

22.90 AUTOMACIÓN INDUSTRIAL

TRABAJO PRÁCTICO FINAL

Controlador de un  
manipulador robótico

50.615 CALVO, Clementina

51.665 VEGA, Juan Pablo

28 de noviembre de 2014

## 1. Introducción

Los parámetros del manipulador robótico utilizado se resumen a continuación.

Param/Link	1	2	EE
$d$	0	0	0
$a$	0	$L_1 = 1\text{ m}$	$L_2 = 1\text{ m}$
$\alpha$	0	0	0
$\theta$	$\theta_1$	$\theta_2$	0
$m$	$1\text{ kg}$	$1\text{ kg}$	-
$r$	$(L_i \ 0 \ 0) = (1 \ 0 \ 0)\text{ m}$		-
$I$	$\mathbb{I} \cdot m_i \cdot L_i^2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ kg} \cdot \text{m}^2$		-
$J_m$	0	0	-
$GR$	1	1	-
$B$	$1\text{ Nm/s}$	$1\text{ Nm/s}$	-

Cuadro 1: Parámetros del robot

Los mismos conforman el manipulador en el archivo `initRobot.m`.

## 2. Control de posición no lineal

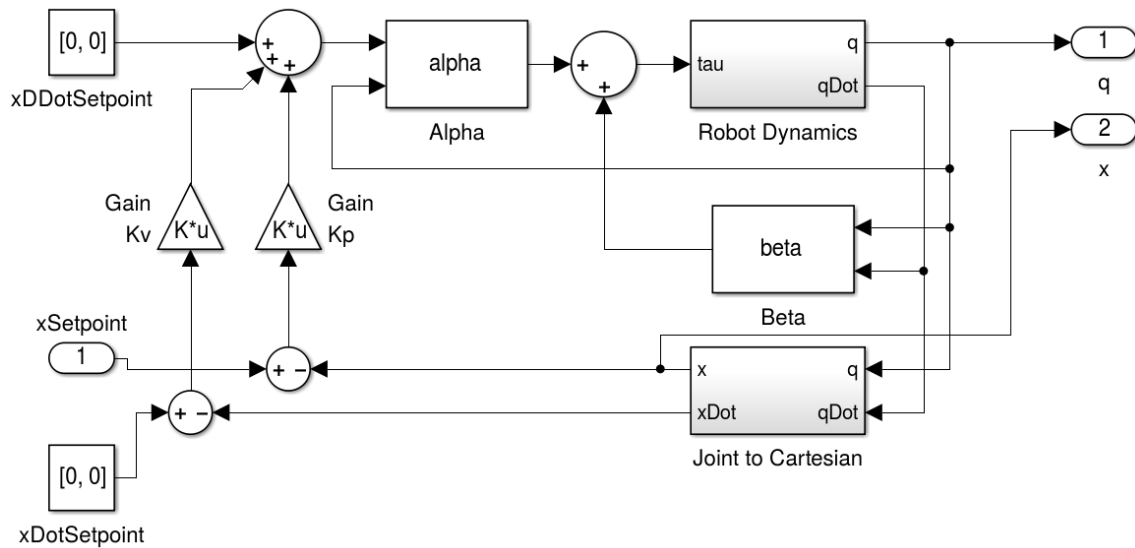


Figura 1: Diagrama en bloques para el control de posición `my_robot_1.slx`

Se utiliza el diagrama de la Figura 1 para implementar control cartesiano, donde *Joint to Cartesian* implementa la dinámica directa, y *Robot Dynamics* implementa la respuesta de la dinámica del robot ante estímulos de torque (donde se incluye o no la perturbación del modelo). También incluye la condición inicial  $\theta_i$ .

Los bloques *alpha* y *beta* están dados por las ecuaciones (2) y (3) respectivamente. Cabe mencionar que para evitar hacer cuentas de más, se premultiplicó los bloques por  $J^T$ , dado que ambos únicamente funcionan como entradas del bloque del robot, por lo cual era necesario convertirlos de fuerzas a torques.

$$\tau = J^T F \quad (1)$$

$$\alpha = M(\theta) \cdot J^{-1} \cdot \ddot{X} \quad (2)$$

$$\beta = V(\theta, \dot{\theta})\dot{\theta} - M(\theta)J^{-1}\dot{J}\dot{\theta} + G(\theta) - B(\dot{\theta}) \quad (3)$$

Para la trayectoria, se eligió como punto inicial el  $[0.5, 0]$ , y como final,  $[1, 1]$ . Se implementó una trayectoria lineal con *blends* parabólicos.

$$[0.5 \ 0] \rightarrow [1 \ 1] \quad (4)$$

Para el control, se utilizó un controlador PD críticamente amortiguado. Dado que no se especificaba ninguna condición de resonancia mecánica, se asumió un sistema ideal. Se eligió como frecuencia de muestreo  $f_s = 200 \text{ Hz}$ . Al ser una simulación discreta, existe un ruido de muestreo en  $\omega_n = 2\pi \cdot f_s$ , por lo tanto  $k_p < (\pi \cdot f_s)^2$ . Se eligió que fuera la mitad para tener un margen de funcionamiento. Finalmente,  $k_v = 2\sqrt{k_p}$  para que sea críticamente amortiguado.

Para correr esta simulación y obtener los gráficos, puede correrse el archivo *Ejercicio1.m*.

Se observa como, si bien el seguimiento es bueno, tiene un error, que se hace más notorio cuando las constantes deben ser menores por causas de resonancias mecánicas, donde se utilizó frecuencia de  $20 \text{ Hz}$  ( $k_{p2} = k_{p1}/100$ ).

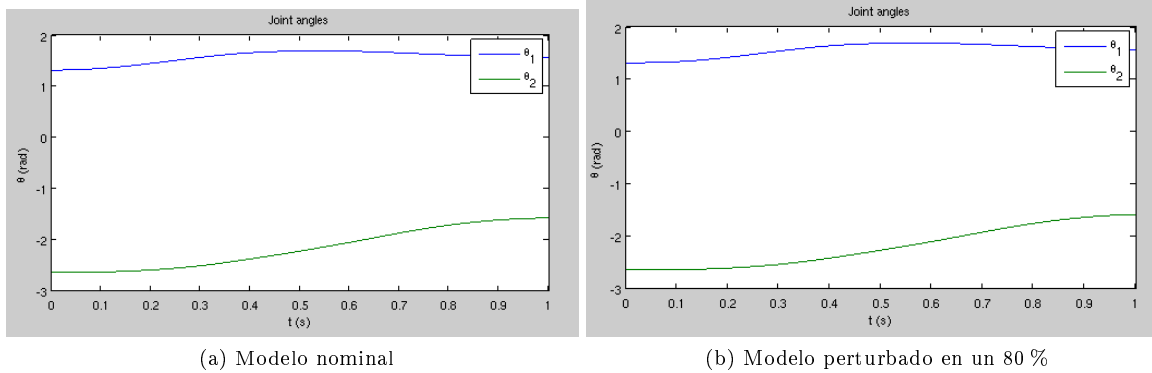


Figura 2: Ángulos de *joint* en función del tiempo

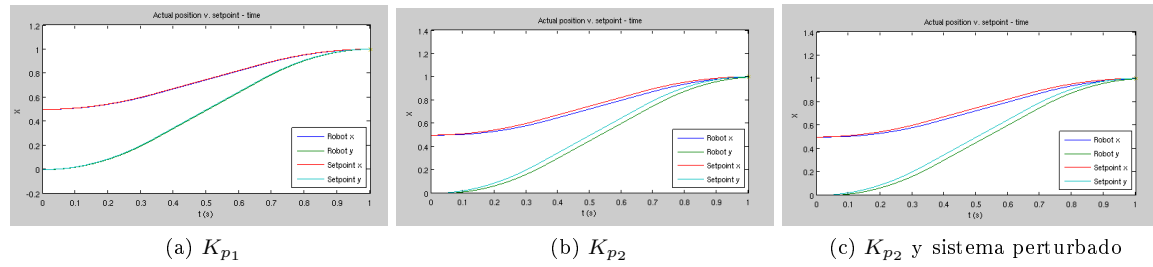
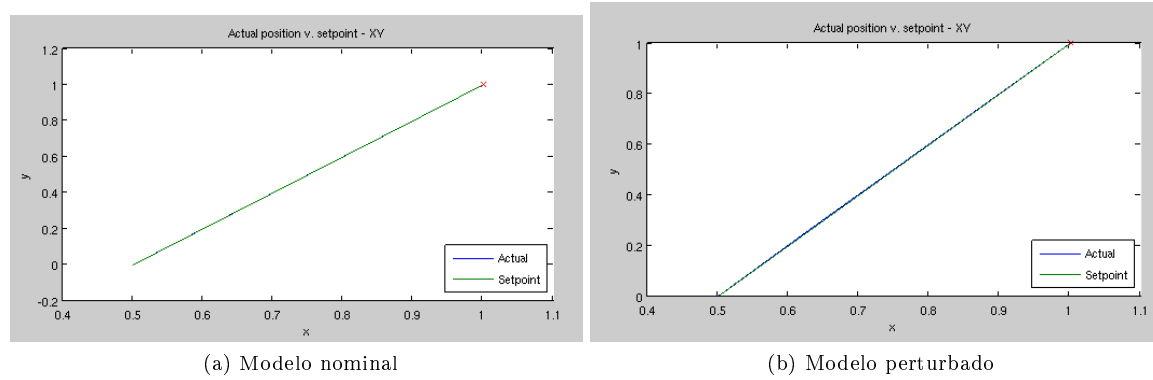


Figura 3: Posición v. setpoint en función del tiempo

Figura 4: Posición v. *setpoint*, vista *overhead*

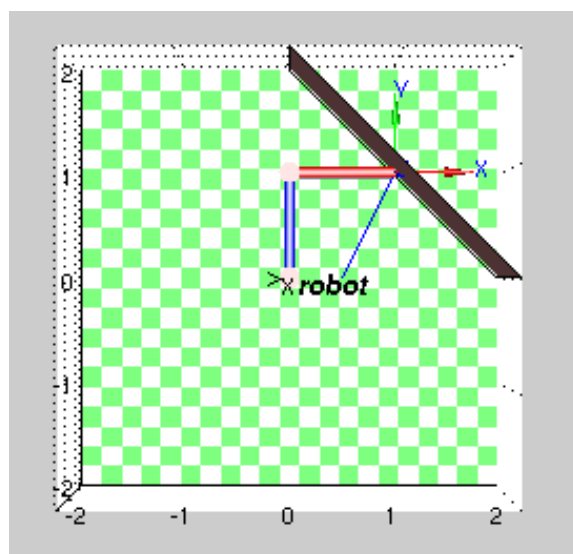


Figura 5: Movimiento

### 3. Control de fuerza no lineal

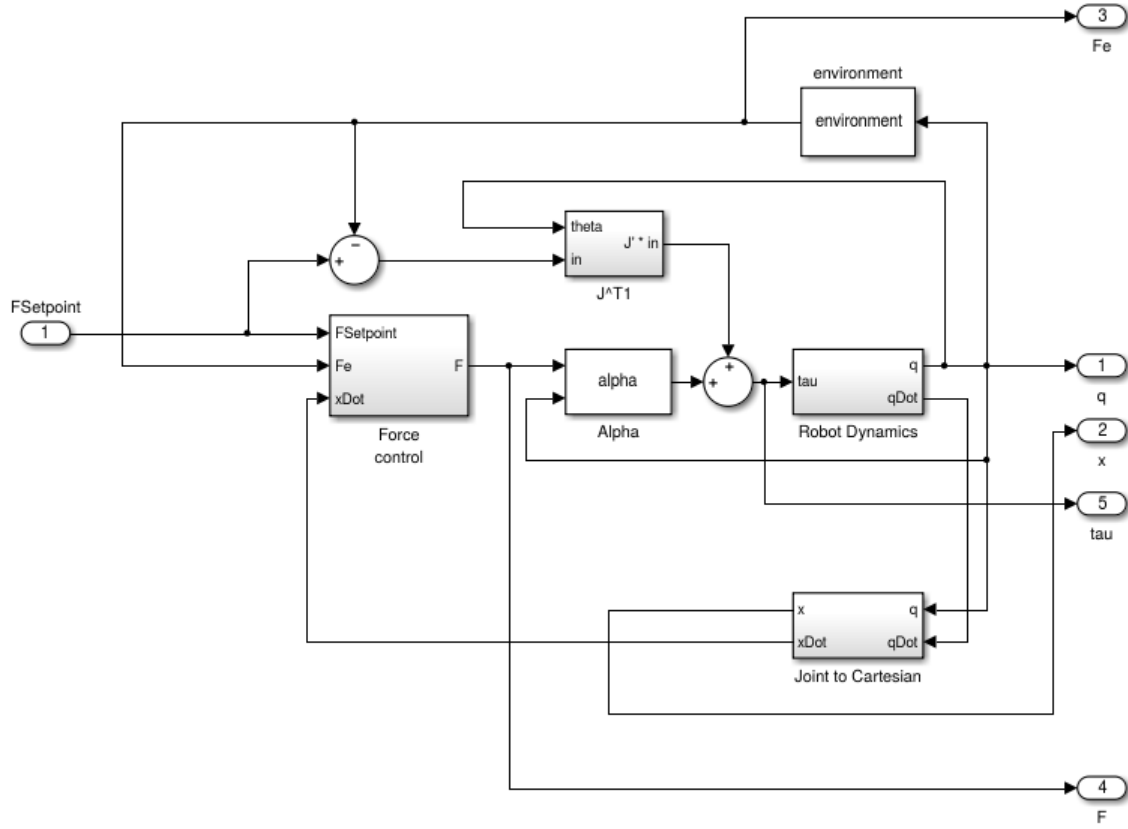


Figura 6: Diagrama en bloques para el control de fuerza my\_robot\_2.slx

Se utiliza el diagrama de la Figura (6) para implementar control cartesiano de fuerza, donde *Joint to Cartesian*, *Robot Dynamics*, y *alpha*, son los mismos que en la simulación anterior.

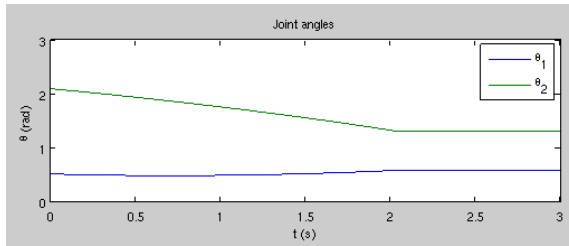
Se incluyen el bloque del Jacobiano traspuesto para realimentar el *setpoint* de fuerza, el control de fuerza que tiene la misma forma de controlador PD que el de posición de la simulación anterior, con la diferencia de que en este caso el *setpoint* es constante. Se tomaron las mismas consideraciones para las constantes.

También se introduce el bloque *environment*. El mismo representa la fuerza que el robot sentiría en sus sensores, dada por el entorno. La misma se calcula como la distancia por la constante de elasticidad de la pared, proyectado en su dirección normal  $\hat{n}$ .

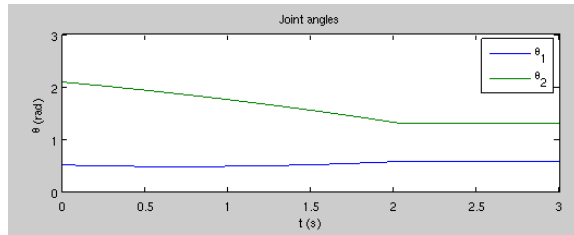
$$d = \hat{n} \cdot (x_{ee} - x_{wall}) \quad (5)$$

$$F_e = (K_e \cdot d) \cdot \hat{n} \quad (6)$$

Para correr esta simulación y obtener los gráficos, puede correrse el archivo `Ejercicio2.m`.



(a) Modelo nominal



(b) Modelo perturbado en un 80 %

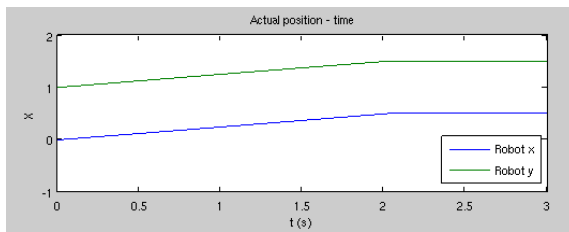
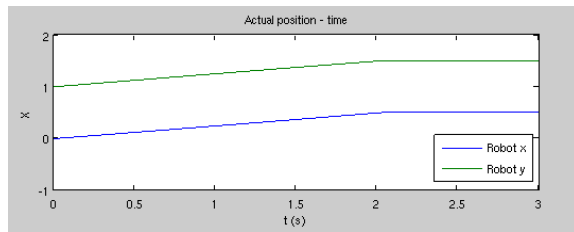
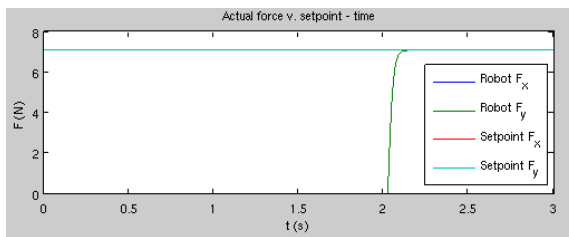
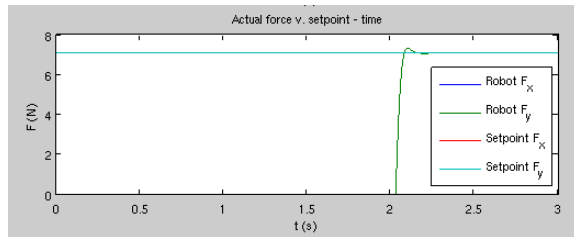
Figura 7: Ángulos de *joint* en función del tiempo(a)  $K_{p2}$ (b)  $K_{p2}$  y sistema perturbado

Figura 8: Posición en función del tiempo



(a) Modelo nominal



(b) Modelo perturbado

Figura 9: Fuerza v. *setpoint* en función del tiempo

#### 4. Control híbrido no lineal

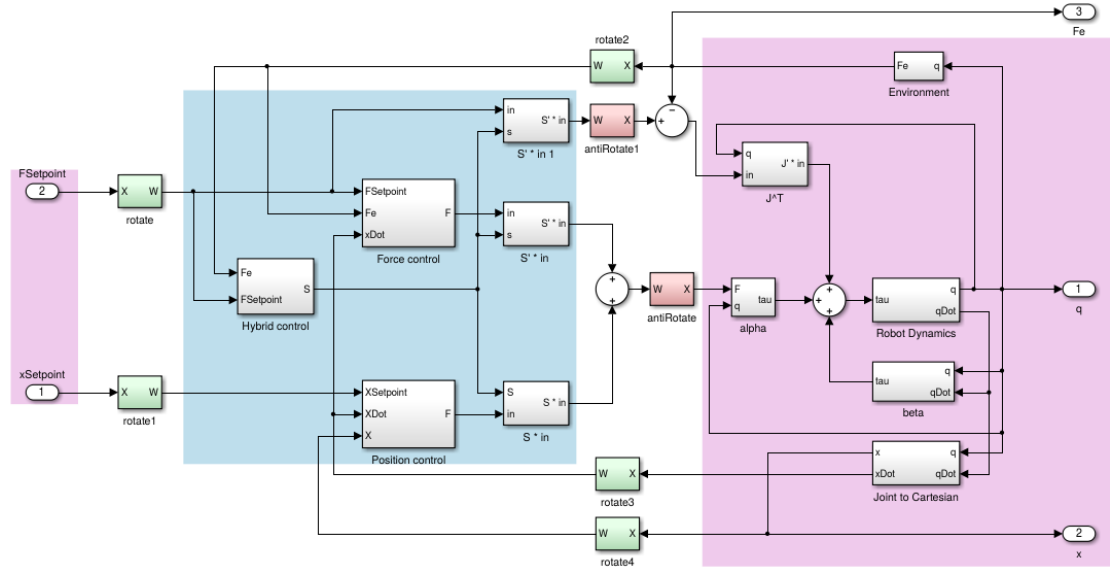
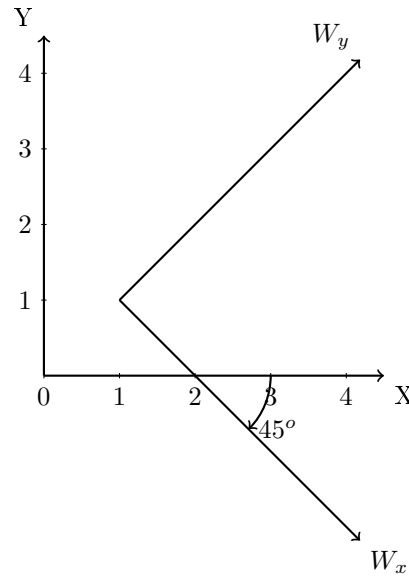


Figura 10: Diagrama en bloques para el control híbrido `my_robot_3_v2.slx`

Para el control híbrido, se combinó los dos controles anteriores, controlados por un controlador adicional: el mismo selecciona si se hace únicamente control de fuerza, de posición o híbrido.

En esta ocasión, para tener desacoplados el control de fuerza y de posición sobre la pared, se trabaja sobre un marco de referencia rotado, en el cual  $\hat{y}' = \hat{n}$ ,  $\hat{z}' = \hat{z}$ . Tanto estos dos bloques de control como la dinámica del robot y los *setpoints* se mantienen: se hace una conversión para pasar de uno a otro  ${}^X_W R$  y su inversa, de manera que todo el control se realiza en el marco de referencia de la pared (celeste), donde fuerza y posición se encuentran desacoplados, pero el resto de los bloques sigue funcionando con las coordenadas de la base (rosa).



Figura 11: Rotación  ${}^X_W R$ 

Se define una trayectoria por tramos, uno inicial en aire libre, hasta un punto de referencia en la pared (7), y luego otro tramo tangente a la misma (8). Se evita llegar al final de la pared por ser coincidente con una singularidad del brazo robótico.

$$[0,5 \quad 0,5] \rightarrow [1 \quad 1] \quad (7)$$

$$[1 \quad 1] \rightarrow [1,7 \quad 0,3] \quad (8)$$

La lógica del controlador consiste de lo siguiente: mientras no se sense ninguna fuerza externa, se realiza un control de posición puro. En cuanto se empieza a sensor una fuerza, se pasa a control de fuerza puro hasta acercarse lo suficiente al *setpoint*, para pasar a un control mixto: en el eje normal se controla fuerza, y en el tangencial posición.

Matemáticamente, estas tres situaciones son modeladas modificando la matriz  $S$  y  $S'$  acordeamente, donde  $S$  controla posición y  $S'$  fuerza, como se detalla en el cuadro 2.

Caso	S		S'	
Posición	1	0	0	0
	0	1	0	0
Fuerza <sup>1</sup>	0	0	1	0
	0	0	0	1
Híbrido	1	0	0	0
	0	0	0	1

Cuadro 2: Control híbrido

Se introdujo a su vez una pequeña histéresis para evitar que estando en la pared oscilando al rededor de valor de fuerza prefijado saltara entre un control y otro, causando sobrepicos de fuerza y/o velocidad.

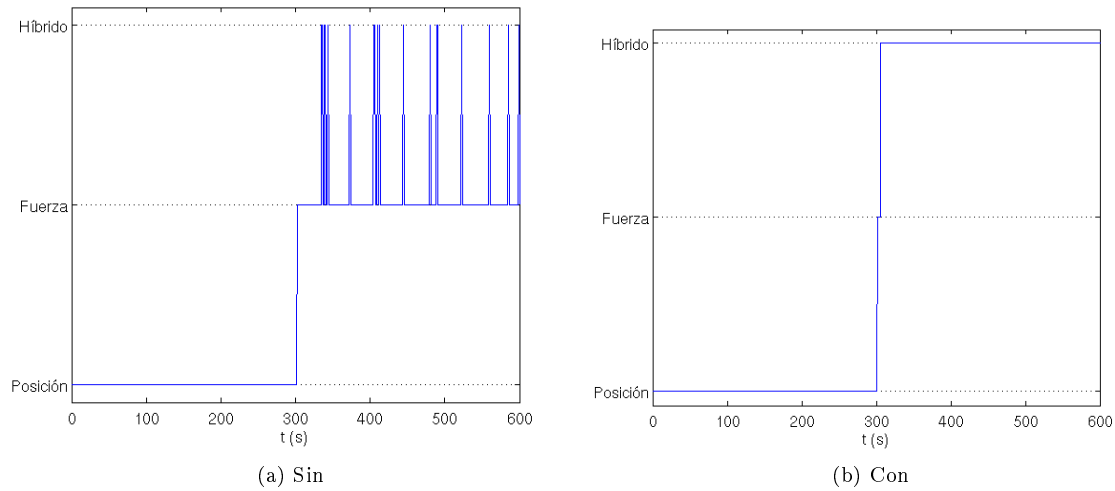


Figura 12: Histéresis

Para correr esta simulación y obtener los gráficos, puede correrse el archivo **Ejercicio3.m**.

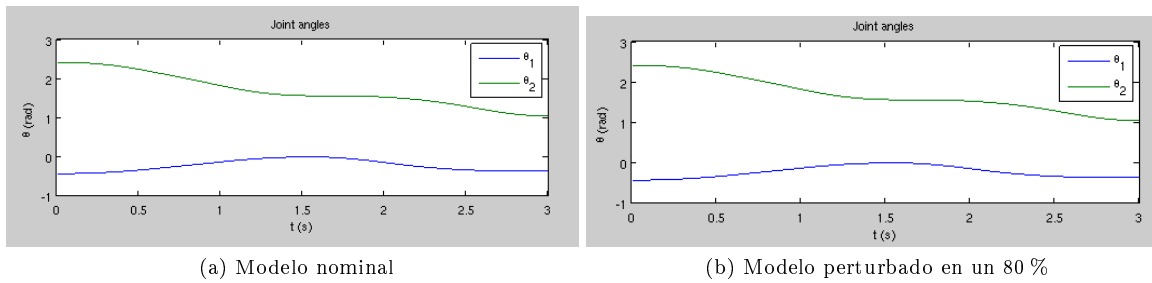
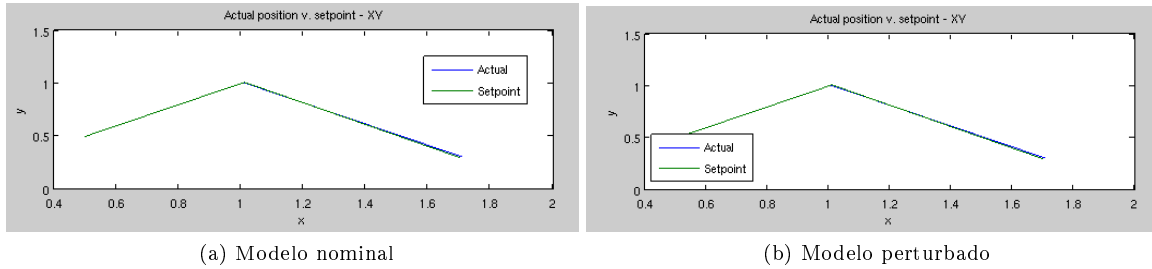
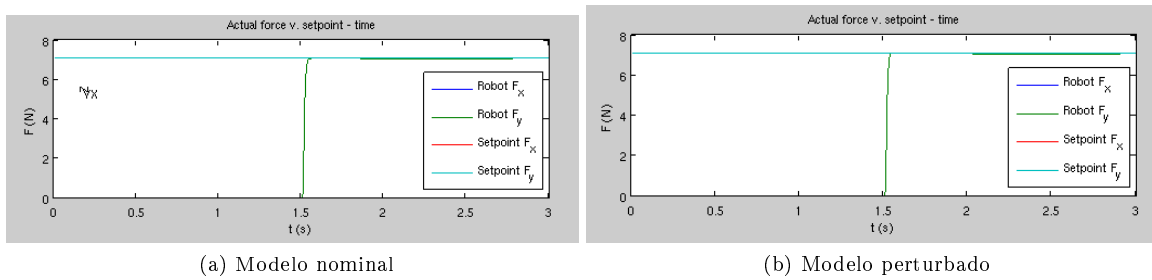


Figura 13: Ángulos de *joint* en función del tiempo

Figura 14: Posición v. *setpoint*, vista *overhead*Figura 15: Fuerza v. *setpoint* en función del tiempo

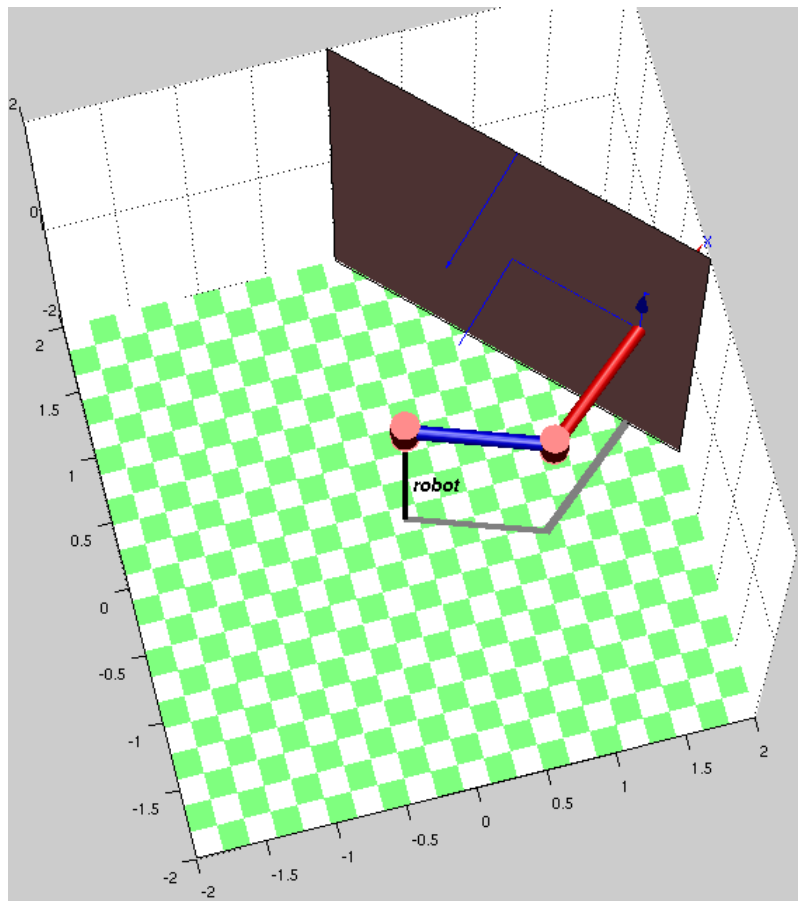


Figura 16: Movimiento

## Referencias

- [1] John J. Craig. *Introduction to Robotics*. Pearson/Prentice Hall, 3rd edition, 2005.