2020

MECA 953 - Robotique

MMT5

UNIVERSITÉ SAVOIE MONT BLANC

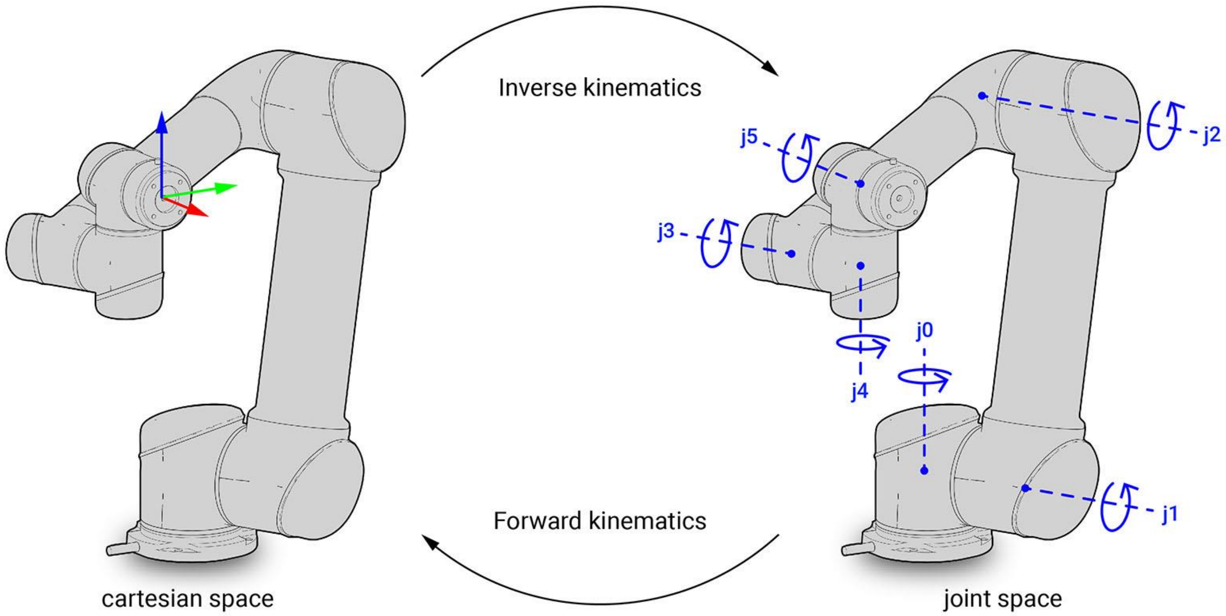
POLYTECH ANNECY-CHAMBÉRY

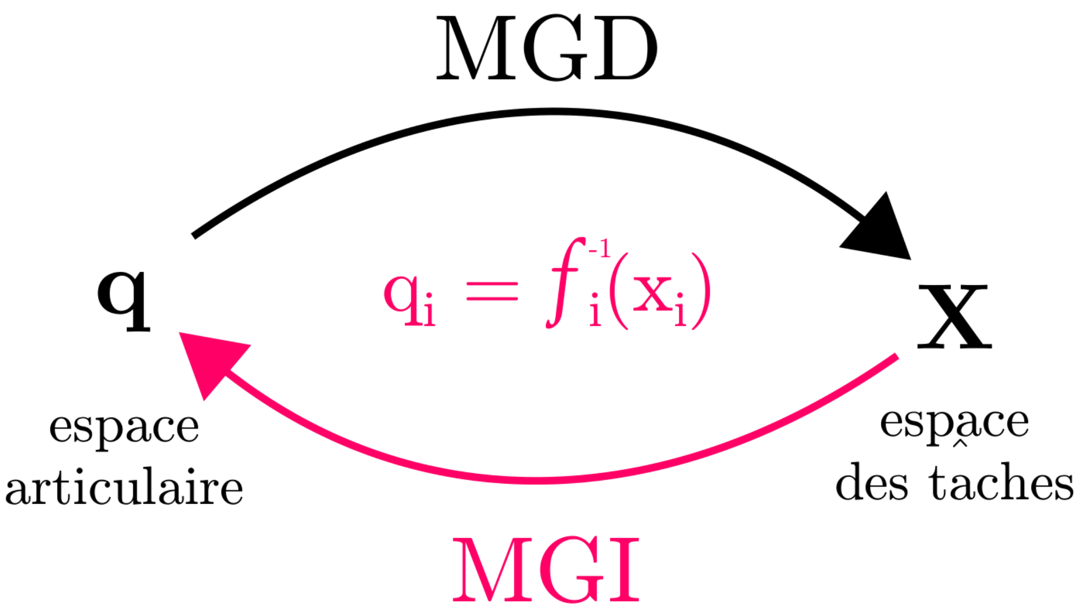
SYMME

5. Modèle Géométrique Inverse (MGI)

Modèle Géométrique Inverse : MGI

Le modèle géométrique inverse est le problème qui permet de connaître toutes les solutions possibles pour les variables articulaires $\boldsymbol{q}$ correspondant à une situation de l'organe terminal $\boldsymbol{X}$ donné

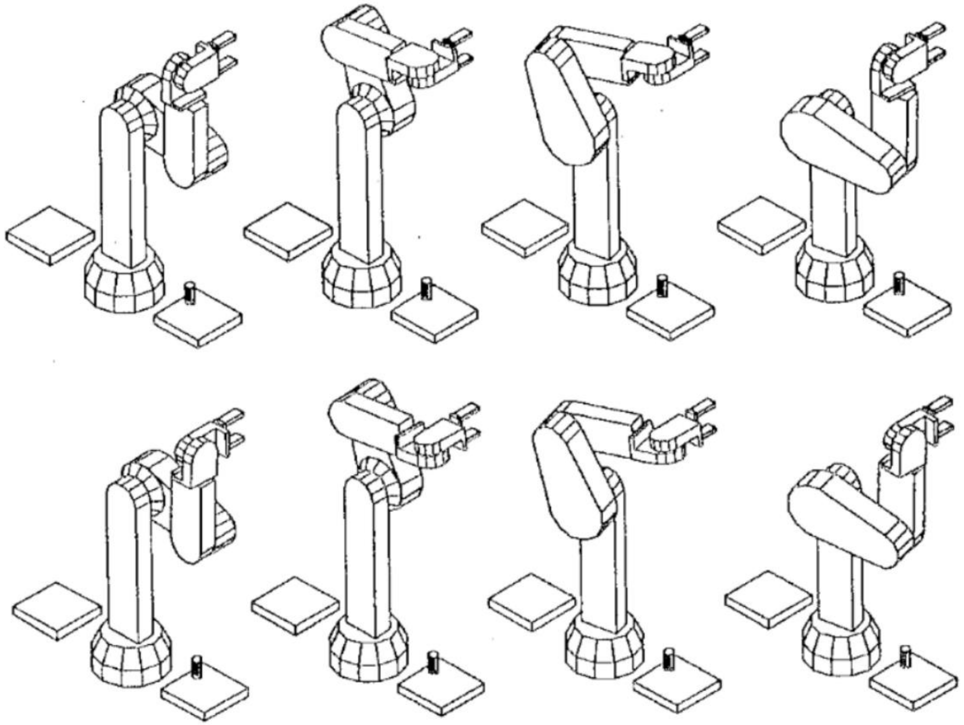




Modèle Géométrique Inverse : MGI

Les équations du problème à résoudre sont, en général, non linéaires (il y'a des *sinus* et *cosinus*). Le problème peut avoir des solutions multiples (exemple Fig.41) voire un nombre infini de solutions (robots redondants cinématiquement) et même n'avoir aucune solution admissible (à cause de la structure cinématique du manipulateur). Le MGD est donc souvent difficile, voire parfois quasi-impossible, de l'inverser directement.

Exemple de solutions du MGI pour un robot 6R.



La position et l'orientation de l'organe terminal est exactement la même pour les 8 possibilités présentées.

Modèle Géométrique Inverse : MGI

Il n'existe pas de méthode systématique pour déterminer le modèle géométrique inverse.
Cependant, 2 méthodes prédominent :

- Géométrique (résolution de triangle)
- Analytique (Inversion de MGD)

Modèle Géométrique Inverse : MGI

Méthode géométrique

Cette méthode n'est intéressante que pour des cas de structures simples.

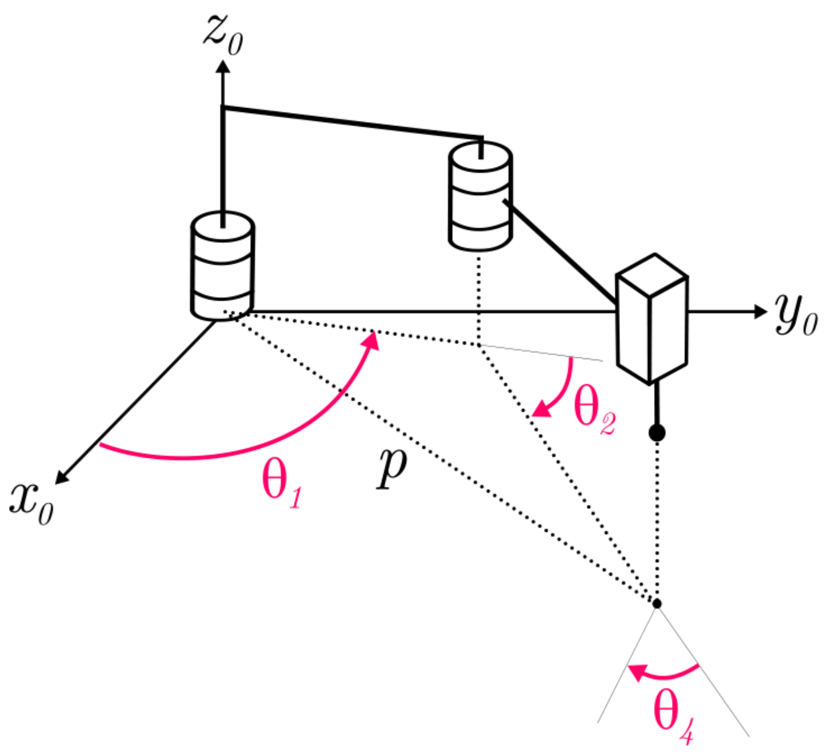
Elle consiste à utiliser les relations trigonométriques d'un triangle quelconque. On essaie alors de ramener le problème à la résolution de triangles, en partant souvent de l'effecteur, pour remonter jusqu'à la base du robot.

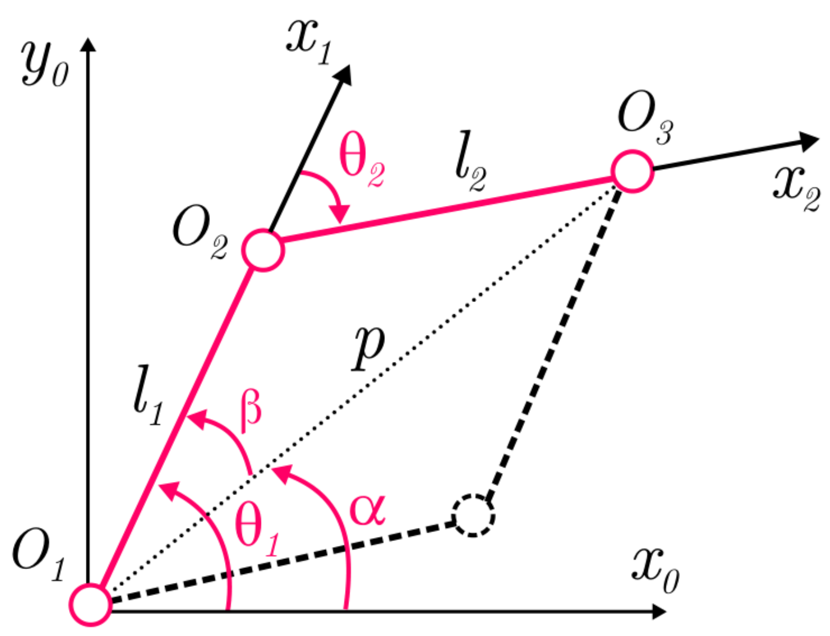
Modèle Géométrique Inverse : MGI

Méthode géométrique

Exemple : MGI du robot Scara

Les coordonnées *opérationnelles* x, y, z sont les coordonnées du point O_4 par rapport au repère de base $\{\mathcal{R}_0\}=(O_0,x_0,y_0,z_0)$





Remarque : Il ya deux configurations possibles qui permettent d'obtenir position de l'organe terminal dans cette position : la configuration "bras haut" ($\theta_2 \in (0,\pi)$) et la configuration "bras bas" ($\theta_2 \in (-\pi,0)$).

Modèle Géométrique Inverse : MGI

Méthode géométrique

Exemple : MGI du robot Scara

- Calcul de θ_1

  Il faut considérer les angles α et β

  $\theta_1 = \alpha + \beta$ en configuration "*bras haut*"

  $\theta_1 = \alpha - \beta$ en configuration "*bras bas*"

  α dépend du signe de p_x, p_y, on calcule donc : $\alpha = \arctan 2(p_x, p_y)$

  $l_2^2 = l_1^2 + p^2 - 2 l_1 p \cos(\beta)$ *(Al Kashî dans le triangle $O_1O_2O_3$)*

  $l_2^2 = l_1^2 + p_x^2 + p_y^2 - 2 l_1 \sqrt{p_x^2 + p_y^2} \cos(\beta)$

  $\beta = \arccos\left(\frac{p_x^2+p_y^2+l_1^2-l_2^2}{2 l_1 \sqrt{p_x^2+p_y^2}}\right)$, avec $\beta \in (0, \pi)$ pour garantir l'existence des triangles

  $$\theta_1 = \varepsilon \arccos\left(\frac{p_x^2+p_y^2+l_1^2-l_2^2}{2 l_1 \sqrt{p_x^2+p_y^2}}\right) + \arctan 2(p_x, p_y)$$

  ($\varepsilon = -1$ ($\theta_2 < 0$) pour la configuration "bras haut",
  $\varepsilon = +1$ ($\theta_2 > 0$) pour la configuration "bras bas")

Modèle Géométrique Inverse : MGI

Méthode géométrique

Exemple : MGI du robot Scara

- Calcul de θ_2

$p^2 = l_1^2 + l_2^2 - 2l_1l_2\cos(\pi - \theta_2)$ *(Al Kashî dans le triangle $O_1O_2O_3$)*

$x^2 + y^2 = l_{11}^2 + l_2^2 + 2l_1l_2\cos(\theta_2)$ (car $\cos(\pi - \theta_2) = -\cos(\theta_2)$)

$$\theta_2 = \varepsilon \arccos\left(\frac{x^2+y^2-l_1^2-l_2^2}{2l_1l_2}\right)$$

sous réserve que $|l_1^2 - l_2^2| < p < l_1^2 + l_2^2$

- Calcul de θ_4

$$\theta_4 = \theta - \theta_1 + \theta_2$$

$$\boldsymbol{\theta_4 = \theta - \varepsilon \arccos\left(\frac{p_x^2+p_y^2+l_1^2-l_2^2}{2l_1\sqrt{p_x^2+p_y^2}}\right) + \arctan 2(p_x,p_y) + \varepsilon \arccos\left(\frac{x^2+y^2-l_1^2-l_2^2}{2l_1l_2}\right))}$$

Modèle Géométrique Inverse : MGI

Méthode géométrique

Exemple : MGI du robot Scara

On définit donc le repère de l'organe terminal dans le repère de travail avec 2 configurations possibles suivant la valeur de ε :

$$\left| \begin{aligned} q_1 &= \theta_1 \\ q_2 &= \theta_2 \\ q_3 &= -z \\ q_4 &= \theta - \theta_1 - \theta_2 \end{aligned} \right.$$

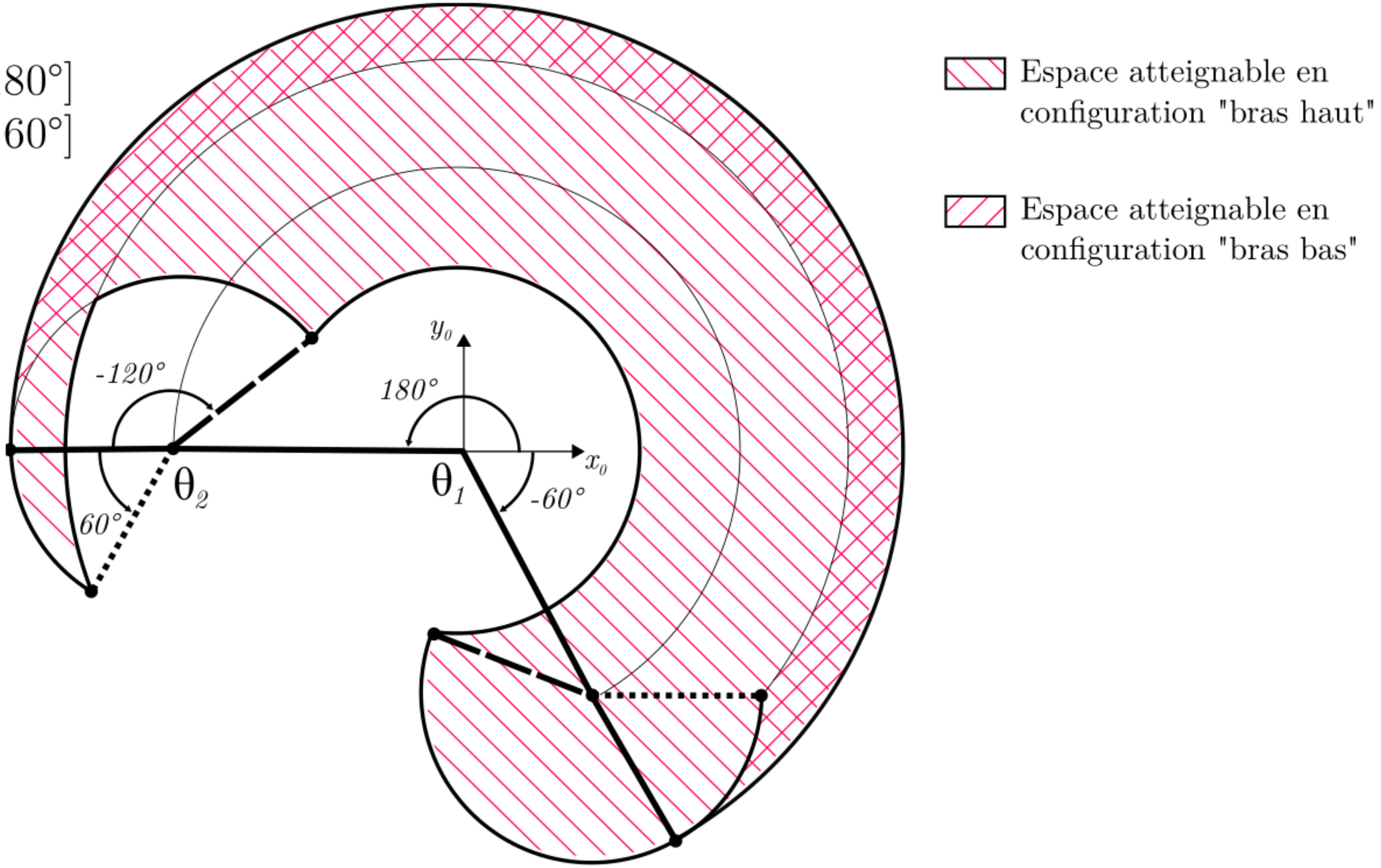
On le voit, le calcul du MGI avec la méthode géométrique s'avère fastidieux même pour un robot possédant une géométrie simple.

Workspace

Robot Scara

En prenant en compte les limites physiques des liaisons (encadrement des variables articulaires $q_{i_{min}} < q_i < q_{i_{max}}$, on définit le volume de travail.

$$\theta_1 \in [-60°; 180°]$$
$$\theta_2 \in [-120°; 60°]$$



Modèle Géométrique Inverse : MGI

Méthode analytique – Méthode de Paul

$\boldsymbol{U_0}$ est la matrice de transformation décrivant l'orientation et la position désirées :

$$\boldsymbol{U_0} = \begin{bmatrix} s_x & n_x & a_x & P_x \\ s_y & n_y & a_y & P_y \\ s_z & n_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Pour un robot à 6 ddl, il s'agit à résoudre le système d'équations suivant dont les q_i sont solutions :

$$\boldsymbol{U_0} = {}^0T_1\,{}^1T_2\,{}^2T_3\,{}^3T_4\,{}^4T_5\,{}^5T_6$$

La méthode consiste à pré-multiplier successivement les deux membres de l'équation précédente par les matrices ${}^i\boldsymbol{T_{i-1}}$ pour i variant de 1 à 6. Cela permet d'isoler et d'identifier l'une après l'autre les variables articulaires q_i recherchées.

Modèle Géométrique Inverse : MGI

Méthode analytique – Méthode de Paul

Le terme de droite est fonction des variables q_2,..., q_6 et son expression analytique à été obtenue lors du calcul du MGD. Le terme de gauche n'est fonction que des éléments de U_0 et de la variable q_1. Une identification terme à terme des deux membres de l'équation permet de réduire le problème à un système d'une ou de deux équations fonction de q_1 uniquement.

Le même procédé est utilisé pour trouver q_2 en pré-multipliant l'expression précédente par $^{\mathbf{2}}\boldsymbol{T}_{\mathbf{1}}$:
Le terme de droite est fonction des variables q_3,..., q_6 dont éléments ont déjà été calculés lors du calcul du MGD. Le terme de gauche n'est fonction que des éléments de U_0 et de la variable q_2.
... et ainsi de suite jusqu'à q_6.

Modèle Géométrique Inverse : MGI

Méthode analytique – Méthode de Paul

Résumé de la Méthode de Paul :

$$U_0 = {}^0T_1{}^1T_2{}^2T_3 \cdots {}^{n-1}T_n$$
$$^1T_0U_0 = {}^1T_2{}^2T_3{}^3T_4{}^4T_5 \cdots {}^{n-1}T_n$$
$$^2T_1U_1 = {}^2T_3{}^3T_4{}^4T_5 \ldots {}^{n-1}T_n$$
$$\vdots$$
$$^{n-1}T_{n-2}U_{n-2} = {}^{n-1}T_n$$
$$\text{avec } U_i = {}^iT_n = {}^iT_{i-1}U_{i-1}$$

Pour un robot à 6 d.d.l on a :

$$U_0 = {}^0T_1{}^1T_2{}^2T_3{}^3T_4{}^4T_5{}^5T_6$$
$$^1T_0U_0 = {}^1T_2{}^2T_3{}^3T_4{}^4T_5{}^5T_6 \rightarrow \text{identification de } q_1$$
$$^2T_1U_1 = {}^2T_3{}^3T_4{}^4T_5{}^5T_6 \rightarrow \text{identification de } q_2$$
$$^3T_2U_2 = {}^3T_4{}^4T_5{}^5T_6 \rightarrow \text{identification de } q_3$$
$$^4T_3U_3 = {}^4T_5{}^5T_6 \rightarrow \text{identification de } q_4$$
$$^5T_4U_4 = {}^5T_6 \rightarrow \text{identification de } q_5$$
$$^5T_6U_6 = I_6 \rightarrow \text{identification de } q_6$$

Modèle Géométrique Inverse : MGI

Méthode analytique – Méthode de Paul

La résolution des équations précédentes nécessite de l'intuition, mais l'utilisation de cette méthode sur un grand nombre de robots industriels a montré que seuls quelques types d'équations fondamentales sont rencontrés. Les solutions de ces équations sont présentées dans le tableau 5, certains types ont une solution évidente, d'autres nécessitent quelques développements.

Type	Equation	Solution
Type 1	$X.r_i = Y$	
Type 2	$X.\sin(\theta_i) + Y.\cos(\theta_i) = Z$	
Type 3	$X1.\sin(\theta_i) + Y1.\cos(\theta_i) = Z1$ $X2.\sin(\theta_i) + Y2.\cos(\theta_i) = Z2$	
Type 4	$X1.r_j.\sin(\theta_i) = Z1$ $X2.r_j.\cos(\theta_i) = Z2$	

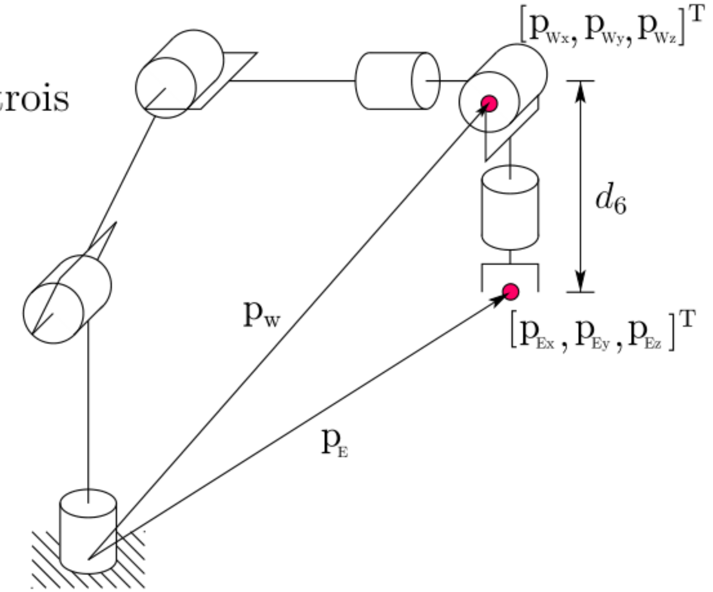
TABLEAU 5 – *Solutions aux types d'équations rencontrés avec la méthode de Paul. (r_i : prismatic joint variable, θ_i : revolute joint variable)*

Modèle Géométrique Inverse : MGI

Méthode analytique – Méthode de Piper (pour les manipulateur à poignet sphérique)

Dans la pratique, les manipulateurs possédant un poignet de type rotule (ou sphérique) sont très répandus. Le centre du poignet est à l'intersection des axes de rotation concourants des trois dernières liaisons pivots du robot. L'origine du repère R6 lié au dernier corps peut être directement positionnée au centre de la rotule. Pour les bras à 6 DDL, il existe donc un découplage entre la position de ce point et l'orientation de l'effecteur :

- Le porteur donne la position du centre poignet. Ce point ne dépend donc que des trois premières variables articulaires q_1, q_2 et q_3.
- Le poignet donne l'orientation de l'organe terminal (pince, outil) qui ne dépend que des trois dernières variables articulaires q_4, q_5 et q_6.



Modèle Géométrique Inverse : MGI

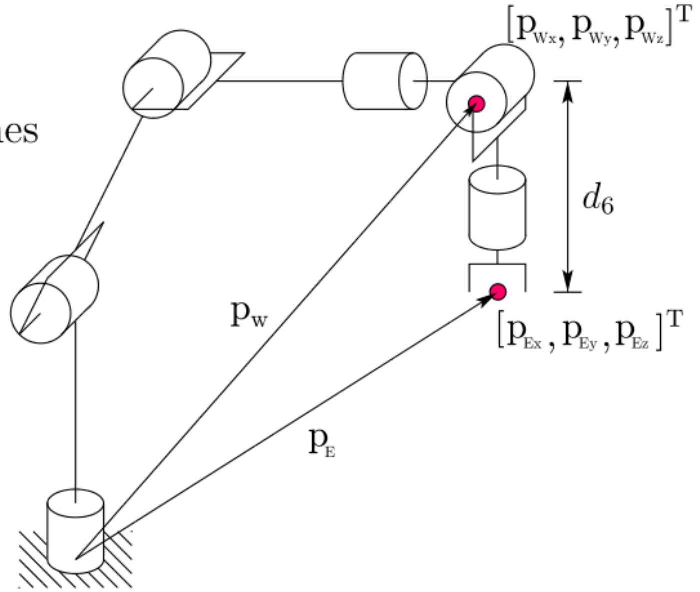
Méthode analytique – Méthode de Piper (pour les manipulateur à poignet sphérique)

$$ {}^0\boldsymbol{T_6} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ; \quad {}^0P_6 = {}^0\boldsymbol{T_6} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} $$

La position du centre du poignet P_W est définie juste par un offset de d_6 par rapport à la position de la pince.

$$ \begin{bmatrix} P_{Wx} \\ P_{Wy} \\ P_{Wz} \end{bmatrix} = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} P_{Ex} - d_6 r_{13} \\ P_{Ey} - d_6 r_{23} \\ P_{Ez} - d_6 r_{33} \end{bmatrix} $$

Par conséquent, sur la base de ce découplage cinématique, le problème se sépare en deux problèmes à 3 équations et 3 inconnues et peut être résolu selon les étapes suivantes :



Modèle Géométrique Inverse : MGI

Méthode analytique – Méthode de Piper (pour les manipulateur à poignet sphérique)

a. Position du poignet

Comme la position du poignet est confondue avec l'origine des repères 4, 5 et 6 on a :

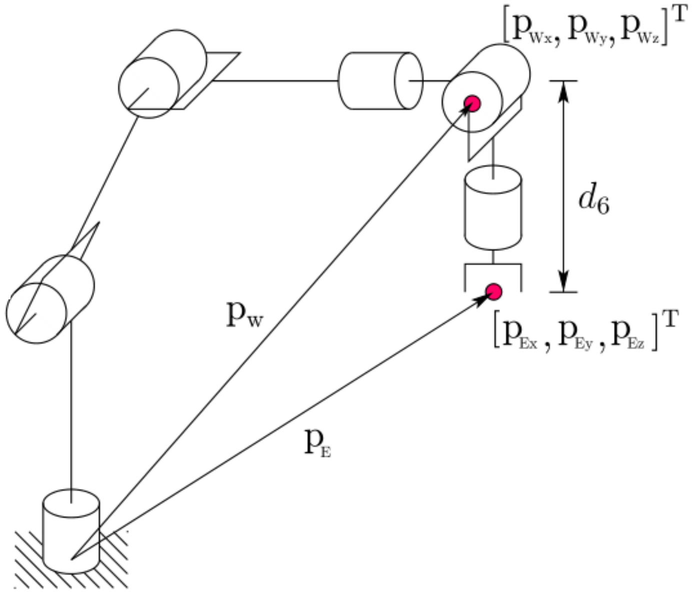
$$^0P_6 = {}^0P_4$$

$$\begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} = {}^0T_1{}^1T_2{}^2T_3{}^3T_4 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

On utilise ensuite la méthode de Paul pour terminer la résolution, en multipliant successivement à gauche les membres de l'équation par $^1T_0$ puis $^2T_1$. L'identification terme à terme des deux matrices permet de déterminer successivement les variables q_1 puis q_2 puis q_3. Les membres de droite étant connus grâce au MGD.

$$^1T_0 \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} = {}^1T_2{}^2T_3{}^3T_4 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$^2T_1{}^1T_0 \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} = {}^2T_3{}^3T_4 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$



Modèle Géométrique Inverse : MGI

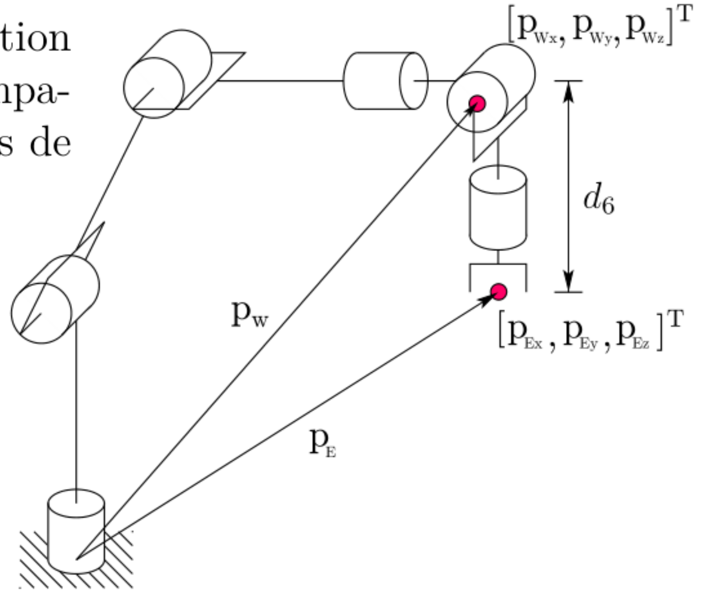
Méthode analytique – Méthode de Piper (pour les manipulateur à poignet sphérique)

b. Orientation du poignet

$$^{0}R_6 = \begin{bmatrix} s_x & n_x & a_x \\ s_y & n_y & a_y \\ s_z & n_z & a_z \end{bmatrix}$$

$$^{3}R_0 \begin{bmatrix} s_x & n_x & a_x \\ s_y & n_y & a_y \\ s_z & n_z & a_z \end{bmatrix} = {}^{3}R_6$$

Les membres de gauche ne dépendent que de q_1, q_2 et q_3. Comme ils sont connus grâce à la position du poignet, il ne reste donc plus qu'à identifier successivement les variables q_4, q_5 et q_6 en comparant terme à terme les matrices après multiplication à gauche par $^{4}R_3$ puis $^{5}R_4$. Les membres de droite étant connus grâce au MGD.



Modèle Géométrique Inverse : MGI

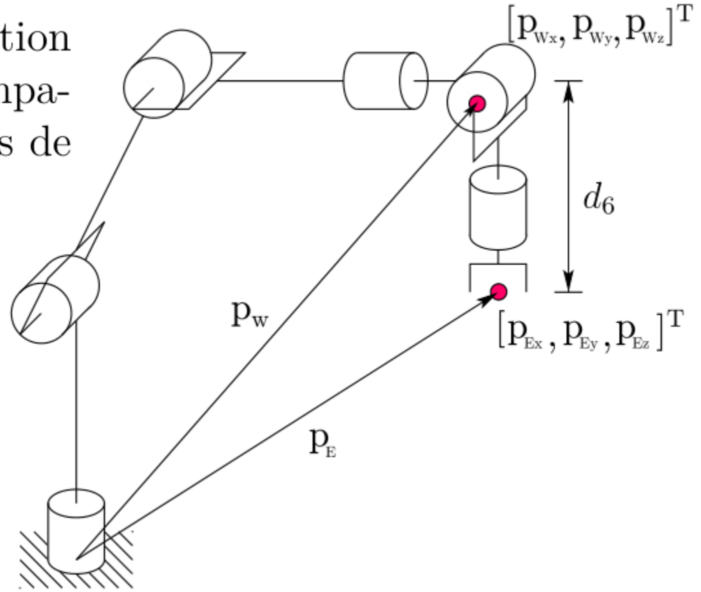
Méthode analytique – Méthode de Piper (pour les manipulateur à poignet sphérique)

b. Orientation du poignet

$$^0R_6 = \begin{bmatrix} s_x & n_x & a_x \\ s_y & n_y & a_y \\ s_z & n_z & a_z \end{bmatrix}$$

$$^3R_0 \begin{bmatrix} s_x & n_x & a_x \\ s_y & n_y & a_y \\ s_z & n_z & a_z \end{bmatrix} = {}^3R_6$$

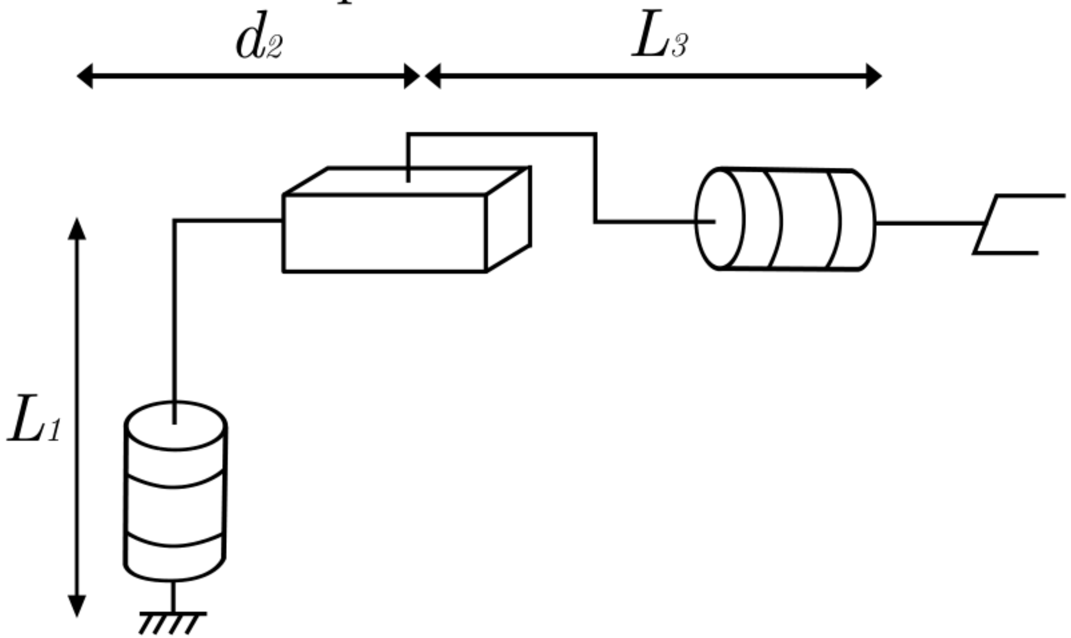
Les membres de gauche ne dépendent que de q_1, q_2 et q_3. Comme ils sont connus grâce à la position du poignet, il ne reste donc plus qu'à identifier successivement les variables q_4, q_5 et q_6 en comparant terme à terme les matrices après multiplication à gauche par $^4R_3$ puis $^5R_4$. Les membres de droite étant connus grâce au MGD.



EXERCICE

MGD MGI Méthode de Paul

Soit le manipulateur RPR suivant :

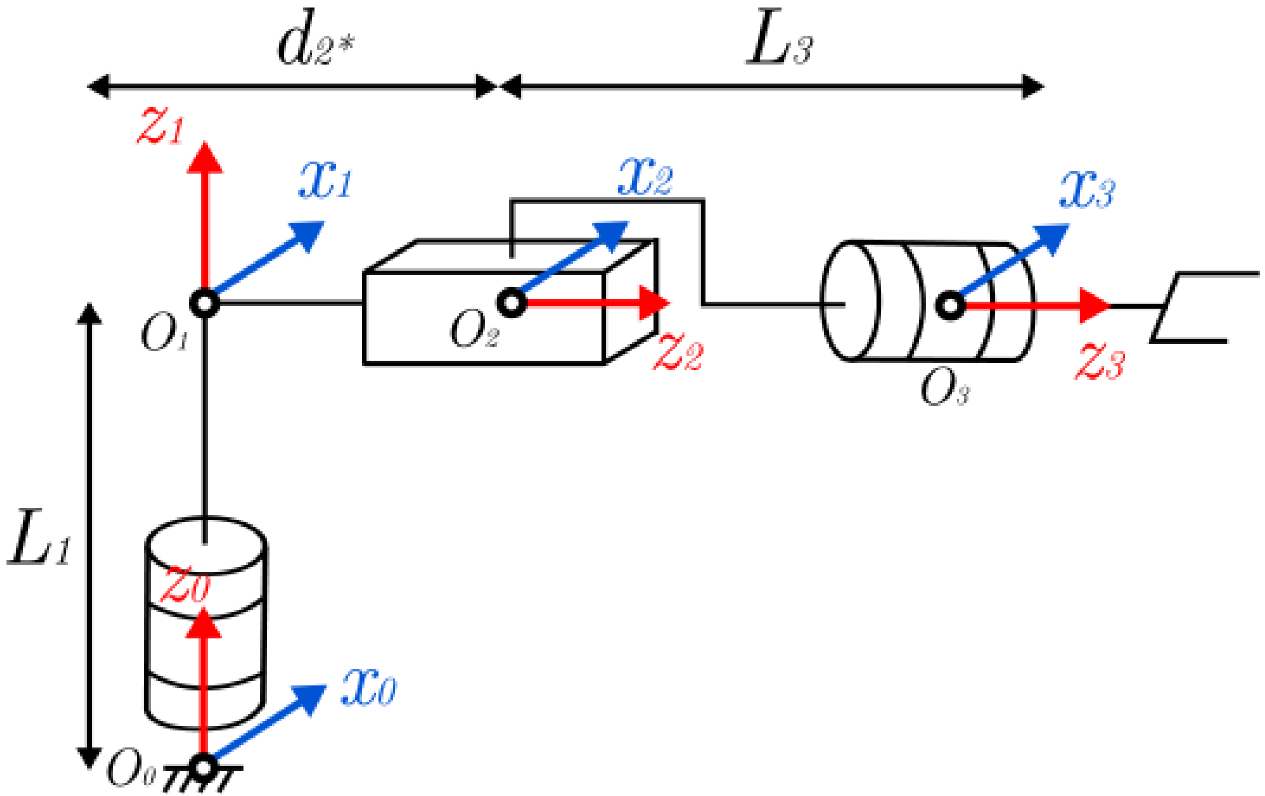


Etablir le MGD du manipulateur.

En utilisant la méthode de Paul, donner le MGI du manipulateur.

EXERCICE

MGD MGI Méthode de Paul



i	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	L_1	θ_1
2	0	$\frac{\pi}{2}$	d_2	0
3	0	0	L_3	θ_3

EXERCICE

MGD MGI Méthode de Paul

$$
{}^{0}\boldsymbol{T}_{1} = \begin{bmatrix} C_1 & -S_1 & 0 & 0 \\ S_1 & C_1 & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^{1}\boldsymbol{T}_{2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -d_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^{2}\boldsymbol{T}_{3} = \begin{bmatrix} C_3 & -S_3 & 0 & 0 \\ S_3 & C_3 & 0 & 0 \\ 0 & 0 & 1 & L_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{0}\boldsymbol{T}_{3} = \begin{bmatrix} C_1C_3 & -C_1S_3 & S_1 & S_1(L_3+d_2) \\ C_3S_1 & -S_1S_3 & -C_1 & -C_1(L_3+d_2) \\ S_3 & C_3 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_x & n_x & a_x & x \\ s_y & n_y & a_y & y \\ s_z & n_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

MGD

$$
{}^{0}\boldsymbol{T}_{3} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = {}^{0}P_3
\qquad
\left|
\begin{aligned}
x &= (L_3 + d_2)\,.sin\,(\theta_1) \\
y &= -\,(L_3 + d_2)\,.cos\,(\theta_1) \\
z &= L_1
\end{aligned}
\right.
$$

Code Python

EXERCICE

MGD MGI Méthode de Paul

$$\boldsymbol{U_0} = \begin{bmatrix} s_x & n_x & a_x & P_x \\ s_y & n_y & a_y & P_y \\ s_z & n_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{U_0} = \boldsymbol{{}^0T_1{}^1T_2{}^2T_3}$$

$$\boldsymbol{{}^1T_0U_0} = \boldsymbol{{}^1T_2{}^2T_3}$$

$$\begin{bmatrix} s_xC_1 + s_yS_1 & n_xC_1 + n_yS_1 & a_xC_1 + a_yS_1 & P_xC_1 + P_yS_1 \\ s_yC_1 - s_xS_1 & n_yC_1 - n_xS_1 & a_yC_1 - a_xS_1 & P_yC_1 - P_xS_1 \\ s_z & n_z & a_z & P_z - L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C_{23} & -S_{23} & 0 & 0 \\ 0 & 0 & -1 & -L_3 - d_2 \\ S_{23} & C_{23} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$P_x.Cos(\theta_1) + P_y.Sin(\theta_1) = 0$ -> equation de TYPE 2 avec X et Y $\neq 0$ et Z $= 0$

$$\boldsymbol{\theta_1 = atan2(-P_x, P_y)}$$

[Code Python](#)

EXERCICE

MGD MGI Méthode de Paul

$$\boldsymbol{{}^2T_1U_1 = {}^2T_3}$$

$$\begin{bmatrix} s_xC_1 + s_yS_1 & n_xC_1 + n_yS_1 & a_xC_1 + a_yS_1 & P_xC_1 + P_yS_1 \\ s_z & n_z & a_z & P_z - L_1 \\ s_xS_1 - s_yC_1 & n_xS_1 - n_yC_1 & a_xS_1 - a_yC_1 & P_xS_1 - P_yC_1 - d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C_3 & -S_3 & 0 & 0 \\ S_3 & C_3 & 0 & 0 \\ 0 & 0 & 1 & L_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P_xS_1 - P_yC_1 - d_2 = L_3 \, -> \, d_2 = P_x.Sin(\theta_1) - P_y.Cos(\theta_1) - L_3$$

$$\boldsymbol{d_2 = P_x.Sin(atan2(-P_x,P_y)) - P_y.Cos(atan2(-P_x,P_y)) - L_3}$$

[Code Python]