

P0017/05B

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

المدرسة الوطنية المتعددة التقنيات  
Ecole Nationale Polytechnique

**ECOLE NATIONALE POLYTECHNIQUE**  
Département de Génie Electrique

**PROJET DE FIN D'ETUDES**

En vue de l'obtention du diplôme  
d'ingénieur d'état en Automatique  
Promotion 2005

**Thème**

**Commande d'un Robot Redondant et en Présence  
d'Obstacle Fixe**

**Proposé et Dirigé par :**

**Dr.H.CHEKIREB**

**Etudié par :**

**KADRI Oualid**

**TAAZOUNT Samir**

**E.N.P. 10, Avenue Hassen Badi, 16200 EL-Harrach, Alger**



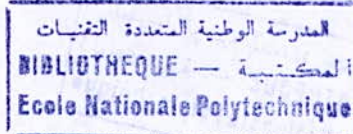
## Remerciements

*Le travail présenté dans ce mémoire a été mené au laboratoire de commande des processus du département de génie Electrique de l'Ecole Nationale polytechnique.*

*Il nous est particulièrement agréable de témoigner notre infinie reconnaissance à monsieur H.CHEKIREB, pour avoir bien voulu nous diriger et nous assister dans notre travail et pour nous avoir fait profiter de leurs précieux conseils.*

*Nous tenons également compte à remercier l'ensemble de nos enseignants d'automatique.*

*En fin nous ne pouvons oublier nos amis: B.Smail, B.Kamel B.Mohamed, Hamache, qui sont pour beaucoup dans l'aboutissement de notre projet grâce à leur présence et encouragement.*



## *Dédicace*

*Je dédie ce modeste travail a mes très chers parents qui m'ont toujours apporter leurs amour et leurs soutient pour affronter les difficultés de la vie.*

*Qu'ils trouvent ici le témoignage de ma reconnaissance la plus dévouée.*

*A mes cher frère Salim, Sofiane et à ma sœur Amina.*

*A tous les membres de ma famille : oncles, tentes, cousins et cousines, en particulier ma cousine Zahia et ma tente Ghania ainsi mes oncles Hakim, Karim, Djamel, Rachid, Rabah.*

*A tous les automaticiens, en particulier : Benssedik, Hamache, Kamel, Boudjellah, Hassen et Nedjma, Saleh, smain, waheb, Mahi et M.smain. ...*

*A tous mes amis en particulier Bilel, A.Bilel, S.Hamza, A.Hamza, Nadjib, Naim et les autres.*

*Samir TAAZOUNT*



# Dédicace

*Je dédie ce modeste travail a mes très chers parents qui m'ont toujours apporter leurs amour et leurs soutient pour affronter les difficultés de la vie.*

*Qu'ils trouvent ici le témoignage de ma reconnaissance la plus dévouée.*

*A mon cher frère Nadjib, et à mes chères sœurs Imène et Manel, ainsi qu'à la mémoire de mon frère Rafik,*

*A tous les membres de ma famille : oncles, tentes, cousins et cousines, en particulier Tata Khadidja et mon oncle amar ainsi Islam, Mohamed, Amel, Hiba,*

*A tous les automaticiens, en particulier : Benssedik, Hamache, Kamel, Boudjellah, Hassen et Nedjma, Saleh, smain, wahab....*

*A tous mes amis en particulier Mimouni, Tsamda de bouaarfa, Ati, Safsafi, Smain, Naim, Noureddine, et les autres.*

*Oualid KADRI*

# Table des Matières

<b>INTRODUCTION GENERALE</b> .....	1
<b>Chapitre 1 : TACHES REDONDANTES POUR LE PUMA 560</b> .....	2
1.1. Description du PUMA 560.....	2
1.1.1. Définition d'un robot redondant.....	2
1.1.2. Présentation du PUMA 560 Redondant.....	2
1.2. Modélisation du PUMA 560.....	3
1.2.1. Modèle géométrique direct « MGD ».....	4
1.2.2. Algorithme de calcul numérique du modèle géométrique inverse.....	4
1.2.3. Modèle cinématique direct « MCD ».....	6
1.3. Présentation des taches redondantes.....	8
1.3.1. Génération de trajectoire dans le plan.....	8
1.3.2. Génération de trajectoire dans l'espace 3D.....	12
1.4. Structure de commande.....	15
1.4.1. Introduction.....	15
1.4.2. Commande par découplage non linéaire dans l'espace articulaire.....	15
1.5. Conclusion.....	18
<b>Chapitre 2 : RESOLUTION DU PROBLEME DE LA CINEMATIQUE      INVERSE BASEE SUR LA PSEUDO INVERSE</b> .....	19
2.1. Introduction.....	20
2.2. Dimension de J.....	20
2.3. Les méthodes basées sur le calcul de la pseudo inverse.....	21
2.3.1. Utilisation de la pseudo inverse sans terme d'optimisation.....	21
2.3.1.1. Utilisation de la pseudo inverse.....	21
2.3.1.1.1. Résultats et simulation.....	22
2.3.1.2. Minimisation d'un critère quadratique.....	23
2.3.1.2.1. Résultats et simulation.....	25
2.3.1.3. Algorithme de Gréville pour calculer la pseudo inverse.....	26
2.3.1.3.1. Résultats et simulation.....	27
2.3.2. Utilisation de la pseudo inverse avec un terme d'optimisation.....	29

2.3.2.1. Utilisation du terme d'optimisation pour l'éloignement des butées articulaires.....	30
2.3.2.1.1. Résultats et simulation.....	31
2.3.2.2. Utilisation du terme d'optimisation pour augmenter la manipulabilité.....	32
2.3.2.2.1. Résultats et simulation.....	32
2.3.3. Utilisation de l'inverse généralisé du jacobien.....	34
2.3.3.1. Utilisation de l'inverse généralisé du jacobien sans terme d'optimisation.....	34
2.3.3.1.1. Résultats et simulation.....	35
2.3.3.2. Utilisation de l'inverse généralisé du jacobien avec un terme d'optimisation pour l'éloignement des butées articulaires.....	36
2.3.3.2.1. Résultats de simulation.....	38
2.4. Conclusion.....	40
<b>Chapitre 3 : LES METHODES BASEES SUR LA DECOMPOSITION DE LA MATRICE JACOBIENNE.....</b>	<b>41</b>
3.1. Nouvelle méthode pour résoudre la cinématique inverse proposé par W.Khalil.....	41
3.1.1. La solution générale du modèle cinématique inverse.....	41
3.1.2. Décomposition de la matrice jacobienne.....	42
3.1.3. Dérivation de la pseudo inverse.....	44
3.1.4. La dérivée de la pseudo inverse combinée avec la solution homogène.....	45
3.1.5. Utilisation d'autre critère .....	46
3.1.5.1. Critère quadratique.....	46
3.1.5.2. Critère min-max.....	47
3.1.6. La pseudo inverse pondéré .....	48
3.1.7. Minimisation des couples articulaires.....	49
3.1.7.1. Résultats et simulation.....	52
3.2. Décomposition en valeurs singulières de la matrice jacobienne (SVD).....	53
3.2.1. Résultats et simulation.....	56
3.3. Méthode des moindres carrés atténuée.....	57
3.3.1. Résultats et simulation.....	59
3.4. Autre méthode.....	60

3.4.1. Ajoute (n-m) contraintes supplémentaires.....	60
3.4.1.1. Résultats et simulation.....	62
3.5. Comparaison entre les méthodes.....	63
2.7. Conclusion.....	63
<b>Chapitre 4 : EVITEMENT D'OBSTACLE.....</b>	<b>64</b>
4.1. Modélisation de l'obstacle.....	65
4.2. Méthodes.....	67
4.2.1. Evitement d'obstacle par l'approche tâche prioritaire.....	67
4.2.2. Evitement d'obstacle par l'approche fonction de coût .....	69
4.2.3. Evitement d'obstacle par gradient de la distance.....	70
4.2.4. Evitement d'obstacle par gradient d'une fonction objectif.....	72
4.3. Simulation.....	75
4.3.1. Résultats sans évitement.....	75
4.3.2. Résultats en utilisant l'approche tâche prioritaire .....	77
4.3.3. Résultats en utilisant l'approche fonction de coût.....	78
4.3.4. Résultats en utilisant le gradient de la distance.....	80
4.3.5. Résultats en utilisant le gradient d'une fonction objective.....	82
4.4. Comparaison des quatre méthodes.....	83
4.5. Conclusion.....	84
<b>CONCLUSION GENERALE.....</b>	<b>85</b>
<b>ANNEXE A</b>	
A.1. Gradient de la distance.....	86
A.1.1. démonstration.....	86
<b>ANNEXE B</b>	
B.1. Algorithme d'évitement d'obstacle par l'approche tâche prioritaire.....	89
B.2. Algorithme d'évitement d'obstacle par l'approche fonction de coût.....	90
B.3. Algorithme d'évitement d'obstacle par l'approche gradient de la distance.....	91
B.4. Algorithme d'évitement d'obstacle par l'approche gradient d'une fonction objective.....	92
B.5. Algorithme de calcul de $X_0$ .....	93
<b>BIBLIOGRAPHIES.....</b>	<b>100</b>

# Table d'illustrations

<b>CHAPITRE 1</b>	
Figure 1.1. PUMA 560.....	3
Figure 1.2. La tâche désirée dans le plan.....	8
Figure 1.3. Variation de $\alpha$ , $\dot{\alpha}$ et $\ddot{\alpha}$ .....	10
Figure 1.4. La tâche désirée dans l'espace.....	13
Figure 1.5. Commande par découplage non linéaire avec correction dans l'espace articulaire.....	17
<b>CHAPITRE 2</b>	
Figure 2.1. Résultats de simulation en utilisant la pseudo inverse sans terme d'optimisation.....	23
Figure 2.2. Résultats de simulation en utilisant la pseudo inverse due à la minimisation d'un critère quadratique.....	25
Figure 2.3. Résultats de simulation en utilisant l'algorithme de Gréville.....	28
Figure 2.4. Résultats de simulation en utilisant la pseudo inverse avec un terme d'optimisation pour l'éloignement des butées articulaires.....	31
Figure 2.5. Résultats de simulation en utilisant la pseudo inverse avec un terme d'optimisation pour augmenter la manipulabilité.....	33
Figure 2.6. Résultats de simulation en utilisant l'inverse généralisée sans terme d'optimisation.....	36
Figure 2.7. Résultats de simulation en utilisant l'inverse généralisé avec un terme d'optimisation pour l'éloignement des butées articulaires.....	39
<b>CHAPITRE 3</b>	
Figure 3.1. La solution $q_c$ en utilisant le critère min-max.....	48
Figure 3.2. Résultats de simulation en utilisant la décomposition de la pseudo inverse avec un terme d'optimisation pour l'éloignement des butées articulaires et la minimisation des couples.....	53
Figure 3.3. L'interprétation géométrique de la décomposition en valeurs singulières.....	54



**Figure 3.4.** la représentation de la transformation des vitesses opérationnelles par la pseudo inverse pour le calcul des vitesses articulaires.....55

**Figure 3.5.** Résultats de simulation en utilisant la décomposition en valeurs singulières de la matrice jacobien (SVD).....57

**Figure 3.6.** Comparaison entre la solution des moindres carrés atténuée et la solution des moindres carrés normale.....58

**Figure 3.7.** Résultats de simulation en utilisant la méthode des moindres carrés atténuée.....60

**Figure 3.8.** Résultats de simulation en ajoutant  $(n - m)$  contraintes supplémentaires.....62

**CHAPITRE 4**

**Figure 4.1.** Un exemple d'évitement d'obstacle.....65

**Figure 4.2.** Choix de  $\alpha_h$ .....68

**Figure 4.3.** Utilisation de la méthode gradient d'une fonction objective pour éviter un obstacle fixe.....72

**Figure 4.4.** Résultats de simulation sans évitement.....76

**Figure 4.5.** Résultats de simulation en utilisant l'approche tâche prioritaire.....78

**Figure 4.6.** Résultats de simulation en utilisant l'approche fonction de coût.....79

**Figure 4.7.** Résultats de simulation en utilisant l'approche gradient de la distance.....81

**Figure 4.8.** Résultats de simulation en utilisant l'approche gradient d'une fonction objective.....83

**Tableau 4.1.** Comparaison entre les 4 techniques d'évitement d'obstacle.....83

# Introduction Générale



Les robots manipulateurs redondants sont particulièrement intéressants pour leur dextérité, leur capacité d'éviter les obstacles et les contraintes mécaniques sans altérer la tâche réalisée par l'effecteur. Cette dernière est toujours spécifiée dans l'espace cartésien appelé aussi l'espace opérationnel, espace où il est plus facile de représenter le mouvement du robot.

C'est pour cela qu'il est naturel de concevoir des lois de commande directement dans cet espace. Dans ce cas, le problème de synthèse de la commande est généralement transformé en un problème de planification de trajectoire de l'espace articulaire à partir de l'espace cartésien. Le passage d'un espace vers un autre est réalisé via des transformations géométriques et cinématique. Ces transformations sont basées sur des modèles direct et inverse décrivant la relation entre les positions cartésiennes en fonction des positions articulaires (modèle géométrique) et les vitesses cartésiennes en fonction des vitesses articulaires (modèle cinématique) du robot. Lorsque le robot est redondant, il existe une infinité de solutions représentant les différentes configurations du robot pour un point dans l'espace cartésien. Le problème à résoudre est donc de déterminer, dans le cas redondant, les positions et les vitesses articulaires optimales correspondant à la tâche demandée. Plusieurs travaux ont été réalisés dans ce domaine.

Un certain nombre de solutions proposées sont basées sur les méthodes d'optimisation non linéaires qui sont difficiles à implémenter et caractérisées par une convergence lente et une possible instabilité. Récemment des travaux basés sur les réseaux de neurones visant à réduire la complexité des calculs dans la procédure de planification de trajectoire, ont été considérées pour la modélisation et l'identification des deux modèles, direct et inverse, des robots manipulateurs redondants. Ces méthodes malgré leurs performances ne permettant pas de donner, en temps réel, une solution nécessaire à l'exécution d'une tâche donnée.

Aux chapitres 2 et 3 on propose des méthodes basées sur la cinématique inverse pour obtenir les vitesses articulaires du robot.



# Tâches redondantes pour le PUMA 560

## 1.1. DESCRIPTION DU PUMA 560 REDONDANT :

### 1.1.1. Définition d'un robot redondant :

Un robot est redondant lorsque le nombre de degré de liberté de l'organe terminal est inférieur au nombre de degrés de liberté de l'espace articulaire (nombre d'articulations Motorisées).

Les robots manipulateurs redondants sont particulièrement intéressants pour leur dextérité, leur capacité d'éviter les obstacles et les contraintes mécaniques sans altérer à la tâche réaliser par l'effecteur. Cette dernière est toujours spécifier dans l'espace cartésien appelé aussi l'espace opérationnel, espace ou il est plus facile de représenter le mouvement du robot.

### 1.1.2. Présentation du PUMA 560 Redondant :

Les robots de la gamme PUMA sont probablement les plus répandus dans les universités et constituent les robots d'assemblages les plus utilisés dans l'industrie.

PUMA signifie "Programmable Universal Machine for Assembly", conçu par Vic Schieman et financé par GENERAL MOTORS au Massachussets institute of Technology au

milieu des années 70. Il fut produit pendant de nombreuses années par unimation, société qui fut rachetée plus tard par Westinghouse, avant d'être revendus à Staubli, une grande société suisse de robotique.

Le PUMA 560 est un robot manipulateur à six articulations rotoïdes lui conférant six degrés de libertés. A ce titre, il est non redondant dans l'espace tridimensionnel, car c'est la le nombre minimum de degrés de liberté nécessaires pour placer arbitrairement un corps dans l'espace : trois pour la position et trois pour l'orientation. Mais, on peut le rendre redondant si la tâche dans l'espace n'exige pas les six degrés de liberté où si le robot travail dans le plan sans tenir compte des contraintes.

La morphologie de ce manipulateur ressemble beaucoup à celle d'un bras humain, il fait partie des manipulateurs dits anthropomorphiques.

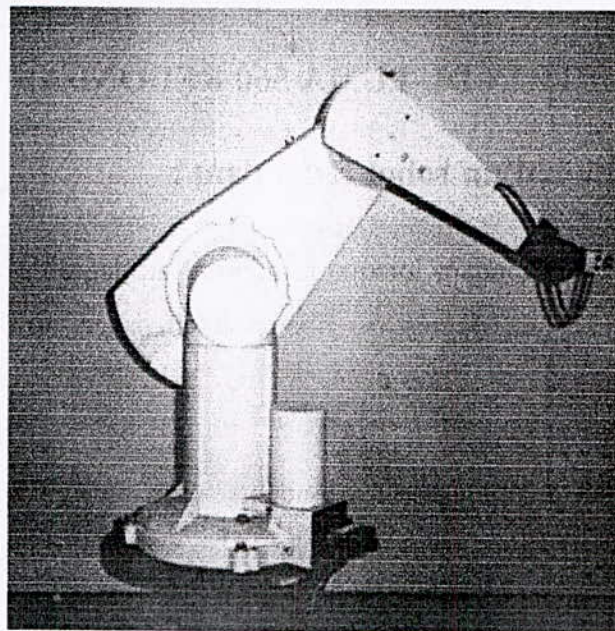


Figure 1.1. PUMA 560

## 1.2. MODELISATION DU PUMA 560 :

Dans l'automatique, modéliser un système consiste à établir un ensemble de relations mathématiques qui permettent de décrire avec une précision suffisante les interactions entre ce système et son environnement extérieur. Lorsque les relations sous citées sont issues de la physique, le modèle obtenu est dit modèle de connaissance, ces relations découlent des

observations disponibles sur le système. On aboutit aussi au modèle de représentation, en passant par l'identification.

L'utilisation de la paramétrisation de Denavit-Hartenberg (**D-H**) facilite la description géométrique du manipulateur, cette dernière nous permet d'aboutir au modèle cinématique et géométrique direct et inverse du robot. La même transformation offre une souplesse dans le calcul du modèle dynamique direct en utilisant le formalisme d'Euler-Lagrange.

Plusieurs modèles sont utilisés pour la commande des robots. Parmi ceux-ci, les plus utilisés sont :

- Modèle géométrique direct et inverse.
- Modèle cinématique direct et inverse.
- Modèle dynamique

### 1.2.1. Modèle géométrique direct (MGD) :

Le modèle géométrique directe permet d'exprimer la position et l'orientation de l'effecteur terminal relativement à un repère fixe (par exemple celui de la base,  $R_0$ ), en fonction des variables articulaires  $q_1, q_2, \dots, q_n$  du mécanisme, sans considération des forces produisent le mouvement.

$$X = \begin{pmatrix} P(q) \\ R(q) \end{pmatrix} = f(q) \quad (1.1)$$

Où  $p(q) \in \mathfrak{R}^3$  représente la position cartésienne de l'effecteur terminal, et  $R(q) \in SO(3)$  représente la rotation du repère de l'effecteur relativement à celui de la base, dont les éléments sont les cosinus directeurs.

### 1.2.2. Algorithme de calcul numérique du modèle géométrique inverse :

Lorsqu'il n'est pas possible de trouver une forme explicite au modèle géométrique inverse, on peut utiliser le modèle cinématique pour calculer itérativement une solution locale numérique. L'algorithme se résume dans les étapes suivantes [KHA 99] :

**Etape1 :**

A partir d'une configuration initiale  $q^c$  (aléatoire ou courante), calculer la situation correspondante  ${}^0T_n^c$  du repère outil en utilisant MGD.

**Etape2 :**

Calculer l'écart  $dX = [dX_p^T \ dX_r^T]^T$  entre la situation désirée  ${}^0T_n^d$  et la situation initiale  ${}^0T_n^c$  tel que  $dX_p = P_n^d - P_n^c$  et  $dX_r = u\alpha$ ,  $u$  et  $\alpha$  désignant l'axe et l'angle correspondant à la rotation  ${}^0A_n^d = \text{rot}(u, \alpha) \times {}^0A_n^c$ . Pour tester dans le domaine de validité du modèle cinématique, qui représente un développement du premier ordre, on doit introduire à chaque pas de calcul des seuils  $S_p$  et  $S_r$  respectivement sur  $dX_p$  et  $dX_r$ , de sorte que :

$$\text{Si } \|dX_p\| > S_p, \text{ alors } dX_p = \frac{dX_p}{\|dX_p\|} \times S_p$$

$$\text{Si } \|dX_r\| > S_r, \text{ alors } dX_r = \frac{dX_r}{\|dX_r\|} \times S_r$$

Des valeurs de 0.3 mètre et 0.3 radian pour ces seuils sont acceptables pour la plupart des robots industriels compte tenu de leurs dimensions. Lorsque  $dX$  est suffisamment petit, arrêter le calcul.

**Etape3 :**

Calculer numériquement la matrice jacobienne directe  ${}^0J_n(q^c)$  et sa pseudo inverse  $J^+$ .

**Etape4 :**

Calculer la variation articulaire correspondante  $dq = J^+ dX$  avec ou sans terme d'optimisation.

**Etape5 :**

Mettre à jour la configuration articulaire courante :  $q^c = q^c + dq$

**Etape6 :**

Retourner à la première étape.

**Remarque importante :**

Cet algorithme est rapide et peut être calculé en temps réel.

**1.2.3. Modèle cinématique direct :**

Le modèle cinématique directe d'un robot manipulateur décrit les vitesses des coordonnées opérationnelles en fonction des vitesses articulaires [KHA 99]. Il est noté :

$$\dot{X} = J(q) \cdot \dot{q} \quad (1.2)$$

Où  $J(q)$  désigne la matrice jacobienne de dimension  $(m \times n)$  du mécanisme, égale à  $\frac{\partial X}{\partial q}$  en fonction de la configuration articulaire  $q$ . La même matrice jacobienne intervient dans le calcul du modèle différentielle directe qui donne les variations élémentaires  $dX$  des coordonnées opérationnelles en fonction des variations élémentaires des coordonnées articulaires  $dq$ , soit :

$$dX = J(q) \cdot dq \quad (1.3)$$

L'intérêt de la matrice jacobienne est multiple :

- Elle est la base du modèle différentielle inverse, permettant de calculer une solution locale des variables articulaires  $q$  connaissant les coordonnées opérationnelles  $X$ .
- En statique en utilisant le jacobien pour établir la relation liant les efforts exercés par l'organe terminal sur l'environnement aux forces et couples des actionneurs.
- Elle facilite le calcul des singularités et de la dimension de l'espace opérationnel accessible du robot.

Le calcul de la matrice jacobienne peut se fait soit :

- Par différentiation du modèle géométrique direct.
- Par un algorithme récursif.
- Par différentiation numérique.

Dans ce qui suit, on développe le modèle cinématique direct pour le PUMA 560, en utilisant la 2<sup>ème</sup> méthode, car la première méthode est plus lourde en calcul et que la dernière ne permet pas d'obtenir une forme analytique.

Le calcul de la matrice jacobienne par la deuxième méthode est donné par [KHA 99] :

$${}^0J_6 = \begin{bmatrix} \bar{\sigma}_1 \cdot (a_1 \times L_{1,n}) \cdots & \bar{\sigma}_n \cdot (a_n \times L_{n,n}) \\ \bar{\sigma}_1 \cdot a_1 & \bar{\sigma}_n \cdot a_n \end{bmatrix} \quad (1.4)$$

Si on projette les éléments de la relation (1.4) dans un repère  $R_i$ , on obtient le jacobien  ${}^iJ_n$  de dimension  $(6 \times n)$ . En effet, on remarque bien que le produit vectoriel  $a_k \times L_{n,k}$  peut se transformer en  $\hat{a}_k \cdot L_{n,k}$ . La  $k^{\text{ième}}$  colonne de  $J_{n,i}$  noté  $J_{n,i}(k)$  et de façon analogue la  $k^{\text{ième}}$  colonne de  ${}^0J_n$  devient :

$${}^0J(k) = \begin{bmatrix} \bar{\sigma}_k \cdot {}^0\hat{a}_k \cdot ({}^0P_n - {}^0P_k) \\ \bar{\sigma}_k \cdot {}^0a_k \end{bmatrix} \quad (1.5)$$

Où :

$\sigma_k = 1$  Si l'articulation est prismatique.

$\sigma_k = 0$  Si l'articulation est rotoïde.

$a_k$  : Le vecteur unitaire porté par l'axe  $Z_k$  de l'articulation  $k$ .

La procédure de calcul de la matrice jacobienne est :

- En calcul les matrices  ${}^0T_k$  pour  $k = 1 \dots n$
- En suite on doit calculer les éléments de la  $k$  colonne qui s'obtiennent à partir de la matrice  ${}^0T_k$  et du vecteur  ${}^0P_n$ .

Soit :  $\dot{q}_d = [\dot{q}_1 \quad \dot{q}_2 \quad \dot{q}_3 \quad \dot{q}_4 \quad \dot{q}_5 \quad \dot{q}_6]_d$ , le vecteur des vitesses articulaires désirées.

$$\text{Et } \begin{bmatrix} \dot{P}_x \\ \dot{P}_y \\ \dot{P}_z \end{bmatrix}_d \Rightarrow \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix}_d : \text{ le vecteur des vitesses cartésiennes.}$$



Et pour les vitesses angulaires  $\Omega_x, \Omega_y, \Omega_z$  on fait une transformation à l'aide de la paramétrisation d'Euler pour trouver les vitesses réelles :

$$\begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix}_d = \begin{bmatrix} 0 & -s_\alpha & c_\alpha s_\beta \\ 0 & c_\alpha & s_\alpha s_\beta \\ 1 & 0 & c_\beta \end{bmatrix} \cdot \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix}_d \quad (1.6)$$

Où :  $\alpha, \beta, \gamma$  sont les angles d'Euler qui définissent l'orientation selon la convention  $ZYZ$ , finalement on peut écrire le modèle cinématique sous la forme suivante :

$$\begin{bmatrix} V_x & V_y & V_z & \Omega_x & \Omega_y & \Omega_z \end{bmatrix}^T = J \cdot \begin{bmatrix} \dot{q}_1 & \dot{q}_2 & \dot{q}_3 & \dot{q}_4 & \dot{q}_5 & \dot{q}_6 \end{bmatrix}^T \quad (1.7)$$

### 1.3. PRESENTATION DES TACHES REDONDANTES :

Dans la plupart des contrôleurs de robot actuels, la géométrie des trajectoires est décrite par une succession de repères. Suivre une trajectoire consiste alors à amener en coïncidence sur ces repères un référentiel lié à l'organe terminal.

Lorsque le nombre de composantes utilisées pour spécifier la tâche est inférieur au nombre de degrés de liberté du robot, il y a redondance de celui-ci vis-à-vis de la tâche et donc, une infinité de solutions pour la réaliser. Cette description, minimale en ce sens qu'elle ne contraint que les degrés de liberté de la tâche ayant un rôle fonctionnel, est intéressante car elle permet de satisfaire des critères d'optimisations supplémentaires lors de son exécution.

#### 1.3.1 Génération de trajectoire dans le plan:

Le problème de la génération de mouvement concerne le calcul des consignes de référence en position, vitesse et accélération qui sont fonction du temps et qui assurent le passage du robot par une trajectoire désirée, définie par une suite de situations de l'organe terminal ou de configurations articulaires. L'outil terminal parcourt le cercle dans le sens des aiguilles d'une montre comme indiqué sur la figure (1.2). La position de l'effecteur sur le cercle est donnée par l'angle  $\alpha$ .

Si on adopte une représentation polaire pour situer l'effecteur terminal, celui-ci aura pour coordonnées dans le repère de base  $\{X_0, Y_0, Z_0\}$  :

$$\begin{cases} X_E = O'_X + R \cdot \cos(\alpha) \\ Z_E = O'_Z + R \cdot \sin(\alpha) \\ Y_E = d_2 + d_3 \end{cases} \quad (1.8)$$

Où :

$R$  : le rayon de cercle.

$O'_X, O'_Z$  : représente les coordonnées du centre de cercle dans le repère  $\{Z_0, X_0\}$

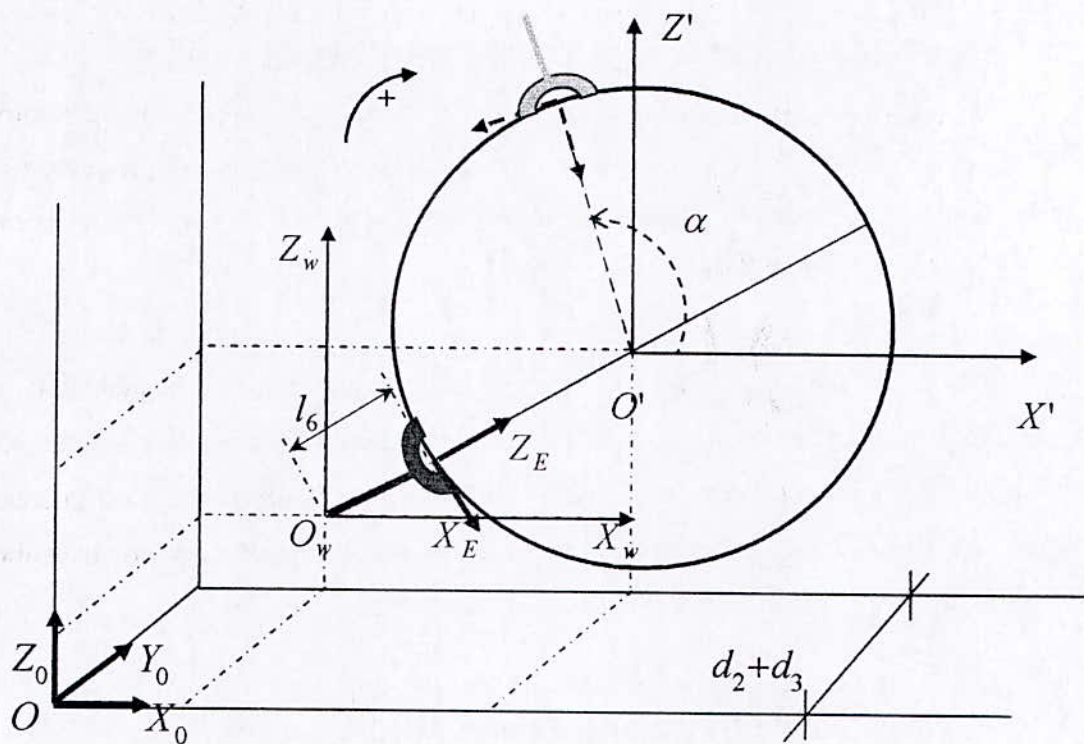


Figure 1.2. La tâche désirée dans le plan.

L'évolution de  $\alpha(t)$  est imposé tel qu'il suit la loi Bang-Bang [KHA 99] :

Avec  $\alpha \in [\alpha_0, \alpha_0 + 2 \cdot \pi]$

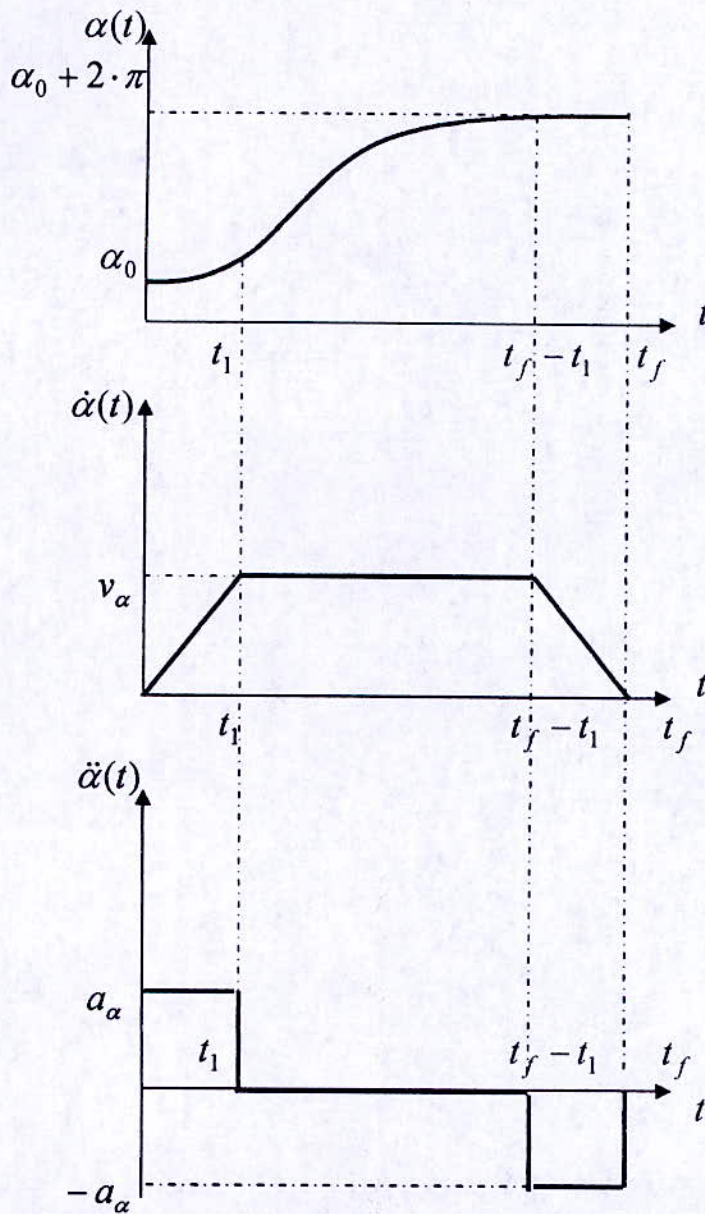


Figure 1.3. Variation de  $\alpha$ ,  $\dot{\alpha}$  et  $\ddot{\alpha}$

D'après la figure (2.2) on peut écrire :

- Pour le premier intervalle de temps  $0 \leq t \leq t_1$  :

$$\begin{cases} \alpha(t) = \alpha_0 + \frac{1}{2} \cdot a_\alpha \cdot t^2 \\ \dot{\alpha}(t) = a_\alpha \cdot t \\ \ddot{\alpha}(t) = a_\alpha \end{cases} \quad (1.9)$$

- Pour le deuxième intervalle de temps  $t_1 \leq t \leq t_f - t_1$  :

$$\begin{cases} \alpha(t) = \alpha_0 + v_\alpha \cdot (t - \frac{t_1}{2}) \\ \dot{\alpha}(t) = v_\alpha \\ \ddot{\alpha}(t) = 0 \end{cases} \quad (1.10)$$

- Pour le troisième intervalle de temps  $t_f - t_1 \leq t \leq t_f$  :

$$\begin{cases} \alpha(t) = \alpha_f - \frac{1}{2} \cdot a_\alpha \cdot (t - t_f)^2 \\ \dot{\alpha}(t) = -a_\alpha \cdot (t - t_f) \\ \ddot{\alpha}(t) = -a_\alpha \end{cases} \quad (1.11)$$

Avec :

$$\alpha_0 = \frac{\pi}{4}, \quad \alpha_f = \alpha_0 + 2 \cdot \pi, \quad v_\alpha = \frac{\alpha_f - \alpha_0}{t_f}, \quad a_\alpha = \frac{v_\alpha}{t_1}.$$

La position, la vitesse et l'accélération de l'effecteur sont exprimées pour les équations suivantes :

↓ **Les positions :**

$$\begin{cases} X_E = O'_x + R \cdot \cos(\alpha) \\ Y_E = d_2 + d_3 \\ Z_E = O'_z + R \cdot \sin(\alpha) \end{cases} \quad (1.12)$$

↓ Les vitesses :

$$\begin{cases} \dot{X}_E = -R \cdot \sin(\alpha) \cdot \dot{\alpha} \\ \dot{Y}_E = 0 \\ \dot{Z}_E = R \cdot \cos(\alpha) \cdot \dot{\alpha} \end{cases} \quad (1.13)$$

↓ Les accélérations :

$$\begin{cases} \ddot{X}_E = -R \cdot \cos(\alpha) \cdot \dot{\alpha}^2 - R \cdot \sin(\alpha) \cdot \ddot{\alpha} \\ \ddot{Y}_E = 0 \\ \ddot{Z}_E = -R \cdot \sin(\alpha) \cdot \dot{\alpha}^2 + R \cdot \cos(\alpha) \cdot \ddot{\alpha} \end{cases} \quad (1.14)$$

### 1.3.2 Génération de trajectoire dans l'espace 3D :

On veut que l'outil terminal suive la trajectoire décrite à la figure 1.4 (glissement sur la surface d'un cylindre de rayon  $R$  et de centre  $O'(x'_0, y'_0, z'_0)$  donné). L'orientation est calculée pour assurer la direction radiale entrante à l'outil afin qu'il puisse réaliser la tâche demandée, sachant que nous avons adopté la représentation d'Euler concernant les orientations.

L'angle  $\psi$  sert à déterminer la position et l'orientation de l'outil terminal dans le repère lié au cylindre. Et par une simple translation de vecteur  $[x'_0, y'_0, z'_0]$ , on retrouve les coordonnées cartésiennes de l'outil dans le repère de la base.

La figure 1.4 représente la trajectoire désirée dans l'espace 3D

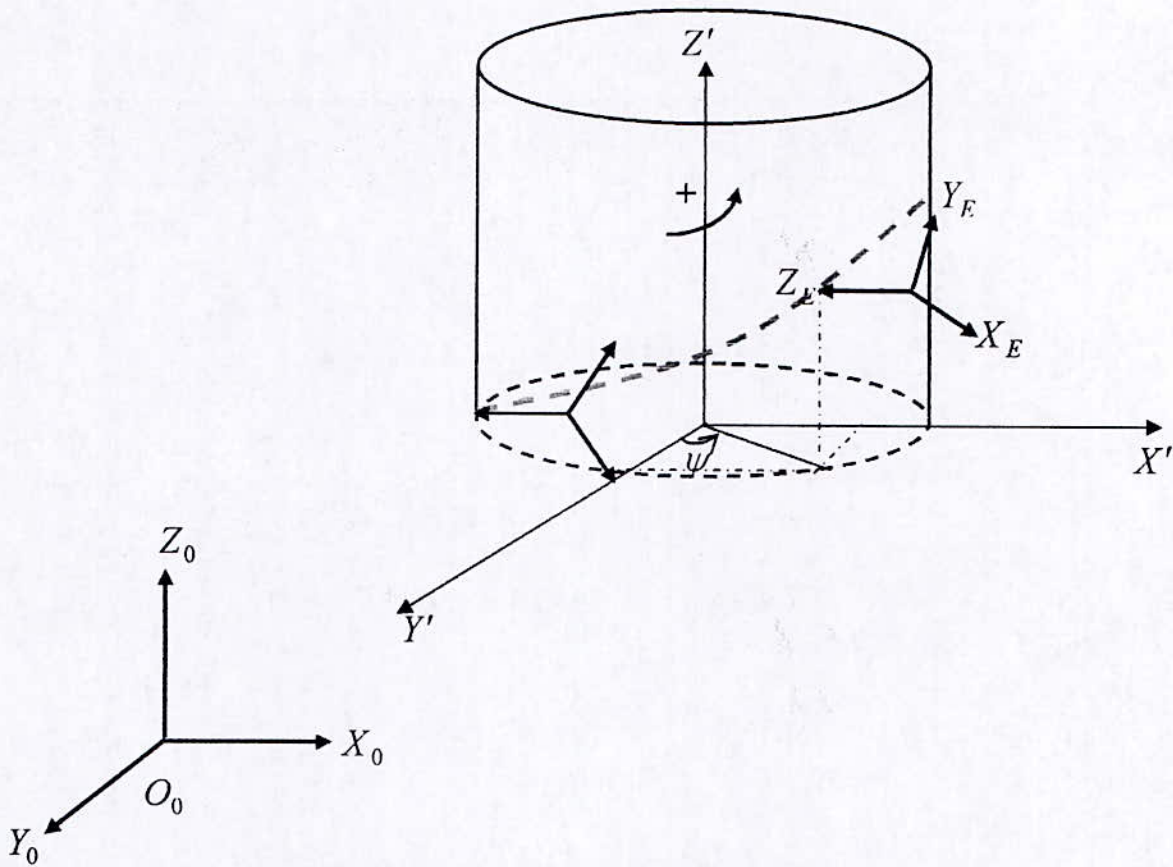


Figure 1.4. La tâche désirée dans l'espace 3D

En fonction des paramètres qui sont l'angle  $\psi$  et la hauteur  $h$  du cylindre qui sont générés par une loi Bang –Bang avec palier de vitesse, on peut trouver les vecteurs positions et orientations comme suit :

↓ Les positions :

$$\begin{cases} x = R \cdot \cos(\psi) + x_0 \\ y = R \cdot \sin(\psi) + y_0 \\ z = h \\ \alpha = \psi \\ \beta = -\frac{\pi}{2} \end{cases} \quad (1.15)$$

Avec :

$$\begin{cases} \psi \in [0, \frac{5\pi}{4}] \\ h \in [0, R] \end{cases}$$

Où :  $x, y, z$  représente la position de l'outil et  $\alpha, \beta$  sont les angles d'Euler correspondantes à l'orientation de l'outil terminal exprimées dans le repère de la base

↓ Les vitesses :

$$\begin{cases} \dot{x} = -R \cdot \sin(\psi) \cdot \dot{\psi} \\ \dot{y} = R \cdot \cos(\psi) \cdot \dot{\psi} \\ \dot{z} = \dot{h} \\ \dot{\alpha} = \dot{\psi} \\ \dot{\beta} = 0 \end{cases} \quad (1.16)$$

↓ Les accélérations :

$$\begin{cases} \ddot{x} = -R \cdot \ddot{\psi} \cdot \sin(\psi) - R \cdot \dot{\psi}^2 \cdot \cos(\psi) \\ \ddot{y} = -R \cdot \dot{\psi}^2 \cdot \sin(\psi) + R \cdot \ddot{\psi} \cdot \cos(\psi) \\ \ddot{z} = \ddot{h} \\ \ddot{\alpha} = \ddot{\psi} \\ \ddot{\beta} = 0 \end{cases} \quad (1.17)$$

**Remarque :** Dans le cas de la génération de la trajectoire dans l'espace 3D, on ne s'intéresse pas à l'angle d'Euler  $\gamma$  dans le but de garder la redondance du PUMA 560, c'est-à-dire la tâche désirée n'exige que 5 degrés de libertés donc le degrés de la redondance est égale à  $n - m = 1$ .

## 1.4. STRUCTURE DE COMMANDE:

### 1.4.1. Introduction :

Généralement, la spécification des tâches que le robot est amené à exécuter (mouvements ou forces de l'effecteur terminal) se fait dans l'espace opérationnel, tandis que les actions de commande (couples des actionneurs) sont entreprises dans l'espace articulaire. Il s'ensuit la classification des structures de commandes en deux catégories [KHA 99] :

- Structure de commande dans l'espace articulaire.
- Structure de commande dans l'espace cartésien.

Cette classification s'applique tout autant aux architectures de commande en position qu'à celle de commande en force.

Dans notre simulation on s'intéresse à la structure de commande par découplage non linéaire avec correction dans l'espace articulaire dont le but d'exploiter des modèles géométrique inverse et cinématique inverse.

### 1.4.2. Commande par découplage non linéaire dans l'espace articulaire :

Une structure de commande dans l'espace articulaire, utilise un régulateur qui permet la poursuite de références données en termes de variables articulaires. Ces derniers sont obtenus à partir de références prescrites dans l'espace opérationnel, en passant par la géométrie inverse et éventuellement la cinématique inverse du robot, qui sont des fonctions assez complexes et peuvent faire l'objet de singularités. Le robot constitue dans ce cas un système dont les entrées (resp. sorties) sont les couples (resp. variables) articulaires.

Soit  $\hat{A}$  et  $\hat{H}$  les estimations de  $A$  et  $H$ . On suppose que les positions et vitesses articulaires sont mesurables et que les mesures ne sont pas bruitées partant de l'équation du modèle dynamique suivante :

$$\Gamma = \hat{A}(q) \cdot w(t) + \hat{H}(q, \dot{q}) \quad (1.18)$$

$$\text{Où : } \ddot{q} = w(t) \quad (1.19)$$

$$\hat{H}(q, \dot{q}) = \hat{B} + \hat{G} + \hat{C} + \hat{F} \quad (1.20)$$



$w(t)$  : considère comme un nouveau vecteur de commande. On se ramène donc à un problème de commande de  $n$  systèmes linéaires, invariants, découplés. Plusieurs choix peuvent être envisagés pour  $w(t)$ .

En prend :

$$w(t) = \ddot{q}_d + K_v \cdot (\dot{q}_d - \dot{q}) + K_p \cdot (q_d - q) \quad (1.21)$$

Où :  $\ddot{q}_d(t), \dot{q}_d(t), q_d(t)$  sont respectivement l'accélération, la vitesse et la position désirées dans l'espace articulaires.

$K_p$  Et  $K_v$  sont des matrices diagonales définies positives de dimension  $(n \times n)$

Alors, d'après l'équation (1.22), la réponse du système en boucle fermée est décrite par l'équation linéaire découplée suivante :

$$\ddot{e} + K_v \cdot \dot{e} + K_p \cdot e = 0 \quad (1.22)$$

Où :  $e = q_d - q$

Un choix évident pour choisir les matrices des gains est le suivant :

$$\begin{aligned} K_{p_j} &= w_j^2 \\ K_{v_j} &= 2 \cdot \xi_j \cdot w_j \end{aligned} \quad (1.23)$$

Où  $\xi_j$  est l'amortissement et  $w_j$  est la pulsation.

Donc l'équation (1.21) devienne :

$$\Gamma = \hat{A}(q) \cdot (\ddot{q}_d + K_v \cdot (\dot{q}_d - \dot{q}) + K_p \cdot (q_d - q)) + \hat{H}(q, \dot{q}) \quad (1.24)$$

Si on suppose que les matrices  $A$  et  $H$  connues, l'équation précédente devient :

$$\Gamma = A(q) \cdot (\ddot{q}_d + K_v \cdot (\dot{q}_d - \dot{q}) + K_p \cdot (q_d - q)) + H(q, \dot{q}) \quad (1.25)$$

Le principe de cette commande est représenté dans le schéma bloc suivant :

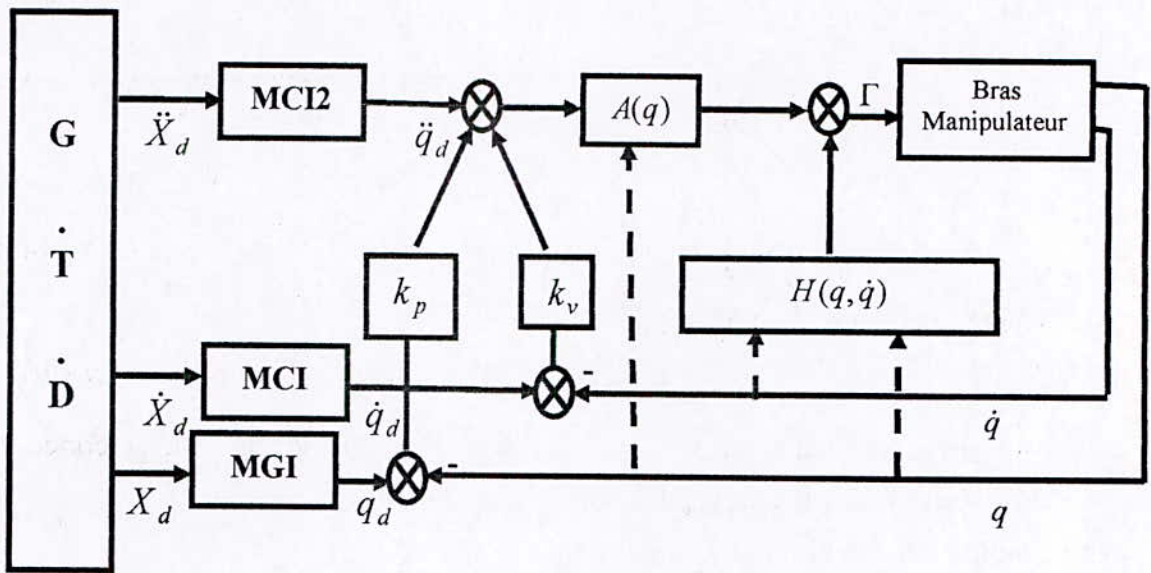


Figure .1.5. Commande par découplage non linéaire avec correction dans l'espace articulaire.

## 1.5. CONCLUSION :

Ce chapitre a permis de présenter le robot PUMA 560 redondant et la génération de trajectoire dans le plan et dans l'espace pour ce robot. Nous avons exposé la commande par découplage non linéaire dans l'espace articulaire, cette dernière pose le problème de la cinématique inverse dont la résolution fait l'objet des deux chapitres suivants.



# Résolution du problème de la cinématique inverse basée sur la pseudo inverse

Un robot manipulateur est redondant lorsqu'il possède plus de degré de liberté  $n$  que la dimension de l'espace opérationnel de l'organe terminal  $m$ . Il existe donc une infinité de solutions articulaires pour réaliser une tâche donnée. La différence  $(n-m)$  représente le degré de la redondance. Le modèle géométrique inverse et le modèle cinématique inverse ont dans ce cas une infinité de solutions, d'où la possibilité de choisir la solution qui satisfait des contraintes d'optimisation supplémentaires telles que :

- Contournement d'obstacles [MAC 85], [BAI 86].
- Evitement des configurations singulières [YOS 84].
- Eloignement des butées articulaires [FOU 80], [KLE 84].
- Répartition des efforts aux articulations [BAI 84], [HOL 85].

Dans ce chapitre on propose des méthodes basées sur le calcul directe de la pseudo inverse ainsi les méthodes basées sur la décomposition de la matrice jacobienne, et l'utilisation de la logique floue pour résoudre le problème de la MCI.

## 2.1. INTRODUCTION :

Suivant le choix qui est fait pour définir le vecteur  $X$  des variables opérationnelles, le modèle cinématique direct peut s'écrire de plusieurs manières :

$$\delta X = J(q) \cdot \delta q \quad \text{Ou} \quad \dot{X} = J(q) \cdot \dot{q} \quad (2.1)$$

$$X \in R^n, J \in R^{n \times m}, q \in R^m$$

Si on prend systématiquement pour vecteur  $\dot{X}$  le vecteur vitesse de rotation du dernier repère du robot et le vecteur vitesse de translation du point extrémité du robot. La formule générale (2.1) englobe toutes les représentations possibles et le problème de la commande cinématique utilisant ce modèle repose sur les possibilités d'inversion de ce jacobien  $J(q)$ .

$$\dot{q} = [J(q)]^{-1} \cdot \dot{X} \quad (2.2)$$

Remarquons préliminairement que l'équation (2.2), à supposer qu'elle soit maîtrisée, ne donne que des accroissements. Il faut donc s'avoir d'où l'on part. Cela est obtenu soit en démarrant toujours d'une configuration connue dite d'initialisation pour la quelle  $X_0$  a été mesuré ou calculé par un moyen quelconque, soit en utilisant en complément du modèle cinématique, quand cela s'avère nécessaire, le modèle géométrique direct qui donne  $X_0$ , en mesurant les valeurs des variables articulaires.

Les modes d'obtention de  $J^{-1}$  et la résolution du problème de la cinématique inverse constitue l'essentiel des préoccupations de ce chapitre.

## 2.2. DIMENSION DE J :

C'est une matrice à  $m$  lignes correspondant au nombre de degré de liberté du robot et à  $n$  colonnes correspondant au nombre de composantes du vecteur  $X$ . Si, pour un robot donné, le nombre de degré de liberté peut être considéré comme fixe, il n'en est pas de même de  $X$  qui peut varier :

- 1) avec la tâche à exécuter.
- 2) avec les choix de calcul faits pour élaborer le modèle.

Quoi qu'il en soit, on peut déjà prendre conscience de deux problèmes :

- i) avec  $m$  degré de liberté, on ne pourra pas commander plus de  $m$  composantes indépendantes du vecteur  $\dot{X}$  (ou  $\delta X$ ). Ceci signifie que l'obtention de  $J^{-1}$  est soumise à une première condition :

$$n \leq m \quad (2.3)$$

- ii) la matrice  $J$  sera rarement carrée. Pour l'inversée il faut avoir un déterminant non nul.

Examinons donc les différents cas de figure possibles pour l'inversion de la matrice jacobienne ainsi la résolution du problème de la cinématique inverse.

## 2.3. LES METHODES BASEES SUR LE CALCUL DE LA PSEUDO INVERSE :

### 2.3.1. Utilisation de la pseudo inverse sans terme d'optimisation :

#### 2.3.1.1. Utilisation de la pseudo inverse :

La pseudo inverse  $J^+$  de la matrice jacobienne  $J$  est la méthode la plus largement adoptée pour résoudre le problème de la cinématique inverse :

$$\dot{q} = J^+ \cdot \dot{X} \quad (2.4)$$

Avec  $J^+ = J^T \cdot (J \cdot J^T)^{-1}$  (2.5)

Ou bien  $J^+ = (J^T \cdot J)^{-1} \cdot J^T$  (2.6)

Pour un système qui a plus d'inconnues que d'équations on va considérer  $J^+$  sous la forme donnée par (2.5).

Pour un système qui a plus d'équations que d'inconnues on va considéré  $J^+$  sous la forme donnée par (2.6).

Cette solution, proposée par Whitney [WHI 69], [WHI 72], minimise la norme euclidienne  $\|\dot{q}\|^2$  et la norme de l'erreur  $\|\dot{X} - J \cdot \dot{q}\|^2$ . Dans une configuration singulière, on distingue les deux cas particuliers :

1)  $\dot{X}$  appartient uniquement à  $I(J)$ . Alors la solution (2.4) est exacte et l'erreur est nulle bien que l'inverse  $J^{-1}$  ne soit pas défini.

2)  $\dot{X}$  appartient uniquement à  $I(J)^\perp$ . Alors la solution (2.4) donne  $\dot{q} = 0$ . Si la consigne de vitesse suivante est définie selon cette direction, le robot se bloque et il faut définir des stratégies de déblocage [CHE 88].

On note :

$I(J)$  : représente l'ensemble de toutes les vitesses de l'organe terminale qui peuvent être engendrées par le robot.

$I(J)^\perp$  : définit l'ensemble le complément de l'espace  $I(J)$ .

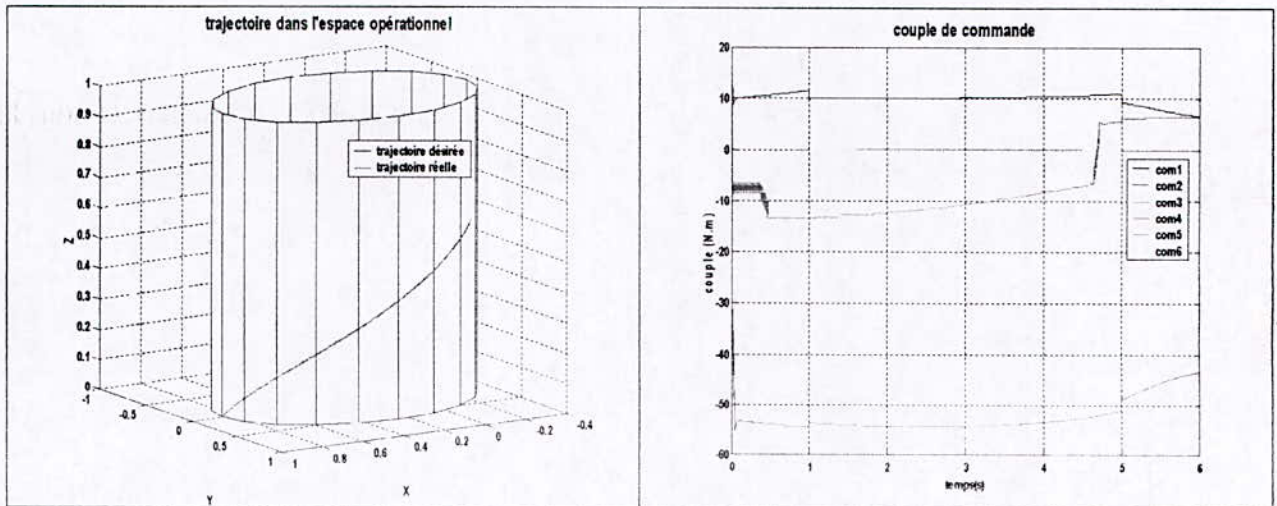
Dans [WAM 86], on montre que la pseudo inverse donne une solution discontinue lors du passage par une configuration singulière, ce qui peut être mise en évidence par la décomposition en valeurs singulières (SVD). En effet, en dehors des singularités :

$$\dot{q} = \sum_{i=1}^m \frac{1}{\sigma_i} \cdot V_i \cdot U_i^T \cdot \dot{X} \tag{2.7}$$

En s'approchant d'une singularité,  $\sigma_{\min}$  devient petit ce qui conduit à une vitesse  $\dot{q}$  élevée. Lorsque la valeur singulière  $\sigma_{\min}$  s'annule, elle n'est plus prise en compte, la somme s'arrêtant à  $(m - 1)$ , et la vitesse  $\dot{q}$  peut chuter considérablement.

### 2.3.1.1.1. Résultats et Simulation :

- Dans l'espace :



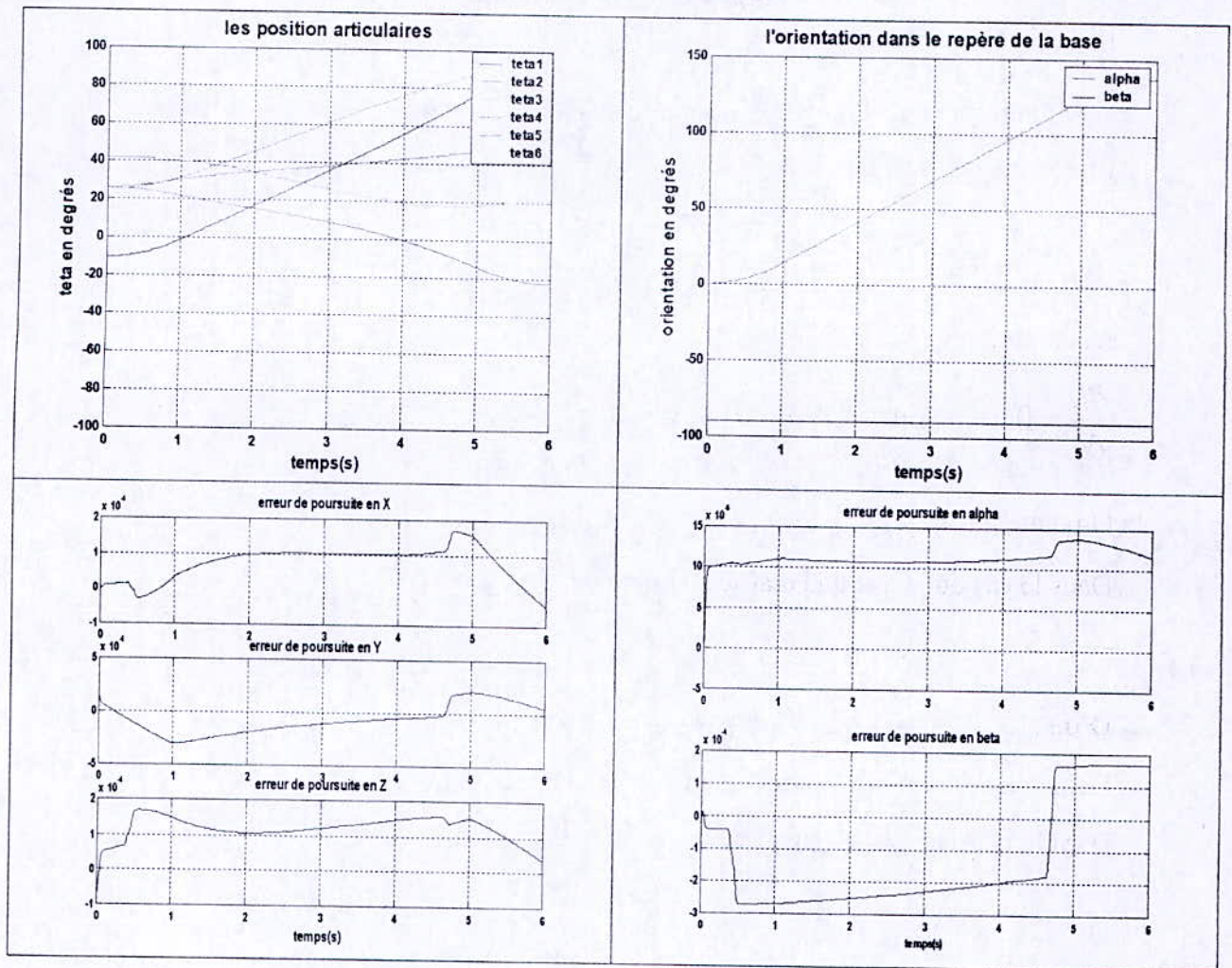


Figure 2.1. Résultats de simulation en utilisant la méthode de la pseudo inverse sans terme d'optimisation

- la pseudo inverse sans terme d'optimisation donne une bonne poursuite de trajectoire dans l'espace cartésienne, ce qui est confirmé par les erreurs qui sont de l'ordre  $10^{-5}$  et on constate un couple qui est admissible.

### 2.3.1.2. Minimisation d'un critère quadratique :

Comme le système linéaire  $J \cdot \dot{q} = \dot{X}$  possède une infinité de solutions, il est intéressant de pouvoir en choisir une solution en fonction d'un certain critère. Il peut être intéressant de minimiser un critère quadratique qui s'exprime dans le cas général de la façon suivante :

$$C_o = \frac{1}{2} \cdot \dot{q}^T \cdot Q \cdot \dot{q} \quad (2.8)$$

Où  $Q$  est une matrice symétrique définie positive.



Il s'agit donc de résoudre le système :

$$J \cdot \dot{q} = \dot{X}$$

En minimisant la quantité  $C_o$ , le lagrangien de ce système est alors donné par :

$$L = \frac{1}{2} \cdot \dot{q}^T \cdot Q \cdot \dot{q} + \Lambda^T \cdot (\dot{X} - J \cdot \dot{q}) = C_o + \Lambda^T \cdot (\dot{X} - J \cdot \dot{q}) \quad (2.9)$$

La condition de stationnarité de  $L$  s'écrit :

$$\frac{\partial L}{\partial \dot{q}} = 0 \quad \text{Soit} \quad Q \cdot \dot{q} - J^T \cdot \Lambda = 0$$

On obtient :  $\dot{q} = Q^{-1} \cdot J^T \cdot \Lambda$  ce qui implique :  $\dot{X} = J \cdot Q^{-1} \cdot J^T \cdot \Lambda$

Dans le cas où  $J$  est une matrice de rang maximal on a :

$$\Lambda = (J \cdot Q^{-1} \cdot J^T)^{-1} \cdot \dot{X}$$

$$\text{D'où :} \quad \dot{q} = Q^{-1} \cdot J^T \cdot (J \cdot Q^{-1} \cdot J^T)^{-1} \cdot \dot{X} \quad (2.10)$$

Deux cas particuliers sont à considérés :

i) Si la matrice  $Q = I$  (matrice unité d'ordre  $n$ ) on trouve :

$$\dot{q} = J^T \cdot (J \cdot J^T)^{-1} \cdot \dot{X} \quad (2.11)$$

Soit :  $J^+ = J^T \cdot (J \cdot J^T)^{-1} \Rightarrow$  La pseudo inverse.

ii) Si la matrice  $Q = E$  (matrice quelconque définie positive), et  $J$  est de rang maximal, la solution est donnée par :

$$\dot{q} = J_E^+ \dot{X} \quad (2.12)$$

Avec :

$$J_E^+ = E^{-1} \cdot J^T \cdot (J \cdot E^{-1} \cdot J^T)^{-1} \Rightarrow \text{La pseudo inverse pondéré.}$$

Benoit et al, [BEN 75] proposent de prendre pour  $E$  la matrice d'inertie du robot afin de minimiser l'énergie cinématique. Konstantinov et al, [KON 81] ont utilisés la pseudo inverse pondéré pour éviter les butées articulaires.

### 2.3.1.2.1. Résultats et Simulation :

- Dans l'espace :

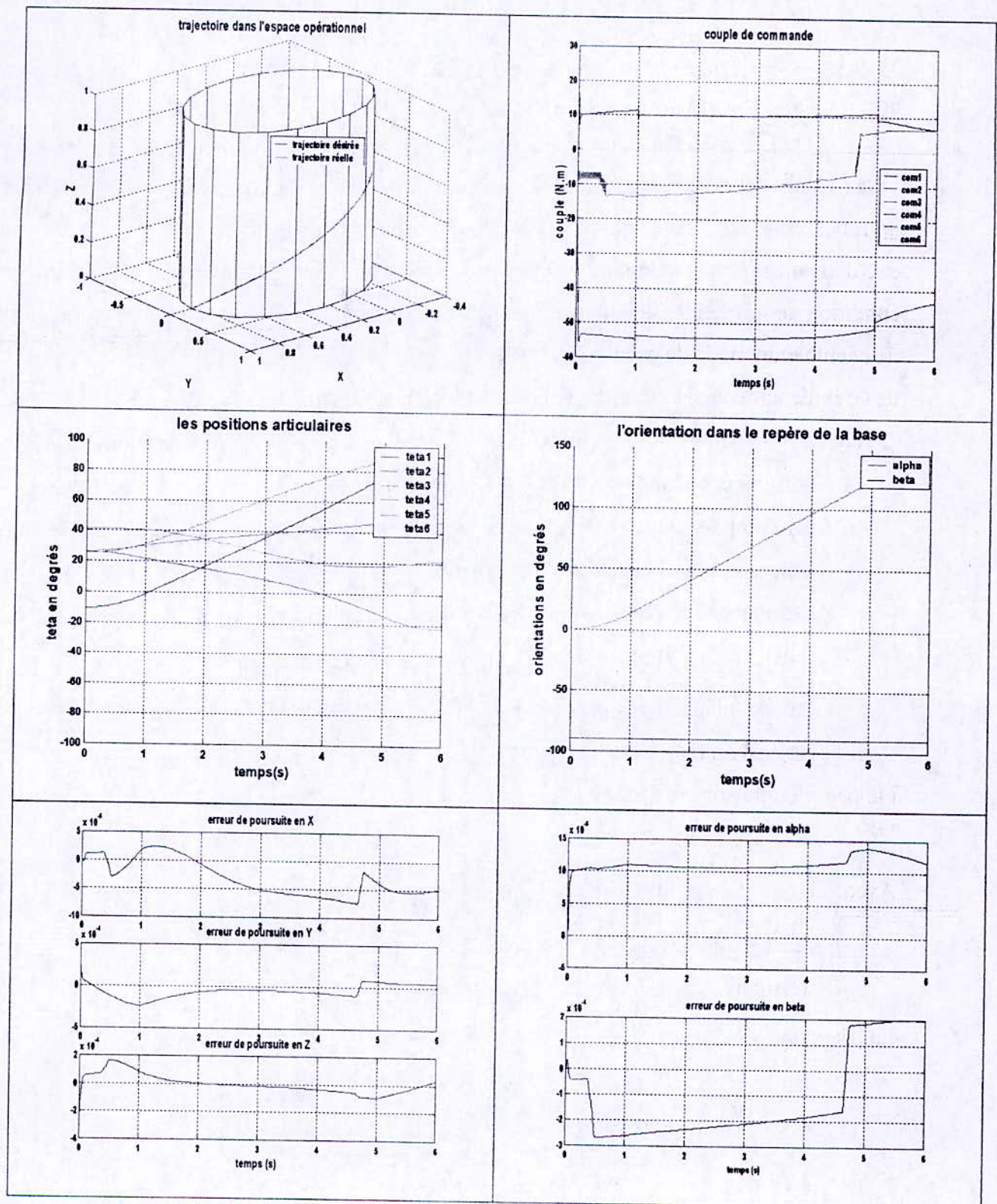


Figure 2.2. Résultats de simulation en utilisant la pseudo inverse pondéré due à la minimisation d'un critère quadratique.

- On conclut qu'en utilisant la solution de la pseudo inverse pondéré où la matrice de pondération est la matrice d'inertie de robot, on obtient des résultats acceptables avec une erreur de poursuite de l'ordre de  $10^{-6}$ .

### 2.3.1.3. Algorithme de Gréville pour calculer la pseudo inverse:

Tout algorithme de calcul de la pseudo inverse, entre dans l'élaboration de la commande du robot, donc fait l'objet de calcul dans un ordinateur, lequel peut se bloquer dès qu'il effectue une division par zéro, si le rang de la matrice à inverser n'est pas celui prévu ou pour toute autre condition de limites, ce qui amène à plus de complication algorithmique. Donc, il est important de chercher un mode de calcul qui permet d'éviter ces singularités algorithmiques et l'algorithme de Gréville entre dans cette catégorie [GRE 80].

De ce point de vue, l'algorithme de Gréville apparaît comme très performant :

- Il est indépendant des dimensions de  $J$  mais aussi de son rang (dimension du plus grand déterminant non nul).
- Il ne comporte aucune division par zéro (pas de méthode « classique » d'inversion de matrice avec division par le déterminant).
- Il détecte automatiquement les singularités (configurations singulières du robot) si on lui fait calculer la trace de  $J^+ J$  (somme des éléments formant la première diagonale) qui est égal au rang de la matrice  $J$ .

On peut l'écrire sous la forme :

$$J \equiv J_k = (J_{k-1}, j_k) \quad (2.13)$$

Avec:

$J_{k-1}$  : Matrice formée avec les  $k - 1$  premières colonnes de  $J(q)$ .

$j_k$  : Dernière colonne de  $J(q)$  considérée comme un vecteur.

La pseudo inverse se calcul par la récurrence :

$$J^+_k = \begin{pmatrix} J^+_{k-1} - d_k \cdot b_k^T \\ b_k^T \end{pmatrix} \quad (2.14)$$

$$d_k = J^+_{k-1} \cdot j_k \quad (2.15)$$

Avec :

$$j^+_1 = 0^T \quad \text{Si } j_1 = 0$$

$$j^+_{i+1} = \frac{j_i^T}{(j_i^T \cdot j_i)} \quad \text{Si } j_i \neq 0$$

Pour  $b_k^T$ , on calcule tout d'abord le terme :

$$c_k = j_k - J_{k-1} \cdot d_k \tag{2.16}$$

Alors si  $c_k^T = [c_1 \dots c_m] \neq 0$

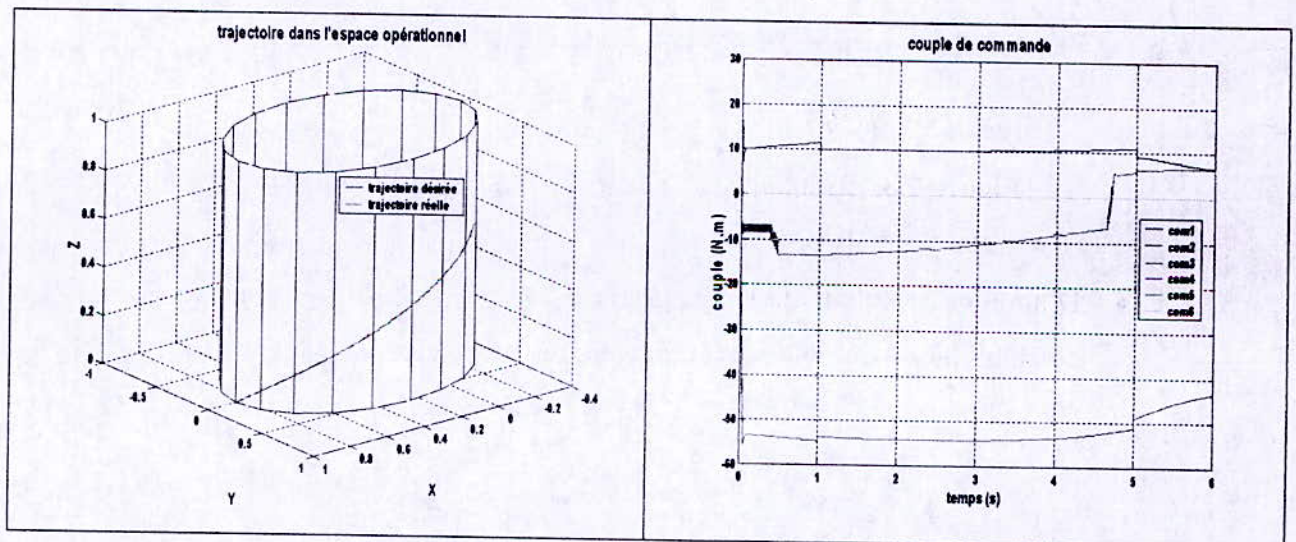
$$b_k^T = c^+_k = \frac{c_k^T}{c_k^T \cdot c_k} \tag{2.17}$$

Sinon  $c_k^T = [c_1 \dots c_m] = 0$

$$b_k^T = \frac{d_k^T \cdot J^+_{k-1}}{(1 + d_k^T \cdot d_k)} \tag{2.18}$$

### 2.3.1.3.1. Résultats et simulation :

- Dans l'espace :



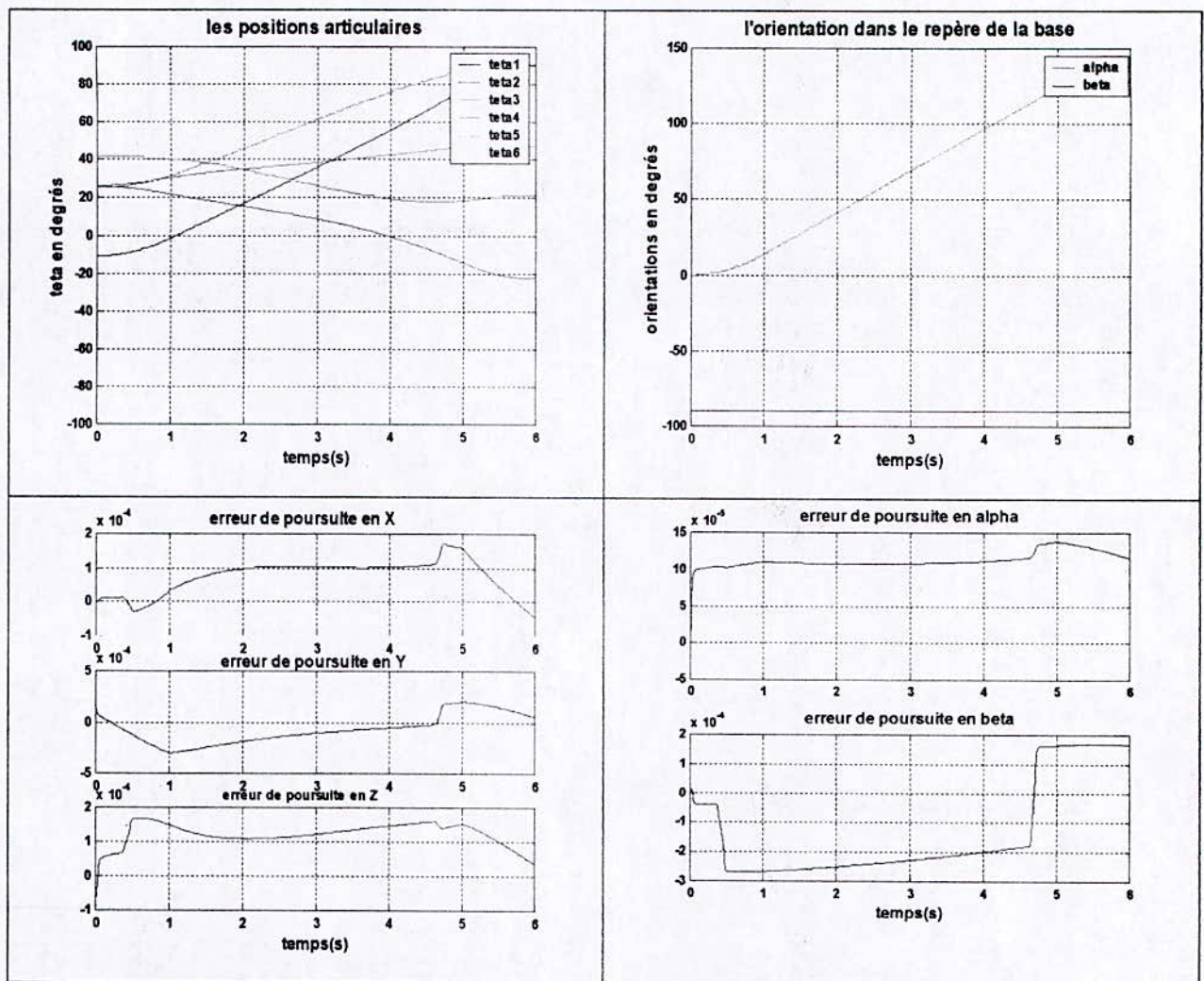


Figure 2.3. Résultats de simulation en utilisant l’algorithme de Gréville.

- D’après les résultats de simulation, on constate qu’on peut résoudre le problème de la cinématique inverse par l’algorithme de Gréville en évitant les singularités algorithmiques, et cela donne les mêmes performances que par la pseudo inverse.

### 2.3.2. Utilisation de la pseudo inverse avec un terme d'optimisation :

La solution générale du système d'équations linéaires  $\dot{X} = J \cdot \dot{q}$  s'écrit [KHA 99] :

$$\dot{q} = J^+ \cdot \dot{X} + (I_n - J^+ \cdot J) \cdot Z \quad (2.19)$$

Où  $J^+$  désigne la pseudo inverse de  $J$  et ou  $Z$  représente un vecteur arbitraire de dimension  $(n \times 1)$ .

Le second terme du membre de droite, appelé solution homogène ou terme d'optimisation, appartient au noyau de  $J$  et n'affecte pas la valeur de  $\dot{X}$ . Il peut être utilisé pour satisfaire des contraintes d'optimisation supplémentaires. Soit  $\Phi(q)$  une fonction scalaire définie positive de l'état  $q$  du mécanisme et soit  $\nabla\Phi$  le gradient de cette fonction en  $q$ . On montre que le choix de  $Z = \alpha \nabla\Phi$  entraîne la décroissance de la fonction  $\Phi(q)$  pour  $\alpha < 0$  et provoque la croissance de cette fonction pour  $\alpha > 0$ . La solution s'écrit alors :

$$\dot{q} = J^+ \cdot \dot{X} + \alpha \cdot (I_n - J^+ \cdot J) \cdot \nabla\Phi \quad (2.20)$$

Avec :

$$\nabla\Phi = \left[ \frac{\partial\Phi}{\partial q_1} \dots \frac{\partial\Phi}{\partial q_n} \right]^T \quad (2.21)$$

Le coefficient  $\alpha$  représente un gain qui permet de trouver un compromis entre les objectifs de minimisation de  $\|\dot{q}\|^2$  et d'optimisation de  $\Phi(q)$ . Nous étudions dans ce qui suit deux choix possibles pour le critère d'optimisation.

**Démonstration :** (Projection sur le noyau de  $J$ )

Nous plaçons dans le cas où le degré de liberté du robot fournit un espace de solutions appelé espace libre. On peut remarquer que :

$$\begin{aligned} \forall Z, \quad J \cdot (J^+ \cdot J - I) \cdot Z &= J \cdot (J^T \cdot (J \cdot J^T) \cdot J - I) \cdot Z \\ &= \left( J \cdot J^T \cdot (J \cdot J^T)^{-1} \cdot J - J \right) \cdot Z \\ &= (J - J) \cdot Z \\ &= 0 \end{aligned}$$

L'opérateur  $(J^+ \cdot J - I)$  projette les vecteurs de coordonnées articulaires sur le noyau de  $J$ , il en résulte que :

$$\forall Z, \quad J \cdot \dot{q} = \dot{X} \Rightarrow J \cdot (\dot{q} + (J^+ \cdot J - I) \cdot Z) = \dot{X}$$

### 2.3.2.1. Utilisation du terme d'optimisation pour l'éloignement des butées articulaires :

Une solution pratique pour commander un robot redondant peut être de faire évoluer les variables articulaires au voisinage de leurs moyennes  $q_{moy}$ . C'est à dire loin de leurs limites  $q_{max}$  et  $q_{min}$ . On pose [KHA 99] :

$$q_{moy} = \frac{1}{2} \cdot (q_{max} + q_{min}) \quad (2.22)$$

$$\Delta q = q_{max} - q_{min} \quad (2.23)$$

Une fonction scalaire possible, dont la décroissance a pour effet l'éloignement du robot de ces butées, peut s'exprimer par la forme quadratique suivante :

$$\Phi(q) = \sum_{i=1}^n \left[ \left( \frac{q_i - q_{i,max}}{\Delta q_i} \right)^2 \right] \quad (2.24)$$

#### Remarque :

a) La division par  $\Delta q_i$  permet de faire varier la contribution de chaque articulation dans  $\Phi$  entre 0 et 1. L'élément  $i$  du vecteur  $Z$  s'écrit (avec  $\alpha < 0$ ).

Si la position moyenne d'une articulation correspond à une configuration singulière. Il convient de remplacer la valeur  $q_{imoy}$  correspondante par une autre valeur.

b) Pour le choix du critère on utilise souvent la forme quadratique pour résoudre les problèmes d'optimisation mais il existe d'autres formes pour l'éloignement de butées en particulier, la forme suivante est la plus appropriée :

$$\Phi = \text{Max} \frac{|q_i - q_{imoy}|}{|\Delta q_i|} \quad \text{Pour } i = 1, \dots, n. \quad (2.25)$$

#### 2.3.2.1.1. Résultats et Simulation :

• Dans l'espace:

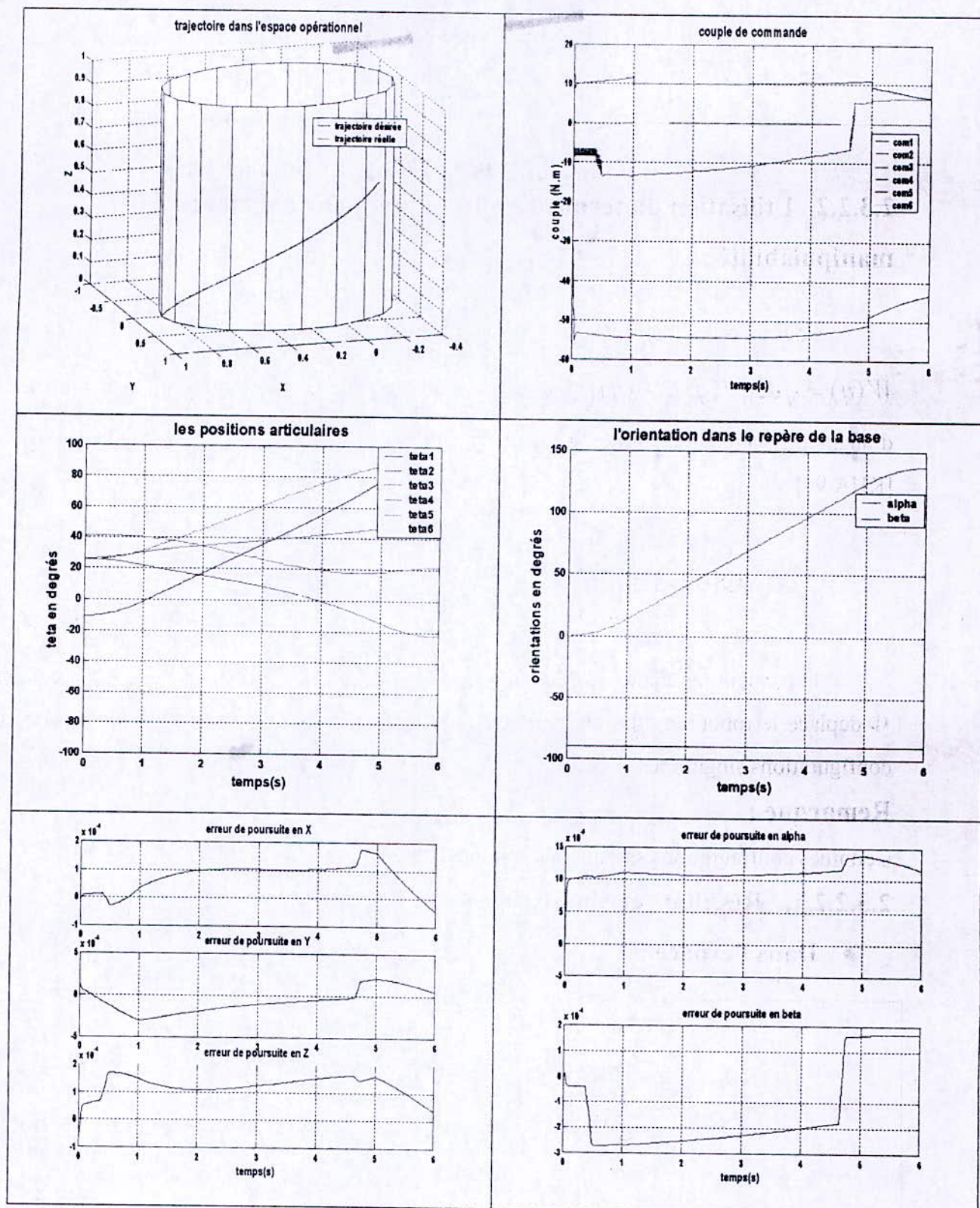


Figure 2.4. Résultats de simulation en utilisant la pseudo inverse avec un terme d'optimisation pour l'éloignement des butées articulaires.



- d'après les résultats obtenus par la simulation en utilisant la méthode de la pseudo inverse avec un terme d'optimisation relatif à l'éloignement des butées articulaires, on constate que le robot suit la trajectoire désirée avec une erreur de l'ordre de  $10^{-5}$  sans qu'il dépasse les butées articulaires dans les deux cas de figures.

### 2.3.2.2. Utilisation du terme d'optimisation pour augmenter la manipulabilité :

Pour un robot manipulateur, la manipulabilité est exprimée par :

$W(q) = \sqrt{\det[J(q) \cdot J^T(q)]}$ . Sur les positions singulières,  $W$  est minimum et s'annule. Afin d'augmenter la manipulabilité du mécanisme, on peut choisir une fonction scalaire telle que [KHA 99] :

$$\Phi(q) = \det[J(q) \cdot J^T(q)] \quad (2.26)$$

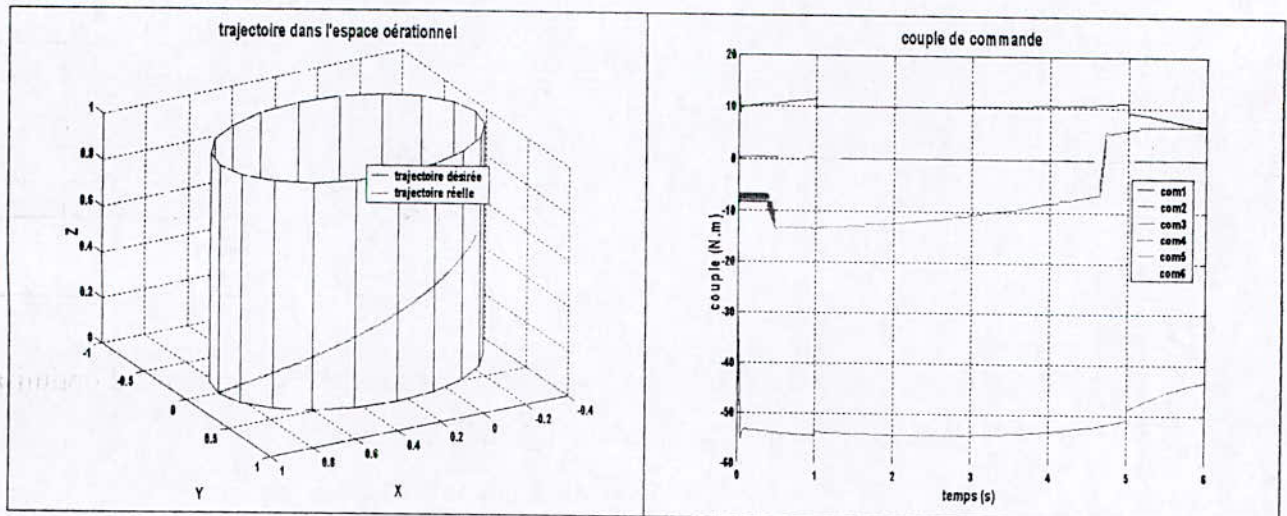
On calcul  $Z$  comme indiqué précédemment et on prend  $\alpha > 0$ . La croissance de  $\Phi$  déplace le robot vers des configurations de manipulabilité optimale et l'éloigne donc de ses configurations singulières.

#### Remarque :

Certaines configurations singulières sont inévitables.

#### 2.3.2.2.1. Résultats et Simulation:

- Dans l'espace :



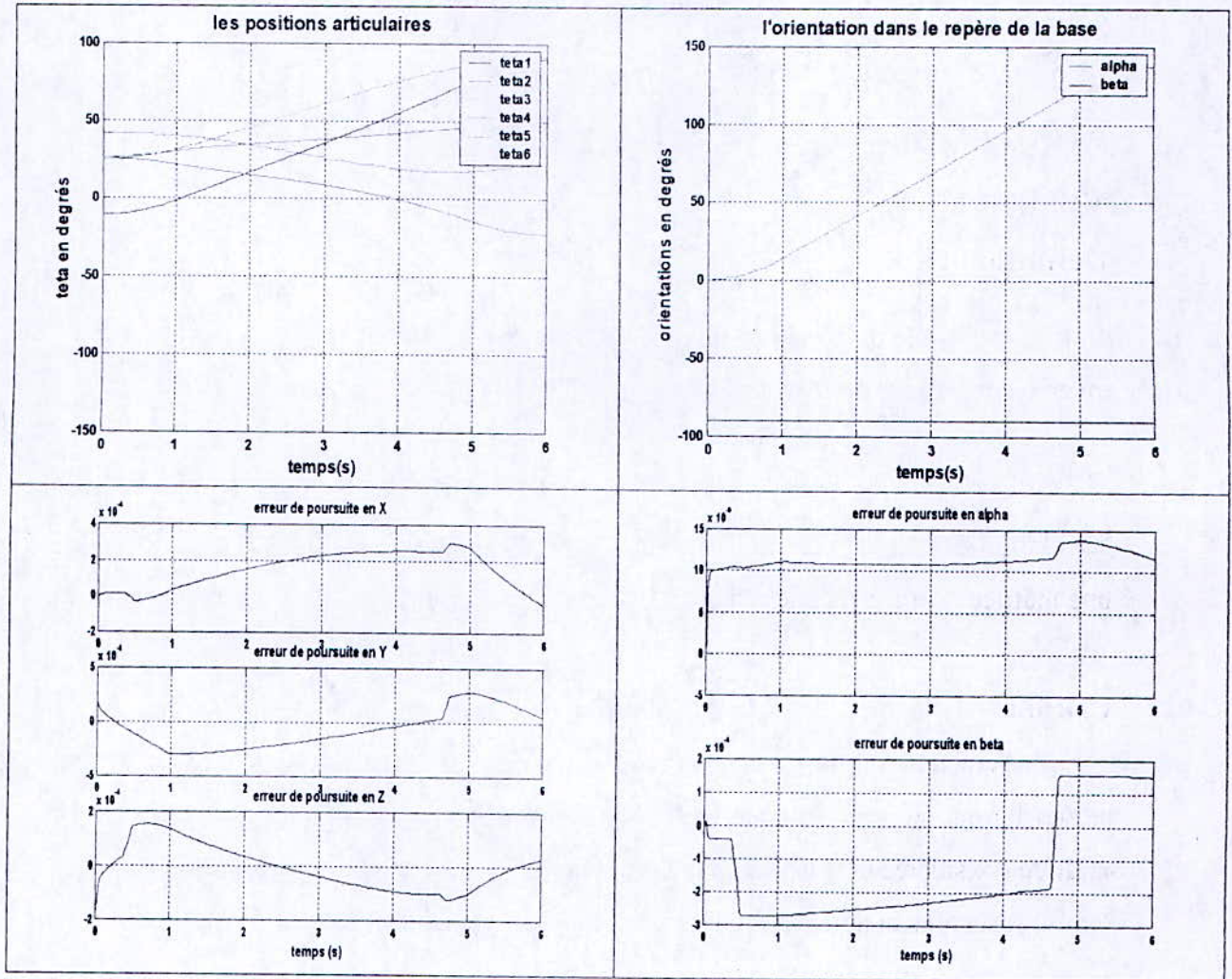


Figure 2.5. Résultats de simulation en utilisant la pseudo inverse avec un terme d'optimisation pour augmenter la manipulabilité.

- D'après les résultats de simulation en utilisant la pseudo inverse avec un terme d'optimisation pour objectif d'augmenter la manipulabilité, on constate que le robot suit très bien la trajectoire, ainsi la manipulabilité du robot est augmentée, ce qui provoque l'éloignement des positions singulières.

### 2.3.3. Utilisation de l'inverse généralisé du jacobien :

#### 2.3.3.1. Utilisation de l'inverse généralisé du jacobien sans terme d'optimisation :

##### Définition :

Soit  $J$  une matrice de dimension  $m \times n$  et de rang  $r$ . On dit que la matrice  $J^*$  est une inverse généralisée de  $J$  si et seulement si [GRE 80]:

$$J \cdot J^* \cdot J = J \quad (2.27)$$

On montre que la matrice  $J$  possède une infinité d'inverses généralisées  $J^*$  sauf lorsque  $J$  est une matrice d'ordre  $n$  et de rang  $n$ ;  $J^*$  s'identifie alors à  $J^{-1}$ , matrice inverse de  $J$ .

##### Calcul :

Actuellement toutes les méthodes proposées pour le calcul d'une inverse généralisée, qui se ne basent pas sur la pseudo inverse, nécessitent de calculer explicitement le rang  $r$  de la matrice considérée et d'isoler un mineur de même ordre que le rang de cette matrice.

Nous présentons ci après la méthode de calcul d'une inverse généralisée la plus utilisée.

Deux cas peuvent se présenter :

##### 1<sup>er</sup> cas :

On partitionne la matrice  $J$  de la façon suivante :

$$J = \begin{pmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{pmatrix} \quad (2.28)$$

De telle sorte que  $J_{11}$  soit une matrice carrée de rang maximal  $r$ .

Alors on peut vérifier aisément que :

$$J^* = \begin{pmatrix} J_{11}^{-1} & 0 \\ 0 & 0 \end{pmatrix} \quad (2.29)$$

Est une inverse généralisées de  $J$ . En effet :

$$J \cdot J^* \cdot J = \begin{pmatrix} J_{11} & J_{12} \\ J_{21} & J_{21} \cdot J_{11}^{-1} \cdot J_{12} \end{pmatrix} \quad (2.30)$$

Or puisque  $J$  et  $J_{11}$ , sont de rang  $r$ , on peut écrire que :

$$J_{21} = K \cdot J_{11} \quad \text{Et} \quad J_{22} = K \cdot J_{12}$$

Donc  $J_{21} \cdot J_{11}^{-1} \cdot J_{12} = K \cdot J_{12} = J_{22}$  d'où le résultat.

2<sup>ème</sup> cas :

On ne partitionne pas la matrice  $J$  comme précédemment. Alors il existe, au moins, une matrice carrée régulière de permutation des lignes  $L$  et une matrice carrée régulière de permutation des colonnes  $C$  telles que :

$$L \cdot J \cdot C = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} = B \quad (2.31)$$

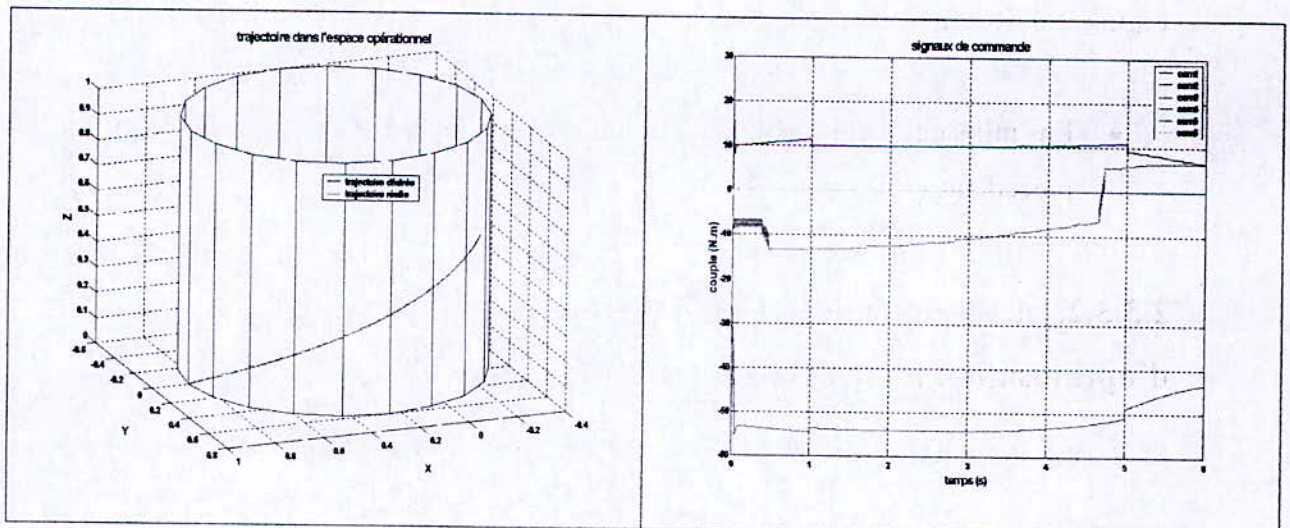
Où  $B_{11}$  est une matrice carrée de rang maximal  $r$ . Si  $B^*$  est une inverse généralisée de  $B$ , définie comme précédemment, alors, on peut vérifier aisément que :

$$J^* = C \cdot B^* \cdot L \quad (2.32)$$

Donc  $J^*$  est une inverse généralisée de  $J$ .

### 2.3.3.1.1. Résultats et Simulation :

- Dans l'espace :



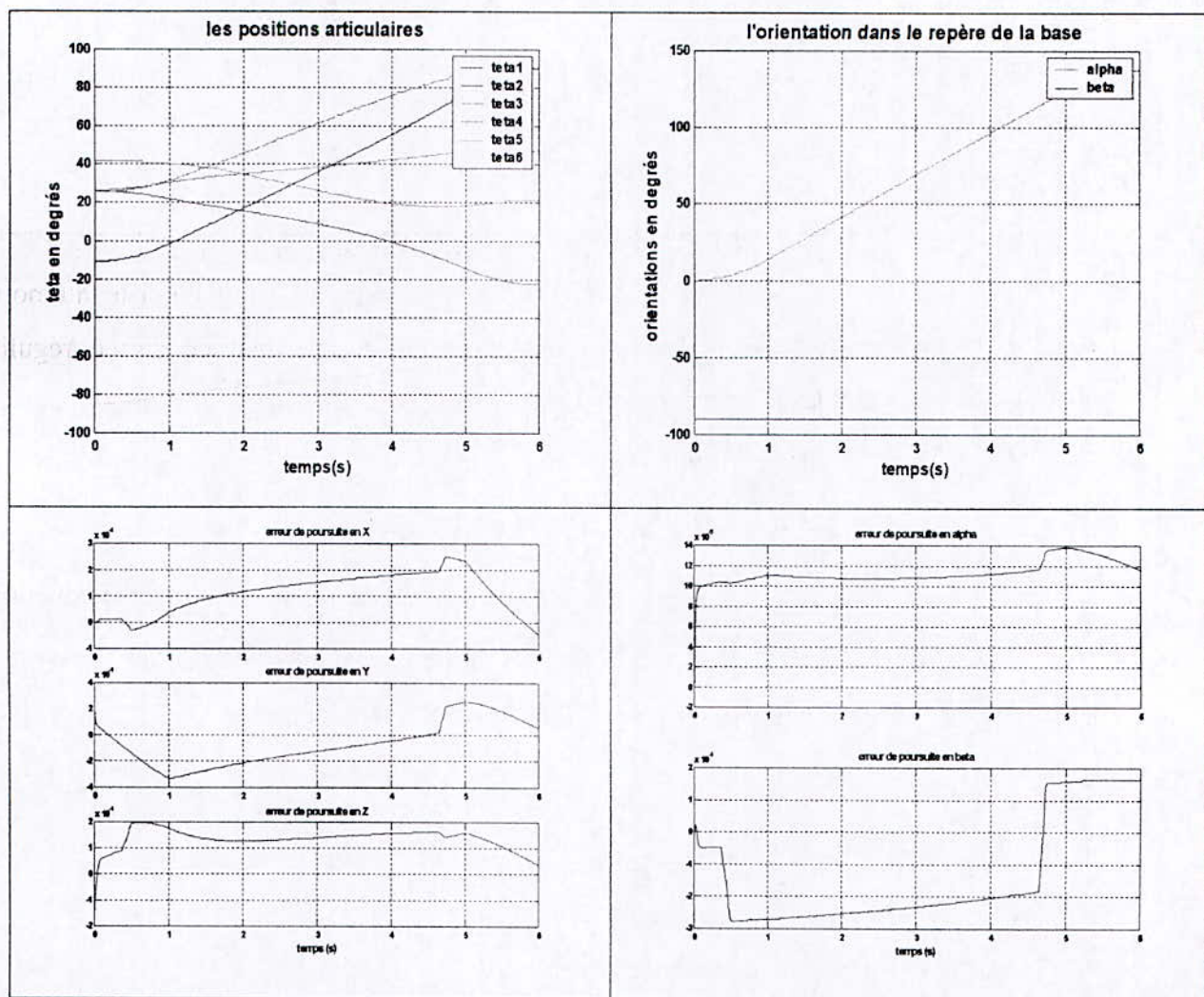


Figure 2.6. Résultats de simulation en utilisant l'inverse généralisé sans terme d'optimisation.

- En utilisant cette méthode on obtient les mêmes performances que les méthodes précédentes.

### 2.3.3.2. Utilisation de l'inverse généralisé du jacobien avec un terme d'optimisation pour l'éloignement des butées articulaires:

Si  $J^*$  est une solution de  $\dot{X} = J \cdot \dot{q}$ , tout  $\dot{q}^*$  tel que :

$$\dot{q}^* = J^* \cdot \dot{X} + (J^* \cdot J - I_n) \cdot Z \tag{2.33}$$

Avec :

$I_n$  : Matrice unité de dimension  $n \times n$ .

$J^*$  : L'inverse généralisé de  $J$  calculée par l'une des méthodes précédentes.

$Z$  : est un vecteur de dimension  $n \times 1$  dont  $(n-m)$  composantes sont arbitraires.

Nous savons calculer l'inverse généralisée de la matrice  $J$  qui n'est pas unique. Reste à déterminer le vecteur  $Z$  dont les  $(n-m)$  composantes sont arbitraires. Le problème peut être résolu si on considère que  $Z$  est le vecteur gradient d'une fonction scalaire dépendant de l'état  $q$  du robot. Sous ces hypothèses, (2.36) il est possible d'assurer la minimisation de cette fonction scalaire qui peut être une fonction potentielle ou une fonction de coût [FOU 80].

Pour le calcul de  $Z$  pour un cas d'un robot à 3 degré de liberté, on a :

$$q_{1\min} \leq q_1 \leq q_{1\max}$$

$$q_{2\min} \leq q_2 \leq q_{2\max}$$

$$q_{3\min} \leq q_3 \leq q_{3\max}$$

Une contrainte physique importante consiste à empêcher les variables articulaires d'aller sur leurs butées quand le robot exécute un mouvement. Pour éviter ceci, on peut construire une fonction de coût évaluant la distance entre la position actuelle de chaque articulation et sa position moyenne, donc de la forme :

$$|q_1 - q_{1\text{moy}}| + |q_2 - q_{2\text{moy}}| + |q_3 - q_{3\text{moy}}| \quad (2.34)$$

Compte tenu du problème posé par les valeurs absolues, il est plus aisé de considérer :

$$(q_1 - q_{1\text{moy}})^2 + (q_2 - q_{2\text{moy}})^2 + (q_3 - q_{3\text{moy}})^2 \quad (2.35)$$

La relation (2.35) conduit à écrire :

$$\Phi(q) = \frac{1}{3} \times \left[ \frac{(q_1 - q_{1\text{moy}})^2}{(q_{1\max} - q_{1\min})^2} + \frac{(q_2 - q_{2\text{moy}})^2}{(q_{2\max} - q_{2\min})^2} + \frac{(q_3 - q_{3\text{moy}})^2}{(q_{3\max} - q_{3\min})^2} \right] \quad (2.36)$$

Le facteur  $1/3$  est introduit pour forcer  $\Phi$  à évoluer dans l'intervalle  $[0 \ 1]$ .

Nous aurons alors une minimisation de  $\Phi$  en choisissant :

$$Z = \nabla \Phi = \left( \frac{\partial \Phi}{\partial q_1}, \frac{\partial \Phi}{\partial q_2}, \frac{\partial \Phi}{\partial q_3} \right)^T \quad (2.37)$$

On aura :

$$Z = \left[ \frac{2}{3} \cdot \frac{(q_1 - q_{1moy})}{(q_{1max} - q_{1min})^2}, \frac{2}{3} \cdot \frac{(q_2 - q_{2moy})}{(q_{2max} - q_{2min})^2}, \frac{2}{3} \cdot \frac{(q_3 - q_{3moy})}{(q_{3max} - q_{3min})^2} \right]^T$$

Qu'on va écrire pour simplifier les notations sous la forme :

$$Z = (\beta_1 \cdot q_1 - \alpha_1, \beta_2 \cdot q_2 - \alpha_2, \beta_3 \cdot q_3 - \alpha_3)^T \quad (2.38)$$

Donc la solution du (2.33) avec prise en compte de (2.38) est une solution qui minimise deux critères simultanément :

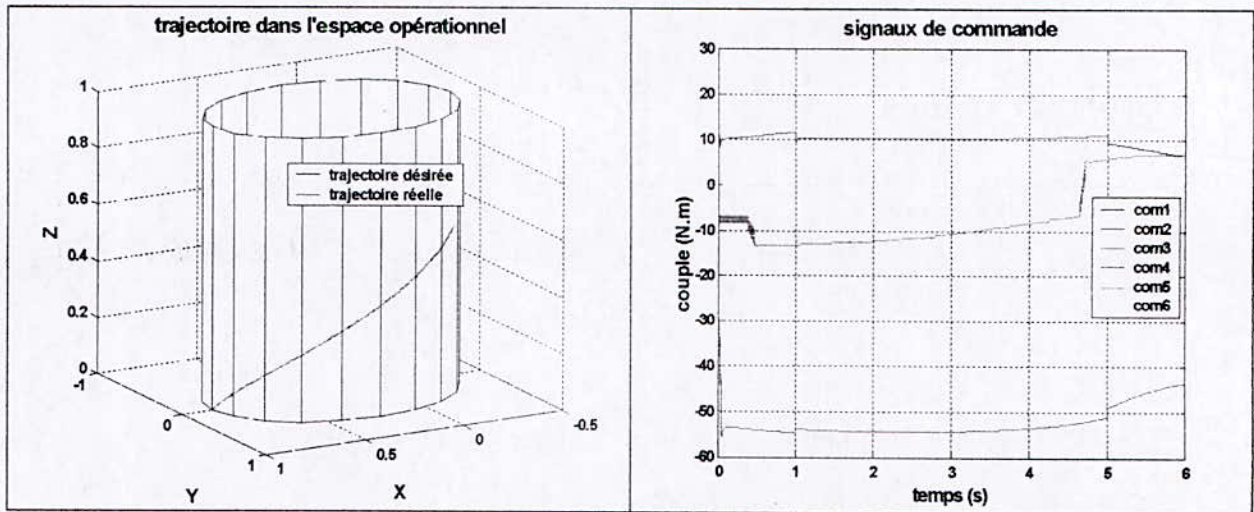
- 1) par la partie  $J^* \cdot \dot{X}$  on minimise  $\sum_{i=1}^n \dot{q}_i^2$ .
- 2) Par sa partie  $(J^* \cdot J - I_n) \cdot Z$  on s'assure que la distance de chaque articulation à sa position moyenne est presque minimale (puisque'on minimise en réalité la somme pondérée des carrés des distances).

**Remarque :**

On peut généraliser cette méthode pour un robot redondant à  $n$  degrés de liberté.

**2.3.3.2.1. Résultats et Simulation :**

- Dans l'espace :



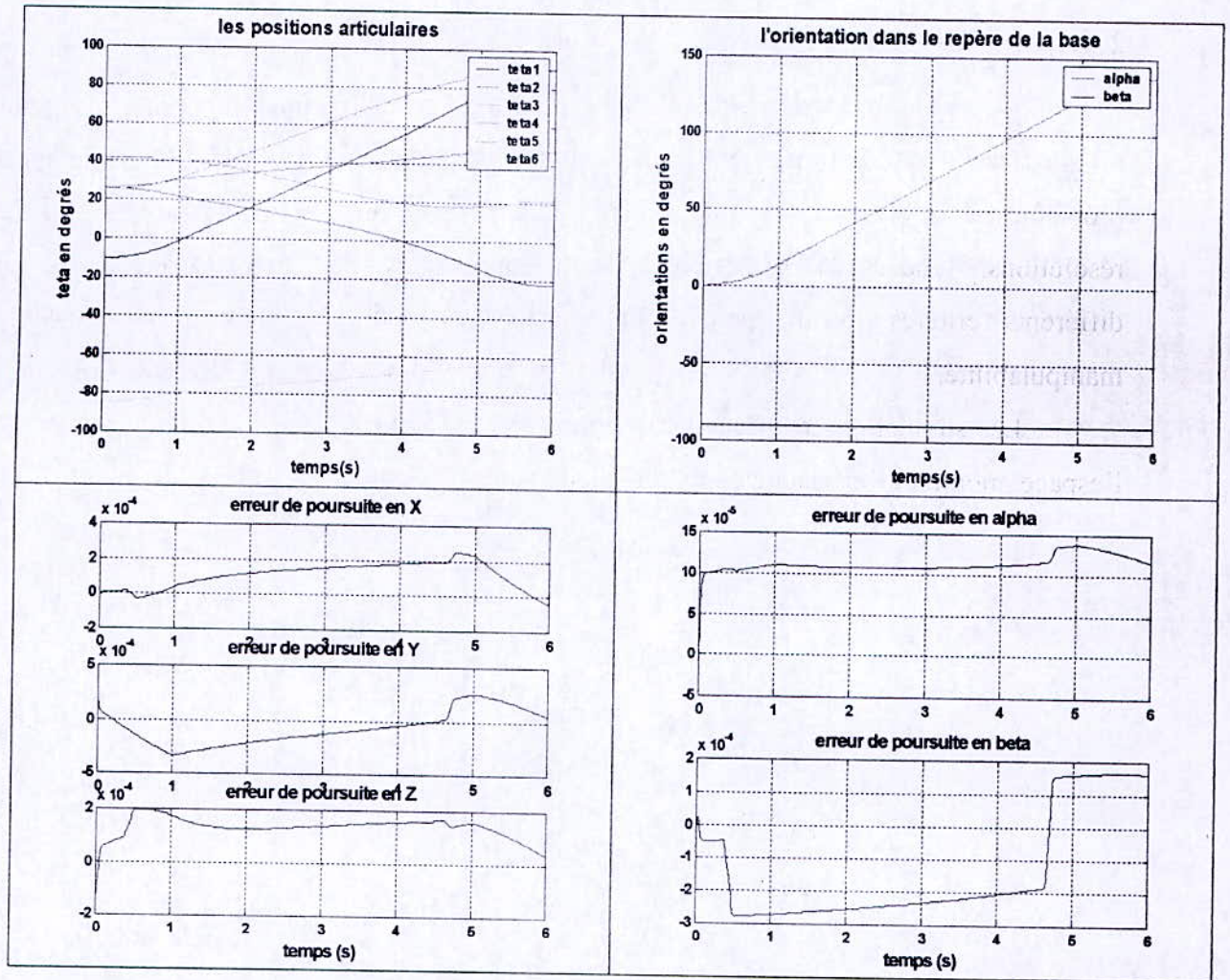


Figure. 2.7. Résultats de simulation en utilisant l'inverse généralisé avec un terme d'optimisation pour l'éloignement des butées articulaires

- Même constat que pour les méthodes précédentes.



## 2.4. CONCLUSION :

Nous avons présenté dans ce chapitre les différentes méthodes pour la résolution du problème de la cinématique inverse des robots manipulateurs redondants. Ces méthodes de résolutions fondées sur la notion de la pseudo inverse qui permet de prendre en compte différents critères locaux pour optimiser l'éloignement des butées mécaniques ou la manipulabilité.

Les simulations réalisées sur le robot PUMA 560 à 6 degrés de libertés évoluant dans l'espace, montrent l'efficacité de toutes les méthodes présentées.



# Les méthodes basées sur la décomposition de la matrice jacobienne

## 3.1. Nouvelle méthode pour résoudre la cinématique inverse proposé par W.Khalil :

La cinématique des robots redondants, correspond à celle d'un robot ayant plus de degrés de libertés que la tâche à réaliser [KHA 88].

### 3.1.1. La solution générale du modèle cinématique inverse :

La relation entre les vitesse articulaires et les vitesses opérationnelles est donnée par :

$$\dot{X} = J(q) \cdot \dot{q} \quad (3.1)$$

Avec :

$J$ : La matrice jacobienne de dimension  $(m \times n)$ .

$n$ : Le nombre du degré de liberté du robot.

$m$ : Le nombre de degré de liberté de la tâche, le degré de redondance dans ce cas est de dimension  $(n-m)$ .

La solution générale de (3.1) est :

$$\dot{q} = J^+ \cdot \dot{X} + (I_n - J^+ \cdot J) \cdot Z \quad (3.2)$$

Avec :

$J^+$ : La matrice pseudo inverse de  $J$ .

$I_n$ : Matrice identité  $(n \times n)$ .

$Z$ : Un vecteur arbitraire.

Le premier terme de (3.2) représente la solution qui minimise la norme des vitesses articulaires, le second terme utilise la projection dans l'espace de la solution homogène de (3.1)

Pour optimiser une fonction  $\Phi(q)$  :

$$Z = \alpha \cdot \nabla \Phi(q) \quad (3.3)$$

Avec :

$\nabla \Phi(q)$  : Le gradient de  $\Phi(q)$ .

$\alpha$  : Constant positif pour une maximisation de  $\Phi(q)$  ou négatif pour une minimisation de  $\Phi(q)$ .

### 3.1.2. Décomposition de la matrice jacobienne :

On considère que  $J$  est de rang égale à  $m$ , dans ce cas on peut écrire la matrice  $J$  sous la forme de produit de deux matrices :

$$J = J_m \cdot C \quad (3.4)$$

Avec :

$J_m$  : Sous matrice de  $J$  régulière de dimension  $m \times m$ .

$C$  : Une matrice de dimension  $m \times n$ .

La matrice  $J_m$  est une sous matrice de  $J$  de sa partie gauche de dimension  $(m \times m)$ .

De (3.4) on peut écrire que :

$$C = [I_m : C_{m+1} \dots C_n] = [I_m : C_1] \quad (3.5)$$

Avec :

$$C_k = J_m^{-1} \cdot J_k \quad (3.6)$$

$I_m$  : Matrice identité de dimension  $m \times m$ .

$J_k$  : La  $k^{\text{ème}}$  colonne de  $J$ .

$C_1$  : Une matrice de dimension  $m \times n - m$ .

Donc après cette décomposition on peut écrire la pseudo inverse comme suite :

$$J^+ = C^+ \cdot J_m^{-1} \quad (3.7)$$

D'où :

$$\dot{q} = C^+ \cdot J_m^{-1} \cdot \dot{X} \quad (3.8)$$

La décomposition physique est comme suit:

$$\dot{q}_p = J_m^{-1} \cdot \dot{X} \quad (3.9)$$

La solution particulière de (3.1) en utilisant les articulations primaires.

$$\dot{q} = C^+ \cdot \dot{q}_p \quad (3.10)$$

Les articulations secondaires permettent d'obtenir  $\dot{q}$  qui satisfait le critère de la minimisation de la norme.

**Remarque :**

- i) Si le robot est dans une position singulière les résultats développés dans les paragraphes (3.1.2, 3.1.3, 3.1.4) peuvent être appliqués, mais il faut remplacer  $\dot{X}$  par le plus proche vecteur commandable  $\dot{X}_a$ .
- ii) Si la solution particulière des articulations primaires  $q_p$  peut être obtenue par l'expression  $q_p = g(X)$ , dans ce cas de figure, on peut appliquer un algorithme donné pour résoudre la cinématique inverse pour les robots redondants.

### 3.1.3. Dérivation de la pseudo inverse :

On peut obtenir le vecteur  $\dot{q}$  de (3.8) par les méthodes suivantes :

i) On cherche  $J_m^{-1}$  et  $C^+$ , et en utilisant (3.9) et (3.10).

La matrice  $C^+$  est donnée par :

$$C^+ = \begin{bmatrix} I_m - C_1 \cdot (I_{n-m} + C_1^T \cdot C_1)^{-1} \cdot C_1^T \\ (I_{n-m} + C_1^T \cdot C_1) \cdot C_1^T \end{bmatrix} \quad (3.11)$$

Ce qui donne :

$$C^+ = \begin{bmatrix} I_m \\ 0_{n-m,m} \end{bmatrix} - \begin{bmatrix} C_1 \\ -I_{n-m} \end{bmatrix} \cdot (I_{n-m} + C_1^T \cdot C_1)^{-1} \cdot C_1^T \quad (3.12)$$

D'où la solution complète est donnée par :

$$\dot{q} = \begin{bmatrix} \dot{q}_p \\ 0_{n-m,m} \end{bmatrix} - \begin{bmatrix} C_1 \\ -I_{n-m} \end{bmatrix} \cdot (I_{n-m} + C_1^T \cdot C_1)^{-1} \cdot C_1^T \cdot \dot{q}_p \quad (3.13)$$

ii) On peut considéré le calcul de  $\dot{q}$  comme un problème d'optimisation.

L'optimisation commence par l'obtention d'une solution particulière  $\dot{q}_p = J_m^{-1} \cdot \dot{X}$ , après en observant la solution qui a la norme minimale.

$$\dot{X} = J_m \cdot \dot{q}_p = J_m \cdot [I_m : C_1] \cdot \dot{q}$$

Avec  $J_m$  : matrice régulière

$$\dot{q}_p = [I_m : C_1] \cdot \dot{q} \quad (3.14)$$

La relation (3.14) peut s'écrire:

$$\dot{q}_p = \dot{q}_m + C_1 \cdot \dot{q}_c$$

D'où :

$$\dot{q}_m = \dot{q}_p - C_1 \cdot \dot{q}_c \quad (3.15)$$

Avec :

$\dot{q}_m$  : Les premiers  $m$  composantes de  $\dot{q}$ .

$\dot{q}_c$  :  $(n-m)$  composantes de  $\dot{q}$ .

En utilisant (3.15) on peut calculer  $\dot{q}_m$  pour n'importe quel valeur de  $\dot{q}_c$ .

Pour la pseudo inverse on doit minimiser le critère :

$$\phi = \|\dot{q}\|^2 = \|\dot{q}_p - C_1 \cdot \dot{q}_c\|^2 + \|\dot{q}_c\|^2 \quad (3.16)$$

La dérivée de (3.16) par rapport  $\dot{q}_c$  donne :

$$C_1^T \cdot C_1 \cdot \dot{q}_c - C_1^T \cdot \dot{q}_p = 0 \quad (3.17)$$

D'où la solution optimale de  $\dot{q}_c$  est :

$$\dot{q}_c = (I_{n-m} + C_1^T \cdot C_1)^{-1} \cdot C_1^T \cdot \dot{q}_p \quad (3.18)$$

La matrice  $(I_{n-m} + C_1^T \cdot C_1)$  doit être définie positive.

Les étapes des calculs de cette méthode sont données par :

1. On calcul  $\dot{q}_p$  en utilisant (3.9).
2. On calcul  $\dot{q}_c$  par la formule (3.18).
3. On calcul  $\dot{q}_m$  par la formule (3.15).

### 3.1.4. La dérivée de la pseudo inverse combinée avec la solution homogène :

On remplace l'expression (3.4) de  $J$  et (3.7) de  $J^+$  dans (3.2), on obtient :

$$\dot{q} = C^+ \cdot J_m^{-1} \cdot \dot{X} + (I_n - C^+ \cdot C) \cdot Z \quad (3.19)$$

a partir de (3.4) on obtient :

$$(I_n - C^+ \cdot C) = \begin{bmatrix} C_1 \\ -I_{n-m} \end{bmatrix} \cdot (I_{n-m} + C_1^T \cdot C_1)^{-1} \cdot [C_1^T \ : \ -I_{n-m}] \quad (3.20)$$

En utilisant (3.12) et (3.20), on peut écrire l'équation (3.19) :

$$\begin{aligned} \dot{q} &= \begin{bmatrix} I_m \\ 0_{n-m,m} \end{bmatrix} - \begin{bmatrix} C_1 \\ -I_{n-m} \end{bmatrix} \cdot (I_{n-m} + C_1^T \cdot C_1)^{-1} \cdot C_1^T \cdot J_m^{-1} \cdot \dot{X} \\ &+ \begin{bmatrix} C_1 \\ -I_{n-m} \end{bmatrix} \cdot (I_{n-m} + C_1^T \cdot C_1)^{-1} \cdot [C_1^T \ : \ -I_{n-m}] \cdot Z \end{aligned} \quad (3.21)$$

On définit  $\dot{Q}$  comme suite :

$$\dot{Q} = \begin{bmatrix} J_m^{-1} \cdot \dot{X} \\ 0_{n-m,1} \end{bmatrix} = \begin{bmatrix} \dot{q}_p \\ 0_{n-m,1} \end{bmatrix} \quad (3.22)$$

D'où (3.21) devient :

$$\dot{q} = \dot{Q} - \begin{bmatrix} C_1 \\ -I_{n-m} \end{bmatrix} \cdot (I_{n-m} + C_1^T \cdot C_1)^{-1} \cdot [C_1^T \ : \ -I_{n-m}] \cdot (\dot{Q} - Z) \quad (3.23)$$

Qui donne :

$$\dot{q}_c = (I_{n-m} + C_1^T \cdot C_1)^{-1} \cdot [C_1^T \ : \ -I_{n-m}] \cdot (\dot{Q} - Z) \quad (3.24)$$

D'où la solution complète est obtenue par les étapes suivantes :

1. On calcul  $\dot{q}_p$  en utilisant (3.9).
2. On calcul  $\dot{q}_c$  en utilisant (3.24).
3. On calcul  $\dot{q}_m$  en utilisant (3.15).

### 3.1.5. Utilisation d'autres critères :

#### 3.1.5.1. Critère quadratique :

Si  $\tilde{q}_i$  la valeur moyen de la  $i^{\text{ème}}$  position, avec  $e_i = (q_{i_{\max}} - q_{i_{\min}})$ , le critère à minimiser est :

$$\Phi_q = \sum_{i=1}^n \left[ \frac{(q_i - \tilde{q}_i)^2}{e_i^2} \right] \quad (3.25)$$

On peut obtenir une solution exacte en remplaçant le critère (3.16) par le critère (3.25).

$q(k)$  : La position articulaire courante, la prochaine position peut être calculer par la relation suivante :

$$q(k+1) = q(k) + \dot{q} \cdot \varepsilon$$

Avec  $\varepsilon$  est le pas d'intégration.

On prend :

$$W = \text{diag}\left(\frac{1}{e_i^2}\right)$$

$$\Delta q = q(k) - \tilde{q}$$

Donc :

$$\tau = (\Delta q + \dot{q} \cdot \varepsilon)^T \cdot W \cdot (\Delta q + \dot{q} \cdot \varepsilon) \quad (3.26)$$

On peut partager les matrices et les vecteurs de (3.26) comme suite :

$$\phi = (\Delta q_m + \dot{q}_m \cdot \varepsilon)^T \cdot W_m \cdot (\Delta q_m + \dot{q}_m \cdot \varepsilon) + (\Delta q_c + \dot{q}_c \cdot \varepsilon)^T \cdot W_c \cdot (\Delta q_c + \dot{q}_c \cdot \varepsilon)$$

On dérivant (3.15) par rapport a  $\dot{q}_c$ , on aura :

$$(W_c + C_1^T \cdot W_m \cdot C_1) \cdot \varepsilon^2 \cdot \dot{q}_c = C_1^T \cdot W_m \cdot (\Delta q_m + \dot{q}_p \cdot \varepsilon) \cdot \varepsilon - W_c \cdot \Delta q_c \cdot \varepsilon \quad (3.27)$$

Donc  $\dot{q}$  est calculée par :

$$\dot{q}_c = (W_c + C_1^T \cdot W_m \cdot C_1)^{-1} \cdot [C_1^T \cdot W_m \cdot (\Delta q_m / \varepsilon + \dot{q}_p) - W_c \cdot (\Delta q_c / \varepsilon)] \quad (3.28)$$

$$\dot{q}_m = \dot{q}_p - C_1 \cdot \dot{q}_c$$

### 3.1.5.2. Critère min-max :

La méthode des moindres carrés est utilisées pour résoudre beaucoup de problèmes, à cause de la simplicité de la mise en oeuvre mais aussi parce quelle donne de bons résultats, le critère idéal pour distribuer les positions articulaires pour minimiser [(3.24), (3.9)] , est :

$$\phi_q = \max\left(\frac{|q_i - \tilde{q}_i|}{e_i}\right) \text{ pour } i=1, \dots, n \quad (3.29)$$

On peut considérer que dans la formule (3.15) que toutes les variables  $q_i$  sont linéaire en fonction de  $\dot{q}_c$  :

$$q_i(k+1) = q_i(k) + \dot{q}_{p_i} \cdot \varepsilon + C_{1i} \cdot \dot{q}_c \cdot \varepsilon$$

Avec :  $C_{1i}$  la  $i^{\text{ème}}$  ligne de  $C_1$ .

Donc, un problème qui représente un critère linéaire avec des contraintes linaires, peut être résolu par la méthode de programmation linéaire (algorithme de simplex).



Si le degré de redondance  $(n-m)$  égal à 1, on peut avoir une solution simple, on peut voir pour un exemple d'un robot planaire à 3 degrés de liberté, l'évolution de critère (3.29) pour obtenue la solution  $\dot{q}_c$  voir figure (3.1).

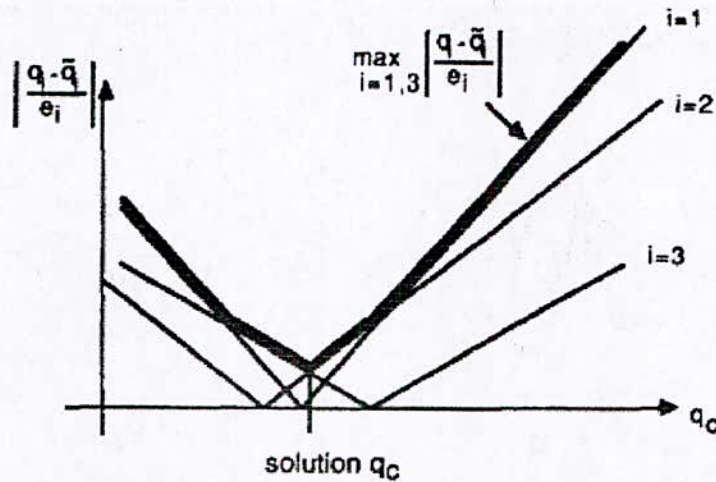


Figure 3.1. La solution  $q_c$  on utilisant le critère min-max.

### 3.1.6. La pseudo inverse pondéré :

La pseudo inverse peut être pondéré car on prévoit un mécanisme additionnel pour réaliser les performances désirées, l'inverse généralisée  $J_w^+$  utilisée pour minimiser  $\dot{q}^T \cdot W \cdot \dot{q}$  donner par (3.20), (3.21) :

$$J_w^+ = W^{-1} \cdot J^T \cdot (J \cdot W^{-1} \cdot J^T)^{-1} \quad (3.30)$$

Avec  $W$  matrice de pondération, On calcule la valeur optimale de  $\dot{q}$  par une méthode similaire qui est proposée dans la remarque paragraphe (3.1.2), dans ce cas le critère à minimiser est fonction de  $\dot{q}$ , on pourra facilement le minimiser, donc :

$$\phi = \dot{q}^T \cdot W \cdot \dot{q} \quad (3.31)$$

On peut partitionner ce critère comme suite :

$$\phi = \begin{bmatrix} \dot{q}_m^T & \dot{q}_c^T \end{bmatrix} \cdot \begin{bmatrix} W_{mm} & W_{mc} \\ W_{cm} & W_{cc} \end{bmatrix} \cdot \begin{bmatrix} \dot{q}_m \\ \dot{q}_c \end{bmatrix}$$

Qui donne :

$$\phi = \dot{q}_m^T \cdot W_{mm} \cdot \dot{q}_m + \dot{q}_c^T \cdot W_{cm} \cdot \dot{q}_m + \dot{q}_m^T \cdot W_{mc} \cdot \dot{q}_c + \dot{q}_c^T \cdot W_{cc} \cdot \dot{q}_c \quad (3.32)$$

En général  $W$  est une matrice symétrique, donc :

$$\phi = \dot{q}_m^T \cdot W_{mm} \cdot \dot{q}_m + \dot{q}_c^T \cdot W_{cc} \cdot \dot{q}_c + 2 \cdot \dot{q}_m^T \cdot W_{mc} \cdot \dot{q}_c \quad (3.33)$$

On remplace  $\dot{q}_m$  de (3.15) dans (3.33), on obtient :

$$\begin{aligned} \phi &= \dot{q}_p^T \cdot W_{mm} \cdot \dot{q}_p + 2 \cdot \dot{q}_p^T \cdot (W_{mc} - W_{mm} \cdot C_1) \cdot \dot{q}_c \\ &+ \dot{q}_c^T \cdot (W_{cc} + C_1^T \cdot W_{mm} \cdot C_1 - 2 \cdot C_1^T \cdot W_{mc}) \cdot \dot{q}_c \end{aligned} \quad (3.34)$$

Pour la valeur optimale de  $\dot{q}_c$ , la dérivée de (3.34) est :

$$\begin{aligned} &\left[ W_{cc} + C_1^T \cdot W_{mm} \cdot C_1 - C_1^T \cdot W_{mc} - W_{mc}^T \cdot C_1 \right] \cdot \dot{q}_c \\ &+ (W_{mc}^T - C_1^T \cdot W_{mm}) \cdot \dot{q}_p = 0 \end{aligned} \quad (3.35)$$

On trouve :

$$\dot{q}_c = (W_{cc} + C_1^T \cdot W_{mm} \cdot C_1 - C_1^T \cdot W_{mc} - W_{mc}^T \cdot C_1)^{-1} \cdot (C_1^T \cdot W_{mm} - W_{mc}^T) \cdot \dot{q}_p \quad (3.36)$$

La matrice  $\left[ W_{cc} + C_1^T \cdot W_{mm} \cdot C_1 - C_1^T \cdot W_{mc} - W_{mc}^T \cdot C_1 \right]$  est de dimension  $(n-m) \times (n-m)$  est définie positive.

Si  $W$  est une matrice diagonale, (3.36) devient :

$$\dot{q}_c = (W_{cc} + C_1^T \cdot W_{mm} \cdot C_1)^{-1} \cdot C_1^T \cdot W_{mm} \cdot \dot{q}_p \quad (3.37)$$

La solution complète est obtenue en utilisant l'équation (3.15).

Pour minimiser l'énergie cinétique, on prend la matrice  $W$  comme étant la matrice d'inertie de robot.

### 3.1.7. Minimisation des couples articulaires :

Les solutions homogènes sont aussi utilisées pour placer les couples articulaires au voisinage des couples articulaires moyens dans cette partie on présente deux méthodes : la méthode classique et la méthode moderne qui n'exige pas beaucoup de calcul.

La dérivée de (3.1) donne :

$$\ddot{X} = J \cdot \ddot{q} + \dot{J} \cdot \dot{q} \quad (3.38)$$

On tire  $\ddot{q}$  à partir de (3.38) :

$$\ddot{q} = J^+ \cdot (\ddot{X} - \dot{J} \cdot \dot{q}) + (I_n - J^+ \cdot J) \cdot Z \quad (3.39)$$

Le couple articulaire est donné par l'équation dynamique suivante :

$$\Gamma = A \cdot \ddot{q} + B(q, \dot{q}) \quad (3.40)$$

Avec :

$A$  : Matrice d'inertie de robot.

$B(q, \dot{q})$  : Matrices : Centrifuge, Gravité, Coriolis et frottement.

On remplace (3.39) dans (3.40), on trouve :

$$\Gamma = A \cdot J^+ \cdot (\ddot{X} - \dot{J} \cdot \dot{q}) + A \cdot (I_n - J^+ \cdot J) \cdot Z + B(q, \dot{q}) \quad (3.41)$$

On peut écrire (3.41) sous la forme :

$$\Gamma = \Gamma' + A \cdot (I_n - J^+ \cdot J) \cdot Z \quad (3.42)$$

Avec :

$$\Gamma' = A \cdot J^+ \cdot (\ddot{X} - \dot{J} \cdot \dot{q}) + B(q, \dot{q}) \quad (3.43)$$

$\Gamma'$  est Indépendant de  $Z$

On prend  $\Gamma_{moy}$  le couple articulaire moyenne égale à  $\frac{[\Gamma_{max} + \Gamma_{min}]}{2}$

Avec :

$\Gamma_{max}$  : Le couple articulaire maximum.

$\Gamma_{min}$  : Le couple articulaire minimum.

Le critère à minimiser est :

$$\phi_{\Gamma} = \left\| \Gamma' + A \cdot (I_n - J^+ \cdot J) \cdot Z - \Gamma_{moy} \right\| \quad (3.44)$$

La solution est obtenue en faisant  $\phi_{\Gamma} = 0$  d'où la solution :

$$Z = \left[ A \cdot (I_n - J^+ \cdot J) \right]^+ \cdot (\Gamma_{moy} - \Gamma') \quad (3.45)$$

Donc le couple articulaire est calculé par l'expression suivante :

$$\Gamma = \Gamma' + \Gamma''$$

Avec :

$$\Gamma' = A \cdot J^+ \cdot (\ddot{X} - \dot{J} \cdot \dot{q}) + B(q, \dot{q})$$

$$\Gamma'' = A \cdot (I_n - J^+ \cdot J) \cdot [A \cdot (I_n - J^+ \cdot J)]^\dagger \cdot (\Gamma_{moy} - \Gamma') \quad (3.46)$$

On peut calculer (3.46) en utilisant une décomposition tel que :

$$H = A \cdot (I_n - J^+ \cdot J)$$

à partir de (3.4), (3.7), (3.20), on aura :

$$H = \begin{bmatrix} A_{mm} & A_{mc} \\ A_{cm} & A_{cc} \end{bmatrix} \cdot \begin{bmatrix} C_1 \\ -I_{n-m} \end{bmatrix} \cdot (I_{n-m} + C_1^T \cdot C_1)^{-1} \cdot [C_1^T \quad -I_{n-m}] \quad (3.47)$$

Ce qui donne :

$$H = U \cdot V$$

Avec :

$$U = \begin{bmatrix} A_{mm} \cdot C_1 & -A_{mc} \\ A_{cm} \cdot C_1 & -A_{cc} \end{bmatrix}, \quad U \text{ est une matrice de dimension } (n \times n-m).$$

$$V = [(I_{n-m} + C_1^T \cdot C_1)^{-1} \cdot C_1^T \quad -(I_{n-m} + C_1^T \cdot C_1)^{-1}]$$

$V$  est une matrice de dimension  $(n-m \times n)$ .

$U$  et  $V$  sont deux matrices de rang plein, donc :

$$H^+ = V^+ \cdot U^+ \Rightarrow H \cdot H^+ = U \cdot V \cdot V^+ \cdot U^+ = U \cdot U^+$$

On général :

$$\Gamma'' = U \cdot U^+ \cdot (\Gamma_{moy} - \Gamma') \quad (3.48)$$

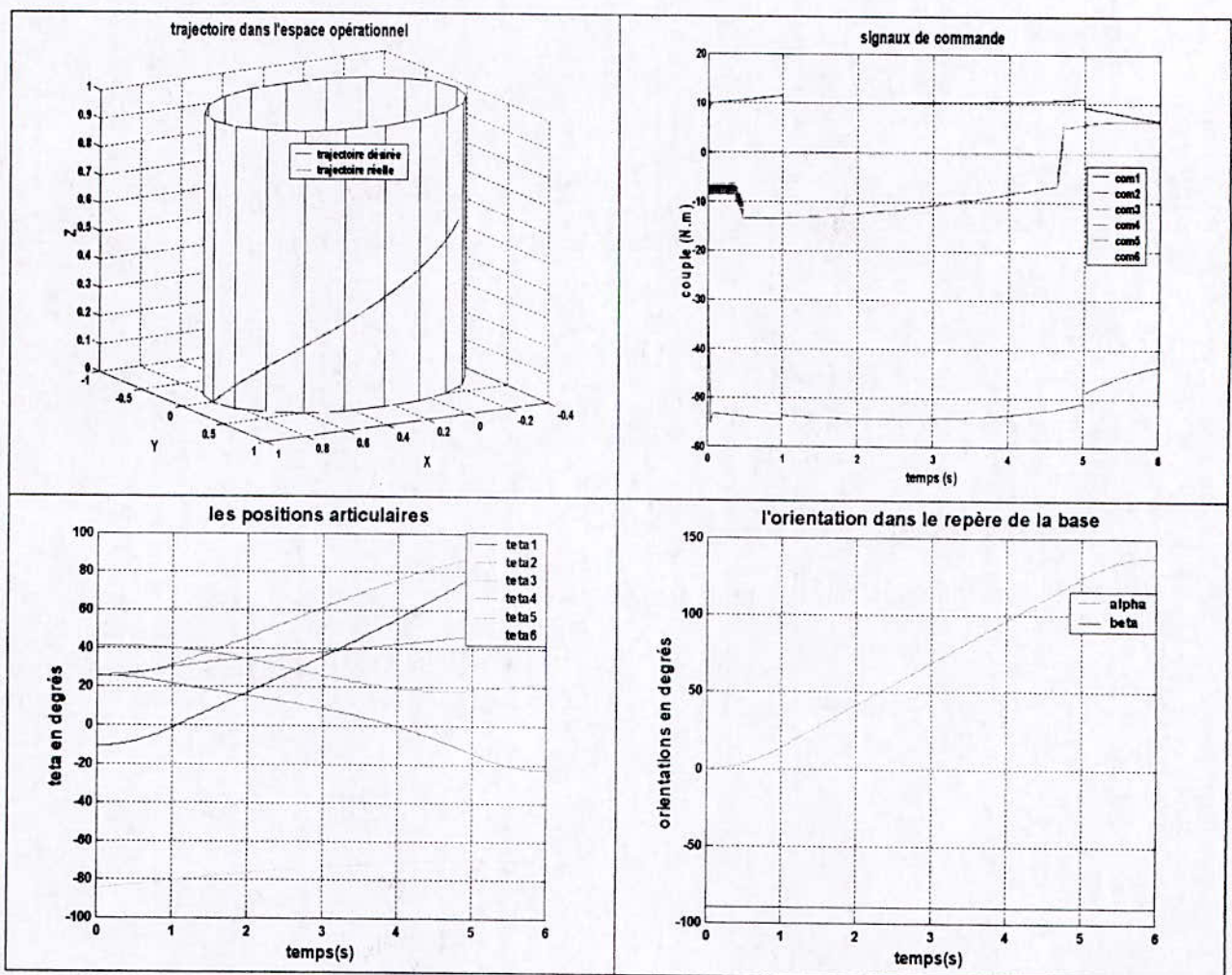
Le calcul de  $\Gamma''$  par (3.48), est beaucoup plus simple que par l'expression (3.46), de plus si on a un redondance d'ordre 1, donc  $U$  sera un vecteur, alors la solution est donnée par :

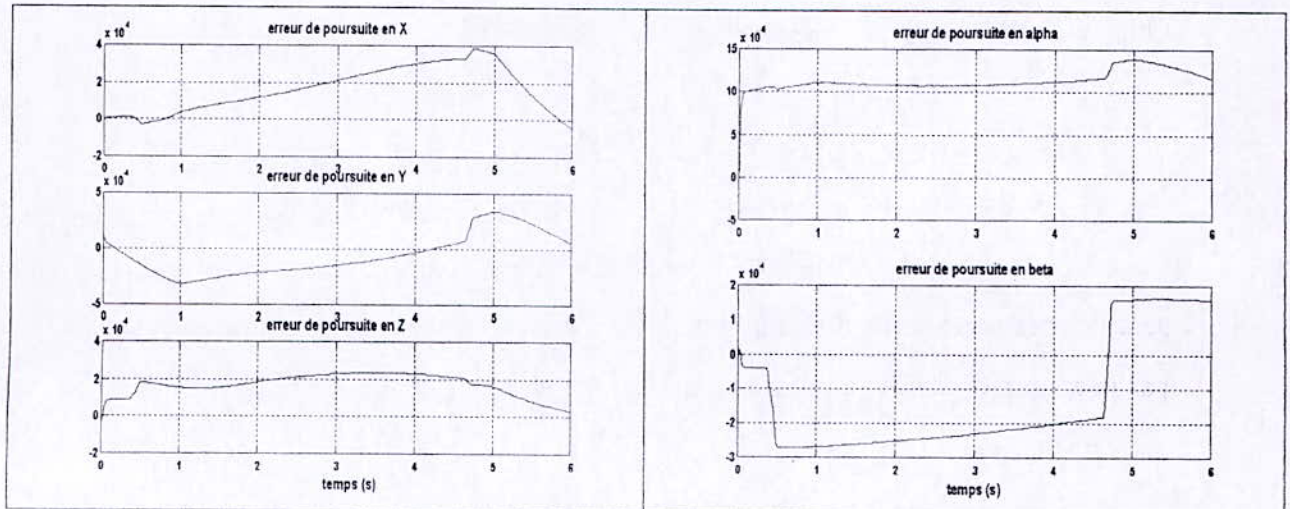
$$\Gamma'' = \frac{U \cdot U^T \cdot (\Gamma_{moy} - \Gamma')}{\|U\|^2} \tag{3.49}$$

### 3.1.7.1. Résultats et simulation :

On utilise la minimisation des couples articulaires combiné avec l'éloignement des butées articulaires, on obtient les résultats suivantes :

- Dans l'espace :





**Figure 3.2.** Résultats de simulation en utilisant la décomposition de la pseudo inverse avec deux termes d'optimisation, pour l'éloignement des butées et la minimisation des couples.

- D'après les résultats de simulation, on conclut qu'on peut réaliser les deux objectifs en même temps, la minimisation des couples articulaires et l'éloignement des butées, ce qui est bien traduit par les résultats obtenus.

### Remarque :

Toutes les méthodes basées sur la décomposition du jacobien sont utiles pour résoudre le problème de la cinématique, et on obtient les mêmes résultats que pour la pseudo inverse sans décomposition.

### 3.2. Décomposition en valeurs singulières de la matrice jacobienne (SVD):

La relation (3.1) entre les vitesses articulaires et les vitesses opérationnelles est, pour une configuration donnée, un système d'équations linéaires dans lequel la matrice  $J$  représente une application linéaire de  $\mathcal{R}^{\dot{X}}$  dans  $\mathcal{R}^{\dot{X}}$ .

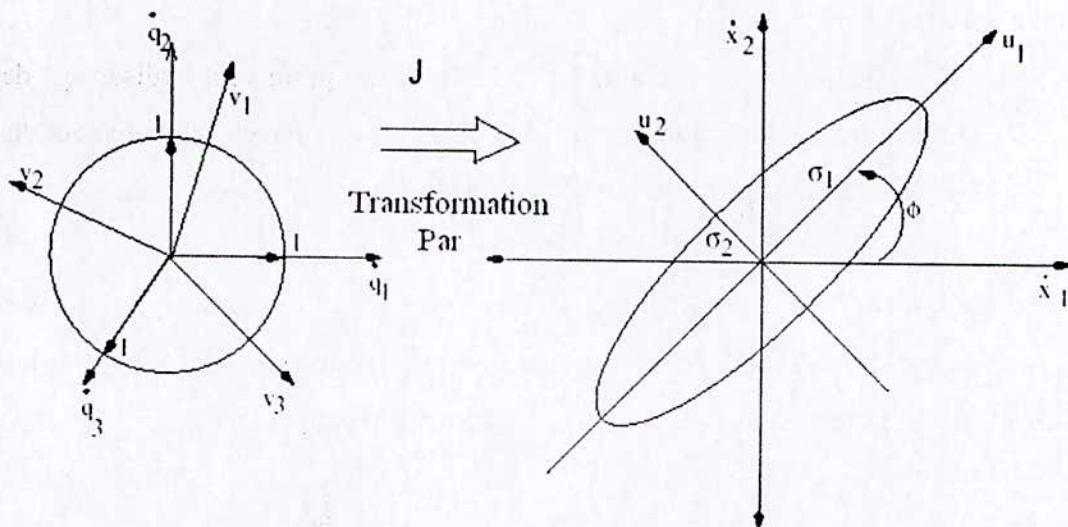
La matrice  $J$  peut être décomposée en un produit de trois matrices faisant intervenir les valeurs singulières de  $J$ , [LAW 74], [DON 79]. Si  $J$  est une matrice de dimension  $(m \times n)$  et de rang  $r$  (c'est à dire le nombre des valeurs singulières  $\sigma_i$  non nulle).

On peut écrire :

$$J=U \cdot D \cdot V^T \quad (3.50)$$

Dans ce qui suit on se limite à étudier le cas d'un robot à trois degré de liberté (mais sera valable pour le cas 6 degré de liberté). On pourra avoir le nouveau repère de la figure (3.3) par une simple rotation autour de l'ancien repère avec une angle  $\phi$ , donc redéfinir les vecteurs dans le nouveau repère par la transformation en utilisant la matrice  $U$  tel que :

$$U = \begin{bmatrix} u_1 & u_2 \end{bmatrix} = \begin{bmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{bmatrix} \quad (3.51)$$



**Figure 3.3.** L'interprétation géométrique de la décomposition en valeurs singulières.

On peut représenter les vecteurs propres  $v_i$  mathématiquement dans la matrice :

$$V = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} \quad (3.52)$$

Donc on peut reformuler l'équation (3.1) dans le nouveau repère, on trouve :

$$U^T \cdot \dot{X} = D \cdot V^T \cdot \dot{q} \quad (3.53)$$

Le terme  $U^T \cdot \dot{X}$  représente la projection de vitesse opérationnel de l'organe terminale dans le repère  $u_1$  et  $u_2$ ,  $V^T \cdot \dot{q}$  représente la projection de vitesse articulaires dans le repère  $v_1$ ,  $v_2$  et  $v_3$ ,  $D$  est une matrice diagonale qui comporte  $\sigma_i$  dans sa diagonal, zéros ailleurs :

$$D = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \end{bmatrix} \quad (3.54)$$

Les valeurs  $\sigma_1$  et  $\sigma_2$  sont obtenue dans la figure (3.3) à partir du schéma de l'ellipse.

$U$  est orthogonale, donc  $U \cdot U^T = I$ . En multipliant les deux parties de l'équation (3.53), on aura :

$$\dot{X} = \underbrace{U \cdot D \cdot V^T}_J \cdot \dot{q} \quad (3.55)$$

On peut formuler la décomposition de la matrice jacobienne sous forme d'une somme, tel que :

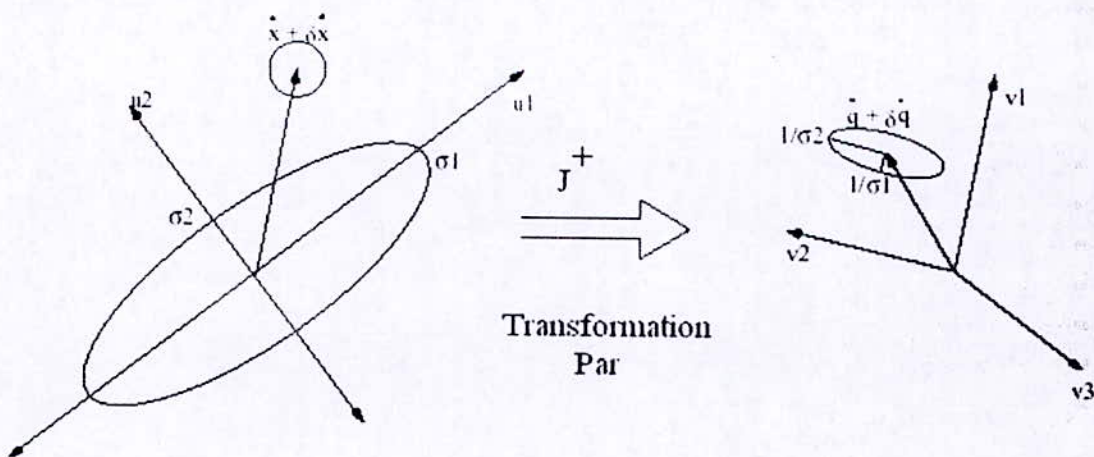
$$J = \sum_{i=1}^{\min(m,n)} \sigma_i \cdot u_i \cdot v_i^T \quad (3.56)$$

On peut donc obtenir la pseudo inverse par la décomposition en valeurs singulières, en utilisant l'inverse de toutes les valeurs singulières non nulles. La pseudo inverse est donnée par :

$$J^+ = \sum_{i=1}^r \frac{1}{\sigma_i} \cdot v_i \cdot u_i^T \quad (3.57)$$

La solution de système (3.1) peut être écrite sous la forme :

$$\dot{q} = J^+ \cdot \dot{X} \quad (3.58)$$



**Figure 3.4.** La représentation géométrique de la transformation des vitesses opérationnelles par la pseudo inverse pour le calcul des vitesses articulaires.

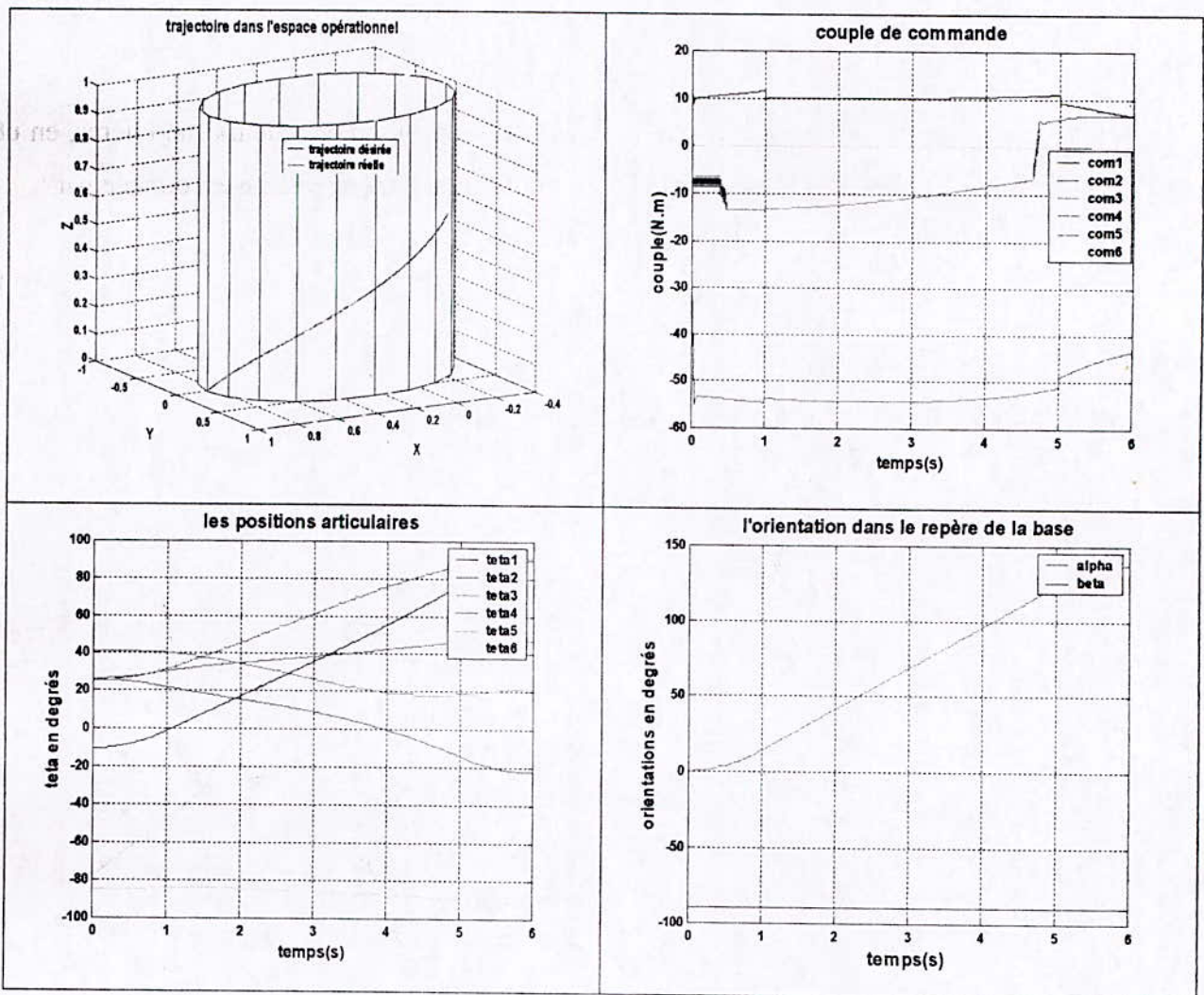


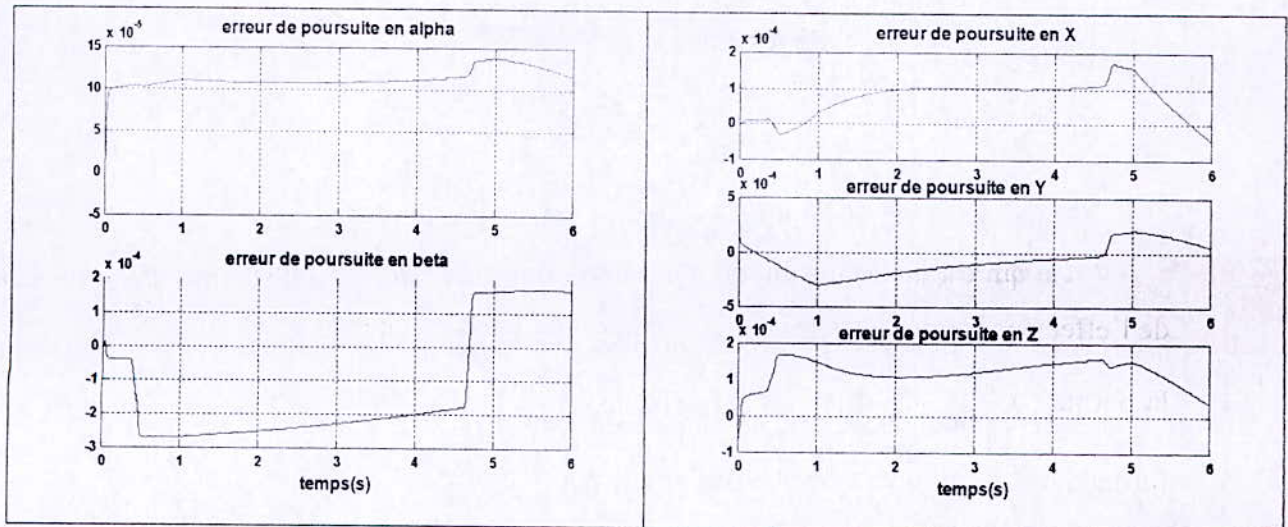
**Remarque :**

La solution (3.58) minimise le résidus donner par la norme  $\|\dot{X}-J\cdot\dot{q}\|$  ce qui signifie physiquement que la vitesse sera près de la vitesse désirée. La minimisation du résidus ne garantit pas la solution unique, mais la solution de pseudo inverse réduit également au minimum la norme de la solution  $\|\dot{q}\|$ , [MAC 90].

**3.2.1. Résultats et simulation :**

• **Dans l'espace :**





**Figure 3.5.** Résultats de simulation en utilisant la décomposition en valeurs singulières de la matrice jacobienne (SVD)

- D'après la simulation, on conclut que la décomposition en valeurs singulières (SVD) résolve le problème de la cinématique inverse.

### 3.3. Méthode des moindres carrés atténuée :

La méthode des moindres carrés atténuée a été utilisée pour résoudre les équations de mal conditionnées. Le critère des moindres carrés atténué est basé sur l'existence d'une solution qui réduit au minimum la somme des normes [MAC 90] :

$$\|\dot{X} - J \cdot \dot{q}\|^2 + \lambda^2 \cdot \|\dot{q}\|^2 \quad (3.59)$$

Où  $\lambda$  est considéré comme facteur d'atténuation, ceci a comme conséquence d'augmenter le système d'équations :

$$\begin{bmatrix} J \\ \lambda \cdot I \end{bmatrix} \cdot \dot{q} = \begin{bmatrix} \dot{X} \\ 0 \end{bmatrix} \quad (3.60)$$

la solution peut être obtenue en résolvant l'équation :

$$(J^T \cdot J + \lambda^2 \cdot I) \cdot \dot{q} = J^T \cdot \dot{X} \quad (3.61)$$

La solution des moindres carrés atténuée est sous la forme :

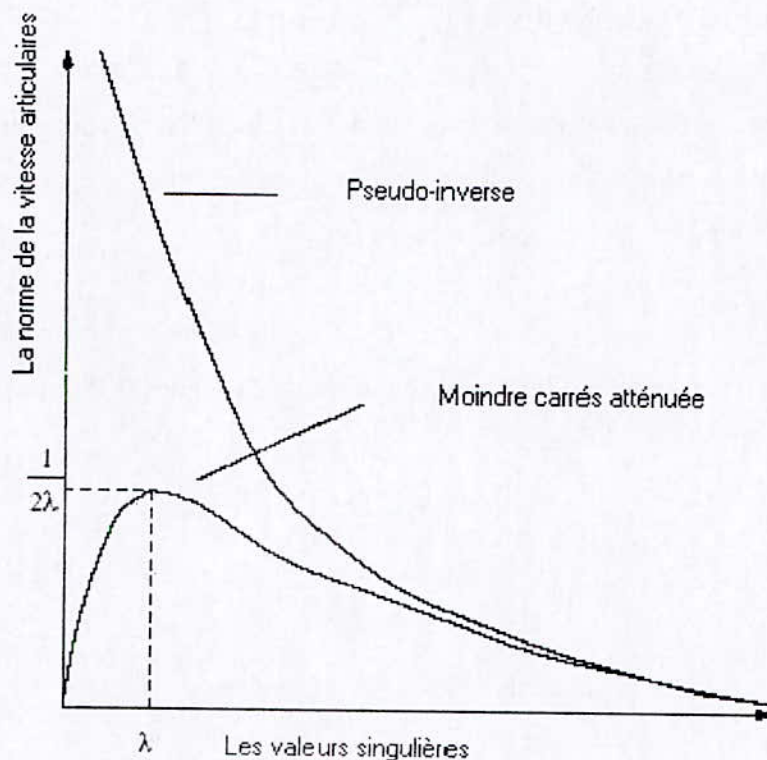
$$\dot{q}^{(\lambda)} = (J^T \cdot J + \lambda^2 \cdot I)^{-1} \cdot J^T \cdot \dot{X} = \sum_{i=1}^r \frac{\sigma_i}{\sigma_i^2 + \lambda^2} \cdot v_i \cdot u_i^T \cdot \dot{X} \quad (3.62)$$

Ce qui est la seule solution réalisant la trajectoire la plus proche de la trajectoire désirée de l'effecteur parmi toutes les combinaisons possibles des vitesses articulaires qui n'excèdent pas la norme  $\|\dot{q}^{(\lambda)}\|$ . De ceci nous pouvons voir que la pseudo inverse est un cas particulier de la formulation des moindres carrés atténuée avec  $\lambda=0$ .

De plus on peut écrire la pseudo inverse sous la forme suivante :

$$J^+ = \sum_{i=1}^r \frac{\sigma_i}{\sigma_i^2 + \lambda^2} \cdot v_i \cdot u_i^T \quad (3.63)$$

La solution des moindres carrés atténuée peut être considérée en fonction des valeurs singulières comme représentées à la figure (3.6).



**Figure 3.6.** Comparaison entre la solution des moindres carrés atténuée et la solution des moindres carrés normale.

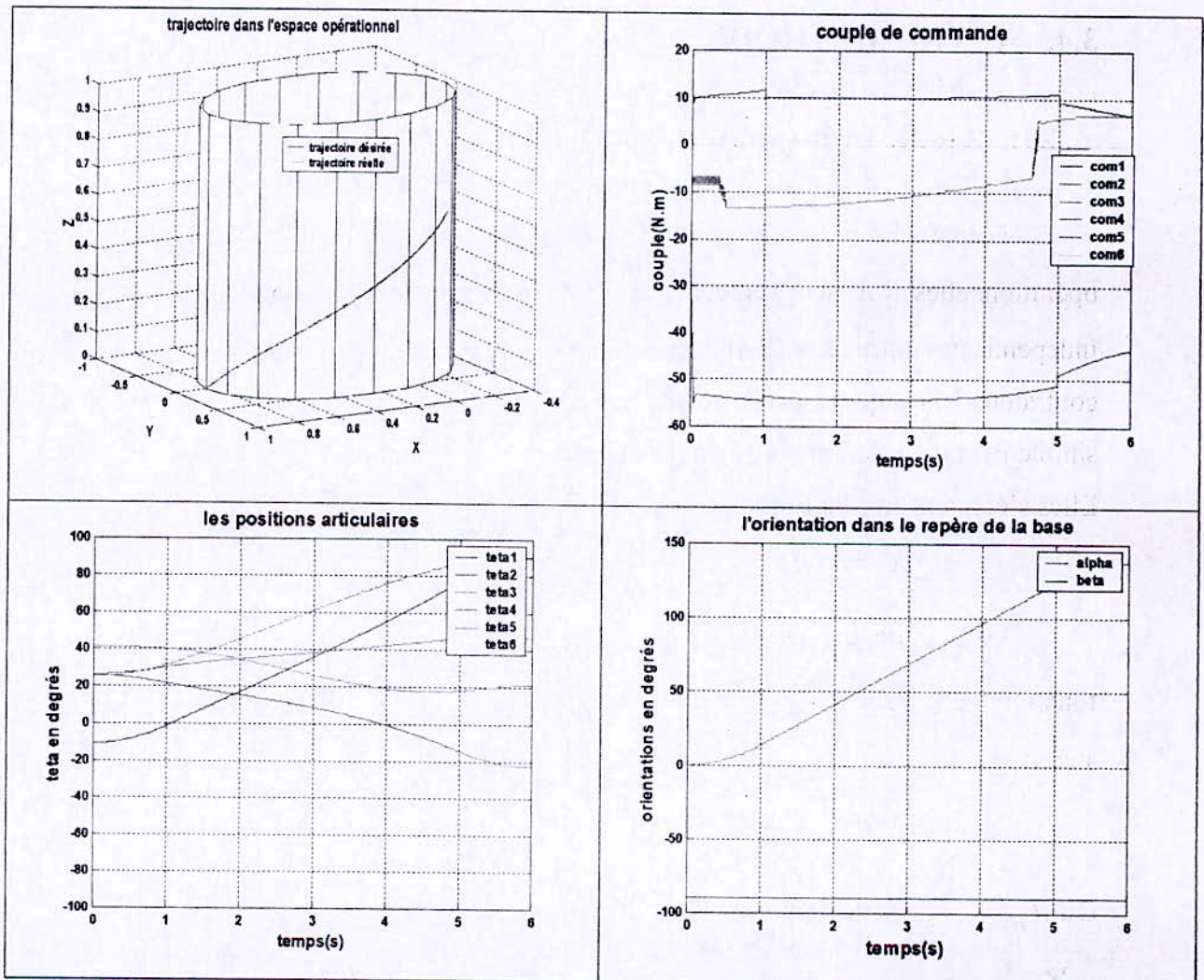
Si une valeur singulière est beaucoup plus grande que le facteur d'atténuation, alors la formulation des moindres carrés atténuée a peu d'effet, parce que :

$$\frac{\sigma_i}{\sigma_i^2 + \lambda^2} \approx \frac{1}{\sigma_i}$$

Ce qui est identique à la solution obtenue en utilisant la pseudo inverse.

### 3.3.1. Résultats et simulation :

- Dans l'espace :



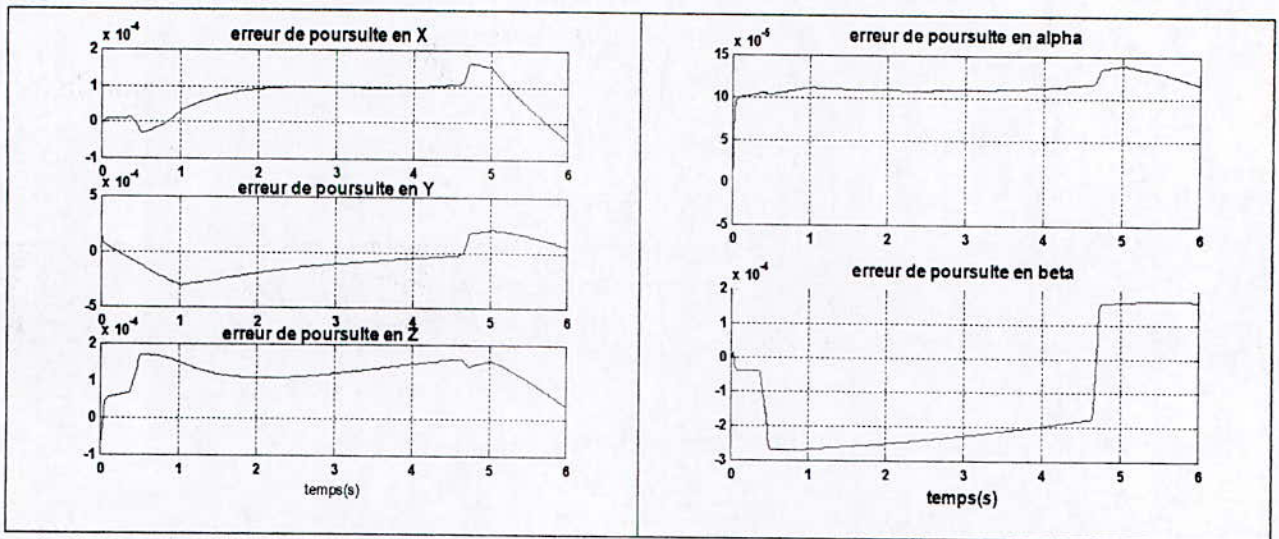


Figure 3.7. Résultats de simulation en utilisant la méthode des moindres carrés atténués

### 3.4. AUTRE METHODE :

#### 3.4.1. Ajoute (n-m) contraintes supplémentaires :

L'approche présentée dans ce paragraphe consiste à ajouter au vecteur de coordonnées opérationnelles  $X$  un vecteur de  $(n - m)$  coordonnées supplémentaires linéairement indépendantes entre elles [BAI 86], [CHA 86], [NEN 92]. Ces relations peuvent traduire soit des contraintes physiques sur le robot, soit des contraintes liées à son environnement ou tout simplement des relations entre différentes positions articulaires du robot.

Elles s'écrivent sous la forme générale suivante :

$$X_c = h_c(q) \tag{3.64}$$

Dans cette expression  $X_c$  est un vecteur de dimension  $(n - m) \times 1$  et  $h_c(q)$  est une fonction vectorielle en  $q$ . Le modèle cinématique associé à ces équations supplémentaires s'écrit :

$$\dot{X}_c = J_H \dot{q} \tag{3.65}$$

Où  $J_H = \frac{\partial h(q)}{\partial q}$  de dimension  $(n - m) \times n$  représente le jacobien de  $h_c(q)$ .

a partir du modèle cinématique directe du robot et de l'équation (3.65), on définit un jacobien augmenté  $J_a$  de dimension  $(n \times n)$  et un vecteur de vitesses  $\dot{X}_a$  tels que :

$$\dot{X}_a = J_a \dot{q} \quad (3.66)$$

Avec :

$$\dot{X}_a = \begin{bmatrix} \dot{X} \\ \dot{X}_c \end{bmatrix} \quad \text{et} \quad J_a = \begin{bmatrix} J \\ J_H \end{bmatrix}$$

Si le jacobien augmenté  $J_a$  n'est pas singulier, la vitesse articulaire  $\dot{q}$  du robot est déterminée comme dans le cas des robots non redondants.

Cette méthode fondée sur le jacobien augmenté présente les inconvénients suivants :

- i) le choix des relations supplémentaires est important.
- ii) le jacobien augmenté  $J_a$  peut être non inversible même si le robot est loin d'une configuration singulière ( $\text{rang}(J) = m$ ). Le fait d'ajouter  $(n - m)$  relations supplémentaires peut conduire à des singularités autres que celles du robot que l'on qualifiera de singularités algorithmique.

Son avantage est lié au quelle permet d'obtenir une réponse articulaire répétable dans le cas ou la trajectoire cartésienne est cyclique ou fermée. En effet, la solution trouvée est unique.

On peut utiliser cette technique pour optimiser une fonction objective désirée  $\phi(q)$  en prenant  $h(q)$  de telle sorte que :

$$h_i(q) = 0 = (\eta_i)^T \nabla \phi(q) \quad \text{Pour } i = 1, \dots, m - n \quad (3.67)$$

Où  $\eta_i$  est un vecteur de la base du noyau de  $J$  et ou  $\nabla \phi(q)$  est le gradient de  $\phi(q)$ .

Le calcul des vecteurs de la base du noyau de la matrice jacobienne devant être effectués de façon analytique, cette méthode ne peut être utilisée que pour des systèmes à faible degré de redondance. Cette difficulté a été récemment résolue par [KLE 95].

### 3.4.1.1 Résultats et simulation :

- Dans l'espace :

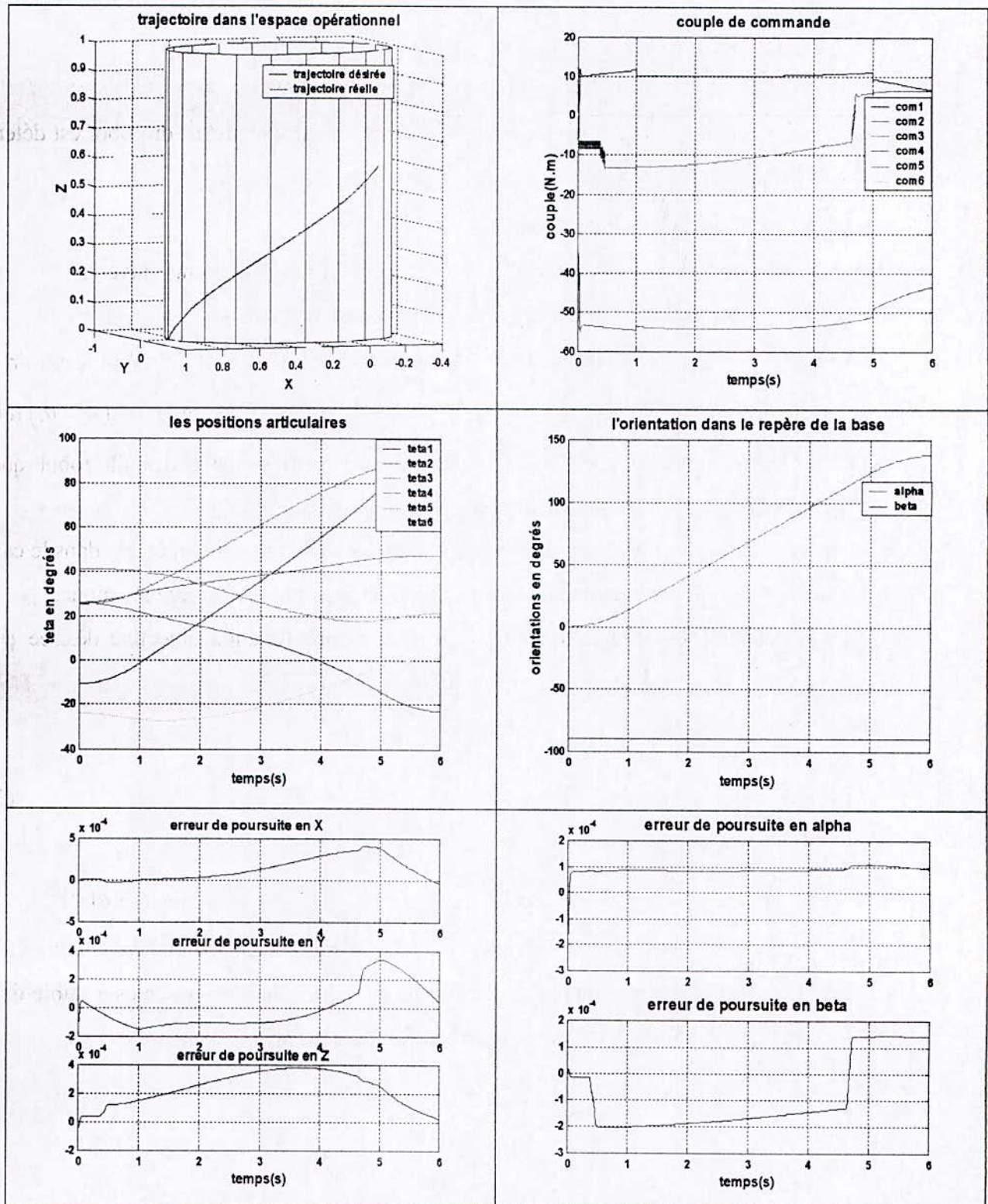


Figure 3.8. Résultats de simulation en ajoutent (n-m) contraintes supplémentaires

- Même constat que pour les méthodes précédentes.

### 3.5. Comparaison entre les méthodes :

Les méthodes basées sur le calcul de la pseudo inverse sont en général rapides et très simple à implémentées, par contre les méthodes qui sont basées sur la pseudo inverse avec un terme d'optimisation, lourdes à implémentées par ce que ces méthodes nécessite le calcul du terme d'optimisation qui est souvent difficile à calculer. Cependant, les méthodes basées sur la décomposition de la matrice jacobienne sont aussi très simple à implémenter sauf que le calcul de la pseudo inverse qui est un peut lent à cause de la décomposition de la matrice jacobienne, par contre les méthodes de la décomposition en SVD sont préférées car elles engendre moins de paramètres et un faible temps de calcul. L'autre méthode comme l'ajoute des contraintes donne des bonnes performances, par ce qu'elle est très simple à implémentées et très rapides.

### 3.6. CONCLUSION :

Nous avons présenté dans ce chapitre les différentes méthodes pour la résolution du problème de la cinématique inverse des robots manipulateurs redondants. Ces méthodes sont basées sur la minimisation d'un critère ou sur la décomposition de la matrice jacobienne, ainsi l'ajoute des contraintes. Les simulations réalisées sur le robot PUMA 560 à 6 degrés de libertés évoluant dans l'espace, montrent l'efficacité de toutes les méthodes présentées.





# Chapitre

## 4

### Evitement d'obstacle

Notre but dans ce chapitre d'appliquer les méthodes proposées par certains articles pour rajouter des objectifs d'évitement d'obstacle dans le cas d'un robot planaire lorsqu'il est redondant. En effet, on rappelle qu'un robot est dit redondant lorsqu'il propose plus de degré de liberté qu'il ne lui nécessaire pour réaliser une tâche donnée. Ainsi notre robot planaire étudié à 3ddl doit suivre une trajectoire  $(x, y)_d$ , donc le nombre de redondance du robot égale à  $n - m = 1$ , donc il est capable d'effectuer des mouvements internes dans le noyau de  $J$ .

On présente les différentes approches pour éviter un obstacle fixe, à savoir l'approche par la tâche prioritaire [MAC 85] et [POU 91], la technique de la fonction de coût [MAR 00], et la méthode de la fonction objective [SAM 90], ainsi la technique de gradient de la distance [SAI 04]. Ces techniques sont alors développées ci-après.

Le modèle cinématique direct est :

$$\dot{X} = J \cdot \dot{q} \quad (4.1)$$

Si on utilise la pseudo inverse comme solution, le modèle cinématique inverse est donc :

$$\dot{q} = J^+ \cdot \dot{X} \quad (4.2)$$

Avec :

$$J^+ = J^T \cdot (J \cdot J^T)^{-1}$$

La forme généralisée du modèle cinématique inverse peut s'écrire de la forme :

$$\dot{q} = J^+ \cdot \dot{X} + (I - J^+ \cdot J) \cdot V \quad (4.3)$$

Où  $V$  est un vecteur qui s'applique aux mobilités internes du robot. Ainsi les variations de  $V$  ne modifie pas la tâche principale  $\dot{X}$ .

#### 4.1. MODELISATION DE L'OBSTACLE :

Sur la base de la figure (4.1), on présente le calcul de la distance entre le centre de l'obstacle et le premier segment du manipulateur de longueur  $l_1$ , [BEN 03].

Avec :

$$\begin{cases} \overline{OA} = \overline{v_1} \\ \overline{OD} = \overline{v_2} \\ \overline{AC} = R \\ \overline{AB} = d \end{cases}$$

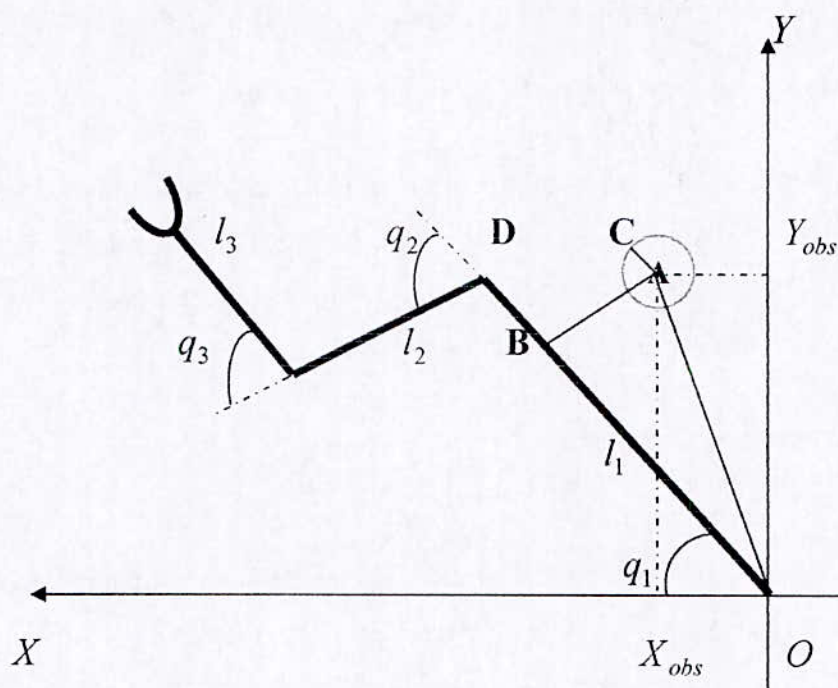


Figure 4.1. Un exemple d'évitement d'obstacle.

On a les relations suivantes :

$$\overline{v_1} = \begin{bmatrix} X_{obs} \\ Y_{obs} \end{bmatrix}, \quad \overline{v_2} = \begin{bmatrix} X_d = l_1 \cdot \cos(q_1) \\ Y_d = l_1 \cdot \sin(q_1) \end{bmatrix}$$

Avec :

$$\overline{v_1} \cdot \overline{v_2} = |\overline{v_1}| \cdot |\overline{v_2}| \cdot \cos(\alpha)$$

D'où nous tirons :

$$\cos(\alpha) = \frac{\overline{v_1} \cdot \overline{v_2}}{|\overline{v_1}| \cdot |\overline{v_2}|}, \text{ et } \sin(\alpha) = \frac{d}{v_1}$$

Ce qui nous permettons d'écrire :

$$\frac{d^2}{v_1^2} + \frac{(\overline{v_1} \cdot \overline{v_2})^2}{|\overline{v_1}| \cdot |\overline{v_2}|} = 1$$

De même il est possible d'écrire :

$$d^2 = v_1^2 \cdot \left[ 1 - \frac{(\overline{v_1} \cdot \overline{v_2})^2}{|\overline{v_1}| \cdot |\overline{v_2}|} \right] \text{ Si } |\overline{v_1}| \cdot |\overline{v_2}| < |\overline{v_2}|$$

Avec :

$$v_1^2 = X_{obs}^2 + Y_{obs}^2$$

$$v_2^2 = l_1^2$$

$$\overline{v_1} \cdot \overline{v_2} = X_{obs} \cdot l_1 \cdot \cos(q_1) + Y_{obs} \cdot l_1 \cdot \sin(q_1)$$

La contrainte d'inégalité s'écrit comme suit:

$$d^2 = X_{obs}^2 + Y_{obs}^2 - (X_{obs} \cdot \cos(q_1) + Y_{obs} \cdot \sin(q_1))^2 \geq R^2 \quad (4.4)$$

La même démarche peut être appliquée pour calculer les autres distances entre l'obstacle et les autres segments  $l_i$  avec  $i = 1, 2, 3$ .

## 4.2. METHODES:

### 4.2.1. Evitement d'obstacle par l'approche de tâche prioritaire :

Soit  $X_e$  le vecteur de position opérationnel de l'effecteur et  $X_0$  le vecteur de position de point du robot le plus proche de l'obstacle [SAI 04].

On a :

$$\dot{X}_e = J_e \cdot \dot{q} \quad (4.5)$$

$$\dot{X}_0 = J_0 \cdot \dot{q} \quad (4.6)$$

Où :  $J_e$  est la matrice jacobienne liant le vecteur de configuration  $\dot{q}$  à  $\dot{X}_e$

$J_0$  est la matrice jacobienne liant  $\dot{q}$  à  $\dot{X}_0$ , la solution par la pseudo inverse de l'équation (4.5) s'écrit :

$$\dot{q} = J_e^+ \cdot \dot{X}_e + (I - J_e^+ \cdot J_e) Z \quad (4.7)$$

Où  $Z$  est un vecteur arbitraire de  $\mathfrak{R}^n$ .

En multipliant (4.7) par  $J_0$  à gauche :

$$\dot{X}_0 = J_0 \cdot J_e^+ \cdot \dot{X}_e + J_0 \cdot (I - J_e^+ \cdot J_e) Z \quad (4.8)$$

On en déduit :

$$Z = [J_0 \cdot (I - J_e^+ \cdot J_e)]^\dagger \cdot (\dot{X}_0 - J_0 \cdot J_e^+ \cdot \dot{X}_e) \quad (4.9)$$

Puis :

$$\dot{q} = J_e^+ \cdot \dot{X}_e + [J_0 \cdot (I - J_e^+ \cdot J_e)]^\dagger \cdot (\dot{X}_0 - J_0 \cdot J_e^+ \cdot \dot{X}_e) \quad (4.10)$$

En introduisant des pondérations sur la mobilité interne pour que l'influence de l'obstacle se limite à une certaine distance, et évolue de façon continue.

On obtient :

$$\dot{q} = J_e^+ \cdot \dot{X}_e + \alpha_h \cdot [J_0 \cdot (I - J_e^+ \cdot J_e)]^\dagger \cdot (\alpha_0 \cdot \dot{X}_0 - J_0 \cdot J_e^+ \cdot \dot{X}_e) \quad (4.11)$$

Le choix de  $\alpha_h$  est donné sur la figure (4.2) suivante :

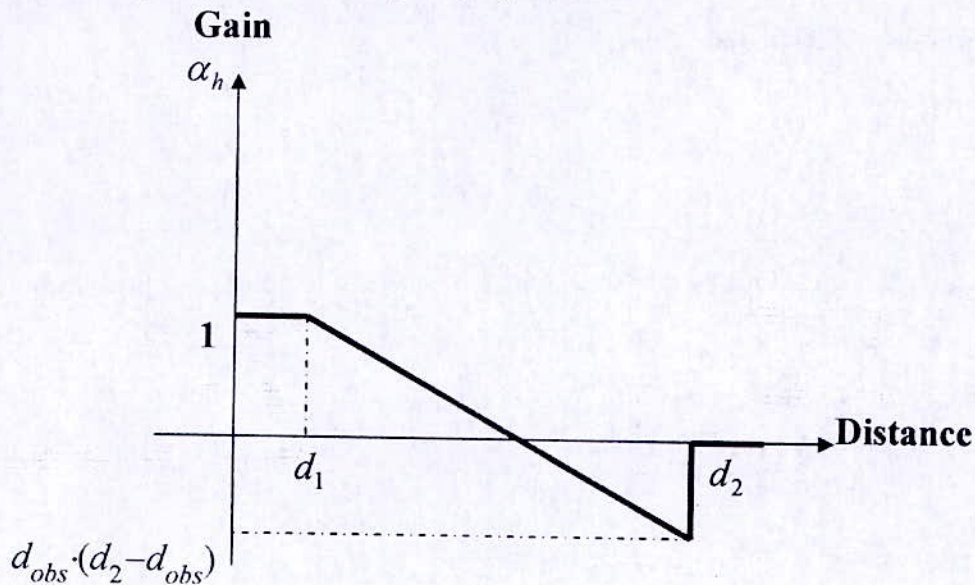


Figure 4.2. Choix de  $\alpha_h$ .

Cette solution est telle que :

- L'objectif  $\dot{X}_e = J_e \cdot \dot{q}$  est prioritaire et toujours réalisé. Il détermine la trajectoire de l'extrémité du manipulateur.
- L'objectif  $\dot{X}_o = J_o \cdot \dot{q}$  est secondaire, il est déterminé grâce aux mobilités internes du manipulateur, via le vecteur  $Z$ .

### Remarque :

L'algorithme implémenté pour réaliser l'évitement d'obstacle fixe en utilisant cette technique, est donné en Annexe B.

### 4.2.2. Evitement d'obstacle par l'approche fonction de coût :

On peut considérer une fonction de coût pour éviter des obstacles fixes [MAR 00], la fonction de coût est définie telle que :

$$h_s = \frac{1}{d^2} \quad (4.12)$$

Où  $d$  est la distance critique (la distance entre l'obstacle et le point critique du robot).

Le vecteur arbitraire  $Z$  est défini suivant :

$$Z = -\alpha \cdot h_s^2 \cdot (\delta x, \delta y, \delta z, 0, \dots, 0)^T \quad (4.13)$$

En remplace le vecteur  $V$  de l'équation (4.3) par le vecteur  $Z$ , on obtient :

$$\dot{q} = J_e^+ \cdot \dot{X} + \alpha \cdot (I - J^+ \cdot J) \cdot h_s^2 \cdot (\delta x, \delta y, \delta z, 0, \dots, 0)^T \quad (4.14)$$

Avec  $\alpha$  facteur positif, et  $(\delta x, \delta y, \delta z)$  est le distance  $X_{obs} - X_0$ .

Malheureusement, la technique de la fonction de coût n'a aucune interprétation physique claire ; en outre, elle emploie seulement les trois premières colonnes de l'opérateur de projection  $(I - J^+ \cdot J)$ , ceci rend cette technique très difficile à généraliser pour éviter des obstacles multiples.

#### Remarque :

L'algorithme implémenté pour réaliser l'évitement d'obstacle fixe en utilisant cette technique, est représenté en détail en Annexe B.

### 4.2.3. Evitement d'obstacle par gradient de la distance :

Soit  $X_e$  le vecteur position de l'effecteur et  $\delta q_0$  les mouvements internes du robot pour éviter l'obstacle [SAI 04].

On a :

$$\dot{X}_e = J_e \cdot \dot{q} \quad (4.15)$$

$$\delta q_0 = (I - J_e^+ \cdot J_e) f \quad (4.16)$$

Où :  $J_e$  est la matrice jacobienne liant le vecteur de configuration  $\dot{q}$  à  $\dot{X}_e$

$(I - J_e^+ \cdot J_e)$  évolue dans le noyau du mouvement liant  $f$  à  $\delta q_0$ .

La solution de l'équation (4.15) s'écrit :

$$\dot{q} = J_e^+ \cdot \dot{X}_e + (I - J_e^+ \cdot J_e) f \quad (4.17)$$

Où  $\delta f$  est un gradient dont le maximum tend vers l'objectif d'évitement d'obstacle.

Ce gradient se calcul de la façon suivante :

$$f = \min[p(q) - o] \quad (4.18)$$

Où  $p(q)$  est la position en fonction des variables  $q$  d'un point du robot et  $o$  le point d'obstacle le plus proche.

On en déduit pour notre robot composé que par des articulations rotoïdes de centre  $O_i$  :

$$\frac{\delta f}{\delta q} = -(y_j - y_i) \cdot (x_i - x_{obs}) + (x_j - x_i) \cdot (y_i - y_{obs}) \quad (4.19)$$

Où  $(x_j, y_j)$  sont les coordonnées du point du robot le plus proche de l'obstacle  $(x_{obs}, y_{obs})$  les coordonnées de l'obstacle considéré et  $(x_i, y_i)$  les coordonnées des centres des articulations  $O_i$ .

Remarquons que pour la simulation nous prendrons  $O_j$  le centre de rotation le plus proche. Cette approximation du gradient permet de simplifier les démonstrations et les calculs :

On pose :

$$\text{grad}(f) = \frac{\delta f}{\delta q_i} \quad (4.20)$$

D'où :

$$\dot{q} = J_e^+ \cdot \dot{X}_e + (I - J_e^+ \cdot J) \frac{\delta f}{\delta q_i} \quad (4.21)$$

la solution est telle que :

- l'objectif  $\dot{X}_e = J_e \cdot \dot{q}$  est prioritaire, et toujours réalité. Il détermine la trajectoire de l'extrémité du manipulateur.
- L'objectif  $(I - J_e^+ J) \frac{\delta f}{\delta q_i} = \dot{q}_0$  est secondaire, il est déterminé grâce aux mobilités

internes du manipulateur, via le gradient  $\frac{\delta f}{\delta q_i}$ .

C'est la solution (4.21) que nous avons implémentée.

### Remarque :

- Pour la démonstration de cette méthode voir Annexe A.
- L'algorithme implémenté pour réaliser l'évitement d'obstacle fixe en utilisant cette technique, est exploité en Annexe B.



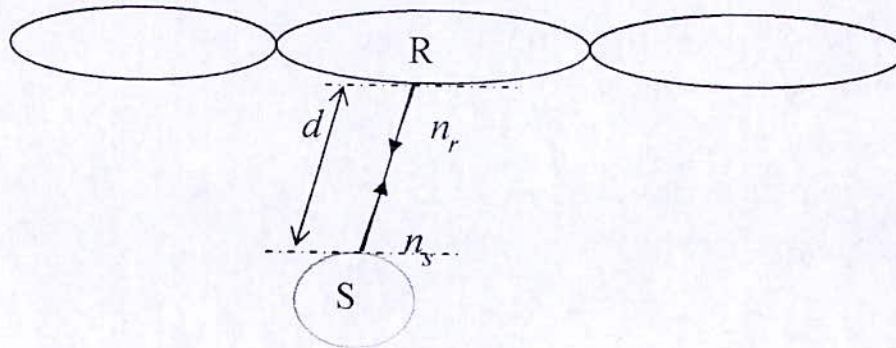
#### 4.2.4. Evitement d'obstacle par le gradient d'une fonction objective :

Une autre méthode pour le calcul de  $Z$ , pour assurer des objectifs d'évitement d'obstacle, est basée sur le gradient d'une fonction objective  $P(q,t)$  telle que [TER 00]:

$$\dot{q} = J_e^+ \cdot \dot{X} + \alpha \cdot (I - J_e^+ \cdot J_e) \frac{\partial P}{\partial q} \quad (4.22)$$

Où  $\alpha$  doit être un gain positif pour maximiser  $P$ , ou bien un gain négatif pour minimiser  $P$ .

La fonction objective  $P$  est définie selon le critère secondaire désiré. Pour réaliser l'évitement d'obstacle, la fonction  $P$  peut être définie pour maximiser la distance entre l'obstacle et le robot manipulateur.



**Figure 4.3.** Utilisation de la méthode gradient d'une fonction objective pour éviter un obstacle fixe.

On peut prendre la fonction objective de la forme suivante selon [SAM 90] :

$$P(q,t) = \lambda \cdot d^{-k}(q,t) \quad (4.23)$$

Avec :

$$\lambda > 0 \text{ et } k \in \mathbb{N}$$

Le gradient de  $P$  est :

$$\frac{\partial P}{\partial q} = -k \cdot \lambda \cdot d^{-(k+1)}(q,t) \cdot \frac{\partial d(q,t)}{\partial q} \quad (4.24)$$

Ceci exige le calcul de  $\frac{\partial d(q,t)}{\partial q}$ , pour le faire nous définissons les vecteurs unitaires  $n_r$  et  $n_s$  suivant les indications de la figure (4.3).

$$n_r = \frac{1}{d(q,t)} (X_S - X_R) \quad (4.25)$$

$$n_s = -n_r$$

Avec :

$X_S$  : le point de contact de l'obstacle.

$X_R$  : le point le plus proche du corps du robot.

On peut écrire la distance critique comme suit :

$$d(q,t) = \langle n_r, X_S - X_R \rangle$$

$$\text{Avec : } \langle A, B \rangle = A^T \cdot B$$

$$\dot{d} = \langle n_r, V_S - V_R \rangle = -\langle n_r, V_R \rangle + \langle n_r, V_S \rangle \quad (4.26)$$

On peut réécrire la vitesse de la tâche secondaire ( $\dot{X}_0 = J_0 \cdot \dot{q}$ ) :

$$V_R = J_0 \cdot \dot{q} \quad (4.27)$$

On rappelle que :

$$\dot{d} = \frac{\partial d}{\partial q} \cdot \dot{q} + \frac{\partial d}{\partial t} \quad (4.28)$$

On remplace  $V_R$  et  $n_r$  par leurs formules dans l'équation (4.28), on obtient :

$$-\left\langle \frac{1}{d(q,t)} \cdot (X_S - X_R) J_0 \cdot \dot{q} \right\rangle + \langle n_r, V_S \rangle = \frac{\partial d}{\partial q} \cdot \dot{q} + \frac{\partial d}{\partial t} \quad (4.29)$$

Par définition :

$$\langle n_r, V_S \rangle = \frac{\partial d}{\partial t}$$

Donc l'équation (4.29) peut être simplifiée comme suite :

$$-\left\langle \frac{1}{d(q,t)} \cdot (X_S - X_R) J_0 \cdot \dot{q} \right\rangle = \frac{\partial d}{\partial q} \cdot \dot{q} \quad (4.30)$$

En remplaçant les vecteurs et la matrice du côté gauche de l'équation (4.30) par leurs éléments, nous obtenons :

$$\begin{aligned} & - \left\langle \frac{1}{d(q,t)} \begin{bmatrix} x \\ y \end{bmatrix} \begin{bmatrix} J_{11} & J_{12} & \dots & J_{1n} \\ J_{21} & J_{22} & \dots & J_{2n} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix} \right\rangle \\ &= \frac{1}{d(q,t)} \left[ x \cdot (J_{11} \cdot \dot{q}_1 + J_{12} \cdot \dot{q}_2 + \dots + J_{1n} \cdot \dot{q}_n) + y \cdot (J_{21} \cdot \dot{q}_1 + J_{22} \cdot \dot{q}_2 + \dots + J_{2n} \cdot \dot{q}_n) \right] \\ &= \left\langle \frac{1}{d(q,t)} \begin{bmatrix} J_{11} & J_{21} \\ J_{12} & J_{22} \\ \vdots & \vdots \\ J_{1n} & J_{2n} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix} \right\rangle \end{aligned}$$

De cette dernière formule on peut écrire que :

$$\frac{\partial d(q,t)}{\partial q} = -\frac{1}{d(q,t)} \cdot J_0^T \cdot (X_S - X_R) \quad (4.31)$$

On remplace (4.31) dans (4.24), finalement le gradient  $P$  devient :

$$\frac{\partial P}{\partial q} = \lambda \cdot k \cdot d^{-(k+2)}(q,t) \cdot J_0^T \cdot (X_S - X_R) \quad (4.32)$$

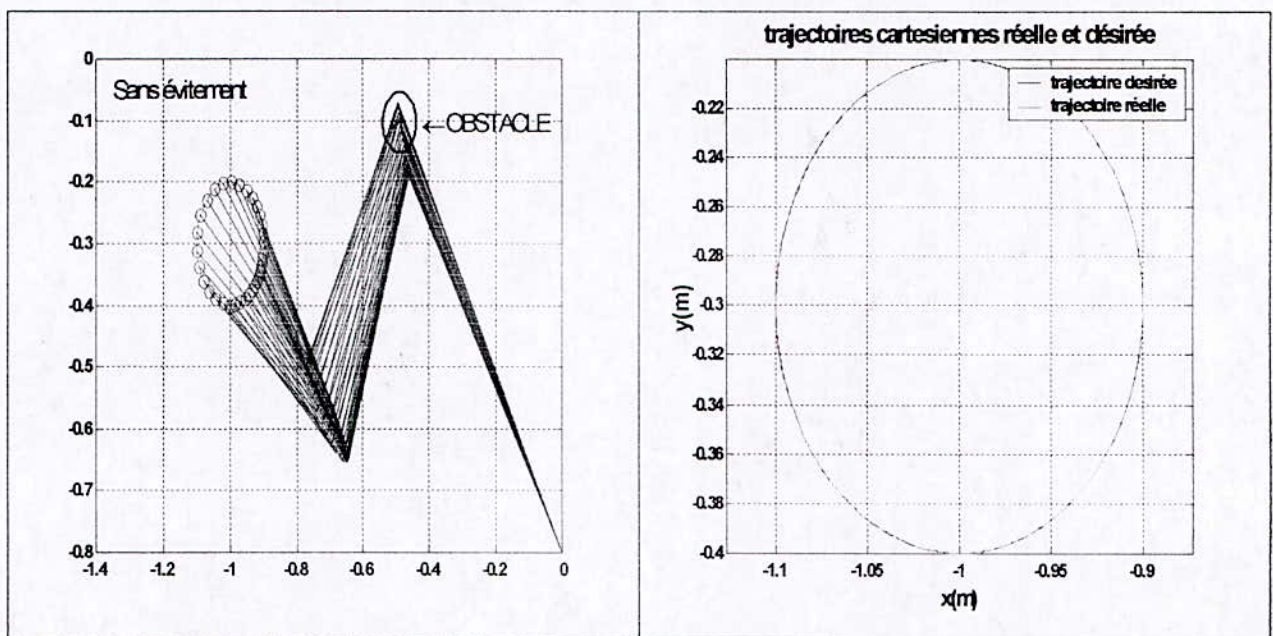
C'est ce dernier résultat qui est implémenté.

### Remarque :

L'algorithme implémenté pour réaliser l'évitement d'obstacle fixe en utilisant cette technique, est exposé en Annexe B.

## 4.3. SIMULATION :

### 4.3.1. Résultats sans évitement :



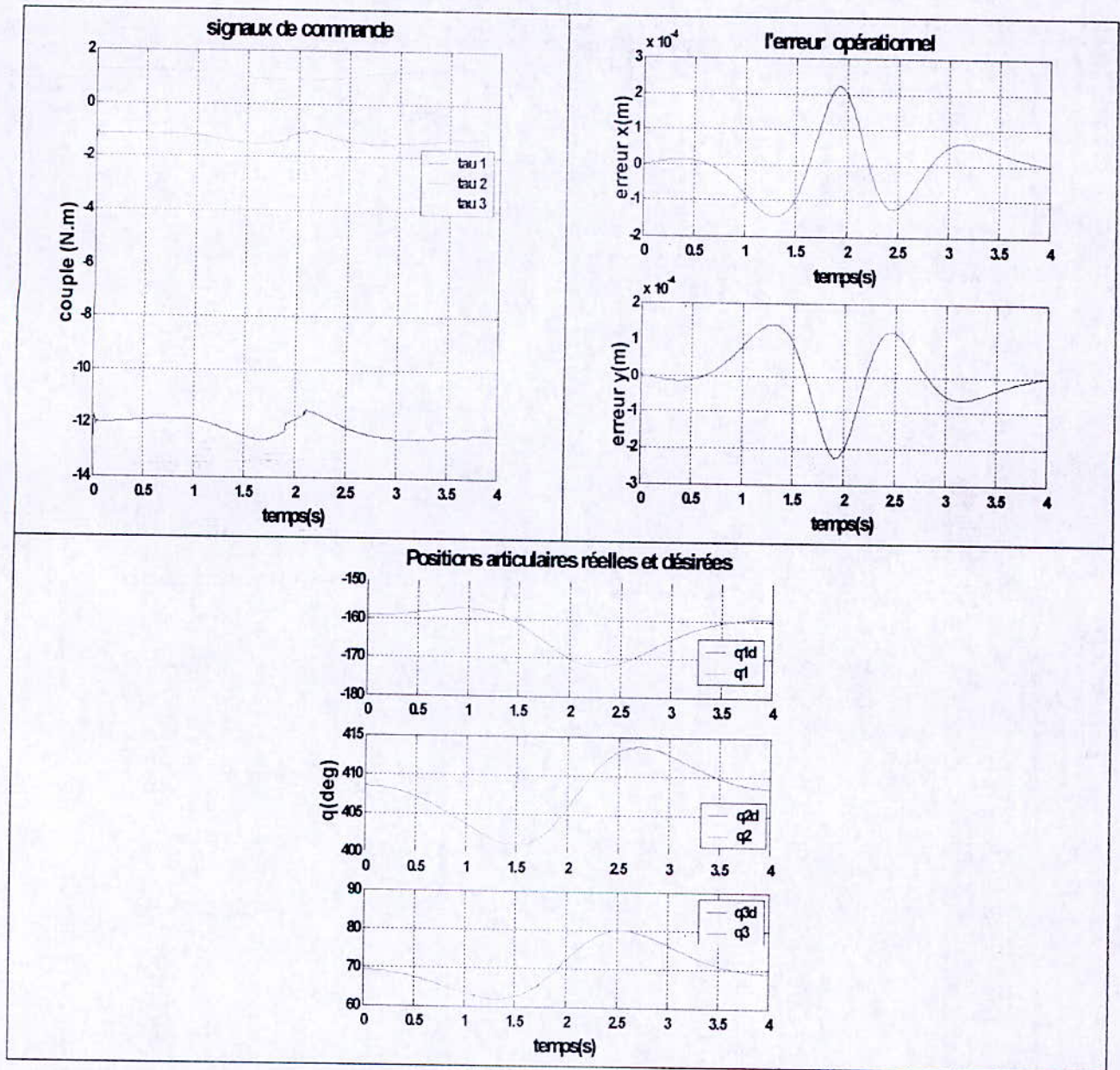
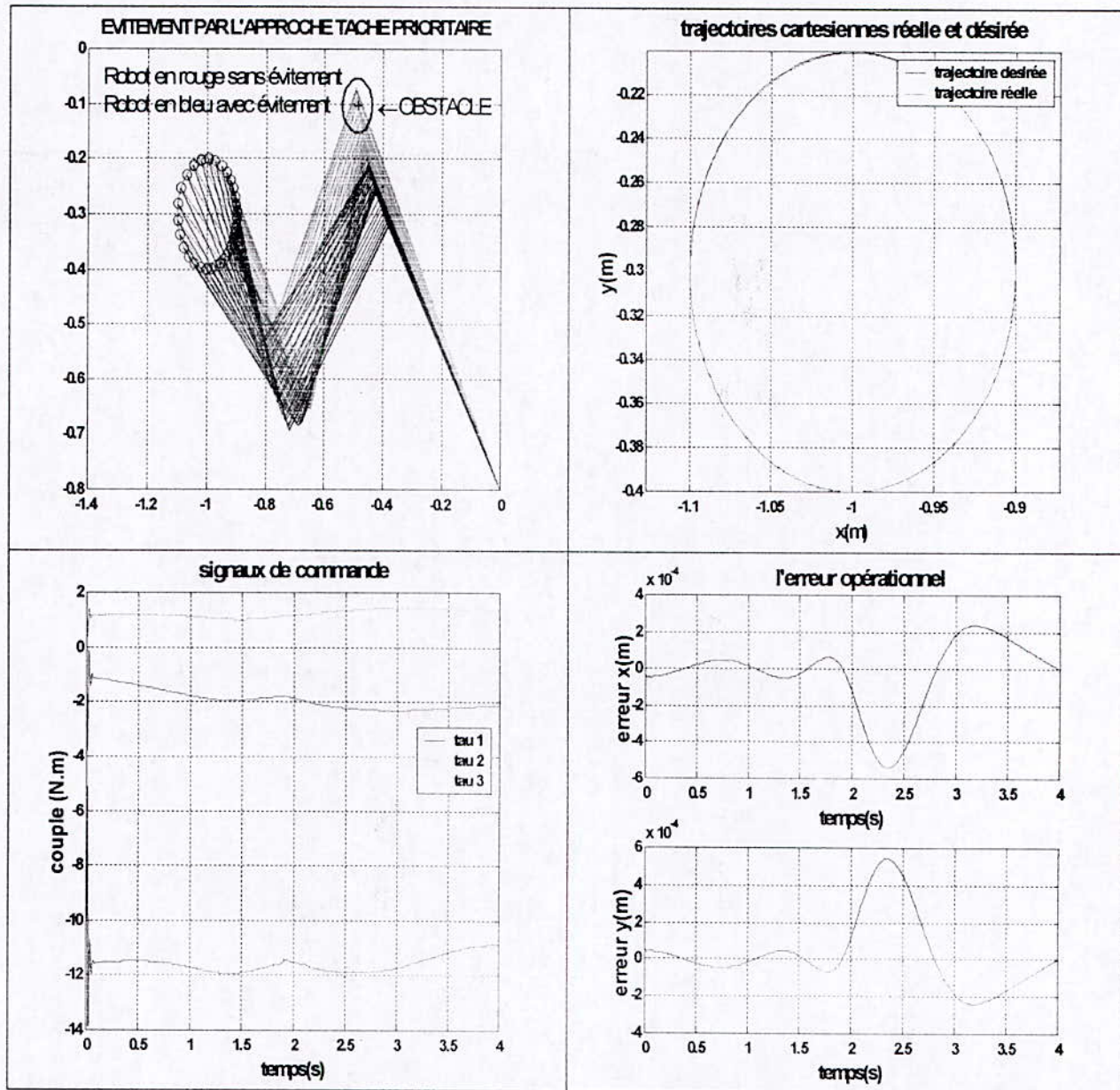


Figure 4.4. Résultats de simulation sans évitement

- On constate que le robot dans le cas sans évitement suit la trajectoire désirée avec une erreur de l'ordre de  $10^{-4}$  et un couple admissible sur les trois articulations (pas de saturation), donc le robot n'évite pas l'obstacle.

### 4.3.2. Résultats en utilisant l'approche tâche prioritaire :



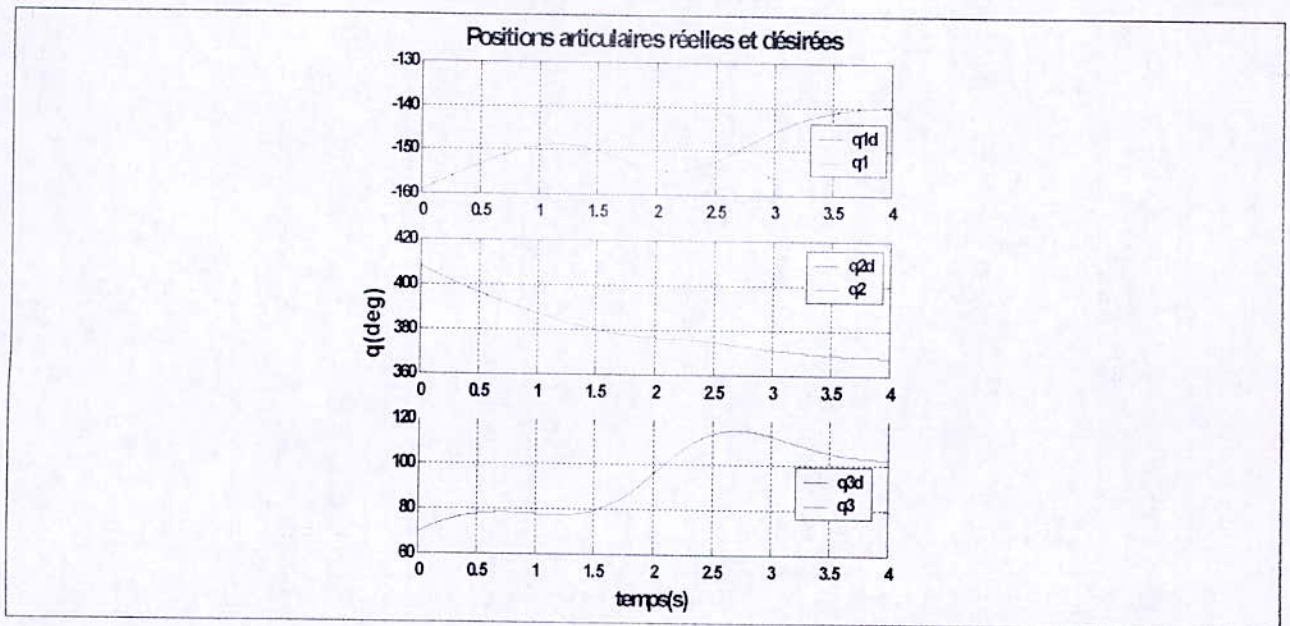
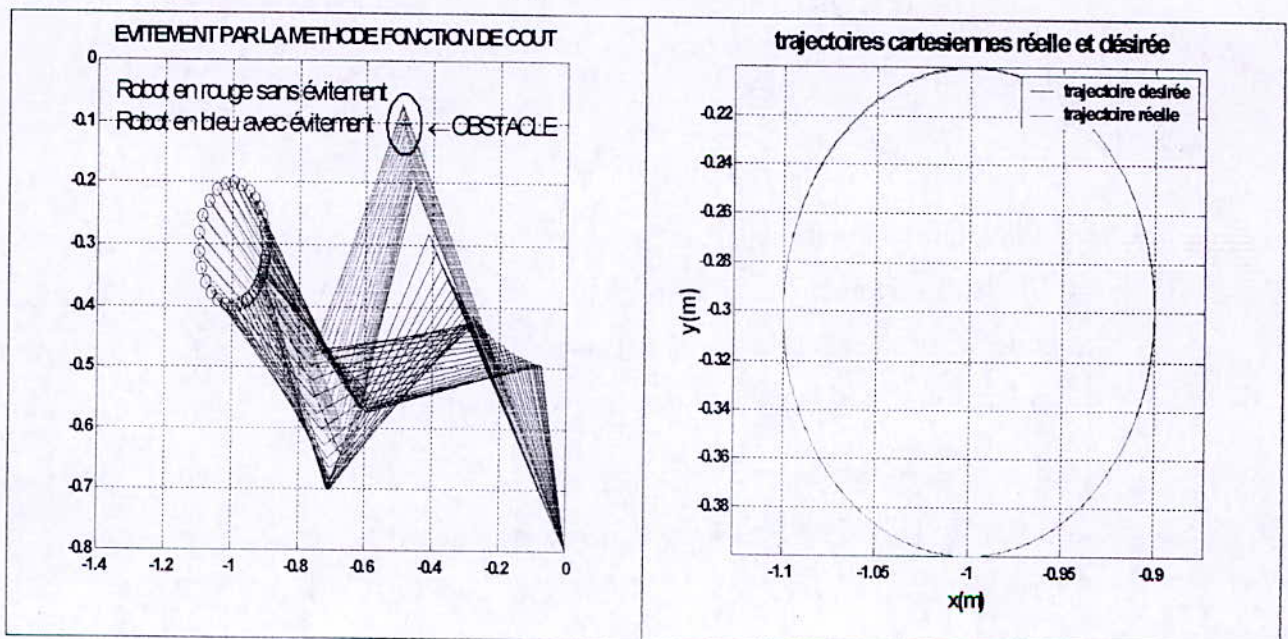


Figure 4.5. Résultats de simulation en utilisant l'approche tâche prioritaire.

- On constate bien que le robot suit la trajectoire désirée dans l'espace opérationnel et évite l'obstacle considéré, avec une erreur opérationnel de l'ordre  $10^{-4}$  et un couple admissible sur les trois articulations du robot car la mobilité interne dans le noyau de  $J_e$  ne modifié pas la vitesse de l'effecteur.

### 4.3.3. Résultats en utilisant l'approche fonction de coût :



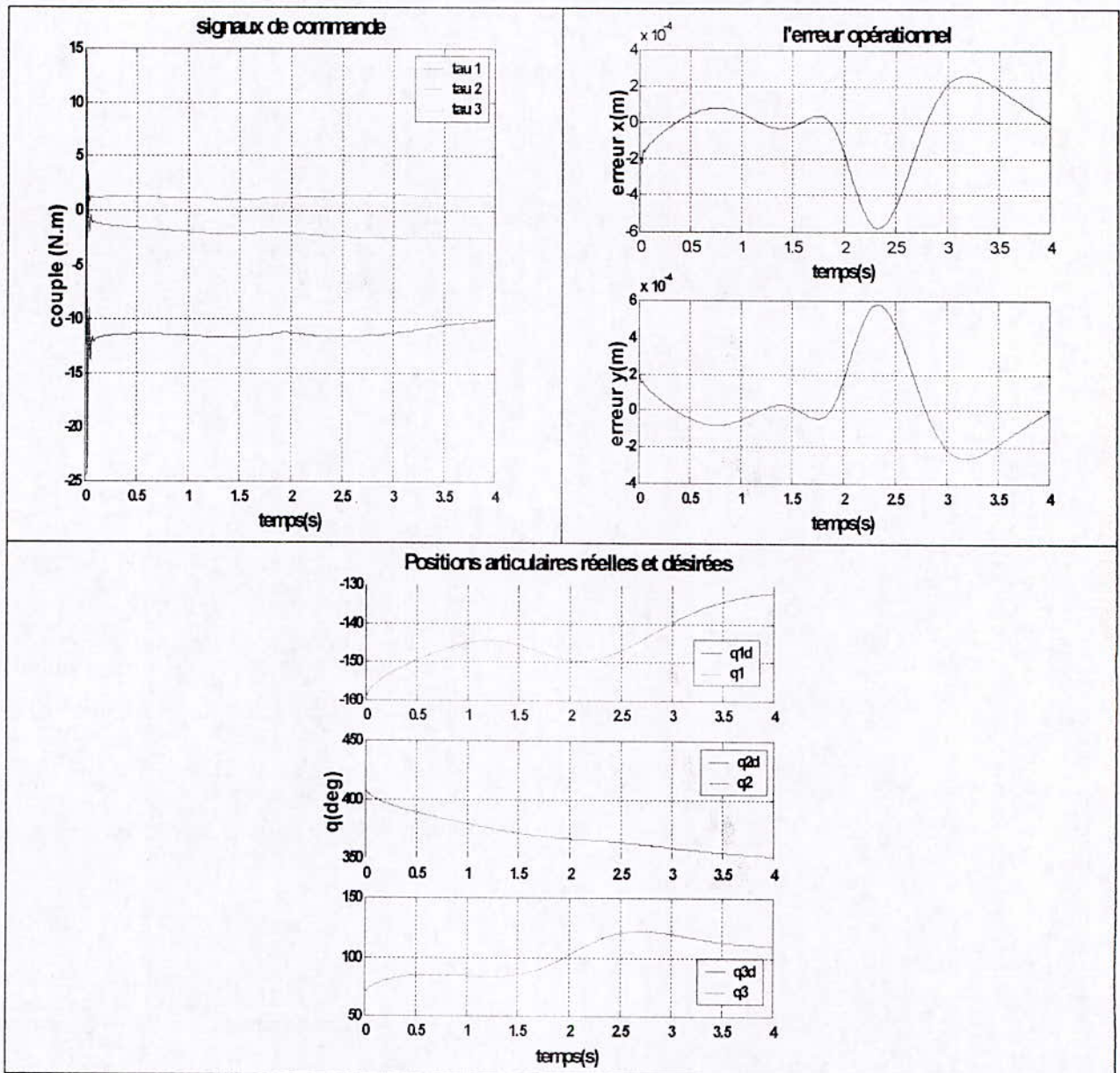
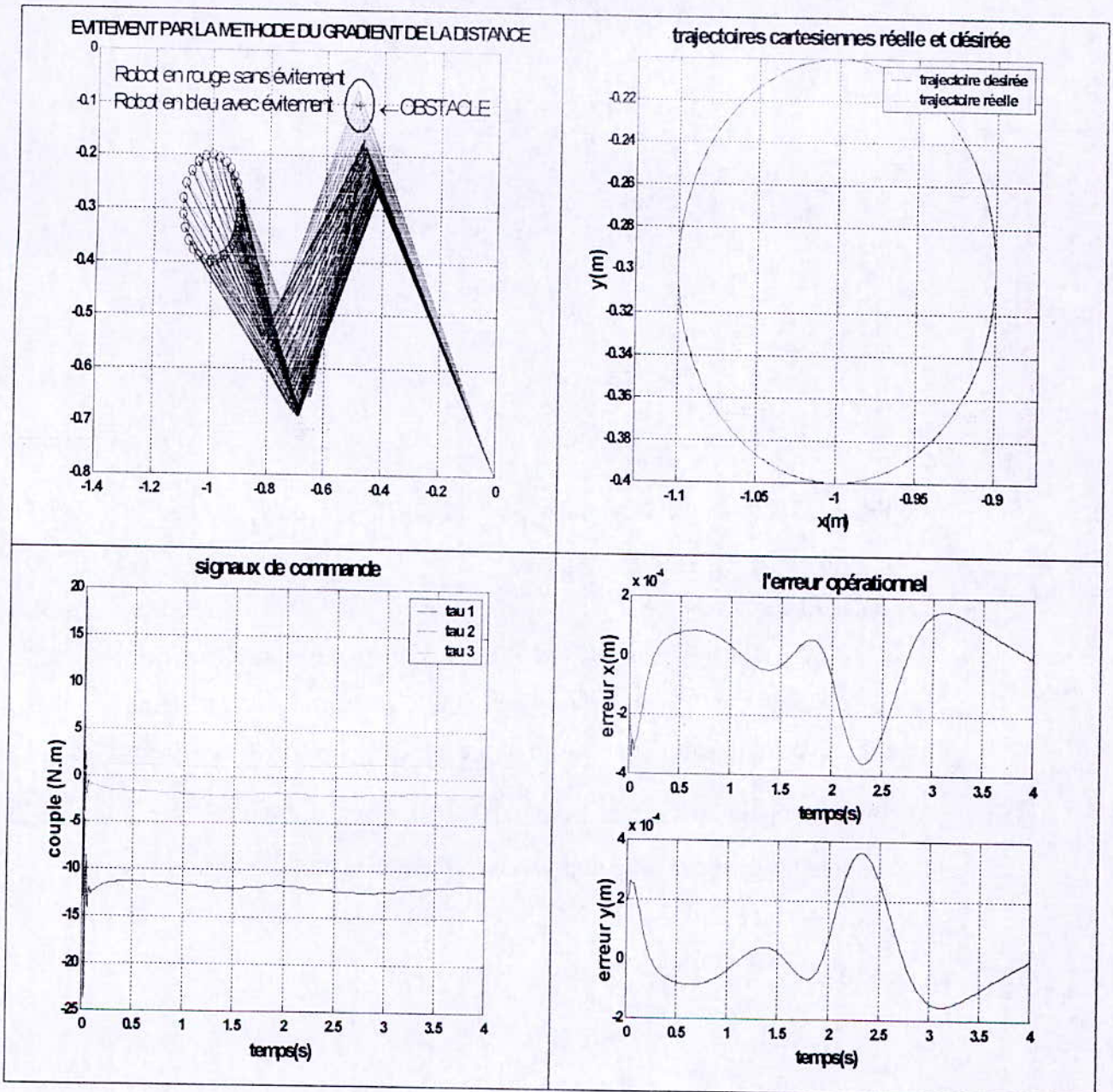


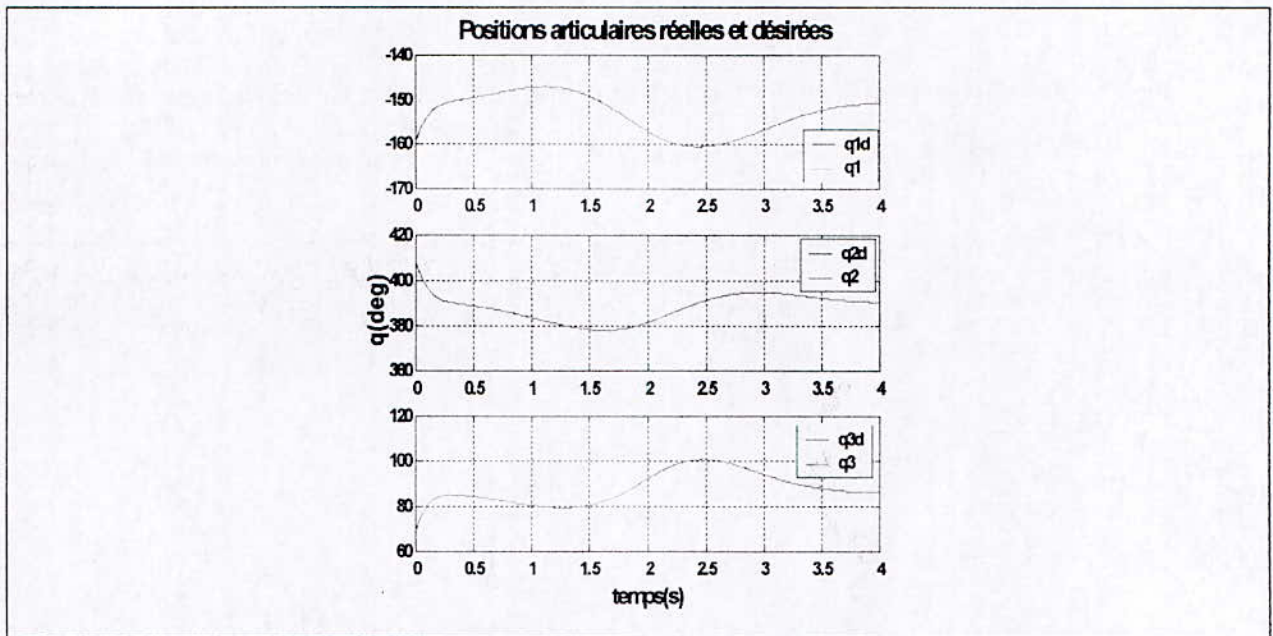
Figure 4.6. Résultats de simulation en utilisant l'approche fonction de coût.

- On constate d'après les résultats de la simulation que le robot suit la trajectoire désirée qui doit être toujours réalisé, en plus de la tâche secondaire d'évitement d'obstacle. On constate aussi que les couples au démarrage sont très grands (non admissible) en raison des saturations. Sans saturation, le robot suit la trajectoire désirée avec un couple admissible et une erreur opérationnelle de l'ordre de  $10^{-4}$  car la mobilité interne dans le noyau de  $J_e$  imposée par cette méthode ne modifie pas la vitesse de l'effecteur.



4.3.4. Résultats en utilisant l'approche gradient de la distance :

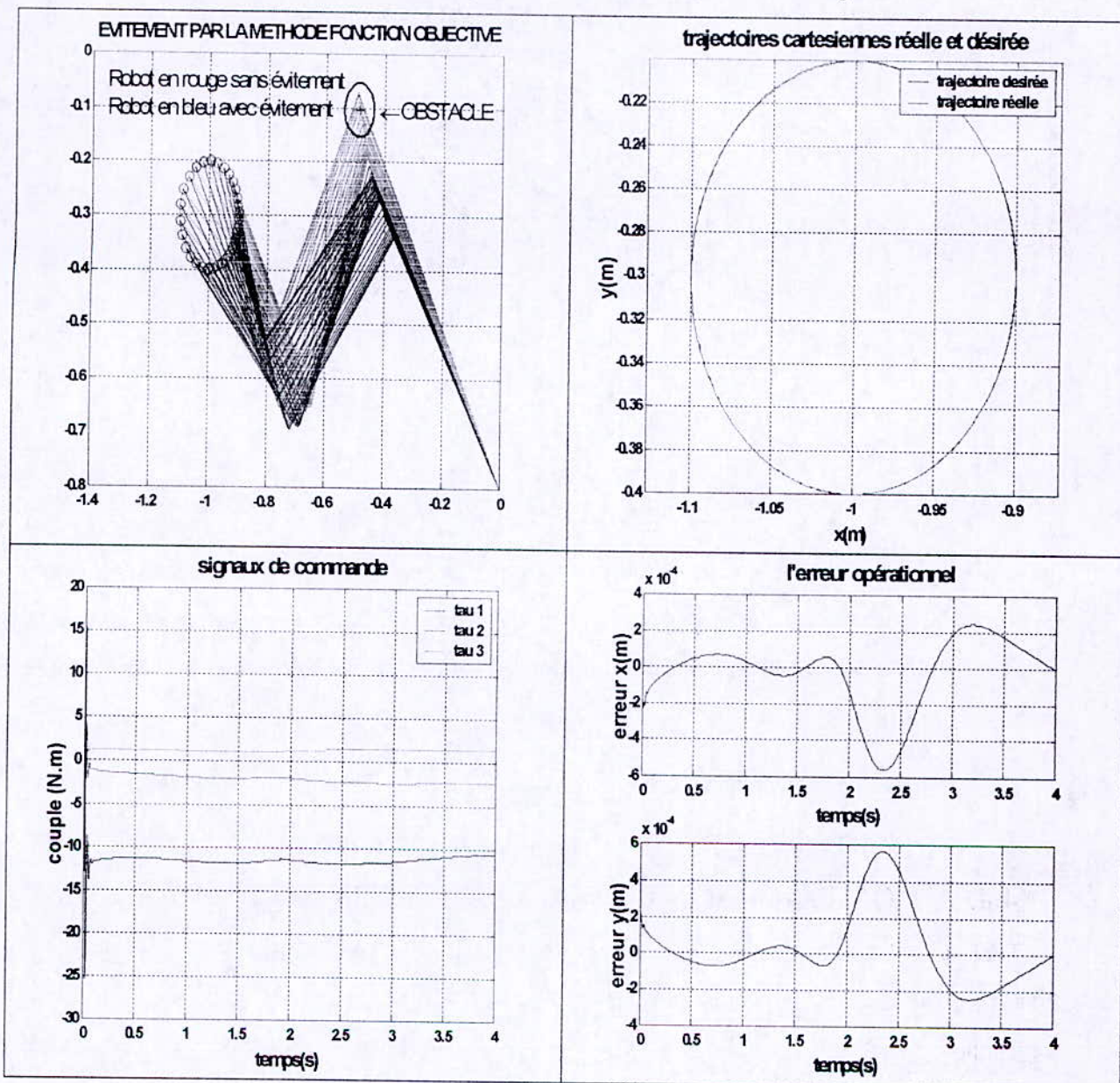


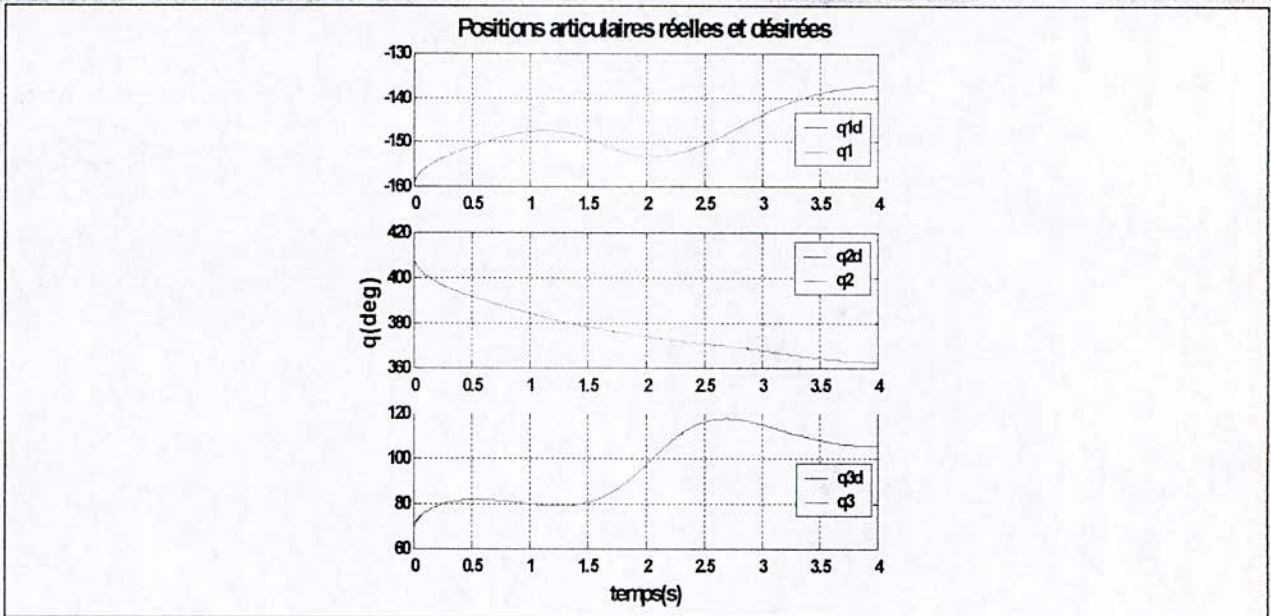


**Figure 4.7.** Résultats de simulation en utilisant l'approche gradient de la distance.

- Les résultats montrent que l'utilisation de l'approche gradient de la distance a permis au robot de suivre la trajectoire désirée (tâche principale) tout en évitant l'obstacle, sauf qu'au démarrage les couples sont grands et cela à cause des saturations, ailleurs (sans saturation) le robot suit la trajectoire désirée et évite l'obstacle considéré avec une erreur de poursuite opérationnelle de l'ordre  $10^{-4}$  parce que la mobilité interne intégrée dans la cinématique inverse par cette approche dans le noyau de  $J_e$  ne modifie pas la vitesse de l'effecteur.

4.3.5. Résultats en utilisant le gradient d'une fonction objective :





**Figure 4.8.** Résultats de simulation en utilisant l’approche gradient d’une fonction objective.

- On constate qu’en utilisant cette approche on a les mêmes performances que les méthodes précédentes et on a le problème de saturation qu’on peut le déduire à partir des couples au démarrage. Sans saturation on remarque que le robot suit la trajectoire désirée plus qu’il évite l’obstacle avec une erreur de poursuite opérationnelle de l’ordre  $10^{-4}$  car la mobilité interne dans le noyau de  $J_e$  n’a aucun effet sur la vitesse de l’effecteur.

**4.4. COMPARAISON DES QUATRE METHODES:**

Techniques	Performance	Paramètres	Interprétation Physique	Extension pour multi obstacles
Tâche prioritaire	lente	$\alpha_0, \alpha_h, d_1, d_2$	Oui	moyenne
Fonction de coût	rapide	$\alpha$	Non	mauvaise
Fonction objective	moyenne	$\lambda, k, \alpha$	Oui	bonne
Gradient de la distance	moyenne	$d_1, d_2, \alpha, \beta$	Oui	bonne

**Tableau 4.1.** Comparaison entre les 4 techniques d’évitement d’obstacle.

#### 4.5. CONCLUSION :

Nous avons étudié dans ce chapitre les solutions pour utiliser les manipulabilités internes d'un robot redondant dans le but d'éviter un obstacle fixe.

Nous avons aussi constaté que le problème principal d'évitement d'obstacle est basé sur la façon de résoudre la cinématique inverse du robot. Suite à l'application des différentes méthodes, nous pouvons dire que celle par le gradient de la distance et celle par la fonction de coût sont les plus intéressantes car elles permettent facilement d'ajouter des objectifs secondaires.

Malheureusement à chaque nouvel objectif nous lui associons un ou plusieurs coefficients de pondérations, le problème consiste à paramétrer ces derniers. Or nous avons constaté que ce réglage dépend de la tâche à accomplir et de la vitesse souhaitée.

D'autres solutions d'optimisation peuvent être envisagées comme les algorithmes génétiques, le filtre de Kalman ou des réseaux de neurones.

De plus dans les critères à ajouter, il serait peut être intéressant de prendre en compte le nombre de degrés de libertés utilisé par l'objectif secondaire et de chercher à réduire ce dernier pour augmenter les chances de résoudre tous les objectifs.

## Conclusion Générale

Bien que la plupart des robots dans l'industrie sont des robots redondants c'est-à-dire le robot accomplir une tâche désirée qui a moins de degrés de libertés que le robot. Etant donné cela on applique dans l'espace opérationnel une commande par découplage non linéaire avec correction dans l'espace articulaire pour réaliser cette tâche, malheureusement la structure de commande pose le problème de la cinématique inverse lors de la synthèse de cette dernière.

Des solutions sont envisagées pour la résolution du problème de la cinématique inverse telle que les méthodes basées sur le calcul de la pseudo inverse, la décomposition de la matrice jacobienne en SVD, des algorithmes numériques comme Gréville, ajout de  $(n - m)$  contraintes.

En suites, nous avons exploité la redondance pour ajouter des objectifs d'évitement d'obstacle, c'est pour cela que nous avons présenté certaines méthodes basées sur le calcul de la cinématique inverse en ajoutant des manipulabilités dans le noyau du jacobien. La mise oeuvre de ces méthodes change selon la tâche à réaliser et l'obstacle considéré.

# ANNEXES

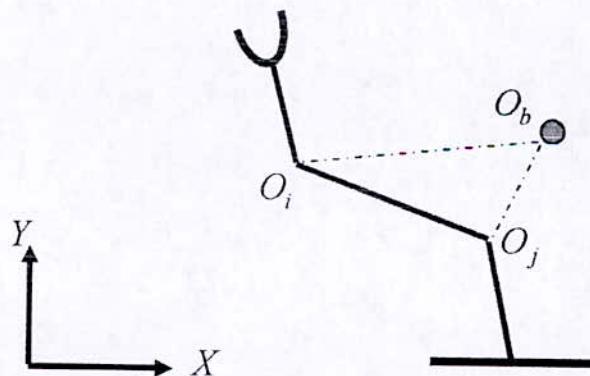
# DEMONSTRATION DE LA METHODE DU GRADIENT

## A.1. GRADIENT DE LA DISTANCE :

### A.1.1 Démonstration :

Pour le développement du calcul du gradient de la distance.

Prenons la situation ci-dessous :



Posons  $O_b(x_b, y_b)$  les coordonnées de l'obstacle. Soit  $O_i(x_i, y_i)$  les coordonnées du  $i^{\text{ème}}$  centre de rotation et  $O_j(x_j, y_j)$  le centre de rotation le plus proche de l'obstacle.

On détermine  $O_j$  de la façon suivante :

$$|O_j O_b| = \min(|O_i O_b|) \quad (\text{A.1})$$



On écrit la fonction de coût de la manière suivante :

$$f(q) = \sqrt{(x_J - x_b)^2 + (y_J - y_b)^2} \quad (\text{A.2})$$

Le gradient peut donc s'écrire :

$$\frac{\partial f(q)}{\partial q_i} = \frac{2 \cdot w(q_i)}{\sqrt{(x_J - x_b)^2 + (y_J - y_b)^2}} \quad (\text{A.3})$$

Avec :

$$w(q_i) = \frac{1}{2} \cdot \frac{\partial}{\partial q_i} \cdot \left( (x_J - x_b)^2 + (y_J - y_b)^2 \right) \quad (\text{A.4})$$

Ou encore :

$$w(q_i) = \frac{\partial \cdot (x_J - x_b)}{\partial q_i} \cdot (x_J - x_b) + \frac{\partial \cdot (y_J - y_b)}{\partial q_i} \cdot (y_J - y_b) \quad (\text{A.5})$$

Or on sait que :

$$(x_J - x_b) = \sum_{k=1, J-1} \left( l_k \cdot \cos \left( \sum_{n=1, k} q_n \right) \right) - x_b \quad (\text{A.6})$$

$$(y_J - y_b) = \sum_{k=1, (J-1)} \left( l_k \cdot \sin \left( \sum_{n=1, k} q_n \right) \right) - y_b \quad (\text{A.7})$$

Avec  $l_k$  longueur du bras du robot de  $O_k$  à  $O_{k+1}$

On en déduit :

$$\frac{\partial (x_J - x_b)}{\partial q_i} = - \sum_{k=1, (J-1)} \left( l_k \cdot \sin \left( \sum_{n=1, k} q_n \right) \right) \quad (\text{A.8})$$

$$\frac{\partial (y_J - y_b)}{\partial q_i} = - \sum_{k=1, (J-1)} \left( l_k \cdot \cos \left( \sum_{n=1, k} q_n \right) \right) \quad (\text{A.9})$$

Ce qui donne :

$$\begin{aligned} w(q_i) = & - \sum_{k=i, (J-1)} \left( l_k \cdot \sin \left( \sum_{n=1, k} q_n \right) \right) \cdot \left( \sum_{k=1, (J-1)} \left( l_k \cdot \cos \left( \sum_{n=1, k} q_n \right) \right) - x_b \right) \\ & - \sum_{k=i, (J-1)} \left( l_k \cdot \cos \left( \sum_{n=1, k} q_n \right) \right) \cdot \left( \sum_{k=1, (J-1)} \left( l_k \cdot \sin \left( \sum_{n=1, k} q_n \right) \right) - y_b \right) \end{aligned} \quad (\text{A.10})$$

En réunissant les indices,

$$\begin{aligned}
w(q_i) = & \sum_{k=i,(J-1)} \left( l_k \cdot \sin \left( \sum_{n=1,k} q_n \right) \right) \cdot \sum_{k=1,(i-1)} \left( l_k \cdot \cos \left( \sum_{n=1,k} q_n \right) \right) \\
& - \sum_{k=i,(J-1)} \left( l_k \cdot \sin \left( \sum_{n=1,k} q_n \right) \right) \cdot \left( \sum_{k=i,(J-1)} \left( l_k \cdot \cos \left( \sum_{n=1,k} q_n \right) \right) - x_b \right) \\
& + \sum_{k=i,(J-1)} \left( l_k \cdot \cos \left( \sum_{n=1,k} q_n \right) \right) \cdot \sum_{k=1,(i-1)} \left( l_k \cdot \sin \left( \sum_{n=1,k} q_n \right) \right) \\
& + \sum_{k=i,(J-1)} \left( l_k \cdot \cos \left( \sum_{n=1,k} q_n \right) \right) \cdot \left( \sum_{k=i,(J-1)} \left( l_k \cdot \sin \left( \sum_{n=1,k} q_n \right) \right) - y_b \right)
\end{aligned} \tag{A.11}$$

Les lignes (1) et (3) de l'équation (A.11) s'annulent ce qui nous donne :

$$\begin{aligned}
w(q_i) = & - \sum_{k=i,(J-1)} \left( l_k \cdot \sin \left( \sum_{n=1,k} q_n \right) \right) \cdot \left( \sum_{k=i,(J-1)} \left( l_k \cdot \cos \left( \sum_{n=1,k} q_n \right) \right) - x_b \right) \\
& + \sum_{k=i,(J-1)} \left( l_k \cdot \cos \left( \sum_{n=1,k} q_n \right) \right) \cdot \left( \sum_{k=i,(J-1)} \left( l_k \cdot \sin \left( \sum_{n=1,k} q_n \right) \right) - y_b \right)
\end{aligned} \tag{A.12}$$

Ce qui est équivalent à :

$$w(q_i) = -(y_j - y_i) \cdot (x_i - x_b) + (x_j - x_i) \cdot (y_i - y_b) \tag{A.13}$$

Ou encore :

$$w(q_i) = \overline{O_i O_j} \wedge \overline{O_b O_i} = \overline{O_i O_b} \wedge \overline{O_i O_j} \tag{A.14}$$

Ce qui donne le gradient suivant :

$$\frac{\partial f(q)}{\partial q_i} = \frac{-(y_j - y_i) \cdot (x_i - x_b) + (x_j - x_i) \cdot (y_i - y_b)}{\sqrt{(x_j - x_b)^2 + (y_j - y_b)^2}} \tag{A.15}$$

Ou encore :

$$\frac{\partial f(q)}{\partial q_i} = \frac{\overline{O_i O_b} \wedge \overline{O_i O_j}}{|O_b O_j|} \tag{A.16}$$

# ALGORITHMES D'ÉVITEMENT D'OBSTACLE

## B.1. Algorithme d'évitement d'obstacle par l'approche tâche prioritaire :

Initialisation des variables.

Tant que  $\|X_f - X_e\| > \text{précision}$

$P_c$  : position courante.

$P_d$  : position désirée.

$$dX_e = P_d - P_c$$

Le calcul de  $J_e$ .

Le calcul de  $X_0$ .

Le calcul de  $J_0$ .

Le calcul de  $d_{obs}$ .

$$dX_0 = \frac{1}{d_{obs}^2} \cdot (X_0 - X_{obs})$$

$$S_p = 0.3$$

$$Xnorm = \|dX_e\|$$

Si  $Xnorm > S_p$  alors  $dX_e = \frac{dX_e}{Xnorm} \cdot S_p$

Fin si

$\alpha_h, \alpha_0$  : coefficients de pondération.

$d_1, d_2$  : deux positions intermédiaires de l'obstacle.

Si  $d_{obs} \leq d_1$  alors

$$\alpha_h = 1$$

Fin si

Si  $d_{obs} \geq d_2$  alors

$$\alpha_h = 0$$

Fin si

Si ( $d_2 > d_{obs}$  et  $d_{obs} < d_1$ ) alors

$$\alpha_h = d_{obs} \cdot (d_2 - d_{obs})$$

Fin si

Si  $d_{obs} < \text{seuil\_contact}$  #abondant la tâche.

$$\text{Alors } dq = J_0^+ \cdot dX_0$$

Sinon

$$dq = J_e^+ \cdot dX_e + \alpha_h \cdot (J_0 \cdot (I - J_e^+ \cdot J_e)) \cdot (\alpha_0 \cdot dX_0 - J_0 \cdot J_e^+ \cdot dX_e)$$

Fin si

$$q_c = q_c + dq$$

Fin tant que

## B.2. Algorithme d'évitement d'obstacle par l'approche fonction de coût :

Initialisations des variables.

Tant que  $\|X_f - X_e\| > \text{précision}$

$$dX_e = P_d - P_c$$

Le calcul de  $J_e$ .

Le calcul de  $X_0$ .

Le calcul de  $d_{obs}$ .

$$h_s = \frac{1}{d_{obs}^2}$$

$$dX = X_{obs} - X_0$$

$$S_p = 0.3$$

$$X_{norm} = \|dX_e\|$$

Si  $X_{norm} > S_p$  alors

$$dX_e = \frac{dX_e}{X_{norm}} \cdot S_p$$

Fin si

$\alpha$  : coefficient de pondération.

$$Z = -\alpha \cdot h_s^2 \cdot [dX \ 0]^T$$

$$dq = J_e^+ \cdot dX_e + (I - J_e^+ \cdot J_e) Z$$

$$q_c = q_c + dq$$

Fin tant que

### **B.3. Algorithme d'évitement d'obstacle par l'approche gradient de la distance :**

Initialisation des variables.

Tant que  $\|X_f - X_e\| > \text{précision}$

$$dX_e = P_d - P_c$$

Le calcul de  $J_e$ .

Le calcul de  $X_0$ .

Le calcul de gradient de la distance :  $grad = [f_1 \ f_2 \ f_3]^T$

Le calcul de  $dq_0$  :  $dq_0 = (I - J_e^+ \cdot J_e) grad$

Le calcul de  $d_{obs}$ .

$$S_p = 0.3$$

$$X_{norm} = \|dX_e\|$$

Si  $X_{norm} > S_p$  alors

$$dX_e = \frac{dX_e}{X_{norm}} \cdot S_p$$

Fin si

Si  $d_{obs} > d_2$  #position loin de l'obstacle.

Alors  $\alpha=1, \beta=0$

Sinon

Si ( $d_2 > d_{obs}$  et  $d_{obs} < d_1$ )

Alors  $\alpha=1, \beta=d_{obs} \cdot (d_{obs} - d_2)$

Sinon #position proche de l'obstacle.

$$\alpha=1-d_{obs}, \beta=\frac{1}{d_{obs}^2}$$

Si  $(d_{obs} - R) \leq \text{seuil\_contact}$  #abondant la tâche.

Alors  $\alpha=0$

Fin si

Fin si

Fin si

$$dq = \alpha \cdot J_e^+ \cdot dX_e + \beta \cdot dq_0$$

$$q_c = q_c + dq$$

Fin tant que

#### **B.4. Algorithme d'évitement d'obstacle par l'approche gradient d'une fonction objective :**

Initialisation des variables.

Tant que  $\|X_f - X_e\| > \text{précision}$

$$dX_e = P_d - P_c$$

Le calcul de  $J_e$ .

Le calcul de  $X_0$ .

Le calcul de  $J_0$ .

Le calcul de  $d_{obs}$ .

$$dX_0 = X_{obs} - X_0$$

$$S_p = 0.3$$

$$X_{norm} = \|dX_e\|$$

Si  $X_{norm} > S_p$  alors

$$dX_e = \frac{dX_e}{X_{norm}} \cdot S_p$$

Fin si

$\alpha, \lambda, k$  : coefficients de pondérations.

$$\frac{dp}{dq} = \lambda \cdot k \cdot d_{obs}^{-(k+2)} \cdot J_0^T \cdot dX_0$$

$$dq = J_e^+ \cdot dX_e + \alpha \cdot (I - J_e^+ \cdot J_e) \frac{dp}{dq}$$

$$q_c = q_c + dq$$

Fin tant que

**NB** : Si on connaît à préalable la position  $X_0$ , on peut aussi calculer la matrice jacobienne par rapport à ce point de façon directe, si non on doit implémenter l'algorithme suivant qui a pour but de calculer à chaque instant d'échantillonnage  $X_0$  et  $J_0$ .

### B.5. Algorithme de calcul de $X_0$ :

$d = \text{inf}$

$X_a^j$  : la position de la  $j^{\text{ème}}$  articulation.

Pour  $j$  de 1 à  $n$  #parcours du robot.

$$u = \frac{X_a^{j+1} - X_a^j}{\|X_a^{j+1} - X_a^j\|} \quad \text{\#vecteur unitaire directeur du segment.}$$

$$c = \text{MAX}[0, \text{MIN}(u^T \cdot (X_{obs} - X_a^j), 1)]$$

Si  $c = 1$  #  $X_0 = X_a^{j+1}$

$$a = \|X_{obs} - X_a^{j+1}\|$$

Sinon Si  $c=0$

$$\# X_0 = X_a^j$$

$$a = \|X_{obs} - X_a^j\|$$

Sinon

$$a = \sqrt{\|X_a^j - X_{obs}\|^2 - c^2}$$

Fin si

Si  $d > a$

$$d = a$$

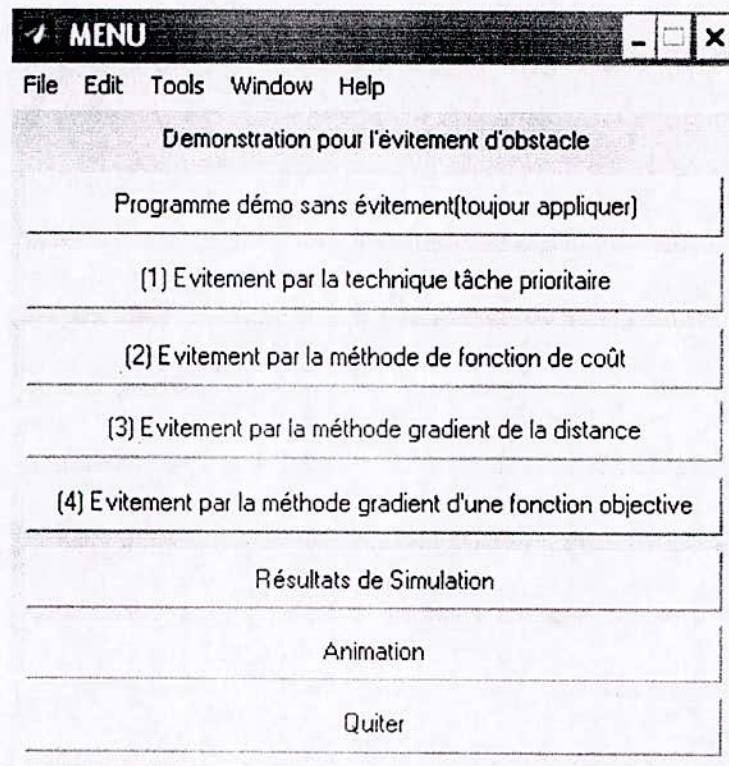
$$J_{oj} = J_{oj} + \begin{pmatrix} -c \cdot u \wedge z \\ 0 \end{pmatrix}$$

$$X_0 = X_a^j + c \cdot u$$

Fin si

Fin pour

Les algorithmes d'évitement sont programmés sous **MATLAB** dans l'interface graphique ci-dessous :





# REFERENCES BIBLIOGRAPHIQUES

- [BAI 84]: **J. Baillieul, J. Hollerbach, R. Brockett**, "Programming and control of kinematically redundant manipulators", Proc. of 23d Conf. on decision and control, 1984.
- [BAI 86]: **J. Baillieul**, "Avoiding Obstacles and resolving kinematic redundancy", Proc. IEEE Conf. on Robotics and Automation, 1986, pp.1698-1704.
- [BAI 86]: **Baillieul J.**, "Avoiding obstacle and resolving kinematic redundancy", Proc. IEEE Int. Conf. on Robotics and Automation, San Francisco, avril 1986, P. 1698-1704.
- [BEN 75]: **Benoit M., Briot M., Donnarel H., Liégeois A., Meyer M.A. Renaud M.**, « Synthèse de la commande dynamique d'un téléopérateur redondant », Revue RAIRO, Série J-2, p.89-103, mai 1975.
- [BEN 03] : **A.Benallegue, B.Daachi, A.Ramdane Cherif**, "Commande neuronale adaptative de robots manipulateurs redondants avec évitement d'obstacle fixes", Laboratoire de robotique de versailles et laboratoire de recherche en informatique (PRISM), France, 2003.
- [CHE 88] : **C. Chevallereau**, "Commande de robots manipulateurs dans l'espace cartésien", thèse de doctorat, Nantes, 1988, to appear.
- [CHA 86]: **Chang P.H.**, "A closed form solution for the control of manipulators with kinematic redundancy ".Proc. IEEE int. Conf. on Robotics and Automation San Francisco, avril 1986. P.9-14.
- [CRA 89]: **JOHN J. CRAIG** « Introduction to Robotics: Mechanics and Control », 2nd ed, Addison-Wesley Publishing Company, Canada, 1989.
- [DON 79]: **Dombre E., Fournier A.**, "yet another calibration technique to reduce the soft between CAD world and real world", Proc.1<sup>st</sup> WAC94 on intelligent Automation and Soft Computing. Hawai. USA.aout 1994, P.47-52.

[FOU 80]: **A.Fournier**, "Génération de mouvements en robotique application des inverses généralisées et des pseudos inverses», Thèse d'état, Montpellier, 1980

[HOL 85]: **J.M. Hollerbach, K.C. Suh**, "Redundancy resolution of manipulators through torque optimization", Proc. IEEE Conf. on Robotics and Automation, 1985, pp.1016-1021.

[GEO 77]: **George E. Forsythe, Michael A. Malcolm, and Cleve B. Moler**. "Computer Methods for Mathematical Computations". Prentice-Hall, 1977.

[GRE 80]: **A. Ben-Israel, T.N.E. Gréville**, " Generalized inverses: Theory and applications", New York, 1980, Robert E. Krieger publishing. Co.

[KHA 99]: **WISAMA KHALIL ET ETIENNE DOMBRE** «Modélisation Identification et Commande des robots », 2<sup>e</sup> édition, Hermes Science Publications, Paris, 1999.

[KHA 88]: **W.Khalil et C.Chevallereau**, « a new method for the solution of the invse kinematics of redundant robots », Laboratoire d'automatique de Nantes, France, 1988.

[KLE 84]: **C.A.Klein**, "Use of redundancy in the design of robotic systems", 2<sup>ème</sup> inter. Symp. On Reserch, Tokyo, pp.58-65. 1984.

[KON 81]: **M.S. Konstantinov, M.D. Markov, D.N. Nenchev**, "Kinematic control of redundant manipulators", Proc. 11th Int. Symp. On Industrial Robots, 1981, Tokyo, Japan, pp.561-568.

[LAW 74]: **Lawson C.L, Hanson R.J.**, Solving Least Squares problems. Prentics-Hall, Englewood Cliffs, 1974.

[LIE 77]: **A.Liégeois**, "Automatic supervisory control of the configuration and behavior of multi-body mechanisms", IEEE Trans. on Syst., Man, and Cybernetics. Vol. SMC-7(12), 1977, pp. 868-871.

- [MAC 85]: **A.A. Maciejewski, C.A. Klein**, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments", *Int. J Robotics Research*, Vol. 4, N0.3, Fall 1985, pp.109-117.
- [MAC 90]: **Anthony A. Maciejewski**, "Dealing with the ill-conditioned equations of motion for articulated figures". *IEEE Computer Graphics and Applications*, pages 63-71, May 1990.
- [MAC 85]: **Maciejewski A.A., Klein C.A.**, "obstacle avoidance for kinematically redundant manipulators in dynamically varying environnement ", *The int.J. Of Robotics Research*, Vol.4 (3), fall 1985, P.109-117.
- [MAR 00]: **Eric Marchand and Nicolas Courty**, "Image-based virtual camera motion strategies". *Graphics Interface*, pages 69-76, 2000.
- [NEN 92]: **Dragomir N. Nenchev**, "Restricted Jacobian matrices of redundant manipulators in constrained motion tasks". *The International Journal of Robotics Research*, 11(6):584-597, December 1992.
- [POU 91] : **Mehdi Pourazady and ligong Ho**, collision avoidance control of redundant manipulators.*Mechanisme and Machine theory*, 26(6):603-611, 1991.
- [SAI 04]: **Ludocic Saint-Bauzel et Damien Lhomme-Desages**, "Evitement d'obstacle dans un robot série redondant", rapport DEA, France, Février 2001.
- [SAM 90] : **Claude Samson, Michel Le Borgne, and Bernard Espiau**. « Robot Control The Task Function Approach ». Oxford Science Publications, 1990.
- [SUN 92]: **Sung-Woo Kim and Ju-Jang Lee**, "Resolved Motion Rate Control of Redundant Robots using Fuzzy Logic", Korea Advanced Institute of Science and Technology, Korea, 1992.
- [SPO 89]: **MARK W.SPONG** «Motion Control of Robot Manipulators», The Coordinated Science laboratory; University of Illinios at Urbana-Champaign; USA, 1989.

[TER 00] : **Kang Teresa Ge**, "Solving Inverse Kinematics Constraint Problems for Highly Articulated Models", A thesis presented to the University of Waterloo in fulfilment of the thesis requirement for the degree of Master of Mathematics in Computer Science, pages 49-58, 2000.

[Wam 86] : **Wampler C.W.**, « Manipulator inverse kinematic solutions based on vector formulation and damped least-squares methods", IEEE Trans. On system, Man, and Cybernetics, Vol. SMC -16, 1986, p.93.101

[WHI 69]: **D.E. Whitney**, "Resolved motion rate control of manipulators and human Prostheses", IEEE Trans. Man. Machine Systems. MMS10, pp.47-53, 1969.

[WHI 72]: **D.E. Whitney**, "The mathematics of coordinated control of prosthetic arms and manipulators," ASME J. Dynamic Systems, Meas., Control, 1972, pp.303-309.

[YOS 84]: **Yoshikawa T**, "Analysis and control of robot manipulators with redundancy", The 1<sup>st</sup> Int. Symp. Of Robotics Reserch. MIT Press, Cambridge, p.735-748. 1984.

## ملخص:

العمل المقدم من خلال هذه المذكرة يتمثل في دراسة مشكلة التحكم في ذراع آلية تكرارية في الفضاء العملي مع وجود حاجز ثابت قمنا بعرض مختلف الطرق لتحديد النموذج الحركي الغير مباشر، أيضا قمنا بعرض مختلف طرق تحديد المسارات العملية، و استخراج النماذج الهندسية، الحركية و الديناميكية المباشرة و الغير مباشرة قمنا باقتراح نظام التحكم بطريقة العزم المحسوب وفق البنية المحسوبة في فضاء المفاصل. أيضا قمنا بتطبيق أربعة طرق لتخطي حاجز ثابت في المستوي و قمنا بالمحاكاة على ذراع آلية من نوع بيما 560.

**كلمات مفتاحية:** ذراع آلية (بيما 560)، الفضاء العملي، المسار المتكرر، النماذج الهندسية، الحركية، الديناميكية، العزم المحسوب، الحركية الغير مباشرة، تخطي حاجز.

## RESUME :

Le travail présenté dans ce mémoire est une étude de la commande d'un bras manipulateur redondant dans l'espace opérationnel et en présence d'obstacle fixe. Nous avons présenté les différentes méthodes pour résoudre le problème de la cinématique inverse. Nous avons déterminé les modèles : géométrique, cinématique et dynamique (direct et inverse). Nous avons appliqué la structure de commande par découplage non linéaire avec correction dans l'espace opérationnelle pour commander le robot redondant en position et orientation. Ainsi on a appliqué quatre méthodes d'évitement d'obstacle fixe dans le plan, les simulations ont été faites sur le robot PUMA 560.

**Mots clés :** bras manipulateur redondant (PUMA560), espace opérationnel, trajectoire redondante, les modèles géométrique, cinématique et dynamique, couple calculé, la cinématique inverse, évitement d'obstacle.

## ABSTRACT:

The work presented in this thesis is a study of the control of an arm redundant manipulator in operational space and in the presence of fixed obstacle. We presented the various methods to solve the problem of inverse kinematics. We determined the models: geometrical, kinematic and dynamic (direct and inverse). We applied the structure of control by nonlinear decoupling with correction in space operational to order the redundant robot in position and orientation. Thus one applied four methods of fixed avoidance of obstacle in the plan, simulations were made on the robot PUMA 560.

**Key words:** arm redundant manipulator (PUMA560), operational space, redundant trajectory, models geometrical, kinematic and dynamic, calculated couple, inverse kinematics, avoidance of obstacle.