

CIATEQ, A. C. Centro de Tecnología Avanzada
Dirección de Posgrado



Control y monitoreo cognitivo basado en máquina de aprendizaje extremo: aplicación a la caracterización y formado de lámina bimetálica para moto-protectores térmicos

TESIS QUE PRESENTA

MC. José Martín Melchor Leal
Asesor: Dr. José Antonio Cantoral Ceballos

Para obtener el grado de

Doctor en
Manufactura Avanzada

Aguascalientes, Ags., México
febrero, 2021

CARTA DE LIBERACIÓN DEL ASESOR




Querétaro, Querétaro, 31 de julio del 2020

Mtro. Geovany González Carlos
Coordinador Académico de Posgrado
CIATEQ, A.C.

Los abajo firmantes, miembros del Comité Tutorial del MC. José Martín Melchor Leal, una vez revisado su Proyecto Terminal de tesis, titulado “Control y Monitoreo Cognitivo Basado en Máquina de Aprendizaje Extremo: Aplicación a la Caracterización y Formado de Lámina Bimetálica para Moto-Protectores Térmicos” **autorizo** que el citado trabajo sea presentado por el alumno para su revisión, con el fin de alcanzar el grado de **Doctor en Manufactura Avanzada**.

Sin otro particular por el momento, agradezco la atención prestada.


Dr. José Antonio Cantoral Ceballos

Grado, nombre completo
Asesor Académico

CARTA DE LIBERACIÓN DEL REVISOR

Querétaro, Qro., 26 de enero del 2021.

Dra. María Guadalupe Navarro Rojero
Directora de Posgrado
CIATEQ, A.C.

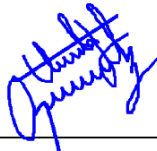
Por medio de la presente me dirijo a usted en calidad de Revisor del proyecto terminal del alumno **JOSÉ MARTÍN MELCHOR LEAL**, cuyo título es:

**"CONTROL Y MONITOREO COGNITIVO BASADO EN MÁQUINA DE APRENDIZAJE
EXTREMO: APLICACIÓN A LA CARACTERIZACIÓN Y FORMADO DE LÁMINA
BIMETÁLICA PARA MOTO-PROTECTORES TÉRMICOS"**

Después de haberlo leído, corregido e intercambiado información con el alumno, y realizado los cambios que le fueron sugeridos, puede ser autorizada su impresión, a fin de que se inicien los trámites correspondientes para su defensa.

Sin otro particular por el momento, y en espera de que mis sugerencias sean tomadas en cuenta en beneficio del estudiante y la Institución, agradezco la atención prestada.

Atentamente,



Dr. Óscar Hernández Uribe

CARTA DE LIBERACIÓN DEL REVISOR



Zapopan, Jalisco, 11 de enero del 2021.

Dra. María Guadalupe Navarro Rojero
Directora de Posgrado
CIATEQ, A.C.

Por medio de la presente me dirijo a usted en calidad de Revisor del proyecto terminal del alumno MC. José Martín Melchor Leal, cuyo título es:

"Control y Monitoreo Cognitivo Basado en Máquina de Aprendizaje Extremo: Aplicación a la Caracterización y Formado de Lámina Bimetálica para Moto- protectores Térmicos"

Después de haberlo leído, corregido e intercambiado información con el (la) alumno(a), y realizado los cambios que le fueron sugeridos, puede ser autorizada su impresión, a fin de que se inicien los trámites correspondientes para su defensa.

Sin otro particular por el momento, y en espera de que mis sugerencias sean tomadas en cuenta en beneficio del estudiante y la Institución, agradezco la atención prestada.

Atentamente,

Firma

Dr. Caín Pérez Wences

DEDICATORIA

... A mi esposa Ernestina y a mis hijas Ana Patricia e Irma Adriana.

AGRADECIMIENTOS

Primero con estas líneas me gustaría agradecer a mi esposa Ernestina que me ha esperado y ayudado pacientemente durante todo el desarrollo de esta tesis, también a mis hijas Ana Patricia e Irma Adriana por apoyo para continuar con esta investigación.

Me gustaría también dar las gracias a mi asesor, Dr. José Antonio Cantoral Ceballos. Sin ninguna duda su aportación ha sido de vital importancia para la realización de esta tesis. Desde el comienzo ha depositado una gran confianza en mí y me ha demostrado su apoyo.

También quiero agradecer a mis compañeros del área de ingeniería en Sensata Technologies de México por el gran apoyo y soporte.

* * *

RESUMEN

En los sistemas de control de automatismos Industriales para manufactura existe la necesidad cada vez más frecuente de usar algoritmos avanzados de control que utilicen las técnicas de Machine Learning. Esto debido a los requerimientos de crear nuevos productos de mayor calidad y complejidad con procesos de alto rendimiento. De aquí nace la necesidad de desarrollar un nuevo algoritmo para optimizar los controles de automatismos industriales capaz de aprender, memorizar las mejores soluciones y adaptarse a los cambios en un ambiente industrial real afectado por un sin número de ruidos y variaciones de materias primas.

Antes de este trabajo no se había conseguido relacionar los sistemas de control de automatismos industriales con el desarrollo vertiginoso de los sistemas de control robótico cognitivo que usan Machine Learning avanzado, debido principalmente a la velocidad de aprendizaje, esta barrera se salva al crear algoritmos híbridos que utilizan la técnica de máquina de aprendizaje extremo (ELM).

El sistema robótico cognitivo tiene un enorme parecido al control de automatismos industriales y ha logrado incorporar las funciones de memoria, reconocimiento automático y resolución de obstáculos; funciones que se pueden adaptar a los procesos industriales.

El presente trabajo, explica la creación dos algoritmos híbridos, el AHN-ELM y el PAT-ELM ambos inspirados en el funcionamiento del hipocampo biológico y su implementación basada en ELM. El algoritmo híbrido PAT-ELM se aplica a la caracterización de láminas bimetálicas, y el AHN-ELM en el control de una máquina industrial para el formado de curvatura de lámina bimetálica termostática, ambos usados en la fabricación de moto-protectores eléctricos. Se demuestra que los algoritmos PAT-ELM y AHN-ELM propuestos pueden caracterizar y controlar de forma eficiente procesos de manufactura industrial reales que los sistemas de control tradicionales basados en modelos no logran con el mismo nivel de eficiencia.

Palabras clave: Redes neuronales, Máquina de aprendizaje extremo, Red hipocámpica, Bimetales termostáticos.

ABSTRACT

In the industrial automation control systems for manufacturing there is an increasingly frequent need to use advanced control algorithms that use Machine Learning techniques. This is due to the requirements of creating new products of higher quality and complexity with high-performance processes. Hence the need to develop a new algorithm to optimize industrial automation controls capable of learning, memorizing the best solutions and adapting to changes in a real industrial environment affected by a number of noises and variations in raw materials.

Before this work, it had not been possible to relate industrial automation control systems with the vertiginous development of cognitive robotic control systems that use advanced Machine Learning, mainly due to the speed of learning, this barrier is overcome by creating hybrid algorithms that use the extreme learning machine (ELM) technique.

The cognitive robotic system has an enormous resemblance to the control of industrial automation and has managed to incorporate the functions of memory, automatic recognition and resolution of obstacles; functions that can be adapted to industrial processes.

The present work explains the creation of two hybrid algorithms, the AHN-ELM and the PAT-ELM, both inspired by the functioning of the biological hippocampus and their implementation based on ELM. The hybrid algorithm PAT-ELM is applied to the characterization of bimetallic thermostatic discs, and the AHN-ELM in the control of an industrial machine for forming curvature of thermostatic bimetallic discs, both used in the manufacture of electrical motor protectors. The proposed PAT-ELM and AHN-ELM algorithms are shown to be able to efficiently characterize and control real industrial manufacturing processes that traditional model-based control systems cannot achieve with the same level of efficiency.

Keywords: Neural network, Extreme learning machine, Hipocampical network, Thermostatic bimetals.

ÍNDICE DE CONTENIDO

RESUMEN.....	vi
ABSTRACT	vii
ÍNDICE DE CONTENIDO.....	viii
ÍNDICE DE FIGURAS	xiii
ÍNDICE DE TABLAS.....	xvii
GLOSARIO	xviii
1. INTRODUCCIÓN A LA INVESTIGACIÓN.....	1
1.1. MOTIVACION DE LA INVESTIGACIÓN	5
1.2. DEFINICIÓN DEL PROBLEMA	6
1.3. PREGUNTA DE INVESTIGACIÓN.....	6
1.4. OBJETIVOS	7
1.4.1. Objetivo general.....	7
1.4.2. Objetivos específicos	7
1.5. HIPÓTESIS.....	8
1.6 DESCRIPCIÓN GENERAL DE LA TESIS	8
1.7 SOLUCIÓN PROPUESTA	13
2. MARCO TEÓRICO.....	15
2.1 SISTEMAS AVANZADOS DE CONTROL AUTOMÁTICO PARA PROCESOS DE MANUFACTURA.....	15
2.2 CONTROL BASADO EN MODELOS.....	16
2.2.1. Modelo de control predictivo	18
2.2.2 Control basado en modelo de sistema en red	20
2.2.3 Control multivariable.....	22
2.3 CONTROL ADAPTATIVO	23
2.4 CONTROL BASADO EN INTELIGENCIA COMPUTACIONAL	24
2.4.1 Sistemas expertos de control.....	26
2.4.2 Control difuso	27
2.4.3 Redes neuronales artificiales (RNA)	29
2.4.4 Algoritmos evolutivos	31
2.4.5 Algoritmos extreme learning machine (ELM)	31
2.4.6 Aprendizaje por refuerzos.....	32
2.4.7 Técnicas integradas de inteligencia computacional.....	35

2.5 SISTEMA COGNITIVO BASADO EN RED NEURONAL.....	37
2.6 SISTEMA DE CONTROL ROBÓTICO COGNITIVO.....	38
2.7 APRENDIZAJE AUTOMÁTICO COGNITIVO	39
2.8 RED HIPOCÁMPICA BIOLÓGICA COMO CONTROL COGNITIVO	40
2.9 CONCLUSIONES DE LA REVISIÓN LITERARIA	42
3. PROCEDIMIENTO	45
3.1 DESARROLLO DEL ALGORITMO HÍBRIDO AHN-ELM PARA CONTROL DE PROCESOS AUTOMÁTICOS EN MANUFACTURA	45
3.1.1 Introducción a desarrollo del nuevo algoritmo híbrido AHN-ELM.....	45
3.1.2 Modelo estructural de la memoria	46
3.1.3. Instancias hipocámpicas	48
3.1.4 Algoritmo y arquitectura de red neuronal híbrida AHN-ELM	53
3.1.5 Diagrama de flujo simplificado de red AHN-ELM	56
3.1.6 Algoritmo híbrido simplificado de red AHN-ELM.....	59
3.1.7 Estructura de conjuntos de datos globales y locales de la red hipocámpica AHN-ELM	60
3.1.8 Eliminación de los datos anormales en la instancia hipocámpica DGa ...	61
3.1.9 Separación patrones similares en instancia DGb	69
3.1.10 Asociando patrones episódicos en instancia CA3	73
3.1.11 Completar patrones episódicos con instancia hipocámpica CA1	80
3.1.12 Interacción y aprendizaje del medio ambiente con instancia SUB	88
3.1.13 Conclusión algoritmo AHN-ELM.....	97
3.2 DESARROLLO DE ALGORITMO HÍBRIDO PAT-ELM PARA LA CARACTERIZACIÓN DE PROCESOS	97
3.2.1 Introducción	97
3.2.2 Modificación de la instancia DG para caracterización	99
3.2.3 Arquitectura de PAT-ELM.....	101
3.2.4 Conclusiones algoritmo híbrido PAT-ELM	104
3.3 EXPERIMENTOS VIRTUALES Y DE LABORATORIO AHN-ELM.....	105
3.3.1. Introducción a experimentos virtuales y de laboratorio.....	105
3.3.2. Experimento con laberinto virtual	105
3.3.3. Propósito del experimento AHN-ELM con laberinto virtual	106
3.3.4. Procedimiento para experimentación con laberinto virtual	106
3.3.5. Implementación AHN-ELM para solución de laberinto virtual.....	107

3.3.6. Arquitectura para AHN-ELM para prueba de laberinto.....	110
3.3.7. Resultados del experimento	114
3.3.8. Conclusiones de experimento virtual.....	115
3.4 ROBOT SEGUIDOR DE LÍNEA USANDO AHN-ELM	116
3.4.1 Experimento de robot explorador: procedimiento	117
3.4.2. Implementación experimento robot explorador con red neuronal AHN-ELM.....	118
3.4.3. Sistema cortical e interfaz con MatLab	118
3.4.4. Desarrollo de sistema de control y comunicación vía bluetooth.....	119
3.4.5. Procesamiento de datos e interfaz con MatLab.....	122
3.4.6. Arquitectura para AHN-ELM para robot explorador.....	123
3.4.7. Experimento con el robot explorador en laberinto marcado en piso ...	126
3.4.8. Resultados del experimento de laboratorio con robot explorador usando red neuronal AHN-ELM.....	128
3.4.9. Conclusiones de prueba de laboratorio	129
4. RESULTADOS	130
4.1 APLICACIÓN DE ALGORITMO HÍBRIDO PAT-ELM A LA CARACTERIZACIÓN DE BIMETALES	130
4.1.1 Introducción.....	130
4.1.2 Caracterización de bimetálicos termostáticos para moto-protectores eléctricos.....	130
4.1.3 Acotando el experimento.....	132
4.1.4. Fundamentos del moto-protector térmico a base de bimetálico termostático	132
4.1.5. Estructura del moto-protector	133
4.1.6. Cálculo de curvatura y fuerza en bimetálico.....	134
4.1.7. Caracterización del bimetálico termostático	136
4.1.8. Caracterización del perfil de fuerza-desplazamiento a T_c	138
4.1.9. Algoritmo híbrido PAT-ELM para caracterización del bimetálico.....	139
4.1.10 Arquitectura PAT-ELM para caracterización de bimetálicos.....	139
4.1.11. Dispositivo medición gradiente de desplazamiento a T_c	142
4.1.12. Salida pronosticada por algoritmo híbrido PAT-ELM.....	144
4.1.13. Indicadores de desempeño de la red PAT-ELM	146
4.1.14. Análisis desempeño algoritmo híbrido PAT-ELM	148

4.1.15. Resultados del monitoreo cognitivo	149
4.1.16. Índices estadísticos.....	150
4.1.17. Conclusiones preliminares algoritmo PAT-ELM	152
4.2. APLICANDO EL ALGORITMO HÍBRIDO AHN-ELM AL CONTROL AUTOMÁTICO DEL FORMADO DE BIMETALES	153
4.2.1. Introducción al AHM para formado de bimetales.....	153
4.2.2. Antecedentes	153
4.2.3. Propósito del experimento	156
4.2.4. Procedimiento experimental	157
4.2.5. Criterio de aceptación de pruebas	159
4.2.6. Selección y adaptación de máquina prototipo para experimento	159
4.2.7. Desarrollo del sistema de control de prototipo	163
4.2.8. Modelo conceptual de la red neuronal AHN-ELM a la máquina prototipo	166
4.2.9. Configuración de red AHN-ELM para sistema de control de máquina prototipo	168
4.2.10. Arquitectura para AHN-ELM para control de formado de curvatura de bimetales termostáticos.....	169
4.2.11. Tamaño de muestra de experimentación	173
4.2.12. Implementación de delimitador de posición	176
4.2.13. Resultados de desempeño del algoritmo híbrido de red neuronal AHN-ELM.....	178
4.2.14. Conclusiones de la aplicación del algoritmo híbrido AHN-ELM como control automático en el proceso de formado de bimetales	186
CONCLUSIONES.....	187
CONCLUSIONES GENERALES	187
CONCLUSIONES DEL EXPERIMENTO 1 PAT-ELM CARACTERIZADOR DE TIRA BIMETÁLICA.....	188
CONCLUSIONES DEL EXPERIMENTO 2 AHN-ELM EN EL CONTROL DEL PROCESO DE FORMADO	188
RECOMENDACIONES.....	190
RECOMENDACIONES Y PROYECCIÓN FUTURA	190
APORTACIÓN DE LA TESIS.....	192
APORTACIÓN SOCIAL DE LA TESIS	193
REFERENCIAS.....	194

ANEXOS.....205

ÍNDICE DE FIGURAS

Figura 1. Mapa mental con la distribución de los capítulos que conforman la tesis.	9
Figura 2. Esquema de la red híbrida AHN-ELM.	14
Figura 3. Esquema de control de un MBC.	18
Figura 4. Esquema del sistema de aprendizaje por refuerzos.	34
Figura 5. Modelo hipocámpico de aprendizaje con memoria de corto y largo plazo.	46
Figura 6. Diagrama a bloques del modelo de la memoria de Atkinson y Shiffrin.	47
Figura 7. Esquema de las rutas neuronales que existen dentro de la formación hipocámpica biológica.	49
Figura 8. Diagrama a bloques del modelo simplificado del hipocampo.	51
Figura 9. Esquema de una neurona del tipo <i>Single Layer Feed-Forward Network</i> o SLFN: red monocapa de avance hacia adelante.	54
Figura 10. Diagrama de flujo simplificado del algoritmo híbrido AHN-ELM.	57
Figura 11. Arquitectura de la red híbrida AHN-ELM.	58
Figura 12. Diagrama de flujo del algoritmo de la instancia dentate gyrus (<i>DGa</i>).	62
Figura 13. Diagrama a bloques del algoritmo para eliminar los datos anormales, basado en método <i>part average testing</i> (PAT).	63
Figura 14. Gráfica de salida del algoritmo.	66
Figura 15. Gráfica de datos y límites dinámicos usando <i>part average testing</i> .	67
Figura 16. Diagrama de flujo del algoritmo base de la instancia hipocámpica dentate gyrus (<i>DGb</i>).	69
Figura 17. Modelo de la red neuronal de la instancia hipocámpica <i>DGb</i> .	71
Figura 18. Vectores en formato 32 x 32 después de ser transformados en la instancia hipocámpica <i>DGb</i> .	72
Figura 19. Diagrama de flujo del algoritmo base de la instancia hipocámpica cornu ammonis 3 (<i>CA3</i>).	74
Figura 20. Modelo de red neuronal de la instancia hipocámpica <i>CA3</i> .	77
Figura 21. Muestra los vectores en formato 32 x 32, que representan los procesos de recuperación de 6 patrones aprendidos, como memorias fundamentales.	78

Figura 22. Diagrama de flujo de la instancia hipocámpica CA1, basada en máquina de aprendizaje extremo secuencial.....83

Figura 23. Modelo de red neuronal de la instancia hipocámpica CA1.....84

Figura 24. Diagrama de flujo de los 2 tipos de entrenamiento que ocurren en instancia CA1.85

Figura 25. Modelo de descarte de la instancia hipocámpica CA1.....87

Figura 26. Diagrama de flujo del algoritmo de la instancia hipocámpica SUB. 90

Figura 27. Modelo de aprendizaje de la instancia hipocámpica subiculum.93

Figura 28. Gráfica de tendencia del algoritmo instancia hipocámpica SUB. 95

Figura 29. Tres ejemplos de vectores de solución provenientes de la instancia hipocámpica subiculum (SUB).95

Figura 30. Ejemplo de la variación de la característica térmica de varios lotes de producción.99

Figura 31. Diagrama de flujo la red neuronal PAT-ELM para el monitoreo cognitivo. 100

Figura 32. Arquitectura de algoritmo híbrido PAT-ELM. 101

Figura 33. Laberinto virtual empleado para realizar una primera prueba al algoritmo híbrido AHN-ELM..... 108

Figura 34. Diferentes geometrías utilizadas en el experimento del laberinto, aplicando la red neuronal AHN-ELM propuesta 109

Figura 35. Arquitectura de la red híbrida AHN-ELM para para experimento de laberinto. 111

Figura 36. Arquitectura del algoritmo híbrido AHN-ELM para experimento de laberinto. 113

Figura 37. Gráfica de desempeño en las 5 diferentes geometrías del laberinto resuelto por el algoritmo híbrido AHN-ELM..... 114

Figura 38. Imagen del robot prototipo explorador utilizado para desarrollar las interfaces de comunicación..... 117

Figura 39. Esquema del sistema de control del robot explorador y la interfaz con el sistema cortical..... 119

Figura 40. En la imagen se puede observar la implementación del controlador embebido arduino mega como sistema de control..... 120

Figura 41 La sub figura (a) muestra al robot explorador y la sub figura (b) muestra la interface bluetooth utilizado en el sistema cortical.	121
Figura 42. Interfaz de módulo bluetooth HC-05 para comunicación.....	122
Figura 43. Arquitectura de la red Híbrida AHN-ELM para experimento del robot explorador.....	124
Figura 44. Arquitectura del algoritmo híbrido AHN-ELM para experimento del robot explorador.....	125
Figura 45. Esquema de ruta a seguir por robot explorador entre el punto de origen y el objetivo.....	126
Figura 46. Robot explorador siguiendo la línea negra del laberinto durante el experimento.....	127
Figura 47. Gráfica de resultados del experimento con el robot explorador.	128
Figura 48. Geometría de un bimetalico termostático típico.....	133
Figura 49. La imagen muestra dos condiciones de curvatura de tira bimetalica termostática al ser expuesta a temperaturas diferentes.....	134
Figura 50. Gráfica X-Y que muestra el perfil fuerza-distancia de un bimetalico termostático.....	139
Figura 51. Arquitectura de la red híbrida PAT-ELM para la caracterización de bimetales.....	140
Figura 52. Configuración del algoritmo Híbrido PAT-ELM para la caracterización de bimetales.....	141
Figura 53. Prototipo caracterizador de lámina bimetalica.	143
Figura 54. Perfil de fuerza de pronóstico de la tira termostática.....	144
Figura 55. Perfil dinámico de desplazamiento de lámina bimetalica.	145
Figura 56. Perfil de desplazamiento de varias láminas bimetalicas al exponerlas en el tester.	149
Figura 57. Circuito básico de un moto-protector de lámina bimetalica.	154
Figura 58. Detalle de cambio mecánico de la lámina bimetalica.....	155
Figura 59. Esquema del troquel con herramientas de formado ajustables que dan forma a la curvatura del bimetel.	156
Figura 60. Máquina prototipo, es una troqueladora formadora de bimetales.....	160

Figura 61. Detalle de las herramientas de formado del bimetál en la máquina prototipo..... 161

Figura 62. Imagen de la máquina de prueba de temperatura para la tira bimetálica termostática. 162

Figura 63. Diagrama a bloques de la máquina prototipo de formado de lámina bimetálica..... 164

Figura 64. Imagen de pantalla de la computadora en la máquina formadora... 164

Figura 65. Diagrama de interconexión de la máquina prototipo troqueladora.... 166

Figura 66. Esquema del modelo conceptual de red neuronal para el experimento de control cognitivo usando el algoritmo híbrido AHN-ELM. 167

Figura 67. Arquitectura de la red híbrida AHN-ELM 170

Figura 68. Arquitectura en bloques del algoritmo híbrido AHN-ELM..... 172

Figura 69. Recuadro de resumen estadístico de la muestra para T_c 174

Figura 70. Recuadro de resumen estadístico de la muestra para T_f 175

Figura 71. Gráfica de potencia del tamaño de muestra mínimo..... 176

Figura 72. Gráfica de porcentaje de error del algoritmo con respecto al tiempo. 177

Figura 73. Gráfica de tendencia yield operacional para 9 lotes de prueba aplicando la red AHN-ELM. 179

Figura 74. Gráfica de intervalos de confianza que muestra la comparación entre el proceso original y el proceso en el que se aplica la red AHN-ELM. 180

Figura 75. El recuadro muestra un resumen de factibilidad de los datos para ejecutar la prueba T de 2 muestras. 182

Figura 76. Gráfica del análisis de normalidad Ryan-Joiner basado en correlaciones..... 183

Figura 77. Recuadro de gráfica de tendencia muestra el yield de salida de la máquina prototipo sin y con el nuevo algoritmo híbrido AHN-ELM. 184

Figura 78. Recuadro de intervalos de confianza de la prueba t de 2 muestras.... 185

ÍNDICE DE TABLAS

Tabla 1. Descripción de variables globales usadas en algoritmo AHN-ELM.....	60
Tabla 2. Descripción de variables locales usadas en algoritmo AHN-ELM.	61
Tabla 3 Parámetros de configuración AHN-ELM para experimento virtual del laberinto.	110
Tabla 4. Características de bimetal termostático para aplicaciones en moto-protectores térmicos.	137
Tabla 5. Análisis de correlación entre el modelo y las mediciones de varios.....	150
Tabla 6. Indicadores del desempeño del algoritmo PAT-ELM.....	151
Tabla 7. Configuración de Instancias hipocámpicas establecidas para aplicar la red AHN-ELM en la máquina prototipo.	169
Tabla 8. Indicadores del desempeño del algoritmo AHN-ELM.....	181

GLOSARIO

ACO: Optimización de Colonias de Hormigas.

AD: Control Adaptativo.

AE: (Auto-Encoder) Auto codificador.

AHA!: Red hipocámpica propuesta por Gideon Kowadlo.

AHN-ELM: Red Hipocámpica Artificial Basada en Máquina de Aprendizaje Extremo.

AI: (*Artificial Intelligence*) Inteligencia Artificial.

CA: Cornu Ammonis.

DES: Control basado en Sistemas de Eventos Discretos.

DG: Dentate Gyrus.

DMP: Control predictivo de modelos.

EC: Entorrinal Cortex.

ECS: Sistemas expertos de control.

ELM: Máquina de Aprendizaje Extremo.

ELM-AE: Auto-Encoder Basado en Máquina de Aprendizaje Extremo.

FMC: Control basado en sistema difuso.

GPC: Controlador predictivo generalizado.

HF: Formación Hipocámpica.

HMSNN: Hippocampus inspired Memory Spiking Neural Network propuesta por Zhang et al.

HOPFIELD: Red Neuronal Recurrente Auto-Asociativa.

LQG: Control lineal cuadrático Gaussiano.

MBC: Control basado en de Modelos.

MDP: Proceso de Decisión de Markov (Markov Decision Processes).

Memoria a corto plazo: Es una acción que establece AHN-ELM en el estado sensorial actual, manteniendo ese estado hasta que se sobrescribe cuando se realiza la siguiente acción.

Memoria a largo plazo: Es una acción producida al recuperar un patrón episódico almacenado en la red.

MRAC: Control adaptativo de referencia de modelo.

MC: Control Multivariable.

MB-NCS: Controlador en red basados en modelos.

NCS: Control Basado en Modelos de Sistemas en Red.

OS-ELM: Máquina de Aprendizaje Extremo para Datos en Lote.

PAT: Prueba de Promedios por Parte o Límites Dinámicos.

PAT-ELM: Red de Monitoreo Cognitivo Basada en Máquina de Aprendizaje Extremo.

RL: Aprendizaje por Refuerzos.

RLS: Mínimos cuadrados recursivos por sus siglas en inglés.

RNA: Red Neuronal Artificial.

SC: Control supervisado.

SISO: Control entrada-única-salida-única.

SLFN: Red de Avance Hacia Adelante de una Sola Capa.

SUB: Subiculum.

SVM: Máquinas de soporte de vectores.

TD: Algoritmo de Aprendizaje por Refuerzos (Temporal Difference).

1. INTRODUCCIÓN A LA INVESTIGACIÓN

En la actualidad, existe un amplio desarrollo en los sistemas de control para automatizar procesos de manufactura utilizando procesamiento computacional, lo cual permite el uso de algoritmos y secuencias de control complejas, que buscan optimizar los procesos, permitiendo la toma de decisiones de manera autónoma. Al mismo tiempo, existe un desarrollo muy significativo en algoritmos inspirados en el funcionamiento del cerebro biológico con aplicaciones en todas las ramas de la ciencia y la tecnología.

El objetivo de esta investigación es implementar un algoritmo de control, capaz de controlar procesos de manufactura que no tienen modelo o que tienen un alto grado de incertidumbre. Esto es importante, ya que conducirá a una optimización sustancial en los procesos, permitiendo generar en la industria manufacturera, productos avanzados con un alto grado de confiabilidad y costos menores de producción, al reducir el desperdicio.

En esta investigación, mostramos que es posible crear un Control basado en inteligencia computacional para las automatizaciones usadas en los procesos de manufactura. Dicho Control, puede lograr resultados relevantes mediante la utilización de un algoritmo hipocámpico soportado por algoritmos de máquinas de aprendizaje extremo.

La existencia de controles computacionales en los procesos de manufactura y el desarrollo de nuevos algoritmos relacionados con el hipocampo biológico, inspiraron el presente trabajo de investigación; el cual se propone desarrollar un algoritmo de control cognitivo que combine algoritmos basados en el funcionamiento del cerebro biológico con la capacidad de operar en ambientes de manufactura, donde es difícil controlar la gran cantidad de variables y tiempos de proceso en los que, la respuesta debe mejorar en forma acumulativa.

Para ello, se selecciona una técnica relativamente nueva en Redes Neuronales, conocida como Máquina de Aprendizaje Extremo (Extreme Learning Machine) o ELM, que ha ganado mucho interés por su capacidad de ser entrenada de forma eficiente y rápida, ya que, los parámetros de la red se obtienen analíticamente

mediante el cálculo de una pseudoinversa, sin necesidad de utilizar costosos ciclos de entrenamiento. Esto hace que este tipo de algoritmo sea altamente aplicable en los procesos de manufactura industrial.

Otra familia de algoritmos incluidos en el desarrollo de esta tesis, también inspirados en funciones neurológicas, son las redes de aprendizaje no supervisado del tipo autoasociativa y de Aprendizaje por Refuerzos. Estos algoritmos tienen la característica de que pueden manejar conjuntos de datos como si fuesen bloques de memoria episódica o de trabajo. La cualidad principal de los algoritmos inspirados en el hipocampo es, que pueden 'memorizar' situaciones complejas similares a las que se encuentran en muchos de los procesos de manufactura avanzados.

Sin embargo, los algoritmos que se han desarrollado hasta hoy, no se orientan a manufactura, han sido utilizados principalmente para el procesamiento de imágenes, reconocimiento de patrones, o para la navegación robótica.

La investigación traza un puente entre los algoritmos inspirados en el hipocampo biológico, los algoritmos de control de ruta robótico y las máquinas de aprendizaje extremo para su aplicación práctica en procesos de manufactura complejos.

El presente trabajo propone dos Algoritmos Híbridos inspirados en el funcionamiento del hipocampo biológico con respecto al manejo de los recuerdos episódicos y la memoria a largo plazo, compuestos por varios bloques o instancias, formados a su vez por algoritmos ELMs y algoritmos no supervisados. A uno de los Algoritmos Híbridos se le denomina Red Neuronal AHN-ELM (por sus siglas en inglés *Artificial Hippocampical Network with support in Extreme Learning Machine*) y tiene como propósito emplearse para controlar procesos de manufactura que requieran estar aprendiendo constantemente de manera autónoma. Al segundo Algoritmo Híbrido se denomina PAT-ELM (Algoritmo de Prueba de Promedios por Parte basado en Máquina de Aprendizaje Extremo) y tiene como propósito emplearse para monitoreo o caracterización de procesos de manufactura.

A diferencia de los algoritmos existentes para el control Robótico de ruta, que usan típicamente 2 ó 3 instancias hipocámpicas, se selecciona en la investigación el

desarrollo de un Algoritmo Híbrido que usa 4 instancias para adecuarlo al manejo de procesos de manufactura.

La ventaja de usar 4 instancias hipocámpicas para el algoritmo de control de automatizaciones es que permite tener las 4 funciones críticas que se requieren para controlar eficientemente un sistema de control automático para procesos de manufactura. Las 4 funciones son: (1) detección de datos anormales y separación de patrones, (2) auto asociación de eventos episódicos, (3) Completar recuperación de eventos episódicos y (4) autoaprendizaje al ejercitar con el entorno.

Los algoritmos de las máquinas de aprendizaje extremo proporcionan una velocidad de aprendizaje mayor que la mayoría de las redes neuronales actuales, esto es necesario si se quiere aplicar en procesos industriales de control.

Se utiliza también un sistema de filtrado denominado *Part Average Testing* que permite eliminar patrones de datos anormales, el algoritmo típicamente se usa en la fabricación de obleas semiconductoras en la industria electrónica, para esta investigación se desarrolla una nueva implementación que permite usarlo para procesar datos secuenciales agregando límites dinámicos.

Se adiciona una red neuronal recurrente para crear un sistema de reconocimiento de patrones, se selecciona una red que tiene una alta velocidad de aprendizaje para que sea compatible con los procesos de manufactura.

También un algoritmo de aprendizaje por refuerzos que permite aprender del entorno en forma dinámica no supervisada.

La implementación de los 4 algoritmos crea un Algoritmo Híbrido denominado Red Hipocámpica Artificial, basada en Máquina de Aprendizaje Extremo (**AHN-ELM**).

Posteriormente el Algoritmo Híbrido PAT-ELM (se denomina red de prueba de partes por promedios parciales basado en máquina de aprendizaje extremo) se desarrolló durante la investigación para poder proporcionar una función de caracterización que emplea solo 1 de las instancias hipocámpicas y que aprovecha la capacidad del algoritmo PAT.

Para verificar la funcionalidad del Algoritmo Híbrido AHN-ELM, se realizaron dos tipos de experimentos: virtuales y de aplicaciones prácticas en manufactura. Los primeros, virtuales, consistieron en aplicaciones del Algoritmo Híbrido tanto en un ambiente computacional como bajo condiciones de laboratorio controladas. Los segundos, de aplicación práctica, consistieron en la aplicación del Algoritmo Híbrido tanto en la fabricación de moto-protectores eléctricos, controlando el automatismo del formado mecánico, como en la caracterización de láminas bimetalicas termostáticas, que son utilizadas en la fabricación de moto protectores. A dichos experimentos se les conoce como técnicas integradas de inteligencia computacional que son una variante del control, basado en inteligencia computacional. El formado de bimetales fue seleccionado como experimento para validar a este algoritmo, por la complejidad que representa debido a la variabilidad y al ruido estadístico, que afectan tanto a la materia prima como a los ajustes del proceso, alterando el desempeño final de la operación.

El Algoritmo Híbrido PAT-ELM es una versión simplificada de la instancia de entrada denominada DG (Dentate Gyrus), que permite la eliminación de datos anormales y logra la categorización o pronóstico del proceso basado en entrenamiento al reemplazarse la Red Neuronal OSELM (Algoritmo Secuencial basado en Máquina de Aprendizaje Extremo) por la ELM básica.

La experimentación en los bimetales tuvo dos momentos clave: la primera parte fue la caracterización de la tira bimetalica antes de procesarse, para identificar si resulta adecuada para manufactura. Esto es de suma importancia, ya que el perfil de curvatura determina varias características del sistema de disparo del moto-protector, tales como el tiempo de apertura, el tiempo de cerrado, la temperatura de apertura y la temperatura de cierre, así como la fuerza que produce la tira bimetalica para realizar el movimiento.

La segunda parte de la validación práctica del Algoritmo Híbrido AHN – ELM, consistió en controlar una máquina de formado mecánico semiautomática, en este caso la máquina cuenta con un cabezal ajustable por computadora para el formado de la curvatura de la tira bimetalica. La retroalimentación de este sistema se basa en un equipo que mide las características antes mencionadas y que

después son transferidas a una computadora, donde se ejecuta el Algoritmo Híbrido AHN-ELM. El objetivo de este experimento es autocontrolar la máquina, esto es, que vaya aprendiendo a optimizar sus procesos a medida que van ocurriendo.

Los resultados de la experimentación, tanto la virtual como la de aplicación práctica en el formado y caracterización de la tira bimetálica, mostraron que los algoritmos propuestos funcionan y por lo tanto, pueden ser aplicados a los procesos de manufactura, alcanzando el objetivo inicial de esta investigación.

Los dos nuevos Algoritmos Híbridos propuestos en esta investigación, tanto el AHN-ELM, como el PAT-ELM, representan una innovación para el control cognitivo hipocámpico con vistas a su aplicación en la manufactura avanzada. Ambos algoritmos pueden ser aplicados en gran variedad de procesos de manufactura, y finalmente, pueden servir como punto de partida para nuevos desarrollos.

1.1. MOTIVACION DE LA INVESTIGACIÓN

La principal motivación para realizar esta investigación fue la de desarrollar un Sistema de Control Cognitivo a semejanza de los sistemas de control robótico, capaz de aprender de manera autónoma cómo optimizar, de manera acumulativa, un proceso de fabricación complejo en un ambiente de manufactura.

Esta investigación busca crear un nuevo Algoritmo Híbrido, tomando como base los algoritmos utilizados en el Sistema de Control Robótico Cognitivo, aplicando Redes Neuronales que no requieran grandes cantidades de datos para dicho aprendizaje y que puedan ser entrenadas rápidamente, lo que fungirá como herramienta clave para mejorar procesos de manufactura avanzada y, a su vez, consistirá en un aporte relevante a nivel técnico sobre Redes Neuronales Artificiales inspiradas en el Hipocampo.

1.2. DEFINICIÓN DEL PROBLEMA

El problema a resolver en esta investigación es el de crear un nuevo Algoritmo Híbrido capaz de aprender rápidamente de las variables generadas en un proceso de manufactura industrial y que proporcione una salida de control que optimice dicho proceso. Este nuevo Algoritmo Híbrido debe ser capaz de aprender de la información recibida de manera autónoma, así como de procesar nuevos datos procedentes de los lotes de producción en forma continua y de discernir qué información es relevante y qué información es anormal para descartarla. Lo que abre las posibilidades de aplicación de la Inteligencia Artificial, particularmente de las Máquinas de Aprendizaje Extremo a procesos de manufactura y aporta a su desarrollo técnico.

1.3. PREGUNTA DE INVESTIGACIÓN

La pregunta que guía la presente investigación es:

¿Cómo desarrollar un nuevo Algoritmo Híbrido para optimizar un proceso secuencial de control de manufactura industrial mediante el aprendizaje automático, autónomo, auto-reforzante, replicante y auto-configurable para asimilar rápidamente información en tiempo real?

Ahora bien, toda pregunta de investigación cumple la función de acotar los alcances de la investigación y dirigirla y la respuesta a ella se ve, a su vez, acotada por los tiempos y recursos disponibles. La presente tesis desarrolla la respuesta obtenida dentro de dicho marco.

A continuación, se definen a detalle los conceptos clave de la pregunta de investigación:

- **Autónomo:** el Algoritmo no requiere que un programador defina una respuesta específica como respuesta del sistema, es capaz de catalogar y descubrir información nueva de manera independiente sin depender de un programador.

- **Auto-reforzante:** el Algoritmo genera un conjunto de respuestas basadas en un conglomerado de datos iniciales; posteriormente, al existir nuevos conglomerados de datos, el conjunto de respuestas se ajusta para minimizar los errores de pronóstico.
- **Replicante:** el Algoritmo puede crear copias de conglomerados de conocimiento que lo capacita para evolucionar en una dirección diferente y finalmente seleccionar el resultado que logre minimizar los errores de predicción.
- **Auto-configurable:** implica que la morfología del Algoritmo puede cambiar de manera automática para minimizar el tiempo requerido de aprendizaje.

1.4. OBJETIVOS

El objetivo general y los objetivos específicos de esta investigación se presentan a continuación:

1.4.1. Objetivo general

El objetivo de la presente investigación consiste en crear un Algoritmo Híbrido, basado en RNA's, que pueda controlar un automatismo industrial, que posea la capacidad de aprender rápidamente la información del entorno para optimizar el resultado del proceso, esto es, debe reaccionar rápidamente a cambios en las propiedades de las materias primas o variaciones inesperadas del proceso. Además, debe tener la capacidad de memorizar ajustes y parámetros para poder compensar eficazmente los cambios de las diferentes y múltiples variables que afectan al proceso.

1.4.2. Objetivos específicos

Los objetivos específicos que se buscan lograr con esta investigación se presentan a continuación:

- a) Desarrollar un Algoritmo Híbrido basado en RNA's para procesos de manufactura que funcione adecuadamente en presencia de ruidos o variaciones típicas en los procesos industriales.

- b) Se busca que el nuevo Algoritmo Híbrido simule el proceso de manejo de memorias en forma semejante a como ocurre en el Hipocampo Biológico para facilitar el aprendizaje y adaptabilidad de la propuesta al manejo de lotes de producción.
- c) Se quiere implementar el nuevo Algoritmo Híbrido al proceso de manufactura de formado de curvatura en bimetales utilizados en la fabricación de moto-protectores eléctricos.

1.5. HIPÓTESIS

Se puede crear un sistema de control para automatismos en manufactura basado en Redes Neuronales Artificiales (RNA) que aprenda a manejar de forma óptima un proceso de manufactura industrial complejo sin necesidad de supervisión. Para lograrlo, se aprovechan las cualidades desarrolladas en los sistemas de control cognitivo utilizados actualmente en la navegación robótica, específicamente utilizando una Red Neuronal Hipocámpica Artificial (AHN), que tiene la cualidad de manejar diferentes tipos de memorias útiles para compensar las variaciones típicas de los procesos de manufactura industrial.

Utilizar la red conocida como Máquina de Aprendizaje Extremo (ELM) permite evitar tiempos elevados de entrenamiento que, en aplicaciones de manufactura reales, hace prohibitiva la utilización de RNAs. La Red ELM tiene una velocidad de aprendizaje elevada y es muy versátil. La ELM permite la creación de algoritmos híbridos capaces de combinar los beneficios del control robótico cognitivo, la gestión de memorias de las redes hipocámpicas y su aplicación a proceso de manufactura complejos reales.

1.6 DESCRIPCIÓN GENERAL DE LA TESIS

La presente tesis está organizada en 8 capítulos. El capítulo 2 presenta el marco teórico y los fundamentos sobre los que descansa la investigación. El capítulo 3 contiene el procedimiento de la investigación; la sección 3.1 muestra el desarrollo

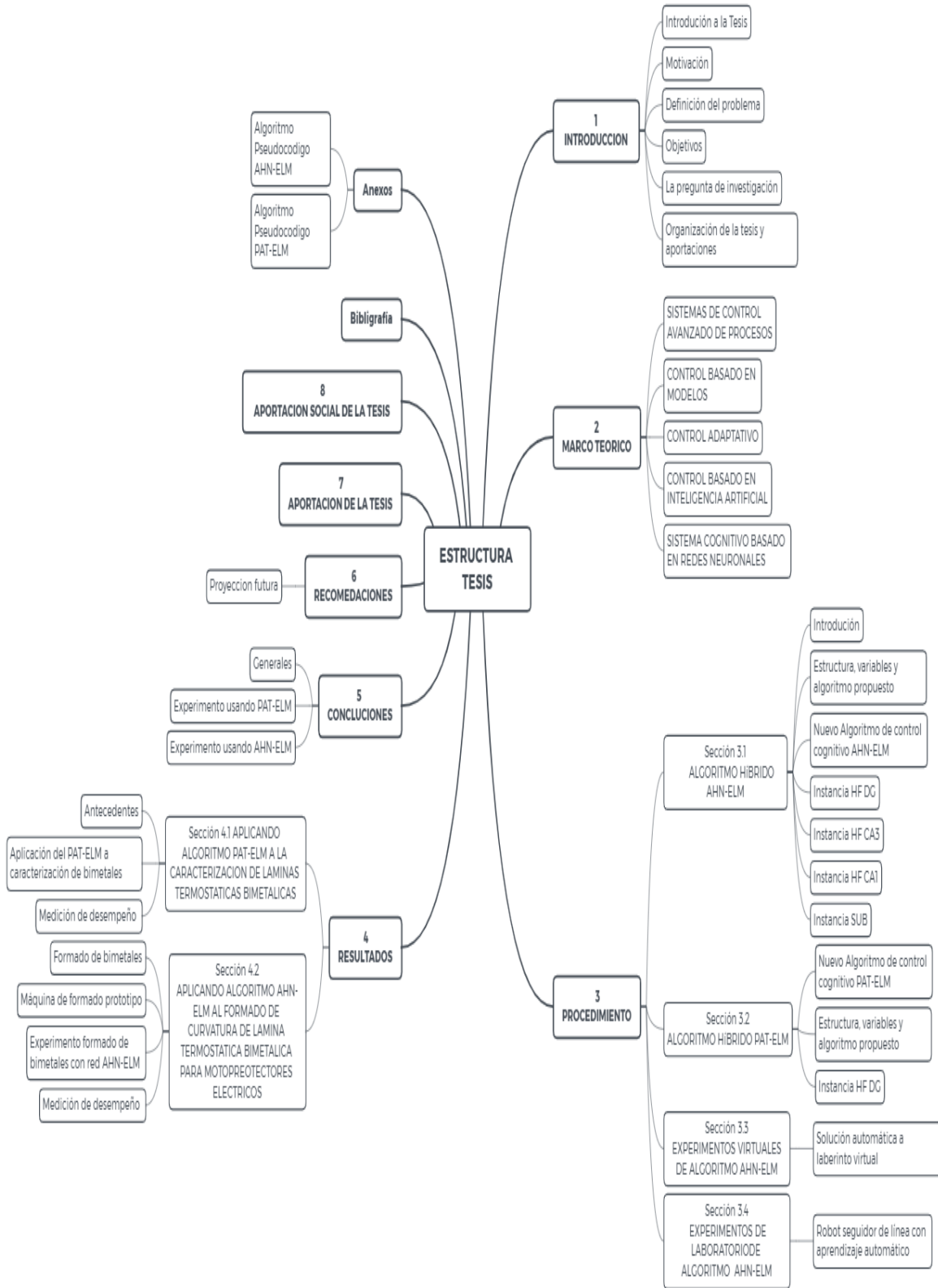


Figura 1. Mapa mental con la distribución de los capítulos que conforman la tesis. Elaboración propia.

del Algoritmo Híbrido AHN-ELM denominado Red Hipocámpica Artificial soportada por máquinas de aprendizaje extremo; la sección 3.2 contiene el desarrollo del Algoritmo Híbrido PAT-ELM denominado Red de Prueba por promedios parciales basada en máquina de aprendizaje extremo. Las secciones 3.3 y 3.4 muestran las pruebas de validación de los algoritmos usando pruebas virtuales y de laboratorio. El capítulo 4 contiene los experimentos y aplicaciones prácticas, para comprobar el funcionamiento de ambos Algoritmos Híbridos y finalmente, el capítulo 5 muestra las conclusiones de la investigación. La Figura 1 muestra un mapa mental de la distribución de los temas para dar al lector una idea la composición completa de la investigación.

A continuación, se describe brevemente el contenido de cada uno de los capítulos, indicando las contribuciones realizadas.

- 1. Introducción a la investigación.** Consiste en la introducción a los sistemas de Control Basado en Modelos para automatizaciones utilizadas en manufactura y monitoreo cognitivo, aplicados a procesos de manufactura basados en RNA's. Se proporcionan las ideas básicas de los Algoritmos Híbridos de control hipocámpico para control de ruta en robots denominados AHA!, HMSNN y HTM-MAT.

También se habla acerca de la motivación para el desarrollo de esta investigación, se explica el problema a resolver y la pregunta de investigación, que da marco y alcance al desarrollo de la investigación que se presenta en el resto de la tesis. Al final del capítulo se presenta la organización de la tesis y las aportaciones que tiene este trabajo, así como las posibles aplicaciones futuras.

- 2. Marco teórico.** Examina los diferentes tipos de control avanzado usados en automatización industrial. Se da el análisis a los sistemas de Control Basado en Modelos para automatizaciones utilizadas en manufactura y monitoreo cognitivo, aplicados a procesos de manufactura basados en RNA's. Se proporcionan las ideas básicas de los Algoritmos Híbridos de control

hipocámpico para control de ruta en robots denominados AHA!, HMSNN y HTM-MAT.

Se presenta una clasificación de los diferentes sistemas en uso para el control de procesos de manufactura complejos que requieren de aprendizaje continuo.

En particular, se centra en las siguientes clases de enfoques de control: control basado en modelos, control, control adaptativo, control basado en sistemas de eventos discretos y, finalmente, basado en inteligencia computacional.

Los beneficios y problemas relacionados con cada tipo de control se examinan a través de un análisis de la literatura.

El resultado del análisis de la literatura es un marco que coincide con el estado actual de los sistemas de control en automatización industrial usados en manufactura avanzada.

- 3. Procedimiento.** Se explica el Desarrollo del Nuevo Algoritmo Híbrido AHM-ELM para Control de Procesos. La sección 3.1 consiste en el desarrollo del Algoritmo Híbrido AHM-ELM , generado para su utilización en el control cognitivo de procesos, el cual se conforma de la Red Hipocámpica Artificial con soporte en Máquinas de Aprendizaje Extremo o AHN-ELM. Se describe su estructura como híbrida, sus algoritmos principales se componen de la máquina de aprendizaje extremo en dos configuraciones, una red neuronal recurrente, y un algoritmo de aprendizaje por refuerzos y un filtro dinámico para eliminación de anomalías. Se explica el funcionamiento de los procesos como la separación y asociación de patrones episódicos, la recuperación de patrones y el ajuste de la respuesta al entorno. Se expone en detalle, cómo se crean los elementos o instancias de la red hipocámpica que integran al nuevo Algoritmo Híbrido: DG (Dentate Gyrus), CA1 y CA3 (Cornu Ammonis 1 y 3) y se habla de la instancia SUB (Subiculum), y de cómo todos estos elementos se interconectan entre sí para formar a la Red Hipocámpica Artificial AHN-ELM.

Además, se explica Desarrollo del Algoritmo Híbrido PAT-ELM para Monitoreo de Procesos. La sección 3.2 conforma de la Red Hipocámpica de Prueba por Promedios Parciales con soporte en Máquinas de Aprendizaje Extremo o PAT-ELM que se puede utilizar para la caracterización o monitoreo de procesos industriales. Se describe su estructura y algoritmos principales de que se compone de la máquina de aprendizaje extremo y el filtro dinámico para eliminación de anomalías.

También se explican los experimentos Virtuales y de Laboratorio AHN-ELM. La sección 3.3 expone los experimentos realizados en laboratorio y experimentos virtuales del Algoritmo Híbrido AHN-ELM en la aplicación de control robótico cognitivo. El objetivo de los experimentos virtuales es demostrar que el Algoritmo Híbrido AHN-ELM es funcional, y las pruebas de laboratorio se realizaron para desarrollar las técnicas para su conexión con el mundo real. Se detalla el experimento virtual, utilizando un agente para resolver un laberinto en la computadora. Además, se presenta también el desarrollo de un robot seguidor de línea con aprendizaje automático y retroalimentación ultrasónica para encontrar un objetivo. Se mencionan los tipos de interfaces utilizados para su conexión con el mundo real.

4. Resultados. Aplicando el Algoritmo Híbrido PAT-ELM a la Caracterización de Bimetales. Experimento 1.

La sección 4.1 explica la utilización del Algoritmo Híbrido PAT-ELM para el monitoreo cognitivo automático (deducido a partir del AHN-ELM), aplicado al proceso de formado mecánico de bimetales. En este caso en particular, para la caracterización física de las propiedades de la tira bimetálica termostática, que se utiliza para formar los discos de disparo del material, antes de ser ensamblados en los moto-protectores eléctricos. Adicionalmente, se muestran los detalles de la selección del equipo y la metodología de experimentación, así como sus resultados.

Aplicando AHN-ELM al Control Automático de Formado de Bimetales. Experimento 2.

La sección 4.2 explica los detalles de la aplicación del Algoritmo Híbrido AHN-ELM para el control robótico cognitivo automático en un proceso de formado de bimetales, los cuales, son utilizados para la fabricación de moto-protectores eléctricos. Se muestran los detalles de la selección del equipo, la metodología de experimentación y los resultados obtenidos.

- 5. Conclusiones.** El capítulo 5 presenta las reflexiones y reportes finales de los experimentos, tanto los de laboratorio, como los de aplicación de las Redes AHN-ELM y PAT-ELM para el control cognitivo del formado del bimetale y la caracterización de discos bimetálicos, respectivamente. Se mencionan los argumentos a favor y en contra de aplicar los Algoritmos Híbridos de control cognitivo para los procesos de manufactura avanzada, así como los requisitos para que sean seleccionados como herramientas de control. También se analizan las posibles aplicaciones futuras de los nuevos Algoritmos Híbridos, así como las posibles proyecciones y aplicaciones futuras de la presente investigación.

1.7 SOLUCIÓN PROPUESTA

Se proponen, para resolver la pregunta de investigación, dos nuevos Algoritmos Híbridos. Al primero se le denomina Red Hipocámpica Artificial Soportada en Máquina de Aprendizaje Extremo, o Red Neuronal AHN-ELM (*Artificial Hippocampical Network based on Extreme Learning Machine*). El segundo es una versión simplificada del primer Algoritmo al que se le denomina PAT-ELM (*Part Average Testing based on Extreme Learning Machine*).

Los dos Algoritmos Híbridos propuestos se inspiran en el funcionamiento del hipocampo biológico, en sus cuatro principales formaciones hipocámpicas: Dentate Gyrus (DG), Cornu Ammonis 3 (CA3), Cornu Ammonis 1 (CA1) y Subiculum (SUB).

La función biológica de la formación hipocámpica Dentate Gyrus en el cerebro (DG) es la separación de patrones, que permite distinguir los eventos episódicos

similares. La formación Cornu Ammonis 3 (CA3) se encarga de la auto asociación de eventos episódicos, que guarda en la memoria de largo plazo. La formación Cornu Ammonis 1 (CA1) se encarga de completar patrones, es decir, de recuperar eventos episódicos de la memoria de largo plazo y completarlos con las correspondientes reacciones iniciales respecto al entorno, dando lugar a la memoria temporal. El Subiculum (SUB) se encarga de generar nuevas respuestas basándolas en las reacciones que van ocurriendo en el momento, buscando obtener la mejor respuesta posible, dando lugar a la memoria de trabajo. La Figura 2 muestra un esquema de la solución propuesta.

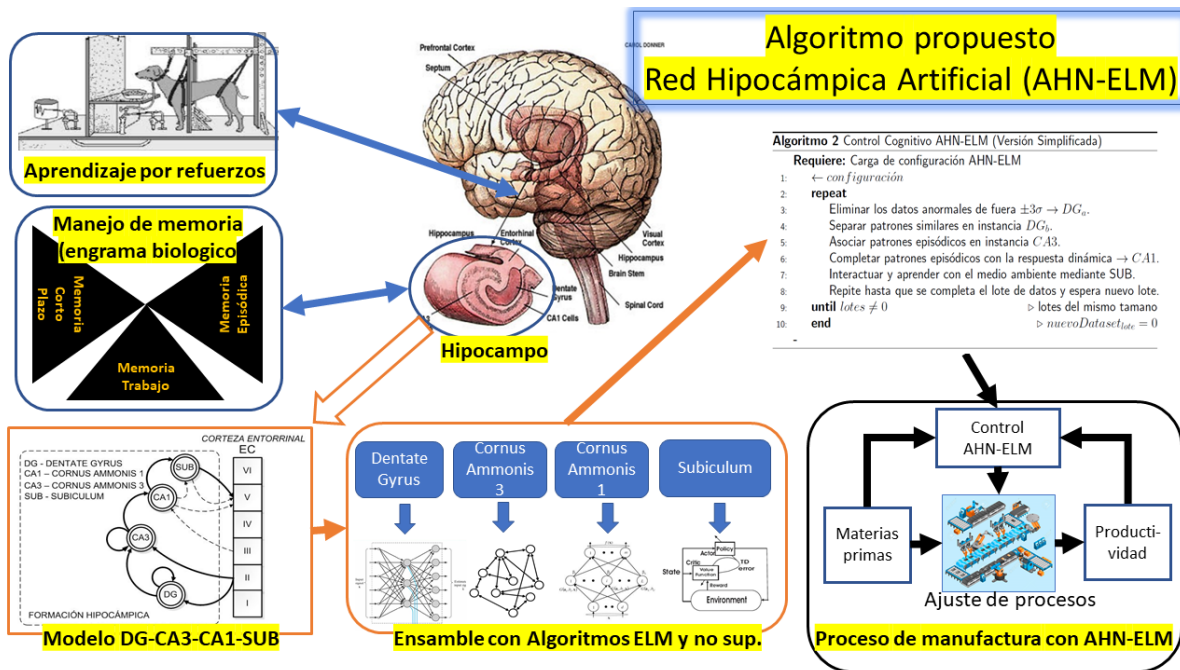


Figura 2. Esquema de la red híbrida AHN-ELM.
Elaboración propia.

Para implementar las formaciones hipocámpicas, se crean instancias que utilizan redes de aprendizaje supervisado, basadas en Máquina de Aprendizaje Extremo, en su modo autocodificador y en modo de aprendizaje secuencial. También se implementan redes de aprendizaje no supervisado, utilizando la Red Neuronal recurrente y el Aprendizaje por Refuerzos.

2. MARCO TEÓRICO

2.1 SISTEMAS AVANZADOS DE CONTROL AUTOMÁTICO PARA PROCESOS DE MANUFACTURA

La producción industrial es en la actualidad, un poderoso motor en la economía mundial. El valor agregado representa más del 30.5% del PIB en las economías avanzadas. Para lograr altos niveles de calidad y eficiencia, las plantas de manufactura industrial están cada vez más automatizadas (Agency, 2020). La automatización de fábricas se ha convertido en una industria en sí misma, con una facturación anual de más de 152 mil millones de dólares. Así pues, la industria de la automatización está fomentando un incremento en la competitividad global que impulsa la comercialización de resultados de la investigación en esta área. La fabricación de nuevos sistemas de control para las automatizaciones promueve la innovación y la productividad, lo que resulta un elemento indispensable en la cadena de innovación que permite la aplicación exitosa de nuevas tecnologías (Mitchell, 2014).

El desarrollo de sistemas de control automático se enfrenta a varios desafíos técnicos y teóricos, encaminados a permitir su aplicación eficiente en la automatización industrial, derivados de las crecientes demandas de los clientes, la alta competitividad y la evolución del entorno económico. Los desafíos son claros, pero aún no se han resuelto satisfactoriamente (Precup, 2011).

La atención se centra en las llamadas técnicas de control avanzado, que están adquiriendo una importancia cada vez mayor en la automatización industrial, gracias a las nuevas tecnologías emergentes como el mayor poder computacional y la existencia de algoritmos más eficientes (Darby, 2012). Dichas técnicas de control avanzado deben responder de manera flexible a los cambios de materias primas y las variaciones de los procesos; generando una salida capaz de interactuar con su entorno (Kostal, 2012).

El término 'sistema de control avanzado' se refiere a una amplia gama de técnicas aplicadas en procesos industriales que, típicamente, integran herramientas de varias disciplinas tales como la ingeniería de sistemas de control, el procesamiento de señales, estadística, teoría de decisiones e inteligencia artificial (Dotoli, 2016). En efecto, todas estas técnicas y algoritmos se caracterizan por su adaptabilidad para ser parte de los diseños de controladores de alto rendimiento que pueden aplicarse a procesos de multivariables, con alto orden, que normalmente son no lineales y están sujetos a un sin número de restricciones. Los diversos enfoques de control avanzado para la automatización de procesos se agrupan en tres clases importantes de técnicas empleadas en manufactura:

- Técnicas de control basadas en modelos.
- Técnicas de control adaptativo.
- Técnicas de control basadas en inteligencia computacional.

Las siguientes secciones del estado del arte resumen en términos de resultados teóricos, cada una de las técnicas de control mencionadas y sus áreas de aplicación más relevantes. Se destacan las estrategias de control particulares y los desafíos o problemas que son inherentes a sus algoritmos en el control avanzado de procesos automáticos (Astrom, 2014).

2.2 CONTROL BASADO EN MODELOS

El Control Basado en Modelos o MBC es una técnica extendida con una amplia gama de aplicaciones industriales, incluidos procesos en sistemas químicos, mecánicos y térmicos. El acceso a sistemas de cómputo más poderosos en la última década ha influenciado en varias direcciones el desarrollo de MBC en la industria de manufactura (Morato, Normey-Rico, & Sename, 2020).

El control basado en modelos se fundamenta en el diseño de un sistema de control y la retroalimentación de su desempeño en un entorno de simulación, por medio de modelos matemáticos que representan el proceso de manufactura (Astrom, 2014). Es común el uso de modelos no lineales, aunque tienen el defecto de tener altos costos en el desarrollo del modelado del proceso específico. Debido a esto,

existe recientemente una fuerte motivación en crear bibliotecas de modelos genéricos, adaptados al control de automatizaciones, para reducir tiempos y costos de implementación (Ozkan, Bombois, & Ludlage, 2016). También en el mismo contexto, las aplicaciones MBC conmutadas o híbridas crean una alternativa de implementación y desarrollo para el uso de modelos no lineales (Morari, 2009). Uno de los inconvenientes de los sistemas de control MBC corresponde a lograr que este tipo de controles sean robustos, ya que actualmente carecen de un método práctico para su implementación en la mayoría de los procesos industriales (Lee, 2011)(Darby, 2012).

La Figura 3 muestra el esquema del funcionamiento del control MBC. En este caso las entradas son la salida del sistema a controlar, y una serie de N referencias en los siguientes N instantes. Los elementos principales que componen al MBC son los siguientes:

- **Modelo del sistema:** Es utilizado para predecir la salida que presentará el sistema a controlar ante la aplicación de una determinada acción de control. Cuanto más preciso sea el modelo, la acción de control calculada será mejor.
- **Función de costo:** Es un elemento fundamental en el control basado en modelos, ya que de ella dependen la acción de control calculada y la estrategia buscada por el controlador. Esta función permite buscar una solución que simplemente minimice la diferencia entre las salidas predichas y la salida real.
- **Optimizador:** Es el algoritmo encargado de resolver el problema de optimización buscando la secuencia de acciones de control que minimicen la función de costo. Es necesario realizar un proceso típicamente iterativo, en el que el optimizador trata de minimizar la función de costo, suponiendo una secuencia de acciones de control, estimando las salidas del sistema mediante el modelo, y calculando el valor de la función para esos valores.

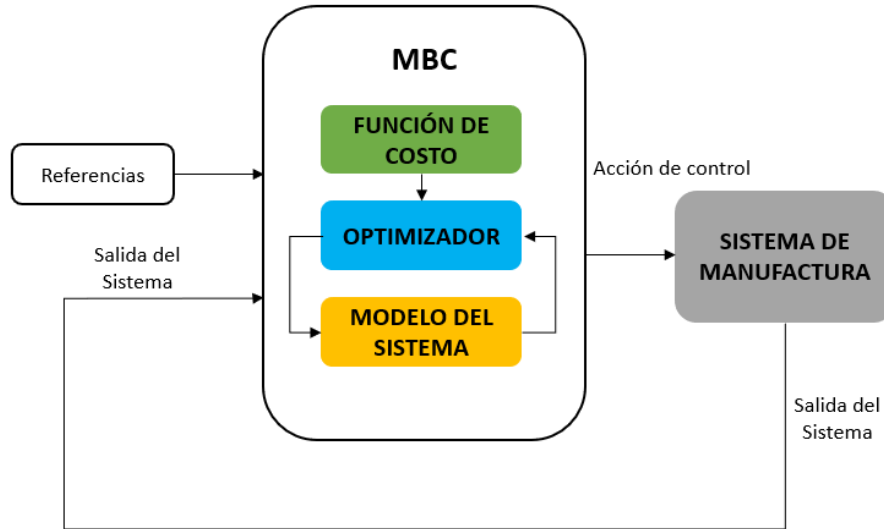


Figura 3. Esquema de control de un MBC.
Elaboración propia.

Los sistemas de control MBC descentralizados y distribuidos o DMPC que se utilizan para aplicaciones de automatización a gran escala, se basan en controladores que realizan los cálculos por separado, utilizan una red de comunicación para poder cooperar y generar el sistema de control distribuido fusionado como si tuviera un lazo de control retroalimentado (Christofides, 2013). Dado que los procesos de la industria manufacturera están cada vez más automatizados, con enfoques de control basados en modelos avanzados y redes de comunicación, sensores y actuadores, el DMPC podría ser una opción en el diseño de los sistemas de control basados en modelos distribuidos de próxima generación (Christofides, 2013). Como se puede inferir, la industria manufacturera sigue siendo, uno de los campos más amplios en los que se pueden aplicar los algoritmos MBC, sobre todo en lo que respecta a MBC de sistemas híbridos a pesar de las limitaciones inherentes (Mayne, 2014).

2.2.1. Modelo de control predictivo

El modelo de Control Predictivo (MPC, por sus siglas en inglés) es considerado como uno de los algoritmos más usados en el contexto del control avanzado en automatización manufacturera, la razón de su uso se debe a que permite

implementar el control en el dominio del tiempo, esta característica está presente en muchas de las aplicaciones industriales (Camacho & Bordons, 2007).

El algoritmo MPC proporciona una solución de control integrado que se puede aplicar a sistemas con multivariables que interactúan entre ellas de forma dinámica, que son complejas y con restricciones. De tal forma que en la actualidad se ha convertido en uno de los algoritmos estándar para el control de procesos de la industria (Darby, 2012), (Mayne, 2014), (Qin & Badgwell, 2003) , (Jämsä-Jounela, 2007).

El MPC emplea un modelo del proceso a controlar para predecir su evolución en un horizonte de tiempo futuro. Las predicciones generadas se utilizan luego para calcular una señal de control, que optimiza el comportamiento del proceso usando las respuestas del sistema y del automatismo. Para determinar la secuencia de movimientos de control, se requiere un modelo dinámico del proceso, un historial de movimientos de control y una función de optimización de costos durante el horizonte de predicción en retroceso (Jiayu, Zongchou, Jiahao, & Gartlib, 2020).

Existe un grupo de algoritmos y técnicas de MPC lineales. como lo son el modelo de control heurístico predictivo, el control matricial dinámico, el control auto adaptativo de predicción extendida y el control predictivo generalizado; y, más recientemente, las aplicaciones de MPC no lineal y sistemas MPC híbridos (Camacho & Bordons, 2007), (Jämsä-Jounela, 2007), (Camacho, Ramirez, Limon, & Peña, 2010).

Los métodos MPC para sistemas y procesos híbridos se empiezan a usar en control de procesos, sistemas de fabricación, sistemas automotrices y sistemas de control de energía (Camacho, Ramirez, Limon, & Peña, 2010). Igualmente, se están desarrollando aplicaciones control en red que tienen técnicas de control híbridas. Las áreas típicas de aplicaciones industriales para la estrategia MPC se presentan en la operación de plantas químicas y petroquímicas, junto con el control de refinerías (Jämsä-Jounela, 2007). Así mismo, las áreas emergentes para aplicaciones MPC incluyen: minería (tratamiento de minerales), procesamiento de láminas, procesamiento de combustible, producción de petróleo, generación de

energía, así como la industria de la pulpa y el papel (Qin & Badgwell, 2003), (Jämsä-Jounela, 2007), (Darby, 2012).

Otra área en la que el MPC se está comenzando a utilizarse es la industria farmacéutica, el procesamiento de alimentos e incluso ha penetrado en la fabricación discreta, sobre todo en la industria automotriz. Para todas estas aplicaciones, se identifica un modelo del sistema a controlar a partir de datos generados por pruebas dedicadas (Darby, 2012).

Una de las limitaciones de los algoritmos MPC es que se basan en la solución de un problema de optimización lineal, esta limitante ha impedido su aplicación a procesos que requieren altas tasas de muestreo. Para reducir la tasa de muestreo promedio, se ha propuesto la estrategia de Control Predictivo Generalizado basado en eventos (Miskowicz, 2017). Las ventajas de los controladores Control Predictivo Generalizado basados en eventos, resultan muy eficientes para problemas de control de temperatura. La explotación de los cálculos basados en un modelo permite reducir la comunicación en los sistemas activados por eventos, tanto para los enlaces de sensor a controlador como de controlador a actuador en los sistemas lineales (Heemels & Donkers, 2013).

2.2.2 Control basado en modelo de sistema en red

Los sistemas de control automático en manufactura marcaron un avance tecnológico muy significativo en el siglo pasado. Después del año 2000 con la llegada de los sistemas industriales que integran una mayor complejidad e interconectividad, incluyendo los sistemas inalámbricos, crearon una serie de nuevos desafíos a las técnicas de control que habían funcionado bien anteriormente (Astrom, 2014). Al respecto, el enfoque en los controles con conectividad a una red ha inspirado el desarrollo de los sistemas de control red basados en modelos (MB-NCS, por sus siglas en inglés). Los MB-NCS son algoritmos de control de procesos lineales en los que, el sensor está conectado a un controlador-actuador lineal a través de una red, y tanto la retroalimentación del estado como de salida se consideran, puede producir respuestas para procesos

tanto continuos como discretos, (Montestruque & Antsaklis, On The Model-Based Control of Networked Systems, 2003)

La idea clave del algoritmo de control MB-NCS es que al conocer la dinámica del proceso que se utiliza se puede reducir el uso de la red al aproximar el comportamiento del proceso durante el tiempo en que los datos del sensor no están disponibles (Montestruque, Garcia, & Antsaklis, Model-based control of networked systems, 2014).

La retroalimentación del sistema se proporciona en momentos discretos mediante la actualización del estado del modelo en el controlador, utilizando el estado del proceso generado por el sensor (Montestruque, Garcia, & Antsaklis, Model-based control of networked systems, 2014).

El principio de los MB-NCS se basa en generar estimaciones del estado real entre intervalos de actualización basadas en la dinámica nominal del sistema. La utilización de la red se minimiza con el uso del algoritmo MB-NCS. Una ventaja de este tipo de algoritmos es que permite reducir el consumo de energía en la comunicación, especialmente en el caso de la comunicación inalámbrica (Gawthrop, Gollee, & Loram, 2014).

El algoritmo MB-NCS se mejora al aplicar el concepto de retroalimentación intermitente. La idea general de la retroalimentación intermitente en NCS es mantener un sistema de control en modo de lazo abierto el mayor tiempo posible, para reducir el uso de recursos de comunicación y cerrar el lazo de retroalimentación durante un intervalo de tiempo finito, y así, recuperar un rendimiento deseado (Estrada, Lin, & Antsaklis, 2006).

Las desventajas de este algoritmo se relacionan con la retroalimentación intermitente en MB-NCS, ya que el sistema se ve significativamente afectado cuando hay retrasos inducidos por la red, procesos con dinámica no lineal e intervalos de actualización que varían con el tiempo (Montestruque, Garcia, & Antsaklis, Model-based control of networked systems, 2014)

En consecuencia, en los algoritmos de Control basado en Modelo de Sistema en Red, el estado de modelo se actualiza cada vez que se accesa a la red, esto

significa que la comunicación ocurre en instantes de tiempo discretos. Como se mencionó, uno de los inconvenientes de este tipo de control consiste en que existe una compensación inherente entre la utilización de la red y el rendimiento de circuito cerrado: el uso bajo de la red se produce a expensas del deterioro del rendimiento del control y viceversa (Sun & El-Farra, 2012), (Conte, Perdon, & Vitaioli, 2009).

2.2.3 Control multivariable

Otro control basado en modelos que se aplica en automatización en el campo de la manufactura es el control multivariable. El diseño y análisis del control multivariable se enfoca en el diseño de control de retroalimentación de múltiples entradas y múltiples salidas para sistemas lineales, combinado con los conceptos de control robusto que han surgido durante las últimas dos décadas (Skogestad & Postlethwaite, 2009).

Los sistemas de control de retroalimentación multivariable incluyen numerosos enfoques, que pueden clasificarse de la siguiente manera:

- Métodos de control de procesos de múltiples-entradas-múltiples-salidas (MIMO) derivado de la idea del control entrada-única-salida-única (SISO), configurando múltiples lazos de control como reguladores PID, desacoplando y creando jerarquías de control en cascada. (Albertos Pérez & Sala, 2004).
- Control tradicional centralizado, en el cual, todas las señales de control y los sensores se gestionan como un todo mediante operaciones matriciales, utilizando el análisis clásico del espacio de estados. Un ejemplo de este tipo de control es el control predictivo generalizado (GPC) que busca enfocarse en las variables de mayor impacto y da tolerancias a las menos relevantes (Skogestad & Postlethwaite, 2009), (Albertos Pérez & Sala, 2004).

Algunos ejemplos de aplicaciones en manufactura que emplean el Control Multivariable son: control de retroalimentación no lineal robusto, en procesos farmacéuticos; control lineal cuadrático gaussiano (LQG), en procesos de

molienda; filtrado Kalman, para el control automático del calibre de láminas y, filtros Kalman extendidos, para controlar procesos discontinuos en aplicaciones químicas y petroquímicas (Samad & Annaswamy, 2011). Otra opción de aplicación del control MPC basado en eventos utiliza la formulación de GPC para sistemas multivariables, esto permite una reducción significativa en la utilización de recursos (Pawlowski, Guzman, Berenguel, & Normey-Rico, 2015).

Se prevé un desarrollo importante en el control de procesos dentro de las siguientes áreas: control óptimo no lineal, estimación combinada de estados y parámetros para sistemas no lineales, análisis de robustez y métodos de síntesis robustos para sistemas no lineales, descomposición espacial, etc. (Jämsä-Jounela, 2007).

2.3 CONTROL ADAPTATIVO

El control adaptativo cubre un conjunto de algoritmos para el ajuste automático de los controladores en tiempo real, con el fin de mantener un nivel de rendimiento deseado cuando los parámetros del sistema son desconocidos y cambian con el tiempo (Shekhar & Sharma, 2018).

El sistema de control adaptativo puede interpretarse como un sistema de retroalimentación, donde, la variable controlada es el índice de desempeño. Así, el diseño de los sistemas de control adaptativo se enfoca en la eliminación del efecto de las perturbaciones de los parámetros en el rendimiento del sistema de control, a diferencia de los sistemas de control de retroalimentación convencionales, que están diseñados para eliminar el efecto de las perturbaciones en las variables controladas solamente (Ioannou & Fidan, 2006).

Las principales clases de técnicas de control adaptativo con aplicaciones a nivel industrial son: programación de ganancia, control adaptativo de referencia de modelo (MRAC) y reguladores de autoajuste (Astrom, 2014). Se han desarrollado numerosas técnicas adaptativas no lineales, que incluyen enfoques de reparametrización, linealización, retroalimentación no lineal y, en general, control adaptativo de sistemas no lineales inciertos en ciertas formas parametrizables. Los

más comunes de estos enfoques son la linealización de retroalimentación adaptativa, el retroceso adaptativo y la aproximación adaptativa (Tao, 2014).

El control adaptativo de referencia de modelo (MRAC) ha encontrado su principal aplicación industrial en el control de procesos. En este sentido, los enfoques MRAC no lineales, difusos y descentralizados son una tendencia emergente para la investigación. Sin embargo, uno de los inconvenientes del sistema adaptativo sigue siendo el modelo que se tiene que desarrollar, aunque este tipo de algoritmos y sistemas se adaptan, requieren tiempos de desarrollo e implementación significativamente largos, lo que conduce a que no se utilicen más que en aplicaciones muy críticas donde se justifique la inversión de tiempo y recursos (Astrom, 2014).

Para concluir, el desarrollo actual de los controles adaptativos, especialmente el control adaptativo multivariable, es una tendencia de investigación difundida. Los problemas para resolver en este tipo de control son: la mejora de la adaptabilidad de los controladores de retroalimentación a las incertidumbres del sistema, nuevas soluciones de control adaptativo en presencia de retrasos, control adaptativo de sistemas en red y control no lineal adaptativo (Tao, 2014).

2.4 CONTROL BASADO EN INTELIGENCIA COMPUTACIONAL

En los últimos años, con el desarrollo de la ciencia y la tecnología modernas, los algoritmos de inteligencia artificial están afectado cada vez más a la industria de la manufactura y a las personas en general, especialmente en el campo de control de la automatización industrial. Por ello, la tecnología de inteligencia artificial está desplazando a los sistemas de control industriales tradicionales basados en modelos.

Sobresale la tendencia en donde los algoritmos de aprendizaje automático, algoritmos evolutivos y el control inteligente se están integrado en sistemas automáticos de toda índole.

El control de sistemas dinámicos complejos, como los usados en los sistemas industriales de manufactura, se caracterizan por tener necesidades que no se cubren adecuadamente aplicando la teoría de los sistemas de control convencional como el MPC, esto es, la capacidad para lidiar con la incertidumbre del proceso y la habilidad para tomar decisiones similares a las humanas, incluido el razonamiento heurístico y el aprendizaje de experiencias pasadas.

Existen muchos métodos de aprendizaje, como los sistemas expertos, RNAs, el aprendizaje por refuerzo, los algoritmos miméticos, los algoritmos evolutivos, las máquinas de soporte de vectores (SVM) y las Maquinas de aprendizaje extremo (ELM) entre otros (Yang, 2020). Igualmente existen varios enfoques de control inteligente, incluido el control difuso, las redes neuronales, el algoritmo genético, la optimización de colonias de hormigas (ACO), las redes inmunes artificiales y la inteligencia de enjambre, que han logrado aplicaciones exitosas en muchos campos de la ingeniería (Zhou, Li, & Weijnen, 2015).

Debido a varias limitaciones en los sistemas de manufactura industrial, el sistema de control automático involucra típicamente supervisión o una interacción directa, tanto de operadores de máquinas como de humanos (Zhou, Li, & Weijnen, 2015). Otras limitaciones son la potencia computacional necesaria y los retraso en la computación en la nube, se vuelve difícil una asignación de recursos eficiente y justa, lo que afecta a las aplicaciones de automatización industrial basadas en Internet de las cosas (IoT). La gestión inteligente y precisa de recursos mediante inteligencia artificial (IA), se ha convertido en un foco de atención, especialmente en aplicaciones de automatizaciones de manufactura industrial (Sodhro, Pirbhulal, & Albuquerque, 2019).

Muchos investigadores se están centrado en el desarrollo de algoritmos para el control de robots humanoides, esto sistemas robóticos incluyen mecanismos de toma de decisiones y procesamiento de información, similares a las necesidades de los sistemas de control de manufactura industrial. En los sistemas de control robótico, debido a la incertidumbre detrás de la toma de decisiones y los procesos de información, se usan desde sistemas de control difusos hasta sistemas que imitan

los procesos biológicos cerebrales como los algoritmos recurrentes y los sistemas hipocámpicos (Zhang, Zeng, Zhao, & Wang, 2016).

Para facilitar el control basado en inteligencia computacional, los objetivos de control a menudo deben descomponerse en subtareas, que plantean la creación de controles compuestos de múltiples secciones o combinaciones de múltiples algoritmos de control independientes como una colectividad sincronizada (Kahraman, Deveci, Boltürk, & Türk, 2020).

El proceso de manufactura industrial es parecido al control robótico, ya que el modelado de procesos, control de equilibrio, navegación, planificación de rutas, control de dirección, optimización de eficiencia energética, toma de decisiones y aplicaciones móviles presentan un grado de complejidad equiparable al que requieren los procesos de manufactura, tanto en la complejidad como en la incertidumbre. El control robótico se basa mucho en la teoría de conjuntos difusos y los argumenta según el propósito de control específico, la teoría de los algoritmos desarrollada es aplicable a los procesos de manufactura (Kahraman, Deveci, Boltürk, & Türk, 2020).

Finalmente, los principales algoritmos más comúnmente utilizados en la automatización de manufactura industrial son: técnicas de sistemas expertos de control (ECS), técnicas basadas en sistemas difusos (FMC), redes neuronales artificiales (ANN) y algoritmos evolutivos (GA). Se incluyen dos clases de vanguardia que, aunque se aplican más a robótica, por sus características son ideales para el control de procesos de manufactura industrial, son las máquinas de aprendizaje extremo (ELM) y los sistemas de control basados en el hipocampo (HNC).

2.4.1 Sistemas expertos de control

El sistema experto de control está basado en reglas, puede procesar las variables, alarmas y eventos para mantener un nivel de respuesta predefinido (Souza, Neto, & Guedes, 2014). Constan de dos componentes principales, en el nivel superior tiene una sección del sistema que se basa en un conjunto de reglas y en el nivel inferior tiene un potente programa de síntesis que intenta encontrar un controlador

lineal que satisfaga la especificación de diseño del sistema. El programa de síntesis se basa en la teoría de factorización, donde el sistema de control lineal multivariable se traduce y se resuelve como un problema de programación lineal cuadrática (Tebbutt, 1994).

Por último, una combinación muy usada como sistema de control experto se denomina sistema experto difuso (FES) basado en reglas. Consiste en usar el sistema de control experto clásico y el control estadístico de procesos (SPC), ambos se combinan como un sistema de control difuso, para la detección de fallas como parte de los sistemas de control de calidad en los procesos de manufactura industrial (El-Shal & Morris, 2000).

2.4.2 Control difuso

Los sistemas de control en red no lineales basados en modelos difusos han logrado resultados importantes en la industria. Existen muchos tipos de algoritmos de control difuso o (*Fuzzy Model-Based Controllers*) como el modelo dinámico difuso T-S. Además, el control de lógica difusa puede considerarse como una especialización de los sistemas expertos convencionales basados en reglas (Qiu, Gao, & Ding, 2016).

Los algoritmos de control difuso consisten en un conjunto de reglas de control heurístico de la forma: 'si las condiciones lingüísticas se cumplen, entonces las consecuencias lingüísticas son válidas'. Conjuntos difusos y lógica difusa se utilizan, respectivamente, para representar términos lingüísticos y evaluar las reglas. Los valores lingüísticos de las reglas se modelan, mediante conjuntos difusos y funciones de pertenencia difusas. El control de lógica difusa es eficaz para sistemas complejos no lineales y definidos de forma imprecisa, para los que, las técnicas de control convencionales no son prácticas. Algunas desventajas del control lógico difuso residen en el hecho de que el diseño de la ley de control es heurístico (Qiu, Gao, & Ding, 2016).

Las principales áreas de aplicación del control difuso son actualmente los procesos químicos y petroquímicos, la industria de procesamiento de alimentos, así como los

reactores nucleares. Las aplicaciones industriales del control difuso se caracterizan por el uso de reglas heurísticas, que pueden ser una alternativa viable al control clásico (no difuso) para el control de procesos o sistemas mal definidos con no linealidades funcionales sujetos a modelos matemáticos difíciles (Wang, Liu, Qiu, & Feng, 2019).

De este modo, en comparación con el control convencional, el control difuso puede basarse fuertemente en la experiencia de un operador humano y, un controlador difuso (FC) puede modelar esta experiencia de manera lingüística (Zahabi & Kaber, 2019).

Así pues, el sistema de control difuso emplea un enfoque sin modelo, que incorpora habilidades humanas y experiencia en el diseño de control. La desventaja inmediata es, por supuesto, la falta de métodos de uso general. Sin embargo, se pueden encontrar numerosas aplicaciones en la automatización industrial. Algunos ejemplos de aplicaciones en manufactura industrial son: control de alimentadores de cinta de pesaje industrial, control de procesos de mecanizado, sistemas de seguimiento láser, moldeo por inyección de plástico, supresión de vibraciones, aplicaciones de robótica, aplicaciones de control de hornos, procesos de filtración, aire acondicionado, intercambiadores de calor o máquinas de forja, así como servo-sistemas y componentes mecatrónicos (Precup, 2011).

Una de las técnicas más extendidas de control difuso en la automatización industrial es la compensación distribuida en paralelo, la cual determina el modelo difuso del proceso no lineal como un conjunto de modelos lineales locales que se fusionan suavemente con la estructura del modelo difuso y se diseña un controlador local para cada modelo. El enfoque de control resultante puede garantizar la estabilidad y robustez del sistema de circuito cerrado, que establece condiciones suficientes en términos de un conjunto de desigualdades de matriz lineal (Precup, 2011).

Se han propuesto innumerables algoritmos de FMC adaptativos y predictivos en el control industrial, especialmente en procesos de la industria textil, la industria química y la petroquímica (Rajendran & Balu, 2010).

En síntesis, se puede resumir que el sistema de control difuso se compone de un conjunto de algoritmos de control que cuentan con un conjunto de reglas programadas por una persona con amplia experiencia en el proceso, creando un control heurístico con un enfoque sin modelo, que registra en las reglas que tratan de capturar las habilidades humanas y experiencia en el diseño de control. La desventaja de este sistema es la falta de método de uso general, siempre son aplicaciones de automatizaciones industriales puntuales.

2.4.3 Redes neuronales artificiales (RNA)

Las Redes neuronales artificiales o (RNA) son mapeos de entrada y salida no lineales, que emplean un modelo matemático para imitar redes neuronales biológicas. Este mapeo se genera por el enlace de unidades simples de procesamiento, altamente interconectadas llamadas neuronas, cada una de las cuales tiene dos tareas, agregación de las entradas de otras neuronas o del entorno, y generación de una salida que luego es alimentada a otras neuronas a través de enlaces ponderados. Las RNA pueden aproximar cualquier mapeo de control de entrada-salida con un grado dado de precisión, cuando se elige un número apropiado de neuronas. La aplicación de las RNA tiene dos fases, la primera fase es de aprendizaje, en la que se ajustan los parámetros y la estructura de la RNA, y la segunda fase es de prueba u operación, en la que opera la RNA entrenada. Existen muchos tipos diferentes de RNA (Yin, Niu, He, Li, & Lee, 2020).

Una clasificación tradicional de las RNA distingue entre redes retroalimentadas o no recurrentes y recurrentes, dependiendo de cómo se procesen los datos a través de la red neuronal. Otra forma de clasificar las RNA es por su método de aprendizaje, que las divide en redes supervisadas y no supervisadas (autoorganizadas) y, finalmente, otra clasificación se da por el refuerzo o aprendizaje gradual en que, la RNA recibe evaluaciones implícitas del conjunto de entrenamiento (Meireles, Almeida, & Simoes, 2003).

Las RNA se aplican cada vez más a los problemas de automatización de procesos, explotando su capacidad para aprender de los procesos físicos y su habilidad para

generalizar el conocimiento, mientras toman decisiones sobre datos imprecisos (Meireles, Almeida, & Simoes, 2003).

Desde el punto de vista industrial, las aplicaciones de RNA se pueden dividir en cuatro categorías principales: modelado e identificación, optimización y clasificación, control de procesos y, finalmente, reconocimiento de patrones (Meireles, Almeida, & Simoes, 2003). La categoría de modelado e identificación se refiere a la capacidad de las RNA para determinar una aproximación del modelo de proceso de entrada-salida, se ha aplicado en procesos químicos, sistemas de soldadura y procesos de laminación de placas. La categoría de optimización y clasificación incluye la planificación optimizada de tareas y la distinción de varias clases de eventos, materiales y objetos en forma funcional; se emplea en ámbitos de control de calidad o detección de fallas y en sistemas robóticos, particularmente para el control de manipuladores robóticos y para la detección de defectos en productos (Fan & Wang, 2008). La categoría de control de procesos se refiere a la aplicación de las capacidades de no linealidad, aprendizaje, procesamiento paralelo y generalización de RNA en control avanzado; incluyen la monitorización del rendimiento térmico, la regulación de la temperatura, el control de las celdas de producción, la predicción de la trayectoria en entornos robóticos y la predicción, a partir de las condiciones de funcionamiento (Tetko, Kůrková, Karpov, & Theis, 2019). Algunas estructuras RNA específicas se aplican al reconocimiento de imágenes (Tetko & Birukou, 2019).

Las RNA han migrado a áreas de automatización de manufactura gracias a los nuevos avances en la potencia de computo. Son empleadas en la clasificación y predicción de la calidad y seguridad de los alimentos, en prescripciones de campo para la aplicación de productos químicos de tasa variable entre muchas otras (Huang, 2009).

Finalmente, se concluye que existe una tendencia creciente en usar RNA para aproximar MPC no lineales, con lo cual se obtiene una buena precisión en el control automático en manufactura.

2.4.4 Algoritmos evolutivos

Los Algoritmos Evolutivos son métodos robustos de búsqueda y optimización, capaces de lidiar con dominios de problemas que contienen multimodalidad, discontinuidad, varianza temporal, aleatoriedad y ruido. Los más populares entre los Algoritmos Evolutivos son los denominados Algoritmos Genéticos (GA) (Yin, Niu, He, Li, & Lee, 2020), son algoritmos de optimización metaheurística, basados en poblaciones que utilizan mecanismos inspirados en biología y la supervivencia de la teoría más apta, para refinar un conjunto de soluciones de forma iterativa (Rahul, Narinder, & Yaduvir, 2011). El valor de la función objetivo asociada con cada solución en la población, controla el esquema de selección iterativo, de modo que las soluciones con alta aptitud tienen una alta probabilidad de aparearse y formar la población de descendientes. Una vez que se selecciona la población de descendientes, se aplican aleatoriamente operadores especiales que simulan el cruce genético y la mutación, para crear nuevas soluciones. Luego, estos pasos se repiten para obtener nuevas soluciones con mayor aptitud hasta que se cumpla un criterio de parada (Haupt & Haupt, 2004), (Ao, 2013).

Los Algoritmos Genéticos y otros algoritmos evolutivos, como la programación genética, se aplican ampliamente al diseño fuera de línea de controladores, particularmente a la optimización de parámetros y la síntesis automática de la topología del controlador (Iglesias, Lowen, & Verschoren, 2008).

En general, las aplicaciones de los Algoritmos Evolutivos para la automatización de manufactura y el control industrial están motivadas por su capacidad para ajustar automáticamente los parámetros del controlador, sin requerir operadores expertos (Haupt & Haupt, 2004).

2.4.5 Algoritmos extreme learning machine (ELM)

En la rama de la Inteligencia Artificial, se emplea una gran variedad de técnicas y algoritmos, entre ellos están las RNA's, que juegan un papel muy importante en los sistemas que requieren aprendizaje. Sin embargo, un punto débil de las RNA's es su lenta velocidad de aprendizaje. Recientemente, las Máquinas de Aprendizaje

Extremo o ELM han atraído la atención de muchos investigadores como una nueva técnica que resuelve el problema de la velocidad de aprendizaje. La técnica ELM trabaja para Redes Monocapa hacia Delante o SLFN. La esencia de las ELM es que la capa de neuronas ocultas de la red no necesita ser entrenada, ya que utiliza un método de aprendizaje analítico en lugar de iterativo. Comparadas con las técnicas de Inteligencia Artificial tradicionales, las ELM mantienen una buena capacidad de generalización con una velocidad rápida de entrenamiento (Huang G.-B. , 2014).

Una vez entrenadas, las ELM funcionan igual que las SLFN (Souza, Neto, & Guedes, 2014). El algoritmo ELM, propuesto por Huang et al. se caracteriza por calcular los parámetros de forma analítica (Huang G.-B. , What are Extreme Learning Machines? Filling the Gap Between Frank Rosenblatt's Dream and John von Neumann's Puzzle, 2015).

Huang demostró que se pueden asignar aleatoriamente los pesos y sesgos de la capa oculta, sin afectar la capacidad de la red de aprender las distintas observaciones relevantes (Xu & Wang, 2016). Esto implica que no es necesario ajustar todos los parámetros de la red. El algoritmo ELM se basa en este principio para ajustar únicamente los parámetros de la capa de salida (Mesquita, Gomes, Rodrigues, & Galvao, 2015).

2.4.6 Aprendizaje por refuerzos

El Aprendizaje por Refuerzos (RL, *Reinforcement Learning*), se describe como la mejora del desempeño mediante prueba y error. La idea de que aprendemos interactuando con nuestro entorno, es probablemente lo primero que se nos ocurre cuando pensamos en el aprendizaje del ser humano. De niños se juega, se agitan los brazos o se mira alrededor, no se tiene un maestro explícito, pero se tiene una conexión sensorimotora directa con el entorno (Barto, 1994). El aprendizaje por refuerzos proporciona un conjunto de técnicas para solucionar problemas relacionados con decisiones secuenciales. Por ejemplo, si tenemos un sistema mecánico, cuyo estado en cualquier instante k , está descrito por una dimensión s .

Dicha magnitud puede ser un vector, cuyos componentes contienen información de diversa índole, por ejemplo: coordenadas de localización, velocidad, peso, suma de productos almacenados y velocidad de manufactura. En tanto el lapso del tiempo pasa, el estado del sistema puede modificarse, es decir, que el valor de s puede sufrir cambios (Escandell-Montero, Martínez-Martínez, & Martín-Guerrero, 2014).

Este tipo de problema lo encontramos en una amplia variedad de aplicaciones, tales como: control automático, manejo autónomo de automóviles, automatismos de producción industrial, y la medicina entre otros (Sutton & Barto, 2018).

Podemos decir que los cambios están influenciados, por las acciones que se realizan sobre el sistema y, la meta es lograr un estado determinado.

Cuando se tiene que resolver qué acción emplear en un intervalo de tiempo, se dice que el problema de decisión es mono etapa; si se requiere una secuencia de decisiones, entonces se trata de un problema de decisión multietapa o secuencial (Escandell-Montero, Martínez-Martínez, & Martín-Guerrero, 2014).

Existen diferentes maneras de resolver un problema de decisión secuencial, la primera consiste en programar un sistema que sea capaz de dar la acción adecuada a todos los estados posibles. Sin embargo, para lograr esto, se debe programar una acción específica para cada estado que contemple a todos sin excepción, ya que de faltar uno, ese estado quedaría con una acción indefinida o se detendría. Este tipo de solución puede ser viable en problemas sencillos, pero no resulta útil en problemas en los que el número de estados es elevado. Otra alternativa consiste en utilizar técnicas de sondeo y planificación; cuando se conoce de antemano la forma en la que evoluciona el estado del sistema, en función de las acciones aplicadas, entonces es posible buscar o planificar la mejor secuencia de acciones desde el estado actual. En caso de que exista cierta incertidumbre sobre el resultado que pueden provocar las acciones, los algoritmos estándar de sondeo y planificación dejan de ser válidos. Otra posibilidad es la solución basada en aprendizaje; esta opción, que es la planteada en un RL, no

requiere información previa del sistema, sino que emplea un conjunto de datos (estados y acciones) para obtener la solución (Sutton, Szepesvári, & Maei, 2008).

Muchos de los algoritmos de Aprendizaje por Refuerzos se basan en el aprendizaje *Temporal Difference*, ya que es el tipo de algoritmo más extendido y ampliamente estudiado. Su objetivo es encontrar una política que maximice la recompensa obtenida por el agente (Tsitsiklis & Van Roy, 1997).

La principal característica de los algoritmos RL es que están basados en la experiencia.

Hay que resaltar que, dado que la política se obtiene a partir de las muestras que adquiere el agente, es necesario que el proceso de muestreo cumpla con ciertas propiedades estadísticas, para asegurar que la política resultante sea óptima.

La Figura 4 muestra el esquema de un Sistema de Aprendizaje por Refuerzos, se observan los dos elementos principales que son: el agente, que modifica al entorno mediante actuadores y el entorno o ambiente, que nos da la información de lo que ocurre en el sistema que queremos controlar.

Existen flujos de información básica, como la recompensa a cada acción ejecutada R_t , la acción a ejecutar A_t y el estado del ambiente después de la acción S_t .

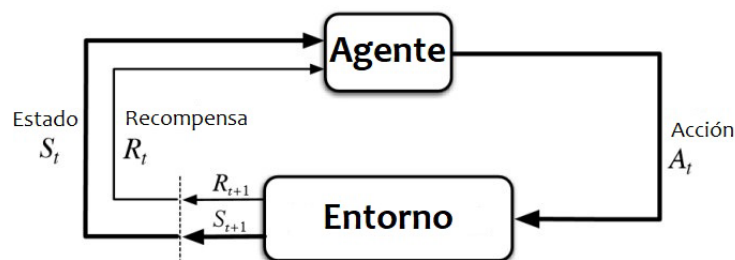


Figura 4. Esquema del sistema de aprendizaje por refuerzos. (Otterlo & Wiering, 2012).

2.4.7 Técnicas integradas de inteligencia computacional

Una de las tendencias emergentes en los sistemas de control automático que se aplica a la automatización en la industria manufacturera, es la combinación de varias técnicas de inteligencia computacional en un mismo algoritmo de control para lograr que sea eficiente, de hecho, la aplicación combinada de lógica difusa, redes neuronales y algoritmos evolutivos con fines de control, permite realizar el aprendizaje en una variedad de aplicaciones, incluida la Manufactura industrial (Yang, 2020), (Feng, 2018).

Hay varias ventajas en la combinación de redes neuronales o algoritmos evolutivos con el control difuso, por un lado, el control neuro-difuso permite explotar las capacidades de aprendizaje de las redes neuronales, superando la complejidad en la estructura; por otro lado, los algoritmos evolutivos combinados con la estructura de control difuso permiten la adaptación de la estructura y los parámetros de los sistemas difusos (Saridakis, Dentsoras, Darzentas, Vosinakis, & Arnellos, 2008).

La combinación de los algoritmos genéticos y redes neuronales aumenta la capacidad de aprendizaje de dichas redes en dominios complejos, con el inconveniente de una convergencia más lenta. Estas técnicas integradas son denominadas computación blanda, que representa un enfoque más poderoso y flexible que aprovecha las ventajas de cada metodología (Yin, Niu, He, Li, & Lee, 2020). Además, el control neuro-difuso presenta un rendimiento mejorado gracias a las capacidades de aprendizaje y las capacidades de procesamiento paralelo de las RNA (Precup, 2011). Así, la mayoría de los enfoques de control neuro-difuso abordan uno de los siguientes: usar RNA para sintonizar sistemas difusos, extraer reglas difusas de datos numéricos dados o desarrollar sistemas híbridos que combinen RNA y sistemas difusos en varias formas (Kamruzzaman, Begg, & Sarker, 2006).

Un enfoque neuro-difuso es el algoritmo ANFIS, una clase de redes adaptativas que son aproximadores universales de funciones no lineales y son funcionalmente equivalentes a los sistemas de inferencia difusos; algunos ejemplos del este tipo de control en la industria manufacturera es el control de proceso de molienda

complejo (Jain & Martin, 1999), el control de defectos mediante una metodología bayesiana-difusa para sistemas de fabricación que producen una gran variedad de piezas (Stewart, Cheraghi, & Malzahn, 2004), el monitoreo del estado de herramientas de corte de planicidad mediante un RNA de retropropagación controlada difusa, el modelado de procesos y la optimización de la dispensación de epóxido para realizar la encapsulación de microchip mediante la técnica de regresión difusa (Kwong & Bai, 2005).

El enfoque más común para la integración de algoritmos de control difuso con algoritmos evolutivos es que pueden utilizarse para superar la necesidad de especificar algunos parámetros por parte de un diseñador humano, es decir, para generar la base de reglas difusas y/o ajustar la membresía asociada con parámetros de función de forma automática (Dotoli, 2016).

Un ejemplo de control usando la técnica integrada de inteligencia computacional es el proceso industrial de dispensado de pasta de soldadura, el cual puede controlarse al usar un enfoque de regresión difusa basado en programación genética para modelar el proceso de fabricación (Chan, Kwong, & Tsim, 2010).

Los recientes avances en algoritmos de Deep Learning, Extreme Learning Machines y CNN son una variante de las RNA que al combinarse con algoritmos difusos, algoritmos evolutivos o de aprendizaje por refuerzos, permiten crear algoritmos híbridos que combinan las propiedades para hacer aplicaciones de control más eficientes basadas en inteligencia computacional (He, Shao, Zhong, & Zhao, 2020), (Reddy, Bhattacharya, & Rishita, 2018).

Finalmente, los controles con algoritmos combinados han recibido gran atención en el área industrial para aplicaciones de control de procesos y aplicaciones de planificación y programación en la fabricación. En el contexto de las técnicas de inteligencia computacional integradas, existen innumerables aplicaciones potenciales de control industrial.

2.5 SISTEMA COGNITIVO BASADO EN RED NEURONAL

Durante las dos últimas décadas, el interés de la comunidad científica en desarrollar Máquinas de Aprendizaje basadas en Redes Neuronales ha crecido de forma espectacular, lo que ha dado lugar a un gran número de algoritmos nuevos y nuevas aplicaciones para ellos (van Heeswijk & He, 2015).

En este sentido, podemos encontrar algoritmos de aprendizaje no supervisado que, combinados con técnicas de control adaptativo resultan sumamente atractivos para aplicaciones industriales (Saridakis, Dentsoras, Darzentas, Vosinakis, & Arnellos, 2008); ya que permiten controlar procesos de forma autónoma, optimizar la productividad, reducir la necesidad de un monitoreo constante, así como mejorar los tiempos de respuesta de los sistemas de control a las fluctuaciones de las materias primas o variaciones inesperadas en el proceso. La mejora significativa en la velocidad de procesamiento de las computadoras actuales, la interconectividad con redes y servicios, la creciente capacidad de almacenamiento de datos y la gran disponibilidad de sensores para monitorear múltiples parámetros, crean un marco ideal para la aplicación de los Sistemas de Control basados en Redes Neuronales. Éstos al ser combinados con secuencias de control complejas y automatismos, logran dar a los procesos industriales la capacidad de toma de decisiones de forma autónoma, permitiendo un control robusto del proceso y mejorando su productividad y competitividad.

Un algoritmo de aprendizaje que recientemente ha ganado gran interés en diversos campos de investigación, tanto teóricos como prácticos, es la Máquina de Aprendizaje Extremo (*Extreme Learning Machine*) o ELM (Huang G.-B. , An Insight into Extreme Learning Machines: Random Neurons, Random Features and Kernels, 2014). Las ELMs se basan en Redes Neuronales Artificiales del tipo Monocapa de Avance hacia Adelante o SLFN (*Single Layer Feedforward Network*), con un novedoso método de aprendizaje que permite calcular los parámetros de la red en forma analítica en lugar de iterativa; lo que ha permitido que la Máquina de Aprendizaje Extremo sea uno de los algoritmos más rápidos que existen con respecto al aprendizaje (Xu & Wang, A fast incremental extreme learning machine algorithm for data streams classification, 2016). El aprendizaje analítico hace que se

requiera menor cantidad de datos en comparación a los métodos iterativos (Yin, Li, & Hu, 2014). Gracias a su capacidad para aprender rápidamente puede ser aplicado a procesos industriales, ya que resulta altamente beneficioso, al contribuir a evitar grandes pérdidas de material de desperdicio o de tiempo durante la fase de entrenamiento (Hussain, Siniscalchi, Lee, & Wang, 2017).

2.6 SISTEMA DE CONTROL ROBÓTICO COGNITIVO

En la robótica existe una tendencia en el desarrollo de controles automáticos, los denominados *Sistemas Robóticos Cognitivos*, que se enfocan en desarrollar automatismos con inteligencia sensomotora, utilizando una arquitectura de control basada en múltiples algoritmos de monitoreo; también conocidos como agentes, y un control inteligente inspirado biológicamente, como lo muestra Kawamura et al. (Kawamura & Gordon, 2006), que presenta un algoritmo que dota a un robot de auto conocimiento de su entorno, gracias al uso de una memoria episódica a corto plazo (Kawamura, Dodd, Ratanaswasd, & Gutierrez, 2005).

Los sistemas de control cognitivo se enfocan por lo general, en la memoria a corto plazo, para el aprendizaje del ambiente que rodea al robot; en la memoria a largo plazo, para el aprendizaje del comportamiento y en la ejecución de tareas, usando memoria de trabajo y aprendizaje automático.

Los sistemas de control cognitivos pueden ser muy diversos, el tipo de variante depende principalmente del tipo de Red Neuronal, estructura cerebral o funcionamiento biológico que se desea emular.

El enfoque de las investigaciones actuales de los Sistemas Robóticos Cognitivos se centra en desarrollar nuevos métodos de procesamiento de conocimiento, es decir, en los métodos para adquisición de conocimiento, representación, razonamiento y aprendizaje; así como en aplicar mecanismos de percepción y acción basados en Inteligencia Artificial a robots autónomos (Anireh & Osegi, 2017).

El creciente desarrollo de robots autónomos con habilidades suficientes para realizar tareas complejas de manipulación y reconocimiento del entorno hace que

los algoritmos de procesamiento sean cada vez más importantes, porque hay más necesidad de que dichos robots adquieran nuevo conocimiento de los diferentes tipos de tareas a realizar de forma autónoma.

2.7 APRENDIZAJE AUTOMÁTICO COGNITIVO

Dentro del campo de la Inteligencia Artificial se ha buscado, en años recientes, desarrollar algoritmos que posean comportamientos inteligentes, que demuestren que son capaces de representar conocimiento del mundo real.

Los algoritmos inteligentes se caracterizan por emular procesos de razonamiento, planificación, aprendizaje y toma de decisiones al emplear dichas aproximaciones para alcanzar diversos objetivos (Schmidhuber, 2015).

El aprendizaje automático es un área que ha atraído el interés de gran cantidad de investigadores, lo que ha contribuido a lograr importantes avances en otras áreas del conocimiento relacionadas, como el desarrollo de buscadores web, la minería de datos, y la robótica cognitiva. Los métodos de procesamiento del conocimiento son un recurso importante para los robots, permitiendo que cumplan con tareas desafiantes en entornos complejos y dinámicos (Tenorth, Jain, & Beetz, 2010).

El Aprendizaje Máquina, a diferencia de la programación convencional, no tiene una estructura explícitamente predefinida sobre cómo procesar los datos para obtener un resultado específico. En el Aprendizaje Máquina, mediante el análisis de datos los algoritmos, descubren automáticamente información relevante y la almacenan, para poder generar los resultados esperados. El Aprendizaje Máquina ha tenido un gran auge en los últimos años, debido principalmente a los avances en la tecnología, que han facilitado la adquisición de una gran cantidad de datos mediante diversos apoyos entre los que destacan sensores distribuidos o el Internet de las cosas *IoT* (por sus siglas en inglés). También ha sido impulsado fuertemente por las tendencias de la industria 4.0 que, conectados a redes de comunicación y servidores, nos permiten tener acceso a las condiciones instantáneas de procesos y por consiguiente, a grandes volúmenes de información y complejidad de datos.

El principal problema en la interpretación de la información recolectada es encontrar su correcta modulación, para poder clasificarla y entender el significado de esos datos dentro del sistema. Por ejemplo, en algunos casos no es posible determinar el modelo de predicción con precisión, debido a los cambios de las variables en los procesos. Una alternativa consiste en emplear Redes Neuronales Artificiales, las cuales son capaces de aproximar sistemas muy complejos sin tener que conocer a detalle cómo es la función de transferencia o el modelo de dicho sistema, requiriendo sólo los datos de entrada y salida de este.

Muchos de los modelos de Redes Neuronales Artificiales utilizados hoy en día se inspiraron en modelos biológicos, principalmente en el cerebro que es sin duda, el órgano más complejo de los mamíferos. El cerebro se compone de diversas partes estructurales, que realizan funciones diferentes y trabajan de forma coordinada a través de miles de conexiones que se establecen entre ellas y con el resto del cuerpo.

Algunos de los modelos de Redes Neuronales Artificiales, imitan el funcionamiento del hipocampo como una de las estructuras claves del cerebro, debido a su relación con los procesos de memorización y aprendizaje, como lo demuestran Treves y Rolls (Treves & Rolls, 1994) en sus modelos computacionales de recuperación de memoria y auto asociación.

2.8 RED HIPOCÁMPICA BIOLÓGICA COMO CONTROL COGNITIVO

El cerebro humano y su operación interna sigue siendo, hoy en día, una caja negra que poco a poco se va descubriendo, sin embargo, en los últimos años los investigadores han logrado obtener avances significativos, permitiendo tener un mejor entendimiento sobre cómo funciona la formación hipocámpica del cerebro biológico. Por ejemplo, las investigaciones de Knierim (Knierim & Neunuebel, 2016) y Zammit (Zammit, Ezzati, Zimmerman, & Lipton, 2017). El creciente entendimiento sobre su funcionamiento ha permitido crear nuevas y variadas Redes Neuronales Artificiales inspiradas en la formación hipocámpica que tratan de aproximar la

naturaleza de la inteligencia del cerebro biológico a través de una simulación precisa, tanto de las estructuras internas como de sus funciones.

Los controles cognitivos buscan generar de alguna manera el aprendizaje y la memoria, que son esenciales para todos los mamíferos, los cuales tienen varios tipos de regiones cerebrales que están relacionadas con diferentes tipos de memoria. El hipocampo es una de las regiones esenciales del cerebro y es fundamental en el proceso de convertir la memoria de corto plazo a memoria de largo plazo.

Se han propuesto muchos modelos acerca de la formación hipocámpica (HF), con la meta principal de representar con precisión los procesos internos. Recientemente se han presentado trabajos de investigación que buscan una aplicación práctica a los controles, que basan su funcionamiento en simular el comportamiento de la formación hipocámpica, principalmente la navegación o control de guía de avance para robots, reconocimiento de patrones como imágenes o sonidos y otras aplicaciones que requieren memorización.

Existen varios trabajos de investigación que han propuesto redes hipocámpicas como la red AHA! (Algoritmo Artificial Hipocámpico) propuesta por Gideon Kowadlo et al. (Kowadlo, Ahmed, & Rawlinson, 2019); cuya característica principal es que no requiere de etiquetas externas para entrenar al modelo o parámetros de la RNA (Red Neuronal Artificial). Este algoritmo fue aplicado con éxito en el reconocimiento de palabras mediante una aplicación web llamada Omniglot, que es una enciclopedia de sistemas de escritura e idiomas que se usa para aprender idiomas, alfabetos y otros sistemas de escritura. AHA! es un modelo que utiliza 3 instancias o formaciones hipocámpicas: la DG, la CA1 y la CA3; si comparamos este modelo con el propuesto en esta investigación (Kowadlo, Ahmed, & Rawlinson, 2019).

Otra red hipocámpica fue propuesta por Tielin Zhang et al. (Zhang, Zeng, Zhao, & Wang, HMSNN: Hippocampus inspired Memory Spiking Neural Network, 2016) llamada "*Hippocampus inspired Memory Spiking Neural Network*" HMSNN que se desarrolló para la navegación de un robot caminante y también fue probada para reconocer imágenes de escritura. Este modelo se basó en el funcionamiento

neuronal denominado "*Neuron-Leaky Integrate & Fire Model*", que permitió almacenar en memoria imágenes del medio ambiente, se enfocó en mimetizar las formaciones hipocámpicas *DG* (Dentate Gyrus) y *CA3* (Cornu Ammonis 3) (una explicación más detallada de los términos *DG* y *CA3* se encuentra en el capítulo 3).

Otra Red Neuronal inspirada en la formación hipocámpica fue propuesta por A. Revel et al. (Revel, Gaussier, & Banquet, 1999) usando un modelo denominado "Place Cell", que permite recuperar un evento espacial o de localización para la navegación de un robot.

Existen otras investigaciones enfocadas en la aplicación o experimentación dentro de la rama médica, que no son incluidas en este trabajo. Resulta relevante mencionar que, al momento de realizar la presente investigación, no se encontró ninguna evidencia de la aplicación de RNA inspiradas en la formación hipocámpica biológica para el control de procesos de manufactura.

2.9 CONCLUSIONES DE LA REVISIÓN LITERARIA

El capítulo 2 muestra el de análisis de la literatura para la revisión literaria en sistemas de control automático de procesos de manufactura. El objetivo es direccionar la investigación para el control automático de procesos en manufactura complejos, con alta incertidumbre que dificulta el modelado y son multivariable.

El análisis consiste en el estudio de las principales técnicas de control avanzado que se adoptan actualmente en la automatización en manufactura industrial. En particular, se discuten las principales contribuciones teóricas, aplicaciones, problemas abiertos y desafíos, clasificando los enfoques en tres clases principales: control basado en modelos, control adaptativo y control basado en inteligencia computacional.

La tendencia de los sistemas de control busca hacerlos más eficientes cuando se aplican a sistemas a gran escala, complejos e inciertos. Esto es necesario para una buena integración en a los procesos de manufactura.

La mayor parte de los enfoques actuales de control demandan mejoras significativas para que sean aplicables a manufactura, en especial cuando se utilizan arquitecturas descentralizadas y distribuidas.

Al respecto, se deben adaptar o redefinir varios métodos para tener en cuenta las crecientes exigencias de hacer sistemas que sean más autónomos, capaces de autoorganizarse y reconfigurarse, garantizando simultáneamente la fiabilidad y la reducción del consumo energético. Como se discutió en la literatura las estrategias basadas en MPC son uno de los enfoques que se aplica más extensamente en la actualidad para resolver problemas de control en sistemas de automatización industrial. Sin embargo, para aumentar aún más su efectividad y aplicabilidad en la automatización de procesos de manufactura, es importante reducir los costos y tiempos de modelado. Debido a que se basa en modelos, su aplicabilidad está limitada a la habilidad de los programadores para modelar apropiadamente los sistemas, por lo que no es una buena opción en donde existe incertidumbre o no hay un modelo a utilizar.

Al revisar el enfoque de control difuso se observó que también ofrece soluciones eficientes, pero está limitado por la falta de métodos de propósito general. Además, definitivamente requiere técnicas de aprendizaje y ajuste en tiempo real más eficientes y sistemáticas que las que se disponen actualmente. Por otra parte, en el área de control adaptativo, la investigación actual está dedicada al desarrollo de enfoques más eficientes que también se apliquen a sistemas multivariable con retardos de tiempo y sistemas de control en red, para hacerlos más útiles en las aplicaciones industriales actuales. El control adaptativo es candidato para sistemas de control de manufactura simples.

En el área del control basado en la inteligencia computacional, es muy prometedora la investigación de las Técnicas integradas de inteligencia computacional, que permiten la aplicación combinada de lógica difusa, redes neuronales en modalidades como redes auto organizables, redes de aprendizaje supervisado, aprendizaje por refuerzo entre muchas otras. Sin embargo, el mayor impedimento para su desarrollo eficiente en el área de automatización de proceso

industriales está relacionada con los largos tiempos de entrenamiento necesario para el funcionamiento de la RNA's.

Por último, las máquinas de aprendizaje extremo son una solución eficiente para eliminar el principal problema de las RNA en el ambiente de manufactura, gracias a que no requieren entrenamientos iterativos si se combina con las Técnicas integradas de inteligencia computacional pueden aplicarse y mejorar la autonomía de los algoritmos, que sean capaces de autoorganizarse y reconfigurarse, garantizando simultáneamente la fiabilidad y la reducción del consumo energético.

3. PROCEDIMIENTO

3.1 DESARROLLO DEL ALGORITMO HÍBRIDO AHM-ELM PARA CONTROL DE PROCESOS AUTOMÁTICOS EN MANUFACTURA

3.1.1 Introducción a desarrollo del nuevo algoritmo híbrido AHN-ELM

La presente sección 3.1 explica el funcionamiento y las partes que componen al Algoritmo Híbrido AHM-ELM propuesto en esta investigación. Se presentan sus características principales, incluyendo su capacidad para manejar las memorias episódica, de corto plazo y de trabajo, a semejanza del hipocampo biológico. Muestra el proceso de integración de los algoritmos, supervisados y no supervisados para gestionar las memorias y crear la Red Hipocámpica AHN-ELM mediante la Máquina de Aprendizaje Extremo (ELM).

También se explican las cuatro diferentes instancias hipocámpicas neuronales, relevantes a emular: la Dentate Gyrus (DG), Cornu Ammonis 3 (CA3), Cornu Ammonis 1 (CA1), Subiculum (SUB) y, por último, la sección de interfaz, que es la Corteza Entorrinal (EC), usadas para simular el manejo de la memoria del hipocampo biológico. Como referencia hay dos gráficas que explican las conexiones de las instancias hipocámpicas tanto en el ámbito biológico como el que se usa en el algoritmo, son las Figura 7 y Figura 8.

La importancia de esta sección reside en que, explica qué es y cómo funciona el Algoritmo Híbrido de la Red Neuronal AHN-ELM como solución propuesta a la pregunta que guía a esta investigación.

La Figura 5 basada en los trabajos de Kawamura et al, Treves et al, y Weisz et al (Kawamura & Gordon, 2006), (Treves & Rolls, 1994) y (Weisz & Argibay, 2012) muestra las 4 instancias principales y la interconexión de las instancias hipocámpicas.

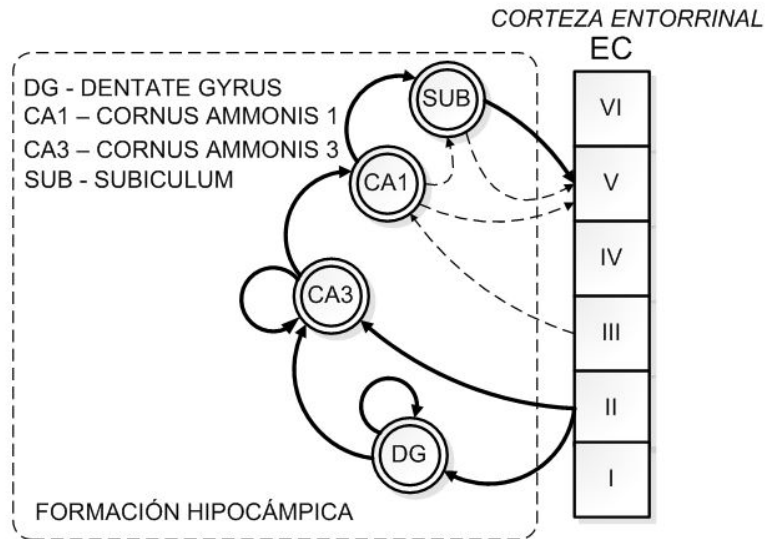


Figura 5. Modelo hipocámpico de aprendizaje con memoria de corto y largo plazo.
Elaboración propia.

3.1.2 Modelo estructural de la memoria

El Modelo Estructural de la Memoria utiliza secciones en las que se almacenan diferentes tipos de información. La memoria se divide en tres tipos principales: memoria sensorial, memoria a corto plazo, y memoria a largo plazo. La memoria sensorial es la que tiene acceso a la zona sensorial y contiene información del entorno como el registro de olores que obtenemos por medio del olfato. La memoria a corto plazo contiene información filtrada del entorno, así como las respuestas o actuaciones del sistema que afectan al entorno; tiene interacción con la zona de actuación, en la que se procesa la información recibida e interactúa con el medio ambiente. La memoria denominada de largo plazo o de recuerdos episódicos, almacena permanentemente datos específicos del entorno, como colores, aromas, situaciones y personas relacionadas con un recuerdo específico.

La Red Neuronal Hipocámpica AHN-ELM está inspirada en el funcionamiento de las memorias: sensorial, de corto y largo plazo, creando recuerdos episódicos.

La Figura 6 es un diagrama a bloques del modelo estructural de Atkinson y Shiffrin, que presenta las memorias sensorial, de corto y largo plazo (Mormu, 2007), muestra el mecanismo de memoria de corto y largo plazo y su relación con la memoria

sensorial del cerebro. Puede verse como los estímulos son recibidos en la memoria sensorial, de ahí solo algunos patrones llegan a la memoria de corto plazo, y finalmente, mediante auto-asociación por repetición se envían patrones a la memoria de largo plazo. Los estímulos o las señales del entorno llegan a la memoria sensorial después de eliminar el ruido ambiental, es decir, cuando se pone atención a cierta información en particular. La información de la memoria sensorial es trasladada a la memoria de corto plazo cuando se pone atención, y estará ahí durante un cierto tiempo, antes de ser olvidada. También los estímulos serán olvidados después de un tiempo. En la memoria de largo plazo se produce un fenómeno que consiste en el almacenamiento de memorias de corto plazo repetitivas, en el diagrama se le denomina repaso. Si no hay repetición, esas memorias pueden ser olvidadas. Otro fenómeno que ocurre en la memoria de largo plazo es la recuperación de un recuerdo episódico, consiste en recuperar un recuerdo almacenado en la red neuronal al comparar una referencia proveniente del ambiente, que, aunque no sea exacto al patrón almacenado posiblemente por contener ruido o distorsiones, logra disparar la salida del patrón almacenado en la memoria de largo plazo.

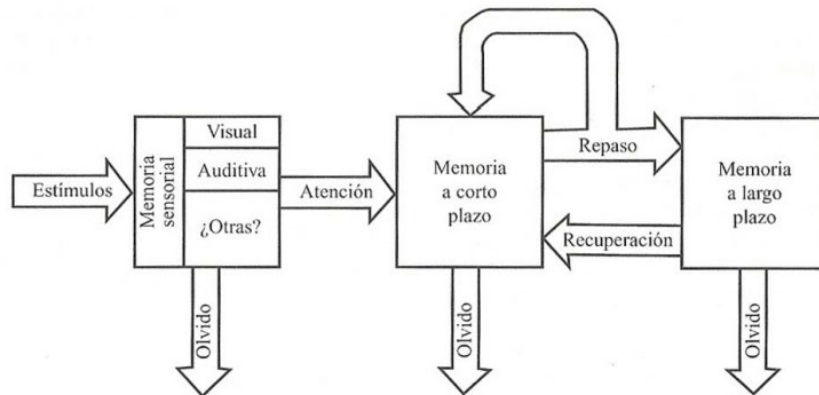


Figura 6. Diagrama a bloques del modelo de la memoria de Atkinson y Shiffrin. (Santalla Peñaloza, 2000).

La memoria de largo plazo tiene tres procedimientos principales: el proceso de almacenaje de patrones episódicos, el proceso de recuperación de patrones episódicos y el proceso de almacenaje de patrones similares. El término de patrón

episódico se define como la información proporcionada por los sensores del proceso que se está controlando y representa el estado actual del mismo. Los términos de almacenaje y recuperación se relacionan, puesto que cuando se identifica un patrón en la Red Neuronal, éste no será almacenado, sino que solamente será recuperado, y en caso de no existir previamente, es cuando se procede al almacenaje. Cuando hay dos patrones que son muy similares entre sí, se realiza un proceso adicional para poder diferenciarlos, dentro del proceso de repaso.

3.1.3. Instancias hipocámpicas

El proceso de acceso a la memoria de corto o de largo plazo, es controlado, en el cerebro biológico, por la estructura neuronal del hipocampo. Esta estructura tiene un rol fundamental en el control de los recuerdos episódicos y el manejo de estímulos en la memoria de corto plazo.

Para aproximar el funcionamiento de la gestión de la memoria del hipocampo, la Red Neuronal Hipocámpica AHN-ELM utiliza cuatro formaciones o instancias hipocámpicas y una interfaz con el entorno, a las que denominaremos de aquí en adelante como instancias hipocámpicas DG, CA1, CA3 y SUB.

Cada una de estas instancias hipocámpicas se convierte en un algoritmo, que, al ser integrados en la Red Neuronal Hipocámpica AHN-ELM, permiten procesar la información proveniente de la sección sensorial y utilizarla para controlar un sistema actuador, gestionando una memoria a lo largo de dicho proceso.

La Figura 7 muestra un esquema de las rutas neuronales que utiliza el hipocampo biológico, que sirve como guía y base para crear las rutas de interconexión en la Red Neuronal Hipocámpica AHN-ELM. En la figura se resaltan las entradas II/IV, III/V y las salidas LEC y MEC que conectan al mundo exterior. Las principales instancias que forman el hipocampo se resaltan de la siguiente forma: en azul la instancia CA3, en rojo la instancia DG, en rosa la instancia CA1 y en verde la instancia SUB. Las rutas van principalmente a las instancias Dentate Gyrus, Cornu Ammonis y Subiculum. La instancia hipocámpica DG o Dentate Gyrus en rojo es la entrada principal para los

conjuntos de datos proveniente de la corteza entorrinal, marcada en la figura con las flechas II/IV y III/V.

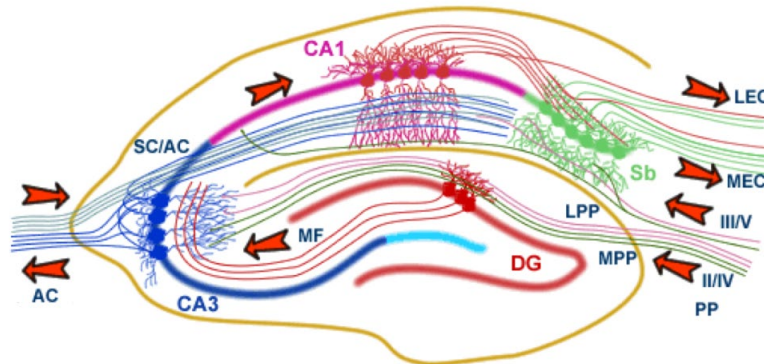


Figura 7. Esquema de las rutas neuronales que existen dentro de la formación hipocámpica biológica. (Murmu, 2007).

La estructura Dentate Gyrus se relaciona con la separación de patrones, en tanto crea ligeras diferencias para poder distinguir entre dos patrones similares.

La forma en que realiza esto es expandiendo la respuesta neuronal recibida de la corteza entorrinal y creando un patrón extendido, que utiliza un mayor número de neuronas respecto al número de estímulos recibidos. Esta es una forma de codificación extendida. Por otro lado, la instancia hipocámpica CA3 o Cornu Ammonis 3, mostrada en color azul en la figura, recibe un conjunto de datos dispersos proveniente de la instancia Dentate Gyrus.

La instancia hipocámpica CA3 genera una auto asociación del conjunto de datos al crear una memoria fundamental que queda almacenada en esta red neuronal.

La instancia CA1 o Cornu Ammonis 1, en color rosa, es la responsable de consolidar la información procesada en la instancia hipocámpica CA3 y la información de la instancia hipocámpica Subiculum SUB. Este proceso se conoce como completar patrones debido a que junta la información de salidas del entorno por el Subiculum y las entradas de información episódica por la instancia hipocámpica CA3.

La instancia hipocámpica SUB ó Subiculum en color verde es la estructura responsable de enviar y recibir respuestas al mundo exterior; envía respuesta a los

actuadores a través de los caminos LEC y MEC y recibe retroalimentación de las respuestas por la ruta III/V. Recibe también la información de la memoria temporal de la instancia hipocámpica CA1.

En síntesis, se puede indicar que las funciones de memoria se identifican con cada instancia de la siguiente manera:

- La instancia CA3 maneja la memoria episódica.
- La instancia CA1 maneja la memoria temporal.
- La instancia SUB maneja la memoria de trabajo.

Al revisar otras investigaciones sobre los modelos hipocámpicos, tales como los de Kawamura et al y Treves et al, (Kawamura & Gordon, 2006), (Treves & Rolls, 1994), se detectó un inconveniente importante, y es que en ellas no resulta clara la separación entre la información proveniente de la memoria episódica y la memoria de trabajo; por lo tanto para el desarrollo de ésta investigación, se creó un nuevo modelo hipocámpico inspirado en el trabajo de Weisz y Argibay (Weisz & Argibay, 2012), que sirve para el manejo de las memorias que están involucradas con ésta formación que es el punto inicial de la propuesta de la presente tesis, y que permite separar los flujos de información que se reciben entre los diferentes tipos de memorias del hipocampo. La nueva RNA propuesta AHN-ELM (Red Hipocámpica Artificial basada en Máquina de Aprendizaje Extremo) tiene la cualidad de contar con todas las estructuras hipocámpicas necesarias para poder manejar los tres tipos de memoria relevantes: la de trabajo, la de corto plazo y la de largo plazo. Además, la nueva RNA parte de que la función de la corteza entorrinal que es la interfaz de entradas-salidas con el mundo exterior.

La Figura 8 muestra el diagrama a bloques del nuevo modelo simplificado del hipocampo.

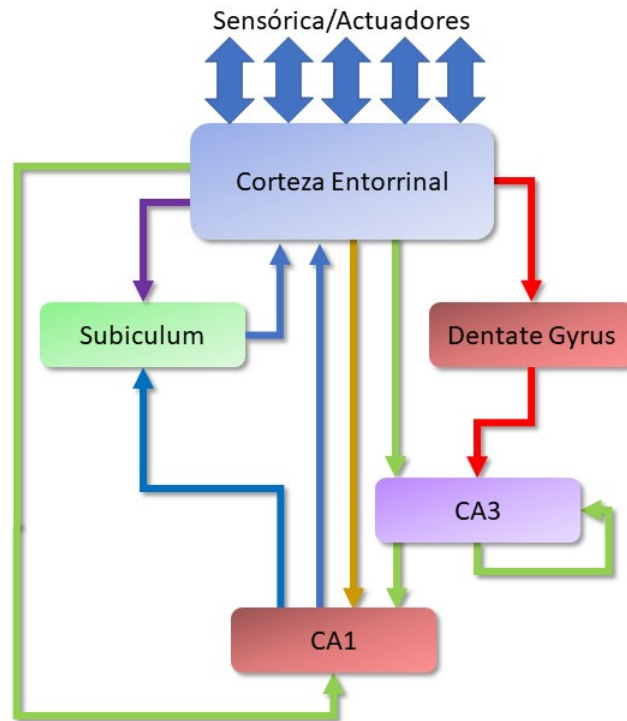


Figura 8. Diagrama a bloques del modelo simplificado del hipocampo. Elaboración propia.

Se pueden observar las instancias hipocámpicas y la interface de la corteza entorrinal que lo componen. Se pueden ver los flujos de información y cómo se relacionan entre ellos. Muestra las zonas Dentate Gyrus (DG) en rojo, Cornu Ammonis 3 (CA3) en morado, Cornu Ammonis 1 (CA1) en rojo y el Subiculum (SUB) en verde, la Corteza Entorrinal (CE) en azul. La interconexión con el mundo exterior se realiza mediante la corteza entorrinal, a través de la memoria sensoria y la memoria de actuación.

El nuevo modelo hipocámpico simplificado que se emplea para generar el modelo de aprendizaje con soporte de memoria de corto y largo plazo las formaciones hipocámpicas seleccionadas que son la DG, CA1, CA3 y SUB y los múltiples flujos de información entre ellas y con la corteza entorrinal.

Este modelo fue creado para contener todas las instancias básicas de la formación hipocámpica y su relación con las diferentes secciones de entrada o salida de la corteza entorrinal. Las secciones I, II, III, IV, V y VI corresponden a la corteza

entorrinal y se usarán posteriormente para el desarrollo del Algoritmo Híbrido Hipocámpico Artificial AHN-ELM. El modelo empieza a procesar los datos de entrada en la instancia *DG*, los datos provienen de la interfaz EC por el canal II, y tiene un ciclo de auto-asociación representado por el flujo que sale y regresa a la misma *DG* que ayuda a la separación de patrones similares. Posteriormente, la información pasa de *DG* a la instancia *CA3*, que recibe tanto esta información como una entrada adicional de la interfaz EC canal II. Nuevamente, se representa un ciclo de auto asociación para aprender patrones relevantes, esta instancia funge como la memoria de largo plazo. Una vez completado este proceso, los patrones relevantes son enviados a la instancia *CA1*, donde aprende la información proveniente de *CA3* junto con una nueva entrada de la interfaz EC canal III. La instancia *CA1* funge como memoria temporal. Después, la información aprendida es enviada a la instancia Subiculum, donde es modificada para interactuar con el entorno; usando la instancia Subiculum como memoria de trabajo, la información de salida al entorno es enviada a la interfaz EC por medio del canal V; la retroalimentación del entorno llega a través de la interfaz EC por el canal IV.

Una vez definidos los flujos de información de la formación hipocámpica, el siguiente paso es determinar los algoritmos para cumplir con las funciones de la Dentate Gyrus y la Cornu Ammonis. Tomando como referencia los trabajos previos (Anireh & Osegi, 2017), (Kowadlo, Ahmed, & Rawlinson, 2019) y (Zhang, Zeng, Zhao, & Wang, HMSNN: Hippocampus inspired Memory Spiking Neural Network, 2016), se optó por utilizar una Máquina de Aprendizaje Extremo por su capacidad de aprendizaje rápido no repetitivo. Esto es particularmente importante, ya que el algoritmo será utilizado en procesos de manufactura, en los que la cantidad de datos puede estar asociada con costos significativos de producción. Adicionalmente, se seleccionó otro algoritmo que consiste en un sistema de filtrado de anomalías para reemplazar la sección de atención, que se produce también dentro de la formación hipocámpica. En este caso, se seleccionó un algoritmo filtro denominado *Part Average Testing* (Haifey & McNally, 2011), (Felding, 2017), (Patterson, 2012) y (Pihlaja, 2013), que permite eliminar anomalías de manera estable para ambientes de producción o de manufactura. Se utiliza una red Hopfield para mimetizar el funcionamiento de la Cornu Ammonis 3 y un

algoritmo de Aprendizaje por Refuerzos para la sección del Subiculum y la corteza entorrinal.

3.1.4 Algoritmo y arquitectura de red neuronal híbrida AHN-ELM

El Algoritmo Híbrido de la Red Neuronal AHN-ELM desarrollado en la presente investigación, se compone de cuatro instancias hipocámpicas y una interfaz para conectar con el entorno. Las instancias que lo componen son DG, CA3, CA1, SUB. Adicionalmente EC es la interfaz (Esta no es una instancia hipocámpica del algoritmo, solo se hace mención para explicar las conexiones con el entorno). Cada una de las instancias hipocámpicas tiene una función particular, como la separación de patrones o la auto asociación, entre otras. En conjunto, las instancias hipocámpicas son capaces de manejar los tres tipos de memorias: episódica, temporal y de trabajo; a semejanza del hipocampo biológico.

Durante la selección de los algoritmos que podrían funcionar como las instancias hipocámpicas se evaluaron una gran variedad de técnicas y algoritmos, entre ellos están las RNA's, que juegan un papel muy importante en los sistemas que requieren aprendizaje. Sin embargo, un punto débil de las RNA's es su lenta velocidad de aprendizaje.

Recientemente, las Máquinas de Aprendizaje Extremo o ELMs han atraído la atención de muchos investigadores como una nueva técnica que resuelve el problema de la velocidad. La técnica ELM trabaja para Redes Monocapa hacia adelante o SLFN. La esencia de las ELM es que la capa de neuronas ocultas de la red no necesita ser entrenada, ya que utiliza un método de aprendizaje analítico en lugar de uno iterativo. Comparado con las técnicas de Inteligencia Artificial tradicionales, las ELM mantienen una buena capacidad de generalización con una velocidad rápida de entrenamiento (Huang G.-B. , An Insight into Extreme Learning Machines: Random Neurons, Random Features and Kernels, 2014) y una muy alta adaptabilidad (Xu & Wang, A fast incremental extreme learning machine algorithm for data streams classification, 2016), (Yin, Li, & Hu, 2014), (Hussain, Siniscalchi, Lee, & Wang, 2017).

La Figura 9 muestra la topología de la máquina de aprendizaje extremo (ELM).

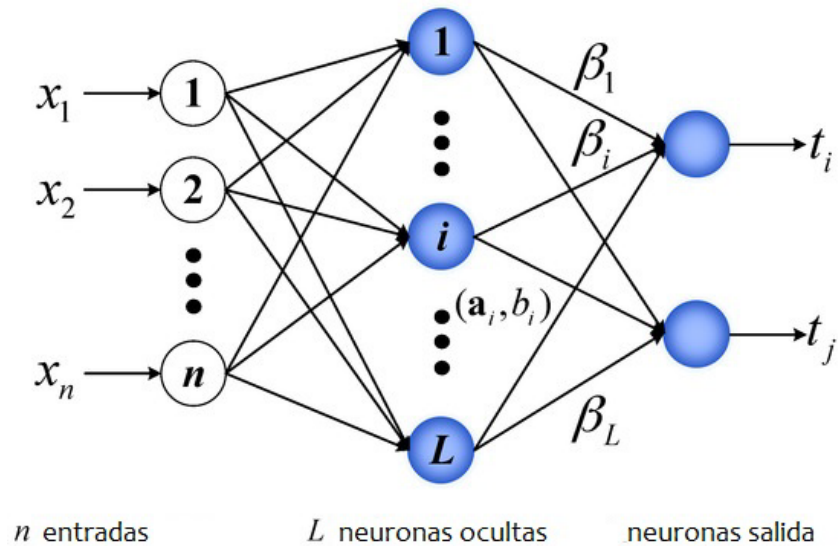


Figura 9. Esquema de una neurona del tipo *Single Layer Feed-Forward Netork* o SLFN: red monocapa de avance hacia adelante. (Huang, Liang, Rong, & Saratchandran, 2005).

Se selecciona el algoritmo ELM como el principal motor de las instancias hipocámpicas por su rapidez de aprendizaje, cualidad esencial en los procesos de manufactura.

Una vez entrenadas, las ELM funcionan igual que las SLFN (Souza, Neto, & Guedes, 2014). El algoritmo ELM, propuesto por Huang et al. se caracteriza por calcular los parámetros de forma analítica (Huang G.-B. , What are Extreme Learning Machines? Filling the Gap Between Frank Rosenblatt's Dream and John von Neumann's Puzzle, 2015) de ahí su extrema velocidad, no requiere iteraciones. Huang demostró que se pueden asignar aleatoriamente los pesos y sesgos de la capa oculta, sin afectar la capacidad de la red de aprender las distintas observaciones relevantes (Xu & Wang, A fast incremental extreme learning machine algorithm for data streams classification, 2016). Esto implica que no es necesario ajustar todos los parámetros de la red. El algoritmo ELM se basa en este principio para ajustar únicamente los parámetros de la capa de salida (Mesquita, Gomes, Rodrigues, & Galvao, 2015).

A la capacidad de lograr resolver un sistema de ecuaciones, se conoce como la capacidad de aproximación. Para la ELM existe el teorema de aproximación universal, que establece que una RNA de una sola capa oculta y un número finito de nodos puede aproximar cualquier función continua con una exactitud arbitraria (Hagan, Demuth, & Beale, 2018), (Gluck, 2009). La SLFN está representada en la Ecuación 1. En ella: L es el número de neuronas ocultas, a_i y b_i representan los parámetros de aprendizaje de los nodos de las neuronas ocultas, β_i es el peso conectado al i – esimo nodo de las neuronas de salida. El valor de salida de la i – esima neurona de salida con respecto a la entrada x se muestra en la Ecuación 2. La Ecuación 1 muestra el nodo oculto aditivo con la función de activación $g(x)$ (Haykin, 2009).

$$f(x) = \sum_{i=1}^L \{\beta_i G(a_i, b_i, x)\} = t_j$$

Ecuación 1. ELM básica.

Las N ecuaciones anteriores pueden ser escritas en forma vectorial, como se muestra en la Ecuación 2:

$$\beta = H^\dagger T$$

Ecuación 2. ELM en forma vectorial (Kasun, Zhou, Huang, & Vong, 2013).

H^\dagger es la matriz salida de la capa oculta de la red neuronal que a su vez es la matriz Inversa de Moore-Penrose, la columna i de la misma es la salida del nodo oculto i , con respecto a las entradas x_1, x_2, \dots, x_N ; $h(x)$ que es el mapeo de las características o parámetros de la capa oculta. La fila i de H es el mapeo de características de la capa oculta con respecto a la entrada x_i : $h(x_i)$, se muestra en la Ecuación 3 (Haykin, 2009).

$$H = \begin{pmatrix} G(a_1, b_1, x_1) & \cdots & G(a_L, b_L, x_1) \\ \vdots & \ddots & \vdots \\ G(a_1, b_1, x_N) & \cdots & G(a_L, b_L, x_N) \end{pmatrix}_{N \times L}$$

Ecuación 3. Matriz de salida de la capa oculta.

La Ecuación 4 es la matriz β que contiene los pesos de las neuronas ocultas (Kasun, Zhou, Huang, & Vong, 2013).

$$\beta = \begin{pmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{pmatrix}_{L \times M}$$

Ecuación 4. Matriz β con los pesos de neuronas ocultas.

La Ecuación 5 es la matriz T que contiene las salidas:

$$T = \begin{pmatrix} T_1^T \\ \vdots \\ T_L^T \end{pmatrix}_{N \times M}$$

Ecuación 5. Matriz de salidas.

Conforme a lo anterior ha probado que, desde un punto de vista de capacidad de interpolación, si la función de activación g es infinitamente diferenciable en cualquier intervalo, los parámetros de la capa oculta pueden ser generados aleatoriamente.

3.1.5 Diagrama de flujo simplificado de red AHN-ELM

En la Figura 10 se presenta el diagrama de flujo simplificado que explica la solución propuesta del Algoritmo Híbrido de Red Hipocámpica Artificial soportada por Máquina de Aprendizaje Extremo, sus componentes y los flujos de información que genera para mimetizar el funcionamiento del hipocampo. Muestra cómo fluyen los conjuntos de datos entre las principales instancias hipocámpicas que componen la Red Hipocámpica Neuronal AHN-ELM. Se indica la función de cada instancia: la instancia DG se encarga de la separación de patrones similares; la instancia CA3 se encarga de almacenar los patrones fundamentales, que se convierten en memorias episódicas; la instancia CA1 se encarga de completar patrones de

respuesta inicial basada en la memoria episódica y la memoria de trabajo y, finalmente, la instancia Subiculum se encarga de interactuar con el entorno. El inicio de flujo de información ocurre de forma dinámica, la instancia DG recibe el conjunto de datos con la información del entorno, del área sensorica. Las instancias CA1 y SUB reciben simultáneamente información sobre la respuesta de los actuadores respecto al entorno y la eficiencia de la respuesta actual.

La salida la produce la instancia SUB, que es la nueva respuesta al entorno, y se retroalimenta a la instancia CA1.

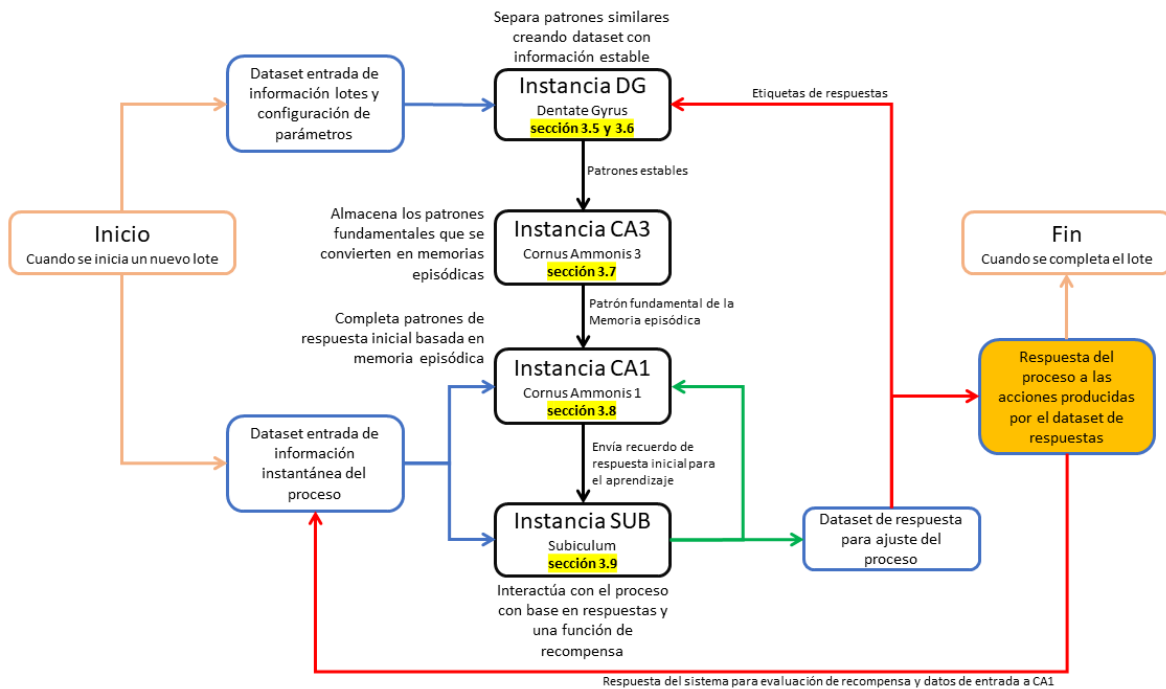


Figura 10. Diagrama de flujo simplificado del algoritmo híbrido AHN-ELM. Elaboración propia.

La Figura 11 muestra la arquitectura de la Red Híbrida AHN-ELM, mostrando la integración de las redes que componen a cada una de las instancias hipocámpicas, se resalta la Dentate Gyrus (DG), Cornu Ammonis 1 (CA3) y CA1 así como el Subiculum (SUB).

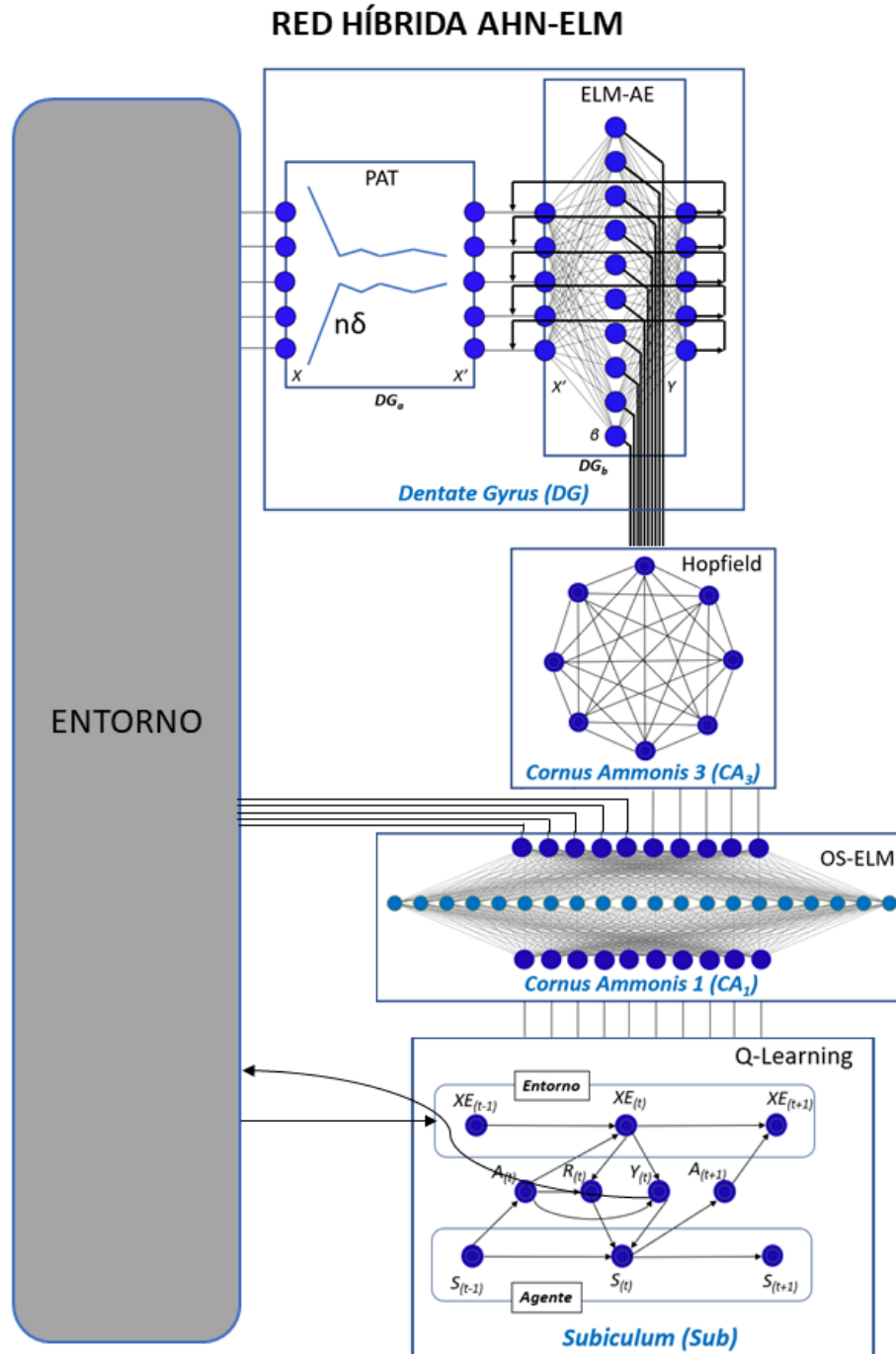


Figura 11. Arquitectura de la red híbrida AHN-ELM.
Elaboración propia.

3.1.6 Algoritmo híbrido simplificado de red AHN-ELM

El Algoritmo Híbrido de la Red Neuronal Hipocámpica AHN-ELM está compuesto por cinco algoritmos, los cuales pueden ser implementados como procedimientos o subrutinas.

El Algoritmo 1 muestra las diferentes instancias hipocámpicas implementadas. El nombre que se le ha dado a cada algoritmo corresponde al de la formación del hipocampo biológico. Tenemos la instancia DG en referencia a la formación Dentate Gyrus, las instancias CA1 y CA3 referentes a las formaciones hipocámpicas Cornu Ammonis 1 y 3, respectivamente, la instancia Subiculum en relación con el Subiculum hipocámpico, y a la interfaz con el entorno se le denomina EC, por la corteza entorrinal. Se muestra también la interacción entre las diferentes instancias hipocámpicas utilizando pseudocódigo. El Algoritmo es un bucle que va llamando a las instancias de manera secuencial y continúa hasta que se termina el conjunto de datos en evaluación, es decir, hasta que el lote en proceso termina en forma práctica. La repetición comienza de nuevo al entrar un nuevo lote o conjunto de datos. El Algoritmo de la Red Neuronal AHN-ELM permite mimetizar la función de gestionar recuerdos episódicos y la memoria de trabajo o memoria temporal.

Algoritmo 1 Control Cognitivo AHN-ELM (Versión Simplificada)

Requiere: Carga de configuración AHN-ELM

```

1:   ← configuración
2:   repeat
3:     Eliminar los datos anormales de fuera  $\pm 3\sigma \rightarrow DG_a$ .
4:     Separar patrones similares en instancia  $DG_b$ .
5:     Asociar patrones episódicos en instancia  $CA3$ .
6:     Completar patrones episódicos con la respuesta dinámica  $\rightarrow CA1$ .
7:     Interactuar y aprender con el medio ambiente mediante SUB.
8:     Repite hasta que se completa el lote de datos y espera nuevo lote.
9:   until lotes  $\neq 0$            ▷ lotes del mismo tamaño
10:  end                         ▷  $nuevoDataset_{lote} = 0$ 

```

Elaboración propia.

3.1.7 Estructura de conjuntos de datos globales y locales de la red hipocámpica AHN-ELM

Variable Global es el nombre que se le da a las variables que utiliza el Algoritmo Híbrido AHN-ELM, estos datos vienen en forma de matrices y vectores y fluyen como información entre los algoritmos de las diferentes instancias hipocámpicas.

Es importante resaltar los nombres de cada uno de los vectores o matrices para poder ir siguiendo las explicaciones de las diferentes instancias.

Para facilitar la identificación de las variables globales, se agrega la

Tabla 1, que muestra los nombres de dichas variables con la información principal del algoritmo de control hipocámpico.

En la primera columna se presentan los nombres de las variables que se utilizan en cada vector o matriz; la segunda, una breve descripción de cada variable y, la última, muestra en qué instancia hipocámpica se está utilizando cada variable.

Tabla 1. Descripción de variables globales usadas en algoritmo AHN-ELM.

Variable	Descripción	Instancia Hipocámpica
x_{EC2}	Vectores de estímulos de entrada	Sensorial $EC_2 \rightarrow DG_a$
x_{DG}	Vector de estímulos de estado estable	$DG_a \rightarrow DG_b$
W_{DG_a}	Vector de patrones dispersos estables	$DG_b \rightarrow CA3_a$
W_{CA3}	Matriz de pesos de memoria	$CA3_a$
Y_{CA3}	Memoria episódica fundamental	$CA3_a$
X_{CA3}	Patrones de estado inestable	$CA3_b \rightarrow CA1$
X_{CA1}	Información episódica de lote	$CA1 \rightarrow EC_3$
$Q_{SUB(0)}$	Patrón memorizado inicial para SUB	$CA1 \rightarrow SUB$
$Q_{SUB(k+1)}(s_k, a_k)$	Respuesta del algoritmo hacia actuadores	$SUB \rightarrow CE_5$

Elaboración propia.

Las variables locales que se utilizan dentro de los diferentes algoritmos de las instancias hipocámpicas se muestran en la Tabla 2. Entre estas variables se encuentran los datos de configuración inicial, es decir, el número de nodos de entrada, el número de neuronas ocultas, el número de neuronas de salida y las variables específicas de cada algoritmo que se utiliza dentro de la Red Neuronal Hipocámpica AHN-ELM. La tabla contiene, en su primera columna, los nombres de

las variables; en la segunda, la descripción de a lo que corresponde cada variable y, en la tercera columna, se indica en qué instancia hipocámpica se utiliza.

Para identificar los nombres de cada variable específica, se les agrega un sufijo, indicando la instancia hipocámpica en que se va a utilizar.

Tabla 2. Descripción de variables locales usadas en algoritmo AHN-ELM.

Variable	Descripción	Instancia Hipocámpica
n	Nodos de entrada	DG y CA1
N_{DG}	Nodos Ocultos	DG y CA1
Y	Nodos de salida	DG y CA1
x_i	Datos de entrada $x_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T$	DG y CA1
T	Variable dependiente y	DG y CA1
β	Vector peso de salida entre neuronas ocultas i_{esima}	DG y CA1
$\hat{\beta}$	Solución de mínimos cuadrados $H^T \beta = T$, que es $\hat{\beta} = H^+ T$	DG y CA1
H	Matriz de salida de capa oculta de la Red Neuronal	DG y CA1
w_i	Vector de peso que conecta la neurona oculta i_{esima} y las neuronas de entrada	DG y CA1
b_i	Umbral de la i_{esima} neurona oculta	DG y CA1
$g(x)$	Función de activación	DG y CA1
\tilde{N}	Número de neuronas ocultas	DG y CA1
n_x	Muestras distintas arbitrarias	DG y CA1
W	Matriz de pesos entre neuronas	CA3
P	Patrones de entrada	CA3
Y_p	Memoria episódica fundamental	CA3
X_p	Entrada episódica inestable	CA3
V^π	Función de valor	SUB
π	Política de control	SUB
Q	Función de un sistema de control	SUB
s	Estado actual del sistema al aplicar una política	SUB
a	Acción de control que se aplica a un sistema	SUB
$Q(s, a)$	Cada par estado-acción de una función	SUB

Elaboración propia.

3.1.8 Eliminación de los datos anormales en la instancia hipocámpica DG_a

La instancia hipocámpica DG se divide en 2 partes, la sección de entrada denominada DG_a y la sección de salida denominada DG_b . El objetivo de la instancia DG_a es mimetizar el proceso básico de atención.

A la atención que la instancia DG_a aplica, es denominada atención selectiva; implica que, para el procesamiento de la información del conjunto de datos entrantes X_{EC2} , se utilizan solo los datos relevantes y se ignoran o bloquean los datos del conjunto que resultan irrelevantes. En este caso, irrelevante significa que no pertenecen al conjunto de datos principal, dicho de otra forma, que son anormales.

Este proceso se simplificó a su mínima expresión, que es la eliminación de los datos anormales que vienen en el conjunto de datos entrantes. Para lograr este proceso, se incorpora el algoritmo de filtrado de datos anormales, para que su atención se centre en el conjunto de datos principal y evite sesgos producidos por el ruido estadístico en el ambiente.

En la Figura 12 se muestra el diagrama de flujo del algoritmo base de la instancia DG_a . En esta instancia se produce la eliminación de datos anormales, mediante el algoritmo PAT. La instancia recibe información de configuración durante el ajuste inicial y los estímulos de la corteza entorrinal mostrado como sensorio EC2.

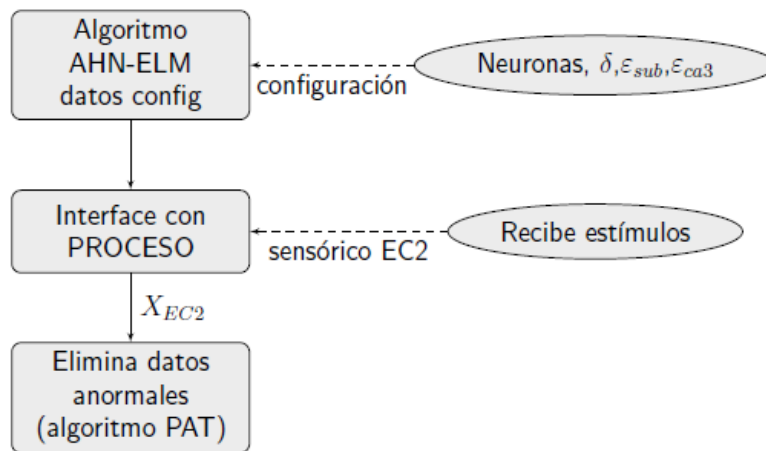


Figura 12. Diagrama de flujo del algoritmo de la instancia dentate gyrus (DG_a).
Elaboración propia.

Para lograr la eliminación de datos anormales en el conjunto de datos, se utiliza el filtro *Part Average Testing* (Felding, 2017) o PAT [por sus siglas en inglés], usado en la industria electrónica.

La Figura 13 muestra el diagrama a bloques del modelo PAT, se pueden observar sus elementos principales. Muestra dos flujos de información, el flujo inicial en color rojo, que usa los límites predefinidos, y el segundo flujo en color azul, que usa la media y la sigma robustas para definir los límites dinámicos (Marinissen, Singh, Glotter, Esposito, & M. Carulli Jr, 2010).

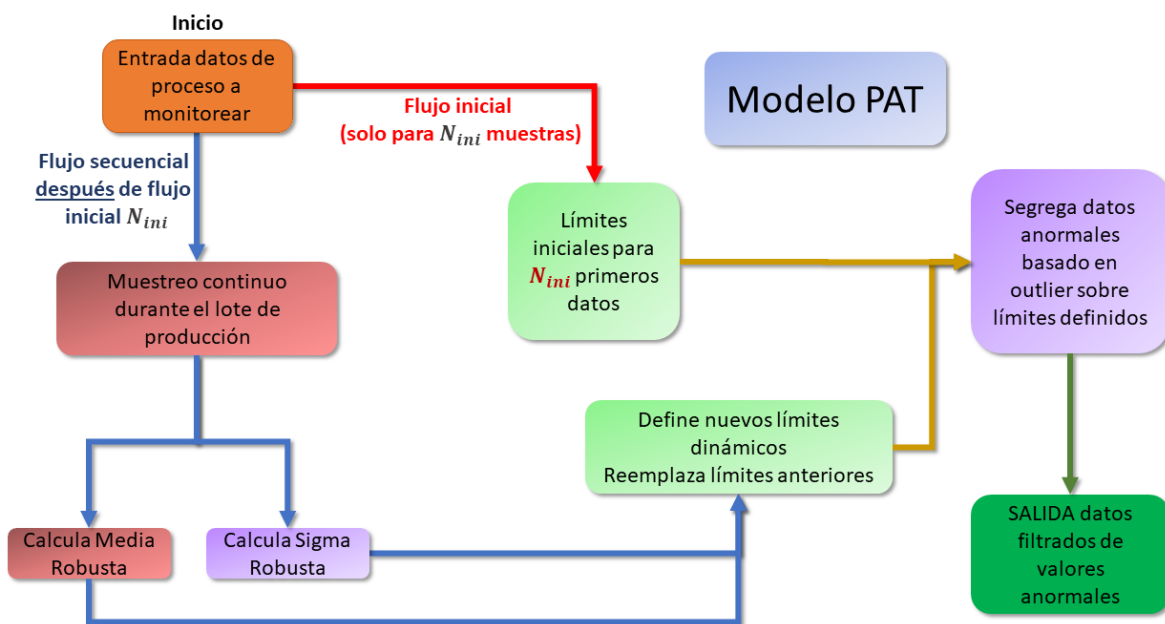


Figura 13. Diagrama a bloques del algoritmo para eliminar los datos anormales, basado en método *part average testing* (PAT).
Elaboración propia.

La Prueba de Promedios Parciales (PAT por sus siglas en inglés) es un método estadístico que permite eliminar mediciones anormales o valores atípicos utilizando lo que se conoce como la media robusta (Ecuación 6) y la sigma robusta (Ecuación 7 y Ecuación 8) (Haifey & McNally, 2011). Los límites PAT dinámicos (Ray, 2005) al inicio del lote de producción tienen el mismo valor que los límites estáticos, ejemplo de esto son los límites de tiempo de apertura y cierre de la tira bimetálica, estos son definidos por el diseño de la geometría del material termostático (Patterson, 2012). A medida que se van midiendo más muestras del lote, se usan los datos para

recalcular el valor sigma (δ) a la sigma robusta, esto se aprovecha para reducir los límites a $\pm n \delta$ del lote en proceso, que normalmente son más pequeños que los límites del diseño. Cualquier dato que está por encima o por debajo de los nuevos límites es descartado; esto permite descartar las mediciones atípicas que podrían sesgar el aprendizaje de la red neuronal. La ventaja de utilizar este tipo de filtrado es su facilidad de implementación y que se puede aplicar a cada una de las variables de entrada del sistema a controlar. El filtro se reinicia con cada lote de producción, por lo tanto, solamente se detectan partes anormales por lote de producción (Pihlaja, 2013).

$$\overline{x_{Robusta}} = Q_{1.3} - \frac{IQR}{2}$$

Ecuación 6. Media robusta para instancia DG_a .

$$PAT_{\text{limite dinámico inferior}} = Q_1 - \frac{(x_i - 0.675)}{1.35} (Q_3 - Q_1)$$

Ecuación 7. Límite dinámico inferior de Instancia DG_a .

$$PAT_{\text{limite dinámico superior}} = Q_3 + \frac{(x_i - 0.675)}{1.35} (Q_3 - Q_1)$$

Ecuación 8. Límite dinámico superior de instancia DG_a

$$x_{PAT} = \begin{cases} \rho = x_\alpha = Q_1 - \frac{(x_i - 0.675)}{1.35} (Q_3 - Q_1) \rightarrow \text{Borrar} \\ \nu = x_\alpha = Q_3 + \frac{(x_i - 0.675)}{1.35} (Q_3 - Q_1) \rightarrow \text{Borrar} \\ \rho < x_a < \nu \rightarrow x_{DG_a} = x_\alpha \end{cases}$$

Ecuación 9. Intervalos válidos para filtrado PAT.

La Ecuación 9 muestra la salida de los intervalos válidos en donde se eliminan los datos atípicos y solo deja pasar los datos filtrados.

El algoritmo PAT crea un filtro estadístico, elimina los datos anormales utilizando los índices de mediana robusta y sigma robusta, estos son usados para definir límites dinámicos de control en el proceso. Los límites son aplicados a todo el conjunto de datos de manera secuencial, el algoritmo va modificando los límites que se aplican a los datos de forma dinámica, lo que significa que se van ajustando continuamente, el filtro descarta todos aquellos datos que están fuera de los límites dinámicos.

El algoritmo inicia tomando los primeros N datos para calcular los límites a los que denominaremos límites de control iniciales UCL_{ini} y LCL_{ini} , que representan la variación natural del proceso dentro de los z niveles σ definidos para segregar la información. El nivel $z \sigma$ se definen desde el inicio del algoritmo.

Los datos entran por el bloque naranja, el proceso comienza usando los primeros N_{ini} datos como flujo inicial.

Posteriormente, a medida que se colectan nuevos datos secuenciales del mismo lote, el algoritmo colecta información secuencial y recalcula la mediana y sigma robustas; las cuales son utilizadas para calcular los nuevos límites dinámicos UCL_{PAT} y LCL_{PAT} , que reemplazan a los límites de control iniciales.

En el bloque morado se aplican los límites al tren de datos que entra y se descartan todos los datos anormales que exceden los límites dinámicos UCL_{PAT} y LCL_{PAT} .

Como señala el proceso *Part Average Testing* en la Figura 14, el resultado del filtrado de datos asegura que todos aquellos componentes del dataset que están por encima del límite superior de control UCL o por debajo del límite inferior de control LCL sean segregados (marcados en color rojo). Este proceso continúa hasta que el lote de datos o dataset es totalmente procesado. La gráfica representa la salida del algoritmo PAT, muestra la variación natural del proceso en la curva de distribución en color azul; la localización de los límites mínimo LSL y máximo USL; y

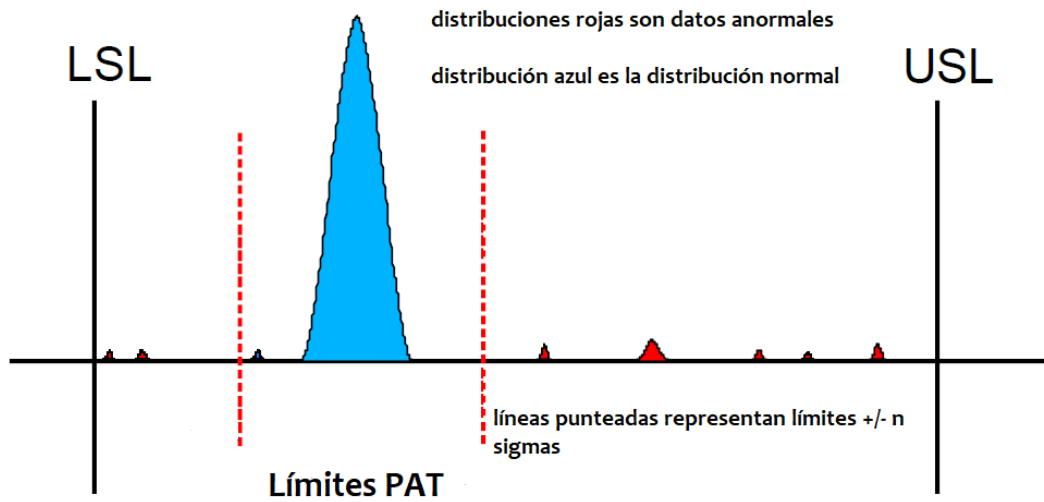


Figura 14. Gráfica de salida del algoritmo.
(Patterson, 2012).

los límites dinámicos representados por líneas punteadas rojas UCL_{PAT} y LCL_{PAT} . Todos los grupos de datos anormales son representados en pequeñas distribuciones en color rojo, están afuera de los límites dinámicos y se clasifican como datos anormales que se deben descartar. El eje de las X es el rango en que pueden aparecer los valores de la variable en monitoreo; LSL y USL indican los límites inferior y superior iniciales, respectivamente. Se puede observar una distribución normal, que representa la población de datos de entrada. Las pequeñas protuberancias en el eje de las X, alejado de esta distribución normal, se denominan outliers, son datos anormales. En situaciones no filtradas, todas las mediciones que están dentro de los límites de especificación pasarían como información correcta. Podemos también observar las líneas punteadas, que representan la variación natural del proceso en $N \delta$. Así como que todos los valores que están por debajo del límite inferior del filtro, representado por la línea punteada de la izquierda, son los datos que deben descartarse, ya que no son parte de la población. Los datos que se encuentran en el lado superior del filtro, después de la línea punteada, también son anormales y deben descartarse.

La Figura 15 muestra cómo se van adaptando los límites superior e inferior en el filtrado PAT, que se van limitando de acuerdo con la variación natural de las

mediciones conforme se van generando. El límite inferior en color café no tiene ninguna medición que lo sobrepase, sin embargo, en el límite superior en color rojo hay una medición que sobrepasa su límite, esto representa una anomalía que está fuera de la variación natural de ese lote de producción (Ray, 2005). Los límites predefinidos iniciales sólo ocurren mientras no se alcance Ψ , una vez que se alcanza el número de muestras Ψ , se inicia el proceso secuencial con los límites dinámicos.

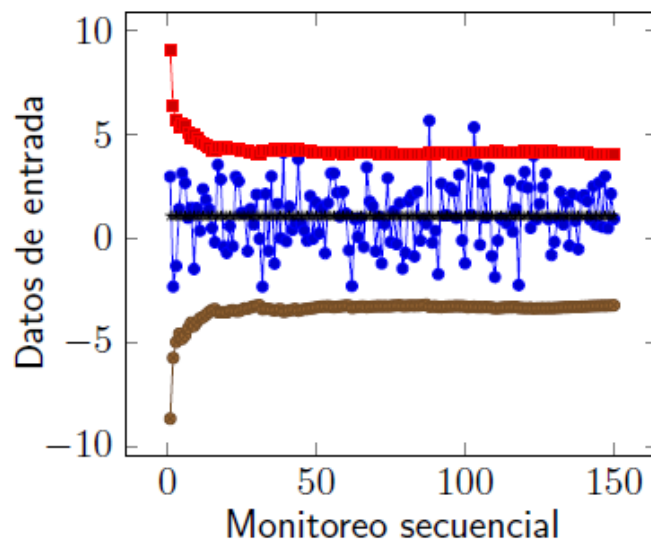


Figura 15. Gráfica de datos y límites dinámicos usando *part average testing*.
Elaboración propia.

La gráfica de tendencia es un ejemplo del filtrado dinámico PAT. Se presentan las mediciones X_{lot} en azul, Límite UCL_{PAT} en rojo, Límite LCL_{PAT} en café. Las mediciones que están fuera de las líneas de límite u o p son descartadas, ya que son datos anormales. Las líneas roja y café representan los límites dinámicos después de ϕ datos de haber sido probados. En ella, se puede observar cómo los límites de especificación son los que inicialmente se utilizan con la población, pero una vez ajustados los límites dinámicos, estos se van cerrando, evitando que todos los puntos anormales de la población pasen como datos normales.

A continuación, el Algoritmo 2 muestra el pseudo código de la instancia hipocámpica DG_a implementada con el algoritmo de Prueba de Promedios Parciales PAT. Se muestra en negritas la información requerida para inicializar el algoritmo, la estrategia de funcionamiento y la salida esperada. La salida del algoritmo Dentate Gyrus inicial DG_a es la matriz $x_{PAT}(n, p)$.

Algoritmo 2 Instancia Hipocámpica DG_a Dentate Gyrus

Requiere: Inicializar Instancia Hipocámpica Baja DENTATE GYRUS (DG_a), requiere Patrones de estado, incluyendo anomalías x_a, x_2, \dots, x_n , número de patrones iniciales ϕ .

Estrategia: El algoritmo descartará los patrones de estímulos sensoriales anormales y solo dejará estímulos de estado estable $x_{PAT}(1), \dots, x_{PAT}(p)$

Salida: Estímulos de estado estable $x_{PAT}(1), \dots, x_{PAT}(p)$

- 1: Carga número de δ para definir $\bar{x}_{robusta}$ y el número de patrones iniciales ϕ , $p = 1$, tamaño del lote n_{lote} , número de lotes L
 - 2: Paso 1: se cargan vectores de estímulos $x_{ec2}(1), \dots, x_{ec2}(n)$ de memoria sensorial que incluyan posibles anomalías
 - 3: Paso 2: se calcula \bar{x} y δ de los ϕ patrones iniciales.
 - 4: **repeat**
 - 5: **for** cada vector de estímulo $x_{ec2}(p)$ **do**
 - 6: Lee $x_{ec2}(p)$
 - 7: Calcula nueva $\bar{x}_{robusta} = Q_{1,3} - \frac{IQR}{2}$
 - 8: Calcula nuevo $PAT_{limiteinferior} = Q_1 - \frac{(x_i - 0,675)}{1,35}(Q_3 - Q_1)$
 - 9: Calcula nuevo $PAT_{limitesuperior} = Q_3 + \frac{(x_i - 0,675)}{1,35}(Q_3 - Q_1)$
 - 10: Se evalúa $x_{ec2}(p)$
 - $$x_{PAT}(p) = \begin{cases} x_{ec2}(p) < PAT_{limiteinferior} & \rightarrow delete \\ x_{ec2}(p) > PAT_{limitesuperior} & \rightarrow delete \\ \rho < x_{ec2}(p) < \vartheta & x_{PAT}(p) = x_{ec2}(p) \end{cases}$$
 - 11: Lee siguiente elemento x_{ec2}
 - 12: **end for**
 - 13: **until** Se completan L lotes evaluados
 - 14: **return** $x_{PAT}(p)$
 - 15: -
-

Elaboración propia.

3.1.9 Separación patrones similares en instancia DG_b

Para realizar la separación de patrones similares y crear la sección de salida del algoritmo Dentate Gyrus DG_b , se seleccionó una Red Neuronal del tipo ELM. La red ELM se configura como un autocodificador, que genera nuevos patrones dispersos a partir de los patrones provenientes del algoritmo PAT. Los nuevos patrones se denominan XDG_b , la Figura 16 muestra el diagrama de flujo de la instancia hipocámpica DG_b . La salida del algoritmo PAT es la entrada XDG_a del algoritmo ELM-AE.

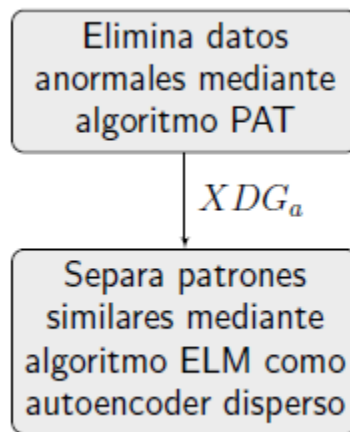


Figura 16. Diagrama de flujo del algoritmo base de la instancia hipocámpica dentate gyrus (DG_b).
Elaboración propia.

Los autocodificadores están considerados como simples SLFN entrenados de forma no supervisada, donde la dimensión de los datos de entrada coincide con la de los datos de salida. En otras palabras, un autocodificador puede aprender a replicar la entrada a red aproximando una función de identidad. Sin embargo, una característica destacada es que, con el autocodificador, se consigue que la red aprenda en su capa intermedia una representación de los datos de entrada sobre un espacio de diferente dimensión (Martínez Álvarez, 2017). La dimensión de los datos puede ser superior, igual o inferior al conjunto de entrada, con lo cual, el código resultante será una representación expandida, igual o comprimida de los mismos.

El autocodificador de la Máquina de Aprendizaje Extremo (ELM-AE) (Zhang, Yin, Zhang, & Shi, 2017), (Kasun, Zhou, Huang, & Vong, 2013) es un algoritmo de aprendizaje no supervisado de una sola capa oculta, donde los datos de entrada también se utilizan como datos de salida.

Se utiliza una representación Esparcida (*Sparse Representation*) donde la ELM-AE tiene mayor número de neuronas ocultas que los nodos de entrada. Para la Representación Escasa o Esparcida ($d < L$), el mapa de características se calcula como: $h(x_j) = g(x_j A + b)$, donde los parámetros de capa oculta son ortogonales aleatorios $A^T = 1$ y $bb^T = 1$ (Martínez Álvarez, 2017), (Hussain, Siniscalchi, Lee, & Wang, 2017).

La función principal de la instancia DG_b es la separación de patrones. La instancia es creada usando el algoritmo de Máquina de Aprendizaje Extremo ELM-AE en modo disperso. El algoritmo ELM-AE recibe como entrada un conjunto de datos XDG_a , provenientes del algoritmo PAT, y los procesa, generando un nuevo conjunto de datos en una codificación dispersa que es más estable que la entrada original DG_b . Los datos de salida de la instancia XDG_a son representaciones esparcidas de los datos originales.

La Figura 17 muestra la estructura de la red neuronal de la instancia hipocámpica DG_b . Se implementa con base en una red del tipo autoencoder en una configuración esparcida (*sparse*).

Las neuronas de entrada x_1, \dots, x_n son iguales que las neuronas de salida y_1, \dots, y_n . Durante el proceso de entrenamiento se determinan los pesos de las neuronas de la capa oculta, que, en este caso, son más que las neuronas de entradas $\widehat{y}_1, \dots, \widehat{y}_6$. Al ser más neuronas, la información es transformada en un proceso esparcido que se convierte en el codificado esperado de esta red neuronal. La información filtrada proveniente del algoritmo PAT se conecta a la capa de entradas x_1, x_2, x_3, x_4 y a la capa de neuronas de salida y_1, y_2, y_3, y_4 . Cuando la red de la capa oculta es entrenada, genera las características dispersas o *sparse features* en inglés. Las características dispersas consisten en la información de entrada, que es la representación en forma estable. El requisito principal para esta configuración es

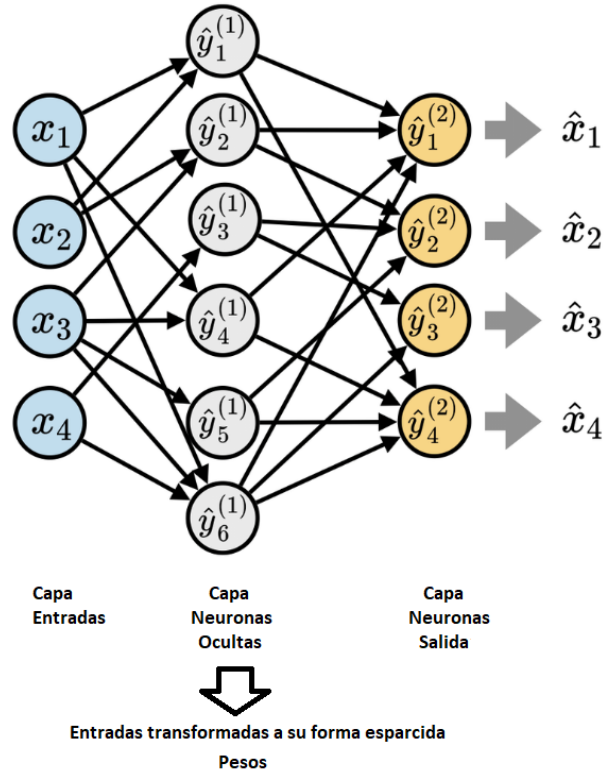


Figura 17. Modelo de la red neuronal de la instancia hipocámpica DG_b . (Dutta, Bai, Jeong, Low, & Grover, 2018).

que el número de neuronas de la capa oculta sea mayor a los nodos de entrada. Entre más grande sea la diferencia entre las entradas y el número de neuronas de la capa oculta, mayor será la dispersión de datos, facilitará el proceso de diferenciación de patrones. Una vez que la información ha sido transformada a un modo disperso, no se parece más la información inicial.

La Figura 18 muestra un ejemplo de cómo se representarían los números del 0 al 9 en su estado disperso. Inicialmente cada patrón representaba a un número 0, ..., 9; después de ser transformados, los patrones se ven como conjuntos de puntos, que tienen la cualidad de diferenciarse mucho, por lo que la posibilidad de confundir los números 6, 8 ó 9, que antes de la transformación era alta, dada su similitud, es ahora muy baja, ya que en su forma esparcida son muy diferentes, lo que evita confusión entre los patrones. Una vez que se han obtenido las características dispersas de los datos provenientes del algoritmo PAT, son enviadas al algoritmo CA3 para que puedan ser auto asociadas a diferentes eventos episódicos.



Figura 18. Vectores en formato 32 x 32 después de ser transformados en la instancia hipocámpica DG_b .
Elaboración propia.

Algoritmo 3 Instancia Hipocámpica DG_b Dentate Gyrus

Requiere: Inicializar Instancia Hipocámpica Alta DENTATE GYRUS (DG_b)

Estrategia: El algoritmo transformará los patrones de estímulos sensoriales de estado estable $x_{PAT}(p)$ a una representación dispersa estable W_{PAT} que permite separar patrones similares

Salida: Patrones dispersos estables $W_{PAT}(1), \dots, W_{PAT}(n)$

- 1: Carga los datos $\{W_{PAT}\} = \{PAT(p)_i\}_{i+1}^N$, número de neuronas i y el coeficiente de penalización C y se espera la salida de la categorización por clusters.
 - 2: **Inicializa DG** Se obtienen los datos de entrenamiento
 - 3: **repeat**
 - 4: Paso 1: se generan pesos y sesgos ortogonales las neuronas ocultas n_h
 - 5: Paso 2: Calcular los pesos de salida β
 - 6: **if** $h_h \leq N$ **then**
 - 7:
$$\beta^* = \left(H^T H + \frac{I_{h_h}}{C} \right)^{-1} H^T x_{PAT}$$
 - 8: **else if** Calcular los pesos de salida β **then** LState $\beta^* = H^T \left(H H^T + \frac{I_N}{C} \right)^{-1} x_{PAT}$
 - 9: **end if**
 - 10: Paso 3: $N_{new} = x_{PAT} \beta^T$.
 - 11: Paso 4: Tratar cada fila $(x_{PAT})_{new}$ como un punto y agrupar los N puntos en K clusters usando algoritmo k-means.
 - 12: **until** Número de patrones faltantes de evaluar del lote = 0
 - 13: **return** W_{PAT}
 - 14: -
-

Elaboración propia.

Como se mencionó antes, al implementar el auto codificador para obtener características dispersas, se utiliza el algoritmo ELM-AE. Este algoritmo aprovecha la característica de Máquina de Aprendizaje Extremo y permite una transformación eficiente de datos de su formato original al formato disperso. Para obtener las características dispersas es necesario entrenar la capa oculta. Este entrenamiento se realiza por lotes, después de que los datos son filtrados por el algoritmo PAT. Para poder manejar la información entre las instancias hipocámpicas, los datos se agrupan por lotes. Una vez agrupados los datos, se entrena a la red ELM-AE.

El algoritmo 3 muestra el pseudo código de la instancia DG_b , implementada con algoritmo ELM-AE.

3.1.10 Asociando patrones episódicos en instancia CA3

La asociación de patrones consiste en el proceso de memorizar información de entrada en una red neuronal no supervisada. La información almacenada puede ser recuperada posteriormente, comparando el patrón de entrada con el patrón almacenado. El patrón almacenado es disparado, aunque la información de entrada no sea exactamente igual.

La Figura 19 muestra el algoritmo base de la instancia hipocámpica Cornu Ammonis 3 (CA3), que se utiliza para auto-asociar los patrones. La selección del algoritmo para crear la instancia CA3 se enfoca en varios algoritmos neuronales recurrentes (RNN) donde resalto el algoritmo de Red Neuronal Hopfield por su velocidad de procesamiento. Una de las características de las Redes Recurrentes es que pueden llegar a ser inestables. Esto es que no logran la convergencia del error a cero. Las redes no recurrentes, como las redes de Máquina de Aprendizaje Extremo dan soluciones estables, mientras que los modelos recurrentes dan soluciones inestables, sin embargo, la red neuronal recurrente Hopfield es del tipo estable y permite su aplicación a algoritmos que busquen la capacidad de memorización por asociación (Ramya, Kavitha, & Shreedhara, 2011).

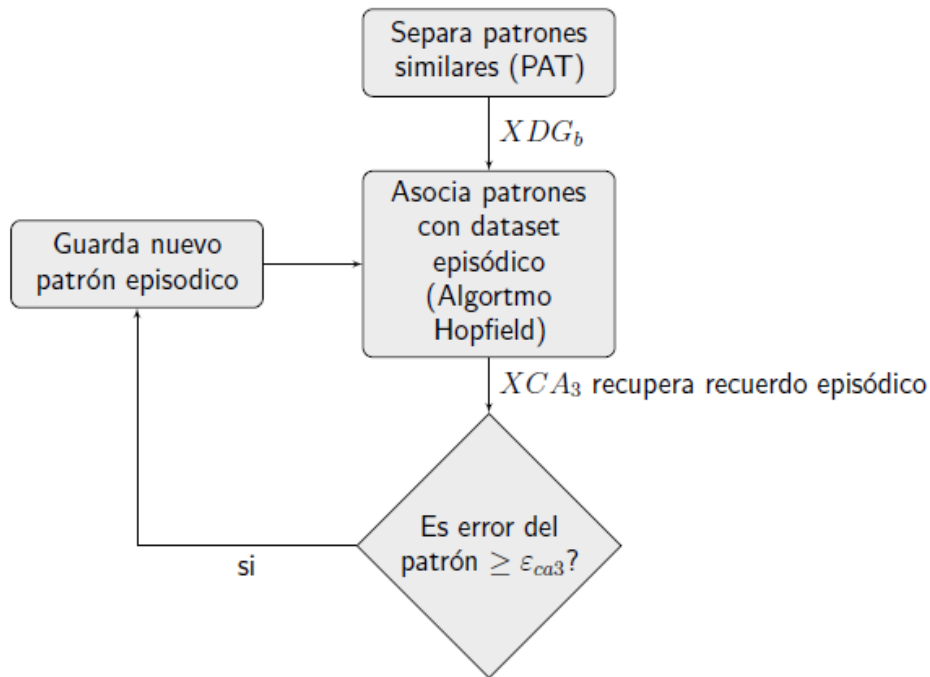


Figura 19. Diagrama de flujo del algoritmo base de la instancia hipocámpica cornu ammonis 3 (CA3).
Elaboración propia.

La red neuronal Hopfield tiene una arquitectura de una capa con interconexión total, utilizan funciones de activación booleana, usualmente 0 o 1, dependiendo de si la estimulación total recibida supera un umbral determinado (Frolov, Húsek, Muraviev, & Polyakov, 2010).

Para implementar la Instancia CA3 empleando el algoritmo tipo Hopfield, se observa la interacción entre la instancia DG_b y el proceso de decisión si debe o no almacenar el nuevo patrón fundamental. El proceso de decisión se da cuando el error del patrón es $\geq \epsilon_{CA3}$ (Kumar & Singh, 2010). Este algoritmo que emula la función de la formación Cornu Ammonis 3 del hipocampo biológico, recibe el conjunto de datos del algoritmo DG_b , el vector de datos con los patrones dispersos XDG_b y genera una serie de memorias fundamentales (Hillar, Mehta, & Koepsell, 2014), que se emplean para reconocer los eventos episódicos que ocurren en el área sensorica y en la información episódica de lotes de datos procesados.

Se recurre a una red Hopfield para realizar la asociación de patrones, puesto que es una red autoasociativa. La información o patrones recibidos son almacenados en la red, empleándose como memoria durante la etapa de aprendizaje. Posteriormente, cuando se presenta una entrada a la red, ésta evoluciona hasta generar una salida, que coincidirá con la que corresponde a esa entrada; o bien la más parecida, si la entrada está distorsionada o incompleta (Szandata, 2015).

La información que recibe la red viene codificada y se representa en forma de vector, con tantos componentes como neuronas (N) tenga la red. La red autoasociativa se configura para trabajar en forma binaria o como un conjunto de valores reales, dependiendo de si la red es discreta o continua. Cada neurona recibe un elemento del vector.

Centrándonos en una sola neurona, el funcionamiento sería el siguiente: recibe como entrada, la salida de cada una de las otras neuronas (por las conexiones laterales); estos valores de salida inicialmente coinciden con las entradas del vector, multiplicadas por los pesos de las conexiones correspondientes. La suma de todos estos valores constituirá el valor de entrada neta de la neurona, al que se debe aplicar la función de activación, obteniéndose así el valor de salida correspondiente. En el instante inicial $t = 0$, la información de entrada es e_1, e_2, \dots, e_N (Hillar, Mehta, & Koepsell, 2014).

La estrategia de aprendizaje de la red Hopfield inicia con la Ecuación 10 que calcula los pesos que conectan los nodos, donde w_{ij} es el peso que va a la neurona i al j y Out_i es el valor de salida del i –ésimo elemento (nodo) (Ramya, Kavitha, & Shreedhara, 2011).

$$net_j = \sum_i w_{ij} \cdot out_i$$

Ecuación 10. Aprendizaje de nodos.

La Ecuación 11 representa la salida y la mecánica de convergencia (Frolov, Húsek, Muraviev, & Polyakov, 2010).

$$\text{Out}_i = 1 \text{ si } \text{net}_i > q_i$$

$$\text{Out}_i = 0 \text{ si } \text{net}_i \leq q_i$$

Ecuación 11. Criterios de salida del nodo.

Para el aprendizaje de los pesos, la elección de la regla de aprendizaje no es trivial, debido a que depende de la interrelación de los patrones que se desean memorizar. Si estos patrones están poco correlacionados (pseudo-ortogonales), podemos aplicar la regla de Cooper-Hebb, basada en la regla de Hebb o regla del producto que señala lo siguiente:

Si tenemos P patrones (binarios) que almacenar ($p1..pm$), calculamos los pesos como se muestra en la Ecuación 12:

$$W_{ij} = W_{ji} = \sum_{i=1}^P Y_P * Y_P^T - P * I$$

Ecuación 12. Ecuación que calcula los patrones.

Donde $w_{ij} = 0$ si $i = j$ (la diagonal vale 0)

El entrenamiento de la red recurrente Hopfield es un proceso iterativo en el que se aplican los conjuntos de datos de entrada para entrenar a las neuronas totalmente conectadas.

El proceso de entrenamiento se repite hasta que la salida es constante, entonces se dice que es una red estable.

El principio de operación de la red Hopfield se basa en un grupo de neuronas n en donde la salida de cada neurona sirve como retroalimentación en las entradas, a excepción de su propia entrada, tal como se muestra en la Figura 20.

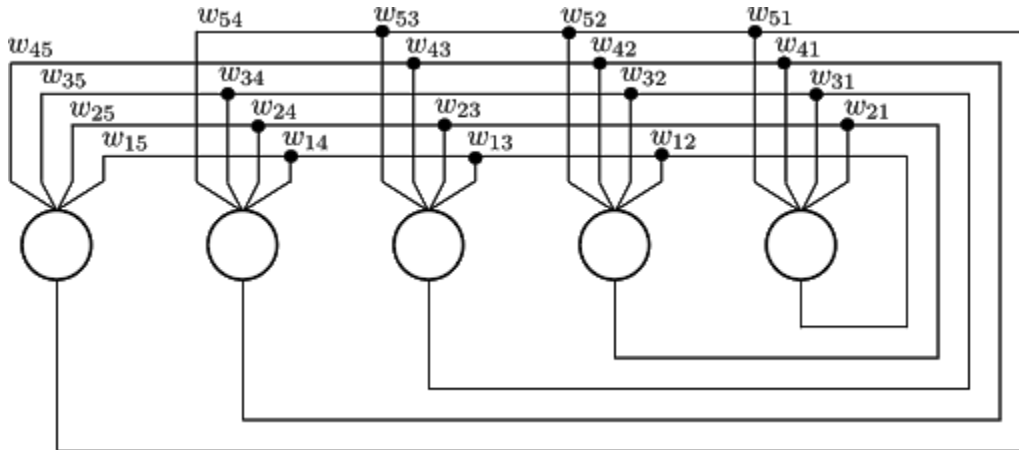


Figura 20. Modelo de red neuronal de la instancia hipocámpica CA3.
Elaboración propia.

En este ejemplo la red está compuesta de 5 nodos. Se especifican las interconexiones de cada neurona con todas las demás, menos consigo misma.

Cada una de las neuronas es normalmente el tipo SLFN, y se emplea una función *sign* durante el entrenamiento.

La Figura 21, muestra un conjunto de imágenes, cada una de ellas representa un vector de 1,024 bits convertido en una matriz de 32 x 32.

La columna marcada como original, muestra los patrones aprendidos en memorias fundamentales originales y, de ahí hacia la derecha, se presentan los diferentes patrones a los que se les agregó ruido, del 10% hasta el 50%, que representan los patrones incompletos que disparan de igual manera al Patrón original. Estas imágenes son un ejemplo de cómo el algoritmo Hopfield puede recuperar la matriz original, independientemente de que la matriz de entrada contenga ruido.

En ella se puede observar, de derecha a izquierda, cómo es capaz de reconocer o reproducir la matriz original almacenada en las neuronas de la red Hopfield, aunque contengan un 10%, 20% o inclusive hasta 50% de ruido.

Este proceso es la recuperación del patrón fundamental, que es lograda por la red recurrente gracias a la auto asociación que hizo con la matriz original durante el proceso de aprendizaje.

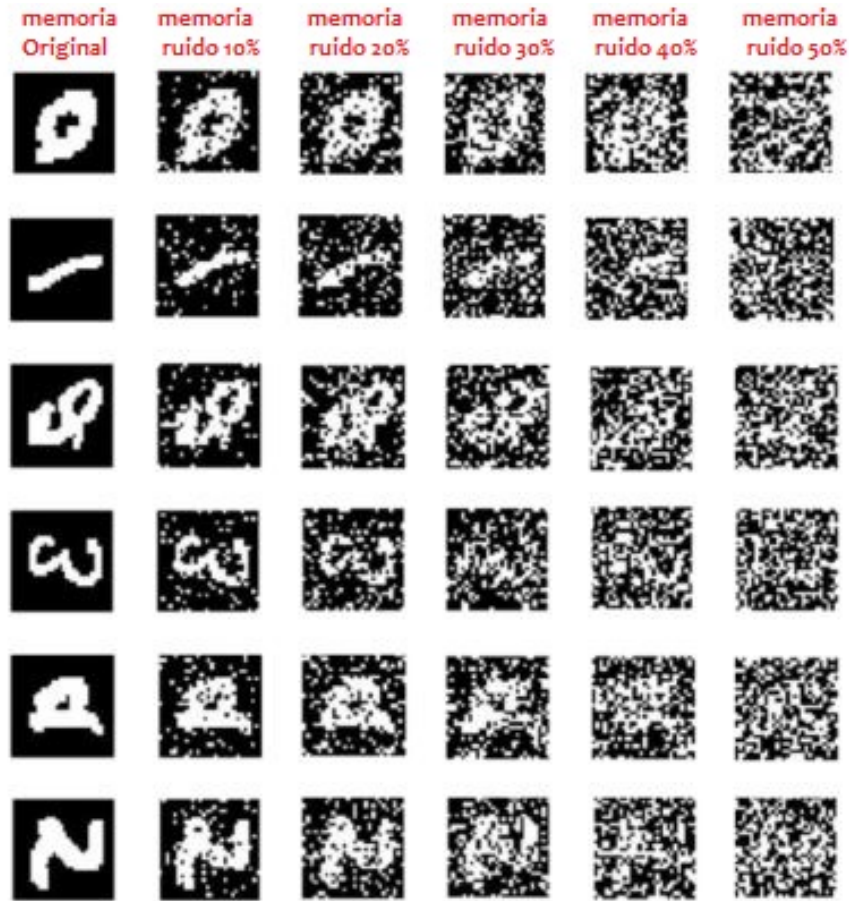


Figura 21 . Muestra los vectores en formato 32 x 32, que representan los procesos de recuperación de 6 patrones aprendidos, como memorias fundamentales.
Elaboración propia.

Los algoritmos que componen a la instancia hipocámpica Cornu Ammonis 3, *CA3* se muestran en los algoritmos 4 y 5, presentados a continuación.

El algoritmo 4, corresponde a la Instancia Hipocámpica *CA3* en modo aprender, refiere al proceso de almacenaje de patrones fundamentales, que ocurre cada vez que se detecta un conjunto de datos nuevo que no está almacenado en las neuronas de la matriz Hopfield. La instancia hipocámpica *CA3* determina si se trata de un nuevo conjunto de datos que deba almacenarse, lo que ocurrirá si, al tratar de recuperar el patrón fundamental basado en ese conjunto de datos, se obtiene un error mayor al 80% cuando se compara al patrón recuperado contra el conjunto de datos.

Algoritmo 4 Instancia Hipocámpica $CA3_a$ en modo Aprender (**Cornus Ammonis 3**)

Requiere: Se inicializa Instancia Hipocámpica $CA3_a$ en modo Aprendizaje

Estrategia: calcular la matriz de pesos Y_P para las memorias episódicas fundamentales $P W_{PAT}$.

Salida: Se calcula la matriz de pesos (W) entre las neuronas, los pesos de la matriz son simétricos, la diagonal principal =0.

- 1: Cargar memoria episódica fundamental (Y_1, Y_1, \dots, Y_P) , la longitud de todos los vectores es igual (t_{th})
 - 2: **if** $i \neq j$ **then**
 - 3: $W = \sum_{i=1}^P Y_P * Y_P^T - P * I$
 - 4: **else if** $W_{ij} = 0$ **then**
 - 5: termina entrenamiento
 - 6: **end if**
 - 7: **return** W
 - 8:
 - 9: **Fin.**
-

Elaboración propia.

Algoritmo 5 Instancia Hipocámpica $CA3_b$ en modo Probar (**Cornus Ammonis 3**)

Requiere: Se inicializa Instancia Hipocámpica $CA3_b$ modo Prueba

Estrategia: La red recordará memorias episódicas fundamentales (Y_P) cuando se presente una entrada inestable (X_P) o X de prueba.

Salida: Cargar patrones de estado estable X_1, X_1, \dots, X_P .

- 1: Cargar patrones de estado inestable (X_1, X_1, \dots, X_P) , la longitud de los vectores (X) es igual que la longitud de los recuerdos episódicos fundamentales (Y).
 - 2: **if** $fX_P = Y_P$ **then**
 - 3: $X_k = \text{sign}((WX_k) - \beta)$
 - 4: **else if** versión incompleta o corrupta (prueba) **then**
 - 5:
 - 6: **end if**
 - 7: $Y(0) = \text{sign}((WX_k) - \beta)$
 - 8: **repeat**
 - 9: $Y(i_{tr+}) = \text{sing}((WX_k) - \beta)$
 - 10: **until** Mientras el estado del vector no cambie
 - 11: **Fin.**
-

Elaboración propia.

El algoritmo 5, corresponde a la Instancia Hipocámpica CA3 en modo funcionamiento, refiere al proceso de recuperación de patrones fundamentales almacenados en la red neuronal Hopfield ya entrenada.

De este modo, el algoritmo compara la representación esparcida que recibe de la instancia hipocámpica DG_b , y busca recuperar el patrón original fundamental almacenado. Este proceso elimina el posible ruido incluido en la representación.

El patrón fundamental de la representación esparcida recuperada es, conceptualmente, la recuperación de un engrama de memoria o recuerdo episódico.

3.1.11 Completar patrones episódicos con instancia hipocámpica CA1

Completar patrones episódicos es un proceso en el que se recupera un recuerdo episódico, que incluye tanto al evento registrado del medio ambiente que lo inició, como a la memoria de actuación o de trabajo. El patrón recuperado contiene en la memoria de trabajo la mejor solución registrada de los actuadores al estar en contacto con el entorno. La memoria de trabajo es actualizada cada vez que hay una nueva respuesta del sistema de actuación y se recibe una retroalimentación del medio ambiente, esto hace que la memoria se actualice secuencialmente.

Continuando con los objetivos de la investigación se selecciona el algoritmo OS-ELM (*Online Sequential Extreme Learning Machine*) (Huang, Liang, Rong, & Saratchandran, 2005), para crear la instancia hipocámpica Cornu Ammonis 1 o CA1; la cual es capaz de completar los patrones episódicos provenientes de CA3 con los de respuesta de memoria temporal o de trabajo. La OS-ELM es una red neuronal basada en una Máquina de Aprendizaje Extremo o ELM, que tiene un aprendizaje rápido y no requiere parámetros que deban seleccionarse al iniciar el aprendizaje, excepto por la cantidad de neuronas \bar{N} . La OS-ELM se construye a partir de una ELM, a la que se le reemplaza la operación de pseudo-inversa por una solución RLS (*Recursive Least Squared*), con el fin de convertir al ELM en un algoritmo de aprendizaje secuencial (Huang, Liang, Rong, & Saratchandran, 2005).

El algoritmo OS-ELM se divide en dos fases: la primera es la etapa *boosting* y la segunda es el aprendizaje secuencial (Chong & Žak, 2001).

En la etapa *boosting*, el primer lote de datos será procesado por el algoritmo a partir del procedimiento presentado en la Ecuación 13, iniciando con:

$$R = \left\{ (x_i, t_i) \mid x_i \in R^n, t_i \in R^m, i = 1, \dots, \hat{N} \right\}$$

Ecuación 13. Etapa *boosting* de CA3.

Primero se asignan arbitrariamente los pesos de W_i y de b_i , donde $i = 1, \dots, \hat{N}$.

Después se calcula la matriz de salida de las neuronas ocultas H_0

$$H_0 = [h_1, \dots, h_{\hat{N}}]^T$$

Ecuación 14. Matriz de salida de neuronas ocultas H_0 .

donde si $i = 1, \dots, \hat{N}$ tenemos

$$h_{_i} = [g(w_1 \cdot x_i + b_1), \dots, g(w_{\hat{N}} \cdot x_i + b_{\hat{N}})]^T$$

Ecuación 15. Matriz con incremento.

Después se estima el peso inicial de las salidas $\beta^{(0)}$

$$\beta^{(0)} = M_0 H_0^T T_0$$

Ecuación 16. Peso inicial de las salidas $\beta^{(0)}$

Donde:

$$M_0 = (H_0^T H_0)^{-1}$$

Ecuación 17. Valor Inicial de entrenamiento.

$$T_0 = [t_1, \dots, t_{\hat{N}}]^T$$

Ecuación 18. Objetivo inicial.

y por último hacemos que $k = 0$, para así completar la fase de arranque.

La segunda fase de aprendizaje del algoritmo OS-ELM es el aprendizaje secuencial, el cual inicia con las observaciones subsecuentes al lote de observaciones inicial. Podemos decir que cada nuevo dato se representa como (x_i, t_i) donde: $X_i \in R^n, t_i \in R^m; i = \hat{N} + 1, \hat{N} + 2, \hat{N} + 3, \dots, \text{siguiente}$.

Primero calculamos el vector de salida de la capa oculta incremento.

$$h_{k+1} = [g(w_1 \cdot x_i + b_1), \dots, g(w_{\hat{N}} \cdot x_i + b_{\hat{N}})]^T$$

Ecuación 19. Vector de salida de la capa oculta incremento.

Enseguida se calculan los nuevos pesos de salida, basándose en el algoritmo RLS:

$$M_{k+1} = M_k - \frac{M_k h_{k+1} h_{k+1}^T M_k}{1 + h_{k+1}^T M_k h_{k+1}}$$

Ecuación 20. Recursive least squared .

$$\beta^{(k+1)} = \beta^{(k)} + M_{k+1} h_{k+1} (t_{k+1}^T - h_{k+1}^T \beta^{(k)})$$

Ecuación 21. Nuevos pesos de salida, basándose en el algoritmo RLS.

Por último:

$$k = k + 1$$

Ecuación 22. Incrementos de aprendizaje.

Por ejemplo, si hay 10 neuronas y tenemos 10 muestras de entrenamiento para el aprendizaje boosting, el rango de igual manera equivaldrá a $rank(H_0) = \hat{N}$, si los primeros \hat{N} datos de entrenamiento son distintos. Después de la fase de aprendizaje boosting, la OS-ELM aprenderá del tren de datos uno por uno, o pedazo por pedazo (mini batch). Todos los datos del entrenamiento inicial serán descartados una vez que se complete el aprendizaje secuencial.

La Figura 22 muestra el flujo de datos que vienen de la instancia hipocámpica CA3 formada por la Red Hopfield y que va hacia la instancia CA1. También se muestra el flujo secundario de la respuesta del entorno que se está controlando.

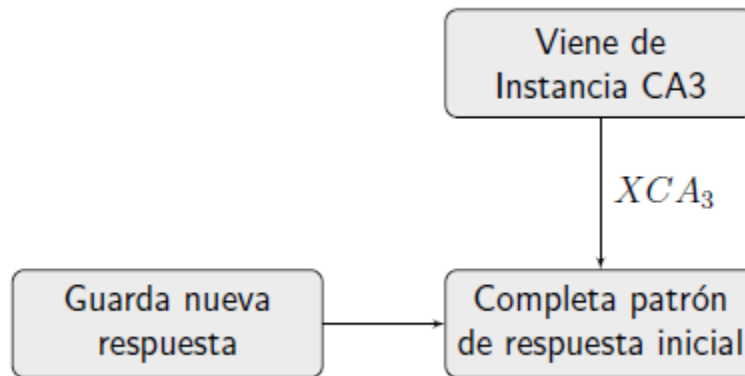


Figura 22. Diagrama de flujo de la instancia hipocámpica CA1, basada en máquina de aprendizaje extremo secuencial.
Elaboración propia.

La instancia Cornu Ammonis 1 trabaja con una función de activación continua como lo explica Huang et al. (Huang, Liang, Rong, & Saratchandran, 2005), en este caso, para la instancia CA1 se selecciona la función de activación continua Sigmoid que funciona apropiadamente con el tipo de datos a memorizar. Durante el proceso de entrenamiento de la Cornu Ammonis 1, los pesos sinápticos W y los valores de sesgo b , son elegidos aleatoriamente y nunca son calculados durante el aprendizaje. Los pesos de salida Q son determinados analíticamente para cada nuevo patrón entrante. Los patrones entrantes ingresan de manera secuencial, lote por lote.

El aprendizaje de la Cornu Ammonis 1 se realiza en dos fases. La primera, denominada boosting, es ejecutada para el par $\{x, y_d\}$, el segundo par, el tercer par, Hasta el N-ésimo par. La segunda fase es denominada fase de aprendizaje secuencial, es ejecutada para el resto de los patrones entrantes, lo que sucede cuando tenemos un patrón almacenado (x_0) , la fase secuencial se encarga de actualizarlo, es decir, para el par $(x_0 + 1)$ –ésimo, el par $(x_0 + 2)$ –ésimo, ..., hasta que el aprendizaje finaliza.

La Figura 23 es un modelo de red neuronal que muestra la estructura básica de la instancia Cornu Ammonis 1, implementada con la red neuronal de Máquina de Aprendizaje Extremo Secuencial, muestra la capa de entrada en color azul, las neuronas ocultas en rojo y las neuronas de salida en verde, el modelo aprende cada vez que recibe k entradas y evalúa la salidas y , contra los objetivos t y determina la pérdida L .

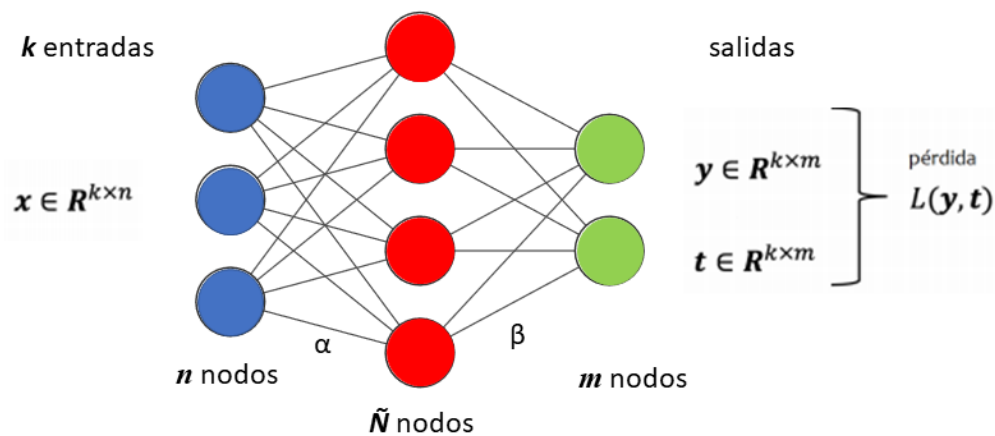


Figura 23. Modelo de red neuronal de la instancia hipocámpica CA1. (Huang, Liang, Rong, & Saratchandran, 2005)

La fase de entrenamiento Boosting se aplica a la red neuronal SLFN usando un lote inicial de datos de entrenamiento. El concepto de entrenamiento boosting en el caso de la OS-ELM, refiere a que se entrena a la red neuronal en una sola ocasión, usando el aprendizaje de la ELM básica.

El lote inicial de datos de entrenamiento requerido puede ser pequeño e igual al número de neuronas ocultas. Por ejemplo, si hay 10 neuronas, es posible que

necesitemos 10 muestras de entrenamiento para el boosting (impulso) de aprendizaje.

La fase de entrenamiento secuencial ocurre inmediatamente después de terminar la fase de entrenamiento boosting, es decir, cuando recibe un nuevo lote de datos adicionales que permite recalculer los pesos y sesgos de la capa de neuronas ocultas mediante el entrenamiento RLS en lugar del ELM. Descarta los datos de entrenamiento inicial, de tal forma que el entrenamiento de la red se va actualizando cada vez que recibe nuevos lotes de información.

El diagrama de flujo de los tipos de entrenamientos que ocurren en la instancia CA1 se muestra en la Figura 24, pueden observarse las dos fases de entrenamiento de la red neuronal secuencial OS-ELM, la fase boosting y la fase secuencial.

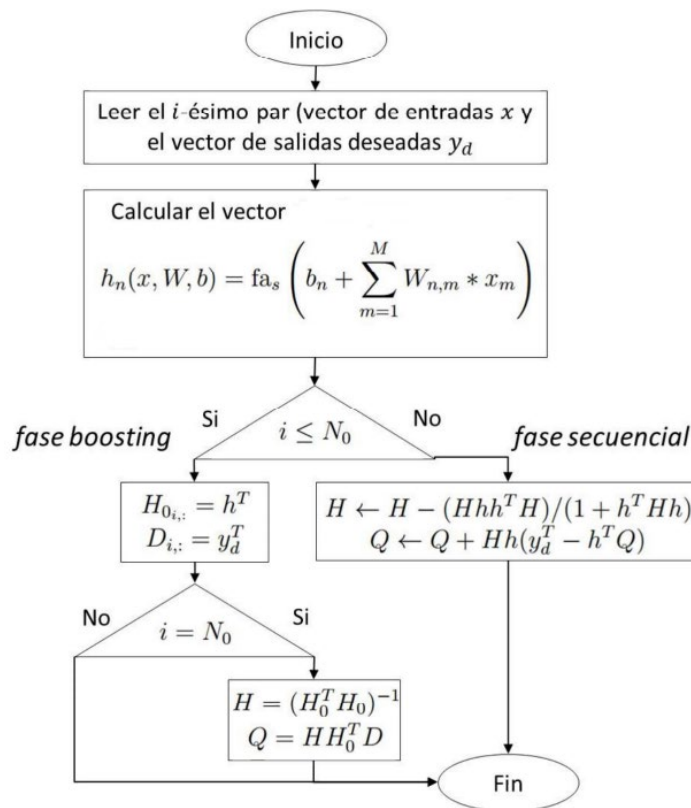


Figura 24. Diagrama de flujo de los 2 tipos de entrenamientos que ocurren en instancia CA1.

Elaboración propia.

La figura muestra el flujo que se usa para el entrenamiento, primero se reciben los vectores de entrada x y salida y_d ; después se calcula el vector $h_n(x, W, b)$, para determinar los pesos y sesgos de la neuronas de salida; después, dependiendo de si se alcanzó el número de muestras predefinidas N_0 , decide si se va a realizar un aprendizaje inicial acelerado denominado "boosting" o un entrenamiento "secuencial".

El número de muestras N_0 depende de cuantos datos se desean utilizar para el aprendizaje inicial, esto es crítico, ya que, si se usa un número muy grande, todos esos datos se olvidarán al pasar a la etapa secuencial, lo que podría genera un error de precisión en cambio de un modo de entrenamiento a otro y si es muy chico, afectará la precisión del entrenamiento inicial. Si fue seleccionado la fase de entrenamiento boosting se calculan $H(0)_{i,:}$ y $D_{i,:}$, que son los pesos de las neuronas ocultas.

Por otro lado, si se selecciona el entrenamiento de fase secuencial, los vectores H y Q se actualizarán después de que se cumplan los N_0 datos.

Para asegurar que la instancia hipocámpica $CA1$ no entre en un bucle de almacenamiento infinito, se utiliza un modelo de descarte de modelos de bajo desempeño que se divide en tres bloques: el bloque inicial, que genera la primera memoria de trabajo; el bloque que crea nuevos modelos y el que elimina los modelos de bajo desempeño.

En la Figura 25 se muestra el modelo de descarte de la instancia hipocámpica $CA1$, en ella se pueden observar los 3 bloques del modelo de descarte, marcados con líneas punteadas. Muestra a detalle el proceso de generación de modelos mediante la actualización de pesos en las neuronas ocultas del algoritmo secuencial. El bloque del lado izquierdo genera nuevos modelos con mayor desempeño. El bloque de la derecha muestra la sección que elimina los modelos de menor desempeño para evitar un bucle de crecimiento infinito. El modelo funciona primero cargando una memoria episódica proveniente de $CA3$, esta genera el primer modelo asociado a este evento episódico. Posteriormente, se manda a la instancia SUB , que reacciona al medio ambiente, predice una salida y

verifica el error; si el error es mayor a σ , crea un nuevo modelo. El último bloque verifica si hay muchos modelos, si se excede el número de modelos predefinido para no saturar la memoria del sistema de control, el algoritmo remueve los modelos almacenados de menor desempeño.

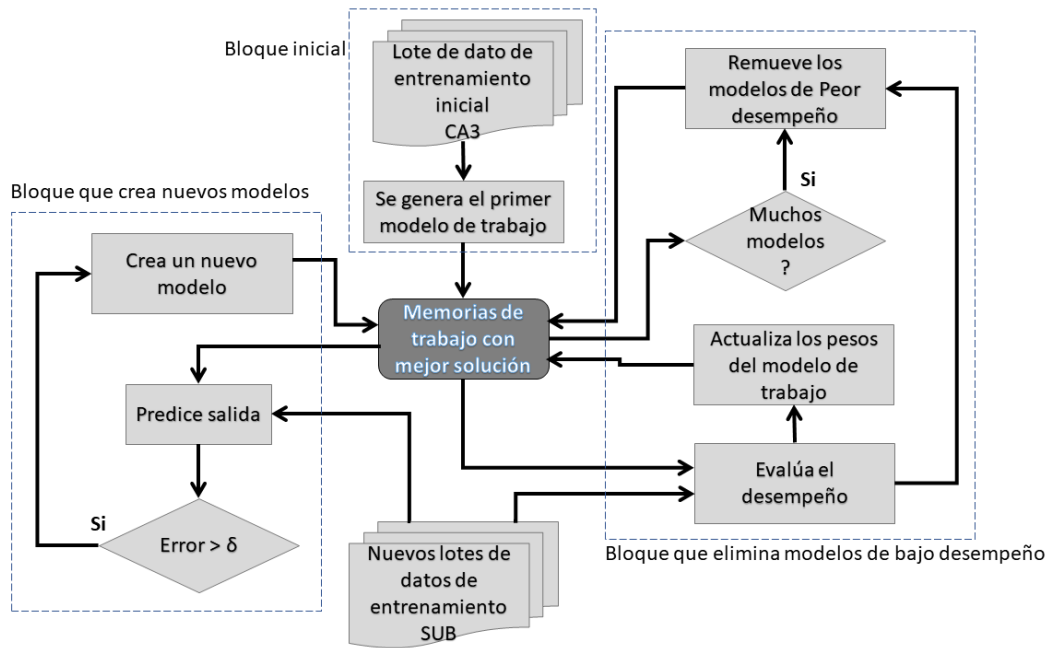


Figura 25. Modelo de descarte de la instancia hipocámpica CA1. Elaboración propia.

El objetivo del algoritmo de la instancia hipocámpica CA1 es recordar memorias episódicas y relacionarlas con la memoria de trabajo. La matriz β_m^0 contiene los pesos iniciales que utilizará esta red neuronal mientras que la matriz $\beta^{(k+1)}$ tiene los pesos de los nuevos modelos de mejor desempeño.

A continuación, el algoritmo 6 muestra la implementación en pseudocódigo de la instancia hipocámpica CA1.

Algoritmo 6 Instancia Hipocámpica CA1 (**Cornus Ammonis 1**)

Requiere: Que se inicialice la Instancia Hipocámpica **CORNUS AMMONIS 1 (CA1)**

Estrategia: La red recordará memorias episódicas junto con memoria de trabajo.

Salida: Patrones completos de recuerdo episódico, más la respuesta de memoria de trabajo inicial almacenada.

- 1: Cargar los patrones de estado inestable (X_1, X_1, \dots, X_P) , las longitudes de los vectores (X) son iguales que la longitud de los recuerdos episódicos fundamentales (Y) .
 - 2: Los Datos $\{X_i, Y_i\}, x_i \in R^d$ y $y_i \in R^q, i = 1, \dots, N$, número de neuronas i , número de nodos i , número de capas ocultas N , función de activación $g(x)$ y se espera una salida β^{k+1}
 - 3: **Inicializa CA3**, donde los datos iniciales son: $\{X_i, Y_i\}, x_i \in R^d$ y $y_i \in R^q, i = 1, \dots, N_0$ con $N_0 \in N$
 - 4: **for** $m = 1 \rightarrow p$ **do**
 - 5:
$$H_0^m = g\left(\left(\beta_m^{(0)}\right)^T H_0^{m-1}\right)$$
 - 6: Donde $\beta_m^{(0)} = \left(H^T H + \frac{I_{h_h}}{C}\right)^{-1} H^T X$
 - 7: **end for**
 - 8: $\beta^{(0)} = P_0 H_0^T Y^0$
 - 9: donde $H_0 = H_0^P$
 - 10: **Aprendizaje secuencial:**
 - 11: **for** $k = 0 \rightarrow K$ **do**
 - 12: **for** $m = 1 \rightarrow p$ **do**
 - 13:
$$H_{k+1}^m = g\left(\left(\beta_m^{(k+1)}\right)^T H_{k+1}^{m-1}\right)$$
 - 14: Donde $\beta_m^{(k+1)} = H^T \left(H H^T + \frac{I_N}{C}\right)^{-1} X$
 - 15: **end for**
 - 16: $\beta^{(k+1)} = \beta^{(k)} + P_{k+1} H^T (Y_{k+1} - H_{k+1} \beta^{(k)})$
 - 17: Donde $H_{k+1} = H_{k+1}^P$
 - 18: **end for**
-

Elaboración propia.

3.1.12 Interacción y aprendizaje del medio ambiente con instancia **SUB**

Para realizar la función de interacción y aprendizaje en la instancia Subiculum o SUB con su contraparte, la corteza entorrinal, se selecciona un algoritmo de Aprendizaje por Refuerzos Q-Learning (Sutton & Barto, 2018).

El Aprendizaje por Refuerzos es un proceso en el que un agente aprende, a través del tiempo, a comportarse de manera óptima en un determinado entorno mediante la interacción continua con el mismo. Durante su curso de aprendizaje el agente experimenta diversas situaciones en el entorno en que se encuentra, a dichas situaciones se les denomina estados. El agente, mientras se encuentra en un estado, puede elegir entre un conjunto de acciones permitidas que pueden obtener diferentes recompensas (o sanciones). Con el tiempo, el agente aprende a maximizar estas recompensas para comportarse de manera óptima en cualquier estado en el que se encuentre (Barto, 1994).

El algoritmo Q-Learning es un algoritmo tipo TD que da respuesta al problema general de aprendizaje de políticas óptimas. A partir de una estimación arbitraria de la función Q , el algoritmo Q-Learning emplea el conjunto $(s_k, a_k, s_{k+1}, r_{k+1})$ para actualizar la estimación de Q mediante la Ecuación 23:

$$Q_{k+1}(s_k, a_k) = Q_k(s_k, a_k) + \alpha_k [r_{k+1} + \gamma \max_{a'} Q_k(s_{k+1}, a') - Q_k(s_k, a_k)]$$

Ecuación 23. Q-Learning para instancia *SUB* .

$$Q^*(s, a) = \rho(s, a) + \gamma \max_{a'} Q^*(f(s, a), a')$$

Ecuación 24. Sustituyendo en ecu. Bellman muestreando.

siendo $\alpha_k \in (0,1)$ la secuencia de tasas de aprendizaje. El término entre corchetes es la diferencia temporal entre la nueva estimación de la función Q y la estimación actual. La nueva estimación se calcula con el operador de optimalidad de Bellman muestreando, es decir, se sustituye en la Ecuación 24 la función de recompensa $\rho(s_k, a_k)$ por su valor observado r_{k+1} y el estado siguiente $f(s_k, a_k)$ por el estado siguiente observado s_{k+1} (Jang, Kim, Harerimana, & Kim, 2019).

El principio de funcionamiento del algoritmo consiste en estimar, de forma iterativa, la función Q óptima. Para que el algoritmo converja se requiere que todos los pares

estado-acción de la función Q sean actualizados indefinidamente, es decir, requiere una política exploratoria y, por otra parte, que la secuencia de tasas de aprendizaje satisfaga las condiciones de convergencia; cabe señalar que no es necesario que el nivel de exploración se reduzca conforme aumenta el número de iteraciones. El algoritmo converge hacia una política óptima independientemente de la política que emplee el agente para interactuar con el entorno se puede considerar que es del tipo *off-policy*.

La Figura 26, muestra el diagrama de flujo del algoritmo de la instancia hipocámpica *SUB*; en el se pueden observar los flujos de datos, empezando por los datos de entrada XCA_1 , que provienen de la instancia hipocámpica *CA1*. Los datos contienen los patrones dispersos recuperados y la mejor respuesta para la memoria de trabajo. También puede verse el bucle de aprendizaje en el que se realiza la evaluación de las acciones con respecto al error ϵ_{sub} , para determinar si se logra el aprendizaje o se requiere una nueva acción en los actuadores. La instancia *SUB* recibe los datos XCA_1 del algoritmo OS-ELM. La respuesta del algoritmo se envía a los actuadores y evalúa la respuesta XCA_1 con respecto al error permitido ϵ_{sub} , si el error es mayor al permitido, una nueva respuesta se manda a la instancia *SUB*.

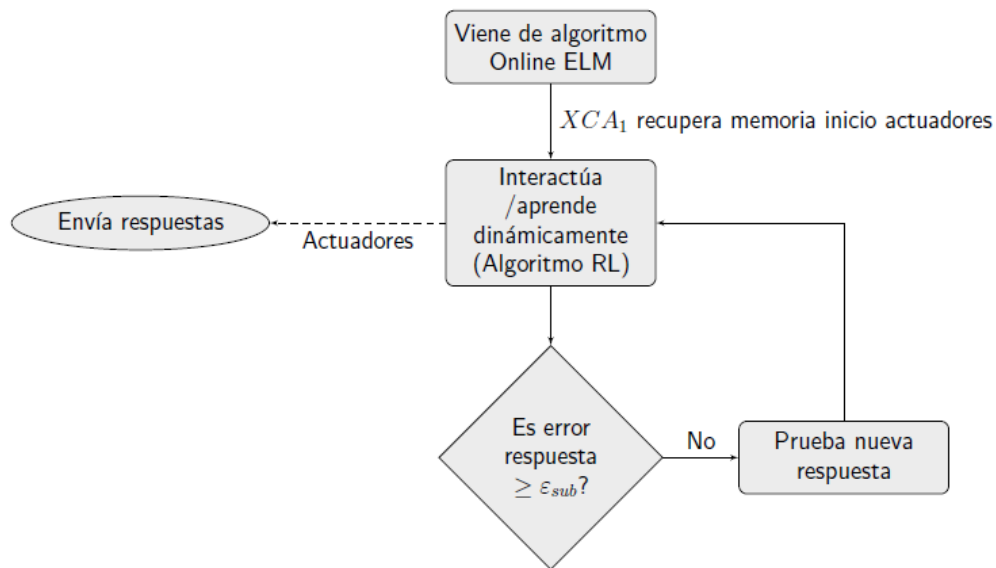


Figura 26. Diagrama de flujo del algoritmo de la instancia hipocámpica *SUB*.
Elaboración propia.

El algoritmo Q-Learning se selecciona para la instancia hipocámpica Subiculum en tanto que se puede comportar como la instancia biológica, al interactuar con la corteza entorrinal, permitiendo acceder a información sensorial y reaccionando, mediante la salida del algoritmo al entorno, modificándolo y recibiendo nueva información sensorial (estados). El Aprendizaje por Refuerzos utiliza valores Q (también llamados valores de acción), para mejorar iterativamente el comportamiento del agente.

Los valores Q , o valores de acción, se definen para estados y acciones. $Q(S, A)$ es una estimación de qué tan buena es una acción A , tomada en el entorno o estado S . La estimación de $Q(S, A)$, se calculará iterativamente utilizando la regla TD-Update (*Time Difference Update*) (Tsitsiklis & Van Roy, *An analysis of temporal-difference learning with function approximation*, 1997).

En el Aprendizaje por Refuerzos, se utiliza la separación de bloques de aprendizaje mediante episodios. Al iniciar el episodio, el agente empieza desde un estado inicial, realiza una serie de transiciones desde su estado actual al siguiente estado, en función de su elección de acción y también del entorno en el que está interactuando. En cada paso de transición, el agente de un estado toma un curso de acción, observa una recompensa del entorno y después transita a otro estado. Si en algún momento el agente termina en uno de los estados de terminación, eso significa que no hay más transición posible. Se dice que esto es la finalización de un episodio.

La instancia *SUB* utiliza la regla de aprendizaje *diferencia temporal* o TD, que se puede representar según muestra la Ecuación 25:

$$Q(s, a) \leftarrow Q(S, A) + \alpha(R + \gamma Q(S', A') - A(S, A))$$

Ecuación 25. Regla de aprendizaje diferencia temporal.

Esta regla de actualización para estimar el valor de Q , se aplica en cada paso de la interacción de los agentes con el entorno. Los términos utilizados en la Ecuación 25 corresponden a:

- S : Estado actual del agente.
- A : Acción actual elegida de acuerdo con alguna política.
- S' : Siguiendo estado donde termina el agente.
- A' : La siguiente mejor acción que se elegirá, utilizando la estimación actual del valor Q , es decir, se elige la acción con el valor Q máximo en el siguiente estado.
- R : Recompensa actual observada del entorno en respuesta a la acción actual.
- $\gamma (> 0 \text{ y } \leq 1)$: Factor de descuento para recompensas futuras. Las futuras recompensas son menos valiosas que las recompensas actuales, por lo tanto, deben descontarse. Dado que el valor Q es una estimación de las recompensas esperadas de un estado, la regla de descuento también se aplica aquí.
- α : Longitud del paso tomado para actualizar la estimación de $Q(S, A)$.

Para elegir la acción a tomar se emplea la política $\epsilon - greedy$ (dos Santos Mignon & de Azevedo da Rocha, 2017), que consiste en una política muy simple para elegir acciones utilizando las estimaciones actuales del valor Q y funciona de la siguiente manera:

- Con probabilidad $(1 - \epsilon)$ se elige la acción que tenga el valor Q más alto.
- Con probabilidad (ϵ) se elige cualquier nueva acción al azar.

Para implementar el algoritmo de Aprendizaje por Refuerzos a la instancia hipocámpica *SUB*, se adaptaron: el agente, la función de recompensa, el entorno, los estados y las acciones; ello con el fin de recibir la información de la instancia hipocámpica *EC* y *CA1* de manera adecuada.

La primera adaptación se realiza al elemento entorno, que deja de ser entorno y se denomina, ahora, memoria de trabajo. Tiene dos segmentos: el primero contiene la información de entrada, proveniente de la memoria sensorial y el segundo

corresponde a la información que se envía hacia la memoria de salida que interactúa con los actuadores. En conjunto, estos dos segmentos de memoria reemplazan en su totalidad al entorno y representan las entradas y salidas del proceso industrial de manufactura que se desea controlar.

La segunda adaptación se realiza al agente, que se compone de las políticas que permiten mejorar el entorno y las limitaciones que tiene. Cada política que se implementa responde al proceso único que se esté controlando, esto se considera como una configuración inicial de la red hipocámpica del proceso.

La tercera adaptación se realiza en la función de recompensa. Se implementa una función que sea capaz de representar la eficiencia del proceso que se desea controlar, esto se consigue con una variable continua, que es constantemente actualizada desde la memoria de trabajo.

El modelo de aprendizaje de la instancia hipocámpica *SUB* utilizado, se muestra en la Figura 27. Se basa en un algoritmo de Aprendizaje por Refuerzos. El agente y la función de recompensa están implementados con el algoritmo Q-Learning.

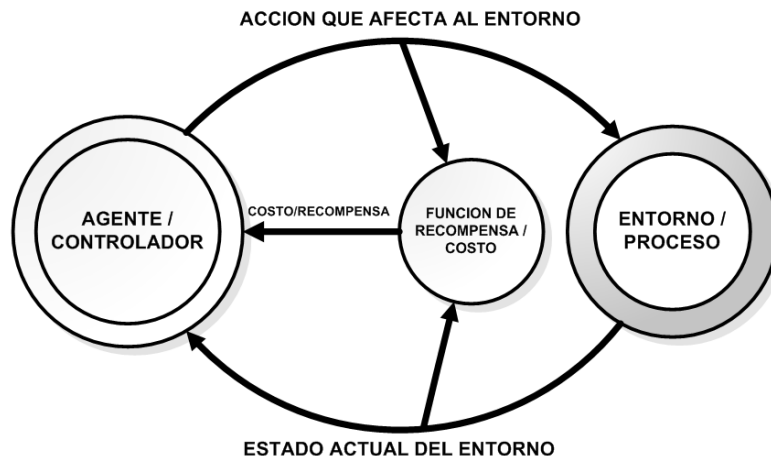


Figura 27. Modelo de aprendizaje de la instancia hipocámpica subiculum. Elaboración propia.

Se muestran en el modelo los flujos de estado del entorno, así como las acciones de la instancia hipocámpica hacia el entorno, en un ciclo regulado por la función de recompensa/costo. La información del entorno proviene de las instancias *CA1* y la memoria de trabajo, la instancia hipocámpica *SUB* recibe y envía información al medio ambiente. Usualmente la memoria de trabajo se interconecta con la instancia *EC*, que es la sección de entrada de sensores. La información de desempeño y la salida se dan en forma de instrucciones que son enviadas a motores, cilindros neumáticos u otros sistemas de interacción, como modificadores o actuadores.

Una vez que la instancia hipocámpica *SUB* recibe una memoria episódica desde *CA1*, procesa la memoria de trabajo, que contiene la última mejor respuesta al medio ambiente, y la asigna al conjunto A' . A partir de este punto, el algoritmo empieza a evaluar el valor de recompensa R y a elegir las acciones de acuerdo con la política $\epsilon - greedy$. La instancia busca optimizar el proceso situado en el medio ambiente, buscando la mejor recompensa R .

La Figura 28, es una gráfica de tendencia del aprendizaje del algoritmo de la instancia *SUB*, muestra el número de acciones necesarias para resolver el entorno en intentos de solución. El eje de las X representa el número de intentos de solución, antes de llegar a la mayor recompensa R ; el eje de las Y representa el número de acciones necesarias para lograr el menor error posible. El número de acciones va disminuyendo a medida que el algoritmo incrementa el número de intentos. Lo que refleja que va aprendiendo por experiencia. Al inicio del entrenamiento, el número de acciones necesarias para obtener un proceso óptimo es alto, ya que inicia sin valores de referencia. La solución se logra mediante acciones aleatorias, pero a medida que el valor Q mejora, el número de acciones se va reduciendo hasta llegar a un óptimo. Hacia la izquierda de la gráfica se puede observar que se requiere de una gran cantidad de acciones para llegar al óptimo y hacia la derecha una línea que inicia con el valor optimizado cada vez que la instancia hipocámpica *SUB* busca optimizar el proceso.

La memoria de trabajo contiene el conjunto de acciones optimizadas para una memoria episódica dada y se representa como el vector Q que, después de ser actualizado a su mejor valor, se transfiere a la instancia hipocámpica $CA1$.

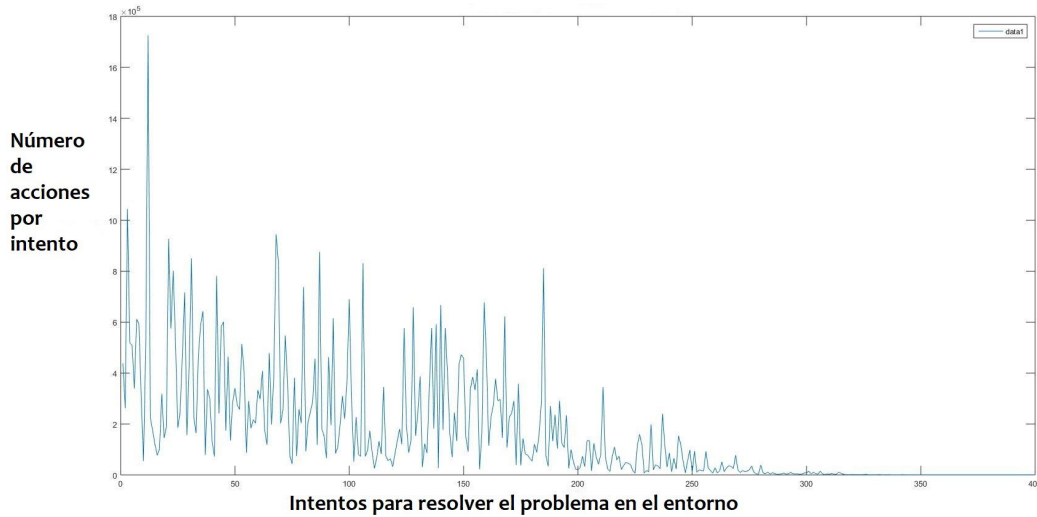


Figura 28. Gráfica de tendencia del algoritmo instancia hipocámpica *SUB*.
Elaboración propia.

La Figura 29, muestra la representación de 3 vectores de 1024 bits como matrices de 32 x 32.



Figura 29. Tres ejemplos de vectores de solución provenientes de la instancia hipocámpica subiculum (*SUB*).
Elaboración propia.

Las 3 matrices representan las respuestas de la instancia hipocámpica Subiculum al entorno bajo diferentes condiciones. La memoria de trabajo se carga en la instancia hipocámpica *SUB*, para iniciar el proceso de Aprendizaje por Refuerzos. En dicho proceso se llega a un momento en que la memoria de trabajo ya no

tendrá cambios, porque ha llegado a su máxima optimización. Cuando se termina el lote en proceso, entonces la información regresa a la instancia CA1 para ser almacenada junto con la memoria episódica y generar una memoria de largo plazo.

El Algoritmo 7 se implementa para la instancia hipocámpica SUB. Se utiliza como base el modelo de Aprendizaje por Refuerzos. Para su configuración, requiere el factor de descuento y las secuencias de aprendizaje y de exploración.

Algoritmo 7 Instancia Hipocámpica SUB (SUBICULUM)

Requiere: Factor de descuento γ , Secuencia de aprendizaje α_k , Secuencia de exploración ϵ_k

- 1: Inicializar estimación de función Q arbitrariamente, ejemplo $Q_0 \leftarrow 0$
 - 2: **repeat**
 - 3: Medir estado inicial s_0
 - 4: **for** Cada paso del episodio **do**
 - 5:
$$a_k = \begin{cases} a \in \operatorname{argmax}_a Q - k(s_k, a) & \text{con probabilidad } (1 - \epsilon_k) \\ \text{acción uniformemente aleatoria en A} & \text{con probabilidad } (\epsilon_k) \end{cases}$$
 - 6: Aplicar a_k , observar el estado siguiente s_{k+1} y la recompensa r_{k+1}
 - 7: $Q_{k+1}(s_k, a_k) = Q_k(s_k, a_k) + \alpha_k [r_{k+1} + \gamma \max_{a'} Q_k(s_{k+1}, a') - Q_k(s_k, a_k)]$
 - 8: **end for**
 - 9: **until** Cumplir condiciones de convergencia
 - 10: **return** π^*
-

Elaboración propia.

Se inicializa la función de estimación Q y se cargan valores aleatorios en el primer ciclo de aprendizaje; posteriormente, al tener memorias episódicas, se carga la mejor respuesta almacenada, proveniente de la instancia hipocámpica CA1. A partir de aquí, el algoritmo de la instancia hipocámpica Subiculum inicia un proceso iterativo de prueba y error, dependiente del resultado de la función de recompensa. Se genera entonces un proceso de convergencia de los errores, es decir se reducen hasta llegar a su mínima expresión, a medida que las acciones del agente se van ejecutando.

3.1.13 Conclusión algoritmo AHN-ELM

La sección 3.1 nos introduce al desarrollo del algoritmo híbrido de Control Cognitivo: la Red Hipocámpica Artificial soportada por la Máquina de Aprendizaje Extremo o Red Neuronal AHN-ELM.

Se explica la implementación de cada una de las cuatro instancias hipocámpicas, *DG*, *CA3*, *CA1* y *SUB*, así como su relación con el subsistema de entradas-salidas, denominado corteza entorrinal o *EC*.

Se puede concluir que el Algoritmo Híbrido tiene cuatro funciones principales en el manejo del conjunto de datos que alimentan a la red: la separación de patrones similares en *DG*; la auto asociación de patrones episódicos en *CA3*; la recuperación de patrones completos, incluyendo el patrón episódico y del entorno, en *CA1* y la interacción de actuación con el entorno en la instancia *SUB*.

3.2 DESARROLLO DE ALGORITMO HÍBRIDO PAT-ELM PARA LA CARACTERIZACIÓN DE PROCESOS

3.2.1 Introducción

Se define como caracterización al proceso de identificación de los materiales midiendo sus principales propiedades físicas, mecánicas o estructurales; existen para ello diferentes técnicas de medición dependiendo la característica específica que se quiere comprobar. El objetivo de una caracterización de materiales es conocer características tales como: estructura, composición, morfología, propiedades térmicas, dureza, flexibilidad, etc. Existen varios métodos para la caracterización de materiales, usualmente están referenciados en las normas técnicas internacionales tales como ISO y ASTM. Estas normas especifican cómo se deben realizar las mediciones y que tipo de herramienta debe utilizarse para que los valores obtenidos sean válidos en cualquier laboratorio.

Los algoritmos de caracterización se utilizan para determinar los rangos válidos de ajuste de la materia prima en procesos industriales.

El algoritmo Híbrido PAT-ELM (*Part Average Testing based on Extreme Learning Machine*) es una variante del Algoritmo Híbrido AHN-ELM, presentado en la sección 3.1 de esta tesis, tiene como propósito realizar la caracterización de procesos industriales.

Puede aplicarse para predecir las características principales de los materiales usados en los procesos. Una parte importante de este algoritmo es que se entrena usando un dataset que se genera basado en alguna norma de medición, como por ejemplo la ASTM.

El cambio que se hace a la instancia hipocámpica *DG* da lugar a un nuevo Algoritmo Híbrido al que denominaremos PAT-ELM, capaz de eliminar los datos anormales y ser entrenado en un mínimo de tiempo.

Este factor es especialmente importante para la industria, puesto que grandes volúmenes de datos de entrenamiento pueden traducirse en pérdidas por desecho de partes y/o tiempos de espera muy largos para la obtención de pronósticos.

El Algoritmo Híbrido PAT-ELM se pensó para un funcionamiento basado en un muestreo separado de la línea y tiempos de producción, lo que es muy idóneo para el monitoreo o caracterización, es muy rápido en su entrenamiento y pruebas, debido a la naturaleza analítica del entrenamiento que utiliza a diferencia de las redes iterativas.

Un factor atractivo del Algoritmo Híbrido PAT-ELM es que sólo requiere que se tenga definido el número de nodos de entrada, el número de neuronas ocultas y de salida en sus capas y el error permitido, no requiere otra configuración inicial.

Una de las principales razones por las que se requiere realizar la caracterización de los lotes de materia prima es porque existen variaciones en las características de esta, que pueden afectar el desempeño óptimo del proceso. Como ejemplo podemos observar la Figura 30, que muestra las variaciones de temperatura en varios lotes de un proceso industrial real. La gráfica de tendencia muestra en un mismo color la materia prima que en teoría debería ser idéntica. Se puede observar que a medida que pasa el tiempo, se presentan variaciones tanto en los valores promedio como en la dispersión de cada lote de producción. Una estrategia muy

utilizada en manufactura es caracterizar la materia prima antes ser procesada para poder realizar el ajuste adecuado en el equipo automático o la máquina que realiza el proceso, para compensar las variaciones de los materiales. Por todo lo antes expuesto se puede entender la necesidad de tener un sistema que pueda predecir tanto la desviación estándar como la dispersión que tiene un lote de materia prima antes de ser procesado. Esta es la función del Algoritmo Híbrido PAT-ELM.

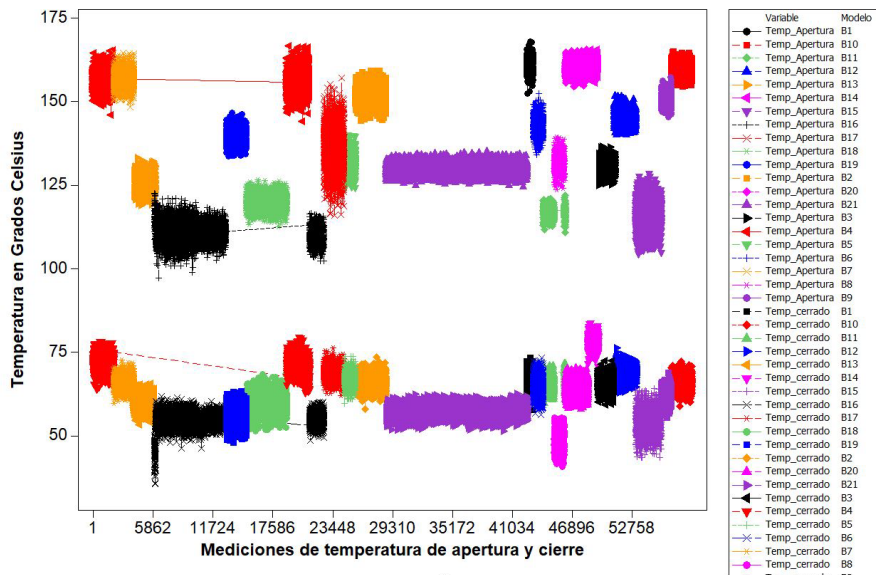


Figura 30. Ejemplo de la variación de la característica térmica de varios lotes de producción.
Elaboración propia.

3.2.2 Modificación de la instancia *DG* para caracterización

Para crear el Algoritmo Híbrido PAT-ELM a partir del Algoritmo Híbrido AHN-ELM, es necesario utilizar solo la instancia *DG* ya que la funcionalidad de las instancias *CA1*, *CA3* y *SUB* solo se utilizan para el control y la memorización del entorno. También es necesario modificar la *DG* para lograr un aprendizaje puntual, que es necesario ya que se va a usar el algoritmo para hacer un pronóstico basado en una medición inicial y no se quiere alterar este aprendizaje posteriormente. La versión original de la instancia *DG* consta de dos partes: la primera es un algoritmo de separación de datos anormales y la segunda un algoritmo para separación de patrones similares.

La parte de separación de datos anormales se utiliza para las aplicaciones de control, así como para las de caracterización, por lo que esta sección de la instancia se deja igual para ambas.

La modificación hecha a la instancia *DG* consistió en cambiar la segunda parte del algoritmo, en lugar de tener un algoritmo de separación de patrones mediante autocodificador basado en ELM, se deja un algoritmo de aprendizaje básico basado en la misma ELM.

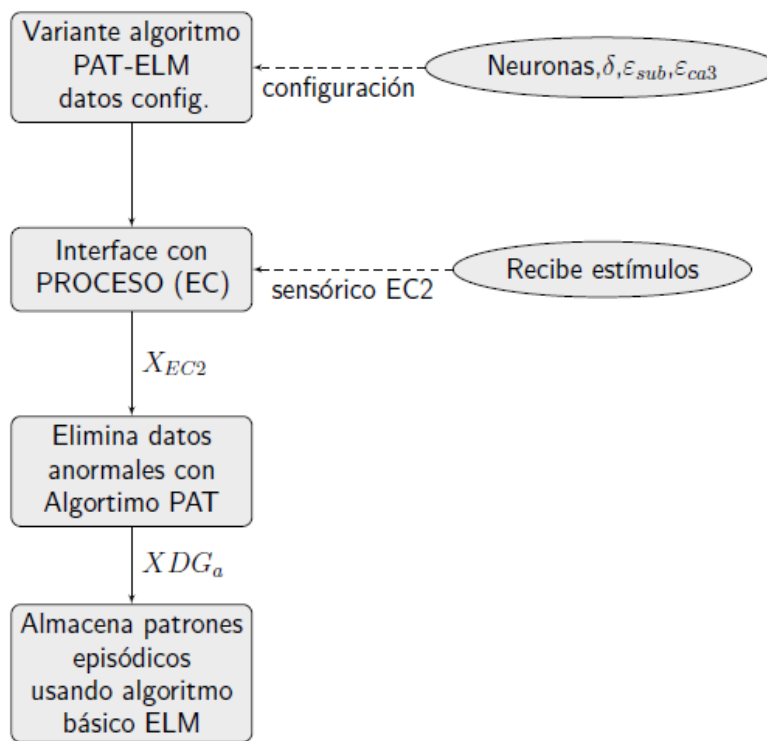


Figura 31. Diagrama de flujo la red neuronal PAT-ELM para el monitoreo cognitivo. Elaboración propia.

El algoritmo original es una máquina de aprendizaje extremo configurado como un *autoencoder* tipo *sparse* (ELM-AE).

La modificación consistió en reemplazar el algoritmo autocodificador para dejar la máquina de aprendizaje extremo en su configuración básica (ELM).

La Figura 31, muestra el diagrama de flujo de la Red Neuronal PAT-ELM. El primer bloque son las configuraciones iniciales del algoritmo PAT-ELM. El segundo bloque muestra la entrada de datos del entorno a aprender al realizar la interfaz con el proceso a monitorear. El tercer bloque se encarga de eliminar los patrones anormales mediante el algoritmo PAT y por último el bloque que representa a la máquina de aprendizaje extremo o ELM, que es la red neuronal que realiza el aprendizaje de los datos libres de patrones anomalías.

3.2.3 Arquitectura de PAT-ELM

La arquitectura de la Red Híbrida PAT-ELM se muestra en la Figura 32.

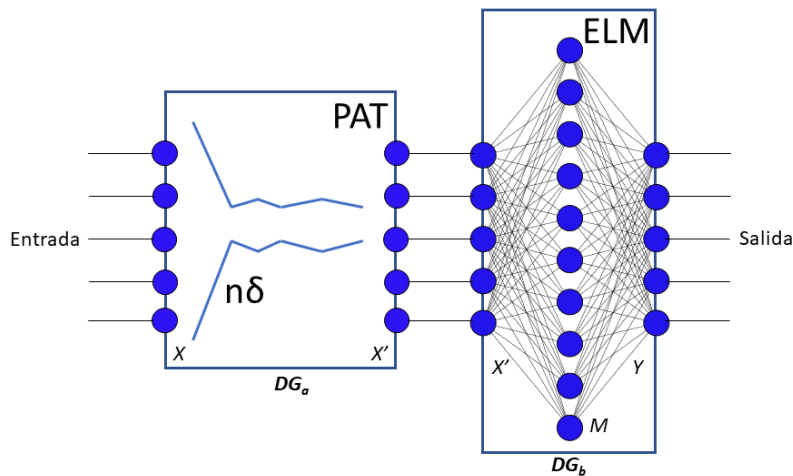


Figura 32. Arquitectura de algoritmo híbrido PAT-ELM.
Elaboración propia.

Se puede observar en el lado izquierdo la red de filtro PAT denominada DG_a y a la derecha el algoritmo ELM denominado DG_b . El algoritmo PAT elimina los datos que están fuera de n desviaciones estándar definidas como $n\sigma$.

El algoritmo PAT tiene el mismo número de nodos tanto a la entrada como a la salida $X = X'$, el parámetro inicial es el número de desviaciones estándar que servirán de filtro a las anomalías, se representa como n . La capa de entrada tiene una dimensión X' y la capa de neuronas de salida tiene una dimensión Y , la capa de neuronas ocultas tiene una dimensión M .

La Ecuación 26 muestra la media robusta, la ecuaciones Ecuación 27 y la Ecuación 28 son los límites superior e inferior, aplicando la sigma robusta.

$$\overline{x_{robusta}} = Q_{1,3} - \frac{IQR}{2}$$

Ecuación 26. Media robusta de DG_a .

$$PAT_{limiteinferior} = Q_1 - \frac{(x_i - 0.675)}{1.35} (Q_3 - Q_1)$$

Ecuación 27. Límite dinámico inferior de DG_b .

$$PAT_{limitesuperior} = Q_3 + \frac{(x_i - 0.675)}{1.35} (Q_3 - Q_1)$$

Ecuación 28. Límite dinámico superior de DG_b .

$$x_{PAT} = \begin{cases} \rho = x_\alpha = Q_1 - \frac{(x_i - 0.675)}{1.35} (Q_3 - Q_1) \rightarrow \text{Borrar} \\ \nu = x_\alpha = Q_3 + \frac{(x_i - 0.675)}{1.35} (Q_3 - Q_1) \rightarrow \text{Borrar} \\ \rho < x_\alpha < \nu \rightarrow x_{DG_a} = x_\alpha \end{cases}$$

Ecuación 29. Intervalos válidos en donde se eliminan los datos atípicos.

La Ecuación 29 muestra la salida de los intervalos válidos en donde se eliminan los datos atípicos y solo deja pasar los datos filtrados.

$$x_{DG_a} = x_{filtrado} - \rho - \nu$$

Ecuación 30. Salida filtrada de de datos atípicos DG_a .

La máquina de aprendizaje extremo genera la predicción usando la salida de la i – *esima* neurona de salida con respecto a la entrada x . Se muestra en la Ecuación 31 el nodo oculto aditivo con la función de activación $g(x)$ (Haykin, 2009).

$$f(x) = \sum_{i=1}^L \beta_i G(a_i, b_i, x) = t_j$$

Ecuación 31. Nodo oculto aditivo con la función de activación $g(x)$ de una SLFN.

La ecuación anterior puede ser escrita en forma vectorial, como se muestra en la Ecuación 32 (Kasun, Zhou, Huang, & Vong, 2013):

$$\beta = H^\dagger T.$$

Ecuación 32. Pesos de salida calculados con la matriz Inversa de Moore-Penrose.

H^\dagger es la matriz de salida de la capa oculta de la red neuronal, que a su vez es la matriz Inversa de Moore-Penrose. La columna i de la misma es la salida del nodo oculto i , con respecto a las entradas x_1, x_2, \dots, x_N ; $h(x)$ que es el mapeo de las características o parámetros de la capa oculta (Haykin, 2009).

El algoritmo 8 muestra una versión reducida del pseudocódigo del Algoritmo Híbrido PAT-ELM. La versión completa del pseudocódigo está disponible en el anexo 2.

Algoritmo 8 Algoritmo Híbrido propuesto para simulador PAT-ELM**Requiere:** Carga la base de datos $x_{PAT}(N.x, N.y)$

- 1: Calcula la variación natural basada en t muestras.
- 2: Elimina los datos anormales de fuera de $+/- n\sigma$.
- 3: Genera la matriz filtrada $x_{PAT}(N.x, N.y)$.
- 4: Inicializa nodos, capa oculta L , y activación $g(x)$.
- 5: Lee $N = \{x_i, t_i \mid x_i \in R^n, t_i \in R^m, i = 1, \dots, N\}$
- 6: Define si es una regresión o una categorización.
- 7: Asigna pesos w_i y sesgos b_i aleatoriamente.
- 8: Calcula matriz capa oculta H basada en función $g(x)$.
- 9: Calcula los pesos de salida $\beta : \beta = H^\dagger$ donde H^\dagger es la matriz inversa generalizada de Moore-Penrose H .

-

Elaboración propia.

3.2.4 Conclusiones algoritmo híbrido PAT-ELM

En esta investigación proponemos un novedoso sistema de caracterización que puede ser utilizado en aplicaciones industriales. Está basado en un algoritmo de máquina de aprendizaje extremo ELM y el filtrado dinámico por prueba de promedios parciales PAT. Una de las principales características por las que puede ser utilizado en ambientes industriales o de manufactura es su capacidad de eliminar de manera simple los datos anormales que podrían alterar las predicciones de algún proceso o materia prima en evaluación. Otra importante característica que lo hace compatible con los procesos industriales es su gran velocidad de aprendizaje que permite evitar desperdicios excesivos durante la etapa de entrenamiento.

Una de las posibles aplicaciones de caracterización podría ser el pronóstico de mediciones basadas en normas como la ASTM.

3.3 EXPERIMENTOS VIRTUALES Y DE LABORATORIO AHN-ELM

3.3.1. Introducción a experimentos virtuales y de laboratorio

Las secciones 3.3 y 3.4 presentan los experimentos virtuales y de laboratorio a los que se sometió a la Red Neuronal Hipocámpica Artificial AHN-ELM propuesta, realizados para validar su funcionamiento y generar así, conocimiento sobre las interfaces de comunicación que se emplearán después en los experimentos de aplicaciones reales en manufactura industrial.

El primer experimento virtual que se presenta es un laberinto que el Algoritmo Híbrido AHN-ELM debe resolver y es realizado en un ambiente de simulación. Después, se presenta un experimento que se realiza en un ambiente de laboratorio, en el que se emplea un robot explorador, basado en un sistema embebido Arduino y un sistema de guía ultrasónico.

En la sección 3.3 se describe la implementación experimental descrita en la sección 3.1; para validar el funcionamiento del Algoritmo Híbrido AHN-ELM. Los resultados de esta sección proporcionan las bases técnicas sobre la mejor forma de implementar las interfaces de comunicación entre la Red AHN-ELM y el mundo exterior.

3.3.2. Experimento con laberinto virtual

Los laberintos virtuales se utilizan ampliamente para verificar el correcto funcionamiento de algoritmos basados en la técnica de Aprendizaje por Refuerzos. En este caso, se selecciona un laberinto como experimento virtual, debido a que permite determinar cuantitativamente la velocidad de aprendizaje, mediante la medición del número de movimientos realizados para alcanzar la posición "Meta". Para solucionar un laberinto como un problema de aprendizaje, el algoritmo debe de ser capaz de resolverlo sobre la marcha, es decir, sin el uso de preparación previa o de algún tipo de aprendizaje supervisado. Los algoritmos que pueden resolver este tipo de problema tienen gran similitud con el objetivo que busca la Red Neuronal Hipocámpica AHN-ELM. El laberinto es capaz de proporcionar información del desempeño a cada acción que realiza el algoritmo, como no

existe una solución conocida inicialmente, el algoritmo que resuelve el laberinto debe probar, aprender y memorizar las mejores acciones para lograr la posición meta.

Se selecciona el programa de laberinto virtual, desarrollado por Muhammad Asad Ali (Asad Ali, 2018) en Laboratorio Masatoshi Ishikawa de la Universidad de Tokio. El código del programa de laberinto se implementa con un robot, que se representa mediante el cursor, el robot tiene que llegar a su destino moviéndose únicamente en las direcciones permitidas izquierda, derecha, arriba y abajo. A cada paso se retroalimenta al algoritmo si el movimiento realizado lo acercó o no más a su meta. El proceso se repite una y otra vez hasta que llega a su destino. En este punto, el proceso comenzará nuevamente, para que se pueda verificar lo aprendido y se olviden los movimientos innecesarios realizados en los recorridos anteriores.

3.3.3. Propósito del experimento AHN-ELM con laberinto virtual

El experimento para resolver el laberinto virtual tiene como objetivo principal demostrar que el código de la Red Neuronal Hipocámpica AHN-ELM es capaz de encontrar una solución a un problema de aprendizaje. Lo que resulta fundamental para el proyecto que se presenta, ya que la principal función que se espera lograr al emular el hipocampo por medio de redes neuronales artificiales es mejorar los distintos procesos que puedan beneficiarse del aprendizaje automático y la resolución de problemas por medio de Inteligencia Artificial, que en este caso se aplican a la industria de la manufactura. El experimento presentado es un paso previo en el proceso de verificación del funcionamiento de la propuesta.

3.3.4. Procedimiento para experimentación con laberinto virtual

El experimento del laberinto está diseñado para ser realizado totalmente en un ambiente virtual. Para realizar la prueba se selecciona la plataforma Matlab. El procedimiento experimental es el siguiente:

1. Primero, se selecciona un algoritmo de laberinto que sea compatible con MatLab.
2. Se crean 5 modelos de laberinto virtual a resolver.
3. Se definen los parámetros de la Red Neuronal Hipocámpica AHN-ELM para *DG*, *CA3*, *CA1* y *SUB*. En especial, se trabaja con la tasa de aprendizaje *SUB* : α y γ para obtener el menor número de ciclos de aprendizaje.
4. Se ejecuta la Red Neuronal en los 5 laberintos y se registra el número de acciones y número de intentos realizados hasta llegar a la solución óptima.
5. Se registran los resultados y se analizan estadísticamente los ciclos de aprendizaje necesarios en cada laberinto, así como su relación con la tasa de aprendizaje *SUB*: α y γ .

3.3.5. Implementación AHN-ELM para solución de laberinto virtual

El tamaño y complejidad del laberinto están determinados en un archivo de texto, en el que se define en dónde estarán localizados tanto la posición inicial del robot como el punto objetivo del laberinto y los obstáculos o movimientos prohibidos para el robot. En la configuración inicial también se puede definir el tamaño del arreglo con el programa de MatLab. Para el experimento se selecciona una configuración de 9 x 9, se busca encontrar la solución en tiempos cortos, sin la necesidad de emplear equipo de cómputo de mucha potencia.

La estructura del laberinto utilizado en la experimentación se muestra en la Figura 33, se muestran, en color verde la ruta válida del laberinto; en azul, las restricciones del entorno; en amarillo claro, la ubicación del robot y en amarillo sólido la meta a conseguir. El laberinto se desarrolla en la plataforma MatLab.

El laberinto está formado por celdas. Las celdas azul oscuro representan los movimientos prohibidos para el robot, las celdas en color verde turquesa son las celdas en las que el cursor o robot puede moverse con libertad, y la celda amarillo sólido es la meta para lograr. La posición actual del robot está representada por una celda de color amarillo claro.

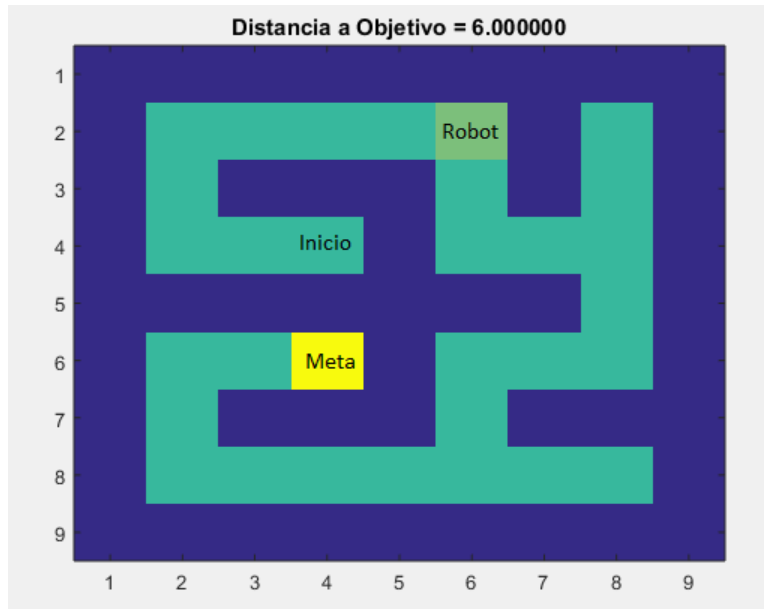


Figura 33. Laberinto virtual empleado para realizar una primera prueba al algoritmo híbrido AHN-ELM.
Elaboración propia.

Se generan 5 geometrías diferentes del laberinto para la experimentación, respetando el tamaño de bloques de 9 x 9, para comparar posteriormente el número de ciclos de aprendizaje necesarios para obtener una solución estable.

La Figura 34 muestra las 5 geometrías del laberinto que se emplearon en el experimento virtual.

Las configuraciones necesarias para la implementación de la Red AHN-ELM a la máquina prototipo, consistieron en definir los parámetros iniciales de las instancias hipocámpicas. Es decir, en definir la cantidad de neuronas ocultas y las funciones de activación de cada instancia hipocámpica, tal como se describió en la sección 3.1. En general son los datos iniciales que se van a emplear en las instancias hipocámpicas *DG*, *CA3*, *CA1* y *SUB*.

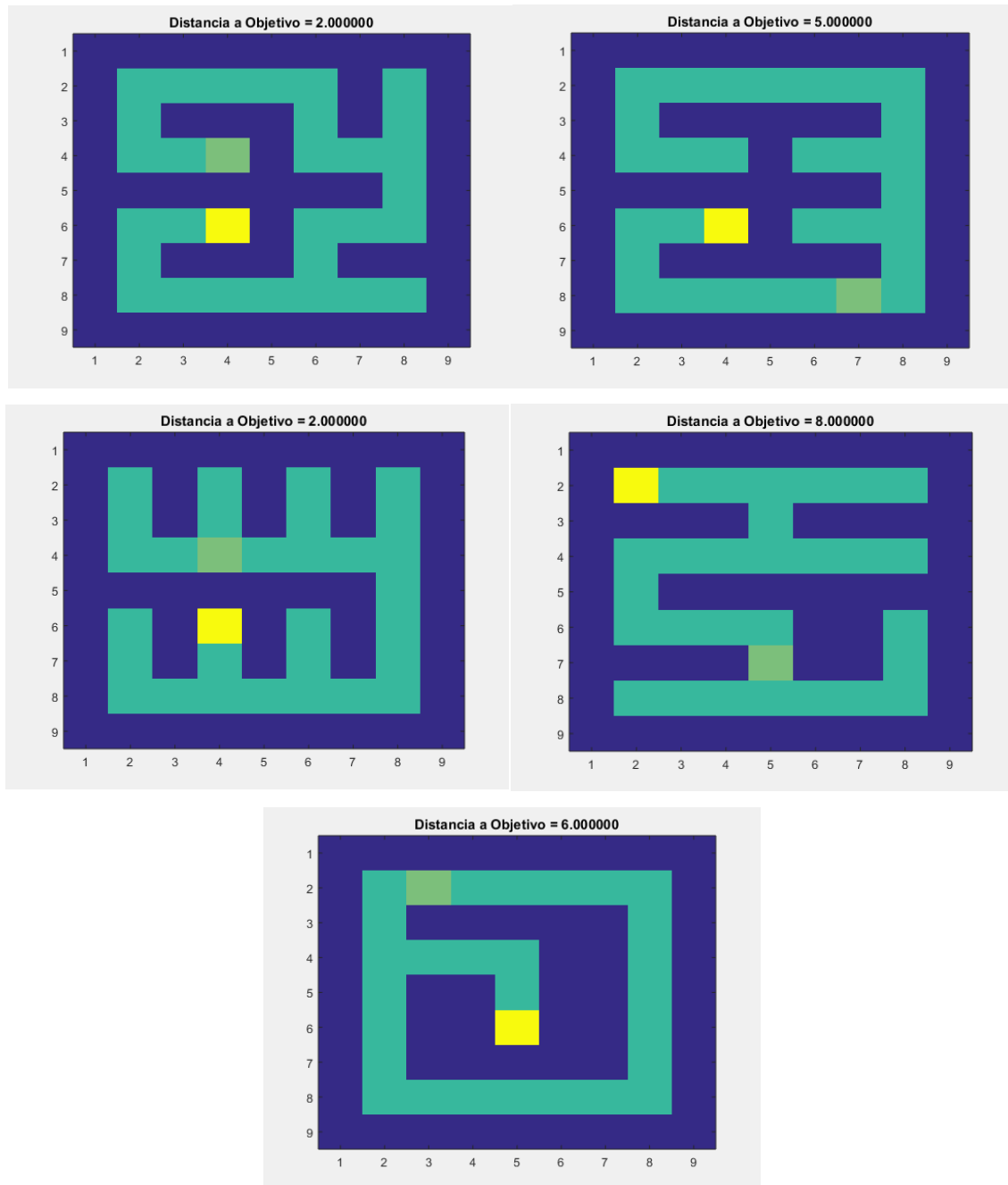


Figura 34. Diferentes geometrías utilizadas en el experimento del laberinto, aplicando la red neuronal AHN-ELM propuesta
Elaboración propia.

La Tabla 3 contiene un resumen de los parámetros principales configurados en el Algoritmo Híbrido de la Red Neuronal AHN-ELM, se presenta la información en 3 columnas: la primera corresponde al elemento de la instancia hipocámpica que

debe configurarse, la segunda es el nombre de la variable en el algoritmo y, la tercera, los valores específicos utilizados.

Tabla 3 Parámetros de configuración AHN-ELM para experimento virtual del laberinto.

Elemento	Variable	Consideración Laberinto
Límite inferior laberinto de DG_a	$T_c - LSL_{DG}$	NA
Límite superior laberinto de DG_a	$T_c - USL_{DG}$	NA
Número de neuronas ocultas DG_b	n_{DGb}	350
Número de neuronas ocultas $CA3$	n_{CA3}	100
Número de neuronas ocultas $CA1$	n_{CA1}	120
Número de neuronas salida DG_b	n_{DGb}	100
Número de neuronas salida $CA3$	n_{CA3}	50
Número de neuronas salida $CA1$	n_{CA1}	120
Datos Entrenamiento inicial $CA1$	$n_{CA1Entren}$	50
Error permitido DG	ϵ_{DG}	0.01
Error permitido $CA3$	ϵ_{CA3}	0.01
Error permitido $CA1$	ϵ_{CA1}	0.01
Error permitido SUB	ϵ_{SUB}	0.01
Factor exploración gamma γ_{SUB}	α_{SUB}	0.3
Salidas de control $SUBN_{outputs}$	γ_{SUB}	3

Elaboración propia.

3.3.6. Arquitectura para AHN-ELM para prueba de laberinto

La arquitectura de la red AHN-ELM se muestra en la Figura 1 en la cual se pueden observar las cuatro instancias hipocámpicas que componen a la red. La entrada es la red de filtrado dinámico PAT la cual se encarga de identificar el tipo de laberinto, esta sección se denomina DG_a . De la sección PAT pasa el dataset filtrado para que se realice la separación de patrones, que consisten en expender los datos de entrada usando el algoritmo codificador AHN-ELM, esta sección de la red se denomina DG_b . Una característica de esta red de *autoencoder* es que las salidas y las entradas reciben los mismos datos, por lo tanto, la salida real de esta sección de la red son los pesos de la red oculta. Un requisito es que esta sección de la red es que se configura en el modo *Sparse*, esto es, el número de neuronas ocultas es

RED HÍBRIDA AHN-ELM APLICADA AL CONTROL DE SOLUCION DE LABERINTO VIRTUAL

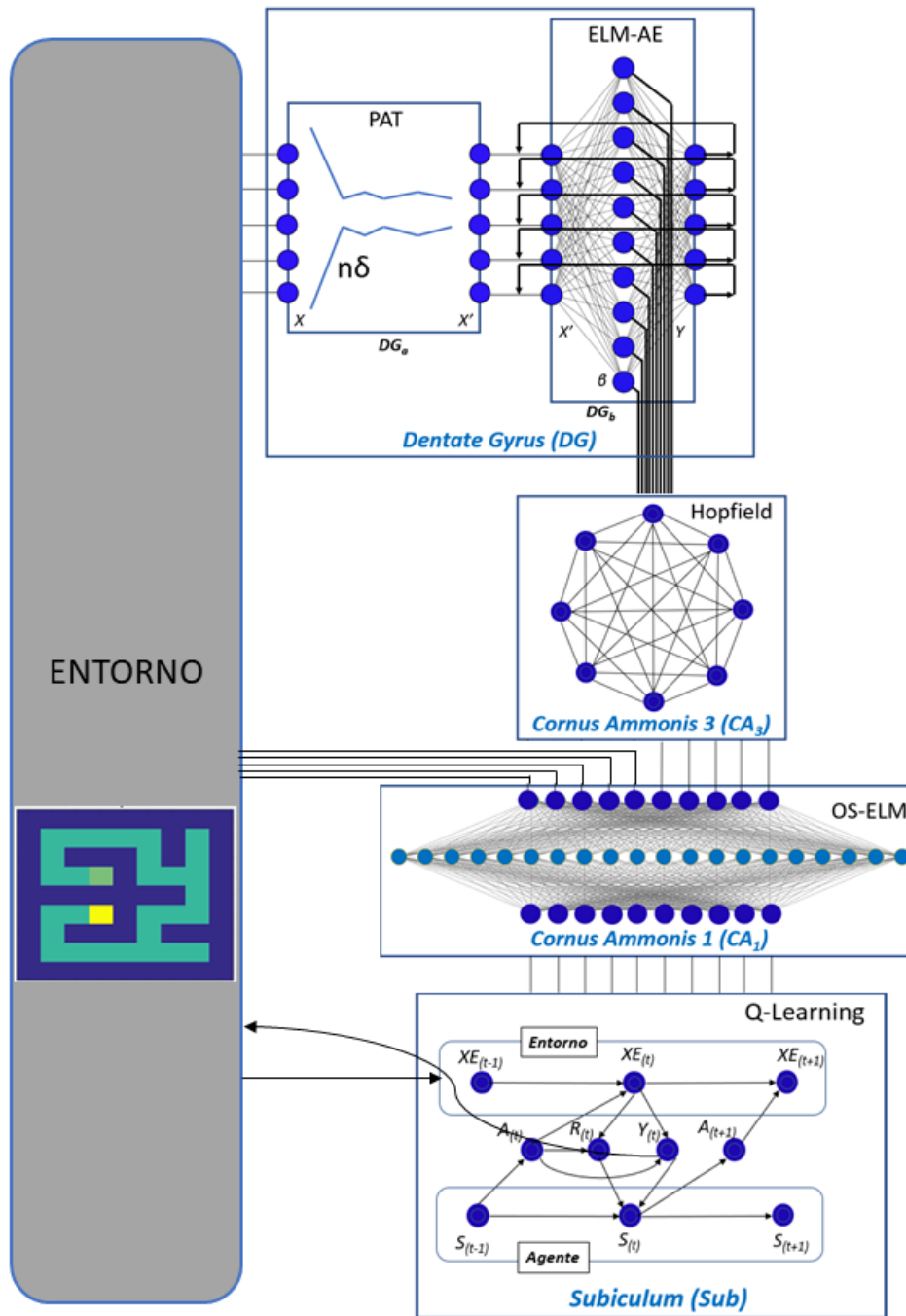


Figura 35. Arquitectura de la red híbrida AHN-ELM para para experimento de laberinto. Elaboración propia.

mayor al número de entradas para lograr la expansión del dataset y facilitar la diferenciación de patrones.

La siguiente red procesa los datos provenientes de DG_b , es la instancia $CA3$ que se implementa con una red recurrente Hopfield, la cual se utiliza para guardar los recuerdos episódicos del entorno. El objetivo de la red Hopfield es la de guardar un patrón y posteriormente recuperarlo. Una cualidad importante de esta sección de la red es que es capaz de reconocer un patrón, aunque éste contenga una pequeña cantidad de ruido.

posteriormente el patrón episódico recuperado se manda a la instancia hipocámpica $CA1$ que es una red de aprendizaje extremo configurada en modo de aprendizaje secuencial denominada OS-ELM, ahí recibe tanto el patrón de recuerdo episódico como la configuración utilizada en el ambiente de control para el sistema de respuesta.

Finalmente, la salida de la instancia hipocámpica $CA1$ alimenta a la instancia hipocámpica Subiculum SUB que es un algoritmo de aprendizaje por refuerzos, en este caso un algoritmo Q-Learning, la salida de este algoritmo va directamente al elemento de control de movimientos de laberinto.

La Figura 36 muestra la configuración y los parámetros principales utilizados en cada una de las instancias hipocámpicas utilizadas en el experimento del laberinto.

Arquitectura AHN-ELM en Laberinto Virtual

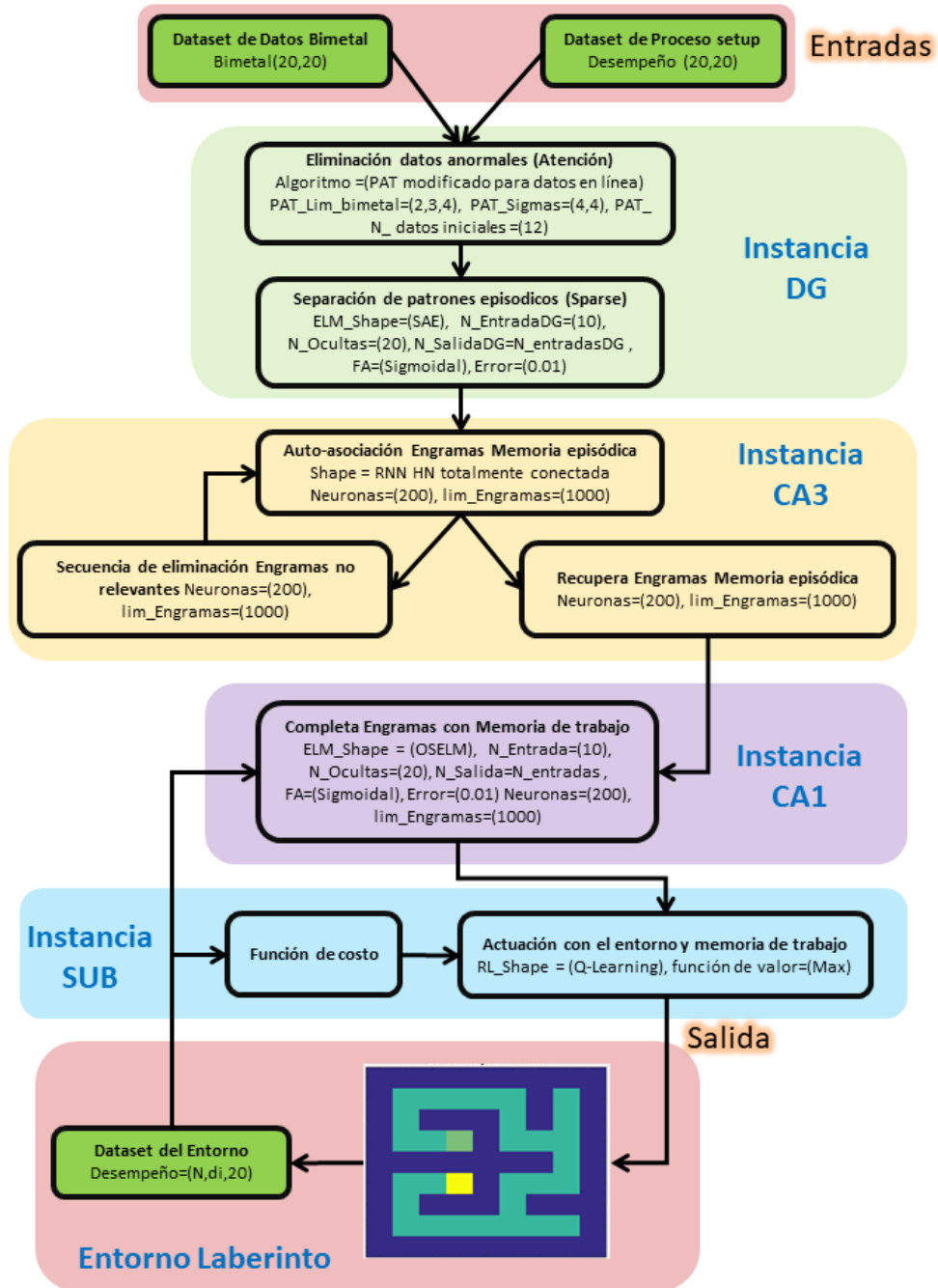


Figura 36. Arquitectura del algoritmo híbrido AHN-ELM para experimento de laberinto. Elaboración propia.

3.3.7. Resultados del experimento

Los resultados del experimento de laberinto se ilustran en la gráfica de la Figura 37, en ella se muestra el número de intentos que el Algoritmo Híbrido AHN-ELM emplea al buscar la solución para cada una de las 5 geometrías del laberinto en evaluación.

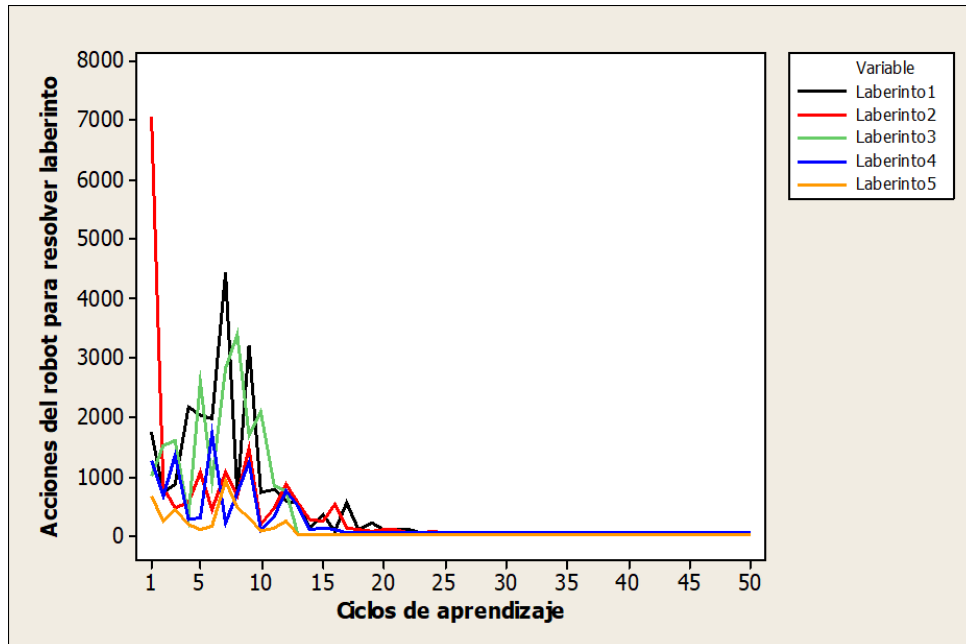


Figura 37. Gráfica de desempeño en las 5 diferentes geometrías del laberinto resuelto por el algoritmo híbrido AHN-ELM.
Elaboración propia.

Se puede observar en la gráfica que al inicio pueden llegar a requerirse hasta 7,000 acciones o movimientos para encontrar la solución, pero al final, cuando aprende el algoritmo, se logra reducir el número de movimientos a la solución óptima, que depende en cada caso de la geometría del laberinto. En el eje X se representa cada intento en el que el cursor regresa a la posición inicial y se mueve hasta alcanzar el objetivo; el eje Y representa el número de movimientos totales utilizados por el Algoritmo para alcanzar la posición Meta.

Puede notarse que, el número óptimo de pasos o acciones para alcanzar la meta desde el punto inicial varía en un rango entre 19 y 43 acciones.

Podemos observar también que, durante los primeros intentos, el desempeño de aprendizaje no es óptimo, ya que el robot (puntero amarillo en los laberintos) realizó, en el peor de los casos, más de 7,000 acciones para alcanzar el objetivo.

También se puede observar que, a medida que se van realizando los intentos de solución, la Red Neuronal va mejorando su desempeño hasta encontrar la solución óptima.

La información colectada durante la solución de los laberintos muestra que fueron resueltos usando entre 14 y 23 ciclos de aprendizaje y que el número de acciones totales fluctuó entre 22,231, para el caso del laberinto 1, que resultó ser el más complejo, a 3,911 en el laberinto 5, que resultó el más sencillo de resolver para el algoritmo.

Para cada uno de los diferentes laberintos, en los resultados de los recorridos iniciales, se observa una fluctuación en el número de acciones que requiere tomar para llegar desde el punto de inicio hasta la meta. Aunque las configuraciones de cada laberinto eran distintas, después de los 23 ciclos de entrenamiento, se alcanza un resultado óptimo. Esto ocurre gracias a la instancia hipocámpica *SUB*, que va almacenando las soluciones parciales en la memoria de trabajo. Cada vez que se cambia la geometría del laberinto, la instancia hipocámpica *CA3* lo identifica, proporcionando la memoria fundamental o recuerdo episódico de este, mientras que, la instancia hipocámpica *CA1*, proporciona la mejor alternativa de solución.

3.3.8. Conclusiones de experimento virtual

Los resultados del experimento virtual de resolución del laberinto nos muestran que el Algoritmo Híbrido AHN-ELM propuesto funciona adecuadamente y conforme a lo esperado. El Algoritmo se aplicó para resolver 5 geometrías distintas del laberinto y logró resolverlos en su totalidad. En un inicio, previo al entrenamiento de la Red Neuronal, fueron empleadas 22,321 acciones para solucionar el laberinto más complejo y 3,911 acciones para resolver el laberinto más simple. Después de entrenada la Red Neuronal Hipocámpica AHN-ELM resolvió los laberintos utilizando únicamente de 19 a 53 acciones. Ahora bien, los ciclos de aprendizaje o

entrenamiento necesarios para optimizar el proceso fluctuaron entre los 14 y 23 ciclos.

Como se puede observar, la Red Neuronal presenta un rendimiento satisfactorio, al encontrar soluciones óptimas después de pocos ciclos de aprendizaje. En dicho contexto virtual, el Algoritmo Híbrido AHN-ELM probó ser capaz de encontrar la estrategia más adecuada para la resolución del problema planteado. A continuación, se procede a presentar la segunda experimentación, que se centra en mostrar la conexión con el mundo real.

3.4 ROBOT SEGUIDOR DE LÍNEA USANDO AHN-ELM

El objetivo de probar el Algoritmo Híbrido AHN-ELM en un robot explorador, es probar su eficacia para obtener datos de un entorno real para procesarlos y así verificar su factibilidad en aplicaciones de control. Se trata de un paso fundamental, aunque ya se ha probado el funcionamiento del Algoritmo en espacios virtuales, el planteamiento original es que sea capaz de aplicarse a procesos de manufactura reales, en el que recibirá información del estado de los procesos que se van realizando y tendrá que ajustar sus estrategias de control para optimizar sus acciones y lograr el objetivo meta. El experimento que se presenta a continuación sirve como puente entre el espacio virtual y la información del mundo real.

Para el experimento se seleccionó la plataforma *VEX Robotics*, combinada con impresiones 3D, para construir un prototipo de robot explorador lo suficientemente robusto para soportar servo motores y baterías, así como varios sistemas embebidos. Los motores son bidireccionales, por lo tanto, logran desplazamientos hacia delante o hacia atrás en cada lado del robot, permitiéndole una movilidad de 360°. La Figura 38 muestra al robot construido para la experimentación el cual fue desarrollado en la plataforma *VEX Robotics*, en combinación con un sistema embebido Arduino Mega.

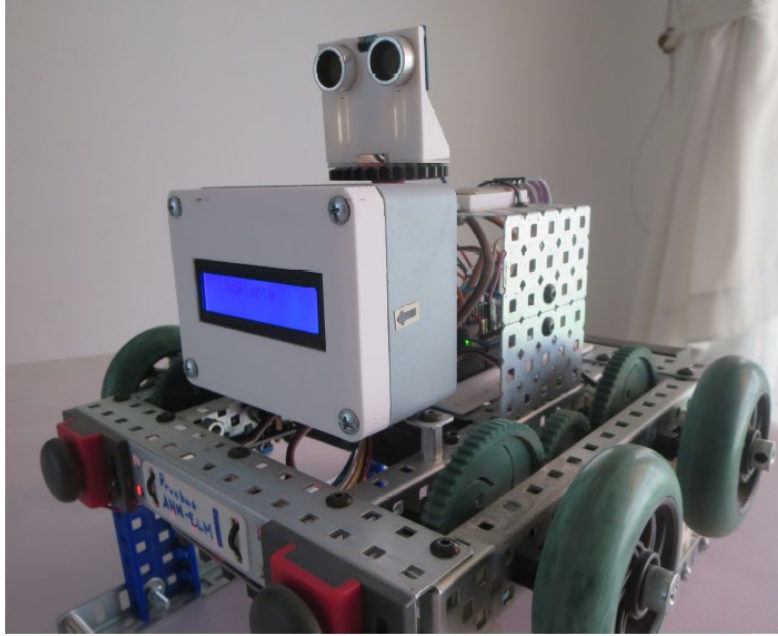


Figura 38. Imagen del robot prototipo explorador utilizado para desarrollar las interfaces de comunicación.
Elaboración propia.

3.4.1 Experimento de robot explorador: procedimiento

El experimento del robot explorador se diseñó para ser probado en un ambiente controlado de laboratorio. Se seleccionó nuevamente la plataforma MatLab para realizar el experimento en lo que se denominó Sistema Cortical, que consiste en una computadora remota en la que se ejecuta la Red Neuronal Hipocámpica AHN-ELM. Adicionalmente, se incorporó un sistema embebido Arduino para controlar al robot explorador. El procedimiento experimental se detalla a continuación:

1. Se selecciona la plataforma robótica para crear al Robot, en este caso se seleccionó el sistema *VEX Robotics*. Se emplea adicionalmente un controlador *Arduino Mega* y una PC compatible con *MatLab*.
2. Se implementa el robot explorador para la experimentación, se instala el controlador *Arduino* y se programa con un compilador de *C++*. Se adicionan accesorios importantes para el experimento, como una interfaz *Bluetooth* y un sensor ultrasónico, entre otros elementos de control.

3. Se desarrolla una interface Bluetooth para conectar la computadora cortical remota al Sistema de Control, que en este caso se localiza en el robot explorador.
4. Se definen los parámetros de la Red Neuronal Hipocámpica AHN-ELM para *DG*, *CA3*, *CA1* y *SUB*. Particularmente, se trabaja con la tasa de aprendizaje *SUB*: α y γ , con el fin de obtener el menor número de ciclos de aprendizaje posible.
5. Se crea un laberinto físico, marcado en el piso con cinta de color oscuro, que señalan las trayectorias que debe seguir el robot explorador.
6. Se registran los resultados y se analizan estadísticamente los ciclos de aprendizaje necesarios en cada recorrido del Robot a través del laberinto, así como su relación con la tasa de aprendizaje *SUB*: α y γ .

3.4.2. Implementación experimento robot explorador con red neuronal AHN-ELM

Para implementar el experimento del Robot explorador, primero se construye la estructura robótica. Las partes principales de la estructura son, en la platina frontal: dos interruptores para detección de golpe o impacto, que evitan que el robot se dañe, así como un sensor seguidor de línea, utilizado para la prueba de laberinto. En la parte posterior, un soporte para el juego de baterías y, sobre la impresión 3D, una caja de platina metálica que se instala como soporte de un servo motor que controlará un sensor ultrasónico para detectar obstáculos. A los lados de la platina hay dos servomotores para el control motriz.

3.4.3. Sistema cortical e interfaz con MatLab

Se instala en la computadora remota un controlador Arduino Uno para implementar el sistema cortical, que se enlaza mediante Bluetooth con el controlador del Sistema de Control ubicado en el Robot explorador y le indica que movimientos deben realizar los servomotores.

La función del controlador del sistema cortical es recibir la información del robot explorador y enviarla en un formato que puede interpretar la Red Neuronal Hipocámpica AHN-ELM, la cual se convertirá en comandos en el controlador del Robot. La Figura 39 muestra el esquema del Sistema de Control del robot explorador y la interfaz con el sistema cortical MatLab. El bloque de la izquierda muestra las secciones del robot que corresponden a Sistema de Control. El bloque de la derecha muestra la interfaz al algoritmo de Red Neuronal Hipocámpica AHN-ELM y es el sistema cortical, que se compone de una computadora remota que contiene a la Red Neuronal AHN-ELM en una plataforma de MatLab. La comunicación entre ambos sistemas se realiza mediante el protocolo Bluetooth.

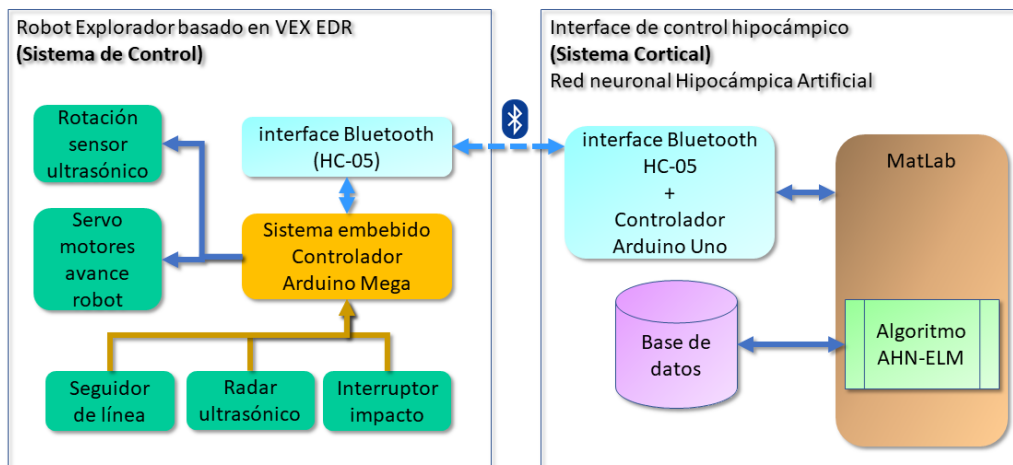


Figura 39. Esquema del sistema de control del robot explorador y la interfaz con el sistema cortical.
Elaboración propia.

3.4.4. Desarrollo de sistema de control y comunicación vía bluetooth

Para el desarrollo del Sistema de Control en el Robot explorador se utiliza un controlador embebido Arduino Mega que se encarga de administrar el movimiento de los servomotores de avance e interpretar las señales de entrada del radar ultrasónico y el sistema seguidor de línea.

Dos servos motores VEX se encargan de dar movilidad al robot, y un tercero se utiliza para direccionar al sensor ultrasónico que se emplea a manera de radar.

Las señales de entrada para el Algoritmo se componen de la información del sensor que percibe la presencia de la línea, las señales del sensor ultrasónico, así como de los interruptores de impacto. La información es enviada a través de la interfaz de Bluetooth HC-05, implementado en el Sistema de Control hacia el sistema cortical.

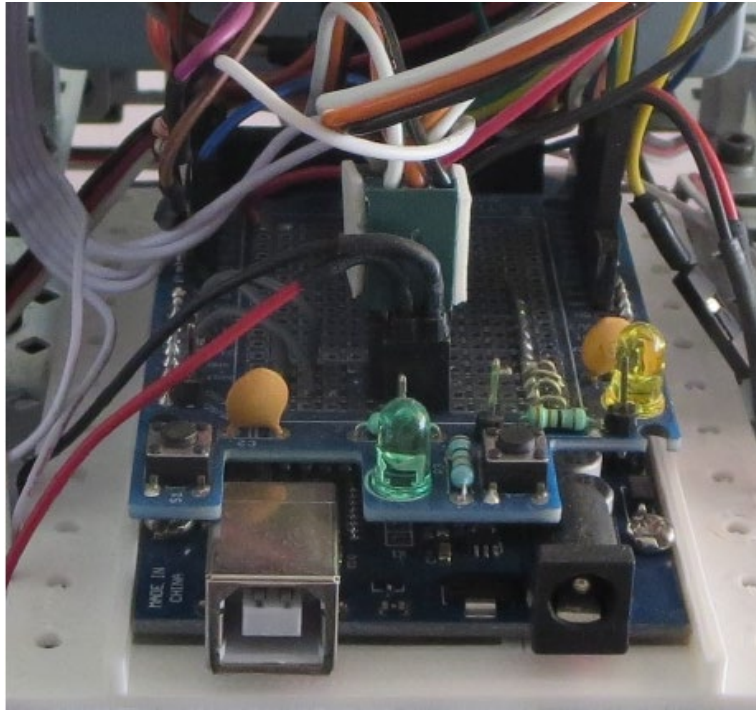


Figura 40. En la imagen se puede observar la implementación del controlador embebido arduino mega como sistema de control.
Elaboración propia.

El controlador Arduino Mega de la Figura 40, tiene como una de sus funciones principales mantener al robot explorador sobre una línea marcada en el piso. Que constituye, en el experimento, información obtenida del exterior. El controlador tiene programadas rutinas que le permiten hacer giros a derecha o izquierda, avanzar, retroceder o detenerse en caso de impacto; todo esto lo realiza de manera autónoma; sin embargo, las decisiones de todo el arreglo vienen ordenadas por el sistema cortical.

Otra de las funciones del Sistema de Control es la de comandar al servomotor que mueve al sensor ultrasónico, dicho servomotor se desplaza de derecha a izquierda y viceversa, a semejanza de un sistema de radar.

Por último, el controlador Arduino Mega tiene una rutina para mantener al robot siguiendo la línea. Si por alguna razón la línea se pierde, el robot hace una pequeña búsqueda para asegurar que sigue sobre ella.

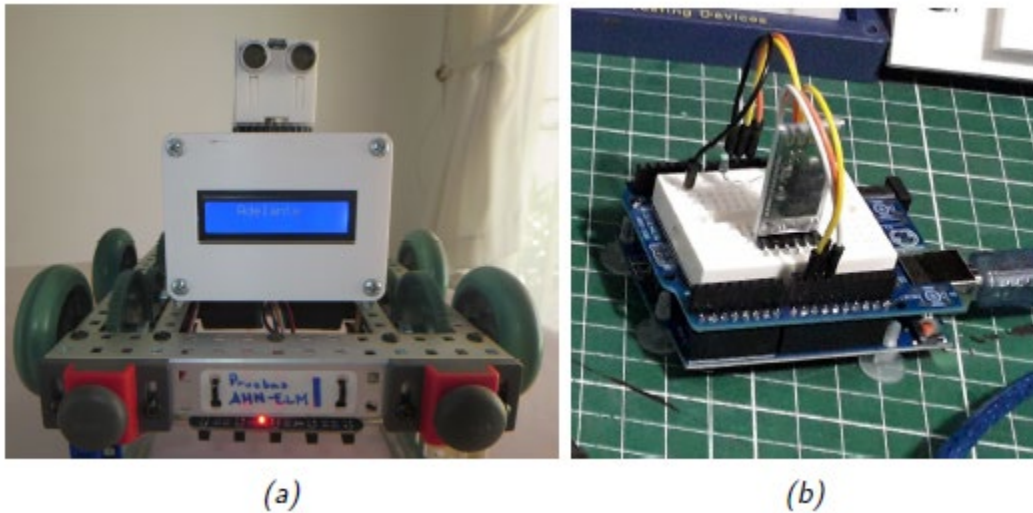


Figura 41 La sub figura (a) muestra al robot explorador y la sub figura (b) muestra la interface bluetooth utilizado en el sistema cortical.
Elaboración propia.

La Figura 41(a) muestra, una vista frontal del robot explorador, se puede observar en su parte inferior el sensor seguidor de línea (donde se encuentra un led rojo encendido). A los lados del led se encuentran los interruptores de impacto. Puede observarse, en su parte superior en color blanco, el sensor ultrasónico giratorio con el que se determina la distancia del objetivo; puede verse también su pantalla de información.

El sensor seguidor detecta cuando el robot no se encuentra sobre la línea oscura y permite al controlado tomar decisiones sobre qué movimiento realizar.

La 41(b) muestra el interface Bluetooth utilizado en el sistema cortical basado en el controlador Arduino Uno y el módulo de Bluetooth HC-05. A continuación, se describe a profundidad el rol de los sensores descritos aquí para lograr la

comunicación entre los datos que recibe el robot de su entorno y el procesamiento de los mismos, llevado a cabo por el Algoritmo Híbrido AHN-ELM.

3.4.5. Procesamiento de datos e interfaz con MatLab

Para procesar los datos con el Algoritmo Híbrido AHN-ELM, se construyó una interface utilizando Bluetooth para recibir la información proveniente del controlador de comunicación montado en el robot explorador.

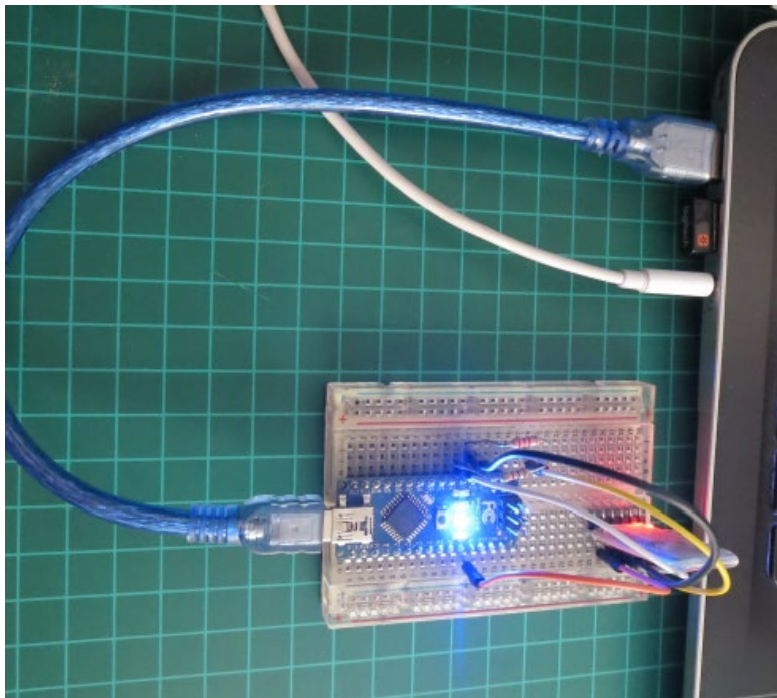


Figura 42. Interfaz de módulo bluetooth HC-05 para comunicación.
Elaboración propia.

Esta configuración le otorga al robot total autonomía para desplazarse sobre el laberinto sin tener que estar conectado y permite la transferencia de información de manera inalámbrica. En el presente experimento se emplea la interface basada en Bluetooth HC-05, pero no implica que la aplicación del Algoritmo esté limitada a este medio de comunicación, puesto que, con pequeños arreglos, es posible implementar la comunicación con Wifi, abriendo la posibilidad de que este sistema pueda utilizarse como un sistema IoT.

La Figura 42 muestra al controlador embebido conectado a la computadora remota, en la que se está ejecutando el programa de MatLab con el Algoritmo Híbrido AHN-ELM.

3.4.6. Arquitectura para AHN-ELM para robot explorador

La arquitectura de la red AHN-ELM se muestra en la Figura 43 en la cual se pueden observar las cuatro instancias hipocámpicas que componen a la red, utilizadas en el experimento del robot explorador. La entrada es la red de filtrado dinámico PAT la cual se encarga de identificar los caminos o posibles rutas a seguir de la ruta a seguir, esta sección se denomina DG_a .

De la sección PAT pasa el dataset filtrado para que se realice la separación de patrones, que consisten en expandir los datos de entrada usando el algoritmo codificador AHN-ELM, esta sección de la red se denomina DG_b . Una característica de esta red de autoencoder es que las salidas y las entradas reciben los mismos datos, por lo tanto, la salida real de esta sección de la red son los pesos de la red oculta. Un requisito es que esta sección de la red es que se configura en el modo Sparse, esto es, el número de neuronas ocultas es mayor al número de entradas para lograr la expansión del dataset y facilitar la diferenciación de patrones.

La siguiente red procesa los datos provenientes de DG_b , es la instancia CA_3 que se implementa con una red recurrente Hopfield, la cual se utiliza para guardar los recuerdos episódicos del entorno. El objetivo de la red Hopfield es la de guardar un patrón y posteriormente recuperarlo. Una cualidad importante de esta sección de la red es que es capaz de reconocer un patrón, aunque éste contenga una pequeña cantidad de ruido, posteriormente el patrón episódico recuperado se manda a la instancia hipocámpica CA_1 que es una red de aprendizaje extremo configurada en modo de aprendizaje secuencial denominada OS-ELM, ahí recibe tanto el patrón de recuerdo episódico como la configuración utilizada en el ambiente de control para el sistema de respuesta.

RED HÍBRIDA AHN-ELM APLICADA A CONTROL DE ROBOT EXPLORADOR

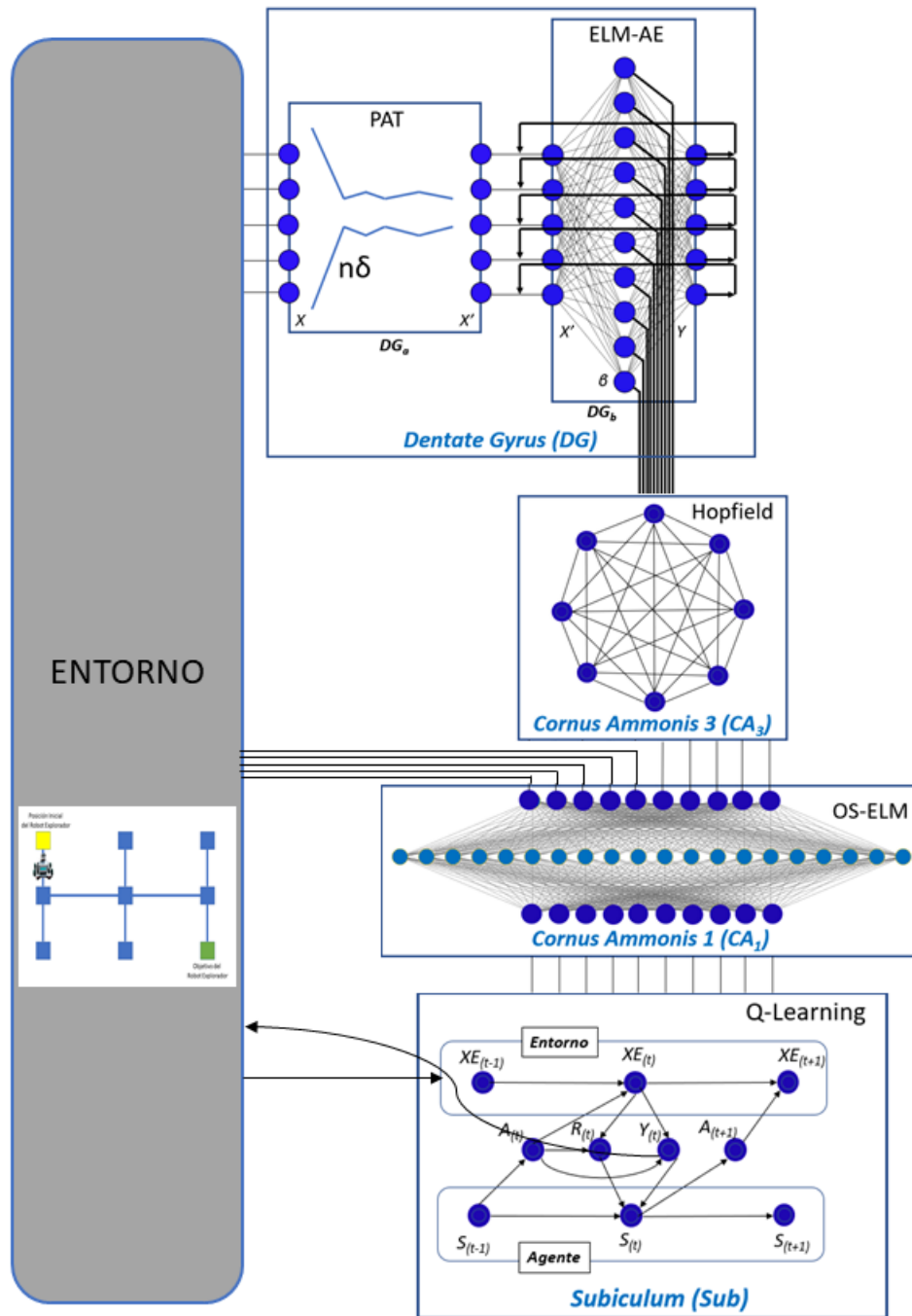


Figura 43. Arquitectura de la red Híbrida AHN-ELM para experimento del robot explorador. Elaboración propia.

Arquitectura AHN-ELM en Robot explorador

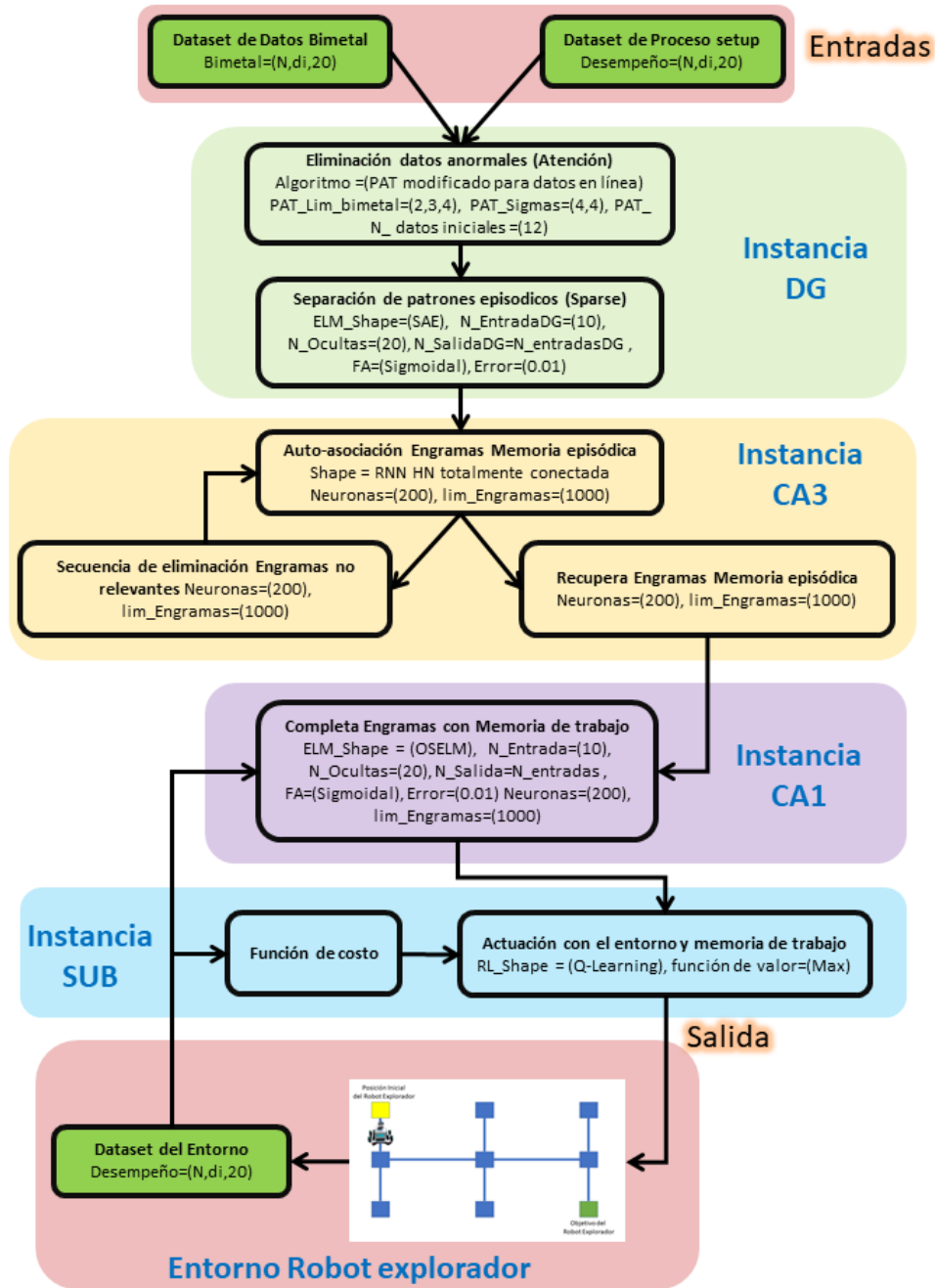


Figura 44. Arquitectura del algoritmo híbrido AHN-ELM para experimento del robot explorador. Elaboración propia.

Finalmente, la salida de la instancia hipocámpica *CA1* alimenta a la instancia hipocámpica Subiculum *SUB* que es un algoritmo de aprendizaje por refuerzos, en este caso un algoritmo Q-Learning, la salida de este algoritmo va directamente al algoritmo de control del Arduino que se encarga de controlar los movimientos del robot, esto para reaccionar de acuerdo con los resultados de la red.

La Figura 44 muestra la configuración y los parámetros principales utilizados en cada una de las instancias hipocámpicas utilizadas en el experimento del control de ruta del robot explorador.

3.4.7. Experimento con el robot explorador en laberinto marcado en piso

El experimento del robot explorador se desarrolla sobre un laberinto marcado en el piso, la Figura 45 muestra las trayectorias escogidas para este experimento. Se define un inicio de ruta mostrado en el recuadro amarillo y un objetivo o meta, indicado en el recuadro verde. Las trayectorias azules son las posibles rutas que tomará el robot explorador. El objetivo de esta prueba es encontrar la ruta más corta entre la posición inicial y el objetivo.

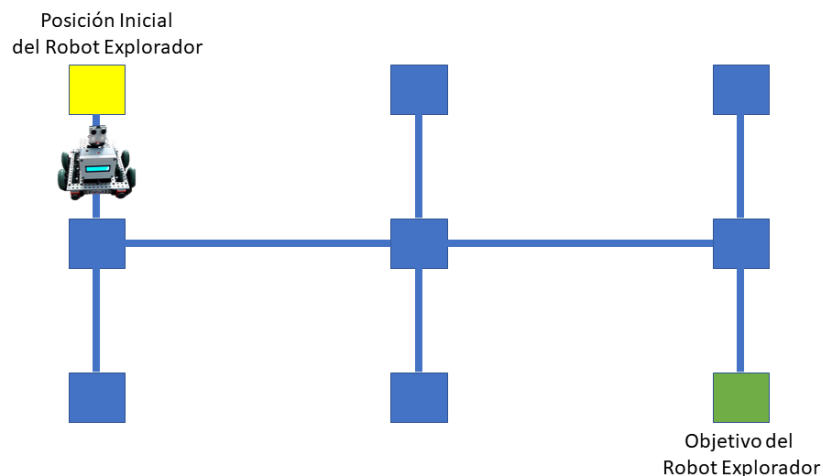


Figura 45. Esquema de ruta a seguir por robot explorador entre el punto de origen y el objetivo.
Elaboración propia.

El Robot cuenta con un sistema seguidor de marcaje en el piso y se programa de forma que se mantenga sobre la línea mientras realiza la exploración hacia el objetivo, marcado con un letrero que está a la misma altura que el sensor ultrasónico, que le indica la distancia que existe entre su posición actual y el objetivo.

Al iniciar el experimento, el robot explorador inicia su búsqueda y, como no ha sido entrenado, realiza una gran cantidad de movimientos adelante-atrás, vuelta izquierda-derecha o giro de 180 grados, según las opciones de ruta que el sistema cortical le envía.

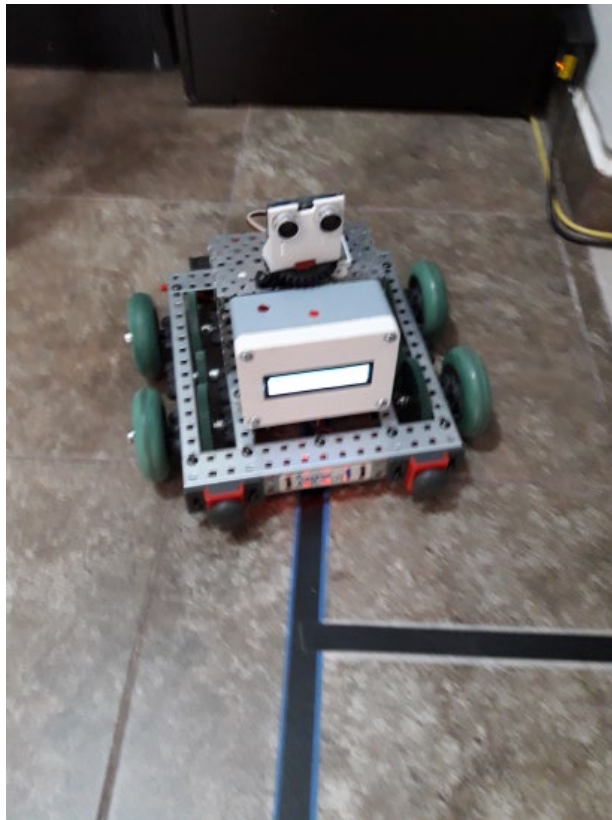


Figura 46. Robot explorador siguiendo la línea negra del laberinto durante el experimento. Elaboración propia.

El Robot avanza muy lentamente en el primer ciclo de aprendizaje, ya que no tiene ninguna referencia sobre cómo llegar al objetivo. El método de exploración-explotación le ayuda al algoritmo a buscar nuevas opciones cuando no logra llegar al objetivo con las secuencias de acciones actuales. Sin embargo, muchas

de ellas son azarosas, por lo tanto, se sigue con este desempeño hasta alcanzar por primera vez al objetivo.

Una vez que alcanza el objetivo, el robot explorador se mueve a la posición inicial y se repite el proceso. Cada vez que inicia el laberinto, a partir de ese momento, se usa el aprendizaje de los recorridos anteriores para optimizar su nuevo recorrido.

La Figura 46 muestra al robot durante el experimento. El robot explorador VEX fue resolviendo un laberinto lineal, marcado por una cinta negra en el piso, buscando la ruta óptima para alcanzar el objetivo.

3.4.8. Resultados del experimento de laboratorio con robot explorador usando red neuronal AHN-ELM

Los resultados del experimento empleando un robot explorador implementado la Red Neuronal AHN-ELM, se muestran en la Figura 47 mediante una gráfica de tiempo.

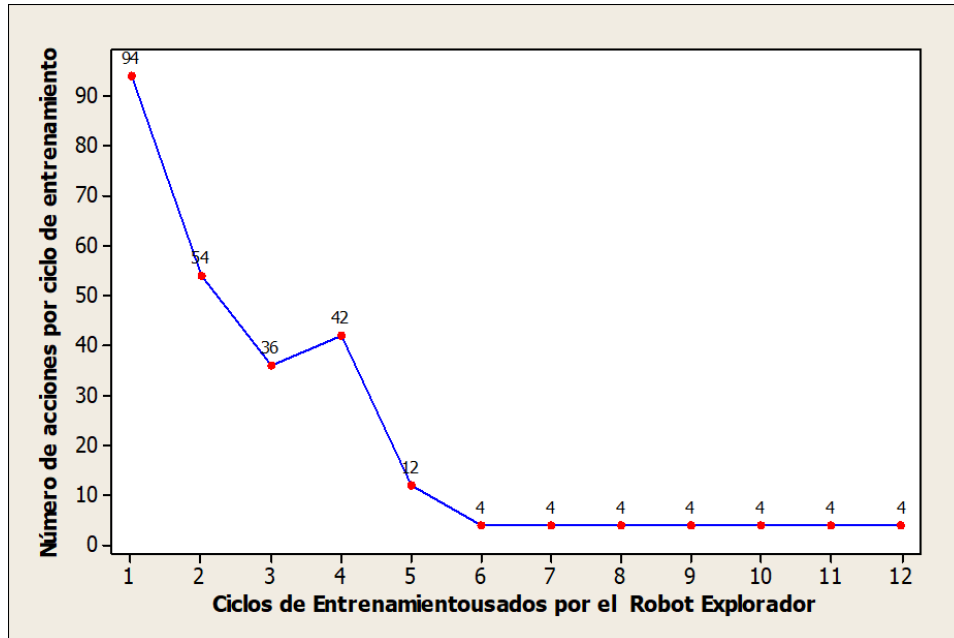


Figura 47. Gráfica de resultados del experimento con el robot explorador. Elaboración propia.

La gráfica muestra, en el eje X, el número de ciclos de entrenamiento que realizó el robot explorador antes de aprender la ruta más corta hasta el punto objetivo. En el eje Y se representa el número de acciones que toma el robot, es decir, las diferentes direcciones que toma: adelante, atrás, izquierda, derecha o giro sobre sí mismo, éste último movimiento se emplea cuando llega a uno de los extremos del laberinto, en el que la ruta se interrumpe. El recorrido entre cada nodo, representado por un recuadro en el laberinto, corresponde a una acción. En la gráfica, puede observarse que en el primer intento el robot realizó 94 acciones para alcanzar el punto objetivo. En el segundo intento, utilizó 54 acciones y, en el sexto intento, logró completar la ruta en sólo 4 acciones, este es el número mínimo requerido para llegar al objetivo dada la geometría del laberinto. El Robot requirió ejecutar 238 acciones para lograr el aprendizaje de la ruta propuesta en el experimento.

3.4.9. Conclusiones de prueba de laboratorio

La conclusión más importante que se obtiene de la presente sección 3.4 es que se logró demostrar que el Algoritmo Híbrido AHN-ELM funciona como Sistema de Control Cognitivo. Se demuestra que es capaz de comunicarse con aplicaciones de robots exploradores basados en controladores embebidos, así como de resolver problemas en ambientes virtuales.

Otra conclusión importante es que es factible emplear una computadora remota para ejecutar la Red AHN-ELM. En el caso de los experimentos de laboratorio, el robot explorador recibe las instrucciones del Algoritmo Híbrido AHN-ELM y ejecuta las acciones seleccionadas en el mundo real.

Los experimentos virtuales o de laboratorio pueden ser, por sí mismos, una rama de experimentación con un alto consumo de recursos. Los experimentos seleccionados permitieron obtener las bases para los interfaces de conexión y para realizar la depuración del Algoritmo. Los principios generados en esta sección sirven como un primer paso para el desarrollo de interfaces más robustas y con aplicaciones más amplias.

4. RESULTADOS

4.1 APLICACIÓN DE ALGORITMO HÍBRIDO PAT-ELM A LA CARACTERIZACIÓN DE BIMETALES

4.1.1 Introducción

La presente sección expone el primer experimento que utiliza la Red Neuronal PAT-ELM, denominado Algoritmo Híbrido PAT-ELM. Esta sección se sitúa en una aplicación real, expuesta a variaciones de materia prima, de ajustes y de procesos.

La aplicación se hace al proceso de manufactura de bimetales termostáticos, necesarios en la creación de moto-protectores eléctricos, que deben cumplir con estándares muy estrictos y delimitados para funcionar correctamente.

El Algoritmo Híbrido PAT-ELM se aplica para optimizar el proceso de producción de dichos bimetales, calculando su curvatura adecuada y asegurando que se cumplen los criterios industriales necesarios, que se traducen en la norma de caracterización ASTM-B106-08.

Los primeros apartados presentan la descripción del proceso de manufactura de los bimetales termostáticos y los criterios que se deben cumplir en él. Enseguida, se explica el Algoritmo Híbrido PAT-ELM y su aplicación para realizar monitoreo cognitivo. Por último, se presentan los indicadores de desempeño del Algoritmo Híbrido PAT-ELM aplicado. Este primer experimento es muy importante para adquirir experiencia sobre el comportamiento del Nuevo Algoritmo en el ámbito de la industria.

4.1.2 Caracterización de bimetales termostáticos para moto-protectores eléctricos

En la fabricación de protectores de motores eléctricos, es de vital importancia conocer el perfil de fuerza de los bimetales termostáticos; este se obtiene mediante una caracterización, basada en la norma ASTM B106-08. Los moto-protectores que

se fabrican a base de material bimetálico termostático se emplean comúnmente para evitar daños debidos al sobrecalentamiento o cortocircuito en motores eléctricos. El bimetálico termostático es un material compuesto, formado por dos o más capas de metal que tienen diferente coeficiente de expansión térmica. Estas capas hacen que el material cambie su curvatura en respuesta a las variaciones de temperatura, producidas por el calentamiento o por una corriente eléctrica excesiva.

La geometría ideal de un bimetálico termostático al ser expuesto a una temperatura definida T_c , debe ser tal, que genere una deflexión en la curvatura de la lámina bimetálica para que produzca la desconexión casi instantánea en el circuito eléctrico, para proteger al motor de condiciones de funcionamiento anormales.

Tradicionalmente, el proceso de ajuste inicial del formado de la curvatura del bimetálico termostático, se realiza midiendo el tiempo de apertura y cierre al ser expuesto a las temperaturas de prueba T_f y T_c , inmediatamente después del curvado de la lámina termostática, seguido de la medición de la distancia de desplazamiento generada por el cambio de temperatura y la fuerza generada durante la transición. Este proceso de ajuste es muy importante para la fabricación de moto-protectores funcionales que sean duraderos y precisos.

La medición en milisegundos de los tiempos de actuación a las temperaturas T_f y T_c , se realiza usando un micrófono muy sensible, que detecta el momento de transición (cambio de curvatura del bimetálico de la posición inicial a la posición de disparo a la temperatura T_c cuando se aplica un aire caliente sobre el material formado.

Para la medición en este primer experimento, se modifica el Algoritmo Híbrido AHN-ELM a una versión simplificada: PAT-ELM, la cual resulta apta para el monitoreo cognitivo. El Algoritmo Híbrido PAT-ELM estima el perfil de fuerza que produce una tira bimetálica termostática curvada al ser expuesta a las temperaturas T_f y T_c .

4.1.3 Acotando el experimento

En este primer experimento práctico se valida el Algoritmo Híbrido PAT-ELM, para minimizar los riesgos de afectar un proceso de manufactura real, por ello se busca reducir el número de variables en el proceso. Con dicha finalidad se escoge un proceso simple que se limita a la caracterización. Para este caso, se aplica el Algoritmo Híbrido PAT-ELM. Con esto se prueba el Algoritmo de monitoreo cognitivo, en un proceso de manufactura en el que se caracterizan los bimetales termostáticos empleados para la fabricación de moto-protectores eléctricos.

4.1.4. Fundamentos del moto-protector térmico a base de bimetal termostático

Los motores de inducción eléctrica son comunes en los escenarios industriales y domésticos, por ejemplo, en bombas, elevadores, grúas, compresores, ventiladores, trituradoras, molinos, transportadores, entre otros. Tienen una tasa de falla estimada del 3 al 5%, que en situaciones extremas alcanza hasta el 12% de fallas. Las estadísticas muestran que el sobrecalentamiento es uno de los principales responsables de tales fallas. Por lo tanto, el moto-protector juega un papel clave para mejorar este escenario (Gedzurs, 2015). Por ejemplo, los motores de voltaje medio multifase modernos, a pesar de ser muy robustos y fiables, suelen funcionar muy cerca de sus límites térmicos, lo que hace que la protección térmica sea esencial (Kullkarni & Ugale, 2015).

La sobrecarga térmica debida a las variaciones de voltaje o de carga, los bloqueos del rotor o las fases en solitario son los principales responsables de las fallas de un motor eléctrico, causando hasta el 44% de los casos de mal funcionamiento (Protection Schemes for Three Phase Induction Motor, 2017).

Los interruptores industriales estándar, activados por temperatura, se basan en tiras bimetálicas termostáticas que proporcionan la fuerza de disparo requerida para el accionamiento del moto-protector (D'Entremont, 1976).

4.1.5. Estructura del moto-protector

Un moto-protector consta de una placa de material bimetálico y una placa de material conductor mono metal; esta última sirve como contacto estacionario, mientras que el material bimetálico funciona como un contacto móvil (American Society of Heating, 2014). Estas dos placas metálicas usualmente están encapsuladas, de manera que no queden expuestas al exterior y que una de las placas pueda moverse libremente. El material que soporta a ambas placas es usualmente un material no conductor. Todo el conjunto se ensambla de tal forma que la placa bimetálica sea sensible, por conducción o radiación, a la temperatura que está monitoreando.

La Figura 48 presenta una imagen con la forma de un bimetálico termostático típico, incluyendo el ensamble de los contactos eléctricos. Esta es una de las muchas posibles geometrías y arreglos para su uso en moto-protectores.

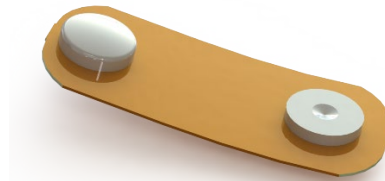


Figura 48. Geometría de un bimetálico termostático típico.
Elaboración propia.

La Figura 49 muestra dos condiciones de curvatura de tira bimetálica termostática. La temperatura T_f es la temperatura ambiente y la temperatura T_c una temperatura elevada, L la longitud de la tira, t el grosor y D la distancia de separación entre ambas temperaturas. Se puede observar que las dos posiciones son dependientes de las temperaturas T_f y T_c , también se muestran los dos metales que componen al bimetálico.

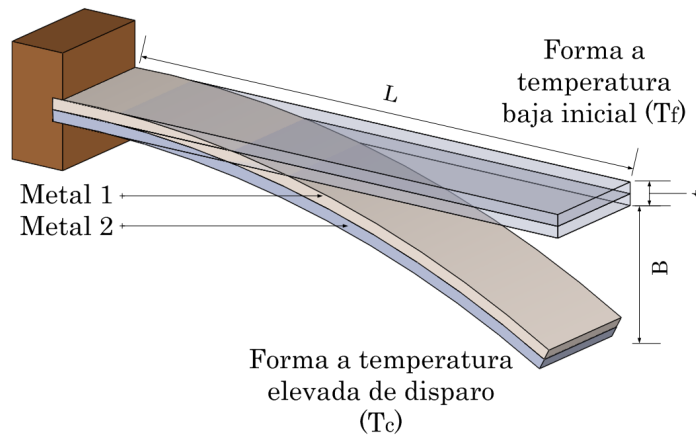


Figura 49. La imagen muestra dos condiciones de curvatura de tira bimetálica termostática al ser expuesta a temperaturas diferentes.
Elaboración propia.

4.1.6. Cálculo de curvatura y fuerza en bimetálico

En un material bimetálico, la curvatura depende de la temperatura (Zang & Feng, 2008), de las características del material y de la fuerza desarrollada por la tira bimetálica. También la desviación es importante para determinar el tiempo necesario para liberar el pestillo de disparo cuando se produce la sobrecarga. En este sentido, el modelo matemático de Timoshenko que se muestra en la Ecuación 33 (Angel & Haritos, 2013), (Zelenka, 2016) y (Khatkhate, Raj, & Mirchandani, 2017) se ha aplicado ampliamente para determinar la curvatura de la tira. El modelo está representado por la siguiente ecuación:

$$\rho = \frac{\left[3 \cdot (1 + m)^2 + (1 + m \cdot n) \left(m^2 + \frac{1}{m \cdot n} \right) \right]}{6 \cdot (\alpha_2 - \alpha_1) \cdot (T_f - T_c) \cdot (1 + m)^2}$$

Ecuación 33. Modelo matemático de Timoshenko

donde: ρ es el radio de curvatura;

$t = t_1 + t_2$ es el grosor total de la tira bimetálica;

t_1, t_2 son los grosores individuales de cada metal que forma el bimetálico;

$m = \frac{t_1}{t_2}$ es la proporción de cada metal;

E_2, E_1 es el módulo lineal de los dos metales;

$n = \frac{E_1}{E_2}$ es la proporción de los módulos de Young's';

α_2 y α_1 son los coeficientes de expansión térmica de los metales; donde α_2 se asume que es numéricamente mayor que α_1 ;

T_c y T_f son las temperaturas caliente y fría iniciales.

La fuerza de la tira bimetálica termostática (Wickeder Group, 2010) se describe con la Ecuación 35 así como Ecuación 36 y la Ecuación 34 representa la deflexión de la tira en forma reducida, que al mismo tiempo genera la fuerza de origen mecánico, producida por la tira bimetálica y representada en la Ecuación 35 describe la fuerza adicional producida de origen térmico de la Ecuación 36.

$$B = \frac{0.53F(T_2 - T_1)L^2}{t}$$

Ecuación 34. Deflexión térmica de la lámina bimetálica termostática.

$$P = \frac{4EBwt^3}{L^3}$$

Ecuación 35. Fuerza mecánica de la lámina bimetálica termostática.

$$P = \frac{2.12EF(T_2 - T_1)wt^2}{L}$$

Ecuación 36. Fuerza térmica de la lámina bimetálica termostática

Donde:

t es el grosor de la tira bimetálica termostática,

w el ancho,

L la longitud,

B la deflexión,

E el módulo de elasticidad,

F la Flexibilidad,

P la fuerza en libras y

$(T_2 - T_1)$ la temperatura de cambio.

4.1.7. Caracterización del bimetálico termostático

Se conoce como caracterización a la metodología que prueba y mide el comportamiento de un material bimetálico termostático, que presenta un comportamiento específico al ser expuesto a cierta temperatura, las propiedades que se miden se muestran en la Tabla 4 (Angel & Haritos, 2013), (Caliskan & Onat, 2017), (Sinclair, 2001) y (Wickeder Group, 2010).

El experimento que se desarrolla aquí, se enfoca en los bimetálicos termostáticos que resultan compatibles para la fabricación de moto-protectores.

Dos de los parámetros de mayor importancia en la operación del protector son su desplazamiento a una cierta temperatura y la fuerza de empuje. La medición del desplazamiento en milímetros es realizada mediante un transformador diferencial de variación lineal (LVDT).

La medición de fuerza en miligramos se realiza usando un sensor piezoeléctrico.

El proceso de caracterización de acuerdo con la norma ASTM B106-08, consiste en sumergir consecutivamente en dos tanques de aceite la placa bimetálica, a las temperaturas T_f , temperatura en la que debe mantenerse sin flexión; y, después a T_c , temperatura a la que debe flexionarse.

El LVDT y el piezoeléctrico, realizan las mediciones simultáneamente al entrar a uno u otro tanque.

Tabla 4. Características de bimetal termostático para aplicaciones en moto-protectores térmicos.

Descripción	Variable	Unidades
Desplazamiento a una cierta temperatura	D	mm
Fuerza de empuje	F	libras fuerza
Radio de curvatura de placa	ρ	mm
Grosor del metal	t, t_1, t_2	mm
Módulo de elasticidad de Young	E, E_1, E_2	kgf/mm^2
Coefficiente de expansión térmica	$\alpha_1, \alpha_2, \dots, \alpha_n$	-
Temperatura de estado frío	T_f	$^{\circ}C$
Temperatura de estado caliente	T_c	$^{\circ}C$
Grosor de la placa termostática	ω	mm
Flexibilidad (Flexibility) de la placa	v	mm
Fuerza producida por la placa a T_c	P	Kgf
Longitud activa de la placa	L	mm
Dureza de la placa	H	mm
Planicidad	L_f	mm
Curvatura transversal	C_c	mm

Elaboración propia.

La medición basada en la norma ASTM-B106-08 es un método de prueba que determina la flexibilidad de los metales termostáticos, en el que la flexibilidad (F) se define como una medida de la tasa de desviación térmica o desviación característica a cierta temperatura (ASTM, 2013), (Middlesex, 2016). La medición de la flexibilidad es una característica importante para esta investigación, debido a que es el origen de la curva fuerza-desplazamiento.

La curvatura del bimetal determina el comportamiento de deformación elástica instantánea (*Snap* en inglés), que va a activar la desconexión de un moto-protector eléctrico.

Dicha curvatura se mide con base en la norma ASTM-B016-08, la cual especifica que el bimetal debe ser sumergido en una tina de aceite a la temperatura de prueba.

Sin embargo, debido al gran número de variables involucradas en la medición y a las variaciones inherentes al proceso de fabricación de los bimetales, el método de medición por inmersión en aceite no sólo es lento, sino que en muchas ocasiones no resulta preciso, lo que genera material de desperdicio durante la fase de ajuste inicial y durante la elaboración del lote de producción.

4.1.8. Caracterización del perfil de fuerza-desplazamiento a T_c

Para generar un perfil de caracterización de fuerza-desplazamiento (Caliskan & Onat, 2017) a una temperatura fija, usualmente es necesario utilizar un arreglo de varios equipos de medición (Kanthal AB, 2008). Para calcular el desplazamiento del material termostático, se utiliza un LVDT, que es un transformador diferencial de voltaje lineal. Para la medición de la fuerza generada, se utiliza una galga extensiométrica. El espécimen de bimetálico a caracterizar se monta en un soporte que permite ambas mediciones simultáneamente y, una vez que el arreglo está completo, se pone dentro de una cámara térmica a una temperatura específica T_c ; también puede utilizarse un contenedor de aceite con agitador (Shivalik, 2012) y (Hitachi Bimetals, 2008). Después de que el espécimen está dentro de la cámara térmica (o en el contenedor), se gráfica tanto la fuerza como el desplazamiento que genera, asegurándose de cubrir el rango de la temperatura y desplazamiento del *snap*. Una vez disparado el bimetálico, se pasa a una nueva cámara térmica a la temperatura de cierre T_f y se continúa graficando los datos, hasta regresar a la posición inicial.

La Figura 50 presenta una gráfica X-Y de una muestra caracterizada usando el método descrito en el párrafo anterior. El eje horizontal muestra la distancia y el eje vertical la fuerza generada al exponerse a la temperatura T_c iniciando en T_f . En la gráfica se puede observar que la fuerza máxima generada por el bimetálico es de 1200 gramos y la máxima deflexión que genera la tira es de 125 mm. El *snap* o desconexión rápida ocurre a los 90mm aproximadamente, cuando se expone a la temperatura T_c , y la distancia de retorno para reconexión a T_f ocurre a los 80mm aproximadamente.

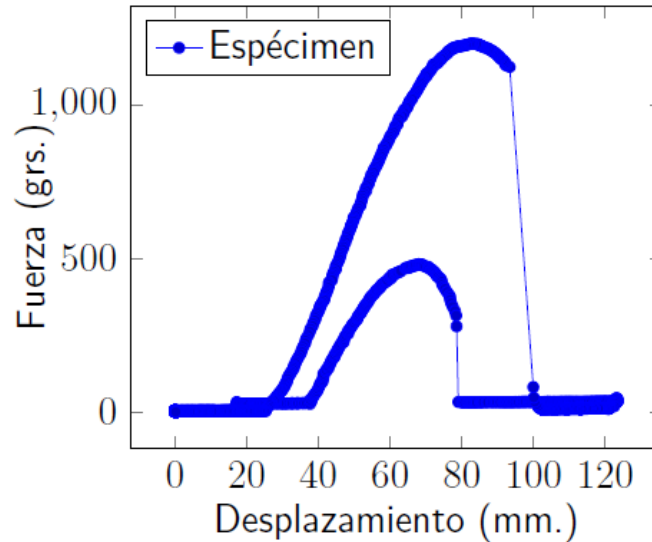


Figura 50. Gráfica X-Y que muestra el perfil fuerza-distancia de un bimetálico termostático. Elaboración propia.

4.1.9. Algoritmo híbrido PAT-ELM para caracterización del bimetal

El propósito del Algoritmo Híbrido Neuronal PAT-ELM es caracterizar el bimetal, para lograrlo se genera un modelo predictivo del comportamiento del perfil de fuerza, usando los datos del desplazamiento dinámico del bimetal termostático; reduciendo así la necesidad de realizar mediciones adicionales para determinar si una placa bimetálica formada cumple o no con los requisitos para usarse apropiadamente en un moto-protector.

El Algoritmo se genera al combinar el filtro dinámico PAT, que elimina los datos anormales y permite al algoritmo ELM aprender sólo la información relevante.

4.1.10 Arquitectura PAT-ELM para caracterización de bimetales

La Figura 51 muestra la configuración de la red PAT-ELM usada en el experimento de Caracterización de la lámina bimetálica termostática. La Figura 52 muestra la arquitectura utilizada.

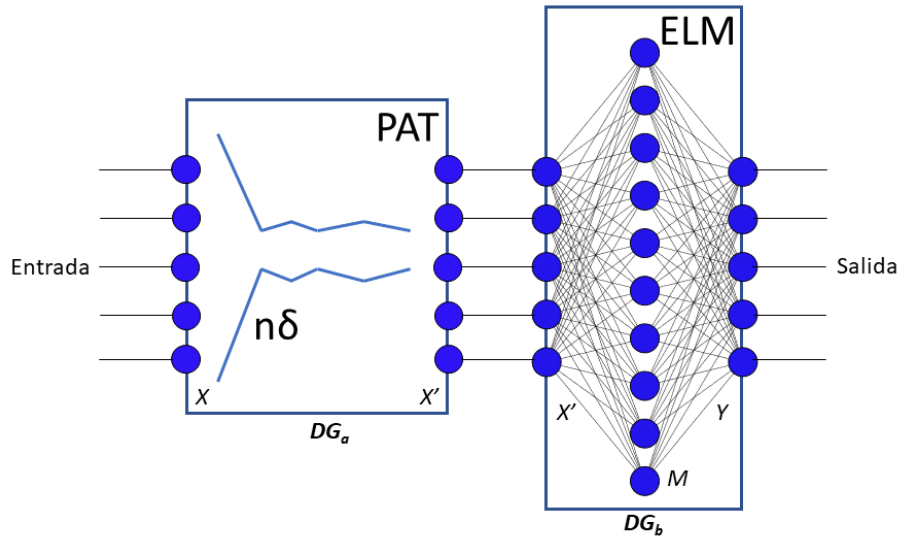


Figura 51. Arquitectura de la red híbrida PAT-ELM para la caracterización de bimetales. Elaboración propia.

El Algoritmo propuesto está usando la RNA PAT-ELM para obtener un caracterizador que logre trabajar con una cantidad mínima de ciclos de entrenamiento. Se compone de dos secciones: la encargada de la eliminación de datos anormales y la RNA o Red Neuronal Artificial ELM, que funciona como modelo de predicción.

La primera sección del Algoritmo elimina los valores atípicos en el conjunto de datos, para minimizar el ruido inducido por el equipo de medición y la variación debida a los cambios en el entorno de fabricación.

La segunda sección es una red neuronal Máquina de Aprendizaje Extremo o ELM, que tiene como entrada el perfil de tiempo-distancia generado por el dispositivo caracterizador prototipo al exponer la lámina bimetálica a un chorro de aire caliente a temperatura T_c . La salida de la red neuronal se entrena para pronosticar los valores de fuerza producidos por la lámina bimetálica en evaluación.

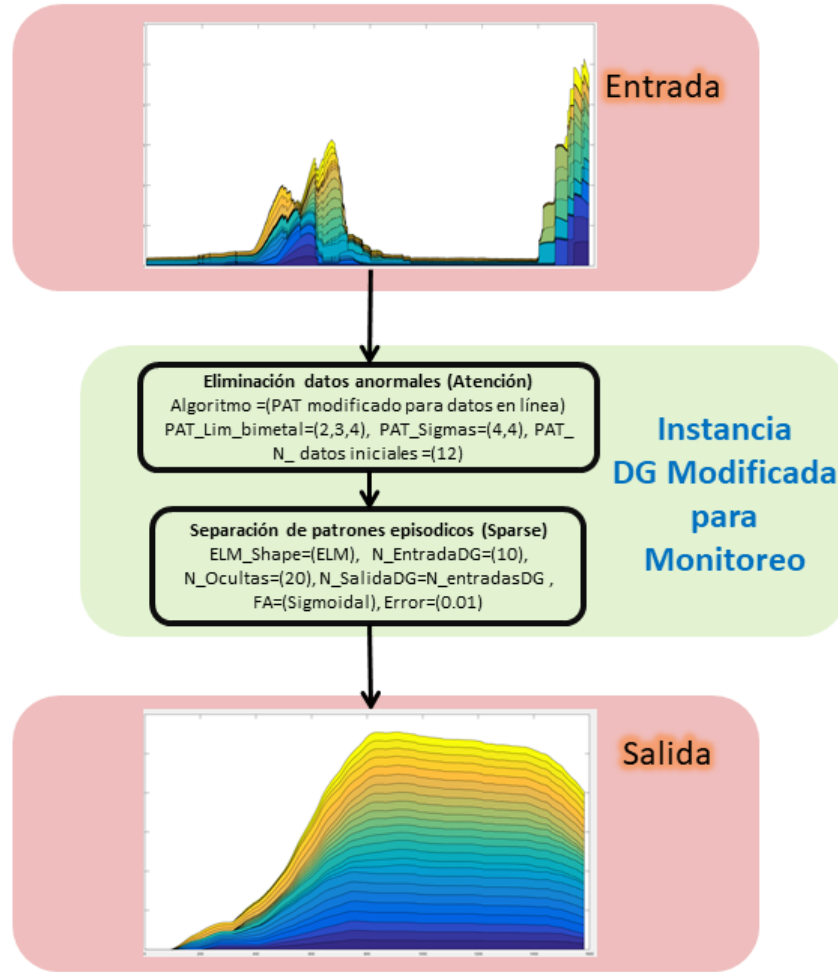


Figura 52. Configuración del algoritmo Híbrido PAT-ELM para la caracterización de bimetales.
Elaboración propia.

Un algoritmo de apoyo que funciona como selector, utiliza el vector de datos de configuración y selecciona la Red PAT-ELM apropiada para el bimetal a analizar. La siguiente fase es la etapa de entrenamiento, que hace uso de una matriz que representa el perfil de desplazamiento dinámico y lo relaciona con la salida en la caracterización del perfil fuerza. Este paso es la fase inicial de entrenamiento del algoritmo, que, una vez concluido, puede usarse para predecir el perfil de fuerza de empuje para el tipo de configuración de bimetal termostático.

El Algoritmo híbrido PAT-ELM usado se muestra en el Algoritmo 9.

Algoritmo 9 PAT-ELM para Monitoreo Cognitivo

Requiere: Patrones de estado incluyendo anomalías x_1, x_2, \dots, x_n , número de patrones iniciales ϕ .

- 1: Se calcula \bar{x} y δ de los ϕ patrones iniciales.
- 2: **repeat**
- 3: **for** Cada paso del episodio **do**
- 4: $\bar{x}_{robusta} = Q_{1,3} - \frac{IQR}{2}$
- 5: $PAT_{limiteinferior} = Q_1 - \frac{(x_i - 0,675)}{1,35}(Q_3 - Q_1)$
- 6: $PAT_{limitesuperior} = Q_3 + \frac{(x_i - 0,675)}{1,35}(Q_3 - Q_1)$
- 7:
$$x_{PAT}(p) = \begin{cases} x_{ec2}(p) < PAT_{limiteinferior} & \rightarrow delete \\ x_{ec2}(p) > PAT_{limitesuperior} & \rightarrow delete \\ \rho < x_{ec2}(p) < \vartheta & x_{PAT}(p) = x_{ec2}(p) \end{cases}$$
- 8: Aplicar a_k , observar el estado siguiente s_{k+1} y la recompensa r_{k+1}
- 9: **end for**
- 10: **until** número de patrones faltantes de evaluar del lote = 0
- 11: **return** π^*

Requiere: Datos $(N.x, N.y)$

- 12: Se inicializa la ELM, número de nodos, número de capas ocultas L , función de activación $g(x)$.
 - 13: Cargar los datos de entrenamiento. $N = \{x_i, t_i \mid x_i \in R^n, t_i \in R^m, i = 1, \dots, N\}$
 - 14: Define el tipo de función del ELM, categorización o regresión.
 - 15: Genera la Matriz y se asignan aleatoriamente los pesos a los vectores de entrada w_i y bias b_i .
 - 16: Calcula la Matriz de la capa oculta de salida H con base en la función de activación.
 - 17: Calcula los pesos de las salidas $\beta : \beta = H^\dagger$ Donde H^\dagger es la inversa generalizada de Moore-Penrose de H .
 - 18: -
-

Elaboración propia.

4.1.11. Dispositivo medición gradiente de desplazamiento a T_c

El dispositivo utilizado para la medición del gradiente de desplazamiento a temperatura fija se muestra en la Figura 53. El dispositivo se basa en una celda de fuerza, o sensor extensiométrico, que mide la fuerza de empuje que produce el bimetal durante la exposición a la temperatura fija T_c , y, simultáneamente, un sensor LVDT determina el desplazamiento que experimenta. Para lograr la medición

de fuerza, la galga extensiométrica es precargada con un 50% de la fuerza teórica esperada $F_{precarga}$, mientras que la fuerza que se produce durante el desplazamiento es la delta de la fuerza de precarga y la medición $F_{real} = F_{medida} - F_{precarga}$.

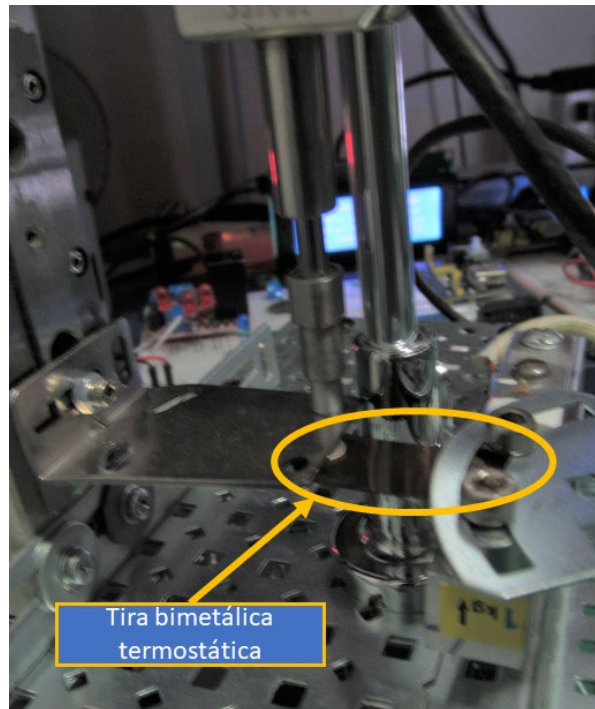


Figura 53. Prototipo caracterizador de lámina bimetalica.
Elaboración propia.

Para la caracterización los bimetales se seleccionan muestras aleatorias de diferentes grupos de bimetales termostáticos y se colocan en la cámara térmica a una temperatura de T_f en $^{\circ}C$.

Posteriormente se genera la matriz con los datos de tiempo-desplazamiento hasta que se realiza la acción de *snap* del bimetale.

Después, la cámara térmica cambia a T_c en $^{\circ}C$ y se repite el proceso, pero ahora se espera que regrese a la posición original. Una vez hecho todo esto, cada una de las muestras es caracterizada con el método B 106-08 tradicional que determina el perfil de fuerza-desplazamiento.

Se agrega a los datos de entrada la información de las características del bimetálico, los espesores de los metales y el coeficiente de expansión térmica de cada uno de ellos, todo este grupo de datos crea un vector de identificación por cada muestra.

Todos los datos obtenidos en las pruebas son alimentados durante el proceso de aprendizaje del Algoritmo Híbrido PAT-ELM.

Los parámetros usados en el Algoritmo se establecen como nodos ocultos $L = 30$ y 100 muestras con una función de activación sigmoïdal.

4.1.12. Salida pronosticada por algoritmo híbrido PAT-ELM

La salida del Algoritmo Híbrido Neuronal PAT-ELM es una predicción del perfil de fuerza que tendrá el bimetel, este representa el comportamiento real del bimetel termostático al exponerse a las temperaturas T_f y T_c . La salida se da en la forma de una matriz de 2×350 , la cual puede ser utilizada posteriormente para determinar la curvatura óptima de una tira bimetálica para un moto-protector.

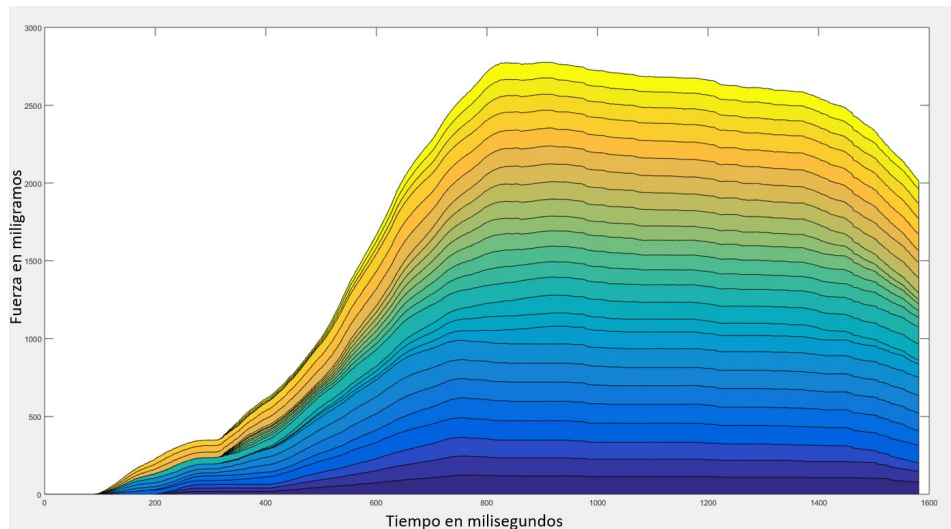


Figura 54. Perfil de fuerza de pronóstico de la tira termostática.
Elaboración propia.

La Figura 54 es una gráfica del perfil de fuerza de la tira termostática. Está formada por una familia de curvas de la fuerza generada por la tira termostática al exponerla a T_c , iniciando a temperatura T_f : el eje horizontal es el tiempo y el eje vertical la fuerza. Las curvas de pronóstico son generadas por el Algoritmo Híbrido AHN-ELM. Cada curva corresponde a una curva de desplazamiento obtenida al calentar el bimetel en caracterización.

La Figura 55 muestra el perfil dinámico de desplazamiento que se proporciona a la Red Neuronal PAT-ELM como entrada durante el entrenamiento. Es una familia de curvas del desplazamiento generado por la tira termostática al exponerla a T_c , iniciando desde la temperatura T_f .

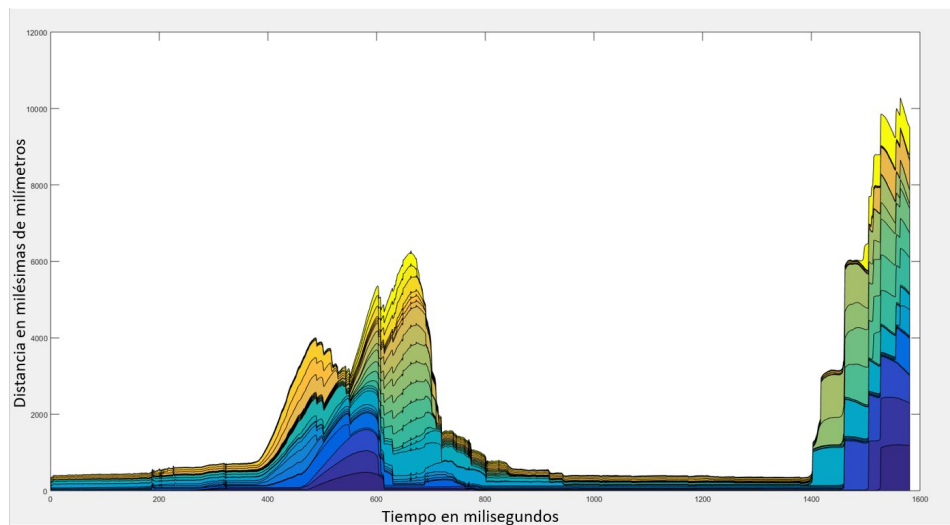


Figura 55. Perfil dinámico de desplazamiento de lámina bimetálica.
Elaboración propia.

Estas curvas se obtienen al aplicar las temperaturas T_f y T_c al bimetel en caracterización y medir dinámicamente el desplazamiento que producen F . Las curvas de desplazamiento y fuerza son la información de entrada y salida objetivo para el entrenamiento del Algoritmo Híbrido PAT-ELM.

Ambas gráficas muestran, en superposición, las diferentes mediciones a diferentes temperaturas de prueba y las salidas en fuerza de empuje que son usadas para el entrenamiento del Algoritmo Híbrido PAT-ELM. De forma tal, que, al recibir un vector

con la información de desplazamiento, el algoritmo predice la respuesta en fuerza generada de la tira termostática.

4.1.13. Indicadores de desempeño de la red PAT-ELM

El desempeño la Red Neuronal PAT-ELM es evaluado usando los siguientes indicadores (Japkowicz & Shah, 2011):

- Coeficiente de correlación R^2 , que proporciona una medida de la variabilidad de los pronósticos producidos por el modelo.
- El error absoluto medio MAE .
- El error cuadrático medio $RMSE$, que mide los errores residuales.

Estos indicadores, en conjunto, proporcionan una idea global de las diferencias entre las observaciones y el Algoritmo.

Primero se calcula el error cuadrático medio de la línea base usando la Ecuación 37 donde y_i son las mediciones directas y la \bar{y}_i es la media de las mediciones observadas.

$$MSE_{lineaBase} = \frac{1}{N} \sum_{i=1}^N (y_i - \bar{y}_i)^2$$

Ecuación 37. MSE error cuadrático medio línea base.

Después se calcula el error cuadrático medio de la predicción del modelo con la Ecuación 38, donde y_i es el resultado real esperado y \hat{Y}_i es la predicción del modelo.

$$MSE_{Modelo} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{Y}_i)^2$$

Ecuación 38. MSE error cuadrático medio del modelo.

Las MSE de línea base, así como la del modelo, miden el error cuadrado promedio de las predicciones. Para cada medición se está calculando la diferencia cuadrada entre las predicciones y el objetivo, y después, se promedian los resultados.

La forma de interpretarlo es, que entre mayor sea el valor de la MSE, peor es el modelo evaluado.

Posteriormente se calcula el error como un promedio de diferencias absolutas entre los valores objetivo y las predicciones. El índice estadístico MAE es un análisis sobre una escala lineal, lo que se traduce en que todas las diferencias individuales se ponderan por igual en la misma media. La Ecuación 39 muestra cómo se calcula.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \bar{y}_i|^2$$

Ecuación 39. MAE Error absoluto medio.

La principal ventaja del índice MAE es que no es tan sensible a los valores atípicos como el error cuadrático medio, por esa razón se evalúan ambos MSE y MAE, ya que permiten observar el comportamiento con y sin el efecto de los datos anormales.

La raíz del error cuadrático medio se usa para hacer que la escala de los errores sea igual a la escala de los objetivos. La Ecuación 40 muestra cómo se realiza.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

Ecuación 40. RMSE Error cuadrático medio.

Una vez calculados los errores medios cuadráticos $MSE_{LineaBase}$ y MSE_{Modelo} , se calcula el coeficiente de determinación R^2 mostrado en la Ecuación 41:

$$R^2 = 1 - \left(\frac{MSE_{Modelo}}{MSE_{LineaBase}} \right)$$

Ecuación 41. R^2 Coeficiente de determinación.

El índice coeficiente de determinación nos da la proporción entre el modelo y la línea base; donde un valor de 1 significa que es un excelente modelo. Si el resultado es 0 o negativo significa que no existe ninguna predicción acertada del modelo.

4.1.14. Análisis desempeño algoritmo híbrido PAT-ELM

El análisis de desempeño del Algoritmo Híbrido PAT-ELM consiste en hacer la comparación entre los indicadores estadísticos de desempeño generados por el Algoritmo y las mediciones físicas. Este análisis nos va a ayudar a entender qué tan buenas son las predicciones hechas por el Algoritmo. Los indicadores a considerar son: fuerza máxima alcanzada para la apertura, medida en equipo de caracterización original F_{Medida} ; fuerza máxima pronosticada por modelo PAT-ELM para la apertura del bimetal $F_{PAT-ELM}$; tiempo de apertura a temperatura de disparo (T_c) $T_{apertura}$; tiempo de cierre a temperatura baja (T_f) T_{cierre} y correlación R^2 entre perfil de fuerza medido vs perfil de fuerza pronosticado.

La Figura 56 muestra la salida del caracterizador prototipo cuando se prueba a la temperatura T_c . El eje horizontal muestra el tiempo en milisegundos en que ocurre el desplazamiento, el eje vertical representa el desplazamiento de la lámina, desde su posición inicial a temperatura T_f , hasta su posición final, cuando alcanza la temperatura T_c .

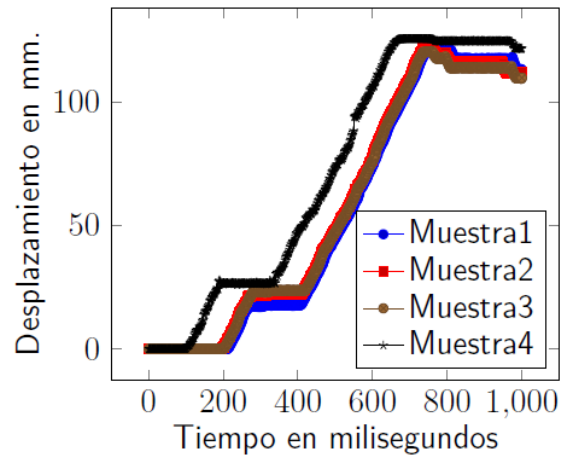


Figura 56. Perfil de desplazamiento de varias láminas bimetalicas al exponerlas en el tester. Elaboración propia.

4.1.15. Resultados del monitoreo cognitivo

Durante la experimentación, se analizaron siete grupos de tiras bimetalicas, cada grupo se compuso por 50 muestras. El tamaño de muestra fue el seleccionado en tanto representa un número típico de piezas que se pueden obtener al iniciar la producción de un moto-protector. Un número mayor de muestras no es normalmente justificable, por el costo de los materiales. El objetivo de analizar los siete grupos es determinar si es posible confiar en los resultados de pronóstico del simulador. Los pronósticos se comparan con los resultados medidos usando el método ASTM-B106-08 en los mismos grupos. Para realizar la comparación, se utilizan los índices estadísticos correlación, *MAE* y *RMSE*.

El índice estadístico de correlación indica que existe una relación entre dos variables. En este caso, el perfil de fuerza medida y el perfil de fuerza pronosticada. Con ello se espera observar que los valores del pronóstico del simulador varían sistemáticamente con los valores de las mediciones realizadas. Para interpretar el resultado de correlación, un valor de 1 significa que tanto la medición como el pronóstico corresponden en la misma dirección y en la misma proporción, por lo tanto, se consideran iguales. Resultados de 0 y cercanos a 0, significan que no son

iguales. Los índices estadísticos *MAE* y *RMSE* representan el error que existe entre la medición y el pronóstico de la Red Neuronal PAT-ELM y qué tanta dispersión existe entre ambos resultados. Por lo tanto, se puede interpretar que entre más pequeños sean ambos índices estadísticos, existe menor error y menor variación en el pronóstico de la Red Neuronal PAT-ELM.

La Tabla 5 indica los grupos experimentales, la fuerza medida promedio del grupo F_M , la fuerza pronosticada por la Red Neuronal PAT-ELM, el tiempo de apertura del bimetálico al exponerse a la temperatura de *snap*, el tiempo para volver a la posición original al exponerse a la temperatura baja T_c el índice estadístico de correlación entre la medición de la fuerza que produce la tira bimetálica y el pronóstico generado por la Red Neuronal PAT-ELM.

Tabla 5. Análisis de correlación entre el modelo y las mediciones de varios grupos de prueba.

Grupo	F_M	$F_{PAT-ELM}$	T_a	T_c	R^2
28.6 x 12.7 A	1117	1100	0.13	0.21	0.82
21.1 x 9.4 A	897	860	0.12	0.22	0.87
16.2 x 7.8 A	937	926	0.12	0.21	0.85
28.6 x 12.7 B	1085	1050	0.12	0.23	0.71
21.1 x 9.4 B	1149	1078	0.14	0.24	0.72
16.2 x 7.8 A	906	1004	0.16	0.21	0.92

Elaboración propia.

4.1.16. Índices estadísticos

Los índices estadísticos nos permiten estudiar la variación de un fenómeno, para comprender la relación que hay entre los cambios que sufre y comparar sus diferencias en forma numérica.

Para evaluar el Algoritmo Híbrido PAT-ELM se seleccionó el índice estadístico coeficiente de determinación, denominado R^2 . En este caso el coeficiente determina la calidad del Algoritmo Híbrido PAT-ELM para replicar los resultados con base en los datos de entrenamiento, y la proporción de variación de los resultados que puede pronosticar el algoritmo.

La Tabla 6 muestra los resultados de correlación que van desde 0.71 a 0.92, lo que implica que el pronóstico de la Red Neuronal PAT-ELM tiene similitud a las mediciones concretas de los mismos grupos. También los indicadores de desempeño, en ella podemos observar que la correlación R^2 en el caso de precisión, es decir, la capacidad de predecir el perfil de fuerza del lote de producción de donde se obtuvieron las piezas de entrenamiento es de 0.86492; lo que significa que tiene una buena capacidad de predecir el perfil de fuerza con respecto a las mediciones reales.

Tabla 6. Indicadores del desempeño del algoritmo PAT-ELM.

Indicador	Precisión	Generalización
R^2	0.86492	0.83217
MAE_{Modelo}	0.19976	0.23663
$RMSE$	0.25798	0.30129

Elaboración propia.

Normalmente, para este tipo de modelos, las correlaciones menores a 0.5 son descartadas, pero en este caso, todos los índices de correlación de cada uno de los grupos están por encima de este criterio de aceptación, por lo que no es necesario descartar.

La correlación con respecto a la generalización es de 0.83217, 3.3% menor con respecto a la precisión, la cual sigue siendo adecuada para predecir el perfil de fuerza, aun cuando las muestras no sean del mismo lote de producción que se está evaluando en la Red Neuronal PAT-ELM.

Esto significa que, con 50 muestras, muchos de estos modelos pueden utilizarse en un ambiente de producción industrial, ya que el material que se utilice para el entrenamiento no tendrá un alto impacto económico.

Otros dos métricos utilizados para analizar los resultados de la Red Neuronal PAT ELM fueron el análisis de precisión y el de generalización, también conocidos como *training error* y *test error*. Dichos análisis nos permiten evaluar la capacidad de

generalización del modelo en otros lotes de producción del mismo tipo de bimetálico y misma geometría.

El índice estadístico de promedio de error absoluto *MAE* en el caso del lote de producción del que se obtuvieron las piezas es 0.199776 que, aunque es relativamente alto, puede ser mejorado significativamente si se incrementa el número de muestras de entrenamiento. Este error absoluto permite determinar si el lote de producción puede utilizarse o no para la fabricación de los moto-protectores. En el caso del índice *MAE* para generalización, se incrementa a 0.23663, que sigue siendo adecuado para su utilización en los moto-protectores. Es importante notar que también puede reducirse si se incrementa el tamaño de muestra. Por último, los índices estadísticos *RMSE*, tanto 0.25798 como 0.30129, indican una variabilidad normal asociada a este tipo de predicción, la cual se considera dentro de los límites normales.

4.1.17. Conclusiones preliminares algoritmo PAT-ELM

Respecto al contenido de la sección 4.1, se puede concluir que el experimento implementado para la caracterización de tiras bimetálicas termostáticas por medio de la aplicación del nuevo Algoritmo Híbrido PAT-ELM ha sido exitoso.

Con él se confirmó la eficacia de usar el Algoritmo Híbrido PAT-ELM como algoritmo de monitoreo cognitivo.

La caracterización de la referencia para el entrenamiento de la Red Neuronal se realizó con base en la norma ASTM-B106-08.

Se probó para seis grupos de bimetálicos y fue capaz de pronosticar las salidas de temperatura de apertura T_f y cierre T_c , con correlaciones para cada grupo, que van desde 0.71 a 0.92, logrando una precisión en general de 0.86 y una generalización del 0.83.

Una ventaja de este algoritmo es que, aunque se entrena con una cantidad muy reducida de piezas, es capaz de tener un buen resultado a diferencia de otras

redes neuronales que no podrían funcionar apropiadamente con este tamaño de muestras.

Se obtuvieron muy buenos resultados dentro de los parámetros que se requieren en el ambiente de manufactura.

4.2. APLICANDO EL ALGORITMO HÍBRIDO AHN-ELM AL CONTROL AUTOMÁTICO DEL FORMADO DE BIMETALES

4.2.1. Introducción al AHM para formado de bimetales

La sección 4.1 presentó el primer experimento de monitoreo cognitivo, en el que se emplea el Algoritmo Híbrido PAT-ELM. En esta sección 4.2 se presenta el segundo experimento de control cognitivo, que consiste en aplicar el Algoritmo Híbrido de la Red AHN-ELM a una máquina de formado de bimetales termostáticos. En un primer momento, se presentan las especificaciones pertinentes sobre el formado de los bimetales en la industria, para que sea clara la función que se busca controlar por medio del Algoritmo Híbrido AHN-ELM propuesto. Después, se expone el propósito del experimento de control cognitivo, ahondando en el procedimiento de la experimentación y en los criterios de aceptación que se esperan obtener. El experimento se realiza en una máquina prototipo de formado de bimetales, cuyas características se presentan a detalle.

Una vez presentado el panorama general del experimento, se profundiza en la funcionalidad de la máquina prototipo automática, en la que se implementa la Red AHN-ELM. Posteriormente, se presentan los criterios de selección de los materiales de pruebas y se describe el desarrollo de la experimentación paso por paso. Finalmente, se presentan los resultados de la implementación de la Red AHN-ELM en un proceso de manufactura y se muestra su desempeño.

4.2.2. Antecedentes

Los interruptores tipo moto-protector tienen uno o más elementos de accionamiento, como fuelles, diafragmas, broches bimetálicos o tiras bimetálicas,

que producen la fuerza necesaria para operar el cambio de transición instantáneo de apagado a encendido o viceversa.

El moto-protector que nos concierne aquí, utiliza una lámina bimetálica que mueve los contactos eléctricos cuando la temperatura aumenta o disminuye, y produce una desconexión o conexión eléctrica para salvaguardar el buen funcionamiento del circuito y componentes que protege.

El circuito básico de conexión del moto-protector basado en lámina bimetálica se muestra en la Figura 57, el circuito muestra un motor que está en serie con una fuente de energía, en este caso se trata de una batería, y, en serie con ellos, se encuentra la lámina bimetálica, que está haciendo contacto eléctrico con un punto de contacto estacionario. La posición de la lámina bimetálica es tal que, cuando se calienta el metal, se separa del contacto estacionario, lo que genera la desconexión del circuito.

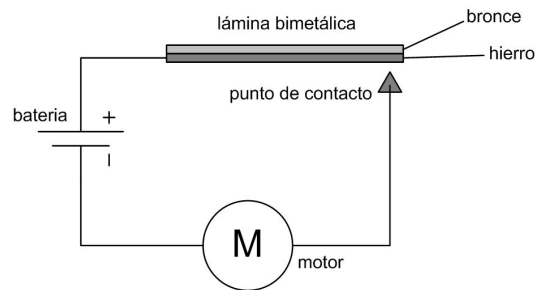


Figura 57. Circuito básico de un moto-protector de lámina bimetálica.
Elaboración propia.

La Figura 58 muestra en detalle el cambio mecánico que ocurre internamente en un moto-protector basado en lámina bimetálica calibrada al ser expuesta a un cambio de temperatura. En la Figura 58(a) la lámina bimetálica está en la condición estacionaria o inicial del interruptor a temperatura ambiente, la cual mantiene el circuito eléctrico del moto-protector cerrado. La Figura 58(b), muestra al moto-protector disparado por una sobre temperatura; en la figura puede apreciarse como, al separarse la lámina del contacto, el circuito eléctrico del moto-protector se abre, desconectando al motor. La lámina bimetálica cuenta

con una curvatura en su estructura que sirve para generar dos funciones: la primera es provocar una desconexión instantánea, al forzar el desplazamiento con una precarga de fuerza y, la segunda, es ajustar con exactitud la temperatura a la que se desplaza la tira, con el fin de producir la apertura y el cierre. Al proceso que se emplea para darle a la lámina la geometría necesaria, en este caso su forma curvada, se le llama formado o curvado de la lámina bimetálica.

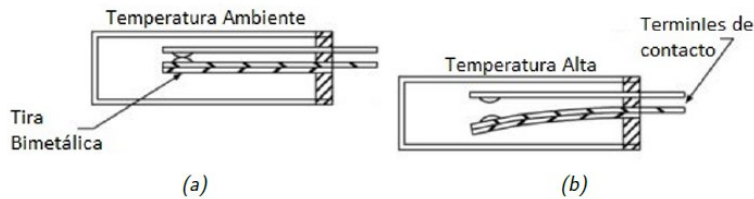


Figura 58. Detalle de cambio mecánico de la lámina bimetálica.
Elaboración propia.

El proceso industrial de curvado de la tira bimetálica termostática para su uso en interruptores del tipo moto-protector, se realiza en una máquina troqueladora. Se llama así porque cuenta con un troquel con múltiples insertos de formado. Las tiras del material bimetálico son cortadas a un tamaño específico antes de ser colocadas en el troquel y una vez cortadas, entran al troquel. El troquelado forma radios en la tira que la hacen funcionar como un resorte lineal ante los cambios de temperatura. Los insertos del troquelado también son llamados herramientas de formado y usualmente son montados en cavidades ajustables en el troquel, estos permiten subir o bajar su posición relativa, ayudando a ajustar los radios generados en la tira bimetálica.

El proceso de formado de la tira bimetálica se muestra mediante un esquema en la Figura 59, el cabezal cuenta con 3 servo motores de control de posición mecánica para la localización de las herramientas de formado.

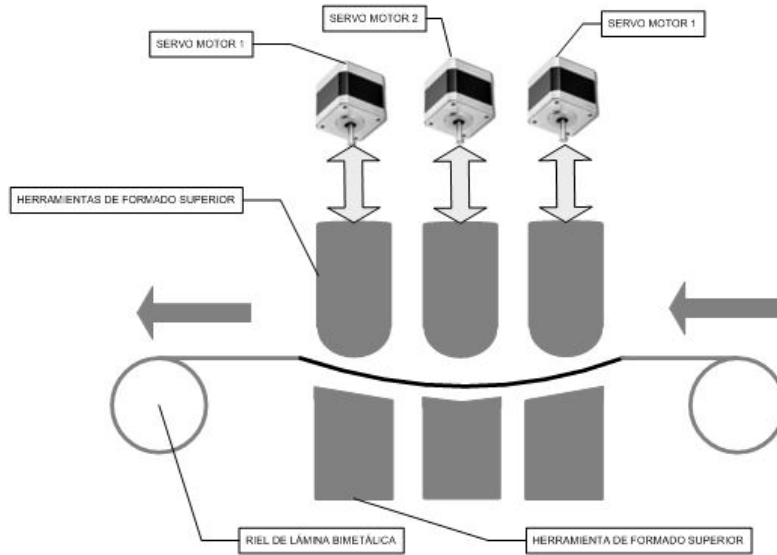


Figura 59. Esquema del troquel con herramientas de formado ajustables que dan forma a la curvatura del bimetal.
Elaboración propia.

Los servomotores y herramientas están montados en el troquel. Se puede observar el rollo de lámina, que se va trasladando dentro del cabezal de formado que cuenta con tres herramientas que, en cada paso, dan un estampado que afecta la geometría de la tira, produciendo una curvatura específica. La primera herramienta, afecta la característica de apertura instantánea; la herramienta central, produce un efecto en la temperatura de cierre instantáneo y, la última herramienta controlada por el servomotor #3, afecta todo el conjunto de apertura y cierre, simultáneamente. En este esquema también se representan tres servomotores, que son los responsables de cambiar la posición de las herramientas de manera automática.

La temperatura de apertura instantánea de la lámina bimetálica depende del formado y de los metales que componen al bimetal.

4.2.3. Propósito del experimento

El propósito de este segundo experimento es validar el funcionamiento de la Red Neuronal Hipocámpica AHN-ELM, en su modalidad de control cognitivo,

aplicándola a una máquina de formado de bimetales, con la intención de evaluar el desempeño del Algoritmo en el control del proceso de manufactura. Se desea evaluar la capacidad de aprendizaje automático y la eficiencia con las que el Algoritmo puede operar. Otro de los propósitos del experimento es determinar los parámetros e interfaces de comunicación requeridos para hacer operacional el Algoritmo en aplicaciones reales.

4.2.4. Procedimiento experimental

Para la implementación de la Red Neuronal Hipocámpica AHN-ELM en la máquina de formado de tira bimetálica, se define el siguiente método experimental paso por paso:

- 1. Selección de máquina prototipo.** Como primer paso se selecciona una máquina de producción para el formado de la tira bimetálica. En este caso, se seleccionó una máquina independiente del flujo principal de producción, lo que permite realizar la experimentación sin afectar la producción durante la implementación del algoritmo. El producto que genera esta máquina se integra posteriormente, a la producción normal.
- 2. Adecuación de máquina prototipo para trabajo automático.** Después, se adecúa la máquina para que pueda tener un actuador automático en el cabezal de formado, así como un medidor de posición automático para conocer la posición actual de cada una de las herramientas de formado. Originalmente, la máquina tenía sólo controles manuales para los ajustes de posición de las herramientas. En este caso, se modificó el cabezal de formado para incorporar 3 servo motores de control de posición y 3 micrómetros electrónicos. También se le agregó una computadora, con el fin de gestionar todos los dispositivos incorporados.
- 3. Desarrollo sistema de control de prototipo.** El tercer paso consiste en desarrollar el programa de control de la máquina prototipo, utilizando el programa Labview de *National Instruments* para facilitar el control de los servos motores y la conexión de los micrómetros a través de los puertos USB. El programa de control de la máquina prototipo se elabora para que sea

capaz de controlar los parámetros de los servos motores. El programa funciona leyendo una tabla de configuración en el disco duro, de donde obtiene los parámetros iniciales de los servos motores y guarda los resultados de los micrómetros. El programa remoto accesa a este archivo cuando ha logrado mejorar los parámetros, de tal forma que el archivo se utiliza para interconectar parte de la información episódica entre los dos programas. Se crea la comunicación Bluetooth para la memoria de trabajo, con el fin de poder ajustar los parámetros que continuamente están modificando la máquina para la mejora del desempeño.

- 4. Interfaces de comunicación AHN-ELM y prototipo.** Luego se desarrollan las interfaces para conectar la máquina de formado con una computadora remota, debido a que el control de la Red Neuronal AHN-ELM no está en la computadora de control de la máquina. Para esto, se implementaron dos tipos de interfaces, una de ellas utilizando Ethernet y, la segunda, a través de Bluetooth
- 5. Implementación de red neuronal AHN-ELM a la máquina prototipo.** Una vez completados los pasos anteriores, se implementa la Red Neuronal AHN-ELM en la computadora remota. Se carga el algoritmo de la Red Neuronal para que sea capaz de controlar los tres servos motores. A continuación, se definen el número de neuronas, los valores de configuración iniciales, la función de recompensa y los demás parámetros necesarios para que la Red Hipocámpica funcione.
- 6. Selección material de prueba.** Después se selecciona el material termostático para realizar el entrenamiento de la máquina.
- 7. Registro de resultados.** Por último, la información que se colecta representa el rendimiento final de la operación, así como el número de cambios o acciones de mejora que realiza el algoritmo para alcanzar el punto óptimo en la formación de las tiras bimetálicas.

4.2.5. Criterio de aceptación de pruebas

El criterio de aceptación para saber si se alcanzó el objetivo del experimento, consiste en evaluar el desempeño de la máquina de formado, al realizar el proceso de aprendizaje automático utilizando el Algoritmo Híbrido AHN-ELM. El desempeño se mide con el métrico denominado yield.

El yield es el rendimiento de un proceso de manufactura, se refiere al porcentaje de productos no defectuosos con respecto al total de productos de un lote dado y, generalmente, se indica mediante la relación entre el número de artículos no defectuosos y el número de artículos fabricados.

El yield se calcula como muestra la Ecuación 42. en la que Y simboliza el yield operacional de la máquina de formado, ω es la cantidad de bimetales formados dentro de la especificación de temperatura, χ es el total de bimetales producidos en el lote de producción evaluado.

$$Yield = Y = \left(1 - \frac{\omega}{\chi}\right) \times 100$$

Ecuación 42. Yield

4.2.6. Selección y adaptación de máquina prototipo para experimento

La máquina prototipo seleccionada es una prensa troqueladora, especializada en el formado de discos bimetálicos. Está diseñada para ser manipulada por un solo operador, cuya función es la de cargar y descargar de forma manual las tiras bimetálicas.

La prensa especializada tiene un troquel de accionamiento neumático de 0.5 toneladas de capacidad, con 3 insertos de formado.

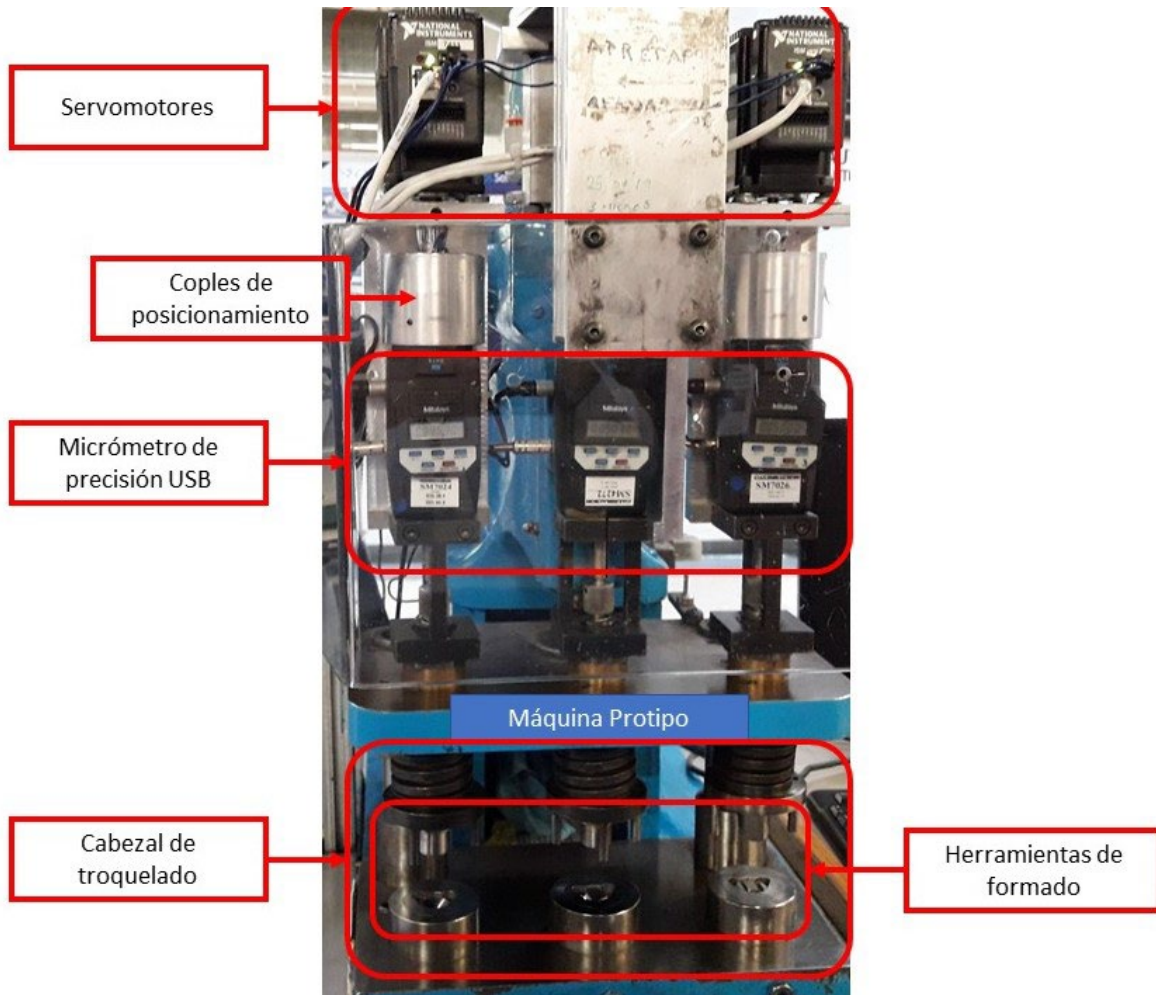


Figura 60. Máquina prototipo, es una troqueladora formadora de bimetales.
Elaboración propia.

La Figura 60 muestra la máquina prototipo de formado de lámina bimetálica para prueba del Control Cognitivo AHN-ELM. En ella, se pueden observar el cabezal con el troquel de formado y 3 servo-herramientas que posicionan los dados de formado. En la parte superior, se pueden observar los servomotores que mueven las herramientas; hacia el centro los 3 micrómetros de precisión electrónica tipo encoder, que mandan la información de posición a la computadora y, en la parte inferior, se pueden ver los 3 nidos de formado de la tira bimetálica.

La máquina seleccionada no cuenta, originalmente, con una estación para medir los tiempos de apertura y cierre de las tiras bimetálicas, que resulta necesario para el aprendizaje automático que se implementa aquí. Sin embargo, su localización

es muy próxima a otra máquina en la que se pueden realizar pruebas de temperatura a la tira bimetálica, por ende, se considera dicha operación de medición como parte de la implementación del Algoritmo, específicamente para la retroalimentación al sistema de control cognitivo que se desarrolla.

Para la adaptación de la máquina prototipo, se realizaron las modificaciones mecánicas y las automatizaciones necesarias a la prensa troqueladora, con el fin de adaptarle 3 servomotores de control y 3 micrómetros electrónicos tipo encoder, que se emplearán para la medición de la posición de las herramientas de formado.

La Figura 61 es una imagen de un detalle del troquel, en donde se pueden observar las tres herramientas de formado usadas en la máquina prototipo.

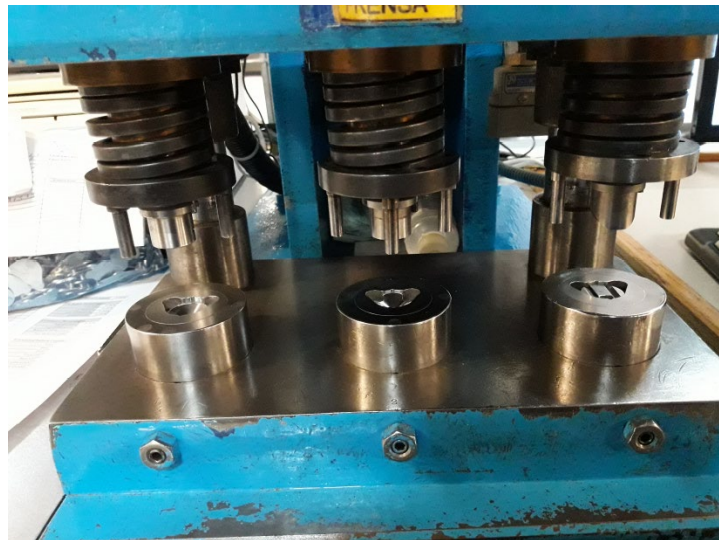


Figura 61. Detalle de las herramientas de formado del bimetál en la máquina prototipo.
Elaboración propia.

Las herramientas funcionan de modo progresivo, es decir, primero se monta el disco sin curvatura en la herramienta izquierda, posteriormente se da el primer ciclo de troquelado, en el que baja el troquel y comprime a la tira bimetálica, después, se cambia manualmente la tira semi formada a la herramienta central, se carga en la herramienta de la izquierda un nuevo disco y se hace otro ciclo de troquelado. Finalmente, la tira semi formada se mueve de la herramienta central a la derecha

y se repiten los dos pasos anteriores con las otras herramientas. Se realiza un tercer ciclo de troquelado y con ello se logra un disco completamente formado, con la curvatura apropiada para producir el *snap* y se use en los moto-protectores.

La Figura 62 muestra la máquina de prueba de temperatura, encargada de proporcionar la respuesta de los tiempos de apertura y cierre de la tira bimetálica termostática a las temperaturas T_c y T_f , respectivamente.

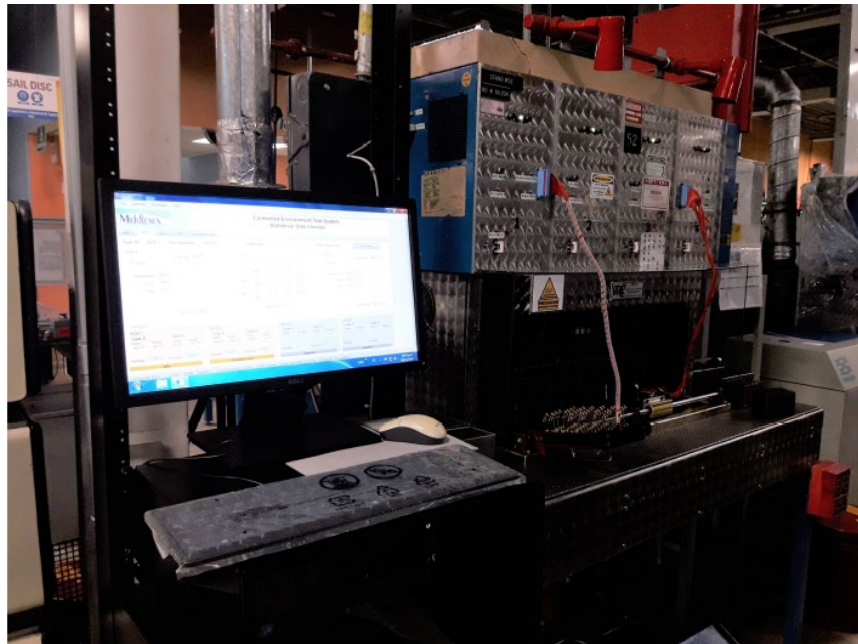


Figura 62. Imagen de la máquina de prueba de temperatura para la tira bimetálica termostática.
Elaboración propia.

Esta máquina se encuentra separada de la máquina prototipo, por tanto, se traslada manualmente el material hasta la máquina de prueba de temperatura.

La forma en que se realiza la prueba de temperatura es la siguiente: primero, se ajusta la máquina prototipo; el Algoritmo Híbrido AHN-ELM sugiere un ajuste de micrómetros conforme a los eventos episódicos memorizados.

Los discos se forman en una muestra, usualmente de 30 piezas, que se lleva a la máquina de pruebas de temperatura.

Las tiras bimetálicas se montan en un soporte especial y se sumergen en distintas finas. Estas finas contienen aceite a diferentes temperaturas establecidas.

A temperaturas altas, los bimetales tienden a hacer *snap*, flexionarse, mientras que, a temperaturas bajas, deben regresar a la posición inicial. La computadora de esta máquina de pruebas guarda la información, que después es enviada a la máquina prototipo.

Posteriormente, se envía dicha información sobre el tiempo de temperaturas de apertura y cierre a la computadora remota en donde se encuentra la red hipocámpica y, finalmente, dependiendo del resultado, si logro formar las láminas dentro de los límites de apertura y cierre esperados, se inicia el lote de producción. En caso de que el sistema hipocámpico sugiera un nuevo valor para optimizar el ajuste del lote a procesar, se realiza dicho ajuste antes de que inicie el siguiente lote de producción.

4.2.7. Desarrollo del sistema de control de prototipo

El desarrollo del Sistema de Control de máquina comienza con la instalación de la computadora que manejará todos los parámetros para el formado de lámina bimetálica en la prensa troqueladora. Esta computadora está montada en la máquina prototipo.

Sus funciones consisten en recibir los datos de ajuste de la temperatura de las cámaras térmicas y enviar los comandos de posición a los servomotores, estos ajustan, a su vez, las herramientas de formado.

El Sistema de Control de máquina se comunica vía Bluetooth con la interfaz cortical de la red hipocámpica AHN-ELM; para esto, utiliza un sistema embebido con módulo de comunicación Bluetooth.

La Figura 63 muestra el diagrama de bloques del Sistema de Control de Máquina implementado en la máquina prototipo, presenta los detalles del control de posición de las servo-herramientas con sus controladores, así como la interface bluetooth y la red intranet.

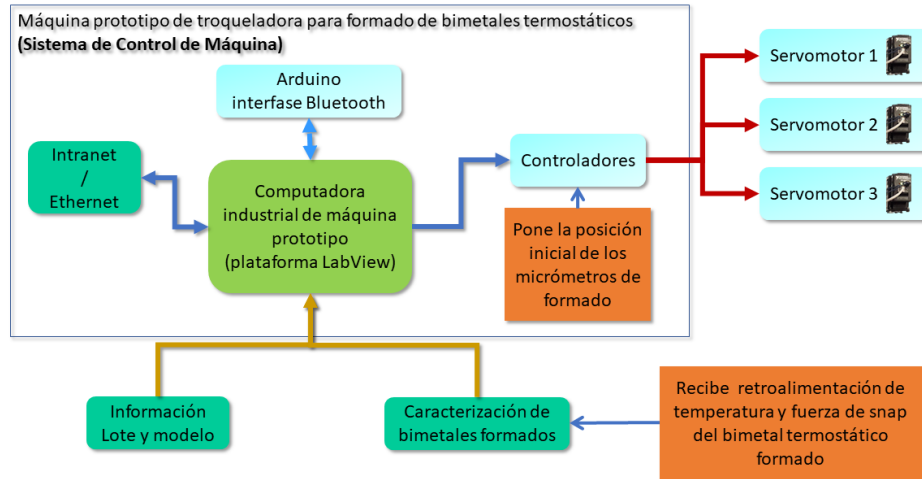


Figura 63. Diagrama a bloques de la máquina prototipo de formado de lámina bimetálica.
Elaboración propia.

Con el fin de implementar la Red Hipocámpica Artificial AHN-ELM, se emplea una interfaz entre el control de posicionamiento de los servomotores y la computadora del sistema cortical que contiene al Algoritmo Híbrido AHN-ELM.

La Figura 64 muestra una captura de pantalla de la computadora que se conectó a la máquina formadora prototipo. Dicha computadora sirve de interface con el operador para monitorear los datos de posición provenientes del control del Algoritmo Híbrido AHN-ELM. La pantalla muestra la base de datos de configuración de la máquina, utiliza el programa Labview de *National Instruments*.

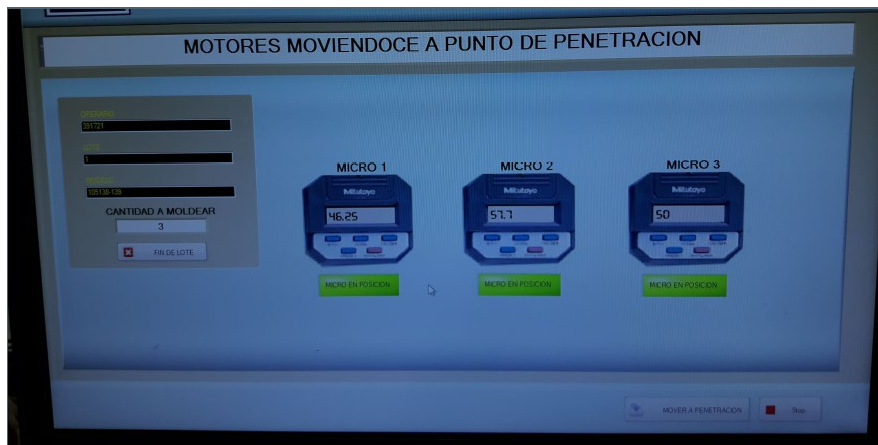


Figura 64. Imagen de pantalla de la computadora en la máquina formadora.
Elaboración propia.

La interfaz permite capturar y/o monitorear los datos del lote, así como la información relevante del material a procesar en la máquina. Esta información se comparte tanto para registros de la máquina prototipo, como para el Algoritmo Híbrido AHN-ELM. El algoritmo de Red Hipocámpica Artificial AHN-ELM está alojado en una segunda computadora. Esta computadora se encarga de recibir información de la aplicación de LabView, la procesa y la regresa a la misma aplicación. La decisión en tiempo real de controlar la máquina recae en el programa LabView, pero los valores de control que se utilizan para el formado son todos generados por el Algoritmo Híbrido AHN-ELM.

La máquina prototipo utiliza, para la comunicación, tres bases de datos que tienen diferentes propósitos.

- La primera base de datos está en la máquina formadora del Sistema de Control de máquina, que en primer lugar contiene la información del proceso, esto es, los parámetros de lámina termostática que se procesan. La retro alimentación de la temperatura de *snap* y las temperaturas de las cámaras térmicas y, por último, la posición de los servomotores.
- La segunda base de datos contiene los parámetros de aprendizaje de la Red Neuronal AHN-ELM
- La tercera base es el buffer temporal que contiene la memoria de trabajo

El sistema cortical maneja dos bases de datos y la máquina formadora maneja una de ellas. Ambas bases de datos interactúan con el Algoritmo Híbrido de la Red Neuronal AHN-ELM.

La Figura 65 muestra el diagrama de interconexión y las bases de datos utilizadas en la máquina prototipo, puede observarse el sistema cortical integrado y su interfaz para ejecutar el Algoritmo Híbrido AHN-ELM.

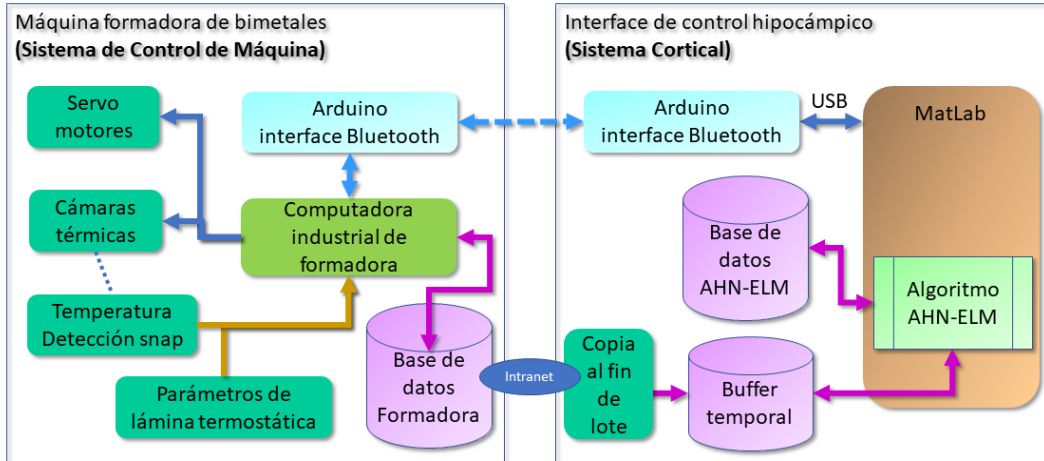


Figura 65. Diagrama de interconexión de la máquina prototipo troqueladora. Elaboración propia.

La interfaz del Sistema de Control de la máquina se comunica con la plataforma MatLab de la computadora remota, a través de las interfaces Bluetooth y el Intranet.

El resultado de la interacción de las tres bases de datos que están tanto en la máquina prototipo como en la computadora donde está instalada la Red AHN-ELM, tienen como resultado generación de la nueva posición de las herramientas formadoras, que, una vez ajustadas, cuando el cabezal neumático baje, dará la forma a la tira bimetálica conforme a los ajustes realizados.

La base de datos de la Red AHN-ELM es el equivalente de la memoria a largo plazo en el modelo de aprendizaje, mientras que la base de datos de la máquina formadora equivale a la memoria de corto plazo y el buffer temporal emula a la memoria de trabajo.

4.2.8. Modelo conceptual de la red neuronal AHN-ELM a la máquina prototipo

El modelo conceptual de la adaptación de la Red Neuronal AHN-ELM a la máquina prototipo muestra, en forma simplificada, los flujos de información que aparecen al aplicar la Red Neuronal. Dicho modelo conceptual se emplea para visualizar las

variables que deben interconectarse en la aplicación de la máquina prototipo, enfocado en los flujos de información.

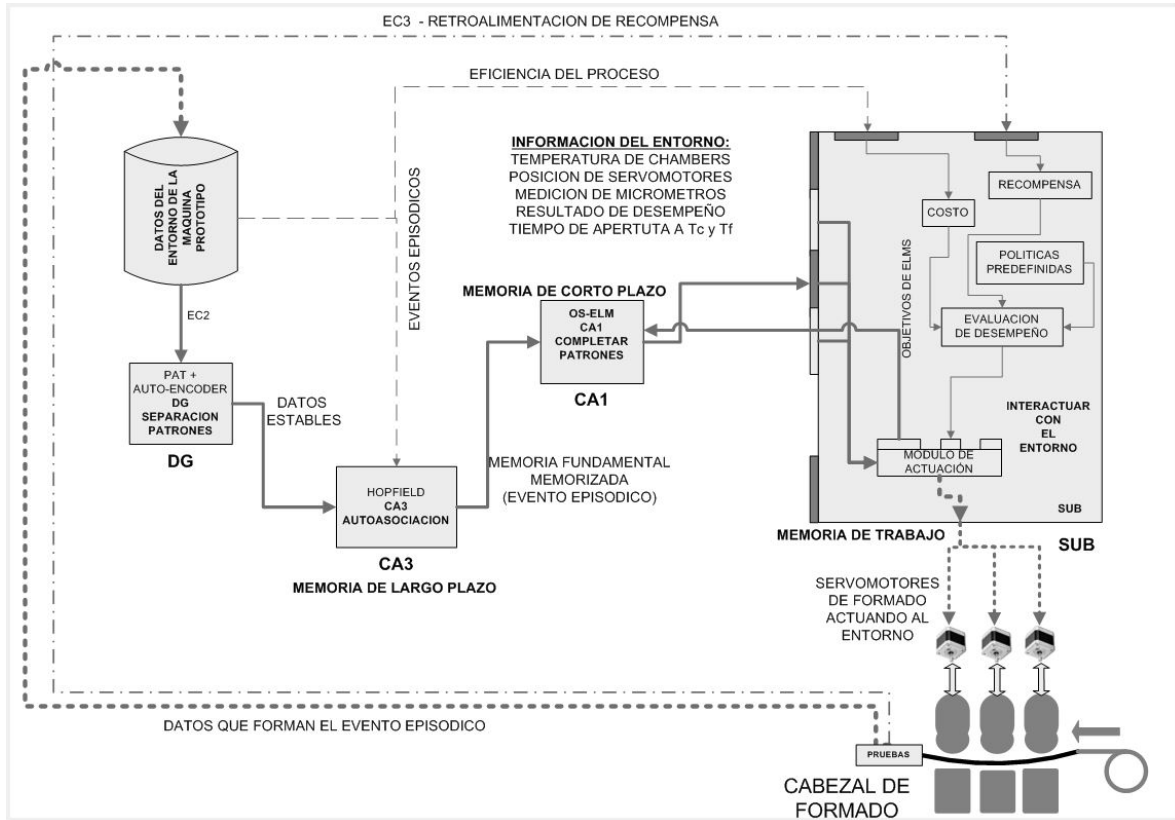


Figura 66. Esquema del modelo conceptual de red neuronal para el experimento de control cognitivo usando el algoritmo híbrido AHN-ELM. Elaboración propia.

La información del entorno se refiere a las mediciones del comportamiento del bimetal después del proceso de formado, tales como la temperatura de las cámaras térmicas, la posición actual de los servomotores, el tipo de bimetal y el desempeño esperado, entre otros.

La Figura 66 muestra el esquema del modelo conceptual de la Red AHN-ELM implementado en la máquina prototipo. En ella, se puede observar la interacción que se da entre las instancias hipocámpicas *DG*, *CA3*, *CA1* y *SUB* con los datos del entorno, que en este caso, consisten en la información sobre los parámetros del proceso, como tiempo de apertura y cierre del bimetal a las temperaturas de

prueba T_c y T_f ; la posición de los servomotores y la eficiencia del proceso especificada mediante el métrico de desempeño denominado yield. También, se detalla el interior de la instancia *SUB*, mostrando la interacción de las variables de recompensa y actuación. Se señala, además, la ubicación de las memorias de corto y largo plazo, así como la memoria de trabajo. El Algoritmo Híbrido AHN-ELM, controla al cabezal de formado con las 3 servo-herramientas.

4.2.9. Configuración de red AHN-ELM para sistema de control de máquina prototipo

La configuración que se requiere para implementar la Red AHN-ELM a la máquina prototipo se enfoca en definir los parámetros iniciales de las instancias hipocámpicas. Esto es, en definir, por ejemplo, cuántas neuronas ocultas o que función de activación se seleccionará según lo precise el proceso que se desea controlar.

La Tabla 7 contiene un sumario de los parámetros principales que es necesario configurar en la Red Hipocámpica AHN-ELM para dar inicio a la prueba. La tabla muestra los datos en 3 columnas, que contienen, en la primera, el tipo de elemento de la instancia hipocámpica que debe configurarse, en la segunda, la variable y, finalmente, los valores utilizados.

Tabla 7. Configuración de Instancias hipocámpicas establecidas para aplicar la red AHN-ELM en la máquina prototipo.

Elemento	Variable	Consideración Laberinto
Límite inferior laberinto de DG_a	$T_c - LSL_{DG}$	0.15
Límite superior laberinto de DG_a	$T_c - USL_{DG}$	0.20
Número de neuronas ocultas DG_b	n_{DGb}	1000
Número de neuronas ocultas $CA3$	n_{CA3}	500
Número de neuronas ocultas $CA1$	n_{CA1}	120
Número de neuronas salida DG_b	n_{DGb}	1000
Número de neuronas salida $CA3$	n_{CA3}	500
Número de neuronas salida $CA1$	n_{CA1}	120
Datos Entrenamiento inicial $CA1$	$n_{CA1Entren}$	50
Error permitido DG	ϵ_{DG}	0.01
Error permitido $CA3$	ϵ_{CA3}	0.01
Error permitido $CA1$	ϵ_{CA1}	0.01
Error permitido SUB	ϵ_{SUB}	0.01
Factor exploración gamma γ_{SUB}	α_{SUB}	0.4
Salidas de control $SUBN_{outputs}$	γ_{SUB}	3

Elaboración propia.

4.2.10. Arquitectura para AHN-ELM para control de formado de curvatura de bimetales termostáticos

La arquitectura de la red híbrida AHN-ELM para el formado de la curvatura de bimetales termostáticos se muestra en la Figura 67 en la cual se pueden observar las cuatro instancias hipocámpicas que componen a la red, utilizadas en el experimento del robot explorador. La entrada es la red de filtrado dinámico PAT la cual se encarga de identificar los tipos de bimetales termostáticos y las características de la lámina bimetálica a formar como ángulo, dimensiones y temperaturas de operación esperadas, esta sección se denomina DG_a .

De la sección PAT pasa el dataset filtrado para que se realice la separación de patrones, que consisten en expandir los datos de entrada usando el algoritmo codificador AHN-ELM, esta sección de la red se denomina DG_b . Una característica de esta red de auto-encoder es que las salidas y las entradas reciben los mismos datos, por lo tanto, la salida real de esta sección de la red son los pesos de la red oculta. Un requisito es que esta sección de la red es que se configura en el modo

RED HÍBRIDA AHN-ELM APLICADA AL CONTROL DE MAQUINA DE UNA FORMADORA DE CURVATURA DE LAMINA BIMETALICA

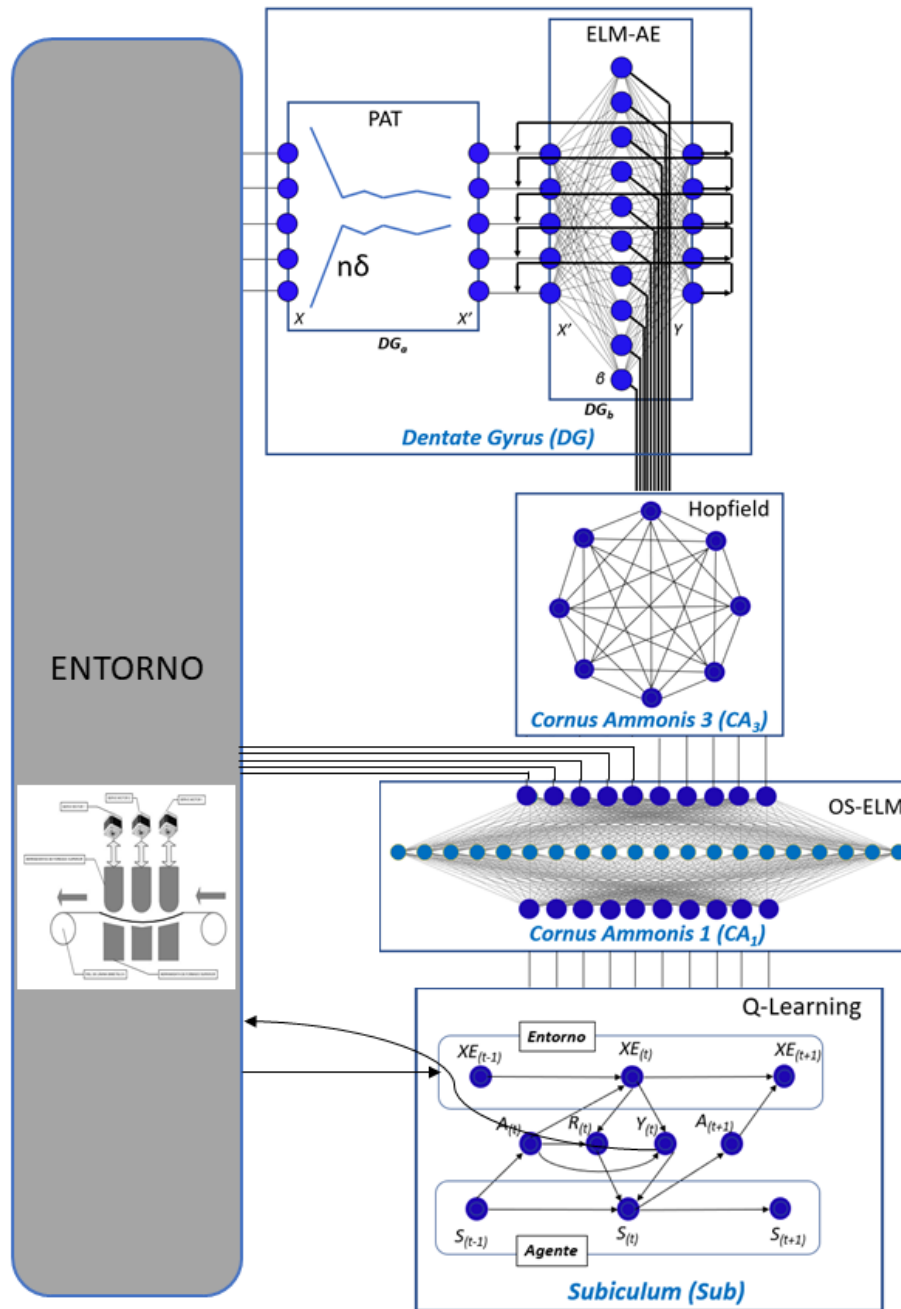


Figura 67. Arquitectura de la red híbrida AHN-ELM
Elaboración propia.

sparse, esto es, el número de neuronas ocultas es mayor al número de entradas para lograr la expansión del dataset y facilitar la diferenciación de patrones.

La siguiente red procesa los datos provenientes de DG_b , es la instancia $CA3$ que se implementa con una red recurrente Hopfield, la cual se utiliza para guardar los recuerdos episódicos del entorno. El objetivo de la red Hopfield es la de guardar un patrón y posteriormente recuperarlo. Una cualidad importante de esta sección de la red es que es capaz de reconocer un patrón, aunque éste contenga una pequeña cantidad de ruido.

posteriormente el patrón episódico recuperado se manda a la instancia hipocámpica $CA1$ que es una red de aprendizaje extremo configurada en modo de aprendizaje secuencial denominada OS-ELM, ahí recibe tanto el patrón de recuerdo episódico como la configuración utilizada en el ambiente de control para el sistema de respuesta.

Finalmente, la salida de la instancia hipocámpica $CA1$ alimenta a la instancia hipocámpica Subiculum SUB que es un algoritmo de aprendizaje por refuerzos, en este caso un algoritmo Q-Learning, la salida de este algoritmo va directamente al algoritmo de control de la máquina formadora que se encarga de mover 3 servomotores para posicionar las herramientas de formado en a las distancias correctas para lograr la curvatura de la lámina bimetálica esperada, esto para reaccionar de acuerdo a los resultados del proceso de formado que en este caso será parte del entorno del proceso..

La Figura 68 muestra la configuración y los parámetros principales utilizados en cada una de las instancias hipocámpicas utilizadas en el experimento del formado de curvatura de la lámina bimetálica termostática.

Arquitectura AHN-ELM en Formato Curvatura

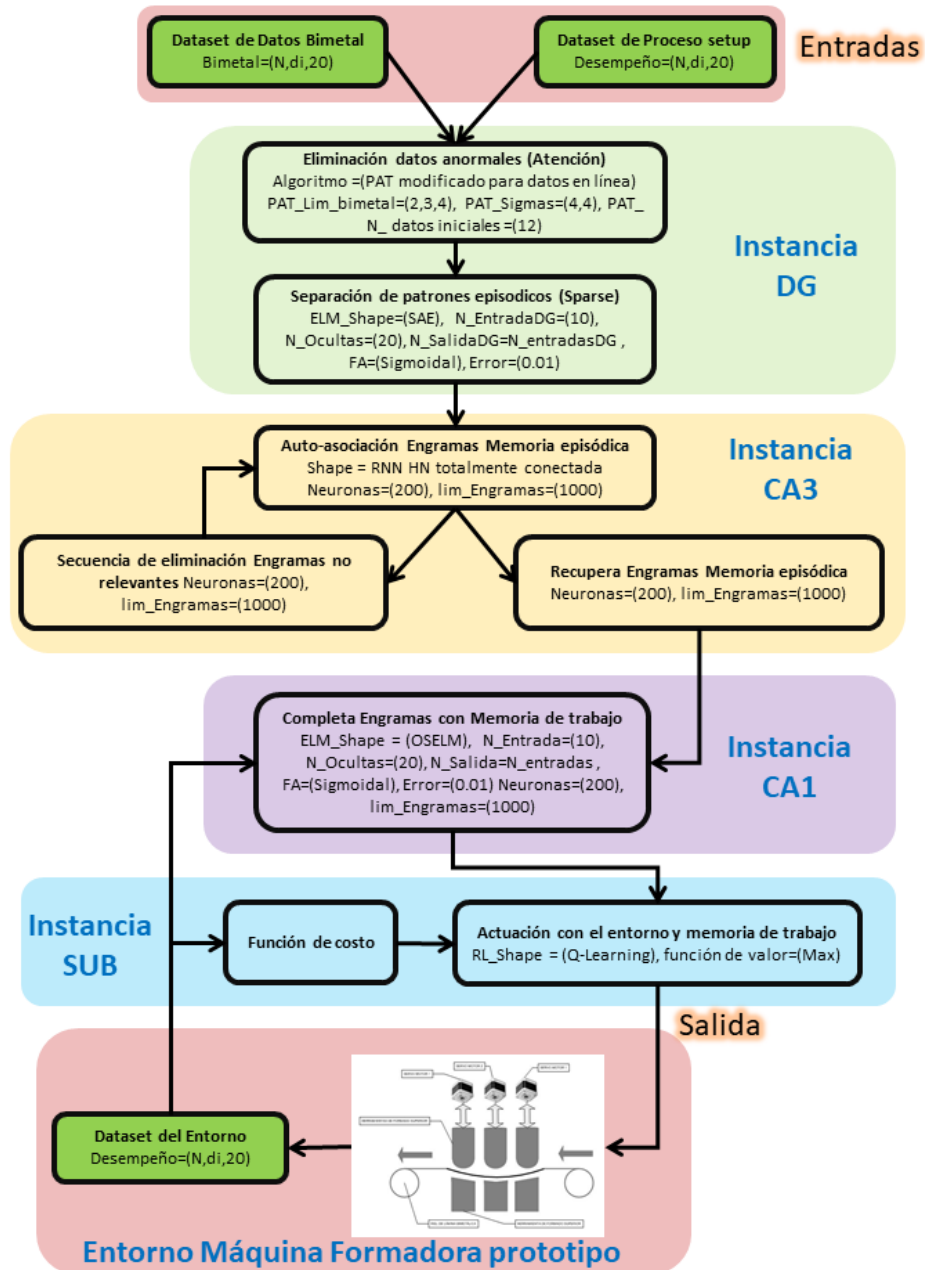


Figura 68. Arquitectura en bloques del algoritmo híbrido AHN-ELM. Elaboración propia.

4.2.11. Tamaño de muestra de experimentación

El tamaño de muestra se calcula a partir de dos variables, la primera es el porcentaje de confianza que queremos obtener de la muestra y la segunda es qué tanta variabilidad presenta la muestra.

Para determinar el tamaño de muestra que se empleará en la experimentación se requiere, primero, determinar los índices estadísticos de la muestra, estos son utilizados para determinar la variabilidad de los datos. Después, se debe definir el porcentaje de confianza que se desea obtener. Con los datos obtenidos, se calcula el error estándar del promedio con la Ecuación 43, la cual nos indica el error estándar con una confianza del 68%, de la población a evaluar, donde s es la estimación de la desviación estándar.

$$SE_{\bar{x}} = \frac{s}{\sqrt{n}}$$

Ecuación 43. Error estándar.

Para poder obtener una confianza del 95% en el tamaño de muestra, se adecúa la ecuación y se despeja Ecuación 44. La constante para $(z_{\alpha/2})^2$ es 1.96 para corresponder con el 95%, σ es la desviación estándar de la población a evaluar y E es el error mínimo permitido para la muestra del experimento.

$$n = \frac{(z_{\alpha/2})^2 \sigma^2}{E^2}$$

Ecuación 44. Tamaño de muestra con confianza del 95%.

En el caso del experimento 2, se determina el tamaño de muestra de dos variables significativas: T_c , que corresponde a la temperatura de apertura y T_f , que corresponde a la temperatura de cierre. Primero, se determinan los índices estadísticos de T_c y T_f , para lo cual se tomó una muestra de varios lotes de

producción, que consistió en total de 9,516 tiras bimetálicas, posteriormente, se utilizó aplicando el software Minitab para determinar los principales índices estadísticos, como la media de la muestra, la desviación estándar, entre otras.

Con base en ello, se generan dos resúmenes con los datos estadísticos más relevantes de las variables de temperatura de apertura T_c y cierre T_f , ambos fueron generados con el software estadístico Minitab, versión 16.

La Figura 69 muestra en un recuadro el resumen estadístico que determina la variación estándar de T_c , para el cálculo de tamaño de muestra con respecto a la temperatura de apertura, en él se puede observar que la media resultante es de 146.1 grados centígrados con una desviación estándar de 18.69 grados.

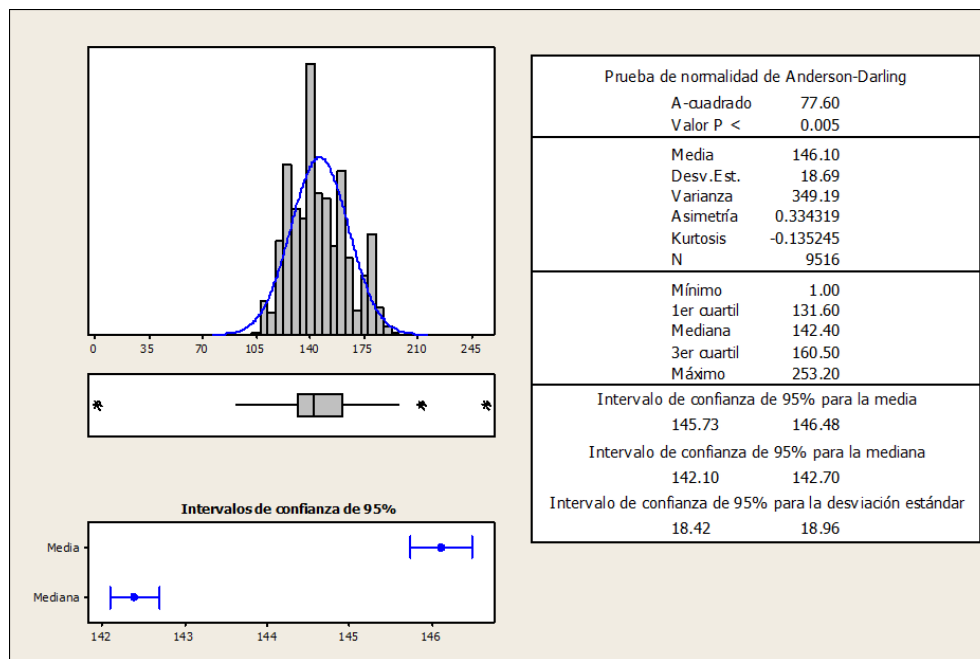


Figura 69. Recuadro de resumen estadístico de la muestra para T_c .
Elaboración propia.

La Figura 70 muestra un recuadro con el resumen estadístico que determina la variación estándar de T_f para el cálculo de tamaño de muestra con respecto a la temperatura de cierre. Se puede observar que la media es 62.112 grados centígrados con una desviación estándar de 3.922 grados.

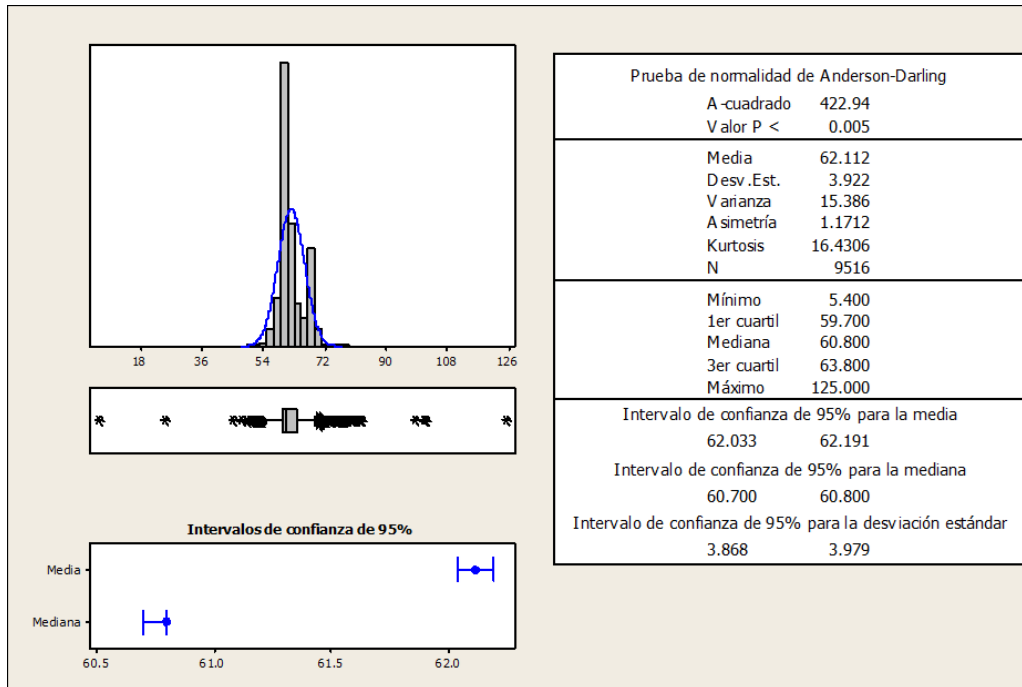


Figura 70. Recuadro de resumen estadístico de la muestra para T_f .
Elaboración propia.

Se tienen dos grupos de mediciones, las mediciones del grupo T_c y del grupo T_f . Para determinar el tamaño de muestra apropiadamente, hay que considerar el grupo que requiere una mayor cantidad de piezas en el tamaño de muestra. Al hacer esto, aseguramos que el tamaño de muestra funcionará para ambos grupos. Entre mayor es la desviación estándar del grupo, mayor será el tamaño de muestra, por lo que se escoge el grupo que tiene la mayor desviación estándar para el cálculo. El grupo T_c tiene la mayor desviación. Es de 18.69, por lo que se selecciona este valor para calcular el tamaño de muestra.

La desviación estándar para la temperatura de apertura es de 18.69 grados y para la temperatura de cierre es de 3.922 grados. Siendo la mayor desviación estándar 18.69, se selecciona este valor para el cálculo del tamaño de muestra a tomar.

El error E corresponde al valor mínimo de diferencia para las muestras, en este caso, en tanto la temperatura se mide en grados centígrados, el valor mínimo a utilizar será de 1°C. Para determinar el tamaño de la muestra se utiliza nuevamente el software estadístico Minitab.

La Figura 71 es el análisis de potencia para la prueba estadística, que nos permite determinar cuántas piezas se requieren como mínimo para realizar el estudio estadístico denominado prueba T de dos muestras. En este caso, el resultado es de 7,342 piezas.

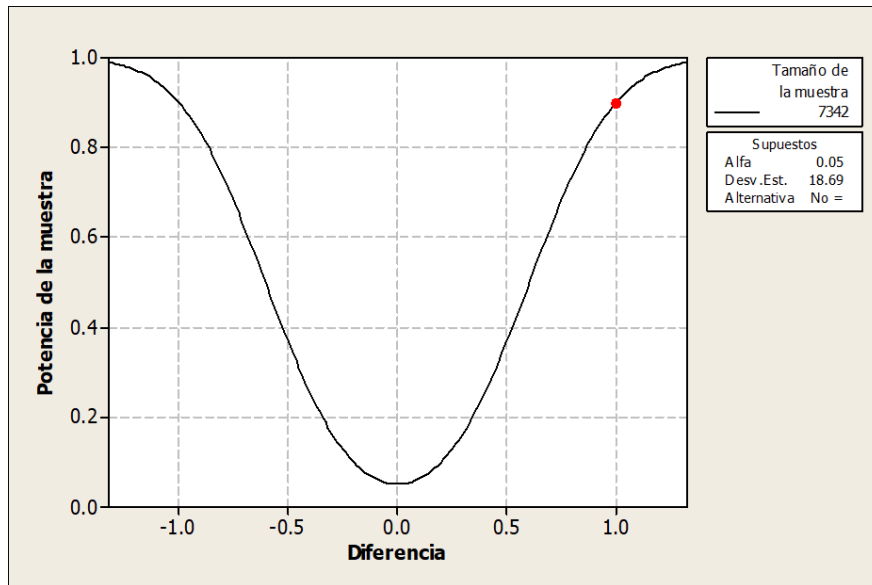


Figura 71. Gráfica de potencia del tamaño de muestra mínimo.
Elaboración propia.

4.2.12. Implementación de delimitador de posición

Una vez definida la cantidad de piezas a analizar, se procede a implementar un delimitador de posición de los servomotores, debido a que el espacio de búsqueda para su ajuste usualmente es muy amplio, lo que puede incrementar el número de ciclos de entrenamiento significativamente, ya que tienen que explorar todo el espacio de ajuste para encontrar el óptimo de formado de la tira bimetalica.

Para evitar ciclos de aprendizaje extremadamente largos, se implementó un algoritmo delimitador del espacio de búsqueda, que funciona de manera similar a un sistema experto (Bulatovic & Stojanovic, 2006) (Moore, Rosenof, & Stanley, 1990) (Vizureanu, Fan, Chen, & Wang, 2010). Se emplea un motor basado en la fórmula de Timoshenko, presentado en la ecuación Ecuación 33 vista anteriormente. Dicha fórmula establece un rango de posiciones válidas para la formación de la tira

bimetálica; dependiendo del tipo de metales que forman el bimetal, el módulo de elasticidad de Young y el coeficiente de expansión térmica, entre otros factores, que ayudan a reducir el espacio de búsqueda.

La Figura 72 muestra una gráfica de error respecto al tiempo de un juego de posiciones en servomotores, que constituyen la propuesta de ajuste de acuerdo con la fórmula de Timoshenko y reglas de experiencia. Estos valores se emplean con la finalidad de reducir el espacio de búsqueda y volverla significativamente más eficiente.

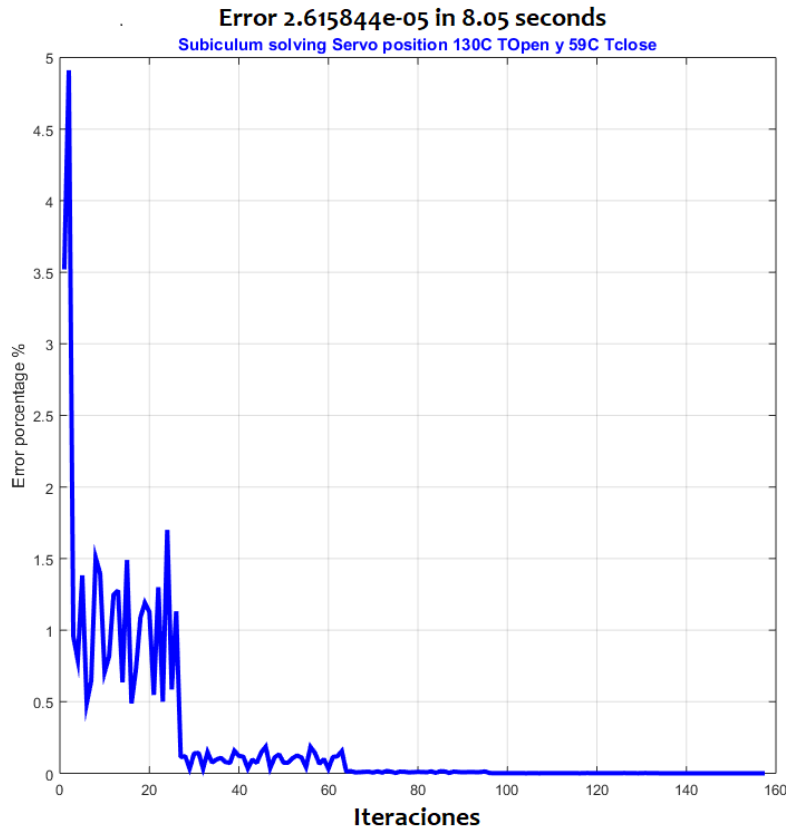


Figura 72. Gráfica de porcentaje de error del algoritmo con respecto al tiempo. Elaboración propia.

4.2.13. Resultados de desempeño del algoritmo híbrido de red neuronal

AHN-ELM

El experimento se realiza en la máquina prototipo para un total de 9 lotes de producción. Se configuran los lotes de producción en la máquina prototipo conforme se van formando los bimetales, puesto que se recibe retroalimentación de la máquina de prueba de temperatura. Cada uno de los lotes fue seleccionado con un diferente tipo de bimetel y radio de curvatura. La razón de la selección es el abarcar diferentes familias de bimetales, las cuales tienen diferentes proporciones de los metales que lo componen. Esto nos sirve para entender cómo afecta las diferentes proporciones en la constitución del bimetel al proceso de aprendizaje de la Red Neuronal AHN-ELM.

Posteriormente, se colectan los resultados de las pruebas y se realiza el análisis de desempeño para el Algoritmo Híbrido AHN-ELM, para lo cual resulta necesario comparar los indicadores de la Tabla 8, que muestran el desempeño de la máquina prototipo de formado de la tira bimetálica. Los campos seleccionados para el análisis son: Lote de bimetel termostático, número de ajustes durante entrenamiento $n_{inicial}$, número de ajustes después de entrenar la red $n_{AHN-ELM}$, eficiencia del proceso original $yield_{original}$ y eficiencia del proceso después de entrenar la red, $yield_{AHN-ELM}$.

Con los resultados obtenidos, y empleando el Minitab, se hace una gráfica de tendencia, la cual se muestra en la Figura 73 y se emplea para verificar si existe visualmente alguna diferencia entre el desempeño de yield cuando se utiliza la máquina prototipo con el sistema original manual (línea negra) y cuando la Red Neuronal AHN-ELM controla la máquina (línea roja). Cabe aclarar que el desempeño mínimo esperado de 90%. En la gráfica puede observarse que, usando el sistema de control original, no se alcanzaba la meta mínima de 90% en yield, ya que los valores están todos ellos debajo del mínimo de 90%, mientras que, cuando se emplea la Red AHN-ELM el 66% de los lotes alcanzan el rango aceptable arriba del 90% de eficiencia. Este desempeño está ligado a las propiedades de los materiales y las geometrías de diseño. En algunos modelos no se logra la

optimización inmediatamente, para alcanzarla, serán necesarios varios ciclos más para que eleven el yield.

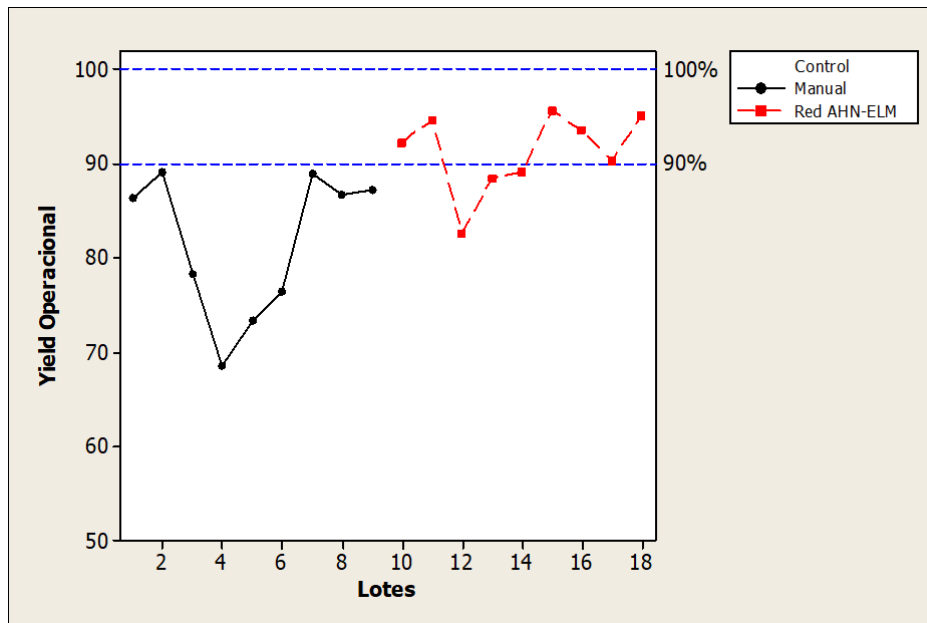


Figura 73. Gráfica de tendencia yield operacional para 9 lotes de prueba aplicando la red AHN-ELM.
Elaboración propia.

Para determinar el promedio, tanto en el sistema original como al aplicar el control con la Red Neuronal AHN-ELM en la máquina prototipo, se realiza una gráfica de intervalos de confianza que incluye los 9 lotes de producción evaluados, la idea es determinar con un 95% de confianza en donde se encontrará la media del desempeño tanto del grupo del sistema original como cuando es controlado por la Red Neuronal AHN-ELM. La gráfica de intervalo de confianza nos ayuda a evaluar si son diferentes los grupos en evaluación, si no hay un traslape entre los intervalos, podemos considerar que son estadísticamente diferentes significativamente, en caso contrario, que se traslapen, significa que no podemos decir que son estadísticamente diferentes debido a algunas mediciones podrían pertenecer a un grupo de control u otro.

La Figura 74 muestra la gráfica con los dos intervalos al 95% de confianza. En ella se puede apreciar que el promedio del control original está entre el 76% y el 88%, mientras que, aplicando el control con la Red Neuronal AHN-ELM desarrollada, se

obtiene un intervalo más cerrado, que va del 88% al 94%. Esta gráfica muestra que el promedio usando la Red Neuronal AHN-ELM es superior al original en 10%, aproximadamente y como no se traslapan los intervalos de confianza podemos decir que existe una diferencia estadística significativa.

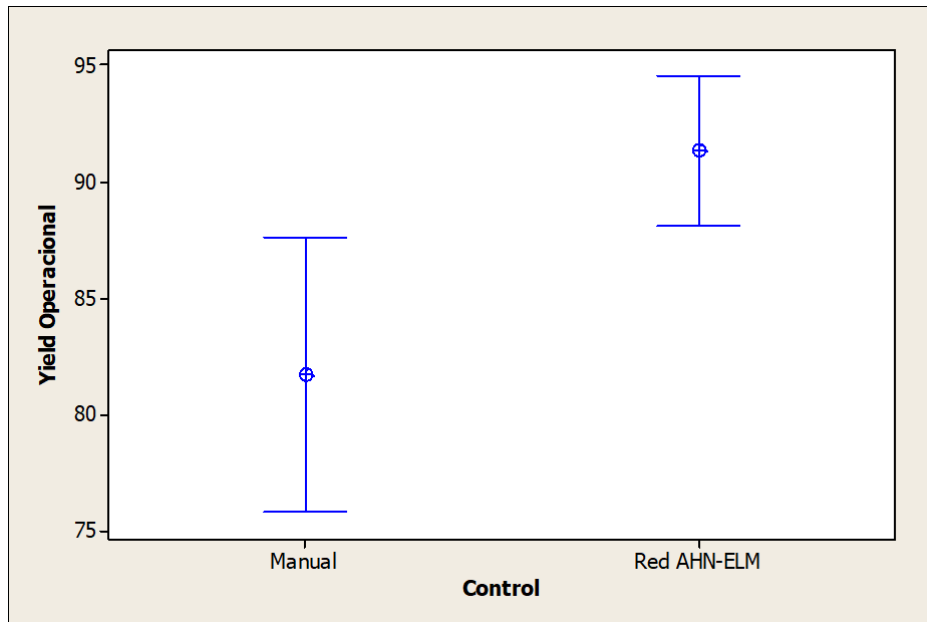


Figura 74. Gráfica de intervalos de confianza que muestra la comparación entre el proceso original y el proceso en el que se aplica la red AHN-ELM. Elaboración propia.

La Tabla 8 muestra los indicadores de desempeño de los lotes en los que se empleó el Algoritmo Híbrido AHN-ELM. Contiene los resultados concretos obtenidos de cada lote de producción. La primera columna indica el número de lote de bimetálico termostático; la segunda, el número de ajustes durante el entrenamiento $n_{Entrenar}$; la tercera corresponde a la eficiencia del proceso original $yield_{original}$ y la última columna a la eficiencia del proceso después de que la Red $yield_{AHN-ELM}$ se entrenó.

Con el fin de demostrar que existe una diferencia estadística significativa entre el desempeño del control original y el control usando la Red Neuronal AHN-ELM, se hace una prueba T de dos muestras (ó 2-sample T, en inglés). La prueba T de 2

Tabla 8. Indicadores del desempeño del algoritmo AHN-ELM.

Lote bimetálico termostático	Número ajustes $n_{inicial}$	Yield proceso $yield_{Original}$	Yield proceso $yield_{AHN-ELM}$
909399	65	86.4%	92.3%
920747	37	89.2%	94.7%
994670	74	78.3%	82.6%
909399	165	68.5%	88.5%
966141	113	73.4%	89.1%
974210	53	76.5%	95.6%
976506	32	88.9%	93.6%
917824	41	86.8%	90.4%
938729	158	87.3%	95.1%

Elaboración propia.

muestras es una prueba de hipótesis para las medias de dos poblaciones, cuya finalidad es determinar si son significativamente diferentes (Ortiz Pinilla & Moreno, 2012). Se utiliza la hipótesis nula, que indica que la diferencia entre las medias de dos poblaciones es igual al valor hipotético ($H_0: \mu_1 - \mu_2 = \mu_0$), y la prueba, comparándola con la hipótesis alternativa, la cual puede ser de cola izquierda ($\mu_1 - \mu_2 < \mu_0$), de cola derecha ($\mu_1 - \mu_2 > \mu_0$) o de dos colas ($\mu_1 - \mu_2 \neq \mu_0$).

Para realizar una prueba T de 2 muestras, las dos poblaciones deben ser independientes; en otras palabras, las observaciones de las 2 muestras no deben tener ninguna relación entre ellas.

Se selecciona la prueba de hipótesis de prueba T de dos muestras de dos colas, donde H_0 en la Ecuación 45 es la hipótesis Nula y H_1 en la Ecuación 46 es la hipótesis Alternativa.

$$H_0: \mu_1 - \mu_2 = 0 \text{ Desempeño igual en ambos controles}$$

Ecuación 45. Prueba de hipótesis nula.

$$H_1: \mu_1 - \mu_2 \neq 0 \text{ Desempeño diferente entre ambos controles}$$

Ecuación 46. Prueba de hipótesis alternativa.

Para aplicar la prueba se utiliza el software Minitab, ya que tiene un módulo dedicado para el cálculo de pruebas T de dos muestras que facilita el análisis. Lo primero que hace el software es determinar la factibilidad de usar la prueba T de dos muestras con base en las características de los datos; busca, entre otras cosas, la existencia de datos anormales. Para esto, hace una prueba de normalidad.

En la Figura 75 se muestra una advertencia de no normalidad de la prueba de Anderson-Darling que hace el software Minitab, implica que no se puede emplear




Prueba t de 2 muestras para la media de Yield Inicia y Yield AHN-EL		
Tarjeta de informe		
Verificar	Estado	Descripción
Datos poco comunes		No hay puntos de datos poco comunes. Los datos poco comunes pueden tener una fuerte influencia sobre los resultados.
Normalidad		Debido a que los tamaños de las muestras son menores que 15, la normalidad puede ser un problema. Si los datos no están distribuidos normalmente, el valor p puede ser inexacto con muestras pequeñas. Dado que la normalidad no se puede verificar de forma confiable con muestras pequeñas, usted debería interpretar los resultados de la prueba con precaución.
Tamaño de la muestra		La muestra es suficiente para detectar una diferencia entre las medias. Debido a que usted ingresó una diferencia de interés, el Informe de diagnóstico provee una evaluación del tamaño de la muestra para esta diferencia. No tiene que preocuparse por el hecho de que la potencia sea baja porque la prueba detectó una diferencia.

Figura 75. El recuadro muestra un resumen de factibilidad de los datos para ejecutar la prueba T de 2 muestras.
Elaboración propia.

la prueba T para dos muestras, por lo tanto, se decide verificar si los datos son realmente no normales haciendo una segunda prueba de normalidad del tipo Ryan-Joiner, y se concluye que los datos son ligeramente no normales. El resultado de la prueba fue 0.071, cuando el valor esperado es $P \leq 0.05$, por lo que se

considera lo suficientemente normal para poder aplicar la prueba T de 2 muestras basado en el resultado de la prueba Ryan-Joiner.

La Figura 76 muestra la prueba adicional de normalidad de los datos basado en el método Ryan-Joiner. Muestra un valor de $P \leq 0.05$, por lo que se considera que los datos son normales. Para su aplicación se utiliza el software estadístico Minitab, con el fin de comparar los dos grupos de resultados de desempeño yield, empleando la prueba T de 2 muestras. Para esto, se compara la columna $yield_{Original}$ con la columna $yield_{AHN-ELM}$ de la Tabla 8.

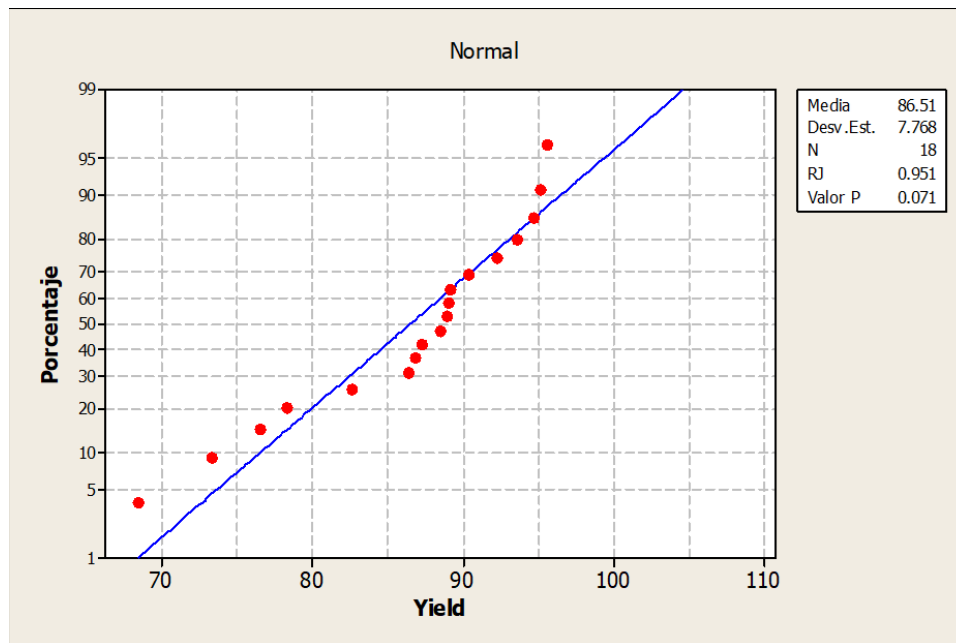


Figura 76. Gráfica del análisis de normalidad Ryan-Joiner basado en correlaciones. Elaboración propia.

El software Minitab también genera el resumen de la verificación de las muestras y del tamaño de muestra para este estudio.

La Figura 78 muestra un recuadro que contiene, en la parte superior, las gráficas de tendencia de yield, el eje de las X corresponde a los lotes de producción y el eje de las Y al yield, mostrando el desempeño de yield en el sistema original, comparado con el que se obtiene con la Red AHN-ELM. Se puede observar como

la media sube. También puede verse gráficamente que no hay datos anormales en la gráfica de tiempo.

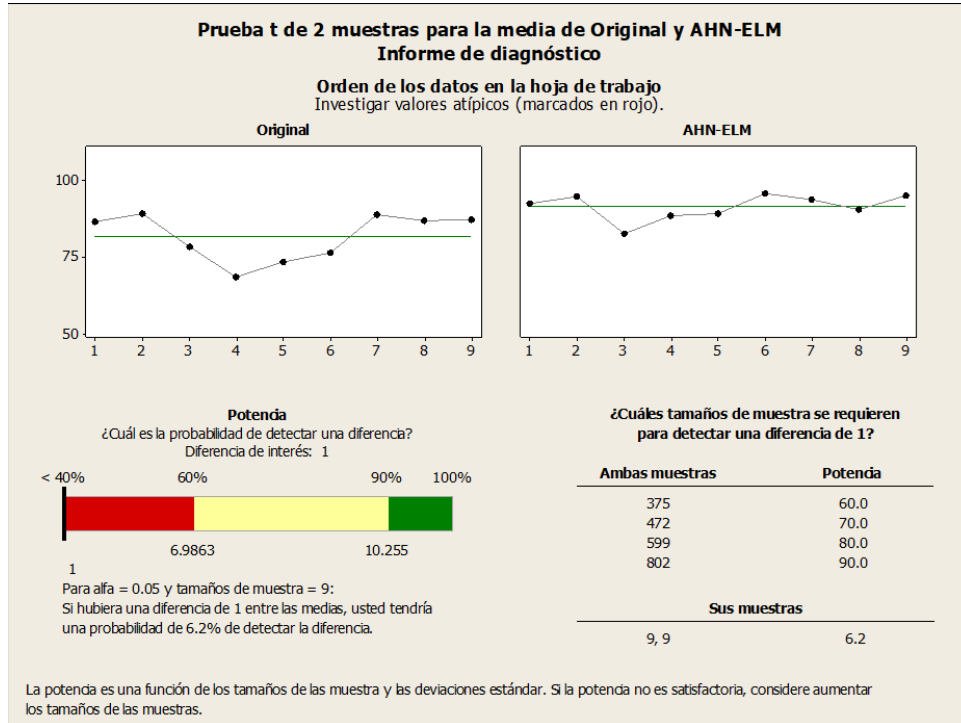


Figura 77. Recuadro de gráfica de tendencia muestra el yield de salida de la máquina prototipo sin y con el nuevo algoritmo híbrido AHN-ELM. Elaboración propia.

El análisis señala que, para poder observar una diferencia de 1% en los grupos, se requiere de un mínimo de 599 muestras, para tener una confianza del 80% de que el análisis podrá ver la diferencia del 1%. Sin embargo, tal como se observó en los gráficos anteriores, la diferencia que se busca es aproximadamente del 10%, por ende, se continúa con la prueba T para dos muestras.

El reporte de resumen de la Figura 78 nos muestra el resultado de la prueba estadística T para 2 muestras. Puede observarse que existe una diferencia estadística significativa entre utilizar o no el Algoritmo Híbrido AHN-ELM. En la parte superior del recuadro, se encuentra una barra de medición que muestra una $P =$

0.006, lo que significa que sí existe una diferencia estadística significativa entre las medias de las dos muestras.

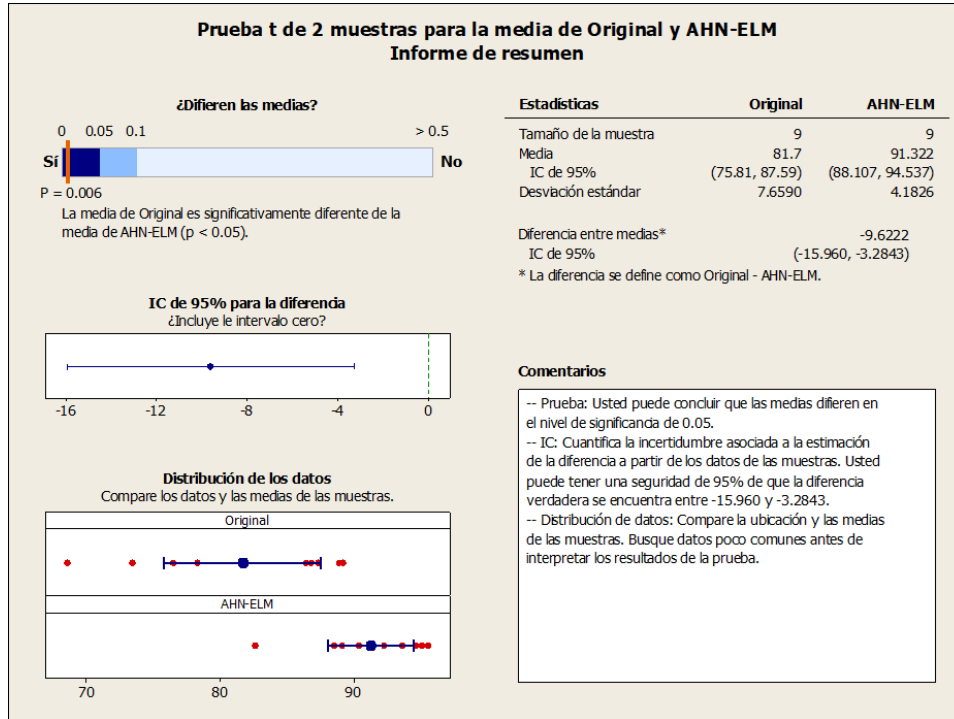


Figura 78. Recuadro de intervalos de confianza de la prueba t de 2 muestras. Elaboración propia.

La gráfica de IC (Intervalo de Confianza) de 95% para la diferencia, muestra una media de 9.6222%, y el intervalo está entre el 3.2843% y el 15.96%, lo que significa que el desempeño de yield usando la RNA AHN-ELM para controlar la máquina prototipo es, en promedio, 9.6222% superior respecto al uso del control original. Resulta importante remarcar aquí que el porcentaje mejorará considerablemente si se corren más ciclos de aprendizaje y que incluso el porcentaje de 9.6222% inicial obtenido se traduce como un ahorro en pruebas, tiempo y materiales muy importante a nivel manufactura.

Los resultados del experimento 2 son congruentes con una mejora de yield del 9.6%, y con ello se demuestra que el sistema puede controlar y aumentar la eficiencia de la máquina prototipo.

4.2.14. Conclusiones de la aplicación del algoritmo híbrido AHN-ELM como control automático en el proceso de formado de bimetales

El impacto de la reducción de desperdicio de la tira bimetálica representa ahorros de hasta un 9.6% en el proceso de producción, ya que, el proceso no solamente evita tirar las piezas de prueba, sino que ayuda a impedir que entre a la línea de producción material que no cumpla con las especificaciones requeridas para la fabricación de moto protectores eléctricos.

El Algoritmo Híbrido AHN-ELM puede ser adaptado también a más procesos industriales y procesos que puedan optimizarse con simulaciones de pruebas para la obtención de valores óptimos de otras índoles, por ende, se continuará evaluando en nuevas aplicaciones en futuros trabajos.

CONCLUSIONES

CONCLUSIONES GENERALES

La Red Hipocámpica AHN-ELM puede considerarse como un Algoritmo Híbrido de la familia del aprendizaje automático autónomo, se desarrolló para la resolución de problemas de decisiones secuenciales en sistemas de control automático.

La Red Hipocámpica Artificial Híbrida AHN-ELM tiene aplicaciones en campos tan diversos como el control automático de maquinaria, la teoría de juegos, economía, mejora del tiempo de respuesta de sensores, entre muchos otros que requieren de la capacidad de autoaprendizaje.

Los problemas de decisiones secuenciales pueden ser abordados mediante múltiples técnicas, sin embargo, una de las características diferenciadoras del aprendizaje empleando la Red Hipocámpica Artificial AHN-ELM desarrollada en la presente investigación, es que, la solución planteada puede estar basada completamente en datos, no necesita estar modelada. Esta característica resulta indispensable en los problemas en los que resulta difícil obtener un modelo que describa de forma precisa la evolución del sistema ante las diferentes acciones posibles.

La Red Hipocámpica Artificial Híbrida PAT-ELM tiene aplicaciones para el monitoreo en medicina, caracterización de materiales en manufactura, monitoreo de condiciones atmosféricas entre muchos otros dependiendo del hardware utilizado. Por otra parte, cada vez es más habitual almacenar de forma sistemática datos relativos a todo tipo de procesos, por lo tanto, el número de aplicaciones potenciales de ambas Redes Hipocámpicas Artificiales es muy amplia, en tanto con sólo esos datos será posible hacer proyecciones y optimizaciones a múltiples procesos de distintas índoles, procesos que resultan fundamentales para la manufactura industrial pero que no se limitan a ella.

CONCLUSIONES DEL EXPERIMENTO 1 PAT-ELM CARACTERIZADOR DE TIRA BIMETÁLICA

En esta investigación proponemos un novedoso sistema de caracterización que puede ser utilizado en aplicaciones industriales, en especial para la fabricación de moto-protectores térmicos.

Dicho sistema está basado en un algoritmo de Máquina de Aprendizaje Extremo ELM y, en el filtrado dinámico por prueba de promedios parciales PAT.

Los resultados obtenidos muestran una variación de 3% entre las mediciones realizadas siguiendo el estándar ASTM y las predicciones obtenidas por nuestro algoritmo.

Estos resultados son relevantes debido a que dicha variabilidad es aceptable para los estándares industriales seguidos en la fabricación de moto-protectores.

Por lo tanto, el Algoritmo Híbrido propuesto, PAT-ELM puede reducir el desperdicio de material termostático usado en el arranque y control del formado de las tiras bimetálicas, que puede llegar a representar al 10% de la producción, mientras que la cantidad de horas hombre dedicadas a los procesos de caracterización se reducirá en un 40%.

El impacto de desperdicio representa ahorros de hasta un 20% en el proceso de producción, ya que no solamente se evita tirar las piezas de prueba, sino que se ayuda a evitar que entre a la línea de producción material que no cumpla con los requisitos para la fabricación de moto-protectores eléctricos.

CONCLUSIONES DEL EXPERIMENTO 2 AHN-ELM EN EL CONTROL DEL PROCESO DE FORMADO

El sistema desarrollado aquí funciona también para controlar procesos de manufactura automáticos que requieren estar constantemente optimizándose, para compensar las variaciones tanto de la materia prima como de las variaciones en los procesos. Los resultados obtenidos muestran que la máquina es capaz de compensar las variaciones hasta con un 3% de error. El tiempo de respuesta del aprendizaje puede variar de 30 a 600 ciclos dependiendo del tipo de tira bimetálica termostática. El impacto de desperdicio representa ahorros de hasta

un 20% en proceso de producción ya que no solamente se evita tirar las piezas de prueba, sino que ayuda a evitar que entre a la línea de producción material que no cumpla los requisitos para la fabricación de moto protectores eléctricos.

Este Algoritmo Híbrido puede ser adaptado otros procesos industriales, por lo tanto, se seguirá evaluando en nuevas aplicaciones.

RECOMENDACIONES

RECOMENDACIONES Y PROYECCIÓN FUTURA

La proyección futura de la presente investigación es amplia y puede enriquecerse. A continuación, se describen algunas posibles mejoras sobre los métodos propuestos para mimetizar las instancias hipocámpicas, y se sugieren algunas posibles direcciones de investigación que pueden ayudar a implementarlas.

Como trabajo futuro se planea hacer una comparación de los Algoritmos Híbridos AHN-ELM y PAT-ELM experimentando con diferentes tipos de Redes Neuronales Artificiales diferentes a las seleccionadas en la presente investigación para verificar la compatibilidad de los algoritmos y buscar mejores desempeños.

Para la instancia hipocámpica Subiculum *SUB* se implementó en esta investigación usando el algoritmo Q-Learning, sin embargo, este algoritmo es uno de muchos tipos de algoritmo de Aprendizaje por Refuerzos. Una de las oportunidades de mejora es relacionada al tiempo de aprendizaje del algoritmo Q-Learning, debido al método se basa en prueba y error. El tiempo de aprendizaje no se ha minimizado. Aunque se detectó esta oportunidad durante la investigación se decidió acotarlo al Q-Learning. Sin embargo, actualmente existen diversos trabajos que combinan las máquinas de aprendizaje extremo con técnicas de regularización, en el contexto del Aprendizaje por Refuerzos, este proceso resulta más complicado debido a que no se dispone de una señal deseada. Aun así, existen algunos trabajos que utilizan regularización por ejemplo el algoritmo LSTD (Kolter & Ng, 2009) y que podrían implementarse en la instancia hipocámpica Subiculum *SUB* para el Algoritmo Híbrido de la Red Neuronal AHN-ELM.

Otra posible ruta para continuar esta investigación es en el área de la auto asociación, la red Hopfield fue seleccionada por su rapidez de aprendizaje, pero puede reemplazarse por alguna otra red auto-asociativa si se tiene suficiente capacidad de cómputo. Debido a que el objetivo es aplicar la Red Neuronal Hipocámpica AHN-ELM al control cognitivo, no se desarrollaron otras posibilidades, pero puede buscarse algún algoritmo auto-asociativo.

Se detectó durante la investigación una posible aplicación de la Red Híbrida AHN-ELM para el control de la aplicación de silicón líquido como sellos en sensores de presión automotrices. Este proceso tiene características muy similares a las del formado de curvatura en bimetales.

Adicionalmente se detectó una aplicación potencial para la Red Híbrida PAT-ELM en la segregación de sensores de presión automotriz que presentan comportamientos anómalos durante su prueba funcional final, por lo que esta sería una aplicación del tipo segregación, que podría abrir un nuevo grupo de aplicaciones adicionales a la caracterización.

Por último, la Red Híbrida AHN-ELM se desarrolla para procesos de control y la red Híbrida PAT-ELM monitoreo cognitivo con el objetivo de aplicarse a los procesos de manufactura, sin embargo, el Algoritmo puede aplicarse a otras áreas como la médica o el manejo autónomo de autos, entre otros.

APORTACIÓN DE LA TESIS

La aportación de esta investigación es la creación de dos nuevos Algoritmos Híbridos: el primero para el control automático cognitivo, que tiene aplicación en procesos industriales prácticos, y el segundo para ayudar al monitoreo cognitivo de un proceso industrial. El primer Algoritmo Híbrido desarrollado es la Red Neuronal AHN-ELM, inspirado en el hipocampo biológico. Es un algoritmo de control, que es capaz de aprender de forma automática sobre los parámetros y variables del entorno, para optimizar un proceso de manufactura y lograr una meta predefinida, basada en una función de recompensa. El segundo Algoritmo Híbrido es la Red Neuronal PAT-ELM, cuya característica es que permite eliminar los datos anormales y guardar sólo la información relevante, para reproducir apropiadamente una caracterización basada en los datos de entrada. Ambos Algoritmos Híbridos fueron probados exitosamente en el formado y caracterización del bimetal que se emplea en la construcción de moto protectores, con lo cual se confirmó su factibilidad práctica para la manufactura industrial.

APORTACIÓN SOCIAL DE LA TESIS

Uno de los principales impactos sociales de los nuevos algoritmos hipocámpicos AHN-ELM y PAT-ELM es que ayudan en las áreas de automatización de manufactura a aumentar la competitividad en el mercado, al ofrecer productos de mejor calidad en menor tiempo, reaccionar de forma más rápida y flexible a los cambios. El aumentar la competitividad en el mercado es de fundamental importancia para asegurar que las fuentes de trabajo de productos críticos como la fabricación de moto-protectores puedan hacer frente a las competencias globales. Esto asegura se puedan mantener en México los empleos y el know-how en este tipo de actividad económica. Otro impacto indirecto de los nuevos algoritmos hipocámpicos AHN-ELM y PAT-ELM es que se incrementa la seguridad de las personas, ya que afecta a los moto protectores eléctricos utilizados principalmente en los refrigeradores y en los aires acondicionados, estos en caso de sobrecargas y picos de tensión que ocurren con cierta regularidad en el país, puede producir incendios, el incremento de calidad de los motor protectores reduce los riesgos de incendio con los usuarios finales.

REFERENCIAS

- Kullkarni, A., & Ugale, S. (2015). Advanced Thermal Overload Protection for High Tension Motors Using Digital Protection Relays. *IEEE International Conference on Computer, Communication and Control (IC4-2015)*.
- Agency, C. (2020). *The world factbook - Field Listing :: GPD -composition, by sector origin*. Agency, Central Intelligence.
- Ahmad, Z. (2006). *Principles of corrosion engineering and corrosion control*. Amsterdam: IChem; Elsevier.
- Albertos Pérez, P., & Sala, A. (2004). *Multivariable control systems: an engineering approach*. London: Springer.
- American Society of Heating. (2014). Refrigerant control devices. En *2014 ASHRAE Handbook - Refrigeration (I-P Edition)* (pág. 112). Knovel.
- Angel, G., & Haritos, G. (2013). An Immediate Formula for the Radius of Curvature of A Bimetallic Strip. *International Journal of Engineering Research*, 8.
- Anireh, V., & Osegi, E. (2017). *HTM-MAT: An online prediction software toolbox based on cortical machine learning algorithm*. Obtenido de <https://hal.archives-ouvertes.fr/hal-01550944>
- Ao, S. (2013). *The 2013 International Conference of Computer Science and Engineering (Vol. 2)*. (I. A. Engineers, Ed.) Hong Kong: IAENG.
- Asad Ali, M. (2018). *Source Code and Datasets - Asad Ali*. Obtenido de <https://sites.google.com/site/asad82/code>
- ASTM, B. (2013). *Test Methods for Flexivity of Thermostat Metals ASTM B106-08*. ASTM International.
- Astrom, K. (01 de 2014). Control: A perspective. *Automatica*, 50(1). doi:10.1016/j.automatica.2013.10.012
- Barto, A. (1994). Reinforcement learning control. *Current Opinion in Neurobiology*, 4(6), 888 - 893. doi:[https://doi.org/10.1016/0959-4388\(94\)90138-4](https://doi.org/10.1016/0959-4388(94)90138-4)
- Bulatovic, M., & Stojanovic, R. (2006). Expert system in function control of maintenance and prediction of machine's failure. *Proceedings of the 10th WSEAS International Conference on COMPUTERS* (págs. 798-804). Vouliagmeni, Athens, Greece: ResearchGate.
- Caliskan, A., & Onat, A. (2017). Characterization - The characterization of bimetal disks used for thermostatic control and over-temperature protection. *The Online Journal of Science and Technology*, 8.

- Camacho, E., & Bordons, C. (2007). *Model Predictive control* (2nd. edition ed.). Springer.
- Camacho, E., Ramirez, D., Limon, D., & Peña, D. (1 de 4 de 2010). Model predictive control techniques for hybrid systems. *Annual Reviews in Control*. doi:10.1016/j.arcontrol.2010.02.002
- Chan, K., Kwong, C., & Tsim, Y. (2010). A genetic programming based fuzzy regression approach to modelling manufacturing processes. *International Journal of Production Research*, 48(7), 1967-1982. doi:10.1080/00207540802644845
- Chong, E., & Żak, S. (2001). *An introduction to optimization* (2nd ed ed.). New York: Wiley.
- Christofides, P. (2013). Distributed model predictive control: A tutorial review and future research directions. *Computers & Chemical Engineering*, 51. doi:10.1016/j.compchemeng.2012.05.011
- Conte, G., Perdon, A., & Vitaioli, G. (2009). A model based control scheme with sampled information. *Mediterranean Conference on Control and Automation*, 0, 1354-1360. doi:10.1109/MED.2009.5164735
- Darby, M. (1 de April de 2012). MPC: Current practice and challenges. *Control Engineering Practice*, 20. doi:10.1016/j.conengprac.2011.12.004
- D'Entremont, J. (1976). An Improved Method for Selecting Thermal Overloads to Provide Half Voltage Protection. *International Compressor Engineering Conference* (pág. 169). Purdue University.
- dos Santos Mignon, A., & de Azevedo da Rocha, R. (2017). An Adaptive Implementation of ϵ -Greedy in Reinforcement Learning. *Procedia Computer Science*, 109, 1146-1151. doi:10.1016/j.procs.2017.05.431
- Dotoli, M. (April de 2016). Advanced control in factory automation: a survey. *International Journal of Production Research*. doi:10.1080/00207543.2016.1173259
- Dutta, S., Bai, Z., Jeong, H., Low, T., & Grover, P. (2018). A Unified Coded Deep Neural Network Training Strategy based on Generalized PolyDot codes. 1585-1589. doi:10.1109/ISIT.2018.8437852
- El-Shal, S., & Morris, A. (5 de 2000). A fuzzy expert system for fault detection in statistical process control of industrial processes. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 30(2), 281-289. doi:10.1109/5326.868449
- Escandell-Montero, P., Martínez-Martínez, J. M., & Martín-Guerrero, J. D. (2014). Least-squares temporal difference learning based on an extreme learning machine. *Neurocomputing*, 37-45. doi:10.1016/j.neucom.2013.11.040

- Estrada, T., Lin, H., & Antsaklis, P. (2006). Model-Based Control with Intermittent Feedback. *Mediterranean Conference on Control and Automation, 0*, págs. 1-6. doi:10.1109/MED.2006.328761
- Fan, S.-K., & Wang, C.-Y. (2008). On-line tuning system of multivariate dEWMA control based on a neural network approach. *International Journal of Production Research, 46*(13), 3459-3484. doi:10.1080/00207540601096932
- Felding, H. (2017). *Implications on ASIC production tests for Fingerprint Cards AB - new requirements and yield consequences*. Sweden: Department of electrical and information technology, Lund University.
- Feng, H. (2018). The application of artificial intelligence in electrical automation control. *Journal of Physics: Conference Series, 1087*, 062008. doi:10.1088/1742-6596/1087/6/062008
- Frolov, A., Húsek, D., Muraviev, I., & Polyakov, P. (2010). Learning and Unlearning in Hopfield-Like Neural Network Performing Boolean Factor Analysis. En *Advances in Machine Learning I: Dedicated to the Memory of Professor Ryszard S. Michalski* (págs. 501–518). Springer. doi:DOI: 10.1007/978-3-642-05177-7_26
- Gawthrop, P., Gollee, H., & Loram, I. (2014). Intermittent Control in Man and Machine. *Event-Based Control and Signal Processing*. doi:10.1201/b19013-16
- Gedzurs, A. (2015). Temperature protection methods of induction motor. *Research for rural development, 258-263*.
- Gluck, M. (2009). *Aprendizaje y memoria: del cerebro al comportamiento*. Mexico: McGraw-Hill Education.
- Hagan, M., Demuth, H., & Beale, M. (2018). *Neural Network Design 2nd edition* (2nd edition ed.). Wrocław.
- Haifei, T., & McNally, F. (2011). *Guidelines for part average testing AEC - Q001 Rev-D*. Automotive Electronics Council. Obtenido de http://www.aecouncil.com/Documents/AEC_Q001_Rev_D.pdf
- Haupt, R., & Haupt, S. (2004). *Practical genetic algorithms* (Vol. 2nd ed). Hoboken, N.J: John Wiley.
- Haykin, S. (2009). *Neural networks and learning machines* (3rd ed ed.). New York: Prentice Hall.
- He, Z., Shao, H., Zhong, X., & Zhao, X. (2020). Ensemble transfer CNNs driven by multi-channel signals for fault diagnosis of rotating machinery cross working conditions. *Knowledge-Based Systems, 207*, 106396. doi:10.1016/j.knosys.2020.106396

- Heemels, W., & Donkers, M. (3 de 2013). Model-based periodic event-triggered control for linear systems. *Automatica*. doi:10.1016/j.automatica.2012.11.025
- Hillar, C., Mehta, R., & Koepsell, K. (2014). A hopfield recurrent neural network trained on natural images performs state-of-the-art image compression. *2014 IEEE International Conference on Image Processing (ICIP)* (págs. 4092-4096). Paris, France: IEEE.
- Hitachi Bimetals. (2008). *Bimetal - Hitachi Bimetals*. Obtenido de Hitachi Bimetals: <http://www.hitachi-metals-neomaterial.co.jp/english/technology/catalog/pdf/bimetal.pdf>
- Huang, G.-B. (2014). An Insight into Extreme Learning Machines: Random Neurons, Random Features and Kernels. *Cognitive Computation*, 6(3), 376-390. doi:10.1007/s12559-014-9255-2
- Huang, G.-B. (2014). An Insight into Extreme Learning Machines: Random Neurons, Random Features and Kernels. *Cognitive Computation*, 6(3), 376-390. doi:10.1007/s12559-014-9255-2
- Huang, G.-B. (2015). What are Extreme Learning Machines? Filling the Gap Between Frank Rosenblatt's Dream and John von Neumann's Puzzle. *Cognitive Computation*, 263-278. doi:10.1007/s12559-015-9333-0
- Huang, G.-B., Liang, N.-Y., Rong, H.-J., & Saratchandran, P. (2005). On-Line Sequential Extreme Learning Machine. *the IASTED International Conference on Computational Intelligence (CI 2005)*, (pág. 6). Calgary, Canada.
- Huang, Y. (2009). Advances in Artificial Neural Networks – Methodological Development and Application. *Algorithms*, 2. doi:10.3390/alg02030973
- Hussain, T., Siniscalchi, S., Lee, C.-C., & Wang, S.-S. (2017). Experimental Study on Extreme Learning Machine Applications for Speech Enhancement. *IEEE Access*, 25542-25554. doi:10.1109/ACCESS.2017.2766675
- Iglesias, M., Lowen, R., & Verschoren, A. (2008). *Foundations of generic optimization* (Mathematical modelling--theory and applications ed.). Dordrecht: Springer.
- Ioannou, P., & Fidan, B. (2006). *Adaptive control tutorial*. Philadelphia, PA: Society for Industrial and Applied Mathematics.
- Jain, L., & Martin, N. (1999). *Fusion of neural networks, fuzzy sets, and genetic algorithms: industrial applications*. Boca Raton: CRC press.
- Jämsä-Jounela, P.-L. (2007). Future Trends in process automation. *IFAC Proceedings Volumes*. 40. ScienceDirect. doi:https://doi.org/10.3182/20070213-3-CU-2913.00003

- Jang, B., Kim, M., Harerimana, G., & Kim, J. (2019). Q-Learning Algorithms: A Comprehensive Classification and Applications. 133653-133667. doi:10.1109/ACCESS.2019.2941229
- Japkowicz, N., & Shah, M. (2011). *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge: Cambridge University Press. doi:978-0-511-92180-3
- Jiayu, L., Zongchou, X., Jiahao, Y., & Gartlib, E. (2020). Analysis of an Automatic Control System based on Linear Controllers and MPC Controller. *2020 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*. doi:10.1109/ICIEAM48468.2020.9111874
- Kahraman, C., Deveci, M., Boltürk, E., & Türk, S. (2020). Fuzzy controlled humanoid robots: A literature review. *Robotics and Autonomous Systems*, *134*, 103643. doi:https://doi.org/10.1016/j.robot.2020.103643
- Kamruzzaman, J., Begg, R., & Sarker, R. (2006). *Artificial neural networks in finance and manufacturing*. Hershey, PA: Idea Group Pub.
- Kanthal AB, a. (2008). *Bimetal - Kanthal Thermostatic bimetal handbook*. Hallstahammar, Sweden: Sandvik.
- Kasun, L., Zhou, H., Huang, G.-B., & Vong, C. (2013). Representational Learning with Extreme Learning Machine for Big Data. *4*.
- Kawamura, K., & Gordon, S. (2006). From Intelligent Control to Cognitive Control. *ISORA 2006 11th International Symposium on Robotics and Applications*. doi:DOI: 10.1109/WAC.2006.376003
- Kawamura, K., Dodd, W., Ratanaswasd, P., & Gutierrez, R. (2005). Development of a Robot with a Sense of Self. (págs. 211-217). Nashville, Tennessee: IEEE. doi:10.1109/CIRA.2005.1554279
- Khatkhate, A., Raj, M., & Mirchandani, P. (2017). A Parametric Approximation for the Radius of Curvature of a Bimetallic Strip. *International Journal of Engineering Research & Technology*, *647-650*. doi:10.17577/IJERTV6IS060317
- Knierim, J., & Neunuebel, J. (2016). Tracking the flow of hippocampal computation: Pattern separation, pattern completion, and attractor dynamics. *Neurobiology of Learning and Memory*, *129*, 38-49. doi:10.1016/j.nlm.2015.10.008
- Kolter, J., & Ng, A. (2009). Regularization and feature selection in least-squares temporal difference learning. *Proceedings of the 26th International Conference on Machine Learning* (págs. 1-8). Montreal, Canada: ACM Press. doi:10.1145/1553374.1553442
- Kostal, P. (2012). The Intelligent Manufacturing Systems. *Advanced science letter*, *19*. doi:10.1166/asl.2013.4816

- Kowadlo, G., Ahmed, A., & Rawlinson, D. (2019). AHA! an 'Artificial Hippocampal Algorithm' for Episodic Machine Learning. *arXiv:1909.10340 [cs, q-bio, stat]*.
- Kumar, S., & Singh, M. (2010). Pattern recall analysis of the Hopfield neural network with a genetic algorithm. *Computers & Mathematics with Applications*, 60(4), 1049-1057. doi:10.1016/j.camwa.2010.03.061
- Kwong, C., & Bai, H. (2005). Fuzzy regression approach to process modelling and optimization of epoxy dispensing. *International Journal of Production Research*, 2359-2375.
- Lee, J. (04 de 06 de 2011). Model predictive control: Review of the three decades of development. *International Journal of Control, Automation and Systems*, 9(3), 415. doi:<https://doi.org/10.1007/s12555-011-0300-6>
- Marinissen, E., Singh, A., Glotter, D., Esposito, M., & M. Carulli Jr, J. (2010). Adapting to adaptive testing. *2010 Design, Automation & Test in Europe Conference & Exhibition*. Dresden, Germany: ResearchGate.
- Martínez Álvarez, R. (2017). Análisis de las máquinas "Sparse Autoencoders" como extractores de características. Cartagena: Universidad Politécnica de Cartagena.
- Mayne, D. (2014). Model predictive control: Recent developments and future promise. *Automatica*, 50(12). doi:<https://doi.org/10.1016/j.automatica.2014.10.128>
- Meireles, M., Almeida, P., & Simoes, M. (2003). A comprehensive review for industrial applicability of artificial neural networks. *Industrial Electronics, IEEE Transactions on*, 50, 585-601. doi:10.1109/TIE.2003.812470
- Mesquita, D., Gomes, J., Rodrigues, L., & Galvao, R. (2015). Pruning Extreme Learning Machines Using the Successive Projections Algorithm. *IEEE Latin America Transactions*, 13(12), 3974-3979.
- Middlesex. (2016). *Middlesex Statistical Process Tools*. Obtenido de Statistical Process Control of Bimetal Disk Manufacturing: <http://www.midsx.com/catalog/automated-material-handling-systems-and-components-for-clean-manufacturing/installation-examples/installation-examples-spc-tools/>
- Miskowicz, M. (2017). *Event-Based Control and Signal Processing*. Boca Raton, FL 33487-2742: CRC Press.
- Mitchell, J. (2014). *Global Industrial automation*. Switzerland: Credit Suisse Global Equity Research.
- Montestruque, L., & Antsaklis, P. (2003). On The Model-Based Control of Networked Systems. *Automatica*, 39, 1837-1843. doi:10.1016/S0005-1098(03)00186-9

- Montestruque, L., Garcia, E., & Antsaklis, P. (2014). *Model-based control of networked systems*. New York: Birkhäuser Basel.
- Moore, R., Rosenof, H., & Stanley, G. (1990). Process control using a real time expert system. *Preprints 11th IFAC World Congress*.
- Morari, M. (2009). *Workshop in celebration of David Clarke's contribution to MPC*. Obtenido de University of Oxford, Oxford.: <http://www2.eng.ox.ac.uk/control/events/slides/Morari.pdf>
- Morato, M., Normey-Rico, J., & Sename, O. (2020). Model predictive control design for linear parameter varying systems: A survey. *Annual Reviews in Control*, 49. doi:10.1016/j.arcontrol.2020.04.016
- Murmu, M. (2007). The impact of prenatal stress and neonatal handling on the neuronal development of the rodent limbic system. University Magdeburg: ResearchGate.
- Ortiz Pinilla, J., & Moreno, E. (2012). ¿Se necesita la prueba de Student para dos muestras independientes asumiendo varianzas iguales? *Comunicaciones en Estadística*, 139-157. doi:10.15332/s2027-3355.2011.0002.05
- Otterlo, M., & Wiering, M. (2012). Reinforcement Learning and Markov Decision Processes. *Reinforcement Learning: State of the Art*, 3-42.
- Ozkan, L., Bombois, X., & Ludlage, J. (2016). Advanced autonomous model-based operation of industrial process systems (Autoprofit): Technological developments and future perspectives. *Annual Reviews in Control*, 42. doi:10.1016/j.arcontrol.2016.09.015
- Patterson, N. (2012). A Robust, Non-Parametric Method to Identify Outliers and Improve Final Yield and Quality. *CS MANTECH Conference* (pág. 3). Boston, Massachusetts, USA: CS MANTECH Organization.
- Pawlowski, A., Guzman, J., Berenguel, M., & Normey-Rico, J. (2015). Event-based GPC for multivariable processes. *2015 International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)* (págs. 1-6). Krakow, Poland: IEEE. doi:10.1109/EBCCSP.2015.7300682
- Pihlaja, D. (2013). Real Time Dynamic Application of Part Average Testing (PAT) at Final Test. *CS MANTECH Conference* (págs. 165 - 168). New Orleans, Louisiana, USA: CS MANTECH Organization.
- Precup, R.-E. (2011). A survey on industrial applications of fuzzy control. *Computers in Industry*, 62(3), 213-226. doi:10.1016/j.compind.2010.10.001
- Protection Schemes for Three Phase Induction Motor. (2017). *International Research Journal of Engineering and Technology (IRJET)*.

- Qin, S., & Badgwell, T. (7 de 2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7). doi:10.1016/S0967-0661(02)00186-7
- Qiu, J., Gao, H., & Ding, S. (2016). Recent Advances on Fuzzy-Model-Based Nonlinear Networked Control Systems: A Survey. *IEEE Transactions on Industrial Electronics*, 63(2), 1207-1217. doi:10.1109/TIE.2015.2504351
- Rahul, M., Narinder, S., & Yaduvir, S. (2011). Genetic Algorithms: Concepts, Design for Optimization of Process Controllers. *Computer and Information Science*. doi:10.5539/cis.v4n2p39
- Rajendran, S., & Balu, K. (2010). Application of Fuzzy Model Predictive Control in Multivariable Control of Distillation Column. *International Journal of Chemical Engineering and Applications*, 1, 38-42.
- Ramya, C., Kavitha, G., & Shreedhara, K. (2011). Recalling of Images using Hopfield Neural Network Model. *National Conference on Computers, Communication and Controls -11 (N4C-11)*. Bangalore: Arvix.
- Ray, M. (2005). Dynamic Parts Average Testing in Real-time. *Southwest Test Workshop*.
- Reddy, R., Bhattacharya, A., & Rishita, G. (2018). Advance Predictions of critical digressions in a noisy industrial process- performance of Extreme Learning Machines versus Artificial Neural Networks. *5th IFAC Conference on Advances in Control and Optimization of Dynamical Systems ACODS 2018*, 98-105. doi:10.1016/j.ifacol.2018.05.017
- Revel, A., Gaussier, P., & Banquet, J. (1999). Taking inspiration from the Hippocampus can help solving robotics problems. (págs. 357-362). ResearchGate.
- Samad, T., & Annaswamy, A. (2011). *The Impact of Control Technology, Overview, Success Stories, and Research Challenges*. IEEE Control Systems Society. Obtenido de <http://ieeecss.org/main/loCT-report>.
- Santalla Peñaloza, Z. (2000). *El sistema de memoria humana: memoria episódica y semántica*. Caracas: Universidad Católica Andrés Bello.
- Saridakis, K., Dentsoras, A., Darzentas, J., Vosinakis, S., & Arnellos, A. (2008). Integration of Computational Intelligence Applications in Engineering Design. *Artificial Intelligence: Theories, Models and Applications* (págs. 276-287). Berlin, Heidelberg: Springer. doi:10.1007/978-3-540-87881-0_25
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85-117. doi:10.1016/j.neunet.2014.09.003
- Sheila, W., Bisson, C., & Duffy, A. P. (2012). Applying a behavioural and operational diagnostic typology of competitive intelligence practice: empirical

- evidence from the SME sector in Turkey. *Journal of Strategic Marketing*, 20(1), 19-33. doi:http://dx.doi.org/10.1080/0965254X.2011.628450
- Shekhar, A., & Sharma, A. (2018). Review of Model Reference Adaptive Control. *2018 International Conference on Information , Communication, Engineering and Technology (ICICET)*, (págs. 1-5). doi:10.1109/ICICET.2018.8533713
- Shivalik. (2012). *Shivalik thermostatic bimetal handbook*. New Delhi, India: Shivalik.
- Sinclair, I. (2001). *Sensors and transducers*. Oxford [England] ; Boston: Elsevier.
- Skogestad, S., & Postlethwaite, I. (2009). *Multivariable feedback control: analysis and design* (2. ed., reprint ed.). Chichester: Wiley.
- Sodhro, A., Pirbhulal, S., & Albuquerque, V. (2019). Artificial Intelligence-Driven Mechanism for Edge Computing-Based Industrial Applications. *IEEE Transactions on Industrial Informatics*, 4235-4243. doi:10.1109/TII.2019.2902878
- Souza, D., Neto, A., & Guedes, L. (2014). BR-PlantExpert: A rule-based expert system for operation support in industrial processes. *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)* (págs. 1-4). IEEE. doi:10.1109/ETFA.2014.7005324
- Stewart, D., Cheraghi, S., & Malzahn, D. (2004). Fuzzy defect avoidance system (FDAS) for product defect control. *International Journal of Production Research*, 3159-3182. doi:10.1080/00207540410001696032
- Sun, Y., & El-Farra, N. (13 de 2 de 2012). Resource aware quasi-decentralized control of networked process systems over wireless sensor networks. *Chemical Engineering Science - CHEM ENG SCI*, 69. doi:10.1016/j.ces.2011.10.010
- Sutton, R., & Barto, A. (2018). *Reinforcement learning: an introduction* (Second edition ed.). Cambridge, Massachusetts: The MIT Press.
- Sutton, R., Szepesvári, C., & Maei, H. (2008). A Convergent O(n) Temporal-difference Algorithm for Off-policy Learning with Linear Function Approximation. *Advances in Neural Information Processing Systems*, (págs. 1609-1616).
- Szandata, T. (2015). Comparison of Different Learning Algorithms for Pattern Recognition with Hopfield's Neural Network. *Procedia Computer Science*, 68-75.
- Tao, G. (2014). Multivariable adaptive control: A survey. *Automatica*, 50. doi:10.1016/j.automatica.2014.10.015
- Tebbutt, C. (1994). *Expert Aided Control System Design*. London: Springer London.

- Tenorth, M., Jain, D., & Beetz, M. (2010). Knowledge Processing for Cognitive Robots. *KI - Künstliche Intelligenz*, 233-240. doi:10.1007/s13218-010-0044-0
- Tetko, & Birukou. (2019). *Artificial Neural Networks and Machine Learning - ICANN 2019: Image Processing*.
- Tetko, I., Kůrková, V., Karpov, P., & Theis, F. (2019). *Artificial Neural Networks and Machine Learning - ICANN 2019: Deep Learning (Vol. PartII)*. Munich, Germany.
- Treves, A., & Rolls, E. (1994). Computational analysis of the role of the hippocampus in memory. *Hippocampus*, 374-391. doi:10.1002/hipo.450040319
- Tsitsiklis, J., & Van Roy, B. (1997). An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5), 674-690. doi:10.1109/9.580874
- Tsitsiklis, J., & Van Roy, B. (1997). An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5), 674-690. doi:10.1109/9.580874
- van Heeswijk, M., & He, H. (2015). *Advances in Extreme Learning Machines*. Helsinki, Finland.
- Vizureanu, P., Fan, X., Chen, X., & Wang, Y. (2010). Expert System for Sintering Process Control. En *Expert Systems*. InTech.
- Wang, A., Liu, L., Qiu, J., & Feng, G. (2019). Event-Triggered Robust Adaptive Fuzzy Control for a Class of Nonlinear Systems. *IEEE Transactions on Fuzzy Systems*, 27(8), 1648-1658. doi:10.1109/TFUZZ.2018.2886158
- Weisz, V., & Argibay, P. (2012). Neurogenesis interferes with the retrieval of remote memories: Forgetting in neurocomputational terms. *Cognition*, 13 - 25.
- Wickeder Group. (2010). *Bimetal - EMS Bimetal_Desingers_Guide - Solutions, Engineered Material*. Attleboro, MA: Wickeder Group.
- Xu, S., & Wang, J. (2016). A fast incremental extreme learning machine algorithm for data streams classification. *Expert Systems with Applications*, 65, 332-344. doi:10.1016/j.eswa.2016.08.052
- Xu, S., & Wang, J. (2016). A fast incremental extreme learning machine algorithm for data streams classification. *Expert Systems with Applications*, 65, 332-344. doi:10.1016/j.eswa.2016.08.052
- Yang, L. (2020). Application of Artificial Intelligence in Electrical Automation Control. *Procedia Computer Science*, 166, 292 - 295. doi:https://doi.org/10.1016/j.procs.2020.02.097
- Yin, J., Li, G., & Hu, J. (2014). A Modular Prediction Mechanism Based on Sequential Extreme Learning Machine with Application to Real-Time Tidal Prediction.

Extreme Learning Machines 2013: Algorithms and Applications. Adaptation, Learning, and Optimization, 16.

- Yin, X., Niu, Z., He, Z., Li, Z., & Lee, D. (2020). An integrated computational intelligence technique based operating parameters optimization scheme for quality improvement oriented process-manufacturing system. *Computers & Industrial Engineering*, 140, 106284. doi:10.1016/j.cie.2020.106284
- Zahabi, M., & Kaber, D. (2019). A fuzzy system hazard analysis approach for human-in-the-loop systems. *Safety Science*, 120, 922-931. doi:10.1016/j.ssci.2019.08.029
- Zammit, A., Ezzati, A., Zimmerman, M., & Lipton, R. (2017). Roles of hippocampal subfields in verbal and visual episodic memory. *Behavioural Brain Research*, 317, 157-162. doi:10.1016/j.bbr.2016.09.038
- Zang, J., & Feng, L. (2008). Modified Timoshenko formula for bending of ultrathin strained bilayer films. *Applied Physics Letters*, 021905. doi:10.1063/1.2828043
- Zelenka, M. (2016). Characterization - Calculation and measurement of thermostatic bimetal deflection in molded case circuit breaker., (págs. 467-471).
- Zhang, J., Yin, J., Zhang, Q., & Shi, J. (2017). Robust sound event classification with bilinear multi-column ELM-AE and two-stage ensemble learning. *EURASIP Journal on Audio, Speech, and Music Processing*. doi:10.1186/s13636-017-0109-1
- Zhang, T., Zeng, Y., Zhao, D., & Wang, L. (2016). HMSNN: Hippocampus inspired Memory Spiking Neural Network. En IEEE (Ed.), *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (págs. 002301-002306). IEEE. doi:10.1109/SMC.2016.7844581
- Zhang, T., Zeng, Y., Zhao, D., & Wang, L. (2016). HMSNN: Hippocampus inspired Memory Spiking Neural Network. *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (págs. 002301-002306). IEEE. doi:10.1109/SMC.2016.7844581
- Zhou, M., Li, H.-X., & Weijnen, M. (2015). Advances and Challenges on Intelligent Learning in Control Systems. En *Contemporary Issues in Systems Science and Engineering* (págs. 241-264). doi:DOI: 10.1002/9781119036821.ch6

ANEXOS

ANEXO 1

ANEXO 1 - Pseudo-código del Algoritmo Híbrido AHN-ELM (Artificial Hippocampical Network based on Extreme Learning Machine)

El algoritmo que se presenta en el anexo 1 es la versión de pseudo código del algoritmo de control cognitivo denominada **Red Hipocámpica Artificial basada en Máquinas de Aprendizaje Extremo** ó red **AHN-ELM** por sus siglas en inglés.

ALGORITMO ANEXO 1 - Red Neuronal AHN-ELM

- 1: carga configuración inicial AHNELM \leftarrow *configuracion*
 \dagger
 ————{ Instancia hipocámpica baja Dentate Gyrus }———
 \dagger
- 2: se inicializa instancia hipocámpica baja **DENTATE GYRUS** (DG_a)
estrategia: el algoritmo descartará los patrones de estímulos sensoriales anormales y solo dejará estímulos los estado estable $x_{PAT}(1), \dots, x_{PAT}(p)$
salida: estímulos de estado estable $x_{PAT}(1), \dots, x_{PAT}(p)$
- 3: carga número de δ para definir $\bar{x}_{robusta}$ y el número de patrones iniciales ϕ , $p = 1$, tamaño del lote n_{lote} , número de lotes L
- 4: Paso 1: se cargan vectores de estímulos $x_{ec2}(1), \dots, x_{ec2}(n)$ de memoria sensorial que incluyen posibles anomalías
- 5: Paso 2: se calcula \bar{x} y δ de los ϕ patrones iniciales.
- 6: **repeat**
- 7: **for** cada vector de estímulo $x_{ec2}(p)$ **do**
- 8: lee $x_{ec2}(p)$
- 9: calcula nueva $\bar{x}_{robusta} = Q_{1,3} - \frac{IQR}{2}$
- 10: calcula nuevo $PAT_{limiteinferior} = Q_1 - \frac{(x_i - 0,675)}{1,35}(Q_3 - Q_1)$
- 11: calcula nuevo $PAT_{limitesuperior} = Q_3 + \frac{(x_i - 0,675)}{1,35}(Q_3 - Q_1)$

12: se evalúa $x_{ec2}(p)$

$$x_{PAT}(p) = \begin{cases} x_{ec2}(p) < PAT_{limiteinferior} & \rightarrow delete \\ x_{ec2}(p) > PAT_{limitesuperior} & \rightarrow delete \\ \rho < x_{ec2}(p) < \vartheta & x_{PAT}(p) = x_{ec2}(p) \end{cases}$$

13: lee siguiente elemento x_{ec2}

14: **end for**

15: **until** se completan L lotes evaluados

16: **return** $x_{PAT}(p)$

‡

—————{ Instancia hipocámpica alta Dentate Gyrus }—————

‡

17: se inicializa instancia hipocámpica alta **DENTATE GYRUS** (DG_b)

estrategia: el algoritmo transformará los patrones de estímulos sensoriales de estado estable $x_{PAT}(p)$ a una representación dispersa estable W_{PAT} que permite separar patrones similares

salida: patrones dispersos estables $W_{PAT}(1), \dots, W_{PAT}(n)$

18: Los datos $\{W_{PAT}\} = \{PAT(p)_i\}_{i+1}^N$, número de neuronas i y el coeficiente de penalización C y se espera la salida de la categorización por clusters

19: **inicializa DG** Se obtienen los datos de entrenamiento

20: **repeat**

21: Paso 1: se generan pesos y sesgos ortogonales las neuronas ocultas n_h

22: Paso 2: Calcular los pesos de salida β

23: **if** $h_h \leq N$ **then**

24:
$$\beta^* = \left(H^T H + \frac{I_{h_h}}{C} \right)^{-1} H^T x_{PAT}$$

25: **else if** Calcular los pesos de salida β **then**

26:
$$\beta^* = H^T \left(H H^T + \frac{I_N}{C} \right)^{-1} x_{PAT}$$

27: **end if**

28: Paso 3: $N_{new} = x_{PAT} \beta^T$.

29: Paso 4: Trate cada fila $(x_{PAT})_{new}$ como un punto y agrupe los N puntos en K clusters usando algoritmo k-means.

30: **until** número de patrones faltantes de evaluar del lote = 0

```

31: return  $W_{PAT}$ 
    ‡
    —————{ Instancia Cornus Ammonis 3 modo aprendizaje( $CA3_a$ ) }—————
    —————
    ‡
32: se inicializa instancia hipocámpica CORNUS AMMONIS 3 ( $CA3_a$ )
    modo Aprendizaje
    Estrategia: calcular la matriz de pesos  $Y_P$  para las memoria episódicas
    fundamentales  $P$   $W_{PAT}$ .
    Salida: Se calcula la matriz de pesos ( $W$ ) entre las neuronas, los pesos
    de la matriz son simétricos, la diagonal principal =0.
33: Memoria episódica fundamental ( $Y_1, Y_1, \dots, Y_P$ ), la longitud de todos los
    vectores es igual ( $t_{th}$ )
34: if  $i \neq j$  then
35:    $W = \sum_{i=1}^P Y_P * Y_P^T - P * I$ 
36: else if  $W_{ij} = 0$  then
37:   termina entrenamiento
38: end if
39: return  $W$ 
    ‡
    —————{ Instancia Cornus Ammonis 3 modo Prueba }—————
    —————
    ‡
40: se inicializa instancia hipocámpica CORNUS AMMONIS 3 ( $CA3_p$ )
    modo Prueba
    Estrategia: La red recordará memorias episódicas fundamentales ( $Y_P$ )
    cuando se presente una entrada inestable ( $X_P$ ) o  $X$  de prueba.
    Salida: Patrones de estado estable  $X_1, X_1, \dots, X_P$ .
41: Patrones de estado inestable ( $X_1, X_1, \dots, X_P$ ), la longitud de los vectores
    ( $X$ ) son iguales que la longitud de los recuerdos episódicos fundamentales
    ( $Y$ ).
42: if  $fX_P = Y_P$  then
43:    $X_k = sign((W X_k) - \beta)$ 
44: else if versión incompleta o corrupta (prueba) then
45:
46: end if
47:  $Y(0) = sign((W X_k) - \beta)$ 

```

```

48: repeat
49:    $Y(i_{tr}+ = \text{sing}((WX_k) - \beta))$ 
50: until Mientras el estado del vector no cambie
    ‡
    _____{ Instancia hipocámpica Cornus Ammonis 1 (CA1) }_____
    ‡
51: se inicializa instancia hipocámpica CORNUS AMMONIS 1 (CA1)
Estrategia: La red recordará memorias episódicas junto con memoria de
trabajo.
Salida: Patrones completos del recuerdo episodico mas la respuesta de
memoria de trabajo inicial almacenada.
52: Patrones de estado inestable  $(X_1, X_1, \dots, X_P)$ , la longitud de los vectores
 $(X)$  son iguales que la longitud de los recuerdos episódicos fundamentales
 $(Y)$ .
53: Los Datos  $\{X_i, Y_i\}$ ,  $x_i \in R^d$  y  $y_i \in R^q, i = 1, \dots, N$ , número de neuronas
 $i$ , número de nodos  $i$ , número de capas ocultas  $N$ , función de activación
 $g(x)$  y se espera una salida  $\beta^{k+1}$ 
54: inicializa CA3, donde los datos iniciales son:  $\{X_i, Y_i\}, x_i \in R^d$  y  $y_i \in$ 
 $R^q, i = 1, \dots, N_0$  con  $N_0 \in N$ 
55: for  $m = 1 \rightarrow p$  do
56:    $H_0^m = g\left(\left(\beta_m^{(0)}\right)^T H_0^{m-1}\right)$ 
57:   Donde  $\beta_m^{(0)} = \left(H^T H + \frac{I_{h_h}}{C}\right)^{-1} H^T X$ 
58: end for
59:  $\beta^{(0)} = P_0 H_0^T Y^0$ 
60: donde  $H_0 = H_0^P$ 
61: Aprendizaje secuencial:
62: for  $k = 0 \rightarrow K$  do
63:   for  $m = 1 \rightarrow \text{top}$  do
64:      $H_{k+1}^m = g\left(\left(\beta_m^{(k+1)}\right)^T H_{k+1}^{m-1}\right)$ 
65:     Donde  $\beta_m^{(k+1)} = H^T \left(H H^T + \frac{I_N}{C}\right)^{-1} X$ 
66:   end for

```

67: $\beta^{(k+1)} = \beta^{(k)} + P_{k+1}H^T (Y_{k+1} - H_{k+1}\beta^{(k)})$

68: Donde $H_{k+1} = H_{k+1}^P$

69: **end for**

‡

—————{ Instancia hipocámpica SUBICULUM (*SUB*) }—————

‡

70: se inicializa instancia hipocámpica **SUBICULUM** (*SUB*)

Estrategia: La red recibe las primeras impresiones episódicas almacenadas en memoria y las usa como configuración inicial del algoritmo de aprendizaje por refuerzo, si el resultado cumple con la función de $error > \epsilon$ no se ejecuta, pero si es menor se inicial el aprendizaje por refuerzos.

Salida: Patrones completos de actuadores.

71: Factor de descuento γ , secuencia de aprendizaje α_k , Secuencia de exploración ϵ_k

72: Inicializar estimación de función Q arbitrariamente, ejemplo $Q_0 \leftarrow 0$

73: **repeat**

74: medir estado inicial s_0

75: **for** Cada paso del episodio **do**

76:

$$a_k = \begin{cases} a \in \operatorname{argmax}_a Q - k(s_k, a) & \text{con probabilidad } 1 - \epsilon_k \\ \text{acción uniformemente aleatoria en } A & \text{con probabilidad } \epsilon_k \end{cases}$$

77: aplicar a_k , observar el estado siguiente s_{k+1} y la recompensa r_{k+1}

78: $Q_{k+1}(s_k, a_k) = Q_k(s_k, a_k) + \alpha_k [r_{k+1} + \gamma \max_{a'} Q_k(s_{k+1}, a') - Q_k(s_k, a_k)]$

79: **end for**

80: **until** Cumplir condiciones de convergencia

81: **return** π^*

ANEXO 2

ANEXO 2 - Pseudo-código del Algoritmo Híbrido PAT-ELM (Part Average Testing based on Extreme Learning Machine)

El algoritmo que se presenta en el anexo 2 es la versión de pseudo código del algoritmo de monitoreo cognitivo denominado **Red de Prueba de Promedios Parciales PAT basada en Máquinas de Aprendizaje Extremo ó PAT-ELM** por sus siglas en inglés.

ALGORITMO ANEXO 2 - Red Neuronal PAT-ELM

- 1: carga configuración inicial PATELM \leftarrow configuración
 \dagger
 ————{ Instancia Hipocámpica Baja Dentate Gyrus }———
 \dagger
- 2: se inicializa Instancia Hipocámpica Baja **DENTATE GYRUS** (DG_a)
estrategia: el algoritmo descartará los patrones de estímulos sensoriales anormales y solo dejará estímulos los estado estable $x_{PAT}(1), \dots, x_{PAT}(p)$
salida: estímulos de estado estable $x_{PAT}(1), \dots, x_{PAT}(p)$
- 3: carga número de δ para definir $\bar{x}_{robusta}$ y el número de patrones iniciales ϕ , $p = 1$, tamaño del lote n_{lote} , número de lotes L
- 4: Paso 1: se cargan vectores de estímulos $x_{ec2}(1), \dots, x_{ec2}(n)$ de memoria sensorial que incluyen posibles anomalías
- 5: Paso 2: se calcula \bar{x} y δ de los ϕ patrones iniciales.
- 6: **repeat**
- 7: **for** cada vector de estímulo $x_{ec2}(p)$ **do**
- 8: lee $x_{ec2}(p)$
- 9: calcula nueva $\bar{x}_{robusta} = Q_{1,3} - \frac{IQR}{2}$
- 10: calcula nuevo $PAT_{limiteinferior} = Q_1 - \frac{(x_i - 0,675)}{1,35}(Q_3 - Q_1)$
- 11: calcula nuevo $PAT_{limitesuperior} = Q_3 + \frac{(x_i - 0,675)}{1,35}(Q_3 - Q_1)$

12: se evalúa $x_{ec2}(p)$

$$x_{PAT}(p) = \begin{cases} x_{ec2}(p) < PAT_{limiteinferior} & \rightarrow delete \\ x_{ec2}(p) > PAT_{limitesuperior} & \rightarrow delete \\ \rho < x_{ec2}(p) < \vartheta & x_{PAT}(p) = x_{ec2}(p) \end{cases}$$

13: lee siguiente elemento x_{ec2}

14: **end for**

15: **until** se completan L lotes evaluados

16: **return** $x_{PAT}(p)$

‡

———{ Instancia hipocámpica Dentate Gyrus variante Monitoreo }———

‡

estrategia: el algoritmo transformará los patrones de estímulos sensoriales de estado estable $x_{PAT}(p)$ a una representación dispersa estable W_{PAT} que permite separar patrones similares

salida: patrones aprendidos estables $Y_{PAT}(1), \dots, Y_{PAT}(n)$

17: carga número de δ para definir $\bar{x}_{robusta}$ y el número de patrones iniciales ϕ , $p = 1$, tamaño del lote n_{lote} , numero de lotes L

18: Se inicializa la ELM, número de nodos, número de capas ocultas L , función de activación $g(x)$.

19: cargar datos $(N.x, N.y)$

20: Cargar los datos de entrenamiento. $N = \{x_i, t_i \mid x_i \in R^n, t_i \in R^m, i = 1, \dots, N\}$

21: Define el tipo de función del ELM, categorización o regresión.

22: Genera la Matriz y se asignan aleatoriamente los pesos a los vectores de entrada w_i y bias b_i .

23: Calcula la Matriz de la capa oculta de salida H en base a función de activación.

24: Calcula los pesos de las salidas $\beta : \beta = H^\dagger$ Donde H^\dagger es la inversa generalizada de Moore-Penrose de H .

25: **return** $x_{PATELM}(p)$

26: **end**
