Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Electrónica
Avance II de proyecto
Laboratorio Experimental de Sistemas Mecatrónicos

Nombre del proyecto:

Desliza-inador

Fecha de entrega: 10 de noviembre de 2020
Fecha de envío: 10 de noviembre de 2020

Alumnos que la presentan:

Díaz Cisneros Sergio Fabrizzio	Jiménez Peña José Carlos
Flores Tecpoyotl Raymundo	Rojas López Daniel
González Montalvo Alejandro	Sánchez Saldaña Carlos Daniel

I. Objetivo

Nuestro proyecto consiste en un dispositivo el cual es capaz de manipular los movimientos de una cámara, la cual estará sujetada al dispositivo a través de un sistema de sujeción, los movimientos se realizarán por medio de motores DC controlados por un potenciómetro. Nuestro dispositivo constará de dos movimientos principales, el primero será el desplazamiento de la cámara a través de una cuerda, el segundo consta de la rotación de la cámara para tener una vista de 360°. Como un adicional, se buscará lograr hacer una estabilización de la toma, aunque este punto no será tomado en cuenta para el mecanismo principal.

II. Planos

Para empezar a plantear nuestro Desliza-inador, primero realizamos un prototipo en SolidWorks, mismo que será necesario en el análisis de este y la manipulación con Matlab y LabView.

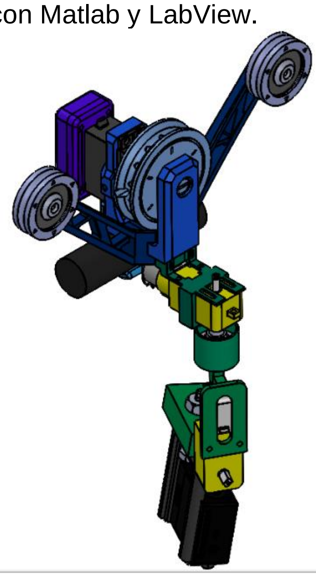

Figura 1. Desliza-inador en vista isométrica SolidWorks.

Planteamos 3 movimientos en nuestro sistema. El primero se trata de una traslación lineal sobre un eje, representado por ΔX en las figuras 2 y 3.
El segundo es un movimiento de giro ¨360¨ sobre un eje, el cuál resulta en el movimiento de giro de rotación sobre la cámara, siendo α en las figuras 2 y 3.
El tercero siendo un movimiento de giro acomodado en cadena bajo el movimiento 2, actuando como inclinación en la cámara, mostrado como γ en las figuras 2 y 3.

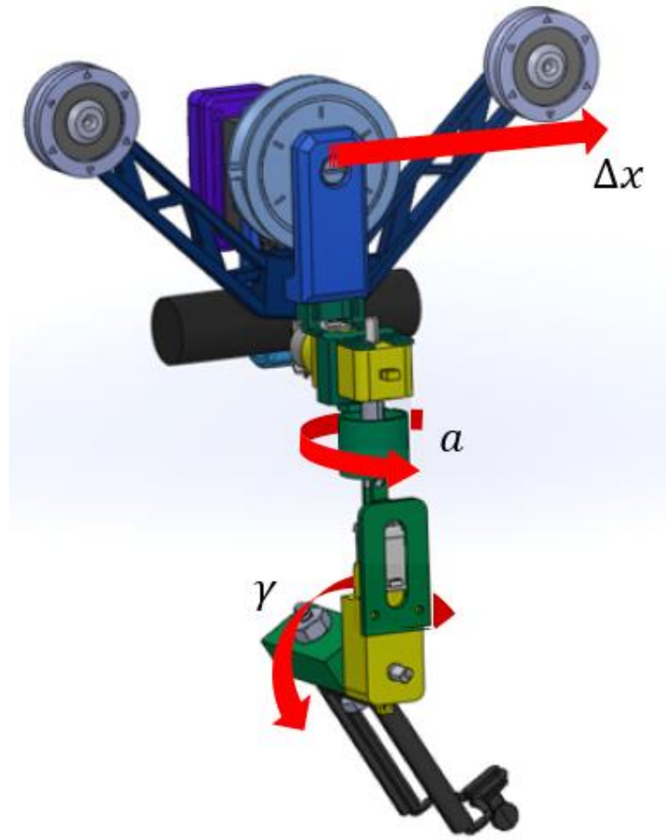

Figura 2. Desliza-inador en vista isométrica con ángulos de movimiento.

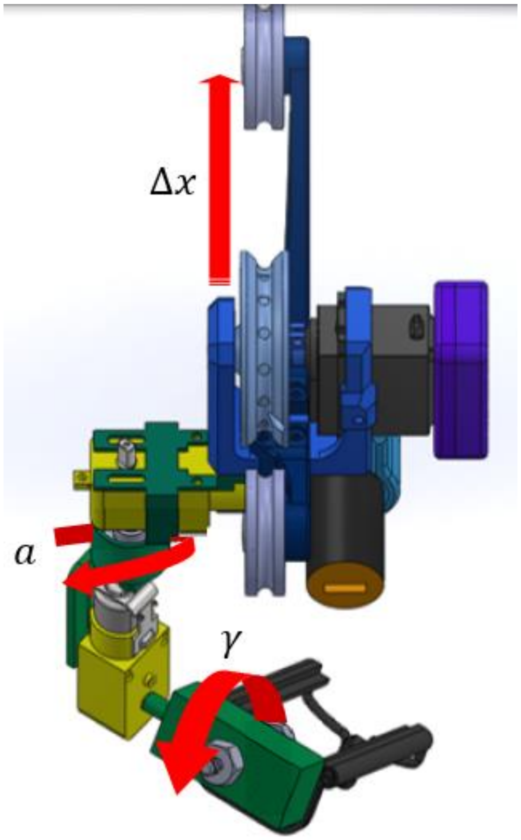

Figura 3. Desliza-inador con ángulos de giro en vista lateral.

Podemos sintetizar los movimientos de nuestro sistema en el siguiente plano

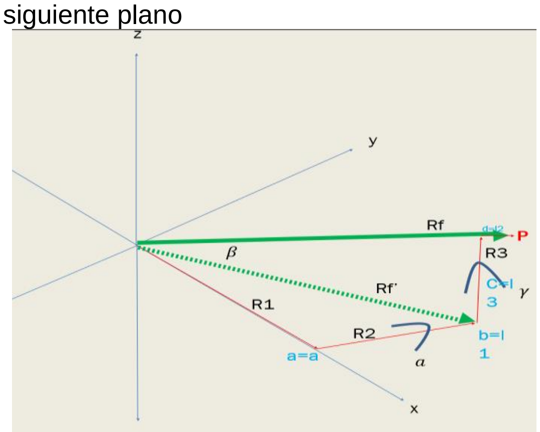

Figura 4. Plano 1

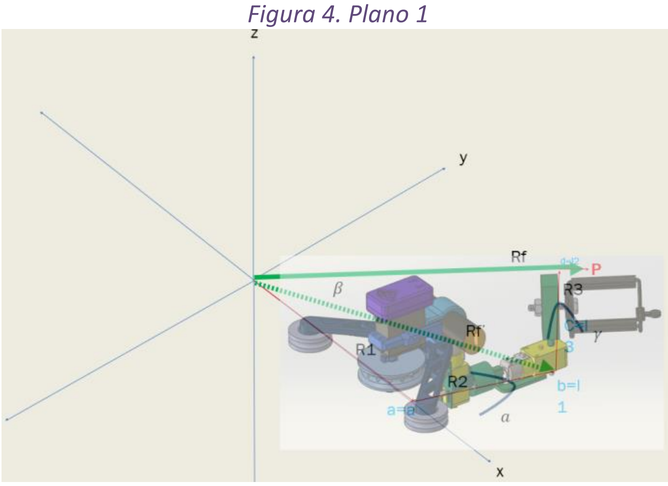

Figura 5. Plano 1 con representación directa.

Decidimos desplazar el Desliza-inador en el plano anterior reacomodando el sistema de referencia global para simplificar las ecuaciones y su adaptación a Matlab. El espacio (L1) que existe entre ΔX y α es una longitud invariable y resulta más fácil si ubicamos α en el eje X.

El deslizamiento principal ΔX sigue siendo sobre el eje X.

α ahora gira en paralelo al eje Y. L3 se puede mover en el plano XZ. Si el sistema está detenido como se muestra en la figura 6, L2 se puede mover en el plano XY, sin embargo, esto puede cambiar. Estos dos movimientos están relacionados estrechamente, el giro de α afecta directamente al punto P, el cual podría lograr movimientos circulares sobre el plano XY, esto cambiaria con el movimiento de γ rotando en el plano ZX.

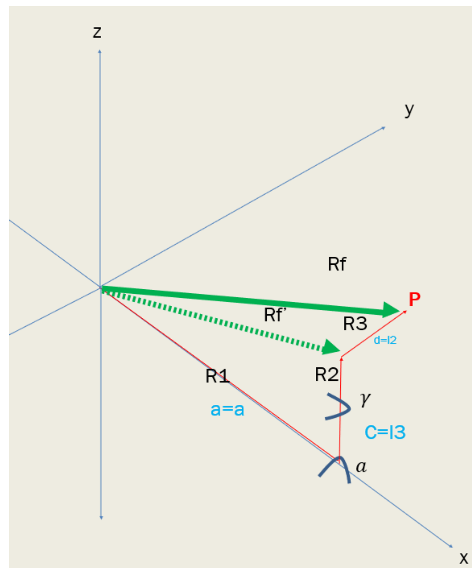

Figura 6. Plano 2

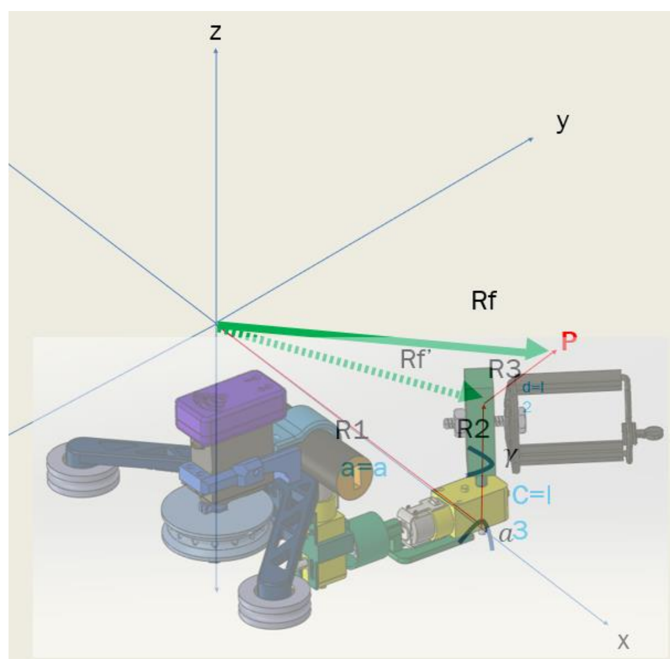

Figura 7. Plano 2 con representación directa.

Entonces para L2, al ser el segundo eslabón, no nos permite relacionar permanentemente solamente con un plano.
El movimiento que podemos imaginar se puede ver representado como la generación de anillos posibles dados por el giro de α. Formando una especie de dona que se puede deslizar en el eje X siendo este nuestro espacio muestral.

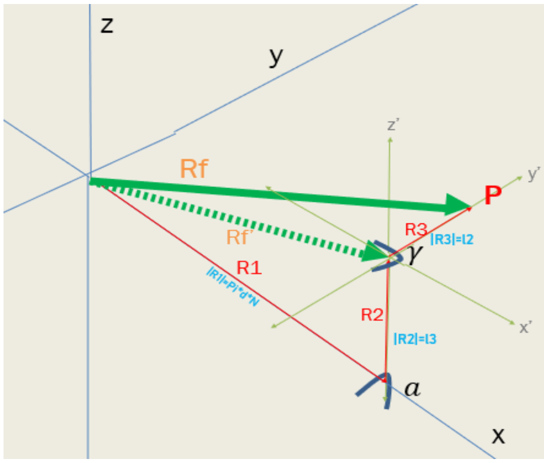

Figura 8. Plano 3

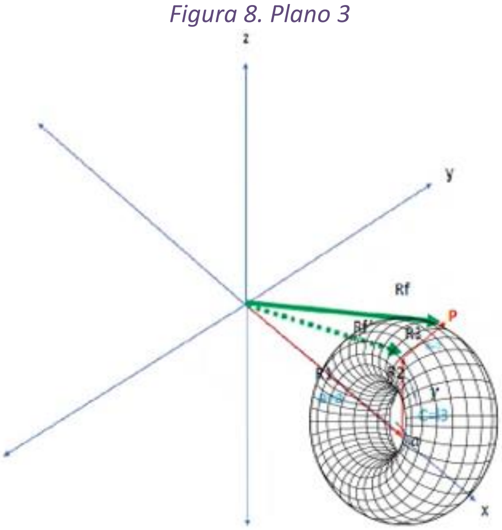

Figura 9. Plano 3 con representación de ¨dona¨

Si el movimiento de la dona se desliza en eje X, se generaría un prisma rectangular de posiciones posibles del punto P

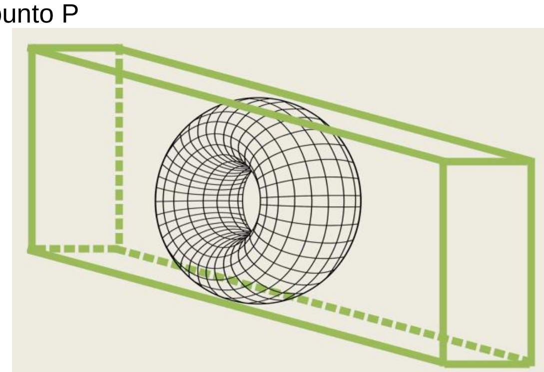

Figura 10. Espacio creado por el deslizamiento de la ¨dona¨

El deslizamiento podría ser considerado como infinito, generaría un rectángulo que se podría mover sobre x, dependiendo entonces de la cuerda sobre la cual este colocado el deslizador. Dominio de a: (-∞, ∞)$\hat{\imath}$
El ancho de la caja seria 2 L2 comprendida de L2: (-L2, L2)$\hat{\jmath}$
La altura L3 tendría dominio L3: (-L3, L3)$\hat{k}$

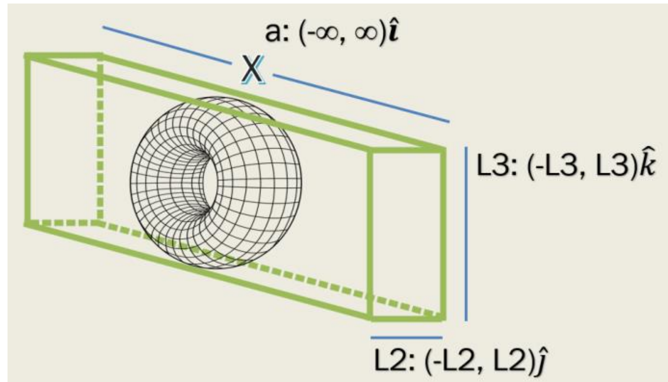

Figura 11. Parámetros del rectángulo generado.

III. Matlab

Ya conociendo el sistema y sus movimientos, implementamos en Matlab para controlar el sistema. El código es el siguiente:

Modelo de sistema de tres grados de libertad

x y z

$$a = [-\infty, \infty]$$
$$L_2 = [-l_2, l_2]$$
$$L_3 = [-l_3, l_3]$$
$$D_r = Cte$$

Deslizador

$$x_1 = d_r * rad * \pi$$
$$y_1 = 0$$
$$z_1 = 0$$

Rotación

$$x_2 = 0$$
$$y_2 = 0$$
$$z_2 = l_3 \sin(\alpha)$$

Inclinación

$$x_3 = l_3 \cos(\alpha) + l_2 \cos(\gamma)$$
$$y_3 = l_2 \sin(\gamma)$$
$$z_3 = 0$$

$$\begin{array}{ccc} 0 & y & 0 \\ 0 & y & -l \\ l\cos\theta\cos\phi & y + l\cos\phi\sin\theta & -l - l\sin\phi \end{array}$$

$$x = (d_r * N * \pi) + l_3 \cos(\alpha) + l_2 \cos(\gamma)$$
$$y = l_2 \sin(\gamma)$$
$$z = l_3 \sin(\alpha)$$
$$\dot{x} = (d_r * \pi)\dot{N} - l_2 \sin(\gamma)\dot{\gamma} - l_3 \sin(\alpha)\dot{\alpha}$$
$$\dot{y} = l_2 \cos(\gamma)\dot{\gamma}$$
$$z = l_3 \cos(\alpha)\dot{\alpha}$$

$$\begin{matrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{matrix} = \begin{pmatrix} (d_r * \pi) & -l_2\sin(\gamma) & -l_3\sin(\alpha) \\ 0 & l_2\cos(\gamma) & 0 \\ 0 & 0 & l_3\cos(\alpha) \end{pmatrix} \begin{pmatrix} \dot{N} \\ \dot{\gamma} \\ \dot{\alpha} \end{pmatrix}$$

```
clc
clear all
close all
ts=0.1;
t=0:ts:10;
a=0.1;
% 1 Condiciones iniciales del manipulador
l2=2;
l3=3;
d=1
q1(1)=2*(pi/180);
q2(1)=2*(pi/180); %alpha
q3(1)=2*(pi/180); %gamma
%2
xr(1)=d*pi()*q1(1)-l2*sin(q3(1))-l3*sin(q2(1));
yr(1)=l2*cos(q3(1));
zr(1)=l3*cos(q2(1));


%3 Referencias deseadas
xrd=5;
yrd=-2;
zrd=3;
for k=1:length(t)
% a) Errores de control
xre(k)=xrd-xr(k);
yre(k)=yrd-yr(k);
zre(k)=zrd-zr(k);
e=[xre(k);yre(k);zre(k)];
%b) Matriz Jacobiana
J=[(d*pi()) -l2*sin(q3(1)) -l3*sin(q3(1));
 0 l2*cos(q3(1)) 0;
0 0 l3*cos(q2(1)) ];
%c) matriz de ganacia
K=[0.3 0 0;
   0 0.3 0;
   0 0 0.3];
%d) ley de control
v=inv(J)*K*e;
q1p_ref(k)=v(1);
q2p_ref(k)=v(2);
q3p_ref(k)=v(3);
q1(k+1)=q1(k)+q1p_ref(k)*ts;
q2(k+1)=q1(k)+q2p_ref(k)*ts;
q3(k+1)=q1(k)+q3p_ref(k)*ts;
xr(k+1)=d*pi()*q1(k+1)-l2*sin(q3(k+1))-l3*sin(q2(k+1));
yr(k+1)=l2*cos(q3(k+1));
zr(k+1)=l3*cos(q2(k+1));
end
%Graficas
figure('Name', 'Errores')
subplot(211)
plot(t,xre,'linewidth',2), grid on
legend('Error en x')
xlabel('tiempo'), ylabel('Error [m]')
subplot(212)
plot(t,yre,'linewidth',2), grid on
legend('Error en y')
xlabel('tiempo'), ylabel('Error [m]')
subplot(213)
plot(t,zre,'linewidth',2), grid on
legend('Error en z')
xlabel('tiempo'), ylabel('Error [m]')
figure('Name', 'Acciones de control')
subplot(211)
plot(t,q1p_ref,'linewidth',2), grid on
legend('Velocidad en q1')
xlabel('tiempo'), ylabel('Velocidad [rad/s]')
subplot(212)
plot(t,q2p_ref,'linewidth',2), grid on
legend('Velocidad en q2')
xlabel('tiempo'), ylabel('Velocidad [rad/s]')
subplot(213)
plot(t,q1p_ref,'g','linewidth',2), grid on
legend('Velocidad en q1')
xlabel('tiempo'), ylabel('Velocidad [rad/s]')
```