

Modèle géométrique

- Modèle géométrique direct
 - Définition
 - Paramétrisation de Denavit hartenberg
 - exercices
- Modèle(s) géométrique(s) inverse(s) MGI
 - Equations de Paul
 - Méthode de Paul
 - Remarques
 - exercices

Modèle géométrique direct

- Définition :
 - Connaître le modèle géométrique direct (MGD) d'un robot correspond à pouvoir exprimer :
 - à partir des coordonnées d'un point/vecteur exprimées dans le repère effecteur du robot
 - Les coordonnées du même point /vecteur, exprimées dans le repère de travail (0 = world frame) du robot
 - En fonction des coordonnées articulaires
 - Le MGD se ramène donc à la matrice de transformation homogène 0T_f :

Resultat MGD robot scara

- Complexité variable, calculs redondants...

- [1] réduction trigo
- [2] expansion trigo

- Notations

- c_i, s_i : $\cos(\theta_i), \sin(\theta_i)$
- c_{ijk} : $\cos(\theta_i + \theta_j + \theta_k)$

$${}^0T_2 = \begin{bmatrix} c_{12} & -s_{12} & 0 & a_2 c_{12} + a_1 c_1 \\ s_{12} & c_{12} & 0 & a_2 s_{12} + a_1 s_1 \\ 0 & 0 & 1 & d_2 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} [1]$$

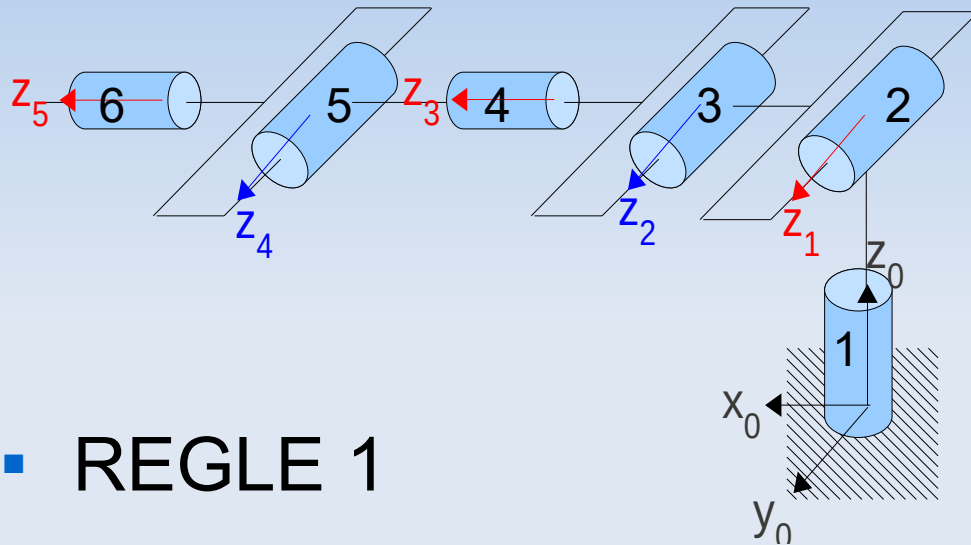
$${}^0T_2 = \begin{bmatrix} c_1 c_2 - s_1 s_2 & -c_1 s_2 - s_1 c_2 & 0 & -a_2 s_1 s_2 + a_2 c_1 c_2 + a_1 c_1 \\ c_1 s_2 + s_1 c_2 & c_1 c_2 - s_1 s_2 & 0 & a_2 c_1 s_2 + a_2 s_1 c_2 + a_1 s_1 \\ 0 & 0 & 1 & d_2 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} [2]$$

Denavit Hartenberg

- Trop de choix possibles :
 - Axes des repères (x,y,z?)
 - Origines des repères (où les placer)
- Nécessité d'un langage commun (normalisation)
- Paramétrisation de Denavit-Hartenberg
 - Règles imposées de choix des axes et origines
 - Unicité des transformations homogènes
 - Un et un seul repère lié à chaque articulation
 - ${}^i T_{i+1} = \text{rot}_{z, \theta_{i+1}} \cdot \text{Trans}_{a_{i+1}, 0, d_{i+1}} \cdot \text{Rot}_{x, \alpha_{i+1}}$

DENAVIT-HARTENBERG

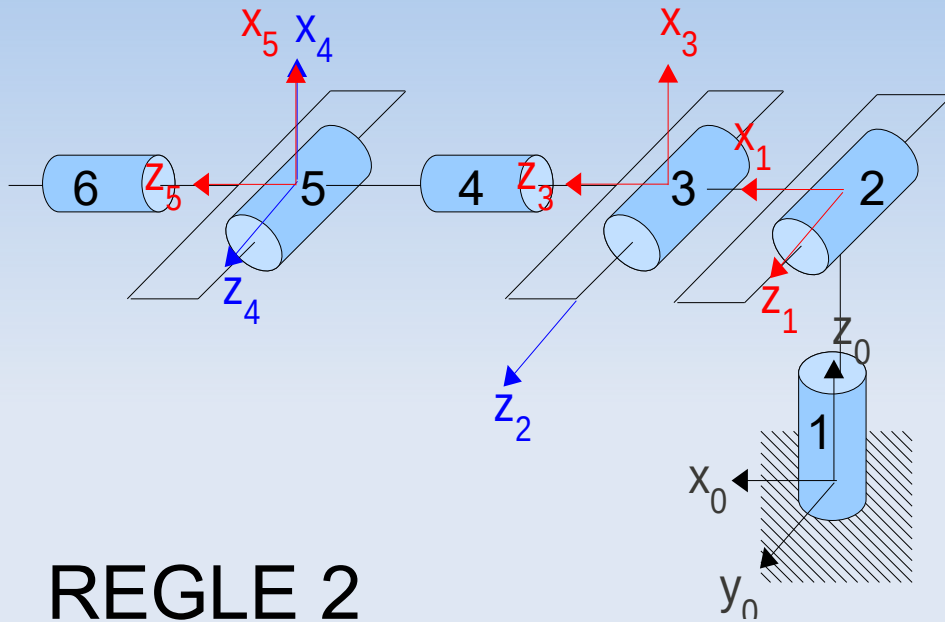
applique sur robot ABB



REGLE 1

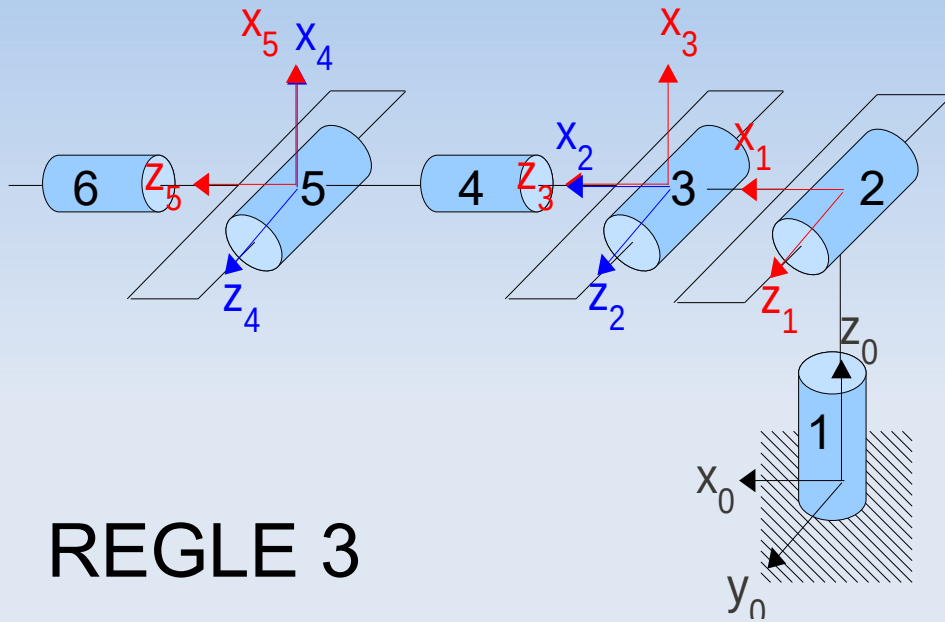
- Axe z_i = axe d'action de l'articulation **$i+1$**
 - Prismatique : axe de translation : distance d_{i+1}
 - Rotoïde axe de rotation : angle θ_{i+1}
- Repère fixe R_0 choisi arbitrairement (sauf z_0)...

DENAVIT-HARTENBERG applique sur robot ABB



- REGLE 2
- Lorsque z_i intersecte z_{i+1} (cas facile)
 - x_{i+1} orthogonal au plan généré par z_i , z_{i+1}
 - O_{i+1} à l'intersection des axes z_i , z_{i+1}

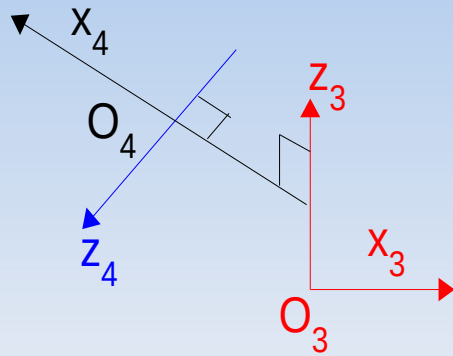
DENAVIT-HARTENBERG applique sur robot ABB



- REGLE 3
- Lorsque z_{i+1} est parallèle à z_i
 - x_{i+1} orthogonal à z_{i+1} et $z_i \Rightarrow$
- Infinite de solutions, levée avec choix arbitraire ci-après
 - x_{i+1} passe par l'origine O_i , ce qui fixe O_{i+1}

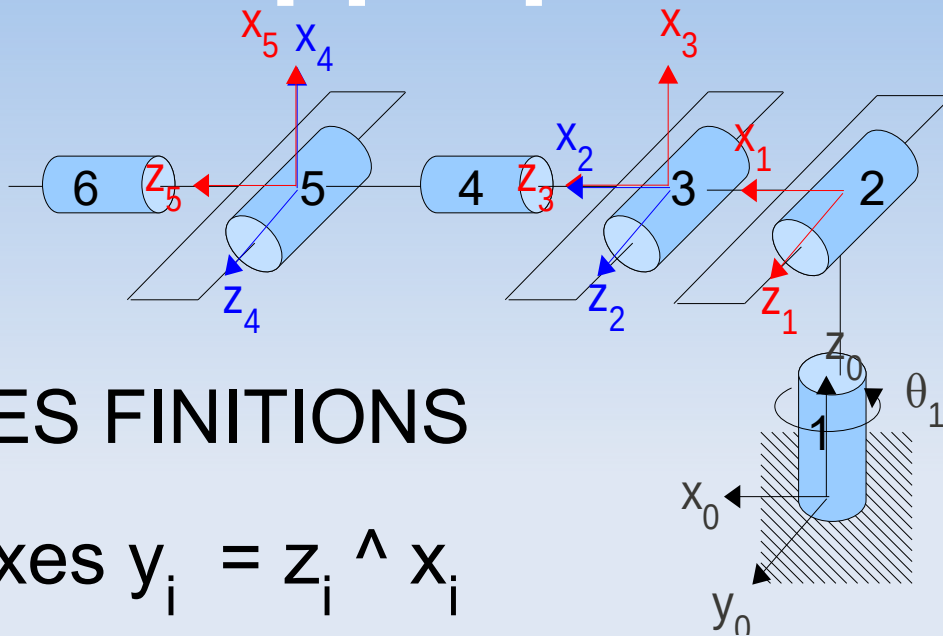
DENAVIT-HARTENBERG

Règle 4



- REGLE 4 (CAS RARE)
- Lorsque z_{i+1} , z_i sont non-coplanaires
 - x_{i+1} orthogonal à z_{i+1} et à z_i
 - x_{i+1} intersecte z_{i+1} et z_i
 - O_{i+1} = intersection entre x_{i+1} et z_{i+1}

DENAVIT-HARTENBERG applique sur robot ABB



- LES FINITIONS

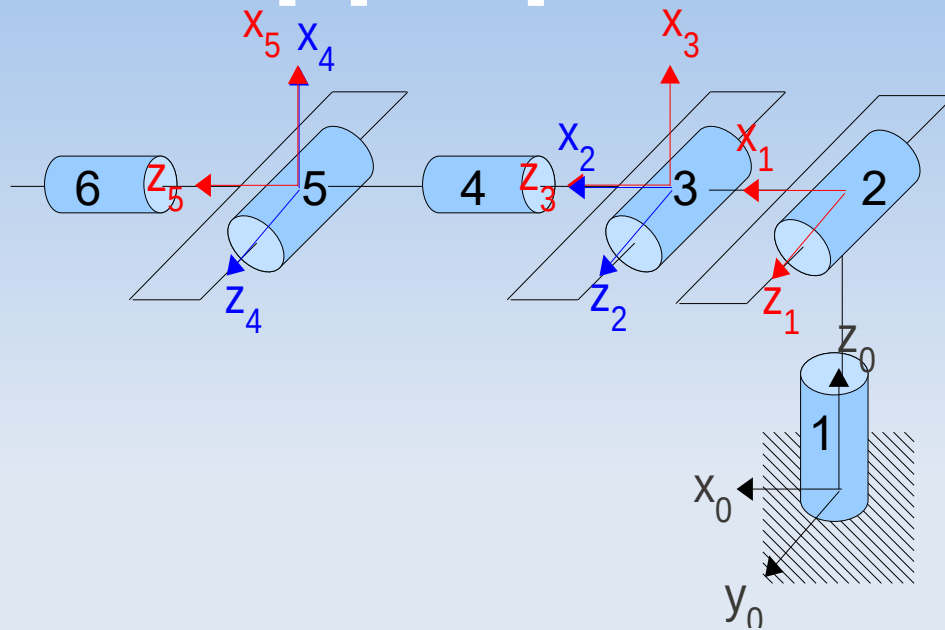
- Axes $y_i = z_i \wedge x_i$

(right hand rule : majeur droit = pouce droit \wedge index droit)

- Variables :

- Angles de rotation autour de z_i : θ_{i+1}
- Translations le long de z_i : d_{i+1}

DENAVIT-HARTENBERG applique sur robot ABB



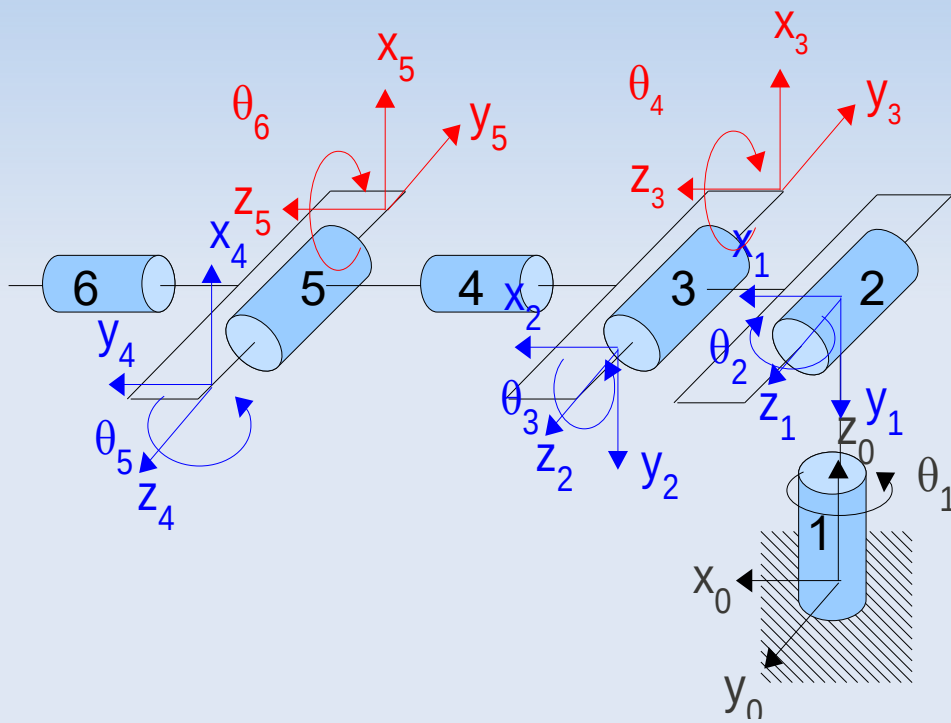
- INTERET :Si on applique ces règles précédentes,
- il existe un et seul quadruplet $(\theta_{i+1}, a_{i+1}, d_{i+1}, \alpha_{i+1})$ tel que

$${}^i T_{i+1} = \text{Rot}_{z, \theta_{i+1}} \cdot \text{Trans}_{a_{i+1}, 0, d_{i+1}} \cdot \text{Rot}_{x, \alpha_{i+1}}$$

- Seuls (θ_{i+1}, d_{i+1}) peuvent être variables

Tableau de denavit-hartenberg

$${}^i T_{i+1} = \text{rot}_{z, \theta_{i+1}} \cdot \text{Trans}_{a_{i+1}, 0, d_{i+1}} \cdot \text{Rot}_{x, \alpha_{i+1}}$$

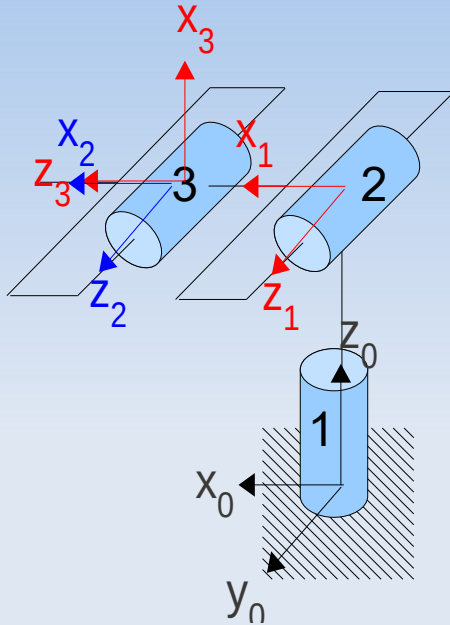


i, i+1	Rot z_i	Trans x_{i+1}	Trans z_i	rot x_{i+1}
0,1	θ_{1^*}		d_1	$\alpha_1 = -90^\circ$
1,2	θ_{2^*}	a_2		
2,3	θ_{3^*}			$\alpha_3 = -90^\circ$
3,4	θ_{4^*}		d_4	$\alpha_4 = 90^\circ$
4,5	θ_{5^*}			$\alpha_5 = -90^\circ$
5,6	θ_{6^*}		d_6	

* : paramètre variable

$${}^0 T_6 = \underbrace{[{}^0 T_1 \cdot {}^1 T_2 \cdot {}^2 T_3]}_{{}^0 T_3: \text{BRAS}} \cdot \underbrace{[{}^3 T_4 \cdot {}^4 T_5 \cdot {}^5 T_6]}_{{}^3 T_6: \text{PINCE ROTULE}}$$

MGD Bras Robot (3 dl)



$${}^i T_{i+1} = \text{rot}_{z, \theta_{i+1}} \cdot \text{Trans}_{a_{i+1}, 0, d_{i+1}} \cdot \text{Rot}_{x, \alpha_{i+1}}$$

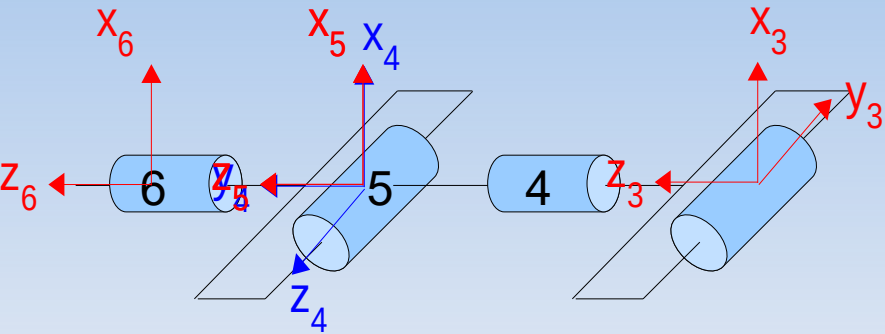
i, i+1	Rot z_i	Trans x_{i+1}	Trans z_i	rot x_{i+1}
0,1	θ_{1^*}		d_1	$\alpha_1 = -90^\circ$
1,2	θ_{2^*}	a_2		
2,3	θ_{3^*}			$\alpha_3 = -90^\circ$

* : paramètre v

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

- INTERET : suffisant pour fixer la position de L'origine de la pince (facile à inverser)

MGD Pince de type Rotule(3 dl)



$${}^i T_{i+1} = \text{rot}_{z, \theta_{i+1}} \cdot \text{Trans}_{a_{i+1}, 0, d_{i+1}} \cdot \text{Rot}_{x, \alpha_{i+1}}$$

i, i+1	Rot z_i	Trans x_{i+1}	Trans z_i	rot x_{i+1}
3,4	θ_{4^*}		d_4	$\alpha_4 = 90^\circ$
4,5	θ_{5^*}			$\alpha_5 = -90^\circ$
5,6	θ_{6^*}		d_6	

* : paramètre v

$${}^3 T_6 = {}^3 T_4 \cdot {}^4 T_5 \cdot {}^5 T_6$$

- INTERET : permet de fixer l'orientation de la pince (facile à inverser, ne dépend pas du robot)

Calcul formel : MGD Bras Robot (invérifiable)

```
dh(theta,a,d,alpha) := rotz(theta) . trans(a,0,d) . rotx( alpha ) $
```

```
l_subst:[cos(t1)=c1,sin(t1)=s1,cos(t2)=c2,sin(t2)=s2,cos(t3)=c3,sin(t3)=s3]$
```

```
T01 : dh(t1 ,0 ,d1, -%pi/2 )$
```

```
T12 : dh(t2 ,a2,0 , 0 )$
```

```
T23 : dh(t3 ,0 ,0 , -%pi/2 )$
```

```
T03 : trigsimp( T01.T12.T23 );
```

```
T03_aff:subst(l_subst,T03);
```

```
print("T03");print(T03_aff);
```

i, i+1	Rot Z_i	Trans X_{i+1}	Trans Z_i	rot X_{i+1}
0,1	θ_{1^*}		d_1	-90°
1,2	θ_{2^*}	a_2		
2,3	θ_{3^*}			-90°

y_n

T03

```
[ c1 c2 c3 - c1 s2 s3 s1 - c1 c2 s3 - c1 c3 s2 a2 c1 c2 ]
[ c2 c3 s1 - s1 s2 s3 - c1 - c2 s1 s3 - c3 s1 s2 a2 c2 s1 ]
[ - c2 s3 - c3 s2 0 s2 s3 - c2 c3 d1 - a2 s2 ]
[ 0 0 0 0 1 ]
```

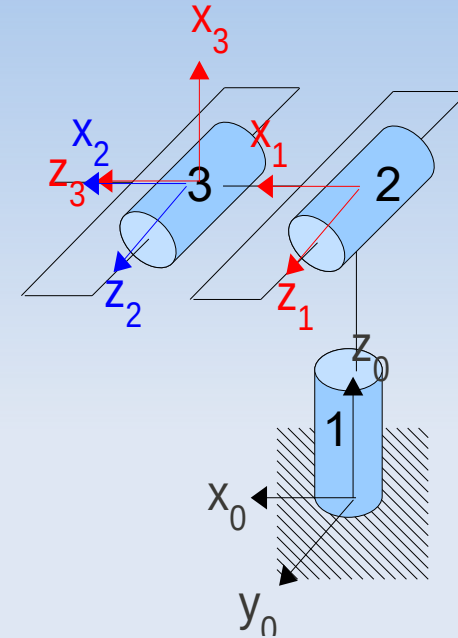
Calcul formel : MGD Bras Robot (vérification indispensable)

```
l_verif_1: [ t1=0 , t2=0 , t3 = -%pi/2 ] $
```

```
print(" T01");print(trigsimp(subst(l_verif_1,T01)));
```

```
print(" T02");print(trigsimp(subst(l_verif_1,T01.T12)));
```

```
print(" T03");print(trigsimp(subst(l_verif_1,T03)))
```



T01

[1	0	0	0]
[]
[0	0	1	0]
[]
[0	- 1	0	d1]
[]
[0	0	0	1]

T02

[1	0	0	a2]
[]
[0	0	1	0]
[]
[0	- 1	0	d1]
[]
[0	0	0	1]

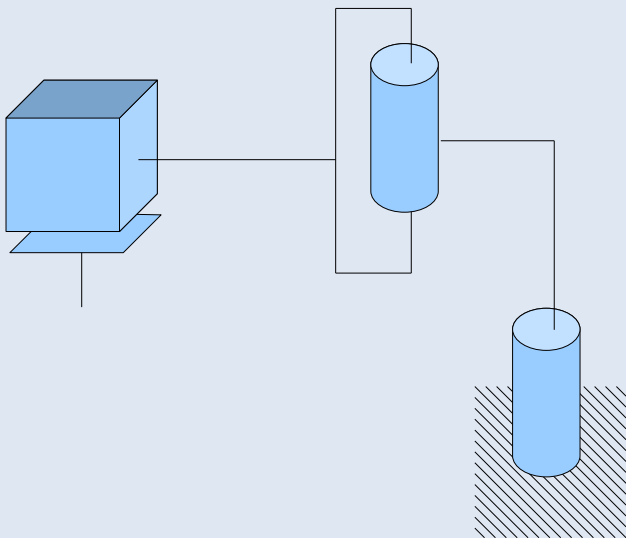
T03

[0	0	1	a2]
[]
[0	- 1	0	0]
[]
[1	0	0	d1]
[]
[0	0	0	1]

Exercice : MGD robot scara

- Compléter tableau Denavit-Hartenberg scara
 - Règle 1: $z_i = \text{axe de l' articulation } i+1$
 - Règle 3: $z_i \text{ et } z_{i+1} \text{ colinéaires} \Rightarrow x_{i+1} \text{ passe par } O_i$
 - Finitions: $y_i = z_i \wedge x_i$

$${}^i T_{i+1} = \text{rot}_{z, \theta_{i+1}} \cdot \text{Trans}_{a_{i+1}, 0, d_{i+1}} \cdot \text{Rot}_{x, \alpha_{i+1}}$$



$i, i+1$	Rot z_i	Trans x_{i+1}	Trans z_i	rot x_{i+1}
0,1				
1,2				
2,3				

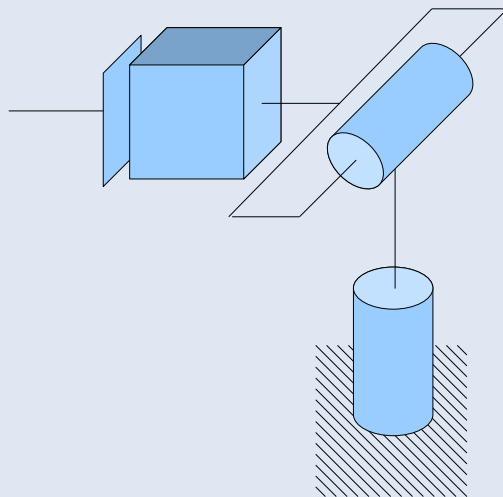
* : paramètre variable

Exercice : MGD manipulateur sphérique

- Compléter tableau Denavit-Hartenberg

- Règle 1: z_i = axe de l'articulation $i+1$
- Règle 2: (z_i et z_{i+1} non colineaires) : O_{i+1} = intersection de z_i et z_{i+1}
- Finitions: $y_i = z_i \wedge x_i$

$${}^i T_{i+1} = \text{rot}_{z, \theta_{i+1}} \cdot \text{Trans}_{a_{i+1}, 0, d_{i+1}} \cdot \text{Rot}_{x, \alpha_{i+1}}$$



$i, i+1$	Rot z_i	Trans x_{i+1}	Trans z_i	rot x_{i+1}
0,1				
1,2				
2,3				

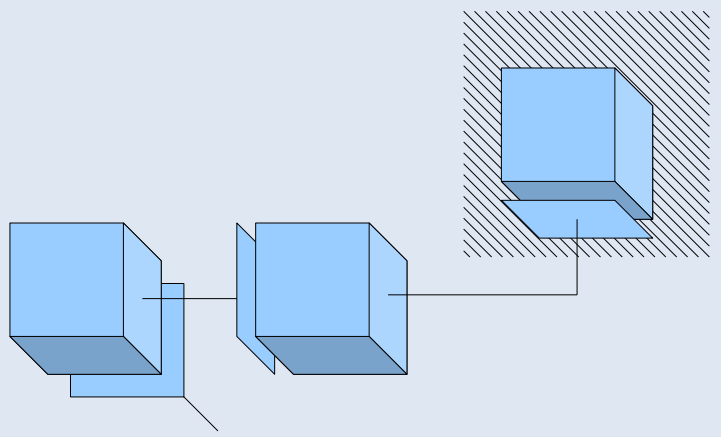
* : paramètre variable

Exercice : MGD manipulateur cartésien

- Compléter tableau Denavit-Hartenberg

- Règle 1: $z_i = \text{axe de l' articulation } i+1$
- Règle 2: (z_i et z_{i+1} non colineaires) : $O_{i+1} = \text{intersection de } z_i \text{ et } z_{i+1}$

- Finitions: $y_i = z_i \wedge x_i$ ${}^i T_{i+1} = \text{rot}_{z, \theta_{i+1}} \cdot \text{Trans}_{a_{i+1}, 0, d_{i+1}} \cdot \text{Rot}_{x, \alpha_{i+1}}$



$i, i+1$	Rot z_i	Trans x_{i+1}	Trans z_i	rot x_{i+1}
0,1				
1,2				
2,3				

* : paramètre variable

Exercice : MGD RPR

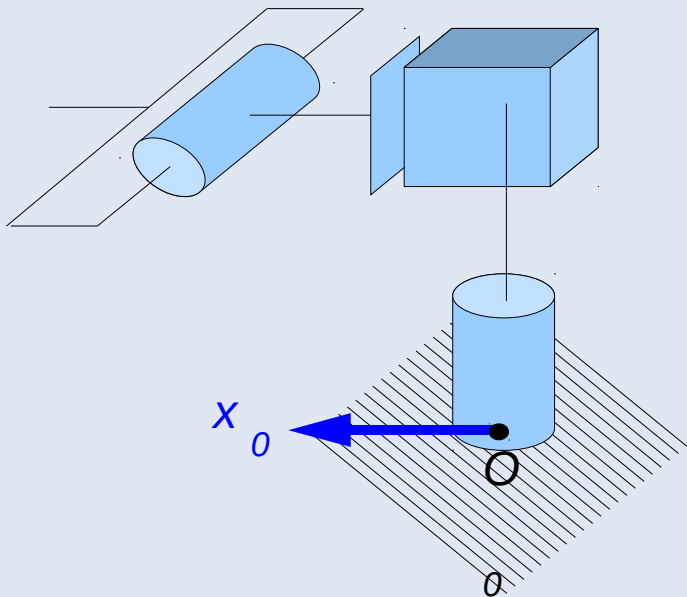
- Compléter tableau Denavit-Hartenberg

- Règle 1: $z_i = \text{axe de l'articulation } i+1$

- Règle 2: (z_i et z_{i+1} non colinéaires) :
 $O_{i+1} = \text{intersection de } z_i \text{ et } z_{i+1}$

- Finitions: $y_i = z_i \wedge x_i$

$${}^i T_{i+1} = \text{rot}_{z, \theta_{i+1}} \cdot \text{Trans}_{a_{i+1}, 0, d_{i+1}} \cdot \text{Rot}_{x, \alpha_{i+1}}$$



$i, i+1$	Rot z_i	Trans x_{i+1}	Trans z_i	rot x_{i+1}
0,1				
1,2				
2,3				

* : paramètre variable

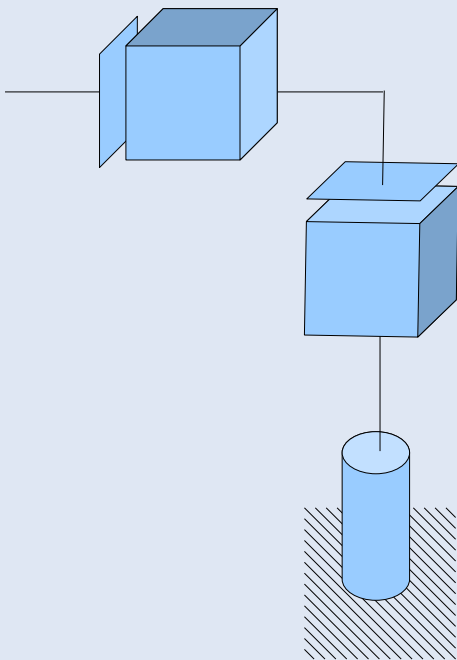
Exercice : MGD RPP

- Compléter tableau Denavit-Hartenberg

- Règle 1: $z_i = \text{axe de l'articulation } i+1$
- Règle 2: (z_i et z_{i+1} non colineaires) :
 $O_{i+1} = \text{intersection de } z_i \text{ et } z_{i+1}$

- Finitions: $y_i = z_i \wedge x_i$

$${}^i T_{i+1} = \text{rot}_{z, \theta_{i+1}} \cdot \text{Trans}_{a_{i+1}, 0, d_{i+1}} \cdot \text{Rot}_{x, \alpha_{i+1}}$$



$i, i+1$	Rot z_i	Trans x_{i+1}	Trans z_i	rot x_{i+1}
0,1				
1,2				
2,3				

* : paramètre variable

Modèle Géométrique Inverse

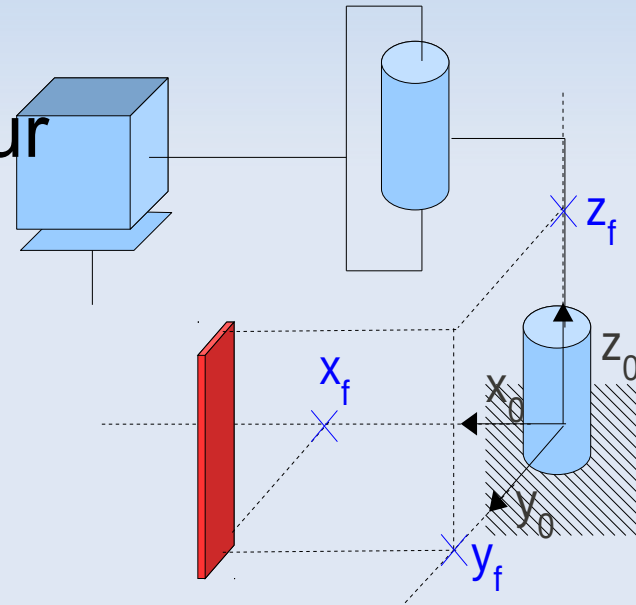
- Le MGD permet
 - de déterminer l'orientation et la position de l'effecteur du robot, en fonction des variables articulaires θ_{i^*} , d_{i^*} (* \Leftrightarrow variables...)
- LE MGI permet
 - de déterminer les variables articulaires θ_{i^*} , d_{i^*} en fonction de l'orientation et la position de l'effecteur du robot, (ou tout autre contrainte géométrique)

Modèle Géométrique Inverse

- Remarques
 - Le MGI n'a rien à voir avec l'inversion de la matrice 0T_f
 - Il s'agit plutôt d'imposer certaines composantes de 0T_f , qui sont des fonctions non-linéaires des θ_{i^*} , d_{i^*}
 - Ce problème n'a pas de solution analytique générale (équations non-linéaires).
- Nombre de solutions (3 cas)
 - Aucune (configuration impossible)
 - Un nombre fini (cas le plus fréquent)
 - Une infinité (redondance => liberté à exploiter)

Exemple : MGI Scara

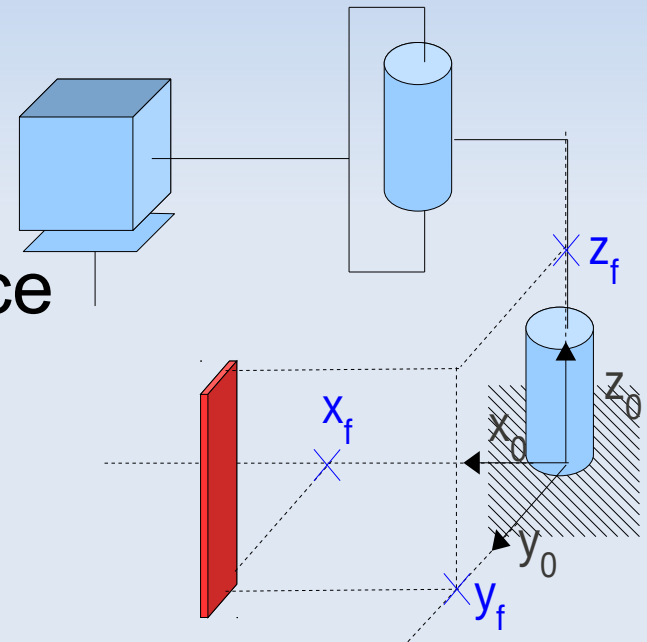
- Scara doit prendre la plaque rouge (par le haut, en pinçant le côté mince de la plaque)
- Contraintes sur la matrice 0T_f
 - PB1** : Coordonnées de l'effecteur (centre de la pince)



$$U_0 = \left[{}^0T_f \right]_{\text{désirée}} = \begin{bmatrix} x & x & x & x_f \\ x & x & x & y_f \\ x & x & x & z_f \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Exemple : MGI Scara

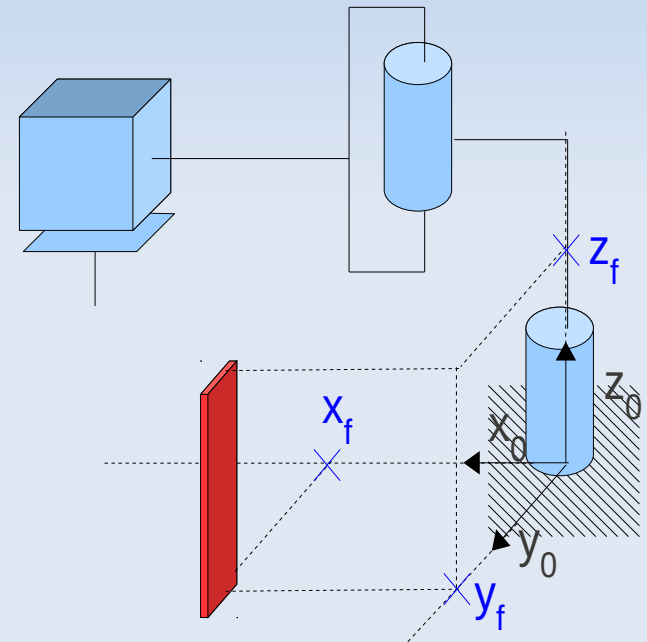
- Scara doit prendre la plaque rouge (par le haut, en pinçant le côté mince de la plaque)
- Repère 6 lié à la pince,
 - axe x_6 parallèle aux mords
 - axe z_6 dirigé dans l'axe de la pince
- PB2** : Orientation de la pince



$$U_0 = \begin{bmatrix} 0 & \pm 1 & 0 \\ \pm 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \begin{matrix} x \\ x \\ x \\ 1 \end{matrix}$$

Exemple : MGI Scara(+ rotule)

- Scara doit prendre la plaque rouge (par le haut, en pinçant le côté mince de la plaque)
- **PB1** = Position de la pince
- + **PB2** = Orientation de la pince



$$U_0 = \left[{}^0T_f \right]_{\text{désirée}} = \begin{vmatrix} 0 & \pm 1 & 0 & x_f \\ \pm 1 & 0 & 0 & y_f \\ 0 & 0 & -1 & z_f \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Calcul direct, trop complexe

- Équations à résoudre :
 - Pour obtenir la posture désirée du robot, il suffit d'écrire l'égalité entre
 - [1] ${}^0T_f(q_i)$, exprimée en fonction des coordonnées articulaires $(q_i = \theta_i^* , d_i^*)$
 - [2] La posture désirée : $U_0 = \left[{}^0T_f \right]_{\text{désirée}}$
 - Et d'en déduire les coordonnées (θ_i^* , d_i^*)
- Ces équations sont souvent trop compliquées pour être résolues directement !..

Exemple : MGI Scara

- Équation à résoudre

$${}^0T_f(q_i) = U_0$$

à gauche de l'équation

$${}^0T_f(q_i) = \begin{bmatrix} {}^0x_f & {}^0y_f & {}^0z_f & {}^0O_f \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

à droite de l'équation

$$U_0 = \begin{bmatrix} 0 & \pm 1 & 0 & x_f \\ \pm 1 & 0 & 0 & y_f \\ 0 & 0 & -1 & z_f \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

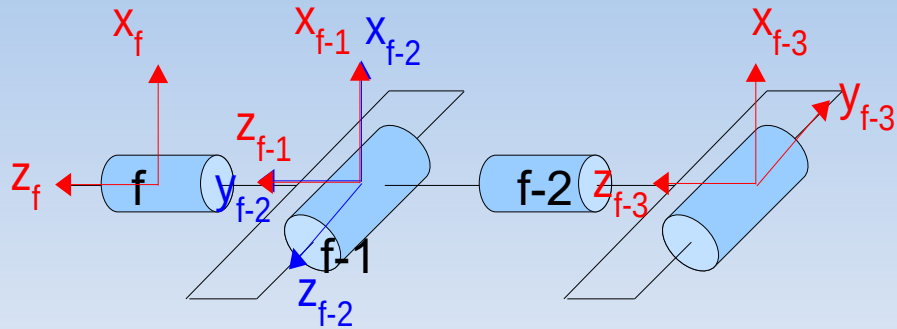
$${}^0x_f = \begin{bmatrix} s1 s2 s4 s6 - c1 c2 s4 s6 + c1 c4 s2 s6 + c2 c4 s1 s6 + c1 c5 c6 s2 s4 + c2 c5 c6 s1 s4 - c4 c5 c6 s1 s2 + c1 c2 c4 c5 c6 \\ -c1 s2 s4 s6 - c2 s1 s4 s6 + c4 s1 s2 s6 - c1 c2 c4 s6 + c5 c6 s1 s2 s4 - c1 c2 c5 c6 s4 + c1 c4 c5 c6 s2 + c2 c4 c5 c6 s1 \\ c6 s5 \end{bmatrix}$$

$${}^0y_f = \begin{bmatrix} -c1 c5 s2 s4 s6 - c2 c5 s1 s4 s6 + c4 c5 s1 s2 s6 - c1 c2 c4 c5 s6 + c6 s1 s2 s4 - c1 c2 c6 s4 + c1 c4 c6 s2 + c2 c4 c6 s1 \\ -c5 s1 s2 s4 s6 + c1 c2 c5 s4 s6 - c1 c4 c5 s2 s6 - c2 c4 c5 s1 s6 - c1 c6 s2 s4 - c2 c6 s1 s4 + c4 c6 s1 s2 - c1 c2 c4 c6 \\ -s5 s6 \end{bmatrix}$$

$${}^0z_f = \begin{bmatrix} c1 s2 s4 s5 + c2 s1 s4 s5 - c4 s1 s2 s5 + c1 c2 c4 s5 \\ s1 s2 s4 s5 - c1 c2 s4 s5 + c1 c4 s2 s5 + c2 c4 s1 s5 \\ -c5 \end{bmatrix} {}^0O_f = \begin{bmatrix} c1 d6 s2 s4 s5 + c2 d6 s1 s4 s5 - c4 d6 s1 s2 s5 + c1 c2 c4 d6 s5 - a2 s1 s2 + a2 c1 c2 + a1 c1 \\ d6 s1 s2 s4 s5 - c1 c2 d6 s4 s5 + c1 c4 d6 s2 s5 + c2 c4 d6 s1 s5 + a2 c1 s2 + a2 c2 s1 + a1 s1 \\ -c5 d6 + d3 + d1 \end{bmatrix}$$

Cas d'un poignet Rotule

équations de position [1]



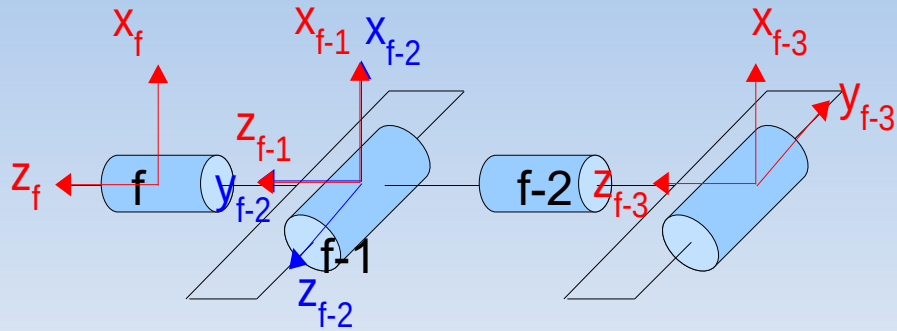
i, i+1	Rot z_i	Trans x_{i+1}	Trans z_i	rot x_{i+1}
f-3,f-2	θ_{f-2}^*		d_{f-2}	$\alpha_{f-2}=90^\circ$
f-2,f-1	θ_{f-1}^*			$\alpha_{f-1}=-90^\circ$
f-1,f	θ_{f^*}		d_f	

- [1] : les coordonnées de l'origine du repère f-1,
- exprimées dans le repère 0, sont indépendantes de q_{f-2} , q_{f-1} , q_f

$${}^0O_{f-1} = {}^0T_{f-3} \cdot {}^{f-3}O_{f-1} = \underbrace{{}^0T_{f-3}}_{fct(q_1, \dots, q_{f-3})} \cdot \underbrace{\begin{bmatrix} 0 \\ 0 \\ d_{f-2} \\ 1 \end{bmatrix}}_{{}^{f-3}O_{f-1}}$$

Cas d'un poignet Rotule

équations de position [2]



i, i+1	Rot z_i	Trans x_{i+1}	Trans z_i	rot x_{i+1}
f-3,f-2	θ_{f-2^*}		d_{f-2}	$\alpha_{f-2}=90^\circ$
f-2,f-1	θ_{f-1^*}			$\alpha_{f-1}=-90^\circ$
f-1,f	θ_{f^*}		d_f	

- [2] : les coordonnées désirées de l'origine du repère f-1, exprimées dans le repère 0, peuvent être calculées depuis la posture désirée

$${}^0O_{f-1}]_{desiree} = [{}^0T_f]_{desiree} \cdot {}^fO_{f-1} = \underbrace{U_0}_{connue} \cdot \underbrace{\begin{bmatrix} 0 \\ 0 \\ -d_f \\ 1 \end{bmatrix}}_{{}^fO_{f-1}}$$

Cas d'un poignet Rotule

équations de position [3]

- L'équation [3] : [1] = [2] , entre
 - [1] les coordonnées ${}^0O_{f-1}$ calculées depuis le MGD
 - [2] les coordonnées $[{}^0O_{f-1}]_{\text{désirées}}$, calculées depuis U_0
- Permet de faire apparaître les équations de position de la pince
 - Indépendantes des variables articulaires de la pince

$$\underbrace{{}^0O_{f-1} = \underbrace{{}^0T_{f-3}}_{\text{fct}(q_1, \dots, q_{f-3})} \cdot \underbrace{\begin{bmatrix} 0 \\ 0 \\ d_{f-1} \\ 1 \end{bmatrix}}_{{}^{f-3}O_{f-1}}}_{[1]} = \underbrace{[{}^0O_{f-1}]_{\text{desiree}} = \underbrace{U_0}_{\text{connue}} \cdot \underbrace{\begin{bmatrix} 0 \\ 0 \\ -d_f \\ 1 \end{bmatrix}}_{{}^fO_{f-1}}}_{[2]}$$

Robot Scara (f=6), équations de position

- [1] : orig. rep. 5 depuis coord. articulaires = 4^{eme} colonne de 0T_5

$${}^0O_5 = \begin{bmatrix} -a_2 s_1 s_2 + a_2 c_1 c_2 + a_1 c_1 \\ a_2 c_1 s_2 + a_2 c_2 s_1 + a_1 s_1 \\ d_3 + d_1 \end{bmatrix} = fct(\theta_1^*, \theta_2^*, d_3^*)$$

- [2] : Orig. du repère 5 depuis la posture désirée (connue)

$$\begin{bmatrix} {}^0O_{5\text{desiree}} \\ -1 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & \pm 1 & 0 & x_f \\ \pm 1 & 0 & 0 & y_f \\ 0 & 0 & -1 & z_f \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{U_0} \cdot \begin{bmatrix} \overbrace{{}^6O_5} \\ 0 \\ 0 \\ -d_f \\ 1 \end{bmatrix} = \begin{bmatrix} x_f \\ y_f \\ z_f + d_6 \\ 1 \end{bmatrix}$$

Méthode de PAUL

$$\underbrace{{}^0T_f(q_1, q_f)}_{\text{COMPLIQUE : } f(q_1, \dots, q_f)} = \underbrace{U_0}_{\text{SIMPLE}} \quad \text{Repère 0}$$

- But : équations plus simples à gauche
 - les exprimer dans un autre repère que 0
 - elles seront plus compliquées à droite
 - les q_i n'apparaîtront que d'un côté à la fois:

$${}^1T_f(q_2, q_f) = {}^1T_0 \cdot U_0 = U_1(q_1) \quad \text{Repère 1}$$

$${}^2T_f(q_3, q_f) = {}^2T_0 \cdot U_0 = U_2(q_1, q_2) \quad \text{Repère 2}$$

$${}^iT_f(q_{i+1}, q_f) = {}^iT_0 \cdot U_0 = U_i(q_1, q_i) \quad \text{Repère } i$$

Equations de PAUL

Equations élémentaires pour la méthode de Paul

	Équation (s)	inconnue(s)
Type 1		
Type 2		
Type 3	$\int [1] X_1 \cdot \sin(A_i) + Y_1 \cdot \cos(A_i) = z$	
Type 4	$\int [1] X_1 \cdot R_j \cdot \sin(A_i) = z$	
Type 5	$\int [1] X_1 \cdot \sin(A_i) = Y_1 + Z_1 \cdot R$	
Type 6	$\int [1] W \cdot \sin(A_k) = X \cdot \cos(A_i) + Y \cdot \sin(A_i) + z$	
Type 7 modifié	$\int W2 \cdot \sin(A_k) + W1 \cdot \cos(A_k) = X \cdot \cos(A_i) + Y \cdot \sin(A_i) + z$	
Type 8 modifié	$\int X \cdot \cos(A_i) + Y \cdot \cos(A_k) = z$	

Résolution des équations : en annexe + exos

Méthode de Paul sur SCARA (position \Leftrightarrow origine rep 5)

- Equs. Repère 0 : [4^{eme} colonne de 0T_5] = [0O_5 désirée]

$${}^0O_5 = \begin{bmatrix} -a_2 s_1 s_2 + a_2 c_1 c_2 + a_1 c_1 \\ a_2 c_1 s_2 + a_2 c_2 s_1 + a_1 s_1 \\ d_3 + d_1 \end{bmatrix} = \begin{bmatrix} x_f \\ y_f \\ z_f + d_6 \end{bmatrix} \quad \boxed{d_3 = z_f + d_6 - d_1}$$

- Equs. Repère 1 : [4^{eme} colonne de 1T_5] = [1O_5 désirée = ${}^1T_0 \cdot {}^0O_{5d}$]
 - à gauche : 1T_5 ne dépend plus de c_1, s_1
 - à droite : 1T_0 ne dépend que de c_1, s_1
 - Les équations lignes 1 et 2 seront linéaires en s_1, c_1, s_2, c_2

Méthode de Paul sur SCARA (position \Leftrightarrow origine rep 5)

- Equs. Repère 1 : [4^{eme} colonne de 1T_5] = [${}^1T_0 \cdot {}^0O_5$ désirée]

$${}^1O_5 = \begin{bmatrix} a2 c2 \\ a2 s2 \\ d3 \end{bmatrix} = \begin{bmatrix} s1 yf + c1 xf - a1 \\ c1 yf - s1 xf \\ zf + d6 - d1 \end{bmatrix}$$

- Les lignes 2 et 1 forment une equ. de Paul de Type 6

[on aura donc aucune ou 2 solutions en (θ_1, θ_2)]

$$\begin{bmatrix} \text{[Ligne 2]} & [a2] \cdot s2 = [yf] \cdot c1 + [-xf] \cdot s1 + [0] \\ \text{[Ligne 1]} & [a2] \cdot c2 = [yf] \cdot s1 - [-xf] \cdot c1 + [-a1] \end{bmatrix}$$

$$\text{TYPE 6 : } \begin{cases} [1] & W \cdot \sin(A_k) = X \cdot \cos(A_i) + Y \cdot \sin(A_i) + Z1 \\ [2] & W \cdot \cos(A_k) = X \cdot \sin(A_i) - Y \cdot \cos(A_i) + Z2 \end{cases}$$

$$\text{VARIABLES : } \quad \theta_2 = A_k, \theta_1 = A_i$$

$$\text{CONSTANTES : } \quad W = a2, X = yf, Y = -xf, Z1 = 0, Z2 = -a1$$

Exercice 1

MGI scara

- Soit U_0 la posture désirée du robot scara (+rotule)
 - Résumer sur une feuille de papier la suite des calculs à effectuer pour que le robot scara atteigne cette posture (employer maxima pour calcul formel)
 - Saisir les instructions correspondantes avec le logiciel scilab (ou matlab, ou octave, ou autre)
 - Vérifier le bon fonctionnement lorsque $d1=a1=a2=1m$, pour des postures du type :

$$U_0 = \begin{pmatrix} 0 & \pm 1 & 0 & x_f = \text{a vous de choisir quelques valeurs} \\ \pm 1 & 0 & 0 & y_f = \text{a vous de choisir quelques valeurs} \\ 0 & 0 & -1 & z_f = \text{a vous de choisir quelques valeurs} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

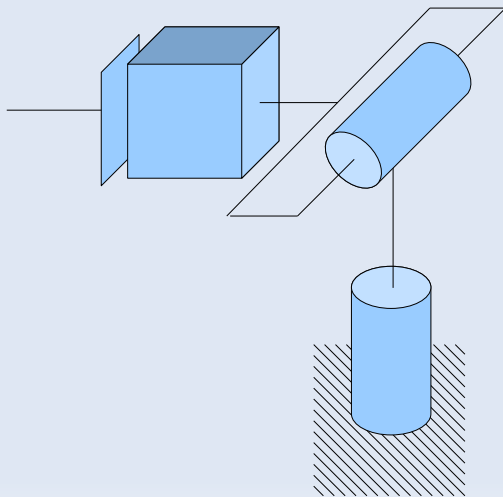
Exercice 2

Comprendre les postures

- On veut que l'effecteur (pince rotule) d'un robot
 - ait son axe z_f à 45° , dans le plan x_0, y_0
 - ait son axe x_f (les mors) vertical, dirigé vers le haut
 - ait une origine de coordonnées algébriques
 - -1 m, suivant l'axe x_0
 - 2 m, suivant l'axe y_0
 - 3 m suivant l'axe z_0
- Ecrire la posture U_0 correspondante

Exercice 3 : MGI manipulateur sphérique (position effecteur)

- Ecrire le MGI d'un manipulateur sphérique 3 axes, permettant de fixer la position de l'effecteur (ne pas écrire les équations d'orientation)
 - Déterminer le type, les variables et les paramètres des équations à résoudre (équations de Paul \Leftrightarrow MGI)
 - Changer éventuellement de repère lorsque les équations sont trop compliquées



$i, i+1$	Rot z_i	Trans x_{i+1}	Trans z_i	rot x_{i+1}
0,1	θ_{1^*}		d_1	$\alpha_1 = +90^\circ$
1,2	θ_{2^*}			$\alpha_2 = -90^\circ$
2,3			d_{3^*}	