

UNIVERSIDAD CATÓLICA BOLIVIANA “SAN PABLO”

TRABAJO PRÁCTICO “Cinemática Inversa DH de un Robot SCARA”



Robótica

Ing. Marcelo Saavedra

José David Valda Peñaranda

29/08/2023

Resumen:

La cinemática inversa es una parte fundamental en la robótica, ya que permite determinar las configuraciones articulares necesarias para lograr una posición y orientación específica del extremo final de un robot. En este laboratorio, se llevó a cabo un análisis de la cinemática inversa utilizando la convención Denavit-Hartenberg (DH) en un robot de tipo SCARA.

El robot SCARA (Selective Compliance Assembly Robot Arm) es un tipo de robot industrial que se caracteriza por su capacidad para moverse en el plano horizontal y realizar operaciones de ensamblaje.

Objetivos:

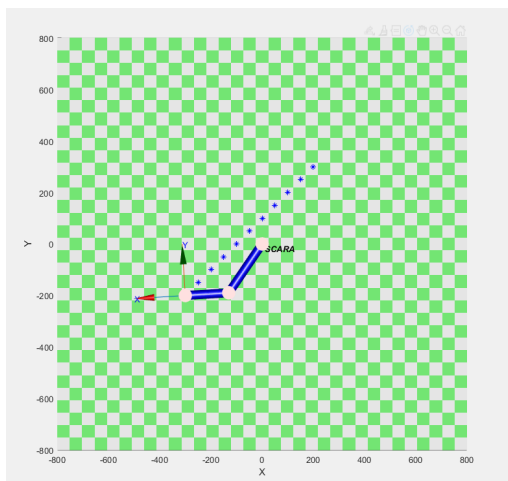
- Comprender los conceptos básicos de la cinemática inversa en robots.
- Aplicar la convención Denavit-Hartenberg (DH) para describir la geometría del robot SCARA.
- Determinar las configuraciones articulares para lograr una posición deseada del extremo final del robot.
- Lograr una trayectoria recta con el robot scara

Procedimiento:

Descripción DH del Robot: Se realizó una descripción detallada de la estructura del robot SCARA utilizando la convención Denavit-Hartenberg. Se identificaron los parámetros DH para cada articulación, incluyendo las longitudes de enlace, ángulos de orientación y desplazamientos.

```
L0 = Link([0 0 230 0]);  
L1 = Link([0 0 170 pi]);  
L2 = Link([0 50 0 0]);  
L3 = Link([0 20 0 0]);  
SCARA=SerialLink([L0 L1 L2 L3], 'name', 'SCARA')
```

Planteamiento del Problema: Se seleccionó una posición y orientación específica en el espacio de trabajo del robot como objetivo. Se ve una trayectoria recta



Cálculo de Cinemática Inversa: Utilizando los parámetros DH y la matriz de transformación homogénea objetivo, se realizó el cálculo de la cinemática inversa. Se determinaron las configuraciones articulares requeridas para lograr la posición y orientación deseada del extremo final. A continuación se muestran las posiciones a ser dirigido el robot.

```
f = [1,0,0,200;
     0,1,0, 300;
     0,0,1, 0;
     0,0,0, 1];

f1 = [1,0,0,150;
      0,1,0, 250;
      0,0,1, 0;
      0,0,0, 1];

f2 = [1,0,0,100;
      0,1,0, 200;
      0,0,1, 0;
      0,0,0, 1];

f3 = [1,0,0,50;
      0,1,0, 150;
      0,0,1, 0;
      0,0,0, 1];

f4 = [1,0,0,0;
      0,1,0, 100;
      0,0,1, 0;
      0,0,0, 1];

f5 = [1,0,0,-50;
      0,1,0, 50;
      0,0,1, 0;
      0,0,0, 1];

f6 = [1,0,0,-100;
      0,1,0, -1;
      0,0,1, 0;
      0,0,0, 1];

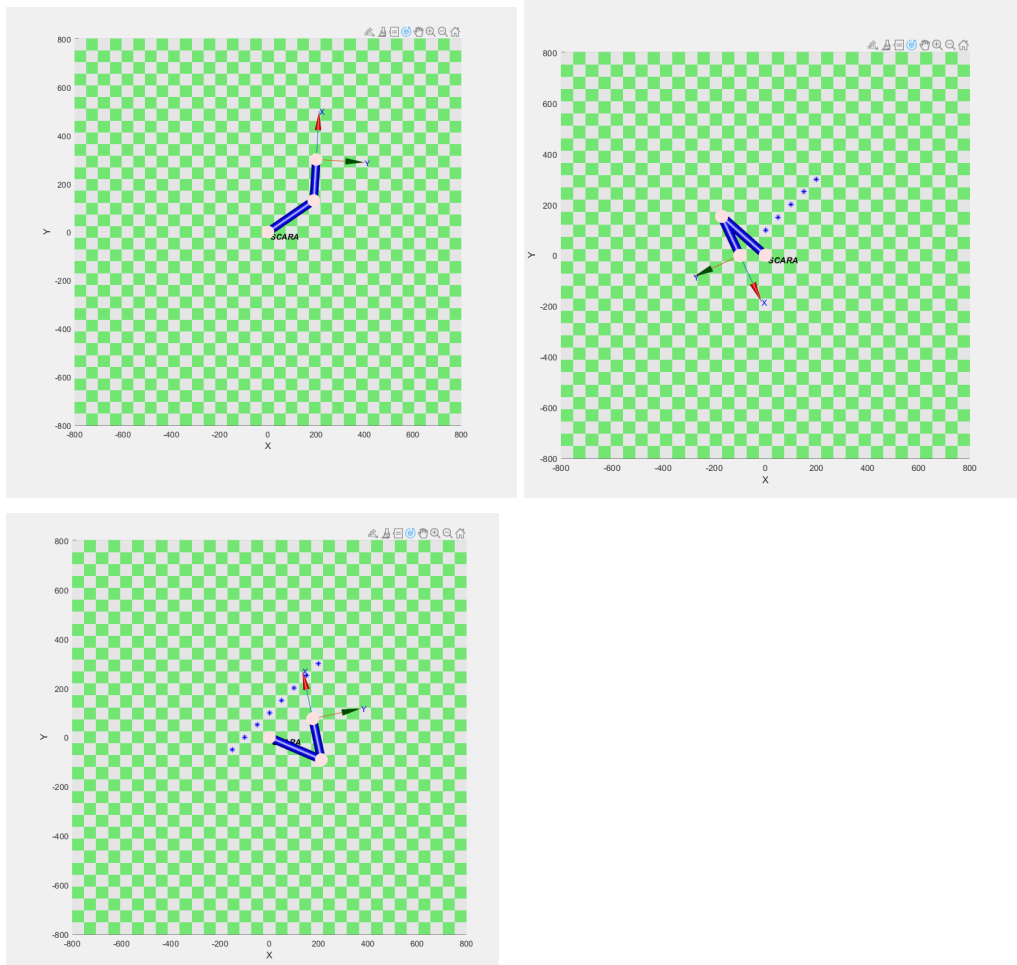
f7 = [1,0,0,-150;
      0,1,0, -50;
      0,0,1, 0;
      0,0,0, 1];

f8 = [1,0,0,-200;
      0,1,0, -100;
      0,0,1, 0;
      0,0,0, 1];

f9 = [1,0,0,-250;
      0,1,0, -150;
      0,0,1, 0;
      0,0,0, 1];

f10 = [1,0,0,-300;
       0,1,0, -200;
       0,0,1, 0;
       0,0,0, 1];
```

Simulación o Control del Robot: Se utilizó Matlab para verificar la validez de las configuraciones articulares calculadas. Se ingresaron los valores obtenidos en el paso anterior y se observó si el robot alcanzaba la posición y orientación deseada. A continuación se muestra el conjunto de imágenes donde se puede ver al robot moverse por la trayectoria:



Los resultados obtenidos demostraron que las configuraciones articulares calculadas utilizando la cinemática inversa DH permitieron que el robot SCARA alcanzara la posición y orientación deseada en el espacio de trabajo. Se observó una concordancia satisfactoria entre las configuraciones simuladas.

ANEXOS:

Código:

```
clc; clear all,close all;
%----- ROBOT: SCARA -----
%%%%%%%%%%%%%% CREACION DEL ROBOT EN MATLAB
%%%%%%%%%%%%%%
L0 = Link([0 0 230 0]);
L1 = Link([0 0 170 pi]);
L2 = Link([0 50 0 0]);
L3 = Link([0 20 0 0]);
```

```

SCARA=SerialLink([L0 L1 L2 L3], 'name', 'SCARA')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% INVERSA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

SCARA.plotopt = {'workspace', [-800,800,-800,800,-200,800]};
figure(1)
%SCARA.plot([0,0, 0, 100])
%figure(2)
SCARA.plot([0,0, -200, -200]);

view([0,0,90])

%Posicion deseada
f = [1,0,0,200;
     0,1,0, 300;
     0,0,1, 0;
     0,0,0, 1];
%ingreso de coordenadas
%Q=SCARA.ikine(f,[0,0,0,0], [1,1,1,1,0,0], 'pinv')
Q = SCARA.ikine(f, 'q0', [0,0,0,0], 'mask',[1, 1, 0, 0, 0, 0])
%definimos workspace

t = [0:1:20]';
QQ=jtraj([0 0 0 0],Q,t)
%figure(2)
SCARA.plot(QQ) %sumar los dos primeros ángulos
hold on; plot(f(1,4),f(2,4),'*b');
%%%%%%%% Ahora mover a otra posicion

f1 = [1,0,0,150;
      0,1,0, 250;
      0,0,1, 0;
      0,0,0, 1];
%Q1=SCARA.ikine(f1,Q, [1,1,1,1,0,0], 'pinv')
Q1 = SCARA.ikine(f1, 'q1', Q, 'mask',[1, 1, 0, 0, 0, 0])

QQ1=jtraj(Q,Q1,t)
SCARA.plot(QQ1) %sumar los dos primeros ángulos
hold on; plot(f1(1,4),f1(2,4),'*b');

f2 = [1,0,0,100;
      0,1,0, 200;
      0,0,1, 0;
      0,0,0, 1];
%Q1=SCARA.ikine(f1,Q, [1,1,1,1,0,0], 'pinv')
Q2 = SCARA.ikine(f2, 'q1', Q1, 'mask',[1, 1, 0, 0, 0, 0])

QQ2=jtraj(Q1,Q2,t)
SCARA.plot(QQ2) %sumar los dos primeros ángulos
hold on; plot(f2(1,4),f2(2,4),'*b');

```

```

f3 = [1,0,0,50;
      0,1,0, 150;
      0,0,1, 0;
      0,0,0, 1];
%Q1=SCARA.ikine(f1,Q, [1,1,1,1,0,0], 'pinv')
Q3 = SCARA.ikine(f3, 'q1', Q2, 'mask',[1, 1, 0, 0, 0, 0])

QQ3=jtraj(Q2,Q3,t)
SCARA.plot(QQ3) %sumar los dos primeros ángulos
hold on; plot(f3(1,4),f3(2,4),'*b');

f4 = [1,0,0,0;
      0,1,0, 100;
      0,0,1, 0;
      0,0,0, 1];
%Q1=SCARA.ikine(f1,Q, [1,1,1,1,0,0], 'pinv')
Q4 = SCARA.ikine(f4, 'q1', Q3, 'mask',[1, 1, 0, 0, 0, 0])

QQ4=jtraj(Q3,Q4,t)
SCARA.plot(QQ4) %sumar los dos primeros ángulos
hold on; plot(f4(1,4),f4(2,4),'*b');

f5 = [1,0,0,-50;
      0,1,0, 50;
      0,0,1, 0;
      0,0,0, 1];
%Q1=SCARA.ikine(f1,Q, [1,1,1,1,0,0], 'pinv')
Q5 = SCARA.ikine(f5, 'q1', Q4, 'mask',[1, 1, 0, 0, 0, 0])

QQ5=jtraj(Q4,Q5,t)
SCARA.plot(QQ5) %sumar los dos primeros ángulos
hold on; plot(f5(1,4),f5(2,4),'*b');

f6 = [1,0,0,-100;
      0,1,0, -1;
      0,0,1, 0;
      0,0,0, 1];
%Q1=SCARA.ikine(f1,Q, [1,1,1,1,0,0], 'pinv')
Q6 = SCARA.ikine(f6, 'q1', Q5, 'mask',[1, 1, 0, 0, 0, 0])

QQ6=jtraj(Q5,Q6,t)
SCARA.plot(QQ6) %sumar los dos primeros ángulos
hold on; plot(f6(1,4),f6(2,4),'*b');

f7 = [1,0,0,-150;
      0,1,0, -50;
      0,0,1, 0;
      0,0,0, 1];
%Q1=SCARA.ikine(f1,Q, [1,1,1,1,0,0], 'pinv')
Q7 = SCARA.ikine(f7, 'q1', Q6, 'mask',[1, 1, 0, 0, 0, 0])

```

```

QQ7=jtraj(Q6,Q7,t)
SCARA.plot(QQ7) %sumar los dos primeros ángulos
hold on; plot(f7(1,4),f7(2,4),'*b');

f8 = [1,0,0,-200;
      0,1,0, -100;
      0,0,1, 0;
      0,0,0, 1];
%Q1=SCARA.ikine(f1,Q, [1,1,1,1,0,0], 'pinv')
Q8 = SCARA.ikine(f8, 'q1', Q7, 'mask',[1, 1, 0, 0, 0, 0])

QQ8=jtraj(Q7,Q8,t)
SCARA.plot(QQ8) %sumar los dos primeros ángulos
hold on; plot(f8(1,4),f8(2,4),'*b');

f9 = [1,0,0,-250;
      0,1,0, -150;
      0,0,1, 0;
      0,0,0, 1];
%Q1=SCARA.ikine(f1,Q, [1,1,1,1,0,0], 'pinv')
Q9 = SCARA.ikine(f9, 'q1', Q8, 'mask',[1, 1, 0, 0, 0, 0])

QQ9=jtraj(Q8,Q9,t)
SCARA.plot(QQ9) %sumar los dos primeros ángulos
hold on; plot(f9(1,4),f9(2,4),'*b');

f10 = [1,0,0,-300;
       0,1,0, -200;
       0,0,1, 0;
       0,0,0, 1];
%Q1=SCARA.ikine(f1,Q, [1,1,1,1,0,0], 'pinv')
Q10 = SCARA.ikine(f10, 'q1', Q9, 'mask',[1, 1, 0, 0, 0, 0])

QQ10=jtraj(Q9,Q10,t)
SCARA.plot(QQ10) %sumar los dos primeros ángulos
hold on; plot(f10(1,4),f10(2,4),'*b');

```