



UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE INGENIERIA
2018 - 2^{do} Cuatrimestre

86.22 - LABORATORIO DE CONTROL AUTOMÁTICO

TP1

Integrantes-

Funes, Pablo Nicolás - 94894

Eichenbaum, Daniel - 95233
15 de octubre de 2018

Índice

1. Introducción	1
2. Ej 1 Tanques	2
2.1. Modelado de un tanque	2
2.2. S-function de un tanque	2
2.3. Pruebas de un tanque.	4
2.4. Prueba tres tanques en serie	6
3. Ej 2	8
3.1. Modelo paramétrico (Caja gris)	8
3.2. Modelo de segundo orden, Método Harriot	9
3.3. Aclaración del solver	11
4. Ej 3	12
4.1. Desarrollo del PID	12
4.2. Validación del controlador PID	14
4.3. Control del modelo del motor	15
4.4. Control del motor	16
5. Conclusión	16
6. Complementación	17

1. Introducción

En el siguiente trabajo practico, se exponen simulaciones realizadas en simulink sobre sistemas físicos elementales. Las simulaciones emplean funciones definidas por el usuario s-functions, para las cuales se necesito realizar un modelado del sistema a fin de expresar el comportamiento del sistema en las s-functions.

2. Ej 1 Tanques

En esta sección se realizara el modelado del sistema físico de un tanque para implementarlo en un sistema de mayor complejidad formado por tres tanques conectados en cascada.

2.1. Modelado de un tanque

En la figura 1 se muestra el sistema a modelar, el cual responde a las siguientes ecuaciones:

$$A \cdot \frac{dH}{dt} = Q_i - Q_o \quad (1)$$

$$Q_o = \frac{1}{R} \cdot \sqrt{H} \quad (2)$$

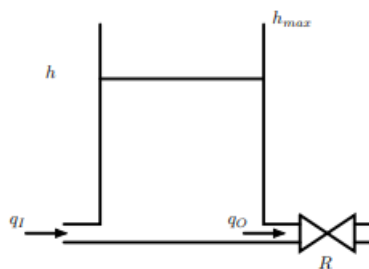


Figura 1: Sistema de un tanque

El modelo del tanque quedara expresado en función de los parámetros listados a continuación:

1. R : La resistencia de salida del caudal.
2. h_{max} : La altura máxima de liquido en el tanque.
3. A : El área del tanque.
4. h_{ini} : La altura inicial de liquido en el tanque.
5. Q_i : El caudal volumétrico de ingreso.
6. H_o : El nivel de salida.
7. H : El nivel del tanque.
8. Q_o : El caudal volumétrico de salida.

2.2. S-function de un tanque

La s-function fue especificada para responder a la interfaz o prototipo propuesto por la cátedra, similar al mostrado en la figura 2.

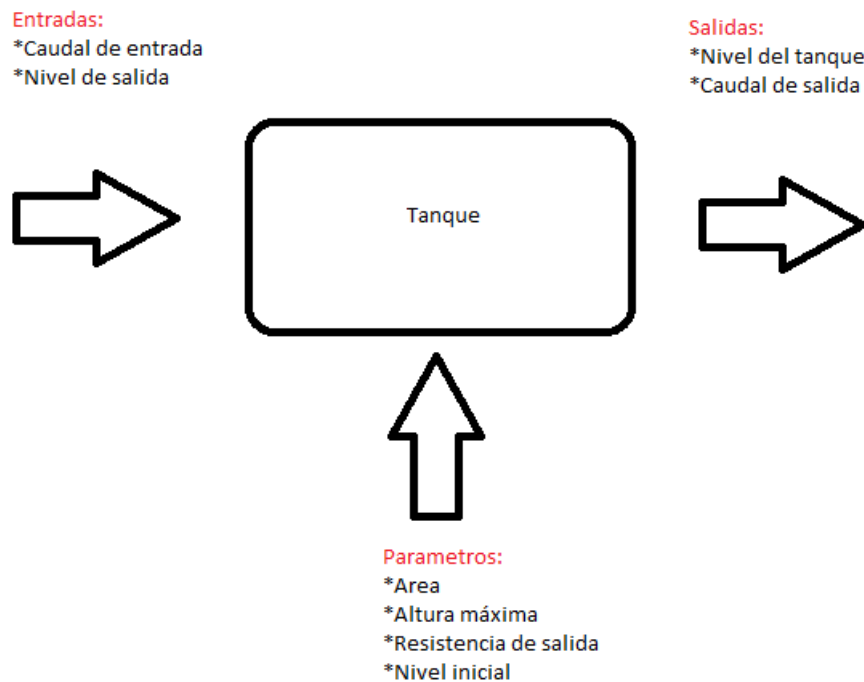


Figura 2: Esquemático del prototipo de la función

Con el prototipo propuesto se realizó la s-function con las siguientes consideraciones, la s-function realiza la serie de pasos mostrada en la figura 3. En este caso se especificó el número de entradas y salidas correspondientes a dos en ambos casos, un estado continuo y que una variable afecta directamente en la salida. En cuanto a las actualizaciones, se comenzó editando la actualización de la salida, en primer lugar se actualizó el nivel del tanque considerando sus valores límite, con este nivel se calculó el caudal de salida. Por último se realizó la actualización del estado, con la ecuación de la derivada mostrada previamente, nuevamente se realizó la limitación de la altura del tanque debido a que se decidió usar el estado sin limitación y limitarlo al necesitarlo.

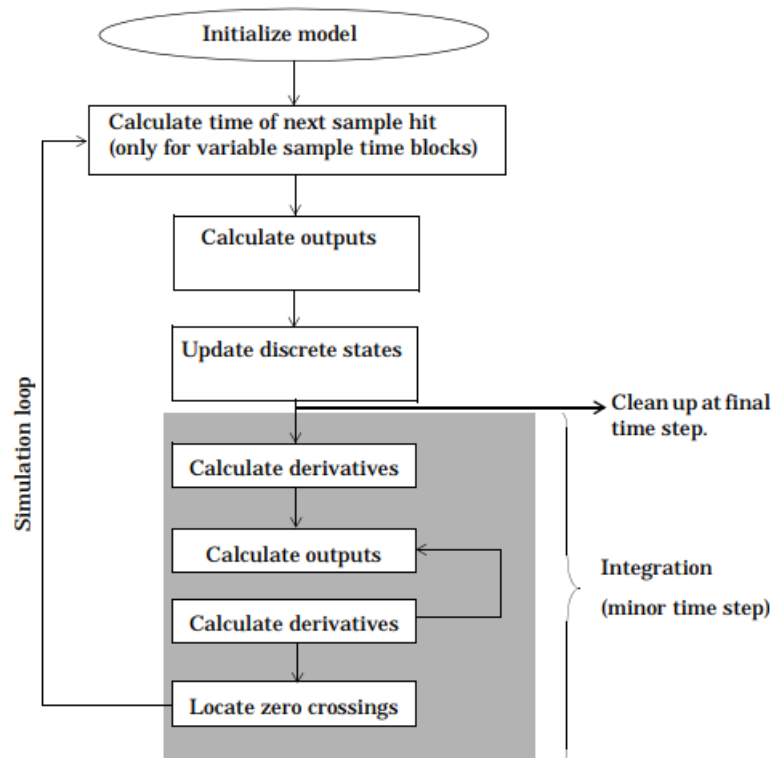


Figura 3: Etapas de una s-function.

2.3. Pruebas de un tanque.

En esta sección se muestran pruebas realizadas sobre un tanque.

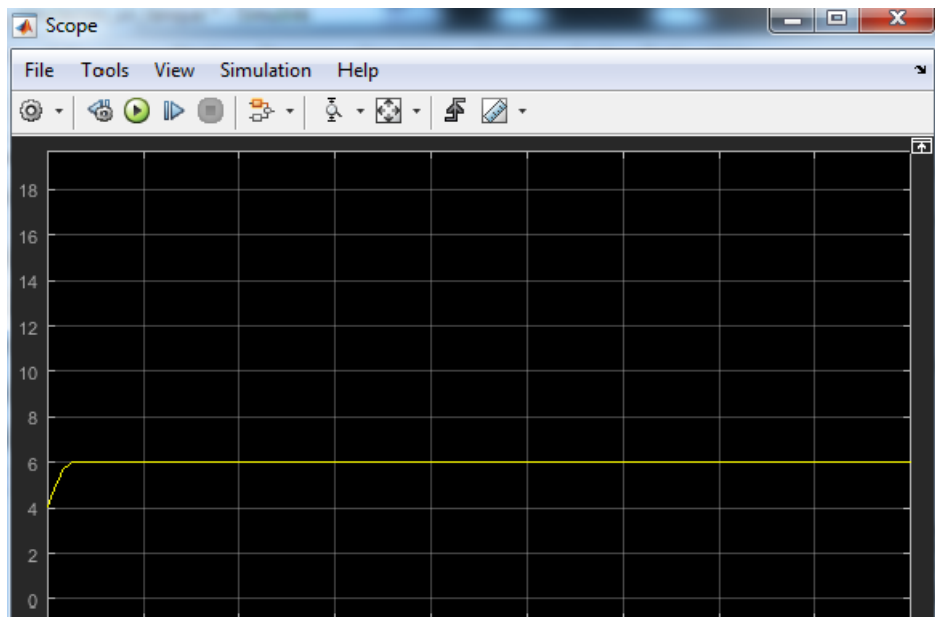


Figura 4: Simulación de tanque saturando por nivel superior al máximo.

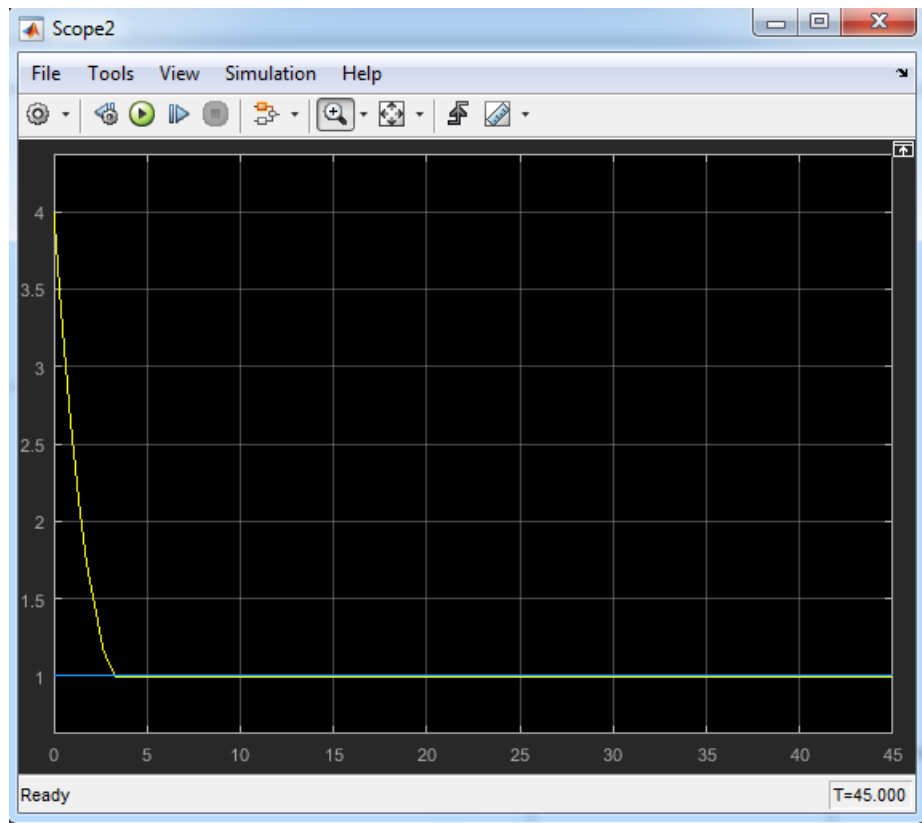


Figura 5: Simulación de tanque bajando el nivel de liquido.

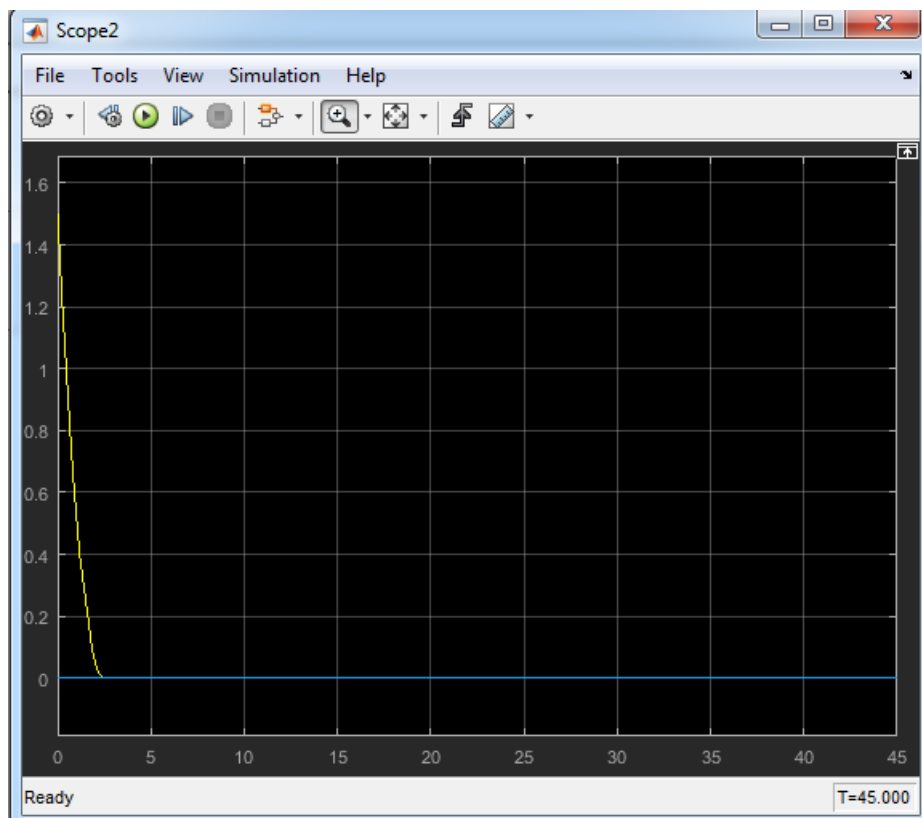


Figura 6: Simulación de tanque bajando el nivel de liquido a cero.

2.4. Prueba tres tanques en serie

A continuación, se muestra la simulación correspondiente a tres tanques iguales en serie, los tanques son idénticos con la salvedad que arrancan con niveles iniciales diferentes. En este caso se obtuvo un nivel idéntico para los tres tanques en estado estacionario, lo cual se muestra en la simulación.

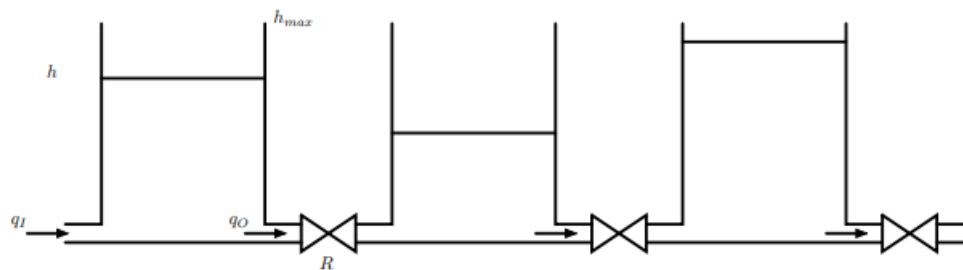


Figura 7: Sistema de tres tanques en cascada

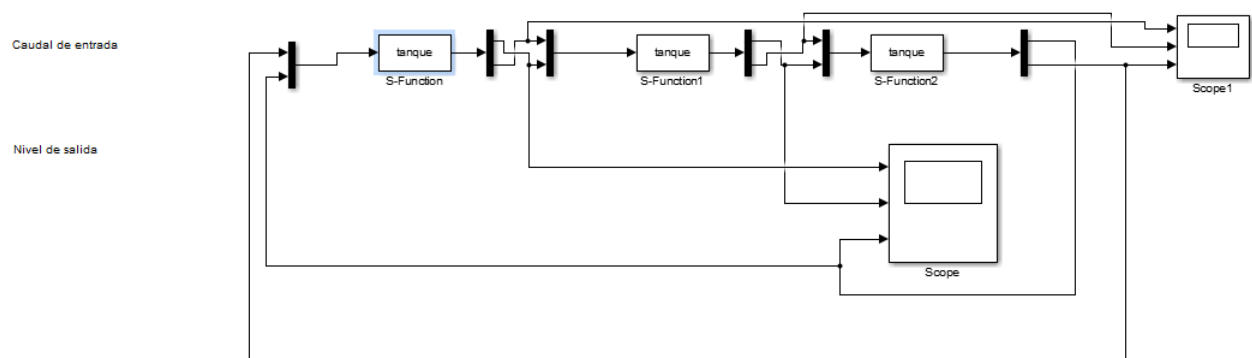


Figura 8: Sistema de tres tanques en cascada en Simulink

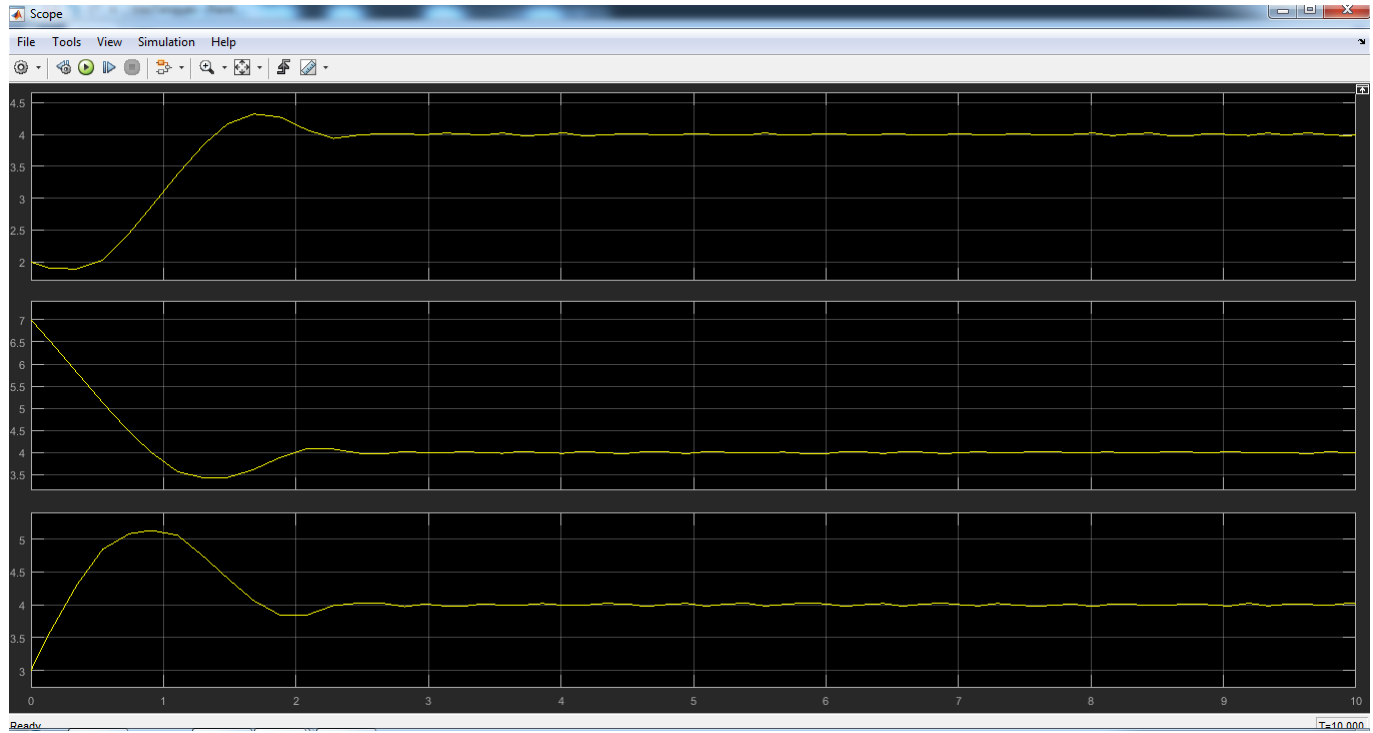


Figura 9: Simulación del nivel de los tres tanques en cascada

3. Ej 2

El ejercicio 2 consiste en la identificación de un motor de corriente continua, para ello se cuenta con la siguiente función que representa el modelo de la planta.

function [sys,x0,str,ts] = dcmotor_qenc(t,x,u,flag,encoder_ppv)

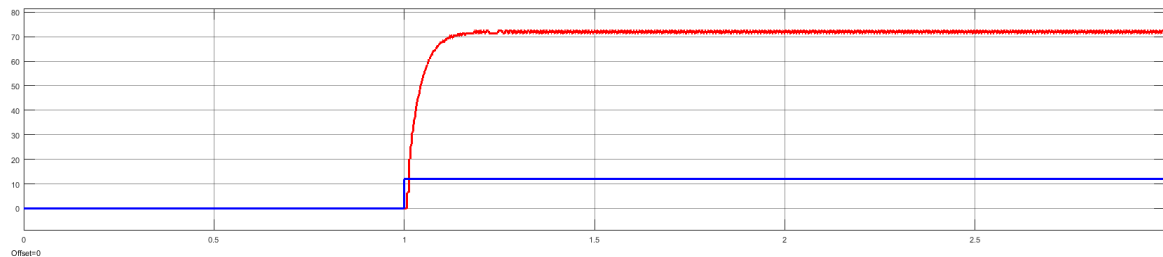


Figura 10: Respuesta al escalón del motor, La tensión 12V y la frecuencia $f = 72Hz$ están colocados bajo el mismo eje superpuesto

El motor posee las siguientes características: Entrada en tensión $u[0]$ cuyo valor nominal es de 12 V. Salidas de medición de un encoder en cuadratura ($y[0]$ e $y[1]$) montado al eje, que genera encoder_ppv pulsos por vuelta.

3.1. Modelo paramétrico (Caja gris)

Para proceder a la identificación de la planta, se propone un modelo de primer orden de la forma

$$H_1(s) = \frac{a}{s t_r + 1} e^{-t_0 s}$$

Donde se define el retardo inicial t_0 y el rise time t_r que se obtiene como el valor de tiempo al cual la señal alcanza el 63 % de su valor nominal. Se estimó $t_0 = 0$ y $a = 72/12 = 6$. Para estimar t_r , se obtuvo que la tensión nominal fué de $72Hz$, un 63 % de este valor es $f_{rc} = 45Hz$ que ocurre a un tiempo $t_r = 0,037s$ cuyos resultados se ilustran en la Fig.11

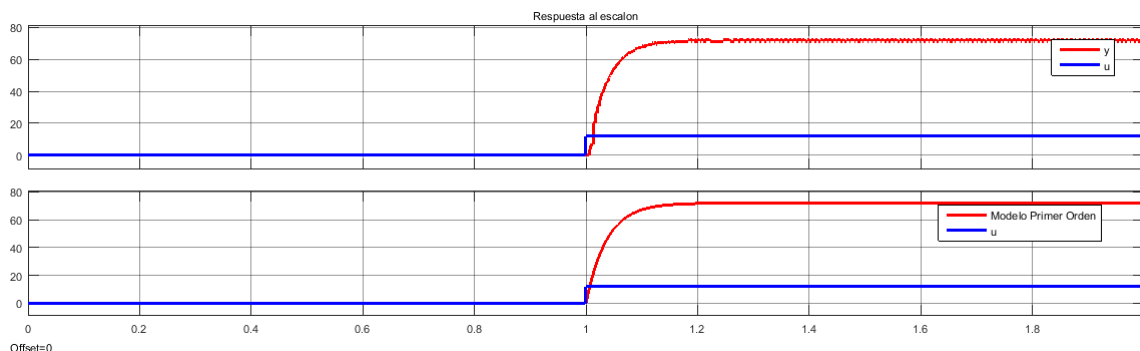


Figura 11: Intento de identificar la planta con un modelo de primer orden. Arriba Respuesta del motor, Debajo Modelo de 1º orden

Se observa, que el modelo de primer orden, puede describir bastante bien a la planta, igualmente se propondrá a continuación un modelo de orden 2.

$$H_{1er} = \frac{6}{0,037S + 1}$$

3.2. Modelo de segundo orden, Método Harriot

Se propone:

$$H(s) = \frac{K}{(\tau_1 s + 1) \cdot (\tau_2 s + 1)}$$

Sistema cuya respuesta al escalon, dió del tipo **Sobreamortiguado**, la identificación del modelo se realizará con el método de Harriot.

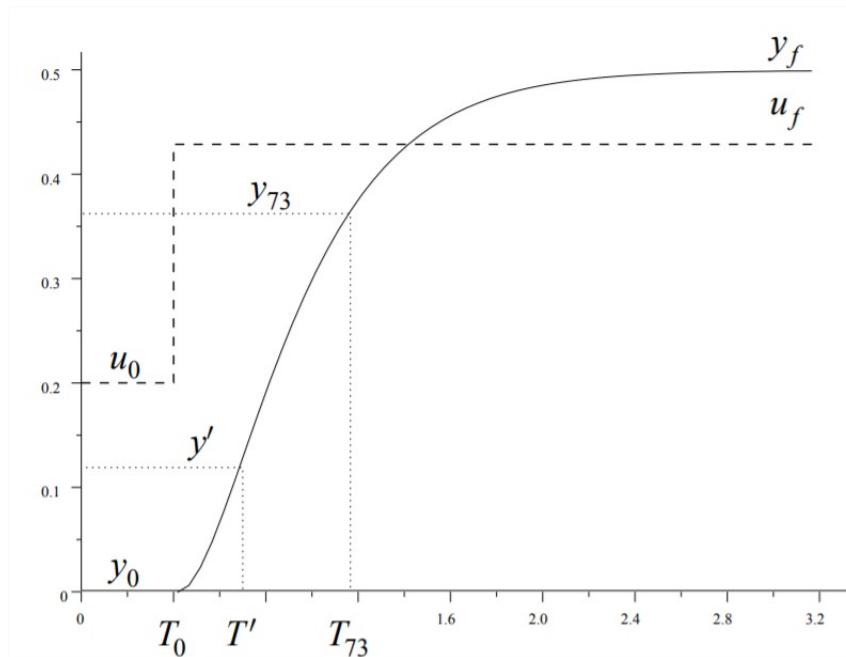


Figura 12: Método Harriot para identificación de sistema sobreamortiguado

El método Harriot consiste en el siguiente algoritmo, el objetivo será obtener los parámetros K , τ_1 y τ_2 :

1. Calcular y_{73} : Tal que $y_{73} = 0,73 \cdot y_f$
2. Obtener T_{73} : Tiempo al cual $y = y_{73}$
3. Calcular T' : $T' = T_0 + \frac{T_{73} - T_0}{2,6}$
4. Obtener y' : Señal al instante T'
5. Calcular τ_{tot} : $\tau_{tot} = \frac{T_{73} - T_0}{1,3}$
6. Calcular y_{fr} : $y_{fr} = \frac{y' - y_0}{y_f - y_0}$, se debe cumplir $y_{fr} \in [0,26; 0,39]$
7. Del gráfico Fig. 13, sabiendo y_{fr} , obtener τ_{rat}

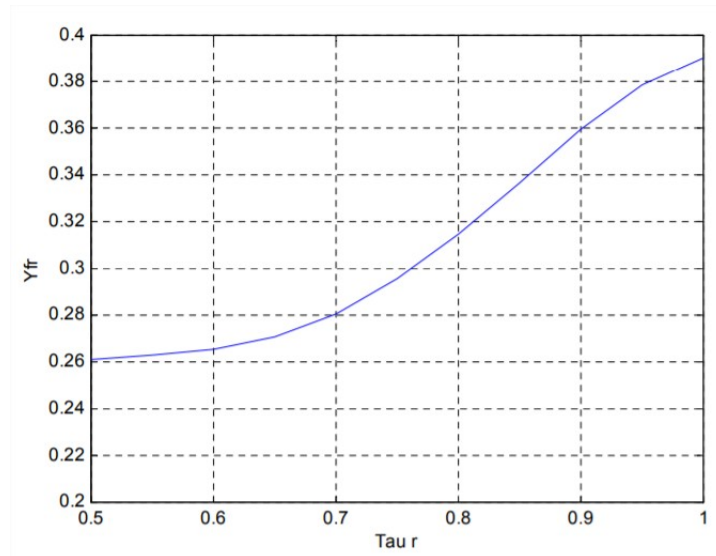


Figura 13: Relación entre y_{fr} y τ_{rat} , $y_{fr} \in [0,26; 0,39]$

8. Calcular parámetro K : $K = \frac{y_f - y_0}{u_f - u_0}$
9. Calcular Parámetro τ_1 : $\tau_1 = \tau_{rat} \cdot \tau_{tot}$
10. Calcular Parámetro τ_2 : $\tau_2 = \tau_{tot} - \tau_1$

Resultados:

1. Calcular $y_{73} = -72Hz \cdot 0,73 = -52,6Hz$
2. Obtener $T_{73} = 1,047seg$
3. Calcular $T' = 1 + \frac{0,047}{2,6} = 1,018seg$
4. Obtener $y' = -26Hz$
5. Calcular $\tau_{tot} = \frac{0,047}{1,3} = 0,036Seg$
6. Calcular $y_{fr} = \frac{-26Hz}{-72Hz} = 0,36$, se debe cumplir $y_{fr} \in [0,26; 0,39]$
7. Obtener $\tau_{rac} = 0,9$
8. Calcular parámetro $K = 72/12 = 6$
9. Calcular Parámetro $\tau_1 = 0,9 \cdot 0,036 = 0,0324$
10. Calcular Parámetro $\tau_2 = 0,036 - 0,0324 = 0,0036$

Resultando la transferencia final como:

$$H(s) = \frac{6}{(0,0324s + 1) \cdot (0,0036s + 1)}$$

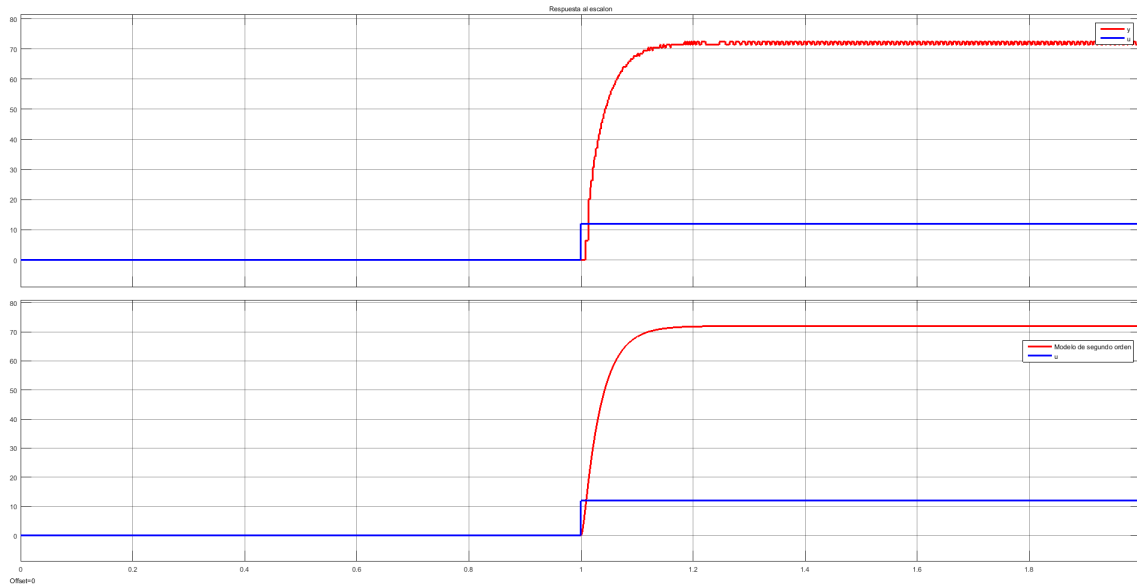


Figura 14: Identificación de la planta con un modelo de segundo orden

De la Fig. 14 el sistema parametrizado parece responder correctamente. Hasta tal vez, responde mejor al comienzo, puesto que su curvatura es mayor respecto al modelo de primer orden.

3.3. Aclaración del solver

En estas simulaciones se utilizó tiempo discreto, el funcionamiento del encoder se basa en realizar un conteo para estimar la velocidad, la velocidad va a depender del número de puntos simulados, por ejemplo si se usa un paso muy chico el encoder podría contar pocas veces aumentando el error en la medición, eligiendo un valor considerable del tiempo entre muestras se puede conseguir que el valor del conteo sea considerable y en caso de contar más o menos una unidad el error sea apto.

4. Ej 3

En este ejercicio, utilizando el modelo identificado del punto anterior, es decir:

$$H(z) = \frac{6}{(0,0324s + 1) \cdot (0,0036s + 1)}$$

Se desea controlar a lazo cerrado la velocidad del motor. El Controlador será un PID, donde una vez parametrizado, se aplicará también al modelo original desconocido `dcmotor_qenc`.

4.1. Desarrollo del PID

Las ecuaciones que describen al PID en tiempo discreto son:

Ecuación de salida u_k :

$$u_k = P_k + I_k + D_k$$

Donde los términos Proporcional, Integrativo y derivativo se calculan de la siguiente forma:

$$P_k = K_p e_k$$

$$I_{k+1} = K_p K_i h \sum_{j=0}^k e_k = I_k + K_p K_i h e_k$$

$$D_k = \frac{\gamma}{\gamma + h} D_{k-1} - \frac{K_p K_d}{\gamma + h} (y_k - y_{k-1})$$

Donde e_k representa la señal de entrada, es decir la señal error:

$$e_k = y_k - r_k$$

OBS. Notar que el término derivativo no responde a la derivada del error, más bien a la derivada de la salida. Además la derivada del error estaría pasando por un filtro pasabajos para evitar inestabilidades del sistema, puesto que el derivador ideal tiene un cero en el origen.

A continuación, se incluye el algoritmo del PID implementado:

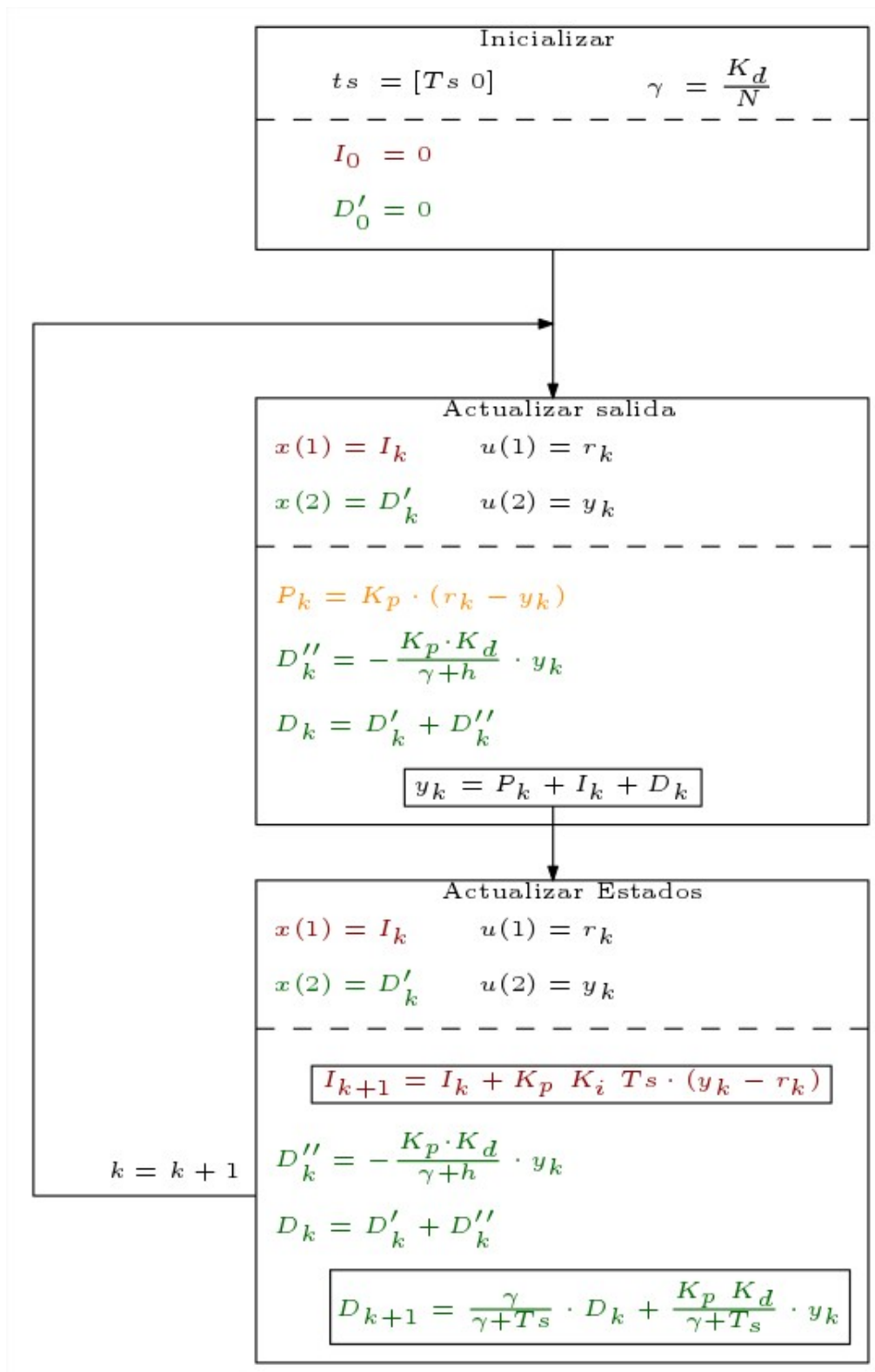


Figura 15: Algoritmo del PID

4.2. Validación del controlador PID

En la Fig. 16 se grafica la respuesta al impulso para los parámetros

$$\begin{cases} K_p = 1 \\ K_d = 0,01 \\ K_i = 10 \\ N = 10 \end{cases}$$

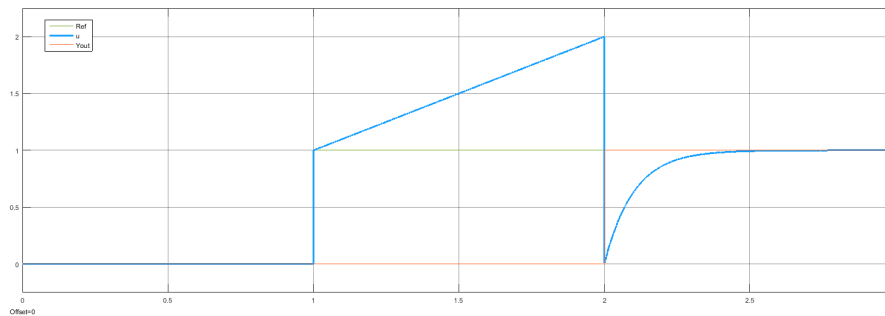


Figura 16: Validación del controlador PID

Las señales de la Fig. 16 corresponde a la señal de referencia, un escalón que actúa en $r = 1$ si $t \geq 1$ y la señal de salida que también es un escalón $y_{out} = 1$ si $t \geq 2$. Que es correcto ya que compone los tres efectos, Proporcional (escalón entre 1 y 2seg), Integrativo (Rampa entre 1 y 2), Derivativo (Delta negativo ya que como se dijo, se realizará la derivada de la salida y no del error, luego como pasa por un filtro tiene una respuesta exponencial).

Se observa que para $t \geq 2$ el error $y_{out} - ref = 0$ se hace nulo.

El otro ensayo considerado, corresponde a la saturación, en la Fig. 17 se muestra el PID configurado con un umbral máximo de 1.5V.

Además se incluyó el efecto antiWindUp que responderá en la conservación del término integrativo, ni bien el sistema sature; Si este efecto no estuviese, cuando la señal de salida superase el umbral, dicho término continuaría acumulando error, pese a no poder responder frente a él; y cuando desee responder habrá tanto error, que para compensarlo se requerirá que el sistema actúe de forma contraria hasta lograr el mismo error con signo negativo; Se obtendría un sistema oscilatorio.

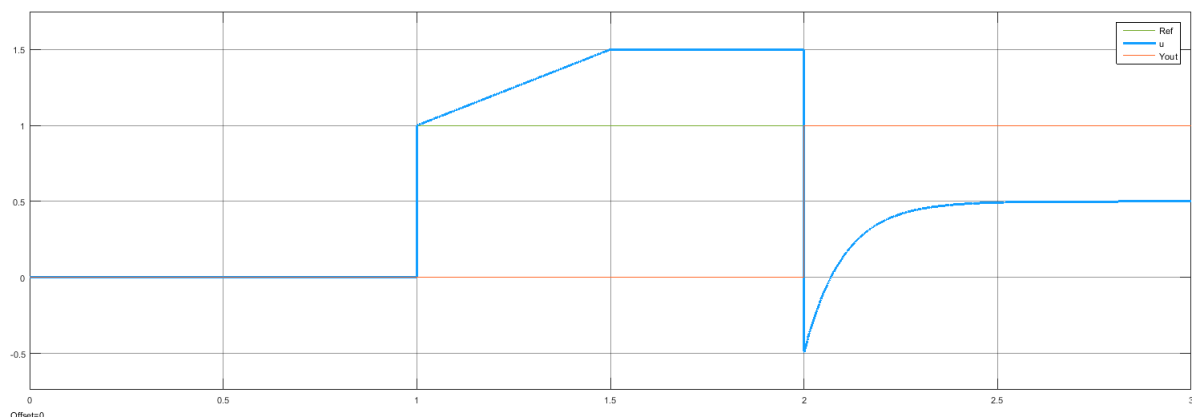


Figura 17: Ensayo anti Wind Up para el controlador PID, el límite de saturación superior es 1.5V

Se observa que PID implementa antiWindUp, ya que al sistema le faltan 0.5 para que la salida sea 1. Esos 0.5 corresponde a la zona donde no pudo integrar; y el sistema no sabe que estamos trabajando a lazo abierto para compensarlo.

4.3. Control del modelo del motor

A continuación, se introducirá el modelo del motor obtenido del inciso anterior y se lo controlará a una velocidad estable de $f_{ref} = 45Hz$

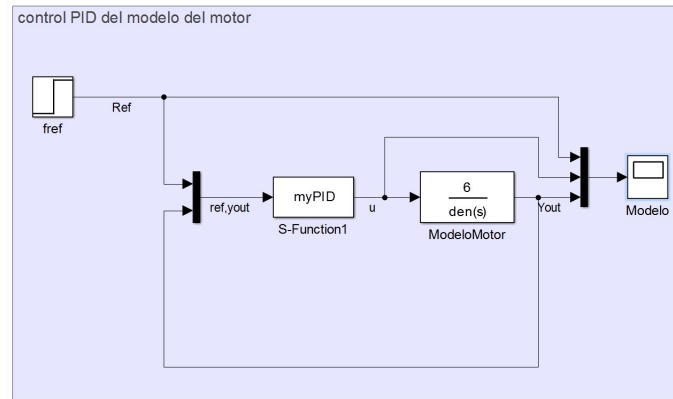


Figura 18: Esquema de simulink para el control del modelo del motor

El ajuste del PID se realizó de la siguiente forma, se ajusto primero $K_p = 1$, $K_d = 0$, $K_i = 0$ de modo que el sistema responda sin oscilar a su estado estacionario.

Luego se introdujo $K_i = 10$ de modo de alcanzar la referencia sin error estacionario.

Finalmente se modificó $K_d = 0,01$ de modo que se redujo el tiempo al cual el sistema convergía a su estado estacionario.

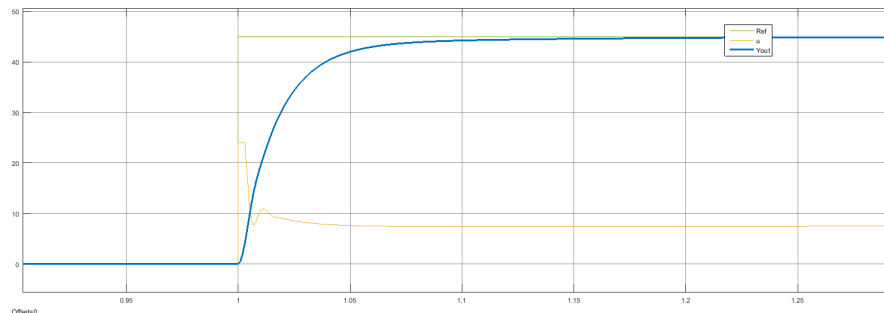


Figura 19: Control de velocidad para el modelo del motor, dada una saturación de 24V

La respuesta de la Fig. 19 corresponde a un escalón de $45Hz$ que se desea seguir, la saturación superior es de 24V ya que el motor no podrá recibir mayor tensión.

Se obtuvo la respuesta deseada.

4.4. Control del motor

Ahora, se procederá a simular el motor original. Este tiene la dificultad de que el encoder procesa en forma lenta la velocidad del motor y que la cantidad de pulsos por vueltas en el encoder es fundamental para estimar su velocidad.

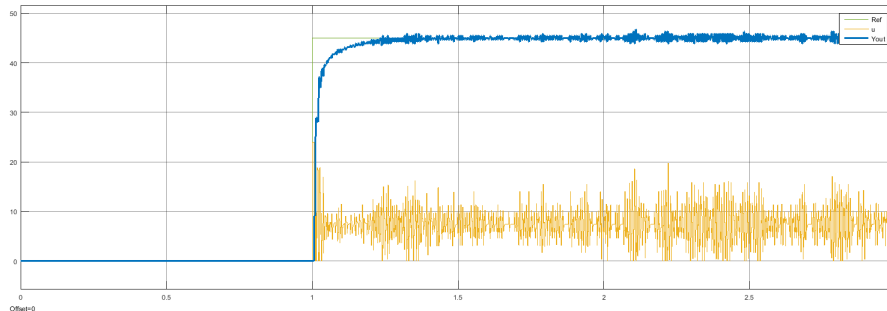


Figura 20: Control de velocidad para del motor, dada una saturación de 24V para el PID

De entrada en la Fig. 20 se observa que el sistema no se comporta como el modelo ideal. Puesto que los datos vienen en forma de paquetes de pulsos y el efecto de la derivada no responde de forma suave.

5. Conclusión

Por un lado, se logró implementar correctamente el modelo de un arreglo de tanques concatenados, implementar un modelo desconocido de un motor e identificar al mismo mediante dos métodos (constante de tiempo para 1^o orden y método Harrot para el de 2^o orden). Finalmente se diseñó un controlado PID discreto capaz de controlar al motor identificado hacia un valor de referencia. Se utilizaron funciones definidas por el usuario en simulink, teniendo que considerar en cada caso el tipo de sistema, los estados requeridos y por ultimo el tipo de integración a realizar en los mismos.

6. Complementación

En esta sección se realiza la identificación de manera alternativa a la vista en el informe, en particular se utiliza el identification tool de matlab, obteniendo el modelo equivalente con arx por ejemplo.

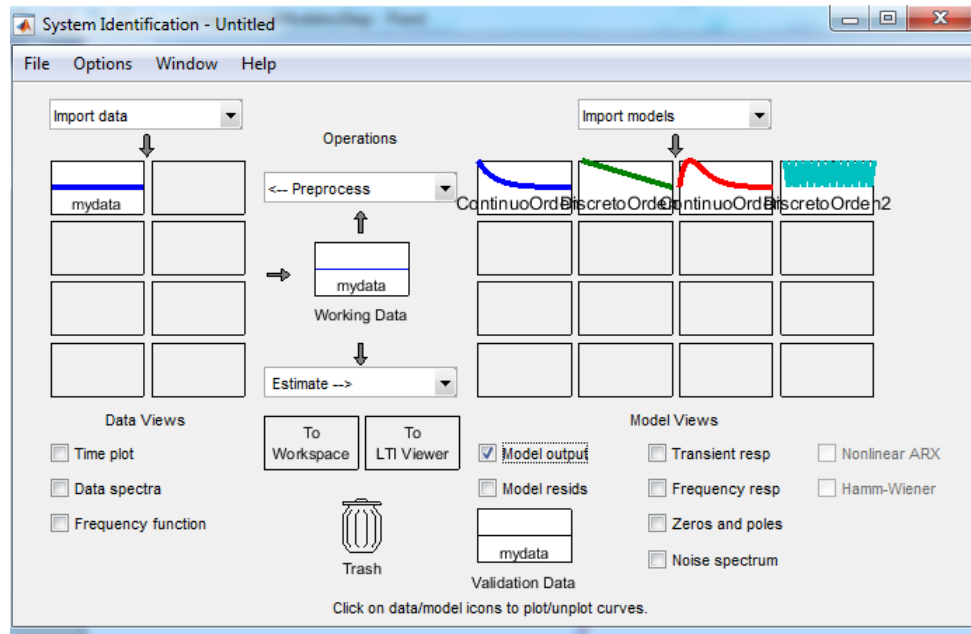


Figura 21: Identification tool

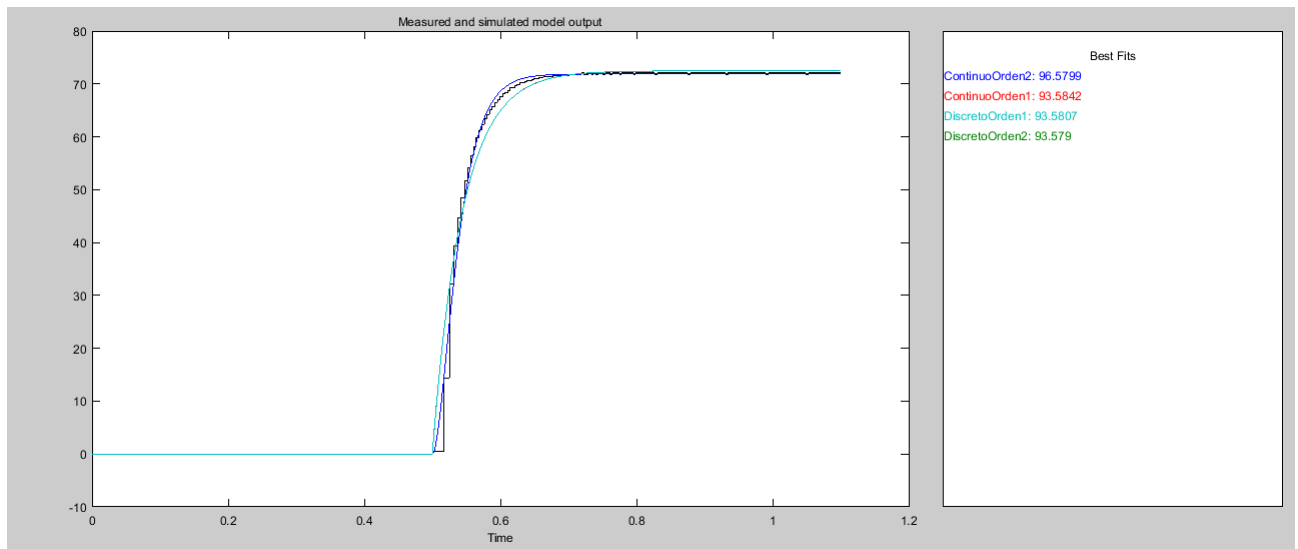


Figura 22: Identificación utilizando modelos continuos y discretos, de primer y segundo orden.

En la figura 22 se muestran los modelos para primer orden y segundo orden, para los casos discretos y continuos con su porcentaje de fit. De esta simulación por ejemplo se obtuvo el modelo discreto de segundo orden.

$$T(z) = \frac{0,002729871894989}{1 - 8,605285413471888 \cdot 10^{-05} z^{-1} - 0,999461913536859 \cdot z^{-2}} \quad (3)$$