ROS 4.PRO

La robotique opensource
au service de mon industrie

Cours de Robotique Théorique

Jean-Baptiste Horel – Ingénieur roboticien

Programme

- Cours de ROS (16h, Y.M.) :
  - Introduction
  - Manipulation (avec Poppy Ergo Jr, le bras robotique)
- Cours de Théorie robotique (4h, J-B.H.)
- Cours de robotique mobile (12h, J-B.H.) :
  - Navigation (avec Turtlebot, le robot roulant)

ROS 4.PRO

Programme

La robotique opensource au service de mon industrie

1) Désactivez votre microphone
2) Répondez à l'appel ou aux questions par le chat ou l'audio
3) On utilisera des salles :
   - Salle principale pour le cours magistral et pour m'interpeller
   - Salles de réunion n°Y pour chaque groupe n°Y :
     - Pensez à rejoindre l'audio/tester l'écho
     - Toujours au moins 1 membre du groupe présent dans la salle
4) A la pause déjeuner je laisse BBB ouvert

Plan du cours

1) Modèle géométrique d’un robot
2) Probabilité pour la robotique

Téléchargement

Ajoutez le dépôt ***ros4pro_robotique_theorique :***

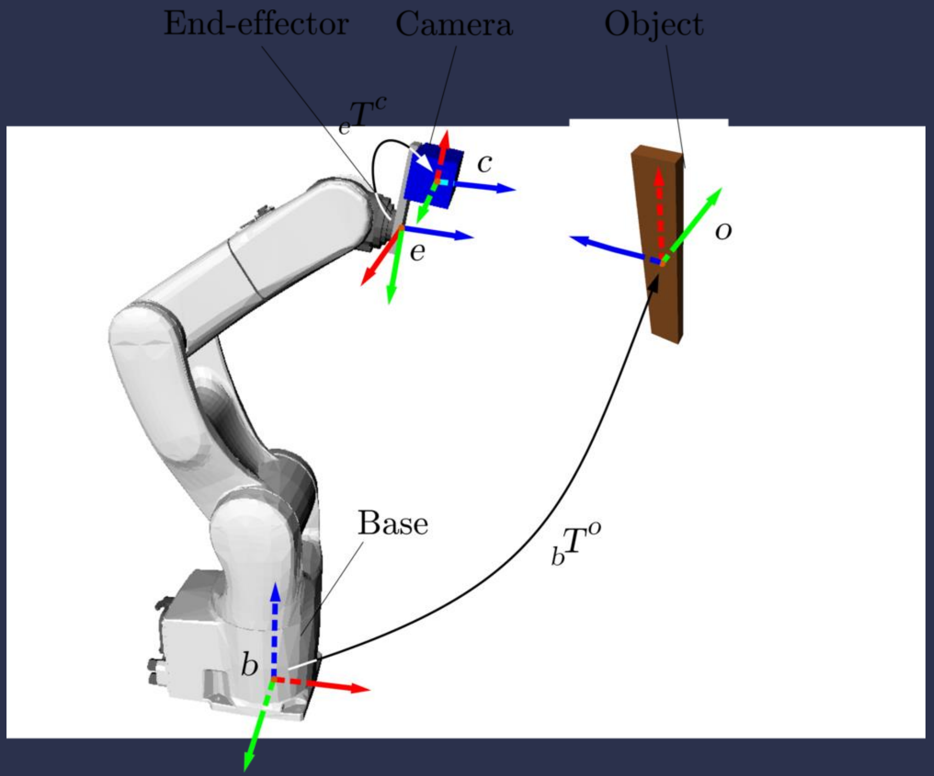
Dans le dossier ***~/catkin_ws/src/***

Exécutez la commande

git clone https://github.com/ros4pro/ros4pro_robotique_theorique.git

Modèle cinématique

Modèle géométrique de l’Homme

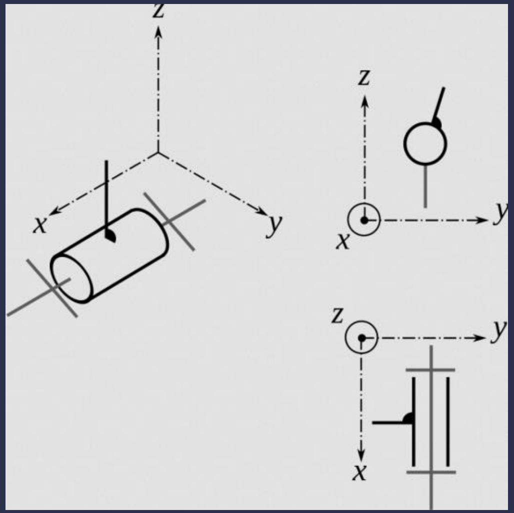


Application du modèle sur un bras

Liaisons cinématiques

Liaison pivot :

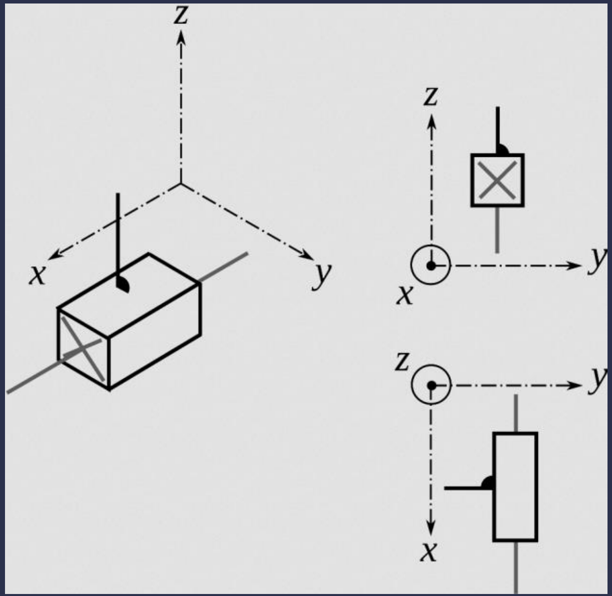
- 1 degré de liberté, rotation sur 1 axe
- Exemples : moteurs, le coude, pédale de vélo
- Liaison la plus fréquente en robotique



Liaisons cinématiques

Liaison glissière ou prismatique :

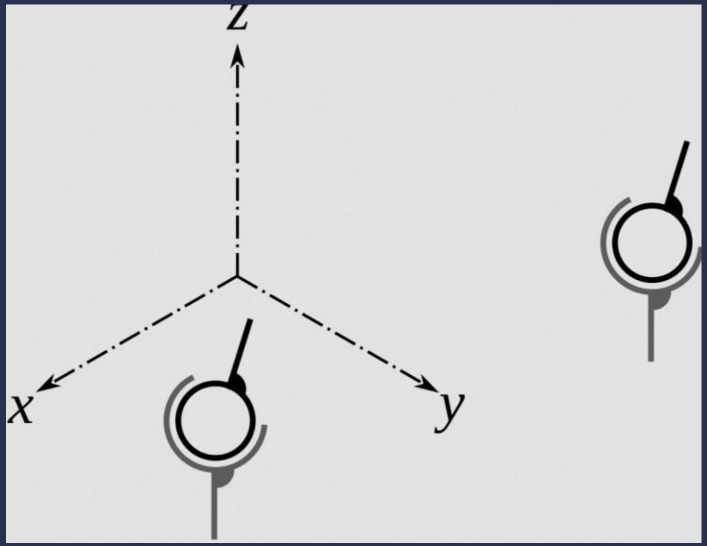
- 1 degré de liberté, translation sur 1 axe
- Exemples : vérin (peut avoir une rotation, pivot glissant), crémaillière

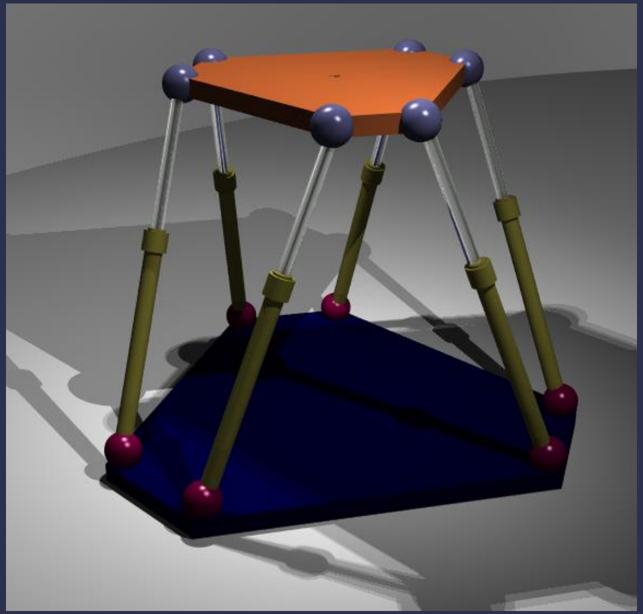


Liaisons cinématiques

Liaison rotule:

- 3 degrés de liberté, rotation sur 3 axes
- Exemples : le poignet, support de caméra





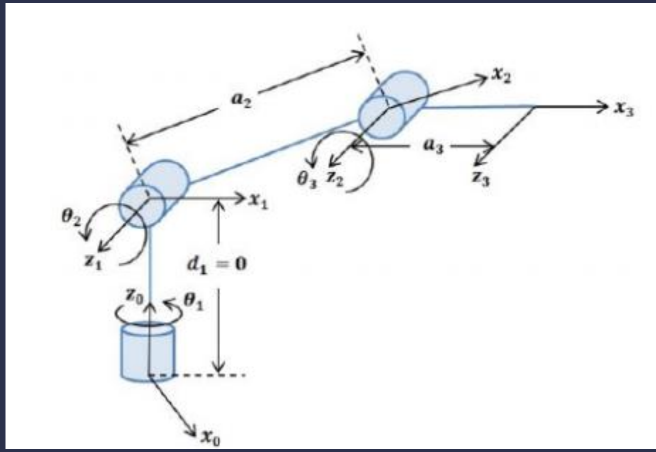
Construction du modèle cinématique

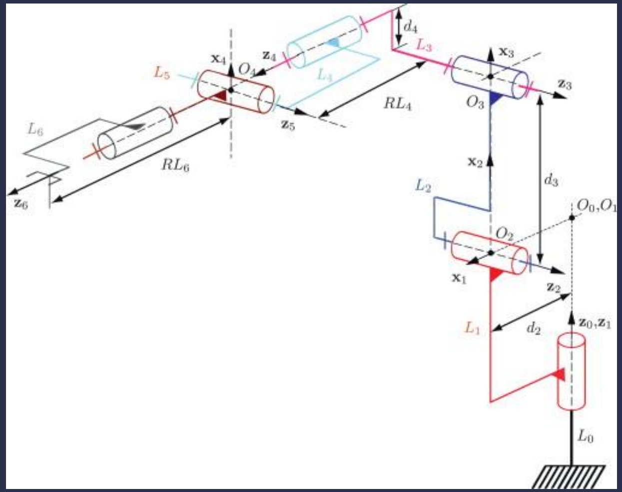
- Le robot est découpé en pièces reliées ensembles par des liaisons / joints
- Un repère est associé à chaque pièce
- Un type de liaison est associé à chaque articulation
- Facilite les calculs,
- Deux grandes familles de robots:
  - Robots Séries
  - Robots Parallèles

Construction du modèle cinématique

Série :

- Articulations en série, les plus fréquents. Chaîne cinématique ouverte.
- Grand espace de travail, polyvalent
- Accumulation des erreurs de chaque moteur, les moteurs supportent le poids des moteurs suivants

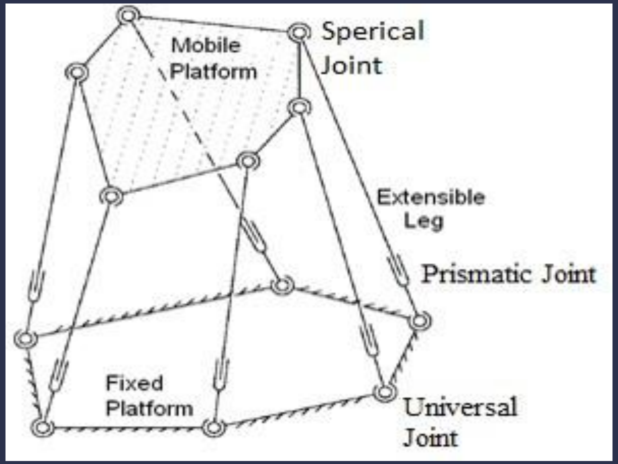


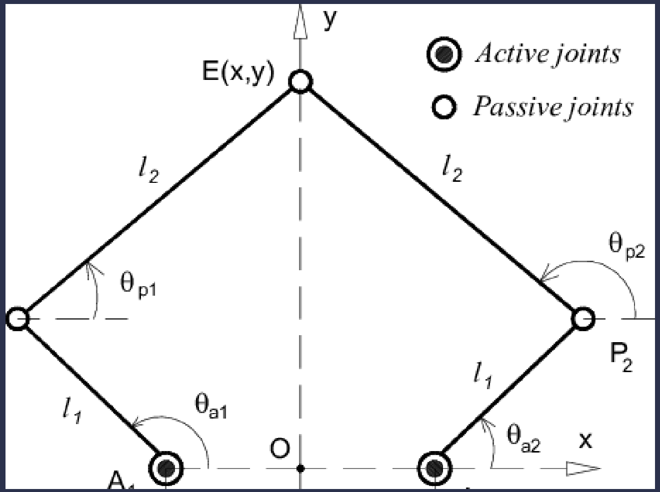


Construction du modèle cinématique

Robot Parallèle :

- Articulations en parallèle. Chaine cinématique fermée.
- Haute précision, haute vitesse, meilleur répartition du poids, poids de la charge de travail plus élevée
- Petit espace de travail, contrôle complexe

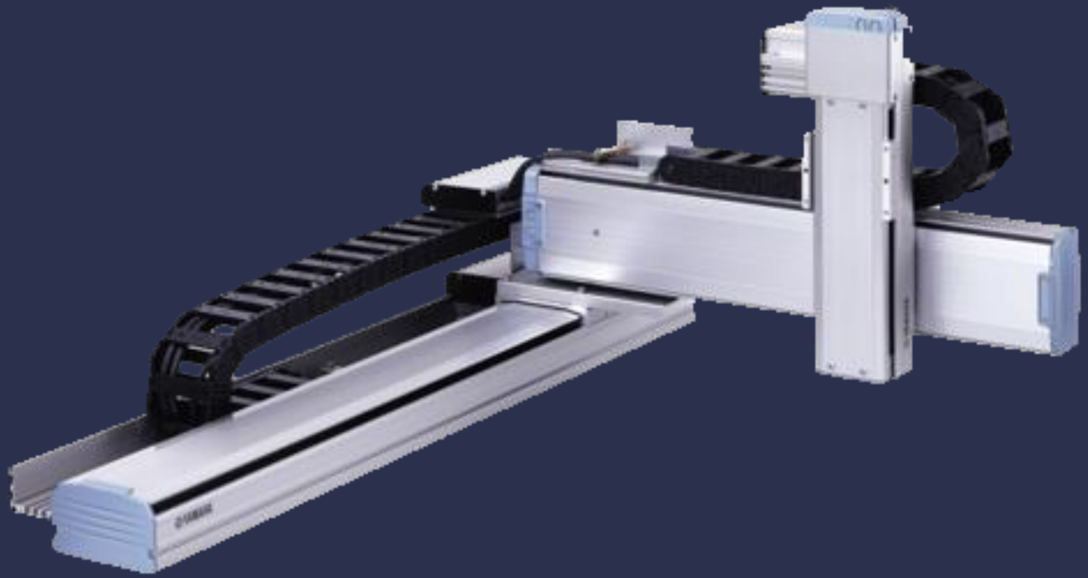


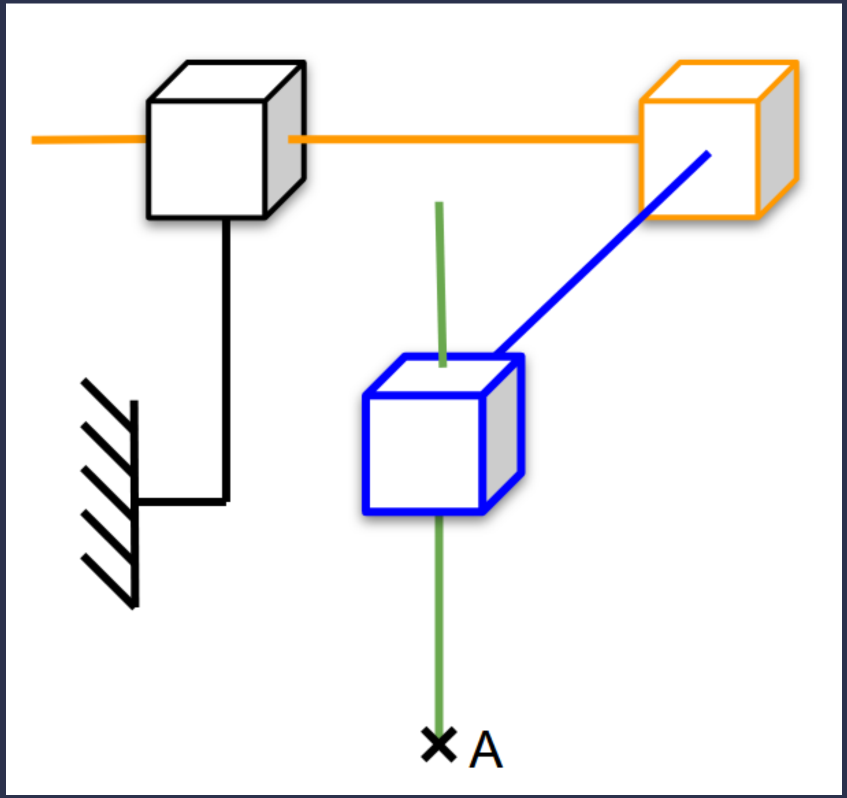


Exemple : schéma d'un robot cartésien

https://www.youtube.com/watch?v=cqQoWbhRXhw&feature=emb_logo

Exemple : schéma d’un robot cartésien



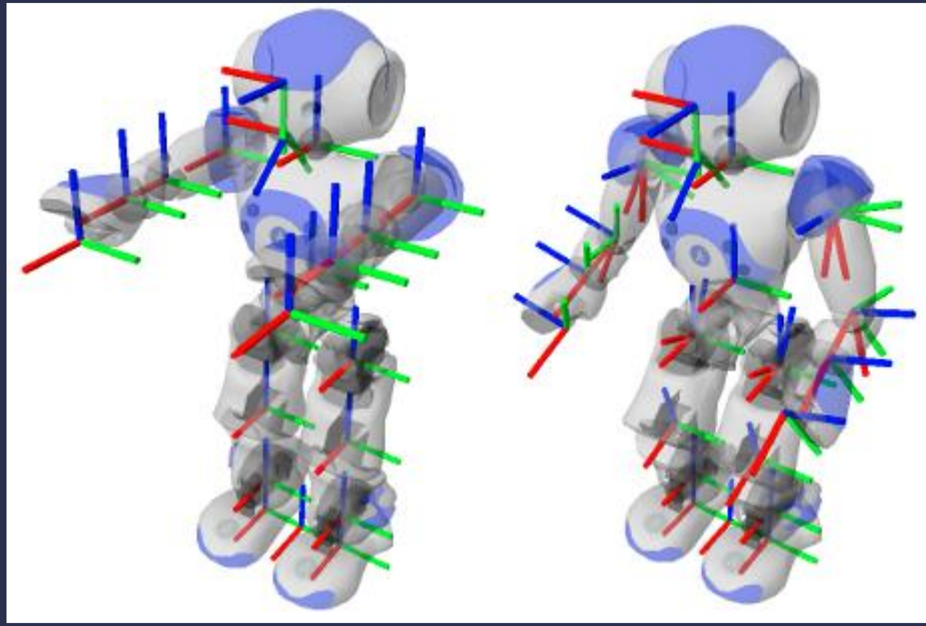


Exercice : schéma d'un SCARA

- https://www.youtube.com/watch?v=vKD20BTkXhk
- Dessinez le modèle du SCARA

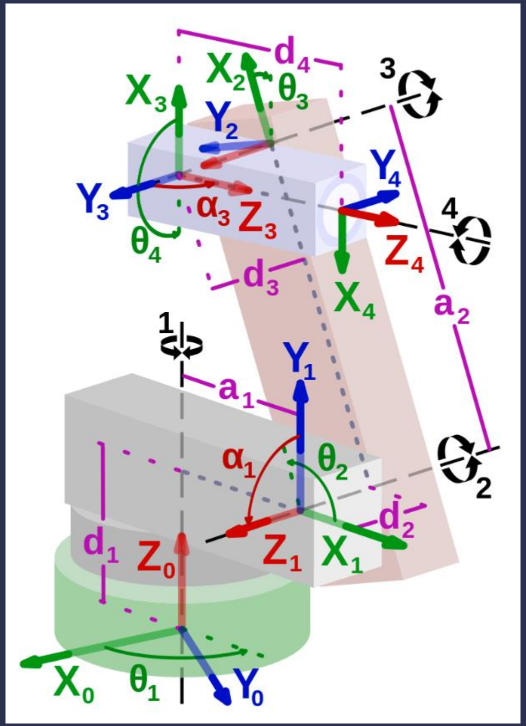
Système de référentiel

- Chaque pièce du robot possède un repère lié à l'articulation.
- Repère principal : *base_link*, utilisé pour la localisation et la navigation.
- Les capteurs et les pièces sont en mouvements, leurs données doivent être transformées dans un même repère.



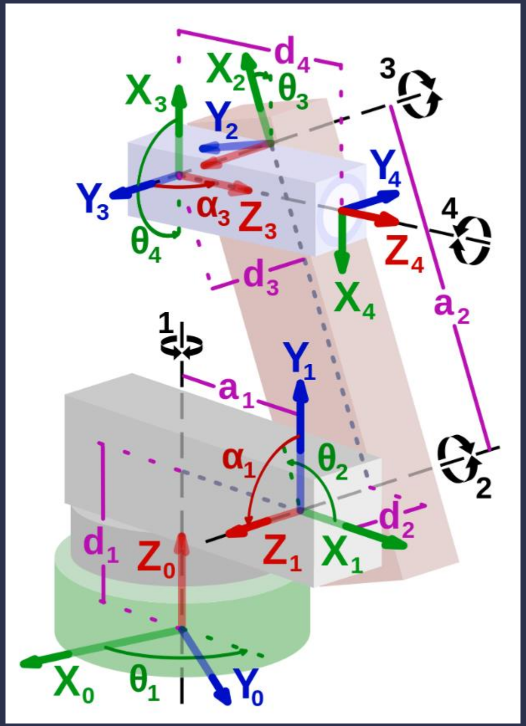
Convention de Denavit-Hartenberg

- Conventions de définitions des repères et des transformations
- Modélisation de la chaîne cinématique avec des pivots et des glissières
- Calcul matriciel simplifié pour les changements de repères
- Pour chaque repère est calculée la transformation T du repère *n* vers *n-1*



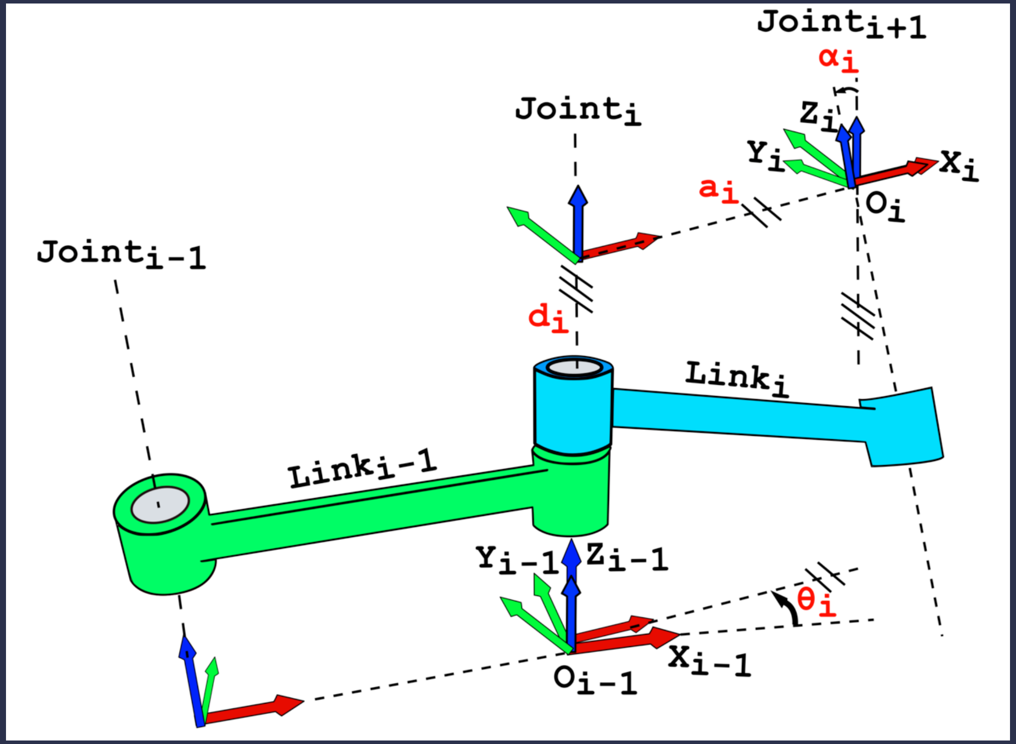
Convention de Denavit-Hartenberg

- axe Zi dans l'axe de la liaison i -> i+1
- axe Xi normal à Zi-1 et Zi
- axe Yi normal à Zi et Xi
- Oi, centre du repère i, sur Zi de tel sorte que Xi intersecte Zi-1



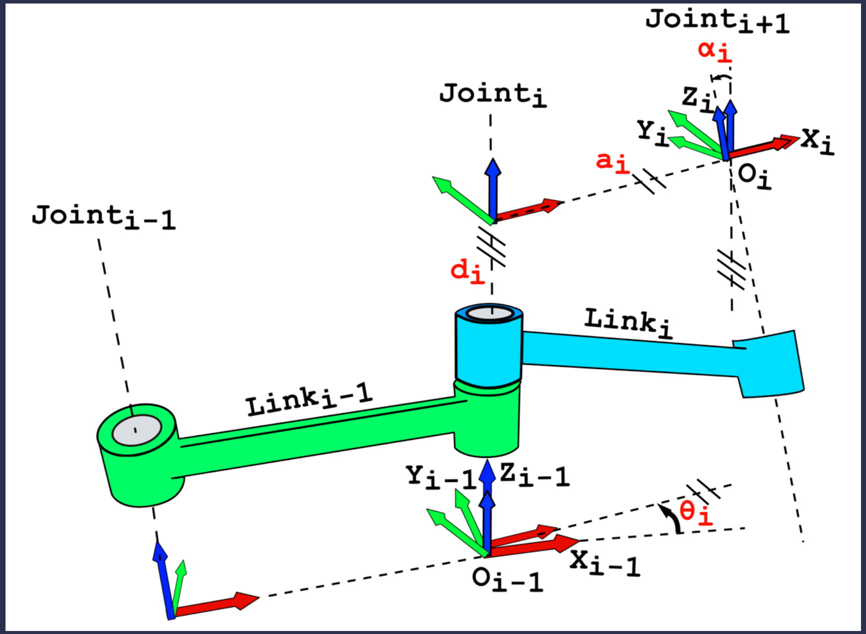
Convention de Denavit-Hartenberg

- axe Zi dans l'axe de la liaison i -> i+1
- axe Xi normal à Zi-1 et Zi
- axe Yi normal à Zi et Xi
- Oi, centre du repère i, sur Zi de tel sorte que Xi intersecte Zi-1

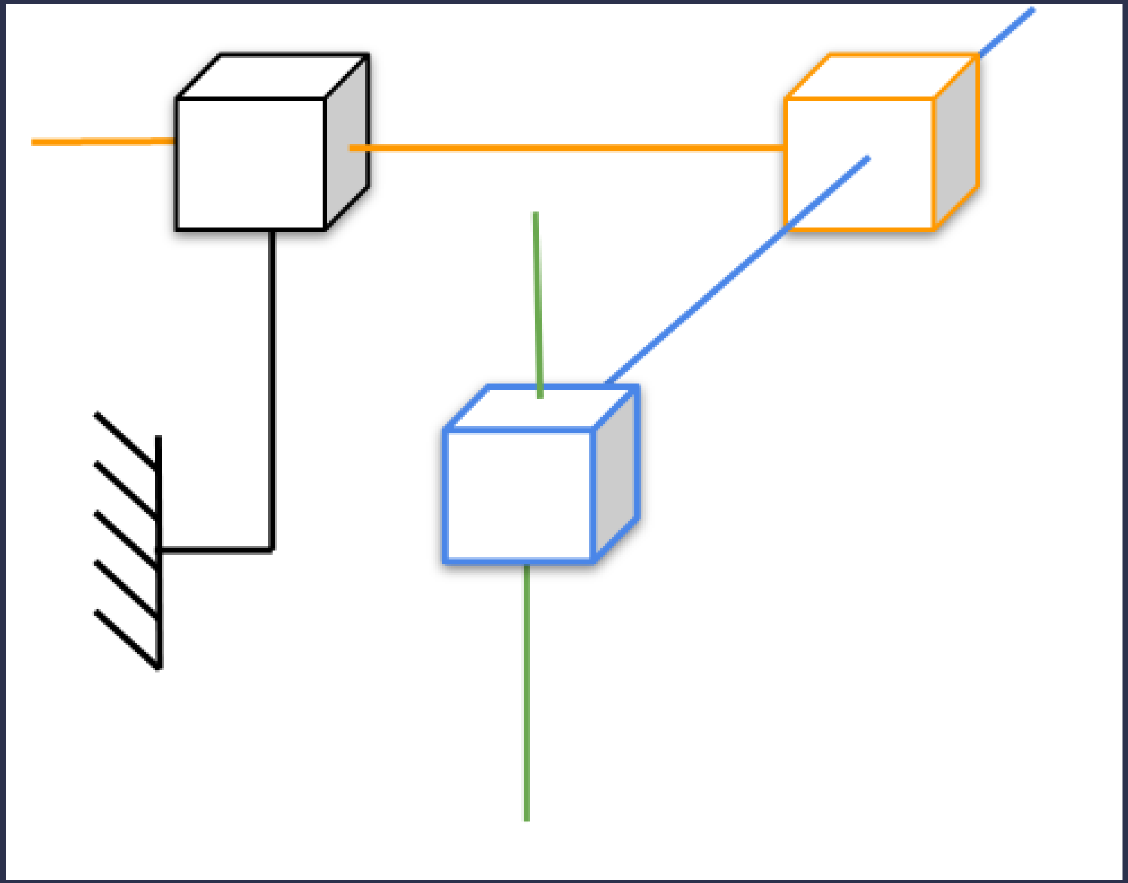


Convention de Denavit-Hartenberg

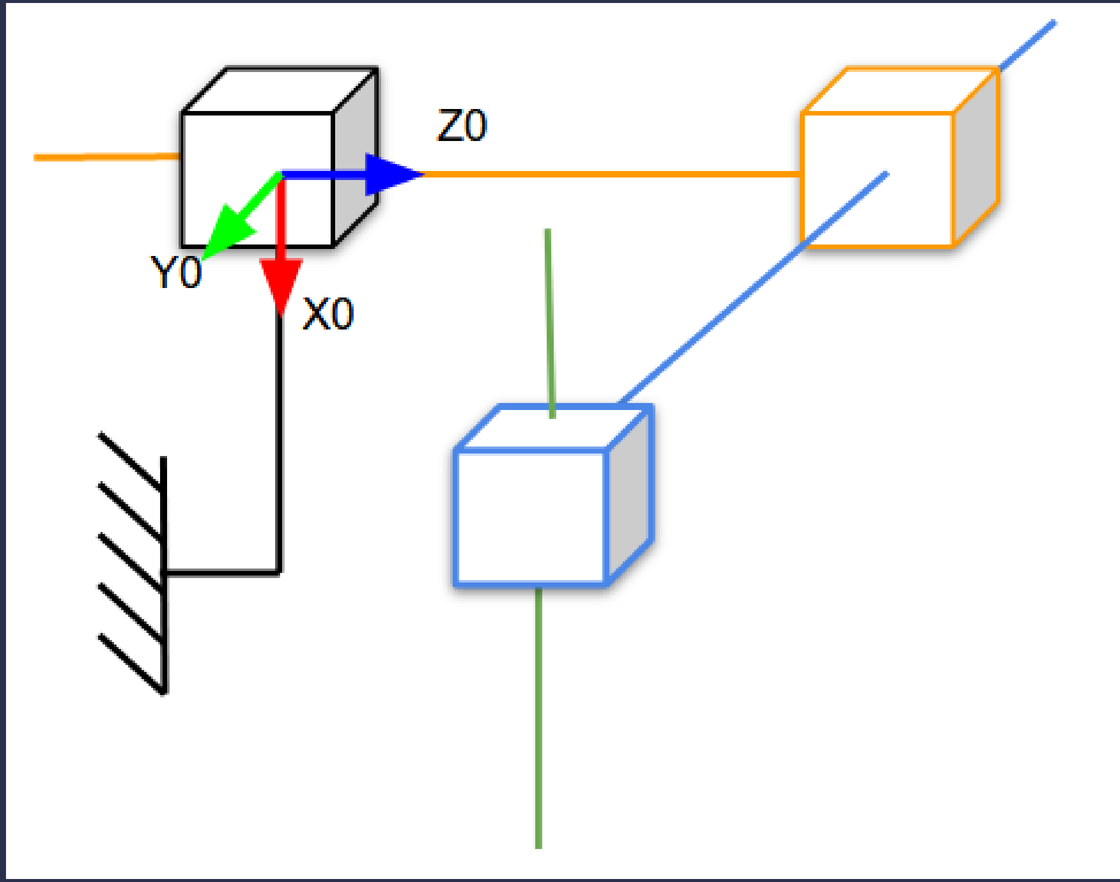
- Matrice de changement de repère de n à n-1 :
- $T_n^{n-1} = \mathrm{Trans}_{z_{n-1}}(d_n) \cdot \mathrm{Rot}_{z_{n-1}}(\theta_n) \cdot \mathrm{Trans}_{x_n}(r_n) \cdot \mathrm{Rot}_{x_n}(\alpha_n)$
- $T_n^{n-1} = \begin{pmatrix} \cos\theta_n & -\sin\theta_n \cos\alpha_n & \sin\theta_n \sin\alpha_n & r_n \cos\theta_n \\ \sin\theta_n & \cos\theta_n \cos\alpha_n & -\cos\theta_n \sin\alpha_n & r_n \sin\theta_n \\ 0 & \sin\alpha_n & \cos\alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{pmatrix}$
- $\mathrm{X}_0 = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} = \mathrm{T}_1^0 \cdot \mathrm{T}_2^1 \cdot \mathrm{T}_3^2 \cdot \mathrm{T}_4^3 \cdot \begin{pmatrix} x_4 \\ y_4 \\ z_4 \end{pmatrix}$



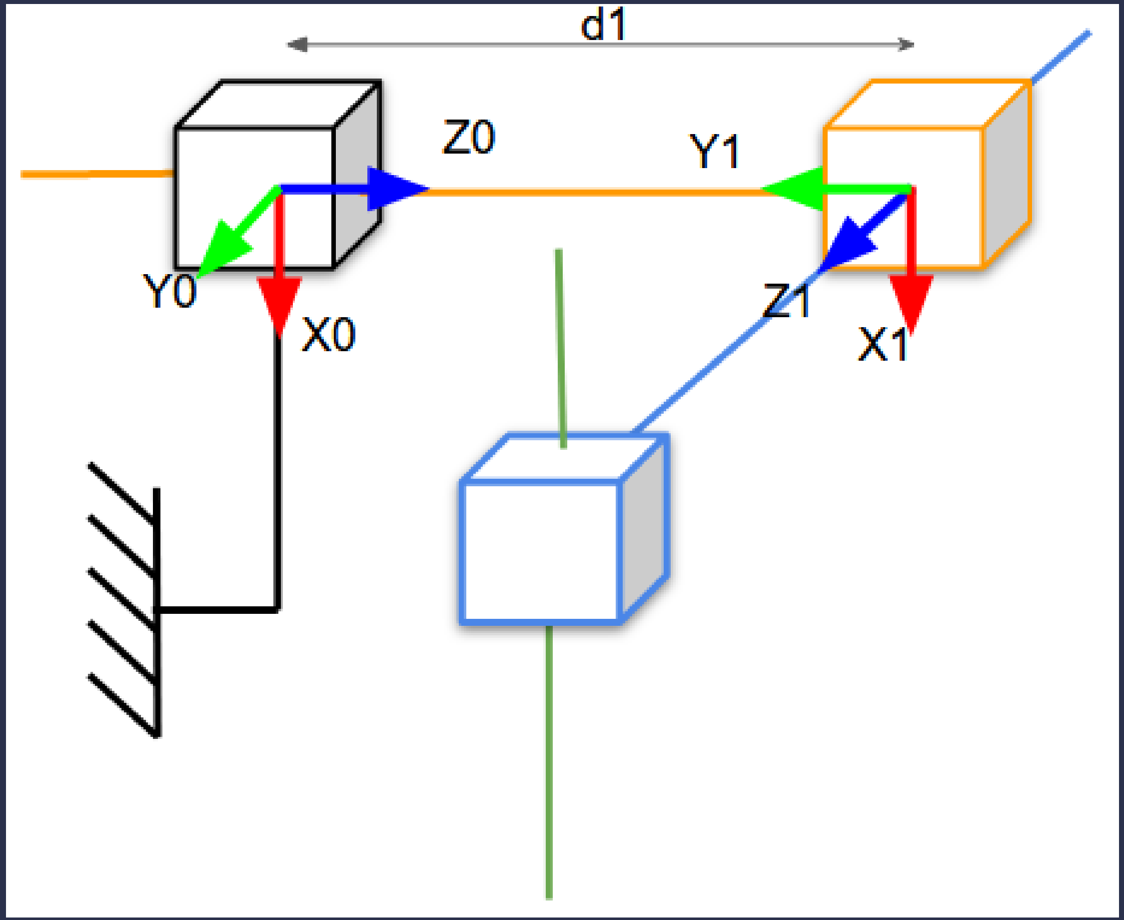
Exemple : système de référentiel d’un robot cartésien



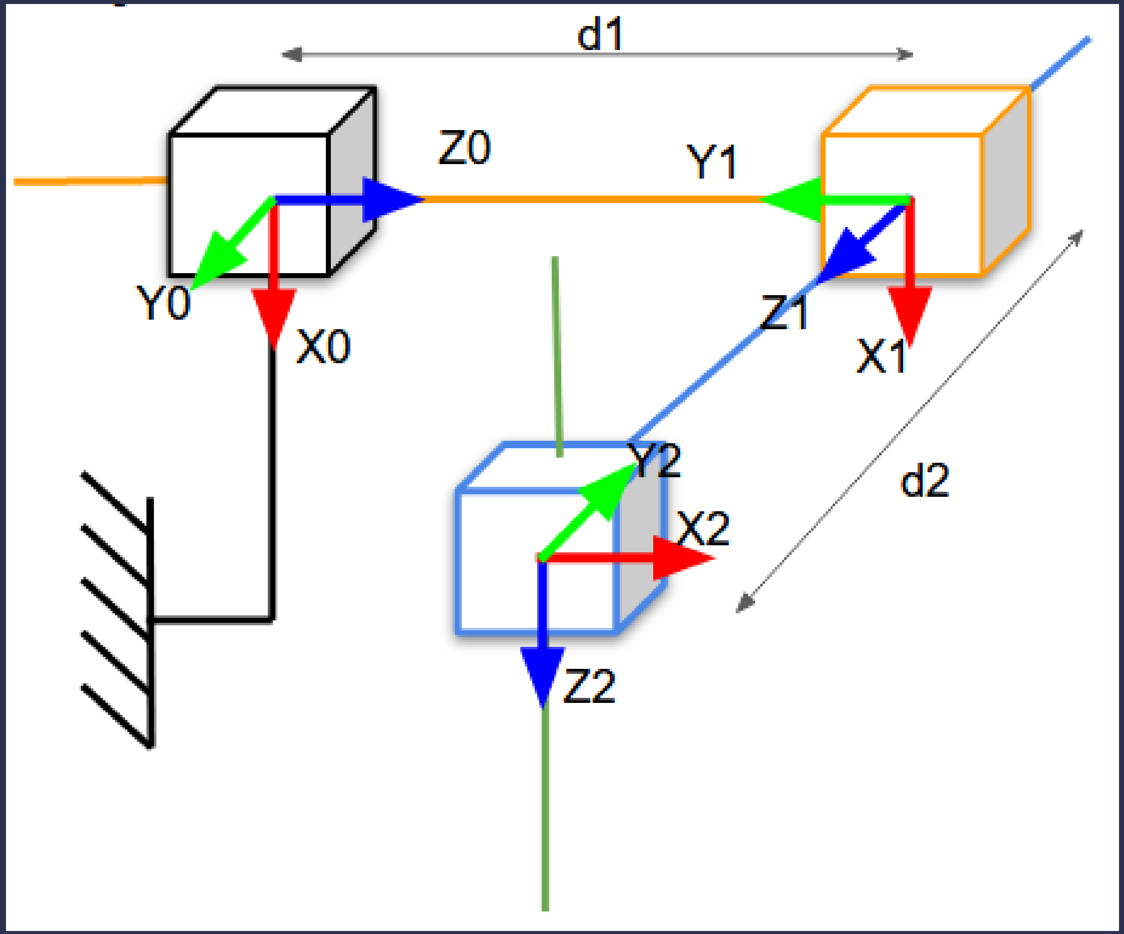
Exemple : système de référentiel d'un robot cartésien



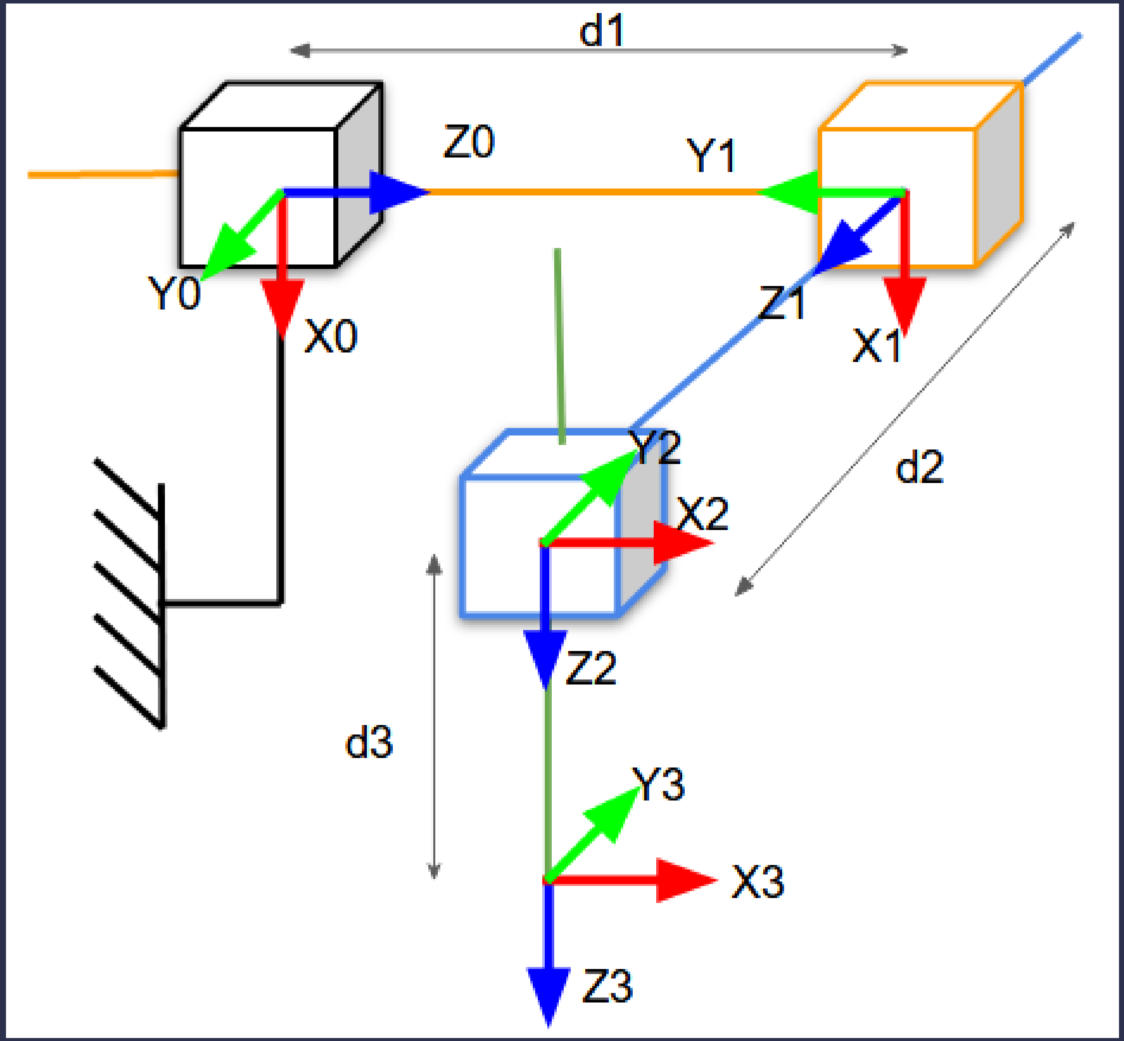
Exemple : système de référentiel d’un robot cartésien



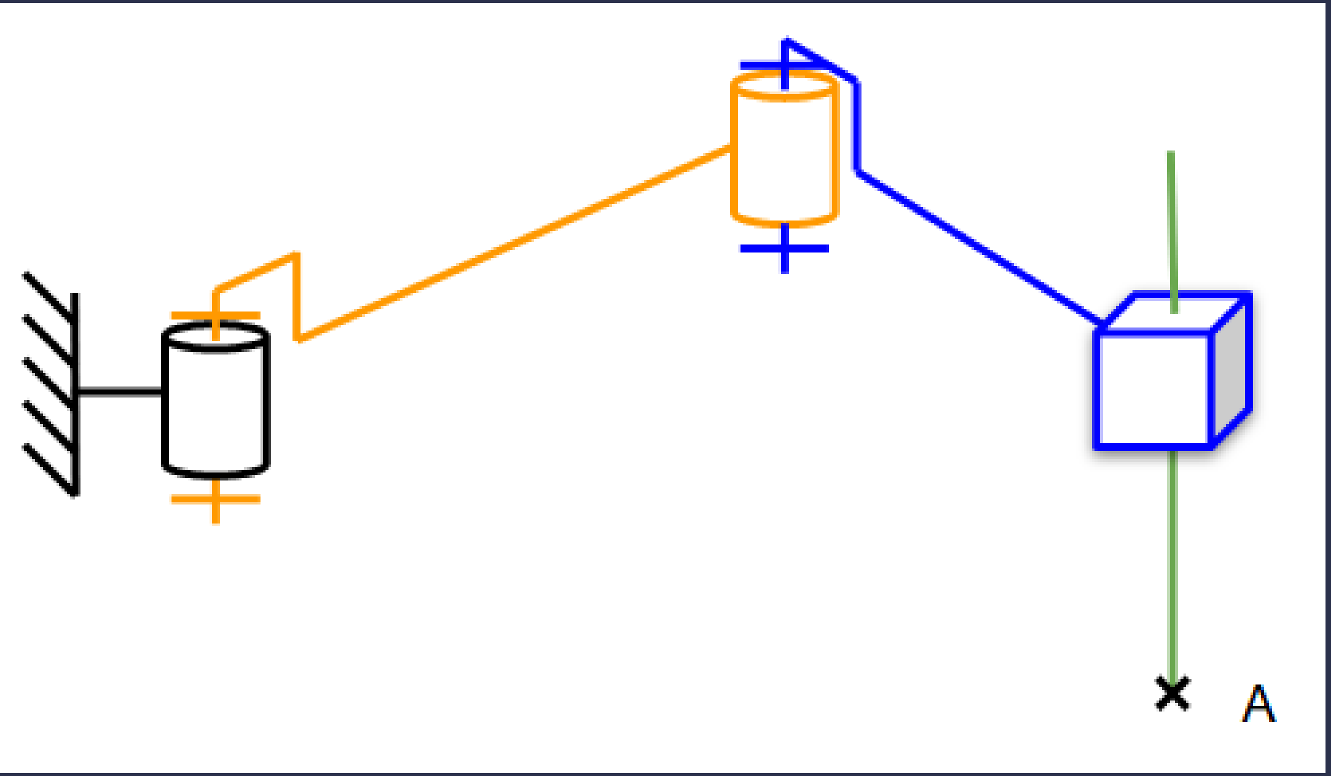
Exemple : système de référentiel d'un robot cartésien



Exemple : système de référentiel d'un robot cartésien



Exercice : système de référentiel d'un SCARA

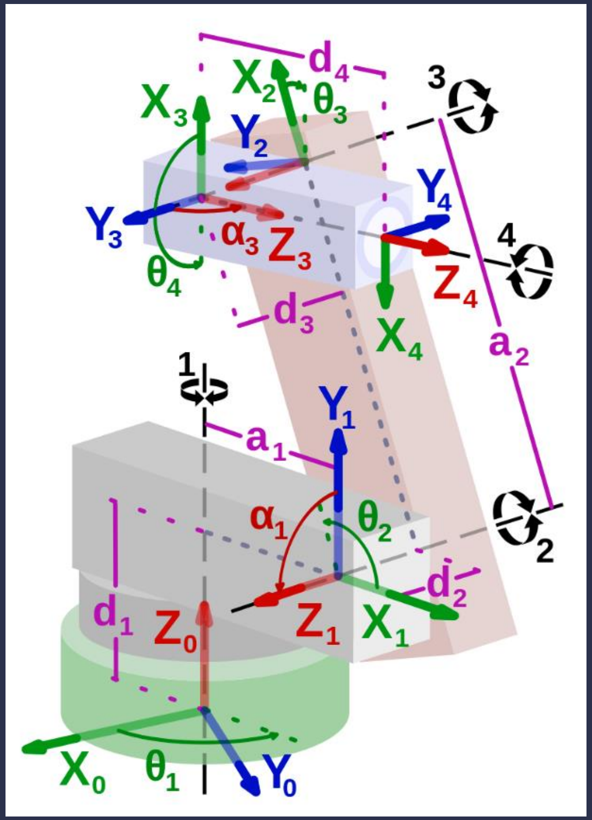


Cinématique directe

- A partir de la position de chaque articulation calculer la position et l’orientation de l’effecteur.
  - angle pour les pivots et longueur pour les glissières
  - pose de l’effecteur dans le repère de la base du robot.
- (x, y, z, r, p, y) = F(a1, a2…an)

Espace articulaire (angles des articulations)

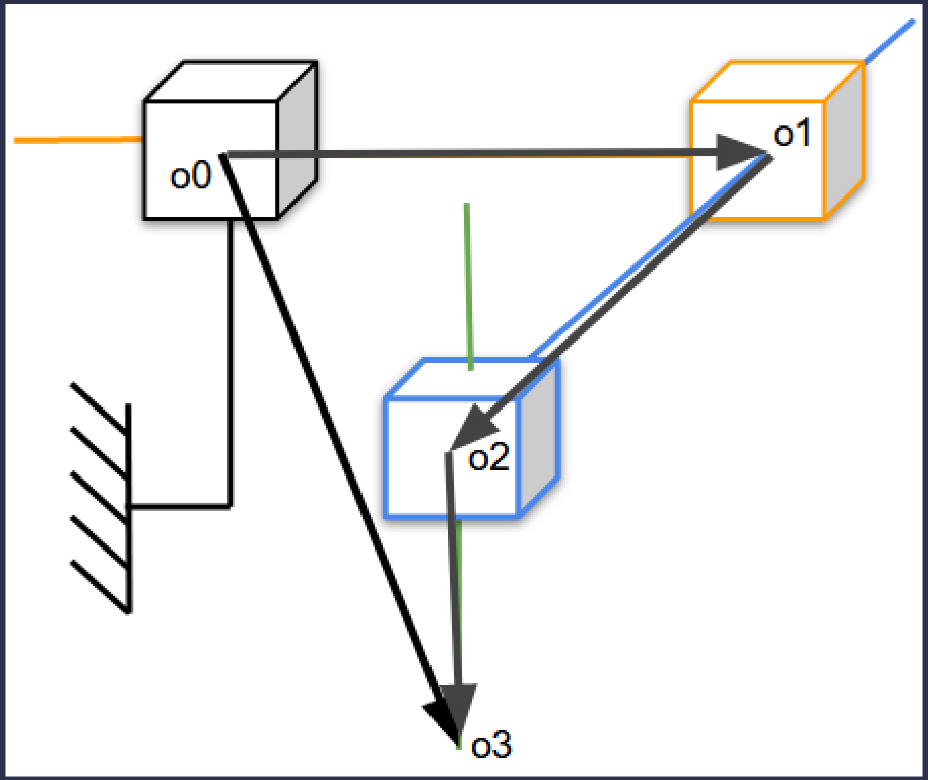
→ espace cartésien (pose de l’effecteur)



Cinématique directe

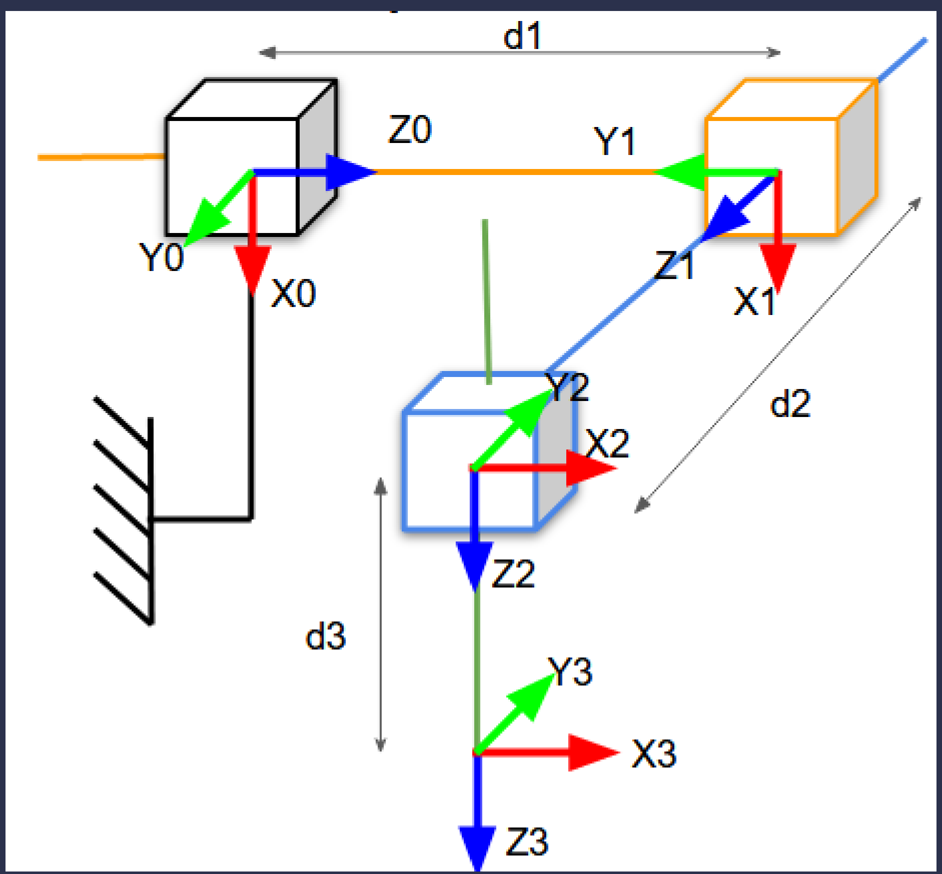
- Connaître relativement à n'importe quel repère (principalement base_link) la pose de l'effecteur : orientation et position.
- Pour connaître la géométrie du robot, de manière générale la pose de chaque pièce.
- Facile sur les robots série en utilisant DH, ce sont des multiplications de matrices et des calculs de cos sin.
- Complexe voir impossible sans algorithme numérique pour les robots parallèles.

Exemple : cinématique directe d'un robot cartésien



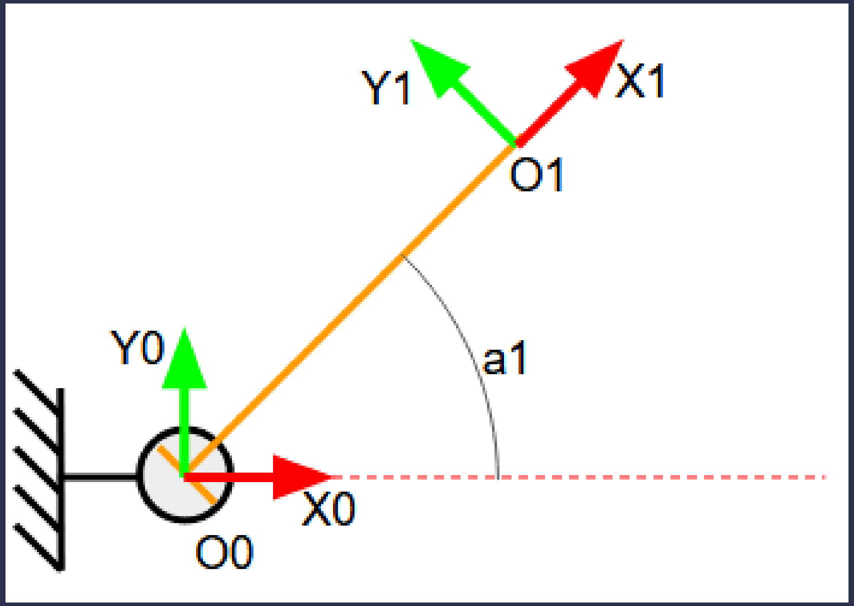
- Les coordonnées $O_3 = (x, y, z)$dans le repère $\mathrm{R}_0, (\overrightarrow{X_0}, \overrightarrow{Y_0}, \overrightarrow{Z_0})$sont celles du vecteur $\overrightarrow{O_0 O_3}$ dans ce même repère
- $\overrightarrow{O_0 O_3} = x \cdot \overrightarrow{X_0} + y \cdot \overrightarrow{Y_0} + z \cdot \overrightarrow{Z_0}$
- On décompose le vecteur en chaîne de vecteurs plus simples à calculer indépendamment.
- On exprime chaque vecteur dans son repère proche puis on effectue des changements de repères pour tout exprimer dans R_0.

Exemple : cinématique directe d'un robot cartésien



- On cherche (x, y, z) tels que :
  - $\overrightarrow{O_0O_3} = x \cdot \overrightarrow{X_0} + y \cdot \overrightarrow{Y_0} + z \cdot \overrightarrow{Z_0}$
- $\overrightarrow{O_0O_3} = \overrightarrow{O_0O_1} + \overrightarrow{O_1O_2} + \overrightarrow{O_2O_3}$
- $\overrightarrow{O_0O_3} = -d_1 \cdot \overrightarrow{Y_1} - d_2 \cdot \overrightarrow{Y_2} + d_3 \cdot \overrightarrow{Z_3}$
- $\overrightarrow{Y_1} = -\overrightarrow{Z_0}$
- $\overrightarrow{Y_2} = -\overrightarrow{Z_1}$ et $\overrightarrow{Z_1} = \overrightarrow{Y_0}$ donc $\overrightarrow{Y_2} = -\overrightarrow{Y_0}$
- $\overrightarrow{Z_3} = \overrightarrow{Z_2} = \overrightarrow{X_1} = \overrightarrow{X_0}$
- $\overrightarrow{O_0O_3} = d_3 \cdot \overrightarrow{X_0} + d_2 \cdot \overrightarrow{Y_0} + d_1 \cdot \overrightarrow{Z_0}$

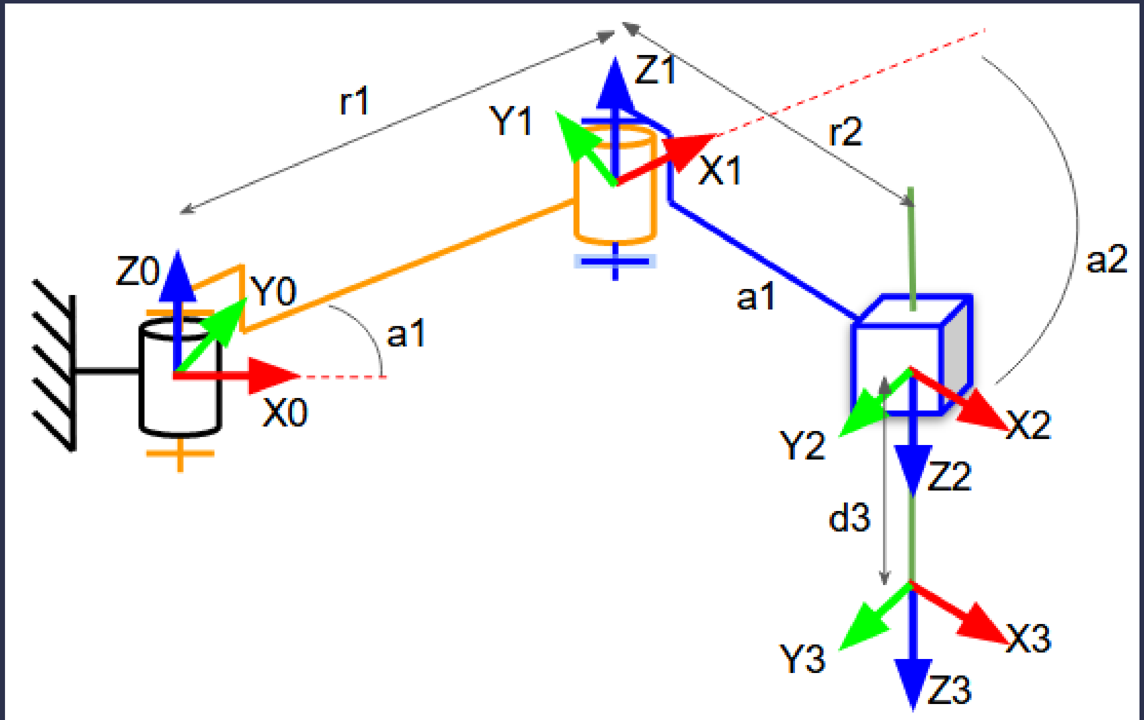
Exemple : cinématique directe d'un pivot



- On cherche (x, y) tels que :
  - $\overrightarrow{O_0O_1} = x \cdot \overrightarrow{X_0} + y \cdot \overrightarrow{Y_0}$
- $\overrightarrow{O_0O_1} = r_1\cos(\alpha_1) \cdot \overrightarrow{X_0} + r_1\sin(\alpha_1) \cdot \overrightarrow{Y_0}$

- $\overrightarrow{O_0O_1} = r_1 \cdot \overrightarrow{X_1}$ et
  $\overrightarrow{X_1} = \cos(\alpha_1) \cdot \overrightarrow{X_0} + \sin(\alpha_1) \cdot \overrightarrow{Y_0}$
- $\overrightarrow{O_0O_1} = r_1(\cos(\alpha_1) \cdot \overrightarrow{X_0} + \sin(\alpha_1) \cdot \overrightarrow{Y_0})$

Exercice : Cinématique directe du SCARA



- Inspirez vous des deux exemples précédents.
- Le problème se découpe en deux parties :
  - Les liaisons 2 et 3 influent sur X et Y
  - a liaison 3 influe sur Z
- Testez votre modèle avec *geometrique_direct.py*, suivez le README.md du dépôt.

Package ROS TF2

- ROS utilise la bibliothèque TF2 (transform version 2)
- Calcul les transformations entre les repères, comme DH et KK, à partir d’un fichier de description des articulations au format urdf / xml
- Partage réseau des données entre les noeuds ROS.
- API de calcul géométrique et matriciel.

Cinématique Inverse

- C'est la fonction inverse de la cinématique directe.
- Calculer la position de chaque articulations pour que l'effecteur ait la pose souhaitée.
- $(\alpha_1, \alpha_2 \ldots \alpha_n) = F(x,\ y,\ z,\ roll,\ pitch,\ yaw)$

Espace cartésien (pose de l'effecteur)

➡ Espace articulaire (angles des articulations)

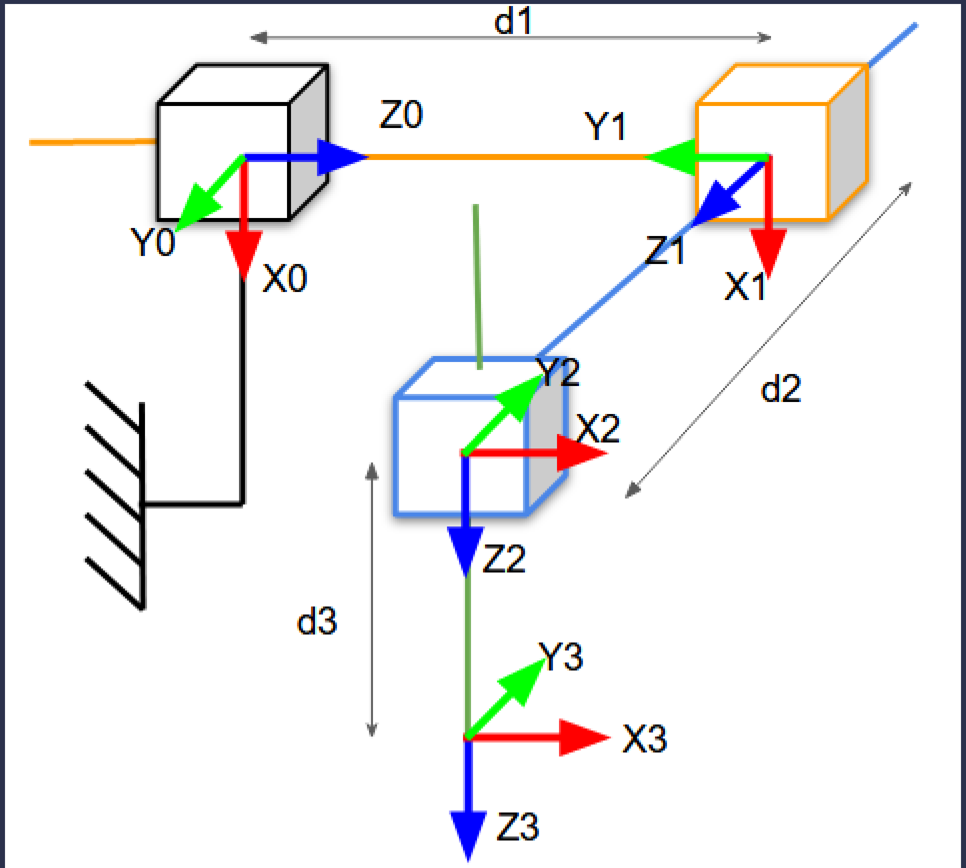
Exemple : Cinématique Inverse d'un robot cartésien

- C'est la fonction inverse de la cinématique directe.
- Calculer la position de chaque articulations pour que l'effecteur ait la pose souhaitée.
- $(\alpha_1, \alpha_2 \ldots \alpha_n) = F(x,\ y,\ z,\ roll,\ pitch,\ yaw)$

Espace cartésien (pose de l'effecteur)

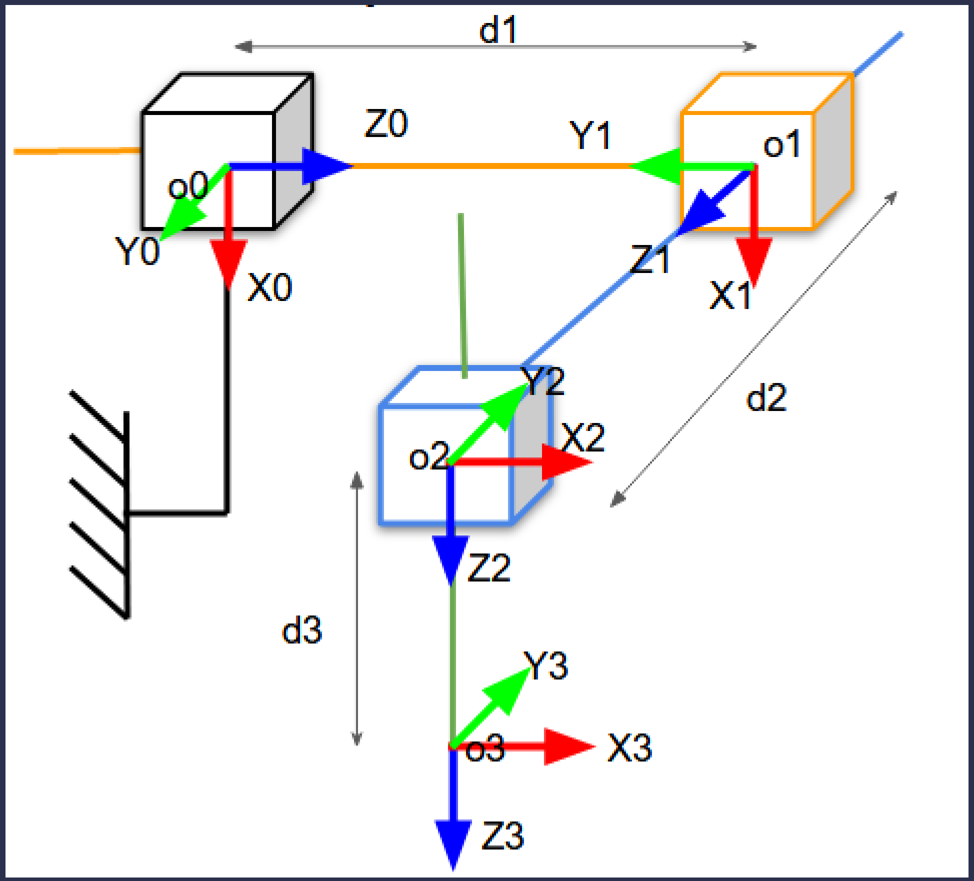
➡ Espace articulaire (angles des articulations)

Exemple : Cinématique Inverse d'un robot cartésien



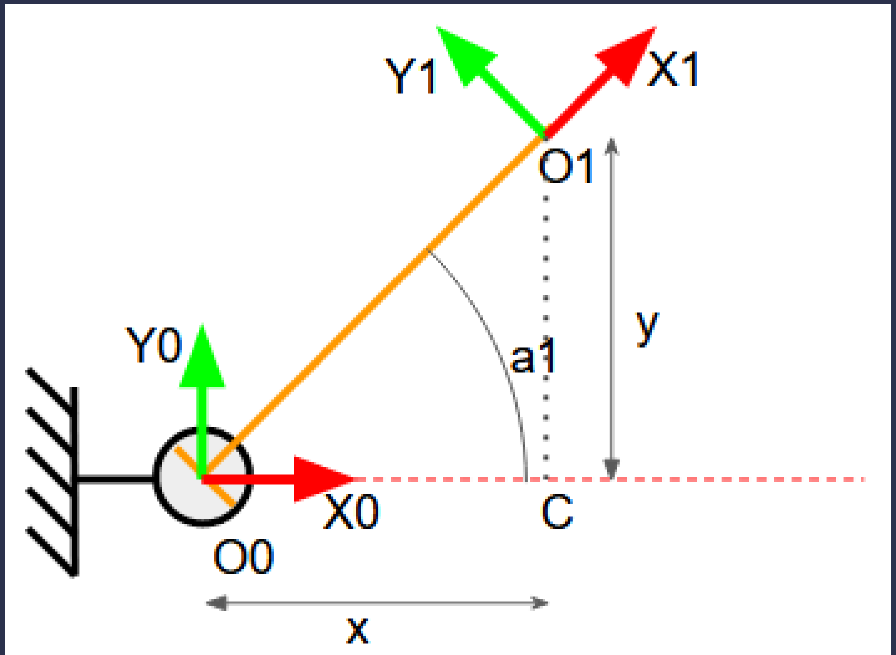
- Chercher des équations qui lient les variables d'entrée (pose de l'effecteur) à celles de sortie (positions des articulations).
- (d1, d2, d3) = F(x, y, z)

Exemple : Cinématique Inverse d'un robot cartésien



- $x^2 + y^2 + z^2 = d_1^2 + d_2^2 + d_3^2$
- $x^2 + y^2 = d_1^2 + d_2^2$
- $\overrightarrow{O_0O_3} = d_3 \cdot \overrightarrow{X_0} + d_2 \cdot \overrightarrow{Y_0} + d_1 \cdot \overrightarrow{Z_0}$
- $x \cdot \overrightarrow{X_0} + y \cdot \overrightarrow{Y_0} + z \cdot \overrightarrow{Z_0} = d_3 \cdot \overrightarrow{X_0} + d_2 \cdot \overrightarrow{Y_0} + d_1 \cdot \overrightarrow{Z_0}$
- $(d_1, d_2, d_3) = (z, y, x)$

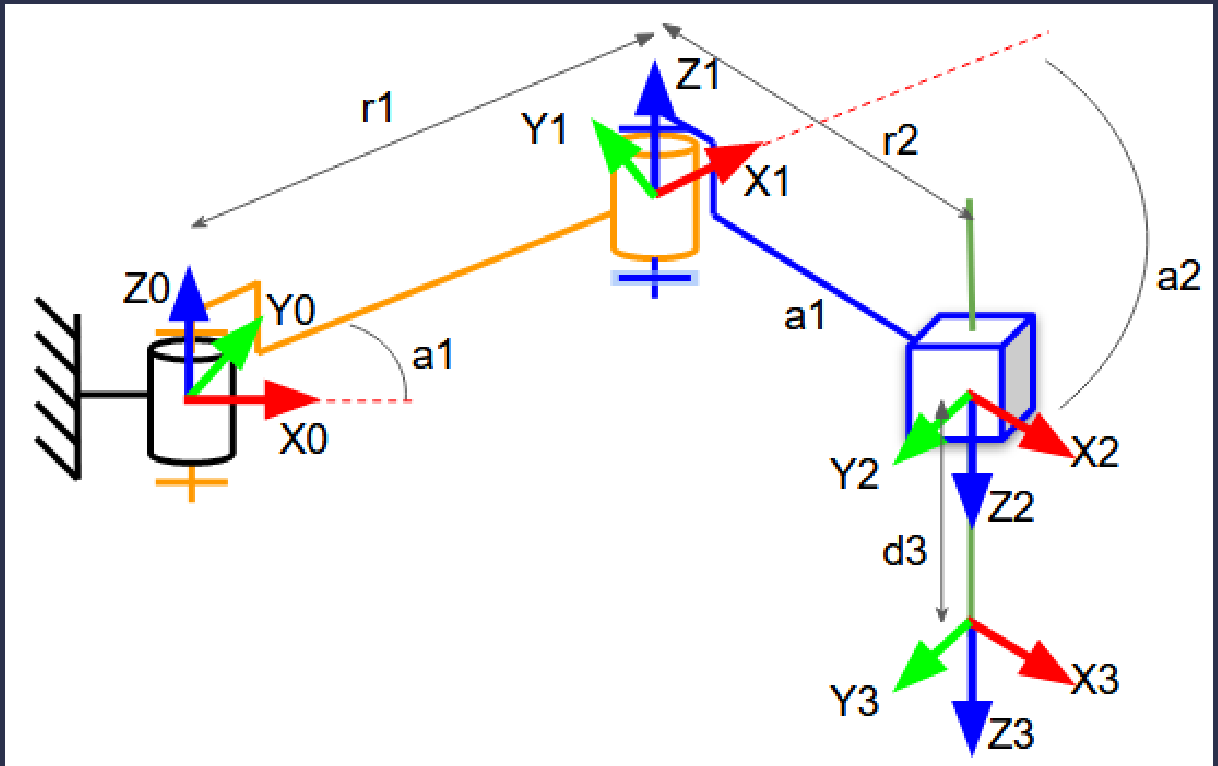
Exemple : Cinématique Inverse d’un pivot



- $\cos(\alpha_1) = {}^{x}/_{r_1}$ et $\sin(\alpha_1) = {}^{y}/_{r_1}$
- $\alpha_1 = \mathrm{acos}({}^{x}/_{r_1})$ et $\alpha_1 = \mathrm{asin}({}^{y}/_{r_1})$

- $\tan(a_1) = {}^{y}/_{x}$
- $a_1 = \mathrm{atan2}(y, x)$

Exercice : cinématique inverse du SCARA



- Inspirez vous des deux exemples précédents.
- Le problème se découpe en deux parties :
  - Les liaisons 2 et 3 influent sur X et Y
  - a liaison 3 influe sur Z
- Testez votre modèle avec *geometrique_inverse.py*, suivez le README.md du dépôt.

Probabilité pour la robotique

- Etude de trois exemples liés à la localisation :
- Filtre à histogramme : environnement représenté par un damier
- Filtre de Kalman : localisation représentée par une gaussienne
- Filtre particulaire : localisation représentée par des particules

Probabilité en robotique

Etude de trois exemples liés à la localisation :

- Filtre à histogramme : environnement représenté par un damier
- Filtre de Kalman : localisation représentée par une gaussienne
- Filtre particulaire : localisation représentée par des particules

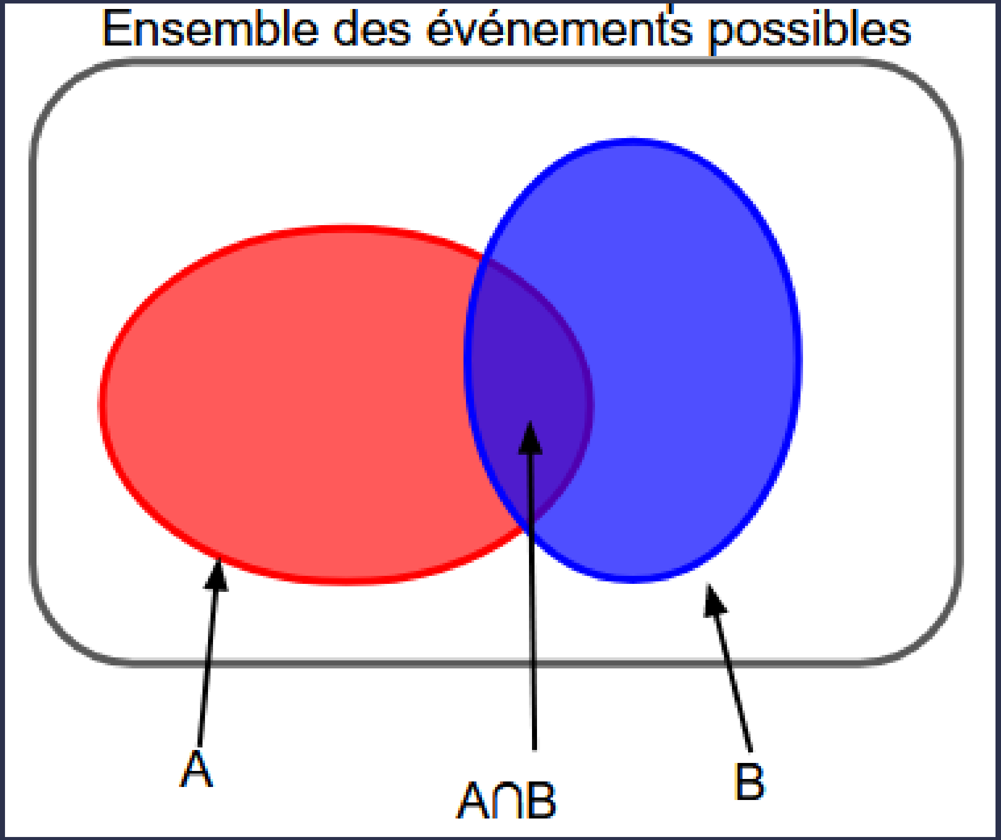
Pourquoi faire des probabilités ?

- Les informations traitées par un robot ne sont pas fiables (bruits et incertitudes des capteurs).
- Les déplacements ne sont pas fiables (patinement des roues, jeu mécanique du bras)
- Les algorithmes ont une incertitude : machine learning, analyse d'image et algorithmes numériques

Pourquoi faire des probabilités ?

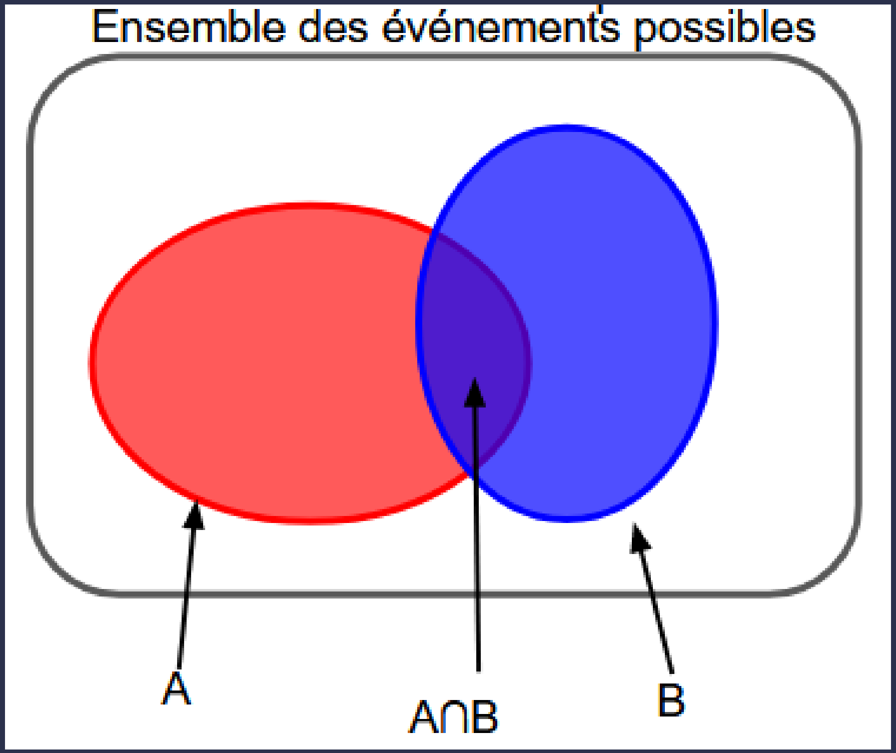
- Le plus possible, représenter par des probabilités (variables aléatoires) et y associer une fonction de probabilité.
- Il faut modéliser l'incertitude des capteurs
- Il faut représenter l'état du robot et son environnement de manière probabiliste :
  - Localisation du robot
  - Détection, segmentation, identification
  - Tracking d'objets

Rappel de probabilités : probabilité conditionnelles



- Probabilité de l'événement A sachant que B s'est produit :
  - $P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A,B)}{P(B)}$
- Quand A et B ne sont pas indépendants, toute information sur B nous renseigne sur l'événement A.

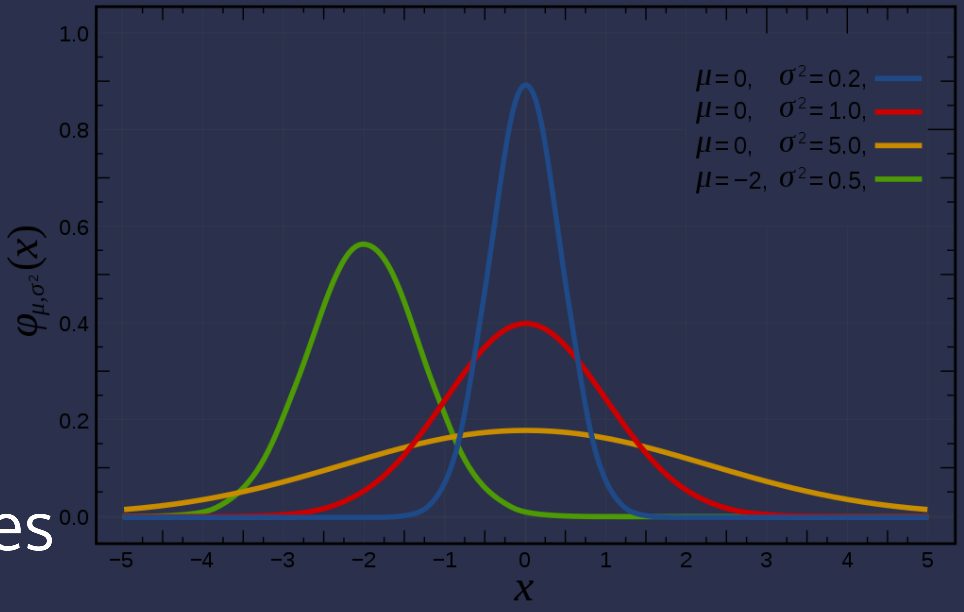
Rappel de probabilités : probabilité conditionnelles



- $P(A, B) = P(B, A)$
- $P(A, B) = P(A|B).P(B) = P(B|A).P(A)$
- $P(A) = \sum_n P(A, B_n) = \sum_n P(A|B_n).P(B_n)$
  - Avec $(B_1 \dots B_n)$ indépendant et complet

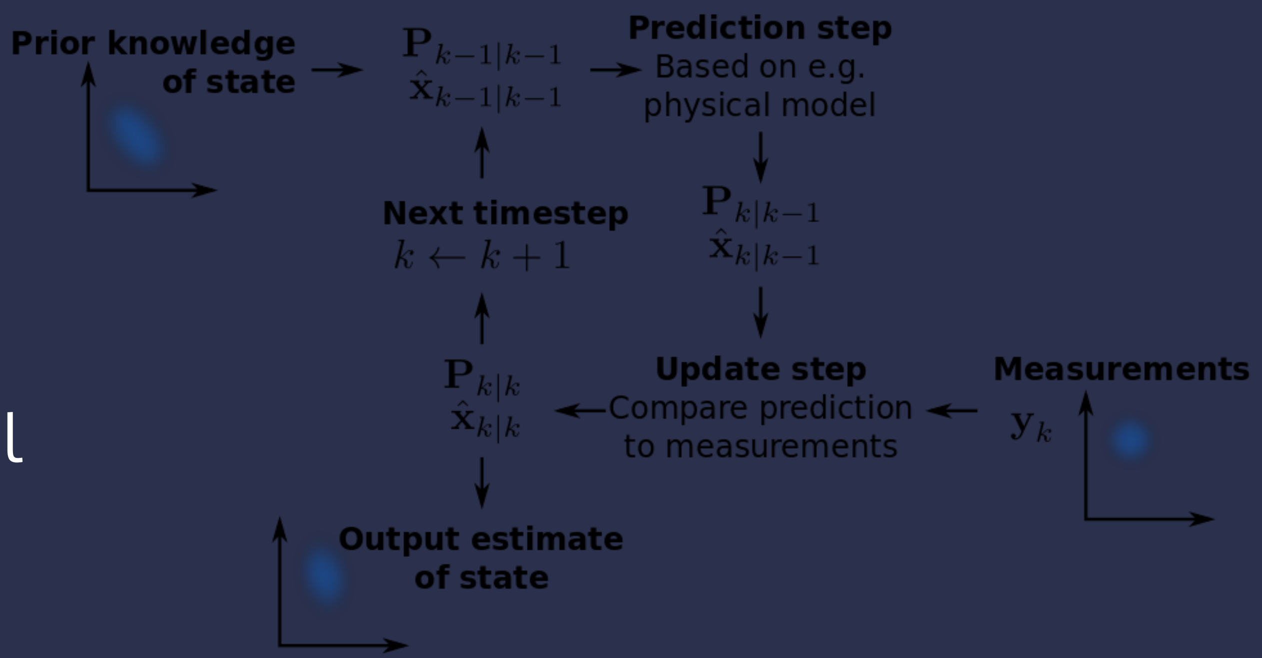
Rappel de probabilités : loi normale

- $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$
  - $\mu = E(x)$ la moyenne
  - $\sigma = \sigma(x)$ l'écart type
- Lois de probabilité représentant bien les phénomènes naturels
- Des propriétés intéressantes :
  - Continue, dérivable
  - Multipliable
- Modélisation du bruit des capteurs, pour la localisation



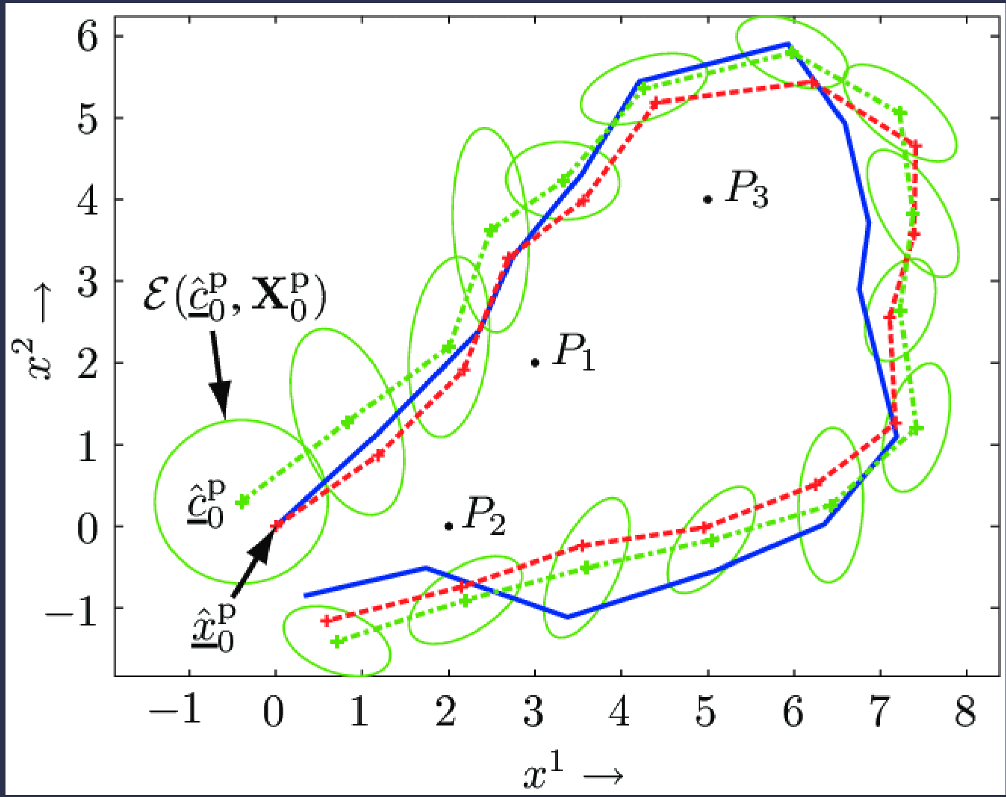
Filtre de Kalman

- L'état du robot est représenté par une loi normale
- Les données des capteurs et les commandes sont représentées par des lois normales
- Performant, rapide mais monomodal et ne fonctionne qu'avec des lois normales



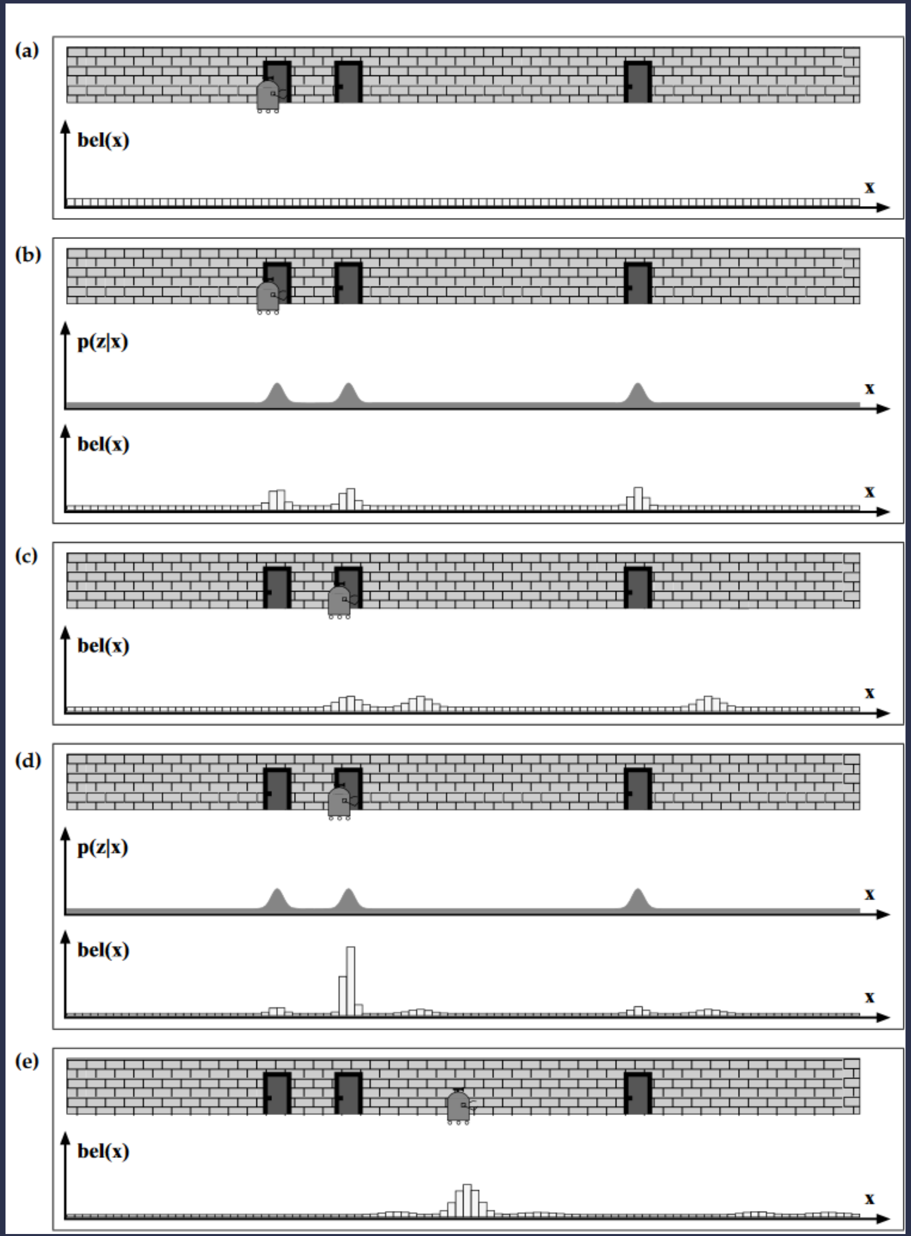
Filtre de Kalman

- Dans le dossier :
  - *catkin_ws/src/ros4pro_robotique_theorique/proba*
- lancez la commande
  - *python3 extended_kalman_filter.py*



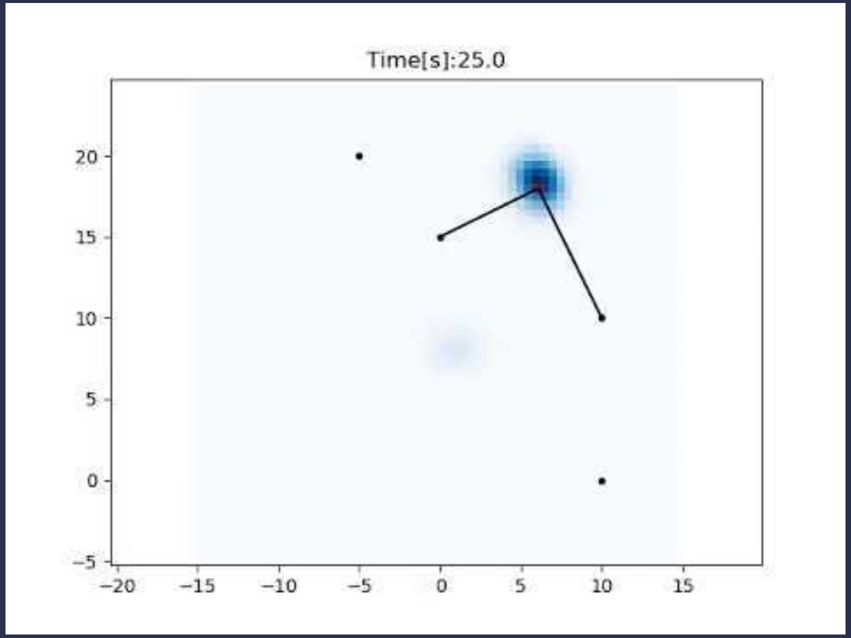
Filtre histogramme

- L’environnement du robot est représenté par une grille
- Chaque case a pour valeur la probabilité que le robot y soit localisé.
- La localisation du robot est la zone de plus haute probabilité
- Méthode flexible, multimodale mais consomme beaucoup de ressources.



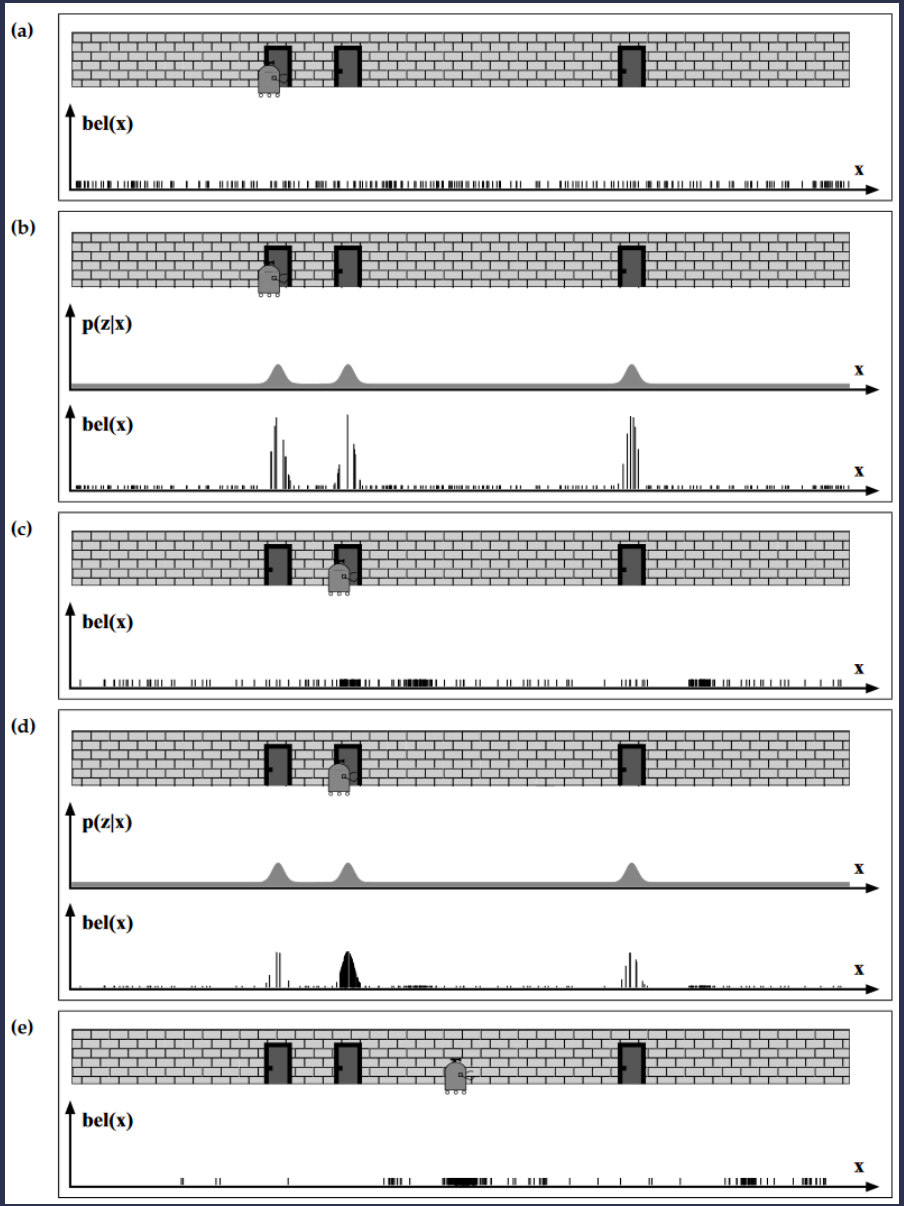
Filtre histogramme

- Dans le dossier :
  - *catkin_ws/src/ros4pro_robotique_theorique/proba*
- lancez la commande
  - *python3 histogram_filter.py*



Filtre Particulaire

- L'environnement du robot est représenté par une grille
- Méthode de
- La localisation du robot est la zone de plus haute probabilité
- Méthode flexible, multimodale, consomme de ressources mais moins que le filtre à histogramme.



Filtre Particulaire

- Dans le dossier :
  - *catkin_ws/src/ros4pro_robotique_theorique/proba*
- lancez la commande
  - *python3 particle_filter.py*