

UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE INGENIERÍA MECÁNICA



TESIS

“IMPLEMENTACIÓN DEL SISTEMA DE CONTROL PROPORCIONAL INTEGRAL DERIVATIVO PARA CONTROLAR EL NIVEL DE AGUA EN UN DEPÓSITO USANDO TARJETA DE ADQUISICIÓN DE DATOS USB-6008”

**PARA OBTENER EL TÍTULO PROFESIONAL DE:
INGENIERO MECATRONICO**

**ELABORADO POR:
RENZO MALCOLM GARCIA PICKMANN,**

ASESOR

MSc. Ing. RICARDO RODRÍGUEZ BUSTINZA

LIMA – PERÚ

2018

AGRADECIMIENTOS

A mis padres, por haberme brindado la gran oportunidad de estudiar en tan prestigiosa casa de estudios, por haberme motivado a la conclusión de este proyecto desde el principio, por los consejos brindados en base a sus experiencias obtenidas durante la vida; seguiremos cumpliendo metas juntos.

Al Msc. Ing. Ricardo Rodríguez Bustinza, por su paciencia, conocimientos brindados y seguimiento durante el desarrollo de este proyecto.

A mis familiares y amigos, porque siempre tuvieron presente el proyecto que estaba realizando y nunca dejaron de hacérmelo recordar debido a la gran importancia que éste tenía en mi vida profesional. Gracias.

RESUMEN

Esta investigación se ubica en la línea de las Ciencias de Ingeniería, desarrolla un procedimiento para el modelado, diseño e implementación experimental de un sistema de control PID para controlar el nivel de agua en el proceso tanque con agua. El algoritmo de control se implementa en la plataforma LabVIEW, donde se realiza el procesamiento de datos y la toma de acciones para el control de la variable de nivel. Los estudios de simulación y experimentación realizados demostraron el buen rendimiento de este controlador, el cual fue probado para distintas consignas, siendo la ley de control capaz de hacer que el sistema de control tenga mínimo sobre impulso y error en estado estacionario nulo.

Tabla de contenido

CÁPITULO 1

INTRODUCCIÓN

1.1 Antecedentes	1
1.2 Planteamiento del problema	2
1.3 Objetivos	3
1.4 Alcances y limitaciones	3
1.5 Estado del arte	4

CÁPITULO 2

MODELAMIENTO DEL PROCESO TANQUE CON AGUA

2.1 Variable de ecuaciones del estado del proceso.....	6
2.2 Modelo no lineal del sistema tanque con agua	7
2.3 Linealización del modelo	10
2.4 Prueba del sistema en lazo abierto usando Matlab	11

CÁPITULO 3

IDENTIFICACIÓN DE LA PLANTA

3.1 Identificación del modelo	13
3.2 Determinación del rango de trabajo.....	14
3.3 Identificación no paramétrica.....	21
3.3.1 Aproximación gráfica	21
3.3.2 Modelo de 1er Orden usando el Toolbox de Identificación de Sistemas	22
3.3.3 Modelo de 2do Orden usando el Toolbox de Identificación de Sistemas.....	25
3.4 Identificación paramétrica.....	27
3.4.1 Estructura ARX.....	28
3.4.2 Estructura ARMAX.....	29

3.5 Algoritmo de los mínimos cuadrados con factor de olvido	30
3.5.1 Modelo ARX de Orden 1	30
3.5.2 Modelo ARMAX de Orden 1	31
3.6 Implementación del algoritmo en el proceso.....	33
3.6.1 Aproximación gráfica.....	33
3.6.2 Modelo de 1er Orden obtenido con el Toolbox de Identificación	33
3.6.3 Modelo de 2do Orden obtenido con el Toolbox de Identificación.....	34
3.6.4 Modelo ARX de Orden 1	35
3.6.5 Modelo ARMAX de Orden 1	35
3.7 Validación de los modelos obtenidos.....	36
3.7.1 Aproximación gráfica.....	36
3.7.2 Modelo de 1er Orden obtenido con el Toolbox de Identificación	38
3.7.3 Modelo de 2do Orden obtenido con el Toolbox de Identificación.....	39
3.7.4 Modelo ARX de Orden 1	40
3.7.5 Modelo ARMAX de Orden 1	41
3.8 Comparación de modelos.....	42

CÁPITULO 4

DISEÑO DEL SISTEMA DE CONTROL

4.1 El controlador PID	43
4.2 Sintonización por Ziegler-Nichols	44
4.3 Método analítico de sintonía por ubicación de polos	45
4.4 Implementación del algoritmo de control	51
4.5 Simulación de la planta a lazo cerrado	51

CÁPITULO 5

IMPLEMENTACIÓN DEL SISTEMA DE CONTROL

5.1 Hardware del sistema.....	54
5.2 Desarrollo del software.....	57
5.3 Pruebas y resultados.....	60

CONCLUSIONES.....	66
--------------------------	-----------

RECOMENDACIONES.....	66
-----------------------------	-----------

REFERENCIAS BIBLIOGRÁFICAS.....	67
--	-----------

APÉNDICE

A. Especificaciones técnicas de la DAQ USB-6008 de National Instruments	69
B. Conceptos básicos – Estimación utilizando mínimos cuadrados recursivos con factor de olvido	73
C. Implementación del algoritmo recursivo ARX.....	81
D. Implementación del algoritmo recursivo ARMAX	87

CAPITULO 1

INTRODUCCION

1.1. **Antecedentes**

En esta sección se presentan las reseñas de algunas investigaciones que sirvieron como apoyo para la realización de este trabajo.

En el año 2007, Parra, A. [1] diseñó e implementó controladores PID para controlar una gran variedad de procesos industriales. Encontramos el modelado de un proceso denominado tanque de agua, un proceso multivariable que posee dos entradas de control: el flujo de entrada al tanque y el calor suministrado al agua mediante una resistencia eléctrica, y dos variables a controlar: el nivel del tanque y la temperatura de salida del agua calentada. Además desarrolla la implementación en tiempo real, lo que sienta las bases para este proyecto de tesis.

En el 2008, Parra, A. y Rojas, A. [2] desarrollan el diseño de un sistema de control PID MIMO para controlar nivel y temperatura en el proceso tanque con agua con una tarjeta de adquisición de datos (DAQ); la planta es descrita por

ecuaciones diferenciales no lineales interconectadas, para las cuales se aplica una técnica de linealización para obtener una descripción lineal MIMO.

En el año 2011, Aquize R. [3] desarrolla la técnica de control backstepping para la planta tanque con agua con un DAQ, sin embargo esta técnica es superior en cuanto a desempeño que el controlador PID MIMO desarrollado en [2].

1.2. **Planteamiento del problema**

En los procesos industriales, los líquidos son generalmente bombeados y almacenados en tanques para realizar tratamientos químicos o mezclas. Por ejemplo, en una refinería de petróleo, un desbordamiento de este líquido es peligroso y costoso, un depósito vacío da lugar a que las bombas o el sistema de drenaje operen sin líquidos; otro caso a considerar es un sistema de control de agua de alimentación de un reactor, el cual debe mantener un adecuado nivel de agua en la vasija del reactor para mantener un balance de refrigerante, mantener cubierto de agua el núcleo del reactor y mantener una eficiencia óptima de los separadores de vapor, si no se cumple con los requerimientos, el sistema incurre en daños.

Un nivel inadecuado provoca fallas en los productos y altos costos, es por ello que un sistema de control de nivel preciso de líquido es vital, especialmente en procesos industriales donde los inventarios, producción por lotes y la eficiencia del proceso son medidas críticas.

Para lidiar con los problemas antes mencionados, el controlador PID es propuesto e implementado en el proceso tanque con agua. Éste presenta una

óptima solución en mediciones de nivel para llevar el líquido a un nivel específico asignado por el operario.

1.3. **Objetivos**

El objetivo principal de este trabajo de tesis es la implementación de un sistema de control PID a fin de mantener el nivel deseado de agua en un tanque, cuya salida varía aleatoriamente, obteniendo un error en estado estable nulo. Asimismo, el controlador PID de instrumentación virtual es implementado usando el software LabVIEW desarrollado por National Instruments.

1.4. **Alcances y limitaciones**

El proyecto tiene en cuenta el diseño del sistema de control, desde la identificación del modelo, ajustes, sintonización y métodos utilizados en el control de nivel de agua deseado. Con el objetivo de diseñar un verdadero sistema de control de nivel de agua a escala, este proyecto es desarrollado con:

- i. Sensor ultrasónico Allen-Bradley 873M, para medir el nivel de agua en el tanque.
- ii. Válvula de control proporcional Danfoss EV260B, para controlar el flujo de agua entrante en el tanque.

Con respecto a las limitaciones, el controlador PID calza perfecto para el proceso tanque con agua, ya que se tiene un sistema SISO, que puede ser linealizado. Sin embargo, en el control de sistemas con mayor complejidad, el desempeño del controlador PID es muy discutido actualmente, para ello se

utilizan técnicas de control avanzado como: FUZZY-PID, ADAPTATIVO-PID, entre otros.

1.5. **Estado del arte**

Se han realizado numerosas investigaciones sobre el control de nivel en tanque tanto en nuestro país como en el mundo. Este apartado se enfoca fundamentalmente en los estudios y experiencias actuales en el ámbito internacional.

Técnicas de control avanzado están siendo implementadas en el control de nivel de líquidos en tanques, por ejemplo, en [6] se desarrolla e implementa un controlador basado en lógica difusa usando una interfaz gráfica en Matlab. Los resultados obtenidos de la simulación son comparados con el controlador clásico PID y son satisfactorios. La particularidad de esta planta es que también se compone por una válvula de control como el actuador, igual al proyecto que yo realizo, mientras que otras investigaciones poseen una motobomba como actuador; de la misma forma en [7], se realiza el diseño de un controlador difuso para el control de nivel en una planta que consta de dos tanques acoplados y como actuador una motobomba. Los estudios de simulación son desarrollados usando Simulink, una poderosa herramienta de Matlab; en [8], el controlador usado es un PLC S7-300 conectado a un HMI SIMATIC WinCC (ampliamente usado en los procesos de producción industrial), que a su vez está conectado a un servidor OPC que envía data a Matlab, donde algoritmos de control complejos pueden ser ejecutados a través de la investigación en el modelo y la simulación del sistema de control. Finalmente, el artículo propone un sistema de

control difuso en tiempo real basado en la tecnología de comunicación OPC, cuya efectividad es verificada.

Las nuevas investigaciones ya no sólo recaen sobre tanques lineales (rectangulares, cilíndricos), sino en tanques de tipo no lineal (cónicos, esféricos, etc.); en [9] se ha diseñado e implementado un controlador para controlar el nivel en un tanque cónico, este proceso no puede ser efectivamente controlado por un controlador lineal PID convencional, sino que un controlador robusto por modelo interno es necesario. La implementación en tiempo real se hizo en Simulink usando Matlab.

La tendencia se encuentra en usar técnicas de control avanzado que permitan diseñar controladores que son implementados usando el software Labview, como ocurre en [10], que presenta un control de nivel en un solo tanque usando dos tipos de controladores; PID y difuso; se pretende utilizar para la adquisición de datos microprocesadores Arduino y diseñar en Labview, la interfaz hombre – máquina (HMI) para así tener una implementación del controlador en tiempo real.

CAPITULO 2

MODELAMIENTO DEL PROCESO TANQUE DE AGUA

En el estudio de los sistemas de control, el modelamiento de un sistema dinámico es decisivo. Un modelo matemático de un sistema dinámico se define como un conjunto de ecuaciones que representan la dinámica del sistema con precisión o, al menos, bastante bien.

Un sistema puede representarse de diversas formas, por lo que puede tener muchos modelos matemáticos, dependiendo de cada perspectiva. Por ejemplo, en problemas de control óptimo, es provechoso usar representaciones en el espacio de estados. En cambio, para los análisis de respuesta transitoria o de la respuesta en frecuencia de sistemas lineales con una entrada y una salida invariantes en el tiempo, la representación mediante la función de transferencia resulta más conveniente que cualquier otro modelo.

2.1. Variables y ecuaciones de estado del proceso

El comportamiento de un sistema dinámico es descrito en su totalidad por las variables de estado. La tabla 2.1 describe las variables y los valores de los parámetros del proceso tanque de agua mostrado en la figura 2.1.

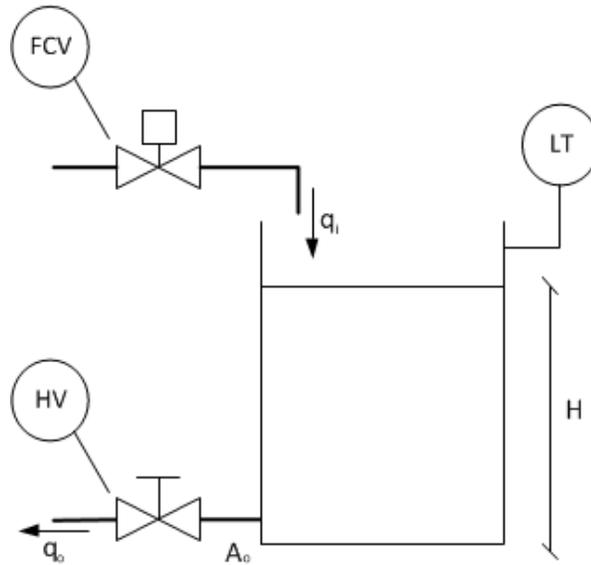
Tabla 2.1 Variables y parámetros valorados del sistema tanque con agua

Símbolo	Descripción	Valor	Unid.
d_A	Diámetro del tanque	0.31	m
\hat{A}	Sección circular del tanque	0.0755	m^2
H	Nivel de agua en estado estacionario	0.30	m
h	Nivel de agua		m
Q	Flujo de agua en estado estacionario	0.38	m^3/h
q_i	Flujo de agua de entrada al tanque		m^3/h
q_o	Flujo de agua de salida del tanque		m^3/h
R_h	Resistencia hidráulica del tanque: $R_h = \bar{H}/Q$	2843	s/m^2
g	Constante gravitacional	9.81	m/s^2
d_o	Diámetro del orificio de salida	0.0127	m
A_o	Sección del orificio de salida	0.000126	m
A_v	Sección de la vena contracta		
C_c	Coeficiente de corrección entre A_o y A_v	0.6 a 1	
C_v	Coeficiente de corrección por pérdidas	0.8 a 0.99	
C_d	Coeficiente de descarga: $C_d = C_v C_c$	0.7	
α	Factor de flujo	0.000388	$m^{2.5}/s$

2.2. Modelo no lineal del sistema tanque de agua

En la Figura 2.1, se muestra esquemáticamente el proceso tanque con agua, donde el flujo de agua entrante q_i es controlado con la finalidad que el nivel h del tanque se mantenga constante, mientras existe un flujo de salida q_o del

mismo. El proceso descrito previamente es del tipo SISO (“Single Input Single Output”) ya que posee sólo una entrada: el flujo entrante q_i , y una salida: el nivel del líquido h .



Leyenda:

- FCV: Válvula de regulación de flujo de entrada al tanque con accionamiento eléctrico.
- HV: Válvula de descarga con accionamiento manual.
- LT: Transmisor de nivel

Figura 2.1 Esquema del proceso tanque de agua

El modelo matemático del sistema tanque de agua se obtiene usando el principio de conservación de la masa:

$$\frac{\text{Acumulación}}{\left[\begin{array}{c} \text{de} \\ \text{masa total} \\ \text{Tiempo} \end{array} \right]} = \frac{\text{Ingreso}}{\left[\begin{array}{c} \text{de} \\ \text{masa total} \\ \text{Tiempo} \end{array} \right]} - \frac{\text{Salida}}{\left[\begin{array}{c} \text{de} \\ \text{masa total} \\ \text{Tiempo} \end{array} \right]} \quad (2.1)$$

$$\frac{d(\rho \hat{A} h)}{dt} = \rho q_i - \rho q_o \quad (2.2)$$

$$\Rightarrow \hat{A} \frac{dh}{dt} = \hat{A} \dot{h} = q_i - q_o \quad (2.3)$$

La ecuación de Bernoulli para flujo estable, no viscoso e incompresible indica que el flujo de salida en el tanque es proporcional a la raíz cuadrada de la columna de agua en el tanque, en consecuencia tenemos que:

$$q_o = \alpha \sqrt{h} \quad (2.4)$$

, donde α es una constante proporcional que depende del coeficiente de descarga C_d , el área de la sección transversal del orificio A_o y la constante gravitacional, esto se aplica para orificios circulares pequeños, como el de la válvula de descarga HV de la figura 2.1.

$$\alpha = C_d A_o \sqrt{2g} \quad (2.5)$$

Además, de [5] se sabe que:

$$C_d = C_v C_c \quad 0.8 \leq C_v \leq 0.99 \quad 0.6 \leq C_c \leq 1$$

, donde C_v es el coeficiente de corrección por pérdidas y C_c es el coeficiente de corrección entre A_o y A_v (sección de la vena contracta). Para el estudio se tomó $C_d = 0.6$.

Sustituyendo en (2.4) en (2.3),

$$\hat{A} \frac{dh}{dt} = q_i - \alpha \sqrt{h} \quad (2.6)$$

Acomodando, la ecuación de estado no lineal que describe la dinámica del sistema del proceso tanque con agua es:

$$\frac{dh}{dt} = -\frac{\alpha}{\hat{A}}\sqrt{h} + \frac{1}{\hat{A}}q \quad (2.7)$$

2.3. Linealización del modelo

El procedimiento de linealización que se presenta aquí se basa en el desarrollo de la función no lineal en series de Taylor alrededor del punto de operación (\bar{H}, \bar{Q}) .

$$\begin{aligned} \frac{dx(t)}{dt} = f(x(t), u(t)) &= -\frac{\alpha}{\hat{A}}\sqrt{x(t)} + \frac{1}{\hat{A}}u(t) \\ y(t) = g(x(t), u(t)) &= x(t) \end{aligned} \quad (2.8)$$

, donde:

$x(t) = h(t)$, variable de estado.

$y(t) = h(t)$, variable de salida (medible)

$u(t) = q_i(t)$, variable de control.

$$\mathbf{A} = \left. \frac{\partial f}{\partial x} \right|_{(\bar{H})} = -\frac{\alpha}{2\hat{A}} \left. \frac{1}{\sqrt{x(t)}} \right|_{(\bar{H})} = -\frac{\alpha}{2\hat{A}\sqrt{\bar{H}}}$$

$$\mathbf{B} = \left. \frac{\partial f}{\partial u} \right|_{(\bar{H})} = \frac{1}{\hat{A}} \left. \frac{1}{1} \right|_{(\bar{H})} = \frac{1}{\hat{A}}$$

$$\mathbf{C} = \left. \frac{\partial g}{\partial x} \right|_{(\bar{H})} = 1 \left. \frac{1}{1} \right|_{(\bar{H})} = 1$$

$$\mathbf{D} = \left. \frac{\partial f}{\partial u} \right|_{(\bar{H})} = 0 \left. \frac{1}{1} \right|_{(\bar{H})} = 0$$

Estas operaciones permiten escribir la siguiente realización linealizada:

$$\frac{ax(t)}{dt} = -\frac{\alpha}{2\hat{A}\sqrt{H}}h(t) + \frac{1}{\hat{A}}q_i(t)$$

$$y(t) = h(t) \quad (2.9)$$

La función de transferencia del sistema viene dada por:

$$G(s) = C(sI - A)^{-1}B + D \quad (2.10)$$

Reemplazando los parámetros anteriormente hallados en (2.10)

$$G(s) = 1\left(s - \left(-\frac{\alpha}{2\hat{A}\sqrt{H}}\right)\right)^{-1}\frac{1}{\hat{A}} + 0$$

$$G(s) = \frac{1/\hat{A}}{s + \frac{\alpha}{2\hat{A}\sqrt{H}}}$$

La función de transferencia del modelo hallado pertenece a un sistema de primer orden. Como $\frac{\alpha}{2\hat{A}\sqrt{H}} > 0$, el único polo será negativo y esto implica estabilidad del sistema.

2.4. Prueba del sistema en lazo abierto usando Matlab

La prueba se realiza para reconocer el tipo de sistema en base a parámetros de respuesta en el tiempo. Esto se consigue excitando el modelo hallado con una entrada escalón.

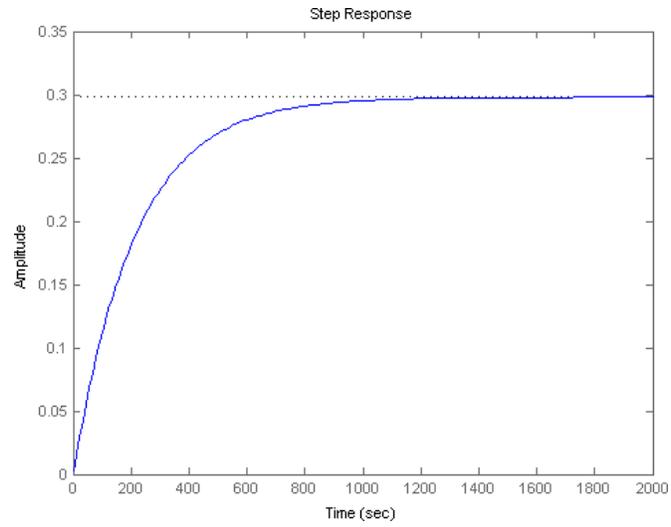


Figura 2.2 Respuesta del sistema de control en lazo abierto ($\bar{H} = 0.30m$)

Podemos comprobar que nuestro sistema dinámico es estable, tal como se mencionó anteriormente. Además es un sistema lento.

CAPITULO 3

IDENTIFICACIÓN DE LA PLANTA

La identificación de sistemas dinámicos es un planteamiento experimental que se define como la obtención de la estructura y parámetros de un modelo matemático, mediante datos experimentales, con la finalidad que dicho modelo represente con exactitud la dinámica real del proceso.

3.1. Identificación del modelo

3.1.1. Adquisición de datos

Incluye la elección de señales a medir, del tipo de entrada, periodo de muestreo, número de muestras a tomar, tratamiento previo de los datos (eliminación de ruidos y de datos erróneos, tratamiento de niveles de continua, etc.).

La extracción de los datos no es una tarea directa, por ello es necesario un cuidadoso tratamiento de los datos.

3.1.2. Selección de una estructura del modelo

Se define a partir del conocimiento previo del proceso y de las perturbaciones. En nuestro caso, trabajaremos con el método gráfico,

modelo de 1er Orden y 2do Orden usando el Toolbox de Identificación de Matlab, y los dos modelos paramétricos más conocidos: ARX y ARMAX.

3.1.3. Estimación de los parámetros del modelo

Ya elegida la estructura del modelo, es necesario determinar el valor de los parámetros que ajustan la respuesta del modelo a los datos de entrada – salida experimentales. Existen métodos de estimación de parámetros off-line y online, nosotros usaremos el algoritmo de los mínimos cuadrados. Para esta etapa usaremos el Toolbox de Identificación de Matlab. En la sección 3.4 se desarrolla este punto.

3.1.4. Validación del modelo identificado

Consiste en determinar si el modelo obtenido satisface el grado de exactitud requerido para la aplicación en cuestión. El Toolbox de Identificación de Matlab se basa en la simulación del modelo obtenido, y la comparación entre su salida y la respuesta real del sistema.

3.2. **Determinación del Rango de Trabajo**

En la identificación de sistemas estables, para hallar la zona lineal de operación del sistema se trabaja con el gráfico denominado característica estática, es decir, para distintas entradas se espera a que el sistema estabilice y anotamos la salida correspondiente, se grafican estos pares de puntos y se determina un rango lineal a partir de la gráfica.

En esta planta se conoce a priori que el sistema es inestable de tipo integrador para un rango de aperturas de válvula, es por esto que la técnica de

identificación se abocará a determinar el rango de valores de apertura de válvula para los cuales el sistema es estable. Se trabajó con distintas aperturas de válvula a partir del 25% ya que con valores inferiores a este, no se apreció cambios significativos en el nivel del tanque.

- *Apertura de válvula entre 25 y 30%*

En la Figura 3.1, se muestra la respuesta del nivel para una apertura de válvula de 25 a 30%. El nivel inicial para una apertura de 25% de válvula es de 1.65cm, luego del cambio de apertura se observa que el nivel llega a estabilizar en un valor aproximado de 3.6cm en un tiempo de $322 - 12.8 \approx 309$ segundos.

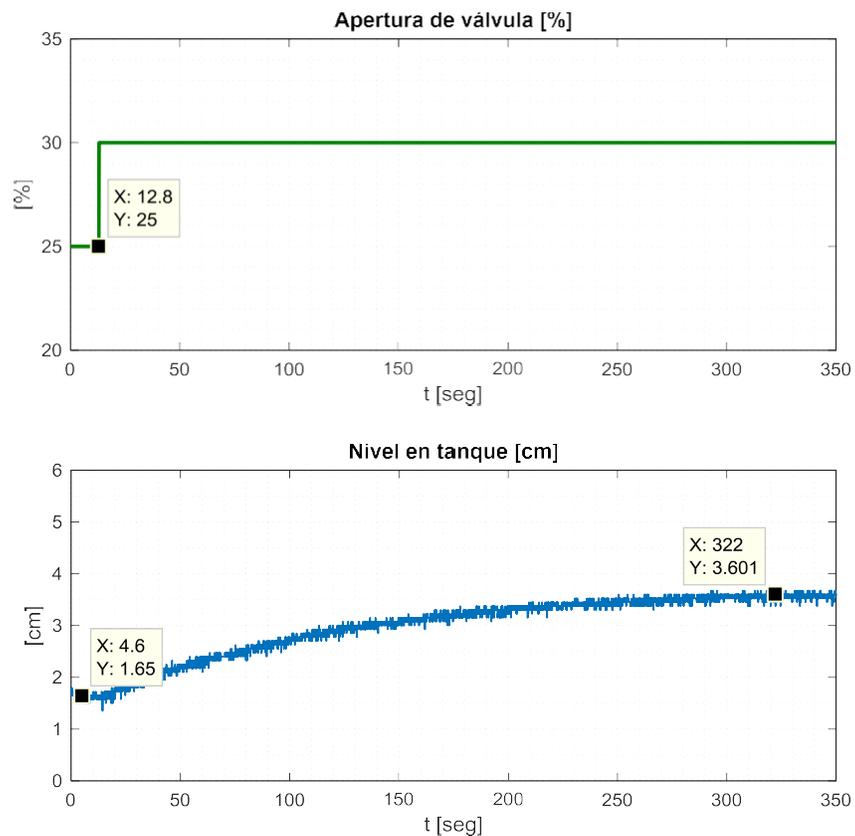


Figura 3.1 Respuesta con apertura de válvula de 25 a 30%

- *Apertura de válvula entre 30 y 35%*

En la Figura 3.2, se observa que para una apertura de válvula de 30 a 35%, el sistema llega a estabilizar en un valor aproximado de 7.2cm en un tiempo de $496.5 - 107.5 = 389$ segundos.

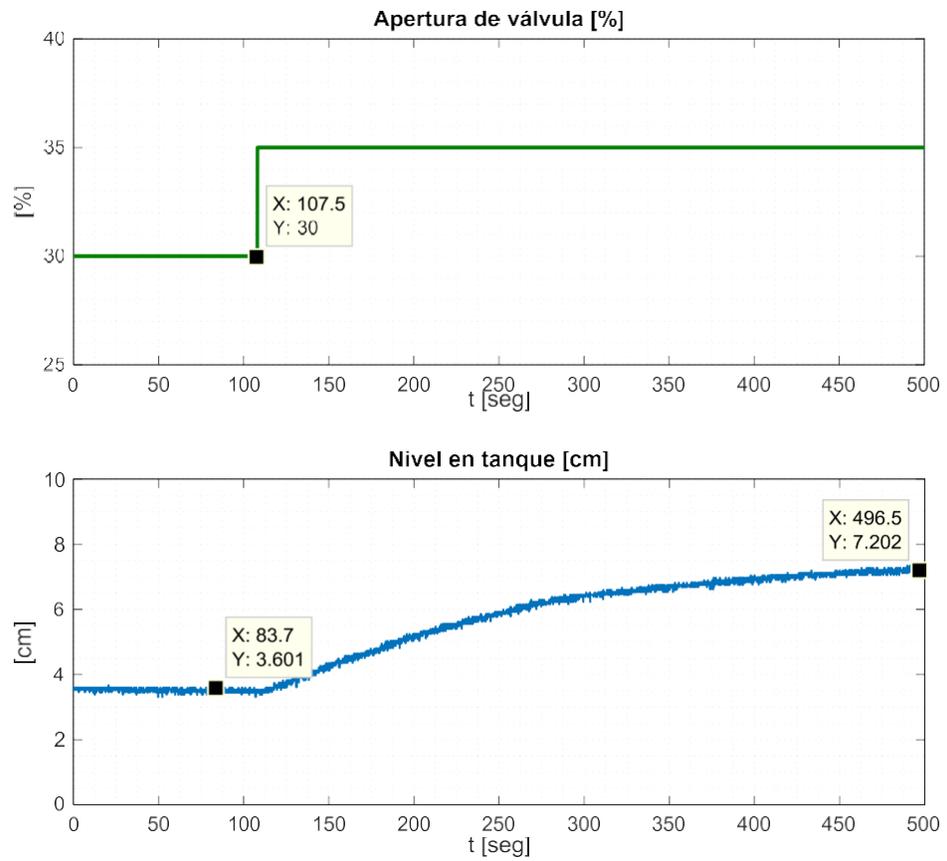


Figura 3.2 Respuesta con apertura de válvula de 30 a 35%

- *Apertura de válvula entre 35 y 40%*

En la Figura 3.3, se observa que para una apertura de válvula de 35 a 40%, el sistema llega a estabilizar en un valor aproximado de 13.6cm en un tiempo de $791.3 - 122.7 \approx 668$ segundos.

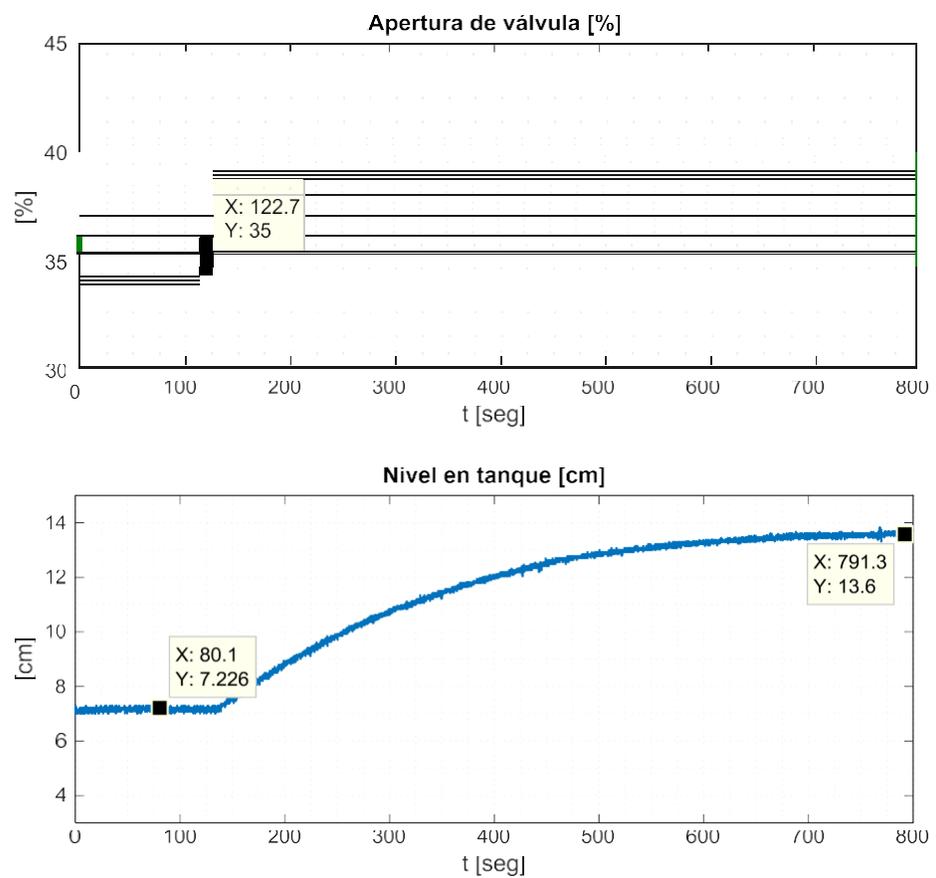


Figura 3.3 Respuesta con apertura de válvula de 35 a 40%

- *Apertura de válvula entre 40 y 45%*

En la Figura 4 se observa que para una apertura de válvula de 40 a 45%, el sistema llega a estabilizar en un valor aproximado de 15.3cm en un tiempo de $214.8 - 30.9 \approx 184$ segundos

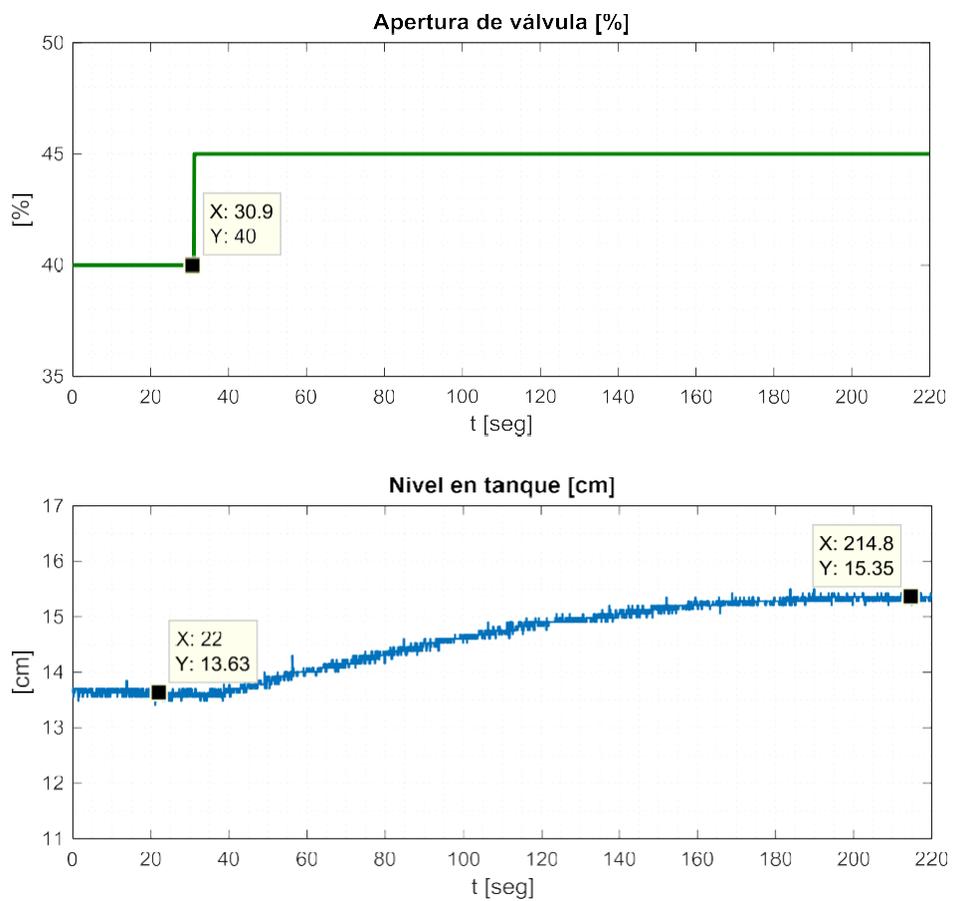


Figura 3.4 Respuesta con apertura de válvula de 40 a 45%

Luego de abrir la válvula a un valor superior a 50% se observa que el sistema tiene un comportamiento tipo integrador, es decir no llega a estabilizar, es por esto que se ha considerado aperturas de válvula solo hasta 45% para hallar el rango de trabajo del sistema.

Finalmente, se obtiene el gráfico para los distintos valores de establecimiento de nivel versus las aperturas de la válvula consideradas:

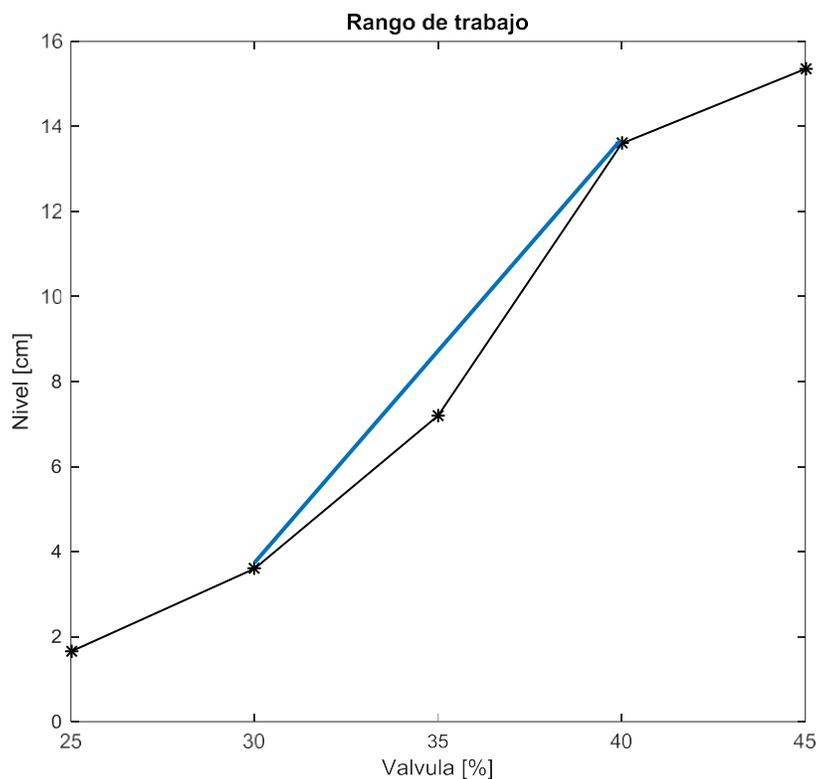


Figura 3.5 Rango de trabajo elegido

De la Figura 3.5, se puede apreciar que el mayor cambio en el nivel corresponde para las aperturas de válvula entre 30 y 40%, por tanto trabajaremos en este rango.

Se envió una señal escalón a la válvula de tal manera que abarque todo el rango de trabajo definido. En la Figura 3.6, se muestran la respuesta del sistema ante la señal de entrada escalón procesada en nuestra herramienta Matlab.

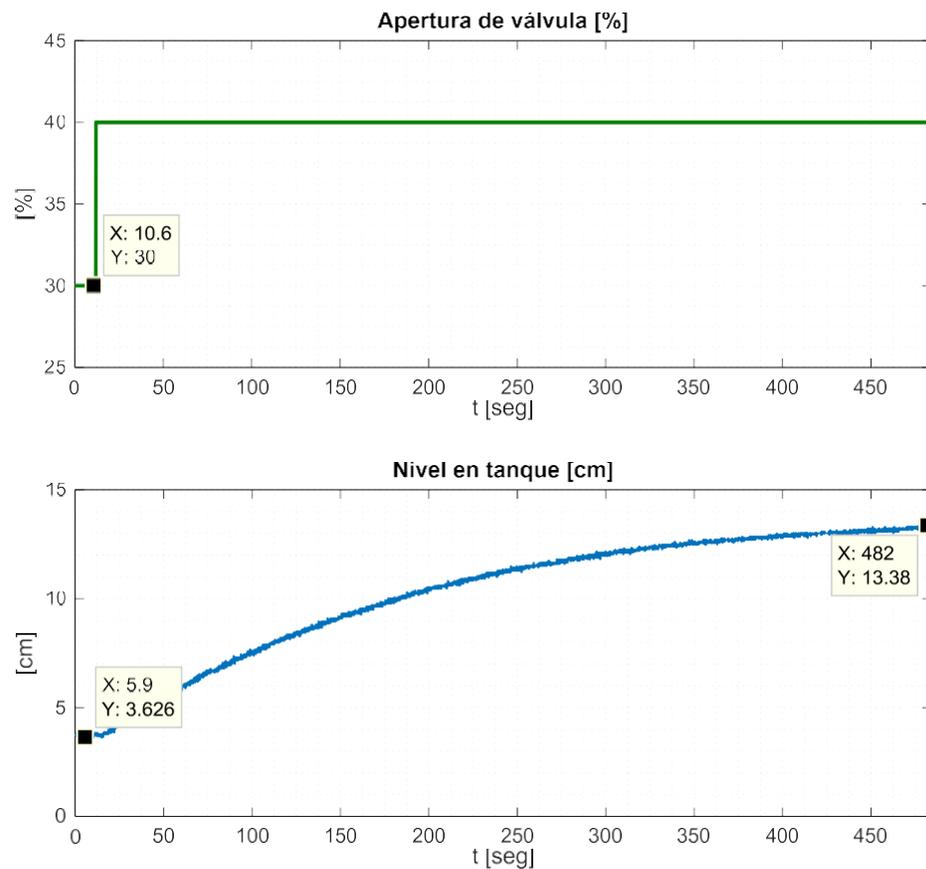


Figura 3.6 Respuesta del sistema ante escalón en el rango de trabajo definido

3.3. Identificación no paramétrica

3.3.1. Aproximación gráfica

El primer método de identificación a usar será el método por aproximación gráfica a un modelo de primer orden con retardo.

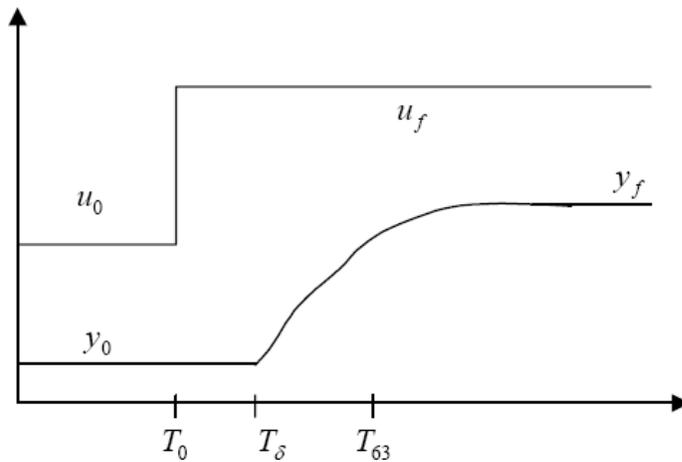


Figura 3.7 Identificación gráfica sistema de primer orden

Se quiere obtener un modelo de la forma:

$$G(s) = \frac{Ke^{-sT_d}}{\tau s + 1} \quad (3.1)$$

, donde:

$$K = \frac{y_f - y_0}{u_f - u_0}$$

$$\tau = T_{63} - T_\delta$$

$$T_d = T_\delta - T_0$$

$$Y_{63} = y_0 + 0.63(y_f - y_0)$$

Reemplazando los valores correspondientes a la respuesta escalón del proceso de la Figura 3.6, se obtiene:

$$K = \frac{13.38 - 3.626}{40 - 30} = 0.9754$$

$$Y_{63} = y_0 + 0.63(y_f - y_0) = 9.771 \rightarrow T_{63} = 179$$

$$T_{63} = 179$$

$$T_0 = 11.9$$

$$T_\delta = 16.1$$

$$T_d = 4.2$$

$$\tau = 179 - 16.1 = 162.9$$

Reemplazando los valores hallados en (3.1), se obtiene la siguiente la primera función de transferencia que aproxima al modelo:

$$G(s) = \frac{0.9754e^{-4.2s}}{162.9s + 1}$$

3.3.2. Modelo de 1er Orden usando el Toolbox de Identificación de Sistemas

Para que el Toolbox de Matlab pueda encontrar un modelo que encaje con los datos reales, necesitamos ingresar un vector de datos de entrada y datos de salida. Estos datos deben ser procesados previamente antes de ser ingresados, tanto los datos de entrada como los de salida deben ser colocados en el origen. En la Figura 3.8, se muestran los datos procesados, de esta figura se midió el retardo, cuyo valor es aproximadamente 4.2 segundos. Los mismos datos serán utilizados para hallar el modelo de 2do orden en la siguiente sección.

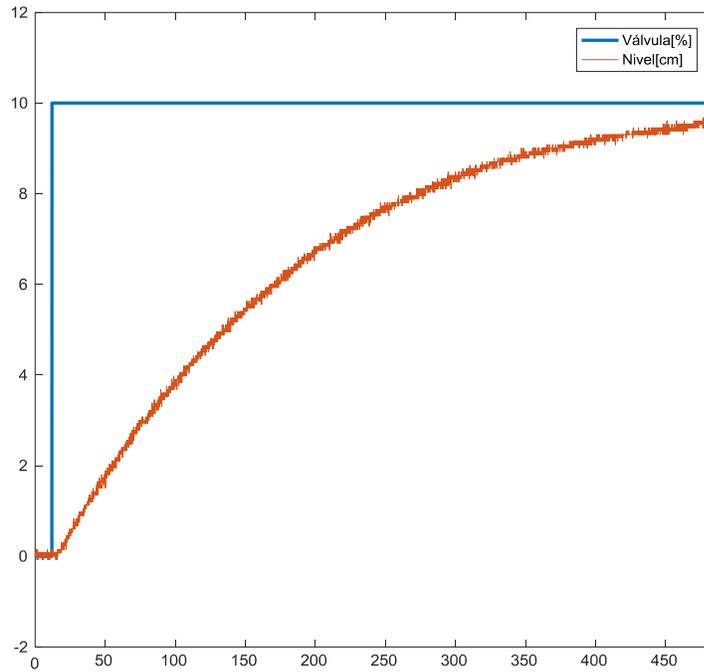


Figura 3.8 Procesamiento de datos

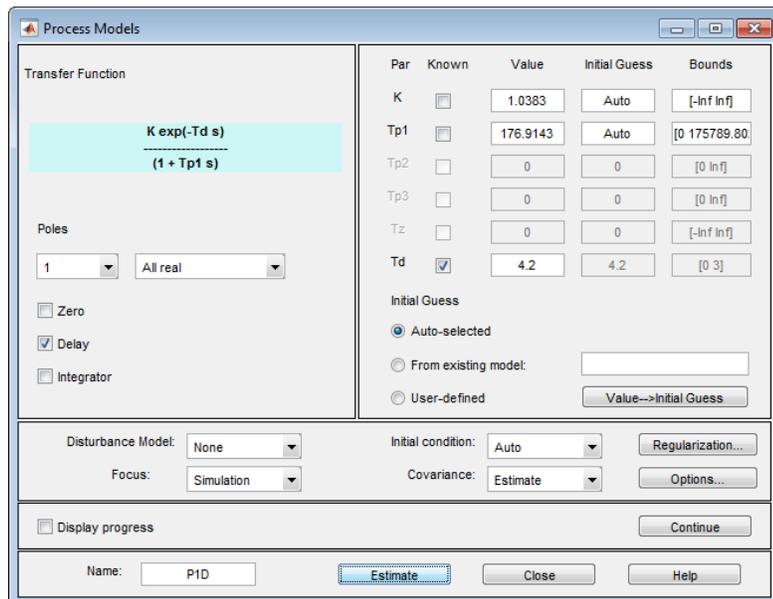


Figura 3.91 Parámetros obtenidos con el Toolbox de Identificación para un modelo de 1er Orden

En la Figura 3.9, se muestran los parámetros del modelo obtenido con la información introducida al Toolbox. Se ha considerado el modelo como aproximación de un sistema de 1er orden con retardo conocido (se ingresó manualmente), esto para facilitar la tarea de identificación al Toolbox y se obtenga un mejor modelo.

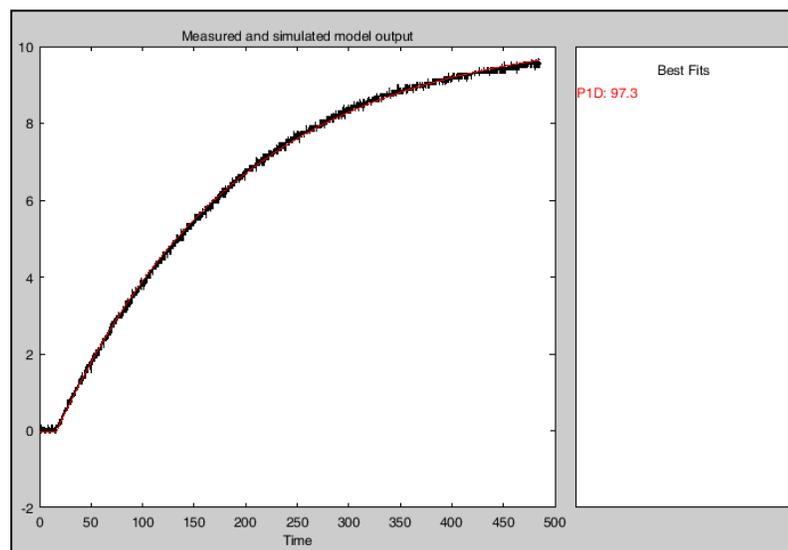


Figura 3.10 Aproximación a un modelo de 1er orden con el Toolbox Identificación

En la Figura 3.10, se muestra la respuesta del modelo de primer orden obtenida, en este caso con un aproximación (FIT) de 97.3%.

Tomando los parámetros hallados en la Figura 3.9, el modelo obtenido es el siguiente:

$$G(s) = \frac{1.038e^{-4.2s}}{176.9s + 1}$$

3.3.3. Modelo de 2do Orden usando el Toolbox de Identificación de Sistemas

Aproximamos a un sistema de 2do orden sobre amortiguado para obtener un mejor modelo y se consideró el retardo como conocido.

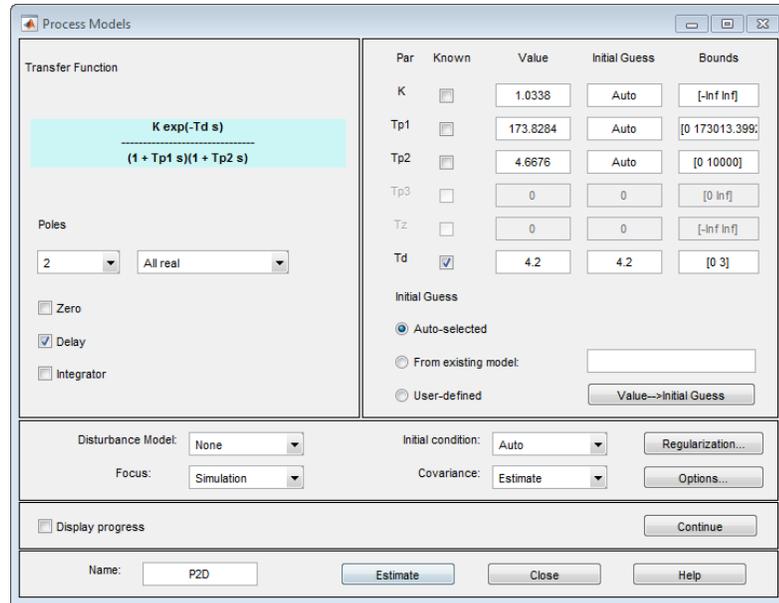


Figura 3.102. Parámetros obtenidos con el Toolbox de Identificación para un modelo de 2do Orden

En la Figura 3.10, se muestran los parámetros del modelo obtenido con la información introducida al Toolbox.

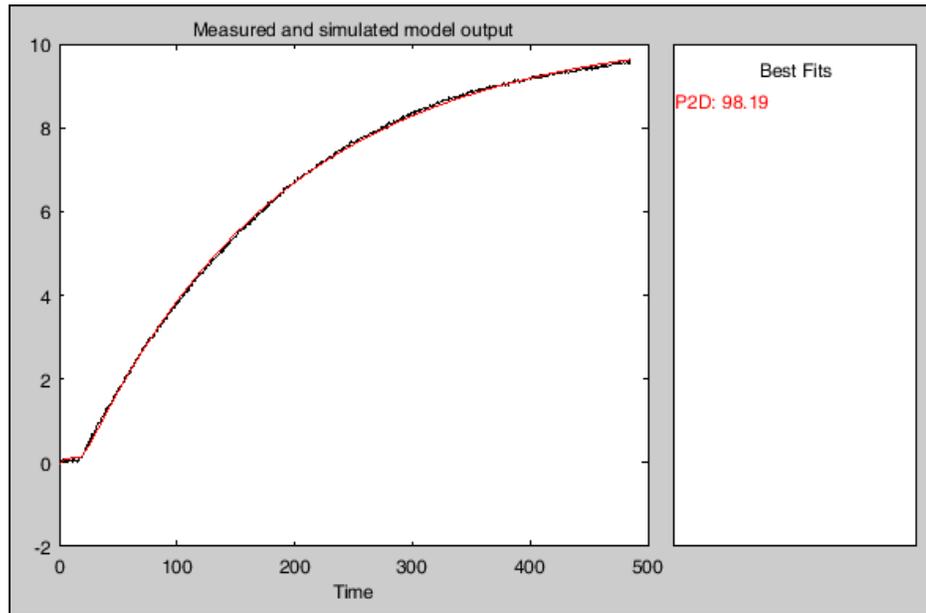


Figura 3.113 Aproximación a un modelo de 2do orden con el Toolbox Identificación

En la Figura 3.11, se muestra la respuesta del modelo de segundo orden que se obtiene, en este caso con una aproximación (FIT) de 98.19%.

Tomando los parámetros hallados en la Figura 3.10, el modelo obtenido es el siguiente:

$$G(s) = \frac{1.0338}{(1 + 173.8284s)(1 + 4.6676s)} e^{-4.2s}$$

3.4. Identificación paramétrica

La estructura general de los modelos paramétricos para sistemas lineales invariantes en el tiempo, se muestra en la Figura 3.12 y es dada por:

$$A(q)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t) \quad (3.2)$$

, donde $u(t)$, $y(t)$, $e(t)$ son la entrada, salida y ruido con valor medio cero.

También, pueden ser representadas mediante una ecuación lineal en diferencias, como:

$$y(t) + a_1y(t-1) + \dots + a_{n_a}y(t-n_a) = b_1u(t-1) + \dots + b_{n_b}u(t-n_b) + e(t)$$

El vector θ , llamado vector de parámetros, es el objetivo del estudio, encontrando dicho vector podemos conocer el modelo discreto y por ende el modelo continuo.

$$\theta = [a_1 \ a_2 \ \dots \ a_{n_a} \ b_1 \ \dots \ b_{n_b}]$$

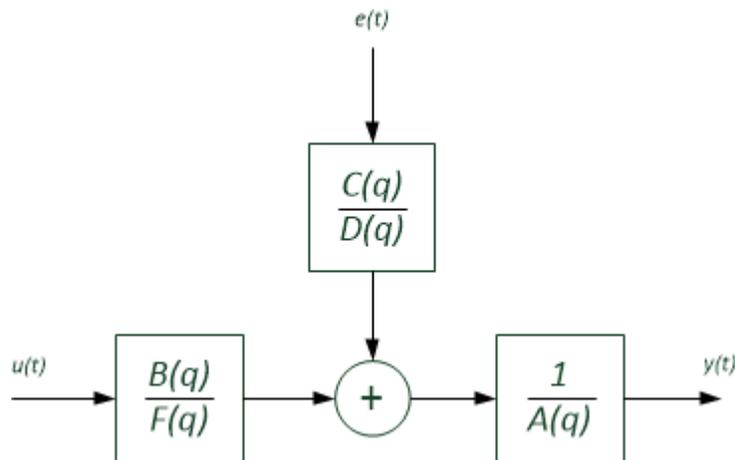


Figura 3.12 Estructura general de los modelos paramétricos para SLIT

Dentro de estas estructuras paramétricas principales que nos proporcionan las herramientas del software de simulación como Matlab está:

- ✓ Estructura Paramétrica ARX
- ✓ Estructura Paramétrica ARMAX

3.4.1. Estructura ARX

Presenta la estructura de la Figura 3.13. La condición sobre la estructura general de modelos es que $F(q) = D(q) = C(q) = 1$

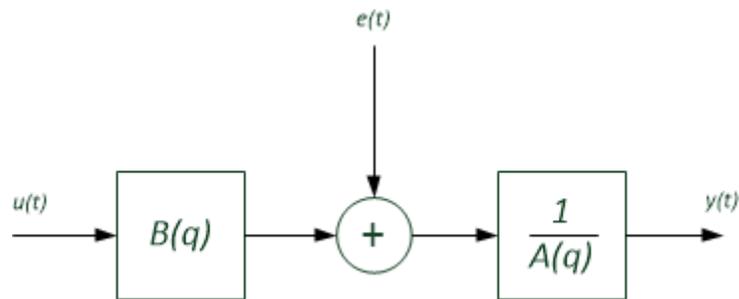


Figura 3.13 Estructura ARX

El modelo ARX presenta la relación entrada y salida descrita por una ecuación en diferencia lineal en la forma:

$$y(t) = \frac{B(q)}{A(q)}u(t) + e(t) \quad (3.3)$$

El ruido blanco $e(t)$ entra como un error directo en la ecuación, entonces debería ser considerado insignificante para obtener una mejor estimación.

3.4.2. Estructura ARMAX

Presenta la estructura de la Figura 3.14. Ésta permite modelar la parte determinística y la parte estocástica del proceso con un promediador móvil.

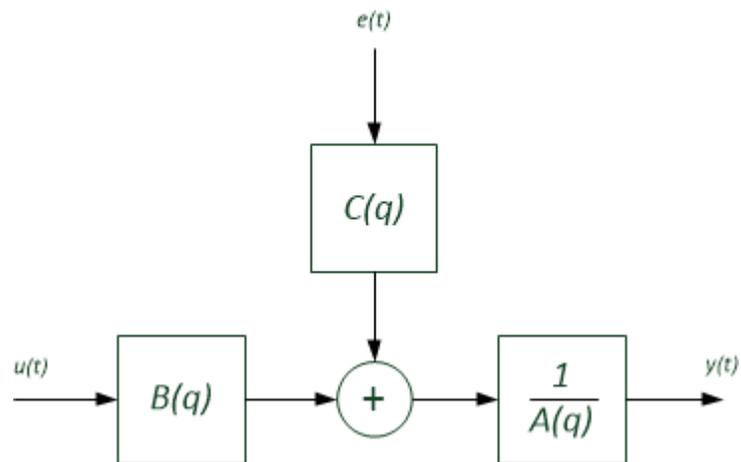


Figura 3.14 Estructura ARMAX

El modelo ARMAX presenta la relación entrada y salida descrita por una ecuación en diferencia lineal en la forma:

$$y(t) = \frac{B(q)}{A(q)}u(t) + \frac{C(q)}{A(q)}e(t) \quad (3.4)$$

3.5. Algoritmo de los mínimos cuadrados con factor de olvido

Es un método usado para la estimación de parámetros en línea, minimizando el error de predicción. Para desarrollar y entender adecuadamente el método empleado para la identificación recursiva es necesario revisar algunos conceptos básicos (Ver Apéndice B).

3.5.1. Modelo ARX de Orden 1

Se ha desarrollado el algoritmo recursivo en Simulink (Ver Apéndice C), para finalmente tener el bloque ARX de la Figura 3.15.

Introducimos las señales de entrada y salida reales a la estructura de identificación ARX como se muestra en la Figura 3.15, para una mejor precisión se enviaron los parámetros obtenidos al Workspace de Matlab.

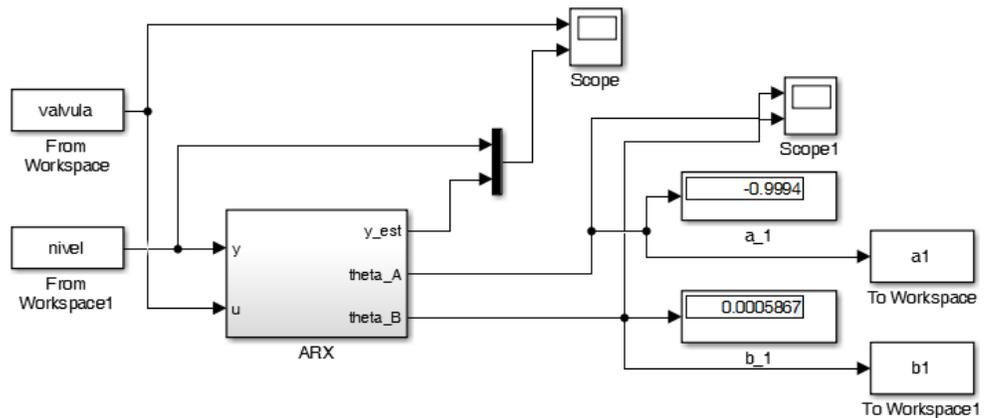


Figura 3.154 Estructura de Identificación ARX para el Proceso Tanque con Agua

Al aplicar la identificación recursiva por mínimos cuadrados se obtuvieron los siguientes parámetros para el modelo ARX.

$$a_1 = -0.999425$$

$$b_1 = 0.0005867$$

Por lo tanto, nuestros polinomios serían:

$$B(q^{-1}) = 0.0005867q^{-1}$$

$$A(q^{-1}) = 1 - 0.99942q^{-1}$$

3.5.2. Modelo ARMAX de Orden 1

Se ha desarrollado el algoritmo recursivo en Simulink (Ver Apéndice D), para finalmente tener el bloque ARMAX en la Figura 3.16.

Introducimos las señales de entrada y salida reales a la estructura de identificación ARMAX como se muestra en la Figura 3.16, se enviaron los parámetros obtenidos al Workspace de Matlab para una mejor precisión.

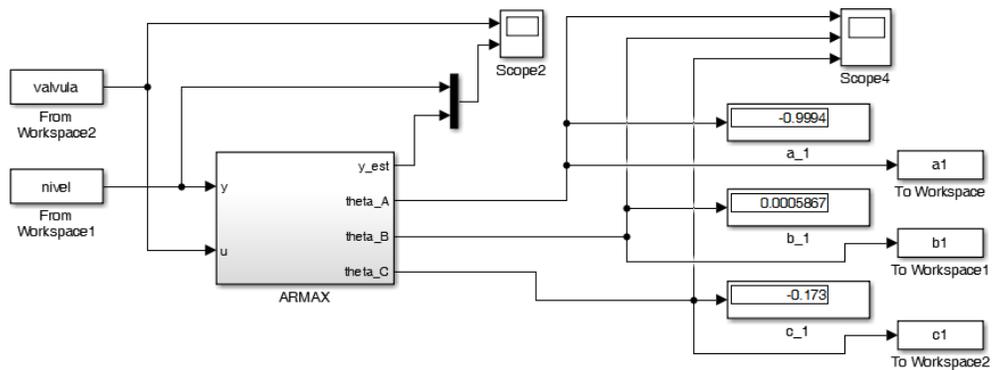


Figura 3.16 Estructura de Identificación ARMAX para el Proceso Tanque con Agua

Al aplicar la identificación recursiva por mínimos cuadrados se obtuvieron los siguientes parámetros para el modelo ARMAX.

$$a_1 = -0.999435$$

$$b_1 = 0.0005867285$$

$$c_1 = -0.172973$$

Por tanto nuestros polinomios serían:

$$A(q^{-1}) = 1 - 0.999435q^{-1}$$

$$B(q^{-1}) = 0.0005867285q^{-1}$$

$$C(q^{-1}) = 1 - 0.172973q^{-1}$$

3.6. Implementación del algoritmo en el proceso

3.6.1. Aproximación gráfica

Se ha implementado el modelo de primer orden obtenido en la sección 3.3.1 y se le ha aplicado un escalón con los mismos porcentajes de apertura al que se le aplicó a la planta real.

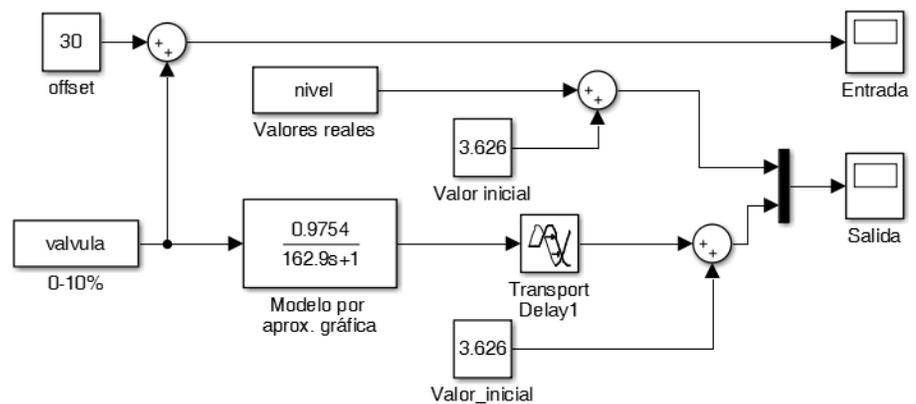


Figura 3.17 Implementación del modelo por aproximación gráfica en Simulink

3.6.2. Modelo de 1er Orden obtenido con el Toolbox de Identificación

Se ha implementado el modelo de primer orden obtenido en la sección 3.3.2 y se le ha aplicado un escalón con los mismos porcentajes de apertura al que se le aplicó a la planta real.

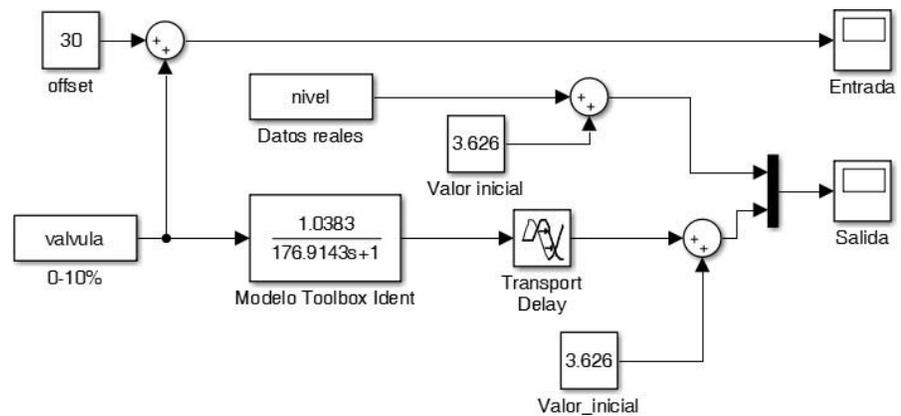


Figura 3.18. Implementación del modelo de 1er Orden obtenido con el Toolbox de Identificación en Simulink

3.6.3. Modelo de 2do Orden obtenido con el Toolbox de Identificación

Se ha implementado el modelo de segundo orden obtenido en la sección 3.3.3, al cual aplicamos un escalón con los mismos porcentajes de apertura al aplicado a la planta real.

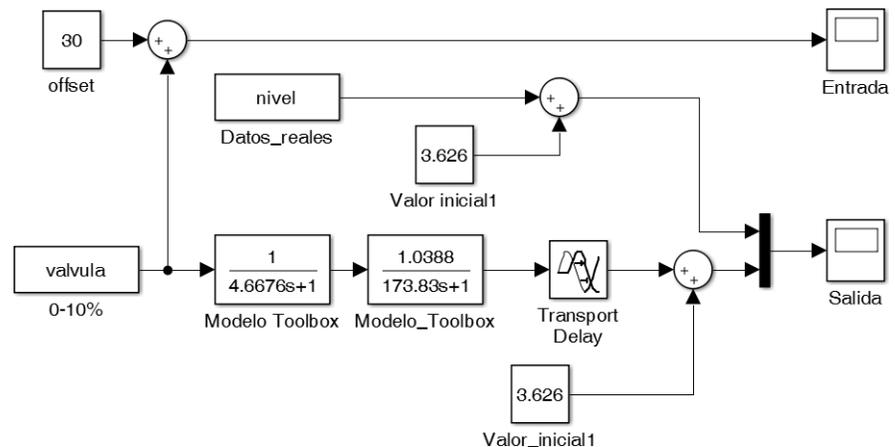


Figura 3.195. Implementación del modelo de 2do Orden obtenido con el Toolbox de Identificación en Simulink

3.6.4. Modelo ARX de Orden 1

Se implementó el modelo, usando los parámetros hallados en la sección 3.5.1 y se comparó su salida con la de la planta real. En la Figura 3.20, se muestra la estructura de validación.

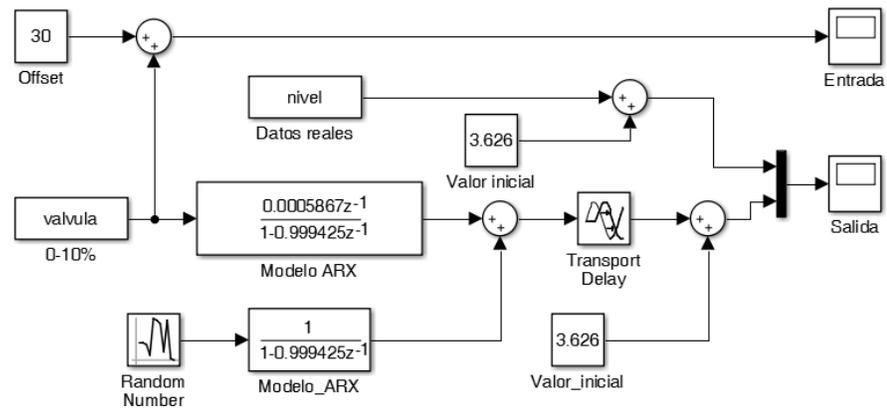


Figura 3.206 Implementación del modelo ARX de Orden 1 en Simulink

3.6.5. Modelo ARMAX de Orden 1

Se implementó el modelo, usando los parámetros hallados en la sección 3.5.2 y se comparó su salida con la de la planta real. En la Figura 3.21, se muestra la estructura de validación.

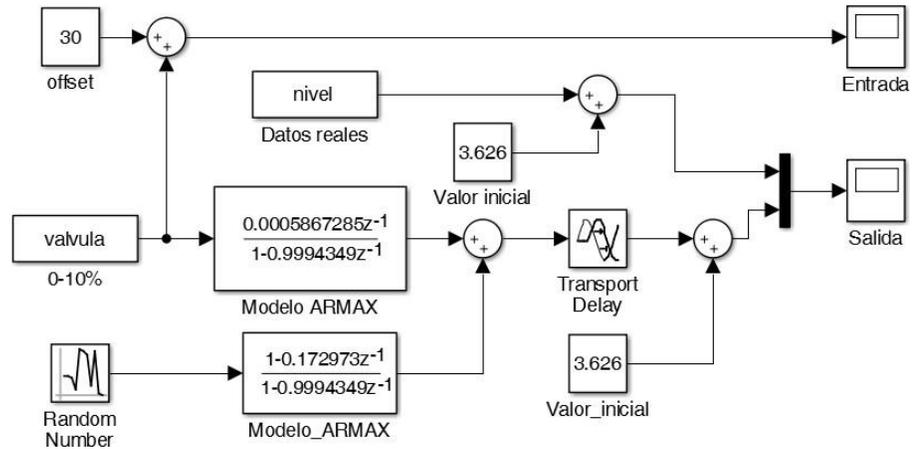


Figura 3.21 Implementación del modelo ARMAX de Orden 1 en Simulink

3.7. Validación de los modelos obtenidos

Para determinar la validez de los modelos propuestos y de los valores de parámetro obtenidos, se realiza la prueba en lazo abierto, que consiste en aplicar una entrada escalón al sistema experimental para comparar, cualitativa y cuantitativamente, los resultados de su respuesta con los generados numéricamente para el modelo empleando MATLAB.

3.7.1. Aproximación gráfica

En la Figura 3.22, se muestra la comparación entre las respuestas de la planta real y del modelo hallado.

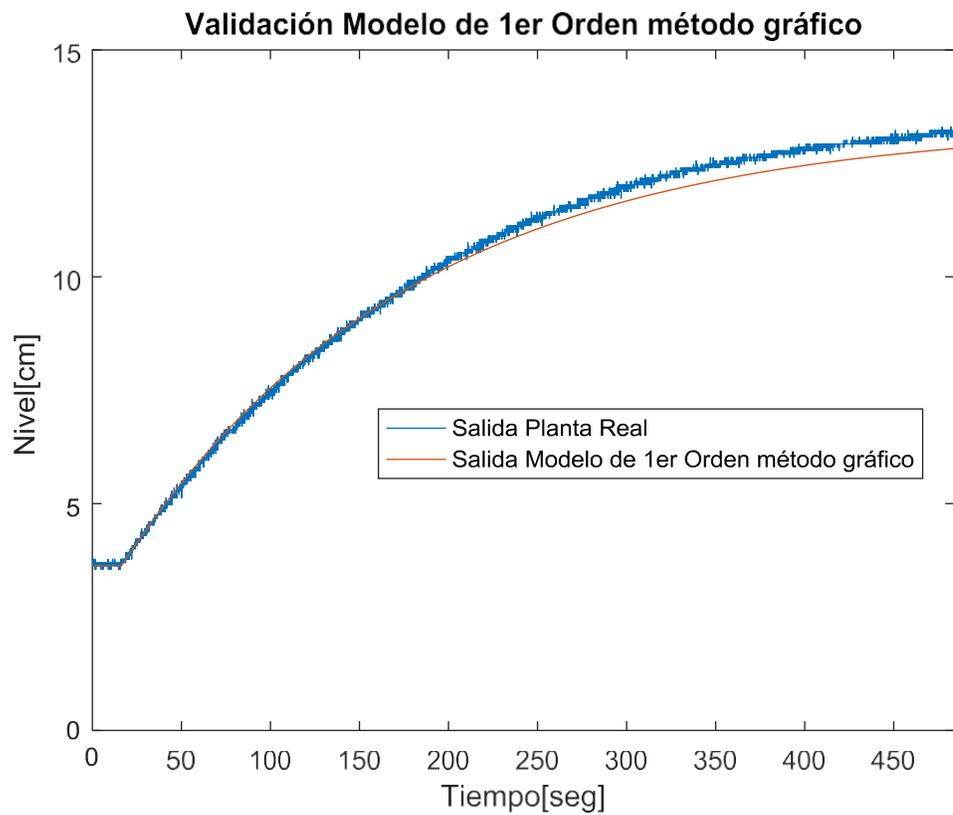


Figura 3.22 Validación del modelo obtenido por el método de aproximación gráfica

Se puede apreciar que la respuesta del modelo es una primera aproximación válida de la planta real. Para obtener modelos más precisos se utilizan otras herramientas como el Toolbox de Identificación de Sistemas de Matlab y los métodos de identificación paramétrica como ARX y ARMAX.

3.7.2. Modelo de 1er Orden obtenido con el Toolbox de Identificación

De manera similar al caso anterior, se ha validado el modelo comparando en una misma gráfica las respuestas de la planta real y del modelo.

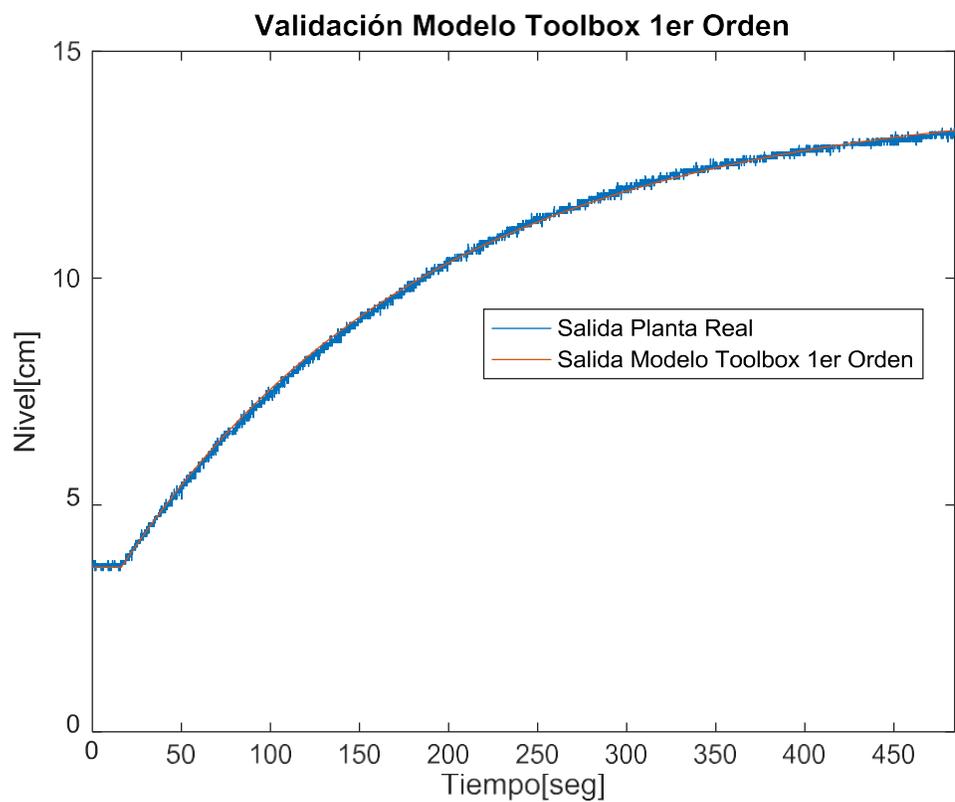


Figura 3.23 Validación del modelo de 1er Orden obtenido con el Toolbox Identificación de Sistemas de Matlab

Se puede apreciar que el modelo obtenido con el Toolbox resulta mejor que el obtenido mediante aproximación gráfica.

3.7.3. Modelo de 2do Orden obtenido con el Toolbox de Identificación

Se ha validado el modelo comparando en una misma gráfica las respuestas de la planta real y del modelo. En la Figura 3.24, se observa una buena aproximación de la respuesta del sistema con la del modelo de 2do orden sobre amortiguado utilizando el Toolbox de identificación.

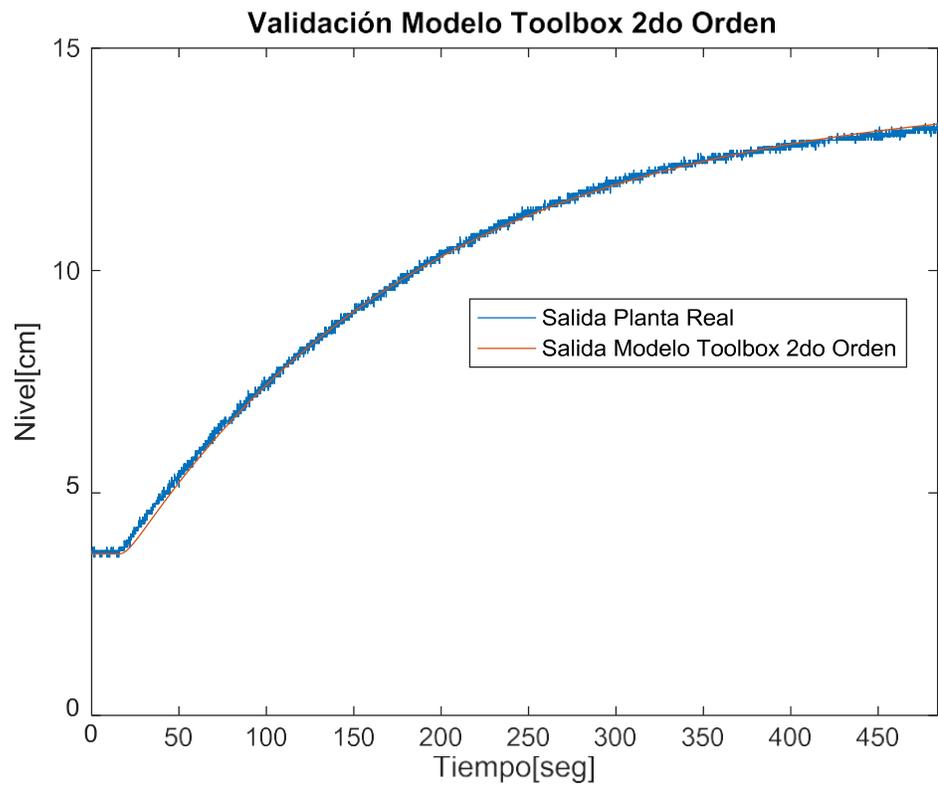


Figura 3.24 Validación del modelo de 2do Orden obtenido con el Toolbox Identificación de Sistemas de Matlab

3.7.4. Modelo ARX de Orden 1

En la Figura 3.25, se muestra la comparación de la salida del modelo ARX de 1er orden con la salida real de la planta. Como se puede apreciar la respuesta obtenida se aproxima bastante a la respuesta de la planta real, el modelamiento paramétrico con la estructura ARX no contempla la dinámica de la perturbación (Parámetro C), es por esto que hay un error en la estimación.

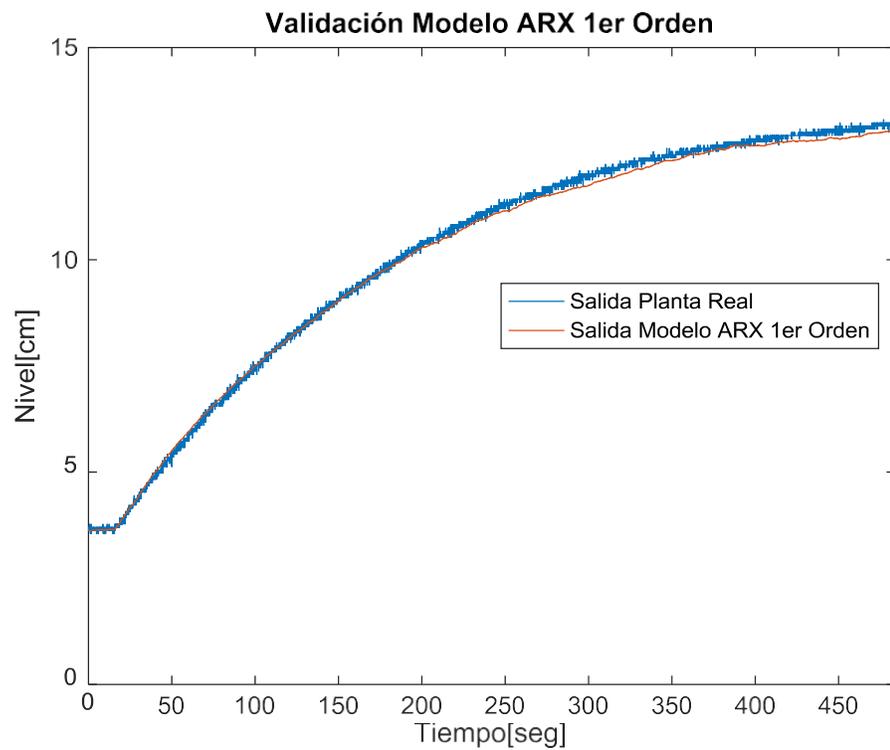


Figura 3.25 Validación Modelo ARX Orden 1

3.7.5. Modelo ARMAX de Orden 1

En la Figura 3.26, se muestra la comparación de la salida del modelo ARMAX de 1er orden con la salida real de la planta. Como se puede apreciar la respuesta obtenida se aproxima mucho mejor a la respuesta de la planta real, esto es debido a que el modelamiento paramétrico con la estructura ARMAX contempla la dinámica de la perturbación (Parámetro C).

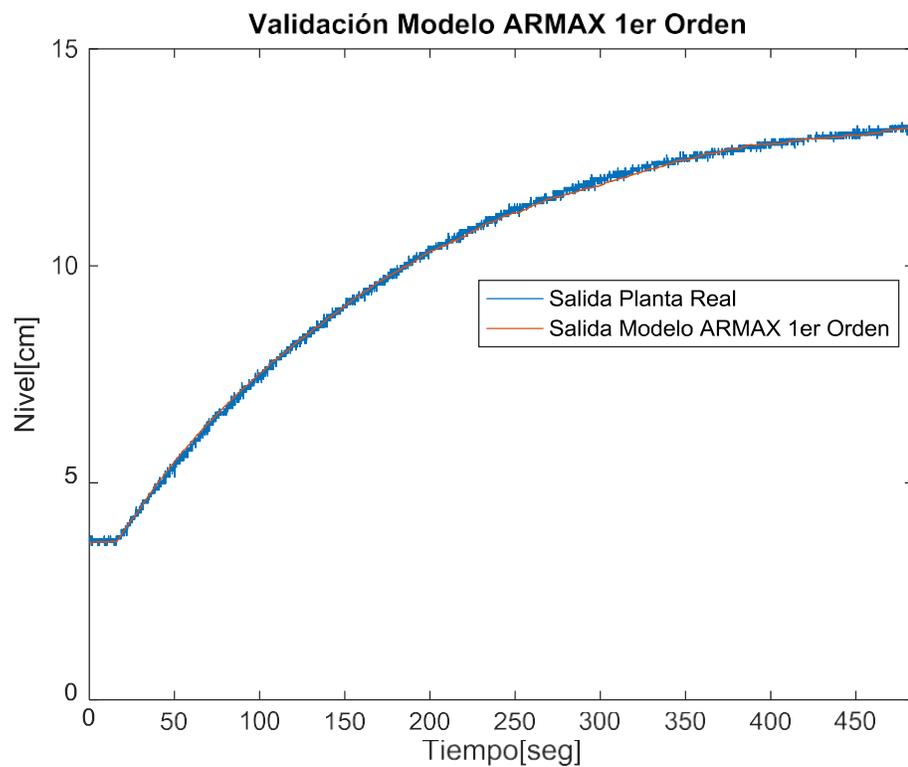


Figura 3.26 Validación Modelo ARMAX 1er Orden

3.8. Comparación de modelos

Se ha utilizado el criterio de minimización de error conocido como error cuadrático medio para comparar el modelo que mejor se aproxima a la planta real. Un menor valor de ECM indica una mejor aproximación a la planta real.

$$ECM = \frac{1}{N} \sum_{i=1}^N (\hat{Y} - Y_i)^2$$

, donde:

Y_i : Vector de datos reales de la salida de la planta.

\hat{Y} : Vector de datos de la salida del modelo de la planta.

N : Número de datos considerados.

Luego de aplicar el criterio de minimización, se obtienen los siguientes resultados para los 5 modelos obtenidos.

Tabla 3.1 Comparación de índices ECM

MODELO	ECM
Modelo gráfico	0.061407103447707
Toolbox Ident 1er Orden	0.006270275871372
Toolbox Ident 2do Orden	0.009235981575384
ARX Orden 1	0.017674176053553
ARMAX Orden 1	0.006344139361459

Como se puede apreciar en la Tabla 3.1, los modelos que presentan el menor ECM y por tanto se aproximan mejor a la planta son el ARMAX de primer orden, Toolbox Ident 1er Orden y Toolbox Ident 2do Orden.

CAPITULO 4

DISEÑO DEL SISTEMA DE CONTROL

4.1. **El controlador PID**

La utilidad de los controladores PID estriba en que se aplican en forma casi general a la mayoría de los sistemas de control. En particular, cuando el modelo matemático de la planta no se conoce y, por lo tanto, no se pueden emplear métodos de diseño analíticos, es cuando los controladores PID resultan más útiles. Se emplea para controlar procesos caracterizados por tener una entrada y una salida.

El objetivo de control consiste en diseñar una señal de control u , generada por el algoritmo de control, que sea capaz de minimizar la señal de error $e = r - y$, cumpliendo determinadas especificaciones de diseño previamente establecidas, rechazando al mismo tiempo la acción de los disturbios que actúan sobre el sistema de control.

4.2. Sintonización por Ziegler-Nichols

El proceso de seleccionar los parámetros del controlador que cumplan con las especificaciones de comportamiento requeridas se conoce como sintonía del controlador.

El método de Ziegler-Nichols en lazo abierto es el más difundido en la sintonización del controlador PID, siendo convenientes cuando no se conocen los modelos matemáticos de las plantas.

Ziegler y Nichols propusieron reglas para determinar los valores de la ganancia proporcional K_p , del tiempo integral T_i y del tiempo derivativo T_d , basándose en las características de respuesta transitoria de una planta dada.

Tabla 4.1 Método de Ziegler – Nichols en lazo abierto

Tipo de Controlador	K_p	T_i	T_d
P	$\frac{\tau}{K * T_d}$	Inf	0
PI	$0.9 * \frac{\tau}{K * T_d}$	$3 * T_d$	0
PID	$1.2 * \frac{\tau}{K * T_d}$	$2 * T_d$	$0.5 * T_d$

Utilizando el modelo de primer orden hallado en la sección 3.3.2, del que se obtuvo una mejor aproximación al sistema real, se obtienen los siguientes parámetros de sintonización:

- Para un controlador tipo PI:

$$K_p = 0.9 \left(\frac{176.9}{1.038 * 4.2} \right) = 36.52$$

$$T_i = 3(4.2) = 12.6$$

- Para un controlador tipo PID:

$$K_p = 1.2 \left(\frac{176.9}{1.038 * 4.2} \right) = 48.69$$

$$T_i = 2(4.2) = 8.4$$

$$T_d = 0.5(4.2) = 2.1$$

4.3. Método analítico de sintonía por ubicación de polos

El método de diseño por ubicación de polos, busca que las raíces de la ecuación característica del sistema sean ubicadas en posiciones deseadas (de acuerdo a requerimientos de la respuesta del sistema)

De la ecuación (2.9),

$$\frac{dx(t)}{dt} = -\frac{\alpha}{2\hat{A}\sqrt{H}} h t + \frac{1}{\hat{A}} q_i(t)$$

Hallamos la ecuación característica del sistema

$$|sI - A| = s - \left(-\frac{\alpha}{2\hat{A}\sqrt{H}} \right) = s + \frac{\alpha}{2\hat{A}\sqrt{H}} = 0$$

$$s = -\frac{\alpha}{2\hat{A}\sqrt{H}} \rightarrow s < 0$$

Por lo tanto, el sistema siempre es estable, ya que su único polo se encuentra en el eje real negativo.

Necesitamos convertir el modelo de 2do orden, obtenido en la sección 3.3.3, a espacio de estados para aplicar el diseño de ubicación de polos.

$$G(s) = \frac{1}{(1 + 4.6676s)} \frac{1.0388}{(1 + 173.83s)}$$

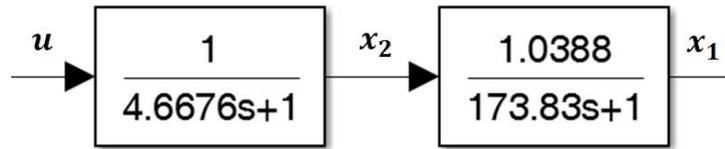


Figura 4.1 Función de transferencia de los estados considerados

Transformando a espacio estados, para cada función de transferencia:

$$x_2 = \frac{1}{4.6676s + 1} u$$

$$4.6676sx_2 + x_2 = u \rightarrow 4.6676\dot{x}_2 + x_2 = u$$

$$\dot{x}_2 = -\frac{1}{4.6676} x_2 + \frac{1}{4.6676} u \quad (4.1)$$

$$x_2 = -\frac{1}{4.6676} x_2 + \frac{1}{4.6676} u$$

$$x_1 = \frac{1.0388}{173.8284s + 1} x_2$$

$$173.8284sx_1 + x_1 = 1.0388x_2 \rightarrow 173.8284\dot{x}_1 + x_1 = 1.0388x_2$$

$$\dot{x}_1 = -\frac{1}{173.8284} x_1 + \frac{1.0388}{173.8284} x_2 \quad (4.2)$$

Usando las ecuaciones (4.1) y (4.2), se obtiene el modelo en espacio de estados:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{173.8284} & \frac{1.0388}{173.8284} \\ 0 & -\frac{1}{4.6676} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 4.6676 \end{bmatrix} u$$

$$y = [1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

De donde las matrices del espacio de estados son:

$$A = \begin{bmatrix} -0.0058 & 0.0060 \\ 0 & -0.2142 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0.2142 \end{bmatrix}$$

$$C = [1 \quad 0]$$

$$D = [0]$$

Usamos el comando de Matlab para obtener el modelo discreto, considerando un tiempo de muestreo ($T_s = 1$):

$$[A_d, B_d, C_d, D_d] = \text{c2dm}(A, B, C, D, T_s, 'zoh')$$

$$A_d = \begin{bmatrix} 0.9943 & 0.0054 \\ 0 & 0.8072 \end{bmatrix}$$

$$B_d = \begin{bmatrix} 0.0006 \\ 0.1928 \end{bmatrix}$$

$$C_d = [1 \quad 0]$$

$$D_d = [0]$$

Para conseguir error en estado estacionario cero, agregamos un integrador a la planta.

$$\hat{A}_d = \begin{bmatrix} A_d & B_d \\ 0 & 0 \end{bmatrix}$$

$$\hat{B}_d = \begin{bmatrix} 0 \\ B_d \end{bmatrix}$$

Se obtienen las siguientes matrices:

$$\hat{A}_u = \begin{bmatrix} 0.9943 & 0.0054 & 0.0006 \\ 0 & 0.8072 & 0.1928 \\ 0 & 0 & 0 \end{bmatrix} \quad \hat{B}_u = \begin{bmatrix} 0 \\ 0 \\ 0.2142 \end{bmatrix}$$

A partir de esto, podemos hallar la matriz de controlabilidad:

$$C_o = [\hat{B}_u \quad \hat{A}_u \hat{B}_u \quad \hat{A}_u^2 \hat{B}_u]$$

$$C_o = \begin{bmatrix} 0 & 0.0001 & 0.0003 \\ 0 & 0.0413 & 0.0333 \\ 0.2142 & 0 & 0 \end{bmatrix} \quad (4.3)$$

Si el rango de la matriz de controlabilidad es igual a la dimensión de \hat{A}_u entonces el sistema es controlable.

$$\text{rank}(C_o) = 3$$

Si el determinante de la matriz de controlabilidad es diferente de cero entonces el sistema es controlable.

$$\det(C_o) = -2.1728(10^{-6})$$

La ubicación de polos se realizará de manera que la respuesta del sistema se aproxime uno de segundo orden. Para calcular la frecuencia natural y el coeficiente de amortiguamiento a partir de un sobre impulso deseado y un tiempo de establecimiento, se usan las siguientes expresiones.

$$T_{es} \approx \frac{4}{\zeta \omega_n}$$

$$m_p = 100e^{-\zeta\pi/\sqrt{1-\zeta^2}}$$

Sin embargo se conoce priori que nuestro sistema no presenta sobre impulso es por esto que se elegirá coeficiente de amortiguamiento es mayor que 1 que se cumple para sistemas sobre amortiguados.

Inicialmente se eligió un $\zeta = 1.5$ y un tiempo de establecimiento de 20s, con estos datos se obtuvo una frecuencia natural de:

$$\omega_n = 0.1333 \frac{rad}{seg}$$

Los polos deseados dominantes en el dominio s son:

$$s_1 = -\zeta\omega_n + (\omega_n\sqrt{1-\zeta^2})i \quad (4.4)$$

$$s_2 = -\zeta\omega_n - (\omega_n\sqrt{1-\zeta^2})i \quad (4.5)$$

Se añade un tercer polo insignificante respecto a los otros, se ubica al menos 10 veces a la izquierda de los otros polos.

$$s_3 = -10\zeta\omega_n \quad (4.6)$$

Reemplazando en las ecuaciones (4.4), (4.5) y (4.6), se obtienen los polos:

$$s_1 = -0.3491 \quad s_2 = -0.0509 \quad s_3 = -2$$

Las partes imaginarias se cancelan debido al valor del coeficiente de amortiguamiento elegido.

Los polos deseados en el dominio Z se obtienen con las expresiones

$$z_1 = e^{T_s S_1} \quad z_2 = e^{T_s S_2} \quad z_3 = e^{T_s S_3}$$

Realizando los reemplazos respectivos se obtienen los polos en el dominio Z

$$z_1 = 0.7053 \quad z_2 = 0.9503 \quad z_3 = 0.1353$$

Hallamos la ecuación característica deseada a partir de los polos deseados

$$\Delta(z) = z^3 - 1.7910z^2 + 0.8944z - 0.0907$$

Evaluamos la ecuación característica en \hat{A}_d

$$\Delta(\hat{A}_d) = \hat{A}_d^3 - 1.8303\hat{A}_d^2 + 0.9594\hat{A}_d - 0.1170$$

$$\Delta(\hat{A}_d) = \begin{bmatrix} 0.010898 & 0.000592 & 0.000071 \\ 0 & -0.009794 & 0.019334 \\ 0 & 0 & -0.090717 \end{bmatrix} \quad (4.7)$$

Cálculo de las ganancias usando la fórmula de Ackermann

$$\hat{G} = [0 \quad 0 \quad 1](Co)^{-1} \Delta(\hat{A}_d) \quad (4.8)$$

Reemplazando las ecuaciones (4.3) y (4.7) en (4.8), se obtiene:

$$G = [44.4004 \quad 2.5395 \quad 0.0485]$$

Factor de corrección para la ganancia de la componente integrativa.

$$[G \quad G_I] = [G + [0 \quad 0 \quad 1]] \begin{bmatrix} A_d - I & B_d \\ CA_d & CB_d \end{bmatrix}^{-1}$$

Resolviendo finalmente se obtienen las ganancias del controlador por ubicación de polos:

$$[G \quad G_I] = \begin{bmatrix} \underbrace{554.2894}_{G(1)} & \underbrace{3.5772}_{G(2)} & \underbrace{47.8544}_{G(3)} \end{bmatrix}$$

4.4. Implementación del algoritmo de control

4.4.1. Controlador por ubicación de polos

Se realizó la implementación del algoritmo para su posterior simulación.

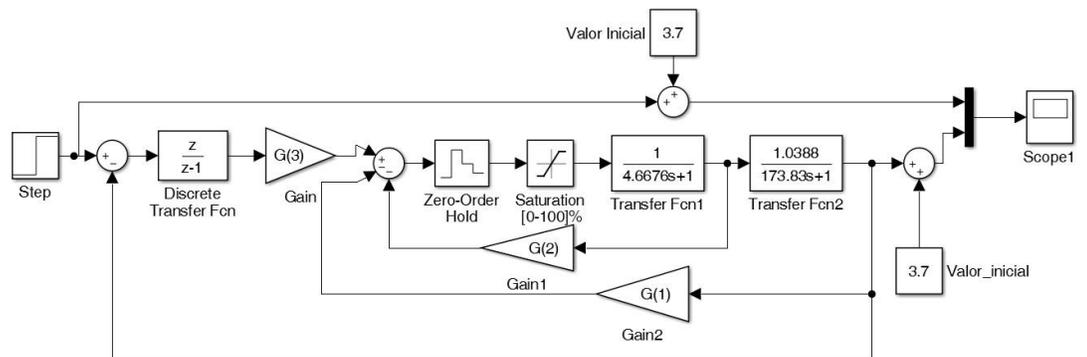


Figura 4.2 Implementación del control por ubicación de polos en Simulink

4.5. Simulación de la planta a lazo cerrado

4.5.1. Control por ubicación de polos

Se simuló para distintos valores del coeficiente de amortiguamiento, a priori sabemos que la planta es sobre amortiguada ($\zeta > 1$)

- $\zeta = 1.5$

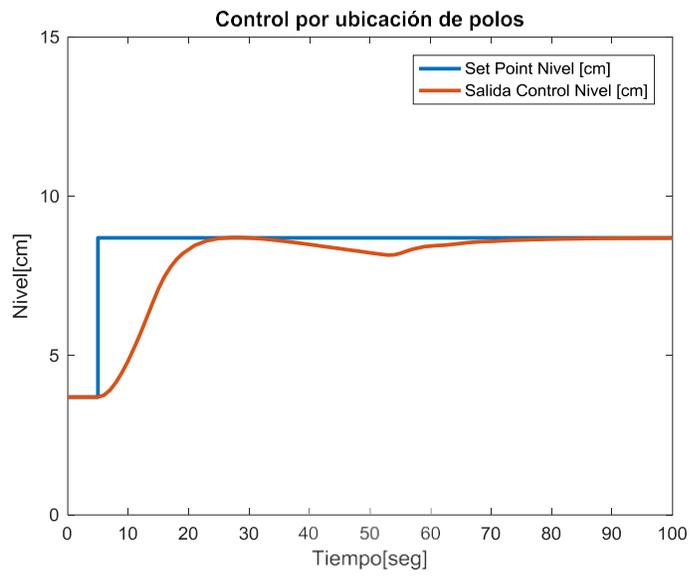


Figura 4.3 Control por ubicación de polos para $\zeta = 1.5$

- $\zeta = 1.8$

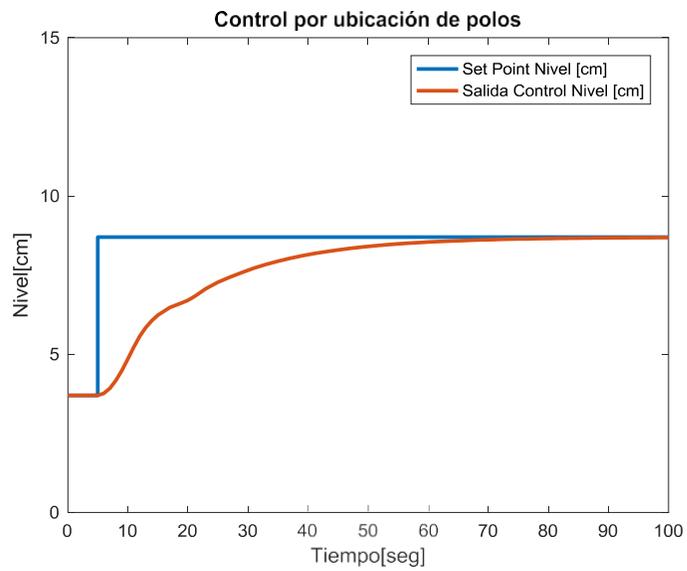


Figura 4.4 Control por ubicación de polos para $\zeta = 1.8$

- $\zeta = 2.2$

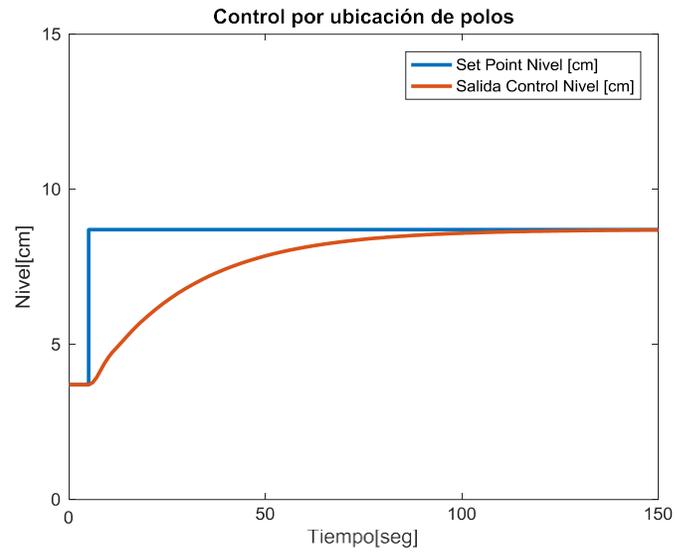


Figura 4.5 Control por ubicación de polos para $\zeta = 2.2$

CAPITULO 5

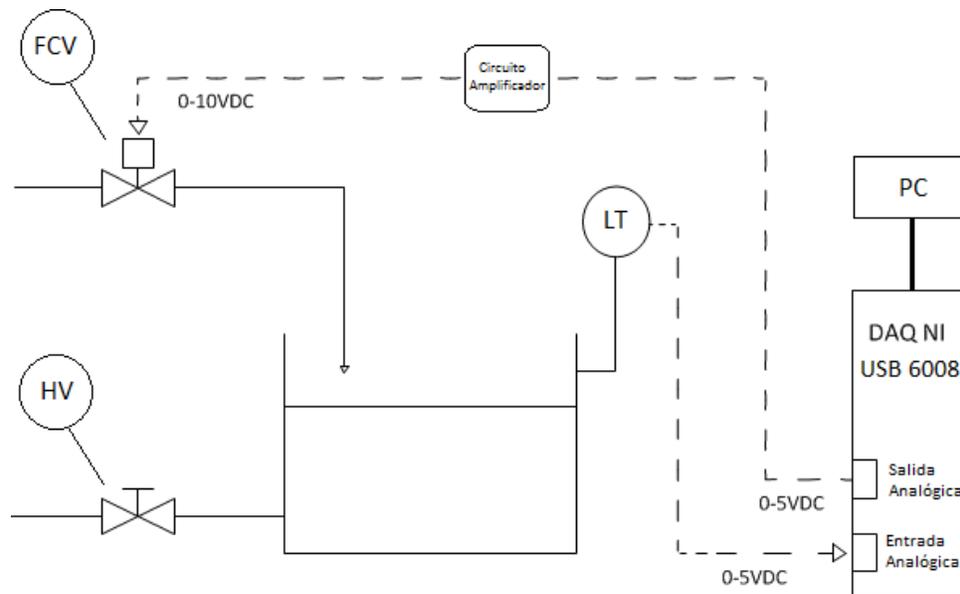
IMPLEMENTACIÓN DEL SISTEMA DE CONTROL

La implementación del sistema de control PID para el proceso tanque con agua requiere llevar a cabo tareas de hardware y software. El algoritmo de control se realizó en el entorno gráfico de programación LabView, que pertenece a National Instruments.

5.1. Hardware del sistema

En la Figura 5.1, se muestra la estructura del sistema usado para el control de nivel. La medición de nivel se logra empleando un sensor ultrasónico Allen Bradley de rango 0 a 60cm.

La medición de nivel es realizada a través de una entrada analógica perteneciente a la tarjeta de adquisición de datos NI-6008 de National Instruments, en un rango de 0 - 5V proveniente del sensor ultrasónico.



Leyenda:

- FCV: Válvula de regulación de flujo de entrada al tanque con accionamiento eléctrico.
- HV: Válvula de descarga con accionamiento manual.
- LT: Transmisor de nivel.

Figura 5.1 Sistema de control para el proceso tanque con agua

El algoritmo de control realizado en el computador procesa la medición de nivel y genera una señal de control. Dicha señal es transmitida a través de una salida analógica de la DAQ NI-6008, cuyo rango es de 0 a 5V, sin embargo la válvula proporcional eléctrica Danfoss EV-260B necesita una señal de excitación de 0 a 10V, es por ello que agregamos un circuito de amplificación de señal.



Figura 5.2 Implementación del sistema tanque con agua



Figura 5.3 Tablero de control, computador y tarjeta de adquisición de datos

5.2. Desarrollo del software

La aplicación desarrollada comprende la interfaz de operador en modo manual y automático, además el algoritmo de control elaborado en LabView, que debido a su gran tamaño, se mostrará en porciones.

- Modo manual

En la Figura 5.4, se muestra la interfaz desarrollada para el modo manual, podemos manipular el porcentaje de apertura de la válvula proporcional. Además, de forma general, en la interfaz podemos observar de manera gráfica el nivel medido y un gráfico de tendencias.

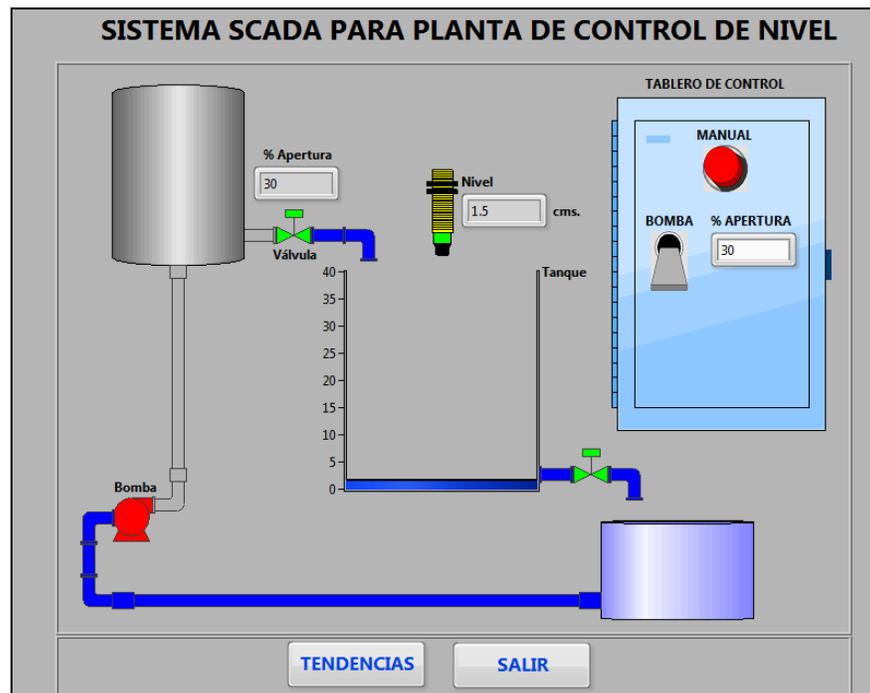


Figura 5.4 Interfaz de usuario en modo manual

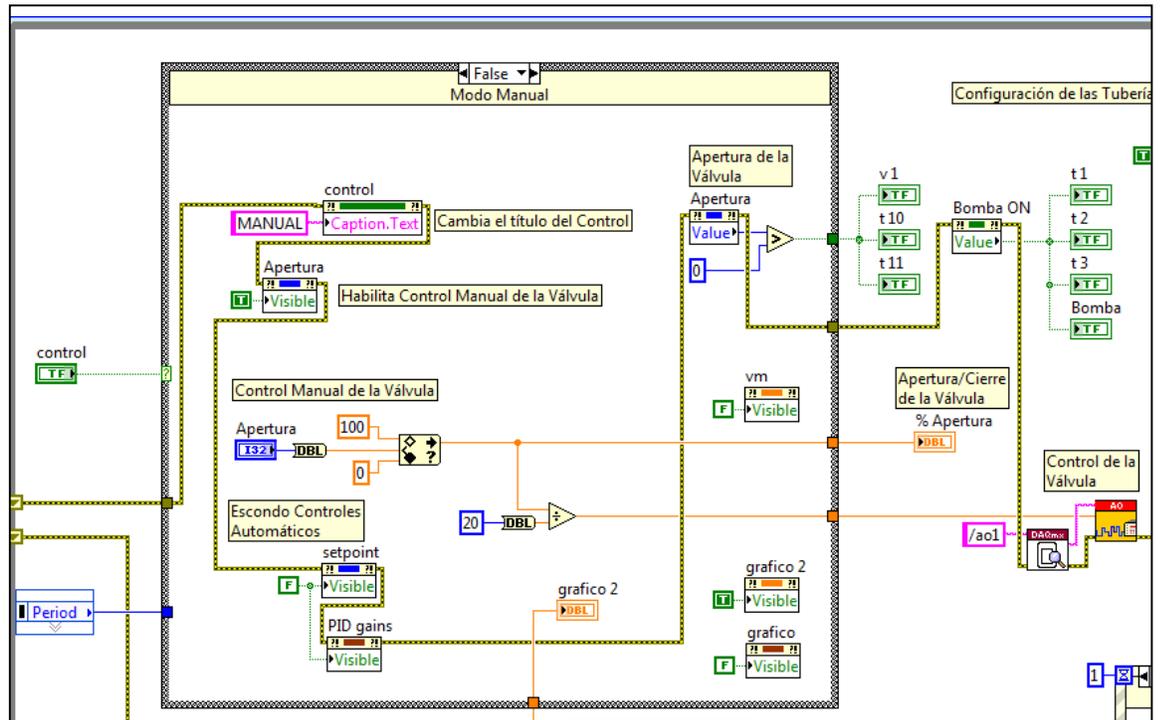


Figura 5.5 Algoritmo de control para el modo manual

- Modo automático

En la Figura 5.6, se muestra la interfaz desarrollada para el modo automático, a diferencia del modo manual, se agrega el valor de referencia de nivel a controlar y la configuración de los parámetros para el controlador PID.

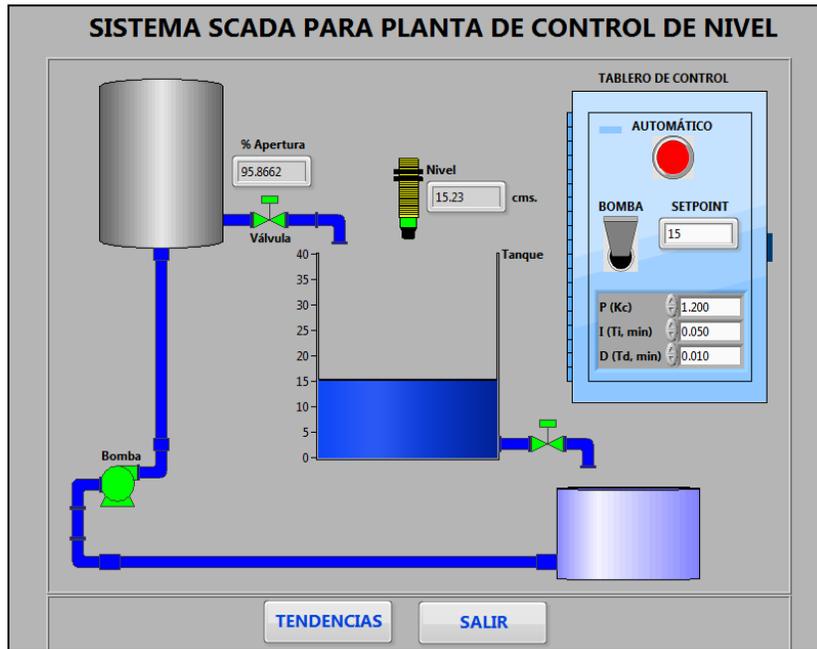


Figura 5.6 Interfaz de usuario en modo automático

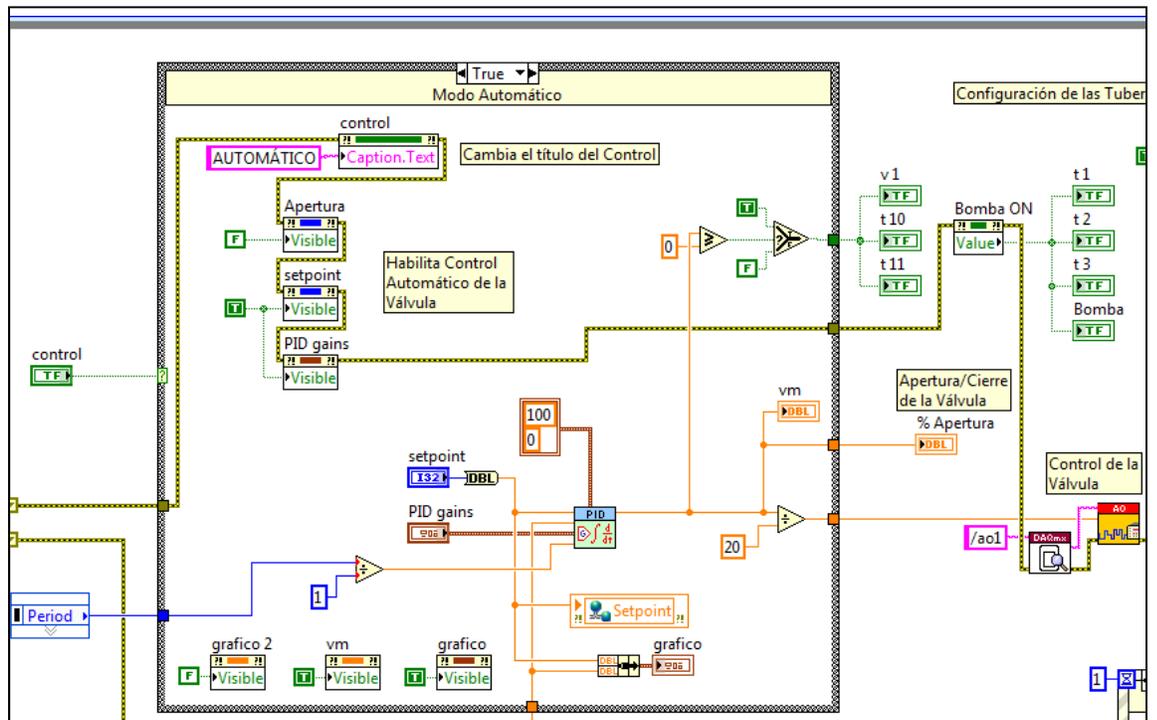


Figura 5.7 Algoritmo de control para el modo automático

Además, también se tiene la implementación en el algoritmo del escalamiento de la medición del sensor de nivel:

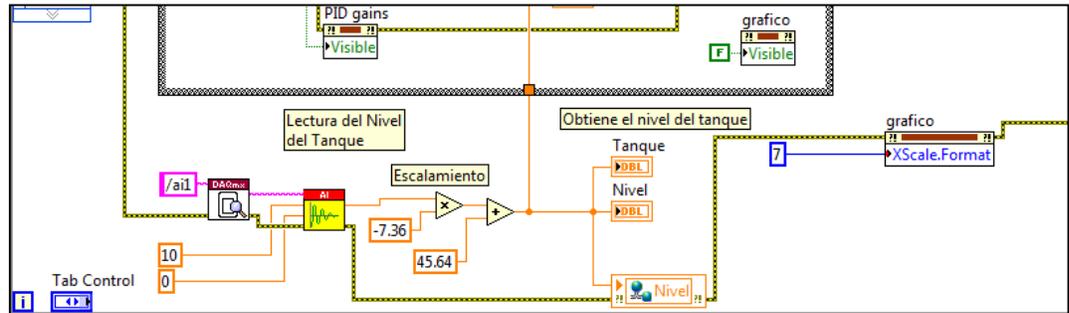


Figura 5.8 Algoritmo de control para el escalamiento de nivel

5.3. Pruebas y resultados

Para realizar las pruebas se ha usado distintos puntos de referencia, con el fin de observar el comportamiento del controlador en los distintos rangos permitidos en la planta (4 a 30cm).

Fijaremos los parámetros del controlador PID obtenidos en la sección 4.2:

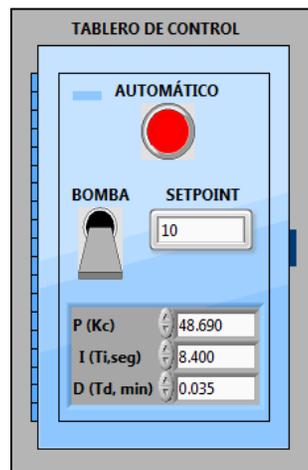


Figura 5.9 Parámetros del controlador PID a utilizar

- Control de nivel en 8cm

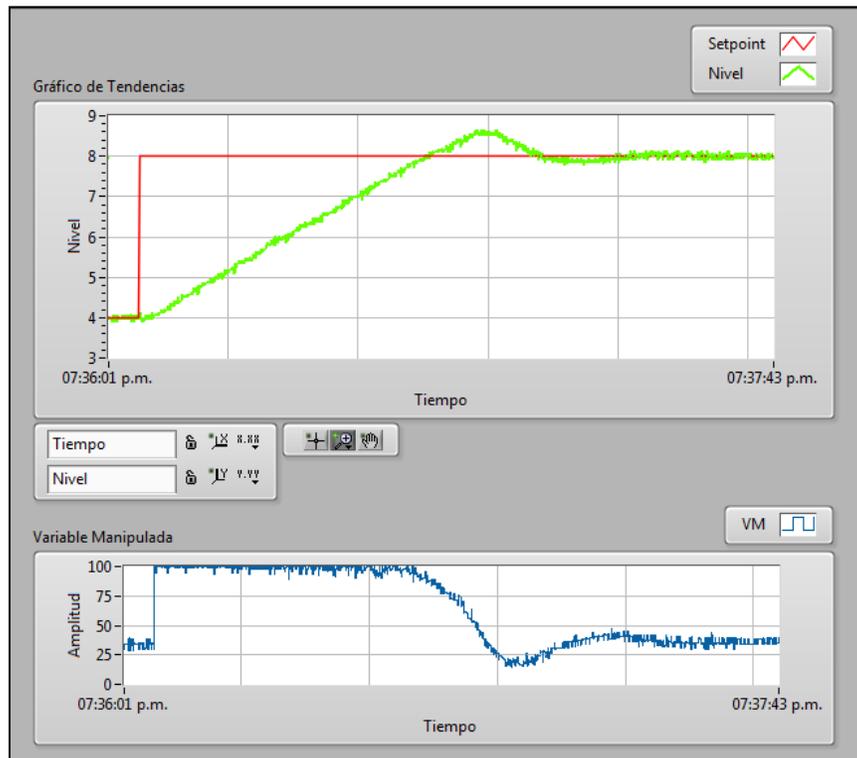


Figura 5.10 Control de nivel en 8cm tomando como referencia 4cm

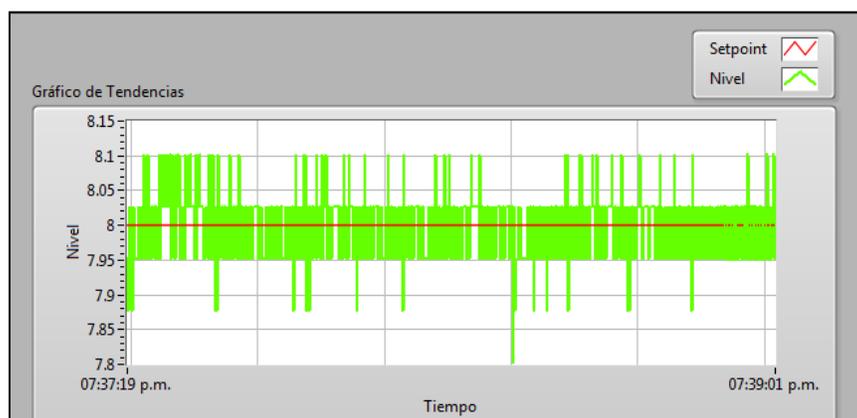


Figura 5.11 Error estacionario al controlar en 8cm

- Control de nivel en 14cm

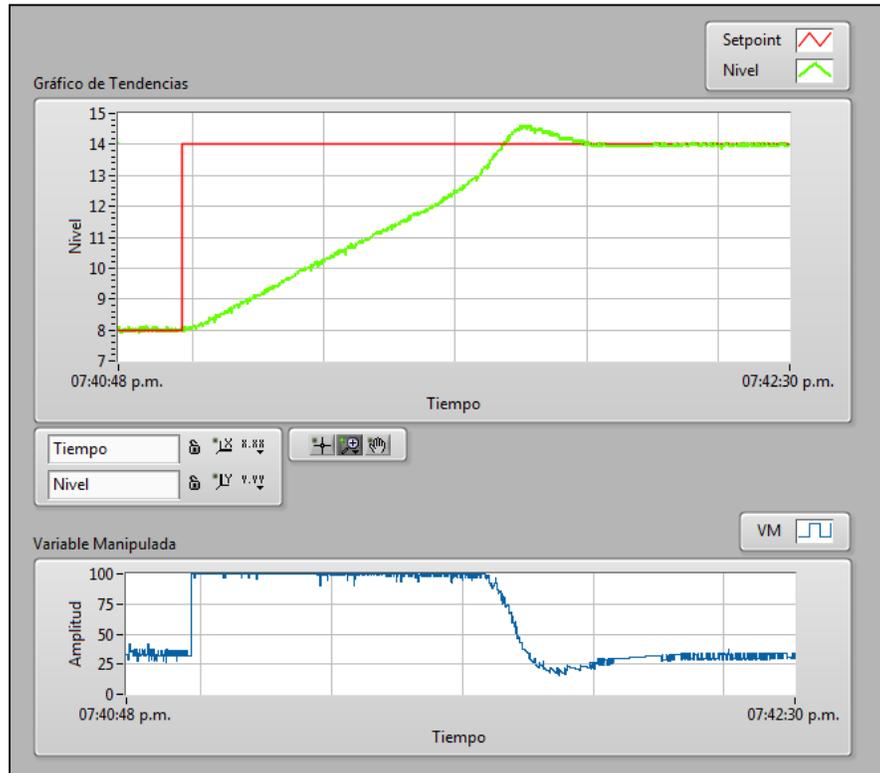


Figura 5.12 Control de nivel en 14cm tomando como referencia 8cm

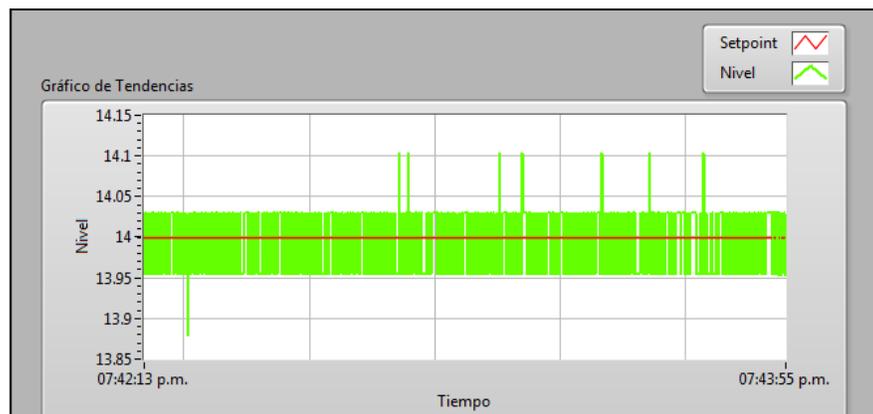


Figura 5.13 Error estacionario al controlar en 14cm

- Control de nivel en 20cm

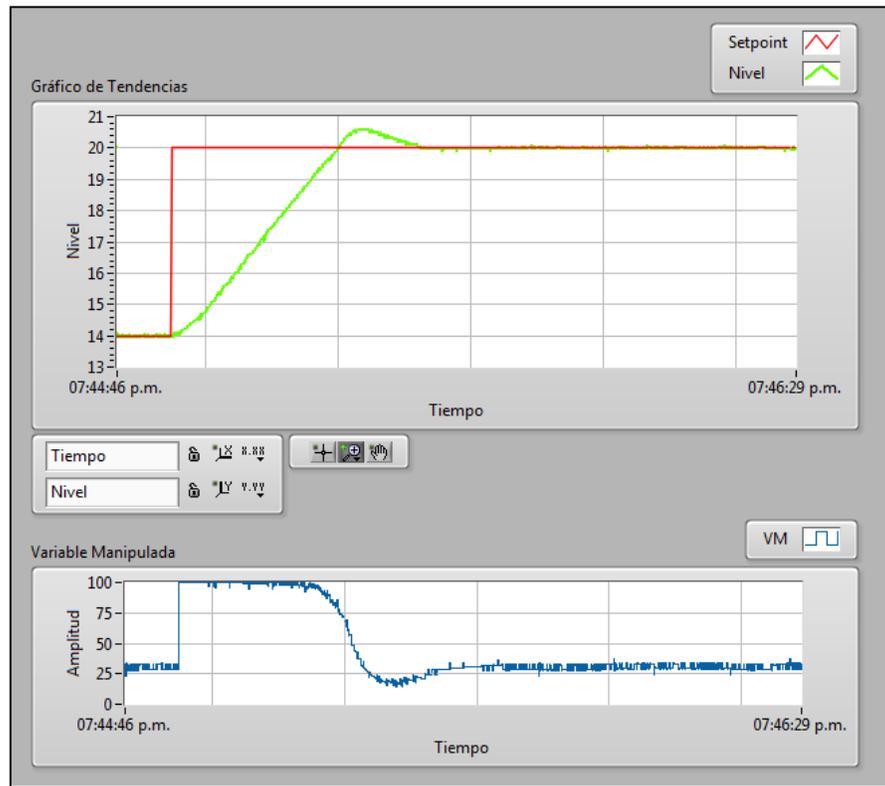


Figura 5.14 Control de nivel en 20cm tomando como referencia 14cm

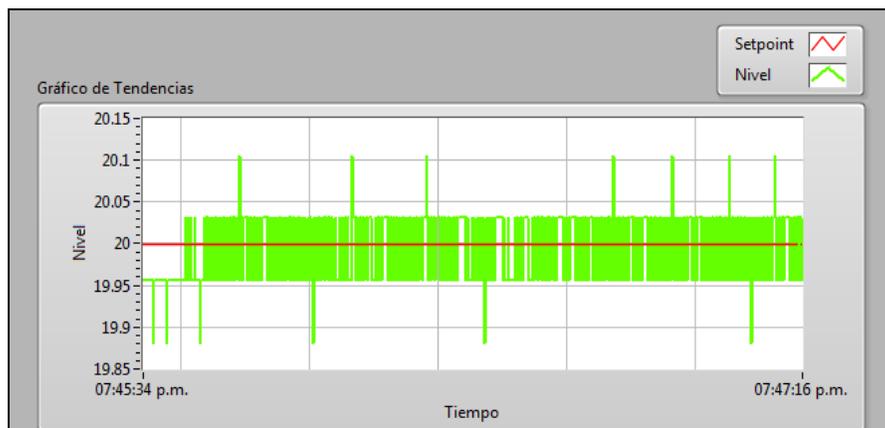


Figura 5.15 Error estacionario al controlar en 20cm

- Control de nivel en 25cm

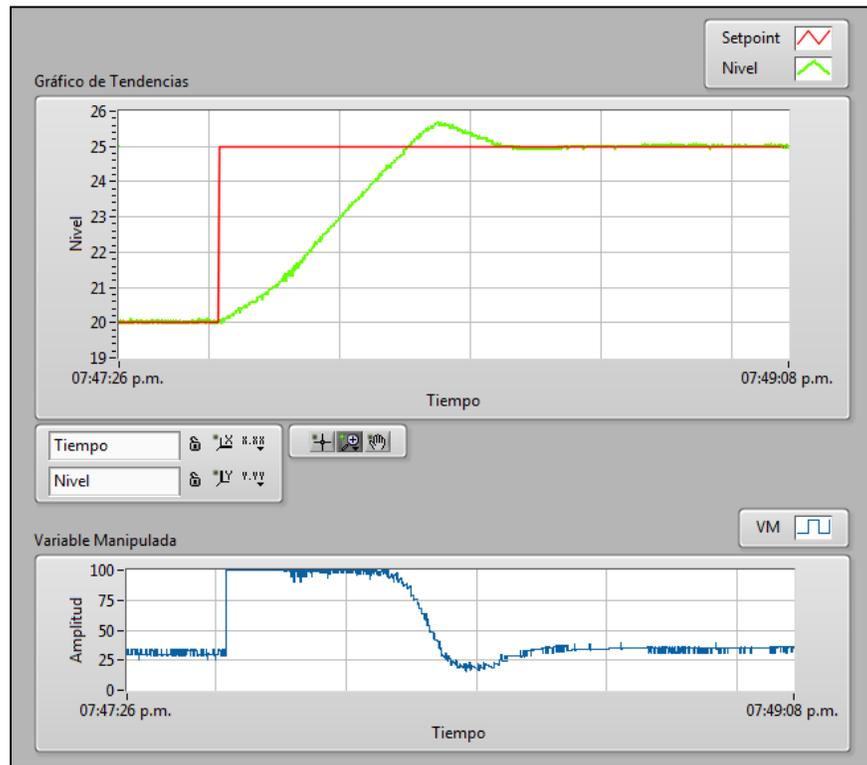


Figura 5.16 Control de nivel en 25cm tomando como referencia 20cm

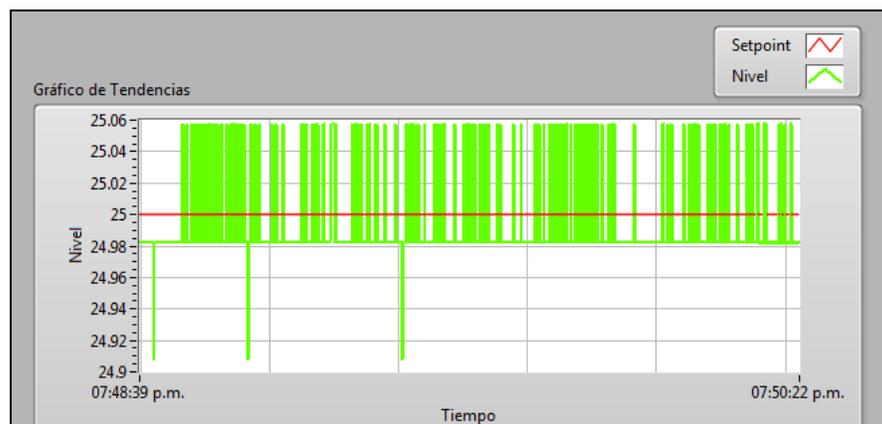


Figura 5.17 Error estacionario al controlar en 25cm

- Control de nivel en 30cm

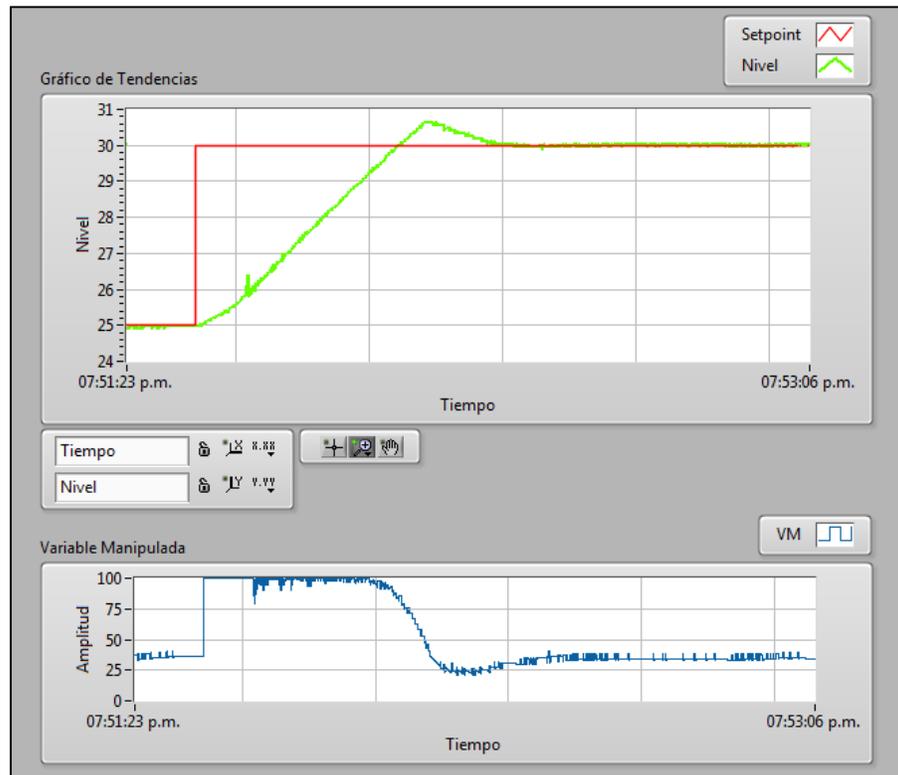


Figura 5.18 Control de nivel en 30cm tomando como referencia 25cm

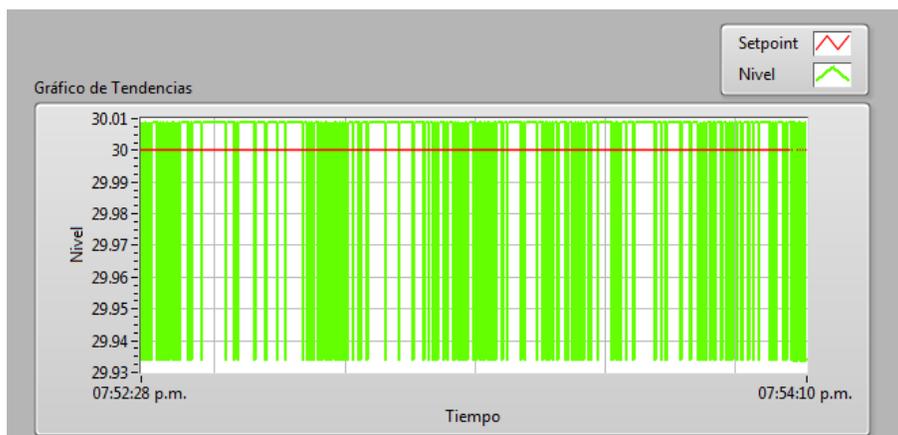


Figura 5.19 Error estacionario al controlar en 30cm

CONCLUSIONES

A fin de controlar el nivel del líquido, un controlador PID convencional ha sido implementado en la plataforma LabView, garantizando un error estable menor a 1mm. Los mejores modelos se obtuvieron usando el método de identificación no paramétrica modelo de 1er Orden usando el Toolbox de Identificación de Matlab y el método de identificación paramétrica Modelo ARMAX de Orden 1.

Se ha demostrado experimentalmente el desempeño del controlador con los parámetros hallados mediante el método de sintonía de Ziegler-Nichols con el modelo de 1er Orden hallado con el Toolbox de identificación de sistemas, al que se le introdujo los datos obtenidos al excitar a la planta.

Se realizó el diseño del controlador usando el método de diseño de polos, obteniendo un buen controlador.

RECOMENDACIONES

Una mala estructura del modelo no ofrece un buen modelo final, independientemente de la cantidad y calidad de datos adquiridos. Por lo tanto, se recomienda realizar una buena toma y tratamiento de datos antes de generar el modelo de la planta.

Es de mucha importancia, tener en cuenta los tipos de señales y rangos que provienen de los sensores hacia la tarjeta de adquisición de datos.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Aníbal Arturo Parra Quispe. Tesis de Maestría “Diseño e Implementación de Controladores PID Industriales”. Sección de Posgrado de la Facultad de Electricidad y Electrónica de la UNI. 2007.
- [2] Rojas-Moreno Arturo, and Parra-Quispe Arturo. Design and Implementation of a Water Tank Control System Employing a MIMO PID Controller. Proceedings of the International Multi-Conference on Engineering and Technological Innovation, Orlando, USA, Vol II, July 2008, pp 344 - 349.
- [3] Rubén Darío Aquize Palacios. Tesis de Maestría “Implementación de un Sistema de Control No Lineal Backstepping Multivariable para la Planta Piloto Tanque con Agua”. Sección de Posgrado de la Facultad de Electricidad y Electrónica de la UNI. 2011.
- [4] Katsuhiko Ogata, Ingeniería de Control Moderna. Pearson Educación, 5ta edición, 2010, pp 101-104, 568-571.
- [5] W. F. Hughes and J. A. Brighton, Theory and Problems of Fluid Dynamics, Schaum’s outlines series, McGraw-Hill, 2nd edition, 1991.
- [6] Elamari A Elayan. Design of Heuristic Fuzzy Logic Controller for Liquid Level Control. Fifth International Conference on Intelligent Systems, Modelling and Simulation, 2014.
- [7] Abdelelah K., and Hussam H. Design Fuzzy Logic Controller for Liquid Level Control. International Journal of Emerging Science and Engineering (IJESE) ISSN: 2319–6378, Vol. 1, Issue 11, September 2013.

- [8] Zhu Jian – Jun. Design of Fuzzy Control System for Tank Liquid Level Based on WinCC and Matlab. 13th International Symposium on Distributed Computing and Applications to Business, Engineering and Science, 2014.
- [9] Rajesh.T, Arun jayakar.S, and Siddharth.S.G. Design and Implementation Of IMC Based PID Controller for Conical Tank Level Control Process. International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering, Vol. 2, Issue 9, September 2014.
- [10] Laith A., and Hussein A. Implementation of Fuzzy and PID Controller to Water Level System using LabView. International Journal of Computer Applications (0975 – 8887), Vol. 116 – No. 11, April 2015.
- [11] Eloy Acero Coila. Tesis de Maestría “Diseño de un Controlador Adaptivo con Identificación en Línea Aplicado a una Planta Modelo de Presión”. Escuela de Posgrado - PUCP. 2006.
- [12] Rachad S., Nsiri B. and Bensassi B. System Identification of Inventory System Using ARX and ARMAX Models. International Journal of Control and Automation, Vol. 8 – No.12, 2015.

APÉNDICE

A. Especificaciones técnicas de la DAQ USB-6008 de National Instruments

Dispositivo DAQ Multifunción de Bajo Costo de 12 Bits, 10 kS/s

- 8 entradas analógicas (12 bits, 10 kS/s)
- 2 salidas analógicas estáticas (12 bits); 12 E/S digitales; contador de 32 bits
- Energizado por bus para una mayor movilidad, conectividad de señal integrada
- La versión OEM está disponible
- Compatible con LabVIEW, LabWindows™/CVI y Measurement Studio para Visual Studio .NET
- Considere el USB-6001 para resolución de 14 bits y velocidades de muestreo más altas.



Información General

El USB-6008 brinda funcionalidad DAQ básica para aplicaciones como registro de datos simple, medidas portátiles y experimentos académicos de laboratorio. Es accesible para uso de estudiantes, pero lo suficientemente poderoso para aplicaciones de medida más sofisticadas. Utilice el USB-6008 que incluye el software registrador de datos para empezar a tomar medidas básicas en minutos o prográmelo usando LabVIEW o C y el software de servicios de medida NI-DAQmx base para un sistema de medida personalizado.

Para cursos suplementarios con experimentos prácticos sobre teoría de simulación, medida y automatización, NI desarrolló un Kit de Estudiante USB-6008 que incluye una copia de LabVIEW Edición de Estudiante. Estos paquetes son exclusivamente para estudiantes, proporcionándoles una herramienta de aprendizaje potente, práctica y a bajo costo.

Los dispositivos DAQ USB son compatibles con las siguientes versiones (o posteriores) de software de aplicación de NI: LabVIEW 7.x, LabWindows/CVI 7.x o Measurement Studio 7.x. Los módulos DAQ USB también son compatibles con Visual Studio .NET, C/C++ y Visual Basic 6.0.

Resumen de Especificaciones

General	
Producto	USB-6008
Familia de Productos	DAQ Multifunción
Tipo de Medida	Voltaje
Factor de Forma	USB
Sistema Operativo / Objetivo	Linux, Mac OS, Pocket PC, Windows
Compatible con RoHS	Sí
Tipo de Aislamiento	Ninguno
Entrada Analógica	
Canales de una sola terminal	8
Canales Diferenciales	4
Resolución de Entrada Analógica	12 bits
Rango de Voltaje Máximo	
Rango	-10 V - 10 V
Precisión	7.73 mV
Rango de Voltaje Mínimo	
Rango	-1 V - 1 V
Precisión	37.5 mV
Número de Rangos	8

General	
Muestreo Simultáneo	No
Memoria Interna	512 B
Salida Analógica	
Número de Canales	2
Resolución	12 bits
Rango de Voltaje Mínimo y Máximo	
Rango	0 V - 5 V
Precisión	7 mV
Razón de Actualización	150 S / s
Capacidad de Corriente Simple	5 mA
Capacidad de Corriente Total	10 mA
E / S Digital	
Canales Bidireccionales	12
Canales de Entrada Únicamente	0
Canales de Salida Únicamente	0
Temporización	Software
Niveles Lógicos	TTL
Filtros de Entrada Programables	No
¿Soporta Estados de Encendido Programables?	No
Entrada Digital	
Tipo de Entrada	Sinking, Sourcing
Rango de Voltaje Máximo	0 V - 5 V
Salida Digital	
Tipo de Salida	Sinking, Sourcing
Capacidad de Corriente Simple	8.5 mA
Capacidad de Corriente Total	102 mA

General	
Rango de Voltaje Máximo	0 V - 5 V
Contadores / Temporizadores	
Temporizador Watchdog	No
Contadores	1
Operaciones a Búfer	No
Debouncing / Glitch Removal	No
Frecuencia Máx. de la Fuente	5 MHz
Generación de Pulso	No
Tamaño	32 bits
Estabilidad de Tiempo	50 ppm
Niveles Lógicos	TTL
Temporización / Disparo / Sincronización	
Disparo	Digital
Bus de Sincronización (RTSI)	No
Especificaciones Físicas	
Longitud	8.51 cm
Ancho	8.18 cm
Altura	2.31 cm
Conector de E / S	Terminales de tornillo
Potencia USB	Energizado por Bus

B. Conceptos básicos – Estimación utilizando mínimos cuadrados recursivos con factor de olvido

- Regresiones Lineales

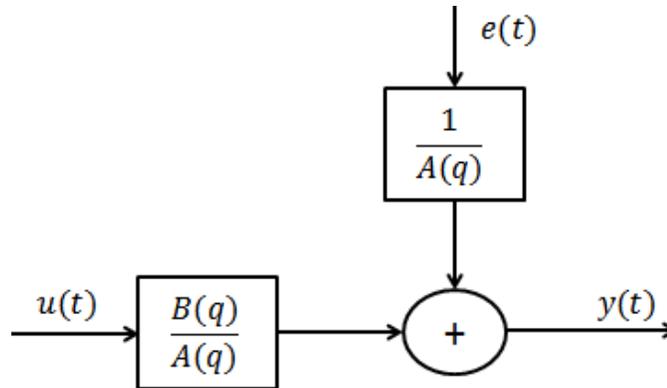


Figura 7. Estructura de un modelo ARX

Para un modelo ARX como el de la Figura 19, podemos calcular el predictor utilizando:

$$\hat{y}(t|\theta) = B(q)u(t) + [1 - A(q)]y(t)$$

Esta expresión puede expandirse utilizando las siguientes igualdades:

$$A(q) = 1 + a_1q^{-1} + a_2q^{-2} + \dots + a_naq^{-na}$$

$$B(q) = b_1q^{-1} + b_2q^{-2} + \dots + b_n bq^{-nb}$$

Esto da:

$$y(t) + a_1y(t-1) + a_2y(t-2) + \dots + a_nay(t-na) = b_1u(t-1) + b_2u(t-2) + \dots + b_nbu(t-nb)$$

Como vemos, esta es una expansión lineal de los parámetros de A y B utilizando como coeficientes las entradas y salidas pasadas.

Haciendo uso del álgebra lineal, podemos entonces definir los siguientes vectores:

$$\varphi(t) = [-y(t-1) \quad \dots \quad -y(t-a) \quad u(t-1) \quad \dots \quad u(t-nb)]^T$$

$$\theta = [a_1 \quad \dots \quad a_{na} \quad b_1 \quad \dots \quad b_{nb}]^T$$

Los cuales permiten escribir la expansión lineal en la forma reducida:

$$\hat{y}(t|\theta) = \varphi^T(t)\theta = \theta^T \varphi(t)$$

Es decir, el predictor es el producto escalar entre un vector de datos conocidos y un vector de parámetros desconocidos. Este modelo se conoce como modelo de regresión lineal. Aquí, el vector de datos $\varphi(t)$ se conoce como vector de regresión, siendo $\theta(t)$ el vector de parámetros a encontrar.

La expresión de regresión lineal en forma reducida ha sido hallada para el modelo ARX que es simple. Sin embargo, podemos generalizar esta expresión para estructuras más generales. Por ejemplo, analizaremos el caso del modelo ARMAX de la figura 20 que también se utilizará.

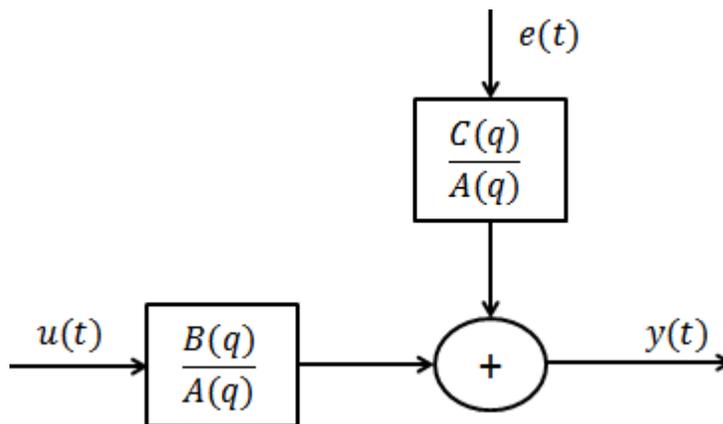


Figura 8. Estructura del Modelo ARMAX

El predictor ahora se calcula como:

$$C(q)\hat{y}(t|\theta) = B(q)u(t) + [C(q) - A(q)]y(t)$$

Debido a que $e(t)$ es ruido blanco, no podemos medirlo ni estimarlo. Estos valores son necesarios, como veremos, para poder hallar los parámetros (θ). Sin embargo, podemos aproximar esta señal a través del error de predicción, el cual tiene características similares. Definiendo el error de predicción como:

$$\varepsilon(t, \theta) = y(t) - \hat{y}(t|\theta)$$

Podemos deducir nuevamente esta expresión a un modelo de regresión pseudo-lineal:

$$\hat{y}(t|\theta) = \varphi^T(t, \theta)\theta + e(t)$$

Donde:

$$\varphi(t) = [-y(t-1) \cdots -y(t-na) \ u(t-1) \cdots u(t-nb) \ \varepsilon(t-1, \theta) \cdots \varepsilon(t-nc, \theta)]^T$$

$$\theta(t) = [a_1 \cdots a_{na} \ b_1 \cdots b_{nb} \ c_1 \cdots c_{nc}]^T$$

Aquí se utiliza el término pseudo-lineal ya que el vector de regresiones no es función únicamente del tiempo, sino también de los parámetros del modelo.

- Estimación de parámetros

Debemos hallar el argumento (θ) que minimice una determinada función de costos del error de predicción.

El error de predicción $\varepsilon(t, \theta)$, casi siempre contiene componentes de ruido de alta frecuencia que no son esenciales para el modelamiento del sistema, y que pueden ser filtradas utilizando el filtro $L(q)$ como se muestra a continuación:

$$\varepsilon_F(t, \theta) = L(q)\varepsilon(t, \theta) \quad 1 \leq t \leq N$$

Una vez filtrado el error de predicción definimos la siguiente función de costos:

$$V_N(\theta, Z^N) = \frac{1}{N} \sum_{t=1}^N \ell(\varepsilon_F(t, \theta))$$

Donde:

$$Z^N = [y(1) \quad u(1) \quad y(2) \quad u(2) \quad \cdots \quad y(N) \quad u(N)]$$

es el vector que contiene todos los datos experimentales de entrada y salida, y $\ell(\cdot)$ es un función escalar por lo general positiva. Esta función define formalmente el “tamaño total del error de predicción”. Como se deduce entonces, si minimizamos esta función, el modelo candidato se aproximará más a la función de transferencia real del sistema. Los parámetros estimados quedan definidos entonces como:

$$\hat{\theta}_N = \hat{\theta}_N(Z^N) = \operatorname{argmin} V_N(\theta, Z^N)$$

Donde argmin significa el “argumento (θ) que minimiza la función V_N ”.

Por lo general, la función $\ell(\cdot)$ es una norma cuadrática. Es decir:

$$\ell(\varepsilon) = \frac{1}{2} \varepsilon^2$$

La elección de esta función cuadrática y del filtro $L(q) = 1$, constituye el *criterio de mínimos cuadrados*. Aplicando esta definición junto con el concepto de regresión lineal, tenemos:

$$V_N(\theta, Z^N) = \frac{1}{N} \sum_{t=1}^N \frac{1}{2} [y(t) - \hat{y}(t|\theta)]^2 = \frac{1}{N} \sum_{t=1}^N \frac{1}{2} [y(t) - \varphi^T(t)\theta]^2$$

Esta función puede ser minimizada analíticamente, lo que nos da como resultado el vector de parámetros estimados:

$$\theta = \underset{N}{\operatorname{argmin}} V(\theta, Z^N) = \left[\sum_{t=1}^N \varphi(t) \varphi^T(t) \right]^{-1} \sum_{t=1}^N \varphi(t) y(t)$$

Si es posible el cálculo del término de la matriz inversa, esta expresión constituye la *Estimación por Mínimos Cuadrados (LSE)*. Si el número total de parámetros a estimar es “d”, entonces la dimensión del vector de parámetros θ será $d \times 1$. De igual forma podemos definir la matriz $d \times d$:

$$R(N) = \frac{1}{N} \sum_{t=1}^N \varphi(t) \varphi^T(t)$$

Y el vector de $d \times 1$:

$$f(N) = \frac{1}{N} \sum_{t=1}^N \varphi(t) y(t)$$

Con lo que podemos reducir $\hat{\theta}$ a la expresión:

$$\hat{\theta} = R^{-1}(N) f(N)$$

Un caso especial de LSE, se produce cuando se pondera, o se asigna diferentes pesos a las distintas mediciones utilizando una secuencia de ponderación $\beta(N, t)$

$$V_N(\theta, Z^N) = \sum_{t=1}^N \beta(N, t) [y(t) - \varphi^T(t)\theta]^2$$

La expresión para la estimación entonces es:

$$\hat{\theta} = \left[\sum_{t=1}^N \beta(N, t) \varphi(t) \varphi^T(t) \right]^{-1} \sum_{t=1}^N \beta(N, t) \varphi(t) y(t)$$

Esta expresión utiliza los N valores adquiridos para la identificación. De forma similar, podemos definir la estimación $\hat{\theta}_t$ para un tiempo t cualquiera:

$$\hat{\theta} = \left[\sum_{k=1}^t \beta(t, k) \varphi(k) \varphi^T(k) \right]^{-1} \sum_{k=1}^t \beta(t, k) \varphi(k) y(k) \quad \dots \quad (1)$$

$$\hat{\theta} = R^{-1}(t) f(t)$$

Con:

$$R(t) = \sum_{k=1}^t \beta(t, k) \varphi(k) \varphi^T(k)$$

$$f(t) = \sum_{k=1}^t \beta(t, k) \varphi(k) y(k)$$

- Estimación en Línea utilizando Mínimos Cuadrados Recursivos con Factor de Olvido

La utilización de un método de estimación recursivo permite identificar un modelo “en línea”, es decir, mientras la planta o el sistema se encuentre en operación.

Para poder utilizar el método de mínimos cuadrados de forma recursiva, es útil definir la secuencia de ponderación $\beta(t, k)$ como:

$$\beta(t, k) = \lambda(t) \beta(t-1, k) \quad 1 \leq k \leq t-1$$

$$\beta(t, t) = 1$$

Observemos entonces, que podemos escribir:

$$R(t) = \lambda(t) R(t-1) + \varphi(t) \varphi^T(t)$$

$$f(t) = \lambda(t) f(t-1) + \varphi(t) y(t)$$

Reemplazando expresiones en (1), y después de algún reordenamiento y manipulaciones algebraicas, llegamos a:

$$\hat{\theta} = \hat{\theta}_{-1} + R^{-1}(t)\varphi(t)[y(t) - \varphi^T(t)\hat{\theta}_{-1}]$$

$$R(t) = \lambda(t)R(t-1) + \varphi(t)\varphi^T(t) \quad \dots \quad (2)$$

Este es el algoritmo Recursivo de Mínimos Cuadrados (RLSE). En esta ecuación, las condiciones iniciales de $R(t-1)$ y $\hat{\theta}_{-1}$ se asumen y por tanto son conocidas. Los coeficientes de la secuencia de ponderación $\lambda(t)$ se eligen y por tanto son conocidos. El vector de regresiones está siendo actualizado permanentemente con los datos adquiridos, por lo que también se conoce. Es decir, que en cada instante de tiempo, podemos calcular la matriz $R(t)$, la cual invertimos y con ese resultado hallamos $\hat{\theta}$.

Ahora, el cálculo de $R^{-1}(t)$, puede ser computacionalmente costoso si se realiza en cada instante de muestreo. Entonces, es necesario el uso del siguiente lemma de inversión de matrices:

$$[A + BCD]^{-1} = A^{-1} - A^{-1}B[DA^{-1}B + C^{-1}]DA^{-1}$$

Si reemplazamos $A = \lambda(t)R(t-1)$, $B = D^T = \varphi(t)$, $C = 1$, y definimos $P(t) = R^{-1}(t)$,

Entonces tenemos:

$$P(t) = \frac{P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)}}{\lambda(t)} \quad \dots \quad (3)$$

Para el cálculo de (2) necesitamos $R^{-1}(t)\varphi(t)$, es decir:

$$L(t) = R^{-1}(t)\varphi(t) = \frac{P(t-1)\varphi(t)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)} \quad \dots \quad (4)$$

Finalmente, tenemos el siguiente algoritmo RLSE computacionalmente eficiente

$$\hat{\theta} = \hat{\theta}_{-1} + L(t)[y(t) - \varphi^T(t)\hat{\theta}_{-1}] \quad \dots \quad (5)$$

$$L(t) = \frac{P(t-1)\varphi(t)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)}$$

$$P(t) = \frac{1}{\lambda(t)} \left[P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)} \right]$$

Por lo general, el valor inicial de $P(t-1)$ es un valor grande y se asume como 10^6 y el de $\hat{\theta}_{-1}$ como cero si es que no se cuenta con un modelo inicial. Nuevamente, los coeficientes de la secuencia de ponderación $\lambda(t)$ se eligen y por tanto también son conocidos. El vector de regresiones se actualiza permanentemente con los datos adquiridos. Es decir, que en cada instante de tiempo, podemos calcular la matriz $P(t)$ y $L(t)$, y con este resultado hallamos $\hat{\theta}$.

La secuencia de ponderación podemos definirla de tal forma que se asigne un menor peso a las mediciones más antiguas, las cuales ya no son tan representativas para el sistema. Para un sistema que cambia gradualmente y de una forma “estacionaria”, la elección más común para la secuencia de ponderación es:

$$\beta(t, k) = \lambda^{t-k} \quad \text{es decir} \quad \lambda(t) \equiv \lambda$$

Donde λ es una constante llamada “factor de olvido”. Como se aprecia en este caso, las mediciones más antiguas son descartadas exponencialmente. Típicamente el valor de λ se escoge entre 0.98 y 0.995.

C. Implementación del algoritmo recursivo ARX

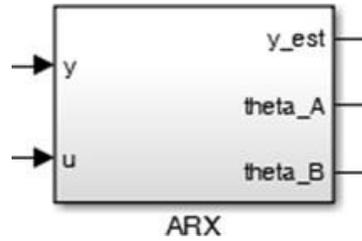


Figura 39. Bloque de Identificación ARX

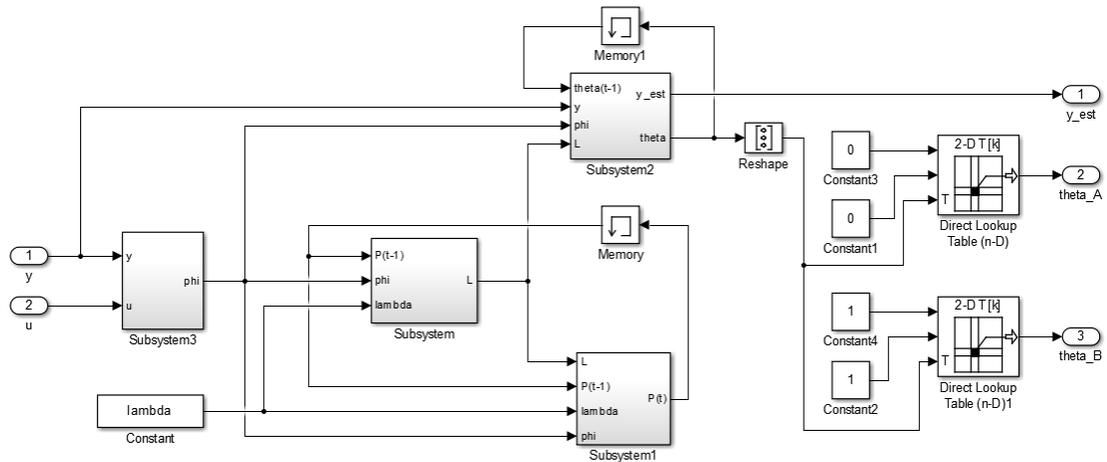


Figura 4. Subsistema ARX

Observaciones:

- El valor inicial de $P(t-1)$ fue establecido a 10^6 según la ventana de configuración del bloque Memory como muestra la siguiente figura:

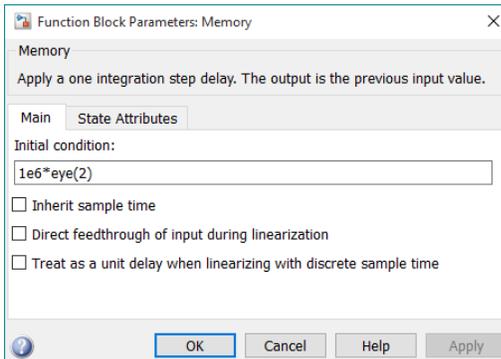


Figura 5. Valor inicial de $P(t-1)$.

- El valor inicial de $\theta(t-1)$ fue establecido a 0 según la ventana de configuración del bloque Memory1 como muestra la siguiente figura:

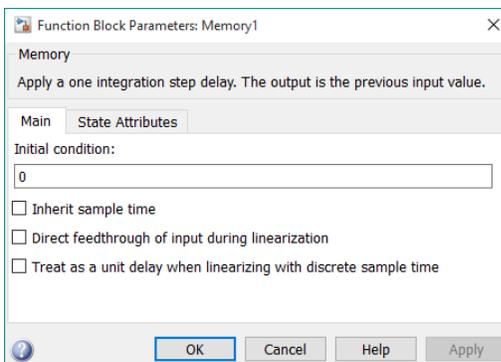


Figura 6. Valor inicial de $\theta(t-1)$

- Para extraer los parámetros del vector $\theta(t)$ se convierte el vector a una tabla 2D mediante los bloques Direct Lookup Table (n-D) y mediante dos constantes en la entrada se especifica la locación del parámetro a mostrar.

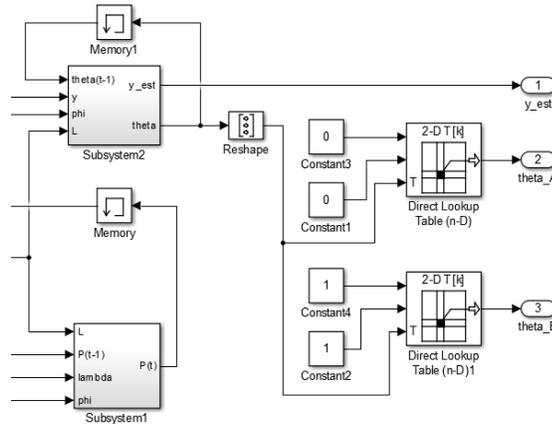


Figura 7. Bloques Direct Lockup Table (n-D) contienen el vector $\theta(t)$.

Bloque Subsystem

Este bloque permite calcular el valor de $L(t)$ según la fórmula:

$$L(t) = \frac{P(t-1)\varphi(t)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)}$$

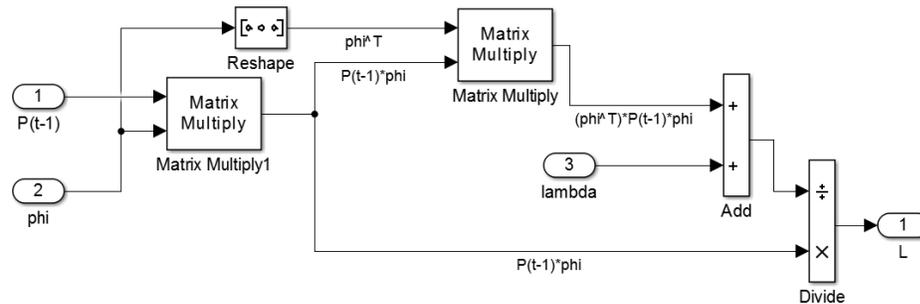


Figura 8. Bloque Subsystem - Calculo de L .

Observaciones:

- La matriz $P(t-1)$ tiene dimensión 2×2 .
- El vector ϕ tiene dimensión 2×1 .
- $L(t)$ tiene dimensión 2×1 .

- El bloque Reshape permite modificar la dimensión de phi, se calcula la transpuesta mediante este bloque.

Bloque Subsystem1

Este bloque permite calcular el valor de P(t) según la fórmula:

$$P(t) = \frac{1}{\lambda(t)} [P(t-1) - L(t)\varphi^T(t)P(t-1)]$$

Con:

$$L(t) = \frac{P(t-1)\varphi(t)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)}$$

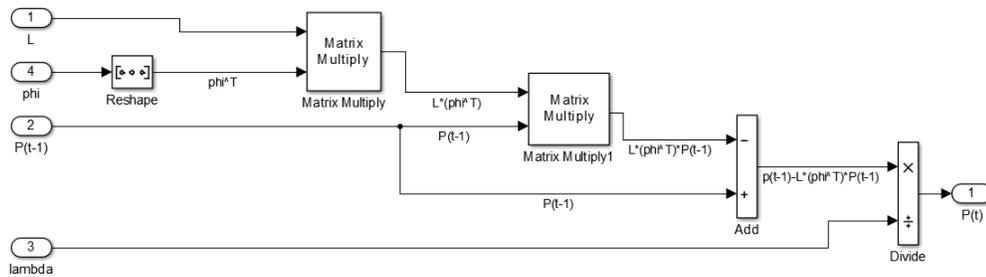


Figura 9. Bloque Subsystem1 - Calculo de P(t).

Observaciones:

- L(t) posee dimensión 2 x 1, phi posee dimensión 2 x 1 y P(t-1) posee dimensión 2 x 2.

Bloque Subsystem2:

Este bloque permite calcular el valor del vector de parámetros theta(t) según la fórmula:

$$\hat{\theta} = \hat{\theta}_{-1} + L(t)[y(t) - \varphi^T(t)\hat{\theta}_{-1}]$$

Además permite hallar el valor estimado de y (y_est) que sirve para estimar el valor de los parámetros.

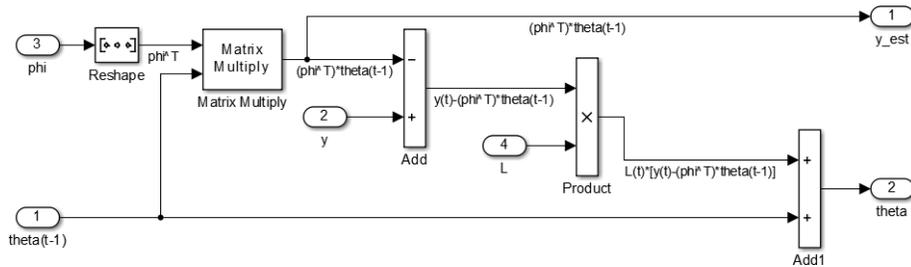


Figura 1010. Bloque Subsystem2 - Calculo de vector $\theta(t)$.

Bloque Subsystem3

Este bloque permite ordenar las muestras anteriores de $u(t)$ y $y(t)$ para formar el vector ϕ , para la simulación se trabajará solo con una planta de primer orden así que el vector ϕ tendrá la siguiente forma:

$$\phi(t) = [-y(t-1) \quad u(t-1)]^T$$

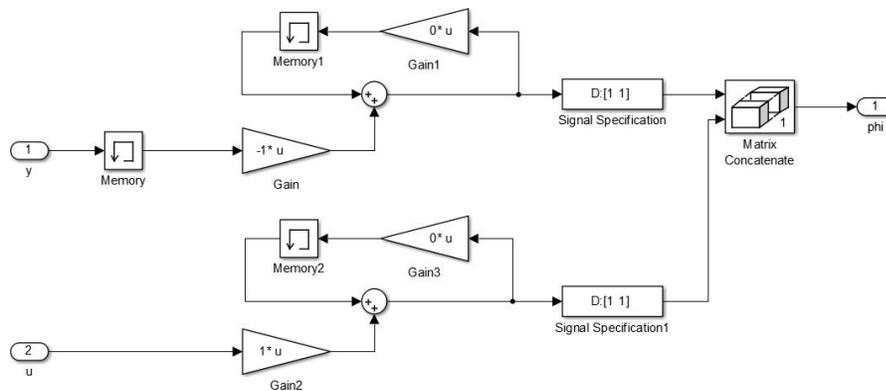


Figura 11. Bloque Subsystem3 - Calculo de ϕ .

Observaciones:

- El bloque Gain posee ganancia negativa mientras que Gain2 la ganancia es positiva. Esto debido a que el vector $\phi(t)$ está compuesto por muestras negativas de $y(t)$ y muestras positivas de $u(t)$.
- El bloque Signal Specification permite a grandes rasgos establecer las características de los datos. Para la simulación por ser de primer orden solo se tomará una muestra anterior por tanto la dimensión de estos es 1×1 .

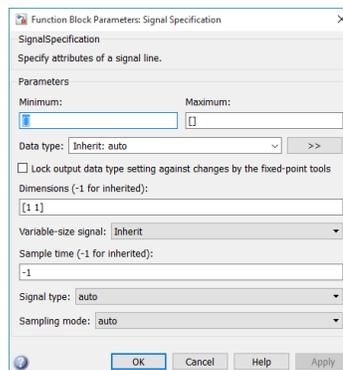


Figura 12. Bloque Signal Specification

- El bloque Matrix Concatenate permite unir vectores para formar una matriz, como las muestras tienen dimensión 1×1 al concatenarlas la dimensión será 2×1 .

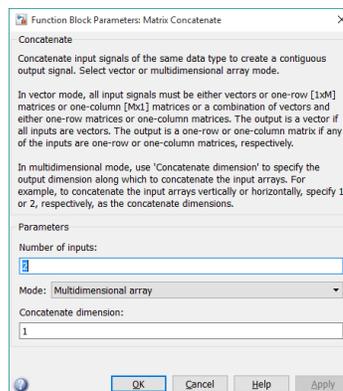


Figura 113. Bloque Matrix Concatenate

D. Implementación del algoritmo recursivo ARMAX

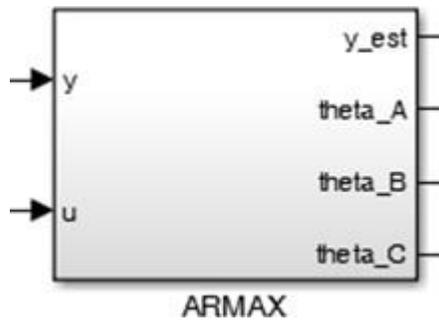


Figura 14. Bloque de identificación ARMAX

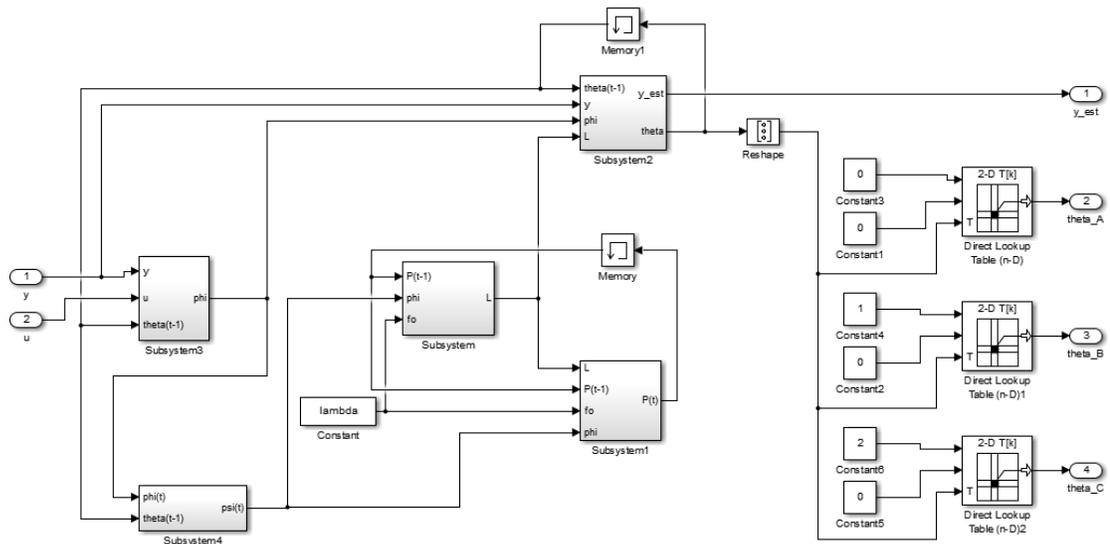


Figura 1512. Subistema ARMAX

Observaciones:

- El valor inicial de $P(t-1)$ fue establecido a 10^6 según la ventana de configuración del bloque Memory como muestra la siguiente figura:

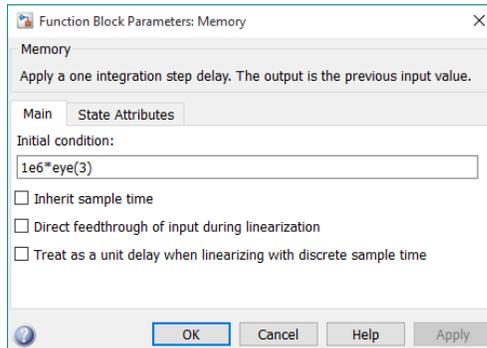


Figura 16. Valor inicial de $P(t-1)$.

- El valor inicial de $\theta(t-1)$ fue establecido a 0 según la ventana de configuración del bloque Memory1 como muestra la siguiente figura:

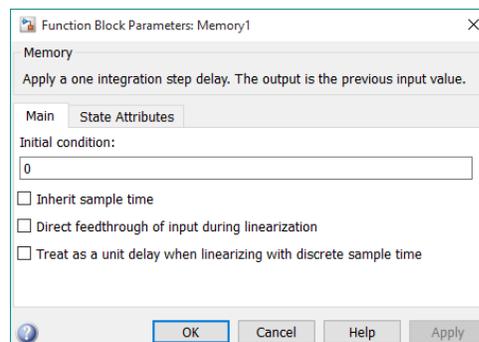


Figura 17. Valor inicial de $\theta(t-1)$.

- Para extraer los parámetros del vector $\theta(t)$ se convierte el vector a una tabla 2D mediante los bloques Direct Lookup Table (n-D) y mediante dos constantes en la entrada se especifica la locación del parámetro a mostrar. Considerar que para el modelo ARMAX son tres parámetros los que se deben estimar.
- En comparación con el modelo ARMAX los bloques de la simulación son en su mayoría idénticos, sin embargo se realizó modificaciones a Subsystem3 y se añadió el bloque Subsystem4. Solo se explicarán estos bloques ya que los demás se desarrollaron anteriormente.

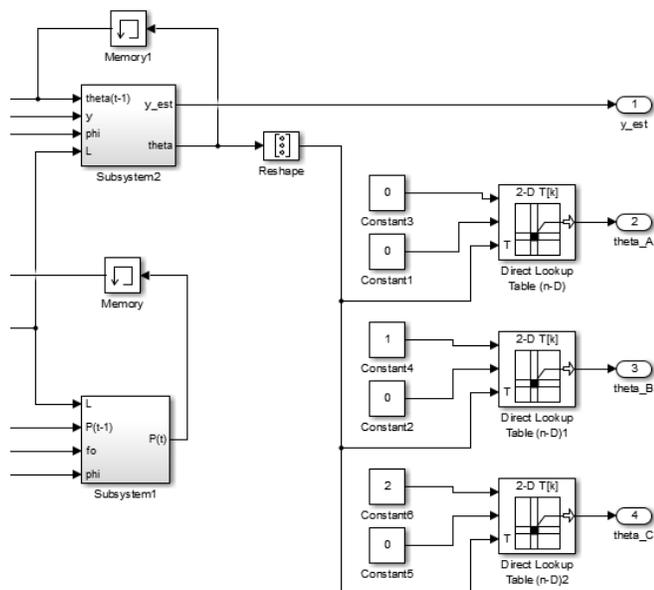


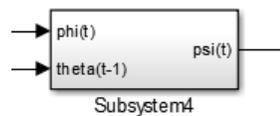
Figura 1813. Bloques Direct Lookup Table (n-D) contienen el vector $\theta(t)$.

Bloque Subsystem4

Para desarrollar el modelo ARMAX se introduce un subsistema (Subsystem4) encargado de aplicar el criterio para la Estimación de Máxima Probabilidad.

$$\Psi(t) = \frac{1}{C(q)} \varphi(t) \Rightarrow C(q)\Psi(t) = \varphi(t)$$

$$\Psi(t) = -c_1 \Psi(t - 1) + \varphi(t)$$



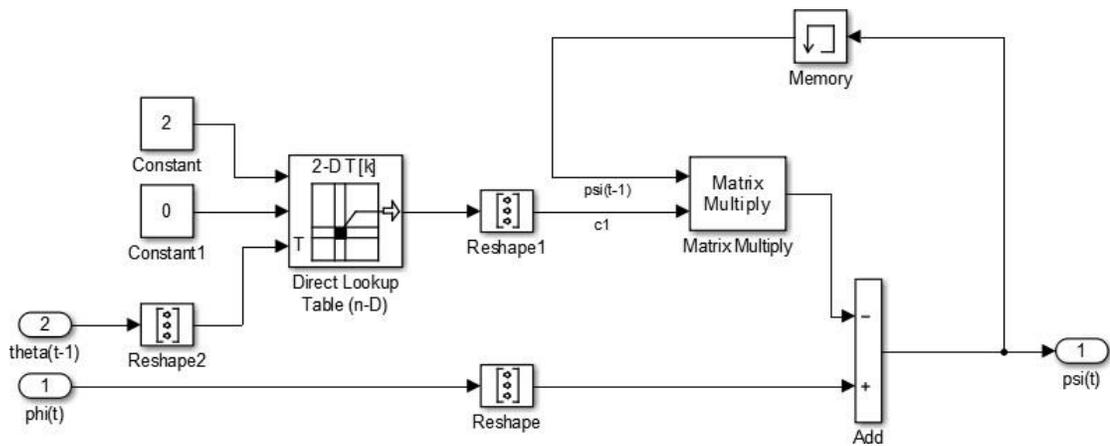


Figura 19. Bloque Subsystem4 - Calculo de $\psi(t)$.

Observaciones:

- La matriz $\theta(t-1)$ tiene dimensión 3×1 .
- El vector $\phi(t)$ y $\psi(t)$ también presentan dimensión 3×1 .
- El bloque Memory tiene valor inicial cero.
- El bloque Reshape permite modificar la dimensión de ϕ , se calcula la transpuesta mediante este bloque.

Bloque Subsystem3

Este bloque permite ordenar los valores anteriores de $u(t)$, $y(t)$ y $e(t)$ para formar el vector ϕ , para la simulación se trabajará solo con una planta de primer orden así que el vector ϕ tendrá la siguiente forma:

$$\varphi(t) = [-y(t-1) \ u(t-1) \ e(t-1)]^T$$

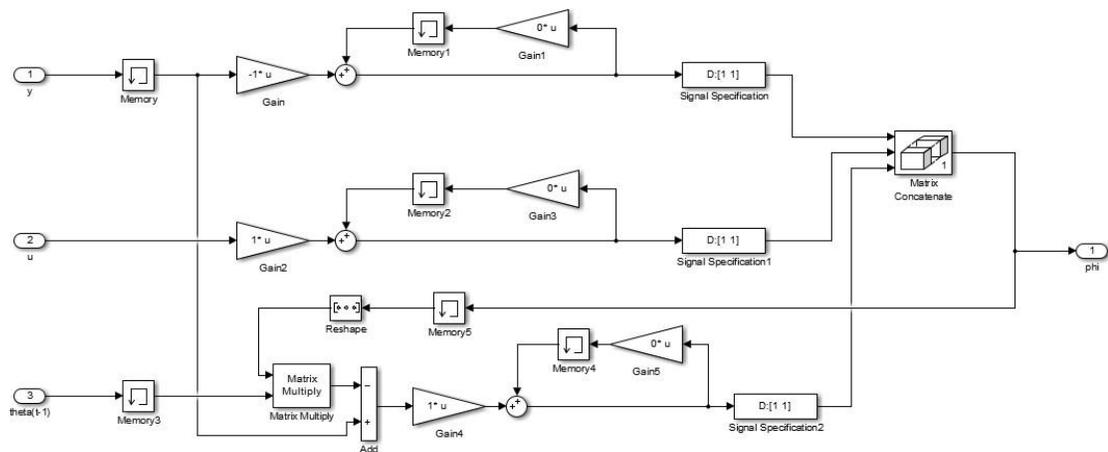


Figura 2014. Bloque Subsystem3 - Calculo de phi.

Observaciones:

- El bloque Gain posee ganancia negativa mientras que Gain2 y Gain4 tienen ganancias positivas. Esto debido a que el vector $\phi(t)$ está compuesto por $y(t-1)$ negativo mientras que los valores de $u(t-1)$ y $e(t-1)$ son positivos.
- El bloque Signal Specification permite a grandes rasgos establecer las características de los datos. Para la simulación por ser de primer orden solo se tomará una muestra anterior por tanto la dimensión de estos es 1×1 .

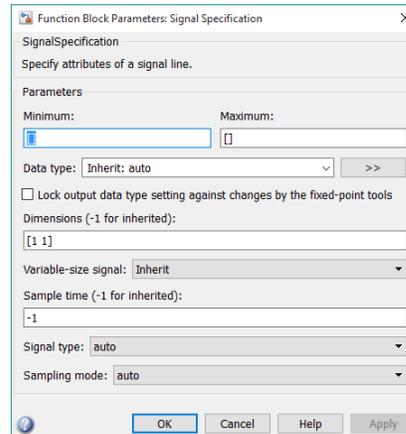


Figura 21. Bloque Signal Specification

- El bloque Matrix Concatenate permite unir vectores para formar una matriz, como las muestras tienen dimensión 1 x 1 al concatenarlas la dimensión será 3 x 1.

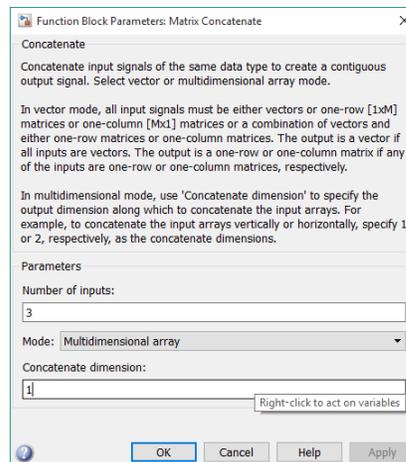


Figura 2215. Bloque Matrix Concatenate