

UNIVERSIDAD
NACIONAL
DE COLOMBIA

UNIVERSIDAD NACIONAL DE COLOMBIA
ROBÓTICA
2021 - 2S

Laboratorio 2

Presentado a:
Ing. Pedro Cárdenas

Presentado por:
Daniel Alejandro Delgado A.
Felipe Rojas Arredondo
Brian Camilo Valencia Peña

Índice

| | |
|--|-----------|
| Índice de figuras | 2 |
| 1. Modelo Geométrico Inverso | 1 |
| 1.1. Determine el modelo geométrico inverso del robot asignado haciendo uso de la metodología explicada en clase. Haga una descripción detallada del proceso haciendo uso de imágenes y dibujos que ayuden visualizar de dónde provienen las ecuaciones encontradas. | 1 |
| 2. Multiplicidad de Soluciones | 8 |
| 3. Cinemática inversa mediante RVC | 8 |
| 4. Cinemática inversa mediante RST | 9 |
| 5. Comparación entre los métodos RVC y RST para cinemática inversa | 12 |
| 6. Compruebe mediante su modelo geométrico inverso la configuración del robot para las posturas de la herramienta halladas en el anterior laboratorio, tales como puntos de calibración | 12 |
| 7. Posturas propuestas | 12 |
| 8. Verificación de los resultados empleando el GUI | 12 |
| 9. Actualice la GUI para que tenga la opción de ingresar la posición de la herramienta | 15 |
| Referencias | 15 |

Índice de figuras

| | |
|--|----|
| 1. Robot IRB 14050 | 1 |
| 2. Ejes de las articulaciones | 1 |
| 3. Posición final del efector con respecto a la base del robot | 2 |
| 4. modelo geométrico de robot con 4 articulaciones | 3 |
| 5. Vista lateral | 3 |
| 6. Parámetros de DH modificados | 5 |
| 7. Dimensiones del robot IRB 14050 | 7 |
| 8. Configuración objetivo | 10 |
| 9. Vector de la configuración objetivo | 11 |
| 10. Configuración solución | 11 |
| 11. Vector de la configuración solución | 11 |
| 12. Información de la configuración solución | 11 |
| 13. Primera pose | 13 |
| 14. Posición de HOME | 14 |
| 15. Posición de HOME | 14 |
| 16. Posición de HOME | 15 |
| 17. interfaz gráfica | 16 |

1. Modelo Geométrico Inverso

- 1.1. Determine el modelo geométrico inverso del robot asignado haciendo uso de la metodología explicada en clase. Haga una descripción detallada del proceso haciendo uso de imágenes y dibujos que ayuden visualizar de dónde provienen las ecuaciones encontradas.

Para iniciar con el análisis de cinemática inversa es necesario establecer algunas características del robot que se mencionan en el informe de laboratorio 1 – Análisis de las características de un robot industrial y modelo geométrico directo. El robot analizado es el ABB IRB 14050 tal como se muestra en la siguiente figura. La principal característica de este robot es que presenta 7 grados de libertad que le dan una gran versatilidad para ser empleado en la industria electrónica permitiéndole ser un robot redundante por lo que para llegar a una pose establecida se pueden generar múltiples soluciones.



Figura 1: Robot IRB 14050

Al analizar la construcción del robot mediante cinemática directa, se evidencia la alta complejidad para construir un modelo de cinemática inversa dado que se tiene la limitante que los 3 ejes de las 3 últimas articulaciones no se intersectan, sin embargo, dado que se tiene un robot redundante y es posible afirmar que van a existir múltiples soluciones, para simplificar el análisis se asumirá que en todas las posiciones analizadas la articulación 6 no rotará y que su eje está alineado con los ejes de las articulaciones 6 y 7 para así poder aplicar el método de desacople cinemático.

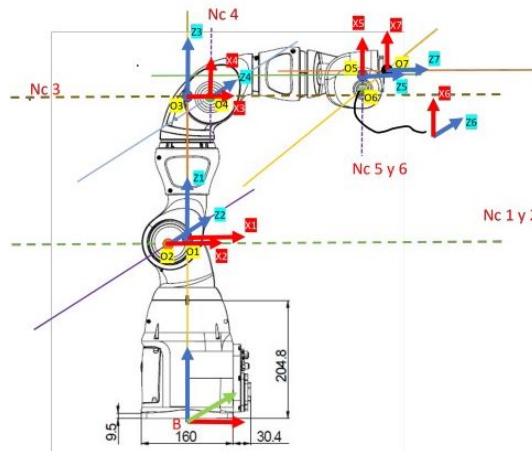


Figura 2: Ejes de las articulaciones

Es posible determinar que en la articulación 5 se genera una muñeca cinemática dado que el movimiento de las últimas 3 articulaciones no genera movimiento de este punto, lo que el análisis de cinemática indirecta se puede realizar a través del método de desacoplo cinemático. Sin embargo, cabe recalcar que los 3 últimos ejes no se intersectan por lo que es necesario tomar tener algunas consideraciones especiales a la hora de realizar el análisis. En primer lugar, es necesario establecer la ubicación exacta de la muñeca cinemática, esto se realiza empleando la pose deseada del efector final con respecto al marco de referencia que es la base del robot la cual se describe mediante una rotación y una translación empleando una matriz homogénea de transformación.

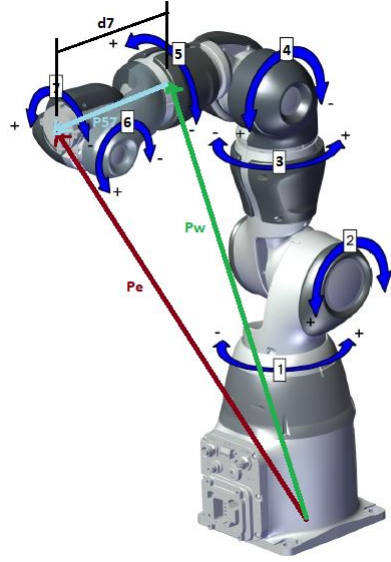


Figura 3: Posición final del efector con respecto a la base del robot

Analizando la anterior imagen es posible establecer que la posición del efector final corresponde a la sumatoria entre la posición de la muñeca y distancia MHT que hay entre la articulación 5 y la 7.

$$P_e = P_w + P_{57} \quad (1)$$

De la anterior ecuación es posible establecer que la posición existente entre la articulación 5 y 7 depende de la distancia $d7$ y una rotación de los ejes ($a_{T_{0e}}$) que dependerá directamente de la posición de q_5 , q_6 y q_7 , de esta manera es posible remplazar la MHT P_{57} por:

$$P_{57} = d7 \cdot a_{T_{0e}} \quad (2)$$

Sin embargo, como se tiene una muñeca cinemática se conoce la rotación entre el eje 5 y 7, la distancia $d7$ y la pose del efector final por lo que es posible despejar la pose de la muñeca cinemática

$$P_e = P_w + d6 \cdot a_{T_{0e}} \quad (3)$$

de esta manera se obtiene la posición de la muñeca estableciendo que cada uno de los anteriores valores es un vector por lo que se obtiene:

$$\begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = \begin{bmatrix} x_{ef} \\ y_{ef} \\ z_{ef} \end{bmatrix} - d7 \begin{bmatrix} r_{xef} \\ r_{yef} \\ r_{zef} \end{bmatrix} \quad (4)$$

La implementación de lo anterior en Matlab se muestra a continuación:

```

1 %%POSICION DE LA MUÑECA
2 syms e_rx e_ry e_rz e_x e_y e_z d7 %variables simbolicas
3
4 %se establece la matriz de transformacion homogenea que
5 %determina la posicion y orientacion del efector final
6 Tef= trotx(e_rz)*troty(e_ry)*trotx(e_rx)*transl(e_x,e_y,e_z);
7
8 %se establecen las coordenadas (x,y,z) de la muñeca
9 x_w=vpa(Tef(1,4)-d7*Tef(1,3),4);
10 y_w=vpa(Tef(2,4)-d7*Tef(2,3),4);
11 z_w=vpa(Tef(3,4)-d7*Tef(3,3),4);

```

Después de obtener la posición exacta de la muñeca, con la posición (x,y,z) encontrada anteriormente se procede a realizar cinemática inversa mediante el método geométrico para las primeras 4 articulaciones para así poder determinar q_1 , q_2 , q_3 y q_4 . Para esto, se estableció el siguiente modelo geométrico:

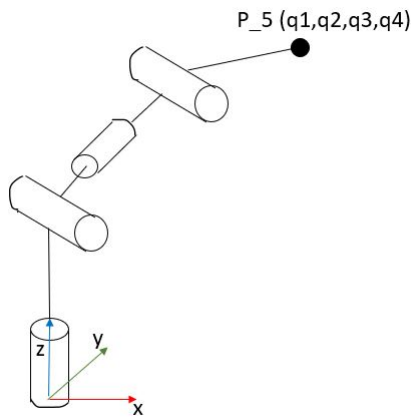


Figura 4: modelo geométrico de robot con 4 articulaciones

Para realizar el análisis se establece la vista superior y lateral del robot que permiten realizar con mayor claridad el análisis geométrico.

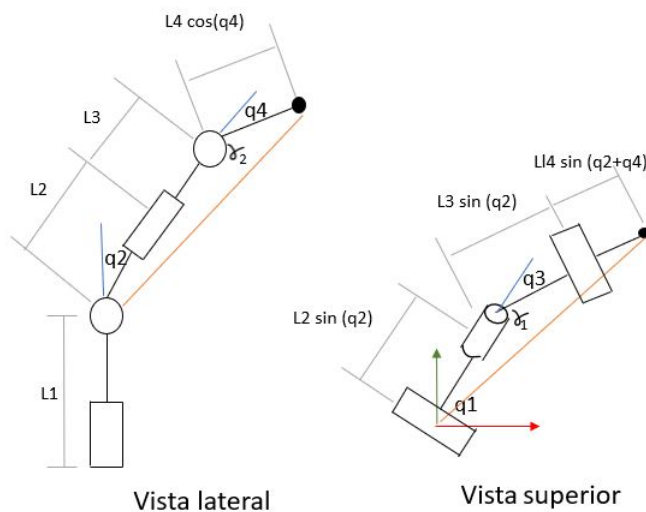


Figura 5: Vista lateral

De la anterior imagen es posible establecer una relación entre cada uno de las articulaciones para así determinar q_1 , q_2 , q_3 y q_4 . En primer lugar se establece q_4 determinando que:

$$q_4 = 180 - \gamma_2 \quad (5)$$

$$\sqrt{(z - L_1)^2 + (x^2 + y^2)}^2 = (L_4 \cos(q_4))^2 + (L_2 + L_3)^2 - 2(L_4 \cos(q_4))(L_2 + L_3) \cos(\gamma_2) \quad (6)$$

Remplazando la ecuación 5 en la 6 se obtiene:

$$\sqrt{(z - L_1)^2 + (x^2 + y^2)}^2 = (L_4 \cos(q_4))^2 + (L_2 + L_3)^2 + 2(L_4 \cos(q_4))(L_2 + L_3) \cos(q_4) \quad (7)$$

$$\cos(q_4) = \sqrt{-\frac{-L_1^2 + 2 * L_1 * z + L_2^2 + 2 * L_2 * L_3 + L_3^2 - x^2 - y^2 - z^2}{L_4 * (2 * L_2 + 2 * L_3 + L_4)}} = D \quad (8)$$

Mediante razones trigonométricas se establece que:

$$\sin^2(q_4) = 1 - \cos^2(q_4) \quad (9)$$

$$\sin(q_4) = \sqrt{1 - D^2} \quad (10)$$

Tomando las anteriores ecuaciones se determina que q_4 es:

$$q_4 = \tan^{-1} \left(\frac{\sqrt{1 - D^2}}{D} \right) \quad (11)$$

Continuando con q_2 se determina la siguiente relación:

$$q_2 = \frac{\pi}{2} - \left(\tan^{-1} \left(\frac{z - L_1}{x^2 + y^2} \right) + \tan^{-1} \left(\frac{L_4 \cos(q_4) \cdot \sin(q_4)}{L_2 + L_3 + L_4 \cos^2(q_4)} \right) \right) \quad (12)$$

Para las otras articulaciones se establecieron las siguientes relaciones:

$$\sqrt{x^2 + y^2}^2 = (L_3 \sin(q_2 + q_4) + L_3 \sin(q_2))^2 + (L_2 \sin(q_2))^2 - 2(L_3 \sin(q_2 + q_4) + L_3 \sin(q_2))(L_2 \sin(q_2)) \cos(\gamma_1) \quad (13)$$

Al establecer que:

$$q_3 = 180 - \gamma_1 \quad (14)$$

se establece la siguiente relación:

$$\sqrt{x^2 + y^2}^2 = (L_3 \sin(q_2 + q_4) + L_3 \sin(q_2))^2 + (L_2 \sin(q_2))^2 + 2(L_3 \sin(q_2 + q_4) + L_3 \sin(q_2))(L_2 \sin(q_2)) \cos(q_3) \quad (15)$$

$$\cos(q_3) = \frac{x^2 + y^2 - (L_3 \sin(q_2 + q_4) + L_3 \sin(q_2))^2 - (L_2 \sin(q_2))^2}{2(L_3 \sin(q_2 + q_4) + L_3 \sin(q_2))(L_2 \sin(q_2))} = D_1 \quad (16)$$

Utilizando las razones trigonométricas:

$$\sin^2(q_3) = 1 - \cos^2(q_3) \quad (17)$$

$$\sin(q_3) = \sqrt{1 - D_1^2} \quad (18)$$

Mediante el anterior análisis se establece que el valor de q_3 se define de la siguiente manera:

$$q_3 = \tan^{-1} \left(\frac{\sqrt{1 - D_1^2}}{D_1} \right) \quad (19)$$

Por ultimo, falta determinar el valor de q1.

$$q_1 = \tan^{-1} \left(\frac{y}{x} \right) + \tan^{-1} \left(\frac{(L_3 \sin(q_2) + L_4 \sin(q_2 + q_4)) \cdot \sin(q_3)}{L_2 \sin(q_2) + (L_3 \sin(q_2) + L_4 \sin(q_2 + q_4)) \cdot \cos(q_3)} \right) \quad (20)$$

Una vez obtenidos estos valores se realiza la implementación en Matlab.

```

1 %% CINEMATICA INVERSA ARTICULACIONES 1, 2, 3 Y 4
2
3 %declaracion de variables simbolicas
4 syms L1 L2 L3 L4 q4
5
6 %se determina el valor de q4
7 co_q4=cos(solve((z_w-L1)^2+x_w^2+y_w^2==(L4*cos(q4))^2+(L2+L3)^2+2*(L4*cos(q4))*(L2
8 +L3)*cos(q4),q4));
9 co_q4=co_q4(1,1);
10 si_q4=sqrt(1-co_q4^2);
11 q4= atan2(si_q4,co_q4);
12
13 %se determina el valor de q2
14 q2=pi/2-(atan2(z_w-L1,x_w^2+y_w^2)+atan2(L4*cos(q4)*sin(q4),L2+L3+L4*cos(q4)^2));
15
16 %se determina el valode de q3
17 co_q3=(x_w^2+y_w^2-(L3*sin(q2+q4)+L3*sin(q2))^2-(L2*sin(q2))^2)/(2*(L3*sin(q2+q4)+
18 L3*sin(q2))*(L2*sin(q2)));
19 si_q3=sqrt(1-co_q3^2);
20 q3=atan2(si_q3,co_q3);
21
22 %se determina el valor de q1
23 q1=atan2(y_w,x_w)+atan2((L3*sin(q2)+L4*sin(q2+q4))*sin(q3),L2*sin(q2)+(L3*sin(q2)+
24 L4*sin(q2+q4))*cos(q3));

```

Continuando con el método de l desacople cinemático, ahora es necesario determinar la matriz de rotaciones que relacionan los ejes de la articulación 4 a la articulación 7 para así poder determinar los valores de q5, q6 y q7. Para esto es necesario resolver la cinemática directa del robot. Empleando la figura 2 se establecen los parámetros de DH modificado los cuales son:

| i | $\alpha_{(i-1)}$ | $a_{(i-1)}$ | d_i | θ_i | Offset |
|---|------------------|-------------|--------|------------|--------|
| 1 | 0 | 0 | 305.83 | Q1 | 0 |
| 2 | -90 | -30 | 0 | Q2 | 0 |
| 3 | 90 | 30 | 251.52 | Q3 | 0 |
| 4 | -90 | 38.51 | 0 | Q4 | -90 |
| 5 | -90 | 42.37 | 267.01 | Q5 | 0 |
| 6 | 90 | -28.8 | 0 | Q6 | 0 |
| 7 | -90 | 27.24 | 35.85 | Q7 | 0 |

Figura 6: Parámetros de DH modificados

```

1 %% CINEMATICA DIRECTA
2 syms q5 q6 q7
3
4 L(1) = Link('revolute','alpha', 0, 'a', 0, 'd', 305.83, 'offset', 0, '
modified', 'qlim', [-168.5*pi/180 168.5*pi/180]);
5 L(2) = Link('revolute','alpha', -pi/2, 'a', -30, 'd', 0, 'offset', 0, '
modified', 'qlim', [-143.5*pi/180 43.5*pi/180]);
6 L(3) = Link('revolute','alpha', pi/2, 'a', 30, 'd', 251.52, 'offset', 0, '
modified', 'qlim', [-168.5*pi/180 168.5*pi/180]);
7 L(4) = Link('revolute','alpha', -pi/2, 'a', 38.51, 'd', 0, 'offset', -pi/2,
'modified', 'qlim', [-123.5*pi/180 80*pi/180]);
8 L(5) = Link('revolute','alpha', -pi/2, 'a', 42.37, 'd', 267.01, 'offset',
0, 'modified', 'qlim', [-290*pi/180 290*pi/180]);
9 L(6) = Link('revolute','alpha', pi/2, 'a', -28.8, 'd', 0, 'offset', 0, '
modified', 'qlim', [-88*pi/180 138*pi/180]);
10 L(7) = Link('revolute','alpha', -pi/2, 'a', 27.24, 'd', 35.85, 'offset', 0,
'modified', 'qlim', [-229*pi/180 229*pi/180]);
11
12 % R = SerialLink(L,'name','IRB 14050')
13 % R.plot([q1 q2 q3 q4 q5 q6 q7],'workspace',[-50 50 -50 50 -50 50])
14 % hold on
15 % trplot(eye(4), 'width',2,'arrow')
16 % axis([-1000 1000 -1000 1000 0 1000])
17 % R.teach([q1 q2 q3 q4 q5 q6 q7])
18 % hold off

```

Con la información obtenida anteriormente y los valores de q_1 , q_2 , q_3 y a_4 obtenidos mediante la cinemática inversa, es posible establecer la matriz de rotación que relaciona la base del robot con la muñeca cinemática. Otro valor que se puede establecer a partir de la cinemática directa es la matriz de rotación que relaciona la muñeca con el efector final la cual esta en términos de q_5 , q_6 y q_7 lo que permite resolver el siguiente sistema de ecuaciones:

$${}^0R_7 = {}^0R_4 \cdot {}^4R_7 \quad (21)$$

Despejando las incógnitas de la anterior ecuación se obtiene:

$${}^4R_7 = {}^0R_4^{-1} \cdot {}^0R_7 \quad (22)$$

Debido a que se tiene un sistema ortogonal y ortonormal y a las propiedades de las matrices de rotación es posible establecer la siguiente relación:

$${}^0R_4^{-1} = {}^0R_4^T \quad (23)$$

Por lo tanto:

$${}^4R_7 = {}^0R_4^T \cdot {}^0R_7 \quad (24)$$

El lado derecho de la ecuación esta compuesto por valores conocidos los cuales se representan de la siguiente manera:

$${}^0R_4^T \cdot {}^0R_7 = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix} \quad (25)$$

Por otro lado, la matriz de incógnitas puede ser representada a partir de rotaciones realizadas mediante los ángulos de euler ZYZ lo que daría como resultado la siguiente matriz:

$${}^4R_7 = \begin{pmatrix} C_5C_6C_7 - S_5S_7 & -C_5C_6S_7 - S_5C_7 & C_5S_6 \\ S_5C_6C_7 + C_5S_7 & -S_5C_6S_7 + C_5C_7 & S_5S_6 \\ -S_6C_7 & S_6S_7 & C_6 \end{pmatrix} \quad (26)$$

El sistema de ecuaciones resultante es:

$$\begin{pmatrix} C_5C_6C_7 - S_5S_7 & -C_5C_6S_7 - S_5C_7 & C_5S_6 \\ S_5C_6C_7 + C_5S_7 & -S_5C_6S_7 + C_5C_7 & S_5S_6 \\ -S_6C_7 & S_6S_7 & C_6 \end{pmatrix} = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix} \quad (27)$$

2. Multiplicidad de Soluciones

Mediante el análisis realizado en el apartado anterior fue posible establecer la cinemática inversa para el robot IRB 14050 al determinar los valores de q_1 , q_2 , q_3 , q_4 , q_5 , q_6 y q_7 para una posición dada del efector final. Los resultados obtenidos se resumen a continuación:

$$D = \sqrt{-\frac{-L_1^2 + 2 * L_1 * z + L_2^2 + 2 * L_2 * L_3 + L_3^2 - x^2 - y^2 - z^2}{L_4 * (2 * L_2 + 2 * L_3 + L_4)}} \quad (37)$$

$$q_4 = \tan^{-1} \left(\frac{\sqrt{1 - D^2}}{D} \right) \quad (38)$$

$$q_2 = \frac{\pi}{2} - \left(\tan^{-1} \left(\frac{z - L_1}{x^2 + y^2} \right) + \tan^{-1} \left(\frac{L_4 \cos(q_4) \cdot \sin(q_4)}{L_2 + L_3 + L_4 \cos^2(q_4)} \right) \right) \quad (39)$$

$$D_1 = \frac{x^2 + y^2 - (L_3 \sin(q_2 + q_4) + L_3 \sin(q_2))^2 - (L_2 \sin(q_2))^2}{2(L_3 \sin(q_2 + q_4) + L_3 \sin(q_2))(L_2 \sin(q_2))} \quad (40)$$

$$q_3 = \tan^{-1} \left(\frac{\sqrt{1 - D_1^2}}{D_1} \right) \quad (41)$$

$$q_1 = \tan^{-1} \left(\frac{y}{x} \right) + \tan^{-1} \left(\frac{(L_3 \sin(q_2) + L_4 \sin(q_2 + q_4)) \cdot \sin(q_3)}{L_2 \sin(q_2) + (L_3 \sin(q_2) + L_4 \sin(q_2 + q_4)) \cdot \cos(q_3)} \right) \quad (42)$$

$$q_5 = \tan^{-1} \left(\frac{R_{23}}{R_{13}} \right) \quad q_5 = \tan^{-1} \left(\frac{-R_{23}}{-R_{13}} \right) \quad (43)$$

$$q_6 = \tan^{-1} \left(\frac{\sqrt{1 - R_{33}^2}}{R_{33}} \right) \quad (44)$$

$$q_7 = \tan^{-1} \left(\frac{-R_{32}}{R_{31}} \right) \quad q_7 = \tan^{-1} \left(\frac{R_{32}}{-R_{31}} \right) \quad (45)$$

Dadas las anteriores ecuaciones es posible evidenciar que q_4 , q_3 y q_6 presentan raíces cuadradas por lo que cada uno tienen 2 diferentes posibilidades para un mismo punto del efector final. Por el otro lado, se evidencia que q_5 y q_7 también presentan dos diferentes posibilidades. Por lo anterior, es posible afirmar que para llevar el efector final a un punto especificado pueden existir hasta 10 diferentes posibilidades. En adición, es posible evidenciar la redundancia que presenta el robot dado que dispone de mas grados de libertad de los que necesita para ubicarse en el espacio, esto es un factor importante dado que le permite tener esta multiplicidad de soluciones.

3. Cinemática inversa mediante RVC

■ ¿Cual es la diferencia entre estas funciones?

1. **ikine:** solución numerica a la cinemateca inversa, esta realizada para 6 grados de libertad o mas idelamente, en caso de tener menos grados de libertad se pueden obtener aproximaciones ignorando ciertos parámetros.
2. **ikine6:** solución analitica para robots de 6 grados de libertad, estos deben estar con parametros dh no modificados.

3. **ikine3**: también para robots no modificados en sus parámetros, funciona de la misma manera que la función ikine6 pero solamente aplicable a robots sin muñeca.

4. **ikine_{sym}** : *solución inversa simbólica, genera varias posibles soluciones, sin embargo no es recomendada*

5. **ikunc**: solución numérica que se obtiene ignorando los límites de las articulaciones.

- ¿Cuál debe usar para su robot y por qué? (Revise la documentación del Toolbox)

Al ser un robot con 7 grados de libertad, sin muñeca y con parámetros dados en DHmod la función que más se ajusta para obtener la cinemática inversa haciendo uso del toolbox en Matlab es **ikine**.

Para realizar la cinemática inversa mediante el RVC en Matlab se genera el robot de 7 grados de libertad a través del SerialLink. Una vez se tienen programados los parámetros del robot se puede hacer uso de la función ikine, la cual depende principalmente de la matriz homogénea de transformación para cierta posición conocida de la herramienta de trabajo TCL y también de otros parámetros opcionales como varargin o el vector máscara que especifica el DOF cartesiano (en el marco de coordenadas de la muñeca) que se ignorará al llegar a una solución. El vector de máscara tiene seis elementos que corresponden a la traslación en X, Y y Z, y la rotación alrededor de X, Y y Z respectivamente. El valor puede ser 0 (para ignorar) o 1. La solución se computa a través de iteraciones.

```

1 %%Mlab2
2 close all
3 % Creacion del modelo geometrico con la función Link
4 L(1) = Link('revolute','alpha', 0, 'a', 0, 'd', 305.83, 'offset', 0, '
    modified', 'qlim', [-337*pi/360 337*pi/360]);
5 L(2) = Link('revolute','alpha', -pi/2, 'a', -30, 'd', 0, 'offset', 0, '
    modified', 'qlim', [-287*pi/360 29*pi/120]);
6 L(3) = Link('revolute','alpha', pi/2, 'a', 30, 'd', 251.52, 'offset', 0, '
    modified', 'qlim', [-337*pi/360 337*pi/360]);
7 L(4) = Link('revolute','alpha', -pi/2, 'a', 38.51, 'd', 0, 'offset', -pi/2,
    'modified', 'qlim', [-247*pi/360 4*pi/9]);
8 L(5) = Link('revolute','alpha', -pi/2, 'a', 42.37, 'd', 267.01, 'offset',
    0, 'modified', 'qlim', [-29*pi/18 29*pi/18]);
9 L(6) = Link('revolute','alpha', pi/2, 'a', -28.8, 'd', 0, 'offset', 0, '
    modified', 'qlim', [-22*pi/45 23*pi/30]);
10 L(7) = Link('revolute','alpha', -pi/2, 'a', 27.24, 'd', 35.85, 'offset', 0,
    'modified', 'qlim', [-229*pi/180 229*pi/180]);
11
12 R = SerialLink(L,'name','Robot URRPR DE 6 GDL')
13 %% R.plot(qq)
14 %%Comparacion
15 q(1,:)= [0 0 0 0 0 0 0];
16 q(2,:)= [0 0 0 pi/2 pi -pi/3 0];
17 q(3,:)= [-pi/3 -pi/2 pi/2 0 0 0 0];
18 q(4,:)= [pi/4 0 pi -pi/3 pi/2 pi/4 0];
19 q(5,:)= [pi/4 0 pi -pi/3 pi/2 pi/4 0];
20 q(6,:)= [pi/4 0 pi -pi/3 pi/2 pi/4 0];
21 q(7,:)= [pi/4 0 pi -pi/3 pi/2 pi/4 0];
22 qsize=size(q);
23
24 T= ttranz(0)*trotz(pi/2)*trotz(0)*transl(341.5,0,598)
25
26 qq = R.ikine(T,[0 0 0 0 0 0 0])

```

El vector de salida qq contiene el valor q de cada articulación que permite obtener la posición de entrada a la función a través de la matriz T de transformación.

4. Cinemática inversa mediante RST

Para realizar la cinemática inversa mediante el Robotic System Toolbox se crea el robot de 7 grados de libertad y con los datos de este programa se obtienen 3 datos importantes: la configuración objetivo a partir de la pose de calibración (en este caso denominada como 'randConfig'), la

configuración por defecto o también llamada 'Home' (en el código denominada como 'initialGuess') y la transformada del efector final (denominada en el código como tform).

Con la información anterior se procede con la cinemática inversa: primero creando un objeto (ik) de la clase 'inverseKinematics'; segundo, se designa un vector de 6 elementos denominado 'weights' usado para el algoritmo que hallará la configuración; tercero, se utiliza la función del objeto 'inverseKinematics' de la cual se obtiene la configuración solución (configSol) y la información del error y las iteraciones (solnInfo).

```

1 clear
2 clc
3
4 %se cargan las variables del robot IRB 14050 en posicion de calibracion
5 load('RSTcalibradoEscala')
6
7 % Obtain puma1 An arbitrary configuration of
8 randConfig = calib;
9
10 % Use this transform as the target pose for the end effector
11 tform = getTransform(robot,randConfig,'b7');
12
13 % Establish robor Model ik object
14 ik = inverseKinematics('RigidBodyTree',robot);
15 % I don't know why I want to set the weight of different components of the pose.
16 % The estimation is related to the specific algorithm
17 % For the bearing component, use a weight smaller than the position component
18 weights = [0.25 0.25 0.25 0.5 1 0.5 ];
19
20 % The robot's home Joint angle of configuration as initial guess
21 initialguess = home;
22 % Inverse solution of joint angle according to the expected position and pose of
23 % end effector
24 [configSoln,solnInfo] = ik('b7',tform,weights,initialguess);
25
26 figure(1)
27 show(robot,configSoln)
28 figure(2)
29 show(robot,randConfig)
30 %https://programmer.help/blogs/robotics-system-toolbox-learning-notes-inverse-
31 % kinematics-related-functions.html

```

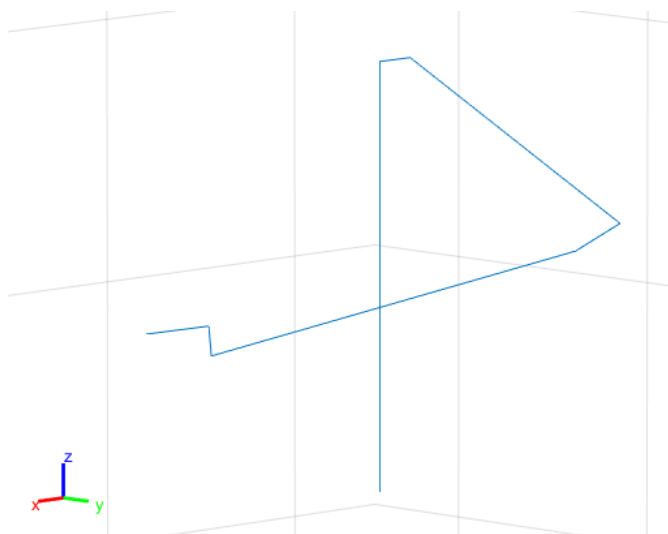


Figura 8: Configuración objetivo

| configSoln × randConfig × solnInfo × | | |
|--------------------------------------|-----------|---------------|
| 1x7 struct with 2 fields | | |
| Fields | JointName | JointPosition |
| | 'j1' | 0 |
| | 'j2' | -2.2689 |
| | 'j3' | -2.3562 |
| | 'j4' | 0.5236 |
| | 'j5' | 0 |
| | 'j6' | 0.6981 |
| | 'j7' | 0 |

Figura 9: Vector de la configuración objetivo

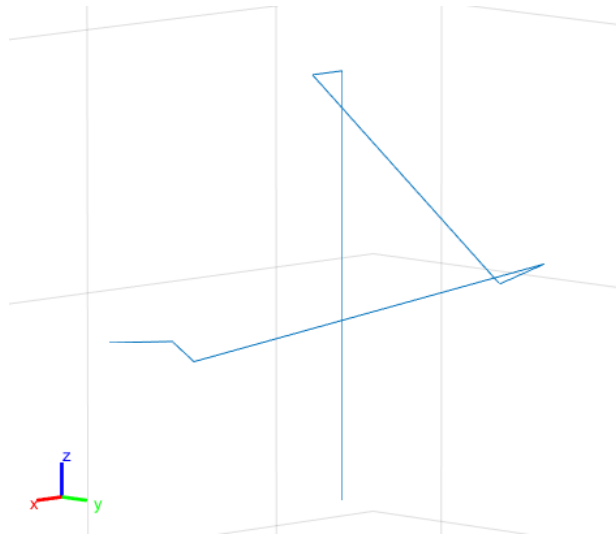


Figura 10: Configuración solución

| configSoln × randConfig × solnInfo × | | |
|--------------------------------------|-----------|---------------|
| 1x7 struct with 2 fields | | |
| Fields | JointName | JointPosition |
| | 'j1' | -3.1303 |
| | 'j2' | 2.1771 |
| | 'j3' | -2.1182 |
| | 'j4' | 3.0833 |
| | 'j5' | -3.1413 |
| | 'j6' | 1.2135 |
| | 'j7' | 0.2437 |

Figura 11: Vector de la configuración solución

| configSoln × randConfig × solnInfo × | | |
|--------------------------------------|------------|--|
| 1x1 struct with 5 fields | | |
| Field | Value | |
| Iterations | 258 | |
| NumRandomRest... | 0 | |
| PoseErrorNorm | 9.9372e-09 | |
| ExitFlag | 1 | |
| Status | 'success' | |

Figura 12: Información de la configuración solución

5. Comparación entre los métodos RVC y RST para cinemática inversa

Al seguir los dos procedimientos sugeridos para cada uno de los toolboxes se evidencia que ambos logran representar con alta precisión todos los elementos del robot. Además, para realizar el código, es necesario tener los conocimientos sobre matrices de transformadas homogéneas, ejes de referencia, orientación, posición, cinemática directa y cinemática inversa, entre otras cosas que son necesarios para la implementación de un adecuado modelo. Su principal diferencia radica en las funciones y la sintaxis empleada ya que ambos tienen la misma finalidad.

Al analizar los dos resultados obtenidos es posible establecer que gráficamente las herramientas de Peter Corke (RVC) tiene mejores representaciones ya que mediante los comandos es posible representar en el gráfico las articulaciones diferenciando su tipo y su eje principal de movimiento. Por el otro lado, mediante la herramienta RST el robot solo se representa el robot mediante líneas y puntos lo cual no es muy fiel a la realidad. Por el otro lado, el toolbox de Peter Corke incluye una interfaz gráfica que permite que los usuarios modifiquen en tiempo real los parámetros de simulación y obtenga la posición exacta del último eslabón, además de que la sintaxis e información requerida por el toolbox es menor y más fácil de introducir. Por todo lo expuesto se concluye que el toolbox de Peter Corke es más completo, eficiente y con más facilidad de uso que el RST toolbox, haciéndolo la mejor alternativa.

6. Compruebe mediante su modelo geométrico inverso la configuración del robot para las posturas de la herramienta halladas en el anterior laboratorio, tales como puntos de calibración

7. Posturas propuestas

Para el robot trabajado se propusieron las siguientes 4 poses, la primera hace referencia a la posición cuando la rotación de todas las posiciones son iguales a 0, la segunda es la posición de Home y las últimas dos son posiciones arbitrarias que se tomaron.

Tabla 1: Posturas del Robot

| x | y | z | roll | pitch | yaw | q1 | q2 | q3 | q4 | q5 | q6 | q7 |
|--------|---------|--------|--------|--------|--------|------|---------|---------|--------|--------|--------|--------|
| 341.50 | 0.00 | 598.00 | 0.00 | 90.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 12.36 | -222.34 | 90.46 | 52.29 | -46.79 | 103.95 | 0.00 | -130.00 | -135.00 | 30.00 | 0.00 | 40.00 | 0.00 |
| 30.54 | -9.55 | 860.29 | 1.62 | -7.08 | 154.03 | 0.00 | 0.00 | -18.07 | -84.08 | 206.00 | -1.46 | -33.62 |
| 252.85 | -82.84 | 406.64 | 175.28 | -2.25 | 81.45 | 0.00 | 0.00 | -18.07 | 38.60 | 179.00 | -46.23 | -55.96 |

8. Verificación de los resultados empleando el GUI

Mediante el GUI se analizaron los resultados obtenidos la tabla 1 para cada una de las poses propuestas. Los resultados obtenidos se muestran a continuación:

Primera posición:

- q1=0
- q2=0

- $q_3=0$
- $q_4=0$
- $q_5=0$
- $q_6=0$
- $q_7=0$

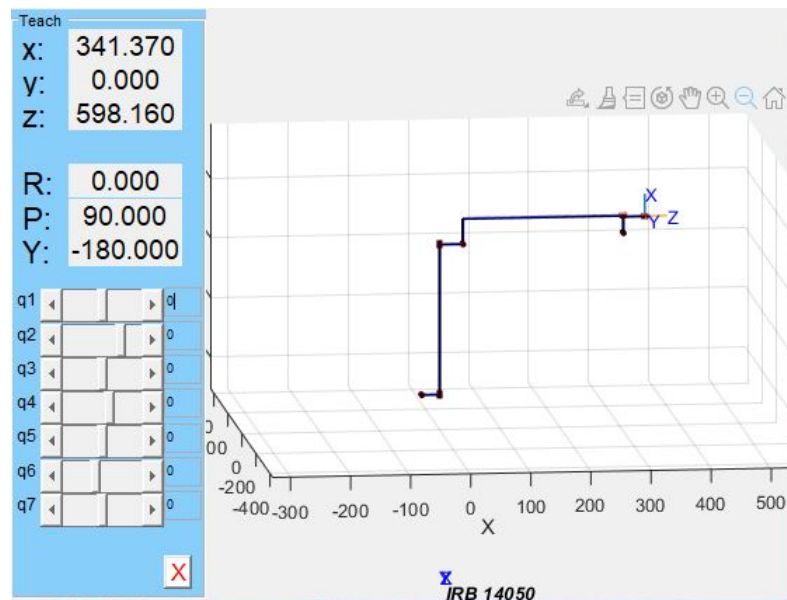


Figura 13: Primera pose

Posición HOME:

- $q_1=0$
- $q_2=-130$
- $q_3=-135$
- $q_4=30$
- $q_5=0$
- $q_6=40$
- $q_7=0$

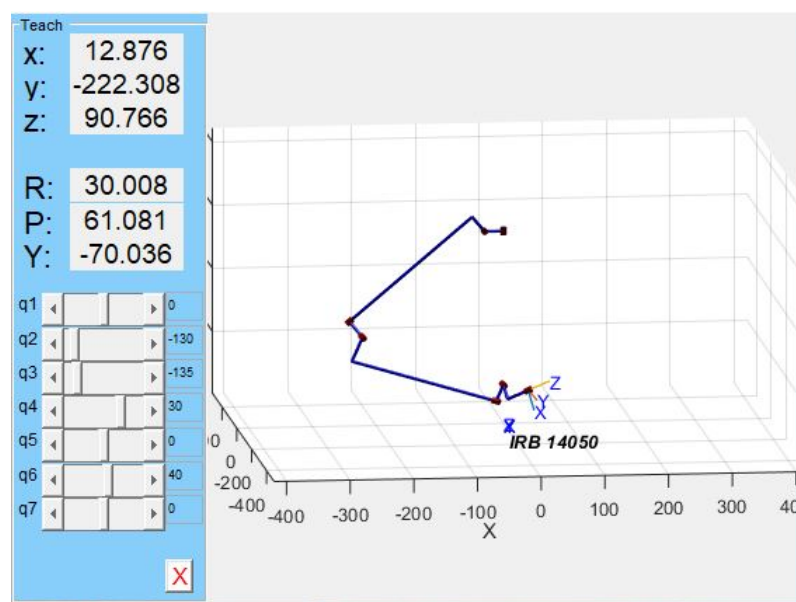


Figura 14: Posición de HOME

Tercera posición:

- $q1=0$
- $q2=0$
- $q3=-18.07$
- $q4=-84.08$
- $q5=206$
- $q6=-1.46$
- $q7=-33.62$

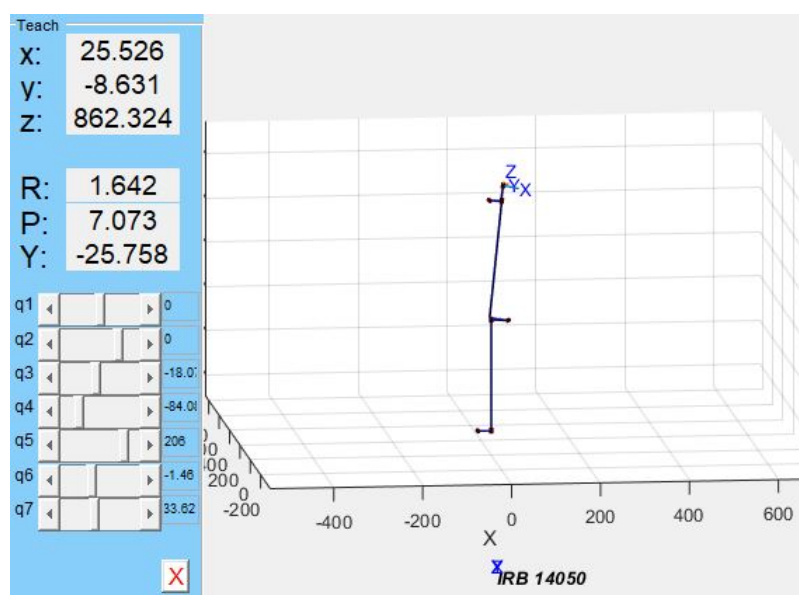


Figura 15: Posición de HOME

Cuarta posición:

- $q_1=0$
- $q_2=0$
- $q_3=-18.07$
- $q_4=38.6$
- $q_5=179$
- $q_6=-46.23$
- $q_7=-55.96$

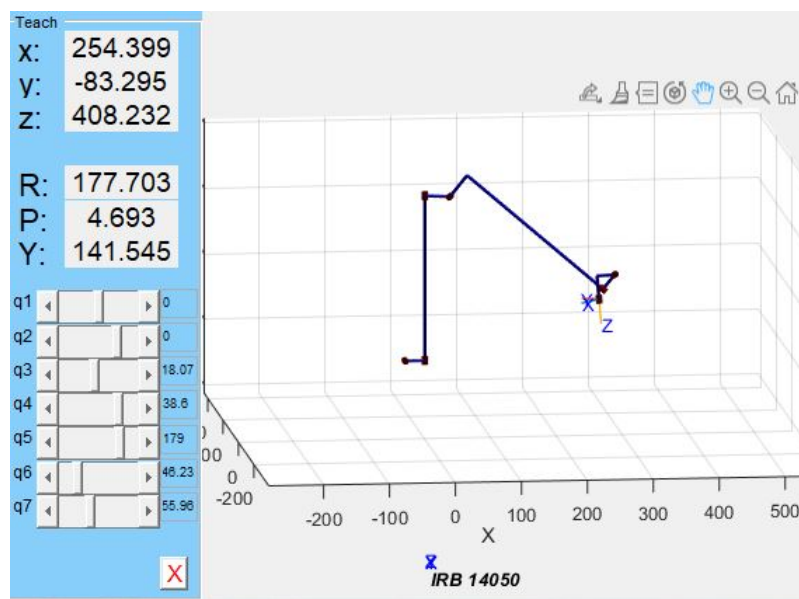


Figura 16: Posición de HOME

9. Actualice la GUI para que tenga la opción de ingresar la posición de la herramienta

Para la actualización de interfaz gráfica se tomó el avance entregado en el laboratorio pasado y empleando todos los códigos desarrollados para este laboratorio se incorporó la cinemática inversa del robot permitiendo al usuario ingresar una posición y orientación determinada en la cual se calculan los ángulos de cada una de las articulaciones y se grafica la posición del robot. El diseño propuesto para la GUI fue el siguiente:

Referencias

- [1] Corke P. Robotics, Vision, Control. 2007
- [2] Craig, John. Introduction to Robotics: Mechanics and Control. 4 Ed. 2017
- [3] ABB. (2021). Product manual IRB 14050. Recuperado de <https://abb.sluzba.cz/Pages/Public/OmniCoreRoboticsDocumentationRW7/Robots/Collaborative%20Robots/IRB%2014050/en/3HAC064625-001.pdf>

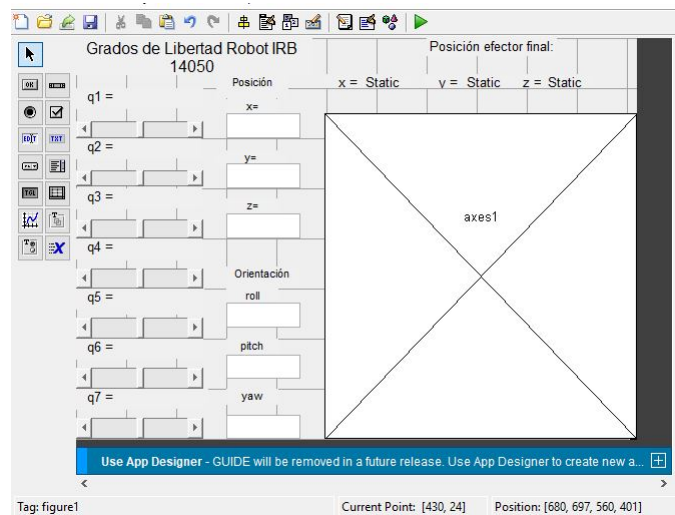


Figura 17: interfaz gráfica