Nom prénom: Lu Zhichen
Numéro de carte d’étudiant: 21117174
SAR

Compte rendu

MU4RBI04 Robotique

(TP 1)

Résolution du modèle géométrique inverse du bras STAUBLI RX90

1. Implémenter le code

1) Fait sur le fichier de code.

2) Pour l'exemple:

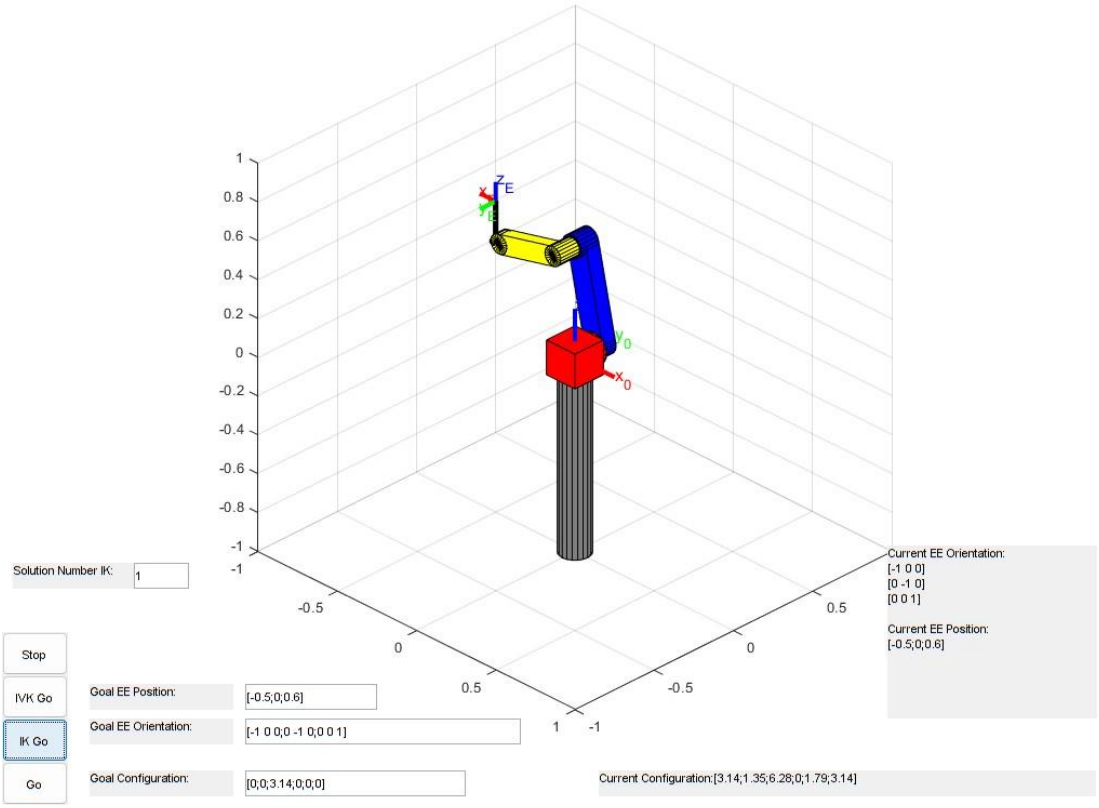
```
% For example :
% For p_E = [-0.5;0;0.6] and R_E = [-1 0 0;0 -1 0;0 0 1];
%
% q =
%
%     3.1416    1.3495         0   -0.0000    1.7921   -3.1416
%     3.1416   -0.0000   -3.1416   -1.5708    0.0000   -1.5708
%     3.1416    1.3495         0    3.1416   -1.7921         0
%     3.1416   -0.0000   -3.1416    1.5708   -0.0000    1.5708
%     6.2832   -3.1416   -0.0000    1.5708    0.0000   -1.5708
%     6.2832    1.7921    3.1416    3.1416    1.7921   -3.1416
%     6.2832   -3.1416   -0.0000    4.7124   -0.0000    1.5708
%     6.2832    1.7921    3.1416    6.2832   -1.7921         0
```

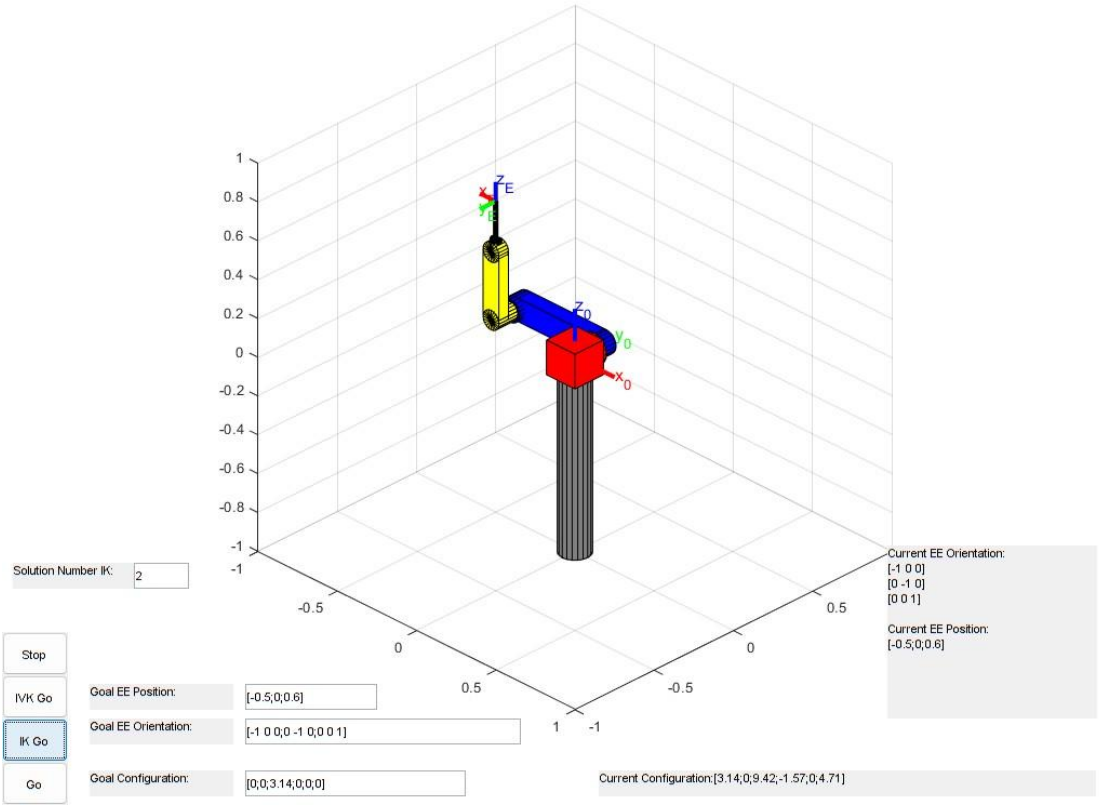
Les solution obtenus sont:

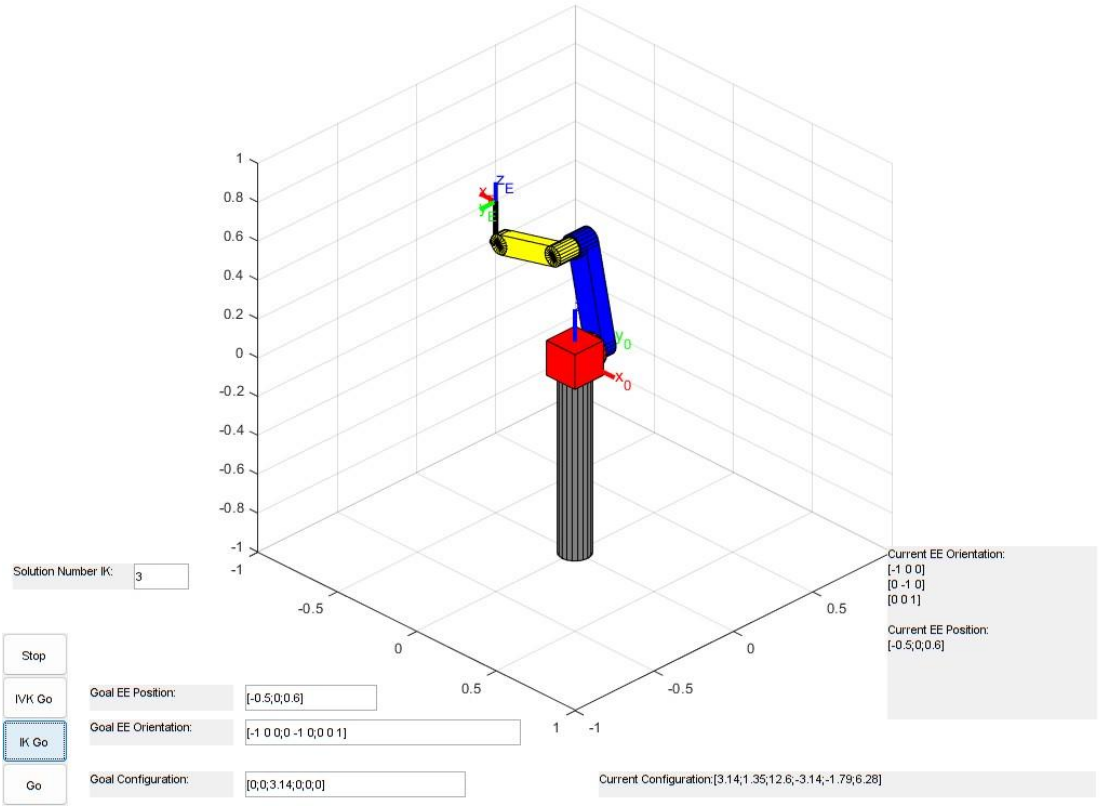
```
Q =
    3.1416    1.3495   -0.0000    0.0000    1.7921   -3.1416
    3.1416   -0.0000   -3.1416   -1.5708    0.0000   -1.5708
    3.1416    1.3495   -0.0000    3.1416   -1.7921         0
    3.1416   -0.0000   -3.1416    1.5708   -0.0000    1.5708
    6.2832   -3.1416   -0.0000    1.5708    0.0000   -1.5708
    6.2832    1.7921   -3.1416    3.1416    1.7921   -3.1416
    6.2832   -3.1416   -0.0000    4.7124   -0.0000    1.5708
    6.2832    1.7921   -3.1416    6.2832   -1.7921         0
```

Notez que le theta3 des sixième et huitième solutions est -pi au lieu de pi. En effet, dans matlab (version 2022a), arctan(0+) est -pi et atan(0-) est pi, mais il est considéré comme correct compte tenu de l'équivalent lors de la rotation.

Représentations graphiques des 8 configurations obtenues par le MGI:







Solution Number IK: 3

Current EE Orientation:
[-1 0 0]
[0 -1 0]
[0 0 1]

Current EE Position:
[-0.5;0;0.6]

Stop

IVK Go

IK Go

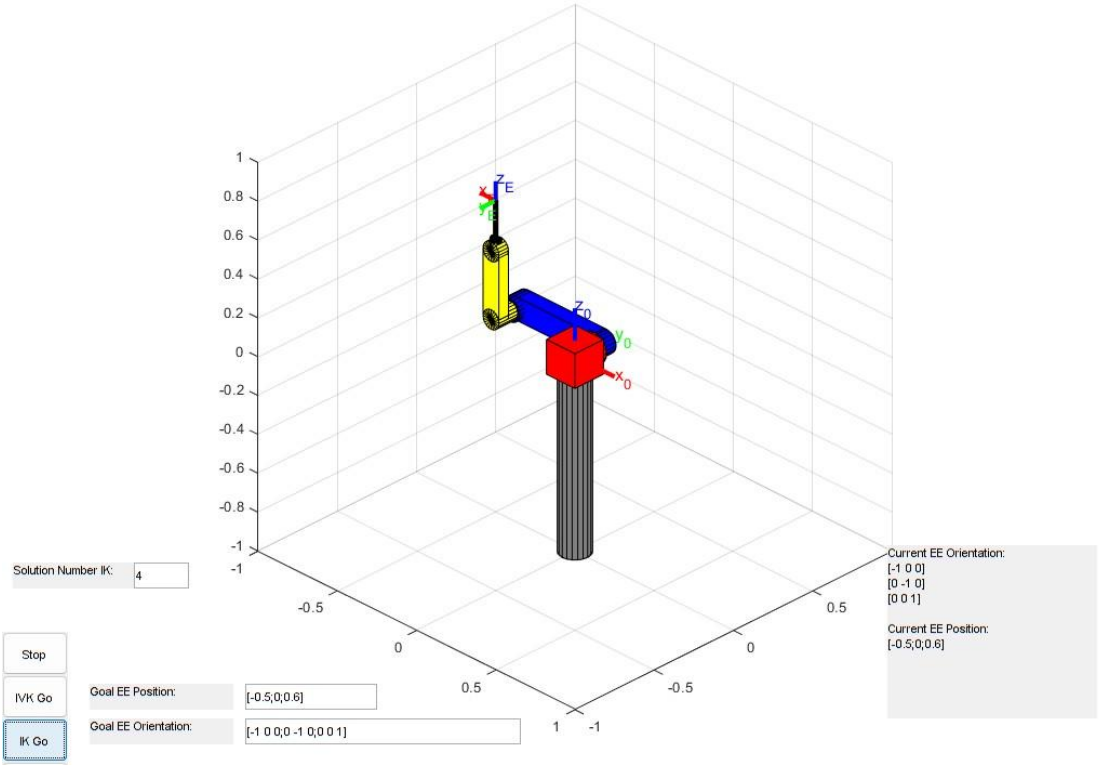
Go

Goal EE Position: [-0.5;0;0.6]

Goal EE Orientation: [-1 0 0;0 -1 0;0 0 1]

Goal Configuration: [0;0;3.14;0;0;0]

Current Configuration:[3.14;1.35;12.6;-3.14;-1.79;6.28]



Solution Number IK: 4

Current EE Orientation:
[-1 0 0]
[0 -1 0]
[0 0 1]

Current EE Position:
[-0.5;0;0.6]

Stop

IVK Go

IK Go

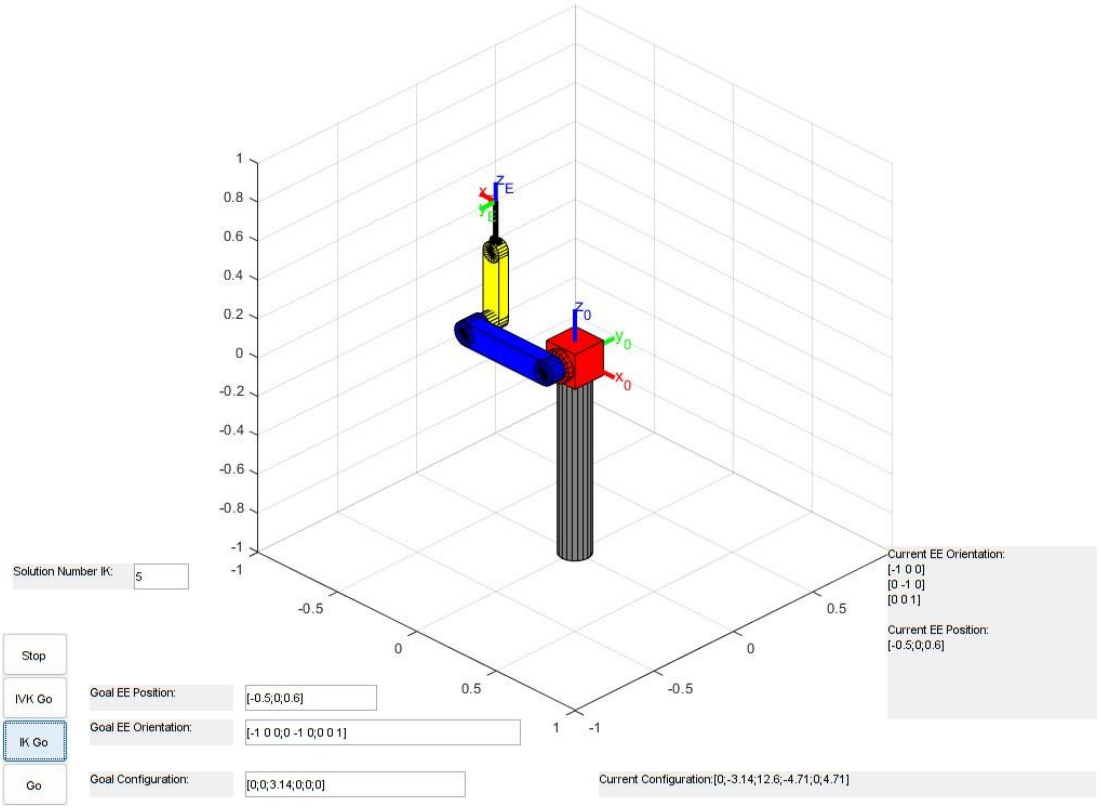
Go

Goal EE Position: [-0.5;0;0.6]

Goal EE Orientation: [-1 0 0;0 -1 0;0 0 1]

Goal Configuration: [0;0;3.14;0;0;0]

Current Configuration:[3.14;0;9.42;-4.71;0;7.85]



Solution Number IK: 5

Current EE Orientation:
[-1 0 0]
[0 -1 0]
[0 0 1]

Current EE Position:
[-0.5;0;0.6]

Stop

IVK Go

IK Go

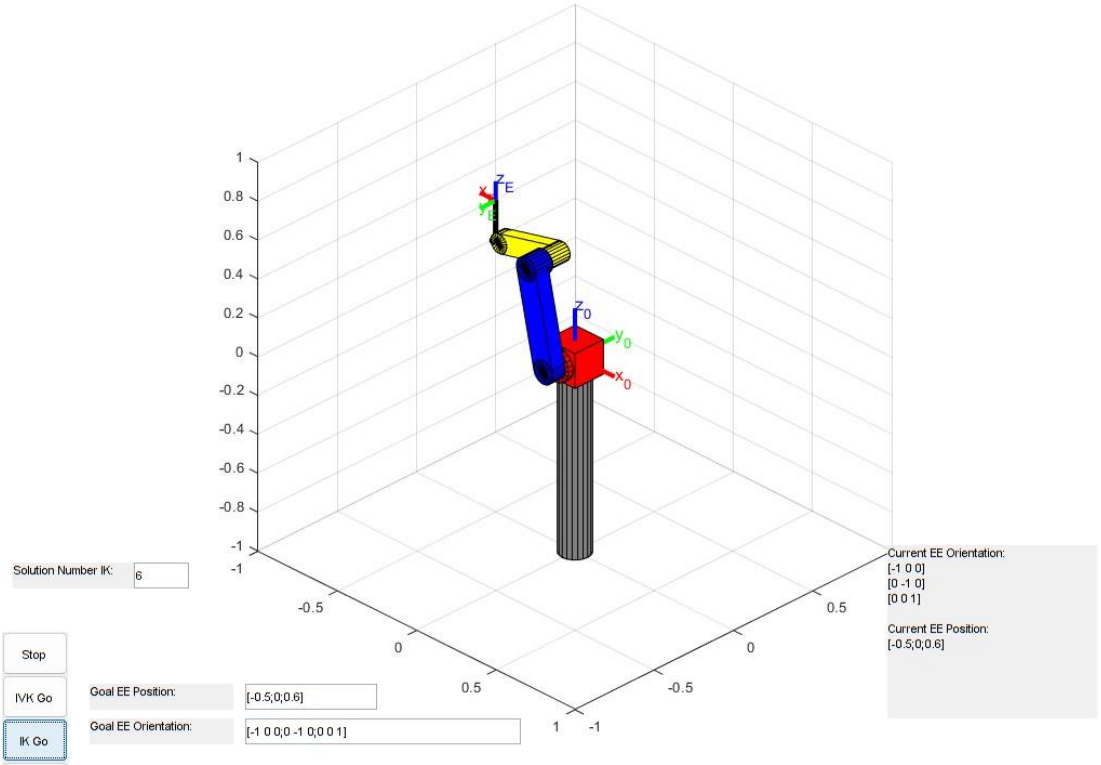
Go

Goal EE Position: [-0.5;0;0.6]

Goal EE Orientation: [-1 0 0;0 -1 0;0 0 1]

Goal Configuration: [0;0;3.14;0;0;0]

Current Configuration:[0;-3.14;12.6;-4.71;0;4.71]



Solution Number IK: 6

Current EE Orientation:
[-1 0 0]
[0 -1 0]
[0 0 1]

Current EE Position:
[-0.5;0;0.6]

Stop

IVK Go

IK Go

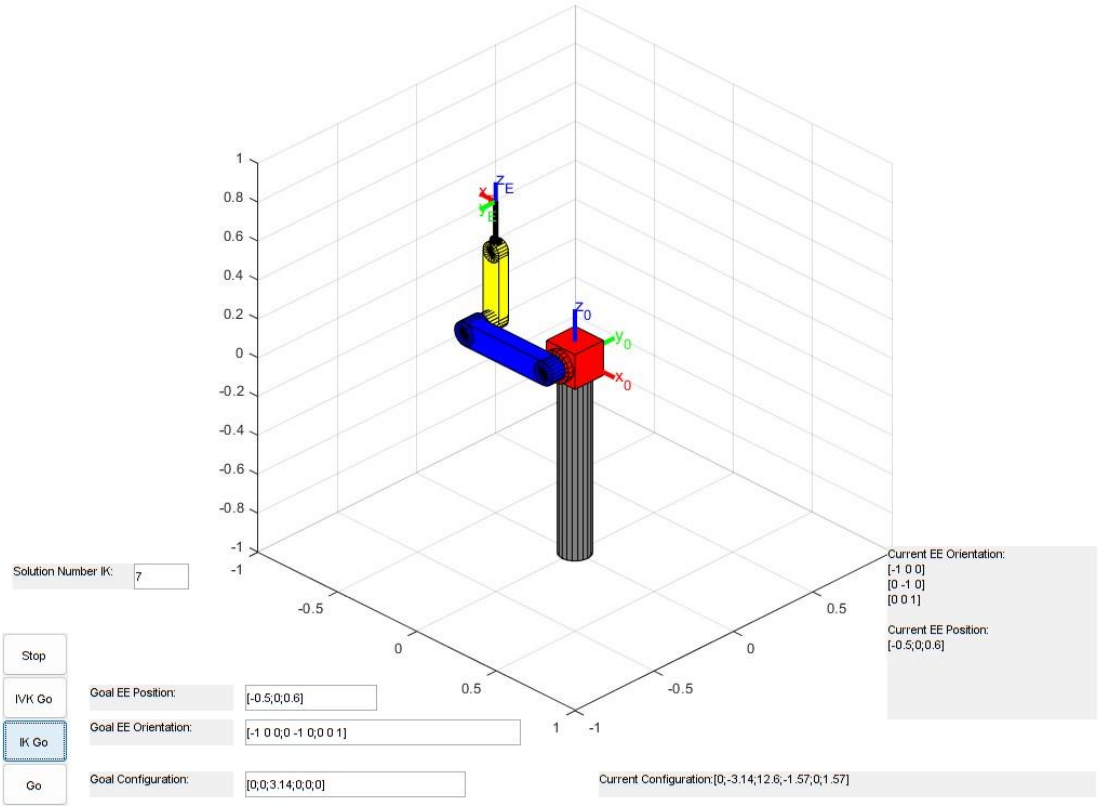
Go

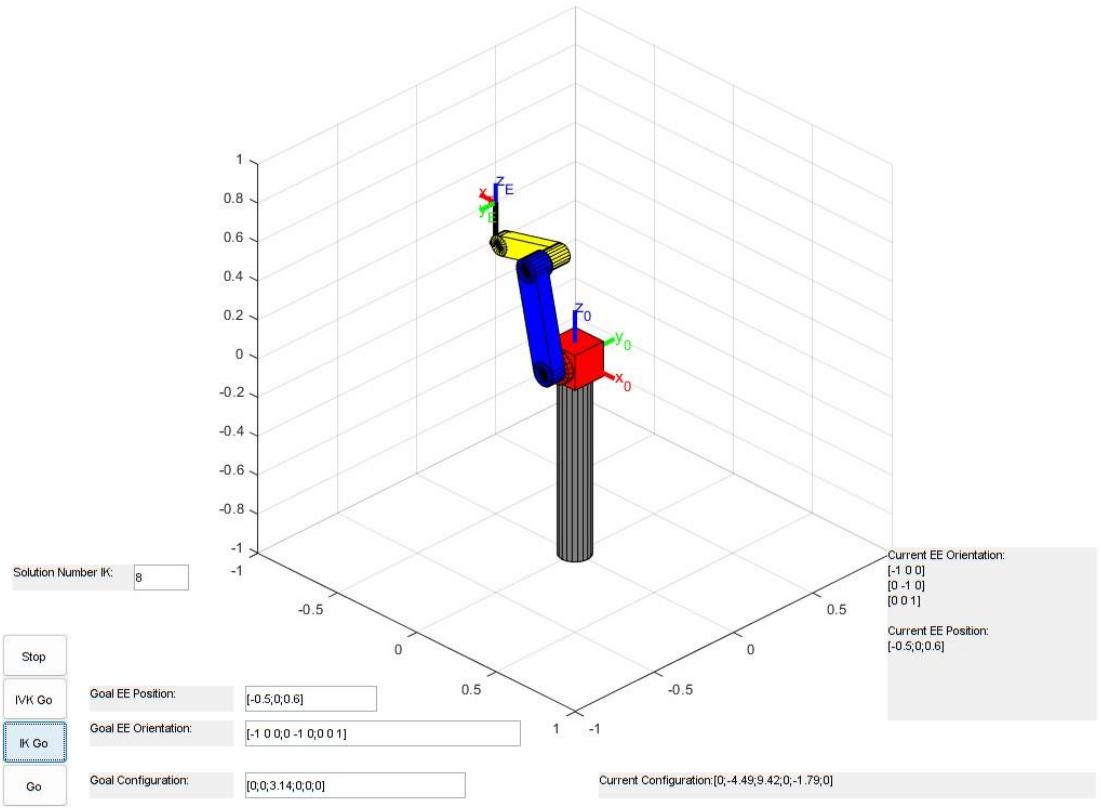
Goal EE Position: [-0.5;0;0.6]

Goal EE Orientation: [-1 0 0;0 -1 0;0 0 1]

Goal Configuration: [0;0;3.14;0;0;0]

Current Configuration:[0;-4.49;9.42;-3.14;1.79;3.14]


E
0
Solution Number IK:
7
Stop
IVK Go
IK Go
Go
Goal EE Position:
[-0.5;0;0.6]
Goal EE Orientation:
[-1 0 0;0 -1 0;0 0 1]
Goal Configuration:
[0;0;3.14;0;0;0]
Current EE Orientation:
[-1 0 0]
[0 -1 0]
[0 0 1]
Current EE Position:
[-0.5;0;0.6]
Current Configuration:[0;-3.14;12.6;-1.57;0;1.57]

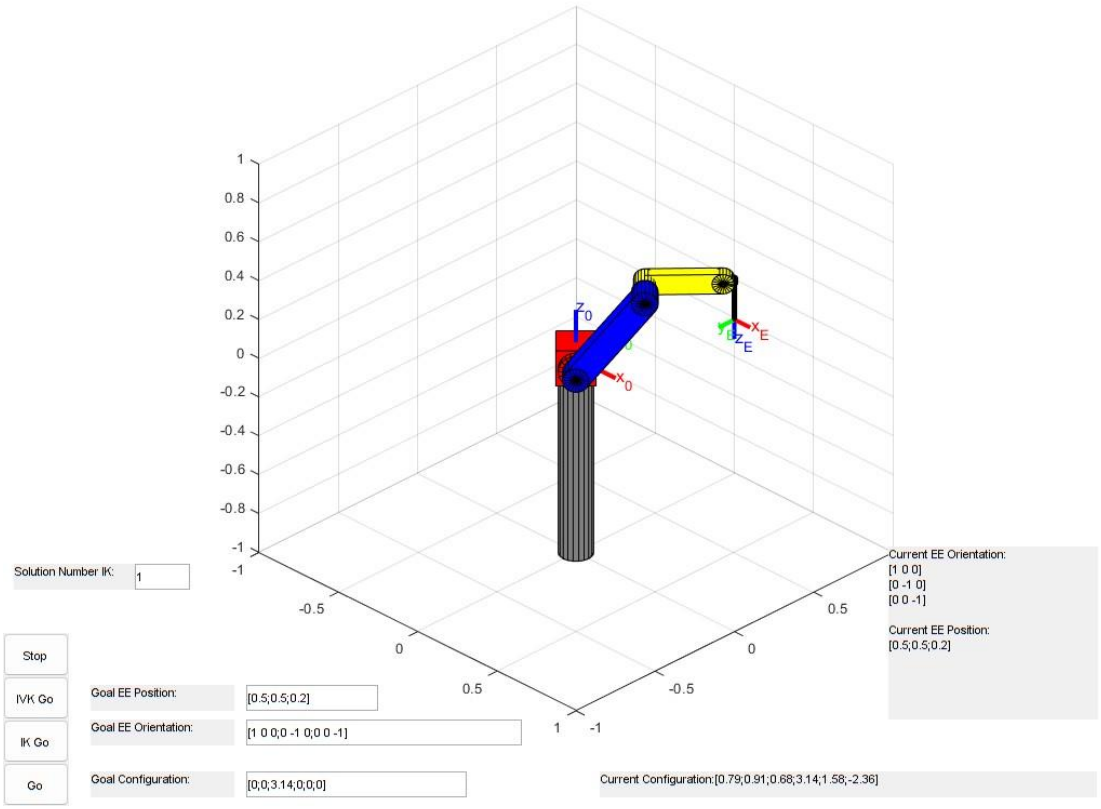

E
0
Solution Number IK:
8
Stop
IVK Go
IK Go
Go
Goal EE Position:
[-0.5;0;0.6]
Goal EE Orientation:
[-1 0 0;0 -1 0;0 0 1]
Goal Configuration:
[0;0;3.14;0;0;0]
Current EE Orientation:
[-1 0 0]
[0 -1 0]
[0 0 1]
Current EE Position:
[-0.5;0;0.6]
Current Configuration:[0;-4.49;9.42;0;-1.79;0]

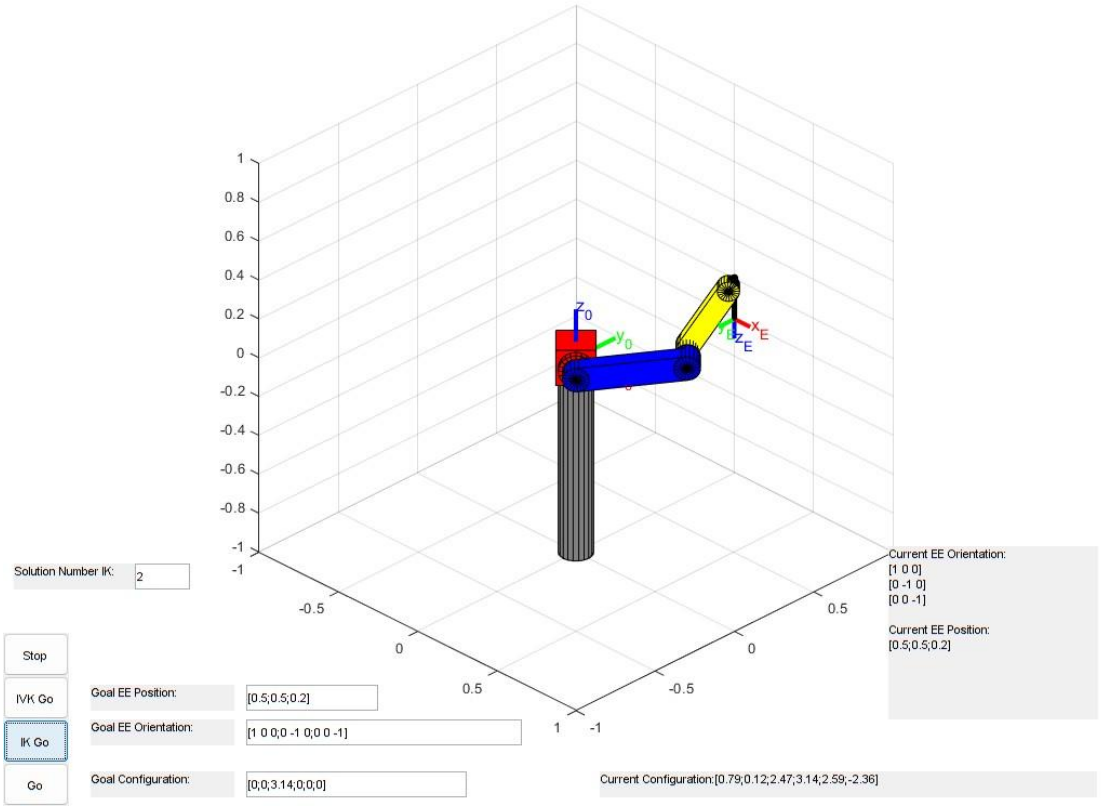
3) La pose nouvelle est choisi:[0.5, 0.5, 0.2] et [1 0 0;0 -1 0;0 0 -1]

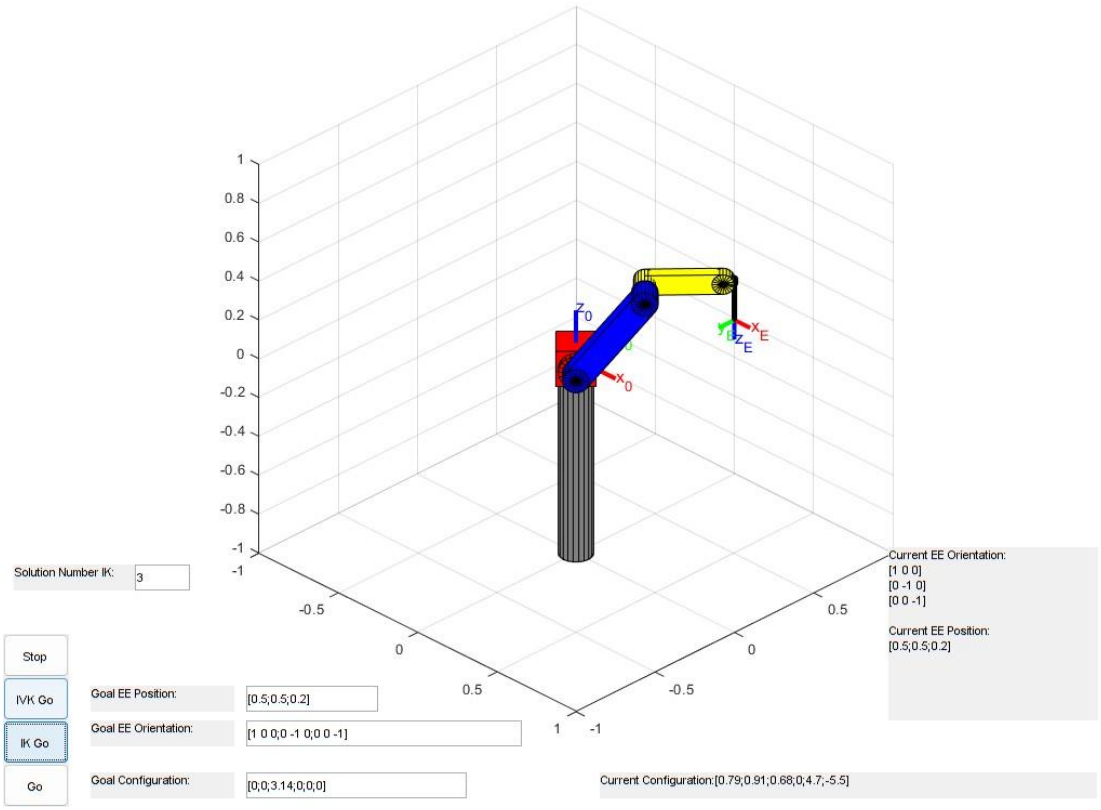
Les solutions obtenus sont:

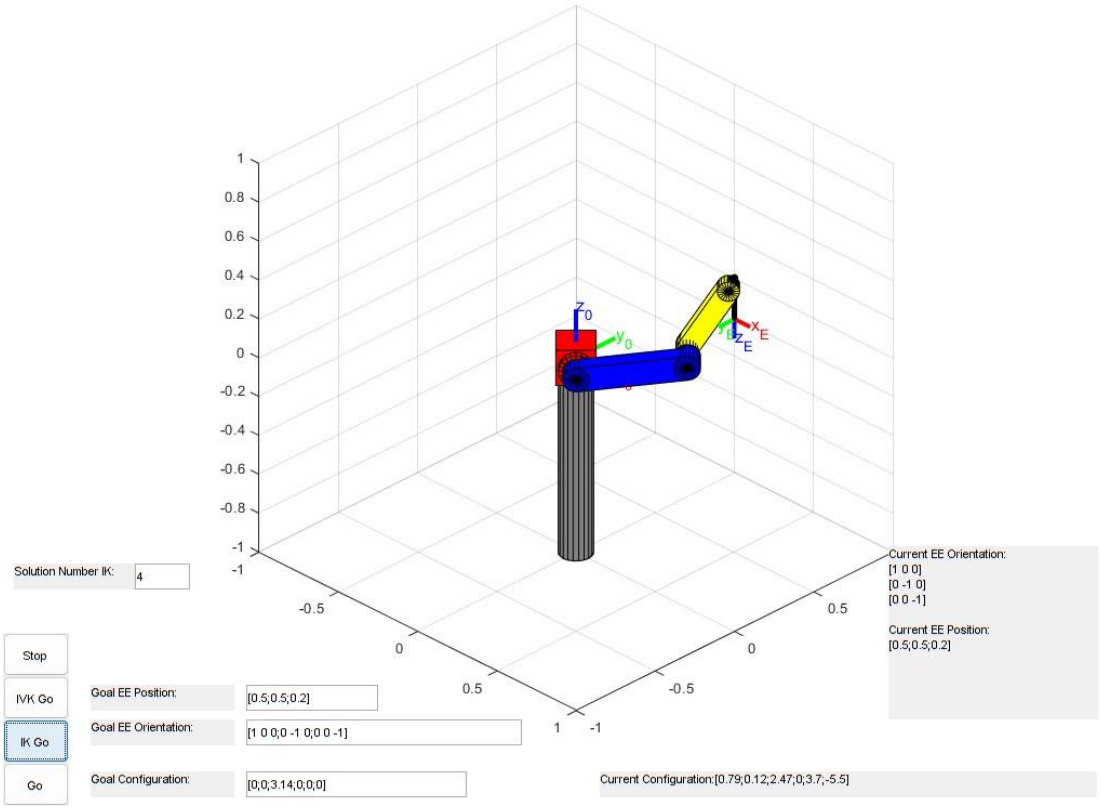
```
Q =
    0.7854    0.9093    0.6751    3.1416    1.5844   -2.3562
    0.7854    0.1203    2.4665   -3.1416    2.5868   -2.3562
    0.7854    0.9093    0.6751    6.2832   -1.5844    0.7854
    0.7854    0.1203    2.4665    0.0000   -2.5868    0.7854
    3.9270    3.0213    0.6751    0.0000    2.5868   -2.3562
    3.9270    2.2323    2.4665    0.0000    1.5844   -2.3562
    3.9270    3.0213    0.6751    3.1416   -2.5868    0.7854
    3.9270    2.2323    2.4665    3.1416   -1.5844    0.7854
```

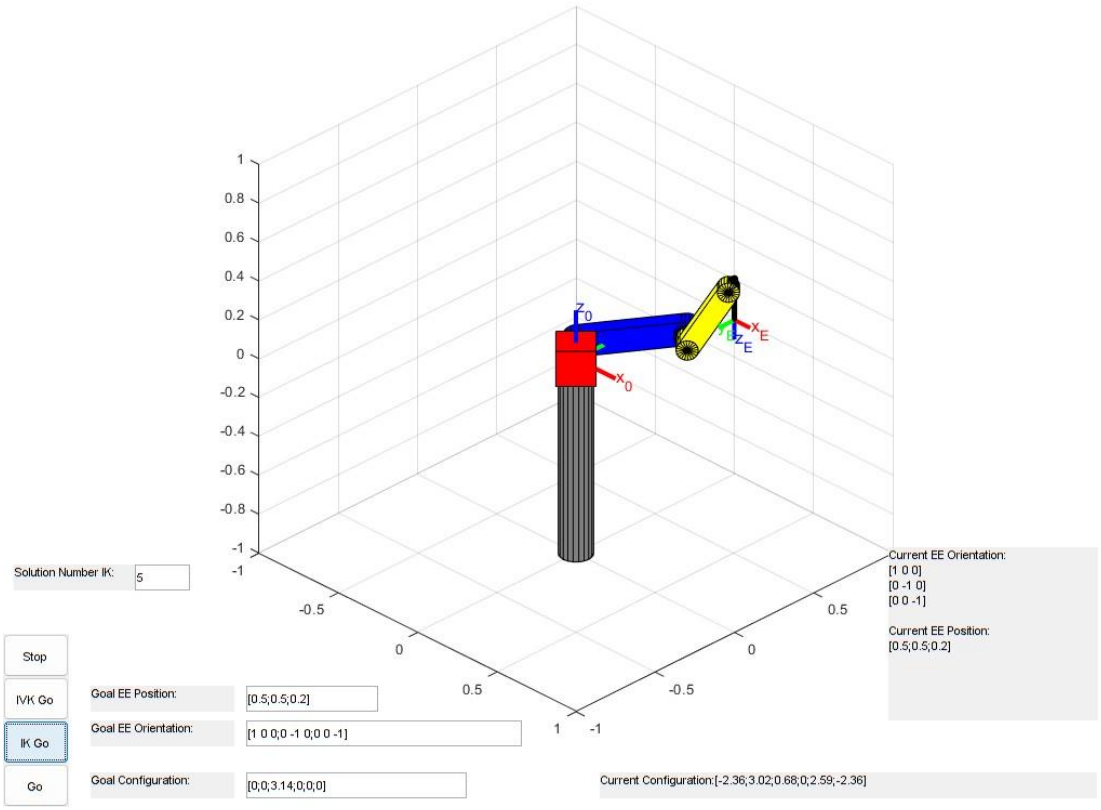
Représentations graphiques des 8 configurations obtenues par le MGI:

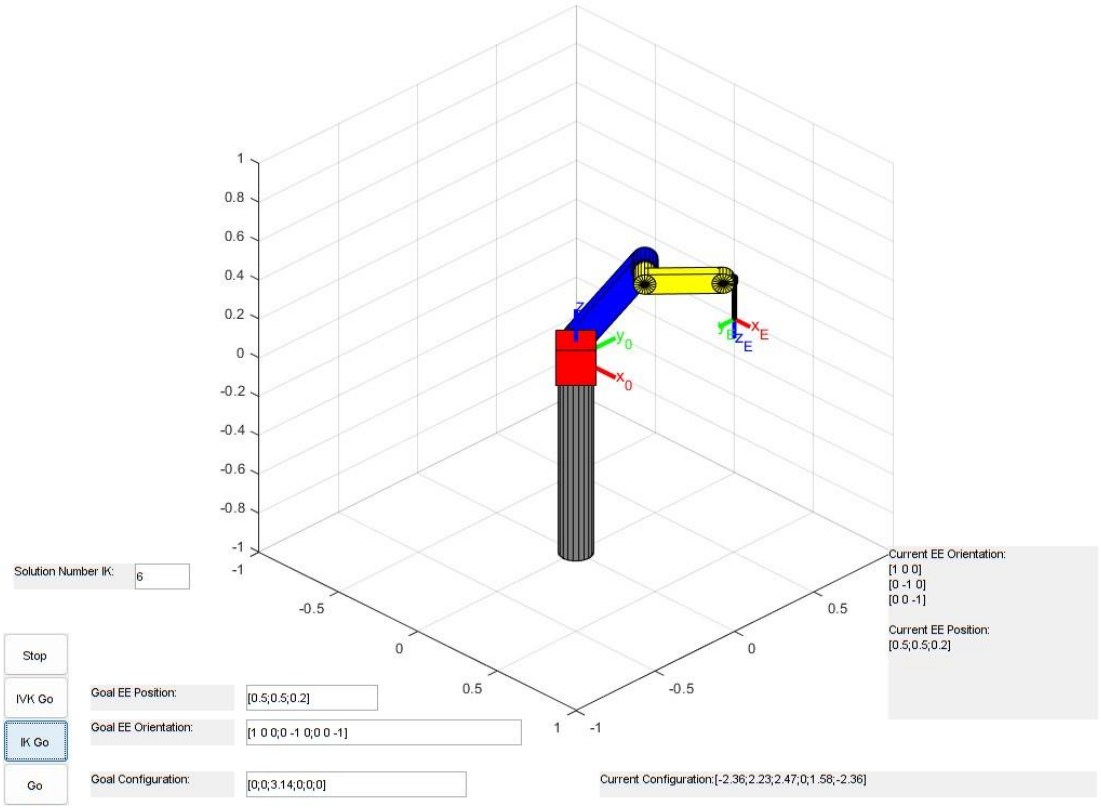


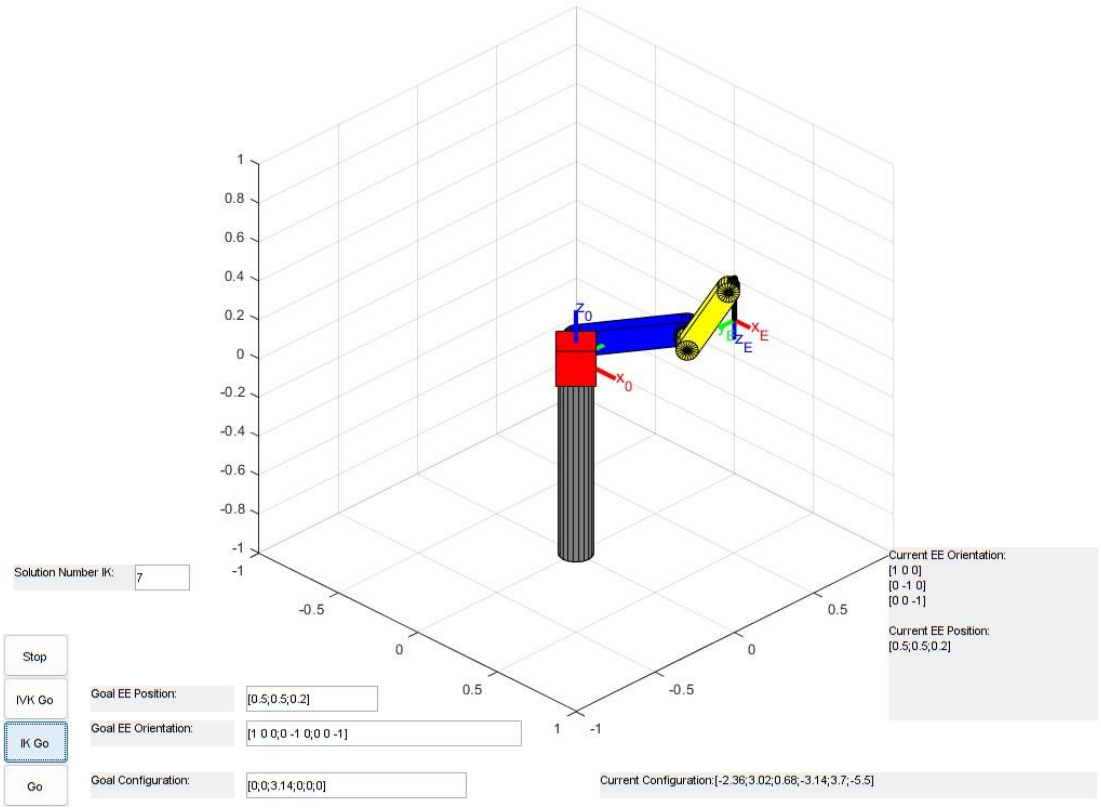

Solution Number IK:
2
Stop
IVK Go
IK Go
Go
Goal EE Position:
[0.5;0.5;0.2]
Goal EE Orientation:
[1 0 0;0 -1 0;0 0 -1]
Goal Configuration:
[0;0;3.14;0;0;0]
Current EE Orientation:
[1 0 0]
[0 -1 0]
[0 0 -1]
Current EE Position:
[0.5;0.5;0.2]
Current Configuration:[0.79;0.12;2.47;3.14;2.59;-2.36]

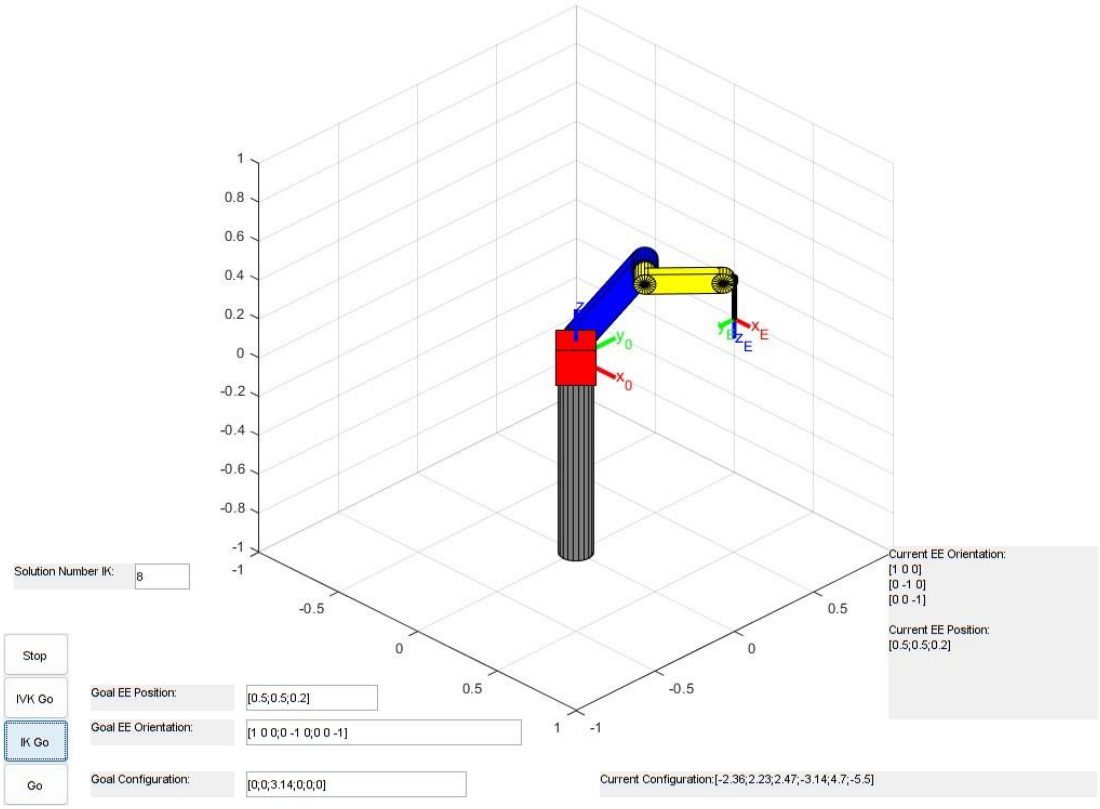

Solution Number IK:
3
Stop
IVK Go
IK Go
Go
Goal EE Position:
[0.5;0.5;0.2]
Goal EE Orientation:
[1 0 0;0 -1 0;0 0 -1]
Goal Configuration:
[0;0;3.14;0;0;0]
Current EE Orientation:
[1 0 0]
[0 -1 0]
[0 0 -1]
Current EE Position:
[0.5;0.5;0.2]
Current Configuration:[0.79;0.91;0.68;0;4.7;-5.5]


Solution Number IK:
4
Stop
IVK Go
IK Go
Go
Goal EE Position:
[0.5;0.5;0.2]
Goal EE Orientation:
[1 0 0;0 -1 0;0 0 -1]
Goal Configuration:
[0;0;3.14;0;0;0]
Current EE Orientation:
[1 0 0]
[0 -1 0]
[0 0 -1]
Current EE Position:
[0.5;0.5;0.2]
Current Configuration:[0.79;0.12;2.47;0;3.7;-5.5]


Solution Number IK:
5
Stop
IVK Go
IK Go
Go
Goal EE Position:
[0.5;0.5;0.2]
Goal EE Orientation:
[1 0 0;0 -1 0;0 0 -1]
Goal Configuration:
[0;0;3.14;0;0;0]
Current EE Orientation:
[1 0 0]
[0 -1 0]
[0 0 -1]
Current EE Position:
[0.5;0.5;0.2]
Current Configuration:[-2.36;3.02;0.68;0;2.59;-2.36]


Solution Number IK:
6
Stop
IVK Go
IK Go
Go
Goal EE Position:
[0.5;0.5;0.2]
Goal EE Orientation:
[1 0 0;0 -1 0;0 0 -1]
Goal Configuration:
[0;0;3.14;0;0;0]
Current EE Orientation:
[1 0 0]
[0 -1 0]
[0 0 -1]
Current EE Position:
[0.5;0.5;0.2]
Current Configuration:[-2.36;2.23;2.47;0;1.58;-2.36]


Solution Number IK:
7
Stop
IVK Go
IK Go
Go
Goal EE Position:
[0.5;0.5;0.2]
Goal EE Orientation:
[1 0 0;0 -1 0;0 0 -1]
Goal Configuration:
[0;0;3.14;0;0;0]
Current EE Orientation:
[1 0 0]
[0 -1 0]
[0 0 -1]
Current EE Position:
[0.5;0.5;0.2]
Current Configuration:[-2.36;3.02;0.68;-3.14;3.7;-5.5]



4) L'existence de singularités doit être prise en compte.

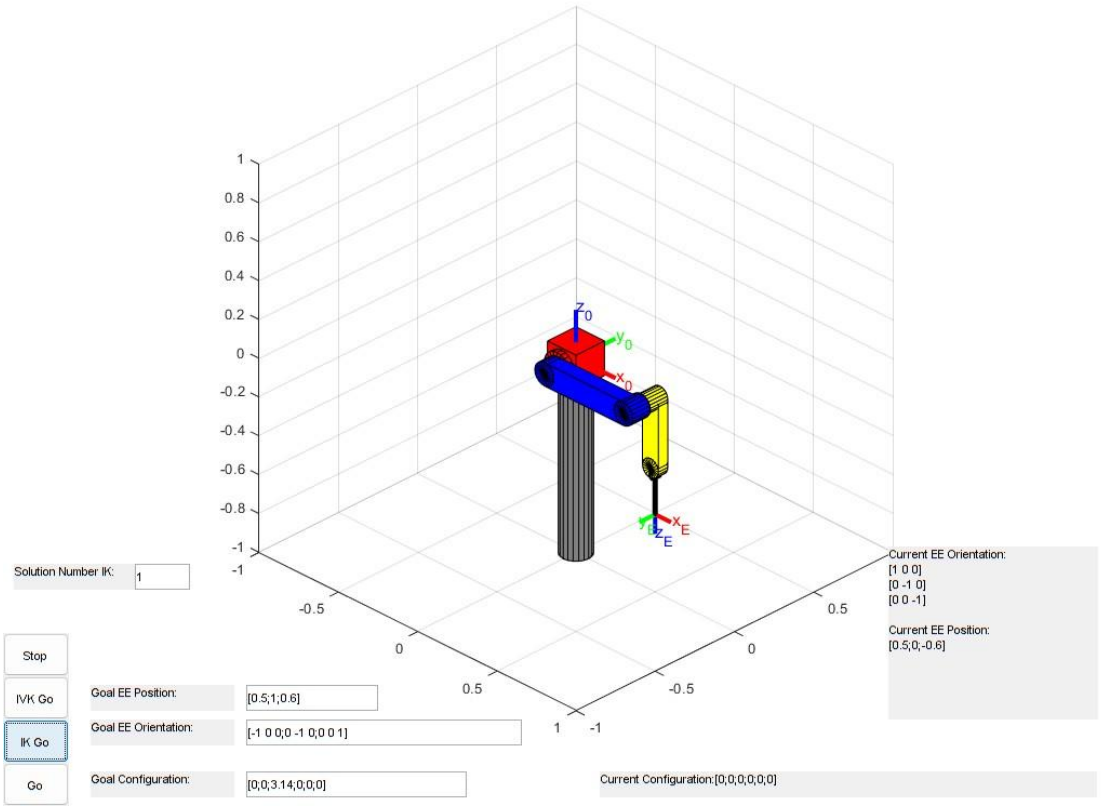
5) Fait sur le fichier de code(Q5.5:L'existence de singularités doit être prise en compte.)

```
    %-------------------------------------------------------------
    %Q5.5:L'existence de singularités doit être prise en compte.
    q_cor = [];
    for i = 1:8
        if abs(q(i,2) - q(i,3)) == pi || abs(q(i,2) - q(i,3)) < c || abs(q(i,3) - q(i,5)) == pi || abs(q(i,3) - q(i,5)) < c || abs(q(
            q_cor = [q_cor;[]];
        else
            q_cor = [q_cor;q(i,:)];
        end
    end
    conf = q_cor(IndSolLoc,:);
    disp('Q(sans singularité) = ');
    disp(q_cor);
end
```

6) Pour ce RX90, l'espece de travail est une sphère de rayon r=L2+L3+L6 =1.1.Car il peut arriver au point (0,0,0).(Fait sur le fichier de code)

```
    %Q5.6:Traiter les poses non-atteignablese.
    %Quand c'est la pose non-atteignable, toutes les thetas sont 0.
    r = L2+L3+L6;
    if p(1)^2 + p(2)^2 + p(3)^2 >= r^2
        fprintf('La pose non-atteignable');
        conf = q(IndSolLoc,:);
        return
    end
```

Dans ce cas, les thetas sont 0 donc l'affiche est suivant:(la pose est[0.5,1.0,d0,6])



2. Analyse

1) Pour une pose de l'effffecteur, est-il possible d'aller d'une confifiguration θ à une autre tout en maintenant cette pose ? pourquoi ?

Réponse:

Ce n'est pas possible.On peut considérer une situation simple pour ce problème.La figure ci-dessous est un bras mécanique composé d'une base, de deux articulations et de deux bras. Il est facile d'imaginer que lorsque la configuration change, il est impossible que l'extrémité de l'effecteur ne se déplace pas.

Donc, pour le bras robotisé en tp, même s'il existe plusieurs ensembles de configurations avec la même position tridimensionnelle de l'effecteur, si on veut passer d'une configuration à l'autre, il est impossible que l'effecteur ne bouge pas .

2) .Est-ce que le MGI prend en compte les contraintes d'évitement d'obstacle ? Comment cela peut être pris en compte ?

Réponse:

Le MGI ne prend en compte les contraintes d'évitement d'obstacle.

À mon avis, afin de permettre au bras du robot d'éviter les obstacles lors du déplacement, nous devons entrer les informations de position de l'obstacle dans l'équation cinématique avant du bras du robot. Mais considérant que les obstacles sont souvent des objets irréguliers, je pense que la méthode de la sphère ou du cône circonscrit peut être utilisée. C'est-à-dire que la position de l'obstacle est remplacée par la position de la sphère ou du cône circonscrit de l'obstacle, et ces positions limitées sont ajoutées dans la plage d'espace de mouvement du manipulateur.

3) Tout autre ajout ou analyse personnel sont le bienvenus.

Réponse:

À mon avis, pour que le bras robotisé puisse changer de configuration tout en conservant la pose de l'effecteur, il faut le faire avec des degrés de liberté supplémentaires. Mais plus de degrés de liberté signifient des calculs cinématiques avant et arrière plus complexes et un évitement d'obstacles plus complexe. Par conséquent, je pense que l'ajout d'un degré de liberté est un choix plus approprié pour le bras robotique 6-DDL couramment utilisé.

De plus, je pense qu'il est possible d'ajouter des capteurs au bras robotique pour identifier automatiquement les obstacles et calculer les postures de mouvement.

Commentaires

Sur Matlab,je trouve que la réponse obtenu par la code suivant:

Dossier: IKGoCallBack
Code:
ligne 14: qGoalLoc = IK_RX90(EEpGoalLoc,EEoGoalLoc,qLoc,IndSolLoc);

```
qGoalLoc = IK_RX90(EEpGoalLoc,EEoGoalLoc,qLoc,IndSolLoc);
disp('qGoqlLoc = ')
disp(qGoalLoc)
%qGoalLoc = transpose(qGoalLoc);
%disp('transpose(qGoqlLoc) = ')
%disp(qGoalLoc)

deltaQ = mod(qGoalLoc,2*pi) - qLoc;
disp('deltaQ = ')
disp(deltaQ)
```

Résultat:

```
Error using  /
Matrix dimensions must agree.

Error in IKGoCallback (line 36)
        qLoc(j) = qLoc(j) + deltaQ(j)/lim;

Error while evaluating UIControl Callback.
```

Mais après le corrigé:
J'ajoute transpose de qGoalLoc:

```
qGoalLoc = IK_RX90(EEpGoalLoc,EEoGoalLoc,qLoc,IndSolLoc);
disp('qGoqlLoc = ')
disp(qGoalLoc)
qGoalLoc = transpose(qGoalLoc);
disp('transpose(qGoqlLoc) = ')
disp(qGoalLoc)

deltaQ = mod(qGoalLoc,2*pi) - qLoc;
disp('deltaQ = ')
disp(deltaQ)
```

Résultat:

```
Q =
    3.1416    1.3495   -0.0000   -0.0000    1.7921   -3.1416
    3.1416   -0.0000   -3.1416   -1.5708    0.0000   -1.5708
    3.1416    1.3495   -0.0000    3.1416   -1.7921         0
    3.1416   -0.0000   -3.1416    1.5708   -0.0000    1.5708
    6.2832   -3.1416   -0.0000    1.5708    0.0000   -1.5708
    6.2832    1.7921   -3.1416    3.1416    1.7921   -3.1416
    6.2832   -3.1416   -0.0000    4.7124   -0.0000    1.5708
    6.2832    1.7921   -3.1416    6.2832   -1.7921         0

Q(sans singularité) =
    3.1416    1.3495   -0.0000   -0.0000    1.7921   -3.1416
    3.1416    1.3495   -0.0000    3.1416   -1.7921         0
    6.2832    1.7921   -3.1416    6.2832   -1.7921         0

qGoqlLoc =
    3.1416    1.3495   -0.0000   -0.0000    1.7921   -3.1416

transpose(qGoqlLoc) =
    3.1416
    1.3495
   -0.0000
   -0.0000
    1.7921
   -3.1416

deltaQ =
    3.1416
    1.3495
    3.1416
    6.2832
    1.7921
    3.1416
```

Il est très étrange que la sortie ***qGoalLoc*** de cette ligne de code soit un vecteur colonne sur l'ordinateur de l'école et les ordinateurs des autres camarades de classe, mais la sortie sur mon ordinateur est un vecteur ligne. J'ai donc besoin de modifier pour transposer la matrice. Mention spéciale en cas d'échec de son exécution sur votre ordinateur. Si cela ne fonctionne pas, c'est un problème avec la sortie de la matrice.