ACTIVIDAD 2: ANÁLISIS DEL ROBOT LINEAL DE 3GDL

Víctor Manuel Vázquez Morales A01736352

OBJETIVO:

Obtener las velocidades lineales y angulares de un robot lineal de tres grados de libertad.

ROBOT LINEAL DE 3GDL:

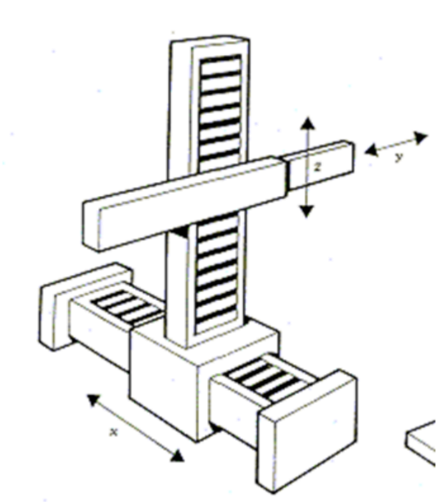

Figura 1: Robot cartesiano de 3 gdl

Para esta actividad analizaremos la cinemática de un robot lineal de tres grados de libertad, lo que nos permitirá obtener las velocidades lineales y angulares para posteriormente poder tener la posibilidad de predecir la posición, estado y comportamiento del robot.

El sistema que analizaremos es el mostrado en la figura 1, el cuál es un robot lineal de tipo cartesiano.

Para obtener las velocidades implementamos el siguiente análisis y código de MATLAB:

SCRIPT

```
%Limpieza de pantalla
clear all
close all
clc
```

Declaración de variables simbólicas. El sistema que analizaremos involucra tres variables, las cuales son las longitudes de cada una de las articulaciones:

```
%Declaración de variables simbólicas
syms l1(t) l2(t) l3(t) t
```

Configuración del robot. Dado que nuestro robot se compone de tres juntas prismáticas, agregaremos tres unos al vector RP:

```
RP=[1 1 1];
```

Coordenadas articulares. Creamos un vector de coordenadas articulares, el cual contendrá las variables de las que depende nuestro sistema:

```
Q= [l1, l2, l3];
disp('Coordenadas generalizadas');
pretty (Q);
```

Velocidades generalizadas. Obtenemos el vector de velocidades generalizadas diferenciando el vector de coordenadas articulares:

```
Qp= diff(Q, t);
disp('Velocidades generalizadas');
pretty (Qp);
```

Respuesta de código de MATLAB

```
Coordenadas generalizadas
(l1(t), l2(t), l3(t))

Velocidades generalizadas
/  d         d         d       \
| -- l1(t), -- l2(t), -- l3(t) |
\ dt        dt        dt       /
```

El resultado es justo el esperado. Las coordenadas generalizadas de nuestro robot dependen de las variables l1, l2 y l3, las cuáles hacen referencia al desplazamiento existente en cada una de las articulaciones. Por otro lado, nuestro vector de velocidades generalizadas es justamente el cambio del vector de coordenadas con respecto al tiempo.

Grados de libertad. Almacenamos en una variable los grados de libertad del robot, ya que será útil para realizar cálculos más adelante:

```
GDL= size(RP,2);
GDL_str= num2str(GDL);
```

Matrices de transformación homogénea. La primera parte será obtener la matriz de transformación homogénea en el efector final. Para este caso, nuestro marco de referencia inercial será el siguiente:

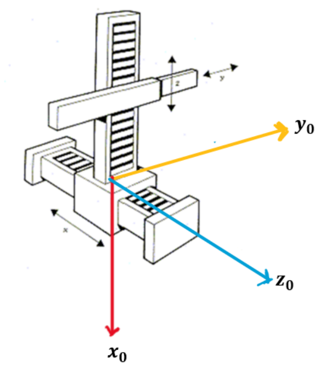

Figura 2. Marco de referencia inercial

Notemos de hecho, que el eje z del marco de referencia inercial está alineado con la primera junta y siempre colocaremos este eje en dirección del movimiento de cada articulación.

Articulación 1:

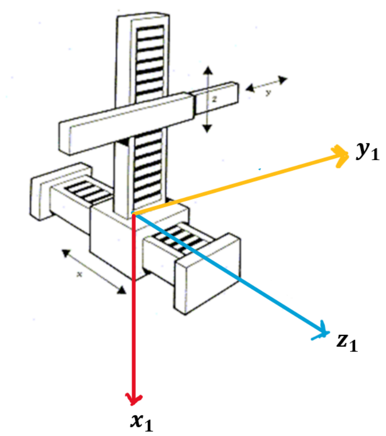

Figura 3. Articulación 1

Para este primer caso, dado que el marco de referencia (y por tanto su marco de referencia local) está alineado con la primera articulación, podemos proceder a declarar la posición de esta primera articulación que, acorde a la referencia, se desplaza sobre el eje z:

```
%Articulación 1
%Posición de la articulación 1 respecto a 0
P(:,:,1)= [0 ; 0; l1];
```

Ahora bien, para alinear la siguiente articulación con el eje z, será necesario rotar el sistema de referencia local negativamente 90 grados con respecto a y1. Sustituyendo este valor en la matriz de rotación para este eje, tenemos que:

```
%Matriz de rotación de la junta 1 respecto a 0.... -90° y1
R(:,:,1)= [0   0 -1 ;
           0   1  0;
           1   0  0];
```

Articulación 2:

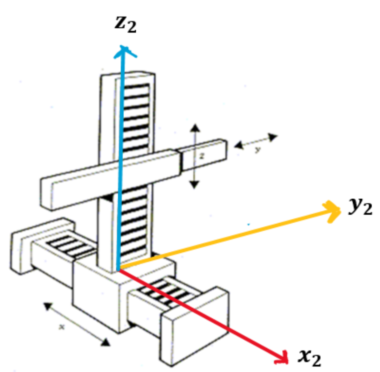

Figura 4. Articulación 2

En esta segunda articulación, hemos alineado ya previamente la junta 2 con el eje z. Acorde a su propio sistema de referencia, existe un desplazamiento en el eje z a lo largo de la longitud de esta articulación:

```
%Articulación 2
%Posición de la articulación 2 respecto a 1
P(:,:,2)= [0; 0 ; l2];
```

Para alinear la articulación siguiente con z, tendremos que rotar el marco de referencia local -90 grados con respecto a x2. Sustituyendo en la matriz de rotación del eje x tenemos que:

```
%Matriz de rotación de la junta 2 respecto a 1  -90° x2
R(:,:,2)= [1  0 0;
           0  0 1;
           0 -1 0];
```

Articulación 3:

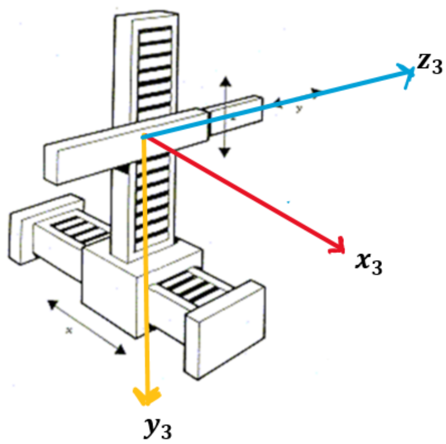

Figura 5. Articulación 3

Para esta última junta, tenemos el desplazamiento de la articulación nuevamente con respecto al eje z:

```
%Articulación 3
%Posición de la articulación 3 respecto a 2
P(:,:,3)= [0; 0 ;13];
```

Dado que no hay una siguiente articulación, no es necesario rotar el marco de referencia, por tanto, declaramos la matriz identidad:

```
%Matriz de rotación de la junta 3 respecto a 2
R(:,:,3)= [1  0 0;
           0  1 0;
           0  0 1];
```

Construcción de matrices. Procedemos a declarar algunos vectores y matrices que nos servirán para almacenar información de nuestro interés, tales como las matrices de transformación homogénea globales y locales:

```
%Creamos un vector de ceros
Vector_Zeros= zeros(1, 3);
%Inicializamos las matrices de transformación Homogénea locales
A(:,:,GDL)=simplify([R(:,:,GDL) P(:,:,GDL); Vector_Zeros 1]);
%Inicializamos las matrices de transformación Homogénea globales
T(:,:,GDL)=simplify([R(:,:,GDL) P(:,:,GDL); Vector_Zeros 1]);
%Inicializamos las posiciones vistas desde el marco de referencia inercial
PO(:,:,GDL)= P(:,:,GDL);
%Inicializamos las matrices de rotación vistas desde el marco de referencia
inercial
RO(:,:,GDL)= R(:,:,GDL);
```

Posteriormente, utilizando un ciclo for calculamos las matrices de transformación:

```
for i = 1:GDL
   i_str= num2str(i);
  %disp(strcat('Matriz de Transformación local A', i_str));
   A(:,:,i)=simplify([R(:,:,i) P(:,:,i); Vector_Zeros 1]);
  %pretty (A(:,:,i));
  %Globales
   try
      T(:,:,i)= T(:,:,i-1)*A(:,:,i);
   catch
      T(:,:,i)= A(:,:,i);
   end
   disp(strcat('Matriz de Transformación global T', i_str));
   T(:,:,i)= simplify(T(:,:,i));
   pretty(T(:,:,i))
   RO(:,:,i)= T(1:3,1:3,i);
   PO(:,:,i)= T(1:3,4,i);
   %pretty(RO(:,:,i));
   %pretty(PO(:,:,i));
end
```

Respuesta de código de MATLAB

No hay mucha ciencia en el caso de las matrices de transformación locales, ya que únicamente es la concatenación de lo que ya hemos analizado. Sin embargo, lo interesante está en las matrices de transformación globales:

```
Matriz de Transformación global T1
/ 0, 0, -1,    0   \
|                  |
| 0, 1,  0,    0   |
|                  |
| 1, 0,  0, l1(t)  |
|                  |
\ 0, 0,  0,    1   /
```

Para el caso de la primera matriz, la transformación global es exactamente igual a la de transformación local, ya que no se ve afectada por articulaciones o juntas previas.

Ahora bien, tanto para la segunda como tercera articulación, el análisis debe realizarse considerando el marco de referencia inercial:

```
Matriz de Transformación global T2
/ 0, 1, 0, -l2(t) \
|                 |
| 0, 0, 1,    0   |
|                 |
| 1, 0, 0,  l1(t) |
|                 |
\ 0, 0, 0,    1   /
```

Con respecto al marco de referencia inercial, la segunda articulación se encuentra en la parte negativa del eje x, lo que explica el valor de p_x (un desplazamiento negativo de -l2(t)). Bajo esta misma referencia, existe un desplazamiento de l1 en el eje z, haciendo sentido así el valor de p_z.

```
Matriz de Transformación global T3
/ 0, 1, 0, -l2(t) \
|                 |
| 0, 0, 1,  l3(t) |
|                 |
| 1, 0, 0,  l1(t) |
|                 |
\ 0, 0, 0,    1   /
```

Ahora bien, para la última articulación, si analizamos con respecto al marco de referencia inercial tenemos que:

- La articulación 2 se encuentra en el eje x, y se desplaza en el sentido contrario del eje. $(p_x = -l2(t))$

- La articulación 3 se encuentra en el eje y, y se desplaza en el sentido del eje ya mencionado. $(p_y = l1(t))$

- Por último, la articulación 1 se encuentra alineada con el eje z, y se desplaza a lo largo de este eje. $(p_z = l1(t))$

Jacobiano. Utilizando el método analítico, realizamos el cálculo del jacobiano, el cuál nos servirá para obtener las velocidades lineales y angulares. Para este caso, tomaremos en cuenta las siguientes fórmulas (para articulaciones prismáticas):

- Jacobiano lineal. Es la tercer columna de la matriz de rotación previa:

$$Jv_i = Z_{i-1}$$

- Jacobiano angular. Es un vector de ceros:

$$J\omega_i = \vec{0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

```
%Calculamos el jacobiano lineal de forma analítica
Jv_a(:,GDL)=PO(:,:,GDL);
Jw_a(:,GDL)=PO(:,:,GDL);
for k= 1:GDL
   if RP(k)==0
      %Para las juntas de revolución
       try
           Jv_a(:,k)= cross(RO(:,3,k-1), PO(:,:,GDL)-PO(:,:,k-1));
           Jw_a(:,k)= RO(:,3,k-1);
       catch
             Jv_a(:,k)= cross([0,0,1], PO(:,:,GDL));%Matriz de rotación de 0
con respecto a 0 es la Matriz Identidad, la posición previa tambien será 0
            Jw_a(:,k)=[0,0,1];%Si no hay matriz de rotación previa se obtiene
la Matriz identidad
        end
    else
%         %Para las juntas prismáticas
       try
           Jv_a(:,k)= RO(:,3,k-1);
       catch
           Jv_a(:,k)=[0,0,1];
       end
           Jw_a(:,k)=[0,0,0];
    end
```

```
end
Jv_a= simplify (Jv_a);
Jw_a= simplify (Jw_a);
%disp('Jacobiano lineal obtenido de forma analítica');
%pretty (Jv_a);
%disp('Jacobiano ángular obtenido de forma analítica');
%pretty (Jw_a);
```

Velocidades lineales y angulares. Finalmente, utilizando el jacobiano, podemos calcular las velocidades:

```
disp('Velocidad lineal obtenida mediante el Jacobiano lineal');
V=simplify (Jv_a*Qp');
pretty(V);
disp('Velocidad angular obtenida mediante el Jacobiano angular');
W=simplify (Jw_a*Qp');
pretty(W);
```

Respuesta de código de MATLAB

```
Velocidad lineal obtenida mediante el Jacobiano lineal
/    ________  \
|     d        |
| - -- l2(t)   |
|    dt        |
|              |
|   ________   |
|    d         |
|   -- l3(t)   |
|   dt         |
|              |
|   ________   |
|    d         |
|   -- l1(t)   |
\   dt         /

Velocidad angular obtenida mediante el Jacobiano angular
/ 0 \
|   |
| 0 |
|   |
\ 0 /
```

Analicemos los resultados obtenidos. Notemos que para el caso de la velocidad lineal, nos indica que en el efector final...

1. La velocidad en x es la diferencial de -l2 con respecto al tiempo. Esto debido a que esta articulación se encuentra en la parte negativa del eje x según el marco de referencia inercial, lo que implica un desplazamiento negativo.

2. Para la velocidad en y, esta es equivalente a la diferencial de l3 con respecto al tiempo. Como te podrás imaginar, esto se debe a que esta articulación se encuentra alineada con el eje y según el marco de referencia inercial.

3. Finalmente, la velocidad en z es la diferencial de l1 con respecto al tiempo y si, se debe a que esta articulación se desplaza en el eje z según el marco de referencial inercial.

Finalmente, notemos que la velocidad angular es nula para cada uno de los ejes. Esto debido a que se trata de un robot prismático que no rota en ninguno de los ejes.