

République Algérienne Démocratique & Populaire  
Université Abou Bekr Belkaïd - Tlemcen  
Faculté de Technologie  
Département de Génie Electrique et Electronique

**Mémoire**  
pour l'obtention du diplôme de  
Magister  
Modélisation et Commande de Systèmes

Titre  
**Génération de trajectoire optimale  
pour les systèmes mécaniques sous  
actionnés**

Présenté par :

Mr ZATLA Hicham

Encadré par :

Mme CHOUKCHOU BRAHAM AMEL

Mr CHERKI BRAHIM

Soutenu en juillet 2012 devant la commission d'examen

Président	Mr. CHERMITTI Ali	Maître de conférences	Université de Tlemcen
Examineurs	Mr. MEGHABBER Abdelmadjid	Maître de conférences	Université de Tlemcen
	Mr. BOUMEDIENE Abdelmadjid	Maître de conférences	Université de Tlemcen

## Résumé

Les systèmes mécaniques sous actionnés sont des systèmes qui possède un nombre de degré de liberté supérieur au nombre de commande, souvent on se trouve dans cette situation lorsque l'un d'actionneur tombe en panne. Dans ces systèmes on cherche les trajectoires qui permettent d'aller d'une configuration initial à une configuration finale désirée dans un minimum de temps.

**Mots-clés.** *Robots sous actionnés, contrainte Non holonome, trajectoire, optimisation, planification.*

## Abstract

Underactuated mechanical systems are systems that has a number of degrees of freedom exceeds the number of control, often we find ourselves in this situation when one actuator fails. In this systems we investigate the trajectories that allow to go from initial configuration to a desired final configuration in the shortest time.

**Key words.** *Underactuated robots, holonomic constraint, trajectory, optimization, planning.*

Je tiens à remercier tous ceux qui m'ont aidé, conseillé et encouragé afin de réaliser ce modeste travail.

Je dédie ce modeste travail à mes très chers parents pour leur soutien et leur affection,

A mes frères.

# Table des matières

<b>1</b>	<b>Génération de trajectoire optimale pour les systèmes mécaniques sous actionnés</b>	<b>8</b>
1.1	Génération de trajectoire [20]	8
1.1.1	Génération de mouvement entre deux points dans l'espace articulaire [20],[27]	9
1.1.2	Génération de mouvement avec points intermédiaires [20][27]	13
1.2	Formulation du problème d'optimisation	16
1.3	Conclusion	17
<b>2</b>	<b>Généralités sur les Systèmes Mécaniques Sous Actionnés</b>	<b>18</b>
2.1	Introduction	18
2.2	Formalisme de Lagrange	18
2.3	Systèmes mécaniques complètement actionnés	19
2.4	Systèmes mécaniques sous actionnés	20
2.4.1	Systèmes mécaniques non holonomes	20
2.5	La linéarisation partielle par feedback	22
2.5.1	Linéarisation partielle localisée (Collocated partial feedback linearization)[28][25]	23
2.5.2	Linéarisation partielle non localisée (Noncollocated partial feedback linearization)	23
2.5.3	Linéarisation partielle sous des entrées couplées	24
2.6	Exemples des systèmes mécaniques sous actionnés	25
2.6.1	La boule et la poutre (beam and ball)	25
2.6.2	Chariot-pendule inversé	25
2.6.3	Acrobot et pendubot	26
2.7	Conclusion	27
<b>3</b>	<b>Systèmes Mécaniques Non holonomes [12]</b>	<b>28</b>
3.1	Introduction	28
3.2	Intégrabilité des Contraintes	30
3.2.1	Stabilisabilité	34
<b>4</b>	<b>Méthodes d'optimisation</b>	<b>37</b>
4.1	Programmation quadratique séquentielle(SQP)[24][4]	37
4.1.1	La Méthode SQP de base	37
4.2	Algorithmes Génétiques [10] [30]	39
4.2.1	Gestion des contraintes	40
4.2.2	Avantages et inconvénients	41
4.3	Les essais particuliers (Particle Swarm Optimization : PSO) [9][22]	41
4.3.1	Description informelle	42

4.3.2	Formalisation . . . . .	44
<b>5</b>	<b>Applications</b>	<b>45</b>
5.1	Cas d'un Robot manipulateur à deux axes planaire type $R_a R_p$ . . . . .	45
5.1.1	Condition d'intégrabilité . . . . .	45
5.1.2	Conclusion . . . . .	51
5.2	Cas d'un Robot manipulateur de type $(n - 1)X_a R_p$ . . . . .	51
5.2.1	La linéarisation dynamique par feedback . . . . .	52
5.2.2	Robot manipulateur de Type PPR . . . . .	58
5.2.3	Conclusion . . . . .	61
<b>A</b>	<b>Les outils de la théorie du contrôle non linéaire</b>	<b>64</b>
A.1	Éléments de géométrie différentielle . . . . .	64
A.1.1	Contrôlabilité . . . . .	67
<b>B</b>	<b>Classification des systèmes non holonomes</b>	<b>70</b>

# Table des figures

1.1	Loi polynomiale de degré trois . . . . .	11
1.2	Loi polynomiale de degré cinq . . . . .	12
1.3	Loi Bang-Bang . . . . .	13
1.4	Notations utilisées pour les splines . . . . .	14
1.5	Exemple sur la génération de trajectoire . . . . .	15
1.6	Schématisation du problème d'optimisation . . . . .	17
2.1	Roue mobile . . . . .	20
2.2	Roue mobile sur le plan $(x, y)$ . . . . .	21
2.3	La boule et la poutre . . . . .	25
2.4	Chariot-pendule inversé . . . . .	26
2.5	(a) Acrobot, (b) Pendubot . . . . .	27
3.1	Coordonnées généralisées d'un disque roulant . . . . .	30
4.1	Exemple sur le croisement de deux individus . . . . .	40
4.2	Exemple sur la mutation . . . . .	40
4.3	Les opérations génétiques pour deux individus . . . . .	40
4.4	Le cercle virtuel pour un essaim de sept particules. . . . .	43
4.5	Schéma de principe du déplacement d'une particule . . . . .	44
5.1	Positions des axes en radian( $rd$ ) . . . . .	48
5.2	Vitesses des axes( $rd/s$ ) . . . . .	48
5.3	Accélération des axes ( $rd/s^2$ ) . . . . .	49
5.4	Couples des axes( $Nm$ ) . . . . .	49
5.5	Positions des axes en radian( $rd$ ) . . . . .	50
5.6	Vitesses des axes( $rd/s$ ) . . . . .	50
5.7	Accélération des axes ( $rd/s^2$ ) . . . . .	50
5.8	Couples des axes( $Nm$ ) . . . . .	50
5.9	Robot manipulateur de type $(n - 1)X_a - R_p$ . . . . .	53
5.10	Interprétation physique de l'équation (5.38) . . . . .	55
5.11	Interprétation physique de l'équation (5.39) . . . . .	55
5.12	Shema du controleur dynamique linearisé . . . . .	57
5.13	Robot manipulateur PPR . . . . .	59
5.14	Les trajectoires de sorties $y_1$ et $y_2$ (m) . . . . .	60
5.15	Planification du repos au repos : $x(m)$ , $y(m)$ et $\theta(rd)$ . . . . .	60
5.16	Les vitesses du Robot $x(m/s)$ , $y(m/s)$ et $\theta(rd/s)$ . . . . .	61
5.17	Foces appliquée au robot $f_1, f_2$ (N) . . . . .	61
A.1	Mouvement par crochet de Lie . . . . .	65

# Introduction générale

Ces dernières années beaucoup de recherches ont été consacrés aux systèmes mécaniques sous actionnés (S.M.S.A), ces systèmes sont caractérisés par un nombre de degrés de libertés supérieur au nombres d'actionneurs. Cette classe des systèmes mécaniques se retrouve soit naturellement c'est à dire le sous actionnement est volontaire par exemple pour des raisons économiques ou de poids, soit involontairement lors d'une panne d'actionneur. Comme exemples de systèmes mécaniques sous actionnés, nous pouvons citer : Les véhicules sous-marins, Les hélicoptères, robots mobiles et les manipulateurs sous actionnés.

Dans ce mémoire on s'intéresse aux manipulateurs sous actionnés, plusieurs chercheurs se sont intéressés à ces problèmes, on peut citer [2] ou un frein est ajouté au niveau de l'axe passif afin de pouvoir simplifier la commande, en fait dans ce cas, on se retrouve avec un actionnement complet asymétrique (on peut freiner mais pas accélérer), l'étude de la stabilité pour un manipulateur sous actionné de type  $2R$  avec le deuxième axe passif est donnée dans [12], la planification de trajectoire pour un manipulateur non holonome type  $3R$  avec le dernier axe passif (libre sans frein) est donnée dans [12].

Le but de notre travail est de trouver une trajectoire optimale qui satisfait la contrainte non holonome. Il n'existe pas de méthode générale pour résoudre ce type de problème, nous tenterons de proposer une solution dans certains cas particulier. Notre mémoire est structué comme suit :

- *Chapitre 1* : Dans ce chapitre on décrit les différents type de génération de trajectoire avec la formalisation du problème d'optimisation.
- *chapitre 2* : Ce chapitre décrit qualitativement les SMSA obtenus par formalisme de Lagrange, nous expliquerons ensuite le concept de sous actionnement et donnons la définition des contraintes holonomes et non holonomes.
- *chapitre 3* : Dans ce chapitre on étudie succinctement quelques méthodes de résolution des problèmes d'optimisation, il est composé de deux parties, une partie consacrée à une méthode classique dite programmation séquentielle quadratique et une deuxième partie consacrée aux méthodes métaheuristique telles que les algorithmes génétiques et les méthodes d'essaims particulaires.
- *chapitre 4* : Dans ce chapitre nous étudions quelques propriétés générales des systèmes non holonomes, en particulier les limites imposées par les contraintes non holonomes.
- *chapitre 5* : Ce chapitre est consacré à deux application, la première la génération de trajectoire pour un manipulateur sous actionné à deux axes planaire (sur le plan horizontale) avec le deuxième sous actionné (libre). la deuxième application traite

le problème de planification de trajectoire pour un manipulateur de type PPR avec le dernier axe est passif (libre).

Enfin nous terminons évidemment le manuscrit par une conclusion générale.

# Chapitre 1

## Génération de trajectoire optimale pour les systèmes mécaniques sous actionnés

Ces dernières années, le problème de planification de mouvement non holonome a revêtu une grande importance. Ce problème consiste à chercher une commande en boucle ouverte qui ramène le système d'un état initial à un état final en un temps fini. Dans ce chapitre on s'intéresse à la recherche d'une trajectoire optimale ou admissible qui relie l'état initial à l'état final et minimise un critère sous certaines contraintes. Cette trajectoire admissible est utilisée par la suite comme trajectoire de référence dans le problème de suivi de trajectoire. Dans une première étape on passera en revue le problème de génération de trajectoire et dans une deuxième étape on s'intéressera au problème de recherche de la trajectoire optimale.

Beaucoup d'auteurs se sont intéressés à la génération de trajectoire on peut citer ([20],[27]) ils se sont attaqués aux méthodes de génération de trajectoire pour les systèmes mécaniques complètement actionnés. Ces méthodes sont basées sur des approches polynomiales. Les techniques de planification de mouvement sont décrites dans ([27],[5]), la planification des mouvements pour les Robots non holonomes est décrite dans ([21],[12]), la plupart des approches utilisées sont consacrées aux systèmes mécaniques non holonomes du premier ordre. Il n'existe pas de théorie générale pour la planification de mouvement pour les systèmes mécaniques avec contraintes non holonomes du second ordre, des résultats partiels sont donnés dans ([2, 3]) où un planificateur de mouvements numérique a été proposé pour un manipulateur sous-actionné planaire 2R. [13], ont résolu le problème de planification de mouvement pour un manipulateur sous actionné planaire 3R en se basant sur la linéarisation dynamique par feedback. D'autres auteurs ont utilisé l'approximation nilpotente pour la stabilisation d'un manipulateur sous-actionnés 2R planaire [15].

### 1.1 Génération de trajectoire [20]

La génération de trajectoire ou de mouvement consiste à calculer les 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 (abusivement appelées points).

On peut distinguer les classes de mouvement suivantes :

- le mouvement entre deux points avec trajectoire libre entre les points ;
- le mouvement entre deux points via des points intermédiaires, spécifiés notamment pour éviter les obstacles, avec trajectoire libre entre les points intermédiaires ;

- le mouvement entre deux points avec trajectoire contrainte entre les points (trajectoire rectiligne par exemple) ;
- le mouvement entre deux points via des points intermédiaires avec trajectoire contrainte entre les points intermédiaires.

Dans les deux premiers cas, la génération de mouvement peut se faire directement dans l'espace articulaire. Dans les deux derniers, la trajectoire étant décrite dans l'espace opérationnel, il est préférable de raisonner dans cet espace.

La génération de trajectoire dans l'espace articulaire présente plusieurs avantages :

- elle nécessite moins de calculs en ligne, puisque'il n'y a pas d'appel au modèle géométrique ou cinématique inverse ;
- le mouvement n'est pas affecté par le passage sur les configurations singulières ;
- les contraintes de vitesses et de couples maximaux sont directement déduites des limites physiques des actionneurs.

### 1.1.1 Génération de mouvement entre deux points dans l'espace articulaire [20],[27]

On considère un robot à  $n$  degrés de liberté. Soient  $q_i$  et  $q_f$  les vecteurs des coordonnées articulaires correspondant aux configurations initiale et finale. On désigne respectivement par  $K_v$  et  $K_a$  les vecteurs des vitesses et accélérations articulaires maximales. Les paramètres  $K_{vj}$  sont généralement calculés de façon exacte à partir des caractéristiques des actionneurs et des rapports de réduction des organes de transmission, tandis que les paramètres  $k_{aj}$  sont approchés par le rapport des couples moteurs maximaux aux inerties maximales vues par les articulations (bornes supérieures des termes  $A_{jj}$  définies dans le modèle dynamique). Une fois le mouvement avec ces contraintes cinématiques calculé, on peut ensuite procéder à un changement d'échelle pour satisfaire les contraintes de couple maximum obtenues avec le modèle dynamique.

Le mouvement entre  $q_i$  et  $q_f$  en fonction du temps  $t$  est décrit par l'équation suivantes ;

$$q(t) = q^i + r(t)D \quad \text{pour } 0 \leq t \leq t_f \quad (1.1)$$

$$\dot{q}(t) = \dot{r}(t)D \quad (1.2)$$

Avec  $D = q_f - q_i$

Les valeurs aux limites de la fonction d'interpolation  $r(t)$  sont données par :

$$\begin{cases} r(0) = 0 \\ r(t_f) = 1 \end{cases} \quad (1.3)$$

Plusieurs fonctions permettent de satisfaire le passage par  $q_i$  à  $t = 0$  et par  $q_f$  à  $t = t_f$ . Nous étudierons successivement l'interpolation polynomiale, la loi bang-bang et la loi bang-bang.

#### Interpolation polynomiale

Les modes d'interpolation polynomiale les plus fréquemment rencontrés sont l'interpolation linéaire et l'interpolation par des polynômes de degrés trois et cinq.

## Interpolation linéaire

Il s'agit de l'interpolation la plus simple : le mouvement de chaque articulation est décrit par une équation linéaire en temps. L'équation du mouvement s'écrit :

$$q(t) = q^i + \frac{t}{t_f} D \quad (1.4)$$

Cette loi de mouvement est continue en position mais discontinue en vitesse. L'utilisation d'une telle loi de mouvement est inacceptable sur les robots réels à cause des à-coups qu'elle provoque.

## Polynôme de degré trois

Si l'on impose une vitesse nulle aux points de départ et d'arrivée, on ajoute deux contraintes de position. Le degré minimal du polynôme qui satisfait ces quatre contraintes est de degré trois et a pour forme générale :

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \quad (1.5)$$

Pour satisfaire les conditions aux limites, les coefficients ont pour expression :

$$\begin{cases} a_0 = q^i \\ a_1 = 0 \\ a_2 = \frac{3}{t_f^2} D \\ a_3 = \frac{-2}{t_f^3} D \end{cases}$$

L'expression (1.5) peut encore s'écrire sous la forme (1.1) en prenant :

$$r(t) = 3\left(\frac{t}{t_f}\right)^2 - 2\left(\frac{t}{t_f}\right)^3 \quad (1.6)$$

La figure 1.1 donne l'évolution des position, vitesse et accélération pour l'axe  $j$ . cette loi de mouvement assure la continuité des vitesses mais pas celle des accélérations. En pratique, les robots industriels sont suffisamment rigides pour que cette discontinuité soit filtrée par la mécanique. L'utilisation d'un polynôme de degré trois est donc généralement satisfaisante dans la plupart des applications. Pour une articulation quelconque  $j$ , la vitesse est maximum lorsque  $t = t_f/2$ . Elle a pour valeur :

$$|\dot{q}_{jmax}| = 3 \frac{|D_j|}{2t_f} \quad \text{avec} \quad |D_j| = |q_j^f - q_j^i| \quad (1.7)$$

L'accélération est maximum à  $t = 0$  et à  $t = t_f$ . Elle vaut :

$$|\ddot{q}_{jmax}| = 6 \frac{|D_j|}{t_f^2} \quad (1.8)$$

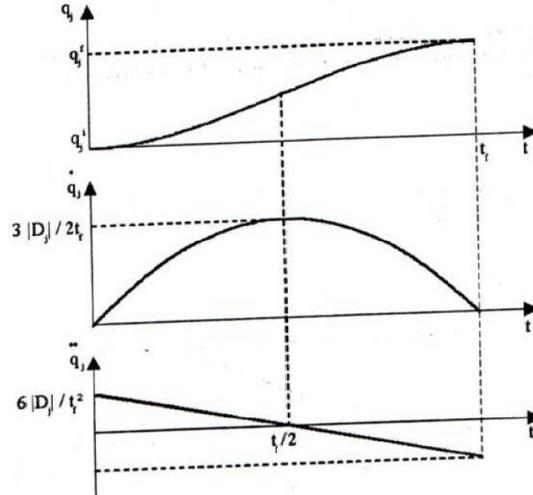


FIGURE 1.1 – Loi polynomiale de degré trois

### Polynôme de degré cinq

Pour les robots à grande vitesse ou transportant des charges importantes, il est nécessaire d'assurer la continuité des accélérations afin d'éviter d'exciter la mécanique. On dit alors que le mouvement est de classe C2. Il faut satisfaire six contraintes et le polynôme d'interpolation doit être de degré cinq. Avec les contraintes supplémentaires :

$$\begin{cases} \ddot{q}(0) = 0 \\ \ddot{q}(t_f) = 0 \end{cases} \quad (1.9)$$

On montre que la fonction de position peut se mettre sous la forme (1.1) avec :

$$r(t) = 10\left(\frac{t}{t_f}\right)^3 - 15\left(\frac{t}{t_f}\right)^4 + 6\left(\frac{t}{t_f}\right)^5 \quad (1.10)$$

Les évolutions des position, vitesse et accélération pour l'articulation j sont présentées à la Figure 1.2. Les vitesses et accélérations maximales ont pour expression :

$$|\dot{q}_{jmax}| = \frac{15|D_j|}{8t_f} \quad (1.11)$$

$$|\ddot{q}_{jmax}| = \frac{10|D_j|}{\sqrt{3}t_f^2} \quad (1.12)$$

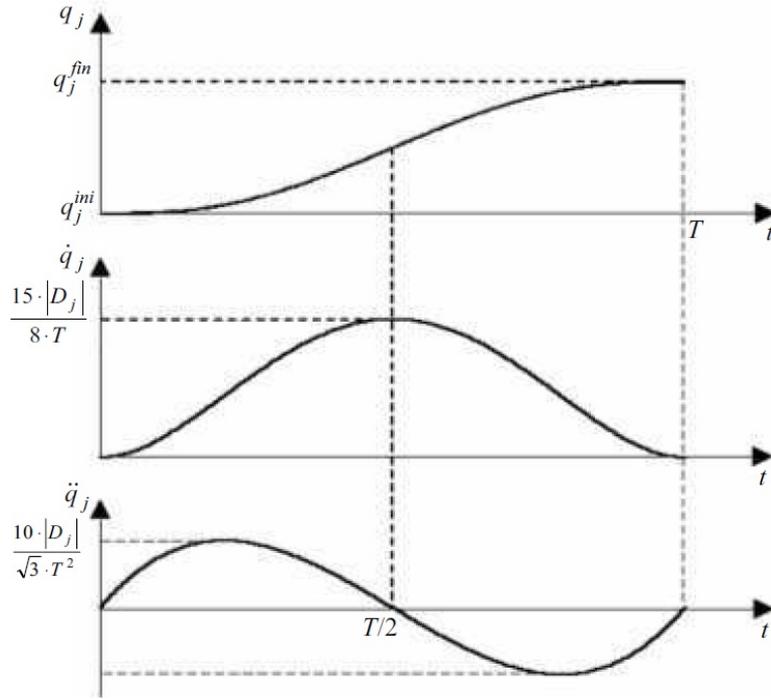


FIGURE 1.2 – Loi polynomiale de degré cinq

### Loi Bang-Bang

Le mouvement est représenté dans ce cas par une phase d'accélération constante jusqu'à  $t_f/2$  puis par une phase de freinage constante Figure 1.3. les vitesses de départ et d'arrivée sont nulles. Le mouvement est donc continu en position et en vitesse, mais discontinu en accélération. La position est donnée par :

$$\begin{cases} q(t) = q^i + 2\left(\frac{t}{t_f}\right)^2 D & \text{pour } 0 \leq t \leq \frac{t_f}{2} \\ q(t) = q^i + \left[-1 + 4\left(\frac{t}{t_f}\right) - 2\left(\frac{t}{t_f}\right)^2\right] D & \text{pour } \frac{t_f}{2} \leq t \leq t_f \end{cases} \quad (1.13)$$

Pour une articulation  $j$  donnée, les vitesse et articulation maximales ont pour expression :

$$|\dot{q}_{jmax}| = \frac{2|D_j|}{t_f}$$

$$|\ddot{q}_{jmax}| = \frac{4|D_j|}{t_f^2}$$

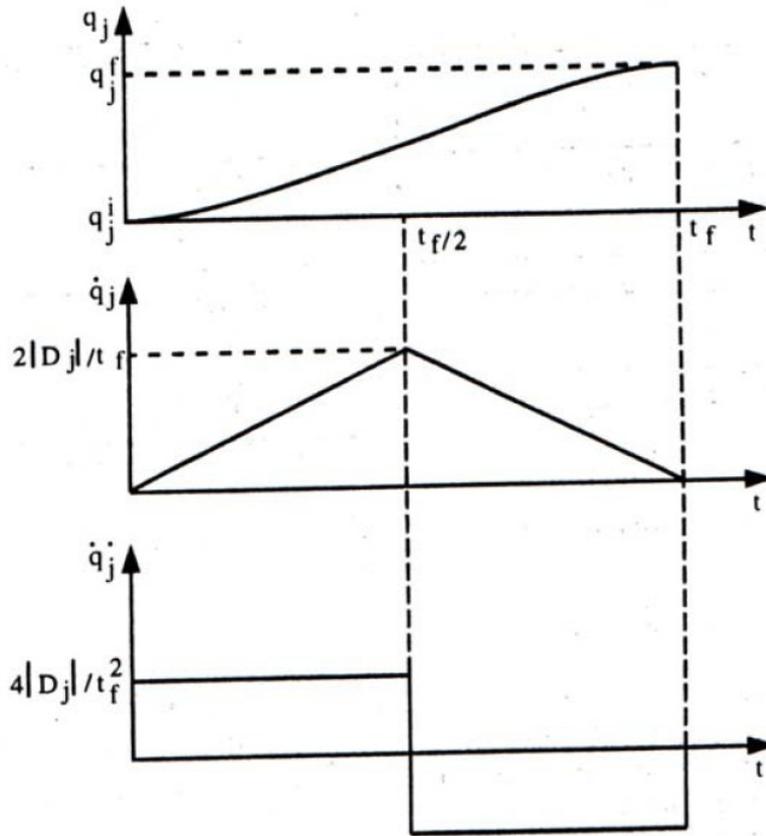


FIGURE 1.3 – Loi Bang-Bang

### 1.1.2 Génération de mouvement avec points intermédiaires [20][27]

Nous abordons, dans ce paragraphe, le problème de la génération de mouvement dans le cas où la trajectoire est contrainte à passer à vitesse non nulle sur un certain nombre de points intermédiaires (encore appelés points de passage). Ces points ont comme rôle de déformer la trajectoire pour éviter les collisions entre le robot et son environnement. Imposer une vitesse non nulle sur les points intermédiaires permet de diminuer le temps de cycle.

Parmi les méthodes de génération de trajectoire les plus utilisées celle de la spline cubique qui passe par tous les points intermédiaires et garantie la continuité en position, vitesse et accélération.

#### Utilisation de fonctions splines

On part d'une séquence de configuration articulaire  $q^0, q^1, \dots, q^n$  avec des temps de parcours  $t_0 = 0 < t_1 < t_2 < \dots < t_n = T$  il est plus judicieux de diviser l'intervalle de temps en parties égales, cela permet de faciliter l'optimisation. Sur chaque segment  $k$ , c'est-à-dire entre le  $k^{ieme}$  et  $(k + 1)^ieme$  point, le mouvement est représenté par une fonction de degré 5 Figure 1.4.

Le principe est de calculer de façon globale les accélérations sur les points de passage pour assurer la continuité en vitesse et en accélération le long de la trajectoire.

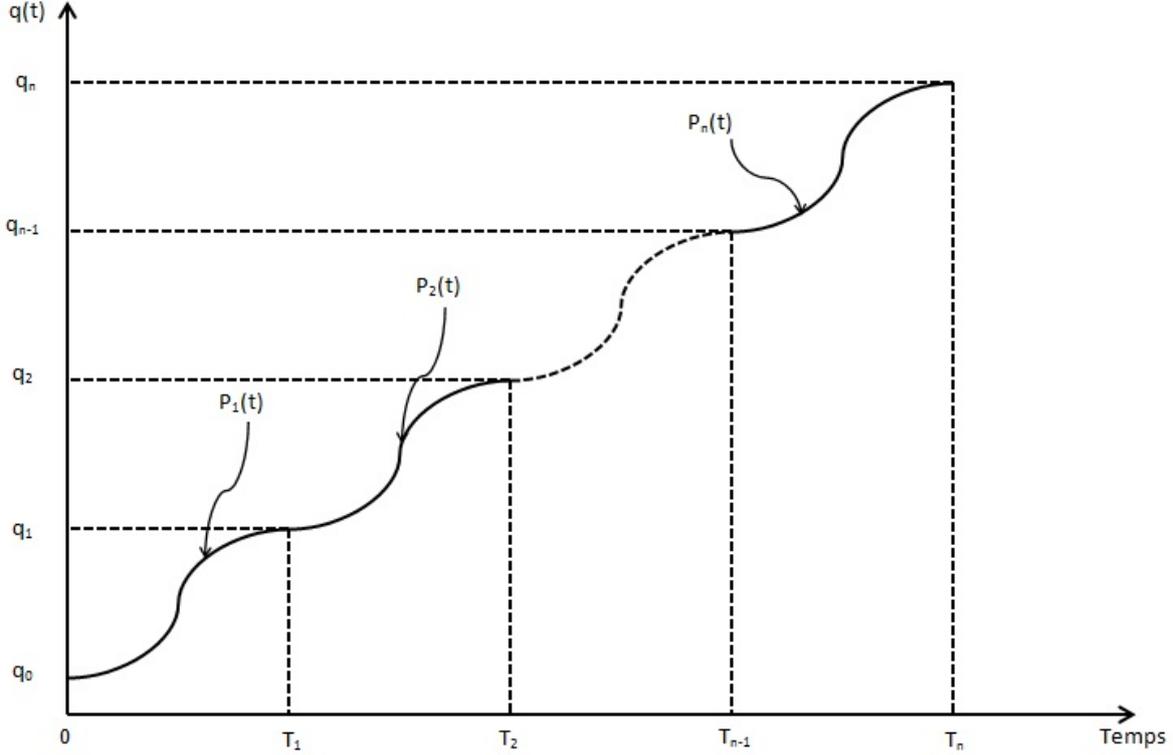


FIGURE 1.4 – Notations utilisées pour les splines

### La procédure adopté

D'après la Figure 1.4, on à  $(n+1)$  points répartis de manière équidistante sur l'intervalle  $[0, T]$ , reliés entre eux par la fonction spline de degré 5 qui s'écrit sous la forme :

$$p(t) = a_0 + a_1t^2 + a_2t^3 + a_3t^3 + a_4t^4 + a_5t^5, \quad (1.14)$$

chaque polynôme contient 6 variables, ce qui fait en tout  $6 \times n$  inconnues, on a besoin de  $6 \times n$  équations de telle sorte qu'elles vérifient la continuité en position, vitesse et accélération, pour cela la courbe qui passe par tous les points doit vérifier les équations suivantes :

$$\begin{cases} P_1(0) = q_0 \\ \dot{P}_1(0) = 0 \\ \ddot{P}_1(0) = 0 \end{cases} \begin{cases} P_1(t_1) = q_1 \\ P_2(t_1) = q_1 \\ \dot{P}_1(t_1) = \dot{P}_2(t_1) \\ \ddot{P}_1(t_1) = \ddot{P}_2(t_1) \\ \ddot{P}_1(t_1) = \ddot{P}_2(t_1) \\ \ddot{P}_1(t_1) = \ddot{P}_2(t_1) \end{cases} \begin{cases} P_2(t_2) = q_2 \\ P_3(t_2) = q_2 \\ \dot{P}_2(t_2) = \dot{P}_3(t_2) \\ \ddot{P}_2(t_2) = \ddot{P}_3(t_2) \\ \ddot{P}_2(t_2) = \ddot{P}_3(t_2) \\ \ddot{P}_2(t_2) = \ddot{P}_3(t_2) \end{cases} \cdots \begin{cases} P_{n-1}(t_{n-1}) = q_{n-1} \\ P_n(t_{n-1}) = q_{n-1} \\ \dot{P}_{n-1}(t_{n-1}) = \dot{P}_n(t_{n-1}) \\ \ddot{P}_{n-1}(t_{n-1}) = \ddot{P}_n(t_{n-1}) \\ \ddot{P}_{n-1}(t_{n-1}) = \ddot{P}_n(t_{n-1}) \\ \ddot{P}_{n-1}(t_{n-1}) = \ddot{P}_n(t_{n-1}) \end{cases}$$

$$\begin{cases} P_n(t_n) = q_n \\ \dot{P}_n(t_n) = 0 \\ \ddot{P}_n(t_n) = 0 \end{cases}$$

ces équation peuvent être écrites sous la forme matricielle  $Ax = B$  le vecteur  $x$  contient les inconnues,  $B$  contient les points de passage et  $A$  est une matrice carrée inversible, elle dépend du temps qui correspond au point de passage.

**Exemple 1.** supposons qu'on veut aller d'un point initial  $q_i = 0rd$  à un point final  $q_f = \frac{\pi}{2}rd$  à l'instant  $T = 12sec$  passant par deux points intermédiaire  $q_2 = 4rd$  et  $q_3 = 6rd$  avec des vitesses et accélérations nulles aux extrémités, on choisit ici deux polynômes de degré cinq, par exemple pour l'axe de temps on le subdivise en parties égales le problème s'écrit :

$$\begin{cases} P_1(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \\ P_2(t) = b_0 + b_1t + b_2t^2 + b_3t^3 + b_4t^4 + b_5t^5 \\ P_3(t) = c_0 + c_1t + c_2t^2 + c_3t^3 + c_4t^4 + c_5t^5 \end{cases} \quad (1.15)$$

Il est clair qu'il existe douze paramètre à trouver pour celà on écrit le système d'équation suivant :

$$\begin{cases} P_1(0) = q_i \\ \dot{P}_1(0) = 0 \\ \ddot{P}_1(0) = 0 \end{cases} \begin{cases} P_1(\frac{T}{3}) = P_2(\frac{T}{3}) \\ \dot{P}_1(\frac{T}{3}) = \dot{P}_2(\frac{T}{3}) \\ \ddot{P}_1(\frac{T}{3}) = \ddot{P}_2(\frac{T}{3}) \\ \ddot{P}_1(\frac{T}{3}) = \ddot{P}_2(\frac{T}{3}) \\ \ddot{P}_1(\frac{T}{3}) = \ddot{P}_2(\frac{T}{3}) \end{cases} \begin{cases} P_2(\frac{2T}{3}) = P_3(\frac{2T}{3}) \\ \dot{P}_2(\frac{2T}{3}) = \dot{P}_3(\frac{2T}{3}) \\ \ddot{P}_2(\frac{2T}{3}) = \ddot{P}_3(\frac{2T}{3}) \\ \ddot{P}_2(\frac{2T}{3}) = \ddot{P}_3(\frac{2T}{3}) \\ \ddot{P}_2(\frac{2T}{3}) = \ddot{P}_3(\frac{2T}{3}) \end{cases} \begin{cases} P_3(T) = q_f \\ \dot{P}_3(T) = 0 \\ \ddot{P}_3(T) = 0 \end{cases} \quad (1.16)$$

L'équation  $P_1(\frac{T}{3}) = P_2(\frac{T}{3})$  représente deux équations de la forme  $P_1(\frac{T}{3}) = q_1$  et  $P_2(\frac{T}{3}) = q_1$ , cet exemple est programmé sous Matlab qui donne l'évolution des position, vitesse et accélération, figure 1.5.

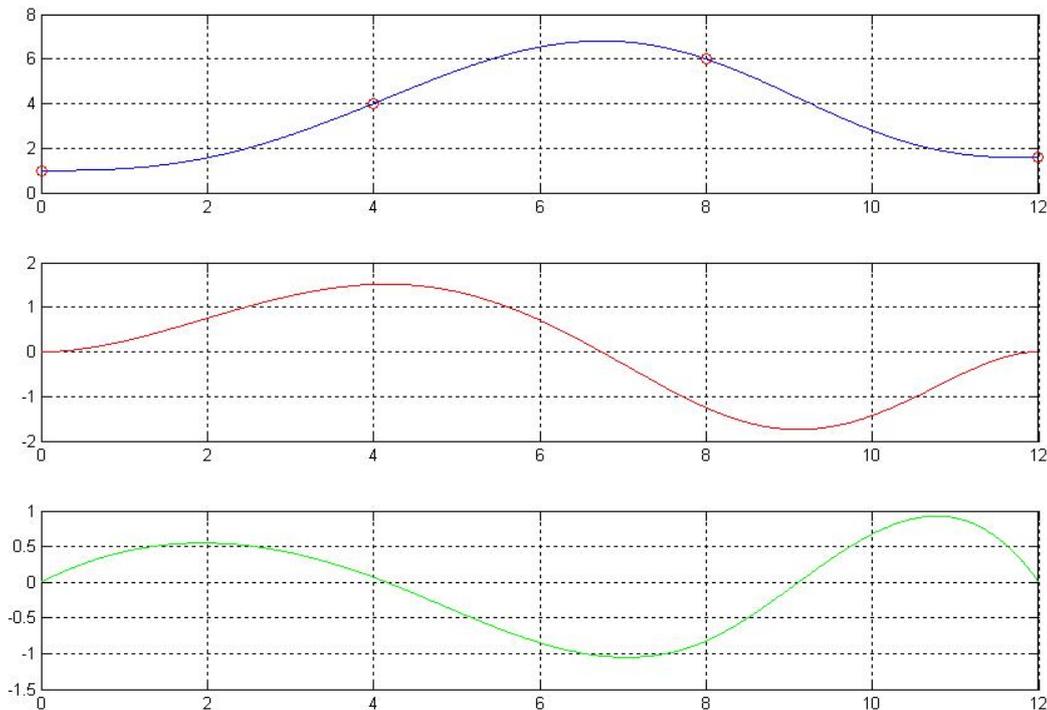


FIGURE 1.5 – Exemple sur la génération de trajectoire

## 1.2 Formulation du problème d'optimisation

Il s'agit de synthétiser ce trajectoire, pour transférer l'effecteur d'un robot d'un point initial à point final en temps minimum tenant compte le modèle du robot. Nous allons formalisée mathématiquement ce problème ci dessus

$$\min_{(T, q^*)} J = \int_0^T dt$$

sous contraintes :

- Les conditions aux limites

$$\begin{cases} q(0) = q_i \\ \dot{q}(0) = 0 \\ \ddot{q}(0) = 0 \end{cases} \quad \begin{cases} q(T) = q_f \\ \dot{q}(T) = 0 \\ \ddot{q}(T) = 0 \end{cases}$$

- La contrainte non holonome

$$M_{pa}\ddot{q}_a + M_{pp}\ddot{q}_p + C_p = 0$$

- Les contraintes géométriques

$$|q_a(t)| \leq \overline{q_{a,p}}$$

- Les contraintes technologiques

$$|\dot{q}_a(t)| \leq \overline{\dot{q}_a}$$

$$|\ddot{q}_a(t)| \leq \overline{\ddot{q}_a}$$

$$|\tau_a(t)| \leq \overline{\tau_a}$$

où  $T$  est la durée globale du mouvement,  $q^*$  les points de passage intermédiaire,  $\overline{q_{a,p}}$ ,  $\overline{\dot{q}_a}$ ,  $\overline{\ddot{q}_a}$ ,  $\overline{\tau_a}$ , débattement articulaire, vitesse, accélération et couples maximaux donnés par les actionneurs actifs.

### La procédure adoptée

Comme première étape il faut subdiviser l'intervalle de temps en parties égales pour diminuer le nombre de variables d'optimisation Figure1.6.

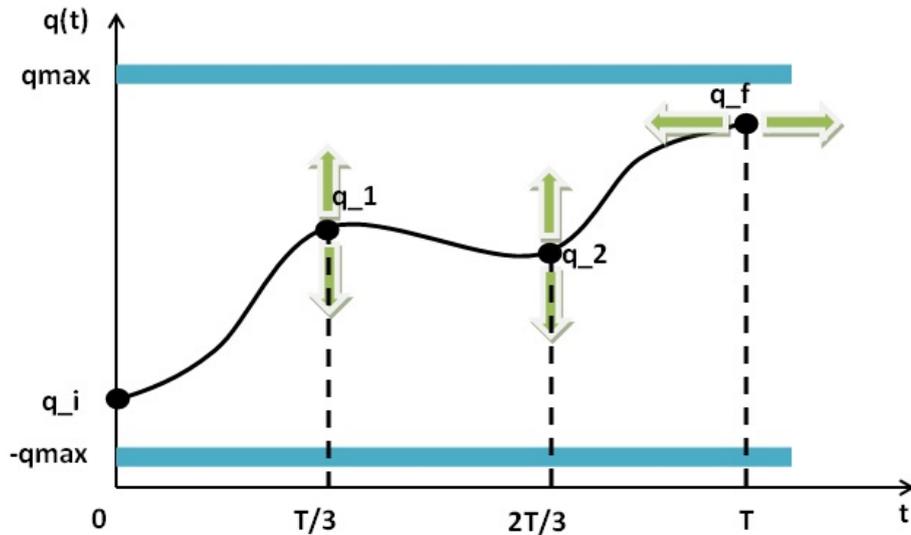


FIGURE 1.6 – Schématisation du problème d'optimisation

Par la suite il faut générer une trajectoire qui vérifie les conditions aux limites et qui passe par un nombre  $n$  de points intermédiaires tel que  $n \geq 2$  de façon qu'elle soit continue en position, vitesse, accélération comme nous l'avons vu précédemment. L'idée de ce travail est la même que pour un manipulateur complètement actionné décrit par [29, 14]. D'après la Figure 1.6 ces points intermédiaires varient entre  $-q_{max}$  et  $q_{max}$  (débattement articulaire), le vecteur de variables d'optimisation est donné par  $X = [T, q_1, q_2]$ . On est devant un problème d'optimisation dynamique, on peut avoir des minimums locaux à cause de la non convexité du domaine. Plusieurs méthodes existent pour résoudre ce genre de problème, parmi les méthodes utilisées dans ce mémoire la programmation séquentielle quadratique (SQP), les algorithmes génétiques (GA) et les essaims de particules (Particle Swarms), toutes ces méthodes seront détaillées dans le chapitre 4.

### 1.3 Conclusion

Le problème de génération de trajectoire est très lourd en terme de calcul, puisque pour chaque trajectoire générée il faut vérifier les conditions aux limites, les contraintes géométriques et les contraintes technologiques plus la contrainte dynamique d'égalité pour  $0 < t \leq T$ . Il n'est facile de vérifier cette dernière, pour que le problème d'optimisation converge vers une solution admissible il faut souvent relaxer la contrainte d'égalité sous forme d'inégalité.

# Chapitre 2

## Généralités sur les Systèmes Mécaniques Sous Actionnés

### 2.1 Introduction

Les systèmes Mécaniques Sous Actionnés (S.M.S.A) ont une grande importance dans l'industrie. Ces systèmes sont caractérisés par un nombre d'entrée de commande inférieur au nombre de degrés de liberté à contrôler, Le nombre de degrés de liberté étant la dimension du vecteur des coordonnées généralisées indépendantes.

Un système mécanique complètement actionnée peut devenir sous actionnée après une durée de fonctionnement, par exemple lors d'une panne d'actionneur, dans ce cas, au lieu de changer d'actionneur on bascule d'un algorithme de commande pour un système à actionneur complet vers un algorithme de contrôle pour les systèmes sous actionnés.

Ces systèmes offrent quelques avantages : la minimisation du coût, du poids et de l'énergie et permettent de gagner en temps.

### 2.2 Formalisme de Lagrange

Les équations de Lagrange ont été à l'origine employées pour décrire la dynamique des systèmes mécaniques, elles sont équivalentes à la modélisation par la loi de Newton. L'avantage du formalisme d'Euler-Lagrange est que la forme de ses équations est invariante.

Un système mécanique avec  $n$  degrés de liberté peut être décrit par les équations d'Euler-Lagrange suivantes :

$$\frac{d}{dt} \left[ \frac{\partial L}{\partial \dot{q}} \right] - \frac{\partial L}{\partial q} = F(q)u \quad (2.1)$$

ou  $q \in Q$  est le vecteur de configuration,  $u \in \mathbb{R}^m$  est le vecteur d'entrée et  $F(q) = (f_1(q), \dots, f_m(q))$  la matrice des forces extérieures et  $L(q, \dot{q})$  le Lagrangien défini par

$$L(q, \dot{q}) = K - V = \frac{1}{2} \dot{q}^T M(q) \dot{q} - V(q)$$

où  $K$  est l'énergie cinétique,  $V$  est l'énergie potentielle du système et  $M(q)$  est la matrice d'inertie symétrique définie positive.

Cette approche permet d'obtenir immédiatement les équations d'évolution en fonction des forces appliquées.

(2.1) peut se mettre sous la forme (2.2) :

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = F(q)u \quad (2.2)$$

Le terme  $C(q, \dot{q})\dot{q}$  contient deux types de termes contenant  $\dot{q}_i\dot{q}_j$  appelés termes Centrifuge ( $i = j$ ) et termes Coriolis ( $i \neq j$ ),  $G(q)$  contient les termes de gravité. Il est à noter que  $S_0 = \dot{M}(q) - 2C(q, \dot{q})$  est une matrice antisymétrique, ce qui nous permet d'avoir  $\dot{M}(q) = C(q, \dot{q}) + C^T(q, \dot{q})$ .

La transformation de Legendre par rapport à  $\dot{q}$  est donnée par :

$$P = \frac{\partial L}{\partial \dot{q}} = M(q)\dot{q}$$

Cette transformation est inversible et la dynamique du système mécanique (2.2) peut se mettre sous la forme canonique suivante :

$$\begin{cases} \dot{q} = M^{-1}(q)p \\ \dot{p} = -G(q) + \tilde{C}^T(q, p)M^{-1}(q)p + F(q)u \end{cases} \quad (2.3)$$

où  $\tilde{C}^T(q, p) = C(q, M^{-1}p)$ . On appelle (2.3) forme normale de Legendre pour un système mécanique. Définissons les variables  $x_1 = q, x_2 = p$ , (2.3) peut s'écrire sous une forme non linéaire affine en la commande

$$\dot{x} = f(x) + g(x)u \quad (2.4)$$

avec  $x = (x_1, x_2)^T$

L'avantage principal de la forme (2.4) est que plusieurs méthodes analytiques sont disponibles pour l'analyse de la commandabilité et de l'observabilité et la conception des contrôles pour les systèmes non linéaires affines en la commande.

## 2.3 Systèmes mécaniques complètement actionnés

Considérons le système mécanique décrit par (2.2). On appelle (2.1) un système mécanique complètement actionné si  $m = \text{rang}(F(q)) = n$ . Pour les systèmes complètement actionnés le nombre d'entrée de commande est égal à la dimension de leur espace de configuration. Par conséquent, les systèmes mécaniques complètement actionnés sont linéarisable par feedback, ceci peut être prouvé en appliquant le contrôle suivant :

$$u = F(q)^{-1}(M(q)v + C(q, \dot{q})\dot{q} + G(q))$$

et en redéfinissant les variables  $x_1 = q, x_2 = \dot{q}$  pour obtenir :

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= v \end{aligned}$$

qui représente clairement un système en double intégrateurs, voilà pourquoi la plupart des problèmes relatifs aux systèmes complètement actionnés peuvent être réduit à des problèmes équivalents pour systèmes linéaires.

## 2.4 Systèmes mécaniques sous actionnés

Un système mécanique de contrôle avec le vecteur de configuration  $q \in Q$  et le lagrangien  $L(q, \dot{q})$  satisfaisant les équations d'Euler-Lagrange (2.1) est appelé un système mécanique sous actionné si  $m = \text{rang}(F(q)) < n = \text{dim}(q)$ . En d'autres termes, les systèmes sous actionnés sont des systèmes mécaniques avec moins d'entrées de commande indépendantes que de degré de liberté à contrôler.

En raison de leur large applications, le contrôle des systèmes mécaniques sous actionnés est extrêmement important. La restriction sur le contrôle de ces systèmes produit des difficultés principales dans la conception de commande.

Comme cas spécial, supposons  $F(q) = (0, I_m)^T$ , alors les premières  $(n - m)$  équations dans (2.5) peuvent être exprimées comme des équations du seconde ordre qui donnent lieu à des contraintes non holonomes. le vecteur de configuration peut se diviser en  $q = (q_1, q_2) \in \mathbb{R}^{n-m} \times \mathbb{R}^m$  où  $q_1$  et  $q_2$  sont les vecteurs de configuration non actionné et actionné respectivement, donc la dynamique peut se mettre sous la forme vectorielle :

$$\begin{bmatrix} m_{11}(q) & m_{12}(q) \\ m_{21}(q) & m_{22}(q) \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} N_1(q, \dot{q}) \\ N_2(q, \dot{q}) \end{bmatrix} = \begin{bmatrix} 0 \\ u \end{bmatrix} \quad (2.5)$$

avec  $u \in \mathbb{R}^m$ .

### 2.4.1 Systèmes mécaniques non holonomes

On parle de deux types de système mécanique sous actionné :

- holonome (intégrable) : pas de restriction sur le déplacement, tout chemin de l'espace de configuration est un mouvement autorisé.
- non-holonome (non intégrable) : restriction sur le déplacement, il faut passer par un chemin imposé.

**Exemple 2.** Soit une roue qui roule sur un axe  $x$  Figure 2.1

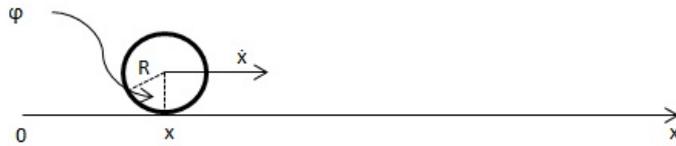


FIGURE 2.1 – Roue mobile

où  $R$  est le rayon de la roue,  $\phi$  l'angle de rotation,  $x$  la distance de déplacement de la roue et  $\dot{x}$  la vitesse correspond au centre de masse :

$$\begin{aligned} \dot{x} &= R\dot{\phi} \\ \dot{x} - R\dot{\phi} &= 0 \end{aligned} \quad (2.6)$$

Par intégration :  $x - R\phi - C = 0 \equiv F(x, \phi, t) = 0$  si on dérive la fonction différentiable  $F(x, \phi, t)$  par rapport aux variables  $(x, \phi, t)$  on trouve l'équation de vitesse d'où (2.6) est holonome.

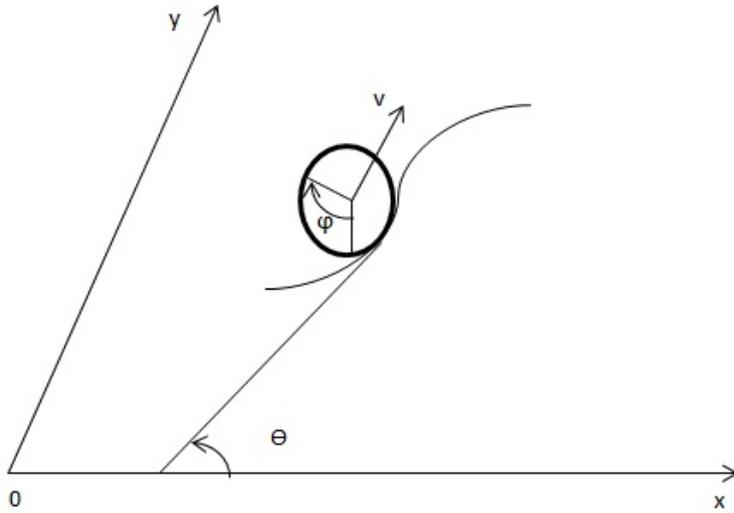


FIGURE 2.2 – Roue mobile sur le plan  $(x, y)$

**Exemple 3.** Soit une roue qui roule sur une surface curviligne dans le plan  $(x, y)$  Figure 2.2 où  $R, \phi$  et  $\theta$  sont respectivement : le rayon de la roue, l'angle de rotation et l'orientation de la roue par rapport à l'axe horizontal, les composantes de vitesse du centre de masse de la roue sont données par :

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \end{cases} \Leftrightarrow \begin{cases} \dot{x} = R \dot{\phi} \cos \theta \\ \dot{y} = R \dot{\phi} \sin \theta \end{cases}$$

$$\begin{cases} \dot{x} - R \dot{\phi} \cos \theta = 0 \\ \dot{y} - R \dot{\phi} \sin \theta = 0 \end{cases} \quad (2.7)$$

prenons la première composante de (2.7), la question qui se pose est la suivante : Est ce qu'il existe une fonction  $F(q)$ , avec  $q = (x, y, \phi, \theta, t)$  de telle sorte que

$$\frac{dF(q)}{dq} \dot{q} = \dot{x} - R \dot{\phi} \cos \theta?$$

En fait, on ne peut pas trouver cette fonction d'où l'équation (2.7) est définie comme contrainte non holonome [11][6].

Dans les systèmes mécaniques sous actionnés on distingue deux cas différents de contraintes non holonomes :

- contraintes non holonomes du premier ordre ou de vitesse, duent à la cinématique du système comme nous avons vu dans l'exemple précédent, on peut trouver ce cas dans les Robots mobiles.
- contraintes non holonomes du second ordre ou d'accélération, duent à la dynamique de système, ce cas existe souvent dans les bras manipulateurs.

## Contraintes non holonomes du premier ordre

Un système mécanique avec des contraintes non holonomes du premier ordre est un système Lagrangien contenant  $m$  contraintes de vitesse ( $m < n$ )

$$W^T(q)\dot{q} = 0 \quad \text{avec } W \text{ matrice } (n \times m)$$

non intégrables (c'est à dire  $\nexists \varphi(q, t) : \dot{\varphi} = W^T(q)\dot{q}$ ). Les contraintes non holonomes de l'exemple précédent peuvent s'écrire sous la forme :

$$\begin{bmatrix} 1 & 0 & -R \cos \theta \\ 0 & 1 & -R \sin \theta \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = 0 \quad (2.8)$$

telle que :

$$W(q) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -R \cos \theta & -R \sin \theta \end{bmatrix} \text{ et } q = \begin{bmatrix} x \\ y \\ \phi \end{bmatrix}$$

Ces systèmes sont caractérisés par l'existence des contraintes cinématiques non intégrables.

Les système unicycle et les robots mobiles constituent les exemples les plus répandus, pour plus detail consulter [21, 1] .

## Contraintes non holonomes du second ordre

Considérons un système mécanique sous actionné, notons  $q = (q_1, \dots, q_n)$  l'ensemble de coordonnées généralisées. Séparant l'ensemble de coordonnées généralisées sous la forme  $q = (q_a, q_b)$ , où  $q_a \in \mathbb{R}^m$  est la partie directement actionnée et  $q_b \in \mathbb{R}^{n-m}$  la partie sous actionnée avec  $u \in \mathbb{R}^m$  vecteur de commande, les équations de mouvement pour un système mécanique sous actionné sont données par :

$$M_{11}(q)\ddot{q}_a + M_{12}(q)\ddot{q}_b + F_1(q, \dot{q}) = B(q)u \quad (2.9)$$

$$M_{21}(q)\ddot{q}_a + M_{22}(q)\ddot{q}_b + F_2(q, \dot{q}) = 0 \quad (2.10)$$

L'équation (2.10) donne  $n - m$  relations non intégrables, c'est à dire il n'existe pas une fonction de la forme  $\sigma(t, q, \dot{q})$  telle que  $d\sigma/dt = 0$ , alors ces  $n - m$  relations peuvent être interprétées comme contraintes non holonomes du second ordre, ou contraintes d'accélération pour plus de détail consulter [8, 26].

Dans le paragraphe suivant nous allons rappeler quelques méthodes de commande des systèmes mécaniques sous actionnés.

## 2.5 La linéarisation partielle par feedback

Une propriété intéressante valable pour les systèmes mécaniques sous actionnés est la linéarisation partielle par feedback cette procédure comprend plusieurs types de linéarisation appelées linéarisation partielle localisée, linéarisation partielle non localisée et linéarisation partielle sous entrée couplées, définies comme suit :

### 2.5.1 Linéarisation partielle localisée (Collocated partial feedback linearization) [28][25]

En raison de l'absence du contrôle dans la première équation (2.5), il n'est pas possible de linéariser complètement ce système sous actionné par l'utilisation d'un changement de contrôle. Cependant il est toujours possible de linéariser partiellement le système tel que la dynamique de  $q_2$  se transforme en double intégrateurs.

Le vecteur de configuration actionné pour le système sous actionné (2.5) est  $q_2$ . Le procédé de la linéarisation de la dynamique de  $q_2$  s'appelle "collocated partial feedback linearization", cette linéarisation partielle peut être considérée comme une linéarisation entrée-sortie en ce qui concerne la sortie  $y = q_2$  et on peut montrer que tous les systèmes sous actionnés qui ont de la forme (2.5) peuvent être partiellement linéarisés en utilisant un changement de contrôle global.

**Proposition 1.** *Il existe un changement de contrôle global (inversible) sous la forme  $u = \alpha(q)\tau + \beta(q, \dot{q})$  qui linéarise partiellement la dynamique de (2.5) :*

$$\begin{aligned}\dot{q}_1 &= p_1 \\ \dot{p}_1 &= f_0(q, p) + g_0(q)\tau \\ \dot{q}_2 &= p_2 \\ \dot{p}_2 &= \tau\end{aligned}\tag{2.11}$$

où  $\alpha(q)$  est une matrice symétrique ( $m \times m$ ) définie positive et  $g_0(q) = -m_{11}^{-1}m_{12}(q)$

**Preuve 1.** *De la première ligne de (2.5) nous avons*

$$\ddot{q}_1 = -m_{11}^{-1}(q)N_1(q, \dot{q}) - m_{11}^{-1}(q)m_{12}(q)\ddot{q}_2$$

ce qui donne l'expression de  $g_0$ , en remplaçant cette équation dans la deuxième ligne de (2.5), on obtient :

$$(m_{22}(q) - m_{21}(q)m_{11}^{-1}m_{12}(q))\ddot{q}_2 + N_2(q, \dot{q}) - m_{21}(q)m_{11}^{-1}(q)N_1(q, \dot{q}) = u$$

donc

$$\begin{aligned}\alpha(q) &= (m_{22}(q) - m_{21}(q)m_{11}^{-1}m_{12}(q)) \\ \beta(q, \dot{q}) &= N_2(q, \dot{q}) - m_{21}(q)m_{11}^{-1}(q)N_1(q, \dot{q})\end{aligned}$$

comme on peut l'observer  $\alpha(q)$  est symétrique définie positive.

### 2.5.2 Linéarisation partielle non localisée (Noncollocated partial feedback linearization)

Dans cette section, nous introduisons une procédure de linéarisation partielle qui linéarise la dynamique de la configuration non actionnée. Il a été montré que ceci est possible si le nombre d'entrée de commande est supérieur ou égal au nombre des variables de configuration non actionnée, on appelle cette procédure "noncollocated partial feedback linearization". Pour être plus précis, considérons le système sous actionné suivant

$$\begin{bmatrix} m_{11}(q) & m_{12}(q) & m_{13}(q) \\ m_{21}(q) & m_{22}(q) & m_{23}(q) \\ m_{31}(q) & m_{32}(q) & m_{33}(q) \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \begin{bmatrix} N_1(q, \dot{q}) \\ N_2(q, \dot{q}) \\ N_3(q, \dot{q}) \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ 0 \end{bmatrix}\tag{2.12}$$

où  $q = (q_1, q_2, q_3)$  représente le vecteur de coordonnées généralisées.

**Proposition 2.** *considérons le système sous actionné (2.12), alors, il existe un changement de contrôle sous la forme  $u = \alpha_1(q)\tau + \beta_1(q, \dot{q})$  qui linéarise partiellement la dynamique de (2.12) dans l'ensemble  $U = \{q \in \mathbb{R}^n / \det(m_{32}(q)) \neq 0\}$*

$$\begin{aligned}
\dot{q}_1 &= p_1 \\
\dot{p}_1 &= \tau_1 \\
\dot{q}_2 &= p_2 \\
\dot{p}_2 &= f_0(q, p) + g_0(q)\tau_1 + g_2(q)\tau_3 \\
\dot{q}_3 &= p_3 \\
\dot{p}_3 &= \tau_3
\end{aligned} \tag{2.13}$$

où  $\tau = (\tau_1, \tau_2)$ ,  $u = (u_1, u_2)$  et  $\tau_2 = \alpha_0(q)\tau_1 + \alpha_2(q)\tau_3 + \beta_2(q, \dot{q})$  avec

$$\begin{aligned}
f_0(q, p) &= -m_{32}^{-1}(q)N_3(q, \dot{q}) \\
g_0(q) &= -m_{32}^{-1}(q)m_{31}(q) \\
g_2(q) &= -m_{32}^{-1}(q)m_{33}(q)
\end{aligned}$$

pour plus de détails, voir [25, 28].

### 2.5.3 Linéarisation partielle sous des entrées couplées

Considérons le système sous actionné (2.1) avec le vecteur de configuration  $q \in \mathbb{R}^n$  et le contrôle  $u \in \mathbb{R}^m$  et  $\text{rang}F(q) = m < n$ .  $F(q)$  peut être écrit comme :

$$\begin{bmatrix} F_1(q) \\ F_2(q) \end{bmatrix}$$

tel que  $F_2(q)$  est une matrice  $(m \times m)$  inversible et  $q$  peut être décomposé en  $(q_1, q_2) \in \mathbb{R}^{n-m} \times \mathbb{R}^m$ . La définition des entrées couplées implique que  $F_1(q) \neq 0$  pour tout  $q$ . Dans la suite, nous fournissons des condition permettant la linéarisation partielle pour le cas de ces système sous actionnés.

**Proposition 3.** *considérons le système sous actionné avec des entrées couplées c'est-à-dire  $(F_1(q) \neq 0, \det(F_2(q)) \neq 0$  pour tous  $q$ )*

$$\begin{bmatrix} m_{11}(q) & m_{12}(q) \\ m_{21}(q) & m_{22}(q) \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} N_1(q, \dot{q}) \\ N_2(q, \dot{q}) \end{bmatrix} = \begin{bmatrix} F_1(q) \\ F_2(q) \end{bmatrix} u \tag{2.14}$$

et supposons que la matrice  $\Lambda = F_2(q) - m_{21}m_{11}^{-1}(q)F_1(q)$  est inversible pour tout  $q$  alors, il existe un changement de contrôle  $u = \alpha(q)\tau + \beta(q, \dot{q})$  qui linéarise partiellement (2.14) sous la forme

$$\begin{aligned}
\dot{q}_1 &= p_1 \\
\dot{p}_1 &= f_0(q, p) + g_0(q)\tau \\
\dot{q}_2 &= p_2 \\
\dot{p}_2 &= \tau
\end{aligned} \tag{2.15}$$

où

$$\begin{aligned}
\alpha(q) &= \Lambda^{-1}(q) [m_{22}(q) - m_{21}(q)m_{11}^{-1}(q)m_{12}(q)] \\
\beta(q, \dot{q}) &= \Lambda^{-1}(q) [N_2(q, \dot{q}) - m_{21}(q)m_{11}^{-1}(q)N_1(q, \dot{q})] \\
f_0(q, p) &= m_{11}^{-1} [F_1(q)\beta(q, \dot{q}) - N_1(q, \dot{q})] \\
g_0(q) &= m_{11}^{-1}(q) [F_1(q)\alpha(q) - m_{21}(q)]
\end{aligned}$$

Pour plus de détails sur la démonstration de ces deux dernières proposition consulter [25].

**Remarque 1.** La procédure de de la linéarisation partielle dans la proposition (3) est particulièrement utilisée pour les systèmes sous actionnés autonomes comme les avions ou les hélicoptères avec six degrés de liberté.

## 2.6 Exemples des systèmes mécaniques sous actionnés

Dans cette section, nous allons donner quelques exemples des systèmes mécaniques sous actionnés, ces exemples incluent la boule et la poutre, le chariot-pendule inversé, l'acrobot et le pendubot.

### 2.6.1 La boule et la poutre (beam and ball)

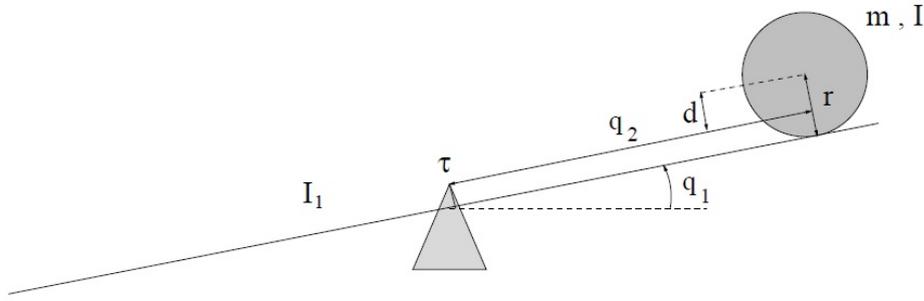


FIGURE 2.3 – La boule et la poutre

ce système est composé d'une poutre pouvant pivoter dans le plan vertical par l'application d'un couple  $\tau$  au point de rotation (le centre) et d'une balle dont on restreindra le mouvement à un glissement sans frottement le long de la poutre.

Dans cette exemple on veut commander deux sorties (deux degrés de liberté), la position angulaire  $q_1$  de la poutre et la position  $q_2$  de la boule par la seule commande  $u$  disponible appliquée au point de rotation de la poutre. Donc, le nombre de sortie qu'on veut commander est supérieur au nombre de commande dont on dispose, ce caractère se nomme "sous actionné".

Soit  $d$  la distance entre le centre de masse de la boule et la poutre ( $d = r$  dans la Figure 2.3) où le modèle conventionnel de ce système correspond à  $d = 0$ . La matrice d'inertie  $M(q)$  et l'énergie potentielle  $V(q)$  sont données par :

$$\begin{bmatrix} I_1 + m(q_2^2 + d^2) & -md \\ -md & m\Lambda \end{bmatrix} \text{ et } V(q) = mg(q_2 \sin q_1 + d \cos q_1)$$

où  $I_1$  est l'inertie de la poutre,  $m, I_2$  la masse et l'inertie de la boule respectivement avec  $\Lambda = 1 + \frac{I_2}{mr^2}$

### 2.6.2 Chariot-pendule inversé

Le système chariot-pendule inversé représenté par la Figure 2.4 est un système mécanique sous actionné constitué d'un chariot pouvant se déplacer sur une surface plane et

d'un pendule connecté à travers une charnière sur ce chariot, le tout est commandé par un moteur électrique. De même que l'exemple précédant, la nécessité de commander  $q_1$  et  $q_2$  avec un seul contrôle  $F$  classe ce système dans la catégorie des systèmes mécaniques sous actionnés.

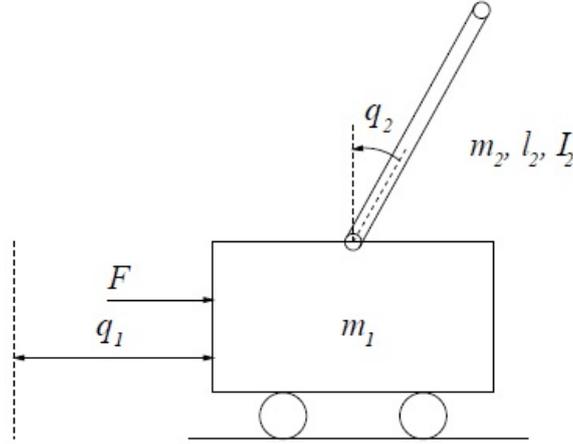


FIGURE 2.4 – Chariot-pendule inversé

La matrice d'inertie  $M(q)$  et l'énergie potentielle  $V(q)$  sont données par :

$$M(q) = \begin{bmatrix} m_1 + m_2 & m_2 l_2 \cos q_2 \\ m_2 l_2 \cos q_2 & m_2 l_2^2 + I_2 \end{bmatrix} \quad \text{et} \quad V(q) = m_2 g l_2 \cos q_2$$

où  $m_1$  masse du chariot et  $m_2, l_2, I_2$  sont la masse, la longueur du centre de masse et l'inertie du pendule respectivement.

### 2.6.3 Acrobot et pendubot

Considérons un robot à deux bras avec un seul actionneur, la mise en action d'une variable (c'est-à-dire  $q_1$  ou  $q_2$ ) donne deux systèmes mécaniques sous actionnés différents, Acrobot Figure 2.5 où  $q_2$  est actionné et Pendubot Figure 2.5 où  $q_1$  est actionné. En réalité, ces deux systèmes mécaniques représentent le même système (ils sont complètement actionnés), la survenue d'une panne ou bien le choix volontaire de supprimer un actionneur donne lieu à deux structures différentes.

La matrice d'inertie des deux systèmes est donnée par

$$\begin{aligned} m_{11} &= I_1 + I_2 + m_1 l_1^2 + m_2 (L_1^2 + L_2^2) + 2m_2 L_1 l_2 \cos q_2 \\ m_{12} &= m_{21} = I_2 + m_2 l_2^2 + m_2 L_1 l_2 \cos q_2 \\ m_{22} &= I_2 + m_2 l_2^2 \end{aligned}$$

et l'énergie potentielle correspondante  $V(q)$  est donnée par

$$V(q) = (m_1 l_1 + m_2 L_1) g \cos q_1 + m_2 l_2 g \cos(q_1 + q_2)$$

où  $m_i, L_i, l_i$  sont la masse, l'inertie, la longueur et la longueur du centre de masse du  $i^{eme}$  bras.

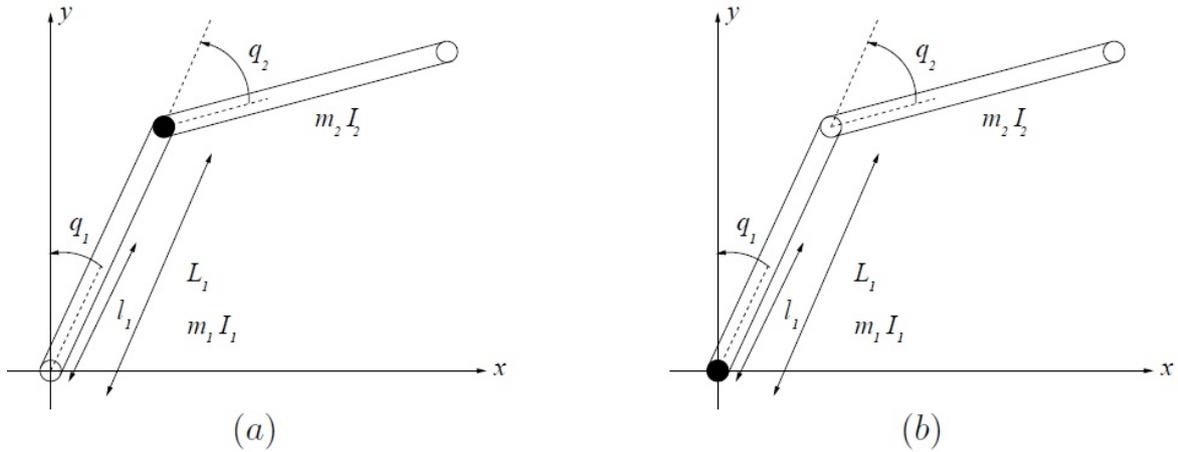


FIGURE 2.5 – (a) Acrobot, (b) Pendubot

Voir [17] pour plus de détails sur la modélisation de ces systèmes.

## 2.7 Conclusion

Les systèmes mécaniques en général présentent des caractéristiques importantes du point de vue de l'analyse et de la commande, en particulier les contraintes non holonomes qui limitent le mouvement et les contraintes du second ordre qui sont une obstruction à la linéarisation entrée/sortie par feedback.

Le nombre de coordonnées généralisées être supérieur au nombre de commande, la synthèse de lois de commande est très difficile pour ce genre de système.

Dans les chapitres suivants, nous allons étudier des méthodes de résolution des problèmes de planification de trajectoire et approfondir les aspects théoriques concernant ces systèmes.

# Chapitre 3

## Systemes Mécaniques Non holonomes [12]

Dans ce chapitre, nous reprenons les résultats d’Oriolo et De Luca donnés dans ‘modelling and control of nonholonomic mechanical systems’.

### 3.1 Introduction

Considérons un système mécanique dont la configuration est décrite par le vecteur de coordonnées généralisées  $q \in Q$ . L’espace de configuration  $Q$  est une variété lisse de dimension  $n$ , localement difféomorphe à sous ensemble ouvert de  $\mathbb{R}^n$ . Soit la trajectoire  $q(t) \in Q$ , la vitesse généralisée à la configuration  $q$  est le vecteur  $\dot{q}$  appartenant à l’espace tangent  $T_q(Q)$ .

Dans de nombreux cas intéressants, le mouvement du système est soumis à des contraintes qui peuvent apparaître de la structure elle-même du mécanisme, ou de la façon dont il est actionné et contrôlé. Diverses classifications de ces contraintes peuvent être distinguées. Par exemple, les contraintes peuvent être exprimées sous forme d’égalités ou d’inégalités (respectivement, bilatérales ou unilatérales) et elles peuvent dépendre explicitement du temps ou pas.

Dans la discussion non exhaustive ci-dessous nous allons traiter uniquement les contraintes bilatérales.

Les restrictions de mouvement qui peuvent être mises sous la forme

$$h_i(q) = 0 \quad i = 1, \dots, k < n \tag{3.1}$$

sont appelées contraintes holonomes. Par convention, les fonctions  $h_i : Q \rightarrow \mathbb{R}$  sont supposées lisses et indépendantes. Un système dont les contraintes sont toutes holonomes, est appelé un système holonome.

L’effet de contraintes (3.1) est de limiter les configurations du systèmes atteignables à une sous variété lisse de  $Q$  de dimension  $(n - k)$ . Une manière simple pour faire face aux contraintes holonomes est fournie par le théorème des fonctions implicites, qui permet de résoudre les équations (3.1) en termes de  $n - k$  coordonnées généralisées, afin d’éliminer les variables  $k$  restantes du problème. En général, cette procédure n’a qu’une validité locale et peut introduire des singularités algébriques. Plus pratique, la configuration du système peut être décrite par  $n - k$  nouvelles coordonnées sur la sous-variété restreinte, qui caractérisent les degrés de liberté réels du système. L’étude du mouvement de ce système réduit est complètement équivalente à celle d’origine. Pour des raisons de simulation,

une approche alternative est de garder les équations de contraintes en tant que telles et d'utiliser un solveur d'équation algèbre différentielles.

Les contraintes holonomes sont généralement introduites par les interconnexions mécaniques entre les différents organes du système. Par exemple, les articulations prismatiques et rotoïdes couramment utilisées dans les robots manipulateurs sont une source de contraintes. Si nous considérons une chaîne cinématique à base fixe composée de  $n$  liens rigides reliés par des articulations élémentaires, l'espace de configuration composite du système est  $(\mathbb{R}^3 \times SO(3))^n$ . Étant donné que chaque articulation impose cinq contraintes, le nombre de degrés de liberté est  $6n - 5n = n$ . Nous mentionnons qu'il est possible de concevoir des robots manipulateurs avec des articulations qui ne sont pas holonomes, comme proposé dans [18].

Les contraintes du système, dont l'expression implique les coordonnées généralisées et des vitesses sous la forme

$$a_i(q, \dot{q}) = 0 \quad i = 1, \dots, k < n, \quad (3.2)$$

sont considérés comme des contraintes cinématiques. Ces contraintes limiteront les mouvements admissibles du système en limitant l'ensemble des vitesses généralisées qui peuvent être atteintes à partir d'une configuration donnée. En mécanique, ces contraintes sont habituellement rencontrées dans la forme de Pfaffian

$$a_i^T(q)\dot{q} = 0, \quad i = 1, \dots, k < n, \quad \text{Où } A^T(q)\dot{q} = 0, \quad (3.3)$$

linéaire par rapport aux coordonnées de vitesses. Les vecteurs  $a_i : Q \rightarrow \mathbb{R}^n$  sont supposés être lisses et linéairement indépendants.

Bien sûr, les contraintes holonomes (3.1) impliquent l'existence de contraintes cinématiques exprimées en

$$\frac{\partial h_i}{\partial q} \dot{q} = 0, \quad i = 1, \dots, k.$$

Cependant, l'inverse n'est pas nécessairement vrai, il peut arriver que les contraintes cinématiques (3.3) ne soient pas intégrables, c'est à dire, ne peuvent pas être mises sous la forme (3.1). Dans ce cas, les contraintes et le système mécanique sont appelés non holonomes.

La présence de contraintes non holonomes limite la mobilité du système d'une manière complètement différente si on la compare à des contraintes holonomes. Pour illustrer ce point, considérons une contrainte simple Pfaffian

$$a^T(q)\dot{q} = 0. \quad (3.4)$$

Si la contrainte (3.4) est holonome, alors elle peut être intégrée comme

$$h(q) = c$$

où  $\partial h / \partial q = a^T(q)$  et  $c$  est une constante d'intégration. Dans ce cas, le système de mouvement est limitée à une surface particulière de  $h$ , selon l'état initial par la valeur de  $c = h(q_0)$ .

Supposons maintenant que la contrainte (3.4) est non holonome. Même si la mobilité instantanée du système est limitée à un espace de dimension  $(n - 1)$  il est encore possible d'atteindre n'importe quelle configuration dans  $Q$ . Parallèlement, le nombre de degrés de liberté est réduit à  $n - 1$ , mais le nombre de coordonnées généralisées ne peut pas être

réduit. Cette conclusion est générale : Pour un système mécanique avec  $n$  coordonnées généralisées et  $k$  contraintes non holonomes, bien que les vitesses généralisées à chaque point sont confinées à un sous espace de dimension  $(n - k)$ , l'accessibilité de l'espace de configuration est conservée.

**Exemple 4.** *Considérons un disque qui roule sans glisser sur un plan, comme indiqué dans la Fig 3.1. Sa configuration est complètement décrite par quatre variables : Les coordonnées de position  $(x, y)$  du point de contact avec le sol dans un repère fixe, l'angle  $\theta$  caractérise l'orientation de disque par rapport à l'axe des  $x$  et  $\phi$  l'angle entre l'axe choisi radial sur le disque et l'axe vertical*

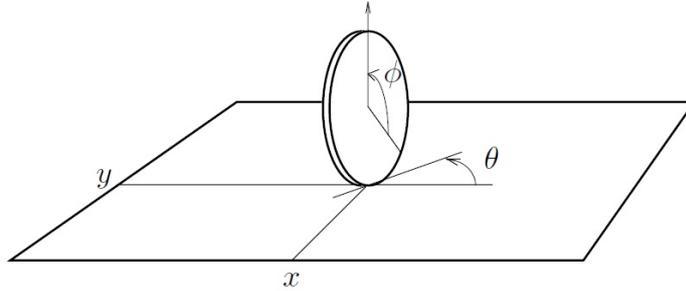


FIGURE 3.1 – Coordonnées généralisées d'un disque roulant

A cause de la contrainte sans glissement, les vitesses du système généralisé ne peuvent pas prendre des valeurs arbitraires. En particulier, en notant  $\rho$  le rayon du disque, ils doivent satisfaire les contraintes

$$\dot{x} - \rho \cos \theta \dot{\phi} = 0 \quad (3.5)$$

$$\dot{y} - \rho \sin \theta \dot{\phi} = 0 \quad (3.6)$$

exprimant ainsi la condition que la vitesse du centre du disque se trouve dans le plan médian du disque (les coordonnées de vitesse sur le plan au point de contact).

Les contraintes ci-dessus cinématiques ne sont pas intégrables et, par conséquent, il n'y a pas de limitation sur les configurations qui peuvent être atteintes par le disque. En fait, le disque peut être piloté à partir d'une configuration  $(x_1, y_1, \theta_1, \phi_1)$  à une configuration  $(x_2, y_2, \theta_2, \phi_2)$  grâce à la séquence de mouvements suivants :

1. Rouler le disque afin d'amener le point de contact de  $(x_1, y_1)$  à  $(x_2, y_2)$  quelque soit la courbe de la longueur  $\rho.(\phi_2 - \phi_1 + 2k\pi)$ , où  $k$  est un entier positif arbitraire.
2. Tournez le disque autour de l'axe vertical de  $\theta_1$  à  $\theta_2$ .

Cela confirme que les deux contraintes imposées sur le mouvement du disque roulant sont non holonomes.

Il devrait être clair que, la présence des contraintes cinématiques, il est essentiel de décider de leur intégrabilité. Nous allons aborder ce problème dans la section suivante.

## 3.2 Intégrabilité des Contraintes

Commençons par examiner le cas d'une contrainte de Pfaffian seule

$$a^T(q)\dot{q} = \sum_{j=1}^n a_j(q)\dot{q}_j = 0 \quad (3.7)$$

Pour que la contrainte (3.7) soit intégrable, il doit exister un facteur d'intégration (non nul)  $\gamma(q)$  tel que

$$\gamma(q)a_j(q) = \frac{\partial h}{\partial q_j}(q), \quad j = 1, \dots, n, \quad (3.8)$$

pour certaines fonctions  $h(q)$ . L'inverse est également vrai : s'il existe un  $\gamma(q)$  tel que  $\gamma(q)a(q)$  est le vecteur gradient de certaines fonctions  $h(q)$ , alors la contrainte (3.7) est intégrable. En utilisant le théorème de Schwarz, la condition d'intégrabilité (3.8) peut être remplacée par

$$\frac{\partial \gamma a_k}{\partial q_j} = \frac{\partial \gamma a_j}{\partial q_k}, \quad j, k = 1, \dots, n, \quad (3.9)$$

qui ne comportent pas la fonction inconnue  $h(q)$ . Notez que les conditions (3.9) impliquent que les contraintes cinématiques linéaires (c'est à dire avec des  $a_j$  constantes) sont toujours intégrables.

**Exemple 5.** Pour la contrainte différentielle suivante dans  $\mathbb{R}^3$

$$\dot{q}_1 + q_1 \dot{q}_2 + \dot{q}_3, \quad (3.10)$$

les conditions d'intégrabilité (3.9) deviennent

$$\begin{aligned} \frac{\partial \gamma}{\partial q_2} &= \gamma + \frac{\partial \gamma}{\partial q_1} \\ \frac{\partial \gamma}{\partial q_3} &= \frac{\partial \gamma}{\partial q_1} \\ \frac{\partial \gamma}{\partial q_3} q_1 &= \frac{\partial \gamma}{\partial q_2} \end{aligned}$$

En substituant les deuxième et troisième équations dans la première, il est possible de voir que la seule solution est  $\gamma \equiv 0$ . Par conséquent, la contrainte n'est pas intégrable. Lorsqu'on a plusieurs contraintes cinématiques de la forme (3.3), la non holonomie de chaque contrainte considérée séparément n'est pas suffisante pour en déduire que l'ensemble des contraintes est non holonomes. En fait, il peut encore arriver que  $p \leq k$  combinaisons linéaires indépendantes des contraintes

$$\sum_{i=1}^k \gamma_{ji}(q) a_i^T(q) \dot{q}, \quad j = 1, \dots, p,$$

soient intégrables. Dans ce cas, il existe  $p$  fonctions indépendantes de  $h_j(q)$  telles que

$$\text{span} \left\{ \frac{\partial h_1}{\partial q}(q), \dots, \frac{\partial h_p}{\partial q}(q) \right\} \subset \text{span} \{ a_1^T(q), \dots, a_k^T(q) \}, \quad \forall q \in Q,$$

et les configurations du système sont limitées à une variété de dimension  $(n-p)$  identifiée par des surfaces plates des  $h_j$ , soit

$$\{ q \in Q : h_1(q) = c_1, \dots, h_p(q) = c_p \},$$

sur laquelle le mouvement est lancé.

Dans le cas particulier  $p = k$ , l'ensemble des contraintes différentielles est complètement équivalent à un ensemble de contraintes holonomes, donc il est lui-même holonome.

**Exemple 6.** Soient les deux contraintes

$$\dot{q}_1 + q_1 \dot{q}_2 + \dot{q}_3 = 0 \quad (3.11)$$

et

$$\dot{q}_1 + \dot{q}_2 + q_1 \dot{q}_3 = 0 \quad (3.12)$$

elles ne sont pas intégrables séparément (en particulier, la première est la contrainte non holonome de l'exemple précédent). Mais lorsqu'elles sont prises ensemble, par des manipulations simples elles peuvent être mises sous la forme :

$$\begin{aligned} \dot{q}_1 + (q_1 + 1)\dot{q}_2 &= 0 \\ \dot{q}_1 + (q_1 + 1)\dot{q}_3 &= 0, \end{aligned}$$

qui est trivialement intégrable, donnant

$$\begin{aligned} \dot{q}_2 + \log(q_1 + 1) &= c_1 \\ q_2 - q_3 &= c_2, \end{aligned}$$

où les  $c_i$  sont des constantes.

Si  $1 \leq p < k$ , l'ensemble des contraintes (3.3) est non holonome selon la définition qui précède. Toutefois, afin de souligner qu'un sous-ensemble de (3.3) est intégrable, nous nous référerons à cette situation comme partiellement non holonome, par opposition on se trouve dans un cas complètement non holonome ( $p = 0$ ).

**Exemple 7.** Considérons les trois contraintes suivantes dans  $\mathbb{R}^6$

$$A_1^T(q) \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} + A_2^T \begin{bmatrix} \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{bmatrix} = 0, \quad (3.13)$$

$$A_1(q) = \begin{bmatrix} \frac{\sqrt{3}}{2} \cos q_3 - \frac{1}{2} \sin q_3 & \sin q_3 & -\frac{1}{2} \sin q_3 - \frac{\sqrt{3}}{2} \cos q_3 \\ \frac{1}{2} \cos q_3 + \frac{\sqrt{3}}{2} \sin q_3 & -\cos q_3 & \frac{1}{2} \cos q_3 - \frac{\sqrt{3}}{2} \sin q_3 \\ l & l & l \end{bmatrix}$$

et

$$A_2 = \begin{bmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & r \end{bmatrix}$$

Cet ensemble de contraintes n'est pas holonome, mais il est partiellement intégrable. En fait, si on les additionne, on obtient

$$\dot{q}_3 + \frac{r}{3l}(\dot{q}_4 + \dot{q}_5 + \dot{q}_6) = 0,$$

qui peut être intégré comme suit :

$$q_3 = -\frac{r}{3l}(q_4 + q_5 + q_6),$$

L'ensemble des contraintes (3.13) caractérise la cinématique d'un robot mobile à trois roues symétriques omnidirectionnel. En particulier,  $q_1$  et  $q_2$  sont les coordonnées cartésiennes du centre de robot par rapport à un repère fixe,  $q_3$  est l'orientation du véhicule,

tandis que  $q_4, q_5, q_6$  mesurent l'angle de rotation des trois roues. En outre,  $r$  est le rayon de la roue et  $a$  est la distance entre le centre du robot au centre de chaque roue. L'intégrabilité partielle des contraintes indique que l'orientation du véhicule est une fonction des angles de rotation des roues, et donc, peut être éliminée de la formulation du problème.

A ce stade, la question de l'intégrabilité de multiples contraintes cinématiques n'est pas évidente. Cependant, les critères d'intégrabilité peuvent être obtenus sur la base d'un point de vue différent, qui est introduit dans le reste de cette section.

L'ensemble des  $k$  contraintes de Pfaffian définit, à chaque configuration  $q$ , les vitesses admissibles généralisées comme celles contenues dans le noyau de matrice  $A^T(q)$  de dimension  $(n - k)$ . De manière équivalente, si  $g_1(q), \dots, g_{n-k}(q)$  est une base de cet espace, toutes les trajectoires possibles pour le système mécanique sont obtenues comme solutions de

$$\dot{q} = \sum_{j=1}^m g_j(q)u_j = G(q)u \quad m = n - k \quad (3.14)$$

pour  $u(t)$  arbitraire. Cela peut être considéré comme un système de contrôle non linéaire avec comme vecteur d'état  $q \in \mathbb{R}^n$  et le contrôle  $u \in \mathbb{R}^m$ . En particulier le système (3.14) à une dérivée  $\dot{q} = 0$  lorsque aucune entrée n'est appliquée. De plus du point de vue mécanique, il est sous actionné alors il existe moins d'entrées que de coordonnées généralisées ( $m < n$ ).

Le choix de  $G(q)$  dans l'équation (3.14) n'est pas unique, par conséquent les composantes de  $u$  peuvent avoir différentes interprétations. En générale on peut choisir les colonnes de  $g_j$  de telle sorte que  $u_j$  aura une interprétation physique évidente. En outre, le vecteur d'entrée  $u$  n'a pas une interprétation en termes des commandes réelles du système mécanique qui sont en général les forces ou les couples des actionneurs. Pour cette raison l'équation (3.14) représente le modèle cinématique du système soumis à des contraintes.

Pour parler d'holonomie/ non holonomie de l'ensemble des contraintes cinématiques, il est utile d'étudier les propriétés de contrôlabilité pour le modèle cinématique associé, en fait

1. Si l'éq. (3.14) est contrôlable, étant donné deux points de  $q_1$  et  $q_2$  arbitraires dans  $Q$ , il existe un choix de  $u(t)$  qui amène le système de  $q_1$  à  $q_2$ . De manière équivalente, il existe une trajectoire  $q(t)$  de  $q_1$  à  $q_2$  qui satisfait les contraintes cinématiques. Par conséquent, ces dernières ne limitent pas l'accessibilité de l'ensemble de l'espace de configuration  $Q$ , et donc, elles sont complètement non holonomes.
2. Si l'éq. (3.14) n'est pas contrôlable, le raisonnement ci-dessus ne tient pas et les contraintes cinématiques impliquent une perte de l'accessibilité de l'espace de configuration du système. Ainsi, les contraintes sous-jacentes sont partiellement ou complètement intégrables, selon la dimension  $\nu (< n)$  de la région accessible. En particulier :
  - (a) Si  $\nu > m$ , la perte de l'accessibilité n'est pas maximale, ce qui signifie que l'équation (3.14) n'est que partiellement intégrable. Selon notre définition, le système est partiellement non-holonome.
  - (b) Si  $\nu = m$ , la perte d'accessibilité est maximale, et l'ensemble (3.14) est intégrable. Ainsi, le système est holonome.

Nous avons déjà adopté ce point de vue dans l'étude du disque roulant. En particulier, la contrôlabilité du système correspondant a été prouvée de manière constructive, c'est à dire, en exposant une stratégie de reconfiguration. Toutefois, pour utiliser

efficacement cette approche, il est nécessaire d'avoir des conditions de contrôlabilité pratiques pour vérifier le système de contrôle non-linéaire (3.14).

Pour ce faire, nous allons examiner les outils de la théorie du contrôle basés sur la géométrie différentielle. Ces outils s'appliquent aux systèmes de contrôle non linéaire de type

$$\dot{x} = f(x) + \sum_{j=1}^m g_j(x)q_j$$

Comme nous le verrons plus tard, la présence du terme de dérive  $f(x)$  caractérise les contraintes cinématiques sous une forme plus générale que (3.3), ainsi que le modèle dynamique des systèmes non holonomes.

### 3.2.1 Stabilisabilité

Le problème de la stabilisation d'un système de contrôle de la forme

$$\dot{x} = f(x) + \sum_{j=1}^m g_j(x)u_j, \quad (3.15)$$

consiste à trouver une loi de commande de la forme

$$u = \alpha(x) + \beta(x)v, \quad u, v \in \mathbb{R}^m,$$

de manière à rendre le point d'équilibre  $x_e$  en boucle fermée ou une trajectoire admissible  $x_e(t)$  asymptotiquement stable. L'adoption de lois de commande par feedback est particulièrement adaptée pour la commande de mouvement, pour contrebalancer la présence de perturbations, les erreurs initiales ou des erreurs de modélisation. Pour la stabilisation en un point, le point  $x_e$  est généralement un point d'équilibre pour le système en boucle ouverte, c'est à dire  $f(x_e) = 0$ . En effet, pour le système de contrôle sans dérive (A.8), tout point est un point d'équilibre en boucle ouverte. Comme pour le cas de suivi de trajectoire, il est nécessaire d'assurer que les trajectoires à stabiliser sont admissibles pour le système. Ceci est d'une importance particulière dans le cas des systèmes non holonome, pour lesquels les contraintes cinématiques excluent la possibilité de suivre une trajectoire générique.

Dans la discussion ci-dessous, nous nous référons uniquement au problème de la stabilisation en un point. Dans les systèmes linéaires, la contrôlabilité implique la stabilisabilité asymptotique (en fait, exponentielle) par retour d'état continu. En fait, si la condition (A.12) est satisfaite, il existe des choix de  $K$  tels que le contrôle linéaire

$$u = K(x_e - x)$$

rend  $x_e$  asymptotiquement stable.

Pour les systèmes non linéaires, cette implication n'est plus vraie. Des résultats locaux peuvent être obtenus à l'aide de la linéarisation approximative du système (A.6) dans un voisinage de  $x_e$

$$\dot{\delta}(x) = \frac{\partial f}{\partial x}(x_e)\delta(x_e) + \sum_{j=1}^m g_j(x_e)u_j = A_e\delta(x) + B_e\delta(u), \quad (3.16)$$

avec  $\delta(x) = (x - x_e)$  et  $\delta(u - u_e) = u$ . En particulier, si le système (3.16) est commandable, le système non linéaire (A.6) peut être localement stabilisé au point  $x_e$  par un feedback continu. Cette condition est suffisante, mais loin d'être nécessaire, comme l'illustre l'exemple suivant :

$$\dot{x} = x^2 u, \quad (3.17)$$

Bien que sa linéarisation en  $x = 0$  est identiquement nulle, ce système peut être stabilisé par la commande continue  $u = -x$ .

D'autre part, si le système (3.16) a des valeurs propres instables non contrôlables, alors la stabilisabilité (en fait, même  $C^1$ ) n'est pas possible, pas même localement. Comme d'habitude, le cas critique est rencontré lorsque la linéarisation approximative à des valeurs propres non contrôlables à partie réelle nulle. Dans ce cas, rien ne peut être conclu sur la base de l'approximation linéaire, sauf que la stabilisation exponentielle ne peut pas être obtenue.

Toutefois, dans certains cas, on peut utiliser les conditions nécessaires d'existence d'un feedback stabilisant  $C^1$  afin de mieux comprendre le cas critique.

**Théorème 2** (Brockett). *Si le système*

$$\dot{x} = \varphi(x, u)$$

*admet un feedback  $C^1$  de forme  $u = u(x)$  qui rend  $x_e$  asymptotiquement stable, alors l'image de l'application*

$$\varphi : \mathbb{R}^n \times U \rightarrow \mathbb{R}^n$$

*contient un voisinage de  $x_e$ .*

**Exemple 8.** *Nous voulons étudier la stabilisabilité du point d'équilibre  $x = 0$  pour le système*

$$\dot{x}_1 = u_1 \quad (3.18)$$

$$\dot{x}_2 = u_2 \quad (3.19)$$

$$\dot{x}_3 = x_2 u_1 - x_1 u_2. \quad (3.20)$$

*En appliquant le théorème de Chow, il est facile de voir que le système est contrôlable. En ce qui concerne stabilisabilité lisse, rien ne peut être conclu en regardant la linéarisation dans un voisinage de  $x = 0$ , en raison de la présence d'une valeur propre à zéro incontrôlable.*

*Toutefois, en remarquant qu'aucun point de la forme*

$$\begin{bmatrix} 0 \\ 0 \\ \varepsilon \end{bmatrix} \quad \varepsilon \neq 0$$

*n'est dans l'image de  $\varphi$ , le théorème de Brockett permet de déduire que la stabilisation au point  $x = 0$  est impossible par un feedback de classe  $C^1$ .*

*Nous signalons les points importants ci-dessous :*

- *Lorsque le théorème de Brockett est appliqué aux systèmes de contrôle sans dérive (A.8) avec les champs de vecteur  $g_j$  linéairement indépendantes en  $x_e$  (comme dans l'exemple précédent), il implique  $m = n$  comme condition nécessaire et suffisante pour la stabilisabilité. Toutefois, si la dimension de la distribution  $\Delta = \text{span}\{g_1, \dots, g_m\}$  chute en  $x_e$ , une telle condition n'est plus nécessaire.*

- Si le système (A.8) ne peut pas être stabilisé par un feedback de classe  $C^1$ , le même résultat négatif tient pour son extension dynamique (A.9) - (A.10). En d'autres termes, l'obstruction topologique pour la stabilisabilité de classe  $C^1$ - exprimé par le théorème 2 est conservée sous l'extension dynamique
- Lorsque le théorème de Brockett ne s'applique pas alors le Feedback est variant  $u = u(x, t)$ .

Comme conclusion, un systèmes sous-actionné ( $m < n$ ) sans dérive qui satisfait l'hypothèse d'indépendance des  $g_j$  ne peut pas être stabilisé par des lois Feedback statiques continument dérivables.

# Chapitre 4

## Méthodes d'optimisation

### 4.1 Programmation quadratique séquentielle(SQP)[24][4]

Les méthodes de programmation quadratique séquentielle (SQP) ont reçu beaucoup d'attention ces dernières années, en raison de leur taux supérieur de convergence. Bien que les méthodes SQP peut être interprétées comme la méthode de Newton appliquée à la solution des conditions Karush-Kuhn-Tucker (KKT), cet aspect sera ne pas être approfondis. La méthode donnée dans la suite fut d'abord publiée par Pshenichny en 1970 en russe et plus tard dans un livre de Pshenichny et Danilin au 1978. Ils appellent leur méthode SQP comme «méthode de linéarisation» et la convergence prouvée sous certaines conditions. Beaucoup de variations des méthodes de SQP existent dans la littérature. En fait, Matlab "fmincon" optimiseur est basé sur un algorithme SQP.

Dans cette section, nous décrirons quelques idées de base de la programmation quadratique séquentielle (SQP) méthode pour résoudre les problèmes d'optimisation non linéaire avec contraintes. nous allons considérer problème de la forme générale :

$$(P) \begin{cases} \min f(x) \\ g(x) \leq 0 \\ h(x) = 0 \end{cases}$$

où au moins l'une des contraintes est non linéaire.

Une des façons de motiver SQP est d'employer la Méthode de Newton (ou de quasi-Newton) pour résoudre directement les conditions de KKT pour le problème (P). Une autre façon de voir l'idée de base de la Méthode SQP, L'idée de SQP est de modéliser ce problème à la  $x_k$  point actuel par un sous-problème quadratique et d'utiliser la solution de ce sous-problème pour trouver le nouveau point  $x_{k+1}$ . Ce processus est itéré pour créer une séquence d'approximations qui, on l'espère, vont converger vers une solution  $x^*$  de (P).

#### 4.1.1 La Méthode SQP de base

Pour comprendre l'utilisation de SQP dans les problèmes avec des contraintes générales, nous commençons par considérer le problème avec contraintes d'égalité

$$\min f(x) \tag{4.1}$$

$$\text{sous contraintes } C_j(x) = 0, \quad j = 1, 2, \dots, m. \tag{4.2}$$

Le lagrangien pour ce problème est définie par  $L(x, \lambda) = f(x) - \lambda^T C(x)$ , nous définissons le Jacobien correspond aux contraintes :

$$A(x)^T = [\nabla c_1(x), \nabla c_2(x), \dots, \nabla c_m(x)] \quad (4.3)$$

Les conditions de premier ordre de KKT pour un un problème avec contrainte d'égalité donnent :

$$F(x, \lambda) = \begin{bmatrix} \nabla f(x) - A(x)^T \lambda \\ c(x) \end{bmatrix} = 0 \quad (4.4)$$

n'importe qu'elle solution  $(x^*, \lambda^*)$  du problème (4.2) doit vérifier que  $A(x^*)$  soit de rang complet et vérifier l'équation (4.4). Une approche pour résoudre les équations non linéaires (4.4) est basé sur la méthode de Newton.

Le Jacobien de (4.4), respectant  $x$  et  $\lambda$  est donnée par :

$$F'(x, \lambda) = \begin{bmatrix} \nabla_{xx} L(x, \lambda) & -A(x)^T \\ A(x) & 0 \end{bmatrix} \quad (4.5)$$

les itérations de Newton à partir de  $(x_k, \lambda_k)$  sont données par :

$$\begin{bmatrix} x_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ \lambda_k \end{bmatrix} + \begin{bmatrix} p_k \\ p_\lambda \end{bmatrix} \quad (4.6)$$

où  $p_k$  et  $p_\lambda$  résolvent le système de Newton-KKT.

$$\begin{bmatrix} \nabla_{xx} L_k & -A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} p_k \\ p_\lambda \end{bmatrix} = \begin{bmatrix} -\nabla f(k) + A(k)^T \lambda_k \\ -c(k) \end{bmatrix} \quad (4.7)$$

Cette itération existe lorsque la matrice de KKT de l'équation (4.7) est non singulière. Cette dernière est garantie si les hypothèses suivantes sont vérifiées pour  $(x, \lambda) = (x_k, \lambda_k)$ .

**Hypothèses 1.** Soient les hypothèses suivantes :

- Le Jacobien correspondant aux contraintes définies par  $A(x)$  est de rang complet.
- La matrice  $\nabla_{xx} L(x, \lambda)$  est définie positive, c'est à dire  $d^T \nabla_{xx} L(x, \lambda) d > 0$  pour tout  $d \neq 0$  et  $A(x)d = 0$  pour  $d = 0$ .

Il ya un autre moyen pour visualiser l'itération (4.6), (4.7), passant par un développement de Taylor à l'itération  $(x_k, \lambda_k)$ , le problème (4.2) prend la forme quadratique suivante :

$$\begin{aligned} \min_p f_k + \nabla f_k^T p + \frac{1}{2} p^T \nabla_{xx}^2 L_k p \\ \text{sous contrainte } A_k p + c_k = 0 \end{aligned} \quad (4.8)$$

Si les hypothèses précédentes sont vérifiées , le problème a une solution unique  $(p_k, l_k)$  qui vérifie :

$$\nabla_{xx}^2 L_k p_k + \nabla f_k - A_k^T l_k = 0 \quad (4.9)$$

$$A_k p_k + c_k = 0 \quad (4.10)$$

Le vecteur  $p_k, l_k$  peuvent être identifié avec la solution des équations de Newton (4.7). Si on remplace le  $p_\lambda = \lambda_{k+1} - \lambda_k$  dans (4.7) on trouve :

$$\begin{bmatrix} \nabla_{xx} L_k & -A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} p_k \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} -\nabla f(k) \\ -c(k) \end{bmatrix} \quad (4.11)$$

Par la non singularité de la matrice, on a  $\lambda_{k+1} = l_k$ , et  $p_k$  résolvent les équations (4.8) et (4.7).

**Algorithme 1.** *Algorithme Local pour résoudre (4.2)*  
 Choisissons une pair initial  $(x_0, \lambda_0)$ ;  $k \leftarrow 0$ ;  
 Répéter jusqu'à la satisfaction du test de convergence  
 Evaluer  $f_k, \nabla f_k, \nabla_{xx}^2 L_k, c_k$  et  $A_k$ ;  
 Résoudre (4.8) pour obtenir  $p_k$  et  $l_k$ ;  
 Poser  $x_{k+1} \leftarrow x_k + p_k$  et  $\lambda_{k+1} \leftarrow l_k$ ;  
 end(Répéter)

Dans ce cas, l'équation (4.8) est une approximation quadratique de la fonction Lagrangienne. Ce fait constitue une motivation pour notre choix du modèle quadratique (4.8) : comme première étape il faut remplacer le programme non linéaire (4.2) par le problème de minimisation du Lagrangien soumis à des contraintes d'égalité (4.8), puis faire une approximation quadratique de Lagrange et une approximation linéaire des contraintes pour obtenir (4.8).

Ce cas peut être généralisé pour un problème de programmation non linéaire (P), pour modéliser ce problème il faut linéariser les contraintes égalité et inégalité les deux à la fois, on obtient :

$$\min_p f_k + \nabla f_k^T p + \frac{1}{2} p^T \nabla_{xx}^2 L_k p \quad (4.12a)$$

$$\text{sous contrainte } C_i(x_k)^T p + C_i x_k = 0 \quad (4.12b)$$

$$\nabla C_i(x_k)^T p + C_i x_k \leq 0 \quad (4.12c)$$

Nous utilisons maintenant l'un des algorithmes de programmation quadratique, parmi ces Algorithmes : l'ensemble active et points intérieurs pour plus de détails consulter [24]. La fonction utilisée dans Matlab pour résoudre ce genre de problème est *Fmincon* cette fonction par défaut utilise l'Algorithme SQP, cette méthode peut converger rapidement, mais elle peut diverger de la solution à cause de choix des conditions initiales.

## 4.2 Algorithmes Génétiques [10] [30]

Parmi les méthodes métaheuristiques on parle de l'algorithme génétique proposé dans les années 1975 par Holland, ces méthodes sont dites évolutionnaires qui manipulent un ensemble de plusieurs solutions simultanément. Une population d'individus évoluant en même temps comme l'évolution naturelle en biologie.

Pour chacun des individus on mesure sa faculté d'adaptation au milieu extérieur par le *Fitness*. Les algorithmes génétiques s'appuient alors sur trois fonctionnalités :

- la sélection : qui permet de favoriser les individus qui ont la meilleure *Fitness* pour notre cas c'est la valeur de la fonction objective de la solution associée à l'individu.
- le croisement : qui combine deux solutions parents pour former un ou deux enfants, Figure 4.1.

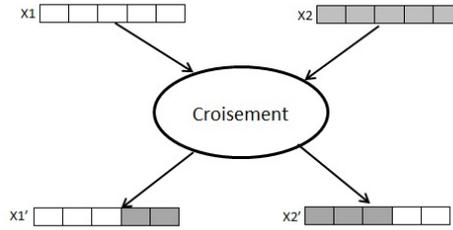


FIGURE 4.1 – Exemple sur le croisement de deux individus

- la mutation : qui permet d'ajouter de la diversité à la population en mutant certain caractéristiques d'une solution comme montre la Figure 4.2 pour deux individus codé en binaire.

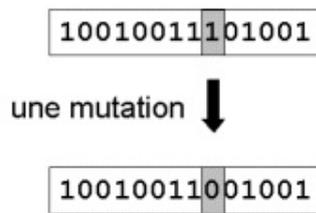


FIGURE 4.2 – Exemple sur la mutation

La représentation des solutions (codage) est un point critique de la réussite d'un algorithme génétique. il faut bien sur qu'il s'adapte le mieux possible au problème et à l'évaluation d'une solution. On peut faire un codage binaire ou bien réel par exemple soient deux individus :  $X_1, X_2$  qui sont des solutions initial du problème les opérations génétiques sur ces deux individus sont données par la Figure 4.3

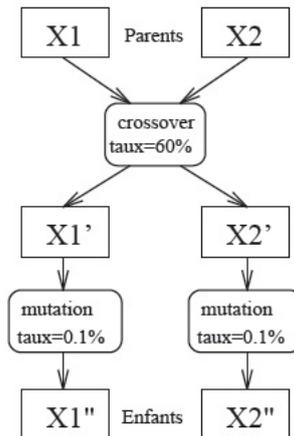


FIGURE 4.3 – Les opérations génétiques pour deux individus

### 4.2.1 Gestion des contraintes

Un élément de population qui viole une contrainte se verra attribuer une mauvaise fitness et aura une probabilité forte d'être éliminée par le processus de sélection. Il peut

cependant être intéressant de conserver, tout en les pénalisant, les éléments non admissibles car ils peuvent permettre de générer des éléments admissibles de bonne qualité. Pour de nombreux problèmes, l'optimum est atteint lorsque l'une au moins des contraintes de séparation est saturée, c'est à dire sur la frontière de l'espace admissible.

**Algorithme 2.** *un simple Algorithme Génétique*

1. *Initialisation : génération d'une population initiale  $P$  des solutions avec taille  $|P| = n$*
2. *Répéter*
3.  $P' \leftarrow \emptyset$
4. *Répéter*
5. *la sélection : choisit 2 solutions  $x$  et  $x'$  à partir de  $P$  avec une probabilité proportionnelle au Fitness*
6. *Croisement : combiner les solutions parents pour former les solutions enfants  $y$  et  $y'$  avec avec une forte probabilité*
7. *mutation  $y$  et  $y'$  avec probabilité faible*
8. *ajouté  $y$  et  $y'$  au  $P'$*
9. *jusqu'à  $|P'| = n$*
10.  $P \leftarrow P'$
11. *jusqu'à la satisfaction d'un critère d'Arrêt*

## 4.2.2 Avantages et inconvénients

D'abord, les algorithmes génétiques sont coûteux en temps de calcul, puisqu'ils manipulent plusieurs solutions simultanément. C'est le calcul de la fonction de performance qui est le plus pénalisant, et on optimise généralement l'algorithme de façon à éviter d'évaluer trop souvent cette fonction.

Le problème d'ajustement d'un algorithme génétique est délicat, parmi les problèmes les plus caractéristiques est celui de la dérive génétique, qui fait qu'un bon individu se met, les méthodes de sélection proportionnelle peuvent en particulier favoriser ce genre de dérive. Un autre problème surgit lorsque les différents individus se mettent à avoir des performances similaires : les bons éléments ne sont alors plus sélectionnés, et l'algorithme ne progresse plus.

Le grand avantage des algorithmes génétiques est qu'ils parviennent à trouver de bonnes solutions sur des problèmes très complexes, et trop éloignés des problèmes combinatoires classiques pour qu'on puisse tirer profit de certaines propriétés connues. Ils doivent simplement déterminer entre deux solutions quelle est la meilleure, afin d'opérer leurs sélections. On les emploie dans les domaines où un grand nombre de paramètres entrent en jeu, et où l'on a besoin d'obtenir de bonnes solutions en quelques itérations seulement.

## 4.3 Les essais particuliers (Particle Swarm Optimization : PSO) [9][22]

l'optimisation par essaim particulaire (OEP) est une méthode proposée en 1995 aux Etats Unis sous le nom de Particle Swarm Optimization (PSO) par Eberhart et Kennedy.

Initialement, ses deux concepteurs, Russel Eberhart et James Kennedy, cherchaient à modéliser des interactions sociales entre des « agents » devant atteindre un objectif donné dans un espace de recherche commun, chaque agent ayant une certaine capacité de

mémorisation et de traitement de l'information. La règle de base était qu'il ne devait y avoir aucun chef d'orchestre, ni même aucune connaissance par les agents de l'ensemble des informations, seulement des connaissances locales. Un modèle simple fut alors élaboré.

Dès les premières simulations, le comportement collectif de ces agents évoquait celui d'un essaim d'êtres vivants convergeant parfois en plusieurs sous-essaims vers des sites intéressants. Ce comportement se retrouve dans bien d'autres modèles, explicitement inspirés des systèmes naturels. Ici, la métaphore la plus pertinente est probablement celle de l'essaim d'abeilles, particulièrement du fait qu'une abeille ayant trouvé un site promoteur sait en informer certaines de ses consœurs et que celles-ci vont tenir compte de cette information pour leur prochain déplacement. Finalement, le modèle s'est révélé être trop simple pour vraiment simuler un comportement social, mais par contre très efficace en tant qu'outil d'optimisation.

Comme nous allons le voir, le fonctionnement de l'OEP fait qu'elle peut être rangée dans les méthodes itératives (on approche peu à peu de la solution) et stochastiques (on fait appel au hasard). Sous ce terme un peu technique, on retrouve un comportement qui est aussi vieux que la vie elle-même : améliorer sa situation en se déplaçant partiellement au hasard et partiellement selon des règles prédéfinies.

### 4.3.1 Description informelle

La version historique peut facilement être décrite en se plaçant du point de vue d'une particule. Au départ de l'algorithme, un essaim est réparti au hasard dans l'espace de recherche, chaque particule ayant également une vitesse aléatoire. Ensuite, à chaque pas de temps :

- chaque particule est capable d'évaluer la qualité de sa position et de garder en mémoire sa meilleure performance, c'est-à-dire la meilleure position qu'elle a atteinte jusqu'ici (qui peut en fait être parfois la position courante) et sa qualité (la valeur en cette position de la fonction à optimiser).
- chaque particule est capable d'interroger un certain nombre de ses congénères (ses informatrices, dont elle-même) et d'obtenir de chacune d'entre elles sa propre meilleure performance (et la qualité afférente).
- à chaque pas de temps, chaque particule choisit la meilleure des meilleures performances dont elle a connaissance, modifie sa vitesse en fonction de cette information et de ses propres données et se déplace en conséquence.

Le premier point se comprend facilement, mais les deux autres nécessitent quelques précisions. Les informatrices sont définies une fois pour toutes de la manière suivante (Figure 4.4).

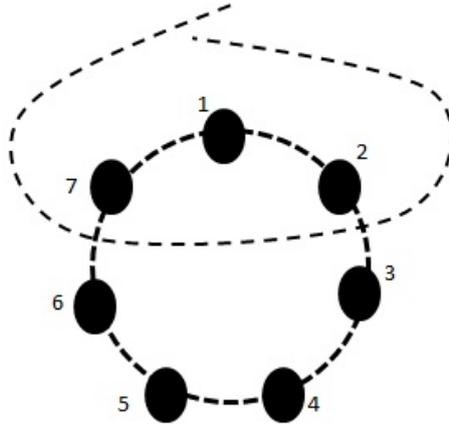


FIGURE 4.4 – Le cercle virtuel pour un essaim de sept particules.

On suppose toutes les particules disposées (symboliquement) en cercle et, pour la particule étudiée, on inclut progressivement dans ses informatrices, d’abord elle-même, puis les plus proches à sa droite et à sa gauche, de façon à atteindre le total requis. Il y a bien sûr de nombreuses variantes, y compris celle consistant à choisir les informatrices au hasard, mais celle-ci est à la fois simple et efficace.

Une fois la meilleure informatrice détectée, la modification de la vitesse est une simple combinaison linéaire de trois tendances, à l’aide de coefficients de confiance :

- la tendance « aventureuse », consistant à continuer selon la vitesse actuelle ;
- la tendance « conservatrice », ramenant plus ou moins vers la meilleure position déjà trouvée ;
- la tendance « panurgienne », orientant approximativement vers la meilleure informatrice.

Les termes « plus ou moins » ou « approximativement » font référence au fait que le hasard joue un rôle, grâce à une modification aléatoire limitée des coefficients de confiance, ce qui favorise l’exploration de l’espace de recherche. La figure 4.5 présente un schéma de principe résumant les explications ci-dessus. Naturellement, pour pouvoir être programmé, tout ceci est formalisé dans des équations de mouvement. Un point intéressant est que, contrairement à bien d’autres heuristiques qui restent purement expérimentales, il existe une analyse mathématique précisant les conditions de convergence et le choix des paramètres.

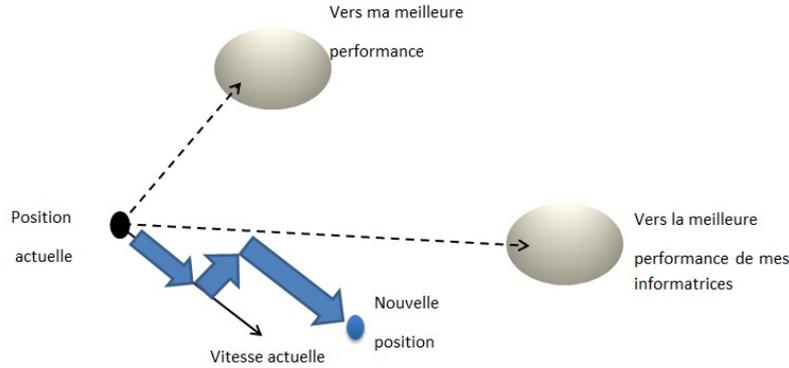


FIGURE 4.5 – Schéma de principe du déplacement d'une particule

### 4.3.2 Formalisation

La version de base de l'OEP peut facilement être formalisée et programmée. L'espace de recherche est de dimension  $D$ . La position courante d'une particule dans cet espace à l'instant  $t$  est donnée par un vecteur  $x(t)$ , à  $D$  composantes, donc sa vitesse courante est  $v(t)$ . La meilleure position trouvée jusqu'ici par cette particule est donnée par un vecteur  $p(t)$ . Enfin, la meilleure de celles trouvées par les informatrices de la particule est indiquée par un vecteur  $g(t)$ . En général, nous écrirons simplement  $x, v, p$  et  $g$ . La  $d^{ieme}$  composante de l'un quelconque de ces vecteurs est indiquée par l'indice  $d$ , par exemple  $x_d$ . Avec ces notations, les équations de mouvement d'une particule sont, pour chaque dimension  $d$  :

$$\begin{cases} v_d \leftarrow c_1 v_d + alea(0, c_{max})(p_d - x_d) + alea(0, c_{max})(g_d - x_d) \\ x_d \leftarrow x_d + v_d \end{cases} \quad (4.13)$$

Pour une bonne convergence, les valeurs de  $c_1$  et  $c_{max}$  ne doivent pas être choisies indépendamment. En pratique le premier doit être un peu inférieur à 1 et le second peut être calculé par la formule

$$c_{max} = (2/0,97725)c_1$$

Plus  $c_1$  est proche de 1 meilleure est l'exploration de l'espace de recherche, au détriment, néanmoins, de la vitesse de convergence.

Plus généralement, on souhaite souvent rester dans un espace de recherche fini donné. Par conséquent, on ajoute un mécanisme pour éviter qu'une particule ne sorte de cet espace. Le plus fréquent est le confinement d'intervalle. Supposons, par simplicité, que l'espace de recherche soit  $[x_{min}, x_{max}]^D$ . Alors ce mécanisme stipule que si une coordonnée  $x_d$ , calculée selon les équations de mouvement, sort de l'intervalle  $[x_{min}, x_{max}]$ , on lui attribue en fait la valeur du point frontière le plus proche. En pratique, cela revient donc à remplacer la deuxième ligne des équations de mouvement par

$$x_d \leftarrow \min(\max(x_d + v_d, x_{min}), x_{max})$$

Plus généralement, le principe même du confinement consiste à dire ceci : « si une particule tend à sortir de l'espace de recherche, alors la ramener au point le plus proche qui soit dans cet espace et modifier sa vitesse en conséquence ». Il permet de définir les confinements nécessaires à des problèmes à granularité non nulle (positions à valeurs entières, par exemple) ou à des problèmes (combinatoires, en général) dont les solutions doivent avoir toutes les coordonnées différentes.

# Chapitre 5

## Applications

Dans cette partie on traite le problème de génération de trajectoire optimale pour un robot manipulateur sous actionné à deux axes sur le plan horizontal de type  $R_aR_p$  et la planification de trajectoire pour un robot  $PPR_p$  sans gravité.

### 5.1 Cas d'un Robot manipulateur à deux axes planaire type $R_aR_p$

Le modèle dynamique d'un robot manipulateur à deux axes est donné par :

$$\begin{bmatrix} B_{aa}(q) & B_{ua}^T(q) \\ B_{ua}^T(q) & B_{uu}(q) \end{bmatrix} \begin{bmatrix} \ddot{q}_a \\ \ddot{q}_p \end{bmatrix} + \begin{bmatrix} C_a(q, \dot{q}) \\ C_u(q, \dot{q}) \end{bmatrix} + \begin{bmatrix} e_a(q) \\ e_u(q) \end{bmatrix} = \begin{bmatrix} \tau_a \\ 0 \end{bmatrix} \quad (5.1)$$

La deuxième équation différentielle présente une contrainte non holonome de second ordre, le premier travail est de chercher s'il existe une intégrale de cette contrainte pour pouvoir minimiser le nombre de coordonnées généralisées.

#### 5.1.1 Condition d'intégrabilité

La contrainte non holonome de second ordre s'écrit sous la forme :

$$H_u(q)\ddot{q} + c_u(q, \dot{q}) + e_u(q) = 0 \equiv F(\ddot{q}, \dot{q}, q, t) = 0, \quad (5.2)$$

Il existe deux notions d'intégrabilité, la première est dite intégrabilité partielle et la deuxième intégrabilité complète.

##### intégrabilité partielle

L'intégrabilité partielle est équivalente à l'existence d'une fonction différentiable  $g(\dot{q}, q, t) = 0$  du premier ordre de telle sorte que

$$\frac{\partial g}{\partial q} = F(\ddot{q}, \dot{q}, q, t) \quad (5.3)$$

maintenant supposons que la fonction  $g$  existe d'où

$$\frac{\partial g}{\partial \dot{q}} \ddot{q} + \frac{\partial g}{\partial q} \dot{q} + \frac{\partial g}{\partial t} = 0, \quad (5.4)$$

- Les conditions d'intégration sont données par comparaison entre les équations (5.2) et (5.4) :
- le terme  $\frac{\partial g}{\partial t}$  est constant, car le temps n'apparaît pas explicitement dans (5.2) donc  $g$  s'écrit sous la forme

$$g(\dot{q}, q, t) = g_1(q, \dot{q}) + k_1 t = 0, \quad k_1 \in \mathbb{R}^{n-m} \quad (5.5)$$

La différentielle de (5.5) est donnée par

$$\frac{\partial g_1}{\partial q} \dot{q} + \frac{\partial g_1}{\partial \dot{q}} \ddot{q} + k_1 = 0 \quad (5.6)$$

Nous allons donner dans ce qui suit des conditions nécessaires d'intégrabilité par considération de cas particuliers soit  $\dot{q} = 0$  dans l'équation (5.2) et (5.6)

$$(5.2) \Rightarrow H_u \ddot{q} + e_u = 0, \quad (5.6) \Rightarrow \frac{\partial g_1}{\partial \dot{q}} \ddot{q} + k_1 = 0$$

A partir de ces deux équations on a :

$$H_u = \frac{\partial g_1}{\partial \dot{q}} \quad \text{et} \quad e_u = k_1. \quad (5.7)$$

- Considérons  $H_u$  en fonction de  $q$  seulement, la vitesse de l'axe  $\dot{q}$  apparaît linéairement dans l'équation (5.5), qui aura la forme suivante

$$g(q, \dot{q}, t) = g_2(q) + H_u(q) \dot{q} + e_u t = 0. \quad (5.8)$$

La différentielle de cette dernière est :

$$\frac{\partial g_2}{\partial q} \dot{q} + \dot{H}_u \dot{q} + H_u \ddot{q} + e_u = 0. \quad (5.9)$$

l'expression explicite de  $c_u$  dans l'équation (5.2) est donnée par

$$c_u = \dot{H}_u \dot{q} - \frac{1}{2} \left( \frac{\partial}{\partial q_u} (\dot{q}^T H \dot{q}) \right)^T, \quad (5.10)$$

remplaçons l'équation (5.10) dans (5.2) et comparons avec (5.9) on trouve

$$\frac{\partial g_2}{\partial q} \dot{q} + \frac{1}{2} \left( \frac{\partial}{\partial q_u} (\dot{q}^T H \dot{q}) \right)^T = 0. \quad (5.11)$$

On peut montrer que (5.11) est équivalente à

$$\frac{1}{2} \left( \frac{\partial}{\partial q_u} (\dot{q}^T H \dot{q}) \right)^T = 0 \quad \text{et} \quad \frac{\partial g_2}{\partial q} = 0, \quad (5.12)$$

c'est à dire l'énergie cinétique ne dépend pas des variables sous actionnées. On a  $H$  symétrique ce qui implique

$$\frac{\partial H}{\partial q_{u,i}} = 0, \quad i = m + 1, \dots, n \quad (5.13)$$

avec  $O$  la matrice nulle de  $n \times n$  éléments. par l'équation (5.12),  $g_2$  est constante égale à  $k_2$ , donc si l'équation (5.2) est partiellement intégrable alors son intégrale s'écrit sous la forme

$$g(q, \dot{q}, t) = H_u(q)\dot{q} + e_u t + k_2 = 0, \quad (5.14)$$

où  $k_2$  une constante qui peut être déterminée à partir des conditions initiales du mouvement.

Les équations (5.7) et (5.13) sont des conditions nécessaires pour l'intégration partielle de (5.2). La proposition suivante montre qu'elles sont aussi une condition suffisante.

**Proposition 4.** *La contrainte (5.2) est partiellement intégrable si et seulement si :*

1. *Le terme de gravitation est constant ;*
2. *Les variables sous actionnées  $q_u$  n'apparaissent pas dans la matrice d'inertie du manipulateur.*

*Si ces deux conditions sont vérifiées, l'intégrale partielle est donnée par (5.14).*

### Intégrabilité complète

Si les conditions d'intégrabilité partielle sont vérifiées, la contrainte (5.2) est intégrable sous la forme (5.14). pour décider si cette dernière est intégrable considérons le cas où

$$H_u(q)\dot{q} = 0 \quad (5.15)$$

Pour n'importe quel  $q$ , (5.15) détermine un sous espace linéaire  $\Delta(q)$  de dimension  $m$ , qui est le noyau de la matrice  $H_u$ . L'application  $\Delta : q \rightarrow \Delta(q)$  est la  $m$  distribution associée à la contrainte (5.15).

Le théorème de Frobenius (A.1) dit que (5.15) est intégrable si et seulement si  $\Delta$  est involutive. si cette condition est vérifiée, l'équation (5.14) est intégrable comme suit :

$$f(q, t) = f_1(q) + \frac{e_u}{2} t^2 + k_2 t + k_3 = 0, \quad (5.16)$$

avec  $\frac{\partial f_1}{\partial q} = H_u$ , et  $k_2, k_3$  sont des vecteurs constants qui dépendent des conditions initiales.

**Proposition 5.** *La contrainte (5.2) est complètement intégrable (holonome) si et seulement si*

1. *Elle est partiellement intégrable ;*
  2. *La distribution  $\Delta$  défini par (5.15) est involutive.*
- dans ce cas l'intégrale de (5.2) est donnée par (5.16).*

### Robot manipulateur à 2 axes sans gravité

On traite ici la génération de trajectoire pour ce robot en utilisant les résultats des chapitres 1 et 4, pour que le programme d'optimisation converge vers une solution il faut relaxer la contrainte d'égalité non holonome sous la forme :

$$|\tau_p(t)| \leq \bar{\epsilon},$$

$\bar{\epsilon}$  la plus petite valeur qui donne une bonne solution, après il faut chercher la commande qui annule cette erreur ( $\bar{\epsilon}$ ).

## Le modèle dynamique d'un manipulateur sous actionné à deux axes type $R_a R_p$

Le modèle dynamique utilisé dans la simulation est donnée par [7] sous la forme

$$\begin{aligned} \tau_1(t) = & (3.78 + 0.272 \cos(q_2) + 0.022 \sin(q_2))\ddot{q}_1 \\ & + (0.08 + 0.136 \cos(q_2) + 0.011 \sin(q_2))\ddot{q}_2 \\ & + (0.011 \cos(q_2) - 0.136 \sin(q_2))(2\dot{q}_1\dot{q}_2 + \dot{q}_2^2) \end{aligned} \quad (5.17)$$

$$\begin{aligned} 0 = & (0.08 + 0.136 \cos(q_2) + 0.011 \sin(q_2))\ddot{q}_1 \\ & + 0.08\ddot{q}_2 - (0.011 \cos(q_2) - 0.136 \sin(q_2))\dot{q}_1^2 \end{aligned} \quad (5.18)$$

on voit que la deuxième équation présente une contrainte non holonome de second ordre puisque la matrice d'inertie ne dépend que de l'axe sous actionné  $q_2$ , d'après la Proposition 4 la contrainte n'est pas partiellement intégrable donc elle est complètement non intégrable.

### Les résultats d'optimisation pour $\epsilon = 1e - 3$

On utilise les points de départ  $q^i = (0, 0)$  et les points d'arrivés désiré  $q^f = (\pi/2, \pi/3)$ , utilisant la trajectoire point à point avec trois points de contrôle intermédiaires décrit dans le (chapitre 1, section 1.2), on obtient les figures suivantes :

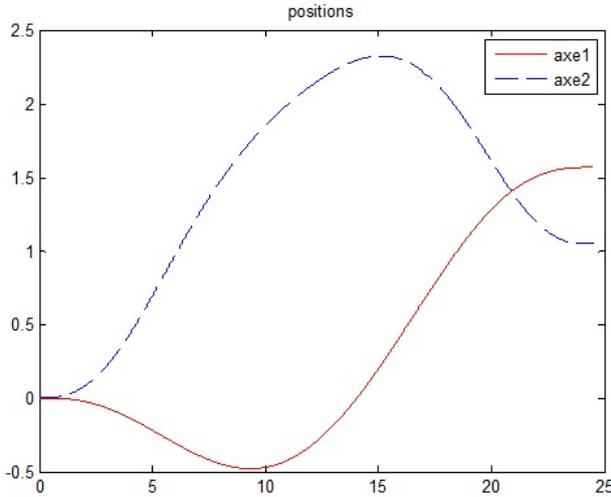


FIGURE 5.1 – Positions des axes en radian( $rd$ )

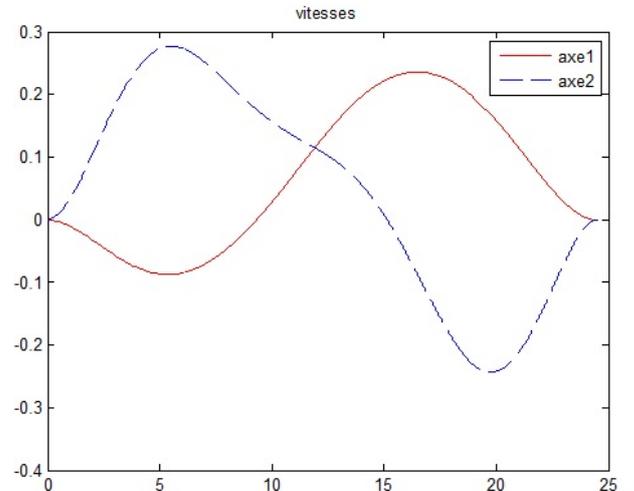


FIGURE 5.2 – Vitesses des axes( $rd/s$ )

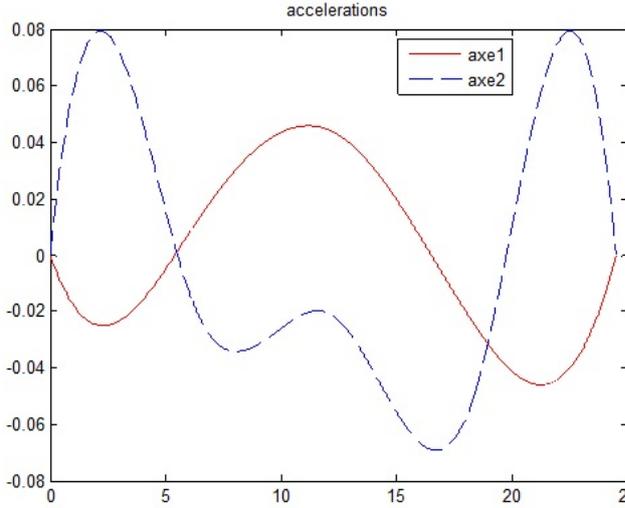


FIGURE 5.3 – Accélération des axes  
( $rd/s^2$ )

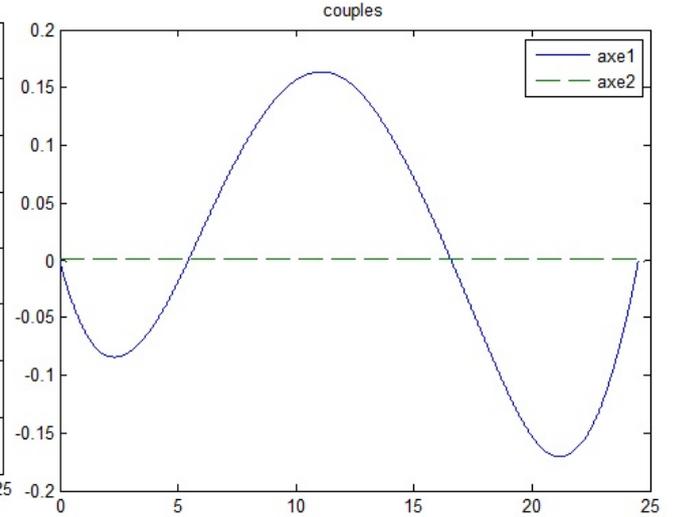


FIGURE 5.4 – Couples des axes( $Nm$ )

On observe que le couple du deuxième axe n'est pas nul tout le temps mais il oscille dans une bande  $\max(\tau_2) = 1e-3$ ,  $\min(\tau_2) = -1e-3$ , dans ce cas les contraintes sont bien vérifiées, mais dans la réalité il n'existe pas de moteur au niveau de deuxième axe (libre), la question qui se pose existe-t-il une commande continue qui permet d'annuler l'erreur dans le couple  $\tau_2$ . Pour répondre à cette question on utilise la linéarisation partielle du modèle (5.17),(5.18) qui s'écrit sous la forme simple :

$$a_{11}(q_2)\ddot{q}_1 + a_{12}(q_2)\ddot{q}_2 + h_1(q, \dot{q}) = \tau_1 \quad (5.19)$$

$$a_{21}(q_2)\ddot{q}_1 + a_{22}(q_2)\ddot{q}_2 + h_2(q, \dot{q}) = 0 \quad (5.20)$$

on a  $a_{22}(q_2) > 0, \forall q_2$ , de la deuxième ligne on tire  $\ddot{q}_2$  et on remplace dans la première équation

$$\left(a_{11} - \frac{a_{12}a_{21}}{a_{22}}\right)u + \left(h_1 - \frac{a_{12}}{a_{22}}h_2\right) = \tau_1$$

$$\ddot{q}_1 = u$$

en utilisant une commande  $u$  en boucle fermé de type PD de la forme

$$u = \ddot{q}_{1d} + k_p(q_{1d} - q_1) + k_v(\dot{q}_{1d} - \dot{q}_1)$$

Dans ce cas on a utilisé les trajectoires optimales précédentes comme trajectoires de références pour le premier axe avec  $\epsilon = 0$ , les résultats trouvés sont donnés par les figures suivantes :

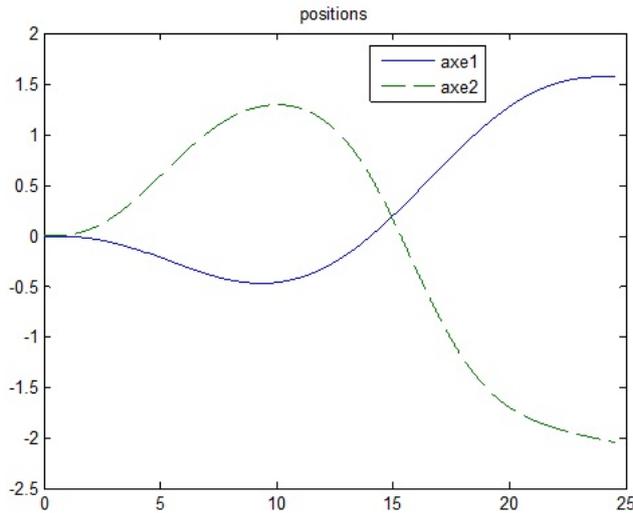


FIGURE 5.5 – Positions des axes en radian( $rd$ )

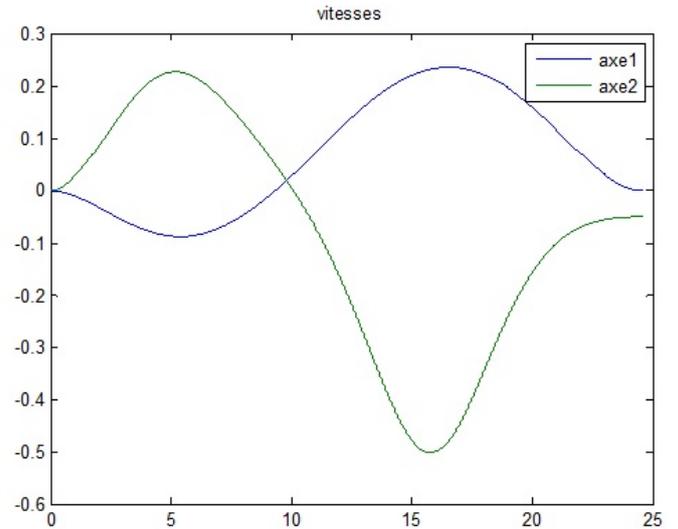


FIGURE 5.6 – Vitesses des axes( $rd/s$ )

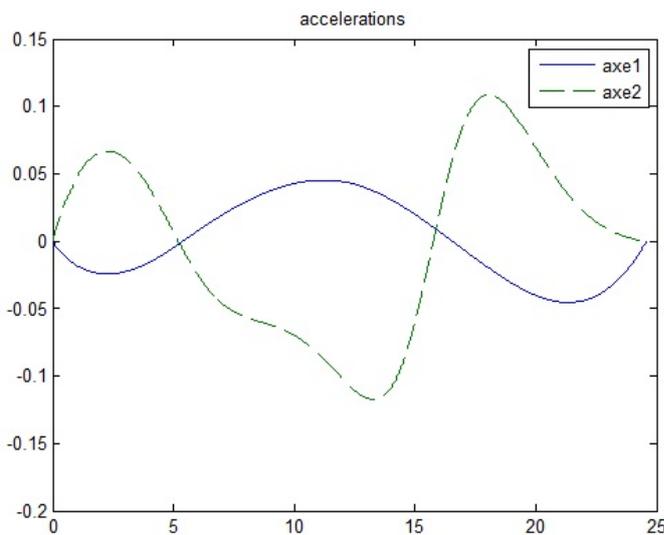


FIGURE 5.7 – Accélération des axes ( $rd/s^2$ )

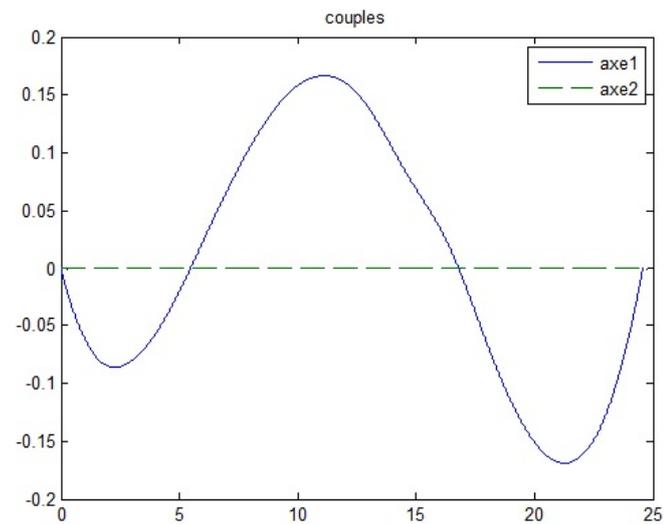


FIGURE 5.8 – Couples des axes( $Nm$ )

D'après ces figures le deuxième axe n'atteint pas sa position finale désirée lorsque le premier axe converge vers la trajectoire désirée, les effets du couplage non linéaire ne permettent pas la convergence du deuxième axe vers sa trajectoire désirée. La question qui se pose : Est ce qu'il existe une commande continue qui résoud le problème ? pour répondre à cette question on se base sur le Théorème 2 (chapitre 3) c'est à dire l'étude de la commandabilité locale autour du point d'équilibre, le Modèle (5.19),(5.20) peut s'écrire sous forme de variables d'états :

$$\dot{x} = \begin{bmatrix} x_2 \\ 0 \\ x_4 \\ -\frac{h_2(x,\dot{x})}{a_{22}} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ -\frac{a_{21}}{a_{22}} \end{bmatrix} u = f(x, u) \quad (5.21)$$

La linéarisation autour du point d'équilibre

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \frac{\partial f_1}{\partial x_4} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \frac{\partial f_2}{\partial x_4} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} & \frac{\partial f_3}{\partial x_4} \\ \frac{\partial f_4}{\partial x_1} & \frac{\partial f_4}{\partial x_2} & \frac{\partial f_4}{\partial x_3} & \frac{\partial f_4}{\partial x_4} \end{bmatrix}_{(x,u)_{eq}} \quad B = \begin{bmatrix} \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial u} \\ \frac{\partial f_3}{\partial u} \\ \frac{\partial f_4}{\partial u} \end{bmatrix}_{(x,u)_{eq}} \quad (5.22)$$

$$(x, u)_{eq} = (q_1, \dot{q}_1, q_2, \dot{q}_2, u)_{eq} = (0, 0, 0, 0, 0)$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -\frac{1}{0.08}(0.08 + 0.136) \end{bmatrix} \quad (5.23)$$

Etude de la commandabilité du système linéarisé (5.26)

$$C = [B \quad AB \quad A^2B \quad A^3B]$$

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -\frac{1}{0.08}(0.08 + 0.136) & 0 & 0 \\ -\frac{1}{0.08}(0.08 + 0.136) & 0 & 0 & 0 \end{bmatrix}, \det(C) = 0$$

D'après le théorème de Brockett le système (5.21) ne peut pas être stabilisé par une commande continue. Le système (5.21) peut être stabilisé par une commande discontinue, mais on n'a pas de planification ni de suivi de trajectoire [12].

### 5.1.2 Conclusion

Le système linéarisé autour du point d'équilibre n'est pas commandable ce qui implique qu'il n'est pas localement commandable en temps petit au voisinage du point d'équilibre, alors le système non linéaire n'est pas localement commandable autour de ce point, pour l'instant nous n'avons pas de critère pour conclure sur la commandabilité du système non linéaire (5.21).

## 5.2 Cas d'un Robot manipulateur de type $(n - 1)X_a R_p$

Dans cette section on traite de la planification de trajectoire pour un Robot de type  $(n - 1)X_a R_p$ , telle que  $(n - 1)X_a$  présentent des liaisons actives qui peuvent être prismatiques ou rotoïdes, la dernière liaison est de type rotoïde de plus cette dernière est passive (libre). Soit un état initial  $(q^0, \dot{q}^0)$  et un état final désiré  $(q^d, \dot{q}^d)$ , il faut trouver la trajectoire admissible  $q(t)$  (c-à-d vérifiant l'équation 5.20 pour un certain  $\tau_a(t)$  avec  $t \in [0, T]$ ,  $T > 0$ ) qui relie l'état initial à l'état final. Si de plus on a  $\dot{q}^0 = \dot{q}^d = 0$ , le problème s'appelle un problème de planification de trajectoire de repos au repos [12]. Pour résoudre ce problème on définit dans la section suivante la notion de la linéarisation dynamique par feedback.

### 5.2.1 La linéarisation dynamique par feedback

La linéarisation exacte par la technique de feedback dynamique représente une solution effective pour le problème précédent, il faut trouver un ensemble de sorties linéarisantes :

$$Z = h(q), \quad Z \in \mathbb{R}^m$$

ayant pour propriété que tout état et les entrées de système peuvent être écrits en terme de  $Z$  et ses dérivées, alors il est possible de construire un compensateur dynamique de la forme

$$\dot{\xi} = \alpha(\xi, q, \dot{q}) + \beta(\xi, q, \dot{q})v \quad (5.24)$$

$$a = \gamma(\xi, q, \dot{q}) + \delta(\xi, q, \dot{q})v \quad (5.25)$$

avec l'état  $\xi \in \mathbb{R}^r$  et la nouvelle entrée  $v \in \mathbb{R}^m$  telle que le système en boucle fermé soit linéaire et découplé entrée-état-sortie, c'est à dire, il est équivalent à une chaîne de  $m$  intégrateurs entre  $v$  et  $z$ .

Considérons le système avec  $m = 2$  pour notre système (PPR), l'algorithme de linéarisation procède comme suit : Les deux sorties linéarisantes  $Z_1$  et  $Z_2$  sont dérivées jusqu'à l'apparition d'une commande au niveau de  $Z_1$  ou  $Z_2$ , si la matrice actuelle (matrice de découplage) multipliée par les entrées est non singulière, alors un feedback statique peut être utilisé pour linéariser le comportement entrée-sortie. Cependant si la somme des degrés de dérivées supérieures est strictement inférieur à la dimension de l'espace d'état, il est impossible d'avoir la linéarisation complète, si la matrice de découplage est singulière (dans ce cas le  $Rang = 1$ ), on peut faire un changement d'état dépendant des coordonnées d'entrée soit une nouvelle entrée qui apparait. Sur cette entrée, l'extension dynamique est réalisée par l'ajout d'un seul intégrateur (qui est comme la première composante  $\xi_1$  d'état du compensateur dynamique) piloté par une nouvelle entrée scalaire. De plus, les dérivées des deux sorties dans cet ordre ne dépendent pas du tout de cette nouvelle entrée et on peut poursuivre la différentiation. Le processus est itéré plusieurs fois jusqu'à ce qu'on arrive à une matrice de découplage non singulière (au moins localement) pour le système prolongé, les états  $\xi$  du compensateur dynamique se construisent de manière itérative. En fin, la somme des ordres de dérivés de sorties est égale à la dimension du système prolongé (robot + compensateur dynamique) et la linéarisation entrée-état-sortie peut être obtenue par inversion.

#### Le modèle dynamique du Robot $(n - 1)X_a R_u$

Soit  $q = (q_1, \dots, q_n)$  n'importe quelle ensemble de coordonnées généralisées telle que  $q_n = \theta$  l'orientation de dernière liaison par rapport à l'axe des  $x$  comme l'illustre la figure 5.9

Le modèle dynamique est donné par

$$B(q)\ddot{q} + c(q, \dot{q}) + g(q) = \begin{bmatrix} \tau \\ 0 \end{bmatrix} \quad (5.26)$$

où  $\tau = (\tau_1, \dots, \tau_{n-1})$  le vecteur des couples correspond aux liaisons actives.  $B(q)$  la matrice symétrique définie positive de l'inertie,  $c(q, \dot{q})$  c'est le vecteur de coriolis et centrifuge et  $g(q)$  le vecteur des termes de gravité. Pour simplifier l'analyse du modèle dynamique on utilise les coordonnées généralisées définies par le vecteur  $q = (q_1, \dots, q_{n-1}, x, y, \theta) =$



dans la méthode du couple calculé, l'idée est de réduire la dynamique des articulations actives à  $n - 1$  chaînes de doubles intégrateurs, de sorte qu'ils peuvent être contrôlés via les entrées d'accélération. A cet effet, on calcule la dynamique passive  $\ddot{\theta}$  à partir de la dernière ligne de l'équation (5.27), et on remplace cette dernière dans la dynamique active pour calculer l'expression de  $\ddot{q}_a$ . D'où la linéarisation partielle est de la forme :

$$F_a = \widehat{B}_a(q)a + c_a(q, \dot{q}) + \widehat{g}_a(q) \quad (5.30)$$

avec la matrice  $\widehat{B}_a(q)$  de  $(n - 1) \times (n - 1)$  est définie par

$$\widehat{B}_a(q) = B_a(q) - \frac{m_n^2 d_n^2}{I_n + m_n d_n^2} \left[ \begin{array}{c|cc} 0_{(n-3) \times (n-3)} & & 0_{(n-3) \times 2} \\ \hline & s^2 \theta & -s\theta c\theta \\ 0_{2 \times (n-3)} & -s\theta c\theta & c\theta^2 \end{array} \right] \quad (5.31)$$

est toujours inversible étant le complément de Schur de  $b_{nn}$  l'élément diagonal de la matrice définie positive inertie  $B$ , où

$$\widehat{g}_a(q) = g_a(q_a) - \frac{g_0 m_n d_n c\theta}{I_n + m_n d_n^2} \begin{bmatrix} 0_{n-3} \\ -m_n d_n s\theta \\ m_n d_n c\theta \end{bmatrix} \quad (5.32)$$

Mettons les deux (5.27) et (5.30), le système de boucle fermée complète devient

$$\begin{aligned} \ddot{q}_1 &= a_1 \\ &\vdots \\ \ddot{q}_{n-3} &= a_{n-3} \\ \ddot{x} &= a_x \\ \ddot{y} &= a_y \\ \ddot{\theta} &= \frac{1}{K}(s\theta a_x - c\theta(a_y + g_0)), \end{aligned} \quad (5.33)$$

où  $K = (I_n + m_n d_n^2)/m_n d_n$  est précisément la distance du centre de percussion (CP) à partir de la base du dernier axe. Si la distribution de masse est supposée uniforme pour le dernier axe, on obtient  $K = 2l_n/3$  ( $l_n$  la longueur du  $n^{ieme}$  axe).

Les équations (5.33) montrent que les coordonnées dynamiques  $q_i, i = 1, \dots, n - 3$ , parmi les degrés de liberté actionnés est maintenant complètement découplée de la dynamique des coordonnées restantes  $(x, y, \theta)$ . En particulier, si une tâche de reconfiguration doit être exécutée, chaque  $q_i$  peut être entraîné indépendamment de sa valeur de consigne en choisissant  $a_i$ , pour  $i = 1, \dots, n - 3$ , comme une commande en boucle ouverte ou d'une loi à feedback linéaire. Par conséquent, à partir de maintenant nous allons mettre  $n = 3$  et considérer que le fond du problème, à savoir la planification de trajectoire et de contrôle pour les variables  $x, y$  et  $\theta$ . En conséquence, après la linéarisation partielle par feedback nous continuons uniquement avec les équations

$$\begin{aligned} \ddot{x} &= a_x \\ \ddot{y} &= a_y \\ \ddot{\theta} &= \frac{1}{K}(s\theta a_x - c\theta(a_y + g_0)), \end{aligned} \quad (5.34)$$

On remarque que, le mouvement du robot se produit sur un plan horizontal ( $\psi = 90^\circ$ ), donc  $g_0 = 0$ .

## Formalisation de la linéarisation dynamique par Feedback

Nous montrons que le modèle dynamique du robot de la forme (5.34) peut être transformé en un système linéaire commandable, utilisant un feedback dynamique non linéaire et un changement de coordonnées.

Pour cela, nous utilisons l'algorithme de linéarisation décrit dans la section (5.2.1), commençons par le choix de la position du centre de percussion comme sortie. Dans l'étape intermédiaire de l'algorithme nous sommes devant une matrice de découplage singulière qui exige d'ajouter un intégrateur pour une entrée. L'adaptation unique de l'algorithme générale consiste à ajouter deux intégrateurs en même temps dans cette étape d'extension dynamique, à cause de la nature du système mécanique du second ordre.

Définissons la position du CP de l'axe dernier comme sortie par

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + K \begin{bmatrix} c\theta \\ s\theta \end{bmatrix} \quad (5.35)$$

La différentiation de l'équation (5.35), donne

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} + K\dot{\theta} \begin{bmatrix} -s\theta \\ c\theta \end{bmatrix} \quad (5.36)$$

et

$$\begin{bmatrix} \ddot{y}_1 \\ \ddot{y}_2 \end{bmatrix} = \begin{bmatrix} c^2\theta & s\theta c\theta \\ s\theta c\theta & s^2\theta \end{bmatrix} \begin{bmatrix} a_x \\ a_y \end{bmatrix} - R(\theta) \begin{bmatrix} K\dot{\theta}^2 \\ g_0 c\theta \end{bmatrix}, \quad (5.37)$$

Dans cette dernière équation on a utilisé l'équation (5.33),  $R(\theta)$  correspondant à la matrice de rotation planaire d'un angle  $\theta$ . On constate que la matrice qui est multipliée par le vecteur  $(a_x, a_y)$  est singulière, on définit alors la transformation par feedback inversible

$$\begin{bmatrix} a_x \\ a_y \end{bmatrix} = R(\theta) \begin{bmatrix} \xi + K\dot{\theta}^2 \\ \sigma_2 \end{bmatrix} \quad (5.38)$$

où  $\xi$  et  $\sigma_2$  sont deux entrées auxiliaires variables. Notons que  $\sigma_2$  représente l'accélération linéaire de la base du dernier axe le long de la normale à son axe comme le montre la Figure 5.10, de plus le résultat de l'éq.(5.38) donne

$$\begin{bmatrix} \ddot{y}_1 \\ \ddot{y}_2 \end{bmatrix} R(\theta) \begin{bmatrix} \xi \\ -g_0 c\theta \end{bmatrix}, \quad (5.39)$$

et par conséquent  $\xi$  est l'accélération linéaire le long du dernier axe, comme illustre la Figure (5.11).

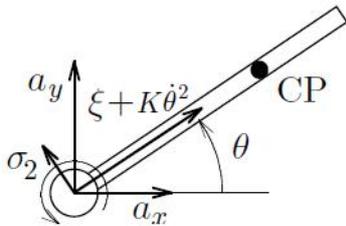


FIGURE 5.10 – Interprétation physique de l'équation (5.38)

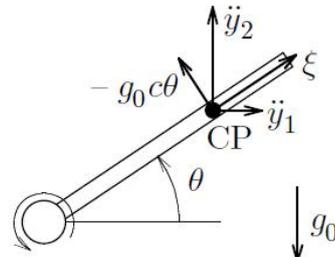


FIGURE 5.11 – Interprétation physique de l'équation (5.39)

Afin de mieux différencier la sortie, tout en évitant la différenciation de l'entrée  $\xi$ , nous ajoutons deux intégrateurs sur le premier canal

$$\xi = \eta \quad (5.40)$$

$$\dot{\eta} = \sigma_1 \quad (5.41)$$

où  $\sigma_1$  est la nouvelle entrée auxiliaire à la place de  $\xi$ , à partir de l'eq(5.39), la troisième dérivée de la sortie est donnée par :

$$\begin{bmatrix} y_1^{[3]} \\ y_2^{[3]} \end{bmatrix} = R(\theta) \begin{bmatrix} \eta + g_0 c \theta \dot{\theta} \\ \xi \dot{\theta} + g_0 s \theta \dot{\theta} \end{bmatrix}, \quad (5.42)$$

où nous avons utilisé l'eq(5.41) et la propriété

$$\dot{R}(\theta) = R(\theta)S(\dot{\theta}) = \begin{bmatrix} c\theta & -s\theta \\ s\theta & c\theta \end{bmatrix} \begin{bmatrix} 0 & -\dot{\theta} \\ \dot{\theta} & 0 \end{bmatrix}$$

à partir de l'équation (5.42),  $\eta$  peut être interprété comme la composante qui n'est pas due à la gravité de la secousse linéaire de la CP le long du dernier axe.

Enfin, on a  $\ddot{\theta} = -(\sigma_2 + g_0 c \theta)/K$ , la dérivée quatrième de la sortie est calculée

$$\begin{aligned} \begin{bmatrix} y_1^{[4]} \\ y_2^{[4]} \end{bmatrix} &= R(\theta) \left\{ \begin{bmatrix} 1 & -\frac{g_0}{K} c \theta \\ 0 & -\frac{\xi + g_0 s \theta}{K} \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \end{bmatrix} + \begin{bmatrix} -(2g_0 s \theta + \xi)\dot{\theta}^2 - \frac{g_0^2}{K} c \theta^2 \\ 2(g_0 c \theta \dot{\theta} + \eta)\dot{\theta} - \frac{g_0}{K}(\xi + g_0 s \theta)c\theta \end{bmatrix} \right\} \\ &\triangleq R(\theta) \left\{ A(\theta, \xi)\sigma + b(\theta, \dot{\theta}, \xi, \eta) \right\}. \end{aligned}$$

sous l'hypothèse de régularité que la matrice  $A(\theta, xi)$  est inversible ou, de façon équivalente, que

$$\rho \triangleq \xi + g_0 s \theta \neq 0, \quad (5.43)$$

l'inversion de commande basée sur

$$\sigma = A^{-1}(\theta, \xi) \left( R^T(\theta)v - b(\theta, \dot{\theta}, \xi, \eta) \right), \quad (5.44)$$

avec  $v = (v_1, v_2)$  les nouvelles entrées, telles que

$$\begin{bmatrix} y_1^{[4]} \\ y_2^{[4]} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \quad (5.45)$$

c'est à dire deux chaînes découplées de quatre intégrateurs entrée-sortie.

Puisque la dimension de l'état de robot  $(q, \dot{q})$ , (c-à-d,6), plus la dimension du compensateur d'état  $(\xi, \eta)$ , (c-à-d,2), est égale à la somme des degrés relatifs  $(4 + 4 = 8)$  des deux sorties en eq. (5.45), alors la linéarisation entrée-sortie est exacte qui a été réalisée. Ce qui implique que le système a une dynamique de zéro triviale par rapport à la sortie choisie (5.35).

Pour le système (5.34) la commande par la linéarisation dynamique est obtenue par combinaison des équations (5.38), (5.41), (5.41) et (5.44). Le bloc diagramme du contrôleur est donné par la Figure 5.12

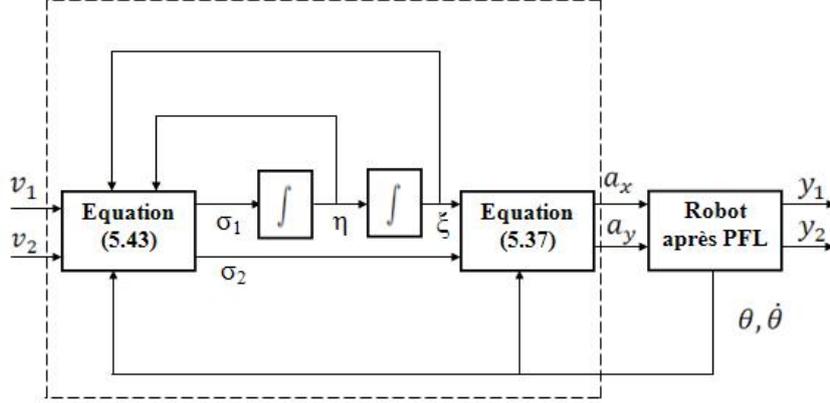


FIGURE 5.12 – Schéma du contrôleur dynamique linéarisé

### Planification et commande de trajectoire

Nous considérons le problème de planification d'une trajectoire admissible qui relie un état initial du robot avec l'état final désiré dans un temps fini  $T > 0$ , en présence de la contrainte différentielle non intégrable, ce problème est un exemple de planification de trajectoires non holonome. La technique de linéarisation par retour dynamique propose une approche simple et systématique pour résoudre le problème.

En fait, la planification d'un mouvement faisable pour le robot sous-actionné est équivalente à la planification d'un transfert d'État à État pour la représentation linéaire équivalent (5.44). Ce dernier peut être formulé comme un problème d'interpolation en utilisant des fonctions paramétriques lisses  $y_1(t)$ ,  $y_2(t)$ .

En particulier, supposons qu'au temps  $t = 0$ , le robot commence à partir d'un état générique  $(q_s, \dot{q}_s) = (x_s, y_s, \theta_s, \dot{x}_s, \dot{y}_s, \dot{\theta}_s)$  pour atteindre l'état final  $(q_g, \dot{q}_g) = (x_g, y_g, \theta_g, \dot{x}_g, \dot{y}_g, \dot{\theta}_g)$  à l'instant  $t = T$ . En vue de la transposition du problème de planification pour le système de bouclage linéarisé (5.44), on a besoin d'associer à  $(q_s, \dot{q}_s)$  et  $(q_g, \dot{q}_g)$  les conditions aux limites appropriées pour les nouvelles variables d'état, c'est à dire,  $y_1$ ,  $y_2$ , et leurs dérivés jusqu'à la troisième ordre, à l'instant  $t = 0$

$$\begin{bmatrix} y_1(0) \\ \dot{y}_1(0) \\ \ddot{y}_1(0) \\ y_1^{[3]}(0) \end{bmatrix} = \begin{bmatrix} y_{1s} \\ \dot{y}_{1s} \\ \ddot{y}_{1s} \\ y_{1s}^{[3]} \end{bmatrix}, \quad \begin{bmatrix} y_2(0) \\ \dot{y}_2(0) \\ \ddot{y}_2(0) \\ y_2^{[3]}(0) \end{bmatrix} = \begin{bmatrix} y_{2s} \\ \dot{y}_{2s} \\ \ddot{y}_{2s} \\ y_{2s}^{[3]} \end{bmatrix}$$

et à l'instant  $t = T$

$$\begin{bmatrix} y_1(T) \\ \dot{y}_1(T) \\ \ddot{y}_1(T) \\ y_1^{[3]}(T) \end{bmatrix} = \begin{bmatrix} y_{1g} \\ \dot{y}_{1g} \\ \ddot{y}_{1g} \\ y_{1g}^{[3]} \end{bmatrix}, \quad \begin{bmatrix} y_2(T) \\ \dot{y}_2(T) \\ \ddot{y}_2(T) \\ y_2^{[3]}(T) \end{bmatrix} = \begin{bmatrix} y_{2g} \\ \dot{y}_{2g} \\ \ddot{y}_{2g} \\ y_{2g}^{[3]} \end{bmatrix}$$

Pour ce dernier on utilise les équations (5.35), (5.35), (5.39) et (5.42), dans lesquels  $\xi(0) = \xi_s$ ,  $\xi(T) = \xi_g$ ,  $\eta(0) = \eta_s$  et  $\eta(T) = \eta_g$  sont encore des paramètres libres.

Une solution simple au problème d'interpolation est de générer des trajectoires comme

des polynômes de septième degré :

$$y_i(t) = \sum_{j=0}^7 a_{ij} \lambda^j, \quad i = 1, 2, \quad (5.46)$$

avec le temps normalisé  $\lambda = t/T$ . supprimons l'indice  $i$  les coefficient  $a_j$

$$\begin{aligned} a_0 &= y_s \\ a_1 &= \dot{y}_s T \\ a_2 &= \frac{1}{2} \ddot{y}_s T^2 \\ a_3 &= \frac{1}{6} y_s^{[3]} \\ a_4 &= 35(y_g - y_s) - (20\dot{y}_s + 15\dot{y}_g)T - (5\ddot{y}_s - \frac{5}{2}\ddot{y}_g)T^2 - (\frac{2}{3}y_s^{[3]} + \frac{1}{6}y_g^{[3]})T^3 \\ a_5 &= -84(y_g - y_s) + (45\dot{y}_s + 39\dot{y}_g)T + (10\ddot{y}_s - 7\ddot{y}_g)T^2 + (y_s^{[3]} + \frac{1}{2}y_g^{[3]})T^3 \\ a_6 &= 70(y_g - y_s) - (36\dot{y}_s + 34\dot{y}_g)T - (\frac{15}{2}\ddot{y}_s - \frac{13}{2}\ddot{y}_g)T^2 - (\frac{2}{3}y_s^{[3]} + \frac{1}{2}y_g^{[3]})T^3 \\ a_7 &= -20(y_g - y_s) + 10(\dot{y}_s + \dot{y}_g)T + 2(\ddot{y}_s - \ddot{y}_g)T^2 + \frac{1}{6}(y_s^{[3]} + y_g^{[3]})T^3 \end{aligned}$$

Le robot plus la trajectoire du compensateur d'état associée à la trajectoire de sortie de linéarisation (5.46) qui résout le problème de planification est obtenue par pur calculs algébriques utilisant les équations précédentes . En outre, les commandes en boucle ouverte qui réalisent cette trajectoire sont

$$v_i(t) = \frac{1}{T^4} (840a_{i7}\lambda^3 + 360a_{i6}\lambda^2 + 120a_{i5}\lambda + 24a_{i4}), \quad i = 1, 2,$$

qui représentent les entrées nominales au système (5.45) ou, de manière équivalente, au compensateur de retour dynamique (voir Figure 5.12), ce qui produit les entrées d'accélération  $(a_x, a_y)$ . La sélection d'états initial et final du compensateur  $(\xi_s, \eta_s)$  et  $(\xi_g, \eta_g)$  affecte les conditions aux limites, et ainsi le mouvement généré à l'intérieur de la classe choisie de fonctions d'interpolation. En particulier, les états du compensateur doivent être choisis de manière à éviter la singularité  $\rho = 0$  pendant le mouvement. Dans la prochaine sous section on utilise cette technique pour le cas d'un robot manipulateur type PPR planaire avec  $g_0 = 0$ .

## 5.2.2 Robot manipulateur de Type PPR

Dans cette section on considère la gravité  $g_0 = 0$ , et on veut réaliser la planification de trajectoire du repos au repos  $(\dot{x}_s = \dot{y}_s = \dot{\theta}_s = \dot{x}_g = \dot{y}_g = \dot{\theta}_g)$ . à partir des équations (5.35), (5.35),(5.39) et (5.42), nous obtenons les conditions aux limites pour la première sortie

$$\begin{bmatrix} y_{1s} \\ \dot{y}_{1s} \\ \ddot{y}_{1s} \\ y_{1s}^{[3]} \end{bmatrix} = \begin{bmatrix} x_s + Kc\theta_s \\ 0 \\ \xi_s c\theta_s \\ \eta_s c\theta_s \end{bmatrix}, \quad \begin{bmatrix} y_{1g} \\ \dot{y}_{1g} \\ \ddot{y}_{1g} \\ y_{1g}^{[3]} \end{bmatrix} = \begin{bmatrix} x_s + Kc\theta_s \\ 0 \\ \xi_s c\theta_s \\ \eta_s c\theta_s \end{bmatrix}, \quad (5.47)$$

et pour la deuxième sortie on a

$$\begin{bmatrix} y_{2s} \\ \dot{y}_{2s} \\ \ddot{y}_{2s} \\ y_{2s}^{[3]} \end{bmatrix} = \begin{bmatrix} y_s + K s \theta_s \\ 0 \\ \xi_s s \theta_s \\ \eta_s s \theta_s \end{bmatrix}, \quad \begin{bmatrix} y_{2g} \\ \dot{y}_{2g} \\ \ddot{y}_{2g} \\ y_{2g}^{[3]} \end{bmatrix} = \begin{bmatrix} y_g + K s \theta_g \\ 0 \\ \xi_g s \theta_g \\ \eta_g s \theta_g \end{bmatrix}, \quad (5.48)$$

on a  $\rho = \xi$  pour éviter la singularité il faut que  $\xi_s$  et  $\xi_g$  soient non nuls et de même signe. Cette condition est cependant nécessaire pour que  $\xi$  ne peut pas traverser zéro pendant le mouvement. D'après l'équation (5.39) il est simple de montrer que

$$\rho^2 = \ddot{y}_1^2 + (\ddot{y}_2 + g_0)^2 \quad (5.49)$$

dans notre cas  $g_0 = 0$  on voit que la singularité peut être atteinte si et seulement si les sorties d'accélération sont nulles en même temps, d'où avec cette méthode on ne peut pas avoir la trajectoire point à point.

### Résultats numériques

Soit le modèle dynamique du robot manipulateur PPR [18] avec la dernière liaison sous actionnée par les deux jonctions prismatique comme illustre la Figure 5.13

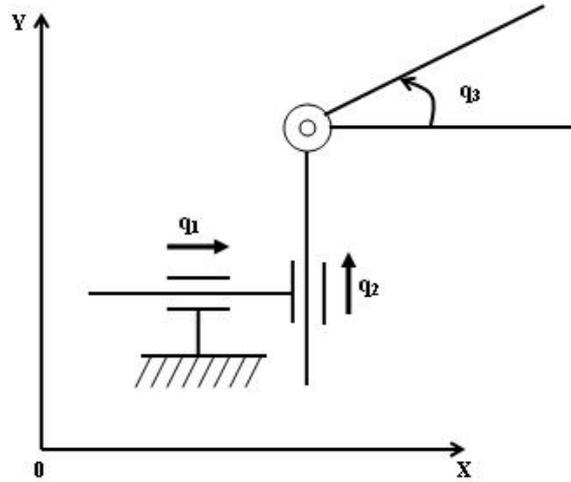


FIGURE 5.13 – Robot manipulateur PPR

avec  $q_1 = x$ ,  $q_2 = y$  et  $q_3 = \theta$ , le modèle est définie par

$$\begin{cases} m_x \ddot{x} - m_3 l \cos \theta \dot{\theta}^2 - m_3 l \sin \theta \ddot{\theta} = f_1 \\ m_y \ddot{y} - m_3 l \sin \theta \dot{\theta}^2 + m_3 l \cos \theta \ddot{\theta} = f_2 \\ I \ddot{\theta} + m_3 l \cos \theta \dot{y} - m_3 l \sin \theta \ddot{x} = 0 \end{cases} \quad (5.50)$$

où  $l = d$  correspond à la distance du centre de masse par rapport à la base du troisième axe. Pour illustrer les performances du planificateur, nous présentons un résultat typique obtenu pour la tâche du repos au repos

$$\begin{bmatrix} x_s \\ y_s \\ \theta_s \end{bmatrix} = \begin{bmatrix} 0.5m \\ 1m \\ 0^\circ \end{bmatrix} \rightarrow \begin{bmatrix} x_g \\ y_g \\ \theta_g \end{bmatrix} = \begin{bmatrix} 1.5m \\ 2m \\ 45^\circ \end{bmatrix},$$

où  $m_x = m_1 + m_2 + m_3$ ,  $m_y = m_2 + m_3$  et  $I = I_3 + m_3 d^2$  avec  $T = 10s$ ,  $K = 2/3(l_3 = 1m)$ ,  $\xi_s = \xi_g = -0.1m/s^2$ .

Le mouvement cartésien du troisième axe correspondant à la trajectoire planifiée pour le CP est donné dans la Figure 5.14, le mouvement du robot est donné par la Figure 5.15, les vitesses correspondantes sont illustrées par la Figure 5.16 et les forces appliquées au robot sont données par la Figure 5.17, les données utilisées dans la simulation sont  $m_1 = m_2 = m_3 = 1Kg$ ,  $d = 0.5m$  et  $I_3 = (m_3 d^2)/3$ .

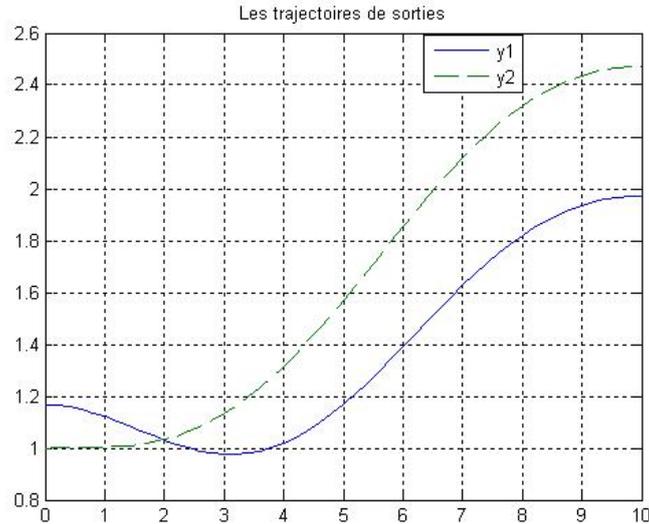


FIGURE 5.14 – Les trajectoires de sorties  $y_1$  et  $y_2$  (m)

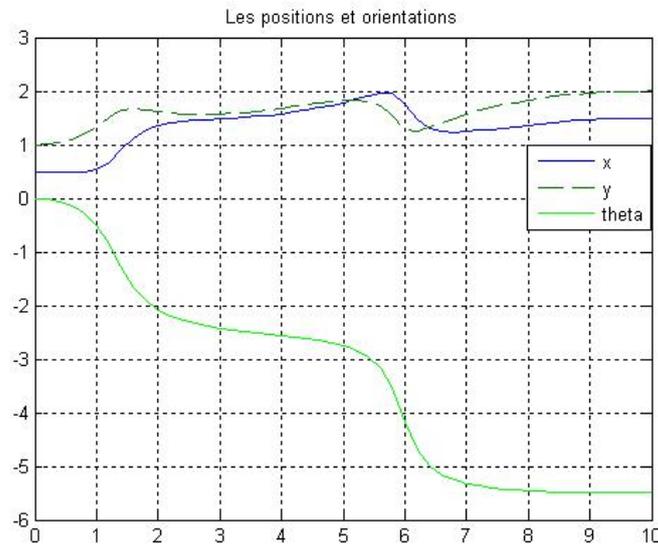


FIGURE 5.15 – Planification du repos au repos :  $x(m)$ ,  $y(m)$  et  $\theta(rd)$

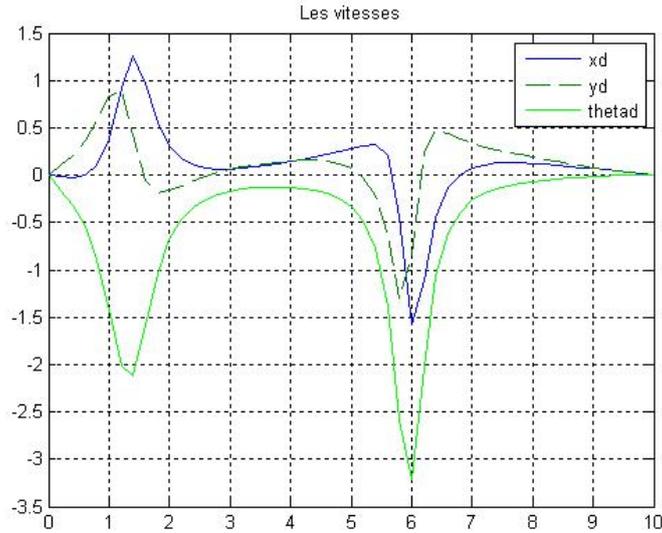


FIGURE 5.16 – Les vitesses du Robot  $x(\text{m/s})$ ,  $y(\text{m/s})$  et  $\theta(\text{rd/s})$

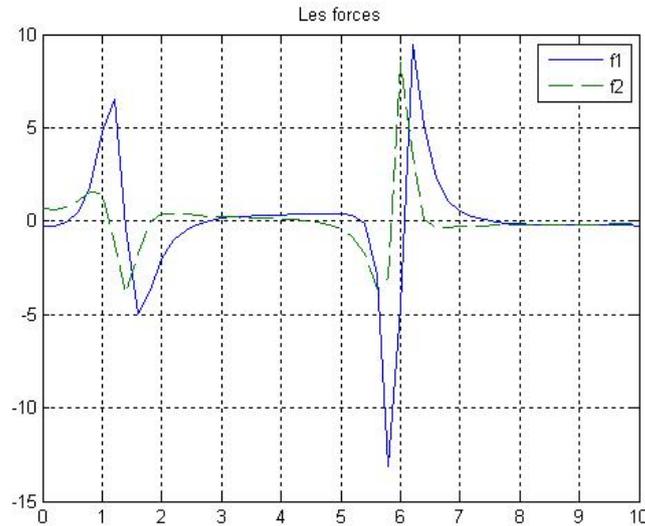


FIGURE 5.17 – Forces appliquée au robot  $f_1$ ,  $f_2$  (N)

Notons que les vitesses élevées et les pics dans les forces autour de  $t = 6\text{s}$ , correspondent à la rotation rapide du troisième axe approximativement autour de son CP et donc diminuent l'indice de régularité  $\rho$ .

### 5.2.3 Conclusion

On voit bien que les entrées correspondant à l'évolution du centre de percussion quand on les a injecté au contrôleur dynamique sont les même en sorties, cela veut dire que la linéarisation est exacte par le compensateur dynamique, le problème de ce dernier est basé sur le choix des sorties du système (pour notre cas les coordonnées du CP), les résultats du compensateur dynamique donne les forces et/ou couples, vitesses et accélération nominaux selon le temps final qu'on veut utiliser  $T$ . Pratiquement il faut d'abord chercher le temps optimal  $T$  passant par l'optimisation sous contraintes (Chapitre1, Section1.2), et utiliser

ce temps dans le processus de linéarisation par compensateur dynamique pour obtenir les différentes quantités nominales qui permettent le fonctionnement du robot.

# Conclusion générale

Dans ce travail on s'est intéressé au problème de génération et planification optimale de trajectoire pour les systèmes mécaniques sous actionnés, pour ceci on a mis l'accent sur les points suivants

- La génération de trajectoire
- Les commandes des S.M.S.A.
- Les systèmes mécaniques non holonomes.
- Méthode d'optimisation

Enfin on a terminé par une application où nous avons montré les difficultés de contrôle d'un manipulateur  $R_a R_p$  sur le plan horizontal, problème n'ayant pas de solution de réponse pour l'instant, après on est passé à la planification de trajectoire pour les manipulateurs qui ont un nombre de degrés de liberté supérieur ou égal à trois à l'aide de la linéarisation dynamique par feedback qui donne des résultats intéressants.

En général, les manipulateurs sous actionnés soumis à des forces de gravité sont stabilisables par une commande continue et admettent des trajectoires asymptotiques pour le problème de planification, comme par exemple l'Acrobot et le Pendubot décrits par Spong.

Les systèmes mécaniques avec des contraintes de second ordre ne peuvent pas être stabilisés par un feedback statique continu, plus la présence des termes non linéaire dans l'équation dynamique de mouvement impose des difficultés au niveau de la planification de trajectoire.

Il n'existe pas de méthode générale pour synthétiser des lois de commande pour Les systèmes mécaniques sous actionnés, il est difficile d'imposer des trajectoires aux axes passifs. Seuls certains cas particulier sont traités dans la littérature. Nous pensons que la commande des systèmes mécaniques sous actionnés est un problème très riche et qui est loin d'être cloturé.

# Annexe A

## Les outils de la théorie du contrôle non linéaire

L'analyse des systèmes de contrôle non-linéaire nécessite de nombreux concepts de la géométrie différentielle. À ce stade, des définitions introductives et des résultats fondamentaux (théorème de Frobenius) sont brièvement examinés. Ensuite, nous rappelons les différents types de contrôlabilité non linéaire et leurs conditions, qui seront utilisées dans la section suivante pour caractériser les contraintes non holonomes. Enfin, les éléments de base du problème de stabilisation des systèmes non linéaires seront introduits.

### A.1 Éléments de géométrie différentielle

Pour plus de simplicité, nous allons travailler avec des vecteurs  $x \in \mathbb{R}^n$  et notons l'espace tangent  $\mathbb{R}^n$  en  $x$  par  $T_x(\mathbb{R}^n)$ . Un champ de vecteurs lisse  $g : \mathbb{R}^n \rightarrow T_x(\mathbb{R}^n)$  est une application lisse attribuant à chaque point  $x \in \mathbb{R}^n$  un vecteur tangent  $g(x) \in T_x(\mathbb{R}^n)$ . Si  $g(x)$  est utilisé pour définir une équation différentielle

$$\dot{x} = g(x),$$

$\phi_t^g(x)$  le flot des champs de vecteurs  $g$  est l'application qui associe à chaque point  $x$  la solution au temps  $t$  de l'équation différentielle en évolution de  $x$  au temps 0, ou

$$\frac{d}{dt}\phi_t^g(x) = g(\phi_t^g(x)).$$

Il est possible de montrer que la famille des applications  $\{\phi_t^g(x)\}$  à un paramètre (à savoir  $T$ ) est le groupe des difféomorphismes locaux sous l'opération de composition. Ainsi

$$\phi_{t_1}^g \circ \phi_{t_2}^g = \phi_{t_1+t_2}^g.$$

Par exemple, dans les systèmes linéaires,  $g(x) = Ax$  et le flot est l'opérateur linéaire  $\phi_t^g = e^{At}$ .

Étant donné deux champs de vecteurs lisses  $g_1$  et  $g_2$ , nous notons que la composition de leur flot est généralement non-commutative :

$$\phi_t^{g_1} \circ \phi_s^{g_2} \neq \phi_s^{g_2} \circ \phi_t^{g_1}.$$

Par ailleurs, le nouveau champ de vecteurs  $[g_1, g_2]$ , dont l'expression en fonction des coordonnées est :

$$[g_1, g_2](x) = \frac{\partial g_2}{\partial x} g_1(x) - \frac{\partial g_1}{\partial x} g_2(x)$$

est appelé le crochet de Lie de  $g_1$  et  $g_2$ . Deux champs de vecteurs  $g_1$  et  $g_2$  sont dit commutat si  $[g_1, g_2] = 0$ . Pour apprécier la pertinence de l'opération crochet de Lie, considérons l'équation différentielle

$$\dot{x} = g_1(x)u_1 + g_2(x)u_2 \quad (\text{A.1})$$

associés aux deux champs vecteurs  $g_1$  et  $g_2$ . Si les deux entrées  $u_1$  et  $u_2$  ne sont jamais actives au même instant, la solution de l'équation A.1 est obtenue par la composition des flots de  $g_1$  et  $g_2$ . En particulier, considérons la séquence d'entrée suivante :

$$u(t) = \begin{cases} u_1(t) = +1, u_2(t) = 0, t \in [0, \xi), \\ u_1(t) = 0, u_2(t) = +1, t \in [\xi, 2\xi), \\ u_1(t) = -1, u_2(t) = 0, t \in [2\xi, 3\xi), \\ u_1(t) = 0, u_2(t) = -1, t \in [3\xi, 4\xi), \end{cases} \quad (\text{A.2})$$

où  $\xi$  est un intervalle de temps infinitésimal. La solution de l'équation différentielle au  $4\xi$  temps est obtenu en suivant le flot de  $g_1$  puis  $g_2$ , puis  $-g_1$  et enfin  $-g_2$  (voir Figure-fig :mouvementselon surface). En calculant  $x(\xi)$  comme un développement en série autour de  $x_0 = x(0)$  le long de  $g_1$ ,  $x(2\xi)$  comme un développement en série autour de  $x(\xi)$  le long de  $g_2$ , et ainsi de suite, on obtient

$$x(4\xi) = \phi_\xi^{-g_2} \circ \phi_\xi^{-g_1} \circ \phi_\xi^{g_2} \circ \phi_\xi^{g_1}(x_0) \quad (\text{A.3})$$

$$= x_0 + \xi^2 \left( \frac{\partial g_2}{\partial x} g_1(x_0) - \frac{\partial g_1}{\partial x} g_2(x_0) \right) + O(\xi^3), \quad (\text{A.4})$$

Pour plus de détails consulter [23]. Notez que, lorsque  $g_1, g_2$  commutent, aucun mouvement net n'est obtenue à la suite de la séquence d'entrée A.2.

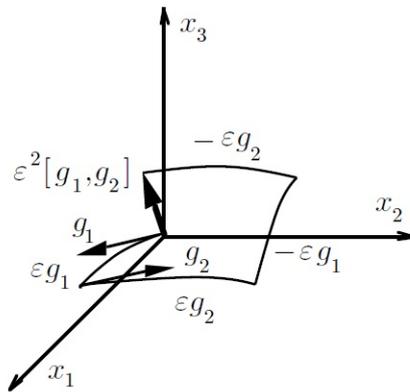


FIGURE A.1 – Mouvement par crochet de Lie

Le calcul ci-dessus montre que, à chaque point, le mouvement infinitésimal est possible non seulement dans les directions contenues dans la direction des champs de vecteurs d'entrée, mais aussi dans les directions de leurs crochets de Lie. Ceci est propre à la non-linéarité des champs de vecteurs d'entrée dans le système de contrôle avec dérivée A.1. De même, on peut prouver que, en utilisant des séquences d'entrée plus compliquées, il est possible d'obtenir un mouvement dans la direction des crochets d'ordre supérieur, comme  $[g_1, [g_1, g_2]]$ . Des procédés similaires constructifs pour la caractérisation des mouvements admissibles peuvent être conçus aussi pour les systèmes de contrôle avec un champ  $f$  de dérive avec les opérations impliquant un mélange "tranche" de  $f$  et  $g_i$ .

**Exemple 9.** pour un système linéaire avec une seule entrée

$$\dot{x} = Ax + bu,$$

avec la dérivée  $f(x) = Ax$  et le vecteur d'entrée  $g(x) = b$ , le mouvement peut être obtenu dans le sens des crochets de Lie (répétés)

$$\begin{aligned} -[f, g] &= AB \\ [f, [f, g]] &= A^2B \\ -[f, [f, [f, g]]] &= A^3B \\ &\vdots \end{aligned}$$

un résultat bien connu.

La dérivée de Lie de  $\alpha : \mathbb{R}^n \rightarrow \mathbb{R}$  long de  $g$  est définie comme

$$L_g\alpha(x) = \frac{\partial\alpha}{\partial x}g(x).$$

Les propriétés les plus importantes des crochets de Lie, qui sont utiles dans les calculs, sont

$$\begin{aligned} [f, g] &= -[g, f] \quad (\text{antisymétrique}) \\ [f, [g, h]] + [h, [f, g]] + [g, [h, f]] &= 0 \\ [\alpha f, \beta g] &= \alpha\beta[f, g] + \alpha(L_f\beta)g - \beta(L_g\alpha)f \end{aligned}$$

avec  $\alpha, \beta : \mathbb{R}^n \rightarrow \mathbb{R}$ . L'espace vectoriel  $V(\mathbb{R}^n)$  des champs de vecteurs lisses sur  $\mathbb{R}^n$ , muni du crochet de Lie en tant que produit, est appelée une algèbre de Lie.

La distribution lisse  $\delta$  associée aux  $m$  champs de vecteurs lisses  $\{g_1, \dots, g_m\}$ , est l'application qui assigne à chaque point  $x \in \mathbb{R}^n$  un sous-espace linéaire de son espace tangent, c-à-d

$$\Delta(x) = \text{span}\{g_1, \dots, g_m\} \subset T_x(\mathbb{R}^n).$$

Ci-après, nous allons utiliser la notation courte

$$\Delta = \text{span}\{g_1, \dots, g_m\}$$

La distribution  $\Delta$  est dite inversible si  $\dim\Delta(x) = r$ , constant pour tout  $x$ . Dans ce cas,  $r$  est appelé la dimension de la distribution. La distribution  $\Delta$  est involutive si elle est fermée sous l'opération crochet de Lie :

$$[g_i, g_j] \in \Delta, \forall g_i, g_j \in \Delta.$$

La fermeture involutive  $\overline{\Delta}$  de la distribution  $\Delta$  est sa fermeture sous l'opération crochet de Lie, d'où  $\Delta$  est involutive si  $\overline{\Delta} = \Delta$ .

Une distribution non singulière  $\Delta$  de  $\mathbb{R}^n$  de dimension  $k$  est complètement intégrable s'il existe  $n - k$  fonctions indépendante  $h_i$  telle que

$$L_{g_j}h_i(x) = \frac{\partial h_i}{\partial x}g_j(x) = 0, \quad \forall x \in \mathbb{R}^n, \forall g_j \in \Delta, \quad i = 1, \dots, n - k. \quad (\text{A.5})$$

Dans ce cas, chaque hypersurface est défini par les surfaces de niveau

$$\{x \in \mathbb{R}^n : h_1(x) = c_1, \dots, h_{n-k}(x) = c_{n-k}\}$$

est appelé une variété intégrale de  $\Delta$ . l'équation A.5 indique que  $\Delta(x)$  coïncide avec l'espace tangent à sa variété intégrale en  $x$ . Le résultat suivant donne une condition nécessaire et suffisante pour l'intégration complète d'une distribution.

**Théorème 3** (Frobenius). *La distribution non singulière est complètement intégrable si et seulement si elle est involutive.*

**Exemple 10.** Soit  $\Delta = \text{span} \{g_1, g_2\}$  avec

$$g_1(x) = \begin{bmatrix} x_2 \\ 0 \\ 1 \end{bmatrix} \quad g_2(x) = \begin{bmatrix} x_3 \\ 1 \\ 0 \end{bmatrix}.$$

La dimension de  $\Delta$  est 2 pour n'importe quel  $x \in \mathbb{R}^3$ . Un calcul simple montre que  $[g_1, g_2] = 0$ , d'où  $\Delta$  est involutive donc intégrable. En effet, il provoque une foliation dans  $\mathbb{R}^3$  de la forme

$$x_1 - x_2x_3 = c,$$

avec  $c \in \mathbb{R}$ .

Notons que une distribution générée par un seul vecteur est toujours involutive par conséquent elle est intégrable.

### A.1.1 Contrôlabilité

Soit le système non linéaire de la forme

$$\dot{x} = f(x) + \sum_{j=1}^m g_j(x)u_j \tag{A.6}$$

est appelé affine en la commande  $u_j$ . le vecteur d'état  $x \in \mathbb{R}^n$  et chaque composante de commande  $u \in \mathbb{R}^m$  prennent des valeurs dans la classe des fonctions polynomiales constante  $\mathcal{U}$  par rapport au temps. Par convention, le vecteur dérive  $f$  est supposé continu avec les vecteurs d'entrées  $g_j$ . Notons par  $x(t, 0, x_0, u)$  la solution unique de l'éq. (A.6) à l'instant  $t \geq 0$ , correspondant à la fonction d'entrée de commande  $u(\cdot)$  et la condition initial  $x(0) = x_0$ .

Le système (A.6) est contrôlable si, pour n'importe quel choix de  $x_1, x_2 \in \mathbb{R}^n$ , il existe un temps fini  $T$  et une commande  $u : [0, T] \rightarrow \mathcal{U}$  telle que  $x(T, 0, x_1, u) = x_2$ . Malheureusement, les critères généraux pour la vérification de cette forme naturelle de la contrôlabilité n'existent pas. Pour cette raison, d'autres caractérisations structurelles du système (A.6) ont été proposées, qui sont liées à la définition précédente.

Soit un voisinage  $V$  de  $x_0$ , noté  $\mathcal{R}^V(x_0, \tau)$  l'ensemble des états  $\xi$  pour lesquels il existe  $u : [0, \tau] \rightarrow \mathcal{U}$  telle que  $x(\tau, 0, x_0, u) = \xi$  et  $x(t, 0, x_0, u) \in V$  pour  $t \leq \tau$ . C'est à dire,  $\mathcal{R}^V(x_0, \tau)$  est l'ensemble des états accessibles au temps  $\tau$  de  $x_0$  avec des trajectoires contenues dans  $V$ . En outre on a

$$\mathcal{R}_T^V(x_0) = \bigcup_{\tau \leq T} \mathcal{R}^V(x_0, \tau),$$

est l'ensemble des états atteignable en un temps  $T$  à partir de  $x_0$  avec les trajectoires contenues dans un voisinage  $V$ .

Le système de commande (A.6) est dit :

1. Localement accessible à partir de  $x_0$  si, pour tout voisinage  $V$  de  $x_0$  et tout  $T$ ,  $\mathcal{R}_T^V(x_0)$  contient un ensemble ouvert non vide  $\Omega$ .
2. Localement contrôlable en temps petit à partir de  $x_0$  si, pour tout voisinage  $V$  de  $x_0$  et tout  $T$ ,  $\mathcal{R}_T^V(x_0)$  contient un voisinage non vide de  $x_0$ .

**Exemple 11.** Soit le système de contrôle dans  $\mathbb{R}^2$

$$\dot{x} = \begin{bmatrix} x_2^2 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u.$$

Pour n'importe quelle condition initiale, l'état initial  $x_1$  augmente par contre la seconde composante ne bouge dans aucune direction. D'où,  $\mathcal{R}_T^V(x_0)$  contient un ensemble non vide ouvert  $\Omega$ , mais il ne contient aucun voisinage de  $x_0 = (x_{01}, x_{02})$ . Comme conséquence le système est localement accessible à partir de n'importe quelle point, mais il n'est pas localement commandable en temps petit, le système n'est pas commandable.

Notons que :

- Les définitions précédentes sont de nature locale. Elles peuvent être globalisées en disant que le système (A.6) est localement accessible, ou localement contrôlable en un temps petit, s'il est de nature pour tous  $x_0 \in \mathbb{R}^n$ .
- La commandabilité locale en temps petit implique l'accessibilité locale, ainsi que la contrôlabilité, mais l'accessibilité locale n'implique pas la contrôlabilité en générale. Toutefois, si aucun vecteur de dérive n'est présent, alors l'accessibilité locale implique la contrôlabilité.

Définition de l'algèbre d'accessibilité  $\mathcal{C}$  pour le système de contrôle (A.6) comme la plus petite sous-algèbre de  $\nu(\mathbb{R}^n)$  contenant  $f, g_1, \dots, g_m$ . Notez que, par définition, tous les crochets de Lie (répétés) de ces champs de vecteurs appartiennent aussi à  $\mathcal{C}$ . La distribution accessible  $\Delta_c$  du système (A.6) est définie par

$$\Delta_c = \text{span} \{v | v \in \mathcal{C}\},$$

Le calcul de  $\Delta_c$  peut être organisé comme une procédure itérative :

$$\Delta_c = \text{span} \{v | v \in \Delta_i, \forall i \geq 1\},$$

avec

$$\begin{aligned} \Delta_1 &= \{f, g_1, \dots, g_m\} \\ \Delta_i &= \Delta_{i-1} + \text{span} \{[g, v] | g \in \Delta_1, v \in \Delta_{i-1}\}, \quad i \geq 2. \end{aligned}$$

La procédure ci-dessus s'arrête après  $k$  étapes, où  $k$  est le plus petit entier tel que  $\Delta_{k+1} = \Delta_k = \Delta_c$ . Comme  $\dim \Delta_c \leq n$  nécessairement, il s'ensuit que l'on s'arrête après au plus  $n - m$  étapes.

La distribution d'accessibilité peut être utilisée pour vérifier l'accessibilité locale, comme indiqué par un résultat de base, à savoir,

**Théorème 4 (Chow).** Si la condition de rang d'accessibilité

$$\dim \Delta_c(x_0) = n \tag{A.7}$$

est vérifiée, alors le système de contrôle (A.6) est localement accessible à partir de  $x_0$ . Si la condition de rang d'accessibilité est vraie pour tout  $x \in \mathbb{R}^n$ , le système est accessible localement.

En particulier, si les champs de vecteurs du système sont analytiques, la condition de rang d'accessibilité est nécessaire et suffisante pour l'accessibilité locale. Si le théorème de Chow est appliquée à un système de contrôle non linéaire

$$\dot{x} = \sum_{i=1}^m g_i(x) u_i, \tag{A.8}$$

il fournit une condition suffisante pour la contrôlabilité. La même chose est vraie pour les systèmes avec dérive  $f(x)$  telle que

$$f(x) \in \text{span} \{g_1(x), \dots, g_m(x)\}, \quad \forall x \in \mathbb{R}^n.$$

Par ailleurs, si le système (A.6) est commandable, puis son extension dynamique

$$\dot{x} = \sum_{i=1}^m g_i(x)v_i \tag{A.9}$$

$$\dot{v}_i = u_i, \quad i = 1, \dots, m, \tag{A.10}$$

est également contrôlable. L'inverse est trivialement vrai.

Comme pour la contrôlabilité locale en temps petit, seule une condition suffisante existe, basée sur le concept suivant : Considérons un champ de vecteurs  $v \in \Delta_c$  obtenu sous forme de crochet de Lie (répété) du champ de vecteur système, et désignons par  $\delta^0(v), \delta^1(v), \dots, \delta^m(v)$  sont même,  $v$  peut être écrite comme une combinaison linéaire des crochets de degré inférieur. Alors, le système (A.6) est localement contrôlable en temps petit à partir de  $x_0$ .

**Théorème 5** (Sussmann). *Supposons que  $\dim \Delta_C(x_0) = n$  et que, pour tout crochet  $v \in \Delta_C$ , tels que  $\delta^0(v)$  est impair et  $\delta^1(v), \dots, \delta^m(v)$  sont pairs,  $v$  peut être écrit comme une combinaison linéaire des crochets de degré inférieur. Alors, le système (A.6) est localement contrôlable en temps petit à partir de  $x_0$ .*

**Remarques 6.** *Quelques remarques en guise de conclusion.*

- *Supposons que l'accessibilité de la distribution  $\Delta_c$  à une dimension constante  $\nu < n$  partout. Alors, sur la base du théorème de Frobenius, il est possible de montrer que, pour tout  $x_0, T$  et  $V$ ,  $R_T^V(x_0)$  est contenu dans une variété intégrale de dimension  $\nu$  de  $\Delta_c$ , de plus,  $R_T^V(x_0)$  contient lui-même un ensemble non vide de dimension  $\nu$ . Comme dans le théorème de Chow, pour avoir la réciproque, il est nécessaire que  $\dim \Delta_c(x) = \nu$  dans une partie ouverte et dense de  $\mathbb{R}^n$ .*
- *Le terme 'Local' peut être relaxé dans les définitions ci-dessus si le système est analytique, puisque l'exigence que les trajectoires restent dans un voisinage de  $x_0$  peuvent être omises.*
- *Dans le cas linéaire*

$$\dot{x} = Ax + \sum_{j=1}^m b_j u_j = Ax + Bu \tag{A.11}$$

*toutes les définitions précédentes sont globales et se réduisent au concept de contrôlabilité linéaire classique. En particulier, la condition de rang d'accessibilité en  $x_0 = 0$  correspond à*

$$\text{rang} [B \quad AB \quad A^2B \quad \dots \quad A^{n-1}B] = n \tag{A.12}$$

*la condition bien connue de Kalman nécessaire et suffisante pour la contrôlabilité.*

# Annexe B

## Classification des systèmes non holonomes

À partir des résultats de contrôlabilité rappelés dans la dernière section, nous allons maintenant donner des conditions pour l'intégrabilité de l'ensemble des contraintes cinématiques (3.3), qui est repris ci-dessous pour plus de commodité

$$a_i^T(q)\dot{q} = 0, \quad i = 1, \dots, k < n, \quad (\text{B.1})$$

ainsi que le modèle cinématique associé

$$\dot{q} = \sum_{j=1}^m g_j(q)u_j = G(q)u, \quad m = n - k. \quad (\text{B.2})$$

la distribution d'accessibilité  $\Delta_c$  du modèle cinématique est la fermeture involutive de la distribution non singulière  $\Delta = \text{span} \{g_1, \dots, g_m\}$  Nous avons maintenant

**Proposition 6.** *L'ensemble des  $k$  contraintes Pfaffienne (B.1) est holonome si et seulement si son modèle cinématique associé (B.2) est tel que*

$$\dim \Delta_c = \dim \Delta = m, \quad (\text{B.3})$$

*c'est-à-dire, la distribution  $\Delta$  est involutive.*

**Remarques 7.** *Deux remarques s'imposent :*

- *Le lecteur peut vérifier que, dans le cas d'une contrainte différentielle unique (3.7), la condition (B.3) coïncide avec les conditions d'intégrabilité (3.9).*
- *Dans le cas particulier  $k = n - 1$  contraintes cinématiques, le modèle cinématique associé se compose d'un champ de vecteur unique ( $m = 1$ ), la distribution correspondante est toujours involutive. Par conséquent, le système mécanique est holonome. En particulier, cela se produit pour un système à deux dimensions soumis à une contrainte différentielle scalaire, comme nous le verrons à travers un exemple.*

La proposition 1 montre que  $\dim \Delta_c > m$  est une condition nécessaire et suffisante pour l'ensemble des contraintes cinématiques (B.1) d'être non holonomes. Cependant, nous pouvons être plus précis, et on peut distinguer entre la holonomie partielle ou complète.

**Proposition 7.** *L'ensemble des  $k$  contraintes pfaffiennes (B.1) contient un sous-ensemble de contraintes  $p$  intégrables si et seulement si le modèle associé cinématique (B.2) est tel que*

$$\dim \Delta_C = n - p.$$

si  $p = 0$  ou

$$\dim \Delta_C = n, \tag{B.4}$$

le système est complètement non holonome.

En particulier, notez que, si  $p \geq 1$ , par le théorème de Frobenius, il existe une variété intégrale de  $\Delta_C$  de dimension  $(n - p)$  sur lequel le mouvement du système est contraint (une feuille du feuilletage correspondant). Dans le cas particulier  $p = 0$ , le théorème de Chow s'applique. Comme indiqué par la proposition 2, un système soumis à  $k$  contraintes cinématiques est complètement non holonome si la distribution d'accessibilité associée  $\Delta_C$  génère  $\mathbb{R}^n$ .

Pour vérifier cette condition, on doit effectuer la procédure itérative de la section A.1.1, ce qui revient aux calculs répétés des crochets de Lie des champs de vecteurs d'entrée,  $g_1, \dots, g_m$  du système (B.2). Pour cette raison, nous donnons ci-dessous une classification des systèmes non holonomes basée sur la séquence et l'ordre des crochets de Lie dans l'algèbre d'accessibilité correspondant.

Soit  $\Delta = \text{span} \{g_1, \dots, g_m\}$ . Définissons la fibration engendrée par la distribution  $\Delta$  comme la séquence  $\{\Delta_i\}$ , avec

$$\begin{aligned} \Delta_1 &= \Delta \\ \Delta_i &= \Delta_{i-1} + [\Delta_1, \Delta_{i-1}], \quad i \geq 2, \end{aligned}$$

où

$$[\Delta_1, \Delta_{i-1}] = \text{span} \{[g_j, v] \mid g_j \in \Delta_1, v \in \Delta_{i-1}\}.$$

Notons que, par la construction  $\Delta_i \subseteq \Delta_{i+1}$ . En outre, à partir de l'identité de Jacobi il résulte que  $[\Delta_i, \Delta_j] \subseteq [\Delta_1, \Delta_{i+j-1}] \subseteq \Delta_{i+j}$ .

Une fibration est régulière dans un voisinage  $V$  de  $q_0$  si

$$\dim \Delta_i(q) = \dim \Delta_i(q_0), \quad \forall q \in V.$$

Pour une fibration régulière, si  $\dim \Delta_{i+1} = \dim \Delta_i$ , alors  $\Delta_i$  est involutive et  $\Delta_{i+j} = \Delta_i$ , pour tout  $j \geq 0$ . Comme  $\dim \Delta_1 = m$  et  $\dim \Delta_i = n$  toujours, la condition d'arrêt est satisfaite après au plus  $n - m = k$  étapes, c'est à dire, le nombre de contraintes cinématiques originales.

Si la fibration générée par une distribution  $\Delta$  est régulière, il est possible de définir le degré de non holonomie de  $\Delta$  comme le plus petit entier  $\kappa$  tel que

$$\dim \Delta_{\kappa+1} = \dim \Delta_\kappa.$$

Le raisonnement qui précède implique que  $\kappa \leq k + 1$ . Le vecteur de croissance  $r \in \mathcal{Z}^\kappa$  est définie comme

$$r_i = \dim \Delta_i, \quad i = 1, \dots, \kappa,$$

et le vecteur de croissance relatif  $\sigma \in \mathcal{Z}^\kappa$  est donné par

$$\sigma_i = r_i - r_{i-1}, \quad i = 1, \dots, \kappa \quad r_0 = 0.$$

Les conditions précédentes d'holonomie, de non holonomie partielle et complète non holonomie peuvent être reformulées en termes des concepts définis ci-dessus. L'ensemble des  $k$  contraintes cinématiques (B.1) est :

1. Holonome, si  $\kappa = 1$  (ou  $\dim \Delta_\kappa = m$ ).
2. Non holonome, si  $2 \leq \kappa \leq k$ . En particulier, l'ensemble des contraintes (B.1) est :
  - (a) Partiellement non holonome, si  $m + 1 \leq \dim \Delta_\kappa < n$ .
  - (b) Complètement non holonome, si  $\dim \Delta_\kappa = n$ .

Nous concluons cette section en soulignant que d'une analyse similaire peut être utilisée pour des contraintes cinématiques dans une forme plus générale que éq. (B.1), à savoir

$$a_i^T(q)\dot{q} = \gamma, \quad i = 1, \dots, k < n, \quad \text{ou} \quad A^T(q)\dot{q} = \gamma, \quad (\text{B.5})$$

avec  $\gamma_i$  des constants. Par exemple, cette forme provient de la conservation de moment angulaire non nulle des robots dans l'espace.

Le modèle cinématique associés à des contraintes différentielles de la forme (7.23) décrit les vitesses admissibles généralisées que

$$\dot{q} = f(q) + \sum_{j=1}^m g_j(q)u_j = f(q) + G(q)u, \quad m = n - k, \quad (\text{B.6})$$

c-à-d, un système de commande non linéaire avec la dérive. Les colonnes de  $G$  sont à nouveau une base pour le noyau à  $m$  dimensions de la matrice  $A(q)$ , tandis que le champ de vecteurs de dérive peut être obtenu, par exemple, par l'intermédiaire pseudo inversion

$$f(q) = A^\#(q)\gamma = A(q)(A^T(q)A(q))^{-1}\gamma.$$

Pour décider si les contraintes de Pfaff généralisées (B.5) sont non holonomes, on peut utiliser les mêmes outils présentés dans cette section, à savoir analyser la dimension de la distribution de l'accessibilité  $\Delta_C$  du modèle cinématique (B.6). Toutefois, lorsqu'il s'agit de la commande des systèmes avec des contraintes de Pfaff généralisées, la présence de terme de dérive  $f \notin \text{span}\{g_1, \dots, g_m\}$  implique que l'accessibilité n'est pas équivalente à la contrôlabilité. Dans ce cas, on peut utiliser une condition suffisante de Sussmann (théorème 5, section A.1.1) pour vérifier la contrôlabilité local dans un petit temps.

# Bibliographie

- [1] N. P. Aneke. "*Control of underactuated mechanical systems*". PhD thesis, Technische Universiteit Eindhoven, 2003.
- [2] H. Arai et S. Tachi. "Position control of a manipulator with passive joints using dynamic coupling". *International Symposium on Industrial Robots*, 23 :617–624, 1989.
- [3] H. Arai, K. Tanie et N. Shiroma. "time-scaling control of underactuated manipulator". *Robotic Systems*, 15 :525–536, 1998.
- [4] M. Bartholomew Biggs. "*Nonlinear Optimization with Engineering Applications*". Springer, 2008.
- [5] L. Biagiotti et C. Melchiorri. *Trajectory Planning for Automatic Machines and Robots*. Springer, 2008.
- [6] A. M. Bloch, P. Crouch, J. Baillieul et J. Marsden. *Nonholonomic mechanics and control*. Springer, 2003.
- [7] T. Chettibi et M. Haddad. "Generating optimal reference trajectories for underactuated serial manipulators with a single passive joint". In *cinquième Conférence sur le Génie Électrique CGE'05*, avril 2007.
- [8] W. Chung. *Nonholonomic Manipulators*. Springer, 2004.
- [9] M. Clerc. *Particle Swarm Optimization*. ISTE, 2006.
- [10] D. A. Coley. *Introduction to Genetic Algorithms for Scientists and Engineers*. World Scientific, 1999.
- [11] J. Cortés. "*Geometric Control and Numerical Aspects of Nonholonomic Systems*". Springer, 2002.
- [12] A. De Luca et G. Oriolo. *Modelling and control of nonholonomic mechanical systems*. Springer Verlag, 1995.
- [13] A. De Luca et G. Oriolo. "trajectory planning and control for planar robots with passive last joint". *Robotics Research*, 21(5-6) :575–590, 2002.
- [14] A. De Luca, L. Lanari et G. Oriolo. "A sensitivity approach to optimal spline robot trajectories". *International Federation of Automatic Control*, 27 :535–539, 1991.
- [15] A. De Luca, R. Mattone et G. Oriolo. "Stabilization of an underactuated planar 2r manipulator". *Robust Nonlinear Control*, 10 :181–198, 2000.
- [16] A. De Luca, S. Ianniti, R. Mattone et G. Oriolo. "Underactuated Manipulators : Control Properties and Techniques". *Machine Intelligence and Robotic Control*, 2004.
- [17] I. Fantoni et R. Lozano. *Non linear control for underactuated mechanical systems*. Springer, 2002.
- [18] J. Imura, K. Kobayashi et T. Yoshikawa. Nonholonomic control of a 3 link planar manipulator with free joint. In *Proceedings of the 35th IEEE Conference on Decision and Control*, pages 1435–1436, 1996.

- [19] A. Isidori. *"Nonlinear control systems"*. Springer, 1995.
- [20] W. Khalil et E. Dombre. *Modélisation, identification et commande des robots*. Hermès Science, 1999.
- [21] J. Laumond. *La Robotique Mobile*. Hermes Science, 2001.
- [22] A. Lazinica. *Particle Swarm Optimization*. In-Tech, 2009.
- [23] H. Nijmeijer et A. van der Schaft. *Nonlinear Dynamical Control Systems*. Springer-Verlag, 1990.
- [24] J. Nocedal et S. J. Wright. *Numerical Optimization*. Springer, 2006.
- [25] R. Olfati Saber. *"Nonlinear Control of Underactuated Mechanical Systems with Application to Robotics and Aerospace Vehicles"*. PhD thesis, Massachusetts Institute of Technology, Departement Electrical Engineering and Computer Science, 2001.
- [26] G. Oriolo et Y. Nakamura. "Control of mechanical systems with second order nonholonomic constraints : Underactuated manipulators". In *IEEE Conference on Decision and Control*, pages 2398 – 2403, 1991.
- [27] B. Siciliano, L. Sciavicco, L. Villani, et G. Oriolo. *Robotics : Modelling, Planning and Control*. Springer, 2009.
- [28] M. W. Spong. *Underactuated mechanical systems*. Springer-Verlag, 1997.
- [29] M. Haddad et S. Hanchi T. Chettibi, H. E. Lehtihet. "Minimum cost trajectory planning for industrial robots". *European Journal of Mechanics A/Solids*, 23 :703–715, 2004.
- [30] Xin-She Yang. *"Engineering Optimization An Introduction with Metaheuristic Applications"*. Wiley, 2010.