

LaserBot-DP: Robot cortador láser

Trabajo Práctico Final - Robótica

Nicolas Nobili

Robotica

July 16, 2025

Contenido

- 1 Objetivo - Motivación
- 2 Especificaciones del Robot
- 3 Diseño
- 4 Modelo: Python
- 5 Generador de Trayectorias
- 6 Control del Robot
 - Control PD
 - Control PD + FF
- 7 Validación de Especificaciones

Objetivo - Motivación

Objetivo: Diseñar un robot de dos grados de libertad, seleccionando los componentes necesarios y verificando su correcto funcionamiento mediante simulaciones.

Motivación:



El proyecto se divide en cuatro partes principales:

- 1 Diseño mecánico del robot doble péndulo, adaptado para portar un cabezal láser.
- 2 Selección de componentes: motores, estructura y electrónica de control.
- 3 Modelado matemático y dinámico del sistema como doble péndulo invertido, incluyendo el diseño del control y el generador de trayectorias.
- 4 Simulación y validación del sistema de control en tareas simuladas de corte.

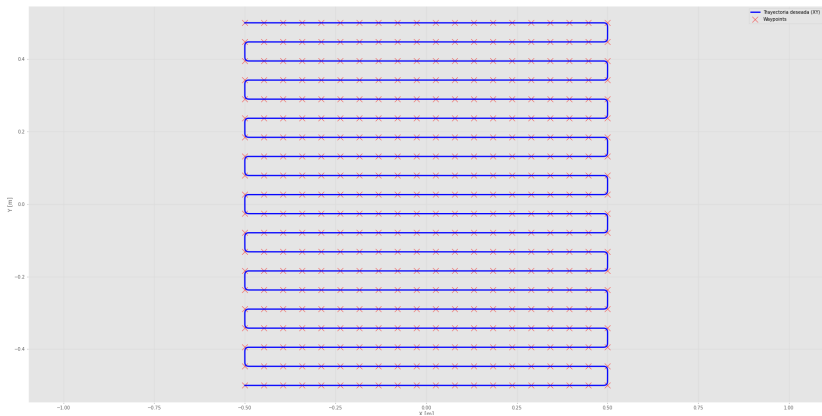
Enfoque académico con orientación al desarrollo de soluciones tecnológicas concretas.

Especificaciones del Sistema Robótico

- **Grados de libertad:** 2 GDL (configuración rotacional R-R) para moverse en el plano XY.
- **Área de trabajo:** Capacidad de movimiento sobre una superficie de $1\text{ m} \times 1\text{ m}$.
- **Carga útil:** Cabezal láser Raytools BT240 (aprox. 4 kg).
- **Precisión:** Error de seguimiento menor a **5 mm** en todo el rango de trabajo de las trayectorias planeadas.
- **Velocidad:**
 - Mínima recomendada: 10mm/s
 - Máxima recomendada: 50mm/s
- **Seguimiento de trayectorias:** Capacidad para seguir curvas y patrones predefinidos con alta precisión.
- **Rigidez estructural:** Alta rigidez para reducir vibraciones y asegurar calidad de corte.
- **Seguridad:** Freno de emergencia y protecciones por software y hardware, etc.

Ensayo de Validación de Especificaciones

Para demostrar la correcta operación del sistema robótico, se diseñó ensayo de validación que reproduce condiciones reales de trabajo. Este ensayo permite comparar el rendimiento obtenido con las especificaciones teóricas propuestas, evaluando aspectos como precisión, velocidad, estabilidad y seguimiento de trayectoria.



- **Material y geometría:** Los eslabones se fabricarán en aluminio para lograr una estructura resistente y ligera. Ambos tendrán la misma longitud, lo cual permite al robot alcanzar cualquier punto dentro de un área de trabajo de $1\text{ m} \times 1\text{ m}$, exceptuando las posiciones de singularidad. Además, el primer eslabón será más ancho que el segundo, con el fin de soportar mayores cargas y torques generados durante la operación.
- **Ubicación de los motores:** Con el fin de simplificar el montaje y reducir el momento de inercia del motor del primer eslabón, ambos motores se montarán sobre el eje del primer eslabón.
- **Transmisión del segundo eslabón:** El movimiento del segundo eslabón se realizará mediante una transmisión por correa, conectada al motor ubicado en el eje 1.

• Eslabones:

- ❶ Eslabón de 40 cm, Aluminio (perfil 6 cm × 4 cm, pared 5 mm)
- ❷ Eslabón de 40 cm, Aluminio (perfil 5 cm × 3 cm, pared 5 mm)

• Actuadores:

❶ Motor 1: Leadshine ELM2H-0750LA80E

- Par nominal: 2.4 Nm
- Reductor planetario 1:50
- Par de salida: 120 Nm
- Velocidad de salida: $360 \frac{^\circ}{s}$



Motor Leadshine
ELM2H-0750LA80E

❷ Motor 2: Leadshine ELM2H-0400LA60E

- Par nominal: 1.27 Nm
- Reductor planetario 1:25
- Par de salida: 31.75 Nm
- Velocidad de salida: $720 \frac{^\circ}{s}$



Motor Leadshine
ELM2H-0400LA60E

Diseño: Actuador 1

1 Torques Estaticos:

$$\tau_1 = m_1 \times g \times d_{cm1} = 1.91\text{Nm} \quad \wedge \quad \tau_2 = m_2 \times g \times d_{cm2} = 4.45\text{Nm}$$

$$\tau_c = m_c \times g \times d_c = 31.39\text{Nm}$$

2 Torque Dinamico:

$$I_{\text{total},1} = I_1 + I_{2,1} + I_{c,1} = I_1 + I_2 + I_{\text{tras},2} + I_{c,1} = 2.773\text{kg} \cdot \text{m}^2$$

Con una aceleracion angular maxima deseada $\gamma = 5 \frac{\text{rad}}{\text{s}}$:

$$\tau_{d,1} = I_{\text{total},1} \times \gamma = 13.865\text{Nm}$$

3 Torque total (con margen):

$$\tau_{1,\text{total}} = (37.75\text{Nm} + 13.865\text{Nm}) \times 1.5 = 77.33\text{Nm}$$

1 Torques Estaticos:

$$\tau_2 = m_2 \times g \times d_{cm2} = 1.48\text{Nm} \quad \wedge \quad \tau_c = m_c \times g \times d_c = 15.70\text{Nm}$$

2 Torque Dinamico:

$$I_{total,2} = I_2 + I_{c,2} = 0.6803\text{kg} \cdot \text{m}^2$$

Con una aceleracion angular maxima deseada $\gamma = 5 \frac{\text{rad}}{\text{s}^2}$:

$$\tau_{d,2} = I_{total,2} \times \gamma = 3.40\text{Nm}$$

3 Torque total (con margen):

$$\tau_{2,total} = (17.18\text{Nm} + 3.40\text{Nm}) \times 1.5 = 30.87\text{Nm}$$

Diseño: Modelado

En función del diseño mecánico y de las especificaciones estructurales previamente definidas, se elaboró el siguiente modelo 3D del robot.

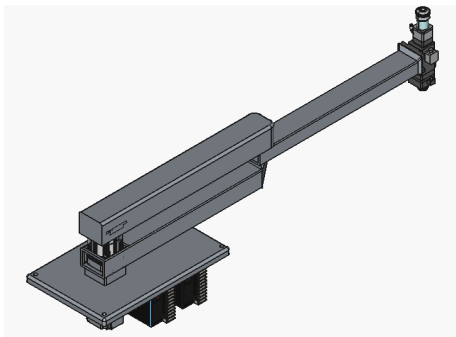


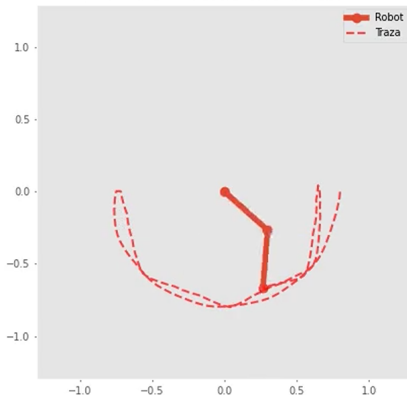
Figure: Modelo 3D del robot propuesto

Modelo Computacional

Utilizando Robotics Toolbox en Python, se definió el modelo computacional del robot. Se emplearon los parámetros Denavit-Hartenberg (DH) junto con los parámetros dinámicos estimados a partir del diseño y los motores seleccionados.

```
dp = rtb.DHRobot(  
[  
    # Eslabón 1  
    rtb.RevoluteDH(  
        a=PARAMS['A1'],      # Longitud del eslabón  
        m=PARAMS['M1'],      # Masa del eslabón  
        r=PARAMS['R1'],      # Vector COM relativo al marco del eslabón  
        I=PARAMS['I1'],      # Tensor de inercia  
        B=PARAMS['B1'],      # Fricción viscosa  
        G=PARAMS['N1'],      # Relación de engranajes  
        Jm = PARAMS['JM2']    # Inercia del motor  
    ),  
  
    # Eslabón 2  
    rtb.RevoluteDH(  
        a=PARAMS['A2'],      # Longitud del eslabón  
        m=PARAMS['M2'],      # Masa del eslabón  
        r=PARAMS['R2'],      # Vector COM relativo al marco del eslabón  
        I=PARAMS['I2'],      # Tensor de inercia  
        B=PARAMS['B2'],      # Fricción viscosa  
        G=PARAMS['N2'],      # Relación de engranajes  
        Jm = PARAMS['JM2']    # Inercia del motor  
    )  
],  
gravity=np.array([0, -9.8, 0]), # Gravedad actuando hacia -Y  
name="dp"  
)
```

Animación de trayectoria del robot



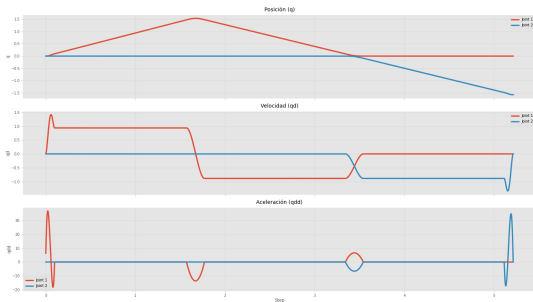
Generador de Trayectorias

Para generar las trayectorias deseadas del robot, es decir las referencias $(q^d, \dot{q}^d, \ddot{q}^d)$, se utilizó la función `mstraj` de la *Robotics Toolbox*.

Esta función permite:

- Generar trayectorias multi-eje de forma simultánea.
- Definir múltiples segmentos (N) de trayectoria.

El método implementa trayectorias con perfil polinómico de tipo **quintic** (grado 5) en las fases de aceleración y desaceleración, y un tramo de **velocidad constante** entre ellas, lo que permite una transición suave y continua en posición, velocidad y aceleración.



Control: Modelo de la Planta

Se utilizó un modelo lineal simplificado de la planta, bajo las siguientes suposiciones:

- Alta relación de reducción.
- Velocidades y aceleraciones bajas.
- Cargas balanceadas.

El modelo dinámico resultante es:

$$q(s) = \frac{1}{(J_{\text{eff}}s + B_{\text{eff}})s} (N\tau_m(s) - \tau_p(s))$$

$q(s) \rightarrow$ Posición angular del eje

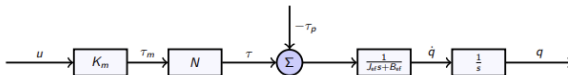
$J_{\text{eff}} \rightarrow$ Inercia efectiva

$\tau_m(s) \rightarrow$ Torque del motor

$B_{\text{eff}} \rightarrow$ Fricción viscosa efectiva

$\tau_p(s) \rightarrow$ Torque perturbador

$N \rightarrow$ Relación de reducción



Controlador PD – Diseño

Se parte de un controlador PD clásico, donde la velocidad de respuesta está determinada por la frecuencia natural ω_n . El lazo de control se diseña para obtener una respuesta críticamente amortiguada, ajustando K_p y K_d según las siguientes expresiones:

$$K_p = \frac{\omega_n^2 J_{\text{eff}}}{NK_M} \quad \wedge \quad K_d = \frac{2\sqrt{NK_M K_p J_{\text{eff}} - B_{\text{eff}}}}{NK_M}$$

El control aplicado tiene la forma:

$$u = K_p(q^d - q) - K_d\dot{q}$$

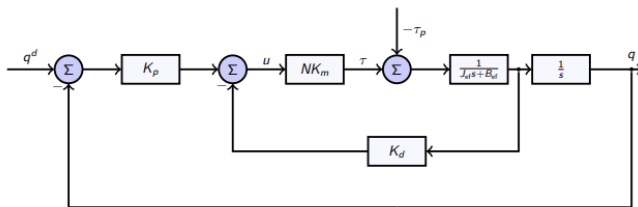
La frecuencia natural ω_n está acotada por el tiempo de muestreo del sistema:

$$\omega_n < \frac{1}{20} \cdot \frac{2\pi}{T_s}$$

Para un tiempo de muestreo $T_s = 2 \text{ ms}$, se selecciona $\omega_n = 80 \text{ s}^{-1}$.

Controlador PD – Simulación

A continuación se presenta el diagrama de simulación del controlador PD utilizado para el sistema:



Control PD

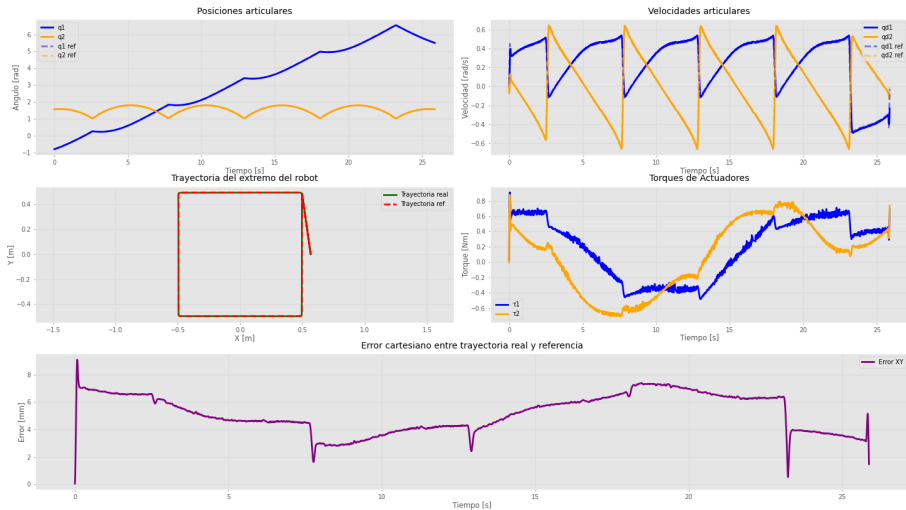


Figure: Simulación con controlador PD

Controlador PD + FF

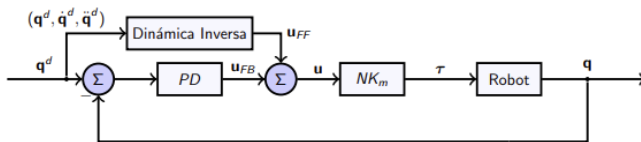
Se propone complementar el controlador PD con una acción **Feedforward** (FF), con el objetivo de:

- Mejorar la respuesta ante cambios rápidos en la referencia.
- Reducir el error en estado estacionario.

La acción de control total se expresa como:

$$u = u_{FB} + u_{FF}$$

donde u_{FF} se calcula a partir del modelo dinámico inverso del sistema, incluyendo la planta y los actuadores. Esta compensación anticipa el comportamiento deseado y contrarresta directamente las perturbaciones y errores no modelados.



Controlador PD + FF

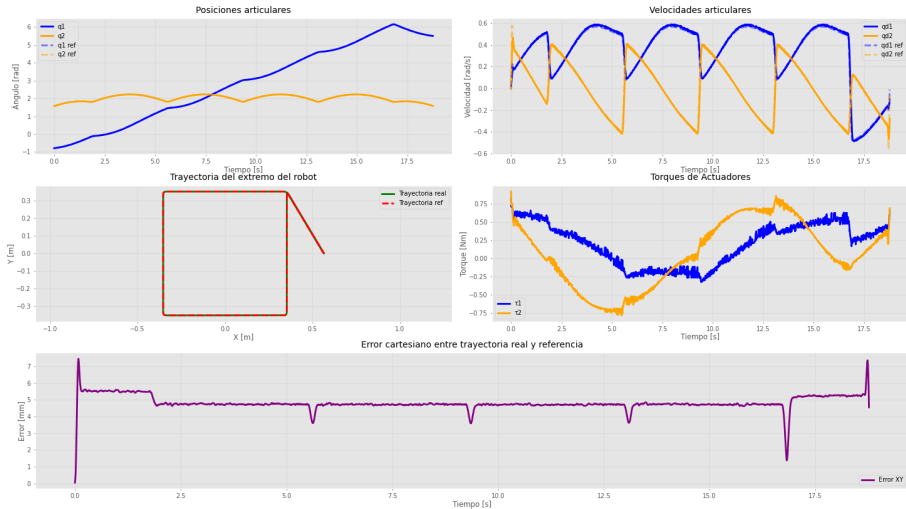


Figure: Simulacion con controlado PD + FF

Validación de Especificaciones

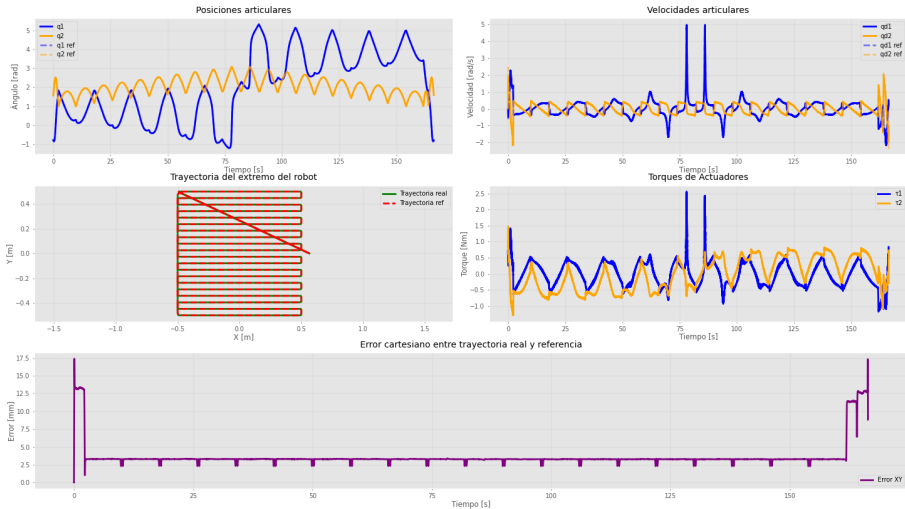


Figure: Simulación de Validación

Validación de Especificaciones + Pasos Futuros/Mejoras

Como se mencionó anteriormente, se verificó que el robot cumple con las especificaciones planteadas en el diseño.

Aspectos validados exitosamente:

- Precisión en el seguimiento de trayectorias.
- Velocidad de operación dentro de los márgenes esperados.
- Estabilidad del sistema de control.
- Correcto funcionamiento del generador de trayectorias.
- Uso adecuado de actuadores, sin superar límites recomendados por el fabricante.

Posibles mejoras y pasos futuros:

- Incorporación de cabezales intercambiables para cortar distintos materiales.
- Aumento de los grados de libertad para mayor versatilidad.
- Implementación práctica del sistema en un entorno real.
- Integración de medidas de seguridad para una operación segura en aplicaciones reales.

¡Muchas gracias!

El modelo 3D y el código están disponibles en GitHub, organizados de forma modular:

- Carpeta components: contiene trayectorias, controladores y el modelo del sistema.
- Scripts de prueba separados, fáciles de leer y modificar.

