

Análisis de posición de eslabones de robot colaborativo

Estudio de las posiciones de los eslabones de un robot colaborativo mediante matrices de transformación homogénea

Resumen—El presente artículo tiene como objetivo determinar las posibles posiciones que pueden ocupar los eslabones de un robot colaborativo sin que se produzca una colisión entre ellos. Para ello se emplea una combinación de matrices de transformación homogéneas, ejecutadas en un script de MatLab.

Palabras Clave—trayectoria; robot; eslabón; posición; matriz de transformación homogénea

I. INTRODUCCIÓN

Con el objetivo de seleccionar la mejor combinación de longitudes de eslabones en el diseño de un robot colaborativo, se llevó a cabo un análisis de trayectoria, en el que se tienen en cuenta las posibles dimensiones y su influencia en la amplitud de alcance del robot. Es decir que la opción seleccionada será aquella que posibilite la mayor cantidad de puntos para el actuador final¹.

Para ello, es necesario realizar un análisis de trayectoria, en el cual se pueda identificar numéricamente la posición del efector final del robot, tomando como entrada la longitud de los eslabones, y el rango de giro de cada eje. Este estudio se basa en una herramienta matemática muy utilizada en la robótica: la matriz de transformación homogénea.

II. MATRIZ DE TRANSFORMACIÓN HOMOGÉNEA

Esta matriz contiene la información numérica necesaria para ubicar un sistema de ejes de coordenadas (en posición y orientación de sus ejes), partiendo desde otro sistema ubicado en algún punto del espacio [1].

En ella se representan los movimientos de rotación y traslación que se llevan a cabo en los 3 ejes.

Siendo M la matriz de transformación homogénea, de dimensión 4×4 , se compone de una parte que define la rotación del eje de coordenadas, y otra que afecta a la traslación.

Además, es posible concatenar distintos movimientos, representados por diferentes matrices, multiplicándolas y formando una nueva matriz que representa la totalidad de los movimientos.

$$M = \begin{bmatrix} \cos \theta & -\text{sen } \theta & 0 & 0 \\ \text{sen } \theta & \cos \theta & 0 & \Delta_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig. 1. Explicación de las partes de la matriz de transformación homogénea.

A continuación, unos ejemplos de movimientos de rotación y traslación pura de sistemas de ejes de coordenadas, representados en sus respectivas matrices y de forma gráfica:

Ejemplo de matriz de transformación homogénea para un caso de traslación pura sobre el eje x :

$$M = \begin{bmatrix} 1 & 0 & 0 & \Delta_x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig. 2. Representación del movimiento de traslación pura.

Ejemplo de matriz de transformación homogénea para un caso de rotación pura alrededor del eje x :

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\text{sen } \alpha & 0 \\ 0 & \text{sen } \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig. 3. Representación del movimiento de rotación pura.

¹ Elemento ubicado en el extremo del último eslabón del robot en el que usualmente se acoplan herramientas.

III. APLICACIÓN DE LA MATRIZ EN EL ANÁLISIS DE TRAYECTORIA

En este caso, se tienen 6 sistemas de coordenadas, uno por eje. Por lo que también se tienen 6 matrices de transformación homogénea. Cada una de estas contiene el ángulo de rotación y traslación de cada eje. Es importante aclarar que, para este estudio, la distancia de traslación de cada eje es constante, ya que con ella se representa la longitud de cada eslabón.

Los valores de rotación y traslación de cada eje están expresados en función del sistema de coordenadas del eje anterior. Sin embargo, al multiplicar las matrices consecutivamente y en orden creciente, se puede obtener la posición de cualquier eje con respecto al primer sistema de coordenadas. De esta manera es posible conocer la posición de cada articulación según la longitud de los eslabones y los ángulos de giro.

En el siguiente esquema, se observa una simplificación de los eslabones de un robot, con los movimientos que puede realizar cada eje. En esta configuración, los tipos de movimiento de cada uno son:

- Eje 1: Rotación.
- Eje 2: Doblado.
- Eje 3: Doblado.
- Eje 4: Rotación.
- Eje 5: Doblado.

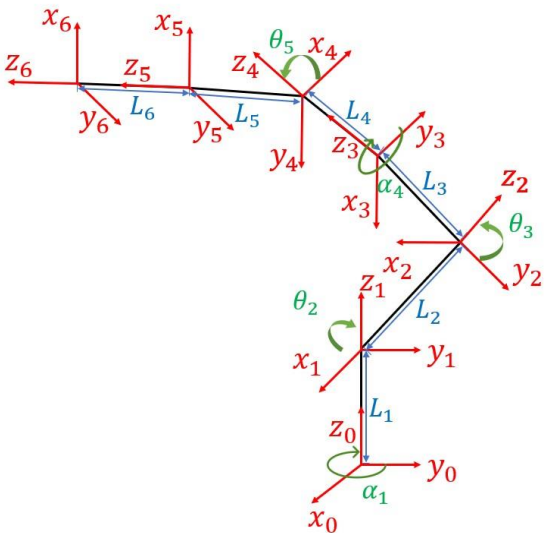


Fig. 4. Esquema simplificado del robot con sus sistemas de coordenadas.

Por lo tanto, las variables de entrada son:

- L_1 = Longitud del eslabón 1
- L_2 = Longitud del eslabón 2
- L_3 = Longitud del eslabón 3
- L_4 = Longitud del eslabón 4

- L_5 = Longitud del eslabón 5
- L_6 = Longitud del efector final
- α_1 = Ángulo de giro de la articulación 1
- θ_2 = Ángulo de giro de la articulación 2
- θ_3 = Ángulo de giro de la articulación 3
- α_4 = Ángulo de giro de la articulación 4
- θ_5 = Ángulo de giro de la articulación 5

Y la variable de salida:

- P_6 = Posición del extremo del efector final

Cabe aclarar que, si bien se pueden conocer las posiciones de todas las articulaciones, la única de interés es la posición del extremo del efector final.

Siguiendo esta lógica se realizó un script en Matlab, cuyo objetivo es poder conocer todas las posiciones posibles del efector final, en un rango de rotación de cada eje:

- Eje 1: de -180° a 180°
- Eje 2: de -140° a 140°
- Eje 3: de -140° a 140°
- Eje 4: de -180° a 180°
- Eje 5: de -140° a 140°

Este script tiene la capacidad de generar todas las alternativas de ángulos de giros posibles para cada articulación (con un paso determinado, por ejemplo: de -180° a 180° , con un incremento de 30° por vez).

Entonces, se ingresan los valores de las longitudes a analizar, y con las combinaciones de ángulos, se llenan las matrices de transformación homogénea, arrojando la posición del efector final para cada combinación. Sin embargo, no todos los puntos son válidos, ya que numerosas posiciones de los eslabones no son coherentes con el comportamiento real del robot, por lo que es necesario analizar cada combinación según las restricciones colocadas.

Las restricciones se establecieron de manera tal que los eslabones no colisionen entre ellos. Por lo tanto, se descartan todas las combinaciones que generen una proximidad inferior a 10 milímetros entre los diferentes eslabones, asegurando que no existirán contactos ni choques entre ellos.

Además, se descartan todas las combinaciones de ángulos que generen posiciones de cualquier eje en $-Z$, ya que esto en la realidad significaría una colisión con el suelo donde está apoyado el robot.

Finalmente, los valores que cumplan con las restricciones establecidas se contabilizan y se grafican.

El objetivo de este análisis es determinar qué combinación de longitudes posibilita la mayor cantidad de posiciones del efector final, es decir, la que otorga mayor rango de movimiento.

En el siguiente gráfico, se observan las posiciones que puede adoptar el efector final del robot colaborativo a analizar, siendo cada una de ellas un punto en el espacio.

Se acotó el gráfico al primer cuadrante, con el fin de facilitar la visualización.

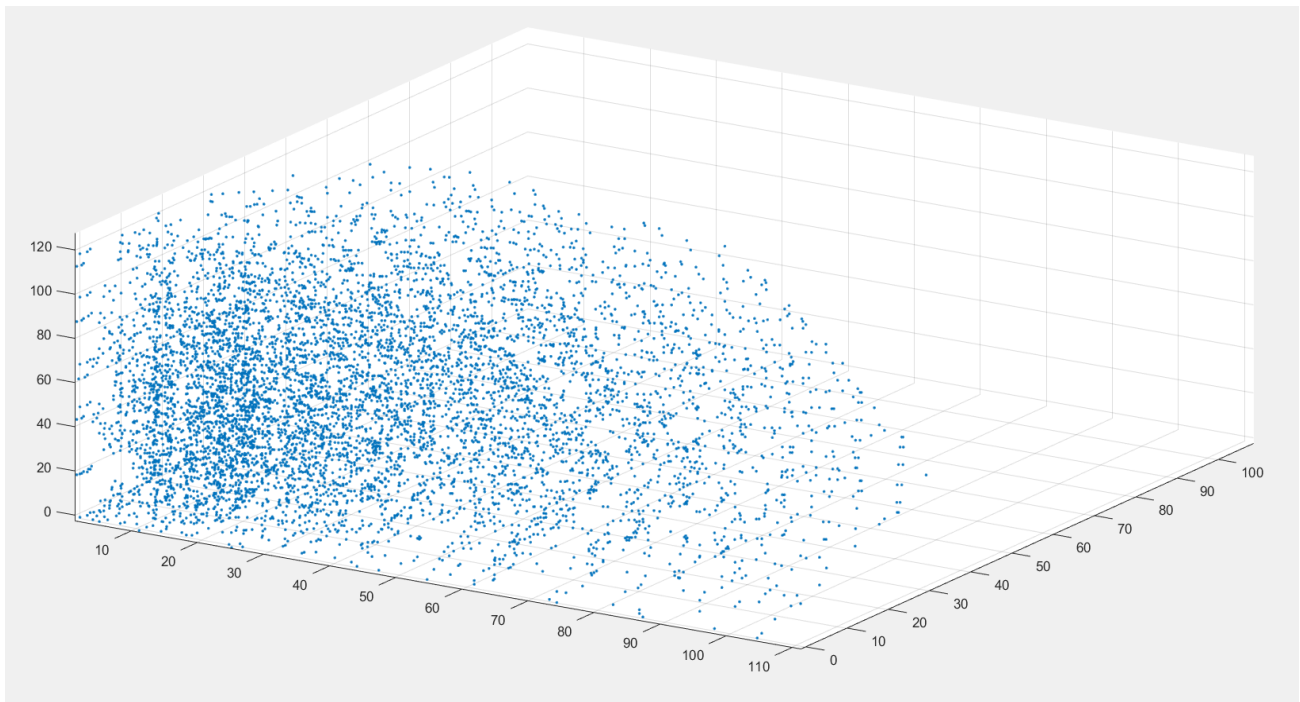


Fig. 5. Ejemplo de posiciones del actuador final.

IV. CÓDIGO

Cabe aclarar que, si bien se pueden conocer las posiciones de todas las articulaciones, la única de interés es la posición del extremo del efector final.

```
%definicion de longitudes%
long1=20;
long2=25;
long3=25;
long4=25;
long5=25;
long6=10;

%declaracion de variables%
a12=0;
a23=0;
a34=0;
a45=0;
a56=0;

%combinaciones de angulos%
paso = 30;
matAng=combvec(-180:paso:180,-
180:paso:180,-180:paso:180,-
180:paso:180,-180:paso:180);

%definicion de variables para el
bucle%
sz=size(matAng);
numfil=sz(1,2);
x=zeros(1,numfil);
y=zeros(1,numfil);
z=zeros(1,numfil);

e = 1;
i = 1;
for n=1:numfil
    %definicion de angulos%
    a12=matAng(1,n)*pi/180;
    a23=matAng(2,n)*pi/180;
    a34=matAng(3,n)*pi/180;
    a45=matAng(4,n)*pi/180;
    a56=matAng(5,n)*pi/180;

    %matrices homogeneas%
    mh1=[cos(a12) -sin(a12) 0
0;sin(a12) cos(a12) 0 0;0 0 1 long1; 0
0 0 1];
    mh2=[1 0 0 0; 0 cos(a23) -sin(a23)
long2*(-sin(a23));0 sin(a23) cos(a23)
long2*cos(a23);0 0 0 1];
    mh3=[1 0 0 0; 0 cos(a34) -sin(a34)
long3*(-sin(a34));0 sin(a34) cos(a34)
long3*cos(a34);0 0 0 1];
    mh4=[cos(a45) -sin(a45) 0
0;sin(a45) cos(a45) 0 0;0 0 1 long4; 0
0 0 1];
    mh5=[1 0 0 0; 0 cos(a56) -sin(a56)
long5*(-sin(a56));0 sin(a56) cos(a56)
long5*cos(a56);0 0 0 1];
    mh6=[1 0 0 0;0 1 0 0;0 0 1 long6;
0 0 0 1];
```

```
pos1=mh1;
pos2=mh1*mh2;
pos3=mh1*mh2*mh3;
pos4=mh1*mh2*mh3*mh4;
pos5=mh1*mh2*mh3*mh4*mh5;
mfin=mh1*mh2*mh3*mh4*mh5*mh6;

%Armado de la matriz con posiciones del
efector final%

x(1,n)=mfin(1,4);
y(1,n)=mfin(2,4);
z(1,n)=mfin(3,4);

%Eliminacion de las posiciones en las
que se intersectan eslabones%

P0 = [0,0,0];
P1 = [pos1(1,4),pos1(2,4),pos1(3,4)];
P2 = [pos2(1,4),pos2(2,4),pos2(3,4)];
P3 = [pos3(1,4),pos3(2,4),pos3(3,4)];
P4 = [pos4(1,4),pos4(2,4),pos4(3,4)];
P5 = [pos5(1,4),pos5(2,4),pos5(3,4)];
P6 = [mfin(1,4),mfin(2,4),mfin(3,4)];

cant_pint=20;

pint1=zeros(cant_pint,3);
pint2=zeros(cant_pint,3);
pint3=zeros(cant_pint,3);
pint4=zeros(cant_pint,3);
pint5=zeros(cant_pint,3);
pint6=zeros(cant_pint,3);

for i=1:cant_pint
    t=i/(cant_pint+1);
    pint1(i,:)=(1-t)*P0+t*P1;
    pint2(i,:)=(1-t)*P1+t*P2;
    pint3(i,:)=(1-t)*P2+t*P3;
    pint4(i,:)=(1-t)*P3+t*P4;
    pint5(i,:)=(1-t)*P4+t*P5;
    pint6(i,:)=(1-t)*P5+t*P6;
end

if pos2(3,4)>0 && pos3(3,4)>0 &&
pos4(3,4)>0 && pos5(3,4)>0 && mfin(3,4)>0
    for g = 1:length(pint1)
        for s = 1:length(pint3)
            d13=norm(pint1(g,:)-
pint3(s,:));
            d14=norm(pint1(g,:)-
pint4(s,:));
            d15=norm(pint1(g,:)-
pint5(s,:));
            d16=norm(pint1(g,:)-
pint6(s,:));
            if or(d13<e,d14<e)
                x(1,n) = 0;
                y(1,n) = 0;
                z(1,n) = 0;
            end
            if or(d15<e,d16<e)
                x(1,n) = 0;
                y(1,n) = 0;
                z(1,n) = 0;
            end
        end
    end
end
```

V. CONCLUSIONES

El código desarrollado ofrece una herramienta eficaz para determinar la mejor combinación de longitudes de eslabones para un robot colaborativo, considerando las posibles posiciones del efector final.

Sin embargo, cabe destacar que este no es el único criterio que debe tenerse en cuenta para la elección de las longitudes, por lo que se sugiere ponderar los resultados obtenidos del análisis con otros factores intervinientes.

Además, con modificaciones menores en el código, se puede llevar a cabo un análisis comparativo entre robots colaborativos con distinta cantidad de eslabones, o con una combinación diferente de movimientos de rotación y doblado.

No obstante, sin conocimientos básicos de programación puede resultar desafiante, por lo que se propone como mejora a futuro la modificación del código, para que adaptarlo a los diferentes casos resulte más accesible.

REFERENCIAS

[1] A. Barrientos, L. P. Peñín, C. Balaguer, y R. Aracil, "Fundamentos de Robótica", 2da ed., Universidad Politécnica de Madrid, 1997.

```
    for a = 1:length(pint2)
        for o = 1:length(pint4)
            d24=norm(pint2(a,:)-
pint4(o,:));
            d25=norm(pint2(a,:)-
pint5(o,:));
            d26=norm(pint2(a,:)-
pint6(o,:));
            if (d24<e||d25<e)
                x(1,n) = 0;
                y(1,n) = 0;
                z(1,n) = 0;
            end
            if d26<e
                x(1,n) = 0;
                y(1,n) = 0;
                z(1,n) = 0;
            end
        end
    end
    for u = 1:length(pint3)
        for r = 1:length(pint5)
            d35=norm(pint3(u,:)-
pint5(r,:));
            d36=norm(pint3(u,:)-
pint6(r,:));
            if (d35<e || d36<e)
                x(1,n) = 0;
                y(1,n) = 0;
                z(1,n) = 0;
            end
        end
    end
    for w = 1:length(pint4)
        for q = 1:length(pint6)
            d46=norm(pint4(w,:)-
pint6(q,:));
            if d46<e
                x(1,n) = 0;
                y(1,n) = 0;
                z(1,n) = 0;
            end
        end
    end
end
end
end
end
plot3(x,y,z, '.')
axis([inf inf inf inf 0 inf])
hold on
grid on
```