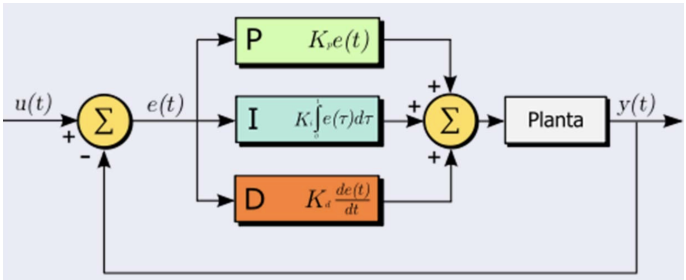
Universidad Complutense de Madrid.

Facultad de Ingeniería Informática

Inteligencia Artificial 1.



Práctica 2

Control PID

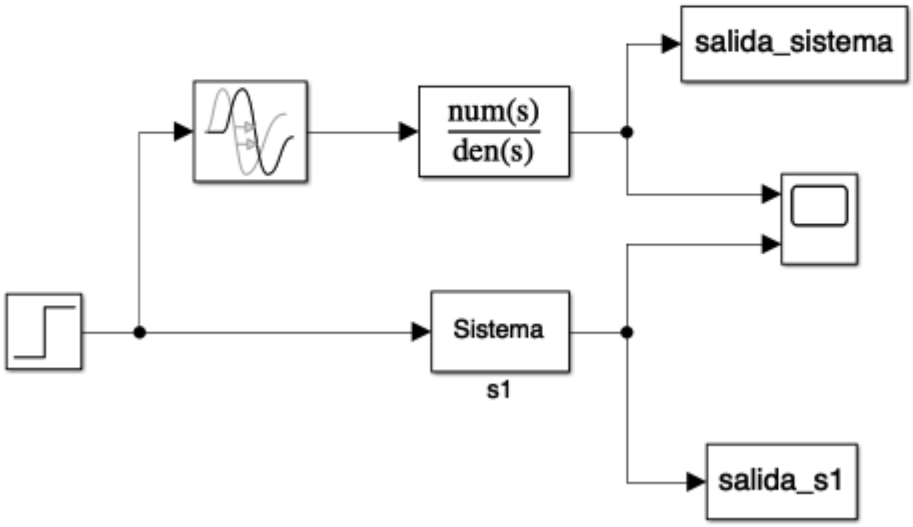
Alumno:

Borges Noronha, Frederick Ernesto

Tabla de contenido

Parte I. Identificación de un sistema sobreamortiguado mediante un modelo de primer orden.

Para este apartado se ha construido el siguiente esquema con simulink:



Y hemos realizado las mediciones pertinentes en el sistema para así obtener los valores necesarios para replicarlos con los bloques de “Transport Delay” y “Transfer Function”, dichas mediciones han sido:

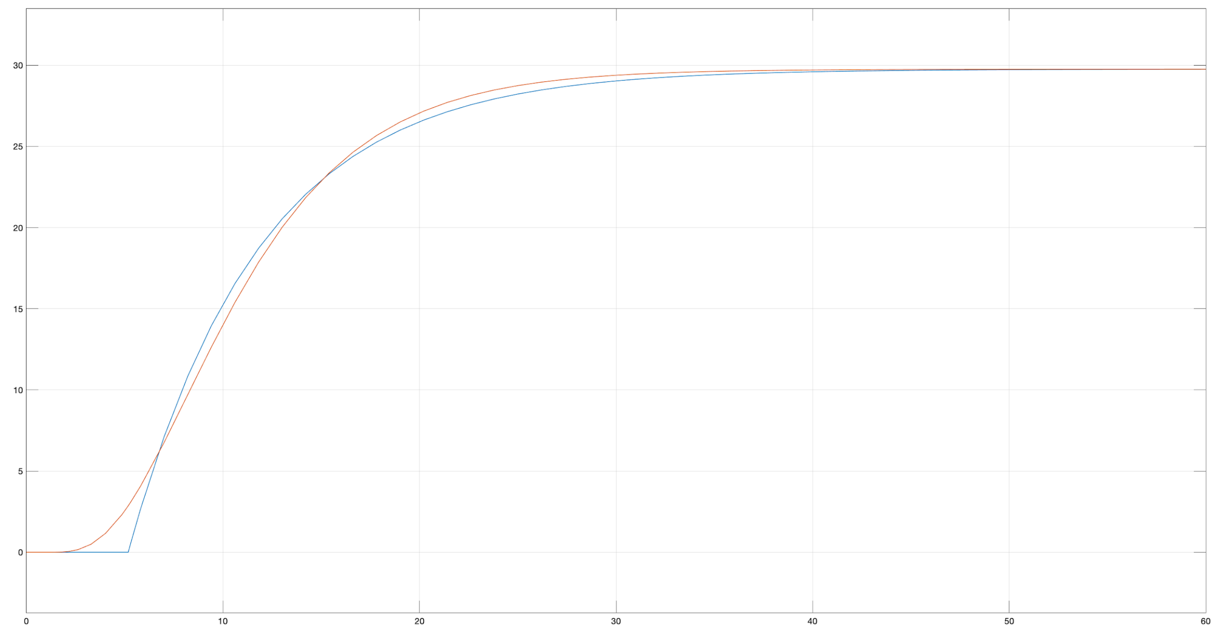
1) Obtener, con el bloque “To workspace” (indicado en el esquema como “salida_s1”), el valor de K, que es el valor en el cual se estabiliza el sistema. El valor de K obtenido ha sido: 29.760966700633900

2) Hemos trazado una recta tangente al tiempo de subida y se ha tomado nota de en qué valor de tiempo corta en el eje x. Este valor se denomina tiempo de retardo y la medida obtenida ha sido: 4.18

3) Utilizando el método de Ziegler-Nichols, hemos procedido al cálculo del valor del 0.635 * K, el cual ha resultado en 10.8627528457314

4) Y por último hemos calculado la constante de tiempo que viene dada por 0.635K - Tr, el cual ha resultado en: 6.68275284573137

Después de todo esto, hemos obtenido la siguiente gráfica, donde la línea naranja representa el sistema S1 dado por el bloque “Sistema” y la línea azul representa el sistema creado a partir de los bloques de “Transport Delay” y “Transfer Function”:

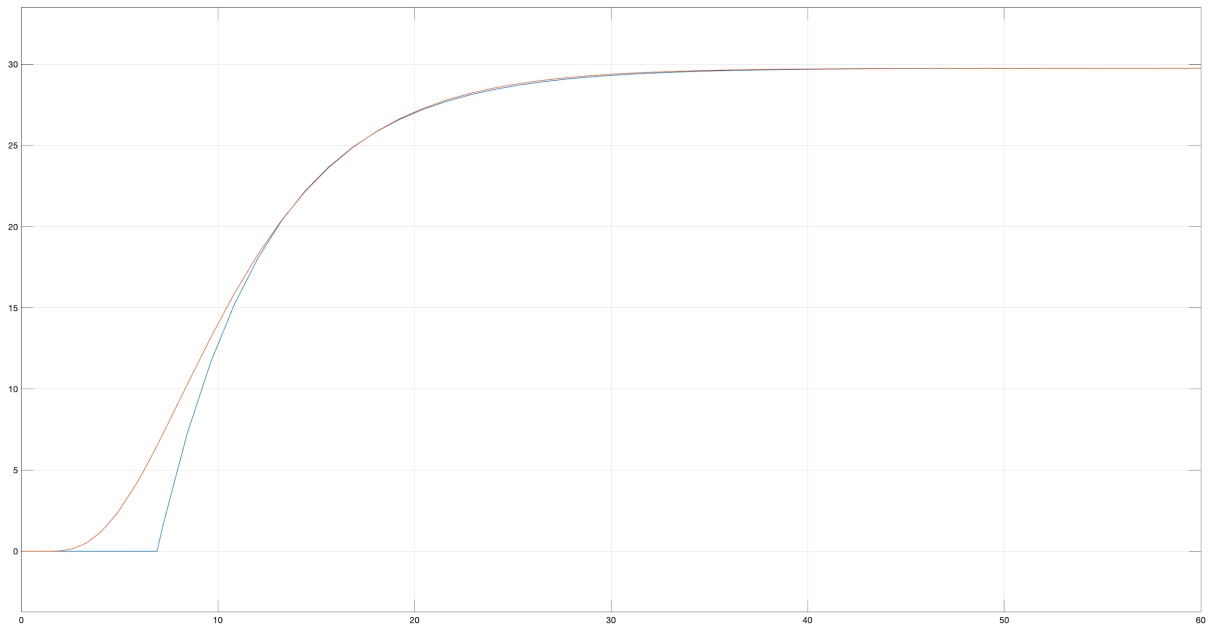


Después de obtener esta gráfica, hemos ajustado un poco los valores de forma manual para obtener un sistema que se comporta de la forma esperada, quedando la siguiente gráfica con los siguientes valores:

K = 29.760966700633900

Retardo = 5.9

Constante de tiempo = 5.5



Como podemos observar, en esta segunda gráfica la línea azul, que representa el sistema construido con los bloques de “Transport Delay” y “Transfer Function”, se comporta como el sistema dado.

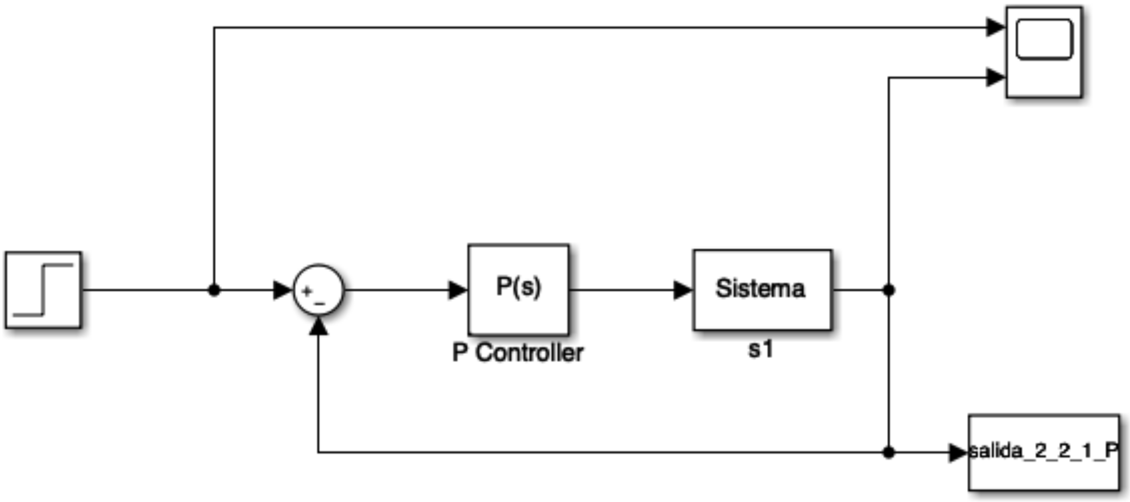
Parte II. Controlador PID.

Este apartado consta de cuatro fases:

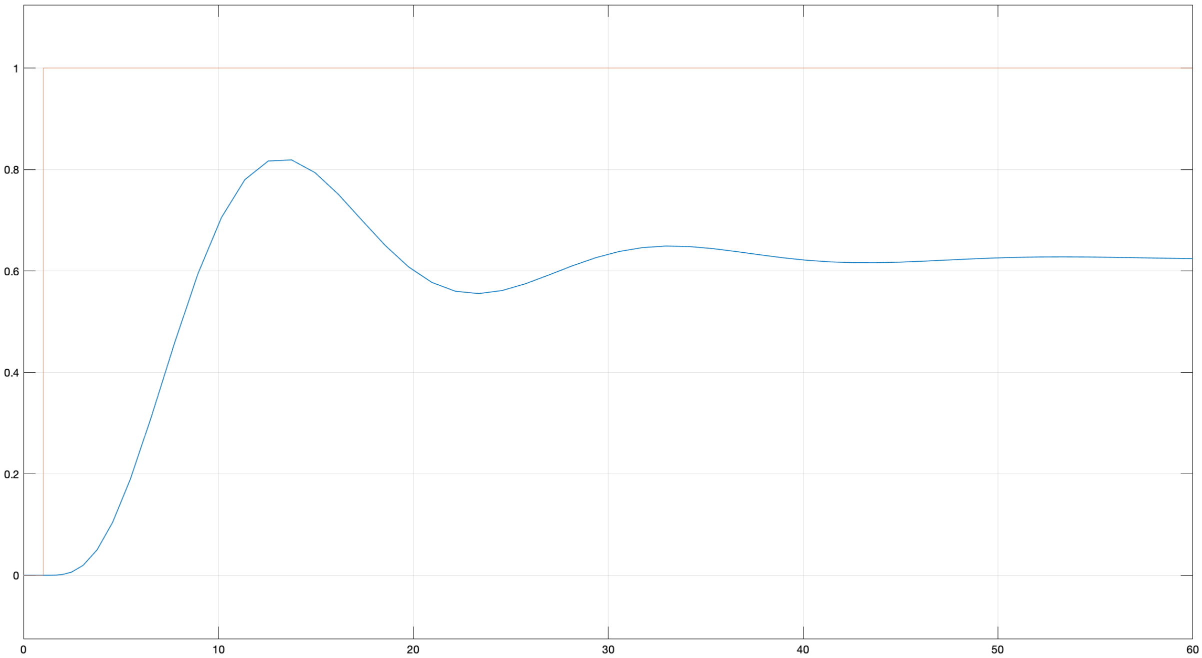
1) *Sintonizar los controladores P, PI y PID con el método de Ziegler-Nichols en un sistema de lazo abierto, para una planta de tipo 1.*

a) Controlador P

Para la sintonización del controlador P en lazo abierto se ha construido el siguiente esquema dentro de la herramienta Simulink:

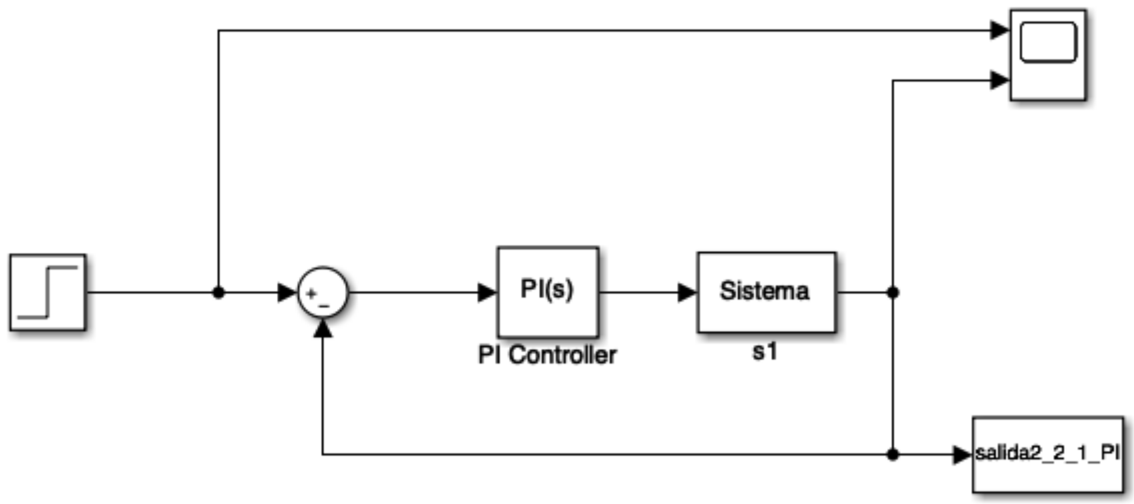


Con ayuda de los valores del apartado anterior hemos procedido a calcular el valor que debemos colocar en el controlador como ganancia proporcional, el cual es: Kp = 0.0559817858263037, quedando como resultado la siguiente gráfica:

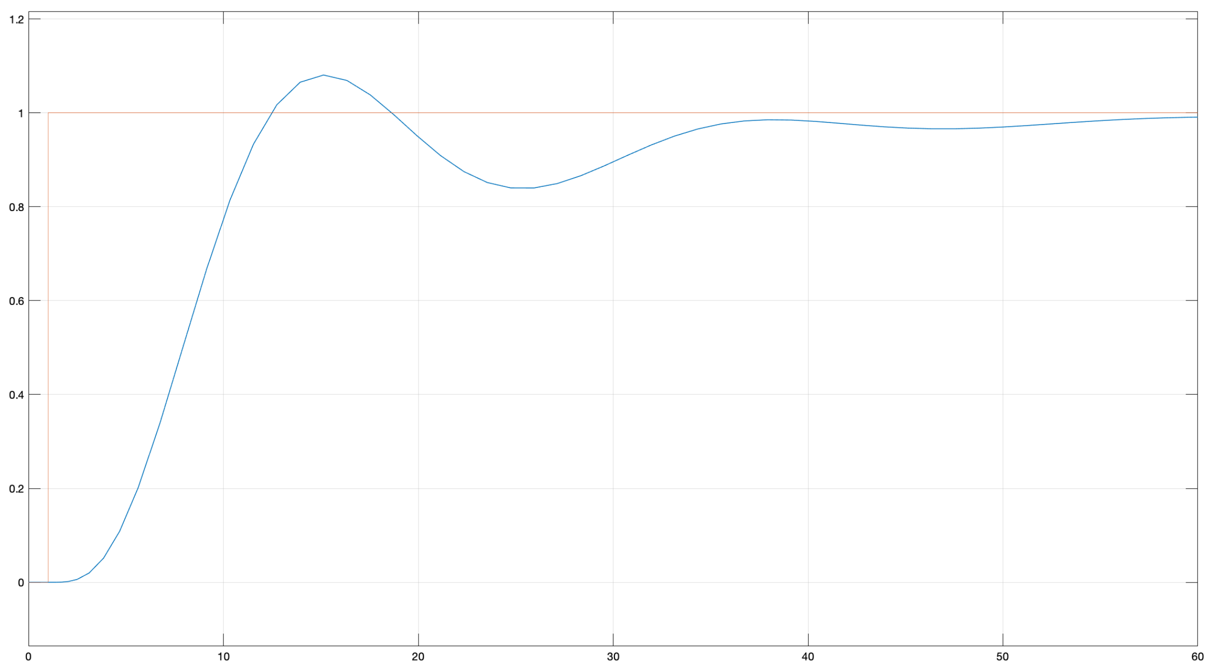


b) Controlador PI

Para la sintonización del controlador PI en lazo abierto se ha construido el siguiente esquema dentro de la herramienta Simulink:

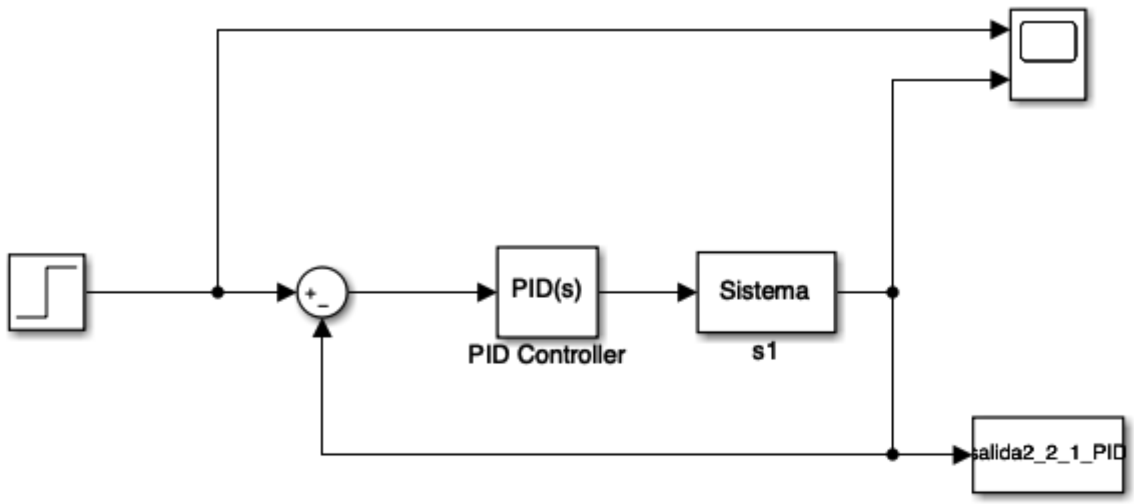


Con ayuda de los valores del apartado anterior hemos procedido a calcular los valores que debemos colocar en el controlador como ganancia proporcional y ganancia integral, los cuales son: Kp = 0.0503836072436733 y Ki = 0.00374417529955147, quedando como resultado la siguiente gráfica:

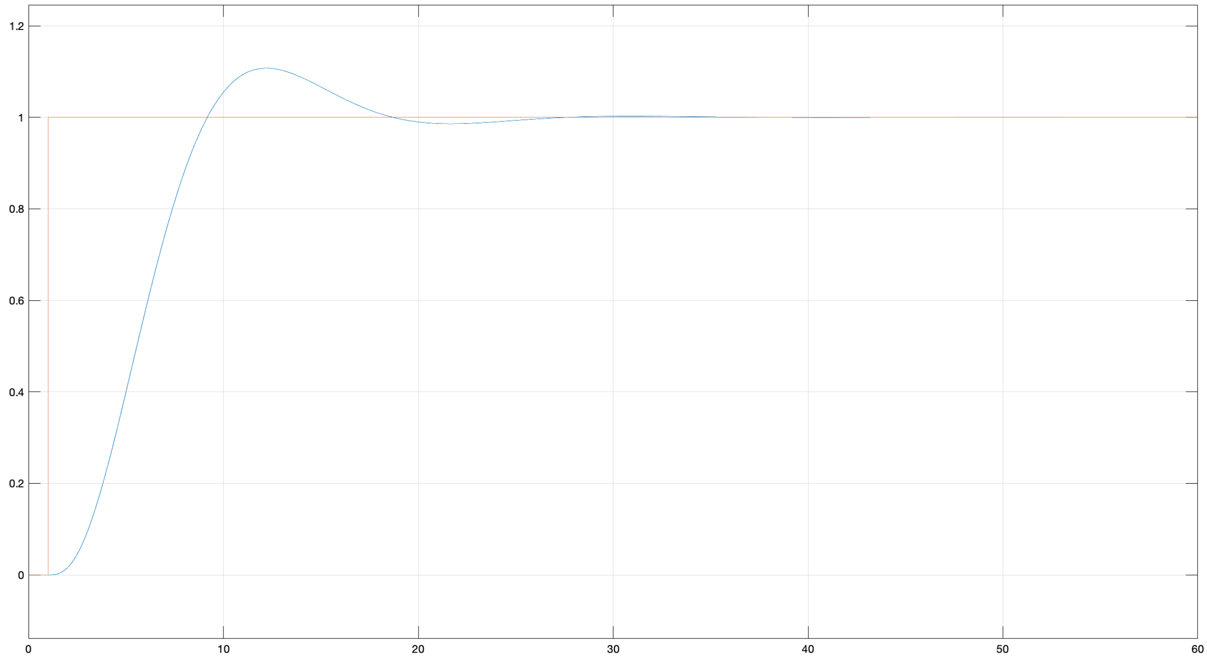


c) <u>Controlador PID</u>

Para la sintonización del controlador PID en lazo abierto se ha construido el siguiente esquema dentro de la herramienta Simulink:



Con ayuda de los valores del apartado anterior hemos procedido a calcular los valores que debemos colocar en el controlador como ganancia proporcional, ganancia integral y ganancia derivativa, los cuales son: Kp = 0.0671781429915645, Ki = 0.00831206916500426 y Kd = 0.135733437914456, quedando como resultado la siguiente gráfica:



Una vez obtenidas las 3 gráficas de comportamiento de los controladores podemos observar que la acción proporcional estabiliza la salida tras un cierto tiempo, pero el valor obtenido no es el deseado, la acción integral obtiene el valor deseado, pero manteniendo sobreelongación y por último la acción derivativa disminuye la sobreelongación que tenga el sistema.

2) *Sintonizar un PID con el método de Ziegler-Nichols en un sistema de lazo cerrado, para una planta de tipo 2.*

Para proceder a la sintonización de un PID en lazo cerrado con el método de Zieglers-Nichols hemos tenido que obtener el valor de la Ganancia Crítica (Ku) de forma práctica intentando generar un sistema críticamente estable, esto lo hemos conseguido utilizando un Ku = 0.54, luego hemos procedido a medir el tiempo entre los picos de la onda para obtener así el Tu, el cual ha sido Tu = 10,5168433493043.

Una vez obtenidos estos valores podemos proceder al cálculo de las ganancias para el controlador PID:

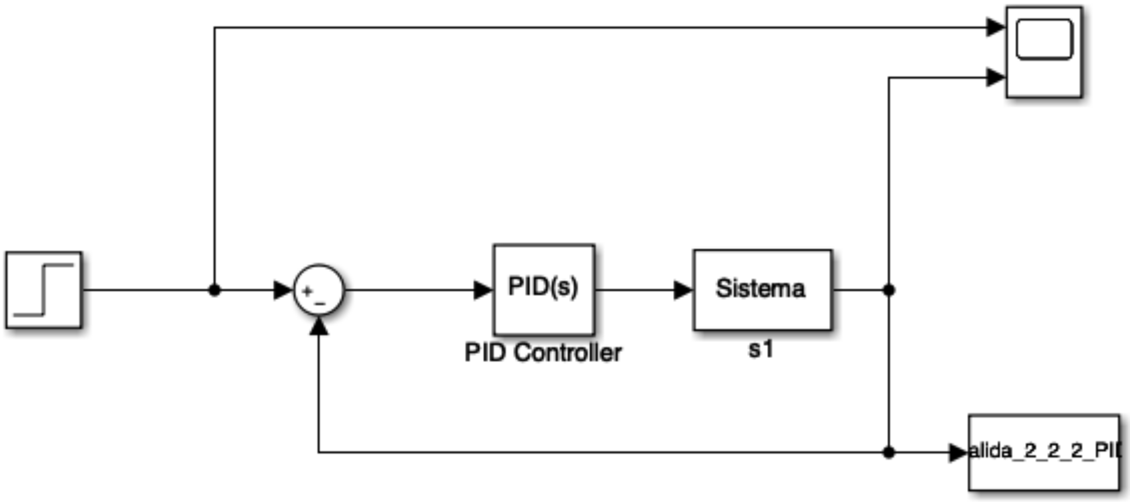
$$Ti = \frac{Tu}{2} = 5.25842167465215$$

$$Td = \frac{Tu}{28} = 1.31460541866304$$

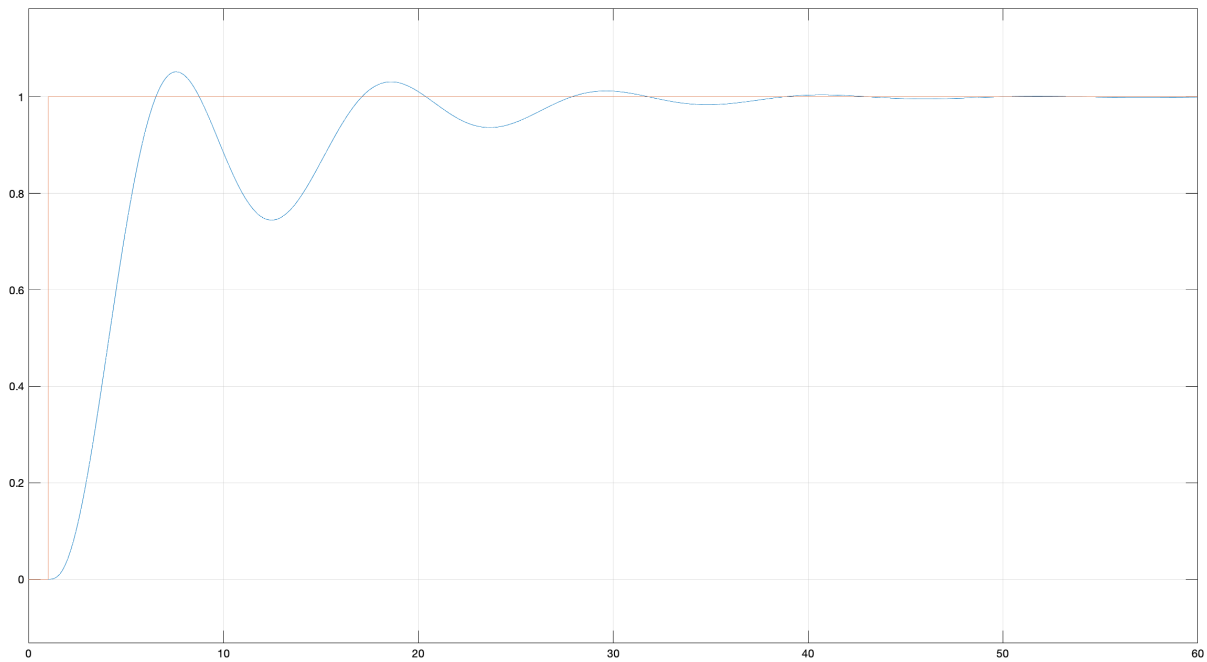
$$Kp = 0.6 * Ku = 0.324$$

$$Ki = \frac{Kp}{Ti} = 0.0616154466200037$$

$$Kd = \frac{Kp}{Td} = 0.425932155646824$$



La gráfica resultante de aplicar las ganancias obtenidas con anterioridad al sistema construido es:



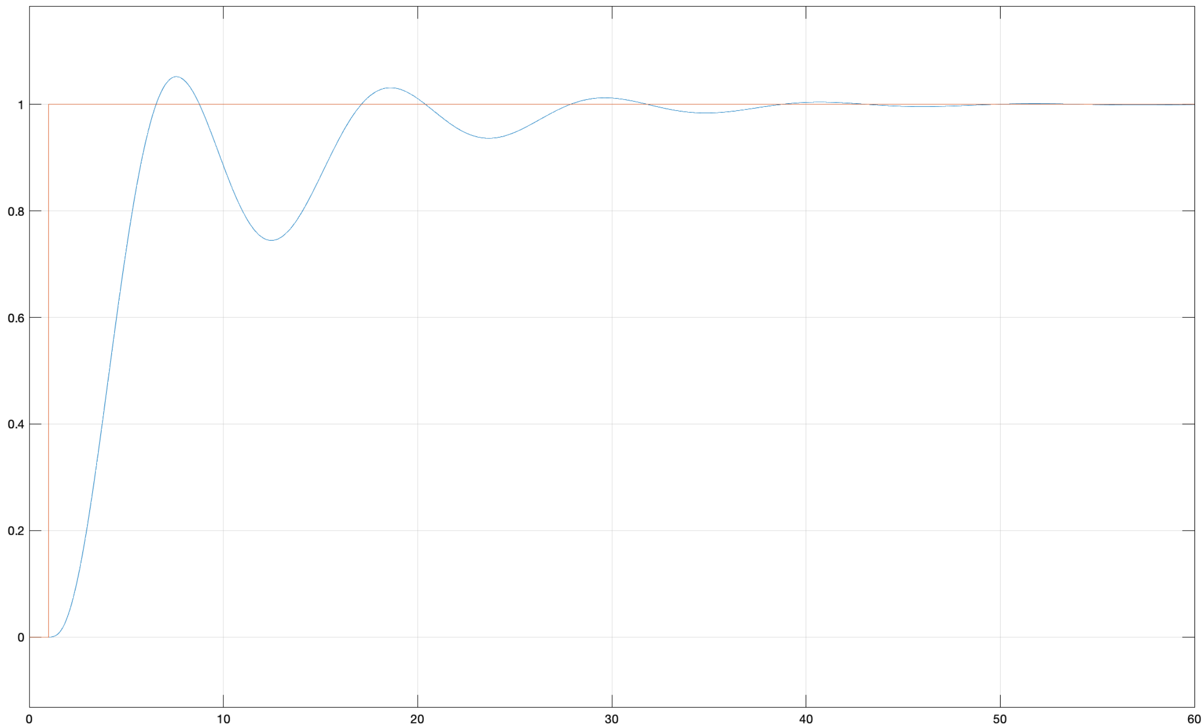
Como podemos observar, la sobreelongación inicial es poca y además, el sistema logra estabilizarse en el valor necesario en un tiempo aproximado de 40s

3) *Modificar los parámetros del PID y mirar cómo varía la respuesta.*

Basados en el apartado anterior, hemos modificado de forma cualitativa los valores y hemos obtenido los siguientes resultados:

a) Inicial:

$$Kp = 0.324$$
$$Ki = 0.0616154466200037$$
$$Kd = 0.425932155646824$$



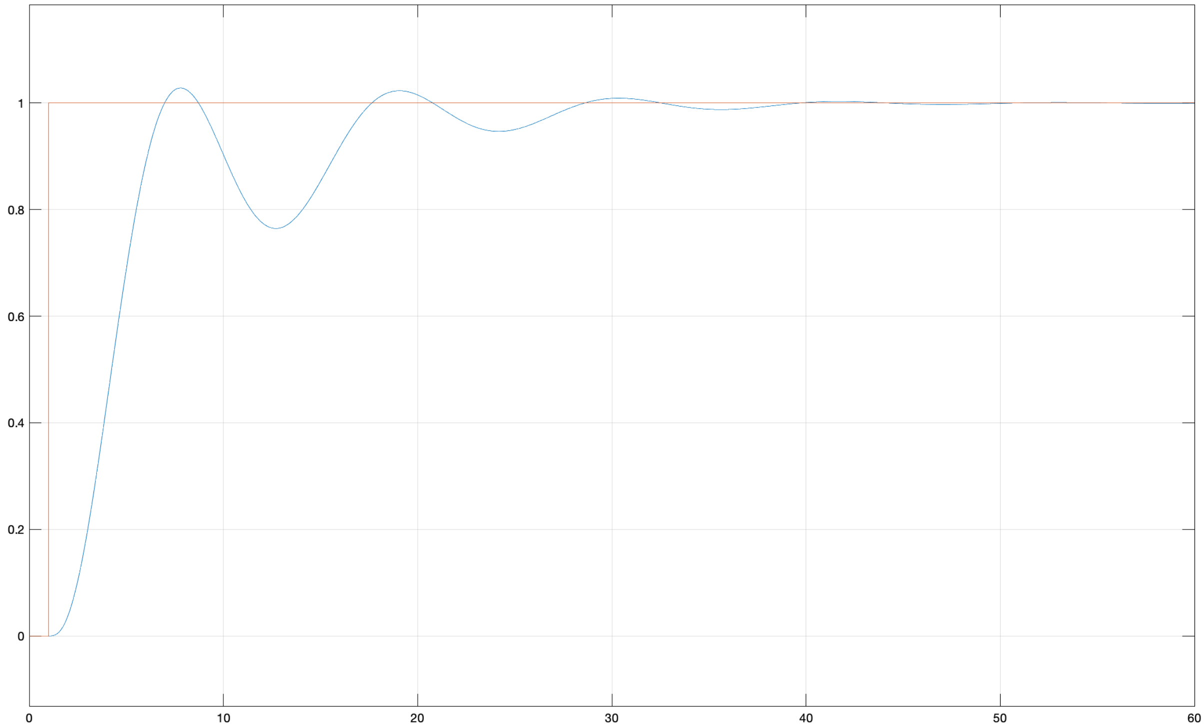
Este es el sistema con los valores calculados de forma teórica, con el que compararemos mejoras que nos ofrezcan valores modificados de forma cualitativa Kp, Ki, Kd

b) 1era modificación: Redondeamos los valores a ver que sucede con el sistema

$$Kp = 0.3$$

$$Ki = 0.06$$

$$Kd = 0.4$$



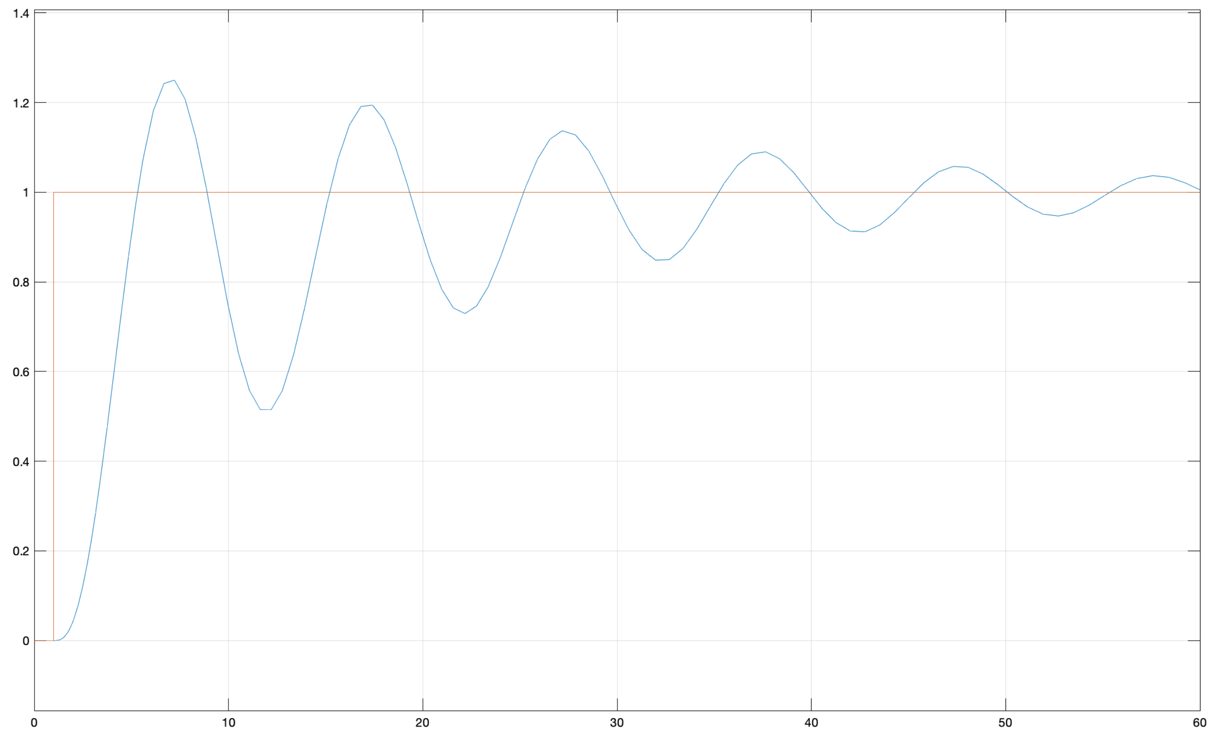
Al realizar esta acción, el sistema se mantiene en el mismo estado, pero con pequeñas variaciones imperceptibles en la gráfica.

c) <u>2da modificación: Incrementamos el valor de Kp</u>

$$Kp = 0.5$$

$$Ki = 0.06$$

$$Kd = 0.4$$



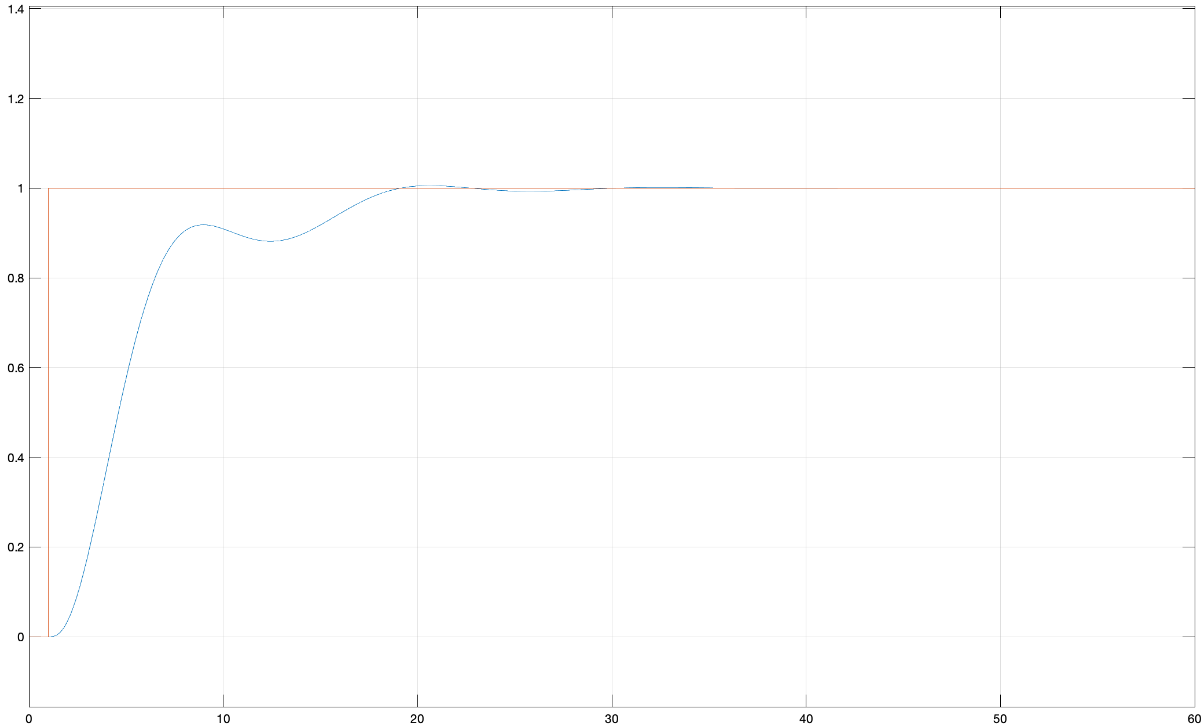
Podemos observar que la gráfica empieza a formar ondas ya que como habíamos indicado anteriormente la acción proporcional estabiliza la gráfica en el valor deseado

d) 3era modificación: Disminuimos el valor de Kp

$$Kp = 0.2$$

$$Ki = 0.06$$

$$Kd = 0.4$$



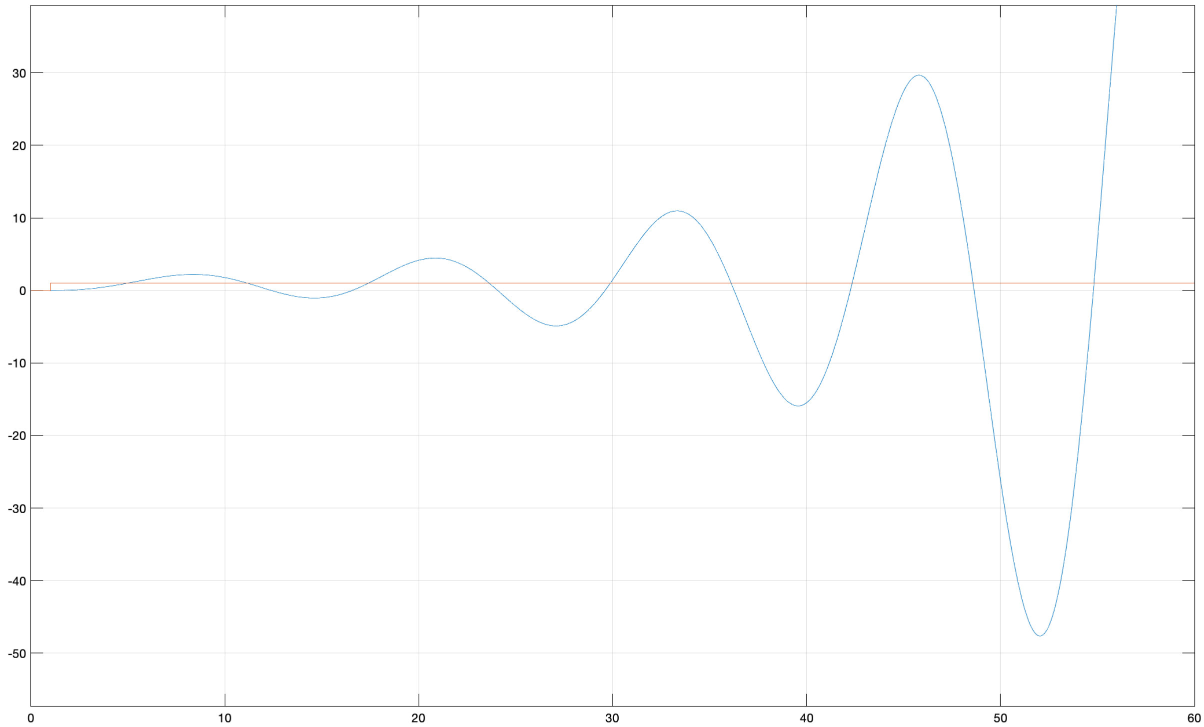
Esta gráfica nos muestra que si disminuimos el valor de Kp nos aseguramos de que el sistema no tendrá sobreelongación ya que se intenta estabilizar desde mucho antes la salida del sistema.

e) <u>4ta modificación: Aumentamos el valor de Ki</u>

$$Kp = 0.3$$

$$Ki = 0.3$$

$$Kd = 0.4$$



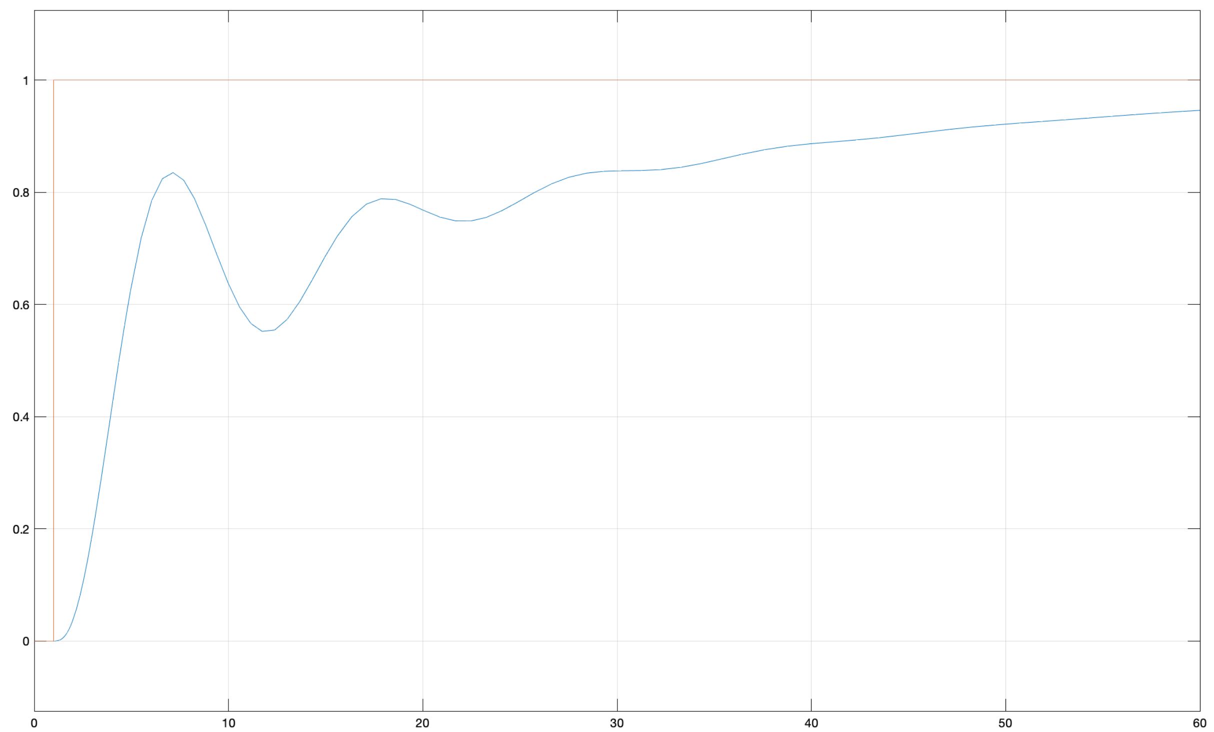
Podemos observar que el aumento de Ki hace que el sistema se vuelva inestable con el tiempo.

f) <u>5ta modificación: Disminuimos el valor de Ki</u>

$$Kp = 0.3$$

$$Ki = 0.02$$

$$Kd = 0.4$$



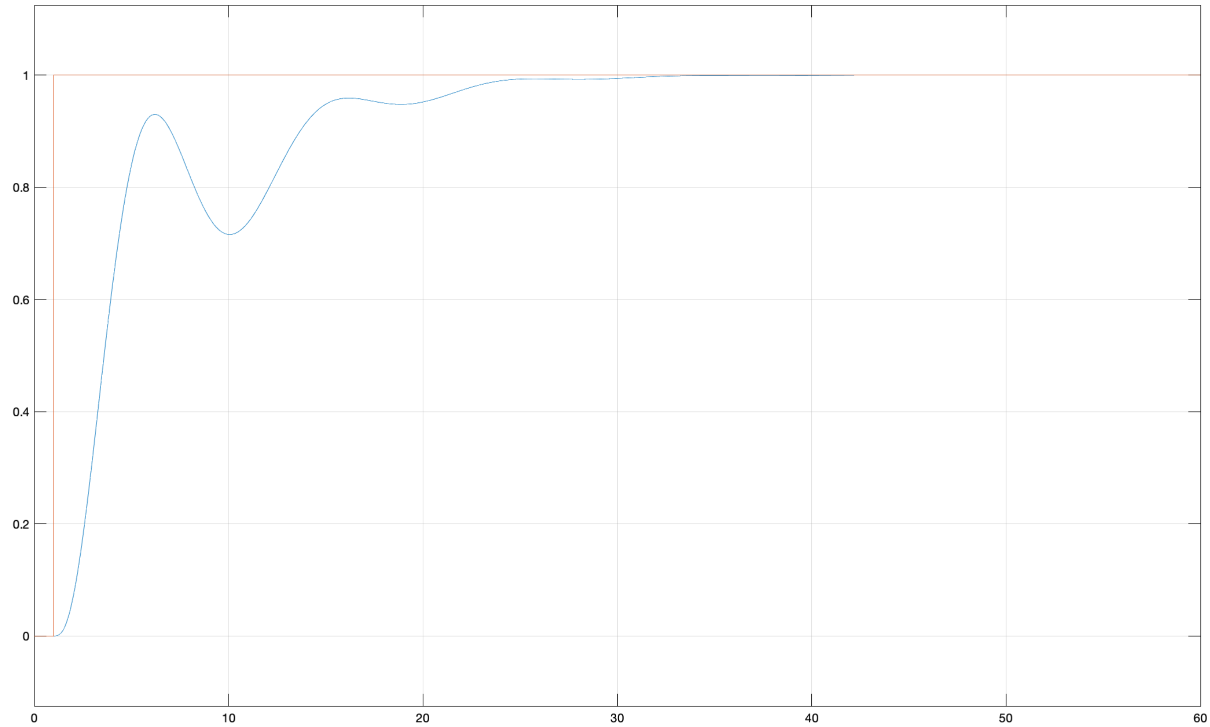
Si disminuimos el Ki, no generamos un sistema inestable, pero en la gráfica obtenida hay que destacar que no se llega nunca al valor esperado o de referencia.

g) <u>6ta modificación: Aumentamos el valor de Kd</u>

$$Kp = 0.3$$

$$Ki = 0.06$$

$$Kd = 0.8$$



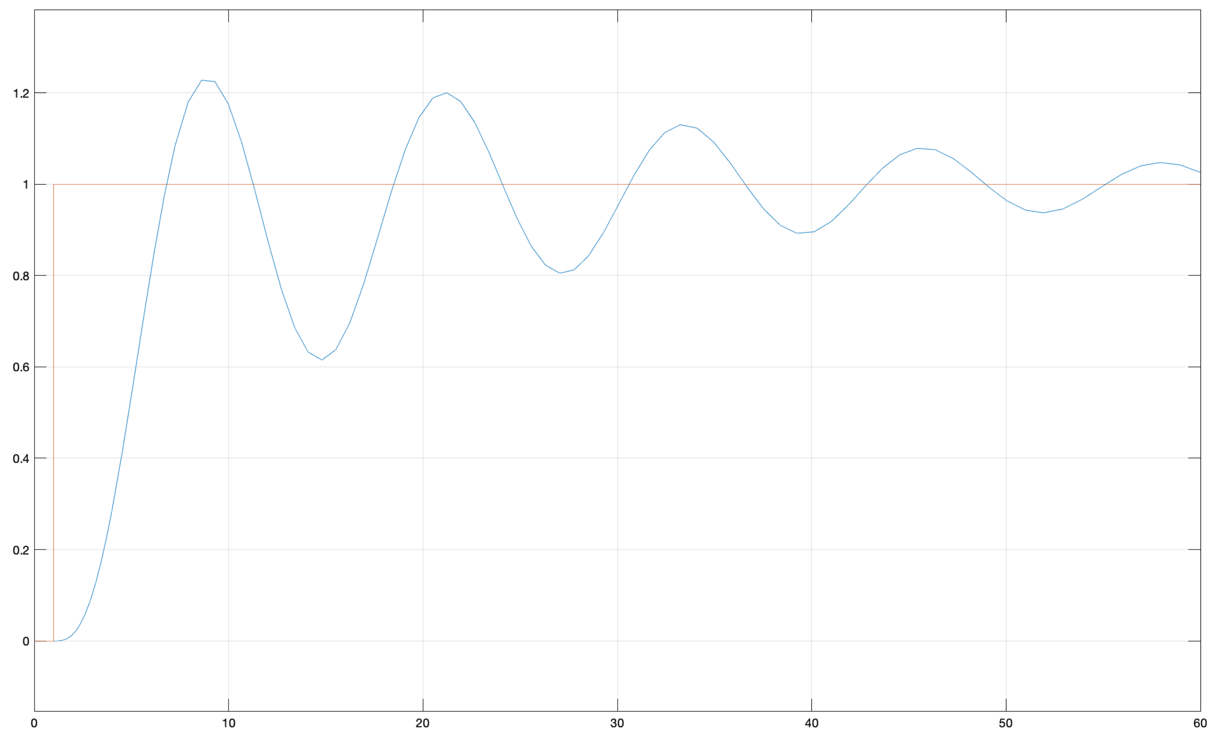
Hay que destacar que, si aumentamos el Kp, el sistema no tendrá sobreelongación, además podemos observar que sucede una acción ***similar*** a disminuir el Kp.

h) <u>7ta modificación: Disminuimos el valor de Kd</u>

$$Kp = 0.3$$

$$Ki = 0.06$$

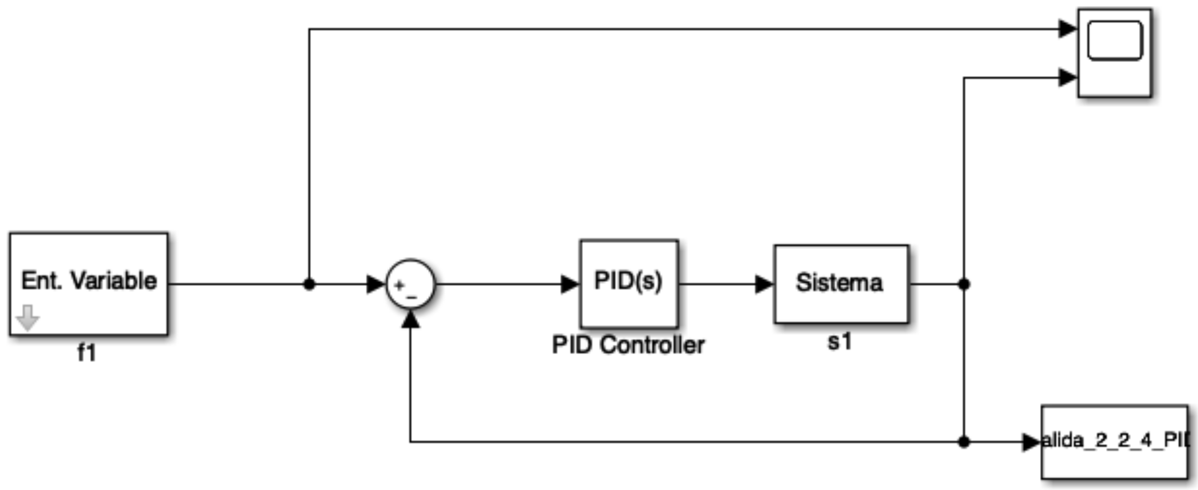
$$Kd = 0.1$$



Cuando se aumenta el valor de Kd, podemos observar que el sistema tarda mucho más tiempo en estabilizarse, lo cual puede no ser el comportamiento deseado.

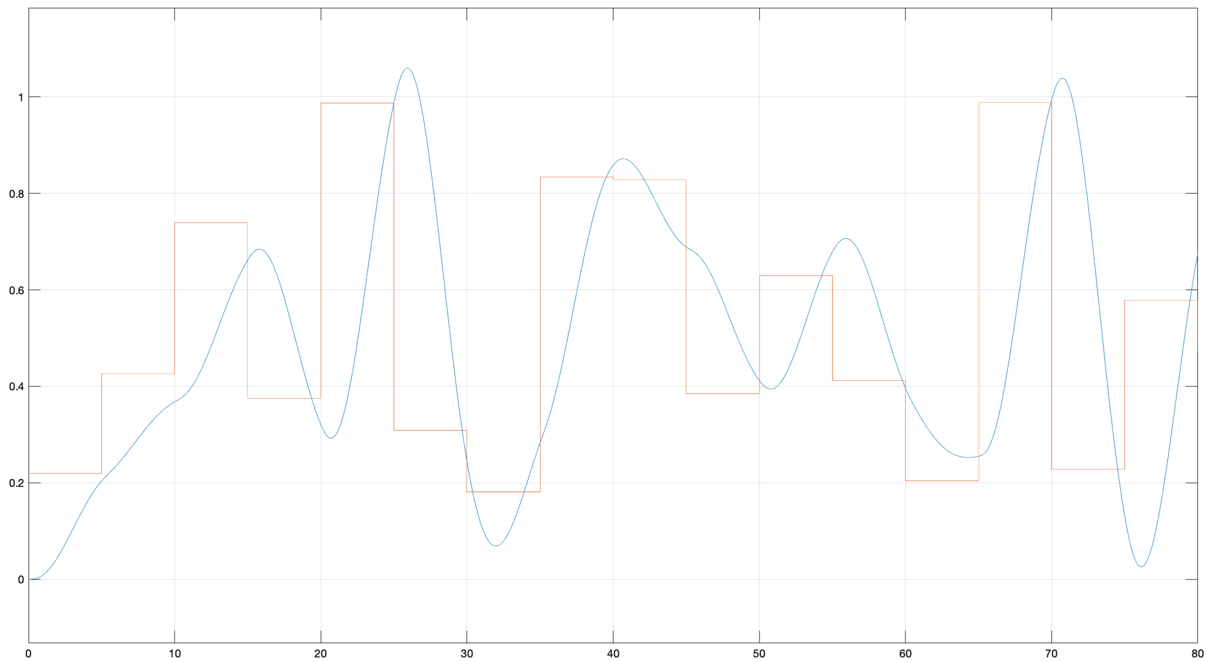
4) <u>*Cambiar el bloque de escalón unitario por uno de “Entrada Variable” y observar qué sucede con la respuesta.*</u>

Manteniendo el sistema construido en el apartado 2, se ha procedido a cambiar la entrada de escalón unitario por la de “Entrada Variable” que se ha dado en el fichero “planta.mdl”, el esquema del sistema ha quedado de la siguiente manera:



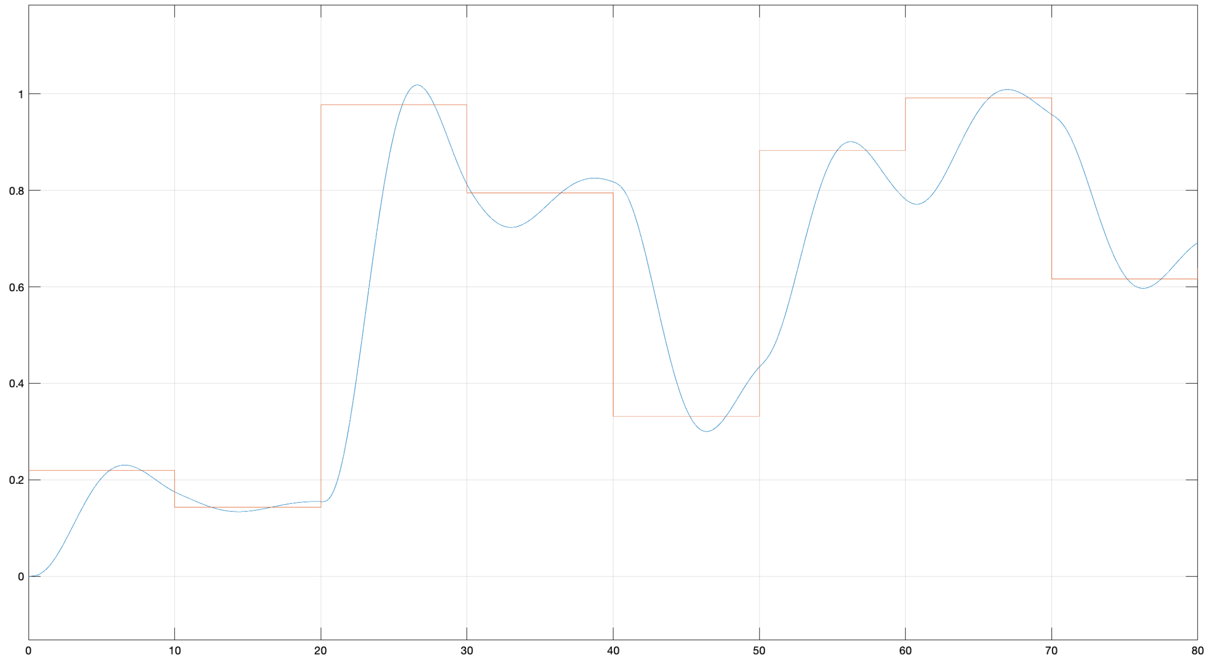
Para evaluar que tan buena es la respuesta de nuestro sistema haremos una comparativa con diferentes tiempos T de la entrada variable y así podremos ver en cual tiene mejor respuesta:

a) Gráfica para un tiempo T = 5



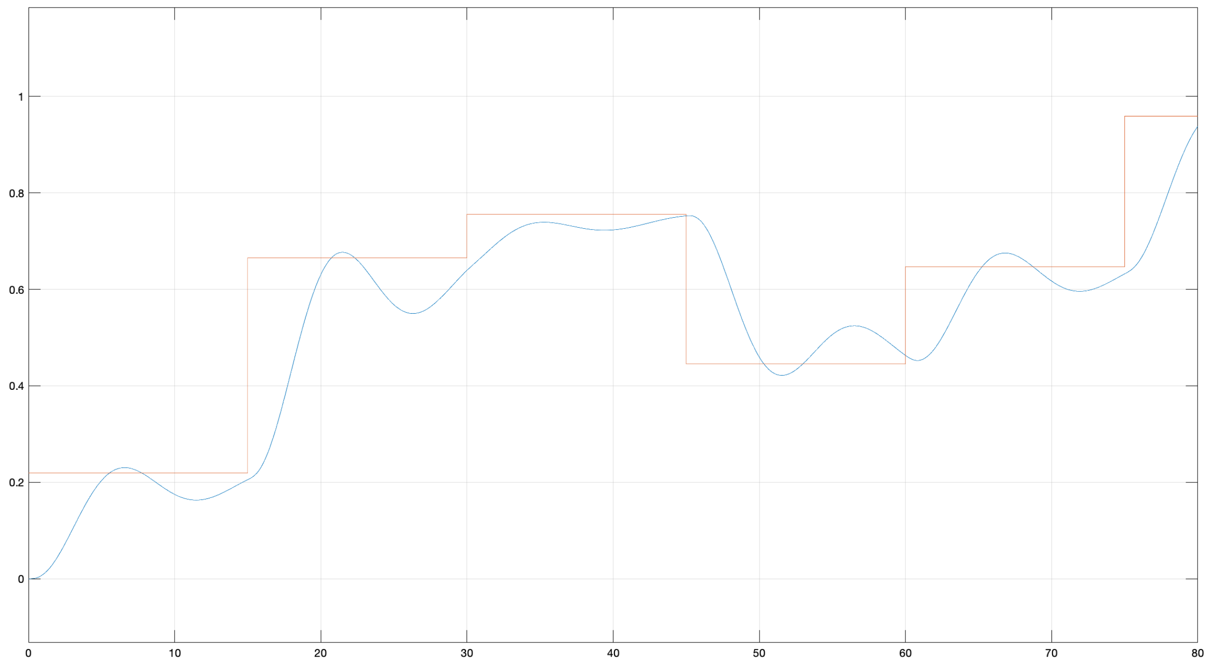
Como podemos observar, el sistema no es lo suficientemente rápido para obtener el valor deseado en un tiempo de 5s.

b) <u>Gráfica para un tiempo T = 10</u>



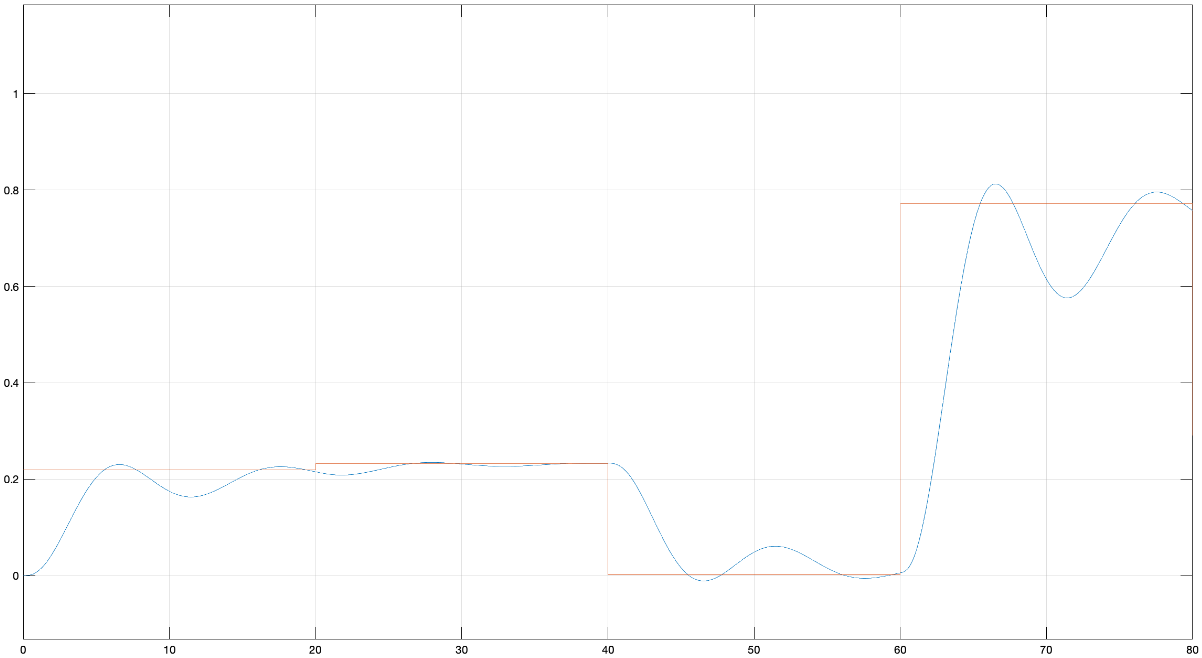
Como podemos observar, el sistema no es lo suficientemente rápido para obtener el valor deseado en un tiempo de 10s aunque podemos observar que intenta estabilizarse un poco.

c) <u>Gráfica para un tiempo T = 15</u>



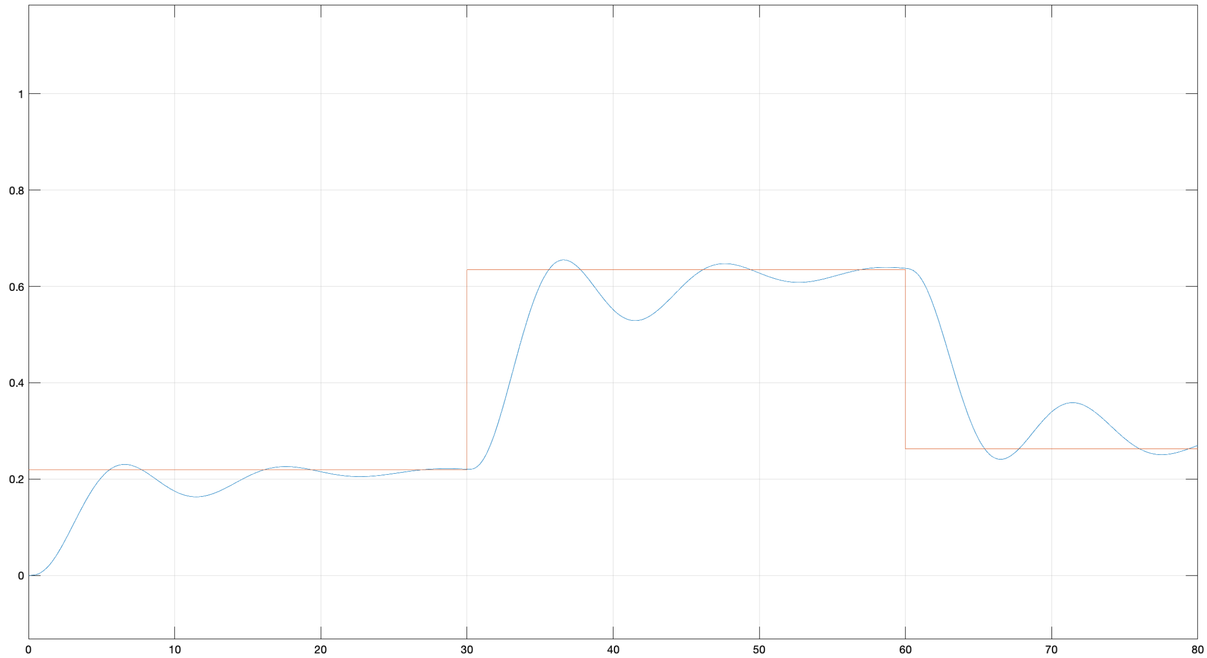
En esta gráfica podemos observar que un tiempo de 15s no es suficiente para que el sistema se estabilice completamente pero el error que presenta se puede asumir en algunos casos.

d) <u>Gráfica para un tiempo T = 20</u>



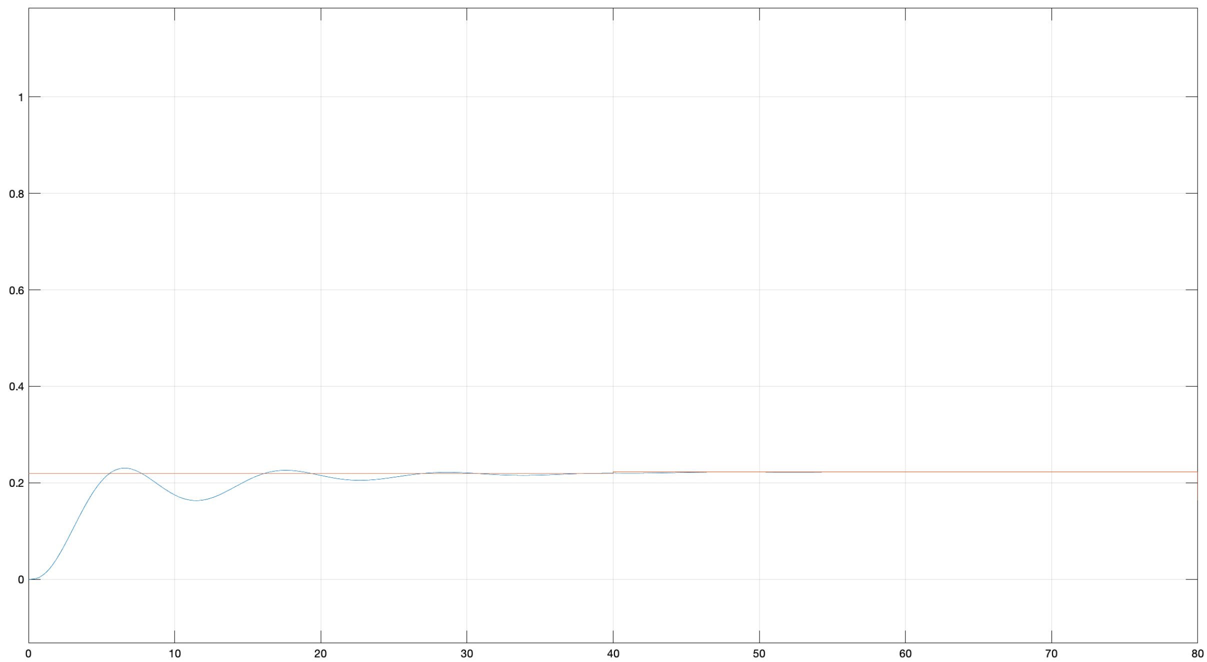
En un tiempo de 20s se puede indicar que el sistema logra estabilizarse con cambios pequeños pero sin embargo si el cambio es brusco no logra estabilizarse con el tiempo.

e) <u>Gráfica para un tiempo T = 30</u>



Como observamos en la gráfica anterior, el sistema es capaz de estabilizarse incluso con cambios bruscos, pero solo al final de cada cambio por un periodo de tiempo muy corto.

f) <u>Gráfica para un tiempo T = 40</u>



En esta última gráfica podemos observar que a partir de 40s es suficiente para obtener la salida deseada, y si no tenemos cambios bruscos en los valores entonces es mucho mejor porque logra estabilizarse mucho antes.

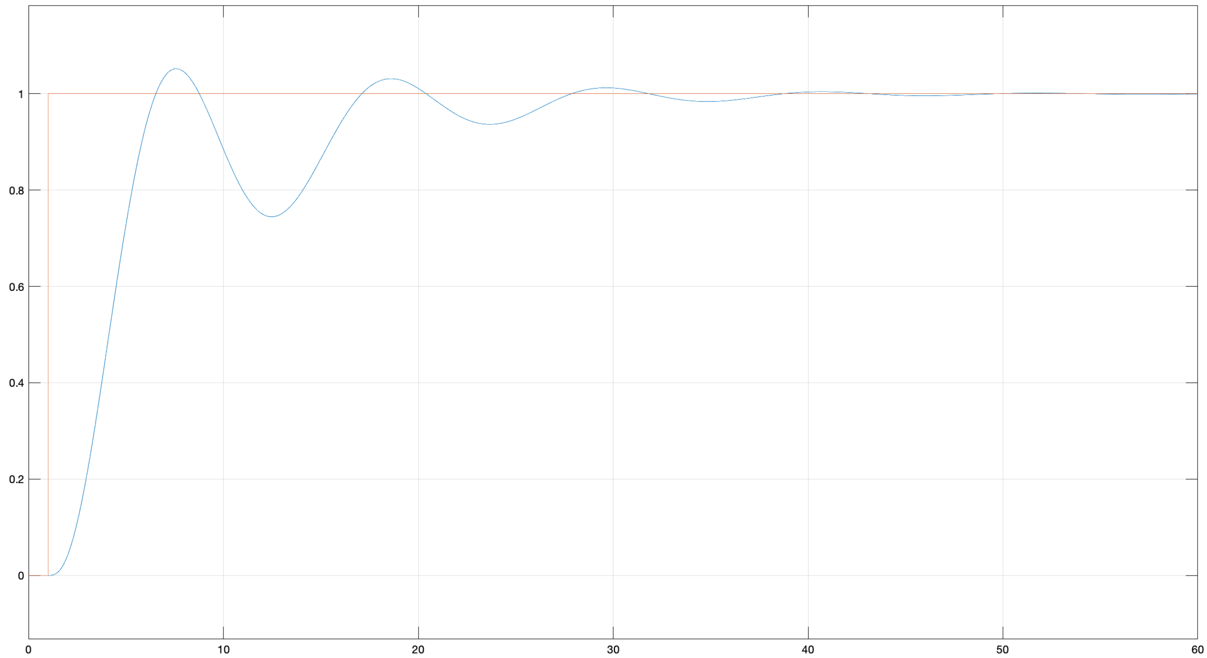
Después de obtener todas estas gráficas para poder realizar una comparativa en la cual podemos decir que, si necesitamos que el sistema tenga una salida estable, se debería utilizar un tiempo de 30s como mínimo y un óptimo de 40s.

Parte III. Controlador PID con ruido gaussiano.

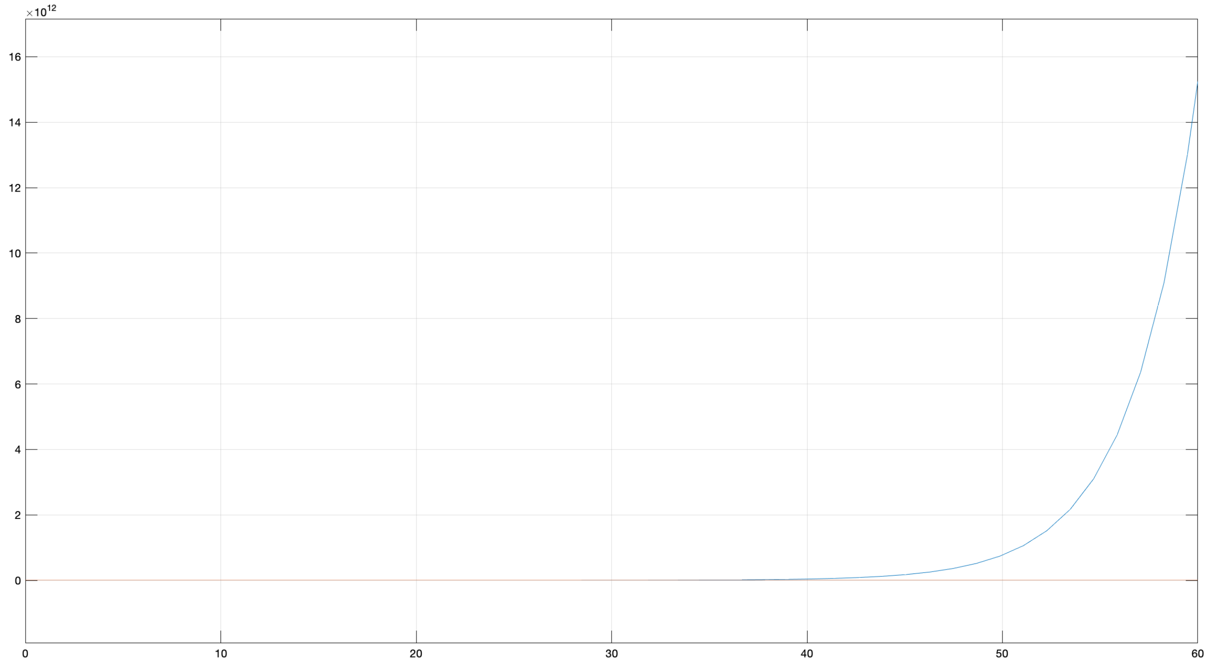
Para realizar este apartado hemos tomado como base el apartado 2.2.2 y así podemos evaluar qué sucede cuando añadimos ruido gaussiano a un a planta de tipo 2.

Para llevar a cabo este experimento presentaremos diferentes gráficas con las acciones que toma el sistema para diferentes ruidos gaussianos entre 0 y 1, tenemos los siguientes ejemplos:

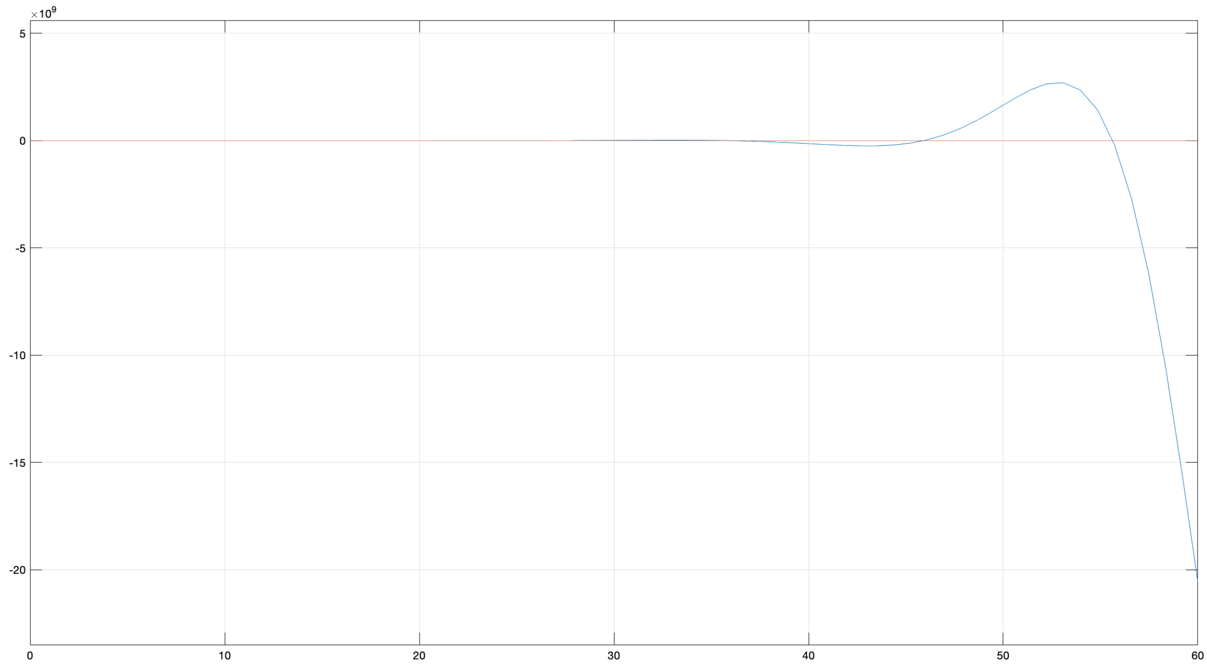
a) Ruido = 0.0



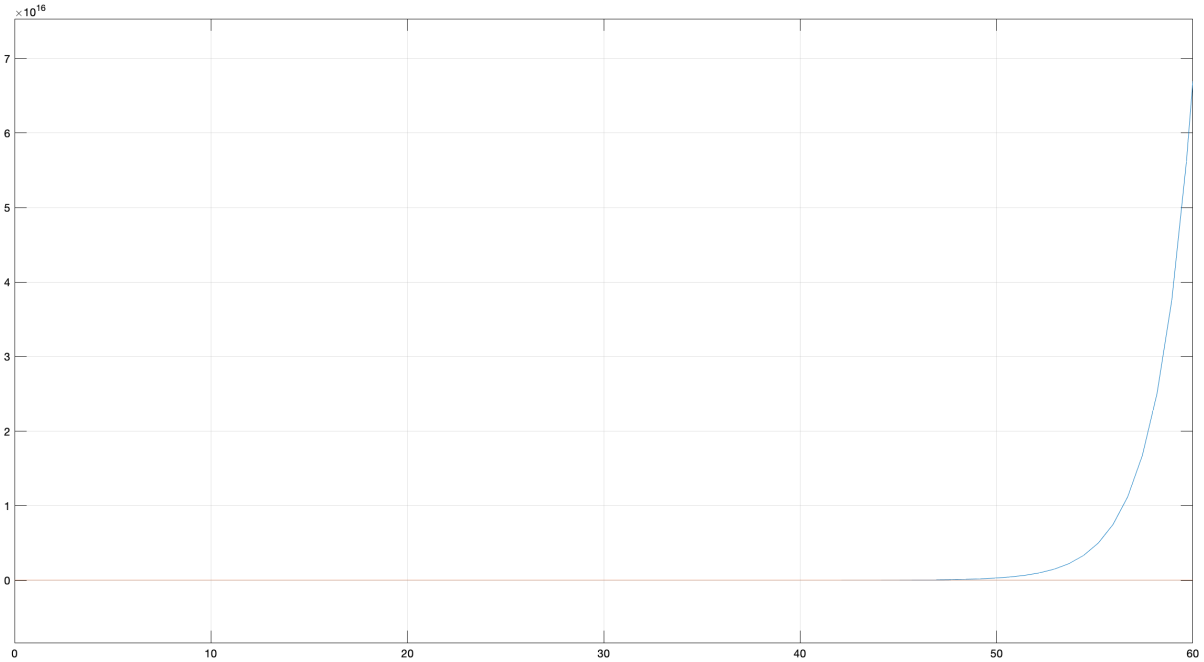
b) <u>Ruido = 0.1</u>



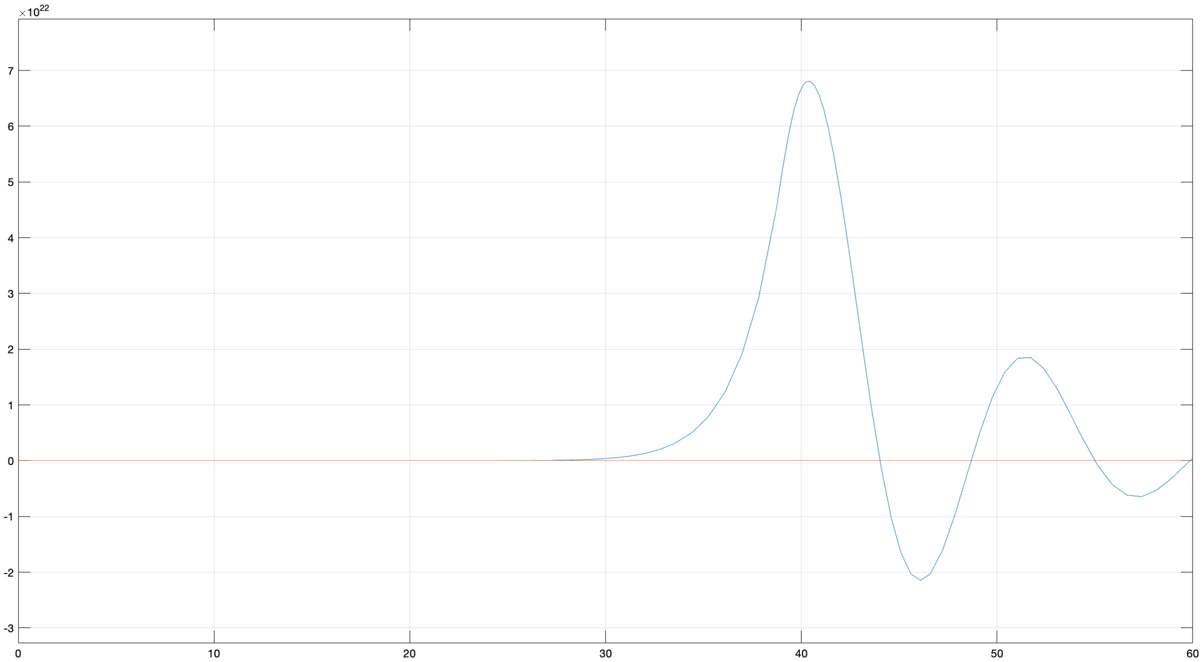
c) <u>Ruido = 0.2</u>



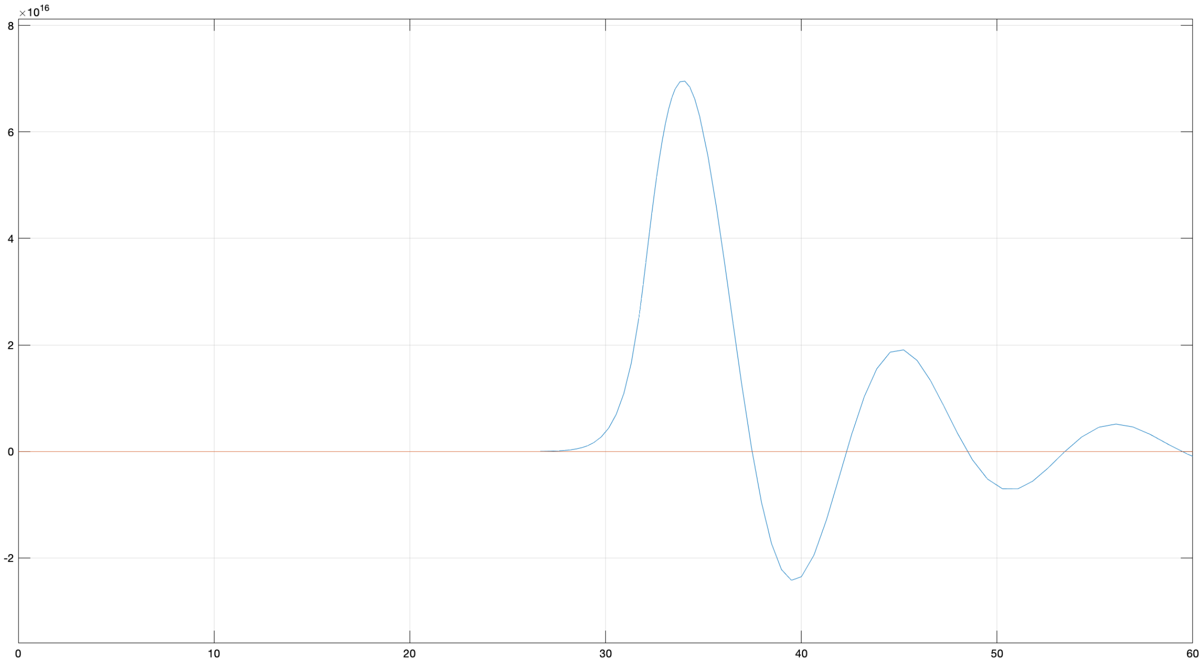
d) Ruido = 0.3



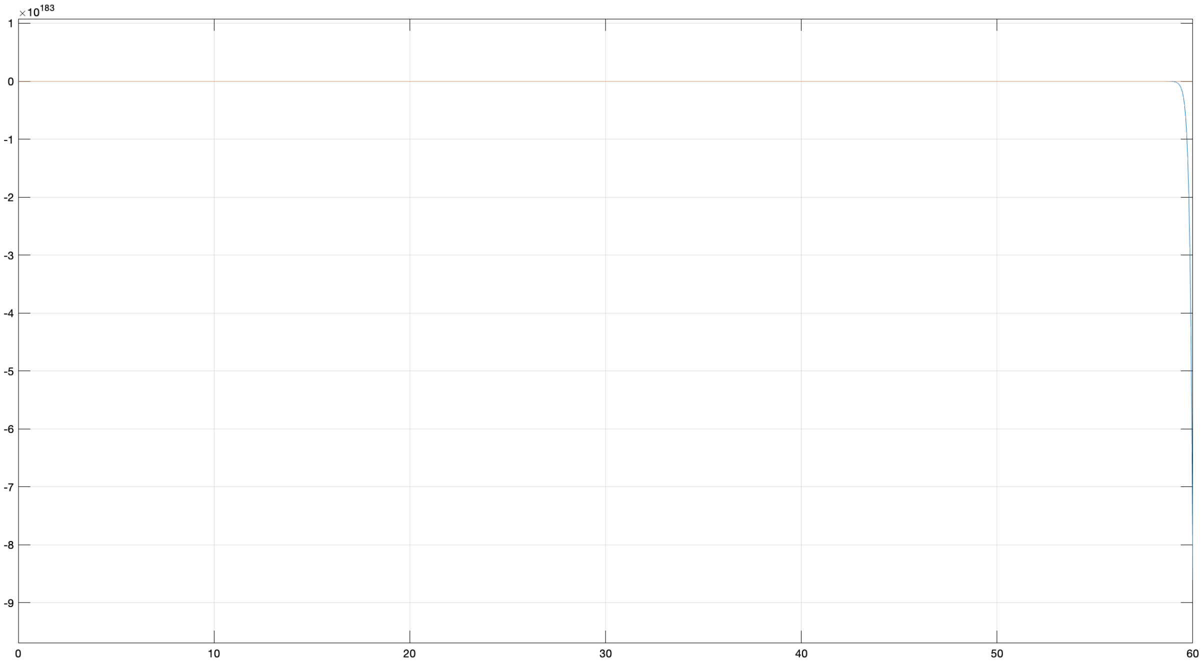
e) Ruido = 0.4



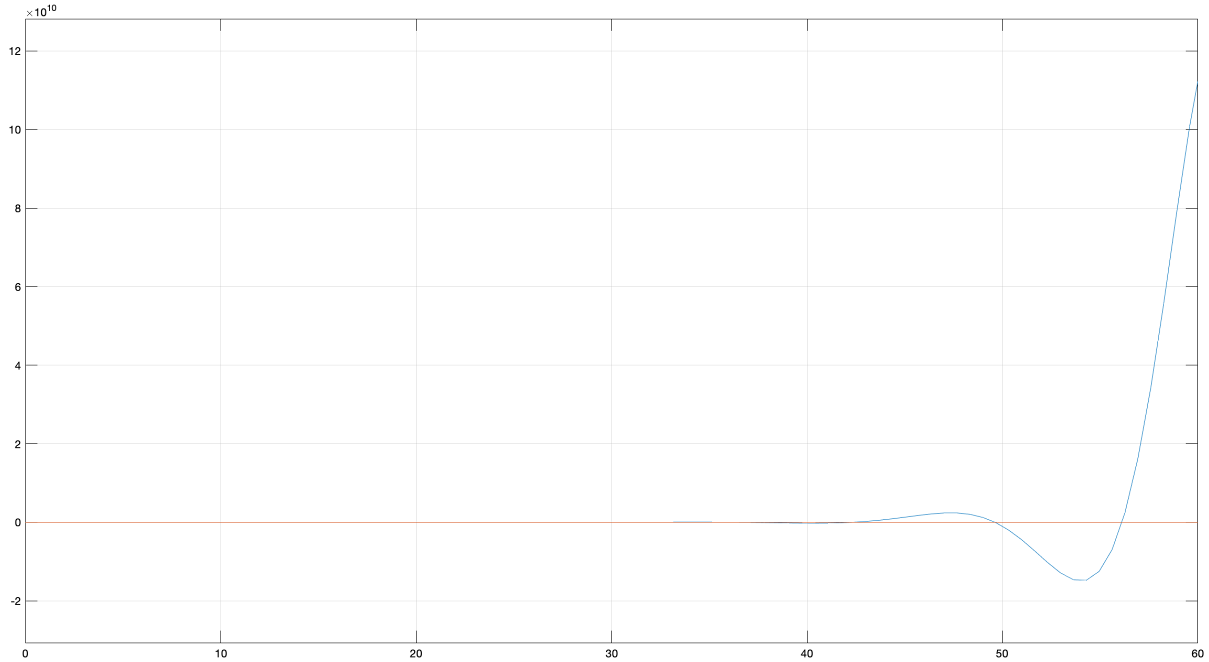
f) Ruido = 0.5



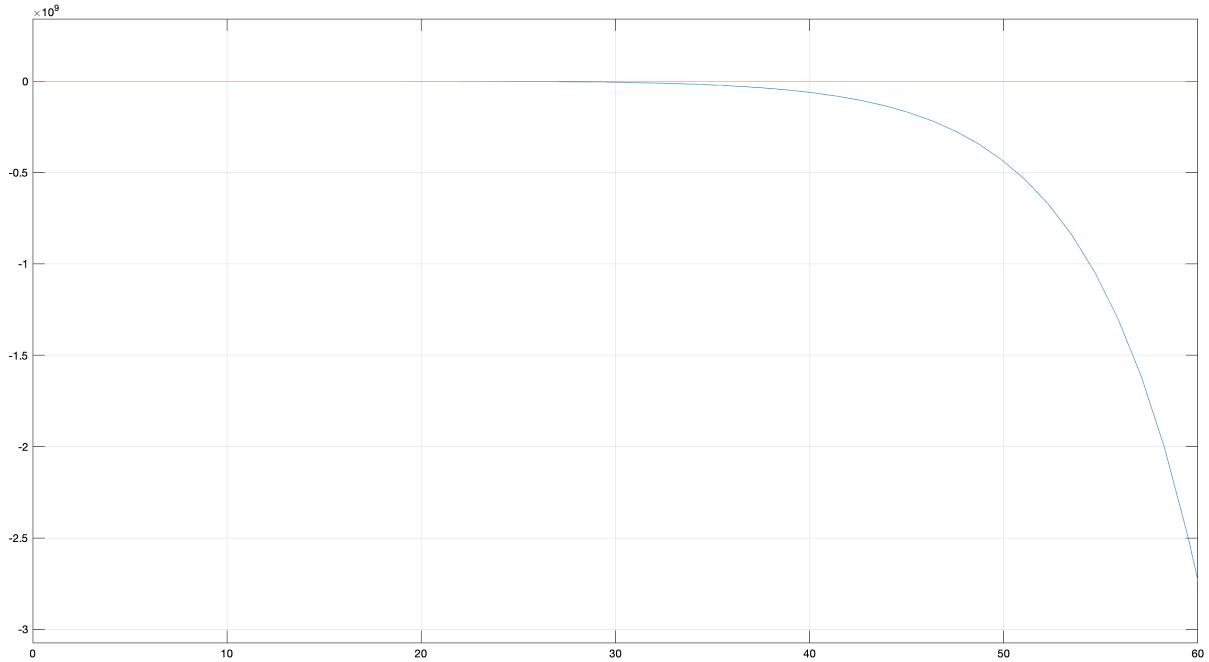
g) Ruido = 0.6



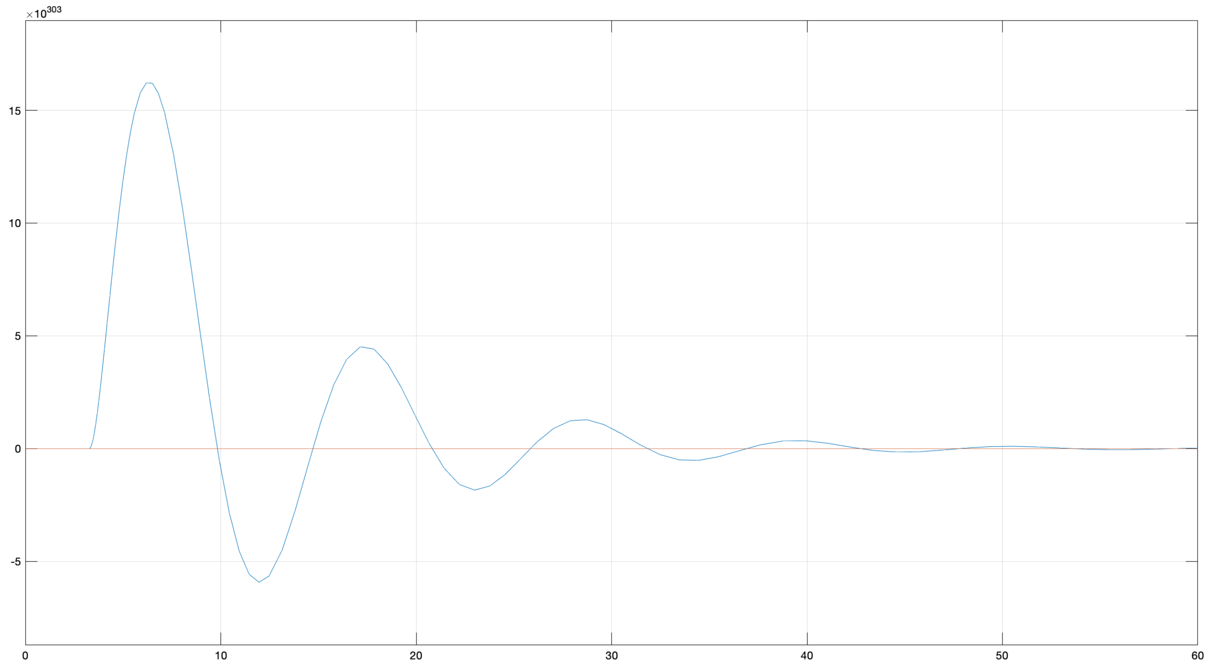
h) <u>Ruido = 0.7</u>



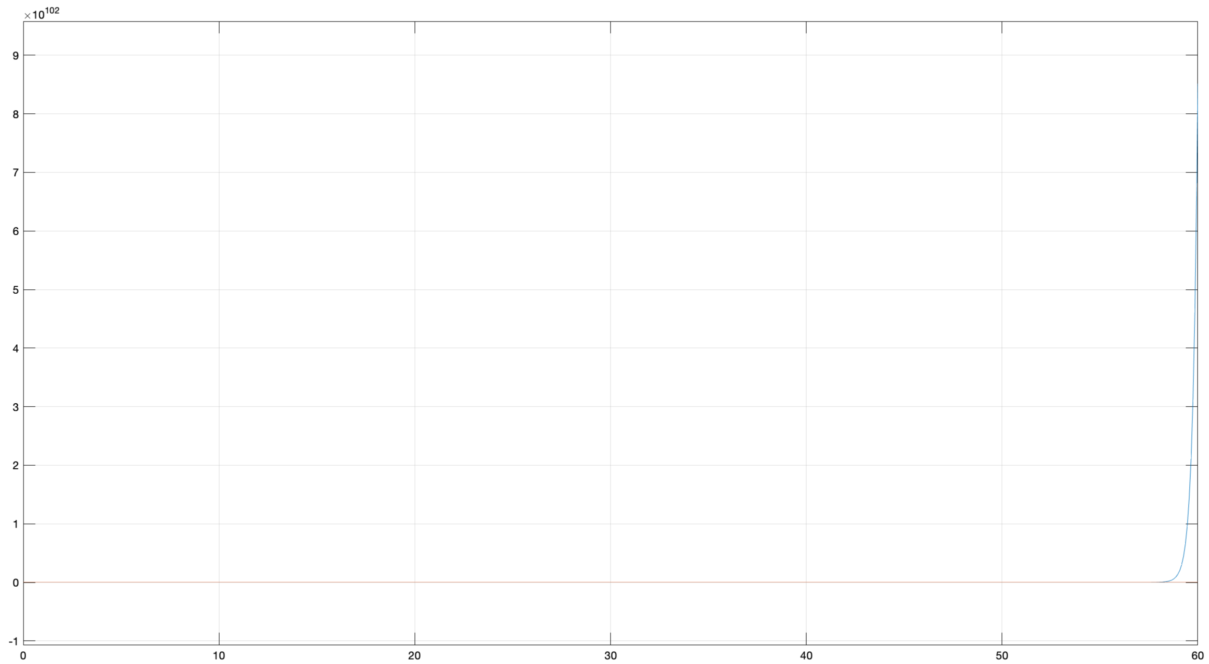
i) <u>Ruido = 0.8</u>



j) <u>Ruido = 0.9</u>



k) <u>Ruido = 1.0</u>



Como hemos visto en las gráficas del sistemas presentes arriba, al añadirle a dicho sistema un ruido gaussiano, se comporta de forma inesperada y no la gráfica deja de tener el valor esperado que indica la referencia.

Un punto que se debe destacar es que cada vez que se intentaba generar las gráficas siempre se obtenía un resultado diferente, por tanto, nos indica que el ruido induce un comportamiento aleatorio dentro del sistema.