



**CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS
AVANZADOS DEL INSTITUTO POLITÉCNICO NACIONAL**

UNIDAD ZACATENCO

**DEPARTAMENTO DE
CONTROL AUTOMÁTICO**

**“Identificación y Control de Sistemas Dinámicos No Lineales
por medio de un Multi-Modelo Neuronal-Difuso Complejo”**

T E S I S

Que presenta

MARIO CESAR MAYA RODRIGUEZ

**Para obtener el grado de
MAESTRO EN CIENCIAS**

**EN LA ESPECIALIDAD DE
CONTROL AUTOMATICO**

**DIRECTOR DE TESIS
Dr. Ieroham Solomon Barouh**

Ciudad de México

DICIEMBRE, 2017

Agradecimientos

Agradezco a mis padres, David y Delia, por esforzarse para brindarme la oportunidad de estudiar y así llegar a este punto de mi vida, gracias.

También extendiendo este agradecimiento, a mis familiares que en algún momento de este camino me apoyaron, Mario, Itzel, Raúl, Elidía, Rosa, Alberto, Mario F. y Manuel.

Al Dr. Ieroham S. Barouh, que, bajo su dirección y apoyo, fue posible realizar este trabajo.

Para aquellos amigos que estuvieron mi lado: Jesus, Mario, Pino, Luis, J. Luis y José.

Al depto. de Control Automático por darme una educación de excelencia y calidad.

Al CONACyT, por su apoyo económico para la obtención de grado y la presentación de un artículo producto de este trabajo.

Al Dr. Moisés Bonilla Estrada y al Dr. Carlos R. Mariaca Gaspar, por sus observaciones y correcciones para mejorar la calidad de este trabajo.

Resumen

En este trabajo, se propone utilizar la poderosa herramienta de un Multi-Modelo Difuso-Neural, cuya estructura consiste en una metodología jerárquica compuesta por: Fuzzificador, reglas de Takagi-Sugeno basadas en reglas difusas y Defuzzificador. Se combina con una topología de redes neuronales recurrentes de valor complejo. La topología está entrenada con dos algoritmos de aprendizaje recursivo, Back-Propagation y Levenberg-Marquardt, ambos expresados en un dominio complejo. El objetivo principal de la identificación del sistema de plantas oscilatorias no lineales es emitir estados y parámetros para el control.

La identificación del sistema usó solo dos funciones de membresía (positiva y negativa) con una pequeña superposición en el vecindario de cero. En este caso, un seno y onda cuadrada se utilizan como entrada. Los resultados de la simulación muestran la comparación entre la salida de la planta oscilatoria (un modelo de robot de articulación flexible de un grado de libertad) y la salida del algoritmo de identificación del sistema. Además, el algoritmo de Levenberg-Marquardt presenta algunas ventajas sobre el algoritmo de propagación hacia atrás para las dos señales de entrada. La función de costo minimizado de Levenberg-Marquardt (Instantáneo Significa Error Cuadrado) es más rápido que la Propagación Reversa. Finalmente, los resultados de la simulación comparativa confirman una mejor calidad del Multi-Modelo Difuso-Neural Jerárquico expresado en una identificación compleja del sistema de dominio sobre la simple identificación del sistema compuesta únicamente por la Red Neuronal Recurrente de Valor Complejo.

El sistema de control lo conforman las siguientes estructuras: modelo inverso de control directo, modelo inverso de control directo con i-term y control de modo deslizante con i-term. Estas estructuras se prueban bajo los dos algoritmos de aprendizaje, Back-Propagation y Levenberg-Marquardt; y los resultados obtenidos son buenos para las tres formas de control.

Abstract

In this work, it is proposed to use the powerful tool of a Fuzzy-Neural Multi-Model, whose structure consists in a Hierarchical methodology composed by: Fuzzyfier, Fuzzy Rule-Based Inference Takagi-Sugeno rules and Defuzzyfier. It is combined with a Complex-Valued Recurrent Neural Network topology. The topology is trained with two recursive learning algorithms, the Back-Propagation and the Levenberg-Marquardt, both expressed in a complex domain. The main objective of system identification of nonlinear oscillatory plants is to issue states and parameters for control.

The system identification used only two membership functions (positive and negative) with a small overlap in the neighborhood of zero. In this case, a sinus and square wave are used as input. The simulation results show the comparison between the output of the oscillatory plant (one degree of freedom flexible-joint robot model) and the output of the system identification algorithm. Moreover, the Levenberg-Marquardt algorithm presents some advantages over the Back-Propagation algorithm for the two input signals. The Levenberg-Marquardt minimized cost function (instantaneous Means Squared Error) it is faster than the Back-Propagation. Finally, the comparative simulation results confirm better quality of the Hierarchical Fuzzy-Neural Multi-Model expressed in a complex domain system identification over the simple system identification composed by only Complex-Valued Recurrent Neural Network.

The control system is made by the following structures: inverse model of direct control, inverse model of direct control with i-term and sliding-mode control with i-term. These structures are proved under the two learning algorithms, Back-Propagation and the Levenberg-Marquardt; and the results has obtained are good for the three ways of control.

Índice General

Agradecimientos	iii
Resumen	v
Abstract	vi
Índice de Figuras	x
Índice de Tablas	xvi
1. INTRODUCCIÓN.....	17
1.1 Planteamiento del Problema y Objetivos	19
1.2 Organización de la Tesis.....	20
2. REDES NEURONALES EN EL DOMINIO COMPLEJO.....	21
2.1 Modelo de una Neurona	22
2.2 Funciones de Activación en el Dominio Real	23
2.3 Redes Neuronales en el Dominio Complejo.....	25
2.4 Funciones de Activación en el Dominio Complejo	27
2.5 Aprendizaje en el Dominio Complejo	27
2.6 Redes Neuronales Recurrentes Entrenables.....	28
3. SISTEMAS DIFUSOS	29
3.1 Lógica Difusa.....	29
3.2 Conjuntos Difusos	30
3.3 Variables Lingüísticas.....	31
3.4 Principio de Extensión	32
3.5 Operaciones Difusas.....	33
3.5.1 Unión.....	33
3.5.2 Intersección	34

3.5.3	Complemento.....	35
3.5.4	Propiedades de Operaciones Difusas para Sub-conjuntos.....	35
3.6	Métodos Básicos de Inferencia	35
3.6.1	Modus Ponens Generalizado	36
3.6.2	Modo Tollens Generalizado	36
3.7	Esquema Genérico de Lógica Difusa	36
3.7.1	Fuzzificador	37
3.7.2	Reglas y Modelo Takagi-Sugeno	37
3.7.3	Defuzzificador	38
4.	MULTI-MODELO DIFUSO-NEURONAL COMPLEJO.....	39
4.1	Fuzzificación en FNMM.....	40
4.2	Inferencia Difusa en FNMM	41
4.3	Topología de CVRNN.....	42
4.4	Defuzzificación de FNMM.....	45
4.5	Descripción de HFNMM	46
4.6	Algoritmos de Aprendizaje.....	48
4.6.1	Back-Propagation.....	48
4.6.2	Levenberg-Marquardt.....	49
4.7	Convergencia y Estabilidad	50
5.	IDENTIFICACIÓN Y CONTROL CON HFNMM	52
5.1	Descripción de la Planta de Primer Orden.....	52
5.2	Descripción de la Planta de Segundo Orden	53
5.3	Resultados de Identificación.....	56
5.3.1	Identificación en una planta de un grado de libertad.....	57
5.4	Control.....	64

5.5	Control Adaptable por Modelo Inverso.....	64
5.5.1	Back-Propagation Salida 1	67
5.5.2	Back-Propagation Salida 2	74
5.5.3	Levenberg-Marquardt Salida 1	81
5.5.4	Levenberg-Marquardt Salida 2	88
5.6	Control Adaptable por Modelo Inverso con Termino Integral.....	95
5.6.1	Back-Propagation Salida 1 con Termino Integral	98
5.6.2	Back-Propagation Salida 2 con Termino Integral	105
5.6.3	Levenberg-Marquardt Salida1 con Termino Integral	112
5.6.4	Levenberg-Marquardt Salida 2 con Termino Integral	119
5.7	Control por Sliding-Mode con termino Integral.....	126
5.7.1	Sliding-Mode Control con i-term para una planta de un Grado de Libertad..	127
5.7.2	Sliding-Mode Control con i-term para una planta de dos Grado de Libertad...	128
6.	CONCLUSIONES.....	129
6.1	Trabajo a Futuro.....	130
	REFERENCIAS	131
	Anexo.....	134
	Background Matemático	134
	Números Complejos.....	134
	Funciones Complejas, Funciones Holomorfas o Analíticas y Funciones Armónicas.....	136
	Identification of nonlinear dynamical systems by means of complex-valued fuzzy-neural multi-model)	139

Índice de Figuras

Figura 1. Diagrama del Modelo de una Neurona Artificial.....	22
Figura 2. Función de Heaviside.....	23
Figura 3. Función Saturación.....	24
Figura 4. Función Sigmoide.	24
Figura 5. Función Tangente Hiperbolica.....	25
Figura 6. Diagrama del Modelo de una Neurona Artificial Compleja.	26
Figura 7. Esquema Genérico de la lógica Difusa.	36
Figura 8. Funciones de Membresía con un recubrimiento en cero.....	40
Figura 9. Nivel Jerárquico Bajo.....	44
Figura 10. Nivel Jerárquico Alto.	46
Figura 11. Diagrama Detallado de HFNMM.....	47
Figura 12. Planta idealizada de un grado de libertad.....	52
Figura 13. Planta idealizada de dos grados de libertad.....	53
Figura 14. Planta de dos grados de libertad.....	54
Figura 15. HFNMM para Identificación.....	56
Figura 16. Parte Positiva de Identificación (BP).....	58
Figura 17. MSE Instantáneo de la Parte Positiva de Identificación (BP).....	58
Figura 18. Parte Negativa de Identificación (BP).....	59
Figura 19. MSE Instantáneo de la Parte Negativa de Identificación (BP).....	59
Figura 20. Señal Completa y de Identificación (BP).....	60
Figura 21. MSE Instantáneo de Identificación (BP).....	60
Figura 22 Parte Positiva de Identificación (LM).....	61
Figura 23. MSE Instantáneo de la Parte Positiva de Identificación (LM).....	61
Figura 24. Parte Negativa de Identificación (LM).....	62
Figura 25. MSE Instantáneo de la Parte Negativa de Identificación (LM).....	62
Figura 26 Señal Completa y de Identificación (LM).....	63
Figura 27 MSE Instantáneo de Identificación (LM).....	63
Figura 28. Esquema de Control Adaptable por Modelo Inverso con HFNMM.....	65
Figura 29. Entrada para la Señal de Salida 1 (BP).....	67
Figura 30. Parte Positiva de la Entrada de la Señal 1 (BP).....	67

Figura 31. Parte Positiva de la Señal de Salida 1 (BP).....	68
Figura 32. Parte Positiva del Control de la Señal de Salida 1 (BP).....	68
Figura 33. Comparación de la Parte Positiva de la Señal de Entrada con la Señal de Salida (BP).	69
Figura 34. Error Medio Cuadrático Instantáneo de la Parte Positiva de la Señal 1 (BP).	69
Figura 35. Parte Negativa de la Entrada de la Señal 1 (BP).	70
Figura 36. Parte Negativa de la Entrada de la Señal de 1 (BP).	70
Figura 37. Parte Negativa del Control de la Señal de 1 (BP).	71
Figura 38. Comparación de la Parte Negativa de la Señal de Entrada con la Señal de Salida (BP).	71
Figura 39. Error Medio Cuadrático Instantáneo de la Parte Negativa de la Señal 1 (BP).....	72
Figura 40. Comparación de la Señal de Entrada con la Señal de Salida para la Señal 1 (BP). ..	72
Figura 41. Error Medio Cuadrático Instantáneo de la Señal 1 (BP).....	73
Figura 42. Entrada para la Señal de Salida 2 (BP).....	74
Figura 43. Parte Positiva de la Entrada de la Señal 2 (BP).....	74
Figura 44. Parte Positiva de la Señal de Salida 2 (BP).	75
Figura 45. Parte Positiva del Control de la Señal de Salida 2 (BP).....	75
Figura 46. Comparación de la Parte Positiva de la Señal de Entrada con la Señal de Salida (BP).	76
Figura 47. Error Medio Cuadrático Instantáneo de la Parte Positiva de la Señal 2 (BP).	76
Figura 48. Parte Negativa de la Entrada de la Señal de 2 (BP).	77
Figura 49. Parte Negativa de la Entrada de la Señal 2 (BP).	77
Figura 50. Parte Negativa del Control de la Señal de 2 (BP).	78
Figura 51. Comparación de la Señal de Entrada con la Señal de Salida para la Señal 2 (BP). ..	78
Figura 52. Error Medio Cuadrático Instantáneo de la Parte Negativa de la Señal 2 (BP).....	79
Figura 53. Comparación de la Señal de Entrada con la Señal de Salida para la Señal 2 (BP). ..	79
Figura 54. Error Medio Cuadrático Instantáneo de la Señal 2 (BP).	80
Figura 55. Entrada para la Señal de Salida 1 (LM).	81
Figura 56. Parte Positiva de la Entrada de la Señal 1 (LM).....	81
Figura 57. Parte Positiva de la Señal de Salida 1 (LM).	82
Figura 58. Parte Positiva del Control de la Señal de Salida 1 (LM).....	82

Figura 59. Comparación de la Parte Positiva de la Señal de Entrada con la Señal de Salida (LM).
..... 83

Figura 60. Error Medio Cuadrático Instantáneo de la Parte Positiva de la Señal 1 (LM). 83

Figura 61. Parte Negativa de la Entrada de la Señal 1 (LM). 84

Figura 62. Parte Negativa de la Entrada de la Señal de 1 (LM). 84

Figura 63. Parte Negativa de la Entrada de la Señal de 1 (LM). 85

Figura 64. Comparación de la Parte Negativa de la Señal de Entrada con la Señal de Salida (LM).
..... 85

Figura 65. Error Medio Cuadrático Instantáneo de la Parte Negativa de la Señal 1 (LM). 86

Figura 66. Comparación de la Señal de Entrada con la Señal de Salida para la Señal 1 (LM).. 86

Figura 67. Error Medio Cuadrático Instantáneo de la Señal 1 (LM). 87

Figura 68. Entrada para la Señal de Salida 2 (LM). 88

Figura 69. Parte Positiva de la Entrada de la Señal 2 (LM). 88

Figura 70. Parte Positiva de la Señal de Salida 2 (LM). 89

Figura 71. Parte Positiva del Control de la Señal de Salida 2 (LM). 89

Figura 72. Parte Positiva del Control de la Señal de Salida 2 (LM). 90

Figura 73. Error Medio Cuadrático Instantáneo de la Parte Positiva de la Señal 2 (LM). 90

Figura 74. Parte Negativa de la Entrada de la Señal 2 (LM). 91

Figura 75. Parte Negativa de la Entrada de la Señal de 2 (LM). 91

Figura 76. Parte Negativa del Control de la Señal de 2 (LM). 92

Figura 77. Comparación de la Parte Negativa de la Señal de Entrada con la Señal de Salida (LM).
..... 92

Figura 78. Error Medio Cuadrático Instantáneo de la Parte Negativa de la Señal 2 (LM). 93

Figura 79. Comparación de la Señal de Entrada con la Señal de Salida para la Señal 2 (LM).. 93

Figura 80. Error Medio Cuadrático Instantáneo de la Señal 2 (LM). 94

Figura 81. Entrada para la Señal de Salida 1 (BP-Ti). 98

Figura 82. Parte Positiva de la Entrada de la Señal 1 (BP-Ti). 98

Figura 83. Parte Positiva de la Señal de Salida 1 (BP-Ti). 99

Figura 84. Parte Positiva del Control de la Señal de Salida 1 (BP-Ti). 99

Figura 85. Comparación de la Parte Positiva de la Señal de Entrada con la Señal de Salida (BP-Ti). 100

Figura 86. Error Medio Cuadrático Instantáneo de la Parte Positiva de la Señal 1 (BP-Ti). ...	100
Figura 87. Parte Negativa de la Entrada de la Señal 1 (BP).	101
Figura 88. Parte Negativa de la Entrada de la Señal 1 (BP-Ti).	101
Figura 89. Parte Negativa del Control de la Señal de 1 (BP-Ti).	102
Figura 90. Comparación de la Parte Negativa de la Señal de Entrada con la Señal de Salida (BP-Ti).....	102
Figura 91. Error Medio Cuadrático Instantáneo de la Parte Negativa de la Señal 1 (BP-Ti)...	103
Figura 92. Comparación de la Señal de Entrada con la Señal de Salida para la Señal 1 (BP-Ti).	103
Figura 93. Error Medio Cuadrático Instantáneo de la Señal 1 (BP-Ti).	104
Figura 94. Entrada para la Señal de Salida 2 (BP-Ti).....	105
Figura 95. Parte Positiva de la Entrada de la Señal 2 (BP-Ti).....	105
Figura 96. Parte Positiva de la Señal de Salida 2 (BP-Ti).	106
Figura 97. Parte Positiva del Control de la Señal de Salida 2 (BP-Ti).	106
Figura 98. Comparación de la Parte Positiva de la Señal de Entrada con la Señal de Salida (BP-Ti).....	107
Figura 99. Error Medio Cuadrático Instantáneo de la Parte Positiva de la Señal 2 (BP-Ti). ...	107
Figura 100. Parte Negativa de la Entrada de la Señal 2 (BP-Ti).	108
Figura 101. Parte Negativa de la Entrada de la Señal de 2 (BP-Ti).	108
Figura 102. Parte Negativa del Control de la Señal de 2 (BP-Ti).	109
Figura 103. Comparación de la Parte Negativa de la Señal de Entrada con la Señal de Salida (BP-Ti).....	109
Figura 104. Error Medio Cuadrático Instantáneo de la Parte Negativa de la Señal 2 (BP-Ti).	110
Figura 105. Comparación de la Señal de Entrada con la Señal de Salida para la Señal 2 (BP-Ti).	110
Figura 106. Error Medio Cuadrático Instantáneo de la Señal 2 (BP-Ti).	111
Figura 107. Entrada para la Señal de Salida 1 (LM-Ti).....	112
Figura 108. Parte Positiva de la Entrada de la Señal 1 (LM-Ti).....	112
Figura 109. Parte Positiva de la Señal de Salida 1 (LM-Ti).....	113
Figura 110. Parte Positiva del Control de la Señal de Salida 1 (LM-Ti).....	113

Figura 111. Comparación de la Parte Positiva de la Señal de Entrada con la Señal de Salida (LM-Ti).	114
Figura 112. Error Medio Cuadrático Instantáneo de la Parte Positiva de la Señal 1 (LM-Ti).	114
Figura 113. Parte Negativa de la Entrada de la Señal 1 (LM-Ti).	115
Figura 114. Parte Negativa del Control de la Señal de 1 (LM-Ti).	115
Figura 115. Parte Negativa del Control de la Señal de 1 (LM-Ti).	116
Figura 116. Comparación de la Parte Negativa de la Señal de Entrada con la Señal de Salida (LM-Ti).	116
Figura 117. Error Medio Cuadrático Instantáneo de la Parte Negativa de la Señal 1 (LM-Ti).	117
Figura 118. Comparación de la Señal de Entrada con la Señal de Salida para la Señal 1 (LM-Ti).	117
Figura 119. Error Medio Cuadrático Instantáneo de la Señal 1 (LM-Ti).	118
Figura 120. Entrada para la Señal de Salida 2 (LM-Ti).	119
Figura 121. Parte Positiva de la Entrada de la Señal 2 (LM-Ti).	119
Figura 122. Parte Positiva de la Señal de Salida 2 (LM-Ti).	120
Figura 123. Parte Positiva del Control de la Señal de Salida 2 (LM-Ti).	120
Figura 124. Comparación de la Parte Positiva de la Señal de Entrada con la Señal de Salida (LM-Ti).	121
Figura 125. Error Medio Cuadrático Instantáneo de la Parte Positiva de la Señal 2 (LM-Ti).	121
Figura 126. Parte Negativa de la Entrada de la Señal 2 (LM-Ti).	122
Figura 127. Parte Negativa de la Entrada de la Señal de 2 (LM-Ti).	122
Figura 128. Parte Negativa del Control de la Señal de 2 (LM-Ti).	123
Figura 129. Comparación de la Parte Negativa de la Señal de Entrada con la Señal de Salida (LM-Ti).	123
Figura 130. Error Medio Cuadrático Instantáneo de la Parte Negativa de la Señal 2 (LM-Ti).	124
Figura 131. Comparación de la Señal de Entrada con la Señal de Salida para la Señal 2 (LM-Ti).	124
Figura 132. Error Medio Cuadrático Instantáneo de la Señal 2 (LM-Ti).	125
Figura 133. Esquema de Control Por Sliding-Mode.	126
Figura 134. Comparación de la Entrada y la Salida (SMC).	127
Figura 135. Error Medio Cuadrático Instantáneo	127

Figura 136. Comparación de la Entrada y la Salida (SMC).	128
Figura 137. -Error Medio Cuadratico Instantaneo.	128

Índice de Tablas

Tabla 1. Modus Ponens.....	36
Tabla 2. Modus Tollens	36
Tabla 3. Tabla de parámetros de la planta de un grado de libertad.	53
Tabla 4. Tabla de parámetros de la planta con dos grados de libertad.	56
Tabla 5. Tabla de parámetros para las CVRNN utilizadas para la identificación de una planta de un grado de libertad	57
Tabla 6. Tabla de parámetros para las CVRNN utilizadas para Control Adaptable por Modelo Inverso de una planta de dos grados de libertad.	66
Tabla 7. Tabla de parámetros para las CVRNN utilizadas para Control Adaptable por Modelo Inverso con Término Integral de una planta de dos grados de libertad.	97
Tabla 8. Tabla de los valores usados de las ganancias integrales para el Control Adaptable por Modelo Inverso con Término Integral.	97

1. INTRODUCCIÓN

La evolución es un factor primordial para que la humanidad haya llegado hasta este punto de la historia, la ciencia y la tecnología son consecuencia de lo anterior; sin embargo, también han adquirido la categoría de “causa”, de tal manera que ahora son factores importantes para el continuo desarrollo del hombre. Durante este proceso se han resuelto complejos enigmas, así como paradigmas correspondientes a cada una de las “eras” de la historia; todo ello posible a que en algún momento se concibiera una “simple idea” dentro del cerebro humano, dado esto y el tiempo era inevitable que se hicieran las siguientes preguntas: ¿Qué es el cerebro? ¿Cómo funciona? Y mucho más complicado de responder, ¿Cómo emulamos sus funciones artificialmente?

Los modelos son un medio, una manera de representar lo abstracto, el cerebro es complejo por su naturaleza; por lo que se ha optado por hacer “caricaturas” que permitan resaltar sus diferentes cualidades, tales como: resolver problemas, aprendizaje, memoria etc. Prueba de ello son las Redes Neuronales Artificiales. Sin embargo, se ha incurrido en abuso de lenguaje dado que no son verdaderos modelos de fenómenos biológicos ó cognitivos. La Red Neuronal, *“Es un procesador distribuido masivamente en paralelo que tiene una propensión natural a almacenar el conocimiento experimental y hacerlo disponible para su uso”*, tal y como se menciona en [1], [2], [3]. De lo anterior es posible destacar que la propiedad más importante es la habilidad para aprender en un entorno, es por ello por lo que uno de los campos de aplicación han sido los sistemas dinámicos, cuyos estados evolucionan con el tiempo, dando la oportunidad a la teoría de control de tener otra vía para resolver dos problemas importantes que son la identificación y el control; esto fue empleado por primera vez por Narendra y Parthasarathy en su artículo de 1990, [4], mostrando sus bondades tanto para sistemas lineales como sistemas no lineales. El trabajo realizado a posteriori de los resultados anteriores y hasta la actualidad, ha mostrado que las redes neuronales artificiales por su habilidad de aproximar funciones lineales y no lineales, sin importar su naturaleza han ganado un lugar fundamental en el control adaptable.

Por otro lado, el uso de la lógica difusa, para la identificación y el control, basado en implicaciones y razonamiento difusos, representan una herramienta matemática para expresar el modelo de un sistema, así como su identificación, además aprovecha la experiencia del humano en una determinada tarea para generar las acciones de control, Takagi y Sugeno en 1985 [5].

Por lo tanto, su unión ha representado la oportunidad de diseñar controladores de lógica difusa con el enfoque de entrenamiento y aprendizaje [6], [7]. La combinación de las redes neuronales y la lógica difusa, han tenido como resultado los multi-modelos neuronales, algunos de sus resultados es posible observarlos en [8], [9]. En la parte de control incluso ha sido combinada con técnicas avanzadas de control tal como los Modos Deslizantes (*Sliding Mode Control*), teniendo buenos resultados como se muestra en [10]. La mayoría de las aplicaciones de las redes neuronales artificiales son expresadas en el dominio real, por lo que el dominio complejo tiene aplicaciones muy escasas, uno de los más importantes autores sobre el tema es A. Hirose, quien en sus libros, [11], [12], muestra diferentes ejemplos de sistemas complejos donde el dominio complejo tiene aplicación, por ejemplo, la mecánica cuántica con la ecuación de Schrödinger ó el mismo cerebro humano, teniendo en, [12], una muy atrevida sección llamada “*Create a “SuperBrain” by Enrichment of the Information Representation*”.

Los algoritmos de aprendizaje tienen un papel fundamental dentro de cualquier red neuronal artificial entrenable, ya que a través de ellos se busca reducir una función de costo ó índice de desempeño, que tienen como variable el error. En 1991, se derivó el algoritmo *Back-Propagation* en el dominio complejo, [13]. En [14], [15], retomaron la idea del algoritmo de aprendizaje y a través de una red adjunta obtenida por el método diagramático propuesto por [16]; esto se utilizó para plantas no lineales oscilatorias con buenos resultados. La misma técnica fue utilizada para obtener el algoritmo de aprendizaje *Levenberg-Marquardt* en el dominio complejo, [17], [18], en los que se demostró que tiene ciertas ventajas sobre el algoritmo *Back-Propagation*.

El uso de las redes neuronales artificiales en el dominio complejo ha incrementado en los últimos años, y la atractiva propuesta de combinarlas con la lógica difusa se presenta a continuación.

1.1 Planteamiento del Problema y Objetivos

El campo de los números complejos \mathbb{C} , posee diversas propiedades que, al ser aplicadas en las Redes Neuronales, da la capacidad de extender sus parámetros y representar mejor las dinámicas de sistemas no lineales oscilatorios, tal y como se ha demostrado en los análisis de circuitos eléctricos, ondas electromagnéticas e incluso la mecánica cuántica.

El hecho de trabajar en el dominio complejo representa dificultades al haber pocos autores y trabajos sobre la rama, es por ello por lo que aplicar las mismas consideraciones teóricas puede desembocar en problemas en la obtención de resultados, tal y como se observa en los trabajos, [14], [15], [17], [18]. Al presentar ciertas irregularidades en la manera de elegir la Topología de la Red Neuronal Recurrente con Valores Complejos.

El presente trabajo busca llevar a cabo la unificación de los trabajos antes mencionados con el fin de realizar un análisis sobre el impacto que tienen los números complejos en la Topología desarrollada dentro de [19].

Además, su implementación en un Multi-Modelo Difuso, para llevar acabo la vieja y conocida maniobra de guerra, “Divide y Vencerás”, teniendo como ejemplos sistemas no lineales oscilatorios de primer y segundo orden; en los cuales son sometidos a tareas de identificación y control, con los algoritmos de entrenamiento *Levenberg-Marquardt* y *Back-Propagation*. A fin de compararlos de forma cuantitativa y cualitativa, bajo entradas de tipo cuadrada y senoidal.

Cabe mencionar que dada la arquitectura que toman los sistemas de identificación y control al ser aplicados en un Multi-Modelo Difuso-Neural Jerárquico, ha sido necesario puntualizar y modificar las formas adoptadas de Topología para los tipos de sistemas: una entrada y una salida, (*SISO*); múltiples entradas y una salida (*MISO*), múltiples entradas y múltiples salidas (*MIMO*).

1.2 Organización de la Tesis

Capítulo 1

Se lleva a cabo una recopilación de los temas directriz del presente trabajo, los cuales son: Redes Neuronales en el Dominio Complejo, pasando por un antecedente hasta su uso en los últimos años. También se plantea el problema a resolver que debe ser entendido como la motivación del trabajo, así como un enfoque con el cual se piensa abordar el problema, esto es realizado a través de objetivos específicos.

Capítulo 2

Se hace una descripción de lo que son las Redes Neuronales en el Dominio Complejos, haciendo mención de sus características principales, su modelo; así como el aprendizaje realizado por las redes recurrentes entrenables. En este capítulo se remarca que la función de activación es la característica más importante de las Redes Neuronales en el Dominio Complejo.

Capítulo 3

Se plantea las bondades del uso de los Sistemas Difusos, como una técnica de control, la cual usa los conocimientos o saberes previamente adquiridos por una persona para el diseño de la estrategia de control que sea capaz de realizar una actividad asignada, así como responder de una manera adecuada para las diferentes situaciones que se presenten durante su ejecución.

Capítulo 4

Se muestra como la combinar las técnicas presentadas en los capítulos 2 y 3, se genera un Multi-Modelo Difuso-Neuronal Jerárquico en el dominio complejo, la cual es aplicada para los problemas de identificación de sistemas, así como diferentes estrategias de control, donde su aprendizaje es a través de los algoritmos B-P y L-M.

Capítulo 5

Se presentan los resultados de Identificación y Control.

Capítulo 6

Conclusiones Generales y Trabajos a Futuro.

2. REDES NEURONALES EN EL DOMINIO COMPLEJO

La Inteligencia, es el discriminante que nos permite ser únicos de entre todos los seres vivos conocidos; es una de las principales características de la naturaleza humana dada su complejidad y su misterio, que intriga por sí misma. Es por ello por lo que no solo ha sido estudiado únicamente desde la perspectiva Científica, sino también desde otros puntos de vista tal y como lo son: la Filosofía, el Psicoanálisis, la Biología, etc. El cerebro es un sistema de procesamiento de información altamente complejo, no lineal y paralelo, la forma en que organiza las neuronas para resolver determinadas tareas se le denomina razonamiento humano. Esto ha generado un gran interés en obtener un entendimiento y comprensión simbólica de los mecanismos físicos que le otorgan la categoría de seres inteligentes. Todos los estudios tienen la intención de diseñar dispositivos basados en el sistema nervioso y el cerebro, tomando su funcionamiento, estructura y propiedades lógicas; para generar sistemas capaces de resolver problemas, con una propiedad de adaptabilidad. Cabe mencionar que el cerebro tiene un desempeño mucho más rápido que uno computacional [2] [20].

El humano ha tratado de imitar las diferentes funciones, características y estructuras desarrollando una red neuronal artificial. [2], ofrece una definición formal de esto:

“Es un procesador distribuido masivamente en paralelo que tiene una propensión natural a almacenar el conocimiento experimental y hacerlo disponible para su uso. Es análogo al cerebro en dos aspectos:

- *El conocimiento es adquirido por la red del medio ambiente a través de un proceso de aprendizaje.*
- *Las fuerzas de conexión inter-neuronal, pesos sinápticos, son usadas para almacenar el conocimiento adquirido.”*

A partir de aquí, la Red Neuronal Artificial será simplemente llamada Red Neuronal; tiene como objetivos principales el emular y obtener *conocimiento*, cuya meta es alcanzada a través de un proceso de *aprendizaje* el cual es llevado a cabo por un algoritmo de entrenamiento, su función es modificar los pesos sinápticos de una forma ordenada, hasta alcanzar la respuesta deseada.

2.1 Modelo de una Neurona

En 1943, W. McCulloch y W. Pitts, [2], [20], [21], en el cual presentan un modelo de la estructura y funcionamiento que permite obtener una representación simplificada de las neuronas del cerebro, donde las consideran como mecanismos binarios, es decir, apagado o encendido.

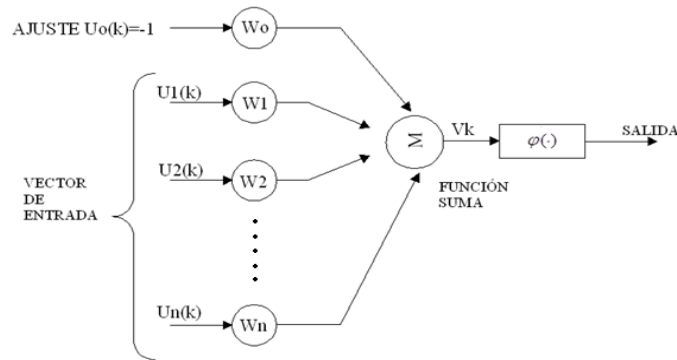


Figura 1. Diagrama del Modelo de una Neurona Artificial.

La Fig. 1, consta de un conjunto de n valores de entrada $U = \{u_1, u_2, \dots, u_m\}$, un estado interno el cual está dado por la sumatoria de los pesos sinápticos $w_i \in \mathfrak{R}$, $i = 1, 2, \dots, n$ y una salida y_k , las ecuaciones que describen lo anterior están dadas por:

$$v(k) = \sum_{i=0}^n w_i u_i(k) \quad (2.1)$$

$$y(k) = \varphi(v(k)) \quad (2.2)$$

Donde $\varphi(\cdot)$ es la función de activación y “k” es la variable en tiempo discreto.

En la Figura 1, se observan 3 tipos de neuronas:

- Aquellas que tienen una función sensorial para recibir estímulos externos, pueden ser provenientes del entorno o medio ambiente o ser salidas de neuronas anteriores; por lo que se le denomina como capa de entrada y forman un vector U de dimensión m .
- Las unidades ocultas, que se encargan del proceso de la información, es donde se encuentran los pesos sinápticos w_i , donde cada uno de los elementos del vector W multiplica a cada elemento del vector entrada, teniendo así una ponderación de la señal de entrada; los pesos no tienen un valor

fijo ya que son modificados con el fin de obtener un mejor desempeño. Estas neuronas internas no tienen relación con la información de entrada ni con la de salida.

- **Neuronas de salidas:** Su propósito es dar la respuesta del sistema, la cual se genera a través de la adición del producto $u_i * w_i$, obteniendo una combinación lineal $v(k)$, siendo este un campo local inducido.

2.2 Funciones de Activación en el Dominio Real

El comportamiento no lineal de una red es proporcionado por la función de activación denotada por $\varphi(\cdot)$, es acotada y representa la salida de la neurona en términos del campo local inducido $v(k)$, por lo que $\varphi : \mathfrak{R} \rightarrow \mathfrak{R}$. El teorema universal de aproximación de funciones da condiciones para determinar si una función φ es una función de activación, [2]. Las funciones comúnmente utilizadas, están acotadas en la salida por $[0,1]$ o $[-1,1]$, también deben ser diferenciables en todo su dominio.

- **Función de Heaviside:** También denominada función escalón, se encuentra definida por la siguiente ecuación:

$$\varphi(v) = \begin{cases} 1 & \text{si } v \geq 0 \\ 0 & \text{si } v < 0 \end{cases} \quad (2.3)$$

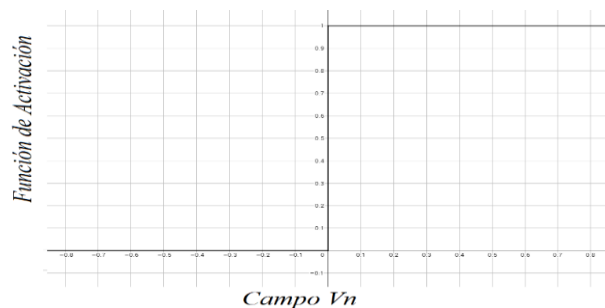


Figura 2. Función de Heaviside.

- **Función saturación:** También llamada función línea con saturación, definida por la ecuación:

$$\varphi(v) = \begin{cases} 0 & \text{si } v \leq -0.5 \\ v + 0.5 & \text{si } -0.5 < v < 0.5 \\ 1 & \text{si } v \geq 0.5 \end{cases} \quad (2.4)$$

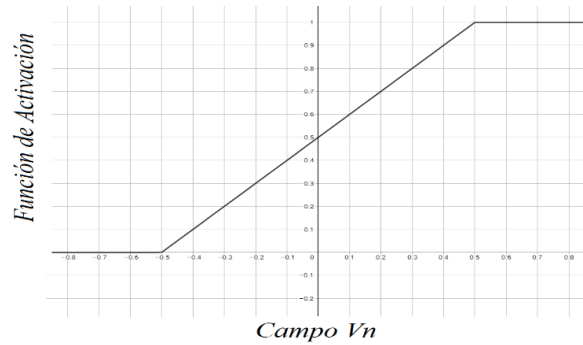


Figura 3. Función Saturación.

- **Función sigmoidea:** Es la función de activación más utilizada en la construcción de redes neuronales. Cuenta con una región de comportamiento lineal, sin embargo, es diferenciable en todo su dominio; y se encuentra definida por la ecuación:

$$\varphi(v) = \frac{1}{1 + e^{-av}} \quad (2.5)$$

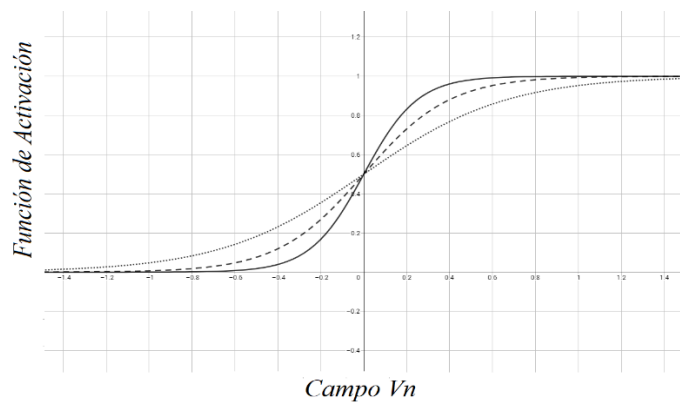


Figura 4. Función Sigmoide.

- **Función tangente hiperbólica:** Tiene un comportamiento similar a la función sigmoidea, su intervalo de salida es de $[-1,1]$. Está definida por la ecuación:

$$\varphi(v) = \frac{1 - e^{-2v}}{1 + e^{-2v}} \quad (2.6)$$

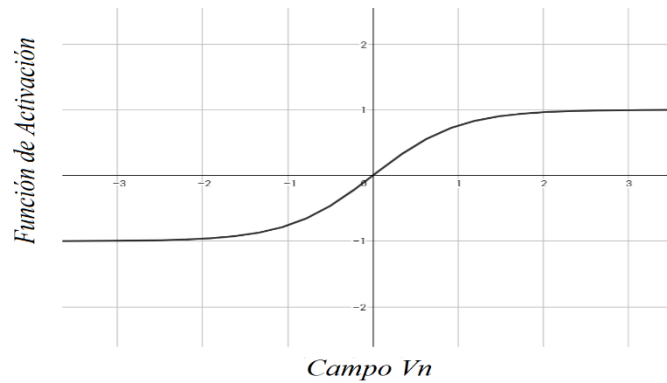


Figura 5. Función Tangente Hiperbólica.

2.3 Redes Neuronales en el Dominio Complejo

La realidad de las redes neuronales es que sólo se ha podido usar el nombre, así como algunas de propiedades generales, esto es porque el hombre no tiene más que una pobre comprensión de sus funciones en redes biológicas. Visto desde un punto de vista computacional, las neuronas son la reflexión espacial y temporal de las sinapsis en las dendritas, donde la función de activación limita el rango de amplitud de la salida. El aprendizaje es la forma de conseguir nuevo conocimiento, uno de los métodos más usuales es el retro-propagación del error basado en el gradiente descendiente, al ser expresado en el dominio real pertenece a dimensión 1 (1D), como ya se ha mencionado, las redes neuronales artificiales se han usado para construir modelos matemáticos que puedan reproducir el comportamiento del intelecto humano. Tienen muchas aplicaciones teóricas y prácticas, las cuales muchas de las veces no requieren suposiciones fuertes. Esto se ha extendido a dos dimensiones (2D) mediante el uso de los números complejos, [22], [23], este enfoque proporciona una solución más eficiente computacionalmente en términos de complejidad y desempeño.

El caso complejo de las redes neuronales es donde todos los pesos, funciones de activación, *bias*, entradas y salidas, son números complejos, [12]. Toda la información es tratada en un plano bidimensional, donde los números complejos están compuestos por dos números reales que en conjunto contienen una fase. Esto se ha utilizado en problemas donde es necesario recolectar información de la fase y además retenerlo duran un periodo de tiempo, por ejemplo, los pulsos que viajan a través del sistema nervioso, donde puede existir más de uno con diferente amplitud y fase.

Este podría ser una justificación por el cual el dominio complejo ayuda a resolver problemas o sistemas complejos de la naturaleza en el momento de emular su comportamiento. El problema de la XOR es uno de los ejemplos favoritos de la bibliografía y ha sido resuelto exitosamente a través del uso dos capas complejas insertadas en una red neuronal, [24], [25]. Otro campo de aplicación ha tenido lugar en la predicción de series de tiempo, lo cual forma parte fundamental del problema de aproximación de funciones; los resultados obtenidos a partir de diferentes arquitecturas y métodos de aprendizaje han sido efectivos en la solución del problema, [26], [27], [28].

Cabe mencionar que las redes neuronales complejas presentan una mayor riqueza en su aprendizaje, cantidad de información que pueden almacenar, esto produce un aumento la capacidad de procesamiento y desempeño en general. La representación diagramática de una red neuronal compleja es una extensión de la contraparte real, por lo que la definición de la función de activación cambiará su arquitectura por lo general.

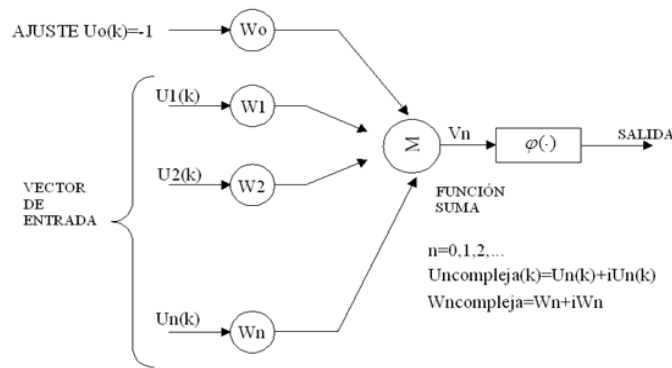


Figura 6. Diagrama del Modelo de una Neurona Artificial Compleja.

Dada la figura anterior, una neurona en el dominio complejo queda expresada en términos matemáticos de la siguiente forma:

$$v_n = \sum_{j=1}^N (w_{Re,nj} + i \cdot w_{Im,nj}) \cdot (x_{Re,j} + i \cdot x_{Im,j}) \quad (2.7)$$

$$y_{Re,n} + i \cdot y_{Im,n} = \varphi(u_n) \quad (2.8)$$

2.4 Funciones de Activación en el Dominio Complejo

Históricamente, la primera función de activación de valor complejo para una neurona fue una función de activación dependiente de la fase y fue propuesta en 1971, [29], algunos trabajos, [13], [22], han desarrollado totalmente una red neuronal en el dominio complejo basado en funciones holomorfas, acotadas y sin puntos de singularidad en su dominio (ver Anexo 1).

Se destaca que por construcción existe una dependencia entre las funciones de activación de forma directa con la arquitectura y derivación del algoritmo de entrenamiento. Dado que no es posible tener una función acotada en todo su dominio, sin puntos de singularidad y que a su vez sea holomorfa; en trabajos relacionados con la extensión del algoritmo de entrenamiento *Back-Propagation* hacia el dominio complejo adoptaron funciones de activación construidas, [23] [30].

Se ha propuesto que la siguiente función de activación dada su simplicidad y habilidad para mantener limitadas la amplitud y fase de señales complejas y además ser una extensión de función de activación real, se encuentra descrita por la ecuación, [28]:

$$f_c(v) = f(\Re(v)) + i \cdot f(\Im(v)) \quad (2.9)$$

Donde, se tiene una adición de f , una función no lineal, con la separación de argumentos reales e imaginarios, se ha convertido en una de las más usadas, [12].

2.5 Aprendizaje en el Dominio Complejo

Como ya se había mencionado, el aprendizaje es el proceso a través del cual una red neuronal modifica sus pesos, cuando los pesos dejan de tener modificaciones, se dice que la red ha aprendido. Una técnica popular para desarrollar algoritmos de aprendizaje o entrenamiento es la basa en el gradiente descendiente, [13], [22], [23], [30], [31], y es dependiente como se mencionó en la sección anterior que la función de activación provee su arquitectura, por lo que, al dividir la parte real y la parte imaginaria, los algoritmos adquieren la habilidad de evitar los puntos de singularidad durante el proceso de entrenamiento. Con la función de activación dividida, la actualización de las reglas son una combinación lineal de derivados de componentes reales e imaginarios de una función de activación, [28].

2.6 Redes Neuronales Recurrentes Entrenables

Las Redes con Neuronas Recurrentes (RNR) tienen como característica principal: crear conexiones de tipo *feed-forward* y *feed-back*, de tal manera que pueden aparecer conexiones de una neurona con ella misma, conexiones entre una misma capa, incluso con una capa anterior. Esto significa un aumento considerable en los parámetros ajustables dentro de la red, por lo tanto, facilita el tratamiento de información temporal o patrones dinámicos, en [20], esto es explicado de forma concisa “*Patrones que dependen del tiempo en el sentido de que el valor del patrón en un determinado instante depende de sus valores en instantes anteriores en el tiempo*”. Las RNR que no son entrenables debido al lazo de realimentación, pueden presentar inestabilidad, [32], en la literatura tratan sistemas sencillos SISO y no se profundiza en problemas de estabilidad y casos MIMO. Desde hace varios años Baruch y otros, han trabajado con Redes Neuronales Recurrentes Entrenables, donde la salida es un campo local inducido, esto corresponde a los estados de la Red Neuronal. Por otro lado, el objetivo que persigue es el de establecer unos valores para el vector de pesos con los cuales el error cometido al evaluar los ejemplos de entrenamiento sea mínimo.

3. SISTEMAS DIFUSOS

Para poder hablar de Sistemas Difusos, es necesario mencionar en primer lugar a los Sistemas Expertos, los cuales según N. Kasabov, [33], los incluye como una parte del área de la Inteligencia Artificial, cuyo principal objetivo es el desarrollo de modelos, métodos y tecnologías básicas para emular el proceso de conocimiento o razonamiento del ser humano.

Los Sistemas Expertos, se basan en el conocimiento previamente adquirido por una persona en una determinada tarea, es decir, todo aquello que la persona ha experimentado como logro, fracaso, etc. Le ha provisto de experiencia y conocimiento suficiente del sistema como para anticipar comportamientos o solucionar problemas cuando se susciten. Cuando lo anterior se combina con una herramienta computacional se obtienen excelentes resultados, ya que un operador sin experiencia puede confiar y guiarse para la toma de decisiones con ayuda de estos sistemas, así como una mejor comprensión de la tarea; una de las maneras que utilizan computacionalmente para el aprendizaje de este tipo de sistemas es a través de las Redes Neuronales Artificiales. Este tipo de sistemas se ha utilizado con éxito en diferentes campos donde ha presencia de la actividad humana tales como lo pueden ser: ingeniería, ciencia, medicina, agricultura, finanzas, etc.

En conclusión, los Sistemas Difusos, se basan en la generación e interpretación de reglas descriptivas que son de fácil entendimiento para todo ser humano, pero en contraparte el cómo “explicar” a una maquina en su lenguaje este tipo de conceptos sin fronteras bien definidas es altamente complejo, debido a que la experiencia adquirida por una persona y su habilidad para comunicarla tiene diferentes sesgos como lo pueden ser ignorancia teórica o ignorancia práctica y en su conjunto genera una incertidumbre en el esquema general de conocimiento sobre una situación o tarea en particular y el razonamiento de la solución a un problema suscitado durante la realización de esa misma actividad.

3.1 Lógica Difusa

Los sistemas difusos son una consecuencia de los sistemas expertos basados en *fuzzy rules* representan de manera directa el conocimiento y las habilidades de "sentido común", o el conocimiento que es subjetivo, empírico, ambiguo, vago o contradictorio; y *fuzzy inference*, esto hace referencia a la interpretación de las *fuzzy rules*, con el fin obtener valores difusos, a partir de las variables de entrada. Algunas aplicaciones las podemos encontrar en el enfoque automático de

las cámaras, control de sistemas de transmisión en nuevos modelos de automóviles, sistemas automáticos de aterrizajes para aeronaves, control automático de subterráneos, etc. Una de las propiedades más distinguidas de la lógica difusa es la interpretación de proposiciones difusas, es decir, la relación de variables lingüísticas como pueden ser: frío, caliente, pequeño, grande, etc. Y a diferencia de la lógica booleana, true o false, acepta toda una gama de grados entre dos extremos, por ejemplo: entre el frío y caliente existe tibio.

Un sistema difuso está definido por tres componentes principales:

- Variables de entrada y salida Difusas
- Conjunto de Reglas Difusas
- Mecanismo de Inferencia Difusa

Las virtudes y éxito de un Sistema Difuso están en que son fáciles de implementar, mantener, de comprender; además son robustos y baratos. Por lo que, se puede decir de una manera formal la Lógica Difusa, *“Es una lógica multi-valuada que permite representar matemáticamente la incertidumbre y la vaguedad, proporcionando herramientas formales para su tratamiento”*, [34].

3.2 Conjuntos Difusos

El inventor de la lógica difusa, L. Zadeh, [35] propone por primera vez la noción de Conjunto Difuso, la colección de elementos que pertenece a un universo, puede tener un grado de pertenencia dentro de dicho universo, es decir, se usan valores de 0 a 1, donde: 0 se puede entender como que no pertenece al conjunto y 1 se entiende como que pertenece completamente al conjunto; y entre estos valores existe un número infinito de opciones para asociar a cada uno de los elementos del conjunto. Por lo que, la Lógica Binaria es un caso particular de la Lógica Difusa, una manera tradicional de representar si un elemento pertenece a un conjunto o no es de la siguiente manera:

$$\mu_A(u) = 1 \quad (3.1)$$

Lo anterior se puede interpretar como: u es un elemento del conjunto A .

$$\mu_A(u) = 0 \quad (3.2)$$

La ecuación (3.2) menciona que: u no es elemento de A . En conclusión, las ecuaciones (3.1) y (3.2), dictaminan si un elemento u pertenece a un conjunto A .

Dentro los conjuntos difusos, un objeto no necesariamente puede pertenecer totalmente a un conjunto, esto abre la posibilidad a que tenga un grado de membresía, dicho de otra manera, forma parte parcialmente y se puede escribir matemáticamente de la siguiente manera:

$$\mu_A(u): U \rightarrow [0,1] \quad (3.3)$$

Donde U es el universo y A es un subconjunto de U.

El proceso denominado Fuzzificación, es el proceso de asociar conceptos a conjuntos difusos, el intervalo de la ecuación (3.3) es de números reales e incluye los extremos, por lo que, al generalizar dichos valores para así indicar el grado de pertenencia de los elementos a ese conjunto, se obtendrá un FUNCIÓN DE PERTENENCIA O MEMBRESÍA de un determinado conjunto difuso. Este tipo de funciones también llamadas de inclusión, son un conjunto de pares ordenados:

$$FM = \{(u, \mu_{FM}(u) / u \in U)\} \quad (3.4)$$

Para definir el tipo de función adecuada o que se usa convencionalmente se usan ciertas formas estándar, por tener una relación con un significado lingüístico, las más frecuentes son campanas de Gauss, trapezoidal, triangular, exponencial, singleton, y tipo π . Se puede tener una información más detallada en la tesis de J. Olivares, [10].

3.3 Variables Lingüísticas

Este tipo de variables nos sirven para expresar la relación que puede existir entre el conocimiento y el razonamiento, es decir, es una variable que acepta palabras del lenguaje natural o artificial, esto surge del hecho que hay fenómenos complejos que no es posible plantearlos en forma numérica y no cuentan con definición puntual.

En, [34], [36], es representado la variable lingüística como la quintupla:

$$(X, T(X), U, G, M) \quad (3.5)$$

Donde:

- X es el nombre de la variable
- T(X) es el conjunto de valores lingüísticos
- U representa el universo en discurso

- G es la gramática libre de contexto por la cual se generan los términos en T(x)
- M es una regla semántica a través de la cual se asocia el valor lingüístico de X

En conjunto con lo anterior, se han añadido una serie de prefijos y sufijos que permiten enfatizar cada una de las variables lingüísticas.

- Términos primarios: “bajo”, “alto”, “pequeño”, “grande”, ...,
- Modificadores: “muy”, “más”, “menos,” lejano de”, “cercano de”
- Conectores lógicos: NOT, AND y OR

Normalmente los términos primeros son lo que dictan la definición de los conjuntos difusos, para luego ser complementado con símbolos terminales de gramática, por ejemplo, “Muy lento” o también puede ser “No muy lento”, esto va depender claramente del experto que tenga la información sobre la dinámica de un proceso. El uso de este tipo de variables es utilizado en los sistemas difusos para interacción de reglas difusas. Cabe mencionar que el universo en discurso U puede tomar cualquier rango de valores, sin embargo, en aplicaciones físicas es conveniente tomar extremos alcanzables de tal forma que los resultados esperados sean obtenidos bajo condiciones normales de operación. Evidentemente de cada variable lingüística se pueden obtener infinitas particiones, y se encuentra debidamente explicado en tesis, [8] [10].

3.4 Principio de Extensión

Este principio permite transformar conceptos no difusos en difusos, por lo que es parte fundamental de la inferencia la cual se ocupa del razonamiento formal con proposiciones, sin embargo, dadas las características de la lógica difusa es posible tomar valores intermedios entre 0 y 1, o entre falso y verdadero.

Al tener dos universos U y V en discurso, existe $f : U \mapsto V$, esto es aplicando el principio que para un conjunto difuso A en U se puede definir un conjunto B en V dado por la siguiente ecuación, [8]:

$$\mu_B(v) = \sup_{u \in f^{-1}(v)} [\mu_A(u)] \quad (3.6)$$

Se interpreta cómo: sí $\mu_B(v)$ es el máximo de $\mu_A(u)$ para todos los $u \in U$ que cumplan con $f(u) = v$ dónde $v \in V$ y bajo la suposición de que $f^{-1}(v)$ es vacío para algún $v \in V$ se definirá a $\mu_B(v) = 0$.

3.5 Operaciones Difusas

A la interacción de conjuntos difusos se les denominara operaciones difusas, las cuales al tener como entrada dichos conjuntos, producirán también otro conjunto, y para al menos dos conjuntos etiquetados como por ejemplo A y B, asociados a una variable lingüística x, es posible definir tres operaciones básicas:

- Unión
- Intersección
- Complemento

Además, se pueden expresar en términos de las funciones de membresía de cada uno de los conjuntos difusos. Cabe destacar que estas operaciones coinciden con las operaciones del mismo nombre que se definen habitualmente para los conjuntos clásicos.

3.5.1 Unión

Se define como, [34]:

$$\mu_{A \cup B}(x) = \perp [\mu_A(x), \mu_B(x)] \quad (3.7)$$

Lo anterior debe satisfacer los siguientes axiomas, para considerarse como una unión difusa, $\forall a, b, c \in [0, 1]$:

- Conmutatividad:

$$\perp(a, b) = \perp(b, a) \quad (3.8)$$

- Asociatividad:

$$\perp(\perp(a, b), c) = \perp(a, \perp(b, c)) \quad (3.9)$$

- Monotonicidad: Si $a \leq c$ y $b \leq d$ entonces:

$$\perp(a, b) = \perp(c, d) \quad (3.10)$$

- Elemento Neutro:

$$\perp(a, 0) = a \quad (3.11)$$

La T-conorma que se utiliza en este trabajo es:

$$\perp(a, b) = \max(a, b) \quad (3.12)$$

3.5.2 Intersección

Se define como, [34]:

$$\mu_{A \cap B}(x) = T[\mu_A(x), \mu_B(x)] \quad (3.13)$$

Lo anterior debe satisfacer los siguientes axiomas, para considerarse como una unión difusa,

$\forall a, b, c \in [0, 1]$:

- Conmutatividad:

$$T(a, b) = T(c, d) \quad (3.14)$$

- Asociatividad:

$$T(T(a, b), c) = T(a, T(b, c)) \quad (3.15)$$

- Monotonicidad: Si $a \leq c$ y $b \leq d$ entonces:

$$T(a, b) = T(c, d) \quad (3.16)$$

- Elemento Neutro:

$$T(a, 0) = a \quad (3.17)$$

La T-norma que se utiliza en este trabajo es:

$$T(a, b) = \min(a, b) \quad (3.18)$$

3.5.3 Complemento

El complemento de un conjunto difuso se define cómo:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (3.19)$$

La anterior ecuación tiene una interpretación particular definida por $c: [0,1] \rightarrow [0,1]$ y que tiene que cumplir con los siguientes axiomas, [34]:

- Condiciones de límite: $c(0) = 1$ y $c(1) = 0$
- Monotonicidad: $\forall a, b \in [0,1]$ si $a < b$ entonces $c(a) \geq c(b)$
- c es una función continua
- c es involutiva $\forall a \in [0,1]$ se tiene $c(c(a)) = a$

3.5.4 Propiedades de Operaciones Difusas para Sub-conjuntos

Además de las operaciones básicas, también se cuentan con las siguientes propiedades, las cuales permiten obtener diferentes combinaciones las cuales servirán para definir el grado de pertenencia de un elemento a una función de membresía.

- Conmutativa $A \cap B = B \cap A$
- Asociativa $A \cup (B \cap C) = (A \cup B) \cap C$
- Distributiva $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
- Idempotencia $A \cup A = A$ y $A \cap A = A$
- Involución $\neg(\neg A) = A$
- Transitiva *if* $(A \subset B) \cap (B \subset C)$ *then* $A \subset C$

3.6 Métodos Básicos de Inferencia

Las reglas difusas son del tipo: antecedente y consecuencia, es decir IF/THEN y expresan una relación o proposición difusa. Recordando que este tipo de lógica, el razonamiento es aproximado, esto es que se puede inferir de una regla una conclusión sin que la premisa (antecedente) se cumpla completamente. Existen dos métodos básicos de inferencia.

3.6.1 Modus Ponens Generalizado

El *modus ponens* proveniente del latín: "el modo que, al afirmar, afirma" y se describe como:

Tabla 1. Modus Ponens

Conocimiento	Hecho	Consecuencia
Sí "x" es A Entonces "y" es B	"x" es A'	"y" es B'

Donde A, A' y B, B', son conjuntos difusos.

3.6.2 Modo Tollens Generalizado

El *modus tollens* proveniente del latín: "el modo que, al negar, afirma" y se describe cómo:

Tabla 2. Modus Tollens

Conocimiento	Hecho	Consecuencia
Sí "x" es A Entonces "y" es B	"x" es B'	"x" es A'

Donde A, A' y B, B', son conjuntos difusos.

3.7 Esquema Genérico de Lógica Difusa

A continuación, se presenta el esquema genérico de la lógica difusa

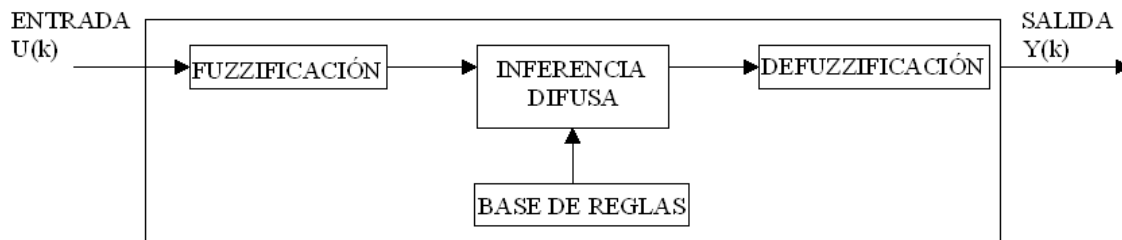


Figura 7. Esquema Genérico de la lógica Difusa.

De la figura anterior se detallan las acciones contenidas en cada una de sus etapas:

3.7.1 Fuzzificador

Dado que los datos del vector de entrada son no difusos, el fuzzificador establece una relación para los valores antes mencionados y los correspondientes a los conjuntos difusos, de tal manera que sea posible determinar el grado de pertenencia. Es evidente que todos los valores fuzzificados estarán limitados por el universo en cuestión y cada valor de entrada, se fuzzifica sobre todas las funciones de membresía utilizadas en las reglas difusas.

3.7.2 Reglas y Modelo Takagi-Sugeno

Las reglas difusas pueden ser del tipo IF/THEN, causa-efecto, ya que al combinar uno o varios conjuntos difusos de entrada llamados antecedentes, se les asocia un conjunto difuso de salida, llamado consecuente, esta asociación se lleva a cabo por medio de conectores lógicos como lo pueden ser: o, y, etc. En síntesis, aquí es donde se logra expresar el conocimiento que se ha adquirido sobre una determinada tarea, al tener una amplia experiencia entre los antecedentes y las consecuencias de los mismos; y con lo cual es posible generar un conjunto de reglas sobre cada situación que pueda presentarse durante un proceso. Existen dos formatos populares para la generación de reglas, Mamdani y Takagi-Sugeno (T-S), en el presente trabajo es usado el enfoque de T-S, se encuentra basado en la composición de reglas de inferencia, las cuales cuenta con la parte antecedente y consecuente son proposiciones difusas.

$$\text{Si } x(k) \text{ es } A_i \text{ Entonces } y \text{ es } B_i \quad (3.20)$$

Donde $i = 1, 2, 3, \dots, k$ A y B son los términos lingüísticos definidos por las funciones de membresía y cada una de esas reglas, establece un valor determinado para cada par ordenado (x, y) .

La mezcla de entre un modelo lingüístico y un modelo de regresión matemática de Takagi y Sugeno, [37], producen un modelo T-S, en el cual la parte antecedente describe las regiones difusas en el espacio de entrada y la parte consecuente es una función matemática no difusa del modelo de entrada, esto es:

$$\mathfrak{R}_i : \text{si } x \text{ es } A_i \text{ entonces } y_i = f_i(x), \quad i=1, 2, 3, \dots, k \quad (3.21)$$

Donde normalmente la función es de la misma estructura y son los parámetros los que varían para cada regla propuesta; una parametrización simple y practica puede ser de la siguiente forma, utilizando una forma lineal:

$$\mathfrak{R}_i : \text{si } x \text{ es } A_i \text{ entonces } y_i = a_i^T x + b_i, \quad i=1,2,3,\dots,k \quad (3.22)$$

Donde a_i es un vector de parámetros y b_i es un offset escalar.

En el trabajo de Olivares, [8], se menciona que el modelo Takagi-Sugeno puede ser visto de la siguiente manera: Tal que la parametrización de los elementos a_i y b_i , corresponden a una linealización de una función no lineal, descrita en la relación del espacio de entrada. Una buena linealización está fundada en las regiones del núcleo de la parte antecedente, y para favorecer la parte consecuente, es necesario identificar la forma adecuada de la función de membresía para que la aproximación del modelo sea efectiva.

3.7.3 Defuzzificador

Una vez realizado el proceso de Fuzzificación para obtener un valor asociado y trabajarlo mediante implicaciones difusas a través de las reglas difusas, concretamente el modelo Takagi-Sugeno, es necesario que el sistema regrese un valor no difuso para obtener así un valor para salida, es decir, en esta etapa se toma como entrada el conjunto difuso anteriormente obtenido para dar un valor a la salida. Esta descrito por la siguiente ecuación, proveniente de la Defuzzificación por singleton, [8], cuenta con la forma conveniente para desarrollar esta etapa dentro de un multi-modelo:

$$y = \frac{\sum_{l=1}^m y^l (\mu_{A^l}(x))}{\sum_{l=1}^m (\mu_{A^l}(x))} \quad (3.23)$$

Donde l es la parte consecuente de la regla i .

4. MULTI-MODELO DIFUSO-NEURONAL COMPLEJO

Una vez abordado las partes que componen al tema central a tratar en este trabajo, las Redes Neuronales Complejas y los Sistemas Difusos, al combinarlos es posible obtener una herramienta poderosa para la identificación y control de sistemas no lineales, ya que al hacer una extensión del caso real, el caso complejo añade ciertas propiedades al FNMM, como lo es una mayor convergencia en la estimación de parámetros, ya que los números complejos permiten una mejor representación de las oscilaciones producidas por los sistemas no lineales oscilatorios, así como la reducción de neuronas necesarias para la identificación. Aquí es donde sale a relucir, las ventajas del modelo T-S, ya que permite que la parte consecuente sea sustituida por un CVRNN.

Tomando en consideración las cualidades que pueden presentar la parte difusa, dado que el comportamiento de una planta cambia dependiendo de la región de operación en el que se encuentre, debido a fallas en el sistema, fallas en sensores y actuadores, perturbaciones externas y variaciones en los parámetros. Por lo anterior se puede pensar en dividir el espacio de operación en zonas, esto permite usar diferentes condiciones de operación, y particularizar la identificación o control para cada zona de operación en la que se encuentre la planta.

El modelo de CVRNN que se propone a continuación, es fácil de aplicar para identificación y control de sistemas, ya que se encuentra descrito en la forma de espacios de estados.

Además de su facilidad para incorporarlo en sistema estructurado, esto hace posible que usando solo dos CVRNN con un pequeño recubrimiento sobre una vecindad cercana a cero, las “funciones de membresía” están dispuestas como positivo y negativo. Por lo que la aproximación de la dinámica completa del sistema no lineal puede ser construida con 2 o más CVRNN; por lo que el número de funciones de membresía puede estar dada según sea la complejidad de la naturaleza del sistema.

También es interesante no solo pensar en cambiar las condiciones iniciales del sistema para cada una de las particiones que se tomen de la región de operación, sino someter a diferentes entradas el sistema no lineal, para validar la identificación y control en esas zonas donde exista un *switch* de señal, a través de ello se pueden hacer estudios de estabilidad y convergencia de los algoritmos de aprendizaje.

4.1 Fuzzificación en FNMM

Recordando lo mencionado en el capítulo anterior, la Fuzzificación es la acción de determinar el grado de pertenencia de un valor medido (u) de una variable a estudiar (U), esto es dictaminado por las funciones de membresía y descrito por $\mu_f(u): u \rightarrow [0,1]$, también llamadas funciones de inclusión; pueden ser de tipo trapezoidal, S, campana de Gauss, etc. Para este trabajo se hace uso de la función trapezoidal, y se encuentra descrito por lo siguiente:

$$T(u; a, b, c, d) = \begin{cases} 0 & u < a \\ (u - a) / (b - a) & a \leq u \leq b \\ 1 & b \leq u \leq c \\ (d - u) / (d - c) & c \leq u \leq d \\ 0 & u > d \end{cases} \quad (4.1)$$

Los parámetros a, b, c, d , permiten el ajuste de la función para modificar su forma, esto sirve para hacer recubrimientos entre funciones de membresía lo cual le permitirá tener una mejor eficiencia, al tomar en cuenta un mayor número de información para determinar una salida.

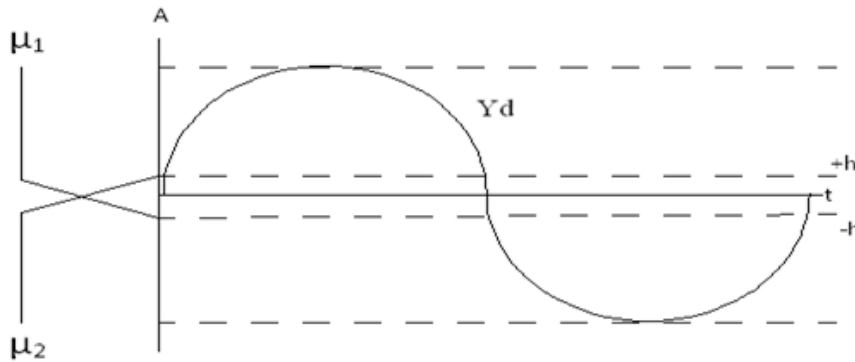


Figura 8. Funciones de Membresía con un recubrimiento en cero

En la figura anterior se muestra, las funciones de membresía de tipo trapezoidal con un pequeño recubrimiento en cero que se utilizan en este trabajo para una función oscilatoria, dado que los parámetros dado el multi-modelo esta descrito en el dominio complejo, el número de funciones de inclusión se reduce debido a las propiedades del número complejo. El que se simplifique en número de funciones de membresía le permite al sistema reducir el computo numérico, dado que el número de reglas e inferencia difusa se ve reducido de igual manera. Dependiendo de la complejidad del sistema puede ser necesario aumentar el número de funciones de membresía para obtener

resultados más precisos y exactos. Es conveniente normalizar el universo en discurso en un rango de $[-1,1]$, sin embargo, no es necesario. Con base en lo anterior las funciones de membresía son descritas por:

$$\mu_N(u) : u \rightarrow [-1, h] \quad (4.2)$$

$$\mu_P(u) : u \rightarrow [-h, 1] \quad (4.3)$$

Donde el termino h , es un valor pequeño para que las funciones de membresía tengan un traslapamiento pequeño en cero. Esto es debido a que los trabajos realizados anteriores a este fueron un tanto ineficaces aplicando una sola red neuronal en la zona cercana a cero. Ahora bien, si se aprovecha el enfoque difuso, permitirá aprender esta región más de una teniendo como resultado una mejor estimación global de la dinámica deseada a identificar; todo esto tiene una dependencia del valor h . La manipulación de los factores a , b , c , d , que se mencionaron para la construcción de las funciones de inclusión, no es necesario ya que la red neuronal, se encargara de todo espectro que este contenido en la región $[-1, h]$ y $[-h, 1]$.

4.2 Inferencia Difusa en FNMM

La inferencia difusa es el arte de interpretar los valores obtenidos en la parte de fuzzificación para deducir una consecuencia a partir de un conjunto de reglas propuestas; lo anterior abre la posibilidad a que existan diferentes maneras de hacer esta tarea, como lo pueden ser: Modelos Lingüísticos, Modelos Fuzzy Relacionales, Modelos Takagi-Sugeno. El primero es conocido como el Modelo de Mamdani, y su principal característica es que tanto antecedente como el consecuente son proposiciones difusas. El segundo, se puede explicar como una extensión del primero, donde el mapeo entre la entrada y la salida de la inferencia difusa se expresa a través de una relación difusa, en otras palabras, existe un conjunto de reglas entre la fuzzificación de los términos lingüísticos de entrada-salida, [8].

En el tercero, el modelo Takagi-Sugeno, surge a través y consecuencia de la evolución de los métodos, retomando lo descrito en el capítulo 3, es la combinación de un modelo lingüístico y un modelo de regresión matemática T-S; dando así la oportunidad de describir la parte antecedente de la regla como regiones difusas en el espacio de entrada. Y la parte consecuente como como una función matemática no difusa del modelo de entrada. En, [6], proponen usar un método de

aproximación de sistemas dinámicos, a través de un modelo neuro-difuso, representado por una red dinámica neuro-difusa, en donde la estructura del arreglo es tipo T-S. Baruch et. Al., [8], [10], han seguido esa línea de investigación de forma creativa, obteniendo buenos resultados, sin embargo, la eficiencia obtenida se puede mejorar al incluir los números complejos en los procesos de aprendizaje, debido a las características antes mencionadas que añaden como propiedades a los sistema de identificación y control de plantas no lineales oscilatorias. Una de las características importantes es que el número de funciones de membresía se verá reducido dado el buen desempeño de las redes recurrentes neuronales con valores complejos.

Dicho lo anterior, al incluir en la parte consecuente una CVRNN, se obtiene que la aproximación queda descrita en espacio de estados, lo cual hace fácil su implementación en un sistema estructurado de control adaptable e identificación y es posible describirlo como un modelo Takagi-Sugeno con consecuencia neuronal y se encuentra descrito por:

$$\mathfrak{R}_i : \text{si } U(k) \text{ es } A_i \text{ entonces } y_i = N_i[x_i(k), u_i(k)], \quad i=1,2,3,\dots,k \quad (4.4)$$

Con lo anterior que establecido usar el modelo de la CVRNN para así formar un multi-modelo neuronal, la estructura de dicho modelo se presenta a continuación. Cabe destacar que el uso de tiempo discreto favorece el método, ya que al poderlo asociar a puntos en el tiempo esto le permite usar el pasado y o el futuro, para predecir el presente, es decir, que cuando se requiere calcular un evento del presente se pueden hacer ciertas suposiciones con el fin de que se obtengan valores aproximados, para realizar un ajuste.

4.3 Topología de CVRNN

La topología en consideración para el caso complejo es presentada en, [14] [17], y ese en una extensión de [19]. La topología es afectada por la estructura jerárquica del FNMM, por lo que es necesario precisar las topologías de cada uno de los siguientes casos.

Se considera el modelo discreto basado en la forma canónica de Jordán para la CVRNN definida en su forma matricial como:

$$X(k+1) = Ax(k) + BU(k) \quad (4.5)$$

$$Z(k) = \Phi[X_{\text{Re}}(k)] + i \cdot \Phi[X_{\text{Im}}(k)] \quad (4.6)$$

$$V(k) = C \cdot Z(k) \quad (4.7)$$

$$Y(k) = \Gamma[V_{\text{Re}}(k)] + i \cdot \Gamma[V_{\text{Im}}(k)] \quad (4.8)$$

De la subsección 2.4 se sabe que la función de activación debe ser construida para que, de esta manera, cumpla con los requisitos de ser holomorfa, acotada y sin puntos de singularidad, por lo que queda descrita por la siguiente ecuación:

$$f(z) = \tanh[\text{Re}(k)] + i \cdot \tanh[\text{Im}(k)] \quad (4.9)$$

Es conveniente hacer una aclaración. Cada vez que se multiplican matrices complejas se produce el siguiente resultado, se tomara como ejemplo el producto que da como resultado $V(k)$,

$$V(k) = \beta(k) + i \cdot (\delta(k)) + i^2(\gamma(k)) \quad (4.10)$$

Donde β es real, $i(\delta)$ es imaginario y γ es real por el término $i^2 = -1$.

Dado lo anterior es posible escribir lo siguiente:

$$V(k) = V_{\text{Re}}(k) + i \cdot V_{\text{Im}}(k) \quad (4.11)$$

Es posible observar que el termino γ , con cada iteración añadirá dinámica a la salida de cada CVRNN, por lo que con esto se cumple una propiedad importante de las redes neuronales complejas, que es la rápida convergencia frente a la realiza por el caso real.

Recordado que el error para el HFNNM es la diferencia que existe entre la salida deseada, es decir, la dinámica de la planta y la salida obtenida de la CVRNN, por lo que queda descrita por la siguiente forma:

$$E(k) = \text{Re}[Yd(k) - Y(k)] + i \cdot \text{Im}[Yd(k) - Y(k)] \quad (4.12)$$

De una manera compacta,

$$E(k) = E_{\text{Re}}(k) + i \cdot E_{\text{Im}}(k) \quad (4.13)$$

Las dimensiones de los vectores y matrices de la topología de la CVRNN están dados por lo siguiente: $A \in C^{n \times n}$, es la matriz de pesos de realimentación, $B \in C^{n \times m}$, es la matriz de pesos de entrada, $X \in C^n$ es un vector interno, $Z \in C^n$ es un vector oculto, $U \in \mathbb{R}^m$ es la entrada de la red

neuronal, $Y \in C^p$ es la salida de la red neuronal, $\Gamma[\cdot]$ y $\Phi[\cdot]$ son funciones de activación $\tanh(\cdot)$. El número de entradas es “m”, el número de neuronas ocultas es “n” y el número de salidas es “p”. La siguiente expresión es condición que preserva estabilidad y es impuesta en la matriz A diagonal a bloques.

$$A = \text{diag}(A_i) \quad |A_i| < 1 \quad i = 1, \dots, N \quad (4.14)$$

Retomando la ecuación 4.4 y sustituyendo la parte consecuente por las ecuaciones 4.5-4.8, se obtiene que:

$$\begin{aligned} \mathfrak{R}_i : \text{sí } U(k) \text{ es } A_i \text{ entonces} \\ X(k+1) &= Ax(k) + BU(k) \\ Z(k) &= \Phi[X_{\text{Re}}(k)] + i \cdot \Phi[X_{\text{Im}}(k)] \\ V(k) &= C \cdot Z(k) \\ Y(k) &= \Gamma[V_{\text{Re}}(k)] + i \cdot \Gamma[V_{\text{Im}}(k)] \end{aligned} \quad (4.15)$$

Compactando la ecuación anterior:

$$\mathfrak{R}_i : \text{sí } U(k) \text{ es } A_i \text{ entonces } y_i = \text{CVRNN}_i, \quad i=1,2,3,\dots,k \quad (4.16)$$

Como se ha mencionado, la aproximación utilizando la lógica difusa depende del número de reglas y funciones de membresía para obtener buenos resultados, aquí se demuestra que para dos ejemplos de plantas no lineales oscilatorias basta con $i=2$, y un pequeño recubrimiento en la región cero. Con todo lo expuesto, ahora es posible nombrar al nivel jerárquico, tal y como se propuso en, [8]:

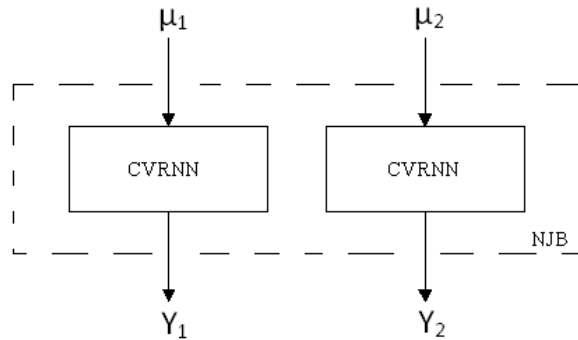


Figura 9. Nivel Jerárquico Bajo

La anterior figura muestra el Nivel Jerárquico Bajo, y utilizando el enfoque de un sistema difuso, se plantean dos CVRNN para la parte consecuente de un Multi-Modelo Jerárquico, por lo tanto, es posible nombrarlo como un conjunto de CVRNNs independientes, en las cuales se tiene un número determinado de neuronas en la capa oculta para realizar un trabajo aceptable. Los números complejos tienen un impacto en estas neuronas ocultas ya que la neurona está compuesta por una parte real y una parte imaginaria, teniendo así el doble de parámetros y en consecuencia el número de neuronas se puede ver reducido debido a esta propiedad.

4.4 Defuzzificación de FNMM

Recordando que el proceso de Defuzzificación consiste en transformar un valor obtenido de la inferencia difusa y la Fuzzificación, en un valor no difuso, y desde el punto de vista de la ecuación 3.23, es decir una suma ponderada, bien se podría describir como un multi-modelo neuro-difuso, [8]. Si esto lo transformamos a un enfoque multi-modelo difuso-neuronal-jerárquico, el cual se revisará de manera formal más adelante, en la sub-sección anterior, ya se habló de un Nivel Jerárquico Bajo, ahora se desarrollará el Nivel Jerárquico Alto, y ambos establecerán el Multi-Modelo Difuso-Neuronal Jerárquico; retomando este enfoque la Defuzzificación quedara denotada por una CVRNN, al ser una extensión del caso real, no modifica que la CVRNN ofrece la relación de la salida con respecto de las funciones y_i entradas provenientes de la ponderación de cada regla de forma independiente, mostrando una similitud al modelo T-S, se omitirá el tratamiento matemático ya que la extensión al caso complejo supone un incremento en los parámetros internos de las variables descritas en ese trabajo, al tener parte real y parte imaginara, obteniendo lo siguiente, [8]:

$$Y = \sum_{i=1}^m W_i y_i \quad (4.17)$$

Donde W_i :

$$W_i = \sum_{j=1}^m \frac{C_i B_{ij}}{(zI - A_i)} \quad (4.18)$$

De tal manera que Nivel Jerárquico Alto, esta descrito por la siguiente figura:

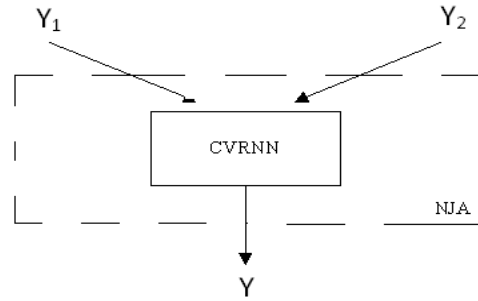


Figura 10. Nivel Jerárquico Alto.

4.5 Descripción de HFNMM

Por sus singlas en ingles HFNMM, significa Multi-Modelo Difuso-Neuronal Jerárquico, a continuación, se presenta su estructura:

El HFNMM, está compuesto por: Fuzzificador, Reglas Difusas de inferencia a través de las reglas de Takagi-Sugeno (T-S) y Defuzzificador. En el Fuzzificador, la entrada de la planta se normaliza y se divide en “Positivo” y “Negativo”, debido a las funciones de membresía. Las reglas T-S son una combinación del modelo lingüístico de Mamdani, y modelos de regresión matemática. Está compuesto de antecedente y consecuente, y representado por una oración IF / THEN. Recordando la ecuación 4.16 y poniendo como limite a $i=2$:

$$\mathfrak{R}_i : \text{si } U(k) \text{ es } M_i \text{ entonces } y_i(k) = \text{CVRNN}_i, i=1,2 \quad (4.19)$$

Ya que el índice 1 corresponde a la parte “Positiva” y el índice 2 corresponde a la parte “Negativa”, entonces $U(k)$ es la entrada, M_i es la función de membresía (Positiva y / o Negativa) que está definida por conjuntos difusos (la parte consecuente está dada por $y_i(k)$). La salida se calcula mediante CVRNN-1 y/o CVRNN. Cabe remarcar que el operador anterior “o” se debe a que se consideró un pequeño recubrimiento sobre la región cero, esto produce un traslapamiento de salidas, y por tanto dos valores de entrada para el NJA.

Cada modelo de espacio de estado podría aprenderse mediante un error de aproximación local:

$$E_i(k) = Y_d(k) - Y_i(k) \quad (4.20)$$

Donde Y_d es la dinámica deseada y Y_i es la dinámica obtenida a través de la aproximación.

La CVRNN tiene una topología que permite calcular la salida Y_i está dada por las ecuaciones 4.5-4.8, las cuales definen la salida local, este procedimiento será matemáticamente denotado por $Y = \Pi (L, M, N, Y_d, U, X, A, B, C, E)$ y reemplaza la parte consecuente de la ecuación 4.19:

$$\mathfrak{R}_i : \text{si } U(k) \text{ es } M_i \text{ entonces } y_i(k) = \Pi (L, M, N, Y_d, U, X, A, B, C, E) \quad (4.21)$$

La salida del HFNMM, que aproxima a la planta no lineal, es una suma ponderada de las salidas de los modelos CVRNN, que aparece en la parte consiguiente 4.17, y viene dada por la siguiente ecuación:

$$Y = \sum_{i=1}^2 W_i y_i \quad (4.22)$$

Las ponderaciones dependen de la forma de las funciones de membresía. Proponemos realizar la defuzzyfication a través de otra CVRNN, donde las entradas son $Y_i (k)$. Por lo tanto, el resultado de la identificación del sistema viene dado por:

$$Y(k) = \Pi (L, M, N, Y_d, Y_o, U, X, A, B, C, E) \quad (4.23)$$

Donde, el vector de entrada Y_o se forma a partir de los vectores $y_i (k)$, el error de aprendizaje $E = Y_d - Y$ y $\Pi (\cdot)$ es un procedimiento de aprendizaje CVRNN, su aprendizaje ha sido probado con dos métodos diferentes el *Back-Propagation* y el *Levenberg-Marquardt*.

La figura, presenta un resumen esquemático de la descripción del HFNMM. En este caso, solo usó dos inferencias basadas en reglas difusas de Takagi-Sugeno.

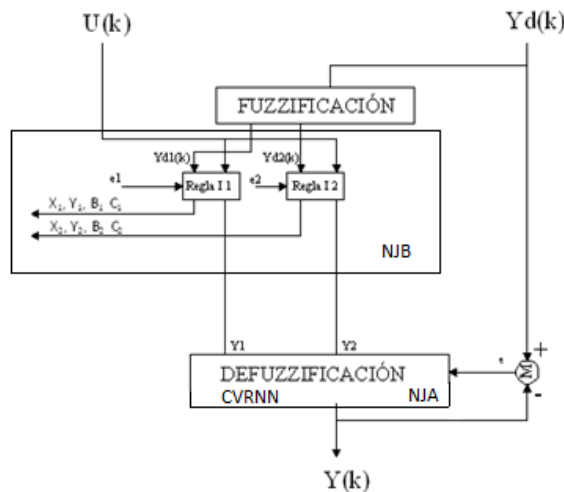


Figura 11. Diagrama Detallado de HFNMM.

4.6 Algoritmos de Aprendizaje

Recordando que al utilizar en este trabajo con Redes Neuronales Entrenables, los algoritmos de entrenamiento tienen un papel fundamental para la evolución y desempeño de las mismas. El proceso de aprendizaje de una red neuronal se puede entender como la capacidad de ajustar los pesos sinápticos para que su relación entre el medio ambiente (dinámica deseada) y la aproximación hecha por la misma tenga un índice de error de prácticamente cero, ya que analizado desde un punto de vista práctico el error nunca va ser cero. En la tesis presentada del trabajo, [14], se presentan los tratamientos para obtener un algoritmo de primer orden en el dominio complejo, Back-Propagation, el cual está basado en el gradiente descendiente. De la misma manera en, [17], se desarrolla el algoritmo de aprendizaje en el dominio complejo recursivo, a partir de algoritmos de segundo orden, Levenberg-Marquardt. Ambos algoritmos desarrollan una red adjunta, [38], para poder encontrar el ajuste correspondiente para que error sea menor a cada iteración; esto fue logrado a través de las reglas diagramáticas, [16], las cuales son una herramienta de teoría graficas.

4.6.1 Back-Propagation

El algoritmo de aprendizaje en tiempo real de CVRNN con término momentum se encuentra dado por la siguiente ecuación

$$W(k+1) = W(k) + \eta \cdot \Delta W(k) + \alpha \cdot \Delta W(k-1) \quad (4.24)$$

Donde: W es una matriz de peso general (A, B, C), ΔW es el cambio en W , η es una constante de aprendizaje, y α es una constante de un término de momentum.

La siguiente desigualdad es una variación de la propuesta en [10], en el presente trabajo se propone que el peso hacia delante sea restringido con respecto al peso anterior:

$$|W(k+1) < W(k)| \quad (4.25)$$

En [16], se aplicaron las reglas esquemáticas que permiten obtener el CVRNN adjunto en el dominio complejo, utilizado esto es posible reescribir el siguiente algoritmo de actualización de pesos:

$$E_1(k) = \phi'[\text{Re}(Y(k))] \cdot E_{\text{Re}}(k) + i \cdot \phi'[\text{Im}(Y(k))] \cdot E_{\text{Im}}(k) \quad (4.26)$$

$$\Delta C(k) = E_1(k) \cdot Z(k) \quad (4.27)$$

$$E_2(k) = C^*(k) \cdot E_1(k) \quad (4.28)$$

$$E_3(k) = \Gamma'[\text{Re}(Z(k))] \cdot E_{2\text{Re}}(k) + i \cdot \Gamma'[\text{Im}(Z(k))] \cdot E_{2\text{Im}}(k) \quad (4.29)$$

$$\Delta A(k) = E_3(k) \cdot X(k) \quad (4.30)$$

$$\Delta B(k) = E_3(k) \cdot U(k) \quad (4.31)$$

La transposición conjugada se denomina por *.

Los parámetros de aprendizaje del algoritmo deben ser restringidos por lo siguiente:

$$\eta > 0 \quad \& \quad \alpha < 1 \quad (4.32)$$

4.6.2 Levenberg-Marquardt

El general CVRNN algoritmo de aprendizaje en tiempo real LM, [[18]], es dado por:

$$W(k+1) = W(k) + P(k) \cdot DY[W(k)] \cdot E(k) \quad (4.33)$$

Con la misma restricción que el algoritmo BP, ecuación 4.25.

A partir de la red adjunta, los términos de gradiente se derivan y se necesitan para los pesos de actualización, que se describen mediante las siguientes ecuaciones:

$$D_1(k) = (\phi'[\text{Re}(Y(k))] + i \cdot \phi'[\text{Im}(Y(k))]) \cdot I \quad (4.34)$$

$$D_2(k) = (\Gamma'[\text{Re}(Z'(k))] \cdot \text{Re}(C') + i \cdot \Gamma'[\text{Im}(Z'(k))] \cdot \text{Re}(C')) D_1(k) \quad (4.35)$$

$$DY[C(K)] = D_1(k) \cdot Z(k) \quad (4.36)$$

$$DY[A(K)] = D_2(k) \cdot X(k) \quad (4.37)$$

$$DY[B(K)] = D_2(k) \cdot U(k) \quad (4.38)$$

La matriz P se calcula recursivamente utilizando la siguiente ecuación de Riccati:

$$P(k) = \alpha^{-1}(k) [P(k-1) - P(k-1) \cdot \Omega_{W(k)} \cdot S_{W(k)}^{-1} \cdot \Omega_{W(k)}^* \cdot P(k-1)] \quad (4.39)$$

Donde las matrices Ω_W y S_W son dadas por:

$$\Omega_{W(k)}^* = \begin{bmatrix} DY^*[W(k)] \\ 0 \dots 1 \dots 0 \end{bmatrix} \quad (4.40)$$

$$S_{W(k)} = \alpha(k) \cdot \Lambda + \Omega_{W(k)}^* \cdot P(k-1) \cdot \Omega_{W(k)} \quad (4.41)$$

$$\Lambda^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & \rho \end{bmatrix} \quad (4.42)$$

De nuevo, las matrices P y S_W tienen dimensiones ($N_W \times N_W$) y (2×2) respectivamente, donde N_W es el número de pesos en el vector W . La matriz Ω_W tiene dimensión ($N_W \times 2$), la segunda fila de Ω_W^* de ($N_W - 1$) ceros y una unidad en la i -ésima posición calculada por:

$$i = k \cdot \text{mod}(N_W) + 1 \quad (4.43)$$

Los parámetros del algoritmo deben ser restringidos por lo siguiente:

$$\begin{aligned} 10^{-6} &\leq \rho \leq 10^{-4} \\ 0.97 &\leq \alpha(k) \leq 1 \\ 10^3 &\leq P(0) \leq 10^6 \end{aligned} \quad (4.44)$$

En el presente trabajo, se encuentra el parámetro $\alpha(k)$ para cada instante k , basado en la obtención del menor error posible en el intervalo α .

4.7 Convergencia y Estabilidad

En [19, 22], se discuten las pruebas de estabilidad y convergencia en la topología, la HFNMM y los algoritmos de aprendizaje, en el dominio real, actualmente estamos trabajando en las pruebas de estabilidad y convergencia para el caso complejo.

Consideramos que la manera de hacer las pruebas de estabilidad y convergencia para el HFNMM en un dominio complejo es seguir las presentadas para el caso real; sin embargo, teniendo en cuenta la sección I, las redes neuronales en un dominio complejo, lo más importante en la topología es la función de activación que necesita ser un holomórfico, limitado y sin puntos de singularidad en su dominio.

Tomando en consideración las siguientes definiciones y teoremas, [39]:

- Definición: Sea f una función de un dominio D y $M > 0$, con $M \in \mathbb{R}$. Se dice que f está limitado si $|f(x)| \leq M$ para cualquier punto $x \in D$.
- Teorema de Liouville: Para todas las funciones holomorfas complejas f , si f está limitado en todo \mathbb{C} , y f es función constante en todo \mathbb{C} .
- Teorema "Pequeño" de Picard: Si una función compleja f es entera y no constante, entonces el conjunto de valores que puede tomar $f(z)$ es el plano complejo \mathbb{C} menos un punto.
- Definición: Si una función f no es holomorfa en un punto z_0 , pero si es para cualquier punto en la vecindad de z_0 , entonces z_0 es un punto de singularidad.

Por lo tanto, una función no puede ser holomorfa y limitada al mismo tiempo y no debe tener puntos de singularidad. Si la función de activación no cumple con esto probablemente el sistema nunca convergería.

La función de activación, ecuación 4.9, es una extensión de la función de activación real y, además, construida, resulta holomorfa, delimitada y sin puntos de singularidad en su dominio. Por su sencillez y capacidad de manejar los parámetros de amplitud y fase de señal compleja, esta función de activación utilizada en el dominio complejo red neuronal, [33].

5. IDENTIFICACIÓN Y CONTROL CON HFNMM

En las sub-secciones siguientes se presentan dos plantas no lineales oscilatorias, de primer y segundo orden, las cuales fueron tomadas como ejemplos para aplicar el HFNMM, en los problemas de identificación y control.

5.1 Descripción de la Planta de Primer Orden

El sistema identificado es un modelo no lineal idealizado de un brazo robótico de articulación flexible, con DOF (Barouh, 2015), [14], ilustrado por la Fig. El modelo de la articulación del robot consiste en un actuador conectado a una carga a través de un muelle de torsión que representa la flexibilidad de la articulación, y este efecto (la flexibilidad) es generado por un accionamiento armónico, que es un tipo de mecanismo de engranaje de robot con alta transmisión de par, y tamaño compacto.

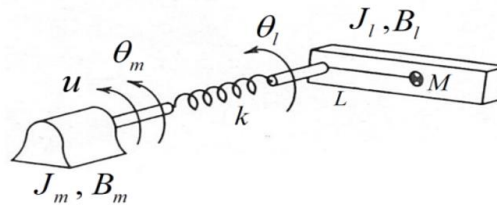


Figura 12. Planta idealizada de un grado de libertad.

Aquí, el par del motor como la entrada u de una planta. Las ecuaciones de movimiento del robot de articulación flexible se dan de la siguiente manera:

$$\begin{aligned} J_l \ddot{\Theta}_l + B_l \dot{\Theta}_l + Mgl \cdot \sin(\Theta_l) + k(\Theta_l + \Theta_m) &= 0 \\ J_m \ddot{\Theta}_m + B_m \dot{\Theta}_m - k(\Theta_l + \Theta_m) &= u \end{aligned} \quad (5.1)$$

Donde: J_l, J_m son inercias de carga y de motor; B_l, B_m son constantes de amortiguación de la carga y del motor; u es el par de entrada aplicado al eje del motor; M y L son la masa del eslabón y la longitud entre el eje y el centro de masa del eslabón; k representa la constante de rigidez de torsión del engranaje impulsor armónico.

El sistema es sub-actuado, porque la planta se describe mediante dos ecuaciones diferenciales de segundo orden, que representan un sistema con dos grados de libertad, pero sólo una entrada.

Donde los parámetros utilizados para esta planta no lineal oscilatoria, representada en su forma idealizada por la figura 12, se presentan en la siguiente tabla:

Tabla 3. Tabla de parámetros de la planta de un grado de libertad.

Parámetros	Magnitud	Unidades
J_l	1	[kg·m]
J_m	1	[kg·m]
B_l	0.1	[kg·m·s ⁻¹]
B_m	0.1	[kg·m·s ⁻¹]
k	0.1	[kg·m·s ⁻²]
M	0.5	[kg]
L	0.5	[m]
g	9.81	[m·s ⁻²]

5.2 Descripción de la Planta de Segundo Orden

Para tener un mejor entendimiento y alcance del sistema se propone una segunda planta que consiste en un brazo robótico idealizado con dos grados de libertad, (Barouh, 2015), [17].

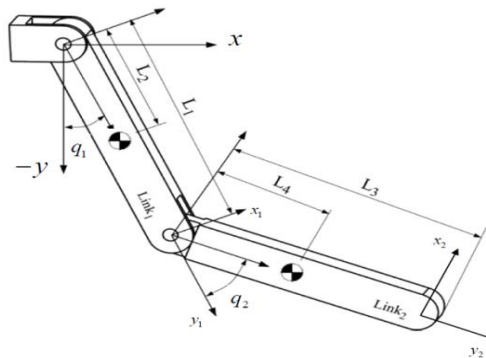


Figura 13. Planta idealizada de dos grados de libertad.

Cada junta o articulación consiste en un actuador conectado, a una carga por medio de un resorte torsional, esto representa la flexibilidad de la articulación. Se respetan las suposiciones hechas en, [18], para obtener un modelo matemático simplificado y se presentan a continuación:

- El momento de inercia del rotor de cada articulación es simétrico es simétrico respecto a su eje rotación. Por lo que el centro de masa de cada eslabón, la velocidad del rotor, así como los términos gravitacionales, son independientes de la posición del motor, [18].
- El movimiento es estrictamente de rotación respecto a su marco inercial, y la energía cinética actuando sobre el rotor de cada eslabón se debe únicamente a su rotación, [18].

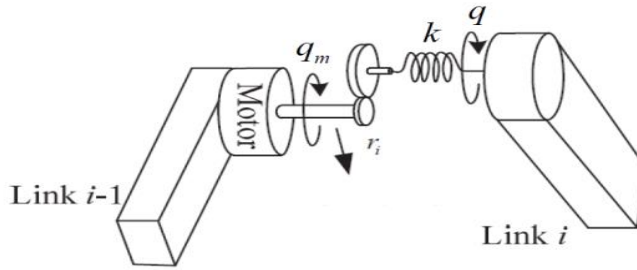


Figura 14. Planta de dos grados de libertad.

Se toma a las posiciones angulares y el par de los motores de cada articulación, como señales de entrada $u(t)$ y salida $y(t)$.

Las ecuaciones que regirán la dinámica del sistema antes mencionado se encuentran dadas por, [18]:

$$\begin{aligned} D(\Theta_l)\ddot{\Theta}_l + C(\Theta_l, \dot{\Theta}_l)\dot{\Theta}_l + G(\Theta_l) + K(\Theta_l - \Theta_M) &= 0 \\ J_M \ddot{\Theta}_M - K(\Theta_l - \Theta_M) &= u \end{aligned} \quad (5.2)$$

Donde Θ_m y $\Theta_l, \in \mathbb{R}$ representan las posiciones angulares en forma vectorial, ya que son vistas desde el motor y el eslabón respectivamente. De las articulaciones $D: \mathbb{R} \rightarrow \mathbb{R}^{n \times n}$ es una matriz definida positiva en la cual sus elementos representan los términos inerciales, de los eslabones. $C: \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$ contiene todos los términos de Coriolis y de fuerzas centrífugas que actúan sobre la dinámica de la planta. En $G: \mathbb{R}^n \rightarrow \mathbb{R}^n$ se presentan los términos gravitacionales. $K: \mathbb{R} \rightarrow \mathbb{R}^{n \times n}$ es una matriz definida positiva diagonal, que contiene los coeficientes de rigidez de los resortes de cada junta. $J_m \in \mathbb{R}^{n \times n}$ es una matriz definida positiva donde se presentan todos los términos inerciales de los motores de cada articulación, por último $u \in \mathbb{R}^n$ es el vector de pares de torsión de entrada.

Tomando como vector extendido:

$$\Theta_f = \begin{bmatrix} \Theta_l \\ \Theta_m \end{bmatrix} \quad (5.3)$$

Es posible escribir de forma matricial a 6.2, de la siguiente forma:

$$D_f(\Theta_l) = \begin{bmatrix} D(\Theta_l) & 0 \\ 0 & J_m \end{bmatrix} \quad (5.4)$$

$$C_f(\Theta_l, \dot{\Theta}_l) = \begin{bmatrix} C(\Theta_l, \dot{\Theta}_l) & 0 \\ 0 & 0 \end{bmatrix} \quad (5.5)$$

$$G_f(\Theta_l) = \begin{bmatrix} G(\Theta_l) \\ 0 \end{bmatrix} \quad (5.6)$$

$$K_f = \begin{bmatrix} K & -K \\ -K & K \end{bmatrix} \quad (5.7)$$

$$u_f = \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad (5.8)$$

Por lo anterior, la planta es un sistema dinámico sub-actuado, no lineal, oscilatorio descrito por cuatro ecuaciones diferenciales ordinarias de segundo orden MIMO por su sigla en inglés (Múltiples entradas y Múltiples salidas).

Donde los parámetros utilizados para esta planta no lineal oscilatoria con dos grados de libertad, cabe mencionar que dicha planta es presentada en la forma idealizada por la figura 14, y los datos recopilados en las tablas 3 y 4 son los usados en el trabajo [17], se presentan en la siguiente tabla:

Tabla 4. Tabla de parámetros de la planta con dos grados de libertad.

Parámetros	Magnitud	Unidades
J_{L1}, J_{L2}	0.5	[kg·m]
J_{M1}, J_{M2}	0.2	[kg·m]
B_{L1}, B_{L2}	0.1	[kg·m·s ⁻¹]
B_{M1}, B_{M2}	0.1	[kg·m·s ⁻¹]
k_1, k_2	0.1	[kg·m·s ⁻²]
M_1, M_2	1	[kg]
L_1, L_2	1	[m]
g	9.81	[m·s ⁻²]

5.3 Resultados de Identificación

El problema de identificación es uno de los tres grandes problemas de la teoría del control automático, por lo que existen diferentes técnicas que permiten encontrar una aproximación de la dinámica de un modelo de una planta desconocida, bajo ciertas condiciones iniciales y señales de entrada. La aplicación de las Redes Neuronales es bastante recurrente para este tipo de tareas ya que no necesita de suposiciones fuertes para hacer la tarea mencionada.

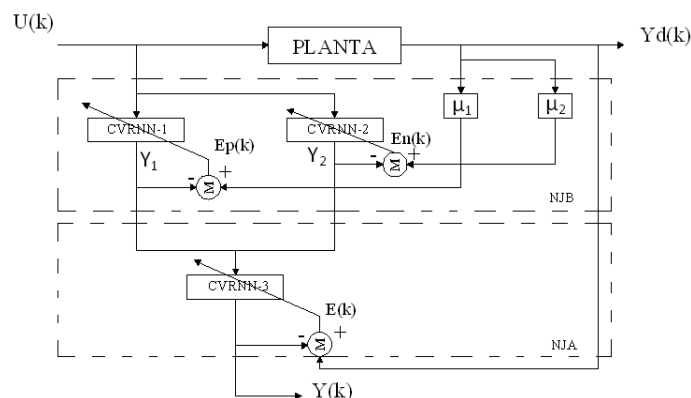


Figura 15. HFNMM para Identificación.

En la figura 15, se presenta el esquema de identificación, con dos niveles jerárquicos, y dos funciones de membresía (Positivo y Negativo).

Cabe recordar que para poder utilizar el HFNMM las señales continuas de entrada y salida se discretizan con un tiempo de muestro (τ).

El error medio cuadrático instantáneo es el índice de desempeño que toma como referencia para evaluar el trabajo realizado por la el HFNMM y se encuentra dado por la siguiente ecuación

$$MSE(k) = \frac{1}{2} (\bar{E}(k) \cdot E(k)) \quad (5.9)$$

Se escogió este parámetro ya que nos interesa saber cuál es el comportamiento de la aproximación con respecto a la dinámica deseada.

5.3.1 Identificación en una planta de un grado de libertad

Al aplicar el esquema de identificación propuesto por la figura 12, utilizando los métodos de aprendizaje *Back-Propagation* y *Levenberg-Marquardt*, además con un tiempo de muestro de $\tau = 0.01$.

Tabla 5. Tabla de parámetros para las CVRNN utilizadas para la identificación de una planta de un grado de libertad

<i>CVRNN</i>	n	p	m
1	4	1	1
2	4	1	1
3	4	1	2

Como entrada para el esquema de identificación se usó una señal senoidal cuya ecuación está dada por:

$$u_i(t) = \text{sen}\left(\frac{1}{10} \cdot t\right) + 0.5 \cdot \text{sen}\left(\frac{1}{25} \cdot t\right) \quad (5.10)$$

Obteniendo lo siguientes resultados:

5.3.1.1 Resultados *Back-Propagation*

En donde los parámetros utilizados durante la simulación son: $\eta = 0.3$ y $\alpha = 0.01$, cabe recordar que para el método propuesto en las sub-sección 4.6, las condiciones iniciales deben ser complejas en una vecindad de radio σ cercano a cero en el plano complejo, y no mayor a un radio de 0.5.

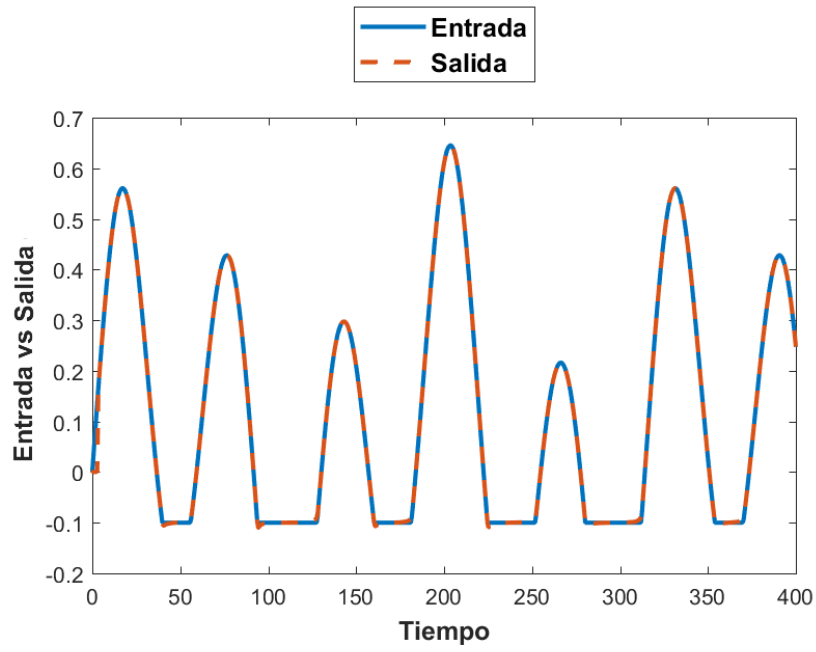


Figura 16. Parte Positiva de Identificación (BP).

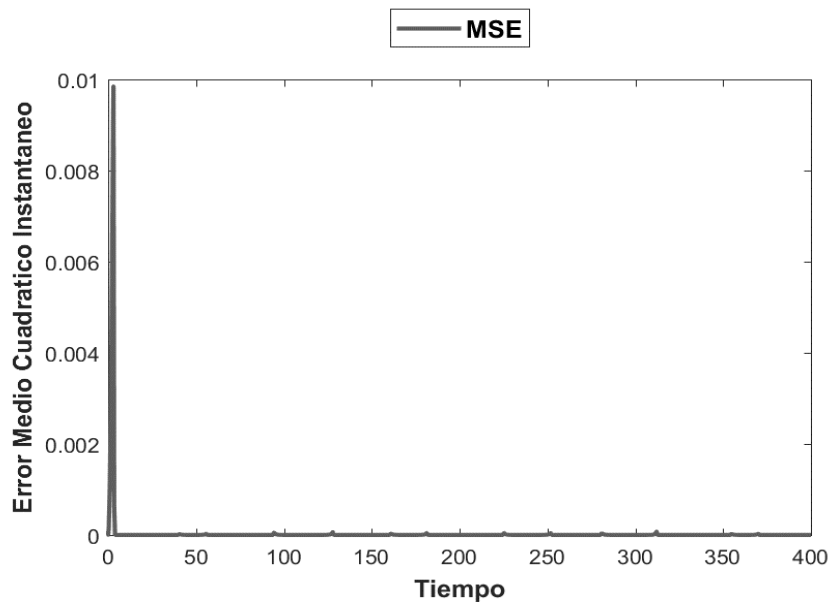


Figura 17. MSE Instantáneo de la Parte Positiva de Identificación (BP).

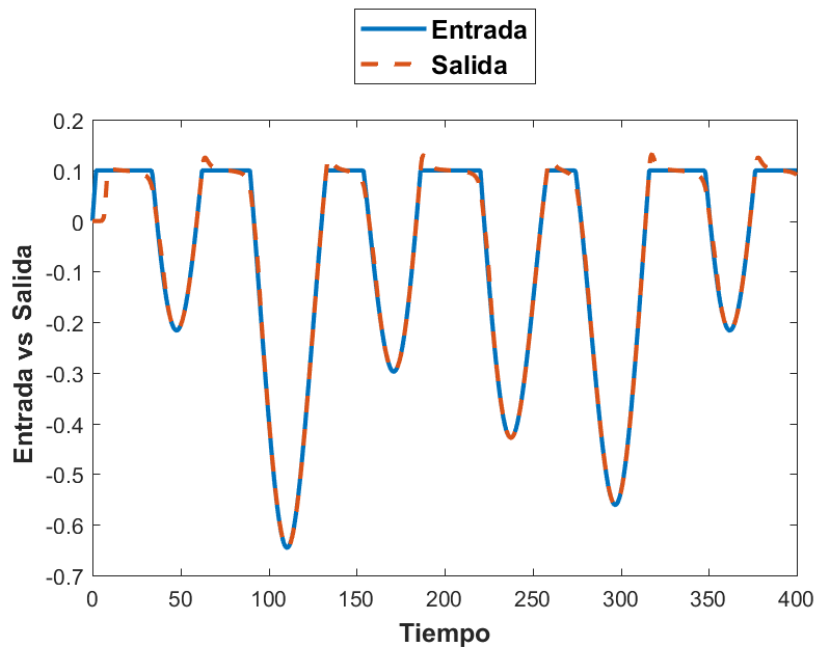


Figura 18. Parte Negativa de Identificación (BP).

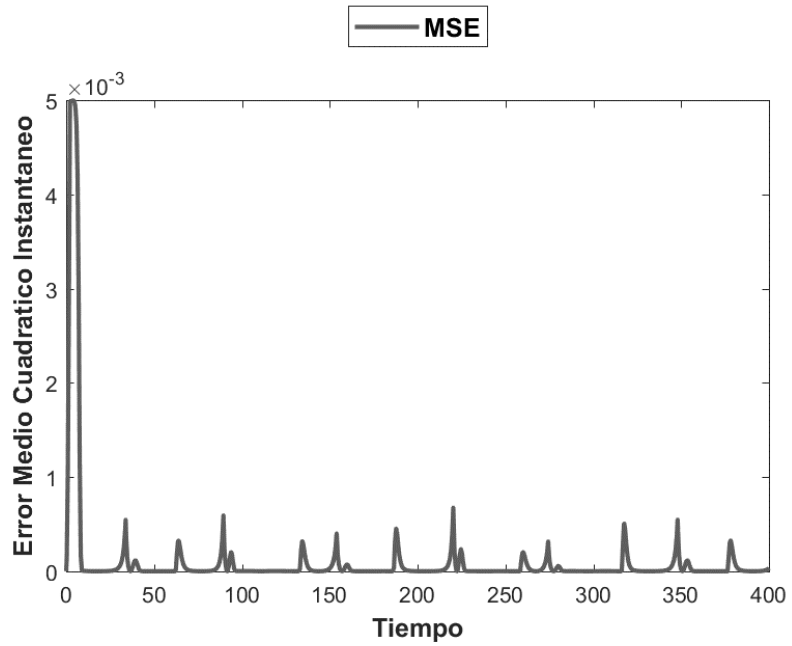


Figura 19. MSE Instantáneo de la Parte Negativa de Identificación (BP).

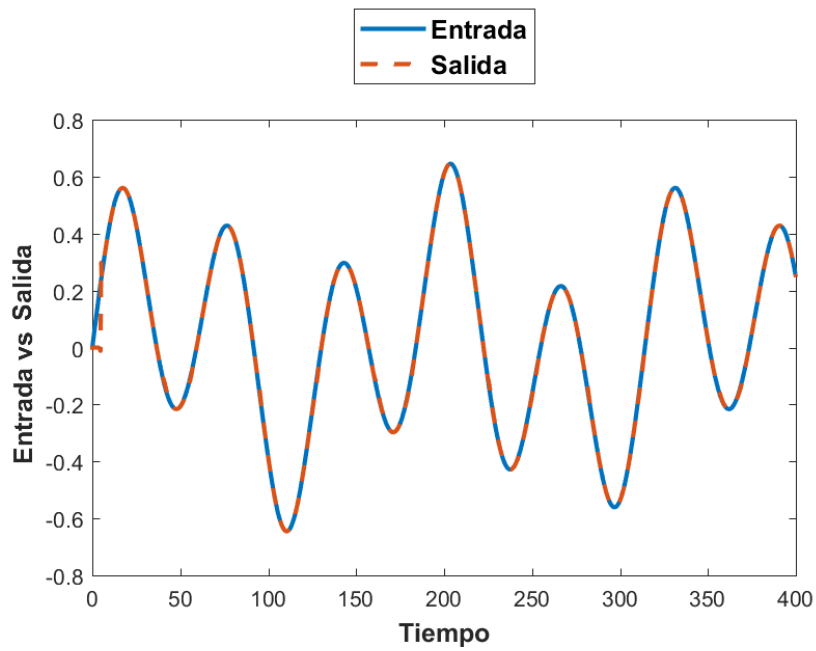


Figura 20. Señal Completa y de Identificación (BP).

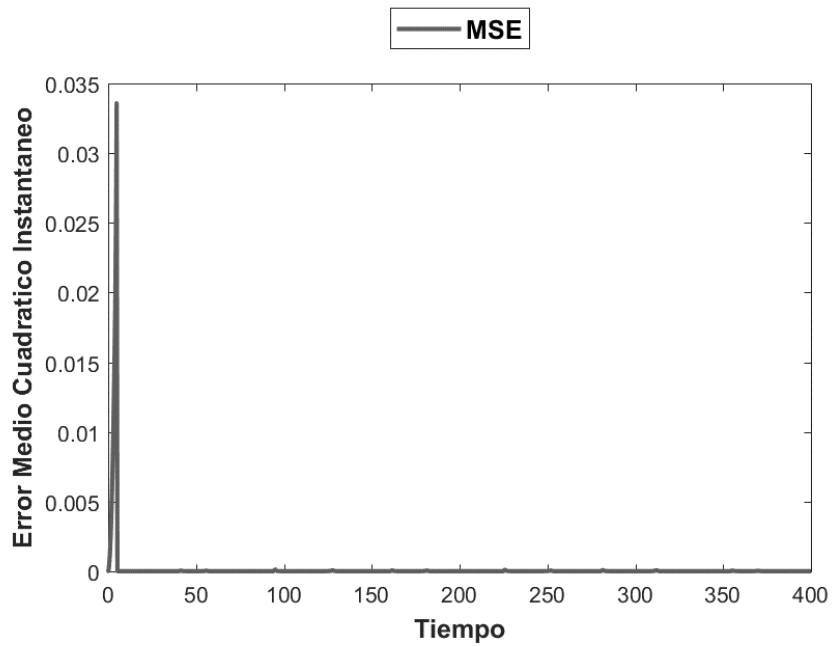


Figura 21. MSE Instantáneo de Identificación (BP).

5.3.1.2 Resultados Levenberg-Marquardt

En donde los parámetros utilizados durante la simulación son: $\eta = 0.3$ y $\alpha = 0.01$, cabe recordar que para el método propuesto en las sub-sección 4.6, las condiciones iniciales deben ser complejas en una vecindad de radio σ cercano a cero en el plano complejo, y no mayor a un radio de 0.5. El tiempo está dado en segundos.

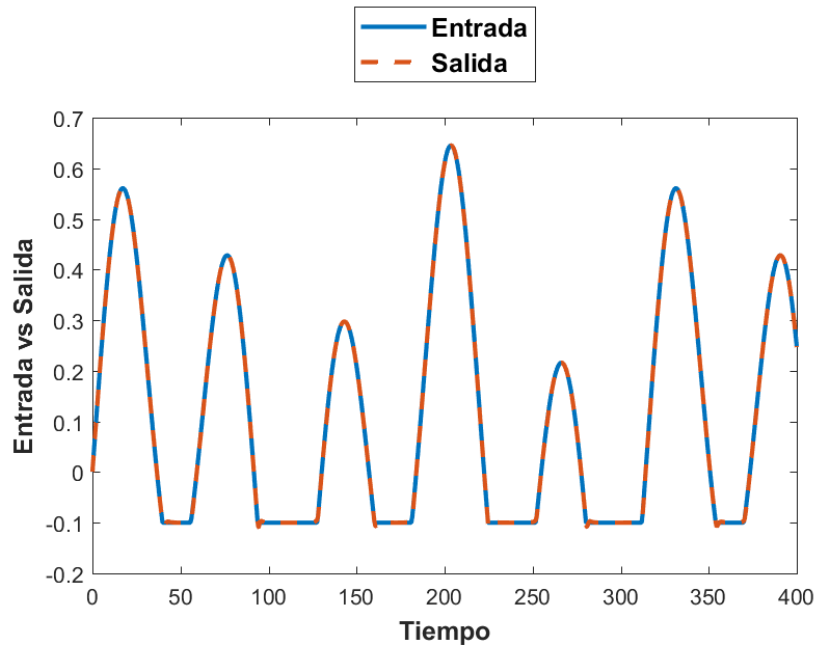


Figura 22 Parte Positiva de Identificación (LM).

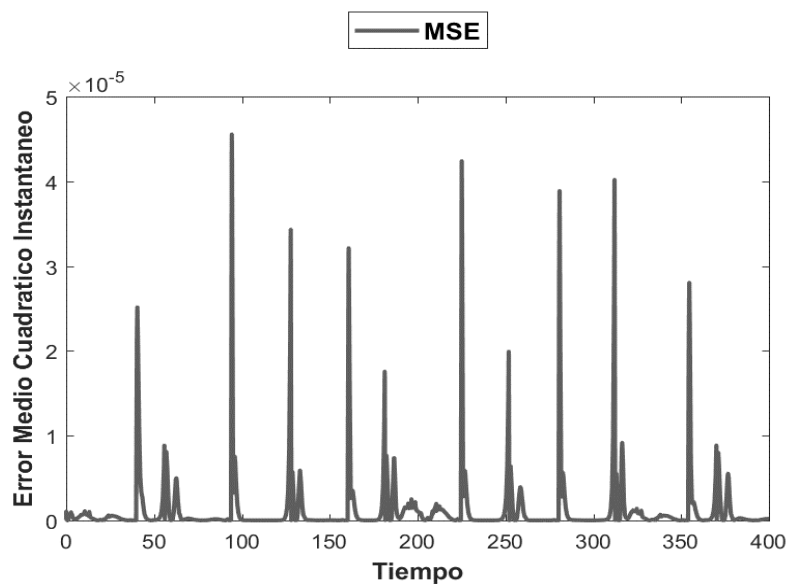


Figura 23. MSE Instantáneo de la Parte Positiva de Identificación (LM).

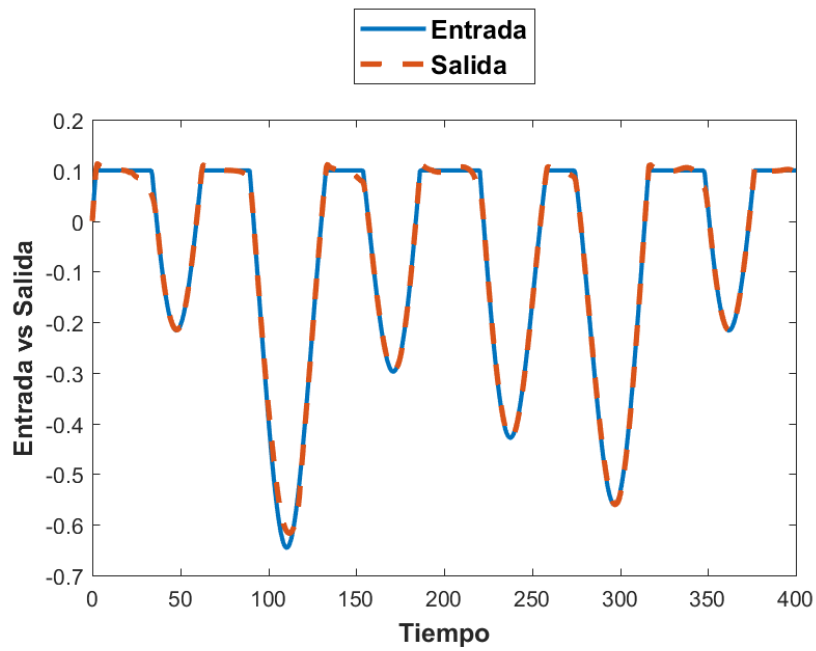


Figura 24. Parte Negativa de Identificación (LM).

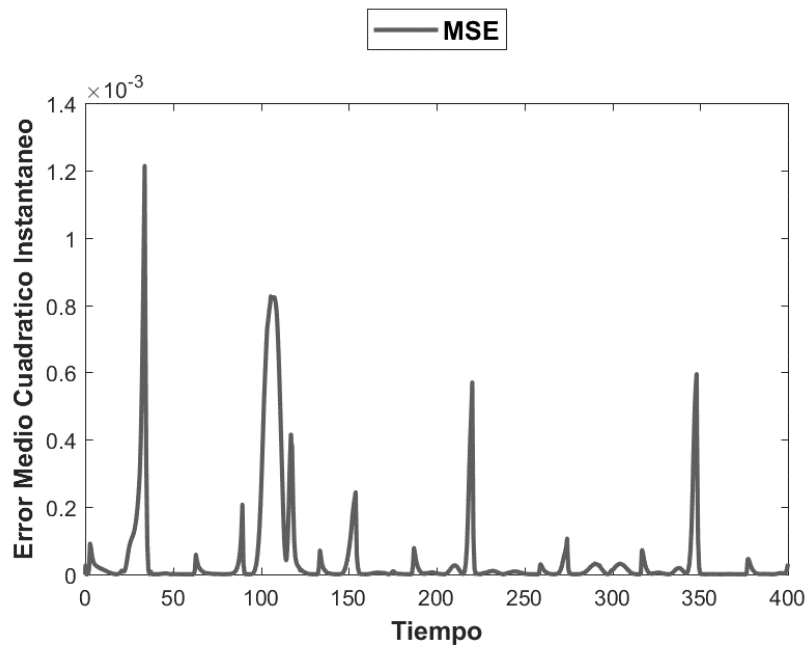


Figura 25. MSE Instantáneo de la Parte Negativa de Identificación (LM).

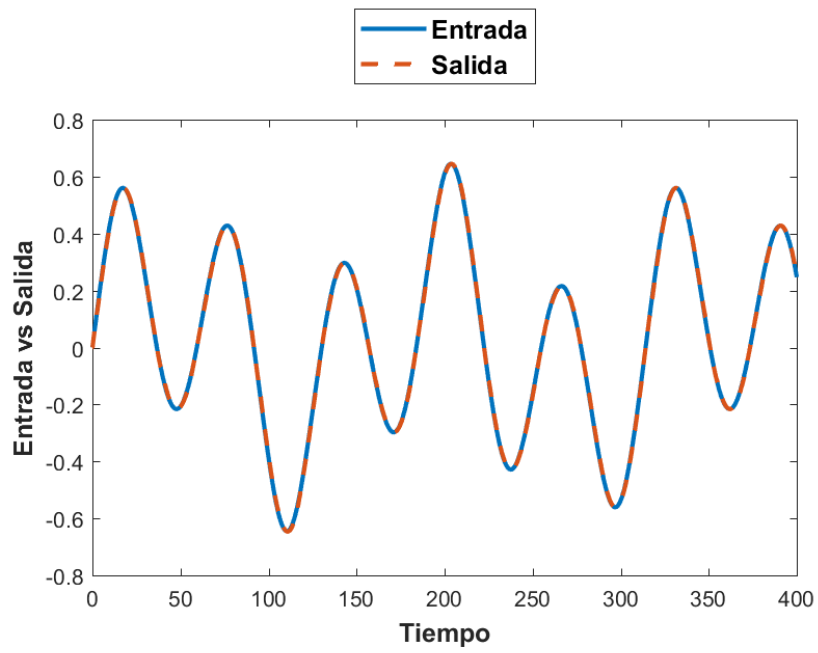


Figura 26 Señal Completa y de Identificación (LM).

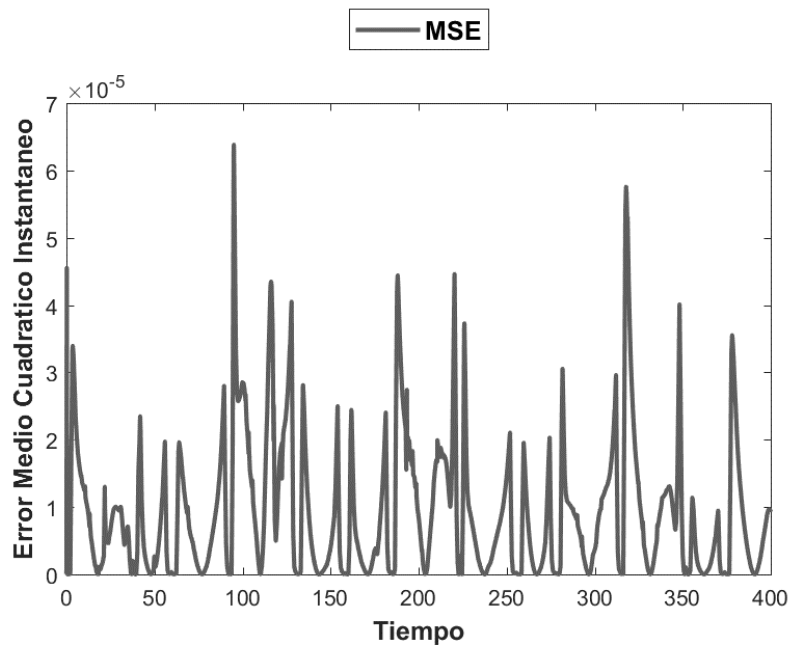


Figura 27 MSE Instantáneo de Identificación (LM).

5.4 Control

Además de la Identificación de sistemas, otro de los problemas de Control Automático, es el problema de Control, es decir, hacer que un sistema haga algo cuando se requiera, la mayor parte de la teoría de sistemas de lineales y sistemas no lineales se enfoca en desarrollar técnicas que permitan gobernar un sistema, las redes neuronales artificiales y los sistemas difusos han tenido un buen aprovechamiento para tal fin, ahora se ejemplificara el uso de HFNMM como medio control se ha propuesto dos esquemas de control:

- Control Adaptable por Modelo Inverso
- Control Adaptable por Modelo Inverso con Terminó Integral

Los cuáles serán aplicados a una planta idealizada de dos grados de libertad, la cual es representada por la figura 13. Donde se sigue respetando, que haya dos funciones de membresía con un recubrimiento en una vecindad cercana a 0. Un tiempo de muestreo de $\tau = 0.01$. Y condiciones iniciales complejas, con las restricciones propuestas en la sección anterior.

5.5 Control Adaptable por Modelo Inverso

A continuación, se presenta el esquema del Control Adaptable a través de un HFNMM aplicado a una planta idealizada de dos grados de libertad.

En el diagrama se observa que, al tener dos entradas y dos salidas, se puede plantear dos controles independientes que a su vez son divididos en parte positiva y parte negativa, debido a las funciones de membresía.

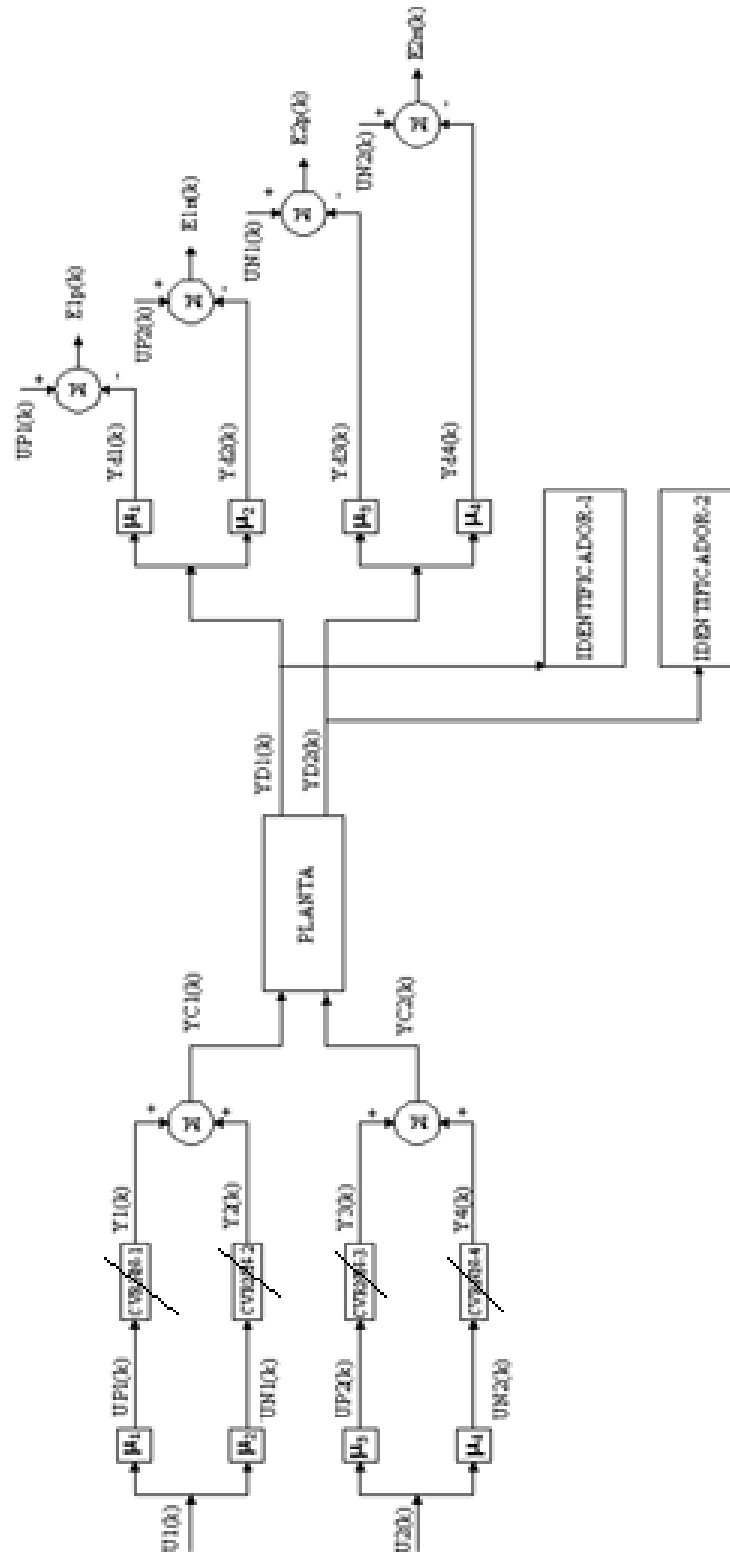


Figura 28. Esquema de Control Adaptable por Modelo Inverso con HFNMM

Por lo que los controles quedan definidos por:

$$Y_{c_1}(k) = Y_1(k) + Y_2(k) \quad (5.11)$$

$$Y_{c_2}(k) = Y_3(k) + Y_4(k) \quad (5.12)$$

Donde los errores de aprendizaje son:

$$E_{1p}(k) = U_{p1}(k) - Y_{d1}(k) \quad (5.13)$$

$$E_{1n}(k) = U_{n1}(k) - Y_{d2}(k) \quad (5.14)$$

$$E_{2p}(k) = U_{p2}(k) - Y_{d3}(k) \quad (5.15)$$

$$E_{2n}(k) = U_{n2}(k) - Y_{d4}(k) \quad (5.16)$$

Al aplicar el esquema anterior, se compara el aprendizaje llevado a cabo por *Back-Propagation* y *Levenberg-Marquardt*, donde cada una de las CVRNN tienen:

Tabla 6. Tabla de parámetros para las CVRNN utilizadas para Control Adaptable por Modelo Inverso de una planta de dos grados de libertad.

<i>CVRNN</i>	n	p	m
1	4	1	1
2	4	1	1
3	4	1	1
4	4	1	1

Como entrada para el esquema de control usaron señales cuadradas las cuales están dadas por:

$$\begin{aligned}
 u_{s1,2}(t) &= Ac * square(w_i * t + \phi), i = 1, 2 \\
 Ac &= 0.2, \phi = 0.5 \\
 w_1 &= 0.001 * \pi, w_2 = 0.003 * \pi
 \end{aligned} \quad (5.17)$$

Donde: Ac es la amplitud, ϕ es el ciclo útil, w_i es la frecuencia.

5.5.1 Back-Propagation Salida 1

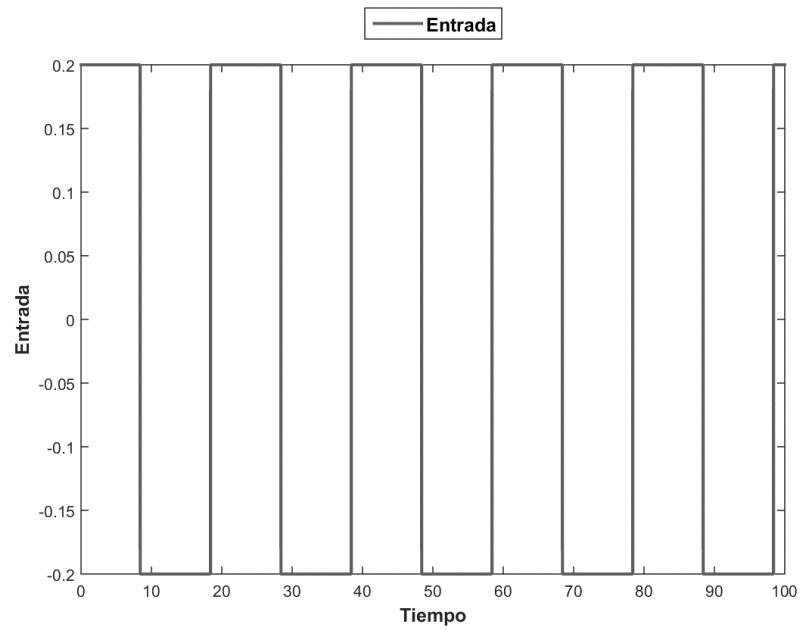


Figura 29. Entrada para la Señal de Salida 1 (BP).

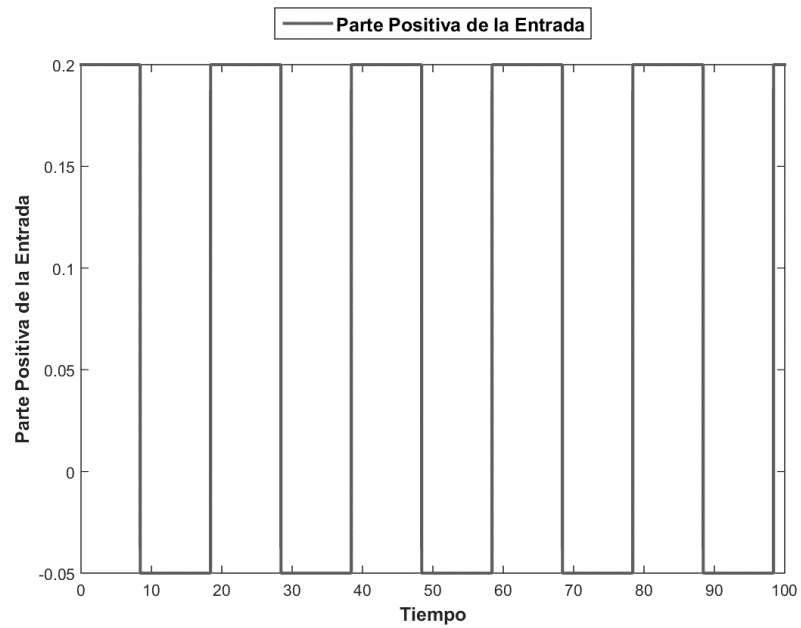


Figura 30. Parte Positiva de la Entrada de la Señal 1 (BP).

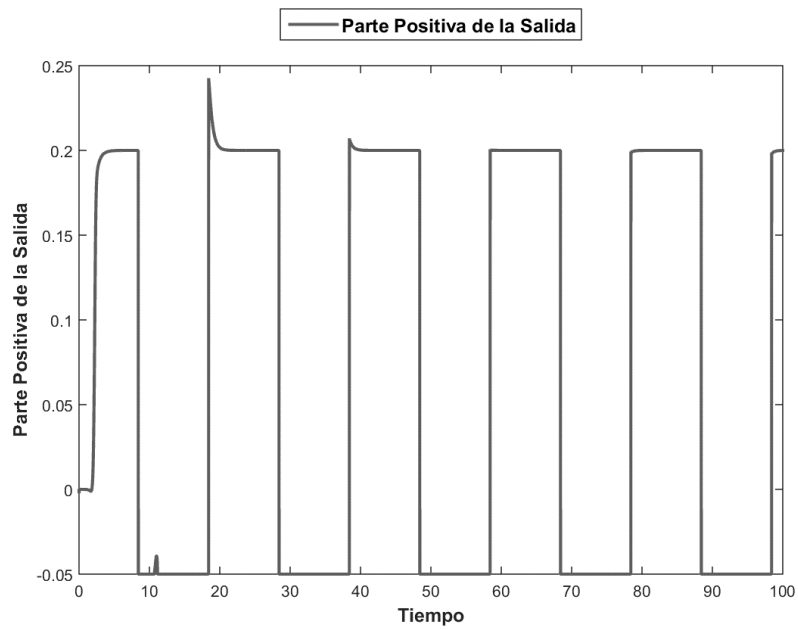


Figura 31. Parte Positiva de la Señal de Salida 1 (BP).

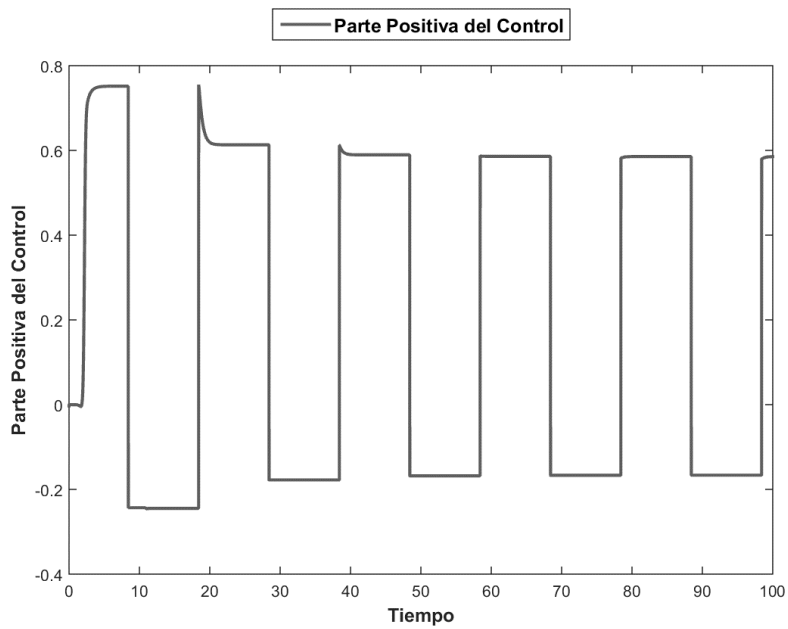


Figura 32. Parte Positiva del Control de la Señal de Salida 1 (BP).

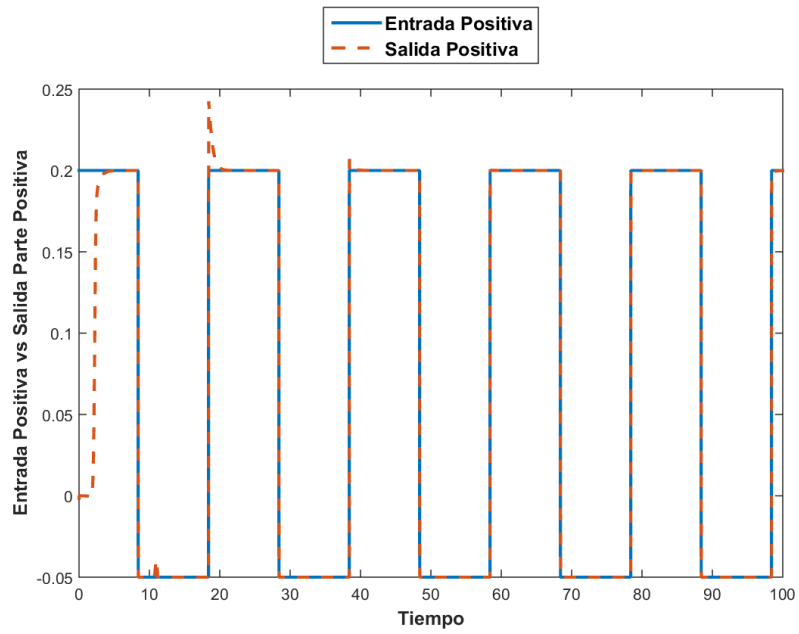


Figura 33. Comparación de la Parte Positiva de la Señal de Entrada con la Señal de Salida (BP).

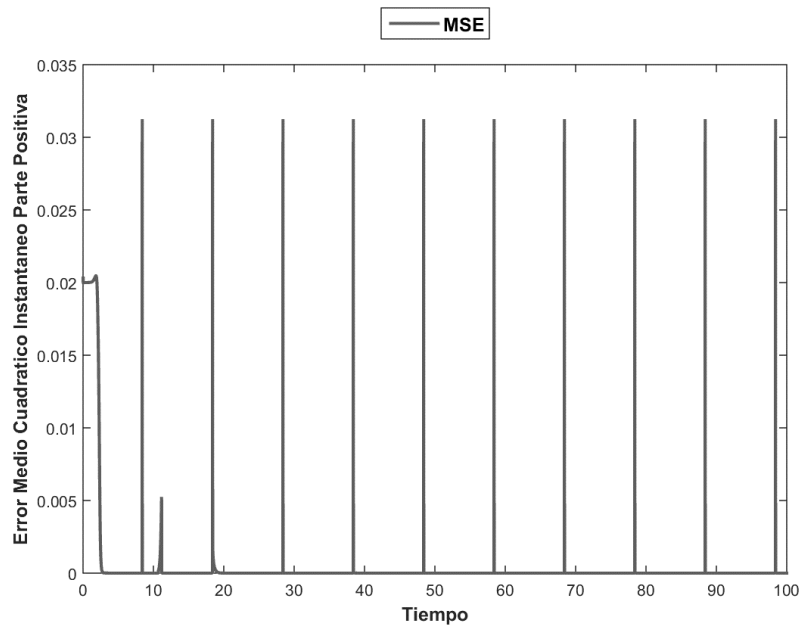


Figura 34. Error Medio Cuadrático Instantáneo de la Parte Positiva de la Señal 1 (BP).

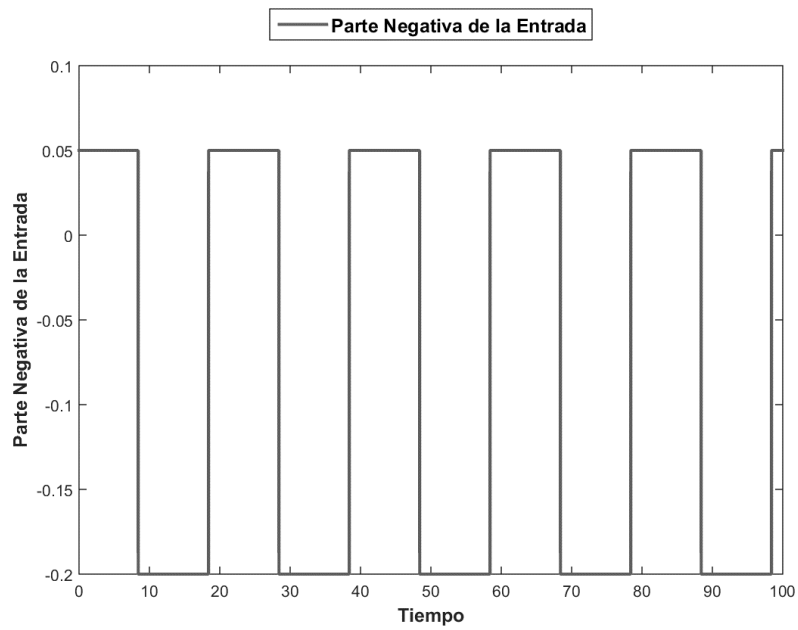


Figura 35. Parte Negativa de la Entrada de la Señal 1 (BP).

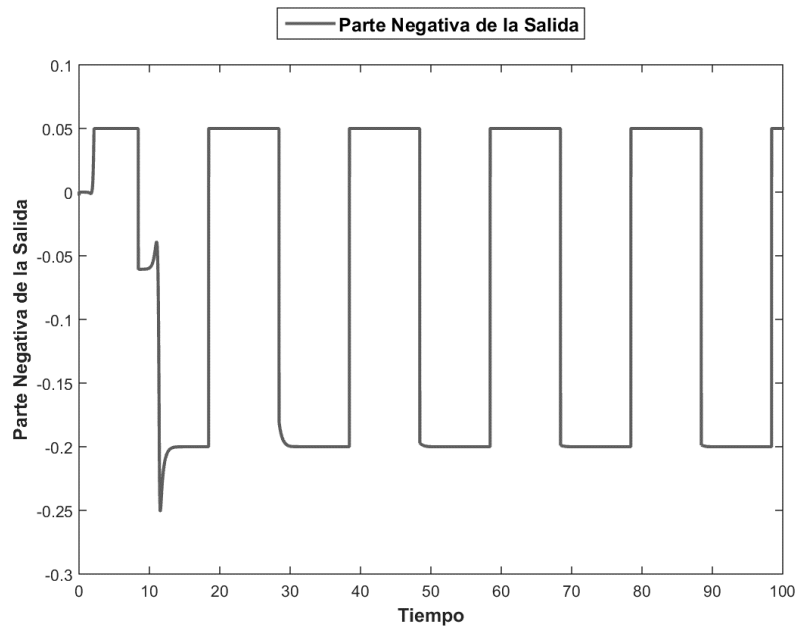


Figura 36. Parte Negativa de la Salida de la Señal de 1 (BP).

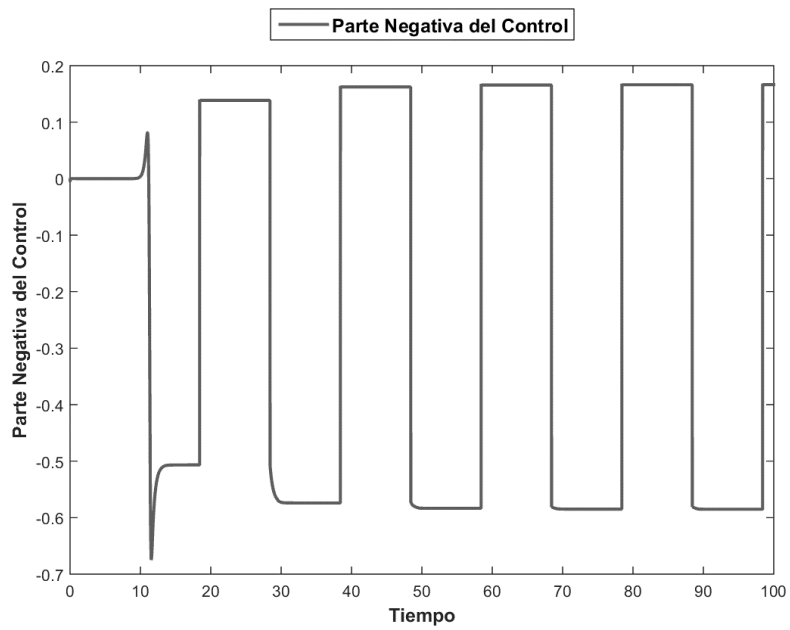


Figura 37. Parte Negativa del Control de la Señal de 1 (BP).

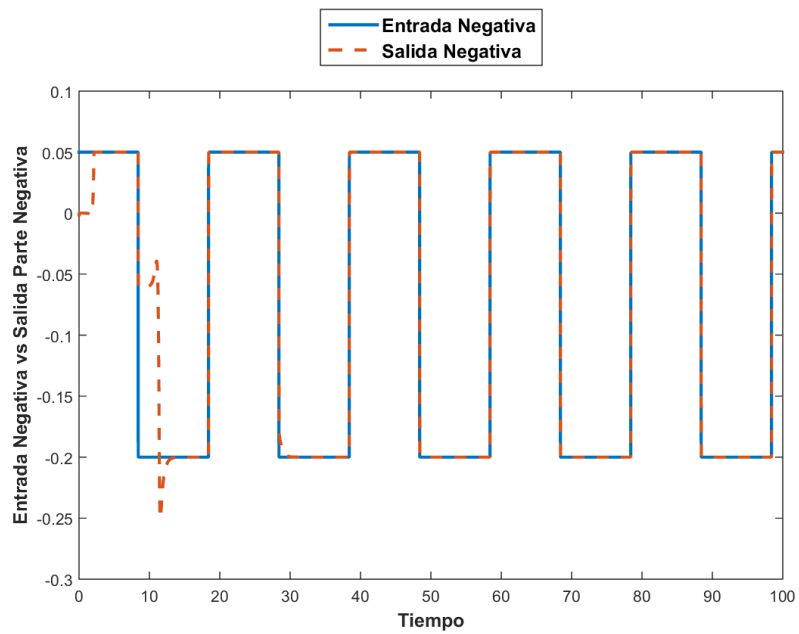


Figura 38. Comparación de la Parte Negativa de la Señal de Entrada con la Señal de Salida (BP).

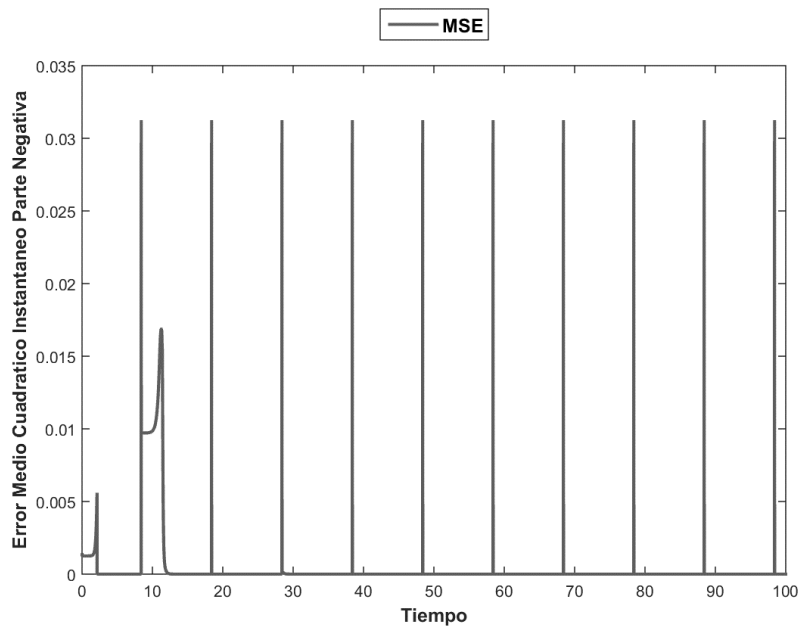


Figura 39. Error Medio Cuadrático Instantáneo de la Parte Negativa de la Señal 1 (BP).

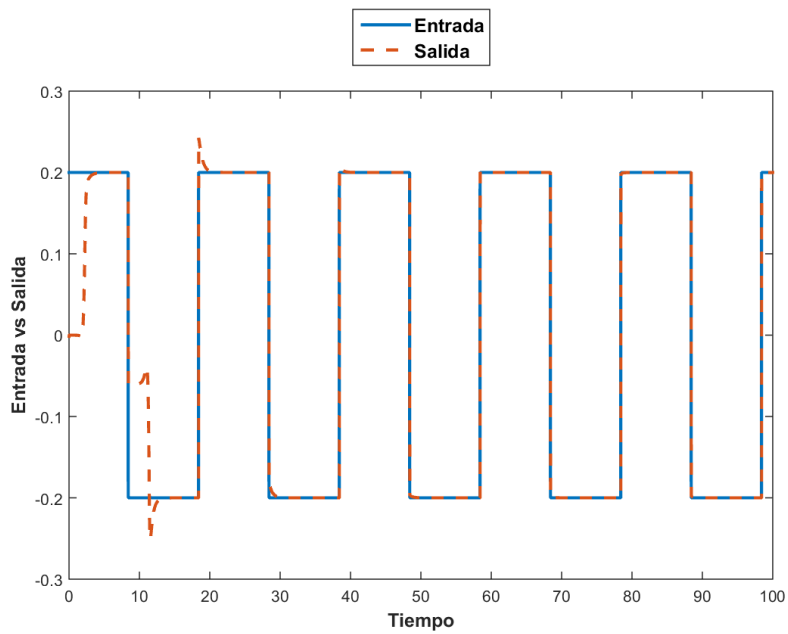


Figura 40. Comparación de la Señal de Entrada con la Señal de Salida para la Señal 1 (BP).

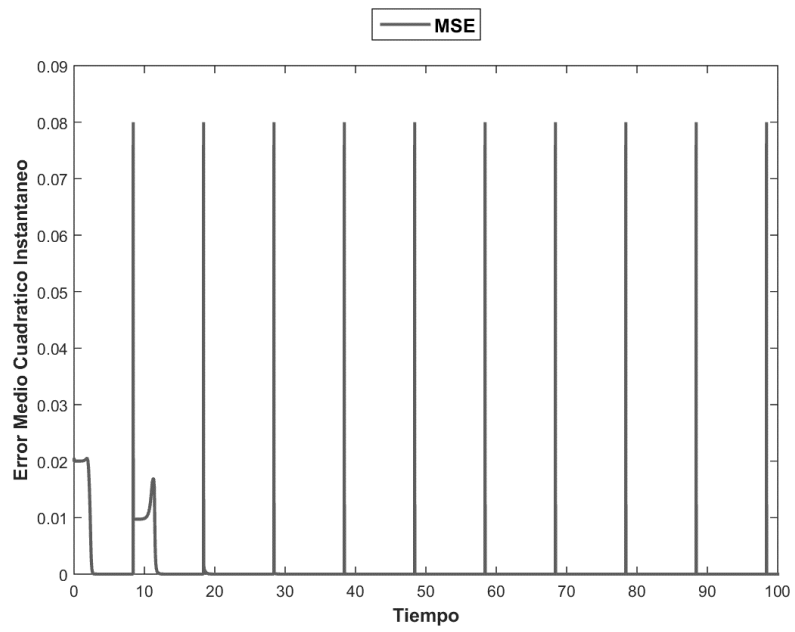


Figura 41. Error Medio Cuadrático Instantáneo de la Señal 1 (BP).

Nota: El tiempo está dado en segundos.

5.5.2 Back-Propagation Salida 2

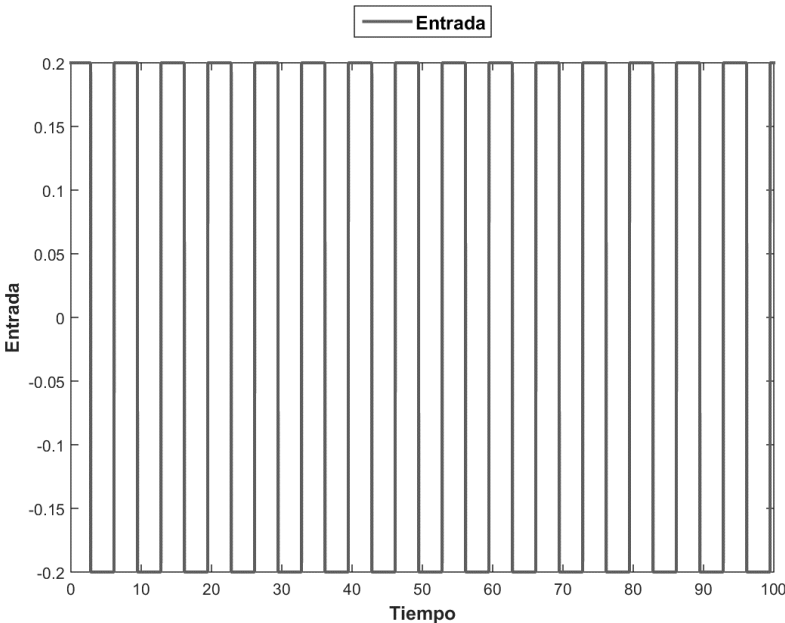


Figura 42. Entrada para la Señal de Salida 2 (BP).

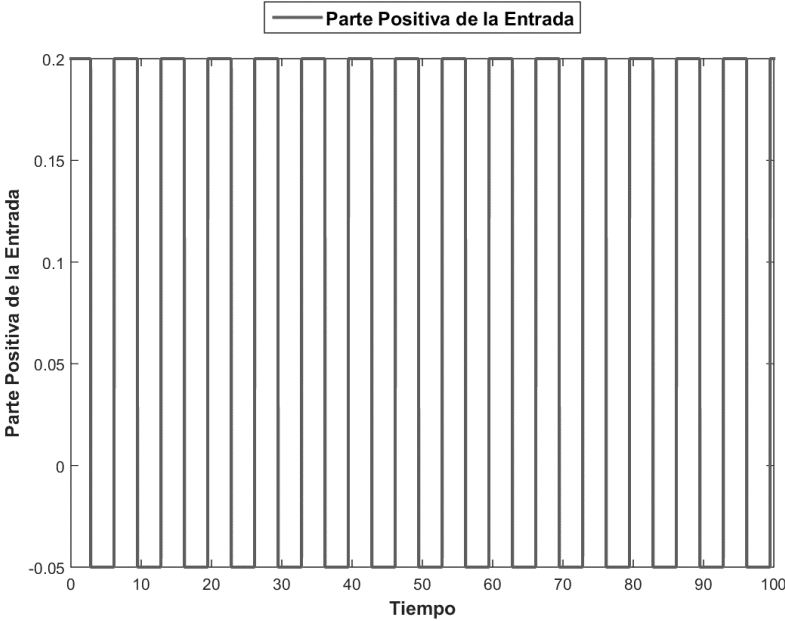


Figura 43. Parte Positiva de la Entrada de la Señal 2 (BP).

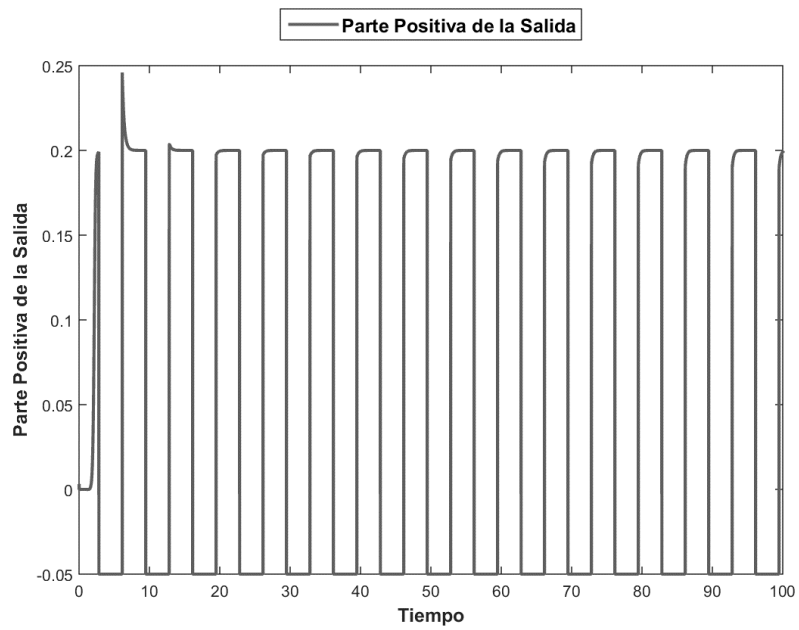


Figura 44. Parte Positiva de la Señal de Salida 2 (BP).

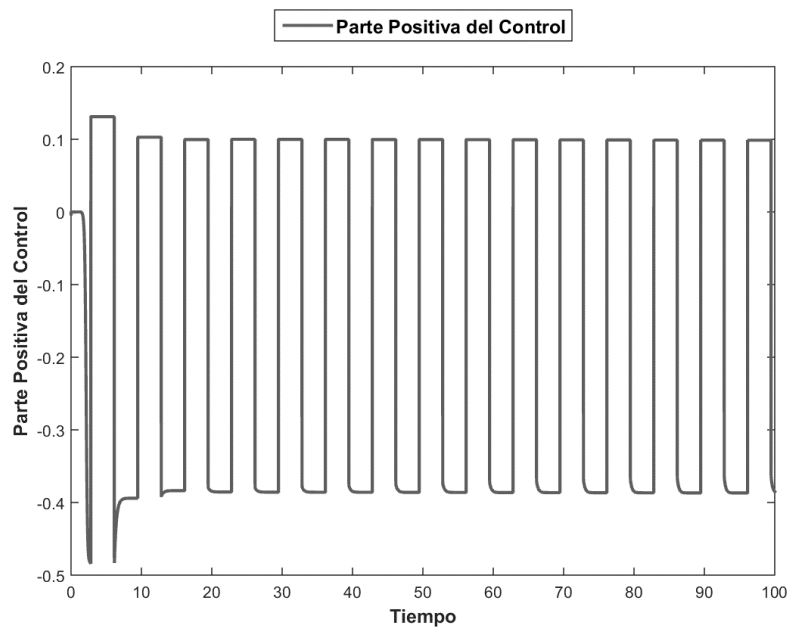


Figura 45. Parte Positiva del Control de la Señal de Salida 2 (BP).

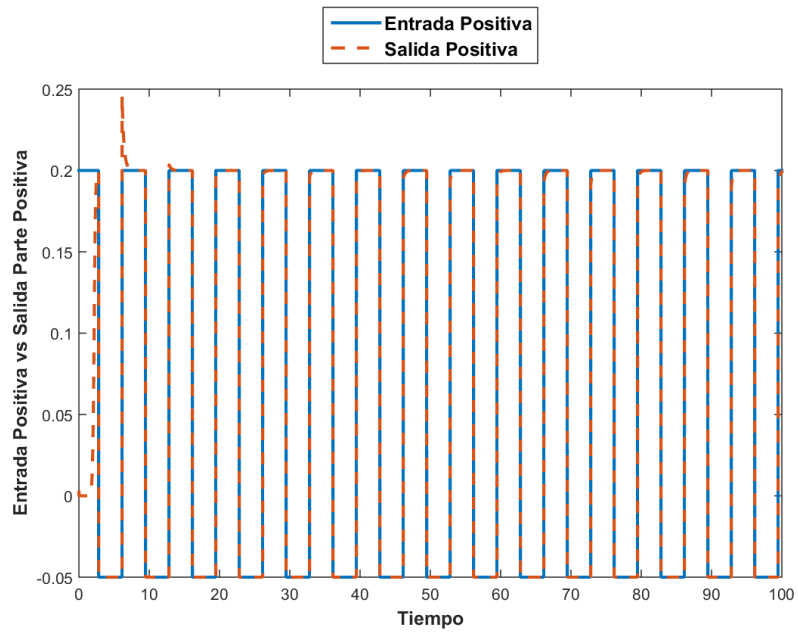


Figura 46. Comparación de la Parte Positiva de la Señal de Entrada con la Señal de Salida (BP).

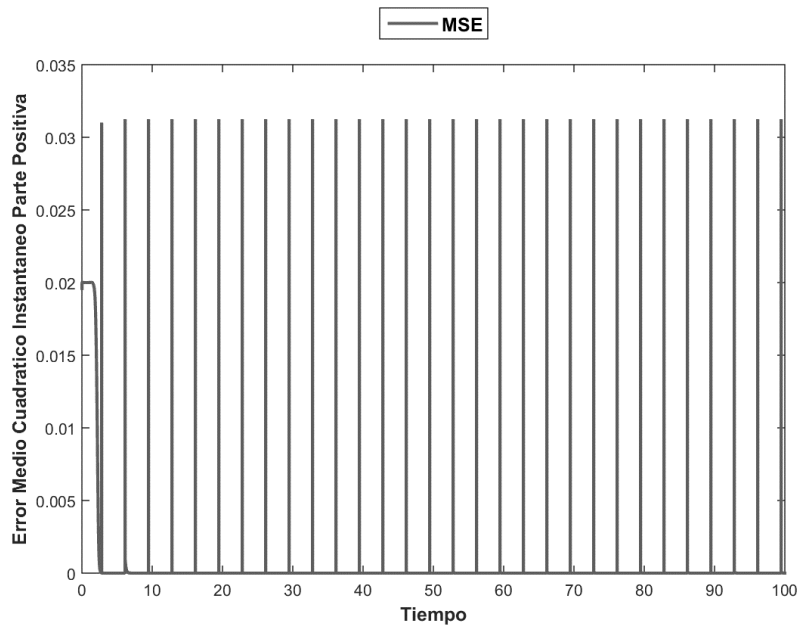


Figura 47. Error Medio Cuadrático Instantáneo de la Parte Positiva de la Señal 2 (BP).

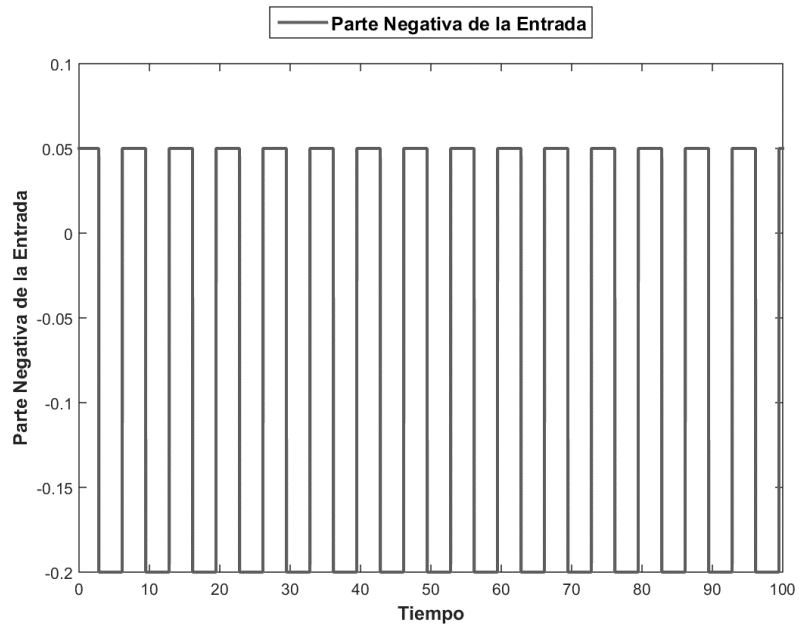


Figura 48. Parte Negativa de la Entrada de la Señal de 2 (BP).

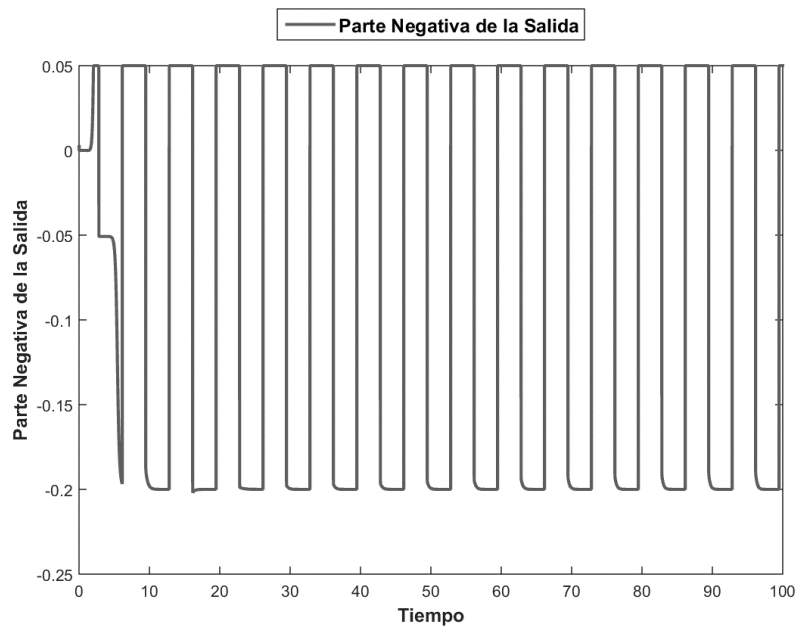


Figura 49. Parte Negativa de la Entrada de la Señal 2 (BP).

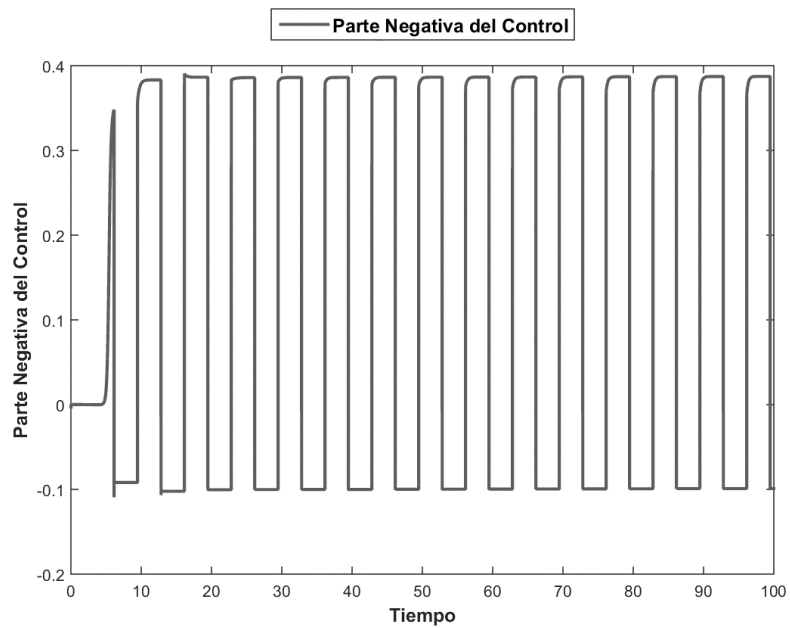


Figura 50. Parte Negativa del Control de la Señal de 2 (BP).

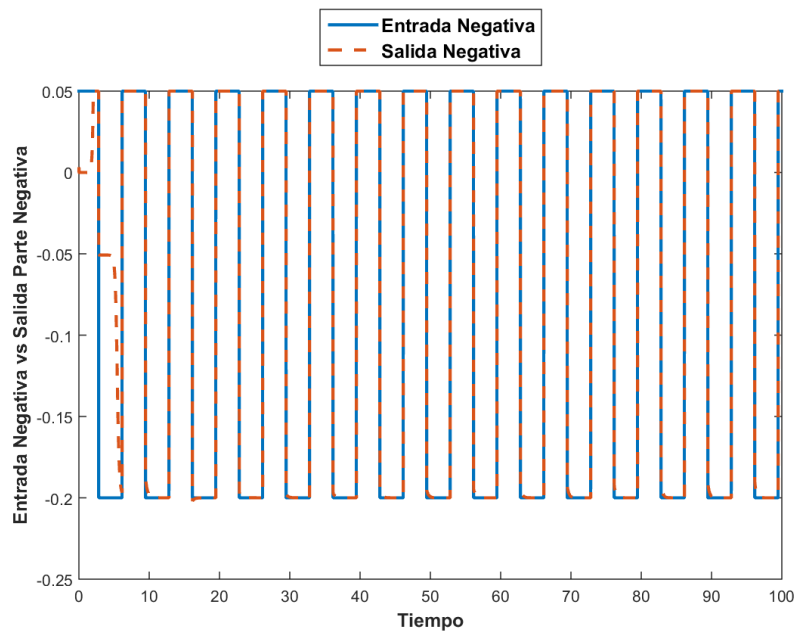


Figura 51. Comparación de la Señal de Entrada con la Señal de Salida para la Señal 2 (BP).

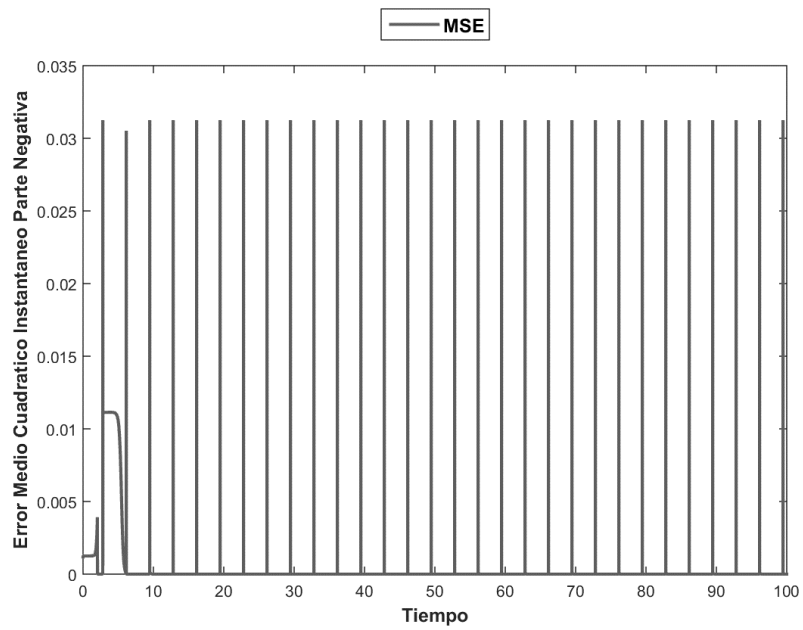


Figura 52. Error Medio Cuadrático Instantáneo de la Parte Negativa de la Señal 2 (BP).

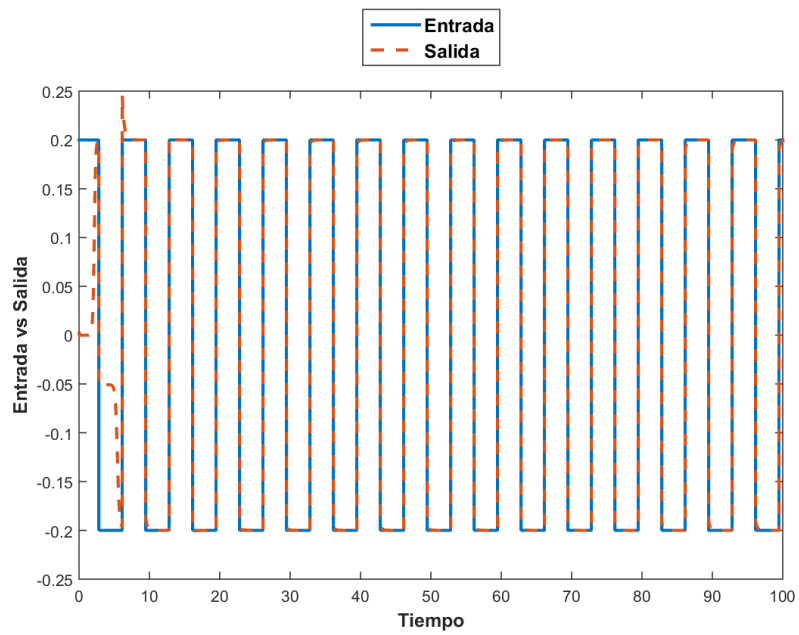


Figura 53. Comparación de la Señal de Entrada con la Señal de Salida para la Señal 2 (BP).

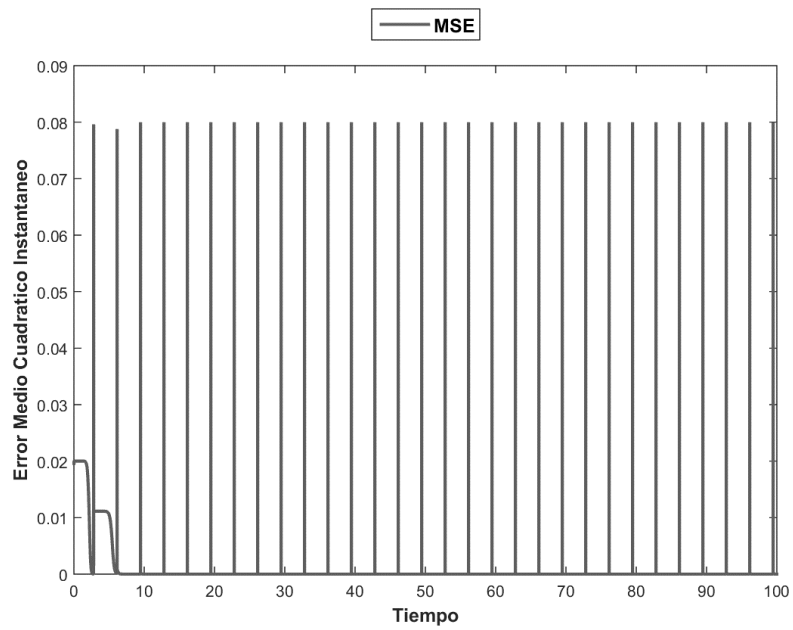


Figura 54. Error Medio Cuadrático Instantáneo de la Señal 2 (BP).

Nota: El tiempo está dado en segundos.

5.5.3 Levenberg-Marquardt Salida 1

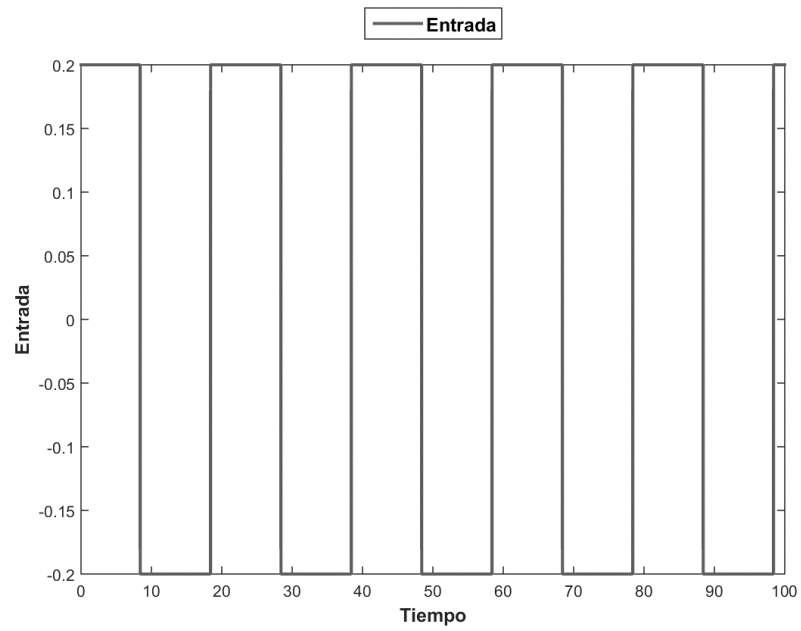


Figura 55. Entrada para la Señal de Salida 1 (LM).

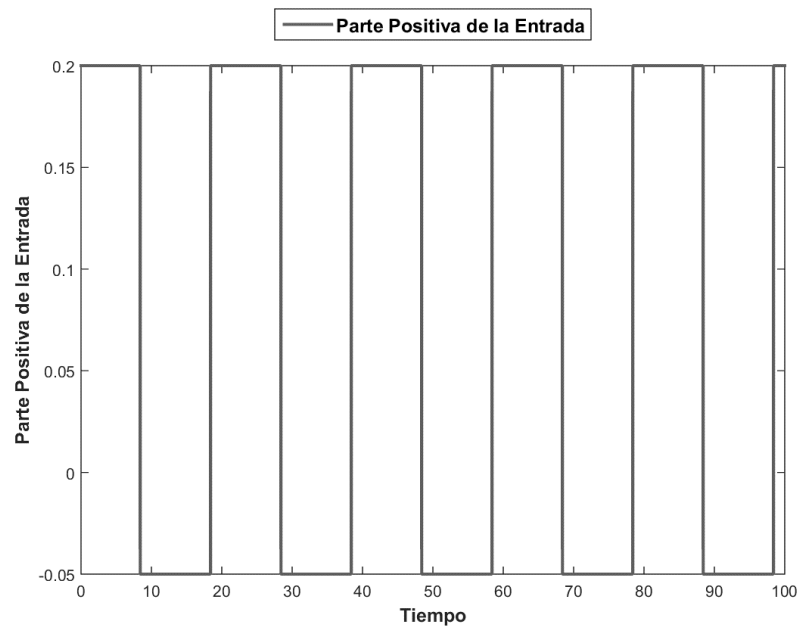


Figura 56. Parte Positiva de la Entrada de la Señal 1 (LM).

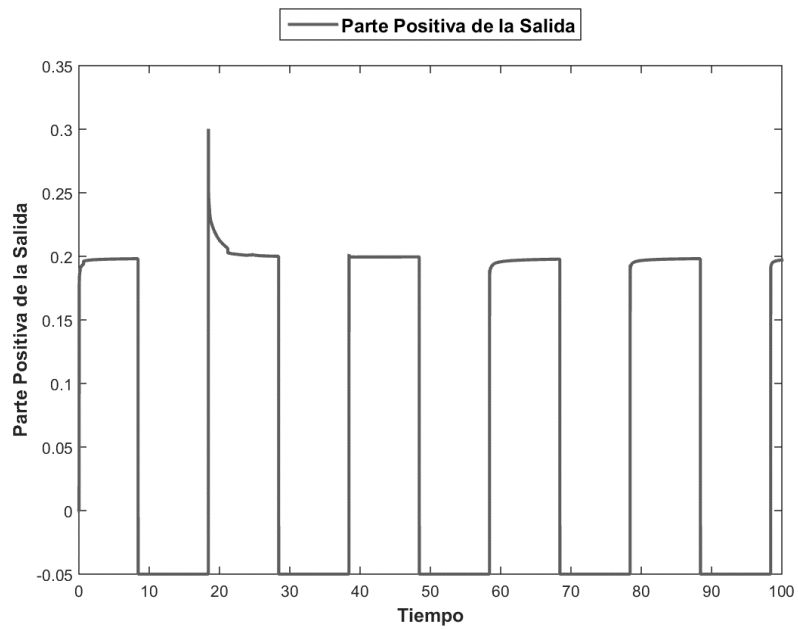


Figura 57. Parte Positiva de la Señal de Salida 1 (LM).

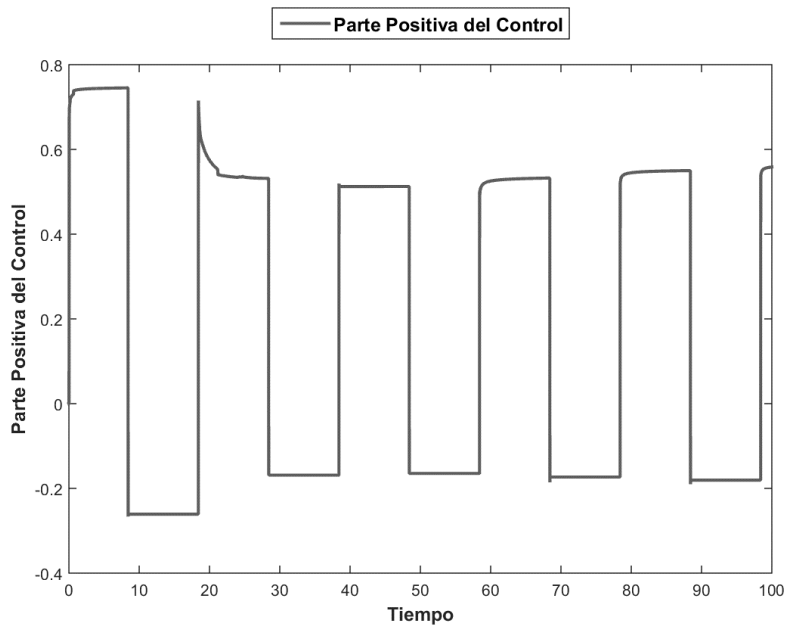


Figura 58. Parte Positiva del Control de la Señal de Salida 1 (LM).

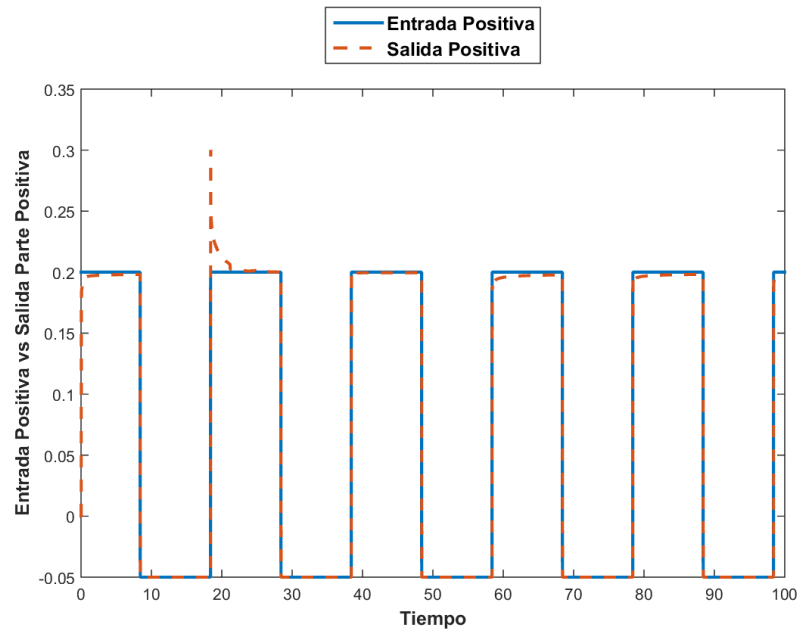


Figura 59. Comparación de la Parte Positiva de la Señal de Entrada con la Señal de Salida (LM).

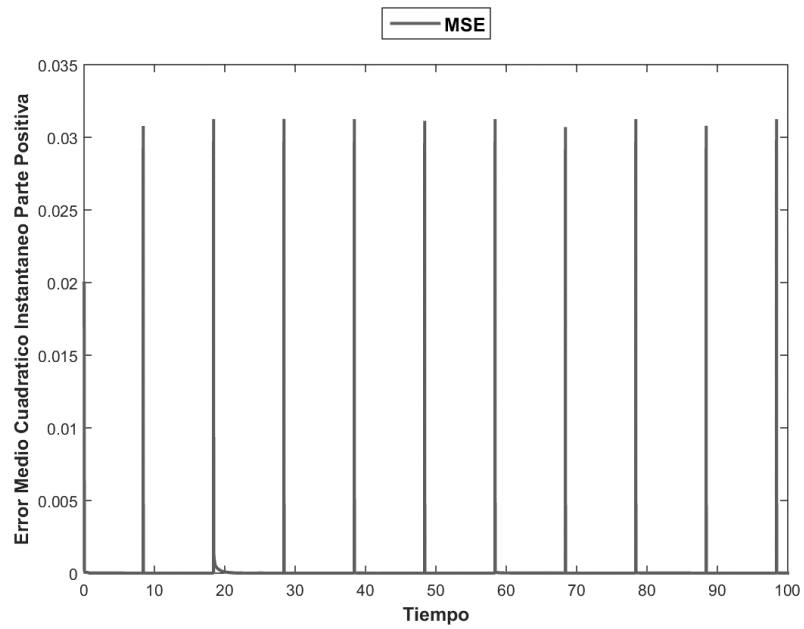


Figura 60. Error Medio Cuadrático Instantáneo de la Parte Positiva de la Señal 1 (LM).

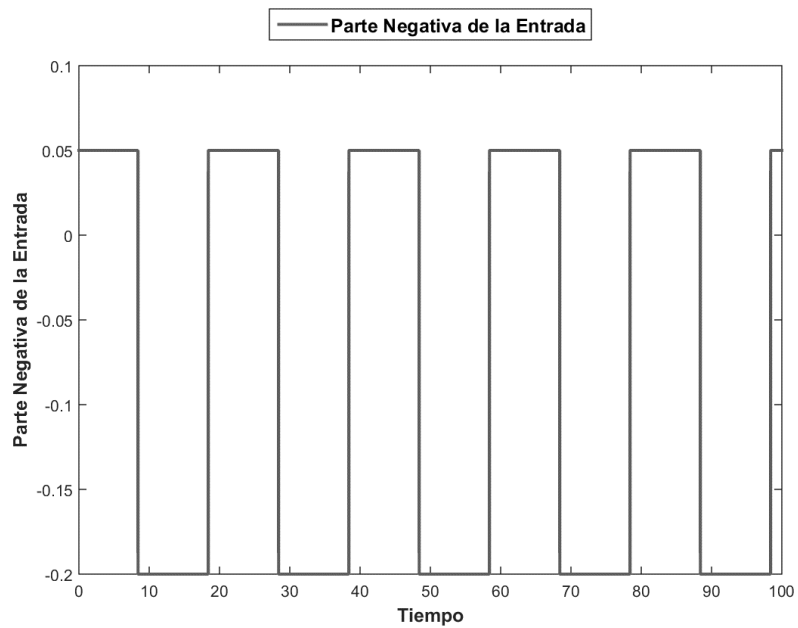


Figura 61. Parte Negativa de la Entrada de la Señal 1 (LM).

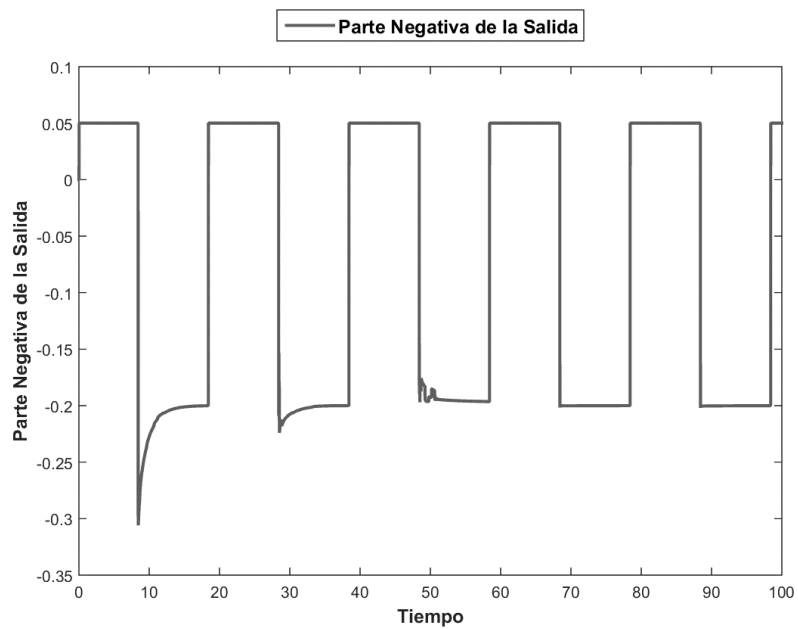


Figura 62. Parte Negativa de la Salida de la Señal de 1 (LM).

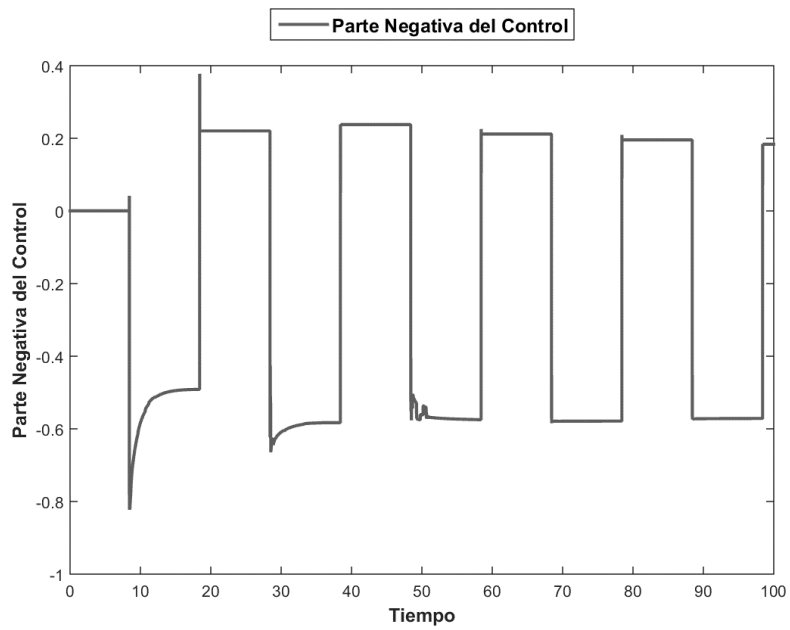


Figura 63. Parte Negativa de la Entrada de la Señal de 1 (LM).

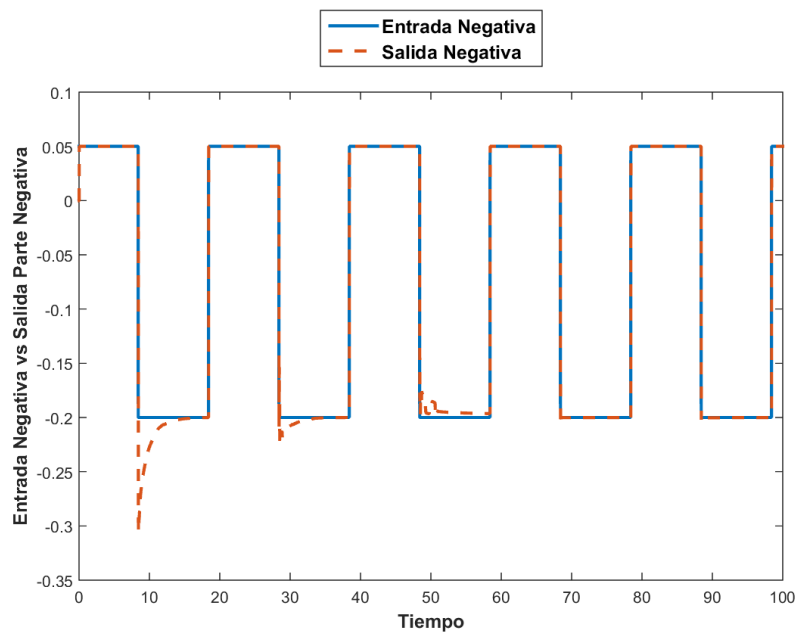


Figura 64. Comparación de la Parte Negativa de la Señal de Entrada con la Señal de Salida (LM).

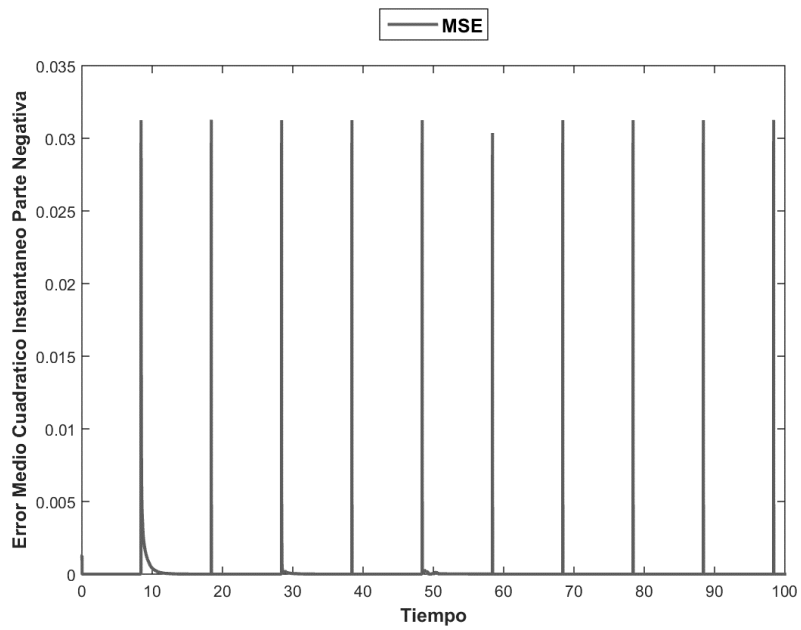


Figura 65. Error Medio Cuadrático Instantáneo de la Parte Negativa de la Señal 1 (LM).

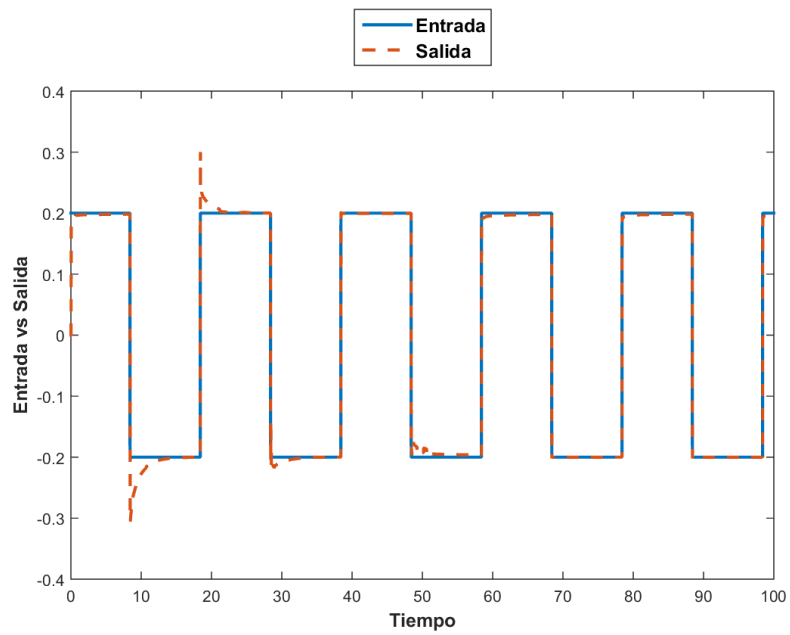


Figura 66. Comparación de la Señal de Entrada con la Señal de Salida para la Señal 1 (LM).

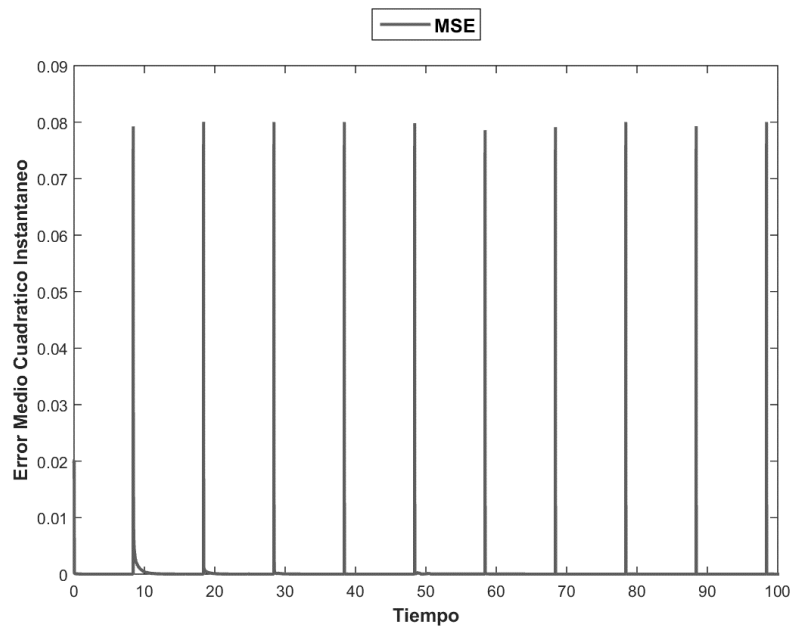


Figura 67. Error Medio Cuadrático Instantáneo de la Señal 1 (LM).

Nota: El tiempo está dado en segundos.

5.5.4 Levenberg-Marquardt Salida 2

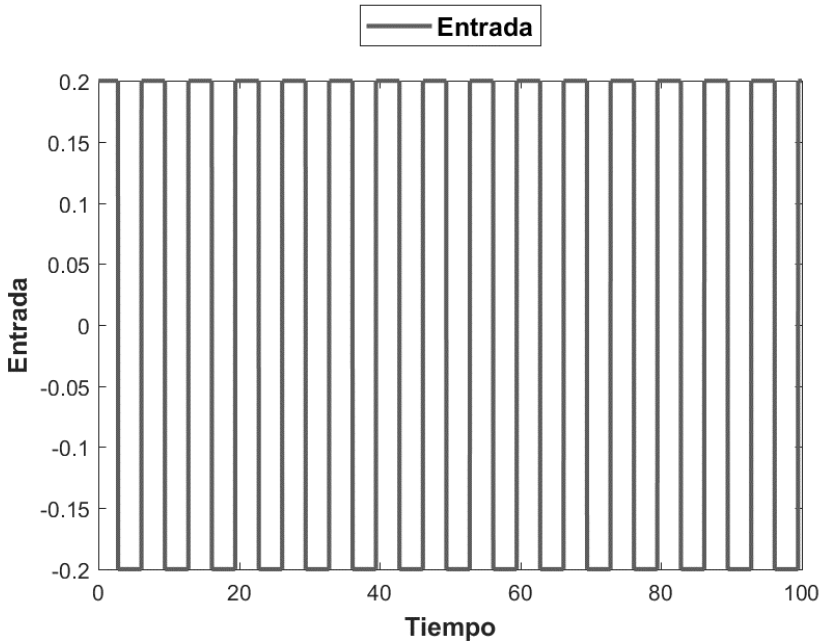


Figura 68. Entrada para la Señal de Salida 2 (LM).

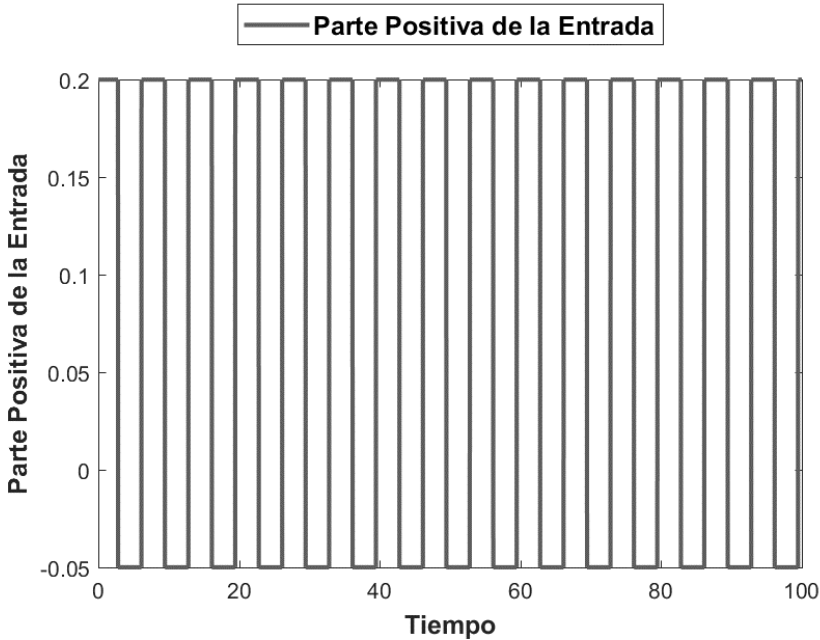


Figura 69. Parte Positiva de la Entrada de la Señal 2 (LM).

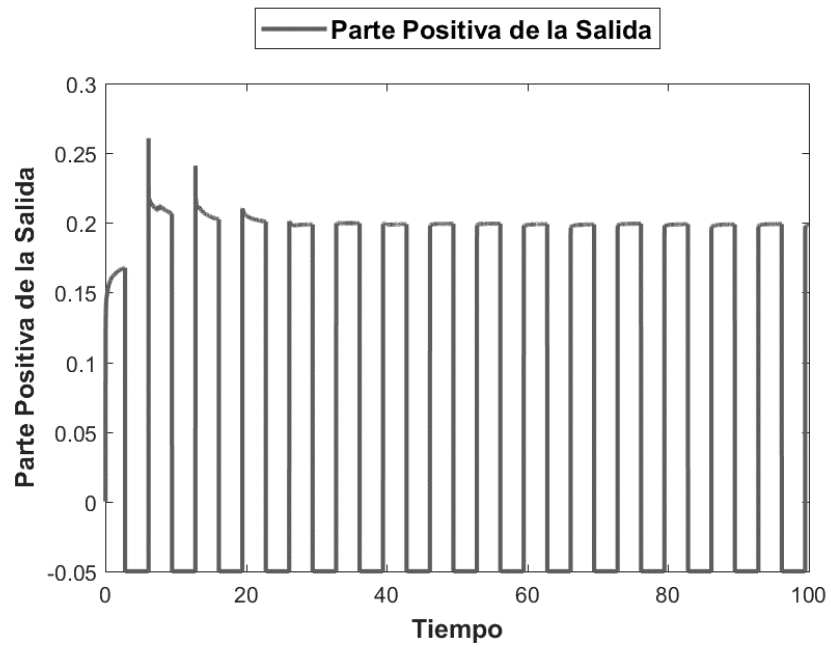


Figura 70. Parte Positiva de la Señal de Salida 2 (LM).

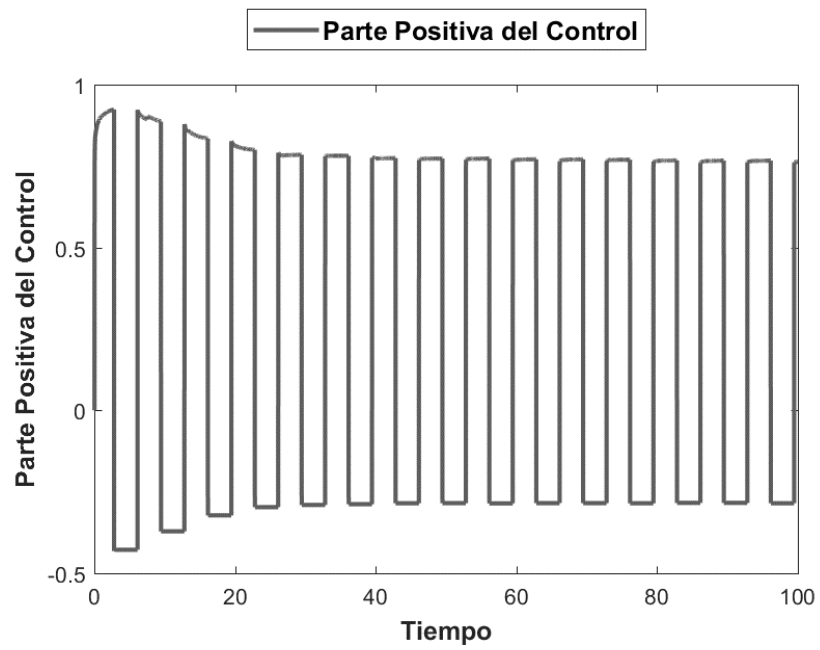


Figura 71. Parte Positiva del Control de la Señal de Salida 2 (LM).

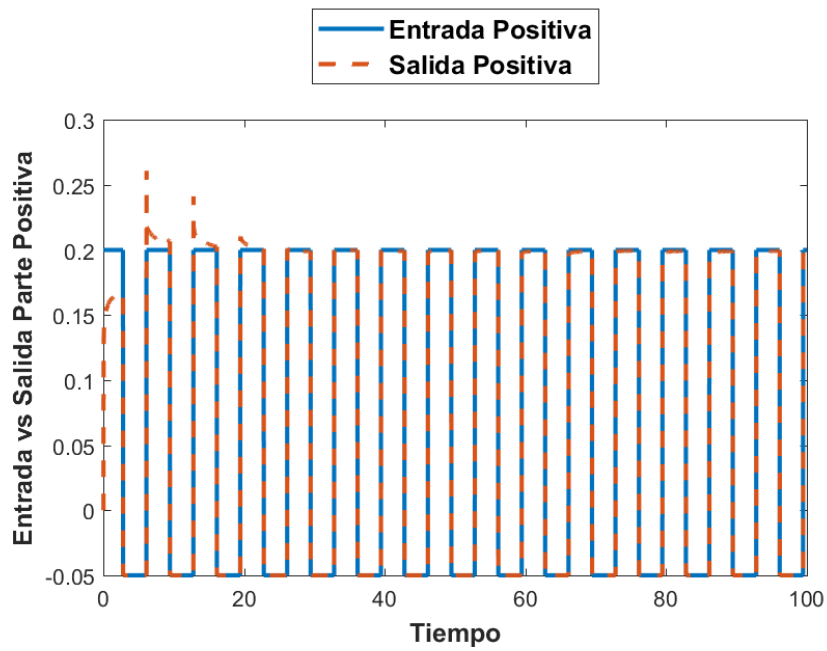


Figura 72. Parte Positiva del Control de la Señal de Salida 2 (LM).

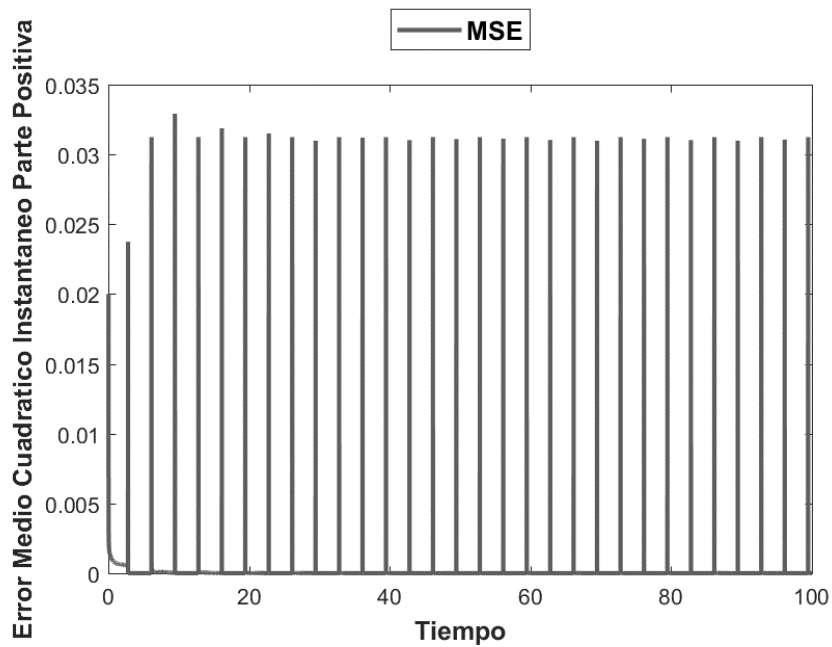


Figura 73. Error Medio Cuadrático Instantáneo de la Parte Positiva de la Señal 2 (LM).

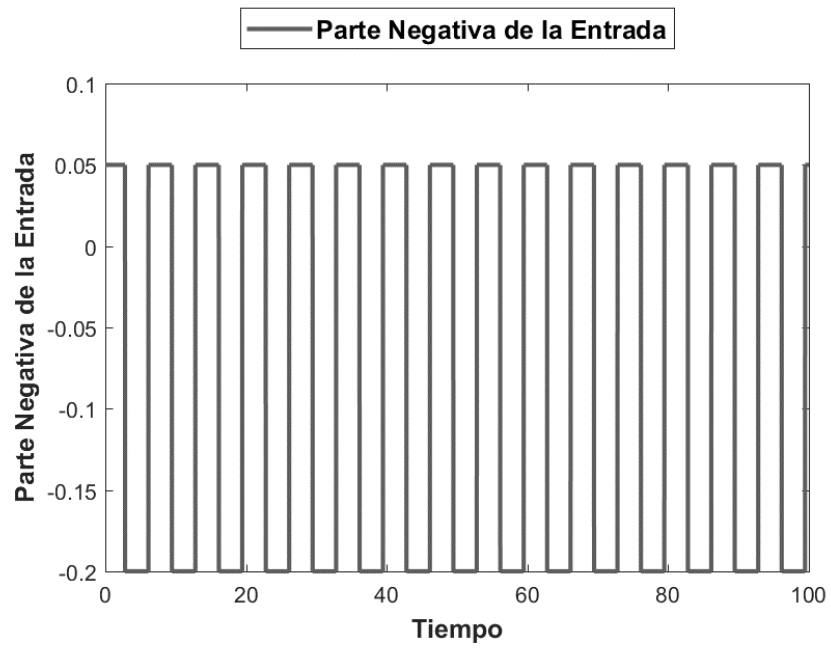


Figura 74. Parte Negativa de la Entrada de la Señal 2 (LM).

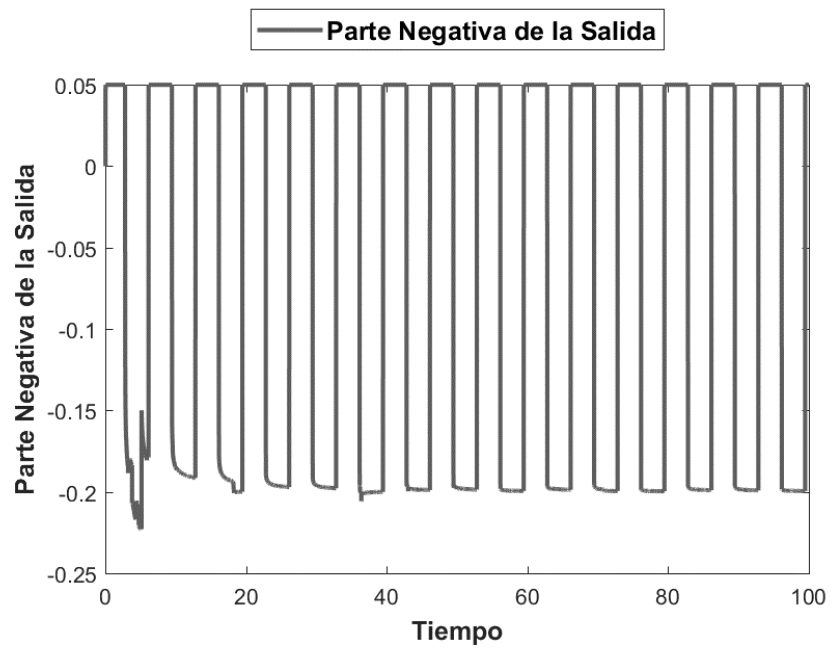


Figura 75. Parte Negativa de la Salida de la Señal de 2 (LM).

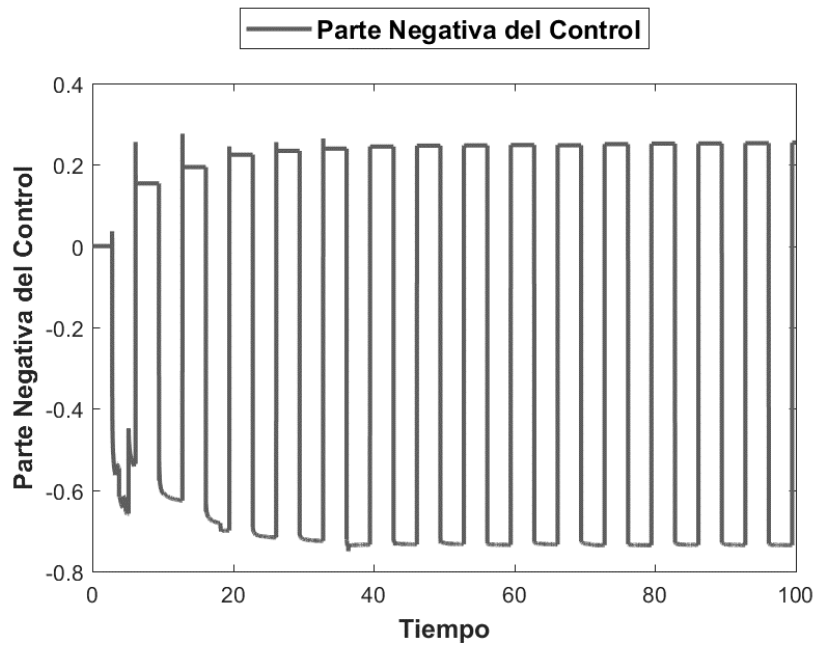


Figura 76. Parte Negativa del Control de la Señal de 2 (LM).

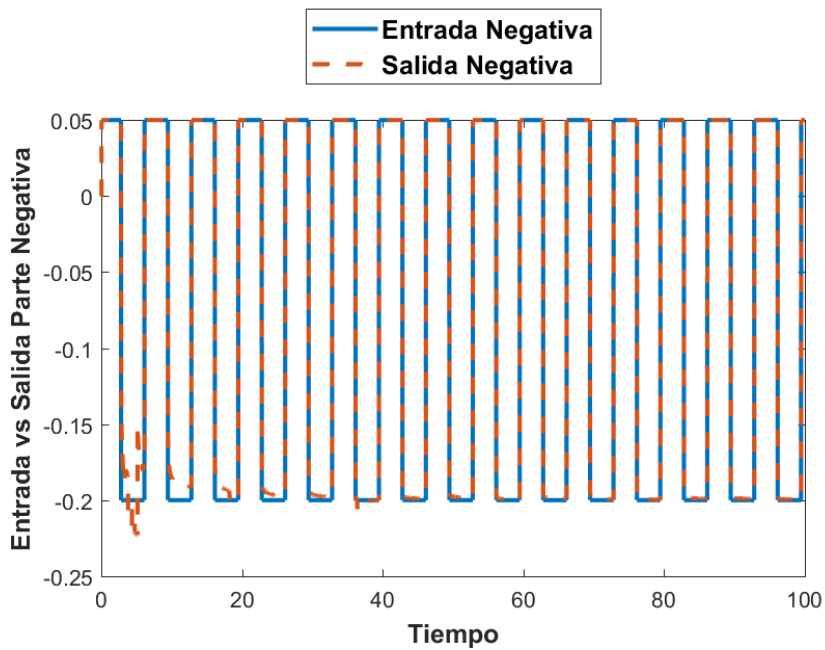


Figura 77. Comparación de la Parte Negativa de la Señal de Entrada con la Señal de Salida (LM).

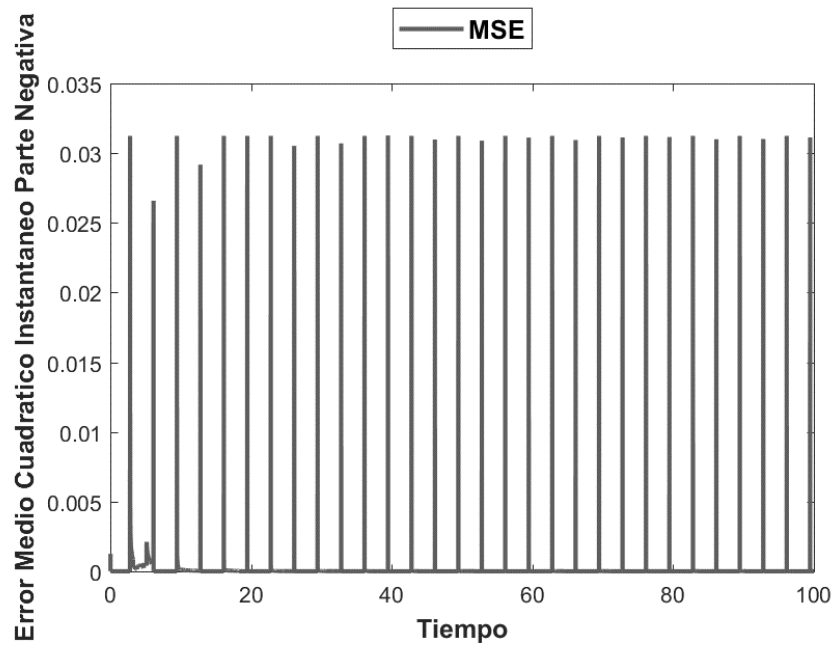


Figura 78. Error Medio Cuadrático Instantáneo de la Parte Negativa de la Señal 2 (LM).

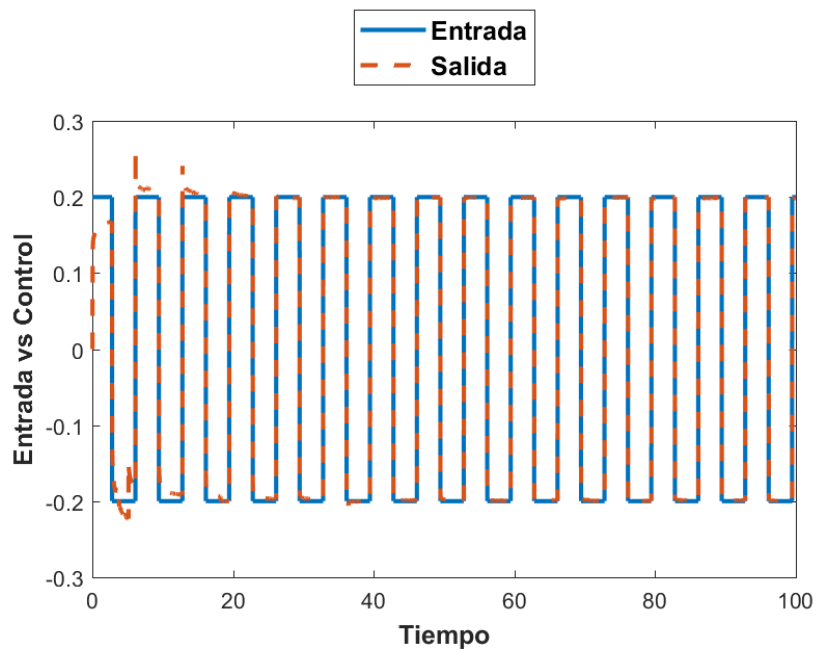


Figura 79. Comparación de la Señal de Entrada con la Señal de Salida para la Señal 2 (LM).

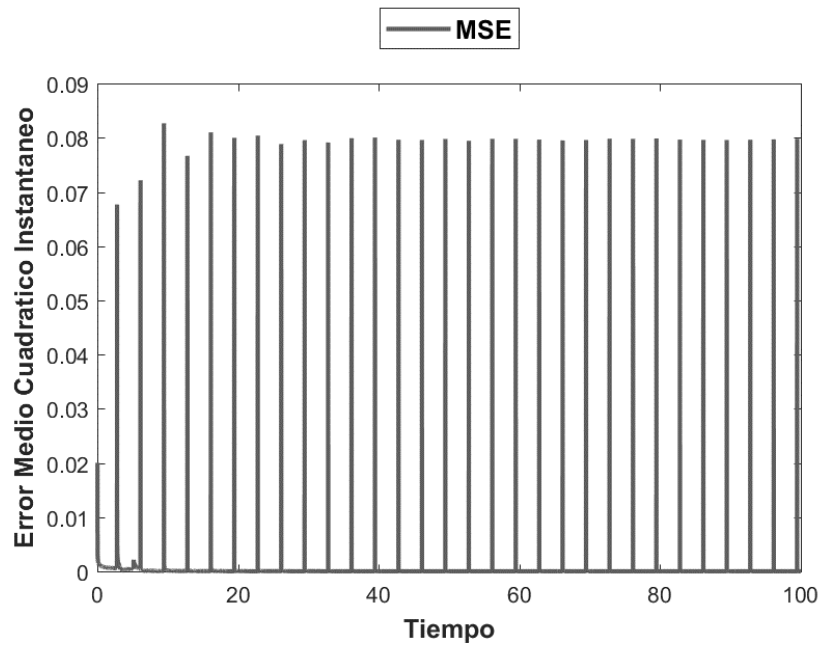


Figura 80. Error Medio Cuadrático Instantáneo de la Señal 2 (LM).

Nota: El tiempo está dado en segundos.

5.6 Control Adaptable por Modelo Inverso con Termino Integral

A continuación, se presenta el esquema del Control Adaptable con termino integral a través de un HFNMM aplicado a una planta idealizada de dos grados de libertad.

En la figura 81, se observa que, al tener dos entradas y dos salidas, se puede plantear dos controles independientes que a su vez son divididos en parte positiva y parte negativa, debido a las funciones de membresía; además se agrega a la ley de control un término integral, cual ayudara a eliminar el error de una manera más rápida, mejorando notablemente los resultados con respecto a los obtenidos en 5.5.

Por lo que los controles quedan definidos por:

$$Y_{C_1}(k) = (Y_1(k) + Ti_1(k)) + (Y_2(k) + Ti_2(k)) \quad (5.18)$$

$$Y_{C_2} = (Y_3(k) + Ti_3(k)) + (Y_4(k) + Ti_4(k)) \quad (5.19)$$

La ecuación que describe el termino integral es:

$$Ti(k+1) = Ti(k) + \tau \cdot Ki \cdot E(k) \quad (5.20)$$

Donde τ es el tiempo de muestreo, Ki es una ganancia y $E(k)$ es el error.

Los errores de aprendizaje son:

$$E_{1p}(k) = U_{p1}(k) - Y_{d1}(k) \quad (5.21)$$

$$E_{1n}(k) = U_{n1}(k) - Y_{d2}(k) \quad (5.22)$$

$$E_{2p}(k) = U_{p2}(k) - Y_{d3}(k) \quad (5.23)$$

$$E_{2n}(k) = U_{n2}(k) - Y_{d4}(k) \quad (5.24)$$

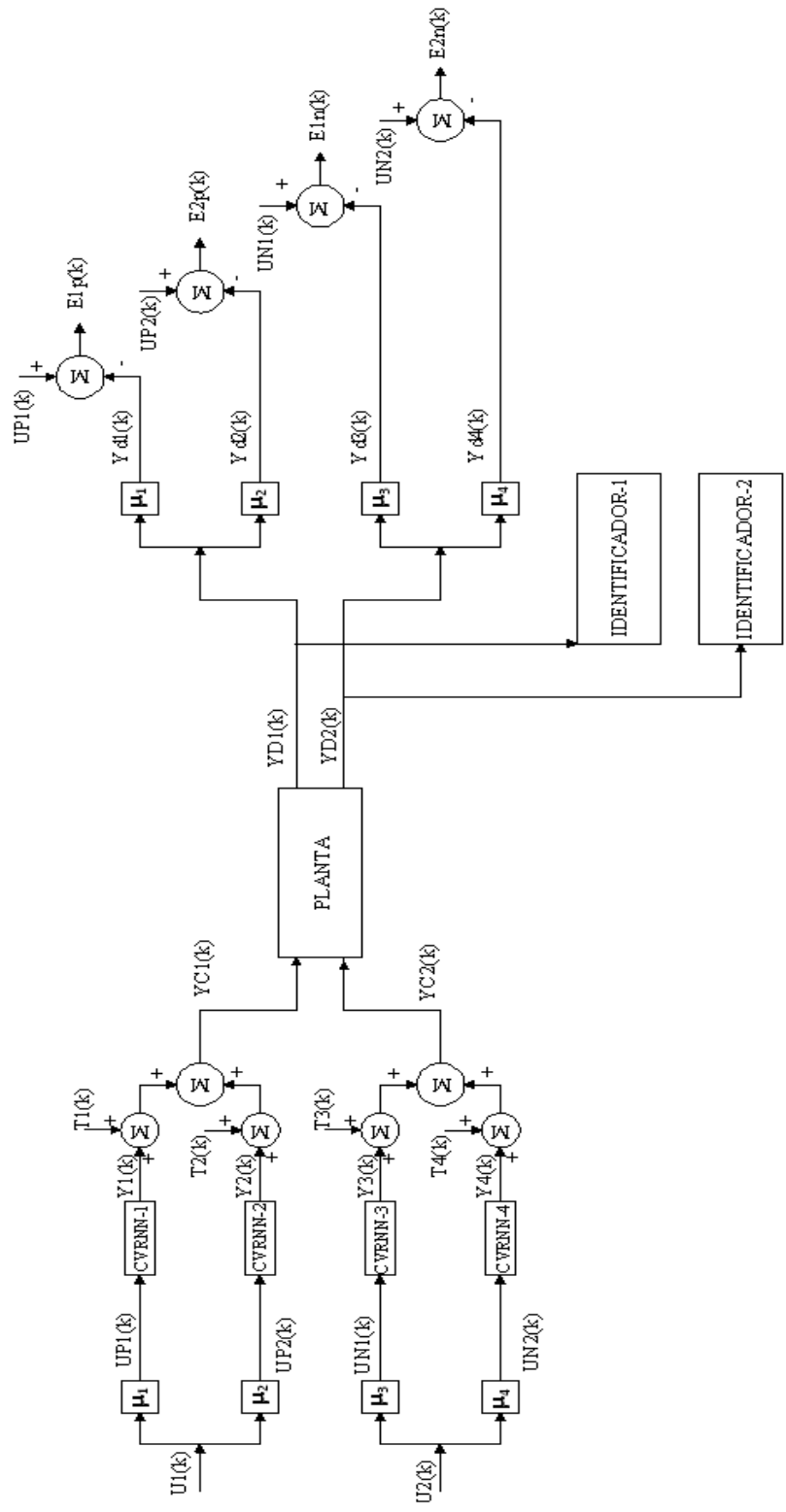


Figura 81 Esquema de Control Adaptable por Modelo Inverso con Termino Integral para una planta de dos grados de libertad

Al aplicar el esquema anterior, se compara el aprendizaje llevado a cabo por *Back-Propagation* y *Levenberg-Marquardt*, donde cada una de las CVRNN tienen:

Tabla 7.Tabla de parámetros para las CVRNN utilizadas para Control Adaptable por Modelo Inverso con Término Integral de una planta de dos grados de libertad.

<i>CVRNN</i>	n	p	m
<i>1</i>	4	1	1
<i>2</i>	4	1	1
<i>3</i>	4	1	1
<i>4</i>	4	1	1

Tabla 8.Tabla de los valores usados de las ganancias integrales para el Control Adaptable por Modelo Inverso con Término Integral.

<i>Ti</i>	Ki-BP	Ki-LM
<i>1</i>	0.02	0.02
<i>2</i>	0.02	0.02
<i>3</i>	0.02	0.02
<i>4</i>	0.02	0.02

Como entrada para el esquema de control usaron señales cuadradas las cuales están dadas por las ecuaciones descritas en 5.17:

5.6.1 Back-Propagation Salida 1 con Termino Integral

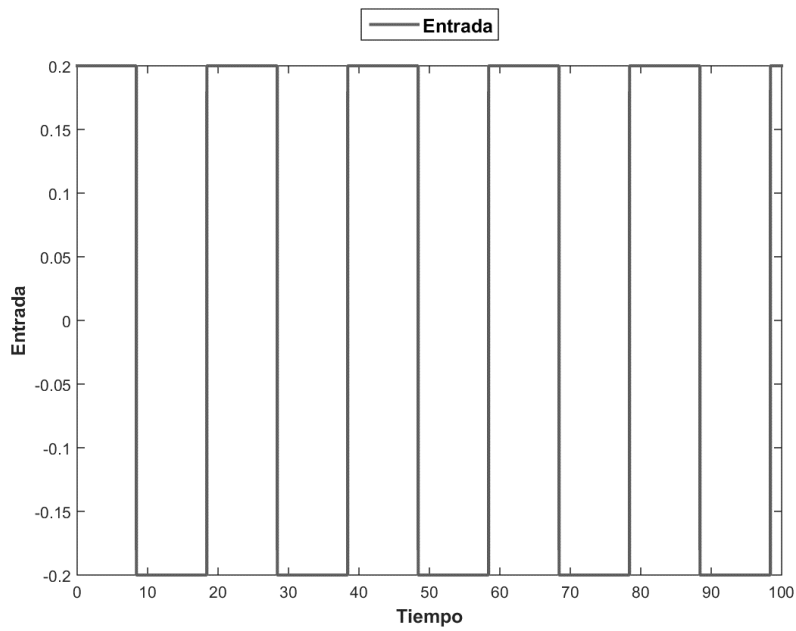


Figura 82. Entrada para la Señal de Salida 1 (BP-Ti).

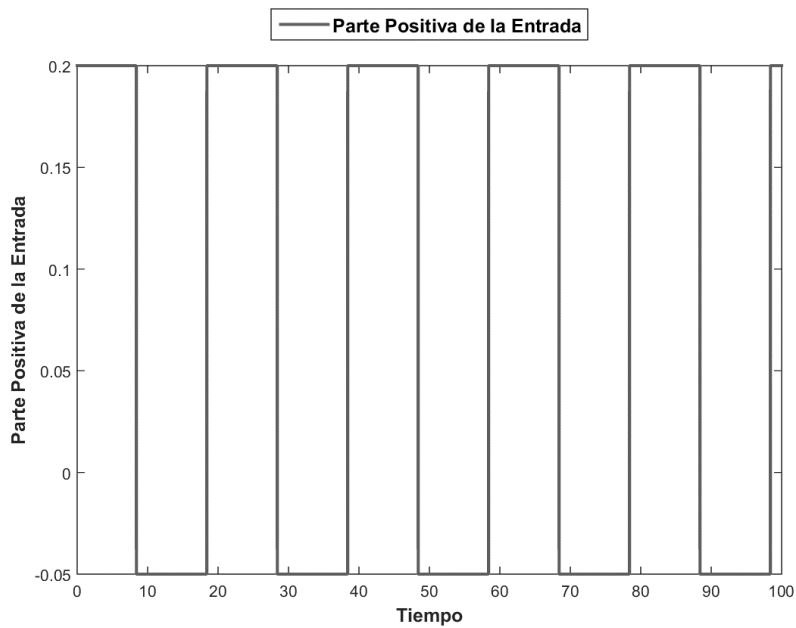


Figura 83. Parte Positiva de la Entrada de la Señal 1 (BP-Ti).

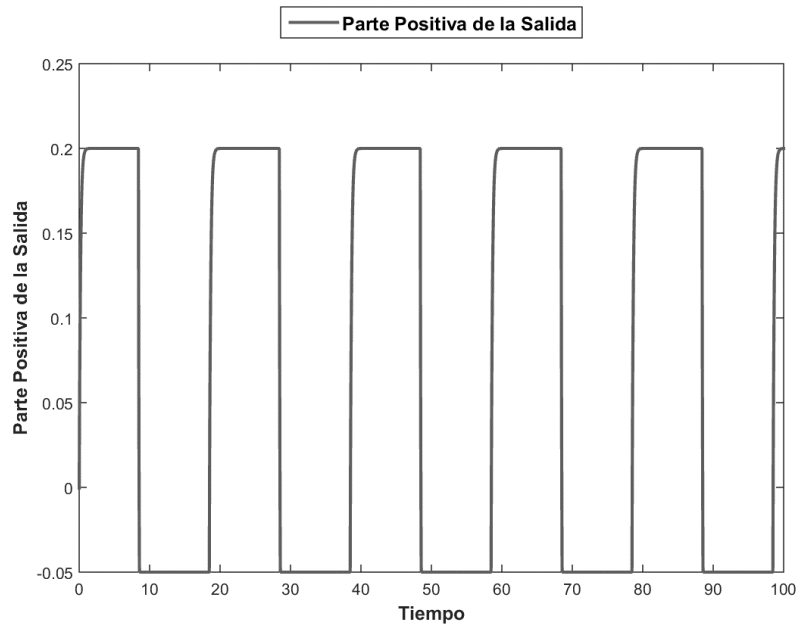


Figura 84. Parte Positiva de la Señal de Salida 1 (BP-Ti).

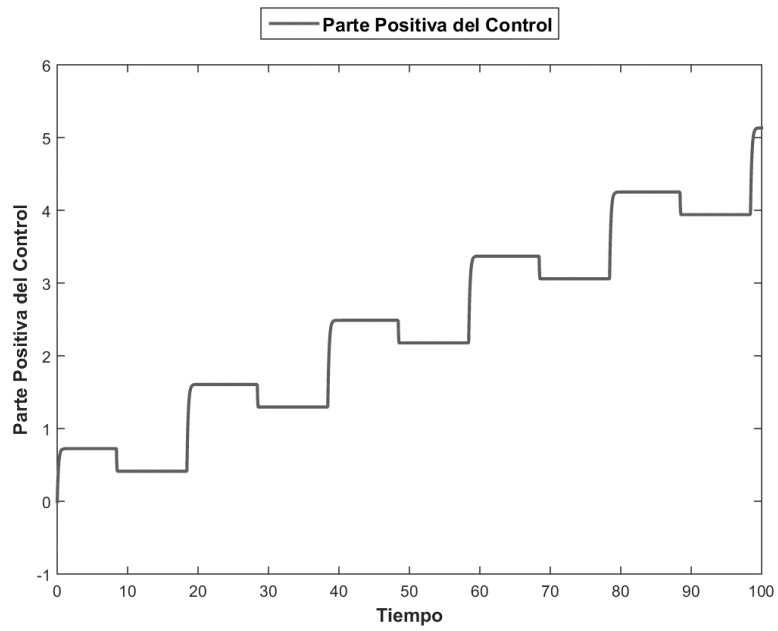


Figura 85. Parte Positiva del Control de la Señal de Salida 1 (BP-Ti).

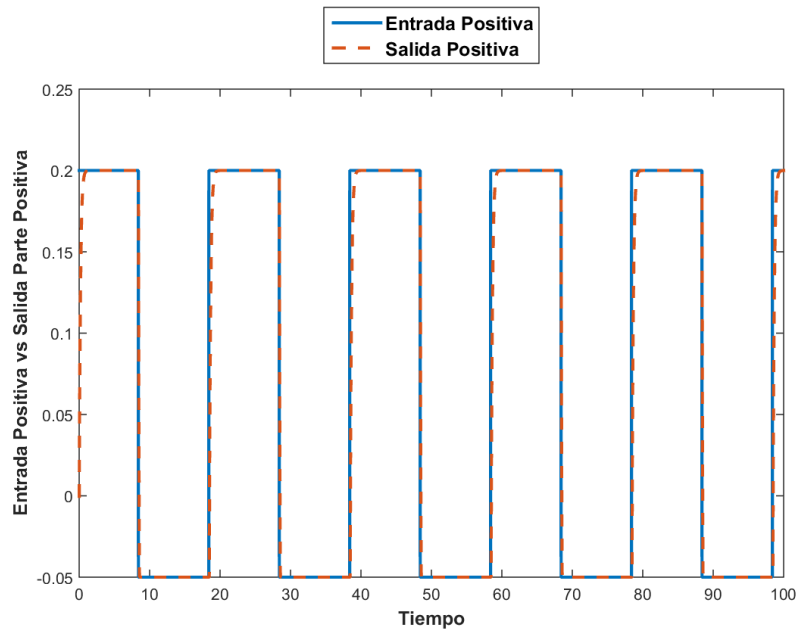


Figura 86. Comparación de la Parte Positiva de la Señal de Entrada con la Señal de Salida (BP-Ti).

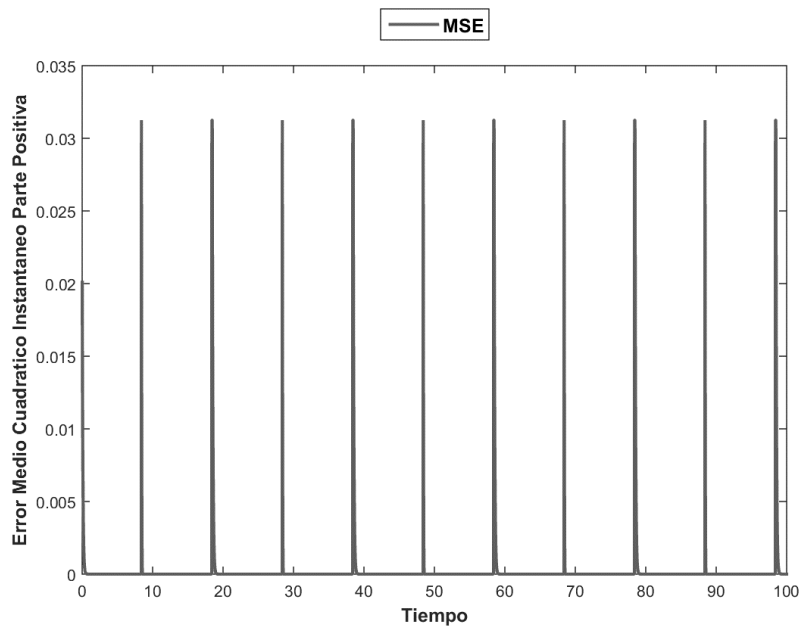


Figura 87. Error Medio Cuadrático Instantáneo de la Parte Positiva de la Señal 1 (BP-Ti).

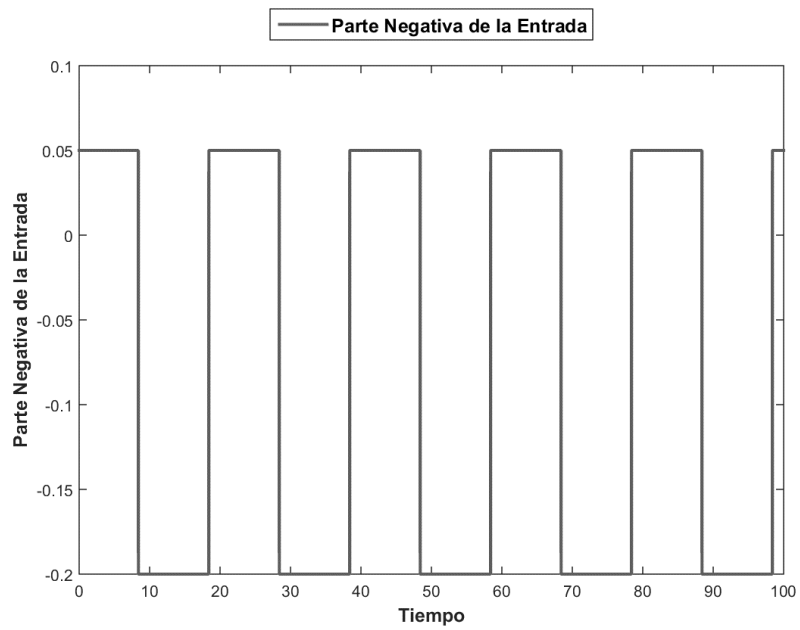


Figura 88. Parte Negativa de la Entrada de la Señal 1 (BP).

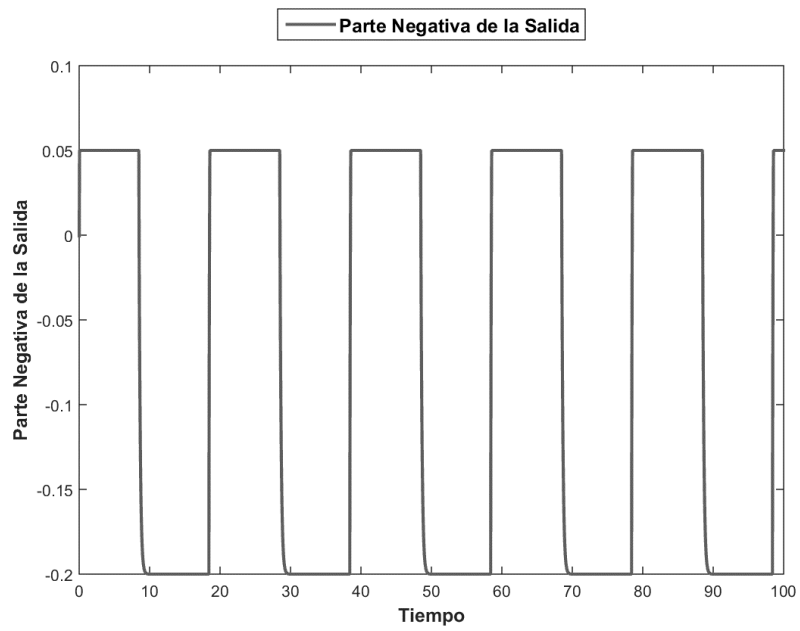


Figura 89. Parte Negativa de la Entrada de la Señal 1 (BP-Ti).

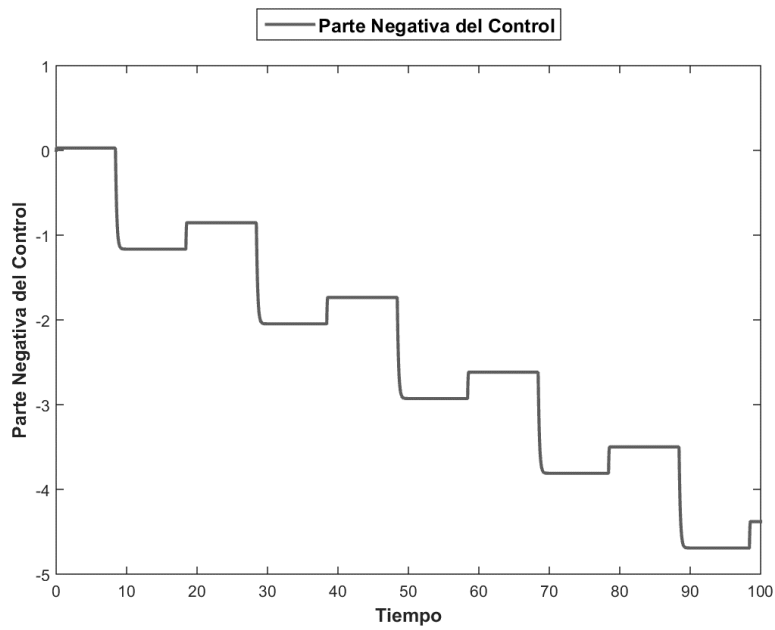


Figura 90. Parte Negativa del Control de la Señal de 1 (BP-Ti).

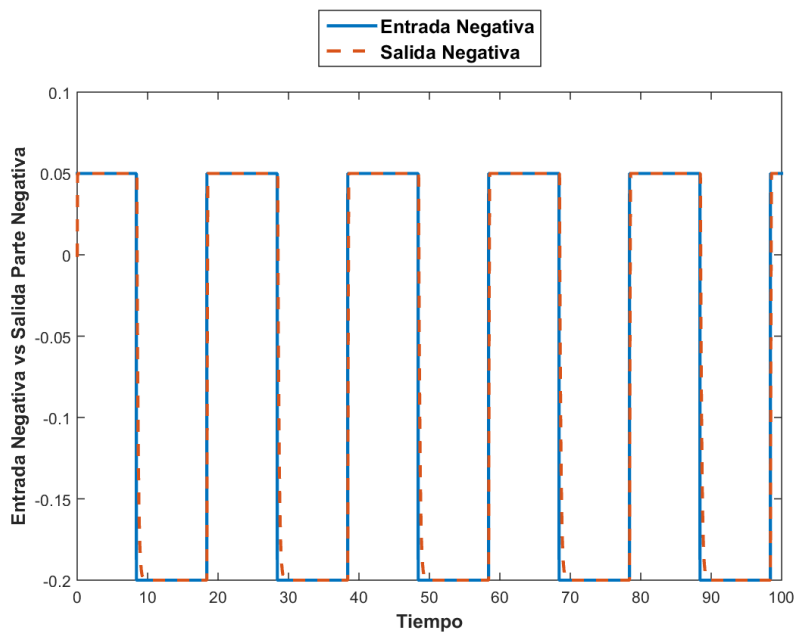


Figura 91. Comparación de la Parte Negativa de la Señal de Entrada con la Señal de Salida (BP-Ti).

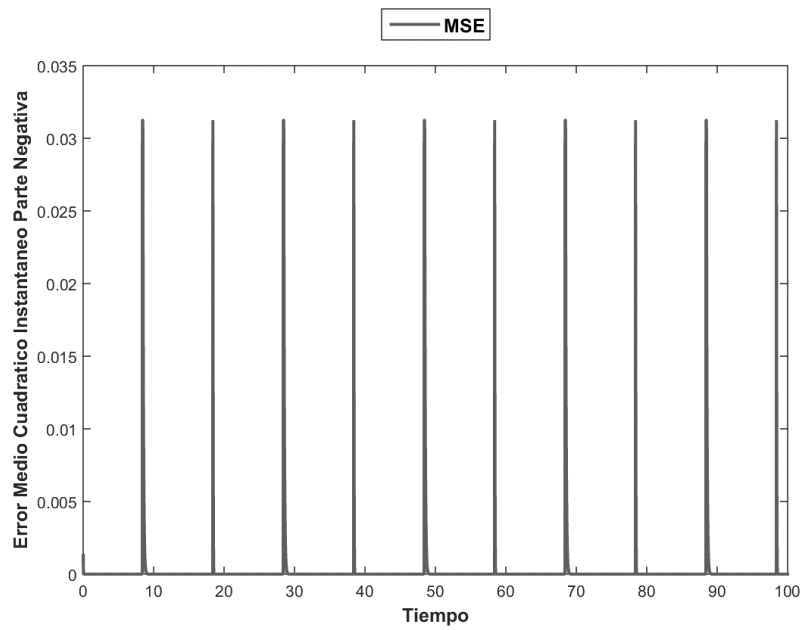


Figura 92. Error Medio Cuadrático Instantáneo de la Parte Negativa de la Señal 1 (BP-Ti).

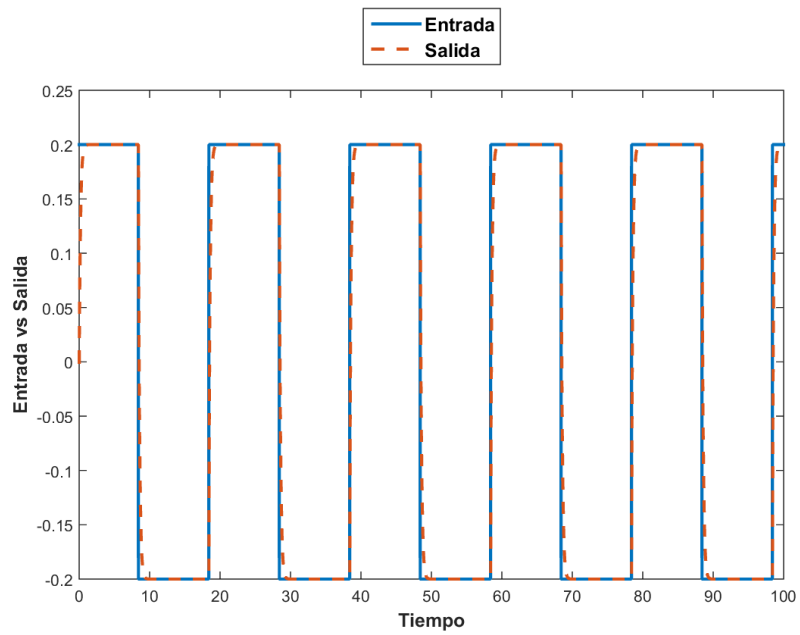


Figura 93. Comparación de la Señal de Entrada con la Señal de Salida para la Señal 1 (BP-Ti).

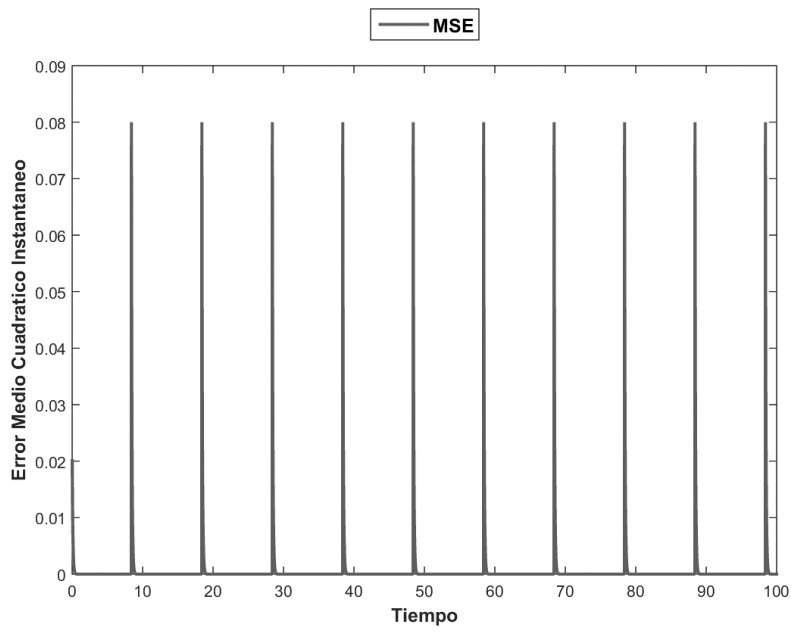


Figura 94. Error Medio Cuadrático Instantáneo de la Señal 1 (BP-Ti).

Nota: El tiempo está dado en segundos.

5.6.2 Back-Propagation Salida 2 con Termino Integral

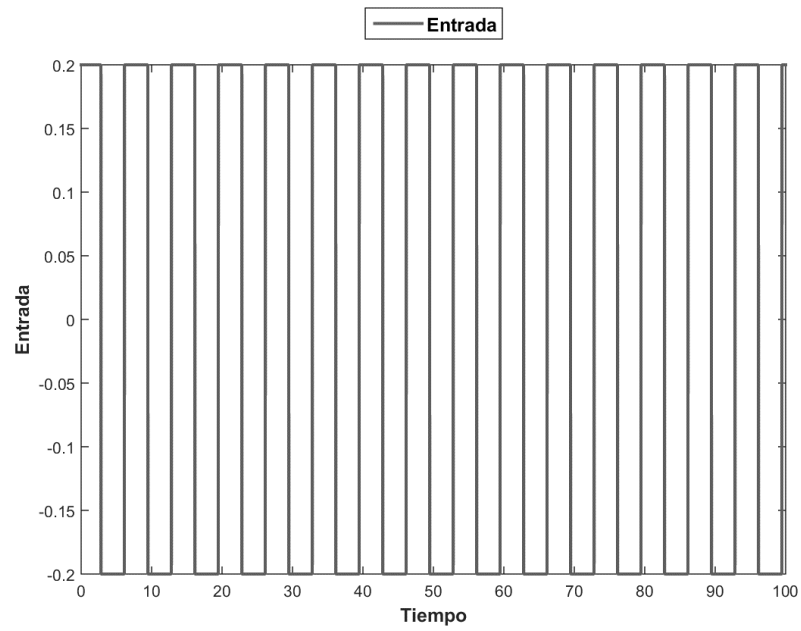


Figura 95. Entrada para la Señal de Salida 2 (BP-Ti).

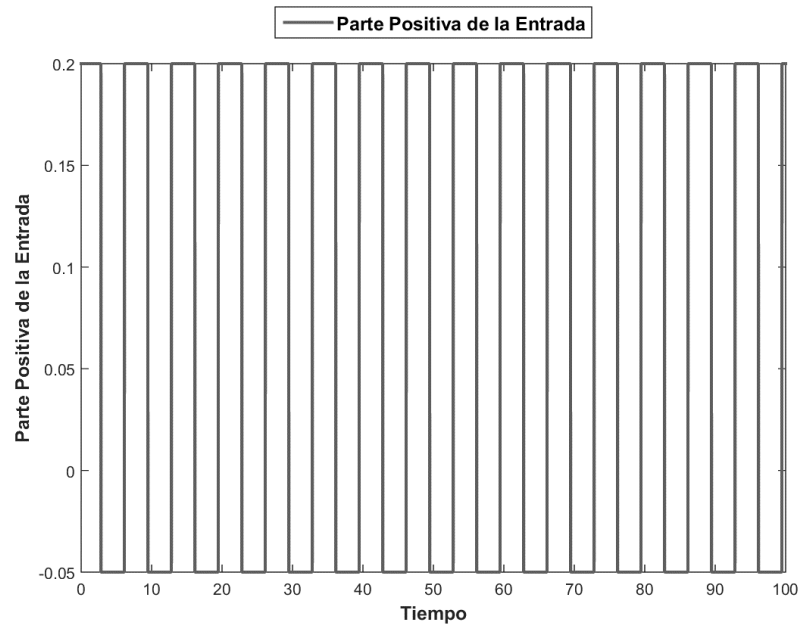


Figura 96. Parte Positiva de la Entrada de la Señal 2 (BP-Ti).

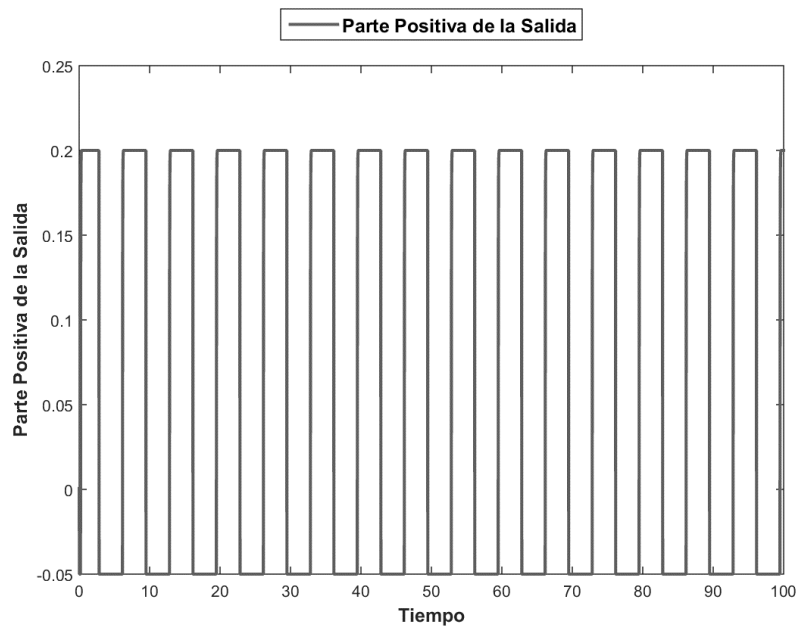


Figura 97. Parte Positiva de la Señal de Salida 2 (BP-Ti).

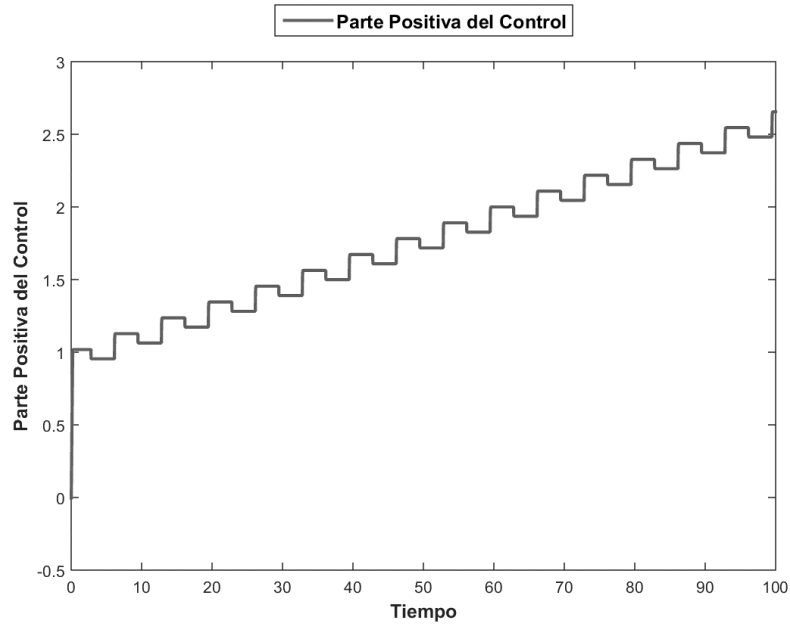


Figura 98. Parte Positiva del Control de la Señal de Salida 2 (BP-Ti).

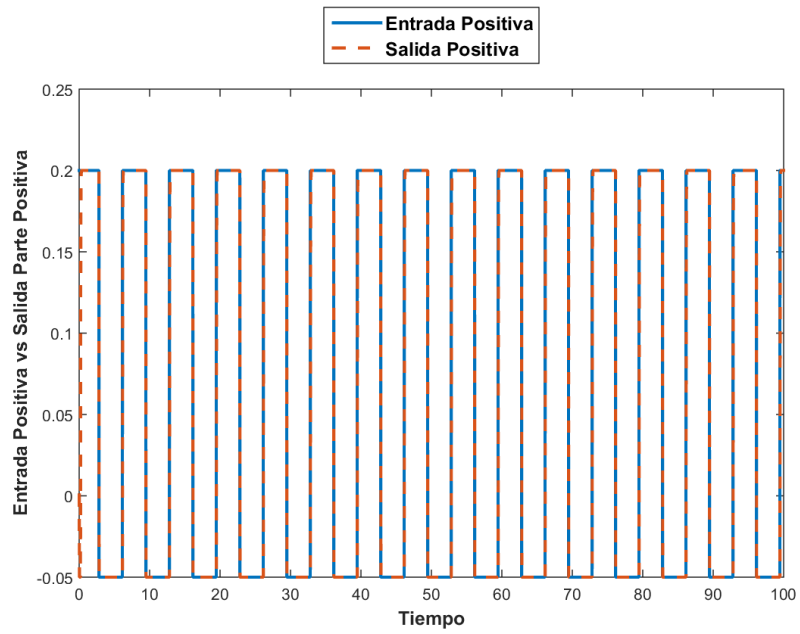


Figura 99. Comparación de la Parte Positiva de la Señal de Entrada con la Señal de Salida (BP-Ti).

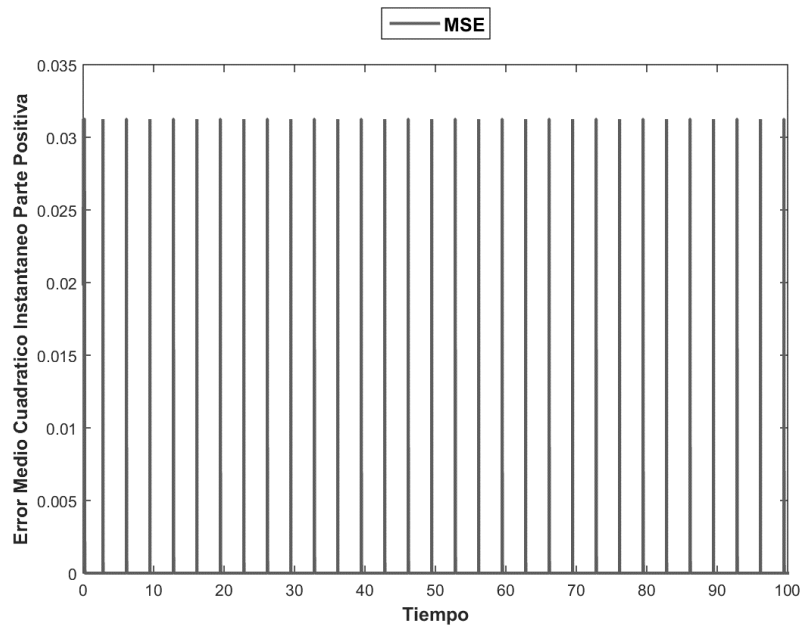


Figura 100. Error Medio Cuadrático Instantáneo de la Parte Positiva de la Señal 2 (BP-Ti).

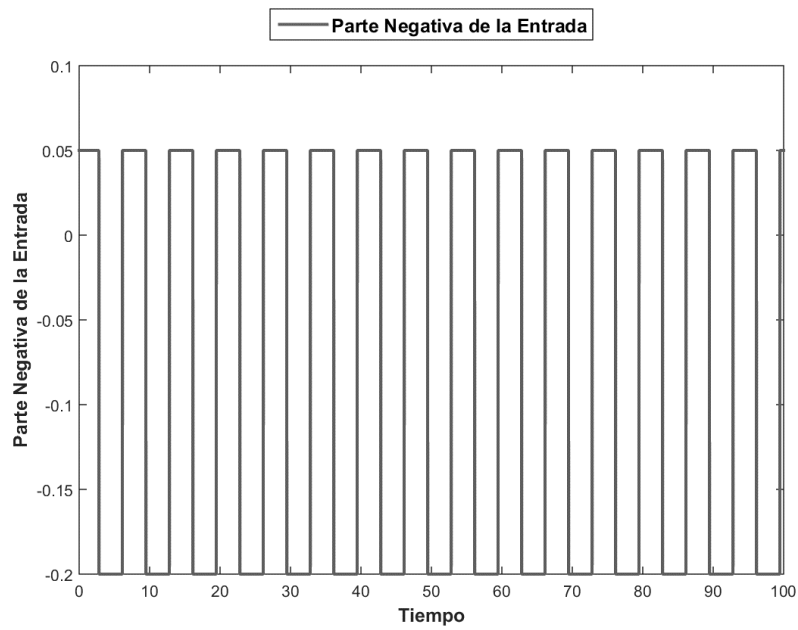


Figura 101. Parte Negativa de la Entrada de la Señal 2 (BP-Ti).

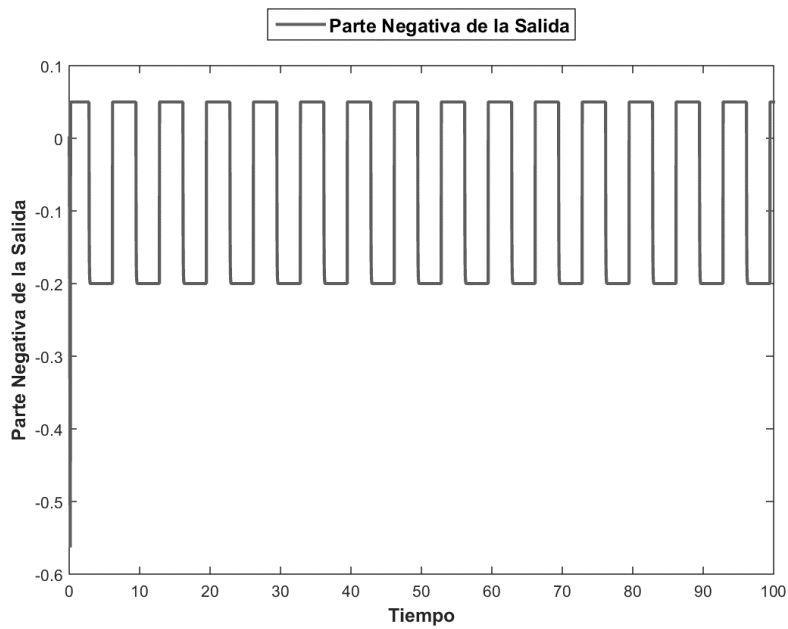


Figura 102. Parte Negativa de la Salida de la Señal de 2 (BP-Ti).

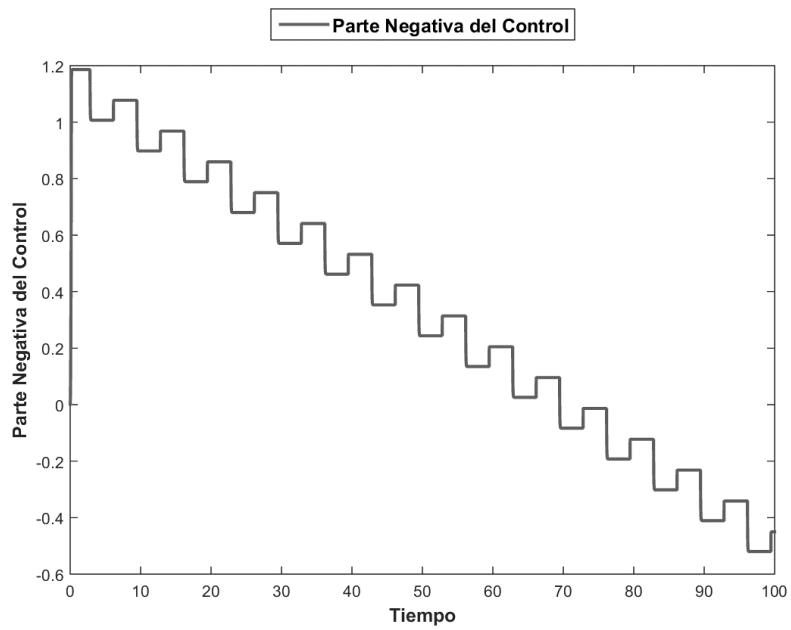


Figura 103. Parte Negativa del Control de la Señal de 2 (BP-Ti).

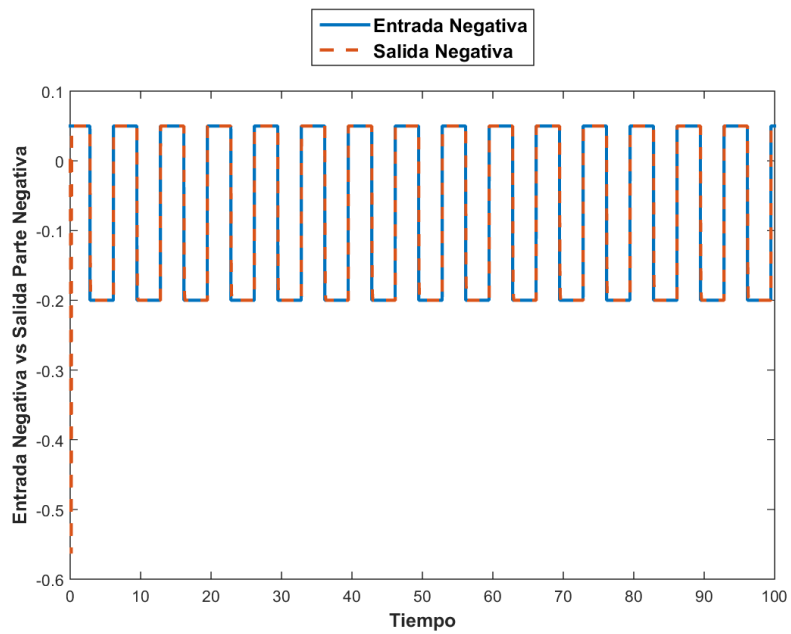


Figura 104. Comparación de la Parte Negativa de la Señal de Entrada con la Señal de Salida (BP-Ti).

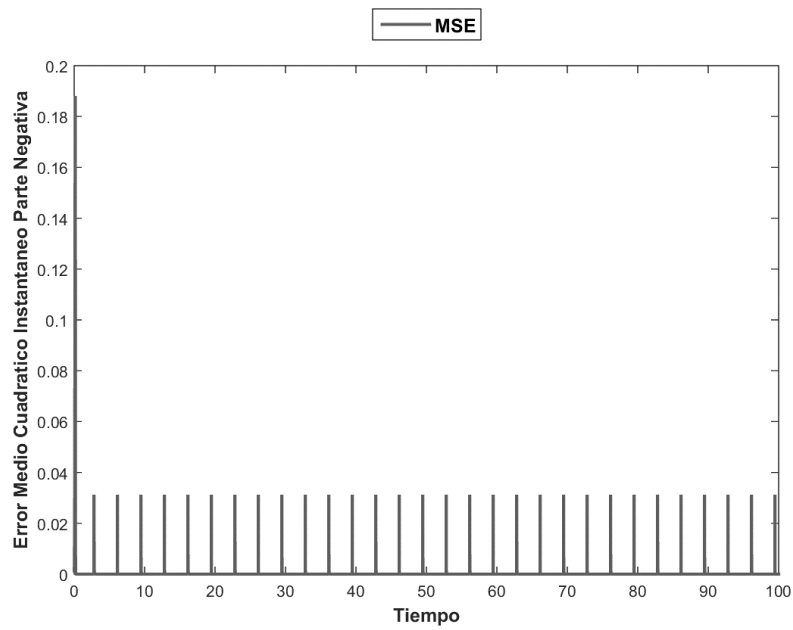


Figura 105. Error Medio Cuadrático Instantáneo de la Parte Negativa de la Señal 2 (BP-Ti).

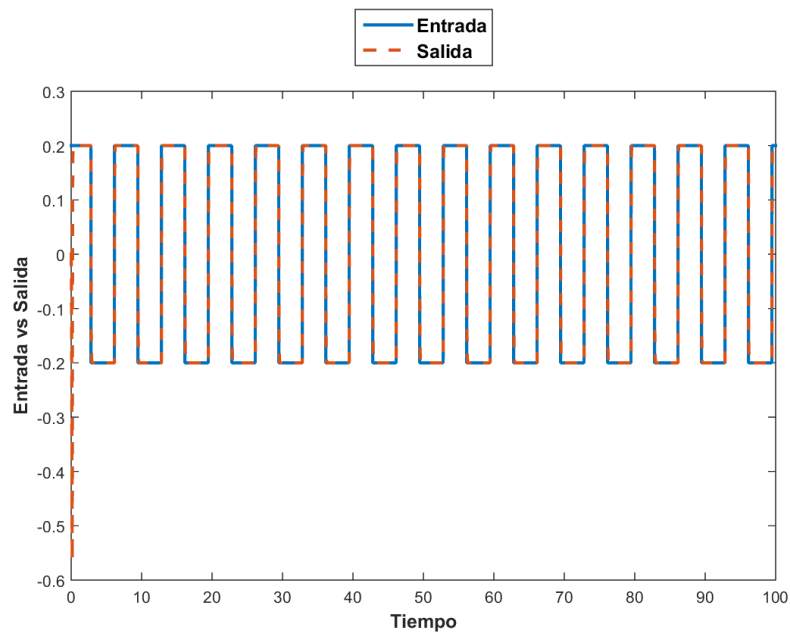


Figura 106. Comparación de la Señal de Entrada con la Señal de Salida para la Señal 2 (BP-Ti).

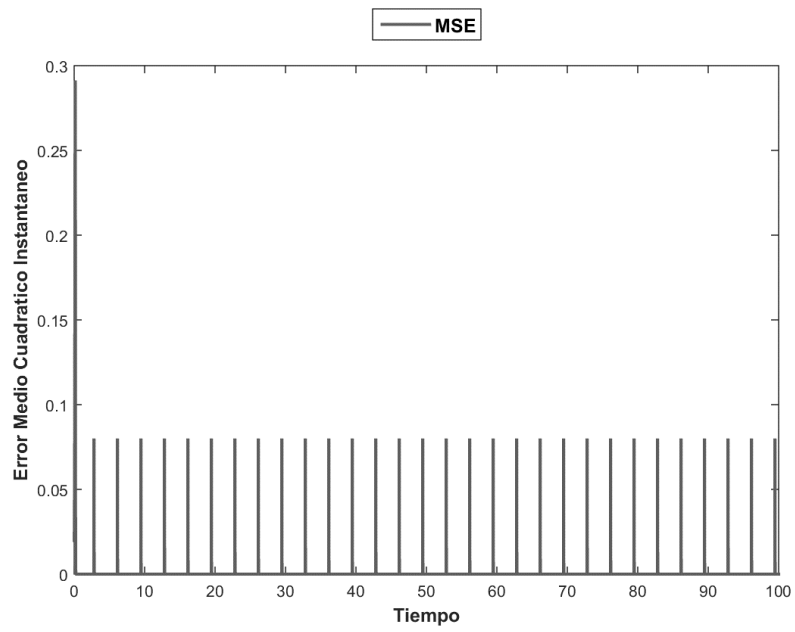


Figura 107. Error Medio Cuadrático Instantáneo de la Señal 2 (BP-Ti).

Nota: El tiempo está dado en segundos.

5.6.3 Levenberg-Marquardt Salida1 con Termino Integral

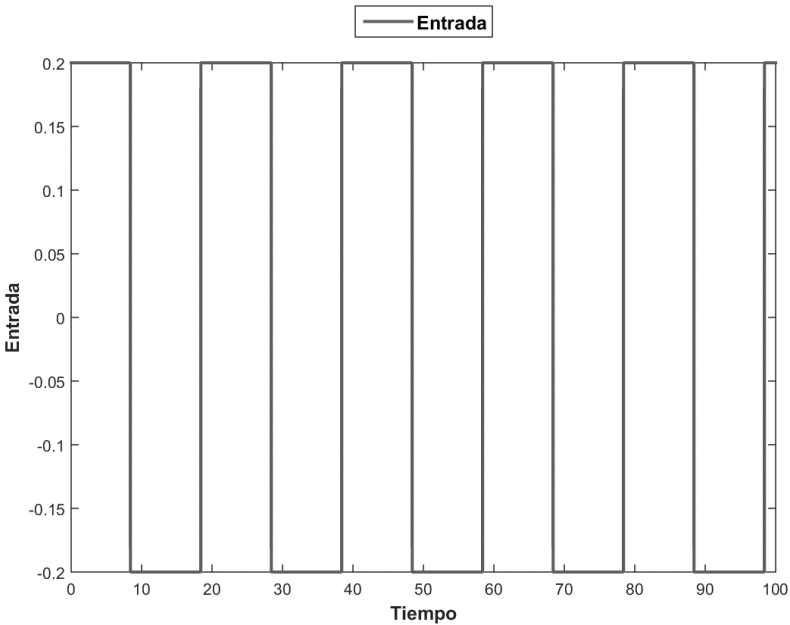


Figura 108. Entrada para la Señal de Salida 1 (LM-Ti).

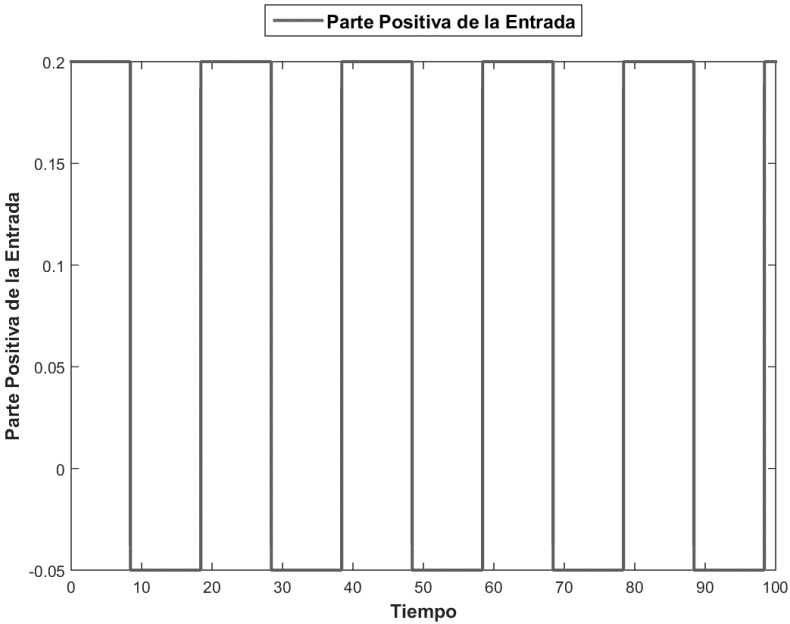


Figura 109. Parte Positiva de la Entrada de la Señal 1 (LM-Ti).

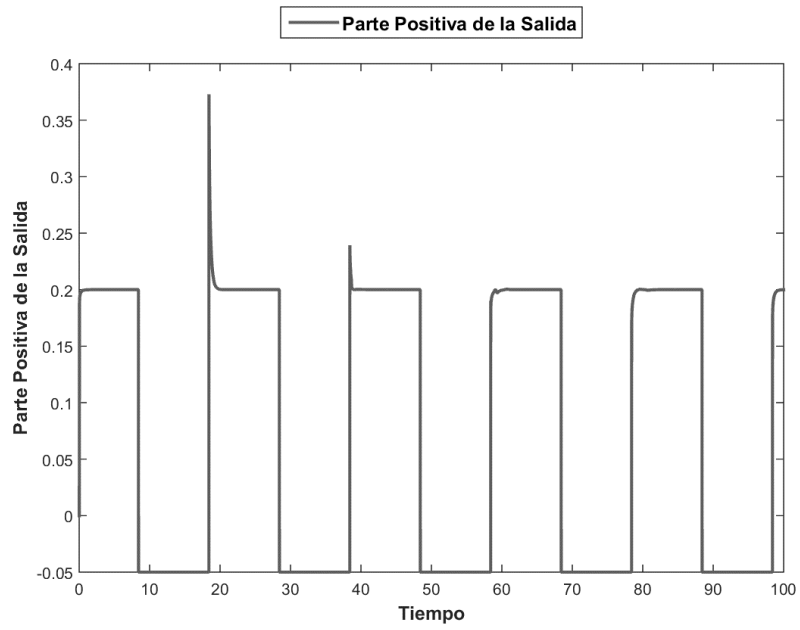


Figura 110. Parte Positiva de la Señal de Salida 1 (LM-Ti).

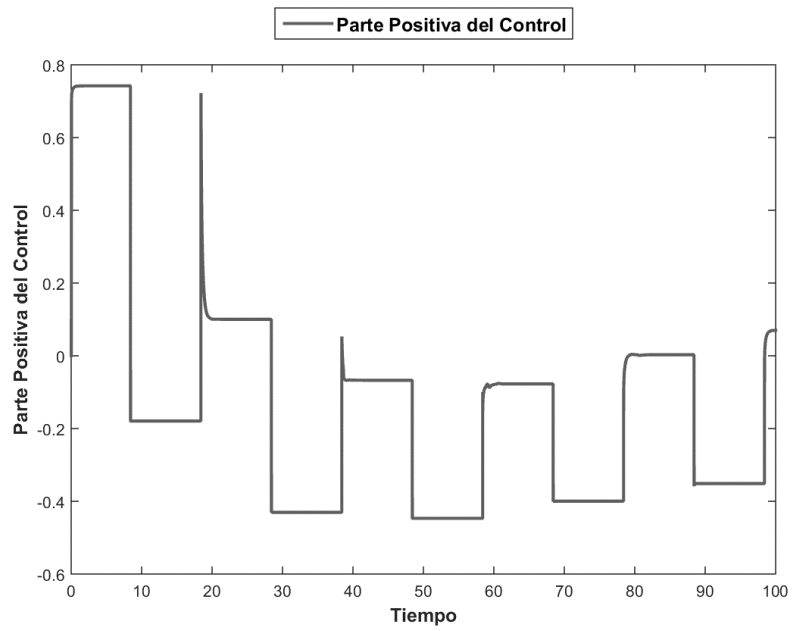


Figura 111. Parte Positiva del Control de la Señal de Salida 1 (LM-Ti).

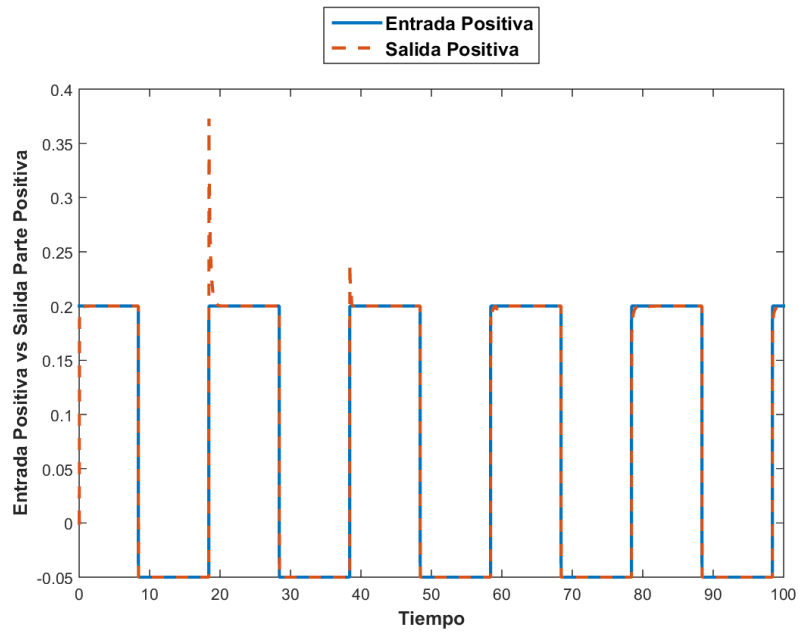


Figura 112. Comparación de la Parte Positiva de la Señal de Entrada con la Señal de Salida (LM-Ti).

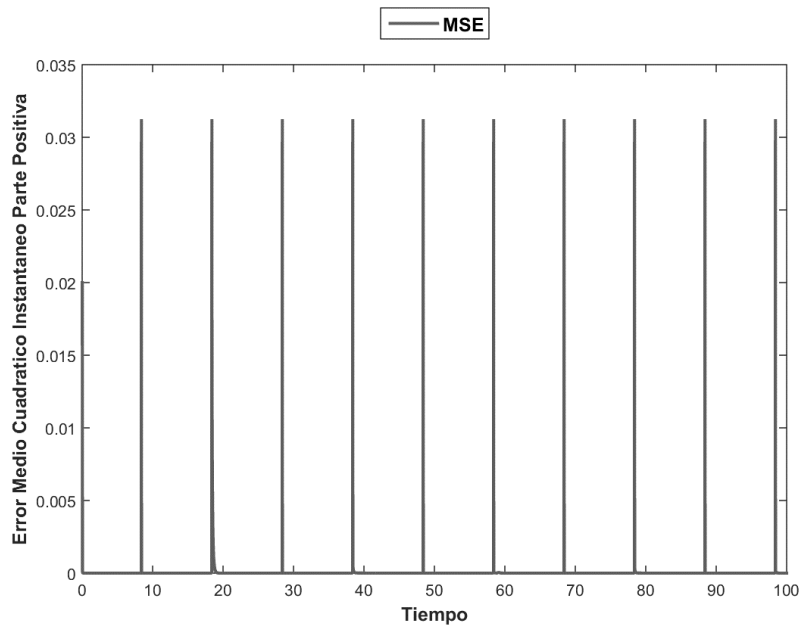


Figura 113. Error Medio Cuadrático Instantáneo de la Parte Positiva de la Señal 1 (LM-Ti).

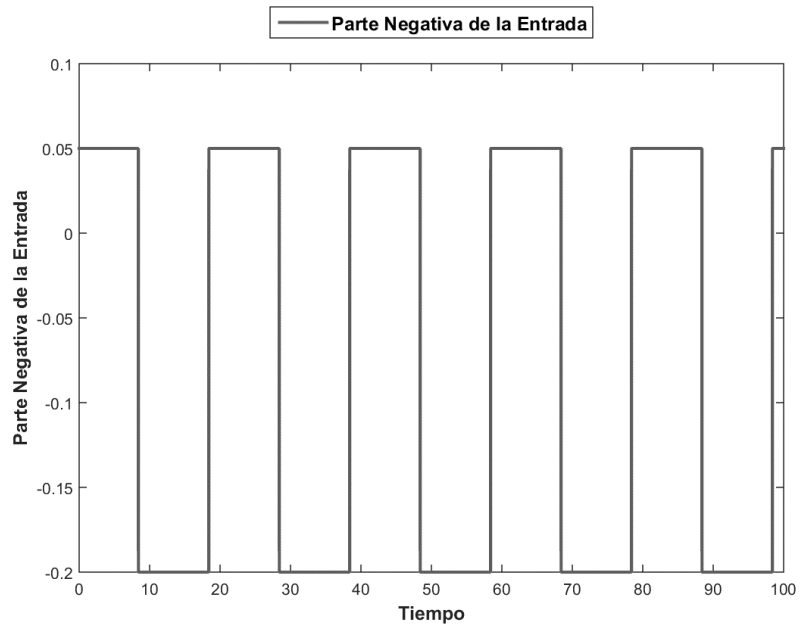


Figura 114. Parte Negativa de la Entrada de la Señal 1 (LM-Ti).

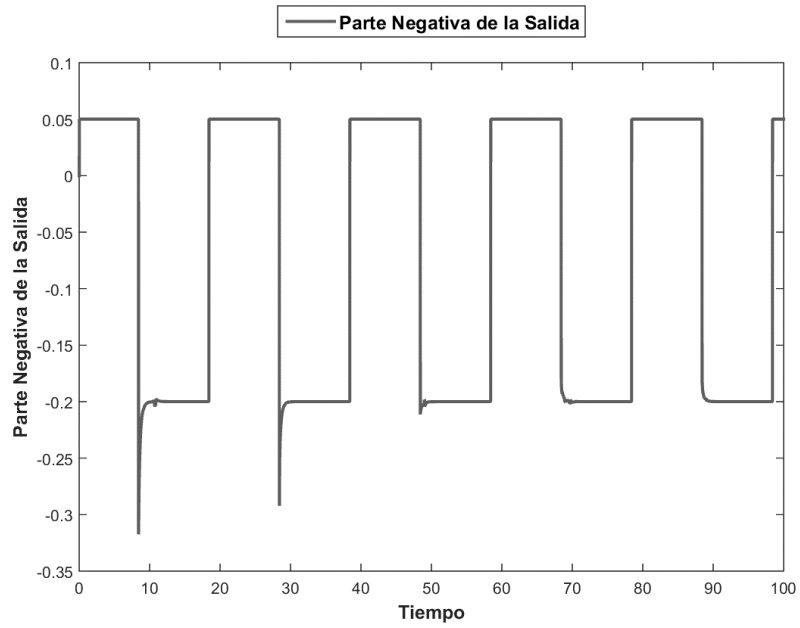


Figura 115. Parte Negativa del Control de la Señal de 1 (LM-Ti).

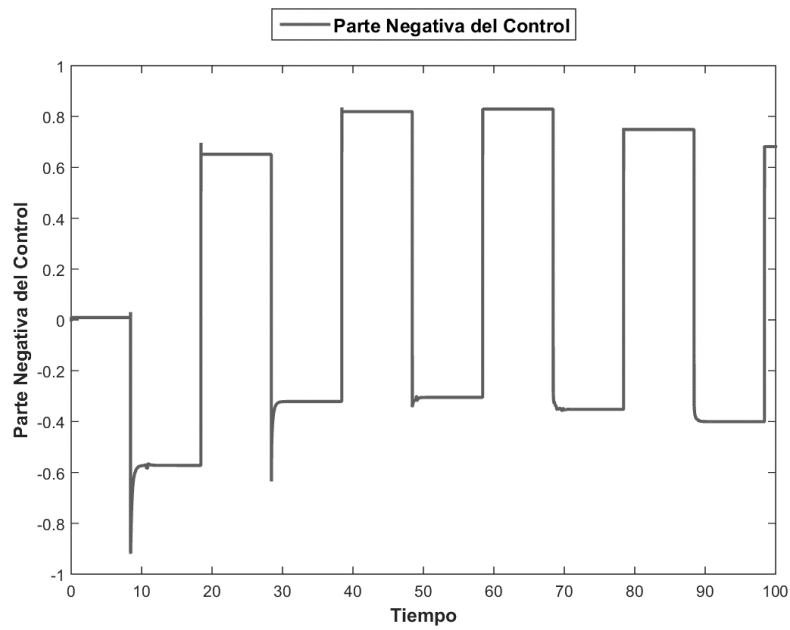


Figura 116. Parte Negativa del Control de la Señal de 1 (LM-Ti).

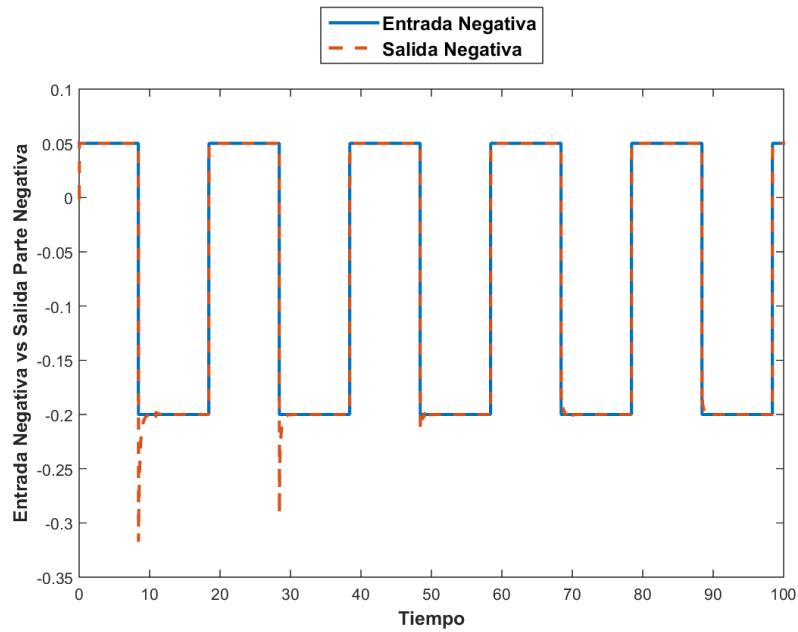


Figura 117. Comparación de la Parte Negativa de la Señal de Entrada con la Señal de Salida (LM-Ti).

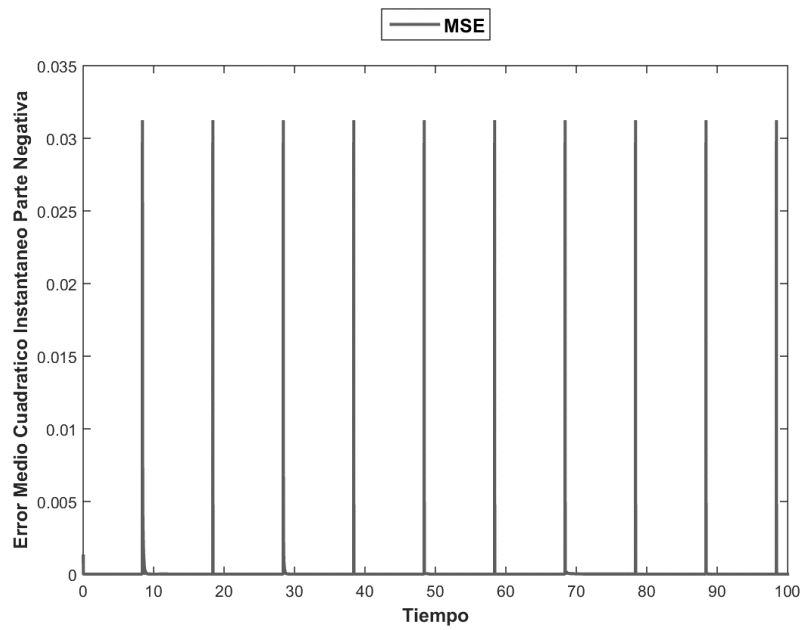


Figura 118. Error Medio Cuadrático Instantáneo de la Parte Negativa de la Señal 1 (LM-Ti).

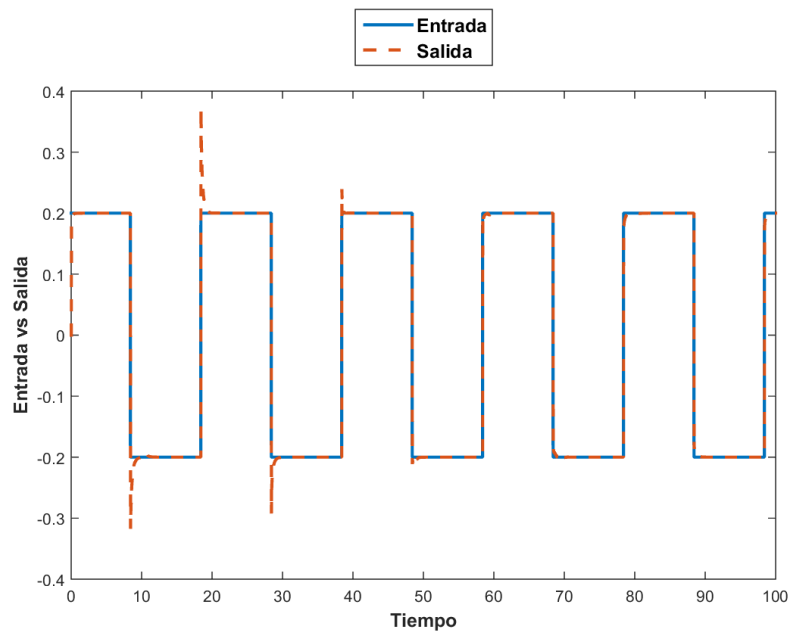


Figura 119. Comparación de la Señal de Entrada con la Señal de Salida para la Señal 1 (LM-Ti).

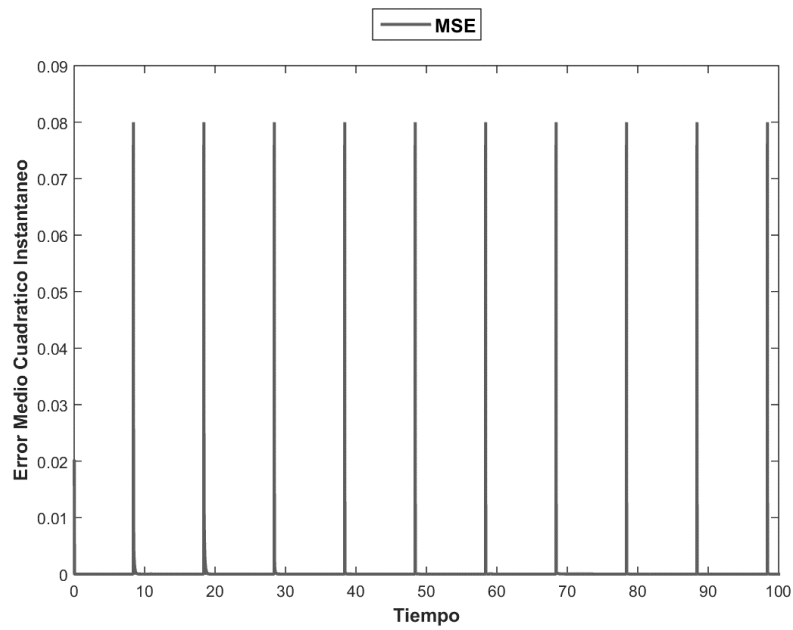


Figura 120. Error Medio Cuadrático Instantáneo de la Señal 1 (LM-Ti).

Nota: El tiempo está dado en segundos.

5.6.4 Levenberg-Marquardt Salida 2 con Termino Integral

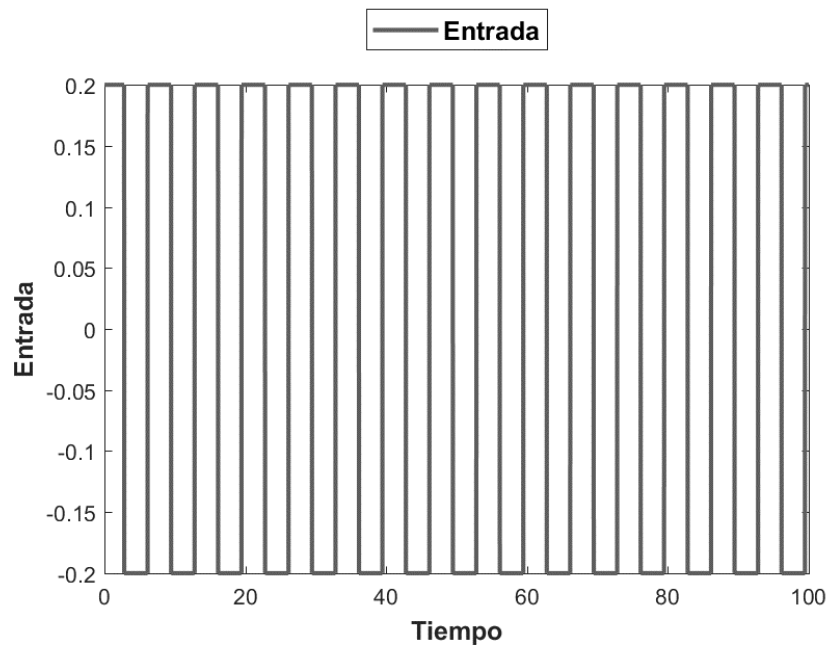


Figura 121. Entrada para la Señal de Salida 2 (LM-Ti).

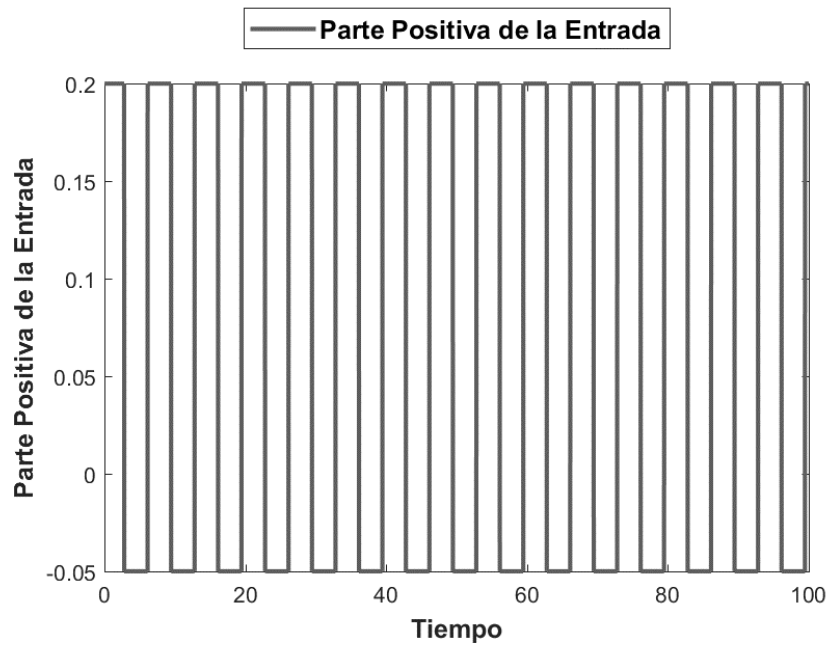


Figura 122. Parte Positiva de la Entrada de la Señal 2 (LM-Ti).

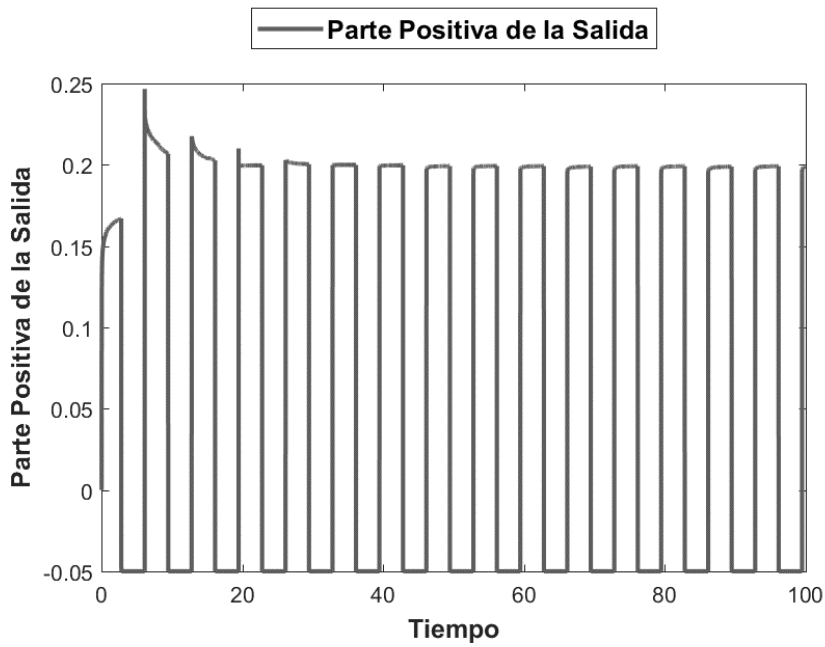


Figura 123. Parte Positiva de la Señal de Salida 2 (LM-Ti).

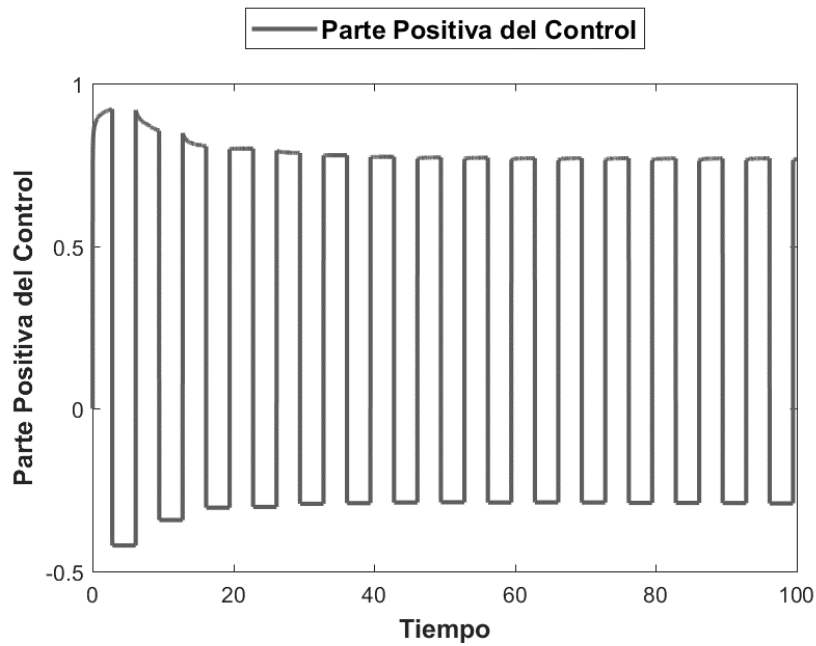


Figura 124. Parte Positiva del Control de la Señal de Salida 2 (LM-Ti).

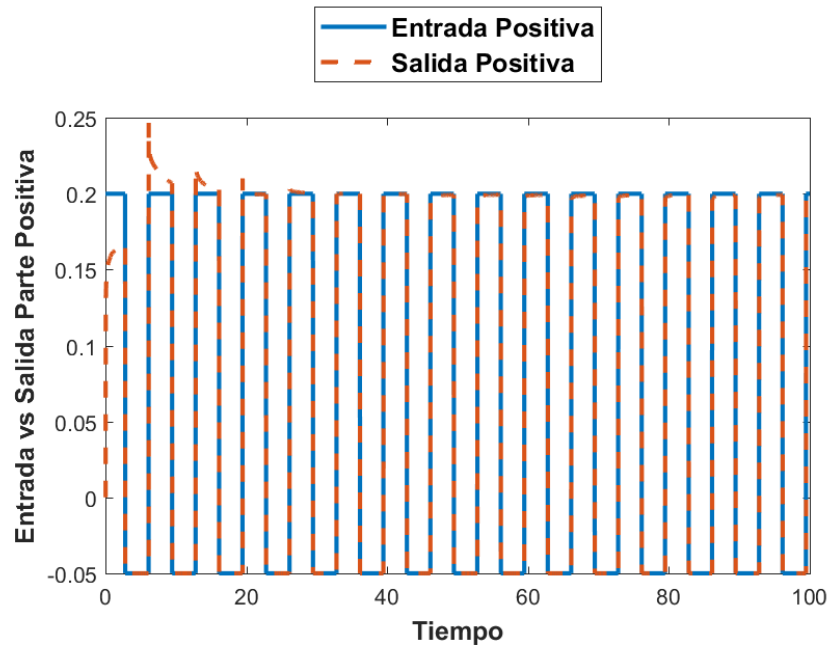


Figura 125. Comparación de la Parte Positiva de la Señal de Entrada con la Señal de Salida (LM-Ti).

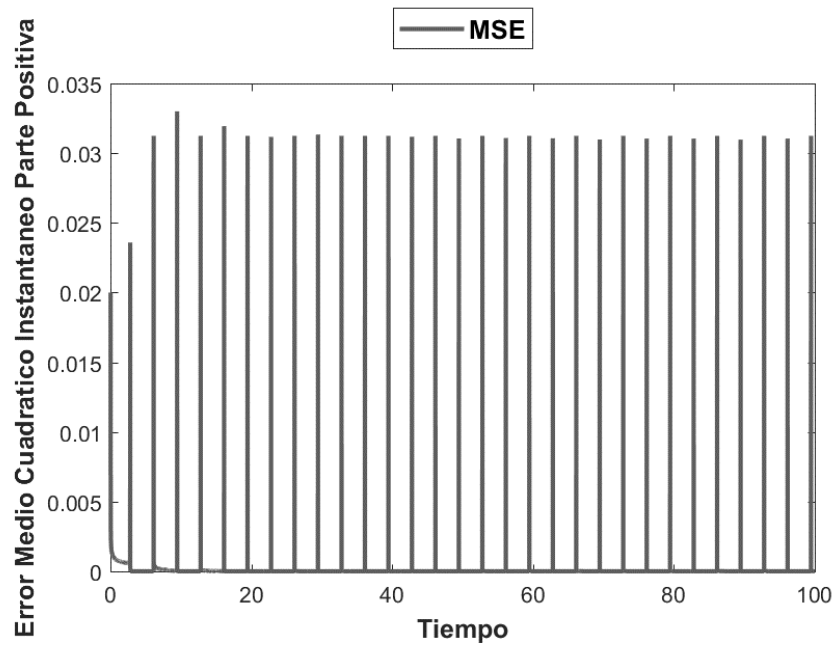


Figura 126. Error Medio Cuadrático Instantáneo de la Parte Positiva de la Señal 2 (LM-Ti).

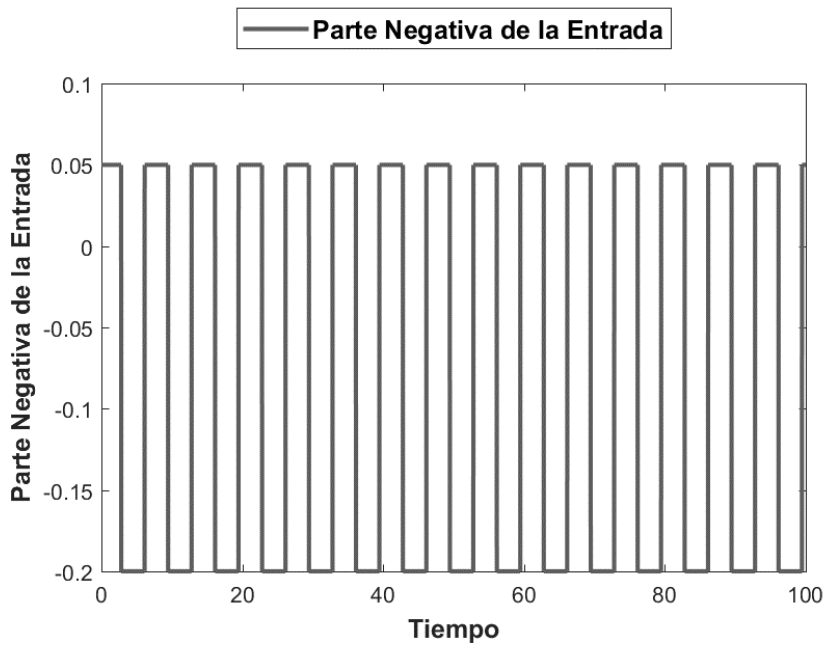


Figura 127. Parte Negativa de la Entrada de la Señal 2 (LM-Ti).

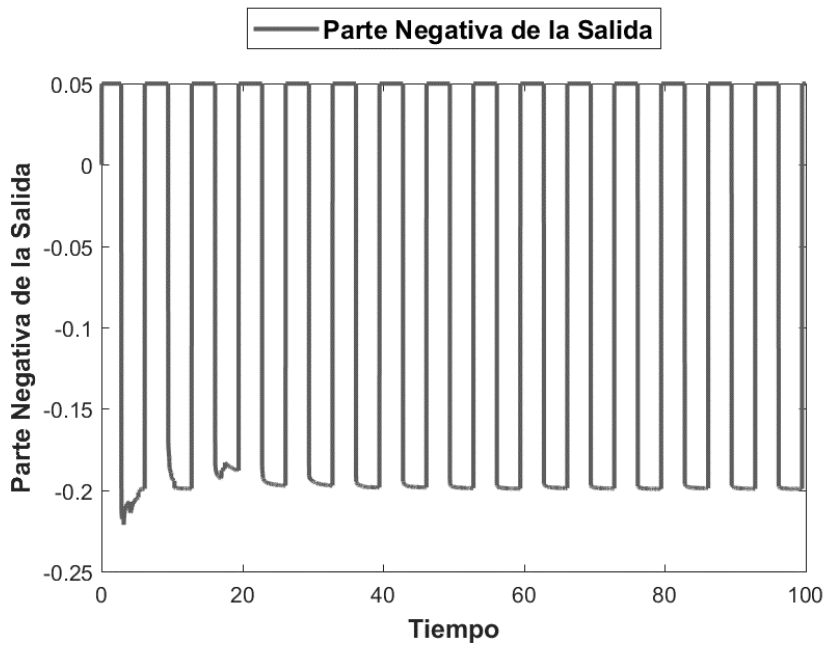


Figura 128. Parte Negativa de la Salida de la Señal de 2 (LM-Ti).

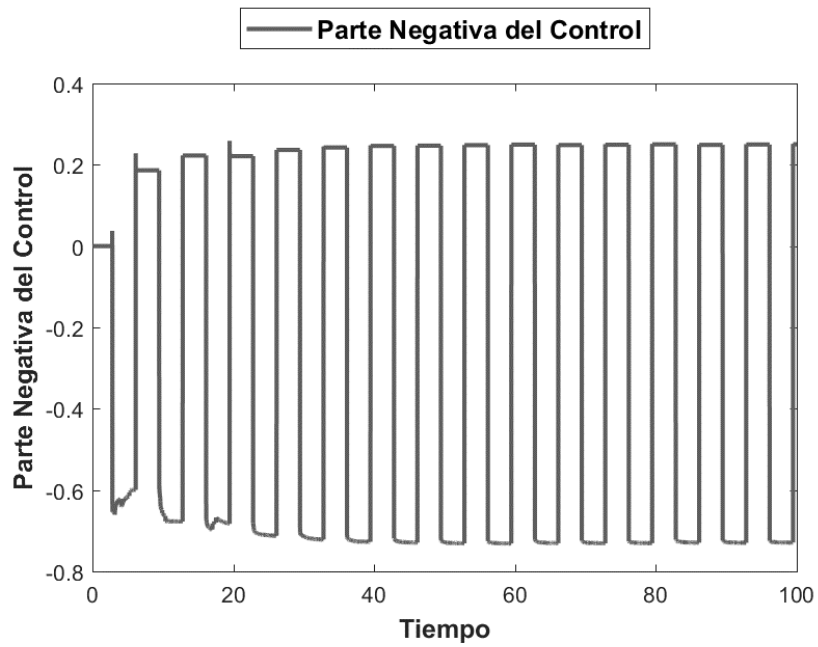


Figura 129. Parte Negativa del Control de la Señal de 2 (LM-Ti).

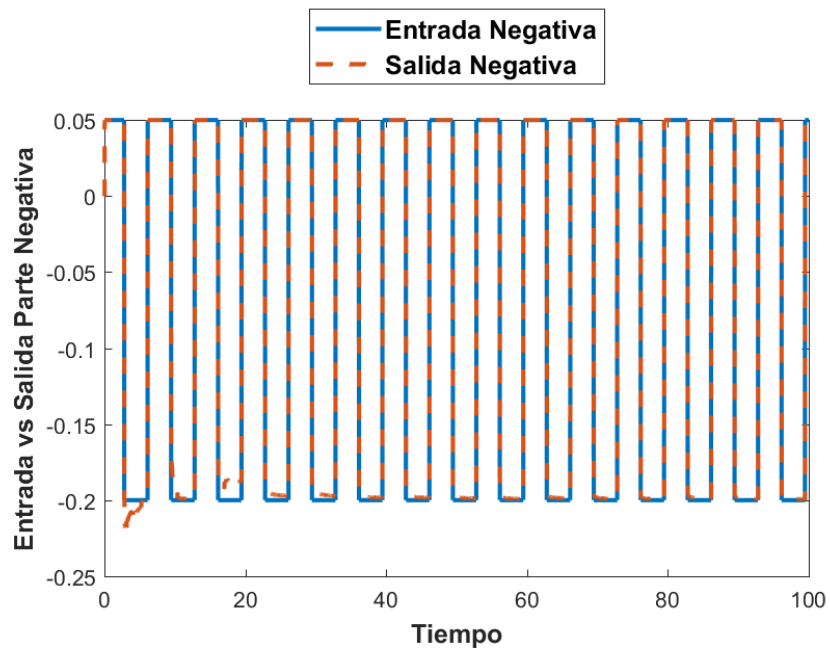


Figura 130. Comparación de la Parte Negativa de la Señal de Entrada con la Señal de Salida (LM-Ti).

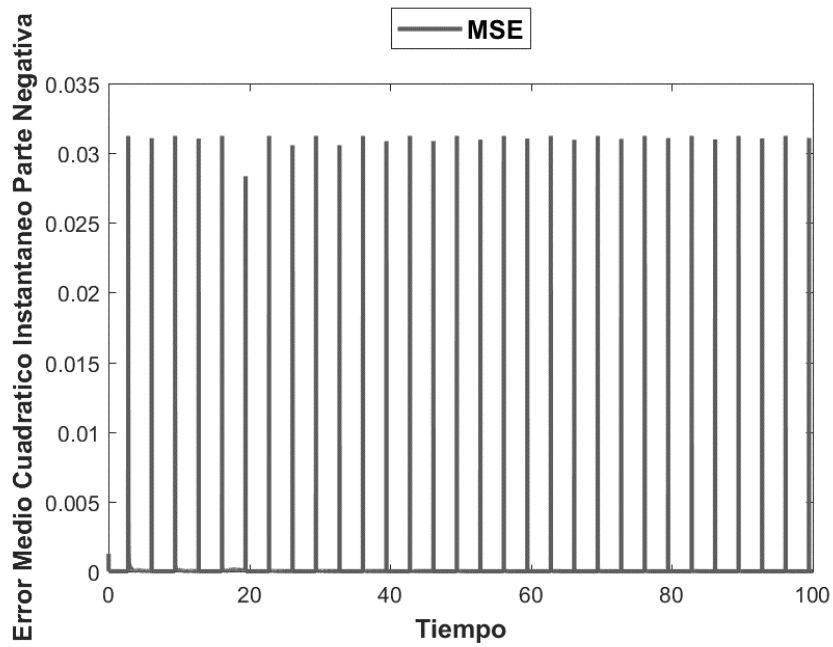


Figura 131. Error Medio Cuadrático Instantáneo de la Parte Negativa de la Señal 2 (LM-Ti).

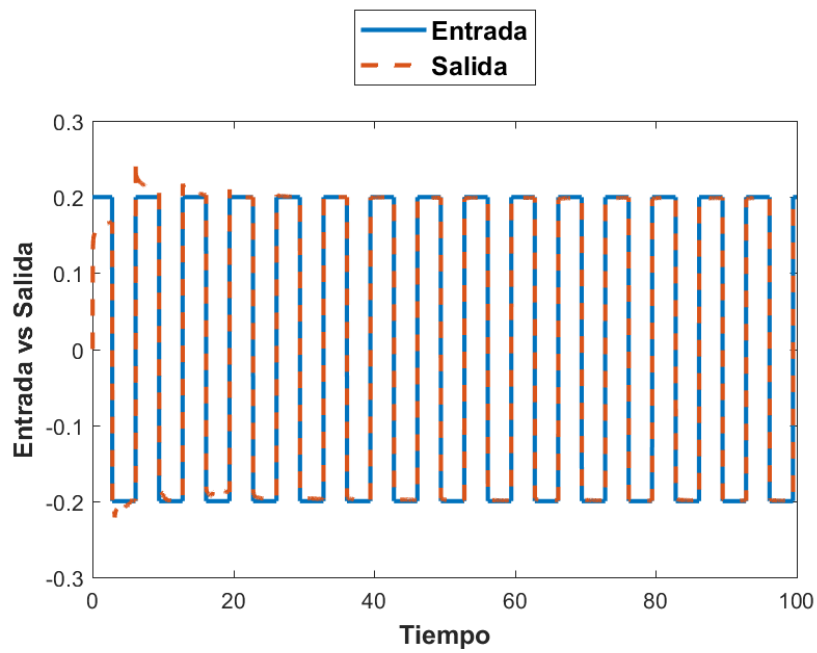


Figura 132. Comparación de la Señal de Entrada con la Señal de Salida para la Señal 2 (LM-Ti).

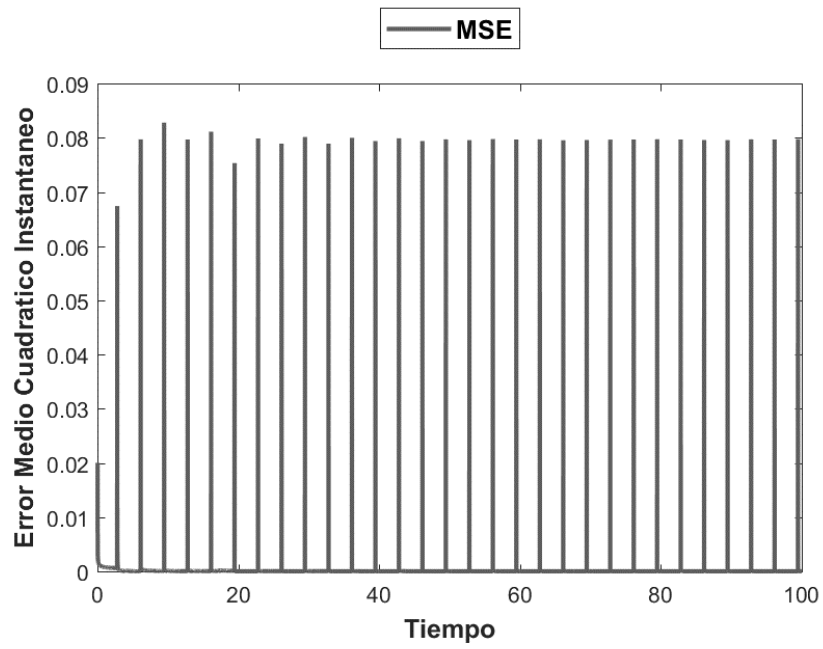


Figura 133. Error Medio Cuadrático Instantáneo de la Señal 2 (LM-Ti).

Nota: El tiempo está dado en segundos.

5.7 Control por Sliding-Mode con termino Integral

A modo de plantear un problema a futuro se propone que en el esquema de control figura 133, una de las entradas de la planta uno y dos grados de libertad sea controlada por un control Sliding-Mode, este control se encuentra dado por la siguiente ecuación:

$$U_{eq}(k) = (C \cdot B)^{-1} \left[-C \cdot J \cdot A(k) + R(k+1) + \sum_{i=1}^p \gamma_i E(k-i+1) \right]; |\gamma_i| < 1 \quad (5.25)$$

$$U^*(k) = \begin{cases} U_{eq}(k) & \|U_{eq}(k)\| < U_0 \\ -U_0 \cdot U_{eq}(k) / \|U_{eq}(k)\| & \|U_{eq}(k)\| \geq U_0 \end{cases} \quad (5.26)$$

Donde U_0 es el límite superior de control, γ_i es un parámetro deseado para el error del modelo.

De esa ecuación se puede observar que la estructura de control depende las matrices A, B, C, por lo que la etapa de identificación tiene que ser adecuada. Cabe destacar que el control necesita la parte real de las matrices antes descritas, así como la parte real del error de control.

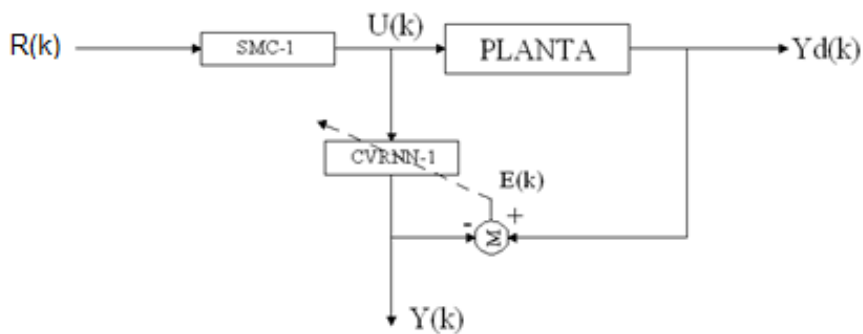


Figura 134. Esquema de Control Por Sliding-Mode.

5.7.1 Sliding-Mode Control con i-term para una planta de un Grado de Libertad

Se ejemplifica con un método de aprendizaje Back-Propagation esta estrategia de control obteniendo los siguientes resultados, bajo condiciones iniciales complejas, y un tiempo de muestreo $\tau = 0.001$. ante una entrada cuadrada, ecuación 5.17 con $i=2$:

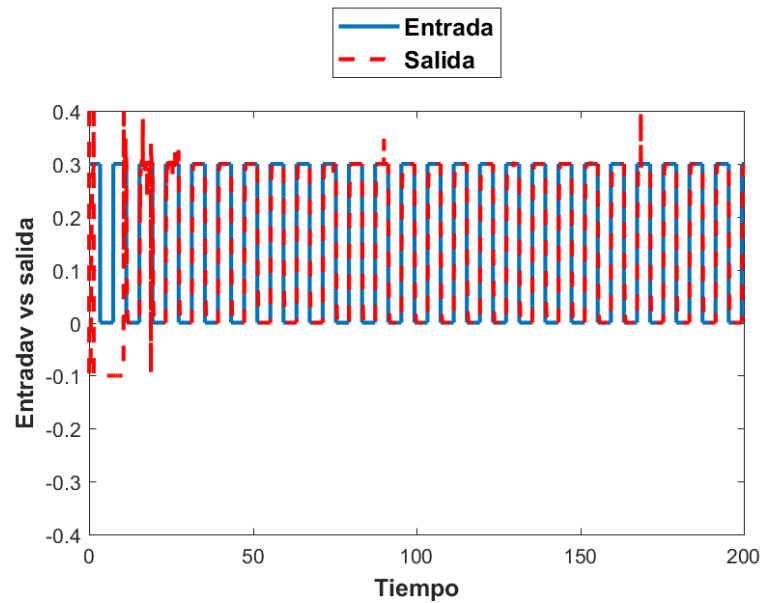


Figura 135. Comparación de la Entrada y la Salida (SMC).

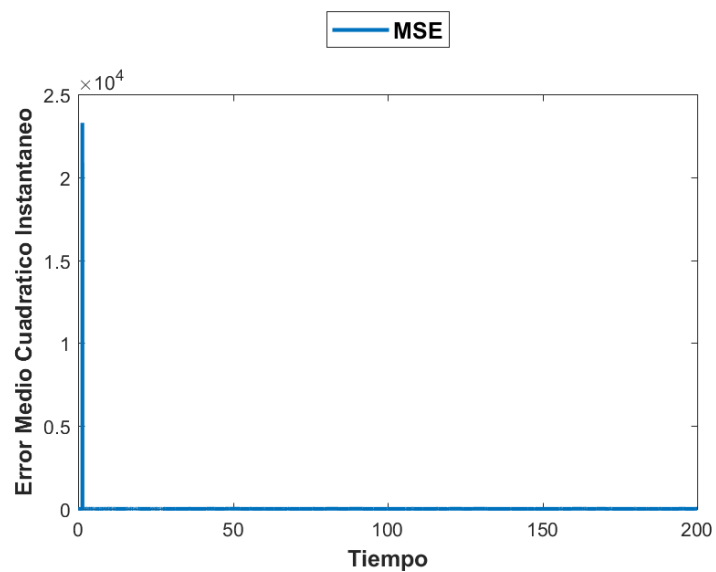


Figura 136. Error Medio Cuadrático Instantáneo

5.7.2 Sliding-Mode Control con i-term para una planta de dos Grado de Libertad

Se ejemplifica con un método de aprendizaje Back-Propagation esta estrategia de control obteniendo los siguientes resultados, bajo condiciones iniciales complejas, y un tiempo de muestreo $\tau = 0.001$. ante una entrada:

$$u_i(t) = 0.1 \operatorname{sen}\left(\frac{1}{10} \cdot t\right) + 0.1 \cdot \operatorname{sen}\left(\frac{1}{25} \cdot t\right) \quad (5.27)$$

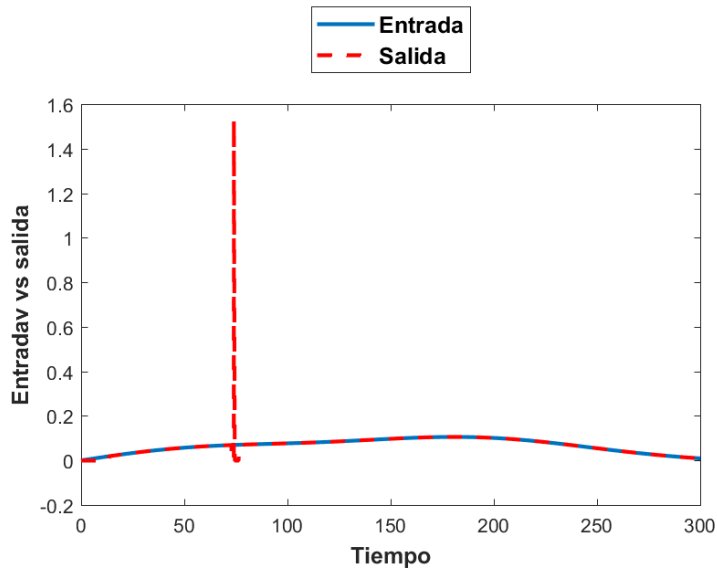


Figura 137. Comparación de la Entrada y la Salida (SMC).

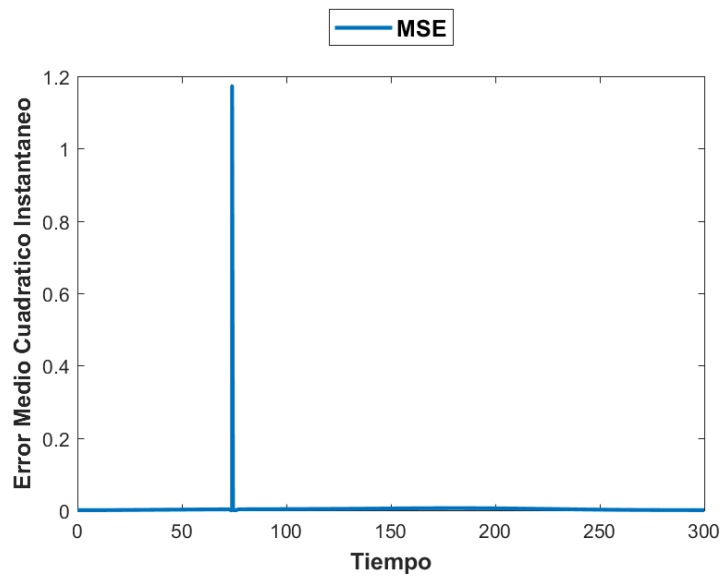


Figura 138. -Error Medio Cuadrático Instantáneo.

6. CONCLUSIONES

El uso del Multi-Modelo Difuso-Difuso Jerárquico como medio para la resolución de los problemas de Identificación y Control de sistemas no lineales, han sido satisfactorios, sin embargo, combinar esta herramienta con una Red Neuronal Recurrente con Valores Complejos, amplía el rango de acción de la misma, ya que los valores complejos dotan de un mejor medio para descripción de modelos no lineales oscilatorios. A través de la experimentación se comprobó que la adición de los números complejos, pueden llegar a simplificar el número de neuronas que se necesitan para llevar a cabo los aprendizajes, ya que provee a cada neurona con dos parámetros, uno real y uno imaginario, al llevar a cabo las simulaciones se encontró una diferencia con las Redes Neuronales Artificiales con Valores Reales es que la convergencia es más rápida debido a los números complejos, esto es así porque en los algoritmos de aprendizaje Back-Propagation y Levenberg-Marquardt, tienen en sí mismos productos de matrices complejos que dan como resultado un dos términos reales, y uno imaginario, la suma de los términos reales se le denomina adición de dinámica, por lo que es claro el porqué de la convergencia es más rápida, una vez que el índice de desempeño (MSE instantáneo) es bajo, el ajuste realizado por los algoritmos de aprendizajes es prácticamente nulo; el incremento de la dinámica es un arma de doble filo ya que ante una mala elección de las condiciones iniciales puede significar la existencia de un error muy grande lo cual implicaría una compensación demasiado grande y ante el producto de matrices complejas, esa adición de dinámica puede volver inestable el sistema. Que una Red Neuronal Artificial con Valores Complejos, converja depende también de la función de activación, la cual debe ser holomorfa, acotada y sin punto de singularidad.

Una forma de saber cuántas funciones membresía son necesarias para tener una identificación certera es probar con una sola Red Neuronal Artificial, y en donde gráficamente se observen desviaciones o dificultades para el aprendizaje, en esa zona se puede proponer un recubrimiento o una función de membresía entera para esa parte en especial, de tal manera que el Multi-Modelo refuerce el trabajo sobre esa región.

En cuanto a identificación de la planta de un grado de libertad, resulta que los resultados obtenidos por el algoritmo de aprendizaje Levenberg-Marquardt, son mejores que los obtenidos por el algoritmo de Back-Propagation. Esto es posible apreciarlo gráficamente.

El control adaptable por Modelo-Inverso, resulta eficaz, ya que al tener como control una Red Neuronal Recurrente con Valores Complejos, y tener como error aprendizaje a la diferencia de la entrada y la salida. Es beneficiosa al Multi-Modelo Neuronal-Difuso, ya que el número de Redes Neuronales es visiblemente reducido, minimizando el costo computacional, y por ende el tiempo de simulación. Del mismo modo que en identificación, el algoritmo Levenberg-Marquardt presenta mejores resultados que el algoritmo Back-Propagation.

Para el Control Adaptable por modelo Inverso con Termino Integral, maximiza el aprendizaje ya que el control en las primeras épocas de aprendizaje elimina el error de manera más rápida debido a la ganancia del Termino integral. Esto mejora aún más los buenos resultados presentados por el algoritmo Levenberg-Marquardt sobre el algoritmo Back-Propagation.

En la discusión de que algoritmo es mejor, entran características tales como resultados, facilidad de implementación, estabilidad, y manejo del algoritmo. Con base en lo anterior, aunque el algoritmo Levenberg-Marquardt presenta mejores resultados, el tiempo y dedicación invertido para su correcto funcionamiento, es mucho mayor que el utilizado para el Back-Propagation por lo que la accesibilidad de uso es mejor el segundo mencionado.

6.1 Trabajo a Futuro

Dentro de este apartado se propone trabajar sobre las pruebas de estabilidad de los algoritmos, ya que las condiciones de las funciones de activación son únicamente necesarias, más no suficientes.

También la implementación de la identificación a través del Multi-Modelo Neuronal-Difuso, como estimador para el uso de un control por Sliding-Mode.

REFERENCIAS

- [1] Aleksander, I., Morton, H., *An Introduction to Neural Computing*, Londres: Chapman & Hall, 1990.
- [2] S. Haykin, *Neural Networks and Learning Machines*, Ontario: Prentice Hall, 2009.
- [3] I. Aizenberg, *Complex-Valued Neural Networks with Multi-Valued Neurons*, Springer-Verlag, 2011.
- [4] K. S. Narendra and K. Parthasaeathy, «Identification and Control of Dynamical Systems Using Neural Networks,» *IEEE*, vol. 1, nº 1, pp. 4-27, 1990.
- [5] Takagi, T., Sugeno M., «Fuzzy Identification of Systems and it's Applications to Modeling and Control,» *IEEE*, Vols. %1 de %2SMC-15, nº 1, pp. 116-132, 1985.
- [6] P. A. Mastorocostas, J. B. Theocharis, «Recurrent Fuzzy-Neural Model for Dynamic System using Fuzzy Models,» *IEEE*, vol. 32, pp. 176-190, 2002.
- [7] P. Frasconi, M Gori, G. Soda , «Local Feeedback Miltilayerd Networks,» *Neural Computation* , vol. 4, pp. 120-130, 1992.
- [8] I. S. Baruoh, J. L. Olivares, «Implementación de un Multi- Modelo Neuronal Jerárquico Para Identificación y Control de Sistemas Mecánicos,» *Computación y Sistemas*, vol. 9, nº 1, pp. 28-40, 2005.
- [9] I. S. Barouh, R. B. Flores, «A Recurrent Multimodel for Systems Identification and Control,» *Proc. Of the European Control Conference, ECC'01*, pp. 3540-3545, 2001.
- [10] I. S. Barouh, R. B. Lopez, J. L. O. Guzman, J. M. Flores, «A fuzzy-neural multi-model for a nonlinear systems identification and control,» *International Conference on fuzy Systems, Neural Networks and Genetic* , vol. 159, pp. 2650-2667, 2008.
- [11] A. Hirose, *Complex-Valued Neural Networks: Theories and Applications*, Tokio: World Scientific Publishing , 2003.
- [12] A. Hirose, *Complex-Valued Neural Networks*, Tokio : Springer-Verlag, 2012.
- [13] H. Leung and S. Haykin, «The Complex Backpropagation Algoritm,» *IEEE Transactions on Signal Processing* , vol. 39, nº 9, pp. 2101-2104, 1991.
- [14] I. S. Baruoh and V. M. Arellano-Quintana , «Identification and Control of Oscillatory Dynamical Systems Using Recurrent Complex-Valued Neural Networks,» *Proceedings of the 18th International Conference on Circuitis, Systems, Communications and Computers*, pp. 331-336, 2015.
- [15] V. M. Arelano-Quintana and I. S. Baruoh, «Identification of Dynamical Systems using Recurrent Complex-Valued Neural Networks,» *Procedings of the 18th International Conference on Circuits, Systems, Communications and Computers* , pp. 74-79, 2014.

- [16] E. A. Wan and F. Beaufayes , «Diagramatic Methods for Deriving and Relating Temporal Neural Network Algorithms,» *Neural Computation* , pp. 182-201, 1996.
- [17] I. S. Baruah and E. P. Reynaud , «Control of Nonlinear Dynamical Systems Using Levenberg-Marquardt Learning Algorithm for Recurrent Complex-Valued Neural Networks,» *Proceedings of the 19th International Conference on Circuits, Systems, Communications and Computers* , pp. 398-403, 2015.
- [18] E. P. Reynaud and I. S. Baruah , «Identification of Dynamical,» *Proceedings of the 19th International Conference on Circuits, Systems, Communications and Computers* , pp. 331-336, 2015.
- [19] I. S. Baruch and C. R. Mariaca-Gaspar, «A Levenberg-Marquardt Learning Applied for Recurrent Neural Identification and Control of Wastwater Treatment Bioprocess,» *International Journal of Intelligent Systems* , nº 24, pp. 1094-1114, 2009.
- [20] P. I. Viñuela y I. M. G. León , *Redes de Neuronas Artificiales: Un Enfoque Práctico*, Madrid : Pearson Prentice-Hall, 2004.
- [21] W. McCulloch and W. Pitts, «A logical Calculus of the ideas immanent in nervous activity,» *Bullentin of Mathematical Biophysics*, vol. 7, pp. 115-133, 1943.
- [22] Kim, T., Adali, T., «Approximation by fully complex multilayer perceptrons,» *Neural*, vol. 15, p. 1641–1666, 2003.
- [23] T. Nitta:, «An extension of the back-propagation to complex numbers,» *Neural Networks*, vol. 10, nº 8, p. 1391–1415, 1997.
- [24] I. Aizenberg, C. Moraga, «Multilayer feed-forward neural network based on multivalued,» *Soft Computing*, vol. 11, nº 2, p. 169–183, 2007.
- [25] B.K. Tripathi, B. Chandra, P.K. Kalra,, «The generalized product neuron model in,» *In: Proc. of International Conference on Neural Information Processing*, 2008.
- [26] B.K. Tripathi, B. Chandra, P.K. Kalra, «The generalized product neuron model in,» *In: Proc. of International Conference on Neural Information Processing*, 2008.
- [27] P.K. Kalra, R.N. Yadav, J. John, «Time series prediction with single multiplicative,» *Applied Soft Computing*, vol. 7, p. 1157–1163, 2007.
- [28] Dilip Kumar Pratihar and Lakhmi C. Jain, *Intelligent Autonomous Systems: Foundations and Applications*, Chennai: Springer-Verlag, 2010.
- [29] N.N. Aizenberg, Y.L. Ivaskiv, D.A Pospelov, «About one generalization of the threshold function,» *The Reports of the Academy of Sciences of the USSR*, vol. 196, nº 6, p. 1287–1290, 1971.
- [30] F. Piazza, N. Benvenuto, «On the complex back-propagation algorithm,» *IEEE Transaction*, vol. 40, nº 4, p. 967–969, 1992.

- [31] S. Goh; «A complex-valued RTRL algorithm for recurrent neural networks,» *Neural Computation*, vol. 16, p. 2699–2713, 2003.
- [32] D. T. Pham and S. Yildirim, «Robot COntrol Using Jordan Neural Networks,» *Proc. of the Interat. Conf. On recent Advances in Mecatronics*, 1995.
- [33] N. K. Kasabov, *Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering*, London: The MIT Press, 1998.
- [34] C. G. Morcillo, «Lógica Difusa: Una Introducción Práctica,» *Técnicas de Softcomputing*.
- [35] L. Zadeh, «Fuzzy set,» *Information and Control*, vol. 8, pp. 338-353, 1965.
- [36] L. A. Zadeh, «The concept of a linguistic variable and its applications to approximate reasoning,» *Information Science*, pp. 199-249, 301-357, 1975.
- [37] T. Takagi and M. Sugeno, «erivation of fuzzy control rules from human operato's control actions,» *Proc. IFAC Symp. Fuzzy Inf. Knowl. Represent. Decis. Anal.*, pp. 55-60, 1983.
- [38] K. Matsuoka, «Learnig of Neural Networks Using Their Adjoint Systems,» *System and Computers*, vol. 22, nº 11, pp. 31-41, 1991.
- [39] V.R. Churchill and J.W. Brown, *Complex Variable and Applications*, U.S.A: McGraw-Hill, 2009.
- [40] L. V. Ahlfors, *Complex Analysis: An Introduction to the Theory of Analytic Functions of One Complex Variable*, U.S.A: McGraw-Hill, 1966.

Anexo

Background Matemático

Números Complejos

Recordando, [39], propiedades básicas de los números complejos, se tiene que:

$$C = \{(a,b) \text{ tal que } a, b \in \mathbb{R}\}$$

Definición. El sistema de los números complejos está formado por todas las parejas (a,b) con a y b números reales.

Por lo que el conjunto \mathbb{C} es posible expresarlo como \mathbb{R}^2 , además para este conjunto es posible definir las operaciones suma y producto de la siguiente manera:

$$SUMA: (a,b) + (c,d) = (a+c, b+d)$$

$$PRODUCTO: (a,b) \cdot (c,d) = (a \cdot c - b \cdot d, b \cdot c + a \cdot d)$$

Las operaciones antes definidas, el resultado de ellas también será un número complejo. Estos números tiene una representación binómica la cual queda representada por:

$$z = a + ib$$

Tal que, $a, b \in \mathbb{R}$, por lo que el número a es la parte real del número z y b es la parte imaginaria de z . Y cumplen con las siguientes propiedades:

SUMA

- *CONMUTATIVA* $z_1 + z_2 = z_2 + z_1$
- *ASOCIATIVA* $z_1 + (z_2 + z_3) = (z_1 + z_2) + z_3$
- *NEUTRO ADITIVO* $\exists 0 = 0 + i0 \text{ tal que } 0 + z = z + 0$
- *INVERSO ADITIVO* $\forall z = a + ib, \exists (-z) = -a - ib \text{ tal que } (-z) + z = z + (-z)$

PRODUCTO

- *CONMUTATIVA* $z_1 \cdot z_2 = z_2 \cdot z_1$
- *ASOCIATIVA* $z_1(z_2 \cdot z_3) = (z_1 \cdot z_2)z_3$
- *NEUTRO MULT.* $\exists 1 = 1 + i \cdot 0$ tal que $1 \cdot z = z \cdot 1 = z$
- *INVERSO MULT.* $\forall z \neq 0, \exists z^{-1} \mid z \cdot z^{-1} = z^{-1} \cdot z = 1$
- *DISTRIBUTIVIDAD* $z_1(z_2 + z_3) = z_1z_2 + z_1z_3$

Lo anterior es definido como propiedades de campo. Y por ellas forman un campo geoméricamente similar a \mathbb{R}^2 .

Definición. Sea $z = a + ib$ un número complejo, por lo que complejo conjugado se define como:

$$\bar{z} = a - ib$$

Definición. Sea $z = a + ib$ un número complejo, su *norma Euclidiana* está definida como:

$$|z| = \sqrt{a^2 + b^2}$$

En su conjunto, el complejo conjugado y la norma Euclidiana, cumplen con las siguientes propiedades:

- $\overline{(z_1 + z_2)} = \bar{z}_1 + \bar{z}_2$
- $\overline{(z_1 z_2)} = \bar{z}_1 \bar{z}_2$
- $z \bar{z} = |z|^2$
- $|z_1 z_2| = |z_1| |z_2|$
- $|z| \geq 0$ entonces $|z| = 0 \Leftrightarrow z = 0$
- $|z_1 + z_2| \leq |z_1| + |z_2|$

Dado que el defecto algebraico los \mathbb{R} , es que no puede ser algebraicamente cerrado, es decir, existen ecuaciones polinómicas con coeficientes reales que no tiene soluciones reales, por esto la principal ventaja de \mathbb{C} esto no ocurre, por lo que algebraicamente es cerrado.

Funciones Complejas, Funciones Holomorfas o Analíticas y Funciones Armónicas

Al hacer uso de funciones complejas la definición y propiedades de la derivación son muy similares a las aplicadas para funciones reales, sin embargo, después de un minucioso estudio se puede llegar a la conclusión que la derivabilidad compleja es una condición más fuerte que la derivabilidad real dadas las condiciones de Cauchy-Riemann, [40]. Además, en la tesis de E. P. Reynaud, [17], se hace un análisis más extenso de lo siguiente:

Definición. Se define como función compleja un mapeo $f : \Omega \subseteq \mathbb{C} \rightarrow \mathbb{C}$, tal que para todo número z al aplicar f , se obtiene otro número complejo.

$$f(z) = f(a, b) = u(a, b) + i \cdot v(a, b)$$

Donde u, v son funciones definidas en \mathbb{R}^2 .

Definición. Sea $\Omega \subseteq \mathbb{C}$, abierto. Sea $f : \Omega \rightarrow \mathbb{C}$ y $z_0 \in \Omega$, entonces f es derivable en z_0 , si existe:

$$f'(z_0) \in \mathbb{C} = \lim_{z \rightarrow z_0} \frac{f(z) - f(z_0)}{z - z_0}$$

De tal manera que el resultado de dicho limite, se le puede llamar derivada de f en z_0 .

Teorema. De una función compleja f , si la derivada es cero para todo un dominio Ω se dice que f es constante en cualquier punto del dominio Ω .

Definición. En el conjunto abierto $\Omega \subseteq \mathbb{C}$, si f es diferenciable, se dice que f es una función holomorfa o analítica.

Condiciones de Cauchy-Riemann

Se tiene que:

$$f : \Omega \rightarrow \mathbb{C}, f(z) = u(a, b) + i \cdot v(a, b)$$

Por lo que

$$z = a + ib \in \Omega \text{ y } z_0 = a_0 + ib_0 \in \Omega$$

Teorema. La función compleja f , cumple con la propiedad de derivabilidad en el punto z_0 , si y solo si:

- u, v son diferenciables en z_0
- Las llamadas condiciones de Cauchy-Riemann se cumplen:

$$\left. \frac{\partial u}{\partial a} \right|_{(a_0, b_0)} = \left. \frac{\partial v}{\partial b} \right|_{(a_0, b_0)}, \quad \left. \frac{\partial u}{\partial a} \right|_{(a_0, b_0)} = - \left. \frac{\partial v}{\partial b} \right|_{(a_0, b_0)}$$

Teorema. Si las derivadas parciales provenientes de las condiciones de Cauchy-Riemann, son continuas en el punto z_0 y para una vecindad δ , entonces $f'(z_0)$ existe.

Definición. Una función compleja f se dice holomorfa, si el teorema anterior se cumple.

Definición. Una función compleja f se dice entera si es holomorfa en \mathbb{C} .

Otra manera de probar que una función es holomorfa es a través del concepto de función compleja armónica.

Definición. Se dice que f armónica sobre un dominio $\Omega \subseteq \mathbb{R}^2$, si tiene derivadas parciales de primer y segundo orden, son continuas y definidas para cualquier punto de Ω . Y además cumplen con la siguiente ecuación.

$$\frac{\partial^2}{\partial a^2} u(a, b) + \frac{\partial^2}{\partial b^2} v(a, b) = 0$$

Teorema. Si $f(z) = f(a, b) = u(a, b) + i \cdot v(a, b)$ es una función compleja holomorfa, entonces las funciones reales u, v son armónicas.

Teorema. La función compleja $f(z) = f(a, b) = u(a, b) + i \cdot v(a, b)$ es holomorfa en Ω y solo si v es el conjugado armónico de u .

Es decir que se cumple

$$\frac{\partial^2}{\partial a^2} f(a,b) + \frac{\partial^2}{\partial b^2} f(a,b) = 0$$

Dado que dos funciones u, v son armónicas en Ω , y sus derivadas parciales de cumplen la anterior ecuación, se dice que v es el conjugado armónico de u .

Identification of nonlinear dynamical systems by means of complex-valued fuzzy-neural multi-model)

Articulo aceptado para presentarse en el 2017 IEEE Symposium Series on Computational Intelligence, Honolulu, Hawaii, November 25th to December 1, 2017.

Identification of nonlinear dynamical systems by means of complex-valued fuzzy-neural multi-model

Mario Cesar Maya Rodriguez, Ieroham S. Baruch

Department of Automatic Control,
CINVESTAV-IPN

07360 México City, MEXICO

e-mail: {mmaya, baruch}@ctrl.cinvestav.mx

Abstract—

In this paper, we propose to use the powerful tool of a Fuzzy-Neural Multi-Model, whose structure consists in a Hierarchical methodology composed by: Fuzzyfier, Fuzzy Rule-Based Inference Takagi-Sugeno rules and Defuzzyfier. It is combined with a Complex-Valued Recurrent Neural Network topology. The topology is trained with two recursive learning algorithms, the Back-Propagation and the Levenberg-Marquardt, both expressed in a complex domain. The main objective of system identification of nonlinear oscillatory plants is to issue states and parameters for control. The system identification used only two membership functions (positive and negative) with a small overlap in the neighborhood of zero. In this case, a sinus and square wave are used as input. The simulation results show the comparison between the output of the oscillatory plant (one degree of freedom flexible-joint robot model) and the output of the system identification algorithm. Moreover, the Levenberg-Marquardt algorithm presents some advantages over the Back-Propagation algorithm for the two input signals. The Levenberg-Marquardt minimized cost function (instantaneous Means Squared Error) it is faster than the Back-Propagation. Finally, the comparative simulation results confirm better quality of the Hierarchical Fuzzy-Neural Multi-Model expressed in a complex domain system identification over the simple system identification composed by only Complex-Valued Recurrent Neural Network.

Keywords— Complex Valued Fuzzy Neural Multi Model, Systems Identification, Nonlinear Dynamical Plant, Hybrid Intelligent Systems, Real Time System, Back-Propagation learning, Levenberg-Marquardt learning

I. INTRODUCTION

The systems identification is an essential part of the control theory because many times the mathematical model and parameters of the plant are not known. The complex domain has several properties that are applied in the Neural Networks: it gives the ability to extend the number of parameters, [1,2], which consequently, benefits the representation of the nonlinear oscillatory systems; as demonstrated in the circuit analysis, electrical systems with higher nonlinearity, power transformers, electromagnetic waves, even the quantum mechanics, [3, 4, 5], thus, this has been used in problems where it is necessary to collect information from the phase and also to retain it for a period of time; for example, pulse traveling through the nervous system, and there may be more than one

with different amplitude and phase. It is remarkable that the activation functions in a complex domain have a direct relationship with the architecture and the learning algorithm, [5], also the conditions to choose them are stronger; therefore, the activation function needs to be a holomorphic, bounded and without singularity points in its domain. For example, the $\tanh(\cdot)$, presents recurrent singularities. Moreover, in several works, [6, 7, 8], it has been proposed to construct a complex function and with these troubles are avoided in the activation function.

The book of N. Kasabov, [9], describes the fuzzy systems as a process of articulating knowledge by using linguistic terms, fuzzy rules and membership functions; these are used for problems such as facial recognition and image retrieval, [5, 10, 11], systems modelling, [12]; also, when is necessary to use the experience of an “expert” operator and the control law based on inaccurate concepts derived from his experience. The fuzzy systems depend on the selection of the membership functions and its parameters, this has an evident impact on the results, in, [13], auto-tuning of fuzzy supervisor is proposed to avoid the selections of parameters. Some other applications of Neural and Fuzzy-Neural Networks have been applied for chaotic oscillatory systems, [14, 15, 16].

Thus, the approach of multi-model, take on consideration, that the model of the plant depends on the region of operation, it is possible to divide the operating space in regions and identify the signal of the plant as a function of these regions, so that it is possible to obtain a state space model with variable parameters.

In [17, 18, 19], it is proposed a Hierarchical Fuzzy-Neural Multi-Model (HFNMM) for state estimation and adaptive control of nonlinear plants, expressed in a real domain, working with three membership functions (Positive, Zero, and Negative), with overlapping. The present work is an extension of the HFNMM expressed in a complex domain, obtained throughout the integration of the Complex- Valued Recurrent Neural Network (CVRNN). The learning of this neural network is made by the Back-Propagation (BP) and the Levenberg-

Marquardt (LM) algorithms. In [8, 20], the diagrammatic rules were used to construct an adjoint network to propagate the complex output error and so to obtain the weight adjustment. The HFNMM used two membership functions (Positive and Negative) with a small overlapping in the neighborhood of zero and the defuzzification is made by a CVRNN. Furthermore, the merit of the topology is a prediction/estimation of the states of the system, directly by the state-space model. This model has the advantage to be completely parallel, so its dynamics depends only on the previous step and not on the other past steps.

The main objective of the paper is the study of an idealized plant of one degree of freedom flexible joint in case of sin and square waves by means HFNMM in a complex domain and his comparison with CVRNN presented in [21], using the learning algorithms BP and LM.

II. TOPOLOGY AND LEARNING OF CVRNN

The general topology in consideration is presented in [21], to the complex-valued (CV) case and it is an extension of [22]. The CVRNN has complex states vectors and complex A, B, C weight matrices, as it is defined as follows:

$$X(k+1) = Ax(k) + BU(k) \quad (1)$$

$$Z(k) = \Phi[X_{re}(k)] + i \Phi[X_{im}(k)] \quad (2)$$

$$V(k) = CZ(k) \quad (3)$$

$$Y(k) = \Gamma[X_{re}(k)] + i \Gamma[X_{im}(k)] \quad (4)$$

In this case, the complex activation function $\tanh(\cdot)$ is used, [3, 7, 21], because it does not have any singularity, and it is represented by the next equation:

$$f(z) = \tanh[\text{Re}(k)] + i \tanh[\text{Im}(k)] \quad (5)$$

For example, let us consider (3). Here the result V is a product of two matrices C and Z which gives:

$$V(k) = \beta(k) + i(\delta(k)) + i^2(\gamma(k)) \quad (6)$$

Where β : is real, $i(\delta)$ is imaginary and γ is real by $i^2 = -1$.

Let us rewrite (6) in the form:

$$V(k) = V_{re}(k) + iV_{im}(k) \quad (7)$$

The i^2 term add some dynamics to the output of CVRNN, with this, the identification improved the estimation and obtained better performance. Thus, the error is expressed by:

$$E(k) = \text{Real}(Yd(k) - Y(k)) + i \text{Imag}(Yd(k) - Y(k)) \quad (8)$$

Otherwise:

$$E(k) = E_{re}(k) + iE_{im}(k) \quad (9)$$

The vectors and matrices dimensions of the CVRNN topology are given as follows: A $\in \mathbb{C}^{(n \times n)}$ is the feedback weight matrix, B $\in \mathbb{C}^{(n \times m)}$ is the input weight matrix, C $\in \mathbb{C}^{(p \times n)}$ is the output weight matrix, X $\in \mathbb{C}^{(n \times 1)}$ is the internal state vector, Z $\in \mathbb{C}^n$ is the hidden state vector, U $\in \mathbb{R}^m$ is the network input, Y $\in \mathbb{C}^p$ the network output and the $\Phi[\cdot]$, $\Gamma[\cdot]$ are complex valued vector activation functions. The number of inputs is (m), the number of neurons in the hidden layer is (n) and the number of outputs is (p). The following inequality is a stability preserving condition, imposed also on the diagonal blocks of the matrix A.

$$A = \text{diag}(A_i), \quad |A_i| < 1, \quad i = 1, \dots, N \quad (10)$$

The performance index to be minimized is given by:

$$\zeta(k) = (1/2) \sum_j [E_j(k) \cdot E_j^*(k)] \quad (11)$$

This function $\zeta(k)$, is a mapping of the error $f: \mathbb{C} \rightarrow \mathbb{R}$, [23].

A. Learning of CVRNN with Back-Propagation algorithm

The general CVRNN real time BP learning algorithm with momentum term is given by:

$$W(k+1) = W(k) + \eta \Delta W(k) + \alpha \Delta W(k-1) \quad (12)$$

Where: W is a general weight matrix (A, B, C), ΔW is the change in W, η is a constant of learning, and α is a constant of a momentum term.

The following inequality is a variation of the proposed in [17], in the present work it is used that the forward weight is restricted with respect of the previous weight:

$$|W(k+1)| < W(k) \quad (13)$$

In [20], the complex-valued diagrammatic rules were applied which permit to obtain the adjoint CVRNN, used to write the next algorithm for weights update:

$$E_1(k) = \Phi'[\text{Real}(Y(k))]E_{re}(k) + i \Phi'[\text{Imag}(Y(k))]E_{im}(k) \quad (14)$$

$$\Delta C(k) = E_1(k) \cdot Z(k) \quad (15)$$

$$E_2(k) = C^*(k) \cdot E_1(k) \quad (16)$$

$$E_3(k) = \Gamma'[\text{Real}(Z(k))]E_{2re}(k) + i \Gamma'[\text{Imag}(Z(k))]E_{2im}(k) \quad (17)$$

$$\Delta A(k) = E_3(k) \cdot X(k) \quad (18)$$

$$\Delta B(k) = E_3(k) \cdot U(k) \quad (19)$$

The conjugate transpose is denoted by *.

The learning parameters of the algorithm should be restricted as follows:

$$0 < \eta \quad \& \quad \alpha < 1 \quad (20)$$

B. Learning of CVRNN with Levenberg-Marquardt algorithm

The general CVRNN real time LM learning algorithm, [24], is given by:

$$W(k+1) = W(k) + P(k) \cdot DY[W(k)] \cdot E(k) \quad (21)$$

With the same restriction that the BP, (14).

From the adjoint network, the gradient terms are derived, and needed to the update weights, which are described by the following equations:

$$D_1(k) = \Phi'[\text{Real}(Y(k))] + i \Phi'[\text{Imag}(Y(k))] I \quad (22)$$

$$D_2(k) = (\Gamma'[\text{Re}(Z'(k))] \text{Re}(C') + i \Gamma'[\text{Im}(Z'(k))] \text{Im}(C')) D_1(k) \quad (23)$$

$$DY[C(k)] = D_1(k) \cdot Z(k) \quad (24)$$

$$DY[A(k)] = D_2(k) \cdot X(k) \quad (25)$$

$$DY[B(k)] = D_2(k) \cdot U(k) \quad (26)$$

The matrix P is computed recursively using following Riccati equation:

$$P(k) = \alpha^{-1}(k) [P(k-1) - P(k-1) \cdot \Omega_{W(k)} \cdot S^{-1}_{W(k)} \cdot \Omega^*_{W(k)} \cdot P(k-1)] \quad (27)$$

Where the matrices Ω_W and S_W are given by:

$$\Omega^*_{W(k)} = \begin{bmatrix} DY^*[W(k)] \\ 0 \dots 1 \dots 0 \end{bmatrix} \quad (28)$$

$$S_{W(k)} = \alpha(k) \Lambda + \Omega^*_{W(k)} \cdot P(k-1) \cdot \Omega_{W(k)} \quad (29)$$

$$\Lambda^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & \rho \end{bmatrix} \quad (30)$$

Again, the matrices P and S_W have dimensions $(N_w \times N_w)$ and (2×2) respectively, where N_w is the number of weights in the vector W. The matrix Ω_W has dimension $(N_w \times 2)$, the second row of Ω_W^* consists of $(N_w \times 1)$ zeros and a unit in the i -th position computed by:

$$i = k \cdot \text{mod}(N_w) + 1 \quad (31)$$

The parameters for the algorithm should be restricted as follows:

$$10^{-6} \leq \rho \leq 10^{-6}, 0.97 \leq \alpha(k) \leq 1, 10^3 \leq P(0) \leq 10^6 \quad (32)$$

In the present work, the parameter $\alpha(k)$ [22], is found for each instant k, based on obtaining the least possible error in the α range.

C. Convergence and Stability

In, [19, 22], it is discussed the proofs of stability and convergence in the topology, the HFNNM and the algorithms of learning, in the real domain, we are currently working on the stability and convergence proofs for the complex case.

We consider that the way to make the proofs of stability and convergence for the HFNNM in a complex domain, is to follow those presented for the real case; however, taking into consideration the section I, the Neural Networks in a complex domain, the most important thing into the topology is the activation function which needs to be a holomorphic, bounded and without singularity points in its domain.

Taken on consideration the following definitions and theorems, [25]:

- *Definition:* Let f a function of a domain D, and let $M > 0$, with $M \in \mathbb{R}$. It is said that f is bounded if $|f(x)| \leq M$ for any point $x \in D$.
- *Theorem of Liouville:* For all complex holomorphic functions f, if f is bounded in all \mathbb{C} , and f is constant function in all \mathbb{C} .
- *"Small" Theorem of Picard:* If a complex function f is entire and not constant, then the set of values that can take $f(z)$ is the complex plane \mathbb{C} minus one point.
- *Definition:* If a function f is not holomorphic in at a point z_0 , but if it is for any point in the neighborhood of z_0 , then z_0 is a point of singularity.

Therefore, a function cannot be holomorphic and bounded at the same time and should not have points of singularity. If the activation function does not fulfill with this probably the system would never converge.

The activation function, (5), is an extension of real activation function and moreover, constructed, results a holomorphic, bounded and without singularity points in its domain. For its simplicity and ability to handle the parameters of amplitude and phase of complex signal, this activation function used in complex domain neural network, [5].

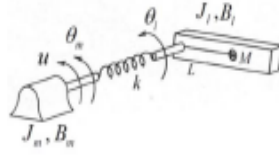


Fig. 3. Idealized model representing robot joint flexibility

Here, the motor torque as the input u of a plant. The equations of motion of the flexible-joint robot are given as follows:

$$\begin{aligned} J_l \ddot{\theta}_l + B_l \dot{\theta}_l + Mgl \sin \theta_l + k(\theta_l + \theta_m) &= 0 \\ J_m \ddot{\theta}_m + B_m \dot{\theta}_m - k(\theta_l + \theta_m) &= u \end{aligned} \quad (38)$$

Where: J_l, J_m are load and motor inertias; B_l, B_m are load and motor damping constants; u is the input torque applied to the motor shaft; M and L are the mass of the link and the length between the shaft and the center of mass of the link; k represents the torsional stiffness constant of the harmonic drive gear.

The system is sub-actuated, because the plant is described by two second order differential equations, representing a system with two degrees of freedom but only one input.

V. SIMULATION RESULTS OF IDENTIFICATION

The plant under identification is represented by a HFNNM and its structure is given in the Fig.2. In this case the LLI has four neurons in the hidden layer, the CVRNN-1 and CVRNN-2 have the following dimensions: $n=4; m=1; p=1$. In the ULI, the CVRNN-3 has the following dimensions $n=4; m=2; p=1$, with a sampling time of $T_0=0.01$. The numerical values of the plant parameters used in the plant model are $J_l = 1, J_m = 0.1, B_l = 1, B_m = 1, M = 0.5, g = 9.81, l = 0.5, k = 0.1$.

The input plant signals used for system identification are $u_s(t)$ and $u_{sq}(t)$. Naturally, both signals are discretized and given by following equations.

$$u_s(k) = \sin(0.1 \cdot k) + 0.5 \cdot \sin(0.04 \cdot k) \quad (39)$$

$$u_{sq}(k) = 0.3 \cdot \text{sign}(0.5 \cdot \sin(0.1 \cdot k) + 0.8 \cdot \sin(0.033 \cdot k)) \quad (40)$$

A. Simulation Results obtained with the BP algorithm

This part illustrates the BP algorithm of identification described in part 2.1. In this case, the learning parameters are chosen as $\eta_{1,2,3} = 0.05, 0.1, 0.25$ & $\alpha_{1,2,3} = 0.005, 0.01, 0.015$. The initial values of the matrices A, B, C and the state vector X are complex with values chosen in a small rank. The graphical results of the plant identification are given in Fig. 4, and Fig. 5. In both cases the output of the plant is compared with the output of the HFNNM.

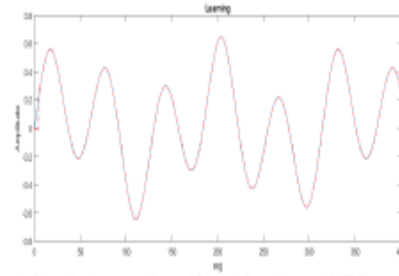


Fig. 4. Simulation results of Identification with BP and sinus wave. The output of the plant (continuous line) is compared with the output of the HFNNM (segmented line).

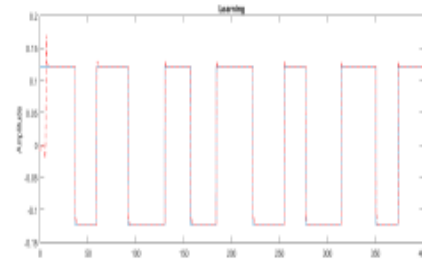


Fig.5. Simulation results of Identification with BP and square wave. The output of the plant (continuous line) is compared with the output of the HFNNM (segmented line).

B. Simulation Results obtained with the LM algorithm

This part illustrates the LM algorithm of identification described in part 2.2. For this case, we use an initial vector as $0.97 \leq \alpha(k) \leq 1$. The initial value of the parameter $P(0)$ and for the matrices $A, B, C, P_{A,B,C}(0) = 10^6$. The value of the small parameter is $\rho = 0.0001$. The initial conditions of the matrices A, B, C and state vector X are complex, with small values. The graphical results of the plant identification, with a sin wave of learning is given in Fig. 6 and the Fig. 7, shows the results for the squared wave. Here the output of the plant is compared with the output of the HFNNM.

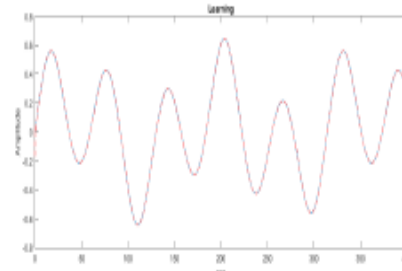


Fig. 6. Simulation results of Identification with LM and sin wave. The output of the plant (continuous line) is compared with the output of the HFNNM (segmented line).

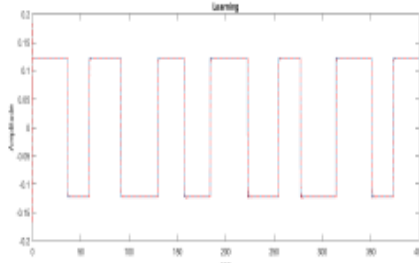


Fig. 7. Simulation results of Identification with LM and square wave. The output of the plant (continuous line) is compared with the output of the HFNNM (segmented line).

C. Comparative Final MSE results. Simulation Results of Nonlinear Oscillatory Plant Identification, with both algorithms.

The MSE instantaneous values obtained during identification experiments with both algorithms (see Table I), demonstrates that LM has better performance over BP.

Table I
Instantaneous MSE(k) of learning during system identification

Learning of Sinus Wave		
N. of Iteration	Back-Propagation	Levenberg-Marquardt
5000	2.2076e-07	1.1607e-07
16000	4.6833e-08	2.2546e-10
24000	4.2415e-08	9.9279e-09
29000	3.4651e-07	1.0235e-07
37000	2.7706e-05	1.1768e-05
40000	1.6068e-07	1.1932e-07
Learning of Square wave		
N. of Iteration	Back-Propagation	Levenberg-Marquardt
5000	1.8501e-16	7.6105e-29
16000	6.3892e-09	1.1830e-07
24000	1.8717e-19	4.6241e-31
29000	1.7376e-15	3.3662e-23
37000	1.3434e-20	5.0941e-32
40000	1.6227e-22	4.2468e-32

D. Final MSE results of the comparison between the HFNNM and LM with a single CVRNN. Simulation Results of Nonlinear Oscillatory Plant Identification with LM.

The MSE instantaneous values obtained during identification experiments with both approaches (see Table II),

Table II
Instantaneous MSE(k) of learning during system identification

Learning of Sinus Wave		
N. of Iteration	LM with HFNNM	LM with CVRNN
5000	1.1607e-07	2.5046e-07
16000	2.2546e-10	4.9873e-08
24000	9.9279e-09	4.6445e-08
29000	1.0235e-07	3.6621e-07
37000	1.1768e-05	2.9736e-05
40000	1.1932e-07	1.8368e-07
Learning of Squared Wave		
N. of Iteration	LM with HFNNM	LM with CVRNN
5000	7.6105e-29	2.4209e-20
16000	1.1830e-07	6.8342e-05
24000	4.6241e-31	3.6464e-25
29000	3.3662e-23	1.5727e-18
37000	5.0941e-32	4.3595e-26
40000	4.2468e-32	3.8546e-28

VI. CONCLUSIONS

In this paper, we present a Hierarchical Fuzzy-Neural Multi-Model, which uses two membership functions (positive and negative) and a small neighborhood in zero. The consequent part of fuzzy rules of the Takagi-Sugeno model we use a Complex Valued Recurrent Neural Network. This Hybrid Topology is learned by two different learning algorithms: The Back-Propagation and the Levenberg-Marquardt learning algorithms.

The first one (the BP) reduces the error gradually, while the second one (the LM) reduces the error faster than de BP. Furthermore, in the simulation results the BP presents greater flexibility in the selections of the initial conditions of the matrix weights; the graphical results show better performance of identification with the LM with respect to the BP.

The learning is validated by the results of MSE given in Table I, where the comparison of the MSE numeric results of the HFNNM-BP and HFNNM-LM, show better results of minimization of the instantaneous MSE (k) by means of LM learning algorithm for the both inputs.

The proposed HFNNM were used for identification of an idealized plant of one DOF flexible joint and exhibited superior results of identification under two types of inputs (sin wave and square wave) with respect of the simple CVRNN and it is validated in Table II.

REFERENCES

- Hirose, A.: *complex-Valued Neural Networks*, Tokio, Springer-Verlag, 2012.
- S. Haykin, *Neural Networks and Learning Machines*, Ontario, Prentice Hall, 2009.
- Minin, A., Chistyakov Y., Kholodova E., Zimmermann H, Knoll A.: Complex valued open recurrent neural network for power transformer modelling, *International Journal of Applied Mathematics and Informatics*, 6 (1), 41-48 2012.
- Ferariu, L.: Nonlinear system identification based on evolutionary dynamic neural network, *Proc. Of European Control conference*, Cambridge, UK, 432 (2003).
- Dilip Kumar Pratihari and Lakhmi C. Jain, *Intelligent Autonomous Systems: Foundations and Applications*, Chennai: Springer-Verlag, 2010.
- Woo, C., Hong, D.S: Adaptive equalization using the complex back-propagation algorithm. In: *Proceedings of IEEE international Conference on Neural Networks*, vol.4, pp. 2136-2441, Washington, DC (1996).
- Miklos, N., Salik, B.: Neural networks with complex activations and connection weights. *Complex Syst.* 8, 115-126 (1994).
- Nava, F., Baruch, I. Poznyak, A., Nenkova, B: Stability proofs of advanced recurrent neural networks topology and learning. *C. R. (Proc. Bul. Acad. Sci.)*57(1), 27-32 (2004).
- Kasabov, N.: *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*, Massachusetts, The MIT Press, 1996.
- Moallem, P., Mousavi, B. S. and S. Sh. Naghibzadeh: Fuzzy Inference System Optimized by Genetic Algorithm for Robust Face and Pose Detection, *International Journal of Artificial Intelligence*, 13(2), 73-88, 2015.
- Nowaková J., Prilepok M. and Snásel V.: Medical Image Retrieval Using Vector Quantization and Fuzzy S-tree, *Journal Medical Systems*, 41 (18), 1-6, 2017.
- Pozna C., Minculete N., Precup R. Kóczy L. and Ballagi Á.: Signatures: Definitions, operators and applications to fuzzy modelling, *Fuzzy Sets and Systems*, 201, 86-104, 2012.
- Davarynejad M., Akbarzadeh R.: Auto-Tuning Fuzzy Granulation for Evolutionary Optimization, *Proceedings of 2008 IEEE Congress on Evolutionary Computation*, Hong Kong, China, 3572-3579, 2008.
- Lin, D., Wang, X., Nian, F. and Zhang, Y.: Dynamic Fuzzy Neural Networks Modeling and Adaptive Back-Stepping tracking control of uncertain chaotic systems. *Neurocomputing*, 73 (11-24), 2873-2881-(2010).
- Zhagn, H., Wang, X., Lin, X., Liu, C.: Stability and synchronization for discrete-time complex-Valued Neural Networks with time varying delays. *PLOS ONE* 9(4), 6 (2014).
- Adamy, J, Kempf, R.: Regularity and chaos in recurrent fuzzy systems, *Fuzzy Sets Systems*, 140(2), 259-284, (2003).
- Baruch, I., Beltran, R., Olivares, J., Martín, J.: A fuzzy-neural multi model for nonlinear systems identification and control. *Fuzzy Sets and Systems*, 159, 2650-2667, 2008.
- Baruch, I. Implementation of a Neural Hierarchal Multi-Model for Identification and Control of Mechanical systems, *Computación y Sistemas (CIC-IPN)*, 9, ISSN:1405-5546, (2005).
- Baruch I, Lopez, R., Guzman, J. and Flores J., «A fuzzy-neural multi-model for a nonlinear systems identification and control.» *Fuzzy Sets and Systems*, vol. 159, pp. 2650-2667, 2008.
- Wan, E., Beaufays, F: Diagrammatic method for deriving and relating temporal neural networks algorithms. *Neural Computation*, 8,182-201 (1196).
- Baruch, I., Quintana V., Reynaud E. P. (2015) Dynamic Systems Identification and control by means of complex-Valued Recurrent Neural Networks. In: Sidorov G., Galicia-Haro S (eds) *Advances in Artificial Intelligence and Soft Computing. Lectures Notes in Computer Science*, vol9413. Springer, Cham.
- Baruch, I. Mariaca-Gaspar, C.: A Levenberg-Marquardt learning applied for recurrent neural identification and control of a wastewater treatment bioprocess. *Int. J. Intell. Syst* ISSN: 0884-8173,24,1094-1114 (2009).
- Baruch, I. Gortcheva, E. fuzzy neural model for nonlinear systems identification, In: *Proc. AARTC IFAC Workshop*, Cancun México, 283-288, (1998).
- Baruch, I., Quintana V., Reynaud E. P.: Identification and control of nonlinear dynamical systems using Levenberg-Marquardt learning algorithm for recurrent complex-valued neural networks, *Int. J of Neural networks and Advanced Applications*, ISSN:2313-0563, 3, (2016).
- Churchill, R. V. and Brown, J. W.: *Complex Variable and Applications*. 8th edition, U.S.A., Ed.: McGraw-Hill, 2009.
- Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control, *IEEE Transactions Systems Man Cybernet*, 15, 133-142, 1985.