Modélisation :

Un manipulateur est considéré comme étant une chaine de liaisons connectées entre elles par des articulations rationnelles ou prismatiques, une extrémité de cette chaine est attachée à la base, tandis que l'autre extrémité est libre et reliée à l'outil terminal, en vue de manipuler les objets ou effectuer des taches d'assemblages, le mouvement relatif de chaque articulation entraine celui des liaisons, ce qui permet de positionner l'extrémité libre et de donner à l'outil terminal l'orientation désirée. En effet, dans la plupart des applications en robotique, on s'intéresse à la trajectoire et la position finale de l'outil terminal par rapport à la référence fixe, la cinématique d'un robot veut l'étude analytique de cette trajectoire sans voir les forces et moments qui causent le mouvement, dans ce qui suit, nous considérons un manipulateur dont les paramètres géométriques sont supposés connus et nous répondrons aux deux questions suivantes :

- Pour un manipulateur dont les paramètres géométriques sont supposés connus, pour chaque vecteur des variable d'articulation [q1 q2 q3…….qn], n étant le nombre de degrés de liberté du manipulateur, quelle est la position et l'orientation de l'outil terminal correspondante par rapport au référentiel fixe (la base).

- Soit une position et orientations désirées de l'outil terminal, le manipulateur peut-il atteindre celle-ci si oui, quelles sont les différentes configurations du manipulateur, c'est à- dire l'ensemble des vecteurs q, qui peuvent nous le garantir, la première de ces questions est appelée problème cinématique direct qui intervient directement dans la modélisation dynamique des robots, et la seconde est appelée problème cinématique inverse.

Modélisation cinématique directe :

La cinématique directe consiste à donner une pose à un squelette d'animation en modifiant uniquement les paramètres de ses articulations (variables d'articulation).
Pour placer la main d'un personnage dans une position voulue, on doit modifier les articulations de l'épaule, du coude puis du poignet afin de les placer dans des angles permettant d'obtenir la pose voulue.

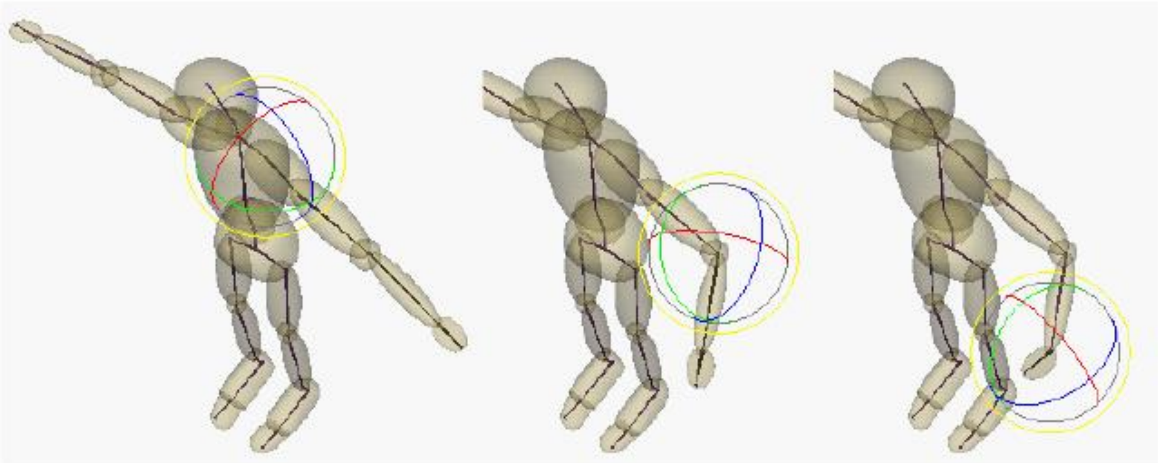
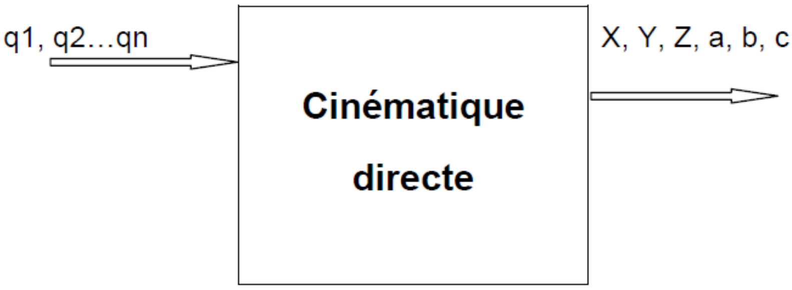


Figure 1:Pose d'un squelette.

La cinématique directe permet de faire des mouvements d'arcs précis et bien définis., puisqu'on contrôle directement les rotations des articulations. On positionne une articulation en modifiant la position de ses propres variables d'articulation et de ses parents.



Méthode de Dénavit-Hartenberg :

Cette méthode permet de caractériser la position relative de deux solides avec seulement 4 paramètres. Tout d'abord il faut bien mettre en place les repères dans les solides du robot.

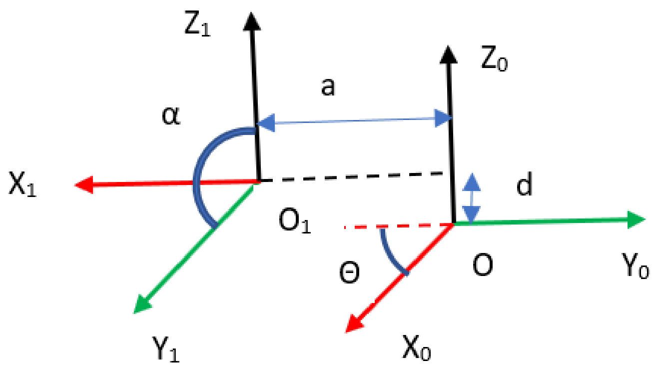
Selon 4 cas possibles :

Soit $R_0(O, X_0, Y_0, Z_0)$ et $R_1(O_1, X_1, Y_1, Z_1)$ deux repères orthonormés direct

- Si Z_0 et Z_1 sont croisés l'axe X_1 doit être perpendiculaire a (Z_0) et (Z_1).
- Même choses au cas précèdent seule l'origine de R_1 est sur Z_0.
- Si Z0 et Z1 sont parallèles on trace le perpendiculaire sur Z1 et Z0, puis on trace son perpendiculaire c'est X_1.
- Les deux repères sont confondus, les axes aussi sont confondus.

Les 4 paramètres déterminantes :

- (Joint angle) thêta1 c'est l'angle de X_0 vers X_1 mesurée par rapport à Z_0.
- (Link offset) d1 distance de O_0 vers O_1 mesurée suivant Z_0.
- (Link tength) a1 distance de Z_0 vers Z_1 mesurée suivant X_1.
- (Link twist) alpha1 c'est l'angle de Z_0 vers Z_1 mesurée par rapport à X_1.



Après la détermination des paramètres on les place dans un tableau comme suit

Link	ai	Alpha i	d i	Thêta i
1				
…				

Après on fait les matrices de passage Ai d'un repère R_i vers R_{i-1} :

Cos (thêta i)	-Cos (alpha i) Sin (thêta i)	Sin (alpha i) Cos (thêta i)	ai Cos (thêta i)
Sin (thêta i)	Cos (alpha i) Cos (theta i)	-Sin (alpha i) Cos (theta i)	ai Sin (thêta i)
0	Sin (alpha i)	Cos (alpha i)	di
0	0	0	1

La matrice de transformation globale :

$$T_n^0 = A1 * ****** An$$

La mise en place du repère :

On utilise les 4 cas déjà cité, en commençant de la base vers la tête et en choisissant l'axe de rotation est le Z pour toutes les articulations on arrive au repérage suivant.

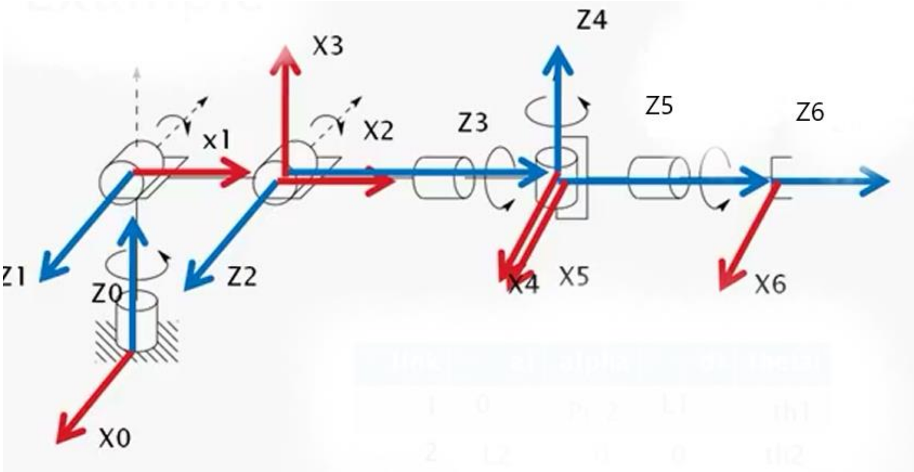


Figure 2:Mise en place des repères.

Le tableau des paramètres :

Link	ai	Alpha i	d i	Thêta i
1	0	Pi/2	L1	Th1
2	L2	0	0	Th2
3	0	Pi/2	0	Th3
4	0	Pi/2	L3	Th4
5	0	-Pi/2	0	Th5
6	0	0	L6	Th6

Simulation :

Matlab (Symbolique) :

```
%%Standar DH parametr matrix
syms th d alpha a

 A=[cos(th)  -cos(alpha)*sin(th)  sin(th)*sin(alpha)   a*cos(th)
    sin(th)  cos(alpha)*cos(th)  -cos(th)*sin(alpha)   a*sin(th)
    0              sin(alpha)      cos(alpha)              d
    0                   0              0                    1    ]

% A=trotz(th)*transl(0,0,d)*transl(a,0,0)*trotx(alpha)

 %%Link1
  syms th1 L1
  A1=subs(A,{a,alpha,d,th},{0,pi/2,L1,th1})

  %%Link2
  syms th2 L2
  A2=subs(A,{a,alpha,d,th},{L2,0,0,th2})

  %%Link3
  syms th3 L3
  A3=subs(A,{a,alpha,d,th},{0,pi/2,0,th3})

   %%Link4
  syms th4 L4
  A4=subs(A,{a,alpha,d,th},{0,pi/2,L3,th4})

   %%Link5
  syms th5 L5
  A5=subs(A,{a,alpha,d,th},{0,-pi/2,0,th5})

  %%Link6
 syms th6 L6
 A6=subs(A,{a,alpha,d,th},{0,0,L6,th6});

 %%Total Forward matrix

 At=simplify(A1*A2*A3*A4*A5*A6);
 At2=subs(At,{L1,L2,L3,L6},{10,9,9,6});

%get X,Y,Z at zero angles
    q1=0;  q2=0;  q3=0;  q4=0;  q5=0;  q6=0;
  At3=subs(At2,{th1,th2,th3,th4,th5,th6},{q1,q2,q3,q4,q5,q6})    ;
```

Figure 3:Calcul de la matrice de passage sur Matlab.

On a obtenu comme coordonnées de l'outil terminale :

X= 9*cos(th1)*cos(th2) - 6*sin(th5)*(sin(th1)*sin(th4) - cos(th4)*(cos(th1)*sin(th2)*sin(th3) - cos(th1)*cos(th2)*cos(th3))) + 6*cos(th5)*(cos(th1)*cos(th2)*sin(th3) + cos(th1)*cos(th3)*sin(th2)) + 9*cos(th1)*cos(th2)*sin(th3) + 9*cos(th1)*cos(th3)*sin(th2).

Y= 9*sin(th2 + th3)*sin(th1) + 9*cos(th2)*sin(th1) + 6*cos(th1)*sin(th4)*sin(th5) + 6*sin(th2 + th3)*cos(th5)*sin(th1) - 6*cos(th2)*cos(th3)*cos(th4)*sin(th1)*sin(th5) + 6*cos(th4)*sin(th1)*sin(th2)*sin(th3)*sin(th5).

Z=   9*sin(th2) - 9*cos(th2 + th3) + 3*sin(th4 - th5)*sin(th2 + th3) - 3*sin(th2 + th3)*sin(th4 + th5) - 6*cos(th2 + th3)*cos(th5) + 10.

Le matrice de passage globale pour des angles q1=0 ; q2=0 ; q3=0 ; q4=0 ; q5=0 ; q6=0 ; est :

```
>> At3

At3 =

[ 1,  0,  0,  9]
[ 0, -1,  0,  0]
[ 0,  0, -1, -5]
[ 0,  0,  0,  1]
```

MATLAB (Peter Corne robotics Toolbox):

Pour utiliser le package de Peter corne dans Matlab il faut le télécharger et l'importer dans ce package il y a une classe s'appeler Link très utiliser dans la robotique pour faire les simulations en se basant sur le tableau du DH paramètres en créant des liens (Links) par la commande :

L(i) = Link ([Th d a alpha]) ;

Pour faire un assemblage des liens (Links) on utilise la commande suivante :

Robot=SerialLink(L) ;

```
>> startup_rvc
Robotics, Vision & Control: (c) Peter Corke 1992-2011 http://www.petercorke.com
- Robotics Toolbox for Matlab (release 9.10)
 - pHRIWARE (release 1.1): pHRIWARE is Copyrighted by Bryan Moutrie (2013-2019) (c)
Run rtbdemo to explore the toolbox
>> L1=10;L2=9;L3=9;L6=6;
>> L(1)=Link([0 L1 0  pi/2]);
L(2)=Link([0 0  L2  0]);
L(3)=Link([0 0  0  pi/2]);
L(4)=Link([0 L3 0  pi/2]);
L(5)=Link([0 0  0  -pi/2]);
L(6)=Link([0 L6 0  0]);
>> Robot=SerialLink(L);
Robot.name='6DoF';
>> Robot
```

Pour afficher le tableau des paramètres on tape Robot dans la fenêtre de commande puis on click entrer.

```
Robot =

6DoF (6 axis, RRRRRR, stdDH, fastRNE)

+---+-----------+-----------+-----------+-----------+-----------+
| j |     theta |         d |         a |     alpha |    offset |
+---+-----------+-----------+-----------+-----------+-----------+
|  1|         q1|         10|          0|      1.571|          0|
|  2|         q2|          0|          9|          0|          0|
|  3|         q3|          0|          0|      1.571|          0|
|  4|         q4|          9|          0|      1.571|          0|
|  5|         q5|          0|          0|     -1.571|          0|
|  6|         q6|          6|          0|          0|          0|
+---+-----------+-----------+-----------+-----------+-----------+

grav =    0  base = 1  0  0  0   tool =  1  0  0  0
          0         0  1  0  0           0  1  0  0
       9.81         0  0  1  0           0  0  1  0
                    0  0  0  1           0  0  0  1
```

Le matrcie de transformation globale pour q1=0 ; q2=0 ; q3=0 ; q4=0 ; q5=0 ; q6=0 ; est :

```
>> Robot.fkine([0 0 0 0 0 0])

ans =

    1.0000         0         0    9.0000
         0   -1.0000   -0.0000   -0.0000
         0    0.0000   -1.0000   -5.0000
         0         0         0    1.0000
```

Comme vous voyez qu'avec des commande simple on a obtenue les mémes résultats de la methode symbolique(Symbolic methode), tous est maintenant en fonctions des angles de rotatios theta autour de l'axe de révolution de chaque liaison, on peut maintenant faire un plot de notre robot en utilisant la commande :

Robot.plot([q1 q2 q3 q4 q5 q6])

Les qi sont les theta.

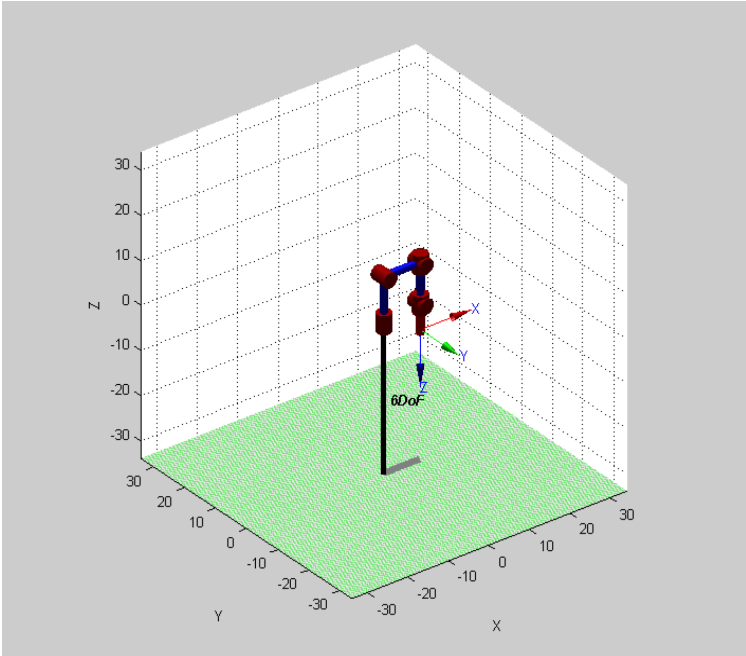


Figure 4:schéma 3d du Robot.

On peut faire une animation en utilisant des boucles for.

```
>> for th1=0:0.1:pi
Robot.plot([th1 0 0 0 0 0]);
pause(0.25)
end
>>
```

Figure 5:code pour faire une animation.

Modélisation cinématique inverse :

La cinématique inverse s’occupe de générer la configuration des joints parents requise pour obtenir le positionnement désiré.

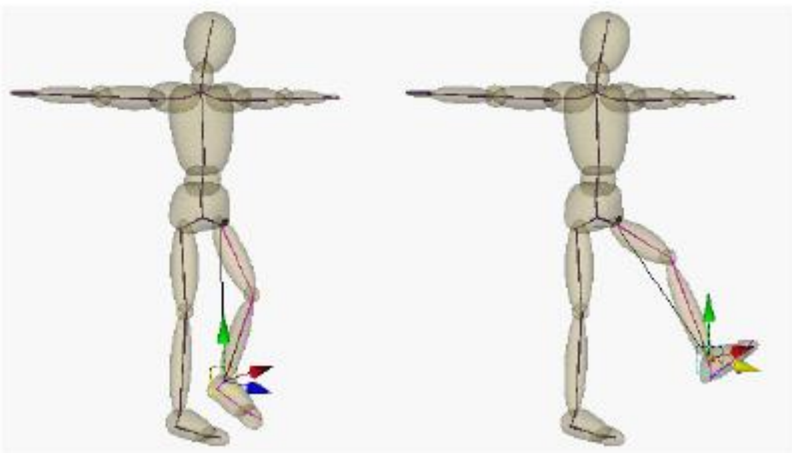
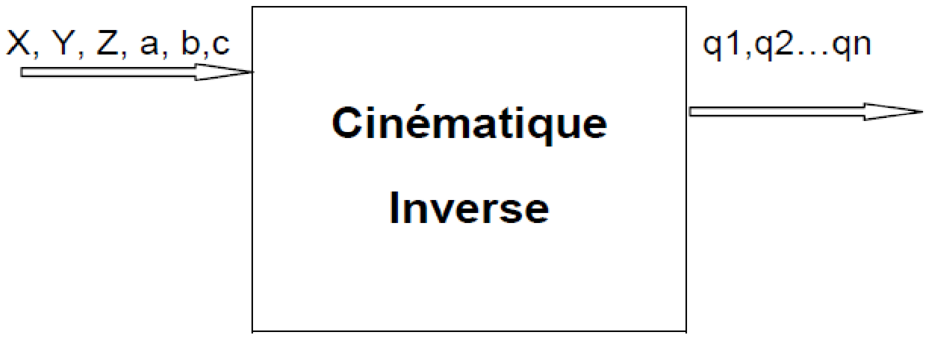
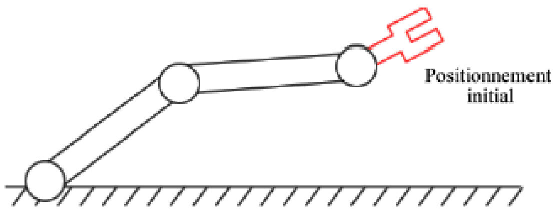


Figure 6:Déplacement du pied d'un squelette.

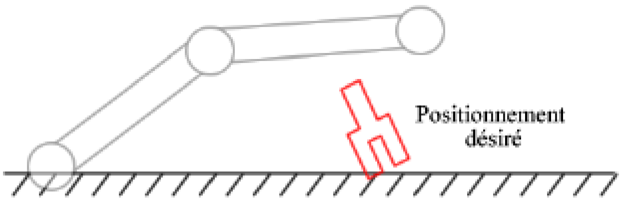
Un problème de cinématique inverse revient donc à trouver une configuration d’articulations dans le squelette permettant de positionner une articulation selon une orientation et une translation définie.
Trouver une configuration d’articulation pour obtenir le positionnement désiré de l’articulation cible est appelé la résolution du système de cinématique inverse.



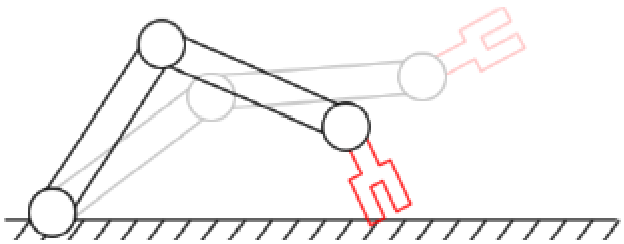
- L'articulation qu'on tente de
Positionner sera appelée "articulation cible".



- Le positionnement de l'articulation cible
  Comprend sa position et son orientation.



On cherche comment modifier les articulations du bras pour arriver au positionnement désiré.



Une configuration valide permettant d'obtenir le positionnement désiré est générée.

Ce problème est beaucoup plus complexe que celui de la cinématique directe car pour une position donnée, la solution n'est pas unique (plusieurs configurations possibles).

Simulation:

MATLAB (Peter Corne robotics Toolbox):

```
>> %% main robotique a six degrés de liberte(6DoF)
%%%%%%%%%%Peter corne robotics toolbox%%%%%%%%%%%
% on définit les longeurs de la main selon la methode de DH
L1=10;
L2=9;
L3=9;
L6=6;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%************L(i)=Link([Th d a alpha])*************

L(1)=Link([0 L1 0  pi/2]);
L(2)=Link([0 0  L2  0]);
L(3)=Link([0 0  0  pi/2]);
L(4)=Link([0 L3 0  pi/2]);
L(5)=Link([0 0  0  -pi/2]);
L(6)=Link([0 L6 0  0]);

Robot=SerialLink(L);
Robot.name='6DoF';
%% inverse kinematics
Px=3 ; Py=3 ; Pz= 3;
T=[1 0 0 Px ;
   0 1 0 Py ;
   0 0 1 Pz ;
   0 0 0 1] ;
J=Robot.ikine(T,[0 0 0 0 0 0 ],[1 1 1 0 0 0 ])   ;
Robot.plot(J)
>>
```

Figure 7:Code Matlab.

Pour savoir les angles des articulations on fait la commande :

```
J =

    0.5007    1.0234   -0.9385    0.5709    0.3764         0
```

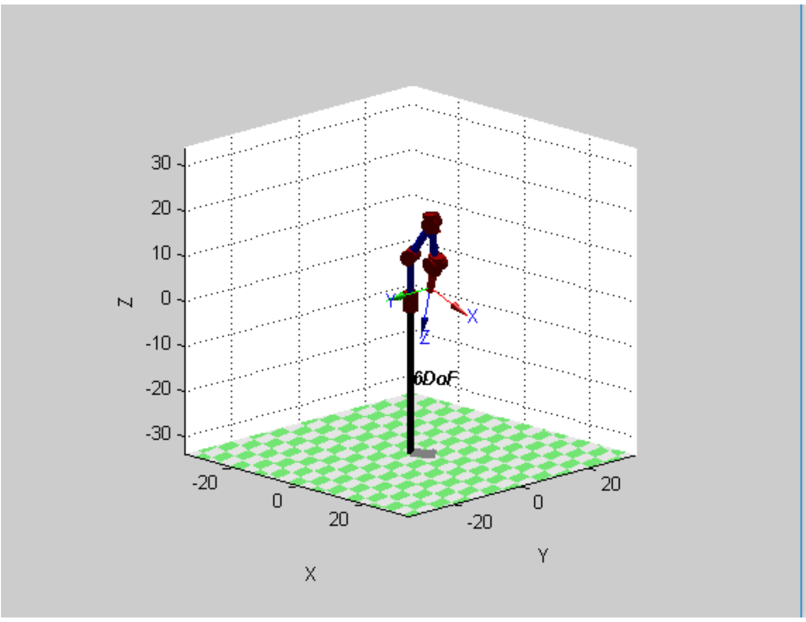


Figure 8:Mouvement du bras au coordonnées demandées.

Conclusion :

Le modèle géométrique directe du robot nous permet de trouver les coordonnées opérationnelles en donnant les informations de l'organe terminal en fonction des coordonnes articulaires. Le problème inverse consiste à calculer les coordonnées articulaires correspondant à une location de l'organe terminal.