TD 3/4 - Kinematics of manipulators

Groupe :

- **MARÇAL Thomas**
- **GOSSELIN Julie**
- **KOSKAS Axel**
- **KERMEL Aurore**

---

1. **On your report, write briefly on what does this script do and how can it be employed in MATLAB?**

The script in the DenaHart.m file, is employed to calculate the homogeneous transformation matrix.

First, we call the function : `function [T0Tn,entities] = DenaHart(alpha, d, theta, r)` on the left side we have the outputs and on the right we have the inputs.

Next we implement the variables :
alpha = rotation about x
theta = rotation about z
d = distance between current and last z
r = distance between current and previous x

After, we enter in a “for loop”:
Firstly we need to calculate T0Tn = ${}^{0}T1$

Secondly, we end up in the ‘else’, here we will calculate ${}^{1}T2$ * T0Tn
The result will be called T0Tn etc…

Last, we form the matrix T : the last column gives you the direct kinematics.

I. The 2-R planar robot

1.
Denavit-Hartenberg table for the mechanism :

i	$\propto_{i-1}$	θ_i	d	r_{i-1}
1	0	θ_1	0	0
2	0	θ_2	L1	0
end-effector	0	0	L2	0

2. With our code we had :

```
TOTn =

[cos(q1 + q2), -sin(q1 + q2), 0, L2*cos(q1 + q2) + L1*cos(q1)]
[sin(q1 + q2),  cos(q1 + q2), 0, L2*sin(q1 + q2) + L1*sin(q1)]
[           0,             0, 1,                            0]
[           0,             0, 0,                            1]
```

position of the end-effector

After identification, the end-effector position is :

(P_x, P_y) = (L2*cos(q1 + q2) + L1*cos(q1), L2*sin(q1 + q2) + L1*sin(q1))

3. Thanks to the code established on Matlab to obtain the Jacobian matrix, we obtain the following Jacobian matrix for the 2-R planar robot:

```
J =

[- L2*sin(q1 + q2) - L1*sin(q1), -L2*sin(q1 + q2)]
[  L2*cos(q1 + q2) + L1*cos(q1),  L2*cos(q1 + q2)]
[                             0,                0]
[                             0,                0]
[                             0,                0]
[                             1,                1]
```

In order to identify the singular position of the system, we will simplify our Jacobian matrix and calculate its determinant, using MATLAB.

```
JSimplify =

[- L2*sin(q1 + q2) - L1*sin(q1), -L2*sin(q1 + q2)]
[  L2*cos(q1 + q2) + L1*cos(q1),  L2*cos(q1 + q2)]


detJ =

L1*L2*sin(q1 + q2)*cos(q1) - L1*L2*cos(q1 + q2)*sin(q1)
```

The following calculation is performed to determine the singular position of the system:

$$L_1 L_2 \sin(q1 + q2)\cos(q1) - L_1 L_2 \cos(q_1 + q_2)\sin(q_1) = 0$$

$$\Leftrightarrow \quad \sin(2q_1 + q2) = 0$$

car $\sin(a+b) = \sin a \cos b + \sin b \cos a$

$$\Leftrightarrow \forall k \in \mathbb{Z}, \; 2q_1 + q2 = 2k\pi$$

II. The 3-R planar robot

1. Denavit-Hartenberg table for the mechanism :

i	**$\propto_{i-1}$**	**θ_i**	**d**	**r_{i-1}**
1	0	θ_1	0	0
2	0	θ_2	L1	0
3	0	θ_3	L2	0
end-effector	0	0	L3	0

2. With our code we obtained this matrix:

```
TOTn =

[cos(q1 + q2 + q3), -sin(q1 + q2 + q3), 0, L2*cos(q1 + q2) + L1*cos(q1) + L3*cos(q1 + q2 + q3)]
[sin(q1 + q2 + q3),  cos(q1 + q2 + q3), 0, L2*sin(q1 + q2) + L1*sin(q1) + L3*sin(q1 + q2 + q3)]
[                0,                  0, 1,                                                  0]
[                0,                  0, 0,                                                  1]
```

By identification, the end-effector position is:

$$\mathbf{P_x = L2*cos(q1 + q2) + L1*cos(q1) + L3*cos(q1 + q2 + q3)}$$
$$\mathbf{P_y = L2*sin(q1 + q2) + L1*sin(q1) + L3*sin(q1 + q2 + q3)}$$

3. Thanks to the code established on Matlab to obtain the Jacobian matrix, we obtain the following Jacobian matrix for the 3-R planar robot:

```
J =

[- L2*sin(q1 + q2) - L1*sin(q1) - L3*sin(q1 + q2 + q3), - L2*sin(q1 + q2) - L3*sin(q1 + q2 + q3), -L3*sin(q1 + q2 + q3)]
[  L2*cos(q1 + q2) + L1*cos(q1) + L3*cos(q1 + q2 + q3),   L2*cos(q1 + q2) + L3*cos(q1 + q2 + q3),  L3*cos(q1 + q2 + q3)]
[                                                     0,                                         0,                     0]
[                                                     0,                                         0,                     0]
[                                                     0,                                         0,                     0]
[                                                     1,                                         1,                     1]
```

In order to identify the singular position of the system, we will simplify our Jacobian matrix and calculate its determinant, using MATLAB.

```
JSimplify =

[- L2*sin(q1 + q2) - L1*sin(q1) - L3*sin(q1 + q2 + q3), - L2*sin(q1 + q2) - L3*sin(q1 + q2 + q3), -L3*sin(q1 + q2 + q3)]
[  L2*cos(q1 + q2) + L1*cos(q1) + L3*cos(q1 + q2 + q3),   L2*cos(q1 + q2) + L3*cos(q1 + q2 + q3),  L3*cos(q1 + q2 + q3)]
[                                                     0,                                         0,                     0]


detJ =

0
```

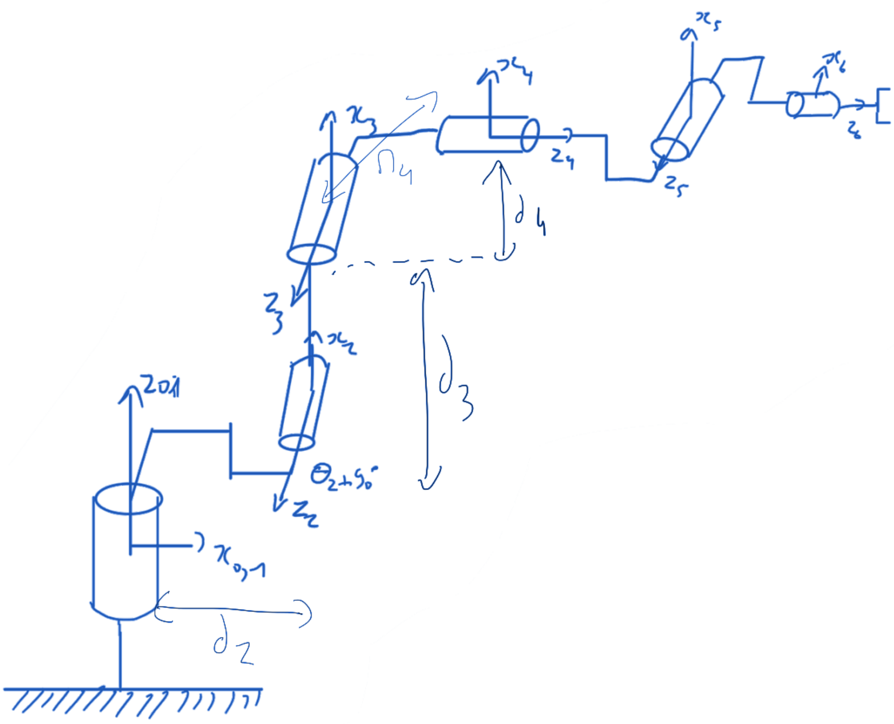
Our system is therefore always in a singular position.

III. FANUC LR MATE 200iB

1. Denavit-Hartenberg table for the mechanism :

i	$\propto_{i-1}$	θ_i	d	r_{i-1}
1	0	θ_1	0	0
2	90°	θ_2+90°	L2	0
3	0	θ_3	L3	0
4	90°	θ_4	L4	r4
5	-90°	θ_5	0	0
6	90°	θ_6	0	0

DH representation of the robot :



2. With our code we obtained this matrix:

[sin(q6)*(cos(q4)*sin(q1) + sin(q2 + q3)*cos(q1)*sin(q4)) + cos(q6)*(cos(q5)*(sin(q1)*sin(q4) - sin(q2 + q3)*cos(q1)*cos(q4)) - cos(q2 + q3)*cos(q1)*sin(q5)), cos(q6)*(cos(q4)*sin(q1) + sin(q2 + q3)*cos(q1)*sin(q4)) - sin(q6)*(cos(q5)*(sin(q1)*sin(q4) - sin(q2 + q3)*cos(q1)*cos(q4)) - cos(q2 + q3)*cos(q1)*sin(q5)), sin(q5)*(sin(q1)*sin(q4) - sin(q2 + q3)*cos(q1)*cos(q4)) + cos(q2 + q3)*cos(q1)*cos(q5), cos(q1)*(L2 - L4*sin(q2 + q3) + r4*cos(q2 + q3) - L3*sin(q2))]
[- sin(q6)*(cos(q1)*cos(q4) - sin(q2 + q3)*sin(q1)*sin(q4)) - cos(q6)*(cos(q5)*(cos(q1)*sin(q4) + sin(q2 + q3)*cos(q4)*sin(q1)) + cos(q2 + q3)*sin(q1)*sin(q5)), sin(q6)*(cos(q5)*(cos(q1)*sin(q4) + sin(q2 + q3)*cos(q4)*sin(q1)) + cos(q2 + q3)*sin(q1)*sin(q5)) - cos(q6)*(cos(q1)*cos(q4) - sin(q2 + q3)*sin(q1)*sin(q4)), cos(q2 + q3)*cos(q5)*sin(q1) - sin(q5)*(cos(q1)*sin(q4) + sin(q2 + q3)*cos(q4)*sin(q1)), sin(q1)*(L2 - L4*sin(q2 + q3) + r4*cos(q2 + q3) - L3*sin(q2))]
[- cos(q6)*(sin(q2 + q3)*sin(q5) - cos(q2 + q3)*cos(q4)*cos(q5)) - cos(q2 + q3)*sin(q4)*sin(q6), sin(q6)*(sin(q2 + q3)*sin(q5) - cos(q2 + q3)*cos(q4)*cos(q5)) - cos(q2 + q3)*cos(q6)*sin(q4), sin(q2 + q3)*cos(q5) + cos(q2 + q3)*cos(q4)*sin(q5), L4*cos(q2 + q3) + L3*cos(q2) + r4*sin(q2 + q3)]
[0, 0, 0, 1]

By identification, the end-effector position is :

$$\mathbf{P_x = cos(q1)*(L2 - L4*sin(q2 + q3) + r4*cos(q2 + q3) - L3*sin(q2))}$$
$$\mathbf{P_y = sin(q1)*(L2 - L4*sin(q2 + q3) + r4*cos(q2 + q3) - L3*sin(q2))}$$
$$\mathbf{P_z = L4*cos(q2 + q3) + L3*cos(q2) + r4*sin(q2 + q3)}$$

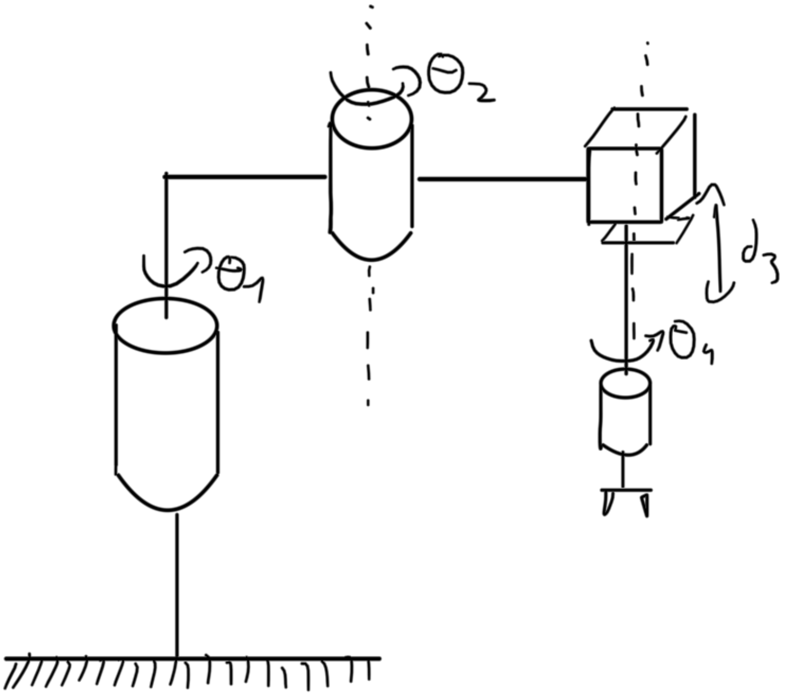
3. The Jacobian matrix of the FANUC system is :

[-cos(q1)*(L4*cos(q2 + q3) + L3*cos(q2) + r4*sin(q2 + q3)),
-cos(q1)*(L4*cos(q2 + q3) + L3*cos(q2) + r4*sin(q2 + q3)),
cos(q2 + q3)*sin(q1)*(L4*cos(q2 + q3) + r4*sin(q2 + q3)) - sin(q2 + q3)*(sin(q1)*(L2 - L4*sin(q2 + q3) + r4*cos(q2 + q3) - L3*sin(q2)) - sin(q1)*(L2 - L3*sin(q2))),
0,
0]
[-sin(q1)*(L4*cos(q2 + q3) + L3*cos(q2) + r4*sin(q2 + q3)),
-sin(q1)*(L4*cos(q2 + q3) + L3*cos(q2) + r4*sin(q2 + q3)),
sin(q2 + q3)*(cos(q1)*(L2 - L4*sin(q2 + q3) + r4*cos(q2 + q3) - L3*sin(q2)) - cos(q1)*(L2 - L3*sin(q2))) - cos(q2 + q3)*cos(q1)*(L4*cos(q2 + q3) + r4*sin(q2 + q3)),
0,
0]
[(L2 - L4*sin(q2 + q3) + r4*cos(q2 + q3) - L3*sin(q2))*cos(q1)^2 + (L2 - L4*sin(q2 + q3) + r4*cos(q2 + q3) - L3*sin(q2))*sin(q1)^2,

cos(q1)*(cos(q1)*(L2 - L4*sin(q2 + q3) + r4*cos(q2 + q3) - L3*sin(q2)) - L2*cos(q1)) +
sin(q1)*(sin(q1)*(L2 - L4*sin(q2 + q3) + r4*cos(q2 + q3) - L3*sin(q2)) - L2*sin(q1)),
cos(q2 + q3)*cos(q1)*(sin(q1)*(L2 - L4*sin(q2 + q3) + r4*cos(q2 + q3) - L3*sin(q2)) -
sin(q1)*(L2 - L3*sin(q2))) - cos(q2 + q3)*sin(q1)*(cos(q1)*(L2 - L4*sin(q2 + q3) + r4*cos(q2 +
q3) - L3*sin(q2)) - cos(q1)*(L2 - L3*sin(q2))),
0,
0]
[sin(q1),
sin(q1),
cos(q2 + q3)*cos(q1),
cos(q4)*sin(q1) + sin(q2 + q3)*cos(q1)*sin(q4), sin(q5)*(sin(q1)*sin(q4) - sin(q2 +
q3)*cos(q1)*cos(q4)) + cos(q2 + q3)*cos(q1)*cos(q5)]
[-cos(q1),
-cos(q1),
cos(q2 + q3)*sin(q1),
sin(q2 + q3)*sin(q1)*sin(q4) - cos(q1)*cos(q4),
cos(q2 + q3)*cos(q5)*sin(q1) - sin(q5)*(cos(q1)*sin(q4) + sin(q2 + q3)*cos(q4)*sin(q1))]
[0,
0,
sin(q2 + q3),
-cos(q2 + q3)*sin(q4),
sin(q2 + q3)*cos(q5) + cos(q2 + q3)*cos(q4)*sin(q5)]

IV. The SCARA robot

1.



2.

Denavit-Hartenberg table for the mechanism :

i	$\propto_{i-1}$	θ_i	d	r_{i-1}
1	0	θ_1	0	0
2	0	θ_2	L1	0
3	0	0	L2	-r3
4	0	θ_4	0	0

3. With our code we obtained this matrix:

```
TOTn =

[cos(q1 + q2 + q4), -sin(q1 + q2 + q4), 0, L2*cos(q1 + q2) + L1*cos(q1)]
[sin(q1 + q2 + q4),  cos(q1 + q2 + q4), 0, L2*sin(q1 + q2) + L1*sin(q1)]
[                0,                  0, 1,                          -r3]
[                0,                  0, 0,                            1]
```

By identification, the end-effector position is :

(P_x, P_y, P_z) = (L2*cos(q1 + q2) + L1*cos(q1), L2*sin(q1 + q2) + L1*sin(q1), -r3)

4. The Jacobian matrix of the SCARA system is :

```
J =

[- L2*sin(q1 + q2) - L1*sin(q1), -L2*sin(q1 + q2), 0]
[  L2*cos(q1 + q2) + L1*cos(q1),  L2*cos(q1 + q2), 0]
[                             0,                0, 1]
[                             0,                0, 0]
[                             0,                0, 0]
[                             1,                1, 0]
```