

29 JUL. 2002

EPM / RT - 02 - 02

**MODÉLISATION D'UN BRAS MANIPULATEUR
À SIX DEGRÉS DE LIBERTÉ :
OUTILS DE VISUALISATION ET EXEMPLE D'APPLICATION**

gratuit

Manuel D. THÉBERGE
Département de Génie Électrique

Projet dirigé par
Professeur Romano M. DeSantis

Janvier 2002

@2002
Romano M. De Santis
Tous droits réservés

Dépôt légal :
Bibliothèque nationale du Québec, 2002
Bibliothèque nationale du Canada, 2002

EPM / RT - 02 - 02

*Modélisation d'un bras manipulateur à six degrés de liberté : Outils de visulation
et exemple d'application*

par : Romano M. De Santis

Département de génie électrique.

École Polytechnique de Montréal.

Toute reproduction de ce document à des fins d'études personnelle ou de recherche est autorisée à 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.peb@courriel.polymtl.ca

EPM / RT - 02 - 02

**MODÉLISATION D'UN BRAS MANIPULATEUR
À SIX DEGRÉS DE LIBERTÉ :
OUTILS DE VISUALISATION ET EXEMPLE D'APPLICATION**

Manuel D. THÉBERGE
Département de Génie Électrique

Projet dirigé par
Professeur Romano M. DeSantis

Janvier 2002

MODÉLISATION D'UN BRAS MANIPULATEUR
À SIX DEGRÉS DE LIBERTÉ :
OUTILS DE VISUALISATION ET EXEMPLE D'APPLICATION

Manuel D. THÉBERGE
Département de Génie Électrique

projet dirigé par
Professeur Romano M. DeSantis

Janvier 2002

MODÉLISATION D'UN BRAS MANIPULATEUR À SIX DEGRÉS DE LIBERTÉ: OUTILS DE VISUALISATION ET EXEMPLE D'APPLICATION

Manuel D. THÉBERGE
Département de Génie Électrique

Sommaire

Ce rapport concerne le développement d'un modèle mathématique et d'un module Matlab-Simulink permettant d'étudier et de simuler le comportement dynamique d'un manipulateur robotique avec 6 degrés de liberté.

Notre objectif est triple: a) augmenter l'ensemble d'exemples non triviaux de systèmes mécaniques illustrant la procédure de modélisation de Kane telle que récemment formalisée par De Santis; b) mettre à la disposition de la communauté scientifique un outil de recherche que nous considérons intéressant dans le cadre d'études sur la commande de systèmes robotiques; c) illustrer le rôle que Matlab et Simulink peuvent jouer dans l'atteinte des objectifs a) et b).

On illustre les résultats obtenus en les appliquant au développement d'un asservissement en force et en position du manipulateur en étude.

Table Des Matières

1	INTRODUCTION	3
2	MODÉLISATION	4
2.1	ÉTAPE 1.....	4
2.2	ÉTAPE 2.....	5
2.3	ÉTAPE 3.....	6
2.4	ÉTAPE 4.....	7
2.5	ÉTAPE 5.....	11
2.6	ÉTAPE 6.....	11
2.7	ÉTAPE 7.....	11
2.8	ÉTAPE 8-9-10.....	13
3	OUTILS DE VISUALISATION	18
3.1	MODULE 1.....	19
3.2	MODULE 2.....	20
3.3	MODULE 3.....	21
4	ASSERVISSEMENT EN POSITION / FORCE	22
4.1.	MODÉLISATION DU CAPTEUR DE FORCE	23
4.2	SCHÉMA DE CONTRÔLE	23
4.3	SIMULATION.....	24
5	CONCLUSIONS	25
6	BIBLIOGRAPHIE	25

Liste des figures

Figure 1–	Coordonnées généralisées du manipulateur	5
Figure 2 –	Points d’application de la force de gravité	15
Figure 3 –	Manipulateur, vue de profil	16
Figure 4 –	Intégration dans Simulink.....	18
Figure 5 –	Visualisation du manipulateur sous Matlab.....	18
Figure 6 –	Acquisition des données de la simulation.....	19
Figure 7 –	Capteur de force.....	22
Figure 8 –	Schéma du contrôleur hybride	23
Figure 9 –	Trajectoire suivie par le bras lors du polissage.....	24

1 Introduction

Nous développerons le modèle dynamique du manipulateur à six degrés de liberté illustré à la figure 1. Pour ce faire, nous utiliserons la procédure formalisée par De Santis [1]. Il s'agit d'une procédure systématique qui est basée sur l'approche de Kane et Levinson [2] ainsi que sur le traitement de Saha et Angeles [3]. Elle est applicable à un système mécanique constitué de corps rigides soumis à des contraintes holonomes et non holonomes.

Tel qu'illustré à la figure 1, le manipulateur en étude est constitué de trois segments. Chacun de ces trois segments est à son tour constitué d'une tige de longueur L_i , $i = 1, 2$ et 3 , d'une masse ponctuelle, m_i , concentrée dans le centre de masse (indiqué à la figure 1), et d'une matrice d'inertie relative au centre de masse, I_i . Le premier segment se déplace relativement au référentiel de base moyennant un mouvement de rotation en azimuth et en élévation. Le deuxième segment se déplace par rapport au premier segment moyennant une rotation en élévation. Le troisième segment se déplace relativement au deuxième moyennant une rotation en azimuth, élévation et lacet.

Les explications et développements qui suivent prennent pour acquis que le lecteur est familier avec la procédure de modélisation décrite à la référence [1] ainsi qu'avec les notions de cinématique et dynamique des corps rigides que l'on peut retrouver dans une référence de nature fondamentale telle que [4].

Divisé en trois sections, le rapport couvre; modélisation, outils de visualisation et asservissement en position / force.

2 Modélisation

2.1 Étape 1

On attribue un repère à chacun des segments du manipulateur. Sachant que la position et l'orientation d'un corps peut être représenté par six variables indépendantes (trois pour la position et trois autres pour l'orientation), on peut alors décrire ces quantités pour chacun des segments du manipulateur dans l'espace opérationnel. Nous avons donc :

$$\chi_1 = [x_1 \quad y_1 \quad z_1 \quad \gamma_1 \quad \beta_1 \quad \alpha_1]^T$$

$$\chi_2 = [x_2 \quad y_2 \quad z_2 \quad \gamma_2 \quad \beta_2 \quad \alpha_2]^T$$

$$\chi_3 = [x_3 \quad y_3 \quad z_3 \quad \gamma_3 \quad \beta_3 \quad \alpha_3]^T$$

En observant la géométrie et la configuration des joints à la figure 1, on remarque qu'il suffit de connaître la valeur des angles de chaque articulation pour ainsi déterminer la position et l'orientation des segments du manipulateur. Ces 6 variables indépendantes viennent donc fixer le nombre de degrés de liberté.

2.2 Étape 2

On peut alors exprimer la position et l'orientation du manipulateur par le vecteur de configuration $q = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ \theta_5 \ \theta_6]^T \in \mathbb{R}^6$. Les divers termes du vecteur de configuration (coordonnées généralisées) sont aussi illustrés dans la figure 1.

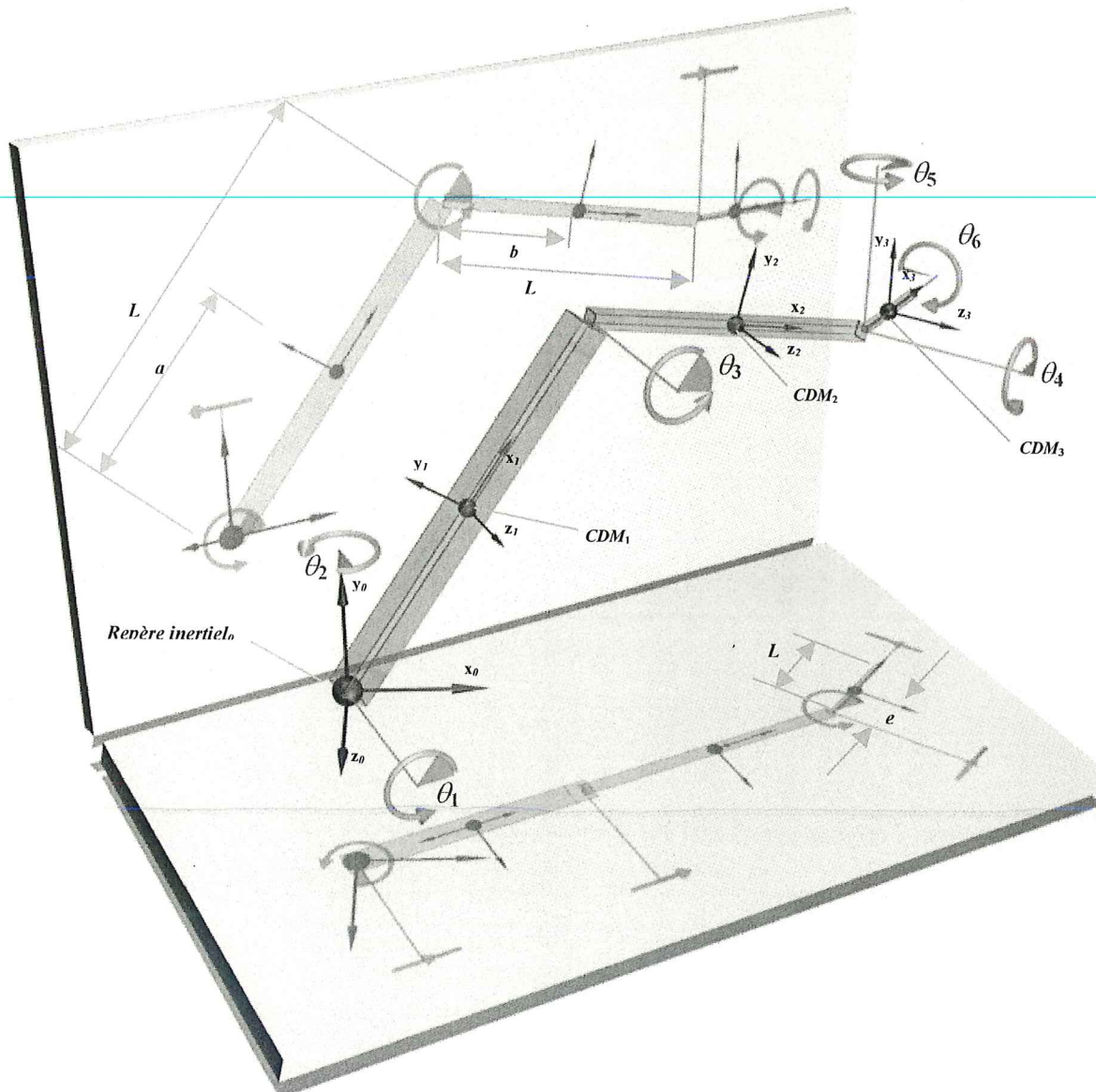


Figure 1– Coordonnées généralisées du manipulateur

2.3 Étape 3

Dans cette étape la fonction de cinématique directe (ou encore la relation entre le vecteur q et la position et l'orientation dans l'espace opérationnel) est déterminée.

Parce que les relations analytiques deviennent rapidement imposantes, surtout pour le 2^e et 3^e segment, il est utile de faire appel aux matrices de rotation pour ainsi rendre la notation compacte et ultérieurement facile à intégrer dans Simulink. Dans cette perspective, posons les matrices de rotation suivantes; Rot(2,3), soit la matrice permettant de passer du repère du 3^e segment au 2^e. D'une manière similaire définissons Rot(1,2) puis Rot(0,1). En précisant ici que 0 représente le repère inertiel. Ces matrices de rotation sont donc :

$$\mathbf{Rot}(2,3) = \mathbf{Rot}(z, \theta_4) * \mathbf{Rot}(y, \theta_5) * \mathbf{Rot}(x, \theta_6)$$

$$\mathbf{Rot}(1,2) = \mathbf{Rot}(z, \theta_3)$$

$$\mathbf{Rot}(0,1) = \mathbf{Rot}(y, \theta_2) * \mathbf{Rot}(z, \theta_1)$$

En utilisant les informations sur la géométrie des segments, figure 1, on peut alors exprimer la fonction de cinématique directe par:

1^{er} segment (CDM), en posant $[\mathbf{r}_{ij}] = \mathbf{Rot}(0,1)$

$$\begin{aligned} [x_1 \ y_1 \ z_1]^T &= \mathbf{Rot}(0,1) * [a \ 0 \ 0]^T \\ \beta_1 &= a \tan 2(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}) \in (-90^\circ, 90^\circ) \\ \alpha_1 &= a \tan 2(r_{21} / \cos \beta_1, r_{11} / \cos \beta_1) \\ \gamma_1 &= a \tan 2(r_{32} / \cos \beta_1, r_{33} / \cos \beta_1) \end{aligned}$$

2^e segment (CDM), en posant $[\mathbf{r}_{ij}] = \mathbf{Rot}(0,1) * \mathbf{Rot}(1,2)$

$$\begin{aligned} [x_2 \ y_2 \ z_2]^T &= \mathbf{Rot}(0,1) * [\mathbf{L}_1 \ 0 \ 0]^T + \mathbf{Rot}(0,1) * \mathbf{Rot}(1,2) * [b \ 0 \ 0]^T \\ \beta_2 &= a \tan 2(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}) \in (-90^\circ, 90^\circ) \\ \alpha_2 &= a \tan 2(r_{21} / \cos \beta_2, r_{11} / \cos \beta_2) \\ \gamma_2 &= a \tan 2(r_{32} / \cos \beta_2, r_{33} / \cos \beta_2) \end{aligned}$$

3^e segment (CDM), en posant $[\mathbf{r}_{ij}] = \mathbf{Rot}(0,1) * \mathbf{Rot}(1,2) * \mathbf{Rot}(2,3)$

$$\begin{aligned} [x_3 \ y_3 \ z_3]^T &= \mathbf{Rot}(0,1) * [\mathbf{L}_1 \ 0 \ 0]^T + \mathbf{Rot}(0,1) * \mathbf{Rot}(1,2) * [\mathbf{L}_2 \ 0 \ 0]^T + \dots \\ &\quad \mathbf{Rot}(0,1) * \mathbf{Rot}(1,2) * \mathbf{Rot}(2,3) * [e \ 0 \ 0]^T \\ \beta_3 &= a \tan 2(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}) \in (-90^\circ, 90^\circ) \\ \alpha_3 &= a \tan 2(r_{21} / \cos \beta_3, r_{11} / \cos \beta_3) \\ \gamma_3 &= a \tan 2(r_{32} / \cos \beta_3, r_{33} / \cos \beta_3) \end{aligned}$$

2.4 Étape 4

Sachant que les corps composant le système mécanique du manipulateur sont soumis à un ensemble de contraintes en position et orientation (contraintes holonomes), on veut ici relier le vecteur des vitesses v à la dérivée des coordonnées généralisées $\dot{q}$. La matrice du jacobienne des contraintes holonomes (J_h) permet ce passage selon la relation suivante :

$$v = J_h(q)\dot{q}$$

Comme le manipulateur possède trois segments (trois corps rigides), la matrice J_h est de dimension 18×6 . En effet, il y a autant de colonnes dans J_h qu'il y a de degrés de liberté (6) et comme chaque corps rigide est décrit avec 6 vitesses (trois linéaires et trois angulaires), il y aura donc un total de 18 lignes. Chaque colonne représente le sous-espace des vitesses dans l'espace opérationnel, qui sont compatibles avec les contraintes holonomes. Le système étudié comporte trois corps rigides, on peut alors décomposer J_h comme suit :

$$J_h = \begin{bmatrix} J_{h1} \\ J_{h2} \\ J_{h3} \end{bmatrix}$$

Où J_{h1} , J_{h2} et J_{h3} sont les matrices jacobiennes 6×6 de chaque corps rigide (segment).

Construction de J_{h1}

Dans un premier temps, observons les entrées non nulles de J_{h1} .

$$J_{h1} = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 & 0 & 0 \\ a_{31} & a_{32} & 0 & 0 & 0 & 0 \\ a_{41} & a_{42} & 0 & 0 & 0 & 0 \\ a_{51} & a_{52} & 0 & 0 & 0 & 0 \\ a_{61} & a_{62} & 0 & 0 & 0 & 0 \end{bmatrix}$$

Parce que le premier segment est orienté via θ_1 et θ_2 , il est donc logique que seul les colonnes associées à $\dot{\theta}_1$ et $\dot{\theta}_2$ interviennent dans J_{h1} . Comme vu précédemment, il faut exprimer les vitesses linéaires et angulaires du premier segment dans l'espace opérationnel mais il faut les transposer dans le système d'axe du corps rigide concerné, dans ce cas, le repère du premier segment.

Parce que l'intégration de J_h dans Simulink demande à ce que le calcul des éléments de chaque colonne soit indépendant, la procédure utilisée sera la suivante :

Détermination des entrées a_{4i} , a_{5i} et a_{6i} (vitesses angulaires):

- 1) Identifier dans quel repère $\dot{\theta}_i$ est invariant en orientation (assumons le repère A), puis l'exprimer $\dot{\theta}_i$ par un vecteur colonne unitaire. Nous avons donc : $[\dot{\theta}_{ie}]_A$
- 2) Soit B, le repère du i^e segment associé à J_{hi} , alors :

$$\begin{bmatrix} a_{4i} \\ a_{5i} \\ a_{6i} \end{bmatrix}_B = Rot(B, A) [\dot{\theta}_{ie}]_A$$

Détermination des entrées a_{1i} , a_{2i} et a_{3i} (vitesses linéaires):

Il faut ici propager les vitesses linéaires de la base vers le segment B (i^e segment associé à J_{hi}).

- 1) Associé à $\dot{\theta}_i$, on identifie le segment sur lequel la coordonnée généralisée θ_i agit directement. Posons C, le segment et P_{jc} , le joint où θ_i agit directement.
- 2) Exprimer $\dot{\theta}_i$ par rapport à C, on a :

$$[\dot{\theta}_{ie}]_C = Rot(C, A) [\dot{\theta}_{ie}]_A$$

- 3) Si le repère du i^e segment associé à J_{hi} est C, alors on fait le produit vectoriel de $[\dot{\theta}_{ie}]_C$ avec le vecteur position $\overrightarrow{P_{jc}P_{CDM}}_C$ du centre de masse de C. Les entrées a_{1i} , a_{2i} et a_{3i} sont :

$$\begin{bmatrix} a_{1i} \\ a_{2i} \\ a_{3i} \end{bmatrix}_C = [\dot{\theta}_{ie}]_C \times \overrightarrow{P_{jc}P_{CDM}}_C$$

- 4) Si le repère du i^e segment associé à J_{hi} n'est pas C, alors on fait le produit vectoriel de $[\dot{\theta}_{ie}]_C$ avec le vecteur position $\overrightarrow{P_{jc}P_{jc+1}}_C$, puis on exprime le résultat par rapport au repère B (soit le repère du i^e segment associé à J_{hi}). C'est ici un cas où l'on propage les vitesses linéaires de la base vers B. Les entrées a_{1i} , a_{2i} et a_{3i} deviennent alors :

$$\begin{bmatrix} a_{1i} \\ a_{2i} \\ a_{3i} \end{bmatrix}_B = Rot(B, C) [\dot{\theta}_{ie}]_C \times \overrightarrow{P_{jc}P_{jc+1}}_C$$

En utilisant la procédure décrite précédemment, nous avons donc les entrées suivantes pour la matrice jacobienne J_{hi} :

$$\begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \end{bmatrix}_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_1 \times \begin{bmatrix} a \\ 0 \\ 0 \end{bmatrix}_1 \quad \begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \end{bmatrix}_1 = Rot(0,1)^T \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}_0 \times \begin{bmatrix} a \\ 0 \\ 0 \end{bmatrix}_1$$

$$\begin{bmatrix} a_{41} \\ a_{51} \\ a_{61} \end{bmatrix}_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_1 \quad \begin{bmatrix} a_{42} \\ a_{52} \\ a_{62} \end{bmatrix}_1 = Rot(0,1)^T \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}_0$$

Construction de J_{h2}

Les entrées non nulles de J_{h2} sont :

$$J_{h2} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & 0 & 0 & 0 \\ a_{51} & a_{52} & a_{53} & 0 & 0 & 0 \\ a_{61} & a_{62} & a_{63} & 0 & 0 & 0 \end{bmatrix}$$

Ici les deux premières colonnes représentent la propagation des vitesses associées à $\dot{\theta}_1$ et $\dot{\theta}_2$, puis la troisième fait directement intervenir $\dot{\theta}_3$. Nous avons donc :

$$\begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \end{bmatrix}_2 = Rot(1,2)^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_1 \times \begin{bmatrix} L_1 \\ 0 \\ 0 \end{bmatrix}_1 \quad \begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \end{bmatrix}_2 = Rot(1,2)^T Rot(0,1)^T \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}_0 \times \begin{bmatrix} L_1 \\ 0 \\ 0 \end{bmatrix}_1$$

$$\begin{bmatrix} a_{41} \\ a_{51} \\ a_{61} \end{bmatrix}_2 = Rot(1,2)^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_1 \quad \begin{bmatrix} a_{42} \\ a_{52} \\ a_{62} \end{bmatrix}_2 = Rot(1,2)^T Rot(0,1)^T \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}_0$$

$$\begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \end{bmatrix}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_2 \times \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix}_2 \quad \begin{bmatrix} a_{43} \\ a_{53} \\ a_{63} \end{bmatrix}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_2$$

Construction de J_{h3}

Finalement, les entrées non nulles de J_{h3} sont :

$$J_{h3} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} \end{bmatrix}$$

Ici les trois premières colonnes représentent la propagation des vitesses associées à $\dot{\theta}_1$, $\dot{\theta}_2$ et $\dot{\theta}_3$, puis les trois dernières décrivent l'effet de $\dot{\theta}_4$, $\dot{\theta}_5$ et $\dot{\theta}_6$. Nous avons donc :

$$\begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \end{bmatrix}_3 = Rot(2,3)^T Rot(1,2)^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_1 \times \begin{bmatrix} L_1 \\ 0 \\ 0 \end{bmatrix}_1$$

$$\begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \end{bmatrix}_3 = Rot(2,3)^T Rot(1,2)^T \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}_0 \times \begin{bmatrix} L_1 \\ 0 \\ 0 \end{bmatrix}_1$$

$$\begin{bmatrix} a_{41} \\ a_{51} \\ a_{61} \end{bmatrix}_3 = Rot(2,3)^T Rot(1,2)^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_1$$

$$\begin{bmatrix} a_{42} \\ a_{52} \\ a_{62} \end{bmatrix}_3 = Rot(2,3)^T Rot(1,2)^T \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}_0$$

$$\begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \end{bmatrix}_3 = Rot(2,3)^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_2 \times \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix}_2$$

$$\begin{bmatrix} a_{14} \\ a_{24} \\ a_{34} \end{bmatrix}_3 = \left[Rot(z, \theta_4) Rot(y, \theta_5) Rot(x, \theta_6) \right]^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_2 \times \begin{bmatrix} e \\ 0 \\ 0 \end{bmatrix}_3$$

$$\begin{bmatrix} a_{43} \\ a_{53} \\ a_{63} \end{bmatrix}_3 = Rot(2,3)^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_2$$

$$\begin{bmatrix} a_{44} \\ a_{54} \\ a_{64} \end{bmatrix}_3 = [Rot(z, \theta_4) Rot(y, \theta_5) Rot(x, \theta_6)]^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_2$$

$$\begin{bmatrix} a_{15} \\ a_{25} \\ a_{35} \end{bmatrix}_3 = \left[Rot(y, \theta_5) Rot(x, \theta_6) \right]^T \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}_2 \times \begin{bmatrix} e \\ 0 \\ 0 \end{bmatrix}_3 \quad \begin{bmatrix} a_{16} \\ a_{26} \\ a_{36} \end{bmatrix}_3 = \left[Rot(x, \theta_6) \right]^T \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}_2 \times \begin{bmatrix} e \\ 0 \\ 0 \end{bmatrix}_3$$

$$\begin{bmatrix} a_{45} \\ a_{55} \\ a_{65} \end{bmatrix}_3 = [Rot(y, \theta_5) Rot(x, \theta_6)]^T \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}_2 \quad \begin{bmatrix} a_{46} \\ a_{56} \\ a_{66} \end{bmatrix}_3 = Rot(x, \theta_6)^T \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}_2$$

2.5 Étape 5

Il faut sélectionner un vecteur des vitesses généralisées $\alpha \in \mathfrak{R}^6$. Sachant que $\dot{q} = J_{nh}(q)\alpha$, où J_{nh} représente la matrice jacobienne associée aux contraintes nonholonomes, et que ce manipulateur n'est restreint par aucune de ces contraintes, il s'ensuit donc que $J_{nh} = I$. Le vecteur des vitesses généralisées devient simplement $\alpha = \dot{q}$.

2.6 Étape 6

On doit ici calculer $J = J_h J_{nh}$ et $\dot{J} = \frac{dJ}{dt}$. En utilisant les simplifications données précédemment, on a : $J = J_h$. Comme cette modélisation sera implantée sous Simulink, la dérivée numérique J sera utilisée pour représenter $\dot{J}$.

2.7 Étape 7

Construction de la matrice M (matrice de masse étendue)

$$M = \begin{bmatrix} M_1 & 0_6 & 0_6 \\ 0_6 & M_2 & 0_6 \\ 0_6 & 0_6 & M_3 \end{bmatrix} \quad \text{où } 0_6 \text{ sont des matrices carrées de zéros } 6 \times 6.$$

$$M_1 = \begin{bmatrix} m_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & m_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{1xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{1yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{1zz} \end{bmatrix}$$

où m_1 est la masse (kg) du premier segment et les termes $I_{1xx}, I_{1yy}, I_{1zz}$ (kg m²) sont les moments d'inertie du premier segment par rapport aux axes principaux. Cependant, dans le cas où l'on aurait placé le repère 1 (repère du premier segment) dans une autre orientation que celle des axes principaux, on aurait eu une matrice 3×3 représentant le tenseur d'inertie suivant:

$$\bar{I} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}$$

On peut donc voir l'avantage à placer les repères des segments de façon à respecter l'orientation des axes principaux. Les matrices M_2 et M_3 sont :

$$M_2 = \begin{bmatrix} m_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & m_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{2xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{2yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{2zz} \end{bmatrix} \quad M_3 = \begin{bmatrix} m_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & m_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{3xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{3yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{3zz} \end{bmatrix}$$

Construction de la matrice W (matrice des vitesses angulaires étendue)

$$W = \begin{bmatrix} W_1 & 0_6 & 0_6 \\ 0_6 & W_2 & 0_6 \\ 0_6 & 0_6 & W_3 \end{bmatrix} \quad \text{où } 0_6 \text{ sont des matrices carrées de zéros } 6 \times 6.$$

$$W_1 = \begin{bmatrix} 0 & -\Omega_{1z} & \Omega_{1y} & 0 & 0 & 0 \\ \Omega_{1z} & 0 & -\Omega_{1x} & 0 & 0 & 0 \\ -\Omega_{1y} & \Omega_{1x} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\Omega_{1z} & \Omega_{1y} \\ 0 & 0 & 0 & \Omega_{1z} & 0 & -\Omega_{1x} \\ 0 & 0 & 0 & -\Omega_{1y} & \Omega_{1x} & 0 \end{bmatrix}$$

On récupère les différentes vitesses angulaires Ω parmi les entrées qui composent le vecteur des vitesses généralisées v , soit :

$$v = J(q)\alpha = J(q)\dot{q} = J(q)\begin{bmatrix} \dot{\theta}_1 & \dot{\theta}_2 & \dot{\theta}_3 & \dot{\theta}_4 & \dot{\theta}_5 & \dot{\theta}_6 \end{bmatrix}^T$$

En explicitant les entrées de v on a donc :

$$v = \begin{bmatrix} \dot{x}_1 & \dot{y}_1 & \dot{z}_1 & \Omega_{1x} & \Omega_{1y} & \Omega_{1z} & \dot{x}_2 & \dot{y}_2 & \dot{z}_2 & \Omega_{2x} & \Omega_{2y} & \Omega_{2z} & \dot{x}_3 & \dot{y}_3 & \dot{z}_3 & \Omega_{3x} & \Omega_{3y} & \Omega_{3z} \end{bmatrix}^T$$

Les matrices W_2 et W_3 sont :

$$W_2 = \begin{bmatrix} 0 & -\Omega_{2z} & \Omega_{2y} & 0 & 0 & 0 \\ \Omega_{2z} & 0 & -\Omega_{2x} & 0 & 0 & 0 \\ -\Omega_{2y} & \Omega_{2x} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\Omega_{2z} & \Omega_{2y} \\ 0 & 0 & 0 & \Omega_{2z} & 0 & -\Omega_{2x} \\ 0 & 0 & 0 & -\Omega_{2y} & \Omega_{2x} & 0 \end{bmatrix}$$

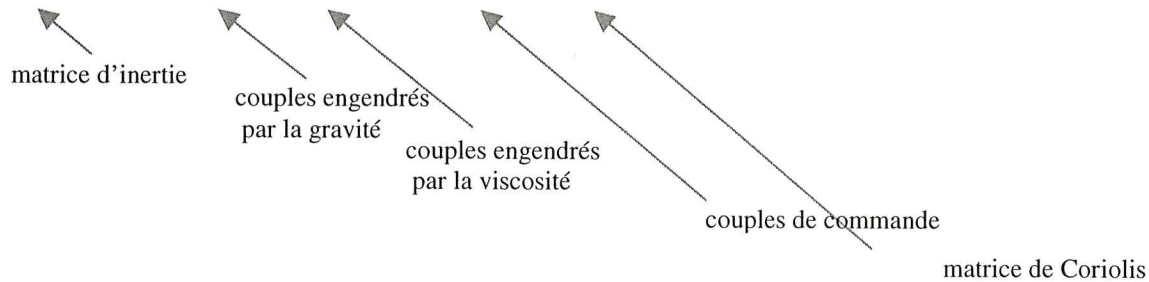
$$W_3 = \begin{bmatrix} 0 & -\Omega_{3z} & \Omega_{3y} & 0 & 0 & 0 \\ \Omega_{3z} & 0 & -\Omega_{3x} & 0 & 0 & 0 \\ -\Omega_{3y} & \Omega_{3x} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\Omega_{3z} & \Omega_{3y} \\ 0 & 0 & 0 & \Omega_{3z} & 0 & -\Omega_{3x} \\ 0 & 0 & 0 & -\Omega_{3y} & \Omega_{3x} & 0 \end{bmatrix}$$

2.8 Étape 8-9-10

Afin de simplifier les développements, ces trois dernières étapes sont réalisées simultanément.

Le modèle dynamique du manipulateur est le suivant :

$$[D]\ddot{\theta} = [G] + [V]\dot{\theta} + [\Gamma] - [C]\dot{\theta}$$



Les termes de l'équation du modèle dynamique sont développés ci-bas :

$$[D] = J^T M J$$

$$[C] = J^T M \dot{J} + J^T W M J$$

$$[\Gamma] = [\Gamma_1 \quad \Gamma_2 \quad \Gamma_3 \quad \Gamma_4 \quad \Gamma_5 \quad \Gamma_6]^T$$

Le vecteur $[\Gamma]$ qui représente les couples de commande aux actionneurs qui sont les entrées du système dynamique.

$$[V] = [-V_1 \quad -V_2 \quad -V_3 \quad -V_4 \quad -V_5 \quad -V_6]^T$$

Ici $[V]$ est le vecteur contenant les constantes de friction visqueuses associées à chaque coordonnée généralisée. Ces valeurs sont évidemment négatives car elles engendrent des couples dissipatifs.

Construction de $[G]$

L'effet de la gravité sera modélisé dans cette section. Le vecteur $[G]$ comporte les termes suivants :

$$[G] = [G_1 \quad G_2 \quad G_3 \quad G_4 \quad G_5 \quad G_6]$$

La figure 2 montre les points d'application de la force de gravité sur le manipulateur. La figure 3 est une vue de profil du système et précise les distances pertinentes à considérer dans le but de modéliser l'effet de la gravité.

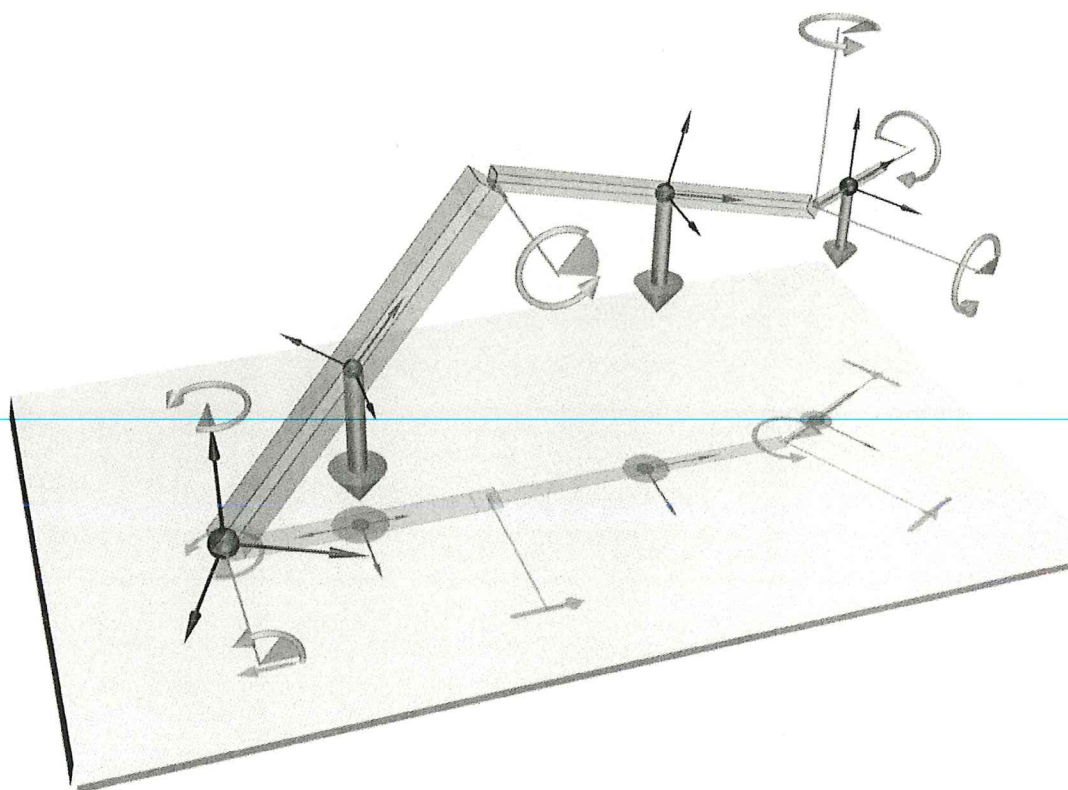


Figure 2 – Points d'application de la force de gravité

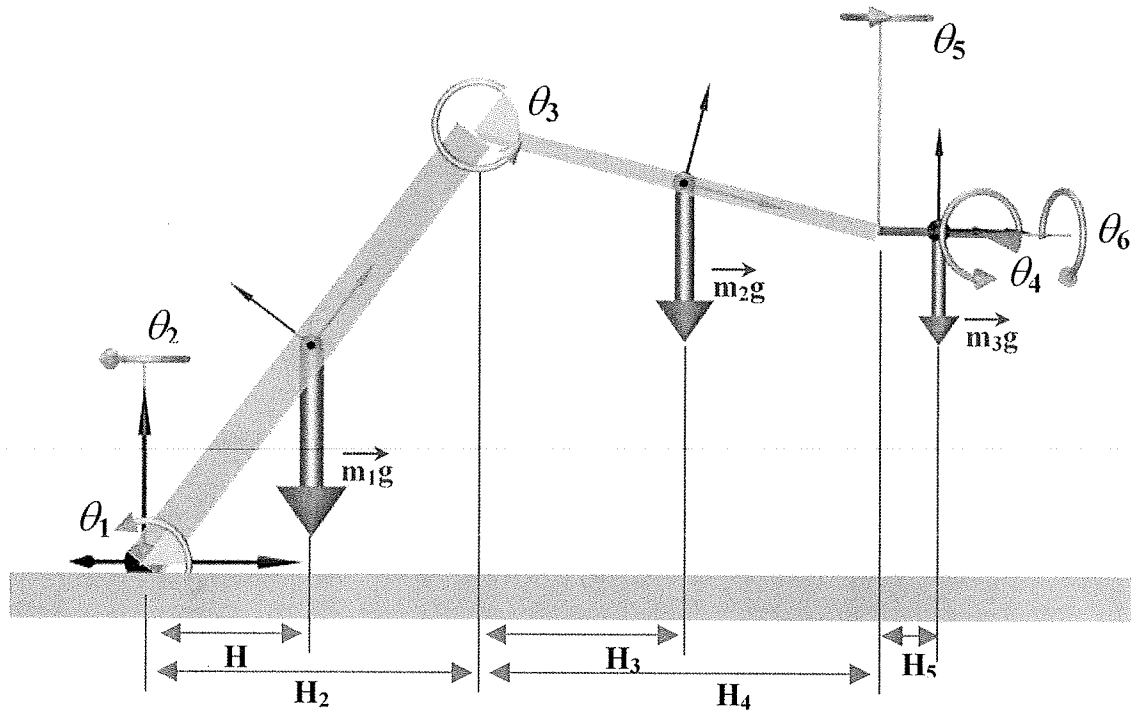


Figure 3 – Manipulateur, vue de profil

Développement des termes G_1 , G_2 et G_3

Les trois premiers termes du vecteur $[G]$ seront tout d'abord développés.

$$G_1 = -m_1gH_1 - m_2g(H_2 + H_3) - m_3g(H_2 + H_4 + H_5)$$

$$G_2 = 0$$

$$G_3 = -m_2gH_3 - m_3g(H_4 + H_5)$$

où les distances H_1 , H_2 , H_3 , H_4 et H_5 sont :

$$H_1 = \left(Rot(z, \theta_1) \begin{bmatrix} a \\ 0 \\ 0 \end{bmatrix} \right)_x$$

$$H_2 = \left(Rot(z, \theta_1) \begin{bmatrix} L_1 \\ 0 \\ 0 \end{bmatrix} \right)_x \quad H_3 = \left(Rot(z, \theta_1) Rot(1, 2) \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix} \right)_x$$

$$H_4 = \left(\text{Rot}(z, \theta_1) \text{Rot}(1, 2) \begin{bmatrix} L_2 \\ 0 \\ 0 \end{bmatrix} \right)_x$$

$$H_5 = \left(\text{Rot}(z, \theta_1) \text{Rot}(1, 2) \text{Rot}(2, 3) \begin{bmatrix} e \\ 0 \\ 0 \end{bmatrix} \right)_x$$

Développement des termes G_4 , G_5 et G_6

Pour identifier ces termes, la procédure est la suivante :

À partir du repère inertiel où le vecteur gravité est orienté selon l'axe $-y$, on projette le vecteur m_3g dans le repère 3 :

$$[m_3g]_3 = \text{Rot}(2, 3)^T \text{Rot}(1, 2)^T \text{Rot}(0, 1)^T \begin{bmatrix} 0 \\ -m_3g \\ 0 \end{bmatrix}_0$$

On fait le produit vectoriel du vecteur position $\overrightarrow{P_{j2}P_{CDM_3}}_3$ avec $[m_3g]_3$ pour trouver le vecteur du couple engendré par la force de gravité sur le centre de masse du 3^e segment :

$$[C_{m_3g}]_3 = \overrightarrow{P_{j2}P_{CDM_3}}_3 \times [m_3g]_3$$

En faisant la projection (produit scalaire) de $[C_{m_3g}]_3$ sur le vecteur unitaire orienté selon l'axe de rotation de θ_4 on obtient G_4 . Similairement, en faisant la projection de $[C_{m_3g}]_3$ sur le vecteur unitaire orienté selon l'axe de rotation de θ_5 on obtient G_5 . G_4 , G_5 et G_6 sont donc :

$$G_4 = \left[\begin{bmatrix} e \\ 0 \\ 0 \end{bmatrix} \times [m_3g]_3 \right] \cdot \left[\text{Rot}(z, \theta_4) \text{Rot}(y, \theta_5) \text{Rot}(x, \theta_6) \right]^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_2$$

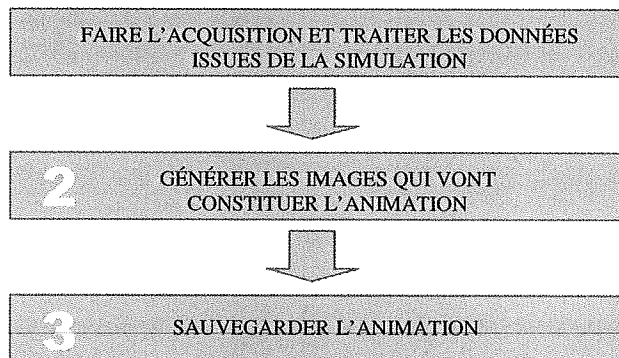
$$G_5 = \left[\begin{bmatrix} e \\ 0 \\ 0 \end{bmatrix} \times [m_3g]_3 \right] \cdot \left[\text{Rot}(y, \theta_5) \text{Rot}(x, \theta_6) \right]^T \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}_2$$

$$G_6 = 0$$

Tous les paramètres de modélisation sont maintenant établis. Il est alors possible de traduire le modèle dynamique du manipulateur à l'aide du programme Simulink. Cette intégration est présentée à la figure 4. On y voit l'arrangement principal où les divers composants sont inter reliés.

Le but du programme de visualisation sera de générer une animation dans laquelle on pourra observer, en trois dimensions, la progression dans le temps d'une simulation faite à partir de Simulink.

Le programme est divisé en trois modules :



3.1 Module 1

Avant d'entreprendre toute action il faut en premier lieu récupérer les données issues de la simulation (Simulink). Pour ce faire, le modèle Simulink devra absolument utiliser le bloc "To Workspace". C'est par celui-ci que les résultats de la simulation sont enregistrés dans l'espace de travail (*Workspace*) de Matlab. Il est pertinent d'exporter les matrices de rotation et les variables qui composent la géométrie du modèle. En effet, ces informations suffisent pour décrire la position et l'orientation de tout corps à chaque instant. La figure 6 illustre cette étape, neuf constantes de géométrie sont exportées vers Matlab.

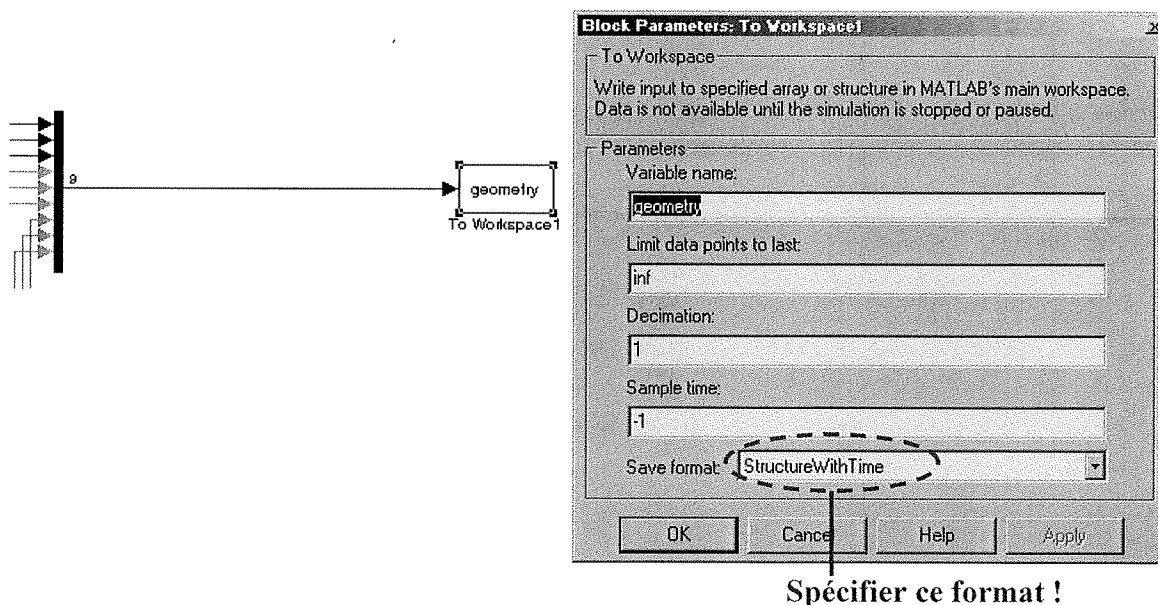


Figure 6 – Acquisition des données de la simulation

Il faut préciser que ces données sont sauvegardées dans Matlab dans un format particulier, le format *Structure*. Il y a donc une manière spécifique d'accéder aux informations exportées. Le code Matlab qui suit permet de le faire :

```

a=geometry.signals.values(1,1,1);
b=geometry.signals.values(1,2,1);
e=geometry.signals.values(1,3,1);
L1=geometry.signals.values(1,4,1);
L2=geometry.signals.values(1,5,1);
L3=geometry.signals.values(1,6,1);
m1=geometry.signals.values(1,7,1);
m2=geometry.signals.values(1,8,1);
m3=geometry.signals.values(1,9,1);

```

On accède simplement aux données en spécifiant le niveau hiérarchique (*variable.signals.values*) puis en indiquant l'index désiré (notation matricielle à trois dimensions).

On fera de même pour les matrices de rotation.

```

ROT_0_1=ROT0_1.signals.values;
ROT_1_2=ROT1_2.signals.values;
ROT_2_3=ROT2_3.signals.values;

```

Il faut noter que Simulink exporte les résultats de simulation en respectant les dimensions du signal qui est envoyé au bloc *"To Workspace"*. Le signal peut donc être une constante fixe ou dynamique, un vecteur colonne, un vecteur ligne, une matrice, ... Cependant, il faut savoir qu'à chaque pas de calcul, Simulink sauvegarde le nouveau signal en incrémentant la troisième dimension de la variable qui sera exportée. Donc on peut voir la construction de la variable exportée comme un empilement de matrice 2D où chaque couche correspond au résultat d'un pas de calcul. Il est important de bien saisir ce concept car c'est la le traitement ultérieur y fait référence.

Dernière étape dans ce premier module, définir la fenêtre de visualisation (nom, position, dimensions), soit :

```
fig=figure('position',[200 200 400 400]);
```

3.2 Module 2

C'est l'étape principale, on crée une boucle à l'intérieur de laquelle une série de commandes graphiques viendront générer la série d'images formant la séquence d'animation. En fait, il s'agit simplement ici de coder en Matlab la fonction de cinématique directe. Voici un exemple qui générera les premières 50 images du déplacement du centre de masse du premier segment du manipulateur :

```

for i=1:50;                                % détermine le nombre nombre d'images (frames) générées

    hold off                                % le graphique suivant remplace complètement le précédent

    axis equal                               % Ajustement du plan de vue...
    axis([-2.5 2.5 -2.5 2.5 -2.5 2.5])       % impose la même échelle aux axes
    set(gca,'CameraPosition',[-226 113 276],... % fixe les limites des axes en X, Y et Z
        'CameraTarget',[0 0 0],...
        'CameraUpVector',[.19 .95 -.24],...
        'CameraViewAngle',[.50])

```

```

view=[-117,48];

CDM1 = ROT_0_1( :, :,i)* [a ;0 ;0]; % vecteur position du centre de masse du premier segment
% fonction de cinématique directe (modélisation, étape 3)

plot3(CDM1(1,1),CDM1(2,1),CDM1(3,1),'r','MarkerSize',15);% point représentant le CDM du premier segment

hold on % le graphique suivant se superpose au précédant

line([0 CDM1(1,1)],[0 CDM1(2,1)],[0 CDM1(3,1)],'Color','b','LineWidth',2) % ligne reliant la base fixe au CDM

% Ajout des axes du repère inertiel...
line([0 2],[0 0],[0 0],'Color','k','LineWidth',1) % ligne représentant l'axe X
text(2,0,0,'\bfX','FontSize',13) % texte identifiant l'axe X
line([0 0],[0 2],[0 0],'Color','k','LineWidth',1)%axe Y % ligne représentant l'axe Y
text(0,2,0,'\bfY','FontSize',13) % texte identifiant l'axe Y
line([0 0],[0 0],[0 2],'Color','k','LineWidth',1)%axe Z % ligne représentant l'axe Z
text(0,0,2,'\bfZ','FontSize',13) % texte identifiant l'axe Z

px=[ 2.5 -2.5 -2.5 2.5]; % surface translucide...
py=[ 0 0 0 0]; % coordonnée en X des 4 coins de la surface
pz=[ 2.5 2.5 -2.5 -2.5]; % coordonnée en Y des 4 coins de la surface
fill3(px,py,pz,[0 0 1],'EdgeColor','none') % coordonnée en Z des 4 coins de la surface
alpha(.05) % fonction qui génère la surface
% pourcentage d'opacité : de 0 = transparent à 1 = opaque

Mov(i) = getframe(fig); %compile les images en une séquence d'animation

end

```

Ce module présente donc toutes les notions de bases nécessaires. En ajoutant les autres éléments de la géométrie on pourra alors représenter le système dynamique au complet, comme montré à la figure 5.

3.3 Module 3

Il suffit ici de sauvegarder l'animation dans un format vidéo standard (.avi) pour ainsi la stocker, la visionner ou encore la diffuser ultérieurement. Il est utile d'employer un menu interactif qui demande à l'utilisateur s'il désire ou non faire la sauvegarde de l'animation. De cette façon, on pourra utiliser le même programme pour visualiser la simulation et au choix sauvegarder l'animation si elle nous convient. Le code Matlab qui suit permet cette fonctionnalité :

```

button = questdlg('Voulez-vous sauvegarder l'animation?',... % menu interactif
'Sauvegarde','oui','non','non');
if strcmp(button,'oui'); % exécute la sauvegarde de l'animation

[file_name, path_file]=uiputfile('anim_6DDL_', 'Enregistrer sous'); % fenêtre de sauvegarde

cd_=cd; % fait la sauvegarde dans le path choisi par l'utilisateur
if ~strcmp(cd_,path_file);
cd(path_file);
end

movie2avi(Mov,file_name,'compression','Cinepak','quality',100,'keyframe',3); % traduit l'animation en format vidéo .avi

end

```

Ces trois modules peuvent être utilisés pour tout type de visualisation provenant d'une simulation faite à partir de Simulink. Bien entendu, d'autres fonction ou modules peuvent s'y greffer pour ainsi s'adapter sur mesure à chaque situation. Le but ici est de jeter les bases pour qu'un néophyte puisse utiliser cet outil très efficace qu'est la visualisation 3D de données sous Matlab.

Note : Le module Simulink et le code Matlab associé à la visualisation sont disponibles sur demande: manuel.theberge@polymtl.ca.

4 Asservissement en position / force

Dans cette section, un asservissement en position et force sera réalisée avec le manipulateur. La mise en situation est la suivante.

On désire polir une surface avec une force constante de 10 N. De plus le centre de l'outil qui poli devra suivre une trajectoire circulaire ayant 40 cm de diamètre (cercle dans le plan ZX, centré aux coordonnées [1.00, 0.80, 0.00] dans l'espace opérationnel).

Afin de réaliser l'asservissement en force il faut disposer d'un capteur de force. Sachant que la simulation peut donner la position de tous les points du manipulateur à chaque instant, on peut utiliser ces informations pour modéliser un retour de force.

En effet, en supposant que le bout de l'outil du bras articulé possède un capteur de force composé d'un système ressort-amortisseur, on pourra alors utiliser les positions relatives entre le cercle à polir et le bout de l'outil à chaque instant et ainsi simuler la force de contact. Le capteur de force est illustré à la figure 7.

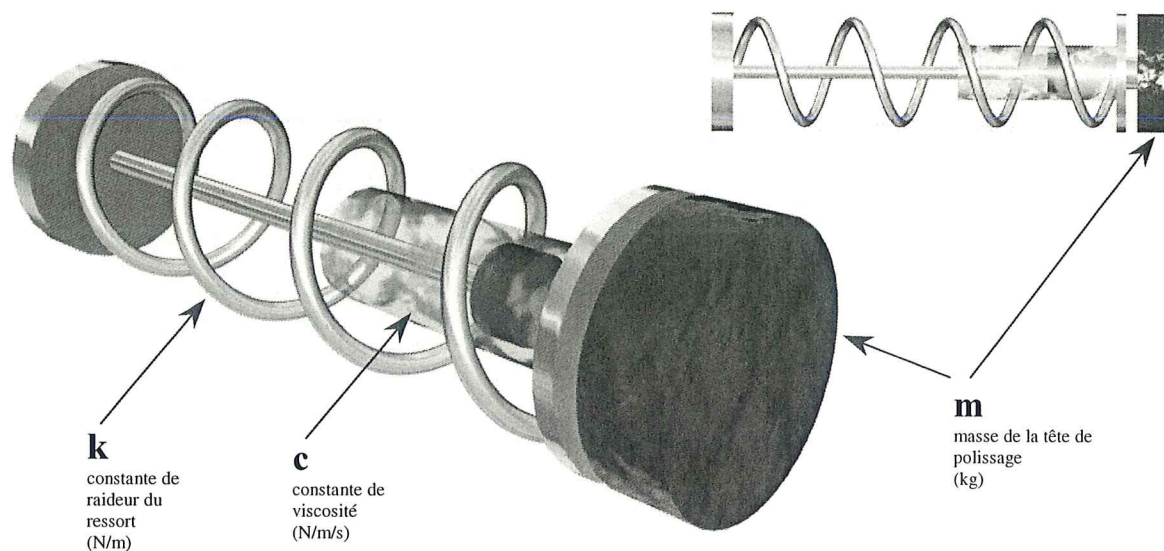


Figure 7 – Capteur de force

4.1 Modélisation du capteur de force

On doit fixer les valeurs des constantes m , k et c . Dans un premier temps, on peut évaluer la masse de la tête de polissage (moteur + surface abrasive) à 700 g (estimation). Ensuite, comme on veut limiter le débattement du capteur en compression, on peut alors fixer la valeur de k en utilisant la loi de Hooke ($F=kx$). Donc si on désire un débattement de 1 cm pour une force de contact (polissage) de 10 N, on obtient alors $k=1000$ N/m. Enfin, pour fixer la constante de viscosité on choisira un amortissement critique. En effet, on veut éviter qu'il y ait des oscillations au moment du contact. Sachant que l'amortissement critique est donné par $c_{cr} = 2\sqrt{km}$, on obtient alors $c=52.9$ N/m/s.

La fonction de transfert du capteur est :

$$H(s) = ms^2 + cs + k$$

Une précision cependant, en situation de contact, la masse sismique (tête de polissage) est immobile dans la direction de l'application de la force de polissage, donc lors du polissage on peut simplifier la fonction de transfert en utilisant seulement les deux derniers termes de $H(s)$, soit :

$$H(s) = cs + k$$

4.2 Schéma de contrôle

Le contrôleur hybride (position/force) est le suivant :

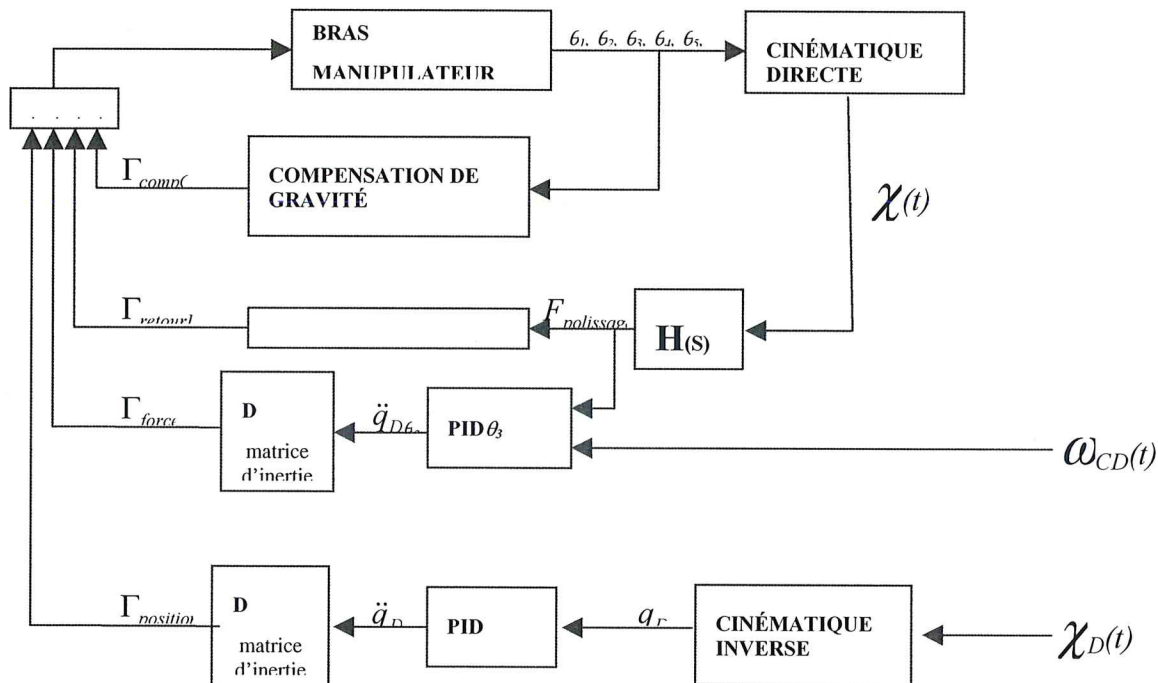


Figure 8 – Schéma du contrôleur hybride

Le vecteur $\Gamma_{retourF}$, soit le couple engendré par la force de polissage se calcule comme suit :

$$\Gamma_{retourF} = \begin{bmatrix} -F_{polissage}(H_2 + H_4) \\ 0 \\ -F_{polissage}(H_4) \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

4.3 Simulation

Les résultats de la simulation est illustré à la figures 9. On peut clairement apercevoir la trajectoire circulaire que le bras suit pendant le polissage. L'erreur sur la force de polissage est d'environ 1%.

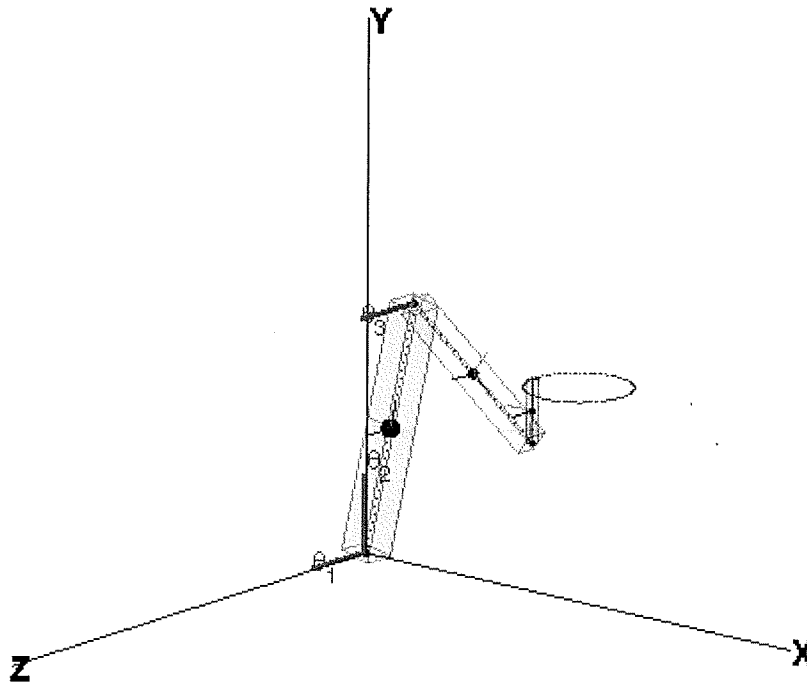


Figure 9 – Trajectoire suivie par le bras lors du polissage

Le temps d'établissement est très court. En effet, il faut se rappeler qu'il s'agit de contact rigide donc la force engendrée par le couple de commande est instantanément transmise.

5 Conclusions

La modélisation d'un manipulateur à six degrés de liberté, la description des outils de base pour la simulation et la visualisation 3D, ainsi qu'un exemple d'application en contrôle hybride ont été couverts.

Les résultats obtenus démontrent bien l'atteinte des objectifs visés par notre étude, à savoir: : a) augmenter l'ensemble d'exemples non triviaux de systèmes mécaniques illustrant la procédure de modélisation de Kane telle que récemment formalisée par De Santis; b) mettre à la disposition de la communauté scientifique un outil de recherche que nous considérons intéressant dans le cadre d'études sur la commande de systèmes robotiques; c) illustrer le rôle que Matlab et Simulink peuvent jouer dans le cadre de ce genre d'étude

6 Bibliographie

- [1] De Santis, R.M., 2000a, *Modélisation de la dynamique des systèmes mécaniques sous contraintes holonomes et nonholonome*, **EPM/RT-2000/XX**, École Polytechnique de Montréal.
- [2] Kane, T.R., Levison, D.A., **Dynamics: Theory and Applications**, McGraw Hill , New York 1985.
- [3] Saha, K.S., Angeles, J., Dynamics of Nonholonomic Mechanical Systems Using a Natural Orthogonal Complement, **Trans of the ASME Journal of Applied Mechanics**, Vol 58, 238-243 (1991).
- [4] Craig, J.J., *Introduction to Robotic*, Addison Wesley 1989.

ÉCOLE POLYTECHNIQUE DE MONTRÉAL



3 9334 00268464 3

École Polytechnique de Montréal
C.P. 6079, Succ. Centre-ville
Montréal (Québec)
H3C 3A7