

# **Modélisation cinématique et dynamique du robot industriel IRB2400 de l'ECAM**

Ing. T. LATOUR  
ECAM – Bruxelles

*Simuler le comportement d'un système multicorps acquiert toute son importance lorsqu'il s'agit d'un manipulateur à mouvements rapides et précis. Dans le but de mieux connaître son évolution spatiale, il est nécessaire de le modéliser. La cinématique directe précède l'étude dynamique. Une simulation informatique est alors réalisée à partir de trajectoires prédéfinies.*

*Mots-clefs : robot, modélisation, simulation, dynamique, cinématique*

*Simulating the behaviour of a multibody system acquires all its importance when it acts of a manipulator with fast and precise movements. With as aim, a better knowledge of its space evolution, it is necessary to model it. Forward kinematics precedes the dynamic study. A computerized simulation is then carried out from preset trajectories.*

*Keywords : robot, modeling, simulation, dynamics, kinematics*

## 1. Introduction

### 1.1 Pourquoi une modélisation ?

La complexité croissante des tâches assignées aux robots manipulateurs implique une étude mathématique plus rigoureuse des différents mouvements lors de la conception. Il est alors possible de dimensionner les actionneurs en fonction des couples maximums requis aux articulations. Ces derniers sont fonction des performances prédéfinies : vitesse de déplacement et charge, ainsi que de la constitution du bras : inertie, masse, frottement.

L'étude de la morphologie de ce bras permet une définition rapide de l'encombrement spatial et des éventuelles collisions lors d'un déplacement programmé.

Enfin, l'adjonction d'une boucle de réglage peut être optimisée grâce à une meilleure connaissance du modèle dynamique de l'ensemble du manipulateur.

### 1.2 Présentation du robot

L'outil étudié est un manipulateur à 6 axes non pourvus de capteurs externes. Ceux-ci sont mus par des actionneurs électriques du type synchrone triphasé. Son rôle dans la cellule flexible de l'ECAM est d'assurer la liaison entre un poste d'usinage et une bande transporteuse.

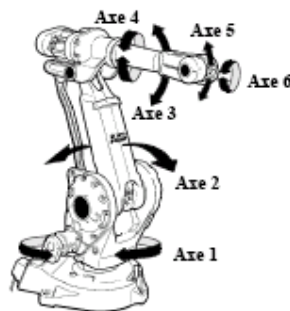


Figure 1 : Le robot à 6 axes ABB IRB2400

### 1.3 Structure de l'étude

L'établissement d'un modèle cinématique direct constitue la première étape. Les équations qui en ressortent expriment la relation entre la configuration articulaire du robot et la position (dans le référentiel cartésien) de

l'extrémité du bras. Cette étude cinématique décrit l'encombrement spatial mais est aussi la base de l'étude dynamique puisqu'elle intervient dans les équations de Newton-Euler.

Le modèle dynamique est alors établi et en programmant une trajectoire, une simulation informatique peut être effectuée. C'est le logiciel *Matlab* qui a ici offert toute sa puissance de calcul. L'outil *Simulink* étant en outre particulièrement adapté à la simulation dynamique.

## 2. Modèle cinématique

### 2.1 Changement de repère et matrice homogène

S'il est question de manipulateur, il en découle automatiquement un déplacement spatial. Celui-ci doit être exprimé dans un référentiel unique afin de mettre en commun l'influence de la vitesse, accélération et position de chaque articulation. L'idée maîtresse est donc l'expression d'un repère fixe dans lequel tous les vecteurs peuvent être exprimés. L'origine de ce repère coïncide avec le centre de la base du robot.

Une matrice, appelée homogène, lie les vecteurs de coordonnées X, Y et Z de 2 articulations. Il est ainsi possible d'exprimer de fil en aiguille le vecteur position de l'outil (décrit depuis le repère de la dernière articulation) dans le référentiel de base. Le résultat de la multiplication de toutes les matrices homogènes simples est la matrice homogène globale telle que présentée à la figure 2 (MCD).

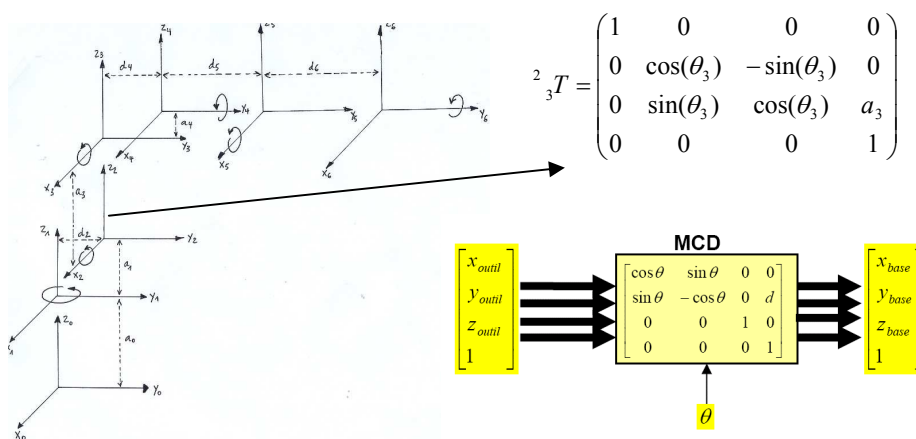


Fig. 2 : Matrice homogène et agencement des 7 repères

## 2.2 Modèle cinématique inverse

La cinématique décrite jusqu'ici est dite directe, c'est-à-dire, qu'à partir d'une configuration donnée du robot (connaissance des différents angles des articulations), nous sommes passés d'un référentiel (celui de l'outil) à un autre (la base).

Une approche plus utile mais loin d'être aussi simple que la cinématique directe, est la cinématique inverse. C'est celle qui intéresse le plus les utilisateurs de robots, puisqu'à partir d'un point de l'espace à atteindre, la configuration géométrique (dépendant des angles articulaires) doit être déduite.

Le problème met en place  $n$  équations fortement non linéaires à  $n$  inconnues. Dans un cas linéaire, nous pouvons connaître le nombre de solutions à partir du déterminant du système. Ce n'est plus le cas ici. Avec  $n$  équations à  $n$  inconnues, nous pouvons très bien obtenir une infinité de solutions.

Plusieurs techniques ont été utilisées dans le cadre du travail. Seules deux seront exposées : l'approche algébrique et l'approche optimisation.

### 2.2.1 Approche algébrique

En plus d'être fortement non linéaire, le système d'équations est transcendant. En effet, les matrices homogènes impliquent une quantité énorme de sinus et cosinus rendant la résolution encore plus complexe.

$$\begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix} = {}^0_1T \cdot {}^1_2T \cdot {}^2_3T \cdot {}^3_4T \cdot {}^4_5T \cdot {}^5_6T \cdot \begin{bmatrix} X_6 \\ Y_6 \\ Z_6 \\ 1 \end{bmatrix} \quad (2)$$

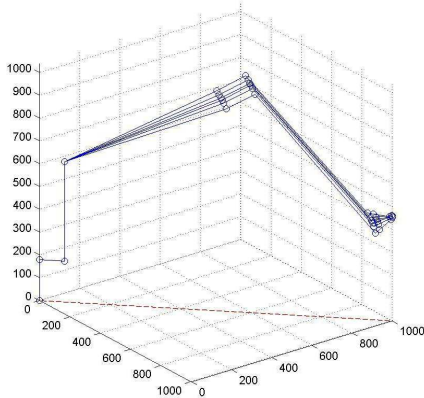
La technique de la substitution de variables peut être appliquée grâce aux relations suivantes :

$$\cos(\theta) = \frac{1-u^2}{1+u^2} \quad \text{et} \quad \sin(\theta) = \frac{2u}{1+u^2} \quad \text{avec} \quad u = \tan\left(\frac{\theta}{2}\right) \quad (3)$$

Dans le but de faciliter les calculs gigantesques avec 6 ddl (degrés de liberté), seulement 4 des 6 articulations sont étudiées. Les autres, moins significatives dans la morphologie du bras, sont supposées fixes.

Par ailleurs la méthode algébrique convient bien aux mécanismes de 4 ddl maximum.

Une fois la substitution effectuée, la fonction *solve* de *Matlab* (toolbox *symbolic*) permet une résolution du système de 3 équations polynomiales, l'orientation de la 1<sup>ère</sup> articulation étant fixée (fig.3).



Articul.	Amplitude (rad)	Amplit. (°)
$\theta_1$	-0,785	-45
$\theta_2$	-1,0029	-57,46
$\theta_3$	0,24	13,75
$\theta_4$	0	0
$\theta_5$	-0,060	-3,44
$\theta_6$	0	0

Fig.3 : Illustration des solutions multiples et valeurs particulières pour une cible en [1000 1000 500] mm ([X Y Z])

### 2.2.2 Résolution numérique

Le système (2) peut être réécrit comme suit :

$$\begin{aligned} x_0 &= g_1(\theta_1, \theta_2, \dots, \theta_6) \\ y_0 &= g_2(\theta_1, \theta_2, \dots, \theta_6) \\ z_0 &= g_3(\theta_1, \theta_2, \dots, \theta_6) \end{aligned} \quad (4)$$

En dérivant chaque fonction par rapport à chaque angle, le système devient ;

$\Delta X = J\Delta\theta$ ,  $J$  étant la matrice jacobienne contenant les dérivées premières.

$$J = \begin{pmatrix} \frac{\partial g_1}{\partial \theta_1} & \frac{\partial g_1}{\partial \theta_2} & \dots & \frac{\partial g_1}{\partial \theta_6} \\ \frac{\partial g_2}{\partial \theta_1} & \frac{\partial g_2}{\partial \theta_2} & \dots & \frac{\partial g_2}{\partial \theta_6} \\ \frac{\partial g_3}{\partial \theta_1} & \frac{\partial g_3}{\partial \theta_2} & \dots & \frac{\partial g_3}{\partial \theta_6} \end{pmatrix} \quad (5)$$

Le théorème des accroissements finis nous énonce que  $f(\theta + h) = f(\theta) + J(\theta)h$  puisque  $J$  est la matrice des dérivées premières. Si le but est de trouver une racine de la fonction  $f(\theta)$ , on peut isoler  $h$  et écrire

$h \approx J^{-1}(\theta)f(\theta + h)$ , ou Taylor limité à l'ordre 1, puisque  $\theta$  annule la fonction. Puisque  $h = \theta(i) - \theta(i+1)$ ;  $\theta(i+1) = \theta(i) - J^{-1}(\theta)f(\theta + h)$  : formule bien connue de la descente de gradient.

Cette technique est utilisée en définissant les fonctions Erreur  $f(\theta)$  :

$$F = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ 1 \end{bmatrix} = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot {}^3T_4 \cdot {}^4T_5 \cdot {}^5T_6 \cdot T_6 \begin{bmatrix} x_6 \\ y_6 \\ z_6 \\ 1 \end{bmatrix} - \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix} \quad (6)$$

Afin de limiter la lourdeur des calculs lors de la présente étude, seulement 3 angles ont été optimisés, les 3 autres étant fixés. La création de la matrice Jacobienne est rendue possible via la fonction de différentiation *diff* de *Matlab*. Le bras tend rapidement vers sa position finale (fig. 4).

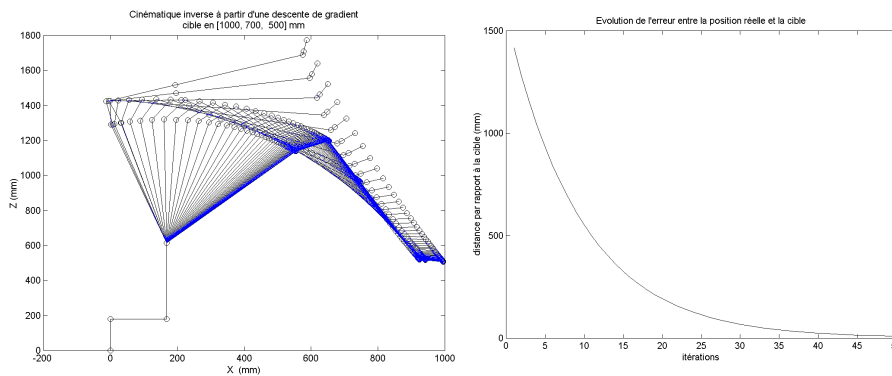


Fig. 4 : Evolution du bras au cours des itérations et à droite évolution de l'erreur de position en fonction du nombre d'itérations.

### 2.2.3 Conclusions

La cinématique inverse apparaît comme un problème complexe encore sujet à de nombreuses études. La méthode numérique offre des résultats sensiblement plus précis que la méthode algébrique : erreur maximum de position de 0,2 mm contre parfois 20 mm pour la méthode algébrique. Une cause de cette imprécision est la longueur des équations, conséquence de la multiplication des 6 matrices homogènes. Une autre piste partiellement exploitée est l'utilisation de routines prédéfinies sous *Matlab*. Il en est ainsi pour la fonction *fminsearch* qui permet de l'optimisation non linéaire sous contraintes pour les 6 angles cette fois, mais a l'inconvénient d'être plus lente que la méthode numérique.

### 2.3 Le Jacobien et son intérêt dans la modélisation

Pour les mécanismes à plus de 4 ddl, la détermination de la matrice jacobienne est beaucoup moins aisée. En se rappelant que la dérivée par rapport au temps d'un espace parcouru est une vitesse

$$\Delta X = J\Delta\theta \text{ devient } {}^0v = {}^0J(\theta)\dot{\theta} \quad (7)$$

où  $v$  est le vecteur des vitesses linéaires exprimées dans le repère de base et  $\dot{\theta}$ , celui des vitesses angulaires de chaque articulation.

Outre l'utilisation en descente de gradient lors de la cinématique inverse, le but premier du jacobien reste la recherche de singularités. Si le robot est en présence de celles-ci, il perd un ou plusieurs degrés de liberté. Autrement dit, l'outil ne peut se mouvoir instantanément dans la direction préalablement déterminée.

Les singularités sont les valeurs d'angles annulant le déterminant du jacobien. En effet, en considérant

$$\dot{\theta} = J^{-1}(\theta)v \quad (8)$$

il n'existe pas de solution unique au problème si la matrice  $J$  n'est pas inversible. Les valeurs singulières sont obtenues via l'instruction `[theta]=solve(det(J))`.

La contrainte est la lourdeur des calculs. Le logiciel retourne d'ailleurs une erreur car la taille des mineurs utilisés dans la fonction `det` est trop grande. Dans le cas évoqué, 4 angles ( $\theta_{1-4-5-6}$ ) ont été fixés et en choisissant 20 valeurs pour  $\theta_2$ , il est possible de trouver consécutivement les points singuliers solutions des équations à une seule inconnue ( $\theta_3$ ).

Les configurations engendrant des singularités peuvent être représentées dans le plan X-Z (fig.5). La zone foncée est l'espace opérationnel du robot pouvant être construit par la cinématique directe.

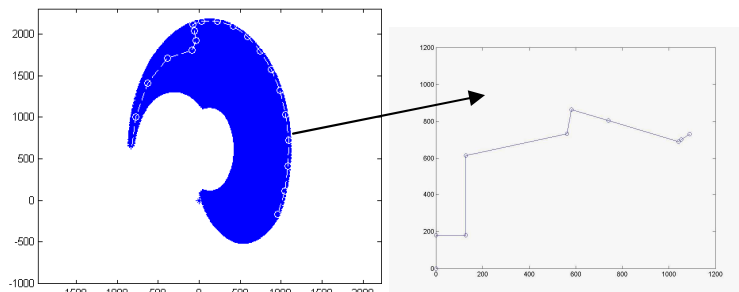


fig. 5 : points singuliers pour 20 valeurs de  $\theta_2$

La représentation de droite à la figure 5 montre la configuration adoptée pour un point singulier particulier situé en [1366 1092 725] mm. Il est intéressant de noter que la majorité des singularités sont de bord (Boundary singularities), présentes lorsque le bras est entièrement déployé.

Il est également à remarquer que dans le cas présent, la matrice jacobienne étant carrée ( $6 \times 6$ ), la recherche du déterminant est possible. Si par contre cette matrice n'est pas carrée, il faut passer par la pseudo-inverse.

### 3. Modélisation dynamique

Le modèle dynamique d'un mécanisme articulé est obtenu à l'aide d'une suite d'équations mathématiques. Cette modélisation est utile dans le cas d'une simulation, le but étant l'obtention de l'évolution des différents angles, vitesses et accélérations angulaires en fonction des couples appliqués aux différentes articulations.

Deux approches sont possibles : l'une partant des équations de conservation des énergies cinétique et potentielle de Lagrange-Euler et l'autre développant l'équation fondamentale de la mécanique, la formule de Newton ( $F = m.a$ ). Cette méthode est celle de Newton-Euler. C'est cette dernière qui a été principalement développée en raison de son algorithme récursif développé plus loin.

#### 3.1 Recherche des tenseurs d'inertie

L'algorithme récursif de Newton-Euler sert à déterminer le couple moteur, appliqué à chaque articulation, nécessaire pour contrer les forces extérieures et le poids des bras agissant depuis l'outil.

Dans les équations de départ ci-dessous,  $F$  et  $N$  sont les couples extérieurs qui, appliqués au centre de masse, provoquent le mouvement du corps.

$$F = m.v_c \text{ et } N = I_c \dot{\omega} + \omega \times I_c \omega \quad (9)$$

L'inertie des différents corps du robot intervient directement dans la formulation du couple. Or ces données ne sont pas fournies par le constructeur. Une idée est la réduction de chaque organe en cylindre simple, permettant un rapide calcul manuel des tenseurs d'inertie.

La solution plus élaborée choisie est la récupération de fichiers DAO du robot. Ceux-ci étant réalisés sous *Catia*, il est possible de calculer l'inertie depuis un référentiel choisi. Les résultats ne font cependant qu'approcher la



réalité puisque ces corps sont supposés pleins lors des calculs, ce qui n'est évidemment pas toujours le cas.

### 3.2 Mise sous forme d'état et simulation

Une fois la formulation finale du couple obtenue, il est plus intéressant de la réécrire sous une pseudo forme d'état. En effet, le couple, système de 4 équations (puisque par facilité, deux articulations sont fixes), fait intervenir la variable angulaire sous ses formes dérivées :  $\theta, \dot{\theta}$  et  $\ddot{\theta}$ . L'un des buts ultimes du travail étant la simulation, l'expression sous forme d'état facilite la construction du modèle dynamique en schéma-blocs (utilisés par le l'outil *Simulink* de *Matlab*). Cette notation matricielle générale est la suivante :

$$\tau = M(\theta)\ddot{\theta} + B(\theta)(\dot{\theta}\dot{\theta}) + C(\theta)(\dot{\theta}^2) + G(\theta) \quad (10)$$

La matrice M est la matrice d'inertie ou de masse (puisque selon le principe de Newton, une force est une masse multipliant une accélération  $\ddot{\theta}$ ). G est un vecteur dépendant uniquement de  $\theta$  et appelé vecteur de Coriolis. C est la matrice centrifuge. Il est à noter que  $\dot{\theta}^2$  n'est pas  $\dot{\theta}\dot{\theta}$ . Ce dernier vecteur comprend les multiplications successives des dérivées angulaires d'angles différents (ex.  $\dot{\theta}_1\dot{\theta}_2$ ).

L'identification de chaque terme afin de recréer les matrices est un travail très laborieux étant donné la longueur des équations du couple. Manuellement ce travail multiplierait les risques d'embrouille et fausserait la formule. Heureusement, la fonction *sym2polys* créée par un utilisateur de *Matlab* permet la factorisation multivariable, si lourde dans notre cas. Il est cependant nécessaire de mettre en place un algorithme de recherche récursif étant donné la diversité de combinaisons de variables.

Dans le cas de la simulation du mouvement du bras suite à une variation de couple moteur, il est plus intéressant d'écrire l'équation (10) sous la forme suivante :

$$\ddot{\theta} = M^{-1}(\theta)[\tau - B(\theta)(\dot{\theta}\dot{\theta}) - C(\dot{\theta}^2) - G(\theta)] \quad (11)$$

La construction du schéma-bloc peut alors réellement débuter. Les matrices d'état sont réalisées à partir de fonctions appelées à chaque temps d'horloge. Le schéma global pour le robot à 4 axes mobiles est illustré ci-dessous (fig.6). Le bloc grisé contient le régulateur PID. Toutes les variables sont

retournées vers l'espace de travail de *Matlab*, ce qui permet de tracer l'évolution du bras et de développer l'animation.

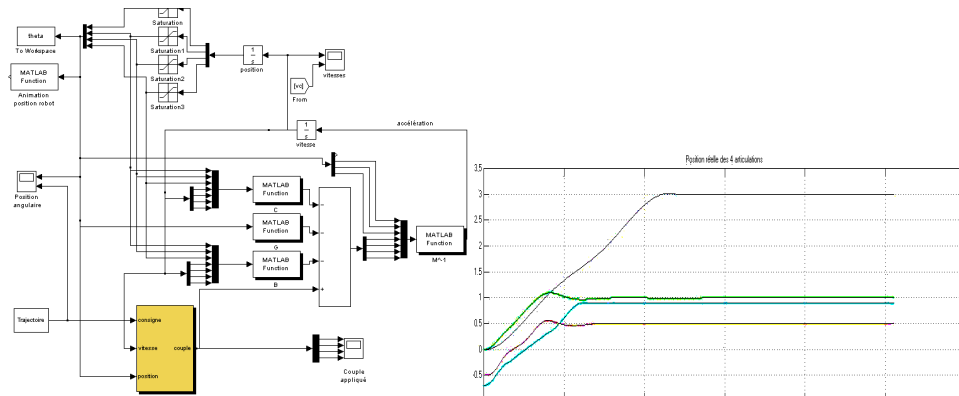


fig. 6 : schéma-bloc du bras réglé et évolutions des positions angulaires

### 3.3 Synthèse d'un régulateur

Une fois le modèle dynamique du robot établi, le plus intéressant reste la commande du manipulateur. Cette matière a fait l'objet de nombreuses études. Certains y ont appliqué les réseaux de neurones artificiels, d'autres encore utilisent le réglage flou. La régulation n'étant pas l'objet principal du travail, un simple régulateur PID a été adjoint au bras. L'avantage est la rapidité de calcul si on ramène la formulation d'état à un système linéaire du second ordre. Cependant, cette technique approchée puisqu'elle utilise une linéarisation du système réel, ne peut être entièrement précise. Un dépassement ainsi qu'une difficulté de suivre les mouvements rapides est à craindre. Dans notre cas, les mouvements n'étant pas excessivement rapides ( $150^\circ/\text{s}$  max), le PID s'accommode bien. De plus, en imposant de plus gros gains (tout en respectant la réalité des actionneurs), le dépassement reste minime (cfr. figure 6 à droite).

### 3.4 Génération de trajectoires

La consigne de position ne peut varier instantanément. Une consigne sous forme d'échelon est tout à fait irréaliste en raison du dépassement immense que cela engendrerait mais surtout vis-à-vis des actionneurs tout à fait incapables de mouvoir le bras durant un temps infiniment court.

Pour cette raison, il y a lieu de créer une trajectoire. Deux d'entre elles seront détaillées ici.

La première, la plus classique, interpole par splines cubiques entre deux points extrêmes. En entrant comme consigne une dérivée nulle au départ et à l'arrivée (= vitesse nulle), il est possible de construire un polynôme, généralement de degré 3, liant les deux (ou plus) points de l'espace angulaire. Cette trajectoire est fonction du temps. Il faut donc rentrer comme paramètre la vitesse maximum de déplacement désirée entre ces points (ici  $150^\circ/\text{s}$ ). La figure 6 (à droite) montre le suivi de consigne lors d'une loi polynomiale.

Cette trajectoire n'est cependant pas celle retenue par les constructeurs. Chez ABB, l'instruction la plus utilisée porte le nom de *MoveJ*. Dans un atelier de production flexible, il est préférable de synchroniser les mouvements des différentes articulations. De la sorte, la vitesse de chaque axe est adaptée afin que toutes les rotations se terminent au même moment. L'articulation qui doit effectuer la plus grande course acquiert la vitesse maximum tandis que les autres l'adaptent en fonction de leur propre course. Une fonction rapidement créée sous *Matlab* permet de programmer cette trajectoire dont l'évolution des vitesses a une forme trapézoïdale (figure 7 à gauche). La fonction s'exécute à partir des valeurs angulaires initiales et finales souhaitées, de l'accélération et de la vitesse maximale, qui fixeront les paramètres du trapèze. Cette forme est due à l'accélération angulaire, choisie commune à tous les joints. L'avantage d'imposer un tel parcours angulaire en simulation est de pouvoir par animation percevoir le déplacement du bras dans l'espace cartésien (figure 7 à droite). A la figure 8, on remarque la synchronisation de fin de course pour toutes les articulations.

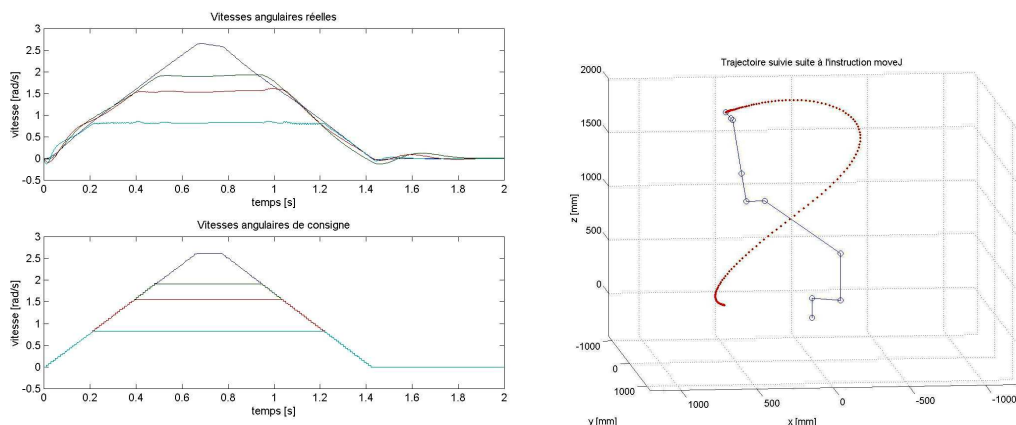


fig.7 : Suivi de vitesses réel et désiré (gauche) et trajectoire cartésienne

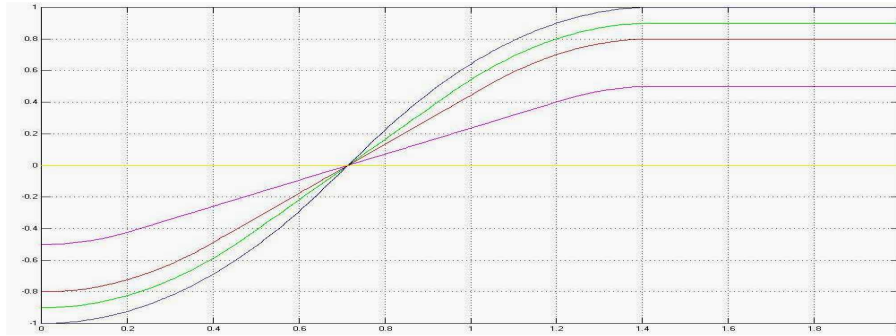


fig.8 : Evolution des positions angulaires au cours du temps

#### 4. Conclusions

Tout d'abord, il est à noter que la modélisation ne s'arrête pas là. Dans le cadre du travail, le moteur électrique d'une des articulations a été modélisé. On peut de ce fait se rapprocher de plus en plus de la réalité même si certaines hypothèses ont été posées, telles que l'accélération angulaire, tout en gardant une certaine logique matérielle. De même, les inerties et masses n'ont été qu'approchées.

L'aboutissement de cette étude serait le relevé de mesures en temps réel sur le robot afin de comparer la cohérence du modèle. C'est dans le cas présent par manque d'expérience quant au démontage et réassemblage du robot, que de tels essais n'ont pu être pratiqués.

Le développement le plus intéressant reste sans conteste la prévision des mouvements. Eviter les collisions est désormais possible.

En ce qui concerne l'approche dynamique, la voie est ouverte vers une étude plus approfondie dans le domaine de la régulation. L'utilisation d'un régulateur simple mais universel offre déjà des résultats plus que satisfaisants.

La conscience des limitations physiques du système a dû être constamment présente. Rappelons que la synthèse d'une loi de commande doit prendre en compte une saturation inévitable des organes de commande.

La voie vers l'animation est également ouverte. Elle fut ici légèrement abordée en représentant l'évolution de l'ossature du robot. La puissance d'un outil, tel que *Matlab*, met à disposition de l'utilisateur un panel

d'outils graphiques (*Graphical User Interface*). Il est d'ailleurs possible de partir de créations existantes puisque bon nombre de développements ont été réalisés sur d'autres manipulateurs.

Finalement, une suite possible de ce travail est la comparaison des résultats issus des équations dynamiques étudiées avec ceux fournis par des outils dédiés à l'étude de la dynamique des systèmes multicorps (exemple du logiciel *ROBOTRAN* développé par l'UCL).

## 5. Références bibliographiques

- [1] CRAIG J., *Introduction to Robotics, Mechanics and Control, Third Edition*, Australie, International Edition, 2005
- [2] SELIG J-M, *Introductory Robotics*  
Hertfordshire, Prentice Hall, , 1992
- [3] FU K.S. GONZALEZ R.C. LEE C.S.G., *Robotics: Control, Sensing, Vision and Intelligence*,. Mc Graw-Hill International Editions
- [4] CORKE P.I., *A Robotics Toolbox for Matlab (release 6)*  
IEEE Robotics and Automation Magazine, Mai 1996
- [5] GRELLET G. CLERC G.: *Actionneurs électriques : principes, modèles, commande*, collection Electrotechnique, Paris, Eyrolles, 1997
- [6] ABB Robotics Products : *User's Guide*  
3HAB 0009-53 M94A, Vasteras (Suède)
- [7] ABB Robotics Products : *Technical Provisions, Motor Specifications, IRB2000*, Vasteras (Suède), 1991