

## Robot Planar

Waldir Toscano, Leiser Angarita, Elimelec Ruiz

02/04/2025

### Resumen

Se implementó una función modificada para visualizar los estados intermedios de un brazo robótico de 2 grados de libertad, basada en transformaciones homogéneas. Mediante gráficas secuenciales, se demostró cómo cada rotación ( $\theta_1, \theta_2$ ) y traslación ( $l_1, l_2$ ) afecta la posición final del efector. Adicionalmente, se verificó que un brazo con extremidades de 2 unidades puede alcanzar cualquier punto dentro de una cuadrícula 2x2, excepto en casos con restricciones físicas no consideradas en el modelo teórico.

## 1. Introducción

El análisis cinemático de brazos robóticos requiere comprender las transformaciones geométricas entre sus articulaciones. Este estudio explora un brazo de 2 eslabones ( $l_1=l_2=2$  unidades) mediante visualización iterativa de sus estados, desde la rotación inicial hasta la posición final. Los objetivos incluyen: (1) representar gráficamente cada transformación matricial ( ${}_{0H1}$ ,  ${}_{1H2}$ , etc.) y (2) evaluar la capacidad de alcance en una cuadrícula 2x2. El enfoque combina herramientas computacionales (Python) con principios de robótica básica, destacando la utilidad de las matrices homogéneas para modelar movimientos articulados.

## 2. Objetivos

Los objetivos de esta actividad son:

- Dibujar los estados anteriores del brazo robotico de 2 extremidades.
- Determinar si un brazo de 2 extremidades con medida de 2 unidades pueden alcanzar cualquier objeto dentro de una cuadrícula 2x2.

## 3. Descripción de la Actividad

Se reutiliza el código entregado por la institución, con el fin de modificar la función dibujar, para que esta permita dibujar no solo el estado final del brazo, si no todos los estados previos que toma el brazo para llegar a dicha posición final. Además se implementa otro código con el cual se pueda evidenciar si un brazo mecánico el cual sus dos extremidades miden 2 unidades puedan alcanzar cualquier objeto en una cuadrícula 2x2.

### 3.1. Codificación

Listing 1: Se generan las siguientes matrices con el fin de analizarlas para tener la idea de como realizar los dibujos de esos estados

```
1
2 H2 = simplify(_OH1@_1H2)
3 H2
4
5 H3 = simplify(_OH1@_1H2@_2H3)
6 H3
```

Listing 2: Funcion modificada para que se dibujen todos los estados previos que toma el brazo para llegar a su posicion final

```
1 # Cargar datos
2 def dibujar_estados_brazo(theta1, theta2, l1, l2):
3     """Funcion para dibujar un brazo mecanico con 2 grados de
4     libertad.
5
6     Parametros:
7         theta1: Angulo de la primera articulacion en radianes.
8         theta2: Angulo de la segunda articulacion en radianes.
9         l1: Longitud del primer extremo
10        l2: Longitud del segundo extremo
11    """
12    # Estado 0: Solo la base (sin transformaciones)
13    plt.figure(figsize=(6, 6))
14    plt.plot(0, 0, 'ro', markersize=10, label='Base')
15    plt.xlim([-l1 - l2, l1 + l2])
16    plt.ylim([-l1 - l2, l1 + l2])
17    plt.title("Estado 0: Solo la base")
18    plt.grid(True)
19    plt.legend()
20    plt.show()
21
22    # Estado 1: Despues de _OH1 (rotacion theta1)
23    x1_rot = 0 # Aun no hay traslacion, solo rotacion
24    y1_rot = 0
25    plt.figure(figsize=(6, 6))
26    plt.plot(0, 0, 'ro', markersize=10, label='Base')
27    plt.plot([0, 0], [0, 0], 'b-', linewidth=3, label='Eslabon
28    1 (rotado)')
29    plt.xlim([-l1 - l2, l1 + l2])
30    plt.ylim([-l1 - l2, l1 + l2])
31    plt.title(f"Estado 1: Rotacion theta1 = {np.rad2deg(theta1)
32    :.1f}")
33    plt.grid(True)
```

```

31 plt.legend()
32 plt.show()
33
34 # Estado 2: Despues de _0H1 @ _1H2 (primer eslabon
extendido)
35 x1 = l1 * np.cos(theta1)
36 y1 = l1 * np.sin(theta1)
37 plt.figure(figsize=(6, 6))
38 plt.plot(0, 0, 'ro', markersize=10, label='Base')
39 plt.plot([0, x1], [0, y1], 'b-', linewidth=3, label='
Eslabon 1')
40 plt.plot(x1, y1, 'ko', markersize=8, label='Articulacion 2
')
41 plt.xlim([-l1 - l2, l1 + l2])
42 plt.ylim([-l1 - l2, l1 + l2])
43 plt.title(f"Estado 2: Traslacion l1 = {l1}")
44 plt.grid(True)
45 plt.legend()
46 plt.show()
47
48 # Estado 3: Despues de _0H1 @ _1H2 @ _2H3 (rotacion theta2
)
49 x2_rot = x1 # Aun no se aplica l2, solo rotacion theta2
50 y2_rot = y1
51 plt.figure(figsize=(6, 6))
52 plt.plot(0, 0, 'ro', markersize=10, label='Base')
53 plt.plot([0, x1], [0, y1], 'b-', linewidth=3, label='
Eslabon 1')
54 plt.plot([x1, x1], [y1, y1], 'g-', linewidth=3, label='
Eslabon 2 (rotado)')
55 plt.plot(x1, y1, 'ko', markersize=8, label='Articulacion 2
')
56 plt.xlim([-l1 - l2, l1 + l2])
57 plt.ylim([-l1 - l2, l1 + l2])
58 plt.title(f"Estado 3: Rotacion theta2 = {np.rad2deg(theta2
):.1f}")
59 plt.grid(True)
60 plt.legend()
61 plt.show()
62
63 # Estado 4: Despues de _0H1 @ _1H2 @ _2H3 @ _3H4 (posicion
final)
64 x2 = x1 + l2 * np.cos(theta1 + theta2)
65 y2 = y1 + l2 * np.sin(theta1 + theta2)
66 plt.figure(figsize=(6, 6))
67 plt.plot(0, 0, 'ro', markersize=10, label='Base')
68 plt.plot([0, x1], [0, y1], 'b-', linewidth=3, label='

```

```

69     Eslabon 1')
    plt.plot([x1, x2], [y1, y2], 'g-', linewidth=3, label='
Eslabon 2')
70     plt.plot(x1, y1, 'ko', markersize=8, label='Articulacion 2
')
71     plt.plot(x2, y2, 'go', markersize=8, label='Efector final'
)
72     plt.xlim([-l1 - 12, 11 + 12])
73     plt.ylim([-l1 - 12, 11 + 12])
74     plt.title(f"Estado 4: Traslacion 12 = {l2} (Posicion final
)")
75     plt.grid(True)
76     plt.legend()
77     plt.show()

```

Listing 3: Cree dos arrays con diferentes angulo de rotacion para theta1 y theta2 y luego ejecute un bucle anidado con el fin de generar como una animacion obteniendo las diferentes imagenes de lo que dibujaba con los diferentes angulos cabe aclarar que la funcion que se utiliza es la misma brindada por la institucion

```

1  theta_e1 = np.arange(15, 361, 15)
2  theta_e2 = np.arange(-15, -361, -15)
3  l1 = 2
4  l2 = 2
5
6  for i in range(len(theta_e1)):
7      for j in range(len(theta_e2)):
8          dibujar_brazo_mecanico_for(np.deg2rad(theta_e1[i]), np.
deg2rad(theta_e2[j]), l1, l2)

```

#### 4. Gráficas

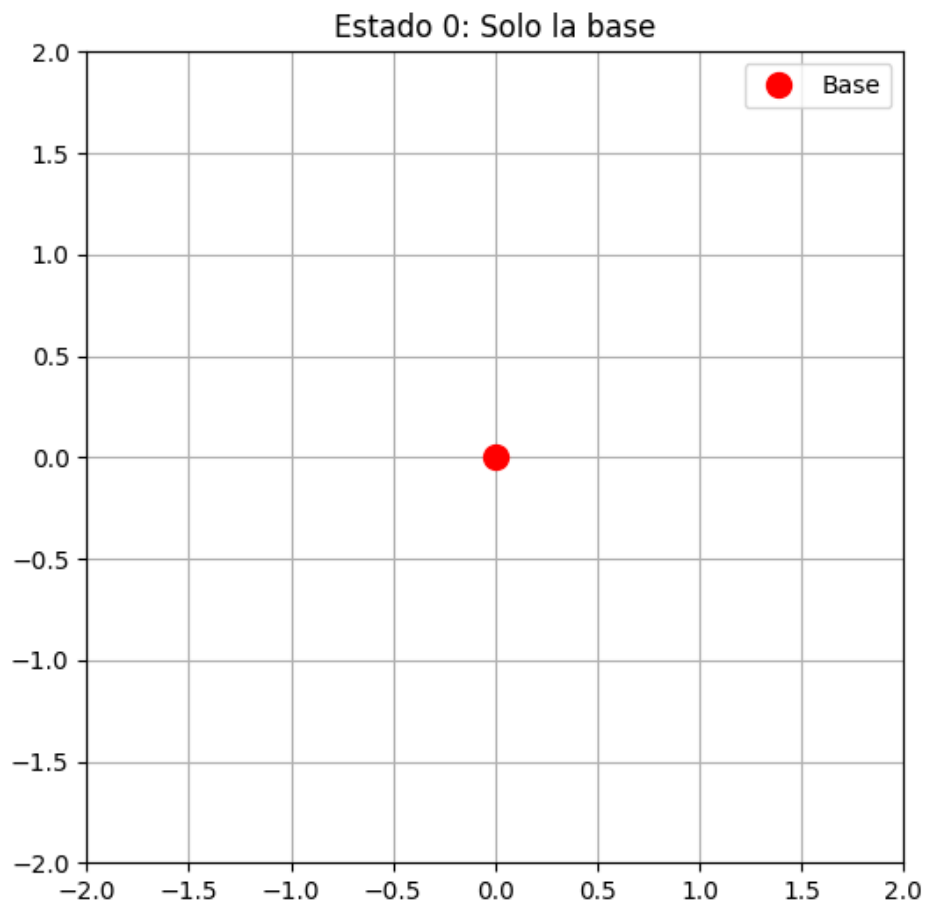


Figura 1: Estado 0: Solo la base (sin transformaciones)

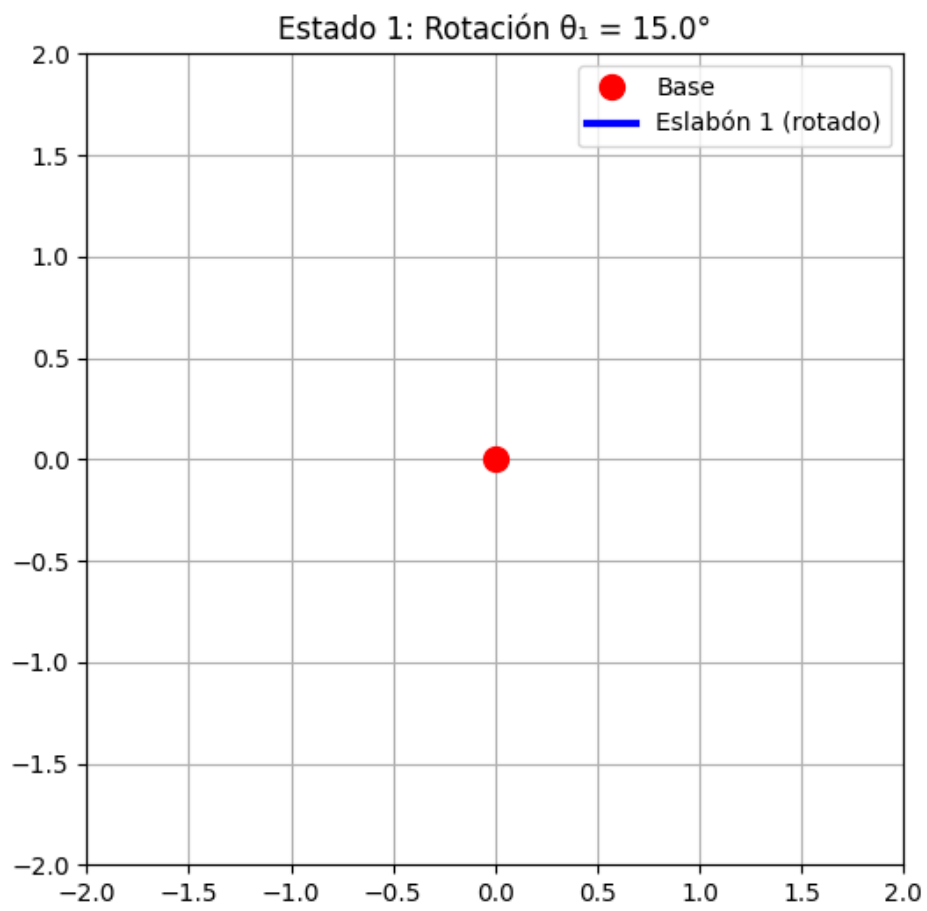


Figura 2: Estado 1: Despues de  $\_0H1$  (rotación  $\theta_1$ )

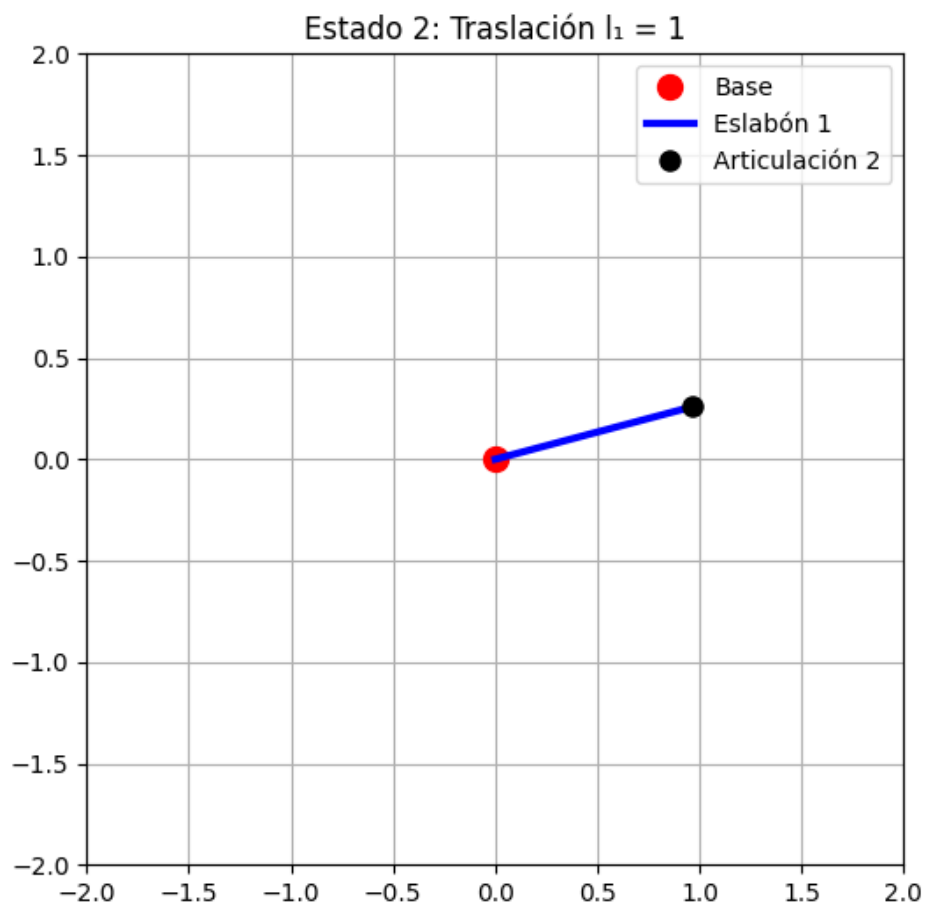


Figura 3: Estado 2: Despues de  $_0H1 @ _1H2$  (primer eslabon extendido)

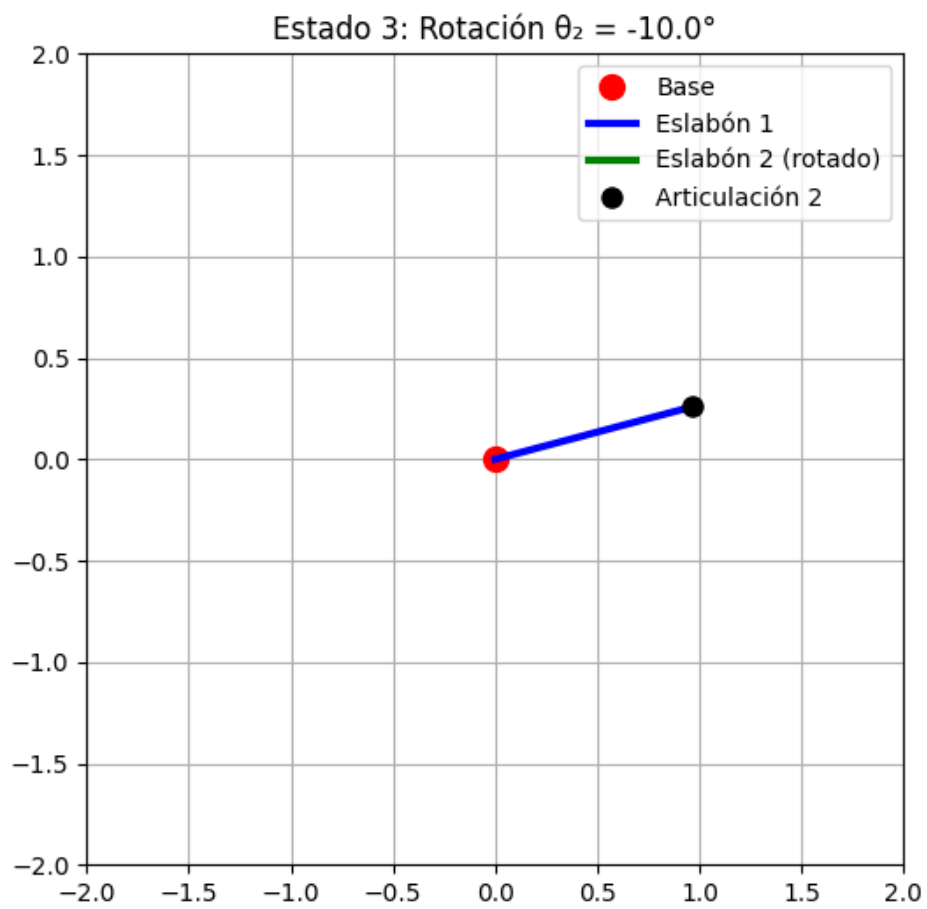


Figura 4: Estado 3: Despues de  $\_0H1 @ \_1H2 @ \_2H3$  (rotacion  $\theta_2$ )



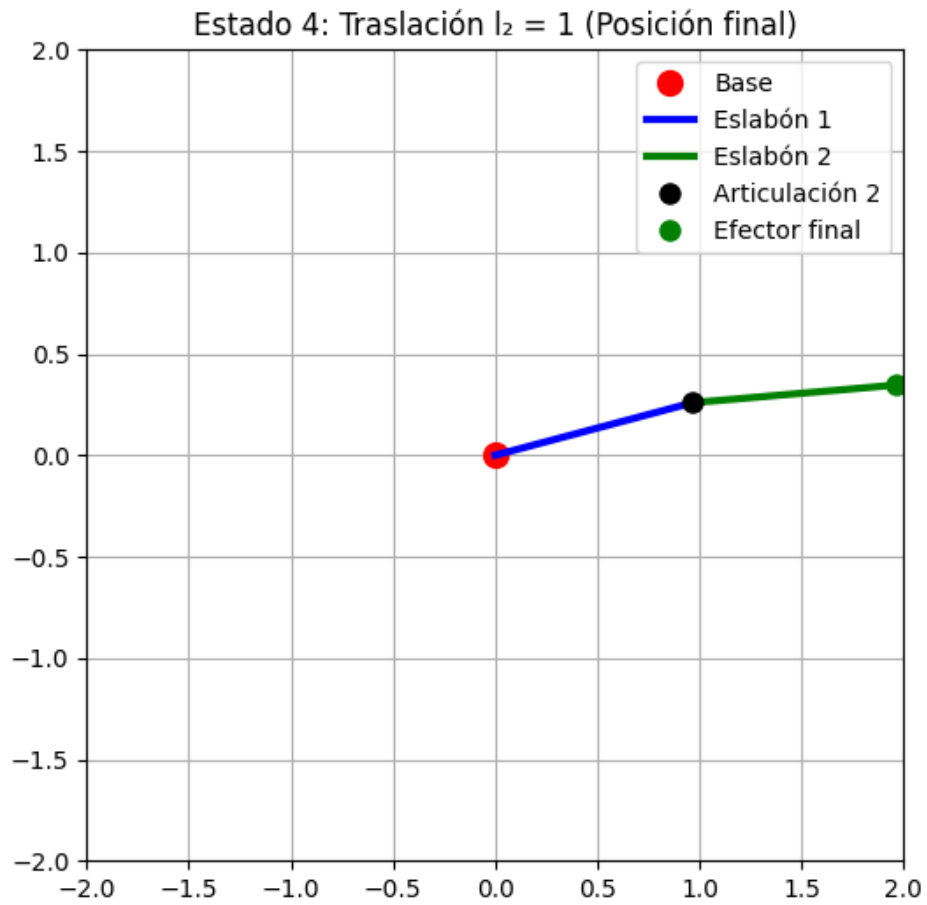


Figura 5: Estado 4: Después de  $\theta_1$  @  $\theta_2$  @  $\theta_3$  @  $\theta_4$  (posición final)

## 5. Análisis de gráficas

- **Figura 1:** Muestra la escena inicial sin ninguna transformación.
- **Figura 2:** Se realiza la rotación de  $\theta_1$  sin embargo no se visualiza porque aún no se aplica la translación.
- **Figura 3:** Se extiende la primera articulación con la rotación ya aplicada
- **Figura 4:** Se realiza la rotación de  $\theta_2$  sin embargo no se visualiza porque aún no se aplica la translación.

- **Figura 5:** Se extiende la segunda articulación con la rotación ya aplicada y se forma la posición final del brazo mecánico.

## 6. Conclusión

Los dibujos realizados nos permiten observar como el brazo va tomando su forma hasta llegar a su punto final, al visualizar estos dibujos podemos comprender como se van aplicando estas transformaciones homogéneas basadas en las matrices vistas, que componen el movimiento total del brazo robótico, mostrando el estado del brazo después de aplicar cada transformación matricial.

Al ejecutar el bucle y ver las distintas posiciones que va tomando el brazo al tomar distintos ángulos, puedo concluir que el brazo con 2 unidades en cada extremidad si puede alcanzar cualquier punto dentro de una cuadrícula  $2 \times 2$ , cabe recalcar que esto también es debido a que no se le da ninguna limitación al brazo, si tomamos la perspectiva desde un ejemplo real, esto no se podría debido a las diferentes limitaciones que se pueden presentar, una de ellas sería que por lógica el brazo no podría alcanzar la posición  $0,0$ .