

NOTES DE COURS

Modélisation, analyse et commande des systèmes robotisés

Préparé par

Pr. Alexandre GIRARD

Département de Génie Mécanique



Dernière mise à jour le 8 juin 2023

Table des matières

I Les robots manipulateurs	4
1 Introduction à la science des robots manipulateurs	5
1.1 L'importance de l'algèbre linéaire pour la robotique	6
1.2 Organisation des notes de cours (partie I)	7
1.3 Lexique	8
1.4 Symboles et notations	9
2 Les robots manipulateurs : définitions	10
2.1 Mécanismes et composants	10
2.2 Configurations	13
3 Cinématique des robots manipulateurs I : La position	15
3.1 Systèmes de coordonnées	15
3.2 L'approche présentée dans ce chapitre	18
3.3 Vecteurs géométriques de positions	20
3.4 Bases vectorielles et composantes d'un vecteur position	24
3.5 Les matrices de rotations	27
3.6 Coordonnées dans un repère et transformations homogènes	39
3.7 Représentation de la pose d'un corps rigide	42
3.8 Cinématique directe d'un manipulateur	47
3.9 Cinématique inverse d'un manipulateur	60
3.10 Résumé du chapitre	64
4 Cinématique des robots manipulateurs II : Le mouvement	65
4.1 Référentiel	65
4.2 Vitesse et dérivée d'un vecteur position	68
4.3 Vitesse angulaire et dérivée d'une matrice de rotation	70
4.4 Accélération et dérivée seconde d'un vecteur position	72
4.5 Cinématique différentielle des robots manipulateurs	73
4.6 Cinématique différentielle inverse	81
4.7 Manipulabilité d'un robot manipulateur	87
4.8 Relation entre les variables d'accélération	90
4.9 Résumé du chapitre	91
5 Statique des robots manipulateurs	92
5.1 Introduction	92
5.2 Les vecteurs forces	93
5.3 Relation entre les forces aux joints et les forces à l'effecteur	93
5.4 Relation statique incluant les forces conservatrices	97
5.5 Relation de la compliance aux joints et à l'effecteur	98
5.6 Manipulabilité en force d'un robot manipulateur	101
5.7 Résumé du chapitre	103

6 Dynamique des robots manipulateurs	104
6.1 Introduction	104
6.2 Structure des équations	104
6.3 Variables, nomenclature et dimensions	105
6.4 Équation des manipulateurs	106
6.5 Conservation de l'énergie	112
6.6 Dynamique inverse	112
6.7 Systèmes sous-actionnés vs. complètement actionnés	112
6.8 Manipulabilité dynamique	113
6.9 Dérivation des équations avec la méthode de Lagrange	114
6.10 Équations dans l'espace de la tâche	115
6.11 Manipulateur en contact avec l'environnement	116
6.12 Dynamique des actionneurs et effet des ratios de réduction	118
7 Introduction à la commande des systèmes robotisés	119
7.1 Architectures	121
8 Commande des robots manipulateurs I : quasi-statique	125
8.1 Commande en vitesse de l'effecteur	126
8.2 Commande en position de l'effecteur	127
8.3 Commande en force de l'effecteur	132
8.4 Commande en impédance et en admittance	133
8.5 Commande en admittance	140
9 Commande des robots manipulateurs II : dynamique	145
9.1 Commande dé-localisée joint par joint	146
9.2 Commande avec la méthode du couple calculé	147
9.3 Commande robuste	151
9.4 Méthode du mode glissant	152
9.5 Commande adaptative	155
9.6 Commande hybride en position et force	159
9.7 Méthodes d'analyse de la stabilité	160
9.8 Commande optimale	161
10 Planification des trajectoires	163
10.1 Introduction	163
10.2 Chemin et profil temporel	164
10.3 Planification cinématique pour les robots manipulateurs	168
10.4 Algorithmes d'optimisation de trajectoires	171
10.5 Algorithmes de recherche	171
11 Commande bas-niveau des actionneurs	172
11.1 Introduction	172
11.2 Comportement des moteurs électriques	172
11.3 Comportement des vérin pneumatiques	172
11.4 Actionneur de type série-élastique	172
II Les véhicules robotisés	173
12 Modélisation des véhicules	174
12.1 Navigation holonomique	174
12.2 Navigation sur un graphe discret	174
12.3 Automobiles	174
12.4 Robot mobiles	178

12.5 Drones	179
12.6 Avions	180
12.7 Fusée	181
13 Navigation	182
14 Planification	183
III Boîte à outils mathématique	184
15 Méthodes de modélisation	185
15.1 Types de modèles mathématiques	185
16 Programmation dynamique : La science des prises de décisions optimales	187
16.1 Introduction	187
16.2 Problèmes déterministes discrets	194
16.3 Problèmes stochastique	195
16.4 Robustesse et approche minimax	195
16.5 Optimisation sur un horizon de temps infini	196
16.6 Solution LQR	198
16.7 Apprentissage par renforcement	199
16.8 Formulation en temps continu	200
17 Calcul vectoriel	201
17.1 Les Vecteurs	201
17.2 Les bases vectorielles	201
17.3 Opérations vectorielles avec les vecteur-colonnes	203
17.4 Summary of vector operations in terms of components	206
17.5 Column-vector and matrix differentiation	207
18 Algèbre linéaire	211
18.1 Opérations matricielles	211
18.2 Les quatre espaces fondamentaux	216
18.3 Moindres carrés	225
18.4 Pseudo-Inverse	228
18.5 Déterminants	230
18.6 Vecteurs et valeurs propres	231
19 Optimisation	236
19.1 Descente du gradient	236
19.2 Les multiplicateurs de Lagrange	236
20 Trigonométrie	237

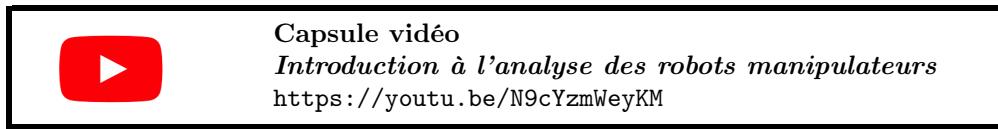
Première partie

Les robots manipulateurs

Chapitre 1

Introduction à la science des robots manipulateurs

La première partie de ces notes discute de la science pour modéliser et analyser le mouvement des robots manipulateurs. Dans un contexte d'ingénierie, modéliser un système robotique est essentiel lors de la phase de conception, par exemple pour valider qu'une géométrie de bras permet d'atteindre plusieurs positions désirées avec son effecteur (i.e. l'outil au bout du bras), et aussi lors de la programmation pour coordonner les différents moteurs et articulations d'un robot. Les outils présentés dans ces notes sont pertinents pour les mécanismes avec plusieurs articulations, on parlera toutefois de *robot* pour alléger le texte.



Comme illustré à la figure 1.1, la modélisation des robots peut être séparée en quatre grandes familles d'analyse.

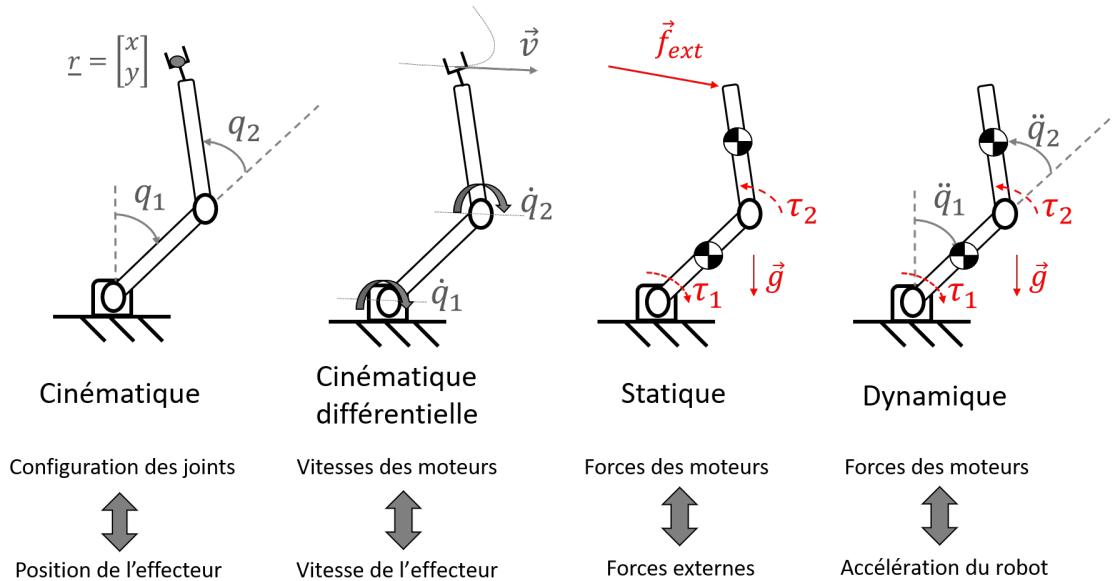


FIGURE 1.1 – Quatre grands domaines de base de l'analyse de robots manipulateurs

Les méthodes de **cinématique directe** visent à calculer la position finale de l'effecteur d'un robot en fonction du positionnement des articulations. Lorsqu'on parle de **cinématique inverse**, l'objectif est de déterminer le positionnement des articulations qui mène à une position désirée de l'effecteur du robot. Mathématiquement, l'analyse consiste à construire et solutionner la fonction non-linéaire qui relie les coordonnées de l'effecteur, groupées dans un vecteur-colonne \underline{r} , aux coordonnées des joints, groupées dans un vecteur-colonne \underline{q} :

$$\text{Cinématique directe : } \underline{r} = f(\underline{q}) \quad \text{inverse : } \underline{q} = f^{-1}(\underline{r}) \quad (1.1)$$

Un des grands défis est que la fonction inverse $f^{-1}(\underline{r})$ peut avoir plusieurs ou aucune solution, selon les coordonnées de l'effecteur qui sont visées.

Le domaine appelé la **cinématique différentielle** vise à calculer la vitesse de l'effecteur du robot en fonction de la vitesse des moteurs qui actionnent les articulations, ou vice-versa. Mathématiquement, l'analyse consiste à construire la matrice Jacobienne $J(\underline{q})$, qui permet de relier le vecteur-colonne $\dot{\underline{r}}$ des vitesses de l'effecteur (la dérivée temporelle des coordonnées \underline{r}) et le vecteur-colonne $\dot{\underline{q}}$ des vitesses des joints (la dérivée temporelle des coordonnées \underline{q}) :

$$\text{Cinématique différentielle directe : } \dot{\underline{r}} = J(\underline{q}) \dot{\underline{q}} \quad \text{inverse : } \dot{\underline{q}} = J^{\#}(\underline{q}) \dot{\underline{r}} \quad (1.2)$$

Un des défis est que pour certaines configurations du manipulateur la matrice $J(\underline{q})$ est singulière, i.e. son inverse J^{-1} n'existe pas. On dit alors que le robot est sur une singularité cinématique où il est impossible de faire bouger l'effecteur dans certaines directions. Un autre défis est que lorsque le nombre de joints est différent du nombre de coordonnées de l'effecteur, la matrice J est rectangulaire et on doit alors utiliser la notion de pseudo-inverse notée $J^{\#}$.

Pour les robots manipulateurs, les analyses de **statique** visent généralement à déterminer les forces/couples nécessaires aux moteurs/actionneurs d'un robot pour le maintenir en place en fonction des forces externes, ou vice-versa. Mathématique, l'analyse peut aussi utiliser la matrice Jacobienne pour faire la relation entre le vecteur-colonne de couples des moteurs $\underline{\tau}$ et le vecteur-colonne des forces externes \underline{f}_{ext} :

$$\text{Statique : } \underline{\tau} = J^T(\underline{q}) \underline{f}_E \quad (1.3)$$

Finalement, la **dynamique** est le domaine qui vise à déterminer les équations différentielles qui représentent la relation entre l'accélération des joints d'un robot et les forces appliquées. Mathématiquement, l'analyse consiste à construire et analyser une équation différentielle non-linéaire (reliant les coordonnées des joints \underline{q} , la vitesse des joints $\dot{\underline{q}}$, l'accélération des joints $\ddot{\underline{q}}$, les couples appliqués $\underline{\tau}$ et les forces externes \underline{f}_{ext}) qui prend la forme :

$$\text{Dynamique : } H(\underline{q})\ddot{\underline{q}} + C(\underline{q}, \dot{\underline{q}})\dot{\underline{q}} + D\underline{\dot{q}} + \underline{g}(\underline{q}) = \underline{\tau} - J^T(\underline{q}) \underline{f}_E \quad (1.4)$$

où les termes $H(\underline{q})\ddot{\underline{q}}$ et $C(\underline{q}, \dot{\underline{q}})\dot{\underline{q}}$ représentent des effets inertIELS, le terme $D\underline{\dot{q}}$ représente des forces dissipatives (ex. : friction) et le terme $\underline{g}(\underline{q})$ représente des forces gravitationnelles. Cette équation c'est $\vec{F} = m\vec{a}$ appliquée aux robots manipulateurs.

Hypothèses de travail : L'approche de modélisation pour la cinématique, la statique et la dynamique utilisée dans cette partie (chapitres 3, 4, 5 et 6) considère les robots comme des systèmes de corps rigides reliés par des articulations qui permettent un nombre restreint de mouvements relatifs. Cette approche de modélisation fait l'hypothèse que **les déformations des sections rigides du robot sont négligeables**, ce qui est par exemple raisonnable pour les robots manipulateurs industriels dans la plupart des applications.

1.1 L'importance de l'algèbre linéaire pour la robotique

Contrairement à plusieurs machines et systèmes, les robots sont caractérisés par la présence de plusieurs dimensions. Les entrées sont multiples, par exemple les forces/vitesses des multiples moteurs, et les sorties

aussi, par exemple la position (x, y, z) de l'outil. Les quantités intéressantes en robotique (entrées et sorties des analyses) peuvent donc être représentées par des vecteur-colonnes, et les relations entre ceux-ci peuvent être représentées par des opérations matricielles qui simplifient grandement les analyses. L'algèbre linéaire est donc un des outils mathématiques les plus importants pour les roboticiens.

1.2 Organisation des notes de cours (partie I)

Le chapitre 2 introduit la nomenclature et les concepts de base impliqués dans l'analyse et la modélisation des robots manipulateurs.

Le chapitre 3 présente des méthodes mathématiques pour modéliser la cinématique des robots manipulateurs. Premièrement, les systèmes de coordonnées importants pour les robots manipulateurs sont introduits. Ensuite, l'utilisation des vecteurs géométriques, des bases vectorielles et des repères pour représenter la position est présentée. Finalement, ces notions sont appliquées à la résolution des problèmes de cinématique pour les robots manipulateurs.

Le chapitre 4 présente des méthodes pour analyser la cinématique différentielle des robots manipulateurs.

Le chapitre 5 présente des méthodes pour analyser la statique des robots manipulateurs.

Le chapitre 6 présente des méthodes pour analyser la dynamique des robots manipulateurs.

1.3 Lexique

Terme technique	En anglais	Définition
Robot Manipulateur	Manipulator Robot	Robot avec une base fixe qui a comme tâche de positionner dans l'espace un objet ou un outil.
Actionneur	Actuator	Dispositif qui transforme l'énergie en travail mécanique. Typiquement des moteurs électriques et des vérins pneumatiques ou hydrauliques pour les robots.
Effecteur	End-effector	L'endroit où est situé l'objet ou l'outil manipulé par un robot.
Corps rigide	Rigid body	Un modèle idéal d'un objet qui ne se déforme pas même lorsque soumis à des forces externes.
Joint	Joint	Jonction mécanique permettant un mouvement relatif entre deux pièces.
Système de coordonnées	Coordinate system	Ensemble de scalaires (angles ou positions) qui permettent de paramétriser la position/configuration d'un système.
Degrés-de-Liberté (DDL)	Degree-of-Freedom (DoF)	Le nombre de variable indépendante qui permettent de paramétriser dans l'espace la position/configuration d'un système.
Base vectorielle	Vector basis	Trois vecteurs unitaires qui définissent une orientation.
Base vectorielle orthonormée	Orthogonal Vector basis	Trois vecteurs unitaires orthogonaux qui définissent une orientation.
Origine	Origin	Un point de référence pour la mesure des positions.
Repère	Frame	Combinaison d'une origine et d'une base vectorielle orthonormée.
Système de coordonnées cartésiennes	Cartesian coordinate system	Ensemble de trois scalaires qui déterminent la position d'un point par rapport à une origine et trois axes orthogonaux.
Référentiel	Reference frame	Point de vue choisi pour analyser/observer un mouvement.

1.4 Symboles et notations

Symbole	Définition
Vecteurs géométriques	
\vec{v}	Vecteur
\hat{a}	Vecteur unitaire
$\vec{r}_{A/B}$	Vecteur position du point A par rapport au point B
Vecteur-colonnes et composantes scalaires	
$\underline{c} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = [c_1 \ c_2 \ c_3]^T$	Vecteur-colonne
$\underline{v}^a = \begin{bmatrix} v_1^a \\ v_2^a \\ v_3^a \end{bmatrix}$	Vecteur-colonne des composantes du vecteur \vec{v} dans la base vectorielle a
v_i^a	Composante du vecteur \vec{v} selon le vecteur unitaire i de la base vectorielle a
$\underline{q} = \begin{bmatrix} q_1 \\ \vdots \\ q_n \end{bmatrix}$	Vecteur-colonne qui regroupe n scalaires représentant la configuration d'un robot à n DDL dans l'espace des joints.
Bases et repères	
$\{\hat{a}_1, \hat{a}_2, \hat{a}_3\}$	Base vectorielle a définie par un ensemble de trois vecteur unitaires orthogonaux
$\{A_O, \hat{a}_1, \hat{a}_2, \hat{a}_3\}$	Repère A défini par un point d'origine A_O et la base vectorielle a
Matrices	
$M = \begin{bmatrix} M_{1,1} & \dots & M_{1,n} \\ \vdots & \ddots & \vdots \\ M_{m,1} & \dots & M_{m,n} \end{bmatrix}$	Matrice de m rangées et n colonnes ($m \times n$)
${}^a R^b = \begin{bmatrix} b_1^a & b_2^a & b_3^a \end{bmatrix}$	Matrice de rotation (3×3) qui représente l'orientation de la base vectorielle b par rapport à la base vectorielle a .
${}^A T^B = \begin{bmatrix} {}^a R^b & \underline{r}_{B_o/A_o}^a \\ 0 \ 0 \ 0 & 1 \end{bmatrix}$	Matrice de transformation homogène (4×4) du repère B vers le repère A

Chapitre 2

Les robots manipulateurs : définitions

Les robots manipulateurs correspondent à des bras artificiels, généralement plusieurs liens rigides connectés par des articulations. La grande majorité des robots industriels sont des robots manipulateurs qui ont comme tâche de positionner des outils pour la soudure, la peinture, etc. La figure 2.1 présente les composants principaux d'un robot manipulateur. La figure 2.3 illustre des exemples concrets de mécanismes et composants pour un robot manipulateur prototype.

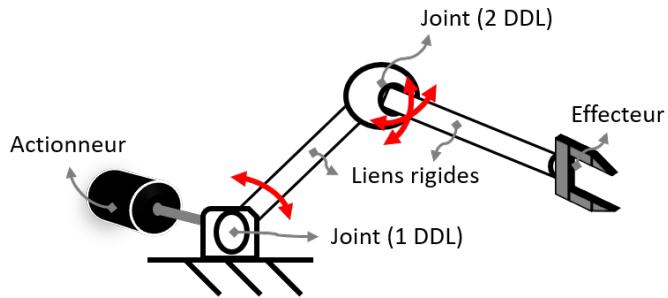


FIGURE 2.1 – Nomenclature pour l'analyse d'un robot manipulateurs

2.1 Mécanismes et composants

Cette section présente les différents composants d'un robot manipulateur ainsi que la nomenclature utilisée.

2.1.1 Liens rigides

Un lien rigide d'un robot, c'est un ensemble de pièces fixes les unes par rapport aux autres qui forment un corps rigide. D'un point de vu de cinématique, deux pièces fixées de telle façon qu'aucun mouvement relatif n'est possible vont être considérées comme un seul lien. Par exemple, plusieurs tubes qui seraient soudés ensembles seraient considérés comme un seul lien rigide.

2.1.2 Joints (articulations)

Les joints sont les articulations d'un robot. Un **joint prismatique** (articulation linéaire), permet la translation selon un axe entre deux pièces, et contraint les rotations relatives. Un **joint rotatif** (articulation angulaire), permet à deux pièces de pivoter relativement selon un axe. La variable q_i utilisée pour décrire la

configuration d'un joint i est une distance pour un joint prismatique et un angle pour un joint rotatif.

$$\text{Joint prismatique : } q_i = x_i \quad [m] \quad (2.1)$$

$$\text{Joint rotatif : } q_i = \theta_i \quad [rad] \quad (2.2)$$

Les joints prismatiques et révolus ont une configuration décrite avec une seule variable, chacun des joints de ce type produit donc un degré de liberté (DDL) pour le robot manipulateur. Un robot utilisant seulement ces types de joints a donc un nombre total de DDL qui est égale au nombre de joints. Toutefois, des mécanismes plus complexes (ex. : joint sphérique comme illustré à la figure 2.1) peuvent produire plusieurs DDL en une seule articulation. Par contre, contrairement au corps humain, les robots utilisent généralement des joints prismatiques et rotatifs seulement pour simplifier la mécanique. Un joint sera dit actif si sa configuration est contrôlée par un actionneur, et passif si sa configuration est libre.

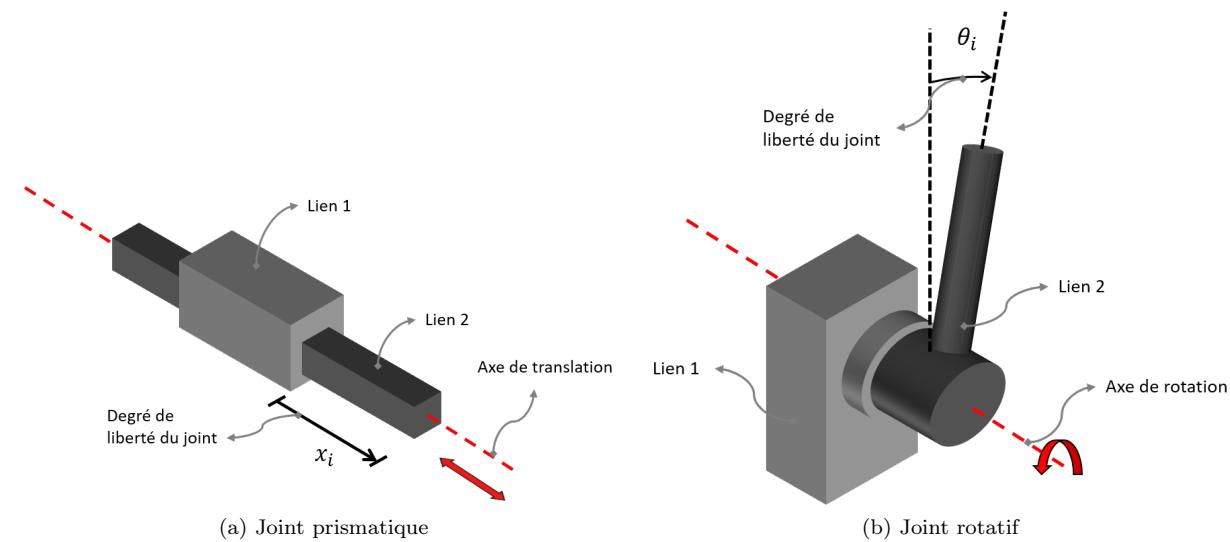


FIGURE 2.2 – Joints à un degré de liberté

2.1.3 Effecteur

L'effecteur est un terme générique qui référence à la position de l'outil d'un robot manipulateur (ex. : une pince, une tête de soudure, une caméra, etc.). Dans un espace tri-dimensionnel, la position d'un objet nécessite au plus six variables pour être complètement décrite (voir section 3.7). L'effecteur d'un robot aura donc au plus 6 DDL.

2.1.4 Actionneurs

Les actionneurs sont les muscles des bras robotiques. Un actionneur est fondamentalement un dispositif qui transforme de l'énergie en travail mécanique. La plupart des robots industriels utilisent des moteurs électriques pour actionner leurs joints, toutefois les actionneurs pourraient aussi être des vérins pneumatiques ou hydrauliques.

2.1.5 Capteurs

Les capteurs sont les dispositifs qui collectent de l'information sur l'état interne d'un robot et/ou l'environnement. Pour la cinématique, les capteurs très régulièrement utilisés sont les encodeurs, qui mesurent directement la position des joints (variables q_i), et les systèmes de visions, qui mesurent la position de points dans le repère de la caméra.

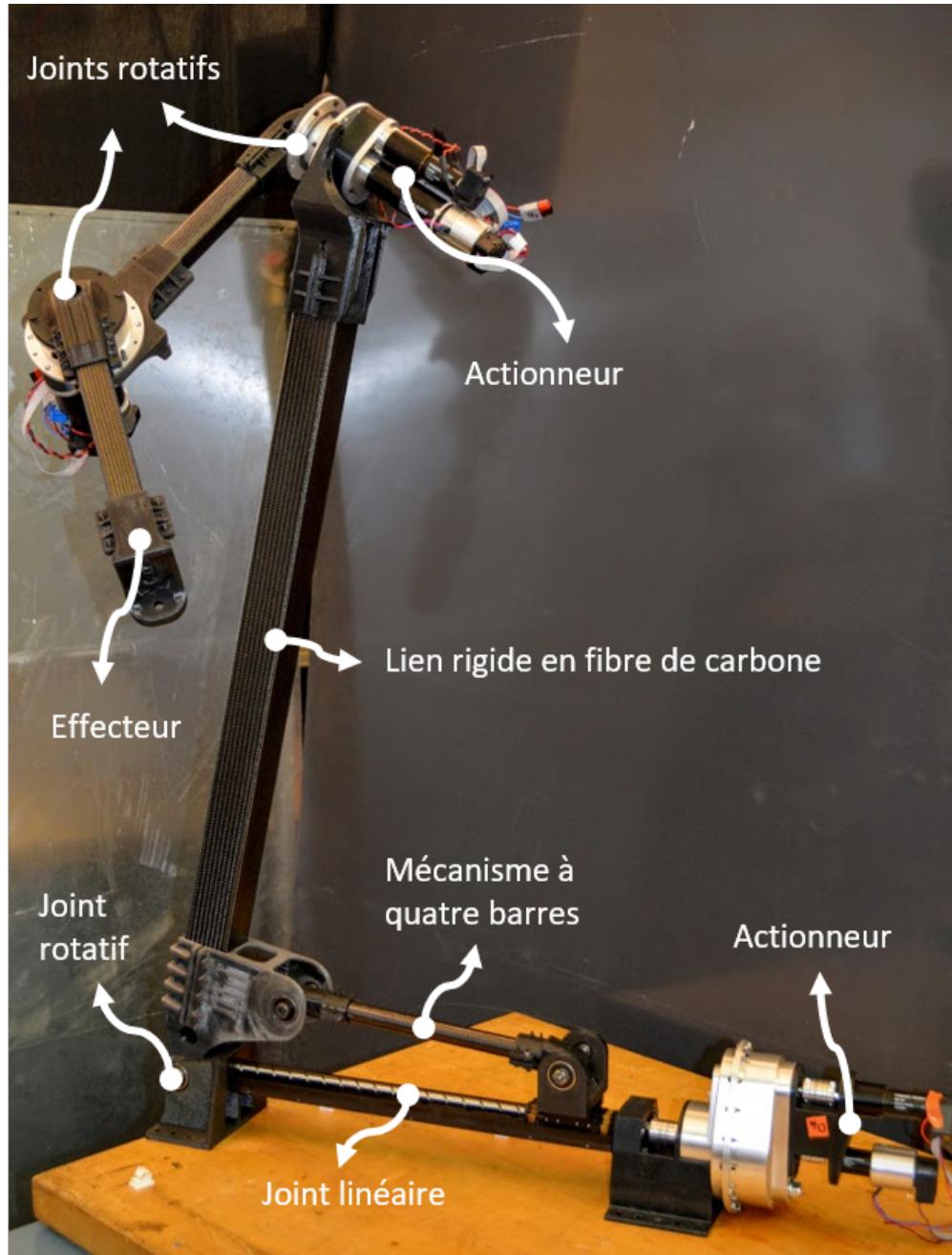


FIGURE 2.3 – Prototype de robot manipulateur avec trois degrés de liberté

2.2 Configurations

Cette section introduit les notions en lien avec la configuration d'un robot manipulateur.

2.2.1 Degrés de liberté

Le nombre de degrés de liberté (DDL), c'est le nombre de variables nécessaires pour complètement décrire la configuration d'un robot ou un mécanisme. La plupart des robots industriels ont six DDL, ce qui est suffisant pour pouvoir contrôler indépendamment les six DDL de l'effecteur. Pour des robots qui utilisent seulement des joints rotatifs comme illustré à la figure 2.4, le nombre de DDL est égale au nombre de joints. Si le nombre de DDL est supérieur à six, le robot est dit **redondant**. Un robot redondant a plusieurs options de configuration des joints pour atteindre une position et une orientation désirées de l'effecteur.

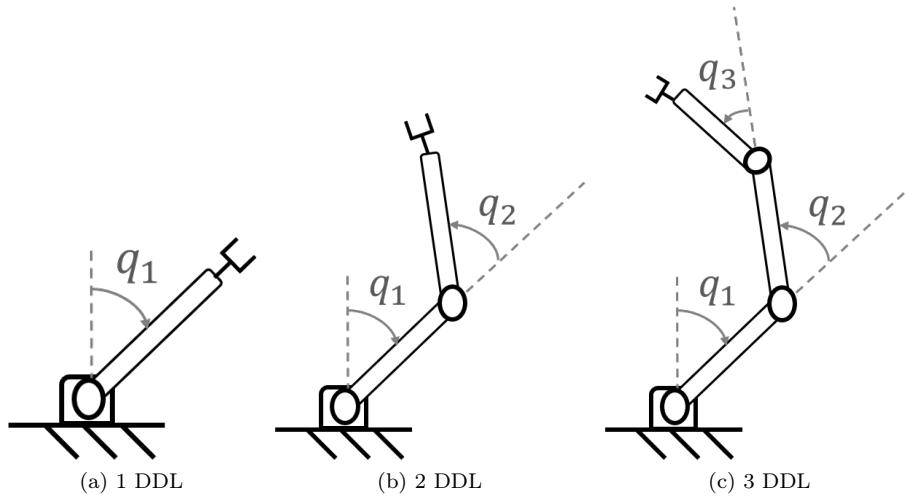


FIGURE 2.4 – Manipulateurs planaires avec des joints rotatifs

2.2.2 Espace de travail

L'espace de travail c'est le volume qui comprend toutes les positions atteignables par l'effecteur du robot. La figure 2.5 illustre un espace de travail dans le plan d'un robot à deux joints. Parfois, des sous-ensembles de l'espace de travail peuvent être définis en fonction des positions atteignables avec une orientation précise de l'effecteur.

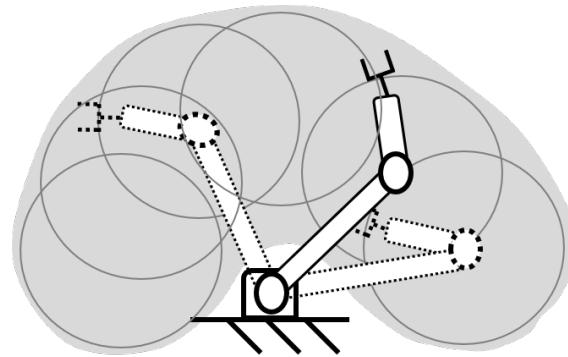


FIGURE 2.5 – Exemple de l'espace de travail d'un robot manipulateur à deux joints

2.2.3 Configurations classiques

À venir !

Chapitre 3

Cinématique des robots manipulateurs I : La position

Ce chapitre présente des méthodes pour modéliser et analyser la position de systèmes à plusieurs degrés de liberté, donc bien adaptés aux robots manipulateurs. Dans un contexte de robotique, les problèmes de cinématique consistent premièrement à bien choisir et définir des systèmes de coordonnées pertinents. Par exemple, des coordonnées généralisées pour décrire la configuration des joints, des coordonnées cartésiennes pour décrire la tâche d'un robot en terme de trajectoire de l'effecteur, des coordonnées sphériques pour décrire les mesures d'un système de vision, etc. Ensuite, le défi est de **calculer les fonctions de transformations pour passer d'un système de coordonnées à un autre**. Par exemple, pour les robots manipulateurs, le principal défi est le calcul de la fonction de la cinématique directe et son inverse, i.e. une fonction avec comme entrées les coordonnées généralisées qui représentent la configuration des joints et comme sortie la pose de l'effecteur. Le calcul de ces transformations dans un contexte de robotique est le sujet principal de ce chapitre. La section 3.1 introduit aux différents systèmes de coordonnées. Les sections 3.3 et 3.4 introduisent les notions de vecteurs de position géométriques, bases vectorielles et vecteur-colonnes de composantes. Les sections 3.5 et 3.6 présentent les changements de base et les changements des repères. Ensuite, la section 3.7 présente les méthodes pour représenter mathématiquement la pose d'un corps rigide. Finalement, les sections 3.8 et 3.9 traitent de la cinématique des robots manipulateurs, i.e. le calcul des transformations pour passer de l'espace des joints à la pose de l'effecteur et vice-versa.

3.1 Systèmes de coordonnées

Un système de coordonnées c'est un ensemble de scalaires, généralement des distances et des angles, qui décrivent la position d'un système.

3.1.1 Position d'une particule

Une particule (un point) dans l'espace tri-dimensionnel possède trois DDL et sa position peut donc être complètement déterminée par trois scalaires qui forment un système de coordonnées. Trois types de systèmes de coordonnées sont généralement utilisés pour décrire la position d'une particule par rapport à des axes : **cartésien** (x, y, z), **cylindrique** (r, θ, z) ou **sphérique** (ρ, θ, ϕ), voir la figure 3.1.

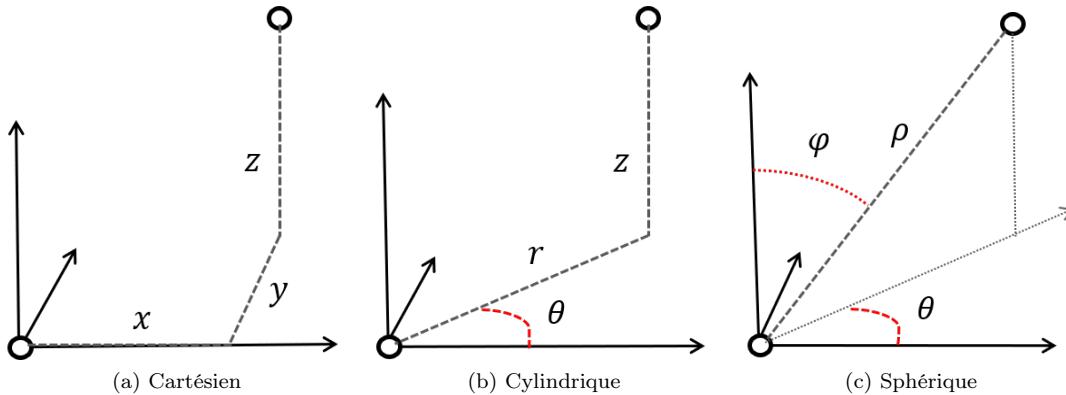


FIGURE 3.1 – Trois systèmes de coordonnées standards pour représenter la position d'un point en 3D

Selon la situation, l'utilisation d'un système de coordonnées particulier peut simplifier l'analyse d'un problème. Par exemple, pour localiser un point géographique sur la Terre, des coordonnées sphériques (ex. : longitude et latitude) sont plus pratiques que des coordonnées cartésiennes (x,y,z) par rapport au centre de la Terre. Parfois l'utilisation d'un type de système de coordonnées est nécessaire car un capteur mesure une coordonnée particulière. Par exemple, le *LIDAR* sur une voiture autonome mesure des distances dans plusieurs directions grâce à un laser pivotant, les données brutes sont des coordonnées sphériques. Il est donc parfois inévitable de faire la conversion d'un système de coordonnées à un autre. Le tableau 3.1 donne les conversions entre des coordonnées cartésiennes, cylindriques et sphériques, qui sont basées sur la même origine et le même système d'axes.

3.1.2 Pose d'un corps rigide

Pour décrire la position d'un corps rigide, trois scalaires qui représentent l'orientation sont nécessaires en plus des trois coordonnées qui spécifient la translation, pour un total de six DDL. La combinaison de la translation et de l'orientation est souvent appelée **la pose** d'un objet. La translation est typiquement spécifiée par la position d'un point sur le corps rigide (ex. : son centre) avec les différentes représentations discutées à la section 3.1.1 (cartésienne, cylindrique ou sphérique) et l'orientation est typiquement spécifiée par trois angles. Comme illustré à la figure 3.2, les six coordonnées suivantes pourraient être utilisées pour spécifier la pose d'un objet :

$$\text{La translation (3 DDL), ex. : } [x, y, z] \quad (3.1)$$

$$\text{L'orientation (3 DDL), ex. : } [\theta_{tangage}, \theta_{roulis}, \theta_{lacet}] \quad (3.2)$$

Il est à noter que **la représentation de l'orientation d'un corps rigide est un problème complexe** et qu'il existe plusieurs méthodes alternatives pour représenter l'orientation (matrices de rotations, angles de Euler, représentation axe-angle, quaternions, etc.) qui seront discutées à la section 3.7. Selon les domaines (robotique, aérospatiale, engins graphiques des jeux vidéos, vision par ordinateur, astronomie, etc.), différents standards et conventions existent en terme de coordonnées utilisées pour décrire l'orientation. Le point à retenir pour le moment est que trois coordonnées indépendantes sont nécessaires pour représenter une orientation arbitraire, donc six totales sont nécessaires pour représenter une pose arbitraire. Par exemple, un minimum de six coordonnées doivent être utilisées pour : spécifier la position de l'effecteur d'un robot, décrire la position d'un avion dans le ciel, décrire la position de la terre par rapport au soleil, etc.

3.1.3 Configuration d'un robot manipulateur

Pour représenter la configuration d'un robot manipulateur il faut suffisamment de variables pour décrire la position relative de chacun des liens rigides qui le compose. Le choix des coordonnées doit être adapté à la géométrie du robot étudié. Un ensemble de variables indépendantes, généralement des angles et des distances, qui permet de déterminer la configuration d'un robot est appelé **coordonnées généralisées**. L'adjectif

	De cartésien	De cylindrique	De sphérique
Vers cartésien	$x = x$ $y = y$ $z = z$	$x = r \cos \theta$ $y = r \sin \theta$ $z = z$	$x = \rho \sin \phi \cos \theta$ $y = \rho \sin \phi \sin \theta$ $z = \rho \cos \phi$
Vers cylindrique	$r = \sqrt{x^2 + y^2}$ $\theta = \arctan \frac{y}{x}$ $z = z$	$r = r$ $\theta = \theta$ $z = z$	$r = \rho \sin \phi$ $\theta = \theta$ $z = \rho \cos \phi$
Vers sphérique	$\rho = \sqrt{x^2 + y^2 + z^2}$ $\theta = \arctan \frac{y}{x}$ $\phi = \arctan \frac{\sqrt{x^2 + y^2}}{z}$	$\rho = \sqrt{r^2 + z^2}$ $\theta = \theta$ $\phi = \arctan \frac{r}{z}$	$\rho = \rho$ $\theta = \theta$ $\phi = \phi$

TABLE 3.1 – Conversions entre les systèmes de coordonnées cartésiennes, polaires et sphériques qui font référence aux même axes et à la même origine.

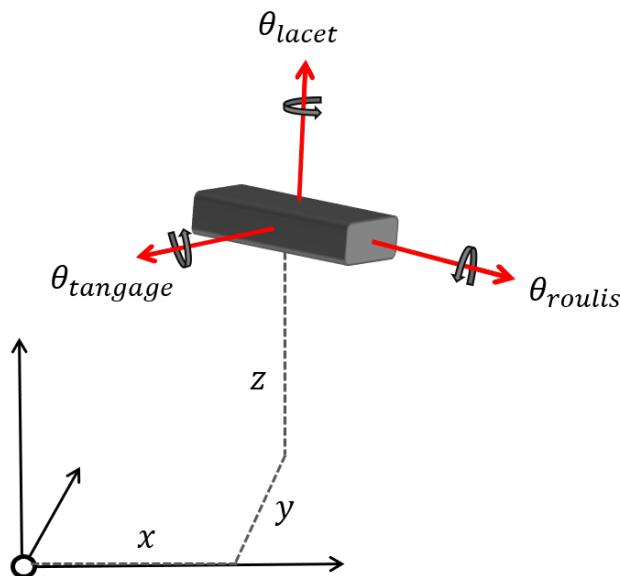


FIGURE 3.2 – Six coordonnées sont nécessaire pour représenter la pose d'un corps rigide, trois pour décrire la translation et trois pour décrire son orientation.

généralisées est historique et a ici le sens de *pas nécessairement cartésiennes*. Le nombre de coordonnées généralisées nécessaire correspond au nombre de **degrés-de-liberté** (DDL) du robot. La figure 3.3 montre trois exemples de deux coordonnées indépendantes qui peuvent être utilisées comme coordonnées généralisées pour décrire la configuration d'un robot à deux joints rotatifs.

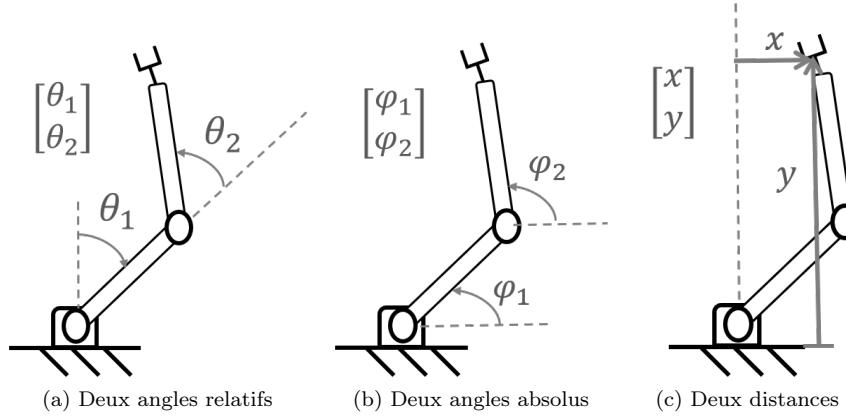


FIGURE 3.3 – Trois possibilités de coordonnées indépendantes pour spécifier la position d'un robot à 2 DDL

Deux types de systèmes de coordonnées sont généralement utilisés en robotique :

Espace des joints/actionneurs Pour les robots manipulateurs, les algorithmes de contrôle utilisent généralement des coordonnées généralisées qui correspondent à des mesures du déplacement relatif de chaque joint comme illustré à la figure 3.4a. Ces coordonnées sont généralement notées $q = [q_1 \ q_2 \ \dots \ q_n]^T$ (où n est le nombre de DDL) et on réfère aussi à ce système comme **l'espace des joints**. Chaque coordonnée q_i représente alors une mesure locale du DDL qui relie deux liens rigides séquentiels. Pour les robots industriels par exemple, ce système de coordonnées est très utile car des encodeurs mesurent directement ces variables et les moteurs les contrôlent de façon indépendante.

Espace de la tâche La tâche d'un robot peut rarement être spécifiée facilement dans l'espace des joints. Un système de coordonnées naturel pour la tâche doit donc généralement être utilisé et on réfère aussi à ce système comme **l'espace de la tâche**. La tâche d'un robot manipulateur est généralement spécifiée plus naturellement en termes de coordonnées cartésiennes de l'effecteur. Il est à noter que le nombre de coordonnées nécessaires pour l'espace de la tâche peut être différent du nombre de DDL total du robot. Comme illustré à la figure 3.4b, pour un robot qui aurait comme tâche d'écrire sur une plaque, l'objectif serait naturellement exprimé par une trajectoire définie avec des coordonnées (x, y) sur la plaque.

3.2 L'approche présentée dans ce chapitre

L'approche de modélisation mise de l'avant dans ce chapitre, et illustrée à la Figure 3.5, ce résume à la séquence :

1. Décrire le problème avec des vecteurs géométriques symboliques ;
2. Choisir des bases appropriées ;
3. Traduire la relation vectorielle en relation matricielle (projection dans une base) ;
4. Déterminer la solution numérique avec les outils de l'algèbre linéaire.

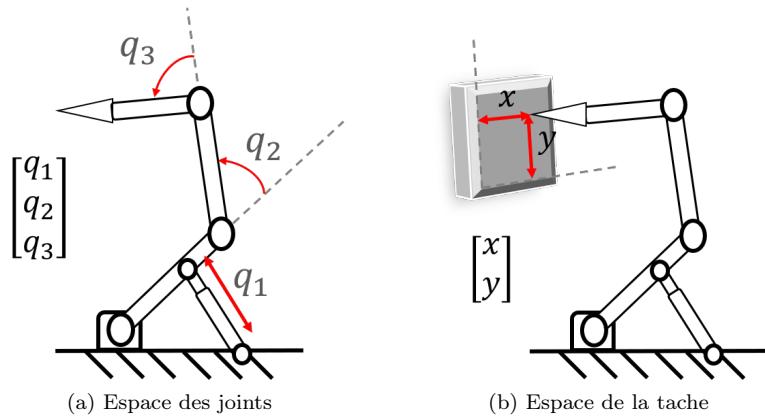


FIGURE 3.4 – Exemple de coordonnées représentant l'espace des joints et l'espace tâche d'un robot manipulateur

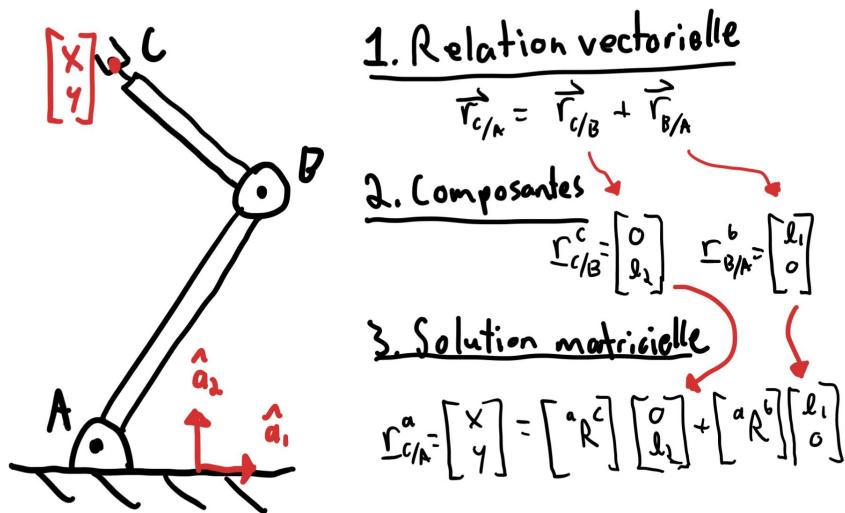
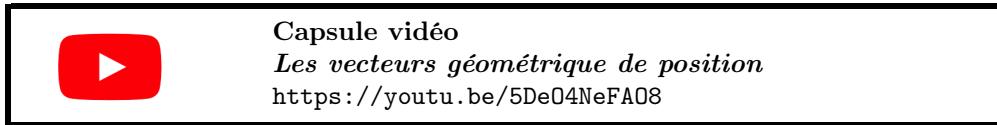


FIGURE 3.5 – Aperçu des grandes lignes de l'approche de modélisation présentée dans ces notes

3.3 Vecteurs géométriques de positions

La notion de vecteur géométrique permet de travailler avec des **vecteurs de position** qui sont indépendants d'un choix de système de coordonnées. Pour les problèmes de cinématique de robots, il est particulièrement adapté de travailler d'abord avec les vecteurs géométrique, plutôt que directement avec des coordonnées. Cette approche 1) simplifie les calculs en 3D et 2) permet de faire le saut facilement entre plusieurs systèmes de coordonnées. Un vecteur géométrique est une quantité définie dans l'espace et possédant une grandeur et une direction. Dans ces notes, les vecteurs géométriques, notés \vec{v} , seront explicitement distingués des vecteur-colonnes (groupe de plusieurs scalaires) qui seront notés $\underline{v} = [v_1 \ v_2 \ v_3]^T$, voir chapitre 17.



Un vecteur position nécessite deux points pour être défini, une origine et une destination. Notez que c'est particulier des vecteurs positions, la plupart des autres vecteurs utilisés en physique (vitesse, accélération, force, champ électrique, etc.) ont un point d'application mais sont indépendants d'un choix d'origine. Dans ces notes, les vecteurs positions seront notés :

$$\vec{r}_{B/A} = \text{Vecteur position du point } B \text{ par rapport au point } A \quad (3.3)$$

où A est le point d'origine et B le point de destination, tel qu'illustré à la figure 3.6. Parfois, la notation équivalente \vec{AB} est utilisée dans la littérature. Ces vecteurs peuvent être utilisés pour faire des constructions géométriques indépendamment d'un choix de système de coordonnées.

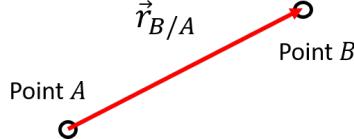


FIGURE 3.6 – Un vecteur position

3.3.1 Propriétés des vecteurs de position

La figure 3.7 illustre les propriétés de base des vecteurs de position.

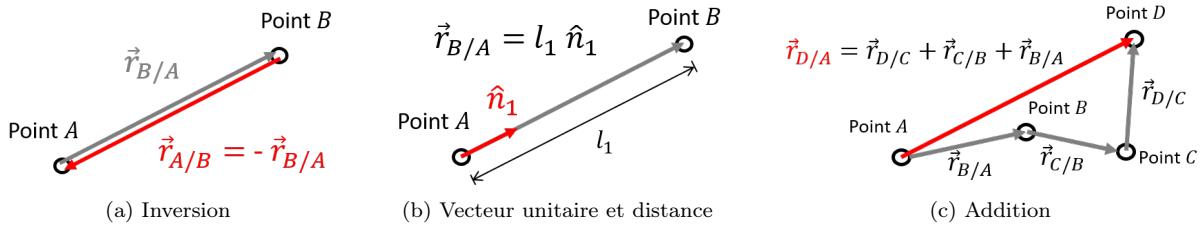


FIGURE 3.7 – Propriétés des vecteurs de position

Propriété 3.3.1 (Addition) *Un vecteur position peut être décomposé en une addition de vecteurs qui passent par des points intermédiaires :*

$$\vec{r}_{Z/A} = \vec{r}_{Z/Y} + \vec{r}_{Y/X} + \dots + \vec{r}_{C/B} + \vec{r}_{B/A} \quad (3.4)$$

Propriété 3.3.2 (Inversion) L'inversion de l'origine et la destination est équivalente à la négation du vecteur de position :

$$\vec{r}_{B/A} = -\vec{r}_{A/B} \quad (3.5)$$

Propriété 3.3.3 (Vecteurs unitaires) Les vecteurs positions peuvent être exprimés comme des **combinaisons de distances et directions** qui sont représentées respectivement par des paires de scalaires l_i et vecteurs unitaires \hat{n}_i (qui représente des directions dans l'espace) :

$$\vec{r} = l_1 \hat{n}_1 + l_2 \hat{n}_2 + \dots + l_n \hat{n}_n \quad (3.6)$$

Propriété 3.3.4 (Norme) La norme d'un vecteur position, i.e. sa longueur, peut être calculée en effectuant un produit scalaire du vecteur avec lui-même :

$$\|\vec{r}\|^2 = \vec{r} \cdot \vec{r} \quad (3.7)$$

Propriété 3.3.5 (Projection) La longueur d'un vecteur position \vec{r} projeté selon un axe particulier correspond au produit scalaire du vecteur position avec un vecteur unitaire \hat{n} aligné sur cet axe :

$$d = \vec{r} \cdot \hat{n} = \|\vec{r}\| \cos \angle(\vec{r}, \hat{n}) \quad (3.8)$$

Cette opération est aussi appelée **projection d'un vecteur sur un axe** et illustrée à la figure 3.8a.

Propriété 3.3.6 (Angle) Le produit scalaire de deux vecteurs unitaires est égale au cosinus de l'angle qui les sépare puisqu'ils ont une dimension unitaire :

$$\cos \angle(\hat{n}_1, \hat{n}_2) = \hat{n}_1 \cdot \hat{n}_2 \quad (3.9)$$

Il est donc possible d'utiliser le produit scalaire pour mesurer un angle entre deux vecteurs, comme illustré à la figure 3.8b.

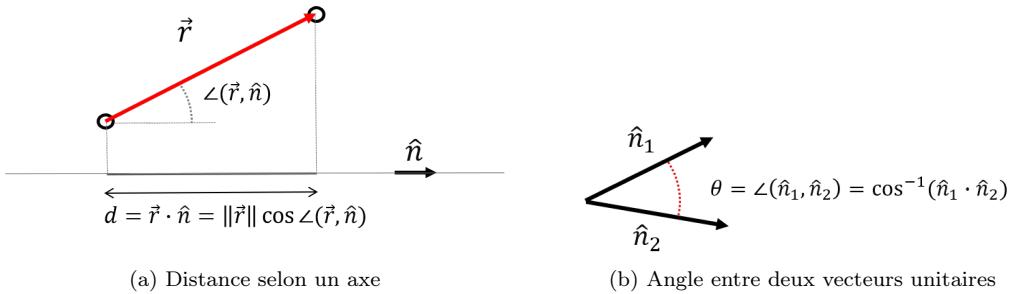


FIGURE 3.8 – Utilisation du produit scalaire pour des mesures géométriques

3.3.2 Procédure d'utilisation pour le calcul de distances

La démarche utilisée pour calculer des positions avec les vecteurs géométriques se résume par les trois étapes suivantes :

1. Par inspection, construire le vecteur position \vec{r} d'intérêt comme une addition/soustraction de plusieurs vecteurs positions \vec{r}_i avec des longueurs et orientations connues :

$$\vec{r} = \sum_i \vec{r}_i \quad (3.10)$$

2. Substituer les vecteurs positions symboliques \vec{r}_i par des variables de distance l_i et des vecteurs unitaires \hat{n}_i représentant des directions qui peuvent être mesurées :

$$\vec{r} = \sum_i l_i \hat{n}_i \quad (3.11)$$

3. Calculer les distances désirées d en effectuant un produit scalaire avec les vecteurs unitaires selon les axes désirés :

$$d_x = \vec{r} \cdot \hat{x} = \sum_i l_i (\hat{n}_i \cdot \hat{x}) = \sum_i l_i \cos \angle(\hat{n}_i, \hat{x}) \quad (3.12)$$

$$d_y = \vec{r} \cdot \hat{y} = \sum_i l_i (\hat{n}_i \cdot \hat{y}) = \sum_i l_i \cos \angle(\hat{n}_i, \hat{y}) \quad (3.13)$$

$$d_z = \vec{r} \cdot \hat{z} = \sum_i l_i (\hat{n}_i \cdot \hat{z}) = \sum_i l_i \cos \angle(\hat{n}_i, \hat{z}) \quad (3.14)$$



Capsule vidéo

Exemple de calcul avec les vecteurs géométrique de position

<https://youtu.be/YqmyBJPJmPO>

Exemple 1. Calcul de la hauteur d'un robot

Un exemple est ici donné pour le calcul de la hauteur de l'effecteur d'un robot à 3 DDL, illustré à la figure 3.9a, avec l'aide des vecteurs géométriques de position. Comme illustré à la Figure 3.9b, par inspection il est possible de déterminer la relation vectorielle suivante :

$$\vec{r}_{D/A} = \vec{r}_{D/C} + \vec{r}_{C/B} + \vec{r}_{B/A} \quad (3.15)$$

Ensuite, comme illustré à la figure 3.9c, des vecteurs unitaires \hat{n}_i alignés avec chacun des liens rigides du robot manipulateur permettent de préciser l'équation (3.15) avec des longueurs et des directions :

$$\vec{r}_{D/A} = l_1 \hat{n}_1 + l_2 \hat{n}_2 + l_3 \hat{n}_3 \quad (3.16)$$

Finalement, pour calculer la hauteur h de ce robot par exemple, il suffit de prendre le produit scalaire avec un vecteur unitaire vertical \hat{y} :

$$h = \vec{r}_{D/A} \cdot \hat{y} \quad (3.17)$$

$$h = (l_1 \hat{n}_1 + l_2 \hat{n}_2 + l_3 \hat{n}_3) \cdot \hat{y} \quad (3.18)$$

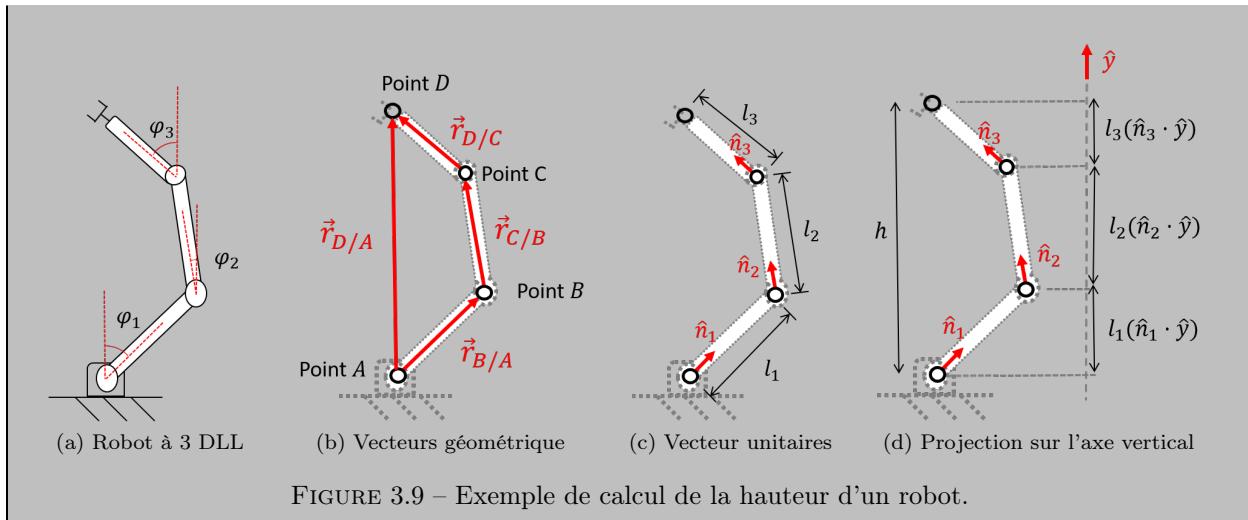
$$h = l_1 (\hat{n}_1 \cdot \hat{y}) + l_2 (\hat{n}_2 \cdot \hat{y}) + l_3 (\hat{n}_3 \cdot \hat{y}) \quad (3.19)$$

$$h = l_1 \cos \angle(\hat{n}_1, \hat{y}) + l_2 \cos \angle(\hat{n}_2, \hat{y}) + l_3 \cos \angle(\hat{n}_3, \hat{y}) \quad (3.20)$$

À noter ici que le produit scalaire de deux vecteurs unitaires correspond au cosinus de l'angle entre les deux vecteurs, voir équation (3.9). Donc avec l'exemple ci-dessus le produit scalaire $\hat{n}_i \cdot \hat{y}$ correspond simplement au cosinus de l'angle du joint i avec l'axe vertical, donné par les angles φ_i à la figure 3.9a. L'équation (3.20) peut donc être précisée comme une expression des variables de distances l_i et des variables d'angles φ_i :

$$h = l_1 \cos(\varphi_1) + l_2 \cos(\varphi_2) + l_3 \cos(\varphi_3) \quad (3.21)$$

Dans un cas simple planaire comme celui-ci il aurait été assez simple de calculer la hauteur du robot directement par trigonométrie. L'approche vectorielle est toutefois plus systématique et se généralise beaucoup plus facilement à des systèmes complexes et tri-dimensionnels comme les robots industriels.



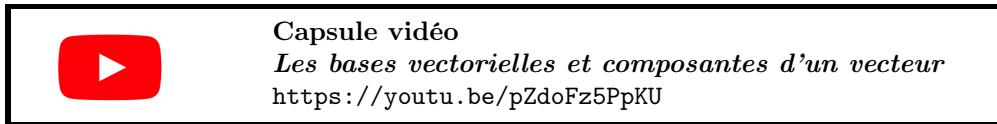
3.4 Bases vectorielles et composantes d'un vecteur position

Les vecteurs positions tri-dimensionnels \vec{r} peuvent être décrits par trois scalaires qui représentent des distances (r_1 , r_2 et r_3) qui multiplient trois vecteur unitaires orthogonaux (\hat{a}_1 , \hat{a}_2 et \hat{a}_3) qui forment une base vectorielle. Les scalaires r_i qui multiplient chaque vecteur unitaire sont appelés composantes et peuvent être regroupés sous la forme d'un vecteur-colonne, la notation suivante sera utilisée :

$$\text{Vecteur-géométrique : } \vec{r} = r_1^a \hat{a}_1 + r_2^a \hat{a}_2 + r_3^a \hat{a}_3 \quad \text{Vecteur-colonne : } \underline{r}^a = \begin{bmatrix} r_1^a \\ r_2^a \\ r_3^a \end{bmatrix} \quad (3.22)$$

où l'exposant a est utilisé pour spécifier la base vectorielle associée aux composantes scalaires.

Note sur la séquence de lecture : Il est suggéré au lecteur de lire les sections 17.1, 17.2 et 17.3 dans le chapitre 17 qui couvre le calcul vectoriel de façon plus générique. Ces sections couvrent plus en détails les fondements qui sont ici utilisés dans un contexte de cinématique.



3.4.1 Rappel sur les bases vectorielles

Une base vectorielle correspond à un ensemble de trois vecteurs unitaires (en 3D) orthogonaux qui déterminent l'orientation de trois axes dans l'espace et forment une base qui permet de décrire n'importe quel vecteur position \vec{r} sous la forme :

$$\vec{r} = r_1^a \hat{a}_1 + r_2^a \hat{a}_2 + r_3^a \hat{a}_3 \quad (3.23)$$

On appellera *base vectorielle* a , une base formée par l'ensemble des vecteurs unitaires $\{\hat{a}_1, \hat{a}_2, \hat{a}_3\}$.

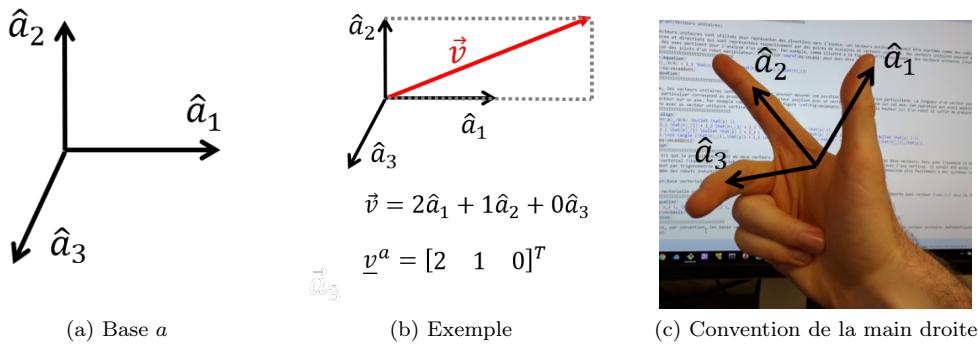


FIGURE 3.10 – Les bases vectorielles

Les bases vectorielles permettent de faire des opérations et combiner des vecteurs en les exprimant avec une base commune de vecteurs unitaires. Dans la littérature, la notation $(\hat{i}, \hat{j}, \hat{k})$ ou $(\hat{x}, \hat{y}, \hat{z})$ est parfois utilisée. Dans ces notes, on utilisera des lettres (a, b, c, \dots) pour identifier des bases, ainsi que des indices $(1, 2, 3)$ pour spécifier les trois axes (voir figure 17.1a). Cette approche simplifie la notation lorsqu'un grand nombre de bases sont utilisées simultanément, comme c'est souvent le cas en robotique.

Base vs. repère : Les bases vectorielles représentent seulement des orientations, la position d'une base est donc sans signification. Dans les dessins et schémas, on dessine une base vectorielle sur un corps rigide pour signifier que celle-ci est attachée au corps rigide et tourne avec celui-ci, toutefois la position de la base vectorielle sur le corps est arbitraire. On parlera de repère, voir section 3.6, lorsqu'un point d'origine est associé à une base vectorielle.

Comme illustré à la figure 17.2a, il est possible de calculer les composantes d'un vecteur exprimé dans une base par un produit scalaire du vecteur géométrique avec chacun des vecteurs unitaires de la base :

$$r_i^a = \vec{r} \cdot \hat{a}_i \quad \Rightarrow \quad \underline{r}^a = \begin{bmatrix} \vec{r} \cdot \hat{a}_1 \\ \vec{r} \cdot \hat{a}_2 \\ \vec{r} \cdot \hat{a}_3 \end{bmatrix} \quad (3.24)$$

Inversement, comme illustré à la figure 17.2b, le vecteur position géométrique peut être reconstruit en effectuant la somme des composantes multipliées avec leurs vecteurs unitaires respectifs :

$$\vec{r} = \sum_i r_i^a \hat{a}_i = r_1^a \hat{a}_1 + r_2^a \hat{a}_2 + r_3^a \hat{a}_3 \quad (3.25)$$

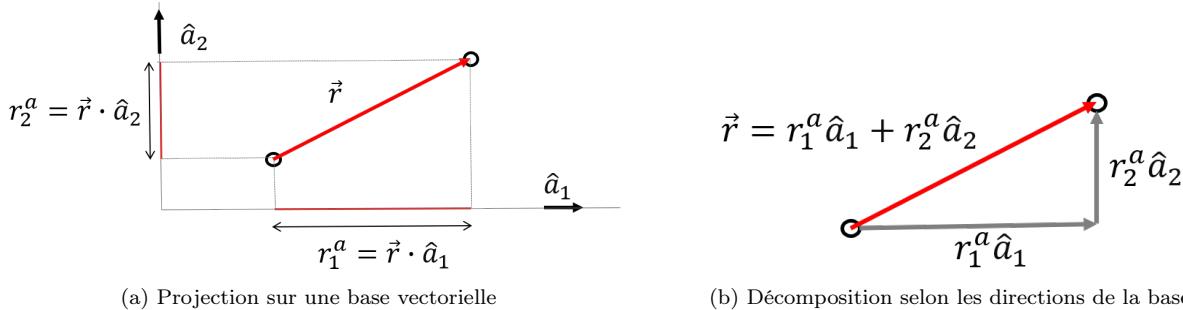


FIGURE 3.11 – Relations entre le vecteur géométrique de position et les composantes du vecteur-colonne

3.4.2 Transfert d'une équation vectorielle vers une équation matricielle

Lorsque les composantes de plusieurs vecteurs-géométrique de position sont exprimées avec la même base, il est possible de faire des opérations directement avec les vecteur-colonnes de composantes. Cela permet de traiter simultanément les calculs de plusieurs axes, et aussi de faire des calculs numériques efficaces en utilisant les outils de l'algèbre linéaire. Par exemple, les additions et soustractions de vecteurs géométriques peuvent être calculées directement en termes des vecteur-colonnes dans une base :

$$\vec{r}_{A/C} = \vec{r}_{A/B} + \vec{r}_{B/C} \quad \Rightarrow \quad \underline{r}_{A/C}^a = \underline{r}_{A/B}^a + \underline{r}_{B/C}^a \quad (3.26)$$

Ce transfert d'une seule équation vectorielle à une équation matricielle (équivalent à un système de trois équations scalaires) correspond à une projection de l'équation vectorielle sur chacun des axes de la base :

$$\vec{r}_{A/C} = \vec{r}_{A/B} + \vec{r}_{B/C} \quad \Rightarrow \quad \begin{bmatrix} (\vec{r}_{A/C} = \vec{r}_{A/B} + \vec{r}_{B/C}) \cdot \hat{a}_1 \\ (\vec{r}_{A/C} = \vec{r}_{A/B} + \vec{r}_{B/C}) \cdot \hat{a}_2 \\ (\vec{r}_{A/C} = \vec{r}_{A/B} + \vec{r}_{B/C}) \cdot \hat{a}_3 \end{bmatrix} \quad \Rightarrow \quad \underline{r}_{A/C}^a = \underline{r}_{A/B}^a + \underline{r}_{B/C}^a \quad (3.27)$$

Les équations vectorielles avec des vecteurs de position peuvent donc être substituées par des équations matricielles équivalentes avec les vecteur-colonnes, **à condition que tous les vecteur-colonnes soient exprimés dans une base commune**.

Programmation de calculs de cinématique : Lors de l'écriture d'un programme informatique pour effectuer des calculs de cinématique, les calculs numériques doivent être fait en termes de composantes. En effet, les ordinateurs sont adaptés à manipuler des chiffres, des vecteur-colonnes de chiffres, des matrices de chiffres, etc. Toutefois, un ordinateur ne peut manipuler directement des éléments conceptuels comme les vecteurs unitaires.

3.4.3 Calcul de longueurs, projections et angles avec les composantes

Lorsqu'on connaît les composantes d'un vecteur position \vec{r} , le produit intérieur peut être utilisé pour calculer sa longueur :

$$\|\vec{r}\|^2 = \vec{r} \cdot \vec{r} = \underline{r}^T \underline{r} = r_1^2 + r_2^2 + r_3^2 \quad (3.28)$$

$$\|\vec{r}\| = \sqrt{\underline{r}^T \underline{r}} \quad (3.29)$$

pour calculer une distance projetée selon un axe décrit par un vecteur unitaire \hat{n} :

$$d_n = \vec{r} \cdot \hat{n} = \underline{r}^T \underline{n} = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} = r_1 n_1 + r_2 n_2 + r_3 n_3 \quad (3.30)$$

pour calculer un angle entre deux vecteurs unitaires \hat{x} et \hat{y} :

$$\cos \angle(\hat{x}, \hat{y}) = \hat{x} \cdot \hat{y} = \underline{x}^T \underline{y} \quad (3.31)$$

$$\angle(\hat{x}, \hat{y}) = \arccos(\underline{x}^T \underline{y}) \quad (3.32)$$

et pour calculer entre deux vecteurs position \vec{r}_A et \vec{r}_B :

$$\vec{r}_A \cdot \vec{r}_B = \|\vec{r}_A\| \|\vec{r}_B\| \cos \angle(\vec{r}_A, \vec{r}_B) = \underline{r}_A^T \underline{r}_B \quad (3.33)$$

$$\angle(\vec{r}_A, \vec{r}_B) = \arccos \left(\frac{\underline{r}_A^T \underline{r}_B}{\sqrt{\underline{r}_A^T \underline{r}_A} \sqrt{\underline{r}_B^T \underline{r}_B}} \right) \quad (3.34)$$

Il est à noter que dans ces équations les vecteur-colonnes de composantes doivent tous être dans la même base. Les résultats sont des scalaires qui ne dépendent pas de la base utilisée.



Capsule vidéo

Opérations vectorielles avec les composantes de vecteurs

<https://youtu.be/7brvShmayAk>

3.5 Les matrices de rotations

Une matrice de rotation notée ${}^a R^b$, est une matrice (3×3 en 3D) qui permet de calculer directement les composants d'un vecteur position dans une base a avec les composants connus dans une base b , i.e. effectuer un **changement de base**, par une multiplication de la forme :

$$\underline{r}^a = {}^a R^b \underline{r}^b \quad (3.35)$$

La matrice de rotation est un raccourci pour projeter un vecteur position sur trois nouveaux axes en une seule opération matricielle, plutôt que d'effectuer trois opérations de projection indépendantes. Cette matrice représente l'orientation relative de deux bases vectorielles et **l'opération de multiplication d'un vecteur-colonne \underline{r}^a par la matrice de rotation ${}^b R^a$ correspond à la projection du vecteur géométrique associé \vec{r} sur la base vectorielle b .**

Deux rôles pour la cinématique : Une matrice de rotation a deux fonctions majeures dans un contexte de cinématique : **1)** effectuer des opérations de changement de base et **2)** représenter une orientation relative. Cette section présente le rôle des matrices de rotation pour les opérations de changement de base, les matrices de rotation seront discutées à nouveau dans le contexte de la représentation de l'orientation d'un corps rigide à la section [3.7](#).



Capsule vidéo

Les matrices de rotation et les changements de base

<https://youtu.be/hVmcRYD1KFQ>

3.5.1 Définitions

Une matrice de rotation est définie par les produits scalaires des vecteurs unitaires qui forment deux bases vectorielles :

$${}^a R^b = \begin{bmatrix} \hat{a}_1 \cdot \hat{b}_1 & \hat{a}_1 \cdot \hat{b}_2 & \hat{a}_1 \cdot \hat{b}_3 \\ \hat{a}_2 \cdot \hat{b}_1 & \hat{a}_2 \cdot \hat{b}_2 & \hat{a}_2 \cdot \hat{b}_3 \\ \hat{a}_3 \cdot \hat{b}_1 & \hat{a}_3 \cdot \hat{b}_2 & \hat{a}_3 \cdot \hat{b}_3 \end{bmatrix} = \begin{bmatrix} \cos \angle(\hat{a}_1, \hat{b}_1) & \cos \angle(\hat{a}_1, \hat{b}_2) & \cos \angle(\hat{a}_1, \hat{b}_3) \\ \cos \angle(\hat{a}_2, \hat{b}_1) & \cos \angle(\hat{a}_2, \hat{b}_2) & \cos \angle(\hat{a}_2, \hat{b}_3) \\ \cos \angle(\hat{a}_3, \hat{b}_1) & \cos \angle(\hat{a}_3, \hat{b}_2) & \cos \angle(\hat{a}_3, \hat{b}_3) \end{bmatrix} \quad (3.36)$$

Les produits scalaires correspondent aussi aux cosinus des angles entre les vecteurs unitaires, c'est pourquoi la matrice de rotation est aussi parfois appelée la **matrice des cosinus directeurs**. Par inspection de l'équation (3.36), on remarque que les colonnes de la matrice ${}^a R^b$ correspondent aux vecteur-colonnes des composantes des vecteurs unitaires \hat{b}_i exprimées dans la base a et que les rangées de la matrice ${}^a R^b$ correspondent à la transposée des vecteur-colonnes des composantes des vecteurs unitaires \hat{a}_i exprimées dans la base b . Cela permet donc des définitions équivalentes plus compactes :

$${}^a R^b = \left[\begin{bmatrix} b_1^a \\ b_2^a \\ b_3^a \end{bmatrix} \begin{bmatrix} b_1^a \\ b_2^a \\ b_3^a \end{bmatrix} \begin{bmatrix} b_1^a \\ b_2^a \\ b_3^a \end{bmatrix} \right] = \left[\begin{bmatrix} (\underline{a}_1^b)^T \\ (\underline{a}_2^b)^T \\ (\underline{a}_3^b)^T \end{bmatrix} \right] \quad (3.37)$$

Finalement, la matrice de rotation peut aussi être définie en termes de composantes et d'indices par :

$${}^a R_{ij}^b = \hat{a}_i \cdot \hat{b}_j \quad (3.38)$$

une équation qui est simple à se rappeler et permet d'éviter de mélanger l'ordre de rotation ${}^a R^b$ vs. ${}^b R^a$ avec la notation utilisée dans ces notes.

Preuve 3.5.1 Si les composantes d'un vecteur \vec{r} sont connus dans une base a :

$$\underline{r}^a = \begin{bmatrix} r_1^a \\ r_2^a \\ r_3^a \end{bmatrix} \quad (3.39)$$

le vecteur géométrique associé est donné par :

$$\vec{r} = r_1^a \hat{a}_1 + r_2^a \hat{a}_2 + r_3^a \hat{a}_3 \quad (3.40)$$

Les composantes dans une base b peuvent être calculées par des produits scalaires suivants :

$$r_1^b = \vec{r} \cdot \hat{b}_1 \quad (3.41)$$

$$r_2^b = \vec{r} \cdot \hat{b}_2 \quad (3.42)$$

$$r_3^b = \vec{r} \cdot \hat{b}_3 \quad (3.43)$$

Si on substitue \vec{r} par l'équation (3.40) :

$$r_1^b = (r_1^a \hat{a}_1 + r_2^a \hat{a}_2 + r_3^a \hat{a}_3) \cdot \hat{b}_1 \quad (3.44)$$

$$r_2^b = (r_1^a \hat{a}_1 + r_2^a \hat{a}_2 + r_3^a \hat{a}_3) \cdot \hat{b}_2 \quad (3.45)$$

$$r_3^b = (r_1^a \hat{a}_1 + r_2^a \hat{a}_2 + r_3^a \hat{a}_3) \cdot \hat{b}_3 \quad (3.46)$$

et on réarrange les termes :

$$r_1^b = (\hat{a}_1 \cdot \hat{b}_1)r_1^a + (\hat{a}_2 \cdot \hat{b}_1)r_2^a + (\hat{a}_3 \cdot \hat{b}_1)r_3^a \quad (3.47)$$

$$r_2^b = (\hat{a}_1 \cdot \hat{b}_2)r_1^a + (\hat{a}_2 \cdot \hat{b}_2)r_2^a + (\hat{a}_3 \cdot \hat{b}_2)r_3^a \quad (3.48)$$

$$r_3^b = (\hat{a}_1 \cdot \hat{b}_3)r_1^a + (\hat{a}_2 \cdot \hat{b}_3)r_2^a + (\hat{a}_3 \cdot \hat{b}_3)r_3^a \quad (3.49)$$

On obtient trois équations qui peuvent être regroupées sous forme matricielle :

$$\underbrace{\begin{bmatrix} r_1^b \\ r_2^b \\ r_3^b \end{bmatrix}}_{\underline{r}^b} = \underbrace{\begin{bmatrix} \hat{a}_1 \cdot \hat{b}_1 & \hat{a}_2 \cdot \hat{b}_1 & \hat{a}_3 \cdot \hat{b}_1 \\ \hat{a}_1 \cdot \hat{b}_2 & \hat{a}_2 \cdot \hat{b}_2 & \hat{a}_3 \cdot \hat{b}_2 \\ \hat{a}_1 \cdot \hat{b}_3 & \hat{a}_2 \cdot \hat{b}_3 & \hat{a}_3 \cdot \hat{b}_3 \end{bmatrix}}_{{}^b R^a} \underbrace{\begin{bmatrix} r_1^a \\ r_2^a \\ r_3^a \end{bmatrix}}_{\underline{r}^a} \quad (3.50)$$

La matrice de rotation ${}^b R^a$ consiste donc à des produits scalaires entre les vecteurs unitaires des bases vectorielles a et b comme définit à l'équation (3.36).

Exemple 2. Matrice de rotation dans le plan calculée par trigonométrie

La figure 3.12 montre un exemple d'un vecteur \vec{r} en 2 dimensions exprimé dans deux base vectorielles b et a . Comme illustré à la figure 3.13, avec une construction de triangles il est possible de déterminer par trigonométrie les relations entre les composantes dans les deux bases, et donc la matrice de rotation ${}^a R^b$. L'approche par trigonométrie est toutefois limitée pour les rotations en 3D, une approche vectorielle plus simple et plus universelle pour calculer des matrices de rotation est présentée à la section 3.5.2.

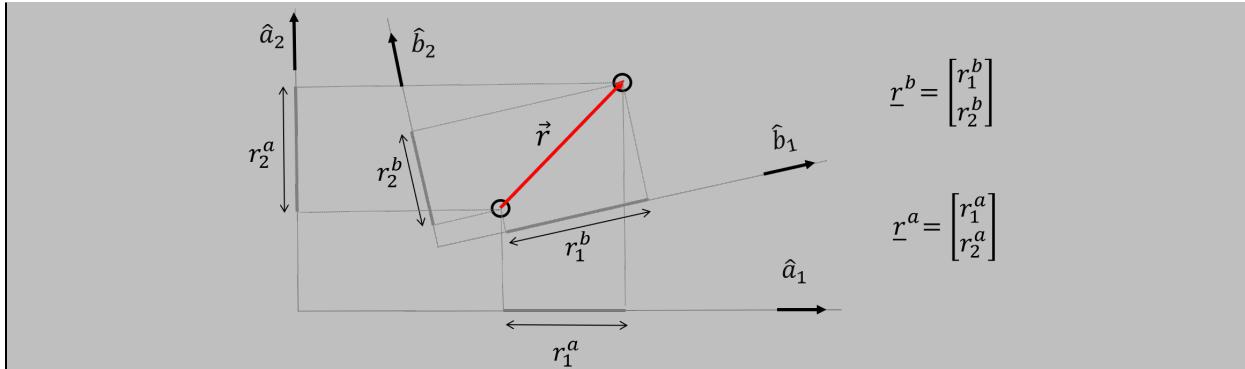
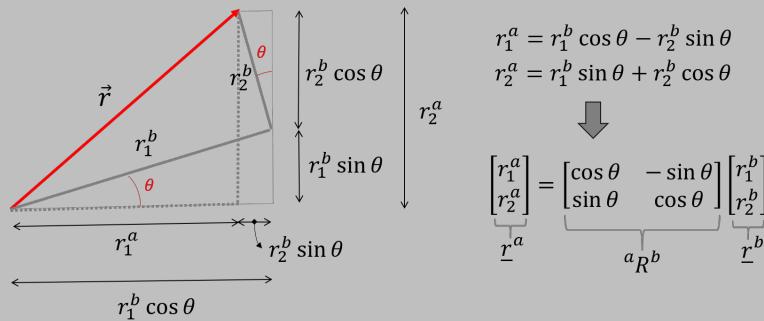


FIGURE 3.12 – Vecteur position exprimé dans deux bases vectorielles

FIGURE 3.13 – Calcul par trigonométrie de la matrice de rotation ${}^a R^b$ pour le changement de base illustré à la figure 3.12

3.5.2 Méthode de calcul des matrices de rotation basé sur les colonnes

La définition des matrices de rotation basée sur les vecteurs unitaires de eq.(3.37) est particulièrement utile pour calculer une matrice de rotation. Une astuce pour ne pas mélanger l'ordre de rotation (${}^b R^a$ vs. ${}^a R^b$) est d'avoir en tête que la multiplication matricielle c'est une combinaison linéaire de vecteur-colonnes :

$$\underline{r}^a = \underbrace{\left[\begin{bmatrix} \underline{b}_1^a \\ \underline{b}_2^a \\ \underline{b}_3^a \end{bmatrix} \right]}_{{}^a R^b} \underbrace{\left[\begin{bmatrix} r_1^b \\ r_2^b \\ r_3^b \end{bmatrix} \right]}_{\underline{r}^b} = \begin{bmatrix} \underline{b}_1^a \\ \underline{b}_2^a \\ \underline{b}_3^a \end{bmatrix} r_1^b + \begin{bmatrix} \underline{b}_1^a \\ \underline{b}_2^a \\ \underline{b}_3^a \end{bmatrix} r_2^b + \begin{bmatrix} \underline{b}_1^a \\ \underline{b}_2^a \\ \underline{b}_3^a \end{bmatrix} r_3^b \quad (3.51)$$

et donc que si on cherche la matrice de rotation ${}^a R^b$ pour calculer le passage de la base b vers la base a , il faut trouver les directions spatiales \hat{b}_1 , \hat{b}_2 et \hat{b}_3 qui seront combinées linéairement selon les coefficients r_1^b , r_2^b , r_3^b .

Une méthode rapide et universelle pour calculer une matrice de rotation est de construire par inspection les vecteur unitaires \hat{b}_i de la base vectorielle initiale, comme une combinaison de vecteurs unitaires \hat{a}_i de la base cible :

$$\hat{b}_1 = a \hat{a}_1 + b \hat{a}_2 + c \hat{a}_3 \quad \hat{b}_2 = d \hat{a}_1 + e \hat{a}_2 + f \hat{a}_3 \quad \hat{b}_3 = g \hat{a}_1 + h \hat{a}_2 + i \hat{a}_3 \quad (3.52)$$

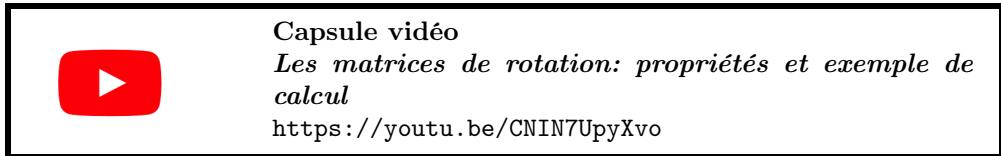
pour ensuite convertir les équations en vecteur-colonnes \underline{b}_i^a :

$$\underline{b}_1^a = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad \underline{b}_2^a = \begin{bmatrix} d \\ e \\ f \end{bmatrix} \quad \underline{b}_3^a = \begin{bmatrix} g \\ h \\ i \end{bmatrix} \quad (3.53)$$

et assembler la matrice avec la définition de l'équation (3.37) :

$${}^a R^b = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix} = \begin{bmatrix} a & b^a & b_3^a \\ b_1^a & b_2^a & b_3^a \end{bmatrix} \quad (3.54)$$

Pour les matrices de rotation élémentaires, i.e. les matrices qui représentent une rotation d'un angle θ autour de l'axe \hat{b}_1 , \hat{b}_2 ou \hat{b}_3 , les neufs éléments a, b, c, d, \dots de la matrice peuvent seulement prendre comme valeurs $\cos \theta$, $\pm \sin \theta$, 1 ou 0. Il est donc relativement facile de les identifier par inspection, comme il est illustré à la section suivante.



3.5.3 Matrices de rotation élémentaires

Les matrices de rotation élémentaires représentent des rotations relatives d'un angle θ autour de l'axe 1, 2 ou 3 entre deux bases vectorielles. Ces matrices seront notées $R_i(\theta)$ où i est l'indice de l'axe de rotation. Ces matrices sont caractérisées par le fait que le vecteur unitaire \hat{b}_i de la base mobile coïncide avec le vecteur unitaire \hat{a}_i de la base fixe, et la relation entre les autres vecteurs unitaires peut être facilement représentée dans le plan perpendiculaire à l'axe de rotation.

Notation simplifiée pour les calculs trigonométriques : Les symboles $s\theta$ et $c\theta$ sont ici introduits pour alléger les équations et représentent les fonctions trigonométriques $\cos(\theta)$ et $\sin(\theta)$ respectivement :

$$s\theta = \sin(\theta) \quad (3.55)$$

$$c\theta = \cos(\theta) \quad (3.56)$$

Pour les calculs de cinématique qui implique plusieurs angles, une notation encore plus compacte sera utilisée avec seulement les lettres s ou c et des indices qui spécifie les angles :

$$s_i = \sin(\theta_i) \quad (3.57)$$

$$c_i = \cos(\theta_i) \quad (3.58)$$

$$s_{ijk} = \sin(\theta_i + \theta_j + \theta_k) \quad (3.59)$$

Rotation selon l'axe 3 Si les bases vectorielles a et b ont une orientation relative décrite par une rotation d'un angle θ autour de l'axe 3, les vecteurs unitaires \hat{a}_3 et \hat{b}_3 sont égaux et, comme illustré à la figure 3.14, les quatre autres vecteurs unitaires (\hat{a}_1 , \hat{b}_1 , \hat{a}_2 et \hat{b}_2) sont dans le plan formé par l'axe 1 et l'axe 2. Les vecteurs unitaires sont reliés par des fonctions trigonométriques simples :

$$\hat{b}_1 = c\theta \hat{a}_1 + s\theta \hat{a}_2 \quad \hat{b}_2 = -s\theta \hat{a}_1 + c\theta \hat{a}_2 \quad \hat{b}_3 = \hat{a}_3 \quad (3.60)$$

La matrice ${}^a R^b$ peut donc être formée grâce à la définition (3.37) :

$$\begin{aligned} b_1^a &= \begin{bmatrix} c\theta \\ s\theta \\ 0 \end{bmatrix} & b_2^a &= \begin{bmatrix} -s\theta \\ c\theta \\ 0 \end{bmatrix} & b_3^a &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & \Rightarrow {}^a R_3^b(\theta) &= \begin{bmatrix} c\theta & -s\theta & 0 \\ s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (3.61)$$

À noter que le coin supérieur gauche de la matrice ${}^a R_3^b(\theta)$ correspond à une matrice de rotation 2×2 dans le plan.

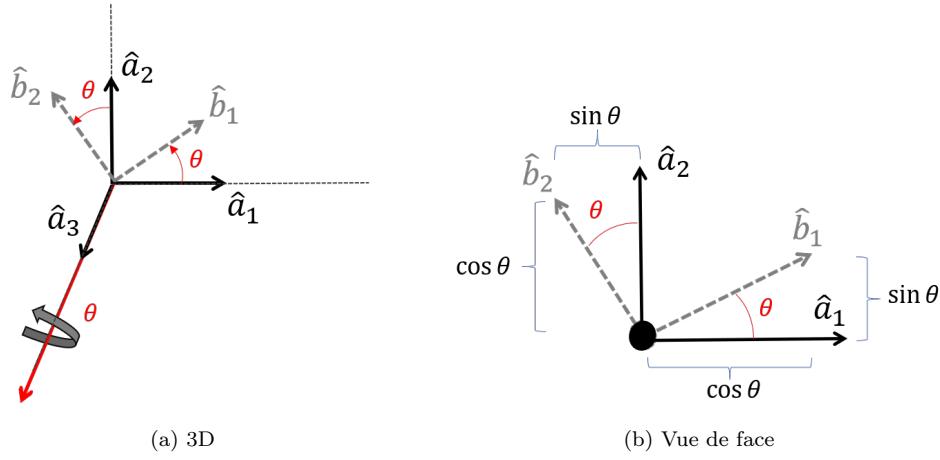


FIGURE 3.14 – Rotation relative selon l’axe 3

Rotation selon l’axe 2 Si les bases vectorielles a et b ont une orientation relative décrite par une rotation d’un angle θ autour de l’axe 2, les vecteurs unitaires \hat{a}_2 et \hat{b}_2 sont égaux et, comme illustré à la figure 3.15, les quatre autres vecteurs unitaires (\hat{a}_1 , \hat{b}_1 , \hat{a}_3 et \hat{b}_3) sont dans le plan formé par l’axe 1 et l’axe 3. Les vecteurs unitaires sont reliés par des fonctions trigonométriques simples :

$$\hat{b}_1 = c\theta \hat{a}_1 - s\theta \hat{a}_3 \quad \hat{b}_2 = \hat{a}_2 \quad \hat{b}_3 = s\theta \hat{a}_1 + c\theta \hat{a}_3 \quad (3.62)$$

La matrice ${}^a R^b$ peut donc être formée grâce à la définition (3.37) :

$$\underline{b}_1^a = \begin{bmatrix} c\theta \\ 0 \\ -s\theta \end{bmatrix} \quad \underline{b}_2^a = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \underline{b}_3^a = \begin{bmatrix} s\theta \\ 0 \\ c\theta \end{bmatrix} \quad \Rightarrow \quad {}^a R_2^b(\theta) = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \quad (3.63)$$

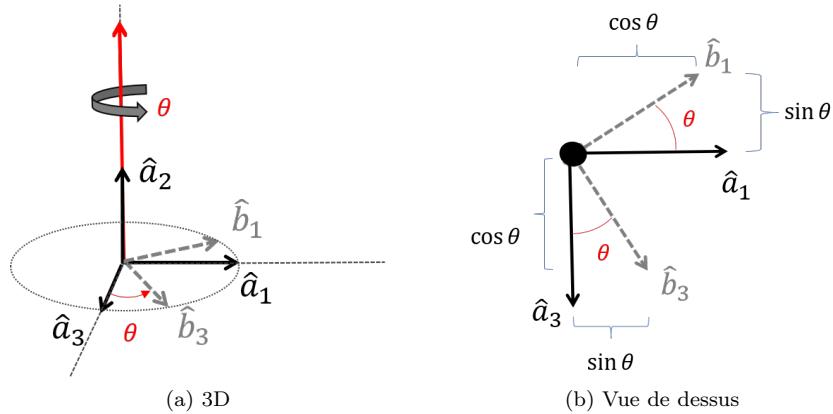


FIGURE 3.15 – Rotation relative selon l’axe 2

Rotation selon l’axe 1 Si les bases vectorielles a et b ont une orientation relative décrite par une rotation d’un angle θ autour de l’axe 1, les vecteurs unitaires \hat{a}_1 et \hat{b}_1 sont égaux et, comme illustré à la figure 3.14, les quatre autres vecteurs unitaires (\hat{a}_2 , \hat{b}_2 , \hat{a}_3 et \hat{b}_3) sont dans le plan formé par l’axe 2 et l’axe 3. Les vecteurs unitaires sont reliés par des fonctions trigonométriques simples :

$$\hat{b}_1 = \hat{a}_1 \quad \hat{b}_2 = c\theta \hat{a}_2 + s\theta \hat{a}_3 \quad \hat{b}_3 = -s\theta \hat{a}_2 + c\theta \hat{a}_3 \quad (3.64)$$

La matrice ${}^a R^b$ peut donc être formée grâce à la définition (3.37) :

$$\underline{b}_1^a = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \underline{b}_2^a = \begin{bmatrix} 0 \\ c\theta \\ s\theta \end{bmatrix} \quad \underline{b}_3^a = \begin{bmatrix} 0 \\ -s\theta \\ c\theta \end{bmatrix} \Rightarrow {}^a R_1^b(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\theta & -s\theta \\ 0 & s\theta & c\theta \end{bmatrix} \quad (3.65)$$

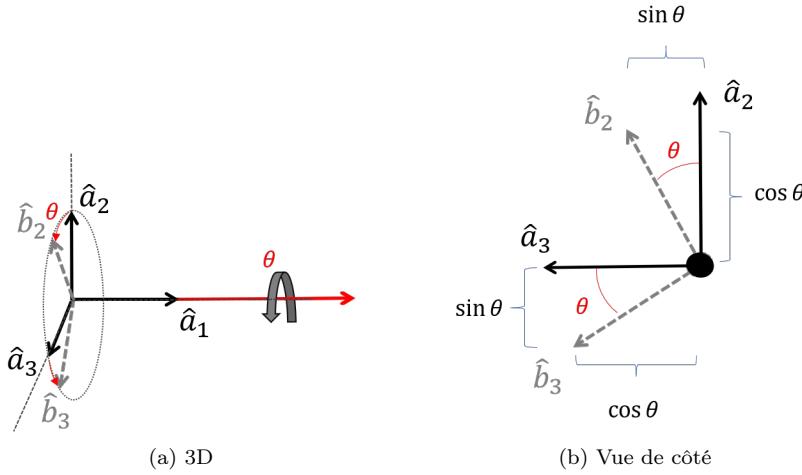


FIGURE 3.16 – Rotation relative selon l'axe 1

3.5.4 Propriétés des matrices de rotation

Cette section présente plusieurs propriétés des matrices de rotation qui sont utiles dans un contexte de changement de bases vectorielles.

Propriété 3.5.1 (Norme des colonnes et rangés) *Les colonnes \underline{c}_i et les rangés \underline{r}_i d'une matrice de rotation ont une norme égale à 1 :*

$$R = \left[\begin{bmatrix} \underline{c}_1 \\ \underline{c}_2 \\ \underline{c}_3 \end{bmatrix} \quad \begin{bmatrix} \underline{c}_2 \\ \underline{c}_3 \\ \underline{c}_1 \end{bmatrix} \quad \begin{bmatrix} \underline{c}_3 \\ \underline{c}_1 \\ \underline{c}_2 \end{bmatrix} \right] = \left[\begin{bmatrix} \underline{r}_1^T \\ \underline{r}_2^T \\ \underline{r}_3^T \end{bmatrix} \right] \Rightarrow (\underline{c}_i)^T \underline{c}_i = 1 \quad \& \quad (\underline{r}_i)^T \underline{r}_i = 1 \quad \forall i \quad (3.66)$$

L'identité trigonométrique suivante est souvent utile pour ce type de calculs :

$$\sin^2(\theta) + \cos^2(\theta) = 1 \quad (3.67)$$

Propriété 3.5.2 (Matrice identité) *Une matrice de rotation est réduite à une matrice identité lorsque les bases vectorielles sont coïncidentes. Par exemple, pour les matrices de rotation élémentaires des équations (3.61), (3.63) et (3.65), lorsque l'angle est égal à zéro, les matrices sont réduites à des matrices identités :*

$$R_1(\theta) = R_2(\theta) = R_3(\theta) = I_{3 \times 3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{si } \theta = 0 \quad (3.68)$$

Propriété 3.5.3 (Inversion) *Pour inverser la direction d'une opération de changement de base, il suffit de multiplier par l'inverse de la matrice de rotation :*

$$\underline{r}^a = {}^a R^b \underline{r}^b \quad (3.69)$$

$$({}^a R^b)^{-1} \underline{r}^a = ({}^a R^b)^{-1} {}^a R^b \underline{r}^b \quad (3.70)$$

$$({}^a R^b)^{-1} \underline{r}^a = \underline{r}^b \quad (3.71)$$

L'inversion d'une matrice de rotation est équivalente à inverser le signe de l'angle θ utilisé dans le calcul de la matrice, à prendre la transposée de la matrice et, en termes de notation, à inverser les bases :

$$R(\theta)^{-1} = R(-\theta) \quad (3.72)$$

$$R^{-1} = R^T \quad (3.73)$$

$$({}^b R^a)^{-1} = {}^a R^b \quad (3.74)$$

Propriété 3.5.4 (Changements de base successifs) Des changements de base successifs $a \rightarrow b \rightarrow c \rightarrow \dots \rightarrow z$ peuvent être combinés en une seule matrice de rotation en multipliant les matrices de rotation intermédiaires par la gauche :

$${}^c R^a = {}^c R^b {}^b R^a \quad (3.75)$$

$${}^d R^a = {}^d R^c {}^c R^b {}^b R^a \quad (3.76)$$

$$\vdots \quad (3.77)$$

$${}^z R^a = {}^z R^y {}^y R^x \dots {}^c R^b {}^b R^a \quad (3.78)$$

Notez que selon la notation utilisée dans ces notes, **les lettres des bases intermédiaires doivent être côté-à-côte dans les équations**, et la matrice de rotation totale conserve la première et la dernière base de la séquence dans le même ordre. Cette propriété découle des définitions, en combinant deux changements de base intermédiaires, une matrice totale peut être obtenue :

$$\left. \begin{array}{l} \underline{r}^b = {}^b R^a \underline{r}^a \\ \underline{r}^c = {}^c R^b \underline{r}^b \\ \underline{r}^c = {}^c R^a \underline{r}^a \end{array} \right\} \Rightarrow \underline{r}^c = \underbrace{{}^c R^b {}^b R^a}_{{}^c R^a} \underline{r}^a \Rightarrow {}^c R^a = {}^c R^b {}^b R^a \quad (3.79)$$

Propriété 3.5.5 (Non-commutativité) Le produit des matrices de rotation n'est toutefois pas commutatif, sauf pour des petites rotations infinitésimales. L'ordre de multiplication des matrices est donc important et généralement :

$${}^c R^b {}^b R^a \neq {}^b R^a {}^c R^b \quad (3.80)$$

Une **exception à cette règle** est pour les rotations successives selon le même axe de rotation. Par exemple pour deux matrices de rotation élémentaires selon un axe i :

$$R_i(\theta_1)R_i(\theta_2) = R_i(\theta_2)R_i(\theta_1) = R_i(\theta_1 + \theta_2) \quad (3.81)$$

Exemple 3. Réorganisation des vecteurs unitaires

La figure 3.17 illustre deux bases vectorielles qui ont des vecteurs unitaires parallèles mais avec des directions et un ordre différents. Pour calculer la matrice de rotation associée ${}^b R^a$, la première étape est d'identifier la relation entre les vecteurs unitaires \hat{a}_i et \hat{b}_i .

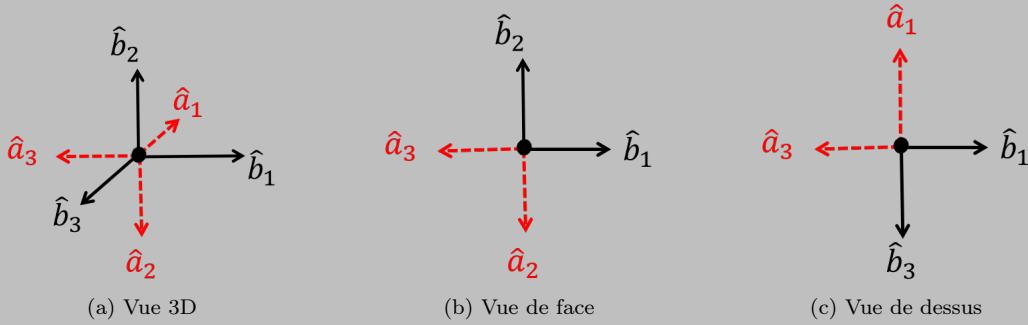


FIGURE 3.17 – Réorganisation des directions des vecteurs unitaires

On peut constater que le vecteur unitaire $\hat{a}_1 = -\hat{b}_3$, le vecteur unitaire $\hat{a}_2 = -\hat{b}_2$ et le vecteur unitaire $\hat{a}_3 = -\hat{b}_1$. Avec ces résultats il est possible d'assembler les vecteur-colonnes \underline{a}_i^b :

$$\underline{a}_1^b = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \quad \underline{a}_2^b = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} \quad \underline{a}_3^b = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \quad (3.82)$$

Il est donc maintenant possible d'assembler la matrice de rotation ${}^b R^a$ basée sur l'équation (3.37) :

$${}^b R^a = [\underline{a}_1^b \quad \underline{a}_2^b \quad \underline{a}_3^b] = \begin{bmatrix} 0 & 0 & -1 \\ 0 & -1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad (3.83)$$

Exemple 4. Réorganisation des vecteurs unitaires et rotation selon un axe

Un exemple de calcul d'une matrice de rotation ${}^b R^a$ est donné pour une base a qui subit une révolution de θ degrés autour de l'axe \hat{b}_3 par rapport à une base b , mais ici avec des vecteurs unitaires qui sont définis avec des directions différentes, tel qu'illustré à la figure 3.18a, et ce n'est donc pas un cas de rotation élémentaire. Pour construire la matrice de rotation, la première étape est de dessiner les deux bases avec une origine commune, dans une configuration où la rotation θ est relativement faible (pour simplifier la visualisation). On peut constater que le vecteur unitaire \hat{a}_1 est aligné avec l'axe de rotation, donc indépendant de θ et toujours donné par $\hat{a}_1 = -\hat{b}_3$. Ensuite, par inspection, il est possible d'exprimer chaque vecteur unitaire \hat{a}_i comme une combinaison des vecteurs unitaires \hat{b}_i , avec des relations trigonométriques simples comme illustrés par les figures 3.18b et 3.18.

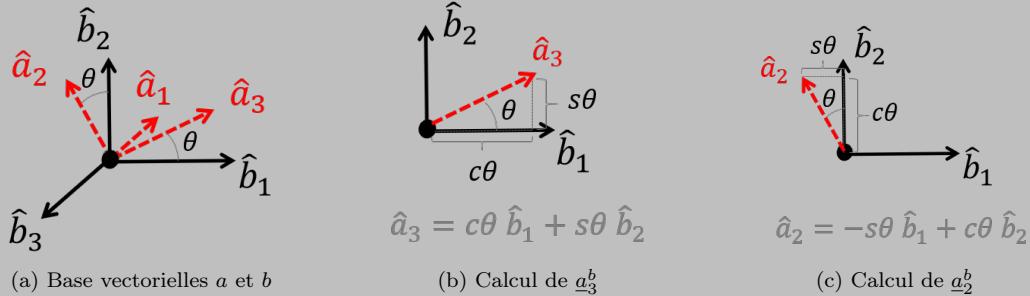


FIGURE 3.18 – Rotation selon l'axe 3 et réorganisation des directions des vecteurs unitaires

Avec ces résultats il est possible d'assembler les vecteur-colonnes \underline{a}_i^b :

$$\underline{a}_1^b = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \quad \underline{a}_2^b = \begin{bmatrix} -s\theta \\ c\theta \\ 0 \end{bmatrix} \quad \underline{a}_3^b = \begin{bmatrix} c\theta \\ s\theta \\ 0 \end{bmatrix} \quad (3.84)$$

Il est donc maintenant possible d'assembler la matrice de rotation ${}^b R^a$ basé sur l'équation (3.37) :

$${}^b R^a = [\underline{a}_1^b \quad \underline{a}_2^b \quad \underline{a}_3^b] = \begin{bmatrix} 0 & -s\theta & c\theta \\ 0 & c\theta & s\theta \\ -1 & 0 & 0 \end{bmatrix} \quad (3.85)$$

Exemple 5. Calcul de la position globale de l'effecteur d'un robot

Dans cet exemple, le robot précédemment étudié (Figure 3.9) est ici analysé à nouveau mais cette fois avec l'aide des bases vectorielles. L'objectif est ici de calculer la position du point D par rapport au point A exprimée dans la base vectorielle a. Comme illustré à la figure ??, la première étape est de décrire la position qui doit être calculée en terme de vecteurs de position géométriques :

$$\vec{r}_{D/A} = \vec{r}_{D/C} + \vec{r}_{C/B} + \vec{r}_{B/A} \quad (3.86)$$

Cette relation vectorielle peut ensuite être convertie en une équation matricielle entre les composantes dans une base a fixe qui correspond aux axes selon lesquels la position doit être exprimée :

$$\underline{r}_{D/A}^a = \underline{r}_{D/C}^a + \underline{r}_{C/B}^a + \underline{r}_{B/A}^a \quad (3.87)$$

Ensuite, comme illustré à la Figure 3.19, les différents vecteurs positions qui représentent chaque joint ont des composantes connues dans des bases vectorielles locales attachées aux différents liens rigides du robot :

$$\underline{r}_{B/A}^b = \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} \quad \underline{r}_{C/B}^c = \begin{bmatrix} 0 \\ l_2 \\ 0 \end{bmatrix} \quad \underline{r}_{D/C}^d = \begin{bmatrix} 0 \\ l_3 \\ 0 \end{bmatrix} \quad (3.88)$$

Il est à noter que ces vecteur-colonnes qui utilisent les bases locales sont constants et indépendants des angles des joints car les bases locales tournent avec les liens rigides. Il est ensuite possible de convertir ces vecteur-colonnes de positions dans la base a par des opérations de changement de base avec les matrices de rotation appropriées :

$$\underline{r}_{D/C}^d = {}^d R^c \underline{r}_{C/B}^c \quad \underline{r}_{C/B}^c = {}^c R^b \underline{r}_{B/A}^b \quad \underline{r}_{B/A}^a = {}^b R^a \underline{r}_{B/A}^b \quad (3.89)$$

Finalement, en substituant dans l'équation (3.87), l'équation matricielle nécessaire pour obtenir les composantes du vecteur position $\vec{r}_{D/A}$ exprimé avec la base a est obtenue :

$$\underline{r}_{D/A}^a = {}^a R^d \underline{r}_{D/C}^d + {}^a R^c \underline{r}_{C/B}^c + {}^a R^b \underline{r}_{B/A}^b \quad (3.90)$$

Les matrices de rotation doivent ensuite être calculées pour solutionner cette équation.

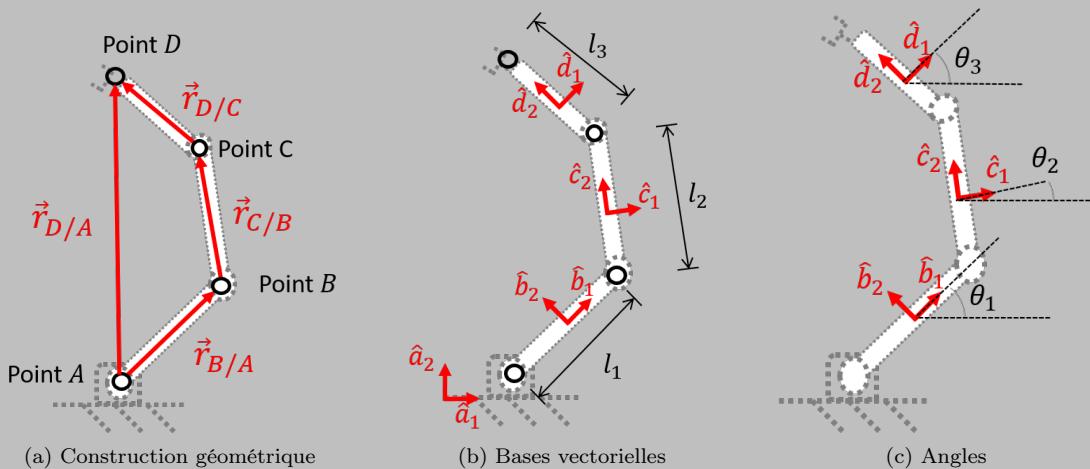


FIGURE 3.19 – Utilisation des bases vectorielles pour les vecteurs de position

Grâce à une inspection des mouvements relatifs des bases vectorielles, comme illustrée à la figure 3.19b, les matrices de rotation peuvent être calculées. Puisque le robot est planaire, les bases vectorielles b , c et d fixées aux pièces mobiles du robot ont tous une rotation relative autour de l'axe \hat{a}_3 par rapport à la base vectorielle fixe a , tel qu'illustré à la figure 3.20.

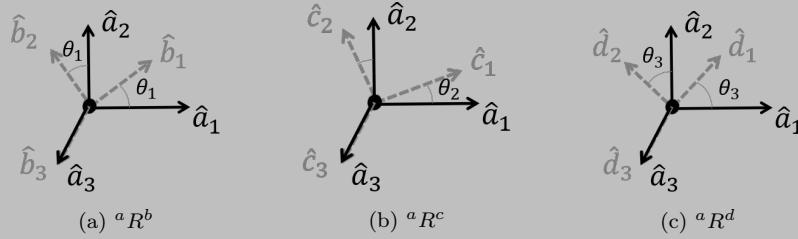


FIGURE 3.20 – Calcul des matrices de rotation pour l'exemple 5

En fonction des angles définis à la figure 3.19c, les matrices de rotation sont donc toutes des matrices élémentaires de rotation autour de l'axe 3 :

$${}^a R^b = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 \\ s\theta_1 & c\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad {}^a R^c = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 \\ s\theta_2 & c\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad {}^a R^d = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 \\ s\theta_3 & c\theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.91)$$

Ces résultats peuvent donc être substitués dans l'équation matricielle qui avait été obtenue à l'exemple 5 :

$$\underline{r}_{D/A}^a = {}^a R^d \underline{r}_{D/C}^d + {}^a R^c \underline{r}_{C/B}^c + {}^a R^b \underline{r}_{B/A}^b \quad (3.92)$$

et on obtient l'expression précise qui permet de calculer la position du point D par rapport au point de référence A exprimée dans la base vectorielle a en fonction des variables de distances l_i et d'angles θ_i :

$$\underline{r}_{D/A}^a = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 \\ s\theta_3 & c\theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ l_3 \\ 0 \end{bmatrix} + \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 \\ s\theta_2 & c\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ l_2 \\ 0 \end{bmatrix} + \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 \\ s\theta_1 & c\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} \quad (3.93)$$

$$\underline{r}_{D/A}^a = \begin{bmatrix} -s\theta_3 l_3 - s\theta_2 l_2 + c\theta_1 l_1 \\ c\theta_3 l_3 + c\theta_2 l_2 + s\theta_1 l_1 \\ 0 \end{bmatrix} \quad (3.94)$$

À noter que ici les angles θ_i sont tous des angles absolus qui représentent l'orientation des bases b , c et d du robot par rapport à la base fixe a directement. Lorsqu'on contrôle un bras robotisé, c'est généralement les angles relatifs entre chacun des joints qui sont les variables mesurées et contrôlées.

Exercice de lecture : La hauteur du robot correspond à la deuxième ligne de l'équation (3.94), qui est la distance $\vec{r}_{D/A}$ selon le vecteur unitaire \hat{a}_2 :

$$h = c\theta_3 l_3 + c\theta_2 l_2 + s\theta_1 l_1 \quad (3.95)$$

Alternativement, à l'exemple 1, l'équation suivante avait été obtenue pour la hauteur du robot :

$$h = l_1 \cos(\varphi_1) + l_2 \cos(\varphi_2) + l_3 \cos(\varphi_3) \quad (3.96)$$

Trouvez la correspondance entre les angles θ_i et φ_i et vérifiez que les deux expressions sont équivalentes.

Exemple 6. Distance projetée calculée avec les vecteur-colonnes

Comme illustré à la figure 3.21, cet exemple démontre comment calculer la distance projetée d_n d'un vecteur $\vec{r}_{C/A}$ selon un axe décrit par \hat{n} , connaissant les vecteur-colonnes de composantes :

$$\underline{r}_{C/B}^b = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \underline{r}_{B/A}^a = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad \underline{n}^b = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} \quad (3.97)$$

Premièrement, l'équation vectorielle de vecteurs géométriques de position est développée :

$$\vec{r}_{C/A} = \vec{r}_{C/B} + \vec{r}_{B/A} \quad (3.98)$$

pour ensuite effectuer le produit scalaire pour calculer la distance projetée :

$$d_n = \vec{r}_{C/A} \cdot \hat{n} = (\vec{r}_{C/B} + \vec{r}_{B/A}) \cdot \hat{n} \quad (3.99)$$

Deuxièmement, on choisit la base b pour transformer la relation vectorielle en équation matricielle :

$$d_n = (\underline{r}_{C/B}^b + \underline{r}_{B/A}^b)^T \underline{n}^b \quad (3.100)$$

$$d_n = (\underline{r}_{C/B}^b)^T \underline{n}^b + (\underline{r}_{B/A}^b)^T \underline{n}^b \quad (3.101)$$

Tous les vecteur-colonnes sont connus sauf le vecteur-colonne $\underline{r}_{C/B}^b$, qui doit être substitué par $\underline{r}_{C/B}^b = {}^b R^a \underline{r}_{C/B}^a$ car les composantes de $\vec{r}_{C/B}$ sont connues seulement dans la base a . On obtient alors :

$$d_n = ({}^b R^a \underline{r}_{C/B}^a)^T \underline{n}^b + (\underline{r}_{B/A}^b)^T \underline{n}^b \quad (3.102)$$

$$d_n = (\underline{n}^b)^T {}^b R^a \underline{r}_{C/B}^a + (\underline{r}_{B/A}^b)^T \underline{n}^b \quad (3.103)$$

et ce qui correspond en termes de composantes à l'équation :

$$d_n = \begin{bmatrix} n_1^b & n_2^b & n_3^b \end{bmatrix} \begin{bmatrix} {}^b R_{11}^a & {}^b R_{12}^a & {}^b R_{13}^a \\ {}^b R_{21}^a & {}^b R_{22}^a & {}^b R_{23}^a \\ {}^b R_{31}^a & {}^b R_{32}^a & {}^b R_{33}^a \end{bmatrix} \begin{bmatrix} r_{C/B,1}^a \\ r_{C/B,2}^a \\ r_{C/B,3}^a \end{bmatrix} + \begin{bmatrix} r_{B/A,1}^b & r_{B/A,2}^b & r_{B/A,3}^b \end{bmatrix} \begin{bmatrix} n_1^b \\ n_2^b \\ n_3^b \end{bmatrix} \quad (3.104)$$

si on substitue par les composantes connues définies à l'équation (3.97) :

$$d_n = \begin{bmatrix} n_x & n_y & n_z \end{bmatrix} \begin{bmatrix} {}^b R_{11}^a & {}^b R_{12}^a & {}^b R_{13}^a \\ {}^b R_{21}^a & {}^b R_{22}^a & {}^b R_{23}^a \\ {}^b R_{31}^a & {}^b R_{32}^a & {}^b R_{33}^a \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} X & Y & Z \end{bmatrix} \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} \quad (3.105)$$

C'est donc cette équation qui devrait être programmée pour faire ce calcul numériquement sur un ordinateur.

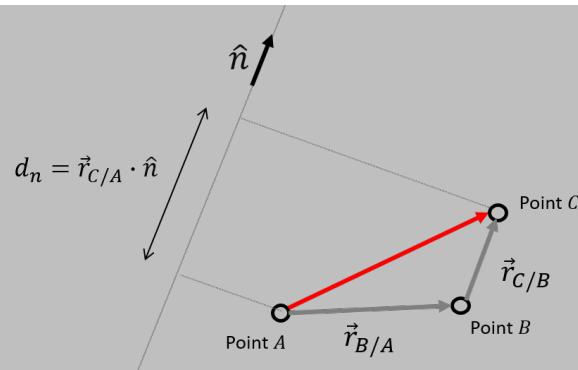


FIGURE 3.21 – Exemple de calcul d'une distance projetée

3.6 Coordonnées dans un repère et transformations homogènes

En robotique, il est souvent utile de placer un point de référence ainsi qu'une base vectorielle sur chaque corps rigide du robot. Bien qu'on peut travailler avec des points de référence et des bases vectorielles de façon indépendante, il est souvent pratique de les jumeler en paires puisqu'il faut ces deux types de références pour exprimer des coordonnées. **La combinaison d'un point d'origine et d'une base vectorielle est appelée repère dans ces notes**, le terme *Frame* est généralement utilisé en anglais. Un repère A dans ces notes fait donc référence à la fois à un point d'origine A_o et à une base vectorielle a qui consiste des trois vecteurs unitaires $\{\hat{a}_1, \hat{a}_2, \hat{a}_3\}$. Autrement dit, on appellera *repère A* le repère formé par l'ensemble $\{A_o, \hat{a}_1, \hat{a}_2, \hat{a}_3\}$.

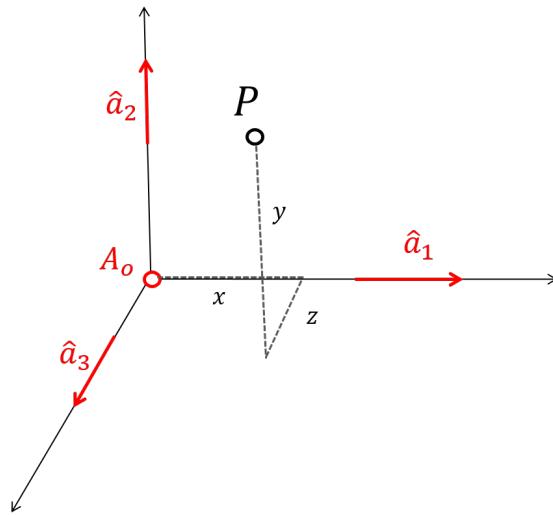


FIGURE 3.22 – Système de coordonnées cartésiennes (x, y, z) basé sur le repère $\{A_o, \hat{a}_1, \hat{a}_2, \hat{a}_3\}$

Comme illustré à la figure 3.22, les coordonnées cartésiennes (x, y, z) d'un point P correspondent directement aux composantes du vecteur position avec comme origine l'intersection des axes du système de coordonnées (le point A_o) et exprimé dans une base vectorielle a alignée sur les axes du système de coordonnées :

$$\underline{r}_{P/A_o}^a = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \Leftrightarrow \vec{r}_{P/A_o} = x \hat{a}_1 + y \hat{a}_2 + z \hat{a}_3 \quad (3.106)$$

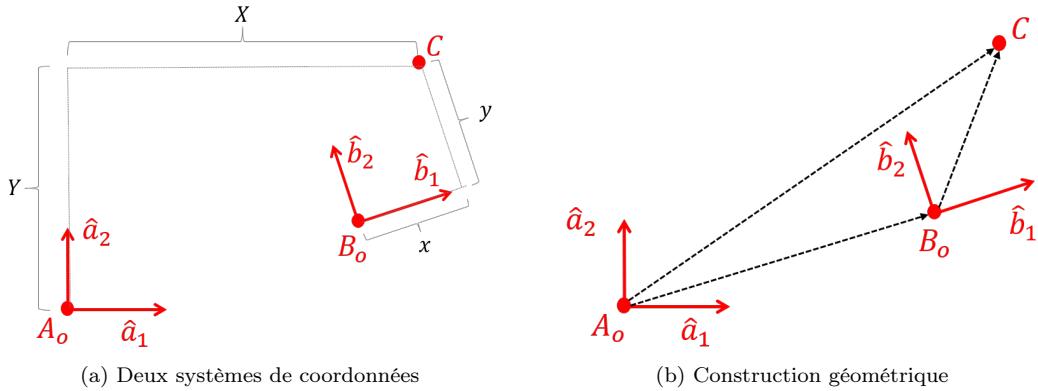
Un système de coordonnées est toujours implicitement attaché à un repère, même s'il n'est pas explicitement défini. L'ensemble de l'origine des axes et des vecteurs unitaires alignés sur ces axes forment un repère.

▶

Capsule vidéo
Les repères et les matrices de transformation
<https://youtu.be/yTPA1R8b8XE>

3.6.1 Changement de repère

Un changement de repère implique deux opérations : **1)** un changement de l'origine utilisée pour les vecteurs de position et **2)** un changement de la base vectorielle utilisée pour exprimer les composantes de ce vecteur position. Comme illustré à la figure 3.23 pour une transformation d'un repère B vers un repère A, les coordonnées cartésiennes d'un point C correspondent au vecteur-colonne $\underline{r}_{C/B}^b$ dans le repère B et au

FIGURE 3.23 – Changement du repère $\{B_o, \hat{b}_1, \hat{b}_2, \hat{b}_3\}$ vers le repère $\{A_o, \hat{a}_1, \hat{a}_2, \hat{a}_3\}$

vecteur-colonne \underline{r}_{C/A_o}^a dans le repère A :

$$\underline{r}_{C/B_o}^b = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \underline{r}_{C/A_o}^a = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (3.107)$$

Pour effectuer la transformation des coordonnées (x, y, z) vers les coordonnées (X, Y, Z) , la première étape est de construire la relation vectorielle qui relie les trois points :

$$\vec{r}_{C/A_o} = \vec{r}_{C/B_o} + \vec{r}_{B_o/A_o} \quad (3.108)$$

et ensuite il suffit de tout exprimer ces vecteurs dans la base a pour effectuer l'addition :

$$\underline{r}_{C/A_o}^a = \underline{r}_{C/B_o}^a + \underline{r}_{B_o/A_o}^a \quad (3.109)$$

$$\underline{r}_{C/A_o}^a = {}^a R^b \underline{r}_{C/B_o}^b + \underline{r}_{B_o/A_o}^a \quad (3.110)$$

ce qui correspond en termes de coordonnées à :

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = {}^a R^b \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \underline{r}_{B_o/A_o}^a \quad (3.111)$$

Un changement de base est nécessaire puisque que le vecteur \vec{r}_{C/B_o} est connu dans la base vectorielle b . Donc les informations nécessaires pour effectuer la transformation sont : 1) les composantes $\underline{r}_{B_o/A_o}^a$ du vecteur position qui relie les origines A_o et B_o et 2) la matrice de rotation ${}^a R^b$ qui donne l'orientation relative des bases vectorielles a et b .

L'**opération inverse** s'obtient à partir de l'équation (3.110), d'abord on soustrait le vecteur translation $\underline{r}_{B_o/A_o}^a$ et on multiplie par la gauche l'inverse de la matrice de rotation pour obtenir :

$$\underline{r}_{C/B_o}^b = {}^b R^a \underline{r}_{C/A_o}^a - {}^b R^a \underline{r}_{B_o/A_o}^a \quad (3.112)$$

ce qui correspond en termes des coordonnées à :

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = {}^b R^a \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + {}^b R^a \underline{r}_{B_o/A_o}^a \quad (3.113)$$

3.6.2 Coordonnées et transformations homogènes

Dans les logiciels de cinématique et de graphique, les opérations de changement de repère peuvent être très fréquentes. Il existe une méthode standard pour combiner l'opération d'addition vectorielle et la multiplication avec la matrice de rotation en une seule opération équivalente qui consiste à multiplier une matrice de transformation 4×4 notée ${}^A T^B$, avec un vecteur-colonne 4×1 de *coordonnées homogènes*. Les coordonnées homogènes sont tout simplement les coordonnées cartésiennes pour les trois premières composantes et la valeur 1 pour la quatrième composante.

Le changement de repère décrit à la section 3.6.1 correspond alors à l'opération suivante :

$$\begin{bmatrix} r_{C/A_o}^a \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} {}^a R^b & r_{B_o/A_o}^a \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{{}^A T^B} \begin{bmatrix} r_{C/B_o}^b \\ 1 \end{bmatrix} \quad (3.114)$$

ce qui correspond en termes de coordonnées à :

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} {}^a R^b & r_{B_o/A_o}^a \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{{}^A T^B} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (3.115)$$

Les trois premières rangées des équations matricielles (3.114) et (3.115) correspondent exactement à l'équation (3.110), et la dernière ligne correspond simplement à une équation 1=1. Les transformations homogènes c'est donc rien de fondamentalement nouveau, mais une astuce bien utile dans un contexte de programmation pour grouper deux opérations en une seule.

Si on substitue les définitions, la matrice de transformation peut être définie directement en termes de vecteurs de vecteurs unitaires et du vecteur géométrique qui définit la translation d'une origine à l'autre :

$${}^A T^B = \begin{bmatrix} {}^a R^b & r_{B_o/A_o}^a \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \hat{a}_1 \cdot \hat{b}_1 & \hat{a}_1 \cdot \hat{b}_2 & \hat{a}_1 \cdot \hat{b}_3 & \hat{a}_1 \cdot \vec{r}_{B_o/A_o} \\ \hat{a}_2 \cdot \hat{b}_1 & \hat{a}_2 \cdot \hat{b}_2 & \hat{a}_2 \cdot \hat{b}_3 & \hat{a}_2 \cdot \vec{r}_{B_o/A_o} \\ \hat{a}_3 \cdot \hat{b}_1 & \hat{a}_3 \cdot \hat{b}_2 & \hat{a}_3 \cdot \hat{b}_3 & \hat{a}_3 \cdot \vec{r}_{B_o/A_o} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.116)$$

L'**opération inverse** de changement de repère avec les coordonnées homogènes est formée par :

$$\begin{bmatrix} r_{C/B_o}^b \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} {}^b R^a & -{}^b R^a r_{B_o/A_o}^a \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{{}^A T^B}^{-1} \begin{bmatrix} r_{C/A_o}^a \\ 1 \end{bmatrix} \quad (3.117)$$

Comme pour les matrices de rotation l'inverse de la matrice de transformation correspond à inverser les repères avec la notation utilisée :

$$({}^A T^B)^{-1} = {}^B T^A \quad (3.118)$$

Dernièrement, les **changements de repères successifs** $A \rightarrow B \rightarrow C \rightarrow \dots \rightarrow Z$ peuvent être combinés en une seule matrice de transformation en multipliant les matrices de transformation intermédiaires par la gauche :

$${}^C T^A = {}^C T^B {}^B T^A \quad (3.119)$$

$${}^D T^A = {}^D T^C {}^C T^B {}^B T^A \quad (3.120)$$

⋮

$${}^Z T^A = {}^Z T^Y {}^Y T^X \dots {}^C T^B {}^B T^A \quad (3.121)$$

Avec la notation utilisée dans ces notes, les **lettres des repères intermédiaires doivent être côté-à-côte dans les équations**, et la matrice de transformation totale conserve le premier et le dernier repère de la séquence dans le même ordre.

3.7 Représentation de la pose d'un corps rigide

De façon générale, six variables sont nécessaires pour définir la position d'un corps rigide par rapport à un repère, trois pour décrire la translation et trois pour décrire l'orientation. Pour un corps rigide sans aucune contrainte, par exemple un avion en vol, le nombre de DDL est donc de six.

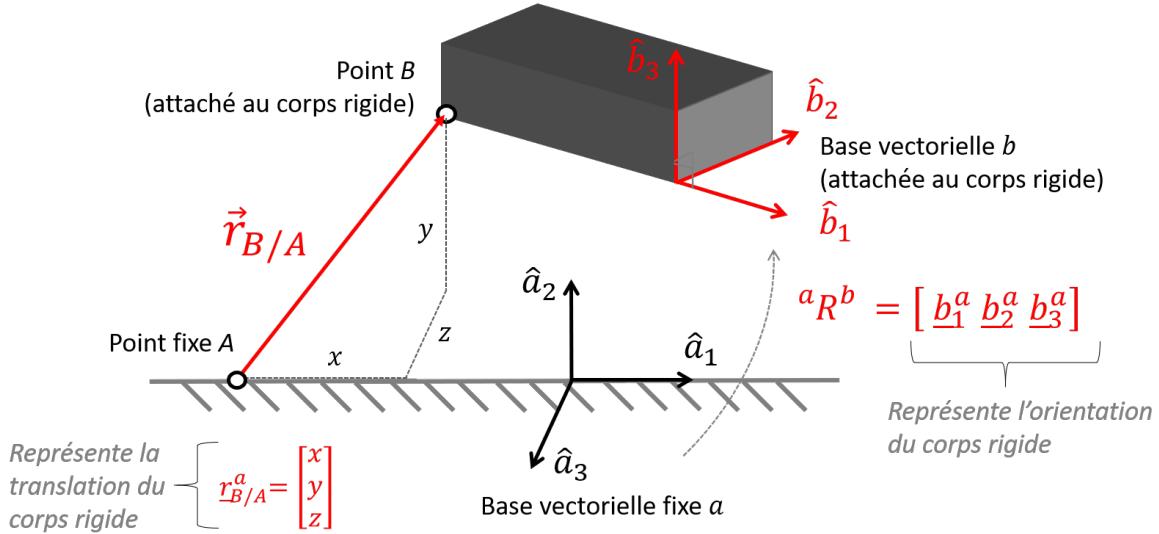


FIGURE 3.24 – Représentation de la pose d'un corps rigide



Capsule vidéo

Pose (position et orientation) d'un corps rigide

<https://youtu.be/wrw4T80E36U>

Translation : Les trois DDL de translation peuvent être représentés par un vecteur position qui a comme origine un point fixe A et comme cible un point arbitraire B sur le corps rigide. Ce vecteur position $\vec{r}_{B/A}$, est typiquement représenté avec trois coordonnées cartésiennes (x, y, z) qui utilisent une base vectorielle fixe a :

$$\underline{r}_{B/A}^a = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \Leftrightarrow \vec{r}_{B/A} = x \vec{a}_1 + y \vec{a}_2 + z \vec{a}_3 \quad (3.122)$$

Orientation : Ensuite, même avec un point défini dans l'espace par un vecteur translation, un corps rigide a plusieurs orientations possibles. Pour définir l'orientation, une autre base vectorielle b attachée au corps rigide doit être utilisée. L'orientation d'un corps rigide est alors représentée par la direction des vecteurs unitaires \hat{b}_i , qui peuvent être représentés par leurs composantes dans la base fixe a . Les trois vecteur-colonnes \underline{b}_i^a de trois composantes peuvent être regroupés sous la forme d'une matrice de rotation, comme il a été discuté à la section 3.5 dans le contexte des changements de base :

$${}^a R^b = \begin{bmatrix} \underline{b}_1^a & \underline{b}_2^a & \underline{b}_3^a \end{bmatrix} = R(\phi, \theta, \psi) \quad (3.123)$$

Malgré les neuf composants d'une matrice de rotation, puisque les vecteurs \hat{b}_i sont unitaires et orthogonaux seulement trois variables suffisent pour la définir ; il y a un total de neuf composantes et six équations de contraintes, voir les équations (17.4) et (17.5). Il est donc possible de paramétriser la matrice ${}^a R^b$ avec trois variables, par exemple trois angles ϕ, θ et ψ , sans limiter les orientations possibles. On appelle trois angles

qui paramétrisent une matrice de rotation des angles de Euler, diverses conventions de paramétrisation sont possibles et c'est le sujet de la section 3.7.1. Selon la situation, d'autres représentations alternatives aux matrices de rotation et aux angles de Euler peuvent aussi être utiles, chaque représentation a ses avantages et inconvénients. La section 3.7.2 présente la représentation axe-angle et la section 3.7.3 présente les quaternions.

Pose : La pose d'un corps rigide, c'est la combinaison de sa translation et de son orientation. Il est possible d'utiliser des représentations indépendantes pour la translation et l'orientation, mais alternativement on peut regrouper une matrice de rotation (pour représenter l'orientation) et un vecteur-colonne (pour représenter la translation) sous la forme d'une matrice de transformation 4x4, qui relie le repère $\{A_o, \hat{a}_1, \hat{a}_2, \hat{a}_3\}$ formé par le point fixe et la base fixe, au repère $\{B_o, \hat{b}_1, \hat{b}_2, \hat{b}_3\}$ formé par le point mobile et la base mobile attachés au corps rigide. Cette matrice peut alors être paramétrée par 6 variables qui représentent tous les DDL du corps rigide :

$${}^A T^B(x, y, z, \phi, \theta, \psi) = \begin{bmatrix} {}^a R^b & r_{B/A}^a \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} R(\phi, \theta, \psi) \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \end{bmatrix} \quad (3.124)$$

La matrice de transformation ${}^A T^B$ a donc deux fonctions : 1) elle peut être utilisée pour effectuer des changements de repères comme vue à la section 3.6.2, et 2) elle peut aussi être utilisée pour représenter la pose d'un repère B (donc aussi la pose du corps rigide sur lequel il est attaché) relativement à un repère A .

3.7.1 Angles de Euler

Une matrice de rotation arbitraire peut être paramétrée avec trois variables. La méthode la plus simple pour y parvenir est de construire une matrice de rotation comme une multiplication de trois matrices élémentaires. Par exemple, l'orientation arbitraire d'une base vectorielle d par rapport à une base vectorielle a peut être décrite par les angles (ϕ, θ, ψ) , avec la convention 3-1-3 :

$${}^a R^d(\phi, \theta, \psi) = R_3(\phi) R_1(\theta) R_3(\psi) \quad (3.125)$$

$${}^a R^d(\phi, \theta, \psi) = \begin{bmatrix} c\phi & -s\phi & 0 \\ s\phi & c\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\theta & -s\theta \\ 0 & s\theta & c\theta \end{bmatrix} \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.126)$$

$${}^a R^d(\phi, \theta, \psi) = \begin{bmatrix} +c\phi c\psi - s\phi c\theta c\psi & -c\phi s\psi - s\phi c\theta c\psi & +s\phi s\theta \\ +s\phi c\psi + c\phi c\theta c\psi & -s\phi s\psi + c\phi c\theta c\psi & -c\phi s\theta \\ +s\theta s\psi & +s\theta c\psi & +c\theta \end{bmatrix} \quad (3.127)$$

Cette matrice peut être interprétée comme une suite de trois rotations, avec des bases vectorielles intermédiaires, illustrées à la figure 3.25.

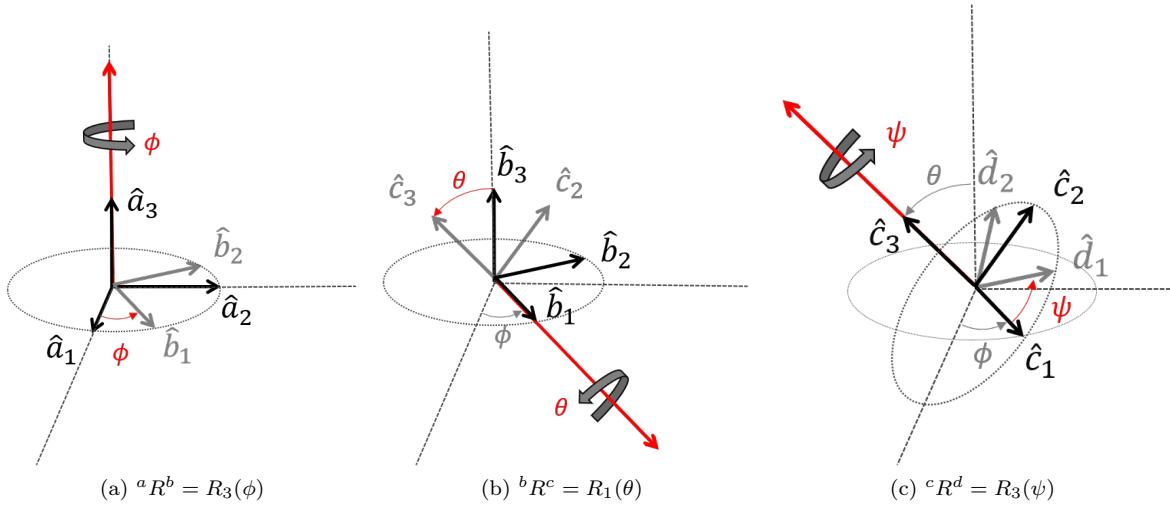


FIGURE 3.25 – Angles de Euler avec la convention 3-1-3 : toutes les orientations possibles de la base d peuvent être décrites par la matrice de rotation ${}^aR^d(\phi, \theta, \psi) = R_3(\phi)R_1(\theta)R_3(\psi)$ paramétrisée avec trois angles.

Une autre convention de trois angles utilisée fréquemment en robotique (typiquement pour les véhicules) sont les angles de roulis, tangage et lacet. Ces angles correspondent à une matrice formée par la multiplication de matrices élémentaires dans un ordre 3-2-1, si l'axe 1 correspond à la direction avant-arrière du véhicule et l'axe 2 vers la direction haut-bas du véhicule :

$$R_{321}(\theta_{tangage}, \theta_{lacet}, \theta_{roulis}) = R_3(\theta_{tangage}) R_2(\theta_{lacet}) R_1(\theta_{roulis}) \quad (3.128)$$

Comme l'ordre des matrices de rotation qui sont multipliées influence la matrice résultante, il y a plusieurs possibilités de définitions des trois angles de Euler qui paramétrisent la matrice. Différents auteurs, domaines et organisations utilisent des définitions différentes, il est donc toujours **important de vérifier la convention utilisée si on travail avec les angles de Euler**.

3.7.2 Représentation axe-angle

Une orientation relative entre deux bases vectorielles peut aussi être représentée par un axe de rotation, décrit par un vecteur unitaire \hat{a} , et un angle θ de rotation autour de celui-ci, comme illustré à la figure 3.26.

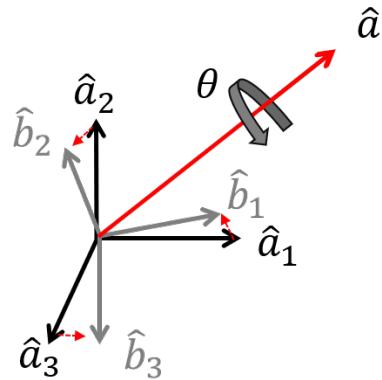


FIGURE 3.26 – Représentation axe-angle d'une orientation relative

Cette représentation nécessite quatre paramètres, trois pour un vecteur-colonne \underline{a} qui représente les composantes du vecteur unitaire qui décris l'axe de rotation, et un pour l'angle θ . La représentation axe-angle permet donc d'utiliser seulement 4 variables plutôt que les 9 d'une matrice 3×3 pour représenter une

rotation dans un espace tridimensionnel. Un paramètre est toutefois redondant, car le vecteur-colonne \underline{a} est unitaire et trois variables sont reliés par l'équation $a_1^2 + a_2^2 + a_3^2 = 1$. De plus, cette représentation présente des singularités lorsque $\sin(\theta) = 0$. L'avantage principal de la représentation axe-angle est le sens physique clair. Les inconvénients sont qu'il faut convertir vers des matrices de rotation (ou alternativement des quaternions) pour effectuer des calculs numériques de changement de base et pour combiner des rotations successives.

Vecteur propre de la matrice de rotation

L'axe de rotation \hat{a} est directement lié à la matrice de rotation qui représente la même orientation relative entre deux bases vectorielles. Le vecteur unitaire \hat{a} aligné avec l'axe de rotation a les mêmes composantes dans les deux bases vectorielles :

$$\underline{a} = \underline{a}^b = {}^bR^a\underline{a}^a = {}^aR^b\underline{a}^b = \underline{a}^a \quad (3.129)$$

Le vecteur-colonne qui représente **l'axe de rotation est donc un vecteur propre de la matrice de rotation**, et la valeur propre associée est de valeur unitaire car le vecteur-colonne est inchangé par la multiplication avec la matrice de rotation. Il est donc possible de calculer l'axe de rotation associé à une matrice de rotation en calculant les valeurs et vecteurs propre de la matrice.

Conversions

La représentation axe-angle peut être calculée à partir d'une matrice de rotation R de la façon suivante :

$$\theta = \cos^{-1} \left[\frac{R_{11} + R_{22} + R_{33} - 1}{2} \right] \quad \underline{a} = \frac{1}{2\sin(\theta)} \begin{bmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{bmatrix} \quad (3.130)$$

où

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \quad (3.131)$$

Inversement, la matrice de rotation peut être déterminée à partir de la représentation axe-angle :

$$R = \cos(\theta) I + (1 - \cos(\theta)) \underline{a} \underline{a}^T - \sin(\theta) \underline{a}^\times \quad (3.132)$$

où

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \underline{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad \underline{a}^T = [a_1 \ a_2 \ a_3] \quad \underline{a}^\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \quad (3.133)$$

3.7.3 Quaternions

Les quaternions sont une autre façon de représenter une orientation avec 4 paramètres, physiquement moins représentative mais beaucoup plus pratique mathématiquement. Similairement à la représentation axe-angle, un quaternion est constitué d'un scalaire et de 3 composantes vectorielles qui peuvent être groupées dans un vecteur colonne $[\eta, e_1, e_2, e_3]^T$. Les quatre paramètres sont reliés par l'équation suivante $\eta^2 + e_1^2 + e_2^2 + e_3^2 = 1$. Cette représentation se dérive facilement à partir de la représentation axe-angle.

$$\eta = \cos \left(\frac{\theta}{2} \right) \quad (3.134)$$

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \sin \left(\frac{\theta}{2} \right) \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad (3.135)$$

Les grands avantages de cette représentation sont **1)** l'absence de singularité et **2)** des méthodes numériquement efficaces pour calculer des rotations successives et des changements de base directement avec le vecteur colonne $[\eta, e_1, e_2, e_3]^T$.

Détails à venir !

Exemple 7. Cinématique d'un robot mobile

La figure illustre un exemple où un véhicule et les points d'intérêts qu'il trouve doivent être localisés par rapport à une base. La cinématique de ce problème peut se résumer à calculer la matrice de transformation ${}^A T^B$, qui peut être construite à partir de la définition :

$${}^A T^B = \begin{bmatrix} {}^a R^b & \vec{r}_{B_o/A_o}^a \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \hat{a}_1 \cdot \hat{b}_1 & \hat{a}_1 \cdot \hat{b}_2 & \hat{a}_1 \cdot \hat{b}_3 & \hat{a}_1 \cdot \vec{r}_{B_o/A_o} \\ \hat{a}_2 \cdot \hat{b}_1 & \hat{a}_2 \cdot \hat{b}_2 & \hat{a}_2 \cdot \hat{b}_3 & \hat{a}_2 \cdot \vec{r}_{B_o/A_o} \\ \hat{a}_3 \cdot \hat{b}_1 & \hat{a}_3 \cdot \hat{b}_2 & \hat{a}_3 \cdot \hat{b}_3 & \hat{a}_3 \cdot \vec{r}_{B_o/A_o} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.136)$$

Cette matrice représente la pose du véhicule par rapport au repère de la base, et peut aussi être utilisée pour transformer des coordonnées cartésiennes relatives au repère du véhicule en des coordonnées cartésiennes relatives au repère de la base :

$$\begin{bmatrix} \vec{r}_{C/A_o}^a \\ 1 \end{bmatrix} = {}^A T^B \begin{bmatrix} \vec{r}_{C/B_o}^b \\ 1 \end{bmatrix} \quad (3.137)$$

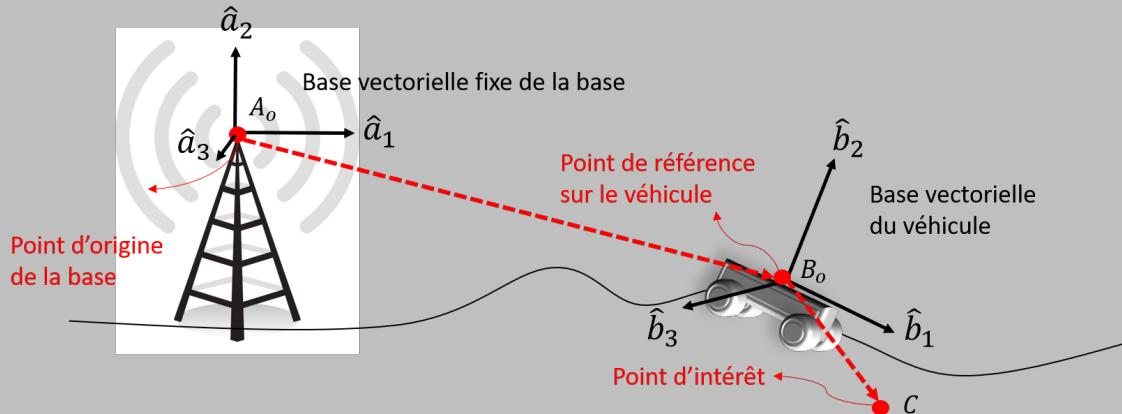


FIGURE 3.27 – Cinématique d'un robot mobile

3.8 Cinématique directe d'un manipulateur

La cinématique directe, c'est le calcul de la fonction de transformation qui permet de passer de l'espace des joints, i.e. le vecteur-colonne \underline{q} , vers l'espace de la tâche. Pour les robots manipulateurs, l'espace de la tâche est normalement décrit par la translation et l'orientation de l'effecteur. Si le point de référence de l'effecteur est noté T_o (souvent dans la littérature il est appelé le *TCP* pour *Tool Center Point*), et la base vectorielle associée à l'orientation de l'effecteur est notée t , voir figure 3.30, alors la fonction de cinématique directe pour la translation est :

$$\vec{r}_{T_o/A_o}^a = f_{Trans}(\underline{q}) \quad (3.138)$$

et la fonction de cinématique directe pour l'orientation est :

$${}^aR^t = f_{Orien}(\underline{q}) \quad (3.139)$$

où le point A_o et la base a décrivent l'origine et la base vectorielle globale qui sont fixes par rapport à la base du robot. Les deux opérations peuvent être combinées en une seule opération qui consiste à calculer la matrice de transformation homogène du repère $\{T_o, \hat{t}_1, \hat{t}_2, \hat{t}_3\}$ par rapport au repère de global $\{A_o, \hat{a}_1, \hat{a}_2, \hat{a}_3\}$

$${}^AT^T = f_{Pose}(\underline{q}) \quad (3.140)$$

3.8.1 Chaîne cinématique ouverte

La plupart des robots manipulateurs sont caractérisés par une chaîne cinématique ouverte de n joints, comme illustré à la figure 3.28. Pour modéliser cette chaîne cinématique, on associera des points de référence et des bases vectorielles sur chaque lien rigide. La fonction de **cinématique directe pour la translation** se résume à calculer l'addition des $n + 1$ vecteurs géométriques de position qui caractérisent la chaîne cinématique, comme illustré à la figure 3.29 :

$$\vec{r}_{T_o/A_o} = \vec{r}_{T_o/H_o} + \dots + \vec{r}_{C_o/B_o} + \vec{r}_{B_o/A_o} \quad (3.141)$$

Ce calcul peut être effectué en termes de composantes dans la base vectorielle globale :

$$\underline{r}_{T_o/A_o}^a = \underline{r}_{T_o/H_o}^a + \dots + \underline{r}_{C_o/B_o}^a + \underline{r}_{B_o/A_o}^a \quad (3.142)$$

La fonction de **cinématique directe pour l'orientation** de l'effecteur se résume à multiplier les $n + 1$ matrices de rotation qui caractérisent les orientations relatives de deux joints successifs, pour obtenir la matrice de rotation totale qui décrit l'orientation de l'effecteur par rapport à la base globale :

$${}^aR^t = {}^aR^b {}^bR^c \dots {}^hR^t \quad (3.143)$$

Alternativement pour combiner les opérations, la fonction de **cinématique directe pour la pose** de l'effecteur peut être calculée en multipliant les $n+1$ matrices de transformations homogènes qui caractérisent la pose relative des repères associés à deux joints successifs :

$${}^AT^T = {}^AT^B {}^BT^C \dots {}^HT^T \quad (3.144)$$

3.8.2 Simplifications pour les chaînes à 1 DDL par joint

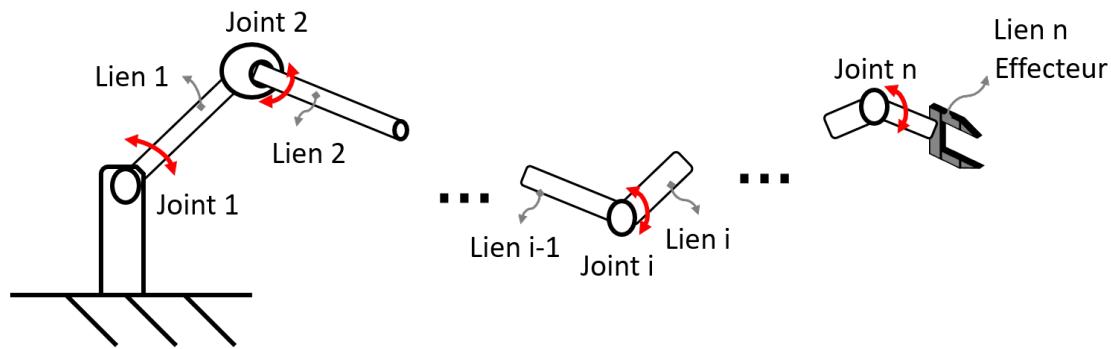
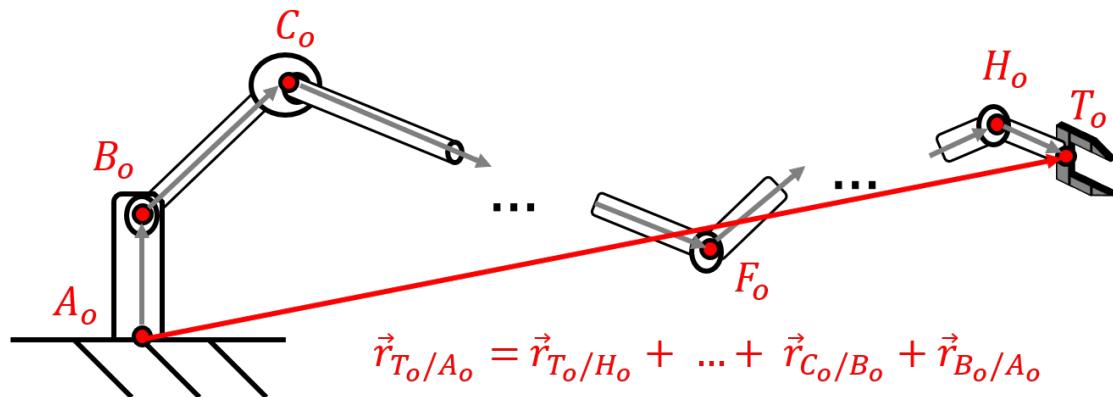
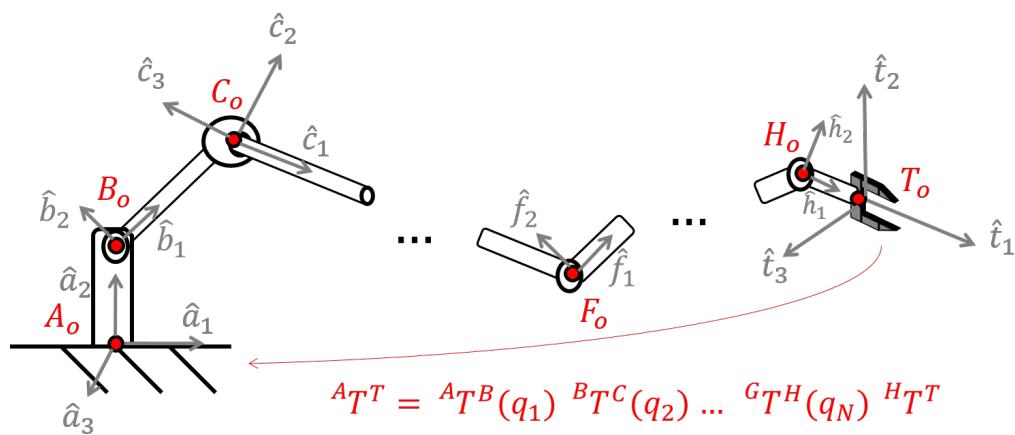
Pour une chaîne cinématique où chaque joint a un seul DDL qui est décrit par la variable q_i , l'équation (3.144) de la cinématique directe avec les coordonnées homogènes a la structure suivante :

$${}^AT^T(q_1, q_2, \dots, q_n) = {}^AT^B(q_1) {}^BT^C(q_2) \dots {}^GT^H(q_n) {}^HT^T \quad (3.145)$$

Chaque matrice de transformation est une fonction d'une seule variable q_i . Notez que ici la dernière matrice de transformation ${}^HT^T$ est constante car elle décrit la position de l'effecteur par rapport au dernier joint n . De plus, si un robot est constitué de **jumlahs rotatifs seulement**, comme la plupart des robots manipulateurs industriels, alors l'équation pour l'orientation (3.143) a aussi une structure similaire :

$${}^aR^t(\theta_1, \theta_2, \dots, \theta_n) = {}^aR^b(\theta_1) {}^bR^c(\theta_2) \dots {}^gR^h(\theta_n) {}^hR^t \quad (3.146)$$

où chaque matrice de rotation est une fonction de la variable q_i , qui est un angle θ_i .

FIGURE 3.28 – Cinématique d'une chaîne ouverte de n jointsFIGURE 3.29 – Cinématique d'une chaîne ouverte de n joints : addition vectorielle de $n+1$ vecteurs géométriques de position.FIGURE 3.30 – Cinématique d'une chaîne ouverte de n joints : succession de $n+1$ transformations homogènes

3.8.3 Transformations relatives entre les joints à 1 DDL

Pour un joint rotatif, le DDL est une rotation relative selon un axe. Le mouvement peut donc être représenté par une matrice de rotation variable qui décrit l'orientation relative des deux liens rigides reliés par le joint. Pour un joint prismatique, il n'y a aucune rotation relative, seulement une translation selon un axe. La mouvement est donc représenté par un vecteur position de longueur variable. Donc pour représenter le mouvement du joint i d'un robot, qui est décrit par la pose relative entre le repère $\{F_o, \hat{f}_1, \hat{f}_2, \hat{f}_3\}$ et le repère $\{E_o, \hat{e}_1, \hat{e}_2, \hat{e}_3\}$, c'est le vecteur position \vec{r}_{F_o/E_o} qui est une fonction de q_i pour un joint prismatique et la matrice ${}^eR^f$ qui est une fonction de q_i pour un joint rotatif :

$$\begin{array}{ll} \text{Joint prismatique : } & \left\{ \begin{array}{l} \underline{r}_{F_o/E_o}^e = f(q_i) \\ {}^eR^f = \text{constante} \end{array} \right. & \text{Joint rotatif : } & \left\{ \begin{array}{l} \underline{r}_{F_o/E_o}^e = \text{constante} \\ {}^eR^f = f(q_i) \end{array} \right. \end{array} \quad (3.147)$$

de façon équivalente en termes des transformations homogènes :

$$\text{Prismatique : } {}^E T^F(q_i) = \begin{bmatrix} {}^eR^f & \underline{r}_{F_o/E_o}^a(q_i) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Rotatif : } {}^E T^F(q_i) = \begin{bmatrix} {}^eR^f(q_i) & \underline{r}_{F_o/E_o}^a \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.148)$$

3.8.4 Procédure de calcul d'une chaîne cinématique directe

Une procédure pour calculer la cinématique d'un chaîne ouverte arbitraire de n joints et $n + 1$ liens rigides, est ici présentée. La notation utilisée est présentée aux figures 3.28, 3.29 et 3.30.



Capsule vidéo
Cinématique directe d'un robot manipulateur
<https://youtu.be/qgUkhZxMTpM>

I : Définition des références La première étape est de définir des coordonnées généralisées (q_1, q_2, \dots, q_m) qui décrive l'état des m DDL des n joints du robot. En générale, pour la plupart des robots manipulateurs, $m = n$ car des joints standards à 1 DDL prismatique ou rotatif sont normalement utilisés. Ensuite, la seconde étape est de définir des points de références ($A_o, B_o, C_o, \dots, T_o$) et des bases vectorielles (a, b, c, \dots, t) sur chaque lien rigide.

Astuces : Pour grandement simplifier le calcul des matrices de rotation, les bases vectorielles locales de deux liens rigides adjacents à un joint rotatif devraient être définies avec un vecteurs unitaire parallèle à l'axe de rotation du joint. Ensuite, l'orientation des deux autres vecteurs unitaires devrait être choisie de façon à maximiser le nombre de composantes nulles dans les vecteurs positions.

II : Calcul des vecteurs positions locaux Ensuite, basé sur les points et bases vectorielles définies, les $n + 1$ vecteurs positions locaux peuvent être calculés par inspection selon la géométrie et les dimensions du robot. Pour un robot qui est constitué de joints rotatifs seulement, toutes les composantes de ces vecteurs positions exprimées dans les bases vectorielles locales sont constantes :

$$\underline{r}_{B_o/A_o}^a = \begin{bmatrix} l_{1,x} \\ l_{1,y} \\ l_{1,z} \end{bmatrix}, \quad \underline{r}_{C_o/B_o}^b = \begin{bmatrix} l_{2,x} \\ l_{2,y} \\ l_{3,z} \end{bmatrix}, \quad \dots \quad \underline{r}_{T_o/H_o}^h = \begin{bmatrix} l_{n+1,x} \\ l_{n+1,y} \\ l_{n+1,z} \end{bmatrix} \quad (3.149)$$

où les variables l représentent les dimensions des liens rigides. Toutefois, si un robot utilise des joints prismatiques, certains vecteurs positions locaux vont être des fonctions des variables q_i représentants l'état de ces joints prismatiques.

III : Calcul des matrices de rotations relatives Ensuite, les matrices de rotation qui décrivent l'orientation relative des bases vectorielles de joints adjacents sont calculées en suivant la démarche décrite à la section 3.5.2 :

$${}^a R^b(q_1) = \begin{bmatrix} b_1^a & b_2^a & b_3^a \end{bmatrix}, \quad {}^b R^c(q_2) = \begin{bmatrix} c_1^b & c_2^b & c_3^b \end{bmatrix}, \quad \dots \quad (3.150)$$

Pour un robot qui est constitué de joints rotatifs seulement, les matrices de rotation relatives entre deux bases locales vont être des fonctions des variables q_i qui décrivent l'état des joints rotatifs.

IV : Calcul des translations et orientations absolues Premièrement, les matrices de rotation qui décrivent l'orientation des bases vectorielles mobiles par rapport à la base vectorielle fixe a sont calculées en multipliant les matrices de rotation relatives :

$${}^a R^b = {}^a R^b \quad (3.151)$$

$${}^a R^c = {}^a R^b {}^b R^c \quad (3.152)$$

$${}^a R^d = {}^a R^b {}^b R^c {}^c R^d \quad (3.153)$$

⋮

$${}^a R^t = {}^a R^b {}^b R^c {}^c R^d \dots {}^h R^t \quad (3.154)$$

où la dernière ligne donne le résultat de cinématique directe pour l'orientation. Ensuite, avec ces matrices, il est possible d'effectuer des changements de base pour exprimer tous les vecteurs de position dans la base fixe a :

$$\underline{r}_{B_o/A_o}^a = \underline{r}_{B_o/A_o}^a \quad (3.155)$$

$$\underline{r}_{C_o/B_o}^a = {}^a R^b \underline{r}_{C_o/B_o}^b \quad (3.156)$$

$$\underline{r}_{D_o/C_o}^a = {}^a R^c \underline{r}_{D_o/C_o}^c \quad (3.157)$$

⋮

$$\underline{r}_{T_o/H_o}^a = {}^a R^h \underline{r}_{T_o/H_o}^h \quad (3.158)$$

Finalement, avec tous les vecteurs de position relatifs connus dans la base fixe a , il est maintenant possible de calculer la position de tous les points de références par rapport à l'origine A_o

$$\underline{r}_{B_o/A_o}^a = \underline{r}_{B_o/A_o}^a \quad (3.159)$$

$$\underline{r}_{C_o/A_o}^a = \underline{r}_{C_o/B_o}^a + \underline{r}_{B_o/A_o}^a \quad (3.160)$$

$$\underline{r}_{D_o/A_o}^a = \underline{r}_{D_o/C_o}^a + \underline{r}_{C_o/B_o}^a + \underline{r}_{B_o/A_o}^a \quad (3.161)$$

⋮

$$\underline{r}_{T_o/A_o}^a = \underline{r}_{T_o/H_o}^a + \dots + \underline{r}_{C_o/B_o}^a + \underline{r}_{B_o/A_o}^a \quad (3.162)$$

où la dernière ligne donne le résultat de cinématique directe pour la translation de l'effaceur. Il est à noter que cette procédure donne aussi la position de tous les points de références ($A_o, B_o, C_o, \dots, T_o$) dans le repère fixe A , ce qui est utile pour tracer le squelette du robot et visualiser la configuration.

IV* : Alternative de calcul avec les transformations homogènes Alternativement, l'étape IV peut être effectuée grâce aux transformations homogènes. Les points de références et bases vectorielles sur chaque lien rigide définissent des repères. Les matrices de transformation relative à deux repères adjacents à un joint peuvent être construites à partir des vecteurs positions locaux et des matrices de rotation relatives :

$${}^A T^B(q_1) = \begin{bmatrix} {}^a R^b(q_1) & \underline{r}_{B_o/A_o}^a \\ 0 & 1 \end{bmatrix}, \quad {}^B T^C(q_2) = \begin{bmatrix} {}^b R^c(q_2) & \underline{r}_{C_o/B_o}^b \\ 0 & 1 \end{bmatrix}, \quad \dots \quad (3.163)$$

Les matrices de transformation globales de tous les repères mobiles vers le repère fixe A peuvent ensuite être calculées par la multiplication des matrices de transformation intermédiaires :

$${}^A T^B = {}^A T^B(q_1) \quad (3.164)$$

$${}^A T^C = {}^A T^B(q_1) {}^B T^C(q_2) \quad (3.165)$$

⋮

$${}^A T^T = {}^A T^B(q_1) {}^B T^C(q_2) \dots {}^G T^H(q_n) {}^H T^T \quad (3.166)$$

Exemple 8. Cinématique pour un robot avec deux joints

Un exemple est ici présenté de calcul de la cinématique directe pour le robot illustré à la figure 3.31 qui possède un joint rotatif et un joint prismatique. L'objectif est ici de calculer les coordonnées du point D_o dans le repère fixe $\{A_o, \hat{a}_1, \hat{a}_2, \hat{a}_3\}$ comme une fonction de la configuration des joints. Trois méthodes de résolution sont présentées, une méthode minimalistre qui utilise seulement les équations vectorielles, une méthode matricielle qui utilise les vecteur-colonnes et les matrices de rotation, ainsi qu'une méthode utilisant les coordonnées homogènes.

La première étape, commune aux trois méthodes, est de définir des bases vectorielles et des points de référence sur le robot comme illustré à la figure 3.31b. Ici les bases vectorielles a et b ont été définies de sorte que les vecteurs unitaires \hat{a}_3 et \hat{b}_3 soient parallèles à l'axe de rotation du joint rotatif (axe perpendiculaire au plan de la page). De plus, le vecteur unitaire \hat{b}_1 a été défini en ligne avec le deuxième lien rigide et l'axe de translation du joint prismatique. Les points de référence secondaires ont été choisis ainsi : un point B_o au centre du joint rotatif et un point C à la base du joint prismatique. Pour les coordonnées généralisées qui décrivent la configuration du robot, l'angle θ avec la vertical du deuxième lien rigide ainsi que la distance x de translation entre le point C et D_o ont été choisis.

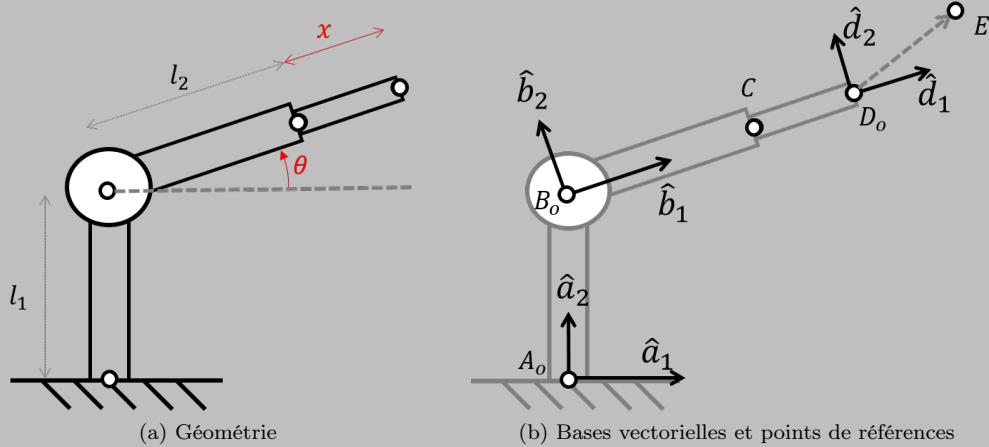


FIGURE 3.31 – Robot à deux joints

Méthode vectorielle Premièrement, avec les points et bases qui ont été définis, il est possible de décrire les vecteurs positions qui représentent la géométrie des liens rigides et ne dépendent pas directement des joints :

$$\text{Éléments constants : } \vec{r}_{B_o/A_o} = l_1 \hat{a}_2 \quad \text{et} \quad \vec{r}_{C/B_o} = l_2 \hat{b}_1 \quad (3.167)$$

Ensuite, la direction \hat{b}_1 dépend de l'état du joint rotatif donc de l'angle θ , et le vecteur position du point D_o par rapport au point C a une longueur variable égale à x :

$$\text{Éléments variables : } \hat{b}_1(\theta) = c\theta \hat{a}_1 + s\theta \hat{a}_2 \quad \text{et} \quad \vec{r}_{D_o/C}(x) = x \hat{b}_1 \quad (3.168)$$

Tous les éléments sont maintenant en place pour effectuer la résolution. Le vecteur pertinent pour la résolution est décomposé en une addition de vecteurs connus :

$$\vec{r}_{D_o/A_o} = \vec{r}_{D_o/C} + \vec{r}_{C/B_o} + \vec{r}_{B_o/A_o} \quad (3.169)$$

qui peuvent être manipulés de sorte à garder seulement les vecteurs unitaires \hat{a}_i :

$$\vec{r}_{D_o/A_o} = x \hat{b}_1 + l_2 \hat{b}_1 + l_1 \hat{a}_2 \quad (3.170)$$

$$\vec{r}_{D_o/A_o} = (x + l_2) \hat{b}_1 + l_1 \hat{a}_2 \quad (3.171)$$

$$\vec{r}_{D_o/A_o} = (x + l_2) (c\theta \hat{a}_1 + s\theta \hat{a}_2) + l_1 \hat{a}_2 \quad (3.172)$$

$$\vec{r}_{D_o/A_o} = \underbrace{[(x + l_2) c\theta]}_{r_1^a} \hat{a}_1 + \underbrace{[(x + l_2) s\theta + l_1]}_{r_2^a} \hat{a}_2 \quad (3.173)$$

Puisque l'équation vectorielle a été réduite aux directions \hat{a}_i seulement, il est possible d'extraire directement les composantes dans la base a :

$$\underline{r}_{D_o/A_o}^a = \begin{bmatrix} (x + l_2) c\theta \\ (x + l_2) s\theta + l_1 \\ 0 \end{bmatrix} \quad (3.174)$$

Méthode matricielle Premièrement, avec les points et bases qui ont été définis, il est possible de décrire des vecteurs de position locaux constants. Les vecteur-colonnes $\underline{r}_{B_o/A_o}^a$ et \underline{r}_{C/B_o}^b représentent la géométrie des liens rigides et sont donc constants :

$$\text{Éléments constants : } \underline{r}_{B_o/A_o}^a = \begin{bmatrix} 0 \\ l_1 \\ 0 \end{bmatrix} \quad \text{et} \quad \underline{r}_{C/B_o}^b = \begin{bmatrix} l_2 \\ 0 \\ 0 \end{bmatrix} \quad (3.175)$$

Toutefois, la matrice de rotation ${}^aR^b$ qui décrit l'orientation relative des bases a et b est variable en fonction de l'état du joint rotatif décrit par la variable θ , et le vecteur position $\underline{r}_{D_o/C}^b$ est associé à la translation du joint prismatique et fonction de la variable x :

$$\text{Éléments variables : } {}^aR^b(\theta) = R_3(\theta) = \begin{bmatrix} c\theta & -s\theta & 0 \\ s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{et} \quad \underline{r}_{D_o/C}^b(x) = \begin{bmatrix} x \\ 0 \\ 0 \end{bmatrix} \quad (3.176)$$

La matrice de rotation peut être construite comme décrit à la section 3.5.2, alternativement il est possible d'identifier directement qu'elle correspond ici à une matrice de rotation élémentaire selon l'axe 3.

L'équation de la cinématique pour calculer la position de l'effaceur est obtenue à partir de la relation vectorielle :

$$\vec{r}_{D_o/A_o} = \vec{r}_{D_o/C} + \vec{r}_{C/B_o} + \vec{r}_{B_o/A_o} \quad (3.177)$$

que l'on exprime dans la base a :

$$\underline{r}_{D_o/A_o}^a = \underline{r}_{D_o/C}^a + \underline{r}_{C/B_o}^a + \underline{r}_{B_o/A_o}^a \quad (3.178)$$

$$\underline{r}_{D_o/A_o}^a = {}^a R^b \underline{r}_{D_o/C}^b + {}^a R^b \underline{r}_{C/B_o}^b + \underline{r}_{B_o/A_o}^a \quad (3.179)$$

$$\underline{r}_{D_o/A_o}^a = \underbrace{{}^a R^b(\theta)}_{\text{joint rotatif}} \left(\underbrace{\underline{r}_{D_o/C}^b(x)}_{\text{joint prismatique}} + \underline{r}_{C/B_o}^b \right) + \underline{r}_{B_o/A_o}^a \quad (3.180)$$

Il est ensuite possible de substituer par les composantes connues et résoudre l'équation matricielle :

$$\underline{r}_{D_o/A_o}^a = \begin{bmatrix} c\theta & -s\theta & 0 \\ s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} x \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} l_2 \\ 0 \\ 0 \end{bmatrix} \right) + \begin{bmatrix} 0 \\ l_1 \\ 0 \end{bmatrix} = \begin{bmatrix} (x + l_2)c\theta \\ (x + l_2)s\theta + l_1 \\ 0 \end{bmatrix} \quad (3.181)$$

Méthode avec les transformations homogènes *Alternativement, une façon plus systématique de procéder, est d'associer un repère à chaque lien rigide et de calculer toutes les matrices de transformation entre ces repères.*

La matrice de transformation du repère B vers le repère A implique une matrice de rotation variable selon l'état du joint rotatif, et un vecteur de translation constant :

$${}^A T^B(\theta) = \begin{bmatrix} {}^a R^b(\theta) & \underline{r}_{B_o/A_o}^a \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta & -s\theta & 0 & 0 \\ s\theta & c\theta & 0 & l_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.182)$$

La matrice de transformation du repère D vers le repère B implique seulement un vecteur de translation variable en fonction de l'état du joint prismatique. Puisque que les bases vectorielles d et b sont alignées la matrice de rotation ${}^b R^d$ est réduite à la matrice identité :

$${}^B T^D(x) = \begin{bmatrix} {}^b R^d & \underline{r}_{D_o/B_o}^b \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} I_{3 \times 3} & (\underline{r}_{D_o/C}^b(x) + \underline{r}_{C/B_o}^b) \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & (x + l_2) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.183)$$

Ensuite la pose de l'effecteur peut être décrite par la matrice de transformation du repère D vers le repère A qui est obtenu en multipliant les deux matrices de transformation locales de chaque joint :

$${}^A T^D = {}^A T^B(\theta) {}^B T^D(x) \quad (3.184)$$

$${}^A T^D = \begin{bmatrix} c\theta & -s\theta & 0 & 0 \\ s\theta & c\theta & 0 & l_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & (x + l_2) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.185)$$

$${}^A T^D = \begin{bmatrix} c\theta & -s\theta & 0 & (x + l_2)c\theta \\ s\theta & c\theta & 0 & (x + l_2)s\theta + l_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.186)$$

Avec la dernière colonne de la matrice ${}^A T^D$, il est possible d'extraire les composantes qui représentent la translation du point D_o dans le repère A :

$$\underline{r}_{D_o/A_o}^a = \begin{bmatrix} (x + l_2)c\theta \\ (x + l_2)s\theta + l_1 \\ 0 \end{bmatrix} \quad (3.187)$$

3.8.5 Les paramètres Denavit–Hartenberg

Les paramètres *DH* (*Denavit–Hartenberg*) consistent en quatre paramètres associés à une convention particulière pour attacher des repères aux liens rigides d'un robot manipulateur. Plutôt que de définir arbitrairement les bases vectorielles et les points d'origine des repères, la convention impose une méthodologie standardisée pour définir les repères et les matrices de transformations associées. Plusieurs librairies de cinématique de robot utilise cette convention, qui a comme principal avantage de décrire la cinématique d'un robot avec un nombre de paramètres et repères minimum.

Capsule vidéo
Les paramètres DH
<https://youtu.be/LT5Fa1tbDFQ>

Convention de positionnement relatif des repères

Avec la convention *DH*, il faut placer les axes 3 (ou z) des repères sur les axe de rotation des joints révolus et sur les axes de translation des joints prismatiques. Ensuite les axes 1 (ou X) d'un joint sont positionnés relativement à la position du joint précédent (parallèle à la normale commune des axes 3), et ensuite les axes 2 sont placés selon la règle de la main droite. L'axe 1 du premier repère est un choix arbitraire. Il y a aussi plusieurs options lorsque les axes 3 de deux joints consécutifs sont parallèles ou coïncident, donc la convention de permet pas de standardisé la représentation de la cinématique d'un à 100%. La Figure 3.32 illustre cette convention de définition des repères d'un robot.

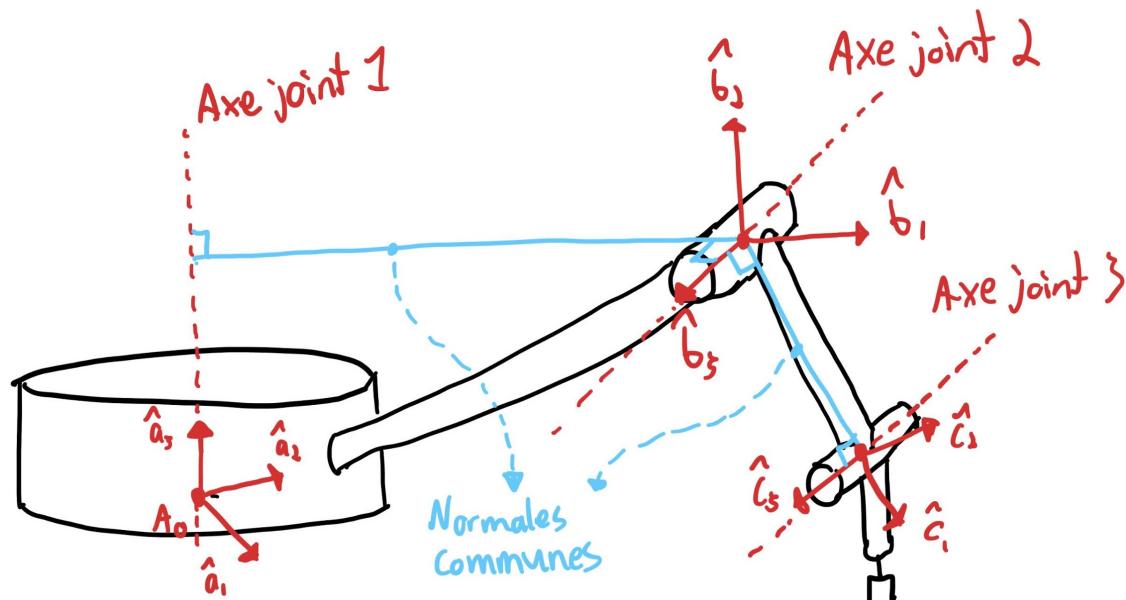
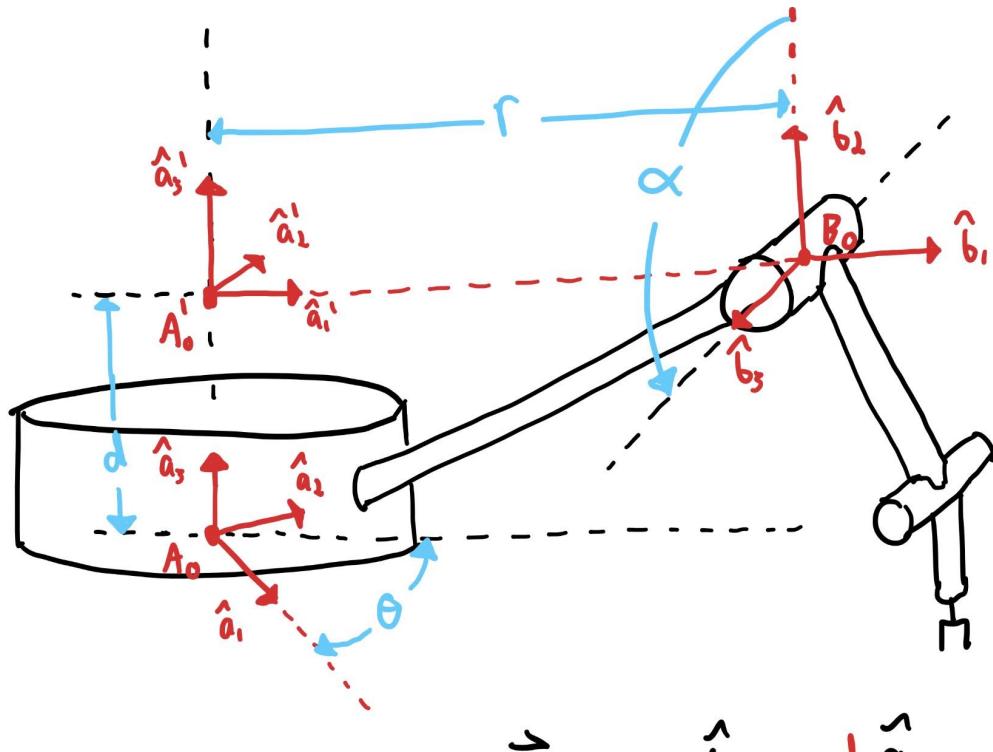


FIGURE 3.32 – Positionnement des repères selon la convention des paramètres *DH*

Repère attachés au lien rigide précédent Il est à noter que avec cette convention, chaque repère est attaché au lien rigide précédent le joint associé à son axe 3, ce qui diffère de la méthode présentée à la section 3.8.4. Par exemple à la figure 3.32, le repère A est fixe et le repère B bouge seulement en fonction de la rotation du 1er joint autour de l'axe \hat{a}_3 .

Les quatre paramètres DH

Avec la convention de positionnement des repères, seuls quatre scalaires (deux distance et deux angles) sont suffisants pour décrire la pose relative des deux repères sur deux joints adjacents. Les quatre paramètres sont illustrés à la Figure 3.33, pour le premier joint d'un robot où le repère A est fixe et aligné avec l'axe du joint 1, et le repère B bouge avec le premier lien rigide et est aligné avec le joint 2. Un repère intermédiaire A' est défini à l'intersection de la normale commune et l'axe 3 du repère A pour bien illustrer le sens des 4 paramètres. Le paramètre d représente la distance entre le point d'origine du premier repère A_o et l'intersection de l'axe \hat{a}_3 avec la normale commune (A'_o). Le paramètre θ est l'angle entre le vecteur unitaire \hat{a}_1 et la normale commune. Le paramètre r est la longueur de la normale commune. Le paramètre α est l'angle de rotation autour de la normale entre les deux axes des joints \hat{a}_3 et \hat{b}_3 .



$$\begin{aligned} \text{DH Params} \quad & \vec{r}_{B/A_0} = r \hat{b}_1 + d \hat{a}_3 \\ d, \theta, r, \alpha \quad & {}^a R {}^b = {}^a R_s(\theta) {}^a R_i(\alpha) \end{aligned}$$

FIGURE 3.33 – Les quatre paramètres DH

Consultez une animation vidéo Les paramètres DH sont plus facile à comprendre en visualisant une animation 3D, il est fortement conseillé de consulter un vidéo explicatif pour solidifier vos connaissances.

Pose relative des repères selon les 4 paramètres

La matrice de rotation qui relie la pose des deux repère associés à deux joints successifs peut être paramétrée par les quatre paramètres d , θ , r et α . Le vecteur position qui relie les origines est :

$$\vec{r}_{B_o/A_o} = d \hat{a}_3 + r \hat{b}_1 \quad (3.188)$$

L'orientation relative peut être exprimée comme la combinaison de deux rotation selon les axes \hat{a}_3 et \hat{b}_1 :

$${}^a R^b = {}^a R_3^{a'}(\theta) {}^{a'} R_1^b(\alpha) \quad (3.189)$$

$${}^a R_3^{a'}(\theta) = \begin{bmatrix} c\theta & -s\theta & 0 \\ s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.190)$$

$${}^{a'} R_1^b(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\alpha & -s\alpha \\ 0 & s\alpha & c\alpha \end{bmatrix} \quad (3.191)$$

Les composantes du vecteur position dans la base a sont donc :

$$\underline{r}_{B_o/A_o}^a = \begin{bmatrix} 0 \\ 0 \\ d \end{bmatrix} + {}^a R^{a'} \begin{bmatrix} r \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} r c\theta \\ r s\theta \\ d \end{bmatrix} \quad (3.192)$$

La matrice de transformation peut alors être assemblée :

$${}^a T^b(d, \theta, r, \alpha) = \begin{bmatrix} c\theta & -s\theta c\alpha & s\theta s\alpha & r c\theta \\ s\theta & c\theta c\alpha & -c\theta s\alpha & r s\theta \\ 0 & s\alpha & c\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.193)$$

Donc lorsque la convention est utilisée, la matrice de transformation entre deux repères successifs a toujours la forme ci-dessus, et il suffit d'identifier les quatre paramètres DH de chaque joints.



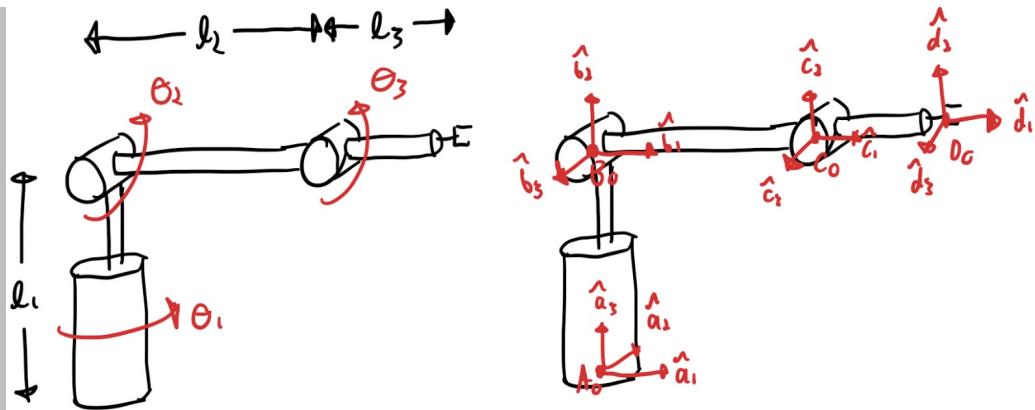
Capsule vidéo

Exemple d'utilisation des paramètres DH

<https://youtu.be/cA0jxfk0gaI>

Exemple 9. Cinématique d'un robot anthropomorphique avec les paramètres DH

La Figure 3.34 illustre l'utilisation de la convention DH pour déterminer la matrice de transformation qui représente la pose de l'effecteur.



	d	θ	r	α
$A T^B$	l_1	θ_1	0	$\pi/2$
$B T^C$	0	θ_2	l_2	0
$C T^D$	0	θ_3	l_3	0

$$A T^D = A T^B B T^C C T^D =$$

$$\begin{bmatrix} c\theta_1 & 0 & s\theta_1 & 0 \\ s\theta_1 & 0 & -c\theta_1 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & l_2 c\theta_2 \\ s\theta_2 & c\theta_2 & 0 & l_2 s\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & l_3 c\theta_3 \\ s\theta_3 & c\theta_3 & 0 & l_3 s\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

FIGURE 3.34 – Exemple de modélisation de la cinématique d'un robot manipulateur anthropomorphe avec la convention des paramètres DH

Joint 1 Premièrement l'axe de rotation du 1er joint doit être repéré, et le repère fixe A doit être placé de sorte à aligner son vecteur unitaire \hat{a}_3 avec cet axe. Les autres variables sont arbitraires, ici on a choisi de positionner l'origine au sol et l'aligner le vecteur unitaire \hat{a}_1 avec les autres liens du robot pour la configuration nominale. Ensuite l'axe de rotation du 2e joint doit être repéré et le repère B doit être fixé de sorte à aligner son vecteur \hat{b}_3 . L'orientation de \hat{b}_1 doit être alignée avec la normale commune entre \hat{a}_3 et \hat{b}_3 , ici comme les axes se croisent la normale est réduite à un seul point et donc l'orientation de \hat{b}_1 est arbitraire, il a donc été choisi de l'aligner avec les liens rigides en position neutre. Avec les repères A et B positionnés selon la convention, il est maintenant possible de déterminer les quatre paramètres de la première matrice de transformation. La distance entre l'origine A_o et la normale

commune (réduite au point B_o) est ici la longueur du premier lien $d = l_1$ et la longueur de la normale commune est nulle $r = 0$. L'angle θ entre les vecteurs unitaires \hat{a}_1 et \hat{b}_1 est nul pour la position neutre, mais lorsque le joint 1 va tourner l'angle correspond en fait à la position active θ_1 . Finalement l'angle entre \hat{a}_3 et \hat{b}_3 autour de l'axe \hat{b}_1 est égale à $\pi/2$ radians. **Joint 2** Pour la prochaine transformation qui caractérise le joint 2, il faut d'abord positionner le repère C selon la convention. Le vecteur unitaire \hat{c}_3 est aligné avec l'axe de rotation du 2e joint, et le vecteur unitaire \hat{c}_1 est aligné avec la normale commune qui est directement alignée avec \hat{b}_1 . Donc ici l'orientation de la base c est complètement fixée par la convention. Toutefois comme les axes de rotation du joint 2 et 3 sont parallèle, c'est l'origine C_o qui pourrait être placée n'importe où sur l'axe de rotation du joint 3 (il y a une infinité de normales communes entre deux axes parallèles). Par simplicité ici on fixe arbitrairement l'origine C_o directement dans l'axe formé par \hat{b}_1 . Avec le repère C positionné selon la convention il est maintenant possible de déterminer les quatre paramètres DH. Le paramètre d est ici nul car le point B_o est coïncident avec la normale commune choisie et le paramètre $r = l_2$ car la normale commune coïncide directement avec le 2e lien rigide. Ensuite, l'angle θ entre les vecteurs unitaires \hat{b}_1 et \hat{c}_1 est nul pour la position neutre, mais lorsque le joint 1 va tourner l'angle correspond en fait à la position active du 2e joint θ_2 . Finalement l'angle entre \hat{b}_3 et \hat{c}_3 est nul donc $\alpha = 0$. **Joint 3** Pour la dernière transformation, il n'y a pas d'axe de rotation de joint puisque que c'est l'effecteur qui reste à localiser, on choisi ici arbitrairement de garder la même orientation que la base précédente. Avec ce choix, la pose du repère D par rapport au repère C a exactement la même structure que celle du repère C par rapport au repère B . La seule différence est les variables impliquées qui sont θ_3 et l_3 . On trouve donc $d = 0$, $\theta = \theta_3$, $r = l_3$ et $\alpha = 0$ comme paramètres DH pour le joint 3. Ensuite, il suffit d'utiliser la formule donnée à l'équation (3.229) pour calculer les matrices de transformation entre chaque repère, et de les multiplier ensemble pour calculer la matrice de transformation totale. Avec la méthode des paramètres DH, le plus long de la démarche est donc de bien positionner et déterminer les paramètres.

3.8.6 Chaîne cinématique fermée

Les chaînes cinématiques fermées, sont des assemblages mécaniques où les liens et articulations ne sont pas assemblés dans une configuration série pure, mais plutôt en parallèle comme illustré à la figure 3.35.

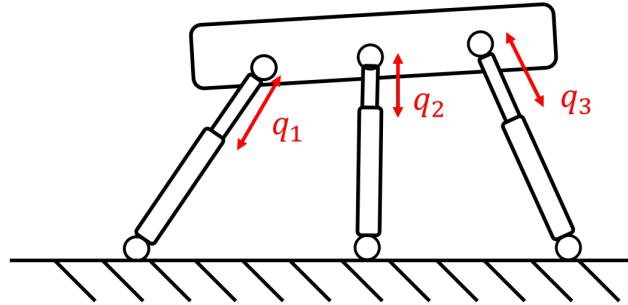


FIGURE 3.35 – Chaîne cinématique fermée (robot parallèle)

Le calcul de la cinématique directe de ce type de mécanisme peut être plus compliqué car chaque chemin parallèle qui relie l'effecteur à la base d'un robot rajoute une contrainte au système. Les mouvements des joints ne sont alors pas nécessairement indépendants. Pour une configuration q des joints, il peut ne pas y avoir aucune pose de l'effecteur qui satisfait toutes les contraintes ou potentiellement plusieurs poses de l'effecteur qui satisfont les contraintes.

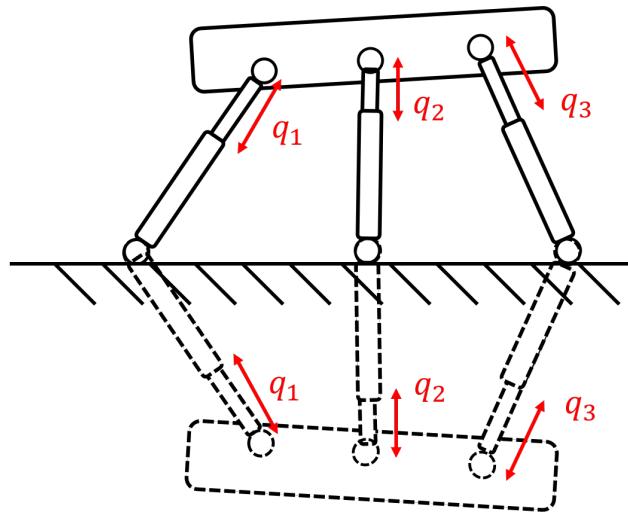


FIGURE 3.36 – Chaîne cinématique fermée avec deux solutions (théoriques) pour la pose de l'effecteur selon une configuration q_1 , q_2 et q_3 des joints.

Détails à venir !

3.9 Cinématique inverse d'un manipulateur

La cinématique inverse consiste à calculer des coordonnées de l'espace des joints \underline{q} qui mènent à une position désirée de l'effecteur. Par exemple, pour la translation, la cinématique inverse est le calcul de la fonction inverse de l'équation (3.138) :

$$\underline{q} = f_{Trans}^{-1}(\underline{r}) \quad (3.194)$$

3.9.1 Chaîne cinématique ouverte

Pour les chaînes cinématiques ouvertes, comme la plupart des robots manipulateurs, le calcul de la cinématique directe est sans ambiguïté : une configuration \underline{q} mène toujours à une seule position de l'effecteur possible. Toutefois, il existe plusieurs configurations \underline{q} qui peuvent mener à la même position de l'effecteur, voir figure 3.37b par exemple. De plus, pour certaines positions de l'effecteur (ceux qui ne sont pas dans l'espace de travail), l'équation (3.194) n'aura pas de solutions. La cinématique inverse est donc généralement plus compliquée que la cinématique directe, elle consiste à résoudre une fonction hautement non-linéaire qui possède potentiellement aucune ou plusieurs solutions. De plus, même lorsque qu'une ou plusieurs solutions existent, il n'est pas toujours possible de résoudre analytiquement l'équation (3.194). Il est alors nécessaire d'utiliser des méthodes numériques pour trouver des solutions, comme la méthode de Newton-Raphson par exemple. Pour la plupart des architectures de robots industriels, la cinématique inverse peut être résolue analytiquement. Pour les bras manipulateurs à 6 DLL, le critère pour que la cinématique inverse puisse être résolue analytiquement est que l'axe de trois joints rotatifs séquentiels se croise en un point. La plupart des robots industriels ont leurs 3 derniers DDL configurés en un poignet (*wrist* en anglais) qui consiste en trois joints rotatifs avec leurs trois axes qui se croisent en un seul point, et ils rencontrent donc ce critère.

En pratique, les contrôleurs de robot trouvent généralement indirectement les solutions à l'équation (3.194). Une méthode standard est de contrôler les déplacement locaux de l'effecteur en direction de la position \underline{r} désirée, voir section 8.2, une méthode qui revient à faire une descente du gradient itérative comme avec la méthode de Newton-Raphson mais en temps réel avec le vrai système.

Exemple 10. Cinématique inverse d'un robot à deux joints rotatifs

Cet exemple illustre une résolution analytique de la cinématique inverse d'un robot simple à deux joints rotatifs dans le plan, illustré à la figure 3.37.

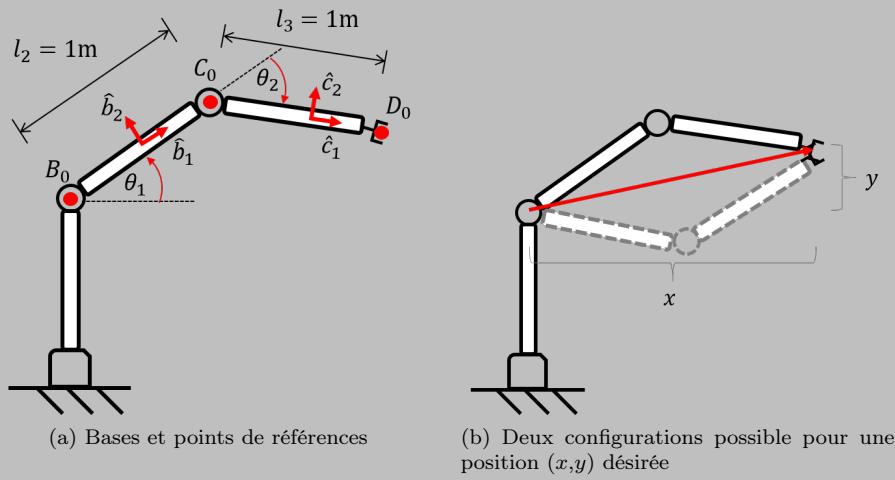


FIGURE 3.37 – Cinématique inverse d'un robot à deux joints rotatif

L'objectif est ici de calculer la fonction de cinématique inverse :

$$\underline{q} = f_{Trans}^{-1} \left(\underline{r}_{D_o/B_o}^a \right) \quad (3.195)$$

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = f_{Trans}^{-1} \left(\begin{bmatrix} x \\ y \end{bmatrix} \right) \quad (3.196)$$

Premièrement la relation vectorielle est établie :

$$\vec{r}_{D_o/B_o} = \vec{r}_{D_o/C_o} + \vec{r}_{C_o/B_o} \quad (3.197)$$

$$\vec{r}_{D_o/B_o} = l_3 \hat{c}_1 + l_2 \hat{b}_1 \quad (3.198)$$

Ensuite il est possible de relier la position x et y de l'effecteur avec l'angle θ_2 en calculant la longueur du vecteur \vec{r}_{D_o/B_o} :

$$\|\vec{r}_{D_o/B_o}\|^2 = \|\vec{r}_{D_o/C_o} + \vec{r}_{C_o/B_o}\|^2 \quad (3.199)$$

$$x^2 + y^2 = (l_3 \hat{c}_1 + l_2 \hat{b}_1) \cdot (l_3 \hat{c}_1 + l_2 \hat{b}_1) \quad (3.200)$$

$$x^2 + y^2 = l_2^2 + l_3^2 + 2l_2 l_3 (\hat{c}_1 \cdot \hat{b}_1) \quad (3.201)$$

$$x^2 + y^2 = l_2^2 + l_3^2 + 2l_2 l_3 \cos \theta_2 \quad (3.202)$$

À noter que la dérivation ci-dessus est équivalente à utiliser la loi des cosinus. Il est donc possible d'obtenir l'angle θ_2 comme une fonctions des coordonnées désirées de l'effecteur et des paramètres géométriques du robot :

$$\cos \theta_2 = \frac{(x^2 + y^2) - (l_2^2 + l_3^2)}{2l_2 l_3} \quad (3.203)$$

L'équation (3.203) peut avoir 0, 1 ou 2 solutions. Si la position désirée (x,y) est hors de l'espace de travail, il n'y a pas de solution :

$$\frac{(x^2 + y^2) - (l_2^2 + l_3^2)}{2l_2 l_3} > 1 \Rightarrow \text{Aucune solution} \quad (3.204)$$

Si la position désirée (x,y) est sur la limite de l'espace de travail, il y a une seule solution où les deux liens rigides sont alignés :

$$\frac{(x^2 + y^2) - (l_2^2 + l_3^2)}{2l_2 l_3} = 1 \Rightarrow \text{Une solution : } \theta_2 = 0 \quad (3.205)$$

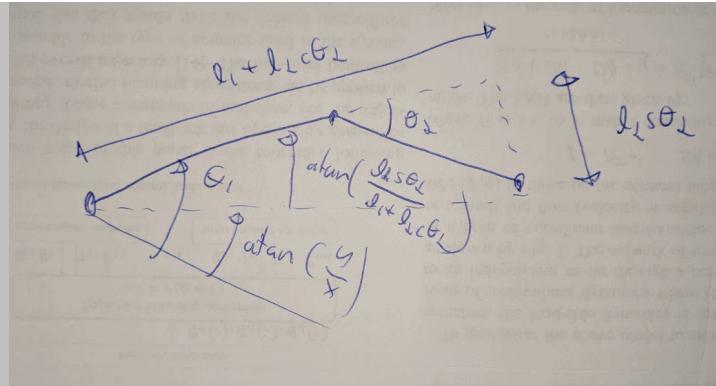
En fait, techniquement il faudrait plutôt dire qu'il y a une infinité de solution périodique pour tout les facteurs entiers multiplicateurs de 2π , i.e. $\theta_2 = 2i\pi$ avec $i = 0, 1, -1, 2, -2, \dots$, si on considère que le 2e joint du robot peut illimité pour sa plage angulaires.

Si la position désirée (x, y) est dans l'espace de travail, il y a deux solutions comme illustré à la figure 3.37b :

$$\frac{(x^2 + y^2) - (l_2^2 + l_3^2)}{2l_2 l_3} < 1 \Rightarrow \text{Deux solutions : } \theta_2 = \pm \arccos \left(\frac{(x^2 + y^2) - (l_2^2 + l_3^2)}{2l_2 l_3} \right) \quad (3.206)$$

Il est aussi possible d'isoler l'angle θ_1 sous forme explicite, comme illustré à la figure 3.38 :

$$\theta_1 = \arctan \left(\frac{y}{x} \right) + \arctan \left(\frac{l_2 s \theta_2}{l_1 + l_2 c \theta_2} \right) \quad (3.207)$$

FIGURE 3.38 – Calcul de θ_1

3.9.2 Chaîne cinématique fermée

Contrairement aux chaînes cinématiques ouvertes, la cinématique inverse d'une chaîne cinématique fermée est généralement plus facile à calculer que la cinématique ouverte. Par exemple, comme illustré à la figure 3.39 pour un robot parallèle simple, si la pose de l'effecteur est précisée, alors les variables de joints q_i peuvent être facilement calculés et il y a une seule solution. L'état des joints prismatique peut être calculée sachant la longueurs des vecteurs \vec{r}_{B_i/A_i} :

$$q_i + l_i = \|\vec{r}_{B_i/A_i}\| \quad (3.208)$$

et la position des points B_i et A_i peut être déterminée avec la géométrie du robot et la pose de l'effecteur. Les variables q_i pourraient donc être calculées ainsi :

$$q_i = \|\vec{r}_{B_i/A_i}\| - l_i \quad (3.209)$$

$$q_i = \|\vec{r}_{B_i/T_o} + \vec{r}_{T_o/W_o} - \vec{r}_{A_i/W_o}\| - l_i \quad (3.210)$$

$$q_i = \sqrt{\left[\underline{r}_{B_i/T_o}^w + \underline{r}_{T_o/W_o}^w - \underline{r}_{A_i/W_o}^w \right]^T \left[\underline{r}_{B_i/T_o}^w + \underline{r}_{T_o/W_o}^w - \underline{r}_{A_i/W_o}^w \right]} - l_i \quad (3.211)$$

$$q_i = \sqrt{\left[{}^w R^t \underline{r}_{B_i/T_o}^t + \underline{r}_{T_o/W_o}^w - \underline{r}_{A_i/W_o}^w \right]^T \left[{}^w R^t \underline{r}_{B_i/T_o}^t + \underline{r}_{T_o/W_o}^w - \underline{r}_{A_i/W_o}^w \right]} - l_i \quad (3.212)$$

où ${}^w R^t$ et $\underline{r}_{T_o/W_o}^w$ sont des fonctions de la pose de l'effecteur et $\underline{r}_{B_i/T_o}^t$ et $\underline{r}_{A_i/W_o}^w$ sont des vecteur-colonnes constantes et seulement fonction de la géométrie du robot.

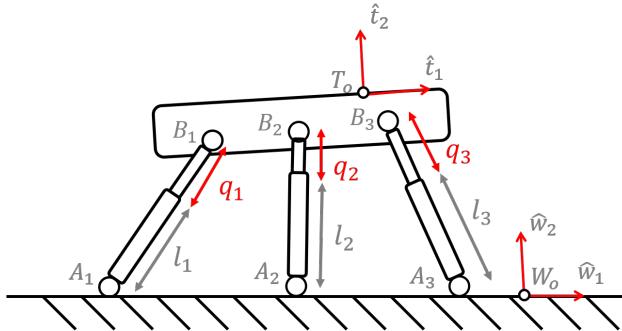


FIGURE 3.39 – Cinématique inverse d'un chaîne cinématique fermée (robot parallèle)

3.10 Résumé du chapitre

Cinématique des robots manipulateurs :

$$\underbrace{\underline{r}}_{\text{Coordonnées dans l'espace de la tâche}} = f(\underbrace{\underline{q}}_{\text{Coordonnées dans l'espace des joints}}) \quad (3.213)$$

Vecteurs de position :

$$\vec{r}_{B/A} = \text{Vecteur position du point } B \text{ par rapport au point } A \quad (3.214)$$

$$\vec{r}_{Z/A} = \vec{r}_{Z/Y} + \vec{r}_{Y/X} + \dots + \vec{r}_{C/B} + \vec{r}_{B/A} \quad (3.215)$$

$$\vec{r}_{B/A} = -\vec{r}_{A/B} \quad (3.216)$$

Bases vectorielles :

$$\text{Base } a = \{\hat{a}_1, \hat{a}_2, \hat{a}_3\} \quad (3.217)$$

Vecteur géométrique vs. vecteur-colonne de composantes :

$$\vec{r} = r_1^a \hat{a}_1 + r_2^a \hat{a}_2 + r_3^a \hat{a}_3 \quad \underline{r}^a = \begin{bmatrix} r_1^a \\ r_2^a \\ r_3^a \end{bmatrix} \quad (3.218)$$

$$\vec{r} = \sum_i r_i^a \hat{a}_i \quad r_i^a = \vec{r} \cdot \hat{a}_i \quad (3.219)$$

Matrice de rotation :

$${}^aR^b = \begin{bmatrix} \hat{a}_1 \cdot \hat{b}_1 & \hat{a}_2 \cdot \hat{b}_1 & \hat{a}_3 \cdot \hat{b}_1 \\ \hat{a}_1 \cdot \hat{b}_2 & \hat{a}_2 \cdot \hat{b}_2 & \hat{a}_3 \cdot \hat{b}_2 \\ \hat{a}_1 \cdot \hat{b}_3 & \hat{a}_2 \cdot \hat{b}_3 & \hat{a}_3 \cdot \hat{b}_3 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} b_1^a \end{bmatrix} & \begin{bmatrix} b_2^a \end{bmatrix} & \begin{bmatrix} b_3^a \end{bmatrix} \end{bmatrix} \quad (3.220)$$

Propriétés des matrices de rotation :

$$\underline{r}^a = {}^aR^b \underline{r}^b \quad (3.221)$$

$${}^aR^c = {}^aR^b {}^bR^c \quad (3.222)$$

$${}^aR_{ij}^b = \hat{a}_i \cdot \hat{b}_j \quad (3.223)$$

$$R(\theta)^{-1} = R(-\theta) \quad (3.224)$$

$$R^{-1} = R^T \quad (3.225)$$

$$({}^bR^a)^{-1} = {}^aR^b \quad (3.226)$$

Repère :

$$\text{Repère } A = \text{Base } a + \text{Origine } A_o = \{A_o, \hat{a}_1, \hat{a}_2, \hat{a}_3\} \quad (3.227)$$

Matrice de transformation :

$${}^AT^B = \begin{bmatrix} {}^aR^b & \underline{r}_{B_o/A_o}^a \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.228)$$

Transformation avec les paramètres DH :

$$T(d, \theta, r, \alpha) = \begin{bmatrix} c\theta & -s\theta c\alpha & s\theta s\alpha & rc\theta \\ s\theta & c\theta c\alpha & -c\theta s\alpha & rs\theta \\ 0 & s\alpha & c\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.229)$$

Chapitre 4

Cinématique des robots manipulateurs II : Le mouvement

Dans le chapitre 3, les méthodes pour modéliser la position, l'orientation ont été présentée. Ce chapitre présente les méthodes associées pour représenter leurs variation dans le temps, c'est-à-dire la vitesse. Les premières sections décrivent les principes généraux et ensuite des méthodes spécifiques pour les robots manipulateurs sont présentées.

4.1 Référentiel

Un référentiel est un solide ou un ensemble de points par rapport auquel un observateur qui mesure un mouvement est fixe. Une position ou un mouvement ne peut être défini par rapport au vide et il n'y a pas de référentiel absolu, le choix du référentiel est donc un choix arbitraire de point de vue. Il est à noter que **contrairement au choix d'un repère, le choix du référentiel influence "la physique" du problème**, par exemple l'énergie cinétique d'un objet n'est pas la même selon le référentiel alors que cette quantité est indépendante du système de coordonnées. Un référentiel dit inertiel ou Galiléen (vitesse et orientation constante) est préférable si on souhaite éventuellement appliquer les équations de Newtons, car des facteurs correctifs (ex. : la force centrifuge) doivent être ajoutés aux équations dynamiques si un référentiel non-inertiel est utilisé. La notion de référentiel est critique pour le calcul de vitesses et accélérations, mais on peut en faire abstraction pour les problèmes de cinématique et statique qui n'impliquent pas la notion d'évolution dans le temps.

4.1.1 Référentiel vs. repère

Pour décrire le mouvement d'un objet, les concepts de référentiel, origine et base vectorielle sont importants à distinguer. En quelques mots, un **référentiel** est un point de vue utilisé pour décrire un phénomène, une **origine** c'est un point par rapport auquel les mesures de position sont faites et une **base vectorielle** représente l'orientation du système d'axe utilisé. Ces trois éléments peuvent être choisis indépendamment. Ensuite, un **repère** c'est la combinaison d'une base et d'une origine. Par exemple, pour décrire la position d'un bateau, typiquement le référentiel utilisé serait la Terre, l'origine des mesures de position pourrait être le port le plus près, et la base pour exprimer cette position serait le système d'axe nord-sud/est-ouest. Il est important de noter que tous ces choix sont indépendants et les confondre est une source d'erreur courante en cinématique. Le reste de cette section discute de ce qui les distingue.



Capsule vidéo
Référentiel vs. repère vs. base vectorielle
https://youtu.be/Uv_Mbbfkis4

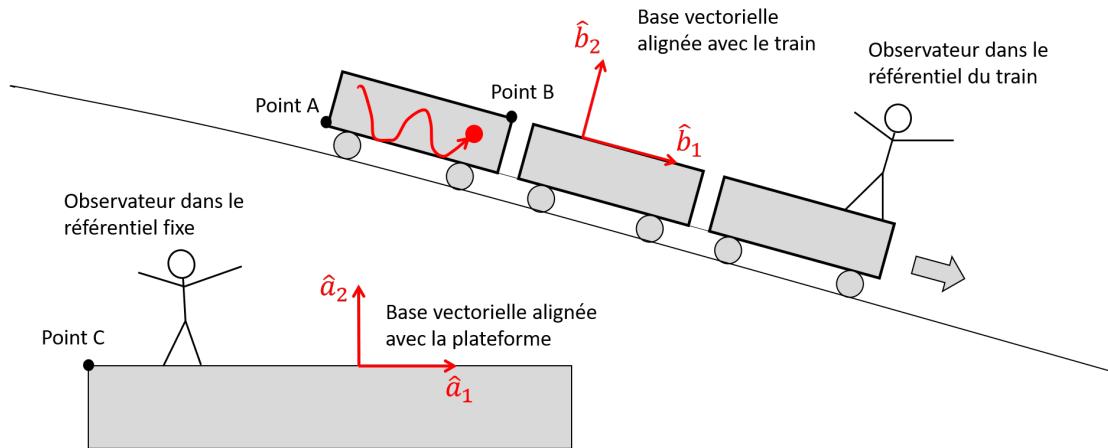


FIGURE 4.1 – Exemple de référentiels, points d'origine et bases vectorielles

La figure 4.1 illustre ces concepts par un exemple où un train descend une pente à vitesse constante et un ballon rebondit sur le sol du troisième wagon. Deux bases vectorielles sont définies, une alignée avec l'horizontale et une alignée avec le train. Plusieurs points d'origine pour les mesures de position sont aussi disponibles. Finalement, deux référentiels sont définis, un référentiel attaché à la Terre (fixe) et un référentiel qui est attaché au train. Les coordonnées d'un vecteur-colonne qui décris la position du ballon :

$$\underline{r}_{Ball} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \quad (4.1)$$

sont illustrés pour différents choix d'origine et de base vectorielle, lorsque que le référentiel fixe est utilisé (figure 4.2) et lorsque que le référentiel du train est utilisé (figure 4.3). Il est à noter que les points d'origine sont ici fixés à la Terre lorsque le référentiel fixe est utilisé, et fixés au train lorsque que le référentiel du train est utilisé. Comme illustré pour cet exemple, **un changement d'origine est analogue à une translation de la trajectoire, un changement de base vectorielle est analogue à une rotation de la trajectoire, et un changement de référentiel est ici analogue à une dilatation de la trajectoire dans la direction \hat{b}_1** , qui correspond à la direction de la vitesse du train.

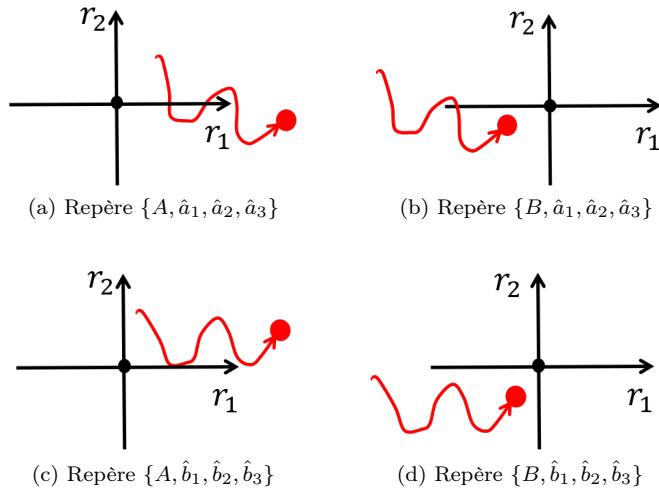


FIGURE 4.2 – Trajectoire du ballon avec un repère attaché au référentiel fixe

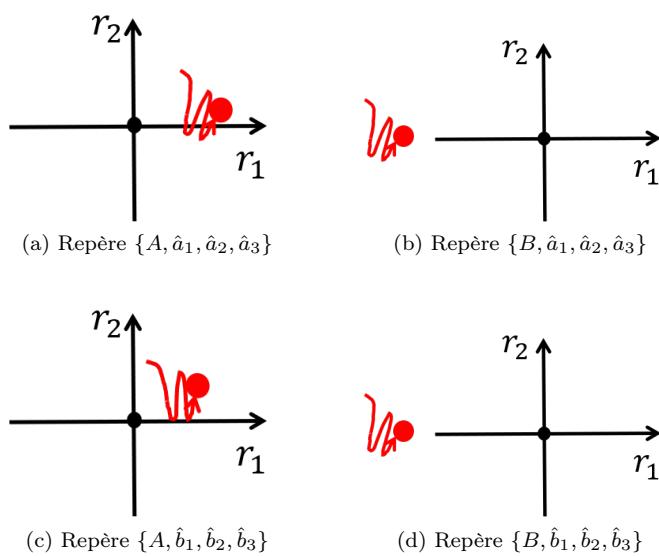


FIGURE 4.3 – Trajectoire du ballon avec un repère attaché au référentiel du train

4.2 Vitesse et dérivée d'un vecteur position

Un vecteur de vitesse est défini comme la dérivée temporelle d'une vecteur de position :

$$\vec{v}_{B/A} = \frac{d}{dt} \vec{r}_{B/A} \quad (4.2)$$

Toutefois, pour avoir la notion de vitesse précise qui correspond à un quantité de mouvement (momentum) qui peut être utilisée dans les divers lois de la physique (ex. conservation du mouvement, conservation de l'énergie, etc.), la dérivée doit être faite par rapport à un observateur Galiléen, aussi appelé un référentiel inertiel. Donc pour avoir la vitesse \vec{v}_B d'un point B il faut 1) que la dérivée soit faite par rapport à une base vectorielle avec un orientation fixe dans le référentiel Galiléen, et 2) que le vecteur position dérivé soit par rapport à un point d'origine fixe dans le référentiel Galiléen.

$$\underbrace{\vec{v}_B}_{\substack{\text{Vitesse du point } B \\ \text{observateur Galiléen}}} = \underbrace{\frac{d}{dt}}_{\substack{\text{Point fixe}}} \underbrace{\vec{r}_{B/A}}_{\substack{\text{Point fixe}}} \quad (4.3)$$

Avec cette définition, le vecteur vitesse est défini seulement par un seul point contrairement au vecteur position qui prend un sens seulement avec une origine. La méthode la plus simple pour déterminer le vecteur vitesse d'un point basé sur une expression connue pour son vecteur position est d'exprimer le vecteur position dans un repère fixe, et ensuite de dériver chaque composante. Donc si il est possible d'exprimer le vecteur position d'un point B par rapport à un point fixe A et en utilisant une base vectorielle fixe a :

$$\vec{r}_{B/A} = x\hat{a}_1 + y\hat{a}_2 + z\hat{a}_3 \quad (4.4)$$

il suffit de dériver l'expression de chaque composante dans cette base pour déterminer le vecteur vitesse du point B :

$$\vec{v}_B = \frac{d}{dt} \vec{r}_{B/A} = \dot{x}\hat{a}_1 + \dot{y}\hat{a}_2 + \dot{z}\hat{a}_3 \quad (4.5)$$

4.2.1 Dérivée d'un vecteur position exprimé avec une base mobile

Notation pour les repères fixes et mobiles Pour les sections suivantes, les expressions sont développées en considérant que les repères notés A (base a et origine A) sont fixes dans le référentiel inertiel et que les repères notés B (base b et origine B) sont arbitraires donc potentiellement mobiles.

Dans certaines situations, le vecteur position est plus naturellement exprimé dans un repère mobile, cette section développe des expressions qui peuvent être utilisées pour déterminer le vecteur vitesse sans avoir à tout transférer dans un repère inertiel. Si on connaît les composantes d'un vecteur position dans une base mobile b :

$$\vec{r}_{B/A} = x\hat{b}_1 + y\hat{b}_2 + z\hat{b}_3 \quad (4.6)$$

Lorsqu'on dérive par rapport au temps, il faut tenir compte que les vecteurs unitaires de la base b ont une direction qui change dans le temps, et donc une dérivée temporelle non-nulle :

$$\vec{v}_B = \frac{d}{dt} \vec{r}_{B/A} = \underbrace{\dot{x}\hat{b}_1 + \dot{y}\hat{b}_2 + \dot{z}\hat{b}_3}_{b \frac{d}{dt} \vec{r}_{B/A}} + \underbrace{\dot{x}\hat{b}_1 + y\dot{\hat{b}}_2 + z\dot{\hat{b}}_3}_{\vec{w}_{b/a} \times \vec{r}_{B/A}} \quad (4.7)$$

Il y a donc deux sources de contribution au vecteur vitesse, le taux de variation des composantes et le taux de variation de la direction des vecteurs unitaires. La dérivée temporelle des vecteurs unitaires est directement reliée à la vitesse angulaire de la base b (voir section 4.3), et cette contribution peut être calculée avec le produit vectoriel d'un vecteur de vitesse angulaire de la base b par rapport à une base fixe a fois le vecteur de position :

$$\vec{v}_B = \underbrace{b \frac{d}{dt} \vec{r}_{B/A}}_{\substack{\text{Observateur mobile}}} + \underbrace{\vec{w}_{b/a} \times \vec{r}_{B/A}}_{\substack{\text{Effet de la vitesse angulaire}}} \quad (4.8)$$

où b est une base mobile, a est une base fixe dans le référentiel inertiel, le terme ${}^b \frac{d}{dt}$ représente le taux de variation vu par un observateur dans le référentiel non-inertiel qui tourne avec la base b et $\vec{w}_{b/a}$ est la vitesse angulaire de la base b p/r à une base a inertuelle.

Équation matricielle équivalente avec les composantes :

On peut trouver une relation équivalente qui implique les vecteur-colonnes de composantes et les matrices de rotation. Le vecteur vitesse est la dérivée temporelle des composantes dans la base fixe a , on peut donc substituer par le vecteur-colonne de composantes dans la base b fois la matrice de rotation ${}^a R^b$ et prendre la dérivée :

$$\underline{v}_B^a = \frac{d}{dt} \underline{r}_{B/A}^a = \frac{d}{dt} \left({}^a R^b \underline{r}_{B/A}^b \right) \quad (4.9)$$

Ici on applique la règle de dérivé d'un produit et comme avec la relation vectorielle de l'équation (4.7), on trouve deux termes :

$$\underline{v}_{B/A}^a = {}^a R^b \dot{\underline{r}}_{B/A}^b + {}^a \dot{R}^b \underline{r}_{B/A}^b \quad (4.10)$$

Toutefois ici le taux de variation de la direction de la base mobile b est représenté par la dérivée temporelle de la matrice de rotation ${}^a R^b$ (voir section 4.3). On peut utiliser une expression qui utilise les composantes de la vitesse angulaire de la base b pour déterminer la dérivée de la matrice de rotation

$$\underline{v}_B^a = {}^a R^b \dot{\underline{r}}_{B/A}^b + (\underline{w}_{b/a}^a)^\times {}^a R^b \underline{r}_{B/A}^b \quad (4.11)$$

$$\underline{v}_B^a = {}^a R^b \dot{\underline{r}}_{B/A}^b + {}^a R^b (\underline{w}_{b/a}^b)^\times \underline{r}_{B/A}^b \quad (4.12)$$

pour obtenir une forme équivalente à l'expression vectorielle précédemment obtenue (eq. (4.8), mais impliquant les vecteurs-colonnes de composantes et la matrice de rotation :

$$\vec{v}_B = {}^b \frac{d}{dt} \vec{r}_{B/A} + \vec{w}_{b/a} \times \vec{r}_{B/A} \Leftrightarrow \underline{v}_B^a = {}^a R^b \left[\dot{\underline{r}}_{B/A}^b + (\underline{w}_{b/a}^b)^\times \underline{r}_{B/A}^b \right] \quad (4.13)$$

4.3 Vitesse angulaire et dérivée d'une matrice de rotation

La représentation de l'orientation a plusieurs difficultés mathématiques : pour représenter seulement 3 DDL les matrices de rotation ont 9 paramètres, les rotations ne sont pas commutatives, etc. Toutefois, les petites rotations infinitésimales peuvent être représentées par un vecteur de trois paramètres et sont commutatives. Un vecteur rotation infinitésimal a comme direction l'axe de rotation et comme amplitude un angle autour de cet axe. Lorsque qu'on divise un vecteur rotation infinitésimal par dt on obtient le concept de vecteur de vitesse angulaire :

$$\vec{w} = \underbrace{w \left[\frac{\text{rad}}{\text{sec}} \right]}_{\text{Taux de variation d'un angle}} \underbrace{\hat{a}}_{\text{axe de rotation instantané}} \quad (4.14)$$

Le vecteur de vitesse angulaire \vec{w} a comme direction l'axe de rotation instantané \hat{a} et comme amplitude un taux de variation w d'un angle autour de cet axe. D'un point de vue matricielle, le vecteur de vitesse angulaire est associé à la dérivée temporelle d'une matrice de rotation :

$$\dot{R} = \frac{d}{dt} R = (\underline{w})^x R \quad \text{avec } (\underline{w})^x = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix} \quad (4.15)$$

et du point de vue vectorielle, de façon équivalente à la dérivée temporelle des vecteurs unitaires de la base mobile peut être calculé avec un produit vectoriel du vecteur de vitesse angulaire et les vecteurs unitaires :

$$\dot{\hat{b}}_i = \vec{w} \times \hat{b}_i \quad (4.16)$$

Lorsqu'on utilise la représentation des angles de Euler, le vecteur de vitesse angulaire est définie directement avec la dérivée temporelle des trois angles de Euler et leur axe de rotation respectif :

$$\vec{w}_{d/a} = \dot{\phi}\hat{a}_3 + \dot{\theta}\hat{b}_1 + \dot{\psi}\hat{c}_3 \quad (4.17)$$

voir la définition des angles et axes à la section 3.7.1. Comme la base vectorielle de l'équation vectorielle précédente n'est pas orthogonale, il faut faire attention car :

$$\begin{bmatrix} \dot{w}_1 \\ \dot{w}_2 \\ \dot{w}_3 \end{bmatrix} \neq \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (4.18)$$

Il faut convertir le vecteur $\vec{w}_{d/a}$ dans la base approprié pour déterminer ses composantes.

4.3.1 Propriétés

Premièrement, une vitesse angulaire est le taux de variation d'une orientation relative entre deux bases. Tout comme les matrices de rotation pour être précis il faut donc noter les bases associés pour bien définir un vecteur de vitesse angulaire :

$$\vec{w}_{b/a} \quad (4.19)$$

est le taux de variation de l'orientation de la base b par rapport à la base a . Selon si le vecteur $\vec{w}_{b/a}$ est exprimé dans la base a ou b deux expressions permettent de calculer la dérivée temporelle de la matrice de rotation associée :

$${}^a\dot{R}^b = (\underline{w}_{b/a}^a)^x {}^aR^b = {}^aR^b (\underline{w}_{b/a}^b)^x \quad (4.20)$$

Ensuite, il est à noter lors du calcul des dérivées temporelles des vecteurs unitaires, que la drivée obtenue est selon un observateur dans la base d'origine pour laquelle le vecteur de vitesse angulaire est calculé :

$${}^a \frac{d}{dt} \hat{b}_i = \vec{w}_{b/a} \times \hat{b}_i \quad (4.21)$$

donc pour obtenir une dérivée temporelle de \hat{b}_i dans un référentiel inertiel, la base a du vecteur de vitesse angulaire doit être inertiel.

Addition Une propriété utile pour calculer une vitesse angulaire totale, i.e. par rapport au référentiel inertiel, lorsqu'on a plusieurs vitesse angulaire relative (par exemple pour chacun des joints d'un robot), est qu'il suffit d'additionner les vecteur de vitesse angulaire.

$$\vec{w}_{d/a} = \vec{w}_{d/a} + \vec{w}_{c/b} + \vec{w}_{b/a} \quad (4.22)$$

Commutativité Contrairement aux rotations, la combinaison de vecteurs de vitesses angulaires est commutative, i.e. l'ordre n'a pas d'importance :

$${}^a R^b {}^b R^c \neq {}^b R^c {}^a R^b \quad (4.23)$$

$$\vec{w}_{c/b} + \vec{w}_{b/a} = \vec{w}_{b/a} + \vec{w}_{c/b} \quad (4.24)$$

4.4 Accélération et dérivée seconde d'un vecteur position

$$\vec{a}_B = {}^b \frac{d}{dt} \vec{v}_B + \vec{w}_{b/a} \times \vec{v}_B \quad \Leftrightarrow \quad \underline{a}_B^a = {}^a R^b \left[\underline{\dot{v}}_B^b + (\underline{w}_{b/a}^b)^\times \underline{v}_B^b \right] \quad (4.25)$$

$$\vec{a}_B = {}^b \frac{d^2}{dt^2} \vec{r}_{B/A} + {}^b \frac{d}{dt} \vec{w}_{b/a} \times \vec{r}_{B/A} + \underbrace{2\vec{w}_{b/a} \times {}^b \frac{d^2}{dt^2} \vec{r}_{B/A}}_{\text{Coriolis}} + \underbrace{\vec{w}_{b/a} \times \vec{w}_{b/a} \times \vec{r}_{B/A}}_{\text{Centrifuge}} \quad (4.26)$$

$$\underline{a}_B^a = {}^a R^b \left[\underline{\ddot{r}}_{B/A}^b + (\underline{\dot{w}}_{b/a}^b)^\times \underline{r}_{B/A}^b + \underbrace{2(\underline{w}_{b/a}^b)^\times \underline{\dot{r}}_{B/A}^b}_{\text{Coriolis}} + \underbrace{(\underline{w}_{b/a}^b)^\times (\underline{w}_{b/a}^b)^\times \underline{r}_{B/A}^b}_{\text{Centrifuge}} \right] \quad (4.27)$$

4.5 Cinématique différentielle des robots manipulateurs

La cinématique différentielle c'est l'étude de la relation entre le mouvement des joints et le mouvement associé de l'effecteur. Dans le chapitre 3, les fonctions reliant la position de l'effecteur \underline{r} et la configuration du robot dans l'espace des joints \underline{q} ont été étudiées :

$$\text{Cinématique directe : } \underline{r} = f(\underline{q}) \quad \text{inverse : } \underline{q} = f^{-1}(\underline{r}) \quad (4.28)$$

Plusieurs méthodes en robotique sont plutôt basées sur les fonctions qui relient la vitesse de l'effecteur $\dot{\underline{r}}$ à la vitesse des joints $\dot{\underline{q}}$. Ces fonctions prennent la forme suivante :

$$\text{Cinématique différentielle directe : } \dot{\underline{r}} = J(\underline{q}) \dot{\underline{q}} \quad \text{inverse : } \dot{\underline{q}} = J^{-1}(\underline{q}) \dot{\underline{r}} \quad (4.29)$$

Les fonctions de cinématique différentielle impliquent une matrice Jacobienne qui relit des déplacements infinitésimales des joints aux déplacements infinitésimales de l'effecteur :

$$\text{Matrice Jacobienne : } J(\underline{q}) = \frac{\partial \underline{r}}{\partial \underline{q}} = \begin{bmatrix} \frac{\partial r_1}{\partial q_1} & \dots & \frac{\partial r_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial r_m}{\partial q_1} & \dots & \frac{\partial r_m}{\partial q_n} \end{bmatrix}_{m \times n} \quad (4.30)$$

ou n est le nombre de joints, i.e. le nombre de DDL du robot, et m est le nombre de coordonnées de l'espace sortie, par exemple la pose de l'effecteur. D'un point de vue mathématique, la matrice Jacobienne J correspond au gradient de la fonction multi-variable $\underline{r} = f(\underline{q})$. Plus de détails sur la dérivation dans un contexte multi-variable sont disponibles à la section 17.5.

La composante J_{ij} de la matrice correspond à la variation de la coordonnée r_i de l'effecteur, due à un déplacement unitaire du joint q_j . La relation entre les déplacements infinitésimales est données par :

$$\begin{bmatrix} d\underline{r} \\ d\underline{q} \end{bmatrix}_{m \times 1} = \begin{bmatrix} J(\underline{q}) \\ \vdots \end{bmatrix}_{m \times n} \begin{bmatrix} d\underline{q} \\ d\underline{q} \end{bmatrix}_{n \times 1} \Leftrightarrow dr_j = \sum_i^n J_{ji} dq_i \quad j \in \{1, \dots, m\} \quad (4.31)$$

La relation qui relie les vitesses est relié à la relation entre les déplacements infinitésimales par le principe des dérivées en chaînes :

$$\text{Dérivées en chaînes : } \underbrace{\frac{d\underline{r}}{dt}}_{\dot{\underline{r}}} = \underbrace{\left(\frac{\partial \underline{r}}{\partial \underline{q}} \right)}_{J(\underline{q})} \underbrace{\frac{d\underline{q}}{dt}}_{\dot{\underline{q}}} \quad (4.32)$$

autrement dit, il suffit de diviser l'équation (4.31) par dt pour obtenir la relation entre les vitesses.

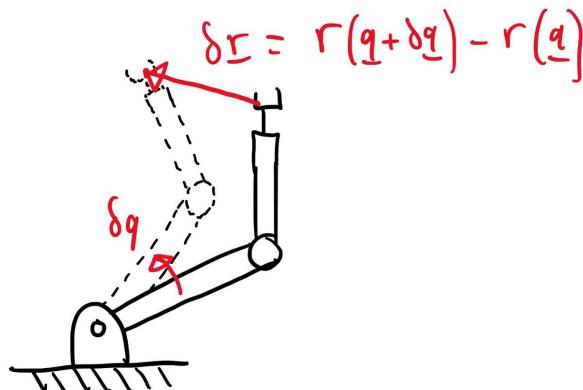


FIGURE 4.4 – Le Jacobien décrit la relation entre des déplacements infinitésimal



Capsule vidéo
Cinématique différentielle et matrice jacobienne
<https://youtu.be/jJqYWNSsJvE>

4.5.1 La matrice Jacobienne

Unités Typiquement pour les analyse de cinématique différentielle, les variables de sortie sont les DDL de translation de l'effecteur, la pose complète de l'effecteur ou bien des variables qui décrivent les DDL de l'espace de la tâche; et les variables d'entrée sont les vitesses angulaire ou linéaire des joints du robot. Les unités des composantes J_{ij} de la matrice Jacobienne associée sont donc typiquement soit des unités de distance qui correspondent à des *bras de levier*, ou bien des ratios adimensionnels.

Non-linéarité Il est important de noter que généralement la matrice Jacobienne est une fonction de la configuration \underline{q} du robot. Comme la fonction de cinématique directe est généralement hautement non-linéaire, l'effet d'un déplacement infinitésimal d'un joint sur l'effecteur dépend considérablement de la configuration initiale. La matrice Jacobienne est donc seulement valide pour décrire des petites variation de déplacement autour de la configuration pour laquelle elle a été évaluée, ou bien pour la relation entre les vitesses momentanément à cette configuration.

Colonnes de la matrice Jacobienne Comme illustré à la Figure 4.5, dans le contexte de cinématique différentiel, la colonne j de la matrice Jacobienne correspond à un vecteur déplacement infinitésimal de l'effecteur due au mouvement du joint j :

$$J = \left[\frac{\partial \underline{r}}{\partial q_1} \cdots \frac{\partial \underline{r}}{\partial q_i} \cdots \frac{\partial \underline{r}}{\partial q_n} \right] \quad (4.33)$$

Chaque colonne a donc une interprétation vectorielle visuelle comme illustré à la Figure suivante.

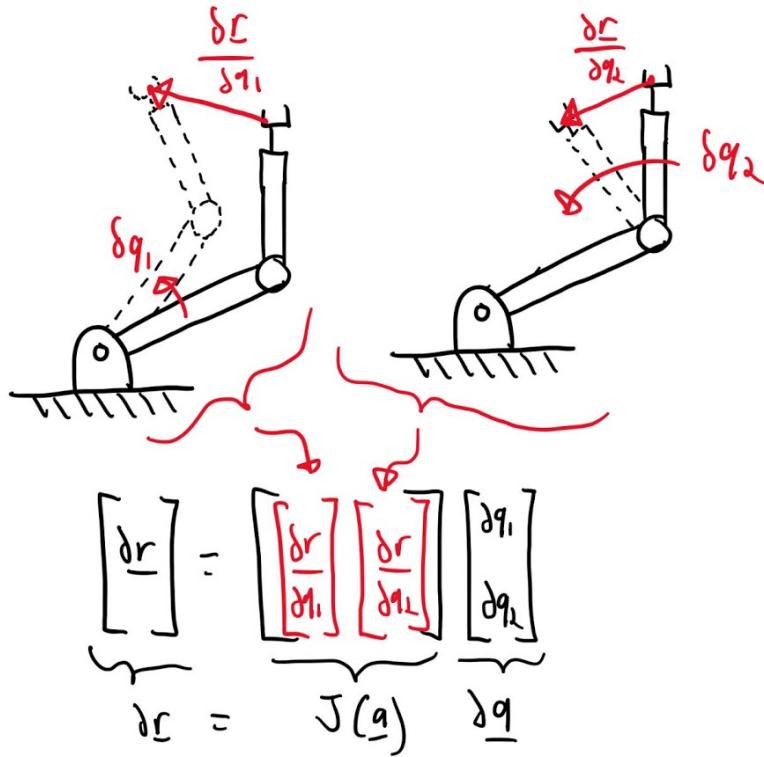


FIGURE 4.5 – Illustration vectorielle du Jacobian du manipulateur (relation de translation)

Produits vectoriels Lorsque l'espace de la tâche associé à un Jacobien correspond à la position cartésienne de l'effecteur d'un robot, chaque colonne du Jacobien peut aussi être déterminée basé sur un calcul vectoriel avec l'axe de rotation du joint associé. Si on considère d'abord un joint prismatique i , l'effet infinitésimal de son déplacement est une translation d'amplitude ∂q_i dans une direction qui correspond à l'axe de translation du joint \hat{z}_i . Pour un joint rotatif i , l'effet est plutôt une rotation des membrures en aval du joint i autour de l'axe de rotation du joint \hat{z}_i . Si on utilise un repère mobile qui tourne avec toutes les membrures en aval du joint i , les composantes de vecteurs position en aval ne varie pas, c'est la vitesse angulaire de la base vectorielle mobile qui explique la vitesse. Donc comme vu à la section 4.2.1, cet effet peut être représenté mathématiquement par un produit vectoriel. Le mouvement à l'effecteur (résultant de la rotation d'un axe seulement) peut donc ici être calculé par le produit vectoriel du vecteur de vitesse angulaire (l'axe de rotation fois le taux de variation de l'angle de rotation) avec le vecteur position entre l'effecteur et le centre de rotation du joint. En termes de déplacements infinitésimaux de l'effecteur dus aux déplacements de joints, la relation est

$$\partial \vec{r} = \begin{cases} (\partial q_i) \hat{z}_i & \text{pour un joint } i \text{ prismatique} \\ (\partial q_i) \hat{z}_i \times \vec{r}_{T_o/Z_i} & \text{pour un joint } i \text{ rotatif} \end{cases} \quad (4.34)$$

où q_i est la position du joint i , \hat{z}_i est l'axe de rotation ou de translation du joint, \vec{r} est le vecteur position de l'effecteur par rapport à une origine fixe et \vec{r}_{T_o/Z_i} est le vecteur position de l'effecteur par rapport à un point d'origine Z_i sur l'axe de rotation du joint i (centre de rotation de ce mouvement). Cette relation vectorielle différentielle peut être utilisée pour obtenir une définition équivalente des colonnes du Jacobien :

$$\text{Colonnes du Jacobien : } \frac{\partial \vec{r}}{\partial q_i} = \begin{cases} \hat{z}_i & \text{pour un joint } i \text{ prismatique} \\ \hat{z}_i \times \vec{r}_{T_o/Z_i} & \text{pour un joint } i \text{ rotatif} \end{cases} \quad (4.35)$$

ou toutes les composantes sont exprimées dans la base vectorielle fixe globale.

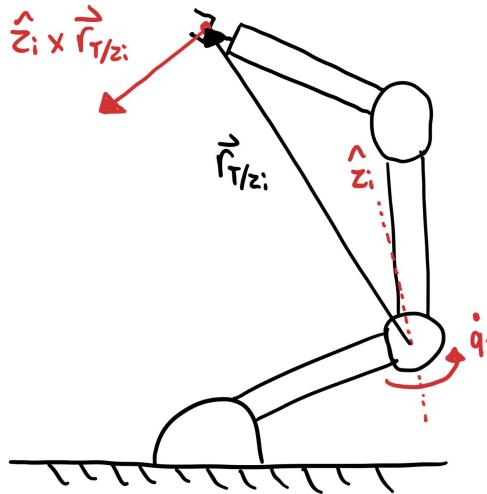


FIGURE 4.6 – Interprétation des colonnes du Jacobian comme un produit vectoriel

Relation avec les paramètres DH La définition des colonnes du Jacobien donnée à l'équation (4.35) est particulièrement utile pour calculer le Jacobien d'un robot manipulateur lorsqu'on a déjà déterminé les paramètres DH. La convention de positionnement des repères fait que les axes de rotation \hat{z}_i et les vecteurs positions qui donnent la position des points Z_i sont déjà calculés dans les matrices de transformation homogènes. Dans ce cas, cette technique peut être plus rapide que de calculer les dérivées partielles basée sur la définition donnée à l'équation (4.30) pour construire la matrice Jacobienne.

Cinématique différentielle pour l'orientation de l'effecteur Lorsque ce n'est pas la position cartésienne de l'effecteur que l'on désire contrôler mais plutôt son orientation, le vecteur sortie $\dot{\underline{r}}$ contient alors les composantes du vecteur de vitesse angulaire de l'effecteur qui définit le taux de variation de l'orientation d'une base vectorielle t fixée sur l'effecteur par rapport à une base fixe a par rapport à la base du robot manipulateur. L'équation de cinématique différentiel est alors caractérisé par l'addition des vecteurs de vitesse angulaire relative pour chacun des joints d'un système, basée sur l'équation (4.22), pour un robot à 6 joints :

$$\vec{w}_{t/w} = \vec{w}_{t/f} + \vec{w}_{f/e} + \vec{w}_{e/d} + \vec{w}_{d/c} + \vec{w}_{c/b} + \vec{w}_{b/a} \quad (4.36)$$

où les bases b, c, d, e et f sont des bases vectorielles intermédiaires sur les joints (voir Figure 3.30). Donc le cas commun où le robot manipulateur a seulement des joints rotatifs à un degré de liberté, chacun vecteur de vitesse angulaire est simplement l'axe de rotation \hat{z}_i fois de taux de variation de l'angle :

$$\vec{w}_{t/w} = \hat{z}_6\dot{q}_6 + \hat{z}_5\dot{q}_5 + \hat{z}_4\dot{q}_4 + \hat{z}_3\dot{q}_3 + \hat{z}_2\dot{q}_2 + \hat{z}_1\dot{q}_1 \quad (4.37)$$

$$\underline{w} = \underbrace{[z_1 z_2 z_3 z_4 z_5 z_6]}_{J_{ori}} \dot{\underline{q}} \quad (4.38)$$

Dans ce cas, les colonnes du Jacobien sont simplement les axes de rotation de chacun des joints. Pour des joints prismatiques, ils n'influencent pas l'orientation de l'effecteur et les colonnes du Jacobien associées à un joint prismatique sont donc nulles :

$$\text{Colonnes du Jacobien (orientation)} : \quad \begin{cases} 0 & \text{pour un joint } i \text{ prismatique} \\ z_i & \text{pour un joint } i \text{ rotatif} \end{cases} \quad (4.39)$$

Un exemple où on s'intéresserait plutôt à l'orientation de système robotique, serait pour asservir une antenne de communication qui doit toujours être alignée avec un satellite. Dans ce cas, l'effecteur du système est la parabole de l'antenne et c'est son orientation spatiale qui est importante seulement.

Cinématique différentielle pour la pose de l'effecteur Si on désire décrire de taux de variation de la pose complète de l'effecteur, il suffit de combiner les systèmes linéaire la translation et l'orientation :

$$\dot{\underline{r}} = [J_{trans}] \dot{\underline{q}} \quad J_{trans} = [\dots \hat{z}_i^\times r_{T_o/Z_i} \dots] \quad (4.40)$$

$$\underline{w} = [J_{ori}] \dot{\underline{q}} \quad J_{ori} = [\dots \hat{z}_i \dots] \quad (4.41)$$

$$\begin{bmatrix} \dot{\underline{r}} \\ \underline{w} \end{bmatrix} = [J_{pose}] \dot{\underline{q}} \quad J_{pose} = \begin{bmatrix} J_{trans} \\ J_{ori} \end{bmatrix} \quad (4.42)$$

Cinématique différentielle pour un espace de la tâche Il est aussi possible de définir une matrice Jacobienne pour une relation spécifique à une tâche. Par exemple, voir Figure 3.4, où il serait désirable d'obtenir les équations qui relit la position/vitesse des joints et celle de la tâche décrite en termes de x et y sur un plan particulier. Si on a un espace de la tâche de dimension m (normalement égale ou inférieure au nombre de DDL de l'effecteur du robot), alors on peut déterminer matrice Jacobienne m par n .

$$\begin{bmatrix} \underline{r}_e \end{bmatrix}_{m \times 1} = f_e(\underline{q}) \quad \Rightarrow \quad \begin{bmatrix} \dot{\underline{r}}_e \end{bmatrix}_{m \times 1} = \begin{bmatrix} J_e \end{bmatrix}_{m \times n} \begin{bmatrix} \dot{\underline{q}} \end{bmatrix}_{n \times 1} \quad (4.43)$$

Note de lecture Pour les sections suivantes, si l'on n'est pas préciser autrement on va toujours travailler avec la relation cinématique de translation seulement de l'effecteur pour l'explication des concepts. Les exemples sont visuellement plus clairs dans cette situation. Toutefois, tous les concepts présentés se généralisent à toutes les situations.



Capsule vidéo
Exemple de calcul de la matrice jacobienne
<https://youtu.be/qqqDscXSsMM>

Exemple 1. Cinématique différentielle d'un robot cartésien

La Figure 4.7 illustre un robot planaire constitué de deux actionneurs linéaires basé sur des vis à billes qui sont assemblés à 90 degrées.

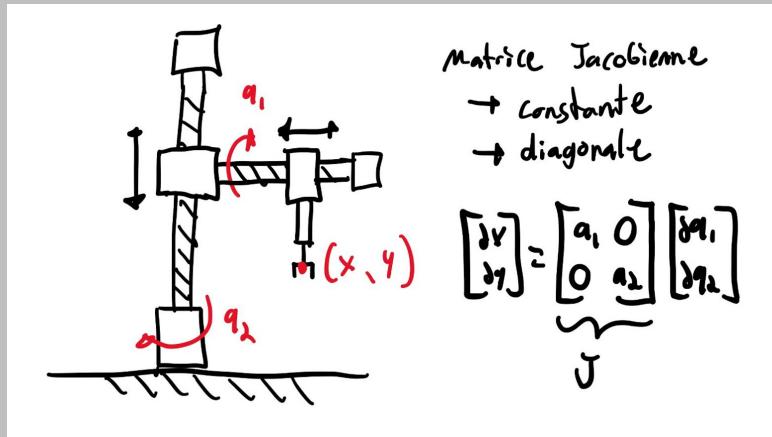


FIGURE 4.7 – Jacobien pour un robot cartésien

Pour ce robot, chaque actionneur linéaire contrôle de façon indépendante la translation x et y de l'effecteur, la fonction de cinématique directe est donnée par :

$$\underline{r} = f(\underline{q}) \quad (4.44)$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_1 q_1 \\ a_2 q_2 \end{bmatrix} \quad (4.45)$$

ou les variables a_i sont des constantes qui relient la rotation angulaire des vis aux déplacements linéaires. La relation différentielle peut alors être calculée :

$$\underbrace{\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}}_{\dot{\underline{r}}} = \underbrace{\begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} \end{bmatrix}}_J \underbrace{\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}}_{\dot{\underline{q}}} \quad (4.46)$$

$$\underbrace{\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}}_{\dot{\underline{r}}} = \underbrace{\begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix}}_J \underbrace{\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}}_{\dot{\underline{q}}} \quad (4.47)$$

(4.48)

Il est à noter ici que : 1) la matrice Jacobienne est diagonale car chaque joint influence indépendamment un seul DDL de l'effecteur. 2) la matrice Jacobienne est constante et indépendante de \underline{q} car la relation de cinématique directe, voir équation (4.45), est linéaire.

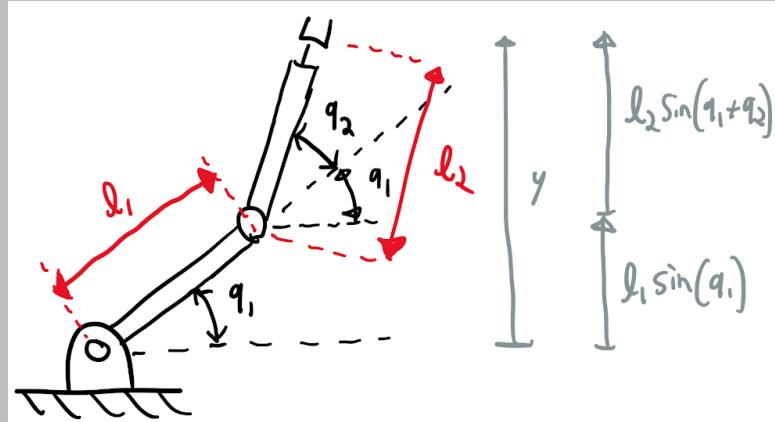
Exemple 2. Cinématique différentielle d'un robot à deux joints

FIGURE 4.8 – Robot à deux joints

Pour un robot manipulateur planaire à deux DDL, comme illustré à la Figure 4.8, la fonction de cinématique directe est données par :

$$\underline{r} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} l_1 c_1 + l_2 c_{12} \\ l_1 s_1 + l_2 s_{12} \end{bmatrix} \quad (4.49)$$

avec

$$c_1 = \cos(q_1) \quad (4.50)$$

$$s_1 = \sin(q_1) \quad (4.51)$$

$$c_{12} = \cos(q_1 + q_2) \quad (4.52)$$

$$s_{12} = \sin(q_1 + q_2) \quad (4.53)$$

On peut alors trouver la relation différentielle en dérivant par rapport au temps :

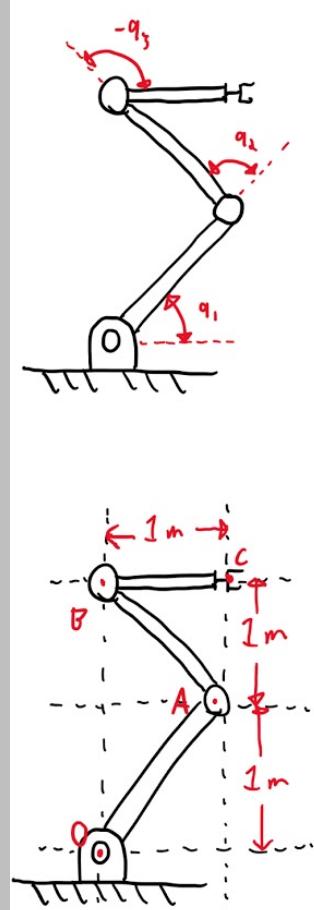
$$\dot{\underline{r}} = \frac{d\underline{r}}{dt} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -l_1 s_1 \dot{q}_1 - l_2 s_{12}(\dot{q}_1 + \dot{q}_2) \\ l_1 c_1 \dot{q}_1 + l_2 c_{12}(\dot{q}_1 + \dot{q}_2) \end{bmatrix} \quad (4.54)$$

$$\dot{\underline{r}} = \underbrace{\begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix}}_{J(\underline{q})} \underbrace{\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}}_{\dot{\underline{q}}} \quad (4.55)$$

$$\dot{\underline{r}} = J(\underline{q})\dot{\underline{q}} \quad (4.56)$$

Exemple 3. Cinématique différentielle d'un robot à trois joints

Les Figures 79 et 4.10 illustre la cinématique différentielle d'un robot à trois joints dans une configuration qui permet de bien visualiser indépendamment les effets du mouvement de chaque joint.



$$\begin{aligned}
 \begin{bmatrix} X \\ Y \end{bmatrix} &= \begin{bmatrix} l_1 c_1 + l_2 c_{12} + l_3 c_{123} \\ l_1 s_1 + l_2 s_{12} + l_3 s_{123} \end{bmatrix} \\
 J &= \begin{bmatrix} \frac{\partial X}{\partial q_1} & \frac{\partial X}{\partial q_2} & \frac{\partial X}{\partial q_3} \\ \frac{\partial Y}{\partial q_1} & \frac{\partial Y}{\partial q_2} & \frac{\partial Y}{\partial q_3} \end{bmatrix} \\
 &= \begin{bmatrix} -[l_1 s_1 + l_2 s_{12} + l_3 s_{123}] & -[l_2 s_{12} + l_3 s_{123}] & -[l_3 s_{123}] \\ [l_1 c_1 + l_2 c_{12} + l_3 c_{123}] & [l_2 c_{12} + l_3 c_{123}] & [l_3 c_{123}] \end{bmatrix} \\
 &= \begin{bmatrix} -y_{c/A} & -y_{c/B} & -y_{c/C} \\ x_{c/A} & x_{c/B} & x_{c/C} \end{bmatrix} \\
 \text{avec} \\
 l_1 &= \sqrt{2} \quad l_2 = \sqrt{2} \quad l_3 = 1 \\
 q_1 &\approx 45^\circ \quad q_2 \approx 90^\circ \quad q_3 \approx -135^\circ \\
 &= \begin{bmatrix} -2 & -1 & 0 \\ 1 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

FIGURE 4.9 – Jacobien d'un robot à trois joints - partie a)

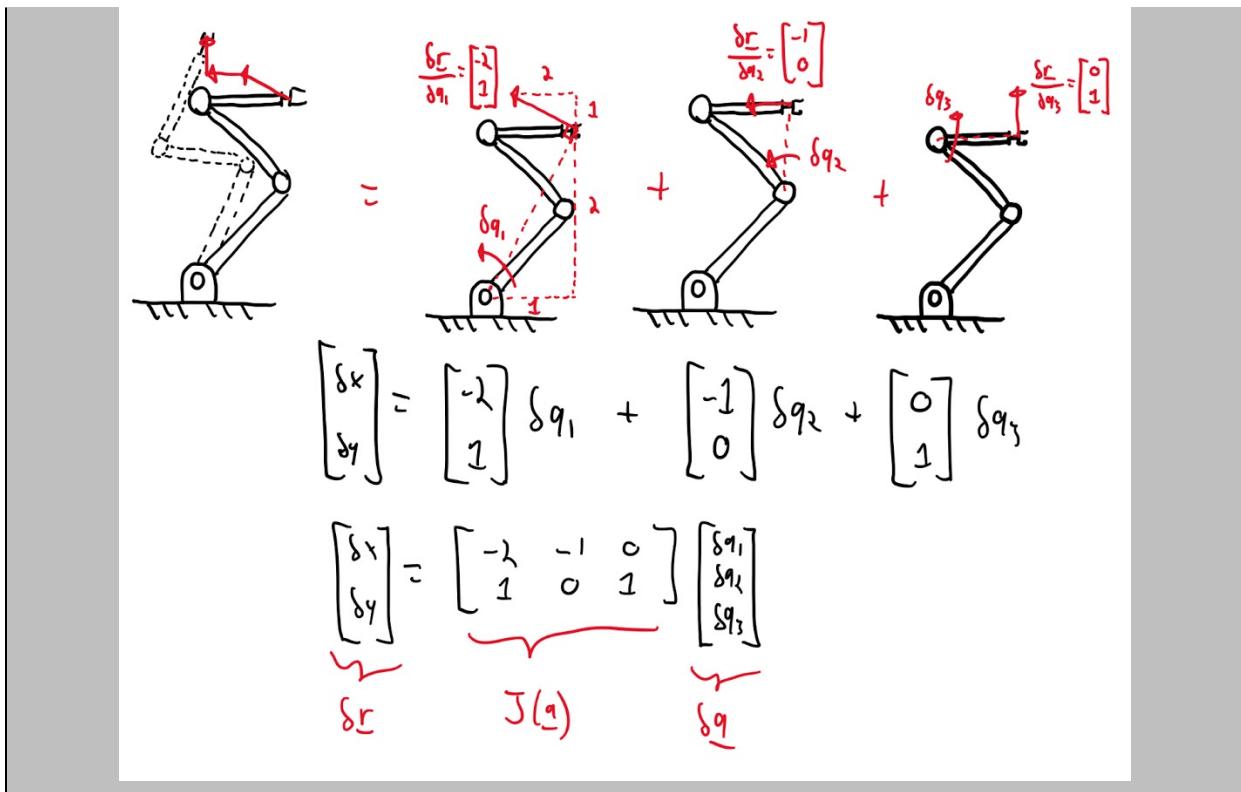


FIGURE 4.10 – Jacobien d'un robot à trois joints - partie b)

4.6 Cinématique différentielle inverse

Lorsque le problème de cinématique différentielle est inversé, c'est-à-dire lorsqu'on cherche un vecteur de vitesse aux joints $\dot{\underline{q}}$ qui va produire une vitesse de l'effecteur spécifiée $\dot{\underline{r}}$, il faut inverser la relation mathématique. Lorsque la matrice Jacobienne est carré et non-singulière, il suffit de multiplier la relation $\dot{\underline{r}} = J(\underline{q}) \dot{\underline{q}}$ par l'inverse du Jacobien des deux côtés de l'équation pour obtenir la relation :

$$\dot{\underline{q}} = J^{-1}(\underline{q}) \dot{\underline{r}} \quad (4.57)$$

Toutefois, cette opération est impossible si le robot a une configuration singulière (voir section 4.6.1). De plus, si les dimensions de \underline{q} et \underline{r} sont différentes, la matrice Jacobienne est rectangulaire et d'autres techniques doivent être utilisées (voir sections 4.6.2 et 4.6.3)



Capsule vidéo

Cinématique différentielle inverse des robots manipulateurs

<https://youtu.be/YqYV10mVVBI>

4.6.1 Les singularités

Certaines configurations \underline{q} d'un robot manipulateur peuvent être singulières, c'est-à-dire des configurations pour lesquelles il est impossible d'inverser la matrice Jacobienne. Les configurations singulières sont l'ensemble des configurations pour lesquelles le déterminant de la matrice Jacobienne est nul :

$$\text{Singularités : } \{\underline{q} \mid \det(J(\underline{q})) = 0\} \quad (4.58)$$

Ces situations correspondent aussi à lorsque les colonnes de J sont co-linéaires, et qu'aucune combinaison linéaire ne permet de déplacer l'effecteur dans au moins une direction (la matrice jacobienne a un espace-nul gauche, voir section 18.2.2 pour les notions d'algèbre linéaire associées).

Exemple 4. Singularités d'un robot manipulateur planaire à 2 joints

Pour un robot manipulateur à deux DDL comme décrit à l'exemple 2, le Jacobien est donné par :

$$J = \begin{bmatrix} -(l_1 s_1 + l_2 s_{12}) & -l_2 s_{12} \\ (l_1 c_1 + l_2 c_{12}) & l_2 c_{12} \end{bmatrix} \quad (4.59)$$

Le déterminant de la matrice est donnée par :

$$\det(J) = l_2 s_{12} (l_1 c_1 + l_2 c_{12}) - l_2 c_{12} (l_1 s_1 + l_2 s_{12}) \quad (4.60)$$

$$= l_1 l_2 s_{12} c_1 + l_2^2 s_{12} c_{12} - l_1 l_2 s_1 c_{12} - l_2^2 s_{12} c_{12} \quad (4.61)$$

$$= l_1 l_2 (s_1 c_{12} - c_1 s_{12}) \quad (4.62)$$

$$= l_1 l_2 s_2 \quad (4.63)$$

Donc les singularités sont les configurations avec un angle q_2 égale à 0 ou 180 degrés. Comme illustré à la Figure 4.11, ces configurations correspondent à lorsque les deux liens rigides du robot sont alignés. Dans ces configurations, il est seulement possible de faire bouger l'effecteur dans une direction perpendiculaire aux liens rigides. Comme illustré à la Figure 4.12, la direction pour laquelle le mouvement est possible correspond à l'espace colonne de la matrice Jacobienne, et la direction orthogonale pour laquelle il est impossible de déplacer l'effecteur correspond au left-nullspace de la matrice Jacobienne, voir section 18.2 pour les notions d'algèbre linéaire associées.

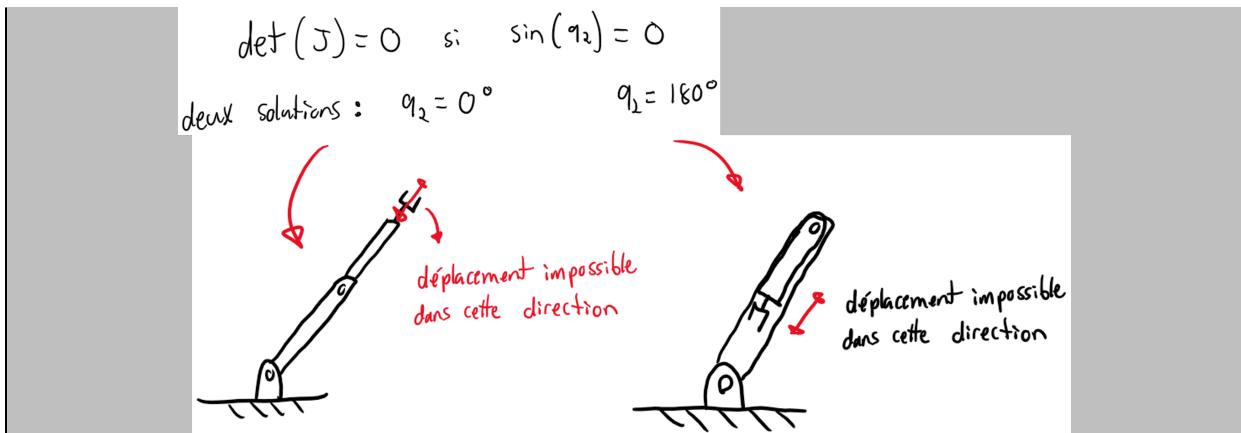


FIGURE 4.11 – Singularités pour un robot manipulateur à deux DDL

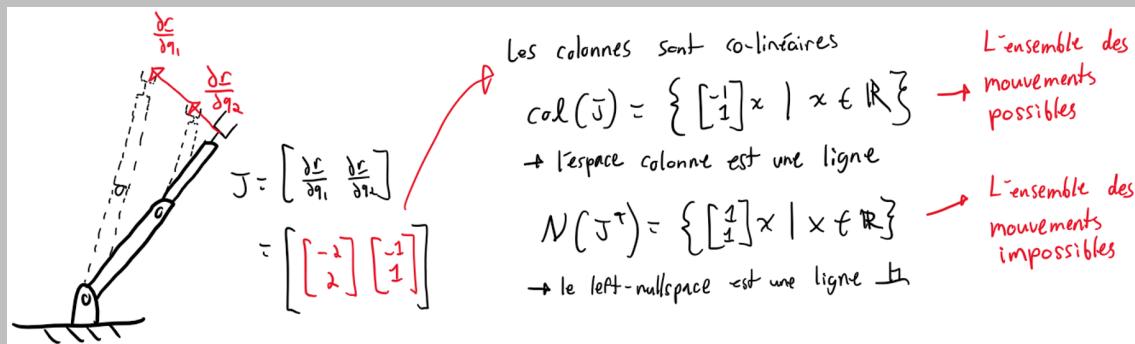


FIGURE 4.12 – Co-linéarité des colonnes du Jacobien pour une singularité

Limites de l'espace de travail

Un robot va nécessairement être sur une singularité aux limites de son espace de travail, car par définition le robot est incapable de bouger son effecteur dans la direction normale à la surface qui définit la limite de l'espace de travail (sinon le robot pourrait sortir de l'espace de travail ce qui est une contradiction). Par exemple, les deux singularités du robot 2DDL de l'exemple précédent correspondent à des configurations où l'effecteur est à la limite intérieure de l'espace de travail ou à la limite extérieure de l'espace de travail.

Types de singularités

Détails à venir !

4.6.2 Cinématique différentielle inverse d'un robot redondant (situations sous-constraines)

Si le nombre d'entrées n est plus grand que le nombre de sorties m :

$$n > m \quad (4.64)$$

avec

$$n = \dim(\underline{q}) = \text{Nombre de DDL du robot} \quad (4.65)$$

$$m = \dim(\underline{r}) = \text{Coordonnées de l'espace de la tâche} \quad (4.66)$$

le robot est dit redondant, et le problème de cinématique différentiel inverse est sous-constraint : il y a plusieurs solutions possibles de déplacement des joints qui produisent un même déplacement de l'effecteur. Le Jacobien d'une telle situation est rectangulaire, il a plus de colonnes que de rangées :

$$\begin{bmatrix} \dot{\underline{r}} \\ \vdots \end{bmatrix}_{m \times 1} = \begin{bmatrix} & J(\underline{q}) \\ & \vdots \end{bmatrix}_{m \times n} \begin{bmatrix} \dot{\underline{q}} \\ \vdots \end{bmatrix}_{n \times 1} \quad (4.67)$$

Cette situation nous mène à un problème d'algèbre linéaire sous-constraint qui peut être résolu avec une méthode qui utilise une matrice pseudo inverse, voir section 18.4 pour les notions d'algèbre linéaire. Cette situation est aussi caractérisée par la présence d'un nullspace : une combinaison de déplacement des joints peut avoir un effet net nul sur l'effecteur, voir section 18.2.3 pour les notions d'algèbre linéaire et l'exemple 18.7.

Une méthode pour travailler avec ce genre de situation consiste à utiliser l'équation de cinématique inverse suivante :

$$\begin{bmatrix} \dot{\underline{q}} \\ \vdots \end{bmatrix}_{n \times 1} = \underbrace{\begin{bmatrix} & J^\# \\ & \vdots \end{bmatrix}_{n \times m}}_{\text{Pseudo-inverse}} \begin{bmatrix} \dot{\underline{r}} \\ \vdots \end{bmatrix}_{m \times 1} + \underbrace{\begin{bmatrix} & I - J^\# J \\ & \vdots \end{bmatrix}_{n \times n}}_{\text{Projection sur le Nullspace}} \begin{bmatrix} \underline{\psi} \\ \vdots \end{bmatrix}_{n \times 1} \quad (4.68)$$

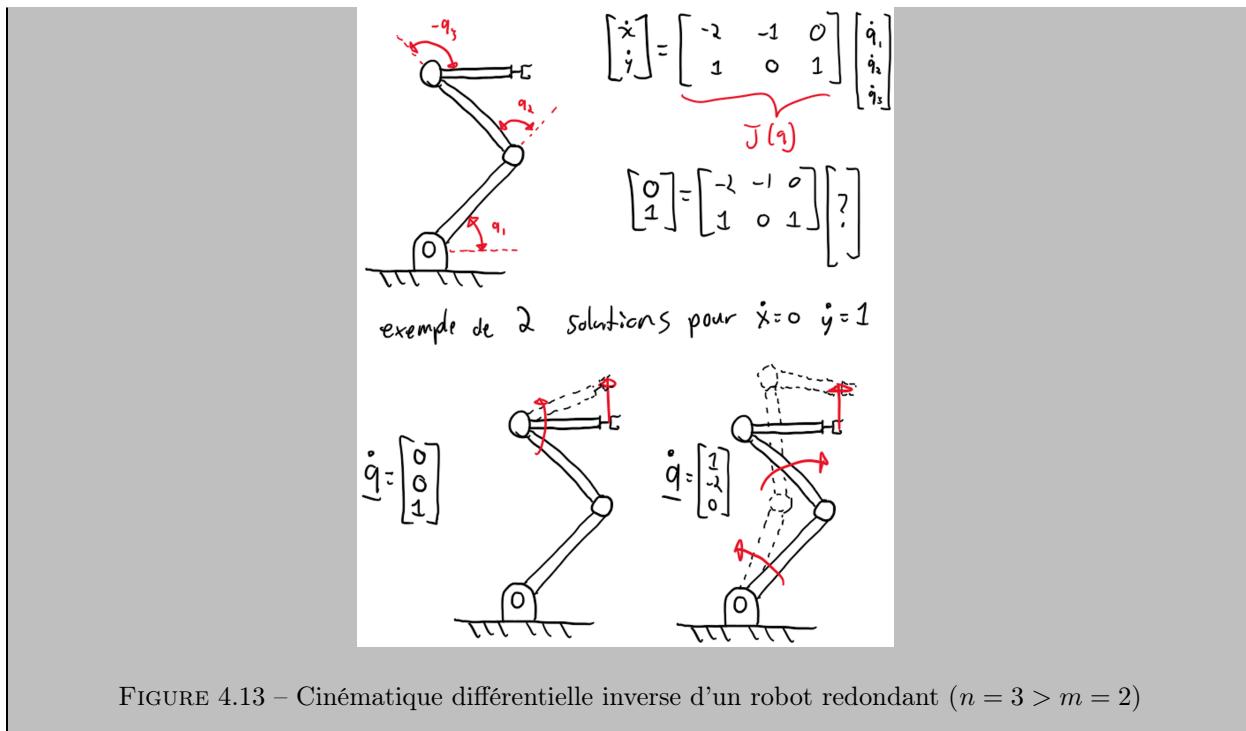
avec $J^\#$ qui est la matrice pseudo-inverse droite de Moore–Penrose, définie par :

$$J^\# = J^T (JJ^T)^{-1} \quad (4.69)$$

Le premier terme $J^\#\dot{\underline{r}}$ produit un vecteur $\dot{\underline{q}}$ qui produit une solution exacte pour le système d'équations (4.67) et minimise la norme du vecteur-colonne $\dot{\underline{q}}$. Le second terme produit une variation pour le vecteur $\dot{\underline{q}}$ qui réside dans l'espace nul de la matrice J , donc cette variation n'a aucun influence sur la sortie et n'affectera pas l'objectif principal d'obtenir un vecteur $\dot{\underline{q}}$ qui produit un mouvement de l'effecteur donné par \underline{r} . Le vecteur $\underline{\psi}$ dans le second terme est arbitraire et peut être utilisé pour prioriser certaines solutions sans influencer la justesse de la solution.

Exemple 5. Exemple de cinématique différentielle inverse pour une situation sous-constrainte

Le robot de l'exemple 3 est ici analysé en termes de cinématique différentielle, la Figure 4.13 illustre la présence de plusieurs solutions possibles et la Figure 4.14 illustre l'utilisation de la matrice pseudo-inverse pour obtenir la solution optimale ainsi que le nullspace.

FIGURE 4.13 – Cinématique différentielle inverse d'un robot redondant ($n = 3 > m = 2$)

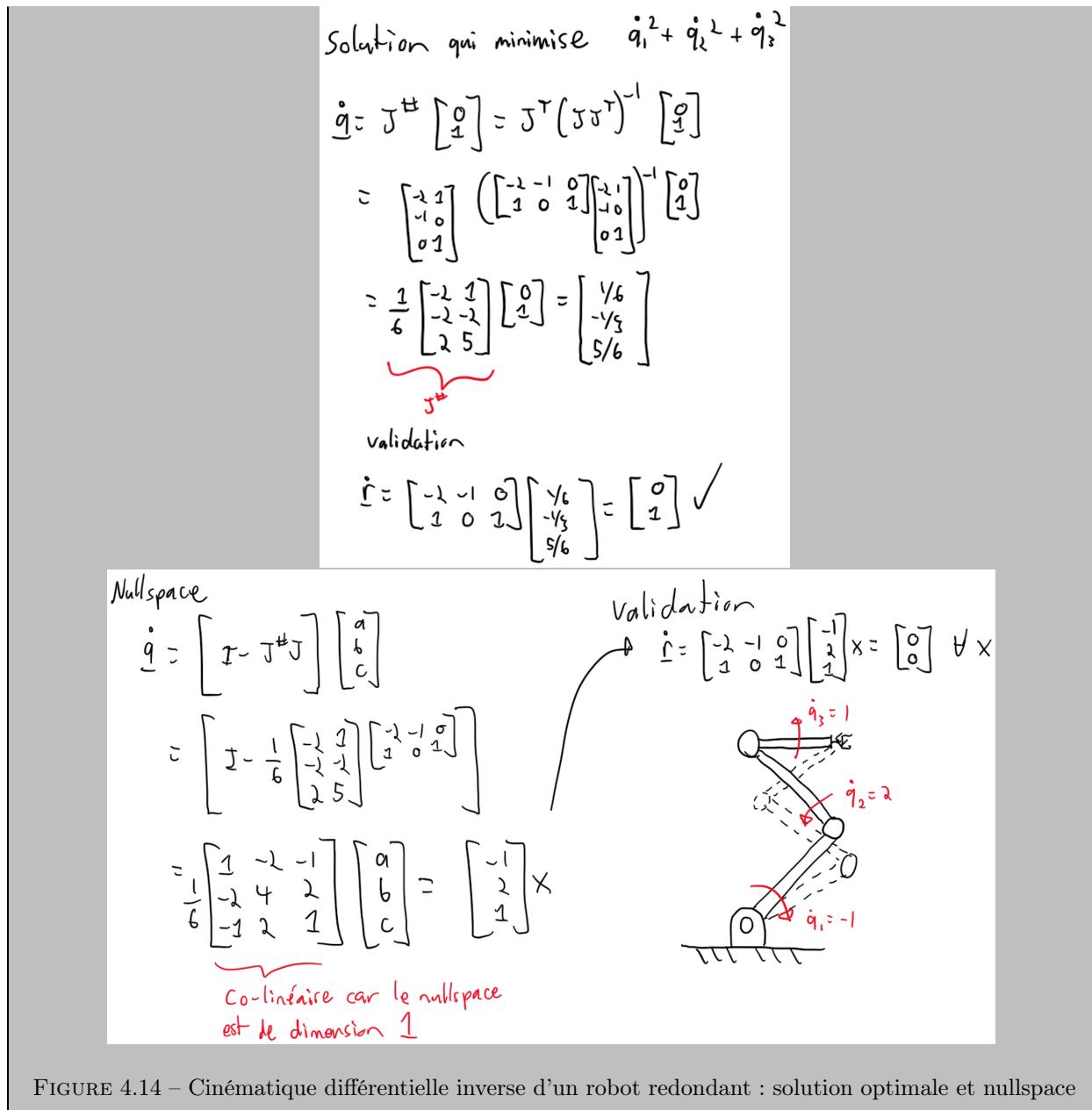


FIGURE 4.14 – Cinématique différentielle inverse d'un robot redondant : solution optimale et nullspace

4.6.3 Cinématique différentielle inverse pour une situation sur-contrainte

Si le nombre d'entrées n est plus petit que le nombre de sorties m :

$$n < m \tag{4.70}$$

avec

$$n = \dim(\underline{q}) = \text{Nombre de DDL du robot} \tag{4.71}$$

$$m = \dim(\underline{r}) = \text{Coordonnées de l'espace de la tâche} \tag{4.72}$$

le problème de cinématique différentiel inverse est sur-contraint : il n'y a pas de solutions exactes possibles sauf pour un sous-ensemble de déplacements de l'effecteur. Le Jacobien d'une telle situation est rectangulaire,

il a plus de rangées que de colonnes :

$$\begin{bmatrix} \dot{\underline{r}} \\ \vdots \end{bmatrix}_{m \times 1} = \begin{bmatrix} J(\underline{q}) \\ \vdots \end{bmatrix}_{m \times n} \begin{bmatrix} \dot{\underline{q}} \\ \vdots \end{bmatrix}_{n \times 1} \quad (4.73)$$

Un méthode pratique pour la cinématique inverse dans ce contexte est la méthode des moindres-carré, voir la section 18.3 pour les notions d'algèbre linéaire. L'utilisation de cette méthode permet de calculer explicitement un vecteur solution approximé $\hat{\underline{q}}$ qui minimise la norme de l'erreur entre une vitesse à l'effecteur désiré et celle obtenue avec la solution approximée. La solution au sens des moindres-carré est calculée ainsi :

$$\hat{\underline{q}} = \underset{\underline{q}}{\operatorname{argmin}} \|J\underline{q} - \dot{\underline{r}}\|^2 = (J^T J)^{-1} J^T \dot{\underline{r}} \quad (4.74)$$

Exemple 6. Exemple de cinématique différentielle inverse pour une situation sur-contrainte

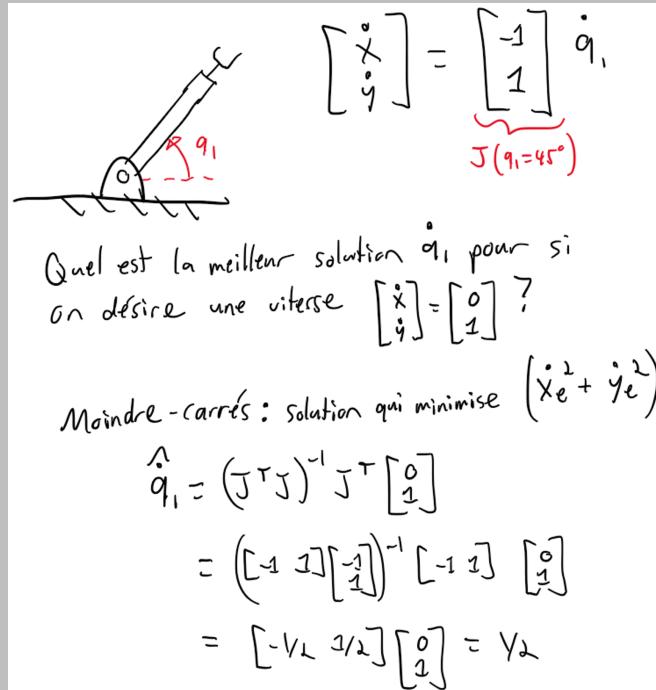


FIGURE 4.15 – Cinématique différentielle inverse pour une situation sur-contrainte ($n = 1 < m = 2$)

4.7 Manipulabilité d'un robot manipulateur

Lorsqu'on désire effectuer une tâche avec l'effecteur d'un robot à une certaine position, la première question est : est-ce que le robot est capable d'atteindre cette position, i.e. est-ce que la position cible est dans l'espace de travail ? Si la position est atteignable, une deuxième question doit être posée : est-ce que le robot est capable de bouger arbitrairement à cette position ? Cette deuxième question est la notion de manipulabilité du robot manipulateur.

Comme discuté à la section 4.6.1, lorsque le robot est à une configuration singulière l'effecteur est incapable de bouger dans certaines directions. On considère donc que la manipulabilité d'un robot n'est pas bonne sur une singularité. Toutefois, il est utile d'avoir une notion non-binaire de la manipulabilité, car même si un robot n'est pas parfaitement sur une singularité, certaines configurations peuvent nécessiter de très grandes vitesses aux joints pour satisfaire des vitesses cartésiennes de l'effecteur. Donc en considérant les limites de vitesse des joints le robot est en fait incapable d'exécuter certain vecteur vitesse à son effecteur proche d'une singularité. Les indices de manipulabilité sont des fonctions scalaires qui ont comme objectif de caractériser si un robot dans une certaine configuration est proche d'une singularité. Ces indicateurs peuvent être utile lorsque qu'on conçoit la cinématique d'un robot ou lorsque qu'on programme des tâches avec un robot existant.

Indices de manipulabilité d'un robot manipulateur

Pour les situations où la matrice jacobienne J du robot est carré, le déterminant est un bon indice de la manipulabilité du robot :

$$m_1(\underline{q}) = \det(J(\underline{q})) \quad (4.75)$$

Le déterminant correspond au volume du prisme généré par les colonnes du jacobien qui correspondent à des vecteurs vitesses de l'effecteur pour des vitesses de joints unitaires de chacun des axes :

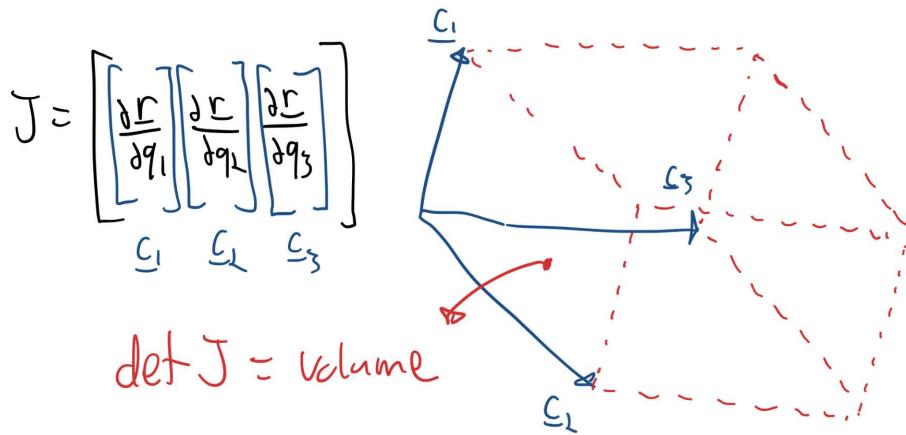


FIGURE 4.16 – Volume de la matrice jacobienne

Dès que un des vecteurs est très petit, ou bien quand les vecteurs sont colinéaires, le volume du prisme devient très petit ce qui correspond à une mauvaise manipulabilité.

Un autre indicateur souvent utilisé est les valeurs singulières de la matrice jacobienne. Les valeurs singulières μ sont reliées aux valeurs propres λ du produit $J^T J$

$$\mu_i^2(J(\underline{q})) = \lambda_i(J(\underline{q})^T J(\underline{q})) \quad (4.76)$$

Pour être relié à un indice de manipulabilité, on peut surveiller la plus petite valeur singulière ou bien le ratio

entre la plus petite et la plus grande :

$$m_2(\underline{q}) = \mu_{\min}(J(\underline{q})) \quad (4.77)$$

$$m_3(\underline{q}) = \frac{\mu_{\min}(J(\underline{q}))}{\mu_{\max}(J(\underline{q}))} \quad (4.78)$$

$$(4.79)$$

La valeur μ_{\min} correspond à la norme minimal du vecteur vitesse de l'effecteur produit par un vecteur de vitesse aux joints unitaire.

Ellipsoïde de vitesse

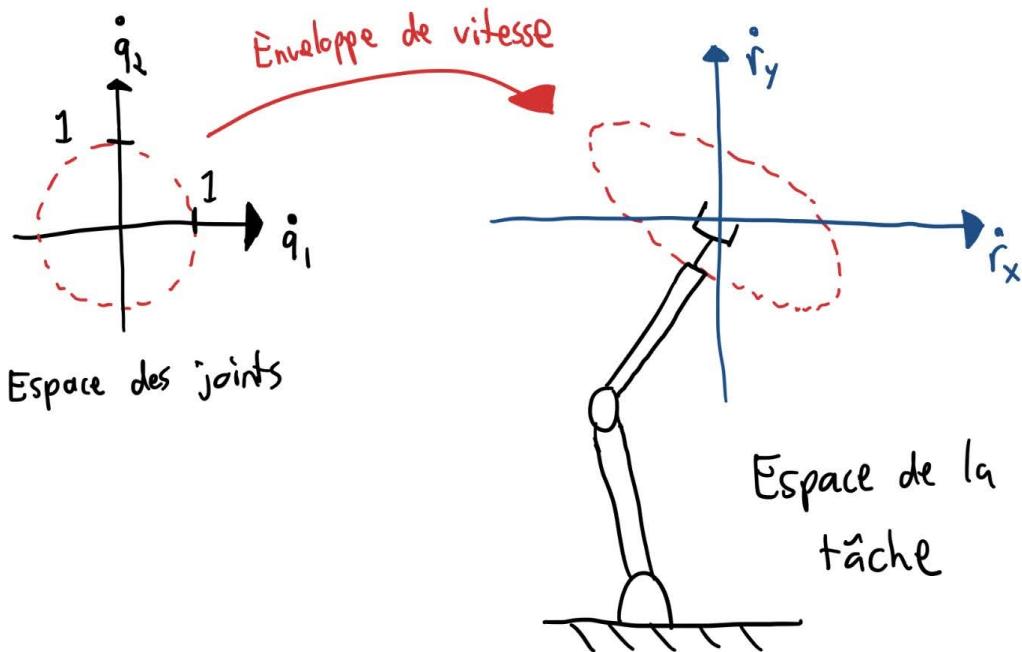


FIGURE 4.17 – Ellipsoïde de vitesse

L'ellipsoïde de vitesse d'un robot manipulateur est l'enveloppe des vecteurs vitesses possibles à l'effecteur pour des vecteurs de vitesse des joints unitaires, voir 4.17. L'enveloppe de vitesses possibles à l'effecteur, résultant d'un vecteur de vitesse des joints unitaire, peut être calculer en débutant avec une équation qui contraint la norme du vecteur de la vitesse des joint et en substituant $\dot{\underline{q}}$ par $J^{-1}\dot{\underline{r}}$ dans l'équation :

$$1 = \dot{\underline{q}}^T \dot{\underline{q}} \quad (4.80)$$

$$1 = \dot{\underline{r}}^T J^{-T} J^{-1} \dot{\underline{r}} \quad (4.81)$$

$$1 = \dot{\underline{r}}^T (JJ^T)^{-1} \dot{\underline{r}} \quad (4.82)$$

$$1 = \dot{\underline{r}}^T A^{-1} \dot{\underline{r}} \quad \text{avec} \quad A = JJ^T \quad (4.83)$$

On obtient donc une équation quadratique pour laquelle l'ensemble des solutions possibles forment une ellipse. Les axes principaux de l'ellipse correspondent aux valeurs propres et vecteurs propres de la matrice JJ^T .

FIGURE 4.18 – TODO : ellipse 3D

Un lien peu aussi être fait directement avec les valeurs singulières de la matrice J . Détails à venir !

Exemple 7. Indice de manipulabilité pour un robot à 2 DDL

Une exemple simple pour illustrer l'indice de manipulabilité est un robot planaire à 2 DDL comme illustré à la Figure 4.19, pour lequel on peut analyser la manipulabilité pour divers position de l'effecteur sur l'axe horizontal. La manipulabilité est nulle lorsque le coude est complètement plié (le robot est sur une singularité et ne peut bouger horizontalement), nulle lorsque le coude est complètement déplié (le robot est sur une singularité de ne peut bouger horizontalement) et optimale entre les deux lorsque le coude est partiellement plié.

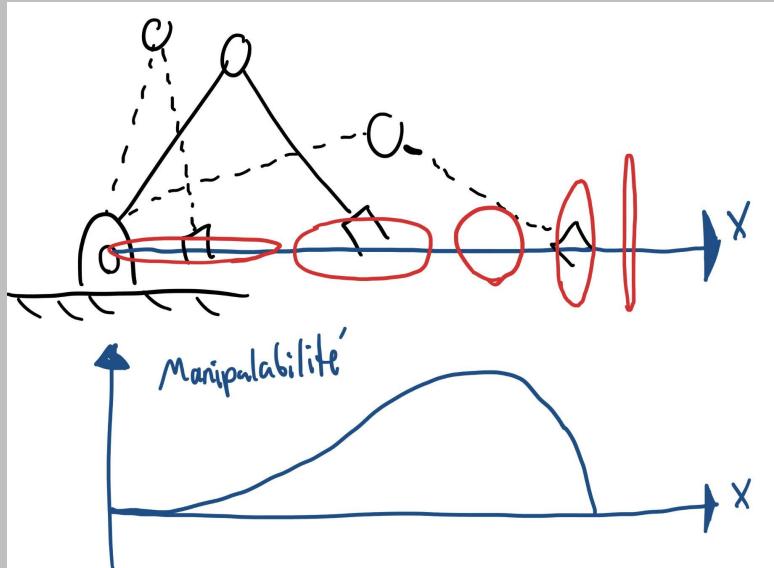


FIGURE 4.19 – Exemple de l'indice de manipulabilité pour un robot à 2 DDL

4.8 Relation entre les variables d'accélération

Pour faire le lien en l'accélération dans l'espace des joints et l'accélération dans l'espace de la tâche, il suffit de dériver par rapport au temps l'expression de la cinématique différentielle, autrement dit on dérive deux fois par rapport au temps l'équation de la cinématique directe :

$$\text{ Cinématique directe : } \underline{r} = f(\underline{q}) \quad (4.84)$$

$$\text{ Cinématique différentielle : } \dot{\underline{r}} = J(\underline{q}) \dot{\underline{q}} \quad (4.85)$$

$$\text{ Cinématique différentielle ordre-2 : } \ddot{\underline{r}} = J(\underline{q}) \ddot{\underline{q}} + \dot{J}(\underline{q}, \dot{\underline{q}}) \dot{\underline{q}} \quad (4.86)$$

Comme on dérive un produit avec deux termes qui varient dans le temps, l'accélération $\ddot{\underline{r}}$ est le résultat de deux effets représentés par deux termes. Le premier terme est simplement l'accélération des joints fois le Jacobien, i.e. les bras de leviers. Le deuxième terme est une contribution à l'accélération de l'effecteur qui vient de la variation temporelle du Jacobien. Comme le Jacobien a seulement une dépendance directe aux positions \underline{q} , on peut exprimer son taux de variation temporelle comme sa dérivée par rapport à \underline{q} fois le vecteur-colonne de vitesse $\dot{\underline{q}}$, selon le principe de dérivée en chaînes :

$$\dot{J} = \frac{\partial J}{\partial \underline{q}} \dot{\underline{q}} \quad (4.87)$$

Toutefois ici, on dérive une matrice $m \times n$ par un vecteur-colonne $n \times 1$, le résultat est donc un tenseur (ses composantes si on veux être rigoureux) d'ordre 3, i.e. une matrice 3D $m \times n \times n$:

$$\left[\begin{array}{c} J \\ \end{array} \right]_{m \times n} = \underbrace{\left[\begin{array}{c} \frac{\partial J}{\partial \underline{q}} \\ \end{array} \right]_{m \times n \times n}}_{\text{Tenseur d'ordre 3}} \left[\begin{array}{c} \dot{\underline{q}} \\ \end{array} \right]_{n \times 1} \quad (4.88)$$

Lorsqu'on travail avec des tenseurs d'ordre 3 et plus, il est normalement plus facile de travailler avec des équations qui relient les composantes avec une notation indicelle. La composante ijk du tenseur qui résultent de la dérivation du Jacobien J par \underline{q} est égale à la dérivée de l'élément ij du Jacobien par rapport à l'élément k du vecteur-colonne \underline{q} :

$$\left[\frac{\partial J}{\partial \underline{q}} \right]_{ijk} = \frac{\partial J_{ij}}{\partial q_k} \quad (4.89)$$

Donc le produit intérieur exprimé à l'équation (4.88) peut être exprimé directement comme une sommation à effectuer pour calculer l'élément ij de la matrice J :

$$J_{ij} = \sum_k \left(\frac{\partial J_{ij}}{\partial q_k} \dot{q}_k \right) \quad (4.90)$$

Finalement, il est possible d'exprimer directement l'équation (4.86) en termes de composantes, d'indices et de sommes :

$$\ddot{r}_i = \sum_j J_{ij} \ddot{q}_j + \sum_j \sum_k \frac{\partial J_{ij}}{\partial q_k} \dot{q}_k \dot{q}_j \quad (4.91)$$

Plus de détails sur les équivalences entre les opérations en termes de matrices/vecteur-colonnes/tenseurs et les opérations en termes de composantes et d'indices sont disponibles à la section 17.4.

4.9 Résumé du chapitre

Vecteur vitesse et dérivée d'un vecteur position (base b non-inertie)

$$\underbrace{\vec{v}_B = {}^b \frac{d}{dt} \vec{r}_{B/A} + \vec{w}_{b/A} \times \vec{r}_{B/A}}_{\text{Relation vectorielle}} \Leftrightarrow \underbrace{\underline{v}_B^a = {}^a R^b \left[\underline{r}_{B/A}^b + (\underline{w}_{b/A}^b)^\times \underline{r}_{B/A}^b \right]}_{\text{Relation matricielle des composantes}} \quad (4.92)$$

Vecteur accélération et dérivée seconde d'un vecteur position

$$\vec{a}_B = {}^b \frac{d^2}{dt^2} \vec{r}_{B/A} + {}^b \frac{d}{dt} \vec{w}_{b/A} \times \vec{r}_{B/A} + \underbrace{2\vec{w}_{b/A} \times {}^b \frac{d^2}{dt^2} \vec{r}_{B/A}}_{\text{Coriolis}} + \underbrace{\vec{w}_{b/A} \times \vec{w}_{b/A} \times \vec{r}_{B/A}}_{\text{Centrifuge}} \quad (4.93)$$

$$\underline{a}_B^a = {}^a R^b \left[\underline{r}_{B/A}^b + (\underline{w}_{b/A}^b)^\times \underline{r}_{B/A}^b + \underbrace{2(\underline{w}_{b/A}^b)^\times \underline{r}_{B/A}^b}_{\text{Coriolis}} + \underbrace{(\underline{w}_{b/A}^b)^\times (\underline{w}_{b/A}^b)^\times \underline{r}_{B/A}^b}_{\text{Centrifuge}} \right] \quad (4.94)$$

Dérivée temporelle d'une matrice de rotation :

$$\dot{R} = \frac{d}{dt} R = (\underline{w})^x R \quad (4.95)$$

$${}^a \dot{R}^b = (\underline{w}_{b/A}^a)^\times {}^a R^b = {}^a R^b (\underline{w}_{b/A}^b)^\times \quad (4.96)$$

Relations différentielles entre l'espace des joints et de la tâche :

$$\underline{r} = f(\underline{q}) \quad (4.97)$$

$$\dot{\underline{r}} = J(\underline{q}) \dot{\underline{q}} \quad (4.98)$$

$$\ddot{\underline{r}} = J(\underline{q}) \ddot{\underline{q}} + J(\underline{q}, \dot{\underline{q}}) \dot{\underline{q}} \quad (4.99)$$

La matrice Jacobienne :

$$J(\underline{q}) = \frac{\partial \underline{r}}{\partial \underline{q}} = \begin{bmatrix} \frac{\partial r_1}{\partial q_1} & \dots & \frac{\partial r_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial r_m}{\partial q_1} & \dots & \frac{\partial r_m}{\partial q_n} \end{bmatrix}_{m \times n} \quad (4.100)$$

Chapitre 5

Statique des robots manipulateurs

5.1 Introduction

Ce chapitre présente des méthodes pour déterminer les relations entre les diverses forces agissant sur un robot manipulateur lorsque le système est à l'équilibre. La matrice jacobienne J , la même matrice de la relation de cinématique différentielle de l'équation 4.29, permet de transformer une force cartésienne \underline{f}_E , que l'effecteur du robot applique sur l'environnement, en couples (ou force pour un joint prismatique) équivalents $\underline{\tau}$ dans le domaine des joints :

$$\underline{\tau} = J(\underline{q})^T \underline{f}_E \quad (5.1)$$

Cette relation permet de calculer les liens entre les forces externes et les forces produites par les actionneurs sans avoir à déterminer toutes les forces internes du système, ce qui est particulièrement utile pour les robots manipulateurs et sera exploité par les méthodes de ce chapitre.

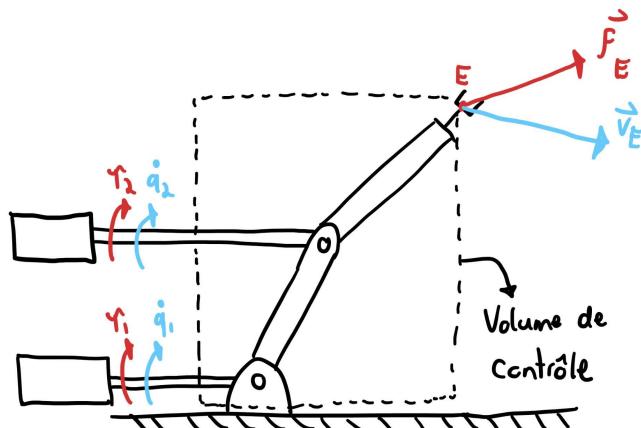


FIGURE 5.1 – Bilan de puissance mécanique pour un robot manipulateur



Capsule vidéo
Statique des robots manipulateurs
<https://youtu.be/kIAfYgPYiOk>

5.2 Les vecteurs forces

Une force est une quantité vectorielle avec une amplitude et une direction. Comme pour les notions de cinématique, nous utiliserons une notation qui distingue le vecteur-géométrique \vec{f} et le vecteur-colonne de composantes \underline{f}^a dans une base a . De plus, une force appliquée sur un corps C à un point A sera notée :

$$\vec{f}_{CA} \quad (5.2)$$

Le principe d'action/réaction indique que si une force est appliquée sur un corps $C1$ il y a nécessairement une réaction égale et opposée sur un corps $C2$. Une force agissant sur un corps $C1$ par un corps $C2$ sera notée :

$$\vec{f}_{C1/C2} \quad (5.3)$$

et le principe d'action/réaction nous donne donc la propriété suivante :

$$\vec{f}_{C1/C2} = -\vec{f}_{C2/C1} \quad (5.4)$$

Cette notation sera utilisée lorsqu'il est question de forces d'interactions entre deux corps car le signe source de confusion lorsqu'on ne distingue pas la force appliquée *sur* à la force appliquée *par* un objet.

Dans ce chapitre, nous définirons un vecteur \vec{f}_{RE} comme une force appliquée sur le robot à un point E (typiquement l'effecteur) par l'environnement, et \vec{f}_E comme son inverse, i.e. la force que le robot applique sur l'environnement :

$$\vec{f}_{RE} = \vec{f}_{Robot/Environment} \quad (5.5)$$

$$\vec{f}_E = \vec{f}_{Environment/Robot} = -\vec{f}_{RE} \quad (5.6)$$

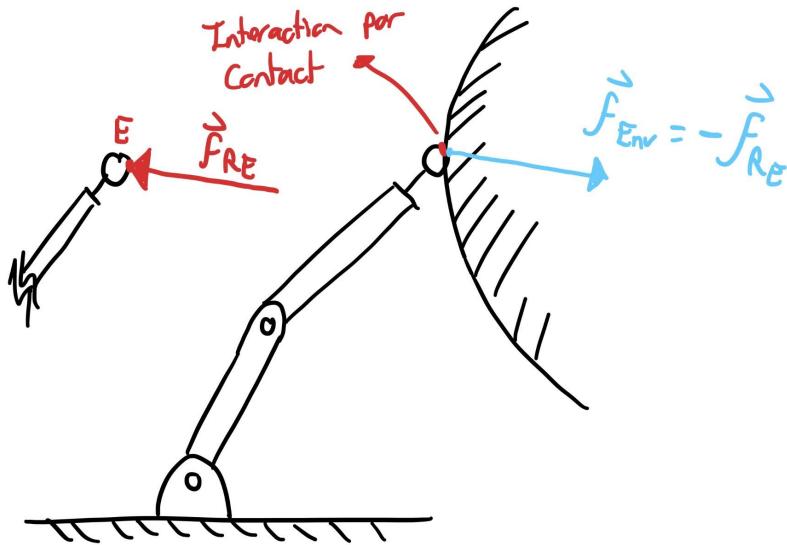


FIGURE 5.2 – Action-réaction pour une force de contact

5.3 Relation entre les forces aux joints et les forces à l'effecteur

La relation entre les forces/couples dans l'espace des joints $\underline{\tau}$ et un vecteur de forces externes \vec{f}_{RE} appliquée sur l'effecteur d'un robot peut être décrite par la relation suivante :

$$\underline{\tau} = J(\underline{q})^T \underline{f}_E = -J(\underline{q})^T \underline{f}_{RE} \quad (5.7)$$

où la matrice Jacobienne J est la même matrice de la relation de cinématique différentielle de l'équation 4.29. Comme illustré à la Figure 5.1, les variables de forces/couples doivent correspondre aux mêmes degrés de liberté que les variables vitesses associées à la relation de cinématique différentielle.

Preuve 5.3.1 La relation statique qui implique le Jacobien (eq. (5.7)) peut être déterminée à partir d'un bilan de puissance. Si on applique la 1ère loi de la thermodynamique au volume de contrôle indiqué à la Figure 5.1 pour calculer le bilan d'énergie :

$$\frac{dE}{dt} = P_{in} - P_{out} \quad (5.8)$$

Les forces internes inertielles et dissipatrices sont ici négligées pour trouver une relation valide dans des conditions quasi-statiques. De plus considérons d'abord un cas sans force conservatrice (voir section 5.4 pour ce cas). Dans ces conditions, il n'y a aucune accumulation d'énergie interne : les seules entrées/sorties de puissance dans le système sont le travail mécanique fait par les actionneurs sur le robot, et le travail mécanique fait par le robot sur l'environnement. L'égalité du travail mécanique en entrée et sortie peut être convertie en relation matricielle :

$$P_{in} = P_{out} \quad (5.9)$$

$$\dot{q}_1\tau_1 + \dot{q}_2\tau_2 + \dots = \vec{v} \cdot \vec{f}_E \quad (5.10)$$

$$\begin{bmatrix} \dot{q}_1 & \dot{q}_2 & \dots \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \end{bmatrix} = \underline{\dot{r}}^T \underline{f}_E \quad (5.11)$$

$$\underline{\dot{q}}^T \underline{\tau} = \underline{\dot{r}}^T \underline{f}_E \quad (5.12)$$

$$(5.13)$$

Ensuite, si on substitut les composantes du vecteur vitesse de l'effecteur $\underline{\dot{r}}$ par l'équation de cinématique différentielle directe (eq. (4.29)), on obtient :

$$\underline{\dot{q}}^T \underline{\tau} = (J(\underline{q}) \underline{\dot{q}})^T \underline{f}_E \quad (5.14)$$

$$\underline{\dot{q}}^T \underline{\tau} = \underline{\dot{q}}^T J(\underline{q})^T \underline{f}_E \quad (5.15)$$

$$\underline{\dot{q}}^T (\underline{\tau}) = \underline{\dot{q}}^T (J(\underline{q})^T \underline{f}_E) \Rightarrow \underline{\tau} = J(\underline{q})^T \underline{f}_E \quad (5.16)$$

Donc les deux termes qui multiplient (produit intérieur) le vecteur colonne $\underline{\dot{q}}$ doivent nécessairement être égaux, ce qui nous donne l'équation qui relie les forces dans l'espace des joints aux forces équivalentes dans l'espace de la tâche.

5.3.1 Flux de puissance

Du point de vue du flux de puissance dans le système robotique, de l'énergie électrique vers le travail mécanique fait par le robot, la Jacobienne peut-être vue comme une matrice de ratios de transformation. Par exemple, les Figures 5.3 et 5.4 illustrent quelques unes des transformations de puissance dans un système robotique et les paramètres associées. Dans chacun des domaines la puissance est caractérisée par deux variables, une de flux (courant, vitesse, débit, etc.) et une de d'effort (tension, couple, force, pression, etc.). La puissance est le produit de ces deux variables, ou le produit intérieur des vecteurs-colonnes de ces variables dans le cas de systèmes multi-variables. Le flux de puissance peut être transformé par des dispositifs qu'on appelle des transformateurs. Un transformateur amplifie la variable de flux et réduit la variable d'effort, ou vice-versa. Par exemple, les transformateurs électriques qui réduisent la tension dans un réseau électrique, les transmissions mécaniques avec ratios de réduction, etc. Lorsqu'un dispositif transfert la puissance sous une forme d'énergie différente, on les appelle des transducteurs, par exemple un moteur électrique. La puissance en entrée et en sortie des transformateurs est conservée, si l'effort est amplifié d'un facteur T le flux est réduit d'un facteur T .

Capsule vidéo
Flux de puissance pour un robot manipulateur
<https://youtu.be/qyrFzkjPhHY>

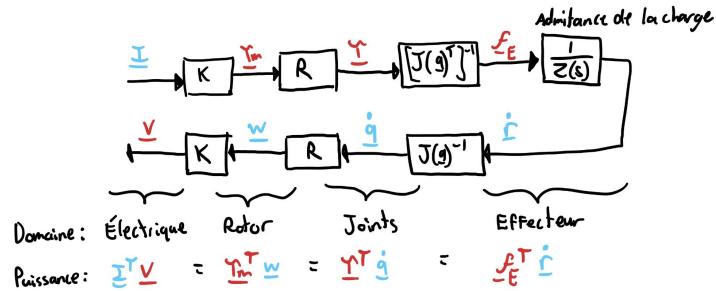


FIGURE 5.3 – Flux de puissance : les variables de flux et d'effort dans les différents domaines et espaces du robot.

Les Figures 5.3 et 5.4 illustrent les trois transformations de puissances principales dans un bras robotique typique. La puissance électrique est transformée en puissance mécanique des l'arbre des moteurs par les moteurs électriques, ensuite celle-ci est transformée en puissance mécanique dans les joints du robot par les transmissions, et finalement la puissance mécanique dans les joints est transformé en puissance mécanique à l'effecteur du robot. La première transformation est caractérisée par les constantes moteurs k_m :

$$\tau = k_m I \quad (5.17)$$

$$v = k_m w \quad (5.18)$$

la deuxième transformation est caractérisée par les ratios de réductions des transmissions R :

$$\tau = R \tau_m \quad (5.19)$$

$$w = R \dot{q} \quad (5.20)$$

et la dernière transformation est caractérisée par la matrice Jacobienne du robot J :

$$\underline{\tau} = J^T \underline{f}_E \quad (5.21)$$

$$\dot{\underline{r}} = J \dot{\underline{q}} \quad (5.22)$$

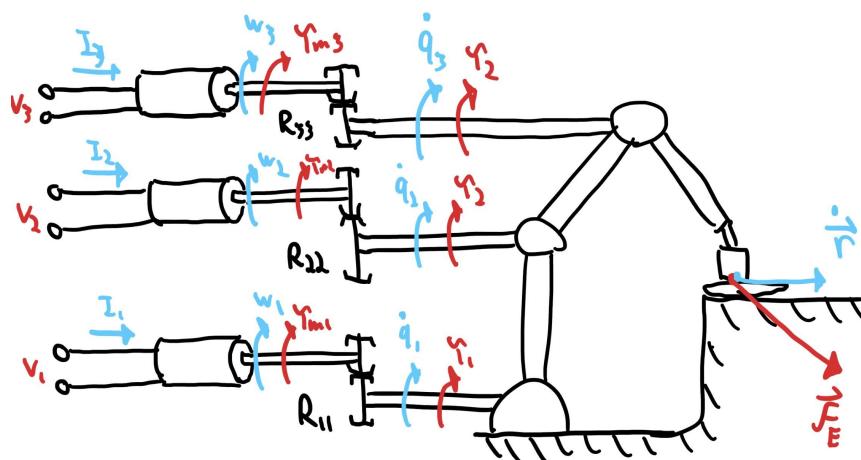


FIGURE 5.4 – Flux de puissance : de l'alimentation électrique vers le travail sur la charge

Exemple 1. Relation statique pour un robot simple à un DDL

La Figure 5.5 illustre la matrice Jacobienne d'un robot simple à 1 DDL, pour lequel l'espace de la tâche est simplement la position horizontale x de l'effecteur. Pour ce système à une entrée et une sortie, la matrice Jacobienne est de dimension 1×1 , donc un scalaire qui ici correspond à la hauteur de l'effecteur du robot. La vitesse \dot{x} pour une vitesse angulaire $\dot{\theta}$ donnée est fonction de cette hauteur. Finalement, le lien entre une force horizontale à l'effecteur et le couple équivalent au joint est aussi fonction de cette même hauteur.

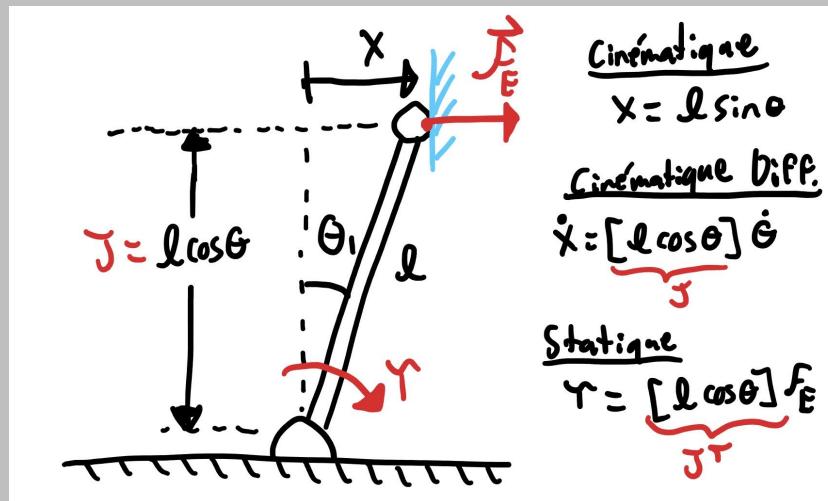


FIGURE 5.5 – Exemple de la relation statique pour un manipulateur à 1DDL

5.3.2 Relation statique sur une singularité

Lorsqu'un robot manipulateur est sur une singularité, la transposée de la matrice Jacobienne est caractérisée par une amplification d'un facteur zéro pour certaines directions spécifiques de vecteur force externe. Cela signifie que aucun couple dans l'espace des joints est nécessaire pour résister à une force externe dans cette direction. De façon générale si un robot est proche d'une singularité, très peu de couple aux joints est nécessaire pour résister à certaines directions de forces externes. La marche humaine est particulièrement efficace car nos jambes sont maintenues dans des configurations proches de singularité qui permettent de résister à la charge gravitationnelle avec peu d'effort musculaire.

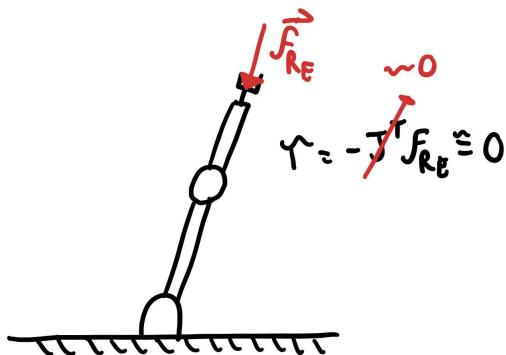


FIGURE 5.6 – Relation statique sur une singularité

5.4 Relation statique incluant les forces conservatrices

Si la bilan d'énergie effectué à la section 5.3 est fait en incluant l'énergie potentielle, un terme de forces conservatrices \underline{g} doit être ajouté à l'équation de la balance des forces dans le systèmes. On nommera ici ce vecteur \underline{g} car pour les robots manipulateur, c'est généralement un vecteur de forces gravitationnelles.

$$\underline{\tau} = J(\underline{q})^T \underline{f}_E + \underline{g}(\underline{q}) \quad \text{avec :} \quad \underline{g}(\underline{q}) = \underbrace{\frac{\partial V(\underline{q})}{\partial \underline{q}^T}}_{\text{Gradient de l'énergie potentielle}} \quad (5.23)$$

Note : Le vecteur colonne par lequel on prend les dérivés partielles de V est ici noté avec une transposition pour respecter la convention (voir section 17.4) pour obtenir pour \underline{g} un vecteur-colonne $n \times 1$ plutôt que un vecteur-rangé $1 \times n$.

Preuve 5.4.1 *Le nouveau terme provient du fait que la variation de l'énergie interne du système n'est pas nulle, car l'énergie potentielle du système varie selon le travail effectué par les forces conservatrices :*

$$E = V(\underline{q}) \Rightarrow \frac{dE}{dt} = \frac{dV}{dt} = \frac{\partial V}{\partial \underline{q}} \frac{d\underline{q}}{dt} = \frac{\partial V}{\partial \underline{q}} \dot{\underline{q}} \quad (5.24)$$

On obtient alors :

$$\frac{dE}{dt} = P_{in} - P_{out} \quad (5.25)$$

$$\frac{\partial V}{\partial \underline{q}} \dot{\underline{q}} = \dot{\underline{q}}^T \underline{\tau} - \dot{\underline{q}}^T J(\underline{q})^T \underline{f}_E \quad (5.26)$$

$$\dot{\underline{q}}^T \left(\frac{\partial V}{\partial \underline{q}^T} \right) = \dot{\underline{q}}^T \left(\underline{\tau} - J(\underline{q})^T \underline{f}_E \right) \quad (5.27)$$

Donc pour que la relation de puissance soit respecté, les termes qui multiplie le vecteur-colonne de vitesses doivent être égaux, ce qui donne la relation entre les forces de l'équation (5.23).



Capsule vidéo
Relation statique d'un robot manipulateur incluant la gravité
<https://youtu.be/RcARRDVzrm8>

5.5 Relation de la compliance aux joints et à l'effecteur

Lorsque les robots sont contrôlé en position par des servo-moteurs à chaque joint, chacun de ces moteurs et sa boucle d'asservissement en position peut être représenté comme une rigidité dans l'espace des joints, i.e. une variation du couple produit par le joint pour un déplacement de ce joint par rapport à la configuration que le contrôleur tente de maintenir :

$$\delta\tau_i = k_i \delta q_i \Rightarrow \delta\tau = K_q \delta q \quad (5.28)$$

où K_q est une matrice de rigidité dans l'espace des joints. Il est ensuite intéressant de déterminer la rigidité d'un robot, comme illustré à la Figure 5.7, lorsqu'il subit une force externe en fonction de la rigidité angulaire de ses servo-moteurs. La variation de position de l'effecteur peut être reliée aux déplacement angulaire dans

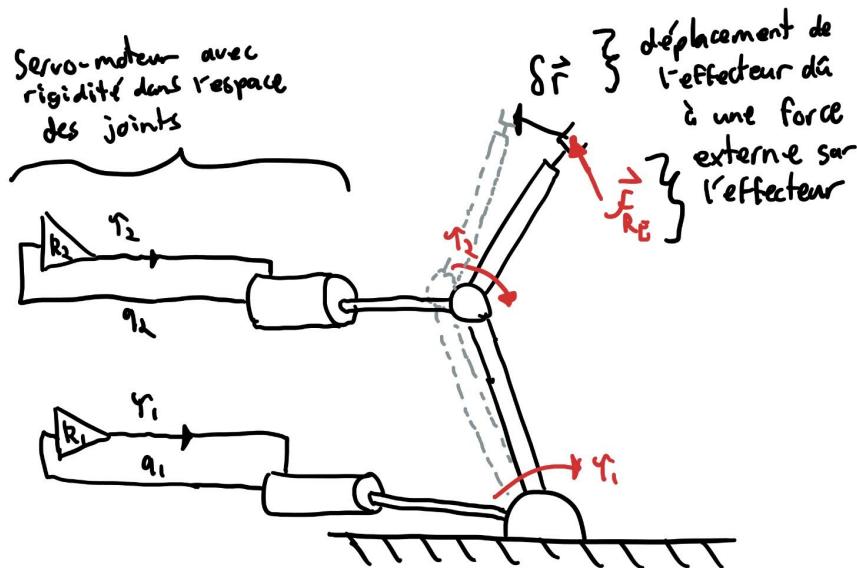


FIGURE 5.7 – Compliance d'un robot à une force externe

l'espace des joints, et la force externe peut être transformée en couples équivalents à chacun des joints (pour un équilibre statique). En manipulant les équations on obtient :

$$\delta r = J \delta q = JK_q^{-1} \tau = \underbrace{[JK_q^{-1} J^T]}_{C_r} f_{R_E} \quad (5.29)$$

où C_r est une matrice de compliance à l'effecteur. La rigidité apparente à l'effecteur, due à un rigidité dans l'espace des joints, est donc donnée par une relation qui relie la variation de la position de l'effecteur à une force externe appliquée sur l'effecteur :

$$f_E = -f_{R_E} = -K_r \delta r \quad \text{avec } K_r = C_r^{-1} = [JK_q^{-1} J^T]^{-1} \quad (5.30)$$

Comme illustré à la Figure 5.8, la matrice de rigidité dans l'espace des joints est généralement diagonale (les moteurs sont indépendants). Il est à noter que la matrice de rigidité à l'effecteur est dépendante de la configuration nominale du robot (i.e. J est une fonction de q en général). Finalement, il est à noter que la matrice C de compliance est singulière lorsque le Jacobien est singulier, donc sur une singularité un robot manipulateur a une ou des directions avec aucune compliance, i.e. infiniment rigide, comme illustré à la Figure 5.9.

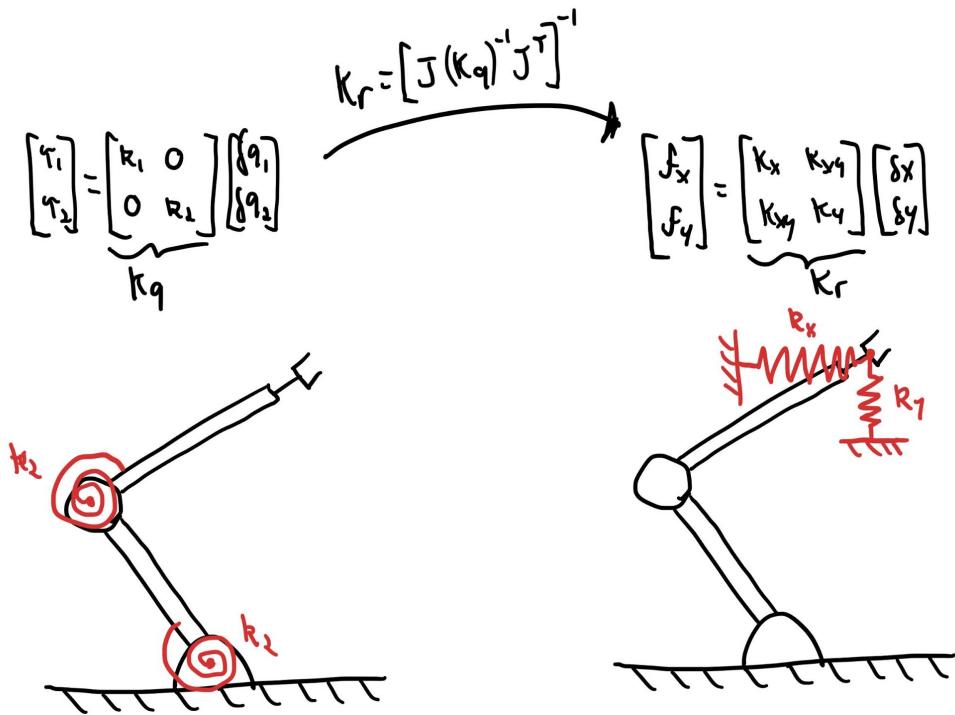


FIGURE 5.8 – Matrice de rigidité équivalente d'un robot dans l'espace de l'effecteur

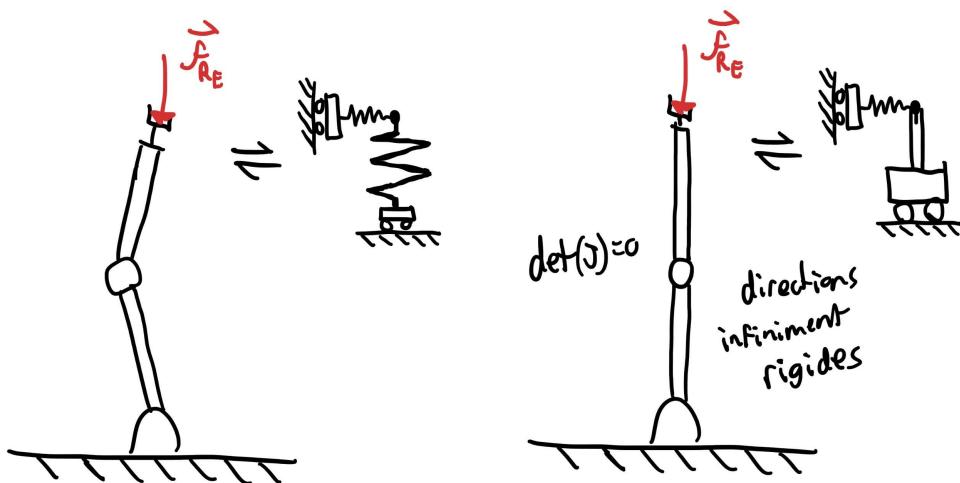


FIGURE 5.9 – Effet des configurations singulières sur la rigidité d'un robot manipulateur



Capsule vidéo

Compliance et rigidité des robots manipulateurs

<https://youtu.be/U3fhFlkVoEg>

5.6 Manipulabilité en force d'un robot manipulateur

L'ellipsoïde de force d'un robot manipulateur est l'enveloppe des vecteurs forces possibles à l'effecteur pour des vecteurs d'entrée τ unitaires (typiquement des couples moteurs), voir ???. L'enveloppe de forces possibles à l'effecteur, résultant d'un vecteur de couples des joints unitaire, peut être calculer en débutant avec une équation qui contraint la norme du vecteur de couple des joint et en substituant $\underline{\tau}$ par $J^T \underline{f}_E$ dans l'équation :

$$1 = \underline{\tau}^T \underline{\tau} \quad (5.31)$$

$$1 = (J^T \underline{f}_E)^T J^T \underline{f}_E \quad (5.32)$$

$$1 = \underline{f}_E^T (JJ^T) \underline{f}_E \quad (5.33)$$

$$1 = \underline{f}_E^T A \underline{f}_E \quad \text{avec} \quad A = JJ^T \quad (5.34)$$

On voit donc que c'est la même matrice JJ^T qui caractérise l'ellipsoïde de manipulabilité en force que celle qui caractérise l'ellipsoïde de vitesse (voir section 4.7, mais ici c'est l'inverse de la matrice qui apparaît). Les axes principaux sont donc les mêmes mais les amplitudes sont inversés par rapport à l'ellipsoïde. Par exemple, si le vecteur de vitesse possible à l'effecteur est faible dans une certaine direction, les forces externes possible à soutenir dans cette même direction vont être grandes. De plus, sur une singularité, il y a des directions pour lesquelles le vecteur vitesse possible est nul, ce qui va correspondre à des direction pour lesquelles le vecteur force possible va tendre vers l'infini, ce qui correspond à une direction pour laquelle une force externe va être reprise par la structure sans effort requis par les actionneurs.

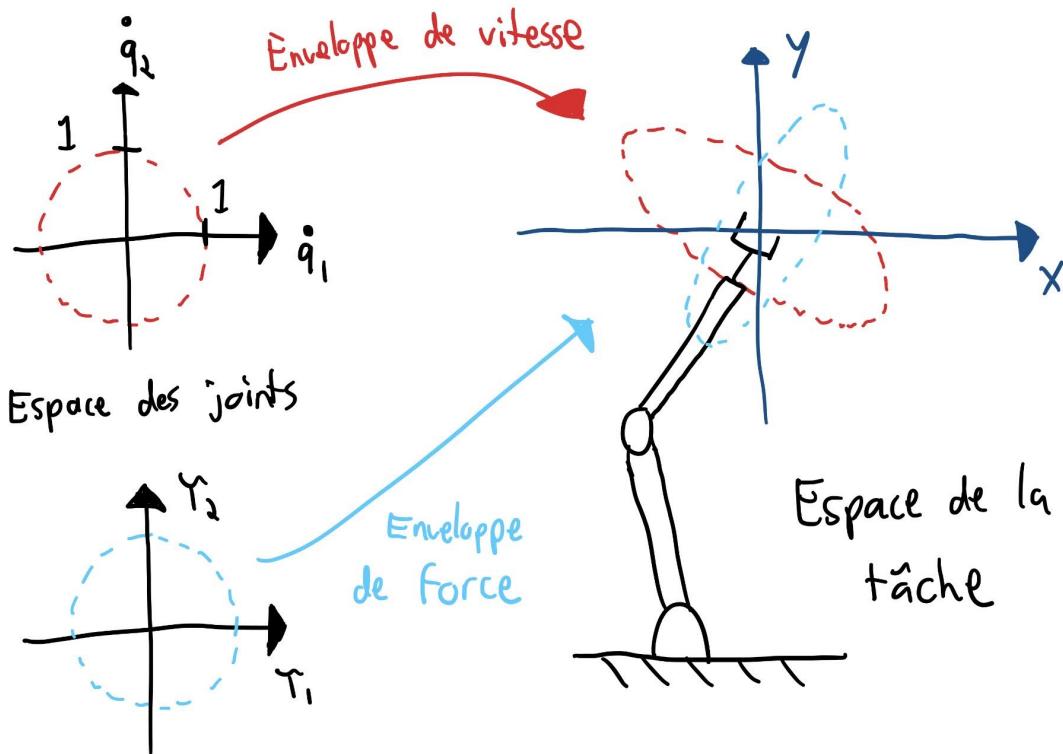


FIGURE 5.10 – Ellipsoïde de manipulabilité pour la force et la vitesse

Exemple 2. Exemples d'ellipsoïdes de manipulabilité pour un robot 2DDL

La Figure 5.11 illustre un robot 2DDL planaire dans quelques configurations. Lorsque le robot est

dans une configuration avec le coude partiellement plié les capacités du robot sont le plus isotropique. Lorsque le coude est déplié le robot est capable de supporter des grande forces dans la direction radiale et l'inverse lorsque le coude est complètement plié.

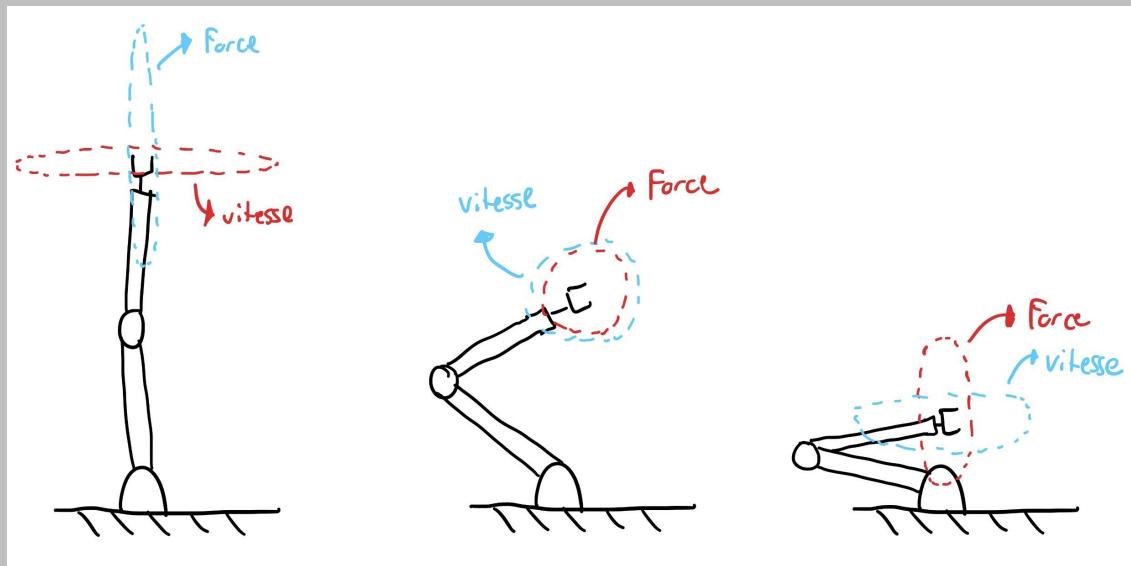


FIGURE 5.11 – Exemples d'ellipsoïde de manipulabilité pour divers configuration

Notre expérience humaine peut nous aider à avoir une intuition à savoir comment des propriétés peuvent être exploitées. Comme exercice de pensée je vous suggère de réfléchir à divers tâches du quotidien, la configuration de nos bras et jambe naturellement utilisé et les requis en force/vitesse : Comment configurerez vous votre coude pour pousser contre une lourde porte ? Quelle est la configuration naturelle de votre bras pour écrire ? Comment configurerez vous votre bras pour lancer une balle le plus loin possible ? Est-ce qu'il serait intéressant de marcher les genoux pliés ?

5.7 Résumé du chapitre

Force externes à l'effecteur :

$$\underline{f}_{R_E} = -\underline{f}_E \quad (5.35)$$

\underline{f}_{R_E} = Force appliquée sur l'effecteur du robot par l'environnement

\underline{f}_E = Force appliquée sur l'environnement par le robot

Relation force joints-effecteur :

$$\underline{\tau} = J(\underline{q})^T \underline{f}_E \quad (5.38)$$

Équilibre statique :

$$\underline{\tau} = J(\underline{q})^T \underline{f}_E + g(\underline{q}) \quad \text{avec : } g(\underline{q}) = \frac{\partial V(\underline{q})}{\partial \underline{q}} \quad (5.39)$$

Rigidité apparente à l'effecteur :

$$K_r = C_r^{-1} = [J K_q^{-1} J^T]^{-1} \quad (5.40)$$

Chapitre 6

Dynamique des robots manipulateurs

6.1 Introduction

Ce chapitre traite de la relation entre la l'évolution temporelle de la position d'un robot et les forces appliquées sur celui-ci. Notre outil mathématique principal pour décrire cette relation est un système d'équations différentielles qui relie des variables d'accélération du robot aux autres variables qui influence les forces internes et externes impliqués. Obtenir cette relation est utile lorsqu'on désire prédire des capacités dynamiques d'un système, simuler numériquement l'évolution temporelle d'un système dans des conditions connues ou bien synthétiser des lois de commandes.



Capsule vidéo
Dynamique des robots manipulateurs
<https://youtu.be/6z3grrVNbj4>

6.2 Structure des équations

La forme la plus générique (pour un système continu) des équations différentielles représentant l'évolution d'un système dynamique dans le temps est les équations d'états :

$$\dot{\underline{x}} = f(\underline{x}, \underline{u}) \quad (6.1)$$

où \underline{x} est un vecteur d'état, i.e. la mémoire ou les niveaux d'énergie du système, et \underline{u} est un vecteur des entrées du système. C'est la représentation qui est utilisée par les simulateurs numériques pour calculer des trajectoires, pour générer des diagrammes de phases, et par plusieurs méthodes numérique pour générer des lois de commandes ou des trajectoires.

Lorsque le système dynamique à représenter est un système purement mécanique où les entrée sont des forces, comme c'est souvent le cas en robotique, il est possible d'utiliser une représentation plus spécifique avec une structure d'ordre 2 qui décrit la relation reliant les dérivées secondes des coordonnées (les accélérations) aux autres forces impliquées :

$$\underbrace{H(\underline{q})\ddot{\underline{q}} + C(\underline{q}, \dot{\underline{q}})\dot{\underline{q}}}_{m\ddot{\underline{a}}} = \underbrace{\underline{f}}_{\vec{f}} \quad (6.2)$$

où \underline{q} est un vecteur de coordonnées généralisée (des positions linéaires ou angulaires qui caractérise la configuration du système mécanique), H est une matrice d'inertie, C est la matrice de coriolis et \underline{f} un vecteur des forces impliquée dans le système.

Il est souvent utile de décomposer \underline{f} ainsi pour représenter spécifiquement les types de forces impliquées dans le système :

$$\underbrace{H(\underline{q})\ddot{\underline{q}} + C(\underline{q}, \dot{\underline{q}})\dot{\underline{q}}}_{\substack{\text{effets inertIELS} \\ m\ddot{\underline{a}}}} + \underbrace{d(\dot{\underline{q}}, \underline{q})}_{\text{forces dissipatrices}} + \underbrace{g(\underline{q})}_{\text{forces conservatrices}} = \underbrace{B(\underline{q})\underline{u}}_{\substack{\text{effet des actionneurs}}} + \underbrace{J^T(\underline{q})\underline{f}_{R_E}}_{\substack{\text{force externes sur l'effecteur}}} \quad (6.3)$$

Cette équation est parfois appelé l'équation des manipulateurs et cette représentation est très utile pour analyser le comportement de système robotique et synthétiser des lois de commandes.

6.3 Variables, nomenclature et dimensions

Le tableau ci-dessous présente les variables qui sont utilisée dans ce chapitre.

TABLE 6.1 – Nomenclature pour la dynamique des manipulateurs

		Dimensions
n	: nombre de DDL et de coordonnées généralisées	
m	: nombre d'actionneur	
c	: nombre de contraintes de contact	
o	: nombre de coordonnée de l'espace de la tâche	
		Scalaires
T	: Énergie cinétique	
V	: Énergie potentielle	
		Vecteurs
\underline{u}	: Forces/couples des actionneurs	m
$\underline{\tau}$: Forces des actionneurs transformées en coordonnées généralisées	n
\underline{q}	: Coordonnées généralisée / espace des joints	n
\underline{d}	: Somme de force dissipatrices dans l'espace des joints	n
\underline{g}	: Somme des force conservatrices dans l'espace des joints	n
\underline{h}	: Sommes des forces internes dans l'espace des joints	n
ϕ_C	: Fonction de contraintes pour un point de contact C	c
\underline{f}_{R_C}	: Forces cartésienne de contact appliquée au point C	c
\underline{f}_{R_E}	: Forces cartésienne externe appliquée sur l'effecteur du robot	o
\underline{r}	: Position de l'effecteur / coordonnées de l'espace de la tâche	o
		Matrices
H	: Matrice d'inertie	$n \times n$
C	: Matrice de coriolis	$n \times n$
B	: Matrice des actionneurs	$n \times m$
J	: Matrice jacobienne de l'effecteur / espace de la tâche	$o \times n$
J_C	: Matrice jacobienne pour les coordonnées contraintes du point de contact C	$c \times n$
J_i	: Matrice jacobienne pour un point arbitraire i sur le robot	$3 \times n$

6.4 Équation des manipulateurs

L'équation des manipulateur caractérise la dynamique d'un système robotique décrite dans l'espace des coordonnées des joints :

$$H(\underline{q})\ddot{\underline{q}} + C(\underline{q}, \dot{\underline{q}})\dot{\underline{q}} + d(\underline{q}, \dot{\underline{q}}) + \underline{g}(\underline{q}) = B(\underline{q})\underline{u} + J^T(\underline{q})\underline{f}_{R_E} \quad (6.4)$$

où \underline{q} est le vecteur des forces généralisée, H est la matrice d'inertie, C est la matrice des forces de Coriolis/centrifuge, d est un vecteur de force dissipatrices, \underline{g} un vecteur de forces conservatrices (typiquement la gravité), B est une matrice qui transforme le vecteur des forces des actionneurs \underline{u} (les entrées du système) en forces généralisée $\underline{\tau}$, et J^T la transposée de la matrice jacobienne associée au point d'application E d'une force externe \underline{f}_{R_E} .



Exercice de code

Exemple interactif pour l'équation des manipulateurs

https://colab.research.google.com/drive/1sGmE681UHv6Y_xy5GMA_k9Qda7SPH4kD?usp=sharing

6.4.1 Forces externes multiples

L'équation (6.4) est pour une situation où une seule force externe est appliquée sur le robot, sur l'effectuer i.e. sur le point du robot dont la position est décrit par l'équation de la cinématique directe $\underline{r} = f(\underline{q})$. Si plusieurs forces externes sont appliquées à divers points sur le robot, il serait possible de décrire cette situation avec une sommation :

$$H(\underline{q})\ddot{\underline{q}} + C(\underline{q}, \dot{\underline{q}})\dot{\underline{q}} + d(\underline{q}, \dot{\underline{q}}) + \underline{g}(\underline{q}) = B(\underline{q})\underline{u} + \sum_i J_i^T(\underline{q})\underline{f}_{R_i} \quad (6.5)$$

où \underline{f}_{R_i} est une force externe appliquée sur le robot à un point i , et J_i est une matrice jacobienne associé au mouvement du point i :

$$J_i(\underline{q}) = \frac{\partial r_i(\underline{q})}{\partial \underline{q}} \quad (6.6)$$

où $r_i(\underline{q})$ est la position du point i exprimée comme une fonction des coordonnées \underline{q} , i.e. la fonction de cinématique directe mais pour un point particulier.

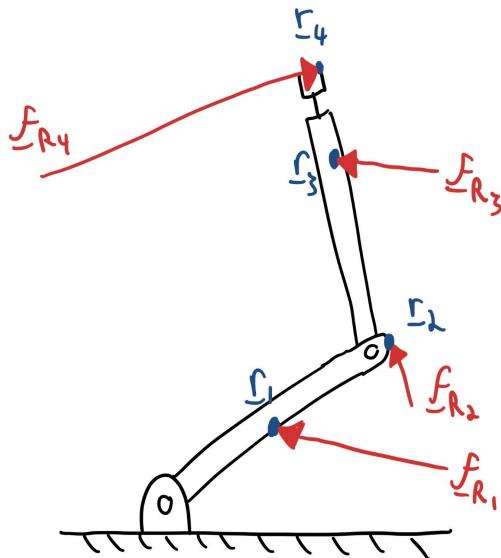


FIGURE 6.1 – Robot avec plusieurs forces externes

6.4.2 Matrice des actionneurs

La matrice $B(\underline{q})$ est une matrice de transformation qui relie le vecteur \underline{u} , représentants les entrées du système, et les forces généralisées $\underline{\tau}$ associées dans le systèmes de coordonnées \underline{q} utilisé pour décrire les équations dans l'espace de la tache :

$$\underline{\tau} = B(\underline{q})\underline{u} \quad (6.7)$$

Note : Dans le chapitre sur la statique (5), les divers équations et analyses étaient présentés en assumant qu'on nos actionneurs permettent de commander le vecteur $\underline{\tau}$ directement. Ici dans ce chapitre nous incluons une matrice de transformation B pour avoir une représentation plus englobante.

La Figure 6.2 illustre qualitativement diverses architectures d'actionneurs qui mènent à différentes structures pour B . Le cas le plus simple est lorsque les entrées τ sont des forces colocalisées avec les coordonnées généralisées q , par exemple si \underline{u} représente des couples net aux joints d'un système alors la matrice B est la matrice identité car les entrées \underline{u} sont directement des forces généralisées aux joints $\underline{\tau}$. Si on désire que les entrées \underline{u} soit directement les couples des moteurs électriques d'un robot alors la matrice B va représenter les ratios de réduction des actionneurs. Ensuite, si les actionneurs ne sont pas colocalisés avec les coordonnées généralisées, alors la matrice B fait le pont entre les forces aux actionneurs et les forces généralisées qu'ils génèrent. Finalement, si il y a moins d'actionneurs que de DDL alors la matrice B ne sera pas carrée.

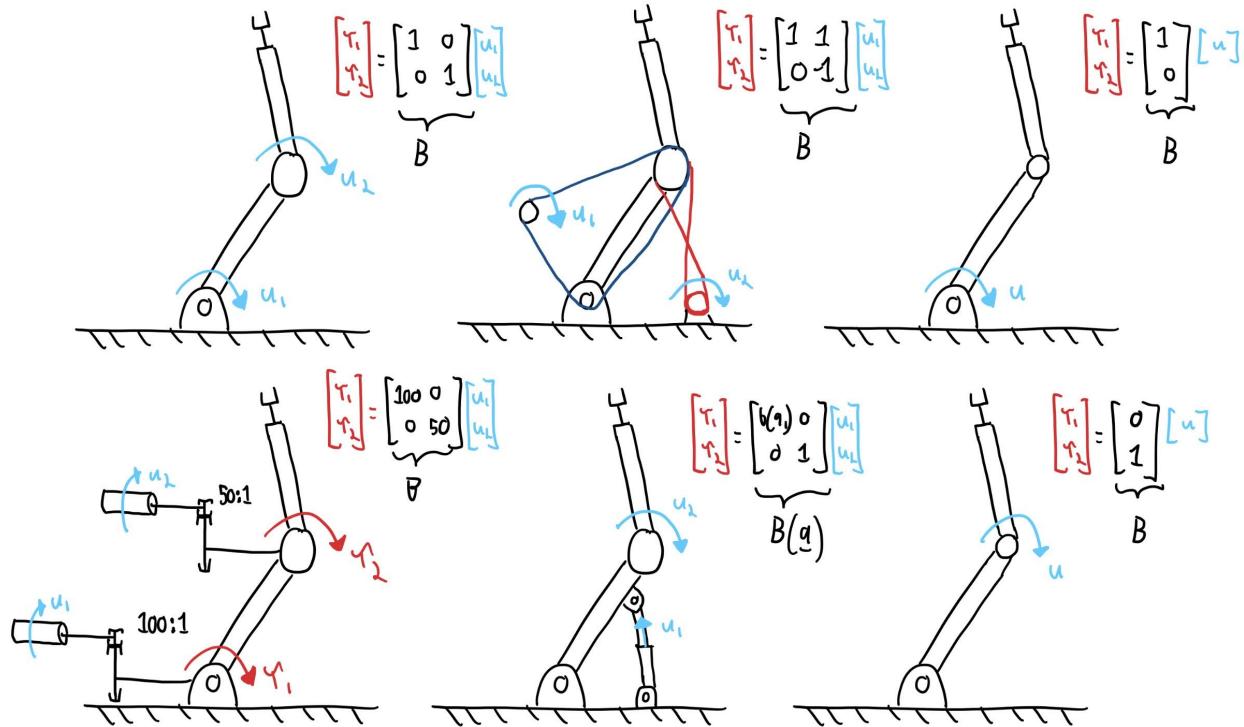


FIGURE 6.2 – Matrice B des actionneurs

La notion de puissance est utile pour définir cette matrice : le produit scalaire entre le vecteur vitesse des coordonnées généralisées \dot{q} et les forces généralisées associées aux actionneurs $B\underline{\tau}$ correspond à la puissance entrante fournie par les actionneurs :

$$P_{actionneur} = \dot{q}^T \underline{\tau} = \dot{q}^T B(\underline{q})\underline{u} \quad (6.8)$$

6.4.3 Forces conservatrices

Le vecteur de force conservatrice est défini comme le gradient de l'énergie potentiel par rapport au vecteur de coordonnées généralisée :

$$\underline{g}(\underline{q}) = \frac{\partial V(\underline{q})}{\partial \underline{q}^T} \quad (6.9)$$

de sorte que le taux de variation de l'énergie potentiel soit donnée par

$$\dot{V} = \frac{\partial V}{\partial \underline{q}} \dot{\underline{q}} = \underline{g}^T \dot{\underline{q}} \quad (6.10)$$

voir section 5.4 pour le détail.

Gravité : Dans ce chapitre ce vecteur est noté \underline{g} car typiquement la seule énergie potentielle significative est l'énergie gravitationnelle et les forces conservatrices sont les forces gravitationnelles. L'énergie potentielle gravitationnelle peut être calculé comme la sommation de l'énergie potentielle de chacun des liens rigides du robot :

$$V_g(\underline{q}) = \sum_i m_i g h_i(\underline{q}) \quad (6.11)$$

où m_i est la masse du lien i , g la constante gravitationnelle et $h_i(\underline{q})$ est la hauteur du centre de gravité (c.g.) du lien i exprimée en fonction des coordonnées généralisées, comme illustré à la Figure 6.3. Il est à noter qu'on doit généralement introduire des paramètres géométriques additionnel pour décrire la position des c.g., ici notés avec l'indice c .

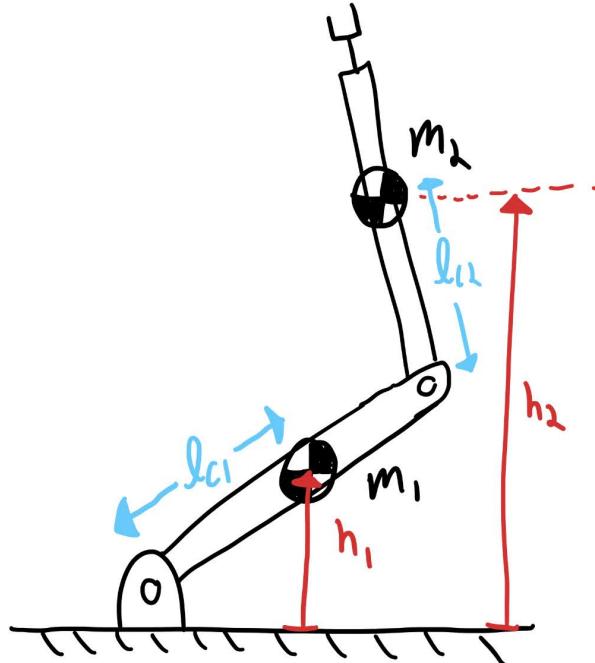


FIGURE 6.3 – La hauteur des centres de gravité des liens rigides

6.4.4 Forces dissipatrices

Les forces dissipatrices représentent les phénomènes qui transforment l'énergie du mouvement mécaniques en pertes thermiques et phénomènes irréversibles. Pour être incluse dans l'équation des manipulateurs, ces force doivent être exprimée comme des forces associées aux coordonnées généralisée de sorte que le puissance dissipée soit égale au produit scalaire entre le vecteur de ces forces et le vecteur vitesse :

$$\dot{Q}_{out} = \dot{\underline{q}}^T \underline{d} \quad (6.12)$$

Pour les robots manipulateurs, la principale source de dissipation vient des joints et des actionneurs, ce qui amène généralement à faire l'hypothèse que la force de dissipation d_i est seulement un fonction de la vitesse du joint associé \dot{q}_i :

$$d_i = f_d(\dot{q}_i) \approx b_s \operatorname{sgn}(\dot{q}_i) + b_v \dot{q}_i + \frac{1}{2} \rho C_d A \operatorname{sgn}(\dot{q}_i) \dot{q}_i^2 \quad (6.13)$$

Modéliser les phénomènes complexes de friction est une science imparfaite, c'est généralement l'aspect du modèle dynamique d'un robot qui est le moins précis et qui limite la fidélité du modèle. Selon l'objectif de modélisation il va y avoir un compromis à faire entre simplicité et fidélité. Dans un contexte de conception, analyse et/ou développement de loi de commande, on se contente normalement d'approximer la friction comme une combinaison de friction sèche, friction visceuse et friction quadratique, voir Figure 6.4, mais c'est une grosse approximation de phénomène microscopique de contact complexes. Comme illustré à la Figure 6.5, le comportement observé réel serait typiquement plus complexe et caractérisé par de l'hystérésis.

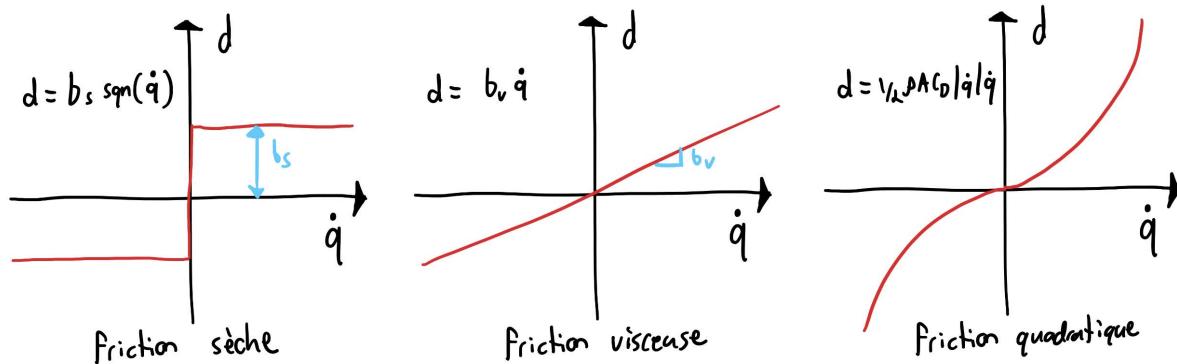


FIGURE 6.4 – Modèles simples de friction

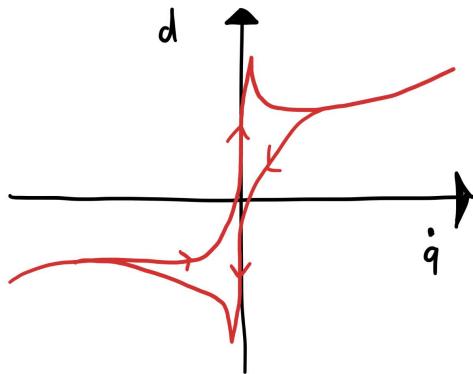


FIGURE 6.5 – Friction : comportement réel

6.4.5 Effets inertIELS

Les termes $H(\underline{q})\ddot{\underline{q}}$ et $C(\underline{q}, \dot{\underline{q}})\dot{\underline{q}}$ dans l'équation des manipulateur sont l'équivalent du terme ma dans l'équation $f = ma$ alors que tout les autres termes sont des forces. Ici le terme ma prend cette forme particulière car il est exprimé dans les coordonnées généralisée du système relié à l'espace des joints. Le terme $H(\underline{q})\ddot{\underline{q}}$ représente la "force" inertielle nécessaire pour obtenir une accélération $\ddot{\underline{q}}$ à un configuration \underline{q} , alors que le terme $C(\underline{q}, \dot{\underline{q}})$ représente une "force" inertielle nécessaire pour maintenir une certaine vitesse $\dot{\underline{q}}$ à la configuration \underline{q} , comme l'effet centrifuge ou de coriolis. Le second terme existe seulement lorsque la matrice H varie dans le temps, lorsque de momentum peut varier sans avoir une variation de vitesse dans l'espace des joints.

Energie cinétique :

L'énergie cinétique T du système est directement reliée à la matrice d'inertie H et la vitesse des joints $\dot{\underline{q}}$. L'équivalent multi-dimension de l'équation de l'énergie cinétique scalaire $\frac{1}{2}mv^2$ est donné par :

$$T(\underline{q}, \dot{\underline{q}}) = \frac{1}{2}\dot{\underline{q}}^T H(\underline{q})\dot{\underline{q}} = \frac{1}{2} \sum_i \sum_j H_{ij} \dot{q}_i \dot{q}_j \quad (6.14)$$

Propriété de la matrice H : Il est a noté que la matrice d'inertie est une matrice $n \times n$ (n est le nombre de DDL), toujours symétrique et définie positive (ce qui veux dire que l'équation ci-dessus ne peut jamais prendre de valeur négative peu importe le vecteur vitesse $\dot{\underline{q}}$). Cela implique que l'inverse H^{-1} va toujours exister, une propriété qui sera utile pour la suite !

La matrice C , souvent appelé la matrice de coriolis, n'est pas une propriété indépendante du système, elle est en fait reliée à la variation temporelle de la matrice H :

$$\dot{H} = C + C^T \quad (6.15)$$

On peut en fait aussi définir directement la matrice C connaissant l'expression de la matrice H en utilisant la relation suivante :

$$C_{ij} = \sum_k \Gamma_{ijk} \dot{q}_k \quad \text{avec} \quad \Gamma_{ijk} = \frac{1}{2} \left(\frac{\partial H_{ij}}{\partial q_k} + \frac{\partial H_{ik}}{\partial q_j} - \frac{\partial H_{jk}}{\partial q_i} \right) \quad (6.16)$$

Si on définit \underline{c} comme le vecteur résultant du terme $C(\underline{q}, \dot{\underline{q}})\dot{\underline{q}}$, on peut aussi obtenir la relation sous cette forme :

$$c_i = \sum_j \sum_k \Gamma_{ijk} \dot{q}_j \dot{q}_k \quad (6.17)$$

$$\underline{c} = C(\underline{q}, \dot{\underline{q}})\dot{\underline{q}} = \dot{\underline{q}}^T \Gamma(\underline{q})\dot{\underline{q}} \quad (6.18)$$

Preuve 6.4.1 À venir !! voir photo ipad alex 26 janvier 2023

6.4.6 Forme compacte des équations des manipulateurs

Parfois pour alléger la manipulation des équations, une version abrégée où \underline{c} est parfois utilisé pour regrouper toute les forces internes (qui sont des fonctions des états du systèmes) :

$$\underline{h} = C(\underline{q}, \dot{\underline{q}})\dot{\underline{q}} + d(\underline{q}, \dot{\underline{q}}) + \underline{g}(\underline{q}) \quad (6.19)$$

ce qui simplifie, pour un cas sans forces externes, l'équation des manipulateurs à :

$$H\ddot{\underline{q}} + \underline{h} = \underline{\tau} \quad (6.20)$$

6.4.7 Systèmes de coordonnées additionnels

TODO : Uniformiser le nom des variables pour cette section.

Parfois il peut être utile de travailler avec des systèmes de coordonnées supplémentaire à l'espace des joints. La Figure 6.6 montre quelques espaces supplémentaires qui peuvent être utilisés.

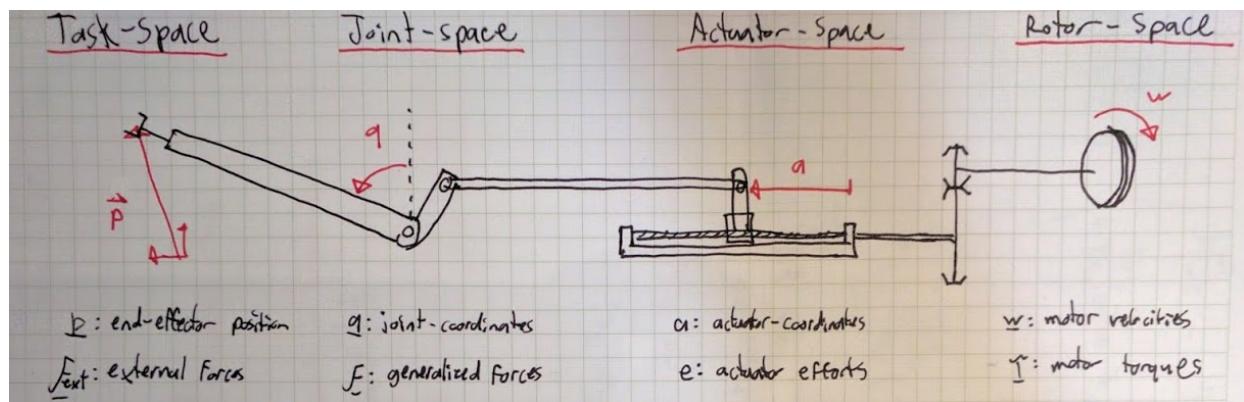


FIGURE 6.6 – Coordinate systems

Si les relations cinématiques entre les coordonnées de ces espaces sont donné par :

$$\dot{\underline{r}} = J_e(\underline{q})\dot{\underline{q}} \quad \text{de l'espace des joints vers l'espace de la tâche} \quad (6.21)$$

$$\dot{\underline{a}} = J_a(\underline{q})\dot{\underline{q}} \quad \text{de l'espace des joints vers l'espace des actionneurs} \quad (6.22)$$

$$\underline{w} = R\underline{a} \quad \text{de l'espace des actionneurs vers l'espace des moteurs} \quad (6.23)$$

On pourrait utiliser ces relations pour inclure directement des forces dans l'espace des actionneurs \underline{f}_a , et définir $\underline{\tau}$ comme les couples moteurs plutôt celui rapporté au joint ainsi :

$$H\ddot{\underline{q}} + C\dot{\underline{q}} + \underline{d} + \underline{g} = \underbrace{J_a^T(\underline{q})R^T}_{B(\underline{q})}\underline{u} + J_e^T(\underline{q})\underline{f}_e + J_a^T(\underline{q})\underline{f}_a \quad (6.24)$$

6.5 Conservation de l'énergie

Si on reprend le bilan d'énergie avec le volume de contrôle de la figure 5.1, mais cette fois ci en incluant aussi l'énergie cinétique dans l'énergie interne E et les pertes thermiques dues aux forces dissipatrices \dot{Q} , on trouve avec la première loi de la thermodynamique :

$$\frac{dE}{dt} = P_{in} - P_{out} \quad (6.25)$$

$$\dot{T} + \dot{V} = \sum \tau_i \dot{q}_i - \sum f_i \dot{r}_i - \dot{Q} \quad (6.26)$$

$$\frac{d}{dt} \left(\frac{1}{2} \underline{\dot{q}}^T H \underline{\dot{q}} \right) + \frac{\partial V}{\partial \underline{\dot{q}}} \underline{\dot{q}} = \underline{\tau}^T \underline{\dot{q}} - \underline{f}_E^T J \underline{\dot{q}} - \underline{d}^T \underline{\dot{q}} \quad (6.27)$$

$$\underline{\dot{q}}^T H \underline{\dot{q}} + \underline{\dot{q}}^T \frac{1}{2} \dot{H} \underline{\dot{q}} + \underline{\dot{q}}^T \underline{g} = \underline{\dot{q}}^T B \underline{u} + \underline{\dot{q}}^T J^T \underline{f}_{R_E} - \underline{\dot{q}}^T \underline{d} \quad (6.28)$$

On peut donc voir toute l'expression comme un produit scalaire entre le vecteur et les termes restant qui peuvent être simplifier si on substitue par notre définition de l'équation des manipulateurs. Les seules termes qui ne s'annulent pas sont \dot{H} et C , ce qui mène à une relation entre les deux :

$$\underline{\dot{q}}^T \left[H \underline{\dot{q}} + \frac{1}{2} \dot{H} \underline{\dot{q}} + \underline{g} + \underline{d} - \underline{\tau} - J^T \underline{f}_E \right] = 0 \quad (6.29)$$

$$\underline{\dot{q}}^T \left[\frac{1}{2} \dot{H} - C \right] \underline{\dot{q}} = 0 \quad (6.30)$$

Cette expression va toujours être égale à zéro si $\dot{H} = C + C^T$, une condition qui correspond donc d'un certain sens à une condition pour la conservation de l'énergie pour le système.

Matrice anti-symétrique Une matrice est dit anti-symétrique, *skew-symmetric* en anglais, si ...

6.6 Dynamique inverse

L'équation des manipulateurs détermine des forces généralisée $\underline{\tau}$ pour une configuration \underline{q} , une vitesse $\underline{\dot{q}}$ et une accélération $\underline{\ddot{q}}$. La dynamique inverse représente la même équation mais lorsqu'on isole le vecteur accélération comme une fonction de toutes les forces internes et externes, c'est en fait le seul moyen de la causalité. En version compacte on peut retrouver :

$$\underline{\ddot{q}} = H^{-1} (\underline{\tau} - \underline{h}) \quad (6.31)$$

ou en version longue :

$$\underline{\ddot{q}} = H^{-1}(\underline{q}) \left(B(\underline{q}) \underline{u} + J^T(\underline{q}) \underline{f}_{R_E} - C(\underline{q}, \underline{\dot{q}}) \underline{\dot{q}} - d(\underline{q}, \underline{\dot{q}}) - \underline{g}(\underline{q}) \right) \quad (6.32)$$

Comme la matrice d'inertie H est symétrique et définie positive, son inverse existe toujours et donc cette équation a toujours une solution.

6.7 Systèmes sous-actionnés vs. complètement actionnés

Une caractéristique importante pour déterminer les capacités physique d'un robot manipulateur est de déterminer si le système est pleinement ou sous-actionné. Un système pleinement actionné a la capacité (théorique) d'imposer un vecteur d'accélération $\underline{\ddot{q}}$ arbitraire si on suppose que le vecteur d'entrée \underline{u} peut prendre des valeurs arbitraires (pas de maximum ou minimum). La condition mathématique pour être pleinement actionné est que le rang (rangé si la matrice n'est pas carrée) de B doit être égal à n , le nombre de DDL. On dira alors que le système est sous-actionné si

$$\text{rang}(B(\underline{q})) < n = \dim(\underline{q}) \quad (6.33)$$

6.8 Manipulabilité dynamique

À la section 4.7, la notion de manipulabilité définissait une enveloppe de vitesse possible à l'effecteur. Il est aussi possible d'analyser l'enveloppe d'accélérations cartésiennes possibles à l'effecteur pour un vecteur de couple des actionneurs unitaire.

Détails à venir !

6.9 Dérivation des équations avec la méthode de Lagrange

Pour des systèmes robotiques et mécanismes articulés relativement simples, la méthode de Lagrange est un outil bien adapté pour calculer rapidement analytiquement les équations du mouvement. Pour utiliser cette méthode, il suffit de déterminer les expressions pour l'énergie cinétique et potentielle en fonction des coordonnées généralisées :

$$\mathcal{L} = T - V = \underbrace{T(\underline{q}, \dot{\underline{q}})}_{\text{Énergie cinétique}} - \underbrace{V(\underline{q})}_{\text{Énergie potentielle}} \quad (6.34)$$

Ensuite il suffit d'appliquer une "recette de cuisine" qui consiste à effectuer des dérivées pour obtenir les équations de la dynamique. Chaque ligne i de l'équation des manipulateurs peut être déterminée avec l'expression suivante :

$$\forall i \quad \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i} = \sum \tau_i \quad (6.35)$$

qui est aussi équivalente à :

$$\forall i \quad \underbrace{\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_i} \right)}_{\text{Effets inertiels}} - \frac{\partial T}{\partial q_i} + \underbrace{\frac{\partial V}{\partial q_i}}_{\text{Force conservatives}} = \underbrace{\sum \tau_i}_{\text{Forces dissipatives, externes et actionneurs}} \quad (6.36)$$

On voit donc que cette méthode est surtout intéressante pour déterminer les termes inertIELS et les forces conservatives, et ne guide pas vraiment la détermination des autres forces ni leur expression comme des forces généralisées. L'avantage de cette méthode est qu'il est possible d'ignorer les forces de contraintes dans la démarche, versus la méthode classique qui demanderait de débuter par des diagrammes de corps libres (DCL), appliqué $f = ma$ sur tout les axes et faire un processus d'élimination des forces de contrainte.



Capsule vidéo
Méthode de Lagrange pour déterminer les équations dynamiques
<https://youtu.be/AUcs0L85liM>

Note : Il est à noté que la méthode de Lagrange n'est pas un type de modèle mais bien une méthodologie, i.e. une recette de cuisine pour déterminer les équations dynamiques d'un système de corps rigides. En utilisant la méthode classique (DCL + lois de Newtons) ou n'importe quelle autre méthode, si on part des mêmes hypothèses de modélisation on va trouver exactement les mêmes équations.

6.10 Équations dans l'espace de la tâche

En dérivant la relation de cinématique différentielle il est possible d'obtenir des relations entre les vitesses et accélérations :

$$\dot{\underline{r}} = J\underline{\dot{q}} \quad (6.37)$$

$$\ddot{\underline{r}} = J\ddot{\underline{q}} + \dot{J}\underline{\dot{q}} \quad (6.38)$$

Ensuite si on utilise ces équations pour remplacer les variables de l'espace des joints par ceux dans l'espace de la tâche dans l'équation des manipulateurs :

$$H\underline{\ddot{q}} + \underline{c} = \underline{\tau} + J^T \underline{f}_{R_E} \quad (6.39)$$

$$HJ^{-1}(\ddot{\underline{r}} - \dot{J}\underline{\dot{q}}) + \underline{h} = \underline{\tau} + J^T \underline{f}_{R_E} \quad (6.40)$$

et qu'on multiplie ensuite par la droite par l'inverse du jacobien transposé ($J^{-T} = (J^T)^{-1}$), nous obtenons :

$$J^{-T} H J^{-1} \ddot{\underline{r}} - J^{-T} H J^{-1} \dot{J} \underline{\dot{q}} + J^{-T} \underline{h} = J^{-T} B \underline{u} + J^{-T} J^T \underline{f}_{R_E} \quad (6.41)$$

$$\underbrace{J^{-T} H J^{-1} \ddot{\underline{r}}}_{H^r} + \underbrace{J^{-T} \underline{h} - J^{-T} H J^{-1} \dot{J} J^{-1} \dot{\underline{r}}}_{h^r} = \underbrace{J^{-T} B \underline{u}}_{B^r} + \underline{f}_{R_E} \quad (6.42)$$

Nous pouvons ensuite réorganiser les termes pour obtenir une nouvelle équation, qui a la même forme originale que l'équation des manipulateurs dans l'espace des joints, mais avec tout les termes par rapport aux coordonnées de la tâche :

$$H^r \ddot{\underline{r}} + h^r = B^r \underline{u} + \underline{f}_{R_E} \quad (6.43)$$

où

$$H^r = J^{-T} H J^{-1} \quad (6.44)$$

$$h^r = J^{-T} \underline{h} - H^r \dot{J} J^{-1} \dot{\underline{r}} \quad (6.45)$$

$$B^r = J^{-T} B \quad (6.46)$$

Cette équation représente la même dynamique mais exprimée avec les coordonnées de l'espace de la tâche directement. Il est à noter que cette transformation de coordonnées n'est pas possible sur une singularité. Si l'espace de la tâche représente des coordonnées cartésienne de l'outil du robot, alors la matrice H^r représente l'inertie ressentie lorsqu'on pousse sur l'effecteur. Le terme H_{11}^r est la masse ressentie dans la direction de l'axe 1.

6.11 Manipulateur en contact avec l'environnement

Cette section présente les outils pour obtenir les équations du mouvement lorsqu'un système robotique est en contact avec l'environnement, ce qui constraint sont mouvement.

6.11.1 Contraintes cinématique

Si un robot manipulateur est en contact avec un objet fixe, il pert certains DDL. Dans le cas de contraintes bilatérales, la contrainte peut être décrite par une fonction :

$$\underline{\phi}(\underline{q}) = 0 \quad (6.47)$$

Si on dérive cette fonction par rapport au temps, il est possible d'obtenir des conditions pour la vitesse et l'accélération du système :

$$\frac{d\underline{\phi}(\underline{q})}{dt} = J_C(\underline{q})\dot{\underline{q}} = 0 \quad (6.48)$$

$$\frac{d^2\underline{\phi}(\underline{q})}{dt^2} = J_C(\underline{q})\ddot{\underline{q}} + \dot{J}_C(\underline{q})\dot{\underline{q}} = 0 \quad (6.49)$$

où J_C est le jacobien des contraintes :

$$J_C(\underline{q}) = \frac{d\underline{\phi}(\underline{q})}{d\underline{q}} \quad (6.50)$$

6.11.2 Forces de contraintes

Le jacobien des contraintes peut être utilisé pour transformer des forces de contact cartésiennes \underline{f}_C en force généralisée au joint dans l'équation des manipulateurs :

$$H\ddot{\underline{q}} + \underline{h} = B\underline{\tau} + J_C(\underline{q})^T \underline{f}_C \quad (6.51)$$

Si on isole $\ddot{\underline{q}}$ dans l'équation (6.51) pour ensuite substituer dans l'équation (6.49), on retrouve une expression des forces de contraintes qui dépend des états actuel du système et des forces/couples aux actionneurs :

$$\underline{f}_C = (J_C H^{-1} J_C^T)^{-1} \left(J_C H^{-1} [\underline{h} - B\underline{\tau}] - \dot{J}_C(\underline{q})\dot{\underline{q}} \right) \quad (6.52)$$

Alternativement il est possible, pour trouver l'accélération $\ddot{\underline{q}}$ et les forces de contact \underline{f}_C simultanément, de résoudre le système d'équation suivant :

$$\begin{bmatrix} H & -J_C^T \\ J_C & 0 \end{bmatrix} \begin{bmatrix} \ddot{\underline{q}} \\ \underline{f}_C \end{bmatrix} = \begin{bmatrix} B\underline{\tau} - \underline{h} \\ -\dot{J}_C \dot{\underline{q}} \end{bmatrix} \quad (6.53)$$

6.11.3 Impulsion lors d'un impact

Quand le robot rentre en contact avec un objet fixe très rigide (frapper le sol par exemple), des forces de contact impulsives vont agir sur le système. Si on intègre l'équation (6.51) sur une période infinitésimal de temps dt donne :

$$\int (H\ddot{\underline{q}} + \underline{h}) dt = \int (B\underline{\tau} + J_C(\underline{q})^T \underline{f}_C) dt \quad (6.54)$$

$$H\dot{\underline{q}}^+ - H\dot{\underline{q}}^- = J_C(\underline{q})^T \int \underline{f}_C dt \quad (6.55)$$

où la contribution des forces non-impulsive est négligée. Si on projette ces équations sur les coordonnées contraintes en multipliant par $J_C H^{-1}$ on trouve :

$$J_C \dot{\underline{q}}^+ - J_C \dot{\underline{q}}^- = J_C H^{-1} J_C^T \int \underline{f}_C dt \quad (6.56)$$

Si on fait l'hypothèse d'une collision complètement inélastique (sans rebond), la contrainte est respectée à l'instant t^+ qui suit immédiatement l'impact. Dons comme $J_C \dot{q}^+ = 0$, il est ensuite possible de résoudre pour obtenir l'impulsion due au contact dans ces conditions :

$$\int \underline{f}_C dt = - (J_C H^{-1} J_C^T)^{-1} J_C \dot{q}^- \quad (6.57)$$

et pour la vitesse des joints qui va suivre immédiatement l'impact :

$$\dot{q}^+ = - [I - H^{-1} J_C^T (J_C H^{-1} J_C^T)^{-1} J_C] \dot{q}^- \quad (6.58)$$

ou pour la variation de vitesse durant l'impulsion :

$$\Delta \dot{q} = [H^{-1} J_C^T (J_C H^{-1} J_C^T)^{-1} J_C] \dot{q}^- \quad (6.59)$$

On peut alternativement résoudre le système de $n + c$ équations suivant pour obtenir ces résultats simultanément :

$$\begin{bmatrix} H & -J_C^T \\ J_C & 0 \end{bmatrix} \begin{bmatrix} \dot{q}^+ \\ \int \underline{f}_C dt \end{bmatrix} = \begin{bmatrix} H \dot{q}^- \\ 0 \end{bmatrix} \quad (6.60)$$

6.12 Dynamique des actionneurs et effet des ratios de réduction

Jusqu'à maintenant les équations étaient développées en supposant que l'entrée sur le système robotique était une source de force pure. La plupart des robots sont actionnés par des moteurs électriques, pour lesquels le couple appliqué par le moteur est relativement proportionnel au courant qui circule dans le moteur qu'il est possible d'asservir avec de l'électronique de puissance adapté. Toutefois, l'inertie et la friction interne de ces moteurs va avoir un gros impact sur la dynamique des robots lorsque de grands ratios de réductions sont utilisés comme c'est le cas pour les manipulateurs industriels.

Pour illustrer ce phénomène, débutons avec un modèle simple de dynamique de moteur qui aurait une inertie I , de la friction visceuse b , un couple transmis net τ_{net} et comme entrée un couple électromagnétique τ_{mag} . La dynamique serait donnée par :

$$I\dot{w} + bw = \tau_{mag} - \tau_{net} \quad (6.61)$$

Si un ratio de réduction R est utilisé entre le moteur et le joint du robot, alors le couplage entre le bras robotique et le moteur est donné par ces deux équations :

$$\tau = R\tau_{net} \quad (6.62)$$

$$w = R\dot{q} \quad (6.63)$$

détails à venir !

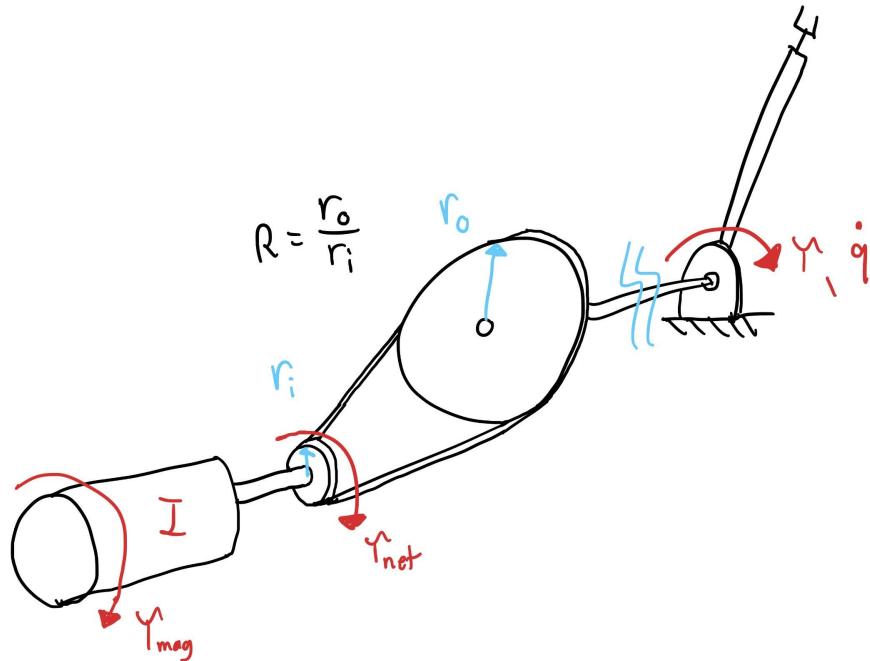


FIGURE 6.7 – Dynamique d'un moteur électrique couplé avec un joint robotique

Chapitre 7

Introduction à la commande des systèmes robotisés

La science de *la commande* traite du développement d'algorithmes qui déterminent les actions qu'un système robotisé doit prendre en fonction de la situation pour atteindre un objectif (par exemple une position cible). Ces algorithmes sont généralement implémentés sur un micro-contrôleur, qui choisit les actions des actionneurs (par exemple le couple des moteurs électriques). Mathématiquement, le problème consiste à choisir les termes qui représentent les actions possibles dans un système d'équations différentielles qui représentent le comportement du robot, afin d'obtenir le comportement désiré.

Lors de la conception des lois de commandes, plusieurs aspects (illustrés qualitativement à la Figure 7.1) peuvent être analysés selon le contexte :

La stabilité Lorsqu'une loi de commande est proposée, un critère essentiel dans la plupart des situations est la stabilité du système. La stabilité caractérise si les états du système convergent vers les états désirés ou bien si ils divergent.

La faisabilité Dans certains contextes, principalement ceux où les actions possibles sont limitées (ex. couple maximum des moteurs), on s'intéresse tout simplement à savoir si il existe une séquence d'actions possibles qui peuvent permettre au système d'atteindre l'objectif.

Le temps de réponse La stabilité et la faisabilité déterminent si le système pourra atteindre l'objectif. Ensuite, on cherche généralement à atteindre l'objectif le plus rapidement possible ce qui est caractérisé par le temps de réponse. Un critère relié est la bande passante, la rapidité du système exprimée dans le domaine fréquentiel.

L'optimalité Il y a généralement un compromis à faire entre la vitesse (le temps de réponse) et l'utilisation agressive des actions possibles du système qui ont généralement un coût énergétique. Un objectif commun lors de la conception d'un asservissement est d'optimiser la performance, ce qui est généralement formulé comme la minimisation d'un coût qui combine les dépenses énergétiques et le temps passé loin de l'objectif.

La robustesse Finalement, dans plusieurs situations il faut considérer l'incertitude dans le système lors de la conception des lois de commande. L'incertitude peut être due à des perturbations externes inconnues (ex. : rafales de vent pour un avion) ou bien à des paramètres incertains du système robotique (ex. : poids de la charge transportée pour un bras manipulateur). L'analyse de robustesse consiste à vérifier si les lois de commande vont fonctionner dans divers scénarios possibles. Des méthodes existent pour garantir le fonctionnement d'un contrôleur malgré un certain niveau d'incertitude (commande robuste ou commande adaptative).

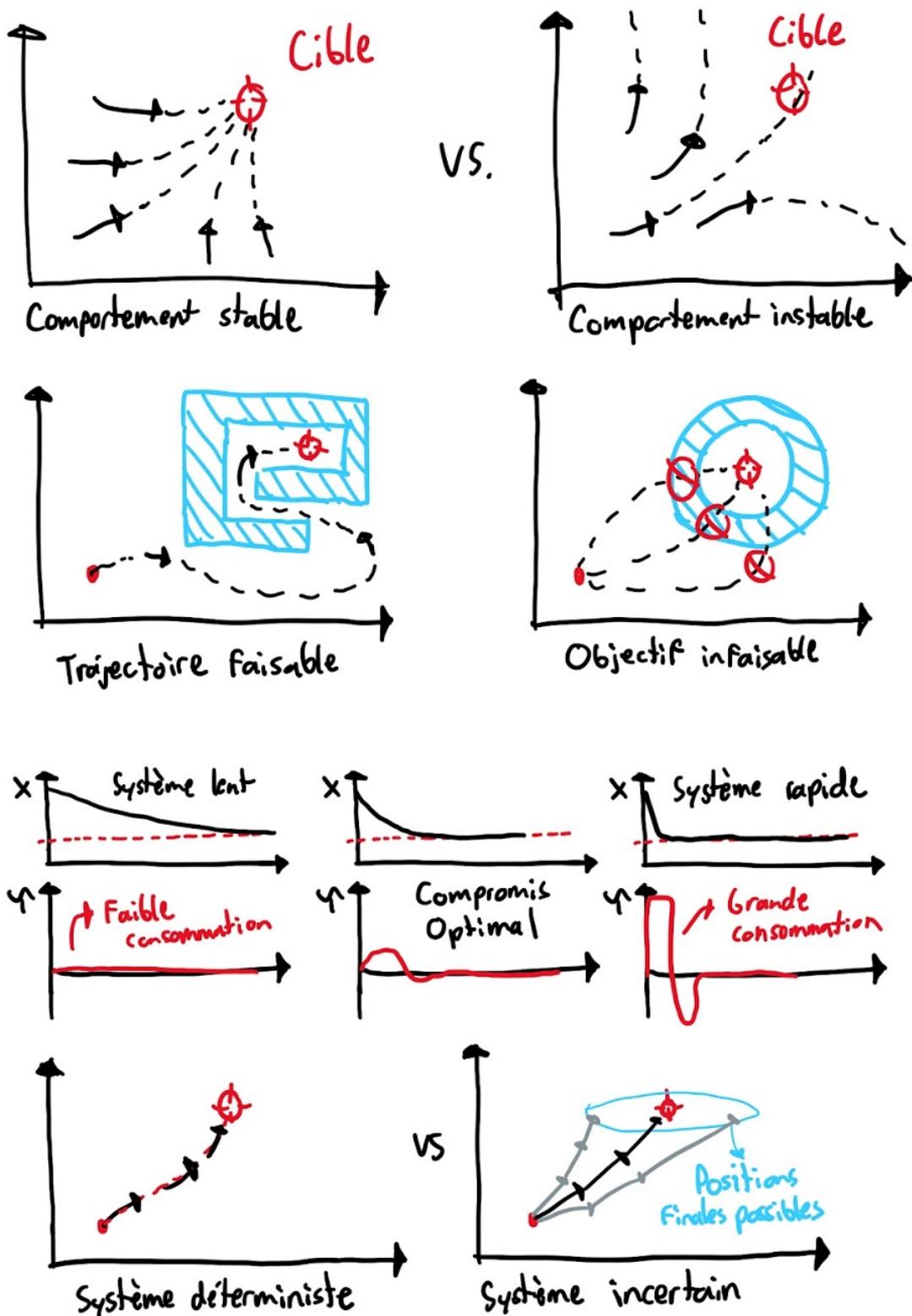


FIGURE 7.1 – Différents aspects qui peuvent être considéré lors de la conception de loi de commandes

7.1 Architectures

Cette section introduit aux grandes catégories d'architecture de commande pour les systèmes robotisés.

7.1.1 Boucle ouverte vs. boucle fermée

La méthode de commande la plus simple est appelée boucle ouverte, voir Figure 7.2. Par exemple, on détermine une séquence d'action pour les actionneurs (plan) qui va théoriquement faire bouger un système robotisé d'un point A au point B (objectif). Dans ce cas, la loi de commande est seulement une fonction du temps qui envoie les consignes pré-déterminées aux actionneurs.

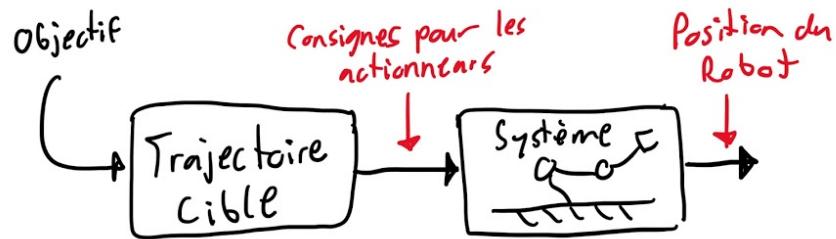


FIGURE 7.2 – Boucle ouverte

En général, cette méthode que l'on pourrait qualifier "d'aveugle" fonctionne seulement lorsque les systèmes ont très peu d'incertitude. Par exemple, une machine outil qui lit un code-G et traduit directement les instructions en consignes pour ses moteurs pas à pas est une approche purement boucle ouverte. Par contre, on peut facilement s'imaginer que cette approche ne fonctionnerait pas pour une voiture autonome qui doit aller d'une ville à une autre. Un véhicule autonome qui utiliserait un contrôleur basé sur une séquence d'angle de volant pré-déterminée divergerait rapidement de la trajectoire désirée du aux incertitudes.

La plupart des contrôleurs utilisent une rétroaction continue basé sur des capteurs pour vérifier si le système évolue de façon approprié. C'est ce qu'on appelle l'approche boucle fermée, illustré à la Figure 7.3. Dans ce cas, la loi de commande est une généralement fonction d'une erreur qui résulte de la comparaison de la position cible et la position réelle mesurée par des capteurs.

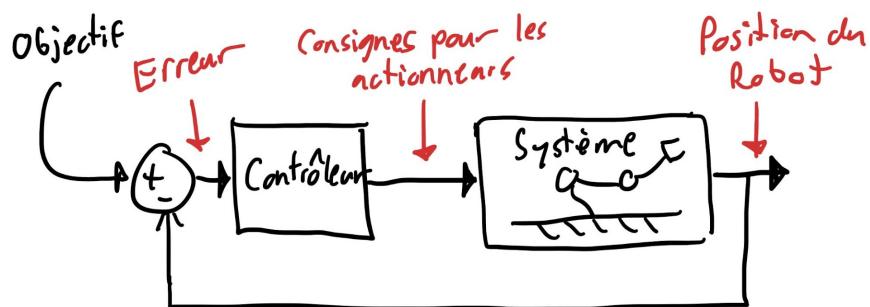


FIGURE 7.3 – Boucle fermée

Le principe de base d'une telle loi de commande peut se résumer à : si le système est trop à gauche pousse vers la droite et si le système est trop à droite pousse vers la gauche. Cette approche fonctionne bien pour des systèmes simples. Par exemple, pour contrôler la position d'un vérin électrique une loi de commande de type "pousse dans le sens inverse de l'erreur" est suffisante. Toutefois pour une voiture autonome, si la direction et la vitesse à prendre serait basée sur l'erreur totale, par exemple 150 km NW par rapport à la ville que l'on désire aller, la voiture tenterait de traverser des champs en ligne droite ! Pour les systèmes plus compliqués, particulièrement avec des limites comme des obstacles, des saturations d'actionneurs, etc, le contrôleur n'est pas simplement une fonction de l'erreur.

Pour plusieurs systèmes robotiques, une loi de commande hybride est utilisée comme illustré à la Figure 7.4. Un plan est déterminé comme avec l'approche en boucle ouverte, toutefois un contrôleur en boucle fermée qui agit sur l'erreur par rapport à la trajectoire cible est ensuite utilisé. La loi de commande dans ce cas, ne dépend pas seulement de la position actuel et de la position désirée, mais aussi du temps car elle compare la position actuelle à celle où le robot doit théorique être rendu à ce moment.

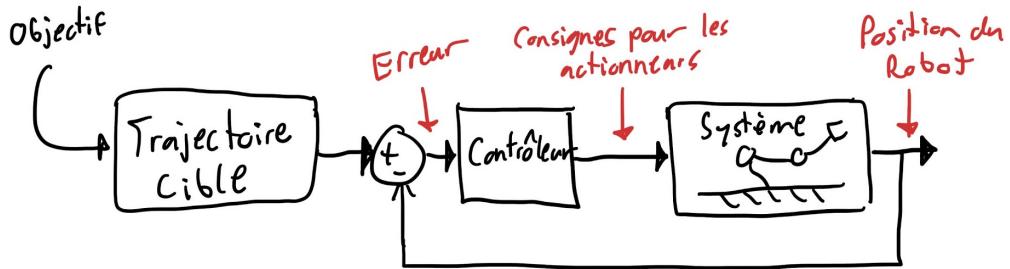


FIGURE 7.4 – Contrôleur de suivi de trajectoire

7.1.2 Boucles imbriquées

La plupart des systèmes robotiques n'ont pas un seul gros contrôleur qui contrôle toutes les actions des actionneurs de façon centralisé, mais plutôt plusieurs sous-systèmes avec chacun des contrôleur locaux. Par exemple, lorsqu'un système robotisé utilise des servo-moteurs, l'électronique des moteurs inclus déjà un contrôleur qui effectue un asservissement en position ou en vitesse.

Une architecture qui est souvent utilisée pour les robots manipulateur est illustrée à la Figure 7.5. Chaque joint du robot du robot a son propre contrôleur en vitesse qui reçoit une consigne d'un contrôleur central. Les contrôleurs locaux comparent la vitesse du joint désirée à une mesure de vitesse et détermine un couple moteur selon l'erreur en vitesse. Le contrôleur central lui compare la position cible à la position mesurée du robot et détermine le mouvement (consignes de vitesse) que les joints devraient faire pour corriger la position.

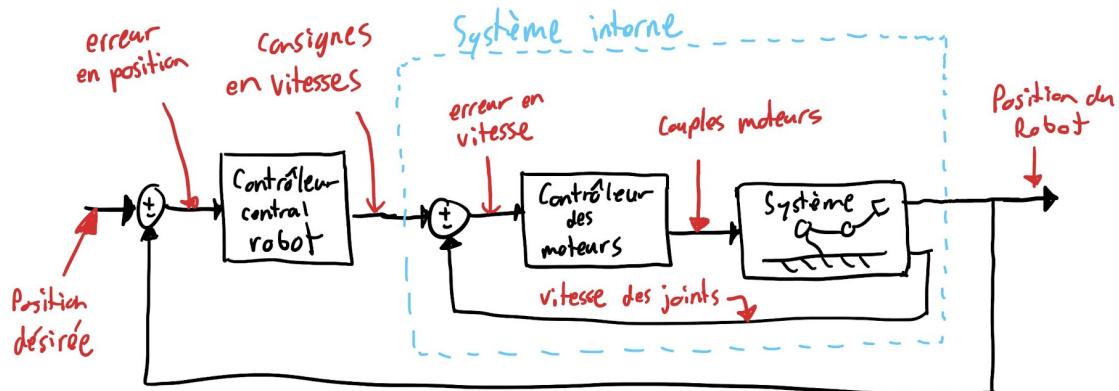


FIGURE 7.5 – Boucles imbriquées

Cette approche a comme avantage de découpler le problème en deux problèmes plus simples et rendre plus modulaires les loi de commande qui sont développées. Par exemple, le concepteur du contrôleur des moteurs pourrait effectuer son travail sans ce souci du modèle cinématique compliqué du bras manipulateur. Inversement, le concepteur du contrôleur central du robot pourrait travailler avec un modèle purement cinématique du robot (plutôt que les équations dynamiques qui relient les couples moteurs aux accélérations), en assumant que les boucles en vitesse des moteurs fonctionnent et sont rapide. Dans ces conditions, comme

illustré à la Figure 7.6, le concepteur du contrôleur central aurait un problème classique avec une seule boucle fermée.

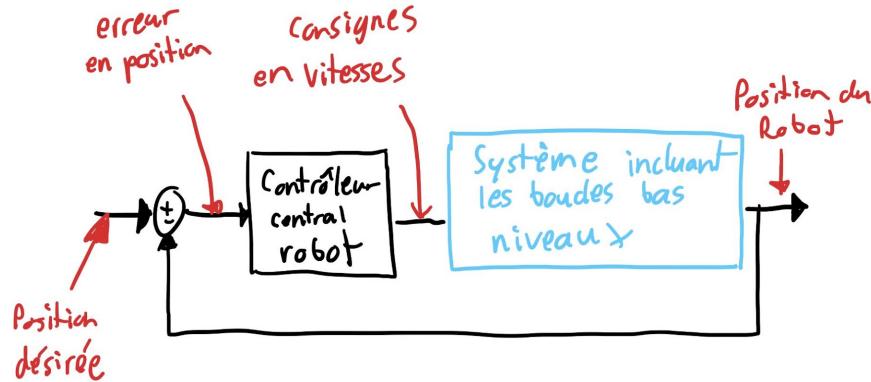


FIGURE 7.6 – Boucle fermée du contrôleur central

Il est donc important de noter que le modèle utilisé pour la conception du ou des contrôleurs d'un système robotisé doit être adapté au contexte et à l'architecture utilisée.

7.1.3 Synthétisation d'un plan hors-ligne ou en-ligne

Un autre aspect pour les méthodes de commandes avancées qui utilise beaucoup de calculs numériques, est qu'est-ce qui est calculé d'avance vs. en temps réel durant l'exécution du contrôleur. Une architecture souvent utilisée avec les algorithmes de planification/optimisation de trajectoire (ex. RRT) est illustrée à la Figure 7.7. Les calculs lourds de recherche et/ou optimisation sont effectués d'avance, et lors de l'exécution la loi de commande utilise seulement la trajectoire cible qui a été synthétisée hors-ligne.

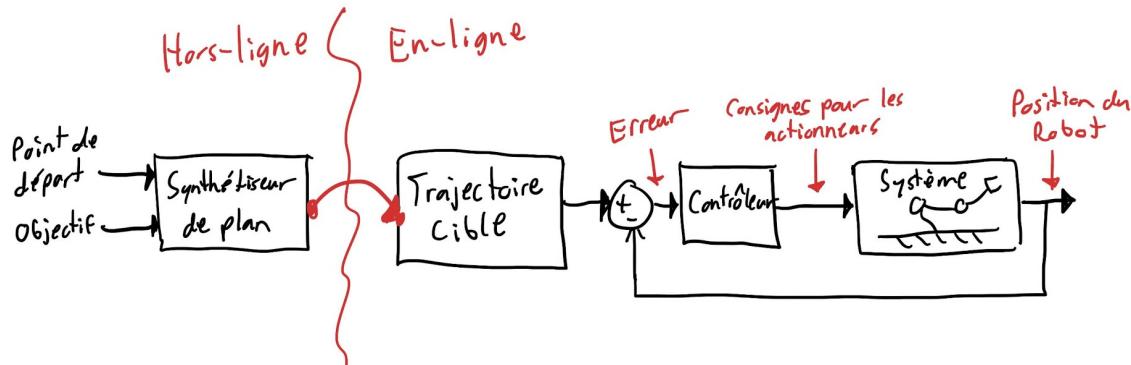


FIGURE 7.7 – Générateur de trajectoire hors-ligne

De façon similaire, les méthodes comme *LQR* ou bien *value-iteration* effectuent des calculs généralement hors-ligne pour synthétiser une loi de commande sous forme de carte $u = c(x)$ de quelles actions u prendre en fonction des états mesurés x du robot. C'est donc ici une fonction qui est synthétisée hors-ligne et utilisée en-ligne, voir Figure 7.8. La fonction synthétisée peut-être analytique, par exemple pour la méthode *LQR* avec un format $u = Kx$, ou bien numérique, comme par exemple pour un contrôleur synthétisé avec la méthode *value-iteration*.

Ensuite, certains types de contrôleurs de systèmes robotiques vont avoir une certaine forme de synthétisation de trajectoire ou de contrôleur mais en ligne, voir un exemple à la Figure 7.9. Typiquement, du au lourd calcul numérique qui doivent être fait en ligne, les architectures vont utiliser des boucles imbriquées avec une boucle interne plus rapide et une boucle externe plus lente qui met à jour la trajectoire cible. L'approche de

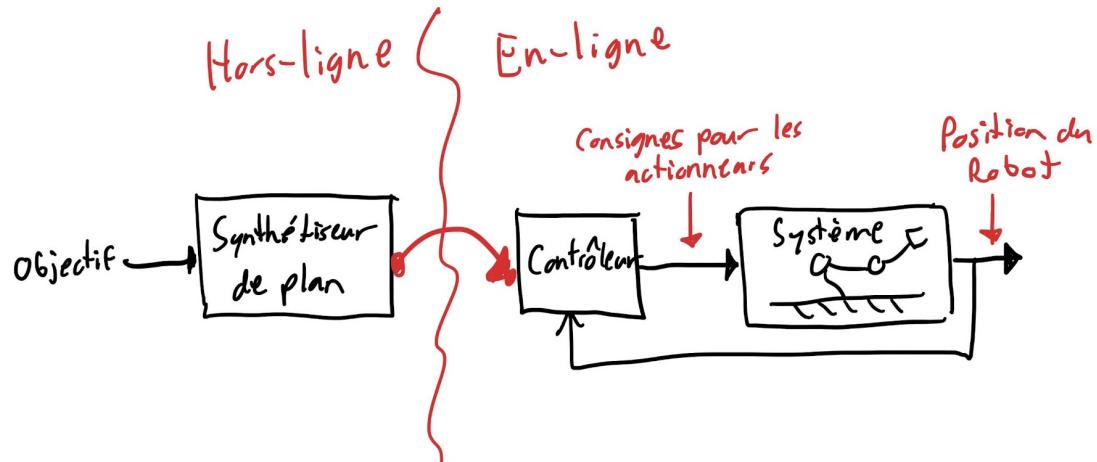


FIGURE 7.8 – Génération de contrôleur hors-ligne

commande prédictive (MPC pour *Model Predictive Control* en anglais), est un exemple où on optimise des trajectoires cible constamment en ligne dans la boucle.

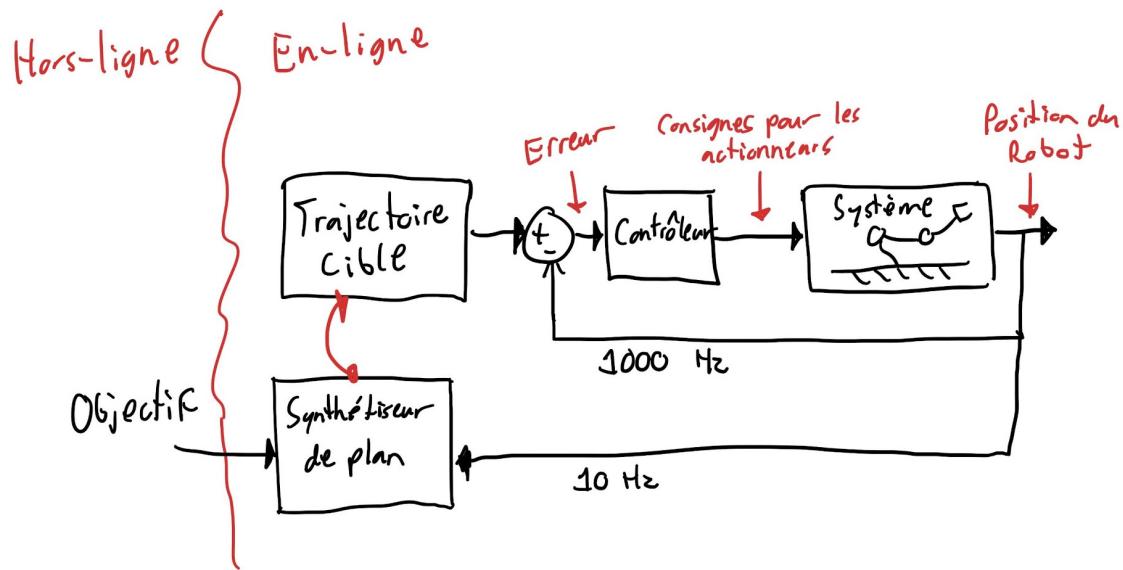


FIGURE 7.9 – Génération de trajectoire en ligne

Chapitre 8

Commande des robots manipulateurs I : quasi-statique

Les objectifs en termes de mouvement désiré d'un robot sont généralement plus naturellement spécifiés en termes de variables dans l'espace de la tâche d'un robot, alors que à bas niveau les consignes des actionneurs sont reliés à des variables dans l'espace des joints. Ce chapitre présente des méthodes de commande qui permettent de calculer les consignes pour les actionneurs basé sur des objectifs directement spécifié dans l'espace de la tâche, malgré la relation géométrique hautement non-linéaire. Les méthodes ici présentées utilisent grandement les notions de cinématique différentiel et de statique présentées dans les sections 4.5 et 5.

Les méthodes présentées dans ce chapitre négligent les effets dynamiques (inerties, frottement, etc.) et considèrent juste un comportement simplifié des robots : la relation statique non-linéaire entre le mouvement/force des actionneurs et le mouvement/force à l'effecteur. Ces méthodes sont généralement performante lorsque les mouvements du robot sont relativement lents, c'est pourquoi ils sont ici regroupé sous la caractéristique *quasi-statique*. Ensuite, selon la nature des actionneurs des systèmes robotiques, différentes variantes peuvent être utilisée :

Actionneurs commandés en vitesse : Pour les méthodes de commande du mouvement de l'effecteur présentées aux sections 8.1, 8.2 et 8.5, il est considéré que le robot a des asservissements bas-niveau en vitesse à chacun des joints. Ces méthodes calculent les consignes en vitesse à envoyer aux joints pour contrôler le mouvement de l'effecteur. Ces méthodes fonctionnent bien dans des situations où le suivi de consigne en vitesse des joints est très performant, c'est généralement le cas des manipulateurs industriels qui ont de très grands ratios de réduction.

Actionneurs commandés en force : Pour les méthodes de commande présentées aux sections 8.3 et 8.4, on considère seulement la relation géométrique entre les forces des actionneurs et ceux à l'effecteur. Les deux méthodes calculent des forces à appliquer au niveau des actionneurs/joint, pour contrôler la force au niveau de l'effecteur. Ces méthodes fonctionnent donc bien pour des systèmes robotisés à basse impédance (peu d'inertie, peu d'effets dissipatifs, transmission réversibles, etc.), comme les systèmes haptiques et certains robots collaboratifs où les forces des actionneurs sont pratiquement proportionnelles au courant dans les moteurs ou à des très petits ratios de transmissions.

8.1 Commande en vitesse de l'effecteur

Si un robot a des actionneurs contrôlés en vitesse à bas niveau, contrôler la vitesse de l'effecteur se résume à mettre en oeuvre la relation de cinématique différentielle inverse (voir section 4.5). Comme illustré par un schéma bloc à la Figure 8.1, les consignes en vitesse pour les actionneurs sont déterminées en multipliant le vecteur-colonne de vitesses désirées dans l'espace des tâches, par l'inverse du Jacobien qui relie l'espace des joints à l'espace des tâches du systèmes. Le Jacobien est généralement dépendant de la position des joints du robot ce qui nécessite une boucle de rétroaction basé sur les capteurs de position des actionneurs pour effectuer le calcul de J en continu basé sur la position actuelle des joints. Finalement, comme mis en évidence à la Figure 8.1, cette méthode s'intègre comme une boucle haut-niveau pour coordonner les différents joints d'un système robotisé où chacun des actionneurs est asservis en vitesse, souvent avec des asservissements bas-niveaux implémentées directement dans l'électronique de contrôle des moteurs.

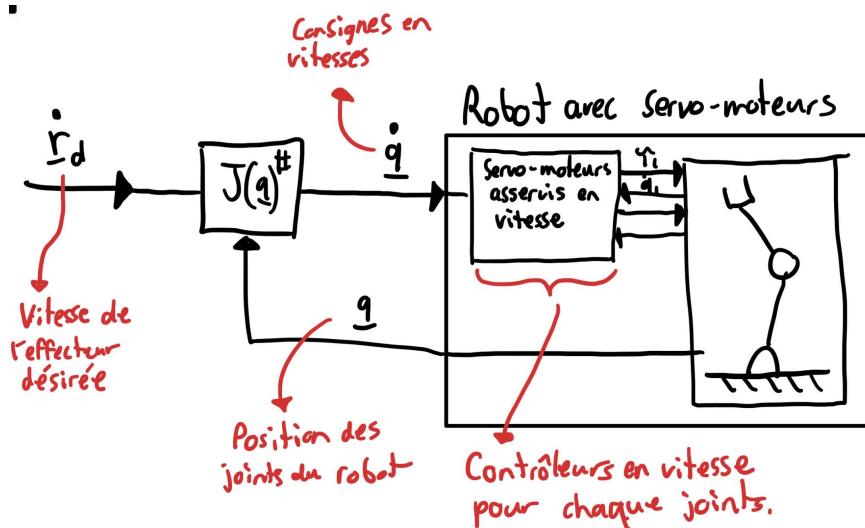


FIGURE 8.1 – Commande de la vitesse de l'effecteur d'un robot : schéma bloc

Si le nombre de joints n est égale au nombre de DDL de l'espace de la tâche m , alors la matrice Jacobienne est carrée et peut être inversée (sauf sur les singularités). Si le nombre de joint n est supérieur au nombre de DDL de l'espace de la tâche m , alors on peut utiliser une matrice pseudo-inverse droite $J^\# = J^T(JJ^T)^{-1}$ (voir section 18.4). La loi de commande en équation peut donc être exprimée comme :

$$\dot{q} = \begin{cases} J(\underline{q})^{-1}\dot{r}_d & \text{if } n = m \\ J(\underline{q})^\# \dot{r}_d & \text{if } n > m \end{cases} \quad (8.1)$$

En substituant la loi de commande dans la relation de cinématique différentielle, on confirme que la vitesse de l'effecteur sera exactement la vitesse désirée :

$$\dot{r} = J(\underline{q}) \dot{q} \quad (8.2)$$

$$\dot{r} = J(\underline{q}) J(\underline{q})^\# \dot{r}_d \quad (8.3)$$

$$\dot{r} = JJ^T(JJ^T)^{-1} \dot{r}_d \quad (8.4)$$

$$\dot{r} = \dot{r}_d \quad (8.5)$$

sous les hypothèses que : 1) le Jacobien utilisé par le contrôleur est exacte, 2) le Jacobien est inversible (i.e. le robot n'est pas sur une singularité) et 3) la vitesse des joints est parfaitement asservis par les boucles bas-niveaux.

8.2 Commande en position de l'effecteur

Pour commander la position de l'effecteur d'un robot, tenter de trouver une solution directement au problème de cinématique inverse (voir section 3.9) n'est généralement pas la méthode la plus appropriée car la cinématique directe des manipulateurs est hautement non-linéaire. La méthode ici présentée utilise plutôt la méthode de commande en vitesse de l'effecteur (section 8.1) pour indirectement résoudre la cinématique inverse du robot. Le principe ce résume au fait que pour contrôler la position de l'effecteur, il suffit de diriger le vecteur vitesse de l'effecteur vers la position cible. La méthode est illustrée graphiquement à la Figure 8.2 : 1) Un vecteur d'erreur \underline{r}_e est calculé en comparant la position désirée \underline{r}_d à la position actuelle \underline{r} . 2) Le vecteur d'erreur \underline{r}_e est multiplié par un paramètre scalaire λ pour déterminer la vitesse cible instantanée pour l'effecteur notée $\dot{\underline{r}}_r$, qui pointe en direction de la cible. 3) L'inverse du Jacobien est utilisé pour convertir la vitesse instantanée désirée de l'effecteur en consignes de vitesse pour les joints.

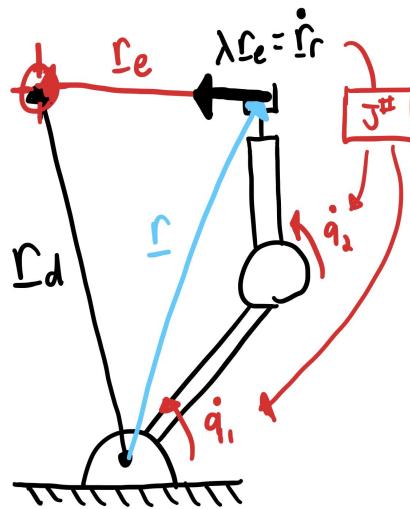


FIGURE 8.2 – Commande de la trajectoire de l'effecteur d'un robot : interprétation géométrique

Capsule vidéo
Commande du mouvement de l'effecteur d'un robot manipulateur
<https://youtu.be/Qo60ySYaMqg>

Formellement, la loi de commande est exprimée par l'expression mathématique :

$$\dot{\underline{q}} = \begin{cases} J(\underline{q})^{-1} \lambda \underbrace{\left(\underline{r}_d - f(\underline{q}) \right)}_{\underline{r}_e} & \text{if } n = m \\ J(\underline{q})^\# \lambda \underbrace{\left(\underline{r}_d - f(\underline{q}) \right)}_{\underline{r}_e} & \text{if } n > m \end{cases} \quad (8.6)$$

ou λ est un paramètre scalaire de gain du contrôleur, J est le Jacobien, la matrice $n \times m$, qui relie l'espace des joints à l'espace de la tâche du manipulateur, et $f(\underline{q})$ la fonction de cinématique directe du manipulateur. La Figure 8.3 illustre cette méthode de commande sous la forme d'un schéma bloc.

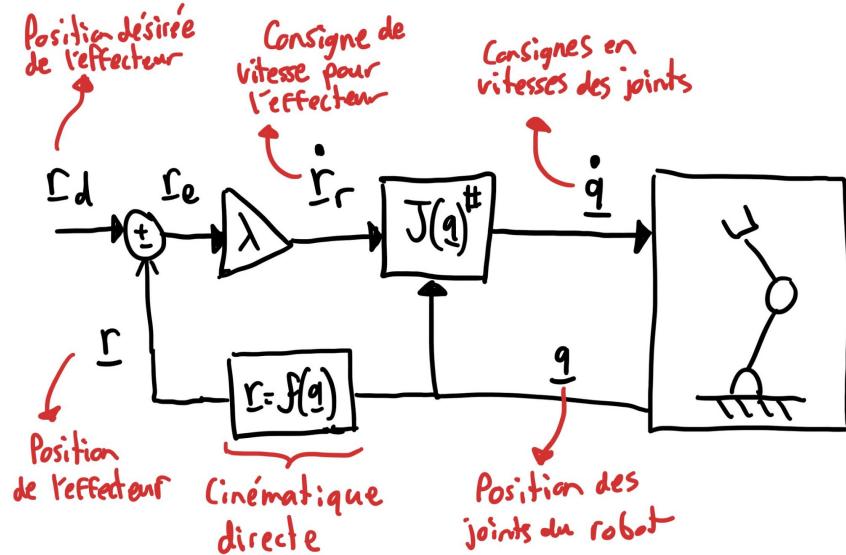


FIGURE 8.3 – Commande de la position de l'effecteur d'un robot : schéma bloc

**Exercice de code***Commande d'un manipulateur en position*https://colab.research.google.com/drive/1M30_nD8iLbRSvZzyUpBn0Mo8KUclrG26?usp=sharing

8.2.1 Suivi de trajectoire

Lorsque le robot doit suivre une position cible de l'effecteur qui varie dans le temps, il est préférable de calculer la dérivée temporelle de la trajectoire et d'utiliser cette information directement dans la loi de commande comme indiqué dans l'équation suivante :

$$\dot{\underline{q}} = \begin{cases} J(\underline{q})^{-1} \left[\dot{r}_d + \lambda \left(r_d - \underbrace{f(\underline{q})}_{\underline{r}} \right) \right] & \text{if } n = m \\ J(\underline{q})^\# \left[\dot{r}_d + \lambda \left(r_d - \underbrace{f(\underline{q})}_{\underline{r}} \right) \right] & \text{if } n > m \end{cases} \quad (8.7)$$

En terme de schéma bloc, la vitesse de la trajectoire doit être utilisée comme illustré à la Figure 8.4, ce qu'on appelle un *feedforward* en anglais, pour garantir la convergence sur la trajectoire.

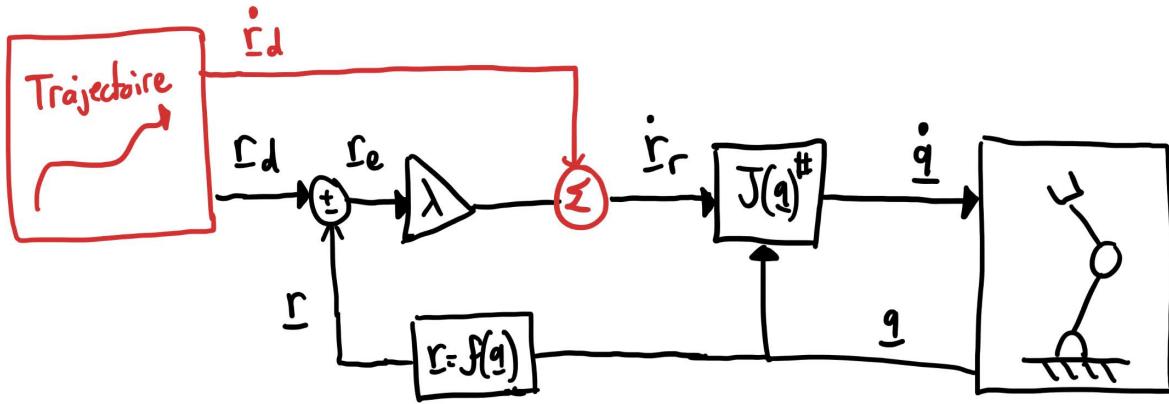


FIGURE 8.4 – Commande de la trajectoire de l'effecteur d'un robot : schéma bloc

8.2.2 Convergence

Les méthodes de commande en position de l'effecteur convergent sous certaines conditions, i.e. l'erreur (sur la position de l'effecteur) converge vers zéro lorsque le temps tend vers l'infini :

$$\lim_{t \rightarrow \infty} r_e(t) = 0 \quad (8.8)$$

Preuve 8.2.1 L'erreur est une fonction de la position désirée et de la position réelle du robot :

$$r_e = r_d - r \quad (8.9)$$

Si on dérive cette équation dans le temps, on obtient une relation différentielle. On peut alors substituer le modèle de cinématique différentiel et les lois de commandes pour obtenir :

$$\dot{r}_e = \dot{r}_d - \dot{r} \quad (8.10)$$

$$\dot{r}_e = \dot{r}_d - J(\underline{q})\dot{\underline{q}} \quad (8.11)$$

$$\dot{r}_e = \dot{r}_d - \underbrace{J(\underline{q})J(\underline{q})^{\#}}_1 \dot{r}_r \quad (8.12)$$

$$\dot{r}_e = \dot{r}_d - \underbrace{J(\underline{q})J(\underline{q})^{\#}}_1 (\dot{r}_d + \lambda r_e) \quad (8.13)$$

$$\dot{r}_e = -\lambda r_e \quad (8.14)$$

La dynamique de l'erreur est une équation différentielle d'ordre 1. Si la constante de temps est positive $\lambda > 0$ (ici directement déterminée par le gain de notre contrôleur), l'erreur converge exponentiellement vers zéro :

$$\dot{r}_e = -\lambda r_e \quad \Rightarrow \quad r_e(t) = r_e(t=0) e^{-\lambda t} \quad \Rightarrow \quad r_e(t=\infty) = 0 \quad (8.15)$$

Notez ici que cette analyse assume un contrôle parfait et instantané de la vitesse des moteurs. Une autre limite est que la convergence cesse si le robot passe sur une singularité, i.e. mathématiquement l'inverse (ou pseudo-inverse) du Jacobien n'excisera pas. Finalement, comme démontré dans la démarche, pour garantir la convergence exponentielle sur une trajectoire (position cible qui varie dans le temps), le terme de *feedforward* illustré à la Figure 8.4 est nécessaire.

8.2.3 Utilisation de l'espace nul pour un objectif secondaire

Comme il a été vu à la section 4.6.2, lorsque le nombre de DDL du robot $n = \dim(\underline{q})$ est supérieur à celui de l'espace de la tâche qu'on désire contrôler $m = \dim(\underline{r})$, il y a plusieurs solutions $\dot{\underline{q}}$ pour lesquels la vitesse cible pour l'effecteur est parfaitement atteinte. Lorsque la consigne de vitesse de l'effecteur $\dot{\underline{r}}_d$ est fixée par la loi de commande qui la fait converger vers \underline{r}_d , l'ensemble des solutions pour $\dot{\underline{q}}$ peuvent être décrite par l'équation suivante :

$$\dot{\underline{q}} = J^\# \underbrace{\left[\dot{\underline{r}}_d + \lambda \left(\underline{r}_d - \underbrace{f(\underline{q})}_{\dot{\underline{r}}} \right) \right]}_{\dot{\underline{r}}_r} + [I - J^\# J] \underline{v} \quad (8.16)$$

où le vecteur-colonne \underline{v} contient les variables libres. On peut donc considérer le vecteur \underline{v} comme une entrée de contrôle secondaire, qu'il est possible d'utiliser pour faire autre chose avec le robot sans affecter la convergence de l'effecteur. Par exemple, les bras humains : on peut les utiliser pour attraper un objet avec notre main (objectif principal), tout en utilisant notre bras pour tenir un cahier contre notre corps (objectif secondaire). On peut aussi utiliser l'espace nul pour tenter de garder la configuration du robot le plus loin possible des configurations singulières. La Figure 8.5 illustre graphiquement l'effet de divers options de vecteur \underline{v} sur les mouvements internes des joints d'un robot manipulateur redondant.

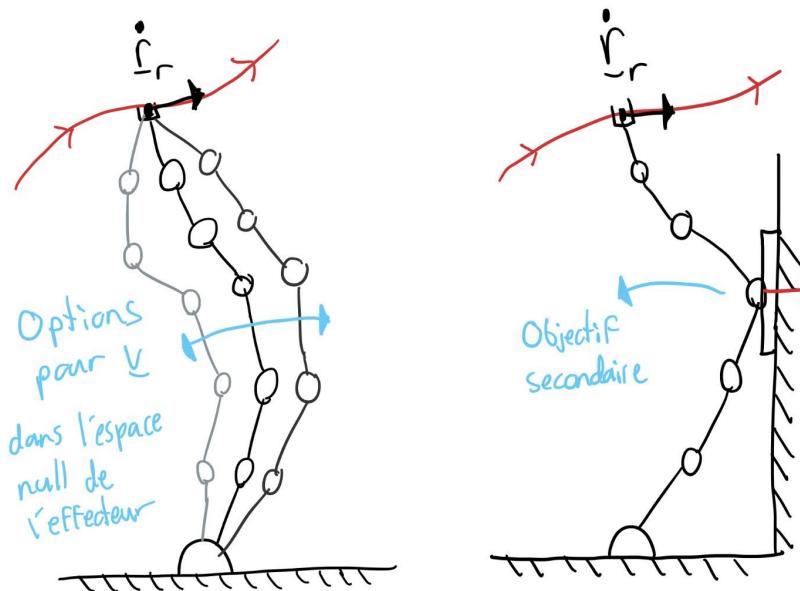


FIGURE 8.5 – Utilisation de l'espace nul d'un robot redondant pour effectuer un objectif secondaire



Capsule vidéo

Commande de l'effecteur d'un robot redondant

<https://youtu.be/ROT4shGPBKo>

De façon général, on peut formuler l'objectif secondaire comme une fonction potentielle scalaire G qui dépend de la configuration (un scalaire qui est le plus grand possible le plus on est loin de l'objectif). La commande secondaire \underline{v} peut alors être déterminée par le gradient de cette fonction évalué à la configuration actuelle du robot :

$$G(\underline{q}) \Rightarrow \underline{v} = \frac{\partial G}{\partial \underline{q}} \quad (8.17)$$

Si l'objectif secondaire est une position cible \underline{q}_d dans l'espace des joints. La fonction objectif G pourrait être la norme de l'erreur, et on obtiendrait :

$$G(\underline{q}) = -\frac{1}{2} \lambda \underline{q}_e^T \underline{q}_e \quad \text{avec : } \underline{q}_e = \underline{q}_d - \underline{q} \quad (8.18)$$

$$\underline{v} = \frac{\partial G}{\partial \underline{q}} = \lambda (\underline{q}_d - \underline{q}) \quad (8.19)$$

où λ est un paramètre de gain de convergence pour cette boucle de commande secondaire avec une cible de configuration \underline{q}_d dans l'espace des joints. Notez que la boucle principale a toujours la priorité absolue avec cette formulation, la convergence de l'objectif principal est garantie mais pas celle de l'objectif secondaire.



Exercice de code

Espace nul d'un robot redondant

<https://colab.research.google.com/drive/16ACenFOLOHVNeReqJTbkATAB3281iVbp?usp=sharing>

8.2.4 Formulation avec régulation de la norme du vecteur vitesse

Dans certaines situations, par exemple lorsque le robot passe proche d'une singularité, les lois de commande présentées dans les sections précédentes peuvent demander des très grandes vitesses irréalistes aux joints. Plutôt que d'inverser directement la matrice Jacobienne, il est possible de trouver une solution au système d'équation suivant qui pénalise aussi la norme du vecteur $\dot{\underline{q}}$:

$$\begin{bmatrix} \dot{\underline{r}}_r \\ \underline{0} \end{bmatrix}_{(m+n) \times 1} = \begin{bmatrix} J(\underline{q}) & \\ \lambda I & \end{bmatrix}_{(m+n) \times n} \begin{bmatrix} \dot{\underline{q}} \\ \underline{q} \end{bmatrix}_{n \times 1} \quad (8.20)$$

où λ est un paramètre de poids sur la pénalité d'utiliser des grandes vitesses de joints. Ce système a donc $m+n$ équations et n variables, il est donc sur-constraint. Une méthode standard est d'utiliser la solution des moindres-carrés (voir section 18.3) pour trouver une solution non-exacte pour $\dot{\underline{q}}$ qui minimise la norme de l'erreur au carré, ce qui correspond ici à minimiser :

$$\|\underline{e}\|^2 = \|\dot{\underline{r}}_r - J\dot{\underline{q}}\|^2 + \lambda^2 \|\dot{\underline{q}}\|^2 \quad (8.21)$$

La solution des moindres carrés a une solution explicite qui correspond à

$$\hat{\dot{\underline{q}}} = \underset{\dot{\underline{q}}}{\operatorname{argmin}} \|\underline{e}\|^2 \quad (8.22)$$

$$\hat{\dot{\underline{q}}} = (J^T J + \lambda^2 I)^{-1} J^T \dot{\underline{r}}_r \quad (8.23)$$

La convergence de cette méthode n'est toutefois pas garantie, le robot peut dériver de la trajectoire désirée surtout si λ est trop grand. Une méthode pour palier à ce problème est d'ajuster dynamiquement λ pour lui assigner de grandes valeurs seulement proche des configurations problématiques, i.e. proche des singularités. Dans la littérature, la méthode présentée dans cette section est appelée *damped least-square*.



Capsule vidéo

Commande en position avec régulation de vitesse

<https://youtu.be/n3G-07cpQTQ>

8.3 Commande en force de l'effecteur

Pour certaines tâches, ce n'est pas la position de l'outil que l'on désire contrôler, mais plutôt la force qu'il applique sur l'environnement. Si un systèmes robotisé a des actionneurs qui sont contrôlable en force ou couple, il suffit d'une transformation géométrique avec le Jacobien pour calculer les consignes en force des actionneurs basé sur une force désirée à l'effecteur, comme illustré à la Figure 8.6.

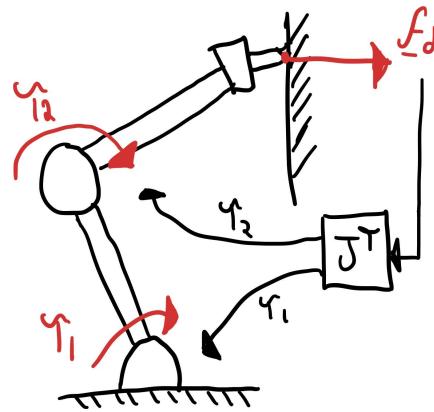


FIGURE 8.6 – Commande en force de l'effecteur d'un robot : interprétation géométrique



Capsule vidéo
Commande en force d'un robot manipulateur
<https://youtu.be/mDQInbWXcj4>

La Figure 8.7 montre cette méthode de commande avec un schéma bloc. À l'exception du calcul du Jacobien qui nécessite de connaître la position des joints actuelle, la méthode peut être considérée comme une boucle ouverte. Par exemple dans le cas des robots à entraînement direct, i.e. utilisant des gros moteurs électrique qui n'ont pas de ratio de réduction, il suffit de traduire la force désirée en couples moteurs à l'aide du Jacobien et ensuite de traduire les couples moteurs en consigne de courant des les moteurs en divisant par la constante des moteurs. Selon le type d'actionneur, des boucles de rétroaction en force à bas niveau peuvent être utilisées, par exemple un asservissement de la pression dans un actionneur pneumatique. La loi de commande présentée ici spécifie les consignes en force pour les actionneurs.

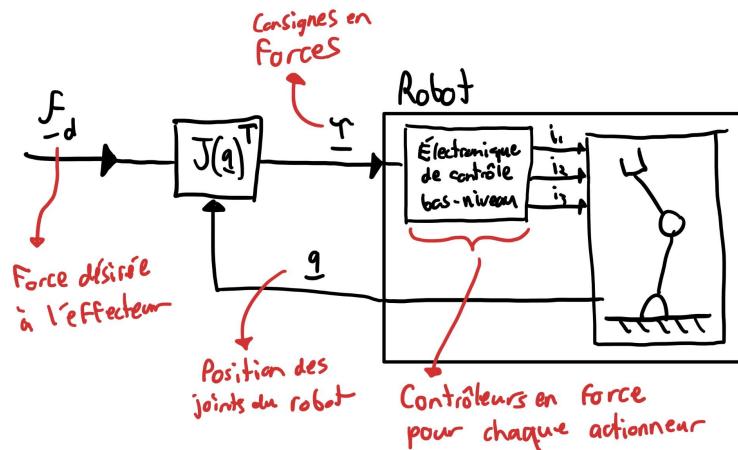
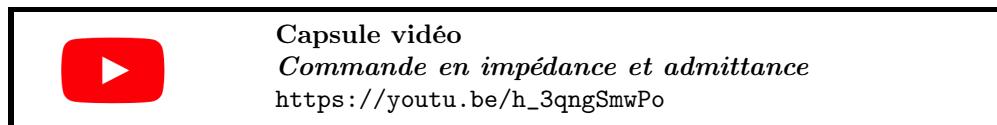


FIGURE 8.7 – Commande en force de l'effecteur d'un robot : schéma bloc

Il est à noter que pour produire une force, le principe action-réaction nous rappelle qu'il faut avoir une résistance. Donc si l'effecteur n'est pas en contact avec un objet le comportement de cette loi de commande de donnera pas nécessairement le résultat attendu, les forces des actionneurs vont alors se balancer avec la résistance interne du robot : forces dissipatrices et inertielles. Le robot va donc accélérer s'il n'y a pas de contact ou de résistance. Comme le mentionne le titre du chapitre, cette loi de commande **quasi-statique** fonctionne bien lorsque le robot est bloqué ou bien bouge très lentement, sinon les forces dissipatives et inertielles internes au robot vont influencer la force de contact.

8.4 Commande en impédance et en admittance

Jusqu'à maintenant, à la section 8.2 une technique a été discutée pour contrôler la position d'un robot et à la section 8.3 une technique a été discutée pour contrôler la force transmise par un robot. La dernière grande catégorie de type de commande est de contrôler la relation entre la position et la force du robot. L'idée est d'imposer une loi de comportement qui relie la force au déplacement du système. Cette approche est particulièrement utile pour les situations où un robot va entrer en contact avec un objet ou un humain. Un des inconvénients d'un contrôle en position pur est que si le robot rencontre un obstacle qui bloque son chemin, le robot va soudainement appliquer une très grande force pour tenter de continuer, ce qui peut risquer de briser soit l'objet rencontré ou le robot lui-même. À l'opposé, le problème avec une approche de contrôle en force pur est que si le robot n'a pas d'obstacle pour créer une résistance et établir la force désirée, le robot va plutôt accélérer jusqu'à ce qu'il atteigne la limite de son espace de travail. La commande en impédance ou en admittance est une approche hybride qui vise plutôt à imposer une relation entre la force et le déplacement. Par exemple, on pourrait désirer que le robot se déplace de façon proportionnelle à une force appliquée sur celui-ci, en imposant ce comportement le robot émulerait le comportement d'un ressort.



Deux approches permettent d'imposer une relation entre la force et le mouvement d'un système : la commande en impédance et en admittance. D'un point de vue théorique les approches sont pratiquement équivalentes, comme illustré à la Figure 8.8, la différence est la causalité. Un contrôleur en impédance mesure le déplacement et impose une force, et un contrôleur en admittance mesure la force pour imposer un déplacement. Dans un monde idéal où les mesures et le contrôle de la force/déplacement seraient parfait, pour la même loi de comportement les deux approches donneraient un résultat identique. Toutefois, en pratique il y a de très grandes différences d'implémentation et de performance entre ces approches.

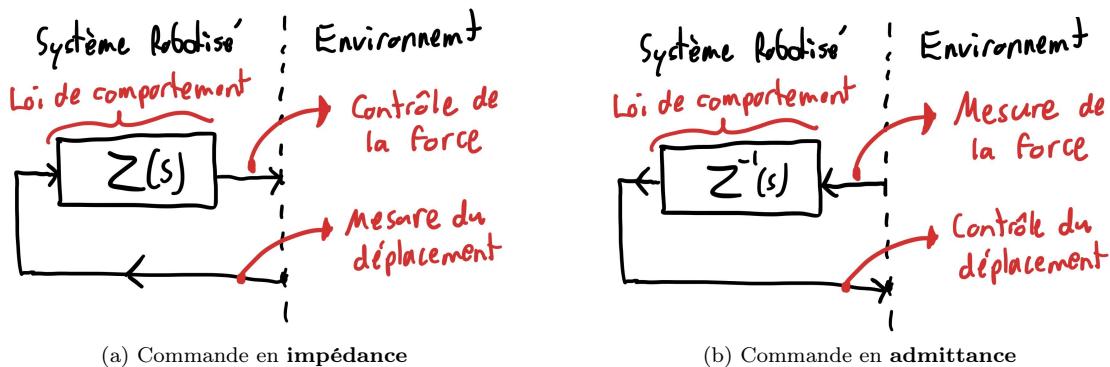


FIGURE 8.8 – Deux façons d'imposer une loi de comportement qui relie la force au déplacement.

Les lois de comportements mécaniques sont souvent décrites avec des fonctions de transfert dans le domaine de Laplace pour une notation compacte. La fonction $Z(s)$ est utilisée pour décrire l'impédance : le signal

de force $f(s)$ divisé par le signal de déplacement $x(s)$ dans le domaine de Laplace. L'admittance $Y(s)$ est l'inverse de la fonction d'impédance.

$$\text{Impédance : } Z(s) = \frac{f(s)}{x(s)} \quad \text{Admittance : } Y(s) = \frac{1}{Z(s)} = \frac{x(s)}{f(s)} \quad (8.24)$$

Note : Dans certains ouvrage le concept d'impédance est définie comme la force divisée par la vitesse (contrairement à la position). La définition relative au déplacement sera toutefois utilisée ici car elle s'intègre plus naturellement au contexte de commande des robots.

Un système mécanique avec une inertie m , un amortissement linéaire b et une rigidité k a une impédance complexe égale à :

$$Z(s) = ms^2 + bs + k \quad (8.25)$$

L'opération de multiplié le signal de déplacement par l'impédance $Z(s)$ est équivalent dans le domaine temporel à :

$$f(s) = Z(s)x(s) = (ms^2 + bs + k)x(s) \iff f(t) = m\frac{d^2}{dt^2}x(t) + b\frac{d}{dt}x(t) + kx(t) \quad (8.26)$$

Exemple Une loi de comportement qui est souvent désirable d'implémenter est la loi de *Hooke*, i.e. un ressort. Comme illustré à la Figure 8.9, l'approche en impédance se résume à mesurer un signal de position x et le multiplier par une constante de rigidité k pour obtenir la force à appliquer. L'approche en admittance se résume à mesurer un signal de force f et le diviser par la constante de rigidité k pour obtenir le déplacement x à imposer. On utilise parfois la notion de compliance $C = 1/k$ qui est l'inverse de la rigidité.

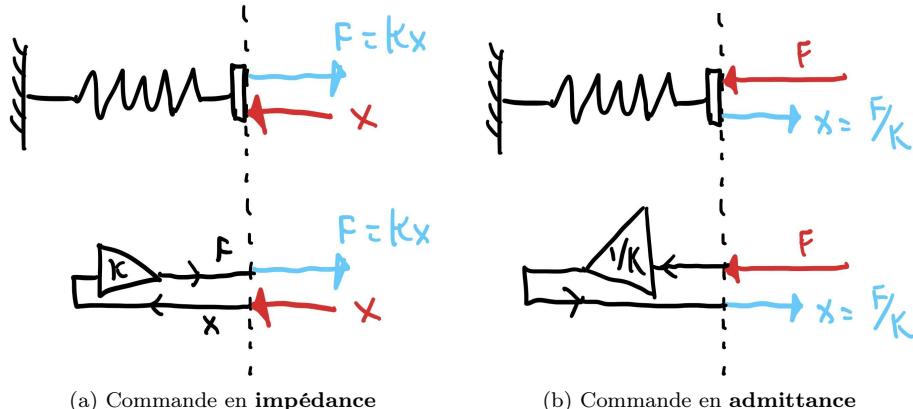


FIGURE 8.9 – Imposition d'une loi de comportement $f = kx$ (ressort)

En pratique, ce qui va guider le choix d'une approche ou d'une autre est principalement le type d'actionneur utilisé dans le système robotisé. L'approche par impédance est plus naturelle pour un système avec des actionneurs qui sont des sources de forces (moteur électrique sans réducteur, cylindre pneumatique, etc.), tandis que l'approche par admittance est plus naturelle pour les systèmes avec des actionneurs qui sont des sources de déplacement, i.e. plus facilement contrôlable en position ou vitesse comme la plupart des moteurs électriques jumelés à des grandes réductions utilisés dans les robots industriels. La Figure 8.10 illustre deux implémentations possibles pour émuler la loi de *Hooke*. Généralement, l'approche en impédance jumelée à des actionneurs à faible résistance hautement réversible (source de force pure) est idéale pour bien émuler des loi de comportement caractérisés par des petites impédances. La performance est toutefois limitée pour émuler des rigidités très élevée. Cette approche est généralement utilisée par les systèmes haptiques et certains systèmes robotiques collaboratif. À l'inverse, l'approche en admittance avec des actionneurs irréversibles (source

de position), est idéale pour émuler de très grandes impédances. La performance est toutefois limitée pour émuler un comportement très compliant (peu de résistance) car l'accélération et la vitesse des actionneurs sont limités en pratique. Cette approche est généralement utilisée par les robots collaboratifs industriels.

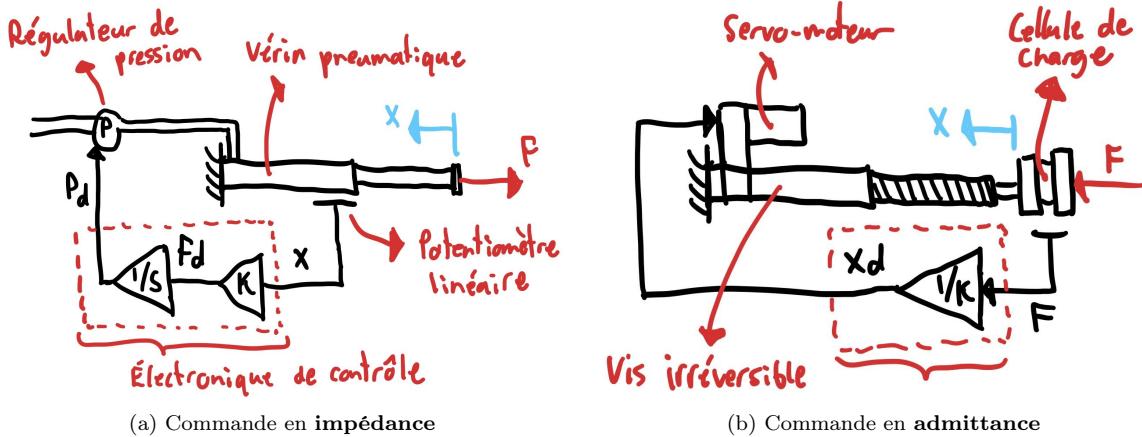


FIGURE 8.10 – Deux exemples d'implémentation de commande pour une loi de comportement $f = kx$ (ressort). L'approche en impédance est adaptée à aux actionneurs qui sont des sources de force avec peu de résistance interne. L'approche en admittance est adaptée aux actionneurs qui sont irréversibles et donc des sources de déplacements peu influencées par la résistance externe.

Les sections suivantes présentent des approches de commande haut-niveau, i.e. la coordination spatiale de plusieurs joints pour obtenir des lois de comportement, soit exprimées dans l'espace des joints ou de la tâche comme illustré à la Figure 8.11. Dans ces sections il est considéré que les actionneurs sont soit des sources parfaites de force ou des sources parfaites de déplacement. La performance dynamique des boucles en forces/impédance/admittance implique une analyse de la dynamique bas-niveau des actionneurs qui est plutôt traité au chapitre 11.

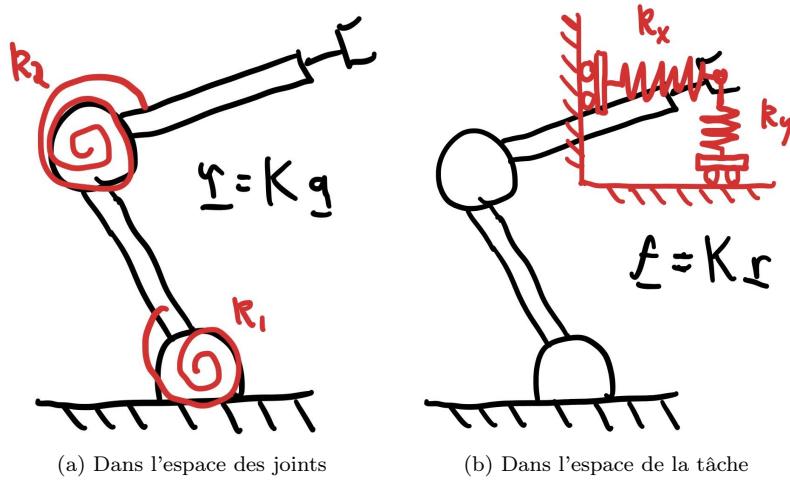


FIGURE 8.11 – Commande de la rigidité/compliance d'un robot

8.4.1 Commande en impédance aux joints

Pour des robots avec des actionneurs contrôlés en force ou couple, la commande en impédance aux joints se résume à relier le déplacement mesuré aux joints aux efforts à chaque joint. Dans le cas le plus simple, comme illustré à la Figure 8.11a, on peut commander des couples moteurs proportionnel aux déplacement

angulaire des joints, ce qui est équivalent à émuler des ressorts angulaires sur chaque joint. De façon général on pourrait émuler des ressorts-amortisseurs sur chaque joint avec la loi de commande :

$$\underline{\tau} = K\underline{q}_e + B\underline{\dot{q}}_e \quad (8.27)$$

ou $\underline{q}_e = \underline{q}_d - \underline{q}$ est une erreur par rapport à la configuration de référence ou les ressorts sont "au repos". Les matrices K et B sont respectivement une matrice de coefficients de rigidité et une matrice de coefficients d'amortissement. Les matrices sont diagonales pour un comportement de chaque joint indépendant. Dans ce cas diagonal, la loi de commande est équivalente à plusieurs boucles d'asservissement indépendantes de type Proportionnel-Dérivé sur la position des joints. La loi de commande est illustrée sous forme de schéma bloc à la Figure 8.12. Comme ici on programme des lois de comportement, il est possible de programmer

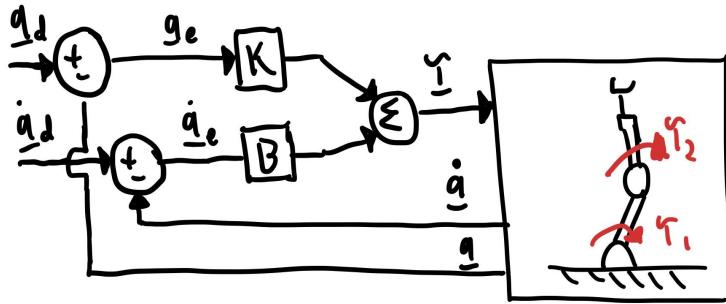


FIGURE 8.12 – Commande de l'impédance au joints d'un robot : schéma bloc

des impédances non-linéaires arbitraires. Par exemple, pour les systèmes haptiques un problème courant est d'émuler un obstacle rigide, ce qui est appelé mur virtuel dans la littérature. L'impédance cible dans ce cas est discontinue, aucune résistance pour une certaine plage de déplacement et une très grande rigidité lorsque la position atteint la position du mur virtuel.

Compensation de friction : Les systèmes robotiques vont normalement avoir de la dissipation naturelle présente dans les joints (roulements, engrenage, etc.). Si la friction naturelle est significative et plus grande que l'amortissement qu'on désire émuler, les coefficients de friction dans B pourraient être négatifs pour appliquer une force opposée à la friction naturelle. La matrice B pourrait être définie par :

$$B = B_d - B_{sys} \quad (8.28)$$

où B_d contient les coefficients d'amortissement que l'on désire émuler et B_{sys} contient les coefficients d'amortissement naturel du robot. Il est toutefois risqué en pratique de tenter de totalement compenser la friction dans les joints d'un robot, car si la compensation est légèrement plus forte que la friction naturelle le système se retrouve instable. De plus le comportement de la friction autour de zéro vitesse est généralement très non-linéaire et avec de l'hystérésis, donc dur à modéliser avec précision.

Note sur l'émulation de l'inertie : On pourrait être tenté de rajouter un terme de type $M\ddot{q}_e$ à l'équation (8.27) pour rajouter l'option d'émuler un effet inertiel sur chaque joint. Toutefois il y a un problème de causalité à une telle opération. L'accélération n'est pas un état du système mais résulte de l'application de forces sur le système incluant τ . Donc les couples dépendraient de l'accélération mesurée, qui dépend des couples appliquées par les actionneurs, qui dépendraient de l'accélération mesurée, qui dépend ... etc. On se retrouve avec une boucle algébrique analogue au paradoxe de la poule et de l'oeuf (lequel vient en premier ?). En pratique, l'accélération mesurée serait celle d'un instant passé dû au délai de calcul dans le contrôleur et aux filtres dans les capteurs, et la rétroaction de la mesure de l'accélération risque de déstabiliser le système.

8.4.2 Commande en impédance de l'effecteur

Pour la commande en impédance de l'effecteur d'un robot, c'est la relation force-déplacement à l'effecteur qu'on impose au robot :

$$\underline{f}_e = K\underline{r}_e + B\dot{\underline{r}}_e \quad (8.29)$$

ou $\underline{r}_e = \underline{r}_d - \underline{r}$ est le vecteur position qui de la position désirée de l'effecteur du robot par rapport à la position actuelle. On peut interpréter l'effet de la loi de commande comme un ressort-amortisseur virtuel qui relit l'effecteur du robot à sa position cible comme illustré à la Figure 8.13. Le ressort-amortisseur virtuel

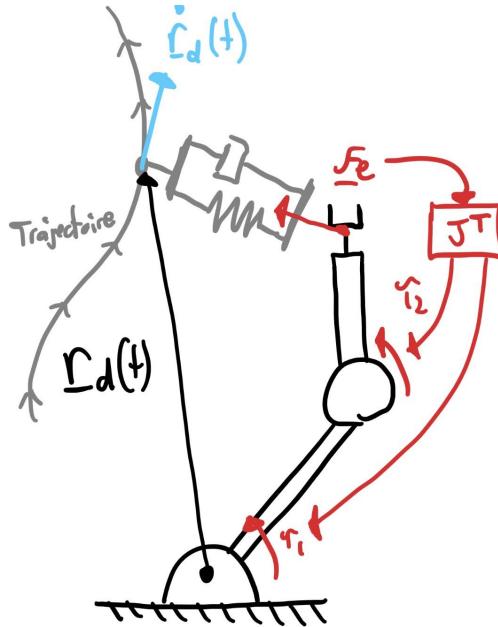


FIGURE 8.13 – Commande de l'impédance de l'effecteur d'un robot : interprétation géométrique

produit une force virtuelle \underline{f}_e désirée à l'effecteur qui est traduite en commande de couple grâce à la relation statique d'un manipulateur :

$$\underline{\tau} = J(\underline{q})^T \underline{f}_e \quad (8.30)$$



Capsule vidéo

Commande en impédance d'un robot manipulateur

<https://youtu.be/R12kDwsYS9E>

La Figure 8.14 illustre cette loi de commande avec un schéma bloc. La première étape du contrôleur est de calculer la position et la vitesse de l'effecteur à partir des capteurs qui mesurent la position et la vitesse des joints. Il faut donc ici utiliser la cinématique directe et la cinématique-différentielle directe pour calculer la position et la vitesse de l'effecteur. Ensuite la position et la vitesse de l'effecteur sont comparées à la position et la vitesse désirée (qui peut varier dans le temps, i.e. une trajectoire). L'erreur en termes de position et vitesse est convertie en force cartésienne basé sur la loi de comportement désirée (l'impédance à l'effecteur). Finalement cette force cartésienne est convertie en couple aux joints grâce au Jacobien du robot manipulateur. On peut aussi ajouter une compensation de gravité, voir la section 8.4.4. L'équation de la loi

de commande est égale à :

$$\underline{\tau} = J(\underline{q})^T \underline{f}_e \quad (8.31)$$

$$\underline{\tau} = J(\underline{q})^T [K \underline{r}_e + B \dot{\underline{r}}_e] \quad (8.32)$$

$$\underline{\tau} = J(\underline{q})^T [K (\underline{r}_d - \underline{r}) + B (\dot{\underline{r}}_d - \dot{\underline{r}})] \quad (8.33)$$

$$\underline{\tau} = J(\underline{q})^T [K (\underline{r}_d - f(\underline{q})) + B (\dot{\underline{r}}_d - J(\underline{q})\dot{\underline{q}})] \quad (8.34)$$

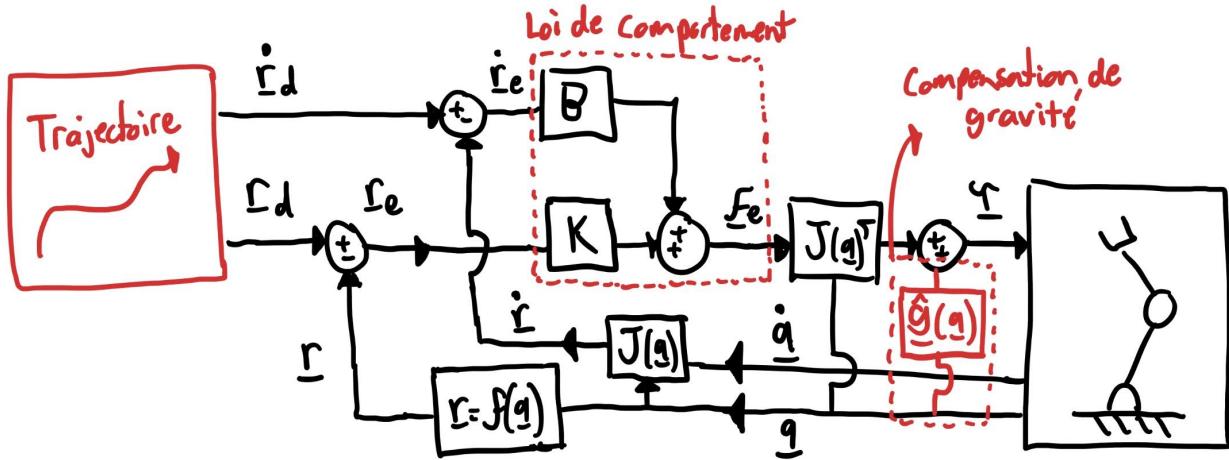


FIGURE 8.14 – Commande de l’impédance de l’effecteur d’un robot : schéma bloc



Exercice de code

Simulation d'un robot planaire contrôlé en impédance

<https://colab.research.google.com/drive/1EM3hNEwz2aiBqx9GDHoM7QGuPF-Afv1Q?usp=sharing>

8.4.3 Équivalence entre une impédance définie aux joints vs. définie à l'effecteur

Il y a une certaine équivalence locale entre une loi de commande en impédance à l'effecteur et une loi de commande en impédance dans l'espace des joints. Pour un cas simplifié où la cible \underline{q}_d et \underline{r}_d sont égales à zéro, les lois de commandes sont données par :

$$\text{Joints : } \underline{\tau} = -K_q \underline{q} - B_q \dot{\underline{q}} \quad \text{Effecteur : } \underline{\tau} = -J(\underline{q})^T [K_e \underline{r} + B_e \dot{\underline{r}}] \quad (8.35)$$

où on a rajouté les indices q ou e aux matrices pour spécifier l'espace, joint ou effecteur, dans lesquelles elles sont définies. Ensuite si on distribue et utilise la relation de cinématique différentielle, on observe que la matrice d'amortissement B_e à l'effecteur a exactement le même effet qu'une matrice d'amortissement dans l'espace des joints égale à $B_q = J^T B_e J$:

$$\underline{\tau} = -J(\underline{q})^T K_e \underline{r} - J(\underline{q})^T B_e \dot{\underline{r}} \quad (8.36)$$

$$\underline{\tau} = -J(\underline{q})^T K_e \underline{r} - \underbrace{[J(\underline{q})^T B_e J(\underline{q})] \dot{\underline{q}}}_{B_q} \quad (8.37)$$

Pour des petites variations autour de la position d'équilibre des ressorts virtuels (celle pour laquelle la force nette est nulle), on peut aussi retrouver une équivalence similaire pour la rigidité $K_q = J^T K_e J$:

$$\delta \underline{\tau} = -\underbrace{[J(\underline{q})^T K_e J(\underline{q})] \delta \underline{q}}_{K_q} - \underbrace{[J(\underline{q})^T B_e J(\underline{q})] \delta \dot{\underline{q}}}_{B_q} \quad (8.38)$$

Il est à bien noter que ces équivalences sont locales seulement due aux effets non-linéaires.

8.4.4 Convergence des lois de commande en impédance

Pour les robots avec des actionneurs contrôlés en force, une bonne façon de contrôler la position de l'effecteur est d'utiliser l'approche en impédance à l'effecteur avec une matrice K très rigide. C'est l'équivalent de prendre un gros ressort très rigide, attacher une extrémité sur la position cible et l'autre sur l'effecteur du robot. Le robot va être attiré vers la position cible. D'un point de vu énergétique, le robot va converger vers le point où l'énergie potentielle du système est minimum, ce qui est la position cible si le ressort est la seule source d'énergie potentielle. Pour que le point de convergence soit stable il faut aussi qu'il y ait une certaine forme de dissipation dans le système (qui peut venir des coefficients du ressort virtuel dans B ou bien de phénomènes physique dans les joints du robot) sinon le robot oscillerait autour du point d'énergie potentiel minimum.

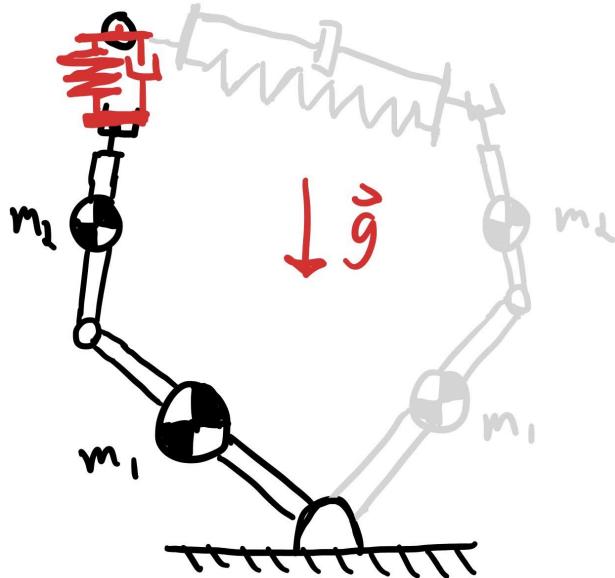


FIGURE 8.15 – Convergence d'un contrôleur en impédance à l'effecteur

Comme illustré à la Figure 8.15, si considère d'abord le robot comme seulement une chaîne de corps rigide avec leurs propriétés inertielles, on peut bien visualiser qu'un ressort à l'effecteur va attirer le robot sur la cible. Les propriétés inertielles du bras et la géométrie non-linéaire vont influencer la trajectoire, mais dans ce cas le point d'équilibre dépend seulement des forces conservatrices. Si le ressort virtuel de la loi d'impédance est le seul élément qui produit une force conservatrice, alors le point d'équilibre sera la configuration où l'effecteur est exactement sur la cible. Si le robot est influencé par des forces gravitationnelles, le point d'équilibre sera le point minimum de l'énergie potentielle totale. Le robot aura donc une erreur finale avec une amplitude qui dépend de la force relative du ressort virtuel et de la gravité. Comme illustré à la Figure 8.14, il est toutefois possible d'inclure une **compensation de gravité** \hat{g} à une loi de commande en impédance pour éliminer cette erreur finale :

$$\tau = J(q)^T [K r_e + B \dot{r}_e] + \hat{g}(q) \quad (8.39)$$

où \hat{g} est l'estimation par le contrôleur (basé sur un modèle) du vecteur des forces gravitationnelles dans l'espace des joints. Finalement, un autre phénomène courant qui pourrait causer une erreur final serait s'il y a de la friction sèche dans les joints du robot. Le robot pourrait rester pris à un point où la force commandée est égale à la friction sèche dans le système.

Cette analyse qualitative basée sur des principes énergétiques peut être formalisé avec une analyse de la stabilité avec la méthode de *Lyapunov*. Si on considère les équations générées de la dynamique du robot

manipulateur avec des actionneurs contrôlés en forces :

$$\underbrace{H\ddot{q} + C\dot{q}}_{\text{Forces inertielles}} + \underbrace{\underline{d}}_{\text{Forces dissipatives}} + \underbrace{\underline{g}}_{\text{Forces conservatrices}} = \underline{\tau} \quad (8.40)$$

avec comme loi de commande, une impédance à l'effecteur incluant une compensation de gravité :

$$\underline{\tau} = J^T [K\underline{r}_e + B\underline{r}\dot{e}] + \hat{g} \quad (8.41)$$

En utilisant comme fonction candidate de *Lyapunov* l'énergie mécanique réelle plus l'énergie potentielle virtuelle dans le ressort du contrôleur :

$$V = \underbrace{1/2 \dot{q}^T H \dot{q}}_{\text{Énergie cinétique réelle du robot}} + \underbrace{1/2 \underline{r}_e^T K \underline{r}_e}_{\text{Énergie potentielle virtuelle du ressort}} \quad (8.42)$$

Il est possible de démontré que cette fonction est toujours décroissante dans le temps sous certaines conditions :

$$\dot{V} = \frac{d}{dt} [1/2 \dot{q}^T H \dot{q} + 1/2 \underline{r}_e^T K \underline{r}_e] \quad (8.43)$$

$$\dot{V} = \dot{q}^T [H\ddot{q} + 2\dot{H}\dot{q}] + \underline{r}_e^T K \dot{r}_e \quad (8.44)$$

$$\dot{V} = \dot{q}^T [\underline{\tau} - \underline{g} - \underline{d}] - \dot{r}_e^T K \dot{r}_e \quad \text{pour une position désirée fixe : } \dot{r}_e = \dot{r}_d - \dot{r} = -\dot{r} \quad (8.45)$$

$$\dot{V} = \dot{q}^T [\underline{\tau} - \underline{g} - \underline{d}] - \dot{q}^T J^T K \underline{r}_e \quad (8.46)$$

$$\dot{V} = \dot{q}^T [\underline{\tau} - \underline{g} - \underline{d}] - \dot{q}^T J^T K \underline{r}_e \quad (8.47)$$

$$\dot{V} = \dot{q}^T [\underline{\tau} - \underline{g} - \underline{d} - J^T K \underline{r}_e] \quad (8.48)$$

$$\dot{V} = \dot{q}^T [J^T [K \underline{r}_e + B \dot{r}_e] + \hat{g} - \underline{g} - \underline{d} - J^T K \underline{r}_e] \quad (8.49)$$

$$\dot{V} = \dot{q}^T [J^T K \underline{r}_e - J^T B \dot{r}_e + \hat{g} - \underline{g} - \underline{d} - J^T K \underline{r}_e] \quad (8.50)$$

$$\dot{V} = \dot{q}^T [-J^T B J \dot{q} + \hat{g} - \underline{g} - \underline{d}] \quad (8.51)$$

$$\dot{V} = \dot{q}^T [-J^T B J \dot{q} - \underline{d}] \quad \text{si la compensation de gravité est parfaite} \quad (8.52)$$

$$\dot{V} = \dot{q}^T [-J^T B J \dot{q} - D \dot{q}] \quad \text{si on considère la friction dans les joints du robot comme linéaire} \quad (8.53)$$

$$\dot{V} = -\dot{q}^T [J^T B J + D] \dot{q} \quad (8.54)$$

$$\dot{V} < 0 \quad \forall \dot{q} \neq 0 \quad \text{si } [J^T B J + D] > 0 \quad \Rightarrow \quad \lim_{t \rightarrow \infty} \underline{r}_e(t) = \underline{0} \quad \& \quad \lim_{t \rightarrow \infty} \dot{q}(t) = \underline{0} \quad (8.55)$$

Ce qui démontre que le système va nécessairement converger vers le point minimum de la fonction V , ce qui correspond à $\dot{q} = \underline{0}$ et $\underline{r}_e = \underline{0}$, avec comme condition que les termes d'amortissement total du système (contrôleur + friction naturelle) sont positifs. Cette analyse peut aussi être faites pour une impédance dans le domaine des joints, et une combinaison des deux, pour obtenir des conclusions équivalentes.

8.5 Commande en admittance

Pour un contrôle en admittance, dans l'espace des joints ou de la tâche, des capteurs mesurent des forces et le déplacement du robot est contrôlé à bas niveau, soit en position ou en vitesse. Si les actionneurs sont asservis en position, la relation générique est :

$$q_i = \frac{1}{Z_i(s)} \tau_i \quad (8.56)$$

où q_i est une consigne de position pour le joint i , Z_i est une impédance désirée pour le joint i et τ_i est la force mesurée au joint i . Si les actionneurs sont asservis en vitesse à bas niveau, la relation générique est :

$$\dot{q}_i = \frac{s}{Z_i(s)} \tau_i \quad (8.57)$$

Lorsqu'on implémente une loi de commande il est préférable d'éviter de dériver des signaux, donc certaines combinaison de type de consigne bas-niveau (position vs. vitesse) avec un type d'impédance sont préférable. Par exemple, pour émuler une force proportionnelle à une vitesse $Z_i(s) = b_i s$ avec des actionneurs contrôlés en vitesse, l'équation (8.57) se réduit à

$$\dot{q}_i = \frac{1}{b} \tau_i \quad (8.58)$$

donc simplement une relation algébrique entre les signaux. Avec des actionneurs contrôlés en position l'équation (8.56) se réduit à

$$q_i = \frac{1}{bs} \tau_i = \frac{1}{b} \int \tau_i \quad (8.59)$$

ce qui implique d'intégrer le signal de force. Par contre, si on désire émuler un système masse-ressort-amortisseur c'est l'approche avec des actionneurs en position qui est préférable. Avec des actionneurs contrôlés en position l'équation (8.56) se réduit dans ce cas à

$$\dot{q}_i = \frac{1}{ms^2 + bs + k} \tau_i \quad (8.60)$$

La fonction de transfert a deux pôles donc l'implémentation inclurait deux intégrales. Avec des actionneurs contrôlés en vitesse l'équation (8.57) se réduit dans ce cas à

$$\dot{q}_i = \frac{s}{ms^2 + bs + k} \tau_i \quad (8.61)$$

une fonction de transfert qui a en plus un zéro, donc demanderait de dériver un signal en plus des deux intégrales. Pour l'implémentation d'une loi de commande, l'idéal lorsque possible est une relation algébrique. Si on doit avoir une relation différentielle il est préférable d'avoir seulement des intégrales. Les opérations de dérivations sont à éviter.

8.5.1 Commande en admittance aux joints

Pour un contrôle en admittance dans l'espace des joints, des capteurs de forces vont mesurer la force actuelle à chaque joint et commander un déplacement à l'actionneur relié. Les lois en admittance peuvent aussi être exprimées en format matricielle pour décrire le comportement de tous les joints en une seule équation. La Figure 8.16 présente un contrôleur en admittance de type amortisseurs avec comme comportement cible la relation :

$$B\dot{q} = \underline{\tau} \quad (8.62)$$

pour un robot contrôlé en vitesse à bas niveau dans l'espace des joints, avec des capteurs qui mesure le couple de chaque joint.

La Figure 8.17 présente le schéma bloc d'un contrôleur en admittance avec comme comportement cible la relation :

$$M\ddot{q} + B\dot{q} + Kq = \underline{\tau} \quad (8.63)$$

pour un robot contrôlé en position à bas niveau dans l'espace des joints, avec des capteurs qui mesure le couple de chaque joint. Pour ce cas il y a deux intégrale dans la loi de commande. On peut interpréter le bloc "loi de comportement" à la Figure 8.17 qui est tout simplement la simulation d'un système masse-ressort-amortisseur qui reçoit en entrée une force réelle lue par un capteur. Le résultat de la simulation est ensuite envoyé à un contrôleur en déplacement. Le robot va donc émuler le système virtuel simulé.

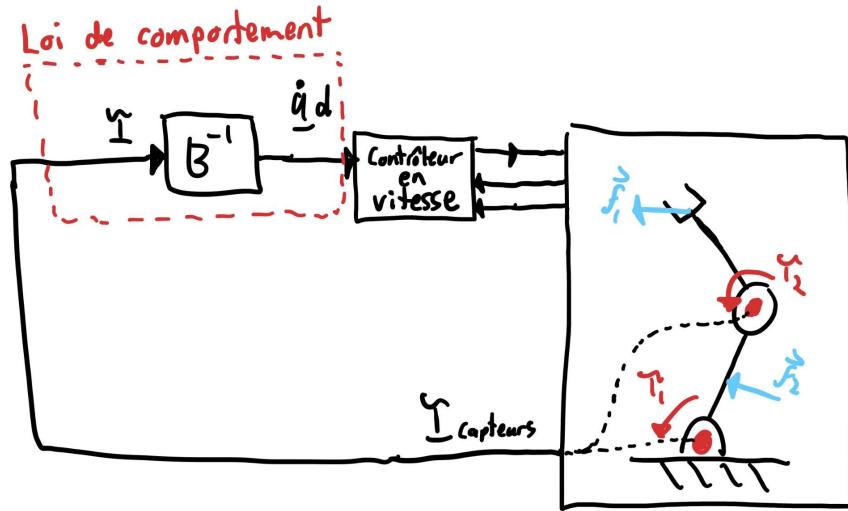


FIGURE 8.16 – Commande de l'admittance au joints d'un robot : schéma bloc

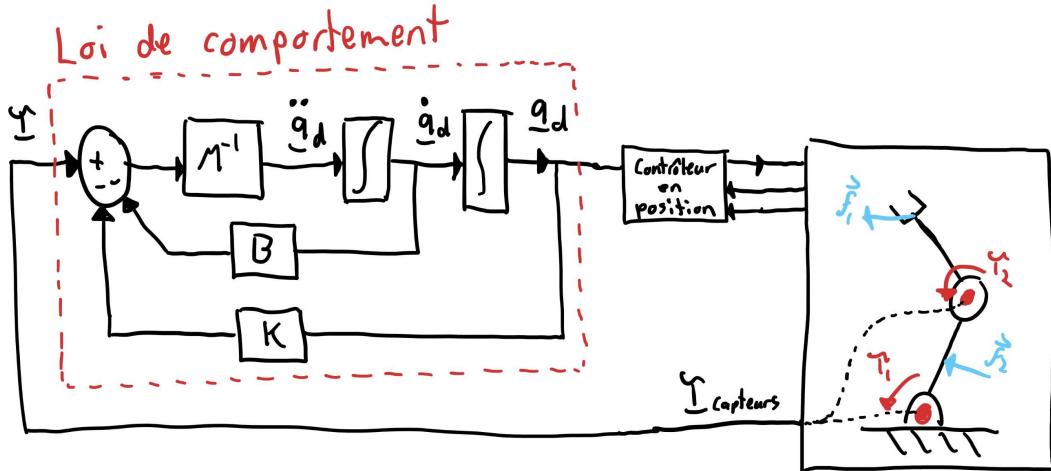


FIGURE 8.17 – Commande de l'admittance au joints d'un robot : schéma bloc

**Capsule vidéo***Commande en admittance d'un robot manipulateur*<https://youtu.be/SP5bISWmkT0>

8.5.2 Commande en admittance de l'effecteur

L'approche en admittance dans l'espace de l'effecteur est basée sur une mesure de la force à l'effecteur (généralement avec une cellule de charge) et le contrôle du déplacement de l'effecteur à bas niveau. La Figure 8.18 présente un schéma bloc d'une loi de commande en admittance pour imposer un comportement :

$$B\dot{r} = \underline{f}_e \quad (8.64)$$

Une chose à noter est que généralement la cellule de charge va mesurer les composantes du vecteur force \underline{f}_e dans une base vectorielle mobile t attachée à l'outil. Il faut donc faire un changement de base avec la

matrice de rotation ${}^w R^t$ pour avoir les composantes dans la base vectorielle fixe w dans laquelle le reste des autres vecteurs sont exprimés. La matrice de rotation ${}^w R^t$ doit donc être obtenue basé sur la position des joints actuels en utilisant la relation de cinématique directe pour l'orientation. La loi de commande de la Figure 8.18 est utilisée pour le mode démonstration (*teach mode*) des robots manipulateur, i.e. le mode où un opérateur peut déplacer le robot manuellement dans l'espace. Avec cette loi de commande, un humain peut pousser sur l'effecteur du robot et le robot se déplacera alors dans le sens de la poussée avec une vitesse proportionnelle à l'amplitude de la poussée. Si il n'y a aucune poussée le robot reste en place.

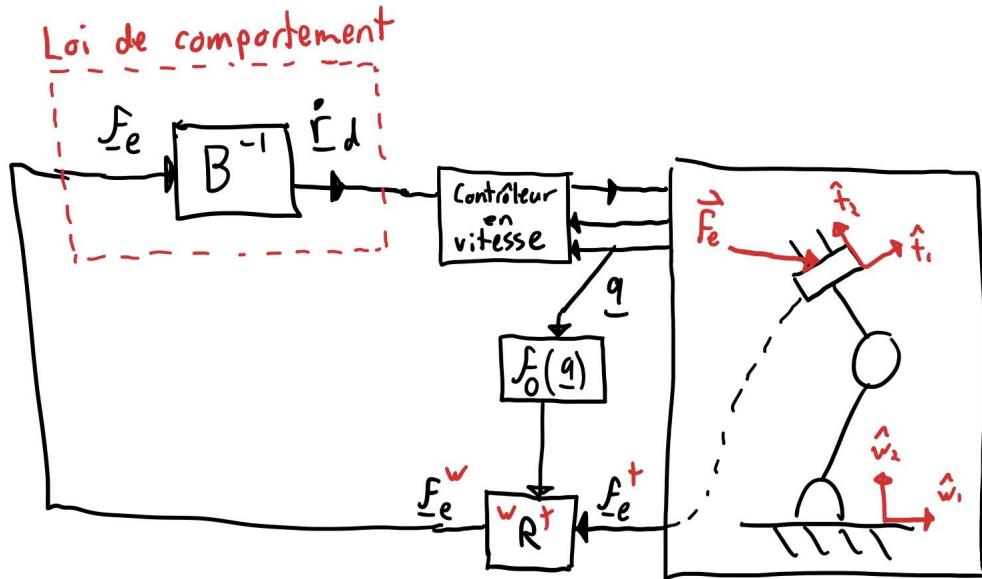


FIGURE 8.18 – Commande de l'admittance à l'effecteur d'un robot avec une loi de comportement de type amortisseur : schéma bloc

La Figure 8.18 montre le schéma bloc d'une loi de commande plus générique pour une loi de comportement égale à une masse-ressort-amortisseur à l'effecteur :

$$M\ddot{\underline{r}} + B\dot{\underline{r}} + K\underline{r} = \underline{f}_e \quad (8.65)$$

Il est à noté que les Figures 8.18 et 8.19 ne montre pas le détail de la boucle interne du contrôleur en vitesse ou position de l'effecteur. Les méthodes présentées aux sections 8.1 et 8.2 peuvent être utilisée. Une différence à noter pour l'approche en admittance basée sur l'utilisation d'une cellule de charge à l'effecteur du robot, par rapport aux autres approches présentées est que seul les forces qui sont appliquées à l'effecteur (en aval de la cellule de charge) vont influencer la position du robot. Si une force est appliquée sur les liens du robot en amont de la cellule de charge, le contrôleur n'aura pas connaissance de cette force externe qui ne sera donc pas pris en compte dans le calcul du déplacement du robot basée sur la loi de comportement désirée.

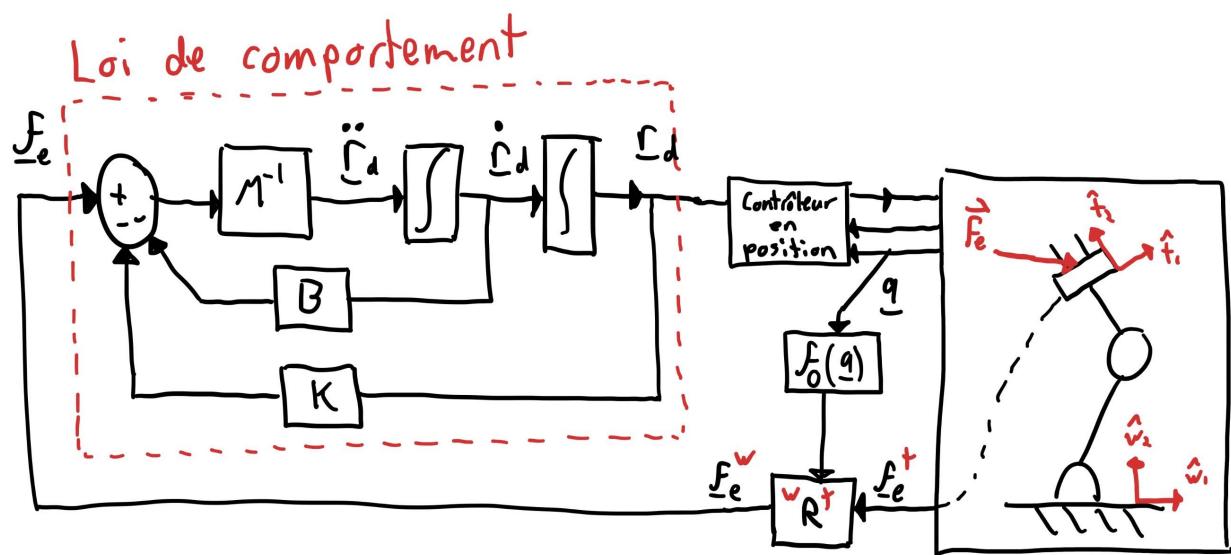


FIGURE 8.19 – Commande de l'admittance à l'effecteur d'un robot avec une loi de comportement de type masse-ressort-amortisseur : schéma bloc

Chapitre 9

Commande des robots manipulateurs II : dynamique

Ce chapitre présente des techniques de commande qui s'applique pour les systèmes robotisés où les entrées du système sont des forces ou couples produites par les actionneurs. Dans ce contexte, le système peut être modéliser comme un système d'équations différentielles non-linéaire d'ordre deux et être représentés par l'équation des manipulateurs décrite au chapitre 6. Les vidéos suivants présentent une introduction, avec un point de vue graphique dans l'espace des phases, aux méthodes pour se contexte :



Capsule vidéo
Introduction à la commande des robots
https://youtu.be/eL6i319X_w4



Capsule vidéo
L'espace des phases
<https://youtu.be/QGPVxZrZq44>



Capsule vidéo
Tour d'horizon des méthodes de commande dans l'espace des phases
<https://youtu.be/wiY-Wjj9-zs>

Au lien suivant, une amorce de code pour tester des lois de commande sur un robot manipulateur à 2 DDL :



Exercice de code
Testez votre loi de commande pour un manipulateur
https://colab.research.google.com/drive/1bnJ9v5kHRFFhnNOZx-_Kcht215sTOHxr?usp=sharing

9.1 Commande dé-localisée joint par joint

Dans certains situations, il est possible de traiter le problème de commande joint par joint avec des méthodes de commande classique comme des lois de commandes de type PID. Cette approche peut bien fonctionner surtout lorsque le système robotique a des actionneurs avec des grands ratios de réduction, ce qui minimise le couplage dynamique entre les divers DDL. Plus les effets inertIELS et plus on désire suivre précisément des trajectoires dynamiques, moins ce type d'approche va être performante.

Si on utilise des lois de commandes indépendantes de type PID pour chaque joint, mais avec le terme intégral à zéro, alors l'approche de commande est équivalente à définir une matrice d'impédance diagonale dans le domaine des joints comme discuté à la section 8.4.1. Une telle approche va être stable mais avec comme principal inconvénient une erreur finale non-nulle, typiquement due à la gravité et/ou de la friction sèche. On a vu qu'il était possible d'ajouter un terme de compensation de gravité mais cela implique de centraliser les calculs et un bon modèle du système. Une alternative intéressante est d'utiliser l'effet intégrateur des PIDs sur chaque joint pour annuler les erreurs finales en position pour chaque joint du à ces effets. L'effet intégrateur va ajouter toutefois une dynamique supplémentaire qui peut créer des oscillations et déstabiliser le système. Il y a donc un compromis entre avec des grands gains intégrateurs pour réduire l'erreur finale rapidement vs. minimiser les oscillations et le risque d'instabilité. L'exercice suivant permet d'explorer ce compromis :



Exercice de code
Simulation d'un manipulateur avec des PIDs
<https://colab.research.google.com/drive/1qaCNY2ohQIbC6dV2ZW4KY6FDM9NhJzH2?usp=sharing>

Il serait possible de faire une analyse de stabilité locale en linéarisant la dynamique d'un robot autour du point d'opération et de faire le choix des gains PIDs avec les outils d'asservissement classique. L'analyse serait toutefois limité à décrire le comportement du robot très proche de ce point d'opération. Si on utilise le robot sur une grande plage d'opération qui implique des grande non-linéarités, alors il est beaucoup plus difficile de garantir la stabilité du système et aussi tout simplement d'avoir des gains de PIDs qui vont bien fonctionner pour une grande plage de conditions initiales et positions cibles.

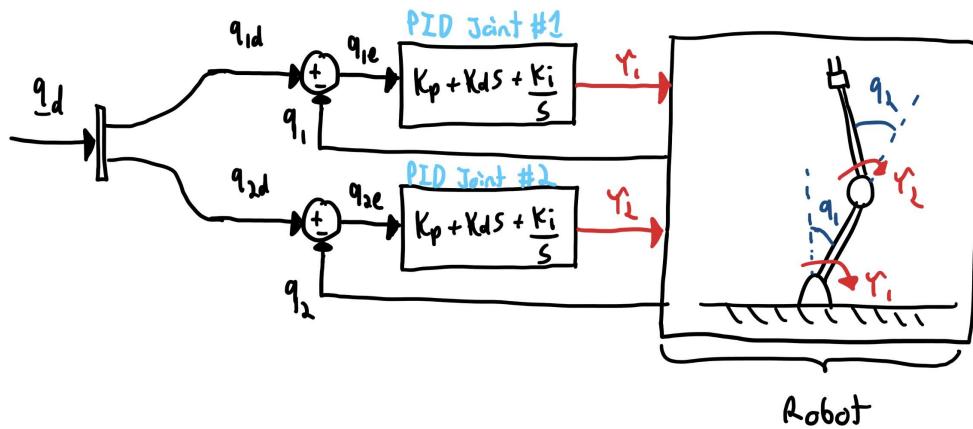
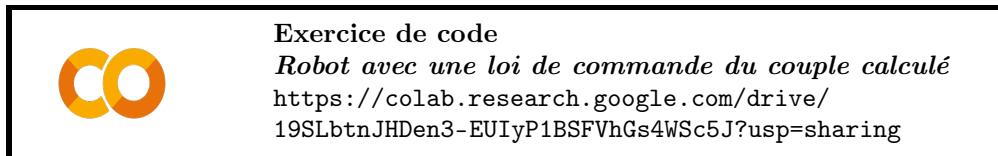
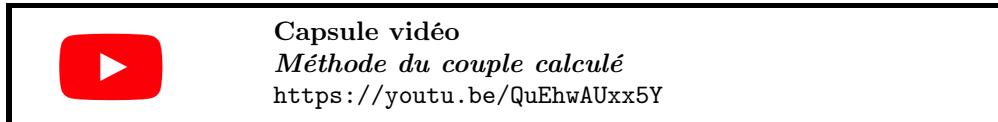


FIGURE 9.1 – Schéma bloc d'un robot avec de PIDs délocalisés

Un avantage non-négligeable pratique lors de l'implémentation d'une telle approche délocalisée, est que les boucles d'asservissements de chaque joint peuvent être implémentées directement dans l'électronique de puissance des moteurs puisque les calculs sont indépendants. Typiquement, l'électronique de puissance permet d'avoir des délais de boucle très court, dans l'ordre des dizaines de kHz, alors un ordinateur qui centralise l'information de tous les joints implique généralement plus de délais et des boucles moins rapides.

9.2 Commande avec la méthode du couple calculé

La méthode du couple calculé consiste à utiliser les équations d'un modèle dynamique d'un système robotique pour déterminer les forces à appliquer pour obtenir une accélération cible. Il est ensuite possible de calculer cette accélération cible de sorte à converger vers la position ou trajectoire désirée.



9.2.1 Commande de l'accélération des joints

Premièrement, si un système est pleinement actionné, il est possible d'imposer d'une accélération arbitraire dans l'espace des joints à un instant donné. Pour un système décrit par l'équation des manipulateurs :

$$H(\underline{q})\ddot{\underline{q}} + C(\underline{q}, \dot{\underline{q}})\dot{\underline{q}} + d(\underline{q}, \dot{\underline{q}}) + g(\underline{q}) = B(\underline{q})\underline{u} \quad (9.1)$$

si on définit une loi de commande avec une accélération référence $\ddot{\underline{q}}_r$:

$$\underline{u} = B(\underline{q})^{-1} \left[H(\underline{q})\ddot{\underline{q}}_r + C(\underline{q}, \dot{\underline{q}})\dot{\underline{q}} + d(\underline{q}, \dot{\underline{q}}) + g(\underline{q}) \right] \quad (9.2)$$

qui assume ici que la matrice B est inversible, donc que le système est pleinement actionné. Alors si on substitut la loi de commande dans l'équation de la dynamique on obtient :

$$H(\underline{q})\ddot{\underline{q}} + C(\underline{q}, \dot{\underline{q}})\dot{\underline{q}} + d(\underline{q}, \dot{\underline{q}}) + g(\underline{q}) = B(\underline{q})B(\underline{q})^{-1} \left[H(\underline{q})\ddot{\underline{q}}_r + C(\underline{q}, \dot{\underline{q}})\dot{\underline{q}} + d(\underline{q}, \dot{\underline{q}}) + g(\underline{q}) \right] \quad (9.3)$$

$$H(\underline{q})\ddot{\underline{q}} = H(\underline{q})\ddot{\underline{q}}_r \quad (9.4)$$

$$\ddot{\underline{q}} = \ddot{\underline{q}}_r \quad (9.5)$$

on obtient comme résultat que l'accélération du système peut être directement commandée. Comme illustré à la figure 9.2, avec cette loi de commande, la relation entre la référence $\ddot{\underline{q}}_r$ et la variable à contrôler \underline{q} est réduite à une relation linéaire, ici un double intégrateur. On réfère aussi parfois à cette méthode comme une boucle linéarisante.

9.2.2 Suivi de trajectoire avec le couple calculé

Lorsqu'on désire qu'un robot suivre une certaine trajectoire, il est possible de définir la référence d'accélération $\ddot{\underline{q}}_d$ comme une fonction de la trajectoire cible pour obtenir que le robot va converger sur la trajectoire. Si une trajectoire cible est définie comme une fonction du temps pour les coordonnées \underline{q} et leur deux premières dérivées :

$$\text{Trajectoire désirée : } \ddot{\underline{q}}_d(t) \quad \dot{\underline{q}}_d(t) \quad \underline{q}_d(t) \quad (9.6)$$

alors si on définit la référence d'accélération par la loi de commande suivante :

$$\ddot{\underline{q}}_r = \ddot{\underline{q}}_d + 2\zeta w \underbrace{(\dot{\underline{q}}_d - \dot{\underline{q}})}_{\dot{\underline{q}}_e} + w^2 \underbrace{(\underline{q}_d - \underline{q})}_{\underline{q}_e} \quad (9.7)$$

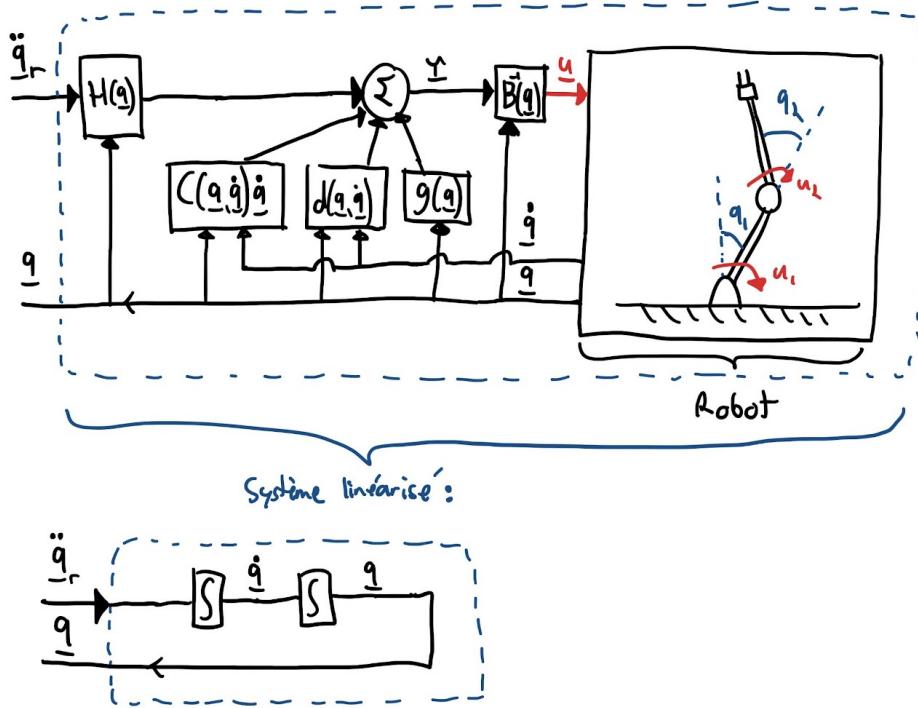


FIGURE 9.2 – Couple calculé et boucle linéarisante.

puisque'on obtient $\ddot{\underline{q}} = \ddot{\underline{q}}_r$ avec la méthode du couple calculé, en combinant cette définition pour $\ddot{\underline{q}}_r$ on va obtenir :

$$\underbrace{(\ddot{\underline{q}}_d - \ddot{\underline{q}})}_{\ddot{\underline{q}}_e} + 2\zeta w \underbrace{(\dot{\underline{q}}_d - \dot{\underline{q}})}_{\dot{\underline{q}}_e} + w^2 \underbrace{(\underline{q}_d - \underline{q})}_{\underline{q}_e} = 0 \quad (9.8)$$

ce qui représente une équation de la dynamique pour l'erreur d'ordre 2 qui converge exponentiellement vers zéro si on choisi $w^2 > 0$ et $2\zeta w > 0$, ce qui implique que la trajectoire du robot va converger sur la trajectoire désirée.

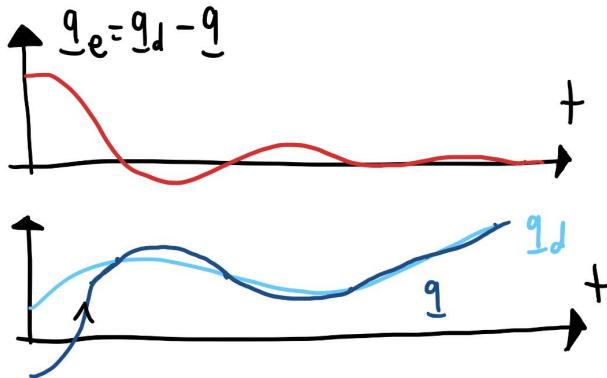


FIGURE 9.3 – Dynamique de l'erreur et convergence sur la trajectoire

On voit ici selon cette équation que ces paramètres dans la loi de commande ont été intégré de sorte qu'ils correspondent directement aux paramètres de fréquence naturelle et d'amortissement pour la forme standard

d'une équation différentielle d'ordre deux. Ici ζ serait un paramètre de la loi de commande directement relié au dépassement et au temps d'établissement de l'erreur du système en boucle fermée et le w au temps de monté et la bande passante du système en boucle fermée. De façon plus général, dans l'équation (9.8) le terme scalaire w^2 pourrait être remplacé par une matrice K_p et le terme $2\zeta w$ par une matrice K_d . Plutôt qu'avoir alors une dynamique d'erreur identique pour chaque DDL, on pourrait paramétriser des comportements plus varié. Par exemple, on pourrait choisir des matrices diagonales et les termes indépendants sur leur diagonale correspondraient à des valeurs de w et ζ qui pourraient être choisi indépendemment pour chaque DDL du robot.

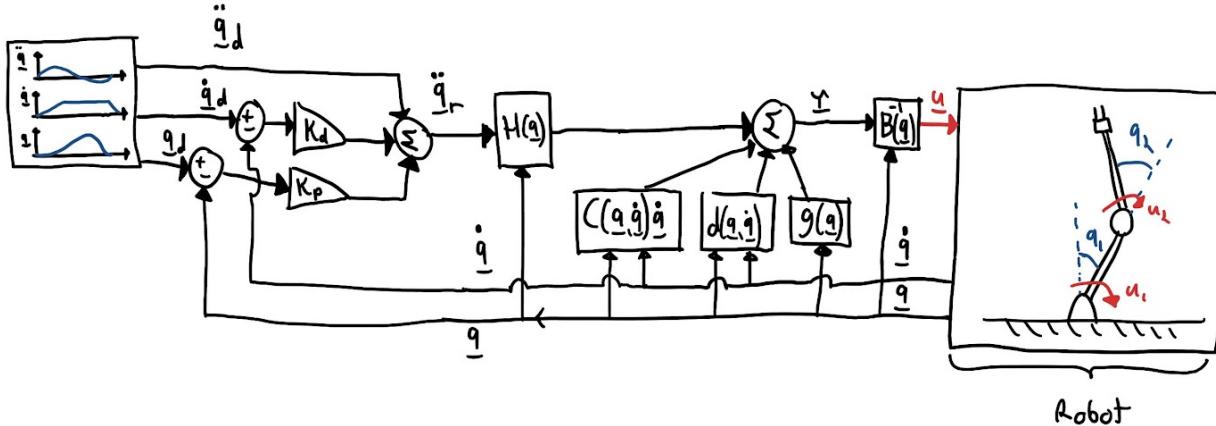


FIGURE 9.4 – Commande par couple calculé pour suivre une trajectoire.

Donc pour résumé, en assumant que notre modèle dynamique est parfait et que nous avons aussi des mesures exacte pour \underline{q} et $\dot{\underline{q}}$, il est possible d'imposer une accélération instantanée \ddot{q}_r arbitraire (si on ignore les limites des actionneurs pour le moment). Ensuite, connaissant la trajectoire cible $\ddot{q}_d(t)$ et ses dérivées d'ordre 1 et 2, il est possible d'imposer une accélération de sorte à avoir une erreur par rapport à la trajectoire $q_e(t)$ qui tend vers zéro.

Si on regarde comment l'erreur est relié à une force demandée par la loi de commande, on peut interpréter cette relation comme une matrice de rigidité, de même que pour un la relation avec la dérivé de l'erreur qui peut être interpréter comme une matrice d'amortissement. Les autres termes qui ne sont pas relié à l'erreur ou l'accélération de la trajectoire cible ont tous comme rôle d'annuler des forces internes. Donc si on décortique la loi de commande, on peut regrouper les termes ainsi :

$$\underline{u} = B(\underline{q})^{-1} [H(\underline{q})\ddot{q}_r + C(\underline{q}, \dot{\underline{q}})\dot{\underline{q}} + d(\underline{q}, \dot{\underline{q}}) + g(\underline{q})] \quad (9.9)$$

$$\underline{u} = B(\underline{q})^{-1} [H(\underline{q}) (\ddot{q}_d + K_d \dot{q}_e + K_p q_e) + C(\underline{q}, \dot{\underline{q}})\dot{\underline{q}} + d(\underline{q}, \dot{\underline{q}}) + g(\underline{q})] \quad (9.10)$$

$$\underline{u} = \underbrace{[B(\underline{q})^{-1} H(\underline{q})] \ddot{q}_d}_{\text{Termes anticipatif}} + \underbrace{[B(\underline{q})^{-1} H(\underline{q}) K_d] \dot{q}_e}_{\text{Termes réactifs}} + \underbrace{[B(\underline{q})^{-1} H(\underline{q}) K_p] q_e}_{\text{Termes de compensation}} + \underbrace{B(\underline{q})^{-1} [C(\underline{q}, \dot{\underline{q}})\dot{\underline{q}} + d(\underline{q}, \dot{\underline{q}}) + g(\underline{q})]}_{(9.11)}$$

et comparer à une loi de commande plus simple d'impédance dans le domaine des joints, présenté à la section 8.4, qui est en fait aussi équivalent à utiliser des lois de commande de type proportionnelle-dérivée indépendantes pour chacun des joints. Premièrement, en plus de termes réactifs la méthode du couple calculé inclus plusieurs termes de compensation des forces internes du système, ainsi qu'un terme anticipatif si elle est utilisée pour faire du suivi de trajectoire. Ensuite, les termes réactifs ont une structure similaire, des vecteurs-colonnes d'erreur multiplient des matrices de gains. Toutefois, on remarque que dans le cas du couple calculé les matrices de gains K_p et K_d sont ajusté avec en étant multipliés par la matrice d'inertie $H(\underline{q})$ et la matrice d'actionnement $B(\underline{q})^{-1}$. Cet ajustement fait que nos gains dans les matrices K_p et K_d spécifient pas un lien linéaire avec un réaction en force, mais plutôt avec une réaction en accélération, pour laquelle on calcul la force nécessaire avec les matrice H et B .

9.2.3 Compensation de la friction

Dans les méthodes de commande comme celle du couple calculé, la loi de commande vise à compenser pour tous les termes dynamiques, incluant les forces dissipatives comme la friction qui sont difficiles à modéliser. Il faut toutefois prendre garde de ne pas sur-estimer l'amplitude de la force de friction dans une boucle de compensation car cela peut rendre le système instable. Les forces dissipatives aident naturellement à la stabilité d'un système, ils dissipent de l'énergie mécanique et donc forcent le système tranquillement vers un état où l'énergie est minimum. Une compensation de friction annule donc l'effet dissipatif de cette force, mais le problème est surtout si on surestime de niveau de friction que notre contrôleur compense trop. Dans ce cas l'effet net de la vraie friction plus la compensation par le contrôleur est une injection d'énergie dans le système, la force net pousse dans le même sens que la vitesse, ce qui risque de rendre le système instable.

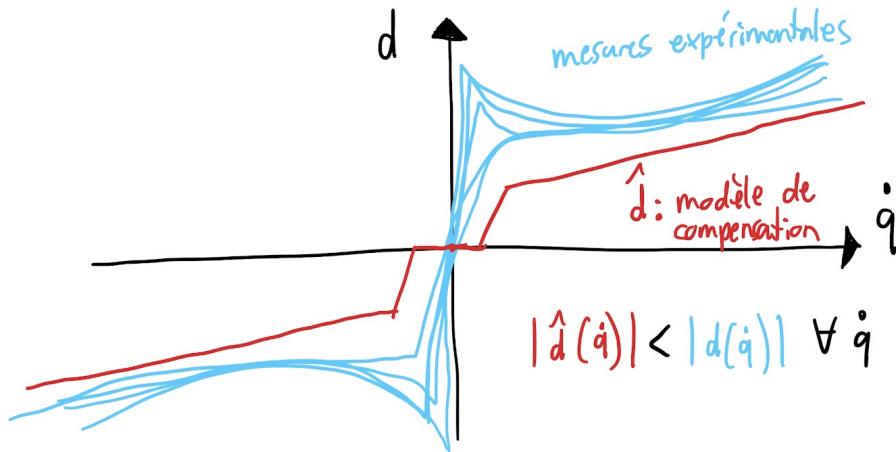


FIGURE 9.5 – Modèle de compensation de friction

Une approche pour éviter cette situation est de toujours avoir un modèle de compensation de friction qui sous-estime l'amplitude de la force de friction de façon conservatrice, comme illustré à la figure 9.5. Mathématiquement, pour garantir que la compensation de friction ne déstabilise pas le système, on peut choisir un modèle de compensation de friction \hat{d} avec une amplitude toujours inférieur à la vrai force de friction d pour toutes les conditions. Lorsqu'on a un modèle de friction pour chaque joint i qui dépend juste de la vitesse de ce même joint, cette condition s'exprime par :

$$|\hat{d}_i(\dot{q}_i)| < |d_i(\dot{q}_i)| \quad \forall \dot{q}_i \quad \forall i \quad (9.12)$$

Sous-estimer la friction dans la loi de commande du couple calculé, ou bien ne pas inclure cette compensation, ne met pas typiquement pas en péril la convergence du système sur la cible. La dynamique de l'erreur va toutefois être affecté, les propriétés d'amortissements vont être affectés. Une approche de conception peut être de retirer le terme dérivé K_d de la loi de commande et le terme de compensation de friction, pour laisser la friction naturelle du système s'occuper de l'amortissement du système.

9.2.4 Suivi d'une trajectoire définie dans l'espace de la tâche

À venir !

9.2.5 Limites de la méthode

À venir !

9.2.6 Commande en impédance incluant les effets inertIELS

TODO voir notes manuscrites alex 3 juillet 2022

9.3 Commande robuste

La commande robuste est la science de gérer l'incertitude lors de la conception d'une loi de commande. Dans les sections précédentes, nous avions des lois de commande qui utilisaient divers propriétés du modèle cinématique et/ou dynamique du robot. Les démonstrations que ces lois de commande convergeaient assumaient alors que nos modèles utilisés dans les lois de commande étaient exacte. En pratique il y a toujours un certain degré d'incertitude, plus l'incertitude est grande plus il y a intérêt à considérer cet aspect explicitement avec des méthodes de commande robuste.

9.3.1 Approche pour modéliser l'incertitude

Dans cette section, la méthode de commande présentée est basée sur une modélisation de l'incertitude comme un vecteur de forces inconnues \underline{w} qui s'additionnent aux autres termes de l'équation des manipulateurs :

$$H(\underline{q})\ddot{\underline{q}} + C(\underline{q}, \dot{\underline{q}})\dot{\underline{q}} + d(\underline{q}, \dot{\underline{q}}) + g(\underline{q}) = B(\underline{q})\underline{u} + \underline{w} \quad (9.13)$$

Ensuite, en fonction de bornes de valeurs minimum et maximum que peut prendre le vecteur \underline{w} il sera possible de synthétiser des loi de commandes qui vont garantir la convergence malgré cette incertitude.

▶

Capsule vidéo
Grandes familles de méthodes de commande pour gérer l'incertitude
<https://youtu.be/hbZBF-0EZEw>

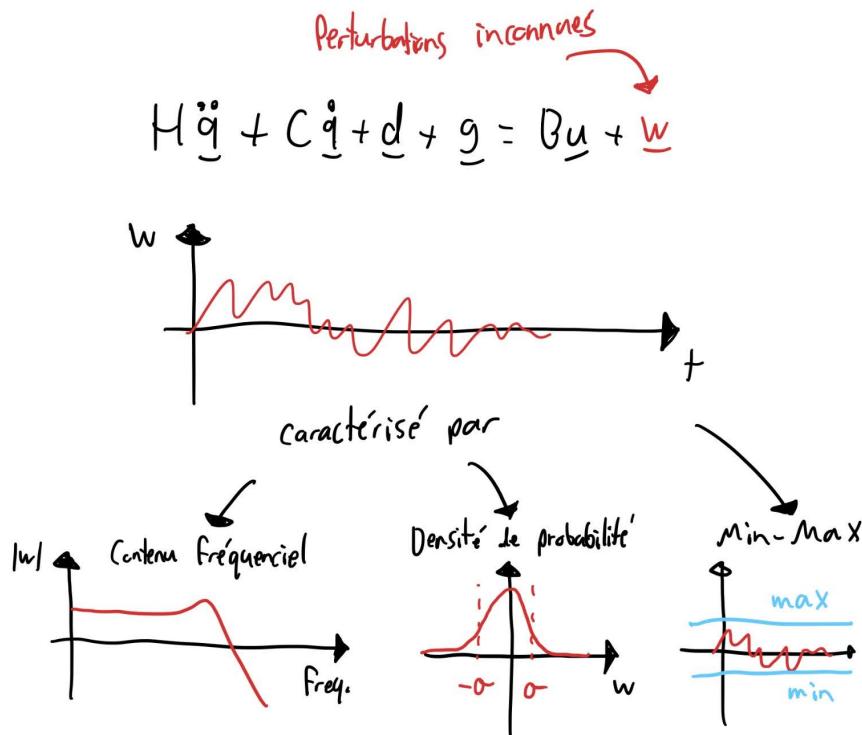


FIGURE 9.6 – Divers métriques pour caractériser l'incertitude

9.4 Méthode du mode glissant

La méthode du mode glissant est une approche de commande qui a comme principal attrait de pouvoir garantir la performance d'un système, malgré la présence de perturbations. En effet, si on connaît les valeurs maximums et minimums qu'une perturbation peut prendre nous aurons une méthode pour sélectionner des gains qui peuvent garantir la convergence.

Capsule vidéo
Commande avec le mode glissant
<https://youtu.be/0Asg81SBjmk>

Premièrement, la stratégie pour garantir la convergence malgré l'incertitude est basé sur le principe de "pousser" très fort sur système vers une trajectoire cible de sorte que peu importe la valeur prise par la perturbation w le système est tout de même forcé sur la trajectoire. Comme illustré à la figure 9.7, toutes les évolutions possibles du système sont toujours dans la bonne direction malgré l'incertitude avec cette approche.

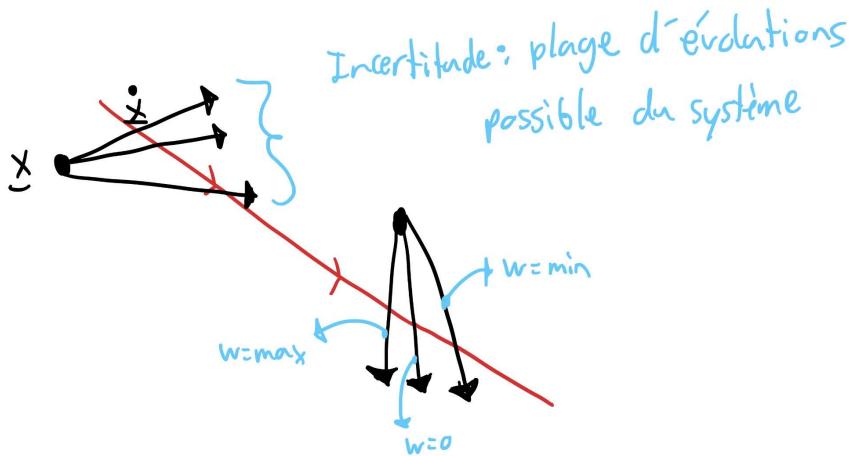


FIGURE 9.7 – Principe de fonctionnement pour la commande par surface glissante

Exercice de code
Robot avec une loi de commande du mode glissant
<https://colab.research.google.com/drive/1DQhoq-NbWHHI6RSf-KQXwoW7SLwfldgk?usp=sharing>

Avec la méthode de la surface glissante, le système va être forcé sur une trajectoire définie par l'équation suivante :

$$\underline{s} = \dot{\underline{q}}_e + \lambda \underline{q}_e = 0 \quad (9.14)$$

où les variables s sont des coordonnées pour chaque DDL du système, une combinaison de la position et de la vitesse (relatif à une trajectoire cible), choisi avec un paramètre λ de sorte que $s = 0$ correspond à une trajectoire qui converge exponentiellement vers $\underline{q}_e = 0$. En effet,

$$\dot{\underline{q}}_e = -\lambda \underline{q}_e \quad \text{si } \underline{s} = 0 \quad (9.15)$$

une équation différentielle linéaire d'ordre un pour chaque DDL avec comme solution :

$$\underline{q}_e(t) = \underline{q}_e(0) e^{-\lambda t} \quad (9.16)$$

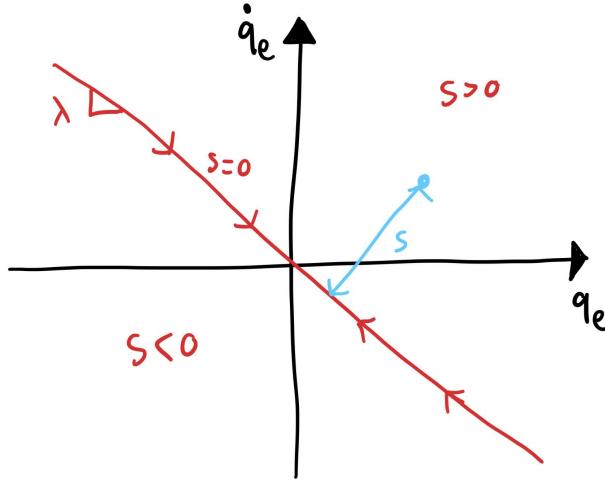


FIGURE 9.8 – Surface glissante

Comme illustré à la figure 9.8, les variables s sont donc des coordonnées pour lesquelles le sous-ensemble $s = 0$ implique une convergence exponentielle vers l'origine, c'est ce sous-ensemble qu'on appelle la surface glissante.

La loi de commande par surface glissante va avoir deux composantes :

$$\underline{u} = B(\underline{q})^{-1} \left[\underbrace{\hat{\tau}}_{\text{Couple équivalent}} + \underbrace{k \operatorname{sgn} s}_{\text{Couple discontinu}} \right] \quad (9.17)$$

Le premier terme, le couple équivalent, est un vecteur de couple calculé de façon continue qui compense plusieurs forces internes de façon similaire à la méthode du couple calculé. Le deuxième terme, est un vecteur de couple discontinu, où chaque composant peut prendre deux valeur selon la valeur de la variable s associé à ce DDL. La figure 9.9 illustre le schéma bloc de cette loi de commande. Le couple équivalent $\hat{\tau}$

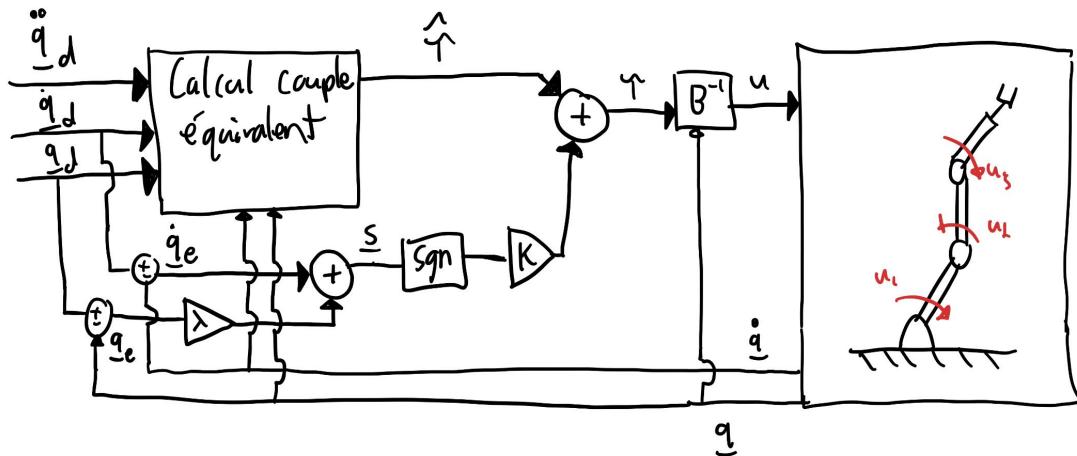


FIGURE 9.9 – Surface glissante

est calculé en fonction de la trajectoire désirée et l'état actuel du robot ainsi :

$$\hat{\tau} = H \ddot{q}_r + C \dot{q}_r + d + \underline{g} \quad (9.18)$$

avec :

$$\dot{\underline{q}}_r = \dot{\underline{q}}_d + \lambda \dot{\underline{q}}_e \quad (9.19)$$

$$\ddot{\underline{q}}_r = \ddot{\underline{q}}_d + \lambda \dot{\underline{q}}_e \quad (9.20)$$

$$(9.21)$$

Le signal total de commande u calculé aura donc typiquement une allure continue lorsque le système est loin de la surface glissante $s = 0$ et le signal fera des saut rapide entre deux valeurs de u lorsque le système est proche de la surface glissante, comme illustré aux figures 9.10 et 9.11. La fréquence des oscillations entre les valeurs de u va dépendre en pratique de la fréquence de la boucle de contrôle. Les oscillations vont aussi se refléter sur la trajectoire du système comme un va et vient d'un côté à l'autre de la surface glissante.

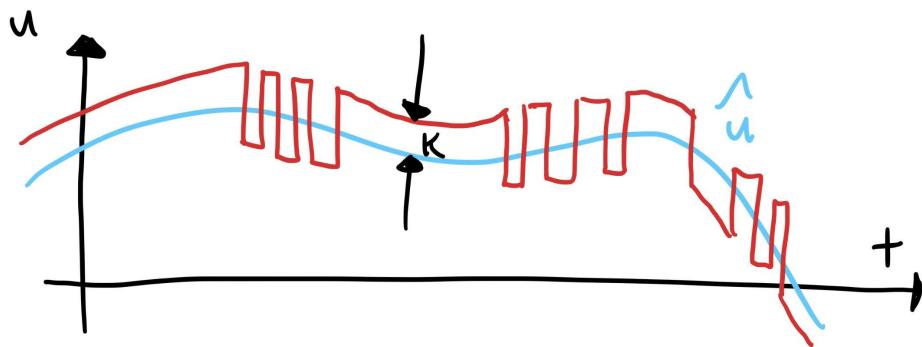


FIGURE 9.10 – Surface glissante

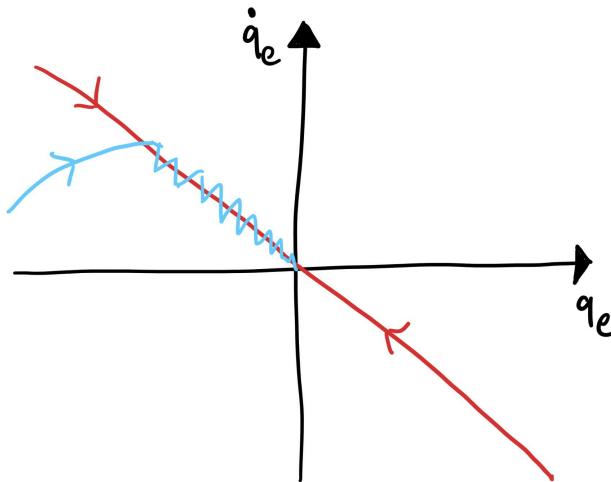
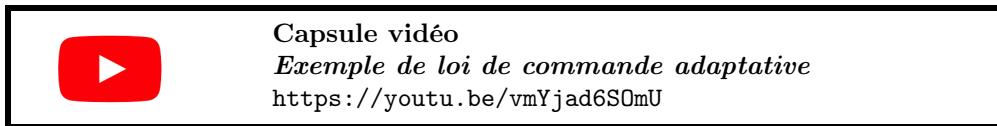


FIGURE 9.11 – Surface glissante

9.5 Commande adaptative

La commande dite adaptative, est une méthode de alternative pour gérer l'incertitude lorsque celle-ci est caractérisé par une erreur sur les paramètres dans l'équation dynamique. Cette approche s'applique bien pour les scénarios où la principale source d'incertitude n'est pas des perturbations externes, mais plutôt une erreur d'estimation sur les paramètres cinématiques ou inertIELS d'un système robotique. La stratégie d'une loi de commande adaptative peut se résumer en deux processus parallèles : **1) identification système** : on utilise les signaux d'entrée et de sortie du système pour effectuer une estimation en continue des paramètres du modèle, et **2) loi de commande** : on utilise l'approximation des paramètres dans une loi de commande de type couple calculé, ou autre approche basé sur un modèle du système. Cette approche est particulièrement adapté lorsqu'un système robotisé va avoir un chargement inconnu lors d'une opération par exemple.



Apprentissage machine : Le principe d'adaptation peut être vu comme une forme d'apprentissage machine en ligne. Avec une loi de commande adaptative, on cherche à "apprendre" automatiquement les bons paramètres d'une loi de commande (vecteur-colonne \underline{a}) à l'aide des données observées. La matrice P de gain d'adaptation peut être vue comme des paramètres de taux d'apprentissage.

9.5.1 Approche pour modéliser l'incertitude

Dans cette section, la méthode de commande présentée est basée sur une modélisation de l'incertitude où on considère que des paramètres dans le modèle dynamique sont des approximations inexactes dans notre loi de commande $\hat{\underline{a}}_i$. Par exemple, la masse d'un robot ou la longueur sont toujours approximés à un certain degré lorsque l'on code une loi de commande. On va ici noter les paramètres exactes a_i , les approximations \hat{a}_i , et l'erreur sur un paramètre sera a_{ei} . Les paramètres seront regroupés des vecteur-colonnes :

$$\underbrace{\underline{a}_e}_{\text{erreur}} = \underbrace{\underline{a}}_{\text{réel}} - \underbrace{\hat{\underline{a}}}_{\text{approximation}} \quad (9.22)$$

Un atout qui sera exploité pour les systèmes dynamiques qui peuvent être représentés par l'équation des manipulateurs, est qu'on peut structurer l'équation de la dynamique comme une multiplication d'une matrice de termes connus Y , et du vecteur-colonnes qui regroupe tous les paramètres constants de l'équation.

$$H(\underline{q})\ddot{\underline{q}} + C(\underline{q}, \dot{\underline{q}})\dot{\underline{q}} + d(\underline{q}, \dot{\underline{q}}) + g(\underline{q}) = \tau \quad (9.23)$$

$$Y(\underline{q}, \dot{\underline{q}}, \ddot{\underline{q}}) \underline{a} = \tau \quad (9.24)$$

Les termes connus dans la matrice Y vont être des fonctions des accélérations, vitesses et positions du robot à un instant donné. Par exemple, on pourrait avoir des termes tels que :

$$Y_{ii} \in \{\ddot{q}_i, \dot{q}_i, \dot{q}_i^2, \dot{q}_i \dot{q}_j, \cos q_i, \sin q_i \dot{q}_i^2, \dots\} \quad (9.25)$$

et dans le vecteur \underline{a} des combinaisons de paramètres géométriques, inertiel et autres qui multiplient les termes de Y pour obtenir l'équation des manipulateurs :

$$a_i \in \{ml^2, mgl, b_i, \dots\} \quad (9.26)$$

Exemple 1. Séparation des paramètres inconnus pour un pendule simple

Pour un pendule simple avec de la friction sèche, malgré les non-linéarités dans les équations il est possible de séparer les termes connus (mesurables) qui sont mesurable et les paramètres constants :

$$ml^2\ddot{q} + \mu \operatorname{sgn} \dot{q} + mgl \sin q = \tau \quad (9.27)$$

$$\underbrace{\begin{bmatrix} \ddot{q} & \operatorname{sgn} \dot{q} & \sin q \end{bmatrix}}_Y \underbrace{\begin{bmatrix} ml^2 \\ \mu \\ mgl \end{bmatrix}}_{\underline{a}} = \tau \quad (9.28)$$

Si on mesure l'angle q (et ses dérivés) on peut construire tout les termes de la matrice Y . Les paramètres constants sont impliqués linéairement par rapport aux termes de Y .

9.5.2 Loi de commande

Cette section présente une approche de commande adaptative générique pour un système pleinement actionné représenté par l'équation des manipulateurs. La loi de commande a une structure similaire aux lois de commandes présenté dans ce chapitre : un terme de compensation de dynamique et un terme supplémentaire pour garantir la convergence sur une cible. Ici la grande différence est que le terme de compensation dynamique est calculé avec l'approximation actuel des paramètres inconnus \hat{a} :

$$\underline{\tau} = \underbrace{Y_r \hat{a}}_{\text{compensation dynamique}} + \underbrace{K_s}_{\text{rétroaction de type PD}} \quad (9.29)$$

où K est une matrice de paramètres de gains.

Ensuite, l'approximation des paramètres inconnus va être continuellement mise à jour avec l'équation suivante qui dicte la dérivée temporelle de nos approximations comme une fonction de variables mesurables :

$$\dot{\hat{a}} = -P^T Y_r^T \underline{s} \quad (9.30)$$

où P est une matrice inversible de gains d'adaptation. La loi de commande va ici avoir des états internes et une mémoire (la consigne $\underline{\tau}$ peut prendre différentes valeur pour un état donné du système physique selon l'historique). Le vecteur d'approximation des paramètres inconnus \hat{a} constitue des états internes qui vont être incrémenté continuellement avec un schéma d'intégration :

$$\hat{a}(t) = \int_{t_0}^t (-P^T Y_r^T \underline{s}) dt + \hat{a}(t_0) \quad (9.31)$$

Des variables intermédiaires ont été utilisées dans les équations précédentes et sont définies si dessous :

$$\underline{q}_e = \underline{q}_d - \underline{q} \quad (9.32)$$

$$\underline{\dot{q}}_r = \underline{\dot{q}}_d + \lambda \underline{q}_e \quad (9.33)$$

$$\underline{\ddot{q}}_r = \underline{\ddot{q}}_d + \lambda \underline{\dot{q}}_e \quad (9.34)$$

$$\underline{s} = \underline{\dot{q}}_e + \lambda \underline{q}_e = \underline{\dot{q}}_r - \underline{\dot{q}} \quad (9.35)$$

Il est aussi à noter qu'il y a deux version de la matrice Y :

$$Y \underline{a} = H \ddot{q} + C \dot{q} + \underline{d} + \underline{g} = \underline{\tau} \quad (9.36)$$

$$Y_r \underline{a} = H \ddot{\underline{q}}_r + C \dot{\underline{q}}_r + \underline{d} + \underline{g} \quad (9.37)$$

Y dépend seulement des positions, vitesses et accélération, alors que Y_r n'utilise pas l'accélération réelle mais est plutôt basé sur l'erreur :

$$Y = Y(\ddot{q}, \dot{q}, q) \quad (9.38)$$

$$Y_r = Y_r(\ddot{\underline{q}}_r, \dot{\underline{q}}_r, \underline{q}) \quad (9.39)$$

**Exercice de code***Commande adaptative pour un pendule simple*

<https://colab.research.google.com/drive/1J8C-CvJdrhDU4q1294057Qopn3tnEBNi?usp=sharing>

Preuve 9.5.1 Il est possible de démontrer que la loi de commande proposée dans cette section, incluant la loi d'adaptation, va faire converger le système sur la trajectoire désirée pour n'importe quel valeur initial d'estimation des paramètres \hat{a} .

Avec comme fonction de Lyapunov :

$$V = \frac{1}{2}\underline{s}^T H \underline{s} + \frac{1}{2}\hat{\underline{a}}^T P^{-1} \underline{a}_e \quad (9.40)$$

Il est possible de démontrer que la dérivée temporelle va être strictement négative sauf lorsque $\underline{s} = \underline{0}$:

$$\dot{V} = \underline{s}^T (H \dot{\underline{s}} + 2\dot{H} \underline{s}) + \hat{\underline{a}}^T P^{-1} \underline{a}_e \quad (9.41)$$

$$\dot{V} = \underline{s}^T (H \ddot{\underline{q}}_r - \underline{\tau} + C \dot{\underline{q}} + \underline{d} + \underline{g} + 2\dot{H} \underline{q}_r - 2\dot{H} \underline{q}) + \hat{\underline{a}}^T P^{-1} \underline{a}_e \quad (9.42)$$

$$\dot{V} = -\underline{s}^T K \underline{s} + \underbrace{\underline{s}^T Y_r \underline{a}_e}_{0} + \hat{\underline{a}}^T P^{-1} \underline{a}_e \quad (9.43)$$

$$\dot{V} = -\underline{s}^T K \underline{s} < 0 \quad \forall \quad \underline{s} \neq \underline{0} \quad (9.44)$$

En utilisant le lemme de Barbalat (voir livre de commande non-linéaire), il est possible de conclure que $\underline{s} \rightarrow \underline{0}$ ce qui implique qu'on va converger sur la trajectoire car $\underline{q}_e \rightarrow \underline{0}$. Il est toutefois important de noter que cela n'implique pas que $\underline{a}_e \rightarrow \underline{0}$.

Exemple 2. Loi de commande et d'adaptation pour un pendule simple

Si on reprend l'exemple du pendule simple décrit par l'équation suivante :

$$\underbrace{\begin{bmatrix} \ddot{q} & \operatorname{sgn} \dot{q} & \sin q \end{bmatrix}}_Y \underbrace{\begin{bmatrix} ml^2 \\ \mu \\ mgl \end{bmatrix}}_{\underline{a}} = \tau \quad (9.45)$$

La loi de commande serait alors :

$$\tau = Y_r \hat{\underline{a}} + K \underline{s} \quad (9.46)$$

$$\tau = \begin{bmatrix} \ddot{q}_r & \operatorname{sgn} \dot{q}_r & \sin q_r \end{bmatrix} \begin{bmatrix} \widehat{ml^2} \\ \widehat{\mu} \\ \widehat{mgl} \end{bmatrix} + K(\dot{q}_e + \lambda q_e) \quad (9.47)$$

$$\tau = \widehat{ml^2}(\ddot{q}_d + \lambda \dot{q}_e) + \widehat{\mu} \operatorname{sgn} \dot{q} + \widehat{mgl} \sin q + K(\dot{q}_e + \lambda q_e) \quad (9.48)$$

$$\tau = \widehat{ml^2} \ddot{q}_d + \widehat{\mu} \operatorname{sgn} \dot{q} + \widehat{mgl} \sin q + (\widehat{ml^2} \lambda + K) \dot{q}_e + K \lambda q_e \quad (9.49)$$

et la loi d'adaptation serait alors :

$$\dot{\hat{a}} = -P^T Y_r^T s \quad (9.50)$$

$$\frac{d}{dt} \begin{bmatrix} \widehat{ml^2} \\ \widehat{\mu} \\ \widehat{mgl} \end{bmatrix} = - \begin{bmatrix} p_1 & 0 & 0 \\ 0 & p_2 & 0 \\ 0 & 0 & p_3 \end{bmatrix} \begin{bmatrix} \ddot{q}_r \\ \operatorname{sgn} \dot{q} \\ \sin q \end{bmatrix} (\dot{q}_e + \lambda q_e) \quad (9.51)$$

$$\frac{d}{dt} \begin{bmatrix} \widehat{ml^2} \\ \widehat{\mu} \\ \widehat{mgl} \end{bmatrix} = - \begin{bmatrix} p_1(\ddot{q}_d + \lambda \dot{q}_e) \\ p_2 \operatorname{sgn} \dot{q} \\ p_3 \sin q \end{bmatrix} (\dot{q}_e + \lambda q_e) \quad (9.52)$$

9.5.3 Estimation avec les moindres carrés

Pourquoi ne pas plutôt estimer les paramètres avec un schéma des moindres carrés ? Est-ce qu'il y a un enjeux de stabilité ?

À venir !

9.5.4 Adaptation à des erreurs dans la matrice jacobienne

À venir !

9.6 Commande hybride en position et force

À venir!

9.7 Méthodes d'analyse de la stabilité



Capsule vidéo

Introduction à l'analyse de stabilité pour les systèmes non-linéaires

<https://youtu.be/q00qa5J3zEk>

À venir !

9.8 Commande optimale

Jusqu'à maintenant, les méthodes de commande présentées la section 8 et la section 9, étaient basé sur l'hypothèse qu'on pourrait résumé à dire que **les actionneurs du système étaient suffisamment nombreux, bien positionnés et puissants pour imposer une accélération arbitraire au système robotique**. Si cette condition n'est pas remplie, le problème de conception d'une loi de commande devient plus compliqué. Dans cette situation il faut gérer **une planification de l'évolution du système sur un horizon de temps futur** et une méthodologie appropriée pour accomplir cela est appelée la **commande optimale** où les effets long-termes des consignes choisies par une loi de commande sont encodés dans une fonction de coût. De plus, même si le robot à commander rencontre les conditions pour utiliser les techniques qui imposent un accélération, la commande optimale est aussi un outils pertinent à utiliser aussi si on désire utiliser avec parcimonie les actionneurs pour, par exemple, **accomplir une tâche en utilisant un minimum d'énergie**.

La plupart des méthodes de commande optimale sont conçus pour travailler avec la forme plus générale des équations différentiels :

$$\dot{\underline{x}} = f(\underline{x}, \underline{u}) \quad (9.53)$$

des contraintes sur les états et les consignes :

$$\underline{x} \in X_{ok} \quad (9.54)$$

$$\underline{u} \in U_{ok} \quad (9.55)$$

$$(9.56)$$

et un objectif caractérisé par une fonction de coût à minimiser :

$$J = \int (g(\underline{x}, \underline{u}) dt + h(\underline{x}_f)) \quad (9.57)$$

Voici quelques liens vers des introductions à la commande optimale appliquée aux systèmes robotiques.



Capsule vidéo
Exemple de loi de commande optimale pour un double intégrateur
<https://youtu.be/wKjEAXFvX1Q>



Capsule vidéo
Exemple de loi de commande optimale pour un pendule
https://youtu.be/iUlkKdEK_dU



Exercice de code
Démo d'introduction aux méthodes de commande optimales
<https://colab.research.google.com/drive/1wXm1IqNGC2LrJkmyj56Y109b5ZDHVboq?usp=sharing>

Détails à venir !

Note sur la séquence de lecture : Comme les méthodes de commande optimale s'applique à une classe de systèmes plus large, incluant les véhicules et autres système robotique pour lesquels il est impossible

d'imposer une accélération avec les actionneurs, ce sujet est reporté à un chapitre ultérieur après ceux qui introduisent les véhicules. Le cadre théorique générique de ces outils est aussi présenté en annexe au chapitre 16

Chapitre 10

Planification des trajectoires

10.1 Introduction

Dans les précédents chapitre ont présentés des lois de commandes qui calculent les commandes à envoyer aux actionneurs pour obtenir soit des positions, des vitesses, des accélérations ou des forces désirée. Certaines loi de commandes nécessitait seulement une position finale cible alors que pour d'autre nous devions spécifier une position, une vitesse et une accélération en tout temps sur une trajectoire cible. Plusieurs tâches en robotique nécessite donc une couche intermédiaire entre des spécifications à haut niveau pour une tâche et les références qui doivent être fournit à une boucle de commande à bas niveau, cette étape intermédiaire est appelée la planification de trajectoire.



Capsule vidéo

Introduction à la planification de trajectoires

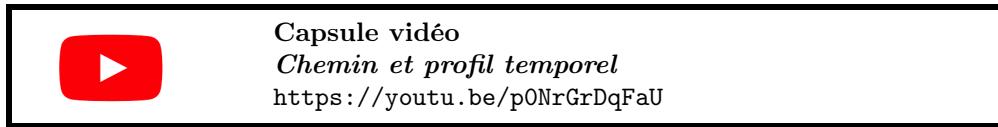
https://youtu.be/_SQumKYdFnM

Pourquoi donner directement la position finale désirée à un contrôleur bas niveau en position n'est pas toujours approprié ? L'étape de planification de trajectoire est généralement essentiel lorsqu'un robot doit faire un mouvement de grande amplitude, i.e. lorsque la position cible est éloignée de la position actuelle. Les loi de commandes bas-niveau sont typiquement conçu pour garantir la convergence vers la position cible mais pas le détail du chemin pour s'y rendre ni la vitesse d'approche; plus la cible est loin, plus il va y avoir de raison de mieux contrôler les étapes intermédiaire pour ce rendre à la cible. Voici une liste non-exhaustive de raisons pour mieux contrôler le trajet jusqu'à une cible :

1. **Obstacles et contraintes en position :** Parfois il y a des positions/configurations sur lesquelles on ne veux pas que le système robotique ce retrouve (collisions avec des obstacles, singularités, etc.). Les lois de commande simples vont typiquement aller directement droit vers la cible et possiblement la dépasser avant de se stabiliser sur celle-ci sans tenir compte de contraintes. Un planificateur de trajectoire peut déterminer un chemin à suivre pour atteindre la cible qui évite des obstacles et positions inadéquates.
2. **Contraintes en vitesse :** Les contrôleurs simples vont typiquement aller vers la cible avec une vitesse proportionnelle avec l'erreur initiale sans tenir compte des limites physiques. Si des saturations à bas niveau font que la vitesse réelle ne suit pas la vitesse commandée le comportement et la convergence de la loi de commande seront incertains. De plus, pour des raisons de sécurité il peut être nécessaire de limiter la vitesse d'un robot. Un planificateur de trajectoire peut déterminer un profil de vitesse à suivre qui respecte des contraintes.
3. **Faisabilité dynamique :** Finalement, surtout pour des robots dynamiques et/ou qui bougent rapidement, les lois de commande de base peuvent facilement se retrouver dans des situations où ils demandent des niveaux d'accélération ou de forces aux actionneurs qui se sont pas physiquement possible. Un planificateur peut permettre de déterminer une séquence d'action physiquement possible qui vont mener le robot jusqu'à la cible tout en respectant des contraintes.

10.2 Chemin et profil temporel

Système de coordonnées pour définir une trajectoire On peut définir des trajectoires dans plusieurs systèmes de coordonnées. Toutefois, pour simplifier la notation, nous présenterons ici d'abord plusieurs concepts en considérant que le trajet est toujours défini dans l'espace des joints d'un robot. Nous verrons ensuite que ces concepts se généralisent pour d'autres systèmes de coordonnées.



Il est parfois utile d'utiliser le concept d'un chemin pour découpler le problème de planification de trajectoire en deux étapes : 1) trouver un chemin, une description géométrique d'une séquence de configuration \underline{q} reliant le point de départ et la cible; et ensuite 2) déterminer un profil temporel de vitesse sur ce chemin pour fixer une trajectoire temporelle.

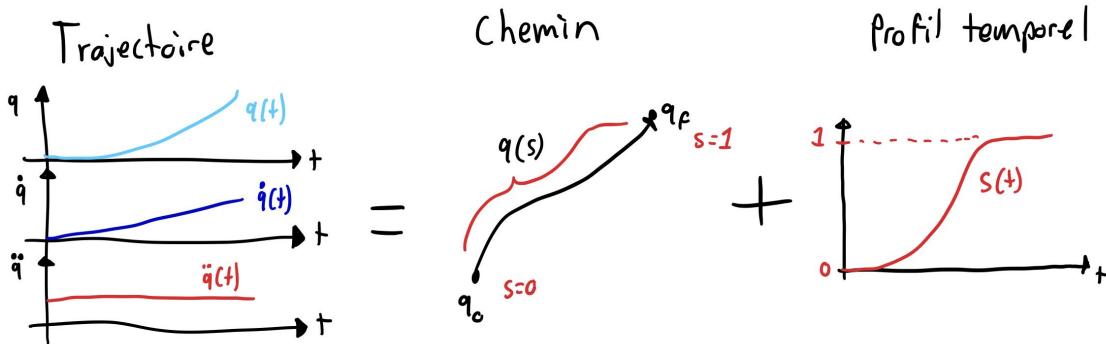


FIGURE 10.1 – Trajectoire (dans l'espace des joints), chemin et profil temporel

Trajectoire Une trajectoire est une fonction qui détermine la position d'un robot comme une fonction du temps $\underline{q}(t)$. Normalement il est désirable d'avoir des fonctions qui sont dérivables deux fois, i.e. que la dérivé temporelle de cette fonction ne soit pas discontinue, ce qui impliquerait des accélérations d'amplitude infinie. La trajectoire est donc ultimement les trois fonctions suivantes :

$$\text{Trajectoire : } \underline{q}(t), \dot{\underline{q}}(t), \ddot{\underline{q}}(t) \quad \text{avec } t \in [0, t_f] \quad (10.1)$$

Chemin Une chemin est une fonction qui détermine la position d'un robot comme une fonction continue d'une variable scalaire $s \in [0, 1]$, qui est égale à 0 au début du chemin et à 1 à la fin du chemin.

$$\text{Chemin : } \underline{q}(s) \quad \text{avec } s \in [0, 1] \quad (10.2)$$

Profil temporel Un profil temporel est une trajectoire pour une variable scalaire intermédiaire qui caractérise la progression du système sur un chemin :

$$\text{Profil temporel : } s(t), \dot{s}(t), \ddot{s}(t) \quad \text{avec } t \in [0, t_f] \quad (10.3)$$

Lorsqu'on construit une fonction de trajectoire basé sur un chemin et un profil temporel on obtient les relations suivantes en appliquant une dérivée en chaîne :

$$\underline{q}(t) = \underline{q}(s(t)) \quad (10.4)$$

$$\dot{\underline{q}}(t) = \frac{\partial \underline{q}}{\partial s} \dot{s} \quad (10.5)$$

$$\ddot{\underline{q}}(t) = \frac{\partial \underline{q}}{\partial s} \ddot{s} + \frac{\partial^2 \underline{q}}{\partial s^2} \dot{s}^2 \quad (10.6)$$

10.2.1 Profils temporels

Cette section présente quelques types de profils temporels qui sont typiquement utilisée pour générer des trajectoires.

Profil temporel trapézoïdal

Une profil de type trapézoïdal, voir Figure 10.2, est utilisé fréquemment pour la génération de trajectoires simples pour plusieurs types de système asservis en position. Ce profil permet d'inclure directement une limite en vitesse v_{max} et une limite en accélération a_{max} , c'est le profile qui donne la trajectoire la plus rapide possible considérant ces deux contraintes. Il existe toutefois des profils temporels plus lisses sans discontinuités, voir les prochaines sections, ce qui peut être un avantage. Le profile trapézoïdal consiste en trois phases : une phase d'accélération avec $\ddot{s} = a_{max}$, une phase de croisière à vitesse constante avec $\dot{s} = v_{max}$ et une phase de décélération avec $\ddot{s} = -a_{max}$. Les profils de positions et d'accélérations associés sont illustrés à la Figure 10.3.

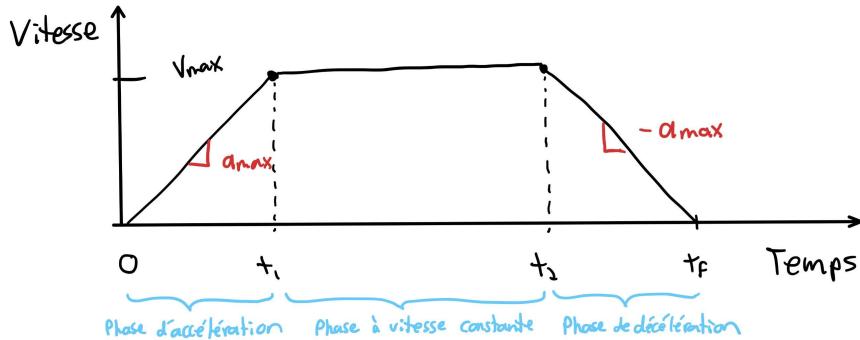


FIGURE 10.2 – Profil temporel de vitesse de type trapézoïdal

Si on utilise le concept de la variable s qui représente la fraction du chemin à parcourir (voir section 10.2), l'intégrale du profile de vitesse $\dot{s}(t)$ doit être égal à 1 (100% du chemin parcouru). Il reste donc seulement trois variable, donc seulement deux sont indépendantes, pour définir complètement le profile temporel de $s(t)$: la vitesse maximale v_{max} , l'accélération maximale a_{max} et la durée totale t_f du trajet. Il peut être intéressant de déterminer les contraintes v_{max} et a_{max} du profile basé sur les limites physiques d'un système et ainsi obtenir la trajectoire qui minimise t_f :

$$t_f = \frac{a_{max} + v_{max}^2}{a_{max} v_{max}} \quad (10.7)$$

Il est à noter que si $v_{max}^2/a_{max} > 1$ le système n'a pas le temps d'atteindre sa vitesse maximale de croisière et le profil est réduit à deux phases : accélération et décélération, voir Figure 10.4. Lorsque $v_{max}^2/a_{max} \leq 1$,

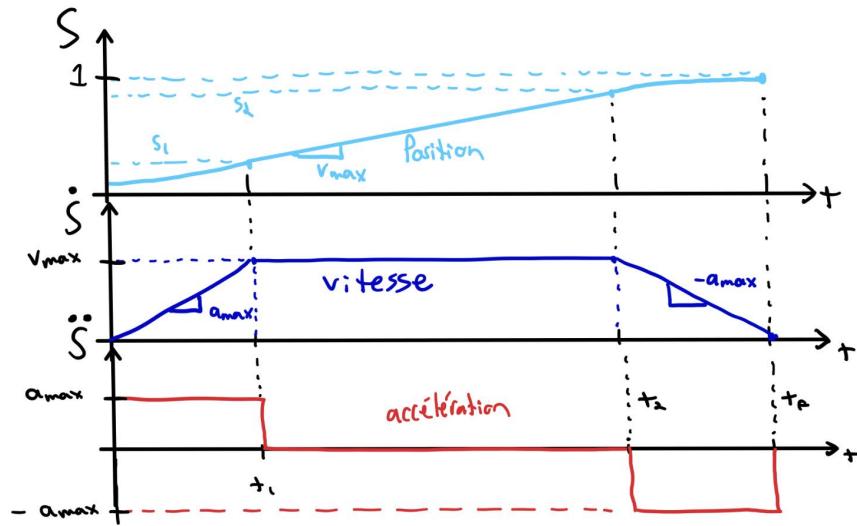


FIGURE 10.3 – Position, vitesse et accélération pour un profil temporel de type trapézoïdal

le profile est bien trapézoïdale et les équations de ce profile pour les trois segments temporels sont :

$$\text{si } 0 \leq t \leq \frac{v_{max}}{a_{max}} \quad \begin{cases} s(t) = \frac{1}{2}a_{max}t^2 \\ \dot{s}(t) = a_{max}t \\ \ddot{s}(t) = a_{max} \end{cases} \quad (10.8)$$

$$\text{si } \frac{v_{max}}{a_{max}} < t \leq T - \frac{v_{max}}{a_{max}} \quad \begin{cases} s(t) = v_{max}t - \frac{v_{max}^2}{2a} \\ \dot{s}(t) = v_{max} \\ \ddot{s}(t) = 0 \end{cases} \quad (10.9)$$

$$\text{si } T - \frac{v_{max}}{a_{max}} < t \leq T \quad \begin{cases} s(t) = \frac{2a_{max}v_{max}T - 2v_{max}^2 - a_{max}^2(t-T)^2}{2a_{max}} \\ \dot{s}(t) = a(T-t) \\ \ddot{s}(t) = -a \end{cases} \quad (10.10)$$

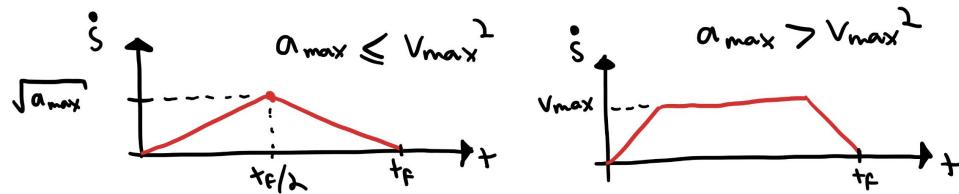


FIGURE 10.4 – Cas limite du profil trapézoïdale

Profil temporel polynomial d'ordre 3

Une alternative à une profil temporel de type trapézoïdale est d'utiliser une fonction polynomiale. Pour obtenir un profile de vitesse qui débute à l'arrêt et termine aussi à l'arrêt, l'ordre minimum de la fonction est de trois. On obtient alors un profile de position cubique, un profile de vitesse parabolique et un profile

d'accélération linéaire, avec comme équations :

$$s(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \quad (10.11)$$

$$\dot{s}(t) = a_1 + 2a_2 t + 3a_3 t^2 \quad (10.12)$$

$$\ddot{s}(t) = 2a_2 + 6a_3 t \quad (10.13)$$

Considérant que nous voulons représenter un mouvement qui débute et termine à l'arrêt, dans un temps t_f , nous avons les 4 contraintes suivantes à imposer : la position initial $s(0) = 0$, la position finale $s(T) = 1$, la vitesse initiale $\dot{s}(t_0) = 0$ et la vitesse terminale $\dot{s}(t_f) = 0$. Il y a donc un seul paramètre libre, la durée du trajet t_f , et on peut trouver que :

$$a_0 = 0 \quad a_1 = 0 \quad a_2 = 3/t_f^2 \quad a_3 = -2/t_f^3 \quad (10.14)$$

ce qui donne comme équations de profile :

$$s(t) = \frac{3}{t_f^2} t^2 - \frac{2}{t_f^3} t^3 \quad (10.15)$$

$$\dot{s}(t) = \frac{6}{t_f^2} t - \frac{6}{t_f^3} t^2 \quad (10.16)$$

$$\ddot{s}(t) = \frac{6}{t_f^2} - \frac{12}{t_f^3} t \quad (10.17)$$

qui sont illustrée à la Figure 10.5a.

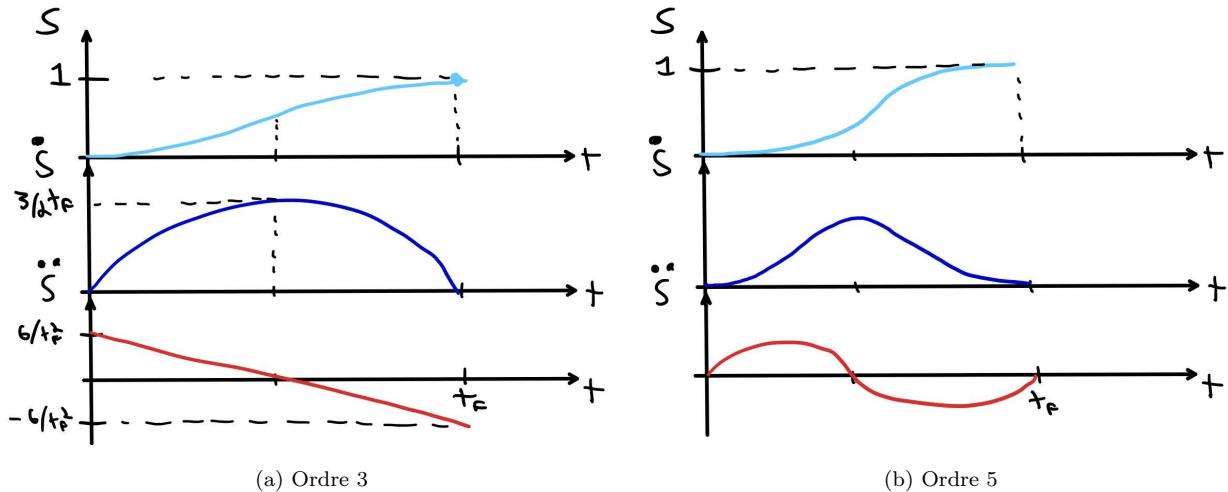
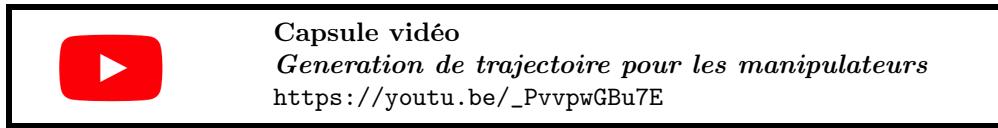


FIGURE 10.5 – Profils temporels polynomiaux

Profil temporel polynomial d'ordre 5

Une option pour avoir des trajectoires plus lisses est d'utiliser une fonction polynomiale d'ordre 5. Avec 5 paramètres il est possible de rajouter deux contraintes pour avoir une accélération qui débute et termine à une valeur de zéro, ce qui permet d'avoir aucune discontinuité en terme d'accélération, voir Figure 10.5b pour un allure de ce type de profile. Il est à noter que puisque nous avons rajouter 2 paramètres et 2 contraintes, comme pour le profile d'ordre 3 le profil d'ordre 5 a seulement t_f comme paramètre libre.

10.3 Planification cinématique pour les robots manipulateurs



10.3.1 Point à point dans l'espace des joints

Si la position désirée d'un robot est déjà connue dans l'espace de ses joints, l'approche la plus simple est de générer un chemin linéaire dans l'espace des joints (voir Figure 10.6) et ensuite d'ajuster le profil temporel en fonction des limites des actionneurs. L'objectif est donc de calculer des fonctions de trajectoires pour les joints en fonction d'une position actuelle, une position désirée et un temps de trajet :

$$\ddot{q}(t), \dot{q}(t), q(t) = plan(q_0, q_f, t_f) \quad (10.18)$$

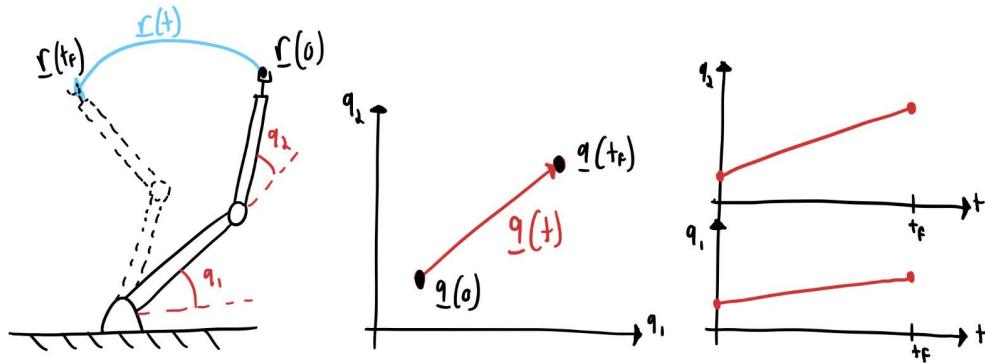


FIGURE 10.6 – Trajectoire en ligne droite dans l'espace de la tâche

Un chemin en ligne droite dans l'espace des joints peut être défini par l'équation :

$$\underline{q}(s) = \underline{q}_0 + s(\underline{q}_f - \underline{q}_0) \quad (10.19)$$

On obtient alors comme dérivés :

$$\frac{\partial \underline{q}}{\partial s} = (\underline{q}_f - \underline{q}_0) \quad (10.20)$$

$$\frac{\partial^2 \underline{q}}{\partial s^2} = 0 \quad (10.21)$$

et la relation entre les vitesses/accélérations des joints et le profil temporel est alors :

$$\dot{q} = (\underline{q}_f - \underline{q}_0)\dot{s} \quad (10.22)$$

$$\ddot{q} = (\underline{q}_f - \underline{q}_0)\ddot{s} \quad (10.23)$$

Donc par exemple si on choisit un profil $s(t)$ trapézoïdale on peut choisir les paramètres v_{max} et a_{max} pour qu'ils correspondent avec les limites physiques des joints :

$$v_{max} (q_i(t_f) - q_i(0)) \leq \max(\dot{q}_i) \quad \forall i \quad (10.24)$$

$$a_{max} (q_i(t_f) - q_i(0)) \leq \max(\ddot{q}_i) \quad \forall i \quad (10.25)$$

ce qui correspond en fait simplement à ajuster pour l'amplitude du déplacement.

Planifier directement dans l'espace des joints a comme avantage de pouvoir vérifier intégralement les contraintes liés aux actionneurs (angles maximums, vitesses et accélérations). L'inconvénient c'est que la trajectoire de l'effecteur du robot n'est pas directement contrôlée.

10.3.2 Point à point dans l'espace de la tâche

Si la tâche se décrit mieux en termes de coordonnées cartésienne de l'effecteur par exemple, il est possible d'appliquer le même principe pour générer directement une trajectoire linéaire pour l'effecteur basé sur une position actuelle, une position cible et un temps de trajet :

$$\ddot{r}(t), \dot{r}(t), r(t) = plan(r_0, r_f, t_f) \quad (10.26)$$

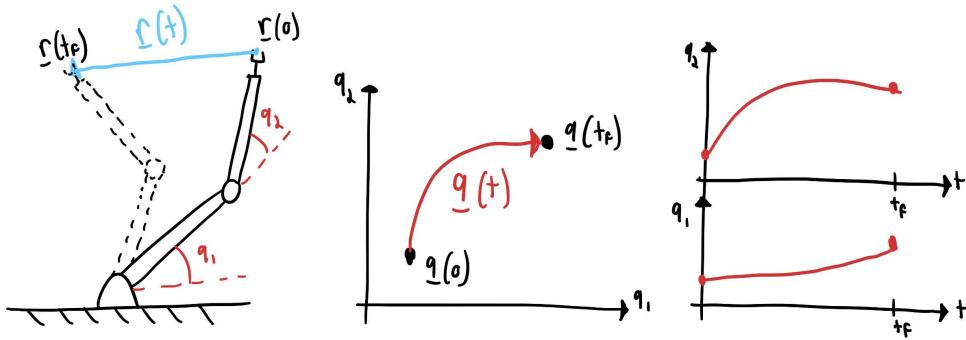


FIGURE 10.7 – Trajectoire en ligne droite dans l'espace de la tâche

Un chemin en ligne droite dans l'espace de la tâche peut être défini par l'équation :

$$\underline{r}(s) = \underline{r}_0 + s(\underline{r}_f - \underline{r}_0) \quad (10.27)$$

On obtient alors la relation suivante entre les vitesses/accélérations de l'effecteur et le profil temporel :

$$\dot{\underline{r}} = (\underline{r}_f - \underline{r}_0)\dot{s} \quad (10.28)$$

$$\ddot{\underline{r}} = (\underline{r}_f - \underline{r}_0)\ddot{s} \quad (10.29)$$

Certaines approches de commande peuvent utiliser directement une trajectoire cible pour l'effecteur, par exemple la loi de commande cinématique présentée à la section 8.2.1, ou la loi de commande en impédance présentée à la section 8.4.2. Alternativement, il est toujours possible de convertir la trajectoire de l'effecteur du robot dans l'espace des joints, avec un modèle cinématique (voir équation (4.86)), pour ensuite l'utiliser avec des lois de commande qui travaillent directement dans l'espace de joints.

Un inconvénient de planifier dans l'espace de la tâche est qu'on peut se retrouver dans des situations où certains points intermédiaires du chemin sont hors de l'espace de travail du robot ou bien proche de singularité, même si le point de départ et la destination sont loin de configurations problématiques. Globalement, par rapport à l'approche d'envoyer directement la position cible dans une loi de commande, les méthodes point à point apportent seulement un meilleur contrôle du profile temporel.

10.3.3 Multi-points dans l'espace de la tâche

Certaines tâches peuvent être décrites en termes de séquence de position à suivre, par exemple le code G pour des machines outils. Aussi, pour des problèmes de planification de chemin dans des environnements avec beaucoup d'obstacles, certains algorithmes de recherche vont retourner une séquence discrétisée de position à suivre pour le robot. Il y a donc plusieurs situations pour lesquelles on doit générer une trajectoire basé sur une séquence de positions intermédiaires :

$$\ddot{r}(t), \dot{r}(t), r(t) = plan(r_0, r_1, r_2, \dots, r_f, t_1, t_2, \dots, t_f) \quad (10.30)$$

Segments

L'approche la plus simpliste est d'utiliser la méthode point à point présentée à la section précédente pour chaque segment, voir Figure 10.8a. Toutefois, l'inconvénient majeur est que pour garantir qu'il n'y a aucune discontinuité dans les vitesses (ce qui mènerait à une accélération infinie), le robot doit s'immobiliser à tout les points où il y a des changements de directions. Ce n'est pas optimal ni en terme de consommation énergétique ni en terme de temps.

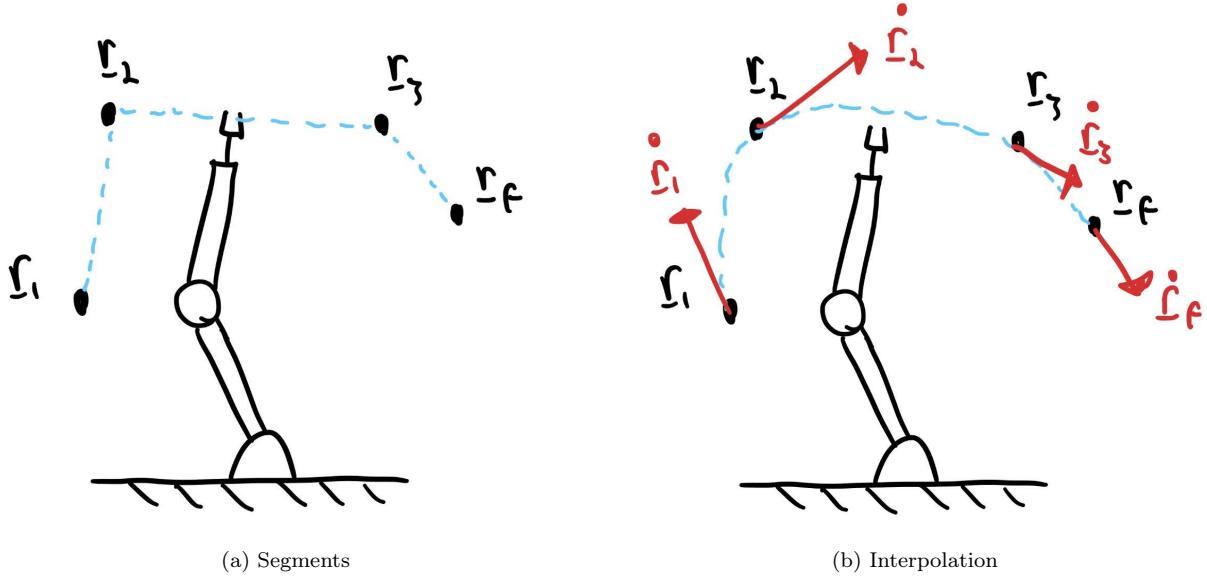


FIGURE 10.8 – Trajectoire multi-points dans l'espace de la tâche

Interpolation

Une alternative est d'utiliser des fonctions d'interpolation pour générer une trajectoire plus lisse. Une approche est d'utiliser des polynômes cubique pour chaque segment, avec des contraintes de continuité en position et vitesse à chaque point intermédiaire. Il est aussi possible de spécifier des vecteurs de vitesses désirées à chaque point intermédiaire, qui peuvent être inclus comme des contraintes pour déterminer les paramètres des polynômes. Il existe plusieurs méthodes d'interpolation qui peuvent être utilisées pour ce type de problème, une autre méthode populaire est les *B-spline*.

10.4 Algorithmes d'optimisation de trajectoires



Capsule vidéo
Introduction à la commande optimale
<https://youtu.be/3x6Vg-RRZ50>



Exercice de code
Démo d'introduction à l'optimisation de trajectoires
https://colab.research.google.com/drive/1yq2GHAKv06fTF2W-tRbACDBa9_scec2k?usp=sharing

À venir !

10.5 Algorithmes de recherche

À venir !

Chapitre 11

Commande bas-niveau des actionneurs

11.1 Introduction

11.2 Comportement des moteurs électriques

11.3 Comportement des vérin pneumatiques

11.4 Actionneur de type série-élastique

À venir !

Deuxième partie

Les véhicules robotisés

Chapitre 12

Modélisation des véhicules

Ce chapitre présente divers modèles cinématiques et dynamiques de véhicules qui sont souvent utilisés dans un contexte de commande et planification. C'est donc ici un survol d'approche de modélisation où un compromis entre fidélité et complexité est effectué. Les approches de modélisation haute fidélité qu'on pourrait utiliser dans un simulateur pour faire de la validation peuvent différer et ne seront pas traité ici.

12.1 Navigation holonomique

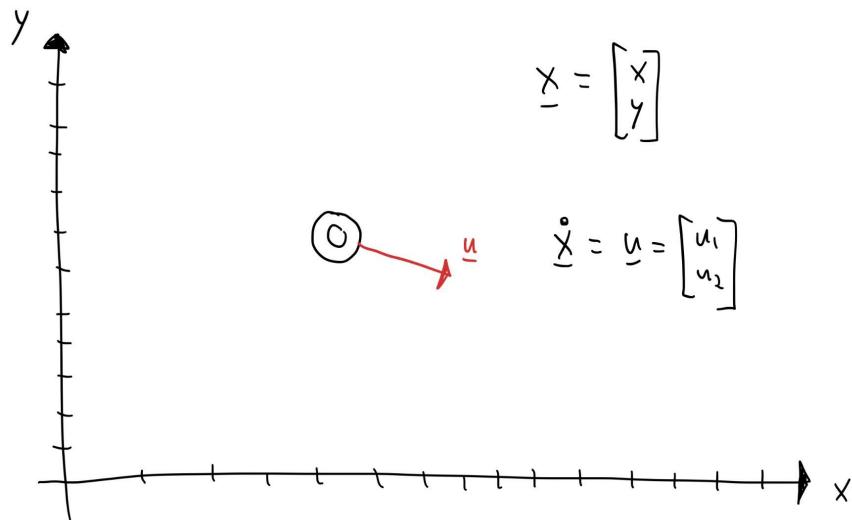


FIGURE 12.1 – Modèle dynamique pour un véhicule holonomique

12.2 Navigation sur un graphe discret

À venir !

12.3 Automobiles

Cette section présente des modèles simples de comportement de véhicules qui ont une architecture de type automobile, c'est-à-dire quatre roues et des roues avant orientable (typiquement une architecture de direction de type ackerman).

12.3.1 Modèle bicyclette

Le modèle souvent appelé bicyclette, est un modèle cinématique simple qui représente bien le comportement d'un véhicule à basse vitesse, et est donc typiquement utilisé pour planifier une trajectoire dans un stationnement par exemple.

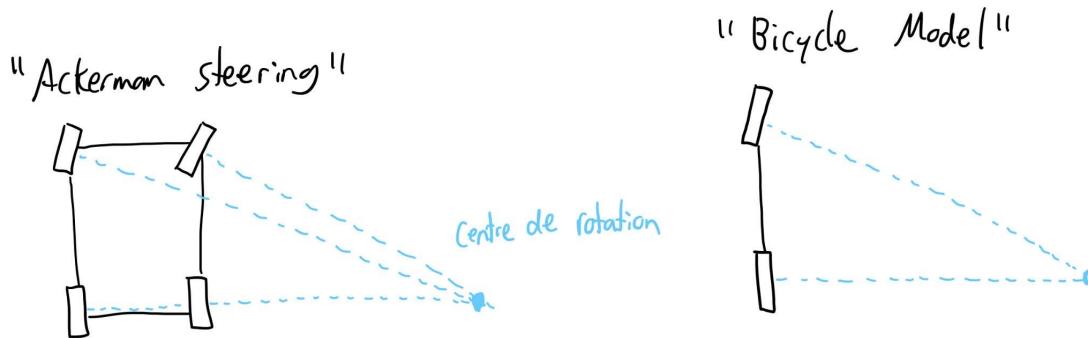


FIGURE 12.2 – Modèle simplifié de type bicyclette

12.3.2 Modèle bicyclette cinématique

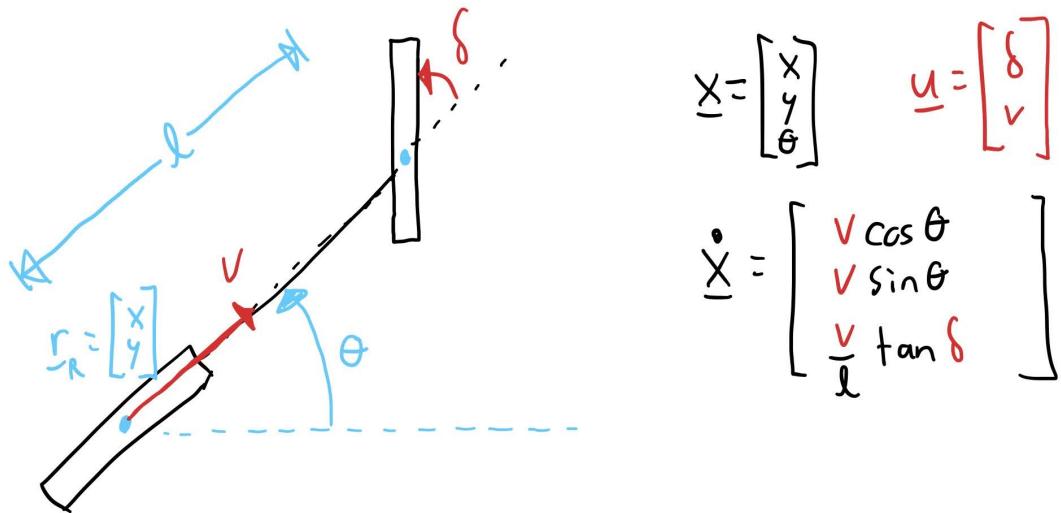


FIGURE 12.3 – Modèle dynamique de type bicyclette cinématique

12.3.3 Modèle bicyclette dynamique

À venir !

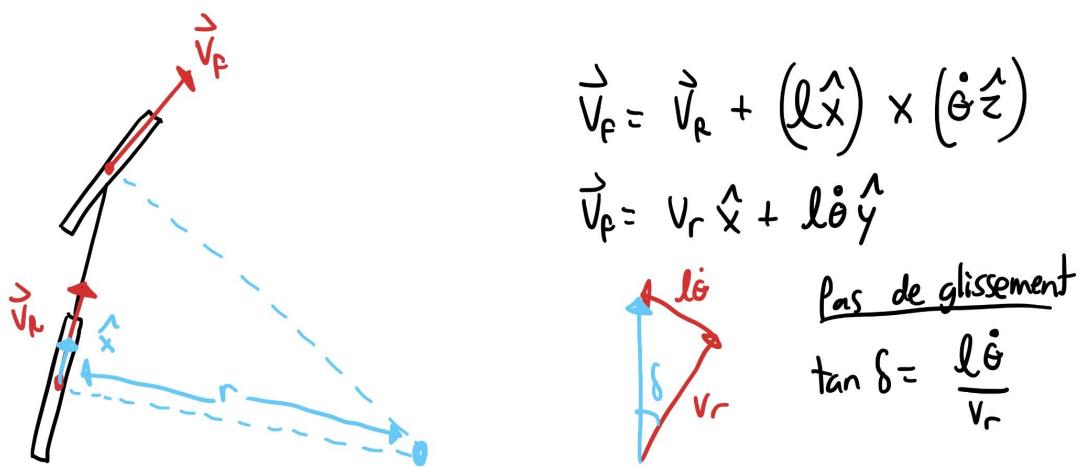


FIGURE 12.4 – Condition pour avoir aucun glissement pour le modèle bicyclette

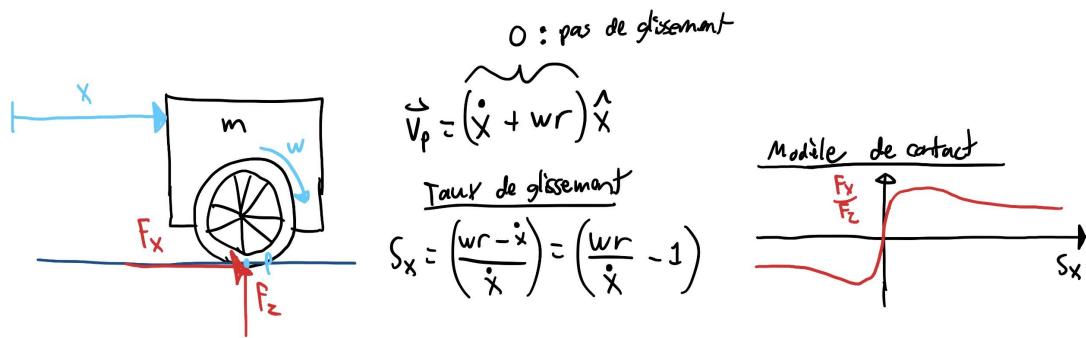


FIGURE 12.5 – Modèle simple de propulsion/freinage longitudinal

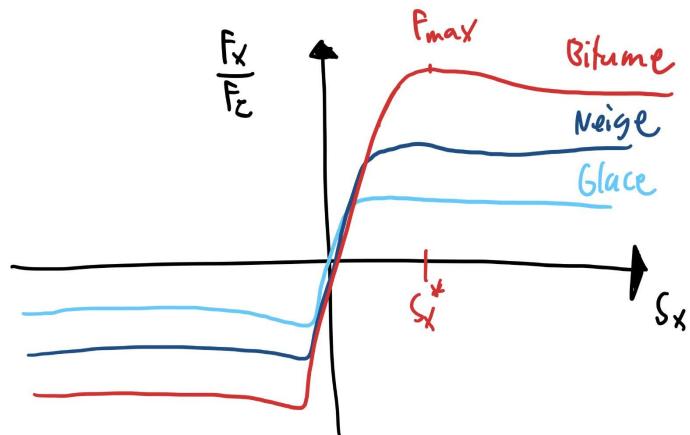


FIGURE 12.6 – Modèle de force de contact pour un pneu

12.3.4 Modèle longitudinal

12.3.5 Modèle avec transfert de poids

12.3.6 Modélisation du contact sol-roue

12.3.7 Modèles de suspension

Quart de véhicule sans pneu

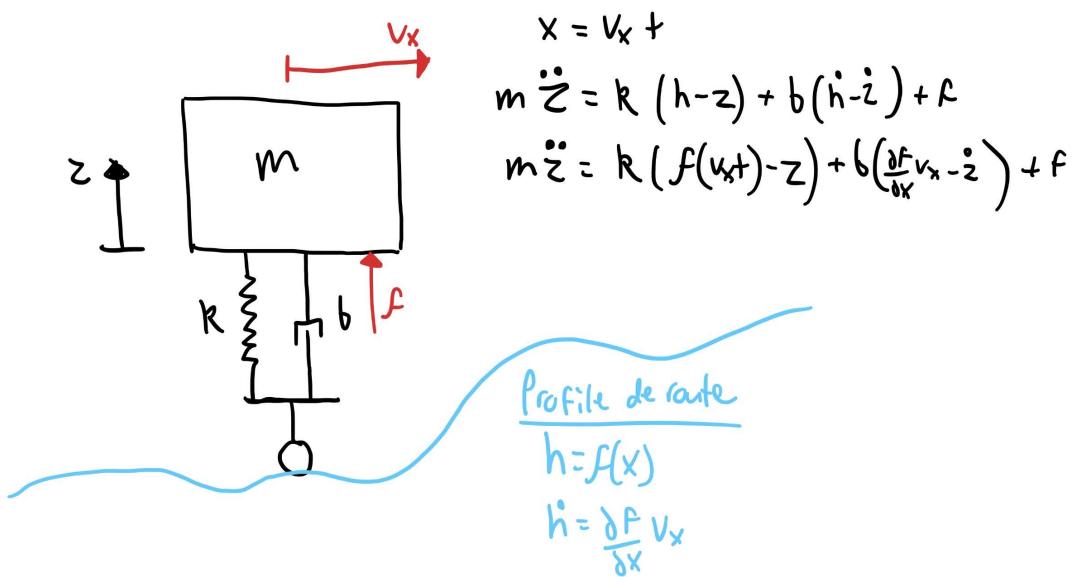


FIGURE 12.7 – Modèle simple de suspension

Quart de véhicule avec pneu

Demi véhicule

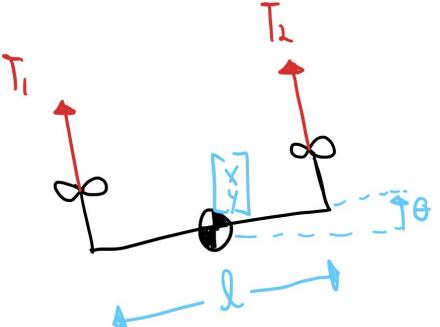
Véhicule complet

12.4 Robot mobiles

Cette section présente des modèles simples pour d'autres architecture de robot mobiles.

12.5 Drones

12.5.1 Modèle de quadrirotor planaire



The diagram shows a planar quadcopter with two rotors at the top and two at the bottom. The center of mass is at the origin. A coordinate system is centered at the origin with axes x and y . The distance from the center of mass to each rotor is $l/2$. The angle of the body frame relative to the fixed frame is θ .

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ -mg \\ 0 \end{bmatrix} = \begin{bmatrix} -s\theta & -s\theta & 0 \\ s\theta & s\theta & 0 \\ -l/2 & l/2 & 0 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ u \end{bmatrix}$$

Handwritten labels: H , \ddot{q} , \underline{g} , $B(q)$, u .

FIGURE 12.8 – Modèle de drone planaire

12.6 Avions

12.7 Fusée

Chapitre 13

Navigation

Chapitre 14

Planification

Troisième partie

Boîte à outils mathématique

Chapitre 15

Méthodes de modélisation

15.1 Types de modèles mathématiques



Capsule vidéo

Types d'équations pour modéliser un système dynamique

https://youtu.be/D_HLuoPrD4w



Capsule vidéo

Grandes familles d'équation différentielles

<https://youtu.be/2ifsprEkJYU>

15.1.2 Équations différentielles partielles

15.1.3 Équations de différences

15.1.4 Chaînes de Markov

15.1.5 Représentation des systèmes hybrides

Hybrid dynamical system can be represented in the general form :

$$\text{Continuous evolution : } (\dot{\underline{x}}, \dot{k}) = (f_k(\underline{x}, \underline{u}, \underline{d}), 0) \quad (15.1)$$

$$\text{Discrete jumps : } (\underline{x}^+, k^+) = (h_{ij}(\underline{x}^-, \underline{u}^-), j) \quad \text{if } (\underline{x}, k, \underline{u}) \in D_{ij} \quad (15.2)$$

where \underline{x} is a continuous state vector, and k is a discrete mode and D_{ij} is the domain mapping conditions leading to a transition $k : i \rightarrow j$. For robotic systems, the discrete mode can represent discrete configurations of the robot , like discrete modes in the control law or contact/non-contact conditions. The jump map then represents the impulsive response when contact is made.

15.1.6 Switched system

A restricted class of hybrid system, called switched system, are hybrid systems for which the jump map for continuous state is the identify function :

$$\text{Continuous evolution : } (\dot{\underline{x}}, \dot{k}) = (f_k(\underline{x}, \underline{u}, \underline{d}), 0) \quad (15.3)$$

$$\text{Discrete jumps : } (\underline{x}^+, k^+) = (\underline{x}^-, j) \quad \text{if } (\underline{x}, k, \underline{u}) \in D_{ij} \quad (15.4)$$

Switched system where the discrete mode is a control input

In the situation where the discrete operating mode k is a control input, then there is no need to keep track of discrete mode evolution and only the piece-wise continuous differential equations are sufficient to model the system evolution :

$$\dot{\underline{x}} = f_k(\underline{x}, \underline{u}, \underline{d}) \quad (15.5)$$

A robot with a continuous dynamics but with a control law using discrete modes of operation would be of this category.

Chapitre 16

Programmation dynamique : La science des prises de décisions optimales

Ce chapitre présente les diverses approches pour prendre des décisions intelligentes sous un cadre théorique unifié basé sur le principe de la programmation dynamique. Elle vise d'abord à établir les liens entre les approches issues du domaine de l'ingénierie (la science des asservissements et la **commande optimale**) et les approches issues des sciences informatiques (recherche opérationnelle et l'**apprentissage par renforcement**) qui ont en fait les **même bases mathématiques**. Ces notes visent principalement à donner à un lecteur issue du domaine de l'ingénierie les bases pour comprendre et utiliser les approches numériques issues des sciences informatiques.



Capsule vidéo

Capsules vidéos sur la programmation dynamique

<https://youtube.com/playlist?list=PL6adNeJOA8UtNs1NQfAHAzcjHcQixBSnu>

16.1 Introduction

La programmation dynamique est un principe mathématique pour optimiser des décisions qui sont prises en séquence après avoir observé l'état d'un système. Le principe peut être utilisé autant pour analyser un système asservis classique, comme contrôler la position d'un moteur en choisissant la tension appliquée à ses bornes, que pour des problèmes probabiliste dans un contexte de finance, comme choisir quand acheter ou vendre une action en observant l'évolution de son prix, ou bien un problème d'IA comme choisir la pièce à déplacer lors d'une partie d'échec en observant la position des pièces sur l'échiquier.

16.1.1 Formulation du problème

Un comportement défini par une loi de commande à concevoir On s'intéresse au problème de concevoir une **une loi de commande**, qui consiste en une fonction c qui détermine l'action u à prendre en fonction de l'état x d'un système observé, et possiblement du temps t :

$$u = c(x, t) \tag{16.1}$$

Cette loi de commande va être implémenté dans un agent qui observe l'état x d'un système et agit en conséquence sur ce système pour l'influencer, formant ainsi une boucle comme illustré à la figure 16.1.

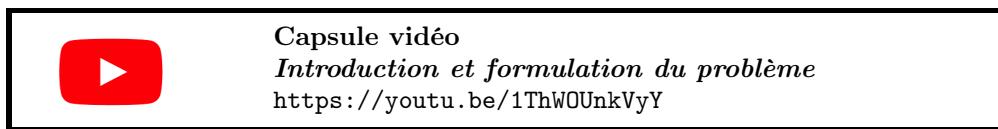
FIGURE 16.1 – La tâche est de concevoir une loi de commande qui définit le comportement d'un agent

Un objectif formulé comme la minimisation d'un coût Ensuite, notre loi de commande sera conçue de sorte à atteindre un objectif qui sera exprimé mathématiquement comme la minimisation d'un coût additif qui dépend de la trajectoire du système et les actions utilisées :

$$J = \int_0^{t_f} g(x, u, t) dt + h(x_f, t_f) \quad (16.2)$$

où g est un coût cumulatif sur la trajectoire du système, h est un coût terminal et t_f est un horizon de temps. La forme cumulative de la fonction coût est centrale pour utiliser le principe de la programmation dynamique, mais ce n'est pas vraiment restrictif car pratiquement tous les objectifs peuvent être formulés comme la minimisation d'une fonction de coût avec cette forme. Lorsque notre agent prend les meilleures décisions possibles en fonction de l'objectif, on dira que sa loi de commande est optimale au sens où elle minimise la fonction de coût.

FIGURE 16.2 – Fonction de coût



Minimiser un coût ou maximiser une récompense ? On peut alternativement formuler l'objectif comme le problème de maximiser une récompense. Les deux approches sont équivalentes et interchangeables. Typiquement, le domaine de la commande optimale utilise la formulation de minimiser un coût, qui est souvent relié à l'erreur par rapport à une trajectoire cible. Alternativement, le domaine de l'apprentissage par renforcement préfère optimiser une récompense, qui est souvent par exemple le pointage dans un jeu pour lequel une IA est développée.

Exemple : loi de commande pour un robot Un exemple d'asservissement classique serait un bras robotique où l'action u déterminée par la loi de commande correspond à un vecteur de couples à appliquer dans les moteurs électriques. Cette action sera calculée en fonction de l'état actuel du robot, donc ici un vecteur de positions et vitesses de ses diverses articulations. L'objectif serait formulé comme la minimisation de l'erreur de position du robot par rapport à une position cible et potentiellement d'une pénalité pour utiliser beaucoup d'énergie. Typiquement notre solution de loi de commande serait ici une équation analytique.

Exemple : navigation d'un véhicule Un exemple de prise de décision à plus haut niveau serait de choisir un trajet sur une carte. La loi de commande déterminerait ici quelle direction prendre en fonction de la position actuelle sur la carte. L'objectif d'atteindre la destination le plus rapidement possible pourrait être formulé comme la minimisation du temps écoulé avant d'atteindre celle-ci. La loi de commande (qui serait une solution globale) pourrait être sous la forme d'une table de correspondance (*look-up table*) où est en mémoire la direction optimale à prendre pour chaque intersection sur laquelle on peut se trouver sur la carte.

Exemple : achats d'une action Un exemple dans un tout autre contexte serait pour un algorithme d'investissement. L'action de la loi de commande serait ici d'acheter ou non une action en fonction d'une observation de son prix. L'objectif pourrait ici être formulé comme la maximisation des gains financiers. La loi de commande serait ici un seuil de prix, qui pourrait varier en fonction du temps, en dessous duquel l'agent décide d'acheter l'action.

16.1.2 Formulation du problème en temps discret

La plupart des outils pour travailler avec ce genre de problèmes sont mieux adapté à une approche de type temps discret. Ces notes vont donc présenter les principes et les algorithmes d'abord avec une approche à temps discret ou un index k indique l'étape actuelle. Il est possible de convertir une problème à temps continu en problème à temps discret en introduisant un pas de temps dt et en considérant que les décisions sont prises en séquence à chaque période de temps dt . Alternativement, une approche pour travailler directement en temps continue est présentée à la section 16.8. De plus, pour plusieurs types de problèmes la nature de l'évolution du système est discrètes, par exemple une partie d'échec. La formulation des problèmes en temps discret est donc très générale et s'applique à un grand nombre de problèmes.

Le problème équivalent à résoudre en temps discret est de déterminer les lois de commande c_k , qui dictent l'action u à prendre lorsque l'état du système est de x à l'étape k :

$$u_k = c_k(x_k) \quad (16.3)$$

de sorte à minimiser un coût additif de la forme :

$$J = \sum_{k=0}^{N-1} g_k(x_k, u_k) + g_N(x_N) \quad (16.4)$$

où N est l'horizon qui représente ici un nombre d'étape. De plus ici l'évolution du système est représentée par une équation de différence :

$$x_{k+1} = f_k(x_k, u_k) \quad (16.5)$$

FIGURE 16.3 – Evolution

Si on reforme tout le problème en une seule équation mathématique :

$$J^*(x_0) = \min_{c_0, \dots, c_{N-1}} \left[\sum_{k=0}^{N-1} g_k(x_k, u_k) + g_N(x_N) \right] \quad \text{avec } x_{k+1} = f_k(x_k, c_k(x_k)) \quad (16.6)$$

on cherche les fonctions c_k , i.e. les loi de commandes, qui vont minimiser le coût cumulatif sur la trajectoire du système, avec l'évolution qui est définie par une dynamique f_k et les lois de commandes c_k .

16.1.3 Terminologie

Variable	Termes	Définition
$f_k(x_k, u_k)$	Dynamique, Système, Environnement, Processus, <i>Plant</i> ,	Équations qui définit l'évolution du système, i.e. de l'environnement de l'agent.
$u = c_k(x)$	Loi de commande, contrôleur, politique de l'agent, <i>policy</i>	Fonction qui définit la décision de l'agent comme une fonction de l'observation de l'état.
x	État, <i>State</i>	Variable qui représente toute l'information pour prédire l'évolution future du système.
u	Action, décision, entrée du système, <i>control input</i>	Variable qui représente la décision de l'agent.
k	index	Entier correspondant à l'étape actuelle (i.e. souvent représentant le temps discréte).
$U_k(x)$	Contraintes, <i>control set</i>	Ensemble représentant les actions qui sont possible lorsque l'état du système est x
$J(x_0, \dots, x_N)$	Fonction de coût cumulative, (opposé de la) fonction de récompense, <i>value fonction</i>	Fonction scalaire qui représente à quel point une trajectoire est bonne en fonction de l'objectif.
$J_c(x)$	Coût à venir	Fonction scalaire qui représente la prédition de coût cummulatif d'une trajectoire qui débute à x en utilisant la loi c
$J^*(x)$	Coût à venir optimal	Fonction scalaire qui représente la prédition de coût cummulatif d'une trajectoire qui débute à x si toutes les actions dans le futur sont optimales
$g_k(x_k, u_k)$	Coût instantanée	Fonction scalaire qui définit le coût instantané à l'étape k
$g_N(x_N)$	Coût terminal	Fonction scalaire qui définit le coût final en fonction de l'état terminal.

16.1.4 Principe d'optimalité

Si une trajectoire est optimale de x_0 à x_N en passant par x_i . Alors nécessairement la séquence de fin de cette trajectoire correspond aussi à la solution optimale d'une trajectoire qui débuterait à x_i pour aller à

x_N .

$$[x_0, x_1, \dots, x_i, \dots, x_N] \quad (16.7)$$

$$[u_0, u_1, \dots, u_i, \dots, u_N] \quad (16.8)$$

$$[x_i, \dots, x_N] \quad (16.9)$$

$$[u_i, \dots, u_N] \quad (16.10)$$



Capsule vidéo
Le principe d'optimalité
<https://youtu.be/EMkpkkyMTg4U>

16.1.5 Programmation dynamique exacte

$$J_N^*(x_N) = g_N(x_N) \quad (16.11)$$

$$J_k^*(x_k) = \min_{u_k \in U_k(x_k)} \left[g_k(x_k, u_k) + J_{k+1}^* \left(\underbrace{f_k(x_k, u_k)}_{x_{k+1}} \right) \right] \quad (16.12)$$

$$c_k^*(x_k) = \operatorname{argmin}_{u_k \in U_k(x_k)} \left[g_k(x_k, u_k) + J_{k+1}^* \left(\underbrace{f_k(x_k, u_k)}_{x_{k+1}} \right) \right] \quad (16.13)$$



Capsule vidéo
L'algorithme de programmation dynamique exacte
<https://youtu.be/dfz9k3BGrHO>

16.1.6 Exemples

Système de chauffage optimal



Capsule vidéo
Exemple de commande optimale pour un système de chauffage
<https://youtu.be/QwXjiAzDENs>

Navigation optimale

16.1.7 Variations sur un thème de programmation dynamique Stochastique

$$J_k^*(x_k) = \min_{u_k} \mathbf{E}_{w_k} \left[g_k(x_k, u_k, w_k) + J_{k+1}^* \left(\underbrace{f_k(x_k, u_k, w_k)}_{x_{k+1}} \right) \right] \quad (16.14)$$

Robuste

$$J_k^*(x_k) = \min_{u_k} \max_{w_k} \left[g_k(x_k, u_k, w_k) + J_{k+1}^* \left(\underbrace{f_k(x_k, u_k, w_k)}_{x_{k+1}} \right) \right] \quad (16.15)$$

À horizon de temps infini

$$J^*(x) = \min_u \left[g(x, u) + \alpha J^* \left(\underbrace{f(x, u)}_{x_{k+1}} \right) \right] \quad (16.16)$$

Sans modèles (apprentissage par renforcement)

$$Q^*(x, u) = g(x, u) + \min_{u_{k+1}} \left[Q^* \left(\underbrace{f(x, u)}_{x_{k+1}}, u_{k+1} \right) \right] \quad (16.17)$$

À temps continu

$$0 = \min_u \left[g(x, u) + \frac{\partial J^*(x, t)}{\partial x} \underbrace{f(x, u, t)}_{\dot{x}} \right] \quad (16.18)$$

16.1.8 Approches de programmation dynamique approximée

16.2 Problèmes déterministes discrets

Lorsqu'on travail avec un problème où les états et actions possibles sont discrets, c'est à dire qu'il peuvent prendre un nombre fini de valeurs. Par exemple, pour une automobile la transmission a généralement autour de 6 options discrètes (marche arrière, neutre, 1ère vitesse, 2e vitesse, etc.) tandis que la commande d'accélération est une variable continue qui peut prendre n'importe quel valeur à l'intérieur d'une certaine plage. Pour les systèmes où tout est discret, il est possible de représenter les états comme des noeuds sur un graphe et les actions possible comme des arcs qui nous mène vers un autre état. Dans ce contexte on peut simplifier la notation.

$$J_k^*(i) = \min_{j \in U(i)} [a_{ij}^k + J_{k+1}^*(j)] \quad (16.19)$$

$$j_k^*(i) = \operatorname{argmin}_{j \in U(i)} [a_{ij}^k + J_{k+1}^*(j)] \quad (16.20)$$

À venir !

Capsule vidéo



Programmation dynamique pour les systèmes discrets et problème du chemin le plus court sur un graphe

<https://youtu.be/QwXjiAzDENs>

Capsule vidéo



Exemple de solution pour le chemin le plus court sur un graphe

<https://youtu.be/1GXUNWVgZOU>

16.3 Problèmes stochastique

On cherche donc ici à minimiser l'espérance du coûts-àvenir :

$$J = E \left\{ \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k) + g_N(x_N) \right\} \quad (16.21)$$

avec l'algorithme de programmation dynamique suivant :

$$J_k^*(x_k) = \min_{u_k} \mathbb{E}_{w_k} \left[g_k(x_k, u_k, w_k) + J_{k+1}^* \left(\underbrace{f_k(x_k, u_k, w_k)}_{x_{k+1}} \right) \right] \quad (16.22)$$

$$u_k^*(x_k) = \operatorname{argmin}_{u_k} \mathbb{E}_{w_k} \left[g_k(x_k, u_k, w_k) + J_{k+1}^* \left(\underbrace{f_k(x_k, u_k, w_k)}_{x_{k+1}} \right) \right] \quad (16.23)$$



Capsule vidéo
Commande stochastique
<https://youtu.be/1GXUNWVgZOU>

16.3.1 Chaînes de Markov



Capsule vidéo
Programmation dynamique pour un processus de Markov
<https://youtu.be/xHoLePda478>

À venir !

16.4 Robustesse et approche minimax

À venir !

16.5 Optimisation sur un horizon de temps infini

$$J = \lim_{N \rightarrow \infty} \mathbb{E} \left[\sum_{k=0}^N \alpha^k g_k \right] \quad (16.24)$$

16.5.1 Équation de Bellman

À venir !



Capsule vidéo
Équation de Bellman
<https://youtu.be/WbpSBaChigQ>



Capsule vidéo
Éléments de l'équation de Bellman
<https://youtu.be/18KrNLHHT3E>



Capsule vidéo
Variantes de l'équation de Bellman
https://youtu.be/98IOSI_jWyY

16.5.2 Méthodes de résolutions



Capsule vidéo
Aperçu des méthodes pour calculer des lois de commandes optimales
<https://youtu.be/jWP9yiIL7gY>

16.5.3 Algorithme d'itération de valeurs (*Value-iteration*)

À venir !



Capsule vidéo
Algorithme d'itération de valeurs
<https://youtu.be/SpWKHB4GupU>

16.5.4 Algorithme d'itération de loi de commande (*policy-iteration*)

À venir !



Capsule vidéo
Algorithme d'itération de politiques
<https://youtu.be/5b1o8808e44>

16.6 Solution LQR

Dans le contexte d'un système dynamique linéaire à états/actions continus représenté par :

$$\underline{x}_{k+1} = f_k(\underline{x}_k, \underline{u}_k, \underline{w}_k) = A_k \underline{x}_k + B_k \underline{u}_k + \underline{w}_k \quad (16.25)$$

où \underline{x}_k et \underline{w}_k sont des vecteurs de dimension n et \underline{u}_k un vecteur de dimension m . Le vecteur \underline{w}_k représente des perturbations aléatoires, indépendantes de l'état actuel, de l'action actuelle et de tous les états passés, et avec des distributions centrées autour de zéro :

$$\mathbb{E}[\underline{w}_k] = 0 \quad (16.26)$$

Si on cherche donc à minimiser l'espérance du coût-àvenir :

$$J = \mathbb{E} \left[\sum_{k=0}^{N-1} \left(\underbrace{\underline{x}_k^T Q_k \underline{x}_k + \underline{u}_k^T R_k \underline{u}_k}_{g_k(\underline{x}_k, \underline{u}_k, \underline{w}_k)} \right) + \underbrace{\underline{x}_N^T Q_N \underline{x}_N}_{g_N(\underline{x}_N)} \right] \quad (16.27)$$

où les matrices Q_k et R_k sont symétriques et définies positives :

$$Q_k \geq 0 \quad R_k > 0 \quad (16.28)$$

En appliquant l'algorithme de programmation dynamique (pour l'étape $N \rightarrow N-1$ ou une étape générique $k+1 \rightarrow k$), on trouve que : 1) le coût-àvenir d'un état \underline{x}_k a la forme quadratique suivante :

$$J_k^*(\underline{x}_k) = \underline{x}_k^T S_k \underline{x}_k + c \quad (16.29)$$

où S_k est une matrice symétrique qui caractérise le coût-àvenir à l'état \underline{x}_k et c est une constante qui ne dépend pas de l'état actuel. 2) la loi de commande optimale a la forme linéaire suivante :

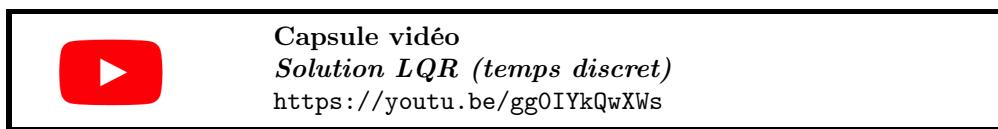
$$\underline{u}_k^* = c_k^*(\underline{x}_k) = -K_k \underline{x}_k \quad (16.30)$$

où K_k est la matrice de gain égale à

$$K_k = [R_k + B_k^T S_{k+1} B_k]^{-1} B_k S_{k+1} A_k \quad (16.31)$$

3) la matrice S_k dans les équations précédentes peut être calculée en partant du coût final à $k = N$ et en remontant dans le temps avec la récursion suivante :

$$S_k = Q_k + A_k^T \left(S_{k+1} - S_{k+1}^T B_k^T [R_k + B_k^T S_{k+1} B_k]^{-1} B_k S_{k+1} \right) A_k \quad (16.32)$$



16.7 Apprentissage par renforcement

À venir!



Capsule vidéo

De la commande optimale à l'apprentissage par renforcement

<https://youtu.be/eyUcnKOpwsE>

16.7.1 TD-Learning

16.7.2 Q-Learning

16.7.3 Sarsa

16.7.4 Approximation de fonctions



Capsule vidéo

Apprentissage par renforcement avec une approximation de fonction

<https://youtu.be/CGbdEDGsJnU>



Capsule vidéo

Méthodes d'approximation de fonctions

<https://youtu.be/p8BiszV8apQ>

16.8 Formulation en temps continu

À venir!

16.8.1 Hamilton–Jacobi–Bellman equation

$$0 = \min_{\underline{u}} \left[g(\underline{x}, \underline{u}) + \frac{\partial J^*(\underline{x}, t)}{\partial \underline{x}} \underbrace{f(\underline{x}, \underline{u}, t))}_{\dot{\underline{x}}} \right] \quad (16.33)$$

Chapitre 17

Calcul vectoriel

Chapitre en construction !

17.1 Les Vecteurs

Un vecteur-géométrique est une quantité physique représentée par **une amplitude et une direction**. Lorsqu'une base vectorielle est spécifiée, un vecteur peut alors être représenté par des composantes scalaires selon chaque vecteur unitaire de la base. Il est important de distinguer la notion de vecteur-géométrique \vec{v} et de ses composantes regroupées dans un vecteur colonne \underline{v} , surtout lorsque plusieurs bases sont utilisés.

La notation suivant est utilisée dans ces notes :

$$\text{Vecteur-géométrique : } \vec{v} = v_1 \hat{a}_1 + v_2 \hat{a}_2 + v_3 \hat{a}_3 \quad (17.1)$$

$$\text{Vecteur-colonne des composantes : } \underline{v}^a = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad (17.2)$$

où l'exposant a est utilisé pour spécifier la base vectorielle associée aux composantes scalaires. Selon les domaines, le terme vecteur réfère soit à la quantité avec une amplitude et une direction (physique, géométrie, etc) soit à une colonne de scalaire (algèbre linéaire, programmation, etc). Comme tous ces domaines sont utilisés en robotique on fera toujours la distinction

17.2 Les bases vectorielles



Capsule vidéo

Les bases vectorielles et composantes d'un vecteur

<https://youtu.be/pZdoFz5PpKU>

Une base vectorielle correspond à trois vecteurs unitaires (en 3D) orthogonaux qui déterminent l'orientation de trois axes dans l'espace et forment une base qui permet de décrire n'importe quel vecteur \vec{v} sous la forme :

$$\vec{v} = v_1^a \hat{a}_1 + v_2^a \hat{a}_2 + v_3^a \hat{a}_3 \quad (17.3)$$

On appellera *base vectorielle a*, une base formée par l'ensemble des vecteurs unitaires $\{\hat{a}_1, \hat{a}_2, \hat{a}_3\}$.

Mathématiquement, les critères de dimensions unitaires et d'orthogonalités sont donnés par les équations :

$$\hat{a}_1 \cdot \hat{a}_1 = 1 \quad \hat{a}_2 \cdot \hat{a}_2 = 1 \quad \hat{a}_3 \cdot \hat{a}_3 = 1 \quad (17.4)$$

$$\hat{a}_1 \cdot \hat{a}_2 = 0 \quad \hat{a}_2 \cdot \hat{a}_3 = 0 \quad \hat{a}_3 \cdot \hat{a}_1 = 0 \quad (17.5)$$

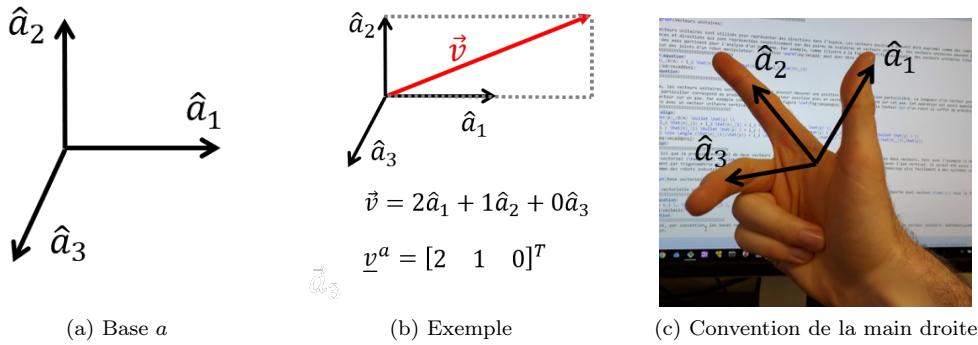


FIGURE 17.1 – Les bases vectorielles

Normalement, par convention, les bases vectorielles suivent la règle de main droite (voir figure 17.1c) pour spécifier la direction du troisième vecteur unitaire. La convention de la main droite permet aussi de spécifier des relations en termes de produits vectoriels entre les vecteurs unitaires :

$$\hat{a}_1 = \hat{a}_2 \times \hat{a}_3 \quad \hat{a}_2 = \hat{a}_3 \times \hat{a}_1 \quad \hat{a}_3 = \hat{a}_1 \times \hat{a}_2 \quad (17.6)$$

Les bases vectorielles permettent de faire des opérations et combiner des vecteurs en les exprimant avec une base commune de vecteurs unitaires. Dans la littérature, la notation $(\hat{i}, \hat{j}, \hat{k})$ ou $(\hat{x}, \hat{y}, \hat{z})$ est parfois utilisée. Dans ces notes, on utilisera des lettres (*a,b,c,...*) pour identifier des bases, ainsi que des indices (1,2,3) pour spécifier les trois axes (voir figure 17.1a).

17.2.1 Relation entre un vecteur et ses composantes dans une base

Comme illustré à la figure 17.2a, il est possible de calculer les composantes d'un vecteur exprimé dans une base par un produit scalaire du vecteur géométrique avec chacun des vecteurs unitaires de la base :

$$v_i^a = \vec{v} \cdot \hat{a}_i \quad \Rightarrow \quad \underline{v}^a = \begin{bmatrix} \vec{v} \cdot \hat{a}_1 \\ \vec{v} \cdot \hat{a}_2 \\ \vec{v} \cdot \hat{a}_3 \end{bmatrix} \quad (17.7)$$

Inversement, comme illustré à la figure 17.2b, le vecteur position géométrique peut être reconstruit en effectuant la somme des composantes multipliées avec leurs vecteurs unitaires respectifs :

$$\vec{v} = \sum_i v_i^a \hat{a}_i = v_1^a \hat{a}_1 + v_2^a \hat{a}_2 + v_3^a \hat{a}_3 \quad (17.8)$$

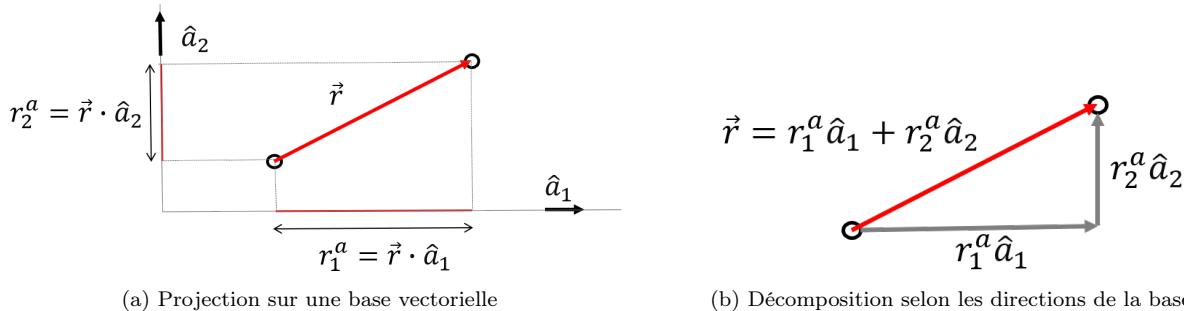


FIGURE 17.2 – Relations entre le vecteur géométrique de position et les composantes du vecteur-colonne

17.2.2 Transfert d'une équation vectorielle vers une équation matricielle

Lorsque les composantes de plusieurs vecteurs sont exprimées avec la même base, il est possible de faire des opérations directement avec les vecteur-colonnes de composantes. Cela permet de traiter simultanément les calculs de plusieurs axes, et aussi de faire des calculs numériques efficaces en utilisant les outils de l'algèbre linéaire. Par exemple, les additions et soustractions de vecteurs géométriques peuvent être calculées directement en termes des vecteur-colonnes dans une base :

$$\vec{u} = \vec{v} + \vec{w} \Rightarrow \underline{u}^a = \underline{v}^a + \underline{w}^a \quad (17.9)$$

Ce transfert d'une seule équation vectorielle à une équation matricielle (équivalent à un système de trois équations scalaires) correspond à une projection de l'équation vectorielle sur chacun des axes de la base :

$$\vec{u} = \vec{v} + \vec{w} \Rightarrow \begin{bmatrix} (\vec{u} = \vec{v} + \vec{w}) \cdot \hat{a}_1 \\ (\vec{u} = \vec{v} + \vec{w}) \cdot \hat{a}_2 \\ (\vec{u} = \vec{v} + \vec{w}) \cdot \hat{a}_3 \end{bmatrix} \Rightarrow \begin{bmatrix} u_1^a = v_1^a + w_1^a \\ u_2^a = v_2^a + w_2^a \\ u_3^a = v_3^a + w_3^a \end{bmatrix} \Rightarrow \underline{u}^a = \underline{v}^a + \underline{w}^a \quad (17.10)$$

Les équations vectorielles avec des vecteurs de position peuvent donc être substituées par des équations matricielles équivalentes avec les vecteur-colonnes, **à condition que tous les vecteur-colonnes soient exprimés dans une base commune**.

17.3 Opérations vectorielles avec les vecteur-colonnes

Les opérations vectorielles ont des équivalents en termes d'opérations matricielles avec les vecteur-colonnes qui sont très utiles pour les calculs numériques. Cette section introduit les opérations principales utiles pour la cinématique.



Capsule vidéo
Opérations vectorielles avec les composantes de vecteurs
<https://youtu.be/7brvShmayAk>

17.3.1 Produit scalaire / produit intérieur

Un produit scalaire peut être calculé directement par une opération matricielle avec les vecteur-colonnes appelée produit intérieur (*inner product*) :

$$\vec{u} \cdot \vec{v} = \underline{u}^T \underline{v} = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = u_1 v_1 + u_2 v_2 + u_3 v_3 \quad (17.11)$$

si \underline{u} et \underline{v} sont les composantes exprimées dans une base vectorielle commune, par exemple :

$$\vec{u} \cdot \vec{v} = (\underline{u}^a)^T \underline{v}^a = (\underline{u}^b)^T \underline{v}^b \quad (17.12)$$

mais attention car :

$$\vec{u} \cdot \vec{v} \neq (\underline{u}^a)^T \underline{v}^b \quad (17.13)$$

$$\vec{u} \cdot \vec{v} \neq (\underline{u}^b)^T \underline{v}^a \quad (17.14)$$

Il est à noter que l'ordre de multiplication ne change pas le résultat pour le produit scalaire :

$$\vec{u} \cdot \vec{v} = \vec{v} \cdot \vec{u} = \underline{u}^T \underline{v} = \underline{v}^T \underline{u} \quad (17.15)$$

17.3.2 Produit vectoriel

Un produit vectoriel (*cross product*) peut être calculé directement par une opération matricielle définie avec les vecteur-colonnes :

$$\vec{u} = \vec{v} \times \vec{w} \Rightarrow \underline{u} = \underline{v}^\times \underline{w} \Rightarrow \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \quad (17.16)$$

L'ordre de multiplication est important pour le produit vectoriel, il change la direction du vecteur résultant :

$$\vec{u} = \vec{v} \times \vec{w} = -\vec{w} \times \vec{v} \Rightarrow \underline{u} = \underline{v}^\times \underline{w} = -\underline{w}^\times \underline{v} \quad (17.17)$$

Vecteurs unitaires d'une base vectorielle : Il est possible de calculer les composantes d'un vecteur unitaire qui complète une base vectorielle avec une opération de produit vectoriel, connaissant les composantes de deux vecteurs unitaires orthogonaux :

$$\hat{b}_3 = \hat{b}_1 \times \hat{b}_2 \Rightarrow \underline{b}_3 = \underline{b}_1^\times \underline{b}_2 \quad (17.18)$$

Note : Pour alléger la présentation, les exposants qui spécifient les bases vectorielles associées aux vecteur-colonnes sont parfois omis dans les équations, on considère alors que tous les vecteur-colonnes impliqués dans l'équation sont exprimés dans une base vectorielle commune.

17.3.3 Produit tensoriel / produit extérieur

Un produit tensoriel (*outer-product*) peut être calculé directement par une opération matricielle définie avec les vecteur-colonnes :

$$\vec{\vec{T}} = \vec{v} \vec{w} \Rightarrow T = \underline{v} \underline{w}^T = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} = \begin{bmatrix} v_1 w_1 & v_1 w_2 & v_1 w_3 \\ v_2 w_1 & v_2 w_2 & v_2 w_3 \\ v_3 w_1 & v_3 w_2 & v_3 w_3 \end{bmatrix} \quad (17.19)$$

où le matrice T correspond aux composantes du tenseur $\vec{\vec{T}}$, avec comme correspondance si toutes les composantes sont relatives à une base a :

$$\vec{\vec{T}} = T_{11}\hat{a}_1\hat{a}_1 + T_{12}\hat{a}_1\hat{a}_2 + \dots + T_{23}\hat{a}_2\hat{a}_3 + T_{33}\hat{a}_3\hat{a}_3 \quad (17.20)$$

17.3.4 Changement de base

Les composantes d'un vecteur géométrique \vec{v} dans une base a peuvent être directement reliées aux composantes du même vecteur géométrique \vec{v} dans une autre base b , par une équation matricielle qui implique les vecteur-colonnes et une matrice 3×3 appelée une *matrice de rotation*. On va noter une matrice ${}^b R^a$ lorsqu'une multiplication par la gauche de cette matrice avec un vecteur colonne \underline{v}^a donne le vecteur-colonne \underline{v}^b :

$$\underline{v}^b = {}^b R^a \underline{v}^a \Rightarrow \begin{bmatrix} v_1^b \\ v_2^b \\ v_3^b \end{bmatrix} = \begin{bmatrix} {}^b R_{11}^a & {}^b R_{12}^a & {}^b R_{13}^a \\ {}^b R_{21}^a & {}^b R_{22}^a & {}^b R_{23}^a \\ {}^b R_{31}^a & {}^b R_{32}^a & {}^b R_{33}^a \end{bmatrix} \begin{bmatrix} v_1^a \\ v_2^a \\ v_3^a \end{bmatrix} \quad (17.21)$$

Les propriétés et le calcul des matrices de rotation seront traités en détails à la section 3.5.

17.3.5 Invariants

Le vecteur géométrique \vec{v} est une quantité constante peu importe la base vectorielle utilisée :

$$\vec{v} = \sum_i v_i^a \hat{a}_i = \sum_i v_i^b \hat{b}_i = \sum_i v_i^c \hat{c}_i \quad (17.22)$$

mais le vecteur-colonne dépend de la base vectorielle. Les vecteur-colonnes sont en générale différents sauf dans le cas particulier de bases vectorielles coïncidentes :

$$\underline{v}^a \neq \underline{v}^b \quad \text{sauf dans le cas particulier : } \hat{a}_1 = \hat{b}_1, \hat{a}_2 = \hat{b}_2 \text{ et } \hat{a}_3 = \hat{b}_3 \quad (17.23)$$

La longueur du vecteur-géométrique étant une constante peut importe la base vectorielle utilisée, il est possible d'établir la relation suivante entre les vecteur-colonnes utilisant différentes bases :

$$\|\vec{v}\|^2 = \vec{v} \cdot \vec{v} = (\underline{v}^a)^T \underline{v}^a = (\underline{v}^b)^T \underline{v}^b = (\underline{v}^c)^T \underline{v}^c \quad (17.24)$$

La norme des vecteur-colonnes est donc invariante par rapport au choix de la base.

17.4 Summary of vector operations in terms of components

Vector and Tensors	Matrix, Row and Columns	Components and index
Definitions		
Vector \vec{v} and Tensor \vec{T} $\vec{v} = x_1 \vec{i} + x_2 \vec{j} + x_3 \vec{k}$ $\vec{T} = x_{11} \vec{i}\vec{i} + x_{12} \vec{i}\vec{j} + \dots + x_{33} \vec{k}\vec{k}$	Column \underline{c} , Row \underline{r} and Matrix M $\underline{c} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ $\underline{r} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}$ $M = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix}$	Scalar components x_i and x_{ij} $x_i \quad i \in \{1, 2, 3\}$ $x_{ij} \quad i \in \{1, 2, 3\}, j \in \{1, 2, 3\}$
Dot or inner product		
$z = \vec{x} \cdot \vec{y}$	$z = \underline{x}^T \underline{y} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$	$z = \sum_{ij} x_i y_j \delta_{ij} = \sum_i x_i y_i$
$\vec{z} = \vec{A} \cdot \vec{x}$	$\underline{z} = A \underline{x} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$	$z_i = \sum_{jk} a_{ij} x_k \delta_{jk} = \sum_j a_{ij} x_j$
$\vec{z} = \vec{x} \cdot \vec{A}$	$\underline{z} = \underline{x}^T A$	$z_j = \sum_{ik} x_k a_{ij} \delta_{ki} = \sum_i x_i a_{ij}$
Dyadic or outer product		
$\vec{Z} = \vec{x} \vec{y}$	$Z = \underline{x} \underline{y}^T = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix}$	$z_{ij} = x_i y_j$
Cross product		
$\vec{z} = \vec{x} \times \vec{y}$	$z = \underline{x} \times \underline{y} = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$	$z_i = \sum_{jk} \epsilon_{ijk} x_j y_k$

17.4.1 Index notation special symbols

Kronecker delta

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \quad (17.25)$$

Levi-Civita permutation symbol

$$\epsilon_{ijk} = \begin{cases} +1 & \text{if } ijk = (123), (231) \text{ or } (312) \\ -1 & \text{if } ijk = (321), (213) \text{ or } (132) \\ 0 & \text{if an index is repeated} \end{cases} \quad (17.26)$$

17.5 Column-vector and matrix differentiation

Note that the shape of vector/matrix resulting from a multi-axis differentiation is a question of layout convention. The numerator layout convention is used here. Identities presented need to be transposed if using a denominator layout instead. There is no ambiguity with the index representation.

Scalar by Scalar

For a scalar function $y = f(x)$:

$$z = \frac{\partial y}{\partial x} \quad (17.27)$$

Vector by Scalar

For the derivation of a vector function $\underline{y} = f(\underline{x})$ where $\underline{y} \in \mathbb{R}^n$ with respect to a scalar x , by convention if the numerator \underline{y} is a $n \times 1$ column vector, the result is a $n \times 1$ column vector too :

$$\underline{z} = \frac{\partial \underline{y}}{\partial x} = \begin{bmatrix} \frac{\partial y_1}{\partial x} \\ \vdots \\ \frac{\partial y_n}{\partial x} \end{bmatrix} \Leftrightarrow z_i = \frac{\partial y_i}{\partial x} \quad (17.28)$$

Scalar by Vector

For the gradient of a multi-inputs scalar function $y = f(\underline{x})$ where $\underline{x} \in \mathbb{R}^n$, by convention if the denominator \underline{x} is a $n \times 1$ column vector, the result is a $1 \times n$ row vector :

$$\underline{z} = \frac{\partial y}{\partial \underline{x}} = \left[\begin{array}{ccc} \frac{\partial y}{\partial x_1} & \dots & \frac{\partial y}{\partial x_n} \end{array} \right] \Leftrightarrow z_i = \frac{\partial y}{\partial x_i} \quad (17.29)$$

TABLE 17.1 – Scalar by a vector : Identities

Scalar $y = f(\underline{x})$ expression	Gradient Vector $\frac{\partial y}{\partial \underline{x}}$	Notes
$\underline{a}^T \underline{x} = \underline{x}^T \underline{a}$	\underline{a}^T	If \underline{a} is not a function of \underline{x}
$\underline{x}^T \underline{x}$	$2 \underline{x}^T$	
$\underline{x}^T A \underline{x}$	$\underline{x}^T (A + A^T)$	If A is not a function of \underline{x}
$\underline{x}^T A \underline{x}$	$2 \underline{x}^T A$	If A is symmetric and not a function of \underline{x}

Vector by Vector

For the gradient of a multi-inputs vector function $\underline{y} = f(\underline{x})$ where $\underline{x} \in \mathbb{R}^n$ and $\underline{y} \in \mathbb{R}^m$, by convention if the numerator \underline{y} is a $m \times 1$ column vector and the denominator a $n \times 1$ column vector, the result is a $m \times n$ matrix, often called the Jacobian matrix :

$$Z = \frac{\partial \underline{y}}{\partial \underline{x}} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \dots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \dots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \dots & \frac{\partial y_m}{\partial x_n} \end{bmatrix} \Leftrightarrow z_{ij} = \frac{\partial y_i}{\partial x_j} \quad (17.30)$$

TABLE 17.2 – Vector by a vector : Identities

Vector $\underline{y} = f(\underline{x})$ expression	Jacobian Matrix $\frac{\partial \underline{y}}{\partial \underline{x}}$	Notes
\underline{x}	I	Identity Matrix
$A\underline{x}$	A	If A is not a function of \underline{x}
$\underline{x}^T A$	A^T	If A is not a function of \underline{x}

Matrix by Scalar

For the gradient of a matrix function $\underline{A} = f(x)$ where x is a scalar and A is a $n \times m$ matrix, the result is also a $n \times m$ matrix :

$$Z = \frac{\partial \underline{A}}{\partial x} = \begin{bmatrix} \frac{\partial a_{11}}{\partial x} & \dots & \frac{\partial a_{1m}}{\partial x} \\ \vdots & \dots & \vdots \\ \frac{\partial a_{n1}}{\partial x} & \dots & \frac{\partial a_{nm}}{\partial x} \end{bmatrix} \Leftrightarrow z_{ij} = \frac{\partial a_{ij}}{\partial x} \quad (17.31)$$

TABLE 17.3 – Matrix by scalar : Identities

Matrix A expression	Differential Matrix $\frac{\partial A}{\partial x}$	Notes
M^{-1}	$-M^{-1} \frac{\partial M}{\partial x} M^{-1}$	

Scalar by Matrix

For the gradient of a multi-inputs scalar function $y = f(A)$ where A is an $n \times m$ matrix, by convention the result is a $m \times n$ matrix :

$$Z = \frac{\partial y}{\partial \underline{A}} = \begin{bmatrix} \frac{\partial y}{\partial a_{11}} & \dots & \frac{\partial y}{\partial a_{1n}} \\ \vdots & \dots & \vdots \\ \frac{\partial y}{\partial a_{m1}} & \dots & \frac{\partial y}{\partial a_{mn}} \end{bmatrix} \Leftrightarrow z_{ji} = \frac{\partial y}{\partial a_{ij}} \quad (17.32)$$

TABLE 17.4 – Scalar by Matrix : Identities

Scalar y expression	Differential Matrix $\frac{\partial y}{\partial A}$	Notes
$\underline{a}^T A \underline{b}$	$\underline{b} \underline{a}^T$	\underline{a} and \underline{b} are not function of A
$\underline{a}^T A^T A \underline{b}$	$\underline{a} \underline{b}^T A^T + \underline{b} \underline{a}^T A^T$	\underline{a} and \underline{b} are not function of A

Matrix by Vector

For the gradient of a matrix function $\underline{A} = f(\underline{x})$ where \underline{x} is a $p \times 1$ column-vector and A is a $n \times m$ matrix, the result is a $n \times m \times p$ tensor :

$$T = \frac{\partial A}{\partial \underline{x}} = \begin{bmatrix} \frac{\partial a_{11}}{\partial \underline{x}} & \dots & \frac{\partial a_{1m}}{\partial \underline{x}} \\ \vdots & \dots & \vdots \\ \frac{\partial a_{n1}}{\partial \underline{x}} & \dots & \frac{\partial a_{nm}}{\partial \underline{x}} \end{bmatrix} \Leftrightarrow z_{ijk} = \frac{\partial a_{ij}}{\partial x_k} \quad (17.33)$$

Exemple 1. Gradient of Quadratic Function

Finding the gradient $\frac{\partial f}{\partial \underline{x}}$ of the quadratic function $f = \underline{x}^T A \underline{x}$. First converting to index notations :

$$f = \sum_{ij} x_i A_{ij} x_j \quad (17.34)$$

Then differentiating per x_k components :

$$\frac{\partial f}{\partial x_k} = \sum_{ij} \frac{\partial}{\partial x_k} (x_i A_{ij} x_j) \quad (17.35)$$

$$\frac{\partial f}{\partial x_k} = \sum_{ij} \left(\underbrace{\frac{\partial x_i}{\partial x_k} A_{ij} x_j}_{\delta_{ik}} + x_i \underbrace{\frac{\partial A_{ij}}{\partial x_k} x_j}_{0} + x_i A_{ij} \underbrace{\frac{\partial x_j}{\partial x_k}}_{\delta_{jk}} \right) \quad (17.36)$$

$$\frac{\partial f}{\partial x_k} = \sum_{ij} \delta_{ik} A_{ij} x_j + \sum_{ij} \delta_{jk} x_i A_{ij} \quad (17.37)$$

$$\frac{\partial f}{\partial x_k} = \sum_j A_{kj} x_j + \sum_i x_i A_{ik} \quad (17.38)$$

$$\frac{\partial f}{\partial x_k} = \sum_i (A_{ki} x_i + x_i A_{ik}) \quad (17.39)$$

$$(17.40)$$

Which correspond in matrix form, with the numerator layout convention, to :

$$\frac{\partial f}{\partial \underline{x}} = \underline{x}^T (A + A^T) \quad (17.41)$$

Exemple 2. Least square solution derivation in vector form

Minimizing the error norm $\|\underline{e}\|^2$ of the estimation problem $\underline{e} = A\underline{\hat{x}} - \underline{b}$ by selecting the parameters $\underline{\hat{x}}$. If the gradient of the error norm with respect to the parameter vector is set to zero :

$$\underline{0} = \frac{\partial \|\underline{e}\|^2}{\partial \underline{\hat{x}}} = \frac{\partial (\underline{e}^T \underline{e})}{\partial \underline{\hat{x}}} = 2 \underline{e}^T \frac{\partial \underline{e}}{\partial \underline{\hat{x}}} = 2 (A\underline{\hat{x}} - \underline{b})^T A \quad (17.42)$$

For which the solution is

$$\underline{\hat{x}} = [A^T A]^{-1} A^T \underline{b} \quad (17.43)$$

Chapitre 18

Algèbre linéaire

18.1 Opérations matricielles

L'algèbre linéaire, c'est à la base une collection d'outils mathématique pour traiter plusieurs fonctions/opérations en parallèle.

18.1.1 L'opération $y = A\underline{x}$

L'opération $y = Ax$ est un outil mathématique pour faire m opérations linéaires sur les variables x_i dans le vecteur-colonne x . Les outils de l'algèbre linéaire sont donc particulièrement utiles pour travailler avec des quantités multivariées comme des vecteurs 3D de position, vitesse, forces, etc.

$$A \in \mathbb{R}^{m \times n} \quad x \in \mathbb{R}^n \quad y \in \mathbb{R}^m$$

$n = \# \text{ colonnes} = \# \text{ variables}$

$m = \# \text{ rangées} = \# \text{ équations}$

$$y_j = \sum_i^n A_{ji} x_i$$

FIGURE 18.1 – L'opération matricielle $y = A\underline{x}$

Système d'équations

L'opération matricielle $\underline{y} = A\underline{x}$ peut d'abord être vue comme un système de m équations linéaires. Par exemple, deux fonctions linéaires, f_1 et f_2 , de deux variables (entrées), x_1 et x_2 , peuvent s'écrire :

$$y_1 = f_1(x_1, x_2) = a_{11}x_1 + a_{12}x_2 \quad (18.1)$$

$$y_2 = f_2(x_1, x_2) = a_{21}x_1 + a_{22}x_2 \quad (18.2)$$

ou y_1 et y_2 sont les variables résultats (sorties) des fonctions, et a_{ii} les paramètres internes des fonctions. Ces équations, avec la notation matricielle prennent la forme :

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leftrightarrow \underline{y} = A\underline{x} \quad (18.3)$$

ou \underline{y} et \underline{x} sont des vecteur-colonnes de dimension 2 et A est une matrice 2x2. De façon générale, une matrice d'un tel système d'équation a une largeur n , correspondant au nombre de variable dans les équations, et une hauteur m , correspondant au nombre d'équations.

$$\begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad (18.4)$$

$$m : \text{nombre de rangs} = \text{nombre d'équations / sorties} \quad (18.5)$$

$$n : \text{nombre de colonnes} = \text{nombre de variables / entrées} \quad (18.6)$$

Combinaison de vecteur-colonnes

La multiplication d'une matrice par un vecteur-colonne peut être vue comme une combinaison linéaire des vecteur-colonnes de la matrice. Cette interprétation est très importante pour comprendre toutes les propriétés importantes et utiles des matrices, qui seront détaillées dans les prochaines sections. L'équation (18.3), correspondant à la multiplication d'un vecteur-colonne de dimension 2 par une matrice 2×2 peut être réécrite comme l'addition pondérée des deux colonnes de la matrice :

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leftrightarrow \underline{y} = \underbrace{\begin{bmatrix} a_{11} \\ a_{21} \end{bmatrix}}_{\underline{c}_1} x_1 + \underbrace{\begin{bmatrix} a_{12} \\ a_{22} \end{bmatrix}}_{\underline{c}_2} x_2 \quad (18.7)$$

Graphiquement, tel illustré à la figure 18.2, la multiplication d'une matrice peut être représenté par une addition pondérée (combinaison linéaire) des vecteurs correspondants aux colonnes de la matrice, dans l'espace sortie $\underline{y} \in \mathbb{R}^m$.

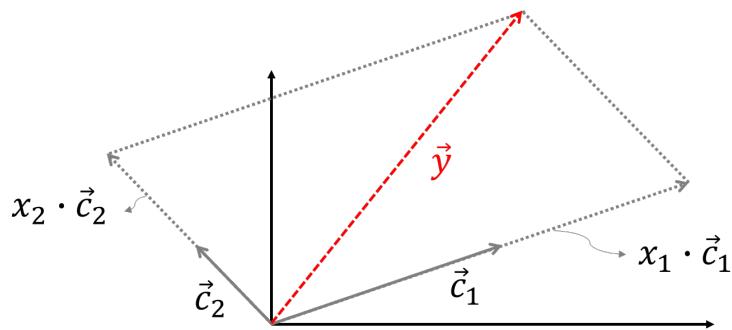


FIGURE 18.2 – La multiplication matricielle comme une combinaison linéaire de vecteurs

De façon générale, le vecteur-colonne \underline{y} résultant d'une multiplication d'une matrice A avec un vecteur-colonne \underline{x} , peut-être exprimée comme une combinaison linéaire des colonnes de la matrice A :

$$\begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} a_{11} \\ \vdots \\ a_{1m} \end{bmatrix} & \cdots & \begin{bmatrix} a_{i1} \\ \vdots \\ a_{im} \end{bmatrix} & \cdots & \begin{bmatrix} a_{n1} \\ \vdots \\ a_{nm} \end{bmatrix} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad (18.8)$$

$$\underline{y} = \sum_{i=1}^n \underline{c}_i x_i \quad (18.9)$$

Produits scalaires avec les rangés de la matrice

Une autre interprétation de l'opération matricielle $\underline{y} = Ax$ est que chaque élément y_j du résultat correspond au produit scalaire du vecteur-rangé j avec le vecteur colonne \underline{x} . L'équation (18.3) peut prendre la forme :

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \underbrace{\begin{bmatrix} a_{11} & a_{12} \end{bmatrix}}_{\underline{r}_1} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ \underbrace{\begin{bmatrix} a_{21} & a_{22} \end{bmatrix}}_{\underline{r}_2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \vec{r}_1 \cdot \vec{x} \\ \vec{r}_2 \cdot \vec{x} \end{bmatrix} \quad (18.10)$$

De façon générale, chaque élément du y_j vecteur-colonne \underline{y} résultant d'une multiplication d'une matrice A avec un vecteur-colonne \underline{x} , peut-être exprimé comme le résultat du produit scalaire entre le vecteur-rangé \underline{r}_j et le vecteur-colonne \underline{x} :

$$y_j = \underline{r}_j \underline{x} = \sum_i a_{ji} x_i \quad (18.11)$$

Opérations matricielles en termes d'indices et de composantes

Finalement, toutes les opérations matricielles peuvent être écrite comme une équation avec des sommations et des indices. Par exemple, une matrice multipliée par un vecteur-colonne :

$$\underline{y} = Ax \Leftrightarrow y_j = \sum_i A_{ji} x_i \quad (18.12)$$

Calcul sur un ordinateur

À bas niveau, un ordinateur effectue des opérations de sommation sur les divers composantes pour faire une opération matricielle. Toutefois, plusieurs langages de programmation, comme *Python* (avec *Numpy*) et *Matlab*, donnent accès à des fonctions pour demander des opérations matricielles à haut-niveau, avec un code source très optimisé en terme d'utilisation de la mémoire de l'ordinateur. Il est donc beaucoup plus rapide d'utiliser de telles fonctions que de coder ces opérations de sommation manuellement. De plus, les cartes graphiques des ordinateurs ont des circuits spécialement conçu pour faire des opérations spécialisées typiques d'algèbre linéaire. Pour faire des calculs avec des très grandes matrices et long vecteurs, il peut y avoir des gains de temps très significatifs en utilisant les capacités des cartes graphiques.

18.1.2 Multiplication de matrices $C = AB$

$C = AB$ en termes d'indices et de composantes

Une matrice multipliée par une autre matrice est équivalent en termes de sommation de composante à l'opération :

$$A = BC \Leftrightarrow A_{ik} = \sum_j B_{ij} C_{jk} \quad (18.13)$$

18.1.3 L'opération matricielle A^T

Prendre la transposée d'une matrice est une opération qui consiste à inverser les indices i et j de la matrice, ce qu'on peut aussi voir comme une réflexion selon la diagonale de la matrice :

$$A_{ij}^T = A_{ji} \quad (18.14)$$

18.1.4 L'opération matricielle A^{-1}

À venir !

Formule pour une matrice 2x2

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \underbrace{\frac{1}{ad - bc}}_{\text{Déterminant de } A} \underbrace{\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}}_{\text{Adjoint de } A} \quad (18.15)$$

18.1.5 Additions de matrices $C = A + B$

Une matrice additionnée à une autre matrice est équivalent en termes de composantes à additionner les termes individuellement :

$$C = A + B \Leftrightarrow C_{ij} = A_{ij} + B_{ij} \quad \forall(i, j) \quad (18.16)$$

18.1.6 Matrice multipliée par un scalaire $B = cA$

Une matrice multipliée par un scalaire est équivalent en termes de composantes à multiplier chaque composante par le scalaire :

$$B = cA \Leftrightarrow B_{ij} = cA_{ij} \quad \forall(i, j) \quad (18.17)$$

18.1.7 Lois pour les opérations de matrices

Additions et soustractions :

$$A + B = B + A \quad (18.18)$$

$$c(A + B) = cA + cB \quad (18.19)$$

$$(A + B) + C = A + (B + C) \quad (18.20)$$

$$AB \neq BA \quad \text{en général} \quad (18.21)$$

$$C(A + B) = CA + CB \quad (18.22)$$

$$(A + B)C = AC + BC \quad (18.23)$$

$$A(BC) = (AB)C \quad (18.24)$$

Transposition et inversions :

$$(A + B)^T = A^T + B^T \quad (18.25)$$

$$(AB)^T = B^T A^T \quad (18.26)$$

$$(AB)^{-1} = B^{-1} A^{-1} \quad (18.27)$$

$$(A^{-1})^T = (A^T)^{-1} \quad (18.28)$$

18.2 Les quatre espaces fondamentaux

La figure 18.3 résume les quatre espaces fondamentaux d'une matrice, deux espaces caractérisent les entrées \underline{x} et deux espaces caractérisent les sorties \underline{y} . La figure 18.4 illustre comment les dimensions de ces espaces sont reliés au rang et dimensions d'une matrice.

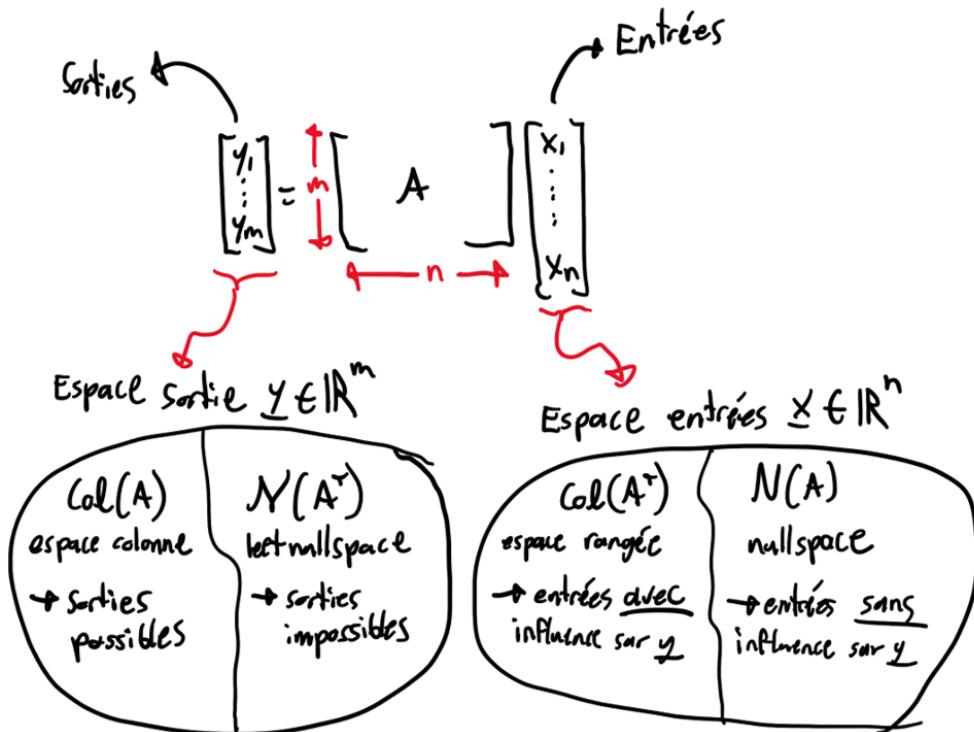


FIGURE 18.3 – Quatres espaces fondamentaux d'une matrice

$$\begin{aligned} \text{rang}(A) + \dim(N(A)) &= n \\ \text{rang}(A) + \dim(N(A^\top)) &= m \end{aligned}$$

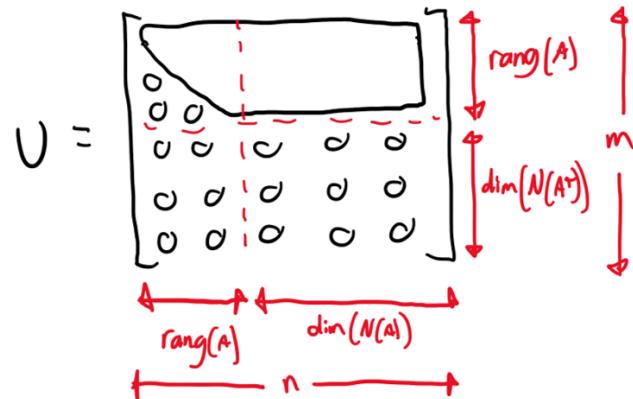


FIGURE 18.4 – Dimensions des quatre espaces fondamentaux d'une matrice

18.2.1 Espace colonne

L'espace colonne d'une matrice A représente l'ensemble de toutes les valeurs possible de la sortie \underline{y} qui résulte de l'opération $\underline{y} = Ax$. Avec l'interprétation présentée à la section 18.1.1, il est possible de voir que la sortie \underline{y} a comme domaine toutes les combinaisons linéaires possibles des vecteur-colonnes de la matrice. Comme illustré à la Figure 18.5, pour une matrice qui aurait deux colonnes, si on interprète ces deux colonnes comme les composantes de vecteurs spatiaux, alors le domaine possible pour la sortie \underline{y} va être le plan formé par ces deux vecteurs.

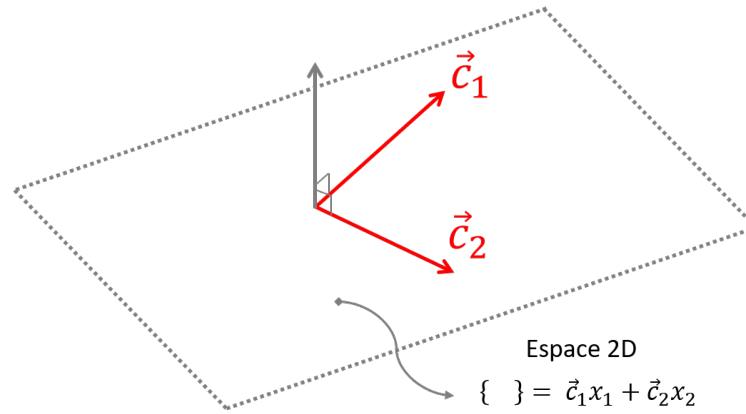


FIGURE 18.5 – Visualisation graphique d'un espace colonne de dimension 2

L'espace colonne d'une matrice A sera noté $col(A)$ et a la définition mathématique suivante :

$$col(A) = \{\underline{Ax} \mid \underline{x} \in \mathbb{R}^n\} \quad (18.29)$$

Notation mathématique pour les ensembles : Les crochets $\{\}$ représentent un ensemble, à l'intérieur on note la liste des éléments de l'ensemble ou bien la définition. La barre verticale $|$ signifie *tel que* et le symbole \in signifie *appartient à*. L'équation (18.29) peut donc se lire *l'espace colonne de la matrice A est égale à l'ensemble des résultats possible de l'opération Ax tel que x appartient à l'espace vectoriel de dimension n*.

Lorsque la matrice A est inversible, i.e. de rang plein, l'espace colonne correspond à \mathbb{R}^m , où m est le nombre de variables de sortie. Par exemple, pour une matrice 3×3 avec une sortie \underline{y} qui représente une position spatiale (x,y,z), si les colonnes ne sont pas co-planaire, alors la sortie \underline{y} va être une valeur arbitraire en 3D. On va donc dire que la matrice est inversible, car ici il est toujours possible de trouver une combinaison des colonnes pour obtenir un point (x,y,z) arbitraire. Toutefois, si les vecteurs-colonnes de la matrice sont co-linéaire (donc ils forment un plan), alors l'espace colonne est restreint à ce plan. La sortie \underline{y} peut seulement prendre des valeurs qui correspondent à ce plan. Dans cette situation la matrice est dite non-inversible.

18.2.2 Espace nul gauche

L'espace nul gauche (*Left-Nullspace*) d'une matrice A , noté $N(A^T)$, correspond au domaine de la sortie \underline{y} qui n'est pas dans l'espace colonne. Ce domaine correspond donc aux vecteurs \underline{y} qui sont impossible à obtenir peut importe le vecteur d'entrée \underline{x} . L'espace nul gauche correspond aussi à l'ensemble de tous les vecteur-colonnes \underline{y} pour lesquels la multiplication avec la matrice A^T donne un vecteur-colonne nulle $\underline{0}$:

$$N(A^T) = \{\underline{y} \in \mathbb{R}^m \mid A^T \underline{y} = \underline{0}\} \quad (18.30)$$

une condition qui revient à dire que c'est l'espace nul gauche correspond à l'ensemble des vecteurs \underline{y} qui sont perpendiculaires à toutes les colonnes de la matrice A .

L'espace des vecteur-colonnes sorties \underline{y} est formé par r (rang) bases qui forment l'espace colonne et $m - r$ bases qui forment l'espace nul gauche.

$$m = \dim(\text{col}(A)) + \dim(N(A^T)) = r + \dim(N(A^T)) \quad (18.31)$$

Les vecteur-colonnes \underline{y} appartenant à l'espace colonne ont une solution \underline{x} possible, tandis que ceux appartenant à l'espace nul gauche n'ont aucune solution exacte \underline{x} possible. Une matrice A va avoir un espace nul gauche seulement lorsqu'elle a une rang déficient. La présence d'un espace nul gauche indique un système sur-constraint, i.e. un surplus d'équations. Il y aura donc plusieurs sorties \underline{y} pour lesquelles il n'y a pas de solutions \underline{x} .

Exemple 1. Espace-nul gauche de la matrice jacobienne d'un robot manipulateur

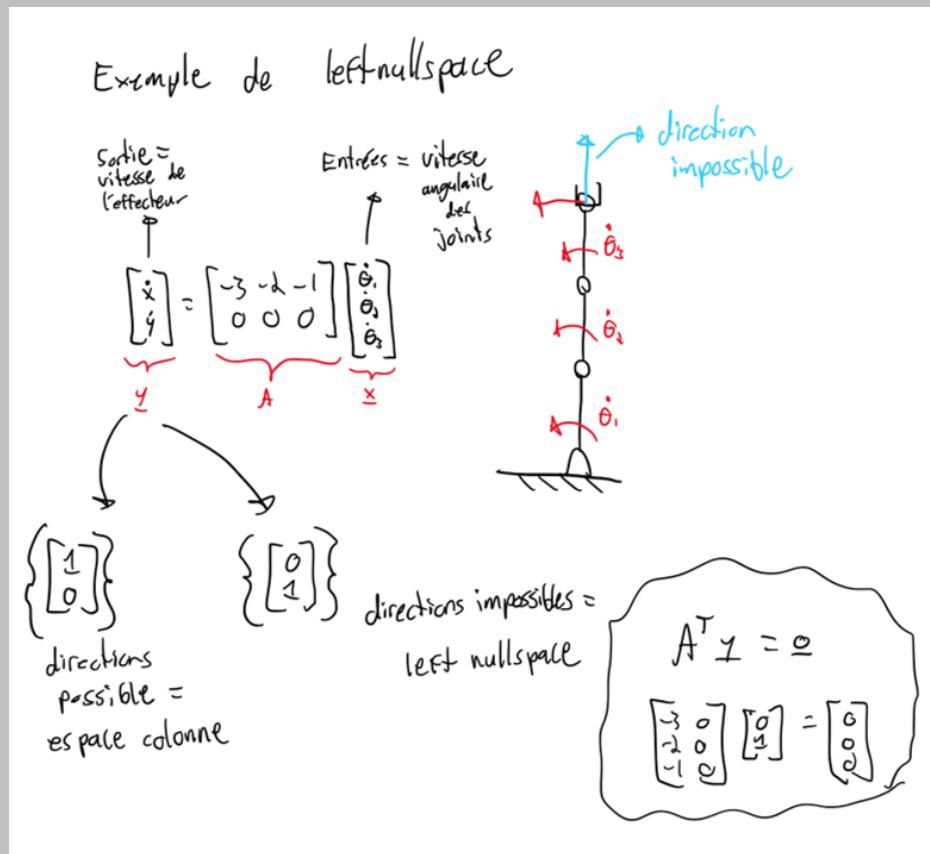


FIGURE 18.6 – Exemple de l'espace-nul gauche d'une matrice

18.2.3 Espace nul

L'espace nul (*Nullspace*) d'une matrice A , noté $N(A)$ correspond à l'ensemble de tous les vecteur-colonnes \underline{x} pour lesquels la multiplication avec la matrice A donne un vecteur-colonne nulle $\underline{0}$:

$$N(A) = \{\underline{x} \in \mathbb{R}^n \mid A\underline{x} = \underline{0}\} \quad (18.32)$$

Une matrice A va avoir un espace nul seulement lorsqu'elle a une rang déficient, donc des colonnes linéairement dépendantes. La présence d'un espace nul indique un système sous-constraint ou il y a un surplus de variables, donc plusieurs solutions possibles d'entrées \underline{x} pour obtenir une même sortie \underline{y} .

Exemple 2. Espace-nul de la matrice jacobienne d'un robot manipulateur

La figure 18.7 illustre la signification physique de l'espace nul d'une matrice qui relie la vitesse des joints d'un robot manipulateur à la vitesse de son effecteur.

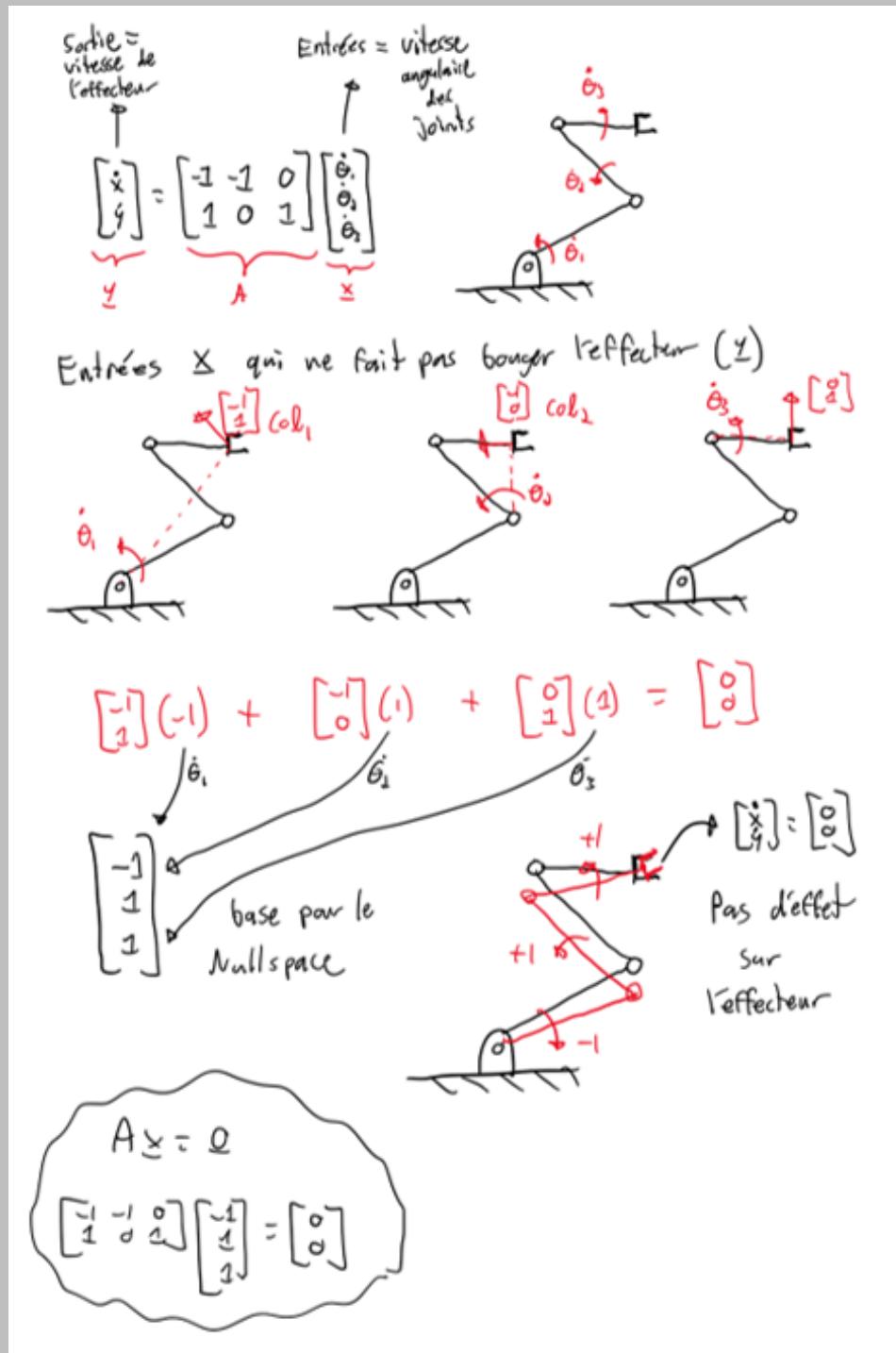


FIGURE 18.7 – Exemple de l'espace nul d'une matrice

18.2.4 Espace rangée

L'espace rangé correspond à toutes les combinaisons possibles des rangées de la matrice A , ce qui est donc équivalent à toutes les combinaisons possibles des colonnes de la matrice transposée A^T . L'espace rangée d'une matrice A sera noté $\text{row}(A)$ et a la définition suivante :

$$\text{row}(A) = \text{col}(A^T) = \{A^T \underline{y} \mid \underline{y} \in \mathbb{R}^m\} \quad (18.33)$$

Lorsque la matrice A est inversible, i.e. de rang plein, l'espace rangé correspond à \mathbb{R}^n , où n est le nombre de variables d'entrées.

L'espace des vecteur-colonnes entrées \underline{x} est formé par r (rang) bases qui forment l'espace rangée et $n - r$ bases qui forment l'espace nul :

$$n = \dim(\text{row}(A)) + \dim(N(A)) = r + \dim(N(A)) \quad (18.34)$$

Les vecteur-colonnes appartenant à l'espace rangé influencent la sortie alors que les vecteur-colonnes appartenant à l'espace nul n'influent pas la sortie.

18.2.5 Rang d'une matrice

Le rang r d'une matrice correspond à la dimension de l'espace colonne d'une matrice, qui est toujours aussi égale à la dimension de l'espace rangé. Ce nombre correspond aussi au nombre de pivots, il sera noté par la variable r :

$$r = \text{rank}(A) = \dim(\text{col}(A)) = \dim(\text{col}(A^T)) = \#\text{pivots} \quad (18.35)$$

Le rang d'une matrice est toujours inférieur ou égale aux dimensions m et n .

$$r \leq n = \#\text{ranges} \quad (18.36)$$

$$r \leq m = \#\text{colonnes} \quad (18.37)$$

Dans le cas d'une matrice carré ($m = n$), il est possible de déterminer si la matrice à une rang dit plein, i.e. si $r = m = n$, en calculant le déterminant de la matrice A . Si le déterminant est non-nul alors la matrice à un rang plein.

18.2.6 Quatre situations possibles pour un système d'équations

Un système d'équations peut être parfaitement contraint (une seule solution existe), sur-contraint (aucune solution exacte n'existe) ou bien sous-contraint (plusieurs solutions sont possibles). Dans le cas d'une relation matricielle de type $Ax = y$, qui représente un système d'équations linéaires, le rang de la matrice A permet de déterminer la situation, voir figure 18.8.

$$r = \text{rang}(A_{m \times n}) \quad \begin{matrix} m \\ \downarrow \\ [A] \\ \leftarrow n \end{matrix} \quad [x] = [y]$$

①	②	③	④
$r = m = n$ \rightarrow matrice inversible $V = \begin{bmatrix} I \\ 0 \\ 0 \end{bmatrix}$ \rightarrow 1 solution	$r = m < n$ \rightarrow trop de variables $V = \begin{bmatrix} I \\ 0 \\ 0 \\ 0 \end{bmatrix}$ \rightarrow Nullspace de dimension $(n - r)$ \rightarrow ∞ solutions possibles \rightarrow Utiliser une "pseudo-inverse" pour optimiser quelque chose \rightarrow sous-contraint	$r = n < m$ \rightarrow trop d'équations $V = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ \rightarrow 0 solution ou 1 solution si $y \in \text{col}(A)$ \rightarrow utiliser la méthode des moindres-carres pour solution approx. \rightarrow sur-contraint	$r < m, r < n$ \rightarrow trop d'équations \rightarrow trop de variables $V = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ \rightarrow 0 solution ou ∞ solutions si $y \notin \text{col}(A)$

FIGURE 18.8 – Quatre situations possibles pour un système d'équations

Les grandes catégories de situations possibles illustrées à la figure 18.8 sont exemplifiées avec les exemples illustrés aux figures 18.9, 18.10, 18.11 et 18.12.

1. Parfaitement contraint ($r = m = n$) : Il n'y a pas d'espace nul ni d'espace gauche nul.
2. Trop de variables ($r < n$) : Il y a un espace nul de dimension $n - r$.
3. Trop d'équations ($r < m$) : Il y a un espace gauche nul de dimension $m - r$.
4. Trop d'équations et de variables ($r < m, r < n$) : Il y a un espace gauche nul de dimension $m - r$ et un espace nul de dimension $n - r$.

Exemple 3. Système d'équation parfaitement constraint

exemple $r=m=n$

$$\begin{aligned} \begin{array}{l} x - y = 0 \\ y = 2 \end{array} &\Rightarrow \begin{bmatrix} A & \begin{matrix} x \\ y \end{matrix} \end{bmatrix} = \begin{bmatrix} \frac{y}{m} \\ \frac{y}{n} \end{bmatrix} \\ &\Rightarrow \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \end{bmatrix} \end{aligned}$$

$$\text{Solution } \Rightarrow \underline{x} = A^{-1} \underline{y} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

FIGURE 18.9 – Exemple d'un système d'équation parfaitement constraint

Exemple 4. Système d'équation sous-constraint

exemple $r=m < n$

$$x - y = 0 \Rightarrow \begin{bmatrix} A & \begin{matrix} x \\ y \end{matrix} \end{bmatrix} = \begin{bmatrix} \frac{y}{m} \\ \frac{y}{n} \end{bmatrix}$$

$$\text{Solution qui minimise } \underline{x}^T \underline{x} = x^2 + y^2$$

$$\underline{x} = \underbrace{\underline{A}^T (\underline{A} \underline{A}^T)^{-1}}_{A^\ddagger: \text{pseudo-inverse}} [\underline{y}] = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-1} \begin{bmatrix} 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

FIGURE 18.10 – Exemple d'un système d'équation sous-constraint

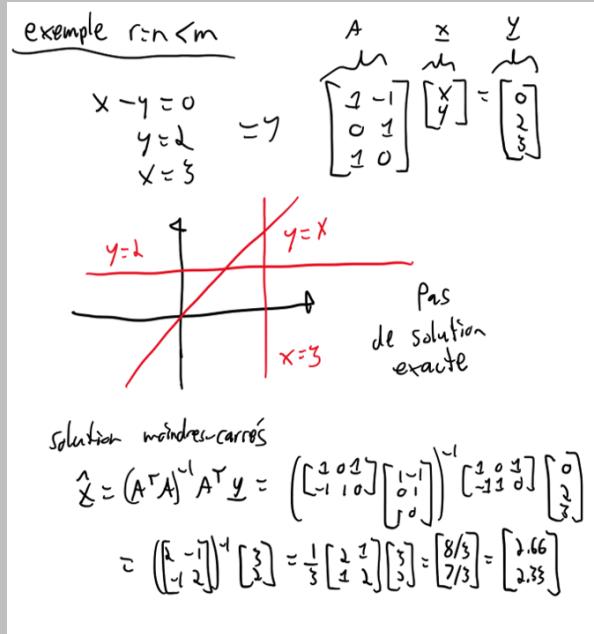
Exemple 5. Système d'équation sur-constraint

FIGURE 18.11 – Exemple d'un système d'équation sur-constraint

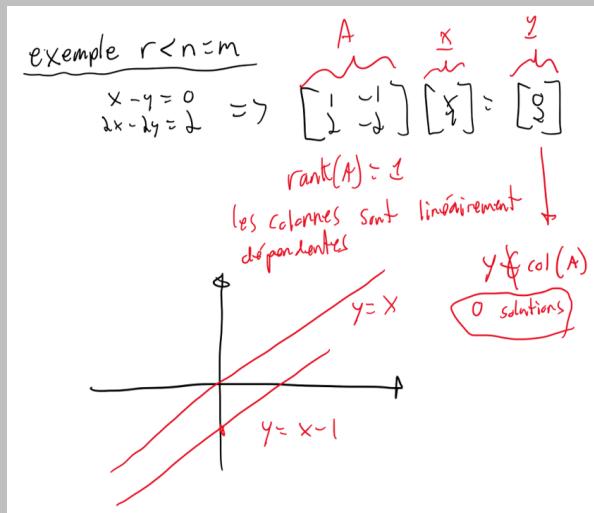
Exemple 6. Système d'équation non-inversible

FIGURE 18.12 – Exemple d'un système d'équation non-inversible

18.2.7 Factorisation LU

Une matrice A peut-être factorisée en deux matrices (triangulaires lorsque A est une matrice inversible) :

$$A = LU \tag{18.38}$$

où la matrice U est une matrice où tout les coefficients sous la diagonale sont nuls et la matrice L correspond aux opérations sur les rangées conduites dans le cadre d'une élimination Gauss-Jordan. Par exemple pour une matrice 3x3 inversible :

$$L = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \quad U = \begin{bmatrix} p_1 & u_{12} & u_{13} \\ 0 & p_2 & u_{23} \\ 0 & 0 & p_3 \end{bmatrix} \quad (18.39)$$

Les coefficients non-nuls p_i sur la diagonale de la matrice U sont appelés les pivots.

18.2.8 Base

Un ensemble $\{\underline{v}_1 \underline{v}_2 \dots \underline{v}_n\}$ de vecteurs de dimensions m dans un espace vectoriel appartenant à \mathbb{R}^m , forme une base de cet espace si : 1) les vecteurs $\underline{v}_1 \underline{v}_2 \dots \underline{v}_n$ sont linéairement indépendants et 2) tout vecteur \underline{w} de cet espace peut être exprimé comme une combinaison linéaire de ces vecteurs :

$$\underline{w} = x_1 \underline{v}_1 + x_2 \underline{v}_2 + \dots + x_n \underline{v}_n \quad (18.40)$$

Par exemple, dans un espace tri-dimensionnel qui correspond à des coordonnées $\begin{bmatrix} x \\ y \\ z \end{bmatrix}$, les vecteurs colonnes $\left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \right\}$ forment une base pour le sous-espace qui correspond au plan $z = 0$.

18.2.9 Indépendance linéaire

Un vecteur est dit linéairement dépendant d'un autre vecteur (ou un ensemble de vecteurs), lorsque celui-ci peut être exprimé comme une combinaison pondérée des autres.

Un ensemble de vecteurs $\{\underline{v}_1, \underline{v}_2, \dots, \underline{v}_n\}$ est **linéairement dépendant** s'il existe des coefficients non nuls x_i tels que la somme pondérée des vecteurs est nulle :

$$\underline{0} = x_1 \underline{v}_1 + x_2 \underline{v}_2 + \dots + x_n \underline{v}_n \quad (18.41)$$

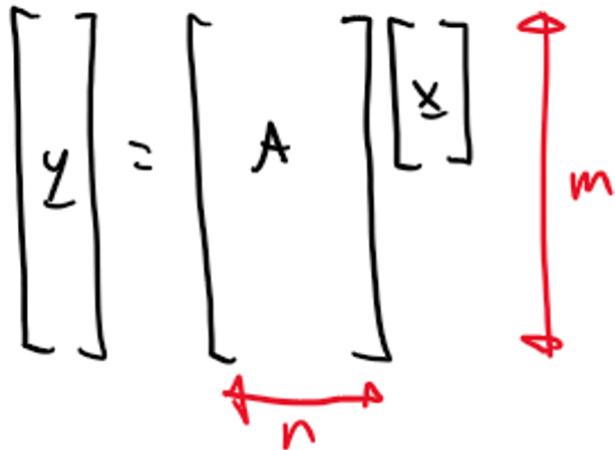
Inversement, un ensemble de vecteurs $\{\underline{v}_1, \underline{v}_2, \dots, \underline{v}_n\}$ est **linéairement indépendant** si :

$$\underline{0} = x_1 \underline{v}_1 + x_2 \underline{v}_2 + \dots + x_n \underline{v}_n \quad \text{implique que : } 0 = x_1 = x_2 = \dots = x_n \quad (18.42)$$

18.3 Moindres carrés

Lorsqu'un système d'équation linéaire $\underline{y} = A\underline{x}$ est sur-constraint, c'est-à-dire que le vecteur \underline{y} n'est pas dans l'espace colonne de la matrice A , il n'y a pas de solution exacte possible. Il est toutefois possible de calculer une solution approximée qui minimise la norme de l'erreur avec une expression explicite :

$$\hat{\underline{x}}^* = \underset{\hat{\underline{x}}}{\operatorname{argmin}} \|A\hat{\underline{x}} - \underline{y}\|^2 = (A^T A)^{-1} A^T \underline{y} \quad (18.43)$$



équations > # variable
 → pas de solution exacte
 → Solution approximée $\hat{\underline{x}}$
 qui minimise $\|A\hat{\underline{x}} - \underline{y}\|^2$

$$\hat{\underline{x}} = [A^T A]^{-1} A^T \underline{y}$$

FIGURE 18.13 – Système sur-constraint : solution des moindres-carrés

La matrice $(A^T A)^{-1} A^T$ est souvent appelée inverse gauche de Moore-Penrose et notée :

$$A^+ = (A^T A)^{-1} A^T \Rightarrow A^+ A = I \quad (18.44)$$

Il est à noter que la matrice A doit avoir des colonnes indépendantes pour que la matrice $A^T A$ soit inversible et donc pour que la méthode des moindres-carrés fonctionne.

18.3.1 Application à des problèmes d'identification de paramètres

La méthode des moindres-carrés est très utile pour identifier des paramètres inconnus dans une relation linéaire. Lorsqu'on a un modèle linéaire qui implique des signaux connus et des paramètres inconnus qui

peut être exprimé sous la forme :

$$\underline{\varphi}^T \underline{\theta} = b \quad \Rightarrow \quad \underbrace{\begin{bmatrix} \varphi_1 & \dots & \varphi_m \end{bmatrix}}_{m \text{ signaux connus}} \underbrace{\begin{bmatrix} \theta_1 \\ \vdots \\ \theta_m \end{bmatrix}}_{m \text{ paramètre inconnus}} = \underbrace{b}_{\text{signal connu}} \quad (18.45)$$

Il est possible de regrouper plusieurs mesures expérimentales dans un système matriciel :

$$A\underline{\theta} = \underline{b} \quad \Rightarrow \quad \underbrace{\begin{bmatrix} \underline{\varphi}^T(1) \\ \vdots \\ \underline{\varphi}^T(N) \end{bmatrix}}_{A : N \times m \text{ known samples}} \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_m \end{bmatrix} = \underbrace{\begin{bmatrix} b(1) \\ \vdots \\ b(N) \end{bmatrix}}_{\underline{b} : N \text{ known samples}} \quad (18.46)$$

Pour ensuite utiliser la méthode des moindres-carrés pour obtenir un estimé des paramètres inconnus qui minimise l'erreur avec les mesures expérimentales :

$$\hat{\underline{\theta}}^* = \underset{\hat{\underline{\theta}}}{\operatorname{argmin}} \|A\hat{\underline{\theta}} - \underline{b}\|^2 = (A^T A)^{-1} A^T \underline{b} \quad (18.47)$$

Exemple 7. Régression linéaire avec les moindres-carrés

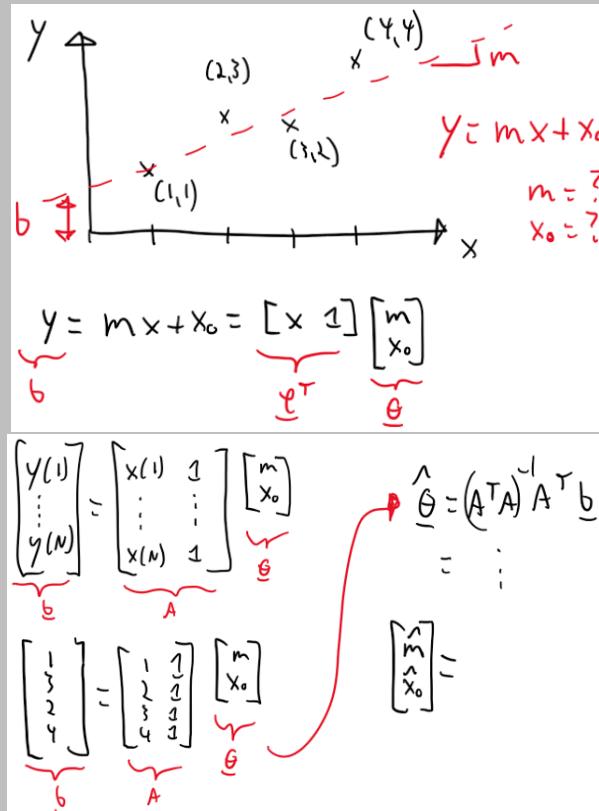


FIGURE 18.14 – Régression linéaire avec les moindres-carrés pour identifier les paramètres d'une ligne

Preuve 18.3.1 La solution des moindres-carrés minimise une fonction coût qui est défini comme le carré de la norme du vecteur d'erreur, qui correspond à la somme des carrés des erreurs individuelles pour chaque sortie du système :

$$J = \|\underline{e}\|^2 = \underline{e}^T \underline{e} = \sum e_i^2 \quad (18.48)$$

L'erreur est définie comme la différence entre les sorties \underline{y} et le résultat du calcul $A\underline{\hat{x}}$:

$$\underline{e} = A\underline{\hat{x}} - \underline{y} \quad (18.49)$$

Il est ensuite possible de trouver le minimum de la fonction coût J en trouvant le point pour lequel toutes les dérivées par rapport à $\underline{\hat{x}}$ sont nulles :

$$\underline{0} = \frac{\partial J}{\partial \underline{\hat{x}}} \quad (18.50)$$

$$\underline{0} = \frac{\partial \underline{e}^T \underline{e}}{\partial \underline{\hat{x}}} \quad (18.51)$$

$$\underline{0} = 2\underline{e}^T \frac{\partial \underline{e}}{\partial \underline{\hat{x}}} \quad (18.52)$$

$$\underline{0} = 2(A\underline{\hat{x}} - \underline{y})^T \frac{\partial(A\underline{\hat{x}} - \underline{y})}{\partial \underline{\hat{x}}} \quad (18.53)$$

$$\underline{0} = 2(A\underline{\hat{x}} - \underline{y})^T A \quad (18.54)$$

$$\underline{0} = \underline{\hat{x}}^T A^T A - \underline{y}^T A \quad (18.55)$$

$$A^T \underline{y} = A^T A \underline{\hat{x}} \quad (18.56)$$

$$(A^T A)^{-1} A^T \underline{y} = \underline{\hat{x}} \quad (18.57)$$

18.4 Pseudo-Inverse

Lorsqu'un système d'équation linéaire $\underline{y} = A\underline{x}$ est sous-constraint, c'est-à-dire que plusieurs vecteurs \underline{x} sont des solutions, il est possible d'utiliser une matrice pseudo-inverse pour obtenir une solution qui optimise une fonction coût quadratique :

$$\underline{x}^* = \underset{\underline{x}}{\operatorname{argmin}} (\underline{x}^T Q \underline{x}) \quad \text{subject to } \underline{y} = A\underline{x} \quad (18.58)$$

$$\underline{x}^* = Q^{-1} A^T (A Q^{-1} A^T)^{-1} \underline{y} \quad (18.59)$$

Dans un cas simplifié où la matrice de poids Q est réduite à une matrice identité on a :

$$\underline{x}^* = \underset{\underline{x}}{\operatorname{argmin}} \|\underline{x}\|^2 \quad \text{subject to } \underline{y} = A\underline{x} \quad (18.60)$$

$$\underline{x}^* = A^T (A A^T)^{-1} \underline{y} \quad (18.61)$$

La matrice $A^T (A A^T)^{-1}$ est souvent appelée inverse droit de Moore-Penrose et notée :

$$A^\# = A^T (A A^T)^{-1} \Rightarrow A A^\# = I \quad (18.62)$$

Il est à noter que la matrice A doit avoir des rangées indépendantes pour que la matrice $A A^T$ soit inversible et donc pour pouvoir utiliser la pseudo-inverse droite.

$$\begin{array}{c} m \\ \downarrow \\ \left[\begin{array}{c} \underline{y} \end{array} \right] = \left[\begin{array}{c|c} & A \\ \hline \underline{A} & \end{array} \right] \left[\begin{array}{c} \underline{x} \end{array} \right] \\ n \end{array}$$

$n > m$: sous-constraint

→ plusieurs solutions possibles

→ Solution qui minimise $\underline{x}^T \underline{x}$

$$\underline{x}^* = \underbrace{A^T (A A^T)^{-1}}_{A^\#} \underline{y}$$

$A^\#$: pseudo-inverse droite de Moore-Penrose

FIGURE 18.15 – Système sous-constraint - solution avec matrice pseudo-inverse

Exemple 8. Utilisation d'une matrice pseudo-inverse

Pour cet exemple, un robot doit produire une force de 14 N sur un mur. Pour y parvenir le courant électrique dans les trois moteurs doit être ajusté. La relation entre la force et le courant dans les moteurs est linéaire et illustrée ci-dessous. On cherche ici une solution qui minimise les pertes joules $r_i i^2$ dans les moteurs. Comme il y a plusieurs solutions possibles pour atteindre le niveau de force désiré, et que la fonction coût à minimiser est quadratique, une matrice pseudo-inverse droite peut être utilisée pour calculer la solution optimale directement, voir Figure 18.16.

$$\begin{aligned} F_y &= \begin{bmatrix} 2 & 1 & -3 \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix} \\ F_y &= 14 \text{ avec } \min \left[r_1 i_1^2 + r_2 i_2^2 + r_3 i_3^2 \right] \\ &= \min \underbrace{x^T}_{Q} \underbrace{\begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{bmatrix}}_{Q} x \end{aligned}$$

$$\begin{aligned} x^* &= Q^{-1} A^T (A Q^{-1} A^T)^{-1} y \\ &= \begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ -3 \end{bmatrix} \left(\begin{bmatrix} 2 & 1 & -3 \end{bmatrix} \begin{bmatrix} 1/r_1 & 0 & 0 \\ 0 & 1/r_2 & 0 \\ 0 & 0 & 1/r_3 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ -3 \end{bmatrix} \right)^{-1} \begin{bmatrix} 14 \\ 1 \\ -3 \end{bmatrix} \\ &= \begin{bmatrix} 2/r_1 \\ 1/r_2 \\ -3/r_3 \end{bmatrix} \frac{14}{4/r_1 + 1/r_2 + 9/r_3} \\ \text{Si } r_1 = r_2 = r_3 = 1 \\ &= \begin{bmatrix} 2 \\ 1 \\ -3 \end{bmatrix} \quad \text{validation} \\ F_y &= \begin{bmatrix} 2 & 1 & -3 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ -3 \end{bmatrix} = 14 \end{aligned}$$

FIGURE 18.16 – Exemple d'utilisation d'une matrice pseudo-inverse

18.5 Déterminants

À venir!

$$\det(A) = \det(U) = p_1 p_2 \dots p_n \quad (18.63)$$

18.6 Vecteurs et valeurs propres

Pour les matrices carrées ($m = n$), une autre caractéristique utile est l'ensemble de ses vecteurs et valeurs propres. Un vecteur \underline{x} est un vecteur propre d'une matrice A si le résultatat $\underline{y} = A\underline{x}$ est parallèle à l'entrée \underline{x} , autrement dit s'il existe un scalaire λ (appelé valeur propre) tel que :

$$A\underline{x} = \lambda \underline{x} \quad (18.64)$$

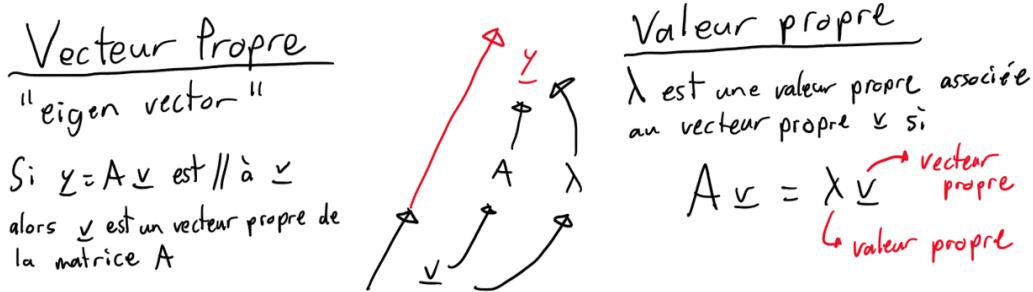


FIGURE 18.17 – Vecteurs et valeurs propres

Une matrice $n \times n$ va avoir n paires de vecteur et valeurs propres (qui peuvent être répétée et aussi des nombres complexes). Si une ou plusieurs valeurs propres de la matrice est égale à zéro alors la matrice est singulière et les vecteurs propres associés forme une base de l'espace nul de la matrice.

La trace d'un matrice est égale à la somme des valeurs propres et le déterminant d'une matrice est égale à la multiplication de toutes les valeurs propres :

$$\text{trace}(A) = \sum_1^n \lambda_i \quad (18.65)$$

$$\det(A) = \prod_1^n \lambda_i \quad (18.66)$$

Pour déterminer les valeurs et vecteurs propres d'une matrice, le problème est posé ainsi :

$$Ax = \lambda \underline{x} \quad (18.67)$$

$$(A - \lambda I)\underline{x} = 0 \quad (18.68)$$

La matrice $(A - \lambda I)$ doit être singulière pour qu'il existe des solutions non-nulles à cette équation. Il est donc possible de trouver les valeurs propres λ_i en solutionnant l'équation suivante :

$$\det(A - \lambda I) = 0 \quad (18.69)$$

Pour chaque solution, i.e pour chaque valeur propre λ_i , il est possible de déterminer le vecteur propre associé \underline{v}_i en déterminant l'espace nul de $(A - \lambda I)$:

$$\underline{v}_i \in \text{nul}(A - \lambda I) \quad (18.70)$$

Il est à noté que pour chaque valeur propre, le vecteur propre n'est pas unique, seule la direction l'est. Si \underline{v}_i est un vecteur propre alors tout multiple de lui même est aussi un vecteur propre. Donc autrement dit pour chaque valeur propre λ_i , on cherche en fait une base pour l'espace 1D associé aux vecteurs propres de cette valeur.

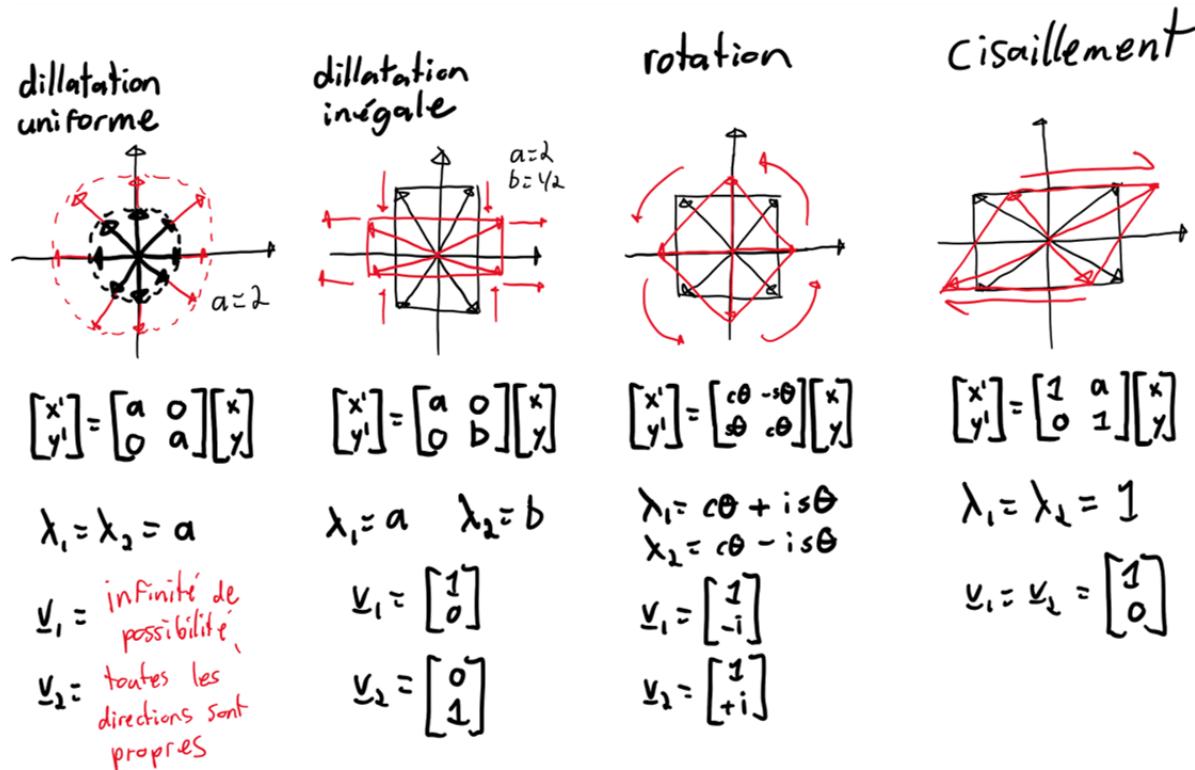


FIGURE 18.18 – Exemple de vecteurs et valeurs propres dans un contexte de transformations géométriques

18.6.1 Diagonalisation

Si une matrice a n vecteurs propres indépendants, alors la matrice peut être diagonalisée, c'est à dire décomposée sous la forme :

$$A = V\Lambda V^{-1} \quad (18.71)$$

où la matrice V regroupe tous les vecteurs propres de A , et la matrice diagonale Λ toutes les valeurs propres de A :

$$V = \left[\begin{bmatrix} \underline{v}_1 \\ \vdots \\ \underline{v}_n \end{bmatrix} \dots \begin{bmatrix} \underline{v}_n \end{bmatrix} \right] \quad \Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \lambda_n \end{bmatrix} \quad (18.72)$$

Lorsqu'une matrice a plusieurs vecteur propres associés à la même valeur propre alors la matrice n'est pas diagonalisable. Il est toutefois possible de faire appelle à une méthode alternative appelée la réduction de Jordan. La diagonalisation d'une matrice peut être interprétée comme un changement de base, vers des coordonnées dites propres, pour lesquelles l'effet de la multiplication de cette matrice est indépendant pour chaque axes.

Preuve 18.6.1 Par définition, si les vecteurs propres \underline{v}_i sont multipliés leur matrice A , la multiplication est équivalente à les multiplier par les valeurs propres associées λ_i :

$$AV = \left[A \begin{bmatrix} \underline{v}_1 \\ \vdots \\ \underline{v}_n \end{bmatrix} \dots A \begin{bmatrix} \underline{v}_n \end{bmatrix} \right] = \left[\lambda_1 \begin{bmatrix} \underline{v}_1 \\ \vdots \\ \underline{v}_n \end{bmatrix} \dots \lambda_n \begin{bmatrix} \underline{v}_1 \\ \vdots \\ \underline{v}_n \end{bmatrix} \right] = V\Lambda \quad (18.73)$$

Ensuite, si la matrice V est inversible (ce qui explique la condition d'indépendance des vecteurs propres pour que la matrice soit diagonalisable), il suffit de multiplier l'équation (18.71) par V^{-1} par la droite pour obtenir l'équation (18.71).

Exemple 9. Exemple de diagonalisation pour une dilatation en 2D

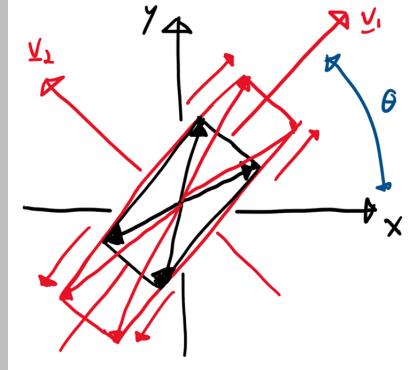


FIGURE 18.19 – Exemple de diagonalisation pour une transformation géométrique

$$A = \begin{bmatrix} c\theta^2 + 1 & c\theta s\theta \\ c\theta s\theta & s\theta^2 + 1 \end{bmatrix} = \underbrace{\begin{bmatrix} c\theta & -s\theta \\ s\theta & c\theta \end{bmatrix}}_V \underbrace{\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}}_{\Lambda} \underbrace{\begin{bmatrix} c\theta & s\theta \\ -s\theta & c\theta \end{bmatrix}}_{V^{-1}} \quad (18.74)$$

18.6.2 Puissance d'une matrice

La puissance p d'une matrice est définie comme la multiplication répétée de p copie de cette matrice. Si la matrice est diagonalisable, la puissance de la matrice A est égale à

$$A^p = V\Lambda^p V^{-1} \quad (18.75)$$

avec :

$$\Lambda^p = \begin{bmatrix} \lambda_1^p t & 0 & 0 & 0 \\ 0 & \lambda_2^p & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \lambda_n^p \end{bmatrix} \quad (18.76)$$

Cette propriété découle de la diagonalisation :

$$A^p = A A \dots A = V\Lambda V^{-1} V\Lambda V^{-1} \dots V\Lambda V^{-1} = V\Lambda\Lambda\dots\Lambda V^{-1} = V\Lambda^p V^{-1} \quad (18.77)$$

Exemple 10. Exemple de calcul de l'évolution d'un système discret

La suite de Fibonacci est définie par une séquence de chiffre pour lequel le prochain est la somme des deux précédents :

$$0 \ 1 \ 1 \ 2 \ 3 \ 5 \ 8 \ 13 \ 21 \ 34 \ 55 \quad (18.78)$$

Le chiffre suivant est calculé en fonction des deux précédents :

$$f_{k+1} = f_k + f_{k-1} \quad (18.79)$$

il faut donc un vecteur d'état \underline{x} de deux variables (f_k et f_{k-1}) pour prédire le prochain chiffre. L'évolution du vecteur d'état peut être décrite par l'équation matricielle suivante :

$$\underbrace{\begin{bmatrix} f_{k+1} \\ f_k \end{bmatrix}}_{\underline{x}_{k+1}} = \underbrace{\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}}_A \underbrace{\begin{bmatrix} f_k \\ f_{k-1} \end{bmatrix}}_{\underline{x}_k} \quad (18.80)$$

Il est possible de calculer la valeur future d'une évolution discrète comme celle-ci sans calculer toute la séquence grâce la propriété (18.75) :

$$\underline{x}_{k+1} = A\underline{x}_k \quad (18.81)$$

$$\underline{x}_{k+2} = A\underline{x}_{k+1} = AA\underline{x}_k \quad (18.82)$$

$$\underline{x}_{k+3} = A\underline{x}_{k+2} = AAA\underline{x}_k \quad (18.83)$$

$$\underline{x}_{k+p} = A^p \underline{x}_k = V \Lambda^p V^{-1} \underline{x}_k \quad (18.84)$$

Pour la suite de Fibonacci, la matrice A qui décrit l'évolution peut être diagonalisée. Premièrement on détermine les valeurs propres :

$$0 = \det(A - \lambda I) = \det \left(\begin{bmatrix} 1 - \lambda & 1 \\ 1 & -\lambda \end{bmatrix} \right) = \lambda^2 - \lambda - 1 \quad (18.85)$$

Il y a donc deux solutions :

$$\lambda_1 = \frac{1 + \sqrt{5}}{2} = 1.61803399 \quad \lambda_2 = \frac{1 - \sqrt{5}}{2} = -0.61803399 \quad (18.86)$$

ou la première valeur propre est égale au très fameux nombre d'or. Il est possible de déterminer les vecteurs propres associés en substituant dans l'équation $A\underline{v} = \lambda \underline{v}$, on trouve alors :

$$\underline{v}_1 = \begin{bmatrix} \lambda_1 \\ 1 \end{bmatrix} \quad \underline{v}_2 = \begin{bmatrix} \lambda_2 \\ 1 \end{bmatrix} \quad (18.87)$$

Il est donc maintenant possible de calculer n'importe quel position future dans la séquence directement grâce à l'équation (18.84). Si on cherche la position $p = 1E9$ à partir du début de la suite de Fibonacci :

$$\underline{x}_p = V \Lambda^p V^{-1} \underline{x}_0 \quad (18.88)$$

$$\begin{bmatrix} f_p \\ f_{p-1} \end{bmatrix} = \begin{bmatrix} \lambda_1 & \lambda_2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1^p & 0 \\ 0 & \lambda_2^p \end{bmatrix} \begin{bmatrix} \lambda_1 & \lambda_2 \\ 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (18.89)$$

lorsque p est très grand il est possible de simplifier le calcul en négligeant la contribution $\lambda_2^p \approx 0$ puisque lorsque que la valeur absolue de la base est inférieure à l'unité, la puissance tend vers zéro lorsque p tend vers l'infini. Il est donc possible de simplifier :

$$\begin{bmatrix} f_p \\ f_{p-1} \end{bmatrix} = \begin{bmatrix} \lambda_1 & \lambda_2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1^p & 0 \\ 0 & 0 \end{bmatrix} \frac{1}{\lambda_1 - \lambda_2} \begin{bmatrix} 1 & -\lambda_2 \\ -1 & \lambda_1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (18.90)$$

$$\begin{bmatrix} f_p \\ f_{p-1} \end{bmatrix} = \begin{bmatrix} \lambda_1^{p+1} & 0 \\ \lambda_1^p & 0 \end{bmatrix} \frac{1}{\lambda_1 - \lambda_2} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad (18.91)$$

$$\begin{bmatrix} f_p \\ f_{p-1} \end{bmatrix} = \frac{1}{\lambda_1 - \lambda_2} \begin{bmatrix} \lambda_1^{p+1} \\ \lambda_1^p \end{bmatrix} \quad (18.92)$$

Il est donc possible de calculer directement la valeur dans la suite pour une positon p lorsque p est très grand :

$$f_p = \frac{\lambda_1^{p+1}}{\lambda_1 - \lambda_2} = \frac{1.618^{(p+1)}}{\sqrt{5}} \quad (18.93)$$

18.6.3 Exponentiel de matrice

À venir !

$$e^{At} = Ve^{\Lambda t}V^{-1} \quad (18.94)$$

$$e^{\Lambda t} = \begin{bmatrix} e^{\lambda_1 t} & 0 & 0 & 0 \\ 0 & e^{\lambda_2 t} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & e^{\lambda_n t} \end{bmatrix} \quad (18.95)$$

Chapitre 19

Optimisation

19.1 Descente du gradient

19.2 Les multiplicateurs de Lagrange

À venir !



Capsule vidéo
Multiplicateurs de Lagrange
<https://youtu.be/W7kzp927xXc>

Chapitre 20

Trigonométrie

20.0.1 Fonctions trigonométriques

À venir !

20.0.2 Identités trigonométriques

Somme au carré :

$$\sin^2(\theta) + \cos^2(\theta) = 1 \quad (20.1)$$

Inversion du signe de l'angle :

$$\sin(-\theta) = -\sin(\theta) \quad (20.2)$$

$$\cos(-\theta) = \cos(\theta) \quad (20.3)$$

Déphasage de 90° :

$$\sin\left(\frac{\pi}{2} - \theta\right) = \cos(\theta) \quad (20.4)$$

$$\cos\left(\frac{\pi}{2} - \theta\right) = \sin(\theta) \quad (20.5)$$

Déphasage de 180° :

$$\sin(\pi - \theta) = \sin(\theta) \quad (20.6)$$

$$\cos(\pi - \theta) = -\cos(\theta) \quad (20.7)$$

Somme de deux angles :

$$\sin(\theta_1 + \theta_2) = \sin(\theta_1)\cos(\theta_2) + \cos(\theta_1)\sin(\theta_2) \quad (20.8)$$

$$\sin(\theta_1 - \theta_2) = \sin(\theta_1)\cos(\theta_2) - \cos(\theta_1)\sin(\theta_2) \quad (20.9)$$

$$\cos(\theta_1 + \theta_2) = \cos(\theta_1)\cos(\theta_2) - \sin(\theta_1)\sin(\theta_2) \quad (20.10)$$

$$\cos(\theta_1 - \theta_2) = \cos(\theta_1)\cos(\theta_2) + \sin(\theta_1)\sin(\theta_2) \quad (20.11)$$

Dérivés :

$$\frac{d}{d\theta} \sin(\theta) = \cos(\theta) \quad (20.12)$$

$$\frac{d}{d\theta} \cos(\theta) = -\sin(\theta) \quad (20.13)$$

Bibliographie

- [1] H. Harry Asada and Jean-Jacques Slotine. *Robotics : Analysis and Control*. 1986.
- [2] Jean de Lafontaine. Course notes for udes gei720 - commande multi-variable, 2011.
- [3] Kevin M Lynch and Frank C. Park. *Modern Robotics : Mechanics, Planning and Control*. 2017.
- [4] Richard M Murray, Zexiang Li, S Shankar Sastry, and S Shankara Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994.
- [5] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics : Modelling, Planning and Control*. Springer-Verlag, 2009.
- [6] Jean-Jacques Slotine. Course notes for mit 2.165, 2014.
- [7] Jean-Jacques Slotine and Weiping Li. *Applied Nonlinear Control*. Pearson, October 1990.
- [8] Gilbert Strang. *Introduction to Linear Algebra*. 2009.
- [9] Russ Tedrake. Underactuated robotics : Algorithms for walking, running, swimming, flying, and manipulation (course notes for mit 6.832).