UNIVERSIDAD DE LA SALLE
FACULTAD DE INGENIERIA

INFORME IDENTIFICACIÓN CON MATLAB

Juan Pablo Velandia Suarez, 45161013
Harold David León Hurtado, 45161031
Automatización de procesos - Ing. Tumialan Borja José Antonio
Universidad de la Salle

RESUMEN

En el siguiente informe se espera hacer uso de los comandos de la herramienta computacional Matlab, resolver modelos de sistemas, lo basado en lo anterior se obtendrá una respuesta visual, permitiendo comprender de mejor manera el comportamiento de los sistemas, por medio de la metodología de los tres puntos y los deltas en su respuesta, por lo que luego de hacer un tratamiento de datos para eliminar los datos innecesarios, dentro del vector de datos reales del módulo, luego por medio de una interpolación, se encuentra los valores del tiempo correspondientes para luego armar la función de transferencia equivalente a este proceso.

I. INTRODUCCIÓN

La identificación de sistemas con Matlab es una de las principales aplicaciones de las variables de estado, ya que de esta manera es posible comprender el funcionamiento de cualquier sistema, sea térmico, dinámico, mecánico, eléctrico, etc.
El presente informe muestra todos los resultados obtenidos en la primera práctica de laboratorio, enfocado a identificación de sistemas con Matlab, las simulaciones respectivas de cada una de las ecuaciones que modelan los sistemas, por medio de la herramienta Matlab. Comprender el comportamiento de los sistemas a través de las variables de estado facilita el estudio de las técnicas de control moderno.

Para cumplir con los objetivos de la práctica, se realizaron 2 actividades con el objetivo de aplicar diferentes técnicas de identificación y comprender la utilidad de estas a la hora de establecer el modelo matemático de un sistema.

La primera actividad consistió en seguir un artículo proporcionado por el docente de clase [1], el cual describe los diferentes métodos de identificación de modelos de orden reducido aplicados a una función de transferencia dada como ejemplo. Estas técnicas son fundamentales ya permiten construir una función de transferencia a partir de datos experimentales para describir el funcionamiento de un sistema y de esta manera diseñar un controlador que se desempeñe de forma correcta con la planta real. El ejemplo del articulo mencionado [1] fue realizado en Matlab para estudiar los resultados de forma óptima.

Con el desarrollo de la actividad anterior, se procedió a replicar las técnicas de identificación dadas por el articulo [1] sobre datos experimentales reales que fueron obtenidos de un módulo de procesos. Dichos datos fueron tomados y proporcionados por el docente de clase, los cuales corresponden a las medidas de la señal de entrada, tiempo, flujo y presión. El objetivo es realizar la identificación para el sistema de flujo y presión teniendo en cuanta la señal de entrada y el tiempo de medida que se usó.
Posteriormente, se compararon los resultados de cada técnica y se realizaron las respectivas conclusiones de acuerdo con cada gráfico.

II. OBJETIVOS

- Manejar la herramienta de Matlab para la estimación de una Data de un sistema cualquiera.

- Reconocimiento y Análisis del sistema.
- Construir el modelo de estimación a partir de las medidas experimentales.
- Comparar cada uno de los modelos mediante el error de predicción.
- Comprender la importancia de una buena estimación.

III. METODOLOGÍA

1. FLUJO

Para la identificación del sistema de flujo- entrada, se dispuso de los datos adquiridos del módulo de procesos de la Universidad de la Salle, luego de realizar un tratamiento de datos para eliminar los valores innecesarios para la identificación, se procedió a realizar la interpolación de los valores al 25, 50 y 75%, para así emplear cada uno de los cuatro métodos de identificación.

A continuación, se muestra fragmentos del código de implementación con su debida explicación de cada uno de los métodos.

```
load Datos

ti = Data(:,1);
fi = Data(:,2);
pi = Data(:,3);
ui = Data(:,4);
```

Para cargar el archivo con los datos suministrados para la práctica se utiliza el comando “load Datos” con ello se obtienen los vectores con la información de las variables involucradas en el sistema, las cuales serán expuestas a un tratamiento y a su posterior asociación a un modelo matemático que describa la dinámica de estos datos.

Luego se procede a la eliminación de los datos que no sirven, esto se hace observando la cantidad de datos del flujo donde se tiene un valor de 0, dicha cantidad de datos se borran al inicio del vector para el caso del flujo y para no crear inconsistencias en los datos se procede a eliminar la misma cantidad de datos del tiempo, pero esta vez al final del vector. Cuando se tienen los vectores listos para su uso, se procede a realizar la interpolación de datos, para la cual se extraen los valores correspondientes al 25, 50 y 75% de la salida en cuestión, para este caso el flujo. Luego por medio de un algoritmo desarrollado para fines de la práctica (figura ##), el programa toma estos valores, recorre los vectores, encuentra los valores asociados y procede a realizar la interpolación para determinar el dato que les corresponde a los valores en tiempo igualmente al 25, 50 y 75%.

```
% interpolar
dt25 = CalcularPunto(vector,dy25)
dt50 = CalcularPunto(vector,dy50)
dt75 = CalcularPunto(vector,dy75)
```

```
function x = CalcularPunto(vector,yc)
data =size(vector)
for i = 1:data(1)
    if  vector(i,2) >= yc
        x = interpolar(yc,vector(i,1),
        vector(i-1,1),vector(i,2),
        vector(i-1,2));
        break;
    end
end
```

Para la identificación se trabajaron los modelos de orden reducido de tres puntos 123c, los cuales utilizan los tiempos en los tres puntos de la gráfica o del vector de datos descritos anteriormente, para luego con estos valores y unos deltas de variación tanto para la entrada como para la salida (Δu y Δy), aplicar las fórmulas para la construcción de la función de transferencia que refleja las diferentes dinámicas e interacciones del sistema. (Víctor M. Alfaro, 2007).

Para síntesis del informe, solo se explicará el primer método, ya que los otros modelos trabajan de manera análoga a esta.

En el método de PDMTM, luego de calcular el delta de la entrada, se calculan los tiempos muertos, con los datos obtenidos de la interpolación y luego se aplica la fórmula para la construcción de la función de transferencia, que en este caso es Gp1.

```
% Modelo PDMTM para flujo
% ---------------------------------
Gp1 = kp*exp(-tm*s)/((tao*s+1)^2)
yi = lsim(Gp1,u,t);
plot(t,yi)
hold on
plot(t,y,'r')
```

La validación de cada uno de los métodos se hace por medio del cálculo de su error cuadrático, por lo que se realiza una comparación punto a punto entre el modelo identificado y los datos reales del módulo de procesos.

```
% Calculo del indice de aproximacion
% definir un tiempo para ambas señales
data_error = y-yi
error = sum(((y-yi).^2))
tam_dato = size(t);
error_promedio = error/tam_dato(1)
```

2. PRESIÓN

Para el análisis del sistema de presión se realizó el mismo procedimiento para cada uno de los pasos, tanto para el llamado de datos, interpolación e implementación de los cuatro métodos de identificación descritos anteriormente, por lo que al final es posible determinar de igual forma que tan bien describe el modelo identificado el sistema, por medio del cálculo de sus error cuadrático para cada uno de los modelos y así poder encontrar la estrategia que mejor se acomoda a este tipo de sistemas.

MÉTODO DE IDENTIFICACIÓN DE MODELOS DE ORDEN REDUCIDO DE TRES PUNTOS 123C:

Para construir los diferentes modelos propuestos por el articulo [1], se debe partir de las medidas o datos experimentales, pero en este caso se nos entrega una función de transferencia que corresponde al sistema que se desea identificar. Esta función es la siguiente:

$$Gpo(s) = \frac{1{,}25e^{-0{,}25s}}{(16s+1)(4s+1)(2s+1)(s+1)} \quad (1)$$

Teniendo la ecuación (1) se procede a simular su dinámica mediante el uso de Matlab, inyectándole como entrada una Step o escalón unitario y guardando en un vector las medidas de tiempo y amplitud obtenidas.

Las medidas de los vectores y, t sirven para construir los modelos estimación. Para eso se necesitan las medidas de amplitud en el 25%,50% y 75%. Las cuales se obtienen con la siguientes formulas:

$$\begin{aligned} y25 &= 0.25 * \mathrm{Ymax} \quad (2) \\ y50 &= 0.5 * \mathrm{Ymax} \quad (3) \\ y75 &= 0.75 * \mathrm{Ymax} \quad (4) \end{aligned}$$

Estas medidas de y, permiten describir el comportamiento del sistema, pero se hace necesario saber en qué tiempo se obtienen esas amplitudes, por lo que se debe revisar los vectores de medidas para saber si existe una amplitud igual a la calculada o si es necesario interpolar los datos para saber en qué tiempo se llega a las amplitudes y25, y50 o y75.

Para interpolar se recurre a la interpolación lineal cuya expresión corresponde a la ecuación (5) y expresa el cálculo de una amplitud Y, a partir de las medidas anteriores y posteriores.

$$y = y1 + \frac{y2-y1}{x2-x1}(x - x1) \quad (5)$$

Sin embargo, para este caso se necesitan interpolar las medidas de x o t, por lo que se debe despejar este término de la expresión (5) y calcular los tiempos para y25, y50 o y75. Por lo cual la expresión quedará de la siguiente manera:

$$x = x1 + \frac{x2-x1}{y2-y1}(y - y1) \quad (6)$$

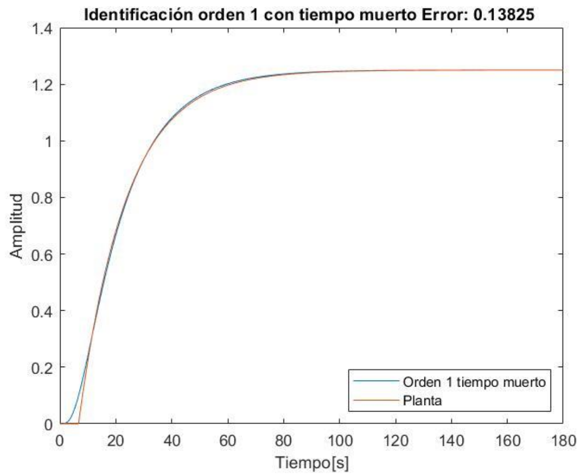
Con las medidas de tiempo para cada Y calculado, se procede a construir los diferentes modelos obteniendo los siguientes resultados:

- **Modelo de primer orden más tiempo muerto (POMTM):**

Construcción del Modelo POMTM en un script de Matlab:

```
% interpolar
dt25 = CalcularPunto(vector,dy25)
dt50 = CalcularPunto(vector,dy50)
dt75 = CalcularPunto(vector,dy75)
kp = dy / du;
tao =  0.9102*(dt75 - dt25)
tm = 1.2620*dt25 - 0.2620*dt75
% -------------------------------------------------
% -------------------------------------------------
Gp1 = kp*exp(-tm*s)/(tao*s+1)
yi = lsim(Gp1,u,t);
plot(t,yi)
hold on
plot(t,y,'r')
legend('Orden 1','Planta','Location','SouthEast')
% -------------------------------------------------
% Calculo del indice de aproximacion
% definir un tiempo para ambas señales
data_error = y-yi
error = sum(((y-yi).^2))
tam_dato = size(t);
error_promedio = error/tam_dato(1)
```

Comparación de respuesta entre el modelo real y el modelo POMTM



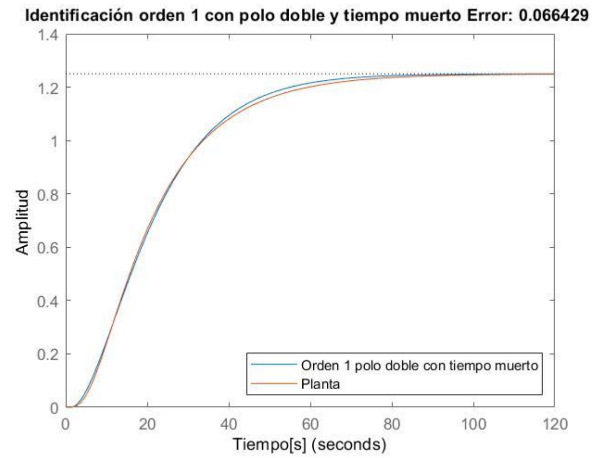
Como se observa en la figura anterior, el comportamiento del modelo POMTM es muy parecido al del sistema real, pero se logra observar gran diferencia en los valores entre 0 y 10 segundos. Esta diferencia es la que ocasiona un error de predicción de 0.13, por lo que aún puede ser mejorado.

- **Modelo de polo doble más tiempo muerto (PDMTM):**

Construcción del Modelo POMTM en un script de Matlab:

```
% interpolar
dt25 = CalcularPunto(vector,dy25)
dt50 = CalcularPunto(vector,dy50)
dt75 = CalcularPunto(vector,dy75)
kp = dy / du;
tao =  0.5776*(dt75-dt25)
tm = 1.5552*dt25-0.5552*dt75
% -----------------------------------------------------
% -----------------------------------------------------
Gp1 = kp*exp(-tm*s)/((tao*s+1)^2)
yi = lsim(Gp1,u,t);
plot(t,yi)
hold on
plot(t,y,'r')
legend('Polo Doble','Planta','Location','SouthEast')
% -----------------------------------------------------
% Calculo del indice de aproximacion
% definir un tiempo para ambas señales
data_error = y-yi
error = sum(((y-yi).^2))
tam_dato = size(t);
error_promedio = error/tam_dato(1)
```

Comparación entre el sistema real y el modelo PDMTM:



Como se observa en la figura anterior, el modelo PDMTM realiza una aproximación más exacta en comparación con el POMTM, ya que la diferencia es apenas visible en algunos tramos de tiempo. El error paso de 0.13 a 0.06 lo cual es una mejora significativa e indica que el modelo real debe ser de un orden superior para obtener un error más pequeño.

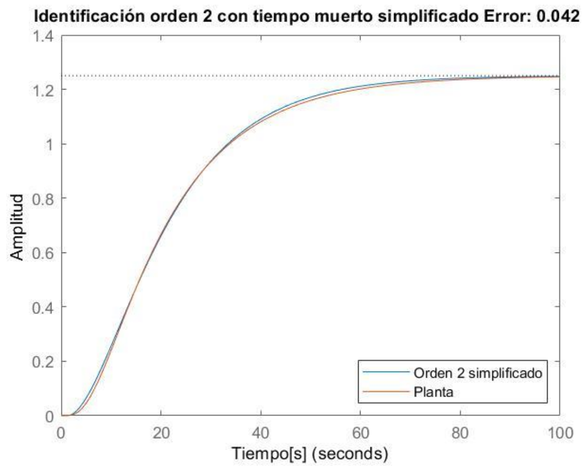
Modelo de segundo orden más tiempo muerto (SOMTM):

- **Método simplificado (SOMTMs):**

Construcción del modelo SMOTMs en un script de Matlab:

```
% interpolar
dt25 = CalcularPunto(vector,dy25)
dt50 = CalcularPunto(vector,dy50)
dt75 = CalcularPunto(vector,dy75)
kp = dy / du;
tao1 = 0.5776*(dt75-dt25);%t'
tm1 = 1.5552*dt25-0.5552*dt75;
tm2 = tm1;
a = (dt50-tm1-1.4362*tao1)/(1.9844*tao1-dt50+tm1);
tao2 = 2*tao1/(1+a);%t''
t1 = tao2;
t2 = tao2*a;
% -----------------------------------------------------
% -----------------------------------------------------
Gp1 = (kp*exp(-tm2*s))/((tao2*s + 1)*(a*tao2*s + 1)) ;
% yi = lsim(Gp1,u,t);
yi = step(27000*Gp1,t)
plot(t,yi)
hold on
plot(t,y,'r')
legend('Simplificado','Planta','Location','SouthEast')
% -----------------------------------------------------
% Calculo del indice de aproximacion
% definir un tiempo para ambas señales
data_error = y-yi
error = sum(((y-yi).^2))
num_data = size(t);
error_medio = error / num_data(1) ;
```

Comparación entre el sistema real y el modelo SOMTMs



Como se observa en la figura anterior, el modelo SOMTMs realiza una gran aproximación al modelo real, con error de predicción de 0.04, el cual es menor al error de 0.06 obtenido con el modelo PDMTM. Es decir, que el modelo real efectivamente se aproxima más a un modelo de segundo orden.

- **Método general (SOMTMg)**

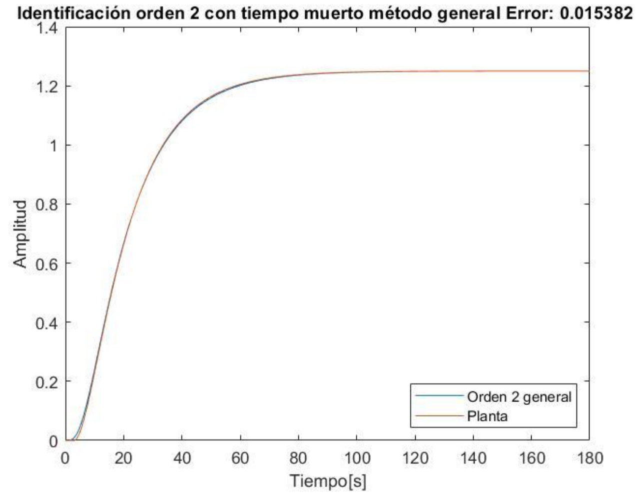
Construcción del modelo SOMTMg en un script de Matlab

```
% interpolar
dt25 = CalcularPunto(vector,dy25)
dt50 = CalcularPunto(vector,dy50)
dt75 = CalcularPunto(vector,dy75)
kp = dy / du;
a = (-0.6240*dt25 + 0.9866*dt50-0.3626*dt75)/
    (0.3533*dt25-0.7036*dt50 + 0.3503*dt75);
tao2 =    (dt75-dt25)/(0.9866 + 0.7036*a);%???
t1 = tao2;
t2 = tao2*a;
tm2 = dt75-(1.3421 + 1.3455*a)*tao2;


% ----------------------------------------------------------
% ----------------------------------------------------------
Gp1 = (kp*exp(-tm2*s))/((tao2*s + 1)*(a*tao2*s + 1)) ;

yi = lsim(Gp1,u,t);
plot(t,yi)
hold on
plot(t,y,'r')
legend('General','Planta','Location','SouthEast')
% ----------------------------------------------------------
% Calculo del indice de aproximacion
% definir un tiempo para ambas señales
data_error = y-yi
error = sum(((y-yi).^2))
tam_dato = size(t);
error_promedio = error/tam_dato(1)
```

Comparación entre el modelo SOMTMg y el modelo real.



Es claro en la figura anterior, que el modelo SOMTM es muy parecido al sistema real, de hecho, no muchas diferencias visibles por lo que el error es de apenas 0.015. Esto confirma que el modelo real se parece a un modelo de segundo orden como el propuesto por la técnica SOMTMg.

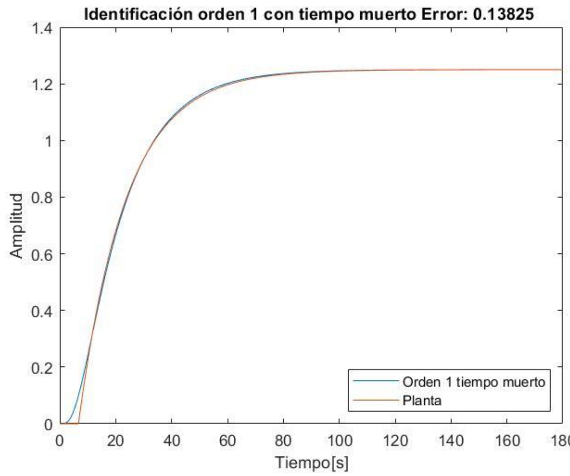
Con cada método se observó como cada técnica busca aproximar al modelo real para obtener una respuesta similar y que garantice su uso para aplicar técnicas de control eficientes.

Es cierto que cada modelo es una aproximación de la realidad, pero es por ello por lo que debe ser lo más exacto posible y con el menor más bajo, aunque en algunos casos se tienen en cuenta más criterios como los de optimización para saber qué modelo debe usarse dependiendo de la aplicación.

Pero en este caso se logra demostrar como un modelo puede ser construido a partir de medidas experimentales reales y que logre representar la dinámica de un sistema.

IV. ANÁLISIS DE RESULTADOS:

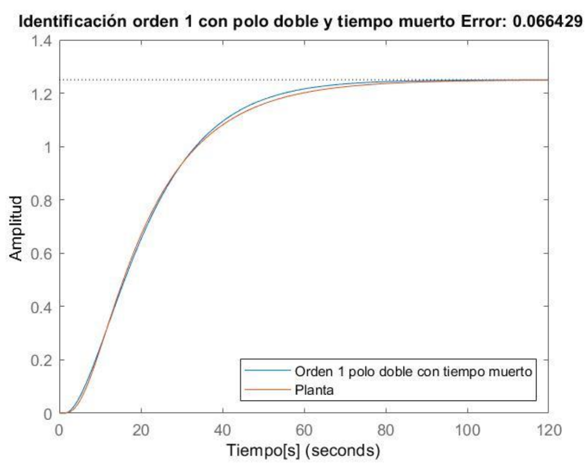
En este apartado se mostrará los resultados del ejemplo #3 de la función de transferencia propuesta en la ecuación (1), en el nombre del título se colocó la información del error cuadrático promedio de cada una de las identificaciones:



Gráfica 1 Identificación de orden uno con tiempo muerto de la planta propuesta

La Gráfica 1 hace alusión a la identificación de orden uno con tiempo muerto la gráfica con línea azul y la planta de color de rojo, como se puede evidenciar la identificación tiene un error de 0.13825, este error es producido en los primeros segundos en la toma de datos y puede ser mejorado. A continuación, se presenta la función de transferencia obtenida de la identificación,

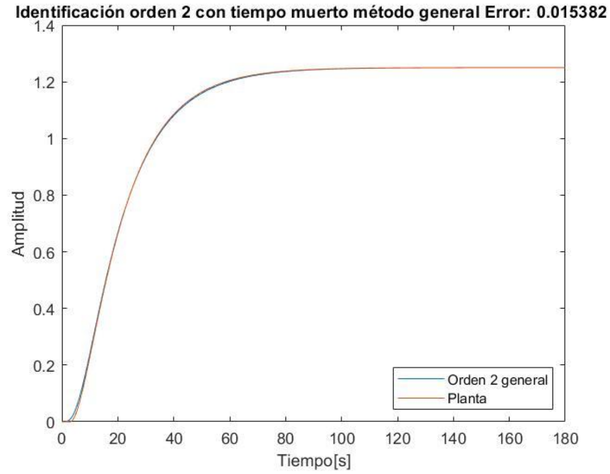
$$Gp1 = \frac{1.25 * e^{-6.69 * s}}{16.94 * s + 1}$$



Gráfica 2 Identificación de orden uno con polo doble y tiempo muerto

En la Gráfica 2 se evidencia la identificación de polo doble más tiempo muerto, con respecto a la Gráfica 1 se puede deducir que es más precisa debido a que el error es menor, aunque es diferente el grado de la función de transferencia obtenida, la función de transferencia obtenida con polo doble se muestra a continuación

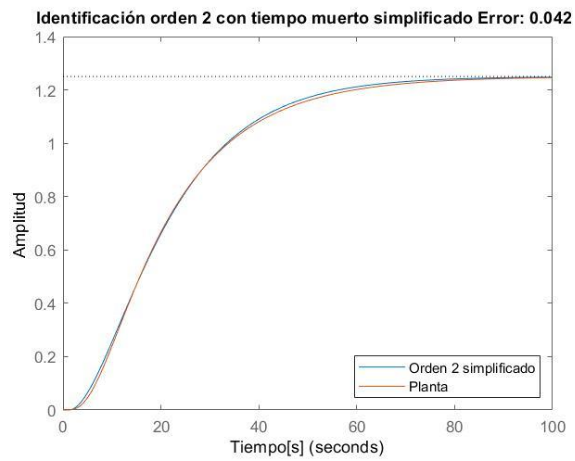
$$Gp_1 = \frac{1.25\ e^{-1.23\ s}}{115.5 * s^2 + 21.5 * s + 1}$$



Gráfica 3 Identificación de orden dos con tiempo muerto método general

En la Gráfica 3 se puede evidenciar una identificación de orden dos con tiempo muerto mediante el método general. A continuación, se presenta la función de transferencia resultante:

$$Gp_1 = \frac{1.25\ e^{-2.98\ s}}{74.45\ s^2 + 20.25\ s + 1}$$



Gráfica 4 Identificación de orden dos con tiempo muerto método simplificado.

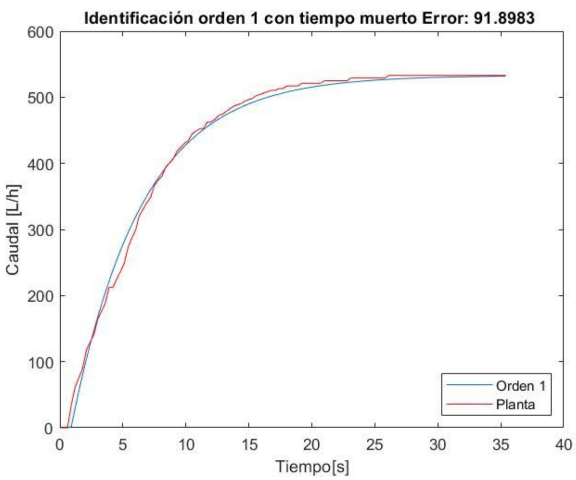
A continuación, se muestra la función de transferencia obtenida para realizar la Gráfica 4 .

$$Gp_1 = \frac{1.25\ e^{-1.23\ s}}{107.7\ s^2 + 21.5\ s + 1}$$

Mediante las funciones de transferencia mostrada en la Gráfica 3 y Gráfica 4 se puede evidenciar que los coeficientes de las linealizaciones son muy similares, sin embargo, el error presentado en la Gráfica 4 es menor, aunque ambas linealizaciones tienen muy buenos resultados.

De la Gráfica 1 hasta Gráfica 4 Identificación de orden dos con tiempo muerto método simplificado. Gráfica 4 se graficó la planta y el sistema identificado para evidenciar visualmente el seguimiento del sistema identificado con respecto al la planta, para este caso en particular se puede deducir que la linealización que presenta mayor aproximación es el de la Gráfica 4 Identificación de orden dos con tiempo muerto método simplificado., con error de 0.042 , se puede visualizar prácticamente están sobrepuestas una de la otra . Con respecto de la Gráfica 1hasta Gráfica 3 se calculó un error menor al 0.14, por ende, son aproximaciones muy exactas a la planta original.

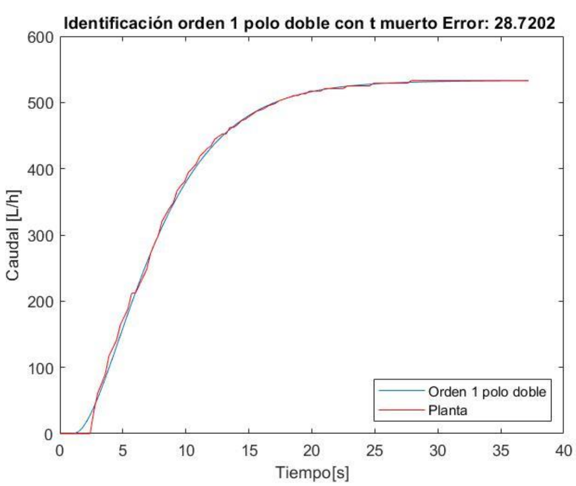
Los resultados mostrados de la Gráfica 5 a Gráfica 8 hacen referencia a las identificaciones realizadas a los datos de flujo [l/s] vs tiempo [s], del módulo de planta de procesos industriales.



Gráfica 5 Identificación de orden uno con tiempo muerto caudal vs Flujo

En la Gráfica 5 se puede evidenciar una identificación de orden uno con tiempo muerto. A continuación, se presenta la función de transferencia resultante:

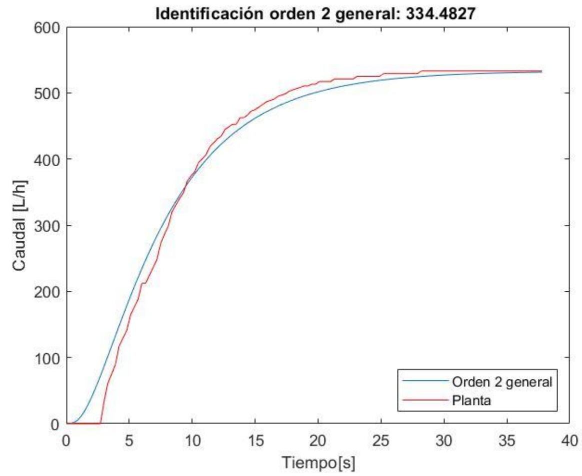
$$Gp_1 = \frac{0.01974\ e^{-0.885\ s}}{5.609\ s + 1}$$



Gráfica 6 Identificación de orden uno de polo doble con tiempo muerto caudal vs tiempo.

En la Gráfica 6 se puede evidenciar una identificación polo doble con tiempo muerto. A continuación, se presenta la función de transferencia resultante:

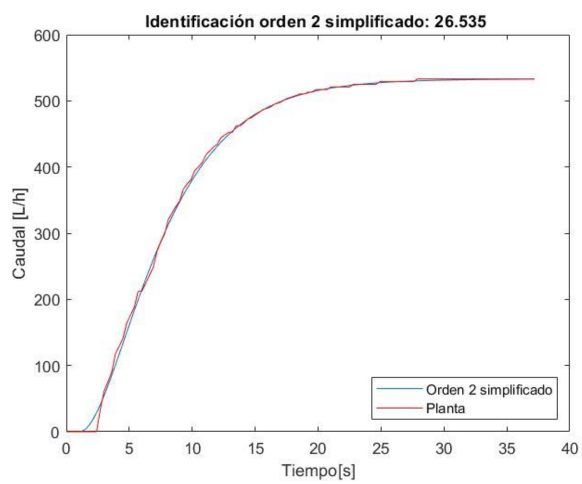
$$Gp_1 = \frac{0.01974\ e^{-0.878\ s}}{12.67\ s^2 + 7.119\ s + 1}$$



Gráfica 7 Identificación de orden dos con tiempo muerto con método general caudal vs tiempo.

En la Gráfica 7 se puede evidenciar una identificación de orden dos con tiempo muerto mediante el método general. A continuación, se presenta la función de transferencia resultante:

$$Gp_1 = \frac{0.01974\ e^{-0.00447\ s}}{11.53\ s^2 + 8.001\ s + 1}$$



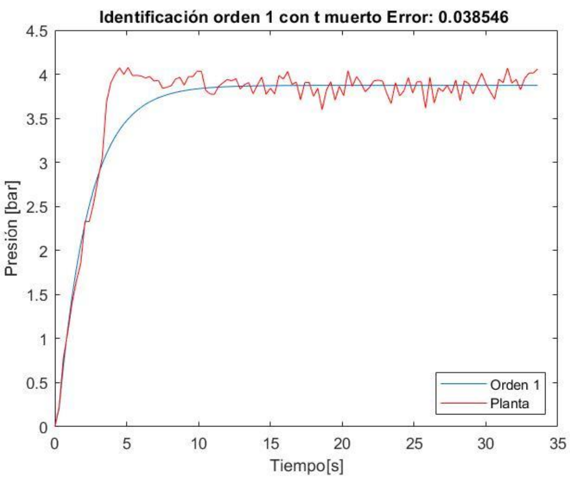
Gráfica 8 Identificación de orden dos con tiempo muerto con método simplificado caudal vs tiempo.

En la Gráfica 8 se puede evidenciar una identificación de orden dos con tiempo muerto mediante el método simplificado. A continuación, se presenta la función de transferencia resultante:

$$Gp_1 = \frac{0.01974\ e^{-0.878\ s}}{12.33\ s^2 + 7.119\ s + 1}$$

La identificación que mas se aproxima a la planta con respecto al flujo es la de la Gráfica 8 con un error promedio de 26.5. Visualmente podemos revisar que está sobrepuesta una de la otra. El error se debe al inicio antes de los 5 segundos donde el sistema identificado presenta variaciones con respecto a la planta.

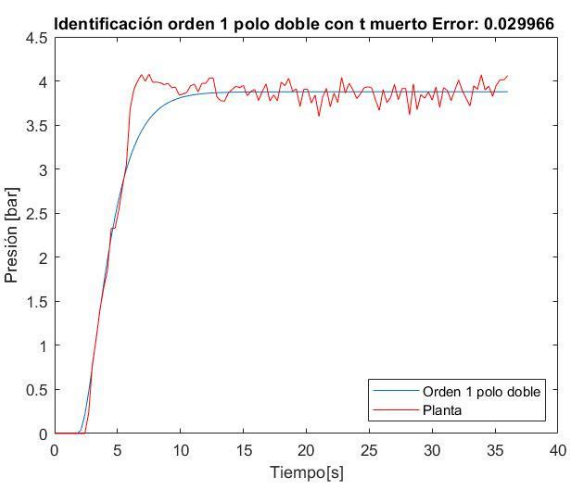
De la Gráfica 9 hasta Gráfica 12 corresponde las linealizaciones propuestas a la planta de procesos con respecto a la presión [bar].



Gráfica 9 Identificación de orden uno con tiempo muerto presión vs tiempo.

En la Gráfica 9 se puede evidenciar una identificación de orden uno con tiempo muerto mediante el método general. A continuación, se presenta la función de transferencia resultante:

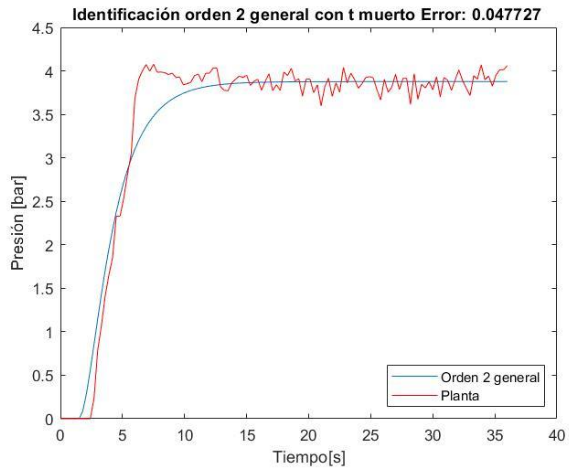
$$Gp_1 = \frac{0.0001435\ e^{-0.184\,s}}{2.119\ s + 1}$$



Gráfica 10 identificación de orden uno con polo doble y tiempo muerto presión vs tiempo

En la Gráfica 10 se puede evidenciar una identificación de polo doble con tiempo muerto. A continuación, se presenta la función de transferencia resultante:

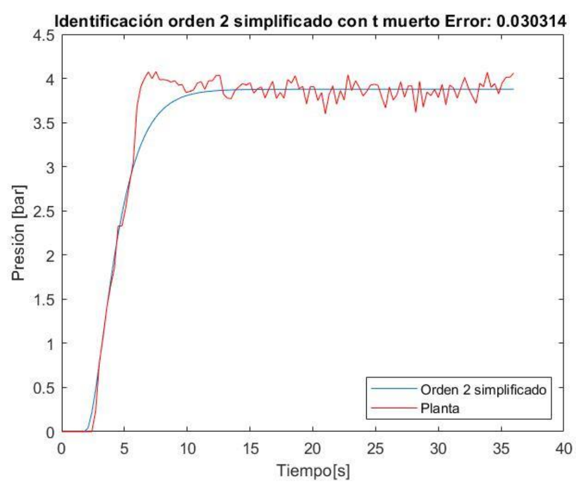
$$Gp_1 = \frac{0.0001436\ e^{-1.9\,s}}{1.812\ s^2 +\ 2.692\ s + 1}$$



Gráfica 11 identificación de orden dos con tiempo muerto con método general presión vs tiempo

En la Gráfica 11Gráfica 5 se puede evidenciar una identificación de orden dos con tiempo muerto mediante el método general. A continuación, se presenta la función de transferencia resultante:

$$Gp_1 = \frac{0.0001436\ e^{-1.5\,s}}{1.737\ s^2 +\ 2.996\ s + 1}$$



Gráfica 12 identificación de orden dos con tiempo muerto con método simplificado presión vs tiempo

En la Gráfica 12 se puede evidenciar una identificación de orden dos con tiempo muerto mediante el método simplificado. A continuación, se presenta la función de transferencia resultante:

$$Gp_1 = \frac{0.0001436\ e^{-1.9\,s}}{1.778\ s^2 +\ 2.692\ s + 1}$$

Con respecto a las identificaciones realizadas a la presión del modulo de procesos podemos decir que cualquiera de las identificaciones mostradas de la Gráfica 9 a la Gráfica 12 cumplen con su cometido con error menor al 0.0047

V. CONCLUSIONES

- La identificación que mas se aproxima a la planta de procesos con respecto a la presión es la identificación de la Gráfica 12.
- La identificación que más se aproxima a la planta de procesos con respecto a la presión es la identificación de la Gráfica 12.
- Para poder realizar una identificación exitosa es necesario de que el tiempo utilizado para la identificación sea el mismo que los datos de la planta.
- Para poder realizar la identificación de la presión de la planta de procesos fue necesario de promediar la altura máxima de la presión en la línea de tiempo mayor a los 5 segundos hasta los 35 segundos
- El tratamiento de datos es necesario para poder realizar la linealización, debido a que no afectan la dinámica del sistema, pero cambian los puntos calculados del tiempo al 25,50 y 75 % y la amplitud de la señal en esos puntos.

REFERENCIAS

[1] Víctor M. Alfaro, M.Sc. (2007) Método de identificación de modelos de orden reducido de tres puntos 123c. Escuela de ingeniería eléctrica, Universidad de Costa rica.