# Actividad 2: Análisis de robot lineal con 3 grados de libertad

**Fundamentación de Robótica**

Fernando Estrada Silva // A01736094

**Objetivo**

Obtener el vector de velocidades lineal y angular simulados en MATLAB para la siguiente configuración:

**Introducción**

Para esta actividad, se espera obtener la velocidad angular y lineal de un robot de 3 grados de libertad, de manera similar a la entrega realizada en la actividad 1. Sin embargo, la mayor diferencia para esta actividad es el uso de un robot lineal. Esta caracteristica implica que el robot utiliza juntas prismáticas, las cuales permiten el desplazamiento lineal relativo entre las piezas de un robot, siendo para este caso eslabones. Un ejemplo de este tipo de juntas es el funcionamiento de un cajón que se mueve hacia adentro y hacia fuera de sobre un mueble fijo. Entonces, se forma una junta prismática.

Del otro lado, en lugar de considerar un cambio con respecto a un ángulo theta, se tiene un cambio con respecto a la longitud de los eslabones a lo largo del tiempo. También, se agrega un nuevo grado de libertad al robot, aumentando significativamente el comportamiento del sistema.

**Metodología**

Como se mencionó anteriormente, el programa inicia declarando las variables simbólicas correspondientes (longitud de eslabones y tiempo). Dadas las juntas prismáticas, se define el arreglo con 1 en RP.

```
syms l1(t) l2(t) l3(t) t
RP=[1 1 1];
```

Las coordenadas generalizadas definen la posición y orientación para cada eslabón del robot. Del otro lado, las velocidades generalizadas, representan las tasas de cambio de las coordenadas generalizadas conforme el sistema se mueve.

```
Q= [l1, l2, l3];
Qp= diff(Q, t);
```

Posteriormente, se indican en un nuevo arreglo los grados de libertad. A su vez, se plantea el análisis para cada una de las articulaciones del sistema. Será necesario modificar el sistema de referencia de forma en que el eje z siempre debe colocarse en dirección el movimiento de la articulación.

```
%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.... -90º y1
R(:,:,1)= [0  0 -1 ;
```

```
            0  1  0;
            1  0  0];
```

Para la segunda articulación, se vuelve a plantear un nuevo sistema de referencia. La matriz de rotación de la junta 2 respecto a 1, se rota -90° con respecto a y1.

```
%Articulación 2
%Posición de la articulación 2 respecto a 1
P(:,:,2)= [0; l3 ; 0];
%Matriz de rotación de la junta 2 respecto a 1  -90º y1
R(:,:,2)= [1  0 0;
           0  0 1;
           0 -1 0];
```

Finalmente, la tercera articulación:

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

Del otro lado, se inicializan las matrices de transformación homogenea local (A), transformación homogénea global (T), matriz de posición (PO) y de rotación (RO).

```
%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);
```

El siguiente ciclo for únicamente funciona para imprimir los resultados de las matrices de forma simplificada. El proceso se realiza en función del numero de grados de libertad.

```
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
```

Después de incializar las matrices para jacobianos lineal y angular, comienza un nuevo ciclo for que enumera cada articulacióin y calcula dicho jacobiano de forma analítica. Se simplifican las matrices al final.

```
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);x
```

Finalmente, para obtener las velocidades lineales y angulares, se obtiene el producto de la tasa de cambio de las velocidades generalizadas y el jacobiano correspondiente:

```
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);
```

**Resultados**

En orden de mención, se presentan los resultados obtenidos del programa:

La matriz de transformación global T1 es equivalente a la matriz de transformación local, pues no hay modificaciones por juntas anteriores. Las demás transformaciones implican cambios en el sistema de referencia inercial.

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

```
Matriz de Transformación global T2
/ 0, 1, 0,    0   \
|                 |
| 0, 0, 1, l3(t)  |
|                 |
| 1, 0, 0, l1(t)  |
|                 |
\ 0, 0, 0,    1   /
```

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

Como se esperaba al inicio, existe un cambio de longitud con respecto al tiempo para cada eslabón (siendo 3 en total), para l1, l2 y l3 respectivamente. Por ejemplo, para la velocidad en x, la articulación se encuentra en un cuadrante negativo del sistema de referencia inercial, obteniendo un desplazamiento negativo.

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

Se obtiene que la velocidad angular del sistema es 0 para toda dimensión. Esto tiene sentido ya que como se mencionó al inicio del programa, la junta es prismática, lo que implica que no haya rotación.

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