

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

**“DISEÑO DE UNA PLANTA DIDÁCTICA DE CONTROL  
DE FLUJO PARA EL LABORATORIO DE CONTROL  
AUTOMÁTICO FIEC – MACI”**

**INFORME DE PROYECTO INTEGRADOR**

Previo la obtención del Título de:

**INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN**

Presentado por:

NEY EDUARDO GARCIA SOTOMAYOR

CARLOS ROOSEVELT ROSERO RUGEL

GUAYAQUIL - ECUADOR

Año: 2020

## **DEDICATORIA**

El presente proyecto se lo dedico a mi esposa Darlen, a mis hijas Amelia y Amira por su apoyo y cariño que siempre me han brindado; a mis padres por el ejemplo y sacrificio que me ha llevado a cumplir con las metas que me propongo.

**Ney Garcia Sotomayor**

## **DEDICATORIA**

El presente proyecto se lo dedico a mi hija Ellie Rosero quien es la felicidad de todos los días de mi vida, a mis padres quienes me han apoyado siempre de manera incondicional, a mis amigos y amigas que siempre han estado apoyándome y aconsejándome en cada paso de mi carrera.

**Carlos Rosero Rugel**

## **AGRADECIMIENTOS**

A Dios por darme la salud y sabiduría para llevar a cabo este logro tan importante en mi vida

Mi especial agradecimiento a mi esposa quien supo alentarme y apoyarme hasta el final.

A mis padres de quienes siempre he recibido esa guía necesaria para cumplir con éxito mis estudios.

A mis amigos que de cualquier forma y circunstancia no dudaron en brindarme su ayuda.

**Ney Garcia Sotomayor**

## **AGRADECIMIENTOS**

Primero le agradezco a Dios por brindarme la oportunidad de estudiar esta hermosa carrera.

A mis padres por ser el apoyo incondicional en cada paso de mi vida.

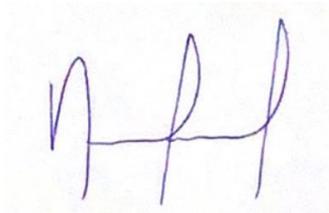
A mi hija por alegrarme cada día de mi vida y estar siempre sonriéndole a la vida.

A mis amigos y amigas por su apoyo durante la carrera.

**Carlos Rosero Rugel**

## DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; *Ney Eduardo Garcia Sotomayor* y *Carlos Roosevelt Rosero Rugel* damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”



---

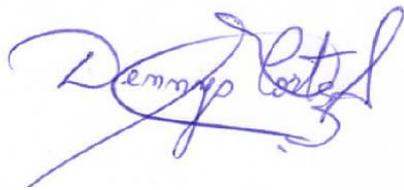
Ney Eduardo Garcia Sotomayor



---

Carlos Roosevelt Rosero Rugel

## EVALUADORES



---

**Ing. Dennys Dick Cortez Alvarez**  
PROFESOR DE LA MATERIA



---

**Ing. Efran Herrera Muentes**  
PROFESOR TUTOR

## RESUMEN

El presente proyecto está enfocado en realizar el diseño e implementación de un prototipo de planta didáctica de control de flujo para el Laboratorio de Control Automático de la ESPOL, ya que existe al momento una limitada cantidad de plantas didácticas para la demanda de estudiantes que las utilizan para realizar sus prácticas. Así mismo adquirir este tipo de material didáctico representa un alto costo para la Institución y es por esto que se ha optado por la construcción de esta planta y además que se considere para un bajo presupuesto.

Una planta de control de flujo está conformada por un tanque reservorio donde se almacena el agua y es de aquí que mediante una bomba sumergible se envía el agua al circuito pasando por un sensor de caudal, una válvula proporcional y un rotámetro que brinda la ventaja de que el flujo se pueda visualizar mientras este en funcionamiento la planta. Para el control de flujo aquellos sensores y actuadores que intervienen son controlados mediante una tarjeta de desarrollo, manipulada por un software de programación gráfico y que permite la comunicación de estos mediante una librería facilitando que se ejecute el controlador.

## **ABSTRACT**

This project is focused on carrying out the design and implementation of a prototype of a flow control didactic plant, for the automatic control laboratory of ESPOL, since there is currently a limited number of didactic plants for the demand of students who use them to carry out their practices. Likewise, acquiring this type of didactic material represents a high cost for the institution and that is why it has opted for the construction of this plant and that it is considered for a low budget.

A flow control plant is made up of a reservoir tank where the water is stored and it is from here that by using a submersible pump the water is sent to the circuit passing through a flow sensor, a proportional valve and a rotameter that provides the advantage of that the flow can be visualized while the plant is in operation for flow control, those sensors and actuators involved are controlled by a development card, manipulated by graphic programming software and which allows communication of these through a library, facilitating the execution of the controller.

# ÍNDICE GENERAL

DEDICATORIA.....	ii
DEDICATORIA.....	iii
RESUMEN .....	viii
ABSTRACT .....	ix
ÍNDICE GENERAL .....	x
ABREVIATURAS.....	xiii
SIMBOLOGÍA.....	xiv
ÍNDICE DE FIGURAS .....	xv
ÍNDICE DE TABLAS.....	xviii
CAPÍTULO 1 .....	1
1. INTRODUCCIÓN.....	1
1.1 Descripción del problema.....	2
1.2 Justificación del problema.....	2
1.3 Objetivos.....	2
1.3.1 Objetivo General .....	2
1.3.2 Objetivos Específicos.....	2
1.4 Marco teórico .....	3
1.4.1 Control de caudal.....	3
1.4.2 DAQ.....	3
1.4.3 Controlador PID .....	4
1.4.4 Arduino Mega 2560.....	4
1.4.5 Arduino Mega Uno .....	5
1.4.6 Puente H.....	5
1.4.7 Bomba de agua sumergible .....	5
1.4.8 Sensor de caudal .....	6

1.4.9	Servomotor .....	6
1.4.10	Válvula de compuerta .....	6
1.4.11	Rotámetro .....	7
1.4.12	Circuito Filtro Paso Bajos .....	7
CAPÍTULO 2 .....		8
2.	METODOLOGÍA .....	8
2.1	Elementos de la planta prototipo de control de flujo .....	9
2.1.1	Arduino mega 2560 .....	9
2.1.2	Arduino Uno .....	9
2.1.3	Bomba sumergible Comet Elegant .....	10
2.1.4	Sensor de flujo de agua YF-S201 .....	11
2.1.5	Servomotor MG995 .....	12
2.1.6	Válvula de compuerta .....	12
2.1.7	Rotámetro .....	13
2.1.8	Módulo Puente H .....	13
2.1.9	Fuente de voltaje .....	14
2.1.10	Diseño de Regulador de voltaje .....	14
2.1.11	Diseño de Filtro pasa bajos .....	15
2.2	Lazo de control del Sistema .....	16
2.3	Programación del bloque para la lectura del sensor de flujo y conversión a voltaje .....	16
2.4	Programación en diagrama de bloques en Simulink para el control de los elementos de la planta prototipo .....	18
2.5	Diagrama de bloques Principal .....	20
2.5.1	Programación interna del bloque “subsystem” para el control del .....	21
2.5.1.1	Asignación de la dirección de giro y estado de operación .....	21

2.5.1.2	Asignación de estado de operación de acuerdo a las circunstancias presentes en la planta. ....	22
2.5.1.3	Control de posición de la válvula en función del tiempo transcurrido. ....	24
2.6	Diagrama esquemático general de conexiones .....	25
2.7	Tablas de distribución de conexiones.....	26
CAPÍTULO 3 .....		27
3.	ANÁLISIS Y RESULTADOS .....	27
3.1	Curva característica de la planta .....	27
3.2	Identificación de la función de transferencia.....	30
3.3	Diseño del controlador de la planta .....	38
3.4	Prueba del controlador de la planta en lazo cerrado .....	39
3.5	Análisis de Costos.....	42
CAPÍTULO 4 .....		44
4.	CONCLUSIONES Y RECOMENDACIONES .....	44
4.1	Conclusiones .....	44
4.2	Recomendaciones .....	44
BIBLIOGRAFÍA .....		45
5.	ANEXOS.....	47

## **ABREVIATURAS**

ESPOL	Escuela Superior Politécnica del Litoral
FIEC	Facultad de Ingeniería en Electricidad y Computación
MACI	Maestría en Automatización y Control
FBD	Function Block Diagram
PID	Proportional, Integral, Derivative
IDE	Integrated development environment
PWM	Pulse Width Modulation
DAQ	Data acquisition
Po	Punto de Operación
Tarr	Tiempo de arranque
Tss	Tiempo de estabilización
P&ID	Piping and instrumentation diagram

## SIMBOLOGÍA

cm	centímetro
L/min	litros por minuto
L/H	litros por hora
V	voltio
A	amperio
mA	miliamperio
W	watt
Hz	hertz
KHz	kilohertz
''	pulgada

## ÍNDICE DE FIGURAS

Figura 1.1 Esquema del proceso de adquisición de datos .....	4
Figura 1.2 Esquema del control PID.....	4
Figura 1.3 Funcionamiento de la configuración puente H.....	5
Figura 1.4 Circuito filtro pasa bajos .....	7
Figura 2.1 Proceso de diseño del prototipo .....	8
Figura 2.2 Arduino Mega 2560.....	9
Figura 2.3 Arduino Uno .....	10
Figura 2.4 Bomba de agua sumergible ELEGANT .....	11
Figura 2.5 Sensor de flujo de agua .....	11
Figura 2.6 Servomotor MG995.....	12
Figura 2.7 Válvula de compuerta .....	12
Figura 2.8 Rotámetro .....	13
Figura 2.9 Módulo Puente H .....	14
Figura 2.10 Fuente de voltaje.....	14
Figura 2.11 Regulador de voltaje .....	15
Figura 2.12 Filtro pasa bajos.....	15
Figura 2.13 Diagrama P&ID del sistema .....	16
Figura 2.14 Código de programación de lector de Frecuencia de Arduino UNO .....	17
Figura 2.15 Librería Arduino del software de programación gráfica .....	18
Figura 2.16 Bloque "PWM" .....	19
Figura 2.17 Bloque "Analog Input" .....	19
Figura 2.18 Bloque "Continuous Servo Write" .....	20
Figura 2.19 Diagrama de bloques principal en programación de bloques .....	20
Figura 2.20 Diagrama de bloques 1 de bloque "subsystem". .....	21
Figura 2.21 Diagrama de bloques 2 del bloque "subsystem". .....	22
Figura 2.22 Diagrama de bloques 3 del bloque "subsystem". .....	24
Figura 2.23 Diagrama esquemático general de conexiones .....	26
Figura 3.1 Prototipo de Planta didáctica de Control de Flujo .....	27
Figura 3.2 Curva característica de la planta .....	28
Figura 3.3 Comportamiento del sistema.....	29

Figura 3.4 Diagrama de bloques en lazo abierto .....	30
Figura 3.5 Configuración del bloque step .....	31
Figura 3.6 Configuración del bloque Pulse Generator .....	31
Figura 3.7 Entrada vs Salida del Sistema en lazo abierto con el punto de operación .....	32
Figura 3.8 Entrada vs Salida del sistema en lazo abierto sin el punto de operación .....	33
Figura 3.9 Importación de datos .....	34
Figura 3.10 Rangos seleccionados .....	35
Figura 3.11 Proceso para la identificación de la planta .....	36
Figura 3.12 Modelo de proceso .....	36
Figura 3.13 Función de transferencia obtenida con datos reales .....	37
Figura 3.14 Función de transferencia obtenida .....	37
Figura 3.15 diseño del controlador Root Locus .....	38
Figura 3.16 Diseño del controlador step response .....	39
Figura 3.17 función de transferencia del controlador .....	39
Figura 3.18 Diagrama de bloques del controlador .....	40
Figura 3.19 Prueba del controlador .....	41
Figura 5.1 Diagrama de bloques .....	54
Figura 5.2 Bloque Step .....	56
Figura 5.3 Bloque Pulse Generator .....	57
Figura 5.4 Entrada vs Salida del Sistema en lazo abierto con el punto de operación .....	58
Figura 5.5 Entrada vs Salida del Sistema en lazo abierto sin el punto de operación .....	59
Figura 5.6 Selección de rango .....	60
Figura 5.7 Rangos seleccionados .....	60
Figura 5.8 Ventana System Identification .....	61
Figura 5.9 Selección de Process Models .....	61
Figura 5.10 Process Models .....	62
Figura 5.11 Proceso para la identificación de la planta .....	63
Figura 5.12 Ventana Model Output .....	64
Figura 5.13 Función de transferencia .....	64

Figura 5.14 Ingreso de Función de transferencia .....	66
Figura 5.15 Ingreso del comando Sisotool .....	66
Figura 5.16 Ventana Control System Designer .....	67
Figura 5.17 Ingreso de Función de transferencia .....	67
Figura 5.18 Ingresar especificaciones del controlador .....	68
Figura 5.19 Tiempo de estabilización .....	68
Figura 5.20 Porcentaje de overshoot .....	68
Figura 5.21 Añadir integrador.....	69
Figura 5.22 Añadir un cero.....	69
Figura 5.23 Diseño del controlador Root Locus .....	70
Figura 5.24 Diseño del controlador step response .....	70
Figura 5.25 Controlador PI.....	71
Figura 5.26 Diagrama de bloques en lazo cerrado.....	71
Figura 5.27 Bloque PID controller .....	72
Figura 5.28 Prueba del controlador .....	72

## ÍNDICE DE TABLAS

Tabla 2.1 Modos de operación del servomotor .....	22
Tabla 2.2 Descripción de pines del Arduino Uno.....	26
Tabla 2.3 Descripción de pines del Arduino Mega .....	26
Tabla 3.1 Datos experimentales de curva característica en lazo abierto .....	28
Tabla 3.2 Detalle de costos de materiales electrónicos.....	42
Tabla 3.3 Detalle de costos de materiales de construcción.....	42
Tabla 3.4 Detalle de costo total de fabricación.....	43

# CAPÍTULO 1

## 1. INTRODUCCIÓN

Hoy en día para el desarrollo de habilidades de un estudiante que realiza estudios de pregrado y posgrado es necesario el empleo de plantas didácticas que le permitan entender y comprender su funcionamiento, por lo que se disponen de cierta cantidad de dichas plantas y debido a esto se ha optado por diseñar una de estas plantas para reducir la demanda que existía en el laboratorio al momento de que los estudiantes realicen sus prácticas.

En el presente trabajo se dará a conocer el diseño de una planta didáctica que permita en control de flujo, y que para el desarrollo de esta se realizó una ingeniería inversa a una planta ya existente en el laboratorio de control automático, con el fin de desarrollar una planta de bajo costo y a su vez que sea de fácil uso para los estudiantes de pregrado y posgrado que acuden a dicho laboratorio.

## **1.1 Descripción del problema**

Adquirir plantas didácticas para uso académico representa un alto costo para la institución en muchos casos estos son mayores a equipos de similares características elaborados para la industrial común, más allá que la asistencia técnica, las mejoras y accesorios extras representan una dependencia tecnológica constante a lo largo de la vida útil del equipo. Dada la gran demanda de estudiantes que deben utilizar estos equipos para el desarrollo de las habilidades prácticas y los diferentes tipos de procesos a analizar, los costos crecen más aún convirtiéndose en adquisiciones prohibitivas y en estos precisos momentos frente a una profunda crisis económica nacional, se vuelve imperativa la producción local.

## **1.2 Justificación del problema**

Se plantea desarrollar un prototipo de planta didáctica con el fin de reducir de manera considerable el costo de adquisición de dichas plantas para la institución, reduciendo la dependencia tecnológica con mano de obra local, en este caso con el diseño e implementación de una planta de control de flujo, con la finalidad de que los estudiantes puedan realizar sus prácticas y proyectos de manera correcta y puedan aplicar y contrastar con sus conocimientos teóricos aprendidos en clases.

## **1.3 Objetivos**

### **1.3.1 Objetivo General**

Diseñar e implementar una planta didáctica de control de flujo aplicando ingeniería inversa para su respectivo uso en el laboratorio de Control Automático.

### **1.3.2 Objetivos Específicos**

- Realizar un diseño electrónico basado en ingeniería inversa realizadas a la planta existente en la institución con el fin de una mejora de esta y que se adapte a las necesidades del cliente. De esta manera obtener mejores resultados al momento de realizar las respectivas prácticas por lo estudiantes de pregrado y posgrado.
- Proponer un diseño estructural mecánico de la planta de tal forma que el control de caudal se ejecute sobre el dispositivo con diferentes materiales aislantes contenidos dentro del mismo.

- Verificar el correcto funcionamiento del prototipo de la planta a través de la identificación del sistema y el diseño del controlador; con el fin de que sea aplicado como base para el desarrollo de prácticas de los estudiantes.
- Elaborar una guía de práctica para el desarrollo de las actividades académicas de los diferentes cursos clientes del Laboratorio de Control Automático.

## **1.4 Marco teórico**

### **1.4.1 Control de caudal**

El control de flujo es importante desde el punto de vista industrial, para mantener en régimen de operación diferentes equipos (Sistemas de refrigeración, operaciones de separación) y para efectos contables, en los cuales importa el suministro de ciertos fluidos (gasoductos).

Dinámicamente es un proceso especial, en que la respuesta a cambios en la posición del elemento final de control (válvula, bomba), es rápida y prácticamente depende de las demoras en los instrumentos. Otra característica la constituyen las oscilaciones permanentes debidas a turbulencia, que genera cambios cíclicos en las tomas de presión con lo cual el sistema está sometido a ruidos de alrededor de 1 cps.

Esta situación exige la utilización de modos de control integral proporcional, ajustados en condiciones desfavorables, para disminuir el valor de la ganancia y minimizar las oscilaciones del instrumento sensor.

### **1.4.2 DAQ**

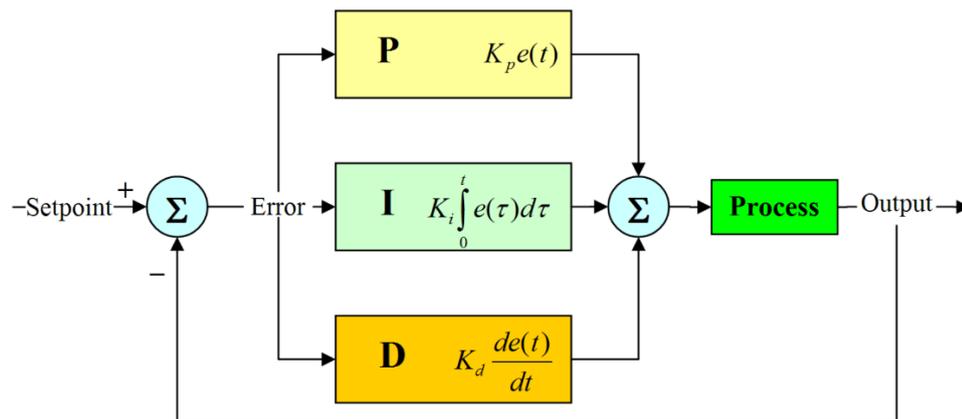
Es el que permite la interacción entre los sensores y el computador del usuario, por lo que su función principal es digitalizar las señales que le son comunicadas desde los sensores a objeto que puedan ser interpretadas por el PC. Asimismo, incorporan un circuito que les permite adecuar las señales a algún protocolo de comunicación, y con el mismo propósito incluyen también un convertidor que permite transformar las señales analógicas en señales digitales, y finalmente, un bus de comunicación a PC, que corresponde al canal que el dispositivo utilizará para comunicarse con el PC, el que puede ser a través de Ethernet, Wi-Fi, ModBus u otro canal de comunicación.



**Figura 1.1 Esquema del proceso de adquisición de datos**

### 1.4.3 Controlador PID

El control PID es un mecanismo de control que a través de un lazo de retroalimentación permite regular la velocidad, temperatura, presión y flujo entre otras variables de un proceso en general. El controlador PID calcula la diferencia entre nuestra variable real contra la variable deseada. En sistemas de bombeo, regularmente nos interesa mantener la presión o flujo constante, por lo tanto, el control PID mide la diferencia entre la presión en la tubería y la presión requerida y actúa variando la velocidad del motor para que podamos tener nuestra presión o flujo constante.



**Figura 1.2 Esquema del control PID**

### 1.4.4 Arduino Mega 2560

El Arduino Mega 2560 es una placa de desarrollo basada en el microcontrolador ATmega2560. Tiene 54 entradas/salidas digitales (de las cuales 15 pueden ser usadas como salidas PWM), 16 entradas analógicas, 4 UARTs, un cristal de 16Mhz, conexión USB, jack para alimentación DC, conector ICSP, y un botón de

reinicio. La placa Mega 2560 es compatible con la mayoría de shields compatibles para Arduino UNO.

#### 1.4.5 Arduino Mega Uno

Arduino UNO es una placa basada en el microcontrolador ATmega328P. Tiene 14 pines de entrada/salida digital (de los cuales 6 pueden ser usando con PWM), 6 entradas analógicas, un cristal de 16Mhz, conexión USB, conector jack de alimentación, terminales para conexión ICSP y un botón de reinicio. Tiene toda la electrónica necesaria para que el microcontrolador opere, simplemente hay que conectarlo a la energía por el puerto USB ó con un transformador AC-DC.

#### 1.4.6 Puente H

Es un circuito electrónico utilizado para controlar un motor DC. El nombre “puente H” se debe a que la forma de circuito se asemeja a la letra H. Está constituido por 4 transistores que permiten controlar de manera bidireccional al motor. Su funcionamiento consiste básicamente en 2 estados de operación: para que el motor gire en sentido hacia delante, Q1 y Q4 deberán estar activados mientras que Q2 y Q3 desactivados; por otro lado, para que el motor gire en sentido inverso, Q2 y Q3 deberán estar activados mientras que Q1 y Q4 desactivados.

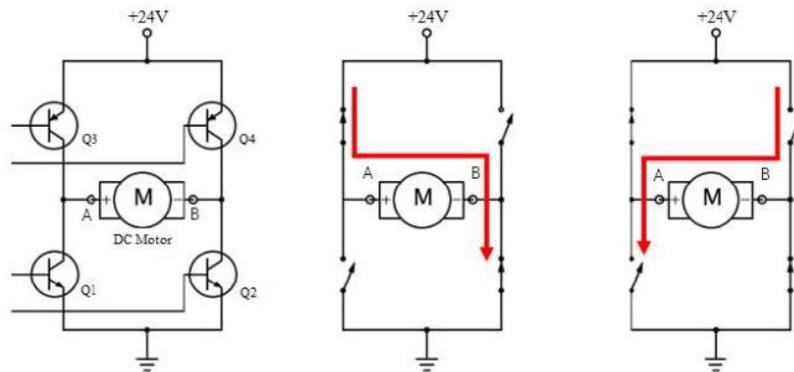


Figura 1.3 Funcionamiento de la configuración puente H

#### 1.4.7 Bomba de agua sumergible

La función básica de toda bomba sumergible es la de transportar un fluido. De ahí se sigue que las características de funcionamiento y los medios de acción sobre el fluido deben de estar estrechamente relacionados con las características del fluido en cuestión. La inmensa mayoría de las aplicaciones de las bombas es bombear fluidos de ahí la distinción entre bombeo de gases y líquidos. Por

definición ambos son fluidos con las mismas características que afectan al bombeo.

Una bomba sumergible, como indica el nombre es una bomba que se sumerge en un líquido. Las bombas sumergibles contienen un impulsor sellado a su carcasa que permite bombear el líquido en el que se encuentran sumergidas hacia el exterior. La ventaja de este tipo de bomba es que puede proporcionar una fuerza de bombeo significativa pues no depende de la presión de aire externa para hacer ascender el líquido.

#### **1.4.8 Sensor de caudal**

Es un instrumento para la medición de caudal o gasto volumétrico de un fluido. El caudal es la cantidad de líquido o fluido (volumen) que circula a través de una tubería por unidad de tiempo, por lo general se expresa en: litros por minutos (l/m), litros por hora (l/h), metros cúbicos por hora (m<sup>3</sup>/h), etc.). Los caudalímetros suelen colocarse directamente en la tubería que transporta el fluido. También suelen llamarse medidores/sensores de caudal, medidores de flujo o flujómetros.

#### **1.4.9 Servomotor**

Son dispositivos de accionamiento para el control de precisión de velocidad, par motor y posición. Constituyen un mejor desempeño y precisión frente a accionamientos mediante convertidores de frecuencia, ya que éstos no nos proporcionan control de posición y resultan poco efectivos en bajas velocidades.

Es un servomotor, aquel que contiene en su interior un encoder, conocido como decodificador, que convierte el movimiento mecánico (giros del eje) en pulsos digitales interpretados por un controlador de movimiento. También utilizan un driver, que en conjunto forman un circuito para comandar posición, torque y velocidad.

#### **1.4.10 Válvula de compuerta**

Una válvula de compuerta es una herramienta cuya función es elevar o abrir una compuerta o cuchilla para permitir el paso de fluidos. Estas compuertas o cuchillas pueden ser redondas o rectangulares. Cuentan con un sello que se logra mediante la colocación de un disco en dos áreas distribuidas. Las caras de éste pueden ser paralelas o tener forma de cuña.

La compuerta generalmente está cerrada. Cuando la válvula está abierta, la compuerta se coloca en el sobrante de la válvula lo que deja una abertura del mismo tamaño de la tubería en la que está instalada, aunque hay válvulas de paso completo y paso restringido.

Por normal general, las válvulas de compuerta se utilizan cuando se precisa que haya un caudal rectilíneo o una restricción mínima del paso de éste.

#### 1.4.11 Rotámetro

Es un caudalímetro industrial que se usa para medir el caudal de líquidos y gases. El rotámetro consiste en un tubo y un flotador. La respuesta del flotador a los cambios de caudal es lineal, y un rango de flujo de 10 a 1 es estándar. Es posible aún más flexibilidad a través del uso de ecuaciones de correlación. El rotámetro es popular debido a que tiene una escala lineal, un rango de medición relativamente largo y una baja caída de presión. Es fácil de usar y simple de instalar y mantener.

#### 1.4.12 Circuito Filtro Paso Bajos

El filtro pasa bajos, es un filtro que se caracteriza por permitir el paso de las frecuencias más bajas y atenuar las frecuencias más altas. Para su construcción se emplean resistencias en serie con capacitores o inductores, tomando el nombre de filtro pasa bajos RC tal como se aprecia en la figura 1.4, o filtro pasa bajos RL.

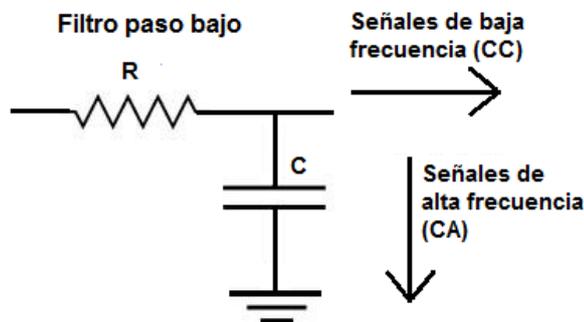


Figura 1.4 Circuito filtro pasa bajos

# CAPÍTULO 2

## 2. METODOLOGÍA

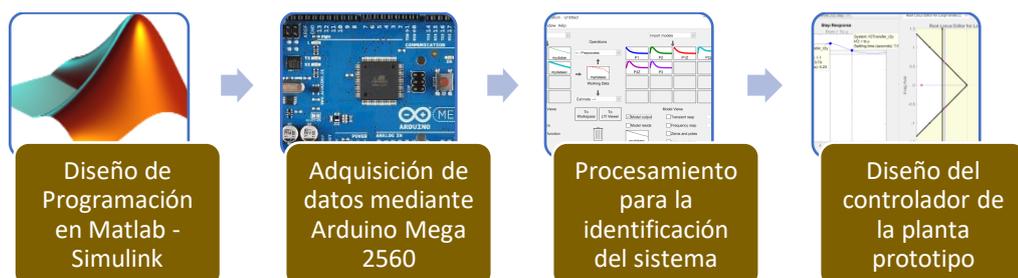
Para realizar el diseño e implantación de un prototipo de la planta didáctica de control de flujo, se analizó la planta existente marca GUNT que se encuentra en el laboratorio de control automático de la FIEC, aplicando la técnica de ingeniería inversa se logró conocer su funcionamiento, elementos que la constituyen, como se encuentran conectados entre sí y de esta manera crear un diseño nuevo para la planta de control de flujo.

En la figura 2.1 se puede observar el proceso de diseño en la que para su desarrollo se optó por tomar como base el Software Matlab, el cual posee un toolbox llamado Simulink que sirve para simular un sistema mediante programación en diagrama de bloques.

Para la etapa de adquisición de datos fue necesario contar con un dispositivo que sea compatible con el Software Matlab; por lo que se empleó una tarjeta de desarrollo Arduino Mega 2560 ya que esta también permite interactuar con los elementos que constituyen la planta y a su vez con el objetivo de diseñar un prototipo de bajo costo.

Una vez que se logró la interacción de Simulink con los elementos de la planta con el uso del Arduino se procedió a realizar la identificación del sistema donde se obtuvo la función de transferencia.

Posteriormente con dicha función se procedió a diseñar el controlador con el uso de la herramienta gráfica Sisotool de Matlab, cumpliendo con los diferentes requisitos y logrando que funcione de manera óptima la planta.



**Figura 2.1 Proceso de diseño del prototipo**

## 2.1 Elementos de la planta prototipo de control de flujo

### 2.1.1 Arduino mega 2560

Especificaciones técnicas:

- Microcontrolador: ATmega 2560
- Voltaje de Operación: 5V DC
- Voltaje de entrada: 7 a 12V DC
- Memoria FLASH: 256 KB
- Memoria SRAM: 8 KB
- Memoria EEPROM: 4 KB
- Entradas y salidas digital: 54 (14 proveen salida PWM)
- Entrada analógica: 16
- Corriente en Pines E/S digital: 40 mA
- Corriente en Pin de 3.3V: 50 mA
- Velocidad de reloj: 16 MHz

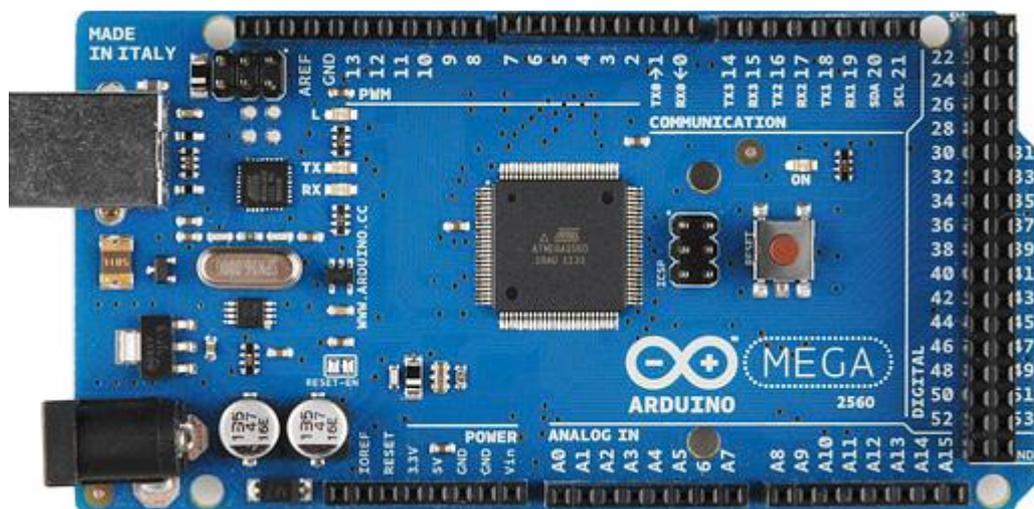


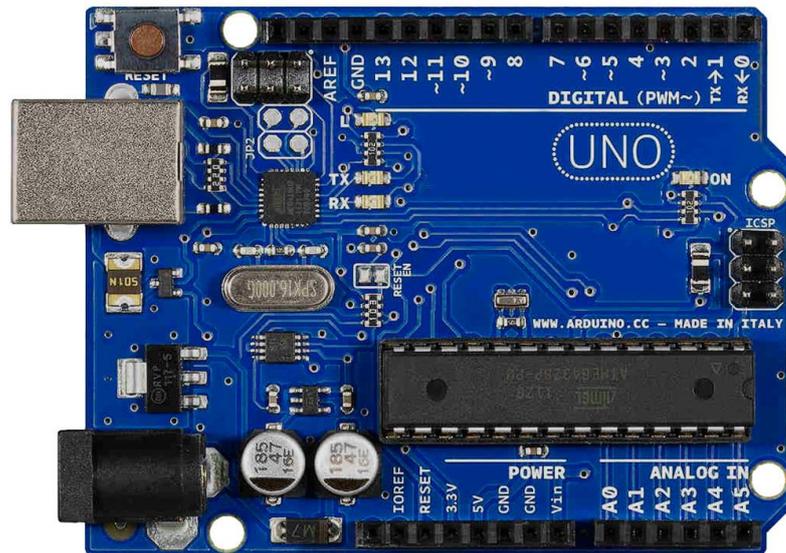
Figura 2.2 Arduino Mega 2560

### 2.1.2 Arduino Uno

Especificaciones técnicas:

- Microcontrolador: ATmega 328
- Voltaje de Operación: 5V DC
- Voltaje de entrada: 7 a 12V DC

- Memoria FLASH: 32 KB
- Memoria SRAM: 2 KB
- Memoria EEPROM: 1 KB
- Entradas y salidas digital: 14 (6 proveen salida PWM)
- Entrada analógica: 6
- Corriente en Pines E/S digital: 40 mA
- Corriente en Pin de 3.3V: 50 mA
- Velocidad de reloj: 16 MHz



**Figura 2.3 Arduino Uno**

### **2.1.3 Bomba sumergible Comet Elegant**

Especificaciones técnicas:

- Modelo: Comet Elegant
- Capacidad de bombeo: hasta 10 L/min a 0.55 bar de presión
- Voltaje de Alimentación: 12V DC
- Corriente Máxima: 2.2 A
- Potencia consumida: 15 a 24 W



**Figura 2.4 Bomba de agua sumergible ELEGANT**

#### **2.1.4 Sensor de flujo de agua YF-S201**

Especificaciones técnicas:

- Modelo: YF-S201
- Rango de trabajo: 1 a 30L/min
- Voltaje de funcionamiento: 5 a 18V DC
- Máxima presión de agua: 2.0 MPa
- Consumo máximo de corriente: 15mA a 5V
- Voltaje de salida: 5V TTL



**Figura 2.5 Sensor de flujo de agua**

### 2.1.5 Servomotor MG995

Especificaciones técnicas:

- Modelo: MG995
- Torque: hasta 5.5Kg/cm
- Voltaje de alimentación: 4.8 a 6V DC
- Corriente Máxima: 1 A



Figura 2.6 Servomotor MG995

### 2.1.6 Válvula de compuerta

Especificaciones técnicas:

- Modelo: Bronce Edesa
- Diámetro: 1/2"
- Número de vueltas: 3.5 vueltas



Figura 2.7 Válvula de compuerta

### 2.1.7 Rotámetro

Especificaciones técnicas:

- Material: plástico.
- Tipo: tubo de plástico.
- Modelo: LZS-15.
- Nivel de medición: 25-250 l/h.
- Precisión: 4 %.
- Temperatura de funcionamiento: 0~140.0 °F.
- Presión de funcionamiento: no más de 0,6 MPa.
- Compatible con: tubo DN15 (1/2").

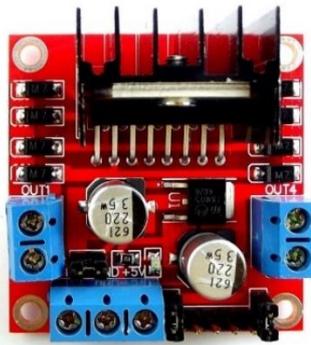


Figura 2.8 Rotámetro

### 2.1.8 Módulo Puente H

Especificaciones técnicas:

- Chip: L298N
- 2 canales
- Voltaje de alimentación: 12 V
- Voltaje lógico: 5 V
- Capacidad de corriente: 2A
- Consumo de corriente: 0 a 36 mA



**Figura 2.9 Módulo Puente H**

### **2.1.9 Fuente de voltaje**

Especificaciones técnicas:

- Modelo: S-120-12
- Fuente AC – DC.
- AC Entrada: 100 – 120V  $\pm$  15%.
- DC Salida: 12V – 10A.



**Figura 2.10 Fuente de voltaje**

### **2.1.10 Diseño de Regulador de voltaje**

En la figura 2.11 se puede observar el diagrama esquemático del regular de voltaje diseñado el cual está conformado por un LM7805, dos capacitores uno de 300nF y otro de 100nF; en el que en su entrada se conecta a 12V para lograr obtener 5V en la salida; empleado para alimentar al servomotor y al puente H colocando en el pin IN1 la salida de 5V y en el pin IN2 se conecta la tierra.

## Regulador de voltaje

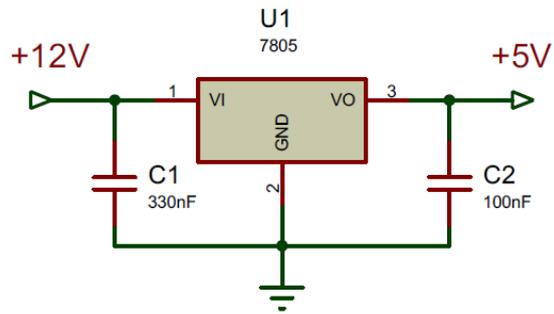


Figura 2.11 Regulador de voltaje

### 2.1.11 Diseño de Filtro pasa bajos

En la figura 2.12 se puede observar el diagrama esquemático del filtro pasa bajos diseñado el cual esta conformado por una resistencia de 100k y un capacitor polarizado de 10uF; empleado para que la señal de salida PWM (PIN 3) del Arduino Uno se parezca más a una señal analógica y que esta pueda ser leída por el Arduino Mega.

## Filtro pasa bajos

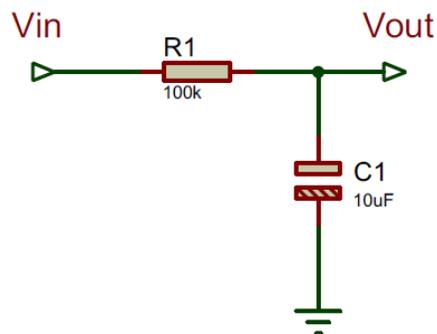
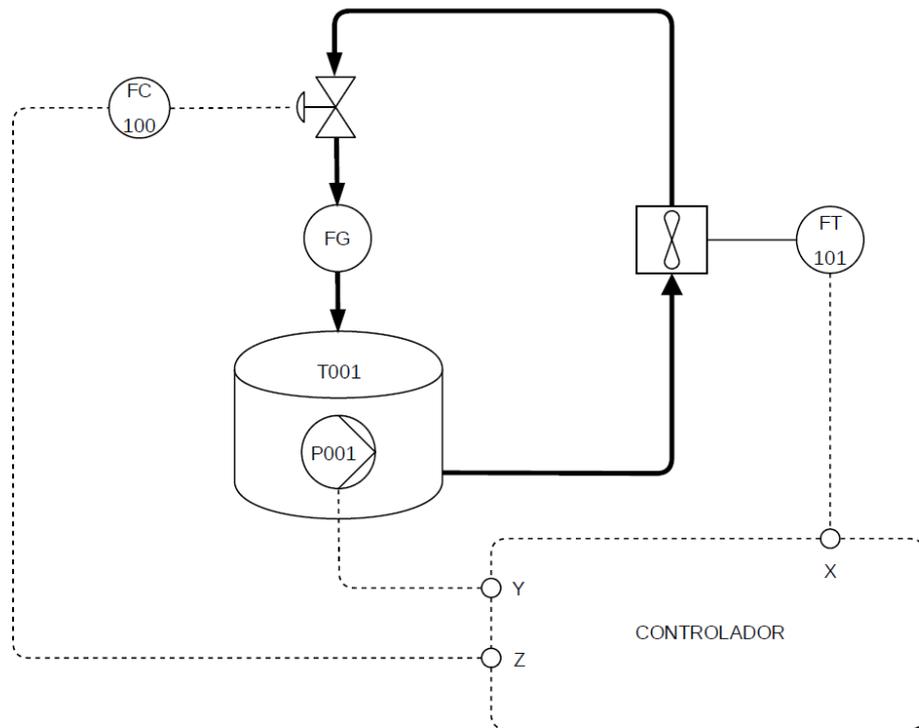


Figura 2.12 Filtro pasa bajos

## 2.2 Lazo de control del Sistema



**Figura 2.13 Diagrama P&ID del sistema**

En la figura 2.13 se muestra el diagrama P&ID donde puede apreciarse toda la instrumentación involucrada, así como el lazo de control de flujo para la presente planta. La bomba sumergible que desplaza el agua desde el reservorio hacia el circuito cerrado en la que se encuentra inicialmente el sensor de flujo de agua el cual se encarga de indicar el valor de flujo de agua presente en el circuito, a continuación se encuentra la válvula de compuerta en la que está acoplado un servomotor, esta corresponde a la señal de perturbación es aquí en donde el flujo de agua va a variar en función a la apertura o cierre de la válvula y finalmente se podrá apreciar de manera visual en el rotámetro el valor de flujo de agua presente en el circuito previo a regresar al reservorio cumpliendo el lazo cerrado del sistema y lo que conlleva el propósito de una planta didáctica de control de flujo.

## 2.3 Programación del bloque para la lectura del sensor de flujo y conversión a voltaje

En la figura 2.14 se muestra el código empleado para la lectura de la frecuencia (número de pulsos generados) del sensor de flujo FY-S201 y así mismo la conversión de este valor a Voltaje. Para su desarrollo se usó el software Arduino

para la programación de la tarjeta de desarrollo Arduino Uno, en la que se usó como método de programación las interrupciones para la lectura de los pulsos generados en el interior del sensor en función al caudal que pasa por el mismo, cada pulso activará una interrupción ejecutando una función de incremento de un contador para así obtener la frecuencia; para la conversión a valores de voltaje se consideró la máxima frecuencia de 28 Hz que lee el sensor es decir el flujo máximo de la planta prototipo y el máximo valor de salida PWM de 255 que equivale a 5V, luego este valor resultante de la conversión es enviado por una salida PWM.

```

volatile int flow_frequency; // Measures flow sensor pulses
unsigned int l_hour; // Calculated litres/hour
unsigned char flowsensor = 2; // Sensor Input
unsigned long currentTime;
unsigned long cloopTime;
void flow () // Interrupt function
{
    flow_frequency++;
}
void setup()
{
    pinMode(flowsensor, INPUT);
    digitalWrite(flowsensor, HIGH); // Optional Internal Pull-Up
    Serial.begin(9600);
    attachInterrupt(0, flow, RISING); // Setup Interrupt
    sei(); // Enable interrupts
    currentTime = millis();
    cloopTime = currentTime;
}
void loop ()
{
    currentTime = millis();
    // Every second, calculate and print litres/hour
    if(currentTime >= (cloopTime + 1000))
    {
        cloopTime = currentTime; // Updates cloopTime
        // Pulse frequency (Hz) = 7.5Q, Q is flow rate in L/min.
        l_hour = (flow_frequency * 60 / 7.5); // flowrate in L/hour
        Serial.println(flow_frequency);
        Serial.println((flow_frequency*255)/28);
        analogWrite(3, (flow_frequency*255)/28);
        flow_frequency = 0; // Reset Counter
        Serial.print(l_hour, DEC); // Print litres/hour
        Serial.println(" L/hour");
    }
}

```

**Figura 2.14 Código de programación de lector de Frecuencia de Arduino UNO**

## 2.4 Programación en diagrama de bloques en Simulink para el control de los elementos de la planta prototipo

La siguiente programación en diagrama de bloques se utilizará para el control de los diferentes instrumentos que componen el prototipo de planta de control de flujo tales como: bomba de agua, sensor de flujo de agua, válvula manual acoplada a un servomotor.

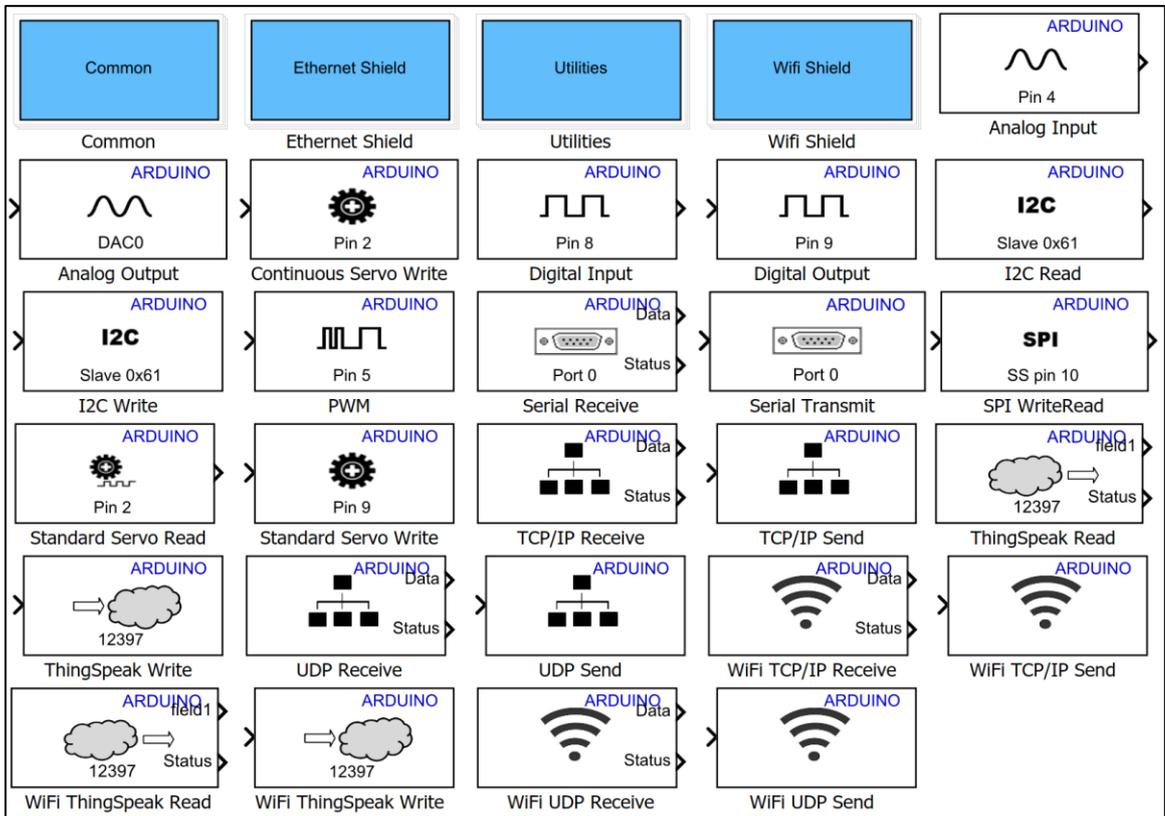


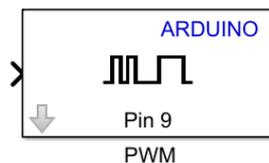
Figura 2.15 Librería Arduino del software de programación gráfica

En la figura 2.15 se puede apreciar los diferentes bloques disponibles en la librería de Arduino que pueden emplearse en Simulink para el uso de los pines de entradas y salidas del Arduino, es por ello que previamente es necesario instalar dicha librería en el software Matlab.

Para la programación en diagrama de bloques de la planta de control de flujo se usaron los siguientes:

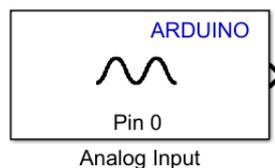
**PWM:** Este bloque emplea modulación por ancho de pulso (PWM) para entregar diferentes niveles de voltaje asemejándose a una señal de salida analógica. Para lograr los diferentes valores de voltaje se cambia el ciclo de trabajo de la señal cuadrada presente a la salida de un pin PWM del hardware Arduino. En la entrada de este bloque se ingresa un valor que va de 0 a 255. Con un valor de 255 que

representa un ciclo de trabajo del 100% se tiene en el pin de salida PWM el máximo nivel de voltaje (5V). Por otra parte, con un valor de 0 que representa un ciclo de trabajo del 0% se tiene el mínimo nivel de voltaje (0V). Para valores intermedios como es el caso del valor 153 que representa un ciclo de trabajo del 60% se tiene en el pin de salida PWM un voltaje de 3V. El parámetro "pin number" se configura de acuerdo con el pin correspondiente a la salida PWM que se utilice del hardware Arduino.



**Figura 2.16 Bloque "PWM"**

**Analog Input:** Permite medir el voltaje de un pin de entrada analógica del Arduino. La medición es un valor de 10 bits que va desde 0 a 1023. El parámetro "pin number" se configura de acuerdo con el pin correspondiente a la entrada analógica que se utilice del hardware Arduino.



**Figura 2.17 Bloque "Analog Input"**

**Continuous Servo Write:** Permite controlar la velocidad y dirección de giro de un servomotor de rotación continua. Este bloque puede recibir valores entre -90 a 90. Con un valor de -90 el servomotor de rotación continua gira a su máxima velocidad hacia una dirección, mientras que con un valor de 90 gira a su máxima velocidad en la dirección contraria. Para que el servomotor se detenga se emplea el valor de 0. Para establecer velocidades intermedias se debe ingresar valores entre -90 y 0 o de 0 a 90 de acuerdo con la dirección en la que se desea que gire. El parámetro "pin number" se configura de acuerdo con el pin correspondiente a la salida digital PWM que se utilice del hardware Arduino.



Figura 2.18 Bloque "Continuous Servo Write"

## 2.5 Diagrama de bloques Principal

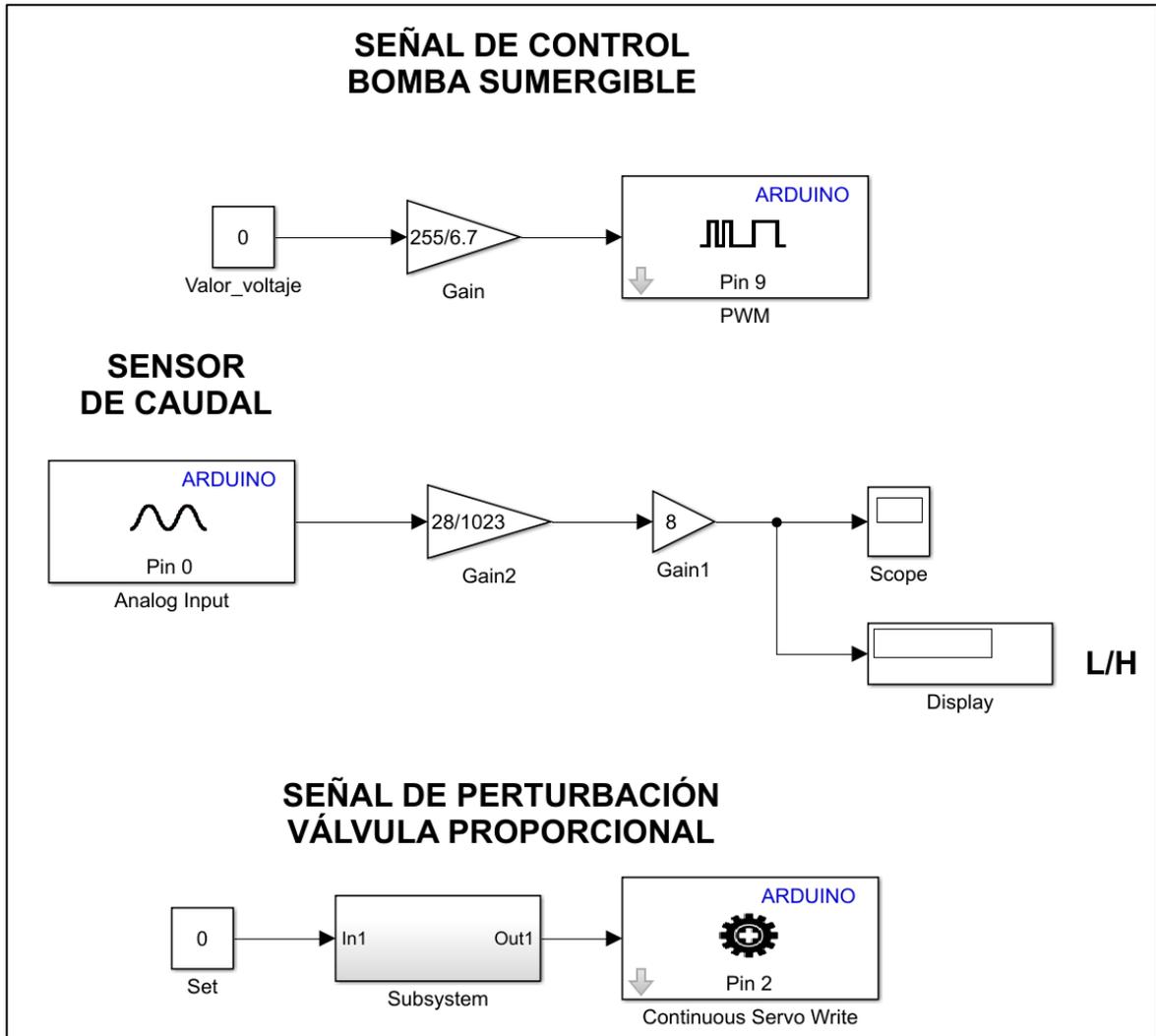


Figura 2.19 Diagrama de bloques principal en programación de bloques

En la figura 2.19 se puede observar el diagrama de bloques principal para el control de los elementos que intervienen en la planta; para la señal de control que corresponde a la bomba sumergible se utilizó un bloque PWM de la librería Arduino, esta señal es enviada desde el PIN 9 del Arduino Mega 2560 hacia el pin ENA del módulo controlador de puente H; para indicar la magnitud de la señal analógica que se desea enviar a dicho módulo, se utiliza el bloque constante en

el que se puede ingresar valores entre 0 y 6.7 que corresponden a valores de voltaje siendo 6.7V el máximo voltaje al que trabaja la bomba, luego pasa por un bloque de ganancia donde el valor de voltaje se lo transforma en un valor analógico siendo 255 el máximo.

Para la señal del sensor de caudal se utilizó un bloque de entrada analógica en la que se leen valores entre 0 a 1023 por lo que después de este bloque se tiene un bloque de ganancia donde se pasan estos valores a valores de frecuencia donde 28 Hz es la máxima frecuencia que va a leer el sensor de caudal posterior a eso se tiene una ganancia de 8 que corresponde al factor de conversión de frecuencia a litros por hora.

Para la señal de perturbación que corresponde a la válvula proporcional, la que se implementó juntar una válvula manual y un servomotor, se utilizó un bloque continuous servo write de la librería Arduino que permite controlar al servomotor, esta señal es enviada desde el PIN 2 al pin de control del servomotor, para indicar la magnitud de la señal analógica que se desea enviar al servomotor, se utiliza el bloque constante en el que se puede ingresar valores entre 0 y 10 para establecer la apertura que se desea en la válvula, luego este valor ingresa a un bloque subsystem en el que se encuentra toda la programación que conlleva al control de posición del servomotor.

## 2.5.1 Programación interna del bloque “subsystem” para el control del servomotor acoplado a la válvula.

### 2.5.1.1 Asignación de la dirección de giro y estado de operación

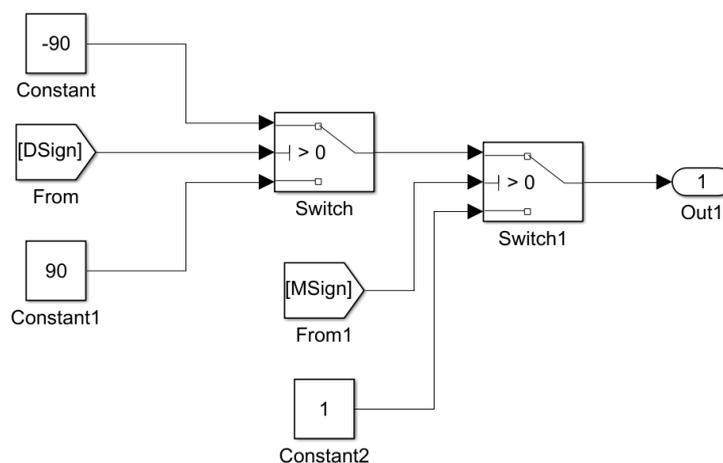


Figura 2.20 Diagrama de bloques 1 de bloque “subsystem”.

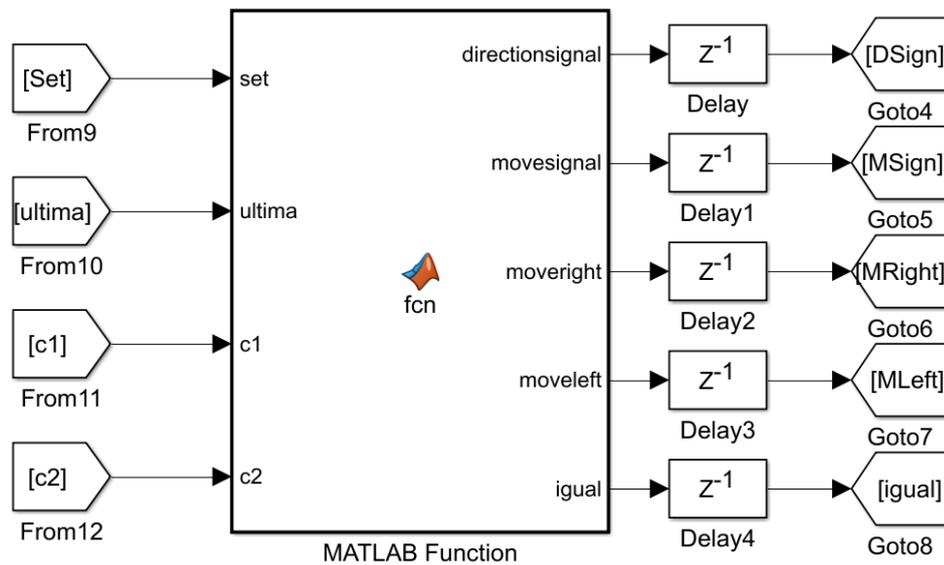
El diagrama de bloques mostrado en la figura 2.20 consiste hacia que dirección debe girar el servomotor y el estado en el que se encuentra ya sea ON u OFF; para ello se utiliza el bloque switch en cascada, presentándose los siguientes casos de operación:

**Tabla 2.1 Modos de operación del servomotor**

	MSign (Move Signal)	DSign (Direction Signal)
Motor OFF	0	0
Motor OFF	0	1
Motor ON horario	1	0
Motor ON antihorario	1	1

**Nota:** Las señales MSign y DSign empleada en esta etapa provienen de la segunda etapa descrita a continuación.

**2.5.1.2 Asignación de estado de operación de acuerdo a las circunstancias presentes en la planta.**



**Figura 2.21 Diagrama de bloques 2 del bloque “subsystem”.**

El diagrama de bloques de la figura 2.21 se encarga de la asignación del estado de operación considerando las circunstancias presentes en la planta. El bloque “Function” involucra la siguiente programación:

```
function[directionsignal,movesignal,moveright,moveleft,igual]=
fcn(set,ultima,c1,c2)
movesignal = 0;
igual = 0;
```

```

moveright = 0;
moveleft = 0;
directionsignal = 0;
if set == ultima || c1==1 || c2==1
    movesignal = 0;
    igual = 1;
    moveright = 0;
    moveleft = 0;
    directionsignal = 0;
else
    movesignal = 1;
    igual = 0;
    if set > ultima
        moveleft = set-ultima;
        moveright = 0;
        directionsignal = 0;
    else if set < ultima
        moveright = ultima - set;
        moveleft = 0;
        directionsignal = 1;
    end;
end;
end;
end

```

A la entrada de esta función se ingresan las señales:

“**set**” = posición hacia donde debe moverse la válvula.

“**ultima**” = es una señal que sirve de memoria para guardar la última posición de la válvula.

“**c1**” = señal proveniente de la etapa tres para indicar que la cuenta del primer contador ha culminado.

“**c2**” = señal proveniente de la etapa tres para indicar que la cuenta del segundo contador ha culminado.

A la salida de esta función se tiene las siguientes señales:

“**directionsignal**” = señal cuya función es indicar en qué sentido debe girar el servomotor.

“**movesignal**” = señal cuya función es indicar si el motor debe encenderse o apagarse.

“**moveright**” = señal habilitadora que permite el incremento de uno de los contadores que forman parte de la tercera etapa de este subsistema. Además,

guarda el valor de cuanto debe girar la válvula desde la posición actual hasta la posición deseada cuando el giro es horario.

“**moveleft**” = señal habilitadora que permite el incremento del otro contador que también forma parte de la tercera etapa de este subsistema. Además, guarda el valor de cuanto debe girar la válvula desde la posición actual hasta la posición deseada cuando el giro es antihorario.

“**igual**” = señal auxiliar que tiene un valor de 1 cuando la válvula a alcanzado la posición deseada por el usuario.

La operación que realiza esta función es establecer el estado de operación del servomotor de acuerdo con las circunstancias presentes en la planta. Es así como:

-Si “set” es igual a “ultima”, es decir, la válvula se encuentra en la posición deseada por el usuario; entonces el servomotor se apagará.

-Si “set” es diferente de “ultima”, es decir, la válvula debe desplazarse desde la posición actual hasta la posición requerida por el usuario; entonces el servomotor girará en el sentido necesario.

### 2.5.1.3 Control de posición de la válvula en función del tiempo transcurrido.

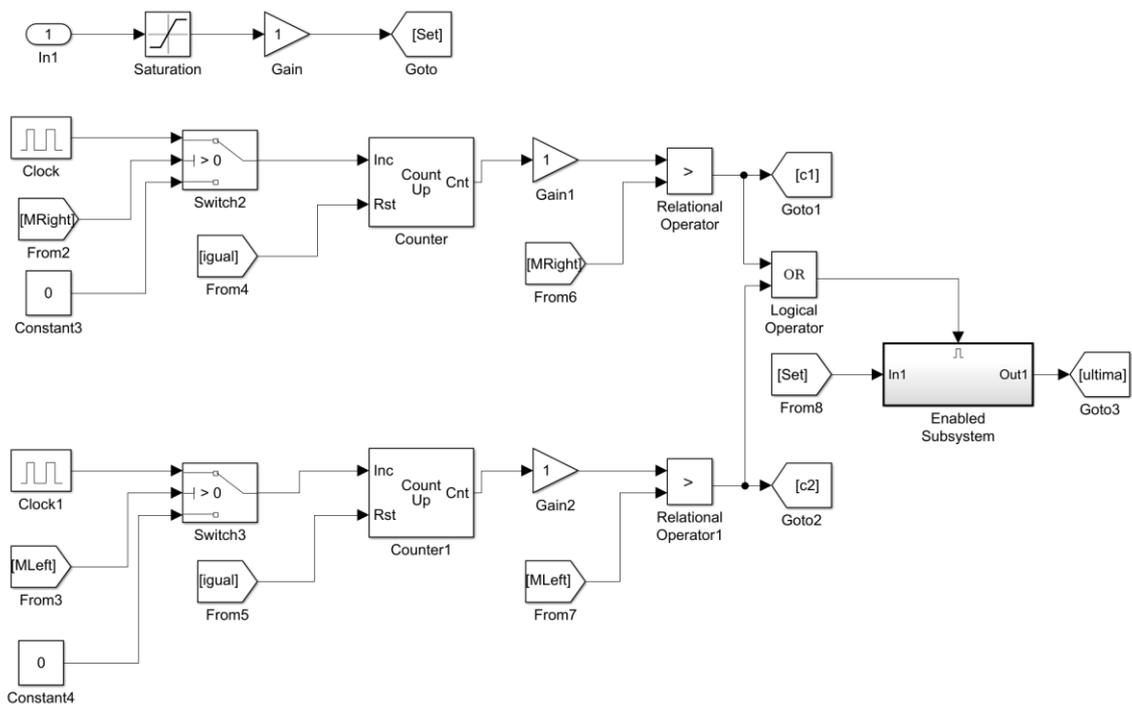


Figura 2.22 Diagrama de bloques 3 del bloque "subsystem".

La figura 2.22 muestra la programación que controla la posición de la válvula en función del tiempo transcurrido; para ello se utilizan dos contadores que simulan la función de temporizadores, por lo que es necesario el uso de señales cuadradas para que incrementen el valor del contador; únicamente empezarán a contar cuando la señal “MRight” (“moveright”) sea mayor que 0 ó la señal “MLeft” (“moveleft”) sea mayor que 0. Una vez que transcurra el tiempo necesario para que la válvula llegue a la posición deseada, se activará la salida de uno de los contadores para que la etapa dos haga que el servomotor se detenga. Estos contadores también permiten actualizar la última posición de la válvula. (Arboleda & Guamán, 2019)

## **2.6 Diagrama esquemático general de conexiones**

En la figura 2.23 se muestra el diagrama esquemático general de conexiones entre los elementos electrónicos que forman parte del prototipo de la planta de control de flujo; donde la tarjeta de desarrollo Arduino Mega se utiliza como el controlador principal del sistema permitiendo realizar el respectivo control de señales y adquiriendo datos. De esta tarjeta se envía una señal PWM (PIN 9) hacia el módulo del puente H, la cual cumplirá con la función de modificar la velocidad de trabajo de la bomba de agua en base a la modulación de ancho de pulso recibido en el pin ENA; también se envía otra señal PWM (PIN 2) que se conecta al servomotor para el control de la válvula de compuerta a la cual se encuentra acoplado; y por medio de una entrada analógica (PIN A0) se adquiere la señal enviada desde el Arduino que corresponde a la lectura del sensor de flujo siendo esta señal filtrada a través de un filtro pasa bajos para que su interpretación corresponda a una señal analógica.

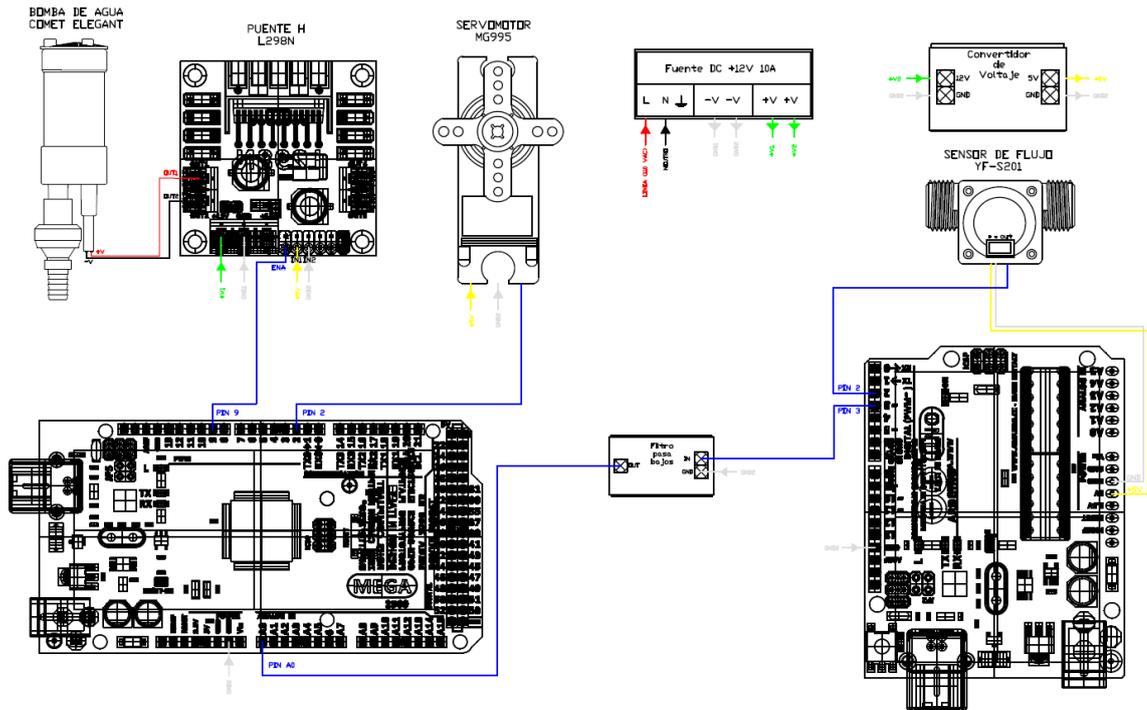


Figura 2.23 Diagrama esquemático general de conexiones

## 2.7 Tablas de distribución de conexiones

En las siguientes tablas se muestra la distribución de todos aquellos pines que intervienen en las tarjetas de adquisición usadas.

Tabla 2.2 Descripción de pines del Arduino Uno

Nº PIN	DESCRIPCIÓN
2	Conectado al pin OUTPUT del sensor de flujo de agua
3	Conectado al pin IN del filtro pasa bajos
5V	Conectado al pin + del sensor de flujo de agua
GND	Conectado al pin – del sensor de flujo de agua
GND	Conectado a GND2

Tabla 2.3 Descripción de pines del Arduino Mega

Nº PIN	DESCRIPCIÓN
A0	Conectado al pin OUT del filtro pasa bajos
2	Conectado al pin OUTPUT del servomotor
9	Conectado al pin ENA del módulo puente H
GND	Conectado a GND2

# CAPÍTULO 3

## 3. ANÁLISIS Y RESULTADOS

En este capítulo se presentan los resultados experimentales obtenidos del prototipo de la planta de control de caudal. Utilizando el software de programación matemática y gráfica Matlab; el mismo que nos permite obtener la función de transferencia necesaria para obtener el controlador del sistema el cual servirá operar la planta de manera correcta. En la figura 3.1 se observa la construcción del prototipo de la planta desarrollada.



**Figura 3.1 Prototipo de Planta didáctica de Control de Flujo**

### 3.1 Curva característica de la planta

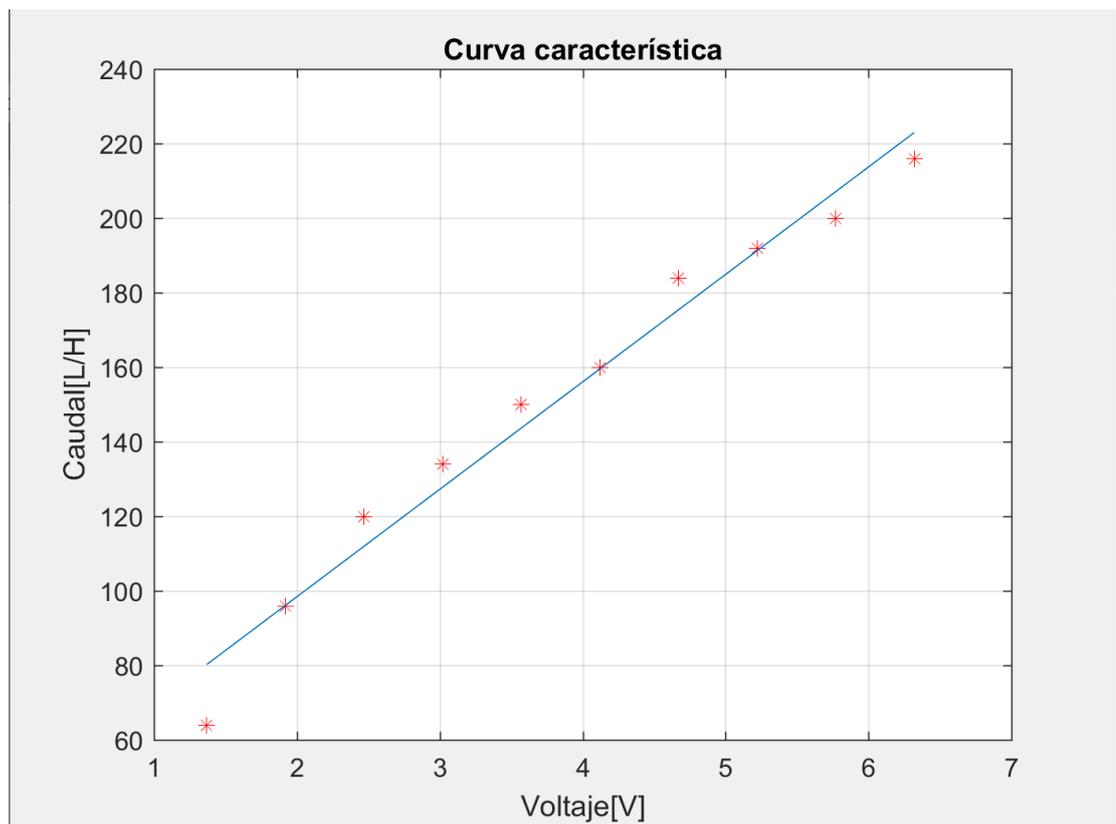
Para la identificación de la función de transferencia primero se debe hallar la curva característica la cual permite encontrar el comportamiento en estado estable de la planta. La programación realizada sirvió para localizar el mejor punto de operación con la que se trabaja haciendo una variación ( $\Delta V$ ) de 0.55 en su voltaje para observar de esta manera como trabaja la planta en ese rango.

El experimento realizado consiste en analizar los diferentes cambios de caudal (variable de salida) con respecto al voltaje de la bomba (variable de entrada) siendo esta la variable manipulada. En la tabla 3.1 se puede observar los datos experimentales obtenidos para graficar la curva característica del sistema;

logrando notar en la figura 3.2 la curva característica de la planta hecha con los diferentes valores experimentales dados.

**Tabla 3.1 Datos experimentales de curva característica en lazo abierto**

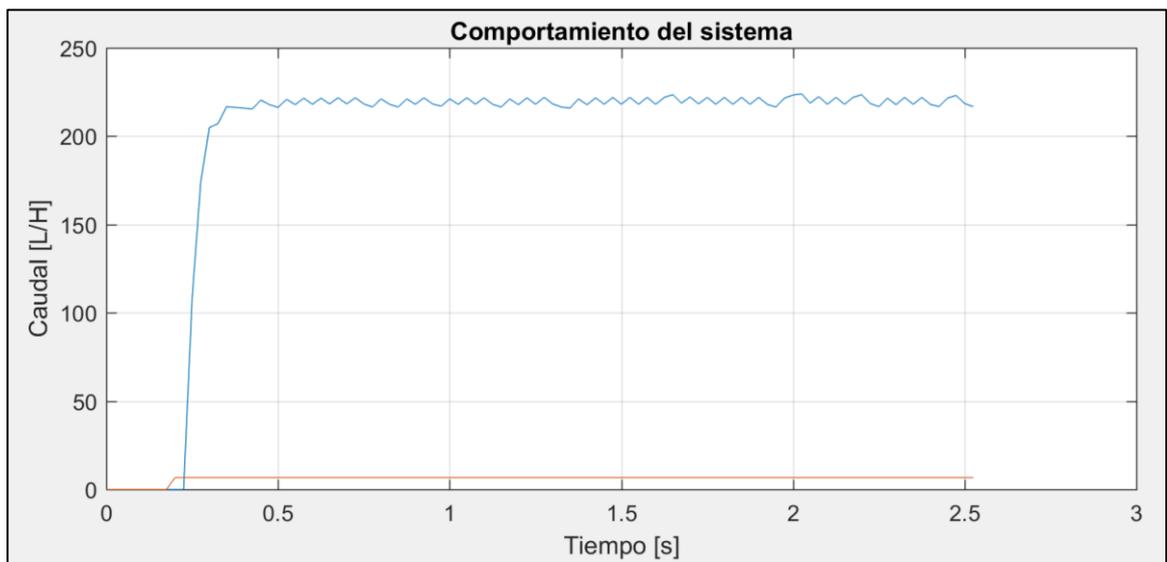
Voltaje V (variable de entrada)	Caudal L/H (variable de salida)
1.37	64
1.92	96
2.47	120
3.02	134
3.57	150
4.12	160
4.67	184
5.22	192
5.77	200
6.32	216



**Figura 3.2 Curva característica de la planta**

De la curva característica elaborada experimentalmente se obtiene el punto de operación escogido en función a la que se presente de forma lineal en la gráfica; por lo que se tomó como punto de operación  $Po=4.12$  el mismo que es utilizado para la obtención del tiempo de estabilización y el tiempo de arranque.

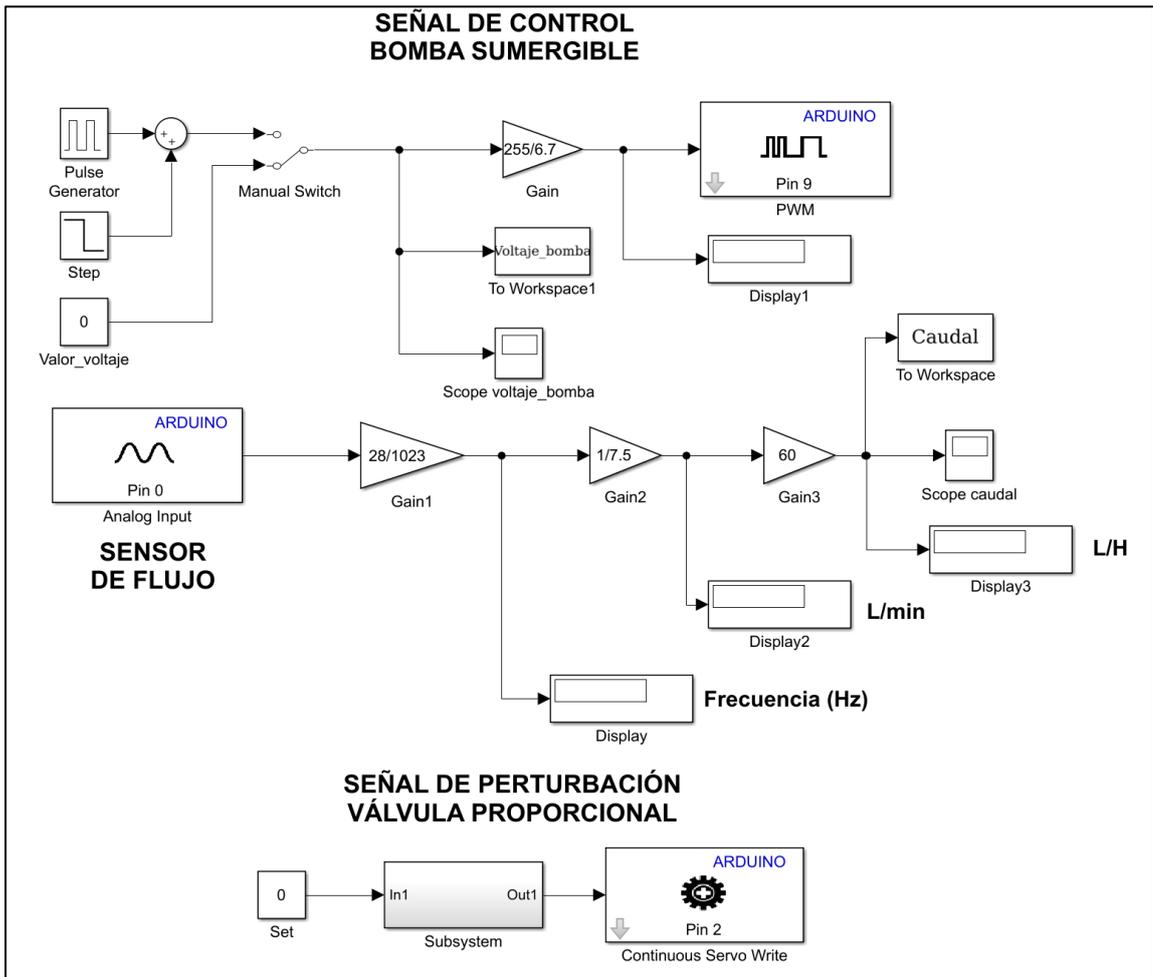
Utilizando los datos experimentales de la figura 3.3 se observan datos importantes como el tiempo de estabilización y el tiempo de arranque del sistema ya que  $tarr$  y  $tss$  son los valores que deben ser ingresados en el sistema para obtener la función de transferencia correcta.



**Figura 3.3 Comportamiento del sistema**

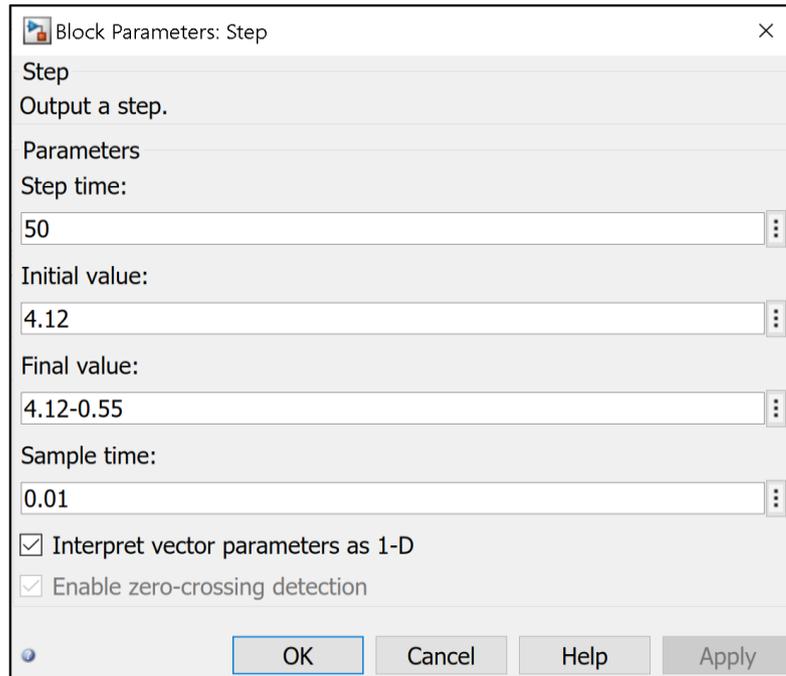
El tiempo de estabilización obtenido es  $tss=8$  segundos y el tiempo de arranque es  $tarr= 50$  segundos.

### 3.2 Identificación de la función de transferencia

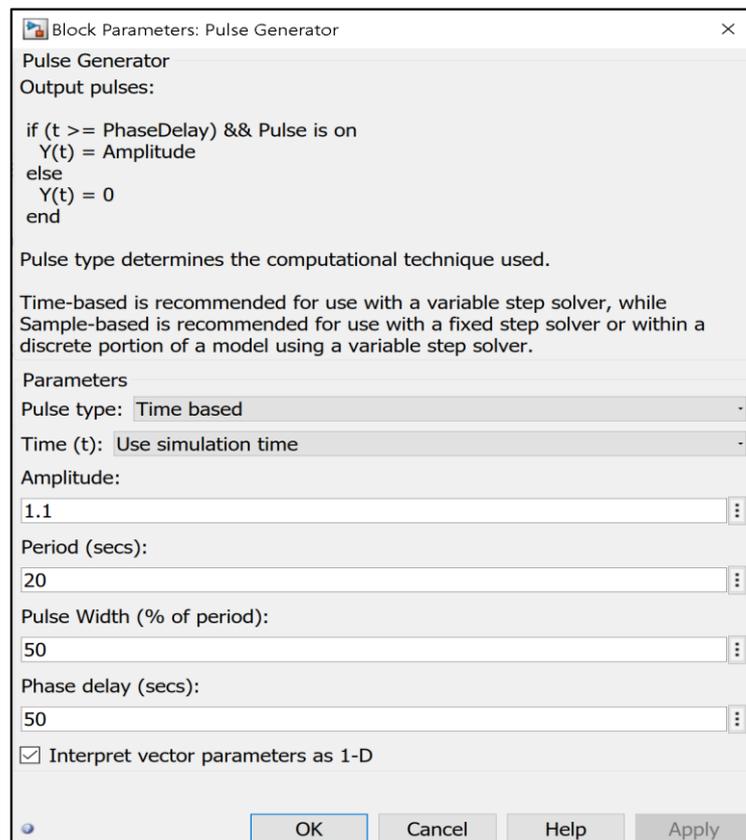


**Figura 3.4 Diagrama de bloques en lazo abierto**

En la figura 3.4 se observa el diagrama de bloques en lazo abierto utilizado para realizar la identificación de la planta, en la cual se configura dos bloques los mismos que son de mayor importancia para realizar el experimento. Primero se configura el bloque “step” como se observa en la figura 3.5 se coloca un  $tarr=50$  segundos, el valor inicial es mi punto de operación  $Po=4.12$ ; por lo tanto, se colocó mencionado valor y el valor final es la resta entre el  $Po$  y  $\Delta V$ , también se configura el bloque “pulse generator” como se observa en la figura 3.6. en la que la amplitud equivale a dos veces el  $\Delta V$ , se coloca el periodo de 20 segundos con el objetivo de observar el funcionamiento de la planta y el retardo de fase equivale al tiempo de arranque utilizado que es  $tarr=50$  segundos.



**Figura 3.5 Configuración del bloque step**

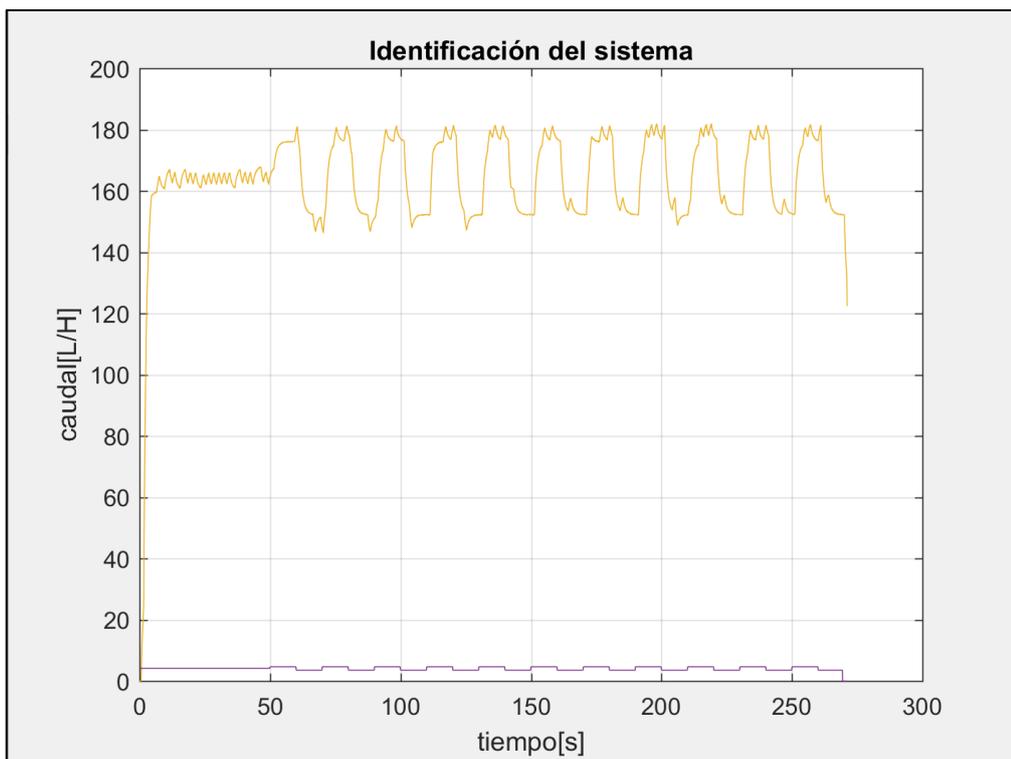


**Figura 3.6 Configuración del bloque Pulse Generator**

Una vez que se ha configurado los dos bloques se procede a realizar la simulación experimental de la planta con el objetivo de observar la respuesta del sistema para el experimento de la identificación de la función de transferencia, el código utilizado para observar el comportamiento del voltaje respecto al caudal fue el siguiente:

```
out=Caudal(:,1);
in=Voltaje_bomba(:,1);
t=Datos(:,1);
plot(t,out)
hold on
plot(t,in)
grid on
xlabel('tiempo[s]')
ylabel('caudal[L/H]')
title('Identificación del sistema')
```

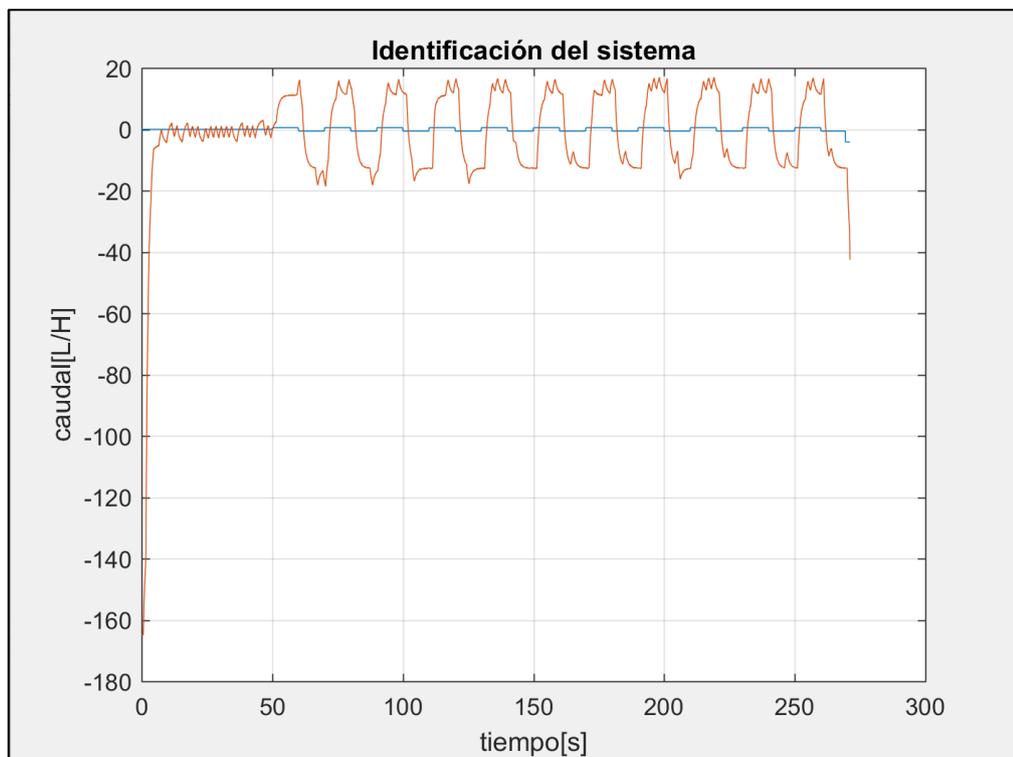
En la figura 3.7 se puede observar el comportamiento del sistema, en el cual la gráfica de color púrpura corresponde al voltaje de entrada hacia el sistema, y el color amarillo corresponde a las lecturas del sensor de caudal en L/H. Se puede observar mediante un análisis que existe simetría en las variaciones, tanto en la entrada como en la salida.



**Figura 3.7 Entrada vs Salida del Sistema en lazo abierto con el punto de operación**

En la figura 3.8 se puede observar el comportamiento del sistema una vez que se retiran los puntos de operación de ambas señales en donde se verifica claramente que la variación es simétrica tanto en la salida como en la entrada; para finalmente proceder a identificar la planta mediante el comando “ident”; el código empleado para restar los puntos de operación y obtener la nueva gráfica mostrada en la figura 3.8 fue el siguiente:

```
in2=in-4.12;  
out2=out-mean(out(40/0.01:50/0.01));  
figure()  
plot(t,[in2 out2])  
grid on  
xlabel('tiempo[s]')  
ylabel('caudal[L/H]')  
title('Identificación del sistema')
```



**Figura 3.8 Entrada vs Salida del sistema en lazo abierto sin el punto de operación**

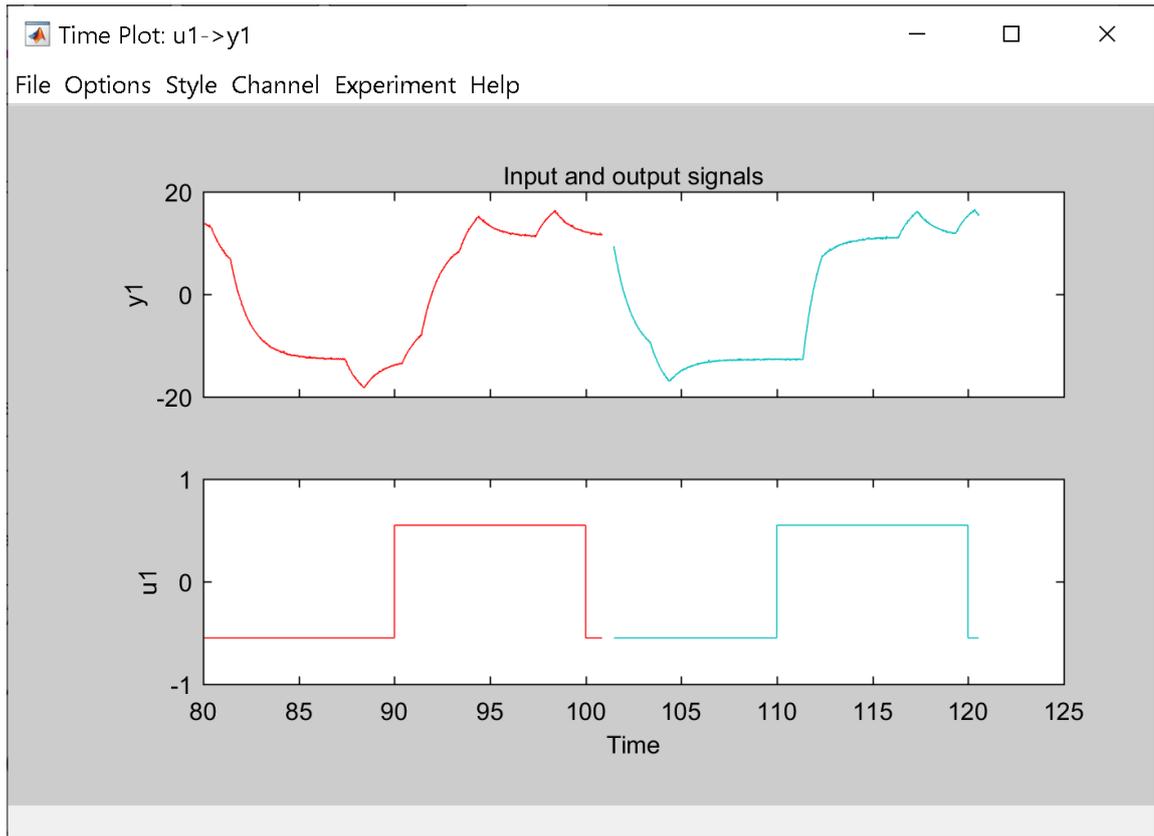
Una vez que se obtienen los datos cuando se retira el punto de operación se ejecuta el comando “ident” y se procede a realizar la importación en el dominio del tiempo de las variables de entrada (in2) y salida (out2) del sistema experimental, colocando un tiempo de inicio de 0 y un tiempo de muestreo de 0.01 con el cual se trabaja como se observa en la figura 3.9.

The screenshot shows a dialog box titled "Import Data" with the following fields and controls:

- Data Format for Signals:** A dropdown menu currently showing "Time-Domain Signals".
- Workspace Variable:** Two input fields. "Input:" contains "in2" and "Output:" contains "out2".
- Data Information:** Three input fields. "Data name:" contains "mydata", "Starting time:" contains "0", and "Sample time:" contains "0.01". A "More" button is located below these fields.
- Buttons:** Four buttons are arranged in a 2x2 grid at the bottom: "Import", "Reset", "Close", and "Help".

**Figura 3.9 Importación de datos**

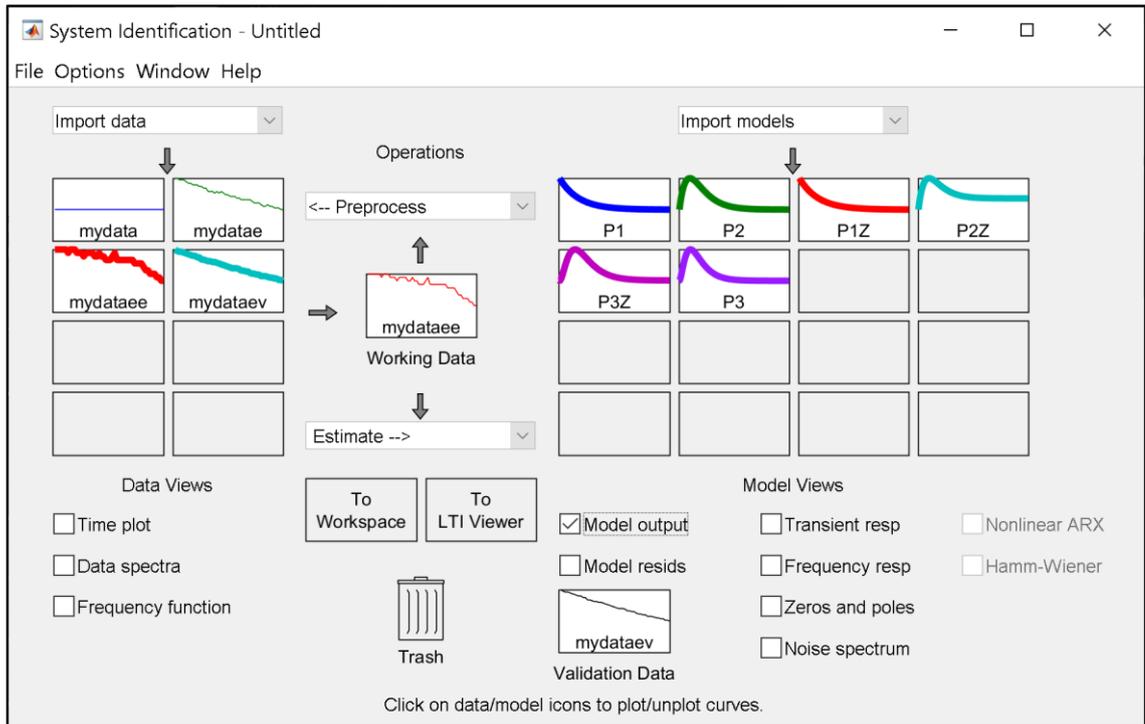
Una vez realizada la importación de las variables de entrada y salida se obtiene una gráfica similar a la figura 3.8 en la cual seleccionamos un rango de datos para proceder a realizar la validación y estimación; cabe recalcar que se deben escoger al menos dos grupos de datos distintos siendo estos de un periodo completo de la señal. Estos datos deben ser lo más similares posibles entre sí, tal y como se muestra en la figura 3.11 en donde la parte de color rojo es el primer rango y la parte de color celeste el segundo rango escogido, pudiendo observar la similitud entre ambos.



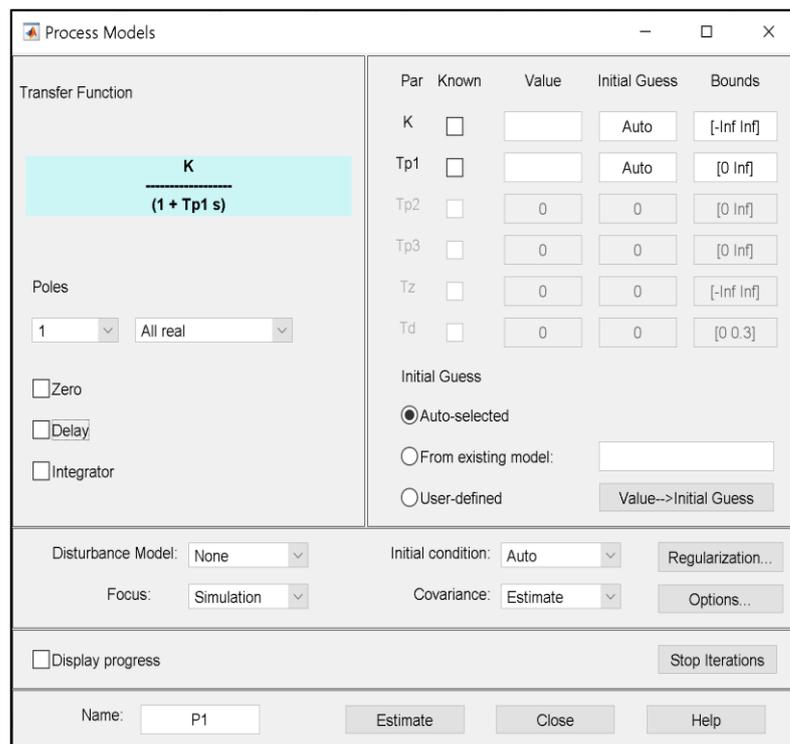
**Figura 3.10 Rangos seleccionados**

Seleccionando los dos rangos similares se procede a realizar la estimación del modelo de proceso en donde la gráfica de color rojo se la arrastra a “working data” que son los datos con los que el programa estima la función de transferencia del sistema y la gráfica de color celeste se la arrastra a “validation data” que son los datos contra los que se compara la respuesta de la función de transferencia encontrada tal como se muestra en la figura 3.11. El siguiente paso fue estimar el modelo; es decir que el “system identification” encuentre una función de transferencia cuya respuesta a la entrada dada sea similar a la respuesta dada en donde aparecerá la ventana de la figura 3.12 donde se colocan las características del modelo a estimar, como el orden de la función de transferencia, presencia de ceros, delay o integrador; o si existen polos complejos conjugados en el caso de orden 2 o superior; se realizó la estimación usando 1 polo, 2 polos, 3 polos, 1 polo y un cero, 2 polos y un cero y finalmente 3 polos y un cero; obteniendo diferentes modelos de función de transferencia los cuales eran válidos todos ya que para que una identificación sea considerada válida el porcentaje debe ser mayor o igual al 70%. En este caso se escogió el modelo P2 que presenta un 83.21% tal como

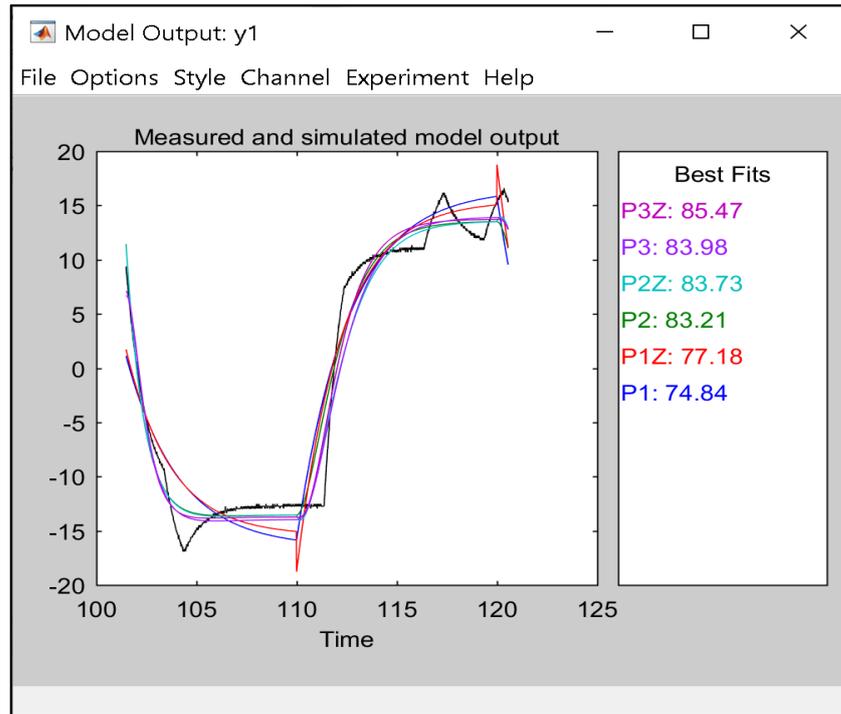
se muestra en la figura 3.13, es decir 2 polos ya que los mismos eran polos cercanos y por lo general un motor de una bomba trabaja mejor con dos polos.



**Figura 3.11 Proceso para la identificación de la planta**



**Figura 3.12 Modelo de proceso**



**Figura 3.13 Función de transferencia obtenida con datos reales**

En la figura 3.14 observamos la función de transferencia obtenida una vez realizada la estimación, en donde podemos observar que tiene dos polos cercanos con los cuales se trabaja mejor al momento de diseñar el controlador de la planta.

```
>> zpk(P2)

ans =

    From input "u1" to output "y1":
      19.845
    -----
    (s+0.8965) (s+0.899)

Name: P2
Continuous-time zero/pole/gain model.
```

**Figura 3.14 Función de transferencia obtenida**

### 3.3 Diseño del controlador de la planta

Para proceder a realizar el diseño del controlador se utiliza la herramienta Sisotool, la cual permite obtener el lugar geométrico de las raíces de manera dinámica. De la función de transferencia encontrada se halla el controlador de la planta usando en Matlab el comando Sisotool(C).

El controlador debe cumplir ciertos requisitos para que la planta funcione de manera correcta; los cuales se los configuró colocando un  $t_{ss}=8$  segundos debido a que debe ser más rápido con respecto al lazo abierto, un overshoot del 10% y finalmente el error de estado estacionario  $E_{ss}=0$ , donde se espera que el controlador cumpla esto en el mapa de polos y ceros de la figura 3.15 y finalmente se lo verifica con la respuesta en estado estable en la figura 3.16.

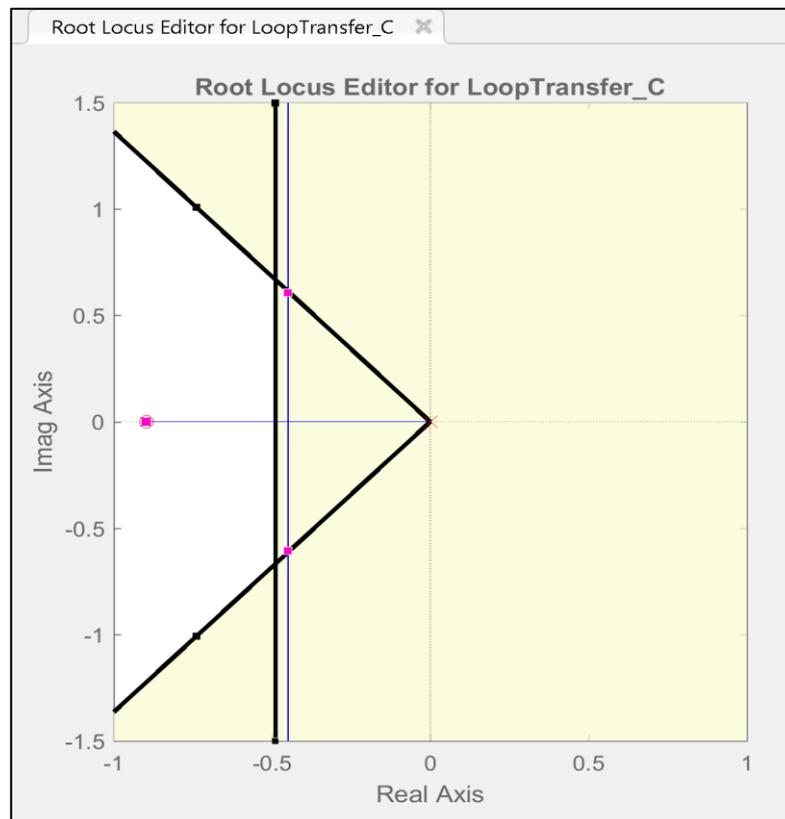
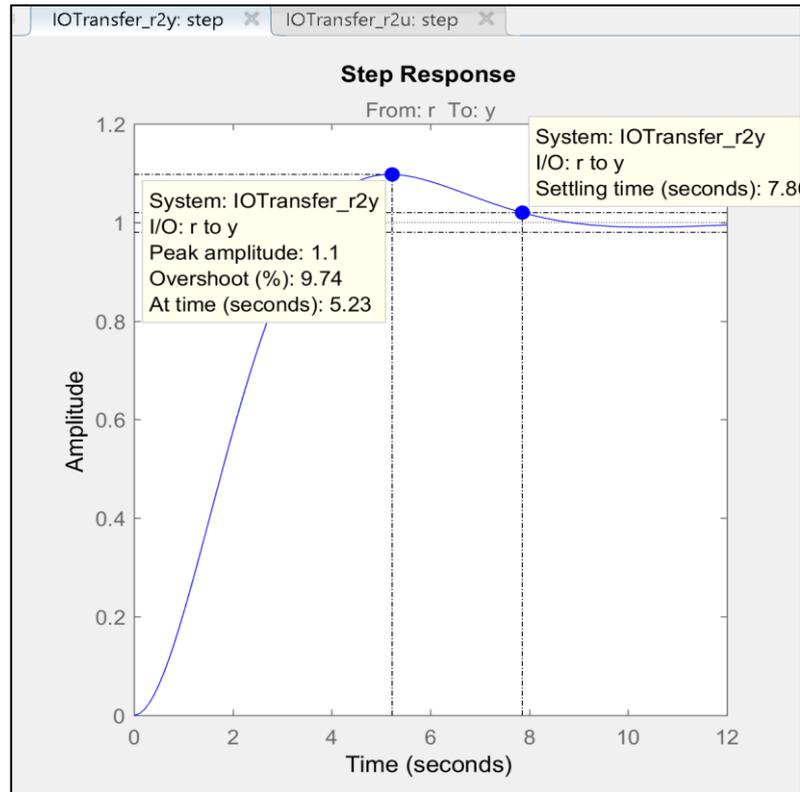


Figura 3.15 diseño del controlador Root Locus



**Figura 3.16 Diseño del controlador step response**

Finalmente se obtiene el controlador con los valores de  $k_p=0.02872$  y  $k_i=0.02575$  tal y como se muestra en la siguiente figura 3.17.

```
>> tf(C)

ans =

    0.02872 s + 0.02575
    -----
           s

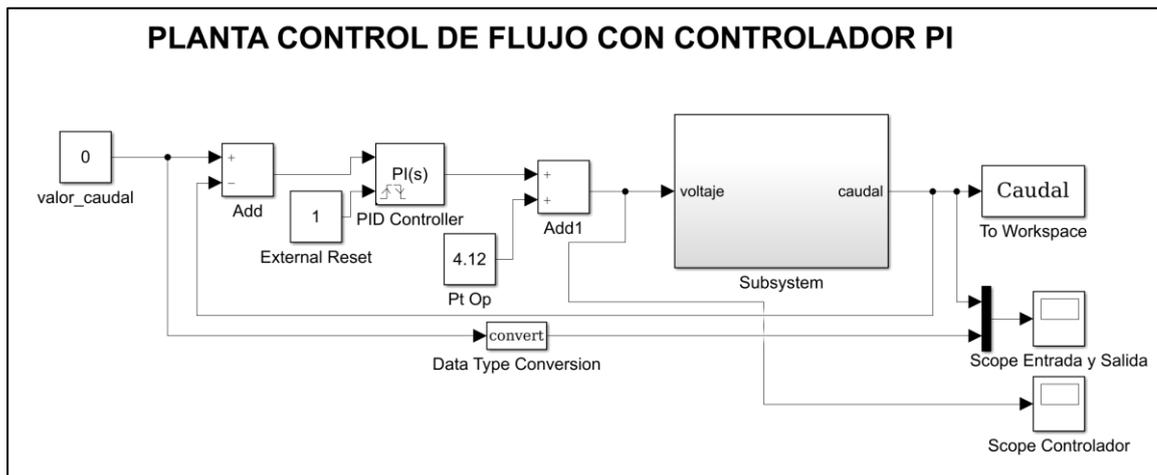
Name: C
Continuous-time transfer function.
```

**Figura 3.17 función de transferencia del controlador**

### 3.4 Prueba del controlador de la planta en lazo cerrado

Para comprobar que el diseño del controlador y la planta funcionen de manera correcta se realiza un diagrama de bloques como en la figura 3.18 donde se verifica la efectividad del diseño del controlador en donde la entrada es un valor

de caudal que no puede exceder los 160 L/H debido a que el diseño del controlador esta para trabajar alrededor del punto de operación, luego de esto se resta con la retroalimentación proveniente del sensor de caudal y pasa por el controlador que actúa sobre la bomba en donde se configura este bloque “PID controller” ingresando los respectivos valores encontrados de  $k_p$  y  $k_i$ . Se uso un external reset para que el valor guardado se haga 0 y el proceso sea directo y finalmente en el bloque subsystem se encuentra toda la programación realizada en el lazo abierto.



**Figura 3.18 Diagrama de bloques del controlador**

El siguiente código fue utilizado para realizar la gráfica de las señales que intervienen en el lazo cerrado:

```

out=Datos_planta(:,2);
in=Datos_planta(:,3);
t=Datos_planta(:,1);
cont=Controlador(:,2);
subplot(3,1,1);
plot(t,in)
grid on
title('Señal de entrada')
xlabel('tiempo[s]')
ylabel('caudal[L/H]')
subplot(3,1,2);
plot(t,out,'r')
grid on
title('Señal de salida')
xlabel('tiempo[s]')
ylabel('caudal[L/H]')

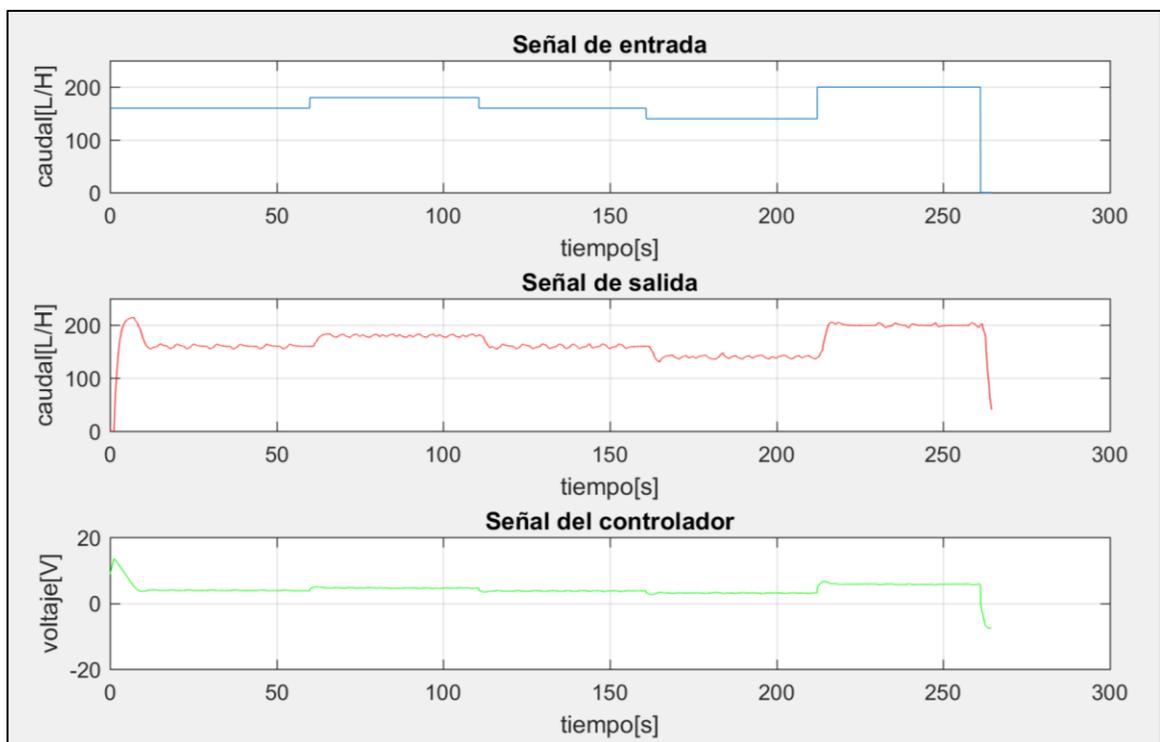
```

```

subplot(3,1,3);
plot(t,cont,'g')
grid on
title('Señal del controlador')
xlabel('tiempo[s]')
ylabel('voltaje[V]')

```

La planta se la hizo trabajar entre valores alrededor de 160 L/H y se pudo llegar a observar que controla la planta desde 120 L/H hasta 220 L/H que sería el máximo que la planta puede operar; además se tomaron dos puntos con el objetivo de observar cuanto era el overshoot y se llegó a obtener un valor de 15% el cual es cercano al valor obtenido en lazo abierto. Como se observa el  $t_{ss}=10$  segundos con un  $ess=0$  entonces la planta se encontraba operando en óptimas condiciones.



**Figura 3.19 Prueba del controlador**

### 3.5 Análisis de Costos

**Tabla 3.2 Detalle de costos de materiales electrónicos**

LISTADO DE MATERIALES ELECTRÓNICOS				
ITEM	CANT	PRECIO UNID	SUBTOTAL	DESCRIPCIÓN
Rotámetro 25-250 L/H	1	\$ 13,64	\$ 13,64	indicador de flujo de agua
Bomba de agua sumergible 12V	1	\$ 53,30	\$ 53,30	bompear agua desde el reservorio
Puente H L298N	1	\$ 3,90	\$ 3,90	doble puente H
Arduino Mega 2560	1	\$ 13,50	\$ 13,50	tarjeta de adquisición
Cable USB tipo A - tipo B	1	\$ 1,50	\$ 1,50	cable USB
Arduino Uno R3	1	\$ 11,00	\$ 11,00	tarjeta de adquisición
Cable USB tipo A - tipo B	1	\$ 1,50	\$ 1,50	cable USB
Servomotor MG995	1	\$ 8,50	\$ 8,50	girar válvula
Sensor YF-S201	1	\$ 6,00	\$ 6,00	medir el flujo de agua
Fuente conmutada 12V - 10A	1	\$ 20,00	\$ 20,00	fuentes de alimentación
Cable fuente	1	\$ 2,20	\$ 2,20	conectar a tomacorriente
Resistencia 100K	1	\$ 0,10	\$ 0,10	componente electrónico
Capacitor polarizado 10uF	1	\$ 0,10	\$ 0,10	componente electrónico
Capacitor 10nF	3	\$ 0,10	\$ 0,30	componente electrónico
Capacitor 100nF	4	\$ 0,10	\$ 0,40	componente electrónico
LM7805	1	\$ 0,30	\$ 0,30	componente electrónico
Disipador de calor	1	\$ 0,20	\$ 0,20	componente electrónico
Bornera doble	2	\$ 0,20	\$ 0,40	conectar cables
Juego de jumpers M-M	1	\$ 3,00	\$ 3,00	realizar conexiones
<b>TOTAL</b>			<b>\$ 139,84</b>	

**Tabla 3.3 Detalle de costos de materiales de construcción**

LISTADO DE MATERIALES DE CONSTRUCCIÓN				
ITEM	CANT	PRECIO UNID	SUBTOTAL	DESCRIPCIÓN
Metro de manguera 3/8"	1	\$ 4,39	\$ 4,39	manguera conectada a la bomba
Metro de manguera 3/8"	1	\$ 4,39	\$ 4,39	manguera de retorno al tanque
Cinta teflón 1/2" x 10m	1	\$ 0,78	\$ 0,78	evitar fugas de agua
Mini abrazadera de bronce 1/2"	2	\$ 0,45	\$ 0,90	asegurar mangueras
Neplo RM PP 1/2" x 6cm	2	\$ 0,22	\$ 0,44	unir componentes
Neplo RM PP 1/2" x 10cm	1	\$ 0,34	\$ 0,34	unir componentes
Codo PP RR H 1/2" x 90°	2	\$ 0,38	\$ 0,76	unir componentes
Reductor buje RR PP 3/4 x 1/2"	3	\$ 0,25	\$ 0,75	unir componentes

Unión RH PP 1/2"	2	\$ 0,48	\$ 0,96	unir componentes
Unión RH PP 3/4"	1	\$ 0,62	\$ 0,62	unir componentes
Adaptador 3/8 x 1/2 hose barb	2	\$ 4,05	\$ 8,10	adaptador para mangueras
Amarras plásticas	1	\$ 3,50	\$ 3,50	sujetar componentes
Válvula de compuerta de 1/2"	1	\$ 5,69	\$ 5,69	perturbar flujo de agua
Abrazadera metálica de tubo 1/2"	3	\$ 0,75	\$ 2,25	asegurar la planta a la base
Tornillo de madera 3x20mm	3	\$ 0,15	\$ 0,45	asegurar abrazadera metálica
Base de madera 60x32x13	1	\$ 3,85	\$ 3,85	sostener planta
Caja de madera 34x24x10	1	\$ 15,00	\$ 15,00	almacenar comp. Electrónicos
<b>TOTAL</b>			<b>\$ 48,78</b>	

**Tabla 3.4 Detalle de costo total de fabricación**

<b>COSTO TOTAL DE FABRICACIÓN</b>	
Materiales electrónicos	\$139,84
Materiales de construcción	\$ 48,78
<b>TOTAL</b>	<b>\$188,62</b>

# CAPÍTULO 4

## 4. CONCLUSIONES Y RECOMENDACIONES

### 4.1 Conclusiones

- El uso del software de programación matemático y gráfico permitió identificar y diseñar un controlador que actuará dentro del punto de operación de la planta con el objetivo de que esta funcione de manera correcta dentro de su rango de operación.
- La planta didáctica permite que el estudiante pueda desarrollar sus habilidades a partir de los conocimientos adquiridos en el aula de clase ya que el estudiante será capaz de identificar y diseñar un controlador en función a los requerimientos que desee, con el objetivo de que la planta de control de caudal actúe de manera correcta.
- La construcción de la planta didáctica ofrece un bajo costo para la universidad debido a que se usó mano de obra local y así evitar comprar plantas de altos costos que funcionan de igual manera que la desarrollada.
- Al aplicar ingeniería inversa en plantas ya existentes en la Escuela Superior Politécnica del Litoral permitió conocer su funcionamiento; el mismo que nos fue de gran ayuda para poder implementarlo en la planta didáctica de control de flujo.

### 4.2 Recomendaciones

- Se debe tener en cuenta que al momento de trabajar con tarjetas de desarrollo; verificar cuanto voltaje utiliza cada componente para que no exista problemas de funcionamiento en el mismo y estas sean consideradas para la adquisición de las fuentes de voltaje.
- Al momento de escoger la mejor función de transferencia se debe tener en cuenta que el valor de porcentaje para el mismo debe ser mayor o igual al 70%.

# BIBLIOGRAFÍA

Arboleda, J. E., & Guamán, A. M. (2019). *Diseño e Implementación de una planta didáctica de control de flujo para el Laboratorio de Control Automático - MACI*. Guayaquil: Escuela Superior Politécnica del Litoral.

*Arduino.cl*. (s.f.). Recuperado el 9 de Julio de 2020, de <https://arduino.cl/arduino-mega-2560/>

*Arduino.cl*. (s.f.). Recuperado el 9 de julio de 2020, de <https://arduino.cl/arduino-uno/>

*Bomba Sumergible*. (2016). Obtenido de [https://www.bombasumergible.com/definicion\\_bomba\\_sumergible.html](https://www.bombasumergible.com/definicion_bomba_sumergible.html)

García, P. (9 de Mayo de 2013). *Franklin Electric*. Obtenido de <https://franklinlinkmx.wordpress.com/2013/09/05/que-es-el-control-pid/>

*Grupo AlmagroMur*. (22 de noviembre de 2017). Obtenido de <https://www.grupoalmagromur.es/las-valvulas-compuerta-elegirlas-correctamente/>

Hareendran, T. K. (s.f.). *Electro Schematics*. Recuperado el agosto de 2020, de <https://www.electroschematics.com/working-with-water-flow-sensors-arduino/>

Hernández, G. M., & Gómez, J. A. (Septiembre de 1982). *Facultad Nacional de Minas*. Obtenido de Laboratorio de Control Automático: [http://bdigital.unal.edu.co/1258/6/182\\_-\\_5\\_Capi\\_5.pdf](http://bdigital.unal.edu.co/1258/6/182_-_5_Capi_5.pdf)

*Naylamp Mechatronics*. (s.f.). Recuperado el julio de 2020, de <https://naylampmechatronics.com/sensores-liquido/108-sensor-de-flujo-de-agua-12-yf-s201.html>

*Omega*. (s.f.). Recuperado el julio de 2020, de <https://es.omega.com/prodinfo/caudalimetros-area-variable.html>

*Revista Electro Industria*. (Agosto de 2018). Obtenido de <http://www.emb.cl/electroindustria/articulo.mvc?xid=3286&ni=que-son-los-sistemas-daq>

*Robots Didácticos.* (s.f.). Recuperado el agosto de 2020, de <http://robots-argentina.com.ar/didactica/uso-de-la-placa-l298n-para-motores-de-cc/>

*The Math Works Inc.* (s.f.). Recuperado el julio de 2020, de <https://www.mathworks.com/help/supportpkg/arduino/examples/servo-control.html>

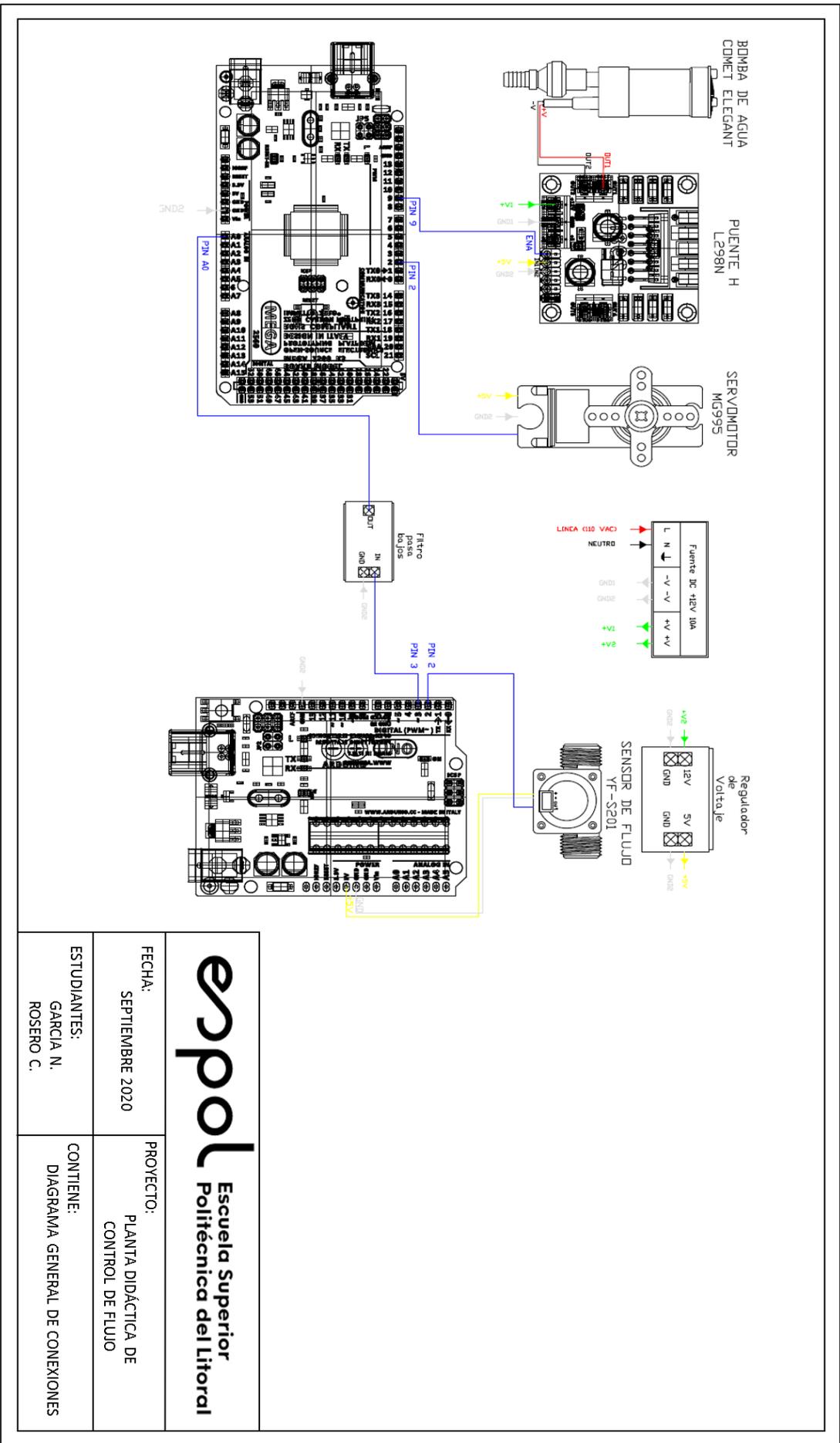
*The Math Works Inc.* (Marzo de 2020). Obtenido de [https://ww2.mathworks.cn/help/pdf\\_doc/supportpkg/arduino/arduino\\_ug.pdf](https://ww2.mathworks.cn/help/pdf_doc/supportpkg/arduino/arduino_ug.pdf)

# 5.ANEXOS

## Anexo 1

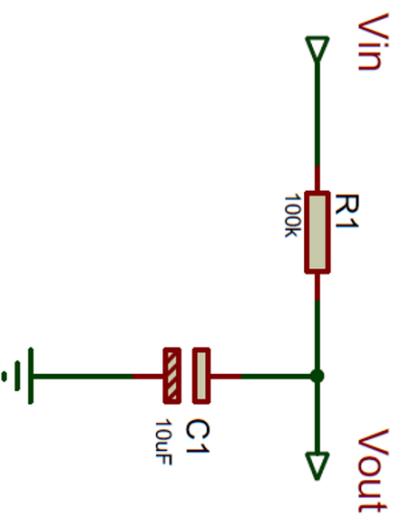
### PLANOS CONSTRUCTIVOS DE LA PLANTA DE CONTROL DE FLUJO

- Planos eléctricos
- Planos electrónicos
- Planos mecánicos
- P&ID

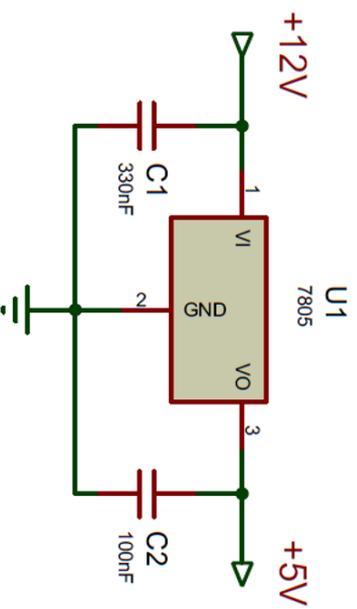


<b>espol</b> Escuela Superior Politécnica del Litoral	
FECHA: SEPTIEMBRE 2020	PROYECTO: PLANTA DIDÁCTICA DE CONTROL DE FLUJO
ESTUDIANTES: GARCIA N. ROSERO C.	CONTIENE: DIAGRAMA GENERAL DE CONEXIONES

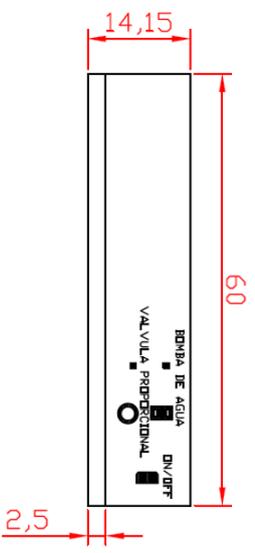
## Filtro pasa bajos



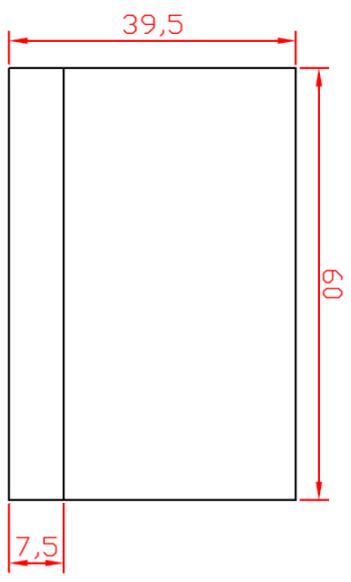
## Regulador de voltaje



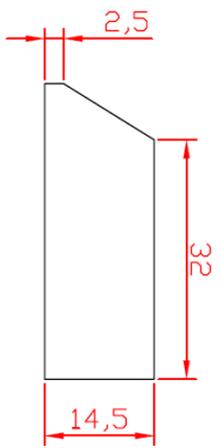
 <b>Escuela Superior Politécnica del Litoral</b>	
FECHA: SEPTIEMBRE 2020	PROYECTO: PLANTA DIDÁCTICA DE CONTROL DE FLUJO
ESTUDIANTES: GARCIA N. ROSERO C.	CONTIENE: DIAGRAMA CIRCUITO FILTRO PASA BAJOS Y REGULADOR DE VOLTAJE



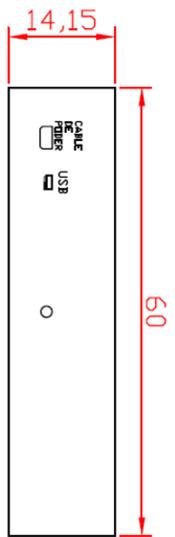
VISTA FRONTAL



VISTA SUPERIOR

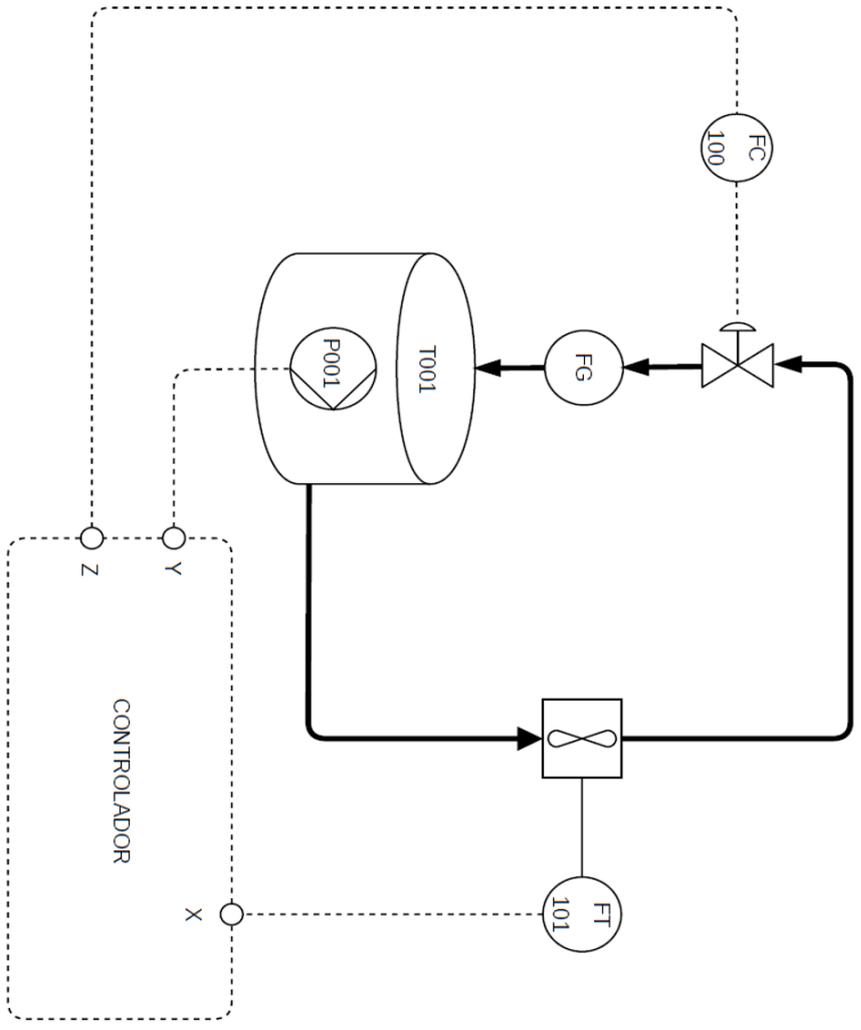


VISTA LATERAL



VISTA POSTERIOR

 <b>Escuela Superior Politécnica del Litoral</b>	
FECHA: SEPTIEMBRE 2020	PROYECTO: PLANTA DIDÁCTICA DE CONTROL DE FLUJO
ESTUDIANTES: GARCIA N. ROSERO C.	CONTIENE: PLANO ESTRUCTURA METÁLICA



**espol** Escuela Superior Politécnica del Litoral

FECHA: SEPTIEMBRE 2020	PROYECTO: PLANTA DIDÁCTICA DE CONTROL DE FLUJO
ESTUDIANTES: GARCIA N. ROSERO C.	CONTIENE: DIAGRAMA P&ID

## Anexo 2

### PRÁCTICA: IDENTIFICACIÓN DEL SISTEMA

#### Objetivos

##### Objetivo General

Al finalizar esta sesión el estudiante estará en capacidad de diseñar un experimento básico para la identificación de sistema de una planta de control de caudal utilizando MATLAB.

##### Objetivos Específicos

Interpretar curva característica y el comportamiento del sistema y estimar puntos de operación a partir de la misma.

Estimar parámetros a usar en la simulación del modelo en Simulink a través de la interpretación de la curva característica y comportamiento del sistema.

Utilizar parámetros calculados y el modelo en Simulink para la ejecución del experimento.

##### Curva característica

Es una gráfica que mapea los posibles valores en estado estable que toman la entrada y salida de un sistema. El eje X suele representar los valores que toma la entrada, mientras que el eje Y suele representar los valores que toma la salida.

Esta gráfica es útil para la selección del punto de operación ya que ofrece información sobre el comportamiento estático del sistema. Esta selección depende de las condiciones deseadas de operación y se debe, en lo posible, seleccionar un punto de operación donde la curva característica de dicho sistema sea aproximadamente lineal. La selección del punto de operación es crítica para la identificación del sistema y por lo tanto para el posterior diseño del controlador.

Para el diseño del experimento de identificación es necesario conocer el punto de operación del sistema ( $In_{op}, Out_{op}$ ), las variaciones que se realizarán con respecto al mismo ( $\Delta In$ ) y el tiempo que le toma al sistema en llegar al punto de operación (tiempo de arranque). Se trata de que la variación sea simétrica tanto en la entrada como en la salida del sistema; es decir que si se realiza un cambio  $\Delta In$  igual a X en la entrada y se

obtiene a la salida un  $\Delta Out$  igual a  $Y$ , al realizar un cambio  $\Delta In$  igual a  $-X$  en la entrada se esperaría obtener un  $\Delta Out$  igual a  $-Y$ .

### **Procedimiento para la identificación de la planta de control de caudal.**

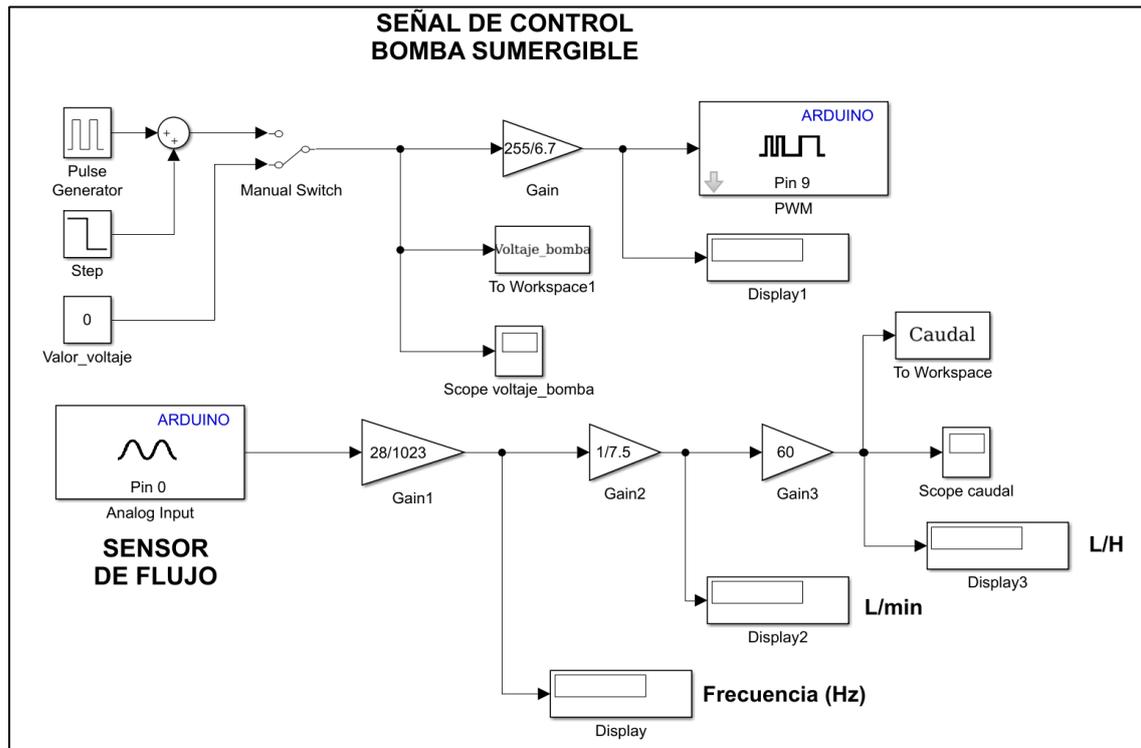
- Diseño del experimento. - Se especifican los parámetros a utilizar para la identificación, tales como: tiempo de muestreo, tiempo de simulación, período del tren de pulsos, etc.
- Adquisición de datos. - Se obtienen datos de la planta para su posterior procesamiento.
- Acondicionamiento de señales. - Se diseña y aplica el filtro, y luego se determinan y restan los puntos de operación a la entrada y salida del sistema.
- Selección de grupos de datos para identificación y validación.
- Identificación hasta obtener un porcentaje de ajuste aceptable.

### **Ejemplo**

Como ejemplo se realizará la identificación de la planta de control de caudal, el mismo que deberán hacerlo con el punto de operación que ustedes escojan o a la vez que se los proporcione el profesor de la materia.

### **Diseño del experimento**

Al realizar el diagrama de bloques se debe de hacer algo similar a la figura 5.1 con el objetivo de realizar mi experimento de manera correcta y de esta manera proceder a colocar los parámetros a utilizar en la identificación.



**Figura 5.1 Diagrama de bloques**

Para realizar la adquisición de datos de manera correcta, es necesario configurar ciertos parámetros como:

1. Amplitud del tren de pulsos: Se la configura como dos veces el valor de la variación de la entrada ( $\Delta I_n$ ).
2. Período del tren de pulsos: Se lo configura con un valor mayor o igual a dos veces el tiempo de estabilización del sistema ( $T_{ss}$ ). La mitad de este parámetro indica cada cuánto tiempo se produce una variación al punto de operación del sistema; este parámetro debe ser múltiplo del tiempo de muestreo.
3. Ancho del tren de pulsos: También conocido como duty cycle; se configura en 50%.
4. Retraso del tren de pulsos: Se lo configura con el tiempo de arranque del sistema ( $T_{arr}$ ). Recuerde que se requiere energizar al sistema en  $t=0$  y que la salida se estabilice en el punto de operación antes de realizar la variación en la entrada.
5. Tiempo de paso de la señal escalón: Se configura para que sea igual al tiempo de arranque del sistema.
6. Valor inicial de la señal escalón: Se configura para que sea igual al valor de operación de la entrada ( $i_{nop}$ ).

7. Valor final de la señal escalón: Se configura para que sea igual al punto de operación de la entrada menos la variación de la entrada ( $inop - \Delta In$ ). Se realiza una variación negativa ya que el tren de pulsos, que se suma a la señal escalón, toma el valor de 0 el valor de  $2\Delta In$ . Al sumar dichos valores al valor final del bloque Step se genera el efecto deseado a la entrada del sistema.
8. Duración del experimento: El tiempo de simulación o duración del experimento debe ser al menos  $4PTP + Tarr$ .
9. Tiempo de muestreo: Se selecciona de manera que se cumpla el teorema de muestreo de Nyquist; dependerá del ancho de banda del sistema en estudio y su estimación se detalla más adelante. Este parámetro indica cada cuánto tiempo se almacena una muestra del sistema.

Para configurar los 8 parámetros desconocidos del sistema, se requiere conocer o estimar los siguientes parámetros:

**Punto de operación:** Niveles o estados donde normalmente opera el sistema y el mismo se encuentra estable. Se selecciona en base a las especificaciones del operador y de la curva característica del sistema.

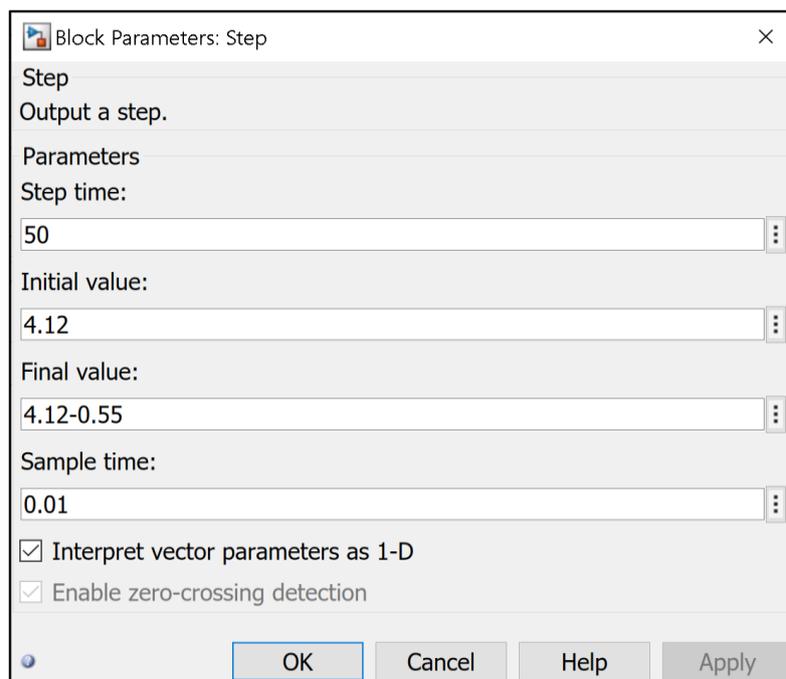
**Variación o incremental de entrada:** Indica cuánto se desvía o varía la señal de entrada con respecto al punto de operación. Para la identificación, estas variaciones deben ser tanto positivas como negativas y de igual amplitud; se selecciona en base a las especificaciones del operador y de la curva característica del sistema.

**Tiempo de arranque del sistema:** Es el tiempo requerido para que la salida se estabilice en su punto de operación al energizar el sistema.

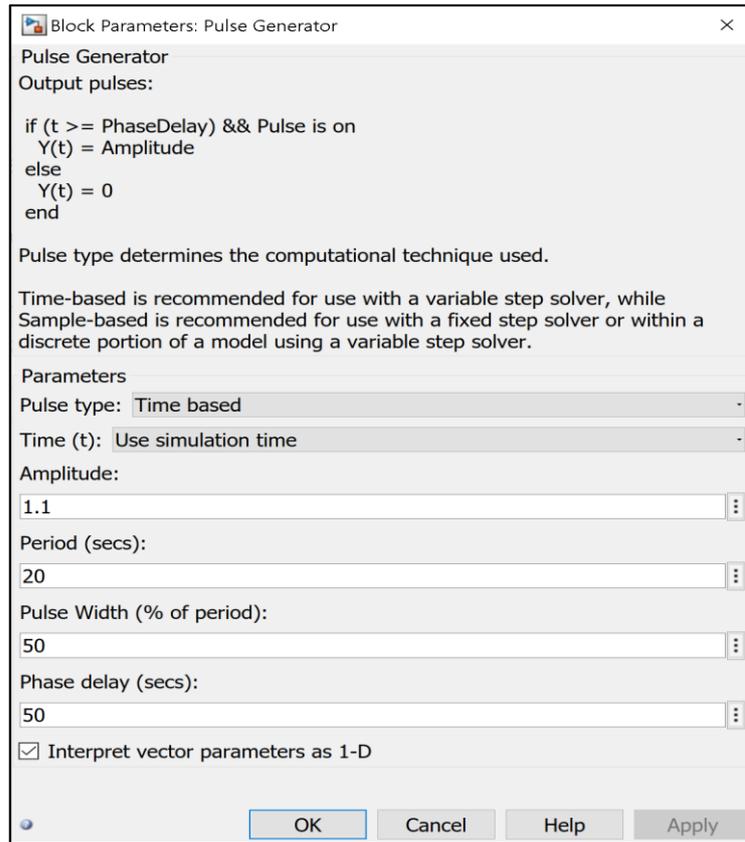
**Tiempo de estabilización:** Es el tiempo requerido para que la salida quede encerrada en la banda del 2% luego de realizar una variación en el punto de operación de la entrada.

**Sobrenivel porcentual y tiempo pico:** Para sistemas cuya dinámica sea similar a la de un sistema de segundo orden subamortiguado, se requiere estimar el coeficiente de amortiguamiento a través del sobrenivel porcentual para la estimación del tiempo de muestreo. De manera alternativa, se puede estimar el tiempo pico en vez del tiempo de estabilización para mejorar la precisión de la estimación del tiempo de muestreo cuando se presenta esta dinámica.

se configura dos bloques los mismos que son de mayor importancia para realizar el experimento. Primero se configura el bloque “step” como se observa en la figura 5.2 se coloca un  $tarr=50$  segundos, el valor inicial es mi punto de operación  $Po=4.12$ ; por lo tanto, se colocó mencionado valor y el valor final es la resta entre el  $Po$  y  $\Delta V$ , también se configura el bloque “pulse generator” como se observa en la figura 5.3 en la que la amplitud equivale a dos veces el  $\Delta V$ , se coloca el periodo de 20 segundos con el objetivo de observar el funcionamiento de la planta y el retardo de fase equivale al tiempo de arranque utilizado que es  $tarr=50$  segundos.



**Figura 5.2 Bloque Step**



**Figura 5.3 Bloque Pulse Generator**

### Adquisición de datos

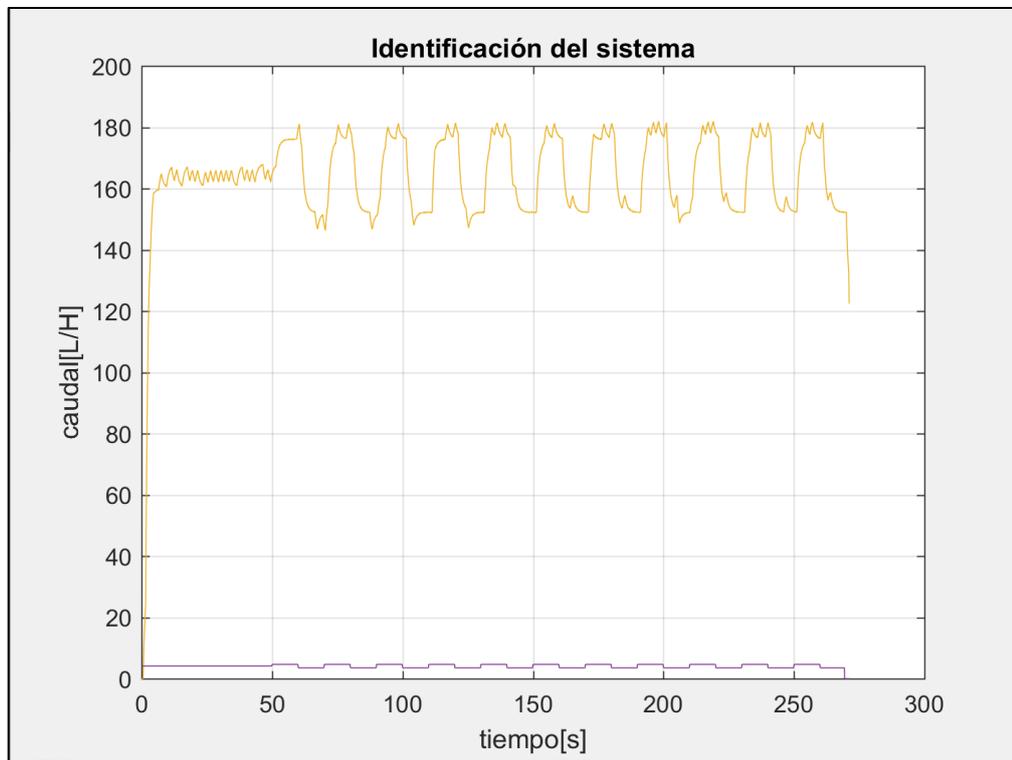
Una vez que se ha configurado los dos bloques se procede a realizar la simulación experimental de la planta con el objetivo de observar la respuesta del sistema para el experimento de la identificación de la función de transferencia, el código utilizado para observar el comportamiento del voltaje respecto al caudal fue el siguiente:

```

out=Caudal(:,1);
in=Voltaje_bomba(:,1);
t=Datos(:,1);
plot(t,out)
hold on
plot(t,in)
grid on
xlabel('tiempo[s]')
ylabel('caudal[L/H]')
title('Identificación del sistema')

```

En la figura 5.4 se puede observar el comportamiento del sistema, en el cual la gráfica de color púrpura corresponde al voltaje de entrada hacia el sistema, y el color amarillo corresponde a las lecturas del sensor de caudal en L/H. Se puede observar mediante un análisis que existe simetría en las variaciones, tanto en la entrada como en la salida.



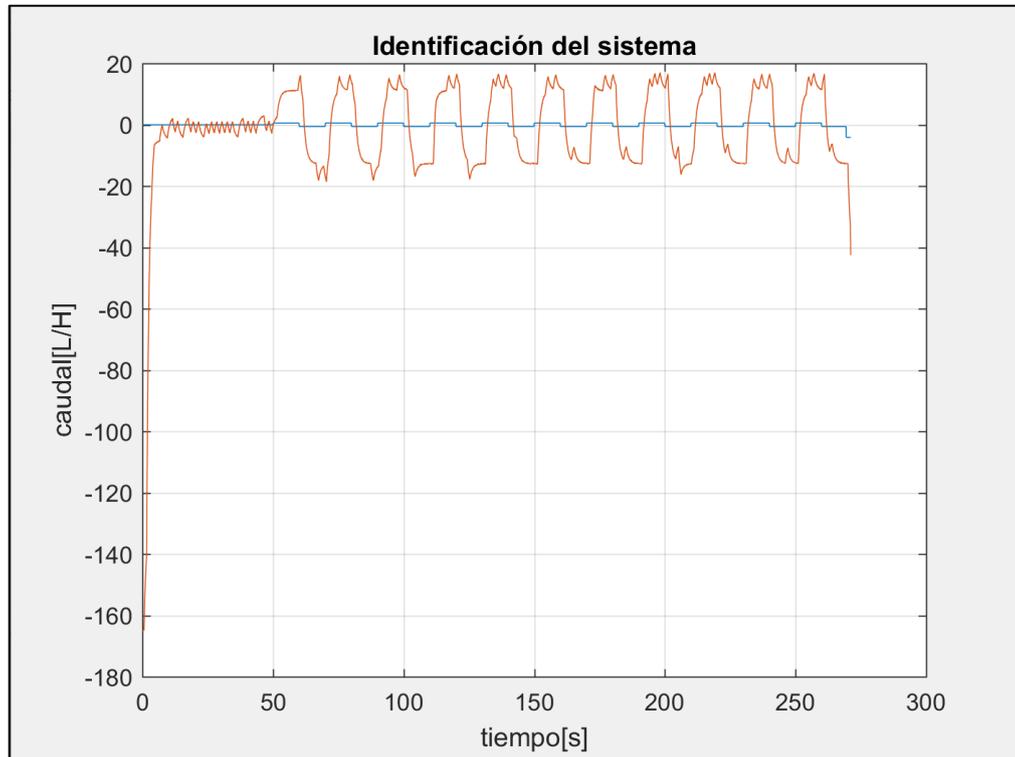
**Figura 5.4 Entrada vs Salida del Sistema en lazo abierto con el punto de operación**

### Acondicionamiento de señales

En la figura 5 se puede observar el comportamiento del sistema una vez que se retiran los puntos de operación de ambas señales en donde se verifica claramente que la variación es simétrica tanto en la salida como en la entrada; para finalmente proceder a identificar la planta mediante el comando "ident"; el código empleado para restar los puntos de operación y obtener la nueva gráfica mostrada en la figura 5.5 fue el siguiente:

```
in2=in-4.12;
out2=out-mean(out(40/0.01:50/0.01));
figure()
plot(t,[in2 out2])
grid on
xlabel('tiempo[s]')
```

```
ylabel('caudal[L/H]')
title('Identificación del sistema')
```



**Figura 5.5 Entrada vs Salida del Sistema en lazo abierto sin el punto de operación**

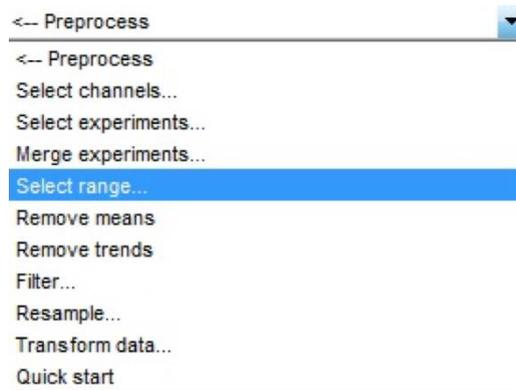
### **Selección de grupos de datos para identificación y validación.**

Se deben seleccionar dos grupos de datos para realizar la identificación.

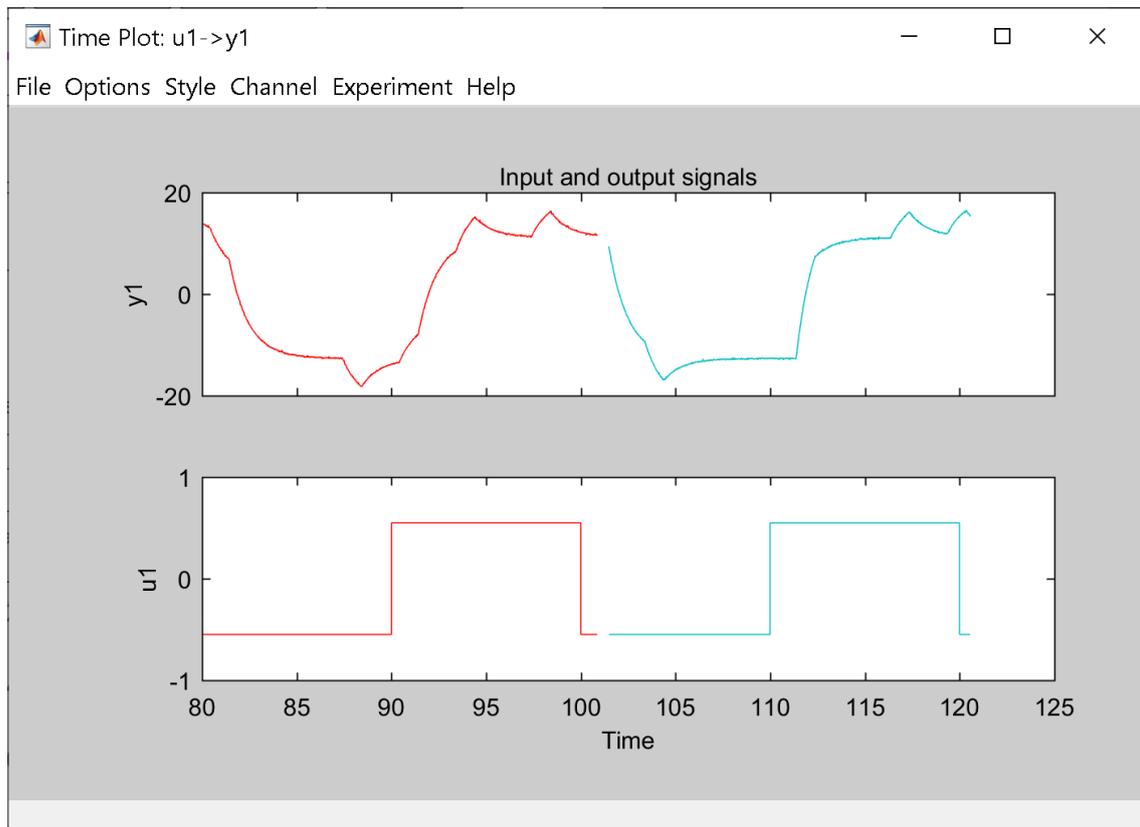
Workingdata: Son los datos con los que el programa estima la función de transferencia del sistema.

Validation data: Son los datos contra los que se compara la respuesta de la función de transferencia encontrada.

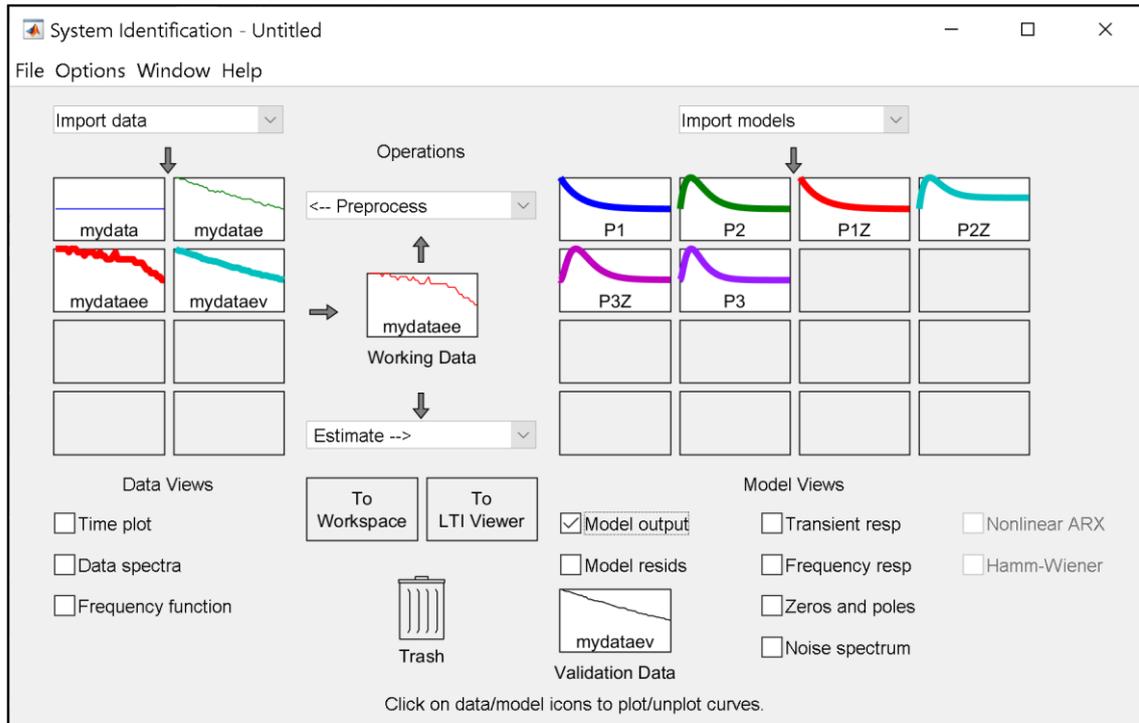
Para seleccionar un rango de datos, dar click en Preprocess y seleccione la opción Select Range, tal como se muestra en la figura 5.6. Se abrirá una ventana como la mostrada en la figura 5.7; debe seleccionar al menos un período completo del tren de pulsos y luego dar click al botón Insert. El grupo de datos seleccionado aparecerá junto a los datos importados en la ventana del System Identification donde se procederá a realizar la estimación como se muestra en la figura 5.8 hasta obtener un porcentaje adecuado.



**Figura 5.6 Selección de rango**



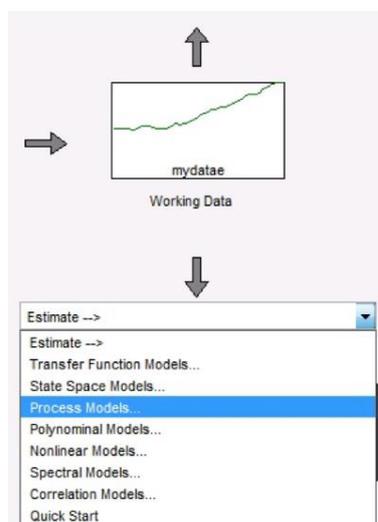
**Figura 5.7 Rangos seleccionados**



**Figura 5.8 Ventana System Identification**

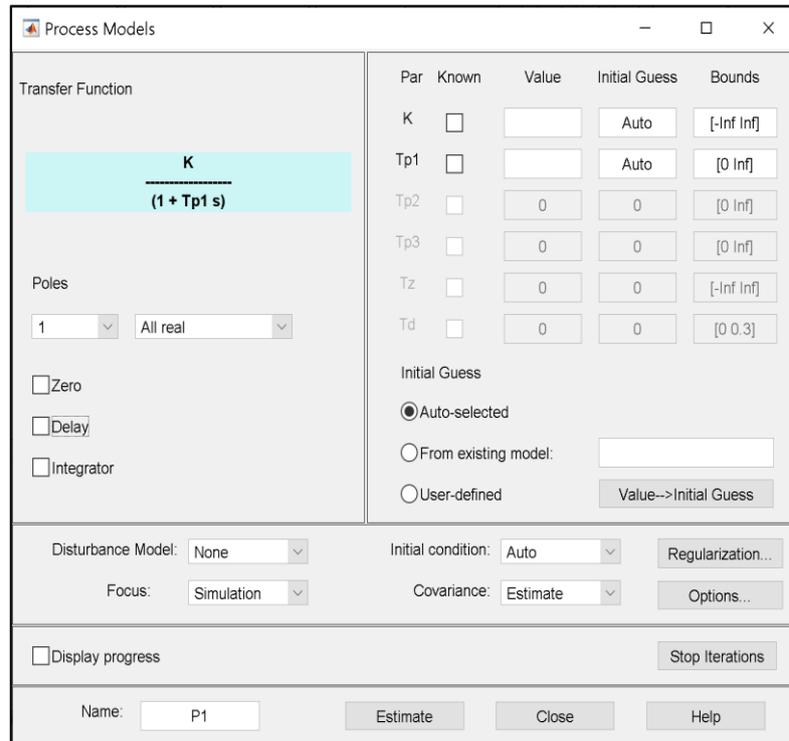
**Identificación hasta obtener un porcentaje de ajuste aceptable.**

El siguiente paso es estimar el modelo; es decir que el System identification encuentre una función de transferencia cuya respuesta a la entrada dada sea similar a la respuesta dada. Dar click en Estimate y luego en Process Models como se muestra en la figura 5.9.



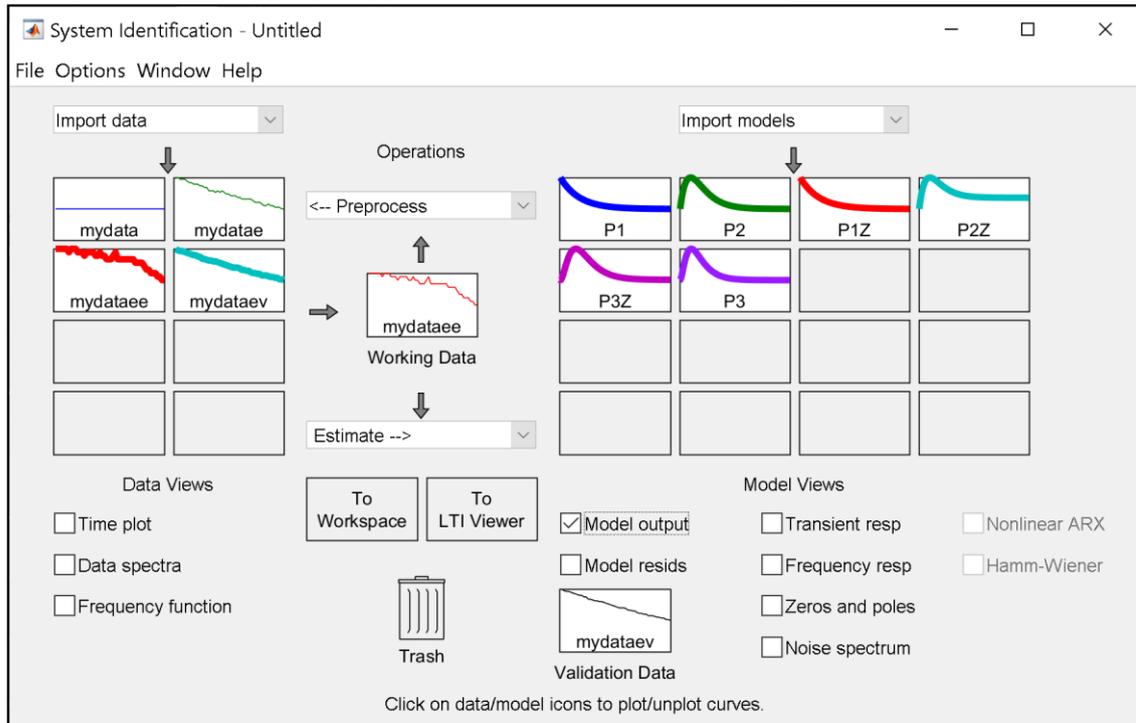
**Figura 5.9 Selección de Process Models**

Aparecerá la ventana mostrada en la figura 5.10; en esta ventana se le puede indicar al System Identification las características del modelo a estimar como el orden de la función de transferencia, presencia de ceros, delay, o integrador, o si existen polos complejos conjugados en el caso de orden 2 o superior. También se puede indicar si se conoce algún parámetro como ganancia estática o algún polo o cero de la función de transferencia resultante.



**Figura 5.10 Process Models**

Los parámetros se escogen de acuerdo con previo conocimiento del modelo matemático del sistema. Por ejemplo, si se conoce que la función de transferencia de un motor de corriente continua tiene un modelo de segundo orden y se requiere realizar la identificación del mismo, se debería escoger el número dos en la opción de Poles y se podrían realizar dos estimaciones, una con polos reales y una con polos complejos conjugados. Luego se debe verificar cuál se parece más al comportamiento real del sistema. Sin previo conocimiento del modelo, puede realizar diferentes combinaciones de los parámetros y mediante prueba y error escoger el modelo cuya respuesta se asemeje más a la respuesta real del sistema. Cuando termine de escoger los parámetros, dé click al botón Estimate. La ventana del System Identification deberá verse como la figura 5.11.

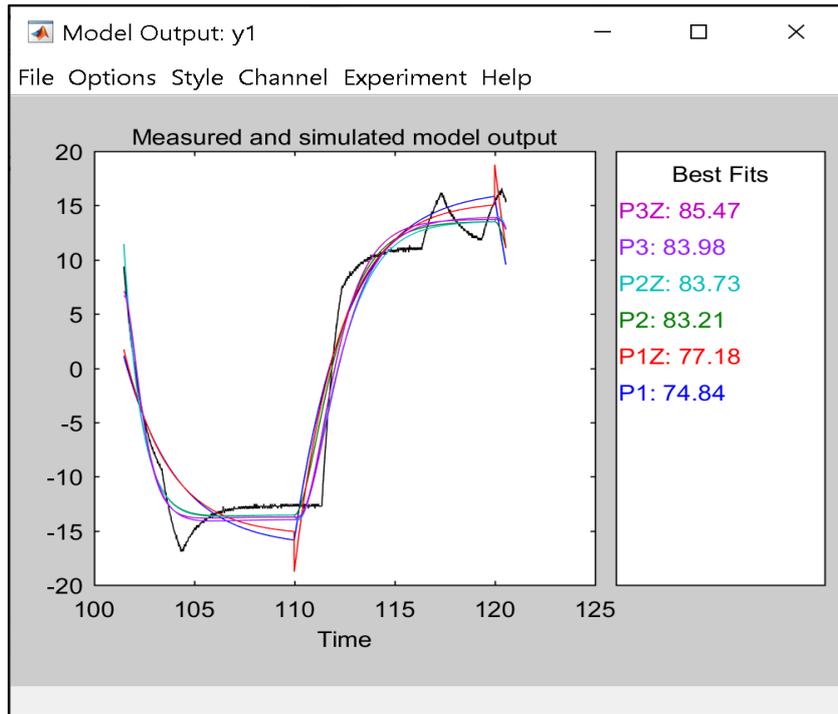


**Figura 5.11 Proceso para la identificación de la planta**

Dé click sobre la casilla de Model Output y se abrirá una ventana similar a la de la figura 5.12, esta ventana indica de manera gráfica y numérica cuánto se parece o se aproxima la respuesta de los modelos estimados a la respuesta real del sistema. Note que cada sistema se representa por un color y nombre diferente. En el lado derecha de la ventana se observa el porcentaje de aproximación de la respuesta de cada modelo con la respuesta real. Recuerde que para considerar la identificación válida este valor debe ser mayor o igual al 80% cuando se trabaja con datos obtenidos de una simulación.

Una vez que decida cuál es el modelo que mejor simula el comportamiento del sistema, dé click sobre el mismo y arrástrelo al recuadro To Workspace. El modelo escogido deberá aparecer ahora en el Workspace.

Escriba el nombre del modelo escogido en el Command Window o de manera alternativa escriba el comando `tf (nombre del modelo)` para visualizar la función de transferencia del modelo que representa el sistema. El resultado para este ejemplo se observa en la figura 5.13.



**Figura 5.12 Ventana Model Output**

```
>> zpk(P2)

ans =

    From input "u1" to output "y1":
    19.845
    -----
    (s+0.8965) (s+0.899)

Name: P2
Continuous-time zero/pole/gain model.
```

**Figura 5.13 Función de transferencia**

## Anexo 3

### PRÁCTICA: DISEÑO DEL CONTROLADOR PID

#### Objetivos

##### Objetivo General

Diseñar un controlador PID o cualquiera de sus combinaciones para la planta de control de flujo.

##### Objetivos Específicos

Realizar el diseño de un controlador PID para que el sistema cumpla ciertos requerimientos usando Sisotool.

Verificar el funcionamiento del controlador diseñado realizando una simulación en Simulink.

##### Procedimiento

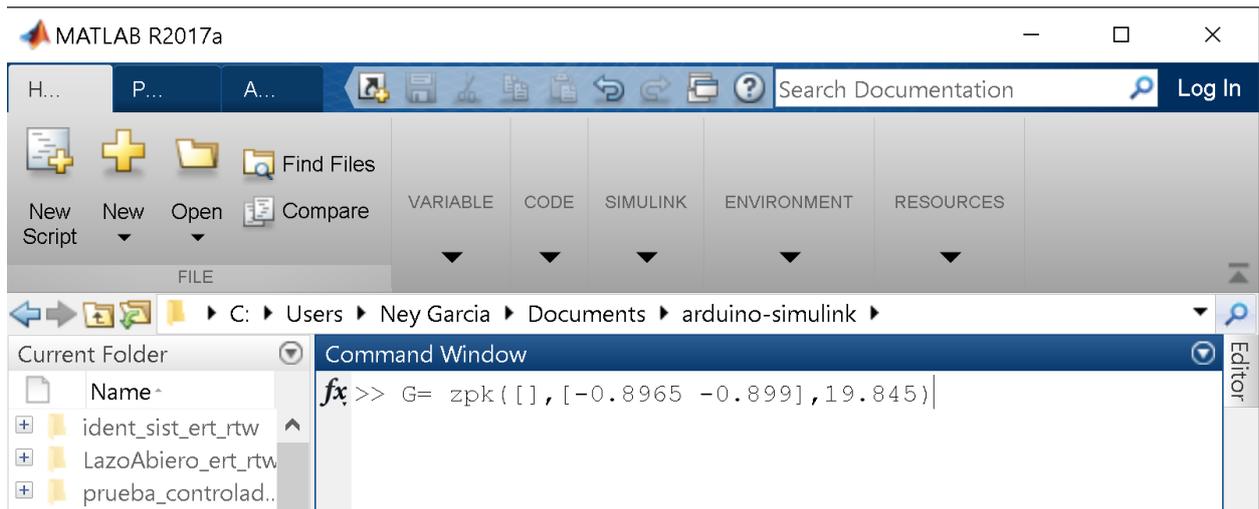
La función de transferencia es:

$$G(s) = \frac{19.845}{s^2 + 1.7955s + 0.80595}$$

Se conoce que el sistema tiene realimentación unitaria y se desea que cumpla con los siguientes requerimientos ante una entrada escalón:

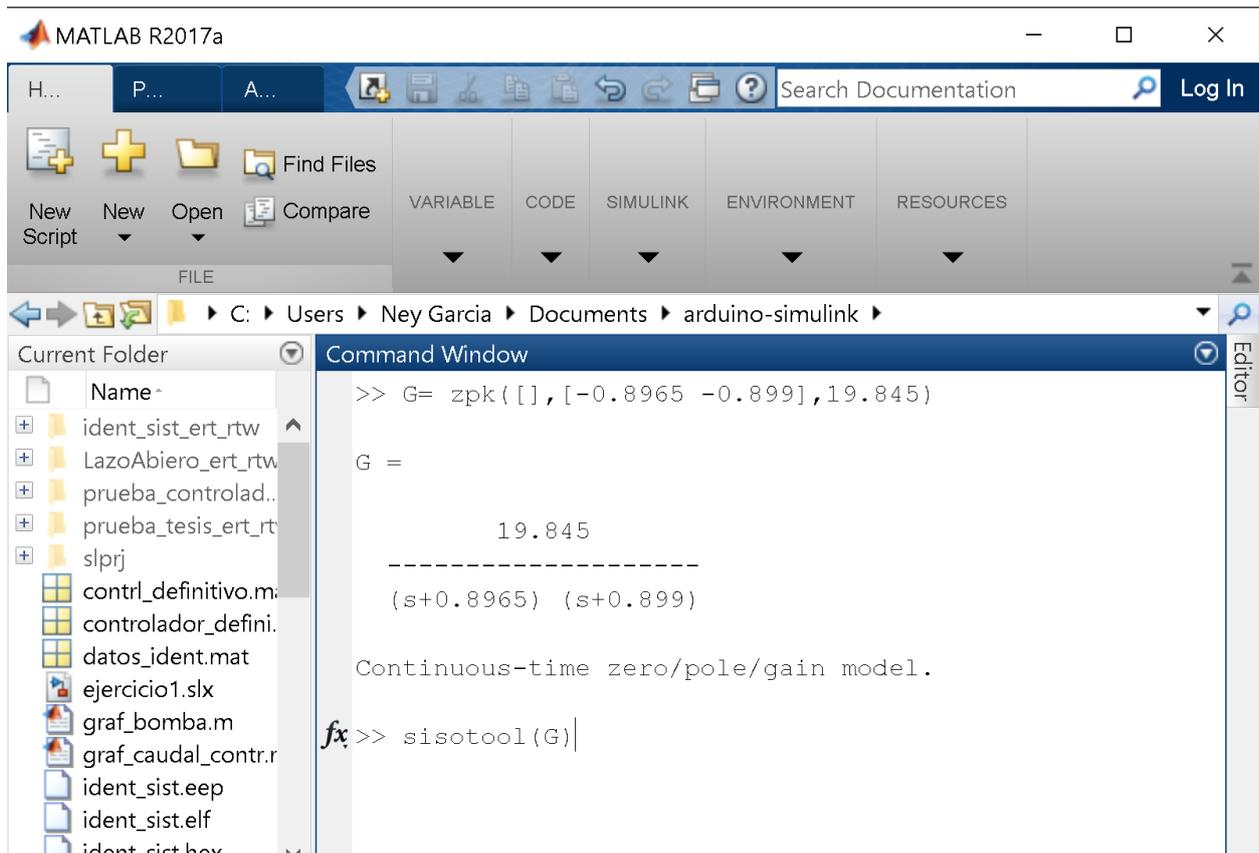
- Error de estado estacionario = 0
- Sobrenivel porcentual = 10%
- Tiempo de estabilización alrededor de 8 segundos

1. Abrir Matlab, en el editor ingresar la función de transferencia y guardar está en una variable, en este caso se usó la letra G; tal como se muestra en la figura 5.14.

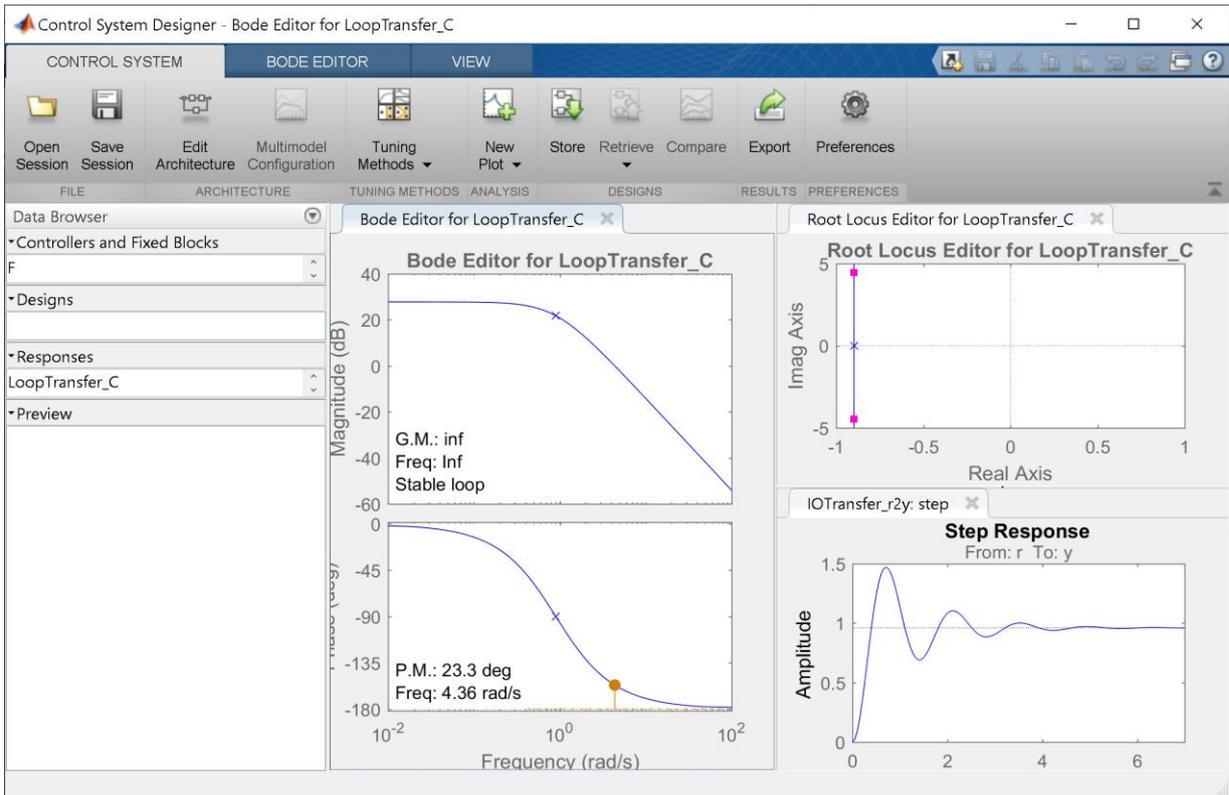


**Figura 5.14 Ingreso de Función de transferencia**

2. Ingresar el comando “sisotool(G)” y ejecutar el programa tal como se muestra en la figura 5.15. Una vez que se ejecuta se abrirá la ventana “Control System Designer” como se puede observar en la figura 5.16 para proceder a diseñar el controlador de la planta de control de flujo.

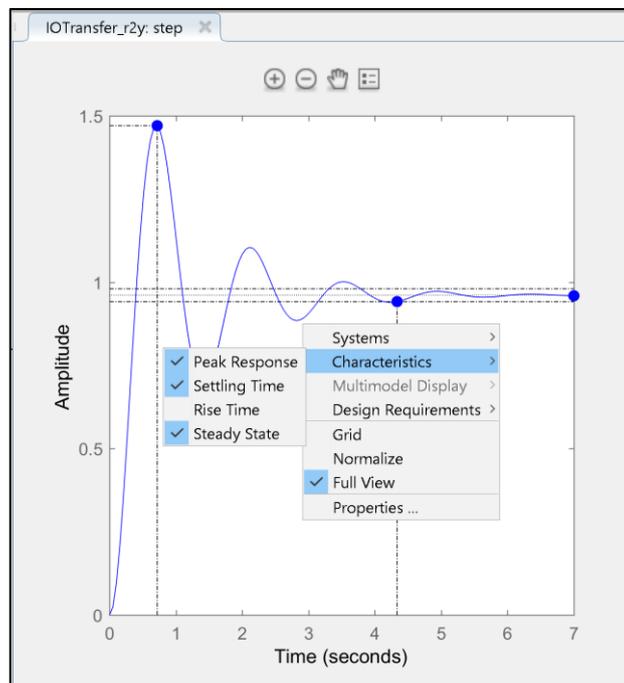


**Figura 5.15 Ingreso del comando Sisotool**



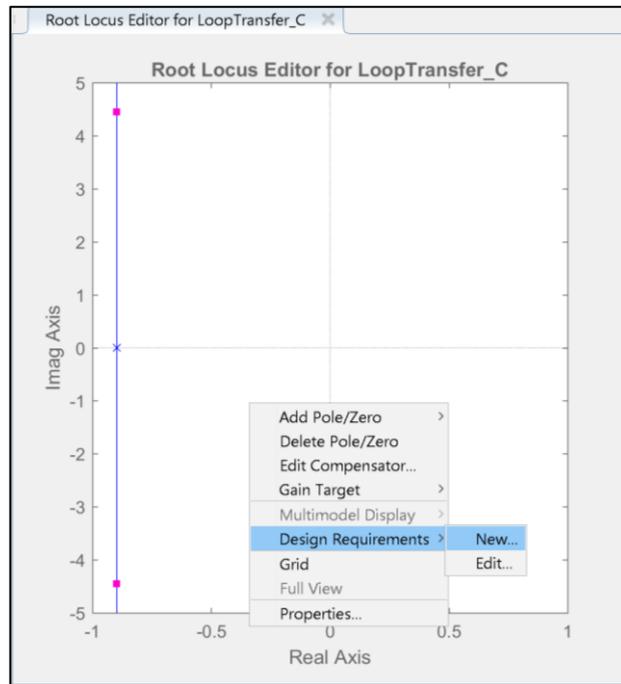
**Figura 5.16 Ventana Control System Designer**

- Nos ubicamos en el gráfico de respuesta escalón, damos clic derecho y seleccionamos la características que deseamos observar, tal como se muestra en la figura 5.17.

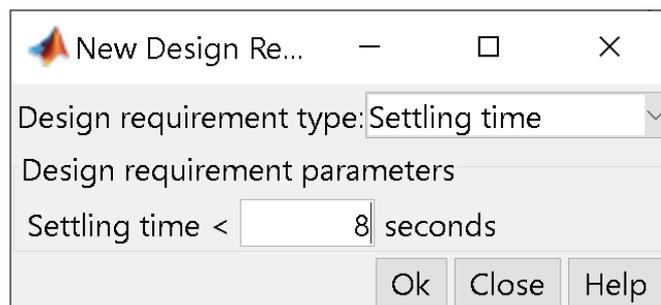


**Figura 5.17 Ingreso de Función de transferencia**

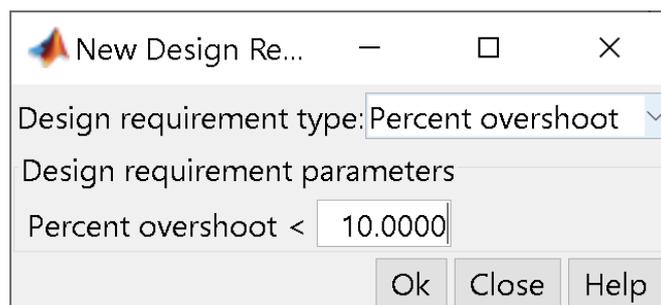
4. En la gráfica “Root Locus Editor” dar clic derecho en la opción “Design Requirements” para añadir las especificaciones del controlador tal como se muestra en las figuras 5.18, 5.19 y 5.20.



**Figura 5.18 Ingresar especificaciones del controlador**

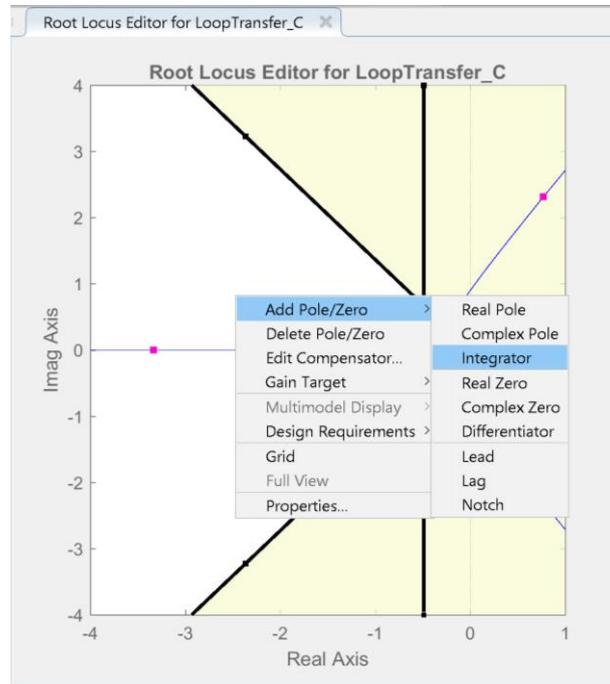


**Figura 5.19 Tiempo de estabilización**



**Figura 5.20 Porcentaje de overshoot**

5. Se procede a diseñar el controlador con el objetivo de alcanzar a las especificaciones requeridas al inicio de la práctica.
- Se añade un integrador en la gráfica de Root Locus Editor tal como se muestra en la figura 5.21.



**Figura 5.21 Añadir integrador**

- Así mismo se añade un cero tal como se muestra en la figura 5.22.



**Figura 5.22 Añadir un cero**

- Se va moviendo el lugar geométrico de las raíces con el cero con el objetivo de eliminar el cero con un polo, hasta obtener la respuesta escalón tal como se muestra en las figuras 5.23 y 5.24; obteniendo valores cercanos en comparación a las especificaciones con un tiempo de estabilización  $t_{ss}=7.8$ , un porcentaje de overshoot de 9.74% y un error de estado estacionario de  $ess=0$ .

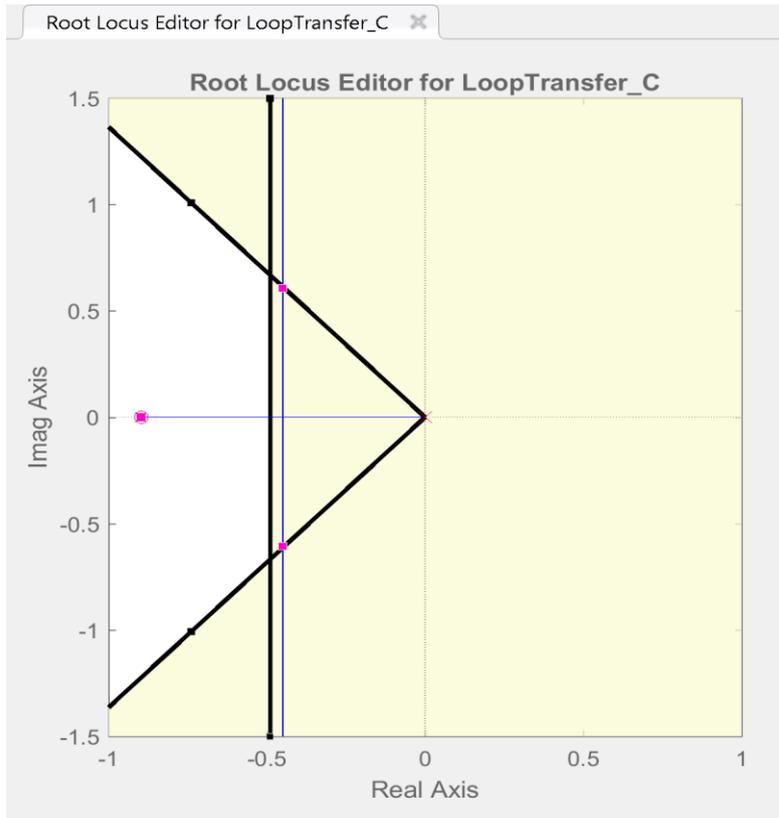


Figura 5.23 Diseño del controlador Root Locus

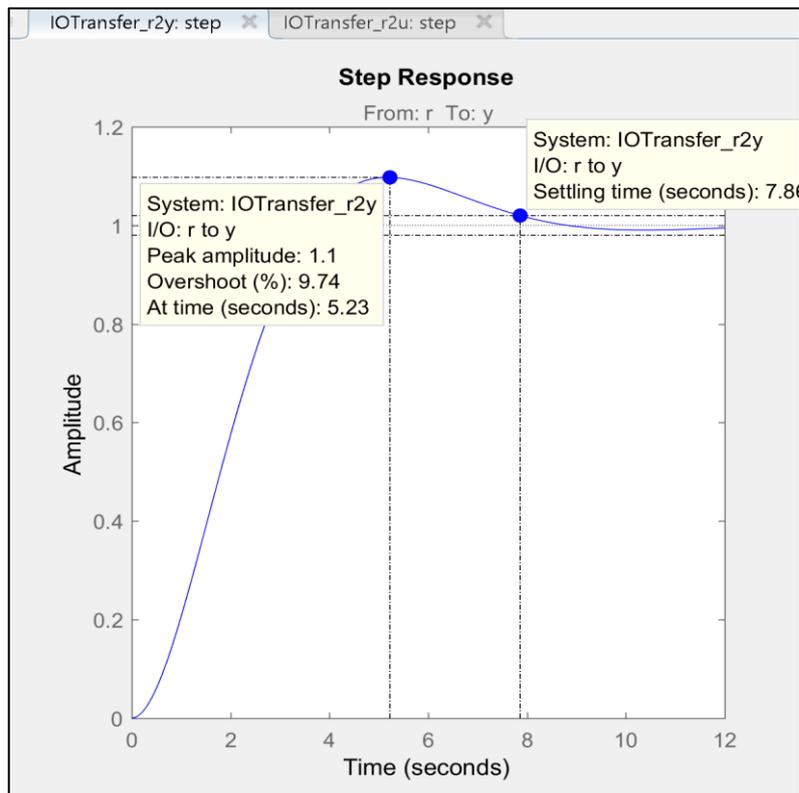


Figura 5.24 Diseño del controlador step response

- El controlador obtenido es un controlador PI tal como se puede observar en la figura 5.25 con las siguientes constantes:
  - $K_p = 0.02872$
  - $K_i = 0.02575$

```
>> tf(C)

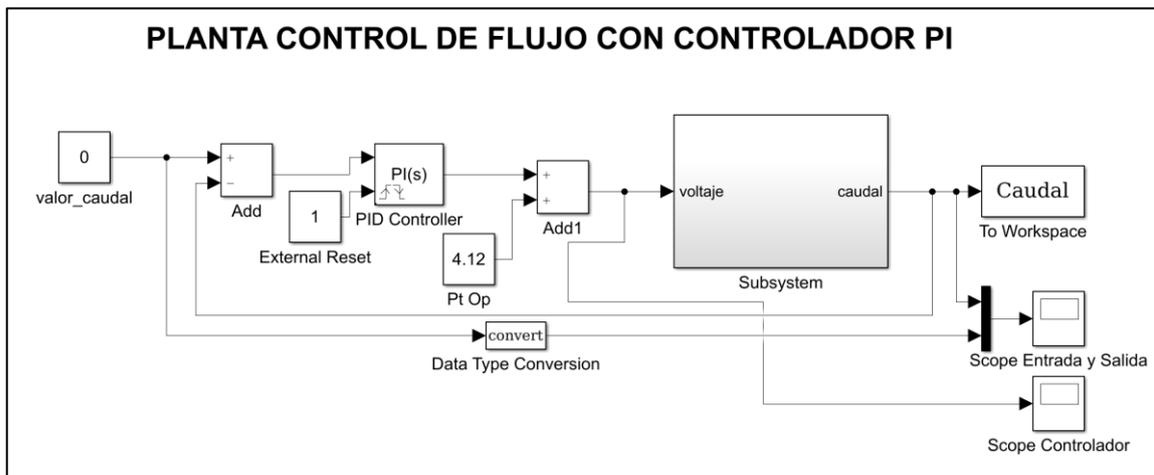
ans =

    0.02872 s + 0.02575
    -----
                   s

Name: C
Continuous-time transfer function.
```

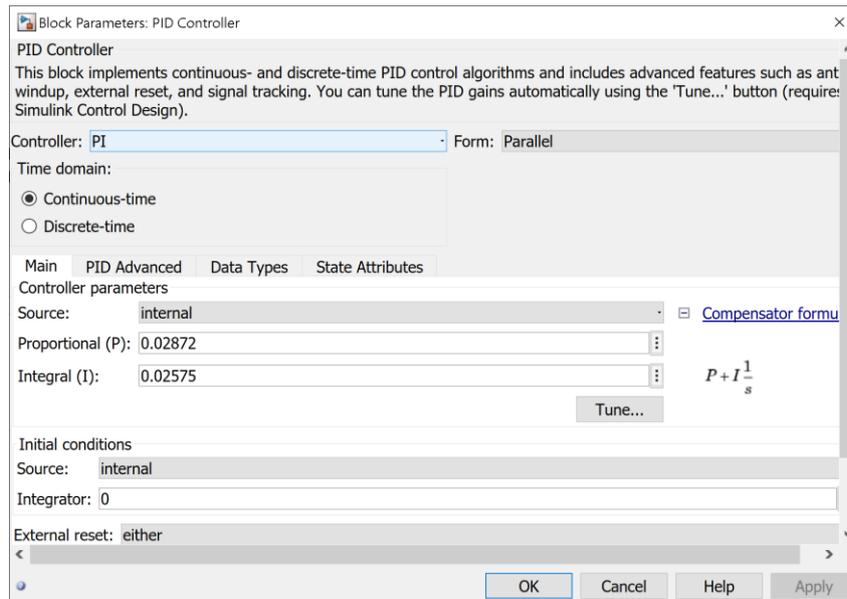
**Figura 5.25 Controlador PI**

6. Una vez que se ha diseñado el controlador se procede a comprobar con la planta de control de flujo para verificar que cumpla con los requerimientos.
7. En Simulink se procede a abrir el diagrama de bloque en lazo cerrado de la planta control de flujo tal como se muestra en la figura 5.26.



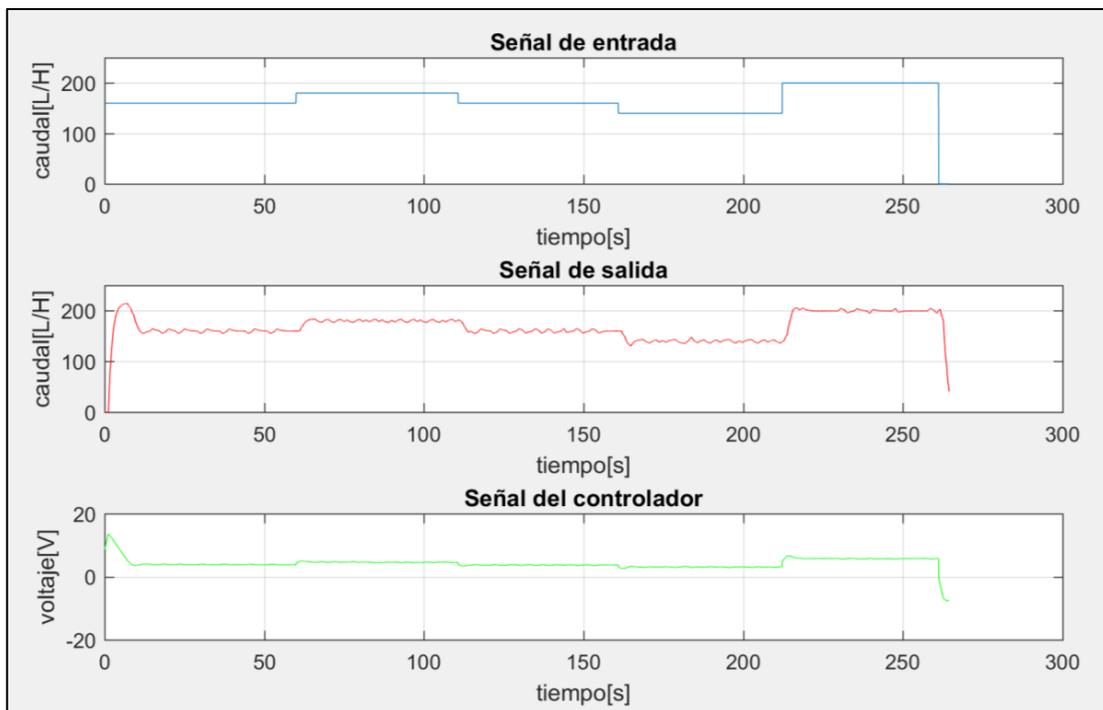
**Figura 5.26 Diagrama de bloques en lazo cerrado**

8. En el bloque PID Controller, se selecciona que corresponde a un controlador PI y así mismo se añaden las constantes del controlador diseñado tal como se muestra en la figura 5.27.



**Figura 5.27 Bloque PID controller**

En el bloque valor\_caudal se procede a ingresar valores de caudal a los que queremos que trabaje la planta y observar el comportamiento de la misma, verificando que el controlador actúe de manera correcta tal como se puede observar en la figura 5.28 donde se determinó que el rango en el que el controlador funciona correctamente es entre 120 y 220 L/H.



**Figura 5.28 Prueba del controlador**