


Figure 6.37: Comparaison de position verticale de l'extrémité du robot à membrures flexibles (PMR)

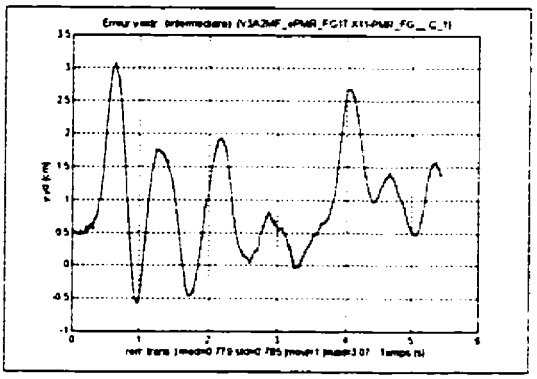


Figure 6.38: Erreur de position verticale de l'extrémité du robot à membrures flexibles (PMR)

Un des facteurs qui explique la différence de résultats est que les gains utilisés pour le robot à membrures flexibles (Tableau 6.8) sont plus faibles que ceux du robot ayant des membrures rigides (Tableau 6.7). Mais comme on l'a déjà expliqué, on ne peut pas utiliser les mêmes gains puisqu'il y aura trop d'oscillations (vibration des membrures flexibles). Les graphiques de statistiques d'erreur sur les coordonnées rigides (Figure 6.39, Figure 6.40) confirment que l'erreur est un peu plus grande avec le robot à membrures flexibles.

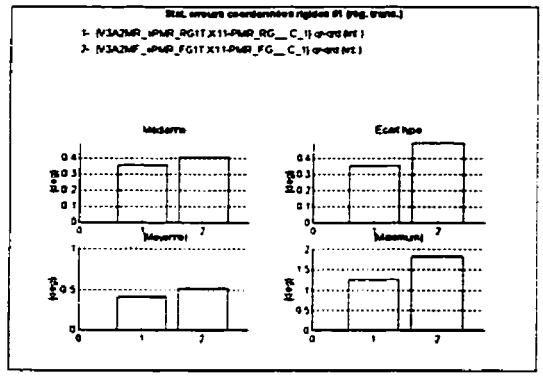


Figure 6.39: Statistiques d'erreurs sur la coordonnée rigide #1 (PMR)

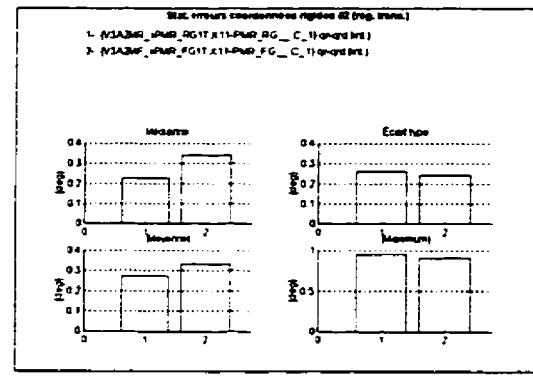


Figure 6.40: Statistiques d'erreurs sur la coordonnée rigide #2 (PMR)

6.8 Conclusion

Les résultats présentés dans ce chapitre ont permis d'exploiter les différentes possibilités du logiciel ROVASIMEX. Les résultats expérimentaux ne sont que des essais préliminaires sur le système. D'autres essais devrait être effectués afin de mieux ajuster les gains du système et inclure l'effet de l'élasticité aux actionneurs lors du calcul de la commande anticipative. De plus d'autres types de contrôleur (exemple: couple pré-calculé) pourrait être implantés de façon à comparer les résultats avec ceux obtenues par le contrôleur utilisé dans ce projet.

7 CONCLUSION

Le robot expérimental conçu et fabriqué au cours de ce projet de maîtrise devrait permettre d'étudier plus à fond le comportement de système à membrures flexibles. Pour faciliter l'analyse, le logiciel ROVASIMEX contient tous les outils nécessaires pour effectuer des essais autant en simulation que sur le montage expérimental. Pour modéliser le système, le logiciel ROVASIMEX fait appel au logiciel SYMOFROS qui à partir de paramètres symboliques établit un modèle. Des valeurs numériques sont assignés aux paramètres symboliques pour ensuite créer un modèle numérique du système. Le suivi de trajectoire désirée des articulations s'effectue à l'aide d'un contrôleur PD auquel on ajoute une commande anticipative calculée à partir du modèle numérique du système. Lors du calcul de la commande anticipative, une compensation pour la friction est ajoutée. De plus, la trajectoire des articulations est corrigée pour tenir compte des déformations des membrures flexibles. Des jauges de déformations sont collés sur les membrures pour mesurer les déformations. La position de l'extrémité est alors estimée à l'aide de ces mesures.

Finalement, les limitations du système sont présentés ainsi que des suggestions pour la poursuite de travaux.

7.1 Limitations du système

Le logiciel Matlab utilisé par le logiciel ROVASIMEX ne fonctionne pas toujours correctement dans l'environnement de Windows 95. Lorsqu'une fenêtre de Matlab est en format plein écran et que l'on effectue des changements (zoom, couleurs, styles des courbes), il n'y pas de mise à jour de l'écran. Il faut alors remettre la fenêtre à sa dimension normal pour ensuite remettre en format plein écran pour voir les changements effectués. De plus, il arrive parfois que le logiciel Matlab cause une erreur fatale (message de Windows 95) qui fait en sorte qu'il est automatiquement fermé. Le logiciel Matlab a été compilé pour Windows 3.1, une nouvelle version du logiciel Matlab pour Windows 95 corrigera probablement ces défauts.

Pour bien fonctionner, les programmes en temps réel doivent être compilés avec l'option code 286 et exécutable sous DOS. Les programmes dans l'environnement DOS sont alors limités à n'utiliser qu'un maximum de un méga octets. Pour utiliser pleinement les capacités de l'ordinateur, il faudrait convertir les programmes d'acquisitions et de contrôle en temps réel pour être directement utilisés par le logiciel Matlab. De plus, cette conversion faciliterait l'implantation de nouveaux contrôleurs.

Il y a deux possibilités de conversion des programmes:

1) sous forme de fonction Matlab (*.MEX);
2) sous forme de S-function de Simulink (*.MEX).

Les graphiques qui sont tracés avec le logiciel ROVASIMEX sont en français. Puisque la plupart du temps les articles dans le domaine sont en anglais, il serait bon d'offrir la possibilité de choisir la langue d'affichage (titres) des graphiques.

7.2 Suggestions pour la poursuite des travaux

Voici une liste de suggestions pour la poursuite des travaux:

- effectuer d'autres essais (simulations et expérimentaux) en incluant l'élasticité aux actionneurs lors du calculs de la commande anticipative;
- effectuer d'autres essais expérimentaux pour ajuster les gains du contrôleur;
- implanter d'autres types de contrôleurs;
- implanter dans le logiciel ROVASIMEX la possibilité de choisir la langue d'utilisation d'affichage des graphiques;
- implanter la nouvelle version du programme SYMOFROS;
- ajouter des méthodes pour déterminer les gains des contrôleurs;
- convertir les programmes d'acquisitions (DOS) pour être utilisés directement avec Matlab;
- compléter la documentation (Manuel technique=> câblages: pin-out des connecteurs, dessins électriques alimentation avec protection);
- compléter l'interface usager (message sur appel de bouton d'aide) du logiciel ROVASIMEX;
- concevoir et installer un nouveau support de crayons feutres à l'extrémité du robot;

- déterminer et corriger la source d'erreur de lecture des encodeurs;

RÉFÉRENCES

BAZERGUI, A., BIRON, A., BUI-QUOC, T., LABERGE, C. et McINTYRE, G. (1987). Résistance de matériaux. Édition corrigée, Édition de l'École Polytechnique de Montréal, Canada.

BEER, F.B. et JOHNSTON, E.R.Jr. (1988). Vector Mechanics for Engineers: Statics. Fifth edition, McGraw-Hill Book Company, United States.

BRADY, M. (1984). Robot Motion: Planning and control. Trajectory Planning. The Massachusetts Institute of Technology, United States

BREMER, H. and PFEIFFER, F. (1994). Experiments with flexible Manipulator. Preprints of the Fourth IFAC Symposium on Robot Control September 19-21, 1994, Capri, Italy.

BUCHAN, K.S., CARUSONE, J. and D'ELEUTERIO, G.M.T. (1996). Experiments in End-Effector Tracking Control for Structurally flexible Space Manipulators. Submission to IEEE Transactions on Robotics and Automation, Special issue on space robotics.

CORKE, P.I. (1996). A Rototics Toolbox for MATLAB. IEEE Robotics and Automation Magazine, March, 1996

CRAIG, J.J. (1989). Introduction to robotics: mechanics and control. Second edition, Addison-Wesley Publishing Company, United States.

DAMAREN, C., SHARF, I. and STANWAY, J. (1996). Validation of a Dynamics Simulation for a Structurally Flexible Manipulator. Proceedings of the 1996 IEEE International Conference on Robotics and Automation, Miniapolis, Minesota, April 1996.

ELMARAGHY, W.H. and ZAKI, A. S. (1994). Experiments on the Modeling and Vibration Control of a Three-degree of Freedom Manipulator with Flexible Links. EIC, Vol.18, No. 1, 1994.

FU, K.S., GONZALES, R.C., LEE, C.S. (1987). Robotics: Control, Sensing, Vision, and Intelligence. McGraw-Hill, United States

KUO, B.C. (1987). Automatic Control Systems. Fifth edition, Prentice-Hall, United States.

MILLER, S. (1995). Design of a two degree of freedom flexible manipulator including joint friction compensation and endpoint position adn orientation measurement. Master thesis, Queen's University, Kingston, Ontario, Canada.

MILLER, S. et PIEDBOEUF, J.C. (1994). Estimation of endpoint position and orientation of a flexible link using strain gauges. Preprints of the Fourth IFAC Symposium on Robot Control September 19-21, 1994, Capri, Italy.

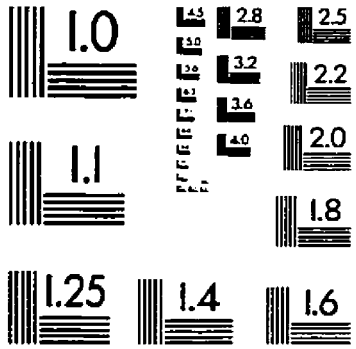
PIEDBOEUF, J.C. (1992). Introduction à la modélisation et au contrôle des robots flexibles. Canada.

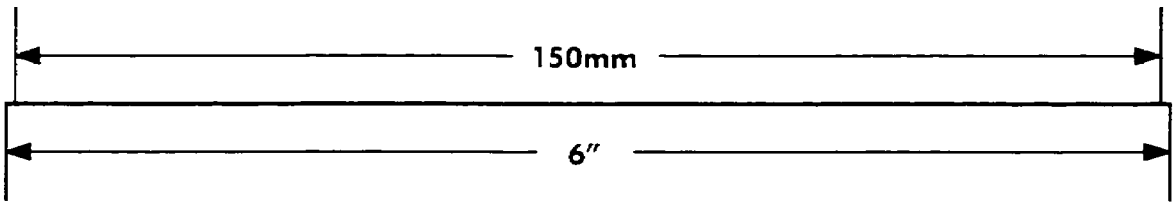
PIEDBOEUF, J.C. (1995). Modeling Flexible Robots with Maple. The Maple Technical Newsletter, Canada.

PIEDBOEUF, J.C. (1995). Symbolic modelling of flexible manipulators. Proc. of the AAS/AIAA Astrodynamics Conference, (Halifax, Nova Scotia, Canada), august 14-17 1995.

ZIARATI, K. (1993). Mise au point d'un contrôleur anticipatif pour un manipulateur flexible. Mémoire de maîtrise, École Polytechnique de Montréal, Canada.

TEST TARGET (QA-3)





APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved