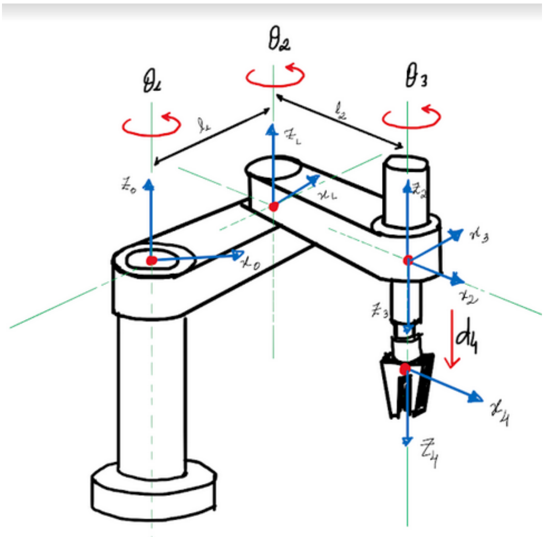
Atividade 2 de Robótica, por José Augusto dos Santos Silva

Cinemática Direta para o Robô SCARA utilizando parâmetros de Denavit-Hartenberg

Regras dos Parâmetros DH	Eixos SCARA
	Θ_i = Rotacionar o eixo X_{i-1} em torno de Z_{i-1} até ficar paralelo ao eixo X_i. d_i = Transladar o frame$_{i-1}$ ao longo de Z_{i-1} até a intersecção entre Z_{i-1} e X_i. a_i = Transladar o frame$_i$ ao longo de X_i até a intersecção entre Z_{i-1} e X_i. α_i = Rotacionar o eixo Z_{i-1} em torno de X_i até ficar paralelo ao eixo Z_i.

Constantes do problema:

- $d_{offset} = 0.1m$;
- $l_1 = 0.475m$;
- $l_2 = 0.4m$;
- $0m \leq^4 d \leq 0.1m$.

1. A tabela de parâmetros DH

```
DHRobot: MTB - Scara ROBOT, 5 joints (RRRPR), dynamics, standard DH parameters
┌──────┬─────┬───────┬────────┬─────────┬────────┐
│ θⱼ   │ dⱼ  │  aⱼ   │   αⱼ   │   q⁻    │   q⁺   │
├──────┼─────┼───────┼────────┼─────────┼────────┤
│ q1   │ 0.1 │     0 │   0.0° │ -180.0° │ 180.0° │
│ q2   │   0 │ 0.475 │   0.0° │ -180.0° │ 180.0° │
│ q3   │   0 │   0.4 │ 180.0° │ -180.0° │ 180.0° │
│ 0.0° │  q4 │     0 │ 180.0° │     0.0 │    0.1 │
│ q5   │   0 │     0 │   0.0° │ -180.0° │ 180.0° │
└──────┴─────┴───────┴────────┴─────────┴────────┘
```

No código, foi adicionado mais uma junta de revolução na origem com o parâmetro $d = d_{offset} = 0.1m$ para não ter que fazer uma translação no eixo y na fkine do DHRobot.

Também preferi re-definir z_2 para baixo, já que a junta prismática é a terceira junta do Scara no coppeliaSim. Também redefini os eixos de z_3 e z_4 na ponta efetuador, por isso foi necessário as duas rotações de π em α_2 e α_3.

2. Cálculo das matrizes de transformação até o efetuador

Seja

$$T_i^{i-1} = \begin{bmatrix} cos(\theta_i) & -sin(\theta_i)cos(\alpha_i) & sin(\theta_i)sin(\alpha_i) & a_i cos(\theta_i) \\ sin(\theta_i) & cos(\theta_i)cos(\alpha_i) & -cos(\theta_i)sin(\alpha_i) & a_i sin(\theta_i) \\ 0 & sin(\alpha_i) & cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

a matriz de transformação homogênia que rotaciona em z, translação em z, translação em x e rotação em x entre os frames i e $i-1$ a partir dos parâmetros DH estabelecidos. Segue abaixo as matrizes de transformação homogênia para cada par sequencial de referências. Observe que, pelo fato de ter que colocar uma junta na origem, com nenhum parâmetro variável apenas com $d = d_{offset}$, teremos, excepcionalmente, uma matriz T_0 para fazer este ajuste.

$$T_0 = \begin{bmatrix} cos(\theta_0) & -sin(\theta_0)cos(\alpha_0) & sin(\theta_0)sin(\alpha_0) & a_i cos(\theta_0) \\ sin(\theta_0) & cos(\theta_0)cos(\alpha_0) & -cos(\theta_0)sin(\alpha_0) & a_i sin(\theta_0) \\ 0 & sin(\alpha_0) & cos(\alpha_0) & d_{offset} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_1^0 = \begin{bmatrix} cos(\theta_1) & -sin(\theta_1) & 0 & l_1 cos(\theta_1) \\ sin(\theta_1) & cos(\theta_1) & 0 & l_1 sin(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} cos(\theta_1) & -sin(\theta_1) & 0 & 0.475cos(\theta_1) \\ sin(\theta_1) & cos(\theta_1) & 0 & 0.475sin(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^1 = \begin{bmatrix} cos(\theta_2) & sin(\theta_2) & 0 & l_2 cos(\theta_2) \\ sin(\theta_2) & -cos(\theta_2) & 0 & l_2 sin(\theta_2) \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} cos(\theta_2) & sin(\theta_2) & 0 & 0.4cos(\theta_2) \\ sin(\theta_2) & -cos(\theta_2) & 0 & 0.4sin(\theta_2) \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

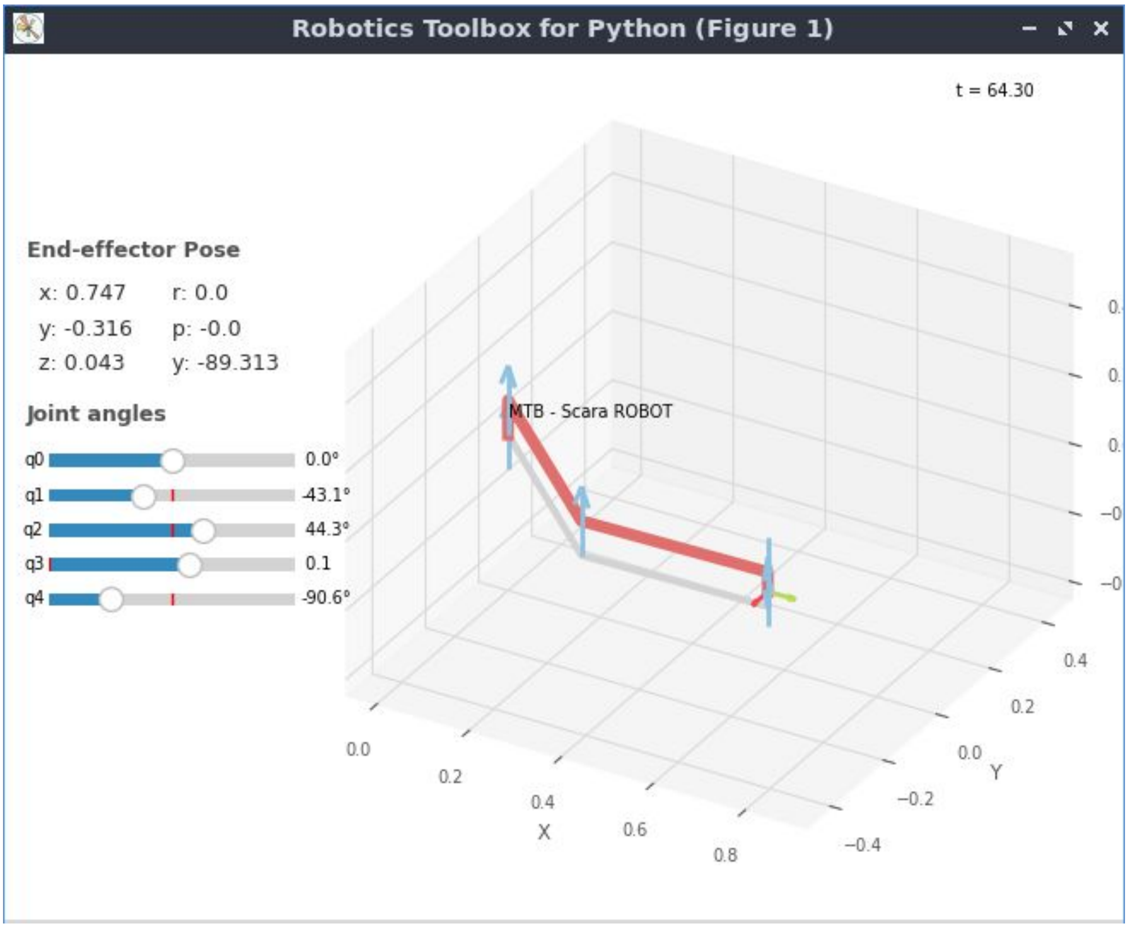
$$T_3^2 = \begin{bmatrix} cos(\theta_3) & -sin(\theta_3)cos(\alpha_3) & sin(\theta_3)sin(\alpha_3) & a_3 cos(\theta_3) \\ sin(\theta_3) & cos(\theta_3)cos(\alpha_3) & -cos(\theta_3)sin(\alpha_3) & a_3 sin(\theta_3) \\ 0 & sin(\alpha_3) & cos(\alpha_3) & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & {}^4d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4^3 = \begin{bmatrix} cos(\theta_4) & -sin(\theta_4)cos(\alpha_4) & sin(\theta_4)sin(\alpha_4) & a_4 cos(\theta_4) \\ sin(\theta_4) & cos(\theta_4)cos(\alpha_4) & -cos(\theta_4)sin(\alpha_4) & a_4 sin(\theta_4) \\ 0 & sin(\alpha_4) & cos(\alpha_4) & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} cos(\theta_3) & -sin(\theta_3) & 0 & 0 \\ sin(\theta_3) & cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. Código da função de cinemática direta fkine

```python
def fkine_diy(qr):
    theta1, theta2, d, theta3 = qr

    costheta1, sintheta1 = np.cos(theta1), np.sin(theta1)
    costheta2, sintheta2 = np.cos(theta2), np.sin(theta2)
    costheta3, sintheta3 = np.cos(theta3), np.sin(theta3)

    T0 = np.array([
        [1, 0, 0,   0],
        [0, 1, 0,   0],
        [0, 0, 1,  do],
        [0, 0, 0,   1],
    ])

    T1 = np.array([
        [costheta1, -sintheta1, 0,   l1*costheta1],
        [sintheta1,  costheta1, 0,   l1*sintheta1],
        [0, 0, 1, 0],
        [0, 0, 0, 1],
    ])

    T2 = np.array([
        [costheta2, sintheta2, 0, l2*costheta2],
        [sintheta2, -costheta2, 0, l2*sintheta2],
        [0, 0, -1, 0],
        [0, 0, 0, 1]
    ])

    T3 = np.array([
        [1,  0,  0, 0],
        [0, -1,  0, 0],
        [0,  0, -1, d],
        [0,  0,  0, 1],
    ])

    T4 = np.array([
        [costheta3, -sintheta3, 0, 0],
        [sintheta3, costheta3, 0, 0],
        [0, 0, 1, 0],
        [0, 0, 0, 1],
    ])

    return functools.reduce(np.matmul, [T0, T1, T2, T3, T4])
```

4. Print do objeto SerialLink gerado na robotics toolbox



5. Comparação do resultado da função fkine implementada e a função fkine dentro do objeto SerialLink gerado para as seguintes configurações:

   1. $\theta_1 = 0$; $\theta_2 = 0$; $\theta_3 = 0$; ${}^4d = 0$

```
Argumentos [theta1, theta2, d, theta3] = [0, 0, 0, 0]

fkine (José Augusto):
 [[1.    0.    0.    0.875]
 [0.    1.    0.    0.   ]
 [0.    0.    1.    0.1  ]
 [0.    0.    0.    1.   ]]

fkine (DHRobot):
    1         0         0         0.875
   0         1         0         0
   0         0         1         0.1
   0         0         0         1
```

   2. $\theta_1 = \frac{\pi}{2}$; $\theta_2 = -\frac{\pi}{2}$; $\theta_3 = 0$; ${}^4d = 0$

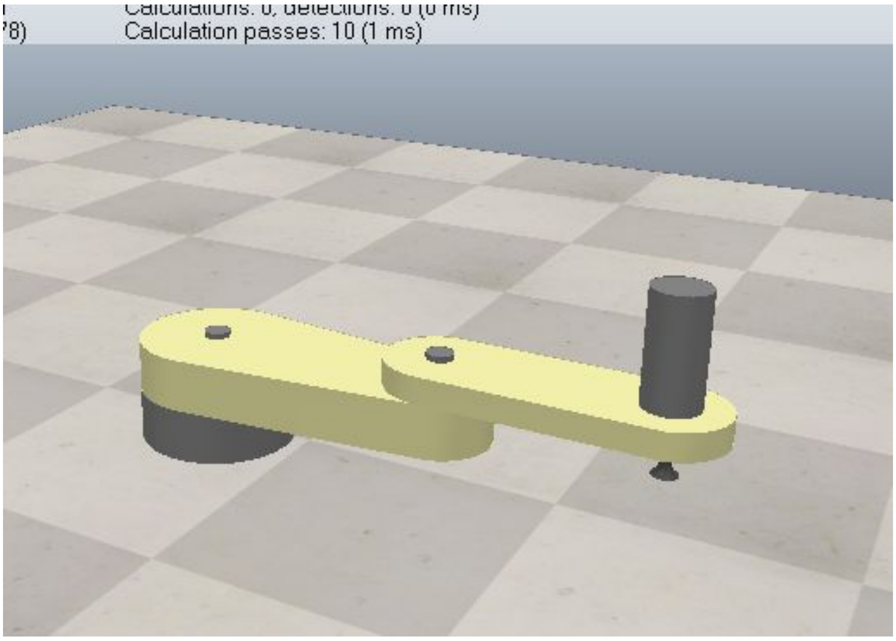
```
Argumentos [theta1, theta2, d, theta3] = [1.5707963267948966, -1.5707963267948966, 0, 0]

fkine (José Augusto):
 [[1.    0.    0.    0.4  ]
 [0.    1.    0.    0.475]
 [0.    0.    1.    0.1  ]
 [0.    0.    0.    1.   ]]

fkine (DHRobot):
    1         0         0         0.4
   0         1         0         0.475
   0         0         1         0.1
   0         0         0         1
```

   3. $\theta_1 = \frac{\pi}{2}$; $\theta_2 = -\frac{\pi}{2}$; $\theta_3 = 0$; ${}^4d = 0.05$

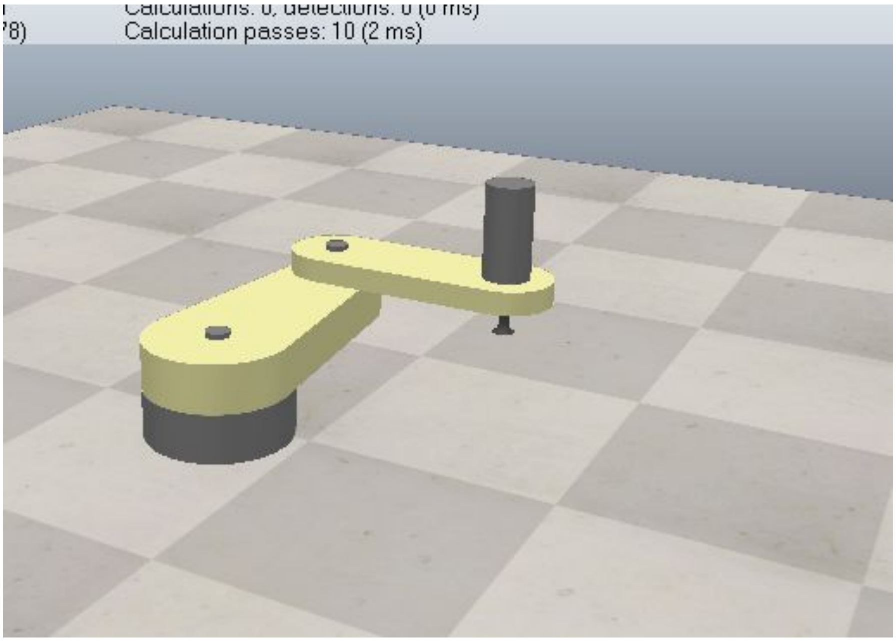
```
Argumentos [theta1, theta2, d, theta3] = [1.5707963267948966, -1.5707963267948966, 0.05, 0]

fkine (José Augusto):
 [[1.    0.    0.    0.4  ]
 [0.    1.    0.    0.475]
 [0.    0.    1.    0.05 ]
 [0.    0.    0.    1.   ]]

fkine (DHRobot):
    1         0         0         0.4
   0         1         0         0.475
   0         0         1         0.05
   0         0         0         1
```

   4. Enviar comandos para o CoppeliaSIM

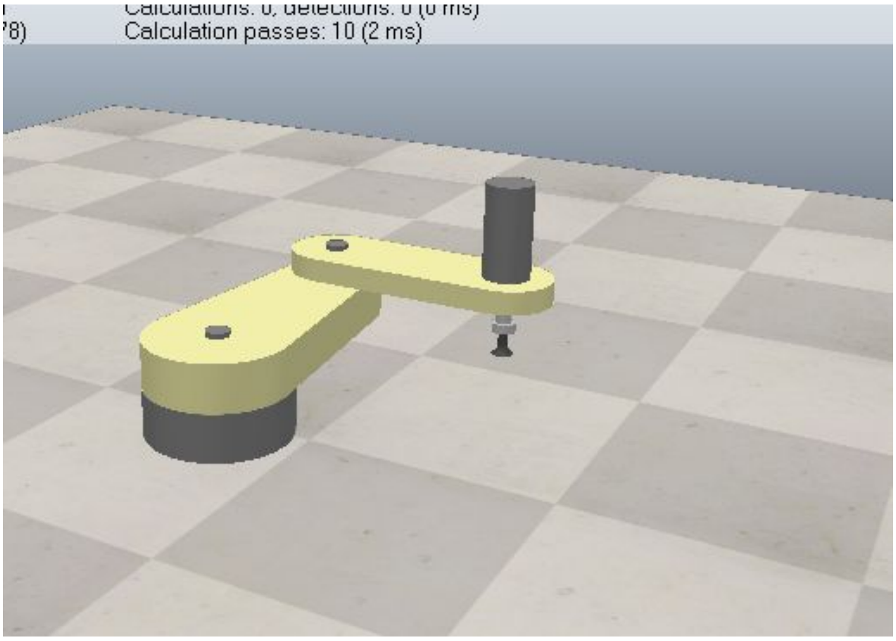
      Primeiro comando $\theta_1 = 0$; $\theta_2 = 0$; $\theta_3 = 0$; ${}^4d = 0$:



Segundo comando $\theta_1 = \frac{\pi}{2}$; $\theta_2 = -\frac{\pi}{2}$; $\theta_3 = 0$; ${}^4d = 0$:



Terceiro comando $\theta_1 = \frac{\pi}{2}$; $\theta_2 = -\frac{\pi}{2}$; $\theta_3 = 0$; ${}^4d = 0.05$:


Calculations: 0, detections: 0 (0 ms)
Calculation passes: 10 (2 ms)