

CONTROL DE TRAYECTORIA DE UN ROBOT ESFERICO DE TRES GRADOS DE LIBERTAD

Integrantes:

Renato Masias Galarza
Alex Ramón Castillo
Rony Rivas Quispe
Leonel Orco Castillo
Felipe Gómez Fernandez
Javier Morales Cabello

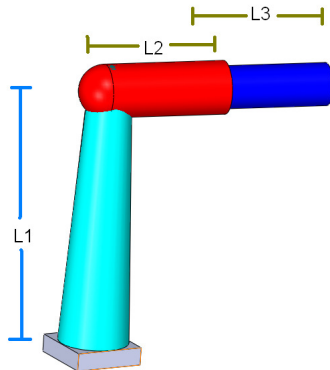
Profesor: Ing. Anchayhua Arestegui, Nilton Cesar

Facultad de Ingeniería Mecánica

ABSTRACT

Este artículo describe el seguimiento de trayectoria y respuesta ante perturbaciones mediante el control monoarticular y multiarticular de un Robot Esférico de tres grados de libertad, así como el cálculo de su cinemática y dinámica. Se usa la herramientas de Simulink y MatLab como software de simulación del motor, el robot y el controlador.

ESQUEMA DEL ROBOT



Sus parámetros son:

$$L_1 = 0.8m$$

$$L_2 = 0.6m$$

$$L_3 = 0.6m$$

$$m_1 = 10kg$$

$$m_2 = 2kg$$

$$m_3 = 0.2kg$$

Los Centros de Masa están ubicados en el centro de cada eslabón.

MARCO TEORICO

1) Cálculo de la Dinámica Directa

Los parámetros DH son:

	a_k	α_k	d_k	θ_k
1	0	$\pi/2$	L1	q_1
2	0	$\pi/2$	0	$q_2 + \pi/2$
3	0	0	r	0

Con esta información se procede a calcular los parámetros necesarios para obtener la dinámica directa, es decir obtener T_1 , T_2 , y F_3 .

Las matrices de transformación serán:

```
%matrices DH
T10=[cos(q1) 0 sin(q1) 0
      sin(q1) 0 -cos(q1) 0
      0 1 0 L1
      0 0 0 1]
T21=[-sin(q2) 0 cos(q2) 0
      cos(q2) 0 sin(q2) 0
      0 1 0 0
      0 0 0 1]
T32=[1 0 0 0
      0 1 0 0
      0 0 1 r
      0 0 0 1]
T20=T10*T21
T30=T10*T21*T32
```

```
T10 =
[ cos(q1), 0, sin(q1), 0]
[ sin(q1), 0, -cos(q1), 0]
[ 0, 1, 0, L1]
[ 0, 0, 0, 1]
T21 =
```

```
[ -sin(q2), 0, cos(q2), 0]
[ cos(q2), 0, sin(q2), 0]
[ 0, 1, 0, 0]
[ 0, 0, 0, 1]
```

```
T32 =
[ 1, 0, 0, 0]
[ 0, 1, 0, 0]
[ 0, 0, 1, r]
[ 0, 0, 0, 1]
```

```
T20 =
[ -cos(q1)*sin(q2), sin(q1), cos(q1)*cos(q2), 0]
[ -sin(q1)*sin(q2), -cos(q1), sin(q1)*cos(q2), 0]
[ cos(q2), 0, sin(q2), L1]
[ 0, 0, 0, 1]

T30 =
[ -cos(q1)*sin(q2), sin(q1), cos(q1)*cos(q2), cos(q1)*cos(q2)*r]
[ -sin(q1)*sin(q2), -cos(q1), sin(q1)*cos(q2), sin(q1)*cos(q2)*r]
[ cos(q2), 0, sin(q2), sin(q2)*r+L1]
[ 0, 0, 0, 1]
```

Posición de los CMk respecto a su propio sistema XkYkZk

```
c1=[0; -L1/2; 0]
c2=[0; 0; L2/2]
c3=[0; 0; -L3/2]
```

Momentos de Inercia, se considera a los eslabones como cilindros huecos:

```
I1=m1*L1^2/12*[1 0 0; 0 0 0; 0 0 1]
I2=m2*L2^2/12*[1 0 0; 0 1 0; 0 0 0]
I3=m3*L3^2/12*[1 0 0; 0 1 0; 0 0 0]
```

Tensores inerciales 3x3 del eslabón k respecto a XoYoZo y trasladado a su Centro de Masa

```
R10=[T10(1:3,1:3)];
R20=[T20(1:3,1:3)];
R30=[T30(1:3,1:3)];
D1=simple(R10*I1*transpose(R10))
D2=simple(R20*I2*transpose(R20))
D3=simple(R30*I3*transpose(R30))
```

Posición de los CM respecto a XoYoZo

```
C1=T10(1:3,4)+R10*c1
C2=T20(1:3,4)+R20*c2
C3=T30(1:3,4)+R30*c3
```

```
C1 =
0
0
1/2*L1
C2 =
1/2*cos(q1)*cos(q2)*L2
1/2*sin(q1)*cos(q2)*L2
L1+1/2*sin(q2)*L2
C3 =
cos(q1)*cos(q2)*r-1/2*cos(q1)*cos(q2)*L3
sin(q1)*cos(q2)*r-1/2*sin(q1)*cos(q2)*L3
sin(q2)*r+L1-1/2*sin(q2)*L3
```

Jacobianas de velocidad lineal y angular respecto a los Centros de Masa CM.

```
Jv=[diff(T30(1:3,4),q1) diff(T30(1:3,4),q2) diff(T30(1:3,4),r)]
Jinv=simple(inv(Jv))
Jv1=[diff(C1,q1) diff(C1,q2) diff(C1,r)]
Jv2=[diff(C2,q1) diff(C2,q2) diff(C2,r)]
Jv3=[diff(C3,q1) diff(C3,q2) diff(C3,r)]
Jw1=[0 0 0; 0 0 0; 1 0 0]
Jw2=[0 sin(q1) 0; 0 -cos(q1) 0; 1 0 0]
Jw3=[0 sin(q1) 0; 0 -cos(q1) 0; 1 0 0]
```

Tensores inerciales k, describe el movimiento traslacional y rotacional del elemento k

```
d1=transpose(Jv1)*m1*Jv1+transpose(Jw1)*D1*Jw1
d2=simple(transpose(Jv2)*m2*Jv2+transpose(Jw2)*D2*Jw2)
d3=simple(transpose(Jv3)*m3*Jv3+transpose(Jw3)*D3*Jw3)
d=simple(d1+d2+d3)
```

Energía cinética total y energía potencia total

```
dq=[dq1; dq2; dr];
Ec=1/2*transpose(dq)*d*dq
Ep=[0 0 -g]*(m1*C1+m2*C2+m3*C3)
```

```
Ec = 1/6*dq1^2*cos(q2)^2*(m2*L2^2+3*m3*r^2-
3*m3*r*L3+m3*L3^2)+1/2*dq2^2*(1/3*m2*L2^2+m3*r^2-
m3*r*L3+1/3*m3*L3^2)+1/2*dr^2*m3
Ep = g*(1/2*m1*L1+m2*(L1+1/2*sin(q2)*L2)+m3*(sin(q2)*r+
L1-1/2*sin(q2)*L3))
```

El Lagrangiano será entonces la resta de la energía cinética y la potencial, dando entonces:

```
L=1/6*dq1^2*cos(q2)^2*(m2*L2^2+3*m3*r^2-
3*m3*r*L3+m3*L3^2)+1/2*dq2^2*(1/3*m2*L2^2+m3*r^2-
m3*r*L3+1/3*m3*L3^2)+1/2*dr^2*m3-
g*(1/2*m1*L1+m2*(L1+1/2*sin(q2)*L2)+m3*(sin(q2)*r+L1-
1/2*sin(q2)*L3))
```

Los torques y la fuerza se calculan usando el Lagrangiano mediante la formulación de Lagrange, esto es:

$$\frac{T_i}{F_i} = \frac{d}{dt} \cdot \frac{\partial}{\partial \dot{q}_i} L_{(q(t), \dot{q}(t))} - \frac{\partial}{\partial q_i} L_{(q(t), \dot{q}(t))}$$

Pero en MatLab se usa el método computacional donde no se deriva respecto al tiempo, el procedimiento para los dos torques y la fuerza es el siguiente:

```
%Torques y fuerzas
%-----
q=[q1; q2; r];
ddq=[ddq1; ddq2; ddr];
m=[m1; m2; m3];
C=[C1 C2 C3];
%Torque 1
%-----
DD1=d(1,1:3)
cor1=0; h1=0;
for k=1:3
```

```

for j=1:3
    cor1=simple(cor1+(diff(d(1,j),q(k))-
1/2*diff(d(k,j),q(1)))*dq(k)*dq(j));
end
cor1
h1=simple(-[0 0 -
g]*(m1*diff(C1,q(1))+m2*diff(C2,q(1))+m3*diff(C3,q(1))))
T1=simple(DD1*ddq+cor1+h1)
%Torque 2
%-----
DD2=d(2,1:3)
cor2=0; h2=0;
for k=1:3
    for j=1:3
        cor2=simple(cor2+(diff(d(2,j),q(k))-
1/2*diff(d(k,j),q(2)))*dq(k)*dq(j));
    end
end
cor2
h2=simple(-[0 0 0 -
g]*(m1*diff(C1,q(2))+m2*diff(C2,q(2))+m3*diff(C3,q(2))))
T2=simple(DD2*ddq+cor2+h2)
%Fuerza 3
%-----
DD3=d(3,1:3)
cor3=0; h3=0;
for k=1:3
    for j=1:3
        cor3=simple(cor3+(diff(d(3,j),q(k))-
1/2*diff(d(k,j),q(3)))*dq(k)*dq(j));
    end
end
cor3
h3=simple(-[0 0 0 -
g]*(m1*diff(C1,q(3))+m2*diff(C2,q(3))+m3*diff(C3,q(3))))
F3=simple(DD3*ddq+cor3+h3)

```

El torque ejercido por el robot en su eje de articulación para el eslabón 1 es:

$$T1=1/3*cos(q2)^2*(m2*L2^2+3*m3*r^2-3*m3*r*L3+m3*L3^2)*ddq1-2/3*cos(q2)*(m2*L2^2+3*m3*r^2-3*m3*r*L3+m3*L3^2)*sin(q2)*dq1^2+1/3*cos(q2)^2*(6*m3*r-3*m3*L3)*dr*dq1$$

Para el eslabón 2 es:

$$T2=(1/3*m2*L2^2+m3*r^2-m3*r*L3+1/3*m3*L3^2)*ddq2+1/3*cos(q2)*(m2*L2^2+3*m3*r^2-3*m3*r*L3+m3*L3^2)*sin(q2)*dq1^2+(2*m3*r-m3*L3)*dr*dq2+1/2*g*cos(q2)*(m2*L2+2*m3*r-m3*L3)$$

La fuerza ejercida por el eslabón 3 hacia su eje de traslación es:

$$F3=m3*ddr-1/2*m3*(2*r-L3)*(cos(q2)^2*dq1^2+2*dq2^2)+g*m3*sin(q2)$$

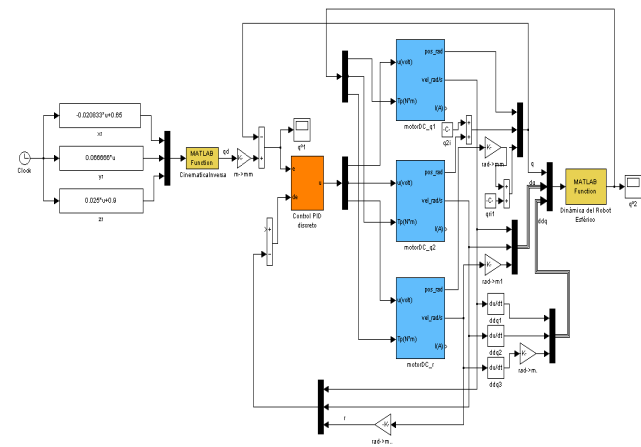
2) Control Monoarticular.

Consiste en controlar cada motor independientemente, sin tomar en cuenta las fuerzas y torques que los demás eslabones ejercen sobre cada motor.

En Simulink están los siguientes bloques de izquierda a derecha:

- Generador de trayectorias cartesianas, una línea recta en el espacio.
- Transformación de coordenadas cartesianas a articulares, usando la cinemática inversa
- Control PID discreto para cada motor.
- Modelamiento con bloques continuos para los tres motores.
- Modelamiento del robot por dinámica directa que produce los torques y fuerzas perturbables.

El esquema en Simulink es el siguiente:



2.1) Generador de Trayectorias Cartesianas.

El extremo del robot recorrerá una línea recta desde el punto A al punto B, siendo el tiempo de simulación de 12 segundos.

$$P_A = (0.65, 0, 0.9)m$$

$$P_B = (0.4, 0.8, 1.2)m$$

Cálculo de velocidades:

$$V_X = \frac{0.4 - 0.65}{12} = -0.020833m/s$$

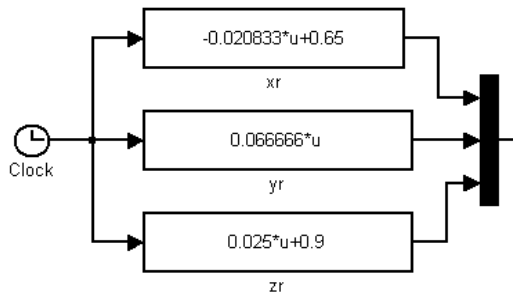
$$V_Y = \frac{0.8 - 0}{12} = 0.066666m/s$$

$$V_Z = \frac{1.2 - 0.9}{12} = 0.025m/s$$

Por tanto las ecuaciones que describen el movimiento en metros, aplicando la fórmula MRU distancia=velocidad*tiempo, son:

$$\begin{aligned} P_x &= -0.020833 * t + 0.65 \\ P_y &= 0.066666 * t \\ P_z &= 0.025 * t + 0.9 \end{aligned}$$

El bloque en simulink es:



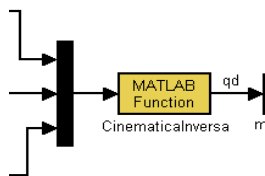
2.2) Transformación de Coordenadas Cartesianas a Articulares.

La referencia que le debe llegar a cada motor debe estar en coordenadas articulares, por tanto se debe emplear la cinemática inversa para pasar coordenadas en metros a rad para el motor 1 y 2 y milímetros para el motor 3.

Las ecuaciones que describen la cinemática inversa son obtenidas por métodos geométricos, estas son:

$$\begin{aligned} q_1 &= \arctan\left(\frac{P_y}{P_x}\right) \\ q_2 &= \arctan\left(\frac{P_z - L_1}{\sqrt{P_x^2 + P_y^2}}\right) \\ r &= \frac{P_z - L_1}{\sin(q_2)} \end{aligned}$$

El bloque MatLab Function llama a la función **Cinv.m** que calcula las coordenadas articulares en cada iteración.



2.3) Control PID discreto para cada motor

Los tres controladores fueron diseñados por el método del Lugar Geométrico de las Raíces. El PID discreto tiene la siguiente forma:

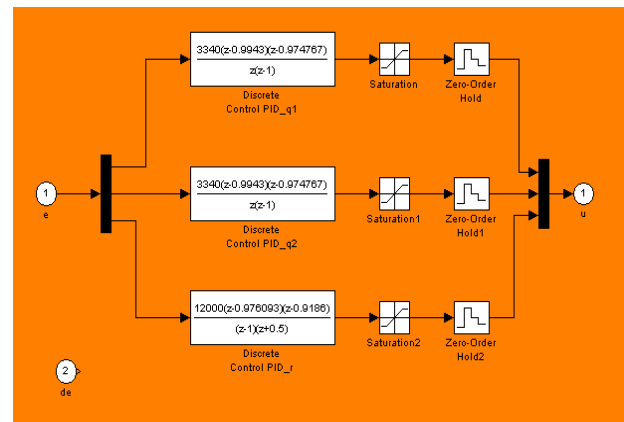
$$C_{(z)} = \frac{k_c * (z - a) * (z - b)}{z * (z - 1)}$$

Los parámetros de diseño escogidos para los tres motores son un tiempo de establecimiento de 0.2seg, un sobreimpulso máximo de 2% y un periodo de muestreo de 0.001seg. ante entradas escalón. Los controladores hallados son.

$$\begin{aligned} C_{1(z)} &= \frac{3340 * (z - 0.9943) * (z - 0.974767)}{z * (z - 1)} \\ C_{2(z)} &= \frac{3340 * (z - 0.9943) * (z - 0.974767)}{z * (z - 1)} \\ C_{3(z)} &= \frac{12000 * (z - 0.976093) * (z - 0.9186)}{z * (z - 1)} \end{aligned}$$

Los motores 1 y 2 reciben el error de posición en radianes y entregan voltaje, mientras que el motor 3 recibe el error en milímetros y entregan voltaje.

Cada control tiene su limitador de voltaje a +-100vdc y su ZOH con un tiempo igual al tiempo de muestreo de 0.001seg. Abriendo el bloque "PID discreto" de color naranja, tenemos:



2.4) Modelamiento de los motores DC

El motor de corriente continua empleado en cada articulación es:

Marca: Baldor
Voltaje=100v
TorqueNominal=4N*m
Potencia: 0.8Hp

Los motores de corriente continua fueron modelados con bloques de suma, multiplicación e integrales de acuerdo a sus parámetros principales. Reciben como entrada el voltaje en voltios y el torque perturbador en $N \cdot m$, y las salidas son la posición en radianes, la velocidad en radianes por segundo en el eje del engranaje (no del piñón) y la corriente. Los parámetros del motor son:

$$R_a = 2\Omega$$

$$L_a = 1.8mH$$

$$J = 8 \cdot 10^{-3} kg \cdot m^2$$

$$B = 4.94 \cdot 10^{-3} N \cdot m / (rad / s)$$

$$k_v = 0.3334 v / (rad / s)$$

$$k_t = 0.3334 N \cdot m / A$$

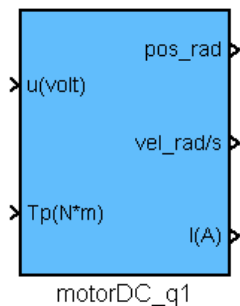
Cada motor está conectado a una caja reductora cuyos factores de engranaje son:

$$N_1 = 8$$

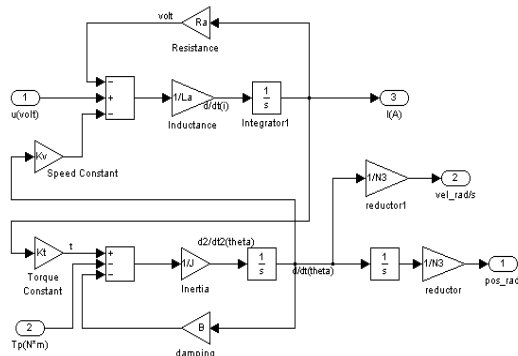
$$N_2 = 12$$

$$N_3 = 2$$

El bloque que usa simulink es este, se puede apreciar las entradas y salidas.



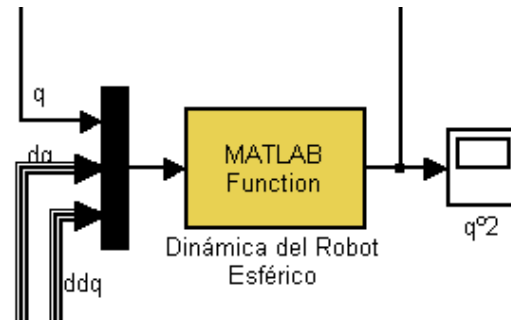
En el bloque interno se puede ver las ecuaciones que describen el movimiento de un motor, la de armadura y la de torque, en forma de bloques.



2.5) Modelamiento del Robot

La consigna de este bloque es obtener los torques perturbadores que produce el robot en cada articulación. Estos torques que produce el robot dependen de las posiciones, velocidades y aceleraciones que ocurren en cada articulación y en todo momento, por tanto se calculará la dinámica directa.

El bloque MatLab Function llama a la función DIN.m para el calculo de la dinámica directa.



El desarrollo de la dinámica directa se basa en la siguiente expresión:

$$T_i / F_i = H_{i(q)} * \ddot{q} + C_{i(q, \dot{q})} + h_{i(q)}$$

Donde:

- $H_{i(q)}$ es el tensor de inercial del eslabón i , por tanto el torque que produce la expresión $H_{i(q)} * \ddot{q}$ solo produce torque o fuerza cuando hay aceleraciones en los eslabones.
- $C_{i(q, \dot{q})}$ representa el torque o fuerza de Coriolis o centrípetas y ocurre cuando hay velocidades en las articulaciones.
- $h_{i(q)}$ es el torque o fuerza ejercida por la gravedad en cada articulación, esta actúa en todo instante y depende de la posición de cada eslabón del robot.

Se muestra el archivo DIN.m a continuación:

```
% tensores inerciales
H11=1/3*cos(q2)^2*(m2*L2^2+3*m3*r^2-
3*m3*r*L3+m3*L3^2);
H22=1/3*m2*L2^2+m3*r^2-m3*r*L3+1/3*m3*L3^2;
%H33=m3;
H33=I3;

% coriolis
cor1=-2/3*cos(q2)*(m2*L2^2+3*m3*r^2-
3*m3*r*L3+m3*L3^2)*sin(q2)*dq2*dq1+1/3*cos(q2)^2*(6*m
3*r-3*m3*L3)*dr*dq1;
cor2=1/3*cos(q2)*(m2*L2^2+3*m3*r^2-
3*m3*r*L3+m3*L3^2)*sin(q2)*dq1^2+(2*m3*r-
m3*L3)*dr*dq2;
%cor3=-1/2*m3*(2*r-L3)*(cos(q2)^2*dq1^2+dq2^2);
```

```
cor3=0;

% efecto de la gravedad
h1=0;
h2=1/2*g*cos(q2)*(m2*L2+2*m3*r-m3*L3);
%h3=g*m3*sin(q2);
h3=0;

% rozamiento viscoso
Fv1=0;
Fv2=0;
Fv3=B3*dq3;

% torques perturbadores que entran a los motores
Tp1=1/N1*(H11*ddq1+cor1+h1+Fv1);
Tp2=1/N2*(H22*ddq2+cor2+h2+Fv2);
% sist. con tornillo, no coriolis, no gravedad, solo I3 y B3
Tp3=1/N3*(H33*ddq3+cor3+h3+Fv3);
```

Vemos en el código de arriba dos detalles muy importantes.

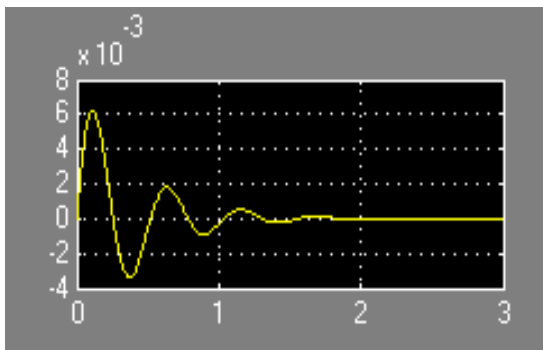
Primero, los torques perturbadores están divididos por el factor de la caja reductora de cada motor, es decir el motor recibe una fracción del torque perturbador que hay en cada eje del articulación.

Segundo, el motor del eslabón 3 recibe torque y no fuerza, esto es debido a que es eslabón 3 está formado por un tornillo sin fin y una nuez que se desplaza, por tanto los efectos de Coriolis y gravedad serán ceros, y el tensor inercial será constante, además hay que agregar el rozamiento viscoso que todo tornillo produce.

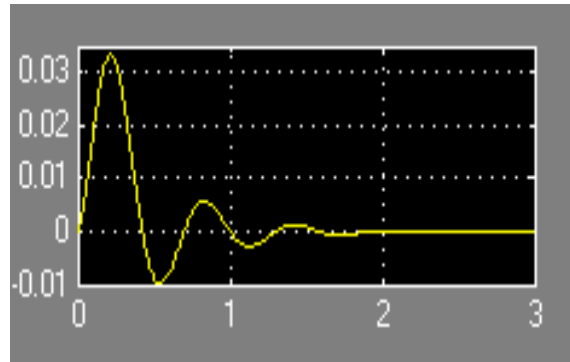
3) Resultados de la Simulación

El tiempo de simulación a usar será de tres segundos, que nos permitirá ver la evolución del error en el tiempo para los tres motores. Recordar que el error es medido y controlado respecto al eje de articulación de cada eslabón del robot y no respecto al eje del motor.

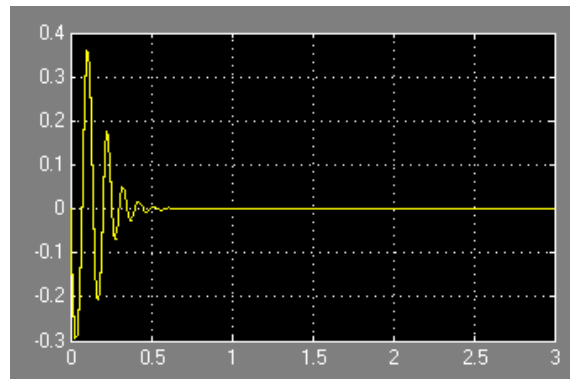
La gráfica del error cometido por el motor en la articulación 1 es como máximo 0.006 rad bastante pequeño y el error en régimen permanente es cero, debido a que el sistema es de tipo II puede hacer el error cero ante entradas tipo rampas.



El error cometido por el motor en la articulación 2 en radianes es:



El error cometido por el motor en la articulación 3 en milímetros es:



4) Recomendaciones.

Se recomienda diseñar los controladores PID discretos para que respondan en forma sobreamortiguada, es decir con un factor de amortiguamiento de 1 ($\zeta=1$) o $M_p=0$ (máximo sobrepulso), con esto se reducirán las oscilaciones que se ven en el motor 3 del eslabón de traslación.

5) Conclusiones

- Aplicar un control monoarticular, sin tener presentes en el control las perturbaciones que produce el robot, es recomendable cuando se tengan motores con caja de engranajes, ya sea interna o externa a él, para que disminuyan el torque perturbador producido por el robot. También este método es aplicable cuando las masas no son considerables y las aceleraciones y velocidades no son muy altas.
- Usar la simulación para el motor DC es una herramienta importante que acerca las

respuestas de simulación a las respuestas reales, siempre y cuando el robot sea construido con una mecánica fina y el puente de H para los motores esté muy bien diseñado.

- Se analizó el control desde el punto de vista del motor, es decir se controla al motor, en el eje de salida de la articulación, y no al robot. Es mejor y más real obtener la posición desde el motor que desde el robot por dinámica inversa.
- Para obtener menos error en el estado transitorio en los motores es necesario aplicar una aceleración tipo rampa. En este caso la aceleración es infinita para alcanzar la velocidad nominal.

6) Referencias

- [1] Anibal Ollero Baturone “ROBÓTICA Manipuladores y Robots Móviles” 2001
- [2] Antonio Barrientos “Fundamentos de Robótica” Segunda edición.