

AFR

Accionamientos con control en posición y velocidad

Álvaro Morales – 18240

TAREA – Opción I

- 1) Representación en Simulink del diagrama de bloques del servoeje (sin no-linealidades) con datos reales de elementos mecánicos.

A partir de los datos dados y las funciones de transferencia para modelizar los elementos a utilizar:

Resistencia entre terminales motor DC	0.573 ohm
Inductancia entre terminales motor DC	0.00009 mH
Constante par/intensidad	13 mN·m/A
Tiempo de arranque	6.82 ms
Momento de inercia del rotor	20 g·cm ²
Momento de inercia del acople	0.000016 kg·m ²
Rigidez torsional del acople	1.1 N·m/arcmin
Función de transferencia del husillo	$H_{BS}(s) = \frac{1}{0.001s^2 + 0.02s + 1}$
Paso del husillo	10 mm/rev
Resolución del encoder	1000 cuentas/rev

Modelado del Motor DC: (pasando los datos de la tabla a unidades del SI)

$$I = \frac{1}{Ls + R} \cdot V \rightarrow \boxed{\frac{1}{0.0000009s + 0.573}}$$

Motor DC (I/V)

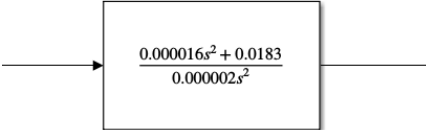
Modelado de la transformación del par: (Unidades SI)

$$M_M(s) = K_{I-M} \cdot I(s) \rightarrow \boxed{0.013}$$

Cte (M/I)

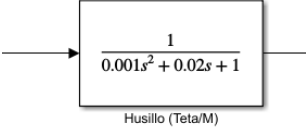
$$\theta_M(s) = \frac{1}{J_R s^2 + \mu_R s} \cdot M_M(s)$$

Juntando la segunda parte de dicho modelado al del acoplamiento elástico para que el grado del numerador no sea elevado y podamos realizar todo correctamente. (Unidades SI y pasando 1.1 N·m/arcmin a 0.018 N·m/arcsec)

$$M_c(s) = (J_c s^2 + k_c) \cdot \theta_M(s)$$


Transformación Mecánica (Teta/M) + Acoplamiento Elástico (M/Teta)

Para el modelado del husillo se nos daba directamente la función a usar.

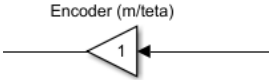
$$\theta_{BS}(s) = \frac{1}{J_{BS} s^2 + \mu_{BS} s + k_{BS}} \cdot M(s)$$


Husillo (Teta/M)

Para que la salida este en las correctas unidades de posición se utiliza el paso dato del paso del husillo. (SI)

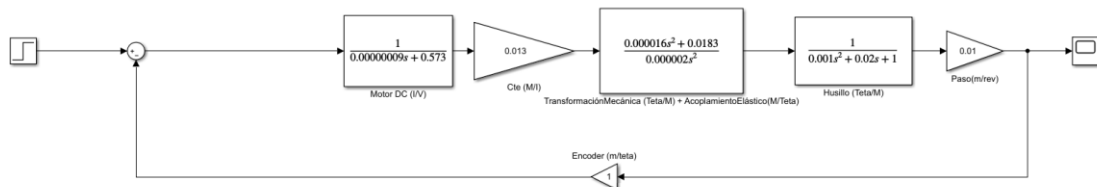


Por último, el modelado del sensor donde volveremos a usar el paso para la Klong-cuenta. (SI)

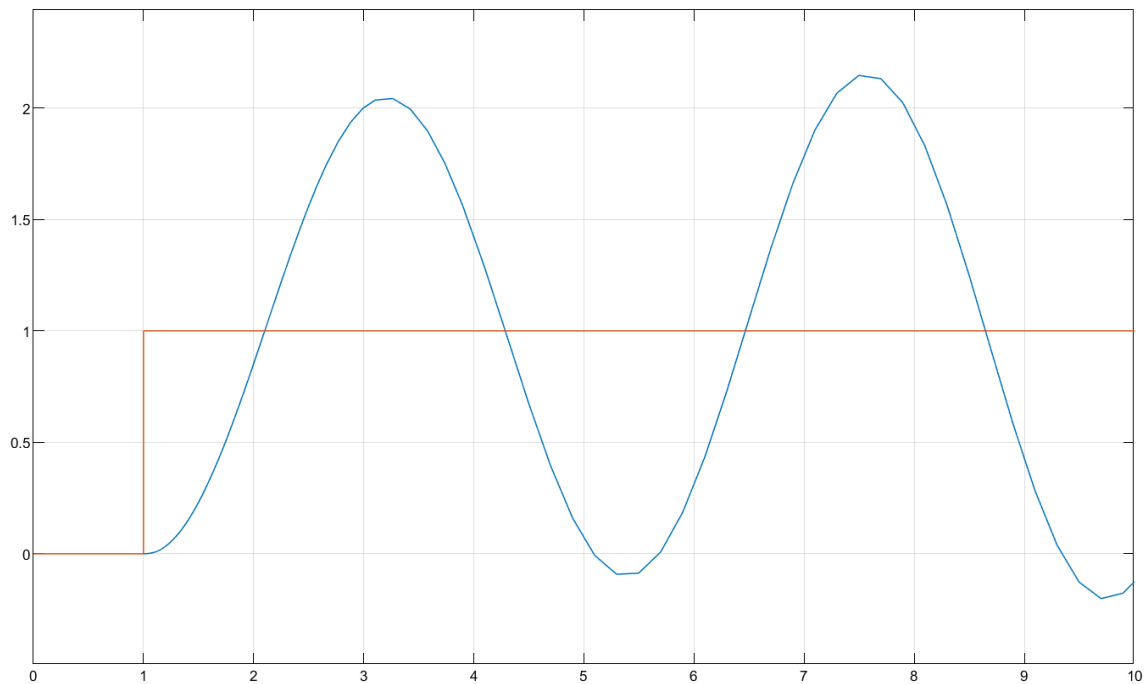
$$x_E(s) = K_{cuentas-rev} \cdot K_{long-cuenta} \cdot \theta(s)$$


Encoder (m/teta)

Para terminar, solo falta implementar dichas funciones en un diagrama de bloques. Para ello tenemos dos opciones destacables: realimentación simple o múltiple, como en este caso solo queremos controlar el parámetro de la posición no es necesario utilizar una realimentación múltiple. Si se deseara controlar la velocidad se utilizaría dicho diagrama.



Simulando, con una entrada step obtendríamos el resultado siguiente:



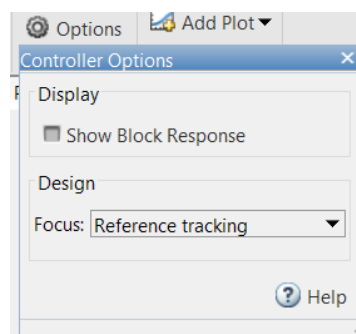
Como podemos observar esta primera estimación del sistema no es aceptable comparándolo con el resultado que se busca. Sería un sistema sin amortiguamiento e inestable. Para solucionar esto añadiremos un regulador PID.

2) Diseño de un regulador PID para el control en posición de dicho servoeje.

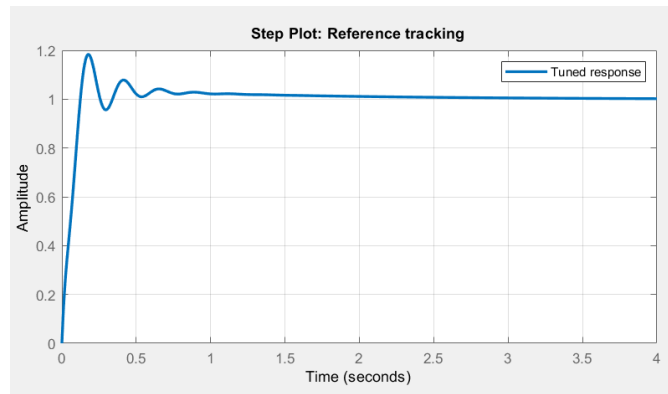
Añadiendo un controlador PID después de la realimentación/error, y ayudándonos con el PID Tuner de Matlab para encontrar un resultado que se ajuste a lo que buscamos.

Primero nos genera una función inicial.

Podemos elegir como deseamos que ajuste el sistema, en nuestro caso queremos que siga una referencia, ósea que entrando una posición el sistema llegue a ella.



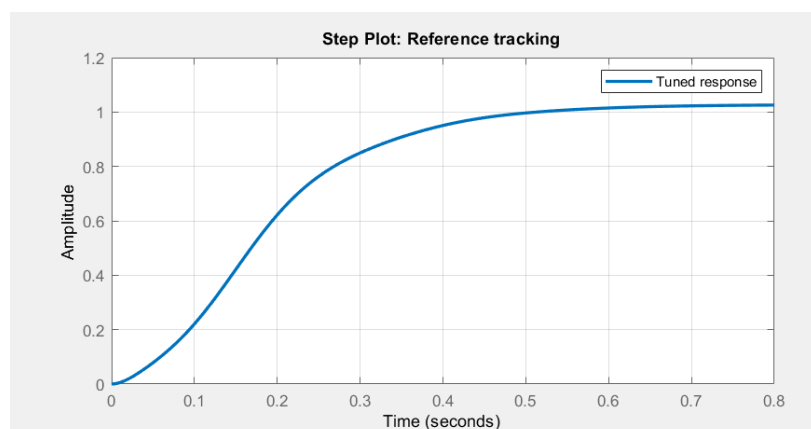
Observamos que solo con esto el sistema ha mejorado, pasando de inestable y sin amortiguamiento a subamortiguado.



Utilizando las Tuning Tools: modificando el tiempo de respuesta y el comportamiento en el transitorio, dentro del PID Tuner, podemos ajustar el sistema a lo que buscamos. En nuestro caso queremos una sobreoscilación lo más pequeña posible y cercana a 0, y que sea estable y rápido en el transitorio, dentro de las posibilidades del sistema.

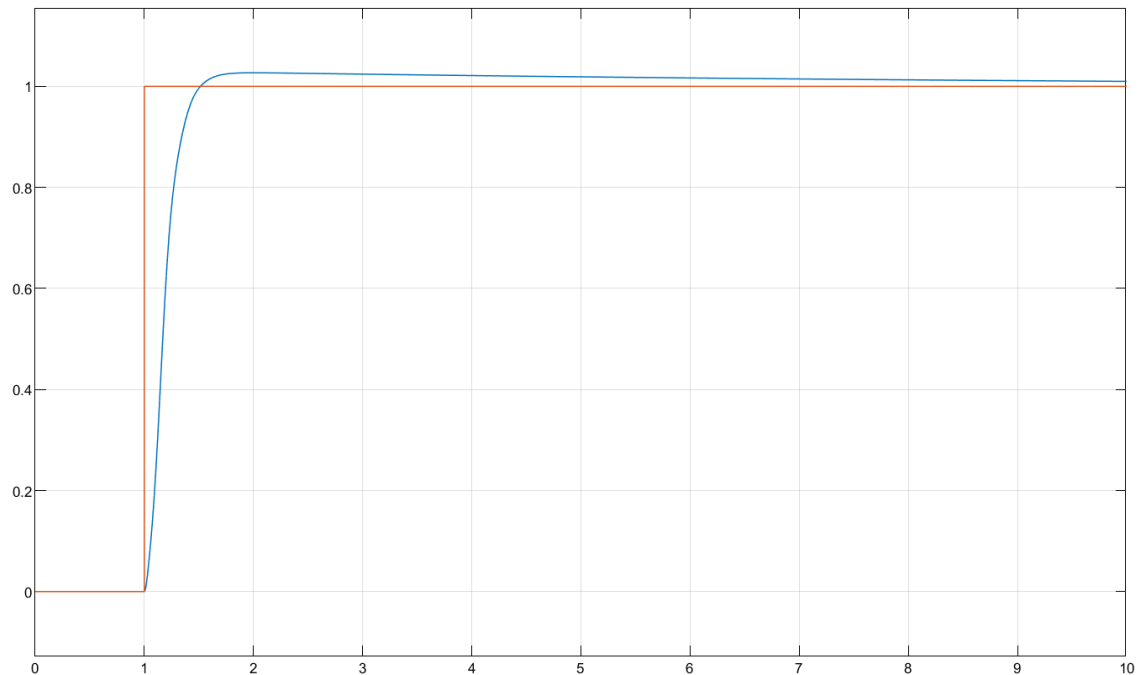
Controller Parameters	
	Tuned
P	0.36765
I	0.011431
D	2.4711
N	27.6166
Performance and Robustness	
	Tuned
Rise time	0.283 seconds
Settling time	NaN seconds
Overshoot	2.69 %
Peak	1.03
Gain margin	18 dB @ 24 rad/s
Phase margin	72 deg @ 5.06 rad/s
Closed-loop stability	Stable

Modificando la robustez del transitorio y el tiempo de respuesta, llegamos a una sobreoscilación mínima del 2.69%.



La respuesta seguiría esa forma.

Volviendo a simular y sacando la respuesta por el scope vemos el siguiente resultado.



Este resultado se ajusta al comportamiento que queremos lograr con el sistema. Quedando por tanto los valores del regulador PID:

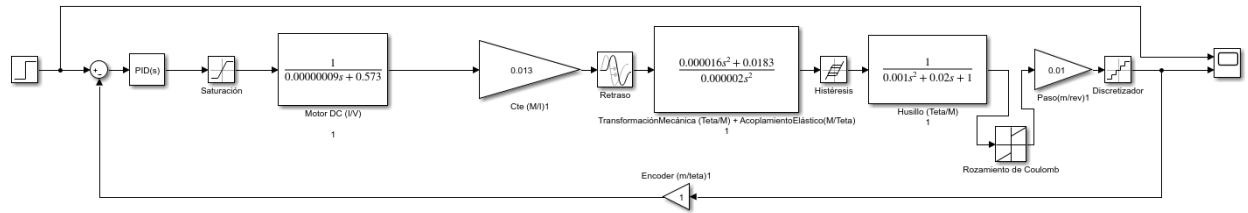
Controller parameters	
Source:	internal
Proportional (P):	0.367654837756262
Integral (I):	0.011430962036
Derivative (D):	2.47107077031181
<input checked="" type="checkbox"/> Use filtered derivative	
Filter coefficient (N):	27.6165777369446

Con la acción proporcional (P) hago el sistema más ágil, con la acción integral (I) minimizo el error y con la acción derivativa (D) mejoro el transitorio. Por eso la D es el valor más alto, es lo que peor estaba en el sistema, era inestable.

3) Representación en Simulink del diagrama de bloques del servoeje incluyendo las no linealidades.

Para este apartado incluiremos todas las no linealidades a la vez en diagrama, para el siguiente las iremos analizando una a una.

Añadiendo las no linealidades el sistema quedaría de la siguiente forma:

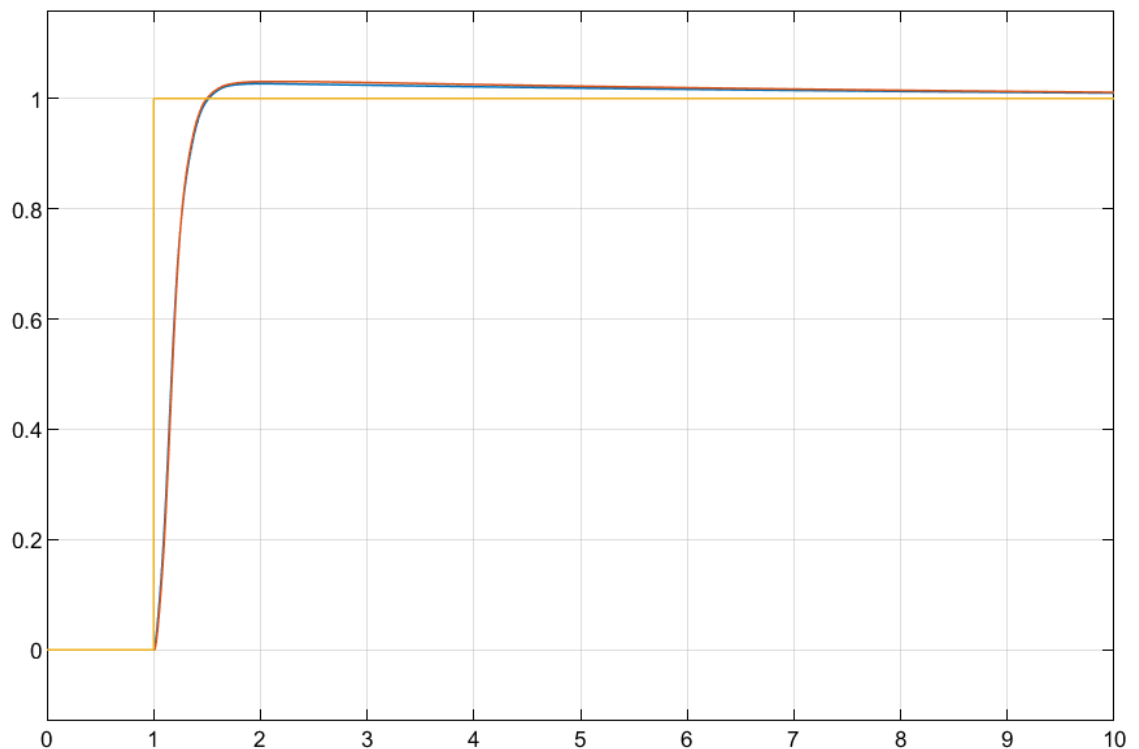


Se coloca cada discontinuidad después de la función del elemento al que suele repercutir en la realidad.

4) Observar el efecto que produce en el control un cambio en los parámetros de las no linealidades.

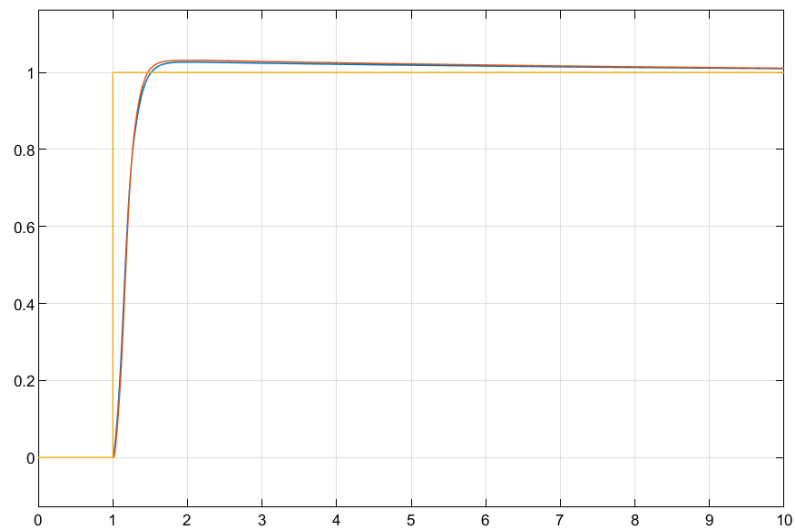
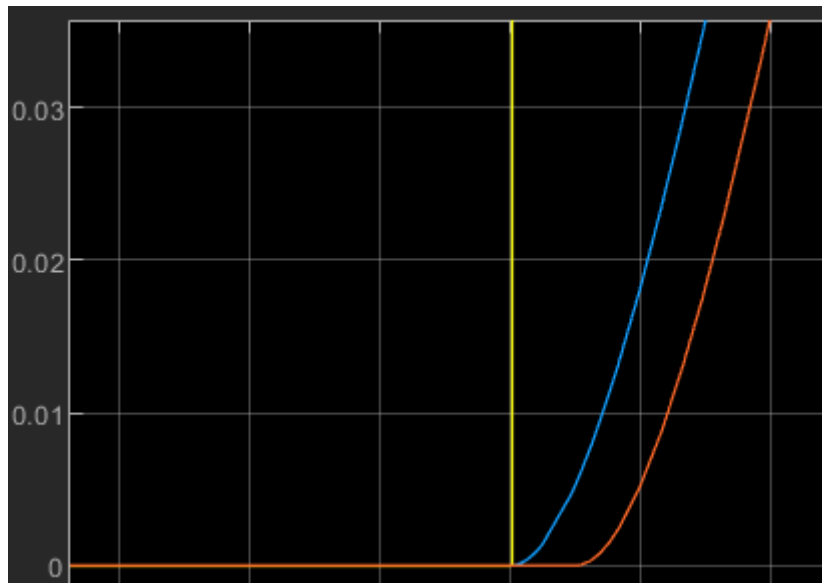
Analizando una a una las no linealidades, comparándolas con el sistema sin no linealidades y cambiando el valor de ellas.

- Retraso: Para una de las pruebas utilizamos el dato que nos daban de 6.28 ms.

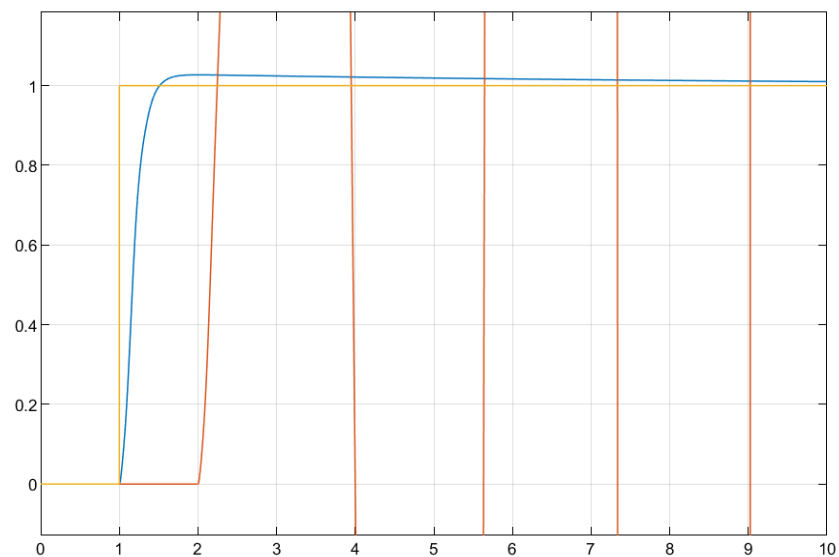


Como se puede observar dicho retraso es mínimo, hablamos de ms y no se observa diferencia comparándolo con el sistema sin no linealidades.

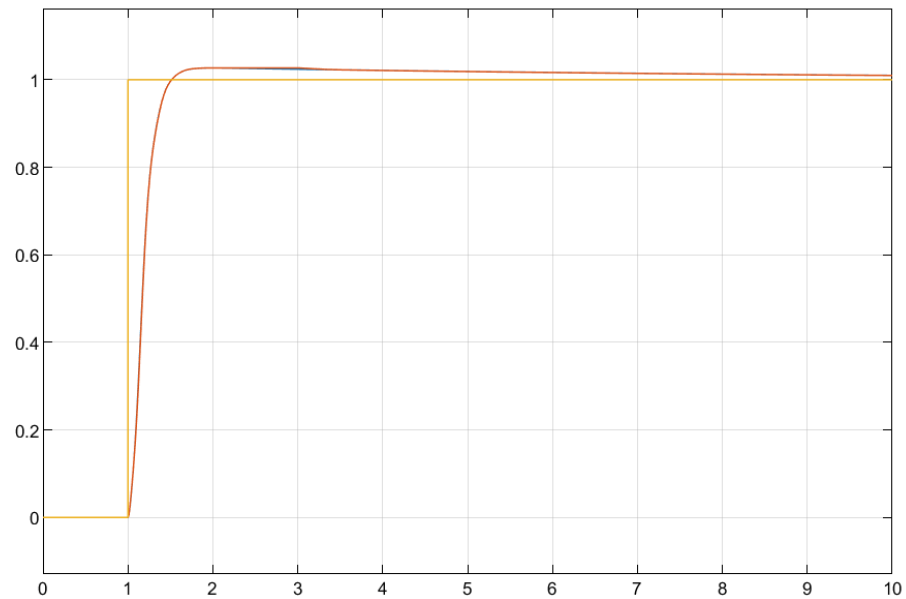
Cambiando el valor a unos 0.01 segundos y ampliando, observamos la situación normal de retraso, el sistema tarda en reaccionar.



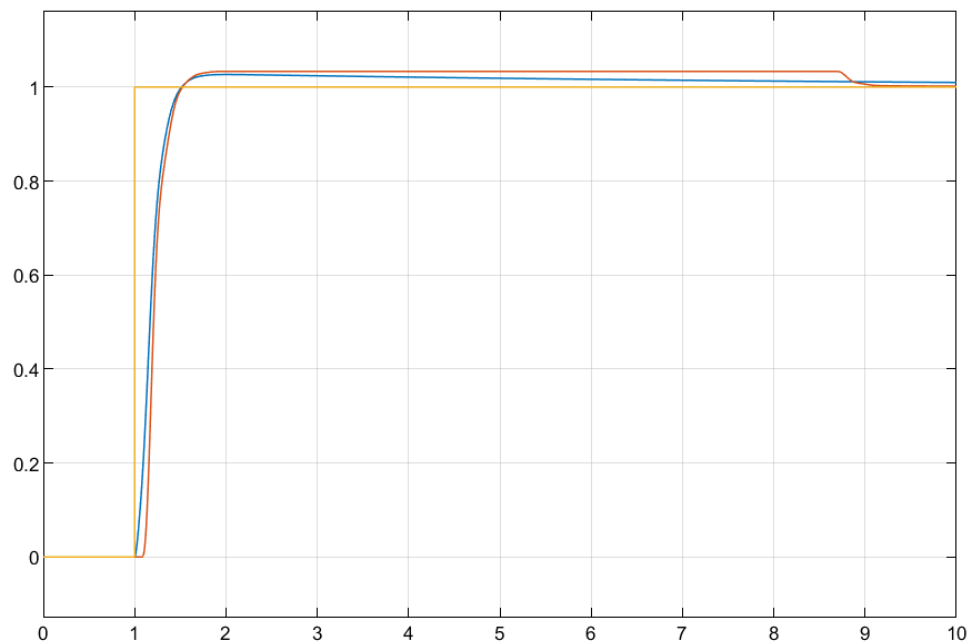
Si ponemos un valor más grande aún, observamos que el sistema cambia completamente. Ejemplo 1 segundo.



- Histéresis: Partiendo del valor automático de Matlab, un 1, solo se ve un pequeño cambio.

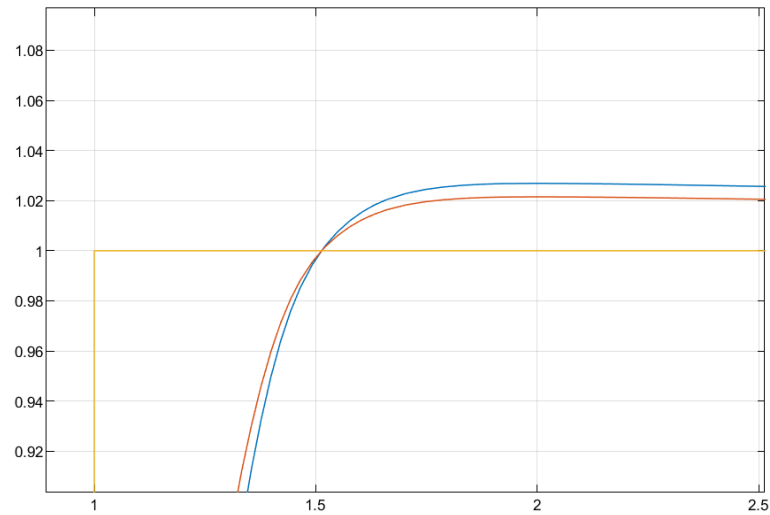


Subiendo, exagerando dicho valor observamos la diferencia. (Con 50)



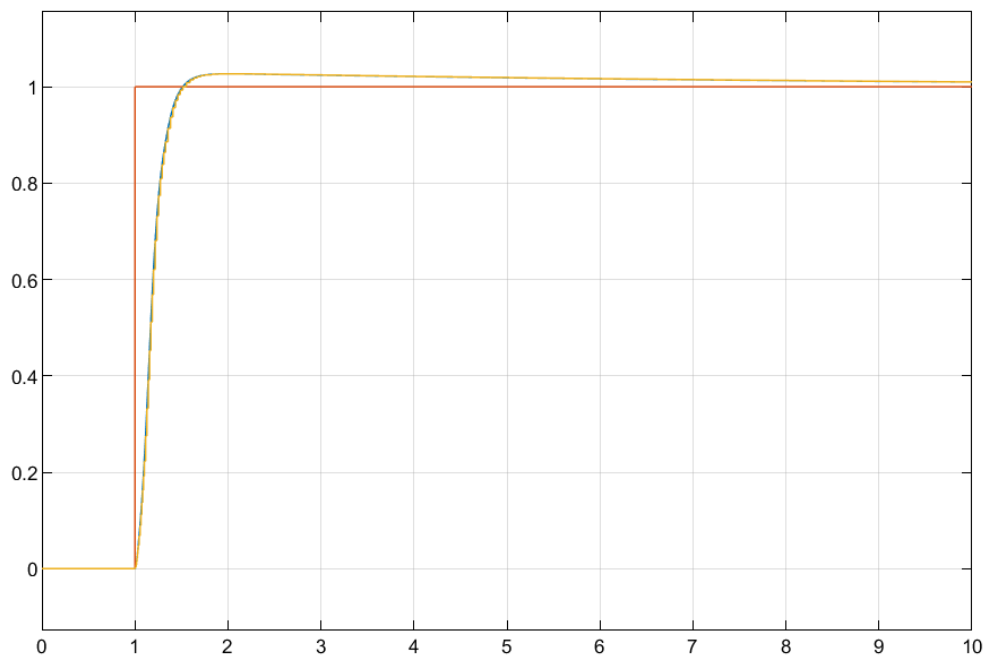
La histéresis será la responsable del error de inversión en nuestro sistema.

- Rozamiento de Coulomb. Modificando el valor de Matlab ya que daba un error con nuestro sistema. Observamos:

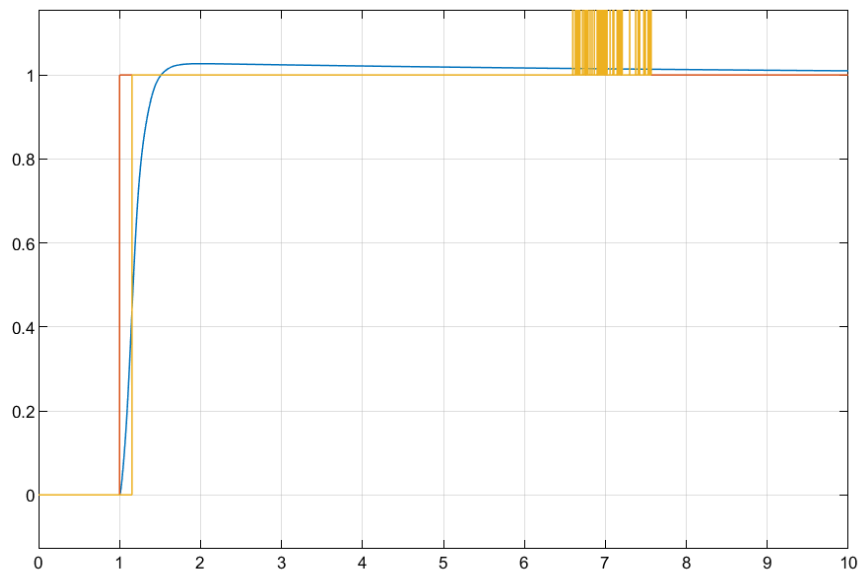


Eso para un valor de 20, como se ve en esa parte el rozamiento de Coulomb también produce error de inversión.

- Discretizador. Usando parámetros muy muy bajos, del orden de 10^{-8} , vemos resultados razonables y escalonamientos que siguen la señal.



Apenas se observarían por lo que la intentan interpolar, pero si utilizamos un parámetro 1, no sigue la forma del sistema.



El parámetro es demasiado grande para cuantizar el sistema.

- Saturación. Al saturar el sistema, imponer unos límites el regulador funciona de forma distinta, dando por tanto una señal diferente. Para una saturación superior de 0.5 y una inferior de -0.5. Se ve limitado su campo de actuación sobre el sistema real.

