Instituto Tecnológico de Estudios Superiores de Monterrey

Campus Puebla

Fundamentación de Robótica (Gpo 101)

Actividad 2 (Análisis de Robot lineal de 3GDL)

Alumno

José Diego Tomé Guardado A01733345

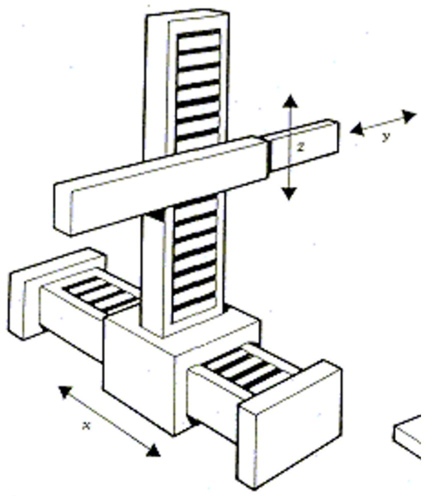
Fecha de entrega

Martes 27 de Febrero de 2024

Diseño de un robot cartesiano 3GDL

Para el diseño de un robot cartesiano se tiene una estructura con un base que se observa en la imagen qué soporta un brazo tanto horizontal como uno vertical.

Cada uno de los ejes tiene un movimiento en específico para que podamos obtener correctamente cómo funcionan cada una de las articulaciones que se tienen en el movimiento de este robot, además de finalmente como resultado de la actividad poder hallar las velocidades lineales y angulares.



Código de Matlab

Tenemos primero los comandos para poder realizar la limpieza de la pantalla y de las variables también.

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

Tenemos la definición de las variables simbólicas donde `l1(t)`, `l2(t)` y `l3(t)` son la representación de las posiciones angulares de las articulaciones en función del tiempo.

También vamos a tener en cuenta la configuración del robot con el vector de `RP` en que pondremos un 1 porque las juntas con las que cuenta nuestro robot son prismáticas.

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

%Configuración del robot, 0 para junta rotacional, 1 para junta prismática
RP=[1 1 1];
```

`Q` se define como nuestro vector de coordenadas articulares que contiene las posiciones angulares del robot que dependen del movimiento de este mismo. `l1(t), l2(t), l3(t)` son estas mismas variables de las que nuestro robot depende.

```
%Creamos el vector de coordenadas articulares
Q= [l1, l2, l3];
disp('Coordenadas generalizadas');
pretty (Q);
```

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

Es importante tomar en cuenta las velocidades articulares para poder llegar al resultado esperado, `Qp` se va a calcular como la derivada temporal de `Q` con respecto al tiempo de *t* lo qué va a representar las velocidades generalizadas.

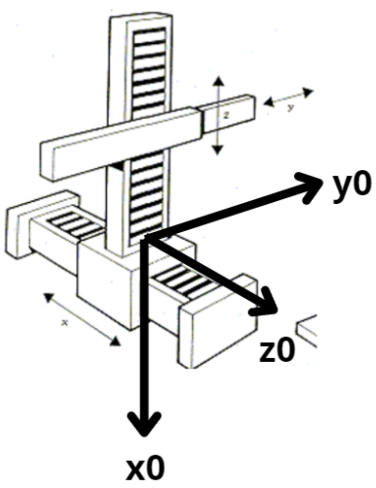
Se representan las velocidades angulares de las articulaciones, que van a depender de la derivada con respecto al tiempo de `l1(t), l2(t)` y `l3(t)`

```
%Creamos el vector de velocidades generalizadas
Qp= diff(Q, t);
disp('Velocidades generalizadas');
pretty (Qp);

%Número de grado de libertad del robot
GDL= size(RP,2);
GDL_str= num2str(GDL);
```

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

Análisis por articulaciones

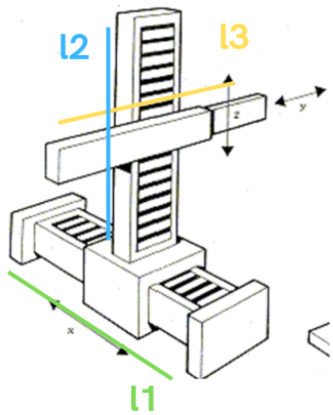


Primeramente vamos a tener en cuenta un análisis para todas las articulaciones con las que cuenta nuestro robot, aunque podemos hacer más matrices de rotación, decidimos dejarlo definido en 3.

Debido a que nos guiamos también por el número de grados de libertad y sobre todo para que sean más entendibles las rotaciones para poder llegar a un correcto desenlace de todo el movimiento del robot.

Comenzaremos con nuestro marco referencia inercial de esta forma como se observa en la imagen, en el que podemos destacar que tendremos en `x0 -> l2, z0 -> l1` y en `y0 -> l3.`

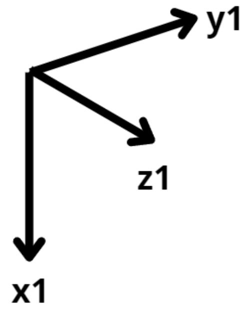
Podemos decir que la posición de las articulaciones va a depender de nuestro marco de referencia, de tal manera que haciendo un pequeño diagrama para guiarme y entender mejor, es así como quedarán determinadas las articulaciones y con ello podremos entender más sobre en qué eje se encuentran en cada una de las rotaciones o traslaciones.



Articulación 1

Tenemos que representar la posición de la articulación 1 con respecto a la base del robot que es 0, para este caso se va a definir que la articulación 1 `(l1)` se encuentra en el `eje z`. Posteriormente se crea la matriz de rotación de la junta 1 lo cual esta matriz va a indicar una rotacion de -90 grados alrededor de nuestro `eje y1` y se desplaza por el `eje z`.

```
%Articulación 1
%Posición de la articulación 1 respecto a 0
P(:,:,1)= [0 ; 0; l1];
%Matriz de rotación de la junta 1 respecto a 0.... -90° y1
R(:,:,1)= [0   0 -1 ;
           0   1  0;
           1   0  0];
```



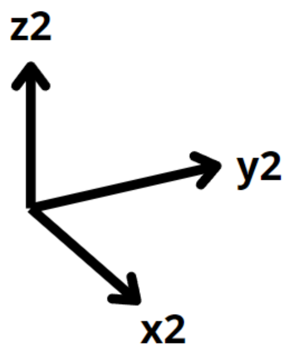
Para la matriz de rotación qué tenemos donde tenemos un giro de -90 grados alrededor del `eje y1` donde vamos a sustituir los -90 grados en el ángulo en la matriz de rotación Ry:

$$R_y(\theta) = \begin{pmatrix} \cos\theta & 0 & \operatorname{sen}\theta \\ 0 & 1 & 0 \\ -\operatorname{sen}\theta & 0 & \cos\theta \end{pmatrix}$$

Articulación 2

Ahora teniendo en cuenta nuestro nuevo marco de referencia, vamos a indicar una rotación ahora de -90 grados pero alrededor del `eje x2,` no contamos con ningún desplazamiento porque lo haremos en la articulación 3 junto.

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



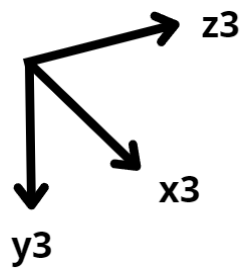
Para la matriz de rotación qué tenemos donde tenemos una rotación de -90 grados alrededor del `eje x2` donde vamos a sustituir los -90 grados en el ángulo en la matriz de rotación Rx:

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\operatorname{sen}\theta \\ 0 & \operatorname{sen}\theta & \cos\theta \end{pmatrix}$$

Articulación 3

Finalmente tenemos la posición de la articulación 3 `(l3)` con el marco de referencia podemos ver que ahora está con respecto a `z` y correspondiente a la articulación 2 `(l2)` tenemos un desplazamiento en el `eje y` negativamente, es por ello en este caso que pondremos `l2` negativo debido a que está en la posición pero contraria a como se encontraría en el `eje y` con respecto a la articulación `(l2)`. También no hay una siguiente articulación por lo tanto no es necesario ya rotar nuestro marco de referencia.

```
%Articulación 3
%Posición de la articulación 3 respecto a 2
P(:,:,3)= [0; -l2 ;l3];
%Matriz de rotación de la junta 3 respecto a 2
R(:,:,3)= [1  0 0;
           0  1 0;
           0  0 1];
```



En este caso definimos la matriz de identidad ya que como se mencionó anteriormente, la articulación 3 no va a tener una rotación adicional porque no contamos con más articulaciones.

$$I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Para la construcción de nuestras matrices vamos a tener que tener qué inicializar las de transformacion homogenea local y global que van a concatenar a la matriz de rotación, la matriz de posición junto también con el vector de ceros seguido de un 1. Por otro lado tendremos inicializados los vectores donde se copia la matriz de posición y rotación.

```
%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);
```

Matrices de transformación

Para poder construirlas vamos a utilizar un ciclo for para poder ir iterando para las matrices de transformación local y también utilizar el uso de try y catch se calcula la matriz de transformación global con el uso de la recursividad.

```
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
```

Ejecución

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

Como ya sabemos la matriz de transformación global será igual que la local debido a que no hay más articulaciones involucradas, sino que es el inicio con nuestra primera articulación `(l1)` .

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

Tomando en cuenta el marco de referencia podemos observar en esta matriz `T2` que también solo va a estar la articulación `(l1)` ya que no declare en la posición a otra articulación, solo teniendo el desplazamiento de `l1(t)` en el `eje 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    /
```

Para la última matriz de transformación global ya tendremos las 3 articulaciones debido a que se juntan por ello la articulación 2 se encuentra en el `eje x` con un desplazamiento pero contrario al eje, por eso se observa que es negativo y la articulación 3 con desplazamiento en `eje y` finalmente en el `eje z` se desplaza la articulación 1.

Resultados de las velocidades lineales y angulares

Finalmente con ayuda del jacobiano que ya sabemos cómo se calcula gracias a la actividad anterior, podemos darnos cuenta de que podremos obtener los resultados de las velocidades del producto del jacobiano lineal `Jv_a` o del jacobiano angular `Jw_a` y las velocidades articulares `Qp`.

```
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);
```

Ejecución

```
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 /
```

Analizando los resultados podemos darnos cuenta de que la velocidad para el `eje x` quedó en la derivada de `-l2(t)` lo cual es correcto ya que como pudimos observar a lo largo de las rotaciones y el marco de referencia inercial que esta articulación se va a encontrar como un desplazamiento negativo, en la parte negativa de nuestro `eje x`.

La velocidad en y es la derivada con respecto al tiempo de `l3(t)`, se encuentra en y porque está alineada con este eje y finalmente para la velocidad en `z` es la derivada con respecto al tiempo de `l1(t)` encontrándose qué se desplaza por este `eje z`. Por otro lado, la velocidad angular es de cero en todos los ejes porque estamos hablando de un robot con juntas prismáticas, lo qué significa qué no existe una rotación para ninguno de los ejes `(x,y,z)`.