



**EDUCACIÓN**

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO

# Tecnológico Nacional de México

Centro Nacional de Investigación  
y Desarrollo Tecnológico

## Tesis de Maestría

**Diseño de un controlador neuronal utilizando la  
identificación de sistemas para un reactor anaerobio  
híbrido (RAH)**

presentada por

**Ing. Christian Rios Enriquez**

como requisito para la obtención del grado de

**Maestro en Ciencias en Ingeniería  
Electrónica**

Director de tesis

**Dr. Manuel Adam Medina**

Codirector de tesis

**Dr. Albino Martínez Sibaja**

Cuernavaca, Morelos, México. Noviembre de 2019.



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO

Centro Nacional de Investigación y Desarrollo Tecnológico  
Departamento de Ingeniería Electrónica

"2019. Año del Caudillo del Sur, Emiliano Zapata"

Cuernavaca, Mor.,

No. de Oficio:

Asunto:

1/1/2019

DIE/247/2019

Aceptación de documentos de tesis

**DR. GERARDO VICENTE GUERRERO RAMÍREZ**  
**SUBDIRECTOR ACADÉMICO**  
**PRESENTE**

Por este conducto, los integrantes de Comité Tutorial del **C. Ing. Christian Ríos Enríquez**, con número de control **M17CE066** de la Maestría en Ciencias en Ingeniería Electrónica, le informamos que hemos revisado el trabajo de tesis profesional titulado **"DISEÑO DE UN CONTROLADOR NEURONAL UTILIZANDO LA IDENTIFICACIÓN DE SISTEMAS PARA UN REACTOR ANAEROBIO HÍBRIDO (RAH)"** y hemos encontrado que se han realizado todas las correcciones y observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

DIRECTOR DE TESIS

Dr. Manuel Adam Medina  
Doctor en Ciencias en Ingeniería Electrónica  
Cédula profesional 5992003

CODIRECTOR DE TESIS

Dr. Albino Martínez Sibaja  
Doctor en Ciencias en Ingeniería Electrónica  
Cédula profesional 7221536

REVISOR 1

Dr. Carlos Manuel Astorga Zaragoza  
Doctor en Ingeniería Electrónica  
Cédula profesional 5505019

REVISOR 2

Dr. Ricardo Fabricio Escobar Jiménez  
Doctor en Ciencias en Ingeniería Electrónica  
Cédula profesional 7534115

C.p. M.E. Guadalupe Garrido Rivera- Jefa del Departamento de Servicios Escolares

Estudiante  
Expediente  
MPS/rr.



Interior Internado Palmira S/N, Col. Palmira, C. P. 62490, Cuernavaca, Morelos.  
Tel. (01) 777 3 62 77 70, ext. 222S, e-mail: die@cenidet.edu.mx  
www.tecnm.mx | www.cenidet.edu.mx





**SEP**  
SECRETARÍA DE  
EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MEXICO

Centro Nacional de Investigación y Desarrollo Tecnológico

Cuernavaca, Morelos, **21/noviembre/2019**

OFICIO No. SAC/332/2019

Asunto: Autorización de impresión de tesis

**ING. Christian Ríos Enríquez.**  
**CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS**  
**EN INGENIERÍA ELECTRÓNICA**  
**PRESENTE**

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado "DISEÑO DE UN CONTROLADOR NEURONAL UTILIZANDO LA IDENTIFICACION DE SISTEMAS PARA UN REACTOR ANAEROBIO HÍBRIDO (RAH)", ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo

**ATENTAMENTE**

*Excelencia en Educación Tecnológica®*  
*"Conocimiento y tecnología al servicio de México"*

**DR. GERARDO VICENTE GUERRERO RAMÍREZ**  
**SUBDIRECTOR ACADÉMICO**



SEP TecNM  
CENTRO NACIONAL  
DE INVESTIGACIÓN  
Y DESARROLLO  
TECNOLÓGICO  
SUBDIRECCIÓN  
ACADÉMICA

C.p. Departamento de Servicios Escolares.  
Expediente

GVGR/ego

**cenidet**<sup>®</sup>  
Centro Nacional de Investigación  
y Desarrollo Tecnológico

Interior Internado Palmira S/N, Col. Palmira, C. P. 62490, Cuernavaca, Morelos.  
Tel. (01) 777 3 62 77 70, ext. 4106, e-mail: dir\_cenidet@tecnm.mx

[www.tecnm.mx](http://www.tecnm.mx) | [www.cenidet.edu.mx](http://www.cenidet.edu.mx)





---

# Resumen

---

La digestión anaerobia, también denominada biometanización o producción de biogás, es un proceso biológico, que tiene lugar en ausencia de oxígeno, en el cual parte de la materia orgánica de las deyecciones se transforma, por la acción de microorganismos, en una mezcla de gases (biogás), constituido por metano y dióxido de carbono. Para mantener este tipo de procesos en condiciones óptimas de operación los controles clásicos presentan una gran deficiencia debido a los cambios estocásticos que tiene la digestión anaerobia.

El control no lineal es hoy en día una de las técnicas más utilizadas por la comunidad científica y de control automático y posee cualidades y deficiencias como otras técnicas de control, sobresaliendo las ventajas de su uso en muchas ocasiones. Pese a las buenas prestaciones del control no lineal, muchos autores consideran que el control lineal es una técnica adecuada y suficiente para resolver muchos de los problemas de control existentes.

Sin embargo, la mayoría de las técnicas de control, basan sus algoritmos en el conocimiento del modelo del sistema. Hay casos en los que diseñar un modelo matemático que represente la dinámica del sistema con el que se desea trabajar resulta demasiado complejo, en esos casos, el procedimiento para la obtención de dichos modelos es mediante la identificación de sistemas. La cual utiliza las señales de entrada-salida del sistema en estudio.

En este contexto, en este documento de tesis se presenta el uso de la metodología para la identificación de sistemas dinámicos en la que se utilizan redes neuronales para el cálculo de los modelos y para el diseño del controlador. Particularmente, la identificación se realiza fuera de línea y solo se requiere un conjunto de datos entrada-salida para entrenamiento de la red neuronal, los sistemas que se pueden identificar utilizando esta metodología son sistemas SISO.

El caso de estudio que se plantea es la digestión anaerobia que se lleva a cabo en un reactor anaerobio híbrido diseñado por el instituto tecnológico de Orizaba (ITO), donde el desarrollo del modelo matemático del proceso resulta demasiado complejo en analizar quedando como opción la identificación del sistema usando los datos experimentales del proceso.

El objetivo principal de esta tesis es la de diseñar un controlador que tenga la capacidad de adaptarse a los cambios estocásticos que pueda presentar el RAH del ITO y controlar la concentración del sustrato en forma continua en cualquier día de operación con el modelo identificado.



---

# Abstract

---

Anaerobic digestion, also called biomethanization or biogas production, is a biological process which takes place in the absence of oxygen, where part of the organic matter of dejections is transformed by the action of microorganisms into a mixture of gases (biogas) consisting of methane and carbon dioxide. To maintain this type of process in optimal operating conditions, classic controls have a great deficiency due to the stochastic changes by the anaerobic digestion.

Nonlinear control is today one of the techniques most used by the scientific and automatic control community and has qualities and deficiencies like other control techniques, highlighting the advantages of its use on many occasions. Despite the good performance of non-linear control, many authors consider linear control to be an adequate and sufficient technique to solve many of the existing control problems.

However, most control techniques base their algorithms on the knowledge of the system model. There are cases in which designing a mathematical model that represents the dynamics of the system you want to work with is too complex, in those cases, the procedure for obtaining such models is by system identification. Which uses the input-output signals of the system under study.

In this context, this thesis presents the use of the methodology for the identification of dynamics systems in which neural networks are used for the calculation of models and for the design of a controller. Particularly, the identification is done offline and only a set of input-output data is required for training the neural network. The systems that can be identified using this methodology are SISO systems.

The case study that is proposed is the anaerobic digestion that is carried out in a hybrid anaerobic reactor designed by the Insituto Tecnológico de Orizaba (ITO), where the development of the mathematical model of the process is too complex to analyze leaving the identification as an option of the system using experimental data.

The main objective of this thesis is to design a controller capable of adapting stochastic changes that are present in the RAH of the ITO and to control the concentration of the substrate continuously on any operational day with the identified model.

---

# Dedicatoria

---

*A mi madre, Beatriz Enriquez Guzmán, y a mi padre, Alfredo Rios Carrera, que me dieron la vida, por ser los mejores padres del mundo y un ejemplo a seguir, por haberme apoyado e inculcado los valores que me regirán toda mi existencia.*

*A mi gran hermano Emmanuel Rios Enriquez y a mi querida hermana Nathaly Rios Enriquez, por los grandes momentos que hemos disfrutado juntos y por orientarme cuando he estado indeciso. Ustedes me motivan a ser mejor que ayer.*

---

# Agradecimientos

---

*Quiero expresar mi gratitud a Dios, quien con su bendición llena siempre mi vida y a toda mi familia por estar siempre presentes.*

*A mi asesor el Dr. Manuel Adam Medina le agradezco el haberme guiado en la realización del trabajo de investigación, mediante sus invaluable observaciones y consejos; de igual forma agradezco la confianza y la amistad brindada.*

*De manera especial agradezco al Dr. Juan Reyes Reyes por sus asesorías brindadas sobre el uso de las redes neuronales para el desarrollo de este trabajo de tesis.*

*De manera especial agradezco al Dr. Alejandro Alvarado Lassman, al Dr. José Pastor Rodríguez Jarquín, al Dr. Erik Samuel Rosas Mendoza, al Dr. Albino Martínez Sibaja y al M.C. Oscar Marín Peña por las asesorías y apoyos brindados en mi estancia en el Instituto Tecnológico de Orizaba.*

*De manera especial agradezco a la Lic. Lorena Ruiz Ramírez por todo el apoyo brindado en los trámites de mi documentación en toda la trayectoria de mi maestría.*

*A los integrantes del comité revisor: el Dr. Carlos Manuel Astorga Zaragoza y el Dr. Ricardo Fabricio Escobar Jiménez, cuyos comentarios y observaciones fueron muy valiosos como guía y para el desarrollo del presente trabajo.*

*A todos mis profesores del CENIDET, quienes me compartieron sus conocimientos y experiencias, así como su paciencia y comprensión en toda la trayectoria de mi maestría.*

*A mis amigos y compañeros de la maestría con quienes compartí momentos muy agradables. Pero sobre todo a mis amigos de generación de control y potencia, Erick, Henry, Juan, Zetina, Jonathan, Eusebio, Elizabeth, Diana, Bryan, José, Hasmanhy, Gerardo, Oscar, Sabido, Tabares, Diego y Jasmín, quienes compartieron conmigo muchos momentos tanto divertidos como difíciles y a pesar de eso siempre estuvimos juntos.*

*Al Consejo Nacional de Ciencia y Tecnología (CONACYT), por brindarme el sustento económico necesario para culminar en tiempo y forma mis estudios de posgrado.*

*Agradezco profundamente al Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) por proporcionarme los medios necesarios para mi formación académica y por las facilidades otorgadas durante mi estancia, así como de tener un ambiente muy agradable de trabajo y una atención excelente en todo momento.*



---

# Índice general

---

Índice general	V
Índice de tablas	VIII
Índice de figuras	IX
Nomenclatura	XII
<b>1. Introducción</b>	<b>1</b>
1.1. Problemática de investigación . . . . .	2
1.2. Estado del arte . . . . .	4
1.3. Propuesta solución . . . . .	11
1.4. Objetivos . . . . .	13
1.4.1. Objetivo general . . . . .	13
1.4.2. Objetivos específicos . . . . .	13
1.5. Metas . . . . .	13
1.6. Justificación . . . . .	13
1.7. Organización del documento . . . . .	14
<b>2. Marco teórico</b>	<b>15</b>
2.1. Digestión anaerobia en reactores fluidizantes . . . . .	15
2.1.1. Etapas de la digestión anaerobia . . . . .	16
2.1.2. Factores que afectan la digestión anaerobia . . . . .	17
2.1.3. Reactores anaerobios . . . . .	17
2.1.4. Fluidización . . . . .	19
2.1.5. Reactores fluidizantes . . . . .	20
2.1.5.1. Reactor de lecho fluidizado (LF) . . . . .	20
2.1.5.2. Reactor de lecho fluidizado inverso (LFI) . . . . .	21
2.2. Redes Neuronales . . . . .	22
2.2.1. Neurona artificial . . . . .	22
2.2.2. Modelo Neuronal . . . . .	22
2.2.2.1. Modelo Neuronal (Formal) Básico - Modelo de McCulloch-Pitts . . . . .	22
2.2.2.2. Modelo Neuronal Discreto . . . . .	23
2.2.2.3. Modelo Neuronal con capacidad de aprendizaje: modelo Perceptrón . . . . .	23
2.2.3. Estrategia de búsqueda y convergencia . . . . .	25
2.2.4. Arquitecturas de redes neuronales y su clasificación . . . . .	26
2.3. Identificación de sistemas . . . . .	26
2.3.1. Identificación de sistemas usando redes neuronales . . . . .	27
2.3.1.1. Redes neuronales recurrentes . . . . .	27
2.3.1.2. Arquitecturas de identificación neuronal tipo SISO . . . . .	29

2.3.1.3. Entrenamiento neuronal . . . . .	31
2.3.2. Metodología de identificación de sistemas dinámicos . . . . .	32
2.3.3. Índices de desempeño . . . . .	33
2.4. Sumario . . . . .	34
<b>3. Diseño de un esquema de identificación y de un controlador basado en RNA para un RAH</b>	<b>36</b>
3.1. Planteamiento del problema para la identificación y control del RAH . . . . .	36
3.1.1. Caso de estudio: Reactor anaerobio híbrido (RAH) del Instituto Tecnológico de Orizaba . . . . .	36
3.1.2. Limitaciones del RAH del Instituto Tecnológico de Orizaba . . . . .	38
3.2. Propuesta de solución para la identificación y control del RAH . . . . .	41
3.3. Diseño de RNA para la identificación del RAH . . . . .	41
3.4. Diseño de controlador neuronal para el RAH . . . . .	46
<b>4. Identificación y control del RAH</b>	<b>57</b>
4.1. Identificación del RAH utilizando RNA recurrentes . . . . .	57
4.1.1. Tratamiento de datos . . . . .	58
4.1.2. Identificación neuronal fuera de línea para variable de interés: $DQO_s$ a la salida . . . . .	58
4.1.3. Validación de identificación neuronal con variable principal de interés: $DQO_s$ a la salida . . . . .	61
4.1.3.1. CASO 1. Identificación neuronal sin atraso en la toma de muestras . . . . .	62
4.1.3.2. CASO 2. Identificación neuronal con atraso en la toma de muestras . . . . .	64
4.1.4. Modelo neuronal identificado . . . . .	67
4.2. Controlador neuronal para variable de interés: $DQO_s$ a la salida . . . . .	69
4.2.1. CASO 1. Controlador neuronal sin atraso en la toma de muestras . . . . .	70
4.2.2. CASO 2. Controlador neuronal con atraso en la toma de muestras . . . . .	73
<b>5. Conclusiones y perspectivas</b>	<b>77</b>
5.1. Conclusiones . . . . .	77
5.2. Aportaciones . . . . .	79
5.3. Trabajos futuros . . . . .	79
<b>Bibliografía</b>	<b>81</b>
<b>Anexos</b>	<b>86</b>
<b>A. Propiedades básicas</b>	<b>87</b>
A.1. La derivada parcial . . . . .	87
A.2. Regla de la cadena . . . . .	87
A.3. Método directo de Lyapunov . . . . .	87
A.3.1. Principio del Teorema de Estabilidad de Lyapunov . . . . .	87
A.3.2. Función Candidata de Lyapunov . . . . .	88
A.3.3. Teorema de Estabilidad de Lyapunov . . . . .	88
A.4. El algoritmo de gradiente descendente . . . . .	88
A.5. Interpolación Lineal . . . . .	89

<b>B. Identificación fuera de línea de otras variables</b>	<b>90</b>
B.1. Identificación de Biogás . . . . .	90
B.2. Identificación de porcentaje de remoción de $DQO_s$ . . . . .	93
B.3. Identificación de porcentaje de metano producido . . . . .	96
<b>C. Control neuronal</b>	<b>99</b>
C.1. Objetivo de control: seguimiento de trayectoria . . . . .	99
C.2. Comportamiento del control neuronal ante perturbaciones . . . . .	103



---

# Índice de tablas

---

1.1.	Resumen de estado del arte. . . . .	9
1.2.	Resumen de estado del arte (continuación). . . . .	10
4.1.	Validación de modelos identificados. . . . .	67
4.2.	Índices de desempeño de identificación. . . . .	67
4.3.	Modelos neuronales identificados. . . . .	67

---

# Índice de figuras

---

1.1.	Estructura general de alternativa energética mediante el proceso de digestión anaerobia.	1
1.2.	Diagrama del RAH. . . . .	3
1.3.	Propuesta de solución para el control del RAH. . . . .	11
1.4.	Esquema interno del bloque <i>controlador neuronal</i> . . . . .	12
1.5.	Esquema interno del bloque <i>identificación de modelos neuronales</i> . . . . .	12
2.1.	Etapas de digestión anaerobia. . . . .	16
2.2.	Reactor de contacto anaerobio. . . . .	17
2.3.	Reactor de filtro anaerobio: a) Flujo descendente y b) Flujo ascendente. . . . .	18
2.4.	Reactor anaerobio de flujo ascendente con manto de lodos (UASB). . . . .	18
2.5.	Reactor de lecho fluidizado. . . . .	19
2.6.	Regímenes de fluidización. . . . .	20
2.7.	Reactor de tipo LFI. . . . .	21
2.8.	a) Neurona biológica b) Neurona artificial. . . . .	22
2.9.	Modelo neuronal discreto [Cichocki and Unbehauen, 1993]. . . . .	23
2.10.	Modelo neuronal perceptrón [Cichocki and Unbehauen, 1993]. . . . .	24
2.11.	Funciones de activación. . . . .	25
2.12.	Arquitecturas de redes neuronales. . . . .	26
2.13.	RNA recurrente de dos capas modelo perceptrón. . . . .	28
2.14.	Tipo I ecuación (2.12). . . . .	29
2.15.	Tipo II ecuación (2.13). . . . .	29
2.16.	Tipo III ecuación (2.14). . . . .	30
2.17.	Tipo IV ecuación (2.15). . . . .	30
2.18.	Identificación de modelo en esquema serie-paralelo. . . . .	31
2.19.	Procedimiento de identificación básico usando RNA. . . . .	32
3.1.	Operación del lecho dentro del RAH. . . . .	37
3.2.	Datos de DQO. . . . .	38
3.3.	Datos de ST y STV. . . . .	39
3.4.	Datos de pH. . . . .	39
3.5.	Datos de porcentaje de remoción de $DQO_s$ y $DQO_t$ . . . . .	40
3.6.	Datos de monitoreo de biogás. . . . .	40
3.7.	Datos de metano producido. . . . .	41
3.8.	RNA recurrente de tres capas esquema de identificación serie-paralelo. . . . .	42
3.9.	RNA recurrente de tres capas. . . . .	45
3.10.	Neurocontrol con modelo de referencia. . . . .	47
3.11.	Neurocontrol de modelo interno. . . . .	47
3.12.	Neurocontrol predictivo. . . . .	48
3.13.	Conexión de señales del controlador neuronal. . . . .	48
3.14.	RNA del controlador tipo recurrente de tres capas. . . . .	49

3.15.	Esquema de control directo. . . . .	50
3.16.	Esquema de control indirecto (en base con el esquema de control propuesto Figura 1.3). . . . .	51
3.17.	Esquema de pruebas para control indirecto (en base con el esquema de control propuesto Figura 1.3). . . . .	52
3.18.	Retropropagación para el entramiento del controlador neuronal [Giles et al., 1998]. . . . .	53
4.1.	Datos de DQO interpolados. . . . .	57
4.2.	Región de operación de la tangente hiperbólica. . . . .	58
4.3.	CVA utilizada para los esquemas de identificación (etapa de arranque). . . . .	59
4.4.	Identificación de la $DQO_s$ a la salida. . . . .	59
4.5.	EMC de $DQO_s$ a la salida. . . . .	60
4.6.	Adaptación de pesos sinápticos por épocas. . . . .	60
4.7.	Pesos sinápticos para modelo neuronal. . . . .	61
4.8.	Nuevos datos proporcionados de la $DQO_s$ . . . . .	62
4.9.	CVA para validación. . . . .	62
4.10.	Validación de identificación. . . . .	63
4.11.	Pesos sinápticos de la identificación. . . . .	63
4.12.	Validación de identificación nuevo esquema. . . . .	64
4.13.	Validación de identificación con 1 retardo (escenario ideal). . . . .	65
4.14.	Validación de identificación con 3 retardos (escenario ideal). . . . .	65
4.15.	Validación de identificación con 5 retardos (escenario real). . . . .	66
4.16.	Validación de identificación con 10 retardos (escenario real). . . . .	66
4.17.	Modelo neuronal entrenado. . . . .	68
4.18.	Modelo neuronal entrenado con 3000 épocas de entrenamiento (Modelo 8). . . . .	68
4.19.	Modelo neuronal entrenado con 3000 épocas de entrenamiento (Modelo 7). . . . .	69
4.20.	Modelo neuronal con control (escenario ideal) con referencia $0.45 \text{ gDQO}_s/L$ . . . . .	71
4.21.	Comportamiento del controlador neuronal y error generado con referencia $0.45 \text{ gDQO}_s/L$ (escenario ideal). . . . .	71
4.22.	Comportamiento de pesos sinápticos del controlador con referencia $0.45 \text{ gDQO}_s/L$ (escenario ideal). . . . .	72
4.23.	Comportamiento del flujo y CVA del RAH con control con referencia $0.45 \text{ gDQO}_s/L$ (escenario ideal). . . . .	72
4.24.	Modelo neuronal con control (escenario ideal) con referencia $0.45 \text{ gDQO}_s/L$ con diferente día de activación. . . . .	73
4.25.	Modelo neuronal con control (escenario real) con referencia $0.45 \text{ gDQO}_s/L$ . . . . .	74
4.26.	Comportamiento del controlador neuronal y error generado con referencia $0.45 \text{ gDQO}_s/L$ (escenario real). . . . .	74
4.27.	Comportamiento de pesos sinápticos del controlador con referencia $0.45 \text{ gDQO}_s/L$ (escenario real). . . . .	75
4.28.	Comportamiento del flujo y CVA del RAH con control con referencia $0.45 \text{ gDQO}_s/L$ (escenario real). . . . .	75
4.29.	Modelo neuronal con control (escenario real) con referencia $0.45 \text{ gDQO}_s/L$ con diferente día de activación. . . . .	76
B.1.	Identificación de biogás. . . . .	90
B.2.	Adaptación de pesos sinápticos por épocas de la identificación del biogás. . . . .	91
B.3.	EMC de biogás. . . . .	91
B.4.	Pesos sinápticos para modelo neuronal de biogás. . . . .	92
B.5.	Modelo entrenado de biogás. . . . .	92
B.6.	Identificación de porcentaje de remoción de $DQO_s$ . . . . .	93



B.7.	Adaptación de pesos sinápticos por épocas. . . . .	94
B.8.	EMC del porcentaje de remoción de $DQO_s$ . . . . .	94
B.9.	Pesos sinápticos para modelo neuronal de remoción de $DQO_s$ . . . . .	95
B.10.	Modelo entrenado de remoción. . . . .	95
B.11.	Identificación de porcentaje de metano. . . . .	96
B.12.	Adaptación de pesos sinápticos por épocas de la identificación del porcentaje de metano. . . . .	97
B.13.	EMC del porcentaje de metano. . . . .	97
B.14.	Pesos sinápticos para modelo neuronal del porcentaje de metano. . . . .	98
B.15.	Modelo entrenado de metano. . . . .	98
C.1.	Modelo neuronal con seguimiento de trayectoria (escenario ideal). . . . .	100
C.2.	Pesos sinápticos del controlador neuronal para seguimiento de trayectoria. . . . .	101
C.3.	Comportamiento de la señal del controlador neuronal y el error generado en el seguimiento de trayectoria. . . . .	101
C.4.	Comportamiento del flujo del reactor y CVA con el controlador para seguimiento de trayectoria. . . . .	102
C.5.	Modelo neuronal con perturbación (escenario ideal). . . . .	103
C.6.	Comportamiento de la señal del controlador neuronal y el error generado con referencia $0.45 \text{ gDQO}_s/L$ con perturbación (escenario ideal). . . . .	104
C.7.	Comportamiento del flujo del reactor y CVA con el controlador con perturbación (escenario ideal). . . . .	104
C.8.	Modelo neuronal con perturbación (escenario ideal) caso especial. . . . .	105

---

# Nomenclatura

---

## Símbolos

$E$	Función costo o de desempeño.
$Z_a, Z_b, Z_c$	Pesos sinápticos de la tercera capa.
$Z_o$	Peso sináptico del umbral de la tercera capa.
$V_b, V_a, V_c, V_o$	Pesos sinápticos de la segunda capa.
$V_o$	Peso sináptico del umbral de la segunda capa.
$W_b, W_a, W_c$	Vectores de pesos sinápticos de la primera capa.
$\omega_{a_o}, \omega_{b_o}, \omega_{c_o}$	Pesos sinápticos de los umbrales de la primera capa.
$q$	Número de neuronas deseadas en la capa 1 para la entrada $u(k)$ .
$p$	Número de neuronas deseadas en la capa 1 para la salida $y_n(k)$ .
$n_a$	Número de retrasos para la entrada de la RNA.
$n_b$	Número de retrasos para la salida de la RNA.
$n_c$	Número de retrasos para la referencia en la RNA.
$ret_u$	Límite de retrasos deseados para la entrada de la RNA.
$ret_y$	Límite de retrasos deseados para la salida de la RNA.
$ret_c$	Límite de retrasos deseados para el controlador.
$z$	Función activación lineal tipo identidad.
$\eta$	Valor de aprendizaje o paso de integración.
$\eta_0$	Valor de aprendizaje inicial para la estrategia de búsqueda y convergencia.
$\mu$	Coefficiente de aprendizaje.
$\theta$	Umbral.
$\varphi$	Función de activación.
$u(k)$	Entrada actual.
$u(k-1)$	Entrada con un atraso.
$u_{CVA}(k)$	Entrada <i>CVA</i> actual.
$u_{CVA}(k-1)$	Entrada <i>CVA</i> con un atraso.
$y(k)$	Salida actual.
$y(k-1)$	Salida con un atraso.
$y_n(k)$	Salida neuronal actual.
$u_{n_e}(k)$	Salida de la neurona inversa actual.
$J_u$	Vector de entradas con atrasos.
$J_y$	Vector de salidas con atrasos.
$J_{y_n}$	Vector de salidas neuronal con atrasos.
$e$	Error.
$e_n$	Error neuronal.

## Abreviaturas

RNA	Red neuronal artificial o redes neuronales artificiales.
ITO	Instituto Tecnológico de Orizaba.
SISO	Una entrada una salida (Simple input simple output).
SIMO	Una entrada múltiples salidas (Simple input multiple outputs).
MISO	Múltiples entradas una salida (Multiple inputs simple output).
MIMO	Múltiples entradas múltiples salidas (Multiple inputs Multiple outputs).
CSTR	Reactor de tanque agitado continuo (Continuous Stirred-Tank Reactor).
UASB	Reactor anaerobio de flujo ascendente con manto de lodos (Upflow Anaerobic Sludge Blanket).
ST	Sólidos totales.
STV	Sólidos totales volátiles.
pH	Medida de acidez o alcalinidad de una disolución.
MRE	Error relativo promedio.
MSE	Error medio cuadrático.
RAH	Reactor anaerobio híbrido.
ADM1	Modelo de digestión anaerobia 1 desarrollado por International Water Association (IWA).
AM2	Modelo de digestión anaerobia 2 para sustratos particulados.
BRP	Batch-recirculated photoreactor (Fotorreactor por lotes recirculado).
ANFIS	Adaptive neuro fuzzy inference system (Distema de inferencia neuro difusa adaptativa).
FLC	Fuzzy logic controller (Controlador de lógica difusa).
AnMBR	Anaerobic membrane bioreactor (Biorreactor de membrana anaerobia).
MPC	Model predictive control (Control predictivo por modelo).
FF	Feedforward.
FC	Fuzzy controller (Controlador difuso).
UASB-AF	Upflow Anaerobic Sludge Blanket-Anaerobic Filter.
MEC	Microbial electrolysis cells (Células de electrólisis microbiana).
UFCEB	Upflow fixed-bed continuous bioreactor.
FIM	Fuzzy inference model (Modelo de inferencia difusa).
POME	Palm Oil Mill Effluent.
UASB-HCPB	Upflow anaerobic sludge blanket-hollow centered packed bed.
ESA	Extremum seeking algorithm (Algoritmo de búsqueda extrema).
EKF	Extended Kalman Filter (Filtro de Kalman extendido).
LTF	Linear fractional transformation.
UF-CW	Ultra-filtered cheese-whey.
DQO	Demanda química de oxígeno.
SST	Sólidos suspendidos totales.
EMC	Error medio cuadrático.
IAE	Integral del error absoluto.
ITAE	Integral del tiempo por el error absoluto.
ISE	Integral del error cuadrático.
ITSE	Integral del tiempo por el error cuadrático.
$CH_4$	Metano.
$CO_2$	Dióxido de carbono.
$H_2$	Dihidrógeno.
$H_2S$	Ácido sulfhídrico.
$DQO_s$	Demanda química de oxígeno soluble.
CVA	Carga volumétrica aplicada.

# Capítulo 1

## Introducción

El agua es un recurso natural indispensable para la vida humana y el medio ambiente. Sin embargo, el agua ha sufrido un alarmante deterioro debido al rápido desarrollo económico y a su inadecuado uso. Durante décadas, toneladas de sustancias biológicamente activas han sido sintetizadas para su uso en la agricultura, industria, medicina y en otro tipo de compuestos químicos, han sido vertidas al medio ambiente de forma, muchas veces, inadecuada. [Rodríguez, 2015].

Con el objetivo de prevenir cambios irreversibles y reducir el impacto de la contaminación ambiental del planeta, diferentes países han decidido apostar a estrategias de diversificación de la producción de energía, utilizando fuentes renovables. Desde hace más de una década, se habla de biocarburantes y de las oportunidades y desafíos que estos combustibles alternativos al petróleo pueden ofrecer.

El potencial de los combustibles alternativos no se relaciona solamente con la mejora ambiental, además incluye dimensiones económicas, culturales y sociales. La bioenergía es un término bastante amplio que abarca a todos los productos energéticos cuya obtención se realiza mediante algún proceso de conversión de productos o residuos agrícolas y animales.

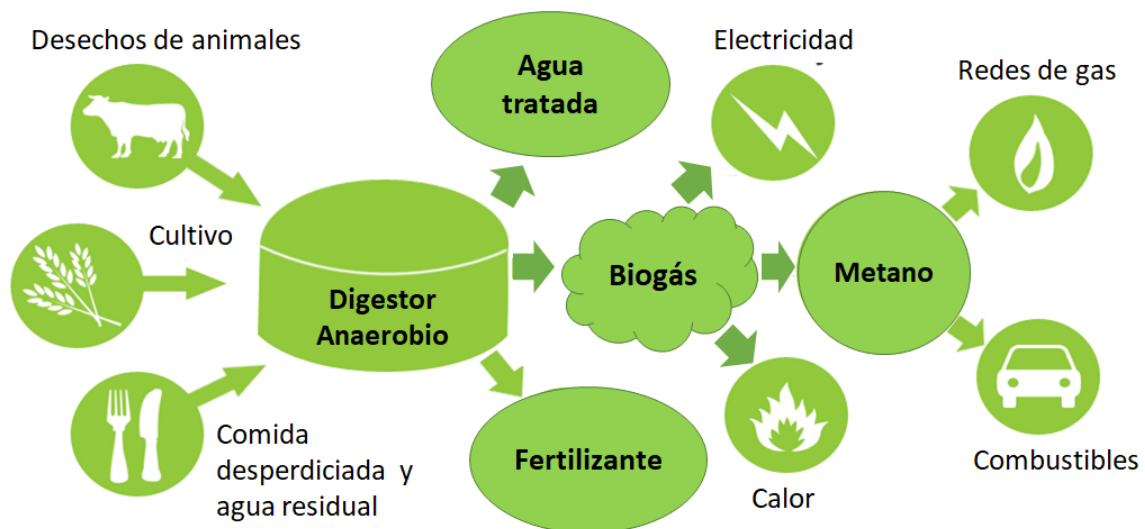


Figura 1.1. Estructura general de alternativa energética mediante el proceso de digestión anaerobia.

Entre los combustibles alternativos tenemos a los biocombustibles o agrocombustibles, que atienden fundamentalmente la demanda del parque automotriz de los países desarrollados. Los conceptos de biocombustible, cultivo energético y biocarburante vienen ganando importancia, cada día con más fuerza, en las políticas agrarias y energéticas tanto de países industrializados como en vías de desarrollo. Una alternativa energética es la obtención de biogás metano a través del proceso de digestión

anaerobia, la cual se lleva a cabo en sistemas conocidos como biorreactores como se muestra en la Figura 1.1.

La actividad agropecuaria y el manejo adecuado de residuos rurales pueden contribuir significativamente a la producción y conversión de residuos animales y vegetales (biomasa) en distintas formas de energía. Durante el proceso de digestión anaerobia se produce *biogás* mediante una serie de reacciones bioquímicas producidas en la biomasa, el cual está constituido principalmente por metano ( $CH_4$ ) y dióxido de carbono ( $CO_2$ ). Este biogás puede ser usado como combustible en muchas aplicaciones.

Siendo más específicos, en el trabajo de [Von Sperling and de Lemos Chernicharo, 2005] mencionan que dentro de un reactor anaerobio conviven diferentes grupos bacterianos en una estrecha simbiosis. En primer lugar, las bacterias acidogénicas degradan la materia orgánica que es introducida al reactor, produciendo un compuesto de ácidos grasos volátiles. Posteriormente, estos ácidos son degradados por bacterias metanogénicas; las cuales producen metano, sin embargo, se deben mantener en ciertas condiciones ambientales para realizar el proceso de digestión de manera óptima.

Por su naturaleza no lineal y dificultad para la medición de las variables que intervienen en el proceso de digestión anaerobia [Jarquin, 2011]; se puede considerar que existe una alta dificultad para controlar dichas variables y asegurar las mejores condiciones para el funcionamiento adecuado de reactores anaerobios.

Uno de los métodos propuestos para mantener condiciones adecuadas en reactores anaerobios ha sido mediante el uso de estrategias de control basadas en modos deslizantes [Jarquin, 2016]. Estas estrategias podrían permitir llevar a cabo la automatización de este tipo de procesos que den confianza a la industria para ser implementados en distintas empresas, con lo cual obtendrían un biogás que puede ser utilizado como gas doméstico y limpiarían al mismo tiempo residuos industriales que, una vez tratados, se pueden utilizar como riego fertilizante.

En la actualidad, la mayoría de las industrias utilizan sistemas de control para aumentar la productividad y seguridad de sus procesos. Entre las diversas técnicas de control existentes se encuentran las técnicas de control basadas en modelo que representan de forma matemática la dinámica del proceso.

Cuando la complejidad de un sistema es muy alta para ser presentado mediante un modelo matemático, muchas veces se recurren a técnicas de identificación de sistemas cuyo objetivo es la obtención de un modelo basado en datos de entrada y salida generados por el sistema.

Las redes neuronales artificiales son una herramienta utilizada para realizar tareas como procesamiento de información, aprendizaje y adaptación. Las redes neuronales están basadas en el método de procesamiento de información del cerebro humano y presentan características que las hacen aplicables a las técnicas de identificación de sistemas.

## 1.1. Problemática de investigación

Los reactores anaerobios que se utilizan para tratamiento de aguas residuales son sistemas complejos que requieren de un sistema de control para su correcto funcionamiento. Los modelados matemáticos para estos tipos de procesos presentan grandes incertidumbres debido a la gran cantidad de factores que se producen en la digestión anaerobia.

---

Para la implementación de una ley de control es indispensable requerir de un modelo matemático que represente de manera precisa al proceso que se busca controlar. Por otra parte, el método de identificación de sistemas solo requiere de datos experimentales donde no es necesario tener un conocimiento tan profundo del proceso a diferencia de un modelo matemático.

Como caso de estudio se seleccionó el *reactor anaerobio híbrido* del ITO que combina dos tipos de lechos, un lecho fluidizante y uno fijo, el cual requiere de una *ley de control para controlar el nivel de concentrado de sustrato* ( $DQO_s$ ). Debido al proceso que se realiza en el reactor para el tratamiento de aguas residuales, realizar un modelo matemático del RAH resulta demasiado complejo. En la Figura 1.2 se presenta como está diseñado e instrumentado el RAH desde la última estancia realizada en el ITO.

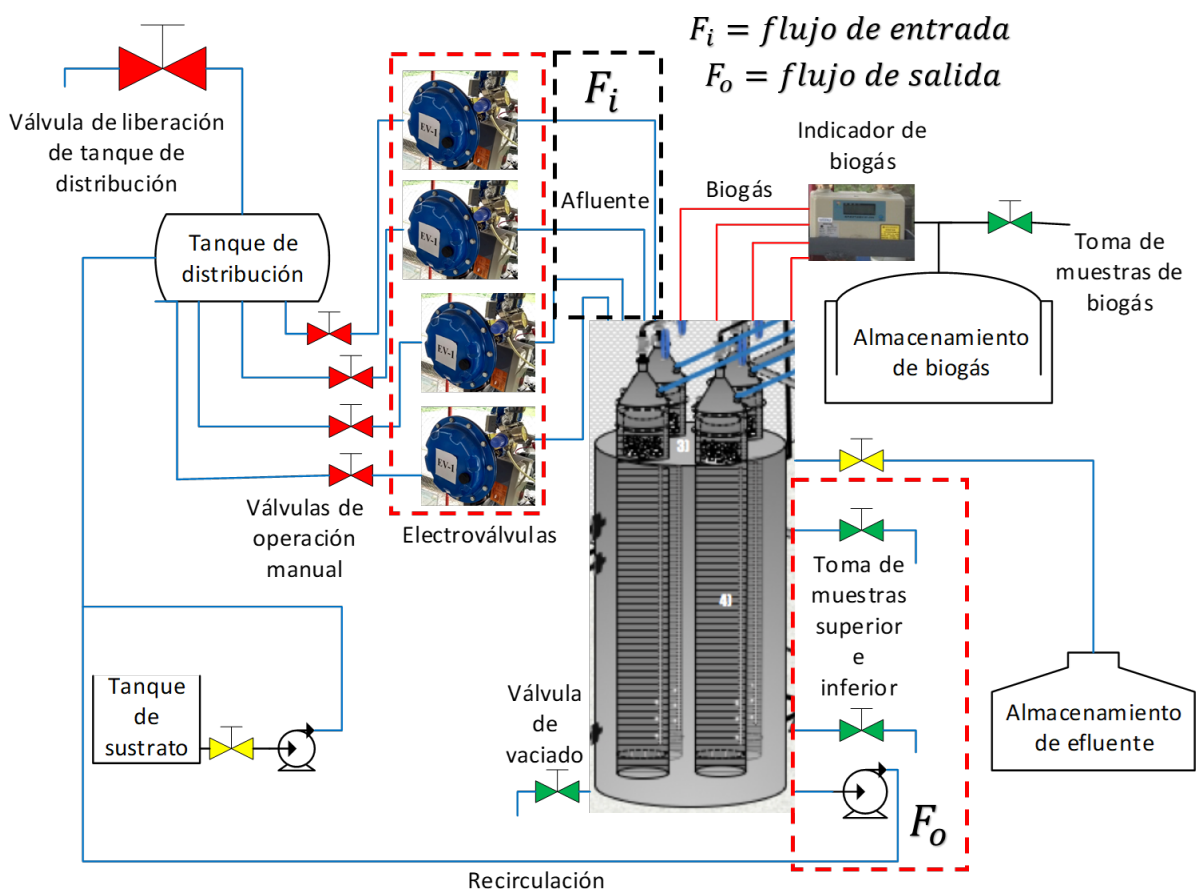


Figura 1.2. Diagrama del RAH.

A continuación se da una descripción de como funciona el RAH de la Figura 1.2:

Cuando el reactor va empezar a trabajar, primero, se activa la motobomba del tanque de sustrato ( $DQO_s$  de entrada) llenado lo que es un tanque de distribución que se encuentra en la parte superior del RAH, una vez lleno el tanque de distribución la motobomba se desactiva y, mediante la fuerza gravitatoria, el contenido del tanque de distribución se vacía pasando por cuatro válvulas manuales y cuatro electroválvulas que alimentan al reactor. Las válvulas manuales funcionan como sistema de emergencia en caso que las electroválvulas fallen.

Con la semi-automatización realizada en la estancia en el ITO el reactor puede operar de manera semi-continua operando de la siguiente manera: cuando el tanque de distribución está por vaciarse, la motobomba, que se encuentra en la parte inferior del reactor, se activa para recircular el flujo del reactor hasta el tanque de distribución, una vez lleno el tanque, la motobomba se desactiva y repite este proceso una y otra vez. Cada hora se activa la motobomba del tanque de sustrato, el cual solo opera dentro de un lapso de 6 a 8 segundos. El reactor no trabaja durante la noche.

## 1.2. Estado del arte

En [Chidrawar and Patre, 2011] se presentó el uso de una RNA inversa como controlador con un compensador de retroalimentación lineal, la RNA propuesta es de tipo MISO con el objetivo de control de seguimiento de trayectoria. Donde este trabajo determina que tiene una mejor eficiencia el uso de pesos sinápticos aleatorios como condición inicial a comparación de valores unitarios para aplicaciones de sistemas no lineales.

En [Ugalde, 2013] la RNA, que se utiliza para identificación, presentó una perspectiva diferente a la RNA de estructura clásica que se usa comúnmente. Se diseña una RNA discreta de una entrada y una salida (SISO) de 3 capas utilizando en la capa 1 una función de activación no lineal tipo sigmoidea y en las capas 2 y 3 se utiliza una función de activación lineal tipo identidad, el diseño de la RNA propuesto le da la libertad de tener enésimas neuronas en la capa 1 siempre y cuando no se cambie la estructura de las capas 2 y 3. En esta RNA se utiliza el algoritmo de aprendizaje basado en el gradiente descendente discreto con una pequeña variante, donde su valor de aprendizaje ( $\eta$ ) es variable teniendo como nombre *algoritmo de búsqueda y convergencia*, dándole mejor capacidad de convergencia a los pesos sinápticos para llegar a sus valores óptimos.

En el trabajo de [Shamiri et al., 2015] se presentó el diseño de un controlador híbrido FLC-ANFIS para el control de temperatura usando un modelo hidrodinámico de dos fases modificado de un RLF para la copolimerización de propileno en fase gaseosa. En los resultados se realizó una comparación de respuesta de control con otros controladores como el FLC y el PID, en donde el controlador híbrido FLC-ANFIS presentaba un desempeño mucho mejor que los otros dos controladores y recomiendan el uso del controlador híbrido FLC-ANFIS para el seguimiento de trayectoria y el rechazo a perturbaciones a comparación de los otros dos controladores.

En el trabajo de [Robles et al., 2015] se diseñó e implementó un sistema ICA para el proceso de tratamiento de aguas residuales urbanas donde utiliza un biorreactor de membrana anaerobia sumergida (AnMBR) en línea. En todo el esquema de la planta se utiliza una gran cantidad de controladores PID y ON/OFF en las capas de menor nivel de la planta y en las capas de mayor nivel se utilizan algunos controladores PID y controladores difusos. En los resultados del proceso se realizó una comparación del uso del PID y el PID con inicio de corrección (FF+PID) donde mostraban que el FF+PID mejoraban el rendimiento de la planta y el FC determinaba la variación de la referencia de acuerdo a la tasa de cambio de flujo.

En [García-Gen et al., 2015] se utilizó una estrategia de control óptimo basado en el método la optimización de programación lineal en un reactor anaerobio híbrido tipo UASB-AF (planta piloto) en línea. Los resultados mostraron que el método de programación lineal con restricciones aparece como un enfoque adecuado para calcular la alimentación óptima de los sistemas de codigestión basados en la caracterización de sustratos convencionales.

---

En [Robles Martinez et al., 2015] se utilizó un controlador de lógica difusa tipo MISO para el control del flujo de metano en un reactor anaerobio tipo UASB. El controlador se utilizó en dos pruebas, una en simulación donde se realizaba un seguimiento de trayectoria del flujo de metano y la otra en aplicación en línea. Los resultados mostraron que el controlador difuso presenta una alta capacidad para compensar posibles perturbaciones en reactor a escala real y presenta respuestas estables para mantener al sistema en estado de equilibrio.

En el trabajo [Azwar et al., 2015] se presentó una comparativa de operación de 3 controladores tipo PID, PID adaptativo y neuronal con objetivo de control el seguimiento de trayectoria y comportamiento ante el rechazo de perturbaciones y mediciones de ruido usando un modelo matemático de células de electrólisis microbiana (MCE) en un reactor anaerobio de lote alimentado (*fed-batch reactor*) para la producción de hidrógeno. Los resultados mostraron que el reactor MEC puede ser controlado para proporcionar una referencia óptima de corriente usando el PID convencional, el PID adaptativo y el controlador neuronal. Sin embargo, el controlador neuronal puede ofrecer un rendimiento más robusto en comparación al PID y PID adaptativo.

En el trabajo [Rincón et al., 2015] se presentó el diseño de un controlador adaptativo basado en el esquema de Lyapunov para reactores biológicos con cinéticas desconocidas. Se utilizó como planta el modelo general de un CSTR en un escenario con efluentes variantes, ruido en las mediciones, cambios abruptos de señal de referencia y saturaciones de la entrada de control, en donde el controlador permitía mantener la referencia deseada a pesar de los cambios desconocidos variantes en el tiempo.

En el trabajo [Yahya et al., 2015] se realizó la comparación de operación de dos controladores tipo PID y PID adaptable usando un modelo identificado basado en el método de superficie de respuesta (RSM) para proceso de células de electrólisis microbiana (MCE) en su punto óptimo de operación. En los resultados se utilizó un objetivo de control de seguimiento de trayectoria en escenarios con ruido y perturbaciones, de modo que mostró que el controlador PID adaptativo presentaba un mejor control del sistema MEC en comparación al PID clásico.

En [Pratiwi et al., 2015] se diseñó un control de temperatura basada en lógica difusa para el control de sustrato usado para la generación de hidrógeno basado en la fermentación oscura dentro de un reactor anaerobio tipo CSTR. Los resultados mostraron que el controlador difuso puede mantener la temperatura del sistema dentro de los  $55^{\circ}\text{C}$  presentando variaciones muy pequeñas, dentro del rango de  $-0.06^{\circ}\text{C}$  a  $0.06^{\circ}\text{C}$ , de error.

En el trabajo [Della Bona et al., 2015] se presentó el diseño de transformación fraccional lineal (LTF) para una versión modificada del modelo AMOCO (versión reducida del modelo ADM1). Se realizaron dos pruebas: en la primera se aplica la identificación paramétrica en el modelo AMOCO modificado con respecto al modelo ADM1 el cual presentó un buen ajuste en las variables deseadas. En la segunda prueba, se realizó otra identificación paramétrica usando el modelo AMOCO modificado con respecto a datos experimentales de digestión anaerobia en UF-CW en un reactor tipo semi-batch, sin embargo, las identificaciones no se mostraban favorables por la limitante de datos por consiguiente se realizó otra prueba utilizando un CSTR donde la identificación paramétrica para el modelo AMOCO presentaba aproximaciones muy cercanas al proceso del CSTR.

En el trabajo de [Venkatesh Prabhu et al., 2016] se utilizó una RNA tipo perceptrón multicapa usando un algoritmo de optimización genético para estimar el porcentaje de decoloración de tinte textil en un RLFI mediante el uso de champiñones ostra (*Pleurotus ostreatus*). En los resultados se comparó la respuesta del porcentaje de decoloración usando un modelo estático basado en la metodo-

---



logía de superficie de respuesta (RSM) y la RNA, donde determinan que el uso del modelo RSM es suficiente para la gestión de aguas residuales textiles en el RLFI. Sin embargo, que el usar la RNA con el algoritmo genético puede ser una mejor alternativa porque presenta una ventaja en la predicción de los datos experimentales y reduce el nivel de error.

En el trabajo de [Poh et al., 2016] se presentó una reseña sobre las limitaciones y oportunidades de investigación que hay en procesos anaerobios, de la cual en aplicaciones industriales los reactores más utilizados son los CSTR y UASB en diferentes ramas y el uso de reactores fluidizantes es mínimo pero es usado en la industria cervecera. También, se presentó los diferentes tipos de modelos matemáticos, donde el más utilizado es el ADM1, además que en el caso de modelos meta-heurísticos se encuentra el uso de redes neuronales, lógica difusa y esquemas híbridos (RNA combinadas con lógica difusa). Además, también se presentó que al algoritmo de PSO es adecuado en el uso de procesos anaerobios. Al final del documento, se presenta los tipos de controladores utilizados, en donde la mayoría se utiliza un controlador tipo PID, de diferentes esquemas como adaptivos, PI, entre otros, para aplicaciones industriales, sin embargo, el uso de controladores neuronales o difusos no son aplicados muy a menudo.

En el trabajo [Shargh and Behnajady, 2016] se presentó el uso de una RNA tipo perceptrón multicapa donde utiliza el algoritmo de Garson para actualizar sus pesos sinápticos, en donde la RNA es utilizada como modelo neuronal para estimar el porcentaje de degradación de la fenazopiridina (producto farmacéutico) en un fotorreactor por lotes recirculado (BRP) donde los resultados muestran que la RNA tiene la capacidad de igualar los datos experimentales.

En el trabajo [Santín et al., 2016] se presentó el uso de diferentes controladores para las decisiones de eliminación de las violaciones de efluentes en plantas de tratamientos de aguas residuales. En este trabajo se utilizó como planta el modelo de simulación Benchmark no. 2 (BSM2), en la cual utilizaban una RNA, tipo perceptron multicapa, para la detección de riesgos de violación de efluentes en concentraciones de nitrógeno, amonio y amoniaco; se realizó una identificación lineal en la parte del reactor de lodos activos para la aplicación del MPC combinado con un controlador anticipado (MPC+FF) con el fin de mantener en la referencia 3 variables generas del proceso ante las fuertes perturbaciones que genera el proceso y se utiliza un controlador difuso (FC) con dos objetivos: la reducción de los costos de producción a través de los cambios generados por los riesgos de violación de efluentes estimados y el cambio de referencias de concentrados de nitrificación en el reactor de lodos activos. Como resultados mostraron que estos esquemas cumplieron el objetivo de reducción de violaciones de efluentes mejorando la calidad del agua y reduciendo costos de operación del proceso.

En [Méndez-Acosta et al., 2016] se presentó el uso de esquema de control híbrido en cascada donde combina un controlador continuo, que regula ácidos volátiles (AGV), y uno discreto, que regula la DQO, basados en una aproximación lineal en un reactor acidogénico y metanogénico tipo UFCB. Los resultados mostraron que el control de las concentraciones de AGV y DQO tiene un buen rendimiento y robustez frente a incertidumbres paramétricas, perturbaciones de carga y referencias variables.

Con [Baruah et al., 2017] se presentó el uso de una RNA multicapa tipo perceptrón para predecir la composición de la producción de gas de la biomasa en términos de concentración de 4 tipo especies de gases que son  $CH_4$  %,  $CO$  %,  $CO_2$  % y  $H_2$  % de un biorreactor de lecho fijo de gasificación de corriente descendente (FBDG). Como resultado, el modelo basado en RNA sirve como una herramienta para optimizar y controlar el proceso de la gasificación.

---

En [Kil et al., 2017] se enfocaron en el control de un modelo no lineal predictivo con identificación en línea utilizando como base el modelo matemático AM2 a diferencia del uso del modelo ADM1 para digestión anaerobia. Estos modelos están basados en tanques CSTR ideales, de la cual realizan una identificación paramétrica en línea utilizando técnicas PCA y filtro de Kalman, además se realiza un NMPC. Donde sus resultados muestran que un NMPC es adecuado para procesos de digestión anaerobia.

En el trabajo [Krzywanski et al., 2017] se utilizó una RNA tipo celular para predecir la *capacidad de enfriamiento* de una bomba de calor de absorción que presenta una configuración en su enfriador de absorción de tres lechos fijos (3BTAC). La RNA usada utiliza dos capas ocultas donde ambas tienen funciones de activación no lineal del tipo tangentes hiperbólicas. El algoritmo genético usado posibilita desarrollar un modelo no iterativo el cual permite realizar un estudio de optimización de la bomba de calor de adsorción con respecto a la capacidad de refrigeración.

En [Reyes-Alvarado et al., 2017] se presentó el uso de un modelo de RNA tipo MIMO (5-11-3) para estimar los valores de eficiencia de la remoción de la DQO (COD RE),  $SO_4^{2-}$  RE y producción de  $S^{2-}$  con el uso de dos reactores de lecho fluidizante inverso (RLFI) anaerobio con el fin de ver los comportamientos que pueden tener cuando hay efecto de las condiciones de alimentación transitoria.

En [Karimi et al., 2017] se presentó el uso se utilizó una RNA tipo perceptrón multicapa utilizando en el algoritmo Levenberg-Marquardt de para estimar los valores de velocidad de fluidización mínima, la mínima velocidad de máxima fluidización y la caída de presión máximo en un reactor de lecho fluidizante cónico (RLFC) con diferentes velocidades de flujo. En sus resultados muestran que la estimación de la RNA presenta igualaciones casi idénticas que los datos experimentales lo cual representa un adecuado desempeño para emplearse eficazmente para una simulación precisa de las características hidrodinámicas de diferentes configuraciones de RLFC.

En el trabajo [Tan et al., 2017] se presentó una identificación del pH, la DQO y los sólidos suspendidos totales (SST) en efluentes basado en el modelo de inferencia difusa (FIM) en un reactor piloto tipo UASB-HCPB para tratar POME bajo condiciones termofílicas. El FIM de las identificaciones dinámicas utilizan las técnicas de los controladores Mamdani para la estimación de comportamientos del pH, DQO y SST donde sus resultados presentaron predicciones favorables con niveles de error promedio dentro de un rango de 2% a 12% y una desviación estándar de 8% a 13%.

En el trabajo [Caraman et al., 2017] se presentó dos esquemas de optimización usados en el modelo ADM1 de procesos anaerobios para la obtención de la máxima cantidad de metano posible. Se realizaron dos pruebas; la primera utiliza el algoritmo de búsqueda extrema (ESA) con señales estocásticas basadas en procesos Wiener y la segunda prueba utiliza el algoritmo basado en modelo para los regímenes óptimos característicos (ORC) del sustrato de entrada en donde se utiliza un filtro de Kalman extendido (EKF) para la estimación de dicho parámetro.

El trabajo de [Li et al., 2018] se basó en predecir el rendimiento de gasificación de  $H_2$ ,  $CO$ ,  $CO_2$ ,  $CH_4$ , producción de gas, valor mínimo de calefacción (LHV) y temperatura utilizando RNA multicapa donde usa como aprendizaje el algoritmo de LMA y es optimizado con el algoritmo de enjambre de partículas en un reactor de lecho fluidizante de ceniza aglomerada (AFB).

En [Khanongnuch et al., 2018] se presentó el uso de una RNA tipo MIMO para la identificación del porcentaje de eficiencia de remoción de tiosulfato de sodio, nitrato y la concentración de sulfato en el proceso de un biorreactor de lecho fluidizante (RLF). Para la identificación se utilizó el *toolbox*

---

de redes neuronales 11.0 de *Matlab R2017b*. La RNA presenta un esquema de dos capas utilizando como aprendizaje el algoritmo de Levenberg-Marquardt (LMA), el cual la RNA predecía con éxito los parámetros de rendimiento del RLF.

Con [Antwi et al., 2018] se presentó el uso de una RNA de estructura MISO tipo discreta para la identificación del porcentaje de remoción de la demanda química de oxígeno en un UASB. La RNA tiene un esquema de dos capas utilizando en la capa 1 una función de activación no lineal como la función sigmoidea y en la capa 2 una función de activación lineal como la función identidad. La estructura RNA utilizada es la red recurrente con el algoritmo de aprendizaje LMA, el uso del error cuadrático medio (ECM) permite ver la eficiencia de la RNA. Además, realiza un esquema de optimización con múltiples modelos de regresión no lineal para mejorar el rendimiento de la RNA recurrente. La RNA recurrente mostró una alta precisión en el ajuste experimental del UASB.

En el trabajo [Li et al., 2019] se presentó el diseño de una RNA adaptativa como controlador combinando técnicas de RNA y funcionales de Lyapunov–Krasovskii (LKFs) para sistemas con retardos no lineales y estados variantes con restricciones. El controlador tiene como objetivo estabilizar el sistema no lineal y mantener la referencia deseada.

En la Tablas 1.1 y 1.2 se presenta un resumen sobre los trabajos que tienen relación con el RAH. La tabla se encuentra organizada de dos formas: una de acuerdo a la estrategia de identificación utilizada y la otra con respecto al controlador utilizado. Ambas formas se encuentran ordenadas por año de publicación y presentan que tipo de procesos se utilizó la identificación y el control.

Los trabajos expuestos muestran que usar la metodología de identificación de sistemas empleando RNA es una opción viable para identificar procesos que presentan dinámicas complejas, ya que tienen la particularidad de aprender y adaptarse a condiciones irregulares que puedan suceder durante el proceso en línea.

En el caso del diseño de controladores, los trabajos expuestos muestran que actualmente el uso de controladores tipo PID siguen siendo una solución robusta, practica y confiable para el control de procesos, sin embargo, cuando los procesos muestran un comportamiento no lineal el controlador PID se vuelve ineficaz. En nuestro caso se presenta un sistema no lineal sin modelo con dinámicas complejas, permite abordar el uso de técnicas más avanzadas para el diseño de controladores como el MPC, RNA, FC, entre otros.

Cuando el sistema o proceso presenta una naturaleza no lineal y no se cuenta con el modelo descriptivo, el uso de controladores basado en RNA presenta una gran ventaja por su capacidad de aprendizaje, adaptación y su posibilidad de combinación con otros tipos de controladores volviéndolos más robustos y complejos.

---

Tabla 1.1. Resumen de estado del arte.

Trabajo	Tipo de proceso	Estrategía de identificación	Controlador
Effect of n/s ratio on anoxic thiosulfate oxidation in a fluidized bed reactor: Experimental and artificial neural network model analysis [Khanongnuch et al., 2018].	RLF	RNA	Ninguno
Modeling of ash agglomerating fluidized bed gasifier using back propagation neural network based on particle swarm optimization [Li et al., 2018].	AFB	RNA	Ninguno
Feedforward neural network model estimating pollutant removal process within mesophilic upflow anaerobic sludge blanket bioreactor treating industrial starch processing wastewater [Antwi et al., 2018].	UASB	RNA	Ninguno
Fuzzy logic modelling for thermophilic anaerobic digestion of palm oil mill effluent (POME) treatment [Tan et al., 2017].	UASB-HCPB	FIM	Ninguno
Model predictive control with on-line model identification for anaerobic digestion processes [Kil et al., 2017].	CSTR	Identificación paramétrica	Ninguno
Optimization of a three-bed adsorption chiller by genetic algorithms and neural networks [Krzywanski et al., 2017].	3BTAC	RNA	Ninguno
Artificial neural network based modeling of biomass gasification in fixed bed downdraft gasifiers [Baruah et al., 2017].	FBDG	RNA	Ninguno
Forecasting the effect of feast and famine conditions on biological sulphate reduction in an anaerobic inverse fluidized bed reactor using artificial neural networks [Reyes-Alvarado et al., 2017].	RLFI	RNA	Ninguno
Designing an Efficient Artificial Intelligent Approach for Estimation of Hydrodynamic Characteristics of Tapered Fluidized Bed from Its Design and Operating Parameters [Karimi et al., 2017].	RLFC	RNA	Ninguno
Modeling and optimization by response surface methodology and neural network–genetic algorithm for decolorization of real textile dye effluent using <i>Pleurotus ostreatus</i> : a comparison study [Venkatesh Prabhu et al., 2016].	RLFI	RNA	Ninguno
A high-efficient batch-recirculated photoreactor packed with immobilized TiO <sub>2</sub> -P25 nanoparticles onto glass beads for photocatalytic degradation of phenazopyridine as a pharmaceutical contaminant: artificial neural network modeling [Shargh and Behnajady, 2016].	BRP	RNA	Ninguno
LFT modelling and identification of anaerobic digestion [Della Bona et al., 2015].	Modelo AMOCO y CSTR	Identificación paramétrica	Ninguno
Identificación de Sistemas Utilizando Redes Neuronales: Un Enfoque Basado en el Balance Entre Sencillez, Precisión y Costo Computacional [Ugalde, 2013].	Múltiples ejemplos	RNA	Ninguno

Tabla 1.2. Resumen de estado del arte (continuación).

Trabajo	Tipo de proceso	Estrategía de identificación	Controlador
Neural Network Controller Design for a Class of Non-linear Delayed Systems With Time-Varying Full-State Constraints [Li et al., 2019].	Múltiples ejemplos	Ninguno	RNA inversa adaptativa y LKFs
Anaerobic Digester Optimization Using Extremum Seeking and Model-Based Algorithms. A Comparative Study [Caraman et al., 2017].	Modelo ADM1	EKF	Optimización: ESA
Optimization of wastewater anaerobic digestion using mechanistic and meta-heuristic methods: current limitations and future opportunities [Poh et al., 2016].	CSTR, UASB y RLF	RNA y Lógica Difusa	PID, PID adaptativo, RNA y difusos
Advanced decision control system for effluent violations removal in wastewater treatment plants [Santín et al., 2016].	Modelo de proceso BSM2	RNA	MPC+FF y FC
A hybrid cascade control scheme for the VFA and COD regulation in two-stage anaerobic digestion processes [Méndez-Acosta et al., 2016].	UFCB	Ninguno	Control continuo y discreto
Modified two-phase model with hybrid control for gas phase propylene copolymerization in fluidized bed reactors [Shamiri et al., 2015].	RLF	Ninguno	FLC-ANFIS, FLC y PID
Control strategy for maximum anaerobic co-digestion performance [García-Gen et al., 2015].	UASB-AF	Ninguno	Optimización: método de programación lineal
Instrumentation, control, and automation for submerged anaerobic membrane bioreactors [Robles et al., 2015].	AnMBR	Ninguno	PID, FC y FF+PID
A robust fuzzy-logic-based controller for bio-methane production in anaerobic fixed-film reactors [Robles Martinez et al., 2015].	UASB	Ninguno	Lógica difusa
A comparative study between Neural Networks (NN)-based and adaptive-PID controllers for the optimal bio-hydrogen gas production in microbial electrolysis cell reactor [Azwar et al., 2015].	Fed-bacth	Ninguno	PID, RNA y PID adaptable
A new adaptive controller for bio-reactors with unknown kinetics and biomass concentration: Guarantees for the boundedness and convergence properties [Rincón et al., 2015].	CSTR	Ninguno	Controlador Adaptable
Modeling, optimization, and control of microbial electrolysis cells in a fed-batch reactor for production of renewable biohydrogen gas [Yahya et al., 2015].	Fed-batch	RSM	PID adaptable y PID
Design of temperature control based fuzzy logic for substrate in thermophilic hydrogen reactor [Pratiwi et al., 2015].	CSTR	Ninguno	Lógica difusa
Hybrid direct neural network controller with linear feed-back compensator [Chidrawar and Patre, 2011].	CSTR y motor de CD	Ninguno	RNA inversa

Derivado del análisis sobre el estado del arte, la sección siguiente, nos permite presentar una alternativa de solución a nuestro problema particular.



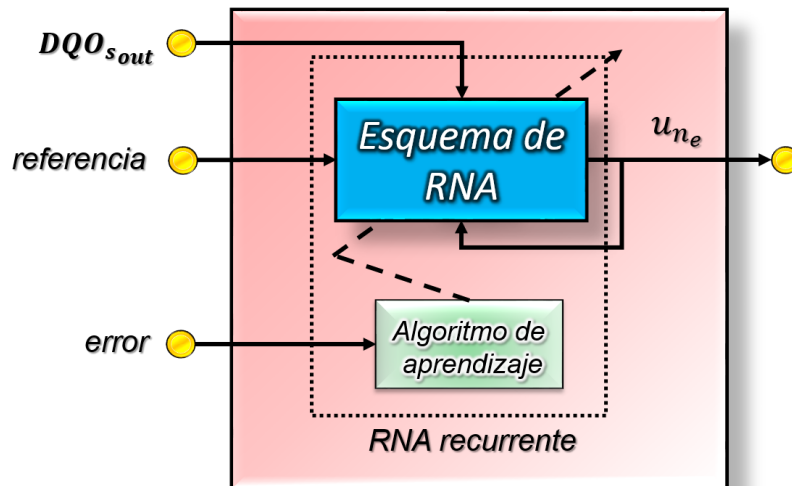


Figura 1.4. Esquema interno del bloque *controlador neuronal*.

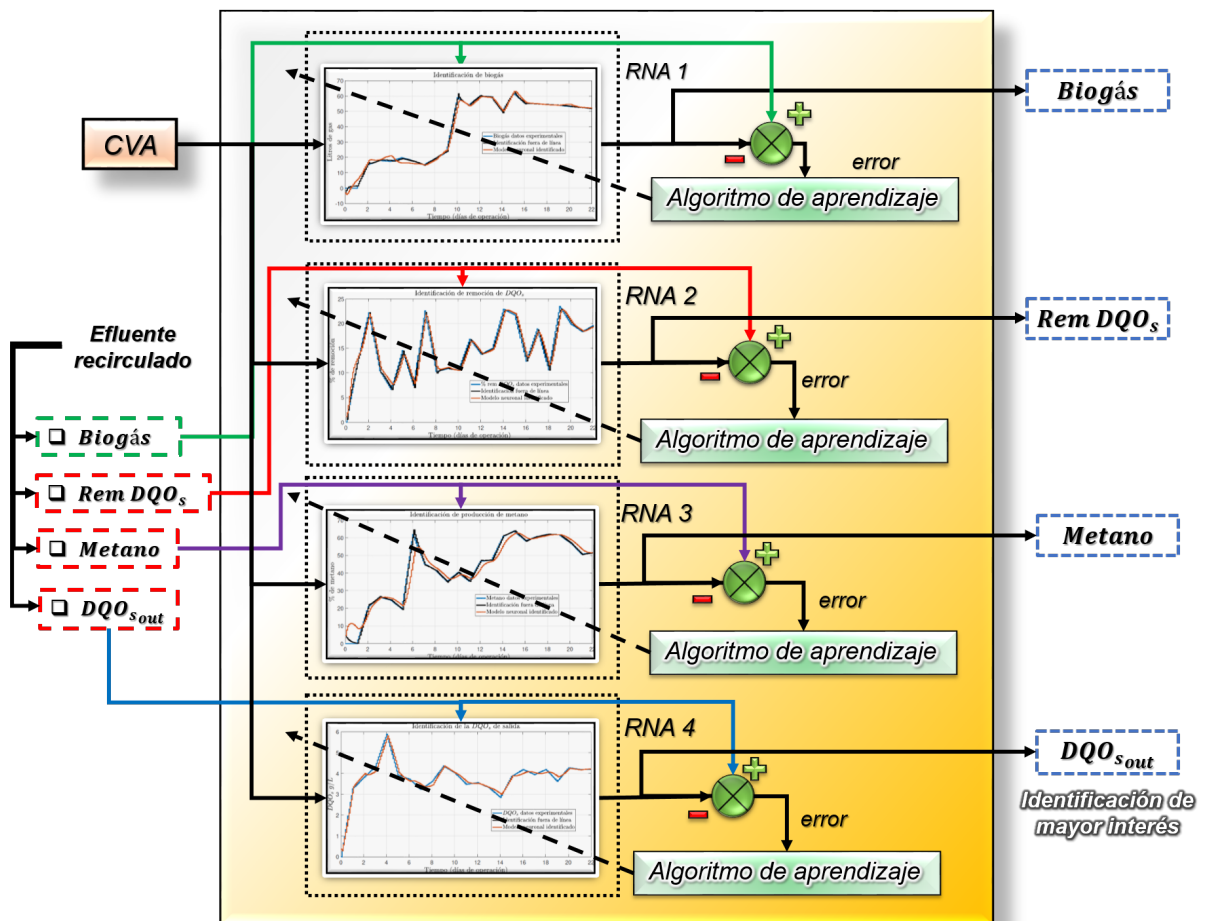


Figura 1.5. Esquema interno del bloque *identificación de modelos neuronales*.

## 1.4. Objetivos

### 1.4.1. Objetivo general

Diseñar un esquema de control neuronal de la concentración del sustrato ( $DQO_s$ ) para el proceso biológico de un RAH identificado con un modelo de RNA.

### 1.4.2. Objetivos específicos

- Realizar la identificación basado en un modelo neuronal que permita determinar los valores estimados del metano, remoción de la  $DQO_s$ , biogás, y  $DQO_s$  de salida que son corroborados por datos de mediciones en laboratorio.
- Determinar a través de la identificación un modelo neuronal entrenado que represente la dinámica de la  $DQO_s$  de salida para aplicar una ley de control.
- Diseñar una estrategia de control utilizando RNA que permita controlar la concentración del sustrato ( $DQO_s$ ) en forma continua en cualquier día de operación del RAH.
- Validación del modelo o modelos identificados en simulación.

## 1.5. Metas

- Obtener la identificación de la  $DQO_s$  de salida, el biogás, el porcentaje de remoción de  $DQO_s$  y el porcentaje de metano producido, generados por el proceso biológico del RAH.
- Mostrar en simulación un controlador neuronal capaz de regular la  $DQO_s$  del sistema en un escenario sin retardos y con retardos en la toma de muestras del proceso.

## 1.6. Justificación

Los tratamientos aerobios y anaerobios constituyen las dos grandes alternativas de depuración biológica de aguas residuales, sin embargo, los procesos anaerobios para el tratamiento de aguas residuales urbanas presentan mayor sostenibilidad que los procesos aerobios debido a la menor producción de lodos, lo que reduce los costos de su tratamiento y la posibilidad de recuperación energética al aprovechar el metano producido en el biogás generado.

La implementación de controladores es necesario si se requieren obtener mejora de resultados de operación en reactores anaerobios; es importante conocer las ventajas y desventajas que presentan los reactores tipo fluidizantes [Sandoval Rojas, 2016]. Para poder implementar un sistema de control se requiere de un modelo matemático preciso de la planta, sin embargo, cuando resulta demasiado complejo desarrollar el modelo matemático se debe aplicar una estrategia para la identificación del sistema no lineal [Vázquez Trejo, 2017].

La configuración híbrida que tiene el reactor anaerobio de ITO es una innovación tecnológica por lo cual información de modelos generalizados y esquemas reducidos todavía no se encuentra en la literatura. Existen aproximaciones de modelos fluidizantes, sin embargo, estos modelos son diseñados bajo esquemas bioquímicos muy particulares y no son aplicables para el RAH, además están diseñados bajo esquemas heurísticos.

Otro problema se debe a la expansión del lecho, la cual es difícil interpretarla cuando se realiza el modelo matemático debido al cambio físico-químico de la materia y las relaciones que puede tener

---



con otras secciones de la digestión anaerobia, además, se debe de considerar la configuración híbrida de los lechos que presenta el reactor del ITO.

## 1.7. Organización del documento

Los capítulos están organizados para familiarizar al lector con temas como teoría y diseño de redes neuronales, aplicaciones de metodologías de identificación de sistemas, procesos biológicos y teoría de control.

En este capítulo se presentó la motivación general por la cual se realiza el proyecto de investigación seguido por los objetivos planteados y las metas a cumplir. También, se da un análisis de los desarrollos teóricos y prácticos más recientes, tanto en identificación de sistemas como de controladores. En los capítulos siguientes se profundiza en la temática involucrada en el desarrollo de la tesis.

En el *Capítulo 2* se presenta la teoría utilizada para el desarrollo de este trabajo de tesis la cual se encuentra estructurada en 3 partes:

- Primero se definen las generalidades de los procesos anaerobios como son conceptos, etapas de la digestión anaerobia, factores que afectan al proceso anaerobio y los tipos de reactores en los que se aplica el proceso anaerobio.
- Después se aborda sobre las generalidades de las redes neuronales como son conceptos, tipos de neuronas artificiales, diseños neuronales y estructuras que se pueden diseñar al combinar las neuronas artificiales.
- Finalmente se aborda la identificación de sistemas presentando como se relacionan las redes neuronales con el esquema de identificación, tipos de entrenamiento al usar las redes neuronales, metodología para aceptar el modelo identificado.

En el *Capítulo 3* se describe la metodología que se usó para abordar la solución del problema de investigación. Se plantea la problemática a resolver junto a las limitaciones de manera más específica que tiene el tema de tesis y se presenta de manera particular la estructura de identificación a utilizar y la estructura del controlador a utilizar de la propuesta solución.

En el *Capítulo 4* contiene las simulaciones realizadas para la identificación y control. Al principio se muestra los resultados que se obtienen al utilizar los esquemas de identificación planteando diferentes escenarios de simulación; posterior a esto, se presentan las simulaciones del controlador planteando los escenarios seleccionados en los resultados de la identificación obtenida.

Finalmente, en el *Capítulo 5* se muestran las conclusiones tras el desarrollo del tema de investigación, las aportaciones y también se muestran algunos trabajos que pueden desarrollarse en el futuro con fundamento en este trabajo de tesis.

---

# Capítulo 2

---

## Marco teórico

---

El objetivo de este capítulo es presentar conceptos requeridos para el entendimiento de este trabajo de tesis. Se presenta información sobre la digestión anaerobia y como se encuentra estructurada, el uso de las redes neurales y como las RNA se relacionan con la identificación de sistemas.

### 2.1. Digestión anaerobia en reactores fluidizantes

Un proceso biológico se basa en el crecimiento de microorganismos (crecimiento de biomasa), por el consumo de nutrientes (sustrato) y bajo condiciones favorables de operación (temperatura, pH, agitación y aeración). Los objetivos de estos procesos biológicos son para la producción de biomasa (levadura para panificación), síntesis de producto (etanol, polímeros biodegradables, metano, entre otros productos) y para el tratamiento de aguas residuales [Dochain, 2013].

La digestión anaerobia es un proceso microbiológico que en condiciones con ausencia de oxígeno permite transformar la materia orgánica en una combinación de gases llamados *biogás* ( $CH_4$ ,  $CO_2$ ,  $H_2$ ,  $H_2S$ , entre otros gases). Para la producción del biogás se utilizan microorganismos anaerobios que crecen utilizando ciertos nutrientes en condiciones favorables.

La eficiencia del proceso de digestión anaerobia se determina en cuanto a la composición del biogás, de la cual el concentrado de gas metano debe de ser alto (80 % en la composición del biogás), para que pueda ser aprovechada como fuente de energía. El biogás puede ser usado en sustitución del gas LP, donde su combustión utiliza energía calorífica, y también puede ser empleado para la generación de energía eléctrica y energía mecánica.

Existen rangos de operación de temperatura para el sistema anaerobio que determina el tipo de digestión anaerobia. Si el proceso se lleva a cabo dentro del rango de temperatura de  $20^{\circ}C$  a  $40^{\circ}C$  se dice que es un proceso de digestión anaerobia mesofílica, mientras que la digestión anaerobia termofílica se lleva a cabo en un rango de temperatura de  $45^{\circ}C$  a  $60^{\circ}C$  [Alvarado-Lassman, 2004].

En este proceso se transforman problemas ecológicos como: la obtención de agua tratada y la obtención de biogás que permita reducir la emisión de gases invernaderos hacia la atmósfera terrestre. Por lo tanto, el proceso anaerobio se considera como una solución energética sustentable.

2.1.1. Etapas de la digestión anaerobia

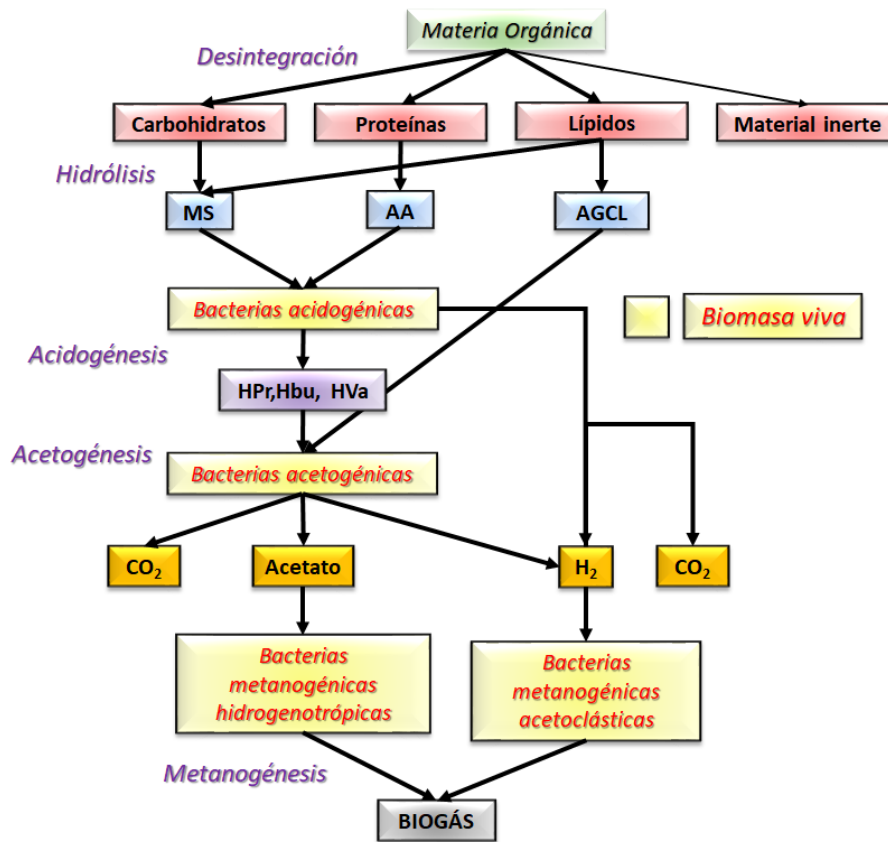


Figura 2.1. Etapas de digestión anaerobia.

En la Figura 2.1 se muestran las diferentes vías de degradación de la materia orgánica y las cinco etapas fundamentales del proceso de digestión anaerobia [Batstone et al., 2002]. Las etapas son las siguientes:

- **Desintegración:** Consiste en un paso no biológico, donde la materia orgánica es convertida en material particular inerte, carbohidratos, proteínas y lípidos.
- **Hidrólisis:** Ocurren tres procesos en paralelo, convirtiendo carbohidratos, proteínas y lípidos en monosacáridos (MS), aminoácidos (AA) y ácidos grasos de cadena larga (AGCL) respectivamente.
- **Acidogénesis:** Dos grupos de bacterias acidogénicas degradan los monosacáridos y aminoácidos en ácidos grasos volátiles, hidrógeno y dióxido de carbono.
- **Acetogénesis:** Las bacterias acetogénicas utilizan los ácidos de cadena larga, los ácidos de cadena corta como propionato (HPr), butirato (Hbu) y valerato (HVa), para convertirlos a acetato, hidrógeno y dióxido de carbono.
- **Metanogénesis:** El hidrógeno y el acetato son convertidos en una mezcla de metano y dióxido de carbono, el hidrógeno producido es convertido por bacterias acetogénicas e hidrogenotrópicas y el acetato por bacterias metanogénicas acetoclásticas.

### 2.1.2. Factores que afectan la digestión anaerobia

Como en cualquier proceso bioquímico, la digestión anaerobia debe de ser monitoreado para asegurar una operación exitosa. Fallas y perturbaciones en reactores anaerobios puede ser causados por sobrecargas hidráulicas y orgánicas, presencia de materiales tóxicos orgánicos o inorgánicos, y variaciones drásticas de operación de temperatura. La digestión anaerobia es un proceso de tres fases (sólida, líquida y gaseosa) en la cual cada fase está relacionada una con otra y algunas veces información adquirida de una fase puede estar directamente relacionada con el estado de las otras dos fases [Borja and Rincón, 2017].

### 2.1.3. Reactores anaerobios

Un reactor anaerobio es un reactor biológico donde se degrada la materia orgánica en ausencia de oxígeno, hasta la obtención de dos componentes principales que son el metano y dióxido de carbono, por medio de múltiples procesos fisicoquímicos y bioquímicos ocasionados por microorganismos o sustancias derivadas de dichos microorganismos. Cuando un reactor se utiliza para tratamientos de aguas residuales, también se denomina *digestor*.

La bacteria anaerobia crece de manera lenta. Si no hay sistema especial que mantenga la recirculación de la bacteria, entonces el tiempo de retención en el reactor es limitado por el crecimiento microbiano. De acuerdo a [Borja and Rincón, 2017] las bacterias son retenidas en los reactores por cuatro métodos básicos:

1. La separación de la biomasa, desde el efluente, se realiza mediante la *sedimentación o filtración*, seguido por la recirculación hacia al reactor. Es tipo de reactor es conocido como reactor de contacto (CR) y su estructura se visualiza en la Figura 2.2.

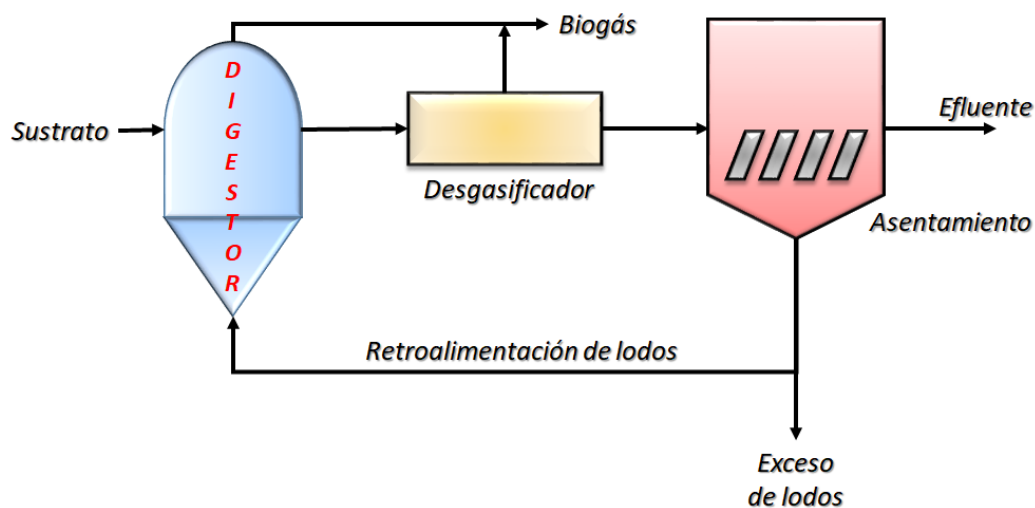


Figura 2.2. Reactor de contacto anaerobio.

2. La retención y fijación de la bacteria se realiza mediante un empaque interno con el fin de reducir la velocidad del flujo ascendente. Este método se realiza en un reactor de *filtro anaerobio* y la Figura 2.3 presenta dos tipos de configuraciones que puede tener el reactor.

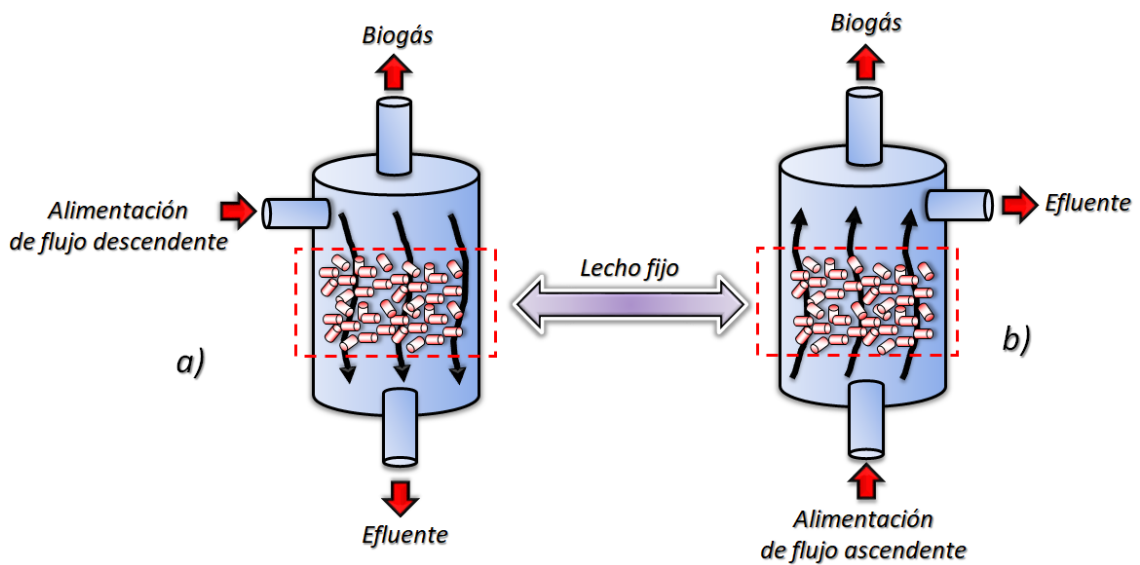


Figura 2.3. Reactor de filtro anaerobio: a) Flujo descendente y b) Flujo ascendente.

3. Granulación y floculación de bacterias naturales asistida por bajas velocidades de flujo ascendente, conocida como reactor anaerobio de flujo ascendente con manto de lodos (UASB) y su estructura se visualiza en la Figura 2.4.

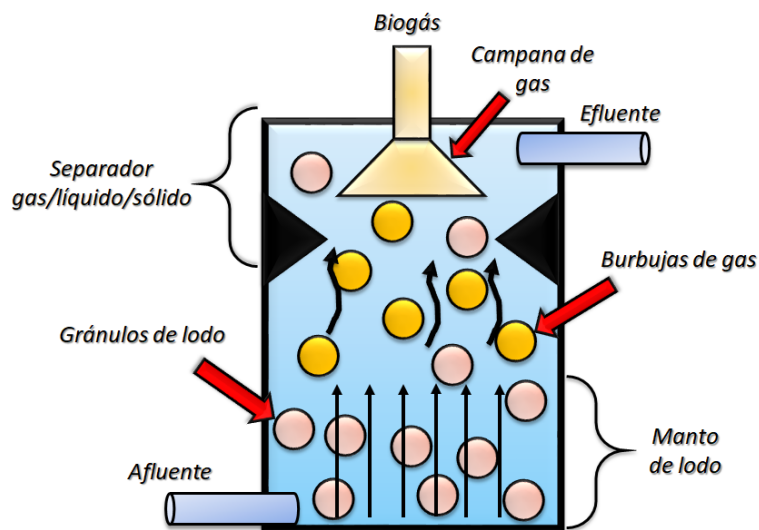


Figura 2.4. Reactor anaerobio de flujo ascendente con manto de lodos (UASB).

4. Fijación de la bacteria en pequeñas partículas de soporte y fluidización para producir la mezcla: *lecho fluidizado o expandido*.

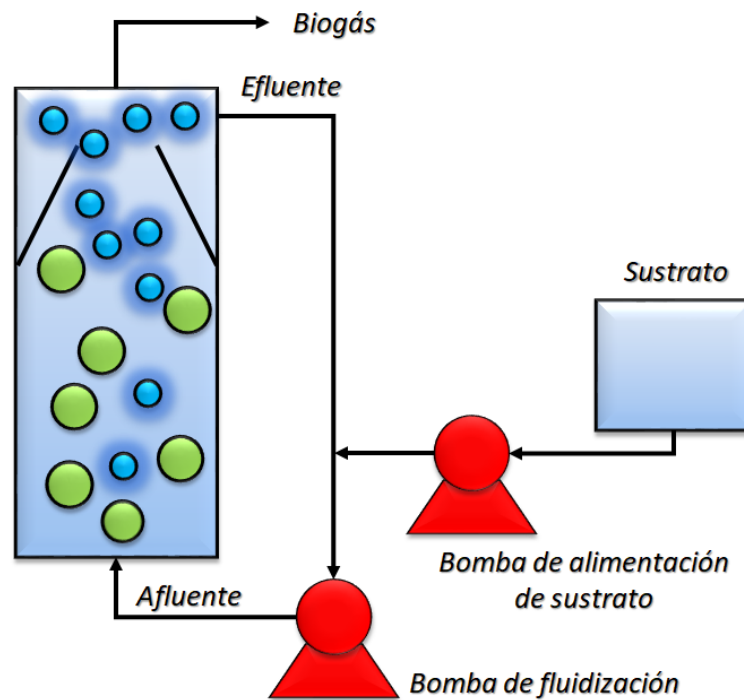


Figura 2.5. Reactor de lecho fluidizado.

#### 2.1.4. Fluidización

Un lecho consiste en una columna formada por partículas sólidas donde un fluido, sea líquido o gas, pasa a través del lecho librando algunas impurezas y, a su vez, sufre una caída de presión. Si el fluido se mueve a velocidades bajas a través del lecho este no produce movimiento de las partículas, pero al ir incrementando gradualmente la velocidad llega un punto donde las partículas no permanecen estáticas, sino que se levantan y se agitan, dicho proceso recibe el nombre de *fluidización*.

A medida que se incrementa la velocidad del fluido también se aumenta el caudal (si el área se mantiene constante). Se pueden distinguir diferentes etapas en el lecho [Meléndez and Gutiérrez, 2004]:

- *Lecho Fijo*: las partículas permiten el paso tortuoso del fluido sin separarse una de otras; esto hace que la altura del lecho se mantenga constante y, por lo tanto, la fracción de vacío en el lecho (porosidad) se mantiene constante. En esta etapa el fluido experimenta la mayor caída de presión del proceso.
- *Lecho prefluidizado*: también es conocido como fluidización incipiente y se trata de un estado de transición entre el lecho fijo y el fluidizado. Una de las características que presenta esta etapa es que la velocidad en este punto recibe el nombre de velocidad mínima de fluidización. También se caracteriza porque la porosidad comienza a aumentar.
- *Fluidización discontinua*: también se conoce como fase densa y es cuando el movimiento de las partículas se hace más turbulento formándose torbellinos. Dentro de esta etapa se pueden distinguir dos tipos de fluidización:
  - *Particulada*: se manifiesta en sistemas líquido-sólido, con lechos de partículas finas en los cuales se manifiesta una expansión suave.

- *Agregativa*: se presenta en sistemas gas-sólido. La mayor parte del fluido circula en burbujas que se rompen en la parte superior dando origen a la formación de aglomerados.
- *Fluidización continua*: todas las partículas son removidas por el fluido, por lo que el lecho deja de existir como tal, mientras que la porosidad tiende a uno.

En la Figura 2.6 se presenta los regímenes de fluidización la cual se debe al aumento de la velocidad del fluido ascendente a través del lecho.

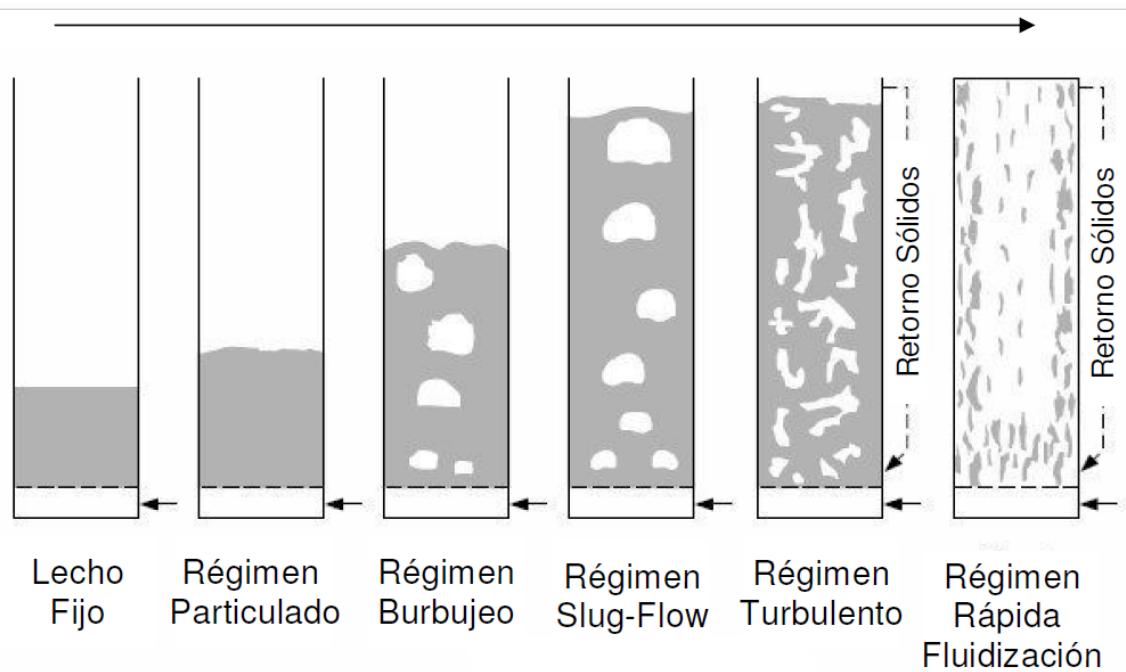


Figura 2.6. Regímenes de fluidización.

### 2.1.5. Reactores fluidizantes

#### 2.1.5.1. Reactor de lecho fluidizado (LF)

En reactores de fluidización ascendente, las partículas sólidas tienen una densidad mayor que el líquido y están fluidizadas por una fuerte corriente de líquido que fluye en dirección opuesta a la gravedad como se observa en la Figura 2.5 [Arnaiz et al., 2003]. El uso de la tecnología de lecho fluidizado para tratamiento de aguas residuales ha sido estudiado a nivel laboratorio y planta piloto, sin embargo, muy pocas plantas industriales han implementado esta tecnología [Karamanev and Nikolov, 1996]. De acuerdo a [Marin et al., 1999] se reconocen las siguientes ventajas de esta tecnología:

- Pueden alcanzarse altas concentraciones de biomasa sobre densos soportes, fácilmente retenidas dentro del reactor.
- Se generan grandes áreas de transferencia de masa a través de partículas de tamaño pequeño que son empleadas como soporte fluidizable.
- Es posible que existan altas velocidades de flujo de fluido, el cual permite el tratamiento de aguas residuales muy diluidas o altas relaciones de recirculación, creando condiciones adecuadas de alcalinidad.
- Existe resistencia a esporádicas sobrecargas orgánicas.

La carencia del éxito de reactores de lecho fluidizado ascendente a nivel industrial, podría ser resultado de una combinación de varios puntos negativos:

- Un alto nivel de mantenimiento derivado de su complejidad de diseño.
- La necesidad de recirculación de líquido o problemas hidrodinámicos.
- Un alto costo energético para la fluidificación.

### 2.1.5.2. Reactor de lecho fluidizado inverso (LFI)

Este tipo de configuración en reactores fluidizantes al presentarse un flujo descendente permite a las partículas sedimentables ser recuperadas en el fondo del reactor; esto se observa en la Figura 2.7. Por otra parte, el líquido y el biogás producido fluyen en direcciones opuestas lo cual ayuda a la expansión del lecho. Este proceso tiene las mismas ventajas que el LF clásico, pero, posee también otras características interesantes [Karamanev and Nikolov, 1996]:

- El lecho se fluidiza más fácilmente, lo que provoca una reducción notable del costo energético.
- La producción de biogás en contracorriente favorece la turbulencia.
- No se requiere un separador sólido/líquido.
- No se requiere un sedimentador.

Al producir tanto biogás en un reactor de tipo LFI, la proporción de gas y líquido en el reactor se modifica considerablemente. El recircular el biogás producido permite alcanzar una excelente fluidificación, reduciendo los gastos ligados a la recirculación de líquido.

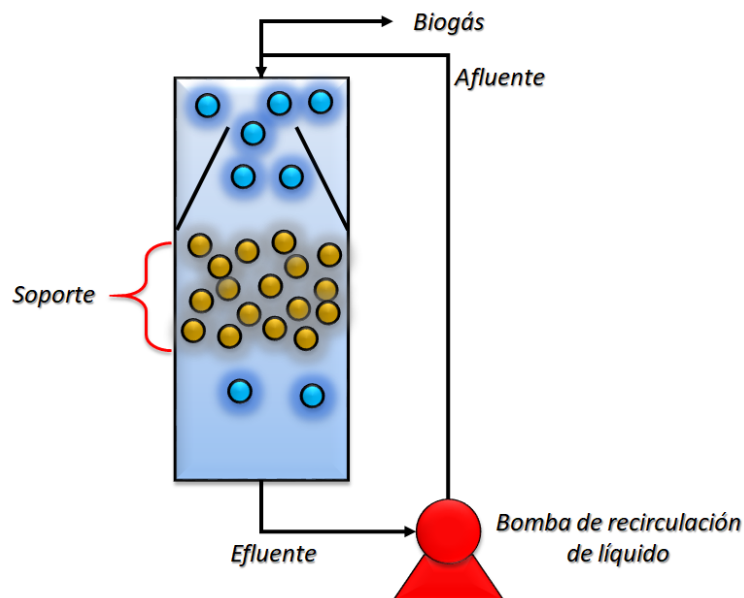


Figura 2.7. Reactor de tipo LFI.



## 2.2. Redes Neuronales

Una red neuronal artificial (RNA), también llamadas sistemas neuronales artificiales, computadores neuronales, procesadores distribuidos en paralelo o modelos conexionistas, son intentos de imitar (simular) al menos la estructura y funciones de sistemas nerviosos y cerebros de criaturas vivas. De forma general se puede decir que una RNA está diseñada para el procesamiento de información o señales.

### 2.2.1. Neurona artificial

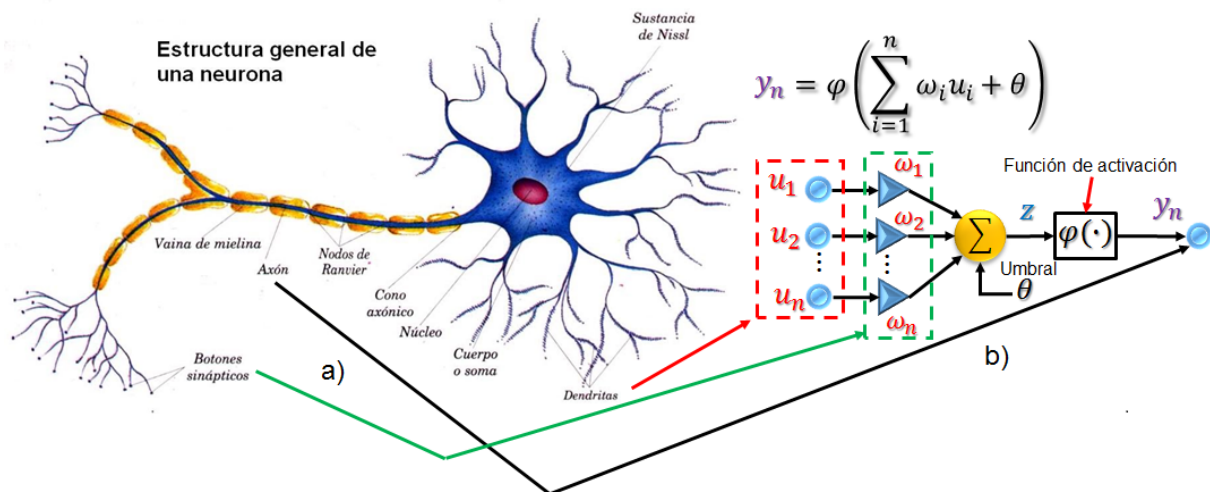


Figura 2.8. a) Neurona biológica b) Neurona artificial.

Una neurona es la unidad fundamental para la operación de una red neuronal donde sus tres componentes principales son: *las dendritas, el cuerpo de la célula o soma, y el axón*. De acuerdo a [Cichocki and Unbehauen, 1993], [Poznyak et al., 2001] y [Sarangapani, 2006], los componentes principales de una neurona biológica trabajan de la siguiente manera: el axón se utiliza para enviar impulsos a otras células nerviosas, cuando el axón está cerca de sus células destino, se divide en muchas ramificaciones que forman sinapsis con las dendritas de otras células. La sinapsis es un pequeño espacio donde se intercambian impulsos entre neuronas. En la Figura 2.8 se observa cómo se relaciona una neurona artificial con la neurona biológica.

### 2.2.2. Modelo Neuronal

Existen distintos modelos neuronales desde arquitecturas simples hasta arquitecturas complejas. Los modelos de neuronas artificiales representan una aproximación al comportamiento de las neuronas biológicas, esta representación es algo difícil de lograr, ya que no se tiene un conocimiento completo sobre el funcionamiento del sistema nervioso. A continuación, se presenta información de algunos modelos neuronales.

#### 2.2.2.1. Modelo Neuronal (Formal) Básico - Modelo de McCulloch-Pitts

La neurona artificial básica más utilizada lleva la estructura de la neurona McCulloch-Pitts, dicha estructura se muestra en la Figura 2.8 b. La neurona puede ser modelada como un equipo de entradas múltiples no lineales interconectadas con  $\omega_i$ , llamados pesos sinápticos. El cuerpo celular (soma) es representado por una limitante no lineal o función de activación  $\varphi(z)$ . De la cual la neurona suma las

enésimas entradas que pasan por los pesos sinápticos y el resultado pasa a través de la no linealidad de acuerdo a la ecuación (2.1):

$$y_n = \varphi \left( \sum_{i=1}^n \omega_i u_i + \theta \right), \quad (2.1)$$

donde  $\varphi$  es la función de activación,  $\theta$  ( $\theta \in \mathfrak{R}$ ) es el umbral,  $u_i$  son las entradas ( $i = .1, 2, \dots, n$ ),  $n$  es el número de entradas,  $y_n$  representa la salida de la neurona [Cichocki and Unbehauen, 1993].

### 2.2.2.2. Modelo Neuronal Discreto

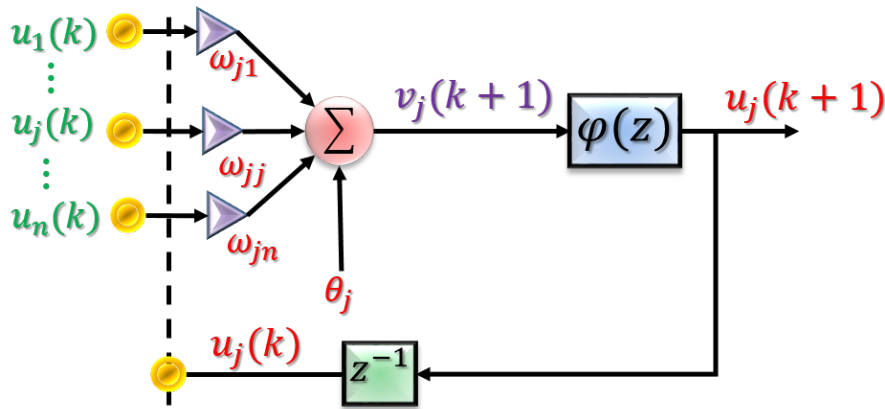


Figura 2.9. Modelo neuronal discreto [Cichocki and Unbehauen, 1993].

En este tipo de modelo, las señales de entrada y salida son adquiridas de manera síncrona con una frecuencia de muestreo constante, por lo tanto, sus valores son calculados en el mismo instante de tiempo discreto  $t_k = k\tau = k/f_c$  ( $k = 0, 1, 2, \dots$ ); donde  $\tau$  es el periodo de muestreo y  $f_c$  es la frecuencia de muestreo. La neurona artificial en tiempo discreto mostrada en la Figura 2.9 está descrita por la ecuación (2.2):

$$u_j(k\tau + \tau) = \varphi \left( \sum_{i=1}^n \omega_{ji} u_i(k\tau) + \theta_j \right),$$

donde

$$u_j(k\tau) = u_j(k),$$

entonces:

$$u_j(k + 1) = \varphi \left( \sum_{i=1}^n \omega_{ji} u_i(k) + \theta_j \right), \quad (2.2)$$

donde  $u_j(k)$  es la salida de la  $j$ -ésima neurona en tiempo discreto,  $n$  representa el número de neuronas,  $\varphi(\cdot)$  representa la función de activación no lineal la cual es comúnmente la función sigmoidea,  $\omega_{ji}$  son pesos sinápticos y  $\theta_j$  el umbral.

### 2.2.2.3. Modelo Neuronal con capacidad de aprendizaje: modelo Perceptrón

Unos de los aspectos más importantes de las neuronas biológicas y artificiales es la capacidad de aprendizaje donde los pesos sinápticos se modifican de acuerdo al algoritmo de adaptación.

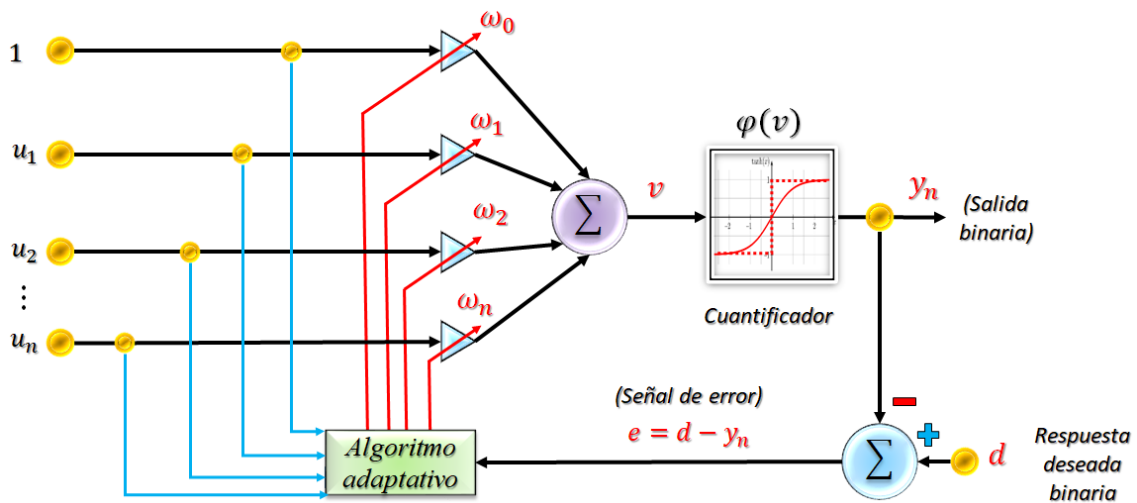


Figura 2.10. Modelo neuronal perceptrón [Cichocki and Unbehauen, 1993].

La Figura 2.10 presenta una neurona artificial tipo **perceptrón**. El primer algoritmo de aprendizaje del perceptrón fue desarrollado por *Frank Rosenblatt* [Cichocki and Unbehauen, 1993]; en donde el algoritmo de aprendizaje actual es el algoritmo de retropropagación descrito por la ecuación (2.3):

$$\omega_i(k + 1) = \omega_i(k) + \eta e(k) \varphi'(v(k)) u_i(k), \quad i = 0, 1, 2, \dots, n. \quad (2.3)$$

Este algoritmo se obtiene definiendo una función de desempeño dada por:

$$E = \frac{1}{2} e^2(k) = \frac{1}{2} (d(k) - y_n(k))^2, \quad (2.4)$$

donde:

$$y_n = \varphi(v(k)),$$

$$v(k) = \sum_{i=0}^n \omega_i(k) u_i(k).$$

Aplicando la expresión (A.12), del algoritmo de gradiente descendente, donde la función de desempeño para los pesos sinápticos,  $\omega_i$ , se minimiza quedando expresado en la ecuación (2.5):

$$\omega_i(k + 1) = \omega_i(k) - \eta \frac{\partial E}{\partial \omega_j(k)}, \quad i = 0, 1, 2, \dots, n, \quad (2.5)$$

del cual:

$$\frac{\partial E}{\partial \omega_i(k)} = \frac{\partial E}{\partial e(k)} \frac{\partial e(k)}{\partial y_n(k)} \frac{\partial y_n(k)}{\partial v(k)} \frac{\partial v(k)}{\partial \omega_i(k)}, \quad (2.6)$$

resolviendo cada derivada parcial:

$$\frac{\partial E}{\partial e(k)} = e(k), \quad \frac{\partial e(k)}{\partial y_n(k)} = -1,$$

$$\frac{\partial y_n(k)}{\partial v(k)} = \varphi'(v(k)), \quad \frac{\partial v(k)}{\partial \omega_i(k)} = u_i(k),$$

donde:

$$\frac{\partial E}{\partial \omega_i(k)} = -e(k) \varphi'(v(k)) u_i, \quad i = 0, 1, 2, \dots, n, \quad (2.7)$$

donde  $u_0 = 1$ ,  $\eta$  regula la velocidad de convergencia y la estabilidad del proceso iterativo de adaptación.

Con lo realizado se obtiene el algoritmo de retropropagación general aplicado a la neurona artificial tipo perceptrón; donde al sustituir la ecuación (2.7) en la ecuación (2.5) se obtiene la expresión (2.3). Existen diferentes tipos de funciones de activación, entre las más comunes se pueden observar en la Figura 2.11.

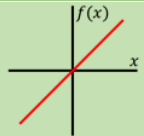
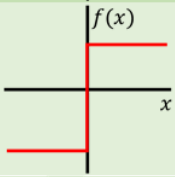
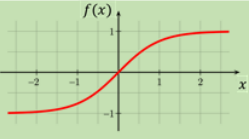

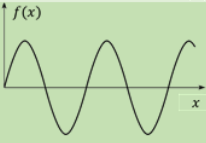
	Función	Rango	Gráfica
<b>Identidad</b>	$y = x$	$[-\infty, \infty+]$	
<b>Escalón</b>	$y = \text{sign}(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
<b>Sigmoidea</b>	$y = \frac{1}{1 + e^{-x}}$ $y = \text{tgh}(x)$	$[0, +1]$ $[-1, +1]$	
<b>Gaussiana</b>	$y = Ae^{-Bx^2}$	$[0, +1]$	
<b>Sinusoidal</b>	$y = A\text{sen}(\omega x + \varphi)$	$[-1, +1]$	

Figura 2.11. Funciones de activación.

Para la comprensión y desarrollo del gradiente descendente puede consultar el anexo A.

### 2.2.3. Estrategia de búsqueda y convergencia

Darken y Moody [Darken and Moody, 1992] propusieron una aproximación simple llamada *estrategia de búsqueda y convergencia*. De acuerdo con esta estrategia la tasa de aprendizaje se reduce gradualmente durante el proceso de aprendizaje. En la *primera fase de aprendizaje* (fase de búsqueda) la tasa de aprendizaje es casi constante, es decir, decrece lentamente y es grande. En la *segunda fase de aprendizaje* (fase de convergencia) la tasa de aprendizaje exponencialmente decrece a cero. Dos posibles programas para la tasa de aprendizaje son propuestos en los trabajos [Darken and Moody, 1992] y [Darken et al., 1992]:

$$\eta^{(k)} = \eta_0 \frac{1}{1 + \frac{k}{k_0}} \quad \text{o} \quad \eta^{(k)} = \eta_0 \frac{1 + \frac{c}{\eta_0} \frac{k}{k_0}}{1 + \frac{c}{\eta_0} \frac{k}{k_0} + k_0 \left(\frac{k}{k_0}\right)^2}, \quad (2.8)$$

donde  $\eta_0 > 0$ ,  $c > 0$ ,  $k_0 \gg 1$  (típicamente  $100 \leq k_0 \leq 500$ ) son parámetros adecuados a elegir.

### 2.2.4. Arquitecturas de redes neuronales y su clasificación

Para que las neuronas artificiales puedan alcanzar su verdadero potencial, estas deben conectarse a otras neuronas artificiales formando distintas arquitecturas de redes neuronales (RNA) [Cichocki and Unbehauen, 1993] y [Poznyak et al., 2001]. Las arquitecturas de las redes neuronales están determinadas por la manera en que se interconectan las neuronas entre sí. Existen distintas arquitecturas de redes neuronales, en la cual, nadie sabe exactamente cuántas existen en la actualidad debido a que se crean nuevas arquitecturas o se modifican las existentes de acuerdo a la aplicación requerida.

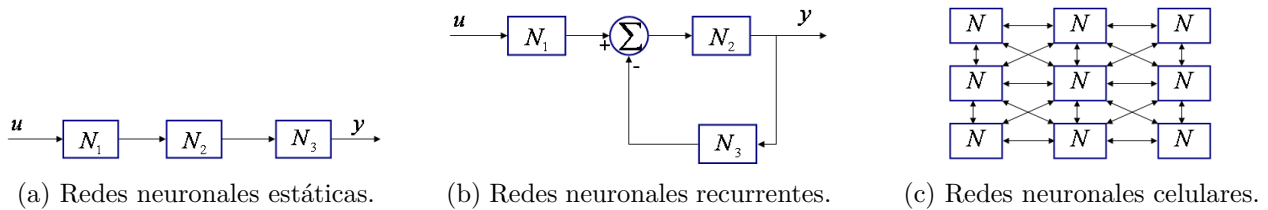


Figura 2.12. Arquitecturas de redes neuronales.

En la Figura 2.12 se presentan 3 diferentes tipos de arquitecturas de RNA que son: *redes neuronales estáticas*, *redes neuronales recurrentes* y *redes neuronales celulares*.

En las *redes neuronales estáticas*, Figura 2.12a, las conexiones entre las neuronas están en línea directa por lo cual no forman lazos de retroalimentación. En este tipo de redes neuronales, usualmente, se produce una respuesta a una entrada rápidamente. Muchas de las arquitecturas neuronales estáticas pueden ser entrenadas utilizando una amplia clase de métodos numéricos convencionales, además, de algoritmos inventados por investigadores de redes neuronales.

En las *redes neuronales recurrentes*, Figura 2.12b, existen lazos de retroalimentación en las conexiones. En algunas redes neuronales recurrentes, cada entrada se presenta en un tiempo, la red neuronal debe hacer iteraciones durante un determinado tiempo antes de producir una respuesta. Las redes neuronales recurrentes son usualmente más difíciles de entrenar que las redes neuronales estáticas.

Las *redes neuronales celulares*, Figura 2.12c, consisten en la conexión de varias neuronas especialmente acomodadas llamadas células. Cada célula está conectada solamente a las células más cercanas, sin embargo, una célula puede afectar a otra célula que no se encuentra conectada directamente mediante la propagación de la señal durante la dinámica de la red neuronal celular. Existen diversas maneras de conectar redes neuronales celulares, las más comunes son: rectangular, triangular, hexagonal y rectangular cuadriculada.

### 2.3. Identificación de sistemas

La identificación de sistemas tiene la tarea de inferir una descripción matemática, *un modelo*, de un sistema dinámico a través de mediciones del sistema. Puede haber diferentes motivos para establecer descripciones matemáticas de sistemas dinámicos. Aplicaciones típicas abarcan lo que son *simulaciones*, *predicciones*, *detección de fallas* y *diseño de sistemas de control*. Si la carga asociada para la construcción de un modelo utilizando leyes de primeros principios es considerado abrumador y complejo, las técnicas de identificación de sistemas son, naturalmente, de interés particular [Nørgård et al., 2000].

Dependiendo del nivel de visión *a priori* sobre el sistema, el problema de identificación se puede abordar de diferentes maneras:

- Si la identificación está basada exclusivamente en datos medidos, asumiendo ningún o poco conocimiento sobre las condiciones y leyes físicas que rigen al sistema, el proceso de identificación es llamado "*modelado de caja negra*".
- La frase "*modelado de caja blanca*", es utilizada para el modelado físico (modelo matemático) de un sistema.
- Cuando hay un cierto nivel de entendimiento sobre el sistema y es utilizado para mejorar el modelado empírico, la frase "*modelado de caja gris*" es utilizado.

De acuerdo a [Ljung, 1999] la relación más básica entre la entrada y la salida de un sistema puede estar descrita por la siguiente *ecuación en diferencias lineal*:

$$y(k) + a_1y(k + 1) + \dots + a_{n_b}y(k - n_b) = b_1u(k - 1) + \dots + b_{n_a}u(k - n_a) \quad (2.9)$$

La ecuación (2.9) representa el sistema en *tiempo discreto*, ya que los datos son adquiridos a través de la toma de muestras. Se considera que el *intervalo de muestras* sea de una unidad de tiempo como se presenta en la ecuación (2.9), sin embargo, puede variar dependiendo del tiempo en el que se adquieran los datos. Una forma usable de la ecuación (2.9) es verla como una forma para *determinar la siguiente salida* dada de la observación anterior:

$$y(k) = -a_1y(k - 1) - \dots - a_{n_b}y(k - n_b) + b_1u(k - 1) + \dots + b_{n_a}u(k - n_a), \quad (2.10)$$

donde:

- $[a_1, \dots, a_{n_b}, b_1, \dots, b_{n_a}]$ , son coeficientes desconocidos para parametrizar un modelo.

### 2.3.1. Identificación de sistemas usando redes neuronales

Las redes neuronales tienen el potencial para ser aplicadas en el modelado de sistemas no lineales. La credibilidad de la identificación sobre las redes neuronales está relacionado a la exclusividad de los valores de los pesos sinápticos y de la capacidad que dos redes neuronales con diferentes parámetros pueden reproducir comportamientos idénticos de entrada y salida del sistema a identificar.

#### 2.3.1.1. Redes neuronales recurrentes

Un enfoque común para cifrar información temporal usando redes neuronales estáticas es incluir retrasos en las entradas y salidas, sin embargo, esta representación está limitada a solo cifrar un número finito de salidas medidas y entradas definidas. Dicho enfoque obstaculiza su uso para la identificación de sistemas dinámicos excepto los de orden relativamente bajo. Como alternativa se utiliza las redes neuronales recurrentes.

Una *red neuronal recurrente o dinámica* se distingue de una red neuronal estática por tener al menos un lazo cerrado. El lazo cerrado envuelve el uso del tiempo discreto como se observa en la Figura 2.9. Existe una gran diversidad de tipos de arquitecturas de redes neuronales recurrentes en la Figura 2.13 se presenta un ejemplo de este tipo de redes.

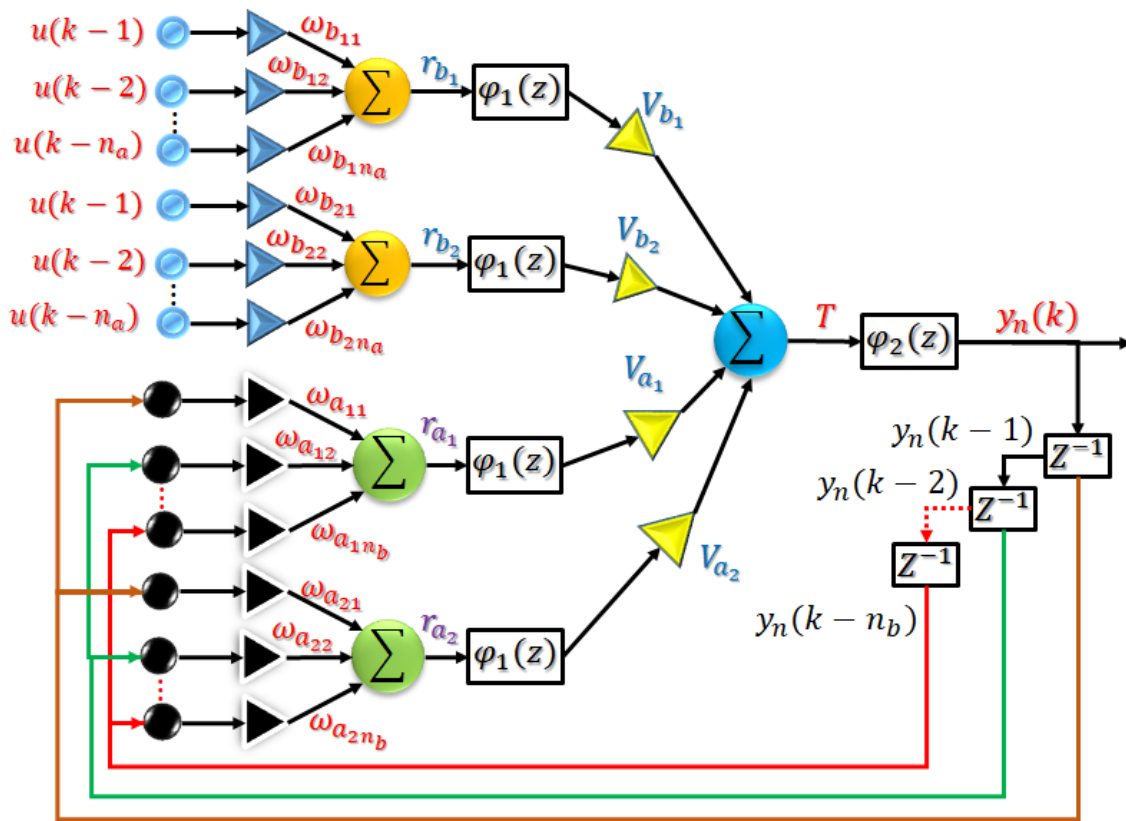


Figura 2.13. RNA recurrente de dos capas modelo perceptrón.

La RNA de la Figura 2.13 utiliza una sola entrada ( $u$ ) con sus respectivos retrasos ( $n_a$ ), donde la misma cantidad de retrasos aplicados se vuelven introducir para la segunda neurona en la capa 1. La salida de la RNA ( $y_n$ ) se retroalimenta con sus respectivos retrasos ( $n_b$ ) en las neuronas de la capa 1. La RNA recurrente realiza un mapeo entrada-salida dado por la ecuación (2.11):

$$y_n(k) = \varphi_2 \left( V_{b_1} \varphi_1(J_u W_{b_1}) + V_{b_2} \varphi_1(J_u W_{b_2}) + V_{a_1} \varphi_1(J_{y_n} W_{a_1}) + V_{a_2} \varphi_1(J_{y_n} W_{a_2}) \right), \quad (2.11)$$

donde:

- $J_u = [u(k-1), u(k-2), \dots, u(k-n_a)]$ , son los datos de entrada pasados.
- $J_{y_n} = [y_n(k-1), y_n(k-2), \dots, y_n(k-n_b)]$ , son los datos de salida pasados de la neurona.
- $W_{b_1}$  y  $W_{b_2}$  son vectores de pesos sinápticos para la entrada.
- $W_{a_1}$  y  $W_{a_2}$  son vectores de pesos sinápticos para la salida neuronal.
- $W_{b_1} = [\omega_{b_{11}}, \omega_{b_{12}}, \dots, \omega_{b_{1n_a}}]^T$ ,      ▪  $W_{b_2} = [\omega_{b_{21}}, \omega_{b_{22}}, \dots, \omega_{b_{2n_a}}]^T$ ,
- $W_{a_1} = [\omega_{a_{11}}, \omega_{a_{12}}, \dots, \omega_{a_{1n_b}}]^T$       ▪  $W_{a_2} = [\omega_{a_{21}}, \omega_{a_{22}}, \dots, \omega_{a_{2n_b}}]^T$ .

2.3.1.2. Arquitecturas de identificación neuronal tipo SISO

Existen diferentes tipos de arquitecturas para identificación de sistemas dinámicos con redes neuronales, en el caso de un sistema de una entrada una salida (SISO) [Narendra and Parthasarathy, 1990], la estructura que pueden tener los modelos son los que se presentan a continuación:

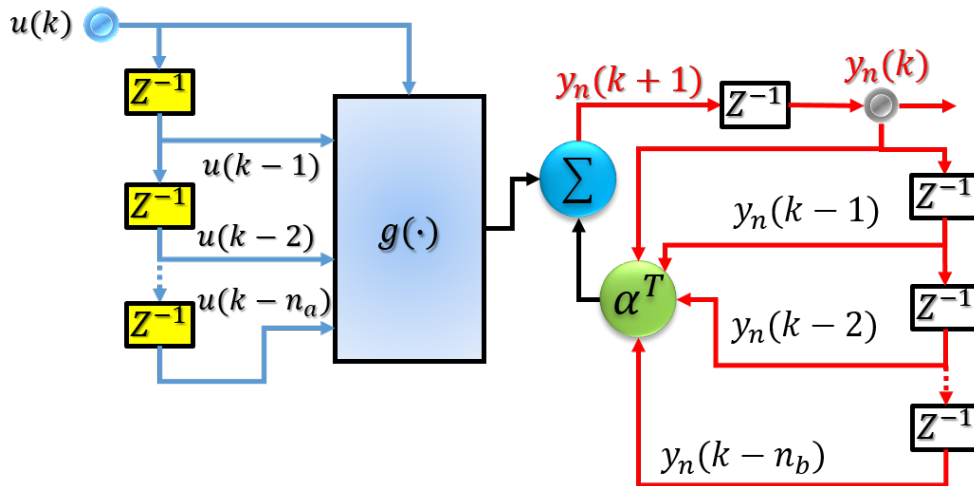


Figura 2.14. Tipo I ecuación (2.12).

$$y_n(k+1) = \sum_{n_b=0}^{n-1} \alpha_{n_b} y_n(k-n_b) + g[u(k), u(k-1), \dots, u(k-n_a+1)], \quad (2.12)$$

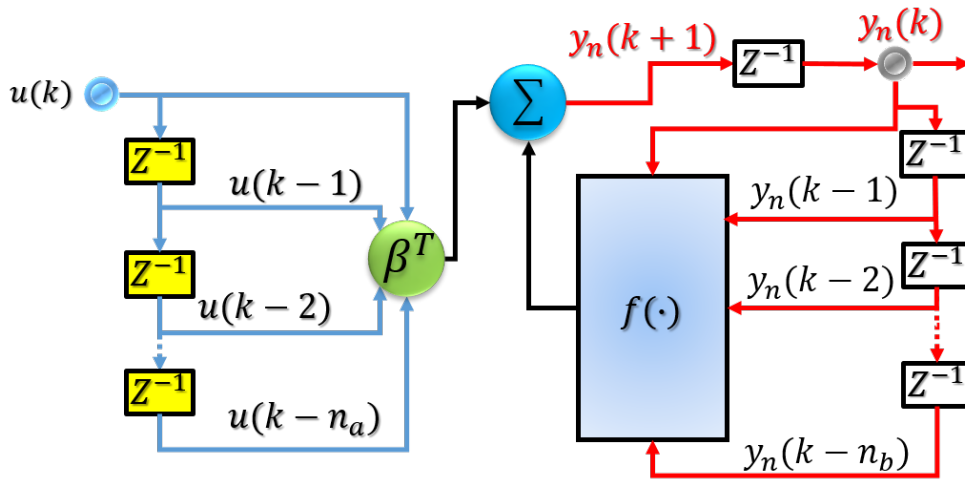


Figura 2.15. Tipo II ecuación (2.13).

$$y_n(k+1) = f[y(k), y(k-1), \dots, y(k-n_b+1)] + \sum_{n_a=0}^{m-1} \beta_{n_a} u(k-n_a), \quad (2.13)$$



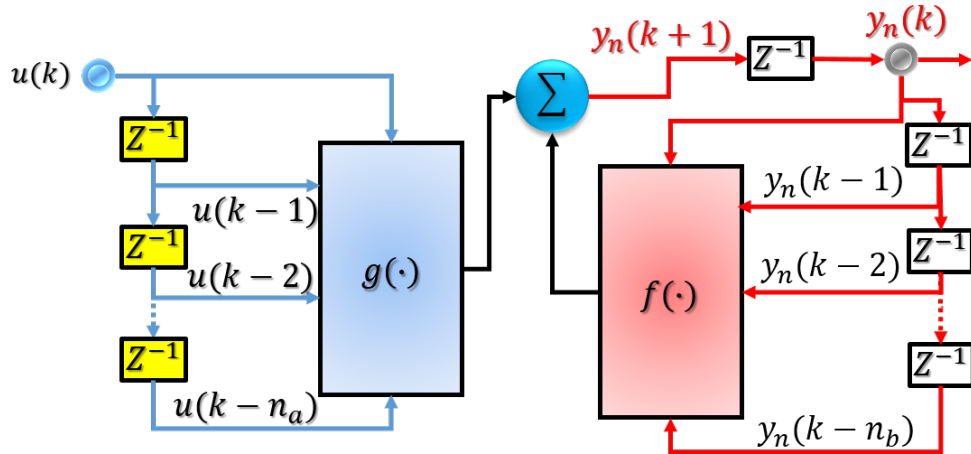


Figura 2.16. Tipo III ecuación (2.14).

$$y_n(k + 1) = f[y(k), y(k - 1), \dots, y(k - n_b + 1)] + g[u(k), u(k - 1), \dots, u(k - n_a + 1)], \quad (2.14)$$

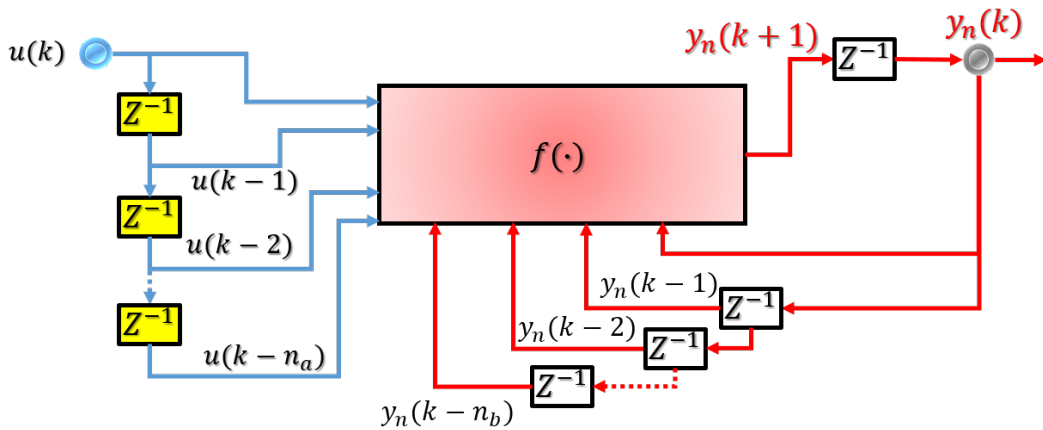


Figura 2.17. Tipo IV ecuación (2.15).

$$y_n(k + 1) = f[y(k), y(k - 1), \dots, y(k - n_b + 1); u(k), u(k - 1), \dots, u(k - n_a + 1)]. \quad (2.15)$$

Se observa en el *modelo I* la dependencia de los valores pasados de la salida son *lineales*; mientras que en el *modelo II* la dependencia de los valores pasados de la entrada se asumen *lineales*. En el caso del *modelo III* la dependencia no lineal de la entrada y la salida se asumen que pueden ser diferentes y en el *modelo IV* hay una función no lineal para la entrada y la salida. Las estructuras de los modelos I al IV se muestran en las Figuras 2.14, 2.15, 2.16 y 2.17.

En los modelos I y II,  $\alpha$  y  $\beta$  son parámetros lineales desconocidos. Las funciones  $f(\bullet)$  y  $g(\bullet)$ , que se presentan en los modelos I al IV, son funciones no lineales que representan la red neuronal, sin embargo, las funciones de activación en la red neuronal recurrente puede ser lineales o no lineales de acuerdo a la Figura 2.11.

2.3.1.3. Entrenamiento neuronal

En el ámbito de redes neuronales artificiales existe un artificio conocido como épocas de entrenamiento, que se define como el ajuste de los pesos sinápticos para todos los pares de entrenamientos, en los cuales la época guarda el último valor de los parámetros de la red neuronal y los actualiza como las condiciones iniciales de los parámetros para la siguiente época y así sucesivamente hasta terminar el número de épocas ejecutadas, esto con el fin de darle más información a la red neuronal. Usualmente una red neuronal requiere muchas épocas para su entrenamiento y lograr minimizar su función objetivo, en pocas palabras, hacer el error cero o cercano a cero [Narendra and Parthasarathy, 1990].

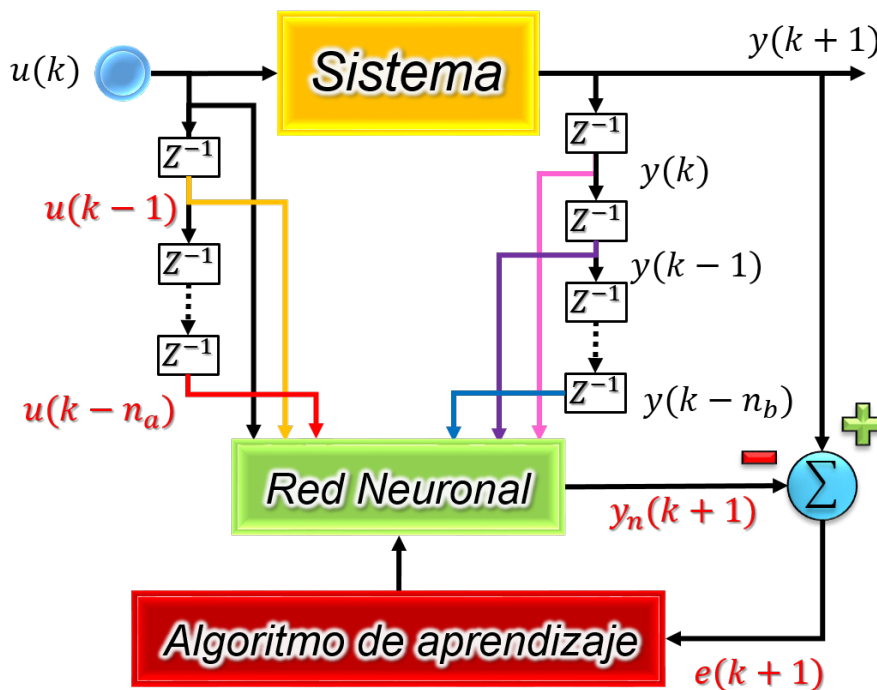


Figura 2.18. Identificación de modelo en esquema serie-paralelo.

En la Figura 2.18 se presenta como se realiza el entrenamiento de la red neuronal basada en los esquemas de identificación SISO. La RNA recibe las señales de entrada,  $u(k), u(k-1), \dots, u(k-n_a)$ , y de salida,  $y(k), y(k-1), \dots, y(k-n_b)$ , del proceso que se utilizan para generar la salida neuronal,  $y_n(k+1)$ , la cual es restada a la salida del proceso  $y(k+1)$  para generar la señal de error  $e(k+1)$  que es utilizada por el algoritmo de aprendizaje para el cálculo y ajuste de los pesos sinápticos de la RNA.

La Figura 2.18 presenta la forma del modelo serie-paralelo que es utilizada para la identificación fuera de línea y en línea. Este esquema utiliza el modelo neuronal tipo IV para realizar la identificación, un ejemplo de este tipo de RNA es la presentada en la Figura 2.13 y se puede apreciar que no presenta uso de umbrales.

Este entrenamiento se repite de acuerdo la cantidad de épocas (repeticiones del entrenamiento) que sean necesarios para que la RNA recurrente pueda aproximar la señal deseada a identificar. Sin embargo, cuando se trata de identificar un modelo de un sistema dinámico con una estructura neuronal definida puede o no puede funcionar para el proceso que se requiere identificar.

En este caso se debe de seguir un procedimiento para la identificación como el que se muestra en la Figura 2.19 y la metodología de identificación de sistemas dinámicos que se presenta en la siguiente sección.

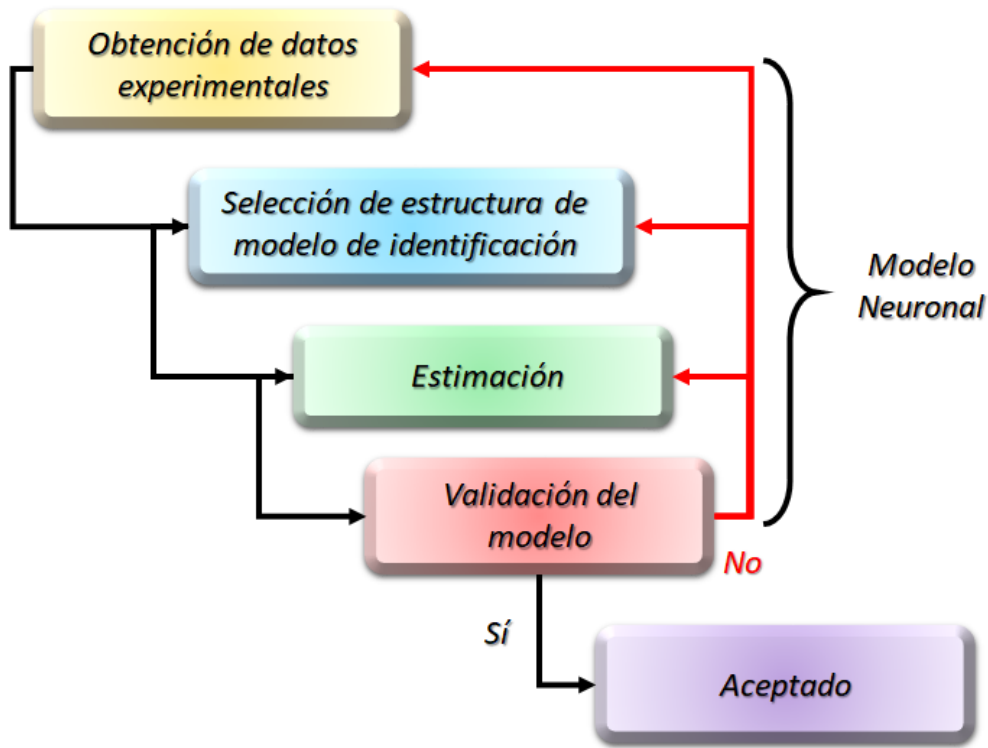


Figura 2.19. Procedimiento de identificación básico usando RNA.

### 2.3.2. Metodología de identificación de sistemas dinámicos

La metodología de identificación de sistemas dinámicos usada es la siguiente de acuerdo a [Ugalde, 2008]:

1. *Obtención del conjunto de datos entrada-salida:* Estos son los datos experimentales medidos de entrada y salida del proceso.
2. *Pre-tratamiento de los datos:* Los datos registrados están generalmente acompañados de señales no deseadas por lo que puede ser necesario realizar algún tipo de filtrado antes de iniciar la identificación.
3. *Seleccionar la arquitectura neuronal:* Consiste en elegir las características de la arquitectura neuronal recurrente, como lo son el número de capas, número de neuronas y funciones de activación a utilizar.
4. *Encontrar el orden del sistema:* Consiste en encontrar el número de atrasos en las entradas  $n_a$  y salidas  $n_b$  adecuados. De acuerdo con la teoría en identificación, *los modelos identificados con retrasos muy grandes presentan más dificultad para aplicar leyes de control* además de la siguiente regla de identificación:

$$\text{retrasos de } u(k) \leq \text{retrasos de } y(k). \tag{2.16}$$

5. *Entrenamiento de la red neuronal*: Consiste en encontrar los pesos sinápticos que mejor ajustan la respuesta del modelo a los datos de entrada-salida obtenidos experimentalmente; los valores iniciales de los pesos sinápticos son aleatorios dentro del rango de  $[-0.5, 0.5]$ , y conforme se entrena la red neuronal, llegan a su valor óptimo.
6. *Validación*: Consiste en aplicar una entrada diferente a la del entrenamiento tanto a la red neuronal como al sistema que se está identificando y realizar las siguientes mediciones de error [Larson, 2002]:

- *Error medio*: Es el promedio del conjunto de datos del error.

$$\bar{e} = \frac{1}{n} \sum_{i=1}^n e_i. \quad (2.17)$$

- *Desviación estándar*: Permite saber que tanto se aleja de la media el error.

$$S_e = \sqrt{\frac{1}{n} \sum_{i=1}^n (e_i - \bar{e})^2}. \quad (2.18)$$

- *Error máximo*: Permite conocer el mayor valor del error.

$$e_{max} = \max(|e|). \quad (2.19)$$

- *Error mínimo*: Permite conocer el menor valor del error.

$$e_{min} = \min(|e|). \quad (2.20)$$

- *FIT*: Indica en porcentaje, que tanto se parece la señal neuronal a la señal real.

$$FIT = 100 \left( 1 - \frac{\|y - \hat{y}\|}{\|y - \frac{1}{N} \sum_{i=1}^N (y_i)\|} \right) \%. \quad (2.21)$$

Si se llega a la conclusión de que el modelo no es válido, se deben revisar los siguientes aspectos como posibles causas:

- El conjunto de datos de entrada-salida no proporciona suficiente información sobre la dinámica del sistema.
- El coeficiente de aprendizaje  $\eta$  no es el adecuado.
- Es necesario aumentar el número de épocas.

### 2.3.3. Índices de desempeño

Los índices de desempeño [Ogata, 2003] son funciones definidas por el usuario que indican la bondad del comportamiento de un sistema de cualquier orden; a su vez también son de utilidad para comparar la similitud entre la respuesta de un sistema (real) y otro (sistema estimado). Entre menor sea el número arrojado por los índices de desempeño mayor será la similitud entre el sistema real y el estimado.

*Integral del error absoluto (IAE, por sus siglas en inglés Integral Absolute Error).*

$$IAE = \int_0^{\infty} |e(t)| dt \rightarrow IAE = \sum_{n_k=1}^N |e(n_k)| \quad (2.22)$$

Pondera de igual manera errores grandes y pequeños, ya que simplemente es la suma del error absoluto (en tiempo discreto).

*Integral del tiempo por el error absoluto (ITAE, por sus siglas en inglés Integral of Time an Absolute Error).*

$$ITAE = \int_0^{\infty} t|e(t)|dt \rightarrow ITAE = \sum_{n_k=1}^N n_k|e(n_k)| \quad (2.23)$$

Es insensible a los errores iniciales, y a veces inevitables, pero penalizan de los errores que permanecen a lo largo del tiempo (duración de las respuestas temporales).

*Integral del error cuadrático (ISE, por sus siglas en inglés Integrated Square Error).*

$$ISE = \int_0^{\infty} e^2(t)dt \rightarrow ISE = \sum_{n_k=1}^N e^2(n_k) \quad (2.24)$$

Se tiene más ponderación para errores grandes, los cuales se presentan generalmente al inicio de la respuesta, y menor ponderación para errores pequeños, los cuales ocurren hacia el final de la respuesta.

*Integral del tiempo por el error cuadrático (ITSE, por sus siglas en inglés Integral of Time and Square Error).*

$$ITSE = \int_0^{\infty} te^2(t)dt \rightarrow ITSE = \sum_{n_k=1}^N n_k e^2(n_k) \quad (2.25)$$

Crece poco conforme aumenta el tiempo, debido a la naturaleza de su expresión; siempre y cuando el error sea pequeño conforme  $n \rightarrow N$ .

*Media del error cuadrático (MSE, por sus siglas en inglés Mean Square Error).*

$$MSE = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T e^2(t)dt \rightarrow \lim_{n_k \rightarrow N} \frac{1}{N} \sum_{k=1}^{n_k} e^2(k) \quad (2.26)$$

Penaliza más errores grandes que pequeños, debido a la naturaleza de su expresión.

*Error medio cuadrático (MCE, por sus siglas en inglés Mean Consequential Error).*

$$MCE = \sqrt{\frac{\sum_{n_k}^N e^2(n_k)}{N}} \quad (2.27)$$

Proporciona el error promedio a lo largo del experimento (duración de las respuestas temporales).

## 2.4. Sumario

Se presentó información sobre el funcionamiento del proceso anaerobio y las diferentes etapas de degradación que puede tener la digestión anaerobia. Una parte fundamental de esta sección es la aplicación que tienen los reactores en procesos anaerobios dando énfasis sobre el comportamiento, las desventajas y ventajas que tienen en los reactores tipo *fluidizantes*.

También, en este capítulo, se definieron conceptos básicos sobre las redes neuronales. Se mostraron ilustraciones sobre tipos de estructuras neuronales, los tipos de algoritmos de aprendizaje usados en las

redes neuronales y las funciones de activación más utilizadas. Además, se presentó como se relacionan las redes neuronales cuando se realiza la identificación de sistemas del tipo caja negra.

Una de las partes fundamentales de la identificación de sistemas usando RNA es el entrenamiento neuronal porque permite a la red neuronal tener una mejor aproximación para la señal deseada. También se mostró los pasos que se siguen para la identificación de sistemas dinámicos; esta metodología permite corroborar si la identificación realizada y la estructura neuronal seleccionada son adecuadas.

En el siguiente capítulo se describe la problemática del tema de tesis con mayor profundidad, además, con lo visto en los capítulos presentados se describirá el procedimiento a utilizar para la identificación del proceso del RAH y se presenta el diseño de la ley de control neuronal para el modelo identificado.

---

# Capítulo 3

---

## Diseño de un esquema de identificación y de un controlador basado en RNA para un RAH

---

En este capítulo se presenta de manera detallada el procedimiento para la solución del tema de investigación. Se inicia con el planteamiento del problema, la cual tiene como meta desarrollar una ley de control utilizando un modelo identificado. Una vez descrito el problema y las limitaciones que hay en la investigación se plantea la propuesta de solución a través de dos hipótesis. Se describe de manera detallada el esquema de identificación neuronal a utilizar. Finalmente se realiza el diseño del tipo de controlador a utilizar.

### 3.1. Planteamiento del problema para la identificación y control del RAH

Para la implementación de una ley de control basado en modelo es indispensable requerir de un modelo matemático que represente de manera precisa al proceso que se busca controlar. Como problemática a solucionar se seleccionó el *reactor anaerobio híbrido (RAH) del ITO*, la cual es un proceso que combina dos tipos de lechos donde información sobre este tipo de combinaciones de procesos es muy escasa y, por consiguiente, solo depende de datos que se generan en las pruebas experimentales. A continuación, se detalla sobre la problemática seleccionada.

#### 3.1.1. Caso de estudio: Reactor anaerobio híbrido (RAH) del Instituto Tecnológico de Orizaba

El RAH opera de la siguiente manera: debido a cierto nivel de concentración que debe de contener el sustrato ( $g\ DQOs/L$ ) de entrada hacia el reactor, este debe de pasar por dos etapas: la etapa de **arranque** (estado transitorio) y la etapa **estable** (lecho fluidizante ya formado).

En el *estado de arranque* del RAH, el sustrato de alimentación debe ser bajo, esto se refiere a dos puntos que son la *concentración de la demanda química de oxígeno soluble ( $DQO_s$ )* y la *carga volumétrica aplicada (CVA)*, estos dos factores juegan un papel importante en el desarrollo del soporte fluidizante.

Una vez que empieza a estabilizarse el proceso anaerobio del RAH, el sistema puede ser alimentado con un CVA mayor, donde el porcentaje de remoción posiblemente puede llegar al 80% y el factor de rendimiento de metano debe de ser capaz de llegar al valor máximo de  $0.35\ LCH_4/g\ DQO_{srem}$ . La ecuación que rige la CVA se muestra en (3.1):

$$CVA\left(\frac{gDQO_{sin}}{L * día}\right) = \frac{DQO_{sin}\left(\frac{gDQO}{L}\right) \times Q\left(\frac{L}{día}\right)}{\text{Vol. útil de trabajo}(L)} \quad (3.1)$$

Cuando el sistema empieza a operar *en la etapa estable*, hay relación entre la densidad de la biopartícula y el porcentaje de remoción de la  $DQO_s$ . El tener un porcentaje de remoción alto indica que el sustrato puede ser consumido con una mayor velocidad, sin embargo, se debe mantener la densidad de la biopartícula porque la partícula está sujeta al soporte fluidizante la cual permite la producción de biogás. Si la biopartícula supera la densidad del agua de  $0.997 \text{ g/cm}^3$  esto puede provocar que el lecho fluidizante se hunda y se recircule llevando al sistema a una inhibición.

Debido a la configuración del reactor no hay forma de conocer el tamaño y la densidad de la biopartícula y solo usando sensores de conductividad sería posible conocer el nivel en el que se ha expandido el lecho. Debido al complicado diseño del RAH, el costo de los electrodos y el ambiente corrosivo que genera el proceso anaerobio por el momento no se han instalado los sensores de conductividad.

En la Figura 3.1 se puede observar cómo está construido el lecho fijo (parte superior) y lecho fluidizante inverso (soporte) del RAH del ITO y el crecimiento de la biopartícula.

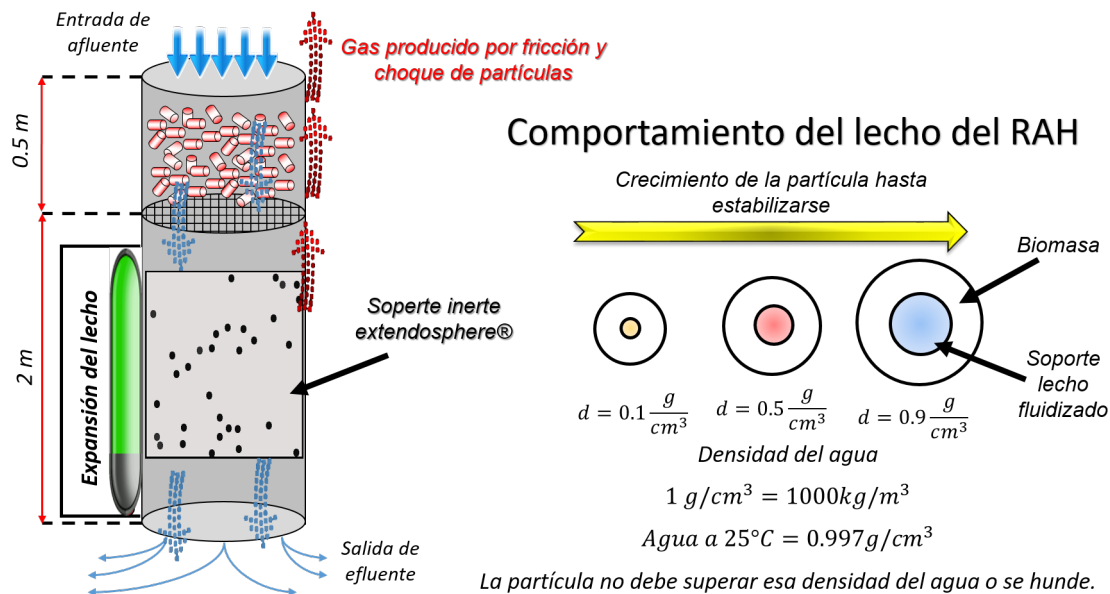


Figura 3.1. Operación del lecho dentro del RAH.

Para mantener el crecimiento de la biopartícula se considera que el reactor siempre está operando entre un 20% y 25% de apertura de las electroválvulas teniendo un flujo de  $0.8 \text{ L/min}$  que serían  $1152 \text{ L/día}$  para la ecuación (3.1). Esta consideración se realiza por las siguientes razones:

- El RAH no cuenta con sensores de flujo instalados debido a su complejo diseño.
- Se realizaron pruebas de paso de flujo con el sensor de flujo  $yf - s201$ , donde el flujo es de  $3.2 \text{ L/min}$  cuando la electroválvula tiene un 100% de apertura. Esta prueba se realizó con agua.
- Se considera que la electroválvula no realiza ningún tipo de cambio en sus aperturas.



Otro aspecto relevante se encuentra en la alimentación del RAH en su *estado estable*. Al desarrollarse un lecho en el estado de arranque la densidad de las biopartículas crece con respecto a la alimentación de CVA y a la fuerza descensional del afluente donde, en procesos anaerobios, entre más producto de sustrato se agrega hay una *mayor producción de biogás*, sin embargo, esto provoca el *crecimiento en la densidad de la biopartícula, aumento de pH, aumento de presión y crecimiento bacteriano (inhibición por sobrealimentación)*.

En el caso de la producción bacteriana suele suceder que existe relación con el biogás, entre menor flujo haya es menor la CVA lo cual la producción de biogás se reduce, pero hay momentos donde sucede una mayor producción de biogás. Sin embargo, si la alimentación CVA se reduce demasiado esto puede provocar una inhibición por falta de sustrato (canibalismo entre las propias bacterias). Lo cual indica que juega un papel muy importante el juicio de alimentación de CVA en este tipo de procesos.

### 3.1.2. Limitaciones del RAH del Instituto Tecnológico de Orizaba

El RAH que se encuentra implementado en el ITO tiene las siguientes limitaciones vistas en la estancia de enero-febrero del año en curso:

- El RAH del ITO no cuenta con sensores de flujo, electrodos de conductividad (usado para conocer la expansión del lecho) y sensores de pH instalados en el reactor.
- De las muestras líquidas que se toman del reactor se puede obtener: *pH, DQOs, % de remoción de DQOs, % de metano, ST, STV y biogás (L)*, que se generan en el proceso y sólo pueden ser procesadas en el laboratorio.
- La toma de las muestras se realiza cada 24 horas debido a la limitación de materiales que son utilizados para su análisis en el laboratorio.
- Debido a la limitación de no tener sensores de flujo se considera como entrada al reactor la CVA.

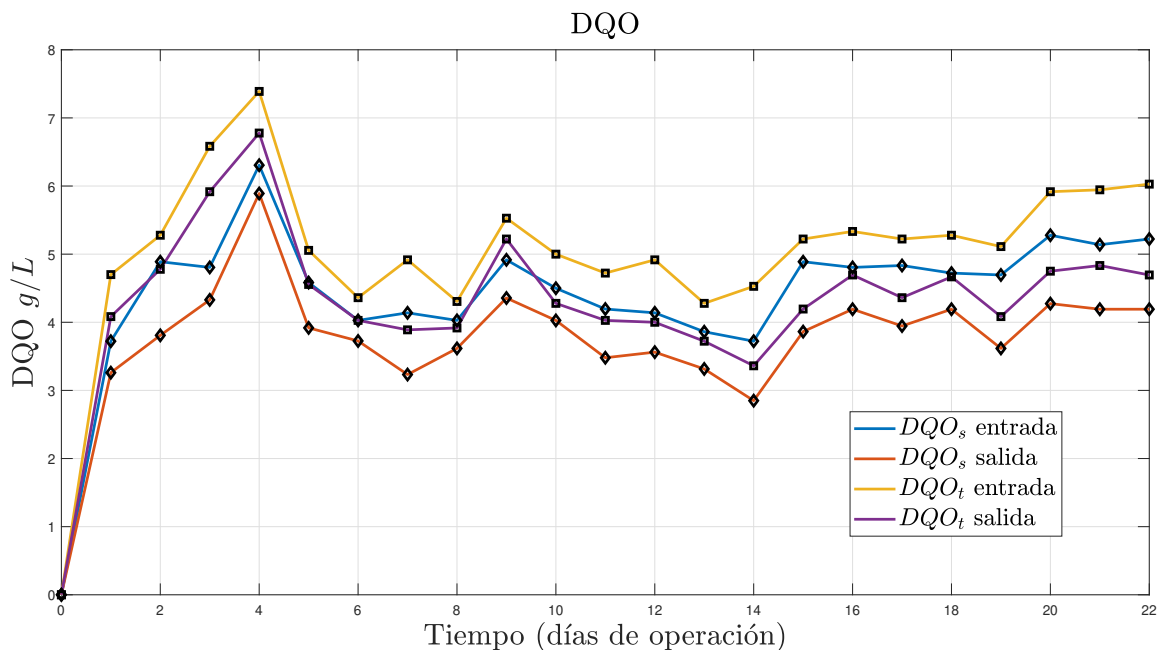


Figura 3.2. Datos de DQO.

Del RAH del ITO solo se tiene la siguiente información reportada en [Marín-Peña, 2017] como se presenta de la Figura 3.2 a la Figura 3.7. Los datos reportados muestran el comportamiento del reactor en la etapa de **arranque**, donde la variable más importante es la  $DQO_s$  de salida, porque representa la cantidad de sustrato que se está reduciendo, dando énfasis que entre más rápido se reduzca el concentrado de sustrato, la biopartícula tendrá una densidad muy cercana a la del agua y el sistema se encontrará muy de cerca de su estado estable de operación.

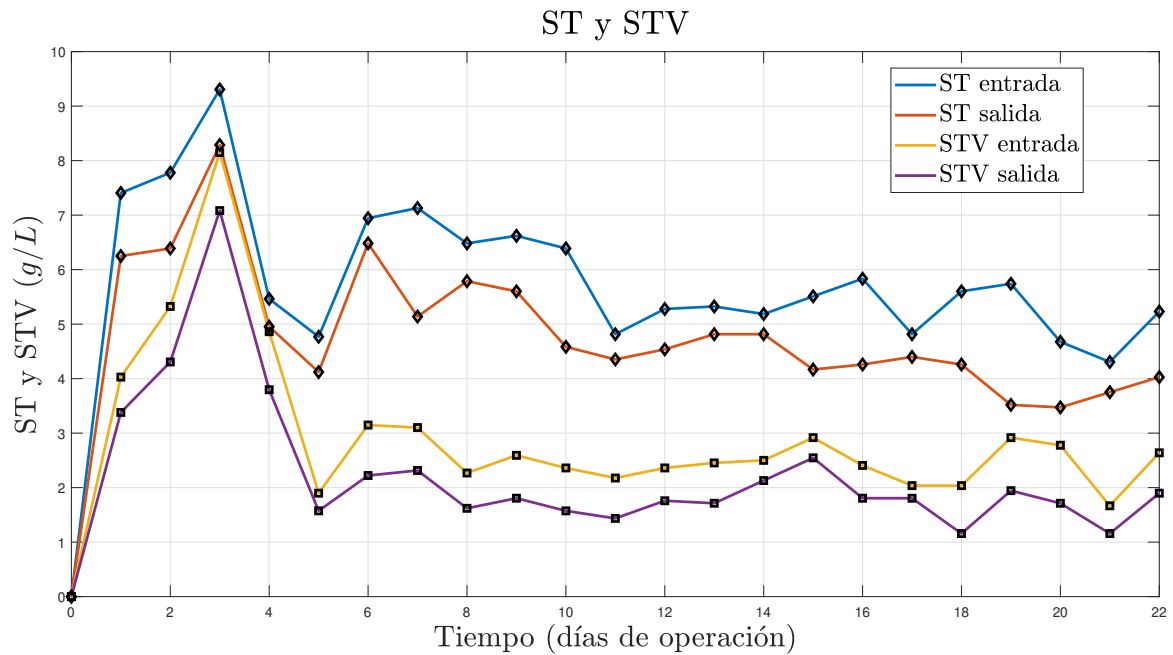


Figura 3.3. Datos de ST y STV.

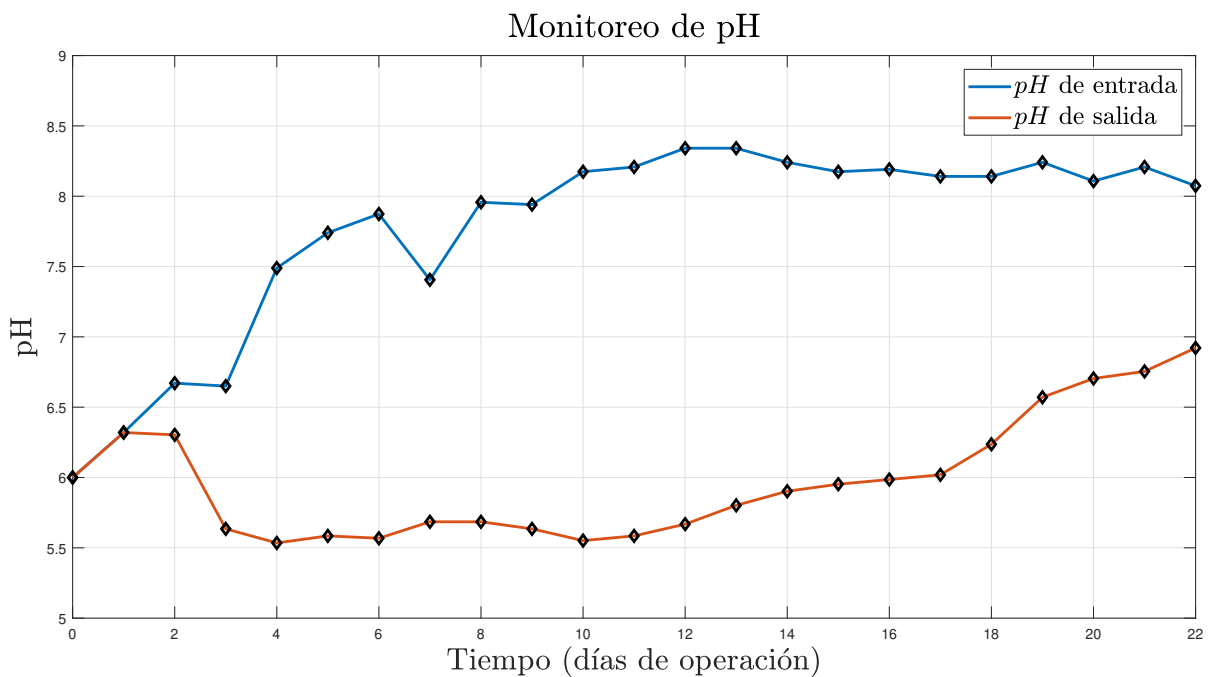


Figura 3.4. Datos de pH.

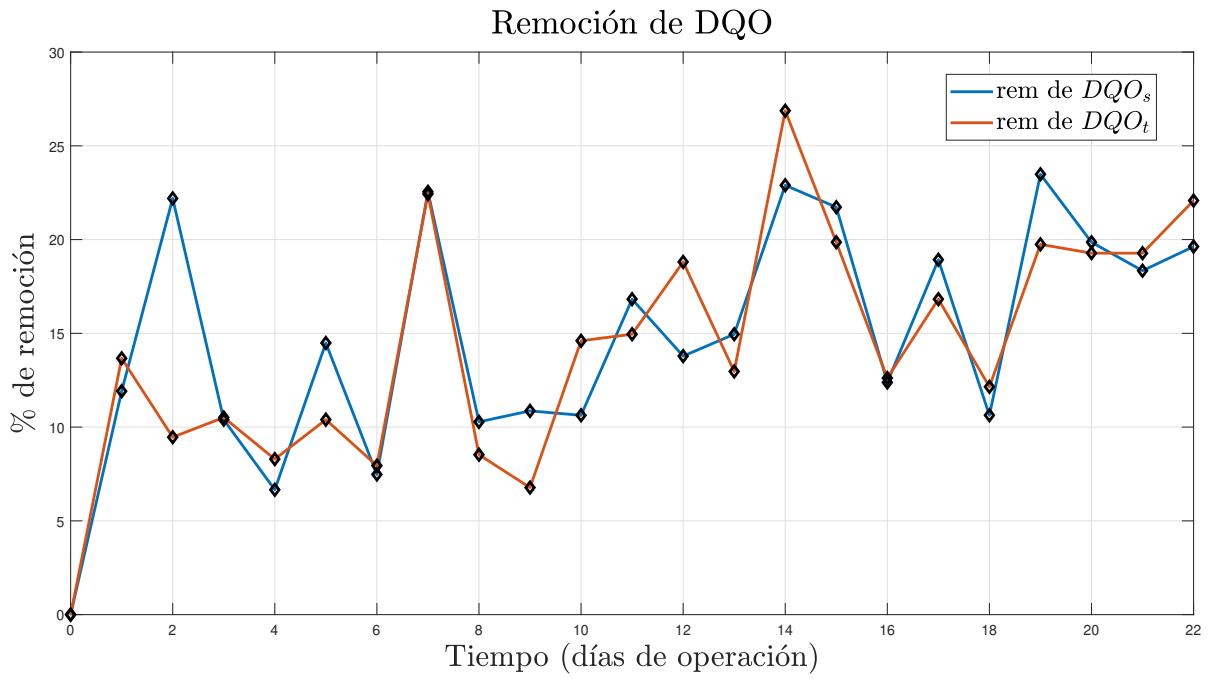


Figura 3.5. Datos de porcentaje de remoción de  $DQO_s$  y  $DQO_t$ .

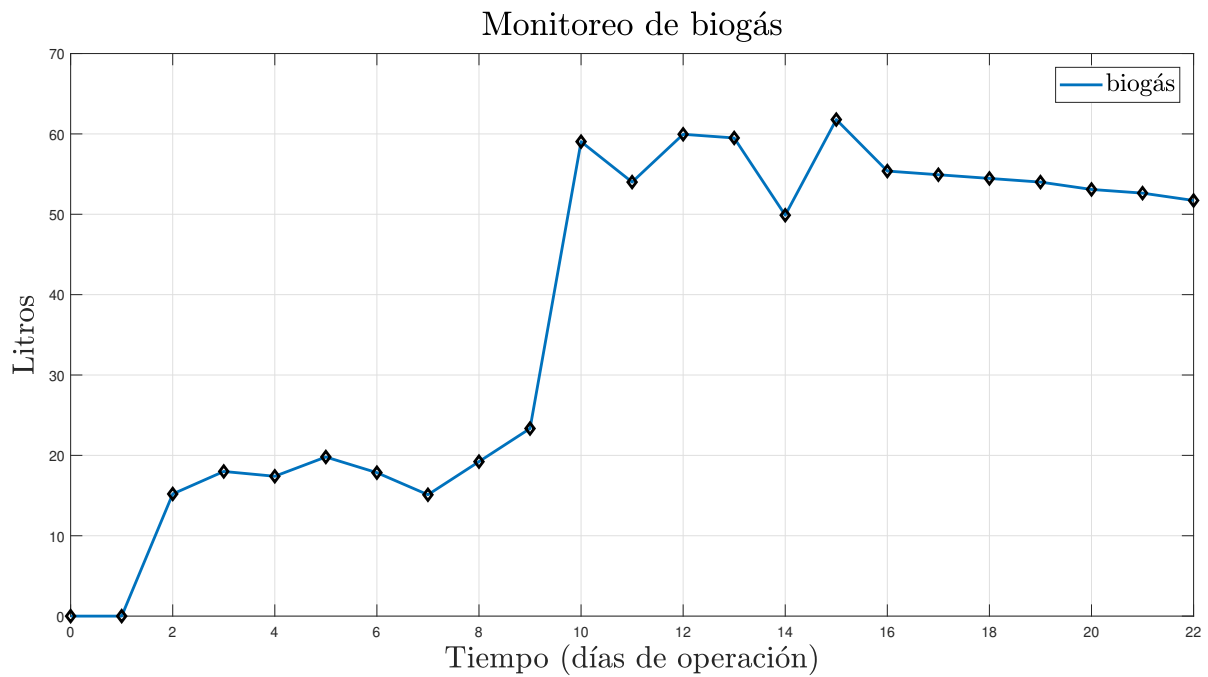


Figura 3.6. Datos de monitoreo de biogás.

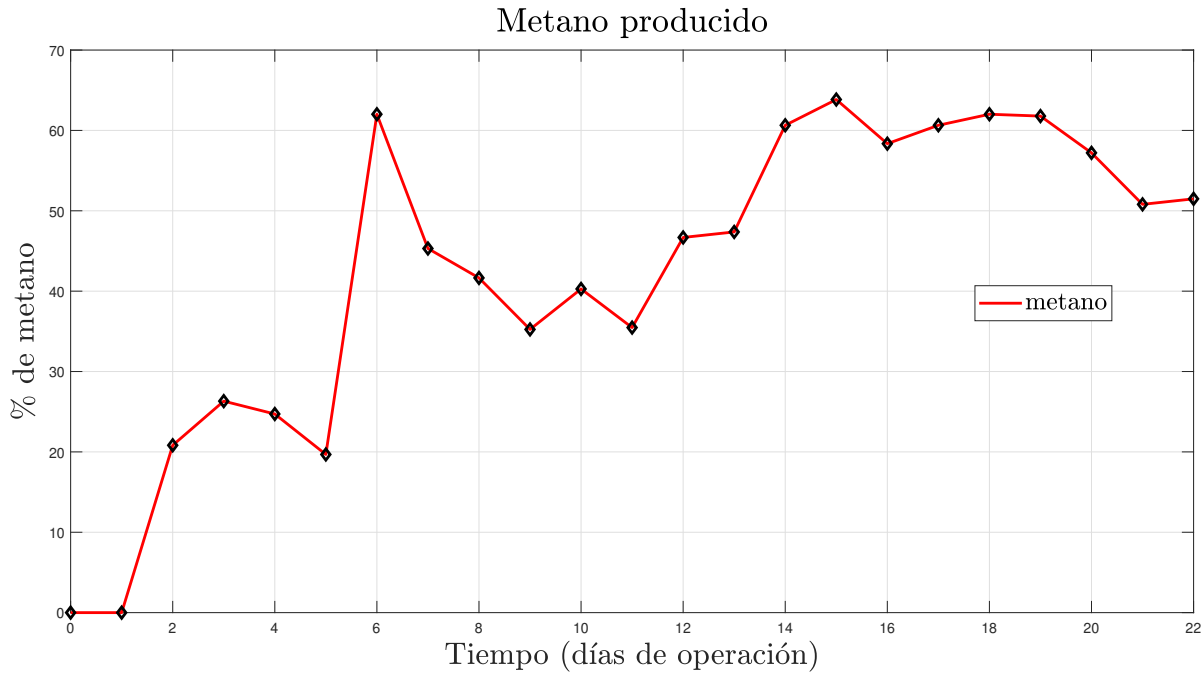


Figura 3.7. Datos de metano producido.

### 3.2. Propuesta de solución para la identificación y control del RAH

Tomando como fundamento los problemas descritos en la sección 3.1 se puede anunciar las siguientes hipótesis:

**Hipótesis 1.** *Con el uso de las RNA enfocada con la metodología de identificación de sistemas es posible obtener una representación del proceso del RAH.*

**Hipótesis 2.** *A partir del modelo identificado es posible diseñar una ley de control usando RNA para controlar el proceso que presenta un comportamiento estocástico.*

### 3.3. Diseño de RNA para la identificación del RAH

Debido a la complejidad para diseñar una estructura adecuada de RNA recurrente, se tomó como referencia utilizar las RNA propuestas en [Ugalde, 2013], en la cual la RNA elegida utiliza el esquema serie-paralelo de la Figura 2.18 para identificar el proceso del RAH, por ello su estructura se presenta en la Figura 3.8 el cual utiliza la forma neuronal tipo IV de identificación SISO presentado en el trabajo de [Narendra and Parthasarathy, 1990]. Esta estructura también se presenta en la Figura 1.5

La RNA, de la Figura 3.8, utiliza una identificación tipo SISO donde la entrada,  $u(k)$ , y la salida,  $y(k + 1)$ , del proceso son alimentados a la RNA con sus respectivos retrasos y la salida neuronal,  $y_n(k + 1)$ , es utilizada para generar el error de comparación,  $e(k + 1)$ , de la Figura 2.18.

La RNA lleva 3 funciones de activación,  $\varphi_1$  es una función no lineal tipo sigmoidea y  $\varphi_2 = \varphi_3$  son funciones lineales tipo identidad, estas condiciones permiten darle mayor libertad de ajuste a la red para generar una mejor identificación.

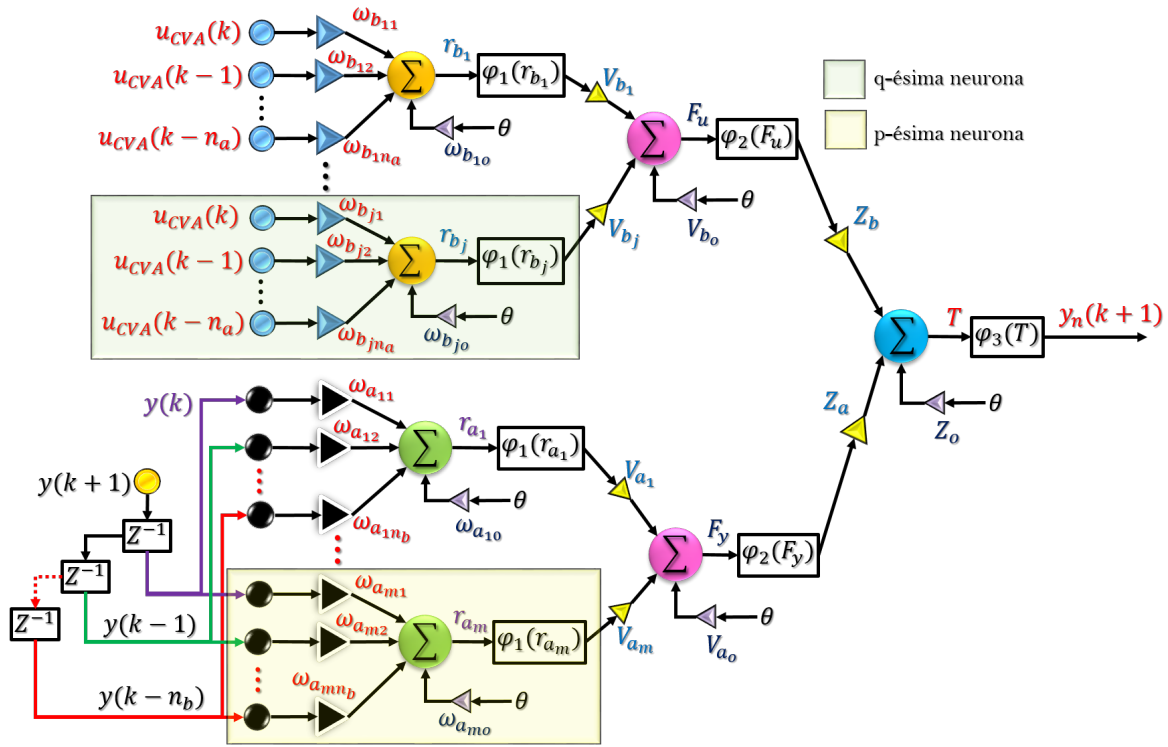


Figura 3.8. RNA recurrente de tres capas esquema de identificación serie-paralelo.

La ecuación que rige la RNA de la Figura 3.8 se muestra en la ecuación (3.2) [Ugalde, 2013], [Cichocki and Unbehauen, 1993]:

$$y_n(k+1) = Z_b \left( \sum_{j=1}^q (V_{b_j} \tanh(J_u W_{b_j} + \theta \omega_{b_{j_o}})) + \theta V_{b_o} \right) + Z_a \left( \sum_{m=1}^p (V_{a_m} \tanh(J_y W_{a_m} + \theta \omega_{a_{m_o}})) + \theta V_{a_o} \right) + \theta Z_o, \quad (3.2)$$

donde:

- $W_{b_j} \in \mathbb{R}^{n_a \times 1}$ ,
- $W_{a_m} \in \mathbb{R}^{n_b \times 1}$ ,
- $J_y \in \mathbb{R}^{1 \times n_b}$ ,
- $J_u = [u_{cVA}(k), u_{cVA}(k-1), \dots, u_{cVA}(k-n_a)]$ ,
- $J_y = [y(k), y(k-1), \dots, y(k-n_b)]$ ,
- $W_{b_j} = [\omega_{b_{j1}}, \omega_{b_{j2}}, \dots, \omega_{b_{j_{n_a+1}}}]^T$ ,
- $W_{a_m} = [\omega_{a_{m1}}, \omega_{a_{m2}}, \dots, \omega_{a_{m_{n_b}}}]^T$ ,
- $J_u \in \mathbb{R}^{1 \times n_a}$ ,
- $\varphi_1 = \tanh(\bullet)$ ,
- $\varphi_2 = \varphi_3 = z$
- $Z_a, Z_b, Z_o \in \mathbb{R}^1$ ,
- $V_{b_j}, V_{a_m}, V_{a_o}, V_{b_o} \in \mathbb{R}^1$ ,
- $\theta = 1$ ,

Para el algoritmo de aprendizaje la RNA recurrente se rige por la siguiente función de desempeño mostrada en (3.3):

$$E(Z_a, Z_b, Z_o, V_{b_j}, V_{a_m}, V_{a_o}, V_{b_o}, W_{b_j}, W_{a_j}, \omega_{a_{m_o}}, \omega_{b_{j_o}}) = \frac{1}{2} e_n^2(k+1) \quad (3.3)$$

donde:

$$e_n = y(k+1) - y_n(k+1).$$

Usando el algoritmo de gradiente descendente discreto, de la ecuación (A.12) del Anexo A, el entrenamiento de los pesos sinápticos queda de la siguiente forma:

$$Z_a(k+1) = Z_a(k) - \eta \frac{\partial E}{\partial Z_a} \Big|_{Z_a=Z_a^{(k)}}, \quad (3.4)$$

$$Z_b(k+1) = Z_b(k) - \eta \frac{\partial E}{\partial Z_b} \Big|_{Z_b=Z_b^{(k)}}, \quad (3.5)$$

$$V_{b_o}(k+1) = V_{b_o}(k) - \eta \frac{\partial E}{\partial V_{b_o}} \Big|_{V_{b_o}=V_{b_o}^{(k)}}, \quad (3.6)$$

$$Z_o(k+1) = Z_o(k) - \eta \frac{\partial E}{\partial Z_o} \Big|_{Z_o=Z_o^{(k)}}, \quad (3.7)$$

$$V_{a_o}(k+1) = V_{a_o}(k) - \eta \frac{\partial E}{\partial V_{a_o}} \Big|_{V_{a_o}=V_{a_o}^{(k)}}, \quad (3.8)$$

$$V_{b_j}(k+1) = V_{b_j}(k) - \eta \frac{\partial E}{\partial V_{b_j}} \Big|_{V_{b_j}=V_{b_j}^{(k)}}, \quad j = 1, 2, \dots, q, \quad (3.9)$$

$$V_{a_m}(k+1) = V_{a_m}(k) - \eta \frac{\partial E}{\partial V_{a_m}} \Big|_{V_{a_m}=V_{a_m}^{(k)}}, \quad m = 1, 2, \dots, p, \quad (3.10)$$

$$W_{b_j}(k+1) = W_{b_j}(k) - \eta \left( \frac{\partial E}{\partial W_{b_j}} \right)^T \Big|_{W_{b_j}=W_{b_j}^{(k)}}, \quad j = 1, 2, \dots, q \quad n_a = 1, 2, \dots, ret_u, \quad (3.11)$$

$$W_{a_m}(k+1) = W_{a_m}(k) - \eta \left( \frac{\partial E}{\partial W_{a_m}} \right)^T \Big|_{W_{a_m}=W_{a_m}^{(k)}}, \quad m = 1, 2, \dots, p \quad n_b = 1, 2, \dots, ret_y, \quad (3.12)$$

$$\omega_{b_{j_o}}(k+1) = \omega_{b_{j_o}}(k) - \eta \frac{\partial E}{\partial \omega_{b_{j_o}}} \Big|_{\omega_{b_{j_o}}=\omega_{b_{j_o}}^{(k)}}, \quad j = 1, 2, \dots, q, \quad (3.13)$$

$$\omega_{a_{m_o}}(k+1) = \omega_{a_{m_o}}(k) - \eta \frac{\partial E}{\partial \omega_{a_{m_o}}} \Big|_{\omega_{a_{m_o}}=\omega_{a_{m_o}}^{(k)}}, \quad m = 1, 2, \dots, p, \quad (3.14)$$

Desarrollando cada derivada parcial de (3.4) a (3.14) se obtiene el siguiente resultado:

Capa 3:

$$\frac{\partial E}{\partial Z_a} = \frac{\partial E}{\partial e_n} \frac{\partial e_n}{\partial y_n} \frac{\partial y_n}{\partial T} \frac{\partial T}{\partial Z_a}, \quad \frac{\partial E}{\partial Z_b} = \frac{\partial E}{\partial e_n} \frac{\partial e_n}{\partial y_n} \frac{\partial y_n}{\partial T} \frac{\partial T}{\partial Z_b}, \quad \frac{\partial E}{\partial Z_o} = \frac{\partial E}{\partial e_n} \frac{\partial e_n}{\partial y_n} \frac{\partial y_n}{\partial T} \frac{\partial T}{\partial Z_o},$$

$$\frac{\partial E}{\partial Z_b} = -e_n(k+1) \sum_{j=1}^q (V_{b_j} \tanh(J_u W_{b_j} + \theta \omega_{b_{j_o}})) + \theta V_{b_o}, \quad (3.15)$$

$$j = 1, 2, \dots, q, \quad n_a = 1, 2, \dots, ret_u,$$

$$\frac{\partial E}{\partial Z_a} = -e_n(k+1) \sum_{m=1}^p (V_{a_m} \tanh(J_y W_{a_m} + \theta \omega_{a_{m_o}})) + \theta V_{a_o}, \quad (3.16)$$

$$m = 1, 2, \dots, p, \quad n_b = 1, 2, \dots, ret_y,$$

$$\frac{\partial E}{\partial Z_o} = -e_n(k+1)\theta. \quad (3.17)$$

Capa 2:

$$\frac{\partial E}{\partial V_{b_j}} = \frac{\partial E}{\partial e_n} \frac{\partial e_n}{\partial y_n} \frac{\partial y_n}{\partial T} \frac{\partial T}{\partial F_u} \frac{\partial F_u}{\partial V_{b_j}}, \quad \frac{\partial E}{\partial V_{a_m}} = \frac{\partial E}{\partial e_n} \frac{\partial e_n}{\partial y_n} \frac{\partial y_n}{\partial T} \frac{\partial T}{\partial F_y} \frac{\partial F_y}{\partial V_{a_m}},$$

$$\frac{\partial E}{\partial V_{a_o}} = \frac{\partial E}{\partial e_n} \frac{\partial e_n}{\partial y_n} \frac{\partial y_n}{\partial T} \frac{\partial T}{\partial F_y} \frac{\partial F_y}{\partial V_{a_o}}, \quad \frac{\partial E}{\partial V_{b_o}} = \frac{\partial E}{\partial e_n} \frac{\partial e_n}{\partial y_n} \frac{\partial y_n}{\partial T} \frac{\partial T}{\partial F_u} \frac{\partial F_u}{\partial V_{b_o}},$$

$$\frac{\partial E}{\partial V_{b_j}} = -e_n(k+1) Z_b (\tanh(J_u W_{b_j} + \theta \omega_{b_{j_o}})), \quad (3.18)$$

$$j = 1, 2, \dots, q, \quad n_a = 1, 2, \dots, ret_u,$$

$$\frac{\partial E}{\partial V_{a_m}} = -e_n(k+1) Z_a (\tanh(J_y W_{a_m} + \theta \omega_{a_{m_o}})), \quad (3.19)$$

$$m = 1, 2, \dots, p, \quad n_b = 1, 2, \dots, ret_y,$$

$$\frac{\partial E}{\partial V_{b_o}} = -e_n(k+1) Z_b \theta, \quad (3.20)$$

$$\frac{\partial E}{\partial V_{a_o}} = -e_n(k+1) Z_a \theta. \quad (3.21)$$

Capa 1:

$$\left( \frac{\partial E}{\partial W_{b_j}} \right)^T = \frac{\partial E}{\partial e_n} \frac{\partial e_n}{\partial y_n} \frac{\partial y_n}{\partial T} \frac{\partial T}{\partial F_u} \frac{\partial F_u}{\partial r_{b_j}} \left( \frac{\partial r_{b_j}}{\partial W_{b_j}} \right)^T, \quad \left( \frac{\partial E}{\partial W_{a_m}} \right)^T = \frac{\partial E}{\partial e_n} \frac{\partial e_n}{\partial y_n} \frac{\partial y_n}{\partial T} \frac{\partial T}{\partial F_y} \frac{\partial F_y}{\partial r_{a_m}} \left( \frac{\partial r_{a_m}}{\partial W_{a_m}} \right)^T,$$

$$\frac{\partial E}{\partial \omega_{b_{j_o}}} = \frac{\partial E}{\partial e_n} \frac{\partial e_n}{\partial y_n} \frac{\partial y_n}{\partial T} \frac{\partial T}{\partial F_u} \frac{\partial F_u}{\partial r_{b_j}} \frac{\partial r_{b_j}}{\partial \omega_{b_{j_o}}}, \quad \frac{\partial E}{\partial \omega_{a_{m_o}}} = \frac{\partial E}{\partial e_n} \frac{\partial e_n}{\partial y_n} \frac{\partial y_n}{\partial T} \frac{\partial T}{\partial F_y} \frac{\partial F_y}{\partial r_{a_m}} \frac{\partial r_{a_m}}{\partial \omega_{a_{m_o}}},$$

$$\left( \frac{\partial E}{\partial W_{b_j}} \right)^T = -e_n(k+1) Z_b V_{b_j} \operatorname{sech}^2(J_u W_{b_j} + \theta \omega_{b_{j_o}}) J_u^T, \quad (3.22)$$

$$j = 1, 2, \dots, q, \quad n_a = 1, 2, \dots, ret_u,$$

$$\left(\frac{\partial E}{\partial W_{a_m}}\right)^T = -e_n(k+1)Z_a V_{a_m} \operatorname{sech}^2(J_y W_{a_m} + \theta \omega_{a_{m_o}}) J_y^T, \quad (3.23)$$

$$m = 1, 2, \dots, p, \quad n_b = 1, 2, \dots, \operatorname{ret}_y,$$

$$\frac{\partial E}{\partial \omega_{b_{j_o}}} = -e_n(k+1)Z_b V_{b_j} \operatorname{sech}^2(J_u W_{b_j} + \theta \omega_{b_{j_o}}) \theta, \quad (3.24)$$

$$j = 1, 2, \dots, q, \quad n_a = 1, 2, \dots, \operatorname{ret}_u,$$

$$\frac{\partial E}{\partial \omega_{a_{m_o}}} = -e_n(k+1)Z_a V_{a_m} \operatorname{sech}^2(J_y W_{a_m} + \theta \omega_{a_{m_o}}) \theta, \quad (3.25)$$

$$m = 1, 2, \dots, p, \quad n_b = 1, 2, \dots, \operatorname{ret}_y.$$

Para la identificación del RAH se utilizará la tasa de aprendizaje basada en la estrategia de búsqueda y convergencia la cual será la siguiente:

$$\eta = \eta_0 \frac{1}{1 + \frac{k}{10k_0}} \quad 0 < \eta_0 < 1, \quad k_0 = \frac{k_{total}}{5} \quad k_{total} = \text{número de datos totales} \quad (3.26)$$

Una vez *identificado el proceso del RAH*, el esquema neuronal de la Figura 3.8 cambia al esquema recurrente de la Figura 3.9 el cual usa los pesos sinápticos ya entrenados y la salida neuronal ( $y_n(k+1)$ ) debe de utilizar los mismos retrasos empleados para la variable de salida del proceso utilizado ( $y(k+1)$ ). Este esquema recurrentes es usado como modelo neuronal identificado para la aplicación de pruebas de estrategias de control que se requieran.

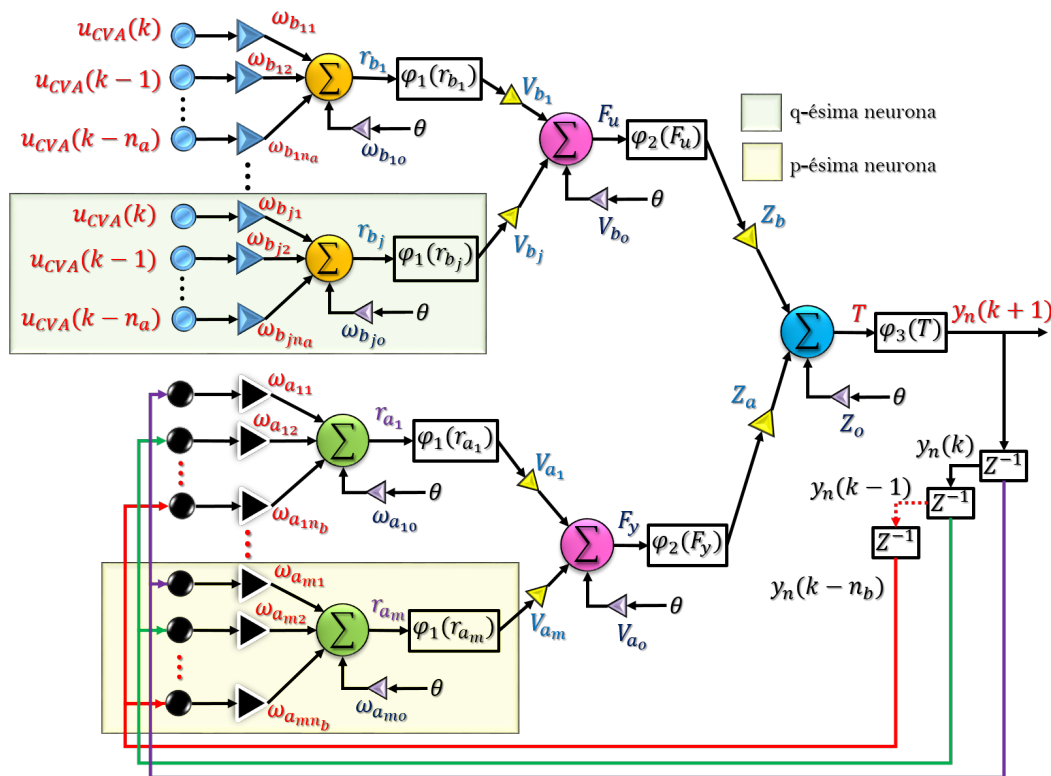


Figura 3.9. RNA recurrente de tres capas.



La ecuación que rige la RNA de la Figura 3.9 se muestra en la ecuación (3.27):

$$y_n(k+1) = Z_b^* \left( \sum_{j=1}^q (V_{b_j}^* \tanh(J_u W_{b_j}^* + \theta \omega_{b_{j_o}}^*)) + \theta V_{b_o}^* \right) + Z_a^* \left( \sum_{m=1}^p (V_{a_m}^* \tanh(J_{y_n} W_{a_m}^* + \theta \omega_{a_{m_o}}^*)) + \theta V_{a_o}^* \right) + \theta Z_o^*, \quad (3.27)$$

donde:

$$\begin{aligned} E(Z_a^*) &= \min_{Z_a} (E\{Z_a\}), & E(Z_b^*) &= \min_{Z_b} (E\{Z_b\}), & E(Z_o^*) &= \min_{Z_o} (E\{Z_o\}), \\ E(V_{b_j}^*) &= \min_{V_{b_j}} (E\{V_{b_j}\}), & E(V_{a_m}^*) &= \min_{V_{a_m}} (E\{V_{a_m}\}), & E(V_{a_o}^*) &= \min_{V_{a_o}} (E\{V_{a_o}\}), \\ E(V_{b_o}^*) &= \min_{V_{b_o}} (E\{V_{b_o}\}), & E(W_{b_j}^*) &= \min_{W_{b_j}} (E\{W_{b_j}\}), & E(W_{a_m}^*) &= \min_{W_{a_m}} (E\{W_{a_m}\}), \\ E(\omega_{a_{m_o}}^*) &= \min_{\omega_{a_{m_o}}} (E\{\omega_{a_{m_o}}\}), & E(\omega_{b_{j_o}}^*) &= \min_{\omega_{b_{j_o}}} (E\{\omega_{b_{j_o}}\}), \end{aligned}$$

- $Z_a^*, Z_b^*, Z_o^*, V_{b_j}^*, V_{a_m}^*, V_{a_o}^*, V_{b_o}^*, W_{b_j}^*, W_{a_m}^*, \omega_{a_{m_o}}^*$  y  $\omega_{b_{j_o}}^*$  son pesos sinápticos ya entrenados,
- $J_{y_n} = [y_n(k), y_n(k-1), \dots, y_n(k-n_b)]$ .

### 3.4. Diseño de controlador neuronal para el RAH

Las aplicaciones de redes neuronales en sistemas de control se han vuelto cada vez más importantes. El procesamiento masivo paralelo, el mapeo no lineal y la habilidad de auto aprendizaje de las redes neuronales han sido factores de motivación para el desarrollo de sistemas de control inteligente. Existe una gran cantidad de formas de estructuras de controladores, de acuerdo con [Poznyak et al., 2001] las estructuras más utilizadas son las siguientes:

- *Control supervisado.*  
Existen plantas donde el humano cierra el lazo de control, debido a la enorme dificultad de implementar un controlador automático usando técnicas estándares. Para esos casos es necesario un controlador que imite las acciones humanas. Una red neuronal es capaz de clonar las acciones humanas.
- *Control inverso directo.*  
En esta estructura, es utilizado directamente un modelo inverso de la planta. Este modelo inverso está en cascada con la planta, el cual el sistema compuesto resulta en un mapeo de identidad entre la salida deseada y la planta. Se basa en gran medida en la calidad del modelo inverso.
- *Control con modelo de referencia.*  
El comportamiento deseado de un sistema en lazo cerrado es especificado a través de un modelo de referencia estable ( $M$ ), usualmente lineal, el cual es definido por una relación entrada-salida  $\{r(k), y^{m_r}(k)\}$ . El objetivo de control es forzar la salida de la planta  $\{y(k)\}$  para que coincida con la salida del modelo de referencia asintóticamente:

$$\lim_{k \rightarrow \infty} \|y^{m_r}(k) - y(k)\| \leq \epsilon, \quad \epsilon \geq 0 \quad (3.28)$$

donde  $\epsilon$  es especificado constante. La Figura 3.10 presenta la estructura del controlador y el error  $E_r$  es usado para entrenar el controlador neuronal.

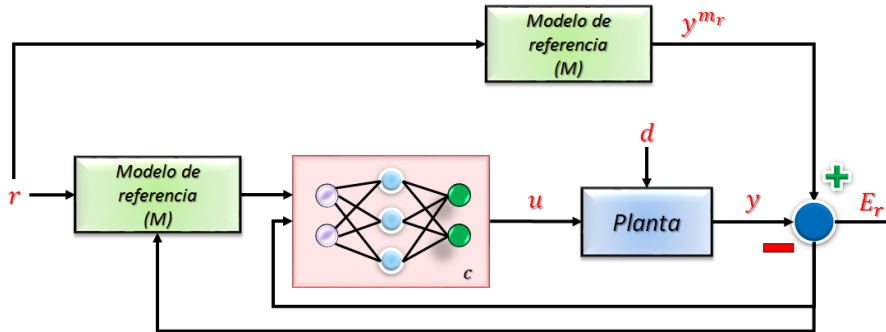


Figura 3.10. Neurocontrol con modelo de referencia.

- *Control de modelo interno (IMC).*

En esta estructura, se usa un modelo de sistema directo e inverso directamente dentro del ciclo de retroalimentación. Donde el modelo de un sistema ( $M$ ) es conectado en paralelo con la planta. La diferencia entre las salidas del sistema y el modelo, es usado como una señal de retroalimentación la cual es procesada por un controlador ( $C$ ); este controlador es implementado como el modelo inverso de la planta. La realización de IMC por redes neuronales es sencilla. El modelo del sistema y su modelo inverso se implementan utilizando redes neuronales como se muestra en la Figura 3.11. Vale la pena señalar que IMC se limita a plantas estables de circuito abierto.

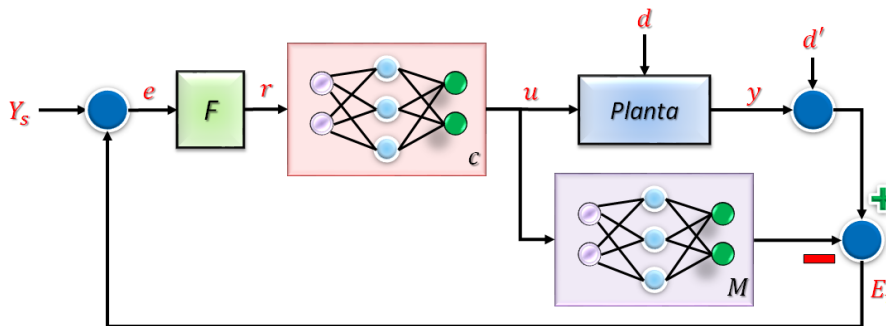


Figura 3.11. Neurocontrol de modelo interno.

- *Control predictivo.*

En esta estructura, un modelo de red neuronal da predicción de la respuesta futura de la planta sobre un horizonte. Estas predicciones se envían a un módulo de optimización para minimizar un criterio de rendimiento donde el resultado de esta optimización es una acción de control apropiada ( $u'$ ). Esta señal de control  $u'$  se selecciona para minimizar el índice.

$$J = \sum_{j=N_1}^{N_2} (y_m(k+j) - y_{mn}(k+j))^2 + \sum_{j=1}^{N_2} \lambda(j)(u'(k+j-1) - u'(k+j-1))^2 \quad (3.29)$$

sujeto a la restricción del modelo dinámico del sistema.

Las constantes  $N_1$  y  $N_2$  definen el horizonte de optimización. Los valores de  $\lambda$  son pesos de las acciones de control. Como se muestra en la Figura 3.12, es posible entrenar una segunda

red neuronal para reproducir las acciones dadas por el módulo de optimización. Una vez que se entrena esta segunda red neuronal, el modelo de planta y la rutina de optimización ya no son necesarias.

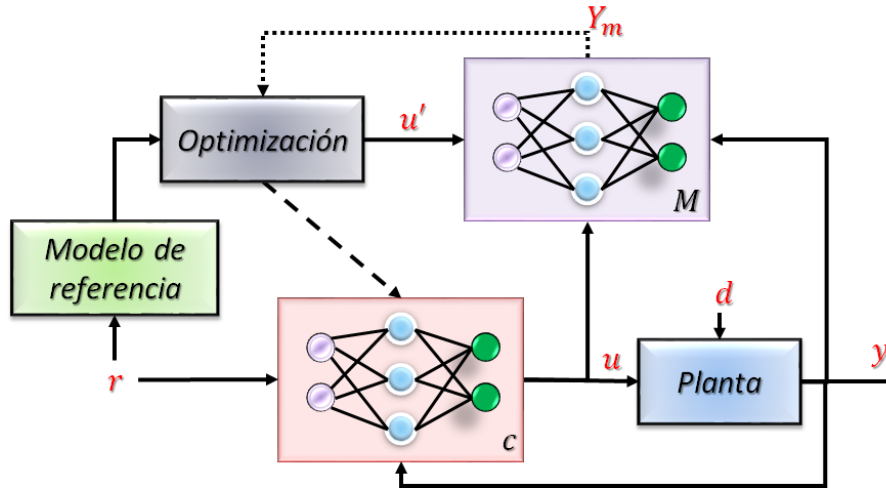


Figura 3.12. Neurocontrol predictivo.

Con los tipos de controladores neuronales presentados, además de las limitaciones y el comportamiento estocástico que tiene como entrada y salida el RAH, se propone utilizar un controlador neuronal tipo directo, también conocido como *modelo neuronal inverso*, donde actualizará sus pesos sinápticos cada vez que se actualicen las muestras de entrada.

El diseño del controlador está basado en el esquema de la RNA de 3 capas de la Figura 3.9 y las señales que entran a la RNA recurrente inversa como entradas, se muestra en la Figura 3.13, son: la salida del proceso ( $y(k + 1)$ ) y la salida de la RNA retroalimentada ( $u_{ne}(k)$ ) con sus respectivos retardos y la referencia ( $r(k + 1)$ ). La estructura de la RNA recurrente del controlador se muestra en la Figura 3.14.

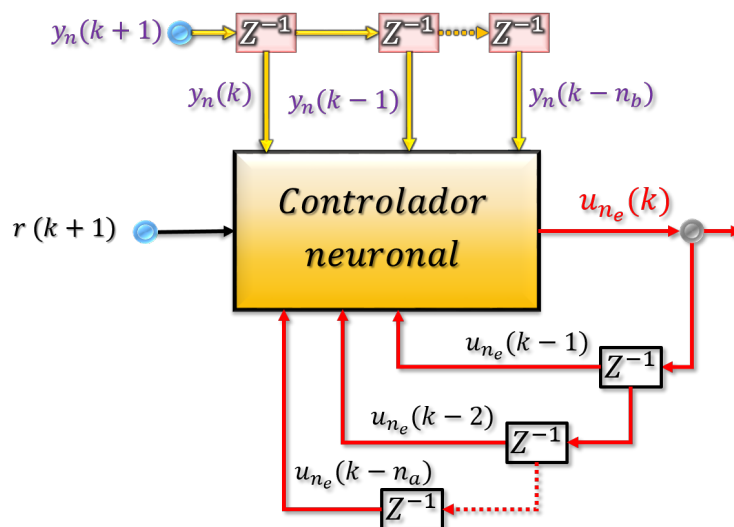


Figura 3.13. Conexión de señales del controlador neuronal.

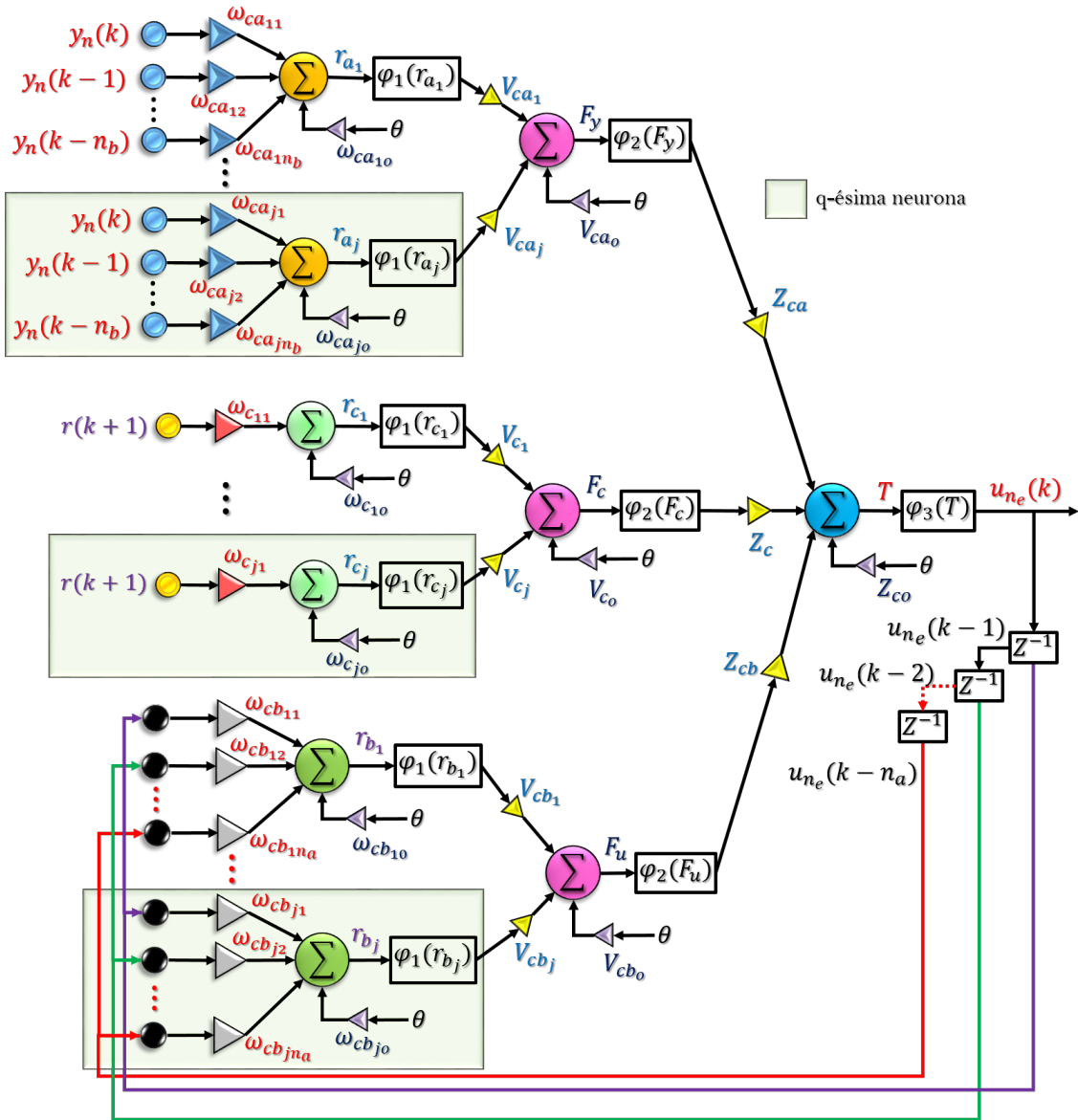


Figura 3.14. RNA del controlador tipo recurrente de tres capas.

La ecuación que rige la RNA de la Figura 3.14 se muestra en la ecuación (3.30):

$$\begin{aligned}
 u_{ne}(k) = & Z_c \left( \sum_{j=1}^q (V_{c_j} \tanh(r(k+1)\omega_{c_{j1}} + \theta\omega_{c_{jo}})) + \theta V_{c_o} \right) \\
 & + Z_{cb} \left( \sum_{j=1}^q (V_{cb_j} \tanh(J_{u_{ne}} W_{cb_j} + \theta\omega_{cb_{jo}})) + \theta V_{cb_o} \right) \\
 & + Z_{ca} \left( \sum_{j=1}^q (V_{ca_j} \tanh(J_{y_n} W_{ca_j} + \theta\omega_{ca_{jo}})) + \theta V_{ca_o} \right) + \theta Z_{co}
 \end{aligned} \tag{3.30}$$

Esta RNA de la ecuación (3.30) tiene ciertas condiciones diferentes a la RNA recurrente de la ecuación (3.2), la RNA tiene el mismo número de neuronas que se definan para la capa 1 y durante el entrenamiento el valor de aprendizaje,  $\eta$ , se considera un valor fijo. De la cual:

- $W_{cb_j} \in \mathbb{R}^{n_a \times 1}$ ,
- $W_{ca_j} \in \mathbb{R}^{n_b \times 1}$ ,
- $J_{y_n} \in \mathbb{R}^{1 \times n_b}$ ,
- $J_{u_{n_e}} \in \mathbb{R}^{1 \times n_a}$ ,
- $\varphi_1 = \tanh(\bullet)$ ,
- $\varphi_2 = \varphi_3 = z$
- $Z_{ca}, Z_{cb}, Z_c, Z_o \in \mathbb{R}^1$ ,
- $V_{cb_j}, V_{ca_m}, V_{ca_o}, V_{cb_o} \in \mathbb{R}^1$ ,
- $\theta = 1$ ,
- $W_{cb_j} = [\omega_{cb_{j1}}, \omega_{cb_{j2}}, \dots, \omega_{cb_{jn_a}}]^T$ ,
- $W_{ca_j} = [\omega_{ca_{j1}}, \omega_{ca_{j2}}, \dots, \omega_{ca_{jn_b}}]^T$ ,
- $J_{u_{n_e}} = [u_{n_e}(k-1), \dots, u_{n_e}(k-n_a)]$ ,
- $V_{CA} = [V_{ca_1}, V_{ca_2}, \dots, V_{ca_p}]$
- $V_{CB} = [V_{cb_1}, V_{cb_2}, \dots, V_{cb_p}]$
- $V_C = [V_{c_1}, V_{c_2}, \dots, V_{c_p}]$

Cuando se aplica el *esquema de control directo* de la Figura 3.15 el error,  $e(k+1)$ , generado de la salida de la planta,  $y(k+1)$ , y la referencia,  $r(k+1)$ , es usado para entrenar los pesos sinápticos de la red neuronal usando el algoritmo de retropropagación. De acuerdo al número de entrenamientos se incrementen, la red neuronal permitirá que el error de la planta se vuelve más pequeña.

Sin embargo, la retropropagación no puede ser usada directamente para entrenar al controlador neuronal. Para ajustar adecuadamente los pesos sinápticos de la red usando el algoritmo de retropropagación, el error de salida en el controlador neuronal directo es  $\epsilon(k) = d(k) - u(k)$ , donde  $d(k)$  es la entrada impulsora deseada a la planta. Puesto que solo el error de salida de la planta  $e(k+1) = r(k+1) - y(k+1)$  es medible o disponible,  $\epsilon(k)$  solo puede ser determinado por la siguiente expresión:

$$\epsilon(k) = e(k+1) \frac{\partial y(k+1)}{\partial u(k)}, \quad (3.31)$$

donde la derivada parcial es el Jacobiano de la planta. Además, esta aplicación de esquema de control directo requiere un conocimiento completo del Jacobiano de la planta [Chan et al., 1993].

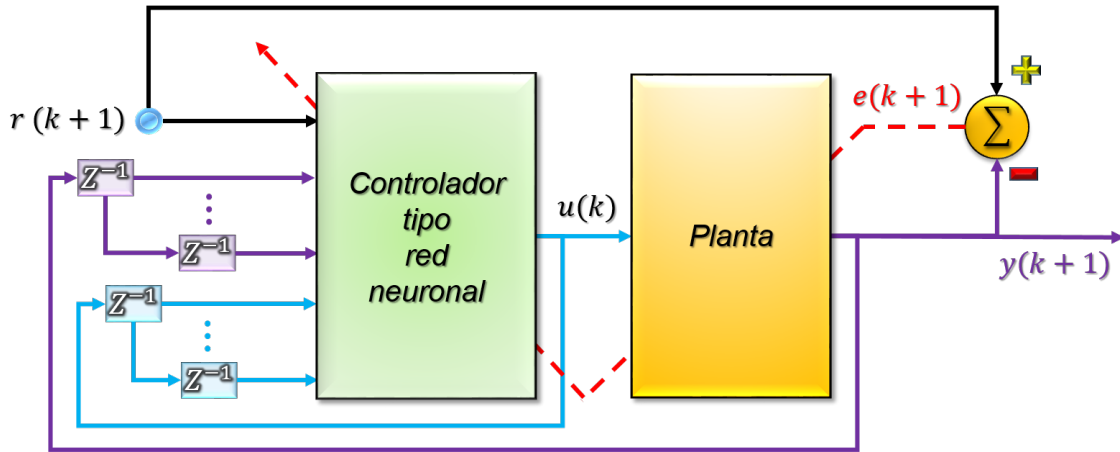


Figura 3.15. Esquema de control directo.

Entonces, en lugar de utilizar el esquema directo de control el *esquema indirecto de control*, de la Figura 3.16, se utiliza como alternativa. Este esquema es aplicable sin tener la necesidad de un modelo de la planta porque no requiere conocer el Jacobiano de la planta, sin embargo, provoca un aumento en la complejidad del diseño de este esquema [Chan et al., 1993].

El esquema indirecto de control requiere del uso de dos redes neuronales: el *controlador neuronal directo* y el *estimador neuronal*. El estimador neuronal es usado como un esquema de identificación de la planta donde el error,  $e_n(k+1) = y(k+1) - y_n(k+1)$ , generado entre la salida de la planta y la RNA de identificación es usado para ajustar sus pesos sinápticos. Para poder realizar la retropropagación para el entrenamiento del controlador neuronal se utiliza el error generado por la referencia y la salida del estimador neuronal:

$$e(k+1) = r(k+1) - y_n(k+1), \quad (3.32)$$

donde al realizar la retropropagación a través de error,  $e(k+1)$ , uno puede realizar el entrenamiento moviéndose a través del estimador neuronal hasta la señal  $u(k)$  que se conecta entre las dos neuronas. Este tipo de entrenamiento es conocido como *entrenamiento neuronal especializado* [Chan et al., 1993], [Zaidi et al., 2019].

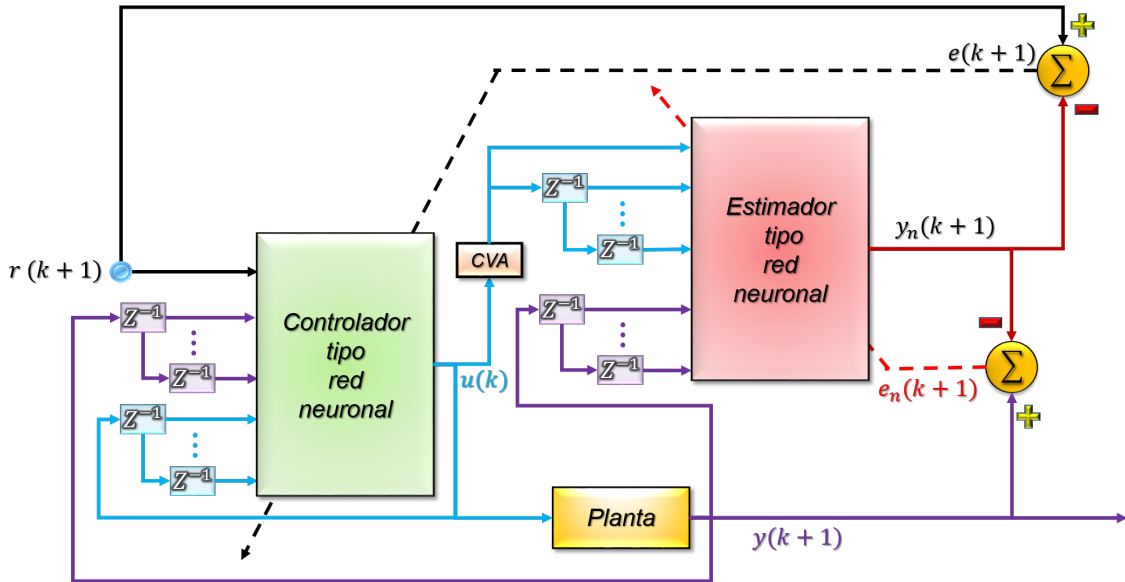


Figura 3.16. Esquema de control indirecto (en base con el esquema de control propuesto Figura 1.3).

Entonces, el algoritmo de aprendizaje, para el controlador neuronal recurrente de la ecuación (3.30), se rige por la siguiente función costo mostrada en (3.33):

$$E(Z_{ca}, Z_{cb}, Z_c, Z_{co}, V_{a_j}, V_{cb_j}, V_{c_j}, V_{ca_o}, V_{cb_o}, V_{c_o}, W_{ca_j}, W_{cb_j}, \omega_{c_{j_1}}, \omega_{a_{j_o}}, \omega_{b_{j_o}}, \omega_{c_{j_o}}) = \frac{1}{2} e^2(k+1), \quad (3.33)$$

donde:

$$e(k+1) = r(k+1) - y_n(k+1).$$

Usando el algoritmo de gradiente descendente discreto se realiza el *entrenamiento especializado en redes de aprendizaje*, en la cual, primero debe de pasar la minimización por la RNA de identificación hasta la señal  $u(k)$  [Zaidi et al., 2019], [Giles et al., 1998]:

$$\frac{\partial E}{\partial u(k)} = \frac{\partial E}{\partial e} \frac{\partial e}{\partial y_n} \frac{\partial y_n}{\partial T} \frac{\partial T}{\partial F_u} \frac{\partial F_u}{\partial r_{b_j}} \frac{\partial r_{b_j}}{\partial \omega_{b_{j_1}}}, \frac{\partial \omega_{b_{j_1}}}{\partial u_{CVA}(k)}. \quad (3.34)$$

Desarrollando cada derivada parcial de la expresión (3.34) se obtiene el siguiente resultado:

$$\frac{\partial E}{\partial u(k)} = -e(k+1)Z_b \sum_{j=1}^q \left( V_{b_j} \operatorname{sech}^2(J_u W_{b_j} + \theta \omega_{b_j o}) u_{CVA}(k) \omega_{b_{j1}} \right), \quad (3.35)$$

$$j = 1, 2, \dots, q, \quad n_a = 1, 2, \dots, \operatorname{ret}_u.$$

Debido a la limitación de no poder implementar el control en línea como se presenta en la Figura 3.16, se opta por utilizar el esquema fuera de línea de la Figura 3.17 para evaluar el desempeño del controlador propuesto. Este esquema tiene relación con la ecuación (3.30) de las Figuras 3.13 y 3.14.

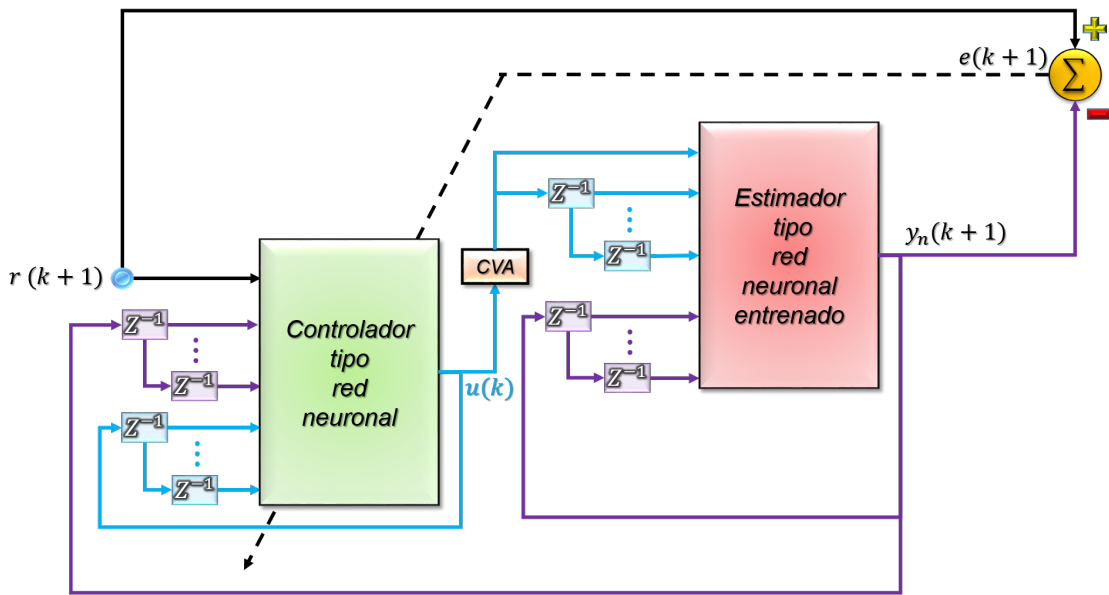


Figura 3.17. Esquema de pruebas para control indirecto (en base con el esquema de control propuesto Figura 1.3).

Además, de acuerdo con [Giles et al., 1998], la conexión entre las dos redes neuronales representa la forma de *redes neuronales en cascadas* en la cual se debe considerar las señales que presentan un atraso durante la retropropagación para el entrenamiento del controlador neuronal como se muestra en la Figura 3.18.

De la cual:

$$\delta_c = \delta_1 + \delta_2 + \dots + \delta_{n_a+1}, \quad n_a = 0, 1, 2, \dots, \operatorname{ret}_u, \quad (3.36)$$

donde:

$$\delta_c = \frac{\partial E}{\partial u(k)} + \frac{\partial E}{\partial u(k-1)} + \dots + \frac{\partial E}{\partial u(k-n_a)}, \quad n_a = 0, 1, 2, \dots, \operatorname{ret}_u, \quad (3.37)$$

$$u(k) := u_{n_e}(k). \quad (3.38)$$

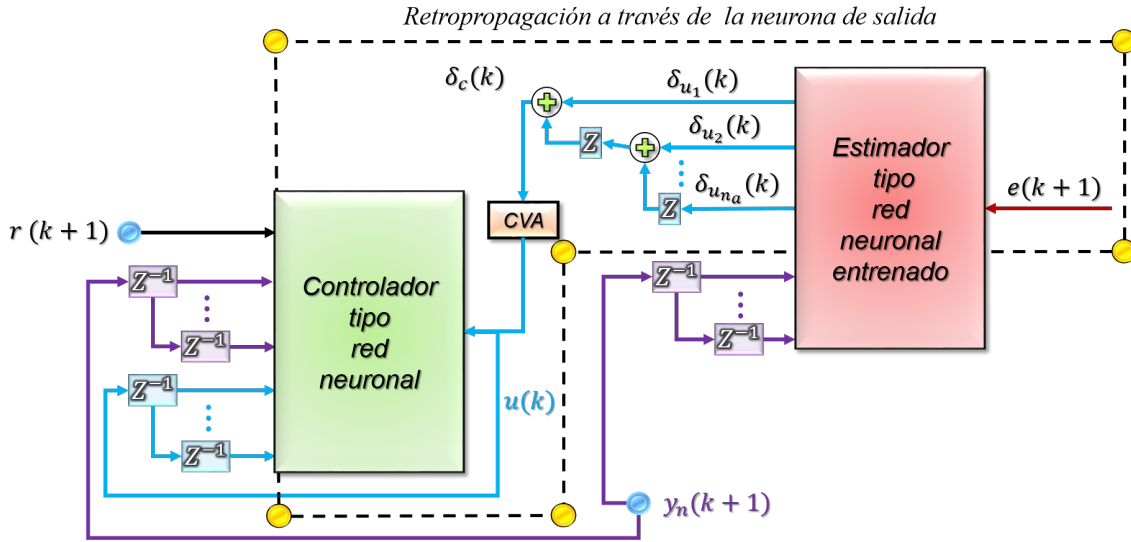


Figura 3.18. Retropropagación para el entrenamiento del controlador neuronal [Giles et al., 1998].

Los entrenamientos de los pesos sinápticos del controlador neuronal quedan de la siguiente forma [Zaidi et al., 2019], [Giles et al., 1998]:

$$Z_{ca}(k+1) = Z_{ca}(k) - \eta \delta_c \frac{\partial u(k)}{\partial Z_{ca}} \Big|_{Z_{ca}=Z_{ca}^{(k)}}, \quad (3.39)$$

$$Z_{cb}(k+1) = Z_{cb}(k) - \eta \delta_c \frac{\partial u(k)}{\partial Z_{cb}} \Big|_{Z_{cb}=Z_{cb}^{(k)}}, \quad (3.40)$$

$$Z_c(k+1) = Z_c(k) - \eta \delta_c \frac{\partial u(k)}{\partial Z_c} \Big|_{Z_c=Z_c^{(k)}}, \quad (3.41)$$

$$Z_{co}(k+1) = Z_{co}(k) - \eta \delta_c \frac{\partial u(k)}{\partial Z_{co}} \Big|_{Z_{co}=Z_{co}^{(k)}}, \quad (3.42)$$

$$V_{ca_o}(k+1) = V_{ca_o}(k) - \eta \delta_c \frac{\partial u(k)}{\partial V_{ca_o}} \Big|_{V_{ca_o}=V_{ca_o}^{(k)}}, \quad (3.43)$$

$$V_{cb_o}(k+1) = V_{cb_o}(k) - \eta \delta_c \frac{\partial u(k)}{\partial V_{cb_o}} \Big|_{V_{cb_o}=V_{cb_o}^{(k)}}, \quad (3.44)$$

$$V_{c_o}(k+1) = V_{c_o}(k) - \eta \delta_c \frac{\partial u(k)}{\partial V_{c_o}} \Big|_{V_{c_o}=V_{c_o}^{(k)}} \quad (3.45)$$

$$V_{ca_j}(k+1) = V_{a_j}(k) - \eta \delta_c \frac{\partial u(k)}{\partial V_{ca_j}} \Big|_{V_{ca_j}=V_{ca_j}^{(k)}}, \quad j = 1, 2, \dots, q, \quad (3.46)$$

$$V_{cb_j}(k+1) = V_{cb_j}(k) - \eta \delta_c \frac{\partial u(k)}{\partial V_{cb_j}} \Big|_{V_{cb_j}=V_{cb_j}^{(k)}}, \quad j = 1, 2, \dots, q, \quad (3.47)$$

$$V_{c_j}(k+1) = V_{c_j}(k) - \eta \delta_c \frac{\partial u(k)}{\partial V_{c_j}} \Big|_{V_{c_j}=V_{c_j}^{(k)}}, \quad j = 1, 2, \dots, q, \quad (3.48)$$

$$W_{ca_j}(k+1) = W_{ca_j}(k) - \eta \delta_c \left( \frac{\partial u(k)}{\partial W_{ca_j}} \right)^T \Big|_{W_{ca_j}=W_{ca_j}^{(k)}}, \quad j = 1, 2, \dots, q \quad n_b = 1, 2, \dots, ret_c, \quad (3.49)$$



$$W_{cb_j}(k+1) = W_{cb_j}(k) - \eta \delta_c \left( \frac{\partial u(k)}{\partial W_{cb_j}} \right)^T \Big|_{W_{cb_j} = W_{cb_j}^{(k)}}, \quad j = 1, 2, \dots, q \quad n_a = 1, 2, \dots, ret_c, \quad (3.50)$$

$$\omega_{c_{j_1}}(k+1) = \omega_{c_{j_1}}(k) - \eta \delta_c \frac{\partial u(k)}{\partial \omega_{c_{j_1}}} \Big|_{\omega_{c_{j_1}} = \omega_{c_{j_1}}^{(k)}}, \quad j = 1, 2, \dots, q, \quad (3.51)$$

$$\omega_{ca_{j_o}}(k+1) = \omega_{ca_{j_o}}(k) - \eta \delta_c \frac{\partial u(k)}{\partial \omega_{ca_{j_o}}} \Big|_{\omega_{ca_{j_o}} = \omega_{ca_{j_o}}^{(k)}}, \quad j = 1, 2, \dots, q, \quad (3.52)$$

$$\omega_{cb_{j_o}}(k+1) = \omega_{cb_{j_o}}(k) - \eta \delta_c \frac{\partial u(k)}{\partial \omega_{cb_{j_o}}} \Big|_{\omega_{cb_{j_o}} = \omega_{cb_{j_o}}^{(k)}}, \quad j = 1, 2, \dots, q, \quad (3.53)$$

$$\omega_{c_{j_o}}(k+1) = \omega_{c_{j_o}}(k) - \eta \delta_c \frac{\partial u(k)}{\partial \omega_{c_{j_o}}} \Big|_{\omega_{c_{j_o}} = \omega_{c_{j_o}}^{(k)}}, \quad j = 1, 2, \dots, q. \quad (3.54)$$

Desarrollando cada derivada parcial de la ecuación (3.39) a la (3.54) se obtiene el siguiente resultado:

Capa 3:

$$\frac{\partial u(k)}{\partial Z_{ca}} = \frac{\partial u(k)}{\partial T} \frac{\partial T}{\partial Z_{ca}}, \quad \frac{\partial u(k)}{\partial Z_{cb}} = \frac{\partial u(k)}{\partial T} \frac{\partial T}{\partial Z_{cb}}, \quad \frac{\partial u(k)}{\partial Z_c} = \frac{\partial u(k)}{\partial T} \frac{\partial T}{\partial Z_c}, \quad \frac{\partial u(k)}{\partial Z_{co}} = \frac{\partial u(k)}{\partial T} \frac{\partial T}{\partial Z_{co}},$$

$$\frac{\partial E}{\partial Z_{ca}} = \sum_{j=1}^q \left( V_{ca_j} \tanh(J_{y_n} W_{ca_j} + \theta \omega_{ca_{j_o}}) \right) + \theta V_{ca_o}, \quad (3.55)$$

$$j = 1, 2, \dots, q, \quad n_b = 1, 2, \dots, ret_c,$$

$$\frac{\partial u(k)}{\partial Z_{cb}} = \sum_{j=1}^q \left( V_{cb_j} \tanh(J_{u_n} W_{cb_j} + \theta \omega_{cb_{j_o}}) \right) + \theta V_{cb_o}, \quad (3.56)$$

$$j = 1, 2, \dots, q, \quad n_a = 1, 2, \dots, ret_c,$$

$$\frac{\partial u(k)}{\partial Z_c} = \sum_{j=1}^q \left( V_{c_j} \tanh(r(k+1)\omega_{c_{j_1}} + \theta \omega_{cb_{j_o}}) \right) + \theta V_{c_o}, \quad j = 1, 2, \dots, q, \quad (3.57)$$

$$\frac{\partial u(k)}{\partial Z_{co}} = \theta. \quad (3.58)$$

Capa 2:

$$\frac{\partial u(k)}{\partial V_{ca_j}} = \frac{\partial u(k)}{\partial T} \frac{\partial T}{\partial F_y} \frac{\partial F_y}{\partial V_{ca_j}}, \quad \frac{\partial u(k)}{\partial V_{cb_j}} = \frac{\partial u(k)}{\partial T} \frac{\partial T}{\partial F_u} \frac{\partial F_u}{\partial V_{cb_j}}, \quad \frac{\partial u(k)}{\partial V_{c_j}} = \frac{\partial u(k)}{\partial T} \frac{\partial T}{\partial F_c} \frac{\partial F_c}{\partial V_{c_j}},$$

$$\frac{\partial u(k)}{\partial V_{ca_o}} = \frac{\partial u(k)}{\partial T} \frac{\partial T}{\partial F_y} \frac{\partial F_y}{\partial V_{ca_o}}, \quad \frac{\partial u(k)}{\partial V_{cb_o}} = \frac{\partial u(k)}{\partial T} \frac{\partial T}{\partial F_u} \frac{\partial F_u}{\partial V_{cb_o}}, \quad \frac{\partial u(k)}{\partial V_{c_o}} = \frac{\partial u(k)}{\partial T} \frac{\partial T}{\partial F_c} \frac{\partial F_c}{\partial V_{c_o}},$$

$$\frac{\partial u(k)}{\partial V_{ca_j}} = Z_{ca} \left( \tanh(J_{y_n} W_{ca_j} + \theta \omega_{ca_{j_o}}) \right), \quad (3.59)$$

$$j = 1, 2, \dots, p, \quad n_b = 1, 2, \dots, ret_c,$$

$$\frac{\partial u(k)}{\partial V_{cb_j}} = Z_{cb} \left( \tanh(J_{u_{ne}} W_{cb_j} + \theta \omega_{cb_{j_o}}) \right), \quad (3.60)$$

$$j = 1, 2, \dots, q, \quad n_a = 1, 2, \dots, ret_c,$$

$$\frac{\partial u(k)}{\partial V_{c_j}} = Z_c \left( \tanh(r(k+1)\omega_{c_{j_1}} + \theta \omega_{c_{j_o}}) \right), \quad j = 1, 2, \dots, q, \quad (3.61)$$

$$\frac{\partial u(k)}{\partial V_{ca_o}} = Z_{ca} \theta, \quad (3.62)$$

$$\frac{\partial u(k)}{\partial V_{cb_o}} = Z_{cb} \theta, \quad (3.63)$$

$$\frac{\partial u(k)}{\partial V_{c_o}} = Z_c \theta. \quad (3.64)$$

Capa 1:

$$\left( \frac{\partial u(k)}{\partial W_{ca_j}} \right)^T = \frac{\partial u(k)}{\partial T} \frac{\partial T}{\partial F_y} \frac{\partial F_y}{\partial r_{a_j}} \left( \frac{\partial r_{a_j}}{\partial W_{ca_j}} \right)^T, \quad \left( \frac{\partial u(k)}{\partial W_{cb_j}} \right)^T = \frac{\partial u(k)}{\partial T} \frac{\partial T}{\partial F_u} \frac{\partial F_u}{\partial r_{b_j}} \left( \frac{\partial r_{b_j}}{\partial W_{cb_j}} \right)^T,$$

$$\frac{\partial u(k)}{\partial \omega_{c_{j_1}}} = \frac{\partial u(k)}{\partial T} \frac{\partial T}{\partial F_c} \frac{\partial F_c}{\partial r_{c_j}} \frac{\partial r_{c_j}}{\partial \omega_{c_{j_1}}}, \quad \frac{\partial u(k)}{\partial \omega_{ca_{j_o}}} = \frac{\partial u(k)}{\partial T} \frac{\partial T}{\partial F_y} \frac{\partial F_y}{\partial r_{a_j}} \frac{\partial r_{a_j}}{\partial \omega_{ca_{j_o}}},$$

$$\frac{\partial u(k)}{\partial \omega_{cb_{j_o}}} = \frac{\partial u(k)}{\partial T} \frac{\partial T}{\partial F_u} \frac{\partial F_u}{\partial r_{b_j}} \frac{\partial r_{b_j}}{\partial \omega_{cb_{j_o}}}, \quad \frac{\partial u(k)}{\partial \omega_{c_{j_o}}} = \frac{\partial u(k)}{\partial T} \frac{\partial T}{\partial F_c} \frac{\partial F_c}{\partial r_{c_j}} \frac{\partial r_{c_j}}{\partial \omega_{c_{j_o}}},$$

$$\left( \frac{\partial u(k)}{\partial W_{ca_j}} \right)^T = Z_{ca} V_{ca_j} \operatorname{sech}^2(J_{y_n} W_{ca_j} + \theta \omega_{ca_{j_o}}) J_y^T, \quad (3.65)$$

$$j = 1, 2, \dots, q, \quad n_b = 1, 2, \dots, ret_c,$$

$$\left( \frac{\partial u(k)}{\partial W_{cb_j}} \right)^T = Z_{cb} V_{cb_j} \operatorname{sech}^2(J_{u_{ne}} W_{cb_j} + \theta \omega_{cb_{j_o}}) J_u^T, \quad (3.66)$$

$$j = 1, 2, \dots, q, \quad n_a = 1, 2, \dots, ret_c,$$

$$\frac{\partial u(k)}{\partial \omega_{c_{j_1}}} = Z_c V_{c_j} \operatorname{sech}^2(r(k+1)\omega_{c_{j_1}} + \theta \omega_{c_{j_o}}) r(k+1), \quad j = 1, 2, \dots, q, \quad (3.67)$$

$$\frac{\partial u(k)}{\partial \omega_{ca_{j_o}}} = Z_{ca} V_{ca_j} \operatorname{sech}^2(J_{y_n} W_{ca_j} + \theta \omega_{ca_{j_o}}) \theta, \quad (3.68)$$

$$j = 1, 2, \dots, p, \quad n_b = 1, 2, \dots, ret_c,$$

$$\frac{\partial u(k)}{\partial \omega_{cb_{j_o}}} = Z_{cb} V_{cb_j} \operatorname{sech}^2(J_{u_{n_e}} W_{cb_j} + \theta \omega_{cb_{j_o}}) \theta, \quad (3.69)$$
$$j = 1, 2, \dots, q, \quad n_a = 1, 2, \dots, ret_c,$$

$$\frac{\partial u(k)}{\partial \omega_{c_{j_o}}} = Z_c V_{c_j} \operatorname{sech}^2(r(k+1)\omega_{c_{j_1}} + \theta \omega_{c_{j_o}}) \theta, \quad j = 1, 2, \dots, q. \quad (3.70)$$

# Capítulo 4

## Indentificación y control del RAH

Este capítulo se enfoca a la validación en simulación de los esquemas de indentificación y del esquema de control propuesto en este trabajo de tesis. Se revisarán escenarios donde la indentificación sea adecuada de acuerdo a los datos proporcionados y el controlador sea capaz de llevar a la referencia deseada en cualquier día de operación del RAH. La programación y la simulación se realizan con el programa de MATLAB<sup>®</sup>.

### 4.1. Identificación del RAH utilizando RNA recurrentes

Para realizar una indentificación de sistemas es necesario contar con una extensa cantidad de datos de acuerdo a lo mencionado en la sección 2.3.2, porque al no tener una cantidad suficiente de datos se llega a la conclusión de que la indentificación no se puede realizar. Este caso sucede con los datos proporcionados debido a que solo se tienen 22 *muestras*, por consiguiente, se realizó una *interpolación lineal* (anexo A.5) de los datos proporcionados incrementándolos a 220 datos como en la Figura 4.1.

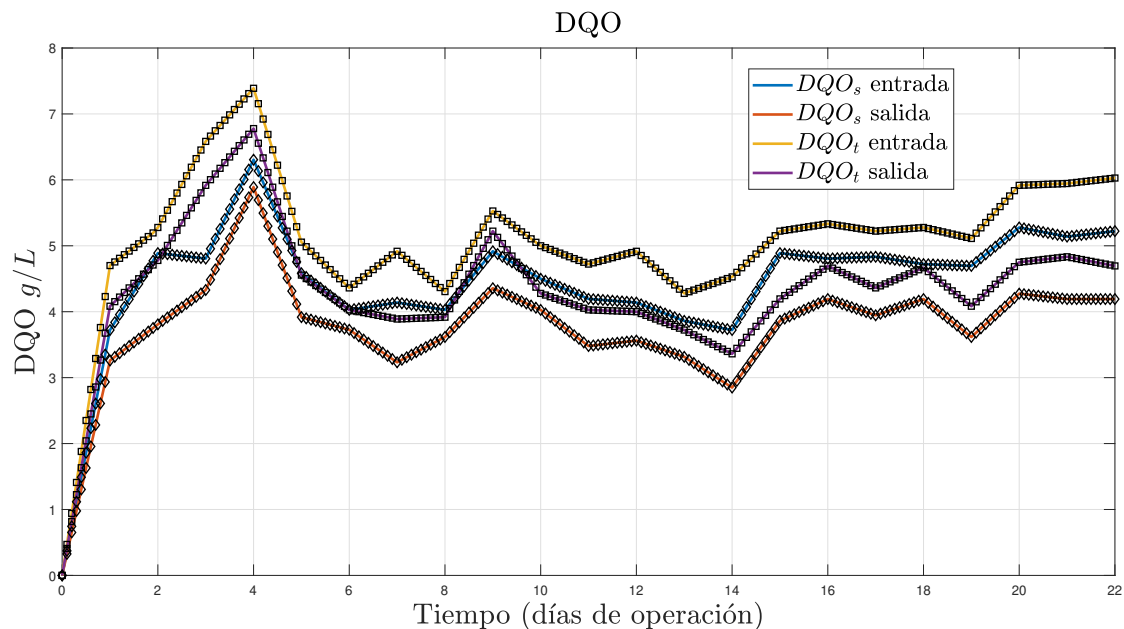


Figura 4.1. Datos de DQO interpolados.

La interpolación permite a la indentificación probarse en diferentes escenarios la toma de adquisición de datos en el RAH de acuerdo a las limitaciones de la toma de muestras en el ITO y tiempo en que se tarda en procesar las muestras en el laboratorio. Con la interpolación realizada se tiene que los datos son adquiridos cada 2 horas y 24 minutos, es decir, aproximadamente cada dos horas y media.

### 4.1.1. Tratamiento de datos

Siguiendo con la metodología de identificación de sistemas dinámicos, se decidió trabajar con una RNA recurrente de 3 capas como se muestra en la Figura 3.9, donde utiliza una función activación *sigmoidea* ( $\tanh(\bullet)$ ) en la capa 1. Para que la identificación sea adecuada y el modelo obtenido se pueda utilizar para las pruebas de control se realiza lo siguiente: los *datos deben de estar escalados* dentro de la región de operación de la tangente hiperbólica.

El área de operación de la tangente hiperbólica se encuentra en un rango de  $[-1, 1]$ , es deseable que los datos escalados se encuentren dentro del rango de  $[-0.5, 0.5]$ , como se muestra en la Figura 4.2, donde permite que la identificación sea más rápida. La escala aplica tanto a la entrada como a la salida de los datos experimentales, si no se realiza esta condición resulta más complicado ajustar la señal deseada.

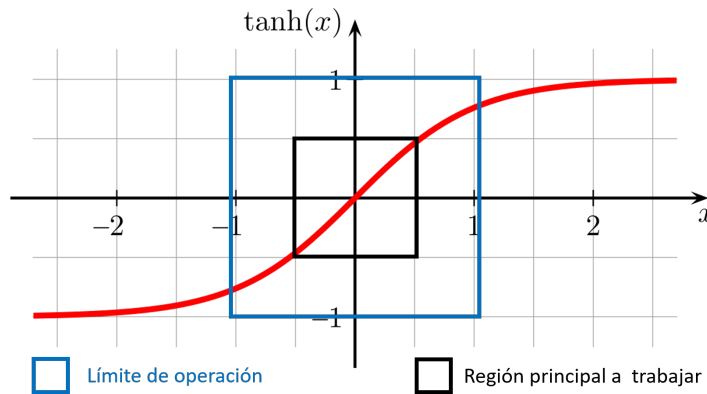


Figura 4.2. Región de operación de la tangente hiperbólica.

### 4.1.2. Identificación neuronal fuera de línea para variable de interés: $DQO_s$ a la salida

La identificación de sistemas puede ser aplicada de dos formas: *en línea o fuera de línea*. Cuando se usan redes neuronales para la identificación fuera de línea esta fase es conocida como *entrenamiento general* y en la identificación en línea se le conoce como *entrenamiento especializado*. Hay una relación en estas dos formas de identificación, donde la red es entrenada fuera de línea con un conjunto de datos y, una vez entrenada, esta es aplicada en la identificación en línea con el fin de aproximar la señal deseada.

Con los datos proporcionados la variable de mayor interés es la  $DQO_s$  de salida, puesto que proporciona información relevante sobre el consumo del sustrato y es la variable que se interesa controlar. Siguiendo el esquema de entrenamiento de la Figura 2.18 las condiciones de la identificación neuronal en esquema *serie-paralelo* son las siguientes:

- Pesos sinápticos iniciales: valores aleatorios en un rango de  $[-0.1, 0.1]$ .
- 2 neuronas en la capa 1 para  $J_u$  y  $J_{y_n}$ .
- Los retardos de  $u$  y  $y$  son:  $ret_u = 1$ ,  $ret_y = 3$ .
- La entrada está escalada 1 : 12 y la salida 1 : 24.
- La identificación es tipo SISO (entrada=  $CVA$  (Figura 4.3) y  $y = DQO_s$  a la salida).
- $\eta_0 = 0.1$

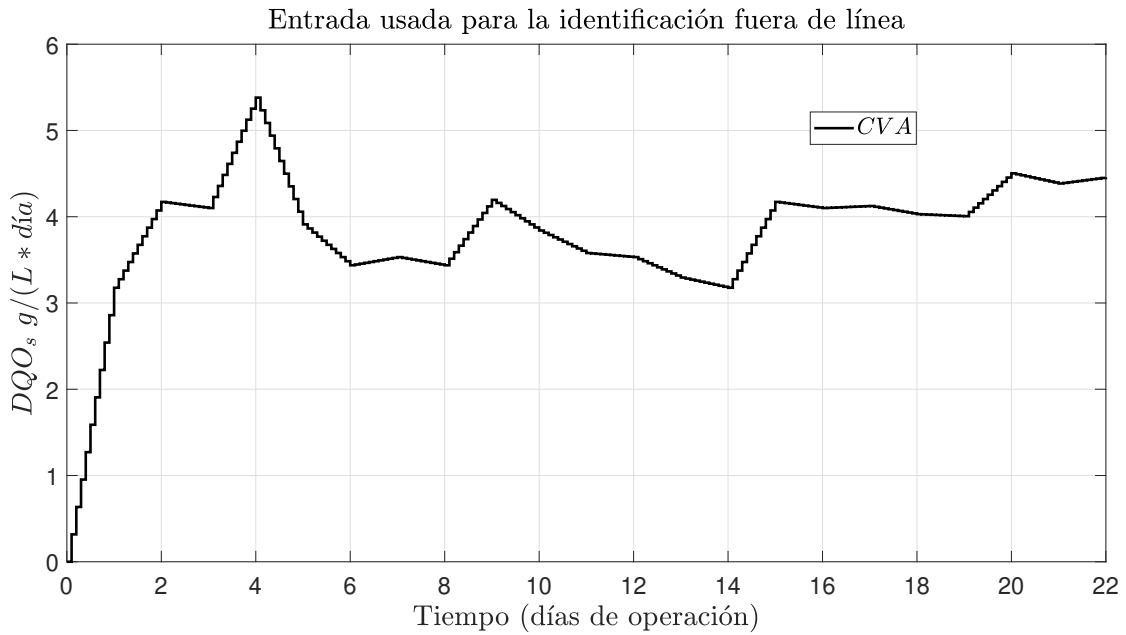


Figura 4.3. CVA utilizada para los esquemas de indentificación (etapa de arranque).

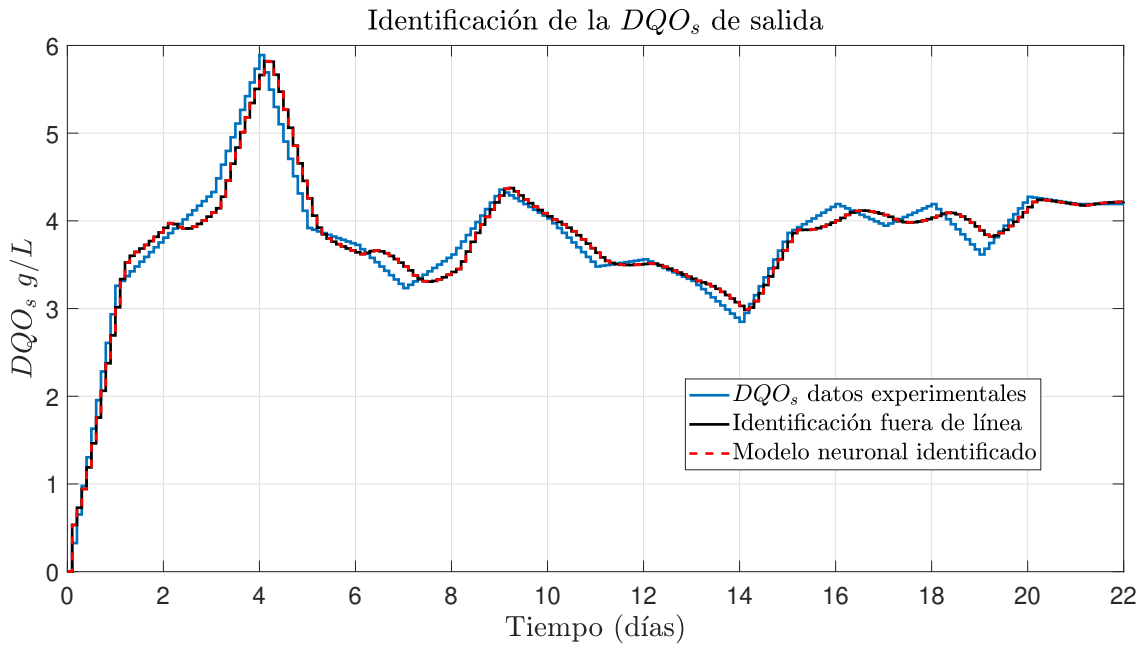


Figura 4.4. Identificación de la  $DQO_s$  a la salida.

Como se observa en la Figura 4.4, la indentificación fuera de línea muestra una aproximación adecuada con el esquema SISO donde usando la ecuación (2.21) presenta un  $FIT = 80.1135\%$ . Una vez teniendo la indentificación, los valores de los pesos sinápticos de la última época de entrenamiento se utilizan en el esquema de la Figura 3.9 para tener el modelo neuronal indentificado el cual presenta un  $FIT = 80.1131\%$ . Se observa que la *indentificación serie-paralelo* y el *modelo neuronal* son, prácticamente, iguales.

La Figura 4.5 muestra que pasadas las 1500 épocas de entrenamiento la variación del error se mantiene constante esto indica que la identificación es suficiente con 1600 épocas. Sin embargo, elegir un número menor de épocas puede ser no suficiente tiempo para que los pesos sinápticos converjan en un valor adecuado y un número mayor de épocas puede provocar un sobre entrenamiento de la red neuronal y, además, ninguno de los dos casos es conveniente. Es importante mencionar que aunque las 1250 épocas son adecuadas para la identificación, el *modelo neuronal* puede presentar un *FIT* bajo y puede ser no apto para aplicar una ley de control. Para este trabajo se seleccionó hasta las 4000 épocas de entrenamiento siendo el límite antes de empezar el sobre entrenamiento de la RNA.

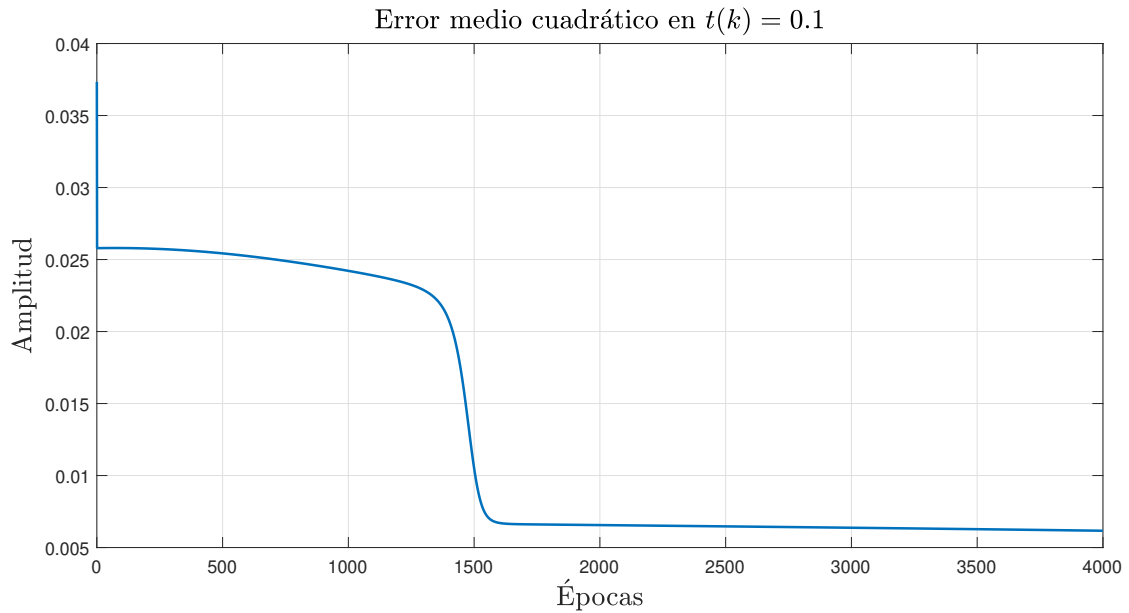


Figura 4.5. EMC de  $DQO_s$  a la salida.

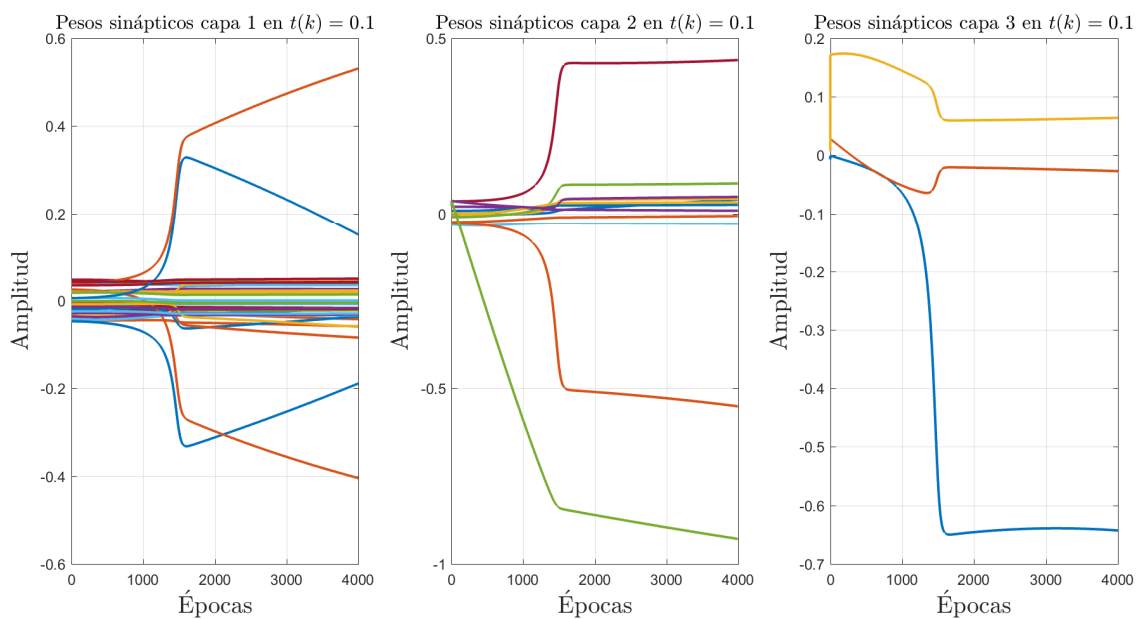


Figura 4.6. Adaptación de pesos sinápticos por épocas.

La Figura 4.6 muestra la adaptación de los pesos sinápticos a través de las épocas de entrenamiento neuronal. Se observa que aún pasando las 4000 épocas de entrenamiento los pesos sinápticos siguen modificándose y no se mantienen constantes a diferencia del EMC que logra mantenerse constante significando que todavía se puede seguir llevando el entrenamiento, pero puede provocar el sobre entrenamiento. La Figura 4.7 presenta los pesos seleccionados para el modelo neuronal de la última época de entrenamiento que son las 4000 épocas.

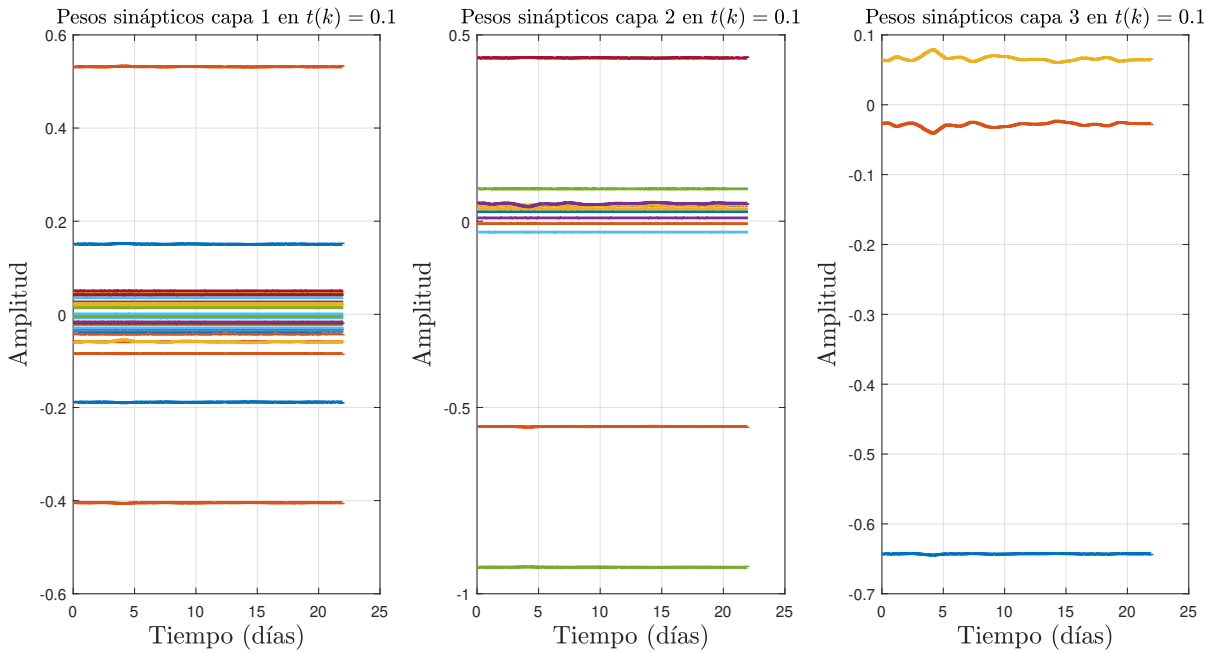


Figura 4.7. Pesos sinápticos para modelo neuronal.

En el anexo B se presentan otras identificaciones de variables de interés del sistema.

### 4.1.3. Validación de identificación neuronal con variable principal de interés: $DQO_s$ a la salida

El objetivo de este punto es de validar que la red neuronal sea capaz de ajustar nuevos datos, los datos proporcionados por el ITO contienen 30 días de operación del RAH, estos nuevos datos fueron proporcionados el mes de abril de 2019, en la Figura 4.8 se muestran los datos proporcionados.

En esta sección, la validación de la identificación no realiza un entrenamiento por épocas como en la identificación fuera de línea, se utilizan los últimos pesos sinápticos de la última época de entrenamiento como condiciones iniciales de los pesos sinápticos para identificar el sistema. Es decir, se sigue usando el mismo esquema de la Figura 2.18 pero sin entrenamiento por épocas y un  $\eta$  fijo lo cual la RNA debe de tener la capacidad de seguir la dinámica del sistema.

Los datos interpolados permiten tener diferentes escenarios de tiempo de adquisición de la toma de muestras del sistema como los que se presentan a continuación.



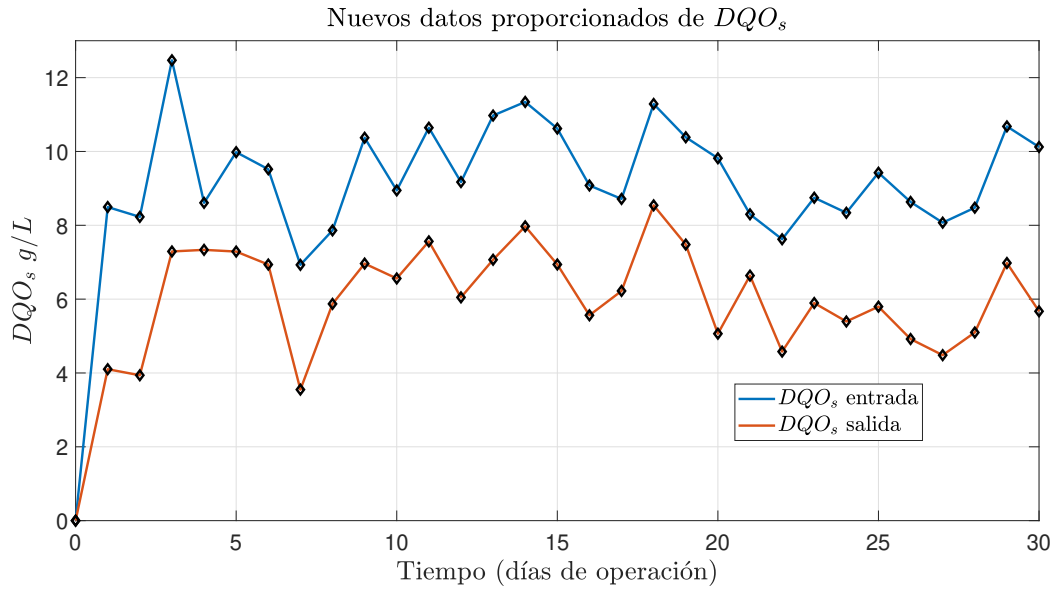


Figura 4.8. Nuevos datos proporcionados de la  $DQO_s$ .

**4.1.3.1. CASO 1. Identificación neuronal sin atraso en la toma de muestras**

En este escenario se plantea que las muestras son adquiridas cada dos horas y media con la interpolación realizada (escenario ideal). Se utiliza la estructura de la RNA planteada en la sección anterior, donde las nuevas condiciones de la RNA son las siguientes:

- Pesos sinápticos iniciales: son los últimos pesos sinápticos de la Figura 4.7.
- $\eta = 0.1$ .
- La entrada está escalada 1 : 12 y la salida 1 : 24.
- La identificación es tipo SISO (entrada: CVA (Figura 4.9) y salida:  $DQO_s$  a la salida).

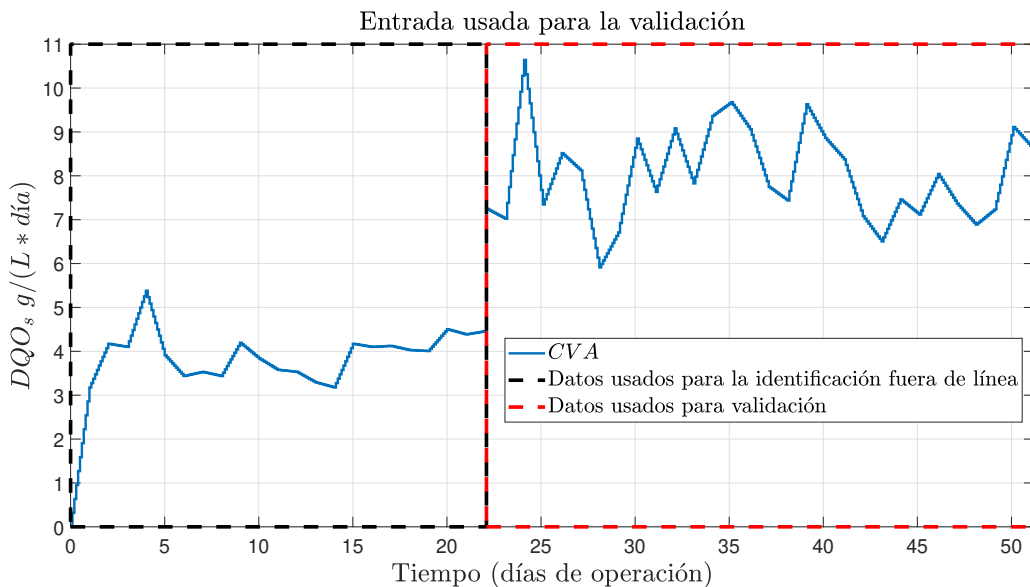


Figura 4.9. CVA para validación.

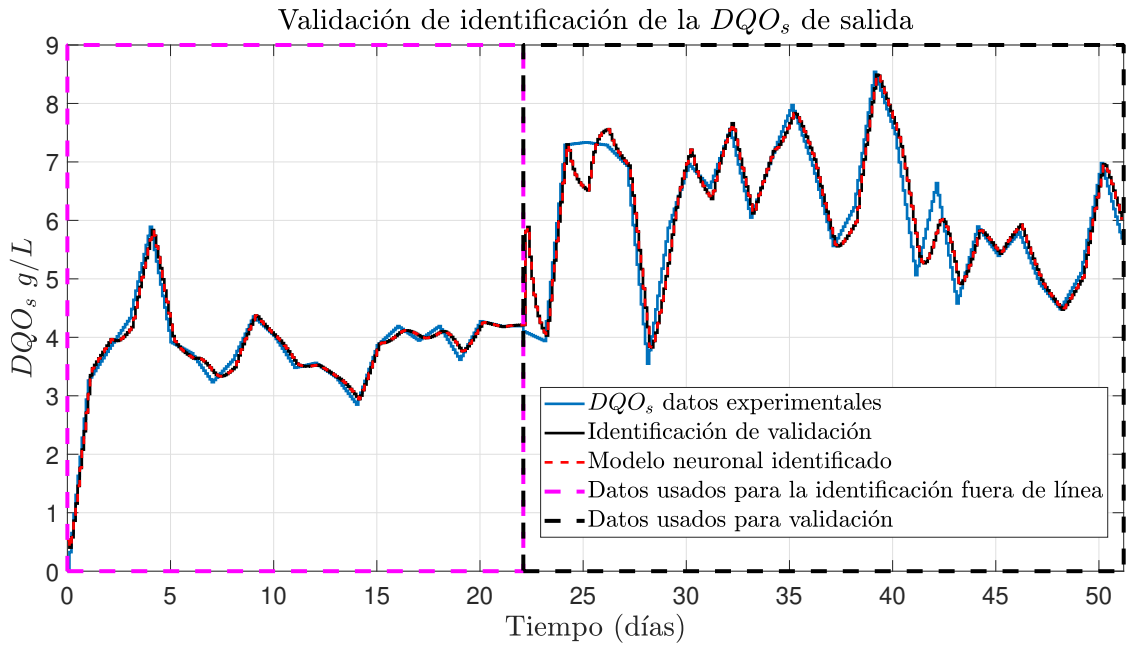


Figura 4.10. Validación de identificación.

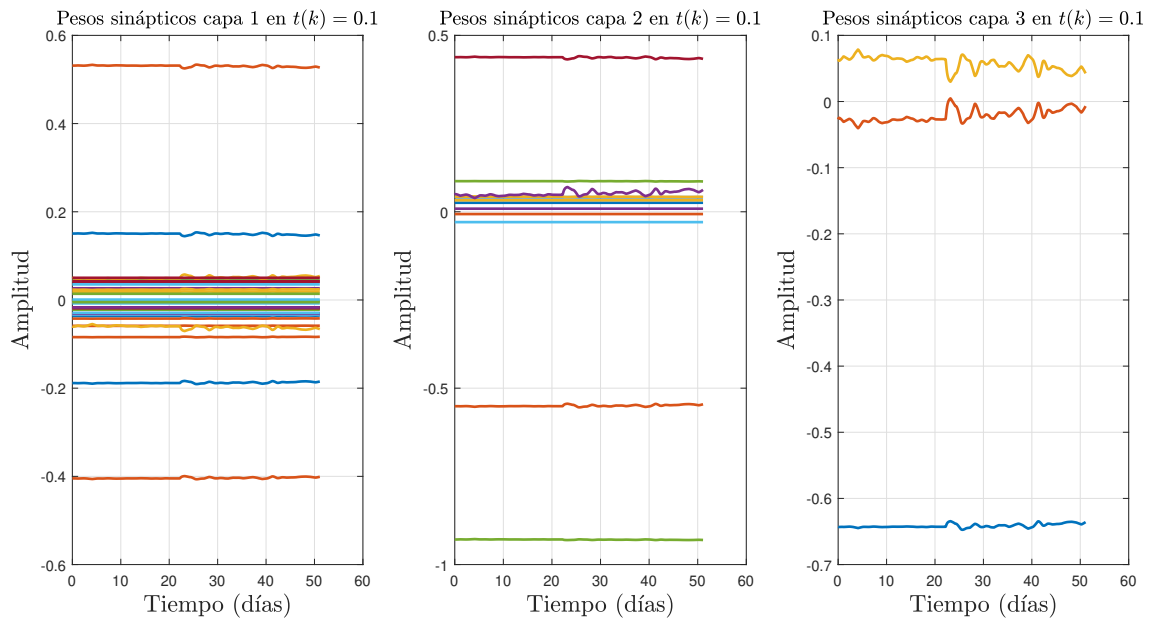


Figura 4.11. Pesos sinápticos de la identificación.

En la Figura 4.10 se visualiza la identificación para la validación donde la RNA puede mantener la dinámica del sistema siendo un esquema SISO después del entrenamiento fuera de línea teniendo un  $FIT = 81.4936\%$  y se puede ver que el modelo neuronal que se obtiene de la identificación presenta un  $FIT = 81.4932\%$ . En la Figura 4.11 se observa el comportamiento de los pesos sinápticos, la cual no muestran cambios demasiados bruscos que indica que se encuentran dentro de un rango adecuado.

Con las simulaciones presentadas en el escenario ideal se afirma que es posible realizar la indentificación con los pocos datos experimentales presentados. En las condiciones propuestas de la RNA se utiliza dos neuronas en la entrada y en la salida de la capa 1 y siguiendo el esquema de la Figura 2.19 se puede afirmar que la estructura seleccionada de RNA recurrente es adecuada para indentificar las variables de interés del RAH. Sin embargo, también se puede probar otro diseño de la RNA como se presenta a continuación.

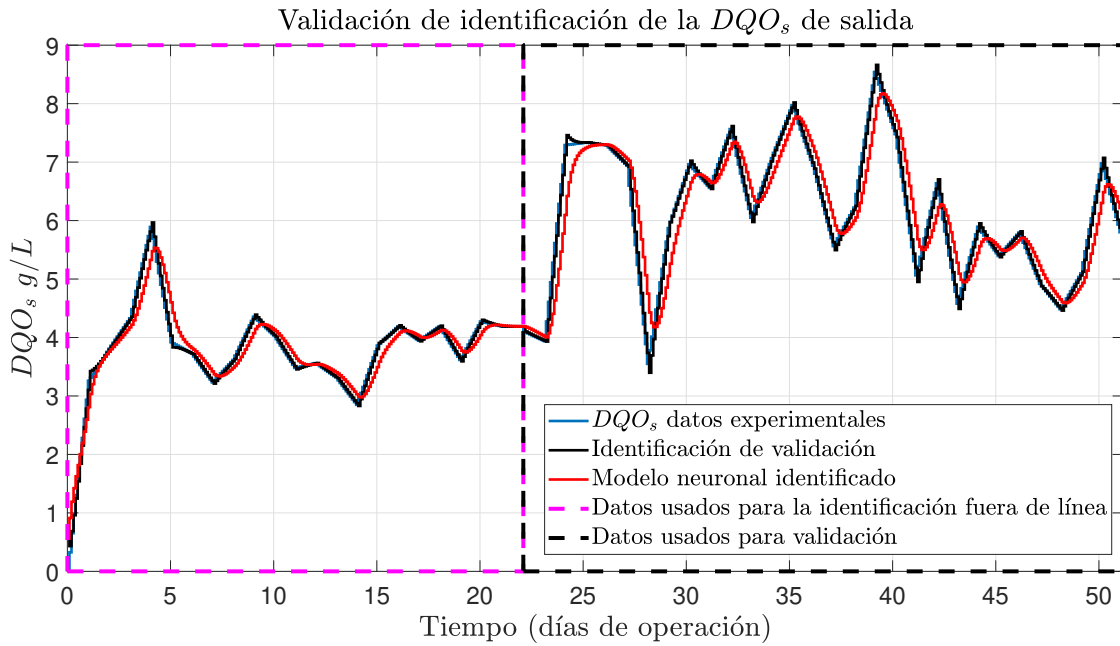


Figura 4.12. Validación de indentificación nuevo esquema.

En la Figura 4.12 se presenta otro esquema de RNA donde su estructura utiliza 5 neuronas para la entrada y la salida en la capa 1, la cual en su entrenamiento fuera de línea se usó 100 épocas de entrenamiento, donde presenta como resultado un  $FIT = 93.39\%$  en la indentificación y el modelo neuronal presenta un  $FIT = 73.09\%$ , lo cual también es un modelo aceptable. A esta arquitectura neuronal se le dio la prioridad a la indentificación y no al modelo neuronal obtenido a diferencia del caso anterior se le dio más prioridad al modelo neuronal que a la indentificación.

La decisión de que modelo se va a utilizar dependerá de que se busca en la indentificación sea una RNA para la indentificación en línea o un modelo neuronal para diseñar una ley de control. En el caso de esta tesis lo que se busca en la indentificación es obtener un modelo representativo del comportamiento de la  $DQO_s$  para poder diseñar una ley de control para el RAH.

#### 4.1.3.2. CASO 2. Indentificación neuronal con atraso en la toma de muestras

La interpolación realizada en los datos permite plantear la indentificación en diferentes tiempos de adquisición de las muestras permitiendo plantear un escenario muy cercano al real con las limitaciones que tiene el ITO para adquirir sus muestras en el laboratorio. Como condición la entrada y la salida de la RNA no se actualiza hasta que pase el tiempo que le toma obtener la muestra. Al igual que en la sección anterior se utilizan las mismas condiciones de la RNA.

El ITO tiene como limitante la cantidad de muestras que pueden tomar en el día, ellos pueden tomar como máximo dos muestras al día del proceso donde se requiere un tiempo de espera entre 2 a 4

horas para obtener los resultados del laboratorio. Con estas limitaciones, a continuación, se muestran diferentes tiempos de adquisición de las muestras.

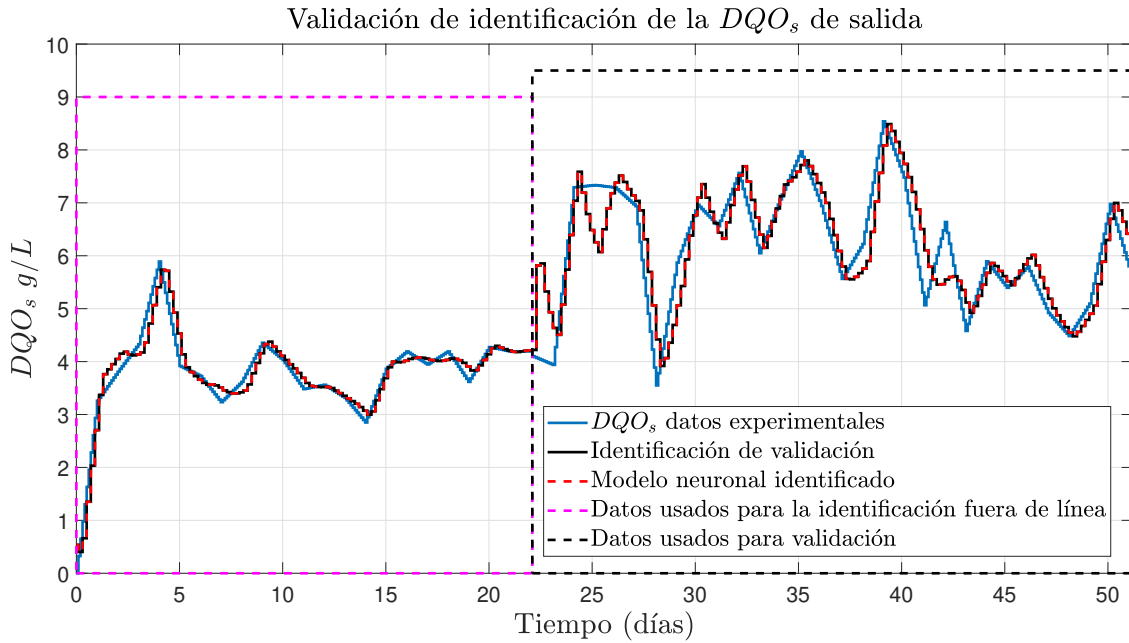


Figura 4.13. Validación de indentificación con 1 retardo (escenario ideal).

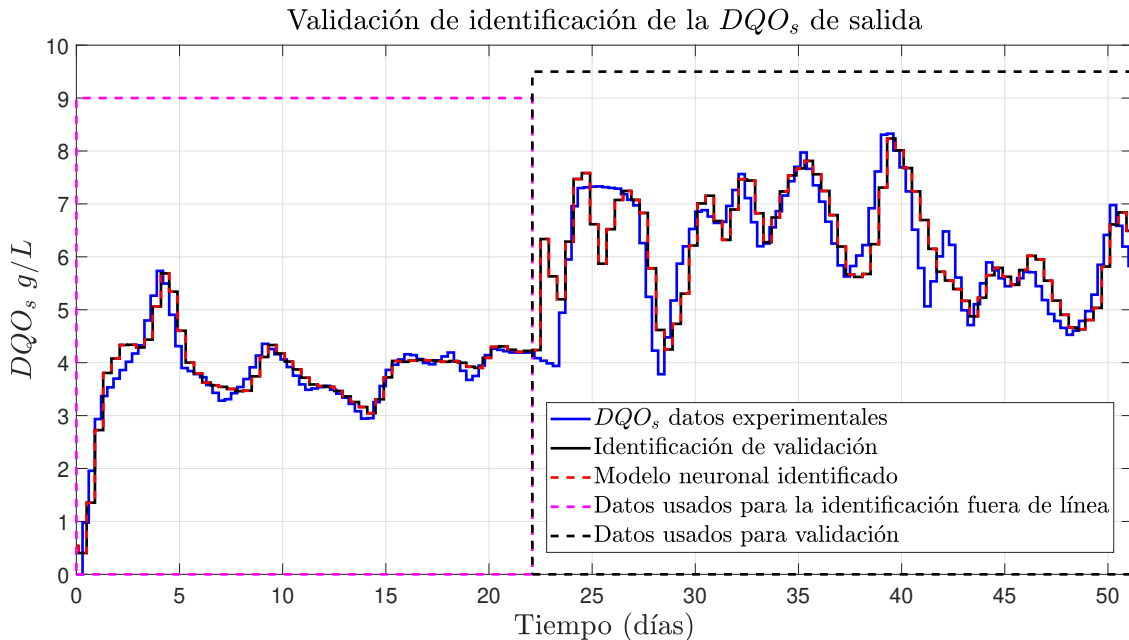


Figura 4.14. Validación de indentificación con 3 retardos (escenario ideal).

En las Figuras 4.13 y 4.14 se refleja que la indentificación con 1 retardo teniendo un  $FIT = 72.2643\%$  y el otro con 3 retardos teniendo un  $FIT = 63.5925\%$  son casos ideales de la indentificación. Para escenarios más reales las Figuras 4.15 y 4.16 reflejan la indentificación con 5 retardos teniendo un  $FIT = 58.2751\%$  y el otro con 10 retardos teniendo un  $FIT = 47.8896\%$ . Se discierne que las identi-

ficaciones en los *casos reales* presentan un bajo *FIT*, mientras que en los *casos ideales* presentan un *FIT* adecuado donde siguen manteniendo la dinámica del sistema.

Con las limitaciones que tiene el ITO para procesar estas muestras se considera que el mejor escenario de la indentificación sería la de tener dos muestras al día como en la Figura 4.15. Aunque la indentificación es baja se puede decir que la RNA puede mantener aún, con dificultad, la dinámica del RAH con los datos experimentales proporcionados.

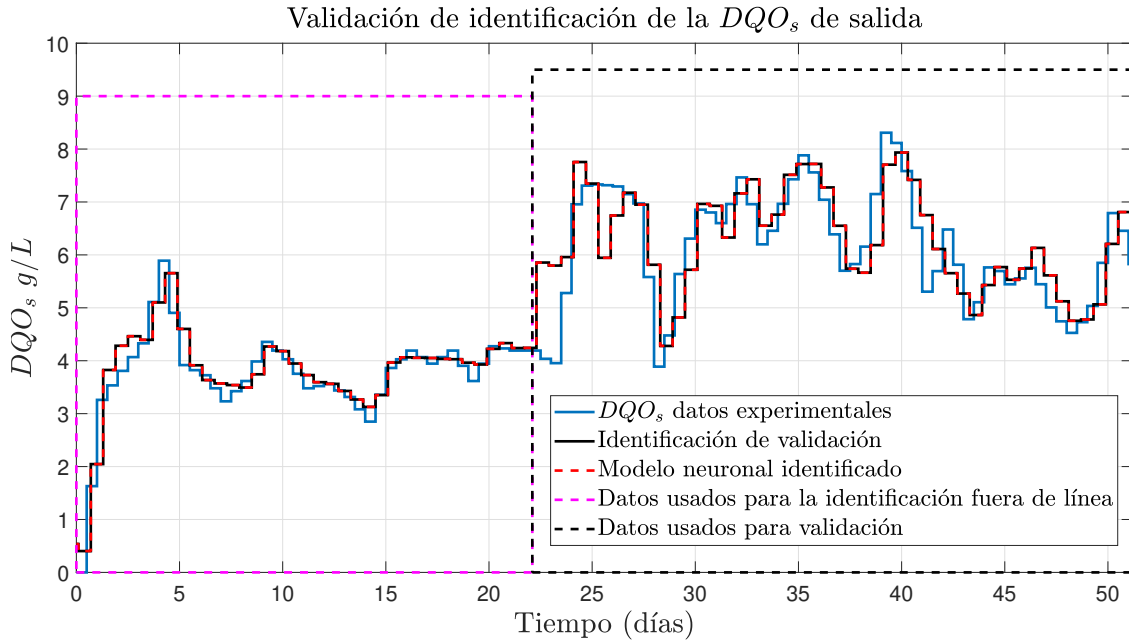


Figura 4.15. Validación de indentificación con 5 retardos (escenario real).

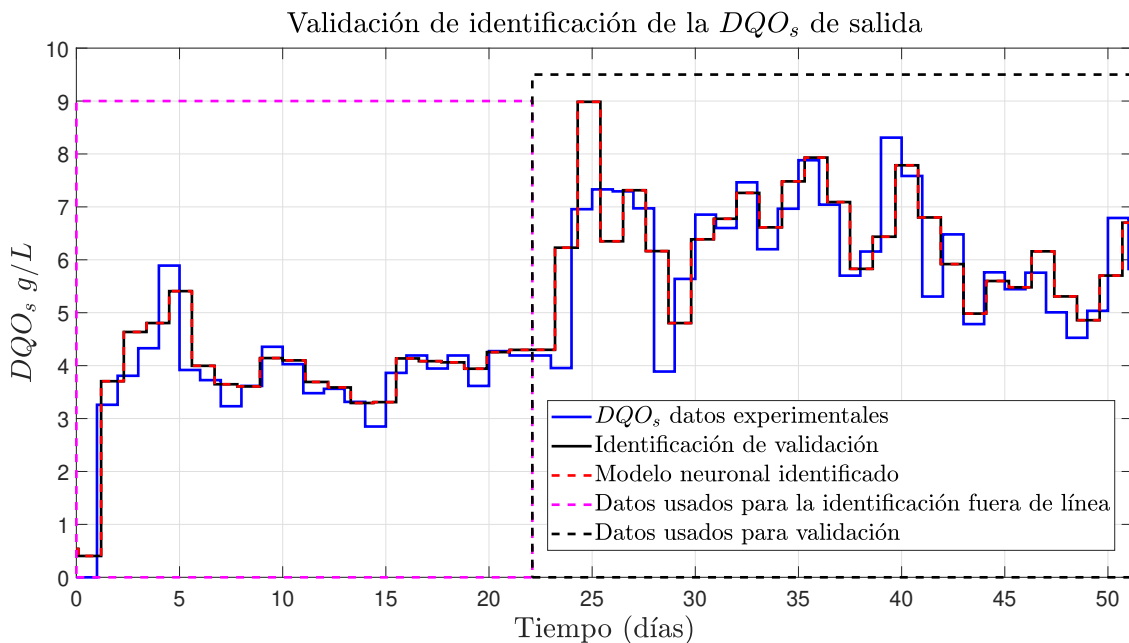


Figura 4.16. Validación de indentificación con 10 retardos (escenario real).

Tabla 4.1. Validación de modelos identificados.

Validación	Modelo sin retardos	Modelo con 1 retardo	Modelo con 3 retardos	Modelo con 5 retardos	Modelo con 10 retardos
Error medio	-0.00043	-0.00096	-0.00198	-0.00287	-0.00483
Desviación estandar	0.01242	0.01741	0.02280	0.02607	0.03240
Error mínimo	-0.07511	-0.07644	-0.09786	-0.09434	-0.10894
Error máximo	0.03509	0.05232	0.06690	0.08849	0.12133
FIT	81.4936 %	72.2643 %	63.5925 %	58.2751 %	47.8896 %

Tabla 4.2. Índices de desempeño de indentificación.

Índices de desempeño	Modelo sin retardos	Modelo con 1 retardo	Modelo con 3 retardos	Modelo con 5 retardos	Modelo con 10 retardos
IAE	4.4680	6.2403	8.0455	9.3142	11.90
ITAE	126.97	178.64	229.18	262.43	331.04
ISE	0.07906	0.15566	0.26822	0.35229	0.54949
ITSE	2.2909	4.5271	7.7858	10.141	14.929

En la Tabla 4.1 se presenta los datos de la validación de los modelos identificados. Se observa que las diferencias entre los modelos identificados, desde los casos ideales hasta los casos reales, presentan niveles de error pequeños y la diferencia entre cada una es pequeña lo cual indica lo que indica que el modelo neuronal genera salidas parecidas a las salidas que genera el sistema real. En la Tabla 4.2 se presenta los índices de desempeño de los modelos identificados.

#### 4.1.4. Modelo neuronal indentificado

Para la prueba y diseño de controladores es necesario el tener un modelo matemático que represente la dinámica del proceso antes de poder implementarlos en el proceso real. Cuando se usan las redes neuronales, se debe de utilizar el último peso sináptico de los entrenamientos realizados. Los resultados presentados en la sección anterior muestran que la indentificación es eficiente utilizando la RNA propuesta de la Figura 3.8.

Tabla 4.3. Modelos neuronales identificados.

Modelos neuronales	Retardos en $u_{CVA}$	Retardos en $y_n$	Cantidad de neuronas	Épocas de entrenamiento	$FIT$ de modelos
Modelo 1	1	1	2	6000	65.4726 %
Modelo 2	1	1	6	4000	65.1416 %
Modelo 3	1	1	12	3000	65.3498 %
Modelo 4	1	2	5	6000	65.1416 %
Modelo 5	1	3	5	4000	65.4347 %
Modelo 6	2	2	5	4000	65.9249 %
Modelo 7	2	7	5	2000	65.2816 %
Modelo 8	3	4	5	2000	65.8934 %

En la Tabla 4.3 se muestran diferentes tipos de modelos neuronales que utilizan diferentes retardos en  $u_{CVA}$  y en  $y_n$  utilizados en la indentificación, la cantidad de neuronas utilizadas en su estructura, la cantidad de épocas de entrenamiento que fueron necesarios y el  $FIT$  que presenta cada modelo.

De la Tabla 4.3 se aprecia que los modelos presentan un *FIT* en un rango de 65 % y 66 % cuando se utiliza un solo peso sináptico indicando que es el máximo ajuste que puede tener la RNA recurrente, aunque se presente una cantidad diferentes de retardos tanto en  $u_{CVA}$  como en  $y_n$  y diferentes cantidades de neuronas, además, las indentificaciones fuera de línea de los 8 modelos presentan un FIT en un rango de 79 % y 85 % como se muestra en la Figura 4.17.

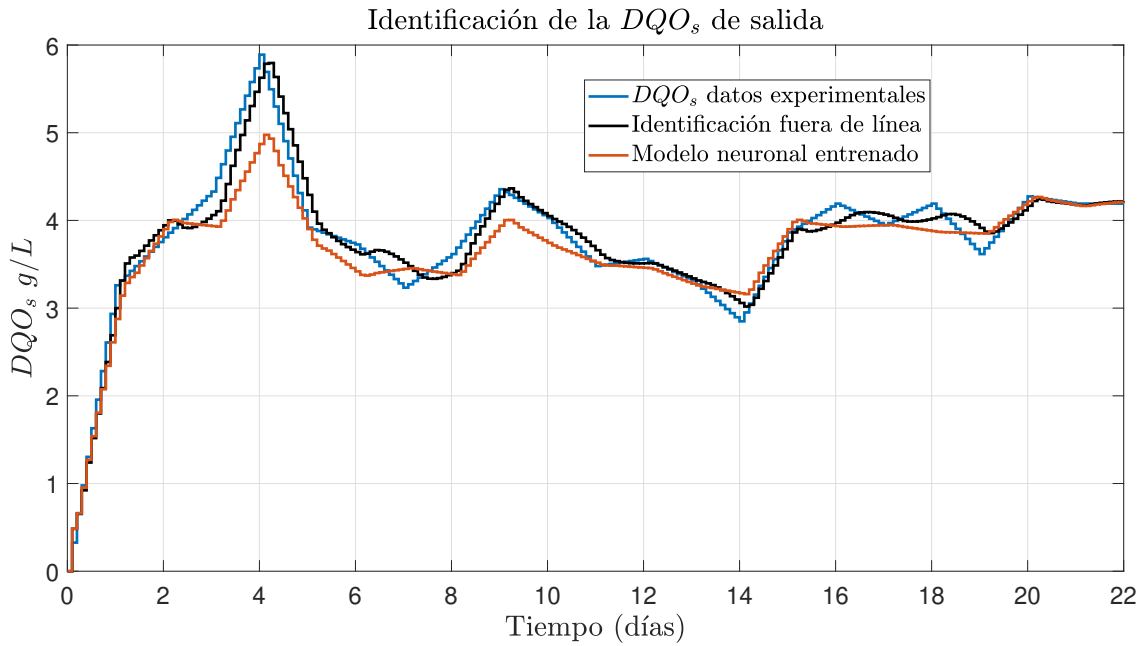


Figura 4.17. Modelo neuronal entrenado.

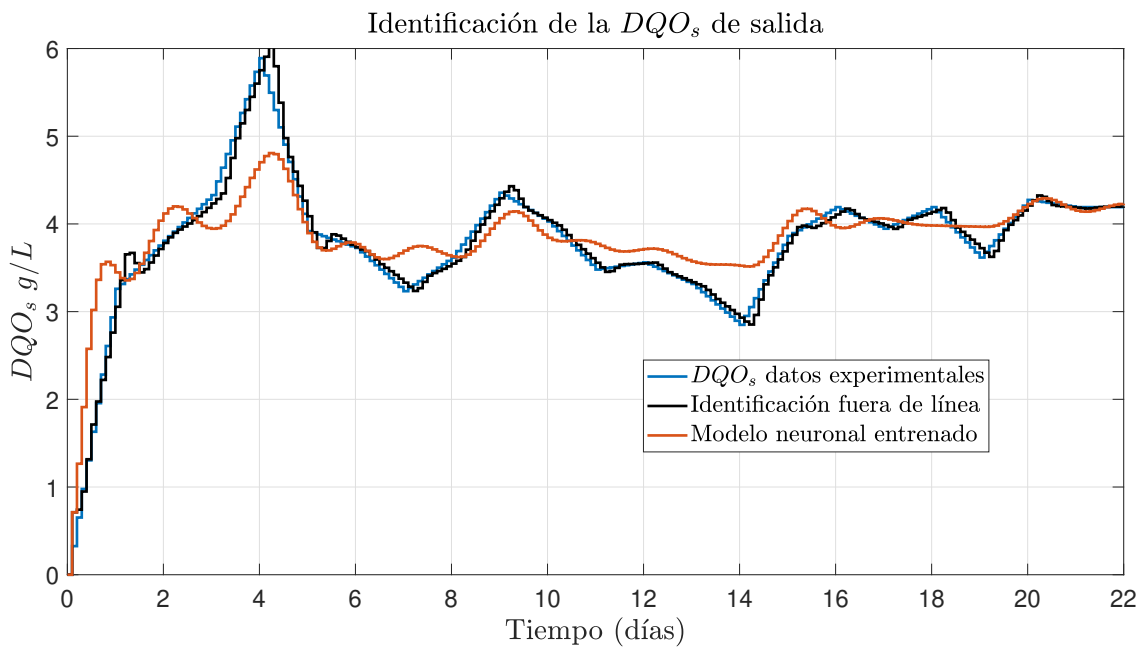


Figura 4.18. Modelo neuronal entrenado con 3000 épocas de entrenamiento (Modelo 8).

En la Tabla 4.3 también se aprecia la cantidad de épocas que son necesarias para identificar el modelo, donde a mayor aumento en los retardos y al número de neuronas se puede reducir el número de épocas de entrenamiento y mejorar la indentificación fuera de línea, sin embargo, eso aumenta el costo computacional y provoca un mayor riesgo al sobre entrenamiento. Un caso sería el modelo 7 y el modelo 8, si pasamos las 2000 épocas de entrenamiento, el modelo entrenado empieza a perder *FIT* hasta volverse un modelo que no puede utilizarse para las pruebas del control como se muestra en las Figuras 4.18 y 4.19.

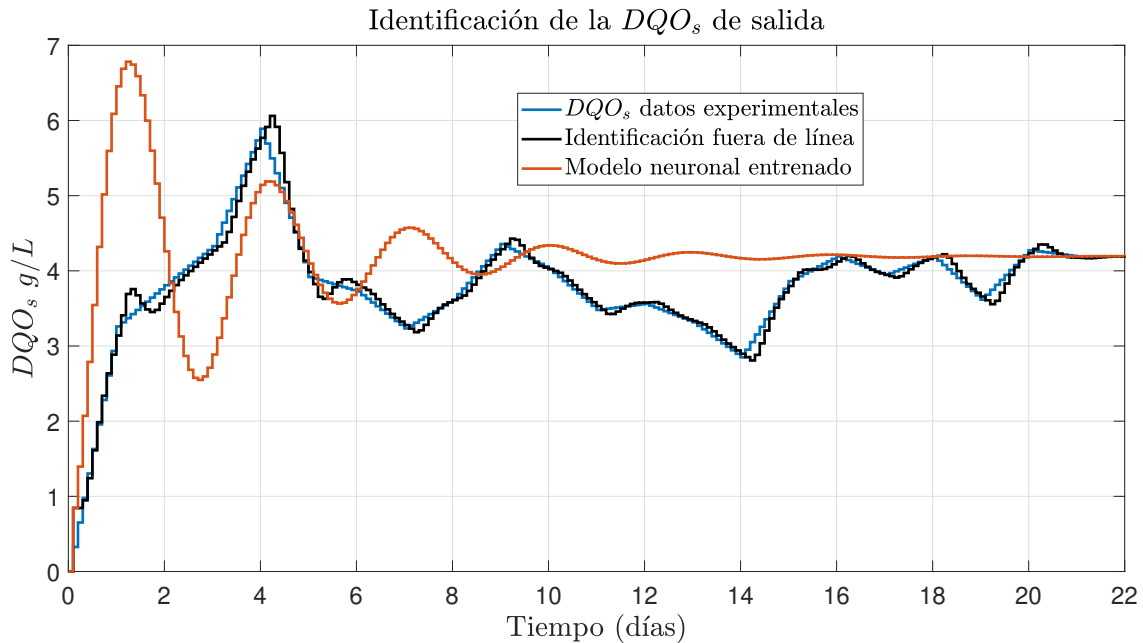


Figura 4.19. Modelo neuronal entrenado con 3000 épocas de entrenamiento (Modelo 7).

Con los resultados de la Tabla 4.3 se considera a utilizar como modelo neuronal para la prueba del controlador el *Modelo 1*, ya que presenta una menor cantidad de neuronas y de retardos utilizados, así reduciendo el costo computacional y la capacidad de mantener el *FIT*.

#### 4.2. Controlador neuronal para variable de interés: $DQO_s$ a la salida

Con los datos proporcionados del RAH que representan el arranque del proceso se tiene que considerar diferentes puntos:

- La dinámica del proceso es muy lenta.
- El controlador debe de ser capaz de llegar a la referencia deseada desde cualquier día que se decida controlar el reactor.
- Si se desea controlar desde el arranque debe de esperar al menos *dos días de operación* del proceso para poder tener la  $DQO_s$  a la salida.

Estos 3 puntos son esenciales porque representan las condiciones reales que tiene el RAH y el controlador debe de tener la capacidad de poder trabajar bajo estas condiciones. A continuación, se presenta el uso del controlador neuronal diseñado en la sección 3.4 con los casos usados en la indentificación.



### 4.2.1. CASO 1. Controlador neuronal sin atraso en la toma de muestras

En este escenario se plantea una situación ideal donde se obtienen todas las muestras al igual que en la identificación; usando el modelo neuronal identificado en esquema SISO se desea llegar a la siguiente referencia deseada  $0.45 \text{ gDQO}_s/L$  en cualquier día de operación además se busca que la respuesta del sistema con el controlador sea *lenta* debido a que el proceso es lento, lo cual eso debe reflejarse con el controlador para tener un escenario muy cercano al proceso real. Las condiciones de la RNA del controlador son las siguientes:

- $T_0 = 0.1$  que representa 2 horas y media de los datos interpolados.
- $\eta = 0.1$
- Se usan 5 neuronas en la capa 1 para  $J_{u_{ne}}, r(k+1)$  y  $J_{y_n}$
- Usa 1 retardo,  $ret_c = 1$ .
- $y_n = DQO_s$  de salida (variable de interés).
- Pesos sinápticos iniciales son:
  - $Z_{ca} = 0.1406$ ,      ▪  $Z_{cb} = -0.1521$ ,      ▪  $Z_c = 0.2462$ ,      ▪  $Z_o = 0.8023$ .
- $V_{CA} = [-0.0477, -0.0278, -0.0289, -0.0017, -0.2401, 0.0770]$ ,
- $V_{CB} = [0.3399, 0.2216, -0.0301, -0.2435, 0.1906, 0.7996]$ ,
- $V_C = [0.0788, -0.0563, 0.3247, -0.0113, 0.3254, 0.0606]$ ,
- $W_{ca1} = [0.0609, 0.3491]$       ▪  $W_{ca3} = [-0.1430, 0.0372]$ ,      ▪  $W_{ca5} = [0.1151, -0.0447]$ ,
- $W_{ca2} = [-0.4488, -0.4472]$ ,      ▪  $W_{ca4} = [-0.1436, -0.3257]$ ,
- $W_{cb1} = [0.3951, 0.2839]$ ,      ▪  $W_{cb3} = [-0.1457, 0.3235]$ ,      ▪  $W_{cb5} = [-0.3749, -0.1982]$ ,
- $W_{cb2} = [-0.4295, -0.0683]$ ,      ▪  $W_{cb4} = [-0.1041, 0.1761]$ ,
- $W_{c1} = [0.3087, 0.3589]$ ,      ▪  $W_{c3} = [0.3836, -0.1814]$ ,      ▪  $W_{c5} = [-0.0237, 0.4047]$ ,
- $W_{c2} = [-0.3713, -0.2153]$ ,      ▪  $W_{c4} = [0.0070, 0.3210]$ ,

En la Figura 4.20 se visualiza el comportamiento del modelo neuronal entrenado utilizando el controlador tipo neuronal, se observa que el modelo no presenta cambios en los primeros dos días puesto que no se ha activado el controlador. Cuando el controlador se activa, pasando dos días de operación, el modelo presenta un ajuste suave con el fin de llegar a la referencia deseada, sin embargo, se aprecia que modelo no logra llegar a la referencia.

En la Figura 4.21 se muestra el comportamiento que tiene la señal generada por el controlador neuronal, se observa que la señal del controlador y el nivel de error llevan un cambio constante desde el momento que se activa el controlador, cuando pasan 5 días de operación, se observa que el modelo neuronal está dentro del valor de  $1 \text{ gDQO}_s/L$  y se ve como la señal del controlador neuronal empieza a tener un cambio mucho más lento reflejándose en el nivel de error como se va acercándose a cero de manera muy lenta.

Esta lentitud de ajuste se ve de manera más clara en los pesos sinápticos del controlador neuronal de la Figura 4.22, en donde llevan un ajuste constante cuando se activa el controlador. Una vez pasando 5 días de operación, se observa que la mayoría de los pesos sinápticos presentan un ajuste mucho más lento cuando el modelo neuronal se va acercando a la referencia deseada. Este ajuste se ve reflejado en el flujo del RAH y en la CVA de la Figura 4.23.

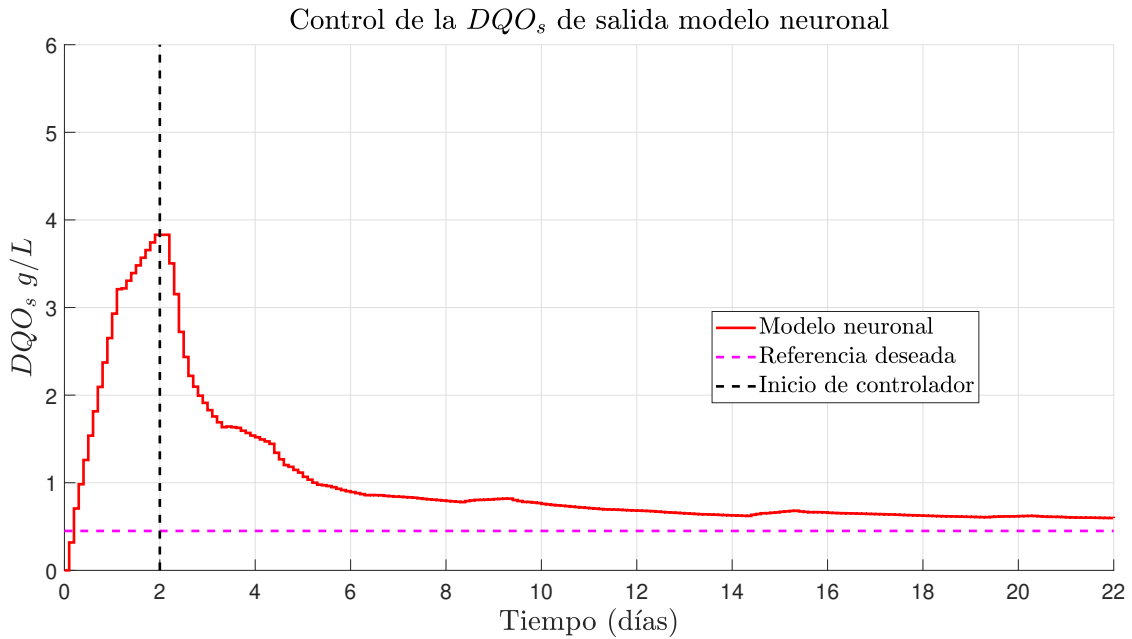


Figura 4.20. Modelo neuronal con control (escenario ideal) con referencia  $0.45 \text{ gDQO}_s/L$ .

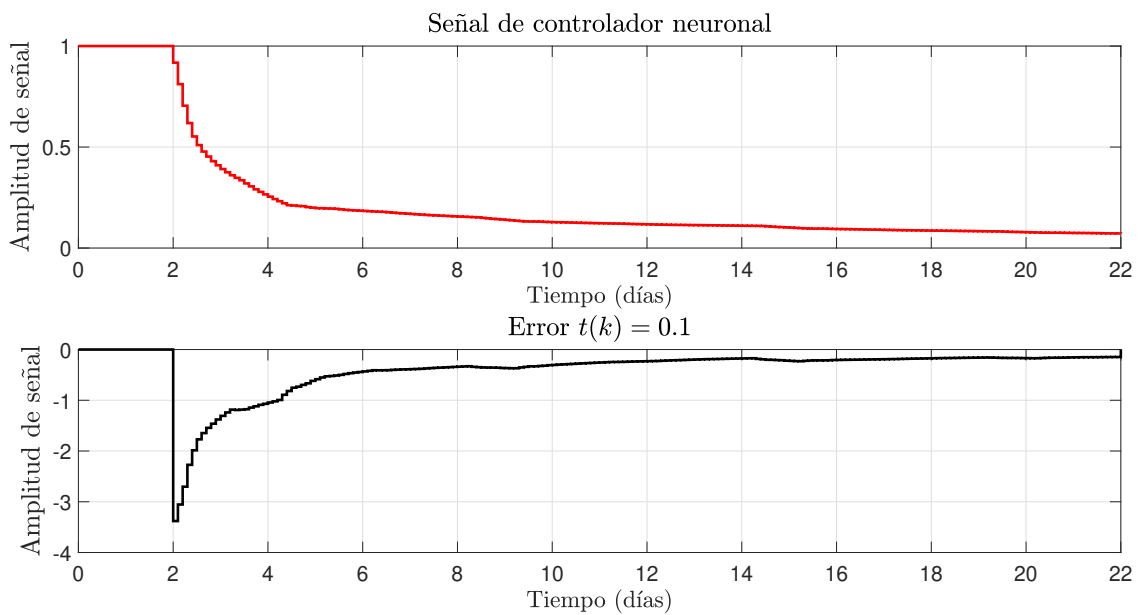


Figura 4.21. Comportamiento del controlador neuronal y error generado con referencia  $0.45 \text{ gDQO}_s/L$  (escenario ideal).

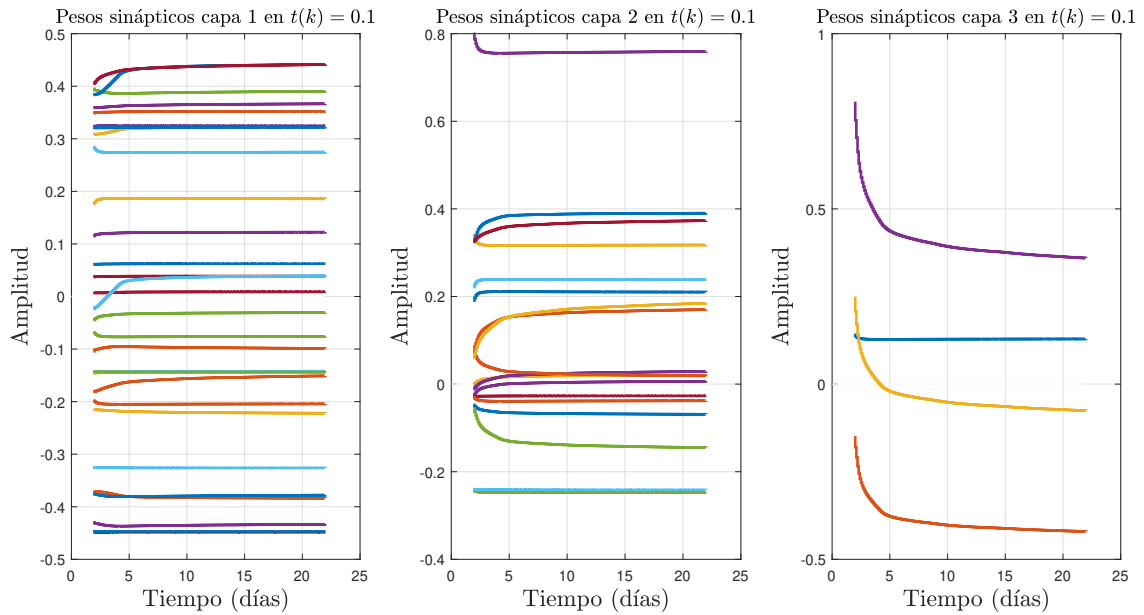


Figura 4.22. Comportamiento de pesos sinápticos del controlador con referencia  $0.45 \text{ gDQO}_s/L$  (escenario ideal).

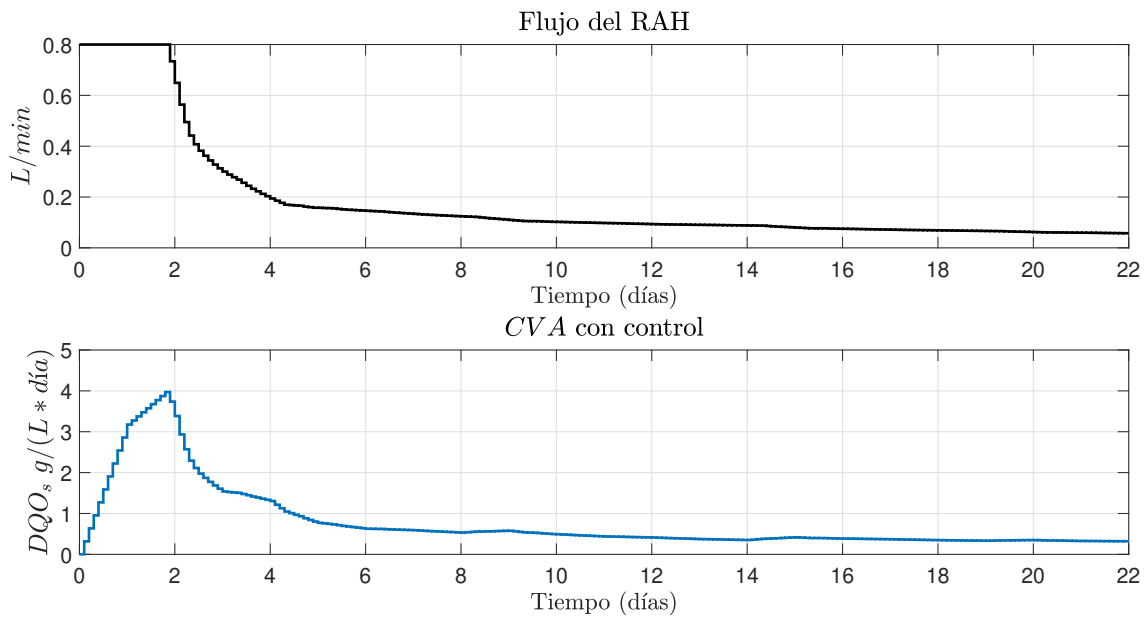


Figura 4.23. Comportamiento del flujo y CVA del RAH con control con referencia  $0.45 \text{ gDQO}_s/L$  (escenario ideal).

En la Figura 4.24 se plantea otro tiempo de activación del controlador, al pasar 8 días de operación se observa como el controlador va ajustando el modelo neuronal presentando un ajuste suave hacia la referencia deseada. Al igual que en la Figura 4.20, el modelo no logra llegar a la referencia deseada presentando el mismo comportamiento cuando se activa el controlador pasando dos días de operación.

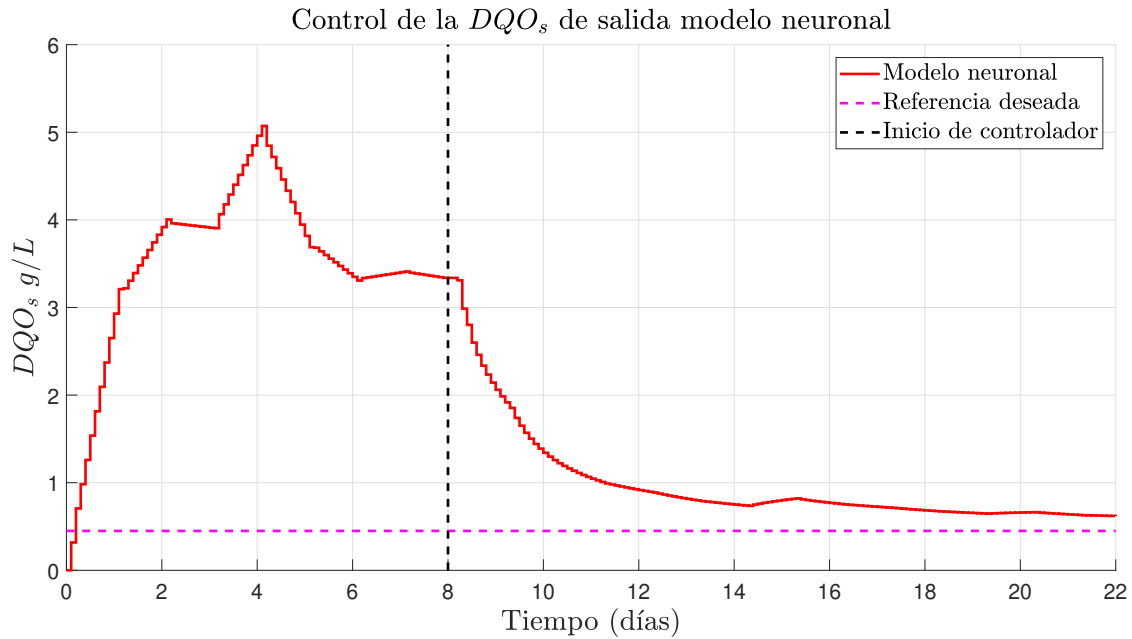


Figura 4.24. Modelo neuronal con control (escenario ideal) con referencia  $0.45 \text{ gDQO}_s/\text{L}$  con diferente día de activación.

#### 4.2.2. CASO 2. Controlador neuronal con atraso en la toma de muestras

El objetivo de esta sección es plantear un escenario muy cercano al real con las limitaciones que tiene el ITO para adquirir sus muestras en el laboratorio. Con lo realizado en la sección 4.1.3.2, se usará el escenario donde se adquieren dos muestras al día donde el modelo neuronal identificado a utilizarse no presentará retardos en la salida, puesto que la identificación está basada en la interpolación de los datos, sin embargo, la entrada considerada (*CVA*) si presentará estos retrasos.

Al igual que la sección anterior, se desea llegar a la siguiente referencia deseada  $0.45 \text{ gDQO}_s/\text{L}$  en cualquier día de operación, además, se busca que la respuesta del sistema con el controlador sea *lenta* debido a que el proceso es lento, lo cual eso debe reflejarse con el controlador para tener un escenario muy cercano al proceso real. Las condiciones de la RNA del controlador son las siguientes:

- $T_0 = 0.5$  que representa 12 horas de los datos interpolados (2 muestras al día).
- $\eta = 0.1$
- Se usan 5 neuronas en la capa 1 para  $J_{u_{ne}}, r(k+1)$  y  $J_{y_n}$
- Usa 1 retardo,  $ret_c = 1$ .
- Pesos sinápticos iniciales son los mismos del CASO 1.

En la Figura 4.25 se visualiza el comportamiento del modelo neuronal entrenado utilizando el controlador tipo neuronal, se observa como el modelo presenta cambios bruscos debido al tiempo que se toma para adquirir las muestras en comparación con el escenario ideal. Cuando el controlador se activa pasando dos días de operación, el modelo neuronal trata de mantener un ajuste suave para llegar a la referencia deseada, sin embargo, se aprecia que modelo no logra llegar a la referencia quedándose en un valor cercano a  $1 \text{ gDQO}_s/\text{L}$ .

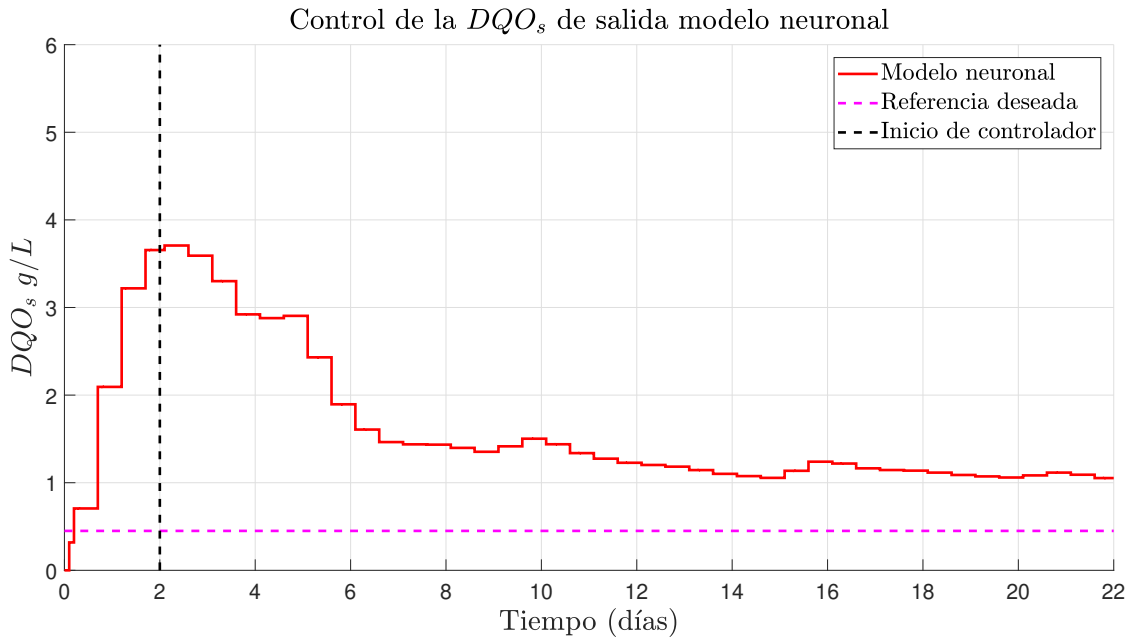


Figura 4.25. Modelo neuronal con control (escenario real) con referencia 0.45  $gDQO_s/L$ .

En la Figura 4.26 se muestra el comportamiento del error y el de la señal del controlador neuronal. Se observa que el ajuste del controlador neuronal mantiene un ajuste continuo desde el día 2 al día 6 de operación y el error presenta cambios algo bruscos, sin embargo, cuando pasan 6 días de operación el ajuste del controlador se vuelve más lento y se observa como la señal del error presenta cambios más suaves y lentos.

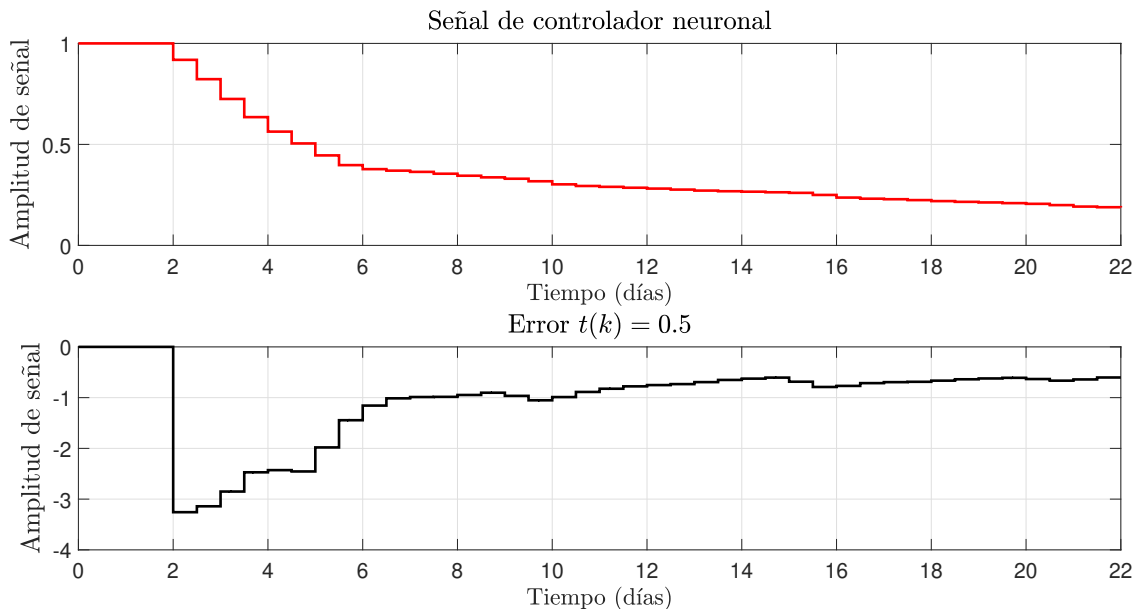


Figura 4.26. Comportamiento del controlador neuronal y error generado con referencia 0.45  $gDQO_s/L$  (escenario real).

Esta lentitud de ajuste se ve de manera más clara en los pesos sinápticos del controlador neuronal

de la Figura 4.27, en donde llevan un ajuste constante cuando se activa el controlador. Cuando se presenta la décima actualización del controlador, se observa que la mayoría de los pesos sinápticos presentan un ajuste mucho más lento cuando el modelo neuronal trata de acercarse a la referencia deseada. Este ajuste se ve reflejado en el flujo del RAH y en la CVA de la Figura 4.28.

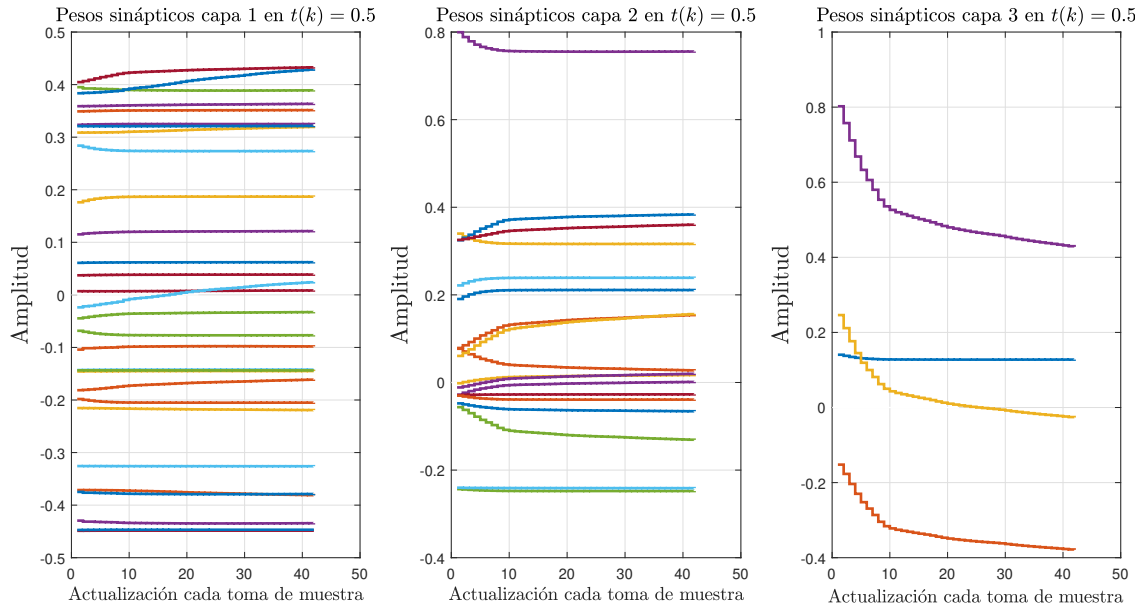


Figura 4.27. Comportamiento de pesos sinápticos del controlador con referencia  $0.45 \text{ gDQO}_s/L$  (escenario real).

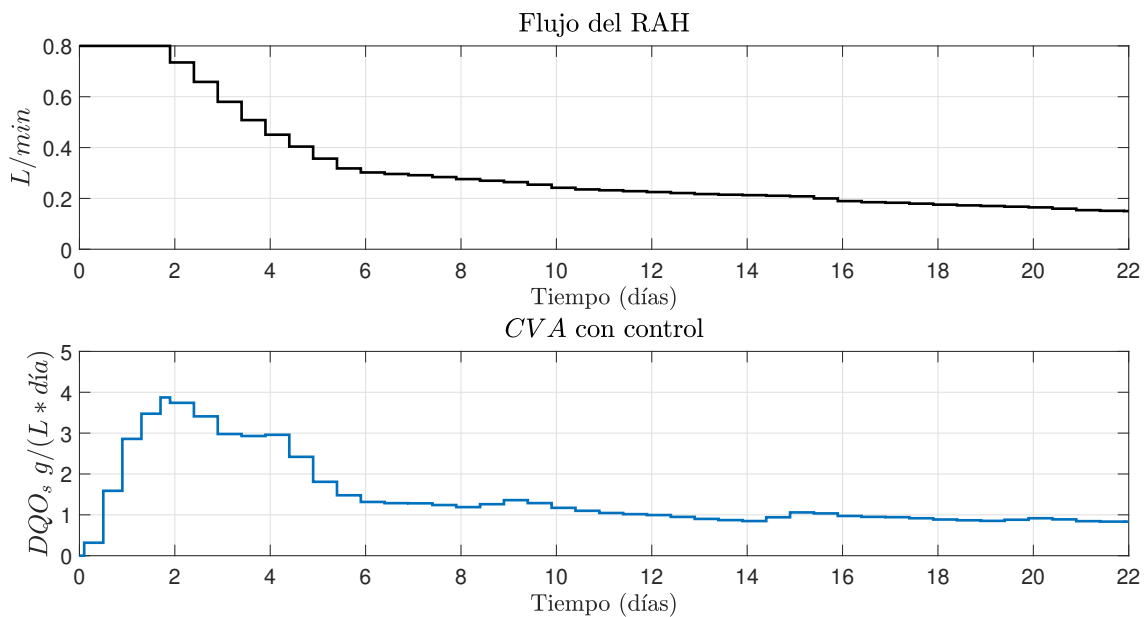


Figura 4.28. Comportamiento del flujo y CVA del RAH con control con referencia  $0.45 \text{ gDQO}_s/L$  (escenario real).

La toma de muestras con atrasos del escenario real afecta de manera significativa al modelo neuronal entrenado y al controlador neuronal donde el nivel de error es mucho mayor a comparación del modelo ideal viéndose como el modelo neuronal, en el escenario real, no logra tener un valor menor de  $1 \text{ gDQO}_s/L$ .

En la Figura 4.29 se plantea otro tiempo de activación del controlador, al pasar 5 días de operación, se observa como el controlador va ajustando el modelo neuronal presentando un ajuste suave hacia la referencia deseada. Al igual que en la Figura 4.25, el modelo no logra llegar a la referencia deseada presentando el mismo comportamiento cuando se activa el controlador pasando dos días de operación.

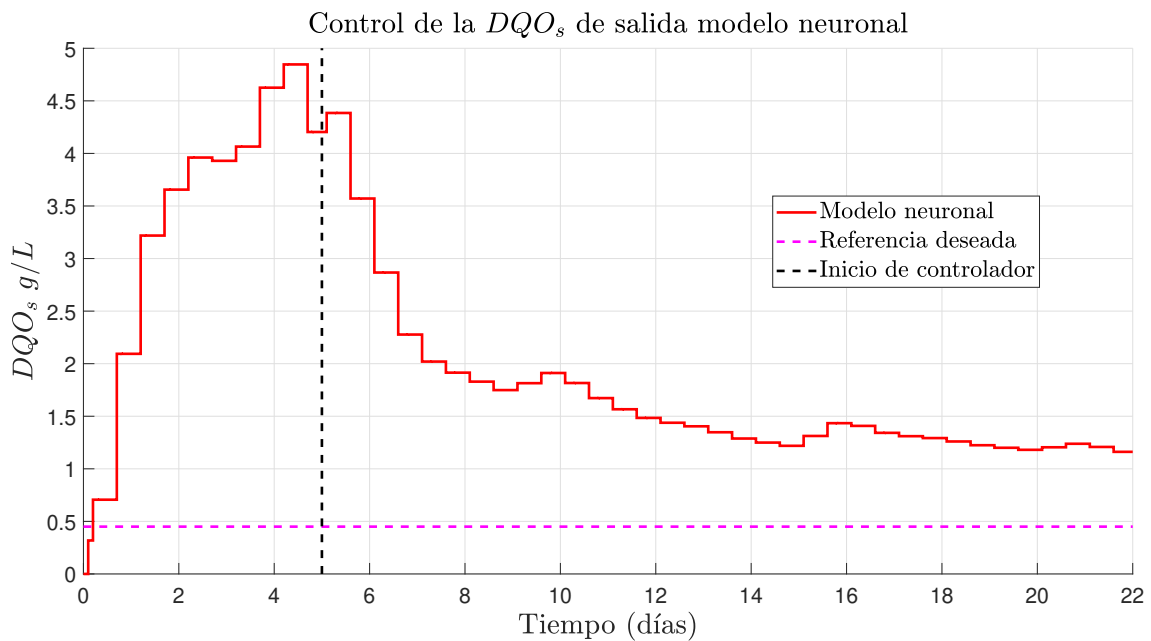


Figura 4.29. Modelo neuronal con control (escenario real) con referencia  $0.45 \text{ gDQO}_s/L$  con diferente día de activación.

# Capítulo 5

---

## Conclusiones y perspectivas

---

En este capítulo final se discuten algunas conclusiones derivadas del análisis de los resultados de las propuestas mencionadas en este trabajo y las simulaciones ejecutadas. También, se proponen algunos trabajos futuros que aportarían riqueza teórica y práctica al trabajo desarrollado aquí.

### 5.1. Conclusiones

El objetivo de esta tesis es desarrollar una estrategia de control que permita mantener un nivel de concentrado de  $DQO_s$  a través de un modelo identificado debido a la falta de un modelo matemático que represente la dinámica del RAH del ITO.

En el presente documento se estudió el desarrollo de la identificación de sistemas utilizando redes neuronales para poder identificar la dinámica del RAH ante la variable de interés que es la  $DQO_s$ , además se identificaron las siguientes variables: *el porcentaje de metano producido, el biogás y el porcentaje de remoción de  $DQO_s$ .*

La cantidad de datos experimentales que se presentaron para la identificación no son suficientes, provocando que no sea viable la identificación por cualquier método sea lineal o no lineal. Se realizaron pruebas de identificación utilizando los 22 datos proporcionados sin utilizar la interpolación y se obtuvo un  $FIT = 25\%$ . Durante estas pruebas de simulación se utilizó una gran cantidad de épocas de entrenamiento y diferentes retardos en la entrada y salida, sin embargo, el  $FIT$  no presentaba mejora alguna, de modo que se utilizó la interpolación en los datos.

Con las interpolaciones realizadas se pueden plantear diferentes tiempos en la toma de las muestras, permitiendo proponer la identificación y el control en escenarios reales de acuerdo a la limitada cantidad de muestras que pueden ser adquiridas en el ITO. Como resultado en la validación se opta por la obtención de dos muestras al día ya que mejora la identificación del proceso del RAH.

En la identificación de sistemas utilizando RNA se debe tener cuidado con su diseño en la estructura neuronal, en el caso de esta tesis, al usar una RNA recurrente de tres capas se debe cuidar *el número de neuronas en la capa 1 y los retardos internos* porque hay una relación entre costo computacional y exactitud. Con las simulaciones realizadas el tener demasiadas neuronas permite tener una mejor identificación, sin embargo, esto puede provocar dos cosas:

- propiciar más dificultad en el ajuste al igual que sus retardos y
- provocar que el modelo neuronal identificado sea demasiado robusto y no permita cambios cuando se aplica una ley de control.



En la identificación neuronal se presentaron dos escenarios: *el ideal y uno real*. En estos escenarios se observó que las identificaciones realizadas son capaces de reproducir las salidas del sistema en diferentes tiempos de adquisición de las muestras, esto se visualiza en las Tablas 4.1 y 4.2. Se observa en las tablas que, aunque los errores son pequeños, el *FIT* de los modelos identificados van cambiando drásticamente desde el 80 % hasta el 47 %.

Los índices de desempeño muestran que, en la identificación sin retardos, presenta valores pequeños lo cual indica que tiene una aproximación cercana al proceso a identificar. Se observa que al comparar los niveles de error de los modelos identificados presentan diferencias muy pequeñas indicando que los modelos deberían parecerse al modelo sin retardos, sin embargo, los índices de desempeño muestran que cada vez que hay un retardo la diferencia entre estos índices se vuelve más grande permitiendo ver ese cambio drástico de *FIT*.

Los cambios drásticos de *FIT* que se ven en los modelos identificados se debe a que las identificaciones realizadas están basadas en la identificación realizada en la fase fuera de línea (*fase de entrenamiento*) con los datos interpolados donde se usaron los 220 datos. La razón de no identificar con diferentes interpolaciones se debe a la limitante de datos proporcionados, como se mencionó anteriormente, con pocos datos no es posible realizar la identificación del RAH.

En esta tesis también se estudió el desarrollo de un controlador con un esquema neuronal con el fin de controlar el concentrado de  $DQO_s$  del RAH a una referencia deseada. Una de las consideraciones del por qué utilizar un control inteligente se debe a que la entrada considera para la identificación neuronal es la *CVA*, esta variable presenta cambios estocásticos al tener relación con el concentrado de la  $DQO_s$  de entrada, y por el factor limitante de sensores debido a que no se tienen datos reportados sobre la operación de flujo del RAH.

Para las pruebas del controlador se utilizó un modelo neuronal entrenado que representa de la  $DQO_s$ , en el caso de las identificaciones del biogás, porcentaje de metano producido y el porcentaje de remoción de  $DQO_s$ , las identificaciones fuera de línea presentaron un *FIT* favorable, sin embargo, como modelos entrenados presentan una deficiencia muy alta debido a la dinámica de las variables en la cual no son capaces de poder reproducir dicho comportamiento como se muestra en las Figuras B.5, B.10 y B.15 del Anexo B.

En las simulaciones del controlador propuesto se observó que el controlador en el escenario ideal presenta una respuesta suave para llegar a la referencia deseada aun cuando la entrada sigue presentando cambios, lo cual el controlador es capaz de cumplir el objetivo de control con el fin de tener una respuesta suave y lenta para plantear un posible comportamiento que pudiera tener el RAH. Al igual que en el caso 2, en un escenario cercano a lo real se muestra que el controlador es capaz de mantener una respuesta lenta y no presentar cambios bruscos al momento de llevar al modelo neuronal a la referencia deseada.

Los resultados en simulación del control de la  $DQO_s$  muestran que cuando la referencia es baja, el ajuste del modelo generado por el controlador, se vuelve más lento y no logra llegar a la referencia en el caso de  $0.45 gDQO_s$ , sin embargo, si se considera un factor de aprendizaje  $\eta$  más grande, por ejemplo  $0.2 \leq \eta < 1$ , el modelo neuronal será capaz de llegar a la referencia ya que aumenta la velocidad de ajuste generado por el gradiente descendente.

No obstante, para referencias pequeñas (debajo de 0.5) el controlador no lograra llevar al sistema a la referencia deseada, aunque el margen de error sea pequeño, esto se debe a que la señal  $u$ , que está

presente en el gradiente descendente del controlador, se disminuye cada vez que el modelo se acerca a la referencia provocando que el ajuste se vuelva muy lento. Además, el incrementar el factor de aprendizaje sacrificaría el objetivo principal del controlador, en sus dos escenarios, que es *tener una respuesta suave y lenta que represente un comportamiento cercano al procesos real*.

El objetivo de control en esta tesis es la *regulación*, además, en el Anexo C.1 también se presenta otro resultado del controlador neuronal el cual se utiliza como objetivo de control el *seguimiento de trayectoria* solamente para la variable de interés que es la  $DQO_s$  a la salida. El fin de esta prueba es la de corroborar que el controlador neuronal también puede ajustar el modelo neuronal a cualquier cambio de referencia que haya durante el proceso.

Con los resultados presentados en simulación, para los dos objetivos de control: *regulación y seguimiento de trayectoria*, se afirma que el usar un control inteligente es adecuado para controlar procesos biológicos que presentan cambios estocásticos en sus entradas y salidas.

Además, se realizó una prueba para comprobar que si el controlador es capaz de rechazar perturbaciones (Anexo C.2). Los resultados muestran que el controlador no es capaz de rechazar perturbaciones con referencias pequeñas, esto se debe a la operación considerada de la electroválvula porque se considera un flujo de  $0.8L/min$  como relación a una apertura al 25 % de la electroválvula donde al tener una referencia muy pequeña lleva al controlador a pedir una apertura cercana al 0 %.

Sin embargo, si se realiza una consideración de una referencia mucho mayor esto permite tener un mayor rango de operación de la electroválvula por lo cual el controlador tiene la capacidad de rechazar las perturbaciones que sucedan al reactor, no obstante, en este trabajo de tesis no se busca modificar la referencia del proceso porque se requiere reducir el sustrato a menor valor posible.

Aunque en simulaciones resulta adecuado la identificación y el control utilizando RNA, el implementar dichas estructuras en el RAH hubieran sido interesantes, ya que se podría hacer una validación en línea de la identificación y comprobar que si el controlador es capaz de mantener el sistema en la referencia deseada de acuerdo a los resultados obtenidos en simulación. Sin embargo, esto no se pudo llevar a cabo por falta de tiempo y otros factores de causa mayor.

## 5.2. Aportaciones

Las aportaciones en este trabajo de tesis son:

- La identificación de las variables:  $DQO_s$  de salida, biogás, porcentaje de metano producido y porcentaje de remoción de  $DQO_s$  del RAH utilizando RNA.
- El diseño de un controlador usando una RNA recurrente para el control de la  $DQO_s$  de salida del RAH, sin embargo, al no tener físicamente la forma de evaluar el controlador se estableció un mecanismo basado en un modelo neuronal entrenado (del proceso) para probar el desempeño del controlador.

## 5.3. Trabajos futuros

Con las actividades realizadas con este tema de investigación existe una gran cantidad de trabajos que se pueden realizar como las que mencionan a continuación:

---

- 
1. Teniendo reportado una mayor cantidad de datos de operación del RAH del ITO desde el arranque hasta el estado estable se puede realizar diferentes tipos de identificación de caja negra y probar la identificación en línea.
  2. Si se cuenta con la instrumentación completa, se puede probar cambiar el esquema de identificación SISO donde la entrada del RAH sería el flujo del sustrato, además se podrían plantear ciertos umbrales del controlador si el electrodo de conductividad se encuentra instalado, ya que el electrodo, de acuerdo a donde se encuentre instalado en la columna del reactor, funciona como un indicador de límite de crecimiento del lecho fluidizante.
  3. Con los modelos identificados se podrían plantear diferentes tipos de controladores para el proceso del RAH e *implementar* los controladores que se propongan, además de probar también el controlador propuesto en esta tesis.
  4. Utilizar otro tipo de función costo para el controlador neuronal en la que permita llevar al sistema a referencias pequeñas.
  5. Como último el desarrollo de un modelo matemático para el RAH del ITO.
-

---

# Bibliografía

---

- [Alvarado-Lassman, 2004] Alvarado-Lassman, A. (2004). *Descripción matemática de la remoción de carbono y nitrógeno en un lecho fluidizado inverso en aguas residuales agroalimentarias*. PhD thesis, Instituto Tecnológico de Veracruz.
- [Antwi et al., 2018] Antwi, P., Li, J., Meng, J., Deng, K., Quashie, F. K., Li, J., and Boadi, P. O. (2018). Feedforward neural network model estimating pollutant removal process within mesophilic upflow anaerobic sludge blanket bioreactor treating industrial starch processing wastewater. *Bioresource technology*, 257:102–112.
- [Arnaiz et al., 2003] Arnaiz, C., Buffiere, P., Elmaleh, S., Lebrato, J., and Moletta, R. (2003). Anaerobic digestion of dairy wastewater by inverse fluidization: the inverse fluidized bed and the inverse turbulent bed reactors. *Environmental technology*, 24(11):1431–1443.
- [Azwar et al., 2015] Azwar, M., Hussain, M., Wahab, A. A., and Zani, M. (2015). A comparative study between neural networks (nn)-based and adaptive-pid controllers for the optimal bio-hydrogen gas production in microbial electrolysis cell reactor. In *Computer Aided Chemical Engineering*, volume 37, pages 1529–1534. Elsevier.
- [Azzaro and Veiga, 2015] Azzaro, J. E. and Veiga, R. A. (2015). Sliding mode controller with neural network identification. *IEEE Latin America Transactions*, 13(12):3754–3757.
- [Baruah et al., 2017] Baruah, D., Baruah, D., and Hazarika, M. (2017). Artificial neural network based modeling of biomass gasification in fixed bed downdraft gasifiers. *Biomass and bioenergy*, 98:264–271.
- [Batstone et al., 2002] Batstone, D. J., Keller, J., Angelidaki, I., Kalyuzhnyi, S., Pavlostathis, S., Rozzi, A., Sanders, W., Siegrist, H., and Vavilin, V. (2002). The iwa anaerobic digestion model no 1 (adm1). *Water Science and technology*, 45(10):65–73.
- [Borja and Rincón, 2017] Borja, R. and Rincón, B. (2017). Biogas production.
- [Caraman et al., 2017] Caraman, S., Barbu, M., Ifrim, G., Titica, M., and Ceanga, E. (2017). Anaerobic digester optimization using extremum seeking and model-based algorithms. a comparative study. *IFAC-PapersOnLine*, 50(1):12673–12678.
- [Chan et al., 1993] Chan, H.-C., Chau, K., and Chan, C. (1993). A neural network controller for switching power converters. In *Proceedings of IEEE Power Electronics Specialist Conference-PESC'93*, pages 887–892. IEEE.
- [Chidrawar and Patre, 2008] Chidrawar, S. and Patre, B. (2008). Generalized predictive control and neural generalized predictive control. *Leonardo J. Sci*, 13:133–152.
- [Chidrawar and Patre, 2011] Chidrawar, S. K. and Patre, B. M. (2011). Hybrid direct neural network controller with linear feedback compensator. In *MAICS*, pages 208–213.

- [Cichocki and Unbehauen, 1993] Cichocki, A. and Unbehauen, R. (1993). *Neural networks for optimization and signal processing*. John Wiley & Sons, Inc.
- [Darken et al., 1992] Darken, C., Chang, J., and Moody, J. (1992). Learning rate schedules for faster stochastic gradient search. In *Neural Networks for Signal Processing II Proceedings of the 1992 IEEE Workshop*, pages 3–12. IEEE.
- [Darken and Moody, 1992] Darken, C. and Moody, J. (1992). Towards faster stochastic gradient search. In *Advances in neural information processing systems*, pages 1009–1016.
- [Della Bona et al., 2015] Della Bona, A., Ferretti, G., Ficara, E., and Malpei, F. (2015). Lft modelling and identification of anaerobic digestion. *Control Engineering Practice*, 36:1–11.
- [DI and PJ, 1997] DI, S. and PJ, H. (1997). Neural generalized predictive control: A newton-raphson implementation.
- [Dochain, 2013] Dochain, D. (2013). *Automatic Control of Bioprocesses*. ISTE. Wiley.
- [García-Gen et al., 2015] García-Gen, S., Rodríguez, J., and Lema, J. M. (2015). Control strategy for maximum anaerobic co-digestion performance. *Water research*, 80:209–216.
- [Giles et al., 1998] Giles, C. L., Gori, M., Goos, G., Hartmanis, J., and Van Leeuwen, J. (1998). *Adaptive Processing of Sequences and Data Structures: International Summer School on Neural Networks, ER Caianiello", Vietri Sul Mare, Salerno, Italy, September 6-13, 1997, Tutorial Lectures*, volume 1387. Springer Science & Business Media.
- [Hernández, 2005] Hernández, P. A. Q. (2005). *Métodos numéricos: con aplicaciones en Excel*. Reverté.
- [Irwin and Wikamowski, 2011] Irwin, J. D. and Wikamowski, B. M. (2011). The industrial electronics handbook: Intelligent systems.
- [Jarquin, 2011] Jarquin, J.-P. R. (2011). Diseño e implementación de un sistema de diagnóstico y aislamiento de fallas de un digestor anaerobio usab. Master's thesis, Instituto Tecnológico de Orizaba.
- [Jarquin, 2016] Jarquin, J. P. R. (2016). *Diseño e implementación de un Control Tolerante a Fallas para un Biorreactor UASB*. PhD thesis, CENIDET.
- [Karamanev and Nikolov, 1996] Karamanev, D. and Nikolov, L. (1996). Application of inverse fluidization in wastewater treatment: From laboratory to full-scale bioreactors. *Environmental progress*, 15(3):194–196.
- [Karimi et al., 2017] Karimi, M., Vaferi, B., Hosseini, S. H., and Rasteh, M. (2017). Designing an efficient artificial intelligent approach for estimation of hydrodynamic characteristics of tapered fluidized bed from its design and operating parameters. *Industrial & Engineering Chemistry Research*, 57(1):259–267.
- [Kasparian and Batur, 1998] Kasparian, V. and Batur, C. (1998). Model reference based neural network adaptive controller. *ISA transactions*, 37(1):21–39.
- [Khanongnuch et al., 2018] Khanongnuch, R., Di Capua, F., Lakaniemi, A.-M., Rene, E. R., and Lens, P. N. (2018). Effect of n/s ratio on anoxic thiosulfate oxidation in a fluidized bed reactor: Experimental and artificial neural network model analysis. *Process Biochemistry*, 68:171–181.
- [Kil et al., 2017] Kil, H., Li, D., Xi, Y., and Li, J. (2017). Model predictive control with on-line model identification for anaerobic digestion processes. *Biochemical engineering journal*, 128:63–75.
-

- [Krzywanski et al., 2017] Krzywanski, J., Grabowska, K., Herman, F., Pyrka, P., Sosnowski, M., Prauzner, T., and Nowak, W. (2017). Optimization of a three-bed adsorption chiller by genetic algorithms and neural networks. *Energy Conversion and Management*, 153:313–322.
- [Kwok et al., 2000] Kwok, K. E., Ping, M. C., and Li, P. (2000). A model-based augmented pid algorithm. *Journal of Process Control*, 10(1):9–18.
- [Larson, 2002] Larson, H. (2002). *Introducción a la Teoría de Probabilidades e Inferencia Estadística*. Limusa.
- [Li et al., 2019] Li, D., Chen, C. P., Liu, Y.-J., and Tong, S. (2019). Neural network controller design for a class of nonlinear delayed systems with time-varying full-state constraints. *IEEE transactions on neural networks and learning systems*.
- [Li et al., 2018] Li, G., Liu, Z., Li, J., Fang, Y., Shan, J., Guo, S., and Wang, Z. (2018). Modeling of ash agglomerating fluidized bed gasifier using back propagation neural network based on particle swarm optimization. *Applied Thermal Engineering*, 129:1518–1526.
- [Ljung, 1999] Ljung, L. (1999). *System identification (2nd ed.): theory for the user*. PTR Prentice Hall.
- [Marin et al., 1999] Marin, P., Alkalay, D., Guerrero, L., Chamy, R., and Schiappacasse, M. (1999). Design and startup of an anaerobic fluidized bed reactor. *Water Science and Technology*, 40(8):63–70.
- [Marín-Peña, 2017] Marín-Peña, O. (2017). Generación de bioenergéticos a partir de residuos sólidos orgánicos en un reactor de lecho fluidizado inverso con arreglo en paralelo. Master’s thesis, Instituto Tecnológico de Orizaba.
- [Meléndez and Gutiérrez, 2004] Meléndez, J. and Gutiérrez, B. (2004). Guía para el laboratorio de fenómenos de transporte i. *Departamento de Termodinámica y Fenómenos de Transferencia, Universidad Simón Bolívar, Valle de Sartenejal*.
- [Méndez-Acosta et al., 2016] Méndez-Acosta, H., Campos-Rodríguez, A., González-Álvarez, V., García-Sandoval, J., Snell-Castro, R., and Latrille, E. (2016). A hybrid cascade control scheme for the vfa and cod regulation in two-stage anaerobic digestion processes. *Bioresource technology*, 218:1195–1202.
- [Narendra and Parthasarathy, 1990] Narendra, K. S. and Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on neural networks*, 1(1):4–27.
- [Nørgård et al., 2000] Nørgård, P. M., Ravn, O., Poulsen, N. K., and Hansen, L. K. (2000). *Neural Networks for Modelling and Control of Dynamic Systems-A Practitioner’s Handbook*. Springer-Verlag London.
- [Ogata, 2003] Ogata, K. (2003). *Ingeniería de control moderna*. Pearson Educación.
- [Piña, 2010] Piña, A. J. B. (2010). *Síntesis de Sistemas de Control Borroso Estables por Diseño: Tesis Doctoral*. A. Javier Barragán Piña.
- [Piskunov et al., 1983] Piskunov, N., Medkov, K., et al. (1983). *Cálculo diferencial e integral*, volume 1. Mir.
-

- [Poh et al., 2016] Poh, P., Gouwanda, D., Mohan, Y., Gopalai, A., and Tan, H. (2016). Optimization of wastewater anaerobic digestion using mechanistic and meta-heuristic methods: current limitations and future opportunities. *Water Conservation Science and Engineering*, 1(1):1–20.
- [Poznyak et al., 2001] Poznyak, A. S., Sanchez, E. N., and Yu, W. (2001). *Differential neural networks for robust nonlinear control: identification, state estimation and trajectory tracking*. World Scientific.
- [Pratiwi et al., 2015] Pratiwi, B., RamdhanKirom, M., and FauziIskandar, R. (2015). Design of temperature control based fuzzy logic for substrate in thermophilic hydrogen reactor. In *2015 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, pages 85–90. IEEE.
- [Reyes-Alvarado et al., 2017] Reyes-Alvarado, L. C., Okpalanze, N. N., Kankanala, D., Rene, E. R., Esposito, G., and Lens, P. N. (2017). Forecasting the effect of feast and famine conditions on biological sulphate reduction in an anaerobic inverse fluidized bed reactor using artificial neural networks. *Process Biochemistry*, 55:146–161.
- [Rincón et al., 2015] Rincón, A., Piarpuzán, D., and Angulo, F. (2015). A new adaptive controller for bio-reactors with unknown kinetics and biomass concentration: Guarantees for the boundedness and convergence properties. *Mathematics and Computers in Simulation*, 112:1–13.
- [Rincón, ] Rincón, J. A. C. Derivación bajo la integral.
- [Robles et al., 2015] Robles, Á., Durán, F., Ruano, M. V., Ribes, J., Rosado, A., Seco, A., and Ferrer, J. (2015). Instrumentation, control, and automation for submerged anaerobic membrane bioreactors. *Environmental technology*, 36(14):1795–1806.
- [Robles Martinez et al., 2015] Robles Martinez, A., Latrille, E., Ruano, M., and Steyer, J.-P. (2015). A robust fuzzy-logic-based controller for bio-methane production in anaerobic fixed-film reactors. In *2015; 14. World Congress on Anaerobic Digestion (AD14), Viña del Mar, CHL, 2015-11-15-2015-11-18,*.
- [Rodríguez, 2015] Rodríguez, G. G. (2015). *Estudio de la influencia de la relación C/N en la desnitrificación y metanogénesis de aguas residuales en un reactor anaeróbico de lecho fluidizado inverso*. PhD thesis, Universidad Politécnica de Madrid.
- [Sandoval Rojas, 2016] Sandoval Rojas, M. E. (2016). *Tratamiento de vinazas provenientes de etanol en un reactor de lecho fluidizado inverso*. PhD thesis, Caminos.
- [Santín et al., 2016] Santín, I., Pedret, C., Vilanova, R., and Meneses, M. (2016). Advanced decision control system for effluent violations removal in wastewater treatment plants. *Control Engineering Practice*, 49:60–75.
- [Sarangapani, 2006] Sarangapani, J. (2006). *Neural network control of nonlinear discrete-time systems*. CRC press.
- [Shamiri et al., 2015] Shamiri, A., Wong, S. W., Zamil, M. F., Hussain, M. A., and Mostoufi, N. (2015). Modified two-phase model with hybrid control for gas phase propylene copolymerization in fluidized bed reactors. *Chemical Engineering Journal*, 264:706–719.
- [Shargh and Behnajady, 2016] Shargh, M. and Behnajady, M. A. (2016). A high-efficient batch-recirculated photoreactor packed with immobilized tio2-p25 nanoparticles onto glass beads for photocatalytic degradation of phenazopyridine as a pharmaceutical contaminant: artificial neural network modeling. *Water Science and Technology*, 73(11):2804–2814.
-

- 
- [Steiner, 2005] Steiner, E. (2005). *Matemáticas para las ciencias aplicadas*. Reverté.
- [Tan et al., 2017] Tan, H., Lew, J., Gouwanda, D., and Poh, P. (2017). Fuzzy logic modelling for thermophilic anaerobic digestion of palm oil mill effluent (pome) treatment. In *2017 4th International Conference on Industrial Engineering and Applications (ICIEA)*, pages 265–269. IEEE.
- [Ugalde, 2008] Ugalde, H. M. R. (2008). Identificación de sistemas utilizando redes neuronales. Master’s thesis, CENIDET.
- [Ugalde, 2013] Ugalde, H. M. R. (2013). *Identificación de Sistemas Utilizando Redes Neuronales: Un Enfoque Basado en el Balance Entre Sencillez, Precisión y Costo Computacional*. PhD thesis, CENIDET.
- [Vázquez Trejo, 2017] Vázquez Trejo, J. A. (2017). Diagnóstico de fallas aplicado a sensores de un fotobiorreactor tipo columna de burbujeo tubular. Master’s thesis, CENIDET.
- [Venkatesh Prabhu et al., 2016] Venkatesh Prabhu, M., Karthikeyan, R., and Shanmugaparakash, M. (2016). Modeling and optimization by response surface methodology and neural network–genetic algorithm for decolorization of real textile dye effluent using pleurotus ostreatus: a comparison study. *Desalination and water treatment*, 57(28):13005–13019.
- [Von Sperling and de Lemos Chernicharo, 2005] Von Sperling, M. and de Lemos Chernicharo, C. A. (2005). *Biological wastewater treatment in warm climate regions*, volume 1. IWA publishing.
- [Yahya et al., 2015] Yahya, A. M., Hussain, M. A., and Abdul Wahab, A. K. (2015). Modeling, optimization, and control of microbial electrolysis cells in a fed-batch reactor for production of renewable biohydrogen gas. *International Journal of Energy Research*, 39(4):557–572.
- [Zaidi et al., 2019] Zaidi, I., Chtourou, M., and Djemel, M. (2019). Robust neural control of discrete time uncertain nonlinear systems using sliding mode backpropagation training algorithm. *International Journal of Automation and Computing*, 16(2):213–225.
-



# **Anexos**

# Anexo A

---

## Propiedades básicas

---

### A.1. La derivada parcial

Para determinar la velocidad o el ritmo de cambio de una función de varias variables respecto a una de sus variables independientes se utiliza el proceso de derivación parcial [Piskunov et al., 1983].

Si  $z = f(x, y)$  la derivada parcial de  $f$  con respecto a las variables  $x$  es (A.1)

$$\frac{\partial z}{\partial x} = \frac{\partial f}{\partial x}(x, y) := \lim_{h \rightarrow 0} \frac{f(x+h, y) - f(x, y)}{h}, \quad (\text{A.1})$$

y la derivada parcial de  $f$  con respecto a  $y$  es (A.2)

$$\frac{\partial z}{\partial y} = \frac{\partial f}{\partial y}(x, y) := \lim_{h \rightarrow 0} \frac{f(x, y+h) - f(x, y)}{h}, \quad (\text{A.2})$$

siempre y cuando el límite en cada caso exista. La definición indica que para calcular  $\frac{\partial f}{\partial x}$  se considera  $y$  constante derivando con respecto a  $x$  y para calcular  $\frac{\partial f}{\partial y}$  se considera  $x$  constante derivando con respecto a  $y$ . Pueden aplicarse por tanto las reglas usuales de derivación.

### A.2. Regla de la cadena

Si se tiene  $F(x) = h[g(x)]$ . Donde  $x \in \mathfrak{R}^n$  y  $F(\cdot), h(\cdot), g(\cdot)$  son funciones escalares. La regla de la cadena de derivación puede ser escrita como se muestra en (A.3) [Cichocki and Unbehauen, 1993]:

$$\frac{\partial F}{\partial x_i} = \frac{\partial h}{\partial g} \frac{\partial g}{\partial x_i}. \quad (\text{A.3})$$

### A.3. Método directo de Lyapunov

El matemático Ruso A.M. Lyapunov formuló una poderosa teoría de estabilidad y propuso una técnica para probarla. Uno de los caminos más comunes para estudiar la convergencia de un sistema dinámico descrito por un conjunto de ecuaciones diferenciales es encontrar una función de Lyapunov, también llamada una función de energía desde el método de Lyapunov el cual se basa en el concepto de energía y la relación de la energía almacenada para la estabilidad del sistema [Cichocki and Unbehauen, 1993].

#### A.3.1. Principio del Teorema de Estabilidad de Lyapunov

De manera general un sistema es estable si su función de energía total  $J$ , es una función definida positiva, además es continuamente decreciente, es decir, la derivada con respecto al tiempo de la  $J$  total

es definida negativa, hasta que alcanza un estado de equilibrio. En un sistema no lineal no tiene por qué existir una manera simple de definir una función de energía, por ello, el método de estabilidad por Lyapunov introduce una función de energía auxiliar denominada función candidata de Lyapunov [Piña, 2010].

### A.3.2. Función Candidata de Lyapunov

Suponga que  $J : \mathfrak{R}^n \rightarrow \mathfrak{R}$  es una función escalar.  $J$  es una función candidata de Lyapunov si es una función positiva definida localmente, es decir, la función candidata de Lyapunov es mayor que cero para todo valor en los reales excepto el cero, y el único valor donde tiene un valor de cero es el mismo cero (A.4):

$$\begin{aligned} J(0) &= 0, \\ J(x) &> 0 \quad \forall x \in \mathfrak{R} \setminus \{0\}. \end{aligned} \tag{A.4}$$

### A.3.3. Teorema de Estabilidad de Lyapunov

**Teorema 2.1.** Considere el sistema  $\dot{x} = f(x, t)$ , donde  $f(0, t) = 0$ . Si  $\exists$  una función escalar  $J(x, t)$  con las primeras derivadas parciales continuas que satisface las condiciones (A.5 y A.6):

$$J(x, t) \text{ es definida positiva, es decir, } J(x, t) > 0 \quad \forall x, t \neq 0, \tag{A.5}$$

$$\dot{J}(x, t) \text{ es definida negativa, es decir, } \dot{J}(x, t) < 0 \quad \forall x, t \neq 0. \tag{A.6}$$

entonces el estado de equilibrio en el origen es uniforme y asintóticamente estable. Si se satisface A.5 pero  $\dot{J}$  es semidefinida negativa, es decir,  $\dot{J}(x, t) \leq 0 \quad \forall x$  entonces el estado de equilibrio en el origen es estable.

## A.4. El algoritmo de gradiente descendente

Para métodos de optimización sin restricciones es utilizada la técnica de gradiente descendente, la cual esta técnica transforma el problema de minimización en un sistema asociado de primer orden como se presenta en (A.7) [Cichocki and Unbehauen, 1993] :

$$\frac{dx_j}{dt} = - \sum_{i=1}^n \mu_{ji} \frac{\partial E}{\partial x_i}, \tag{A.7}$$

Con condiciones iniciales  $x_j(0) = x_j^0$ . El sistema (A.7) puede ser escrito en la forma matricial compacta como se expresa en (A.8).

$$\frac{dx_j}{dt} = -\mu(x, t) \frac{\partial E}{\partial x_i} \tag{A.8}$$

Donde:  $\dot{x} = [x_1, x_2, \dots, x_n]^T$ ,  $x = [x_1, x_2, \dots, x_n]^T$ ,  $\frac{\partial E}{\partial x} = \left[ \frac{\partial E}{\partial x_1}, \frac{\partial E}{\partial x_2}, \dots, \frac{\partial E}{\partial x_n} \right]^T$ .

$\mu(x, t)$  es una matriz simétrica positiva definida también llamada matriz de aprendizaje con entradas dependientes. Además para encontrar el vector deseado  $x^*$  que minimiza la función de energía o función costo  $E(x)$ , que debe cumplir con el teorema de estabilidad de Lyapunov, la cual es necesaria para resolver o simular el sistema de ecuaciones diferenciales con condiciones iniciales. Esto quiere decir que el mínimo de la función de energía o la función objetivo es determinado por la siguiente expresión que se muestra en (A.9).

$$x^* = \lim_{t \rightarrow \infty} x(t) \tag{A.9}$$

Para mostrar que el sistema de ecuaciones diferenciales expresado en (A.8) es estable dejamos la derivada con respecto al tiempo de la función energía como se muestra en (A.10).

$$\frac{dE}{dt} = \sum_{j=1}^n \frac{\partial E}{\partial x_j} \frac{dx_j}{dt} = - \left( \frac{\partial E}{\partial x} \right)^T \mu(x, t) \left( \frac{\partial E}{\partial x} \right) \leq 0 \quad (\text{A.10})$$

Tomando la condición que la matriz  $\mu(x, t)$  es simétrica y positiva definida. La ecuación (A.10) garantiza que la función de energía  $E(x)$  decrece en el tiempo y converge a un mínimo local estable cuando  $t \rightarrow \infty$ . La velocidad de convergencia depende de la elección de la matriz de entradas  $\mu(x, t)$ . En el caso más simple la matriz  $\mu(x, t)$  es reducida a una matriz unitaria multiplicada por una constante positiva  $\mu_0$ , dando como resultado un sistema simplificado de la forma expresada en (A.11).

$$\frac{dx_j}{dt} = -\mu_0 \frac{\partial E(x)}{\partial x_j}, \quad x(0) = x^0, \quad (j = 1, 2, \dots, n). \quad (\text{A.11})$$

Donde el coeficiente positivo  $\mu_0$ , es llamado el parámetro de aprendizaje. La forma discreta del gradiente descendente toma la siguiente forma expresada en (A.12)

$$x_j(k+1) = x_j(k\tau) - \eta^{(k)} \frac{\partial E(x)}{\partial x_j} \Big|_{x_j=x_j^{(k)}} \quad (\text{A.12})$$

donde  $0 < \eta^{(k)} < \eta_{max}$ . el parámetro  $\eta^{(k)}$  es comúnmente referido como el valor de aprendizaje o paso de integración.

## A.5. Interpolación Lineal

La interpolación lineal es el método más simple en uso hoy en día. Es el método usado por los programas de generación de gráficas, donde se interpola con líneas rectas entre una serie de puntos que el usuario quiere graficar [Steiner, 2005], [Hernández, 2005].

La idea básica es conectar los 2 puntos dados en  $x_i$ , es decir  $(x_0, y_0)$  y  $(x_1, y_1)$ . La función interpolante es una línea recta entre los dos puntos. Para cualquier punto entre los dos valores de  $x_0$  y  $x_1$  se debe seguir la ecuación de la recta:

$$\frac{y - y_0}{y_1 - y_0} = \frac{x - x_0}{x_1 - x_0}, \quad (\text{A.13})$$

que se puede derivar geoméricamente.

En lo anterior, el único valor desconocido es  $y$ , que representa el valor desconocido para  $x$ , despejando queda:

$$y = y_0 + \frac{y_1 - y_0}{x_1 - x_0} (x - x_0), \quad (\text{A.14})$$

donde se asume que  $x_0 < x < x_1$ , de otra forma esto se conocería como extrapolación.

Si se tienen más de dos puntos para la interpolación, es decir  $N > 2$ , con puntos  $x_0, x_1, \dots, x_N$ , simplemente se concatena la interpolación lineal entre pares de puntos continuos.

# Anexo B

---

## Identificación fuera de línea de otras variables

---

Aquí se presentan las identificaciones tipo SISO de otras variables de interés, al igual que la  $DQO_s$  se realizó identificación tipo SISO usando como entrada la CVA.

### B.1. Identificación de Biogás

Siguiendo el esquema de entrenamiento de la Figura 2.18 las condiciones de la identificación neuronal en esquema *serie-paralelo* son las siguientes:

- Pesos sinápticos iniciales: valores aleatorios en un rango de  $[-0.1, 0.1]$ .
- 5 neuronas en la capa 1 para  $J_u$  y  $J_{y_n}$ .
- Solo usa un retardo  $ret_u = 1$ ,  $ret_y = 1$ .
- La entrada está escalada 1 : 12 y la salida 1 : 140.
- La identificación es tipo SISO (entrada: CVA (Figura 4.3) y salida: biogás).
- $\eta_0 = 0.1$

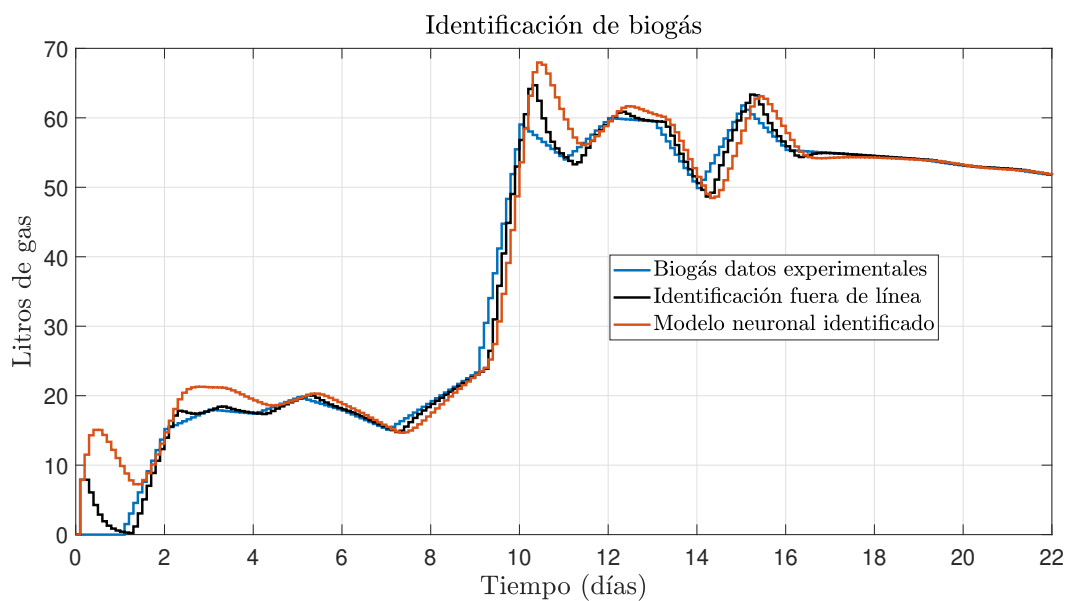


Figura B.1. Identificación de biogás.

Biogás:

- Ajuste de identificación fuera de línea  $FIT = 90.7638\%$
- Ajuste de modelo identificado  $FIT = 79.6621\%$

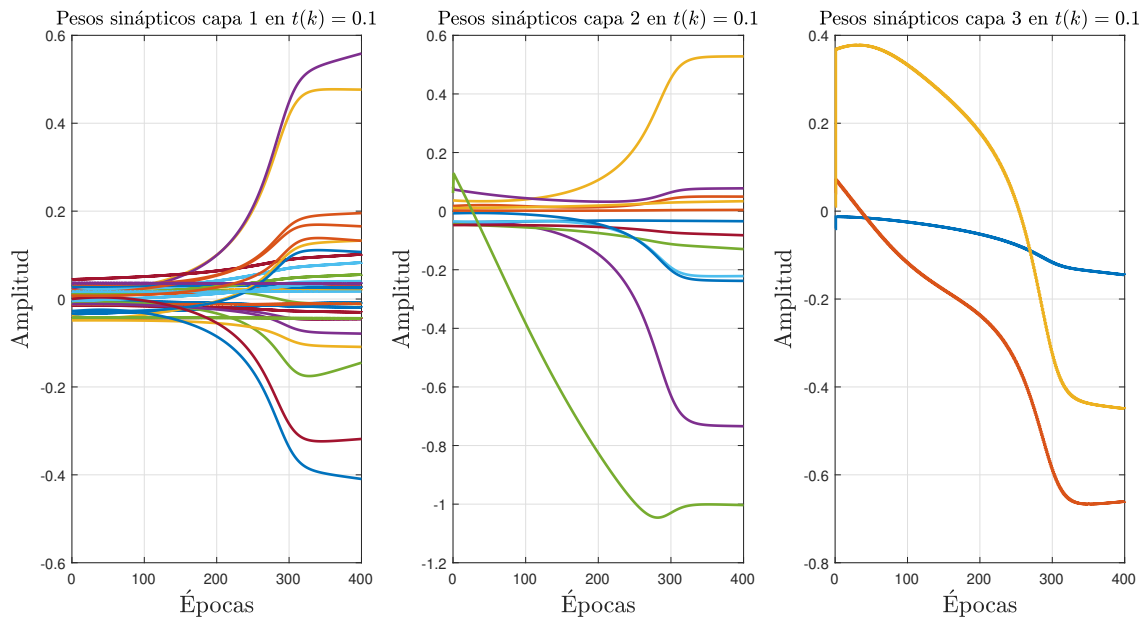


Figura B.2. Adaptación de pesos sinápticos por épocas de la identificación del biogás.

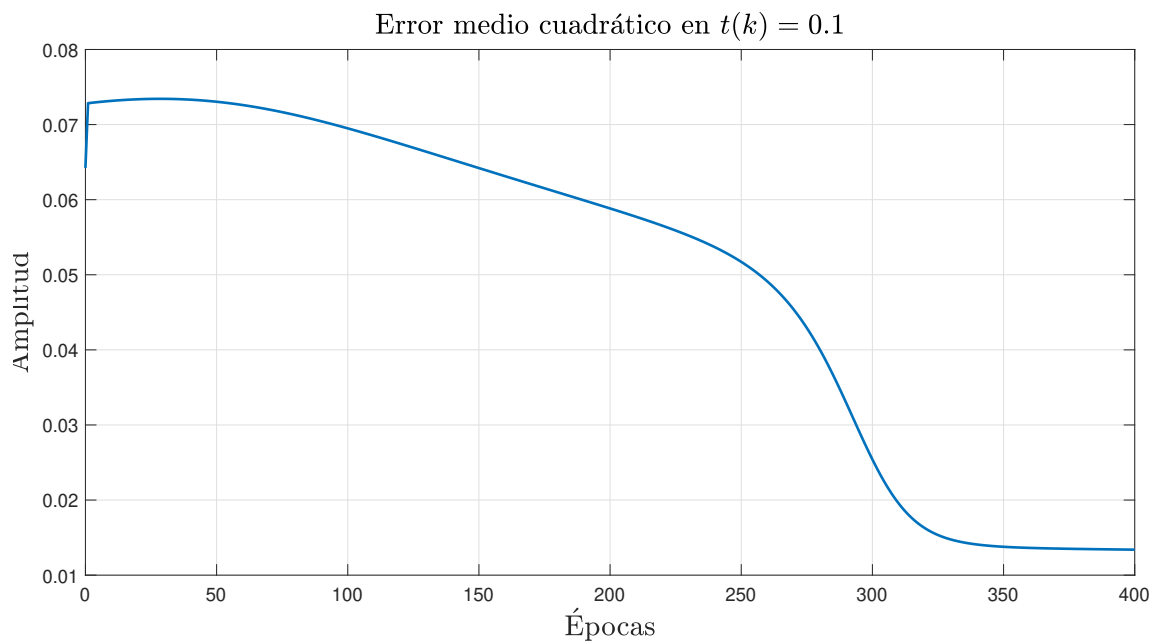


Figura B.3. EMC de biogás.

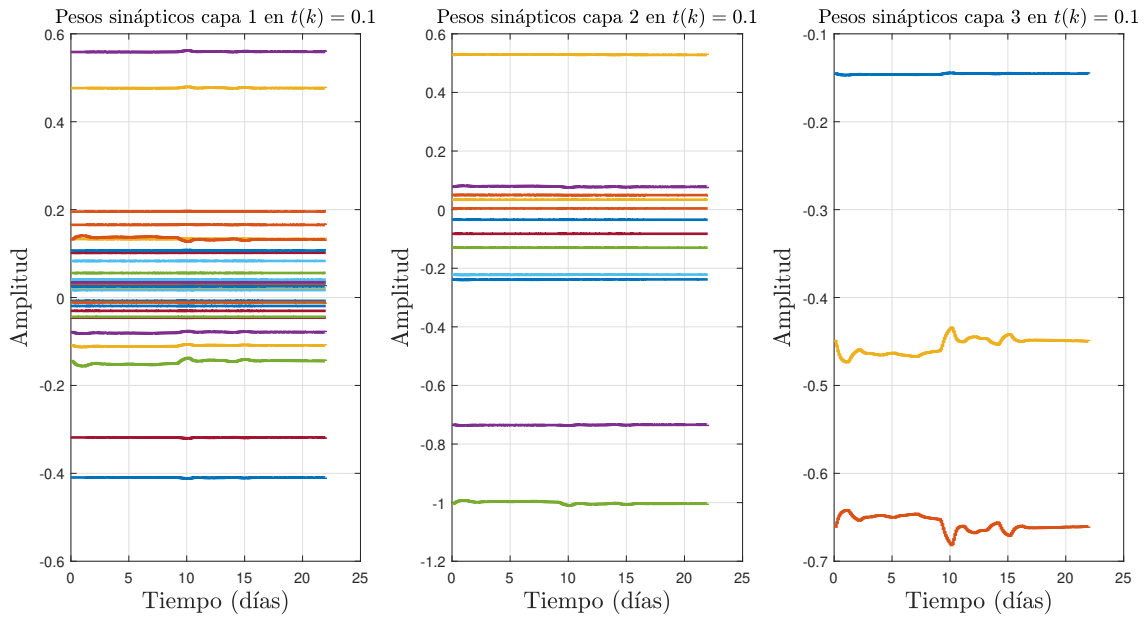


Figura B.4. Pesos sinápticos para modelo neuronal de biogás.

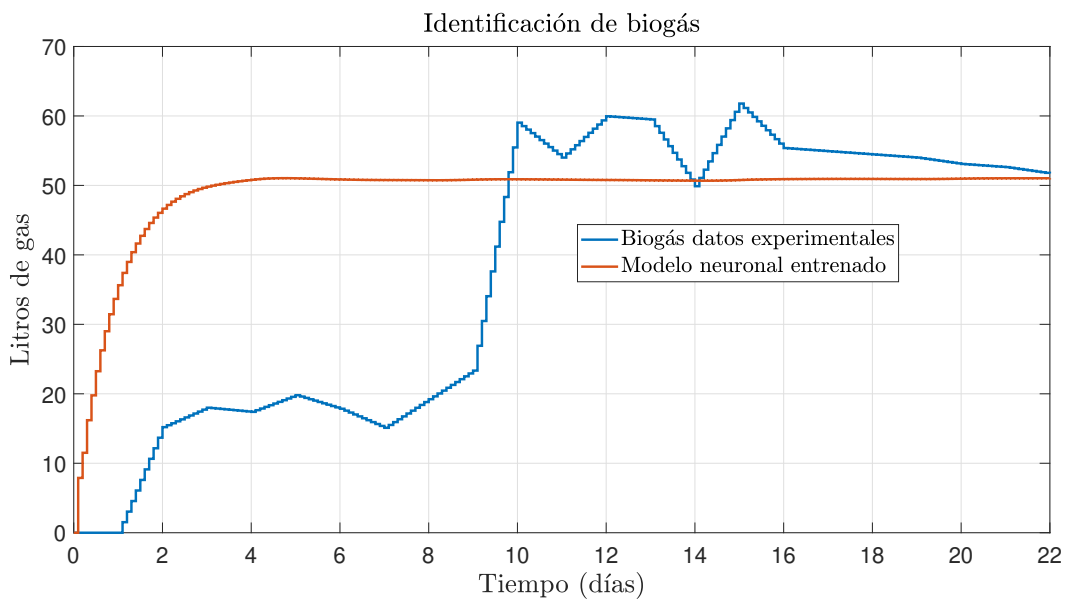


Figura B.5. Modelo entrenado de biogás.

## B.2. Identificación de porcentaje de remoción de $DQO_s$

Siguiendo el esquema de entrenamiento de la Figura 2.18 las condiciones de la identificación neuronal en esquema *serie-paralelo* son las siguientes:

- Pesos sinápticos iniciales: valores aleatorios en un rango de  $[-0.1, 0.1]$ .
- 2 neuronas en la capa 1 para  $J_u$  y  $J_{y_n}$ .
- Solo usa un retardo  $ret_u = 1$ ,  $ret_y = 1$ .
- La entrada está escalada 1 : 12 y la salida 1 : 100.
- La identificación es tipo SISO (entrada:  $CVA$  (Figura 4.3) y salida: remoción de  $DQO_s$ ).
- $\eta_0 = 0.1$

Remoción de  $DQO_s$ :

- Ajuste de identificación fuera de línea  $FIT = 66.9853\%$
- Ajuste de modelo identificado  $FIT = 50.7014\%$

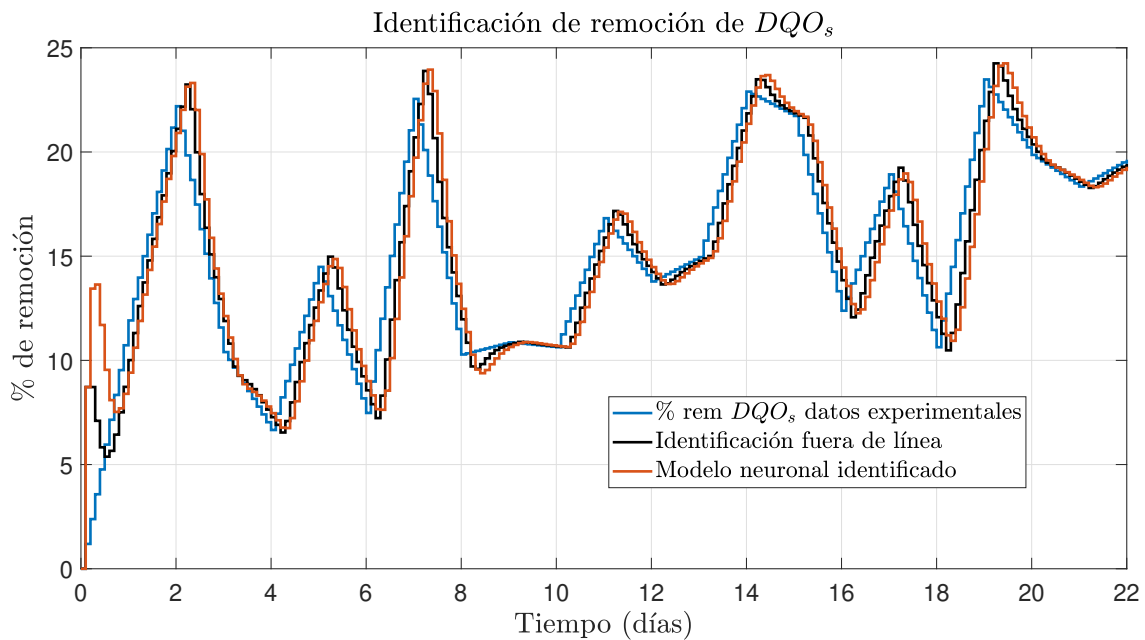


Figura B.6. Identificación de porcentaje de remoción de  $DQO_s$ .



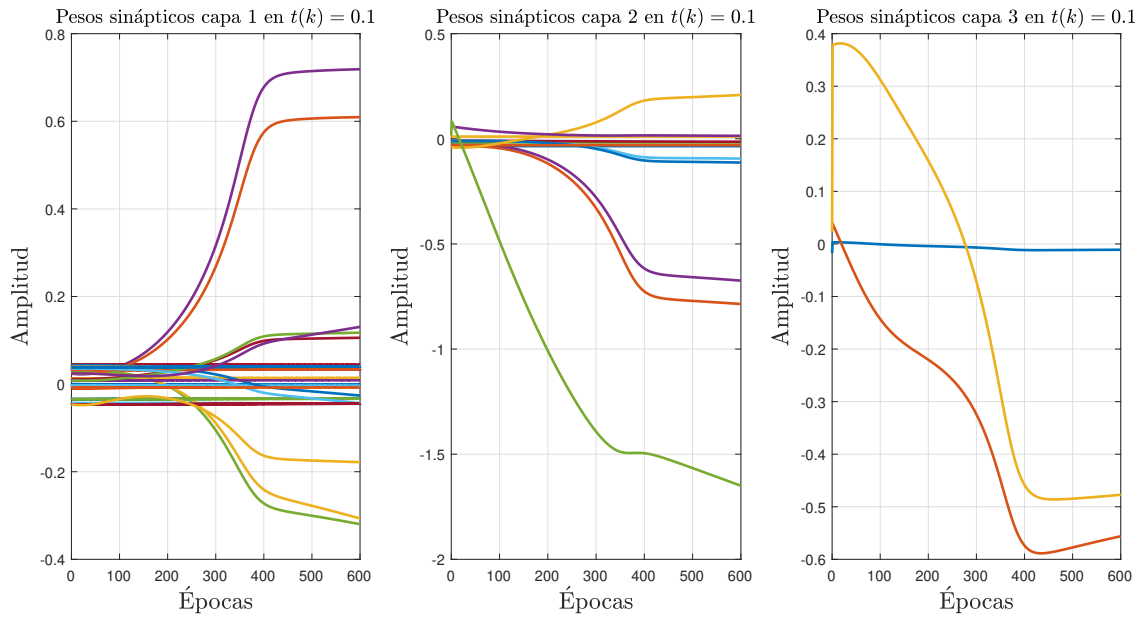


Figura B.7. Adaptación de pesos sinápticos por épocas.

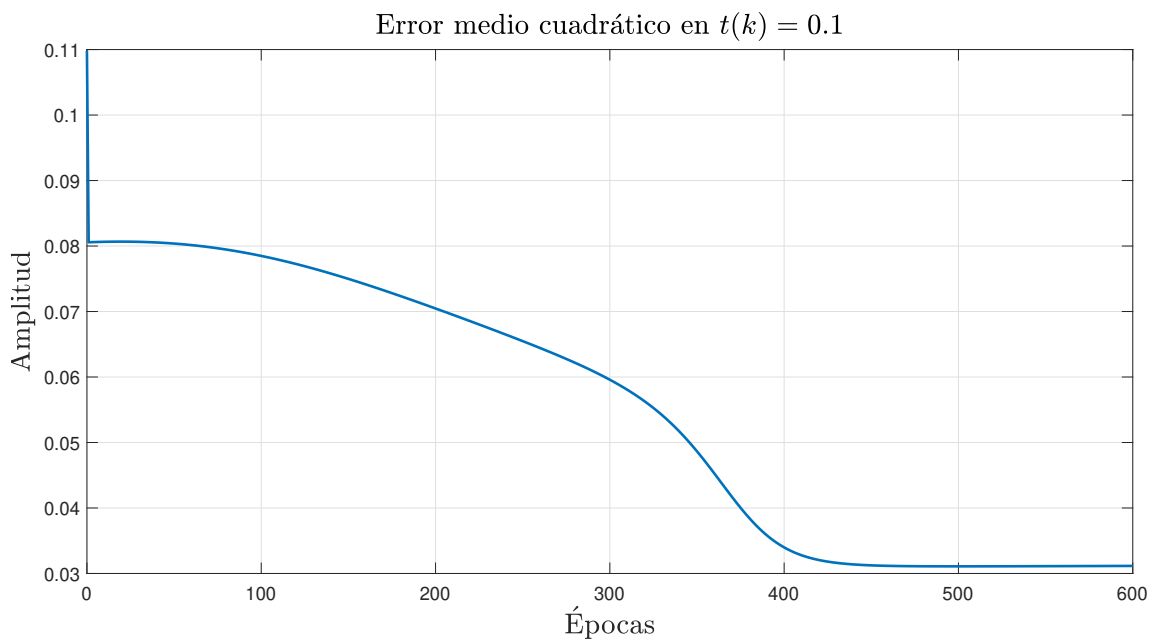


Figura B.8. EMC del porcentaje de remoción de  $DQO_s$ .

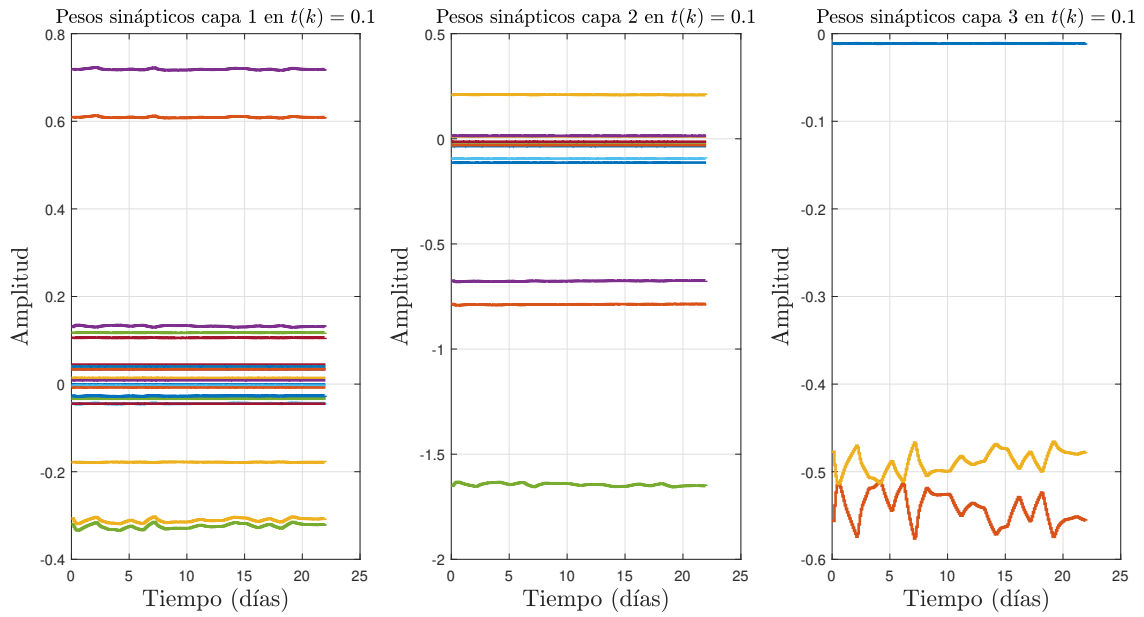


Figura B.9. Pesos sinápticos para modelo neuronal de remoción de  $DQO_s$ .

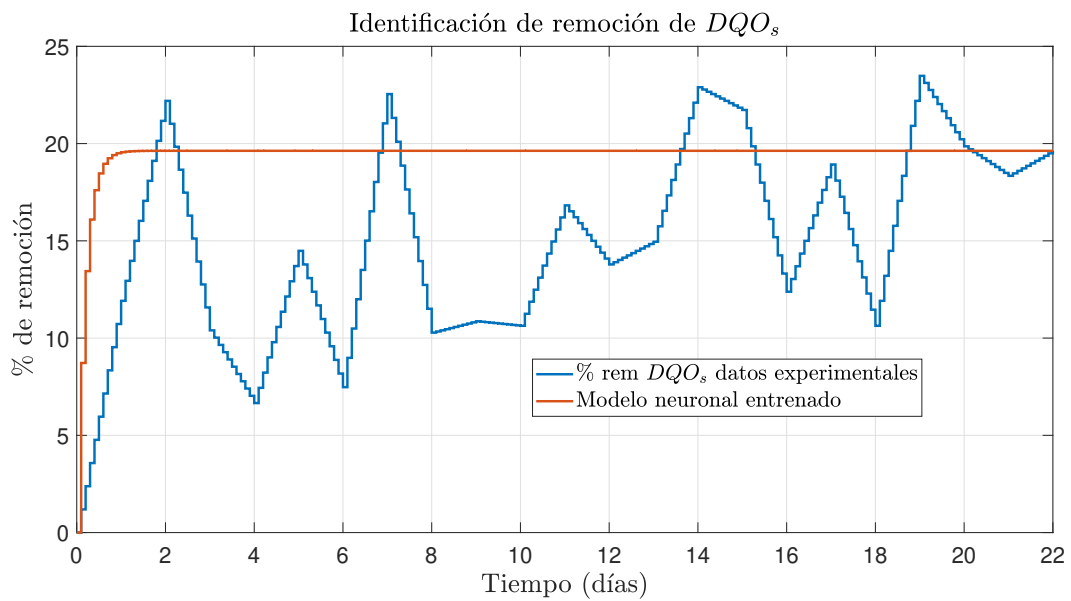


Figura B.10. Modelo entrenado de remoción.

### B.3. Identificación de porcentaje de metano producido

Siguiendo el esquema de entrenamiento de la Figura 2.18 las condiciones de la identificación neuronal en esquema *serie-paralelo* son las siguientes:

- Pesos sinápticos iniciales: valores aleatorios en un rango de  $[-0.1, 0.1]$ .
- 3 neuronas en la capa 1 para  $J_u$  y  $J_{y_n}$ .
- Solo usa un retardo  $ret_u = 1$ ,  $ret_y = 1$ .
- La entrada está escalada 1 : 12 y la salida 1 : 200.
- La identificación es tipo SISO (entrada: *CVA* (Figura 4.3) y salida: porcentaje de metano).
- $\eta_0 = 0.1$

Porcentaje de metano producido:

- Ajuste de identificación fuera de línea  $FIT = 87.7098\%$
- Ajuste de modelo identificado  $FIT = 77.8711\%$

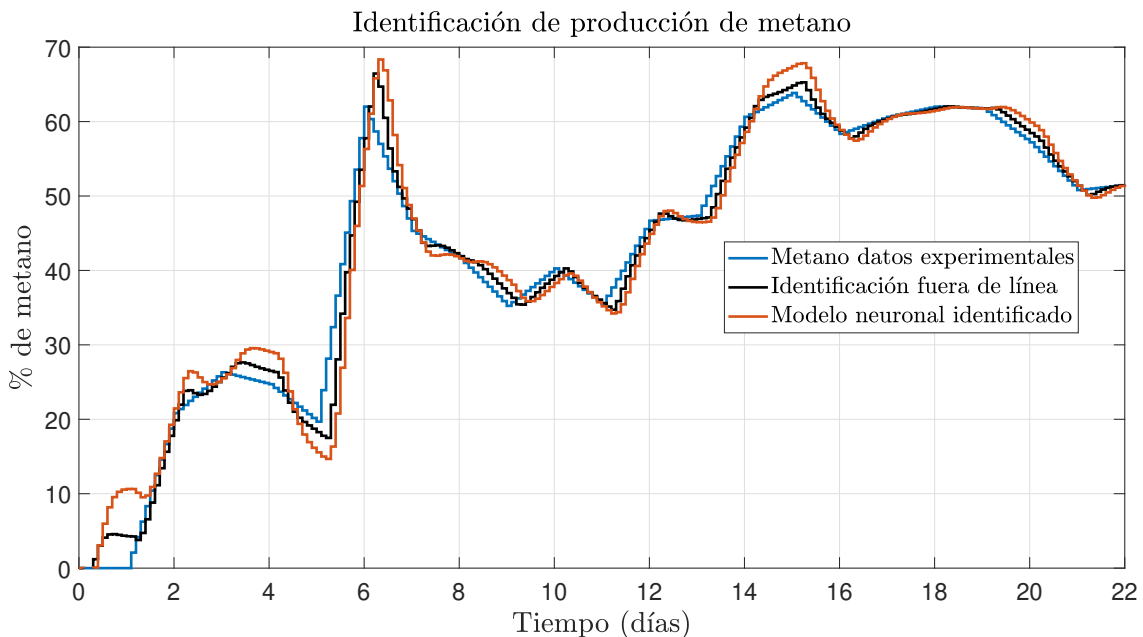


Figura B.11. Identificación de porcentaje de metano.

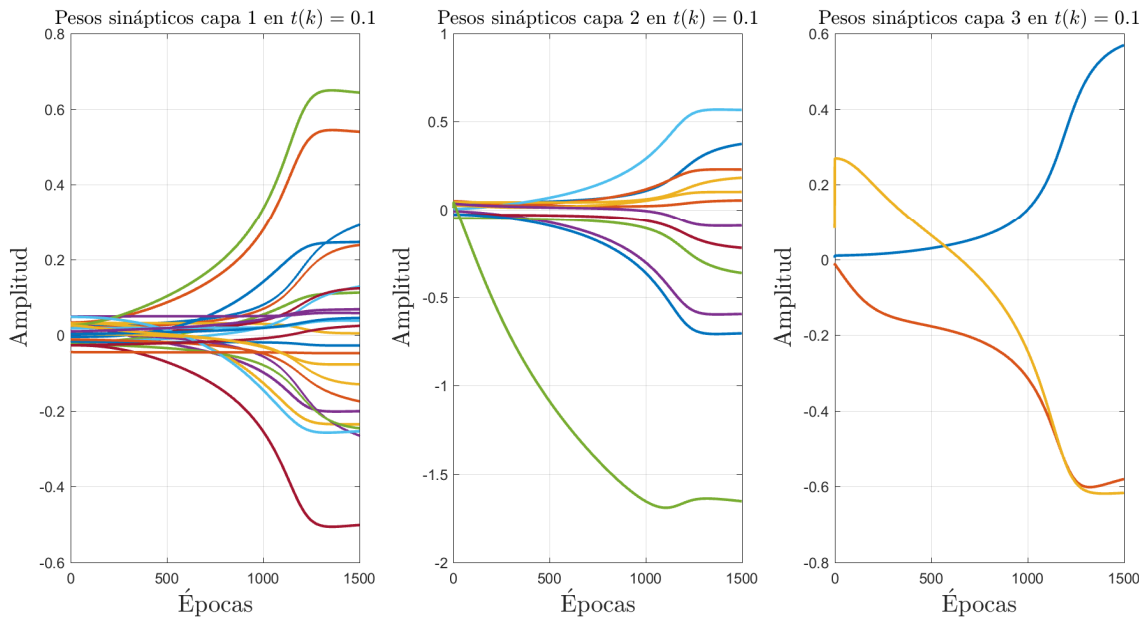


Figura B.12. Adaptación de pesos sinápticos por épocas de la identificación del porcentaje de metano.

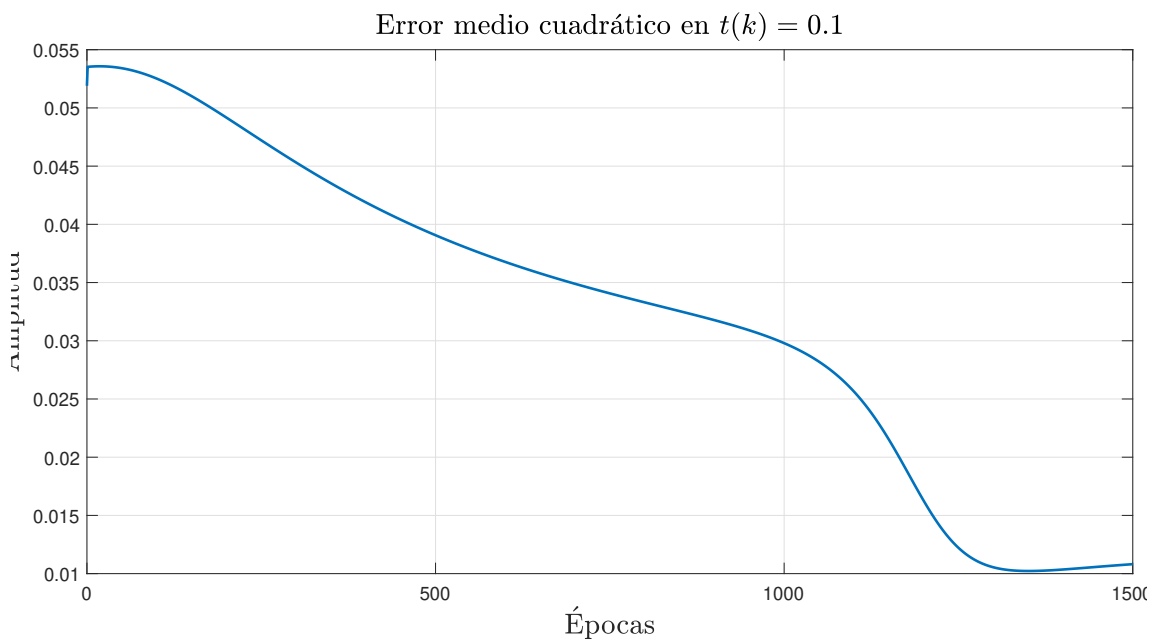


Figura B.13. EMC del porcentaje de metano.

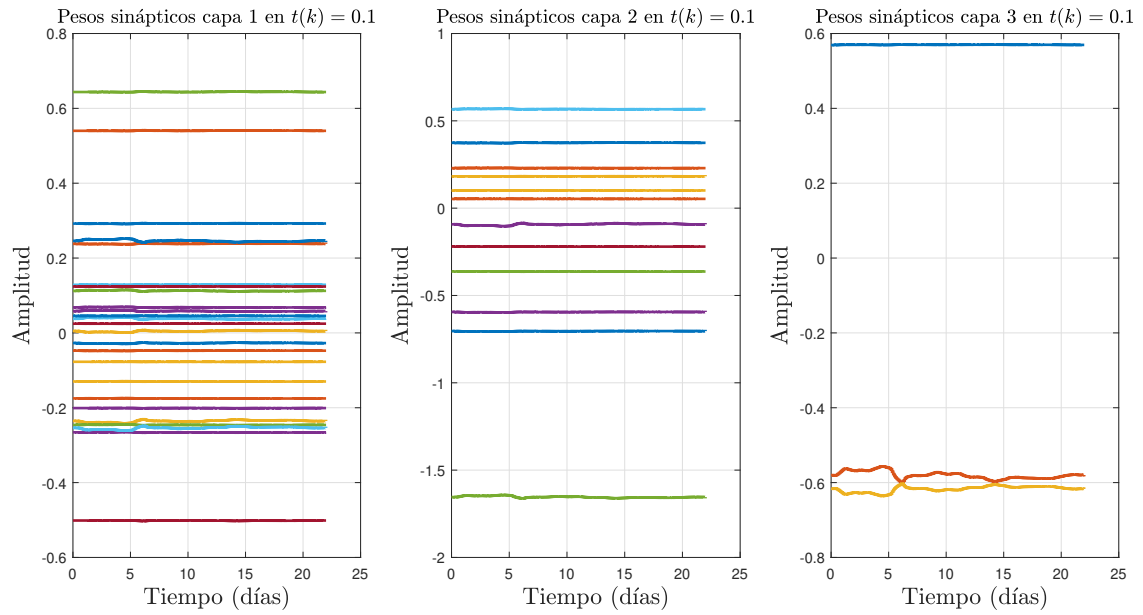


Figura B.14. Pesos sinápticos para modelo neuronal del porcentaje de metano.

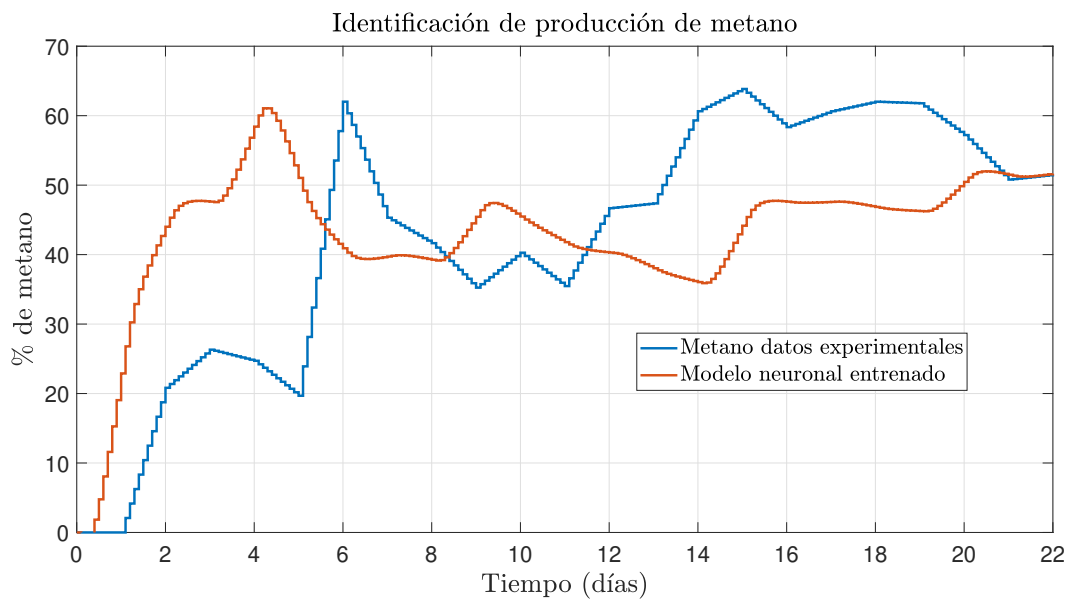


Figura B.15. Modelo entrenado de metano.

# Anexo C

---

## Control neuronal

---

### C.1. Objetivo de control: seguimiento de trayectoria

En este escenario se plantea una situación ideal donde usando el modelo neuronal identificado en esquema SISO se desea llegar a la siguiente referencia deseada 1  $DQO_s$   $g/L$  y después siga un cambio de referencia. Al igual que en las simulaciones del capítulo 4 se busca que la respuesta del sistema con el controlador sea *lenta*.

El objetivo de esta prueba es la de comprobar que el controlador neuronal también es capaz de controlar el sistema y manejar un seguimiento de trayectoria a la vez. Las condiciones del controlador neuronal son las siguientes:

- $T_0 = 0.1$  que representa 2 horas y media de los datos interpolados.
- $\eta = 0.1$
- Se usan 5 neuronas en la capa 1 para  $J_{une}$ ,  $r(k+1)$  y  $J_{yn}$
- Usa 1 retardo,  $ret_c = 1$ .
- Pesos sinápticos iniciales son:
  - $Z_{ca} = 0.1406$ ,      ▪  $Z_{cb} = -0.1521$ ,      ▪  $Z_c = 0.2462$ ,      ▪  $Z_o = 0.8023$ .
- $V_{CA} = [-0.0477, -0.0278, -0.0289, -0.0017, -0.2401, 0.0770]$ ,
- $V_{CB} = [0.3399, 0.2216, -0.0301, -0.2435, 0.1906, 0.7996]$ ,
- $V_C = [0.0788, -0.0563, 0.3247, -0.0113, 0.3254, 0.0606]$ ,
- $W_{ca1} = [0.0609, 0.3491]$       ▪  $W_{ca3} = [-0.1430, 0.0372]$ ,      ▪  $W_{ca5} = [0.1151, -0.0447]$ ,
- $W_{ca2} = [-0.4488, -0.4472]$ ,      ▪  $W_{ca4} = [-0.1436, -0.3257]$ ,
- $W_{cb1} = [0.3951, 0.2839]$ ,      ▪  $W_{cb3} = [-0.1457, 0.3235]$ ,      ▪  $W_{cb5} = [-0.3749, -0.1982]$ ,
- $W_{cb2} = [-0.4295, -0.0683]$ ,      ▪  $W_{cb4} = [-0.1041, 0.1761]$ ,
- $W_{c1} = [0.3087, 0.3589]$ ,      ▪  $W_{c3} = [0.3836, -0.1814]$ ,      ▪  $W_{c5} = [-0.0237, 0.4047]$ ,
- $W_{c2} = [-0.3713, -0.2153]$ ,      ▪  $W_{c4} = [0.0070, 0.3210]$ ,

En la Figura C.1 se visualiza el comportamiento del modelo neuronal utilizando el controlador tipo neuronal, se observa que el modelo no presenta cambios en los primeros dos días ya que no se ha activado el controlador. Cuando el controlador se activa, pasando dos días de operación, se observa que el modelo presenta un ajuste suave para llegar a la referencia deseada hasta el día 7 tomándole un tiempo de 5 días, después se realiza un cambio de referencia hasta un valor de  $3 \text{ gDQO}_s/L$  hasta el doceavo día de operación cambiando a una referencia de  $1.25 \text{ gDQO}_s/L$ , después de 17 días de operación se cambia la referencia a  $2.15 \text{ gDQO}_s/L$ .

Se observa como el modelo neuronal trata de seguir la referencia establecida, debido a la configuración diseñada del controlador con el objetivo de cumplir un comportamiento lento y suave, se puede decir, que el modelo no llega en tiempo a las referencias deseadas, sin embargo, si se considera un  $\eta$  más grande el modelo neuronal será capaz de llegar a las referencias en menor tiempo, pero esto provocaría el no cumplir el objetivo de tener un comportamiento lento y suave para procesos anaerobios.

En la Figura C.2 se presenta el comportamiento que tienen los pesos sinápticos del controlador neuronal, se observa como los pesos sinápticos se van ajustando de una manera suave viéndose reflejado en la señal del controlador neuronal en la Figura C.3. Este reflejo también se visualiza en el flujo y la CVA controlada del RAH en la Figura C.4.

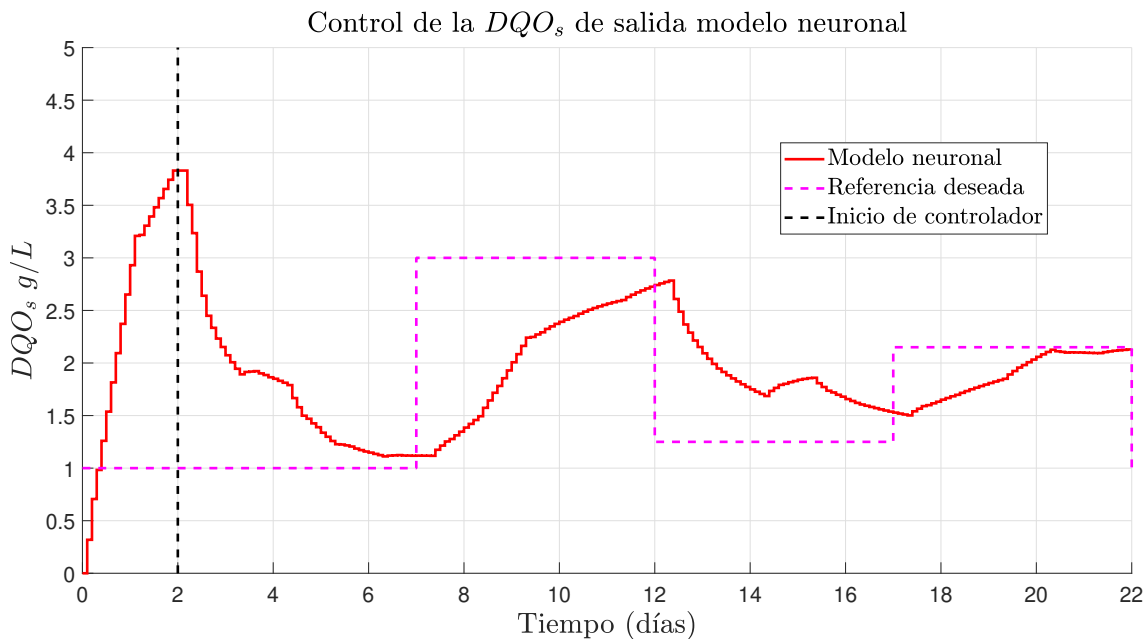


Figura C.1. Modelo neuronal con seguimiento de trayectoria (escenario ideal).

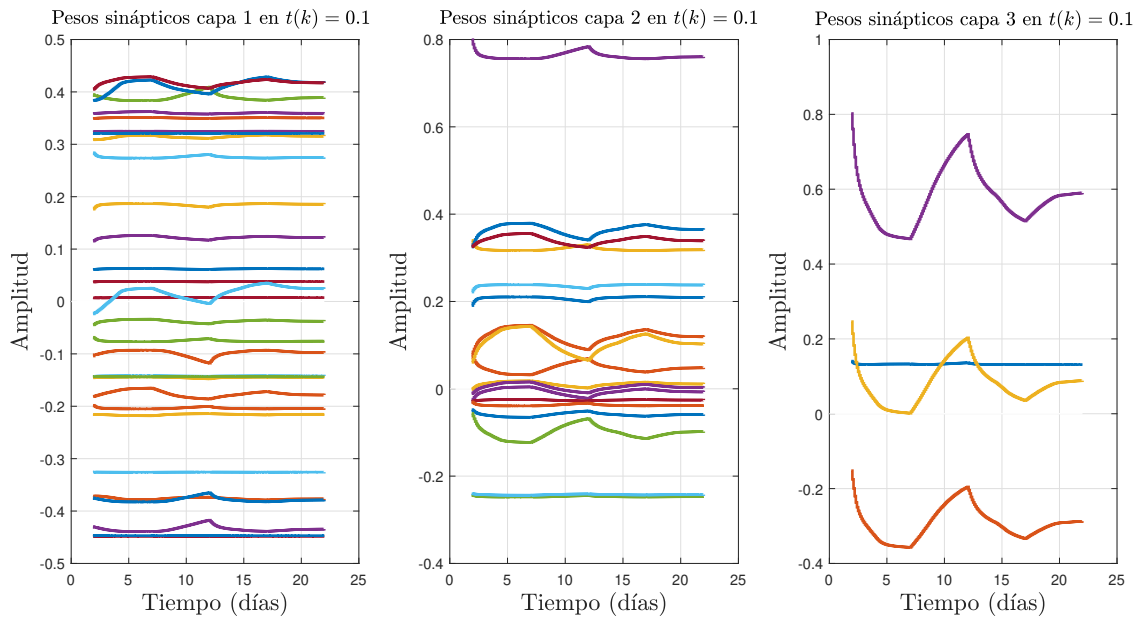


Figura C.2. Pesos sinápticos del controlador neuronal para seguimiento de trayectoria.

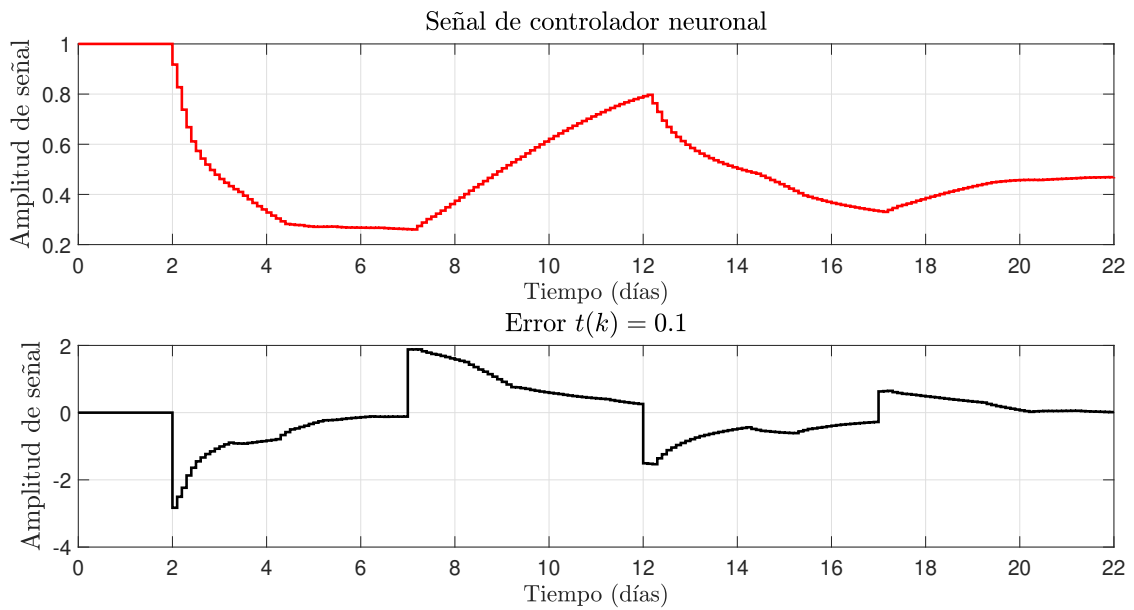


Figura C.3. Comportamiento de la señal del controlador neuronal y el error generado en el seguimiento de trayectoria.



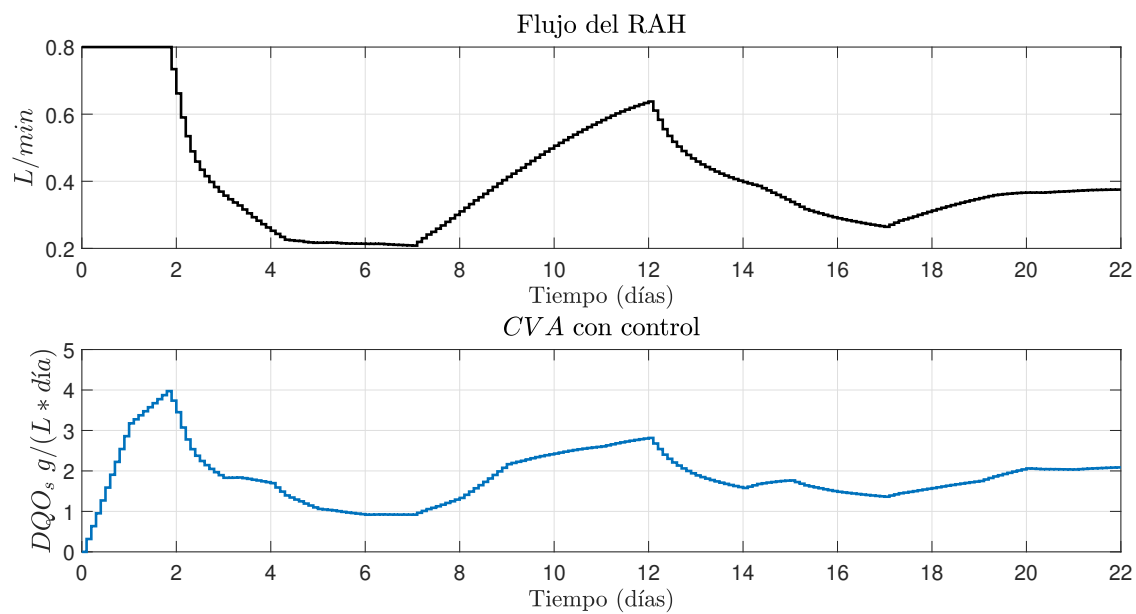


Figura C.4. Comportamiento del flujo del reactor y CVA con el controlador para seguimiento de trayectoria.

## C.2. Comportamiento del control neuronal ante perturbaciones

Usando el modelo neuronal identificado en esquema SISO se desea llegar a la siguiente referencia deseada  $0.45 \text{ gDQO}_s/L$  y después de 10 días de operación el sistema tendrá una perturbación tipo escalón en el flujo con un valor de  $0.2 \text{ L/min}$  en el escenario ideal. El objetivo de esta prueba es el de corroborar cómo se comporta el controlador neuronal ante perturbaciones y si es capaz de controlar el sistema. Las condiciones del controlador neuronal son las mismas utilizada en los resultados del controlador en el capítulo 4.

En la Figura C.5 se visualiza un comparativa del comportamiento del modelo neuronal entrenado con y sin perturbación utilizando el controlador tipo neuronal, se observa que al pasar 10 días de operación se presenta una perturbación en el sistema y el controlador trata de ajustar el cambio generado en un día, sin embargo, se visualiza que el modelo neuronal no puede reducir su valor hacia la referencia en comparación al modelo sin perturbación.

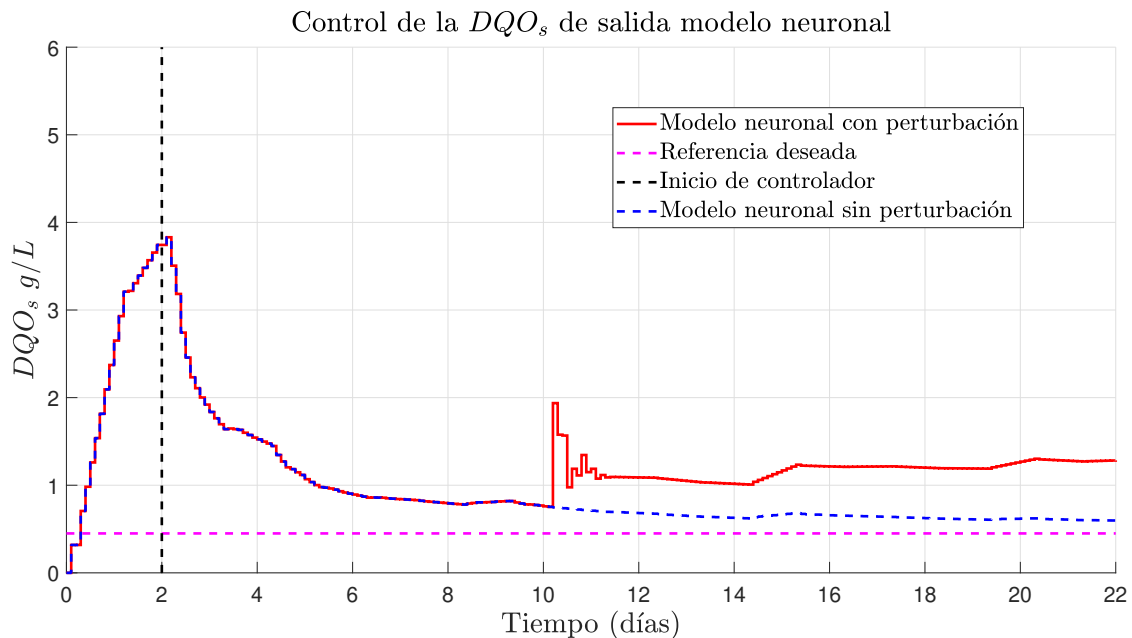


Figura C.5. Modelo neuronal con perturbación (escenario ideal).

En la Figura C.6 se presenta el comportamiento que tiene el controlador neuronal cuando se presenta la perturbación después de 10 días de operación, donde el controlador trata de ajustar dicho cambio, sin embargo, la señal del controlador se limita volviéndose cero viéndose reflejado en el error donde se mantiene en un valor cercano a  $-1$ . Esto también se refleja en el flujo y la CVA del RAH de la Figura C.7.

Suponiendo un caso especial, donde se considera que la referencia deseada fuera  $1.5 \text{ gDQO}_s/L$  y la perturbación fuera una pérdida de flujo (fuga) con un valor de  $0.2 \text{ L/min}$  como se muestra en la Figura C.8, se observa en el momento que entra la perturbación, pasando 10 días de operación, como el modelo neuronal logra llegar a la referencia en el día 20 de operación tomándole 10 días para estabilizarse.

Si se comparan estas dos pruebas, en el caso de la Figura C.5, se puede decir que el controlador neuronal no podrá ajustar el modelo debido al dimensionamiento que tiene la electroválvula porque

la señal que envía el controlador se mantiene operando en un flujo muy pequeño (una apertura de electroválvula pequeña) lo cual al presentarse la perturbación el controlador manda un valor de señal negativo provocando el cierre de la electroválvula y por eso no se visualiza un ajuste alguno en el modelo.

En el caso de la Figura C.8, al tener una referencia más grande también se tiene un mayor rango de operación de la electroválvula, por lo cual vemos como el controlador si puede ajustar el modelo hasta llevarlo a la referencia establecida.

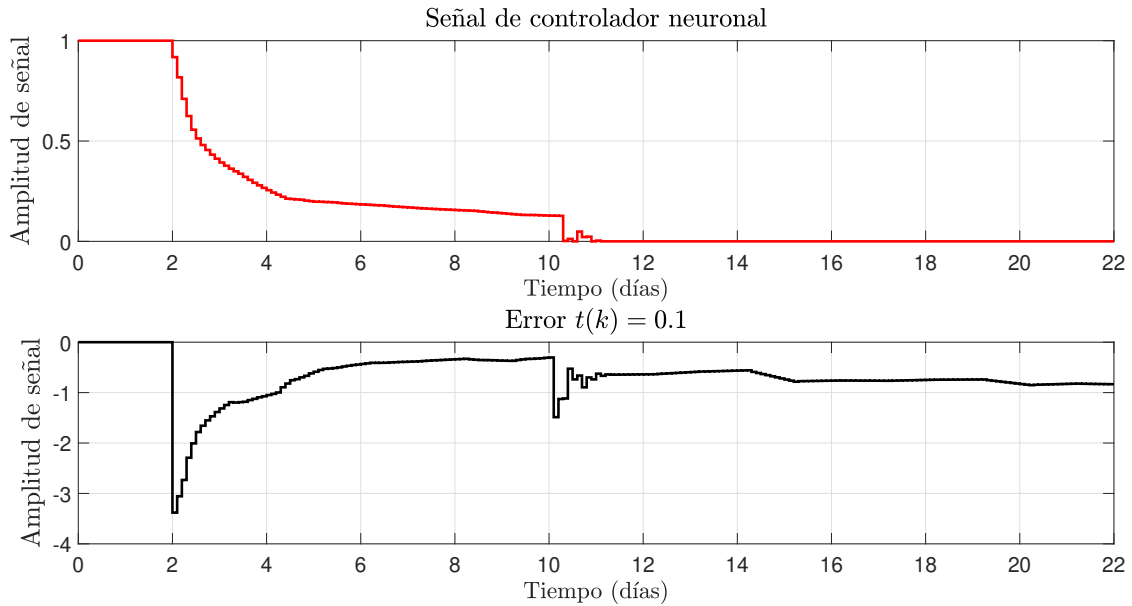


Figura C.6. Comportamiento de la señal del controlador neuronal y el error generado con referencia  $0.45 \text{ gDQO}_s/L$  con perturbación (escenario ideal).

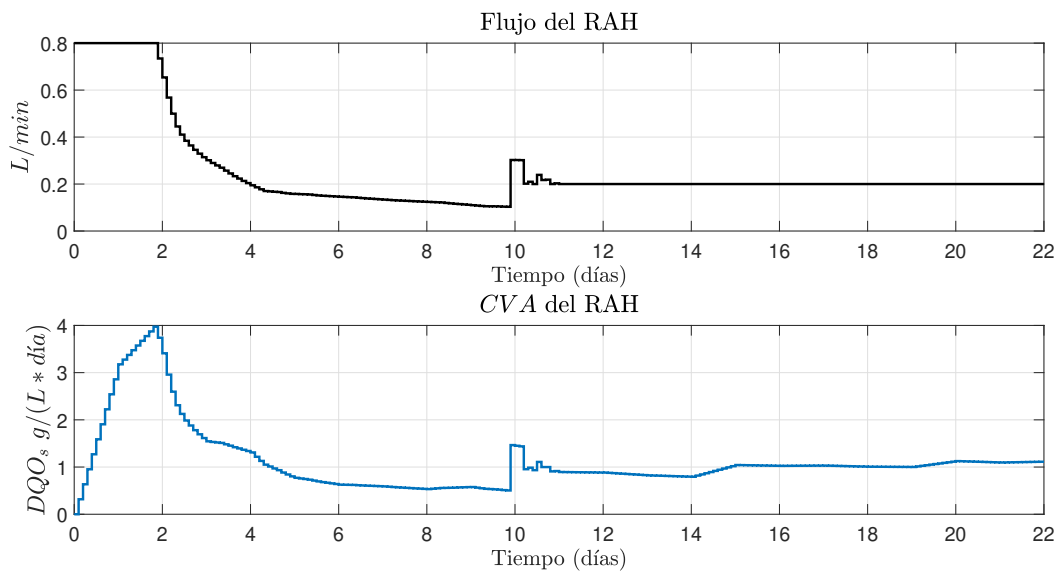


Figura C.7. Comportamiento del flujo del reactor y CVA con el controlador con perturbación (escenario ideal).

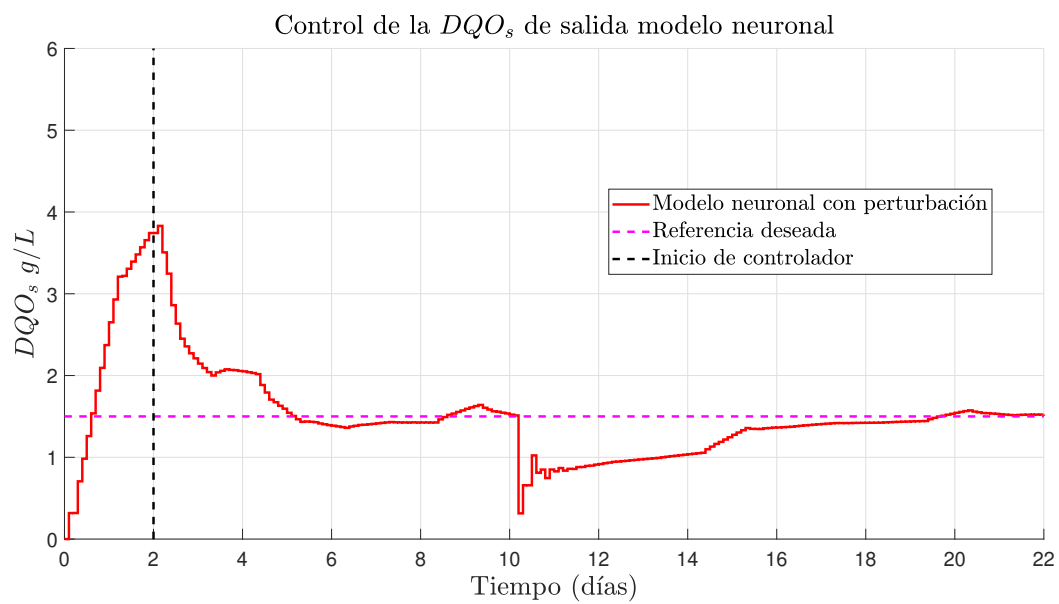


Figura C.8. Modelo neuronal con perturbación (escenario ideal) caso especial.