Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ingeniería Eléctrica
IEE2613 – Control Automático

Proyecto Viga Cilindro

Entrega 1

Ignacio Andrés Pérez Rojas

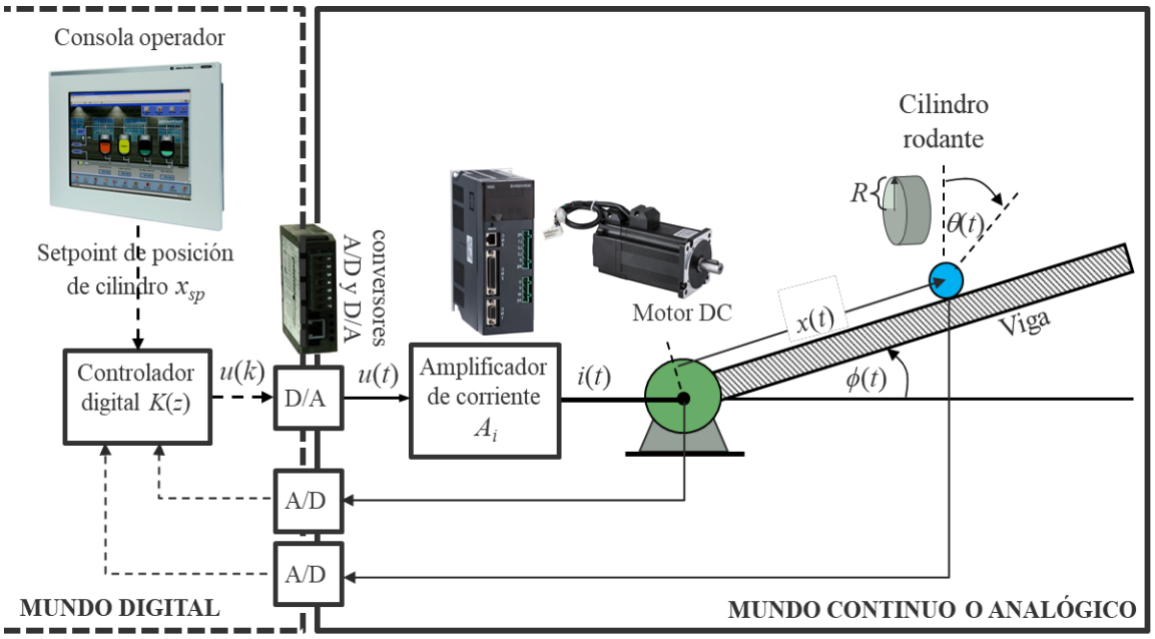

Figura 1: Esquema General del proyecto

Índice

1. Introducción

A lo largo de este informe, se desarrolla un controlador mediante el programa Matlab usando el sistema viga-cilindro entregado por el ayudante del curso Control Automático. El sistema entregado es no lineal, por lo que cada uno de los controladores se parten sintonizando en base a prueba y error por separado para finalmente juntarlos. Obteniendo finalmente un error permanente igual a cero, se hace un análisis, discusión de supuestos y resultados y conclusiones.

2. Implementación del controlador

2.1. Tipo de controlador escogido y característicasticas

El controlador escogido debe de cumplir con las cafracterísticas de poder controlar un sistema no lineal y discreto, entonces se opta por usar un sistema de PID discreto visto las diapositivas del curso. Además el controlador cumple con el requisito de incluir la retención de orden como ocurre en la realidad solicitado.

Un controlador PID se peude vere de la siguiente forma:

$$\frac{U(z)}{E(z))} = k(z) = kp + ki' * \frac{z}{z-1} + kd' * \frac{z-1}{z} \tag{1}$$

El segundo término corresponde a la acción integra y el tercer término a la derivativa. Este tipo de control se aplica de forma igual a ambos controladores explicados en el informe.

$$u_k = u_{k-1} + (kp + ki' + kd') * e_k - (kp + 2 * kd') * e_{k-1} + kd' * e_{k-2} \tag{2}$$

Donde tenemos que kd' = kd*delta y ki'=ki/delta
Con Delta el la diferencia de tiempo del flujo del programa

2.2. Comienzo de los valores de controlador PID - Sólamente controlando la posición

Para el controlador, se escoge el PID debido a que es el visto con mayor profunidad a lo largo dle curso, por lo que el uso de este en base a lo aprendido se considera la mejor herramienta para poder enfrentarse a este problema de modelación

Para realizar este controlador, se siguió una serie de pasos. Inicialmente se tiene la duda sobre si hacer un controlador en cascada o un controlador competitivo. Para poder tomar esta decisión se llega a la conclusión de que lo mejor es inicialmente hacer un controlador PID para la posición. Inicialmente en el archivo V̈B_cont_pos.m:. Comenzando con el primer controlador sólom para la posición, tenemos los siguientes resultados.
Nota 1: el tipo de fallo se refiere a si el cilindro se cae del lado del motor o el otro lado **Nota 2:** el tipo de fallo se refiere a si el cilindro se cae del lado del motor o el otro lado, no se anota el tiempo debido a que no se espera que el controlador sea capaz de mantener mucho tiempo el valor del cilindro dentro de la viga por mucho tiempo

Kp	Ki	Kd	Tiempo (seg)	Tipo de fallo	referencia	Inicial (x,xp,Ω, Ωp)
10000	0	0	0,586	Motor	0,2	[0.2 0.0 0.0 0.0]
5000	0	0	0,588	Motor	0,2	[0.2 0.0 0.0 0.0]
500	0	0	0,572	Motor	0,2	[0.2 0.0 0.0 0.0]
100	0	0	0,446	Extremo	0,2	[0.2 0.0 0.0 0.0]
200	0	0	0,514	Extremo	0,2	[0.2 0.0 0.0 0.0]
300	0	0	0,748	Extremo	0,2	[0.2 0.0 0.0 0.0]
350	0	0	1,36	Extremo	0,2	[0.2 0.0 0.0 0.0]
400	0	0	0,664	Motor	0,2	[0.2 0.0 0.0 0.0]
375	0	0	1,274	Motor	0,2	[0.2 0.0 0.0 0.0]
350	0	0	0.36	Motor	0,3	[0.2 0.001 15.0º 0.0º]
350	0	0	0.2	Motor	0,3	[0.4 0.001 15.0º 0.0º]
350	0	0	0.58	Motor	0,3	[0.4 0.001 15.0º 0.0º]

Para los valores de kp escogidos, tenemos de que un kp igual a 350 fue el que más duró para un sistema de referencia específica, por lo que se cree de que es un buen valor. Al trabajar, se llega a la conclusión de que un kp aceptable, oscila entre los 150 y 450. Para poder incluir mejores resultados, se parte por agregar un el controlador integrador.

Kp	Ki	Kd	Tiempo (seg)	Tipo de fallo	referencia	Inicial (x,xp,Ω, Ωp)
450	150	1	1,36	Extremo	0,25	[0.35 0.0 0.0 0.0]
450	150	15	0,184	Extremo	0,25	[0.35 0.0 0.0 0.0]
400	100	0	1,234	motor	0,25	[0.35 0.0 0.0 0.0]
400	100	1	1,232	motor	0,25	[0.35 0.0 0.0 0.0]
400	100	2,5	0,25	extremo	0,25	[0.35 0.0 0.0 0.0]
400	100	1,5	0,44	extremo	0,25	[0.35 0.0 0.0 0.0]
400	100	1,2	1.23	extremo	0,25	[0.35 0.0 0.0 0.0]

Para estos valores de variables controladas (los últimos de la tabla), tenemos que se controla más tiempo, por lo que para estos valores se prueban distintas referencias.

Kp	Ki	Kd	Tiempo (seg)	Tipo de fallo	referencia	Inicial (x,xp,Ω, Ωp)
400	100	1,2	0.634	motor	0,1	[0.25 0.0 0.0 0.0]
400	100	1,2	0.834	motor	0,1	[0.4 0.0 0.0 0.0]
400	100	1,2	0.386	Extremo	0,4	[0.1 0.0 0.0 0.0]
400	100	1,2	0.386	Extremo	0,3	[0.2 0.0 0.0 0.0]
400	100	1,2	0.328	Extremo	0,25	[0.25 0.0 0.0 0.0]

En base a estos resultados, se llega a la conclusión de que por ejemplo, cuando parte teniendo una referencia igual o parecida a la posición actual, no se le da fuerza suficiente al motor para que pueda mantenerse estable. Esto se debe a que los valores de la integral empiezan a hacer efecto en el tiempo y el proporcional no hay mucho que pueda hacer. Es por esta razón que se debe de hacer un controlador que ayude a mantener estable la barra, debido a que con los valores dado la barra toma una velocidad angular muy grande. Además la idea es mantener limitada la velocidad que pueda tomar el cilindro. Es por esto que se propone hacer un controlador para el ángulo.
Se muestra los gráficos para kp=400, ki=100, kd = 1

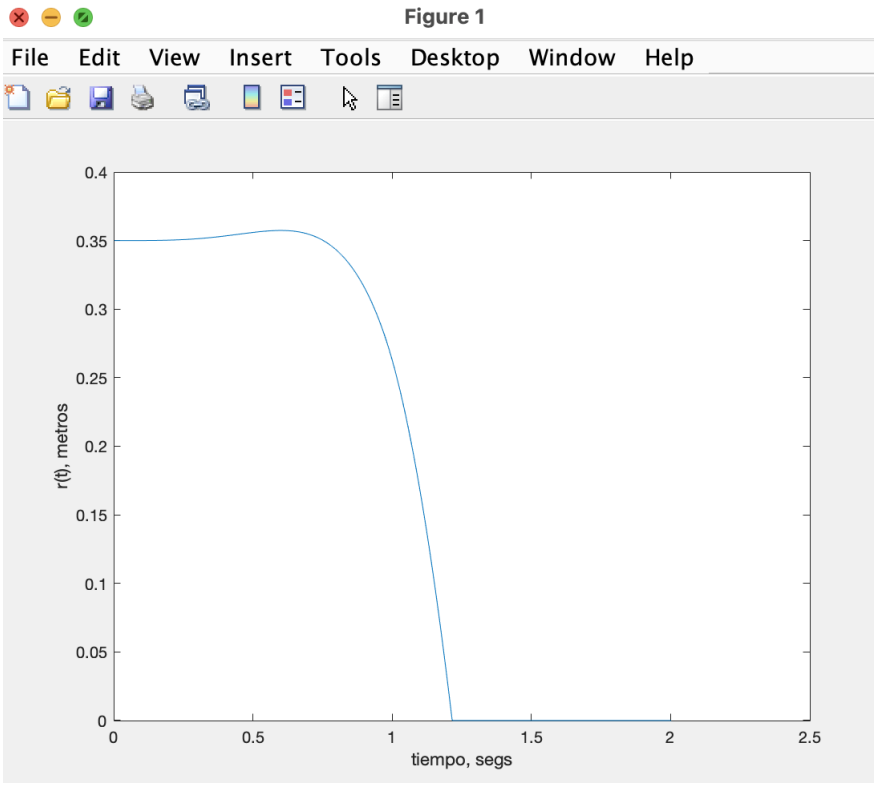

Figura 2: Posicion en el tiempo del cilindro

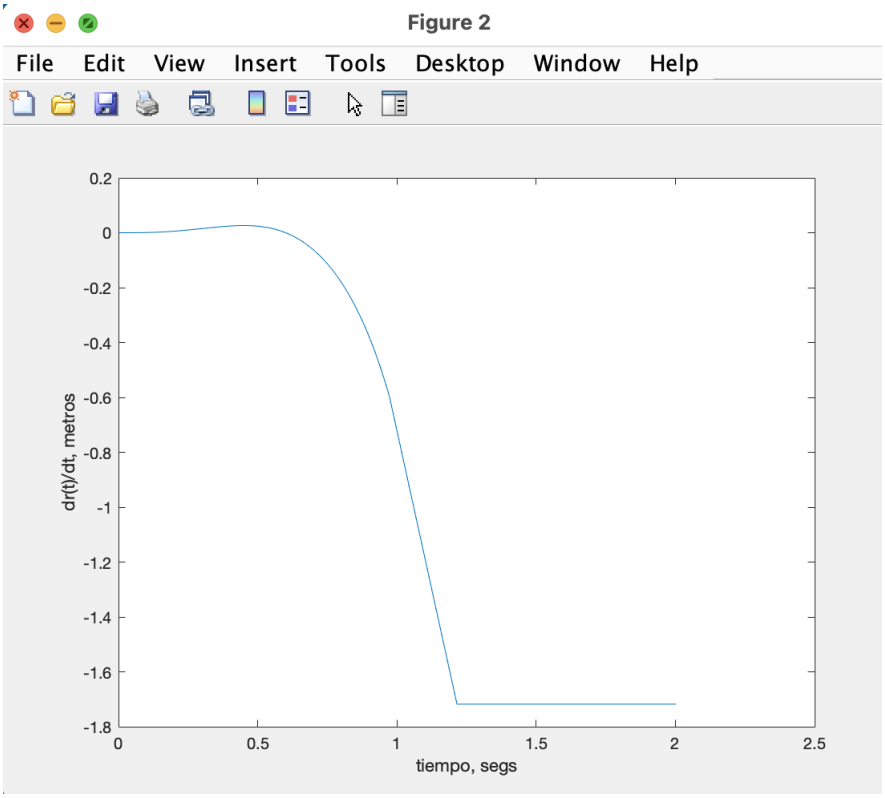

Figura 3: Velocidad en el tiempo del cilindro

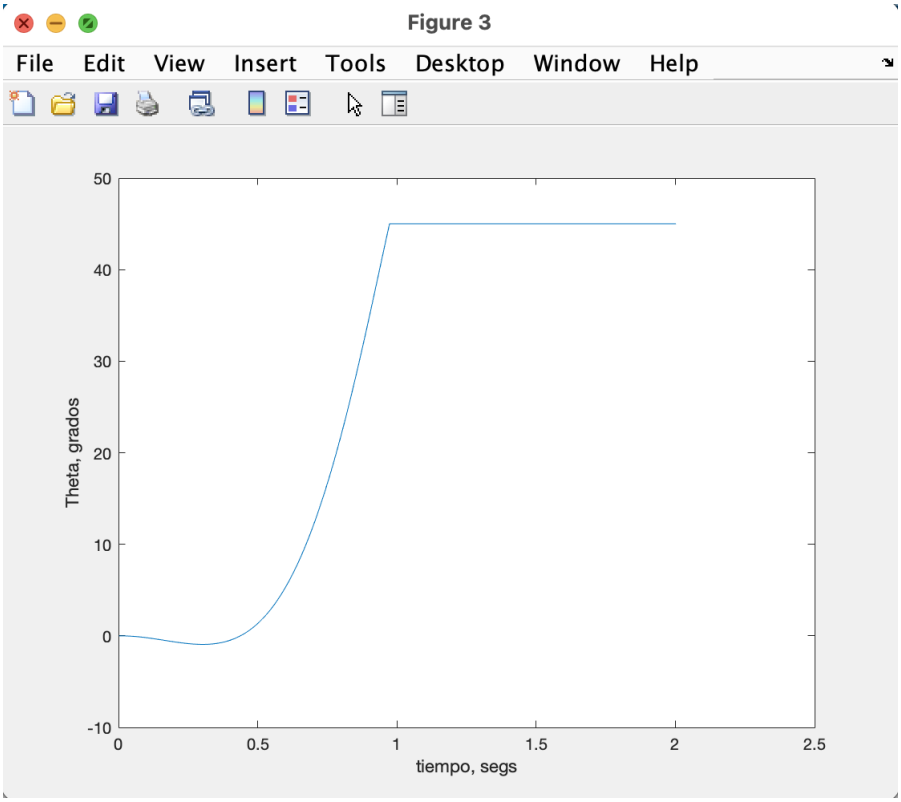

Figura 4: Ángulo

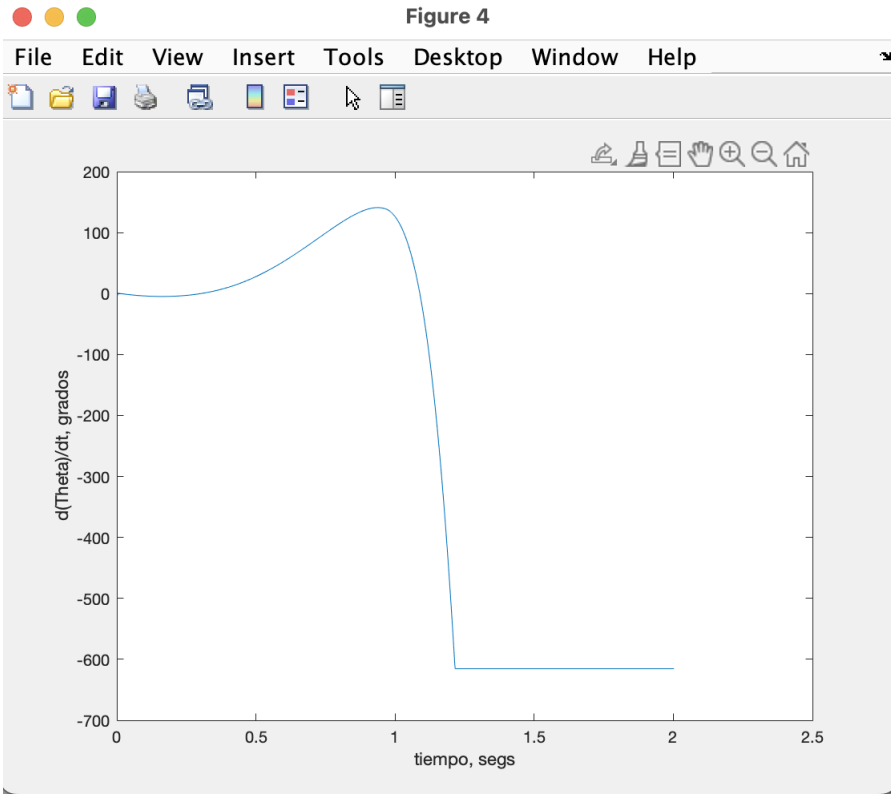

Figura 5: Derivada del angulo

2.3. Controlador para el ángulo

En base a los resultados de los gráficos para los mejores valores encontrados, falta el poder mantener el angulo de la barra en 0^o, es por esto que se decide implementar un controlador para el ángulo usando el mismo método anterior (sin importar la posición del cilindro)

Kp	Ki	Kd	Tiempo	Tipo de fallo	referencia (angular)	Inicial (x,xp,Ω, Ωp)
100	0	0	0,34	Extremo	0,0	[0.25 0.0 0.0 0.0]
100000	0	0	1,654	Extremo	0.0	[0.25 0.001 0^o 0.0^o]
500	0	0	0.786	Extremo	0,0	[0.25 0.000 0^o 0.0^o]
1000	0	0	0.956	Extremo	0,0	[0.25 0.000 0^o 0.0^o]
5000	0	0	1.22	Extremo	0,0	[0.25 0.000 0^o 0.0^o]

En la segunda fila, se nota de que el kp es excesivamente alto, sin embargo, controlado el ángulo la pelota se mantuvo más tiemop en posición que cualquier otro intento con el primer controlador. Que al segundo intento ya mejore el tiempo es un buen indicio de que esta vez si se está yendo por el buen camino. Se agrega al kp de 5000 el derivador para evitar que hayan cambios tan bruscos en la derivada. La referencia

Kp	Ki	Kd	Tiempo (seg)	Tipo de fallo	comentario
5000	0	100	1,774	Extremo	Demasiado alto el kd
5000	0	50	1,7	Extremo	Demasiado alto el kd (mejora)
5000	0	6	2,188	Extremo	Demasiado alto el kd (inicio = -1.0º)
5000	0	6	2,86	Extremo	Demasiado alto el kd (inicio = 0.0º)
5000	1000	7	2,932	Extremo	Demasiado alto el kd (inicio = 0.0º)

Para ilustrar los resultados obtenidos con un controlador anguloar de valores kp = 5000, ki = 1000, kd = 7

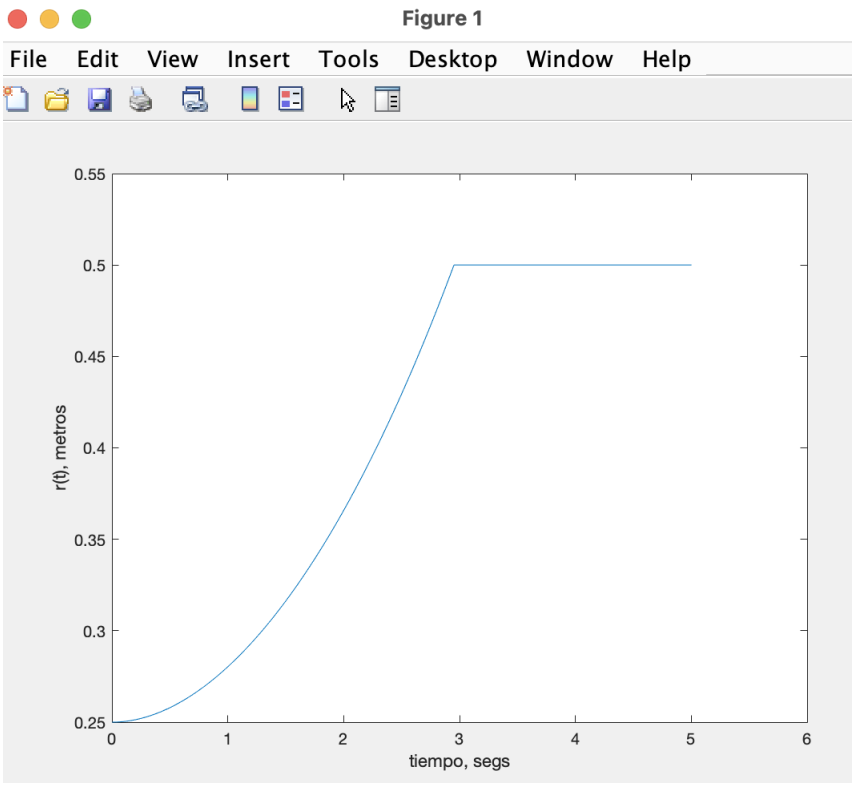

Figura 6: Posicion en el tiempo del cilindro

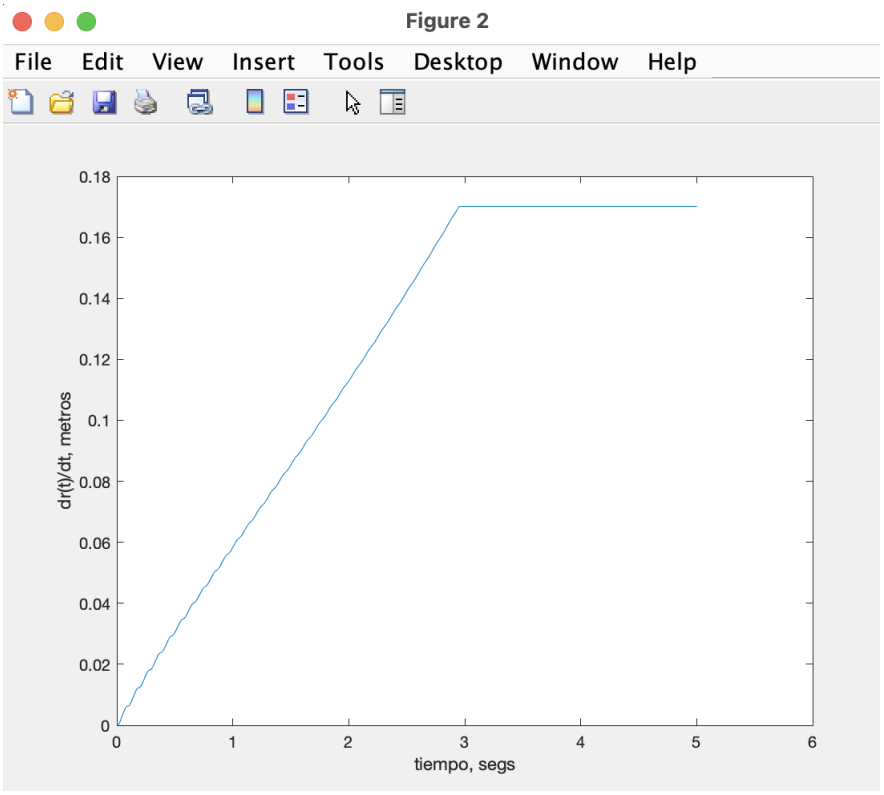

Figura 7: Velocidad en el tiempo del cilindro

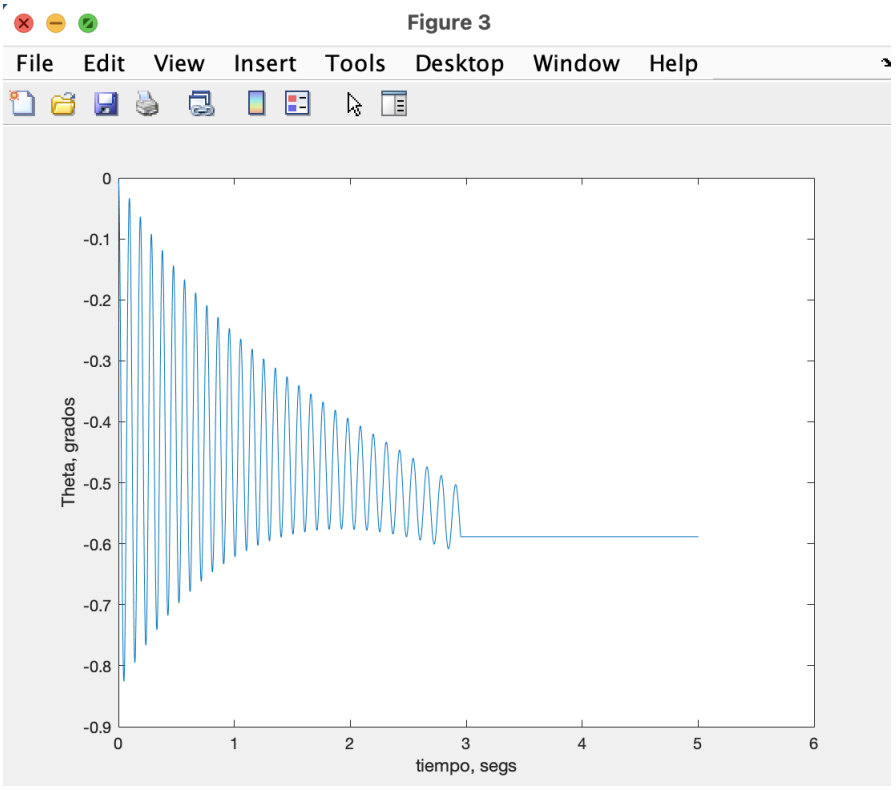

Figura 8: Ángulo

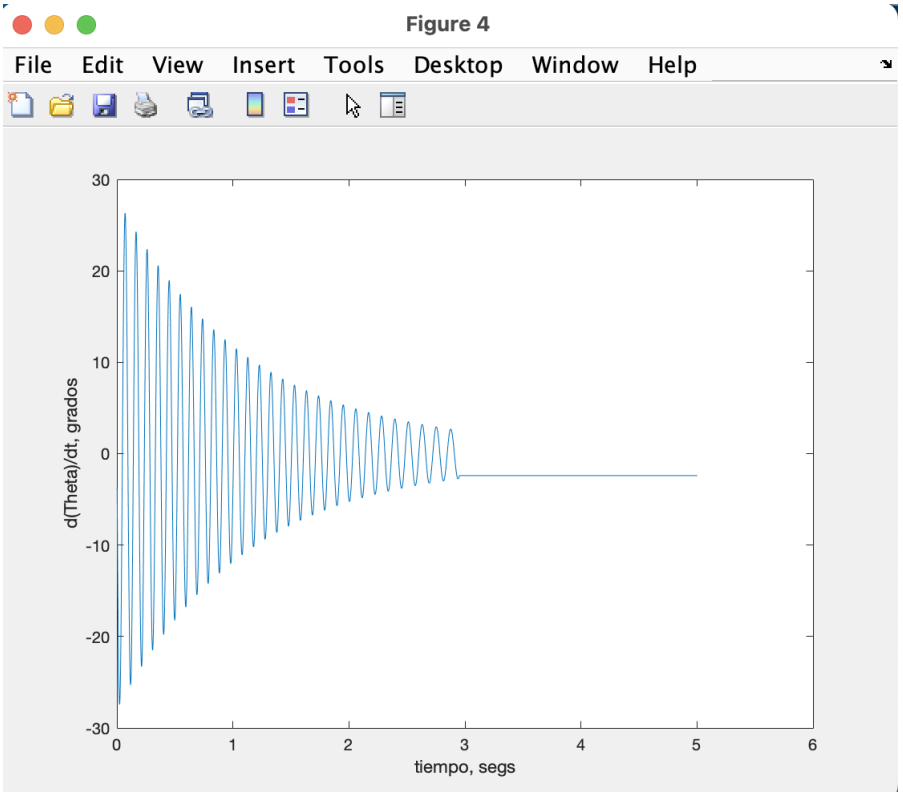

Figura 9: Derivada del angulo

Con los excelentes resultados del controlador, se tiene que falta el integrador para poder dejar el error con un valor igual a cero. para el ángulo.

Kp	Ki	Kd	Tiempo (seg)	Tipo de fallo	comentario
5000	10	7	2,944	Extremo	Demasiado alto el kd (inicio = 0.0^{o})
5000	100	7	2,956	Extremo	Demasiado alto el kd (inicio = 0.0^{o})
5000	200	7	2,99	Extremo	Demasiado alto el kd (inicio = 0.0^{o})
5000	500	7	4,95	Extremo	Demasiado alto el kd (inicio = 1.0^{o})

Al probar con estos valores de kp, ki y kd, se llega a la conclusión de que lo mejor es el poder hacer kp, ki y kd variables según la referencia para poder obtener mejores resultados. Esto es porque mientras más alejado del centro está el cilindro, mayor torque es el que necesitará. Es por esto que se decide el poder obtener los valores de kp, ki y kd para diferentes puntos de referencia. Se asume que el toque es lineal, por lo que se calcula para la referencia 0,1 y 0,4.

Kp	Ki	Kd	Tiempo (seg)	Tipo de fallo	comentario
5000	10	7	2,944	Extremo	Demasiado alto el kd (inicio = 0.0^{o})
5000	100	7	2,956	Extremo	Demasiado alto el kd (inicio = 0.0^{o})
5000	1000	7	6,01	Extremo	Bien

Con estos valores, se llega a la conclusión de que el controlador mantiene el cilindro dentro de la simulación por unos 6 segundos. El mejor resultado hasta ahora, por lo que se decide juntar ambos controladores. La idea es que el segundo intente mantener el error angular en un valor igual a cero y que primer controlador ayude a dar la fuerza necesaria para que la viga llegue al valor de la bola necesaria. Se partirá por usar los valores angulares obtenidos fijos e ir cambiando los valores de

los controladores asociados a el de posición.

2.4. Controlador final - juntar controlador de posición y controlador angular

Este controlador se usa en el archivo V̈B_final.m
Volviendo al sistema de anotar en tablas los resultados, tenemos de que para un kp = 5000, ki = 1000 y kd = 7 (para el controlador angular), y valores de referencia iguales a [0.25 0.0 0.0 0.0], para la posición, velocidad, angulo y velocidad angular, respectivamente. La referencia de posición está en 0,25:

Kp	Ki	Kd	Tiempo (seg)	Tipo de fallo	comentario
200	100	5	4,12 segundos	Extremo	$\text{pos}_k pbajo$
400	50	5	¡10 segundos	Extremo	Aumentar tpo simulación
400	50	5	No se cae segundos	Extremo	Overshoot alto; aumentar kd y $\text{pos}_k d$

En base a los resultados obtenidos previamente, se decide simular por 200 segundos con el fin de verificar si alcanza a llegar a la referencia. Se muestran los gráficos de los resultados.

Como se puede ver, hay un overshoot muy alto por que se decide afinar este detalle antes de continuar. Se parte por aumentar tanto el kd, como pos_kd dejando fijaos varios valores, estos son:

Kp : Ki : Kd	$\text{pos}_k p : pos_k d : pos_k i$	Tiempo	Fallo
4000:1000:30	3000:200:100	No se cae	Mucho overshoot
4000:1000:30	3000:200:1000	Buen trabajo	Tiempo establecimiento
4000:1000:30	3000:200:2000	Buen trabajo	Tiempo establecimiento
4000:1000:30	3000:200:3000	Se cae	No se mantiene en la barra
4000:1000:30	3000:400:3000	No se cae	Buen resultado
4000:1000:30	3000:300:3000	No se cae	Buen resultado 4000:1000:30
3000:250:3000	No se cae	Buen resultado	

A partir de la tabla anterior, se tienen excelentes resultados, con un tiempo de establecimiento muy rápido, por lo que con estos mismos valores se sigue intentando para distintas referencias. Al intentar con unas cuantas referencias, me percato de que el derivativo del controlador angular es demasiado bajo, causando de que en la mayoría de las referencias se caiga el cilindro, por lo tanto se decide subir solamente este valor (kd está asociado al controlador angular) hasta que se mantenga dentro de la viga controladora. Luego de hacer esto, un buen valor de kd encontrado fue igual a 200.

3. Resultados

Si bien es cierto que para el cálculo de los controladores se pusieron resultados intermedios, es esta sección van los resultados finales.

Los mejores valores encontrados para el controlador competidor de angular y de posición fueron:

$$kp = 6000 \tag{3}$$

$$ki = 1000 \tag{4}$$

$$kd = 200 \tag{5}$$

$$pos_{kp} = 3000 \tag{6}$$

$$pos_{kd} = 250 \tag{7}$$

$$pos_{ki} = 200 \tag{8}$$

$$pos_{inicio} = 0.4 \tag{9}$$

Con estos valores se decide poner a prueba el controlador bajo las siguientes referencias:

```
for t1=0:dt:tmax

    if t1<30, pos_ref = 0.09;
    elseif t1<60, pos_ref = 0.41;
    elseif t1<90, pos_ref = 0.12;
    elseif t1<100, pos_ref = 0.38;
    elseif t1<110, pos_ref = 0.08;
    elseif t1<120, pos_ref = 0.43;
    end
```

Figura 10: referencias para la posición del cilindro (MatLab)

Luego, bajo las condiciones mencionadas anteriormente, los valores de las distintas componendes son igual a:

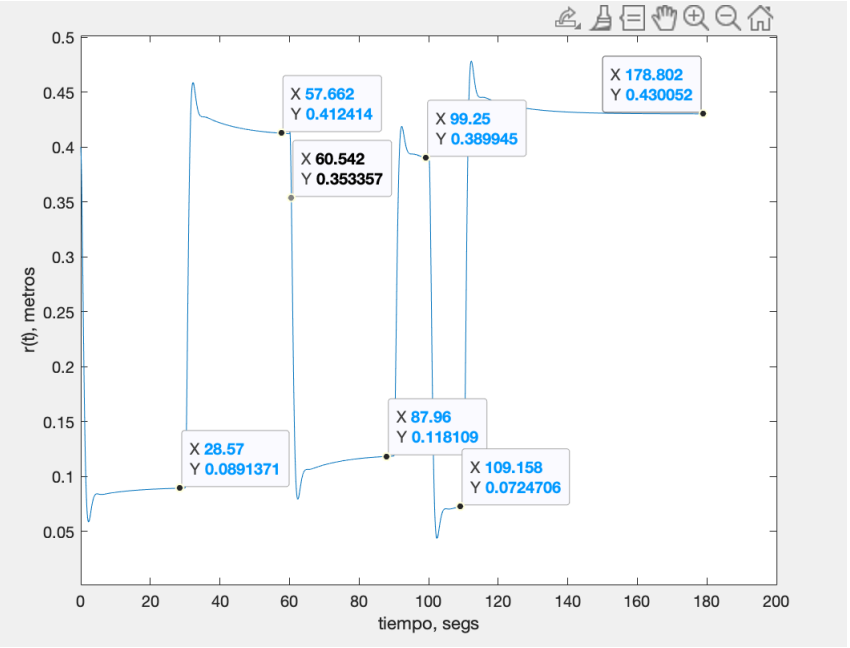

Figura 11: Posicion en el tiempo del cilindro

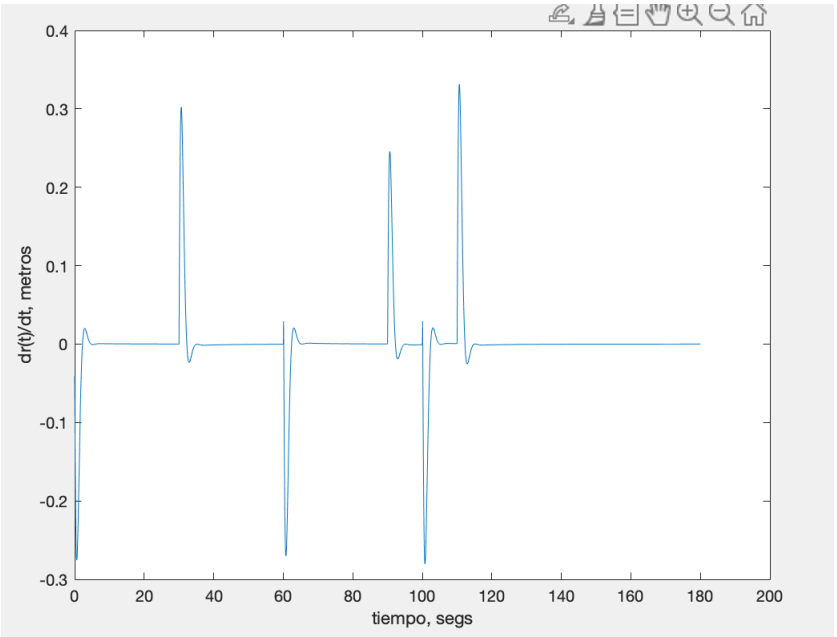

Figura 12: Velocidad en el tiempo del cilindro

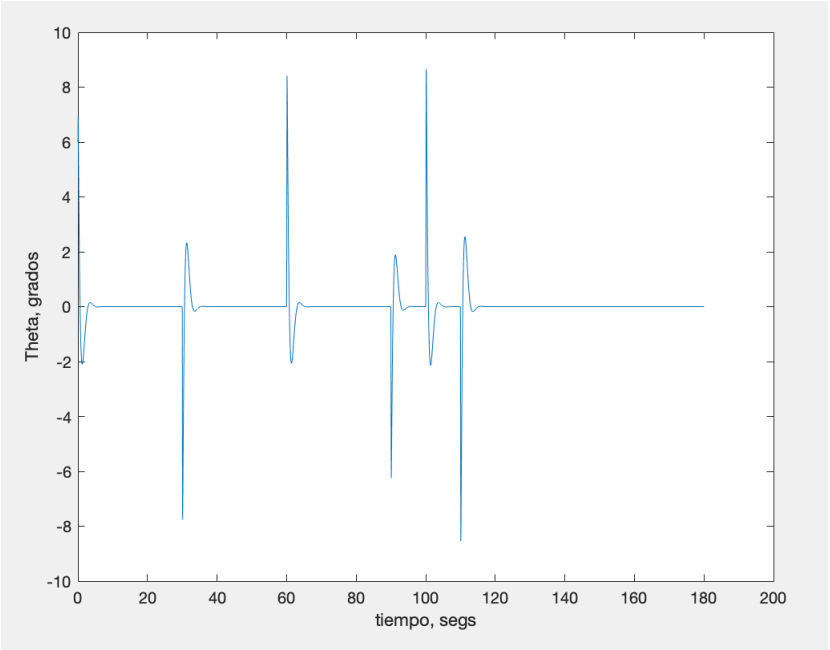

Figura 13: Ángulo

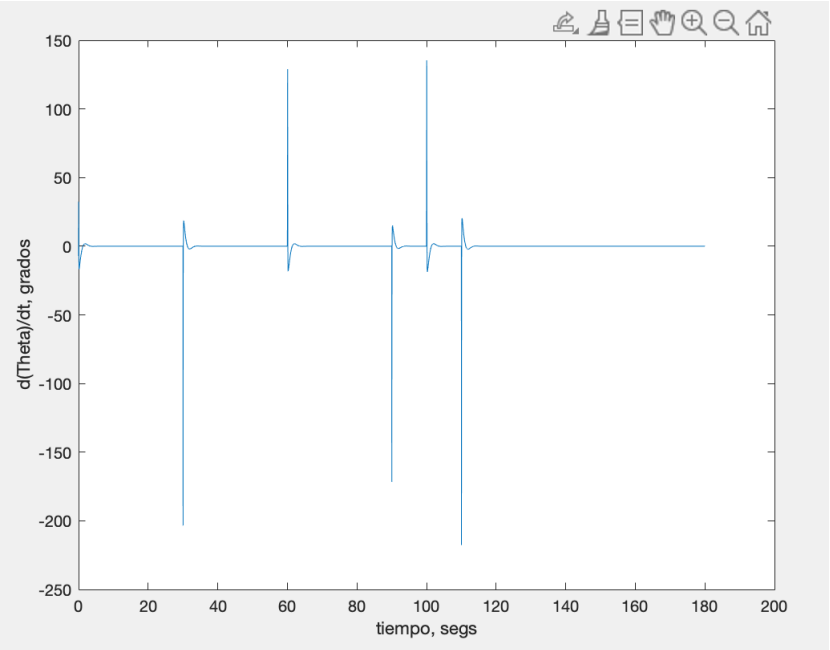

Figura 14: Derivada del angulo

4. Análisis y supuestos

Para poder llegar a los resultados, se usa un método de prueba y error para cada poder confeccionar dos controladores, la idea es que uno se encargue de mantener el ángulo en 0^{o} como referencia, y que el segundo sea capaz de llegar a una referencia. El PID para este tipo de sistema se volvió muy útil. La razón para utilizar este sistema de prueba y error fue que la mayoría de los métodos usados en clases son para sistemas lineales, en cambio, el sistema de este proyecto es no lineal.

Para la realización de este controlador, se hizo la suposición de que encontrar valores aceptable a los controladores por separado era la mejor opción. En base a los resultados obtenidos se llega a la conclusión de que si bien no se puede concluir que el método seguido para encontrar valores corresponde al mejor método (básicamente prueba y error), si se puede afirmar que se llega a buenos resultados con este método. No se realizan

Finalmente, otro supuesto que se hace es que ambos competidores necesitan competir por obtener buenos resultados, por lo que si por ejemplo, el controlador de posición quiere llegar muy rápido a la referencia, el angular lo detendría debido a que intenta mantener la barra en cero. A grandes rasgos al visualizar los gráficos finales se llega a la conlcusión de que esta suposición es cierta para algunos casos importantes.

5. Discusión de los resultados

El controlador final con los mejores resultados tiene una sobre oscilación grande, lo que implica dos cosas. Por un lado aumenta el tiempo de llegada al valor final. Se logra estabilizar y tiene un error permanente que tiende a cero en el tiempo, como se puede ver en la simulación de la figura 11, llega a un valor igual a 0.43052 luego de un poco más de un minuto. Esto es gracias a la acción integral tanto del controlador encargado de la referencia como del controlador encargado del ángulo.

En caso de querer que haya una sobre oscilación menor, sería suficiente reduciendo el pos_ki y el pos_kp y aumentando el pos_kd (dentro de un rango aceptable para que el cilindro no se caiga). Con esto sucederían dos cosas importantes que se podría notar en el controlador. La primera es que aumenta el tiempo de llegada a un error permanente menor a 2 % y la segunda es que disminuye considerablemente la sobreoscilación. Para determinar la mejor forma de controlar el cilindro, depende de lo que se pide. La función de costos penaliza tanto por tiempo de llegada a la referencia como por el error cuadrático medio, dado esto, se usa el controlador que a mí parecer fue el más adecuado.

Respecto a lo esperado inicialmente, el controlador funcionó mejor de lo esperado. En lo personal creí que lo mejor que podía lograr con un PID hecho a prueba y error, sería el llegar a la referencia después de más tiempo y con mayor número de sobreocilaciones.

Con las referencias intermedias también se logra buenos resultados, mostrando que el error permanente tiende a cero en el tiempo (a medida que pasa el tiempo disminuye tendiendo al valor de la referencia).

Si bien se obtienen buenos resultados, no es el mejor controlador. Otros tipos de controladores que podría funcionar sería uno en casacada, un controlador PD para la parte de la referencia, entre otros.

6. Conclusiones

Como conclusión se logra el poder usar un modelo de un sistema implementado en matlab, agregándole dos controladores, uno para el ángulo y otro para la distancia de referencia. Para simulaciones lo suficientemente largas, se llega a que no existe error permantente, por lo tanto se considera que el PID fue un éxito.

7. Anexos

1. Ibarra, Lizana y Rojas (Octubre, 2020), "Modelo Cohete". IEE2613 - Control Automático

2. Extraído de: "https://www.google.com/search?q=how+to+do+differential+equations+in+p
$how+to+do+differential+equations+in+python+3aqs = chrome..69i57j0i22i30.8257j0j7sour$
$chromeie = UTF-8kpvalbx = \_mQtrYtD1DOmJ1sQPhauz6A424$"