

# UNIVERSIDAD DE CONCEPCIÓN

FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA



Profesor Patrocinante:

**Dr. Alejandro J. Rojas N.**

Informe de Memoria de Título  
para optar al título de:

**Ingeniero Civil Electrónico**

## Uso de Estrategia de Control Fuzzy para Microcanal

UNIVERSIDAD DE CONCEPCIÓN  
Facultad de Ingeniería  
Departamento de Ingeniería Eléctrica

Profesor Patrocinante:  
Dr. Alejandro J. Rojas N.

# Uso de Estrategia de Control Fuzzy para Microcanal

Ricardo Elías Reyes Narváez

Informe de Memoria de Título  
para optar al Título de

Ingeniero Civil Electrónico

Enero 2016

## Resumen

La implementación de control automático en canales de regadío permite una manipulación eficiente del agua, principalmente en la agricultura. Una de las estrategias de control comúnmente utilizadas es el control PI, el cual, en la mayoría de los casos, permite controlar el proceso de forma segura. Sin embargo, si se tiene conocimiento del proceso, es posible la implementación de estrategias de control más adecuadas, tales como el control de lógica difusa (fuzzy logic), que constituye una estrategia de control de tipo experto. En este trabajo se plantea la utilización de una estrategia de control difusa para la planta microcanal ubicada en el Laboratorio de Control de la Universidad de Concepción.

Para dar solución al problema propuesto, en primer lugar se realizó una identificación del modelo de la planta, para lo cual se realizaron pruebas de tipo escalón a fin de obtener los parámetros más representativos del sistema. Una vez identificado el modelo se procedió a diseñar controladores de lógica difusa para cada piscina y, de forma paralela, controladores PI. Ambos conjuntos de controladores fueron validados mediante simulaciones para luego ser implementados en la planta Microcanal. Por último se comparó el desempeño de los controladores de lógica difusa frente a los controladores PI.

Los resultados muestran que el uso de una estrategia de control difusa permite que el sistema alcance estado estacionario de forma más rápida que usando una estrategia de control PI; se observó una mejora de un 50% en el tiempo de asentamiento en el caso de la piscina 1 y un 25% en el caso de la piscina 2.

# Agradecimientos

En primer lugar a mi madre, Susana Narvez S., por su esfuerzo, sacrificio, fe y apoyo incondicional, sin los cuales no hubiese sido posible terminar este trayecto.

A todos quienes me han ayudado en estos aos, de distintas formas, en distintas circunstancias, sin ninguna obligacion y de manera desinteresada.

A mi profesor patrocinante por su buena voluntad, paciencia y consejos durante el desarrollo de este trabajo.

A todos los profesores y profesionales que cooperaron en la realizacion de este proyecto, tanto con consejos y correcciones como con conocimientos que se vieron plasmados en este trabajo.

A todos mis compaeros con los que he compartido durante estos aos, tanto momentos buenos como momentos malos.

Gracias a todos de corazon.



# Tabla de Contenidos

<b>LISTA DE TABLAS .....</b>	<b>VII</b>
<b>LISTA DE FIGURAS .....</b>	<b>VIII</b>
<b>ABREVIACIONES .....</b>	<b>IX</b>
<b>CAPÍTULO 1. INTRODUCCIÓN .....</b>	<b>1</b>
1.1. INTRODUCCIÓN GENERAL .....	1
1.2. TRABAJOS PREVIOS .....	1
1.2.1. <i>De los Modelos</i> .....	2
1.2.2. <i>Del Control</i> .....	3
1.2.3. <i>Discusión</i> .....	9
1.3. HIPÓTESIS DE TRABAJO .....	10
1.4. OBJETIVOS .....	10
1.4.1. <i>Objetivo General</i> .....	10
1.4.2. <i>Objetivos Específicos</i> .....	10
1.5. ALCANCES Y LIMITACIONES .....	10
1.6. TEMARIO Y METODOLOGÍA .....	11
<b>CAPÍTULO 2. MODELACIÓN DEL MICROCANAL.....</b>	<b>12</b>
2.1. INTRODUCCIÓN .....	12
2.2. CARACTERÍSTICAS DE LA PLANTA.....	13
2.2.1. <i>Microcanal</i> .....	13
2.3. TIPOS DE MODELO.....	14
2.3.1. <i>Modelo no lineal de Saint-Venant</i> .....	14
2.3.2. <i>Modelo no lineal fenomenológico</i> .....	15
2.3.3. <i>Modelo lineal simplificado</i> .....	18
2.4. MODELO DEL MICROCANAL .....	21
2.4.1. <i>Filtrado de señales</i> .....	21
2.4.2. <i>Obtención del modelo del Microcanal</i> .....	23
<b>CAPÍTULO 3. DISEÑO DE CONTROLADORES.....</b>	<b>27</b>
3.1. INTRODUCCIÓN .....	27
3.2. ESQUEMAS DE CONTROL .....	27
3.2.1. <i>Control aguas arriba</i> .....	27
3.2.2. <i>Control aguas abajo</i> .....	28
3.3. CONTROLADOR PI.....	29
3.3.1. <i>Introducción</i> .....	29
3.3.2. <i>Diseño del controlador</i> .....	30
3.4. CONTROLADOR DE LÓGICA DIFUSA.....	32
3.4.1. <i>Introducción</i> .....	32
3.4.2. <i>Diseño del controlador</i> .....	37
<b>CAPÍTULO 4. SIMULACIÓN DE CONTROLADORES .....</b>	<b>40</b>
4.1. INTRODUCCIÓN .....	40
4.2. CONTROLADOR PI.....	40
4.3. CONTROLADOR DE LÓGICA DIFUSA .....	41
4.4. CONCLUSIONES .....	42
<b>CAPÍTULO 5. IMPLEMENTACIÓN DE CONTROLADORES.....</b>	<b>43</b>
5.1. INTRODUCCIÓN .....	43
5.2. CONTROLADOR PI.....	43
5.3. CONTROLADOR DE LÓGICA DIFUSA.....	44
5.4. CONCLUSIONES .....	45

<b>CAPÍTULO 6. CONCLUSIONES .....</b>	<b>46</b>
6.1. SUMARIO .....	46
6.2. CONCLUSIONES .....	46
6.3. TRABAJO FUTURO .....	47
<b>BIBLIOGRAFÍA.....</b>	<b>48</b>
<b>ANEXO A. DESARROLLO DE ECUACIONES .....</b>	<b>50</b>
A.1. MODELO DE PRIMER ORDEN CON RETARDO.....	50
A.1.1 Transformada de Laplace a Transformada Z.....	50
A.1.2 Transformada Z a Modelo Discreto .....	51
A.2. CONTROLADOR PI.....	52
A.2.1 Integral tiempo continuo a discreto.....	52
A.2.2 Discretización del controlador .....	54
<b>ANEXO B. INSTRUMENTACIÓN.....</b>	<b>55</b>
B.1. PLC .....	55
B.2. SENSORES DE NIVEL.....	55
B.3. COMPUERTAS.....	56
<b>ANEXO C. CÓDIGOS .....</b>	<b>57</b>
C.1. SIMULACIÓN MODELO .....	57
C.2. CONTROLADOR PI.....	59
C.3. CONTROLADOR FUZZY .....	61
C.3.1. Controlador .....	61
C.3.2. Fusificación .....	66
C.3.3. Defusificación.....	67



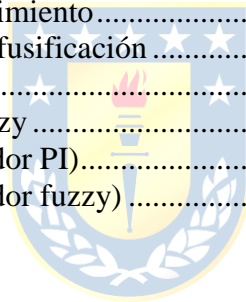
## Lista de Tablas

Tabla 2.1. Medidas microcanal .....	13
Tabla 2.2. Parámetros microcanal .....	23
Tabla 3.1. Método de sintonización de Ziegler-Nichols en lazo abierto.....	31
Tabla 3.2. Parámetros Controlador PI.....	31
Tabla 3.3. Ejemplo Fusificación .....	33



## Lista de Figuras

Figura 2.1. Microcanal .....	13
Figura 2.2. Diagrama P&ID de planta microcanal.....	14
Figura 2.3. Representación de microcanal .....	16
Figura 2.4. Señal filtrada por sensor .....	21
Figura 2.5. Señales filtradas por código.....	22
Figura 2.6. Simulación Microcanal (Escalón Compuerta 1).....	24
Figura 2.7. Simulación Microcanal (Escalón Compuerta 2).....	25
Figura 2.8. Simulación Microcanal (Escalón Compuertas 1 y 2) .....	26
Figura 3.1. Control Aguas Arriba.....	28
Figura 3.2. Control Aguas Abajo .....	29
Figura 3.3. Diagrama de controlador PI.....	29
Figura 3.4. Controlador de lógica difusa.....	32
Figura 3.5. Ejemplo Universo Discurso Fusificación .....	33
Figura 3.6. Ejemplo Base de conocimiento.....	34
Figura 3.7. Universo Discurso Fusificación Error de Nivel.....	37
Figura 3.8. Universo Discurso Fusificación Derivada del Error.....	37
Figura 3.9. Base de conocimiento .....	39
Figura 3.10. Distribución base de conocimiento.....	39
Figura 3.11. Universo Discurso para Defusificación .....	39
Figura 4.1. Simulación Controlador PI .....	40
Figura 4.2. Simulación Controlador Fuzzy .....	41
Figura 5.1. Respuesta escalón (controlador PI).....	43
Figura 5.2. Respuesta escalón (controlador fuzzy) .....	44





# Abreviaciones

## Mayúsculas

FDT	: función de transferencia
$Q_e$	: Flujo de entrada microcanal
$Q_i$	: Flujo de salida i-ésima piscina
$C_d$	: Coeficiente de descarga
L	: Ancho de microcanal
$A_i$	: Área basal i-ésima piscina
$u_i$	: Apertura compuerta de salida de i-ésima piscina
g	: Aceleración de gravedad
$y_i$	: Nivel i-ésima piscina
PI	: controlador proporcional integral
SISO	: sistema de una entrada y una salida (single input single output).
MIMO	: sistema de varias entradas y varias salidas (multiple inputs multiple outputs)
FLC	: controlador de lógica difusa
PB	: positivo grande / aumentar acción de control (grande)
PM	: positivo medio / aumentar acción de control (medio)
PS	: positivo pequeño / aumentar acción de control (pequeño)
ZE	: cero / mantener acción de control
NS	: negativo pequeño / disminuir acción de control (pequeño)
NM	: negativo medio disminuir acción de control (medio)
NB	: negativo grande / disminuir acción de control (grande)

# Capítulo 1. Introducción

---

## 1.1. Introducción General

El agua es un recurso vital para la vida de las personas y, en general, la vida en el planeta. Actualmente ésta se utiliza como si fuera un recurso prácticamente ilimitado, sin pensar en su sustentabilidad. La mayor parte del agua disponible es utilizada en la agricultura, y es precisamente una de las áreas que es menos eficiente en su uso (sólo se aprovecha alrededor de 30% del total de agua utilizada). Es por esta razón que surge la necesidad de encontrar nuevas formas de aprovechar mejor este recurso, de modo que su sustentabilidad no se vea afectada.

En este trabajo se busca ayudar a resolver este problema mediante la implementación de una estrategia de control de tipo fuzzy para gestionar el agua que fluye por una planta microcanal ubicada en el Laboratorio de Control del Departamento de Ingeniería Eléctrica de la Universidad de Concepción, el cual simula un canal de regadío. El microcanal está dividido por una serie de compuertas, las cuales forman piscinas entre ellas. Estas compuertas permiten que el agua fluya controladamente por el canal. Así, y en base al modelo del microcanal, se diseña, simula e implementa el controlador difuso, el cual es comparado con un controlador del tipo PI, con el fin de determinar cuál estrategia es más conveniente de implementar.

## 1.2. Trabajos Previos

En esta sección se explican algunos de los puntos más importantes de un grupo de trabajos utilizados en este informe. Se distinguen dos secciones: modelos y control.

### 1.2.1. De los Modelos

- ♣ Xavier Lítico and Vincent Fromion, *Modeling and Control of Hydrosystems*. 1st Edition, Springer, 2009 [1].

En este libro se pueden encontrar las bases para la modelación y control de hidrosistemas. Se detallan distintos tipos de modelos no lineales que describen el comportamiento de este tipo de sistemas, basados en las ecuaciones de Saint-Venant. Luego se introducen modelos lineales simplificados basados en rangos de operación de los modelos no lineales.

Esto permite simplificar el modelo del microcanal, asumiendo cada piscina como un modelo de primer orden con retardo. De esta forma se puede tratar al sistema como lineal para puntos de operación determinados.

- ♣ Bruno Moraga Ramírez, “Modelación de Microcanal para Control y Estimación”, Memoria de Título para optar al título de Ingeniero Civil Electrónico, Marzo 2014, Departamento de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de Concepción [2].

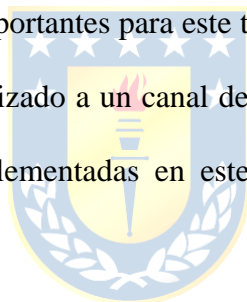
En este trabajo se aborda la modelación de canales de regadío con énfasis en los procesos de identificación de parámetros que rigen el comportamiento de este tipo de sistemas. En particular, se realizó la modelación de la planta microcanal ubicada en el Laboratorio de Control de la Universidad de Concepción, que es la planta a utilizar en el presente trabajo. El modelo presentado asume cada piscina como un proceso de primer orden con retardo para puntos de operación dados.

Es de gran importancia conocer el proceso a controlar mediante un correcto y validado modelo, pues de esta forma es posible diseñar controladores que controlen, valga la redundancia, de la forma más óptima el proceso. Además, un buen modelo permite que los resultados simulados se ajusten bastante a los resultados reales.

- ♣ Carlos Sepúlveda Toepfer, “Modelación Dinámica y Control con Restricciones de Sistemas de Canales de Regadío”, Memoria de Título para optar al título de Ingeniero Civil Electrónico, Abril 2002, Departamento de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de Concepción [3].

En este trabajo se abordan distintos métodos de modelación de canales de regadío, cuyo flujo se rige por las ecuaciones de Saint-Venant. Luego estudia la aplicabilidad de un controlador para el manejo del nodo hidráulico Remulcado, el cual es parte del sistema de riego Digua, ubicado en la 7ma Región de Chile. También se estudió el control descentralizado de los tres primeros tramos de uno de sus canales.

Los resultados obtenidos son importantes para este trabajo, ya que demuestran que es posible aplicar estrategias de control descentralizado a un canal de regadío y obtener resultados favorables. Así se espera que las estrategias implementadas en este trabajo, al tratarse de una planta más pequeña, resulten exitosas.



### **1.2.2. Del Control**

- ♣ Dennis Calvo Salinas, “Estrategias de Control SISO para una Red de Canales de Regadío”, Memoria de Título para optar al título de Ingeniero Civil Electrónico, Abril 2001, Departamento de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de Concepción [4].

En este trabajo se detalla el desarrollo de un simulador de canales de regadío basado en las ecuaciones de Saint-Venant. Se muestran métodos de validación de modelos matemáticos con el fin de demostrar que el modelo desarrollado es representativo y confiable. También se diseñaron controladores basados en dos estrategias de control SISO: lógica difusa y control adaptativo-predictivo.

La información obtenida de este trabajo es relevante, ya que demuestra mediante simulaciones que es posible implementar un controlador de lógica difusa a un canal de regadío, y presenta pruebas a fin de comparar este control con uno del tipo adaptativo-predictivo. Las pruebas de validación del modelo del canal también son importantes, pues permiten determinar si el modelo a utilizar en este trabajo es representativo o no del proceso real. Por último, muestra que, debido a las características de los canales de regadío, para realizar un control aguas abajo óptimo es necesario incluir un controlador esclavo de flujo de salida para cada compuerta, cuya referencia es entregada por el controlador maestro de nivel.

- ♣ Wagner Ramírez Ortega, “Control en Tiempo Continuo para Planta Microcanal”, Memoria de Título para optar al título de Ingeniero Civil Electrónico, Agosto 2013, Departamento de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de Concepción [5].

En este trabajo se aborda la modelación de canales de regadío con énfasis en los procesos de identificación de parámetros que rigen el comportamiento de este tipo de sistemas. Se entregan conocimientos para la modelación, simulación y control de microcanales, los cuales son versiones a pequeña escala de canales de regadío. Define el concepto de piscina (también conocido como tramo) que corresponde al área ubicada entre dos compuertas. También presentan estrategias de control del microcanal como son control PI con predictor Smith y de asignación de polos.

Si bien el objetivo de este trabajo no es realizar control con alguna de esas estrategias, el trabajo muestra métodos de sintonización válidos para controladores PI en general. Además, la planta utilizada es la misma usada en este trabajo.

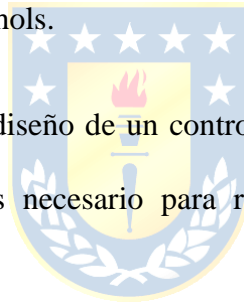
Una de las cosas relevantes que presenta este paper es que es posible aplicar control difuso a plantas no lineales, ya que este control se basa en el conocimiento experto de la planta y no en el modelo exacto de esta. Otro dato relevante son los criterios de estabilidad presentados, ya que es

importante tenerlos en consideración al momento de diseñar el controlador difuso, de modo que no se produzca inestabilidad del proceso por una mala elección de reglas, interfase de fusificación y/o interfase de defusificación.

- ♣ Karl J. Astrom and Tore Haggund, *PID Controllers: Theory, Design and Tuning*. 2nd Edition, ISA, 1995 [6].

En este libro se muestran los conceptos básicos para realizar control de tipo PI. Se introducen conceptos básicos, entre los que se encuentran respuesta escalón, respuesta en frecuencia y estimación de parámetros. Se introduce también el concepto de control PID, que constituye una de las estrategias de control más usadas. Se introducen métodos de diseño, entre los que se encuentra el método de sintonización de Ziegler-Nichols.

Esto constituye la base para el diseño de un controlador PI que, a pesar de no ser la mejor opción para este tipo de sistemas, es necesario para realizar la comparación entre este y el controlador de lógica difusa presentado.



- ♣ D. Driankovm H. Hellendoorn and M. Reinfrank, *An Introduction to Fuzzy Control*. 2nd Edition, Springer, 2010 [7].

En este libro se presentan las bases para la realización de controladores de lógica difusa. Se presentan terminología y operaciones sobre conjuntos difusos, además de otras definiciones relevantes. También se presenta la estructura básica de un controlador de lógica difusa, explicando cada una de sus etapas esenciales y ejemplificando su uso.

Se presenta también el diseño de controladores de lógica difusa multivariable, explicando en detalle cada etapa y usando ejemplos. Este último tópico se vuelve relevante al momento de querer

controlar más de una piscina del microcanal a la vez.

- ♣ Michio Sugeno and Takahiro Yasukawa. “A Fuzzy-Logic-Based Approach to Qualitative Modeling”, *IEEE Transactions on Fuzzy Systems*, vol 1, no 1, pp. 7-31, Feb. 1993 [8].

En este trabajo se presenta un acercamiento general a la modelación cualitativa basada en lógica difusa. El método de modelación cualitativa se divide en dos partes: modelación difusa y aproximación lingüística. Se propone utilizar un método de agrupación difusa para identificar la estructura del modelo difuso. También muestra algunos ejemplos de modelación, entre ellos un modelo de un proceso dinámico y un modelo de acción de control de operador humano.

Debido a las características del proceso a controlar en este trabajo, este paper resulta de ayuda ya que muestra, a modo de ejemplo, control difuso aplicado a distintos procesos. En particular resulta de utilidad aquel que involucra un sistema no lineal.

- ♣ Hua O. Wang, Kazuo Tanaka and Michael F. Griffin. “An Approach to Fuzzy Control of Nonlinear Systems: Stability and Design Issues”, *IEEE Transactions on Fuzzy Systems*, vol 4, no 1, pp. 14-23, Feb. 2006 [9].

En este trabajo se presenta una metodología de diseño para la estabilización de sistemas no lineales mediante control difuso. Primero se representa una planta no lineal con un modelo difuso. Luego es empleado un diseño de controlador difuso basado en el modelo utilizando el concepto de la llamada “compensación distribuida paralela”. Finalmente se presentan criterios de estabilidad para sistemas basados en lógica difusa.

- ♣ M. Santos, S. Dormido and J.M. de la Cruz. “Fuzzy-PID Controllers vs. Fuzzy-PI Controllers”, *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems*, vol 3, pp. 1598-1604, September 1996 [10].

En este trabajo se realiza un estudio cuantitativo y cualitativo de controladores difusos para el caso más completo de un controlador Fuzzy-PID. El desempeño analítico es resumido en términos de tres variables de entrada, lo que permite obtener valores iniciales para los factores de escalamiento del controlador difuso en términos de los parámetros de un controlador PID clásico.

- ♣ Rames C. Panda, *Introduction to PID Controllers – Theory, Tuning and Application to Frontier Areas*. 1st Edition, Intech, 2012 [11].

En este trabajo se dan algunas directrices y recomendaciones para aumentar la calidad de los sistemas de control del tipo difuso, basados en la experiencia del autor. Se muestra un método que permite realizar una pseudo-equivalencia entre controladores PID convencionales y controladores de lógica difusa. Se realizan algunas consideraciones relacionadas al análisis de estabilidad de los sistemas de control basados en control fuzzy. También se presentan algunos métodos de diseño de controladores Fuzzy PID.

- ♣ A. Lacasta, M. Morales-Hernández, P. Brufau, P. García-Navarro. “Simulation of PID control applied to irrigation channels”, *Procedia Engineering*, vol 70, pp. 978-987, 2014 [12].

En este trabajo se implementó un controlador PID discreto descentralizado, en conjunto con un modelo hidráulico. Se hace una introducción a la modelación del sistema explicando las ecuaciones de Saint-Venant y cómo resolverlas numéricamente. Se muestra el diseño del controlador PID utilizado y los resultados obtenidos al implementarlo.



En este paper se puede ver que es posible controlar un canal de regadío utilizando sólo control de tipo PID, y que el uso de predictores más sofisticados u otras técnicas de control más avanzadas resultan en un control más preciso y no constituyen una forma única de controlar el canal. De esto se puede intuir que es posible controlar el microcanal utilizando un controlador PI, el cual servirá de punto de comparación para el controlador de lógica difusa.

- ♣ Petra Bernhoff, “System identification and control of an irrigation channel with a tunnel”, Tesis, 2009, Teknisk-naturvetenskaplig fakultet, UTH-enheten, Uppsala Universitet [13].

En este trabajo se realizó la identificación del sistema de un canal de regadío, detallando el procedimiento utilizado. Se hace hincapié en la estimación de parámetros y validación del modelo. Se diseñó un controlador PI, el cual fue sintonizado mediante técnicas de respuesta en frecuencia. Finalmente se muestran resultados simulados de la implementación del control.

Lo presentado en esta tesis es de valor pues muestra una forma distinta de sintonización para controladores PI (sintonización en frecuencia) lo cual resulta particularmente útil en este tipo de sistemas en donde perturbaciones, tales como ondas estacionarias, pueden afectar la estabilidad del sistema.

### 1.2.3. Discusión

Actualmente en el país existen canales de regadío, los que en su mayoría son accionados directamente por operadores en base a su experiencia, sin ayuda de alguna estrategia de control realimentado. Sin embargo, es evidente que automatizar estos procesos lleva a un mayor aprovechamiento del agua y, al existir la tecnología necesaria para esto, el accionar manual de las compuertas se vuelve obsoleto. Es así como se vuelve necesario determinar qué estrategia de control es la más adecuada para automatizar el proceso.

La revisión realizada indica que es factible aplicar estrategias de control PI y control difuso al microcanal, ya que es posible realizar un modelo que represente fielmente las dinámicas que rigen el sistema. El uso de controladores PI en canales de regadío se considera aceptable, pero resultan en controladores con sobrepaso considerable en los niveles de cada piscina, lo que induce a perturbaciones en las piscinas aguas abajo y, de forma menos directa, las piscinas aguas arriba. Solucionar el problema del sobrepaso en un controlador PI implica aumentar el tiempo de asentamiento, por lo que se prefiere usar estrategias del tipo predictivo.

El control difuso (fuzzy) es una estrategia de control predictivo de tipo experto. Esto quiere decir que toma decisiones en base al conocimiento previo del proceso a controlar, por lo que presenta una ventaja en los procesos del tipo no lineal, ya que no se necesita del modelo exacto. Esto permite el control del nivel de las piscinas de forma que se reduzca el sobrepaso y el tiempo de asentamiento que tendría un controlador PI.

En este trabajo se pretende utilizar el enfoque de control difuso para el control del microcanal. Se desarrollan algoritmos para efectuar un control exitoso, realizando simulaciones y pruebas experimentales comparativas entre una estrategia de control tipo fuzzy y una del tipo PI.

### 1.3. Hipótesis de Trabajo

Es posible aplicar una estrategia de control difuso al microcanal, asegurando estabilidad del sistema, mejorando el tiempo de asentamiento y disminuyendo el sobrepaso de la respuesta frente a un controlador PI.

### 1.4. Objetivos

#### 1.4.1. Objetivo General

Comparar el desempeño en simulación y en planta de un controlador difuso y un controlador PI convencional.

#### 1.4.2. Objetivos Específicos

- Modelar el microcanal.
- Diseñar un controlador PI para el microcanal.
- Diseñar un controlador difuso para el microcanal.
- Simular ambos controladores.
- Implementar ambos controladores.



### 1.5. Alcances y Limitaciones

Se utiliza la planta microcanal ubicada en el Laboratorio de Control del Departamento de Ingeniería Eléctrica de la Universidad de Concepción. Se utiliza el equipamiento disponible, esto es PLC Allen Bradley, software RsLogix 5000 y las compuertas con actuadores neumáticos. En el caso de las simulaciones se trabaja sobre el software Matlab. El control a implementar será descentralizado para dos piscinas con esquema de control aguas arriba.

## 1.6. Temario y Metodología

El Capítulo 1 da una instrucción al tema tratado en la presente Memoria de Título; se realiza una revisión bibliográfica. El Capítulo 2 corresponde a la modelación del microcanal. El Capítulo 3 corresponde al diseño de controladores, esto es, el controlador PI y el controlador de lógica difusa. El Capítulo 4 muestra las simulaciones de los controladores y una comparación de desempeño. El Capítulo 5 presenta la implementación de ambos controladores en la planta Microcanal. Finalmente, el Capítulo 6 presenta las conclusiones del trabajo realizado.



## Capítulo 2. Modelación del Microcanal

---

### 2.1. Introducción

Por lo general, la modelación de canales de regadío de gran envergadura se realiza a partir de las ecuaciones de Saint-Venant. Estas ecuaciones están basadas en las leyes físicas que describen el proceso de flujo unidimensional en lámina libre. Debido a que estas ecuaciones son no lineales, la mayoría de las veces no tienen solución analítica, por lo que deben resolverse mediante métodos numéricos.

Otra forma de modelar canales de regadío es utilizando un modelo no lineal basado en estanques. Este tipo de modelación por lo general no es utilizada en canales de gran envergadura debido al retardo de transporte de los sensores de nivel, pero es una buena aproximación en canales pequeños de tramos más cortos.

Por último, existen los modelos tipo caja negra. Estos constituyen una modelación simplificada, obteniendo parámetros mediante experimentación a fin de obtener un modelo lineal válido en un rango de operación determinado.

En este capítulo se presentan las características de la planta microcanal a utilizar, se dan a conocer las distintas estrategias de modelación y se determina el modelo con el que se trabaja en los siguientes capítulos.

## 2.2. Características de la planta

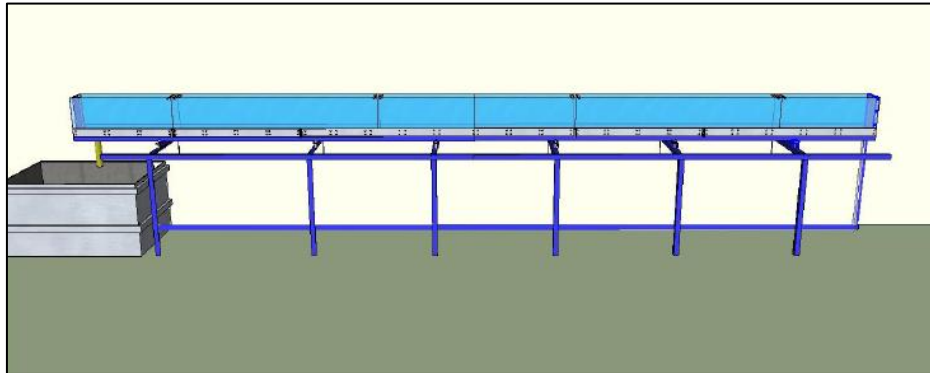
### 2.2.1. Microcanal

El microcanal consiste de un estanque de 7 [m] de largo, 0.345 [m] de alto y 0.145 [m] de ancho. Este cuenta con tres compuertas, dos de las cuales son manipulables mediante actuadores neumáticos, mientras que la tercera mantiene una apertura fija de 0.08 [m]. Las compuertas dividen el microcanal en cuatro piscinas de distintos largos (ver Tabla 2.1).

**Tabla 2.1. Medidas microcanal**

Piscina	Largo
1	2.58[m]
2	2.01[m]
3	1.74[m]
4	0.67[m]

Un almacenador provee de agua al microcanal mediante la utilización de una bomba centrífuga, la cual es activada por un variador de frecuencia. El desagüe del microcanal va a dar al mismo almacenador, por lo que el agua es recirculada. La Figura 2.1 muestra el microcanal, mientras que la Figura 2.2 muestra el diagrama P&ID del microcanal.



**Figura 2.1. Microcanal**

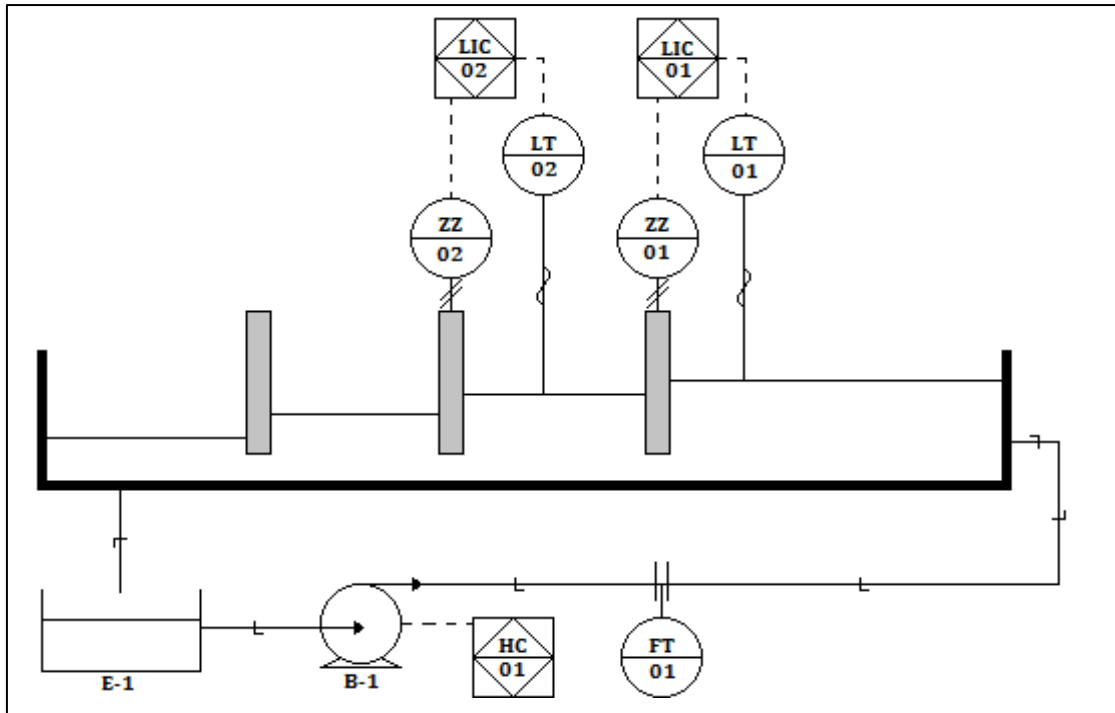


Figura 2.2. Diagrama P&ID de planta microcanal

## 2.3. Tipos de modelo

### 2.3.1. Modelo no lineal de Saint-Venant

En la mayoría de los casos, los canales de riego se modelan en base a las ecuaciones de Saint-Venant, mostradas en (2.1).

$$\frac{\partial y}{\partial t} + v \frac{\partial y}{\partial x} + y \frac{\partial v}{\partial x} = 0$$

$$\frac{\partial v}{\partial t} + g \frac{\partial y}{\partial x} + v \frac{\partial v}{\partial x} = g(I_0 - I_f) \quad (2.1)$$

Donde  $v$  es la velocidad media del agua,  $y$  es el nivel de agua,  $g$  es la aceleración de gravedad,  $I_0$  es la pendiente del canal e  $I_f$  es la pendiente de fricción (también llamada pendiente motriz), la cual representa la pérdida continua de presión debido a la fricción del canal. Este es un conjunto de ecuaciones hiperbólicas en derivadas parciales que describen el flujo bajo superficie de presión en un fluido. Estas constituyen un sistema no lineal en derivadas parciales, por lo que

generalmente se resuelven mediante métodos numéricos de alto coste computacional.

Las ecuaciones de Saint-Venant explican el comportamiento desde la compuerta aguas arriba hasta las cercanías de la compuerta aguas abajo. En el lugar inmediatamente antes de la compuerta aguas abajo los fenómenos hidráulicos son provocados por las diferencias de altura entre las dos piscinas. La cantidad de agua que pasa de la piscina superior a la inferior se puede determinar por medio del principio de Torricelli (2.2).

$$Q = C_d \cdot L \cdot W \cdot \sqrt{2 \cdot g \cdot (h_1 - h_2)} \quad (2.2)$$

Donde  $Q$  es el flujo bajo la compuerta,  $C_d$  es el coeficiente de descarga (normalmente se considera 1, pero también es común encontrar valores entre 0.4 y 0.6),  $L$  es el ancho de la compuerta,  $W$  es la apertura de la compuerta,  $g$  es la aceleración de gravedad,  $h_1$  es el nivel de agua de la piscina aguas arriba y  $h_2$  es el nivel de agua de la piscina aguas abajo. Así, la unión de ambas zonas permite modelar de forma completa un canal de regadío.

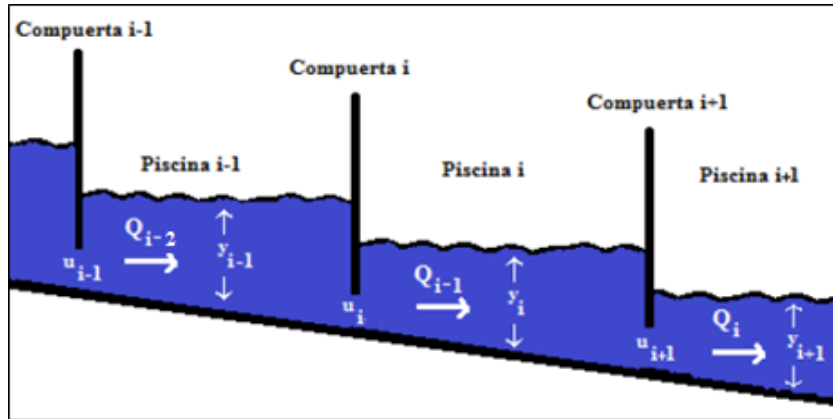
### 2.3.2. Modelo no lineal fenomenológico

Otra forma de modelar el microcanal es considerar cada piscina como un estanque, donde la variación de volumen no es más que la diferencia entre el flujo de entrada y el flujo de salida (flujo de entrada de la piscina aguas abajo). Esto es representado de forma matemática en (2.3) y de forma gráfica en la Figura 2.1.

$$\frac{\partial V_i}{\partial t} = Q_{i-1} - Q_i \quad (2.3)$$

Donde  $Q_i$  es el flujo de salida de la  $i$ -ésima piscina y  $V_i$  es el volumen de la  $i$ -ésima piscina.





**Figura 2.3. Representación de microcanal**

Considerando el microcanal con sección transversal regular, (2.3) es equivalente con (2.4).

$$A_i \frac{\partial y_i}{\partial t} = Q_{i-1}(t) - Q_i \quad (2.4)$$

Donde  $A_i$  es el área de la sección transversal de la  $i$ -ésima piscina e  $y_i$  es el nivel de la  $i$ -ésima piscina. El flujo que pasa bajo una compuerta se puede modelar usando la ecuación mostrada en (2.2). Haciendo un cambio de notación de esta ecuación se obtiene (2.5).

$$Q_i = C_d \cdot L_i \cdot u_i \cdot \sqrt{2 \cdot g \cdot (y_i - y_{i+1})} \quad (2.5)$$

Donde  $L_i$  es el ancho de la  $i$ -ésima piscina y  $u_i$  la compuerta de salida de la  $i$ -ésima piscina. En el caso particular de la última piscina, el flujo de descarga no depende de una compuerta, sino que de un desagüe vertical. Por principio de Torricelli, este flujo puede ser modelado como muestra (2.6).

$$Q_f = C_d \cdot S \cdot \sqrt{2 \cdot g \cdot y_f} \quad (2.6)$$

Donde  $f$  es el número de la piscina final y  $S$  es la superficie de descarga.

Finalmente, a partir de las ecuaciones (2.4), (2.5) y (2.6) es posible modelar por completo el microcanal. Considerando el flujo de entrada de la primera piscina como  $Q_e$ , el modelo de la primera piscina es el mostrado en (2.7), el modelo de la  $i$ -ésima piscina intermedia el mostrado en

(2.8) y el modelo de la piscina final el mostrado en (2.9).

$$\frac{\partial y_1}{\partial t} = \frac{1}{A_1} Q_e(t) - \frac{1}{A_1} C_d \cdot L_1 \cdot u_1 \cdot \sqrt{2 \cdot g \cdot (y_1 - y_2)} \quad (2.7)$$

$$\frac{\partial y_i}{\partial t} = \frac{1}{A_i} C_d \cdot L_{i-1} \cdot u_i \cdot \sqrt{2 \cdot g \cdot (y_{i-1} - y_i)} - \frac{1}{A_i} C_d \cdot L_i \cdot u_i \cdot \sqrt{2 \cdot g \cdot (y_i - y_{i+1})} \quad (2.8)$$

$$\frac{\partial y_f}{\partial t} = \frac{1}{A_{f-1}} C_d \cdot L_{f-1} \cdot u_{f-1} \cdot \sqrt{2 \cdot g \cdot (y_{f-1} - y_f)} - \frac{1}{A_f} C_d \cdot S \cdot \sqrt{2 \cdot g \cdot y_f} \quad (2.9)$$

En particular, el sistema matricial para un sistema de cuatro piscinas (al igual que la planta microcanal) en el mostrado en (2.10).

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \\ \dot{y}_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{A_1} Q_e(t) - \frac{1}{A_1} C_d \cdot L_1 \cdot u_1 \cdot \sqrt{2 \cdot g \cdot (y_1 - y_2)} \\ \frac{1}{A_1} C_d \cdot L_1 \cdot u_1 \cdot \sqrt{2 \cdot g \cdot (y_1 - y_2)} - \frac{1}{A_2} C_d \cdot L_2 \cdot u_2 \cdot \sqrt{2 \cdot g \cdot (y_2 - y_3)} \\ \frac{1}{A_2} C_d \cdot L_2 \cdot u_2 \cdot \sqrt{2 \cdot g \cdot (y_2 - y_3)} - \frac{1}{A_3} C_d \cdot L_3 \cdot u_3 \cdot \sqrt{2 \cdot g \cdot (y_3 - y_4)} \\ \frac{1}{A_3} C_d \cdot L_3 \cdot u_3 \cdot \sqrt{2 \cdot g \cdot (y_3 - y_4)} - \frac{1}{A_4} C_d \cdot S \cdot \sqrt{2 \cdot g \cdot y_4} \end{bmatrix} \quad (2.10)$$

Este sistema se puede discretizar, considerando la discretización de  $\dot{y}_1$  como  $(y_{i_{k+1}} - y_{i_k})/T_m$ , donde  $T_m$  es el tiempo de muestreo. El modelo discreto de (2.10) se muestra en (2.11).

$$\begin{bmatrix} y_{1k} \\ y_{2k} \\ y_{3k} \end{bmatrix} = \begin{bmatrix} \frac{T_m}{A_1} Q_{e_{k-1}} - \frac{T_m}{A_1} C_d \cdot L_1 \cdot u_{1_{k-D_1-1}} \cdot \sqrt{2 \cdot g \cdot (y_{1_{k-1}} - y_{2_{k-1}})} + y_{1_{k-1}} \\ \frac{T_m}{A_1} C_d \cdot L_1 \cdot u_{1_{k-D_1-1}} \cdot \sqrt{2 \cdot g \cdot (y_{1_{k-1}} - y_{2_{k-1}})} - \frac{T_m}{A_2} C_d \cdot L_2 \cdot u_{2_{k-D_2-1}} \cdot \sqrt{2 \cdot g \cdot (y_{2_{k-1}} - y_{3_{k-1}})} + y_{2_{k-1}} \\ \frac{T_m}{A_2} C_d \cdot L_2 \cdot u_{2_{k-D_2-1}} \cdot \sqrt{2 \cdot g \cdot (y_{2_{k-1}} - y_{3_{k-1}})} - \frac{T_m}{A_3} C_d \cdot L_3 \cdot u_{3_{k-D_3-1}} \cdot \sqrt{2 \cdot g \cdot (y_{3_{k-1}} - y_{4_{k-1}})} + y_{3_{k-1}} \\ \frac{T_m}{A_3} C_d \cdot L_3 \cdot u_{3_{k-D_3-1}} \cdot \sqrt{2 \cdot g \cdot (y_{3_{k-1}} - y_{4_{k-1}})} - \frac{T_m}{A_4} C_d \cdot S \cdot \sqrt{2 \cdot g \cdot y_{4_{k-1}}} + y_{4_{k-1}} \end{bmatrix} \quad (2.11)$$

Nótese que en la expresión (2.11) se incluyeron los retardos  $D_1$ ,  $D_2$  y  $D_3$ , los cuales

representan el retardo de transporte. Las dinámicas de la planta microcanal pueden ser representadas de forma fehaciente por este modelo, sin embargo, al ser un modelo no lineal no permite de forma directa obtener los parámetros para un controlador PI, por lo que es necesario un modelo lineal de la planta para el diseño de este tipo de controlador.

### 2.3.3. Modelo lineal simplificado

Como se mencionó anteriormente, para el diseño de un controlador PI es necesario un modelo lineal de la planta. En [2] se determinó que las dinámicas del microcanal se pueden modelar de forma fehaciente mediante modelos de primer orden con retardo para rangos de operación determinados.

Un modelo de primer orden se caracteriza por tener una ganancia en estado estacionario y un tiempo de asentamiento determinados. La FDT de un sistema de primer orden con retardo se muestra en (2.12).

$$H(s) = \frac{k}{(\tau s + 1)} e^{-\theta s} \quad (2.12)$$

Donde  $k$  es la ganancia en estado estacionario,  $\tau$  es la constante de tiempo y  $\theta$  es el retardo. Para las simulaciones de controlador es necesario el modelo discreto de la planta incluyendo un retentor de orden cero, por lo que se debe aplicar transformada Z a la FDT mostrada en la (2.13).

$$\frac{y(s)}{u(s)} = \frac{(1 - e^{-T_m})}{s} \cdot \frac{k}{(\tau s + 1)} e^{-\theta s} \quad (2.13)$$

Donde  $T_m$  es el tiempo de muestreo. Luego, la transformada Z de la FDT es la mostrada en (2.14).

$$\frac{y(z^{-1})}{u(z^{-1})} = z^{-(D+1)} \cdot \frac{k \left(1 - e^{-\frac{T_m}{\tau}}\right)}{\left(1 - z^{-1} e^{-\frac{T_m}{\tau}}\right)} \quad (2.14)$$

Donde  $D = \theta/T_m$  es el retardo discreto. Finalmente, el modelo discreto es el mostrado en (2.15).

$$y_k = e^{-\frac{T_m}{\tau}} \cdot y_{k-1} + k \left(1 - e^{-\frac{T_m}{\tau}}\right) \cdot u_{k-D-1} \quad (2.15)$$

El desarrollo en detalle de cómo llegar desde la FDT en Laplace al modelo discreto se puede ver en el Anexo A.

### 2.3.3.1. Identificación por mínimos cuadrados para sistemas de primer orden

La identificación por mínimos cuadrados permite obtener los parámetros de un sistema con una estructura determinada (en este caso, un sistema de primer orden) a partir de datos obtenidos de manera experimental.

La estructura general de un modelo paramétrico puede escribirse como:

$$y = \varphi^T \theta \quad (2.16)$$

Donde  $y$  es la salida del sistema,  $\varphi$  es un vector de  $n$  cantidades conocidas y  $\theta$  es un vector de  $n$  parámetros conocidos.

Se debe encontrar el estimador  $\hat{\theta}$  del vector de parámetros  $\theta$ . Para esto se reduce (2.16) a un sistema matricial:

$$Y = M \hat{\theta} \quad (2.17)$$

Este sistema se puede resolver fácilmente como se muestra en (2.18) si la matriz  $M$  es no singular, es decir, cuadrada invertible.

$$\hat{\theta} = M^{-1}Y \quad (2.18)$$

Este es el caso ideal, sin embargo, rara vez sucede debido a que, como se mencionó antes, los conjuntos de datos se obtienen por experimentación, por lo que la cantidad de conjuntos suele exceder el número de variables observables. Al no ser invertible la matriz  $M$  (por no ser cuadrada) se multiplica por su traspuesta a ambos lados de la ecuación mostrada en (2.17), obteniendo:

$$M^T Y = M^T M \hat{\theta} \quad (2.19)$$

La matriz  $M^T M$  es cuadrada, por lo que a partir de (2.19) se puede obtener  $\hat{\theta}$  como se muestra en (2.20), asumiendo que la matriz  $M^T M$  es invertible.

$$\hat{\theta} = (M^T M)^{-1} M^T Y \quad (2.20)$$

Como se mostró anteriormente, el modelo discreto de un sistema de primer orden con retardo corresponde al mostrado en (2.15), el cual puede escribirse como:

$$y_k = \alpha \cdot y_{k-1} + \beta \cdot u_{k-D-1} \quad (2.21)$$

Donde:

$$\alpha = e^{-\frac{T_m}{\tau}} \quad (2.22)$$

$$\beta = k \left( 1 - e^{-\frac{T_m}{\tau}} \right) = k(1 - \alpha) \quad (2.23)$$

La ecuación (2.21) puede escribirse como:

$$y_k = [y_{k-1} \quad u_{k-D-1}] \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (2.24)$$

Ahora, si se considera que se tiene un conjunto de  $n$  datos obtenidos mediante experimentación, es correcto escribir:

$$\begin{bmatrix} y_{2+D} \\ y_{3+D} \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} y_{1+D} & u_1 \\ y_{2+D} & u_2 \\ \vdots & \vdots \\ y_{n-1} & u_{n-D-1} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (2.25)$$

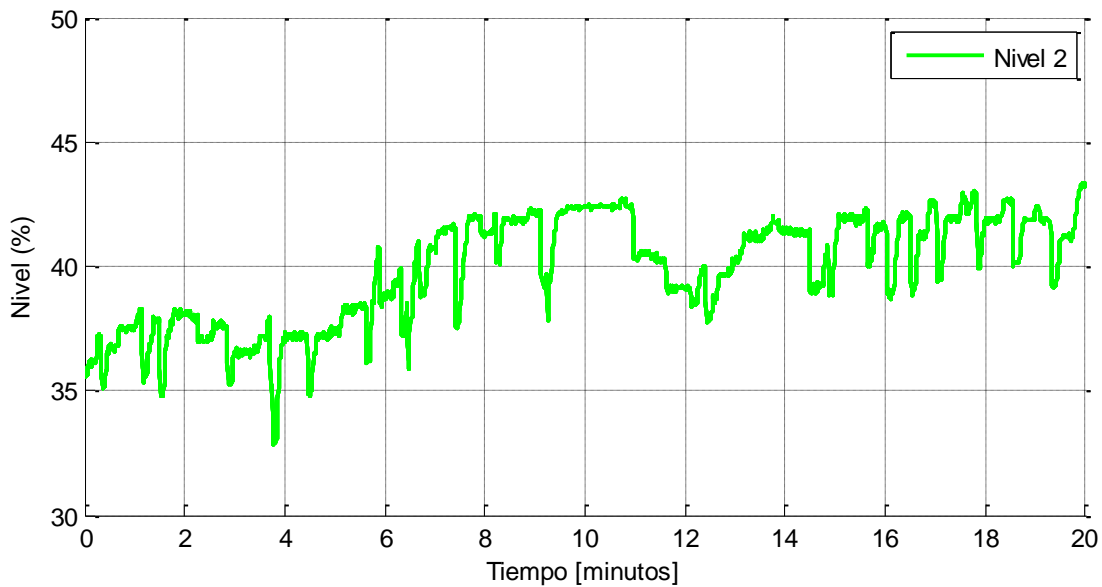
La ecuación (2.25) muestra la misma estructura mostrada en (2.17), por lo que es posible obtener  $\alpha$  y  $\beta$  mediante mínimos cuadrados de la forma mostrada en (2.20). Finalmente, los parámetros del sistema se despejan de (2.22) y (2.23), obteniendo lo mostrado en (2.26).

$$k = \frac{\beta}{(1 - \alpha)} \quad \tau = \frac{T_m}{\ln(\alpha)} \quad (2.26)$$

## 2.4. Modelo del Microcanal

### 2.4.1. Filtrado de señales

Al obtener los primeros datos para la realización del modelo de la planta se observaron ciertos errores de medición; los datos indicaban aumento o disminución de nivel que en realidad no ocurrían (ver Figura 2.4).

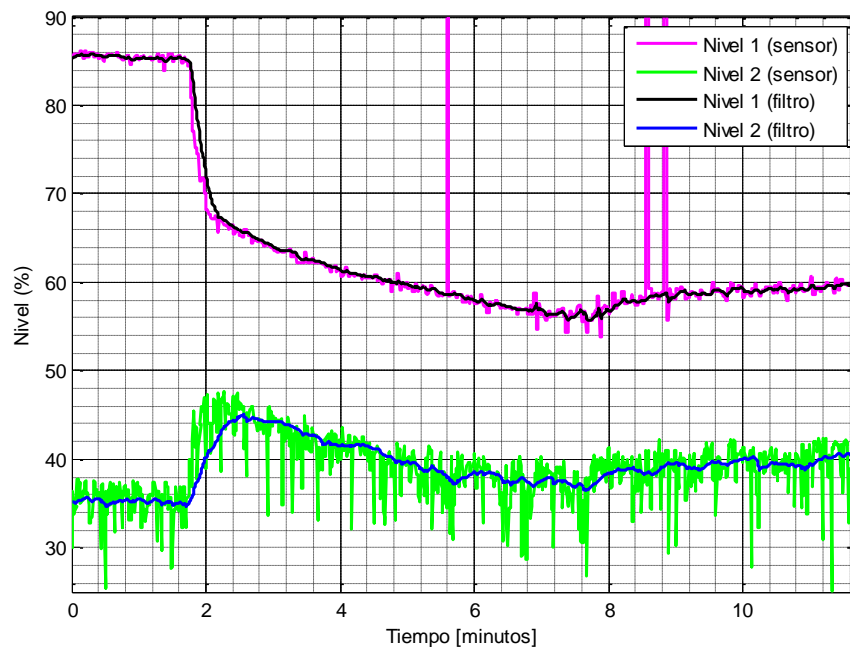


**Figura 2.4. Señal filtrada por sensor**

Esto se producía debido a que los sensores instalados incluyen un tipo de filtro (no especificado por el fabricante) muy sensible al ruido. Debido a esto se optó por desactivar el filtro de los sensores y filtrar las señales directamente en el código del PLC utilizando el filtro mostrado en (2.27), el cual está basado en el suavizado de señales (damping).

$$yf_k = \left(1 - \frac{T_m}{T_f}\right) yf_{k-1} + \frac{T_m}{T_f} y_k \quad (2.27)$$

Donde  $yf_k$  es la señal filtrada,  $T_m$  es el tiempo de muestreo,  $T_f$  es el tiempo de filtrado e  $y_k$  es la señal entregada por el sensor. El funcionamiento del filtro es relativamente simple: para un instante  $k$ , sólo se considera cierto porcentaje de la señal medida (dado por el factor  $T_m/T_f$ ) y el resto se obtiene del valor filtrado en el instante  $k-1$ . La Figura 2.5 muestra la comparación entre las señales entregadas por los sensores y las señales filtradas. Es importante mencionar que al filtrar la señal se introduce cierto retardo, por lo que se optó por utilizar un tiempo de filtrado lo menor posible para no afectar el control en demasía. Aun así, el retardo de este filtro es menor al introducido por el filtro de los sensores.



**Figura 2.5. Señales filtradas por código**

### 2.4.2. Obtención del modelo del Microcanal

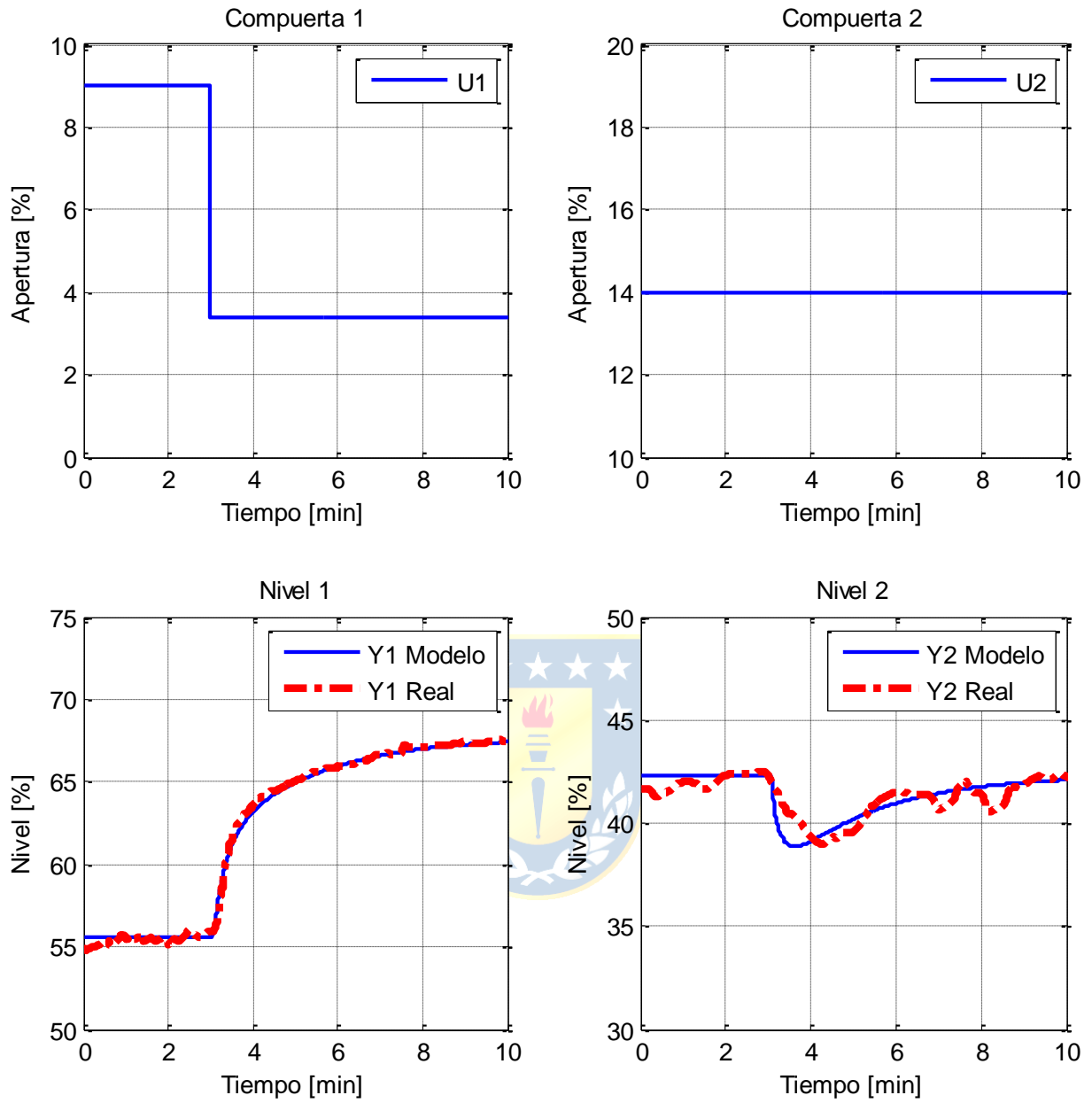
Con vista en las simulaciones a realizar luego de diseñar los controladores, se decide utilizar el modelo no lineal basado en estanques (sección 2.3.2). Para la realización de este modelo se necesitan los parámetros de la planta, los cuales se muestran en la Tabla 2.2. Se escogió un tiempo de muestreo de 500 [ms] ya que, para las dinámicas del microcanal, un tiempo de muestreo menor no genera una ventaja importante y, en el caso del controlador de lógica difusa, puede llegar a ser perjudicial debido a la cantidad de cálculos que se realizan por ciclo.

**Tabla 2.2. Parámetros microcanal**

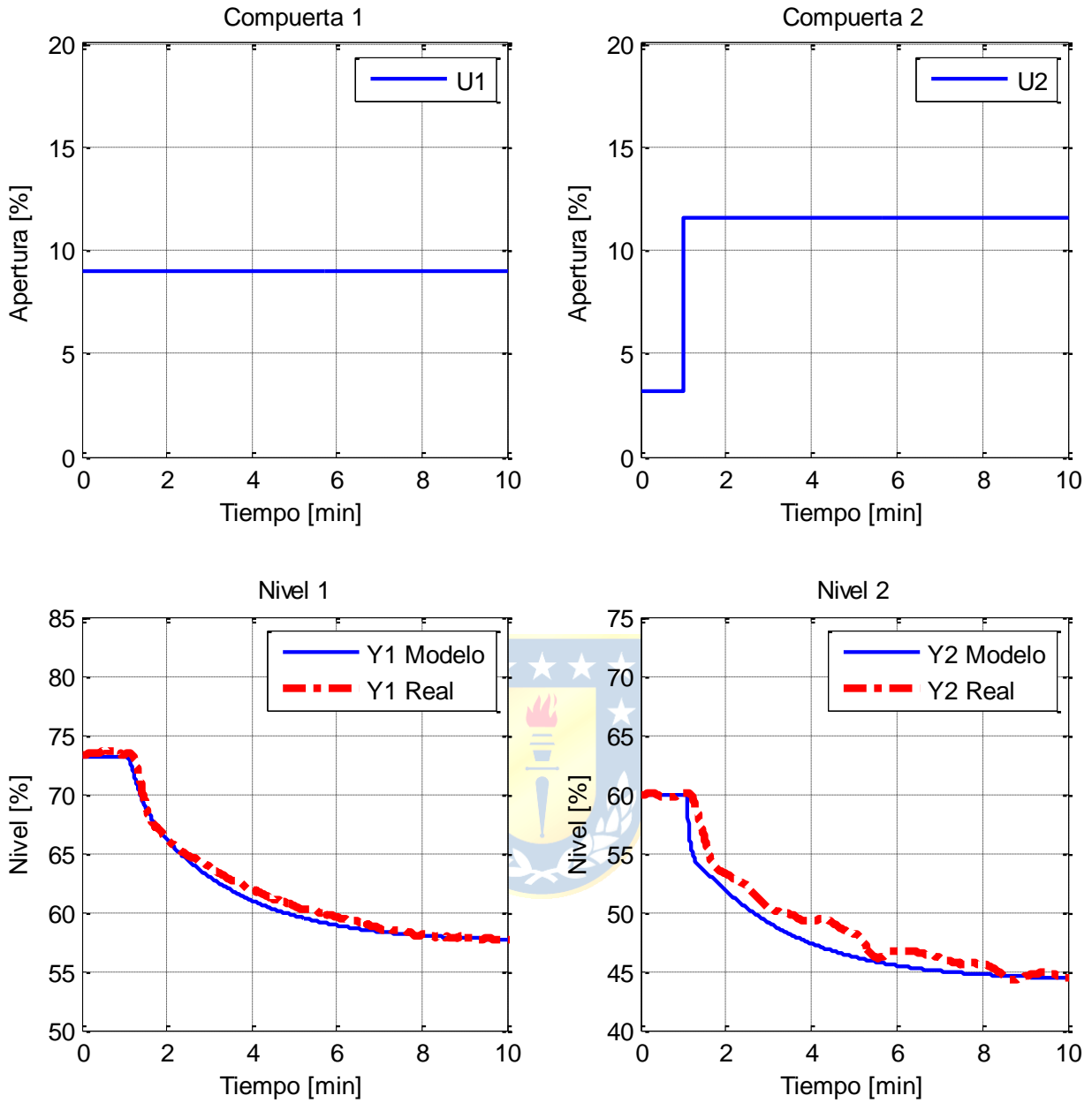
<b>Parámetro</b>	<b>Valor</b>
Frecuencia máxima variador	50[Hz]
Flujo a frecuencia máxima	0.0043 $\left[\frac{m^3}{s}\right]$
Coefficiente de descarga	0.25
Ancho microcanal	0.145[m]
Área basal piscina 1	0.3741[m <sup>2</sup> ]
Área basal piscina 2	0.2914[m <sup>2</sup> ]
Área basal piscina 3	0.2523[m <sup>2</sup> ]
Área basal piscina 4	0.0769[m <sup>2</sup> ]
Apertura mínima compuertas 1 y 2	0.02[m <sup>2</sup> ]
Apertura máxima compuertas 1 y 2	0.20[m <sup>2</sup> ]
Apertura compuerta 3	0.08[m <sup>2</sup> ]
Superficie de descarga piscina 4	0.0034[m <sup>2</sup> ]

La Figura 2.6 y la Figura 2.7 muestran los resultados del modelo, realizando una comparación con los datos reales obtenidos en planta. La Figura 2.6 muestra la respuesta a un escalón de la compuerta 1 y la Figura 2.7 muestra la respuesta a un escalón en la compuerta 2. Por último, la Figura 2.8 muestra una simulación realizada en un rango más amplio, con escalones en ambas compuertas. El 0% y 100% de nivel de agua corresponden a 0[cm] y 30[cm], respectivamente. El 0% y 100% de apertura de compuerta corresponden a 2[cm] y 20[cm], respectivamente (bajo los 2[cm] las compuertas presentan una histéresis muy alta, por lo que se optó por fijar ese valor como mínimo de apertura). Las simulaciones fueron realizadas sobre Matlab R2014a (código en Anexo C).

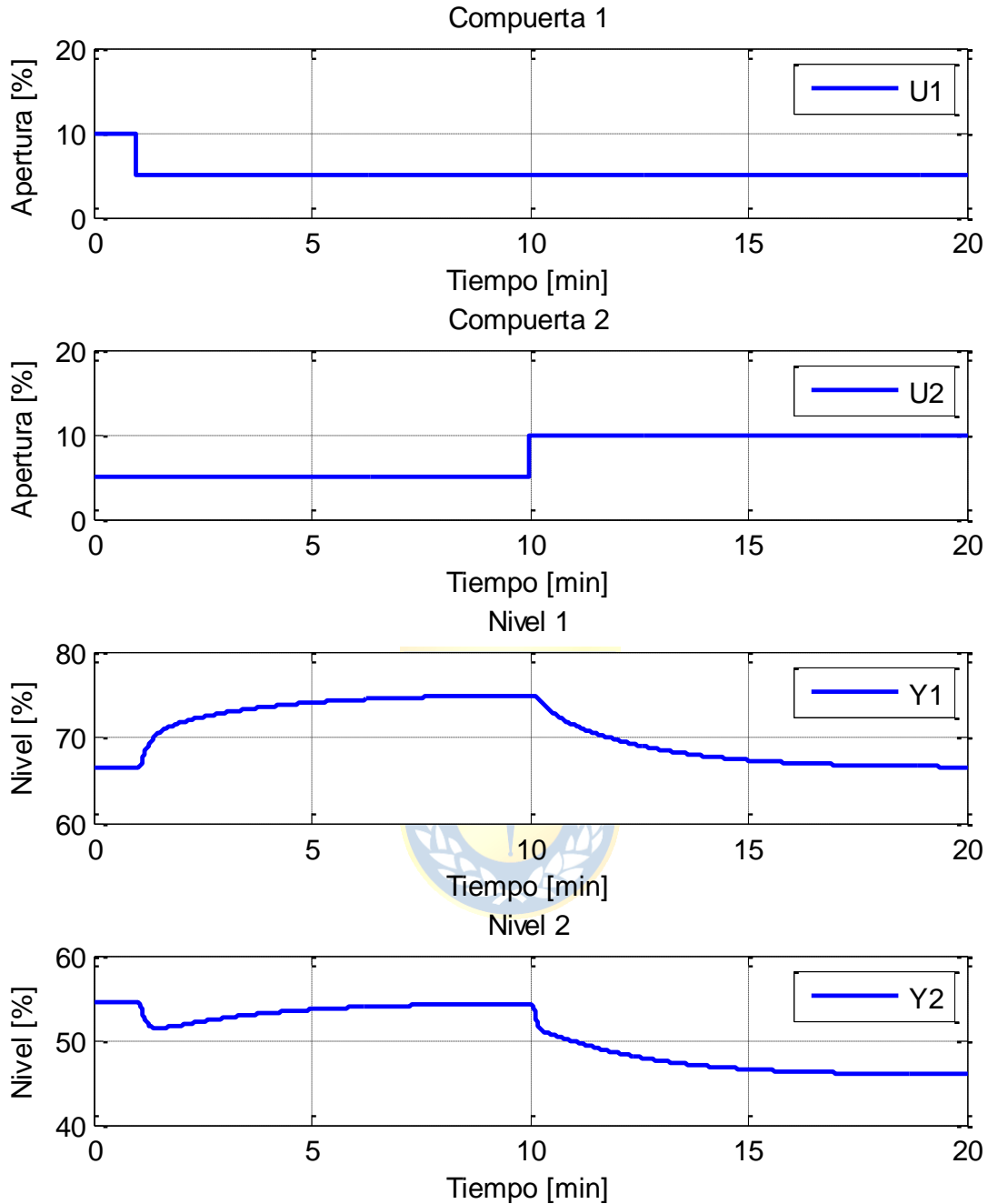




**Figura 2.6. Simulación Microcanal (Escalón Compuerta 1)**



**Figura 2.7. Simulación Microcanal (Escalón Compuerta 2)**



**Figura 2.8. Simulación Microcanal (Escalón Compuertas 1 y 2)**

Se puede ver que el modelo se ajusta bastante al comportamiento real del microcanal. Sin embargo, se puede ver la segunda piscina presenta más perturbaciones que la primera piscina, esto debido a que en la primera piscina hay un aquietador que disminuye las ondas estacionarias, a diferencia de la segunda.

## Capítulo 3. Diseño de controladores

---

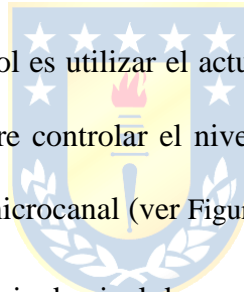
### 3.1. Introducción

En este capítulo se aborda el diseño de los controladores (PI y difuso) implementados en la planta microcanal. En primer lugar se muestran los esquemas de control existentes para canales de regadío, indicando las ventajas de cada uno. Luego se procede al diseño del controlador PI. Finalmente se procede a diseñar el controlador de lógica difusa. La sintonización de ambos controladores se realiza en base al modelo determinado en el Capítulo 2.

### 3.2. Esquemas de control

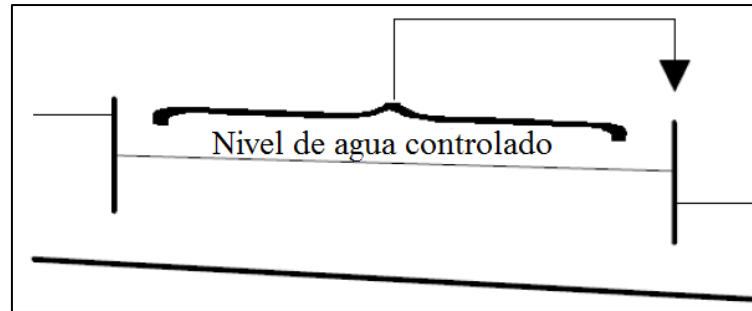
#### 3.2.1. Control aguas arriba

El objetivo de este tipo de control es utilizar el actuador ubicado aguas abajo de la piscina a controlar. Esto significa que si se quiere controlar el nivel de agua en la  $i$ -ésima piscina, se debe utilizar de actuador la compuerta  $i$  del microcanal (ver Figura 3.1).



Realizar un cambio en la referencia de nivel de una de las piscinas provocará un movimiento en la compuerta que controla ese nivel con el objetivo de disminuir el error local de esa piscina. Esto genera una perturbación en el nivel de la piscina aguas debajo de la compuerta que provocará una acción de control en la siguiente compuerta, produciéndose así un efecto en cadena aguas abajo de la compuerta que realiza la acción de control.

Conociendo este efecto de encadenamiento es posible aplicar una estrategia de control predictiva, ya que cuando se accione una compuerta se sabrá con anticipación las posibles perturbaciones sobre las piscinas aguas abajo. De esta forma se puede realimentar aguas abajo esta información para obtener una mejor respuesta.



**Figura 3.1. Control Aguas Arriba.**

### **3.2.2. Control aguas abajo**

El objetivo de este tipo de control es utilizar el actuador ubicado aguas arriba de la piscina a controlar. Esto significa que si se quiere controlar el nivel de agua en la  $i$ -ésima piscina, se debe utilizar de actuador la compuerta  $i-1$  del microcanal (ver Figura 3.2).

Al igual que en el control aguas arriba, realizar un cambio en la referencia de nivel de una de las piscinas provoca perturbaciones en cadena en las piscinas aguas abajo de la compuerta que realiza la acción de control, y también en la piscina aguas arriba. Al igual que en el control aguas arriba, esta serie de perturbaciones se pueden considerar al momento de diseñar un controlador predictivo.

Un problema que surge debido a las perturbaciones generadas en las piscinas aguas abajo de la compuerta accionada es el efecto de “ola dominante”. Esta ola puede provocar un problema en el control que puede ser mitigado limitando la velocidad de las acciones de control. De no imponer esta limitación, la ola puede generar movimientos en las compuertas aguas abajo que produzcan más olas, produciendo que el sistema oscile en torno a la referencia.

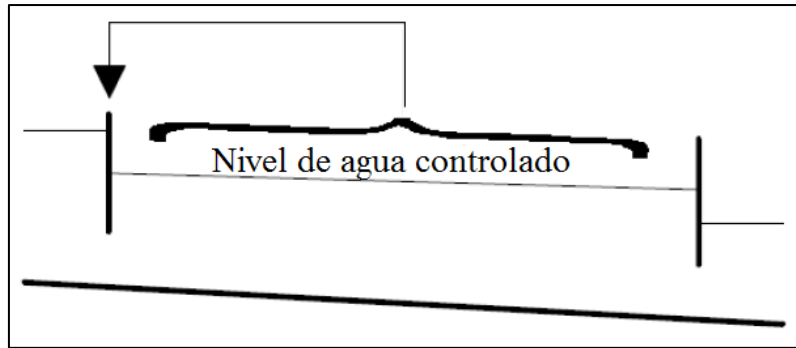


Figura 3.2. Control Aguas Abajo

### 3.3. Controlador PI

#### 3.3.1. Introducción

Los controladores PI constituyen una de las estrategias de control más utilizadas a nivel industrial. Un controlador PI tiene dos componentes: parte proporcional y acción integral. La parte proporcional es la encargada de lograr que el error (diferencia entre referencia y salida del proceso) se aproxime a cero en estado estacionario, mientras que la acción integral se encarga de disminuir y eliminar tal error. El diagrama básico de este tipo de controlador se muestra en la Figura 3.3.

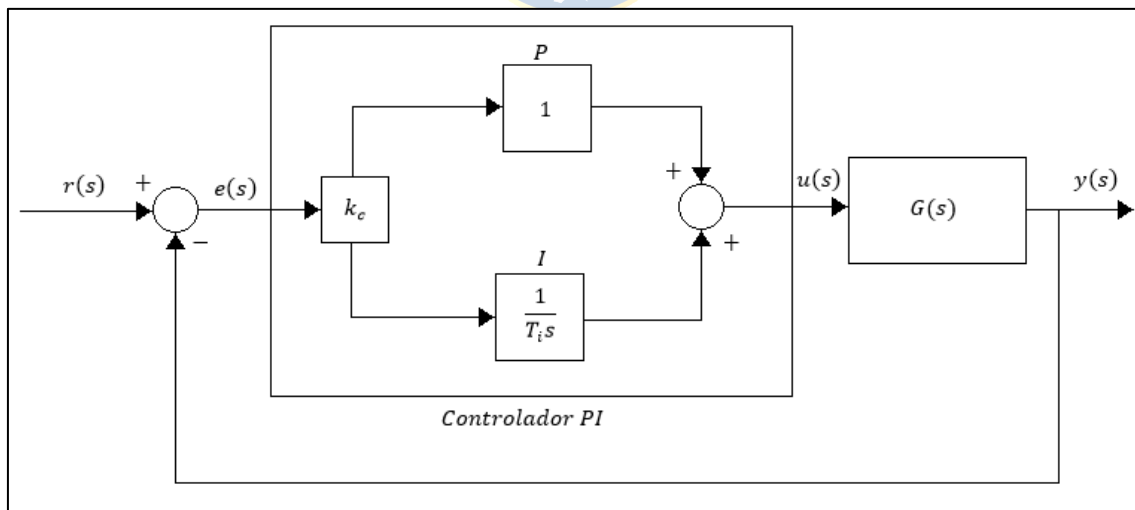


Figura 3.3. Diagrama de controlador PI

Un controlador PI se define de forma continua como muestran (3.1). Su transformada de Laplace se muestra en (3.2).

$$u(t) = k_c \left( e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau \right) \quad (3.1)$$

$$u(s) = k_c \left( 1 + \frac{1}{T_i s} \right) e(s) \quad (3.2)$$

Donde  $k_c$  es la constante proporcional,  $T_i$  es la constante integrativa,  $e(t)$  es el error, esto es la diferencia entre la salida  $y(t)$  y la referencia  $r(t)$ , y  $u(t)$  es la entrada. La forma discreta del controlador se muestra en (3.3) y (3.4). En el Anexo A se detalla la obtención de la forma discreta del controlador.

$$u_k = u_{k-1} + \Delta u_k \quad (3.3)$$

$$\Delta u_k = k_c \left( 1 + \frac{T_m}{2T_i} \right) e_k + k_c \left( -1 + \frac{T_m}{2T_i} \right) e_{k-1} \quad (3.4)$$

### 3.3.2. Diseño del controlador

El principal criterio utilizado en el diseño de este controlador es evitar sobrepasos, dejando en segundo plano la rapidez de la respuesta. Esto debido a que, en algunos casos, un sobrepaso grande puede significar un rebalse del microcanal.

En el Capítulo 2 se determinó que el modelo a utilizar sería el modelo no lineal fenomenológico, pero para diseñar el controlador PI es necesario un modelo lineal del proceso. Si bien se podría linealizar el modelo no lineal en torno a un punto de operación, se escoge realizar aproximaciones lineales de primer orden con retardo para distintas zonas de operación, esto debido a que el comportamiento del microcanal es similar a un proceso de primer orden con retardo cuando se trabaja dentro de ciertos límites.

A partir del modelo no lineal (representativo del microcanal), y utilizando mínimos cuadrados, se determinaron los parámetros para las aproximaciones de primer orden mencionadas anteriormente, y a partir de estos se determinaron los parámetros del controlador PI utilizando el

método de sintonización en lazo abierto de Ziegler-Nichols (también llamado método de curva de reacción de proceso) mostrado en la Tabla 3.1, tal como se indica en [14]. Este método entrega respuestas sin sobrepaso, pero más lentas que las entregadas por otros métodos de sintonización. Este método no se debe confundir con el método de Ziegler-Nichols en lazo cerrado (también llamado método de oscilación de Ziegler-Nichols), el cual se caracteriza por entregar respuestas con grandes sobrepasos.

**Tabla 3.1. Método de sintonización de Ziegler-Nichols en lazo abierto**

Tipo de controlador	$k_c$	$T_i$	$T_d$
<b>P</b>	$\frac{\tau}{k\theta}$		
<b>PI</b>	$0.9 \frac{\tau}{k\theta}$	$\frac{\theta}{0.3}$	
<b>PID</b>	$1.2 \frac{\tau}{k\theta}$	$2\theta$	$0.5\theta$

Los parámetros tanto de proceso como de controlador se muestran en la Tabla 3.2. Notar que los parámetros para ambas piscinas son los mismos, esto debido a que el comportamiento en ambas piscinas es prácticamente igual. Se considera el mismo retardo en todos los casos, debido a que la diferencia es ínfima entre ellos. Por último es importante mencionar que se utiliza un planificador de ganancia (Gain Scheduler), el cual determina la ganancia del controlador por medio de interpolación a fin que no se produzcan cambios bruscos de actuación.

**Tabla 3.2. Parámetros Controlador PI**

Rango compuerta	Ganancia de proceso	Tiempo de asentamiento	Retardo	Ganancia controlador	Tiempo integrativo
0%-2%	-7.578	81.48	20.0	0.9677	66.6
2%-4%	-4.767	66.79	20.0	1.2609	66.6
4%-7%	-2.937	53.80	20.0	1.6489	66.6
7%-10%	-1.774	45.228	20.0	2.2939	66.6
10%-15%	-1.027	37.38	20.0	3.277	66.6
15%-20%	-0.572	32.50	20.0	5.110	66.6
20%-25%	-0.351	29.48	20.0	7.551	66.6
25%-40%	-0.169	25.34	20.0	13.5102	66.6
40%-55%	-0.0676	22.99	20.0	30.592	66.6
55%-100%	-0.0216	19.99	20.0	83.3007	66.6



Es importante notar que las ganancias de proceso son negativas debido a que el esquema de control utilizado es aguas arriba. Esto quiere decir que la entrada del proceso (y variable de control) es la compuerta de salida de cada piscina.

### 3.4. Controlador de lógica difusa

#### 3.4.1. Introducción

Un controlador de lógica difusa (FLC: Fuzzy Logic Controller) representa acciones de control de tipo experta, es decir, que a diferencia de los controladores del tipo adaptativo-predictivo, conoce el proceso de forma previa. Esta es la principal ventaja del control difuso frente a otras estrategias de control. Una segunda gran ventaja es la robustez de este tipo de controlador en procesos no lineales.

La estructura básica de un controlador de lógica difusa consta principalmente de cuatro bloques esenciales: interfase fusificadora, base de conocimiento, lógica de toma de decisión e interfase defusificadora. La Figura 3.4 muestra el diagrama básico de un controlador de lógica difusa.

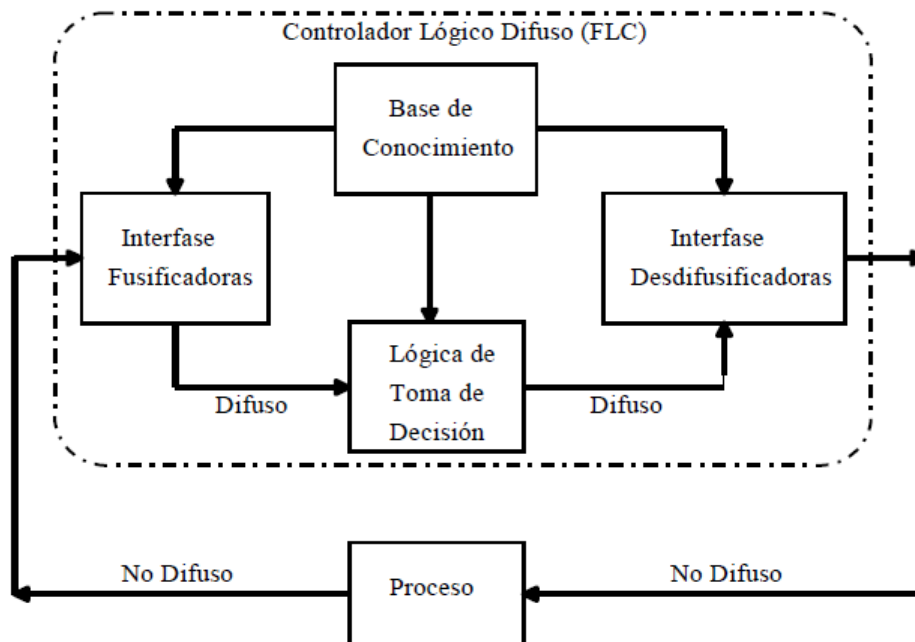


Figura 3.4. Controlador de lógica difusa

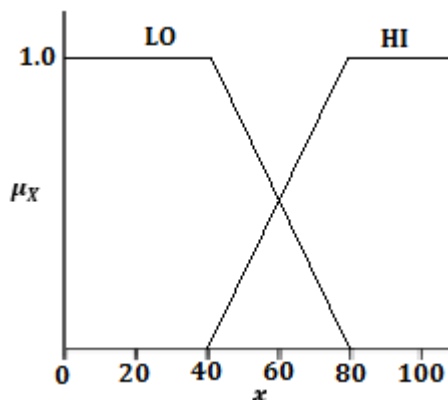
La función de cada uno de estos bloques se explica a continuación.

### 3.4.1.1. Interfase Fusificadora

Las funciones de la interfase fusificadora son las siguientes:

- Medir el valor de las variables de entrada.
- Escalar las variables de entrada dentro del universo discurso.
- Desarrollar la función fusificadora para convertir un dato de entrada en una variable lingüística, con el fin de hacerlo compatible con las funciones de pertenencia de las variables de estado dadas por la base de conocimiento.

Por ejemplo, si se considera un universo discurso como el mostrado en la Figura 3.5 para una variable de estado  $x$ , la fusificación para distintos valores de dicha variable es la mostrada en la Tabla 3.3. La representación de las variables de entrada fusificadas se obtiene a partir de las funciones de pertenencia  $\mu_X^{LO}(x)$  y  $\mu_X^{HI}(x)$ .



**Tabla 3.3. Ejemplo Fusificación**

$x$	$\mu_X^{LO}(x)$	$\mu_X^{HI}(x)$
20	1	0
60	0.5	0.5
90	0	1

**Figura 3.5. Ejemplo Universo Discurso Fusificación**

### 3.4.1.2. Base de conocimiento

La base de conocimiento consiste en una base de datos y una base de reglas de control lingüísticas (difusas). La base de datos provee las definiciones necesarias, las cuales son usadas para definir las reglas de control lingüístico y la manipulación de los datos difusos en el controlador. La

base de reglas caracteriza el objeto de control y el curso del control en el dominio del experto a través de un conjunto de reglas de control lingüísticas.

La Figura 3.6 muestra un ejemplo de base de conocimiento en forma matricial, en donde  $e$  es el error,  $\Delta e$  la derivada del error, N, ZE y P indican variable fusificada ‘Negativa’, ‘Zero’ (o ‘Nula’) y ‘Positiva’, respectivamente, y D, M y A indican acción de control ‘Disminuir’, ‘Mantener’ y ‘Aumentar’, respectivamente.

$\Delta e \backslash e$	N	ZE	P
N	D	D	M
ZE	D	M	A
P	M	A	A

**Figura 3.6. Ejemplo Base de conocimiento**

En el ejemplo anterior, cada elemento de la matriz se considera una regla. Cada una de estas reglas se puede escribir de forma lingüística. Por ejemplo, la regla dada por la intersección de la primera fila con la primera columna puede definirse como ‘Regla 1’ y puede escribirse como:

$$R_1 : \text{Si } e_k \text{ es N y } \Delta e_k \text{ es N entonces } \langle \text{acción de control} \rangle \text{ es D}$$

La representación fusificada de la Regla 1 se escribe como:

$$\mu_{R_1} = \min(\mu_{e_k}^N, \mu_{\Delta e_k}^N) \quad (3.5)$$

Nótese que en la expresión anterior se define el valor fusificado de la regla como el mínimo entre el error fusificado y la derivada del error fusificada. Esto es porque la unión entre estas variables corresponde al mínimo valor entre ellas.

### 3.4.1.3. Lógica de toma de decisión

La lógica de toma de decisión, también llamada ‘Motor de Inferencia’, corresponde al corazón de un controlador de lógica difusa. Tiene la capacidad de simular el comportamiento de

toma de decisión humana basándose en conceptos difusos, además de poder inferir acciones de control difuso utilizando implicaciones difusas. La toma de decisión se realiza en base a un conjunto de reglas heurísticas (*if-else*) basadas en conocimiento experto, las cuales son definidas por la base de conocimiento.

Usando el ejemplo de la sección anterior, las reglas que definen la acción de control '*Disminuir*' se escriben de forma lingüística como:

$R_1$  : Si  $e_k$  es N y  $\Delta e_k$  es N entonces <acción de control> es D

$R_2$  : Si  $e_k$  es ZE y  $\Delta e_k$  es N entonces <acción de control> es D

$R_4$  : Si  $e_k$  es N y  $\Delta e_k$  es ZE entonces <acción de control> es D

Luego, la representación fusificada de la acción de control '*Disminuir*' se escribe como:

$$\mu_D = \max(\mu_{R_1}, \mu_{R_2}, \mu_{R_4}) \quad (3.6)$$

Nótese que en la expresión anterior se define el valor fusificado de la acción de control como el máximo entre las reglas fusificadas. Esto es porque se está considerando la intersección entre las reglas, lo que corresponde al máximo valor entre ellas.

#### 3.4.1.4. Interfase defusificadora

Las funciones de la interfase defusificadora son las siguientes:

- Convertir las variables difusas en un valor escalar.
- Escalamiento de las variables de salida dentro del universo discurso.

Existen varios métodos para encontrar el valor escalar en el proceso de defusificación, de los cuales son tres los más utilizados:

- **Criterio de los Máximos:** corresponde al punto donde se encuentra el máximo absoluto entre las acciones de control.
- **Promedio de los Máximos:** es el promedio de todas las acciones de control máximas. En el caso discreto se expresa de la forma mostrada en (3.7).

$$z_0 = \sum_{j=1}^l \frac{w_j}{l} \quad (3.7)$$

Donde  $w_j$  es el valor en que la función de pertenencia  $u_z(w_z)$  alcanza el valor máximo y  $l$  es la cantidad de valores en que se encuentra el máximo.

- **Centro de Área o Centroide:** es el método comúnmente más usado. Este método ubica el centro de gravedad de la distribución de la acción de control; también se puede interpretar como el promedio ponderado. Para el caso de un universo discurso se tiene lo mostrado en (3.8).

$$z_0 = \frac{\sum_{j=1}^n u_z(w_j)w_j}{\sum_{j=1}^n u_z(w_j)} \quad (3.8)$$

Donde  $n$  es el número de discretizaciones del universo discurso de salida.

### 3.4.1.5. Controlador de lógica difusa de tipo PI

La ecuación que rige un controlador PI convencional es la mostrada en (3.1). Si se toma la derivada con respecto al tiempo de dicha expresión se obtiene la expresión equivalente mostrada en (3.9).

$$\frac{d}{dt}u = k_c \left( \frac{d}{dt}e + \frac{1}{T_i}e \right) \quad (3.9)$$

Las reglas para un controlador de lógica difusa de tipo PI son de la forma

$R_i$ : Si  $e$  es  $\langle Propiedad \rangle$  y  $\Delta e$  es  $\langle Propiedad \rangle$  entonces  $\Delta u$  es  $\langle Propiedad \rangle$ .

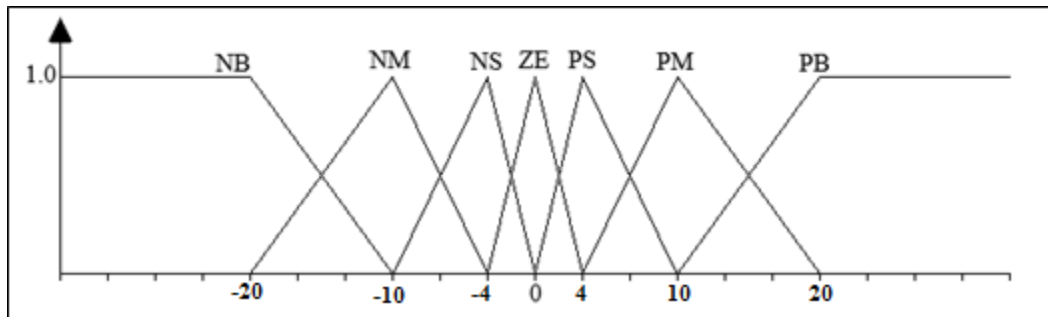
De este modo, para obtener el valor de la variable de control  $u_k$ , la variable de salida  $\Delta u_k$  (la cual está definida en función de  $e$  y  $\Delta e$ ) es añadida a  $u_{k-1}$ . Esto se puede escribir como:

$$u_k = u_{k-1} + f(e, \Delta e) \quad (3.10)$$

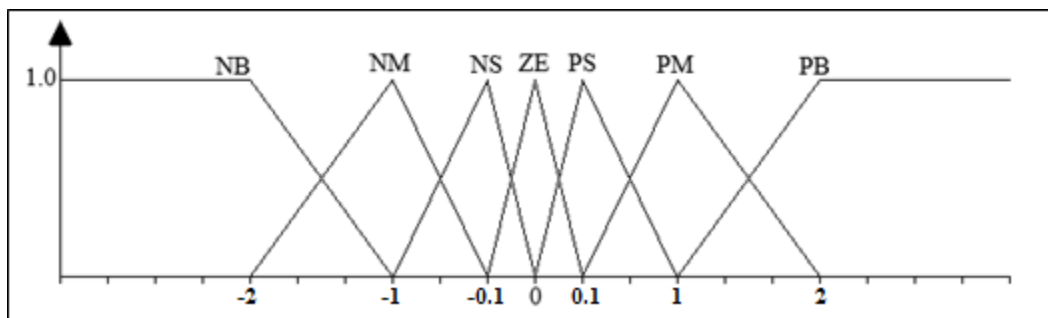
### 3.4.2. Diseño del controlador

El diseño del controlador se llevó a cabo analizando la respuesta del proceso en lazo abierto a fin de determinar tanto la base de conocimiento como las funciones de pertenencia (fusificación y defusificación) bajo el criterio de evitar sobrepasos, sin embargo, y a diferencia del control PI convencional, el control difuso permite tener una respuesta rápida incluso bajo la exigencia de evitar sobrepasos, pues se basa en el conocimiento previo que se tiene del proceso.

Para la interfase fusificadora se eligieron funciones de pertenencia distintas para el error de nivel y para la derivada del error. La Figura 3.7 y la Figura 3.8 muestran la distribución del universo discurso de las funciones de pertenencia.



**Figura 3.7. Universo Discurso Fusificación Error de Nivel**



**Figura 3.8. Universo Discurso Fusificación Derivada del Error**

Las funciones de pertenencia del error de nivel se escogieron de forma que se tomen acciones de control grandes si el error absoluto es grande o medio, y pequeñas si el error absoluto es pequeño. Las funciones de pertenencia de la derivada del error se escogieron de forma que el nivel

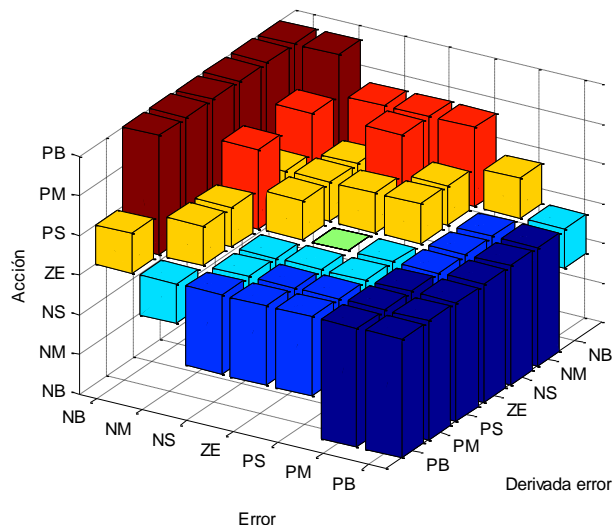
se acerque rápidamente a la referencia cuando el error grande o medio, pero disminuya su velocidad considerablemente cuando esté por alcanzarla. Los rangos de error y derivada del error de cada función de pertenencia se escogieron, como se menciona al inicio de esta sección, en base al análisis de la respuesta del proceso en lazo abierto (principalmente mediante simulación); primero se determinaron las funciones de pertenencia de la derivada del error, ya que estas definen las “velocidades” del nivel para los distintos casos, y luego se definieron las funciones de pertenencia del error, ya que estas indican en que rangos usar las “velocidades” antes mencionadas (en particular, el rango de error absoluto más pequeño para que no ocurra sobrepaso con la “velocidad” más baja). Es importante mencionar que, debido a que el control difuso se basa en el conocimiento empírico del proceso, tanto la cantidad de funciones de pertenencia como los rangos de cada una de ellas pueden variar dependiendo del criterio de cada diseñador.

Para la toma de decisiones se optó por una base de conocimiento basada en la acción de un controlador PI, pero con una variante. Las acciones a tomar alrededor de un error pequeño y/o una derivada del error pequeña sean predictivas, de manera que el desempeño del controlador sea adecuado. De esta forma se obtiene una base de conocimiento que toma acciones grandes cuando el error es grande, y acciones predictivas cuando el error es menor, lo cual indica que la variable del proceso está por llegar al valor deseado. Otro factor que se consideró al diseñar el motor de inferencia es que se debe evitar sobrepasos positivos por sobre los negativos, por lo que el sistema reacciona de forma más cautelosa ante un error positivo, y de forma más agresiva ante error negativo.

La base de conocimiento para la toma de decisiones se muestra en la Figura 3.9. La distribución de la base de conocimiento se muestra en la Figura 3.10.

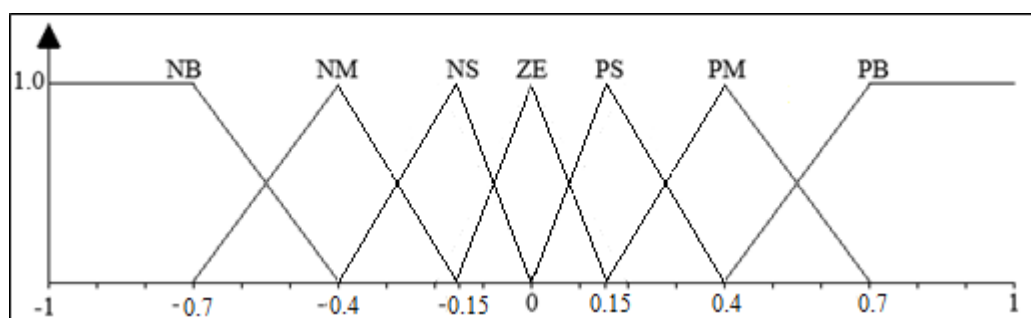
$\Delta E \backslash E$	NB	NM	NS	ZE	PS	PM	PB
NB	PB	PB	PM	PM	PM	PS	NS
NM	PB	PM	PS	PM	PS	NM	NB
NS	PB	PS	PS	PS	PS	NM	NB
ZE	PB	PM	PS	ZE	NS	NM	NB
PS	PB	PS	NS	NS	NS	NB	NB
PM	PB	PS	NS	NM	NM	NB	NB
PB	PS	NS	NM	NM	NM	NB	NB

**Figura 3.9. Base de conocimiento**



**Figura 3.10. Distribución base de conocimiento**

Finalmente las funciones de pertenencia para la defusificación se muestran en la Figura 3.11.



**Figura 3.11. Universo Discurso para Defusificación**

El universo discurso de defusificación se definió considerando que mientras mayor es la acción a tomar, el incremento (o decremento) que se suma a la acción de control es mucho mayor. Esto se debe a que la apertura de una compuerta es inversamente proporcional a la raíz de la diferencia entre el nivel aguas arriba y el nivel aguas abajo (ver Capítulo 2, sección 2.3.2). Por último, la variación de la acción de control se determina mediante el método del centroide.

Es importante mencionar que, si bien se determinaron las reglas de acuerdo al criterio mencionado al comienzo de esta sección (evitar sobrepasos), se hicieron ajustes en base a ensayo y error en simulación, obteniendo así un motor de inferencia más refinado.



## Capítulo 4. Simulación de controladores

### 4.1. Introducción

En este capítulo se muestran los resultados de las simulaciones de ambos controladores y se compara el desempeño de ambos. En la simulación realizada con cada uno se utilizaron referencias de tipo escalón para ambas piscinas: escalón de 45% a 65% a los 5 minutos para la primera piscina y escalón de 40% a 50% a los 10 minutos para la segunda piscina. . Las simulaciones fueron realizadas sobre Matlab R2014a (códigos en Anexo C).

### 4.2. Controlador PI

Los resultados de la simulación se muestran en la Figura 2.1.

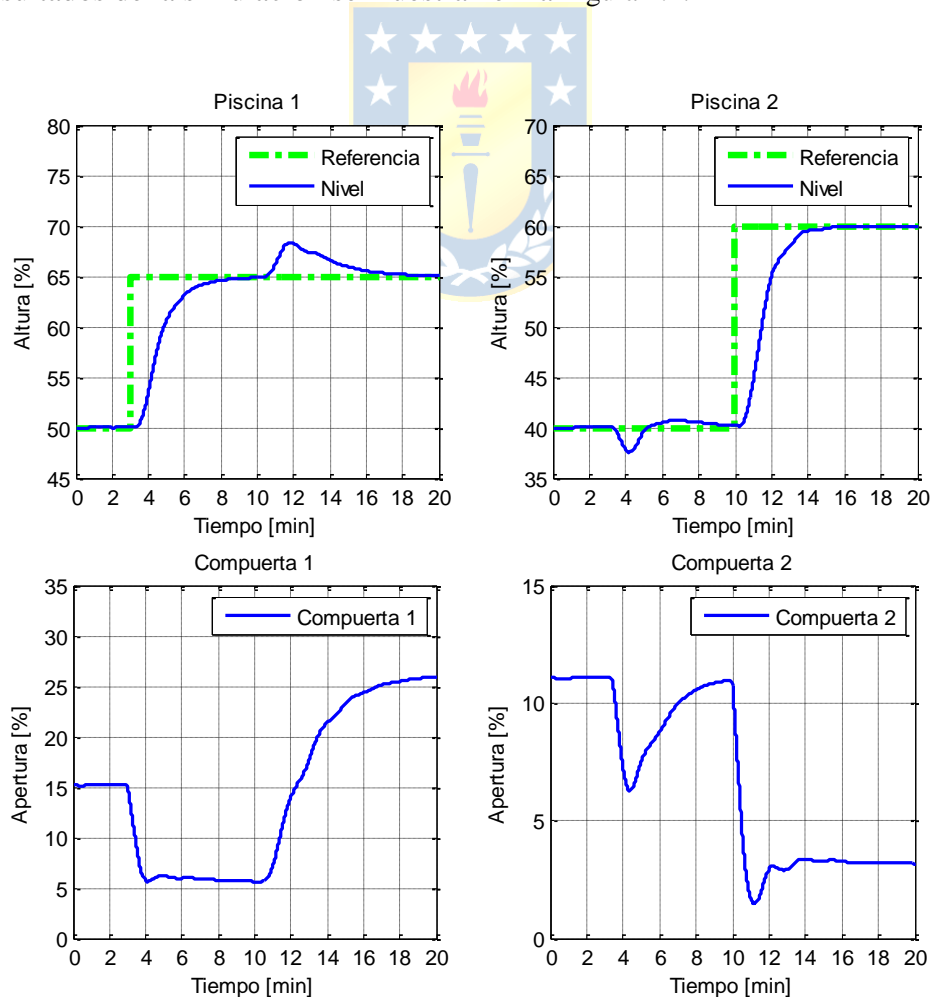
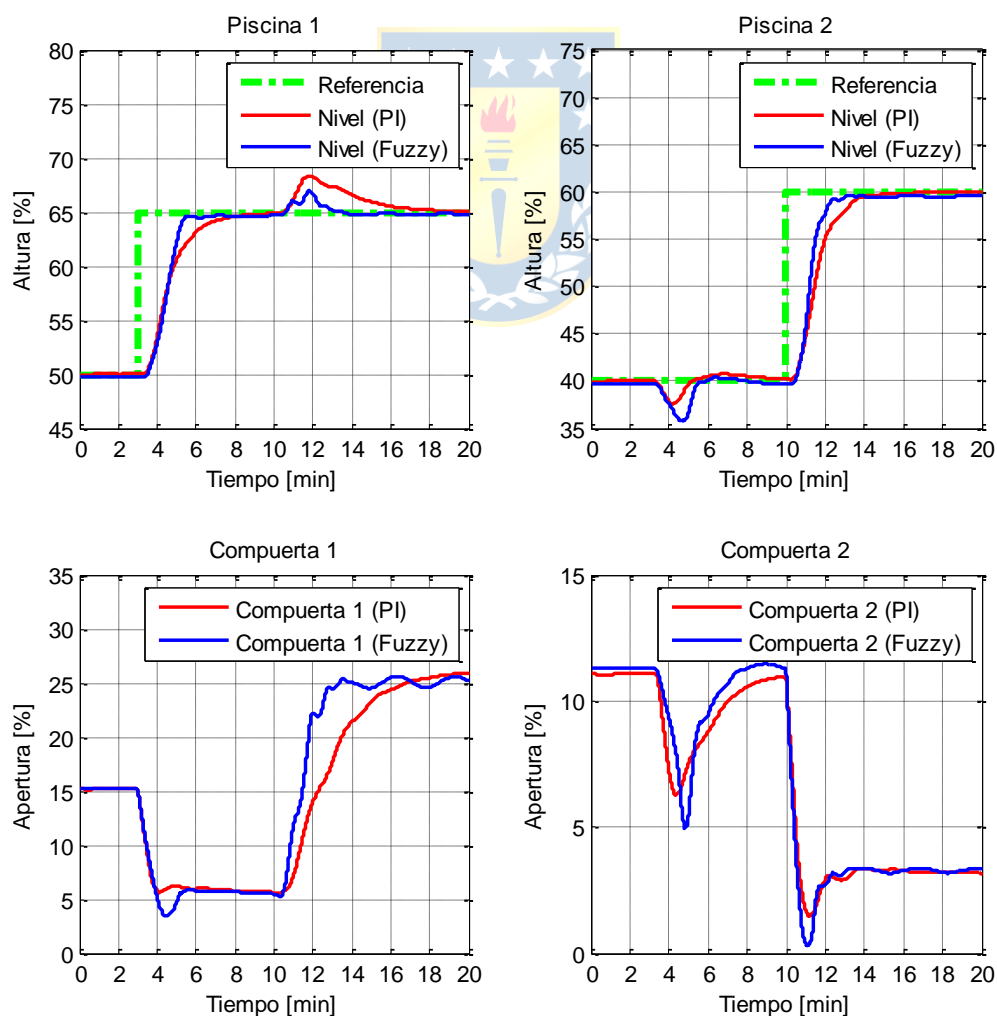


Figura 4.1. Simulación Controlador PI

Los resultados obtenidos de la simulación muestran que, ante un cambio de referencia, el controlador es capaz de hacer que el sistema alcance la referencia deseada sin producir sobrepaso. No obstante, es importante mencionar que al hacer un cambio de referencia en alguna de las piscinas se produce cierto sobrepaso en la otra, ya que un cambio de nivel en alguna de las piscinas produce un cambio en el flujo bajo la compuerta 1. Sin embargo, si bien el controlador no es capaz de predecir las perturbaciones producidas desde una piscina a otra, si es capaz de mitigar este efecto, disminuyendo la duración de estos sobrepasos.

### 4.3. Controlador de lógica difusa

Los resultados de la simulación se muestran en la Figura 4.2.



**Figura 4.2. Simulación Controlador Fuzzy**

Analizando la simulación del controlador difuso se puede apreciar una disminución importante tanto en los tiempos de asentamiento como en los sobrepasos. También es importante notar que el sobrepaso positivo producido en una piscina por cambios en la otra es menor con el controlador difuso, mientras que el negativo es mayor. Esto se debe a que, como se menciona en el Capítulo 3, cuando el error es negativo las acciones de control tomadas por el controlador difuso son más agresivas en comparación a las tomadas por el controlador PI.

#### **4.4. Conclusiones**

De las simulaciones realizadas se puede concluir que el controlador difuso presenta un mejor desempeño que el controlador PI; se redujo el tiempo de asentamiento de la piscina 1 en un 50% y el de la piscina 2 en un 25%. Esto se debe a los criterios de diseño utilizados en el controlador difuso (lo que derivó en la obtención de las reglas elegidas), permitiendo que la atenuación de la acción de control se realice más cerca de la referencia y de forma más agresiva.

Es importante notar que al utilizar el controlador de lógica difusa la perturbación en la piscina 2 debido a la piscina 1 es mayor debido a que, como se menciona en el Capítulo 3, el criterio de diseño es evitar sobrepasos, principalmente sobrepasos positivos por sobre los negativos. De esta forma, al haber un sobrepaso positivo en la piscina 1, la acción de control (compuerta 1) es más agresiva, mientras que ante el sobrepaso negativo de la piscina 2, la acción de control (compuerta 2) es menos agresiva.

## Capítulo 5. Implementación de controladores

### 5.1. Introducción

En este capítulo se muestran los resultados de la implementación del controlador de lógica difusa y del controlador PI en la planta microcanal. Las referencias utilizadas son las mismas que se utilizaron en las simulaciones del Capítulo 4. La implementación en el PLC se realizó mediante el software RsLogix 5000 v16.03.

### 5.2. Controlador PI

La Figura 5.1 muestra los resultados obtenidos de la implementación del controlador PI en la planta microcanal.

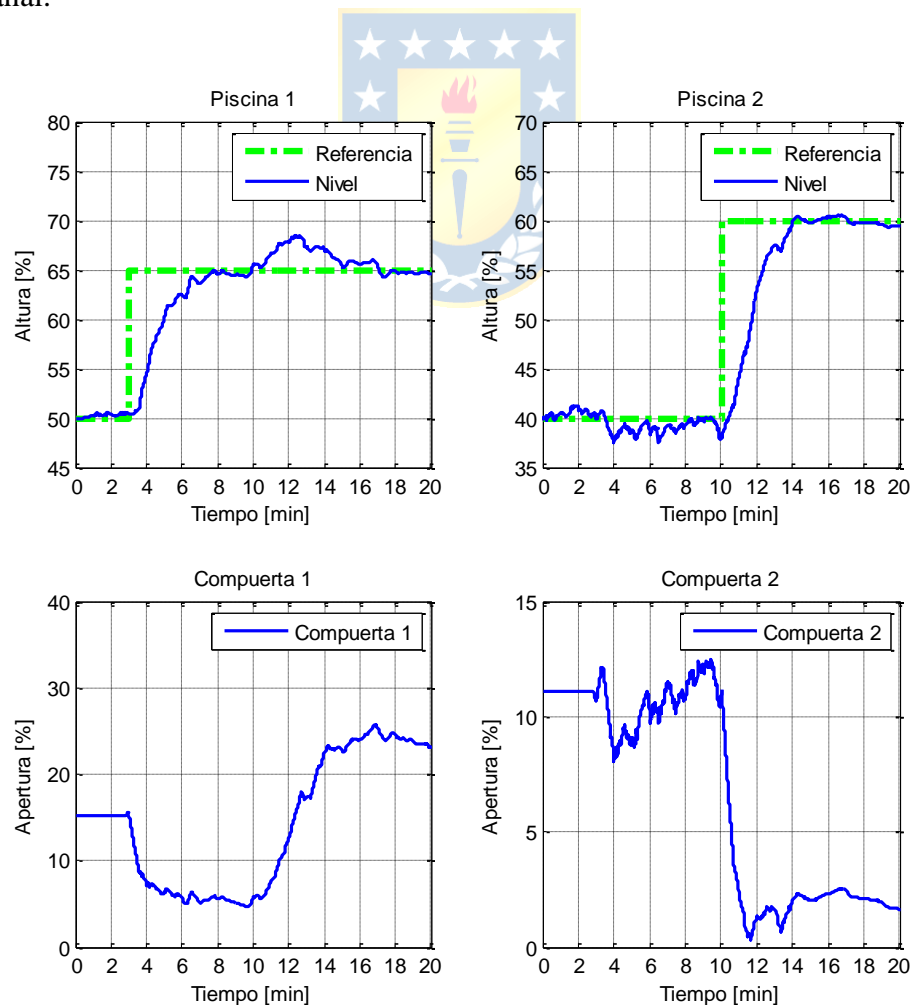


Figura 5.1. Respuesta escalón (controlador PI)

Se puede ver que, al igual que los resultados obtenidos mediante simulación, ante un cambio de referencia en una de las piscinas el controlador es capaz de seguir la referencia deseada sin producir sobrepaso, pero no es capaz de evitar el sobrepaso producido en la otra piscina. La principal diferencia entre simulación e implementación es la presencia de ruido, el cual afecta a la variable de control de manera directa. Sin embargo, se puede observar que a pesar de esto el controlador es capaz de mantener las variables de salida del sistema en su debida referencia. También es importante mencionar que el sobrepaso negativo en la segunda piscina es menor que en la simulación, esto se atribuye principalmente a que el modelo usado no considera perturbaciones presentes en el canal, tales como los soportes inferiores de las compuertas.

### 5.3. Controlador de lógica difusa

La Figura 5.2 muestran los resultados obtenidos de la implementación del controlador difuso en la planta microcanal.

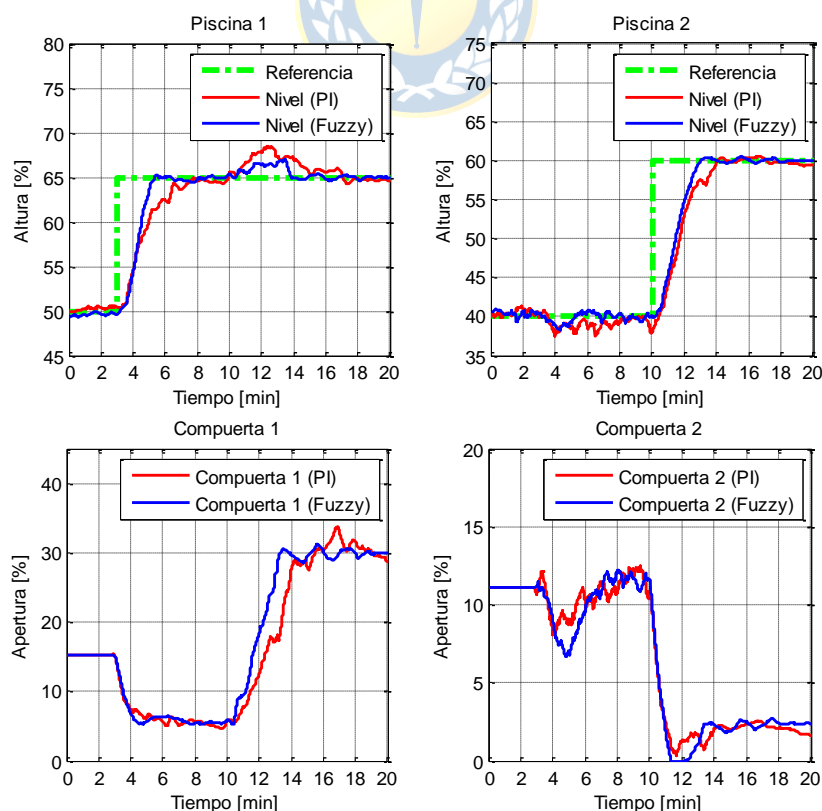


Figura 5.2. Respuesta escalón (controlador fuzzy)

De forma similar al caso del controlador PI, se puede observar que el controlador es capaz de mantener el nivel de cada piscina en su respectiva referencia, a pesar del ruido presente. Los resultados obtenidos se condicen con los resultados obtenidos en las simulaciones, por lo que se cumple con los criterios de diseño mencionados en el Capítulo 3. Se puede ver como el controlador difuso hace que el nivel en ambas piscinas alcance la referencia deseada de forma más rápida que el controlador PI. Sin embargo, y a diferencia de las simulaciones, se observa un leve sobrepaso en el caso de la primera piscina al utilizar el controlador difuso. También se puede observar que la compuerta 2 se satura durante un breve periodo de tiempo. Ambas situaciones pueden deberse tanto a perturbaciones como a pequeñas diferencias entre el modelo y la planta.

#### **5.4. Conclusiones**

Se cumplió con el objetivo propuesto, ya que se logró controlar ambas piscinas mediante control PI y control difuso. Se puede ver que el modelo definido en el Capítulo 2 (el cual constituye un modelo no lineal multivariable) y usado en las simulaciones del Capítulo 4 se ajusta bastante a lo observado al implementar los controladores en la planta microcanal. Se observó una evidente disminución en el tiempo de asentamiento al pasar de control PI convencional a control fuzzy, específicamente se redujo el tiempo de asentamiento de la primera piscina en un 60% y el de la segunda piscina en un 25% (lo cual se condice con lo obtenido en las simulaciones realizadas en el Capítulo 4). Esto se debe a que las reglas definidas en el motor de inferencia permiten que el nivel alcance la referencia más rápido y que el sistema se estabilice de forma más agresiva cuando el error es pequeño. El sobrepaso observado en la piscina 1 no es significativo, pero podría evitarse con el uso de un modelo no lineal más preciso, esto es, que incluya perturbaciones no consideradas en el modelo utilizado en este trabajo. Se puede observar saturación en la compuerta 2, pero debido a su duración no es significativa, sin embargo, puede ser evitada mediante el uso de alguna herramienta que mitigue este tipo de situaciones, por ejemplo, algún mecanismo anti wind-up.

## Capítulo 6. Conclusiones

---

### 6.1. Sumario

Se realizó una revisión bibliográfica enfocada en la modelación del proceso, estrategias de control PI y estrategias de control difuso.

Se revisaron modelos consistentes con la revisión bibliográfica realizada, con el fin de determinar el modelo que mejor se ajustaba a la necesidad de este trabajo. Se identificó un modelo no lineal para dos piscinas y se realizaron simulaciones para verificar dicho modelo. Se identificaron modelos lineales de primer orden para la sintonización del controlador PI.

Se diseñó y simuló un controlador de lógica difusa, al igual que un controlador PI, para el microcanal. Se contrastaron los resultados obtenidos entre ambos. Se implementaron los controladores al microcanal y se contrastaron los datos reales.



### 6.2. Conclusiones

- El modelo no lineal basado en estanques representa las dinámicas del canal, por lo que es válida su utilización en el diseño y simulación de controladores. En particular, permite sintonizar de manera correcta controladores de lógica difusa.
- Para la sintonización de controladores PI es recomendable la utilización de un modelo de primer orden con retardo para la zona en la cual se trabajará, ya que permite obtener parámetros que se ajustan a dicha necesidad.
- Es posible implementar un controlador PI al microcanal, pero no es recomendable debido a la cadena de perturbaciones que se produce aguas abajo por los posibles sobrepasos de nivel producidos. Un controlador PI que evite sobrepasos, como el utilizado en este trabajo, lleva a un control más lento comparado a otras estrategias de control, como el control difuso.

- La implementación de un controlador de lógica difusa representa una mejora con respecto a un controlador PI. Esto se puede ver en la reducción del tiempo de asentamiento de la respuesta del sistema.
- Por último, se puede concluir que un controlador de lógica puede ser tanto o más efectivo que un controlador PI convencional, sin la necesidad de ser más complejo. De hecho, se puede considerar que un controlador PI es uno de los posibles casos de un controlador fuzzy, no así a la inversa.

### 6.3. Trabajo Futuro

- En este trabajo se diseñó un controlador de lógica difusa de tipo PI con el fin de comparar su desempeño frente a un controlador PI convencional, sin embargo, no se realizaron comparaciones frente a otras estrategias de control. El controlador difuso presentado puede ser utilizado como punto de comparación para futuros trabajos con enfoque en otra estrategia de control.
- Es posible mejorar el desempeño del controlador fuzzy mediante alguna estrategia de tipo predictor, por ejemplo, la utilización de predictor Smith. Esto puede ser contrastado con trabajos anteriores de control PI con predictor Smith sobre el microcanal.
- Si bien el modelo no lineal basado en estanques representa las dinámicas presentes en el canal, no es perfecto, por lo que a futuro se espera el desarrollo de un modelo no lineal basado en las ecuaciones de Saint-Venant a fin de realizar simulaciones aún más exactas de este y futuros controladores.
- Se espera a futuro la implementación de actuación para la tercera compuerta, así como de un sensor para la tercera piscina, con el fin de realizar control sobre las tres primeras piscinas del microcanal.



## Bibliografía

- [1] Xavier Líttrico and Vincent Fromion, *Modeling and Control of Hydrosystems*. 1st Edition, Springer, 2009.
- [2] Bruno Moraga Ramírez, “Modelación de Microcanal para Control y Estimación”, Memoria de Título para optar al título de Ingeniero Civil Electrónico, Marzo 2014, Departamento de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de Concepción.
- [3] Carlos Sepúlveda Toepfer, “Modelación Dinámica y Control con Restricciones de Sistemas de Canales de Regadío”, Memoria de Título para optar al título de Ingeniero Civil Electrónico, Abril 2002, Departamento de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de Concepción.
- [4] Dennis Calvo Salinas, “Estrategias de Control SISO para una Red de Canales de Regadío”, Memoria de Título para optar al título de Ingeniero Civil Electrónico, Abril 2001, Departamento de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de Concepción.
- [5] Wagner Ramírez Ortega, “Control en Tiempo Continuo para Planta Microcanal”, Memoria de Título para optar al título de Ingeniero Civil Electrónico, Agosto 2013, Departamento de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de Concepción.
- [6] Karl J. Astrom and Tore Hagglund, *PID Controllers: Theory, Design and Tuning*. 2nd Edition, ISA, 1995.
- [7] D. Driankov, H. Hellendoorn and M. Reinfrank, *An Introduction to Fuzzy Control*. 2nd Edition, Springer, 2010.
- [8] Michio Sugeno and Takahiro Yasukawa. “A Fuzzy-Logic-Based Approach to Qualitative Modeling”, *IEEE Transactions on Fuzzy Systems*, vol 1, no 1, pp. 7-31, Feb. 1993.

- [9] Hua O. Wang, Kazuo Tanaka and Michael F. Griffin. “An Approach to Fuzzy Control of Nonlinear Systems: Stability and Design Issues”, *IEEE Transactions on Fuzzy Systems*, vol 4, no 1, pp. 14-23, Feb. 2006.
- [10] M. Santos, S. Dormido and J.M. de la Cruz. “Fuzzy-PID Controllers vs. Fuzzy-PI Controllers”, *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems*, vol 3, pp. 1598-1604, September 1996.
- [11] Rames C. Panda, *Introduction to PID Controllers – Theory, Tuning and Application to Frontier Areas*. 1st Edition, Intech, 2012.
- [12] A. Lacasta, M. Morales-Hernández, P. Brufau, P. García-Navarro. “Simulation of PID control applied to irrigation channels”, *Procedia Engineering*, vol 70, pp. 978-987, 2014.
- [13] Petra Bernhoff, “System identification and control of an irrigation channel with a tunnel”, Tesis, 2009, Teknisk-naturvetenskaplig fakultet, UTH-enheten, Uppsala Universitet.
- [14] M. Kurien, A. Prayagkar and V. Rajeshirke. “Overview of Different Approaches of PID Controller Tuning”, *International Journal of Research in Advent Technology*, vol 2, issue 1, pp. 167-175, January 2014.
- [15] Ignacio Hernández Pino, “Implementación de Soluciones de Control para Planta Microcanal”, Memoria de Título para optar al título de Ingeniero Civil Electrónico, Marzo 2014, Departamento de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de Concepción.

## Anexo A. Desarrollo de ecuaciones

---

### A.1. Modelo de primer orden con retardo

#### A.1.1 Transformada de Laplace a Transformada Z

Sea la transformada de Laplace de un sistema de primer orden con retardo con retentor de orden cero:

$$\frac{y(s)}{u(s)} = \frac{(1 - e^{-Tm})}{s} \cdot \frac{k}{(\tau s + 1)} e^{-\theta s} \quad (\text{A. 1})$$

Se aplica transformada Z a ambos lados de la ecuación:

$$Z \left\{ \frac{y(s)}{u(s)} \right\} = Z \left\{ \frac{(1 - e^{-Tm})}{s} \cdot \frac{k}{(\tau s + 1)} e^{-\theta s} \right\} \quad (\text{A. 2})$$

$$\frac{y(z)}{u(z)} = Z \left\{ \frac{(1 - e^{-Tm})}{s} \cdot \frac{k}{(\tau s + 1)} e^{-\theta s} \right\} \quad (\text{A. 3})$$

Desarrollando el término de la derecha:

$$\frac{y(z)}{u(z)} = k \cdot Z \left\{ \left( \frac{e^{-\theta s}}{s(\tau s + 1)} \right) - \left( \frac{e^{-Tm} \cdot e^{-\theta s}}{s(\tau s + 1)} \right) \right\} \quad (\text{A. 4})$$

$$\frac{y(z)}{u(z)} = k \left( Z \left\{ \left( \frac{e^{-\theta s}}{s(\tau s + 1)} \right) \right\} - Z \left\{ \left( \frac{e^{-Tm} \cdot e^{-\theta s}}{s(\tau s + 1)} \right) \right\} \right) \quad (\text{A. 5})$$

$$\frac{y(z)}{u(z)} = k \left( z^{-\frac{\theta}{Tm}} Z \left\{ \left( \frac{1}{s(\tau s + 1)} \right) \right\} - z^{-\left(\frac{\theta}{Tm} + 1\right)} Z \left\{ \left( \frac{1}{s(\tau s + 1)} \right) \right\} \right) \quad (\text{A. 6})$$

$$\frac{y(z)}{u(z)} = k \left( (1 - z^{-1}) z^{-\frac{\theta}{Tm}} Z \left\{ \left( \frac{1}{s(\tau s + 1)} \right) \right\} \right) \quad (\text{A. 7})$$

Luego, separando en fracciones parciales y desarrollando se tiene:

$$\frac{y(z)}{u(z)} = k \left( (1 - z^{-1}) z^{-\frac{\theta}{T_m}} Z \left\{ \left( \frac{1}{s} - \frac{\tau}{(\tau s + 1)} \right) \right\} \right) \quad (\text{A. 8})$$

$$\frac{y(z)}{u(z)} = k \left( (1 - z^{-1}) z^{-\frac{\theta}{T_m}} \left( Z \left\{ \left( \frac{1}{s} \right) \right\} - Z \left\{ \left( \frac{1}{\left( s + \frac{1}{\tau} \right)} \right) \right\} \right) \right) \quad (\text{A. 9})$$

$$\frac{y(z)}{u(z)} = k \left( (1 - z^{-1}) z^{-\frac{\theta}{T_m}} \left( \frac{1}{1 - z^{-1}} - \frac{1}{1 - z^{-1} e^{-\frac{T_m}{\tau}}} \right) \right) \quad (\text{A. 10})$$

$$\frac{y(z)}{u(z)} = k \left( (1 - z^{-1}) z^{-\frac{\theta}{T_m}} \left( \frac{z^{-1} - z^{-1} e^{-\frac{T_m}{\tau}}}{(1 - z^{-1}) \left( 1 - z^{-1} e^{-\frac{T_m}{\tau}} \right)} \right) \right) \quad (\text{A. 11})$$

$$\frac{y(z)}{u(z)} = k \left( z^{-\left( \frac{\theta}{T_m} + 1 \right)} \left( \frac{1 - e^{-\frac{T_m}{\tau}}}{\left( 1 - z^{-1} e^{-\frac{T_m}{\tau}} \right)} \right) \right) \quad (\text{A. 12})$$

### A.1.2 Transformada Z a Modelo Discreto

Multiplicando los denominadores de ambos lados de la ecuación (A. 12) se obtiene:

$$y(z) \left( 1 - z^{-1} e^{-\frac{T_m}{\tau}} \right) = k \left( z^{-\left( \frac{\theta}{T_m} + 1 \right)} \left( 1 - e^{-\frac{T_m}{\tau}} \right) \right) u(z) \quad (\text{A. 13})$$

$$y(z) - e^{-\frac{T_m}{\tau}} z^{-1} y(z) = k \left( 1 - e^{-\frac{T_m}{\tau}} \right) z^{-\left( \frac{\theta}{T_m} + 1 \right)} u(z) \quad (\text{A. 14})$$

$$y(z) = e^{-\frac{T_m}{\tau}} z^{-1} y(z) + k \left( 1 - e^{-\frac{T_m}{\tau}} \right) z^{-\left( \frac{\theta}{T_m} + 1 \right)} u(z) \quad (\text{A. 15})$$

Finalmente, de la expresión (A.15) se obtiene:

$$y_k = e^{-\frac{T_m}{\tau}} y_{k-1} + k \left(1 - e^{-\frac{T_m}{\tau}}\right) u_{k-D-1} \quad (\text{A. 16})$$

Donde  $D = \theta/T_m$ .

## A.2. Controlador PI

### A.2.1 Integral tiempo continuo a discreto

Existen tres métodos numéricos de cálculo de integrales para un instante  $k$ .

#### A.2.1.1. Método Euler hacia Adelante

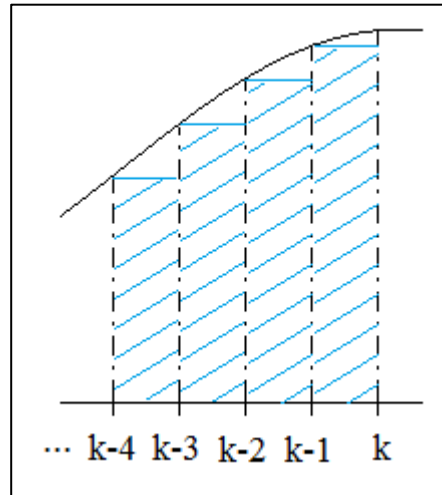


Figura A.1. Método de Euler hacia adelante

Consiste en la sumatoria:

$$\sum_{i=0}^{k-1} T_m e_i \quad (\text{A.17})$$

### A.2.1.2. Método Euler hacia Atrás

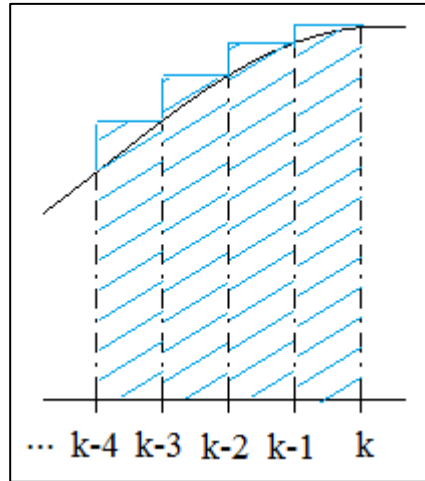


Figura A.2. Método de Euler hacia atrás

Consiste en la sumatoria:

$$\sum_{i=1}^k T_m e_i \quad (\text{A.18})$$

### A.2.1.3. Método del Trapecio

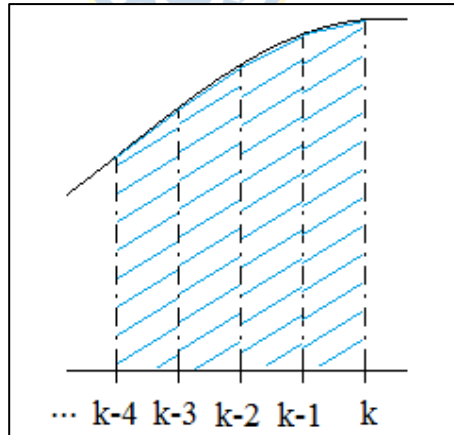


Figura A.3. Método del Trapecio

Consiste en la sumatoria:

$$\sum_{i=1}^k T_m \left( \frac{e_i + e_{i-1}}{2} \right) \quad (\text{A.19})$$

## A.2.2 Discretización del controlador

Sea el controlador PI definido por:

$$u(t) = k_c \left( e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau \right) \quad (\text{A.20})$$

Utilizando el método del trapecio para discretizar se tiene:

$$u_k = k_c \left( e_k + \frac{1}{T_i} \sum_{i=1}^k T_m \left( \frac{e_i + e_{i-1}}{2} \right) \right) \quad (\text{A.21})$$

La ecuación (A.21) se puede escribir como sigue:

$$u_k = k_c \left( e_k + \frac{1}{T_i} \left( \frac{e_k + e_{k-1}}{2} + \sum_{i=1}^{k-1} T_m \left( \frac{e_i + e_{i-1}}{2} \right) \right) \right) \quad (\text{A.21})$$

Si se considera el instante  $k - 1$  en la ecuación (A.20) se tiene:

$$u_{k-1} = k_c \left( e_{k-1} + \frac{1}{T_i} \sum_{i=1}^{k-1} T_m \left( \frac{e_i + e_{i-1}}{2} \right) \right) \quad (\text{A.22})$$

Realizando la diferencia entre las ecuaciones (A.21) y (A.22) se obtiene:

$$u_k - u_{k-1} = \Delta u_k = k_c \left( e_k - e_{k-1} + \frac{1}{T_i} \left( \frac{e_k + e_{k-1}}{2} \right) \right) \quad (\text{A.23})$$

Finalmente, reordenando, se obtiene:

$$u_k = \Delta u_k + u_{k-1} \quad (\text{A.24})$$

$$\Delta u_k = k_c \left( 1 + \frac{1}{2T_i} \right) e_k + k_c \left( -1 + \frac{1}{2T_i} \right) e_{k-1} \quad (\text{A.25})$$

## Anexo B. Instrumentación

---

### B.1. PLC

En la planta microcanal se utiliza un PLC “Allen Bradley ControlLogix 1756-A7/A” con siete slots para módulos I/O, de los cuales solo se utilizan cuatro: Ethernet (1756-EN2T), CPU (1756-L61), Entrada Análoga (1756-IFH8) y Salida Análoga (1756-OFH8). La programación del PLC se realiza utilizando el programa RsLogix 5000. La comunicación entre el computador y el PLC se realiza mediante conexión Ethernet. La Figura B.1 muestra el PLC con los módulos instalados tal como se encuentra en la planta microcanal.



Figura B.1. PLC Allen-Bradley ControlLogix 1756-A7/A

### B.2. Sensores de nivel

Los sensores instalados en la planta microcanal son de la marca Endress&Hauser y corresponden al modelo Prosonic M FMU40. La entrada (alimentación) de este sensor es de 8-30[V] y su salida es de 4-20 [mA] (protocolo Hart). Su rango de medición es de 0-5[m]. La tasa de transmisión es de 100[ms].

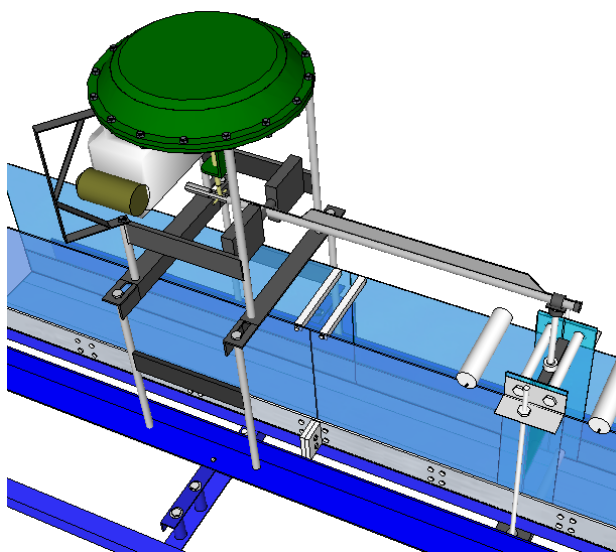




**Figura B.2** Sensor Prosonic M FMU40

### **B.3. Compuertas**

Tanto las compuertas como su actuación fueron diseñadas e implementadas en el trabajo desarrollado por Ignacio Hernández Pino en [15]. Las compuertas son accionadas de forma neumática utilizando posicionadores Fisher y actuadores neumáticos Baumann, los cuales son accionados por una señal de 4-20[mA] (protocolo Hart), y utilizan aire comprimido suministrado por el sistema de la Universidad de Concepción. La Figura B.3 muestra el diseño de una de las compuertas instaladas en el microcanal.



**Figura B.3** Compuerta de microcanal (Actuación neumática)

## Anexo C. Códigos

---

### C.1. Simulación Modelo

```

%% Parametros planta
L=14.5e-2;

C1min=2e-2;
C1max=20e-2;

C2min=2e-2;
C2max=20e-2;

N1max=30e-2;
N2max=30e-2;

ro=1;
g=9.8;

%Piscina 1
Tramola=207e-2;
Ala=L*Tramola;
Tramolb=258e-2;
Alb=L*Tramolb;
A1=Alb;

u10=10/100*(C1max-C1min)+C1min;
N10=57.1/100*N1max;

%Piscina 2
Tramo2=201e-2;
A2=L*Tramo2;

u20=10/100*(C2max-C2min)+C2min;
N20=41.5/100*N2max;

%Flujo y constante
Cd=0.25;
Qemax=0.00426;
Qe0=0.75*Qemax;

%%%% Piscina 3 %%%
N30=N20-(N10-N20)*(u10/u20)^2;
Tramo3=174e-2;
A3=L*Tramo3;

%%%% Piscina 4 %%%
u30=8e-2;
N40=N30-(1/(2*g))*(Qe0/(Cd*L*u30))^2;
A0=Qe0/(Cd*real(sqrt(2*g*N40)));
Tramo4=71e-2;
A4=L*Tramo4;

%% Simulación

```



```

tfinal=20*60;
t0=0.1;%tiempo "muestreo" modelo
t=0:t0:tfinal;
t0c=0.5;
tc=0:t0c:tfinal;

nf=length(t);

Qe=0.75*Qemax*ones(length(t),1);
% Qe((20*60/t0):nf)=1*Qemax;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
nfc=length(tc);

Qec=Qe(1)*ones(1,nfc);

ulc=10*ones(1,nfc);
ulc=5*ones(5*60/t0,nfc);

u2c=11.1160*ones(1,nfc);

theta1=20;
Ndelay1=ceil(theta1/t0);

theta2=20;
Ndelay2=ceil(theta1/t0);

for i=(2+max(t0c/t0,(max(Ndelay1,Ndelay2)))):nf
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
    W1=u1(i-1);
    W2=u2(i-1);
    W3=u30;

    Qc1=Cd*L*W1*real(sqrt(2*g*(y1(i-1)-y2(i-1))));
    Qc2=Cd*L*W2*real(sqrt(2*g*(y2(i-1)-y3(i-1))));
    Qc3=Cd*L*W3*real(sqrt(2*g*(y3(i-1)-y4(i-1))));
    Qs=Cd*A0*real(sqrt(2*g*(y4(i-1))));

    y4(i)=t0*(Qc3-Qs)/A4+y4(i-1);
    y3(i)=t0*(Qc2-Qc3)/A3+y3(i-1);
    y2(i)=t0*(Qc1-Qc2)/A2+y2(i-1);
    y1(i)=t0*(Qe(i-1)-Qc1)/A1+y1(i-1);
end

```



## C.2. Controlador PI

```

if mod((i*t0),t0c)==0

    j=i*t0/t0c;

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Controlador 2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %Analog input
    y2c(j)=y2(i-Ndelay1)*100/N2max;
    %Calculo de error
    e2(j)=sp2(j)-y2c(j);
    %Accion de control
    e2(j)=swa*e2(j);
    % Banda muerta
    if abs(e2(j))<Nb, e2(j)=0; end
    for jj=2:length(sch2_u)
        if (u2c(j-1)>=sch2_u(jj-1)) && (u2c(j-1)<=sch2_u(jj))
            sch2_sw=jj;
        end
    end
    end

    kc2=sch2_kc(sch2_sw-1)+(u2c(j-1)-sch2_u(sch2_sw-1))/(sch2_u(sch2_sw)-
        sch2_u(sch2_sw-1))*(sch2_kc(sch2_sw)-sch2_kc(sch2_sw-1));

    % calculo del control
    q02=kc2*(1+(t0c/(2*ti2)));
    q12=kc2*(-1+(t0c/(2*ti2)));

    duk2=q02*e2(j)+q12*e2(j-1);

    %limitador duk
    if abs(duk2)>dumax, duk2=sign(duk2)*dumax; end

    if sw_AM2==1
        u2c(j)=u2c(j-1)+duk2;
    else
        sp2(j)=y2c(j);
    end;

    if u2c(j)>umax, u2c(j)=umax; end
    if u2c(j)<umin, u2c(j)=umin; end

    %Analog output
    u2(i:(i+t0c/t0-1))=C2min+u2c(j)*(C2max-C2min)/100;

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Controlador 1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Analog input
    y1c(j)=y1d(j-theta1/t0c)+ruido_d1;
    %Calculo de error
    e1(j)=sp1(j)-y1c(j);
    %Accion de control
    e1(j)=swa*e1(j);

```

```

% Banda muerta
if abs(e1(j))<Nb, e1(j)=0; end

for jj=2:length(sch1_u)
    if (ulc(j-1)>=sch1_u(jj-1)) && (ulc(j-1)<=sch1_u(jj))
        sch1_sw=jj;
    end
end

kcl=sch1_kc(sch1_sw-1)+(ulc(j-1)-sch1_u(sch1_sw-1))/(sch1_u(sch1_sw)-
sch1_u(sch1_sw-1))*(sch1_kc(sch1_sw)-sch1_kc(sch1_sw-1));

% calculo del control
q01=kcl*(1+(t0c/(2*ti1)));
q11=kcl*(-1+(t0c/(2*ti1)));

duk1=q01*e1(j)+q11*e1(j-1);

%limitador duk
if abs(duk1)>dumax, duk1=sign(duk1)*dumax; end

if sw_AM1==1
    ulc(j)=ulc(j-1)+duk1-0*duk2;
    %         if u1(i)>1.01*y1(i)
%             u1(i)=1.01*y1(i);
%         end
else
    spl(j)=y1c(j);
end;

if ulc(j)>umax, ulc(j)=umax; end
if ulc(j)<umin, ulc(j)=umin; end

%Analog output
u1(i:(i+t0c/t0-1))=C1min+ulc(j)/100*(C1max-C1min);

Qec(j)=Qe(i);

end

```



## C.3. Controlador Fuzzy

### C.3.1. Controlador

```

if mod((i*t0),t0c)==0

j=i*t0/t0c;
%%%Controlador 2      %%%

%Analog input
y2d(j)=y2(i)*100/N2max;
y2c(j)=y2d(j-theta2/t0c)+ruido_d2;

%Calculo de error
e2(j)=sp2(j)-y2c(j);

% Banda muerta
if abs(e2(j))<Nb, e2(j)=0; end

%Derivada del error
de2(j)=(e2(j)-e2(j-1))/t0c;

Es2(j)=e2(j)*Se2;
DEs2(j)=de2(j)*Sde2;

%INTERFASE FUSIFICADORA
%Error
Me2_NB=fuzzyNB(Es2(j));
Me2_NM=fuzzyNM(Es2(j));
Me2_NS=fuzzyNS(Es2(j));
Me2_ZE=fuzzyZE(Es2(j));
Me2_PS=fuzzyPS(Es2(j));
Me2_PM=fuzzyPM(Es2(j));
Me2_PB=fuzzyPB(Es2(j));
%Delta Error
Mde2_NB=fuzzyNBd(DEs2(j));
Mde2_NM=fuzzyNMd(DEs2(j));
Mde2_NS=fuzzyNSd(DEs2(j));
Mde2_ZE=fuzzyZEd(DEs2(j));
Mde2_PS=fuzzyPSd(DEs2(j));
Mde2_PM=fuzzyPMd(DEs2(j));
Mde2_PB=fuzzyPBd(DEs2(j));

%LOGICA TOMA DE DECISIONES

%Reglas
R11=min([Mde2_NB Me2_NB]);
R12=min([Mde2_NB Me2_NM]);
R13=min([Mde2_NB Me2_NS]);
R14=min([Mde2_NB Me2_ZE]);
R15=min([Mde2_NB Me2_PS]);
R16=min([Mde2_NB Me2_PM]);
R17=min([Mde2_NB Me2_PB]);

R21=min([Mde2_NM Me2_NB]);

```



```

R22=min([Mde2_NM Me2_NM]);
R23=min([Mde2_NM Me2_NS]);
R24=min([Mde2_NM Me2_ZE]);
R25=min([Mde2_NM Me2_PS]);
R26=min([Mde2_NM Me2_PM]);
R27=min([Mde2_NM Me2_PB]);

```

```

R31=min([Mde2_NS Me2_NB]);
R32=min([Mde2_NS Me2_NM]);
R33=min([Mde2_NS Me2_NS]);
R34=min([Mde2_NS Me2_ZE]);
R35=min([Mde2_NS Me2_PS]);
R36=min([Mde2_NS Me2_PM]);
R37=min([Mde2_NS Me2_PB]);

```

```

R41=min([Mde2_ZE Me2_NB]);
R42=min([Mde2_ZE Me2_NM]);
R43=min([Mde2_ZE Me2_NS]);
R44=min([Mde2_ZE Me2_ZE]);
R45=min([Mde2_ZE Me2_PS]);
R46=min([Mde2_ZE Me2_PM]);
R47=min([Mde2_ZE Me2_PB]);

```

```

R51=min([Mde2_PS Me2_NB]);
R52=min([Mde2_PS Me2_NM]);
R53=min([Mde2_PS Me2_NS]);
R54=min([Mde2_PS Me2_ZE]);
R55=min([Mde2_PS Me2_PS]);
R56=min([Mde2_PS Me2_PM]);
R57=min([Mde2_PS Me2_PB]);

```



```

R61=min([Mde2_PM Me2_NB]);
R62=min([Mde2_PM Me2_NM]);
R63=min([Mde2_PM Me2_NS]);
R64=min([Mde2_PM Me2_ZE]);
R65=min([Mde2_PM Me2_PS]);
R66=min([Mde2_PM Me2_PM]);
R67=min([Mde2_PM Me2_PB]);

```

```

R71=min([Mde2_PB Me2_NB]);
R72=min([Mde2_PB Me2_NM]);
R73=min([Mde2_PB Me2_NS]);
R74=min([Mde2_PB Me2_ZE]);
R75=min([Mde2_PB Me2_PS]);
R76=min([Mde2_PB Me2_PM]);
R77=min([Mde2_PB Me2_PB]);

```

%Alpha cuts

%Desde NB a PB (Cerrar mucho a Abrir mucho)

```
ac2_NB=max([R27 R37 R47 R56 R57 R66 R67 R76 R77]);
```

```
ac2_NM=max([R46 R64 R65 R73 R74 R75]);
```

```
ac2_NS=max([R36 R17 R45 R53 R54 R55 R63 R72]);
```

```
ac2_ZE=max([R44]);
```

```
ac2_PS=max([R26 R16 R23 R25 R32 R33 R34 R35 R43 R52 R62 R71]);
```

```
ac2_PM=max([R13 R14 R15 R22 R24 R42]);
```

```
ac2_PB=max([R11 R12 R21 R31 R41 R51 R61]);
```

```

duk2=COA_pmt(ac2_NB,ac2_NM,ac2_NS,ac2_ZE,ac2_PS,ac2_PM,ac2_PM);
duk2=duk2*Sdu2;

%limitador duk
if abs(duk2)>dumax, duk2=sign(duk2)*dumax; end

if sw_AM2==1
    u2c(j)=u2c(j-1)+duk2;
else
    sp2(j)=y2c(j);
end;

if u2c(j)>umax, u2c(j)=umax; end
if u2c(j)<umin, u2c(j)=umin; end

%Analog output
u2(i:(i+t0c/t0-1))=C2min+u2c(j)*(C2max-C2min)/100;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Controlador 1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
y1d(j)=y1(i)*100/Nlmax;
%Analog input
y1c(j)=y1d(j-theta1/t0c)+ruido_d1;

%Calculo de error
e1(j)=sp1(j)-y1c(j);
% Banda muerta
if abs(e1(j))<=Nb
    e1(j)=0;
end

%Derivada del error
del1(j)=(e1(j)-e1(j-1))/t0c;

Es1(j)=e1(j)*Sel;
DEs1(j)=del1(j)*Sdel;

%INTERFASE FUSIFICADORA
%Error
Me1_NB=fuzzyNB(Es1(j));
Me1_NM=fuzzyNM(Es1(j));
Me1_NS=fuzzyNS(Es1(j));
Me1_ZE=fuzzyZE(Es1(j));
Me1_PS=fuzzyPS(Es1(j));
Me1_PM=fuzzyPM(Es1(j));
Me1_PB=fuzzyPB(Es1(j));
%Delta Error
Mdel1_NB=fuzzyNBd(DEs1(j));
Mdel1_NM=fuzzyNMD(DEs1(j));
Mdel1_NS=fuzzyNSd(DEs1(j));
Mdel1_ZE=fuzzyZEd(DEs1(j));
Mdel1_PS=fuzzyPSd(DEs1(j));
Mdel1_PM=fuzzyPMd(DEs1(j));
Mdel1_PB=fuzzyPBd(DEs1(j));

```





```

%LOGICA TOMA DE DECISIONES
%Reglas
R11=min([Mdel_NB Mel_NB]);
R12=min([Mdel_NB Mel_NM]);
R13=min([Mdel_NB Mel_NS]);
R14=min([Mdel_NB Mel_ZE]);
R15=min([Mdel_NB Mel_PS]);
R16=min([Mdel_NB Mel_PM]);
R17=min([Mdel_NB Mel_PB]);

R21=min([Mdel_NM Mel_NB]);
R22=min([Mdel_NM Mel_NM]);
R23=min([Mdel_NM Mel_NS]);
R24=min([Mdel_NM Mel_ZE]);
R25=min([Mdel_NM Mel_PS]);
R26=min([Mdel_NM Mel_PM]);
R27=min([Mdel_NM Mel_PB]);

R31=min([Mdel_NS Mel_NB]);
R32=min([Mdel_NS Mel_NM]);
R33=min([Mdel_NS Mel_NS]);
R34=min([Mdel_NS Mel_ZE]);
R35=min([Mdel_NS Mel_PS]);
R36=min([Mdel_NS Mel_PM]);
R37=min([Mdel_NS Mel_PB]);

R41=min([Mdel_ZE Mel_NB]);
R42=min([Mdel_ZE Mel_NM]);
R43=min([Mdel_ZE Mel_NS]);
R44=min([Mdel_ZE Mel_ZE]);
R45=min([Mdel_ZE Mel_PS]);
R46=min([Mdel_ZE Mel_PM]);
R47=min([Mdel_ZE Mel_PB]);

R51=min([Mdel_PS Mel_NB]);
R52=min([Mdel_PS Mel_NM]);
R53=min([Mdel_PS Mel_NS]);
R54=min([Mdel_PS Mel_ZE]);
R55=min([Mdel_PS Mel_PS]);
R56=min([Mdel_PS Mel_PM]);
R57=min([Mdel_PS Mel_PB]);

R61=min([Mdel_PM Mel_NB]);
R62=min([Mdel_PM Mel_NM]);
R63=min([Mdel_PM Mel_NS]);
R64=min([Mdel_PM Mel_ZE]);
R65=min([Mdel_PM Mel_PS]);
R66=min([Mdel_PM Mel_PM]);
R67=min([Mdel_PM Mel_PB]);

R71=min([Mdel_PB Mel_NB]);
R72=min([Mdel_PB Mel_NM]);
R73=min([Mdel_PB Mel_NS]);
R74=min([Mdel_PB Mel_ZE]);
R75=min([Mdel_PB Mel_PS]);

```



```

R76=min([Mdel_PB Me1_PM]);
R77=min([Mdel_PB Me1_PB]);

%Alpha cuts
%Desde NB a PB (Cerrar mucho a Abrir mucho) ORIGINALES
ac1_NB=max([R27 R37 R47 R56 R57 R66 R67 R76 R77]);
ac1_NM=max([R46 R64 R65 R73 R74 R75]);
ac1_NS=max([R36 R17 R45 R53 R54 R55 R63 R72]);
ac1_ZE=max([R44]);
ac1_PS=max([R26 R16 R23 R25 R32 R33 R34 R35 R43 R52 R62 R71]);
ac1_PM=max([R13 R14 R15 R22 R24 R42]);
ac1_PB=max([R11 R12 R21 R31 R41 R51 R61]);

duk1=COA_pmt(ac1_NB,ac1_NM,ac1_NS,ac1_ZE,ac1_PS,ac1_PM,ac1_PB);
duk1=duk1*Sdul;

%limitador duk
if abs(duk1)>dumax, duk1=sign(duk1)*dumax; end

if sw_AM1==1
    ulc(j)=ulc(j-1)+duk1;
%    ulc(j)=2*ulc(j-1)-ulc(j-2)+duk1;
else
    spl(j)=y1c(j);
end;

if ulc(j)>umax, ulc(j)=umax; end
if ulc(j)<umin, ulc(j)=umin; end

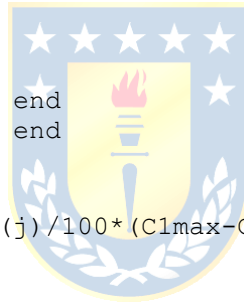
%Analog output
ul(i:(i+t0c/t0-1))=C1min+ulc(j)/100*(C1max-C1min);

u12=u11;
u11=u10;
u10=ulc(j);

Qec(j)=Qe(i);

end

```



### C.3.2. Fusificación

```

%%%%%%%%NB%%%%%%%%
function y=fuzzyNB(x)
if x<(-20)
    y=1;
elseif ((x>=-20)&&(x<(-10)))
    y=(0-1)/(-10-(-20))*(x-(-10))+0;
else
    y=0;
end
%%%%%%%%NM%%%%%%%%
function y=fuzzyNM(x)
if x<(-20)
    y=0;
elseif ((x>=(-20))&&(x<(-10)))
    y=(0-1)/(-20-(-10))*(x-(-20))+0;
elseif ((x>=(-10))&&(x<(-4)))
    y=(1-0)/(-10-(-4))*(x-(-4))+0;
else
    y=0;
end
%%%%%%%%NS%%%%%%%%
function y=fuzzyNS(x)
if x<(-10)
    y=0;
elseif ((x>=(-10))&&(x<(-4)))
    y=(0-1)/(-10-(-4))*(x-(-10))+0;
elseif ((x>=(-4))&&(x<(0)))
    y=(1-0)/(-4-(0))*(x-(0))+0;
else
    y=0;
end
%%%%%%%%ZE%%%%%%%%
function y=fuzzyZE(x)
if x<(-4)
    y=0;
elseif ((x>=(-4))&&(x<(0)))
    y=(0-1)/(-4-(0))*(x-(-4))+0;
elseif ((x>=(0))&&(x<(4)))
    y=(1-0)/(0-(4))*(x-(4))+0;
else
    y=0;
end
%%%%%%%%PS%%%%%%%%
function y=fuzzyPS(x)
if x<(0)
    y=0;
elseif ((x>=(0))&&(x<(4)))
    y=(0-1)/(0-(4))*(x-(0))+0;
elseif ((x>=(4))&&(x<(10)))
    y=(1-0)/(4-(10))*(x-(10))+0;
else
    y=0;
end

```



```

%%%%%%%%PM%%%%%%%%
function y=fuzzyPM(x)
if x<(4)
    y=0;
elseif ((x>=(4))&&(x<(10)))
    y=(0-1)/(4-(10))*(x-(4))+0;
elseif ((x>=(10))&&(x<(20)))
    y=(1-0)/(10-(20))*(x-(20))+0;
else
    y=0;
end
%%%%%%%%PB%%%%%%%%
function y=fuzzyPB(x)
if x<(10)
    y=0;
elseif ((x>=(10))&&(x<(20)))
    y=(0-1)/(10-(20))*(x-(10))+0;
else
    y=1;
end

```

### C.3.3. Defusificación

```

function y=COA_pmt(Mnb,Mnm,Mns,Mze,Mps,Mpm,Mpb)
h=1;
du = -100:h:100;
NB=zeros(1,length(du));
NM=zeros(1,length(du));
NS=zeros(1,length(du));
ZE=zeros(1,length(du));
PS=zeros(1,length(du));
PM=zeros(1,length(du));
PB=zeros(1,length(du));
borde=zeros(1,length(du));
for k = 1:length(du)
    %% NB
    if du(k)<(-70)
        if Mnb < 1
            NB(k)=Mnb;
        else
            NB(k)=1;
        end
    elseif ((du(k)>=-70)&&(du(k)<(-40)))
        if Mnb < ((0-1)/(-40-(-70))*(du(k)-(-40))+0)
            NB(k)=Mnb;
        else
            NB(k)=(0-1)/(-40-(-70))*(du(k)-(-40))+0;
        end
    else
        NB(k)=0;
    end
    %% NM
    if du(k)<(-70)
        NM(k)=0;
    elseif (du(k)>=(-70))&&(du(k)<(-40))
        if Mnm < ((0-1)/(-70-(-40))*(du(k)-(-70))+0)
            NM(k)=Mnm;
        else

```



```

        NM(k)=(0-1)/(-70-(-40))*(du(k)-(-70))+0;
    end
elseif (du(k)>=(-40))&&(du(k)<(-20))
    if Mnm < ((1-0)/(-40-(-20))*(du(k)-(-20))+0)
        NM(k)=Mnm;
    else
        NM(k)=(1-0)/(-40-(-20))*(du(k)-(-20))+0;
    end
else
    NM(k)=0;
end
%% NS
if du(k)<(-40)
    NS(k)=0;
elseif (du(k)>=(-40))&&(du(k)<(-20))
    if Mns < ((0-1)/(-40-(-20))*(du(k)-(-40))+0)
        NS(k)=Mns;
    else
        NS(k)=(0-1)/(-40-(-20))*(du(k)-(-40))+0;
    end
elseif (du(k)>=(-20))&&(du(k)<(0))
    if Mns < ((1-0)/(-20-(0))*(du(k)-(0))+0)
        NS(k)=Mns;
    else
        NS(k)=(1-0)/(-20-(0))*(du(k)-(0))+0;
    end
else
    NS(k)=0;
end
%% ZE
if du(k)<(-20)
    ZE(k)=0;
elseif (du(k)>=(-20))&&(du(k)<(0))
    if Mze < ((0-1)/(-20-(0))*(du(k)-(-20))+0)
        ZE(k)=Mze;
    else
        ZE(k)=(0-1)/(-20-(0))*(du(k)-(-20))+0;
    end
elseif (du(k)>=(0))&&(du(k)<(20))
    if Mze < ((1-0)/(0-(20))*(du(k)-(20))+0)
        ZE(k)=Mze;
    else
        ZE(k)=(1-0)/(0-(20))*(du(k)-(20))+0;
    end
else
    ZE(k)=0;
end
%% PS
if du(k)<(0)
    PS(k)=0;
elseif (du(k)>=(0))&&(du(k)<(20))
    if Mps < ((0-1)/(0-(20))*(du(k)-(0))+0)
        PS(k)=Mps;
    else
        PS(k)=(0-1)/(0-(20))*(du(k)-(0))+0;
    end
elseif (du(k)>=(20))&&(du(k)<(40))
    if Mps < ((1-0)/(20-(40))*(du(k)-(40))+0)

```

```

        PS(k)=Mps;
    else
        PS(k)=(1-0)/(20-(40))*(du(k)-(40))+0;
    end
end
else
    PS(k)=0;
end
%% PM
if du(k)<(20)
    PM(k)=0;
elseif (du(k)>=(20)) && (du(k)<(40))
    if Mpm < ((0-1)/(20-(40))*(du(k)-(20))+0)
        PM(k)=Mpm;
    else
        PM(k)=(0-1)/(20-(40))*(du(k)-(20))+0;
    end
elseif (du(k)>=(40)) && (du(k)<(70))
    if Mpm < ((1-0)/(40-(70))*(du(k)-(70))+0)
        PM(k)=Mpm;
    else
        PM(k)=(1-0)/(40-(70))*(du(k)-(70))+0;
    end
else
    PM(k)=0;
end
end
%% PB
if du(k)<(40)
    PB(k)=0;
elseif (du(k)>=(40)) && (du(k)<(70))
    if Mpb < ((0-1)/(40-(70))*(du(k)-(40))+0)
        PB(k)=Mpb;
    else
        PB(k)=(0-1)/(20-(40))*(du(k)-(20))+0;
    end
else
    if Mpb < 1
        PB(k)=Mpb;
    else
        PB(k)=1;
    end
end
borde(k) = max([NB(k), NM(k), NS(k), ZE(k), PS(k), PM(k), PB(k)]);
end
du=du./100;
centroide = trapz(du,borde.*du)/trapz(du, borde);
%end;

y = centroide;

```

