



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Facultad de Ingeniería en Electricidad y Computación

“IMPLEMENTACIÓN DEL ALGORITMO DE CONTROL AUTO-
ADAPTATIVO DE PREDICCIÓN EXTENDIDA, EVALUADO
SOBRE UN SISTEMA DE REGULACION DE CAUDAL”

TRABAJO DE TITULACIÓN

Previa a la obtención del Título de:

MAGÍSTER EN AUTOMATIZACION Y CONTROL

Presentado por:

LUIS ANTONIO UBE CONSUEGRA

FERNANDO ADRIAN RODRIGUEZ GONZALEZ

GUAYAQUIL – ECUADOR

2019

AGRADECIMIENTO

No me alcanzarán estas sencillas palabras para que entiendan lo agradecido que estoy con Dios, con mi Padres: Tlgo. Luis Antonio Ube Fajardo y Sra. Deysy Eddy Consuegra Sayas; hermano: Israel Paul Ube Consuegra; a mi esposa la Sra. Brigitte Leonor Eugenio Acosta e hijos: Mia Lynnete Ube Eugenio y Liam Exequiel Ube Eugenio; de igual forma a mis amigos que formaron parte en este proceso; por brindarme su apoyo incondicional en todo momento, enseñándome que sin sacrificio, dedicación y voluntad propia no se podrían alcanzar las metas que uno se propone en la vida.

Agradecer al Msc. Holger Cevallos e Ing. Gabriel Intriago por ser los mentores en el desarrollo de este proyecto.

Luis Antonio Ube Consuegra

Quiero agradecerle a Dios por permitirme seguir superando, porque gracias a él he alcanzado otro escalón más en mi vida profesional. A mi madre por darme la vida y enseñarme que uno puede seguir superándose si se lo propone. A mis hijos por ser los motores que me impulsan a seguir luchando. A esa personita especial que me ayudó y me levantó cuando el mundo se me vino encima. A mi compañero de tesis por el esfuerzo y dedicación que le puso cada día. A mis amigos por el apoyo brindado durante el tiempo que duró la MACI.

De una manera especial mis agradecimientos al Msc. Holger Cevallos e Ing. Gabriel Intriago quienes con sus conocimientos nos iban guiando en el desarrollo de nuestro proyecto de titulación.

Fernando Adrián Rodríguez González

DEDICATORIA

Le deseo brindar un homenaje a Dios por darme uno de los pilares más fundamentales de la vida, la cual es la familia, donde se resume toda la clave de mis éxitos.

A mis padres, hermano, esposa e hijos que este triunfo es para ustedes, debido que sin la presencia de ustedes no hubiera sido capaz de seguir adelante con la culminación de este proyecto.

Luis Antonio Ube Consuegra

Este trabajo se lo dedico a Dios, a mi madre, a mis hijos y a todas aquellas personas que estuvieron conmigo en todas las situaciones de mi vida y que con su apoyo incondicional me supieron guiar y me dieron sabios consejos para no desmayar.

Fernando Adrián Rodríguez González

TRIBUNAL DE EVALUACION

Nombre del Profesor
SUBDECANO DE LA FIEC

Msc. Holger Cevallos
DIRECTOR DEL TRABAJO DE TITULACION

Nombre del Profesor
MIEMBRO PRINCIPAL DEL TRIBUNAL

DECLARACION EXPRESA

“La responsabilidad y la autoría del contenido de este trabajo de titulación nos corresponde exclusivamente y 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”

Luis Antonio Ube Consuegra

Fernando Adrián Rodríguez González

RESUMEN

En el capítulo 1 se describe el problema investigativo con la justificación y sus objetivos, seguido del tipo de metodología que se utilizará para el desarrollo del proyecto, que contribuya al inicio de la búsqueda de la implementación de nuevos controladores en un proceso donde se necesita regular el caudal de un líquido.

En el capítulo 2 se detallan antecedentes y se hace una breve introducción a los controladores EPSAC, PID FUZZY y PID_COMPACT propio del S7-1200, los mismos que serán utilizados en el sistema de regulación de caudal de un líquido.

En el capítulo 3 se describe el diseño de los controladores desde la identificación matemática del proceso, la obtención de la función de transferencia de la planta para el rango de trabajo establecido y la sintaxis de programación y/o reglas de control de cada uno de los controladores EPSAC, PID FUZZY y PID_COMPACT junto con el diseño de interfaz.

En el capítulo 4 se mencionarán las partes de las cuales está compuesta la Planta Reguladora de Caudal y se describirá la implementarán los controladores EPSAC, PID FUZZY y PID_COMPACT sobre esta Planta.

En el capítulo 5, se detallan los resultados obtenidos entre la implementación de los controladores EPSAC, PID FUZZY y PID_COMPACT y el sistema de regulación de caudal.

ÍNDICE GENERAL

AGRADECIMIENTO	ii
DEDICATORIA	iii
TRIBUNAL DE EVALUACION	iv
RESUMEN	vi
ÍNDICE GENERAL DE FIGURAS	4
ÍNDICE GENERAL DE TABLAS	6
ABREVIATURAS	7
CAPITULO 1	8
1. PLANTEAMIENTO DEL PROBLEMA	8
1.1. Descripción del problema	8
1.2. Justificación	8
1.3. Solución propuesta	8
1.4. Objetivos	9
1.4.1. Objetivo General	9
1.4.2. Objetivos Específicos	9
1.5. Metodología	9
1.6. Resultados esperados	10
1.7. Elementos diferenciadores o innovadores	10
CAPITULO 2	11
2. ESTADO DEL ARTE	11
2.1. EPSAC	11
2.1.1. Antecedentes	11
2.1.2. Introducción	11
2.2. PID FUZZY	12
2.2.1. Antecedentes	12
2.2.2. Introducción	12
2.3. PID_COMPACT	14
2.3.1. Antecedentes	14
2.3.2. Introducción	14

CAPITULO 3	16
3. DISEÑO DE LOS CONTROLADORES	16
3.1. EPSAC	16
3.1.1. Diseño del controlador EPSAC	16
3.1.1.1. Identificación matemática del proceso	16
3.1.1.1.1. Estimulación del proceso	18
3.1.1.1.2. Interpretación de la respuesta	20
3.1.1.1.3. Importación de los datos	20
3.1.1.1.4. Selección del rango de datos	21
3.1.1.1.5. Identificación del modelo del proceso	22
3.1.2. Función de minimización de costo	23
3.1.2.1. Introducción	23
3.1.2.2. Diseño de las funciones	26
3.1.2.2.1. Algoritmo de predicción $xt + k t$	26
3.1.2.2.2. Algoritmo de predicción $nt + k t$	27
3.1.3. Estructura de la programación del EPSAC en MATLAB	27
3.1.3.1. Declaración de variables	28
3.1.3.2. Función de transferencia	28
3.1.3.3. Condiciones iniciales	29
3.1.3.4. Periodo de muestreo	29
3.1.3.5. Función del controlador EPSAC	30
3.1.4. Diseño de interfaz en simulink	31
3.2. PID FUZZY	32
3.2.1. Diseño del controlador PID FUZZY	32
3.2.1.1. Control Difuso	32
3.2.1.1.1. Conjuntos borrosos	32
3.2.1.1.2. Reglas del control	34
3.2.1.1.3. Salida o defusificación	34
3.2.1.2. PID de Simulink	36
3.2.2. Diseño de interfaz en Simulink	36
3.3. PID_COMPACT	37

3.3.1. Introducción	37
3.3.2. Diseño del algoritmo de control.....	39
CAPITULO 4.....	44
4. IMPLEMENTACION DE LOS CONTROLADORES	44
4.1. Controlador EPSAC.....	48
4.2. Controlador PID FUZZY	51
4.3. Controlador PID_COMPACT.....	52
CAPÍTULO 5.....	55
5. RESULTADOS.....	55
5.1. Respuestas de los controladores.....	55
CONCLUSIONES.....	57
RECOMENDACIONES	57
BIBLIOGRAFÍA.....	59

ÍNDICE GENERAL DE FIGURAS

Figura 1. Diagrama de proceso del controlador Fuzzy.....	13
Figura 2. Diagrama de bloques en lazo cerrado.	16
Figura 3. Diagrama de flujo para la identificación del sistema.	17
Figura 4. Relación lineal de las variables de entrada analógica vs caudal.	18
Figura 5. Relación lineal de las variables del set point vs salida analógica.	19
Figura 6. Respuesta del sistema en lazo abierto.....	19
Figura 7. Cambio de punto operación de la repuesta en lazo abierto.....	20
Figura 8. Ventana del comando “Ident” de Matlab.....	21
Figura 9. Selección de datos para la identificación del sistema.....	21
Figura 10. Gráfica del sistema identificado.....	22
Figura 11. Información del modelo de la función de transferencia del proceso.	23
Figura 12. Principio del MBPC.....	24
Figura 13. Diagrama de bloques de la estrategia del MBPC.	26
Figura 14. Modelo de implementación serie y serie-paralelo.	27
Figura 15. Programa en Simulink del control EPSAC.....	32
Figura 16. Conjunto borroso de la entrada del controlador Fuzzy.	33
Figura 17. Parametrización de la salida del controlador Fuzzy.....	35
Figura 18. Ventana de parametrización del controlador PID Controller	36
Figura 19. Programación del controlador PID FUZZY.....	37
Figura 20. Descripción PID_COMPACT.....	39
Figura 21. Bloque de la instrucción tecnológica del PID_COMPACT del TIA Portal.....	39
Figura 22. Asignación de las variables de entradas y salida	40
Figura 23. Parametrización de los ajustes básicos del PID_COMPACT.....	41
Figura 24. Parametrización de los límites del valor de salida.	41
Figura 25. Valores del PID adquiridos por la sintonización del bloque.....	42
Figura 26. Programación del controlador PID_COMPACT	42
Figura 27. Declaración de las variables en el OPC Server.	43
Figura 34. Estructura del sistema regulador de caudal.	44
Figura 35. Válvula controladora de caudal.	45
Figura 36. Rotámetro.	46
Figura 37. Sensor de caudal con salida analógica.....	47
Figura 28. Repuesta 1 del controlador EPSAC.	49
Figura 29. Repuesta 2 del controlador EPSAC.	50
Figura 30. Repuesta 1 del controlador PID FUZZY.	51
Figura 31. Repuesta 2 del controlador PID FUZZY.	52
Figura 32. Repuesta 1 del controlador PID_COMPACT.....	53
Figura 33. Repuesta 2 del controlador PID_COMPACT.....	53
Figura 38. Gráfica 1 de las repuestas de los tres controladores a una entrada paso de 0.6 a 0.8 m ³ /h.....	55

Figura 39. Gráfica 2 de las repuesta de los controladores a una entrada paso de 0.7 a 0.9 m^3/h 56

ÍNDICE GENERAL DE TABLAS

Tabla 1. Velocidad de muestreo de datos de la CU1214C.	48
Tabla 2. Cuadro comparativo número uno de las repuestas de los controladores.	55
Tabla 3. Cuadro comparativo número dos de las repuestas de los controladores	56

ABREVIATURAS

EPSAC	Control Auto-Adaptativo de Predicción Extendida.
PID	Controlador Proporcional, Integral y Derivativo.
PID FUZZY	Controlador Proporcional, Integral y Derivativo Difuso.
ESPOL	Escuela Superior Politécnica del Litoral.
MBPC	Modelos Basados en un Control Predictivo.
MACI	Maestría en Automatización y Control Industrial.

CAPITULO 1

1. PLANTEAMIENTO DEL PROBLEMA

1.1. Descripción del problema

En la actualidad, podemos encontrar en el laboratorio de instrumentación industrial de la ESPOL, una planta reguladora de caudal que está automatizada con un PLC S7-1200 y controlada a través de un controlador de procesos.

Al aplicar un control PID a la planta reguladora, podemos apreciar que no se realiza una acción de control apropiada sobre la regulación del caudal. Por tal motivo y debido a que no existe información de la implementación del control Auto-Adaptativo de Predicción Extendida (EPSAC) para esta planta, se desea aportar con el análisis del performance de este controlador usando EPSAC para que se lo evalúe en el sistema de regulación de caudal.

1.2. Justificación

Se desea implementar el controlador EPSAC en la planta reguladora de caudal del laboratorio de Instrumentación Industrial debido a que en la documentación revisada no existen capítulos ni secciones que evidencien si es factible o no la utilización de este tipo de algoritmo de control en un sistema dinámico de control de caudal.

1.3. Solución propuesta

En vista que la planta con sistema de regulación de caudal no tiene un control predictivo, se desea realizar el análisis con los conocimientos adquiridos durante el desarrollo de la Maestría en Control y Automatización Industrial (MACI). Este análisis consiste en aplicar el controlador predictivo EPSAC y compararlo con el PID FUZZY y PID_COMPACT del S7 1200.

1.4. Objetivos

1.4.1. Objetivo General

Realizar el análisis comparativo del desarrollo de tres tipos de algoritmos EPSAC, PID FUZZY y PID_COMPACT para el control del flujo de caudal de una planta utilizando los programas MATLAB y TIA Portal.

1.4.2. Objetivos Específicos

- Realizar la identificación de la planta.
- Implementar un controlador EPSAC que compruebe si se ejerce una acción de control adecuada ante la presencia de una perturbación en el sistema.
- Implementar un controlador PID_COMPACT que controle la planta.
- Implementar un controlador PID FUZZY que controle el proceso ante la presencia de una perturbación en el sistema.
- Desarrollar un sistema de supervisión y monitoreo de datos de la planta.
- Analizar mediante un estudio comparativo el rendimiento entre los controladores EPSAC, PID FUZZY y PID_COMPACT.

1.5. Metodología

Para llevar a cabo los objetivos descritos en la sección anterior es preciso empezar por la identificación de la planta sobre la cual vamos a implementar el algoritmo de control. Para la obtención de los datos preliminares con el fin de poder realizar la identificación, se realizarán mediciones cambiando su punto de ajuste o set point. Una vez identificada la planta, se diseñará e implementará un algoritmo de control EPSAC en nuestro proceso que controle su salida ante cualquier perturbación presentada en el sistema.

Las señales tanto de entrada (transmisor de flujo) como de salida (válvula de control) llegarán a un controlador lógico programable (PLC) en el cual se desarrollarán sentencias para leer estas señales logrando de esta forma que la planta sea autónoma. Finalmente, se desarrollará un sistema de

monitoreo y supervisión de los datos de la planta automatizada usando MATLAB.

1.6. Resultados esperados

Con la implementación del algoritmo EPSAC al sistema de regulación de caudal, se espera que éste genere una apropiada rapidez en su respuesta y no genere un sobre nivel en la salida cuando se presente algún cambio en la entrada de la planta o presencia de alguna perturbación.

Además, con los tres tipos de controladores detallados anteriormente, se espera conocer su performance para poder comparar tres parámetros entre las repuestas, los cuales son: tiempo de estabilización (T_s), sobre nivel porcentual (O_s) y el error (e) durante la acción de control ya sea esta al momento de realizar un cambio en la entrada o ser presente una perturbación.

1.7. Elementos diferenciadores o innovadores

Con la evaluación de los controladores predictivos en el sistema de regulación de caudal, evidenciaremos su desempeño y así aportaremos de forma innovadora ya que en el campo industrial no son muchos los estudios que podemos encontrar relacionados a la regulación de caudal a través del controlador EPSAC. Adicionalmente la evaluación del desempeño del controlador se lo realizará en una planta real la cual está en el Laboratorio de Instrumentación Industrial de la ESPOL.

La aplicación del Modelo de Control Predictivo “MPC” tiene una gran acogida por el gran desarrollo de aplicaciones computacionales que hay en la actualidad, esto ha permitido implementarlo en aplicaciones de procesos complejos como se lo puede apreciar en: [8], [9], [10], [11] y [12].

CAPITULO 2

2. ESTADO DEL ARTE

En este capítulo se mencionarán los antecedentes que tienen los controladores EPSAC, PID FUZZY y PID_COMPACT en un proceso regulador de caudal, aportando una breve introducción teórica de cada uno de los algoritmos de control.

2.1. EPSAC

2.1.1. Antecedentes

Según investigación previa no se encontró aporte alguno del EPSAC controlando un proceso donde se regule el caudal de un líquido, por tal motivo se desea contribuir en la implementación de esta técnica de control para este proceso y verificando si la respuesta del sistema es igual al valor deseado sin generar sobre picos y disminuyendo el tiempo de estabilización a cambios generados en sus entradas.

2.1.2. Introducción

Este algoritmo de control tiene su base en un MBPC. El MBPC es una técnica de control desarrollada en torno a dos criterios claves, los cuales son:

- El uso explícito de un modelo en proceso que pronostica la salida del proceso en el futuro durante un instante de tiempo.
- Cálculo de una acción de control óptima basada en la minimización del costo de una o más funciones, incluyendo la posibilidad de que existan restricciones en las variables del proceso.

La diversidad de los algoritmos de control MBPC, tienen diferencia principalmente en:

- El tipo de modelo utilizado para representar el proceso y sus perturbaciones.

- Las funciones de costo ayudan a minimizar el error con o sin restricciones de sus variables.

Las estrategias de un MBPC son:

- Un modelo de procesos utiliza la predicción según el avance de la respuesta del sistema en función del futuro.
- Minimiza el índice del costo específico, el cual incluye los errores entre los resultados del proceso deseados y pronosticados.

En resumen, se reconocen algunos elementos importantes que caracterizan al MBPC

- Predicción mediante un modelo de proceso.
- Especificación de una trayectoria de referencia.
- Estructuración de la ley futura de control.
- Definición de una función de costo.
- Cálculo de la acción de control óptimo.

Tal como lo menciona Robbin De Keyser en su documento: [7]

2.2. PID FUZZY

2.2.1. Antecedentes

El aporte investigativo de Alejandra Casa [6] indica que la implementación de este control en un proceso donde se regule el caudal de un líquido, su acción de control tiene el siguiente comportamiento:

- Lentitud en la acción de control.
- Durante la presencia de una perturbación, la acción de control es tardía.
- Tiene un mayor tiempo de estabilización.

2.2.2. Introducción

Un controlador PID FUZZY es aquel que está compuesto por dos tipos de controladores:

- Lógica difusa. Es implementada en sistemas donde la información tiene un alto grado de imprecisión. La estructura del controlador difuso es la siguiente:

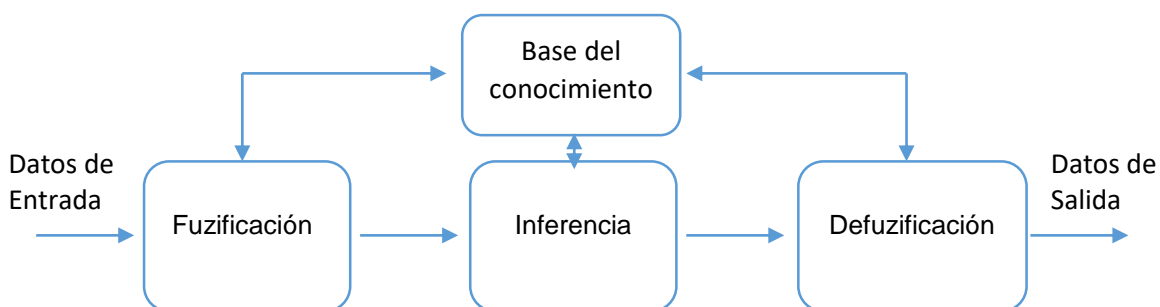


Figura 1. Diagrama de proceso del controlador Fuzzy.

En el bloque fuzificador ingresan los datos, los cuales son procesados para determinar el grado de membresía que tendrán dentro del controlador, el resultado pasa a la etapa de inferencia que junto con la base de conocimiento determinarán la decisión con la que actuará el sistema y dicho valor obtenido, es ingresado en la última etapa de defuzificación donde son interpretados y transformados en valores no difusos útiles para el proceso que se desea controlar.

Cabe recalcar que, si el dato ingresado al método de inferencia se encuentra declarado en los conjuntos difusos y en la base del conocimiento donde se hallan sus reglas, el controlador podrá tomar una decisión.

- Control PID. Es el algoritmo de control más utilizado por su fácil acceso y configuración de sus parámetros. La base de este control consiste en la incorporación de tres parámetros distintos: proporcional, integral y derivativo. La variable proporcional depende del error actual (K_p), la integral se obtiene integrando el error (K_i) y la derivativa es proporcional a la derivada de la señal o velocidad del error (K_d).

Entonces la integración de los algoritmos de control Fuzzy con el PID puede ser utilizado para implementarlos en procesos no lineales, en el control de sistemas lineales retardados y exactitud en su respuesta.

2.3. PID_COMPACT

2.3.1. Antecedentes

El objeto tecnológico PID_COMPACT que se encuentra en la librería de bloques en el TIA PORTAL es utilizado en muchas aplicaciones industriales por su robustez y bajo costo en su implementación.

María Guano y Diana Romero indican que la implementación de este controlador genera oscilaciones continuas en la variable de control cuando el error es aproximadamente cero durante la ejecución de la optimización fina. Cabe recalcar que la variable controlada fue aplicada a la bomba de agua.

2.3.2. Introducción

En el área de ajustes básicos de la ventana de inspección o de configuración se pueden configurar las siguientes propiedades del objeto tecnológico PID_COMPACT

- Magnitud física.
- Lógica de control.
- Comportamiento en arranque después de un reinicio.
- Punto de ajuste.
- Valor del proceso.
- Valor de salida.

Un aumento en el valor de salida generalmente tiene la intención de causar un aumento en el valor del proceso. Esto se conoce como una lógica de control normal. El controlador PID_COMPACT no funciona con ganancia proporcional negativa. En este caso debe seleccionar *“Invertir lógica de control”* para reducir el valor del proceso con un valor de salida más alto.

El objeto tecnológico necesita una cierta cantidad de tiempo para realizar cambios en la salida. El tiempo de muestreo del algoritmo PID representa el tiempo entre dos cálculos del valor de salida, el cual se calcula durante la sintonización y se redondea a un múltiplo del tiempo del ciclo.

CAPITULO 3

3. DISEÑO DE LOS CONTROLADORES

En este capítulo se detallará el diseño de los algoritmos de control que regulen el caudal de un fluido a un valor deseado. El diseño de los controladores deberá tener las siguientes características, regular el fluido del agua comprendido entre el rango de $[0.6 \frac{m^3}{h} a 1.0 \frac{m^3}{h}]$ y su acción de control será llevada a una válvula de control de tipo proporcional que tiene una entrada analógica 4-20mA. La salida será monitoreada a través de un dispositivo electrónico que cense la cantidad de caudal del agua. A continuación, se realizará un diagrama de bloques de la planta en la cual se va a incorporar el controlador.

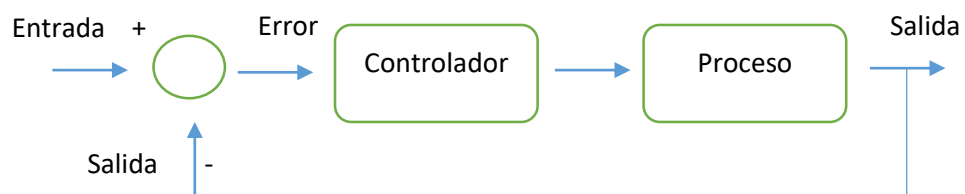


Figura 2. Diagrama de bloques en lazo cerrado.

3.1. EPSAC

3.1.1. Diseño del controlador EPSAC

Se simplifica el diseño constructivo del algoritmo de control en dos etapas fundamentales.

3.1.1.1. Identificación matemática del proceso

En el siguiente diagrama de flujo se detalla la secuencia de pasos necesarios para la obtención de la función de transferencia.

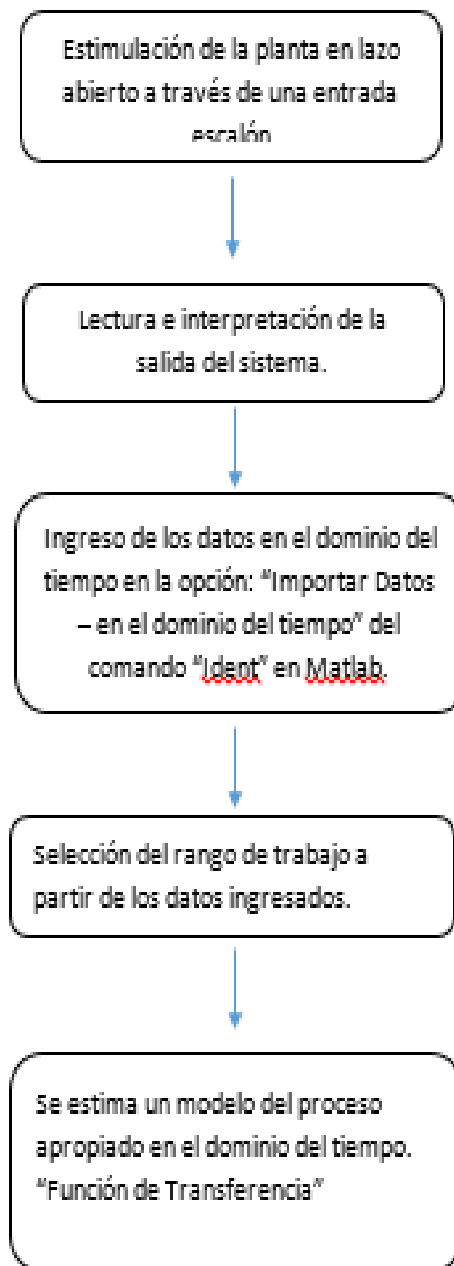


Figura 3. Diagrama de flujo para la identificación del sistema.

A continuación, se describirá como se obtuvo la función de transferencia.

3.1.1.1.1. Estimulación del proceso

Se inició con la programación de la entrada analógica IW96 y la salida QW96 de la CPU 1212C con la siguiente lógica de control

Entrada analógica

La señal de entrada tiene una naturaleza distinta a la de salida porque cuando no existe un flujo que pase por el sensor, éste tiene un offset de $0.3 \frac{m^3}{h}$ y cuando circula un fluido por el sensor, éste comienza a dar una señal de lectura que inicia desde el offset hasta $1.2 \frac{m^3}{h}$ que es el caudal máximo.

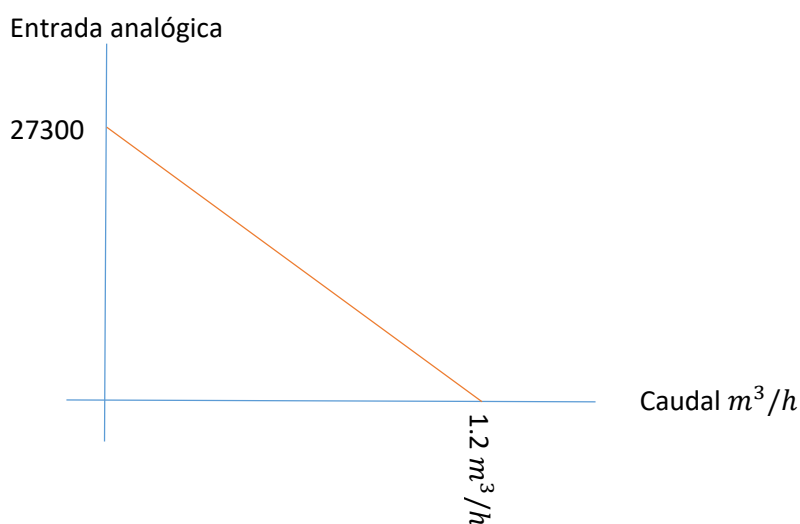


Figura 4. Relación lineal de las variables de entrada analógica vs caudal.

Salida analógica

Si se desea un caudal de $0 \frac{m^3}{h}$ se cierra toda la válvula y si se desea un caudal de $1.2 \frac{m^3}{h}$ se abra toda la válvula.

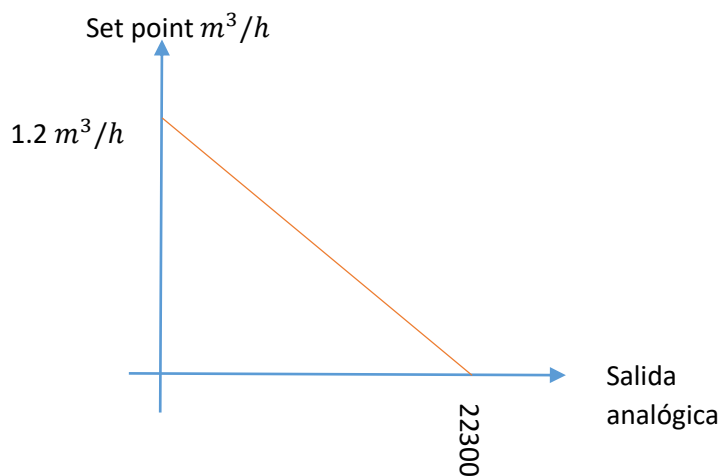


Figura 5. Relación lineal de las variables del set point vs salida analógica.

Luego se arrancó el sistema en lazo abierto sin aplicar ningún cambio en su entrada, posteriormente se comenzaron a generar varios cambios en la entrada del proceso, tal como se muestra en la siguiente figura.

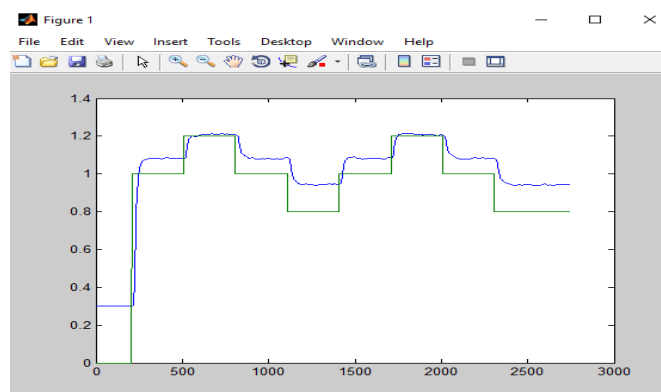


Figura 6. Respuesta del sistema en lazo abierto.

Donde,

La señal de color verde es el valor deseado y la señal de color azul es la salida del sistema.

3.1.1.1.2. Interpretación de la respuesta

Se puede apreciar en la figura 6 que el punto de operación de la respuesta tiene un set point inicial de 1. Si se procede a obtener el modelo matemático del proceso con este punto de referencia, puede generar una identificación matemática del sistema errónea, tal como se ilustra en el documento [17]. Ante esta generación posiblemente errónea, se aconseja desplazar el punto de operación de 1 a 0 para evitar una no adecuada identificación. A continuación, se mostrará la respuesta del sistema con el punto de operación en cero.

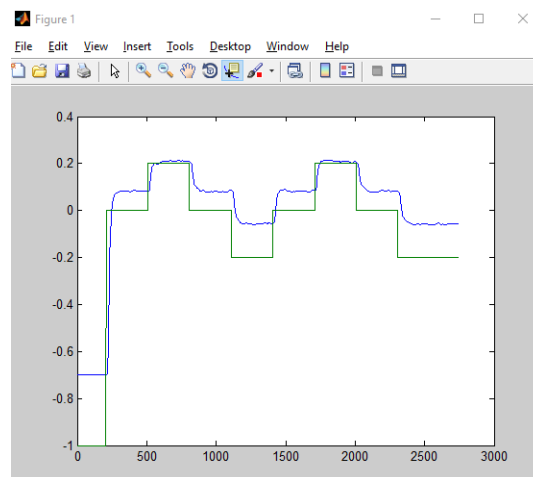


Figura 7. Cambio de punto operación de la repuesta en lazo abierto.

3.1.1.1.3. Importación de los datos

Se ejecuta el comando *ident* en MATLAB para comenzar con el inicio de la obtención del modelo matemático de la planta. Para aquello se necesita que los datos obtenidos en tiempo continuo se los guarden con el formato “.mat”, luego se importa la información a la ventana de ejecución *System Identification Tool* en el menú “Import Data” opción *time domain data*, como se muestra en la siguiente imagen:

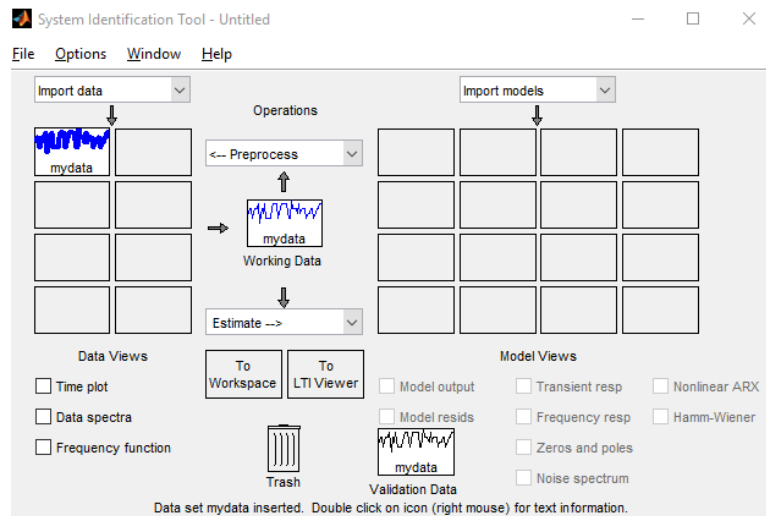


Figura 8. Ventana del comando “Ident” de Matlab.

3.1.1.1.4. Selección del rango de datos

Para obtener un modelo matemático con mayor exactitud en la predicción de la función de transferencia del proceso, se debe seleccionar un rango que abarque toda la dinámica de trabajo.

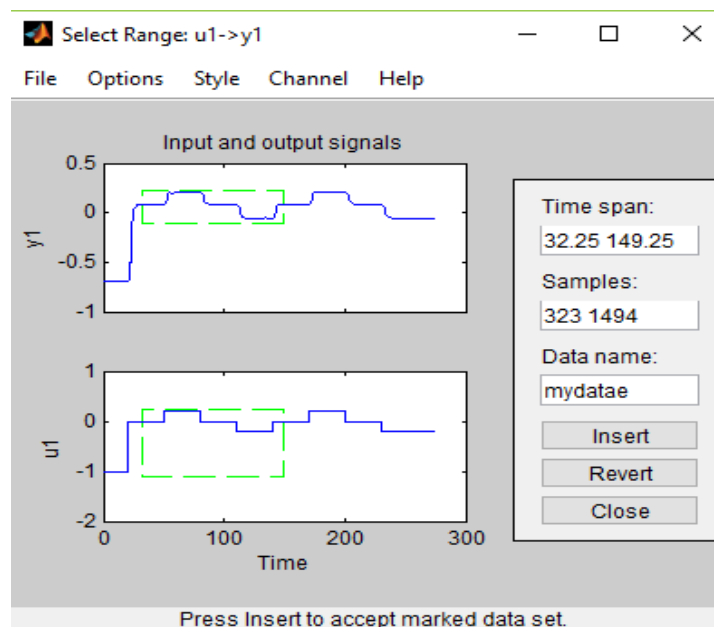


Figura 9. Selección de datos para la identificación del sistema.

3.1.1.1.5. Identificación del modelo del proceso

Se identificó un modelo matemático con las siguientes características: una integral, un zero y dos polos complejos, obteniendo una aproximación del **92.48%**, como se muestra a continuación

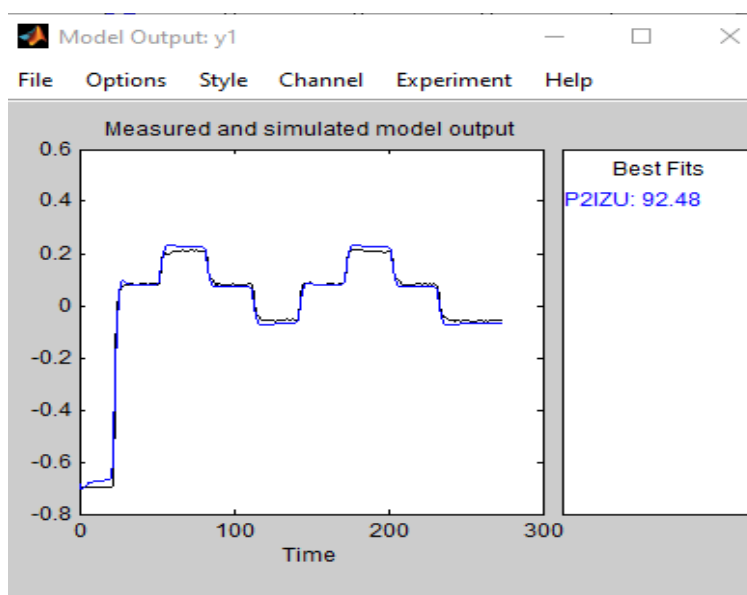


Figura 10. Gráfica del sistema identificado.

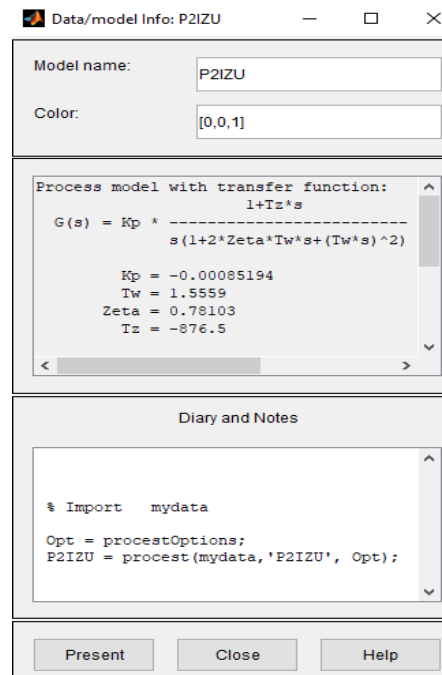


Figura 11. Información del modelo de la función de transferencia del proceso.

Donde:

K_p Ganancia estática.

T_z Proceso del zero.

T_w Inversa a la frecuencia natural.

Zeta Coeficiente de amortiguamiento

Y su función de transferencia está representada por

$$G = \frac{0.7467 s - 0.0008519}{5.884 s^4 + 2.421 s^3}$$

3.1.2. Función de minimización de costo

3.1.2.1. Introducción

Antes de entrar al diseño de las funciones de minimización de costo para que produzca una acción de control optima, se detallará la notación de las variables que están involucrada en este control:

- t : tiempo discreto.
- $u(t)$: entrada del proceso, (variable manipulada).
- $y(t)$: salida del proceso, (variable controlada).
- $w(t)$: punto de ajuste, (set point).
- $u_{(t+k | t)}$: valores futuros de la entrada del proceso, postulados en el tiempo.
- $y_{(t+k | t)}$: valores pronosticados de la salida en función de:
 - Mediciones disponibles en el tiempo t :
 $\{y(t), y(t-1), \dots, u_{(t-1)}, u_{(t-2)}, \dots\}$
 - Futuros valores postulados de la entrada:
 $\{u_{(t | t)}, u_{(t+1 | t)}, \dots\}$

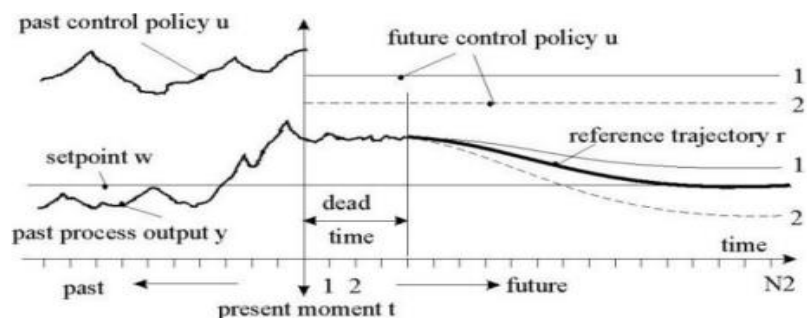


Figura 12. Principio del MBPC.

En cada instancia de tiempo actual t , la salida del proceso $y_{(t+k)}$, es pronosticado a lo largo del horizonte del tiempo en $k=1 \dots N_2$. Los valores pronosticados y son indicados en Y evaluado en $(t+k)$ dado t expresado en " $y_{(t+k | t)}$ " y el valor N_2 es denominado el horizonte de predicción. La predicción se realiza mediante un modelo del proceso, su pronóstico depende de entradas y salidas pasadas, pero también en el escenario del control futuro $\{u_{(t+k | t)}, k = 0 \dots N_2 - 1\}$

Una referencia en su trayectoria $\{r_{(t+k | t)}, k = 0 \dots N_2\}$, comenzando en $r_{(t | t)} = y_{(t)}$ evolucionando hacia el punto de ajuste w , es

definido como el horizonte de predicción, describiendo como se desea guiar la salida del proceso desde su actual $y_{(t)}$. Para su punto de ajuste w ; en caso de que el proceso tenga un retraso de tiempo (tiempo muerto), se recomienda comenzar la trayectoria de referencia después del tiempo de demora, como se muestra en la figura 12.

El vector de control $\{u_{(t+k|t)}, k = 0 \dots (N_2 - 1)\}$ de una función específica de la minimización de costo, se calcula dependiendo del error del control predictivo $\{[r_{(t+k|t)} - y_{(t+k|t)}], k = 0 \dots (N_2 - 1)\}$. Además, en la mayoría de los métodos hay algunas estructuras sobre la ley futura de control $\{u_{(t+k|t)}, k = 0 \dots (N_2 - 1)\}$ y podría también haber restricciones en las variables del proceso.

En el primer elemento $u_{(t|t)}$ del vector de control $\{u_{(t+k|t)}, k = 0 \dots (N_2 - 1)\}$ es aplicado en el proceso real. El vector puede ser olvidado porque en el instante de muestreo todas las secuencias de tiempo se desplazan, se obtiene una nueva medida de salida $y_{(t+1)}$ y el conjunto se repite; esto lleva a una nueva entrada de control $u_{(t+1|t+1)}$, que es generalmente diferente de la u calculada previamente $(t + 1 | t)$; este enfoque es llamado principio del horizonte en retroceso.

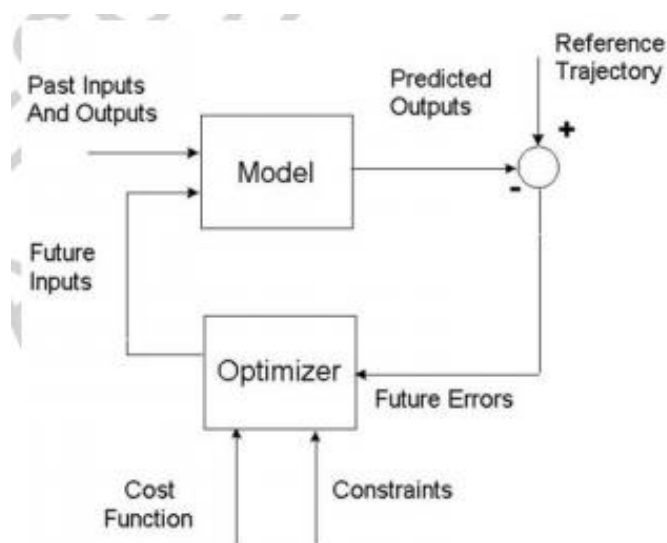


Figura 13. Diagrama de bloques de la estrategia del MBPC.

3.1.2.2. Diseño de las funciones

Nuestro diseño de algoritmo de control está compuesto por dos funciones, valores de entrada pasada $x_{(t+k|t)}$ y predicción de una perturbación $n_{(t+k|t)}$.

3.1.2.2.1. Algoritmo de predicción $x_{(t+k|t)}$

En cada instante de muestreo t , la recursividad se inicia con $k = 0$ y $x(t|t)$ se calcula utilizando el vector de entrada del modelo $[x_{(t-1)}, x_{(t-2)}, \dots, u_{(t-1)}, u_{(t-2)}, \dots]$, el cual contiene valores del pasado que son conocidos en el tiempo t (disponible en la base de datos de la computadora). Observe $x_{(t)} = x_{(t|t)}$ que este valor debe guardarse en la base de datos para su posterior utilización en los últimos instantes de tiempo. Luego para $k = 1$, la x calculada previamente $(t|t)$, se usa en la entrada del modelo para calcular $x_{(t+1|t)}$.

El modelo paralelo (también llamado modelo independiente) solo se puede usar para procesos estables. El modelo serie paralelo (también llamado modelo realineado) incorpora el uso también

para procesos inestables, tal como se muestra en la figura a continuación.

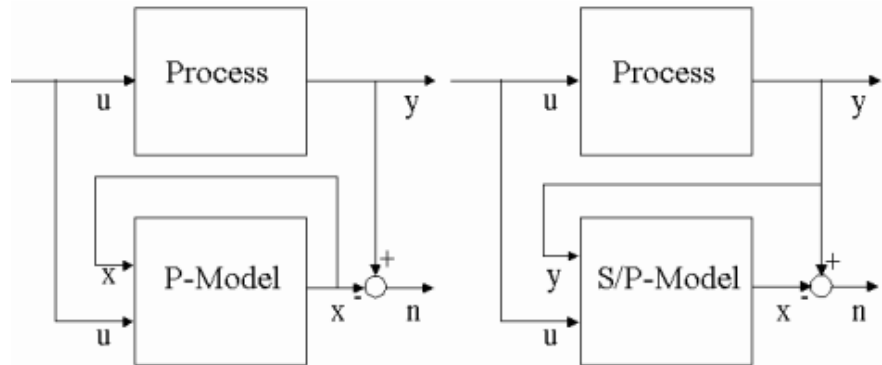


Figura 14. Modelo de implementación serie y serie-paralelo.

3.1.2.2. Algoritmo de predicción $n_{(t+k|t)}$

En el tiempo t , la función $x_{(t)}$ usando las lecturas del proceso de salida $y_{(t)}$ le es posible calcular el valor actual de la perturbación $n_{(t)}$ con el modelo de proceso genérico. Observe que los valores anteriores $n_{(t-1)}, n_{(t-2)}$, también están disponibles en la base de datos como han sido calculados en el instante anterior.

3.1.3. Estructura de la programación del EPSAC en MATLAB

Este tipo de algoritmo trabaja en conjunto y simultáneamente con simulink ya que necesita tomas de lectura de datos para poder ingresarlos en la función de costo donde se determinará la acción de control correspondiente.

Una vez que se obtuvo la función de transferencia y el bosquejo de las funciones de minimización de costos, se procede a realizar la estructura del controlador que está basada en la lógica de control de Robbin De Keyser.

3.1.3.1. Declaración de variables

Se declaran las variables que ayudarán a llamar e inicializar el programa de simulink las cuales son: tiempo de muestreo $T_{(s)}$, cuantos datos de lectura se tomarán $N_{(s)}$, matriz DATA $data = []$ que almacena la información tanto de entrada deseada como de salida del sistema, acción de control, simulación de una perturbación virtual; el vector de estados $State = []$ que representa cuantos estados tiene la planta con la que se va a trabajar y todas las otras constantes necesarias que se aplica para sintonizar el controlador. A continuación, se mostrará la declaración de variables realizada para nuestro proyecto.

```
clear all
At=12.5; Kt=0.7; Kv=0; Kp1=100; Kp2=0.5; Kp3=0.5;
Kp=0.5; val_fin=[0 0];% datos para sintonización
del controlador.
%Ts=0.3; Ns=100; data=[];%nyquist
Ts=0.227; % tiempo de muestreo del simulink
Ns=100; %tamaño de lectura de datos del simulink
data=[];%nyquist información de las variables wt,
yt, ut, kv.
State=[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];
%%matriz de estados de la planta real
u=[0 4];%%condiciones iniciales de la acción de
control
[T,State]=sim('control_v2',Ts,simset('InitialState'
,State(size(State,1),:))); %% comando utilizado
para llamar e inicializar el programa "control_v2"
en simulink.
```

3.1.3.2. Función de transferencia

Se llama al archivo que contiene el modelo matemático del proceso y se lo declara como una función de transferencia en tiempo continuo. Es importante indicar que el controlador EPSAC solo trabaja con la transformada discreta del modelo de la planta debido a que el controlador envía y recibe datos en un determinado tiempo de muestreo $T_{(s)}$, luego se obtiene su numerador y denominador de la

función en tiempo discreto. A continuación, se describirá el código utilizado para obtener la función de transferencia.

```
%%función de transferencia
%%%VARIABLES DE LA ECUACION DE LA FT
Tea=10; %(TEA).
Vector=185; %Dimensión de la matriz Epsac.
EPSACmem=zeros(3,Vector);%Declaración de los datos en
ceros de la matriz controladora.
N2=Vector-Tea;%Variable de lectura.
load('P2IZUV1.mat')% Llamada del archivo que contiene la
FT en tiempo continuo del sistema.
kp=P2IZU.Kp;% Constante de ganancia.
Tz=P2IZU.Tz;% Proceso del cero.
Tw=P2IZU.Tw; % transformada inversa de la frecuencia
natural.
Zeta=P2IZU.Zeta; % coeficiente de amortiguamiento.
s=tf('s'); % declaración de la variable "s" como una
variable de una función.
G= ((1+Tz*s)/(s*(1+2*Zeta*Tw*s)*(Tw*s)^2))*kp %% Función
de transferencia en el tiempo continuo.
num=G.num{1,1}; %numerador de la función de transferencia
en el dominio S
den=G.den{1,1}; %denominador de la función de
transferencia en el dominio S
[numd,demd]=c2dm(num,den,0.09,'zoh')%% Conversión de la
función en tiempo a continuo a tiempo discreto
Gz= tf(numd,demd,0.1)%%transformada discreta de la modelo
matemático de la planta.
numerador_d=Gz.num{1,1}%numerador de la función de
transferencia en el dominio Z.
denominador_d=Gz.den{1,1}%denominador de la función de
transferencia en el dominio Z.
```

3.1.3.3. Condiciones iniciales

Código utilizado para inicializar el valor de la acción de control

```
%Condiciones Inicial
YValvue_inicial= 0;%Se declara la variable que va
contener la matriz de acción de control.
u=[YValvue_inicial YValvue_inicial]; %Inicialización de
la variable de control.
```

3.1.3.4. Periodo de muestreo

Para realizar un periodo de muestreo es necesario saber cuántas muestras vamos a requerir, esto nos lo da nuestra variable $N_{(s)}$ que implementada con un lazo "for" ejecutará la función "function [Ut, EPSACmem]=EPSAC_RDK(Wt, Yt, Ut, N2, numerador_d, denomina

dor_d, EPSACmem, Tea) $N_{(s)}$ veces. Dentro del *bucle for* se realizan los cálculos de las variables, acción de control, entradas pasadas y las perturbaciones, así como su almacenamiento en la base de memoria. Además, se mostrará una gráfica en tiempo real de la salida vs la entrada.

```
for Tm=1:Ns, %Inicio de la toma de datos y cálculo de los
datos del EPSAC.
Yt=valvula(end); %Toma en último dato que se generó en la
salida del sistema.
Wt=0.7;if Tm>50, Wt=1.0; end; if Tm>100, Wt=1.0; end;
%Ingreso del valor deseado.
Kv=0; if Tm>2*75, Kv=0.5; end; if Tm>5*75, Kv=0; end; %
Declaración de una perturbación simulada.
if Ut<0, Ut= 0; end; if Ut>2, Ut=1.3; end; %límites
superior e inferior de la acción de control.
[Ut, EPSACmem]=EPSAC_RDK(Wt, Yt, Ut, N2, numerador_d, denominad
or_d, EPSACmem, Tea); % Llamada de la función del EPSAC.
data=[data; Wt Yt Ut Kv]; % Escritura de los datos
generados por el controlador, variable del valor deseado,
salida del sistema, preturbación.
BufOld=Buf; Buf=[Wt Yt]; % Variable a graficar.
plot([Tm Tm-1], [Buf; BufOld], 'Erasemode', 'none');
drawnow; %comando para realizar la animación de la
grafica en tiempo real.
u=[0 Ut]; % Matriz de la acción de control.
[T, State]=sim('control_v2', Ts, simset('InitialState', State
(size(State,1), :))); % Llamado del programa del simulink.
Ut=interp1([06 18], [Ut5 Ut21], Yt, 'linear', 'extrap'); % [04
18 12 valores de respuesta de la salida de diferetens U]
end; % fin de la instrucción.
```

3.1.3.5. Función del controlador EPSAC

En este algoritmo se encuentran dos subfunciones:

- *Entradas pasadas*, dependen de la dinámica de la función de transferencia en tiempo discreto.
- *Perturbaciones*, dependen de la diferencia de la salida con la entrada en el tiempo inicial t .

El valor de la acción de control futura se obtiene de la suma del valor presente más la pasada.

$$U_t = U(TEA) + \Delta U$$

El valor de acción pasada se lo obtiene de la siguiente forma:

$$\Delta U = \frac{G' * (Ref - Ybase)}{(G' * G)}$$

Donde,

G Función de transferencia en tiempo discreto.

Ref Matriz de los valores deseados.

$Ybase$ Resultado de la suma de las matrices inversas de las entradas y perturbaciones pasadas.

A continuación, se muestra el código de la función

```
function
[Ut, EPSACmem]=EPSAC_RDK(Wt, Yt, Ut, N2, numerador_d, denominador_d, EPSACmem, Tea); %discretizado
TEA=Tea; X=EPSACmem(1, :); U=EPSACmem(2, :); N=EPSACmem(3, :);
U(TEA-1)=Ut; %put the REAL u(t-1) at U(TEA-1)
Na=length(denominator_d)-1; Nb=length(numerador_d)-1;
U(TEA:end)=U(TEA-1); %put(t+k|t)=Ubase=u(t-1) for
k=0...N2-1
for k=0:N2,
X(TEA+k)=-denominator_d(2:Na+1)*X(TEA+k-1:-1:TEA+k-
Na)'+numerador_d(2:Nb+1)*U(TEA+k-1:-1:TEA+k-Nb)';
end;
N(TEA)=Yt-X(TEA) %compute n(t)=y(t)-x(t)
for k=1:N2,
N(TEA+k)=1.8*N(TEA+k-1)-0.8*N(TEA+k-2);
end;
Ybase=X(TEA+1:TEA+N2)'+N(TEA+1:TEA+N2)'; %compute
y(t+k|t)=x(t+k|t), k=1...N2
Ref=Wt*ones(N2, 1);
StepResponse=filter(numerador_d, denominator_d, ones(N2+1, 1)
); G=StepResponse(2:end);
DeltaU=G'*(Ref-Ybase)/(G'*G); %new control input
Ut=U(TEA)+DeltaU;
X=[X(2:end) 0]; U=[U(2:end) 0]; N=[N(2:end) 0]; %shift EPSAC
database
EPSACmem(1, :)=X; EPSACmem(2, :)=U; EPSACmem(3, :)=N;
```

3.1.4. Diseño de interfaz en simulink

Se declaró el OPC para que permita recibir y enviar datos del CU 1214C al simulink, luego se procedió a multiplicar la ganancia y se sumó una constante que ayuda a cambiar la acción de control proveniente de la función del EPSAC.

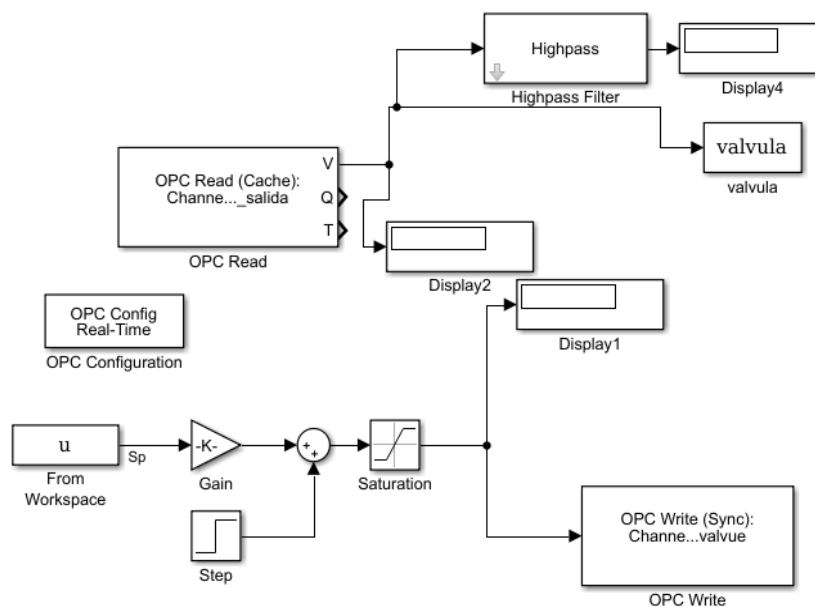


Figura 15. Programa en Simulink del control EPSAC.

3.2. PID FUZZY

3.2.1. Diseño del controlador PID FUZZY

Para el diseño del algoritmo de control del PID difuso que regule el caudal ya establecido en el inicio de este capítulo, se lo ha dividido en dos etapas:

3.2.1.1. Control Difuso

El diseño del control difuso se lo realizó en Simulink. La programación de los conjuntos difusos, sus reglas de correspondencia y el método de defusificación se lo realizó a través del comando *fuzzy*. Los pasos para el diseño de control fueron los siguientes:

3.2.1.1.1. Conjuntos borrosos

En esta sección se detallan las funciones de pertenencia de los conjuntos borrosos, donde la variable leída es el error y el método AND aplicado a las funciones de membresía es el “producto”. El error puede ser positivo y negativo, eso va a depender de los cambios que se aplique a la entrada.

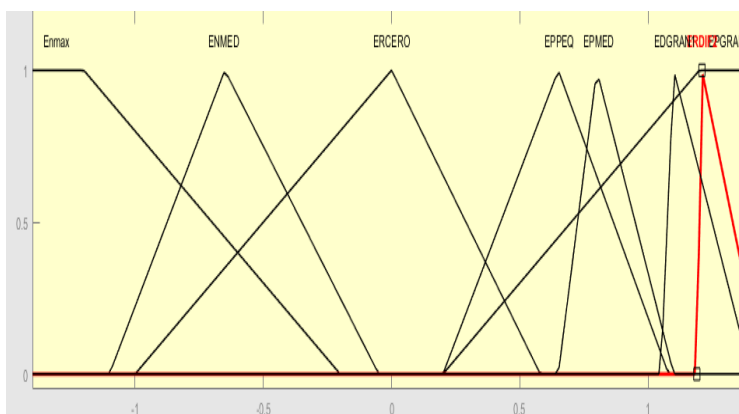


Figura 16. Conjunto borroso de la entrada del controlador Fuzzy.

Donde:

- El eje x representa la magnitud del error.
- El eje y representa la magnitud de la función de pertenencia.
- El primer conjunto borroso “ENMAX” tiene un rango de error: [-2.03 -1.47 -1.2 -0.2] con una forma geométrica trapezoidal.
- El segundo conjunto borroso “ENMED” tiene un rango de error: [-1.1 -0.65 -0.05] con una forma geométrica triangular.
- El tercer conjunto borroso “ERCERO” tiene un rango de error: [-1 0 0.58] con una forma geométrica triangular.
- El cuarto conjunto borroso “EPPEQ” tiene un rango de error: [0.2 0.65 1.08] con una forma geométrica triangular.
- El quinto conjunto borroso “EPMED” tiene un rango de error: [0.65 0.8 1.1] con una forma geométrica triangular.
- El sexto conjunto borroso “EDGRAN1” tiene un rango de error: [1.05 1.1 1.4] con una forma geométrica triangular.
- El séptimo conjunto borroso “ERDIF2” tiene un rango de error: [1.19 1.21 1.45] con una forma geométrica triangular.
- El octavo conjunto borroso “EPGRAN” tiene un rango de error: [0.2 1.2 1.47 2.2] con una forma geométrica trapezoidal.

3.2.1.1.2. Reglas del control

Se describen varias reglas de tipo IF-THEN difusas, las cuales tienen la nomenclatura: if X es A then Y es B. donde A (antecedente) y B (consecuencia) son variables lingüísticas definidas por los conjuntos borrosos en el universo de X y Y. A continuación, se detallan las reglas de correspondencia que se utilizaron para el diseño de este controlador.

- If(Input is EPPEQ) then (output is mf4) (1)
- If(Input is ERCERO) then (output is mf5) (1)
- If(Input is Enmax) then (output is mf6) (1)
- If(Input is ENMED) then (output is mf5) (1)
- If(Input is ENMED) then (output is mf2) (1)
- If(Input is ERDIF2) then (output is mf7) (1)
- If(Input is EDGRAN1) then (output is mf3) (1)
- If(Input is EPMED) then (output is mf8) (1)

3.2.1.1.3. Salida o defusificación

Para la salida de nuestro controlador se escogió el modelo de Takagi-Sugeno [3], el cual realiza la suma ponderada de cada regla cuando existe una intersección entre los conjuntos borrosos de la entrada. Para este modelo, la salida debe ser un número exacto. A continuación, se mostrará la salida que se parametrizó para este controlador.

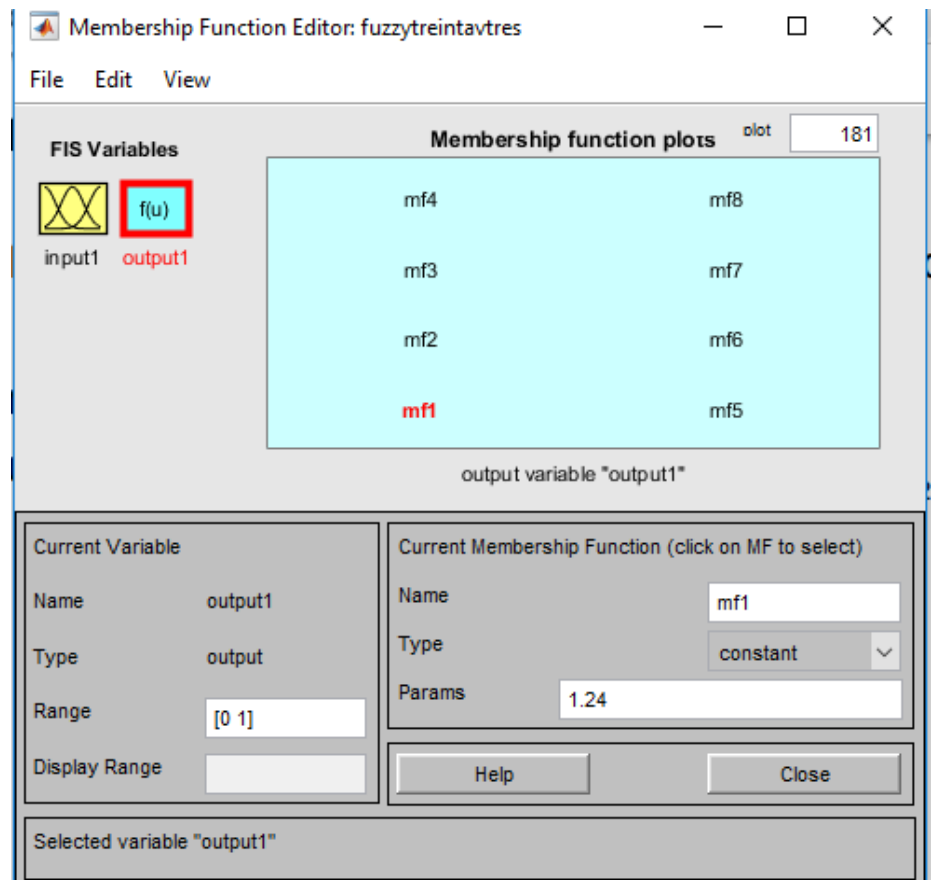


Figura 17. Parametrización de la salida del controlador Fuzzy.

Donde:

- La salida mf1 esta parametrizada en 1.24
- La salida mf2 esta parametrizada en 0.25
- La salida mf3 esta parametrizada en -0.15
- La salida mf4 esta parametrizada en 0.1
- La salida mf5 esta parametrizada en 0.35
- La salida mf6 esta parametrizada en 0.35
- La salida mf7 esta parametrizada en 0.005
- La salida mf8 esta parametrizada en 0.05

3.2.1.2. PID de Simulink

Para poder ganar dinámica y velocidad en la salida del control fuzzy se incorporó un bloque PID-Controller, el cual se lo parametrizó con los siguientes datos:

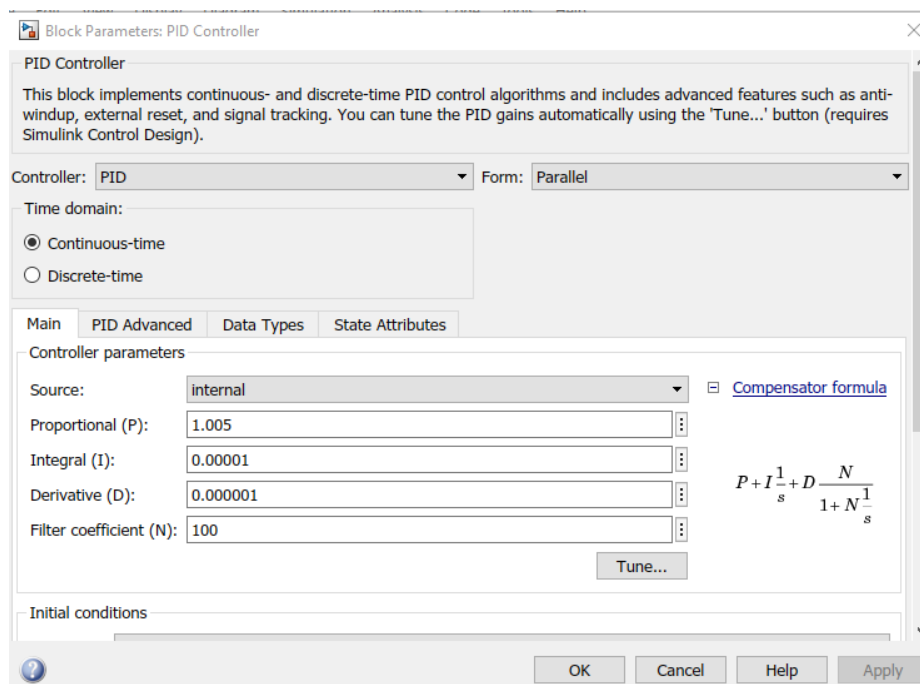


Figura 18. Ventana de parametrización del controlador PID Controller

Donde:

- Está configurado en tiempo continuo.
- La acción de las constantes es en forma paralela.
- La variable proporcional (P), esta seteada en 1.005
- La variable integral (I), esta seteada en 0.00001
- La variable derivativa (D), esta seteada en 0.000001
- Filtro del coeficiente (N) es 100.

3.2.2. Diseño de interfaz en Simulink

Se configuró un OPC para poder recibir y enviar datos del PLC S7-1200 al Simulink

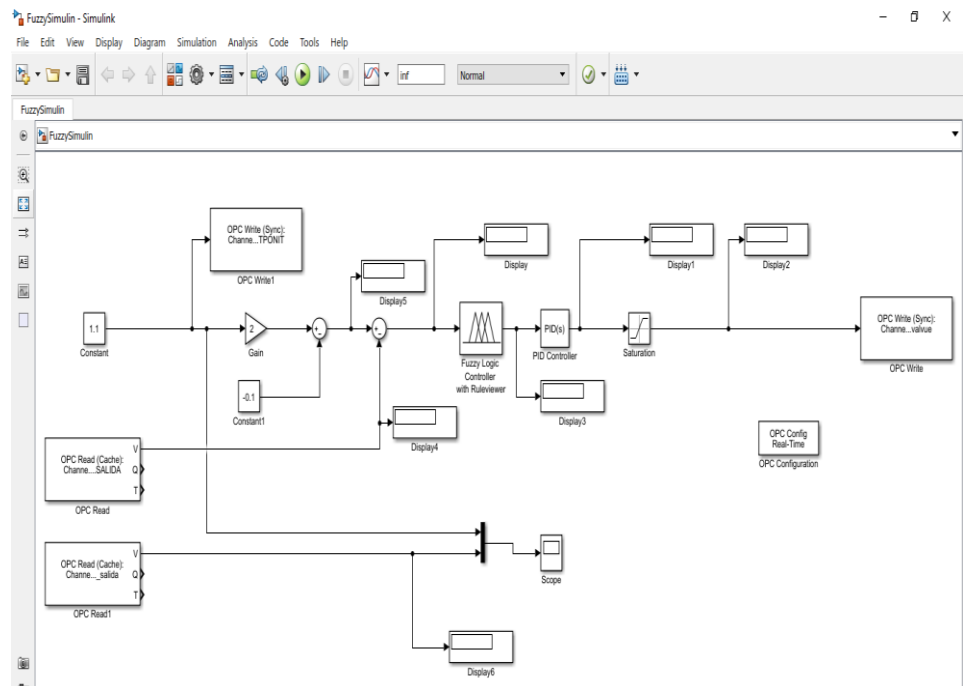


Figura 19. Programación del controlador PID FUZZY

Nota: Durante el diseño de los límites de las funciones de membresía de la variable leída “error”, ocasionó que se le incorpore la multiplicación y suma de constantes para que el usuario pueda generar los cambios del caudal deseado en el rango comprendido desde $[0.6 \text{ a } 1.0 \frac{m^3}{h}]$.

3.3. PID_COMPACT

3.3.1. Introducción

Se procede a diseñar un control PID que regule un caudal comprendido en el siguiente rango $[0.6 \text{ a } 1.0 \frac{m^3}{h}]$, que está basado en el desarrollo de un objeto tecnológico del TIA Portal llamado PID_COMPACT. Dicha instrucción brinda la implementación de un regulador PID que incluye optimización para actuadores de acción proporcional. Tiene una interfaz de sintonización donde dispone los siguientes modos de operación.

- Inactivo.
- Optimización fina.
- Optimización inicial.
- Modo Automático.
- Modo Manual.
- Lectura del estado actual del bloque con lectura de errores.

El algoritmo de control es un regulador PIDT1 y ponderación de las acciones P y D, el cual funciona con las siguientes fórmulas [4]:

$$y = K_p \left[(b \cdot w - x) + \frac{1}{T_i \cdot s} (w - x) + \frac{T_D \cdot s}{a \cdot T_D \cdot s + 1} (c \cdot w - x) \right]$$

Donde:

- y Valor de salida del algoritmo PID
- Kp Ganancia proporcional.
- s Operador laplaciano.
- b Ponderación de la acción P.
- w Consigna.
- x Valor real.
- Ti Tiempo de integración.
- Td Tiempo de derivativo.
- a Coeficiente para el retardo de la acción derivativa.
- c Ponderación de la acción D.

Diagrama de bloques del controlador

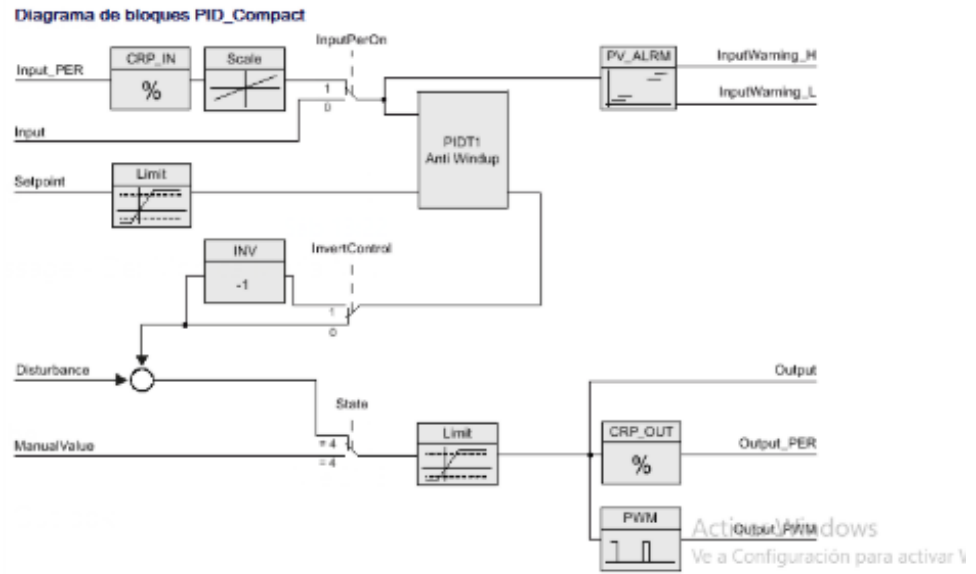


Figura 20. Descripción PID_COMPACT

3.3.2. Diseño del algoritmo de control

A continuación, se detalla el desarrollo del diseño del controlador:

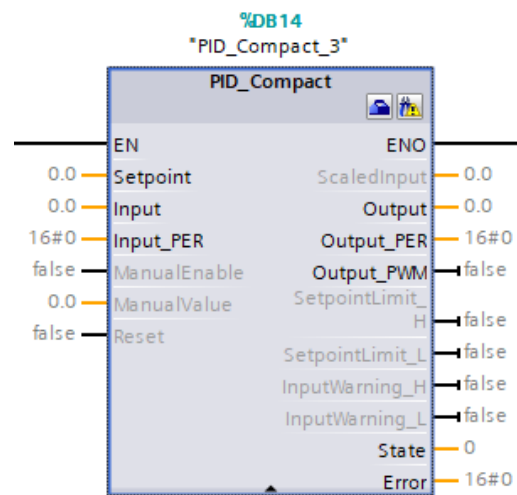


Figura 21. Bloque de la instrucción tecnológica del PID_COMPACT del TIA Portal

Se declaran las variables de entrada y salida del bloque más importante a utilizar, entre las cuales tenemos: set point, input, reset, output y se

presiona el botón con el icono de una casa para parametrizar el bloque al rango con el cual se desea trabajar.

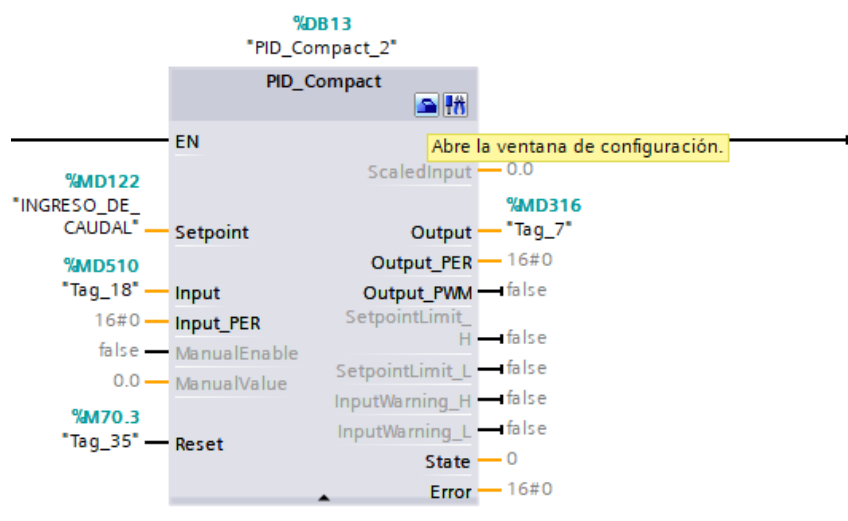


Figura 22. Asignación de las variables de entradas y salida

La primera parte se trata de indicar el tipo de variable física que se desea controlar e indicar que tipo de entrada es la que va a utilizar en la lógica interna del controlador.

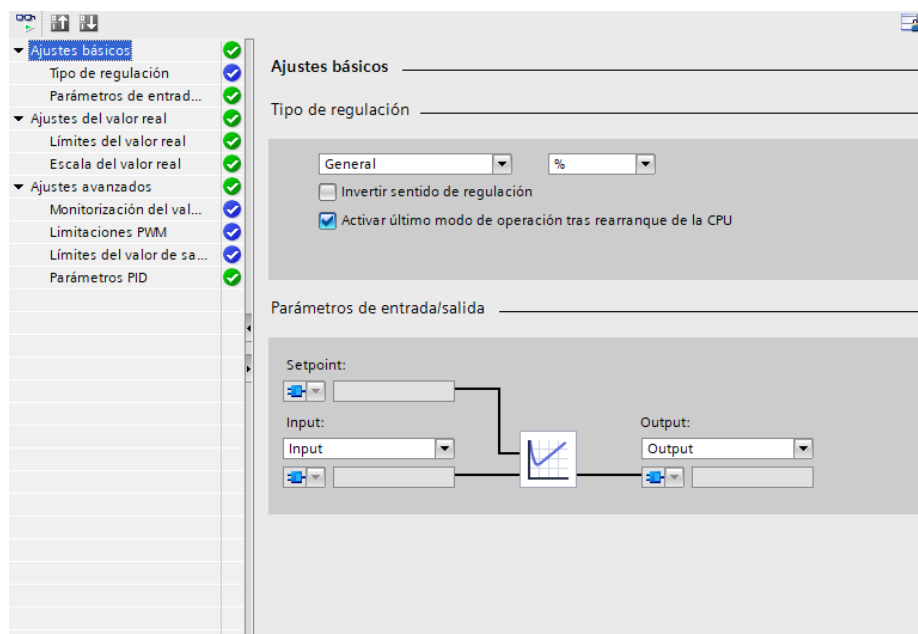


Figura 23. Parametrización de los ajustes básicos del PID_COMPACT.

Se declaran los límites del valor de la salida para la acción de control.

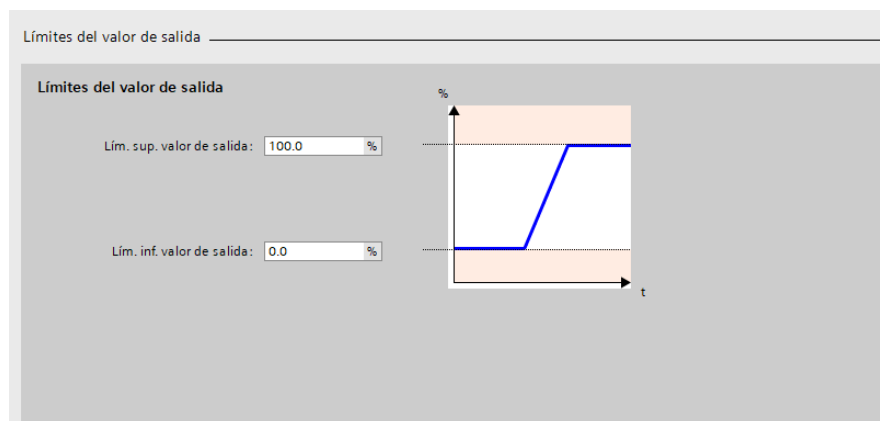


Figura 24. Parametrización de los límites del valor de salida.

Después de parametrizar las variables, se procede a realizar una sintonización inicial para que realice un barrido y luego una sintonización fina para calcular las constantes del valor para el controlador PID de este bloque la cual se la muestra a continuación:

Parámetros PID

Activar entrada manual

Ganancia proporcional: 471.9977

Tiempo de integración: 7.354435 s

Tiempo derivativo: 1.287026 s

Coefficiente retardo derivativo: 0.1

Ponderación de la acción P: 0.8

Ponderación de la acción D: 0.0

Tiempo muestreo algoritmo PID: 9.999996E-2 s

Regla para la optimización

Estructura del regulador: PID

Figura 25. Valores del PID adquiridos por la sintonización del bloque.

3.3.3. Diseño de interfaz de usuario

Se realizó una pantalla donde se puede ingresar el valor deseado del set point, el reset del boque del PID cuando este origine una falla y la lectura de las variables de entrada y salida

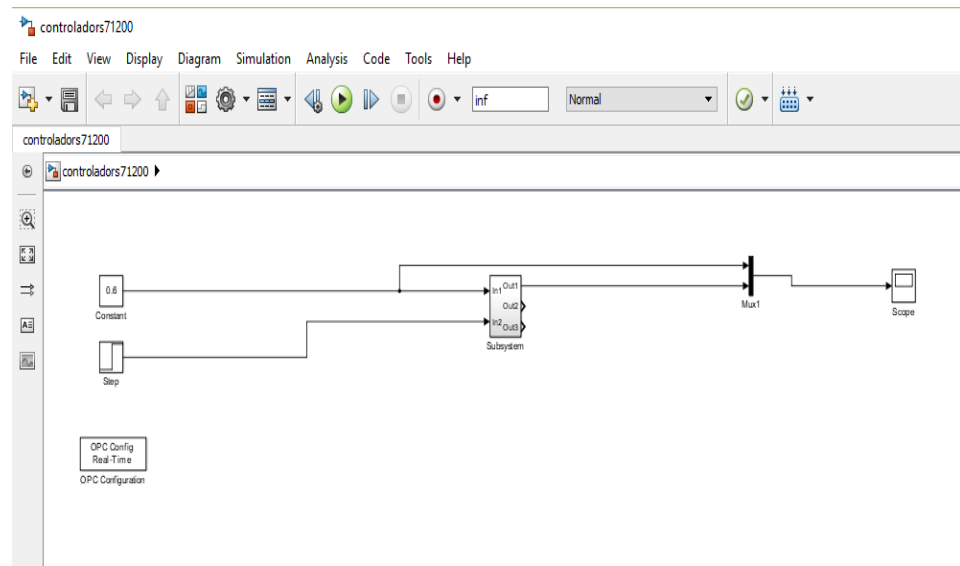


Figura 26. Programación del controlador PID_COMPACT

3.4. OPC SERVER

3.4.1. Introducción

El servidor OPC es una aplicación virtual que ejecuta una o varias especificaciones establecidas por la OPC Foundation. El servidor funciona como una interfaz de comunicación de datos punto a punto o varias funetes de datos a la vez. Los protocolos de datos nativos típicamente usados son los PLC's, DCS's, Básculas, Módulos I/O, Controladores, etc. y por otro lado con clientes OPC, tales como: SCADA's, HMI's, generadores de informes, gráficos, aplicaciones de cálculo, etc. donde la arquitectura de red declara al cliente OPC como maestro y al servidor OPS como esclavo. El sentido de comunicación es bidireccional.

3.4.2. Diseño del Servidor OPC

El software que se utilizó para la construcción del servidor OPC fue el KEPserverEX 6 en el cual se declararon todas las variables necesarias para la interacción de la comunicación bidireccional entre el PLC S7 1200 y MATLAB.

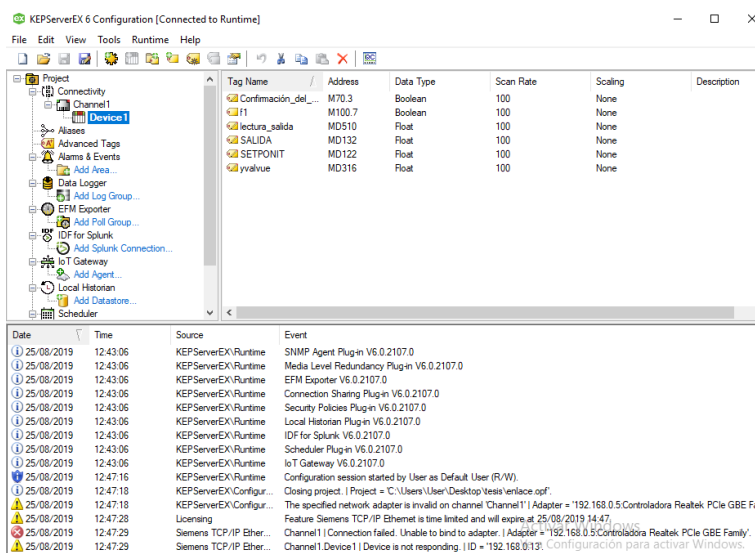


Figura 27. Declaración de las variables en el OPC Server.

CAPITULO 4

4. IMPLEMENTACION DE LOS CONTROLADORES

La Planta reguladora de caudal que se utilizó un módulo didáctico que regula el caudal, se encuentra en el laboratorio de instrumentación industrial, marca Gunt, el cual está conformado por los siguientes equipos:

- **Bastidor RT450**

La estructura de la Planta Reguladora de Caudal denominada Gunt RT450. Aquí se encuentra un reservorio en acero inoxidable fino con un almacenamiento máximo de 75 litros; en la descarga del tanque se encuentra acoplada una bomba centrífuga de agua con las siguientes características:

- Potencia: 370W.
- Caudal máximo: 5.4 m³/h.
- Alimentación eléctrica: 120Vac.
- Altura máxima de elevación: 20m.



Figura 28. Estructura del sistema regulador de caudal.

- **Válvula de control**

El bastidor RT450 tiene instalado una válvula con su respectivo posicionador electro-neumático necesaria para mantener el control de nivel o caudal en un recipiente.

La característica de este posicionador es que necesita aire para que abra a cierre la válvula dependiendo de la señal eléctrica que se le envíe que puede ser de 4 a 20 mA. La posición de falla de cerrada. A continuación, algunos datos técnicos de la válvula y posicionador.

- DN 15
- PN 16
- Valor K_{vs} : 0.4
- Curva característica: lineal
- Superficie de membrana: 120cm²
- Carrera: máx. 15mm
- Rango de señal nominal: 0.2 a 1bar
- Señal de entrada al posicionador: 4 a 20mA
- Adaptador con acoplamiento rápido: 6mm

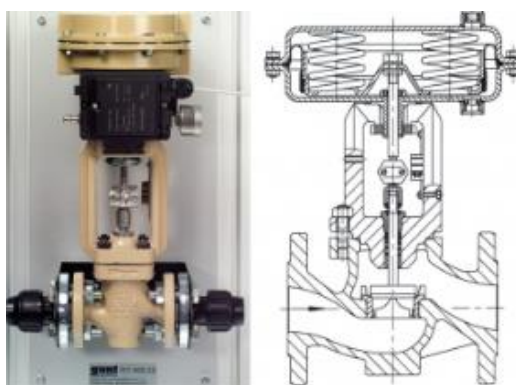


Figura 29. Válvula controladora de caudal.

- **Rotámetro de área variable**

Este rotámetro es de tipo plástico que nos va a permitir visualizar directamente todos los cambios que se producen en el flujo causado por cambios en el punto de operación o cualquier perturbación presente en el sistema y va montado de forma vertical en el bastidor. Las características son las siguientes:

- Rango de lectura: 0 a 2.5 m³/h



Figura 30. Rotámetro.

- **Sensor de caudal**

Es un instrumento de lectura compacto compuesto por un transductor de medida y un sensor y va montado de forma vertical en el bastidor. Las características técnicas del sensor son:

- Principio de medición: electromagnético.
- Rango de lectura: 0 a 2.5 m³/h.
- Voltaje de alimentación: 24Vdc.

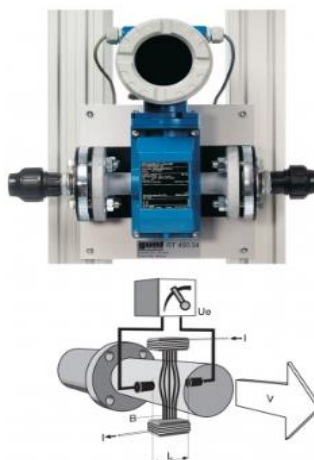


Figura 31. Sensor de caudal con salida analógica.

- **Controlador S7 1200**

Pertenece a la gama de los controladores de gama baja, con un diseño escalable y compacto, incorporando instrucciones avanzadas y tecnológicas que permiten el control de una amplia gama de procesos industriales. La CPU utilizada en este proyecto es la 1214C y su hardware es el siguiente:

Puerto de comunicación Profinet, el cual es utilizado para:

- Crear una topología de red Maestro-Eslavo.
- Adquisición de datos.
- Realizar carga y descarga de la programación del dispositivo.
- Supervisión online del dispositivo.
- Controlar periféricas descentralizadas.

Entradas y salidas digitales:

- Entradas digitales:
Clase: sumidero. Tensión nominal de trabajo 24Vdc a 4mA.
- Salidas digitales:
Salida lente es de tipo relé y tiene un rango de trabajo de 5 a 30 Vdc o 5 a 250Vac, con una corriente máxima de 0.5A.

Salida rápida es tipo transistor y tiene un rango de trabajo de 20.4 a 28.8 Vdc con una corriente máxima de 0.5A.

Entradas analógicas:

- Tiene dos entradas de tipo unipolares (tensión), operando en un rango de 0 a 10Vdc con una resolución de 10 bits.

A continuación, se detalla el rendimiento de las velocidades de ejecución más importante que tiene una CPU 1214C:

Rendimiento	Unidad de Procesamiento Central
Velocidad de ejecución booleana	0.1 micro segundos por instrucción
Velocidad de ejecución por desplazamiento de palabra	12 micro segundos por instrucción
Velocidad de ejecución de la instrucción matemática.	18 micro segundos por instrucción

Tabla 1. Velocidad de muestreo de datos de la CU1214C.

La CPU tiene un módulo de entradas y salidas analógicas SM 1234 AI/AQ incorporadas a su hardware con las siguientes características:

Entradas analógicas:

- Resolución: 13 bits.
- Alimentación del módulo: 24 Vdc.
- Tipo de señal: ± 10 Vdc, ± 5 Vdc, ± 2.5 Vdc o 4 – 20 mA

Salidas analógicas:

- Resolución: 14 bits.
- Tipo de señal: ± 10 Vdc o 4 – 20 mA

4.1. Controlador EPSAC

Ahora se implementará un control EPSAC en dicho proceso. Para verificar el desempeño del controlador se realizaron dos pruebas:

- En la primera prueba se excitó la planta con dos entradas de caudal que inicia con 0.6 hasta 0.8 m³/h. obteniendo la siguiente respuesta.

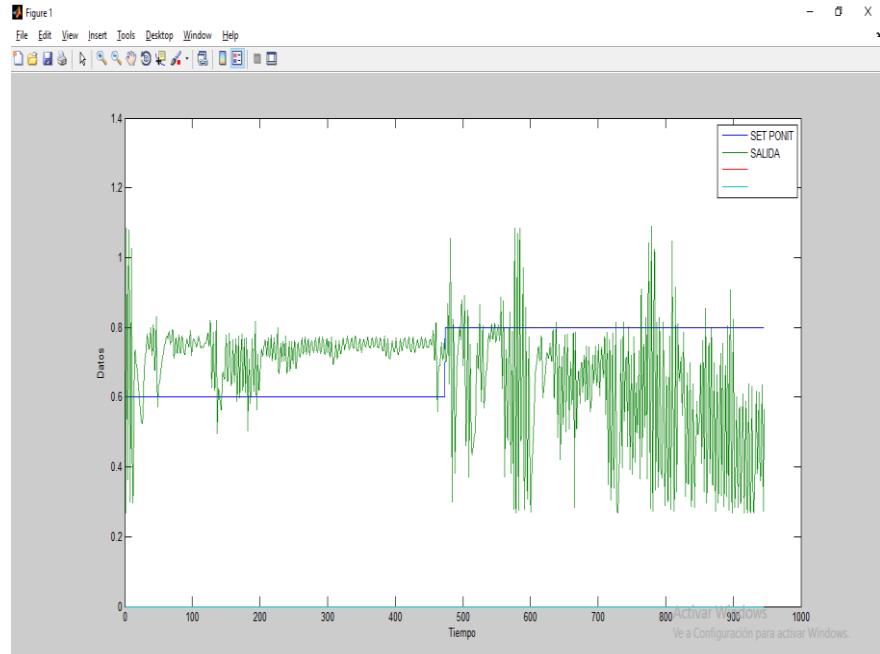


Figura 32. Respuesta 1 del controlador EPSAC.

Como se puede observar en la gráfica, el EPSAC no puede controlar la planta debido a que existe retardo en los tiempos de muestreo y de adquisición de datos.

- En la segunda prueba se excitó la planta con dos entradas de caudal que inicia con 0.7 hasta 0.9 m³/h. obteniendo la siguiente respuesta.

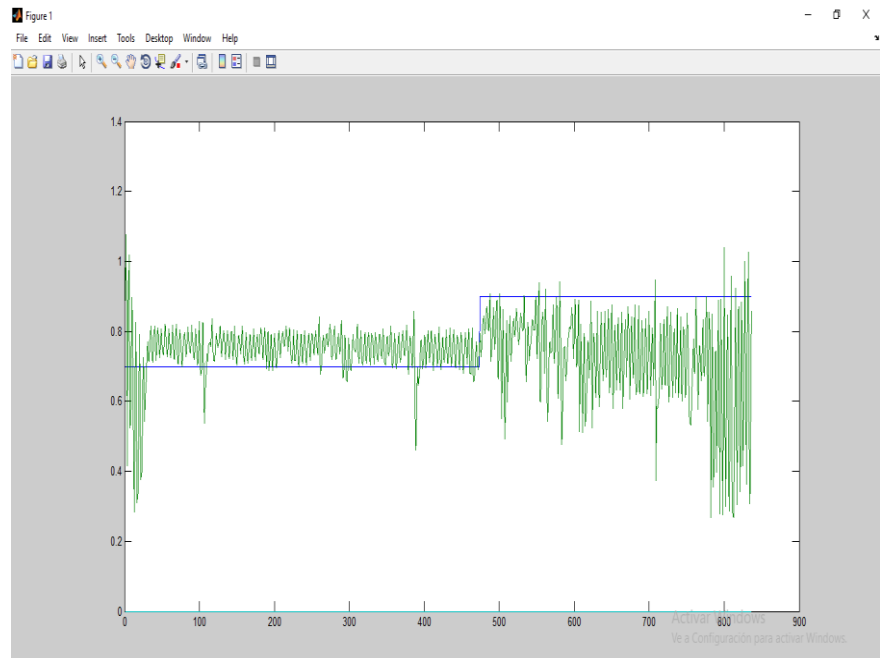


Figura 33. Respuesta 2 del controlador EPSAC.

Como se puede observar en las dos respuestas no se genera un buen control, esto se debe a que existen diferencias en el tiempo de muestreo entre los equipos y el software de control en la adquisición y escritura de los datos en las variables declaradas en el PLC y MATLAB.

El proceso de adquisición de datos inicia desde el censado del líquido que pasa a través del sensor que emite la señal analógica de 4-20 mA que ingresa al módulo de entradas y salidas analógicas del PLC, luego lo convierte en una señal binaria para poder ser transmitida al computador por medio del protocolo de comunicación Profinet. Dicha señal binaria ingresa a un servidor OPC que transmite los datos al Simulink para escribirlos en el Workspace de MATLAB para que sean leídos por la función del EPSAC cada vez que esta sea llamada.

La escritura de datos realiza un proceso inverso que la lectura de datos puesto que empieza desde MATLAB hasta el posicionador electro-neumático de la válvula estranguladora de caudal.

Este proceso de transmisión de datos causa que la señal no sea monitoreada en tiempo continuo sino tiempo discreto y esto causa que la acción de control no sea la adecuada para el instante de tiempo que se adquirió el dato.

4.2. Controlador PID FUZZY

Se implementará un algoritmo de control PID FUZZY. Para observar el desempeño del controlador se realizaron dos pruebas:

- En la primera prueba se excitó la planta con dos entradas de caudal que inicia con 0.6 hasta 0.8 m³/h. obteniendo la siguiente respuesta.

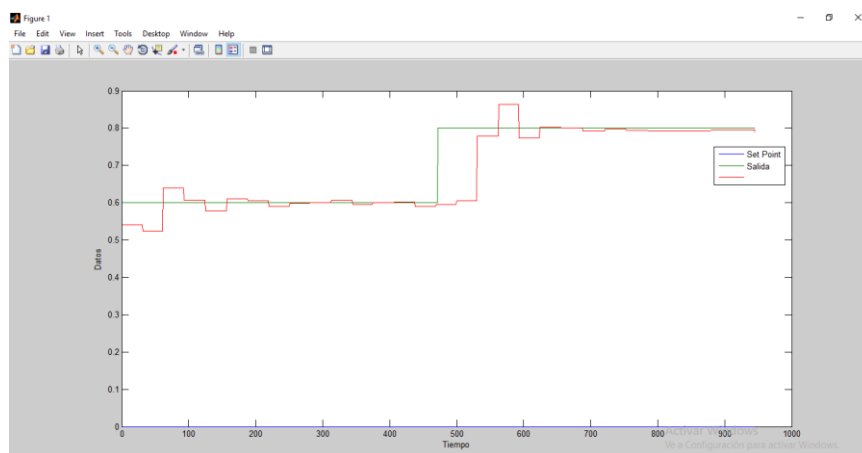


Figura 34. Respuesta 1 del controlador PID FUZZY.

En la presente gráfica se puede analizar que el sistema se estabiliza en 159 segundos con un valor de 0.61 m³/h.; a los 473 segundos se realiza un cambio en el set point pasando a un caudal de 0.8 m³/h. pero no se ve una reacción inmediata sino hasta los 624 segundos donde se estabiliza con un valor de 0.802 m³/h.

- En la segunda prueba se excitó la planta con dos entradas de caudal que inicia con 0.7 hasta 0.9 m³/h. obteniendo la siguiente respuesta.

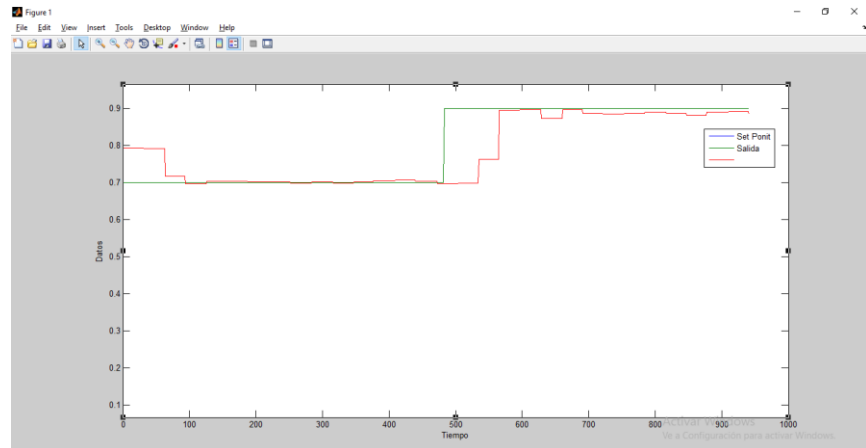


Figura 35. Respuesta 2 del controlador PID FUZZY.

En esta gráfica se puede visualizar que comienza con un valor en la entrada de 0.7 m³/h. estabilizándose en un tiempo de 95 segundos con un valor de 0.69 m³/h. Hicimos un cambio en el set point a los 482 segundos pasando a un caudal de 0.9 m³/h. estabilizándose en 570 segundos con un valor de 0.8965 m³/h. La forma de la respuesta se debe a que se escogió un método de defusificación llamado Takagi-Sugeno [3].

4.3. Controlador PID_COMPACT

Se implementará un objeto tecnológico del TIA Portal denominado PID_COMPACT. Para observar el desempeño del controlador se realizaron dos pruebas.

- En la primera prueba se excitó la planta con dos entradas de caudal que inicia con 0.6 hasta 0.8 m³/h. obteniendo la siguiente respuesta.

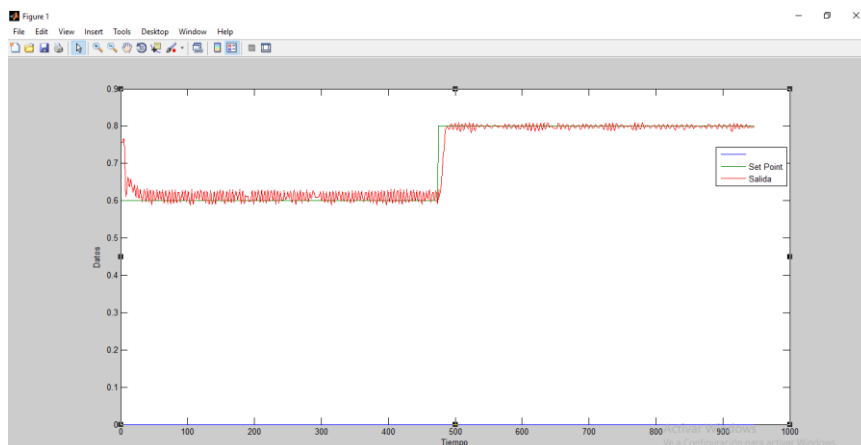


Figura 36. Respuesta 1 del controlador PID_COMPACT

En esta gráfica se puede analizar que la salida del sistema genera oscilaciones constantes con una mayor magnitud respecto a la otra cuando se encuentra con un punto de operación es $0.6 \text{ m}^3/\text{h}$. El tiempo de estabilización cuando está en el primer punto de operación es de 26 segundos con un valor de $0.6233 \text{ m}^3/\text{h}$. y en el segundo punto de operación es de 486 segundos con un valor de $0.7954 \text{ m}^3/\text{h}$.

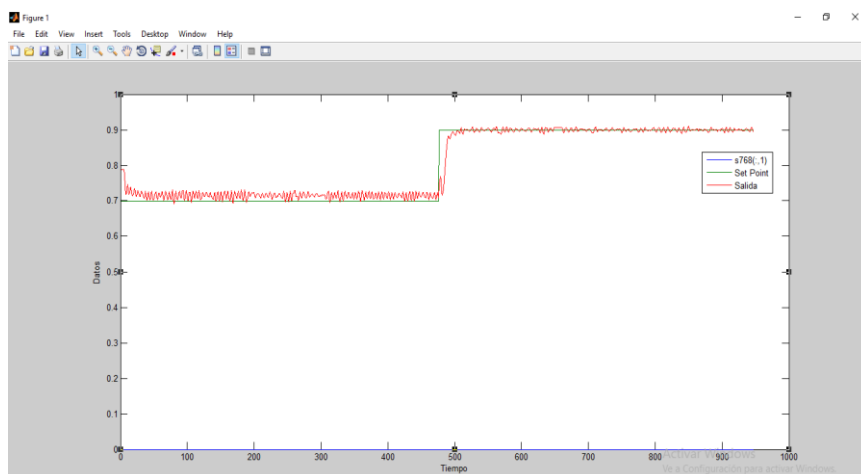


Figura 37. Respuesta 2 del controlador PID_COMPACT

En esta gráfica se puede analizar que la salida tiene el mismo comportamiento que la respuesta mostrada en la figura anterior. El tiempo de estabilización cuando esté en el primer punto de operación es de 36 segundos con un valor de $0.7152 \text{ m}^3/\text{h}$. y en el segundo punto de operación es de 503 segundos con un valor de $0.8981 \text{ m}^3/\text{h}$.

Las perturbaciones de la respuesta de la planta se deben a la naturaleza del algoritmo de control detallada en el capítulo 3 de este documento.

CAPÍTULO 5

5. RESULTADOS

5.1. Respuestas de los controladores

Se muestran las respuestas de los controladores a una entrada escalón con un valor inicial de 0.6 a 0.8 m³/h.

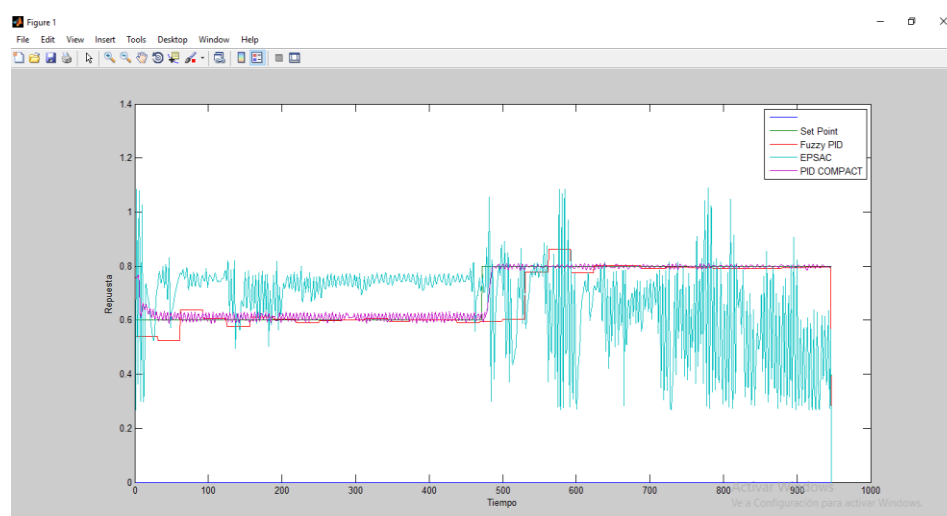


Figura 38. Gráfica 1 de las repuestas de los tres controladores a una entrada paso de 0.6 a 0.8 m³/h

Donde se puede verificar cual controlador tuvo una mejor respuesta respecto al otro. En la siguiente tabla se muestran los resultados obtenidos.

Controlador	Set Point= 0.6			Set Point= 0.8		
	Ts (seg)	Os (m ³ /h.)	e	Ts (seg)	Os (m ³ /h.)	e
EPSAC	N/A	1.1	0.218	N/A	0.3	0.289
PID FUZZY	159	0.6387	0.0063	624	0,8635	-0.0085
PID_COMPACT	26	0,7331	0.0174	486	0,7954	0.0103

Tabla 2. Cuadro comparativo número uno de las repuestas de los controladores.

En la siguiente gráfica se visualizan las respuestas de los controladores a una entrada escalón con un valor inicial de 0.7 a 0.9 m³/h.

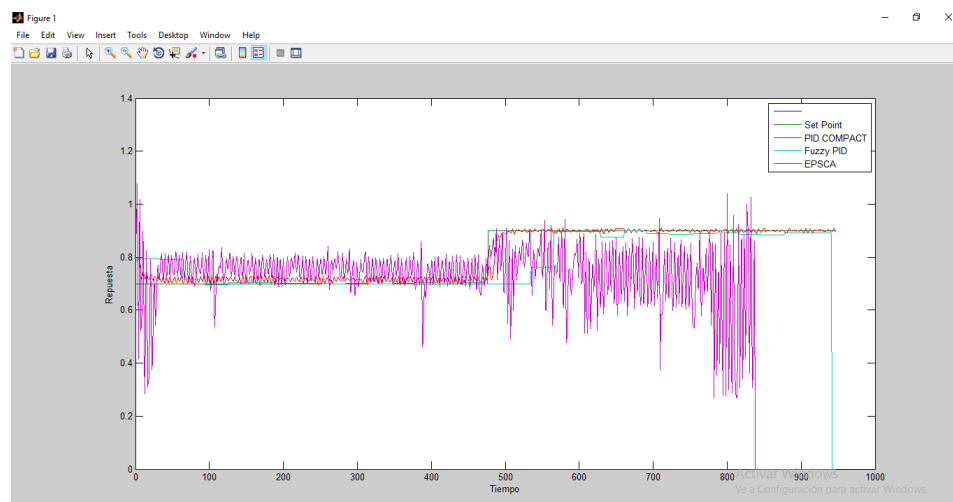


Figura 39. Gráfica 2 de las repuesta de los controladores a una entrada paso de 0.7 a 0.9 m³/h

Donde se puede verificar cual controlador tuvo una mejor respuesta respecto al otro. En la siguiente tabla se muestran los resultados obtenidos.

Controlador	Set Point= 0.7			Set Point= 0.9		
	Ts (seg)	Os (m ³ /h.)	E	Ts (seg)	Os (m ³ /h.)	e
EPSAC	N/A	1.076	-0.137	N/A	-0.23	-0.6329
PID FUZZY	95	N/A	0.0065	570	N/A	-0.0094
PID_COMPACT	36	0,746	0.026	503	0,8981	0.0069

Tabla 3. Cuadro comparativo número dos de las repuestas de los controladores

CONCLUSIONES

Observando las figuras 38 y 39 que representan la repuesta del sistema en los diferentes tipos de controles, se puede concluir lo siguiente:

- El control con la mejor respuesta es el PID FUZZY debido a que no genera muchas oscilaciones en la salida a lo largo del tiempo, pero su estabilización o respuesta a cualquier cambio es tardío si se la compara con el controlador PID_COMPACT.
- El algoritmo PID_COMPACT hace que la salida del sistema sea sensible ante la velocidad del error por lo que ocasiona que la salida sea oscilante y su magnitud es proporcional a la presión de aire en la entrada de la válvula, es decir, si la entrada es mayor igual a 2 PSI, la oscilación en la salida es más pronunciada.
- El controlador EPSAC no realiza un control adecuado porque la salida no es leída en tiempo real debido a que existe un retraso en el valor de la acción de control con la que fue calculado, es decir que cuando es aplicado el valor, su entrada no es la misma con la que se calculó generando un sistema inestable en el tiempo.
- La identificación de la planta obtuvo una aproximación del 92.48% utilizando el comando *ident* de MATLAB.
- El control PID de MATLAB no genera las oscilaciones cuando se encuentra en el punto de trabajo deseado.

RECOMENDACIONES

- Aportes anteriores recomiendan que tiene mejor resultado implementar la acción de control en una bomba de caudal variable que suministra el líquido al proceso.
- Implementar el control EPSAC en una CPU que tenga los recursos necesarios, tales como: procesador de alta velocidad, puerto de comunicación Profinet, memoria RAM de 64 GB, con lo cual mejore significativamente la velocidad de transmisión de datos y que la acción de control sea aplicada en el instante correcto.

- Implementar una red Profinet para obtener una velocidad de transmisión de datos mayor a 100 Mbits/seg full dúplex.

BIBLIOGRAFÍA

- [1] I. De Avance, “NACIONAL Facultad Regional Reconquista”.
- [2] E. D. E. I. Mecánica, “Trabajo de titulación”, 2017.
- [3] “Introducción a la Lógica Difusa Tomás Arredondo Vidal”.
- [4] F. Manual, “No Title”.
- [5] S. Del, “4. sección del controlador difuso”, pp. 35–61.
- [6] Y. Caudal, “Escuela politécnica nacional”, 2017.
- [7] R. D. E. Keyser, “SA NE M SC PL O E – C EO AP LS TE S M SC PL O E – C EO”, vol. XI.
- [8] I. Nascu, I. Nascu, C. M. Ionescu, y R. De Keyser, “Adaptive EPSAC predictive control of the hypnotic component in anesthesia”, *2012 IEEE Int. Conf. Autom. Qual. Testing, Robot. AQTR 2012 - Proc.*, pp. 103–108, 2012.
- [9] R. M. C. De Keyser y A. R. Van Cauwenberghe, “Extended Prediction Self-Adaptive Control”, *IFAC Proc. Vol.*, vol. 18, núm. 5, pp. 1255–1260, 1985.
- [10] a. Draeger, S. Engell, y H. Ranke, “Model predictive control using neural networks”, *Control Syst. IEEE*, vol. 15, núm. 5, pp. 61–66, 1995.
- [11] R. L. Tousain y O. H. Bosgra, “Efficient dynamic optimization for nonlinear model predictive control-application to a high-density poly-ethylene grade change problem”, *Decis. Control. 2000. Proc. 39th IEEE Conf.*, vol. 1, núm. Froisy 1994, pp. 760–765 vol.1, 2000.
- [12] M. A. Duarte-Mermoud y F. Milla, “Estabilizador de Sistemas de Potencia usando Control Predictivo basado en Modelo”, *Rev. Iberoam. Automática e Informática Ind.*, vol. 0, núm. 0, pp. 286–296, 2018.
- [13] C. Nlfsr, P. Control, D. E. U. N. Sistema, I. D. E. T. Sostenible, y C. Sandoval-ruíz, “NEURO-ADAPTIVE MODEL IN VHDL, BASED ON CIRCUITS NLFSR, FOR CONTROL OF AN SMART”, vol. 21, pp. 140–149, 2017.
- [14] F. Liu, J. Ma, X. Feng, y Y. Wang, “Simultaneous integrated design for heat exchanger network and cooling water system”, *Appl. Therm. Eng.*, vol. 128, pp. 1510–1519, 2018.

- [15] C. Tingting, Z. Jili, Z. Tianyi, y X. Li, “ScienceDirect ScienceDirect Application study on a duty ratio fuzzy control method for fan coil units in the building in Chengdu, China”, *Procedia Eng.*, vol. 205, pp. 1323–1330, 2017.
- [16] A. Aguado, A. Gómez, y A. del Pozo, “Controlador predictivo neuro-genético”, *Rev. Iberoam. Automática e Informática Ind. RIAI*, vol. 4, núm. 4, pp. 94–108, 2007.
- [17] “System Identification Toolbox™ User’s Guide R 2019 a How to Contact MathWorks”, 2019.