Informe de laboratorio.

Emanuel Bedoya Gallego
Aimer Daniel Hernández Hernández
Santiago Hernández Zapata
Envigado, Institución Universitaria de Envigado
Facultad de Ingeniería
EbedoyaG@correo.iue.edu.co
Adhernandez@correo.iue.edu.co
Shernandezz@correo.iue.edu.co

***Abstract*— This document presents the first laboratory report of the subject "Advanced Control" of the University Institution of Envigado, in the course of this the procedure to take response data in a plant with a certain process will be shown, and from these Various identification models such as non-recursive least squares, recursive least squares, etc. will be calculated from the data. In the references section, you will find the link to GitHub with the codes used during the practice.**

***Resumen*— En el presente documento se presenta el primer informe de laboratorio de la materia "Control avanzado" de la Institución universitaria de Envigado, en el transcurso de este se mostrará el procedimiento para tomar datos de respuesta en una planta con cierto proceso, y a partir de estos datos se calcularán varios modelos de identificación tales como mínimos cuadrados no recursivos, mínimos cuadrados recursivos, etc. En la sección de referencias, se encuentra el enlace a GitHub con los códigos utilizados durante la práctica.**

I. INTRODUCCIÓN

La adquisición y análisis de datos son herramientas fundamentales en la investigación y el desarrollo de proyectos en distintas áreas de la ciencia y la tecnología. En este informe de laboratorio se presenta una práctica sobre la toma de datos con Arduino y LabVIEW, y cómo se han utilizado estos datos para obtener modelos aproximados de una planta. Se mostrará combinación de la plataforma de hardware y softwar. Para ello, se han desarrollado seis modelos diferentes que permiten visualizar y analizar el comportamiento de la planta en diferentes condiciones de operación. Para esta práctica, utilizaremos una planta de control de flujo (Fig. 0)

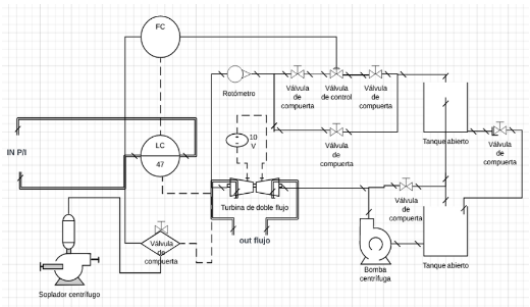

Fig. 0 Diagrama de flujo de la planta
(Tomado de [1])

II. OBJETIVOS

Los objetivos específicos para este informe son

1) *Descripción del proceso.*
2) *Calculo de los modelos.*
3) *Ilustración por medio de graficas de la planta y de los modelos a obtener.*
4) *Comparación de los modelos.*
5) *Validación en Simulink.*

III. MARCO TEÓRICO

A. *Para la adquisición de datos se hizo uso de LabVIEW (Fig. 1), un entorno de programación que permite la adquisición y el procesamiento de datos analógicos y digitales, y además facilita el monitoreo y el control de dichos procesos, además es compatible con una gran variedad de tarjetas de adquisición de datos lo que permite que su usabilidad sea una gran ventaja al aplicarlo.*

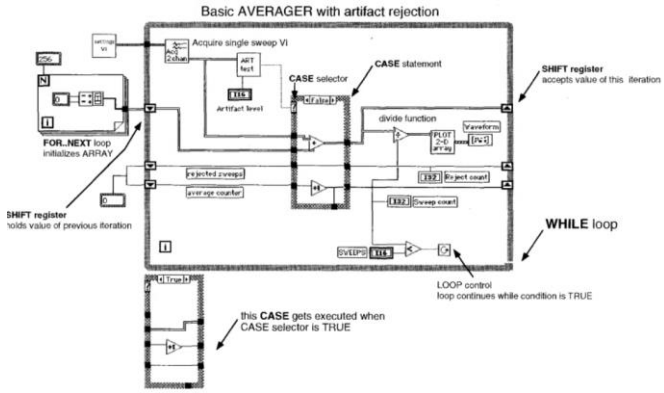

Fig. 1 Interfaz de LabVIEW
(Tomado de [1])

B. *Arduino, "El Arduino Uno es una placa electrónica basada en el microprocesador Atmega32, (Fig. 2). Cuenta con 14 pines digitales de entrada/salida (de los cuales 6 pueden ser utilizados como salidas PWM), 6 entradas analógicas, un resonador cerámico 16 MHz, una conexión USB, un conector de alimentación, un header ICSP, y un botón de reinicio. Contiene todo lo necesario para apoyar al microcontrolador, basta con conectarlo a un ordenador con un cable USB" [2]. Para la toma de datos se utilizó una tarjeta Arduino, que cuenta con muchas ventajas en comparación con otros equipos de adquisición, tanto en el precio como en la usabilidad y compatibilidad, ya que funciona en una gran variedad de entornos, incluso en tiempo real en este caso con LabVIEW y esto permite que se monitoree el sistema de una manera más precisa y confiable*

Fig. 2 Placa Arduino UNO
(Tomado de [2])

C. *System Identification* El uso de Matlab para estimar modelos es muy eficiente, gracias al toolbox "System Identification" el cual identifica los parámetros del sistema para luego construirlos y representarlos gráficamente, en este caso se usó para la estimación de los modelos POR, SOR y ARX, que realiza dichos cálculos de una manera eficiente y mucho más rápida en comparación a otros métodos.

D. *Sistemas en tiempo discreto* son aquellos en los que las señales de entrada y salida se definen únicamente en puntos discretos en el tiempo. Es decir, la señal se muestrea a intervalos regulares de tiempo y se procesa en cada uno de esos puntos discretos (Fig. 3). Estos sistemas son ampliamente utilizados en el procesamiento digital de señales, ya que permiten la implementación de algoritmos numéricos y operaciones matemáticas en sistemas electrónicos o computacionales.

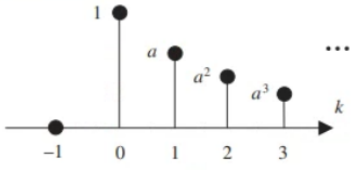

Fig. 3 Ejemplo función exponencial en tiempo discreto
(Tomado de [4])

IV. PROCEDIMIENTO

Lo primero que se debe hacer, es una toma de datos de la planta a analizar, con estos datos podremos usar softwares especiales para calcular modelos aproximados a la planta, posterior a ello, realizar sistemas de control para esta. Actualmente hay muchos métodos para tomar datos de una planta, para el caso de esta práctica, utilizaremos un Arduino Uno, el cual estará conectado a la planta (Fig. 5), acompañado a esta tarjeta, utilizaremos el software para computadores "LabVIEW". Con un pequeño programa (Fig. 4) en el cual evidenciamos 3 etapas, la primera funciona envió de datos a la planta por medio del PWM del Arduino, en esta práctica se comenzó enviando un escalón de 30, posteriormente se incrementó este valor de 4 en 4 hasta llegar a 66. La segunda etapa que se observa la parte en la cual el Arduino toma los datos de la planta por medio del Pin de lectura análoga, estos datos (Entrada y salida) se unen en una gráfica y en un archivo lvm llamado "Test.lvm" en la tercera etapa.

El archivo entregado por LabVIEW expresa las cifras decimales con "Comas" (,). En un computador se reemplazaron estas comas por "puntos" (.), el motivo de esto es por el lenguaje que se va a utilizar posteriormente (Matlab interpreta las cifras decimales con puntos).

Teniendo la base de datos en archivo txt, podemos ver la base de datos (Fig. 6), se evidencian 4 columnas, siendo la columna 1 y 3 los datos equivalentes al tiempo, la columna 2 equivale a la respuesta de la planta y la columna 4 nos muestra el escalón enviado por el Arduino.

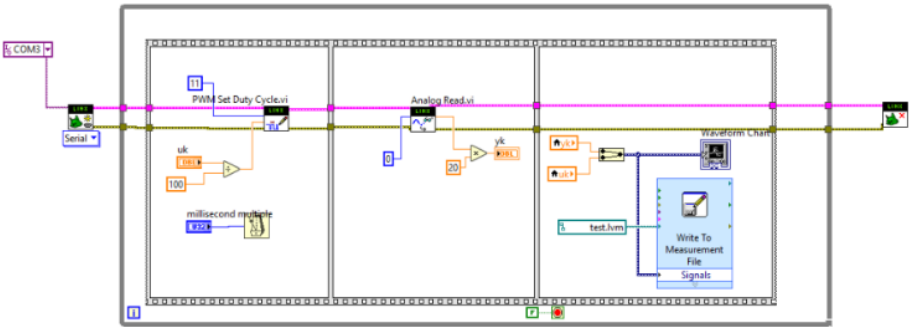

Fig. 4 Programa utilizado para la toma de datos
(Imagen proporcionada por el docente Luis Eduardo)

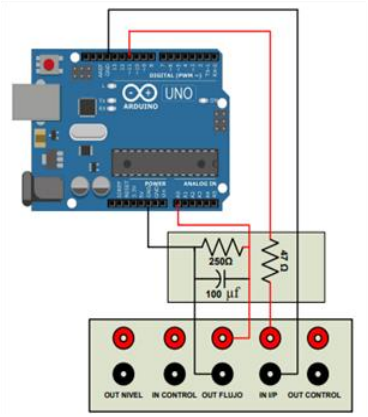

Fig. 5 Conexión del Arduino a la planta para toma de datos (Imagen proporcionada por el docente Luis Eduardo)

0.000000	23.242187	0.000000	30.000000
0.028919	23.730469	0.028919	30.000000
0.062887	23.242187	0.062887	30.000000
0.084819	23.339844	0.084819	30.000000
0.109802	23.339844	0.109802	30.000000
0.131912	23.339844	0.131912	30.000000
0.155617	23.242187	0.155617	30.000000
0.179765	23.339844	0.179765	30.000000
0.205452	23.339844	0.205452	30.000000
0.229949	23.535156	0.229949	30.000000
0.262294	23.535156	0.262294	30.000000
0.289221	23.242187	0.289221	30.000000
0.310374	23.242187	0.310374	30.000000
0.335135	23.339844	0.335135	30.000000
0.355204	23.242187	0.355204	30.000000
0.377120	23.242187	0.377120	30.000000
0.398136	23.339844	0.398136	30.000000
0.416978	23.144531	0.416978	30.000000
0.447236	23.242187	0.447236	30.000000
0.471760	23.242187	0.471760	30.000000
0.490719	23.242187	0.490719	30.000000
0.512623	23.242187	0.512623	30.000000
0.532577	23.242187	0.532577	30.000000
0.552518	23.242187	0.552518	30.000000
0.572465	23.242187	0.572465	30.000000
0.597433	23.242187	0.597433	30.000000
0.617497	23.242187	0.617497	30.000000
0.638290	23.242187	0.638290	30.000000
0.658367	23.046875	0.658367	30.000000
0.684447	23.242187	0.684447	30.000000
0.708248	23.242187	0.708248	30.000000
0.728197	23.144531	0.728197	30.000000
0.750216	23.242187	0.750216	30.000000
0.769083	23.144531	0.769083	30.000000
0.785920	23.242187	0.785920	30.000000
0.802269	24.121094	0.802269	30.000000

Fig. 6 Base de datos utilizada (elaboración propia)

Ya teniendo la base de datos organizada, introducimos está a Matlab, en este caso con la función “uigetfile” seleccionamos el archivo y luego lo cargamos al código con “load”, teniendo la base de datos en el entorno, la podemos separa en entrada o escalón (u), respuesta (y) y tiempo (t); con los datos de esta forma podemos graficar los datos obtenidos en LabVIEW (Fig. 7) viendo los datos de la planta (Línea negra) y los escalones indicados (Línea roja). Teniendo esto, escogeremos un escalón para generar el modelo en esta parte, en este caso, se seleccionó el escalón de 46, el motivo fue la buena respuesta de la planta en este punto, el tiempo de muestreo es largo y los datos no muestran un ruido muy invasivo. Esta sección del escalón la separamos y la iniciamos desde el punto (0,0) (Fig. 8) para operarlo de una forma mas cómoda y selecta

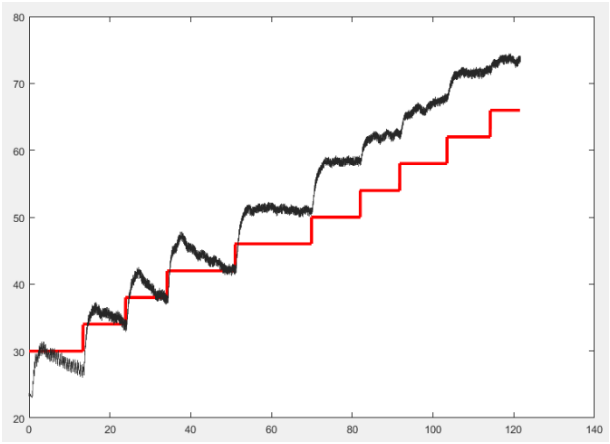

Fig. 7 Grafica de la base de datos obtenida (elaboración propia)

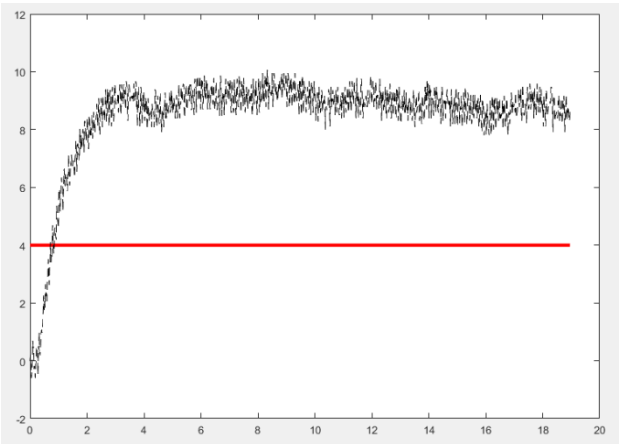

Fig. 8 Escalón seleccionado para el análisis (elaboración propia)

Con estos datos desplazados, procedemos a calcular el modelo de aproximación, primero por el método de mínimos cuadrados no recursivos, realizando las operaciones respectivas, obtenemos una la función de transferencia en tiempo discreto con la variable en “Z” (Fig. 9). Teniendo esta función podemos ingresarla a la función “step” de Matlab, ésta ingresará un escalón unitario (este escalón se debe multiplicar por la amplitud del escalón ingresado en la planta, para este caso, se multiplica por cuatro) a esta función para ver su funcionamiento y similitud con los datos reales (Fig. 10), observamos concordancia entre el modelo calculado con los datos reales de la planta.

$$\frac{0.04883\ z + 0.04359}{z^2 - 0.7167\ z - 0.2406}$$

Fig. 9 Función de transferencia obtenida por el método de mínimos cuadrados no recursivos
(elaboración propia)

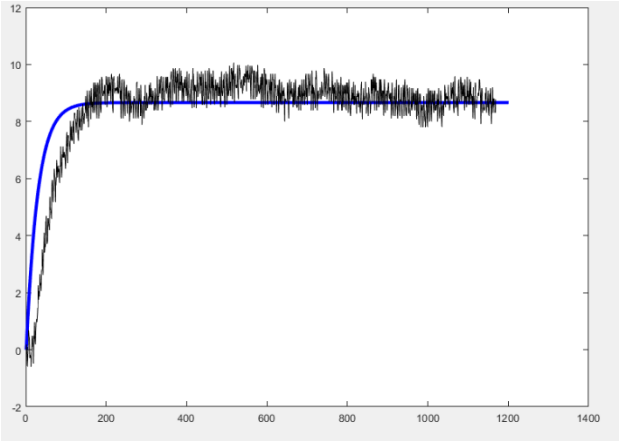

Fig. 10 Grafica obtenida con la función de mínimos cuadrados no recursivos
(elaboración propia)

Este proceso mencionado, se realiza en 2 casos más, los cuales corresponden a mínimos cuadrados recursivos por método 1 y método 2.

La única diferencia que tienen estos dos métodos es que requieren un dato de planta nuevo, ya que se esta usando los datos de la planta desplazada, este dato nuevo equivale a "8.7891".

Como en el caso anterior, obtenemos 2 funciones de transferencia en tiempo discreto, una para el método 1 (Fig. 11) y otra para el método 2 (Fig. 12)

$$\frac{0.04883\ z + 0.04361}{z^2 - 0.7169\ z - 0.2404}$$

Fig. 11 Función de transferencia obtenida por el método de mínimos cuadrados recursivos (Método 1)
(elaboración propia)

$$\frac{0.04621\ z + 0.04621}{z^2 - 0.7167\ z - 0.2406}$$

Fig. 12 Función de transferencia obtenida por el método de mínimos cuadrados recursivos (Método 2)
(elaboración propia)

Con las dos funciones obtenidas, podemos realizar el mismo paso anteriormente explicado con la función "Step" para obtener la grafica en la cual observamos el método 1 con los datos reales (Fig. 13) y el método 2 con los datos reales (Fig. 14).

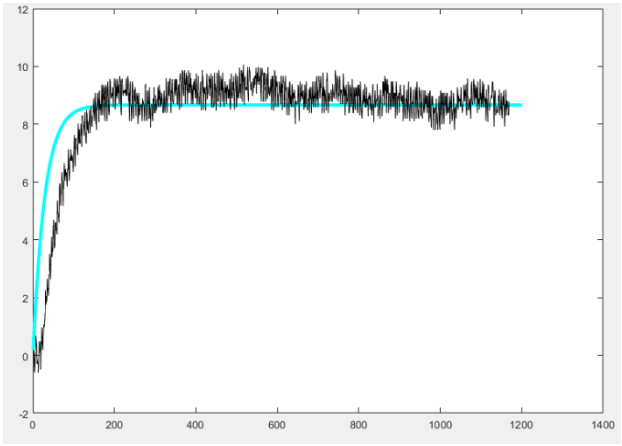

Fig. 13 Grafica obtenida con la función de mínimos cuadrados recursivos (Método 1)
(elaboración propia)

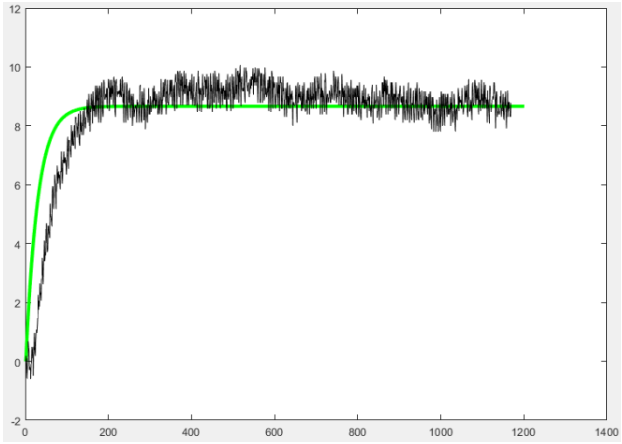

Fig. 14 Grafica obtenida con la función de mínimos cuadrados recursivos (Método 2)
(elaboración propia)

Para estimar esta planta por otros 3 métodos, se utilizarán las herramientas que nos brinda el toolbox de System

Identification, este toolbox se puede iniciar desde la ventana de comandos, aquí se abre directamente el toolbox (Fig. 15), en la ventana a la vista, ingresamos a la lista desplegable que nos permite importar los datos necesarios, en este caso, se escoge para ingresar datos en el dominio del tiempo, esta opción nos muestra una ventana nueva (Fig. 16), en esta ventana ingresamos los datos, nuevamente, los datos del escalón seleccionado y desplazado al origen, siendo estos el escalón "ut", la planta "yt", y, en el caso del tiempo, se debe ingresar el tiempo de muestreo, este se puede ver en la segunda posición del vector de tiempo, ingresándolo como "tt(2)"

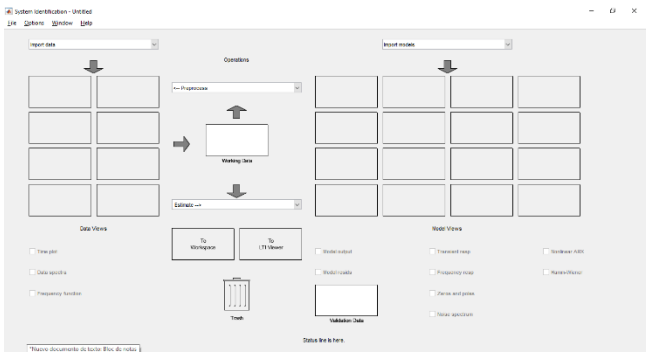

Fig. 15 Ventana principal del Toolbox "SystemIdentification" (elaboración propia)

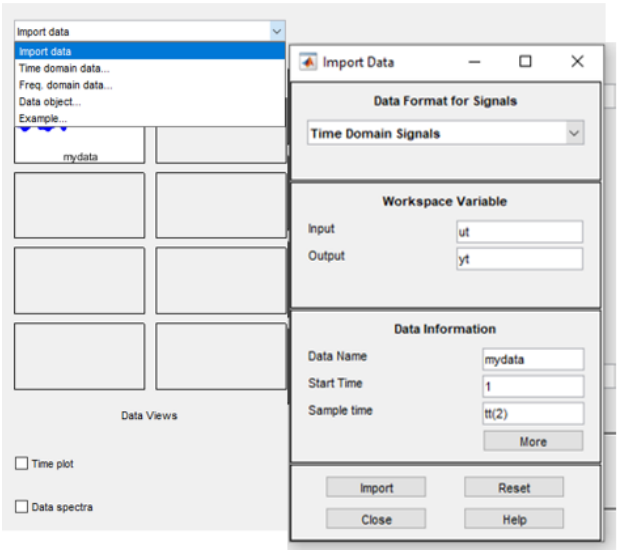

Fig. 16 Configuración de los datos a importar en el toolbox (elaboración propia)

Luego de esto, el toolbox tiene lo necesario para estimar los modelos, empezando por el ARX, esto se logra desde la lista desplegable de estimar, en donde escogeremos modelos polinomiales, esta opción nos abre una ventana nueva (Fig. 17), aquí basta con pulsar en estimar y el toolbox automáticamente nos va a estimar el modelo ARX y lo añadirá en las casillas de la parte derecha del toolbox.

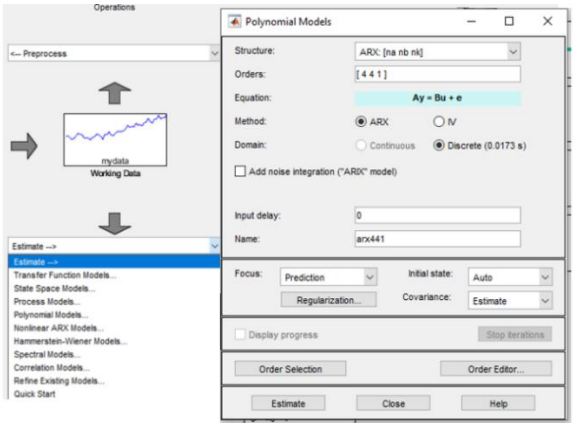

Fig. 17 Opción para estimar el modelo ARX (elaboración propia)

Para estimar el modelo POR y el modelo SOR, el proceso es similar; en la lista desplegable de estimar, ingresamos en la opción de modelo de proceso, una vez mas se nos abre una ventana nueva (Fig. 18), aquí ingresamos el numero de polos que deseamos para la estimación (1 polo para el caso POR y 2 polos para el caso SOR)

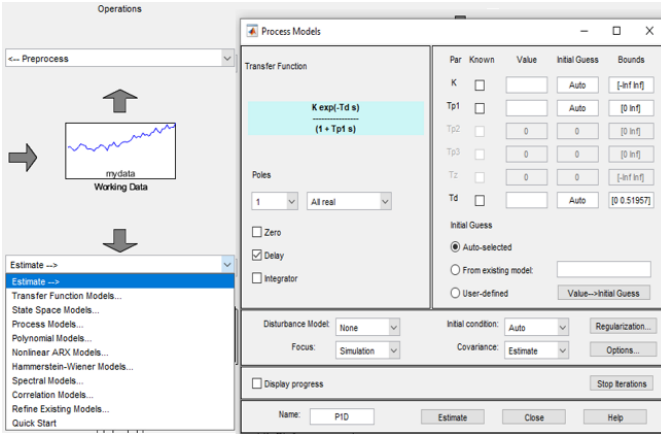

Fig. 18 opción para estimar el modelo POR (elaboración propia)

Luego de tener los 3 modelos en las casillas de la derecha, podemos ver la función de transferencia estimada por cada uno de los modelos, ARX (Fig. 19), POR (Fig. 20) y SOR (Fig. 21)

```
Discrete-time ARX model: A(z)y(t) = B(z)u(t) + e(t)
  A(z) = 1 - 0.7167 z^-1 - 0.2406 z^-2

  B(z) = 0.09242 z^-1
```

Fig. 19 Modelo ARX estimado
(elaboración propia)

```
Process model with transfer function:
                Kp
  G(s) = ---------- * exp(-Td*s)
           1+Tp1*s

          Kp = 2.2449
         Tp1 = 0.84756
          Td = 0.35428
```

Fig. 20 Modelo POR estimado
(elaboración propia)

```
Process model with transfer function:
                  Kp
  G(s) = ----------------- * exp(-Td*s)
         (1+Tp1*s)(1+Tp2*s)

          Kp = 2.2439
         Tp1 = 0.80567
         Tp2 = 0.17803
          Td = 0.19318
```

Fig. 21 Modelo SOR estimado
(elaboración propia)

Estos tres modelos obtenidos se pueden graficar desde el mismo toolbox, activando la opción "Model output" e ingresando a la ventana de "Validation data", aquí el programa nos mostrará los datos reales (yt) y los 3 modelos estimados por el mismo (Fig. 22).

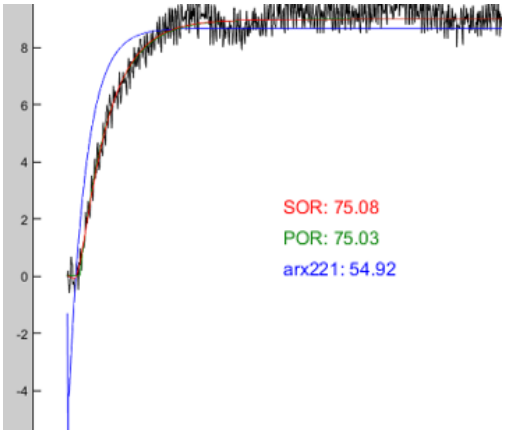

Fig. 22 Grafica de los modelos ARX, POR y SOR en comparación con los datos ingresados
(elaboración propia)

Por ultimo, ingresaremos los modelos a Simulink, aquí, vamos a validar los modelos de acuerdo con el criterio de error IAET (Fig. 24). los modelos mínimos cuadrados no recursivos, mínimos cuadrados recursivos M1, mínimos cuadrados recursivos M2 y el modelo ARX tienen un desempeño similar, con valores de IAET muy cercanos entre sí (Fig. 23). El modelo POR, por otro lado, tiene un valor de IAET notablemente más alto que los otros modelos, con un valor de 232.1. Finalmente, el modelo SOR tiene un valor de IAET de 226.3, que se encuentra en el rango de los modelos mínimos cuadrados no recursivos al modelo ARX, pero aún es ligeramente mayor que los valores más bajos.

De aquí se observa que el modelo que presenta un error menor es el modelo de mínimos cuadrados recursivos M2 y el modelo ARX, pues tienen un error de 225.2.

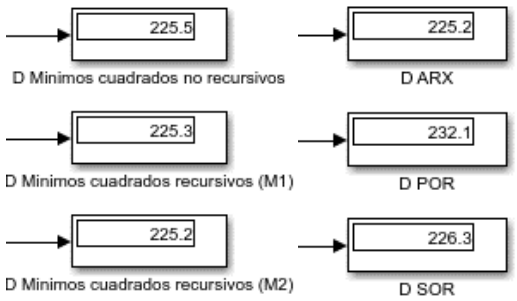

Fig. 23 Resultados del error por el criterio IAET
(elaboración propia)

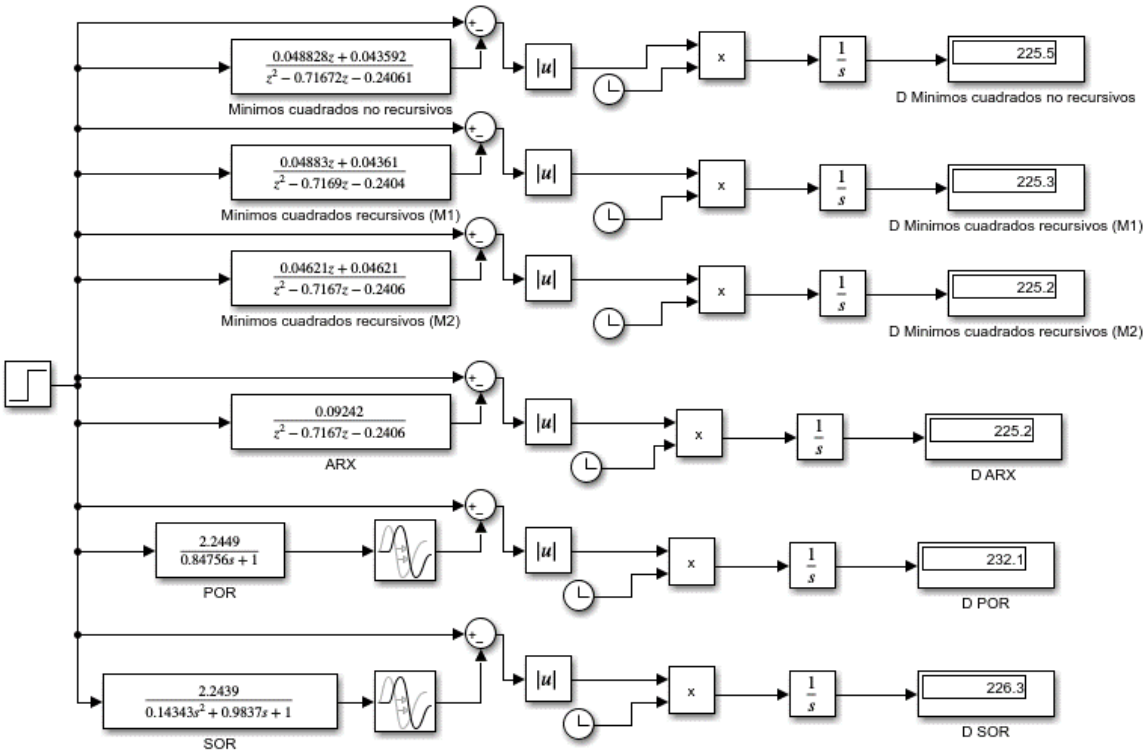

Fig. 23 Programa en Simulink para calcular el error por el criterio IAET (elaboración propia)

V. CONCLUSIONES

1) Se puede concluir que los modelos de mínimos cuadrados no recursivos, recursivos por método 1 y 2, y el modelo ARX son los que tienen mejor desempeño según el criterio de IAET, mientras que el modelo POR tiene un desempeño inferior y el modelo SOR tiene un desempeño aceptable pero aun ligeramente inferior a los mejores modelos.
2) Gracias a la comparación de los 6 modelos, podeos concluir que los datos obtenidos son confiables y se pueden utilizar para un futuro controlador, pues obtuvimos estimaciones de planta muy similares, sin mucha dispersión en el error calculado.
3) Un punto muy importante en esta practica es tomar datos con los mayores escalones posibles, pues se observó que algunos se dañan, ya sea por fallas en el Arduino, ruido en los datos o mala respuesta de la planta, esto puede dejar inútil la base de datos que usemos al trabajar. En este caso se realizó con 9 escalones, separados por una magnitud de 4 entre sí, de estos 9 escalones, solo 2 escalones se acordaron como aptos, pues no contaban con ruido muy determinante, contaban con buena respuesta de la planta y con buen tiempo de muestreo.

VI. REFERENCIAS

[1] https://doi.org/10.1007/BF01627421
[2] http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S2518-44312015000100006&lng=es&tlng=es
[3] https://doi.org/10.1119/1.4978720
[4] https://doi.org/10.1016/B978-0-12-814433-6.00002-8
[5] https://github.com/Emanuel215B/ControlAvanzado.git