

Introduction et Modèle Géométrique Direct

Ludovic Hofer

14 septembre 2022

Objectifs généraux

- Se familiariser avec les fondamentaux de la robotique
 - Modèles géométriques et cinématiques
 - Planification de trajectoire
 - Lois de contrôle
- Apprendre à réaliser des expérimentations robotiques
 - Prise en main d'outils de simulation
 - Tester votre code
 - Comparer différentes solutions

Organisation cours et évaluation

12 séances : 5 blocs

- 1 bloc = 2 à 3 semaines de cours
- Dans chaque bloc
 - Cours
 - TP noté

Travail de groupe

- Taille des groupes : 2
- Groupes tirés au sort
- Un rendu par groupe

Évaluation des rendus

Correction semi-automatiques

- Respect des consignes
- Scénarios publics (Parfois)
- Scénarios privés

Qualité du code

- Pas de magic number
- Des commentaires de code, pas du code commenté
- Éviter la duplication de code
- Lisibilité du code

Compte-Rendu

Format libre (txt, markdown, pdf)

Calendrier thématique prévisionnel

- Séances 1 et 2 : Introduction et Modèle Géométrique Direct
- Séances 3,4 et 5 : Jacobienne et Modèle Géométrique Inverse
- Séances 6,7 et 8 : Trajectoires
- Séances 9 et 10 : Contrôle
- Séances 11 et 12 : À définir

Références

- Ce cours se base sur les documents suivants :
 - Bases de la modélisation et de la commande des robots-manipulateurs de type série (Khalil, Dombre)¹
 - Craig, J. J. (2004). Introduction to Modern Robotics : Mechanics, Planning, and Control. In Prentice Hall.
 - Niku, S. B. (2001). Introduction to Robotics : Analysis, Control, Applications. Prentice Hall New Jersey.

1. http://cours-online.gdr-robotique.org/Khalil-Dombre_Modelisation/Khalil-Dombre_Modelisation.pdf

Quels outils pour ce cours ?

- Webots : simulateur physique 3D avec visualisation²
- numpy : bibliothèque d'algèbre linéaire
- plotly : affichage de graphiques pour visualiser les données

2. <https://cyberbotics.com/>

Comment développer en robotique ?

- Tester unitairement ce qui peut l'être
- Comparer les résultats obtenus à ceux attendus
- Quand il y a une différence, s'interroger sur la raison
 - Bug ?
 - Modèle imparfait ?
 - Accumulation d'erreurs ?

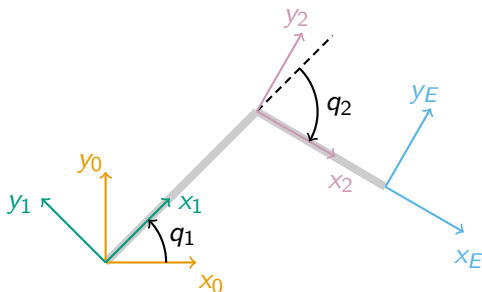
Grande difficulté : « D'où vient le problème ? »

- Problème logiciel : ne pas blâmer le matériel.
- Problème matériel : ne pas chercher de bugs dans le logiciel.
- Système multi-couche non reproductible → post-mortem.

Robots articulés et modèle géométriques

- Deux types d'articulations
 - Rotoïde (liaisons angulaires)
 - Linéaire (liaisons prismatiques)
- Deux types d'architecture
 - Séries
 - Parallèles (hors cours)
- Trois niveaux de modèles
 - Position
 - Vitesse
 - Accélération, Forces (hors cours)

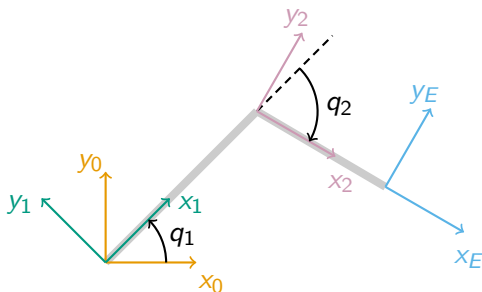
Vocabulaire



Espace articulaire : Q

- n : nombre d'articulations (degrés de liberté)
- Configuration du robot $q = (q_1, \dots, q_n)$
 - Ici : $q = (\pi/6, -5\pi/12)$

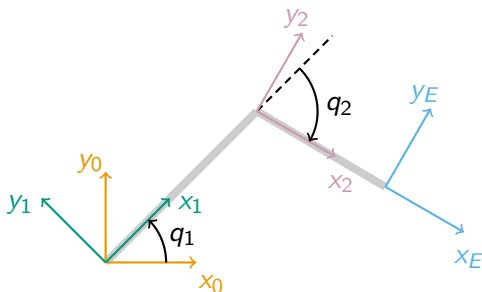
Vocabulaire



Les repères du robot

- 0 : La base du robot
- i : Repère du corps rigide après i articulations
- E : L'effecteur ou outil du robot (pince, etc. . .)

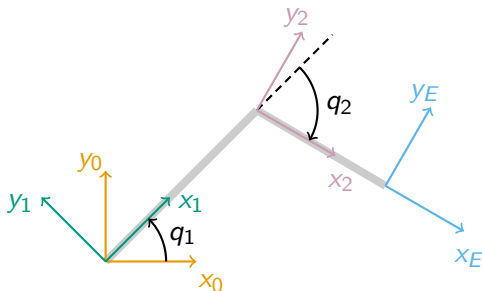
Vocabulaire



Espace opérationnel : \mathcal{O}

- Position de l'outil dans le repère 0
- Orientation du repère E dans le repère 0

Vocabulaire



Les modèles géométriques

Direct : $\mathcal{Q} \rightarrow \mathcal{O}$ (MGD, *Forward Kinematics*)

Inverse : $\mathcal{O} \rightarrow \mathcal{Q}$ (MGI, *Inverse Kinematics*)

Repères

Note :

On se restreint ici aux repères orthonormés directs

Notations

O_i L'origine du repère i

\vec{x}_i Le vecteur unitaire \vec{x} du repère i

${}^i\vec{v}$ Le vecteur \vec{v} dans le repère i

iP La position du point P dans le repère i

Remarque

O_i , \vec{x}_i et \vec{y}_i suffisent à spécifier un repère, car :

$$\vec{z}_i = \vec{x}_i \wedge \vec{y}_i$$

Rotation en 3D

Notation

$${}^iR_j = \begin{pmatrix} {}^i\vec{x}_j & {}^i\vec{y}_j & {}^i\vec{z}_j \end{pmatrix}$$

- Conversion coordonnées de j vers i :
 - ${}^iR_j {}^j\vec{v} = {}^i\vec{v}$

Propriétés

- ${}^iR_j = {}^jR_i^T$
- ${}^iR_j = {}^jR_i^{-1}$
- ${}^iR_j {}^jR_k = {}^iR_k$

Transformation homogène en 3D

Notation

$${}^i T_j = \begin{pmatrix} {}^i R_j & {}^j \vec{O}_i \\ 0_{1 \times 3} & 1 \end{pmatrix}$$

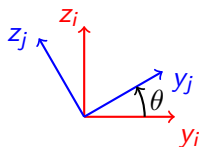
- Conversion coordonnées de j vers i :
 - ${}^i T_j {}^j p = {}^i p$
 - ${}^i T_j {}^j \vec{v} = {}^i \vec{v}$

Propriétés

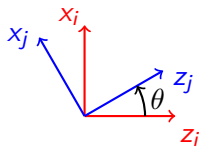
- ${}^i T_j {}^j T_k = {}^i T_k$
- ${}^i \vec{O}_j = -{}^i R_j {}^j \vec{O}_i$
- ${}^i T_j^{-1} = {}^j T_i = \begin{pmatrix} {}^i R_j^T & {}^i \vec{O}_j \\ 0_{1 \times 3} & 1 \end{pmatrix} = \begin{pmatrix} {}^i R_j^T & -{}^i R_j^T {}^j \vec{O}_i \\ 0_{1 \times 3} & 1 \end{pmatrix}$

Rotation autour des axes unitaires

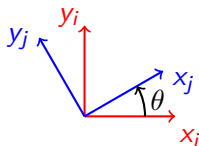
$$\mathcal{R}(\vec{x}, \theta) = {}^i T_j = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



$$\mathcal{R}(\vec{y}, \theta) = {}^i T_j = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



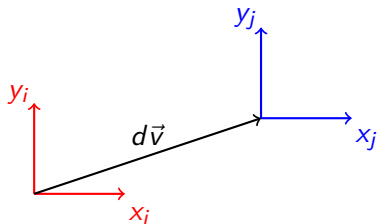
$$\mathcal{R}(\vec{z}, \theta) = {}^i T_j = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



Translation selon un axe

- $\vec{v} \in \mathbb{R}^n$: un vecteur unitaire indiquant l'axe de translation
- $d \in \mathbb{R}$: La distance de la translation

$$\mathcal{T}(\vec{v}, d) = {}^i T_j = \begin{pmatrix} I_{3 \times 3} & d\vec{v} \\ 0_{1 \times 3} & 1 \end{pmatrix}$$



Quelques représentations différentes pour les rotations

- **Matrices de rotations**
- Quaternions
- Angle et axe de rotation
- Vecteurs de rotation
- Angles d'euler (roll,pitch,yaw)

Transformation en fonction de q

- À chaque $q \in \mathcal{Q}$ correspond une matrice 0T_E
- Architecture série : transformations successives :

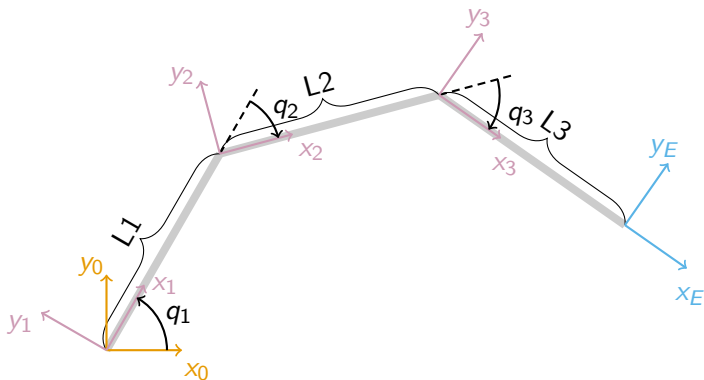
$${}^0T_E(q) = {}^0T_1(q_1) {}^1T_2(q_2) \dots {}^{n-1}T_n(q_n) {}^nT_E$$

- Chaque transformation peut encore être décomposée

$${}^i T_j = {}^i T_{j''} {}^{j''} T_{j'}(q_i) {}^{j'} T_j$$

- Rotoïde : ${}^{j''} T_{j'}(q_i) : \mathcal{R}(\vec{v}, q_i)$, avec \vec{v} l'axe de rotation
- Linéaire : ${}^{j''} T_{j'}(q_i) : \mathcal{T}(\vec{v}, q_i)$, avec \vec{v} l'axe de translation

Exemple : Bras à 3 degrés de libertés



$$\underbrace{
 \begin{matrix}
 {}^0 T_1(q_1) & {}^1 T_2(q_2) & {}^2 T_3(q_3) & {}^3 T_E \\
 \mathcal{R}(\vec{z}, q_1) & \mathcal{T}(\vec{x}, L1) & \mathcal{R}(\vec{z}, q_2) & \mathcal{T}(\vec{x}, L2) & \mathcal{R}(\vec{z}, q_3) & \mathcal{T}(\vec{x}, L3)
 \end{matrix}
 }_{{}^0 T_E(q)}$$

Notes sur l'espace opérationnel

- 0T_E contient des informations redondantes
 - 9 valeurs pour décrire l'orientation
 - 3 valeurs suffiraient
- 0T_E contient parfois des informations inutiles
 - Par exemple, intérêt uniquement pour la position (x,y)

Notation

- $\mathcal{P}({}^0T_E)$: transformation \rightarrow l'espace opérationnel
- $\mathcal{G}(q) = \mathcal{P}({}^0T_E(q))$
- $\mathcal{G}_i(q)$ Le i -ème élément du vecteur $\mathcal{G}(q)$

Quelques exemples de projection dans \mathcal{O}

Position et orientation

$$\mathcal{P}({}^0T_E) = \mathcal{P}\left(\begin{pmatrix} {}^0R_E & {}^0O_E \\ 0_{1 \times 3} & 1 \end{pmatrix}\right) = \left(\begin{array}{c} {}^0O_E \\ \left. \begin{array}{l} \text{roll} \\ \text{pitch} \\ \text{yaw} \end{array} \right\} {}^0R_E \end{array} \right)$$

Position uniquement

$$\mathcal{P}({}^0T_E) = \mathcal{P}\left(\begin{pmatrix} {}^0R_E & {}^0O_E \\ 0_{1 \times 3} & 1 \end{pmatrix}\right) = {}^0O_E$$