

# Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas



## Reporte de Practicas de la materia de control de sistemas robóticos

Por: Carlos Antonio Dzohl Vega González

Curso de recuperación

Ciudad de México lunes, 14 de  
septiembre de 2020

Profesor: Guillermo Puriel Gil

# Índice

## Contenido

Índice .....	2
Introducción Practica 1: Péndulo simple en simscape multibody .....	4
Metodología .....	4
Diseño y construcción del péndulo robot .....	6
Validación del modelo.....	6
Esquemas de control .....	7
Resultados .....	7
Conclusiones .....	9
Introducción Practica 2: Seguimiento de trayectoria de onda senoidal .....	10
Metodología .....	10
Esquema de control .....	11
Resultados .....	12
Conclusiones .....	13
Introducción Practica 3: Segundo seguimiento de trayectoria, polinomio de tercer grado.....	13
Metodología .....	13
Esquema de control .....	14
Resultados .....	14
Conclusiones.....	16
Introducción Practica 4: Tercer seguimiento de trayectoria, polinomio de quinto grado .....	16
Metodología .....	16
Esquema de control .....	19
Resultados .....	19
Conclusiones.....	22
Introducción Practica 5: Doble péndulo robot en simscape multibody .....	22
Metodología .....	22
Diseño y construcción .....	24
Validación .....	25
Esquema de control .....	25
Resultados .....	26
Practica 6: Instalación del complemento simscape multibody link .....	28

Practica 7: Exportar un ensamble a Matlab .....	28
Introducción Practica 8: Cuarto seguimiento de trayectoria, plano cartesiano .....	28
Metodología .....	28
Modelo dinámico .....	28
Diseño y construcción .....	29
Validación .....	30
Diseño de la trayectoria .....	31
Esquema de control .....	31
Resultados .....	32
Bibliografía .....	34

## Introducción Practica 1: Péndulo simple en Simscape Multibody

Creación de un controlador PD con compensación de gravedad de posición y utilización de Simscape Multibody. La validación de la construcción del péndulo con su control utilizando sus ecuaciones dinámicas.

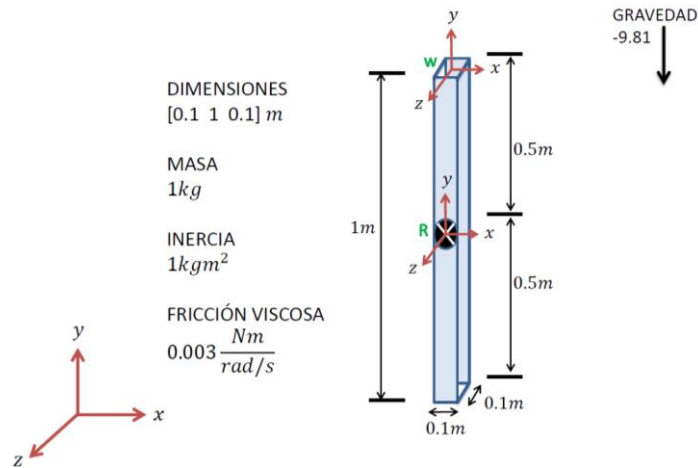
Objetivos:

Crear a partir de Simscape Multibody un péndulo simple y validarlo con sus ecuaciones dinámicas.

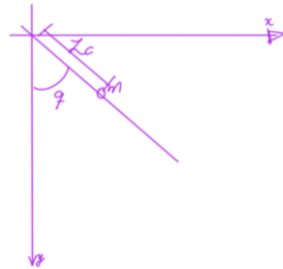
Realizar un controlador de posición con control P con retroalimentación de velocidad, validar esto con sus ecuaciones dinámicas.

## Metodología

Primero se obtiene el modelo dinámico del péndulo con los siguientes parámetros:



Para la obtención del modelo dinámico se realizan las ecuaciones de Euler-Lagrange con sus respectivos parámetros



Con posiciones

$$x = l * c * \sin(q)$$

$$y = l * c * \cos(q)$$

Con velocidades

$$\dot{x} = l * c * \cos(q) * \dot{q}$$

$$\dot{y} = l * c * \sin(q) * \dot{q}$$

$$v^2 = \dot{x}^2 + \dot{y}^2$$

$$\dot{x}^2 = l^2 * c^2 * \cos^2(q) * \dot{q}^2$$

$$\dot{y}^2 = l^2 * c^2 * \sin^2(q) * \dot{q}^2$$

$$v^2 = l^2 * c^2 * \cos^2(q) * \dot{q}^2 + l^2 * c^2 * \sin^2(q) * \dot{q}^2$$

$$v^2 = l^2 * c^2 * \dot{q}^2 * [\cos^2(q) + \sin^2(q)]$$

Energía cinética

$$E_c = \frac{1}{2} m v^2 + \frac{1}{2} I \dot{q}^2$$

$$E_c = \frac{1}{2} m \dot{q}^2 + \frac{1}{2} I \dot{q}^2$$

$$E_c = \frac{1}{2} \dot{q} (m l c^2 + I)$$

Energía potencial

$$h = l c - l c \cos(q)$$

$$h = l c (1 - \cos(q))$$

$$E_p = m g l c (1 - \cos(q))$$

Lagrangiano

$$\mathcal{L} = E_c - E_p$$

$$\mathcal{L} = \frac{1}{2} (m l c^2 + I) \dot{q}^2 - m g l c (1 - \cos(q))$$

Ecuaciones de movimiento

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} = \tau - f_f(\dot{q})$$

$$\frac{\partial \mathcal{L}}{\partial \dot{q}} = (m l c^2 + I) \dot{q}$$

$$\frac{\partial \mathcal{L}}{\partial q} = -m g l c \sin(q)$$

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}} = (m l c^2 + I) \ddot{q}$$

$$(mlc^2 + I)\ddot{q} + mglcsen(q) = \tau - f_f(\dot{q})$$

$$f_f(\dot{q}) = \beta\dot{q} + f_csgino(\dot{q})$$

$$(mlc^2 + I)\ddot{q} + mglcsen(q) + \beta\dot{q} + f_csgino(\dot{q}) = \tau$$

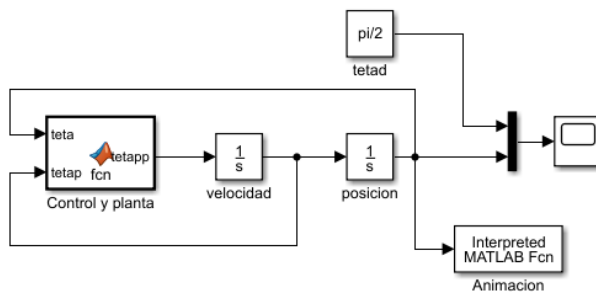
Ecuación de planta:

$$\ddot{q} = \frac{\tau - mglcsen(q) - \beta\dot{q} - f_csgino(\dot{q})}{(mlc^2 + I)}$$

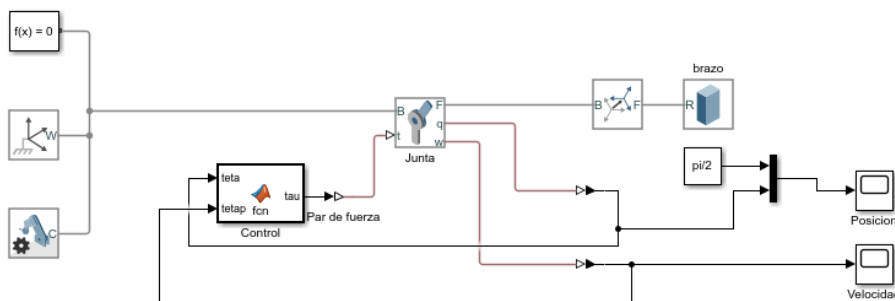
### Diseño y construcción del péndulo robot

Para la validación de la plantase construyó la simulación con los siguientes bloques, el código del control y la planta se encuentran el repositorio de Github en :

<https://github.com/xDzohlx/Controldesistemasroboticos/blob/master/Practica1/Control.m>



El modelo de simscape multibody:



Se muestran todos los bloques que se utilizaron para la práctica,

### Validación del modelo

Para la validación del modelo de simscape multibody comprobamos con valores iniciales en el segundo integrador y observamos como se comporta con la grafica conectada en la salida y como referencia la posición final que es una constante.

Esta la comparamos con el modelo hecho con la herramienta de Simscape Multibody utilizando los mismos parámetros y con esto comprobamos que no exista discrepancia entre un modelo y el otro para esto se utilizan las mismas condiciones iniciales en ambos y mismos parámetros en la planta condiciones iniciales en el integrador y el modelo en la revoluta.

### Esquemas de control

El controlador que se utiliza es un proporcional con retroalimentación de velocidad por lo que no toma en cuenta el modelo para llevar el control mide errores de posición y velocidades con:

$$\tau = k_p e - k_d \dot{q}$$

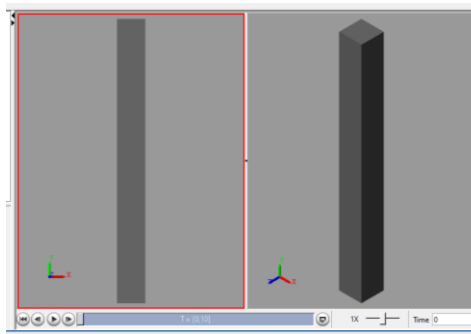
$$e = q_d - q$$

Las ganancias utilizadas fueron altas ya que este controlador tiene error estacionario por lo que se compensa con ganancias muy altas no muy factible para un actuador físico.

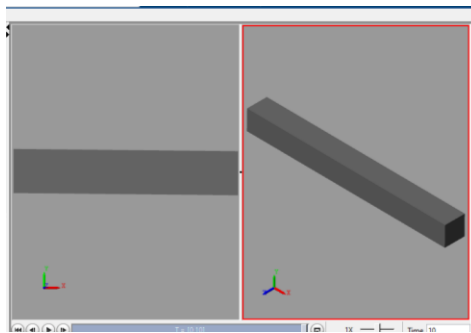
### Resultados

Para la referencia de  $\pi/2$  se muestran las siguientes gráficas son el desempeño del controlador.

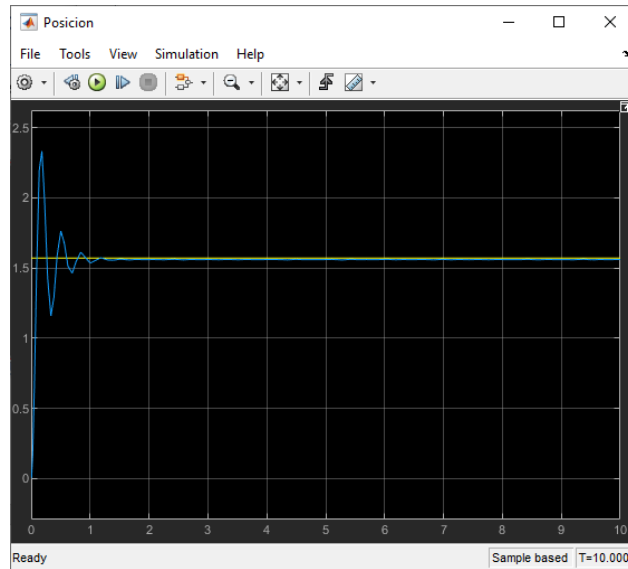
Posición inicial



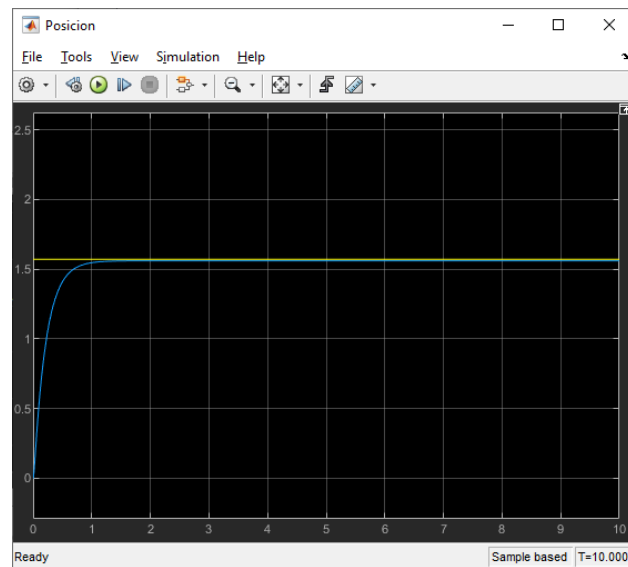
Posición final



Gráfica de posición contra referencia tiene bastante oscilación, pero con poco error estacionario con el modelo de Simscape Multibody  $K_p = 450$ ,  $K_d = 100$ ;

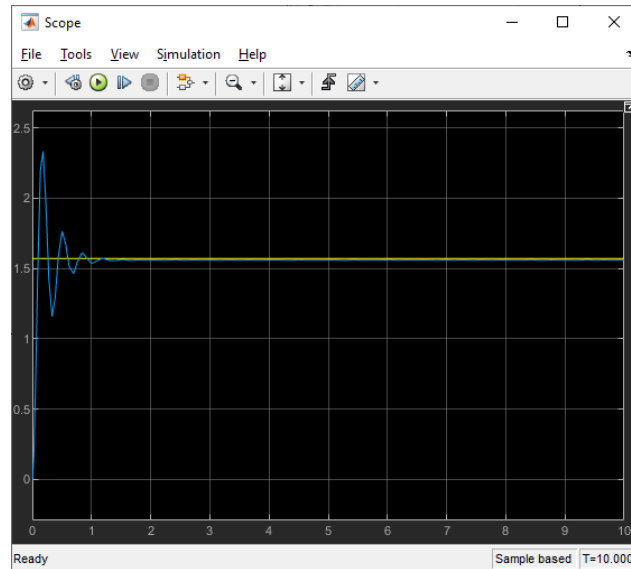


Posición contra referencia  $K_p=450$ ,  $K_d=100$ ;

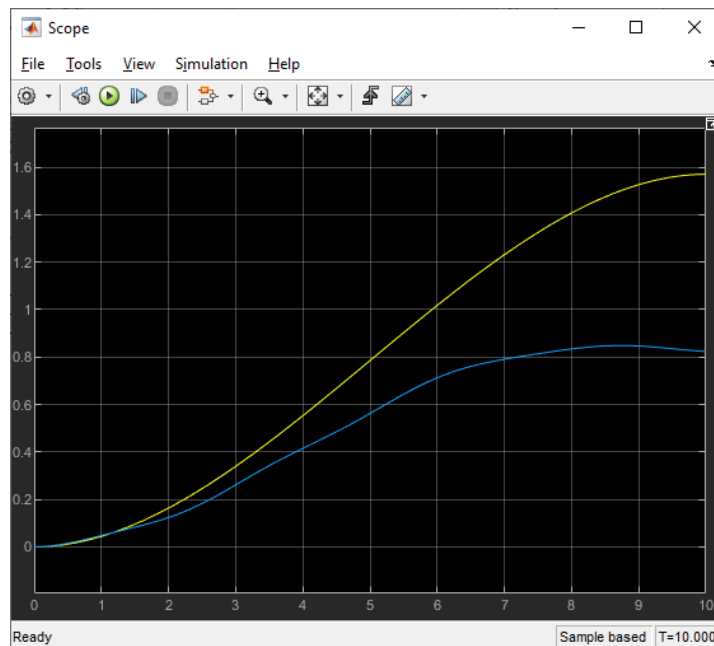


Gráfica de posición contra referencia tiene bastante oscilación, pero con poco error estacionario con la planta simulada con sus ecuaciones dinámicas  $K_p= 450$ ,  $K_d=100$ ;





Este controlador requiere ganancias muy altas para evitar el error estacionario.



## Conclusiones

Esta práctica fue la introductoria, la creación del modelo en Simscape Multibody fue sencillo y ya teniendo la planta la validación de este modelo es igualmente fácil. El control de posición para nuestro controlador no tuvo ningún problema más que el error estacionario, pero arreglado con altas ganancias de  $k_p$  y  $k_d$  para que no tenga mucho error estacionario ni oscilaciones.

## Introducción Practica 2: Seguimiento de trayectoria de onda senoidal

Creación de un controlador PD con compensación de gravedad para seguimiento de trayectoria, la trayectoria se crea a partir de funciones senoidales y su validación con sus respectivas ecuaciones dinámicas.

Objetivos:

Crear un seguimiento de trayectorias con un péndulo simple

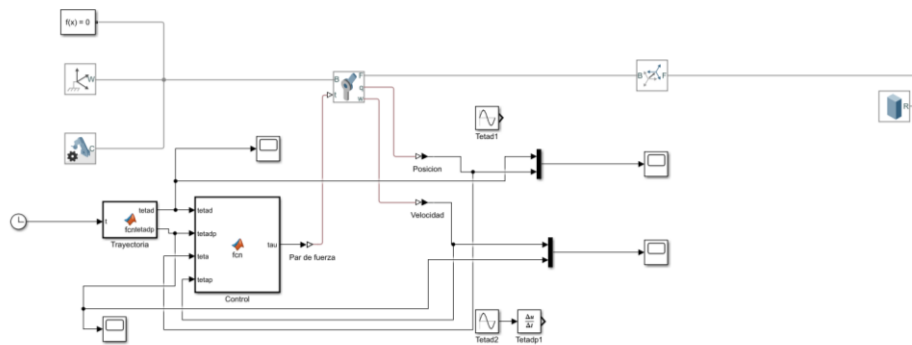
Generar varias trayectorias a partir de funciones senoidales

Crear un controlador de seguimiento de trayectoria proporcional derivativo con compensación de gravedad.

### Metodología

El modelo dinámico es idéntico al de la práctica 1 que se encuentra arriba.

Creamos el modelo del péndulo simple con los mismos parámetros anteriores de la práctica 1 con la herramienta de Simscape Multibody. El diagrama a bloques final de la práctica se muestra a continuación.

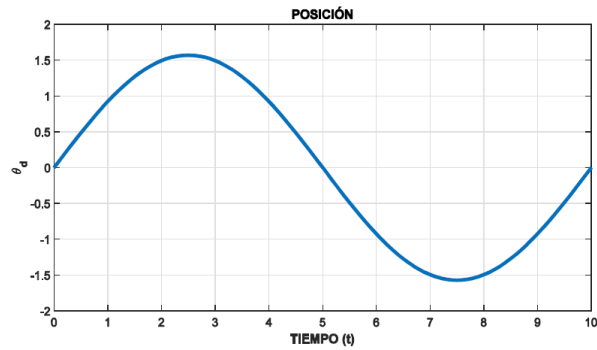


Para la trayectoria se proponen dos sinusoidales para que el péndulo las siga.

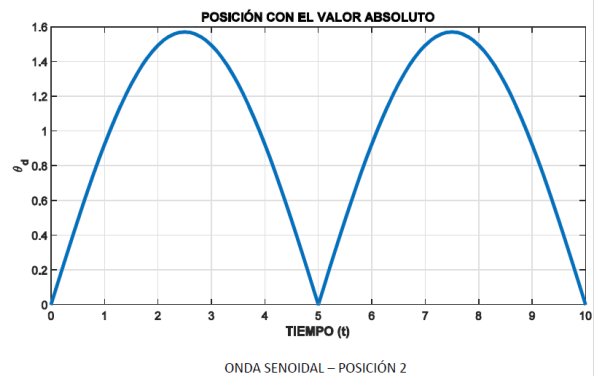
El código de las trayectorias generadas se encuentra en:

<https://github.com/xDzohlx/Controldesistemasroboticos/blob/master/Practica2/Trayectoria.m>

Senoidal



Senoidal de valores absolutos:



### Esquema de control

Para el seguimiento de trayectoria se utilizaron dos controladores

El control proporcional derivativo

$$\tau = k_p e + k_d e_p$$

$$e = \theta_d - \theta$$

$$e_p = \dot{\theta}_d - \dot{\theta}$$

Y el control proporcional derivativo con compensación de gravedad

$$\tau = k_p e + k_d e_p + G(\theta)$$

$$e = \theta_d - \theta$$

$$e_p = \dot{\theta}_d - \dot{\theta}$$

$$G(\theta) = m * g * lc * \sin(\theta)$$

El código del controlador se encuentra en:

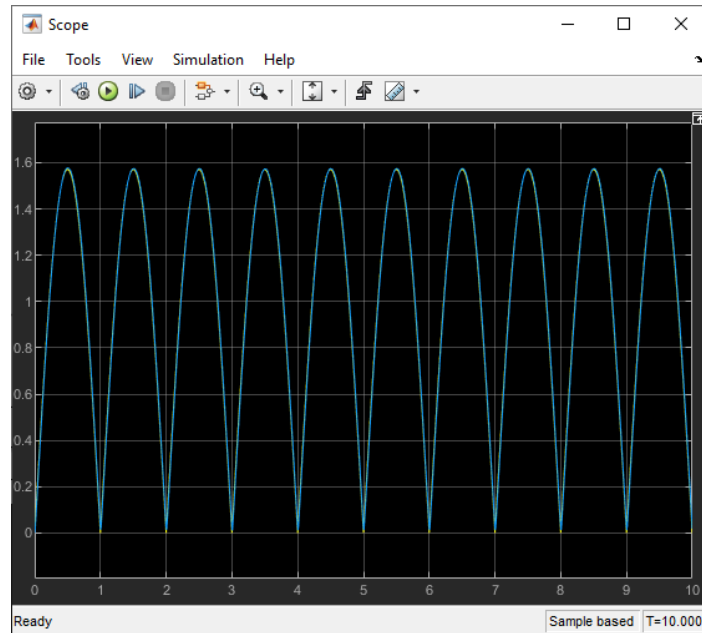
<https://github.com/xDzohlx/Controldesistemasroboticos/blob/master/Practica2/Control.m>

Este se programo en una función de Matlab y se integro en la simulación de simulink.

## Resultados

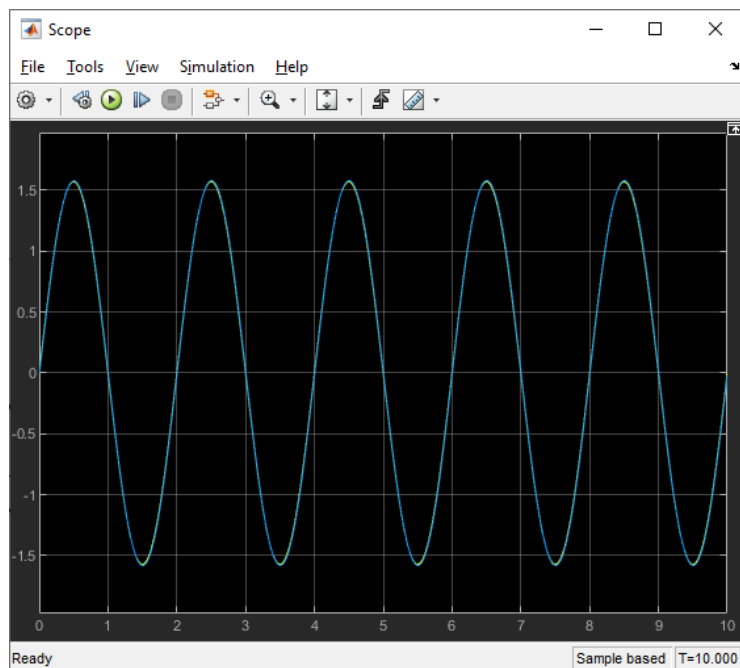
En este control de trayectoria las posiciones finales e iniciales no son tan importantes la planta y el modelo se simscape multibody es exactamente igual al de la practica anterior

Grafica del seguimiento de trayectoria de una función seno con valores absolutos utilizando el modelo de simscape multibody. Con ganancias de  $K_p=350$  y  $K_d=250$ .



La trayectoria esta sobrepuesta y se nota que sigue esta perfectamente.

Para el seguimiento de la trayectoria senoidal la grafica siguiente muestra el resultado con las mismas ganancias de  $K_p=350$  y  $K_d=250$ .



## Conclusiones

Ya estos controladores funcionan mejor para el control de trayectorias a pesar de tener ganancias tan grandes estas fueron menores que en la práctica anterior e igualmente el error estacionario fue menor. Pero uno de los problemas por los que este controlador requiere ganancias tan grandes son los cambios súbitos de posición que provocan cambios bruscos de velocidad que el controlador hace que se cumplan por lo que se requiere de estas ganancias tan grandes

## Introducción Practica 3: Segundo seguimiento de trayectoria, polinomio de tercer grado

Creación de controladores por precompensación para seguimiento de trayectoria de un péndulo simple creado con Simscape Multibody y validado con sus respectivas ecuaciones dinámicas. Las trayectorias se crean a partir de un polinomio de tercer grado.

Objetivos:

Crear un seguimiento de trayectoria con un péndulo simple

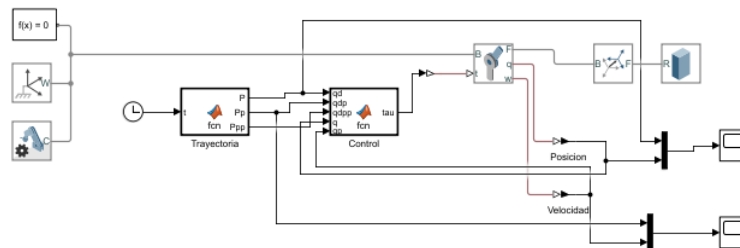
Generar varias trayectorias a partir de un polinomio de tercer grado

Crear dos controladores uno por precompensación (lazo abierto), y otro proporcional derivativo con precompensación (lazo cerrado).

## Metodología

El modelo dinámico es idéntico al de la práctica 1 que se encuentra arriba.

Para la construcción del modelo del péndulo simple de igual forma utilizamos la herramienta de Simscape Multibody y creamos los siguientes bloques con los mismos parámetros que hemos estado utilizando.

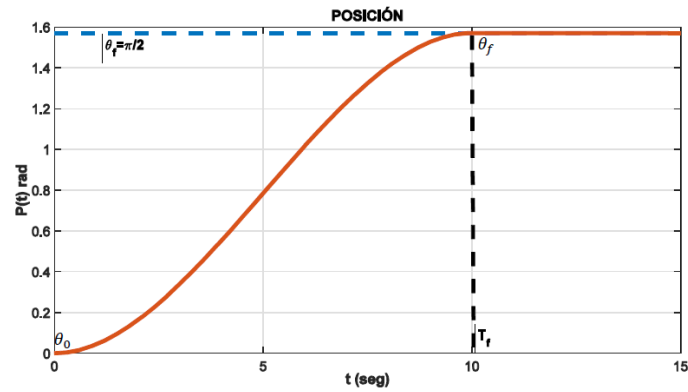


Para la trayectoria creamos un polinomio de tercer grado el código se encuentra en el git:

<https://github.com/xDzohlx/Controldesistemasroboticos/blob/master/Practica3/Trayectoria.m>

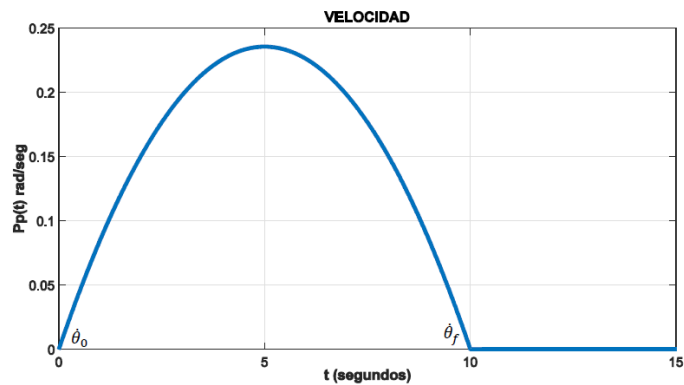
Este cumple con las siguientes gráficas, primero la de posición:

Seguimiento de Trayectoria II, Polinomio de 3er Grado.



Y la de velocidad:

Seguimiento de Trayectoria II, Polinomio de 3er Grado.



### Esquema de control

Utilizamos dos controladores

El control por precompensación el cual es un lazo abierto el cual mueve el péndulo dependiendo de sus parámetros por lo que un pequeño cambio en estos puede cambiar mucho la trayectoria.

Con:

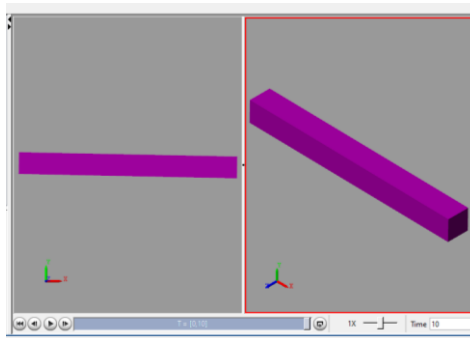
$$\tau = M(q)\ddot{q}_d + C(q_d + \dot{q}_d)\dot{q}_d + B\dot{q}_d + G(q_d)$$

Y el control PD con precompensación que y agrega el modelo dinámico inverso de la precompensación con un control proporcional derivativo con:

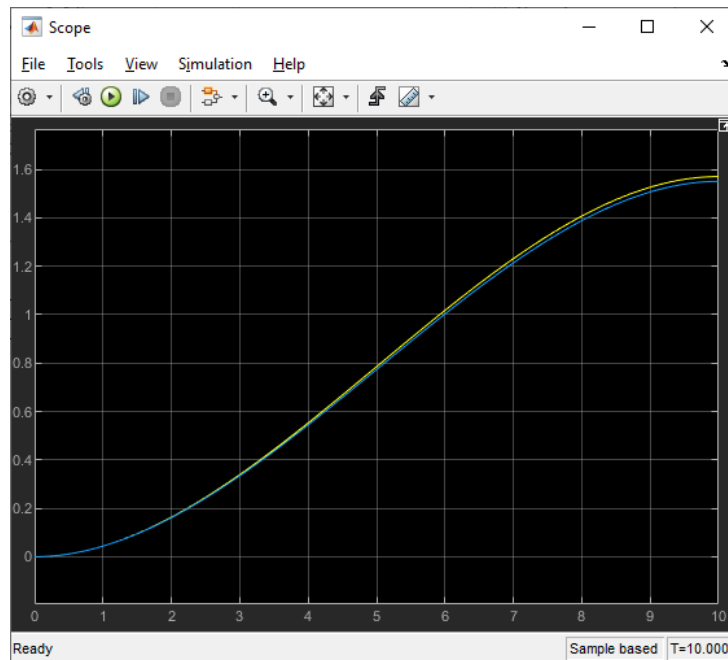
$$\tau = k_p \tilde{q} - k_d \dot{\tilde{q}} + M(q)\ddot{q}_d + C(q_d + \dot{q}_d)\dot{q}_d + B\dot{q}_d + G(q_d)$$

### Resultados

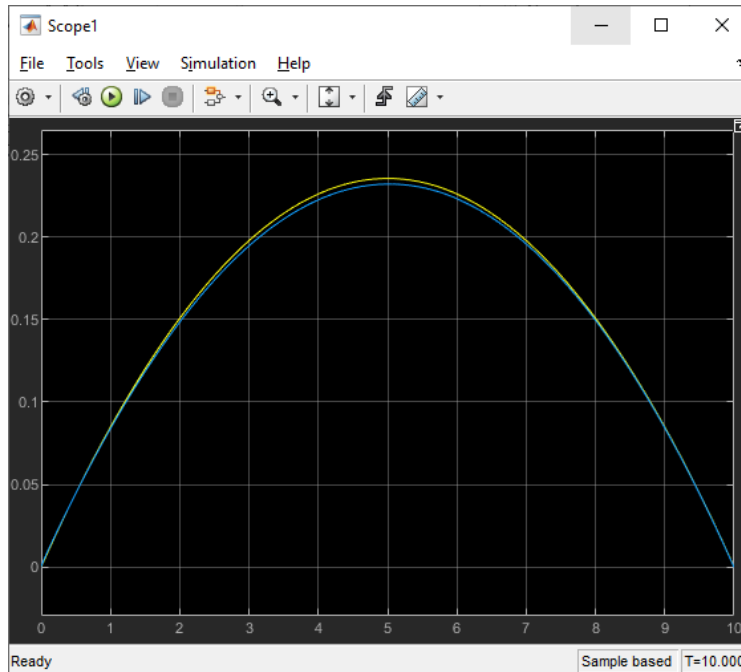
Posición final el péndulo empieza desde abajo.



Seguimiento de trayectoria con control PD con precompensación variando un poco los parámetros, con el momento de inercia a la mitad y 10% más pesada, con  $K_p=80$ ,  $K_d=35$ .



Seguimiento de trayectoria con control por precompensación variando un poco los parámetros, con el momento de inercia a la mitad y 10% más pesada, este control no cuenta con retroalimentación se encuentra en lazo abierto.



Para la velocidad se nota cuando es que se pierde esta que es justo en medio, pero se recupera al final de la trayectoria

### Conclusiones

El controlador por precompensación ofrece un seguimiento de trayectoria muy fiel a la dinámica ya que esta basado en los parámetros y la dinámica del péndulo pero si se cambian los parámetros al no tener retroalimentación la trayectoria resultante puede ser muy diferente al esperado por lo que el agregarle el control pd ayuda con este problema así que con la combinación de estos se tiene un seguimiento de trayectoria muy fiel y muy poco error estacionario.

## Introducción Practica 4: Tercer seguimiento de trayectoria, polinomio de quinto grado

Creación de controladores y trayectorias para un péndulo simple creado con Simscape Multibody y validado con sus respectivas ecuaciones dinámicas. La trayectoria es creada a partir de un polinomio de quinto grado para poder agregar aceleración al controlador.

### Objetivos:

Crear un seguimiento de trayectoria con un péndulo simple

Generar varias trayectorias a partir de un polinomio de quinto grado para especificar posición velocidad y aceleración.

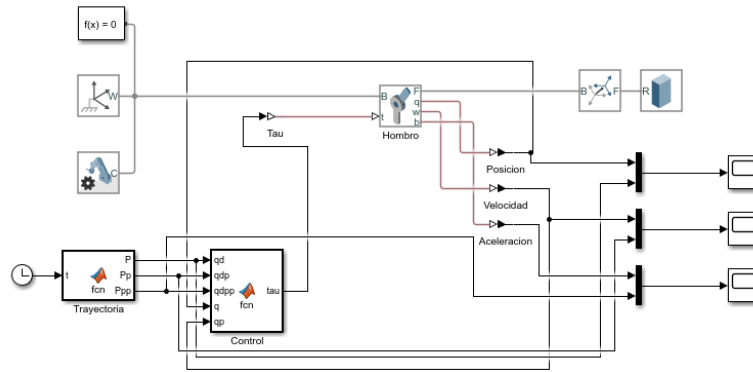
Crear dos controladores uno proporcional derivativo +, y por par calculado.

### Metodología

El modelo dinámico es idéntico al de la práctica 1 que se encuentra arriba.

Para la construcción del modelo en Simscape Multibody se creó el siguiente modelo.





El desarrollo del polinomio es el siguiente:

$$P(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5,$$

**Para la POSICIÓN**

$$\theta_0 = a_0$$

$$\theta_f = a_0 + a_1t_f + a_2t_f^2 + a_3t_f^3 + a_4t_f^4 + a_5t_f^5,$$

**Para la VELOCIDAD**

$$\dot{P}(t) = a_1 + 2a_2t + 3a_3t^2 + 4a_4t^3 + 5a_5t^4,$$

$$\dot{\theta}_0 = a_1,$$

$$\dot{\theta}_f = a_1 + 2a_2t_f + 3a_3t_f^2 + 4a_4t_f^3 + 5a_5t_f^4,$$

**Para la ACCELERACIÓN**

$$\ddot{P}(t) = 2a_2 + 6a_3t + 12a_4t^2 + 20a_5t^3,$$

$$\ddot{\theta}_0 = 2a_2,$$

$$\ddot{\theta}_f = 2a_2 + 6a_3t_f + 12a_4t_f^2 + 20a_5t_f^3,$$

Y con esto tenemos:

$$\theta_0 = a_0$$

$$\theta_f = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5,$$

Restricciones de la posición

$$\dot{\theta}_0 = a_1,$$

$$\dot{\theta}_f = a_1 + 2a_2 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4,$$

Restricciones de la velocidad

$$\ddot{\theta}_0 = 2a_2,$$

$$\ddot{\theta}_f = 2a_2 + 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3,$$

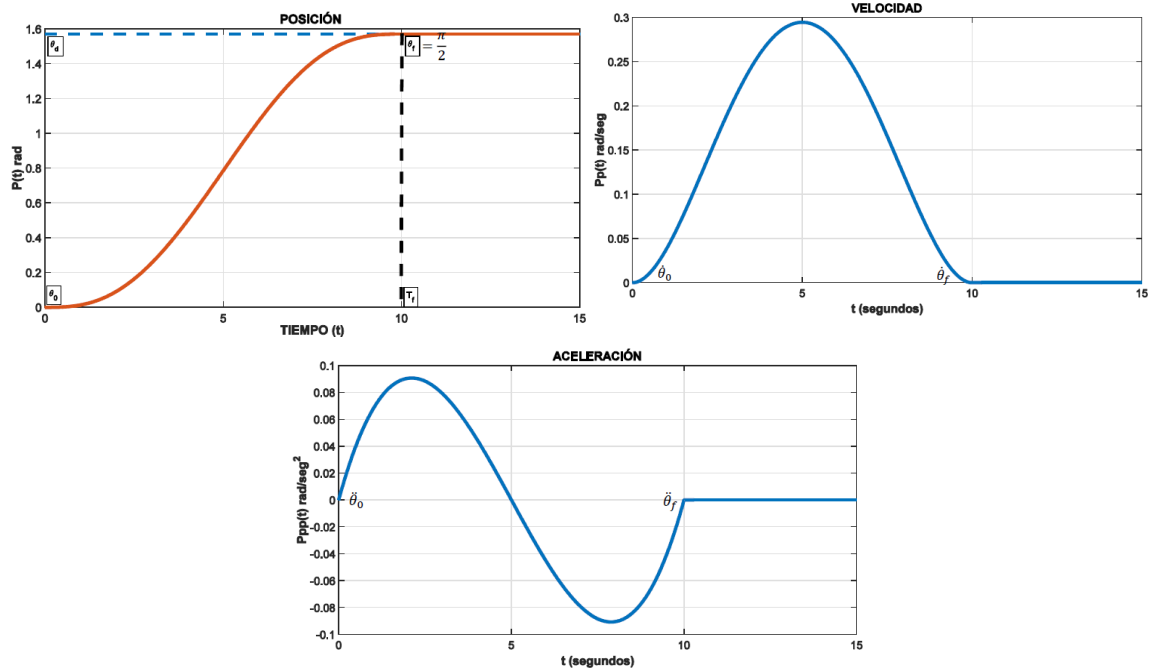
Restricciones de la aceleración

Para la creación de la trayectoria se utilizó un polinomio de quinto grado el código se encuentra en el siguiente git:

<https://github.com/xDzohl/Controldesistemasroboticos/blob/master/Practica%204/Trayectoria.m>

Este polinomio responde a las siguientes posiciones, velocidades y aceleración.

Seguimiento de Trayectoria caso (III), Polinomio de 5to Grado.



Este polinomio es requerido para tener aceleración en el controlador ya que si el polinomio es menor la aceleración se vuelve constante.

## Esquema de control

En esta práctica utilizamos dos controladores para el seguimiento de trayectorias.

El control de par calculado

Este se basa en el comportamiento dinámico, este desacopla los eslabones para evitar interacciones entre ellos y esta dado por:

$$\tau = M(q)[\ddot{q}_d + K_d\dot{\tilde{q}} + K_p\tilde{q}] + C(q + \dot{q})\dot{q} + B\dot{q} + G(q)$$

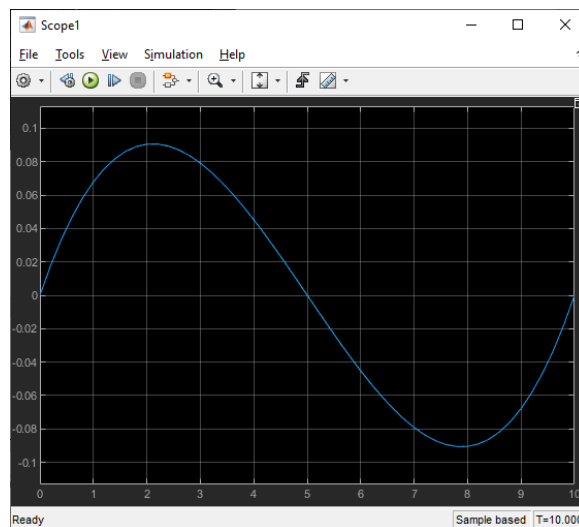
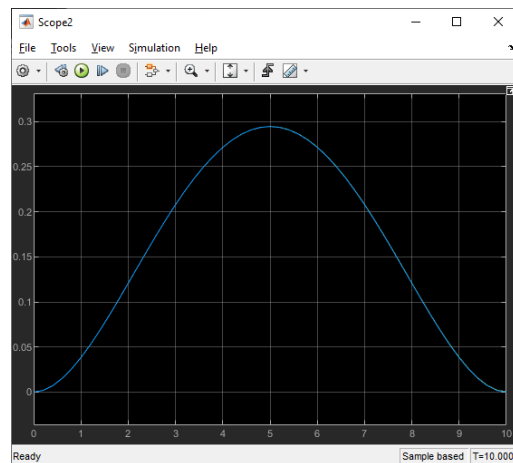
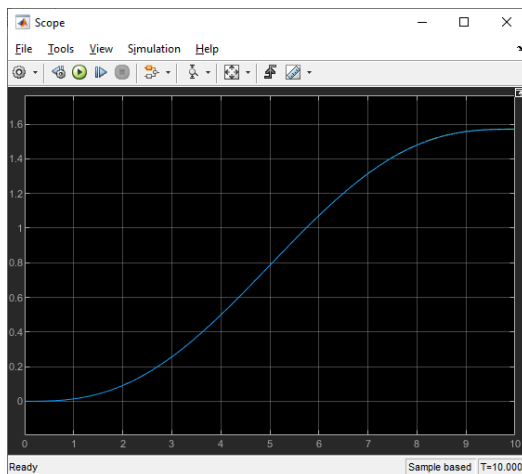
Controlador PD+

Este es un controlador muy general este cuenta con el comportamiento del modelo dinámico y la regulación del control proporcional derivativo dado por:

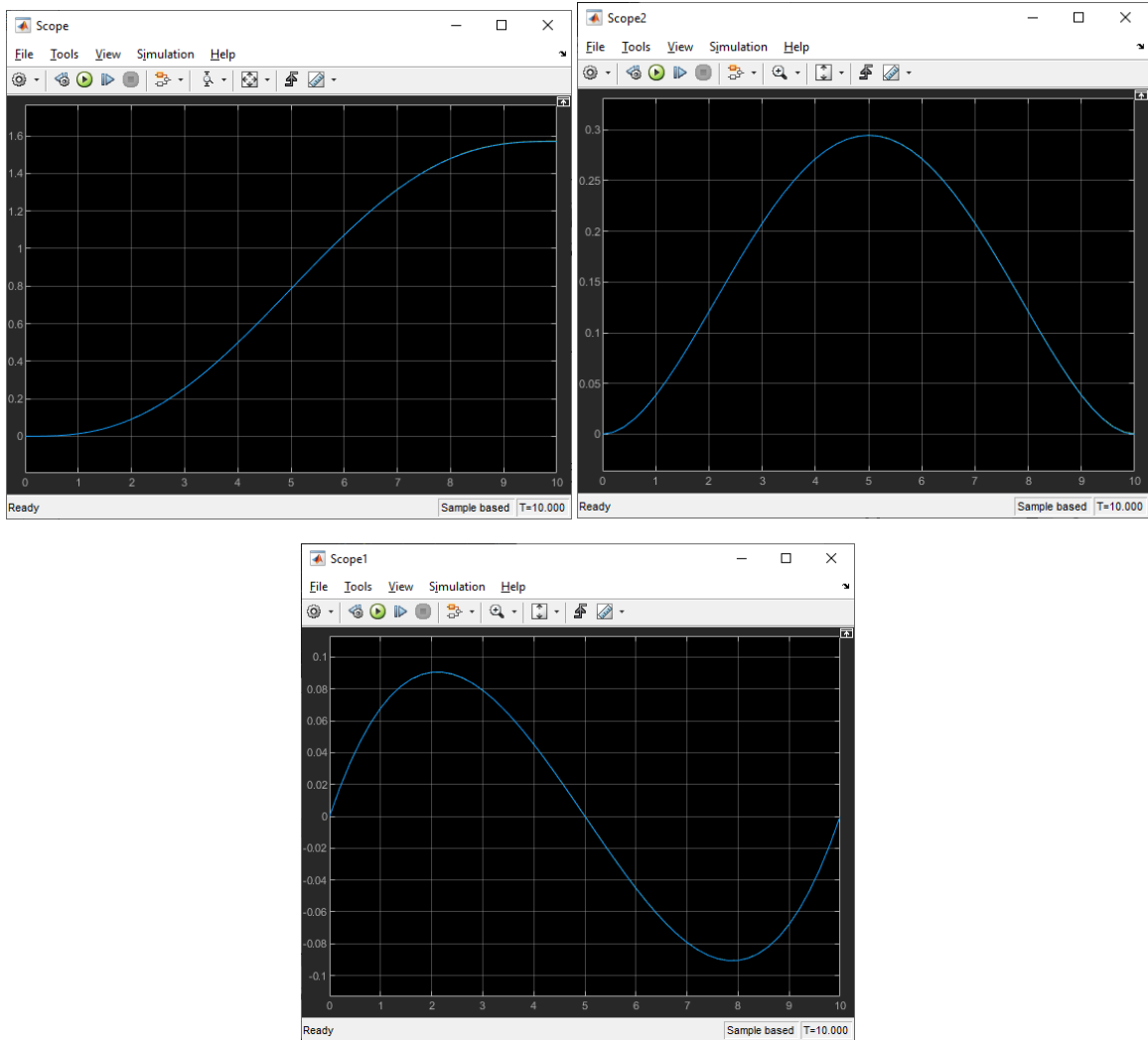
$$\tau = K_p\tilde{q} + K_d\dot{\tilde{q}} + M(q)\ddot{q}_d + C(q + \dot{q})\dot{q}_d + B\dot{q}_d + G(q)$$

## Resultados

Con los mismos parámetros, Con  $K_p, K_d=1$ ; y utilizando el controlador PD+ Las gráficas muestran la posición, velocidad y aceleración respectivamente.

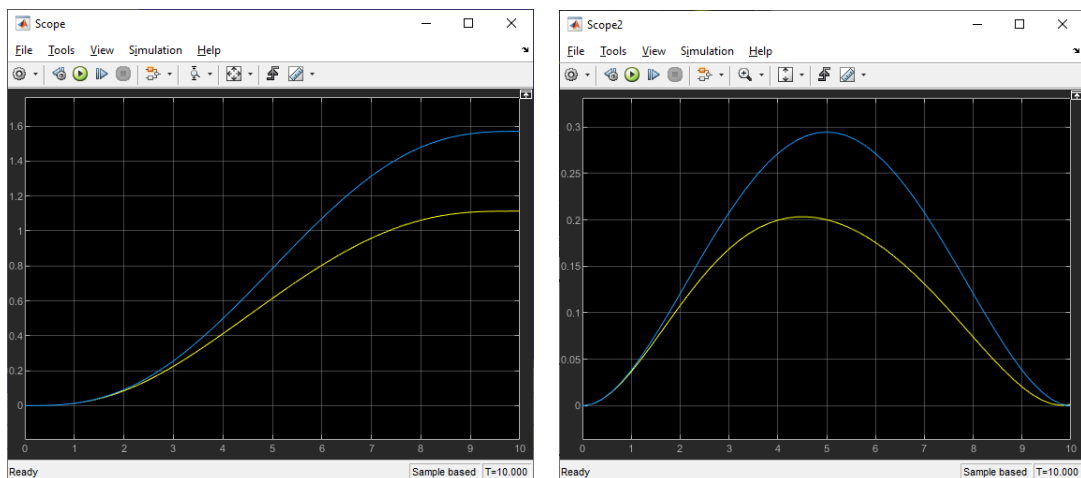


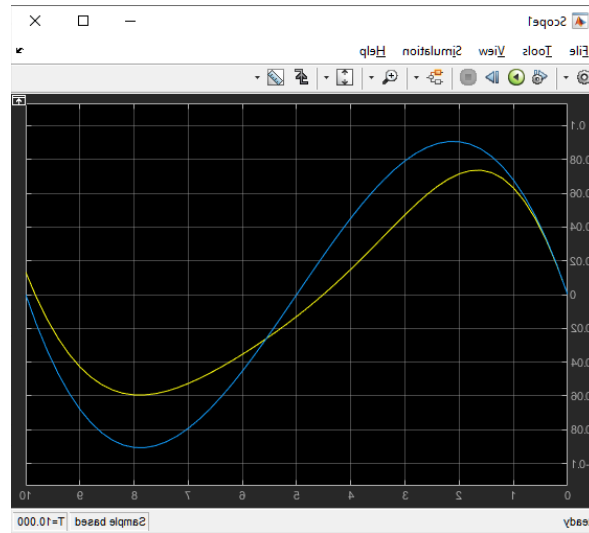
Controlador de par calculado con  $k_p$ ,  $k_d=1$ ;



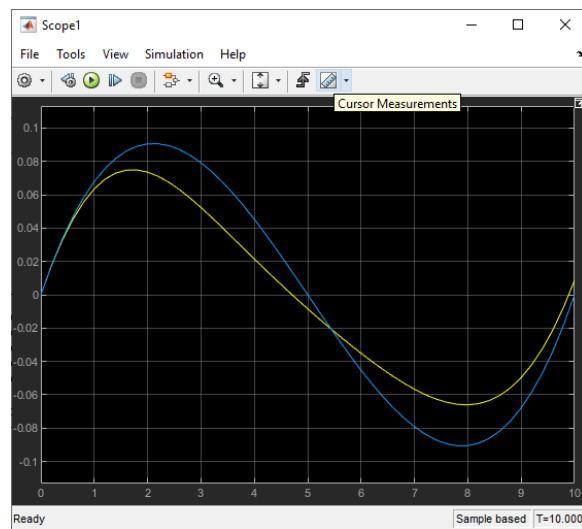
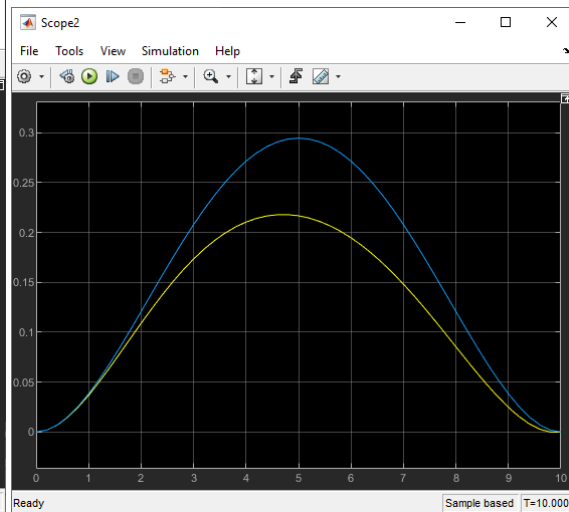
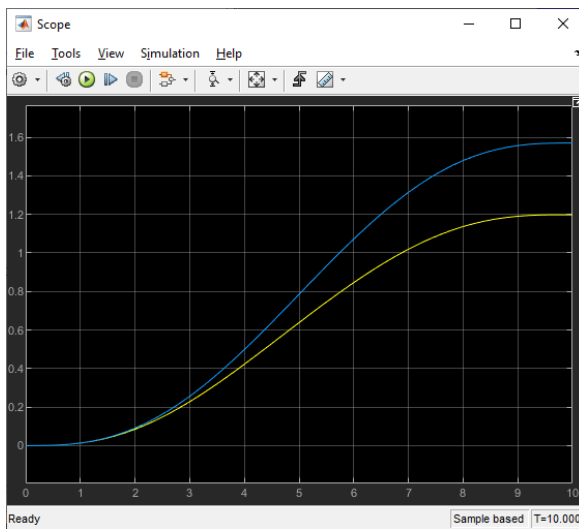
Ambos controladores responden de forma ideal al estar basado en la dinámica del modelo.

Si cambiamos la masa del bloque 10% los controladores quedan de la siguiente forma con PD+





Controlador de par calculado con  $k_p, k_d=1$ ;



## Conclusiones

Estos controladores son muy fieles a la trayectoria y como están basados en la dinámica del péndulo el error estacionario es muy bajo con los cambios en la masa y las ganancias no tienen que ser tan grandes para compensar los defectos físicos o la precisión de nuestras medidas a pesar de que cambien un poco los parámetros estos se pueden ajustar únicamente con las ganancias.

## Introducción Practica 5: Doble péndulo robot en Simscape Multibody

Creación de varios controladores de posición y trayectoria de un doble péndulo robot creado en Simscape Multibody y validarlo con sus respectivas ecuaciones dinámicas. Las trayectorias creadas son por polinomios de tercer y quinto orden.

### Objetivos:

Crear a partir de Simscape Multibody un doble péndulo robot y validarlo con sus ecuaciones dinámicas.

Crear varias trayectorias con polinomios de tercer y quinto orden.

Crear tres controladores de posición: proporcional con retroalimentación de velocidad y control proporcional con compensación de gravedad.

Crear cuatro controladores de posición y de trayectoria: por precompensación, proporcionar derivativo con precompensación, proporcional derivativo + y de par calculado.

## Metodología

Para la práctica primero debemos de obtener las ecuaciones dinámicas para la validación del modelo en este caso de un doble péndulo robot y obtenemos este modelo dinámico mediante las ecuaciones de Euler-Lagrange como se muestra a continuación.

### Eslabón 1

#### Posiciones

$$x_1 = l_{c1} \sin(q_1)$$

$$y_1 = l_{c1} \cos(q_1)$$

#### Velocidades

$$\dot{x} = l_{c1} \cos(q_1) \dot{q}_1$$

$$\dot{y} = l_{c1} \sin(q_1) \dot{q}_1$$

$$v_1^2 = \dot{x}_1^2 + \dot{y}_1^2$$

$$\dot{x}_1^2 = l_{c1}^2 \cos^2(q_1) \dot{q}_1^2$$

$$\dot{y}_1^2 = l_{c1}^2 \sin^2(q_1) \dot{q}_1^2$$

$$v_1^2 = l_{c1}^2 \cos^2(q_1) \dot{q}_1^2 + l_{c1}^2 \sin^2(q_1) \dot{q}_1^2$$

$$v_1^2 = l_{c1}^2 \dot{q}_1^2 [\cos^2(q_1) + \sin^2(q_1)]$$

Energía cinética

$$E_{c1} = \frac{1}{2}m_1v_1^2 + \frac{1}{2}I_1\dot{q}_1^2$$

$$E_{c1} = \frac{1}{2}m_1lc_1^2\dot{q}_1^2 + \frac{1}{2}I_1\dot{q}_1^2$$

$$E_{c1} = \frac{1}{2}\dot{q}_1^2(m_1lc_1^2 + I_1)$$

Energía potencial

$$h = lc_1 - lc_1\cos(q_1)$$

$$h = lc_1(1 - \cos(q_1))$$

$$E_{p1} = m_1g lc_1(1 - \cos(q_1))$$

Eslabón 2

Posiciones

$$x_2 = l_1\sin(q_1) + lc_2\sin(q_1 + q_2)$$

$$y_2 = -l_1\sin(q_1) - lc_2\cos(q_1 + q_2)$$

Velocidades

$$\dot{x}_2 = l_1\cos(q_1)\dot{q}_1 + lc_2\cos(q_1 + q_2)[\dot{q}_1 + \dot{q}_2]$$

$$\dot{y}_2 = l_1\sin(q_1)\dot{q}_1 + lc_2\sin(q_1 + q_2)[\dot{q}_1 + \dot{q}_2]$$

$$v_2^2 = \dot{x}_2^2 + \dot{y}_2^2$$

$$v_2^2 = l_1^2\dot{q}_1^2 + lc_2^2[\dot{q}_1^2 + 2\dot{q}_1\dot{q}_2 + \dot{q}_2^2] + 2l_1lc_2[\dot{q}_1^2 + \dot{q}_1\dot{q}_2]$$

Energía cinética

$$E_{c2} = \frac{1}{2}m_2v_2^2 + \frac{1}{2}I_2[\dot{q}_1 + \dot{q}_2]^2$$

$$E_{c2} = \frac{1}{2}m_2l_1^2\dot{q}_1^2 + \frac{m_2}{2}[\dot{q}_1^2 + 2\dot{q}_1\dot{q}_2 + \dot{q}_2^2] + m_2l_1lc_2[\dot{q}_1^2 + \dot{q}_1\dot{q}_2]\cos(q_2) + \frac{1}{2}I_2[\dot{q}_1 + \dot{q}_2]^2$$

Energía potencial

$$E_{p2} = -m_2gl_1\cos(q_1) - m_2g lc_2\cos(q_1 + q_2)$$

Lagrangiano

$$\mathcal{L} = E_{c1} + E_{c2} - E_{p1} - E_{p2}$$

$$\begin{aligned}\mathcal{L} = & \frac{1}{2}(m_1lc_1^2 + I_1)\dot{q}_1^2 + \frac{1}{2}m_2l_1^2\dot{q}_1^2 + \frac{m_2}{2}lc_2^2[\dot{q}_1^2 + 2\dot{q}_1\dot{q}_2 + \dot{q}_2^2] + m_2l_1lc_2[\dot{q}_1^2 + q_1\dot{q}_2]\cos(q_2) \\ & + \frac{1}{2}I_2[\dot{q}_1 + \dot{q}_2]^2 + m_2gl_1\cos(q_1) + m_2glc_2\cos(q_1 + q_2) \\ & + m_1glc_1(1 - \cos(q_1))\end{aligned}$$

Ecuaciones de movimiento

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = \tau_i - f_f(q_i)$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \dot{q}_1} = & (m_1lc_1^2 + m_2l_1^2)\dot{q}_1 + m_2lc_2^2\dot{q}_1 + m_2lc_2^2\dot{q}_2 + 2m_2l_1lc_2\cos(q_2)\dot{q}_1 + m_2l_1lc_2\cos(q_2)\dot{q}_2 \\ & + I_1\dot{q}_1 + I_2[\dot{q}_1 + \dot{q}_2]\end{aligned}$$

$$\begin{aligned}\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_1} = & (m_1lc_1^2 + m_2l_1^2 + m_2lc_2^2 + 2m_2l_1lc_2\cos(q_2))\ddot{q}_1 + [m_2lc_2^2 + m_2l_1l_2^2\cos(q_2)]\ddot{q}_2 \\ & - 2m_2l_1lc_2\sin(q_2)q_1\dot{q}_2 - m_2l_1lc_2\sin(q_2)\dot{q}_2^2 + I_1\ddot{q}_1 + I_2[\ddot{q}_1 + \ddot{q}_2]\end{aligned}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \dot{q}_2} = & m_2lc_2^2\dot{q}_1 + m_2lc_2^2\dot{q}_2 + m_2l_1lc_2\cos(q_2)\dot{q}_1 + m_2lc_2^2\dot{q}_2 + 2m_2l_1lc_2\cos(q_2)\dot{q}_1 \\ & + m_2l_1lc_2\cos(q_2)\dot{q}_2 + I_1\dot{q}_1 + I_2[\dot{q}_1 + \dot{q}_2]\end{aligned}$$

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_2} = m_2lc_2^2\ddot{q}_1 + m_2lc_2^2\ddot{q}_2 + m_2l_1lc_2\cos(q_2)\ddot{q}_1 - 2m_2l_1lc_2\sin(q_2)q_1\dot{q}_2 + I_2[\ddot{q}_1 + \ddot{q}_2]$$

$$\begin{aligned}\tau_1 = & (m_1lc_1^2 + m_2l_1^2 + m_2lc_2^2 + 2m_2l_1lc_2\cos(q_2) + I_1 + I_2)\ddot{q}_1 + [m_2lc_2^2 + m_2l_1l_2^2\cos(q_2)]\ddot{q}_2 \\ & - 2m_2l_1lc_2\sin(q_2)q_1\dot{q}_2 - m_2l_1lc_2\sin(q_2)\dot{q}_2^2 + b_1\dot{q}_1 + [m_1lc_1 + m_2l_1]g\sin(q_1) \\ & + m_2glc_2\sin(q_1 + q_2)\end{aligned}$$

$$\begin{aligned}\tau_2 = & [m_2lc_2^2 + m_2l_1lc_2\cos(q_2) + I_2]\ddot{q}_1 + [m_2lc_2^2 + I_2]\ddot{q}_2 + m_2l_1lc_2\sin(q_2)q_1^2 + b_1\dot{q}_2 \\ & + m_2glc_2\sin(q_1 + q_2)\end{aligned}$$

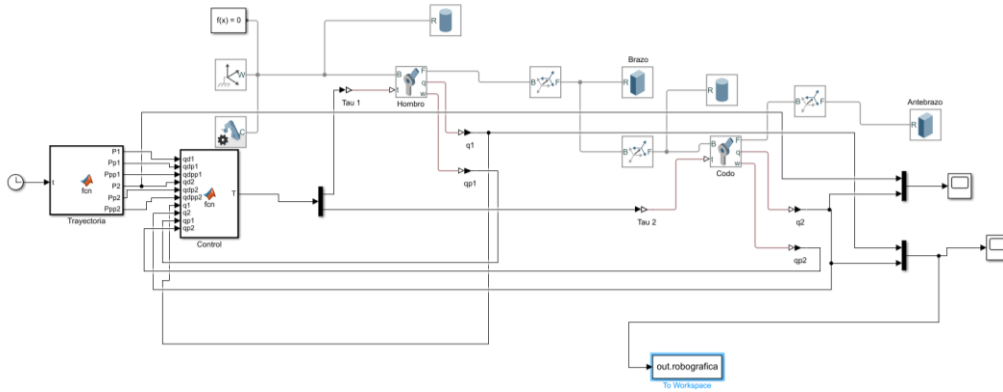
Ecuación general del robot

$$\tau = M(q)\ddot{q} + C(q + \dot{q})\dot{q} + B\dot{q} + G(q)$$

Diseño y construcción

El modelo creado con Simscape Multibody es el siguiente, estamos utilizando los mismos parámetros, pero esta vez es un doble péndulo robot.





Ambos modelos utilizan los mismos parámetros para poder validar su funcionamiento.

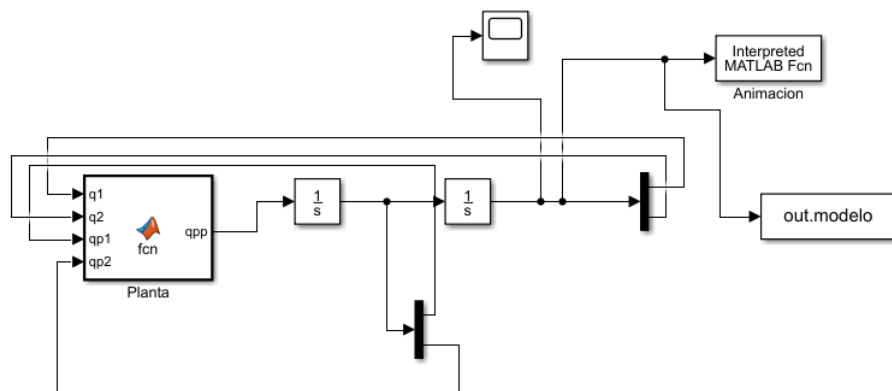
El código de trayectoria y control se encuentran en los siguientes links respectivamente:

<https://github.com/xDzohlX/Controldesistemasroboticos/blob/master/Practica5/Trayectoria.m>

<https://github.com/xDzohlX/Controldesistemasroboticos/new/master/Practica5>

### Validación

Ya con la ecuación general del robot se puede simular en planta con simulink utilizando los siguientes bloques



El código de la planta se encuentra en el siguiente git:

<https://github.com/xDzohlX/Controldesistemasroboticos/blob/master/Practica5/Planta.m>

Ambos modelos utilizan los mismos parámetros para poder validar su funcionamiento.

### Esquema de control

Los controladores que se probaron fueron el proporcional con retroalimentación compensado por gravedad con:

$$\tau = K_p \tilde{q} - K_d \dot{\tilde{q}} + G(q)$$

Y con el control de par calculado

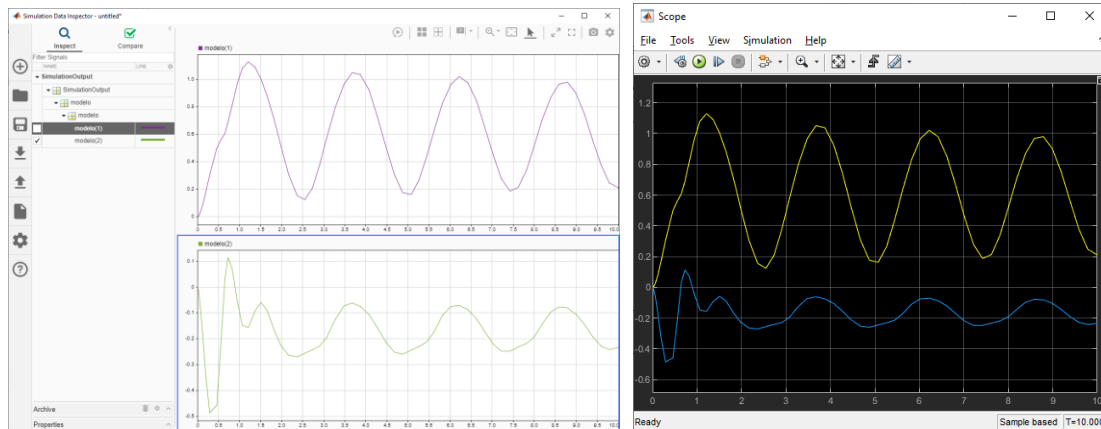
Como se ha dicho este control toma en cuenta la dinamica del sistema y esta dado por:

$$\tau = M(q)[\ddot{q}_d + K_d\dot{\tilde{q}} + K_p\tilde{q}] + C(q + \tilde{q})\dot{q} + B\dot{q} + G(q)$$

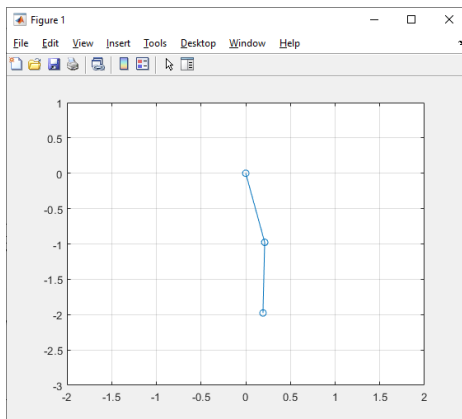
## Resultados

Se obtuvieron los datos de Matlab para hacer la validación igualmente con un scope.

Las gráficas muestran los valores de posición de ambos péndulos con una entrada constante

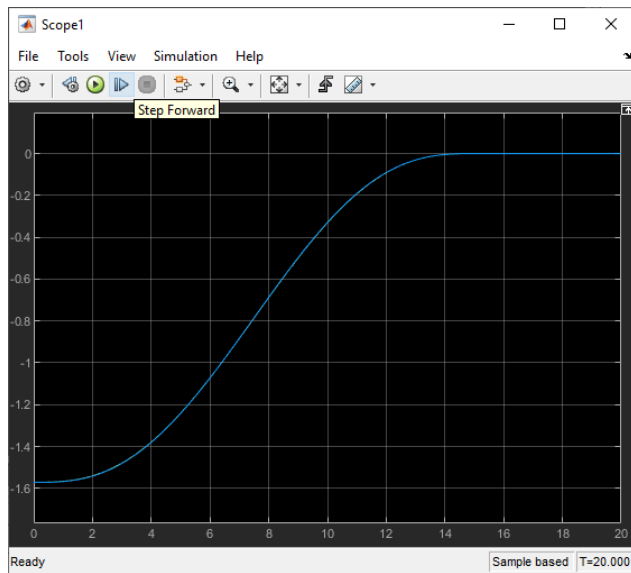


## Simulación de la planta

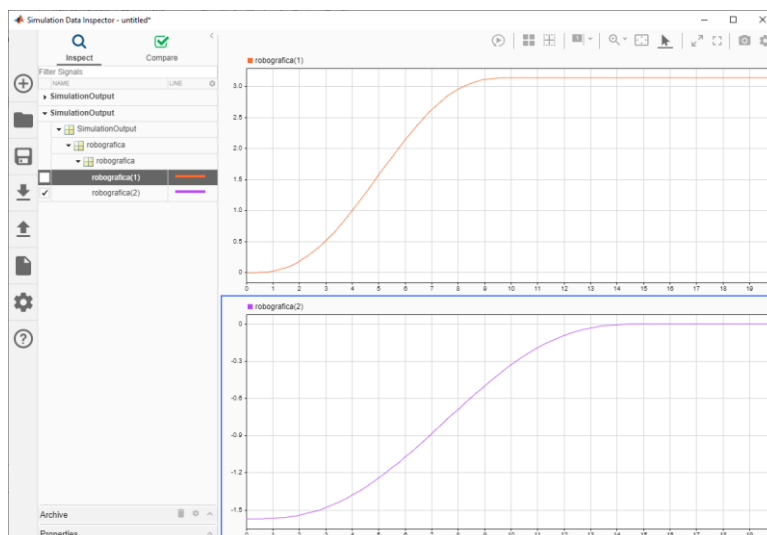


## Seguimiento de trayectoria

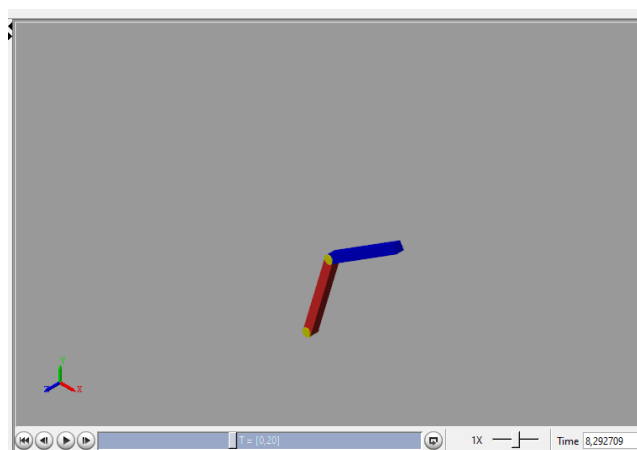
Por parte del modelo



Grafica de posiciones del modelo



Posicionamiento del brazo



Conclusiones:

La dinámica del brazo con dos grados de libertad es mucho mas complicada, la dinámica de este hace que cuando se mueve un brazo se afecta al otro pero gracias al los controladores este fácilmente puede seguir trayectorias sin mayor problema y estos controladores basados en el modelo y la dinámica el robot no tienen problemas haciendo esto y las ganancias no son muy grandes.

#### Practica 6: Instalación del complemento Simscape Multibody Link

Instalar el complemento de Simscape Multibody Link en SolidWorks para poder exportar ensamblajes a Matlab Simulink.

Objetivo:

Instalar el complemento de Simscape Multibody Link en SolidWorks

#### Practica 7: Exportar un ensamble a Matlab

Creación de un doble péndulo robot en SolidWorks y exportarlo al ambiente de Matlab Simulink y validarlo con las ecuaciones dinámicas anteriores.

Objetivos:

Exportar un ensamblaje de un doble péndulo robot a Matlab Simulink con la herramienta de Simscape Multibody Link instalada anteriormente.

Validar el ensamblaje y los parámetros obtenidos de SolidWorks con las ecuaciones dinámicas.

### Introducción Practica 8: Cuarto seguimiento de trayectoria, plano cartesiano

Creación de dos controladores de seguimiento de trayectoria para un doble péndulo robot creado en SolidWorks e importarlo a Matlab Simulink, validarlo con sus respectivas ecuaciones dinámicas. Son tres trayectorias definidas por sus ecuaciones paramétricas.

Objetivos:

Crear un doble péndulo robot con SolidWorks y exportarlo a Matlab Simulink y validarlo con sus ecuaciones dinámicas.

Generar trayectorias para el robot con ecuaciones paramétricas con tres geometrías: un círculo, una lemniscata y una flor de cuatro pétalos.

Validar en control de seguimiento de trayectoria con sus ecuaciones dinámicas y después en el modelo importado.

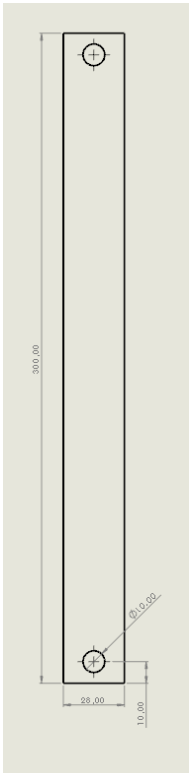
#### Metodología

##### Modelo dinámico

El modelo dinámico es igual al de la práctica cinco.

## Diseño y construcción

Para el diseño del doble péndulo robot se diseño primero en solidworks con los eslabones de las siguientes medidas e iguales



Las medias del hombro son irrelevantes ya que no se mueve y el material utilizado fue aluminio 7075 t6 solo para variar.

Este después se exporó a simulink quedando de la siguiente forma



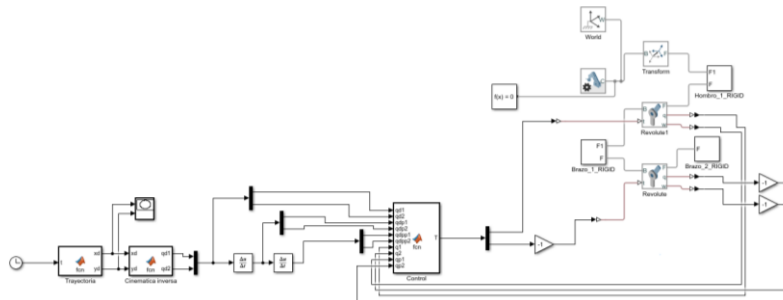
Y los siguientes parámetros

```
m1=0.06948782;  
l1=0.3-20/1e3;  
lc1=150/1e3-10/1e3;  
I1=0.00050975;  
b1=0.0017;  
m2= 0.06948782;  
l2=0.3-20/1e3;  
lc2=150/1e3-10/1e3;
```

$I2=0.00050975;$

$b2=0.0013;$

Ya con el modelo importado se le agrego el controlador y las trayectorias quedando los bloques de la siguiente manera.



Se debe tener cuidado con la orientación del ensamblaje porque solidworks y Matlab si lo tienen en cuenta.

También en este diseño es requerida la cinemática inversa al estar utilizando ecuaciones paramétricas.

El código del controlador, la cinemática inversa y la trayectoria se encuentran en el git respectivo:

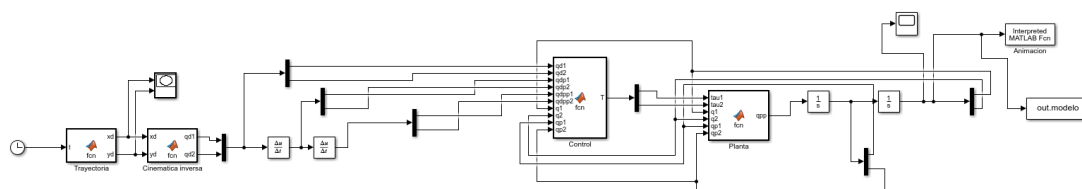
<https://github.com/xDzohlx/Controldesistemasroboticos/blob/master/Practica8/Control.m>

<https://github.com/xDzohlx/Controldesistemasroboticos/blob/master/Practica8/Cinematicainversa.m>

<https://github.com/xDzohlx/Controldesistemasroboticos/blob/master/Practica8/Trajectoryparametrica.m>

## Validación

Para la validación se utilizo la planta de la practica cinco con su simulación cambiando los parámetros por los actuales y generando la trayectoria requerida. Quedando los bloques de la siguiente manera.

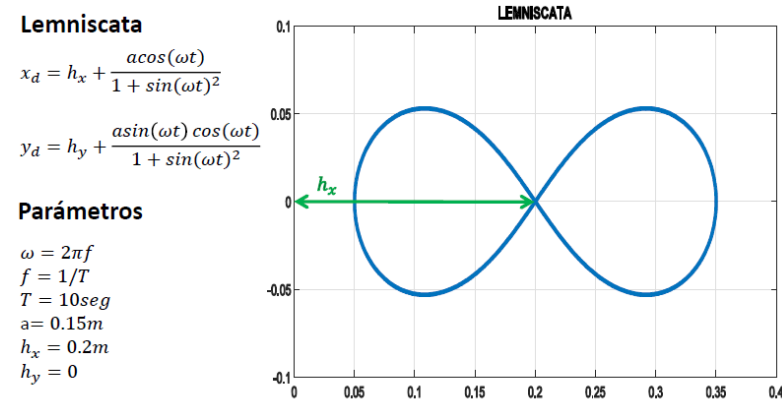
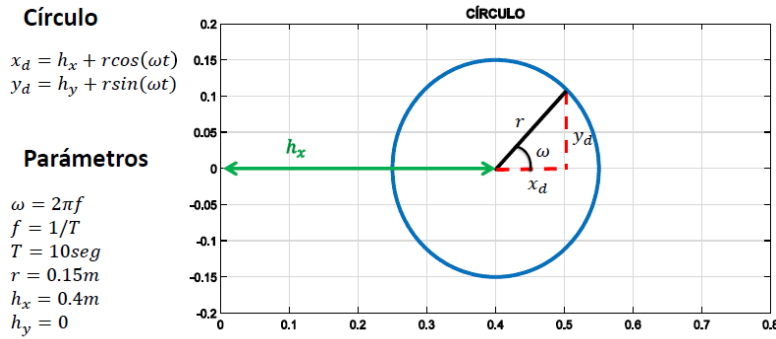


Los códigos en la trayectoria y controlador pero el código de la planta se encuentra el en siguiente enlace.

<https://github.com/xDzohlx/Controldesistemasroboticos/blob/master/Practica8/Planta.m>

## Diseño de la trayectoria

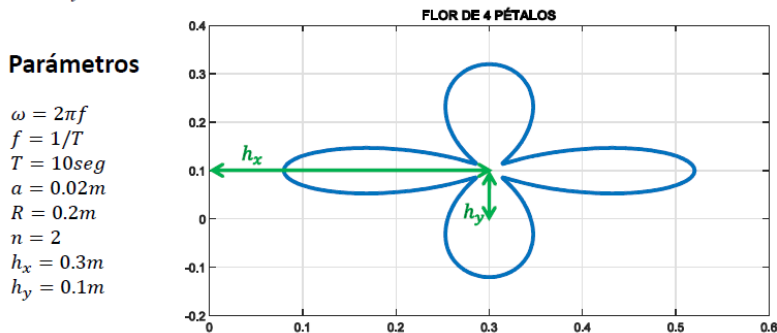
Se utilizaron ecuaciones paramétricas para que la trayectoria de la figura este variando con respecto al tiempo para que se guie la trayectoria directamente de estas y la ayuda de la cinemática inversa. Se generaron tres trayectorias las cuales son:



## Flor de 4 Pétalos

$$x_d = h_x + (a + R \cos(n\omega t)^2) \cos(\omega t)$$

$$y_d = h_y + (a + R \cos(n\omega t)^2) \sin(\omega t)$$



Estas se generaron a la derecha del plano cartesiano y no indeterminar la cinemática inversa.

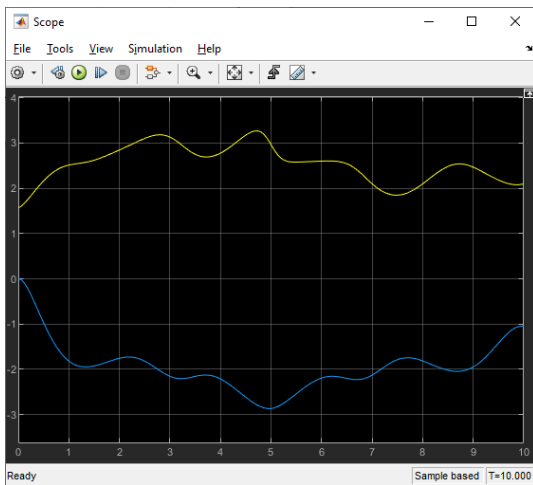
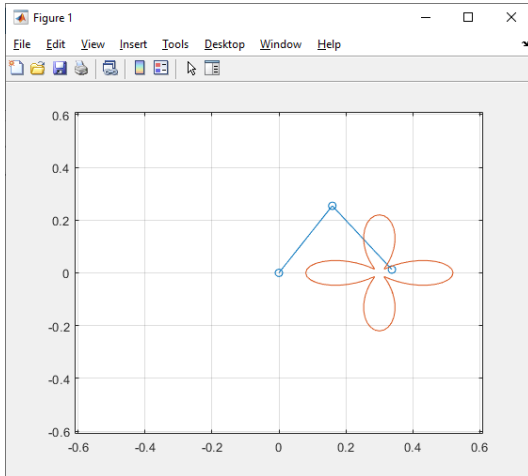
## Esquema de control

Para esta practica utilizamos el control de par calculado:

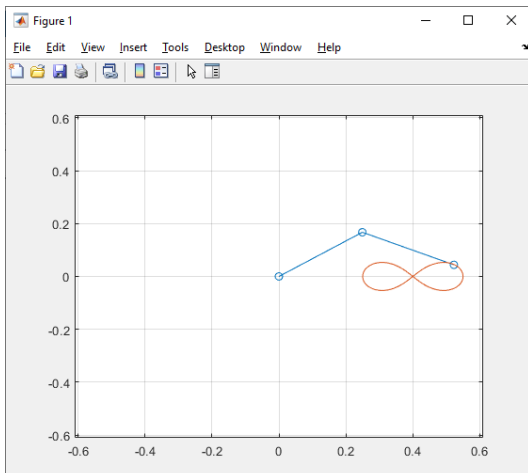
$$\tau = M(q)[\ddot{q}_d + K_d \tilde{q} + K_p \dot{\tilde{q}}] + C(q + \tilde{q})\dot{q} + B\dot{q} + G(q)$$

## Resultados

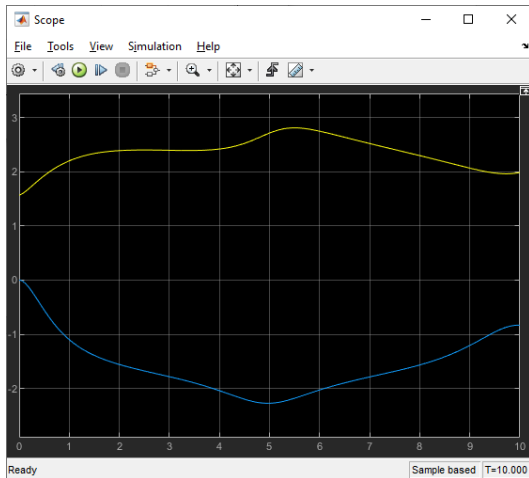
Trayectoria de la flor de cuatro pétalos.



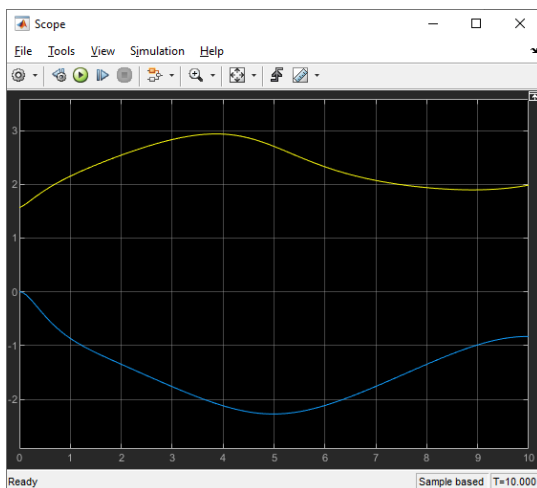
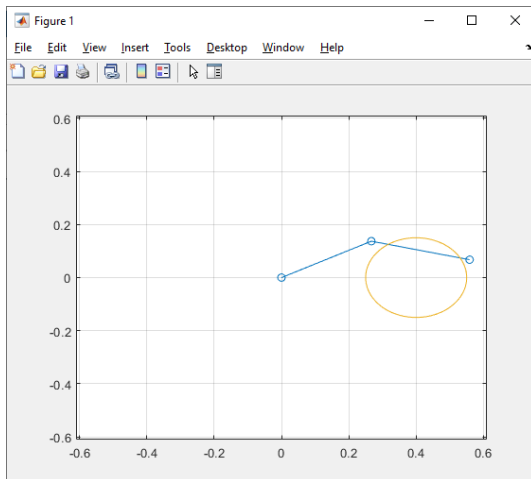
Trayectoria de la lemniscata







## Trayectoria del círculo



## Conclusiones

Esto fue la culminación de todas las prácticas utilizando uno de los mejores controladores y utilizando la cinemática inversa en conjunto con ecuaciones paramétricas para poder generar

trayectorias mas complicadas que solo funciones seno o ecuaciones de polinomios. El control de par calculado parece ser de los mejores controladores aun con ganancias bajas el error es muy bajo y siempre se cumple la trayectoria aun en cambios de posición y velocidad radicales.

## Bibliografía

- [1] V. S. a. A. L. R. Kelly, Control of Robot Manipulators in Joint Space, Londres: Springer, 2005.
- [2] F. d. robotica, Antonio barrientos, Luis Felipe, Calos Balaguer, Rafel, Madrid: Concepcion Fernandez Madird, 1997.
- [3] J. J. Craig, Robótica, Ciudad de Mexico: Pearson, 2006.
- [4] F. R. Cortes, Robotica Control de Robots Manipuladores, Ciudad de Mexico: Alfaomega, 2011.