

PRÁCTICA N°1: CONTROL II

EMULACIÓN DE UN SISTEMA DE TERCER ORDEN.

(octubre de 2023)

Dago Mauricio Quiroz Hoyos
dagom.quirozh@uqvirtual.edu.co
Universidad Del Quindío
Armenia, Quindío

Resumen-

En esta práctica se llevó a cabo el diseño y análisis de un sistema de tercer orden, el cual fue simulado en matlab e implementado por medio variables de estado en labVIEW, además emulando el comportamiento del mismo, en labVIEW, con ayuda del ide de arduino (hardware) y dos ESP32(software), con el fin de obtener y analizar el comportamiento en la lazo abierto de un sistema de tercer orden.

I. INTRODUCCIÓN

En el ámbito de la ingeniería de sistemas de control automático, la emulación y simulación de sistemas dinámicos desempeñan un papel fundamental en el diseño y análisis de sistemas complejos.

Este laboratorio se centra en la emulación de un sistema dinámico de tercer orden utilizando herramientas digitales, mediante las cuales se espera observar la respuesta de dicho sistema enfrentado a diferentes valores de entrada. Al culminar esta práctica, se habrá empleado MatLab y LabVIEW para simular, y acondicionados para emular el sistema representando dicha respuesta gráficamente.

Esto se logra discretizando el sistema propuesto y aplicando ecuaciones de diferencia para obtener una ecuación que rige el sistema. Una vez hallada, debe ser simulada mediante variables de estado en LABVIEW y combinada con el hardware ESP 32 es emulada y sometida a diferentes escalones, que arrojaron la respuesta del sistema mediante un archivo de texto, con el cual es posible graficar en MatLab para su posterior análisis.

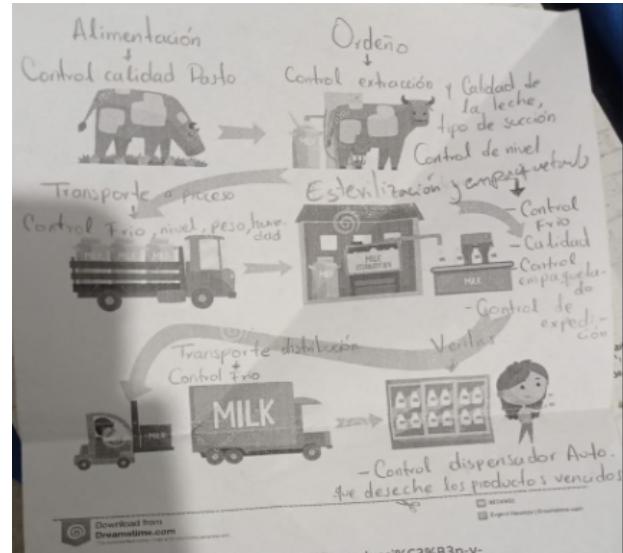


Figura 1. Proceso de industrialización de la leche.

La alimentación del bovino es demasiado importante, es el indicador que permite tener una guía de cuánto se puede esperar de producción en la leche, además las fuentes de agua también van a influir. Para llevar un control del medio se plantea generar una aplicación en la cual van a estar todos los datos que podamos adquirir del ordeño, además en los potreros un sensor para medir la altura del pasto y otros para medir la acidez del agua que consume el bovino.

Para la etapa del ordeño automático, se planteó un control de calidad del producto que se está adquiriendo, poder sensar todo tipo de microorganismos dañinos, además que esta herramienta sea auto ajustable, debido a que todos los animales no tienen la misma dureza en las mamarias, esto puede evitar dañar o estropear los órganos del animal.

La recolección de la misma (transporte) desde el sitio de producción a la planta de tratamiento, debe cumplir con requisitos de sanidad, para llevar un control exhaustivo se planteó monitorear dos parámetros muy importantes: la temperatura y el diagnóstico de las partículas de leche, con la finalidad de llevar un producto en óptimas condiciones a su tratamiento de industrialización.

En la esterilización de la leche se planteó implementar sensores que sin ser invasivos me permitan tener datos de temperatura y tiempo de la leche cuando se somete a altas temperaturas.

En el proceso de empaquetado se propuso obtener datos como la temperatura a la que va a hacer empaquetado el producto, la esterilización previa de cada uno de los envases, posteriormente, poder tener un sensado en donde se obtenga el valor(limpio), por último para o tener problema con el llenado y sellado del envase se implementa un sensor que me permite tener un dato del sellado y por otro lado implementar un sensor no invasivo de nivel.

El transporte desde la planta a su sitio de venta, se plantea llevar el producto en un ambiente controlado, para esto se implementa un sensor de humedad y de temperatura.

Para la venta se planteó el poder implementar un dispensador automático, que aparte de entregar los productos de manera automática, pueda desechar los productos próximos a vencer o que ya estén vencidos.

La estructura del documento se presenta a continuación:

1. Introducción.
2. Métodos e instrumentos
3. Análisis y Resultados.
4. Conclusiones.
5. Bibliografía
6. Anexos

II. MÉTODOS E INSTRUMENTOS

En la figura (2) se proporciona la metodología con la cual se abordó esta práctica de laboratorio, por medio de esta se van a presentar los resultados obtenidos después de la práctica.

En la tabla (1) se presentan los materiales que se necesitaron para poder realizar la práctica de laboratorio.

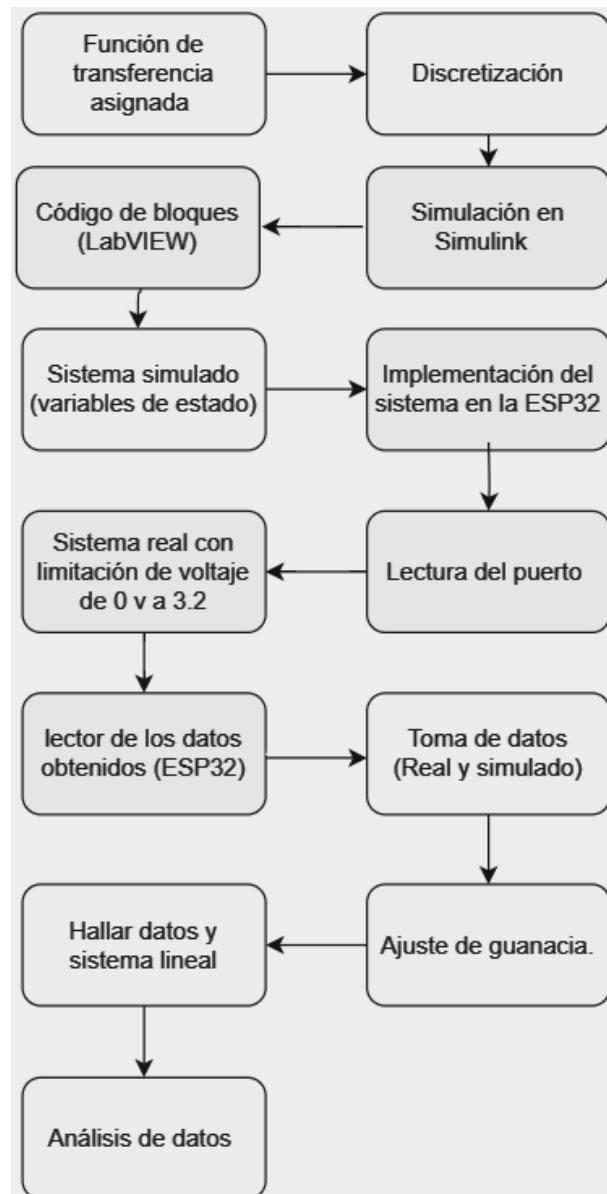


Figura 2. Diagrama flujo del procedimiento realizado

A. INSTRUMENTOS.

Tabla I. Lista de componentes		
Cantidad	Elemento	Valor / Características
2	ESP32	32 bits
1	MatLab	software
1	labVIEW	software

III. ANÁLISIS Y RESULTADOS

Punto 1. Simulación del sistema en simulink.

La discretización del sistema asignado en matlab con la función c2d permite discretizar el sistema, luego con ecuaciones de diferencia se multiplica por Z^{-3} debido a que es un sistema de tercer orden, luego de esto, se despeja la salida en función de la entrada en variables de estado con el fin de poder aplicar esta ecuación en el software LabVIEW.

El sistema asignado es:

$$G_s = \text{tf}([0.001776, 0.002211], [1, 0.414926, 0.055896, 0.002457]) \quad (0)$$

La función de transferencia que se obtuvo fue la siguiente:

```
>> Gz=c2d(Gs, 0.1)
```

$G_z =$

$$\frac{9.123e-06 z^2 + 1.323e-06 z - 8.281e-06}{z^3 - 2.959 z^2 + 2.918 z - 0.9594}$$

Sample time: 0.1 seconds
Discrete-time transfer function.

Figura 3. Función discretizada a 0.1 segundos.

Se introduce el comando format long para conseguir el numerador y el denominador de la función discretizada, el siguiente paso es implementar ecuaciones de diferencia para obtener una representación en variables de estado y de esa manera implementar el sistema en bloques, para la implementación en labVIEW.

$$G_z = \frac{9.122737060047859e-06 z^2 + 1.323450065995669e-06 z - 8.280530175789369e-06}{s^3 - 2.958807726042199z^2 + 2.918166567207017z - 0.959356434552821} \quad (1)$$

Se reemplazaron las constantes para hacer más fácil el despeje y la implementación del método de ecuaciones de diferencia.

$$\begin{aligned} b_0 &= 9.122737060047859e - 06 \\ b_1 &= 1.323450065995669e - 06 \\ b_2 &= 8.280530175789369e - 06 \\ a_1 &= -2.958807726042199 \\ a_2 &= 2.918166567207017 \\ a_3 &= -0.959356434552821 \end{aligned}$$

Siguiendo este orden encontramos que la ecuación 1 queda de la siguiente manera:

$$G_z = \frac{b_0 z^2 b_1 z b_2}{z^3 a_1 z^2 a_2 z a_3} \quad (2)$$

Continuando con el ejercicio se busca la s^n de mayor exponente y se multiplica tanto en el numerador como en el denominador por Z^{-3} , obteniendo:

$$G_z = \frac{b_0 z^2 + b_1 z + b_2}{z^3 + a_1 z^2 + a_2 z + a_3} \cdot \frac{z^{-3}}{z^{-3}} \quad (3)$$

Aplicando distributiva y planteando que G_z es igual a $Y(Z)/U(Z)$, es decir, salida sobre entrada, se despeja $Y(Z)$ para encontrar así la ecuación en variables de estado que representa el sistema de tercer orden.

$$\frac{Y(Z)}{U(Z)} = \frac{b_0 z^{-1} + b_1 z^{-2} + b_2 z^{-3}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}} \quad (4)$$

Despejando la ecuación 4 se obtiene la representación en variables de estado:

$$Y(Z) = b_0 u(z^{-1}) + b_1 u(z^{-2}) + b_2 u(z^{-3}) - a_1 y(z^{-1}) - a_2 y(z^{-2}) - a_3 y(z^{-3}) \quad (5)$$

Por último se reemplaza Z^n según el valor del exponente por $K-n$, esto representa el retraso que obtiene la variable en cada caso.

$$Y(Z) = b_0 u(k-1) + b_1 u(k-2) + b_2 u(k-3) - a_1 y(k-1) - a_2 y(k-2) - a_3 y(k-3) \quad (6)$$

Para corroborar que la discretización del sistema es correcta se realiza un montaje de bloques en simulink figura (4), el cual cuenta con un step, el encargado de generar un escalón de 0 a 1, luego se implementa un bloque transfer Fcn en el cual se indicará el numerador y el denominador de nuestro sistema ecuación (0), y otro con Discrete transfer Fcn en el cual se introduce la ecuación (1) y si le indica indica el tiempo de muestreo el cual fue 100 ms, por último la salida de los bloques va conectada a un scop en el cual ambas representaciones deben ser iguales

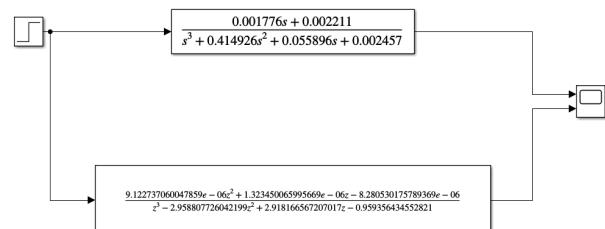


Figura 4. Diagrama de bloques simulación tiempo continuo y tiempo discreto.

Para verificar el comportamiento del diagrama de bloques se ajusta el tiempo de simulación a 100 segundos y se inicia la simulación, en la figura (5) se observa la respuesta de ambos sistemas, donde se ve claramente que coinciden y de esa manera se obtiene la comprobación del procedimiento matemático.

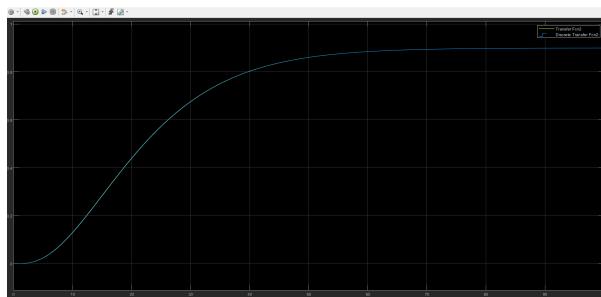


Figura 5. Respuesta de los sistemas simulación tiempo continuo y tiempo discreto.

Punto 2 . Simulación del sistema en LabVIEW.

La implementación de la ecuación (6) se lleva a cabo en LabVIEW obteniendo tanto el comportamiento simulado como el emulado. En la figura (6) se puede observar la implementación de dicha ecuación, en donde se utiliza un sumador, el cual a la entrada tiene los datos que pertenecen a la componente o con sus respectivos retrasos (feedback) y a la salida se obtiene la Y con sus constantes y retrasos respectivamente (feedback).

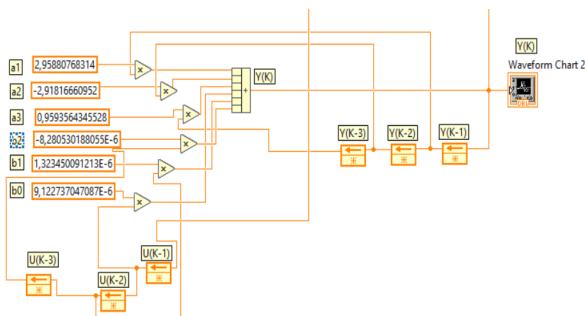


Figura 6. Simulación del sistema en variable de estados.

El diagrama de bloques de la figura (7) explica el procedimiento por medio del cual se puede representar el sistema (planta) con el fin de poder tener una representación real (emulada) del sistema.

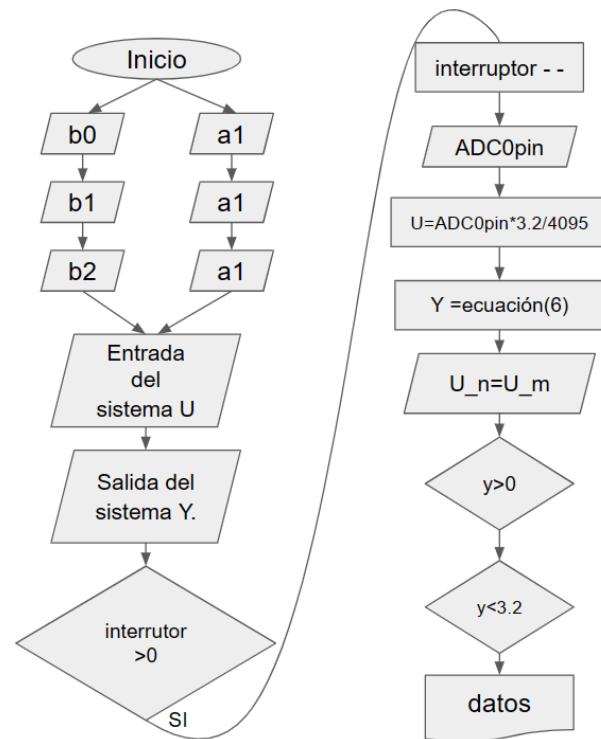


Figura 7. Diagrama de bloques para la implementación del sistema emulado.

Para la lectura del puerto se debe de sincronizar un reloj y además siempre debe ser limpiada la entrada con el objetivo de no agregar valores basura a la toma de datos. En la figura (8) se observa con mayor claridad lo que se debe llevar a cabo.

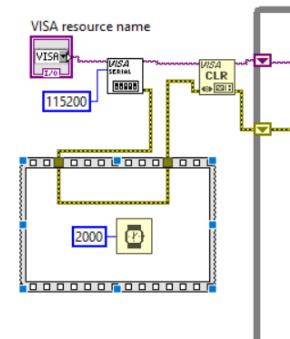


Figura 8. Diagrama de bloques lectura del puerto de la esp32(tarjeta de adquisición).

Debido a que es un sistema real tiene limitantes como en este caso lo es el voltaje, ya que la esp32 trabaja en el rango 0 voltios a 3.2 voltios, y la señal PWM genera valores de la resolución (4095) entre valores de 0 y 255. A continuación en la figura (9) se puede observar claramente el proceso que se ha realizado para asociar la resolución con escalones (dac) de 0 a 1 voltio.

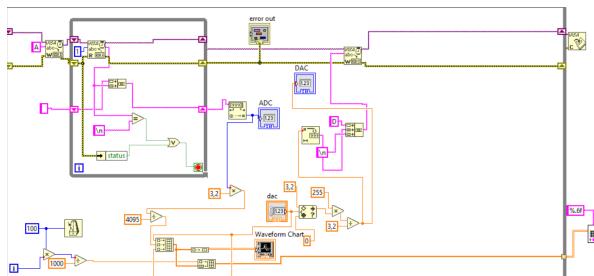


Figura 9. Sistema para la toma de datos en labVIEW para el sistema emulado.

Además se puede observar cómo se realiza la toma de los datos obtenidos, la inclusión en de un reloj y de la parte simulada como emulada (figura 6).

La segunda Esp32 se utiliza para la toma de datos, por esa razón se trabaja bajo la resolución y el voltaje que esta misma ofrece. Para realizar esta labor se ejecuta un código en el cual se pueden obtener los valores generados por la Esp 32 uno (planta) y enviarlos por adc labVIEW, para generar el escalón. Mediante la aplicación se modifica el dac de la Esp de lectura y está a su vez le envía la orden a la planta y a sí, se puede evidenciar el comportamiento del sistema emulado.

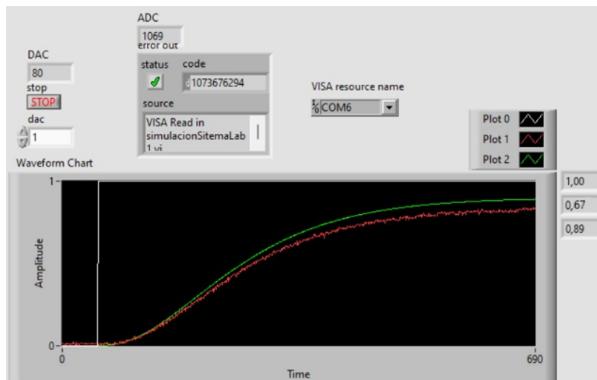


Figura 10. Respuesta del sistema para la toma de datos en LabVIEW.

En la figura (10) se presentan los resultados tomados en la primera toma de datos, en donde se puede observar claramente una diferencia entre lo que se espera y lo que se obtiene.

Para solucionar ese problema se agrega una ganancia a todos los componentes de "b" esto de manera empírica partiendo de la base de emulada/real y de esta manera se obtuvo que la ganancia que me generaba la respuesta más aproximada fue de 1,25.

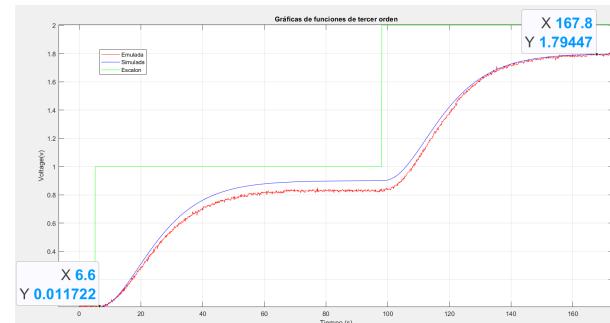


Figura 11. Puntos de equilibrio del sistema emulado y simulado.

Los puntos de equilibrio del sistema se pueden evidenciar en la figura (11), en donde se evidenció el comportamiento de la señal emulada y simulada ante dos escalones.

El siguiente paso es hallar los datos de sobreimpulso y tiempo de establecimiento, para realizar un análisis más detallado se planteó ver cada respuesta de manera individual.

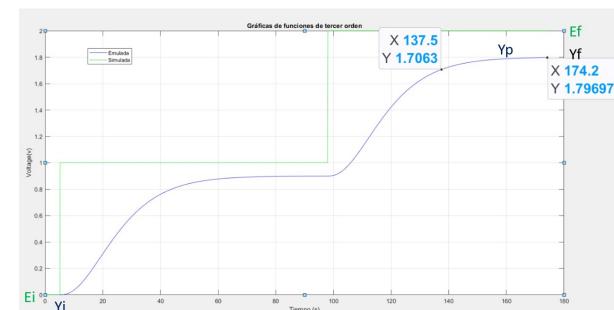


Figura 12. Toma de datos ganancia(G), tiempo de establecimiento(te) y sobreimpulso(Mp).

Para determinar la ganancia del sistema se realiza la resta en el numerador del punto final(Yf) menos el inicial(Yi) de la simulada y en el denominador la resta del dato final(Ef) del escalón menos el inicial del mismo(Ei), obteniendo como ganancia (G)=0.898485.

$$G = \frac{1.79697 - 0}{2 - 0} \quad (7)$$

A través del criterio del 5% para hallar el tiempo de establecimiento, aplicando la ecuación (8) se encontró que era de 132.4 segundos.

$$te = 137.5s - 5.1s \quad (8)$$

El sobre impulso se halla con la ecuación (9), pero debido a que Yp y Yf son los mismos el Mp es 0 para este sistema en especial.

$$Mp = \frac{Yp - Yf}{Yf - Yi} \quad (9)$$

Para obtener la representación lineal de la respuesta simulada del sistema implementado, con los datos obtenidos de los dos escalones de la figura (11) se saco solo la respuesta del escalón de 0 a 1 un Volt se implementa en un script en matlab y se siguieron los siguientes pasos:



Figura 13. Barra superior de herramientas en la sección de apps.

Se procede a abrir la herramienta system identification como se puede observar en la figura (13).

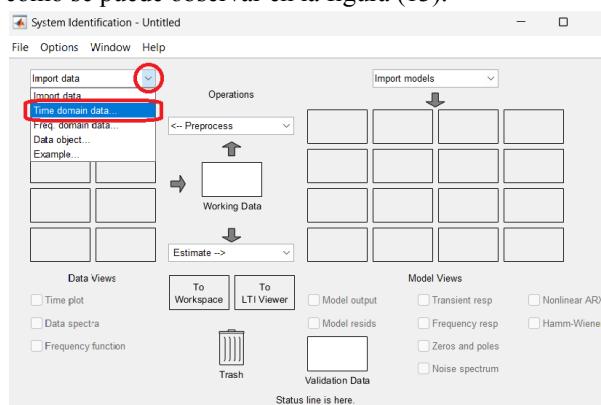


Figura 14. Se seleccionó el dominio en el cual se estaban subiendo los datos.

Con las variables obtenidas por el script se procedió a determinar las variables de entrada y de salida del sistema y más importante el tiempo de muestreo, como se puede observar en la figura (15).

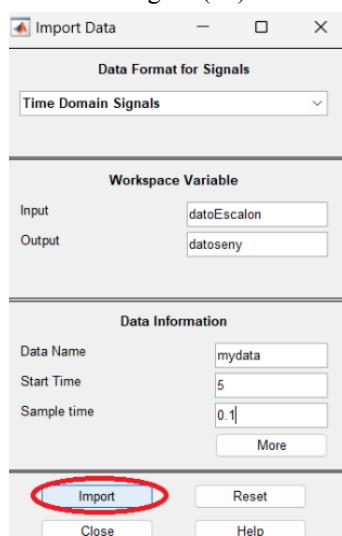


Figura 15. Se importan los datos ingresados.

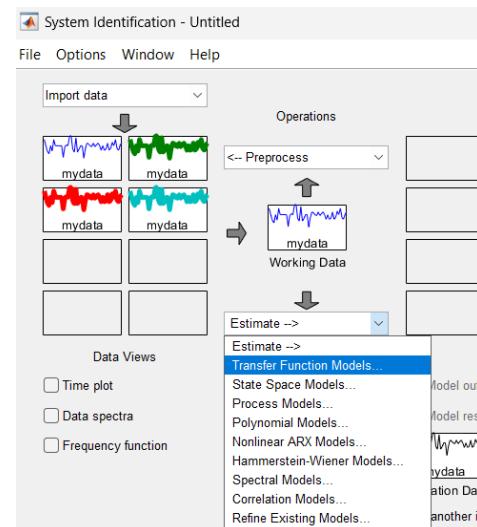


Figura 16. Se procede a realizar la estimación de la función de transferencia con esos datos.

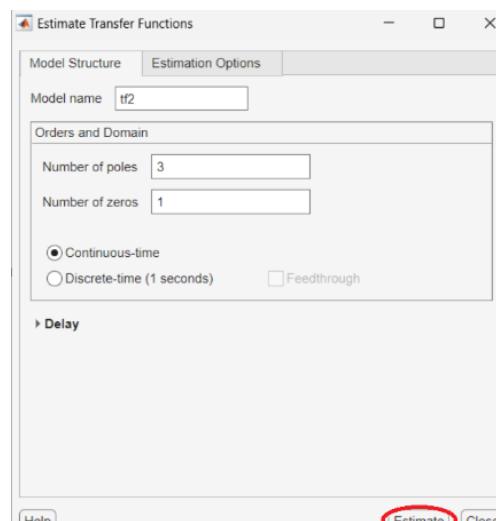


Figura 17. Se agregaron los polos y ceros para la representación lineal.

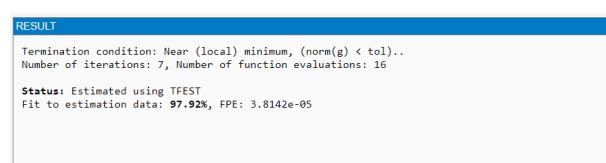


Figura 18. Se evidencia el porcentaje de similitud entre el resultado y los datos obtenidos.

$$\begin{aligned} & -0.0009835 \text{ s} + 0.003157 \\ & \hline \\ & s^3 + 0.5855 s^2 + 0.08252 s + 0.003805 \end{aligned}$$

Figura 19. Modelo lineal.

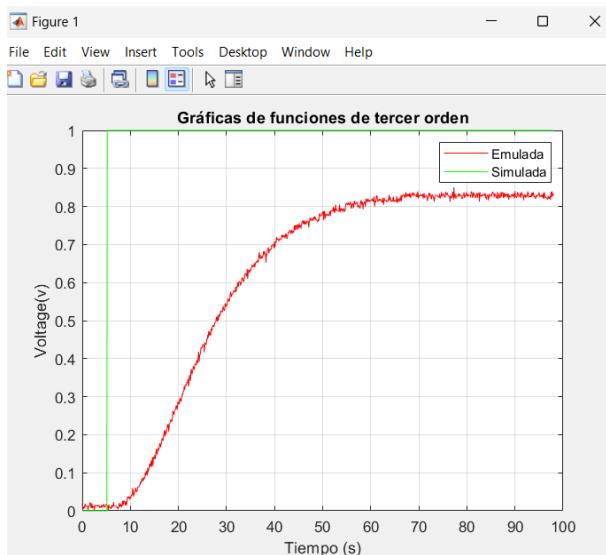


Figura 20. Representación escalón 0 a 1 voltio.

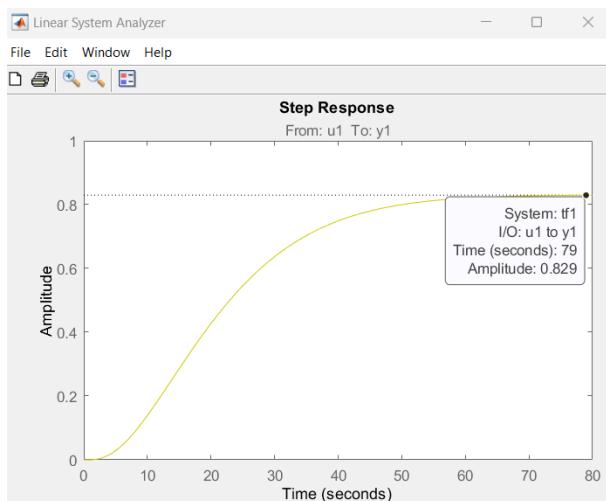


Figura 21. Representación lineal de 0 a 1 voltio.

Para analizar los resultados obtenidos se implementó un script en matlab con las tres respuestas obtenidas de nuestro sistema, en la figura (22) se pueden observar la gráfica de cada uno de los modelos ante un escalón.

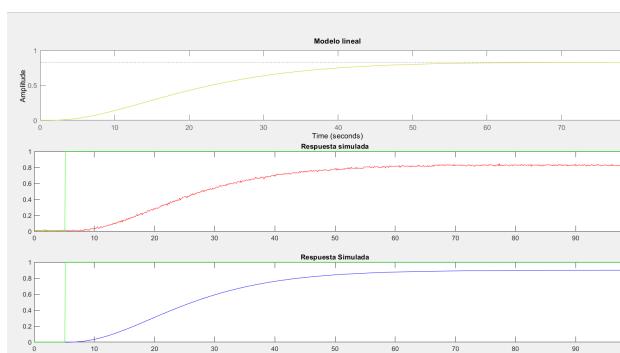


Figura 21. Representación lineal, emulada y simulada del sistema de tercer orden.

En la tabla (2) se presentan los resultados obtenidos de (G), (M_p) y (t_e) de cada una de las representaciones.

Tabla 2. Evaluación con cada una de las representaciones

	Modelo lineal	Emulada	Simulada
$G(v)$	0.829	0.82989	0.899322
M_p	0	0	0
t_e (s)	46.6	49.4	47.5

Se observa una pequeña diferencia entre los resultados de la emulada y modelo lineal con los resultados de la simulada. Esto se puede presentar por distintas razones, claramente el entorno de emulación no es ideal, pero los resultados entre la función lineal y la emulada son muy cercanos debido a que es una aproximación matemática, son muy congruentes con los datos obtenidos, además el error que tienen con la simulada en (G) es de 7.72% aproximadamente, y en el (t_e) es de 4 %.

IV. CONCLUSIONES

1. Las herramientas computacionales han facilitado el trabajo, ya que por medio de estas se pueden obtener modelos de x sistema y de esa manera implementar un control.
2. La adquisición de datos en la planta ante un escalón, permite conocer el comportamiento de la misma, es decir, que por medio de este proceso se puede adquirir los datos suficientes para hallar un modelo lineal (función de transferencia).
3. Torbolls en matlab como system identification, permiten obtener de manera empírica el modelo lineal de la planta.
4. Para encontrar datos de (G), (M_p) y (t_e) la simplicidad de graficar los datos en un script llevan a la obtención de los diferentes datos, de cada una de las tres representaciones.
5. En el sistema de lazo abierto sólo se pudo ajustar la ganancia, esto con el fin de sincronizar lo más que se pueda ambas representaciones y así poder obtener los puntos de equilibrio.
6. Este laboratorio proporciona una valiosa experiencia práctica en la emulación y simulación de sistemas de tercer orden, al tiempo que fomenta la adquisición de

habilidades esenciales para el diseño y análisis de sistemas de control automático.

V. BIBLIOGRAFÍA

- [1] MathWorks.(s.f). Matemáticas. Gráficas, Programación. MATLAB - El lenguaje del cálculo técnico (mathworks.com)

VI. ANEXOS

Lector: código de la Esp32 dos.

```
#define DAC 1 pin 25 //Pin del DAC1
#define ADC0pin 36 //Pin del ADC0

char selección; //Variable para caracter de selección
int sensor = 0; //Variable para leer el valor del ADC0
char control = 0; //Variable para recibir el valor del
DAC1

void setup() {
    Serial.begin(115200); //Inicia comunicación serial
}

void loop() {
    if (Serial.available() > 0) { //Verifica si hay un nuevo
byte para leer
        selección = Serial.read(); //Lee el nuevo byte

        if (selección == 'A') //"A' indica lectura del ADC
        {
            sensor = analogRead(ADC0pin); //Leer el dato del
ADC (12 bits)
            Serial.println(sensor); //Envía el valor leído
        };
        if (selección == 'D') //"D' indica que se escribirá en el
DAC
        {
            control = Serial.parseInt(); //Leer el número a
escribir
            dac Write(DAC 1 pin, control); //Escribir en el
DAC1
        };
    };
}
```