Modelado y Simulación de Brazo Robótico de 2 Grados de Libertad

Proyecto de mecatrónica de un brazo robótico plano de 2 enlaces o “links”. Primero se modela el robot usando Solidworks y luego se exporta el modelo a Matlab usando la herramienta *Simscape multibody link* para que podamos construir el algoritmo de control y simular el comportamiento real del robot tanto gráficamente como usando el explorador de mecánica de Matlab.

Modelado del robot con SolidWorks

A continuación se mostrarán las piezas que componen al robot y el ensamblaje para construir el brazo robótico final.

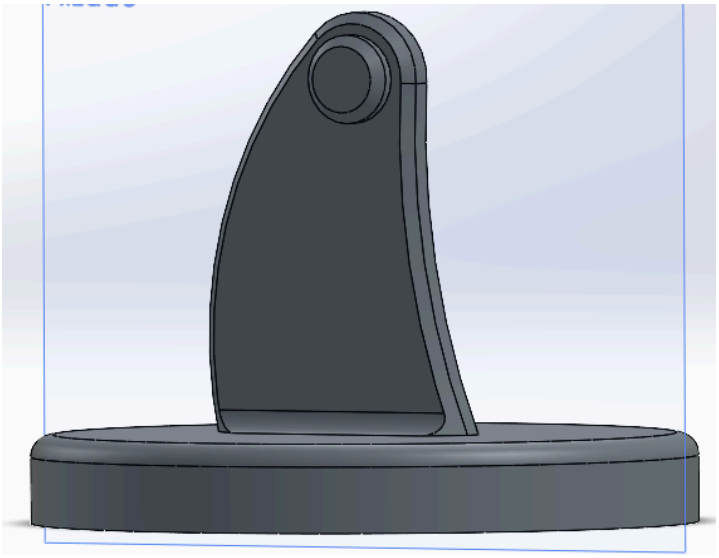

Figura 1. Pieza 1 del brazo robótico

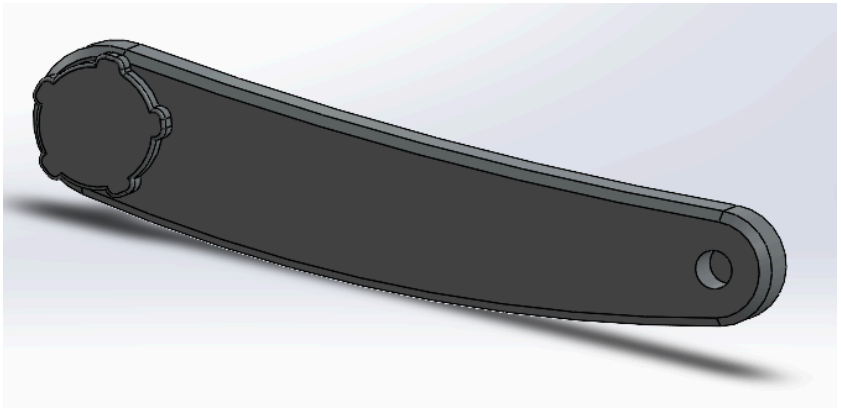

Figura 2. Pieza 2 del brazo robótico

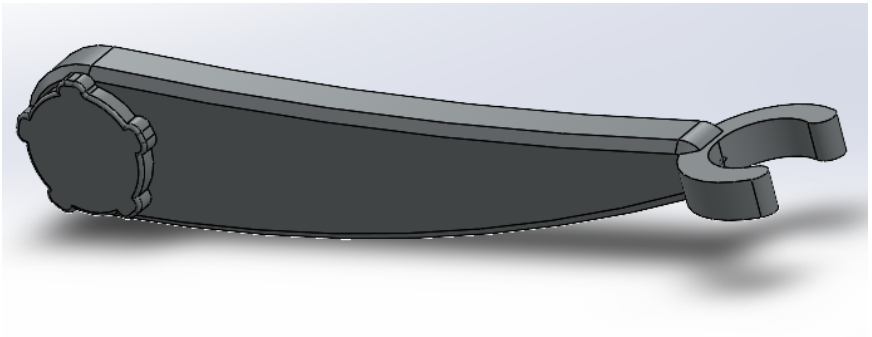

Figura 3. Pieza 3 del brazo robótico

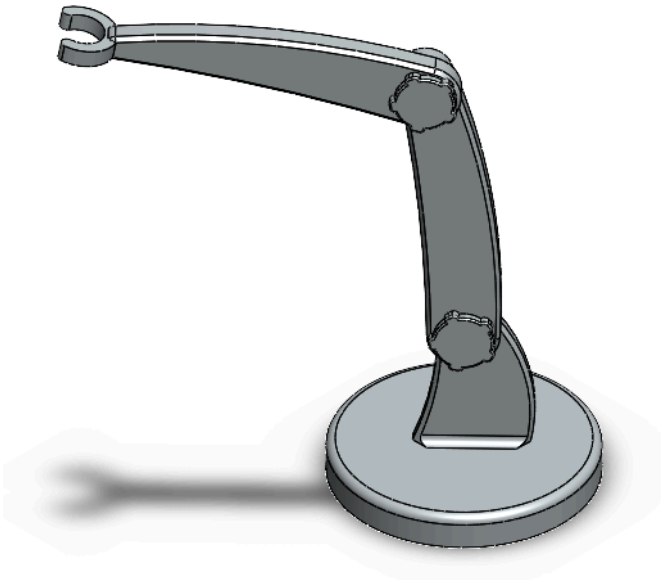

Figura 4. Modelo final del Brazo robótico con SolidWorks

Cinemática Directa y Cinemática Inversa del brazo robótico

Cinemática Directa

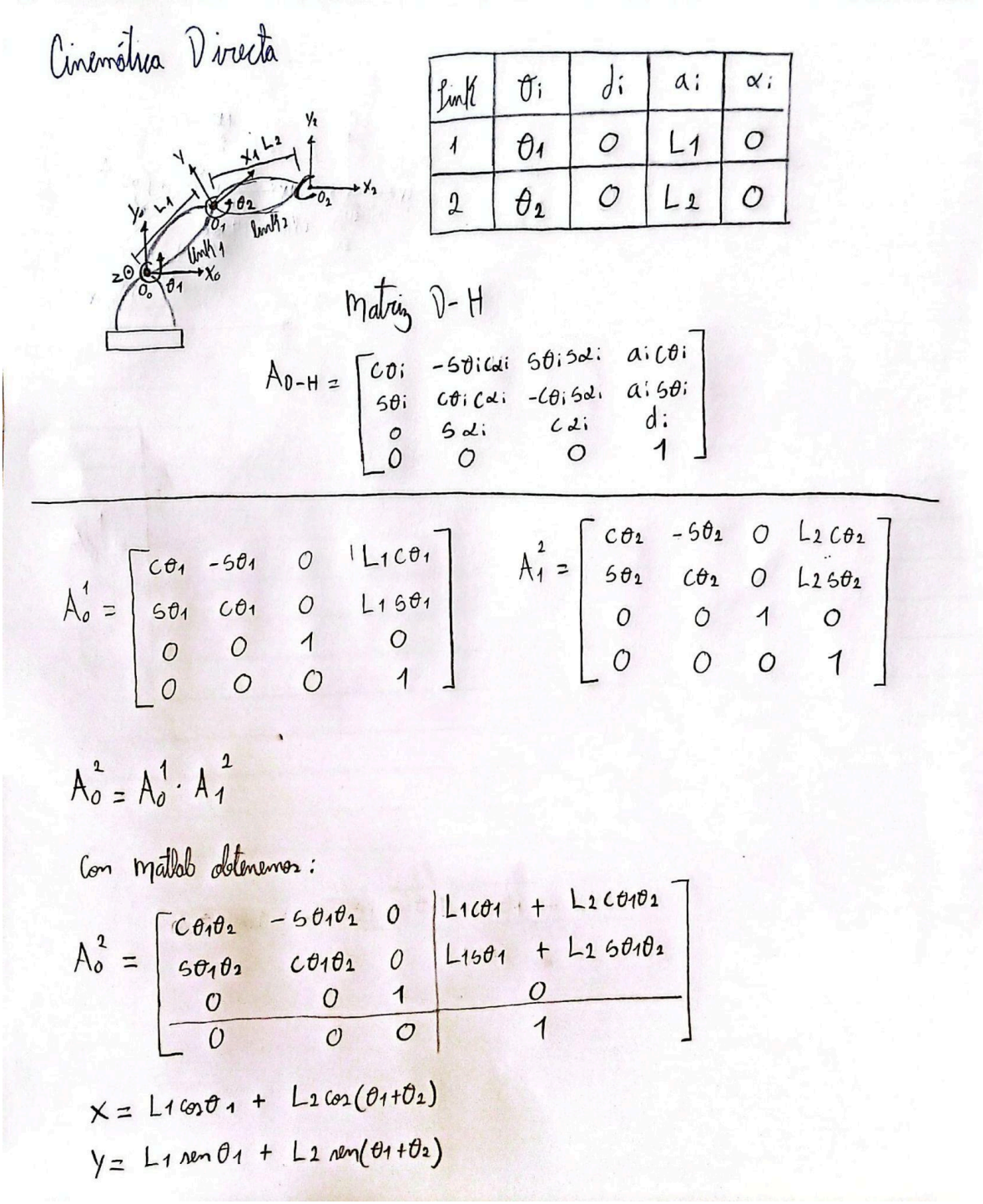

Figura 5. Cálculo de la cinemática directa

Cinemática Inversa

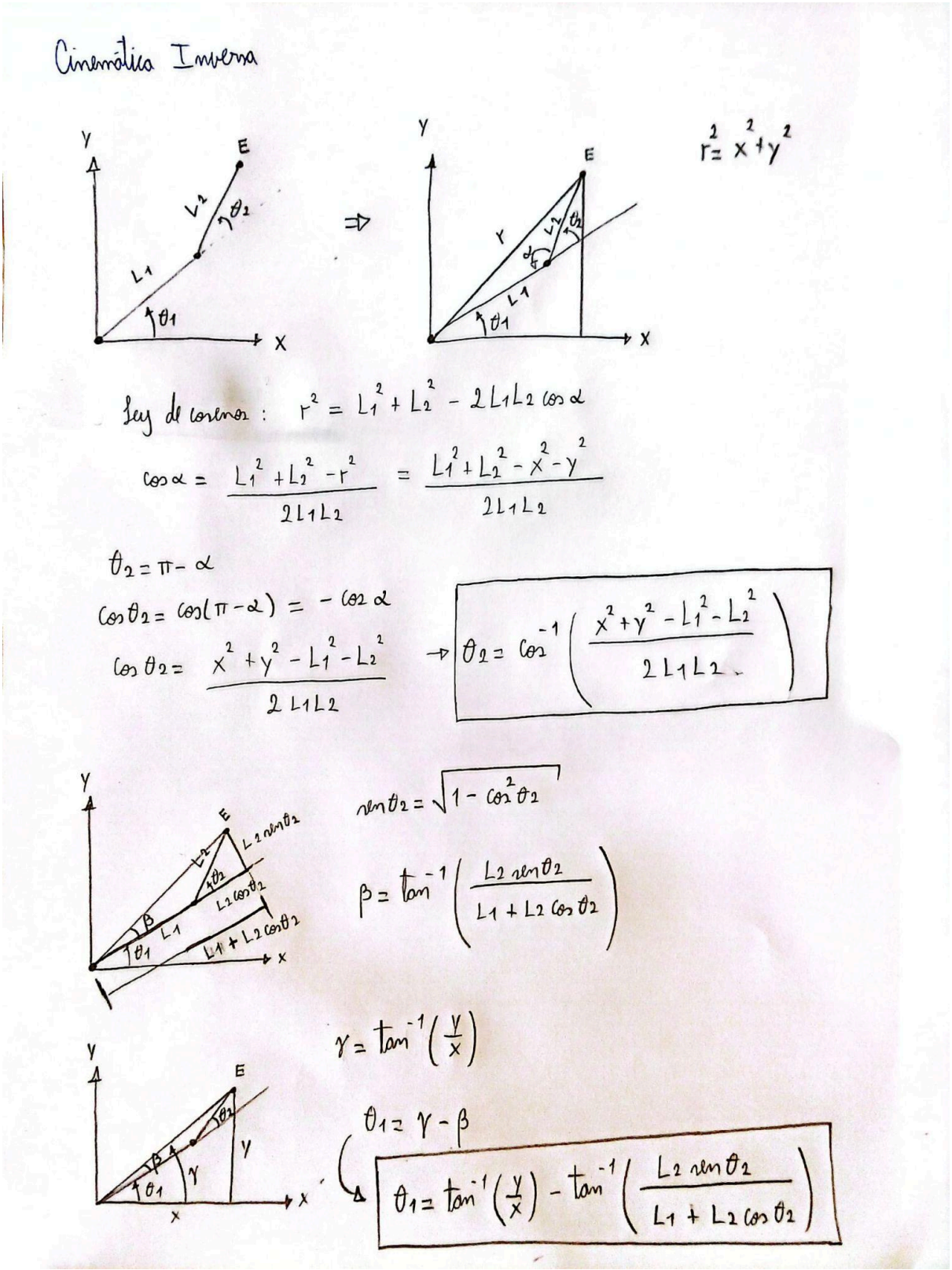

Figura 6. Cálculo de la cinemática inversa

Modelado del robot con Matlab y Simulink

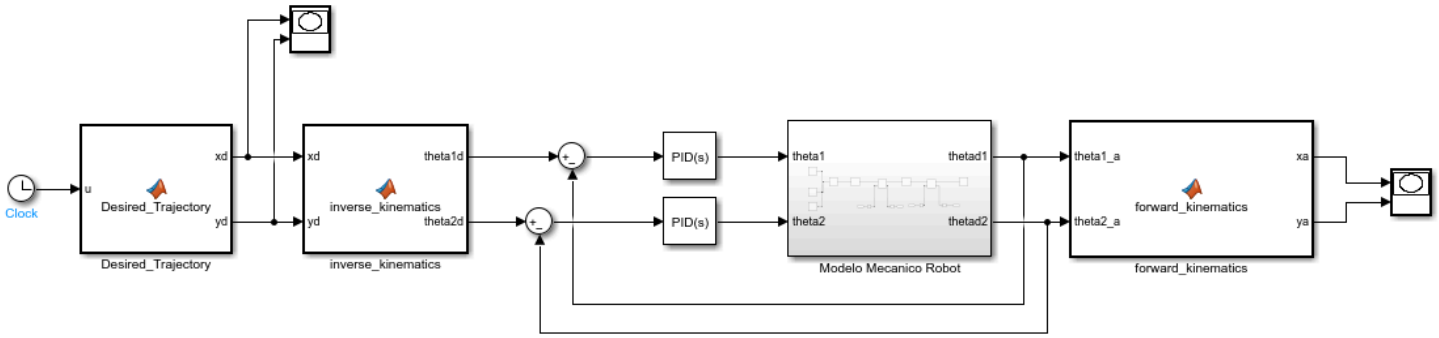

Figura 7. Diagrama de bloques del algoritmo de control del robot

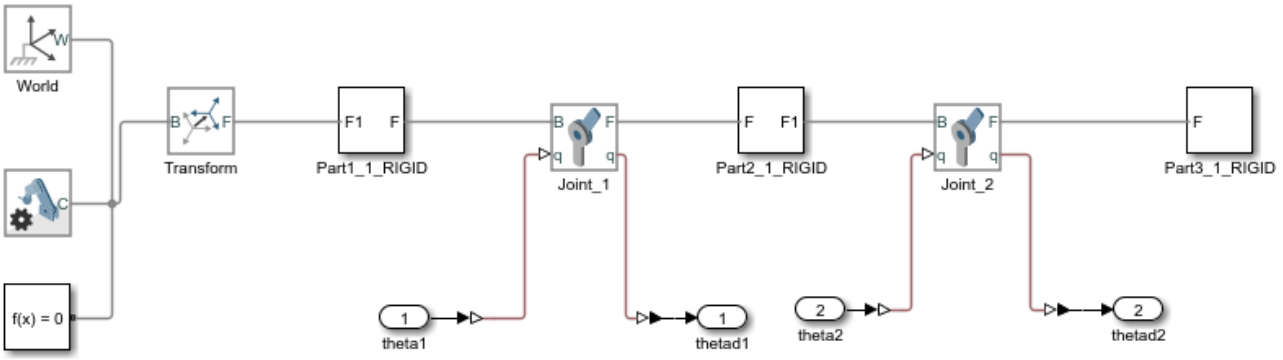

Figura 8. Modelo mecánico del robot en Simulink

Como se puede observar en el diagrama de la **Figura 7**, hay una señal de Reloj que nos ayuda a dar una variable de tiempo para determinar la trayectoria deseada de nuestro robot respecto al tiempo.

- **Bloque Desired_Trajectory**

  Este primer bloque corresponde a inicializar la trayectoria deseada.

  La trayectoria deseada elegida tiene una forma circular y está dada por la siguiente expresión paramétrica:

  $x = 1 + 0.5 * \sin((2 * pi / 5) * t + pi / 2)$

  $y = 1 + 0.5 * \cos((2 * pi / 5) * t + pi / 2)$

  Si usamos una calculadora gráfica como Desmos, la gráfica de la trayectoria se verá de la siguiente forma:

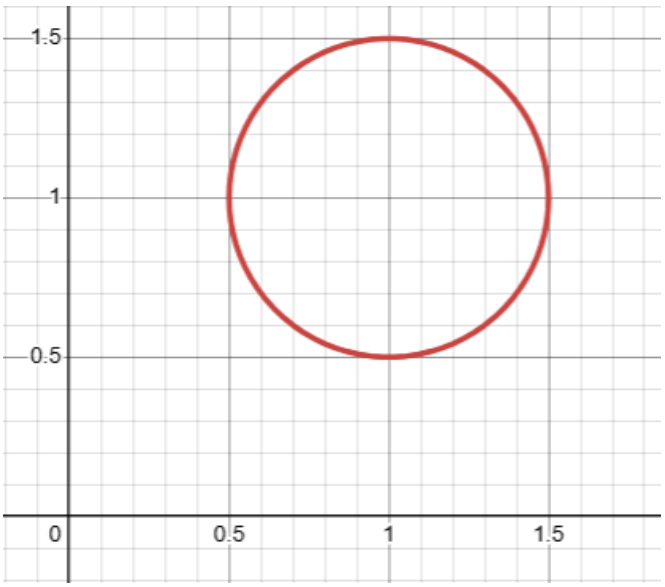

Figura 9. Forma de la trayectoria deseada para t = [0, 10]

La trayectoria deseada es en un plano, ya que se trata de un robot planar, por lo que tenemos dos variables como salida: ***xd*** y ***yd***

```
function [xd,yd] = Desired_Trajectory(u)

xd=1+0.5*sin((2*pi/5)*u+pi/2);
yd=1+0.5*cos((2*pi/5)*u+pi/2);
```

Como se puede observar en el código, ambas variables tienen la misma entrada ***u*** que es el tiempo reloj antes mencionado.

- **Bloque inverse_kinematics**

  El siguiente bloque corresponde a la Cinemática Inversa del robot. Dicho bloque obtiene la trayectoria deseada ***xd*** y ***yd*** y nos da como resultado una posición angular del robot para seguir su trayectoria en un tiempo determinado.

  Las posiciones angulares son ***theta1d*** y ***theta2d***. Estas dos posiciones angulares son accionadas por un motor. Estos motores pueden ser de paso a paso o un servomotor. Dichos motores deben ser controlados por un controlador.

```
function [theta1d,theta2d] = inverse_kinematics(xd,yd)
l1=1;
l2=1;
theta2d=acos((xd^2+yd^2-l1^2-l2^2)/(2*l1*l2));
theta1d=atan(yd/xd)-atan((l2*sin(theta2d))/(l1 + l2*cos(theta2d)));
```

- **Control del robot**

  El control de robot, específicamente el control de los motores, se realiza a través de los controladores PIDs. Dichos controladores tomarán la señal de error y la intentarán eliminar para obtener una trayectoria lo más parecida posible a la trayectoria deseada de la **Figura 9.** La señal de error es creada sustrayendo la posición angular deseada (***theta1d*** y ***theta2d***), que es la salida generada por el motor del modelo mecánico. Las posiciones angulares provienen de la cinemática inversa. Dichos controladores fueron sintonizados usando la función de *Automated tuning (PID Tuner App)* de simulink.

- **Bloque forward_kinematics**

  El bloque final corresponde a la Cinemática Directa del robot. Dicho bloque se utiliza para la visualización. El bloque toma la posición angular del robot y nos da la posición del efector final del brazo robótico respecto al tiempo.

```
function [xa,ya] = forward_kinematics(theta1_a,theta2_a)
l1=1;
l2=1;
xa=l1*cos(theta1_a)+l2*cos(theta1_a+theta2_a);
ya=l1*sin(theta1_a)+l2*sin(theta1_a+theta2_a);
```

Resultados simulación

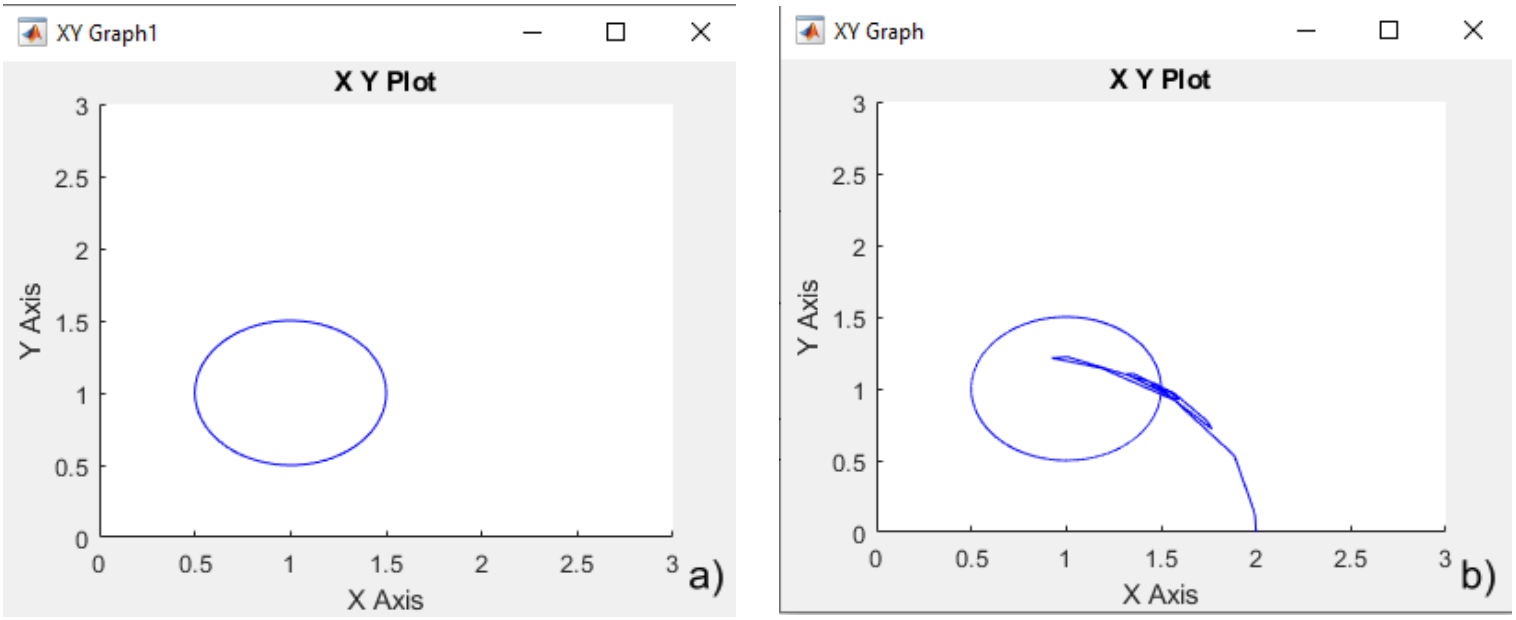

Figura 10. Resultados de la simulación. **a)** Trayectoria deseada; **b)** Trayectoria final del robot (salida del sistema)

Como observamos en la **Figura 10b**, los controladores PID permiten que el robot realice una trayectoria bastante similar a la trayectoria deseada de la **Figura 10a**.

En la trayectoria final del robot se observa que existe un gran *"Jerk"* (tasa de cambio de la aceleración). Sin embargo, a pesar de esto, la referencia es muy similar a la trayectoria deseada.

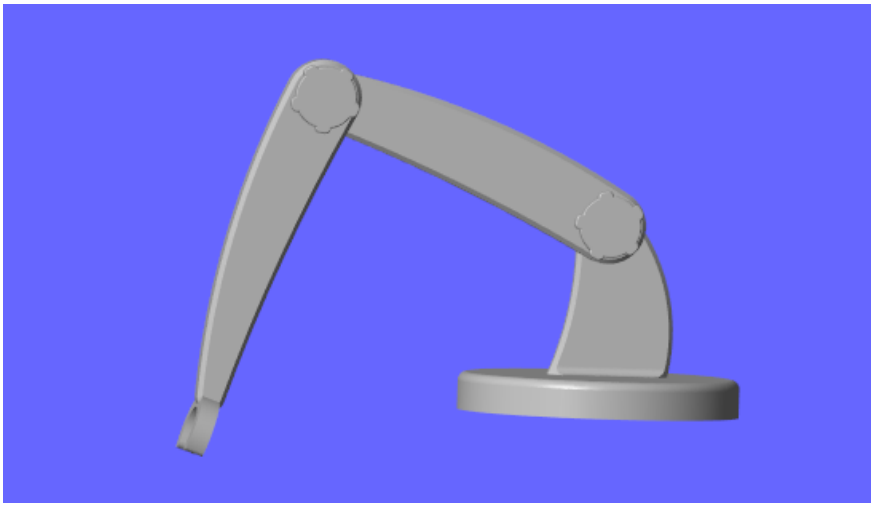

Figura 11. Modelo del brazo robótico en Matlab

Para ver el movimiento del brazo robótico, hacer click en el siguiente enlace: https://youtu.be/IvltNLjfZPE