

Titre: Modélisation et contrôle d'un manipulateur à 3 degrés de liberté
Title:

Auteurs: Sylvain Marleau
Authors:

Date: 2003

Type: Rapport / Report

Référence: Marleau, S. (2003). Modélisation et contrôle d'un manipulateur à 3 degrés de liberté. (Technical Report n° EPM-RT-2003-13).
Citation: <https://publications.polymtl.ca/2613/>

Document en libre accès dans PolyPublie

Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/2613/>
PolyPublie URL:

Version: Version officielle de l'éditeur / Published version

Conditions d'utilisation: Tous droits réservés / All rights reserved
Terms of Use:

Document publié chez l'éditeur officiel

Document issued by the official publisher

Institution: École Polytechnique de Montréal

Numéro de rapport: EPM-RT-2003-13
Report number:

URL officiel:
Official URL:

Mention légale:
Legal notice:

(3 0 ± 5 7 ±

0 2 ' e / , 6 \$ 7 , 2 1 (7 & 2 1 7 5 0 / (' ¶ 8 1 0 \$ 1 , 3 8 / \$ 7 (8 5
- ' (* 5 e 6 ' (/ , % (5 7 e

6 \ O Y D L Q 0 D U O H D X
' p S D U W H P H Q W G H * p Q L H p O H F W U L T X H
e F R O H 3 R O \ W H F K Q L T X H G H 0 R Q W U p D O

2 F W R E U H

Poly

EPM / RT – 2003 - 13

MODÉLISATION ET CONTÔLE D'UN MANIPULATEUR
À 3 DEGRÉS DE LIBERTÉ

Sylvain Marleau
École Polytechnique de Montréal

Projet dirigé par

Professeur Romano DeSantis
Professeur Richard Gourdeau

Octobre 2003

©2003
Sylvain Marleau, Romano DeSantis et Richard
Gourdeau
Tous droits réservés

Dépôt légal :
Bibliothèque nationale du Québec, 2003
Bibliothèque nationale du Canada, 2003

EPM-RT-2003-13
Modélisation et contrôle d'un manipulateur à 3 degrés de liberté
par : Sylvain Marleau
Département de génie électrique.
École Polytechnique de Montréal

Toute reproduction de ce document à des fins d'étude personnelle ou de recherche est autorisée à la condition que la citation ci-dessus y soit mentionnée.

Tout autre usage doit faire l'objet d'une autorisation écrite des auteurs. Les demandes peuvent être adressées directement aux auteurs (consulter le bottin sur le site <http://www.polymtl.ca/>) ou par l'entremise de la Bibliothèque :

École Polytechnique de Montréal
Bibliothèque – Service de fourniture de documents
Case postale 6079, Succursale «Centre-Ville»
Montréal (Québec)
Canada H3C 3A7

Téléphone : (514) 340-4846
Télécopie : (514) 340-4026
Courrier électronique : biblio.sfd@polymtl.ca

Pour se procurer une copie de ce rapport, s'adresser à la Bibliothèque de l'École Polytechnique.

Prix : 25.00\$ (sujet à changement sans préavis)
Régler par chèque ou mandat-poste au nom de l'École Polytechnique de Montréal.

Modélisation et contrôle d'un manipulateur à 3 degrés de liberté

Sylvain Marleau
Département de Génie électrique

Sommaire

Ce rapport concerne l'application de l'algorithme de commande adaptative de Slotine-Li à l'asservissement d'un manipulateur à 3 degrés de liberté.

Parmi les points d'intérêts, on rencontre :

1. l'obtention du modèle du manipulateur en appliquant le procédé de Kane telle que formulé par M. De Santis et la validation de ce même modèle en utilisant la librairie Roboop mise au point par M. Gourdeau.
2. une analyse par voie de simulation du comportement du manipulateur lorsque sous l'asservissement d'un contrôleur conçu à partir d'une estimation suffisamment précise des paramètres du modèle.
3. une analyse par voie de simulation du comportement du manipulateur lorsque sous l'asservissement d'un contrôleur adaptatif de Slotine-Li implanté dans sa forme originale ainsi que dans une forme convenablement modifiée.

Les résultats illustrent clairement les avantages et les désavantages des différents types de compensateurs ainsi que le potentiel que Matlab, Simulink, le "Virtual Modeling Tools Box" et Roboop offrent pour simuler des systèmes et des lois de commandes complexes.

Table des matières

| | |
|--|-----------|
| 1 Remerciements | 6 |
| 2 Introduction | 7 |
| 3 Modélisation d'un manipulateur à 3 degrés de liberté | 7 |
| 3.1 Présentation du manipulateur | 7 |
| 3.2 Convention de Denavit-Hartenberg | 8 |
| 3.3 Cinématique directe | 9 |
| 3.4 La matrice jacobienne étendue | 11 |
| 3.4.1 Calcul de J_1 | 12 |
| 3.4.2 Calcul de J_2 | 13 |
| 3.4.3 Calcul de J_3 | 14 |
| 3.4.4 Calcul de la matrice jacobienne étendue | 16 |
| 3.4.5 La dérivée de la matrice jacobienne étendue | 17 |
| 3.5 La matrice de masse étendue | 18 |
| 3.6 La matrice d'inertie | 20 |
| 3.7 La matrice des vitesses angulaires étendues | 20 |
| 3.8 La matrice de Coriolis | 22 |
| 3.9 La matrice de l'attraction gravitationnelle | 23 |
| 3.10 Le modèle dynamique du manipulateur | 24 |
| 3.11 Le modèle Simulink du manipulateur | 25 |
| 3.12 Validation du modèle | 26 |
| 3.12.1 Test 1 | 27 |
| 3.12.2 Test 2 | 28 |
| 3.12.3 Test 3 | 29 |
| 4 Asservissement dans l'espace articulaire | 30 |
| 4.1 Compensateur à couple-précalculé | 30 |
| 4.1.1 Le modèle du compensateur | 30 |
| 4.1.2 Choix des gains du compensateur | 31 |
| 4.1.3 Résultats | 33 |
| 4.2 Compensateur PID industriel avec compensation de gravité | 36 |
| 4.2.1 Le modèle du compensateur | 36 |
| 4.2.2 Choix des gains du compensateur | 36 |
| 4.2.3 Résultats | 37 |
| 4.3 Commande adaptative de Slotine-Li | 39 |
| 4.3.1 Le modèle du compensateur | 39 |
| 4.3.2 Choix des gains du compensateur | 42 |
| 4.3.3 Résultats | 42 |
| 4.4 Commande adaptative de Slotine-Li modifiée | 45 |
| 4.4.1 Le modèle du compensateur | 45 |
| 4.4.2 Choix des gains du compensateur | 47 |
| 4.4.3 Résultats | 47 |

| | |
|--|-----------|
| 5 Asservissement dans l'espace opérationnel | 51 |
| 5.1 Cinématique inverse | 52 |
| 5.2 Utilisation du simulateur | 53 |
| 6 Conclusion | 54 |
| 7 Bibliographie | 55 |
| 8 Annexe | 56 |
| 8.1 Figures Supplémentaires | 56 |

Table des figures

| | |
|--|----|
| 1 Manipulateur à 3 degrés de liberté | 7 |
| 2 Position initiale du manipulateur | 10 |
| 3 Centre de masse et force gravitationnelle | 23 |
| 4 Modèle Simulink du manipulateur | 25 |
| 5 Modèle Simulink de la matrice d'inertie | 26 |
| 6 Sous-module non-linéaire | 26 |
| 7 Modèle Simulink de la matrice de gravité | 26 |
| 8 Modèle Simulink de la matrice de Coriolis | 26 |
| 9 Résultats du test 1 obtenus à partir du modèle développé sur Simulink | 27 |
| 10 Résultats du test 1 obtenus à partir de Roboop | 27 |
| 11 Résultats du test 2 obtenus à partir du modèle développé sur Simulink | 28 |
| 12 Résultats du test 2 obtenus à partir de Roboop | 28 |
| 13 Résultats du test 3 obtenus à partir du modèle développé sur Simulink | 29 |
| 14 Résultats du test 3 obtenus à partir de Roboop | 29 |
| 15 Compensateur à couple-précalculé | 31 |
| 16 Modèle Simulink du compensateur à couple-précalculé | 31 |
| 17 Commande appliquée aux simulations 1 et 3 | 33 |
| 18 Commande appliquée aux simulations 2 et 4 | 33 |
| 19 Réponse à un échelon sans perturbation | 34 |
| 20 Réponse à une sinusoïdale sans perturbation | 34 |
| 21 Erreur à un échelon sans perturbation | 34 |
| 22 Erreur à une sinusoïdale sans perturbation | 34 |
| 23 Couple, échelon sans perturbation | 35 |
| 24 Couple, sinusoïdale sans perturbation | 35 |
| 25 Réponse à un échelon avec perturbation | 35 |
| 26 Réponse à une sinusoïdale avec perturbation | 35 |
| 27 Compensateur PID industriel avec compensation de la gravité | 36 |
| 28 Modèle Simulink du compensateur PID industriel avec compensation de la gravité | 36 |
| 29 Réponse à un échelon sans perturbation | 37 |
| 30 Réponse à une sinusoïdale sans perturbation | 37 |
| 31 Erreur à un échelon sans perturbation | 38 |
| 32 Erreur à une sinusoïdale sans perturbation | 38 |

| | | |
|----|--|----|
| 33 | Couple, échelon sans perturbation | 38 |
| 34 | Couple, sinusoïdale sans perturbation | 38 |
| 35 | Réponse à un échelon avec perturbation | 39 |
| 36 | Réponse à une sinusoïdale avec perturbation | 39 |
| 37 | Commande adaptative de Slotine-Li | 41 |
| 38 | Modèle Simulink de la commande adaptative de Slotine-Li | 41 |
| 39 | Paramètres pour un échelon, (sans perturbation) | 43 |
| 40 | Paramètres pour une sinusoïdale, (sans perturbation) | 43 |
| 41 | Réponse à un échelon sans perturbation | 44 |
| 42 | Réponse à une sinusoïdale sans perturbation | 44 |
| 43 | Erreurs à un échelon sans perturbation | 44 |
| 44 | Erreurs à une sinusoïdale sans perturbation | 44 |
| 45 | Couple, échelon sans perturbation | 44 |
| 46 | Couple, sinusoïdale sans perturbation | 44 |
| 47 | Réponse à un échelon avec perturbation | 45 |
| 48 | Réponse à une sinusoïdale avec perturbation | 45 |
| 49 | Commande adaptative de Slotine-Li modifiée | 46 |
| 50 | Modèle Simulink de la commande adaptative de Slotine-Li modifiée | 46 |
| 51 | Paramètres pour un échelon, (sans perturbation) | 48 |
| 52 | Paramètres pour une sinusoïdale, (sans perturbation) | 48 |
| 53 | Réponse à un échelon sans perturbation | 48 |
| 54 | Réponse à une sinusoïdale sans perturbation | 48 |
| 55 | Erreurs à un échelon sans perturbation | 49 |
| 56 | Erreurs à une sinusoïdale sans perturbation | 49 |
| 57 | Couple, échelon sans perturbation | 49 |
| 58 | Couple, sinusoïdale sans perturbation | 49 |
| 59 | Réponse à un échelon avec perturbation | 50 |
| 60 | Réponse à une sinusoïdale avec perturbation | 50 |
| 61 | Réponse à un échelon paramètres aléatoires | 50 |
| 62 | Évolution des paramètres aléatoires | 50 |
| 63 | Modèle Simulink du système complet | 51 |
| 64 | Image du Bras Robotique dans la fenêtre réalité virtuelle | 53 |
| 65 | Erreurs, échelon, Couple Pré-Calculé (avec perturbation) | 56 |
| 66 | Erreurs, sinusoïdale, Couple Pré-Calculé (avec perturbation) | 56 |
| 67 | Couple, échelon, Couple Pré-Calculé (avec perturbation) | 56 |
| 68 | Couple, sinusoïdale, Couple Pré-Calculé (avec perturbation) | 56 |
| 69 | Erreurs, échelon, PID (avec perturbation) | 57 |
| 70 | Erreurs, sinusoïdale, PID (avec perturbation) | 57 |
| 71 | Couple, échelon, PID (avec perturbation) | 57 |
| 72 | Couple, sinusoïdale, PID (avec perturbation) | 57 |
| 73 | Erreurs, échelon, Slotine-Li (avec perturbation) | 58 |
| 74 | Erreurs, sinusoïdale, Slotine-Li (avec perturbation) | 58 |
| 75 | Couple, échelon, Slotine-Li (avec perturbation) | 58 |
| 76 | Couple, sinusoïdale, Slotine-Li (avec perturbation) | 58 |
| 77 | Paramètres, échelon, Slotine-Li (avec perturbation) | 59 |

| | | |
|----|---|----|
| 78 | Paramètres, sinusoïdale, Slotine-Li (avec perturbation) | 59 |
| 79 | Erreur, échelon, Slotine-Li Modifié (avec perturbation) | 59 |
| 80 | Erreur, sinusoïdale, Slotine-Li Modifié (avec perturbation) | 59 |
| 81 | Couple, échelon, Slotine-Li Modifié (avec perturbation) | 60 |
| 82 | Couple, sinusoïdale, Slotine-Li Modifié (avec perturbation) | 60 |
| 83 | Paramètres pour un échelon, Slotine-Li Modifié (avec perturbation) | 60 |
| 84 | Paramètres pour une sinusoïdale, Slotine-Li Modifié (avec perturbation) | 60 |

Liste des tableaux

| | | |
|---|---|----|
| 1 | Paramètres du manipulateur | 8 |
| 2 | Paramètres Denavit-Hartenberg du manipulateur | 9 |
| 3 | Paramètres du manipulateur | 33 |
| 4 | Paramètres pour un échelon, Slotine-Li | 42 |
| 5 | Paramètres, Slotine-Li modifiée sans perturbation ($M_3 = 2$) | 47 |
| 6 | Paramètres, Slotine-Li modifiée avec perturbation ($M_3 = 4$) | 47 |

1 Remerciements

Le travail rapporté dans les sections 4 et 5 de ce rapport (*Asservissement dans l'espace articulaire* et *Asservissement dans l'espace opérationnel*) est le fruit d'une collaboration de l'auteur avec Jean-Julien Carrière. Je tiens à remercier sincèrement toutes les personnes qui m'ont aidé à réaliser ce travail. Un remerciement particulier à M. Romano DeSantis pour son aide, à mon directeur de recherche M. Richard Gourdeau et à ma copine Marie-Hélène Murray pour avoir bien voulu corriger les fautes d'orthographes.

Sylvain

2 Introduction

Vous trouverez dans ce rapport tout le détail de mes travaux réalisés dans le cadre du cours *Commande de systèmes robotiques* de l'école Polytechnique de Montréal. Ainsi, le modèle du manipulateur à 3 degrés de liberté a été réalisé en suivant la démarche de modélisation proposée par le professeur M. Romano De Santis. Vous trouverez donc dans ce rapport, toutes les étapes de la modélisation du manipulateur, le modèle Simulink utilisé pour valider le modèle mathématique, 4 exemples de compensateurs utilisés pour asservir en position le manipulateur ainsi qu'un exemple de contrôle dans l'espace opérationnel. De plus, je vous invite à consulter dans un premier temps les ouvrages de M. Romano DeSantis cités en référence, si vous n'êtes pas familiés avec la méthode de modélisation utilisée.

3 Modélisation d'un manipulateur à 3 degrés de liberté

3.1 Présentation du manipulateur

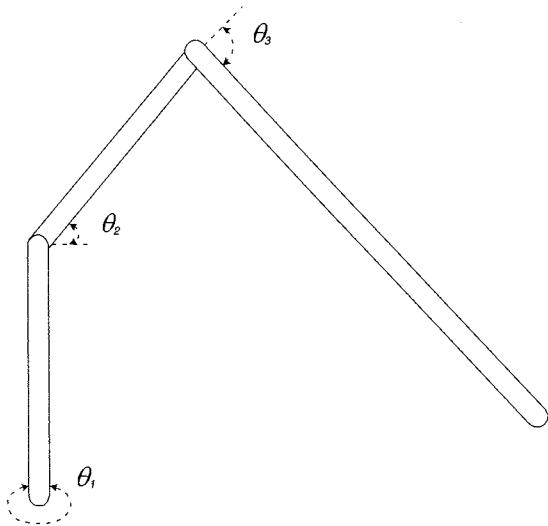


FIG. 1 – Manipulateur à 3 degrés de liberté

La figure 1 présente le schéma du manipulateur modélisé dans ce rapport. Il s'agit d'un manipulateur à trois membres qui sont reliés entre eux par des joints rotatifs. Les joints 2 et 3 du manipulateur lui permettent d'atteindre n'importe quelle position x, y située dans le plan où se trouve le manipulateur. Le joint 1 permet au manipulateur de changer l'orientation de ce plan et ainsi atteindre n'importe quelle position x, y et z autour du manipulateur. Bien entendu, puisque ce manipulateur n'a que 3 degrés de liberté, il est impossible d'imposer à la fois une position désirée et une orientation désirée. De plus, on a pris pour acquis qu'un capteur est placé sur chacun des joints rotatifs pour déterminer l'orientation du membre par rapport au précédent.

Ainsi, le vecteur de configuration \vec{q} est défini par :

$$\vec{q} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \end{bmatrix}$$

Les masses et les longueurs des segments sont des paramètres qui n'influenceront pas la modélisation qui va suivre, car elle a été réalisée indépendamment de ces paramètres. Ainsi, vous pourriez aisément changer la masse et la longueur d'un segment sans modifier le modèle dynamique final.

Voici la signification des paramètres utilisés dans cette modélisation :

| Paramètres | Description | Valeur par défaut |
|------------|-----------------------|-------------------|
| m_1 | masse du segment 1 | 1 kg |
| d_1 | longueur du segment 1 | 1 m |
| m_2 | masse du segment 2 | 1 kg |
| l_2 | longueur du segment 2 | 1 m |
| m_3 | masse du segment 3 | 2 kg |
| l_3 | longueur du segment 3 | 2 m |
| r_1 | rayon du segment 1 | 0.05 m |

TAB. 1 – Paramètres du manipulateur

3.2 Convention de Denavit-Hartenberg

Il existe 2 conventions de Denavit-Hartenberg : la convention originale et la convention modifiée. Les conventions se ressemblent beaucoup, mais la convention originale de Denavit-Hartenberg sera utilisée dans ce travail. Cette convention impose une façon bien définie de choisir les repères du système afin que chaque membrure soit représentée d'une façon similaire à la précédente et ainsi faciliter la cinématique du robot.

La convention de Denavit-Hartenberg propose de toujours utiliser 4 transformations successives pour passer du repère de la membrure i à la membrure $i + 1$. Ces 4 transformations sont les suivantes :

- Une rotation d'un angle θ autour de l'axe z .
- Une translation d'une distance d le long de l'axe z .
- Une translation d'une longueur l le long de l'axe x .
- Une rotation d'un angle α autour de l'axe x .

Ces transformations peuvent être exprimées sous la forme d'une matrice de transformations homogènes :

$${}^nT_{n+1} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \cos(\alpha) & \sin(\theta) \sin(\alpha) & l \cos(\theta) \\ \sin(\theta) & \cos(\theta) \cos(\alpha) & -\cos(\theta) \sin(\alpha) & l \sin(\theta) \\ 0 & \sin(\alpha) & \cos(\alpha) & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Cette matrice représente la transformation du repère n au repère $n + 1$. Elle sera utilisée pour déterminer la cinématique du manipulateur.

Dans le cas d'un joint rotatif, la variable de joint sera θ et dans le cas d'un joint prismatique, la variable de joint sera d . Les autres paramètres peuvent être numériques s'ils sont connus ou des variables si leur valeur n'est pas définie. Les 4 paramètres de chacune des membrures forment ce qu'on appelle les paramètres de Denavit-Hartenberg.

Afin d'établir les paramètres de Denavit-Hartenberg, chacun des repères est établi en se basant sur les règles suivantes :

- L'axe z_{i-1} est selon l'axe de mouvement du joint i .
- L'axe x_i est normal aux axes z_{i-1} et z_i , et passe par les points les plus rapprochés de ces axes.
- L'axe y_i complète le repère i en respectant la règle de la main droite.
- Les repères sont numérotés à partir de la base (repère 0) jusqu'à l'extrémité.
- Le dernier repère peut être placé arbitrairement tant que l'axe x_i est normal à l'axe z_{i-1} .

3.3 Cinématique directe

Pour déterminer la cinématique directe du manipulateur, il faut dans un premier temps déterminer les paramètres de Denavit-Hartenberg. Pour déterminer ces paramètres, il est important de placer le manipulateur dans sa position initiale, c'est-à-dire lorsque tous les angles du vecteur de configuration \vec{q} égalent 0. La figure 2 montre le manipulateur dans sa position initiale. De plus, cette figure montre le choix des repères. Le repère 0 est à la base du manipulateur et les repères sont numérotés de la base à l'extrémité du manipulateur. Les paramètres de Denavit-Hartenberg sont inscrits dans le tableau 2.

| Articulation | θ | d | l | α |
|--------------|------------|-------|-------|------------|
| 1 | θ_1 | d_1 | 0 | 90° |
| 2 | θ_2 | 0 | l_2 | 0 |
| 3 | θ_3 | 0 | l_3 | 0 |

TAB. 2 – Paramètres Denavit-Hartenberg du manipulateur

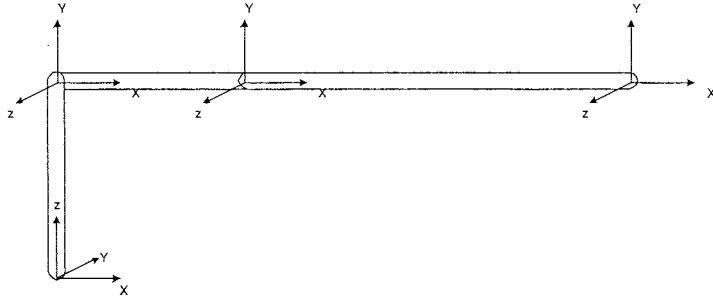


FIG. 2 – Position initiale du manipulateur

En remplaçant ces paramètres dans la matrice de transformation de Denavit-Hartenberg, on obtient les matrices suivantes :

$${}^0T_1 = \begin{bmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) & 0 \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1T_2 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & l_2 \cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & l_2 \sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2T_3 = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & l_3 \cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & l_3 \sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Les matrices de transformation de la base à la fin de la membrure 0T_2 et celle de la base à l'outil 0T_3 sont définies comme suit :

$${}^0T_2 = {}^0T_1 {}^1T_2 = \begin{bmatrix} \cos(\theta_1) \cos(\theta_2) & -\cos(\theta_1) \sin(\theta_2) & \sin(\theta_1) & \cos(\theta_1) l_2 \cos(\theta_2) \\ \sin(\theta_1) \cos(\theta_2) & -\sin(\theta_1) \sin(\theta_2) & -\cos(\theta_1) & \sin(\theta_1) l_2 \cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & l_2 \sin(\theta_2) + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0T_3 = {}^0T_1 {}^1T_2 {}^2T_3 = \begin{bmatrix} \cos(\theta_1) \cos(\theta_2 + \theta_3) & -\cos(\theta_1) \sin(\theta_2 + \theta_3) & \sin(\theta_1) & \cos(\theta_1) (l_3 \cos(\theta_2 + \theta_3) + l_2 \cos(\theta_2)) \\ \sin(\theta_1) \cos(\theta_2 + \theta_3) & -\sin(\theta_1) \sin(\theta_2 + \theta_3) & -\cos(\theta_1) & \sin(\theta_1) (l_3 \cos(\theta_2 + \theta_3) + l_2 \cos(\theta_2)) \\ \sin(\theta_2 + \theta_3) & \cos(\theta_2 + \theta_3) & 0 & l_3 \sin(\theta_2 + \theta_3) + l_2 \sin(\theta_2) + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La dernière colonne de la matrice de transformation représente la translation effectuée. Ainsi, la position de l'extrémité du manipulateur est donnée par les 3 équations de la dernière colonne.

$$\begin{aligned} {}^bT_a &= \left[\begin{array}{ccc|c} & & {}^bR_a & \begin{matrix} p_x \\ p_y \\ p_z \end{matrix} \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \\ \vec{p} &= \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} \cos(\theta_1) (l_3 \cos(\theta_2 + \theta_3) + l_2 \cos(\theta_2)) \\ \sin(\theta_1) (l_3 \cos(\theta_2 + \theta_3) + l_2 \cos(\theta_2)) \\ l_3 \sin(\theta_2 + \theta_3) + l_2 \sin(\theta_2) + d_1 \end{bmatrix} \end{aligned}$$

3.4 La matrice jacobienne étendue

La matrice jacobienne étendue sera principalement utilisée pour trouver la matrice d'inertie et la matrice de Coriolis. La matrice jacobienne étendue est en fait la superposition des trois matrices jacobienne des trois membres du robot. Ces matrices jacobienne seront représentées dans le repère de la membrure qu'elles représentent respectivement afin de simplifier les équations. De plus, ces matrices jacobienne 6×3 peuvent être divisées en deux sous-matrices jacobienne 3×3 soit la matrice jacobienne des vitesses linéaires et celle des vitesses angulaires. Ainsi, nous appellerons J_n la matrice jacobienne 6×3 de la membrure n représentée dans le repère n . J_{vbn} sera la matrice jacobienne des vitesses linéaires de la membrure n représentée dans le repère de base et J_{van} sera la même matrice représentée dans le repère de la membrure n . $J_{\Omega bn}$ sera la matrice jacobienne des vitesses angulaires de la membrure n représentée dans le repère de base et $J_{\Omega an}$ sera la même matrice représentée dans le repère de la membrure n . Pour avoir plus d'information sur la matrice jacobienne, consultez les ouvrages cités en référence ou les définitions jointes en annexe de la version originale de ce rapport.

La méthode utilisée pour déterminer ces matrices est la suivante :

1. Trouver la matrice J_{vbn} en dérivant le vecteur \vec{p} de la matrice de transformation 0T_n .
2. Convertir J_{vbn} dans le repère de la membrure n en posant $J_{van} = {}^0R_n^T J_{vbn}$.
3. Utiliser la loi de propagation $J_{\Omega bn} = J_{\Omega bn-1} + {}^0R_n \vec{w}$ où \vec{w} est un vecteur indiquant si la rotation qui s'effectue toujours (par convention) autour de l'axe z a été réalisé selon θ_1, θ_2 ou θ_3 .
4. Convertir $J_{\Omega bn}$ dans le repère de la membrure n en posant $J_{\Omega an} = {}^0R_n^T J_{\Omega bn}$.
5. Superposer les deux matrices jacobienne afin d'obtenir J_n .

3.4.1 Calcul de J_1

Déterminer 0R_1 et \vec{p}_1 :

$${}^0T_1 = \left[\begin{array}{c|ccc} {}^0R_1 & p_x & p_y & p_z \\ \hline & 1 & & \\ 0 & 0 & 0 & 1 \end{array} \right] = \left[\begin{array}{cccc} \cos(\theta_1) & 0 & \sin(\theta_1) & 0 \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{array} \right]$$

$${}^0R_1 = \left[\begin{array}{ccc} \cos(\theta_1) & 0 & \sin(\theta_1) \\ \sin(\theta_1) & 0 & -\cos(\theta_1) \\ 0 & 1 & 0 \end{array} \right] \quad \vec{p}_1 = \left[\begin{array}{c} p_x \\ p_y \\ p_z \end{array} \right] = \left[\begin{array}{c} 0 \\ 0 \\ d_1 \end{array} \right]$$

Calcul de la jacobienne de vitesse linéaire :

$$J_{vb1} = \left[\begin{array}{ccc} \frac{\partial p_x}{\partial x} & \frac{\partial p_x}{\partial y} & \frac{\partial p_x}{\partial z} \\ \frac{\partial p_y}{\partial x} & \frac{\partial p_y}{\partial y} & \frac{\partial p_y}{\partial z} \\ \frac{\partial p_z}{\partial x} & \frac{\partial p_z}{\partial y} & \frac{\partial p_z}{\partial z} \end{array} \right] = \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array} \right]$$

$$J_{va1} = {}^0R_1^T J_{vb1} = \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array} \right]$$

Calcul de la jacobienne des vitesses angulaires :

$$J_{\Omega b1} = \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{array} \right]$$

$$J_{\Omega a1} = {}^0R_1^T J_{\Omega b1} = \left[\begin{array}{ccc} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{array} \right]$$

Finalement, la matrice J_1 :

$$J_1 = \left[\begin{array}{c} J_{va1} \\ J_{\Omega a1} \end{array} \right] = \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{array} \right]$$

Ainsi,

$$\vec{v}_1 = \begin{bmatrix} v_{x1} & v_{y1} & v_{z1} & \Omega_{x1} & \Omega_{y1} & \Omega_{z1} \end{bmatrix}^T = J_1 \begin{bmatrix} \dot{\theta}_1 & \dot{\theta}_2 & \dot{\theta}_3 \end{bmatrix}^T$$

3.4.2 Calcul de J_2

Déterminer 0R_2 et \vec{p}_2 :

$$\begin{aligned} {}^0T_2 &= \left[\begin{array}{c|c} {}^0R_2 & \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \\ \hline 0 & 0 & 0 & 1 \end{array} \right] = \left[\begin{array}{cccc} \cos(\theta_1) \cos(\theta_2) & -\cos(\theta_1) \sin(\theta_2) & \sin(\theta_1) & \cos(\theta_1) l_2 \cos(\theta_2) \\ \sin(\theta_1) \cos(\theta_2) & -\sin(\theta_1) \sin(\theta_2) & -\cos(\theta_1) & \sin(\theta_1) l_2 \cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & l_2 \sin(\theta_2) + d_1 \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \\ {}^0R_2 &= \begin{bmatrix} \cos(\theta_1) \cos(\theta_2) & -\cos(\theta_1) \sin(\theta_2) & \sin(\theta_1) \\ \sin(\theta_1) \cos(\theta_2) & -\sin(\theta_1) \sin(\theta_2) & -\cos(\theta_1) \\ \sin(\theta_2) & \cos(\theta_2) & 0 \end{bmatrix} \quad \vec{p}_2 = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} \cos(\theta_1) l_2 \cos(\theta_2) \\ \sin(\theta_1) l_2 \cos(\theta_2) \\ l_2 \sin(\theta_2) + d_1 \end{bmatrix} \end{aligned}$$

Calcul de la jacobienne de vitesse linéaire :

$$\begin{aligned} J_{vb2} &= \begin{bmatrix} \frac{\partial p_x}{\partial x} & \frac{\partial p_x}{\partial y} & \frac{\partial p_x}{\partial z} \\ \frac{\partial p_y}{\partial x} & \frac{\partial p_y}{\partial y} & \frac{\partial p_y}{\partial z} \\ \frac{\partial p_z}{\partial x} & \frac{\partial p_z}{\partial y} & \frac{\partial p_z}{\partial z} \end{bmatrix} = \begin{bmatrix} -\sin(\theta_1) l_2 \cos(\theta_2) & -\cos(\theta_1) l_2 \sin(\theta_2) & 0 \\ \cos(\theta_1) l_2 \cos(\theta_2) & -\sin(\theta_1) l_2 \sin(\theta_2) & 0 \\ 0 & l_2 \cos(\theta_2) & 0 \end{bmatrix} \\ J_{va2} &= {}^0R_2^T J_{vb2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & l_2 & 0 \\ -l_2 \cos(\theta_2) & 0 & 0 \end{bmatrix} \end{aligned}$$

Calcul de la jacobienne des vitesses angulaires :

$$\begin{aligned} J_{\Omega b2} &= J_{\Omega b1} + {}^0R_2 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & \sin(\theta_1) & 0 \\ 0 & -\cos(\theta_1) & 0 \\ 1 & 0 & 0 \end{bmatrix} \\ J_{\Omega a2} &= {}^0R_2^T J_{\Omega b2} = \begin{bmatrix} \sin(\theta_2) & 0 & 0 \\ \cos(\theta_2) & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \end{aligned}$$

Finalement, la matrice J_2 :

$$J_2 = \begin{bmatrix} J_{va2} \\ J_{\Omega a2} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & l_2 & 0 \\ -l_2 \cos(\theta_2) & 0 & 0 \\ \sin(\theta_2) & 0 & 0 \\ \cos(\theta_2) & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Ainsi,

$$\vec{v}_2 = [v_{x2} \ v_{y2} \ v_{z2} \ \Omega_{x2} \ \Omega_{y2} \ \Omega_{z2}]^T = J_2 [\dot{\theta}_1 \ \dot{\theta}_2 \ \dot{\theta}_3]^T$$

3.4.3 Calcul de J_3

Déterminer 0R_3 et \vec{p}_3 :

$${}^0T_3 = \begin{bmatrix} \cos(\theta_1) \cos(\theta_2 + \theta_3) & -\cos(\theta_1) \sin(\theta_2 + \theta_3) & \sin(\theta_1) & \cos(\theta_1) (l_3 \cos(\theta_2 + \theta_3) + l_2 \cos(\theta_2)) \\ \sin(\theta_1) \cos(\theta_2 + \theta_3) & -\sin(\theta_1) \sin(\theta_2 + \theta_3) & -\cos(\theta_1) & \sin(\theta_1) (l_3 \cos(\theta_2 + \theta_3) + l_2 \cos(\theta_2)) \\ \sin(\theta_2 + \theta_3) & \cos(\theta_2 + \theta_3) & 0 & l_3 \sin(\theta_2 + \theta_3) + l_2 \sin(\theta_2) + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0R_3 = \begin{bmatrix} \cos(\theta_1) \cos(\theta_2 + \theta_3) & -\cos(\theta_1) \sin(\theta_2 + \theta_3) & \sin(\theta_1) \\ \sin(\theta_1) \cos(\theta_2 + \theta_3) & -\sin(\theta_1) \sin(\theta_2 + \theta_3) & -\cos(\theta_1) \\ \sin(\theta_2 + \theta_3) & \cos(\theta_2 + \theta_3) & 0 \end{bmatrix}$$

$$\vec{p}_3 = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} \cos(\theta_1) (l_3 \cos(\theta_2 + \theta_3) + l_2 \cos(\theta_2)) \\ \sin(\theta_1) (l_3 \cos(\theta_2 + \theta_3) + l_2 \cos(\theta_2)) \\ l_3 \sin(\theta_2 + \theta_3) + l_2 \sin(\theta_2) + d_1 \end{bmatrix}$$

Calcul de la jacobienne de vitesse linéaire :

$$J_{vb3} = \begin{bmatrix} -\sin(\theta_1) l_3 \cos(\theta_2 + \theta_3) + l_2 \cos(\theta_2) & \cos(\theta_1) (-l_3 \sin(\theta_2 + \theta_3) - l_2 \sin(\theta_2)) & -\cos(\theta_1) l_3 \sin(\theta_2 + \theta_3) \\ \cos(\theta_1) l_3 \cos(\theta_2 + \theta_3) + l_2 \cos(\theta_2) & \sin(\theta_1) (-l_3 \sin(\theta_2 + \theta_3) - l_2 \sin(\theta_2)) & -\sin(\theta_1) l_3 \sin(\theta_2 + \theta_3) \\ 0 & l_3 \cos(\theta_2 + \theta_3) + l_2 \cos(\theta_2) & l_3 \cos(\theta_2 + \theta_3) \end{bmatrix}$$

$$J_{va3} = {}^0R_3^T J_{vb3} = \begin{bmatrix} 0 & l_2 \sin(\theta_3) & 0 \\ 0 & l_3 + l_2 \cos(\theta_3) & l_3 \\ -l_3 \cos(\theta_2 + \theta_3) - l_2 \cos(\theta_2) & 0 & 0 \end{bmatrix}$$

Calcul de la jacobienne des vitesses angulaires :

$$J_{\Omega b3} = J_{\Omega b2} + {}^0 R_3 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & \sin(\theta_1) & \sin(\theta_1) \\ 0 & -\cos(\theta_1) & -\cos(\theta_1) \\ 1 & 0 & 0 \end{bmatrix}$$

$$J_{\Omega a3} = {}^0 R_3^T J_{\Omega b3} = \begin{bmatrix} \sin(\theta_2 + \theta_3) & 0 & 0 \\ \cos(\theta_2 + \theta_3) & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Finalement, la matrice J_3 :

$$J_3 = \begin{bmatrix} J_{va3} \\ J_{\Omega a3} \end{bmatrix} = \begin{bmatrix} 0 & l_2 \sin(\theta_3) & 0 \\ 0 & l_3 + l_2 \cos(\theta_3) & l_3 \\ -l_3 \cos(\theta_2 + \theta_3) - l_2 \cos(\theta_2) & 0 & 0 \\ \sin(\theta_2 + \theta_3) & 0 & 0 \\ \cos(\theta_2 + \theta_3) & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Ainsi,

$$\vec{v}_3 = [v_{x3} \ v_{y3} \ v_{z3} \ \Omega_{x3} \ \Omega_{y3} \ \Omega_{z3}]^T = J_3 [\dot{\theta}_1 \ \dot{\theta}_2 \ \dot{\theta}_3]^T$$

3.4.4 Calcul de la matrice jacobienne étendue

La matrice jacobienne étendue est simplement la superposition des matrices J_1, J_2 et J_3 . On obtient donc la matrice 18×3 suivante :

$$J_e = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & l_2 & 0 \\ -l_2 \cos(\theta_2) & 0 & 0 \\ \sin(\theta_2) & 0 & 0 \\ \cos(\theta_2) & 0 & 0 \\ 0 & 1 & 0 \\ 0 & l_2 \sin(\theta_3) & 0 \\ 0 & l_3 + l_2 \cos(\theta_3) & l_3 \\ -l_3 \cos(\theta_2 + \theta_3) - l_2 \cos(\theta_2) & 0 & 0 \\ \sin(\theta_2 + \theta_3) & 0 & 0 \\ \cos(\theta_2 + \theta_3) & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Cependant, les matrices d'inertie et de Coriolis doivent être obtenues à partir de la matrice jacobienne qui décrit la vitesse des centres de masse des membrures. Ainsi, nous aurions pu utiliser la matrice précédente si la masse de chaque membrure était concentrée en son extrémité. Toutefois, dans le cas précédent, nous avons fait l'hypothèse que la masse de chaque membrure est répartie uniformément sur toute sa longueur. Les centres de masse se situent donc au milieu de chaque membrure.

La matrice jacobienne étendue devient alors :

$$J = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{2}l_2 & 0 \\ -\frac{1}{2}l_2 \cos(\theta_2) & 0 & 0 \\ \sin(\theta_2) & 0 & 0 \\ \cos(\theta_2) & 0 & 0 \\ 0 & 1 & 0 \\ 0 & l_2 \sin(\theta_3) & 0 \\ 0 & \frac{1}{2}l_3 + l_2 \cos(\theta_3) & \frac{1}{2}l_3 \\ -\frac{1}{2}l_3 \cos(\theta_2 + \theta_3) - l_2 \cos(\theta_2) & 0 & 0 \\ \sin(\theta_2 + \theta_3) & 0 & 0 \\ \cos(\theta_2 + \theta_3) & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

3.4.5 La dérivée de la matrice jacobienne étendue

La dérivée de la matrice jacobienne étendue sera utilisée pour calculer la matrice de Coriolis. Elle est définie comme suit :

$$\dot{J} = \frac{\partial J}{\partial \theta_1} \dot{\theta}_1 + \frac{\partial J}{\partial \theta_2} \dot{\theta}_2 + \dots + \frac{\partial J}{\partial \theta_n} \dot{\theta}_n$$

Il est important d'inclure à la matrice \dot{J} les termes $\dot{\theta}_n$ résultants de la dérivée qui doivent multiplier les termes $\frac{\partial J}{\partial \theta_n}$.

On obtient donc :

$$J = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{2} \dot{\theta}_2 l_2 \sin(\theta_2) & 0 & 0 & 0 \\ \dot{\theta}_2 \cos(\theta_2) & 0 & 0 & 0 \\ -\dot{\theta}_2 \sin(\theta_2) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \dot{\theta}_3 l_2 \cos(\theta_3) & 0 & 0 \\ 0 & -\dot{\theta}_3 l_2 \sin(\theta_3) & 0 & 0 \\ \dot{\theta}_2 \left(\frac{1}{2} l_3 \sin(\theta_2 + \theta_3) + l_2 \sin(\theta_2) \right) + \frac{1}{2} \dot{\theta}_3 l_3 \sin(\theta_2 + \theta_3) & 0 & 0 & 0 \\ \dot{\theta}_2 \cos(\theta_2 + \theta_3) + \dot{\theta}_3 \cos(\theta_2 + \theta_3) & 0 & 0 & 0 \\ -\dot{\theta}_2 \sin(\theta_2 + \theta_3) - \dot{\theta}_3 \sin(\theta_2 + \theta_3) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

3.5 La matrice de masse étendue

La matrice de masse étendue M sera utilisée pour calculer les matrices d'inertie et de Coriolis. Cette matrice est une concaténation des matrices de masse de chacun des membres.

La matrice des masses étendues M est définie comme suit :

$$M = \begin{bmatrix} M_1 & 0 & \dots & 0 \\ 0 & M_2 & \dots & 0 \\ \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & M_n \end{bmatrix}$$

Les dimensions de la matrice M sont donc de $6n \times 6n$.

La matrice M_n est la matrice de masse d'une membrure et elle est définie de la façon suivante :

$$M_n = \begin{bmatrix} m_n & 0 & 0 & 0 & 0 & 0 \\ 0 & m_n & 0 & 0 & 0 & 0 \\ 0 & 0 & m_n & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{xx} & -I_{xy} & -I_{xz} \\ 0 & 0 & 0 & -I_{xy} & I_{yy} & -I_{yz} \\ 0 & 0 & 0 & -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}$$

où m_n est la masse de la membrure et I_{xx} , I_{yy} et I_{zz} sont les moments d'inertie de la membrure au centre de masse autour des axes x, y et z.

Dans le cas présent, ces axes coincident avec les axes du repère de la membrure, donc les moments d'inertie I_{xy} , I_{xz} et I_{yz} égalent 0. De plus, le rayon des membrures est petit et le moment d'inertie qu'il engendre sera négligé dans le modèle.

Voici les 3 matrices de masse :

$$M_1 := \begin{bmatrix} m_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & m_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} m_1 r_1^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$M_2 := \begin{bmatrix} m_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & m_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{12} m_2 l_2^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{12} m_2 l_2^2 \end{bmatrix}$$

$$M_3 := \begin{bmatrix} m_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & m_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{12} m_3 l_3^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{12} m_3 l_3^2 \end{bmatrix}$$

La matrice M a donc une dimension de 18×18 et elle est égale à :

$$M = \begin{bmatrix} M_1 & 0 & 0 \\ 0 & M_2 & 0 \\ 0 & 0 & M_3 \end{bmatrix}$$

3.6 La matrice d'inertie

La matrice d'inertie D est un élément clef du modèle dynamique du manipulateur. Elle est définie par l'équation suivante :

$$D = J^T M J$$

où J est la matrice jacobienne étendue (les matrices jacobiniennes de toutes les membrures empilées les unes par dessus les autres) et M est la matrice des masses étendues.

En effectuant le calcul, on obtient la matrice d'inertie D suivante :

$$D = \begin{bmatrix} D_{11} & 0 & 0 \\ 0 & \frac{1}{3} m_2 l_2^2 + l_2^2 m_3 + \frac{1}{3} m_3 l_3^2 + l_3 m_3 l_2 \cos(\theta_3) & \frac{1}{3} m_3 l_3^2 + \frac{1}{2} l_3 m_3 l_2 \cos(\theta_3) \\ 0 & \frac{1}{3} m_3 l_3^2 + \frac{1}{2} l_3 m_3 l_2 \cos(\theta_3) & \frac{1}{3} m_3 l_3^2 \end{bmatrix}$$

$$D_{11} = \frac{1}{2} m_1 r_1^2 + \frac{1}{6} m_2 l_2^2 (\cos(2\theta_2) + 1) + \frac{1}{6} m_3 l_3^2 (\cos(2\theta_2 + 2\theta_3) + 1) + \frac{1}{2} l_3 m_3 l_2 (\cos(\theta_3) + \cos(2\theta_2 + \theta_3)) + \frac{1}{2} m_3 l_2^2 (\cos(2\theta_2) + 1)$$

Si on pose les valeurs suivantes pour différents paramètres, on obtient une matrice d'inertie D identique à celle obtenue numériquement par *Roboop*, une librairie C++ développé par M. Richard Gourdeau, ce qui valide les calculs effectués jusqu'à maintenant.

$$m_1 = 1kg \quad m_2 = 1kg \quad m_3 = 2kg \quad d_1 = 1m \quad l_2 = 1m \quad l_3 = 2m \quad r_1 = 0.05m \quad \theta_1 = 0 \quad \theta_2 = 0 \quad \theta_3 = 0$$

$$D = \begin{bmatrix} 9.001250000 & 0 & 0 \\ 0 & 9. & 4.666666667 \\ 0 & 4.666666667 & 2.666666667 \end{bmatrix}$$

3.7 La matrice des vitesses angulaires étendues

La matrice des vitesses angulaires étendues W sera utilisée pour calculer la matrice de Coriolis. La matrice des vitesses angulaires étendues W est égale à :

$$W = \begin{bmatrix} W_1 & 0 & \dots & 0 \\ 0 & W_2 & \dots & 0 \\ \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & W_n \end{bmatrix}$$

Les dimensions de la matrice W sont de $6n \times 6n$.

La matrice W_n est la matrice des vitesses angulaires d'une membrure. Les vitesses angulaires Ω_x, Ω_y et Ω_z sont les vitesses angulaires de la membrure par rapport au référentiel de la membrure. Elles peuvent être obtenues à l'aide de la seconde partie de la matrice jacobienne de la membrure. Dans le cas d'un manipulateur robotique, les vitesses angulaires Ω_x, Ω_y et Ω_z sont en fonction des variables de joint θ_n et de leur dérivée $\dot{\theta}_n$. Il est donc important d'inclure la dérivée de la variable joint $\dot{\theta}_n$ à la vitesse angulaire Ω .

La matrice W_n est définie comme suit :

$$W_n = \begin{bmatrix} 0 & -\Omega_z & \Omega_y & 0 & 0 & 0 \\ \Omega_z & 0 & -\Omega_x & 0 & 0 & 0 \\ -\Omega_y & \Omega_x & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\Omega_z & \Omega_y \\ 0 & 0 & 0 & \Omega_z & 0 & -\Omega_x \\ 0 & 0 & 0 & -\Omega_y & \Omega_x & 0 \end{bmatrix}$$

Les vitesses angulaires peuvent être obtenues à partir de la matrice jacobienne comme suit :

$$\vec{\Omega}_n = J_{\Omega an} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}$$

On obtient donc :

$$\vec{\Omega}_1 = \begin{bmatrix} 0 \\ \dot{\theta}_1 \\ 0 \end{bmatrix} \quad \vec{\Omega}_2 = \begin{bmatrix} \sin(\theta_2) \dot{\theta}_1 \\ \cos(\theta_2) \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \quad \vec{\Omega}_3 = \begin{bmatrix} \sin(\theta_2 + \theta_3) \dot{\theta}_1 \\ \cos(\theta_2 + \theta_3) \dot{\theta}_1 \\ \dot{\theta}_2 + \dot{\theta}_3 \end{bmatrix}$$

À partir de ces vitesses angulaires, on peut obtenir la matrice des vitesses angulaires de chacune des membrures :

$$W_1 = \begin{bmatrix} 0 & 0 & \dot{\theta}_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\dot{\theta}_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dot{\theta}_1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\dot{\theta}_1 & 0 & 0 \end{bmatrix}$$

$$W_2 = \begin{bmatrix} 0 & -\dot{\theta}_2 & \cos(\theta_2) \dot{\theta}_1 & 0 & 0 & 0 \\ \dot{\theta}_2 & 0 & -\sin(\theta_2) \dot{\theta}_1 & 0 & 0 & 0 \\ -\cos(\theta_2) \dot{\theta}_1 & \sin(\theta_2) \dot{\theta}_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\dot{\theta}_2 & \cos(\theta_2) \dot{\theta}_1 \\ 0 & 0 & 0 & \dot{\theta}_2 & 0 & -\sin(\theta_2) \dot{\theta}_1 \\ 0 & 0 & 0 & -\cos(\theta_2) \dot{\theta}_1 & \sin(\theta_2) \dot{\theta}_1 & 0 \end{bmatrix}$$

$$W_3 = \begin{bmatrix} 0 & -\dot{\theta}_2 - \dot{\theta}_3 & \cos(\theta_2 + \theta_3) \dot{\theta}_1 & 0 & 0 & 0 \\ \dot{\theta}_2 + \dot{\theta}_3 & 0 & -\sin(\theta_2 + \theta_3) \dot{\theta}_1 & 0 & 0 & 0 \\ -\cos(\theta_2 + \theta_3) \dot{\theta}_1 & \sin(\theta_2 + \theta_3) \dot{\theta}_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\dot{\theta}_2 - \dot{\theta}_3 & \cos(\theta_2 + \theta_3) \dot{\theta}_1 \\ 0 & 0 & 0 & \dot{\theta}_2 + \dot{\theta}_3 & 0 & -\sin(\theta_2 + \theta_3) \dot{\theta}_1 \\ 0 & 0 & 0 & -\cos(\theta_2 + \theta_3) \dot{\theta}_1 & \sin(\theta_2 + \theta_3) \dot{\theta}_1 & 0 \end{bmatrix}$$

La matrice des vitesses angulaires étendues W est donc une matrice 18×18 et elle est égale à :

$$W = \begin{bmatrix} W_1 & 0 & 0 \\ 0 & W_2 & 0 \\ 0 & 0 & W_3 \end{bmatrix}$$

3.8 La matrice de Coriolis

La matrice de Coriolis est au sens physique la somme de l'effet de la force centrifuge et de l'accélération de Coriolis. Cette matrice 3×3 peut être calculée à partir de l'équation suivante :

$$C = J^T W M J + J^T M J$$

Dans le modèle dynamique, la matrice de Coriolis est multipliée par \vec{q} . Une fois cette multiplication réalisée, la matrice de Coriolis devient plutôt un vecteur colonne 3×1 dont les équations sont plus simples et surtout plus compactes.

Ainsi, on obtient la matrice de Coriolis suivante :

$$C =$$

$$\left[\left(-\frac{1}{3} l_2^2 m_2 \sin(2\theta_2) - \frac{1}{3} m_3 l_3^2 \sin(2\theta_2 + 2\theta_3) - m_3 l_3 l_2 \sin(2\theta_2 + \theta_3) - m_3 l_2^2 \sin(2\theta_2) \right) \dot{\theta}_2 \dot{\theta}_1 \right.$$

$$+ \left(-\frac{1}{3} m_3 l_3^2 \sin(2\theta_2 + 2\theta_3) - \frac{1}{2} m_3 l_3 l_2 \sin(2\theta_2 + \theta_3) - \frac{1}{2} l_2 \sin(\theta_3) m_3 l_3 \right) \dot{\theta}_3 \dot{\theta}_1 \left. \right]$$

$$\left[\left(\frac{1}{6} l_2^2 m_2 \sin(2\theta_2) + \frac{1}{2} m_3 l_3 l_2 \sin(2\theta_2 + \theta_3) + \frac{1}{2} m_3 l_2^2 \sin(2\theta_2) + \frac{1}{6} m_3 l_3^2 \sin(2\theta_2 + 2\theta_3) \right) \dot{\theta}_1^2 \right.$$

$$- l_3 m_3 l_2 \sin(\theta_3) \dot{\theta}_3 \dot{\theta}_2 - \frac{1}{2} l_3 m_3 l_2 \sin(\theta_3) \dot{\theta}_3^2 \left. \right]$$

$$\left[\left(\frac{1}{6} m_3 l_3^2 \sin(2\theta_2 + 2\theta_3) + \frac{1}{4} m_3 l_3 l_2 \sin(2\theta_2 + \theta_3) + \frac{1}{4} l_2 \sin(\theta_3) m_3 l_3 \right) \dot{\theta}_1^2 + \frac{1}{2} l_3 m_3 l_2 \sin(\theta_3) \dot{\theta}_2^2 \right]$$

3.9 La matrice de l'attraction gravitationnelle

Nous allons déduire la matrice de gravité 3×1 à partir du schéma du robot.

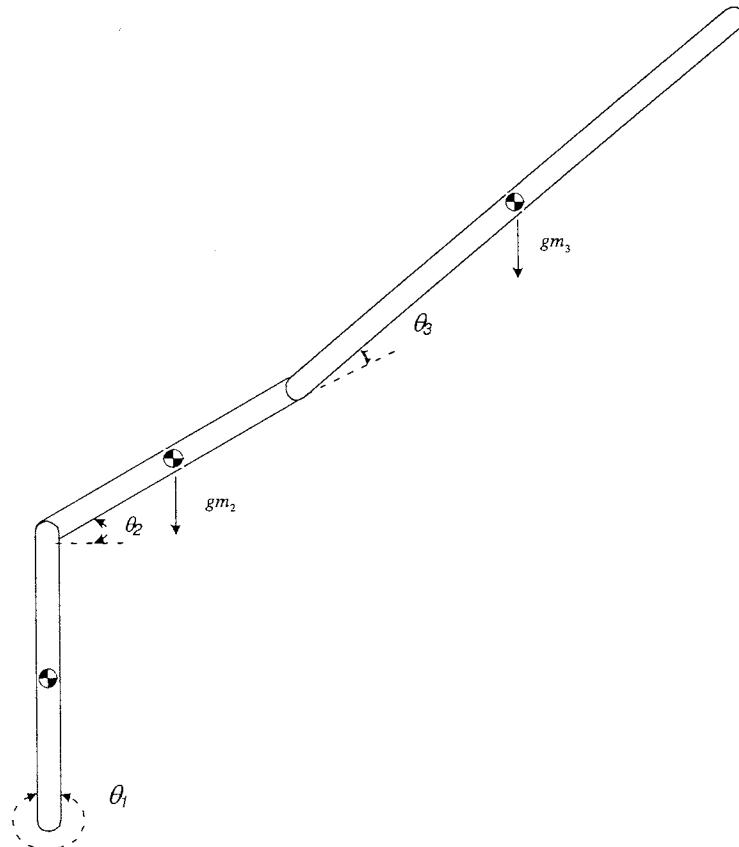


FIG. 3 – Centre de masse et force gravitationnelle

On peut donc déduire la matrice suivante :

$$G = \begin{bmatrix} 0 \\ \frac{1}{2} g \cos(\theta_2 + \theta_3) m_3 l_3 + g \left(m_3 l_2 \cos(\theta_2) + \frac{1}{2} m_2 l_2 \cos(\theta_2) \right) \\ \frac{1}{2} g \cos(\theta_2 + \theta_3) m_3 l_3 \end{bmatrix}$$

3.10 Le modèle dynamique du manipulateur

Maintenant, nous avons calculé toutes les matrices essentielles pour définir le modèle dynamique du manipulateur. À titre de rappel, nous avons calculé les matrices d'inertie D et de Coriolis C à partir des relations suivantes :

$$D = J^T M J \quad C = J^T W M J + J^T M \dot{J}$$

où J est la matrice Jacobienne étendue qui exprime la vitesse des centres de masse dans le repère de chacune des membrures en fonction de la variation des variables de joint, \dot{J} est la dérivée par rapport au temps de cette matrice jacobienne, M est la matrice de masse étendue et W est la matrice des vitesses angulaires étendues.

Ainsi, nous avons l'ensemble des matrices nécessaires pour exprimer la dynamique du manipulateur :

$$D(\vec{q})\ddot{\vec{q}} = \vec{\tau} - C(\vec{q}, \dot{\vec{q}})\dot{\vec{q}} - G(\vec{q})$$

où $D(\vec{q})$ est la matrice d'inertie exprimée en fonction de \vec{q} , $\vec{\tau}$ est le vecteur des couples appliqués à chaque articulation, $C(\vec{q}, \dot{\vec{q}})$ est la matrice de Coriolis exprimée en fonction de \vec{q} et $\dot{\vec{q}}$ (dans notre cas, la matrice C présentée précédamment est égale à $C(\vec{q}, \dot{\vec{q}})\dot{\vec{q}}$) et G la matrice de la force gravitationnelle est exprimée en fonction de \vec{q} .

Le vecteur \vec{q} est bien entendu le vecteur de configuration, c'est-à-dire le vecteur des variables de joint.

Dans notre cas :

$$\vec{q} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} \quad \dot{\vec{q}} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \quad \ddot{\vec{q}} = \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix}$$

Finalement, le modèle d'état du manipulateur est donné par :

$$\ddot{\vec{q}} = D^{-1}(\vec{\tau} - C\dot{\vec{q}} - G)$$

3.11 Le modèle Simulink du manipulateur

Pour simuler le modèle mathématique que nous venons d'obtenir, nous pouvons utiliser le modèle *Simulink* de la figure 4.

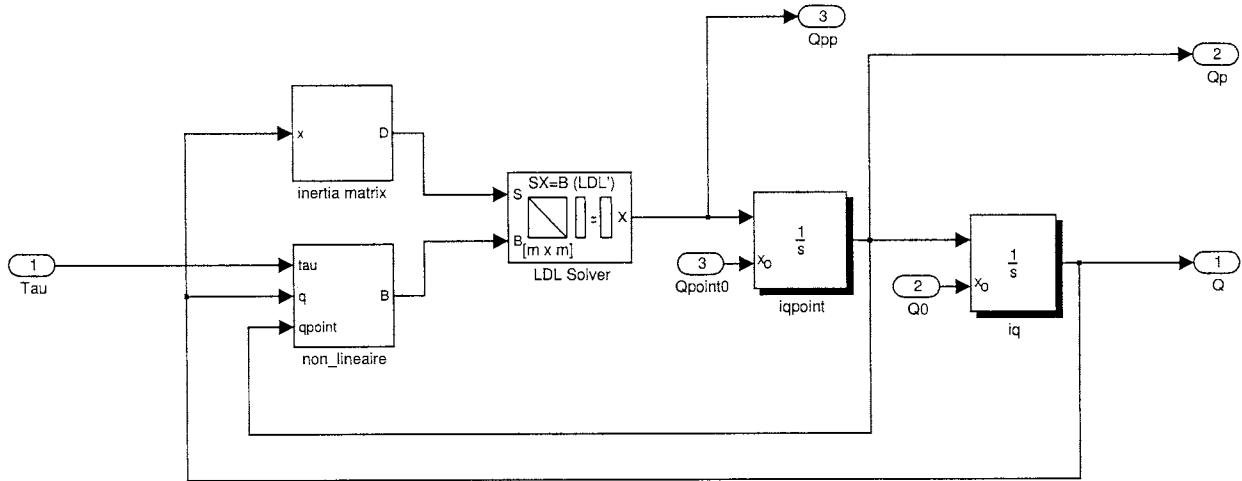


FIG. 4 – Modèle Simulink du manipulateur

Le sous-module *inertia matrix* (voir figure 5) renferme le calcul de la matrice d'inertie en fonction du vecteur \vec{q} . Le sous-module *non-linéaire* (voir figure 6) renferme le calcul suivant :

$$\vec{B} = \vec{\tau} - C\vec{q} - G$$

La matrice B est en fait un vecteur colonne puisque les matrices C et G sont aussi des matrices 3×1 . Le module *LDL solver* inverse la matrice D et il la multiplie par le vecteur \vec{B} ce qui nous permet d'obtenir le vecteur \vec{q} soit l'accélération angulaire de chaque membrure. Ensuite, on intègre 2 fois le vecteur \vec{q} afin d'obtenir les vecteurs \vec{q}_0 et \vec{q} . Les vecteurs q_0 et q_{point0} sont les conditions initiales du système soit respectivement les angles de départ et les vitesses de départ de chacun des membres. Le module de manipulateur prend donc comme entrées des vecteurs constants pour les valeurs de q_0 et q_{point0} ainsi qu'un vecteur τ représentant le couple fourni par les moteurs placés à chacun des joints du manipulateur.

Comme vous le voyez, les matrices sont définies par une série de boîtes de calcul redimensionnées sous forme de matrice. Chacune de ces boîtes renferme l'équation mathématique d'un champ de la matrice en fonction de \vec{q} et $\vec{\dot{q}}$.

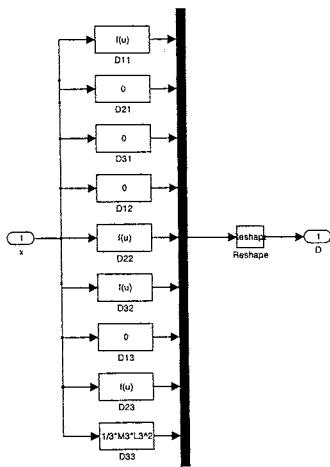


FIG. 5 – Modèle Simulink de la matrice d'inertie

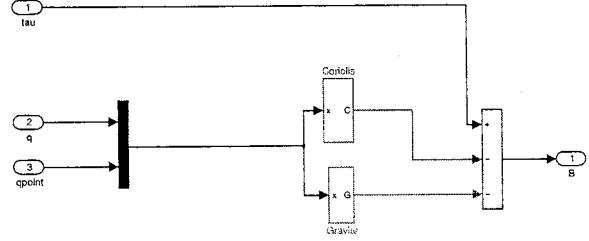


FIG. 6 – Sous-module non-linéaire

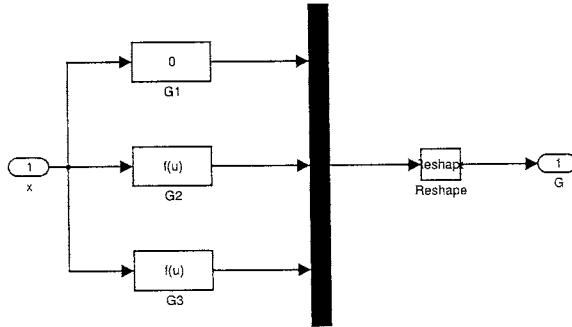


FIG. 7 – Modèle Simulink de la matrice de gravité

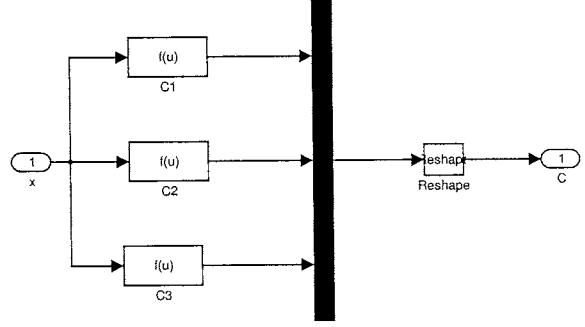


FIG. 8 – Modèle Simulink de la matrice de Coriolis

3.12 Validation du modèle

Pour valider le modèle, le manipulateur a été placé dans une position initiale puis relâché pour voir son comportement oscillatoire. Ainsi, aucun couple $\vec{\tau}$ n'est appliqué par les moteurs du manipulateur, il ne subit que l'effet de la gravité. Pour valider le comportement du manipulateur, le même test a été simulé à l'aide de la librairie Roboop écrite en C++ par M. Richard Gourdeau. Je ne m'attarderai pas sur le fonctionnement de la librairie Roboop puisque son fonctionnement est plus complexe que Simulink et qu'elle est présentement en développement pour permettre une interface plus facile à utiliser pour les non-initiés à la programmation en C++. Une dernière chose, il est important de vous rappeler que $\theta_2 = 0$ lorsque le membre est à l'horizontale.

Les paramètres utilisés pour effectuer les 3 tests qui suivent sont les suivants :

$$m_1 = 1kg \quad m_2 = 1kg \quad m_3 = 2kg \quad d_1 = 1m \quad l_2 = 1m \quad l_3 = 2m \quad r_1 = 0.05m$$

3.12.1 Test 1

Voici les conditions initiales :

$$\vec{q}_0 = \begin{bmatrix} 0 \\ -\pi/2 \\ -\pi/4 \end{bmatrix} \quad \dot{\vec{q}}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

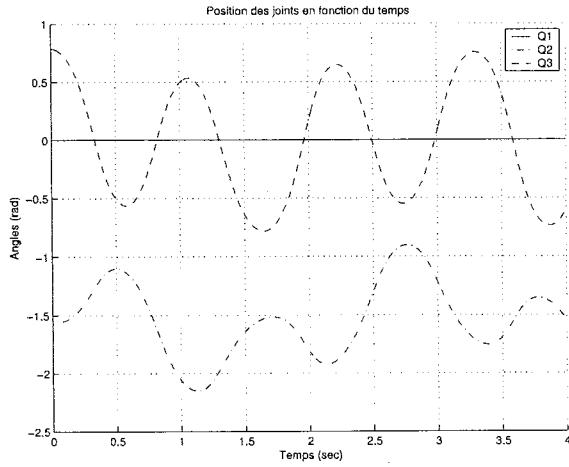


FIG. 9 – Résultats du test 1 obtenus à partir du modèle développé sur Simulink

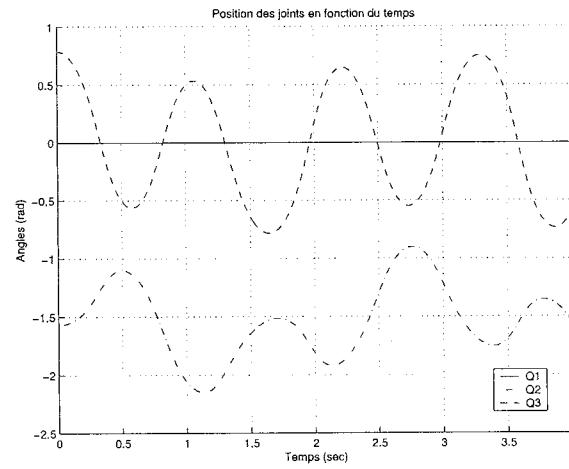


FIG. 10 – Résultats du test 1 obtenus à partir de Roboop

On remarque que les résultats sont identiques. Il est donc peu probable que notre modèle soit erroné.

3.12.2 Test 2

Voici les conditions initiales :

$$\vec{q}_0 = \begin{bmatrix} 0 \\ -\pi/4 \\ 0 \end{bmatrix} \quad \dot{\vec{q}}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

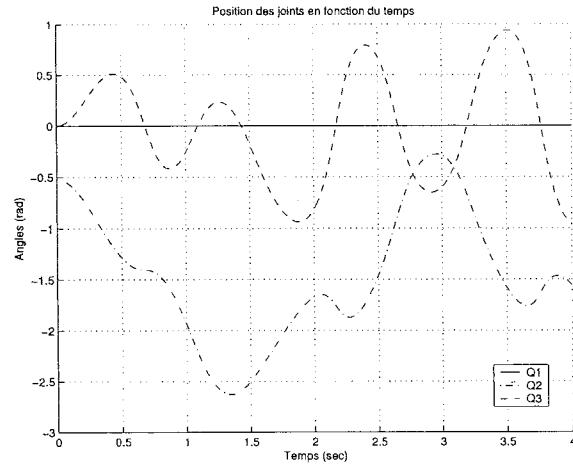


FIG. 11 – Résultats du test 2 obtenus à partir du modèle développé sur Simulink

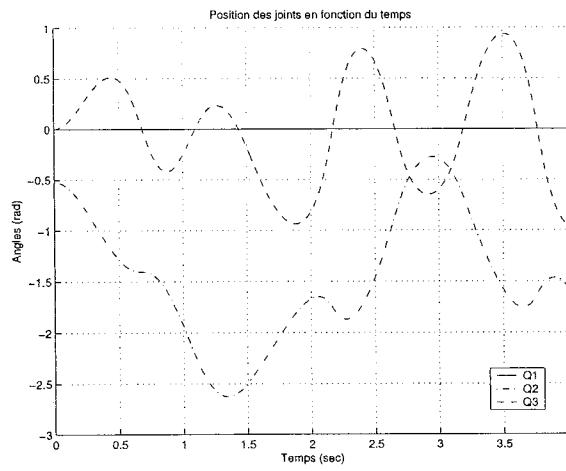


FIG. 12 – Résultats du test 2 obtenus à partir de Roboop

On remarque encore une fois que les résultats sont identiques.

3.12.3 Test 3

Voici les conditions initiales :

$$\vec{q}_0 = \begin{bmatrix} 0 \\ -\pi/2 \\ \pi/6 \end{bmatrix} \quad \dot{\vec{q}}_0 = \begin{bmatrix} 4 \\ 0 \\ 0 \end{bmatrix}$$

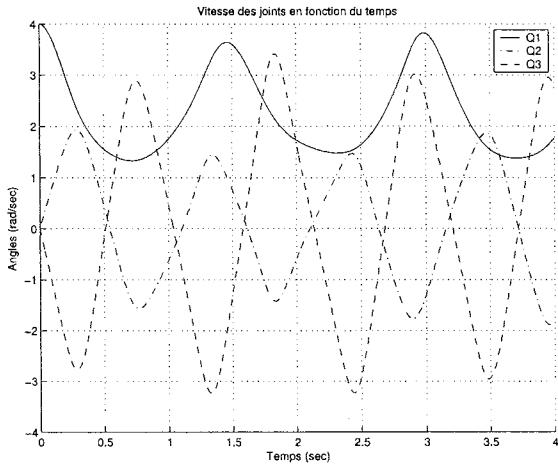


FIG. 13 – Résultats du test 3 obtenus à partir du modèle développé sur Simulink

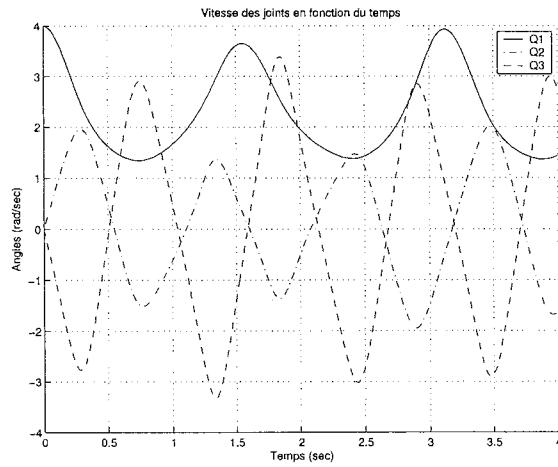


FIG. 14 – Résultats du test 3 obtenus à partir de Roboop

Dans ce test, le manipulateur tourne sur lui-même à une vitesse de 4 rad/sec lorsque les segments 2 et 3 sont relâchés. On remarque que les résultats sont légèrement différents. Cela est attribuable à une petite différence dans la façon de calculer le moment d'inertie du segment 1. Toutefois, comme on peut le constater, ce moment d'inertie a peu d'influence.

4 Asservissement dans l'espace articulaire

Cette section du travail a été élaborée conjointement par Jean-Julien Carrière et moi-même afin de rallier nos forces pour produire un travail de plus grande envergure et parfaire davantage nos connaissances. À partir du modèle développé précédemment, nous avons implanté quatre compensateurs différents soit : un compensateur à couple pré-calcué, un compensateur PID industriel, la commande adaptative de Slotine-Li et la commande adaptative de Slotine-Li modifiée. La section qui suit traite en détail de chacun de ces compensateurs et les compare entre eux.

Nous verrons dans cette section différents tests que nous avons effectués dans l'espace articulaire. C'est à dire que nous avons directement utilisé les angles désirés comme variable d'entrée. Ceci nous a permis de bien comprendre le comportement de chacun des contrôleurs.

4.1 Compensateur à couple-précalculé

4.1.1 Le modèle du compensateur

Dans un premier temps, nous avons réalisé un compensateur à couple pré-calcué. Ce compensateur est simple à réaliser lorsqu'on connaît la dynamique du système et il offre des performances très surprenantes. En effet, il utilise la dynamique du système pour calculer le couple nécessaire pour atteindre l'accélération désirée. Dans le cas de notre manipulateur, le compensateur à couple pré-calcué offre une compensation des couples perturbateurs engendrés par l'effet de Coriolis et de la gravité. De plus, l'accélération désirée, calculée à partir d'un simple PID, est multipliée par la matrice d'inertie du robot afin de découpler parfaitement les entrées du système et calculer ainsi les couples nécessaires pour atteindre l'accélération désirée de chacune des articulations sans affecter l'accélération des autres articulations. Ce sont les deux grandes propriétés du compensateur à couple pré-calcué : commande parfaitement découplée et compensation de l'effet de Coriolis et de la gravité. La figure 15 présente le modèle théorique d'un compensateur à couple pré-calcué et la figure 16 présente le modèle Simulink du compensateur que nous avons réalisé.

Bien entendu, ce compensateur fonctionne très bien si on connaît parfaitement la dynamique et les paramètres du robot. Pour illustrer ce comportement, nous avons fait les tests de deux façons différentes : en utilisant les paramètres exacts du système et en changeant la masse de la membrure 3 pour simuler les répercussions sur le robot d'une masse soulevée à l'extrémité de la membrure 3. En effet, lors des tests avec perturbation, le contrôleur calcule la compensation de la gravité et de l'effet de Coriolis en croyant que la masse de la membrure 3 est de 2 kg quand en fait elle est de 4 kg. Vous verrez les résultats ainsi qu'une discussion dans la section suivante du rapport.

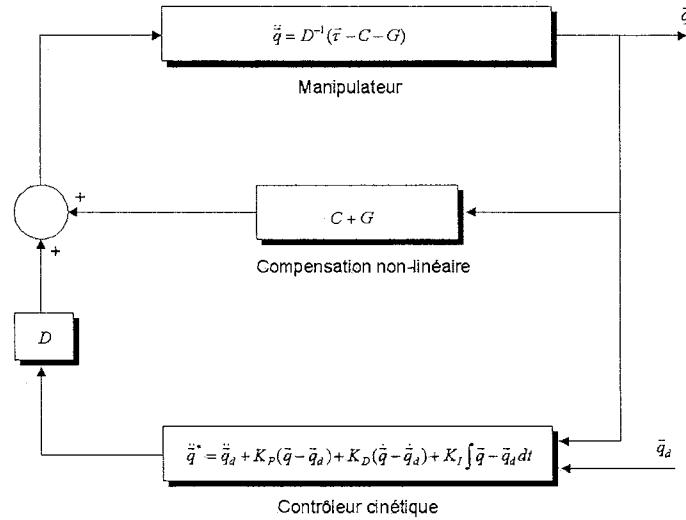


FIG. 15 – Compensateur à couple-précalculé

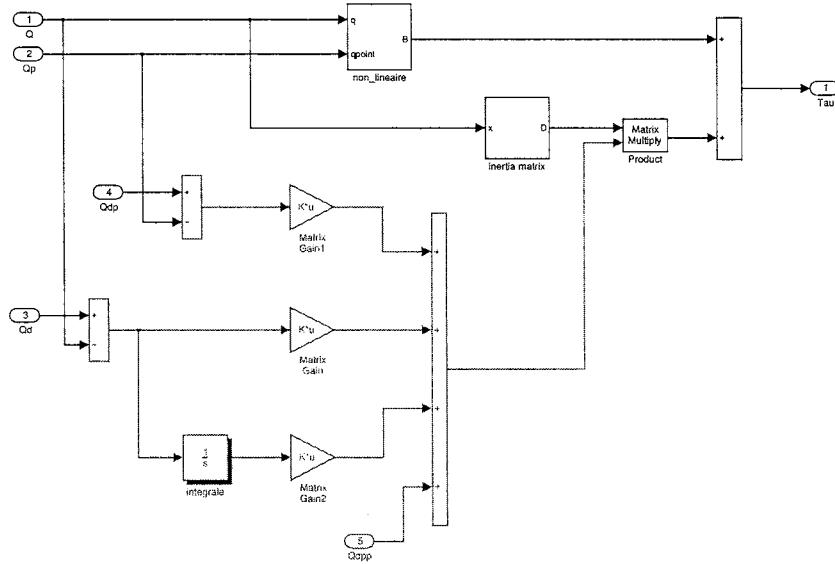


FIG. 16 – Modèle Simulink du compensateur à couple-précalculé

4.1.2 Choix des gains du compensateur

Avant de vous présenter les résultats, il est important de déterminer les gains du système. Dans un premier temps, nous avons calculé les gains en plaçant les pôles du compensateur.

La loi de commande du contrôleur cinétique est la suivante :

$$\ddot{q}^* = \ddot{q}_d + K_P(\ddot{q}_d - \ddot{q}) + K_D(\dot{q}_d - \dot{q}) + K_I \int (\ddot{q}_d - \ddot{q}) dt$$

En posant $\vec{\varepsilon} = \vec{q}_d - \vec{q}$ et en posant l'hypothèse que $\vec{q}^* = \vec{q}$, on obtient :

$$0 = \ddot{\vec{\varepsilon}} + K_P \vec{\varepsilon} + K_D \dot{\vec{\varepsilon}} + K_I \int \vec{\varepsilon} dt$$

On effectue la transformée de Laplace :

$$0 = s^2 \vec{\varepsilon} + K_P \vec{\varepsilon} + s K_D \vec{\varepsilon} + \frac{K_I \vec{\varepsilon}}{s}$$

$$0 = s^3 \vec{\varepsilon} + s K_P \vec{\varepsilon} + s^2 K_D \vec{\varepsilon} + K_I \vec{\varepsilon}$$

Les pôles p_1 , p_2 et p_3 doivent satisfaire l'équation suivante :

$$0 = (s - p_1)(s - p_2)(s - p_3) = s^3 - (p_1 + p_2 + p_3)s^2 + (p_1p_2 + p_1p_3 + p_2p_3)s - p_1p_2p_3$$

Ainsi, on peut déduire que :

$$K_P = p_1p_2 + p_1p_3 + p_2p_3 \quad K_D = -(p_1 + p_2 + p_3) \quad K_I = -p_1p_2p_3$$

Les performances recherchées sont les suivantes :

- Un temps de réponse à 2% pour un échelon de 1 rad inférieur à 1 sec.
- Un dépassement le plus faible possible.
- Une erreur en régime permanent nulle.
- La meilleure résistance possible aux perturbations.

Ainsi, on peut se dire que la constante de temps doit être d'environ 0.5 sec, ce qui correspond à des pôles placés à -2.

On obtient alors les gains suivants :

$$K_P = 12 \quad K_D = 6 \quad K_I = 8$$

Les performances avec ces gains sont très désastreuses. Le gain intégral K_I est beaucoup trop grand ce qui entraîne un dépassement énorme. De plus, le temps de réponse est beaucoup trop grand.

Nous avons donc choisi par essais et erreurs de nouveaux gains :

$$K_P = 120 \quad K_D = 20 \quad K_I = 10$$

Vous verrez dans la section des résultats les performances du compensateur utilisant ces gains.

| Paramètres | Descriptions | Valeurs |
|------------|-----------------------|---------|
| m_1 | masse du segment 1 | 1 kg |
| d_1 | longueur du segment 1 | 1 m |
| m_2 | masse du segment 2 | 1 kg |
| l_2 | longueur du segment 2 | 1 m |
| m_3 | masse du segment 3 | 2 kg |
| l_3 | longueur du segment 3 | 2 m |
| r_1 | rayon du segment 1 | 0.05 m |

TAB. 3 – Paramètres du manipulateur

4.1.3 Résultats

Quatre tests différents, deux tests sans perturbation et deux tests avec perturbation, ont été effectués à partir de ce compensateur.

La première simulation permet de voir la réponse du système à 3 échelons de 1 rad ($q_{1d}=1$ à $t=0$ sec, $q_{2d}=1$ à $t=2$ sec et $q_{3d}=1$ à $t=4$ sec). Dans la seconde simulation, on peut voir le manipulateur suivre 3 sinusoïdes qui commencent aux temps 0 sec, 2 sec et 4 sec. Les figures 17 et 18 présentent la commande appliquée au compensateur soit les vecteurs \vec{q}_d .

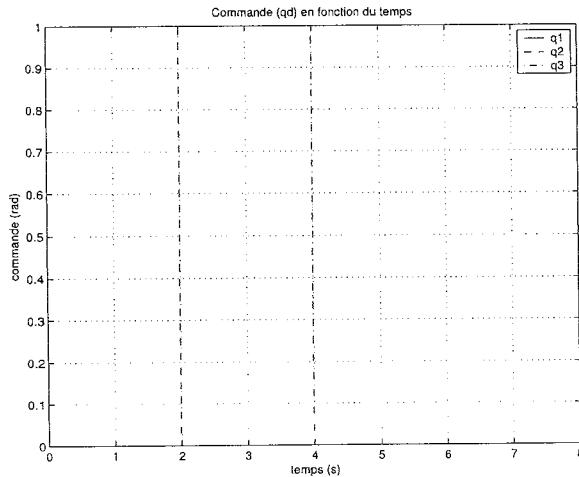


FIG. 17 – Commande appliquée aux simulations 1 et 3

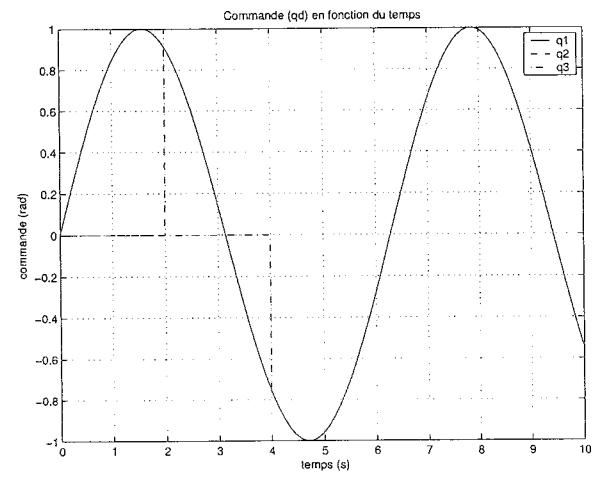


FIG. 18 – Commande appliquée aux simulations 2 et 4

Les mêmes commandes sont appliquées au compensateur pour les simulations 3 et 4 sauf que cette fois la masse du membre 3 a été changée. Le compensateur croit que la masse est de 2 kg alors quelle est maintenant de 4 kg. Cela permet de simuler le comportement du robot lorsque celui-ci soulève un objet.

Vous trouverez donc dans l'ordre les résultats aux simulations 1, 2, 3 et 4.

Comme nous avons discuté dans la section précédente, le couple pré-calculé devrait être plus performant que le PID classique, car en plus de découpler les mouvements des bras, il tient compte des différents effets externes non-linéaires comme la force de Coriolis et la force de Gravité.

Les figures 19 et 20 montrent la position Q en fonction du temps. On note tout d'abord que le temps de réponse est très court à environ 0,5 sec. En régime permanent, l'erreur de position est pratiquement nulle soit d'environ 10^{-3} . Les figures 21 et 22 illustrent le comportement de l'erreur en fonction du temps.

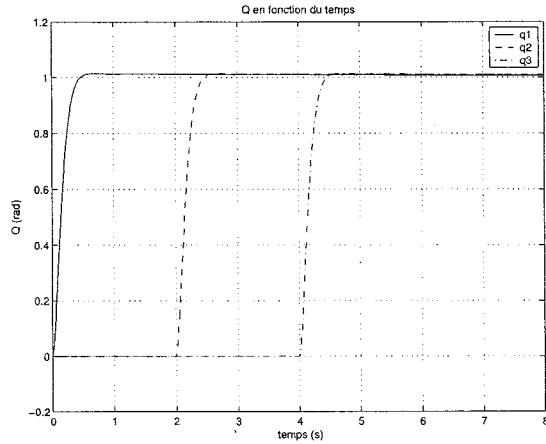


FIG. 19 – Réponse à un échelon sans perturbation

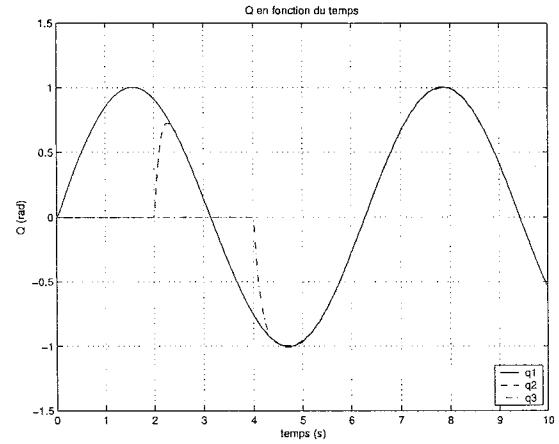


FIG. 20 – Réponse à une sinusoïdale sans perturbation

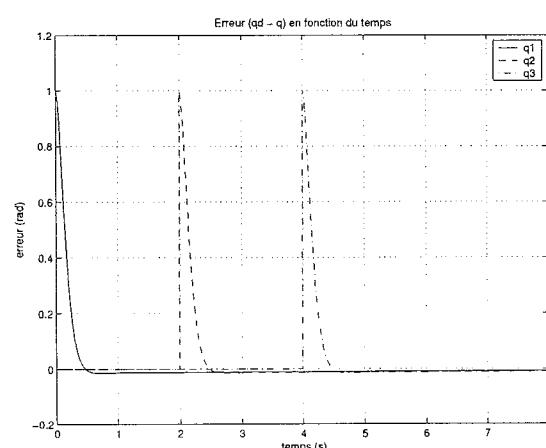


FIG. 21 – Erreur à un échelon sans perturbation

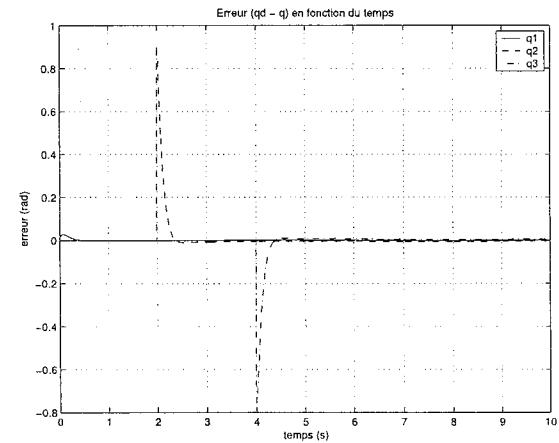


FIG. 22 – Erreur à une sinusoïdale sans perturbation

Les figures 23 et 24 montrent le couple (τ) calculé par le contrôleur à couple pré-calculé en fonction du temps pour chacune des commandes. On remarque que les couples calculés sont

très grands. En pratique, il faudrait réduire les gains, saturer les couples demandés ou éviter de demander des commandes de type échelon afin d'éviter de demander des gains exorbitants aux moteurs.

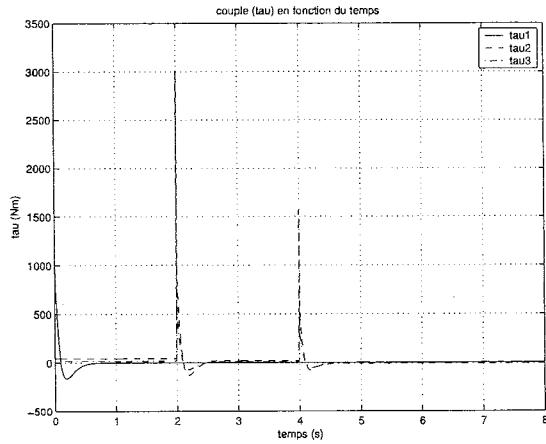


FIG. 23 – Couple, échelon sans perturbation

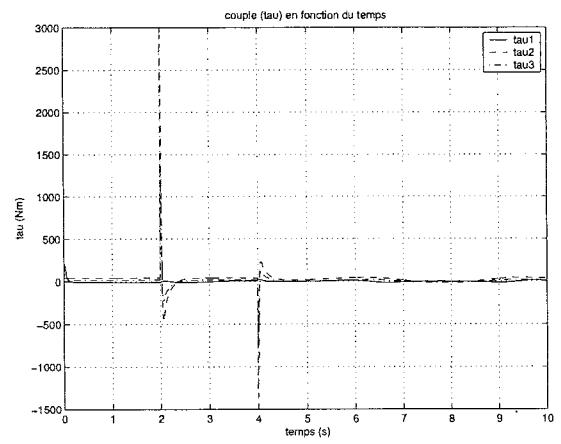


FIG. 24 – Couple, sinusoïdale sans perturbation

Observons maintenant la réponse à une perturbation. Les figures 25 et 26 montrent la réponse du système à la commande appliquée. On remarque que le contrôleur à couple pré-calculé apporte une amélioration substantielle quant à la performance face aux perturbations. En effet, les effets perturbateurs sur les autres articulations sont réduits, ainsi que le dépassement et le temps de réponse. Vous trouverez en annexe des figures supplémentaires montrant les courbes des couples et de l'erreur pour ce contrôleur avec perturbation.

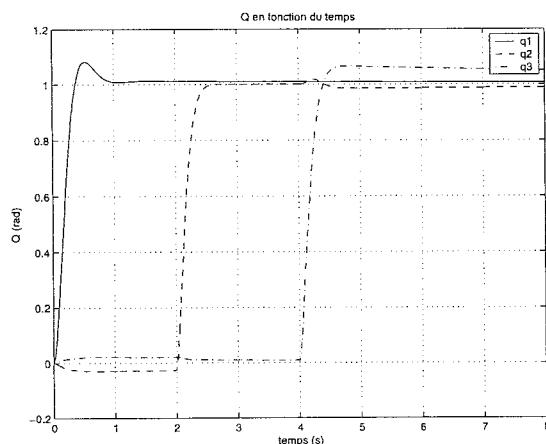


FIG. 25 – Réponse à un échelon avec perturbation

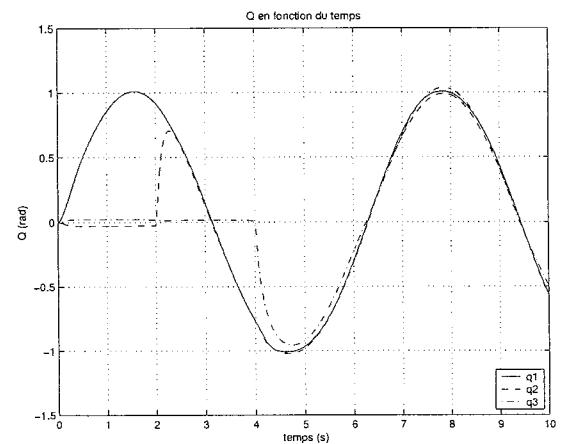


FIG. 26 – Réponse à une sinusoïdale avec perturbation

4.2 Compensateur PID industriel avec compensation de gravité

4.2.1 Le modèle du compensateur

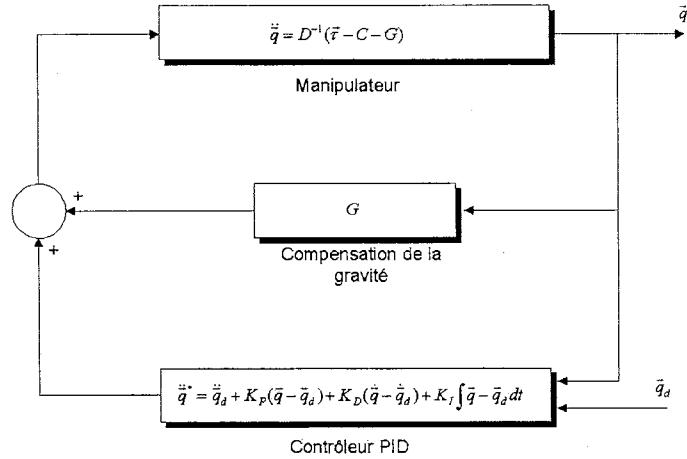


FIG. 27 – Compensateur PID industriel avec compensation de la gravité

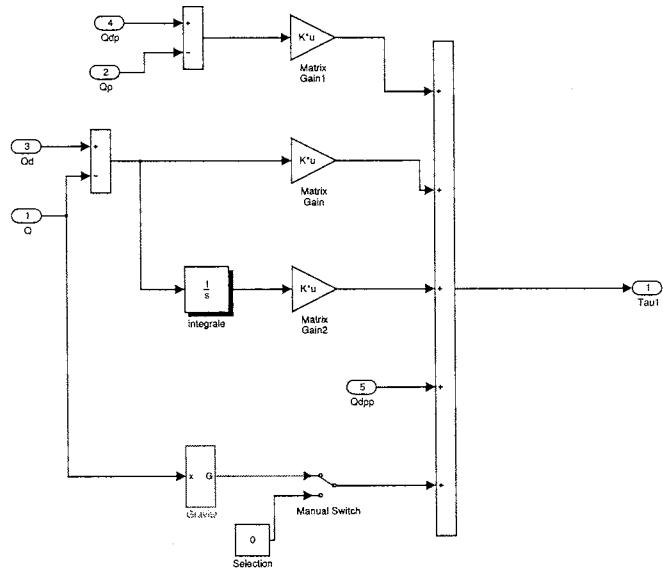


FIG. 28 – Modèle Simulink du compensateur PID industriel avec compensation de la gravité

4.2.2 Choix des gains du compensateur

Les gains du compensateur ont été déterminés par essais et erreurs. L'objectif était toujours d'obtenir une réponse rapide, un dépassement faible et une bonne résistance aux perturbations.

Ainsi, nous avons obtenu les gains suivants :

$$K_P = 120 \quad K_D = 20 \quad K_I = 10$$

4.2.3 Résultats

Nous avons effectué les mêmes quatre tests que pour le compensateur à couple pré-calculé (deux commandes, avec et sans perturbation). Ce sont d'ailleurs les mêmes tests qui seront répétés pour tous les contrôleurs.

Observons les performances du contrôleur PID avec compensation de gravité. Cette dernière compensation permet d'enrayer les effets perturbateurs de la force de gravité présente dans le système ce qui améliore grandement les performances du contrôleur.

Les figures 29 et 30 montrent la position Q en fonction du temps. On peut constater que le temps de réponse est d'environ 1 sec, mais qu'il varie beaucoup en fonction des autres articulations en mouvement. On remarque notamment que le mouvement des bras n'est pas découpé lors de l'utilisation de ce contrôleur. En effet, on voit que les mouvements des bras 2 et 3 s'influencent mutuellement. Ceci est dû à la structure du contrôleur. En effet, la commande ne tient compte que de l'écart entre l'angle de chacune des articulations et l'angle désiré et agit de façon indépendante sur chacun des joints. Les figures 31 et 32 illustrent le comportement de l'erreur en fonction du temps.

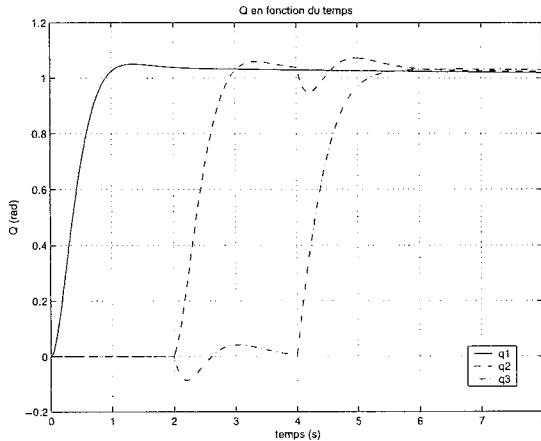


FIG. 29 – Réponse à un échelon sans perturbation

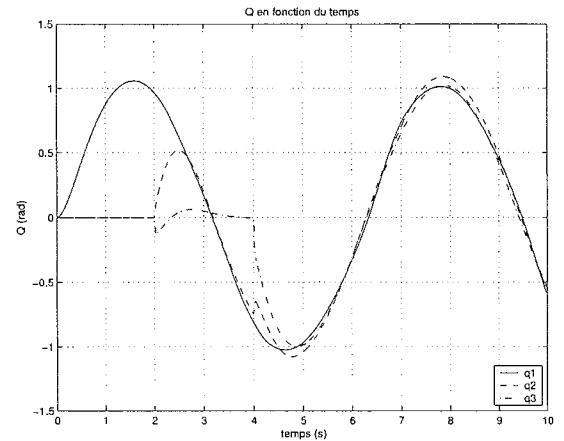


FIG. 30 – Réponse à une sinusoïdale sans perturbation

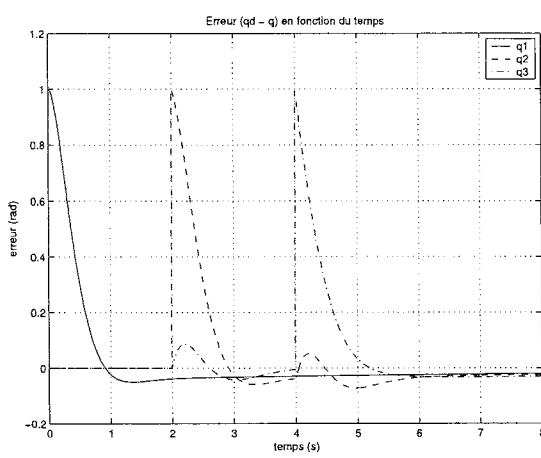


FIG. 31 – Erreur à un échelon sans perturbation

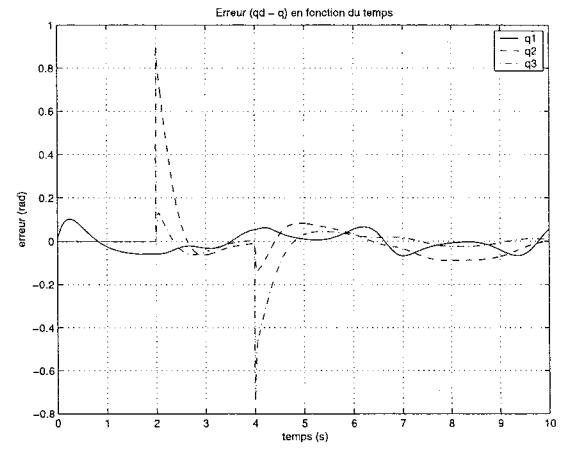


FIG. 32 – Erreur à une sinusoïdale sans perturbation

Les figures 33 et 34 montrent le couple (τ) calculé par le PID en fonction du temps pour chacune des commandes. On remarque des impulsions dans le couple demandé au démarrage de chacune des articulations. Cela provient de l'erreur qui passe instantanément de 0 à 1 rad, ce qui entraîne une réaction violente du contrôleur. Ce comportement sera observé pour chacun des contrôleurs et il est dû à la commande que nous avons choisi d'appliquer.

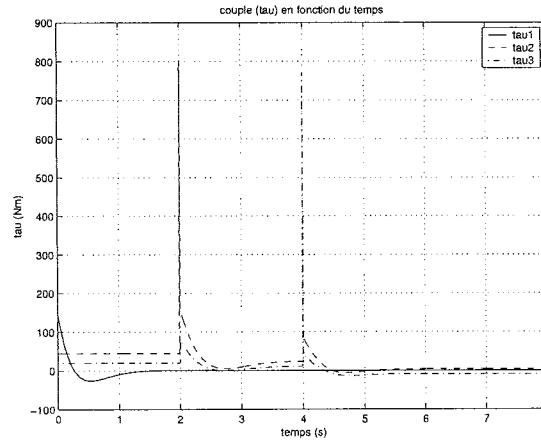


FIG. 33 – Couple, échelon sans perturbation

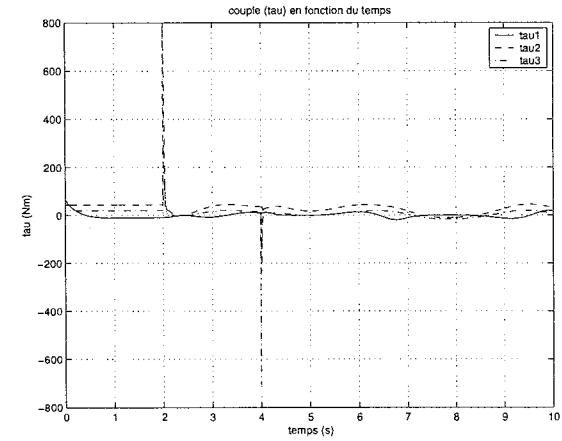


FIG. 34 – Couple, sinusoïdale sans perturbation

Observons maintenant la réponse à une perturbation. Les figures 35 et 36 montrent la réponse du système à la commande appliquée. On remarque que le PID ne permet pas de contrôler efficacement le système. Le dépassement est de beaucoup augmenté, ainsi que la propagation des effets négatifs causés par l'accélération des autres articulations. Vous trouverez en annexe les figures supplémentaires des couples et de l'erreur pour ce contrôleur avec perturbation.

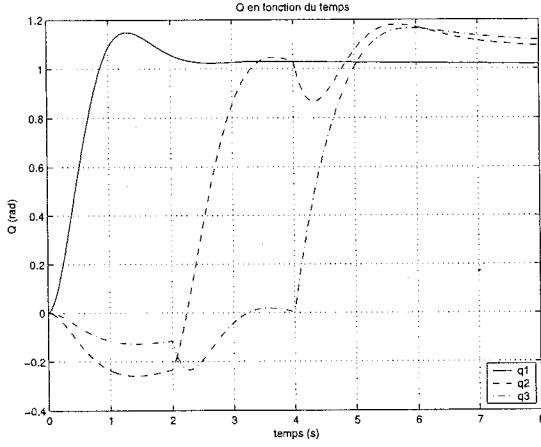


FIG. 35 – Réponse à un échelon avec perturbation

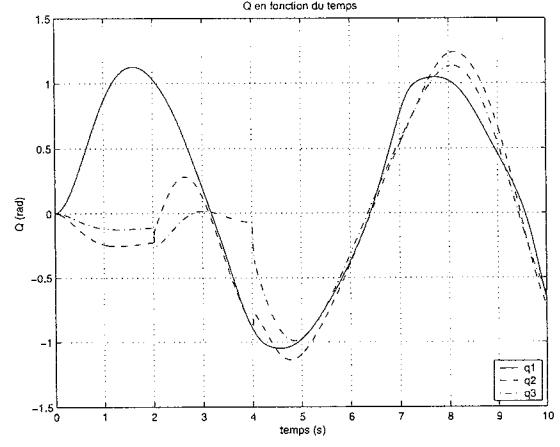


FIG. 36 – Réponse à une sinusoïdale avec perturbation

4.3 Commande adaptative de Slotine-Li

4.3.1 Le modèle du compensateur

Le compensateur à commande adaptative de Slotine-Li est tiré des références [7], [6] et [4]. La loi de commande est présentée à la figure 37. Comme vous le voyez, le contrôleur cinétique détermine la vitesse désirée \dot{q}^* . Cette vitesse ainsi que sa dérivée sont utilisées par le contrôleur dynamique. Le contrôleur dynamique calcule le couple demandé à partir de l'erreur entre la vitesse actuelle et la vitesse désirée ainsi qu'à partir de l'accélération désirée, la vitesse désirée et la dynamique du système inscrite dans la matrice $Y(\vec{q}, \dot{\vec{q}}, \ddot{\vec{q}}^*, \ddot{\vec{q}}^*)\vec{\alpha}$. L'estimateur de paramètre recalcule le vecteur $\vec{\alpha}$ en fonction de l'erreur entre la vitesse obtenue et la vitesse demandée. Le plus difficile est de représenter le système sous la forme $Y(\vec{q}, \dot{\vec{q}}, \ddot{\vec{q}}^*, \ddot{\vec{q}}^*)\vec{\alpha}$.

Nous sommes habitués de représenter le système sous la forme suivante :

$$\vec{\tau} = D(\vec{q})\ddot{\vec{q}}^* + C(\vec{q}, \dot{\vec{q}}) + G(\vec{q})$$

Mais, pour pouvoir évaluer les paramètres, il faut séparer les fonctions (la matrice Y) des paramètres (le vecteur $\vec{\alpha}$) :

$$\vec{\tau} = Y(\vec{q}, \dot{\vec{q}}, \ddot{\vec{q}}^*, \ddot{\vec{q}}^*)\vec{\alpha}$$

Pour déterminer la matrice Y et le vecteur $\vec{\alpha}$, il faut d'abord trouver les matrices D , C et G de la dynamique du système. Ensuite, il faut développer les expressions du vecteur $\vec{\tau}$. À partir de ces expressions, il faut isoler les termes des paramètres qui vont s'insérer dans le vecteur $\vec{\alpha}$. Ensuite, on enlève les termes des paramètres des expressions et on les réorganise de façon à les placer dans la même disposition que dans le vecteur $\vec{\alpha}$.

Comme vous pouvez le constater, ce développement est assez laborieux. Voici tout de même le vecteur $\vec{\alpha}$ et la matrice Y du modèle :

$$\vec{\alpha} = \left[\begin{array}{cccccccc} \frac{1}{2} m_1 r_1^2 & m_2 l_2^2 & m_3 l_3^2 & l_2^2 m_3 & m_3 l_3 l_2 & l_3 m_3 & m_3 l_2 & m_2 l_2 \end{array} \right]$$

$$Y(\vec{q}, \dot{\vec{q}}, \ddot{\vec{q}}) = \begin{bmatrix} Y1 \\ Y2 \\ Y3 \end{bmatrix}$$

$$Y1 = \begin{bmatrix} \ddot{\theta}_1 & , & \left(\frac{1}{6} \cos(2\theta_2) + \frac{1}{6} \right) \ddot{\theta}_1 - \frac{1}{8} \sin(2\theta_2) \dot{\theta}_2^* \dot{\theta}_1 - \frac{5}{24} \sin(2\theta_2) \dot{\theta}_1^* \dot{\theta}_2 & , \\ \left(\frac{1}{6} \cos(2\theta_2 + 2\theta_3) + \frac{1}{6} \right) \ddot{\theta}_1 + \left(-\frac{1}{8} \sin(2\theta_2 + 2\theta_3) \dot{\theta}_2^* \right. \\ \left. - \frac{1}{8} \sin(2\theta_2 + 2\theta_3) \dot{\theta}_3^* \right) \dot{\theta}_1 - \frac{5}{24} \sin(2\theta_2 + 2\theta_3) \dot{\theta}_1^* \dot{\theta}_2 - \frac{5}{24} \sin(2\theta_2 + 2\theta_3) \dot{\theta}_3 \dot{\theta}_1^* & , \\ \left(\frac{1}{2} \cos(2\theta_2) + \frac{1}{2} \right) \ddot{\theta}_1 - \frac{1}{2} \sin(2\theta_2) \dot{\theta}_2^* \dot{\theta}_1 - \frac{1}{2} \sin(2\theta_2) \dot{\theta}_1^* \dot{\theta}_2 & , \\ \left(\frac{1}{2} \cos(\theta_3) + \frac{1}{2} \cos(2\theta_2 + \theta_3) \right) \ddot{\theta}_1 + \left(-\frac{1}{2} \sin(2\theta_2 + \theta_3) \dot{\theta}_2^* \right. \\ \left. + \left(-\frac{1}{4} \sin(2\theta_2 + \theta_3) - \frac{1}{4} \sin(\theta_3) \right) \dot{\theta}_3^* \right) \dot{\theta}_1 - \frac{1}{2} \sin(2\theta_2 + \theta_3) \dot{\theta}_1^* \dot{\theta}_2 \\ \left. + \left(-\frac{1}{4} \sin(2\theta_2 + \theta_3) - \frac{1}{4} \sin(\theta_3) \right) \dot{\theta}_3 \dot{\theta}_1^* \right) & , & 0 & , & 0 & , & 0 \end{bmatrix}$$

$$Y2 = \begin{bmatrix} 0 & , & \frac{1}{3} \ddot{\theta}_2 + \frac{1}{6} \sin(2\theta_2) \dot{\theta}_1 \dot{\theta}_1^* & , & \frac{1}{6} \sin(2\theta_2 + 2\theta_3) \dot{\theta}_1 \dot{\theta}_1^* + \frac{1}{3} \ddot{\theta}_3 + \frac{1}{3} \ddot{\theta}_2 & , & \ddot{\theta}_2 + \frac{1}{2} \sin(2\theta_2) \dot{\theta}_1 \dot{\theta}_1^* & , \\ \cos(\theta_3) \ddot{\theta}_2 + \frac{1}{2} \cos(\theta_3) \ddot{\theta}_3 - \frac{1}{2} \sin(\theta_3) \dot{\theta}_3^* \dot{\theta}_2 + \left(-\frac{1}{2} \sin(\theta_3) \dot{\theta}_2^* - \frac{1}{2} \sin(\theta_3) \dot{\theta}_3^* \right) \dot{\theta}_3 + \frac{1}{2} \sin(2\theta_2 + \theta_3) \dot{\theta}_1 \dot{\theta}_1^* & , \\ \frac{1}{2} g \cos(\theta_2 + \theta_3) & , & g \cos(\theta_2) & , & \frac{1}{2} g \cos(\theta_2) \end{bmatrix}$$

$$Y3 = \begin{bmatrix} 0 & , & 0 & , & \frac{1}{6} \sin(2\theta_2 + 2\theta_3) \dot{\theta}_1 \dot{\theta}_1^* + \frac{1}{3} \ddot{\theta}_3 + \frac{1}{3} \ddot{\theta}_2 & , & 0 & , \\ \frac{1}{2} \cos(\theta_3) \ddot{\theta}_2 + \frac{1}{2} \sin(\theta_3) \dot{\theta}_2 \dot{\theta}_2^* + \left(\frac{1}{4} \sin(2\theta_2 + \theta_3) + \frac{1}{4} \sin(\theta_3) \right) \dot{\theta}_1 \dot{\theta}_1^* & , \\ \frac{1}{2} g \cos(\theta_2 + \theta_3) & , & 0 & , & 0 \end{bmatrix}$$

Cette matrice a été utilisée dans le modèle *Simulink* de ce compensateur et du compensateur à commande adaptative modifiée qui est expliqué à la section suivante.

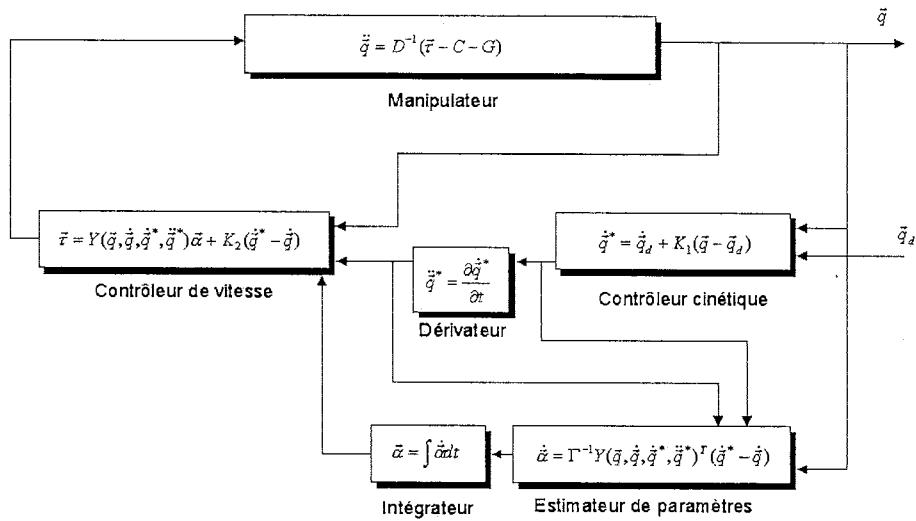


FIG. 37 – Commande adaptative de Slotine-Li

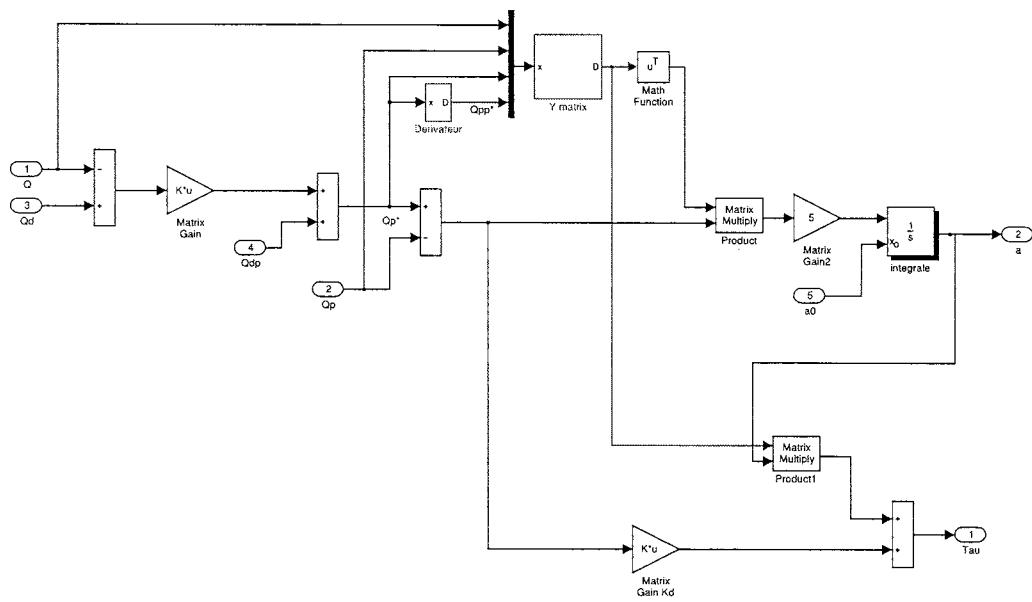


FIG. 38 – Modèle Simulink de la commande adaptative de Slotine-Li

4.3.2 Choix des gains du compensateur

Les gains utilisés pour ce compensateur ont été déterminés par essais et erreurs. Nous avons remarqué qu'il est important de respecter la règle suivante :

$$K_2 > K_1 > \Gamma^{-1}$$

où Γ^{-1} est simplement une matrice de gain diagonale définie positive.

Les gains choisis sont les suivant :

$$K_2 = 100 \quad K_1 = 10 \quad \Gamma^{-1} = 5$$

4.3.3 Résultats

Nous avons effectué les mêmes quatre tests (deux commandes, avec et sans perturbation) que pour les autres compensateurs.

Il est à noter que nous avons choisi comme paramètres initiaux des paramètres qui sont proches des paramètres réels soit un écart entre 0 et 20%. Le tableau 4 et les figures 39 et 40 permettent de voir les valeurs choisies, les valeurs atteintes et leur évolution. Les valeurs choisies initialement sont assez proches des vraies valeurs ce qui est logique, car on peut penser que bien que la valeur exacte ne soit pas connue, nous avons une bonne idée de l'ordre de grandeur de ces paramètres.

| <i>Paramètres</i> | <i>Valeurs</i> | <i>Paramètres initiaux</i> | <i>Paramètres finaux</i> |
|-------------------|----------------|----------------------------|--------------------------|
| α_1 | 0.005 | 0.005 | 5.5 |
| α_2 | 1 | 1 | 8.1 |
| α_3 | 8 | 9 | 34.5 |
| α_4 | 2 | 1.2 | 23.0 |
| α_5 | 4 | 5 | 19.5 |
| α_6 | 4 | 3 | 2.2 |
| α_7 | 2 | 2 | 0 |
| α_8 | 1 | 0.5 | -0.5 |

TAB. 4 – Paramètres pour un échelon, Slotine-Li

On remarque d'après les figures que le contrôleur a tendance à faire stabiliser les valeurs des paramètres vers des valeurs tout à fait différentes des valeurs attendues. En fait, les paramètres ne convergent pas nécessairement vers les valeurs attendues, mais vers d'autres valeurs équivalentes qui fonctionnent bien. Effectivement, les graphiques de l'erreur qui seront vus plus loin permettent d'accepter ces valeurs puisque l'erreur en régime permanent est de l'ordre de 10^{-3} .

On remarque également des discontinuités dans les valeurs des paramètres aux temps $t = 2$ sec

et $t = 4$. Elles sont dûes à la mise en mouvement des articulations 2 et 3 respectivement et bien entendu aux discontinuités présentent dans la commande. En effet, ce sont les mouvements des différents bras qui influencent le comportement du compensateur, il doit calculer les nouvelles valeurs des paramètres en fonction de l'écart entre la vitesse obtenue et la vitesse désirée. Le compensateur doit agir rapidement pour contrer les changements brusques provoqués par les discontinuités, mais les paramètres se stabilisent à nouveau dans l'espace de quelques secondes par la suite.

Évidemment nous pourrions jouer avec les gains pour influencer le couple et l'erreur que nous désirons.

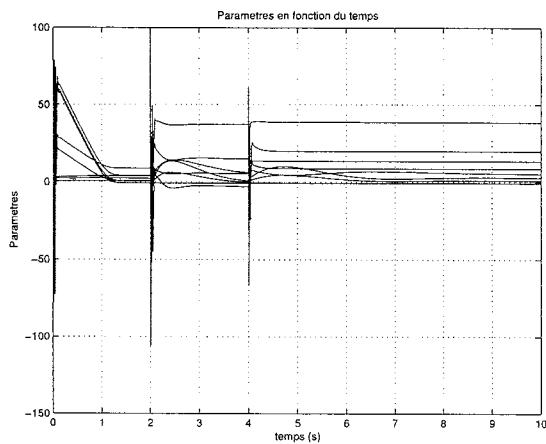


FIG. 39 – Paramètres pour un échelon, (sans perturbation)

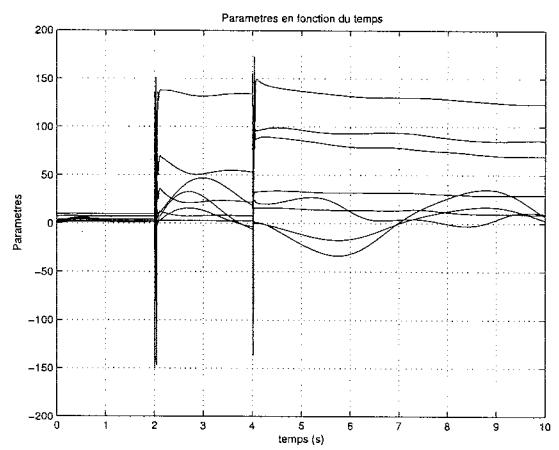


FIG. 40 – Paramètres pour une sinusoïdale, (sans perturbation)

Les figures 41 et 42 montrent la position Q en fonction du temps. Dans ce cas-ci, les gains choisis sont de $\delta = 10$, $K_d = 100$ et $\lambda = 5$. On remarque que la performance de ce contrôleur est inférieure à celle du compensateur à couple pré-calculé en ce qui concerne l'erreur. Ceci est normal, car le couple pré-calculé connaît avec exactitude le système et applique une loi de commande parfaitement découpée qui est en fonction de ces paramètres connus. On remarque cependant que le dépassement est inférieur au dépassement observé dans le cas du PID et que la propagation des effets du mouvement d'un membre sur les autres membres est inférieure au cas du PID. L'erreur en régime permanent est minime. Les figures 43 et 44 illustrent le comportement de l'erreur en fonction du temps.

Les figures 45 et 46 illustrent le couple calculé par le contrôleur à commande adaptative en fonction du temps pour chacune des commandes. On remarque que le couple calculé est de l'ordre de 10^4 . Évidemment nous sommes en simulation, mais nous ne pourrions pas planter un tel contrôleur sur un vrai système. Afin de simuler de façon plus exacte le système, dans le cas où on chercherait à l'implanter, il faudrait ajouter des saturateurs pour tenir compte des limites physiques réelles des moteurs.

