

INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERÍA
MECÁNICA Y ELÉCTRICA

SECCIÓN DE ESTUDIOS DE POSGRADO E
INVESTIGACIÓN

**REGULADOR NEURODIFUSO ARTMAP
DE VOLTAJE DE UN GENERADOR
SÍNCRONO**

T E S I S

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN CIENCIAS CON ESPECIALIDAD
EN INGENIERÍA ELÉCTRICA

PRESENTA

MARTÍN EMMANUEL CÁRDENAS RIVAS



MÉXICO D.F.

JUNIO 2011



INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de MEXICO siendo las 11:00 horas del día 15 del mes de JUNIO del 2011 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de la: E. S. I. M. E. ZAC. para examinar la tesis titulada:

“REGULADOR NEURODIFUSO ARTMAP DE VOLTAJE DE UN GENERADOR SÍNCRONO”

Presentada por el alumno:

CÁRDENAS

Apellido paterno

RIVAS

Apellido materno

MARTÍN EMMANUEL

Nombre(s)

Con registro:

| | | | | | | |
|---|---|---|---|---|---|---|
| A | 0 | 9 | 0 | 4 | 8 | 2 |
|---|---|---|---|---|---|---|

aspirante de:

MAESTRÍA EN CIENCIAS EN INGENIERÍA ELÉCTRICA

Después de intercambiar opiniones, los miembros de la Comisión manifestaron **APROBAR LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

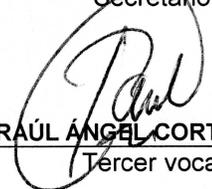
Director(a) de tesis


DR. DAVID ROMERO ROMERO


DR. TADEUSZ NIEWIEROWICZ SWIECICKA
Presidente


DR. DANIEL OLGÚN SALINAS
Secretario


DR. LESZEK KAWECKI ZLOTKOWSKA
Segundo vocal


DR. RAÚL ÁNGEL CORTÉS MATEOS
Tercer vocal

PRESIDENTE DEL COLEGIO DE PROFESORES


DR. JAIME ROBLES GARCIA





INSTITUTO POLITÉCNICO NACIONAL
COORDINACIÓN GENERAL DE POSGRADO
E INVESTIGACIÓN

CARTA CESIÓN DE DERECHOS

En la Ciudad de México, el día **21** del mes de **Junio** del año **2011**, el que suscribe **Martín Emmanuel Cárdenas Rivas**, alumno del Programa de **Maestría en Ciencias en Ingeniería Eléctrica** con número de registro **A090482** adscrito a la Sección de Estudios de Posgrado e Investigación de la ESIME Unidad Zacatenco, manifiesta que, bajo la dirección del **Dr. David Romero Romero**, es autor intelectual del presente trabajo de tesis titulado **“Regulador Neurodifuso ARTMAP de Voltaje de un Generador Síncrono”** y cede los derechos del mismo al Instituto Politécnico Nacional para su difusión con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Éste puede ser obtenido escribiendo a la siguiente dirección de correo electrónico: **mcr2180@yahoo.com.mx**. Si el permiso se concede, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Martín Emmanuel Cárdenas Rivas

Nombre y firma

AGRADECIMIENTOS

Deseo expresar mi más profundo y sincero agradecimiento a:

Dios y a la Virgen de Guadalupe por permitirme alcanzar una meta más en la vida.

Mis amadas esposa e hija, Raquel García González y Ximena Raquel Cárdenas García, que me han provisto de ánimo y fortaleza en los momentos que más lo he necesitado; ellas son el motor de mi vida.

Mis queridos padres, Inés Rivas Jiménez y Abdón Cárdenas Rojas, por su siempre incondicional apoyo y por enseñarme el valor de la responsabilidad, la honestidad y el trabajo duro; elementos indispensables en mi formación personal y profesional.

Mis queridas hermanas, Juanita Eloísa Cárdenas Rivas y María Guadalupe Cárdenas Rivas, por el ánimo y entusiasmo que siempre me han transmitido.

El Doctor David Romero Romero, que con sus valiosos consejos, apoyo y conocimientos transmitidos hacia mí fue posible la realización de este trabajo de tesis.

Los profesores del Departamento de Ingeniería Eléctrica de la Sección de Estudios de Posgrado e Investigación de la ESIME Unidad Zacatenco por brindarme parte de su tiempo y conocimientos.

El Consejo Nacional de Ciencia y Tecnología y el Instituto Politécnico Nacional por el apoyo económico otorgado durante la realización de este trabajo.

Mis compañeros y amigos durante mi estancia en esta institución, que con su amistad y consejos me confirmaron una vez más que el trabajo en equipo es la mejor manera de alcanzar un objetivo.

RESUMEN

En este trabajo se diseña y simula un controlador neurodifuso para regular el voltaje en terminales de un generador síncrono que se encuentra conectado a un bus infinito. Para ello se emplea la arquitectura ARTMAP difusa, que es uno de los modelos neurodifusos creados a partir de la Teoría de Resonancia Adaptable para generar mapeos multidimensionales entre patrones de entrada analógicos por medio de un aprendizaje supervisado estable e incremental.

En primera instancia, se aplica la arquitectura ARTMAP difusa directamente como regulador de voltaje del generador bajo prueba; esquema en el que se utiliza la técnica de modelo de referencia y sustitución de un controlador existente. Los resultados que se obtienen con esta técnica demuestran la capacidad de la arquitectura ARTMAP difusa para modelar y mejorar la respuesta de un sistema en particular en distintos puntos de operación.

Posteriormente, se usa la arquitectura ARTMAP difusa para sintonizar un controlador difuso proporcional integral (PI) tipo Mamdani; técnica con la que se consiguen resultados favorables bajo diferentes condiciones de operación.

Con la finalidad de verificar el funcionamiento del controlador neurodifuso de una forma más amplia, el sistema máquina bus infinito es sometido a diversas pruebas en las que se simulan diferentes eventos como fallas de corto circuito trifásicas, cambios de carga y cambios de referencia. Los resultados obtenidos se comparan con los respectivos que ofrece el controlador de referencia, el cual consta de un sistema de excitación tipo ST1. De esta comparación se observa una regulación de voltaje más eficiente y confiable por parte del controlador neurodifuso que del regulador ST1.

ABSTRACT

In this work, a neuro-fuzzy controller based on the fuzzy ARTMAP architecture is designed and simulated for regulating the terminal voltage of a synchronous generator connected to an infinite bus. Such architecture is one of the neuro-fuzzy models built from the Adaptive Resonance Theory for generating multidimensional maps between analog input patterns by an incremental and supervised learning.

To begin with, the fuzzy ARTMAP architecture is directly applied as voltage regulator for the under testing generator; this task is accomplished by means of modeling an existing controller and replacing it later. The results obtained with this technique demonstrate the ability of fuzzy ARTMAP architecture for modeling and enhancing the response of a particular system in different operation points.

Later, the fuzzy ARTMAP architecture is used for tuning a Mamdani proportional integral (PI) fuzzy controller. The results obtained with this technique are satisfactory for different operation conditions.

In order to verify the neuro-fuzzy controller performance in a wider manner, the infinite bus machine system is subject to several tests within different events simulations such as three-phase short circuit faults, load changes and reference changes. The obtained results are compared with those offered by the reference controller, which is a type ST1 excitation system. From this comparison it can be observed that the neuro-fuzzy controller provide a more efficient and reliable voltage regulation than the ST1 regulator.

ÍNDICE

| | |
|--|-------------|
| RESUMEN | I |
| ABSTRACT | III |
| ÍNDICE | V |
| ÍNDICE DE FIGURAS | IX |
| ÍNDICE DE TABLAS | XVII |
| ABREVIATURAS | XIX |
| CAPÍTULO 1. GENERALIDADES | 1 |
| 1.1 Introducción | 1 |
| 1.2 Objetivo de la tesis | 2 |
| 1.3 Justificación | 2 |
| 1.4 Estado del arte | 3 |
| 1.5 Contenido de la tesis | 4 |
| 1.6 Aportaciones | 5 |
| 1.7 Publicaciones | 6 |
| CAPÍTULO 2. ARQUITECTURA ARTMAP DIFUSA | 7 |
| 2.1 Introducción | 7 |
| 2.2 Teoría de resonancia adaptable | 8 |
| 2.2.1 Estructura y funcionamiento generales de la red ART1 | 8 |
| 2.2.2 Algoritmo de la red ART1 | 10 |
| 2.2.3 Difusificación de la red ART1 | 13 |
| 2.3 Arquitectura ARTMAP difusa | 14 |
| 2.3.1 Arquitectura ARTMAP | 14 |
| 2.3.2 Difusificación de la arquitectura ARTMAP | 15 |
| 2.3.3 Algoritmo de la arquitectura ARTMAP difusa | 16 |
| 2.3.4 Normalización de los vectores de entrada | 18 |
| 2.3.5 Aprendizaje rápido y recodificación lenta | 19 |
| | V |

| | |
|---|-----------|
| 2.3.6 Elección de subconjunto difuso en el límite conservativo | 20 |
| 2.3.7 Interpretación geométrica de las redes ART difusas | 20 |
| 2.3.8 Teorema de estabilidad en el aprendizaje de categorías | 23 |
| CAPÍTULO 3. APLICACIÓN DE LA ARQUITECTURA ARTMAP DIFUSA EN LA REGULACIÓN DE VOLTAJE DE UN GENERADOR SÍNCRONO | 25 |
| 3.1 Introducción | 25 |
| 3.2 Definición de los factores necesarios para el entrenamiento de la arquitectura ARTMAP difusa | 26 |
| 3.2.1 Esquema de entrenamiento | 26 |
| 3.2.2 Acondicionamiento y normalización de los vectores de entrada | 27 |
| 3.2.3 Selección de las variables de entrada | 29 |
| 3.2.4 Ajuste de parámetros | 30 |
| 3.3 Entrenamiento de la arquitectura ARTMAP difusa | 31 |
| 3.3.1 Falla de corto circuito trifásica para el entrenamiento | 32 |
| 3.3.2 Condiciones iniciales para el entrenamiento | 32 |
| 3.3.3 Categorías de salida | 33 |
| 3.4 Arquitectura ARTMAP difusa como regulador de voltaje | 34 |
| 3.4.1 Resultados para una condición inicial medianamente crítica | 35 |
| 3.4.2 Resultados para una condición inicial no crítica | 38 |
| 3.4.3 Resultados para una condición inicial crítica | 41 |
| 3.4.4 Análisis de los resultados | 44 |
| 3.5 Uso de la arquitectura ARTMAP difusa para sintonizar un controlador difuso | 45 |
| 3.5.1 Selección y funcionamiento del controlador difuso | 46 |
| 3.5.2 Sintonización del controlador difuso seleccionado a través de la arquitectura ARTMAP difusa | 49 |
| 3.5.3 Configuración resultante de la arquitectura ARTMAP difusa | 50 |
| 3.6 Desarrollo del programa de simulación | 51 |
| CAPÍTULO 4. PRUEBAS Y RESULTADOS | 59 |
| 4.1 Introducción | 59 |
| 4.2 Respuestas sin evento | 60 |
| 4.2.1 Resultados para una condición inicial no crítica | 60 |
| 4.2.2 Resultados para una condición inicial crítica | 63 |
| 4.2.3 Análisis de los resultados | 66 |
| 4.3 Respuestas ante fallas de corto circuito trifásicas sin apertura de línea | 67 |
| 4.3.1 Resultados para una condición inicial no crítica | 68 |
| 4.3.2 Resultados para una condición inicial crítica | 75 |
| 4.3.3 Análisis de los resultados | 82 |
| 4.4 Respuestas ante fallas de corto circuito trifásicas con apertura de línea | 84 |
| 4.4.1 Resultados para una condición inicial no crítica | 85 |
| 4.4.2 Resultados para una condición inicial crítica | 88 |
| 4.4.3 Análisis de los resultados | 92 |

| | |
|---|------------|
| 4.5 Respuestas ante cambios de carga | 93 |
| 4.5.1 Análisis de los resultados | 97 |
| 4.6 Respuestas ante cambios de referencia | 97 |
| 4.6.1 Resultados para una condición inicial no crítica | 98 |
| 4.6.2 Resultados para una condición inicial crítica | 101 |
| 4.6.3 Análisis de los resultados | 105 |
| CAPÍTULO 5. CONCLUSIONES, APORTACIONES Y RECOMENDACIONES PARA TRABAJOS FUTUROS | 107 |
| 5.1 Conclusiones | 107 |
| 5.2 Aportaciones | 109 |
| 5.3 Recomendaciones para trabajos futuros | 109 |
| REFERENCIAS | 111 |
| APÉNDICE A. SISTEMA MÁQUINA BUS INFINITO BAJO PRUEBA | 115 |
| A.1 Modelo del generador síncrono | 115 |
| A.2 Modelo de la red de transmisión | 116 |
| A.3 Modelo del gobernador de velocidad y de la turbina de vapor | 117 |
| A.4 Modelo del sistema de excitación y del regulador de voltaje | 117 |
| A.5 Ecuaciones dinámicas | 118 |
| A.6 Determinación de las condiciones iniciales | 119 |
| A.7 Datos del sistema bajo prueba | 122 |
| APÉNDICE B. CONDICIONES INICIALES | 123 |
| APÉNDICE C. PROGRAMA DESARROLLADO | 125 |
| C.1 Ejecución del programa | 125 |
| C.2 Código del programa | 128 |
| C.2.1 Programa principal | 129 |
| C.2.2 Módulos de declaración de variables | 140 |
| C.2.3 Subrutinas referentes a la máquina síncrona | 140 |
| C.2.4 Subrutinas referentes a la arquitectura ARTMAP difusa | 143 |
| C.2.5 Subrutinas referentes a los eventos a simular | 147 |
| C.3 Archivos de entrada y salida | 148 |
| C.3.1 Archivos de entrada | 149 |
| C.3.2 Archivos de salida | 150 |
| APÉNDICE D. FUNDAMENTOS DE CONTROL DIFUSO | 156 |
| D.1 Conjuntos difusos | 156 |
| D.2 Operaciones con conjuntos difusos | 158 |
| D.3 Sistema de control difuso | 159 |
| D.4 Controles difusos | 160 |

| | |
|--|------------|
| APÉNDICE E. CONTROL CON REDES NEURONALES | 165 |
| E.1 Modelado de un controlador existente | 166 |
| E.2 Identificación de sistemas | 166 |
| E.3 Identificación del sistema inverso | 166 |
| E.4 Diferenciación de un modelo | 168 |
| APÉNDICE F. CONTROLADOR PROPUESTO COMPARADO CON UN CONTROLADOR DIFUSO CON PARÁMETROS FIJOS ANTE UNA FALLA TRIFÁSICA | 169 |

ÍNDICE DE FIGURAS

CAPÍTULO 2

| | | |
|--------------------|--|----|
| Figura 2.1. | Estructura general de una red neuronal ART1. | 9 |
| Figura 2.2. | Representación esquemática de una red neuronal ART1. | 11 |
| Figura 2.3. | Diagrama de bloques de un sistema ARTMAP. | 15 |
| Figura 2.4. | Arquitectura ARTMAP difusa (FAM). | 16 |
| Figura 2.5. | (a) Interpretación geométrica de un vector de pesos \mathbf{w}_k^b con codificación con complemento para un vector de entrada bidimensional. (b) Expansión durante el entrenamiento de R_K a $R_K \oplus \mathbf{b}$ con aprendizaje rápido. | 21 |
| Figura 2.6. | Rectángulo R_K con aprendizaje rápido y codificación con complemento. | 23 |

CAPÍTULO 3

| | | |
|--------------------|---|----|
| Figura 3.1. | Procedimiento inicial para la aplicación de la arquitectura FAM como regulador de voltaje de un generador síncrono. (a) Entrenamiento. (b) Producción. | 27 |
| Figura 3.2. | Ejemplo de los dos tipos de acondicionamiento de una variable real para que sus valores oscilen en el intervalo $[0, 1]$. | 28 |
| Figura 3.3. | Procedimiento para la aplicación de la arquitectura FAM como regulador de voltaje de un generador síncrono incluyendo las variables adecuadas así como su acondicionamiento. (a) Entrenamiento. (b) Producción. | 31 |
| Figura 3.4. | Falla de corto circuito trifásica para un sistema máquina bus infinito. | 32 |
| Figura 3.5. | Voltaje en terminales para la condición inicial No. 1 con 1, 5 y 10 CIE. | 35 |
| Figura 3.6. | Índice de comportamiento del voltaje en terminales para la condición inicial No. 1 con 1, 5 y 10 CIE. | 35 |
| Figura 3.7. | Voltaje de campo para la condición inicial No. 1 con 1, 5 y 10 CIE. | 36 |
| Figura 3.8. | Ángulo de carga para la condición inicial No. 1 con 1, 5 y 10 CIE. | 36 |
| Figura 3.9. | Potencia activa para la condición inicial No. 1 con 1, 5 y 10 CIE. | 37 |

| | | |
|---------------------|---|----|
| Figura 3.10. | Potencia reactiva para la condición inicial No. 1 con 1, 5 y 10 CIE. | 37 |
| Figura 3.11. | Voltaje en terminales para la condición inicial No. 8 con 1, 5 y 10 CIE. | 38 |
| Figura 3.12. | Índice de comportamiento del voltaje en terminales para la condición inicial No. 8 con 1, 5 y 10 CIE. | 38 |
| Figura 3.13. | Voltaje de campo para la condición inicial No. 8 con 1, 5 y 10 CIE. | 39 |
| Figura 3.14. | Ángulo de carga para la condición inicial No. 8 con 1, 5 y 10 CIE. | 39 |
| Figura 3.15. | Potencia activa para la condición inicial No. 8 con 1, 5 y 10 CIE. | 40 |
| Figura 3.16. | Potencia reactiva para la condición inicial No. 8 con 1, 5 y 10 CIE. | 40 |
| Figura 3.17. | Voltaje en terminales para la condición inicial No. 28 con 1, 5 y 10 CIE. | 41 |
| Figura 3.18. | Índice de comportamiento del voltaje en terminales para la condición inicial No. 28 con 1, 5 y 10 CIE. | 41 |
| Figura 3.19. | Voltaje de campo para la condición inicial No. 28 con 1, 5 y 10 CIE. | 42 |
| Figura 3.20. | Ángulo de carga para la condición inicial No. 28 con 1, 5 y 10 CIE. | 42 |
| Figura 3.21. | Potencia activa para la condición inicial No. 28 con 1, 5 y 10 CIE. | 43 |
| Figura 3.22. | Potencia reactiva para la condición inicial No. 28 con 1, 5 y 10 CIE. | 43 |
| Figura 3.23. | Voltaje de campo para la condición inicial No. 1 proporcionado por la arquitectura FAM con 10 CIE sobre las 33 categorías definidas en la red FART _b . | 44 |
| Figura 3.24. | Procedimiento de aplicación de la arquitectura FAM para sintonizar un controlador difuso. (a) Entrenamiento. (b) Producción. | 46 |
| Figura 3.25. | Representación gráfica de la operación del controlador difuso PI tipo Mamdani propuesto para regular el voltaje del sistema máquina bus infinito bajo prueba. | 47 |
| Figura 3.26. | Configuración resultante de la arquitectura FAM para su aplicación en la sintonización del controlador difuso seleccionado. | 51 |
| Figura 3.27. | Diagrama de flujo del programa principal desarrollado (1ª parte). | 52 |
| | Diagrama de flujo del programa principal desarrollado (2ª parte). | 53 |
| | Diagrama de flujo del programa principal desarrollado (3ª parte). | 54 |
| | Diagrama de flujo del programa principal desarrollado (4ª parte). | 55 |
| Figura 3.28. | Diagrama de flujo de la aplicación de la arquitectura FAM en la regulación de voltaje (1ª parte). | 56 |
| | Diagrama de flujo de la aplicación de la arquitectura FAM en la regulación de voltaje (2ª parte). | 57 |
| CAPÍTULO 4 | | |
| Figura 4.1. | Voltaje en terminales para la condición inicial No. 25 sin evento. | 60 |
| Figura 4.2. | Voltaje de campo para la condición inicial No. 25 sin evento. | 61 |

| | | |
|---------------------|--|----|
| Figura 4.3. | Ángulo de carga para la condición inicial No. 25 sin evento. | 61 |
| Figura 4.4. | Potencia activa para la condición inicial No. 25 sin evento. | 62 |
| Figura 4.5. | Potencia reactiva para la condición inicial No. 25 sin evento. | 62 |
| Figura 4.6. | Pesos X_1 y X_2 para la condición inicial No. 25 sin evento. | 63 |
| Figura 4.7. | Voltaje en terminales para la condición inicial No. 30 sin evento. | 63 |
| Figura 4.8. | Voltaje de campo para la condición inicial No. 30 sin evento. | 64 |
| Figura 4.9. | Ángulo de carga para la condición inicial No. 30 sin evento. | 64 |
| Figura 4.10. | Potencia activa para la condición inicial No. 30 sin evento. | 65 |
| Figura 4.11. | Potencia reactiva para la condición inicial No. 30 sin evento. | 65 |
| Figura 4.12. | Pesos X_1 y X_2 para la condición inicial No. 30 sin evento. | 66 |
| Figura 4.13. | Voltaje en terminales para la condición inicial No. 25 ante una falla de 12 ciclos de duración. | 68 |
| Figura 4.14. | Índice de comportamiento del voltaje en terminales para la condición inicial No. 25 ante una falla de 12 ciclos de duración. | 68 |
| Figura 4.15. | Voltaje de campo para la condición inicial No. 25 ante una falla de 12 ciclos de duración. | 69 |
| Figura 4.16. | Ángulo de carga para la condición inicial No. 25 ante una falla de 12 ciclos de duración. | 69 |
| Figura 4.17. | Potencia activa para la condición inicial No. 25 ante una falla de 12 ciclos de duración. | 70 |
| Figura 4.18. | Potencia reactiva para la condición inicial No. 25 ante una falla de 12 ciclos de duración. | 70 |
| Figura 4.19. | Pesos X_1 y X_2 para la condición inicial No. 25 ante una falla de 12 ciclos de duración. | 71 |
| Figura 4.20. | Voltaje en terminales para la condición inicial No. 25 ante una falla con una duración igual al tiempo crítico de liberación de cada controlador. | 71 |
| Figura 4.21. | Índice de comportamiento del voltaje en terminales para la condición inicial No. 25 ante una falla con una duración igual al tiempo crítico de liberación de cada controlador. | 72 |
| Figura 4.22. | Voltaje de campo para la condición inicial No. 25 ante una falla con una duración igual al tiempo crítico de liberación de cada controlador. | 72 |
| Figura 4.23. | Ángulo de carga para la condición inicial No. 25 ante una falla con una duración igual al tiempo crítico de liberación de cada controlador. | 73 |
| Figura 4.24. | Potencia activa para la condición inicial No. 25 ante una falla con una duración igual al tiempo crítico de liberación de cada controlador. | 73 |
| Figura 4.25. | Potencia reactiva para la condición inicial No. 25 ante una falla con una duración igual al tiempo crítico de liberación de cada controlador. | 74 |
| Figura 4.26. | Pesos X_1 y X_2 para la condición inicial No. 25 ante una falla de 24.30 ciclos de duración. | 74 |
| Figura 4.27. | Voltaje en terminales para la condición inicial No. 30 ante una falla de 1.8 ciclos de duración. | 75 |

| | | |
|---------------------|--|----|
| Figura 4.28. | Índice de comportamiento del voltaje en terminales para la condición inicial No. 30 ante una falla de 1.8 ciclos de duración. | 75 |
| Figura 4.29. | Voltaje de campo para la condición inicial No. 30 ante una falla de 1.8 ciclos de duración. | 76 |
| Figura 4.30. | Ángulo de carga para la condición inicial No. 30 ante una falla de 1.8 ciclos de duración. | 76 |
| Figura 4.31. | Potencia activa para la condición inicial No. 30 ante una falla de 1.8 ciclos de duración. | 77 |
| Figura 4.32. | Potencia reactiva para la condición inicial No. 30 ante una falla de 1.8 ciclos de duración. | 77 |
| Figura 4.33. | Pesos X_1 y X_2 para la condición inicial No. 30 ante una falla de 1.8 ciclos de duración. | 78 |
| Figura 4.34. | Voltaje en terminales para la condición inicial No. 30 ante una falla con una duración igual al tiempo crítico de liberación de cada controlador. | 78 |
| Figura 4.35. | Índice de comportamiento del voltaje en terminales para la condición inicial No. 30 ante una falla con una duración igual al tiempo crítico de liberación de cada controlador. | 79 |
| Figura 4.36. | Voltaje de campo para la condición inicial No. 30 ante una falla con una duración igual al tiempo crítico de liberación de cada controlador. | 79 |
| Figura 4.37. | Ángulo de carga para la condición inicial No. 30 ante una falla con una duración igual al tiempo crítico de liberación de cada controlador. | 80 |
| Figura 4.38. | Potencia activa para la condición inicial No. 30 ante una falla con una duración igual al tiempo crítico de liberación de cada controlador. | 80 |
| Figura 4.39. | Potencia reactiva para la condición inicial No. 30 ante una falla con una duración igual al tiempo crítico de liberación de cada controlador. | 81 |
| Figura 4.40. | Pesos X_1 y X_2 para la condición inicial No. 30 ante una falla de 10.74 ciclos de duración. | 81 |
| Figura 4.41. | (a) Sistema bajo prueba original. (b) Sistema propuesto para simular fallas trifásicas con apertura de línea. | 84 |
| Figura 4.42. | Voltaje en terminales para la condición inicial No. 25 ante una falla de 21 ciclos de duración con apertura de línea. | 85 |
| Figura 4.43. | Índice de comportamiento del voltaje en terminales para la condición inicial No. 25 ante una falla de 21 ciclos de duración con apertura de línea. | 85 |
| Figura 4.44. | Voltaje de campo la condición inicial No. 25 ante una falla de 21 ciclos de duración con apertura de línea. | 86 |
| Figura 4.45. | Ángulo de carga para la condición inicial No. 25 ante una falla de 21 ciclos de duración con apertura de línea. | 86 |
| Figura 4.46. | Potencia activa para la condición inicial No. 25 ante una falla de 21 ciclos de duración con apertura de línea. | 87 |

| | | |
|---------------------|---|-----|
| Figura 4.47. | Potencia reactiva para la condición inicial No. 25 ante una falla de 21 ciclos de duración con apertura de línea. | 87 |
| Figura 4.48. | Pesos X_1 y X_2 para la condición inicial No. 25 ante una falla de 21 ciclos de duración con apertura de línea. | 88 |
| Figura 4.49. | Voltaje en terminales para la condición inicial No. 30 ante una falla de 5.4 ciclos de duración con apertura de línea. | 88 |
| Figura 4.50. | Índice de comportamiento del voltaje en terminales para la condición inicial No. 30 ante una falla de 5.4 ciclos de duración con apertura de línea. | 89 |
| Figura 4.51. | Voltaje de campo para la condición inicial No. 30 ante una falla de 5.4 ciclos de duración con apertura de línea. | 89 |
| Figura 4.52. | Ángulo de carga para la condición inicial No. 30 ante una falla de 5.4 ciclos de duración con apertura de línea. | 90 |
| Figura 4.53. | Potencia activa para la condición inicial No. 30 ante una falla de 5.4 ciclos de duración con apertura de línea. | 90 |
| Figura 4.54. | Potencia reactiva para la condición inicial No. 30 ante una falla de 5.4 ciclos de duración con apertura de línea. | 91 |
| Figura 4.55. | Pesos X_1 y X_2 para la condición inicial No. 30 ante una falla de 5.4 ciclos de duración con apertura de línea. | 91 |
| Figura 4.56. | Voltaje en terminales ante los cambios de carga. | 93 |
| Figura 4.57. | Índice de comportamiento del voltaje en terminales ante los cambios de carga. | 94 |
| Figura 4.58. | Voltaje de campo ante los cambios de carga. | 94 |
| Figura 4.59. | Ángulo de carga ante los cambios de carga. | 95 |
| Figura 4.60. | Potencia activa ante los cambios de carga sobre la manipulación de la entrada del gobernador de velocidad u_g . | 95 |
| Figura 4.61. | Potencia reactiva ante los cambios de carga. | 96 |
| Figura 4.62. | Pesos X_1 y X_2 ante los cambios de carga. | 96 |
| Figura 4.63. | Voltaje en terminales para la condición inicial No. 25 ante los cambios de referencia. | 98 |
| Figura 4.64. | Índice de comportamiento del voltaje en terminales para la condición inicial No. 25 ante los cambios de referencia. | 98 |
| Figura 4.65. | Voltaje de campo para la condición inicial No. 25 ante los cambios de referencia. | 99 |
| Figura 4.66. | Ángulo de carga para la condición inicial No. 25 ante los cambios de referencia. | 99 |
| Figura 4.67. | Potencia activa para la condición inicial No. 25 ante los cambios de referencia. | 100 |
| Figura 4.68. | Potencia reactiva para la condición inicial No. 25 ante los cambios de referencia. | 100 |
| Figura 4.69. | Pesos X_1 y X_2 para la condición inicial No. 25 ante los cambios de referencia. | 101 |
| Figura 4.70. | Voltaje en terminales para la condición inicial No. 30 ante los cambios de referencia. | 101 |
| Figura 4.71. | Índice de comportamiento del voltaje en terminales para la condición inicial No. 30 ante los cambios de referencia. | 102 |

| | | |
|---------------------|--|-----|
| Figura 4.72. | Voltaje de campo para la condición inicial No. 30 ante los cambios de referencia. | 102 |
| Figura 4.73. | Ángulo de carga para la condición inicial No. 30 ante los cambios de referencia. | 103 |
| Figura 4.74. | Potencia activa para la condición inicial No. 30 ante los cambios de referencia. | 103 |
| Figura 4.75. | Potencia reactiva para la condición inicial No. 30 ante los cambios de referencia. | 104 |
| Figura 4.76. | Pesos X_1 y X_2 para la condición inicial No. 30 ante los cambios de referencia. | 104 |
| APÉNDICE A | | |
| Figura A.1. | Diagrama del sistema máquina bus infinito utilizado en el presente trabajo. | 115 |
| Figura A.2. | Diagrama de bloques del sistema gobernador-turbina. | 117 |
| Figura A.3. | Diagrama de bloques del regulador ST1. | 118 |
| Figura A.4. | Diagrama vectorial del sistema máquina bus infinito en estado subtransitorio. | 120 |
| ÁPENDICE C | | |
| Figura C.1. | Diagrama de bloques de las opciones que ofrece el programa desarrollado en tiempo de ejecución. | 126 |
| Figura C.2. | Pantalla resultante de simular el generador sin regulador, ante una falla en el bus incluyendo aislamiento de falla, para la condición inicial No.1. | 127 |
| Figura C.3. | Pantalla resultante de simular el generador con regulador ST1 ante cambios de carga. | 127 |
| Figura C.4. | Pantalla resultante de simular el generador con la arquitectura FAM actuando como sintonizador de un controlador difuso, ante cambios de referencia para la condición inicial No.25. | 128 |
| APÉNDICE D | | |
| Figura D.1. | Sistema de control difuso. | 159 |
| Figura D.2. | Control difuso tipo Mamdani para dos entradas y una salida. | 161 |
| Figura D.3. | Controles difusos tipo Sugeno y Tsukamoto para dos entradas y una salida. | 164 |
| APÉNDICE E | | |
| Figura E.1. | Configuraciones básicas para el entrenamiento de redes neuronales en la identificación de sistemas de control. (a) Modelado de un controlador existente. (b) Identificación de sistemas. (c) Identificación de un sistema inverso. (d) Retropropagación a través de un modelo progresivo de la planta para determinar los errores del controlador. | 167 |

APÉNDICE F

| | | |
|-------------------|---|-----|
| Figura F.1 | Voltaje en terminales para la condición inicial No. 30 ante una falla de 1.8 ciclos de duración. | 170 |
| Figura F.2 | Índice de comportamiento del voltaje en terminales para la condición inicial No. 30 ante una falla de 1.8 ciclos de duración. | 170 |
| Figura F.3 | Voltaje de campo para la condición inicial No. 30 ante una falla de 1.8 ciclos de duración. | 171 |
| Figura F.4 | Ángulo de carga para la condición inicial No. 30 ante una falla de 1.8 ciclos de duración. | 171 |
| Figura F.5 | Potencia activa para la condición inicial No. 30 ante una falla de 1.8 ciclos de duración. | 172 |
| Figura F.6 | Potencia reactiva para la condición inicial No. 30 ante una falla de 1.8 ciclos de duración. | 172 |

ÍNDICE DE TABLAS

CAPÍTULO 2

| | | |
|-------------------|---|----|
| Tabla 2.1. | Comparación entre las operaciones de la red ART1 y la red FART. | 13 |
|-------------------|---|----|

CAPÍTULO 3

| | | |
|-------------------|--|----|
| Tabla 3.1. | Parámetros para el entrenamiento. | 30 |
| Tabla 3.2. | Condiciones iniciales bajo las que el regulador ST1 responde eficientemente. | 32 |
| Tabla 3.3. | Condiciones iniciales críticas. | 33 |
| Tabla 3.4. | Número de categorías generadas en las redes FART _a y FART _b utilizando las condiciones iniciales 1, 1-5 y 1-10 para el entrenamiento de la arquitectura FAM. | 33 |
| Tabla 3.5. | Base de reglas para el controlador difuso. | 48 |

CAPÍTULO 4

| | | |
|-------------------|--|----|
| Tabla 4.1. | Condiciones iniciales utilizadas para las distintas pruebas. | 60 |
| Tabla 4.2. | Errores en estado estacionario con ambos controladores para las dos condiciones iniciales simuladas. | 66 |
| Tabla 4.3. | Comparación de tiempos críticos entre el regulador ST1 y el controlador propuesto. | 83 |
| Tabla 4.4. | Cambios de potencia activa para la simulación de los cambios de carga. | 93 |
| Tabla 4.5. | Cambios del voltaje de referencia. | 97 |

APÉNDICE B

| | | |
|-------------------|--|-----|
| Tabla B.1. | Condiciones iniciales consideradas. | 123 |
| | Condiciones iniciales consideradas (continuación). | 124 |

APÉNDICE C

| | | |
|------------------|--|-----|
| Tabla C.1 | Nombres de los archivos de salida para la escritura de resultados en función de la simulación realizada. | 150 |
|------------------|--|-----|

APÉNDICE D

| | | |
|-------------------|--|-----|
| Tabla D.1. | Funciones de membresía más comunes y su caracterización matemática. | 157 |
| | Funciones de membresía más comunes y su caracterización matemática (continuación). | 158 |
| Tabla D.2. | Operaciones básicas entre conjuntos nítidos y difusos. | 158 |
| Tabla D.3. | Base de reglas para el control difuso de la figura D.2. | 160 |
| Tabla D.4. | Métodos de dedifusificación más comunes. | 162 |
| | Métodos de dedifusificación más comunes (continuación). | 163 |

ABREVIATURAS

En el presente trabajo se utilizan las siguientes abreviaturas.

| | |
|-----------|--|
| ANFIS | Sistema adaptable de inferencias neurodifusas (adaptive neuro-fuzzy inference system). |
| ANN | Red neuronal artificial (artificial neural network). |
| ART | Teoría de resonancia adaptable (adaptive resonance theory). |
| ART1 | Red neuronal básica de la teoría de resonancia adaptable. |
| ART2 | Red neuronal creada a partir de la teoría de resonancia adaptable para procesar y clasificar patrones de entrada tanto binarios como analógicos. |
| ART2-A | Red neuronal creada a partir de la teoría de resonancia adaptable para procesar y clasificar patrones de entrada tanto binarios como analógicos con una velocidad dos o tres veces mayor que la red ART2 |
| ART3 | Red neuronal creada a partir de la teoría de resonancia adaptable para llevar a cabo una búsqueda paralela, o prueba de hipótesis, de códigos de reconocimiento distribuidos en una jerarquía de una red multinivel. |
| ARTMAP | Red neuronal creada a partir de la teoría de resonancia adaptable para el mapeo multidimensional entre vectores de entrada binarios. |
| CIE | Condiciones iniciales para el entrenamiento. |
| CP | Control propuesto. |
| DSP | Procesador de señales digitales (digital signal processor). |
| FALCONART | Red de control con aprendizaje adaptable difuso usando la teoría de resonancia adaptable (fuzzy adaptive learning control network using the adaptive resonance theory). |
| FAM | Red neurodifusa creada a partir de la teoría de resonancia adaptable para el mapeo multidimensional entre vectores de entrada analógicos (fuzzy ARTMAP). |
| FAMR | Red neurodifusa creada a partir de la teoría de resonancia adaptable para el mapeo multidimensional entre vectores de entrada analógicos con relevancias de entrada (fuzzy ARTMAP with input relevances). |
| FART | Red neurodifusa creada de la incorporación de operadores difusos en la red ART1 (fuzzy ART). |
| LTM | Memoria a largo plazo (long-term memory). |
| MVA | Mega volts-amperes. |

| | |
|-----|--|
| PI | Proporcional integral. |
| PSS | Estabilizador de sistemas de potencia (power system stabilizer). |
| RAV | Regulador automático de voltaje. |
| SEP | Sistema eléctrico de potencia. |
| ST1 | Sistema de regulación de tipo estático No. 1. |
| STM | Memoria a corto plazo (short-term memory). |
| VAR | Volts-amperes reactivos. |

CAPÍTULO 1

GENERALIDADES

1.1 INTRODUCCIÓN

Los sistemas eléctricos de potencia (SEP's) son el medio por el cual se genera, transmite y distribuye la energía eléctrica. Un SEP apropiadamente diseñado y operado debe ser capaz de cumplir con los siguientes requisitos fundamentales [1]:

- Proveer satisfactoriamente potencia activa y reactiva bajo los continuos cambios en la demanda de carga.
- Suministrar energía con el mínimo costo e impacto ambiental.
- La calidad de la energía suministrada tiene que encontrarse dentro de determinados estándares relacionados con: la constancia de la frecuencia, la constancia del voltaje y el nivel de confiabilidad.

Para cubrir estos requisitos, se utiliza una serie de controles distribuidos a lo largo de todo el SEP, el cual se divide en tres grandes subsistemas: generación, transmisión y distribución de la energía eléctrica. Respecto al subsistema de generación, se tienen los controles para el primo motor y para la excitación. Los controles del primo motor están estrechamente relacionados con la regulación de velocidad y el control de las variables del sistema que proporciona la energía primaria, como presiones, temperaturas y flujos [2]; mientras que la función del control de excitación es regular el voltaje del generador y la potencia reactiva de salida [3].

Los controles del subsistema de transmisión incluyen dispositivos de control para el voltaje y la potencia, como compensadores estáticos de VARS, condensadores síncronos, reactores y capacitores conmutados, transformadores con cambio de taps, transformadores de cambio de fase y controles de transmisión de alto voltaje en corriente directa [1].

Los controles mencionados mantienen los voltajes, la frecuencia y otras variables del sistema dentro de sus respectivos límites aceptables, lo que contribuye a una operación satisfactoria de todo el SEP. Sin embargo, el gradual incremento en tamaño y complejidad

de los SEP's a través de los años, debido principalmente al aumento y diversidad de demanda de carga y a las interconexiones con otros SEP's, ha complicado enormemente las labores de control, puesto que los controles utilizados en estos sistemas han sido diseñados y desarrollados tradicionalmente en base a la teoría de control convencional. Es por ello que recientemente ha surgido un gran interés por técnicas de control más sofisticadas.

Actualmente existe una gran variedad de técnicas y metodologías para controlar cualquier proceso o máquina eléctrica de forma más eficiente que las convencionales, dentro de las cuales se encuentran las que involucran algoritmos difusos y relacionados. La principal ventaja que presentan estas técnicas con respecto a las convencionales es la emulación del razonamiento humano e inteligencia biológica por medio de la manipulación de información de naturaleza lingüística. Además, estas técnicas poseen una gran flexibilidad para combinarse con las redes neuronales artificiales, las cuales son muy conocidas por su habilidad de aprender, recordar, organizar, categorizar y generalizar a partir de datos o patrones de entrenamiento. Así, de la integración de algoritmos difusos con las redes neuronales nacen los sistemas neurodifusos.

1.2 OBJETIVO DE LA TESIS

El objetivo principal de este trabajo de tesis es diseñar un control neurodifuso mediante la aplicación de la arquitectura ARTMAP difusa aprovechando sus características y funcionalidad para regular el voltaje en terminales de un generador síncrono, con la finalidad de obtener un desempeño y una operación más eficientes que los controles convencionales bajo diferentes condiciones de operación y ante disturbios diversos.

1.3 JUSTIFICACIÓN

La principal fuente de energía eléctrica en los SEP's la constituyen los generadores síncronos, por lo que una adecuada y confiable regulación de voltaje de estas máquinas es de suma importancia. Esta regulación se realiza por medio de un regulador automático de voltaje (RAV). Los RAV's que se encuentran actualmente en operación funcionan eficientemente para un determinado punto de operación; pero, debido a que están diseñados en base a la teoría de control convencional (lo que implica la obtención de modelos matemáticos detallados y explícitos de los generadores), generalmente resulta complicado realizar modificaciones en su estructura para adaptarlos a nuevos puntos de operación [4, 5], por lo que algunas veces tienen que ser sintonizados y reajustados manualmente.

En atención a lo citado en el párrafo anterior, en el presente trabajo se propone desarrollar un regulador de voltaje neurodifuso que supere las limitantes de los RAV's convencionales en un sistema máquina bus infinito. La red o arquitectura neurodifusa propuesta es la ARTMAP difusa (FAM por sus siglas en inglés), dada la estabilidad y el tipo de aprendizaje que la caracterizan en el reconocimiento, categorización y relación de patrones de entrada analógicos.

1.4 ESTADO DEL ARTE

El control de SEP's y de procesos y máquinas eléctricas en general es una de las áreas que más se ha visto beneficiada con el surgimiento y el desarrollo de los sistemas neurodifusos. En el transcurso de las últimas décadas se han realizado numerosos trabajos en este rubro a nivel mundial.

En 1999, investigadores de la Universidad de Tecnología Victoria en Melbourne, Australia, desarrollaron un estabilizador de sistemas de potencia (PSS por sus siglas en inglés) difuso usando las variaciones de la velocidad y de la potencia activa como variables de entrada del controlador [6]. El mecanismo de inferencia del controlador difuso utilizado se representa por una tabla de decisión de 7×7 premisas; es decir, 49 reglas difusas. Este controlador es sintonizado por una red neuronal previamente entrenada para identificar el estado del sistema de potencia, lo que permite que el PSS difuso se adapte a cambios en las condiciones de operación. El PSS difuso resultante fue probado a través de diversas simulaciones usando un modelo no lineal de una máquina bus infinito, obteniéndose respuestas superiores a las de un PSS difuso con parámetros fijos y a las de un PSS convencional (lineal).

Otro trabajo similar al anterior fue desarrollado en 2006 por investigadores de las Universidades de Yarmouk y de Ciencia y Tecnología de Jordania, ambas en Jordania, así como de la Compañía de Petróleo Agip, en Libia [7]. En tal trabajo se sintoniza un PSS convencional de adelanto-atraso de fase a partir de mediciones en línea de las condiciones de carga del generador usando un sistema adaptable de inferencias neurodifusas (arquitectura ANFIS por sus siglas en inglés). Los patrones de entrada, conformados por las potencias activa y reactiva, se asocian con los correspondientes patrones de salida deseados, que constan de los parámetros del PSS (ganancia y constante de tiempo), sobre una amplia gama de condiciones de operación. Los resultados obtenidos ante pequeños disturbios y bajo diferentes condiciones de carga demostraron la efectividad y robustez del ANFIS-PSS propuesto en el mejoramiento de la estabilidad del sistema de potencia en tales condiciones.

Recientemente, en 2010 un trabajo interesante fue publicado por investigadores del Departamento de Ingeniería Eléctrica y Electrónica del Instituto de Ciencia y Tecnología de Berhampur en Orissa, India [8]. En este trabajo se compara el rendimiento de controladores convencional integral, difuso y neurodifuso híbrido en el control automático de generación para un sistema hidro-térmico interconectado. El control difuso utilizado es un tipo Sugeno con 25 reglas y 5 funciones de membresía para cada variable; los parámetros de este controlador se ajustan por medio de una red neuronal con retropropagación para formar el controlador neurodifuso híbrido. Los resultados obtenidos después de diferentes simulaciones muestran que el comportamiento transitorio del sistema interconectado es mejor con el controlador neurodifuso que con los otros dos controladores cuando se suscita un disturbio en la carga, ya sea en la unidad hidráulica o en la térmica.

La arquitectura FAM regularmente es usada en el procesamiento y clasificación de información para emitir o predecir algún tipo de diagnóstico, ya sea con su estructura original [9] o con algún tipo de relevancia de entrada (conocida como FAMR por sus siglas en inglés) [10]; esta última comúnmente es utilizada como un estimador de probabilidades. Una de las aplicaciones más recientes de la arquitectura FAMR es la realizada en el presente año por un grupo de investigadores de la Universidad de Washington Central en

Washington, E.U.A. [11], en la cual se propone la arquitectura FAMR para predecir las actividades biológicas para inhibidores potenciales de proteasas VIH-1 usando un reducido conjunto de datos moleculares. Los resultados obtenidos en este trabajo son alentadores y abren aún más la aplicabilidad de la arquitectura FAM en el campo de la bioinformación.

Una de las pocas aplicaciones de la arquitectura FAM en el campo del control de máquinas eléctricas es el trabajo realizado en el Departamento de Ingeniería Eléctrica y de Cómputo de la Universidad Tecnológica Lawrence en Southfield, Michigan, E.U.A., en 2010 [12]. Este trabajo usa la arquitectura FAM para copiar la dinámica de un controlador H_∞ y luego lo sustituye en el control de un sistema de dirección eléctrica de un automóvil eléctrico moderno. Los resultados obtenidos muestran un apego fiel por parte de la arquitectura FAM al funcionamiento del controlador H_∞ , pero con la ventaja evidente de un algoritmo ampliamente reducido en lo que se refiere a complejidad matemática.

Particularmente, en el Departamento de Ingeniería Eléctrica de la Sección de Estudios de Posgrado e Investigación de la Escuela Superior de Ingeniería Mecánica y Eléctrica del Instituto Politécnico Nacional en la Ciudad de México, México (lugar de realización del presente trabajo), a la fecha no se ha desarrollado un control neurodifuso basado en la arquitectura FAM para regular el voltaje o cualquier otra variable de salida de un generador síncrono; aunque sí se han realizado controles neurodifusos para este tipo de máquinas utilizando otras arquitecturas o modelos de redes neurodifusas. Por ejemplo, en 2001 se concluyó un trabajo de investigación en el que se regula el voltaje en terminales de un generador síncrono conectado a un bus infinito usando una red de control con aprendizaje adaptable difuso basada en la teoría de resonancia adaptable (FALCONART por sus siglas en inglés) [13]; los resultados de este trabajo fueron favorables ante fallas de corto circuito trifásicas. En 2006 se concluyó otro trabajo relacionado con la aplicación de algoritmos inteligentes en la regulación de voltaje de una máquina síncrona [14]; trabajo en el que se desarrolla un control difuso sintonizado por búsqueda tabú y algoritmos genéticos, obteniéndose mayor robustez y confiabilidad ante fallas de corto circuito trifásicas.

Los trabajos aquí citados son un claro ejemplo de la superioridad de los controles neurodifusos, e inteligentes en general, sobre los controles convencionales en cuanto a robustez y confiabilidad.

1.5 CONTENIDO DE LA TESIS

El presente trabajo de tesis se compone de cinco capítulos y seis apéndices.

El Capítulo 1 (presente) comprende las generalidades necesarias e introductorias para el desarrollo del presente trabajo.

En el Capítulo 2 se presenta inicialmente una breve descripción de la teoría de resonancia adaptable así como del modelo básico que la caracteriza. Después se muestra la forma en que se puede obtener un modelo generalizado de esta teoría a través de la aplicación de los conceptos de la lógica difusa. Finalmente, se describen a detalle la estructura, el funcionamiento y el algoritmo de la arquitectura ARTMAP difusa (FAM).

En el Capítulo 3 se presentan los elementos necesarios para la aplicación de la arquitectura FAM como regulador del voltaje en terminales de un generador síncrono

conectado a un bus infinito. Posteriormente, se explora una técnica distinta para utilizar esta arquitectura en la sintonización de un controlador PI difuso tipo Mamdani.

En el Capítulo 4 se presentan los resultados obtenidos con el control propuesto resultante en comparación con los respectivos que ofrece un regulador de voltaje convencional tipo ST1. Estos resultados incluyen la simulación de diferentes disturbios en el sistema bajo prueba, como fallas, cambios de carga y cambios de referencia.

En el Capítulo 5 se presentan las conclusiones derivadas de las simulaciones realizadas y de los resultados obtenidos. También se presentan aportaciones y algunas recomendaciones para trabajos futuros sobre temas relacionados.

El Apéndice A contiene el modelo y los datos del sistema máquina bus infinito utilizado para probar el control desarrollado. Además, en este apéndice se desarrollan las ecuaciones para la obtención de las condiciones iniciales y la definición del resto de las variables de interés.

En el Apéndice B se presentan las condiciones iniciales utilizadas para el desarrollo de las simulaciones

El Apéndice C contiene el código del programa desarrollado así como la forma en que se ejecuta y la forma en que son presentados los resultados.

En el Apéndice D se presenta un resumen de la estructura y funcionamiento de los controles difusos más comunes.

El Apéndice E es una breve descripción de la forma en que las redes neuronales pueden y han sido aplicadas en control.

Finalmente, en el Apéndice F se presenta una comparación entre resultados del control propuesto resultante y de un controlador difuso tipo Mamdani con parámetros fijos.

1.6 APORTACIONES

En el presente trabajo se desarrollan dos metodologías para regular el voltaje en terminales de un generador síncrono por medio de la utilización de la arquitectura FAM. La primera de ellas contempla la sustitución directa de un controlador convencional por la mencionada arquitectura; mientras que en la segunda se usa la arquitectura FAM para inferir los valores apropiados que sintonizan un controlador difuso, siendo esta última con la que se obtiene un mejor comportamiento del sistema máquina bus infinito bajo prueba.

Se desarrolla un programa computacional en lenguaje FORTRAN 90 para simular el sistema máquina bus infinito sin regulador de voltaje, con regulador convencional ST1 debidamente sintonizado o con regulación por medio de la arquitectura FAM; esta última opción se subdivide en otras dos opciones: aplicación de la arquitectura FAM directamente como regulador de voltaje o uso de la misma para sintonizar un controlador difuso PI tipo Mamdani.

El sistema máquina bus infinito bajo prueba se simula incluyendo diferentes disturbios como fallas de corto circuito trifásicas, cambios de carga y cambios de referencia con diferentes sistemas de regulación, lo que permite recopilar más información para una

visualización y un análisis más completos acerca del comportamiento del sistema bajo prueba en diferentes escenarios.

1.7 PUBLICACIONES

Durante la realización del presente trabajo se efectuaron las siguientes publicaciones:

- Martín E. Cárdenas R. y David Romero R., “Aplicación de la arquitectura ARTMAP difusa en la regulación de voltaje de un generador síncrono”, *XII Congreso Nacional de Ingeniería Electromecánica y de Sistemas*, Ciudad de México, México, Nov. 2010.
- Martín E. Cárdenas R. y David Romero R., “Regulación de voltaje de un generador síncrono utilizando la arquitectura ARTMAP difusa”, *XXIII Reunión de Verano de Potencia de la IEEE Sección México*, Acapulco Gro., México, Jul. 2010.

CAPÍTULO 2

ARQUITECTURA ARTMAP DIFUSA

2.1 INTRODUCCIÓN

Las redes neuronales han demostrado una gran habilidad para clasificar, almacenar, recordar y asociar información [15]. Mediante la incorporación de los principios de la lógica difusa [16] en una red neuronal, se obtiene lo que se conoce como *red neurodifusa* (o *sistema neurodifuso*), con la que se logra una mayor flexibilidad en el procesamiento de patrones de entrada, dado que estas redes son capaces de operar tanto con entradas binarias como analógicas. Estos sistemas neurodifusos poseen además una representatividad y velocidad de entrenamiento mayores que los sistemas neuronales convencionales [17].

Actualmente existen muchas redes neurodifusas creadas a partir de la incorporación de los conceptos de la teoría de conjuntos difusos en modelos de redes neuronales convencionales. Dentro de estos modelos se encuentran: el perceptron, las memorias asociativas, las redes de agrupamiento de Kohonen, las redes de energía de Coulomb restringida y las redes desarrolladas a partir de la *teoría de resonancia adaptable* (modelos o redes ART por sus siglas en inglés), entre otras [17].

La arquitectura ARTMAP difusa (FAM) es uno de los sistemas neurodifusos creados a partir de la unión de la teoría de conjuntos difusos y la teoría de resonancia adaptable. Esta arquitectura, propuesta por Gail A. Carpenter y Stephen Grossberg en 1992, es capaz de llevar a cabo un aprendizaje supervisado incremental de categorías de reconocimiento y de mapeos multidimensionales [9, 10].

Puesto que la arquitectura FAM es el resultado de una serie de adaptaciones y modificaciones al modelo básico de la teoría de resonancia adaptable, en este capítulo primeramente se proporciona una breve explicación de esta teoría junto con el modelo básico que la representa, posteriormente se presenta la difusificación de este modelo y, finalmente, se describe a fondo la arquitectura FAM.

2.2 TEORÍA DE RESONANCIA ADAPTABLE

La Teoría de Resonancia Adaptable, que en sus inicios se introdujo como una teoría de procesamiento de información cognitiva humana (Stephen Grossberg, 1976, 1980), fue desarrollada por Gail A. Carpenter y Stephen Grossberg en 1987 para superar el dilema de *estabilidad-plasticidad* que se presenta en los modelos con aprendizaje competitivo [17, 18, 19]. Desde entonces, esta teoría ha conducido al desarrollo de una serie de modelos de redes neuronales ART para el aprendizaje en tiempo real de categorías y reconocimiento de patrones.

Los modelos ART son capaces de aprender categorías de reconocimiento estables en respuesta a secuencias de entrada arbitrarias ya sea rápida ó lentamente [18, 20]. Por lo tanto, al igual que los modelos con aprendizaje competitivo, los modelos ART también cumplen con el propósito de descubrir agrupaciones o categorías a través de aprendizaje no supervisado, sólo que lo realizan de forma controlada; además, pueden producir las agrupaciones por sí mismos sin información previa alguna acerca del número o tipo de categorías.

Dado que estas redes están diseñadas para resolver el dilema de estabilidad-plasticidad, son lo suficientemente estables como para preservar aprendizaje pasado significativo permaneciendo lo suficientemente adaptables como para incorporar nueva información cuando sea que ésta aparezca [17].

Dentro de los modelos básicos desarrollados a partir de la teoría de resonancia adaptable se encuentran: el ART1 (Carpenter y Grossberg, 1987a), que aprende a categorizar de forma estable patrones binarios de entrada presentados en orden arbitrario [18]; el ART2 (Carpenter y Grossberg, 1987b), que aprende a categorizar de forma estable patrones de entrada tanto binarios como analógicos presentados en orden arbitrario [21]; el ART2-A, que realiza la misma función que el ART2 pero con una velocidad dos o tres veces mayor [22]; y el ART3 (Carpenter y Grossberg, 1990a, 1990b), que puede llevar a cabo una búsqueda paralela, o prueba de hipótesis, de códigos de reconocimiento distribuidos en una jerarquía de una red multinivel [23]. Actualmente existen muchas variaciones de estos modelos que han surgido para adaptarse a las demandas de distintas aplicaciones desarrolladas por un gran número de autores.

La red ART1 es la representación de la teoría de resonancia adaptable, lo que la ha convertido en la base del desarrollo de los distintos modelos ART. Por lo tanto, es muy común y conveniente que se emplee esta red para la explicación y entendimiento de dicha teoría.

2.2.1 Estructura y Funcionamiento Generales de la Red ART1

La estructura y funcionamiento de la red ART1 que se describen a continuación fueron tomados de [24].

En la figura 2.1 se muestra la estructura típica de una red ART1. Un vector binario de entrada \mathbf{I} genera un patrón idéntico \mathbf{X} en el nivel F_1 . El vector de salida \mathbf{S} de F_1 se transmite a través de un filtro adaptable conformado por los pesos que emanan de F_1 ; es decir, el vector \mathbf{S} es multiplicado por la matriz de pesos adaptables ascendentes, o memoria a largo plazo (LTM1), para generar un vector de entrada \mathbf{T} al nivel F_2 . La dinámica competitiva

menor cantidad de ejemplares de entrada. A medida que $\rho \rightarrow 1$, el aprendizaje de prototipos se reduce a un aprendizaje de ejemplares.

Si la comparación entre el prototipo \mathbf{V} y el ejemplar \mathbf{I} no satisface el criterio de similitud, la hipótesis definida se descarta y se inicia una nueva búsqueda para encontrar otra categoría en F_2 cuyo prototipo represente de mejor manera a la entrada \mathbf{I} . Un subsistema de orientación regula el proceso de la nueva búsqueda (figura 2.1). Este proceso reinicia el vector \mathbf{Y} descartando la categoría elegida para después definir una nueva categoría cuyo prototipo sea lo suficientemente similar a la entrada \mathbf{I} como para satisfacer el criterio de similitud. Si \mathbf{I} es demasiado diferente de cualquier prototipo aprendido previamente, entonces se selecciona un nodo no comprometido de F_2 para generar una nueva categoría con el ejemplar \mathbf{I} en cuestión como prototipo.

El aprendizaje en línea de un sistema ART1 como el de la figura 2.1 procede de forma estable, con las entradas familiares activando directamente las categorías correspondientes, mientras que las entradas no reconocidas por ninguna categoría definida anteriormente continúan iniciando búsquedas adaptables de mejores categorías, hasta que se agote la capacidad de memoria de la red.

2.2.2 Algoritmo de la Red ART1

En el procedimiento descrito en la Sección 2.2.1 se puede observar lo siguiente:

1. $\mathbf{S} = \mathbf{X} = \mathbf{I}$,
2. $\mathbf{T} = \text{LTM1 } \mathbf{S}$,
3. $\mathbf{U} = \mathbf{Y} = \{0, 0, \dots, 1, 0, \dots, 0\}$, donde el valor de 1 corresponde al nodo ganador en F_2 (es decir, al elemento con mayor valor en \mathbf{T}),
4. $\mathbf{V} = \text{LTM2 } \mathbf{U}$ y
5. $\mathbf{X}^* = \mathbf{I} \cap \mathbf{V}$.

Por consiguiente, se puede desarrollar una versión simplificada y compacta de de la red ART1 sin omitir ninguna de las características de la teoría de resonancia adaptable.

En la figura 2.2 se muestra una representación esquemática de la red ART1. Cada vector de entrada \mathbf{x} tiene m componentes binarios. Los pesos se denotan por w , utilizando tanto en los ascendentes, de x_i a y_j , como en los descendentes, de y_j a x_i , primero el subíndice correspondiente al nodo de la capa superior y enseguida el subíndice correspondiente al nodo de la capa inferior, esto por razones de simetría entre los pesos que unen dos determinados nodos. Los pesos ascendentes se muestran testados debido a que son las normalizaciones de los pesos descendentes.

Antes de presentar el algoritmo que caracteriza a la red ART1, es conveniente definir lo siguiente:

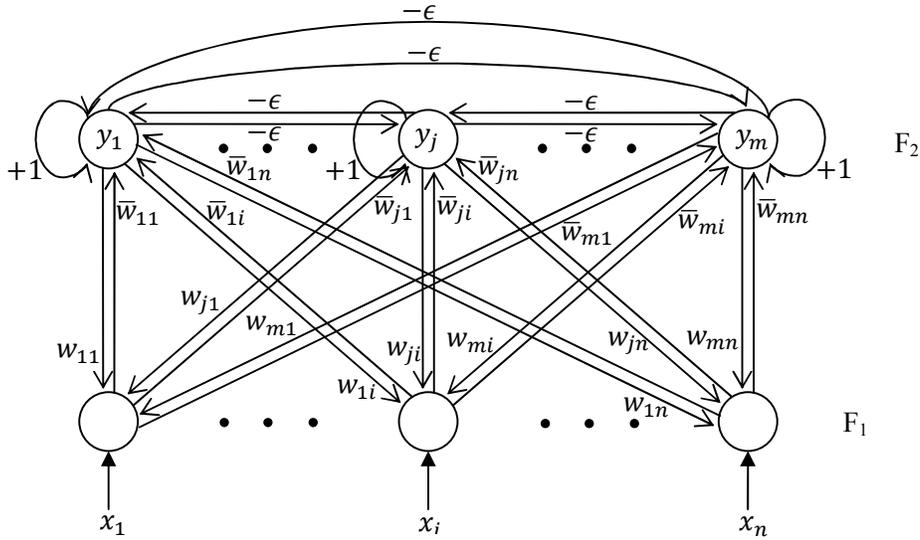


Figura 2.2. Representación esquemática de una red neuronal ART1 [17].

Entrada: conjunto de vectores patrón \mathbf{x} a ser categorizados, donde $\mathbf{x} \in \{0, 1\}^n$.

Salida: conjunto de vectores de pesos $\mathbf{w}_j = (w_{j1}, w_{j2}, \dots, w_{jn})^T$, $j = 1, 2, \dots, m$, que representan los vectores prototipo de las categorías descubiertas, donde m es el número de categorías encontradas.

El algoritmo que a continuación se presenta es una versión resumida tomada y adaptada de [17].

INICIO

1. Inicialización: Ajustar $w_{ji}(0) = 1$, para $0 < \rho \leq 1$.
2. Presentar un nuevo patrón \mathbf{x} a los nodos de entrada.
3. Habilitar todos los nodos de salida.
4. Determinar el valor de activación de cada nodo de salida mediante

$$y_j = \mathbf{x}^T \bar{\mathbf{w}}_j = (\bar{\mathbf{w}}_j)^T \mathbf{x} = \sum_{i=1}^n \bar{w}_{ji} x_i, \tag{2.1}$$

donde \bar{w}_{ji} es la normalización de w_{ji} dada por

$$\bar{w}_{ji} = \frac{w_{ji}}{\alpha + \sum_i w_{ji}}, \quad i = 1, \dots, n. \tag{2.2}$$

El factor α es un pequeño valor (usualmente $\alpha = 0.5$) que se incluye para romper igualdades entre dos vectores \mathbf{w}_j que contengan todos sus bits en \mathbf{x} , seleccionando el mayor de éstos.

5. Seleccionar el nodo de salida con el mayor valor de activación (ganador) e identificar al vector de pesos correspondiente como \mathbf{w}_{jg} .

6. Verificar que \mathbf{x} realmente pertenece a la j^g –ésima categoría llevando a cabo lo siguiente:

$$\text{Si } \frac{\sum_{i=1}^n w_{j^g i} x_i}{\sum_{i=1}^n x_i} \geq \rho, \quad (2.3)$$

entonces \mathbf{x} pertenece a la j^g –ésima categoría, y dirigirse al paso 7; de otro modo si la capa superior aún tiene nodos habilitados, entonces dirigirse al paso 8; de otro modo, crear un nuevo nodo de salida j^* con sus pesos iniciales ajustados como en el paso 1 y dirigirse al paso 7.

7. Actualizar los pesos de la ya existente j^g –ésima categoría o de la recientemente creada j^* –ésima categoría de acuerdo a

$$w_{ji}(t+1) = w_{ji}(t)x_i, \quad i = 1, 2, \dots, n, \quad (2.4)$$

y dirigirse al paso 2.

8. Deshabilitar el nodo de salida j^g haciendo $y_{j^g} = 0$; por lo que este nodo no participa en la actual búsqueda de categoría. Dirigirse al paso 4 para tratar de establecer una nueva categoría diferente de j^g para el patrón \mathbf{x} .

FIN

En el algoritmo anterior se puede observar que para un vector de entrada \mathbf{x} , es posible finalizar de dos formas. Primera, si se encuentra un prototipo \mathbf{w}_{j^g} que cumpla con la condición de similitud de la ecuación (2.3), entonces éste es ajustado en el paso 7 de acuerdo a la ecuación (2.4). Segunda, si no se encuentra un prototipo adecuado entre las categorías almacenadas, entonces se crea un nuevo nodo de salida j^* que representa una nueva categoría con el vector de entrada \mathbf{x} como prototipo. En la práctica, comúnmente sólo se cuenta con un número finito de nodos de salida disponibles; en consecuencia, una vez que todos los nodos de salida disponibles se han agotado y aún no ha sido posible encontrar un prototipo que cumpla con la condición de similitud para un determinado patrón de entrada, se concluye el ciclo sin salida.

La regla de adaptación de la ecuación (2.4) elimina los bits de $\mathbf{w}_j(t)$ que no están en \mathbf{x} ; esto es equivalente a la operación lógica AND. Debido a que esta regla de adaptación únicamente remueve bits de los vectores prototipo y nunca agrega alguno, estos prototipos nunca regresan a un valor anterior. Esto provoca que los cambios en los pesos disminuyan paulatinamente hasta ser nulos después de un número finito de presentaciones de cualquier conjunto fijo de entradas; con lo que se asegura la estabilidad del algoritmo. Por otra parte, la capacidad que tiene la ART1 de incrementar los nodos de salida le provee de plasticidad, mientras no se hayan ocupado todos estos nodos. Esto explica también el por qué la red ART1 puede resolver el dilema de estabilidad-plasticidad.

2.2.3 Difusificación de la Red ART1

La red ART1 aprende de manera no supervisada y estable a categorizar patrones binarios de entrada; es decir, sólo clasifica ejemplares binarios de entrada en base a las características de éstos. Los modelos ART2 y ART2-A superan esta restricción, dado que son capaces de trabajar con patrones de entrada tanto binarios como analógicos; sin embargo, estos modelos son considerablemente más complejos en su estructura y funcionamiento. Por otra parte, resulta evidente que las operaciones contenidas en el algoritmo de la red ART1 pertenecen a la teoría de conjuntos clásicos; entonces, si estas operaciones se sustituyen por las correspondientes en la teoría de conjuntos difusos [16], se obtiene un modelo denominado ART difuso (FART por sus siglas en inglés) [24, 25], al cual se le considera una generalización del modelo ART1, puesto que es capaz de trabajar con patrones de entrada tanto binarios como analógicos conservando prácticamente la misma estructura y funcionamiento de la red ART1.

La red FART es el resultado de sustituir las operaciones de intersección de la teoría de conjuntos clásicos que utiliza la red ART1 por las operaciones de conjunción pertenecientes a la teoría de conjuntos difusos [24]. En la tabla 2.1 se resume la forma en que las operaciones de elección de categoría, criterio de similitud y aprendizaje de la red ART1 se convierten en las operaciones de la red FART mediante el reemplazo del operador de intersección (\cap) de la teoría de conjuntos clásicos por la conjunción de la teoría de conjuntos difusos, u operador mín (\wedge). La norma que usualmente se maneja en las redes FART es la definida por $|\mathbf{p}| = \sum_{i=1}^n p_i$ [9, 10].

Tabla 2.1. Comparación entre las operaciones de la red ART1 y la red FART [24].

| ART1 (Binario) | FART (Analógico) |
|---|---|
| Elección de Categoría | |
| $\mathbf{T}_j = \frac{ \mathbf{I} \cap \mathbf{w}_j }{\alpha + \mathbf{w}_j }$ | $\mathbf{T}_j = \frac{ \mathbf{I} \wedge \mathbf{w}_j }{\alpha + \mathbf{w}_j }$ |
| Criterio de Similitud | |
| $\frac{ \mathbf{I} \cap \mathbf{w} }{ \mathbf{I} } \geq \rho$ | $\frac{ \mathbf{I} \wedge \mathbf{w} }{ \mathbf{I} } \geq \rho$ |
| Aprendizaje (Rápido) | |
| $\mathbf{w}_j(t+1) = \mathbf{I} \cap \mathbf{w}_j(t)$ $\cap = \text{AND lógica, intersección}$ | $\mathbf{w}_j(t+1) = \mathbf{I} \wedge \mathbf{w}_j(t)$ $\wedge = \text{AND difusa, mínimo}$ |

El funcionamiento, estructura (figuras 2.1 y 2.2) y algoritmo de la red FART básicamente son los mismos que los descritos anteriormente para la red ART1. Las ecuaciones contenidas en la tabla 2.1 corresponden a las ecuaciones (2.1), (2.3) y (2.4), respectivamente. La ecuación (2.2) se omite en el caso de la red FART, puesto que los pesos ascendentes y descendentes son iguales entre sí. Un punto importante a destacar es

que los elementos de los vectores de entrada a la red FART se deben encontrar en el intervalo $[0, 1]$, esto para que el uso de los operadores difusos de la tabla 2.1 sea adecuado.

2.3 ARQUITECTURA ARTMAP DIFUSA

Al igual que la red FART es una generalización de la red ART1 mediante la incorporación de operadores difusos, la arquitectura FAM es una generalización de otra arquitectura denominada ARTMAP mediante esta misma incorporación. Por consiguiente, es conveniente primero mencionar las características, estructura y funcionamiento de la arquitectura ARTMAP para después abordar el tema de la arquitectura FAM.

2.3.1 Arquitectura ARTMAP

Otra de las limitaciones que presenta la red ART1 es la información que puede proporcionar en un determinado momento, la cual consta únicamente de los prototipos de las categorías definidas hasta ese instante. La arquitectura ARTMAP supera esta limitante, puesto que provee la capacidad de introducir información (en forma de vector) específica relacionada con cada categoría definida. Esta arquitectura es una red neuronal que lleva a cabo un aprendizaje supervisado incremental de categorías de reconocimiento y de mapeos multidimensionales [26, 27]; es decir, es capaz de organizar rápidamente y por sí misma mapeos entre vectores en \mathbb{R}^n a vectores en \mathbb{R}^m mediante un aprendizaje supervisado, por lo que también se le conoce como red ART predictiva.

Los principales elementos de un sistema ARTMAP se muestran en la figura 2.3. Dos redes, ART_a y ART_b , leen los vectores binarios de entrada \mathbf{a} y \mathbf{b} , respectivamente. Estas redes están conectadas por una tercera denominada *campo de mapeo*, que controla el aprendizaje de la asociación de los mapeos entre las categorías reconocidas en ambas redes ART [27]. La conexión entre los nodos de salida de la red ART_a y los nodos del campo de mapeo es completa; es decir, cada nodo de salida de la red ART_a está conectado a cada uno de los nodos del campo de mapeo. Cada uno de los nodos de salida de la red ART_b está conectado solamente a un nodo del campo de mapeo, por lo que el número de nodos del campo de mapeo es igual al número de nodos de salida de la red ART_b .

Los mapeos no asocian directamente ejemplares \mathbf{a} y \mathbf{b} , sino representaciones simbólicas y comprimidas de las familias de estos ejemplares. El campo de mapeo también controla el *seguimiento de similitud* del parámetro de vigilancia de la red ART_a . Una asociación errónea en el campo de mapeo entre la categoría activada en la red ART_a por una entrada \mathbf{a} y la categoría activada en la red ART_b por una entrada \mathbf{b} incrementa el parámetro de vigilancia de la red ART_a en un valor mínimo necesario para que el sistema busque y, si así se requiere, aprenda una nueva categoría en la red ART_a que proporcione una correcta asociación con la categoría de la red ART_b en cuestión.

En la etapa de producción, el vector \mathbf{b} se convierte en la salida de la arquitectura ARTMAP dado un vector de entrada \mathbf{a} . Durante esta etapa, la red ART_a realiza la misma función que en la etapa de aprendizaje, mientras que la red ART_b trabaja de forma inversa. De este modo, un vector de entrada \mathbf{a} activa una determinada categoría en la red ART_a , la

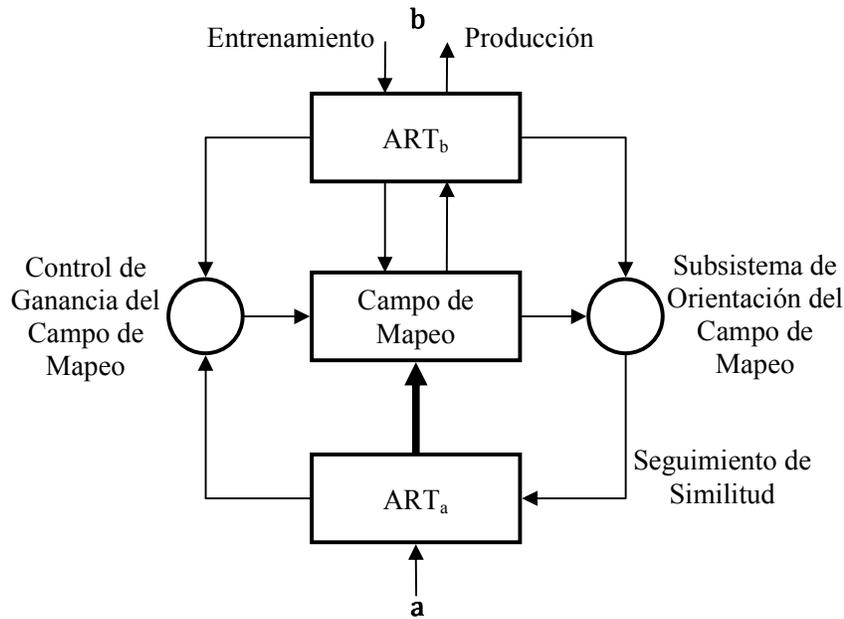


Figura 2.3. Diagrama de bloques de un sistema ARTMAP [27].

cual, a su vez, activa la correspondiente categoría en la red ART_b mediante las asociaciones de los mapeos aprendidos en el campo de mapeo. El vector correspondiente a la categoría activada en la red ART_b se convierte en el vector de salida \mathbf{b} de la arquitectura ARTMAP.

2.3.2 Difusificación de la Arquitectura ARTMAP

La arquitectura ARTMAP, puesto que está compuesta principalmente por dos redes ART1, sólo es capaz de trabajar con vectores binarios. La difusificación de esta arquitectura permite trabajar con vectores tanto binarios como analógicos, lo que da origen a la arquitectura ARTMAP difusa o FAM. Esta difusificación se logra de manera muy similar al caso de la red ART1; es decir, por medio de la sustitución de operaciones de la tabla 2.1. Evidentemente, esto es igual a sustituir las redes ART1 por redes ART difusas en la arquitectura ARTMAP.

En la figura 2.4 se muestra la estructura de la arquitectura FAM, la cual consta de dos redes FART, $FART_a$ y $FART_b$, conectadas por un campo de mapeo, de forma similar a la estructura de la arquitectura ARTMAP (figura 2.3) [9]. La red $FART_a$ contiene la capa de entrada F_1^a y la capa competitiva F_2^a . También se añade una capa de pre procesamiento F_0^a por razones que se verán más adelante en la Sección 2.3.4. M_a es el número de nodos de F_1^a , N_a es el número de nodos de F_2^a y \mathbf{W}^a es la matriz de pesos entre F_1^a y F_2^a . Análogamente, esta notación se aplica en la red $FART_b$.

El campo de mapeo consta de la capa F^{ab} , cuyo número de nodos es igual al número de nodos de F_2^b . Al igual que en la arquitectura ARTMAP, la conexión entre los nodos de F_2^a y los nodos de F^{ab} es completa y la conexión entre los nodos de F_2^b y los nodos de F^{ab} es uno a uno.

El funcionamiento de la arquitectura FAM se resume en el siguiente algoritmo.

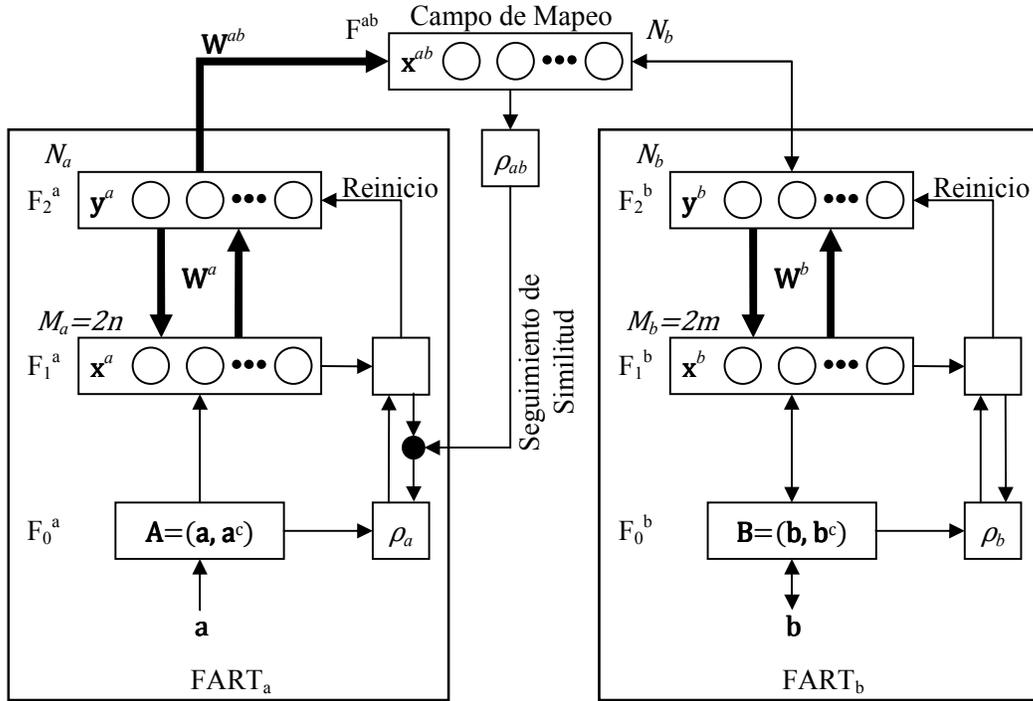


Figura 2.4. Arquitectura ARTMAP difusa (FAM) [9].

2.3.3 Algoritmo de la Arquitectura ARTMAP Difusa

El siguiente algoritmo es una versión tomada y adaptada de [10].

INICIO

1. Inicializar todos los pesos con un valor igual a la unidad.
2. Ajustar el parámetro de vigilancia de la red $FART_a$ a un valor base, $\rho_a = \bar{\rho}_a$; ajustar el parámetro de vigilancia de la red $FART_b$ a un valor fijo, y considerar que ningún nodo se encuentra inhibido (es decir, todos los nodos participan en la búsqueda de una categoría apropiada para el patrón de entrada actual).
3. Para cada entrada \mathbf{B} ($\mathbf{B} \in [0, 1]^m$) en la capa F_0^b , se usa la siguiente función de elección difusa para obtener la respuesta de cada categoría en F_2^b :

$$T_k(\mathbf{B}) = \frac{|\mathbf{B} \wedge \mathbf{w}_k^b|}{\alpha_b + |\mathbf{w}_k^b|}, \quad k = 1, 2, \dots, N_b, \quad (2.5)$$

donde $\alpha_b > 0$ es un parámetro de elección. Si $0 < \alpha_b \ll 1$, $T_k(\mathbf{B})$ mide el grado en que \mathbf{B} es un subconjunto difuso de \mathbf{w}_k^b .

4. Identificar como K al índice correspondiente al mayor valor calculado en la ecuación (2.5) por medio de

$$K = \text{ind}_{k=1, \dots, N_b} \max \{T_k\}. \quad (2.6)$$

5. Verificar si se cumple la condición de resonancia; es decir, si la entrada es lo suficientemente similar al prototipo ganador, mediante

$$\frac{|\mathbf{B} \wedge \mathbf{w}_K^b|}{|\mathbf{B}|} \geq \rho_b. \quad (2.7)$$

Si esta condición se cumple (si existe resonancia), entonces dirigirse al paso 6; de otro modo, inhibir el K -ésimo nodo de tal manera que no participe en las competencias consecuentes para el patrón actual. Si aún existen nodos desinhibidos, dirigirse al paso 4; de no ser así, crear una nueva categoría que represente al patrón de entrada y definir como K al índice del nuevo nodo.

6. Aplicar los pasos 3, 4 y 5 a la red $FART_a$ de tal manera que J sea el índice del nodo ganador.
7. El vector de salida de la capa F_2^b es ajustado de acuerdo a

$$y_k^b = \begin{cases} 1, & \text{si } k = K \\ 0, & \text{si } k \neq K \end{cases}, \quad k = 1, \dots, N_b. \quad (2.8)$$

En el campo de mapeo se forma un vector de salida \mathbf{x}^{ab} definido por

$$\mathbf{x}^{ab} = \mathbf{y}^b \wedge \mathbf{w}_J^{ab}, \quad (2.9)$$

debido a que ambas capas, F_2^a y F_2^b , están activas. En la fase de producción, únicamente la capa F_2^a se encuentra activa, por lo que en este caso el vector \mathbf{x}^{ab} queda definido por

$$\mathbf{x}^{ab} = \mathbf{w}_J^{ab}, \quad (2.10)$$

8. En el campo de mapeo se realiza una prueba de similitud entre el vector predicho \mathbf{x}^{ab} y el vector objetivo \mathbf{y}^b por medio de

$$\frac{|\mathbf{x}^{ab}|}{|\mathbf{y}^b|} \geq \rho_{ab}, \quad (2.11)$$

donde $\rho_{ab} \in [0, 1]$ es el parámetro de vigilancia del campo de mapeo. Si se cumple la prueba de la ecuación (2.11), entonces se lleva a cabo el aprendizaje en las redes $FART_a$, $FART_b$ y en el campo de mapeo (paso 9); de otro modo, se inicia el seguimiento de similitud (paso 10).

9. Realizar el proceso de aprendizaje en ambas redes FART y en el campo de mapeo. Para la red $FART_b$ se tiene

$$\mathbf{w}_K^b(t+1) = \beta_b [\mathbf{B} \wedge \mathbf{w}_K^b(t)] + (1 - \beta_b) \mathbf{w}_K^b(t), \quad (2.12)$$

para la red $FART_a$ se aplica análogamente la ecuación (2.12), y para el campo de mapeo se tiene

$$\mathbf{w}_{Jk}^{ab} = \begin{cases} 1, & \text{si } k = K \\ 0, & \text{si } k \neq K \end{cases}. \quad (2.13)$$

Y dirigirse al paso 11.

10. Seguimiento de similitud: incrementar ρ_a de acuerdo a

$$\rho_a = \frac{|\mathbf{A} \wedge \mathbf{w}_j^a|}{|\mathbf{A}|} + \delta, \quad (2.14)$$

donde $0 < \delta < 1$. Si $\rho_a > 1$, entonces se rechaza la entrada actual; si no es así, dirigirse al paso 6.

11. Si se van a procesar más patrones, dirigirse al paso 2; de otro modo, detener el proceso.

FIN

En el algoritmo anterior se puede apreciar que los pasos 1-5 y la ecuación (2.12) del paso 9 constituyen el algoritmo de una red FART.

Existen algunos análisis que conviene considerar en la aplicación de las redes neurodifusas creadas a partir de la teoría de resonancia adaptable, debido en gran medida a la naturaleza analógica de los patrones a procesar y a los operadores difusos relacionados. A continuación se presentan los análisis más trascendentes relacionados con las redes FART y, por consiguiente, con la arquitectura FAM. Para ello se utiliza la notación de la red FART_b de la figura 2.4 al igual que en el algoritmo de la arquitectura FAM; en el caso de la red FART_a, estos análisis se aplican de la misma manera.

2.3.4 Normalización de los Vectores de Entrada

B. Moore (1989) describió un problema de proliferación de categorías que puede ocurrir en algunos sistemas ART análogos cuando un gran número de patrones de entrada erosionan la norma de los vectores de pesos. Esta proliferación de categorías se puede evitar normalizando las entradas; esto es, para algún $\gamma > 0$,

$$|\mathbf{B}| \equiv \gamma, \quad (2.15)$$

para todas las entradas \mathbf{B} [24]. Esta normalización se puede lograr mediante el pre procesamiento de cada vector de entrada \mathbf{b} ; por ejemplo, haciendo que

$$\mathbf{B} = \frac{\mathbf{b}}{|\mathbf{b}|}. \quad (2.16)$$

Una regla alternativa, denominada *codificación con complemento* [24], logra la normalización mientras preserva información significativa. Esta regla contempla las características tanto presentes como ausentes en los patrones de entrada. Para definir esta regla en su forma más simple, se considera que el vector \mathbf{b} representa en sí las características presentes. El complemento de \mathbf{b} , que se denota por \mathbf{b}^c , representa las características ausentes, donde

$$b_i^c = 1 - b_i. \quad (2.17)$$

La entrada \mathbf{B} codificada con complemento duplica su dimensión y queda como

$$\mathbf{B} = (\mathbf{b}, \mathbf{b}^c) = (b_1, \dots, b_{\dim \mathbf{B}}, b_1^c, \dots, b_{\dim \mathbf{B}}^c). \quad (2.18)$$

La norma de \mathbf{B} es

$$|\mathbf{B}| = |(\mathbf{b}, \mathbf{b}^c)| = \sum_{i=1}^{\dim \mathbf{B}} b_i + \sum_{i=1}^{\dim \mathbf{B}} (1-b_i) = \sum_{i=1}^{\dim \mathbf{B}} b_i - \sum_{i=1}^{\dim \mathbf{B}} b_i + \dim \mathbf{B} = \dim \mathbf{B}; \quad (2.19)$$

así, las entradas pre procesadas mediante la codificación con complemento se normalizan automáticamente. De este modo, las entradas de la arquitectura FAM codificadas con complemento quedan definidas por

$$\begin{aligned} \mathbf{A} &= (\mathbf{a}, \mathbf{a}^c) = (a_1, \dots, a_n, 1-a_1, \dots, 1-a_n) \\ \mathbf{B} &= (\mathbf{b}, \mathbf{b}^c) = (b_1, \dots, b_m, 1-b_1, \dots, 1-b_m) \end{aligned} \quad (2.20)$$

y sus respectivas normas son

$$\begin{aligned} |\mathbf{A}| &= n \\ |\mathbf{B}| &= m \end{aligned} \quad (2.21)$$

Por lo tanto, el número de nodos en la capa F_1^a es $M_a = 2n$ y el número de nodos en la capa F_1^b es $M_b = 2m$, tal y como lo muestra la figura 2.4.

2.3.5 Aprendizaje Rápido y Recodificación Lenta

En muchas aplicaciones es muy útil combinar un aprendizaje inicial rápido con un factor de olvido menor. A esta combinación se le conoce como opción de compromiso rápido y recodificación lenta [24]. Un nodo no comprometido es aquel que aún no ha sido elegido para representar una categoría; esto es, los pesos del vector prototipo que representa este nodo son aún los iniciales (con valor de 1); mientras que un nodo comprometido es aquel que ya ha sido elegido para representar una categoría en específico, por lo que sus pesos ya han experimentado el proceso de aprendizaje.

El aprendizaje rápido proporciona una adecuada respuesta a entradas que se presentan de forma esporádica y a entradas que demandan un funcionamiento rápido y preciso. La recodificación lenta previene que la información ya incorporada con anterioridad en los prototipos de las categorías sea erróneamente eliminada en respuesta a ruido o entradas parciales. En un sistema con recodificación lenta, sólo un cambio estadísticamente persistente de una característica relevante contenida en una determinada categoría puede eliminar tal característica del prototipo correspondiente.

La opción de compromiso (o aprendizaje) rápido y recodificación lenta indica que β sea igual a la unidad cuando el nodo elegido sea no comprometido, y $\beta < 1$ cuando el nodo elegido sea comprometido. De esta manera, la primera vez que se activa una determinada categoría, la ecuación (2.12) se convierte en

$$\mathbf{w}_K^b(t+1) = \mathbf{B} \wedge \mathbf{w}_K^b(t), \quad (2.22)$$

que es la ecuación de aprendizaje de la tabla 2.1, con lo que el prototipo correspondiente es igual al ejemplar en turno.

2.3.6 Elección de Subconjunto Difuso en el Límite Conservativo

Si $\alpha_b \rightarrow 0$ en la ecuación (2.5), la función de elección T_k refleja el grado en que el vector de pesos \mathbf{w}_k^b es un subconjunto difuso del vector de entrada \mathbf{B} . Si

$$\frac{|\mathbf{B} \wedge \mathbf{w}_k^b|}{|\mathbf{w}_k^b|} = 1, \quad (2.23)$$

entonces \mathbf{w}_k^b es un subconjunto difuso de \mathbf{B} , y se dice que la categoría k es una *elección de subconjunto difuso* para la entrada \mathbf{B} [24]. En este caso, si la categoría k es seleccionada y si se contempla el aprendizaje rápido, no ocurre recodificación en la ecuación (2.12), dado que $\mathbf{B} \wedge \mathbf{w}_k^b = \mathbf{w}_k^b$. Por lo anterior, al límite cuando $\alpha_b \rightarrow 0$ se le conoce como *límite conservativo*.

La resonancia depende en el grado en que \mathbf{B} es un subconjunto difuso de \mathbf{w}_k^b (ecuación (2.7)). En particular, si la categoría k es una elección de subconjunto difuso, entonces el valor de la función de la ecuación (2.7) queda como

$$\frac{|\mathbf{B} \wedge \mathbf{w}_k^b|}{|\mathbf{B}|} = \frac{|\mathbf{w}_k^b|}{|\mathbf{B}|}. \quad (2.24)$$

En consecuencia, la elección de K para maximizar a $|\mathbf{w}_k^b|$ de entre las elecciones de subconjunto difuso también maximiza la posibilidad de que ocurra resonancia en la ecuación (2.7). Si no ocurre tal resonancia para el nodo que maximiza a $|\mathbf{w}_k^b|$, entonces tampoco ocurrirá para todas las demás elecciones de subconjunto difuso.

2.3.7 Interpretación Geométrica de las Redes ART Difusas

Un aspecto fundamental a considerar en el reconocimiento y categorización de patrones es la estabilidad del aprendizaje con que se realiza tal labor. Para analizar la estabilidad del aprendizaje de las redes FART con codificación con complemento es conveniente primero desarrollar su interpretación geométrica. Por simplicidad, considérese un conjunto de vectores de entrada bidimensionales \mathbf{b} , entonces la entrada a la red FART_b es

$$\mathbf{B} = (\mathbf{b}, \mathbf{b}^c) = (b_1, b_2, 1-b_1, 1-b_2). \quad (2.25)$$

En este caso, cada categoría k tiene una interpretación geométrica dada por un rectángulo R_k , como se describe a continuación. De acuerdo con la ecuación (2.25), el vector de pesos \mathbf{w}_k^b en forma de codificación con complemento se puede escribir como

$$\mathbf{w}_k^b = (\mathbf{u}_k^b, (\mathbf{v}_k^b)^c), \quad (2.26)$$

donde \mathbf{u}_k^b y \mathbf{v}_k^b son vectores bidimensionales. Si se define que el vector \mathbf{u}_k^b representa una esquina de un rectángulo R_k y que el vector \mathbf{v}_k^b representa otra esquina del mismo rectángulo. El tamaño de R_k se define como

$$|R_k| = |\mathbf{v}_k^b - \mathbf{u}_k^b|, \quad (2.27)$$

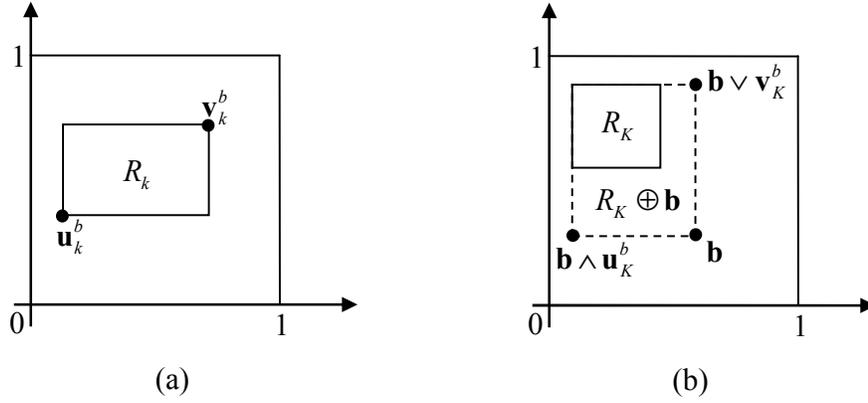


Figura 2.5. (a) Interpretación Geométrica de un vector de pesos \mathbf{w}_k^b con codificación con complemento para un vector de entrada bidimensional [24]. (b) Expansión durante el entrenamiento de R_k a $R_k \oplus \mathbf{b}$ con aprendizaje rápido. [24].

$$|R_k| = |\mathbf{v}_k^b - \mathbf{u}_k^b|, \quad (2.27)$$

el cual es igual a la altura más el ancho de R_k en la figura 2.5a.

En la red FART_b con aprendizaje rápido ($\beta = 1$ en la ecuación (2.12)), $\mathbf{w}_K^b(t+1) = \mathbf{B} = (\mathbf{b}, \mathbf{b}^c)$ cuando K es un nodo no comprometido. Las esquinas de $R_K(t+1)$ entonces están dadas por \mathbf{b} y por $(\mathbf{b}^c)^c = \mathbf{b}$; por lo tanto, $R_K(t+1)$ sólo es el punto \mathbf{b} . El aprendizaje incrementa el tamaño de cada R_k ; de hecho, el tamaño de R_k crece a medida que el tamaño de \mathbf{w}_k^b decrece durante el aprendizaje, y el tamaño máximo de R_k es determinado por el parámetro de vigilancia ρ_b , como se muestra enseguida. Durante cada proceso de entrenamiento rápido, R_K se expande a $R_K \oplus \mathbf{b}$, el mínimo rectángulo que contiene a R_K y \mathbf{b} (figura 2.5b). Las esquinas de $R_K \oplus \mathbf{b}$ están dadas por $\mathbf{b} \wedge \mathbf{u}_K^b$ y $\mathbf{b} \vee \mathbf{v}_K^b$, donde $(\mathbf{x} \vee \mathbf{y})_i = \max(x_i, y_i)$. Entonces, de acuerdo con la ecuación (2.27), el tamaño de $R_K \oplus \mathbf{b}$ está dado por

$$|R_K \oplus \mathbf{b}| = |(\mathbf{b} \vee \mathbf{v}_K^b) - (\mathbf{b} \wedge \mathbf{u}_K^b)|. \quad (2.28)$$

Sin embargo, el reinicio conduce a una nueva elección de categoría si $|R_K \oplus \mathbf{b}|$ es demasiado grande. Estas propiedades se prueban a continuación.

Supóngase que $\mathbf{B} = (\mathbf{b}, \mathbf{b}^c)$ elige una categoría K por medio de la ecuación (2.6). El vector de pesos \mathbf{w}_K^b se actualiza de acuerdo a la ecuación (2.22) de aprendizaje rápido sólo si se satisface el criterio de similitud de la ecuación (2.7). Debido a la codificación con complemento, $|\mathbf{B}| = m$ (ecuación (2.21)). Por tanto, cuando $m = 2$, se satisface el criterio de similitud de la ecuación (2.7) si y sólo si

$$|\mathbf{B} \wedge \mathbf{w}_K^b| \geq 2\rho_b. \quad (2.29)$$

No obstante,

$$|\mathbf{B} \wedge \mathbf{w}_K^b| = |(\mathbf{b}, \mathbf{b}^c) \wedge (\mathbf{u}_K^b, (\mathbf{v}_K^b)^c)| = |(\mathbf{b} \wedge \mathbf{u}_K^b), (\mathbf{b}^c \wedge (\mathbf{v}_K^b)^c)|$$

$$\begin{aligned}
&= \left| (\mathbf{b} \wedge \mathbf{u}_k^b), (\mathbf{b} \vee \mathbf{v}_k^b)^c \right| = |\mathbf{b} \wedge \mathbf{u}_k^b| + 2 - |\mathbf{b} \vee \mathbf{v}_k^b| \\
&\therefore |\mathbf{B} \wedge \mathbf{w}_k^b| = 2 - |R_k \oplus \mathbf{b}|,
\end{aligned} \tag{2.30}$$

por la ecuación (2.28). Así, de acuerdo con las ecuaciones (2.29) y (2.30), el criterio de similitud se cumple si y sólo si el rectángulo expandido $R_k \oplus \mathbf{b}$ satisface

$$|R_k \oplus \mathbf{b}| \leq 2(1 - \rho_b). \tag{2.31}$$

En la ecuación (2.31) se aprecia que si el parámetro de vigilancia es cercano a 1, entonces todos los R_k 's son pequeños. Si ρ_b es cercano a 0, entonces es probable que algunos R_k 's crezcan para abarcar el cuadrado unitario $[0, 1] \times [0, 1]$ casi en su totalidad.

Ahora, supóngase que se cumple el criterio de similitud. Entonces, de acuerdo a la ecuación (2.22) se tiene

$$\begin{aligned}
\mathbf{w}_k^b(t+1) &= \mathbf{B} \wedge \mathbf{w}_k^b(t) = (\mathbf{b}, \mathbf{b}^c) \wedge (\mathbf{u}_k^b(t), (\mathbf{v}_k^b(t))^c) \\
&= (\mathbf{b} \wedge \mathbf{u}_k^b(t), \mathbf{b}^c \wedge (\mathbf{v}_k^b(t))^c) = (\mathbf{b} \wedge \mathbf{u}_k^b(t), (\mathbf{b} \vee \mathbf{v}_k^b(t))^c) \\
&\therefore \mathbf{w}_k^b(t+1) = (\mathbf{u}_k^b(t+1), (\mathbf{v}_k^b(t+1))^c).
\end{aligned} \tag{2.32}$$

Por lo que

$$R_k(t+1) = R_k(t) \oplus \mathbf{b}. \tag{2.33}$$

En particular, no ocurre cambio en los pesos si $\mathbf{b} \in R_k(t)$. En resumen, con aprendizaje rápido, cada R_k es igual al rectángulo más pequeño que incluya todos los vectores \mathbf{b} que activen la categoría k bajo la restricción de $|R_k| \leq 2(1 - \rho_b)$.

En general, si \mathbf{b} es de dimensión m , el hiperrectángulo R_k incluye los dos vértices $\wedge_k \mathbf{b}$ y $\vee_k \mathbf{b}$, donde el i -ésimo componente de cada vector es

$$(\wedge_k \mathbf{b})_i = \min\{b_i : \mathbf{b} \text{ ha sido codificado por la categoría } k\} \tag{2.34}$$

y

$$(\vee_k \mathbf{b})_i = \max\{b_i : \mathbf{b} \text{ ha sido codificado por la categoría } k\}, \tag{2.35}$$

como se muestra en la figura 2.6. El tamaño de R_k está dado por

$$|R_k| = |\vee_k \mathbf{b} - \wedge_k \mathbf{b}|. \tag{2.36}$$

Similarmente que en la ecuación (2.32), se tiene que

$$\mathbf{w}_k^b = (\wedge_k \mathbf{b}, (\vee_k \mathbf{b})^c), \tag{2.37}$$

por lo que

$$|\mathbf{w}_k^b| = \sum_i (\wedge_k \mathbf{b})_i + \sum_i [1 - (\vee_k \mathbf{b})_i] = m - |\vee_k \mathbf{b} - \wedge_k \mathbf{b}|. \tag{2.38}$$

Entonces, el tamaño del hiperrectángulo R_k es

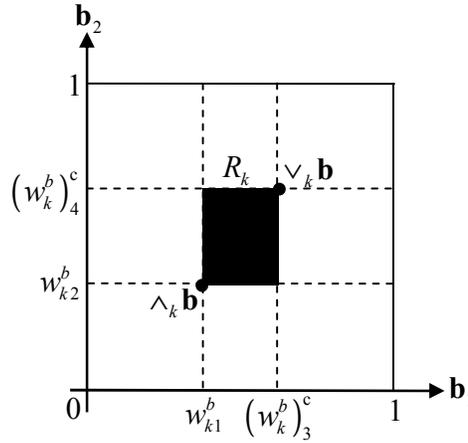


Figura 2.6. Rectángulo R_k con aprendizaje rápido y codificación con complemento [24].

$$|R_k| = m - |\mathbf{w}_k^b|. \tag{2.39}$$

De este modo, al igual que en la ecuación (2.29), el criterio de similitud se satisface si y sólo si

$$|\mathbf{w}_k^b| \geq m\rho_b. \tag{2.40}$$

De acuerdo con las ecuaciones (2.39) y (2.40), se tiene que el tamaño de R_k debe ser

$$|R_k| \leq m(1 - \rho_b). \tag{2.41}$$

Consecuentemente, en el caso m -dimensional, un parámetro de vigilancia alto ($\rho_b \cong 1$) también conduce a pequeños R_k 's, mientras que un parámetro de vigilancia bajo ($\rho_b \cong 0$) permite R_k 's grandes. Si k es un nodo no comprometido, entonces $|\mathbf{w}_k^b| = 2m$, debido a que todos los elementos de los pesos se inicializan con un valor igual a 1; así, $|R_k| = -m$, de acuerdo con la ecuación (2.39). Estas observaciones se pueden combinar en el siguiente teorema.

2.3.8 Teorema de Estabilidad en el Aprendizaje de Categorías

“En respuesta a una secuencia arbitraria de vectores de entrada binarios o analógicos, un sistema ART difuso con codificación con complemento y aprendizaje rápido define categorías estables en forma de hiperrectángulos R_k , las cuales crecen durante el aprendizaje hasta un tamaño máximo $|R_k| \leq m(1 - \rho_b)$ a medida que $|\mathbf{w}_k^b|$ decrece monótonamente. En el límite conservativo, se suscita lo que se conoce como *aprendizaje de un paso* de tal manera que no ocurren reinicios o aprendizajes adicionales en subsecuentes presentaciones de cualquier entrada” [24].

En el caso del aprendizaje rápido y recodificación lenta, se tienen propiedades similares, excepto que se pueden necesitar presentaciones repetidas de un conjunto de entradas antes de que ocurra la estabilización.

B. Moore (1989) describió una variedad de algoritmos para el reconocimiento y categorización de patrones, algunos de ellos clásicos y otros basados en la red ART1, similares a la red FART. Sin embargo, en todos ellos se usa una función de elección que incluye una medida del producto interno o de la distancia euclidiana, lo cual difiere de la función de elección T_k en la ecuación (2.5). Además, no utilizan la codificación con complemento. Por ejemplo, el algoritmo de *categorización euclidiana* (B. Moore, 1989, p. 176) elige la categoría K cuyo vector de pesos \mathbf{w}_K^b ofrece la distancia euclidiana mínima $d(\mathbf{w}_K^b, \mathbf{B})$ a \mathbf{B} , y el cual satisface

$$d(\mathbf{w}_K^b, \mathbf{B}) \leq 0. \quad (2.42)$$

Si tal categoría existe, entonces \mathbf{w}_K^b es actualizado por medio de

$$\mathbf{w}_K^b(t+1) = \beta \mathbf{B} + (1 - \beta) \mathbf{w}_K^b(t); \quad (2.43)$$

de lo contrario, se elige un nodo no comprometido K y $\mathbf{w}_K^b(t+1) = \mathbf{B}$, como en la opción de compromiso rápido. El algoritmo de *categorización unidireccional* (B. Moore, 1989, p. 177) es similar, excepto que los pesos se actualizan de acuerdo a la ecuación (2.12). Moore destacó que el algoritmo de categorización euclidiana es inestable en el sentido de que los vectores de pesos y las fronteras de las categorías pueden variar cíclica e interminablemente. Moore mostró también que la regla unidireccional de actualización de pesos de la ecuación (2.12) evita este tipo de inestabilidad, pero introduce el problema de proliferación de categorías si no se contempla el pre procesamiento de las entradas mediante la codificación con complemento.

La normalización de las entradas usando la codificación con complemento provee a los sistemas ART difusos de la capacidad de superar el problema de la proliferación de categorías a la vez que mantienen las propiedades de codificación estable de la regla de actualización de pesos de la ecuación (2.12), tal y como se menciona en el teorema anterior. La gran estabilidad y rápida convergencia que caracterizan a los modelos ART difusos se deben a la directa relación entre la función de elección de la ecuación (2.5), la regla de reinicio y la regla de actualización de pesos de la ecuación (2.12). La elección, búsqueda y aprendizaje son computacionalmente consistentes debido al uso común del vector $\mathbf{B} \wedge \mathbf{w}_K^b$.

CAPÍTULO 3

APLICACIÓN DE LA ARQUITECTURA ARTMAP DIFUSA EN LA REGULACIÓN DE VOLTAJE DE UN GENERADOR SÍNCRONO

3.1 INTRODUCCIÓN

La habilidad de llevar a cabo un aprendizaje supervisado incremental de categorías de reconocimiento estables para el mapeo multidimensional entre patrones analógicos de entrada y salida que caracteriza a la arquitectura FAM, despierta el interés de aplicar esta arquitectura en el control de máquinas y procesos eléctricos.

En este capítulo se describe la manera en que la arquitectura FAM puede ser utilizada particularmente en la regulación del voltaje en terminales de un generador síncrono que se encuentra conectado a un bus infinito. El entrenamiento necesario para tal fin involucra la definición de una serie de factores que afectan directamente el funcionamiento de esta arquitectura en la fase de producción. Por lo tanto, primero se definen los factores con los que la arquitectura FAM proporciona mejores resultados actuando como regulador de voltaje del sistema bajo prueba, después se analizan tales resultados y, por último, se explora una técnica para utilizar la arquitectura FAM para sintonizar un controlador difuso con la finalidad de obtener una regulación de voltaje más eficiente en diferentes condiciones de operación.

El modelo y los datos del sistema máquina bus infinito que se utilizan en este trabajo se presentan en el Apéndice A.

3.2 DEFINICIÓN DE LOS FACTORES NECESARIOS PARA EL ENTRENAMIENTO DE LA ARQUITECTURA ARTMAP DIFUSA

Los factores a considerar en el desarrollo del entrenamiento de la arquitectura FAM son: el propio esquema de entrenamiento, el tipo de acondicionamiento y normalización de los vectores de entrada, la selección de variables para formar estos vectores de entrada y el ajuste de diversos parámetros. Dentro de estos últimos se tienen el tiempo de muestreo, el número de épocas de entrenamiento, los parámetros de vigilancia (ρ_a , ρ_b y ρ_{ab}), los parámetros de elección (α_a y α_b) y el tipo de aprendizaje. Todos estos factores, a excepción del esquema de entrenamiento, se definen por medio del método de prueba y error; es decir, se realizan diferentes simulaciones con diferentes asignaciones para estos factores y se toman aquellas que conducen a un mejor funcionamiento de la arquitectura FAM como regulador de voltaje del sistema bajo prueba.

3.2.1 Esquema de Entrenamiento

Dado que la arquitectura FAM es inherentemente una red neuronal con aprendizaje supervisado, el procedimiento para su aplicación como regulador de voltaje de un generador síncrono que se adopta en este trabajo es el modelado de un controlador existente (véase el Apéndice E). El controlador elegido es el regulador de voltaje convencional ST1 (excitador con fuente de potencial regulada por rectificadores controlados) por ser uno de los más empleados en sistemas de potencia de grandes dimensiones, dado que posee una alta capacidad de esfuerzos post-falla, un relativo bajo costo y un fácil mantenimiento. El modelo de este regulador se presenta en el Apéndice A.

En la figura 3.1 se muestra el procedimiento de copiado, donde V_{ref} es el voltaje de referencia, V_T es el voltaje en terminales, e es el error entre el voltaje de referencia y el voltaje en terminales, y E_{FD} es el voltaje de campo. La fase de entrenamiento se realiza colocando la arquitectura FAM en paralelo con el regulador ST1 (figura 3.1a). En la fase de producción se sustituye este regulador por la arquitectura FAM (figura 3.1b).

Las condiciones propuestas para el entrenamiento son las siguientes:

1. El regulador ST1 debe estar previamente sintonizado sin carga con la finalidad de obtener mejores resultados.
2. Se debe simular que el sistema máquina bus infinito parte de condiciones iniciales específicas.
3. Se debe incluir la simulación de una falla de corto circuito trifásica una vez que los transitorios sean mínimos, con una duración significativa en la que el regulador ST1 ofrezca resultados satisfactorios.
4. Las simulaciones se deben repetir n_c veces, cada una con diferentes condiciones iniciales. A este ciclo se le conoce como época de entrenamiento.

El espacio de salida del regulador ST1 depende de unos límites que se incluyen en su modelo, los cuales mantienen al voltaje de campo en el intervalo $[E_{FD \text{ mín}}, E_{FD \text{ máx}}]$; por lo

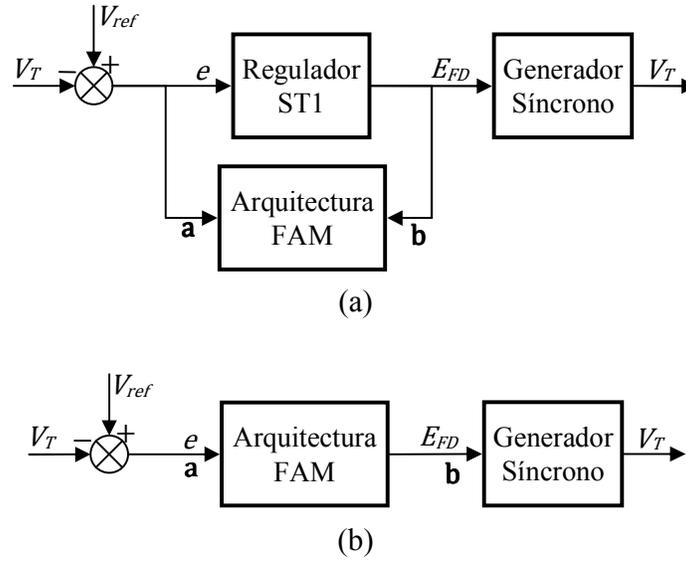


Figura 3.1. Procedimiento inicial para la aplicación de la arquitectura FAM como regulador de voltaje de un generador síncrono. (a) Entrenamiento. (b) Producción.

tanto, la arquitectura FAM debe ser capaz de mapear a este mismo espacio de salida una vez que se emplee como regulador. Por esta razón, se incluye la simulación de una falla de corto circuito trifásica en las condiciones para el entrenamiento, puesto que este evento provoca que el voltaje de campo abarque todo el espacio de salida.

3.2.2 Acondicionamiento y Normalización de los Vectores de Entrada

Las variables de entrada son los elementos de los vectores de entrada de la arquitectura FAM. En el Capítulo 2 se menciona que estos elementos deben encontrarse en el intervalo $[0, 1]$, por lo que cualquier variable que se considere en el entrenamiento debe cumplir este requisito. Esto se puede lograr incluyendo una etapa inicial de acondicionamiento de estas variables, para lo cual es necesario definir los máximos y mínimos globales de cada variable. Estos valores se pueden obtener destinando una época de entrenamiento para tal propósito. De esta manera, cualquier variable acondicionada v_{ac} está dada por

$$v_{ac} = \frac{v_{real} - v_{\min global}}{v_{\max global} - v_{\min global}}, \quad (3.1)$$

donde v_{real} es el valor real de dicha variable.

No obstante, en algunas aplicaciones resulta conveniente acondicionar las variables de entrada en base a aquel máximo o mínimo global que posea el mayor valor absoluto; es decir, a un valor v_M que está dado por

$$v_M = \max(|v_{\max global}|, |v_{\min global}|), \quad (3.2)$$

donde $|\cdot|$ es el valor absoluto. Así, la ecuación (3.1) se convierte en

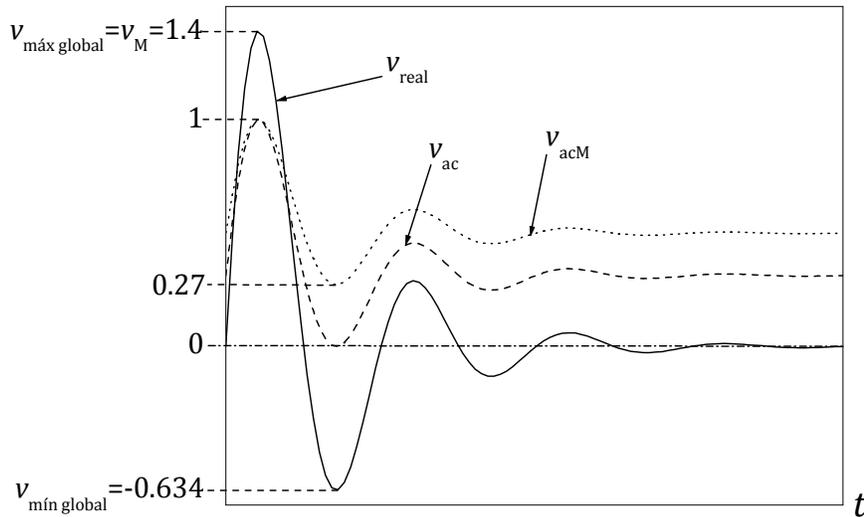


Figura 3.2. Ejemplo de los dos tipos de acondicionamiento de una variable real para que sus valores oscilen en el intervalo $[0, 1]$.

$$v_{acM} = \frac{v_{real} + v_M}{2v_M}. \quad (3.3)$$

En la figura 3.2 se muestran estos dos tipos de acondicionamiento.

Es recomendable utilizar el acondicionamiento de las ecuaciones (3.2) y (3.3) en los casos en que se desee que los prototipos de las categorías identificadas presenten simetría con respecto a un valor central. En cualquier caso, el proceso de acondicionamiento también representa un medio para definir prototipos adimensionales, lo que es una característica muy conveniente en tareas de reconocimiento y categorización de patrones.

Además del acondicionamiento de las variables de entrada, es necesario incluir algún tipo de normalización de los vectores que forman. En el Capítulo 2 se menciona que la normalización más apropiada para los vectores de entrada de una red FART es la codificación con complemento. Por lo tanto, de acuerdo con la ecuación (2.18), una entrada \mathbf{B} con elementos acondicionados y normalizada por la codificación con complemento queda definida por

$$\mathbf{B} = (\mathbf{b}, \mathbf{b}^c) = (b_{ac1}, \dots, b_{acm}, 1 - b_{ac1}, \dots, 1 - b_{acm}) \quad (3.4)$$

o por

$$\mathbf{B} = (\mathbf{b}, \mathbf{b}^c) = (b_{acM1}, \dots, b_{acMm}, 1 - b_{acM1}, \dots, 1 - b_{acMm}). \quad (3.5)$$

En este trabajo se utiliza el acondicionamiento de las ecuaciones (3.2) y (3.3), dado que proporciona resultados más satisfactorios que el definido por la ecuación (3.1). Por consiguiente, la normalización de los vectores de entrada que se adopta es la expresada en la ecuación (3.5).

3.2.3 Selección de las Variables de Entrada

En la figura 3.1 se aprecia que la variable de entrada al regulador ST1 es el error existente entre el voltaje de referencia y el voltaje en terminales, mientras que la variable de salida es el voltaje de campo. Si se eligen sólo estas dos variables para formar los respectivos vectores de entrada de la arquitectura FAM, entonces las entradas acondicionadas y normalizadas para el entrenamiento, de acuerdo con la ecuación (3.5), son

$$\mathbf{A}(\tau) = (\mathbf{a}(\tau), \mathbf{a}^c(\tau)) = (e_{acM}(k), 1 - e_{acM}(k)) \quad (3.6)$$

para la red FART_a, y

$$\mathbf{B}(\tau) = (\mathbf{b}(\tau), \mathbf{b}^c(\tau)) = (E_{FDacM}(k), 1 - E_{FDacM}(k)) \quad (3.7)$$

para la red FART_b, donde k es el paso de integración y τ es el tiempo de muestreo. La relación entre estos dos parámetros se describe en la Sección 3.2.4.

El funcionamiento de la arquitectura FAM considerando sólo las variables de las ecuaciones (3.6) y (3.7) es muy deficiente una vez que se aplica como regulador de voltaje del sistema bajo prueba, por lo que es necesario incluir otras variables. Después de distintas pruebas realizadas se define que las variables de entrada a la red FART_a son el error del voltaje en terminales e , el incremento de este error Δe y su sumatoria $\sum e$, las cuales están dadas por

$$e(k) = V_{ref}(k) - V_T(k), \quad (3.8)$$

$$\Delta e(k) = e(k) - e(k-1) \quad (3.9)$$

y

$$\sum e(k) = e(k) + e(k-1); \quad (3.10)$$

mientras que la variable de entrada a la red FART_b es la diferencia entre el voltaje de campo vigente y el voltaje de campo inicial, denotada por dE_{FD} ; es decir,

$$dE_{FD}(k) = E_{FD}(k) - E_{FD}(1). \quad (3.11)$$

De este modo, las entradas definitivas acondicionadas y normalizadas por codificación con complemento de las redes FART_a y FART_b de la arquitectura FAM son

$$\mathbf{A}(\tau) = (e_{acM}(k), \Delta e_{acM}(k), \sum e_{acM}(k), 1 - e_{acM}(k), 1 - \Delta e_{acM}(k), 1 - \sum e_{acM}(k)) \quad (3.12)$$

$$\mathbf{B}(\tau) = (dE_{FDacM}(k), 1 - dE_{FDacM}(k))$$

donde

$$e_{acM}(k) = \frac{e(k) + e_M}{2e_M}, \quad (3.13)$$

$$\Delta e_{acM}(k) = \frac{\Delta e(k) + \Delta e_M}{2\Delta e_M}, \quad (3.14)$$

$$\sum e_{acM}(k) = \frac{\sum e(k) + \sum e_M}{2\sum e_M} \quad (3.15)$$

y

$$dE_{FDacM}(k) = \frac{dE_{FD}(k) + dE_{FDM}}{2dE_{FDM}}. \quad (3.16)$$

3.2.4 Ajuste de Parámetros

El número de épocas de entrenamiento, el tiempo de muestreo y los parámetros de la arquitectura FAM tienen un efecto decisivo en el reconocimiento y generación de categorías durante el entrenamiento, lo que influye directamente en el funcionamiento de esta arquitectura como regulador de voltaje del sistema bajo prueba.

Los parámetros de vigilancia (ρ 's) en la fase de entrenamiento deben ser lo suficientemente altos como para generar categorías cuyos prototipos no sean muy distintos entre sí; esto con la finalidad de que la variable de salida de la arquitectura FAM no sufra cambios muy drásticos de un instante a otro durante la etapa de producción. Sin embargo, no es conveniente que el valor de estos parámetros sea demasiado alto, puesto que esto genera un número excesivo de categorías, lo que eleva el tiempo de cómputo en la fase de producción.

Los parámetros de elección (α 's) y el tipo de aprendizaje están estrechamente relacionados con el número de épocas de entrenamiento. Si estos parámetros son muy bajos (cerca del límite conservativo descrito en la Sección 2.3.6) y si el aprendizaje es rápido ($\beta = 1$), entonces el número de épocas de entrenamiento necesario para crear categorías con las características definidas en el párrafo anterior se reduce considerablemente.

El tiempo de muestreo debe ser un múltiplo del paso de integración, evidentemente para evitar ambigüedades acerca de las entradas de la arquitectura FAM durante el entrenamiento en un determinado instante. Así, se tiene que

$$\tau = ck, \quad c \geq 1, \quad c \in \mathbb{N}. \quad (3.17)$$

Si $c = 1$, entonces el tiempo de cómputo requerido para el entrenamiento se incrementa, lo cual no es muy perjudicial si se considera que los parámetros de elección son cercanos a cero, el aprendizaje es rápido y el número de épocas de entrenamiento es reducido.

En base a lo anterior y a diversas pruebas realizadas, el ajuste definido para todos estos parámetros es el que se indica en la tabla 3.1.

Tabla 3.1. Parámetros para el entrenamiento.

| Parámetro | Valor | |
|--|-------------------------|-------------------------|
| No. Épocas de entrenamiento n_{ep} | 1 | |
| Tiempo de muestreo τ | $k, c = 1$ | |
| Parámetro de vigilancia del campo de mapeo ρ_{ab} | 0.975 | |
| | FART_a | FART_b |
| Parámetro de vigilancia ρ | 0.975 | 0.975 |
| Parámetro de elección α | 0.01 | 0.01 |
| Aprendizaje | Rápido ($\beta = 1$) | Rápido ($\beta = 1$) |

Los parámetros de vigilancia, los parámetros de elección y el tipo de aprendizaje son los mismos para ambas redes FART debido a que éstas, junto con el campo de mapeo, deben presentar características similares para evitar inconsistencias en el funcionamiento de la arquitectura FAM.

3.3 ENTRENAMIENTO DE LA ARQUITECTURA ARTMAP DIFUSA

Después de definidos los factores necesarios para el entrenamiento de la arquitectura FAM, el procedimiento de copiado de la figura 3.1 se convierte en el mostrado en la figura 3.3, donde ya se contemplan las variables adecuadas así como su acondicionamiento.

El entrenamiento de la arquitectura FAM se realiza de acuerdo a la figura 3.3a, con los parámetros ajustados como lo indica la tabla 3.1 y bajo las condiciones descritas en la Sección 3.2.1.

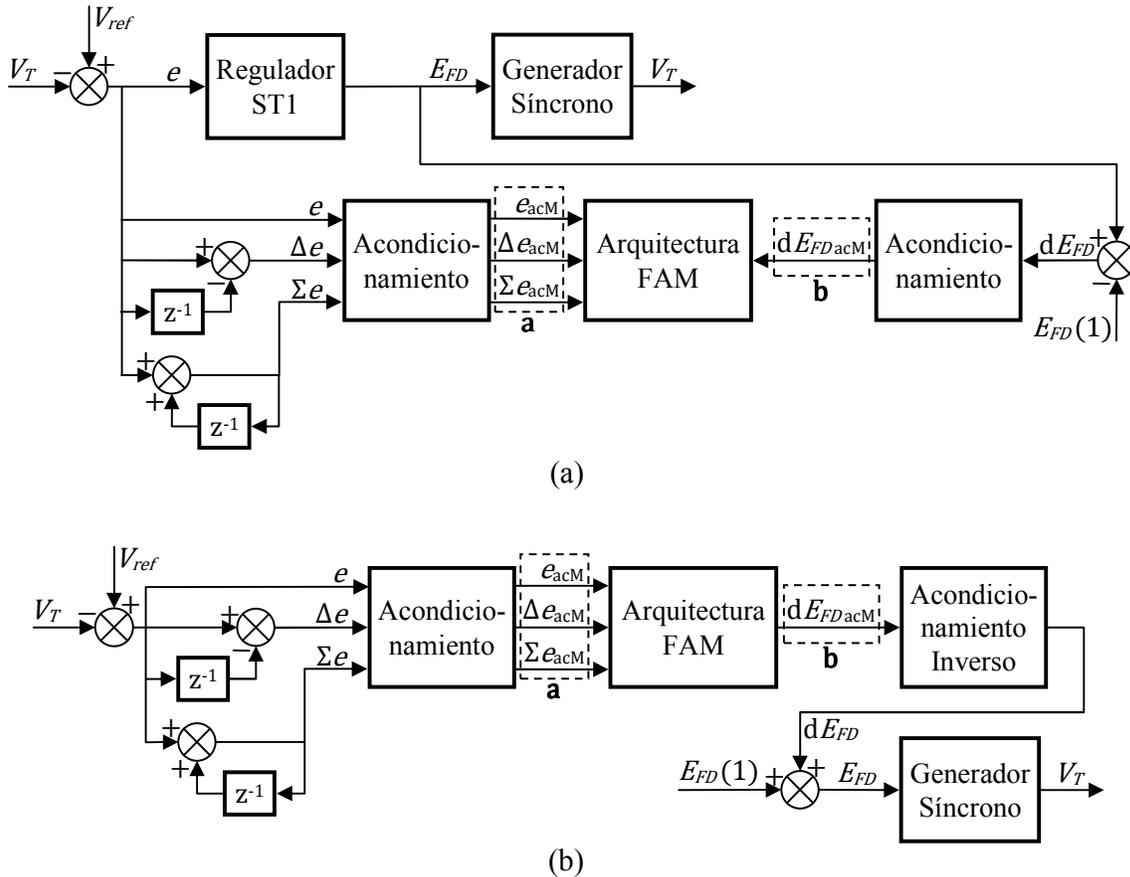


Figura 3.3. Procedimiento para la aplicación de la arquitectura FAM como regulador de voltaje de un generador síncrono incluyendo las variables adecuadas así como su acondicionamiento. (a) Entrenamiento. (b) Producción.

3.3.1 Falla de Corto Circuito Trifásica para el Entrenamiento

Evidentemente, se pretende que la arquitectura FAM copie respuestas satisfactorias ante eventos severos; por lo tanto, se simula que la falla que se contempla en las condiciones para el entrenamiento de la Sección 3.2.1 inicia en $t = 5$ seg. (instante en el que ya no existen transitorios) con una duración de 6 ciclos (100 mseg.). En el diagrama de la figura 3.4 se muestra la forma en que se suscita este evento, el cual se logra simular haciendo que el voltaje en el bus infinito caiga hasta un valor cercano a cero (0.1 p.u.) durante el tiempo de duración de la falla, mientras que en cualquier otro instante, el valor de este voltaje es de 1 p.u. La base que se toma para el sistema por unidad en este trabajo es la capacidad nominal del generador, que es de 645 MVA [28].

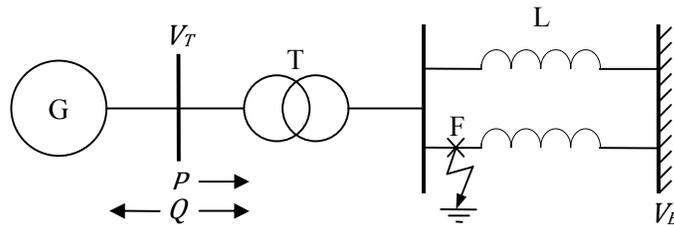


Figura 3.4. Falla de corto circuito trifásica para un sistema máquina bus infinito.

3.3.2 Condiciones Iniciales para el Entrenamiento

La arquitectura FAM debe ser entrenada con condiciones iniciales bajo las cuales el regulador ST1 ofrezca resultados favorables ante la falla descrita en la Sección 3.3.1. En la tabla 3.2 se muestran diez de estas condiciones iniciales; mientras que en la tabla 3.3 se muestran tres condiciones iniciales bajo las que el regulador ST1 cae en la inestabilidad ante la falla mencionada, razón por la cual estas condiciones iniciales se consideran críticas.

Tabla 3.2. Condiciones iniciales bajo las que el regulador ST1 responde eficientemente.

| No. de Condición Inicial | Voltaje en Terminales (p.u.) | Ángulo de Carga (grados) | Potencia Activa (p.u.) | Potencia Reactiva (p.u.) |
|--------------------------|------------------------------|--------------------------|------------------------|--------------------------|
| 1 | 1.0000000 | 71.8097608 | 0.8805000 | -0.1361000 |
| 2 | 1.0426910 | 53.0227670 | 0.8971906 | 0.2394987 |
| 3 | 1.0291030 | 46.5707683 | 0.6517569 | 0.1460104 |
| 4 | 0.9998438 | 65.8952537 | 0.7357140 | -0.1215235 |
| 5 | 1.0152080 | 53.4362629 | 0.6588190 | 0.0210546 |
| 6 | 1.0453240 | 43.0389266 | 0.7031847 | 0.2880037 |
| 7 | 0.9813626 | 69.8150523 | 0.5678285 | -0.2549196 |
| 8 | 1.0332250 | 47.3260930 | 0.6972846 | 0.1772876 |
| 9 | 1.0563940 | 38.2865221 | 0.6808600 | 0.3954524 |
| 10 | 1.0016400 | 52.8155086 | 0.5273286 | -0.0787794 |

Tabla 3.3. Condiciones iniciales críticas.

| No. de Condición Inicial | Voltaje en Terminales (p.u.) | Ángulo de Carga (grados) | Potencia Activa (p.u.) | Potencia Reactiva (p.u.) |
|---------------------------------|-------------------------------------|---------------------------------|-------------------------------|---------------------------------|
| 28 | 1.0000000 | 75.7887306 | 1.0000000 | -0.1473878 |
| 29 | 0.9896917 | 79.0575924 | 0.9319504 | -0.2272967 |
| 30 | 0.9904293 | 79.4581517 | 0.9592349 | -0.2236688 |

Es más, las condiciones iniciales de la tabla 3.3 se encuentran más cerca de los límites seguros de operación en una curva típica de capacidad reactiva de un generador [29] que las condiciones iniciales de la tabla 3.2.

El número de cada condición inicial de las tablas 3.2 y 3.3 corresponde al orden definido en el Apéndice B, apartado en el que se presenta el conjunto de condiciones iniciales completo considerado para la realización de este trabajo.

3.3.3 Categorías de Salida

Durante la fase de entrenamiento, las entradas de las respectivas redes FART (ecuación (3.12)) son leídas, aprendidas, categorizadas y asociadas entre sí a través del campo de mapeo. El número de categorías generadas en estas redes aumenta con el número de condiciones iniciales utilizadas para el entrenamiento (CIE), por lo que el funcionamiento de la arquitectura FAM en la etapa de producción (figura 3.3b) se ve afectado también por este factor. En la tabla 3.4 se muestra el número de categorías generadas en ambas redes FART habiendo utilizado la primera, las primeras cinco y las diez condiciones iniciales de la tabla 3.2 para el entrenamiento. Si se utilizan más condiciones iniciales, el número de categorías en la red FART_a sigue aumentando; sin embargo, el número de categorías en la red FART_b se estabiliza en 33, es por ello que la arquitectura FAM se entrena hasta con un máximo de diez condiciones iniciales.

Tabla 3.4. Número de categorías generadas en las redes FART_a y FART_b utilizando las condiciones iniciales 1, 1-5 y 1-10 para el entrenamiento de la arquitectura FAM.

| Condiciones iniciales utilizadas en el entrenamiento (CIE) | Número de categorías generadas en la red FART_a | Número de categorías generadas en la red FART_b |
|---|--|--|
| 1 | 24 | 16 |
| 1-5 | 76 | 21 |
| 1-10 | 292 | 33 |

3.4 ARQUITECTURA ARTMAP DIFUSA COMO REGULADOR DE VOLTAJE

En esta sección se presentan los resultados que se obtienen aplicando la arquitectura FAM como regulador de voltaje del sistema bajo prueba (fase de producción) partiendo de tres diferentes condiciones iniciales, habiendo utilizado las condiciones iniciales indicadas por la tabla 3.4 para el entrenamiento. Esta aplicación se lleva a cabo de acuerdo a la figura 3.3b, donde el bloque de acondicionamiento inverso proporciona dE_{FD} por medio de

$$dE_{FD}(k) = dE_{FDM}(2b-1), \quad (3.18)$$

donde se utiliza b y no \mathbf{b} porque, de acuerdo con la ecuación (3.12), el vector \mathbf{b} es unidimensional. El voltaje de campo, como se puede observar en la figura 3.3b, se determina por

$$E_{FD}(k) = E_{FD}(1) + dE_{FD}(k). \quad (3.19)$$

El vector \mathbf{b} constituye el primer elemento del prototipo almacenado en la categoría de la red $FART_b$ que está asociada a la categoría de la red $FART_a$ a la cual pertenece la entrada \mathbf{A} en un determinado instante; es decir, dada una entrada \mathbf{A} , la red $FART_a$ identifica la categoría apropiada y, a través del campo de mapeo, se determina la categoría en la red $FART_b$ cuyo prototipo se convierte en la salida de la arquitectura FAM.

Con la finalidad de asegurar que la arquitectura FAM proporcione una salida en cada instante de integración, los parámetros de vigilancia de la red $FART_a$ y del campo de mapeo para la fase de producción son reajustados a un valor menor que el definido para el entrenamiento. Por otra parte, conviene que el parámetro de elección de la red $FART_a$ también sea reajustado, puesto que un valor mayor de éste proporciona estabilidad en la elección de categorías, lo que conduce a mejores resultados en estado estacionario. Por lo tanto, el valor que se define para estos parámetros durante la fase de producción es $\rho_a = 0.85$, $\rho_{ab} = 0.85$ y $\alpha_a = 0.05$.

Debido a que el objetivo es evaluar la forma en que la arquitectura FAM copia la dinámica del regulador ST1 independientemente del tiempo, en las simulaciones realizadas se incluye también una falla trifásica con un número de ciclos de duración igual al contemplado en el entrenamiento, pero con un inicio en $t = 2.5$ seg. En todos los casos, los resultados que se obtienen con la arquitectura FAM se comparan con los respectivos que proporciona el regulador ST1. Estos resultados constan de las respuestas para: el voltaje en terminales, el voltaje de campo, las potencias activa y reactiva, el ángulo de carga y el índice de comportamiento del voltaje en terminales. Las cuatro primeras variables son presentadas en el sistema por unidad mencionado en la Sección 3.3.1, mientras que el ángulo de carga está dado en grados y el índice de comportamiento del voltaje en terminales está dado por

$$\text{Índ } V_T(k) = e(k)^2 + e(k-1)^2. \quad (3.20)$$

3.4.1 Resultados para una Condición Inicial Medianamente Crítica

En las figuras 3.5-3.10 se muestran los resultados que se obtienen para la condición inicial medianamente crítica correspondiente a la No. 1 de la tabla 3.2 utilizando las condiciones iniciales 1, 1-5 y 1-10 de esta misma tabla para el entrenamiento de la arquitectura FAM.

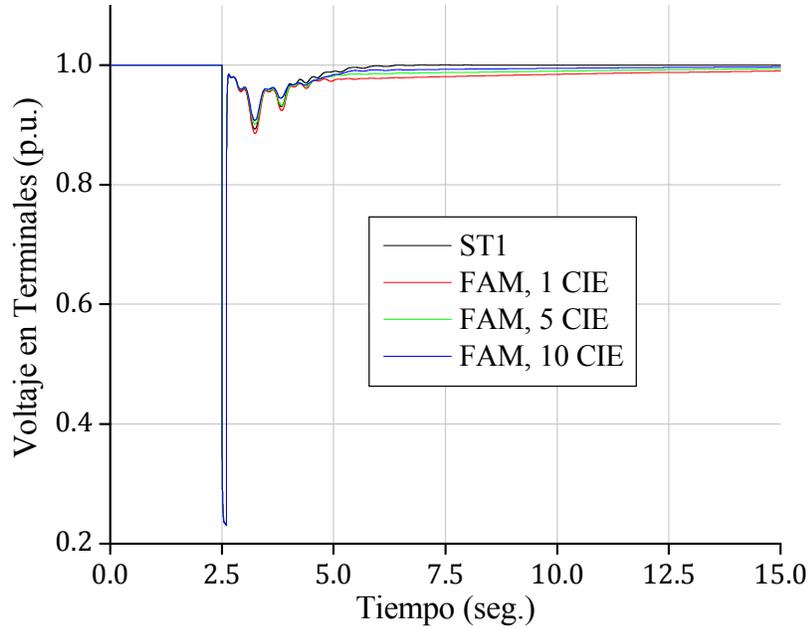


Figura 3.5. Voltaje en terminales para la condición inicial No. 1 con 1, 5 y 10 CIE.

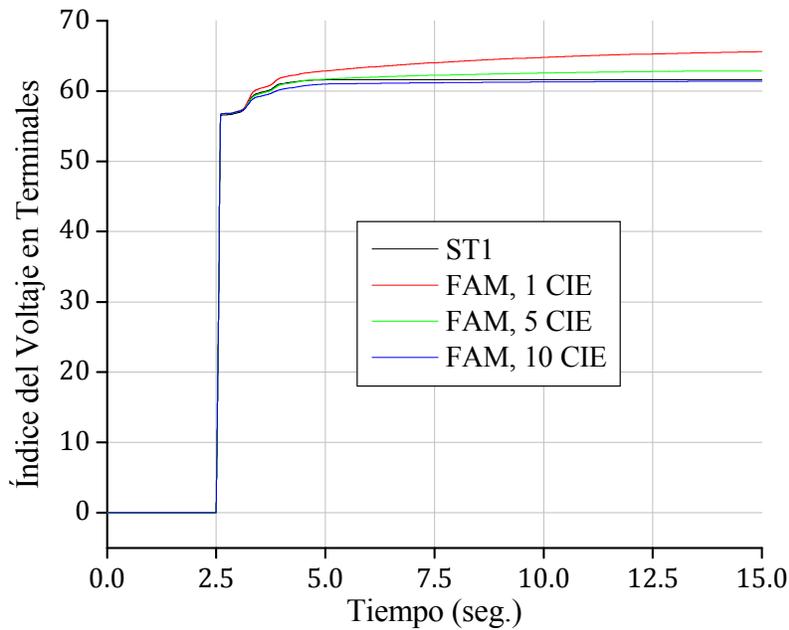


Figura 3.6. Índice de comportamiento del voltaje en terminales para la condición inicial No. 1 con 1, 5 y 10 CIE.

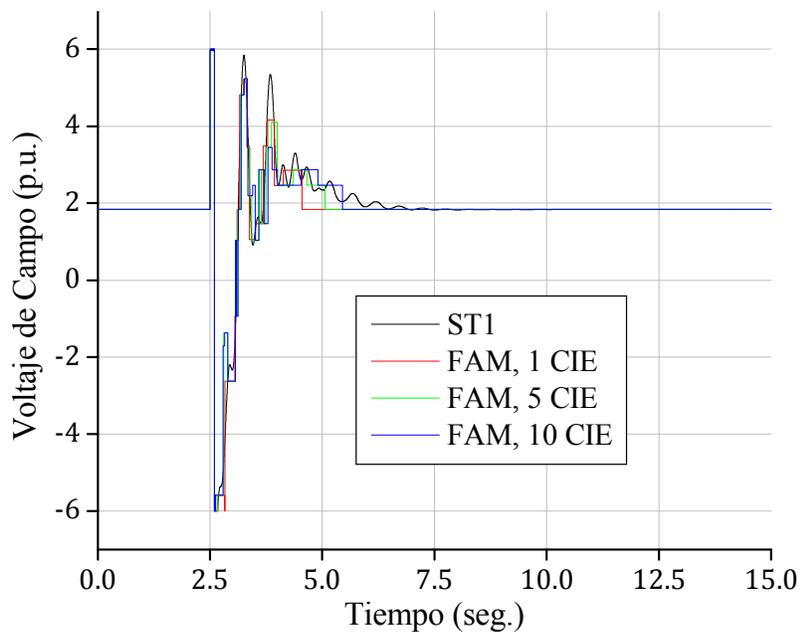


Figura 3.7. Voltaje de campo para la condición inicial No. 1 con 1, 5 y 10 CIE.

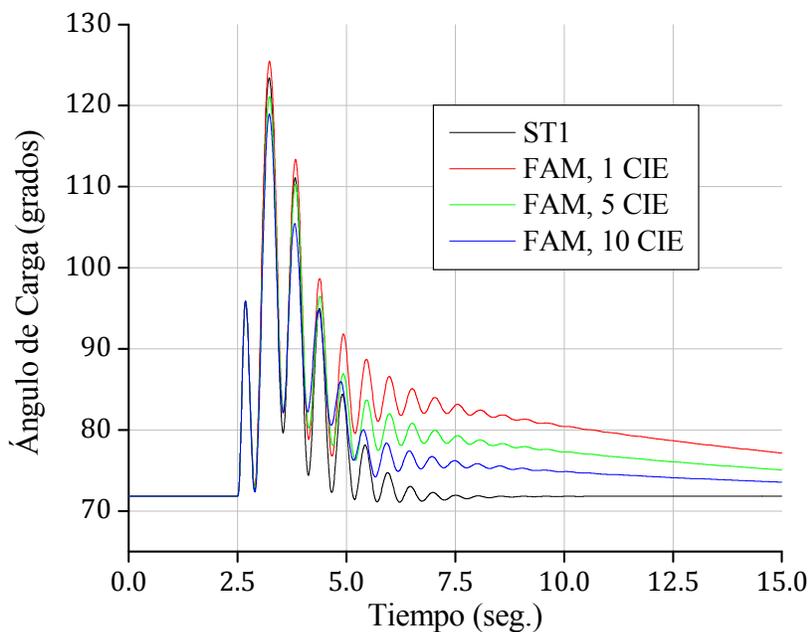


Figura 3.8. Ángulo de carga para la condición inicial No. 1 con 1, 5 y 10 CIE.

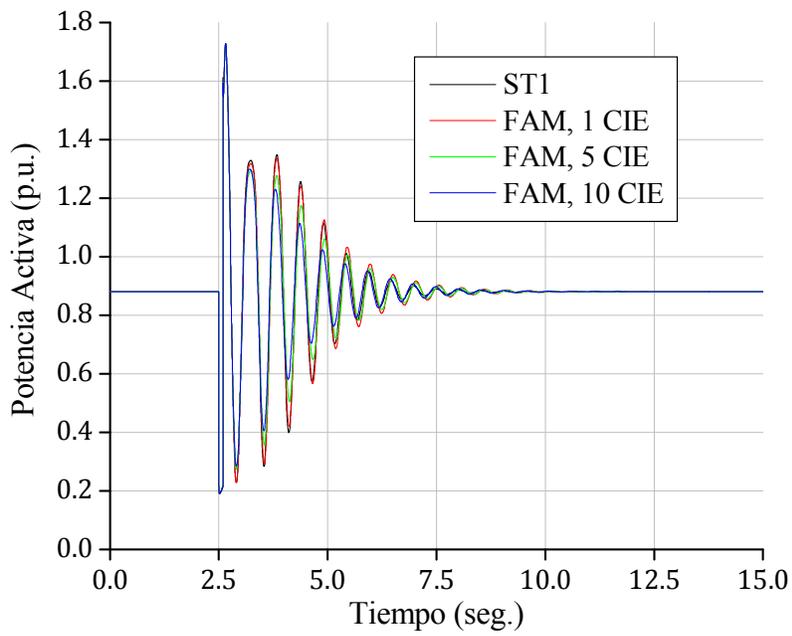


Figura 3.9. Potencia activa para la condición inicial No. 1 con 1, 5 y 10 CIE.

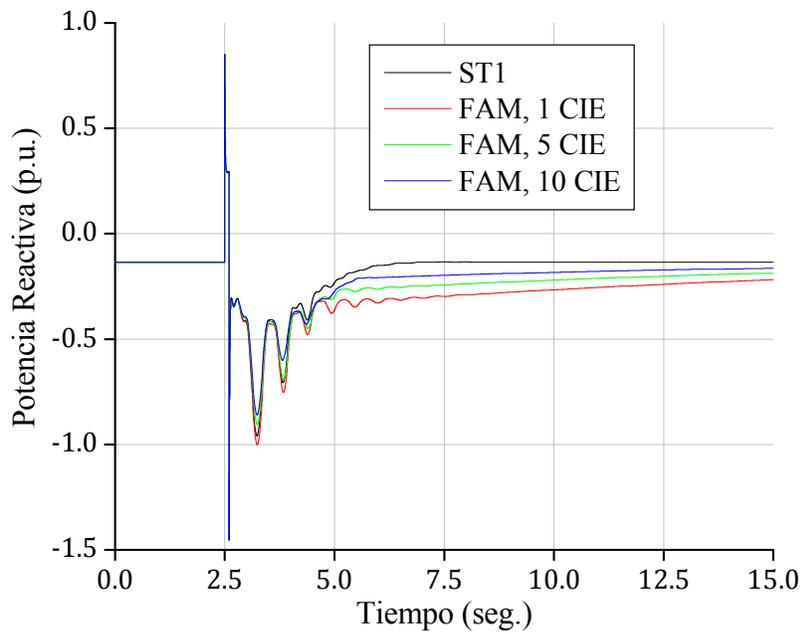


Figura 3.10. Potencia reactiva para la condición inicial No. 1 con 1, 5 y 10 CIE.

3.4.2 Resultados para una Condición Inicial No Crítica

En las figuras 3.11-3.16 se muestran los resultados que se obtienen para la condición inicial no crítica correspondiente a la No. 8 de la tabla 3.2 utilizando las condiciones iniciales 1, 1-5 y 1-10 de esta misma tabla para el entrenamiento de la arquitectura FAM.

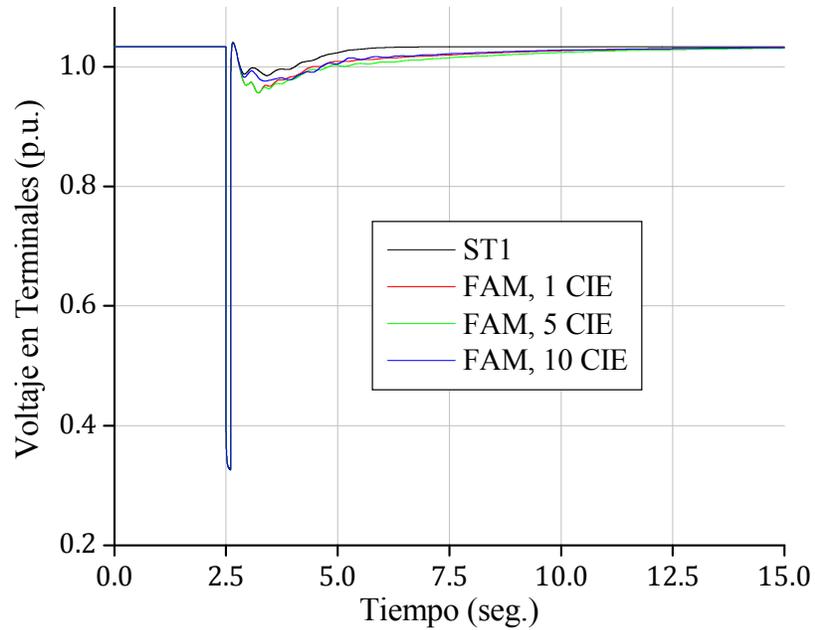


Figura 3.11. Voltaje en terminales para la condición inicial No. 8 con 1, 5 y 10 CIE.

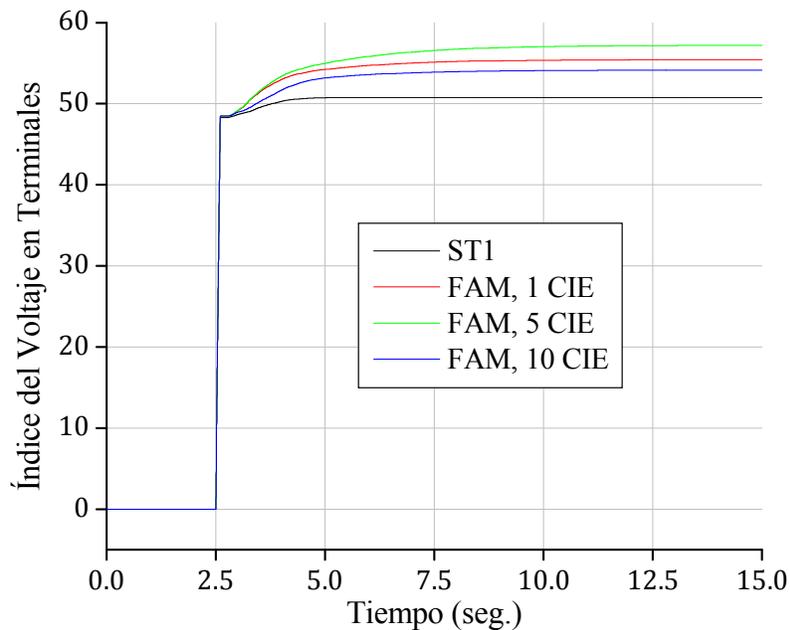


Figura 3.12. Índice de comportamiento del voltaje en terminales para la condición inicial No. 8 con 1, 5 y 10 CIE.

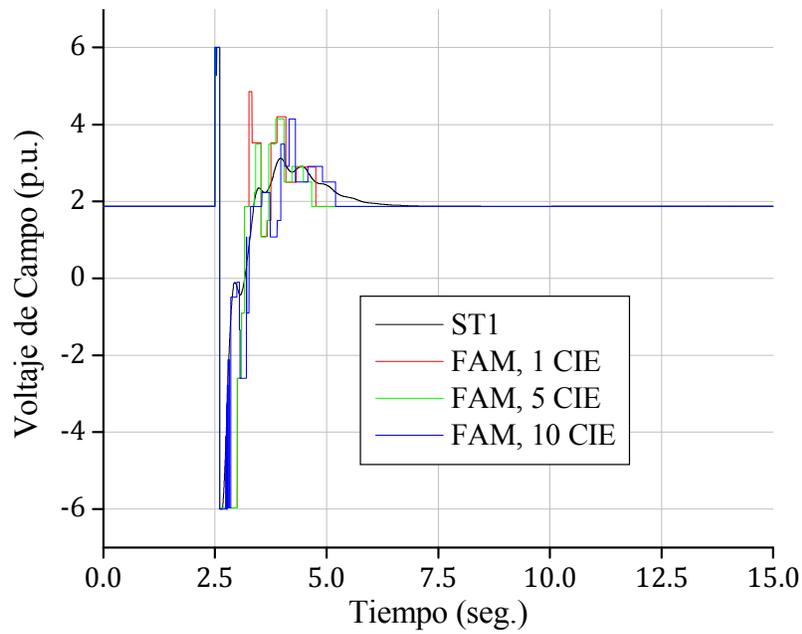


Figura 3.13. Voltaje de campo para la condición inicial No. 8 con 1, 5 y 10 CIE.

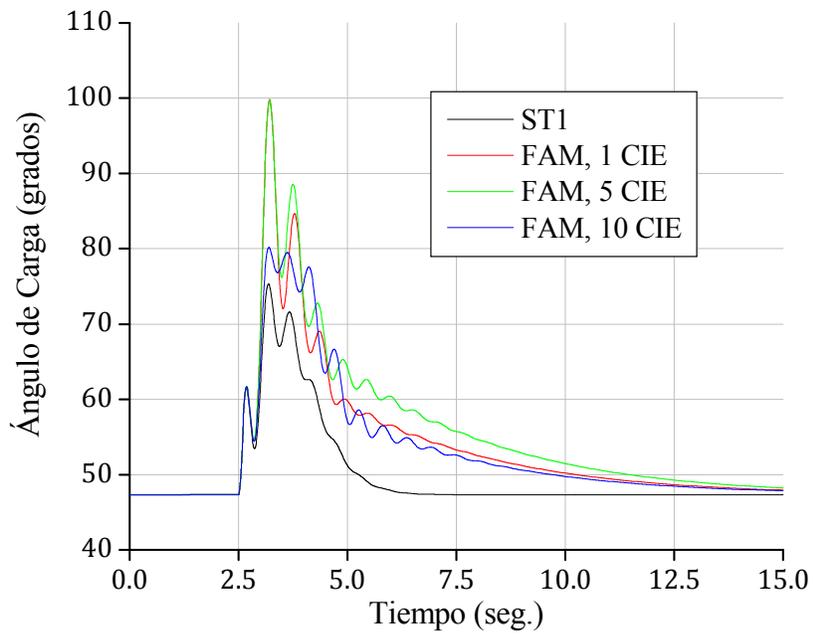


Figura 3.14. Ángulo de carga para la condición inicial No. 8 con 1, 5 y 10 CIE.

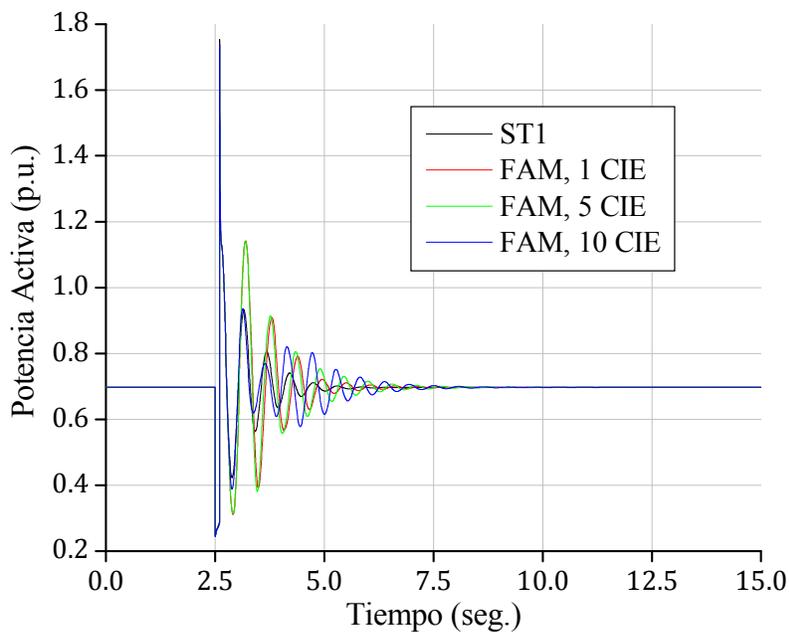


Figura 3.15. Potencia activa para la condición inicial No. 8 con 1, 5 y 10 CIE.

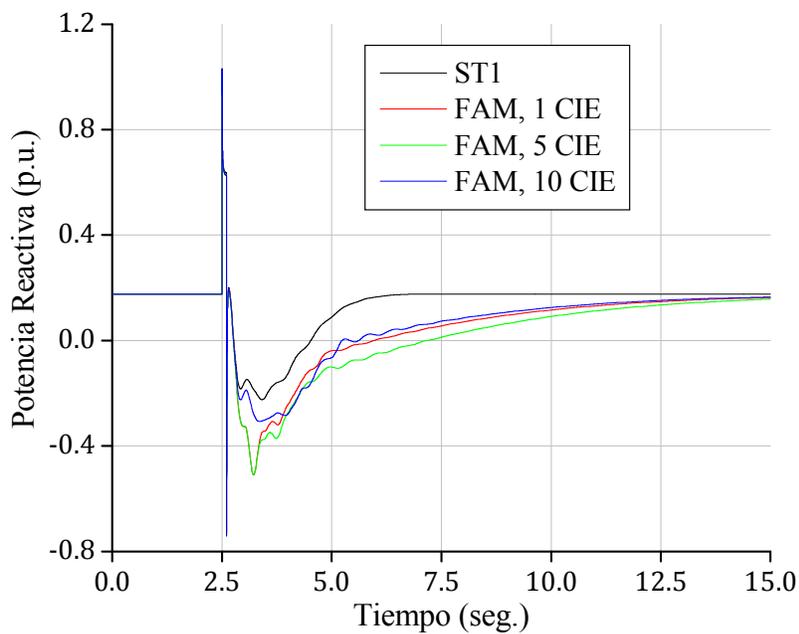


Figura 3.16. Potencia reactiva para la condición inicial No. 8 con 1, 5 y 10 CIE.

3.4.3 Resultados para una Condición Inicial Crítica

En las figuras 3.17-3.22 se muestran los resultados que se obtienen para la condición inicial crítica correspondiente a la No. 28 de la tabla 3.3 utilizando las condiciones iniciales 1, 1-5 y 1-10 de la tabla 3.2 para el entrenamiento de la arquitectura FAM.

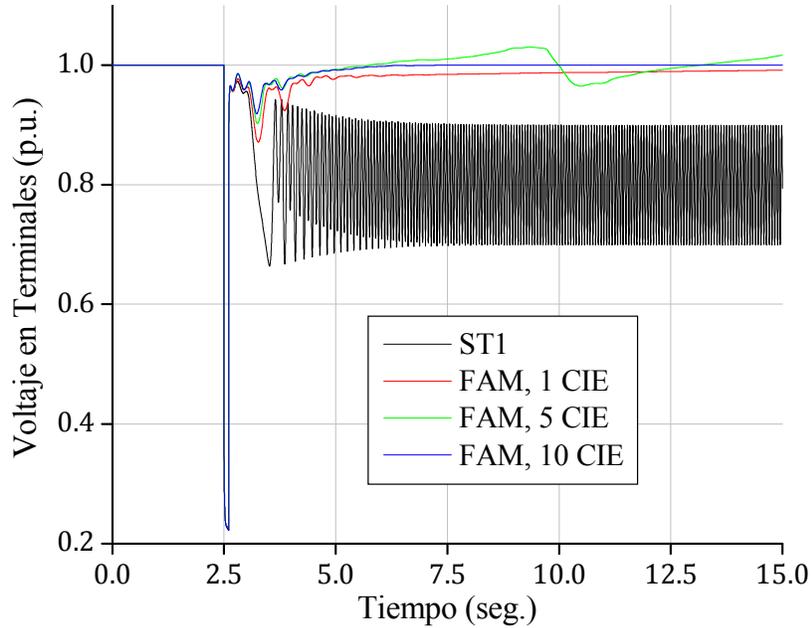


Figura 3.17. Voltaje en terminales para la condición inicial No. 28 con 1, 5 y 10 CIE.

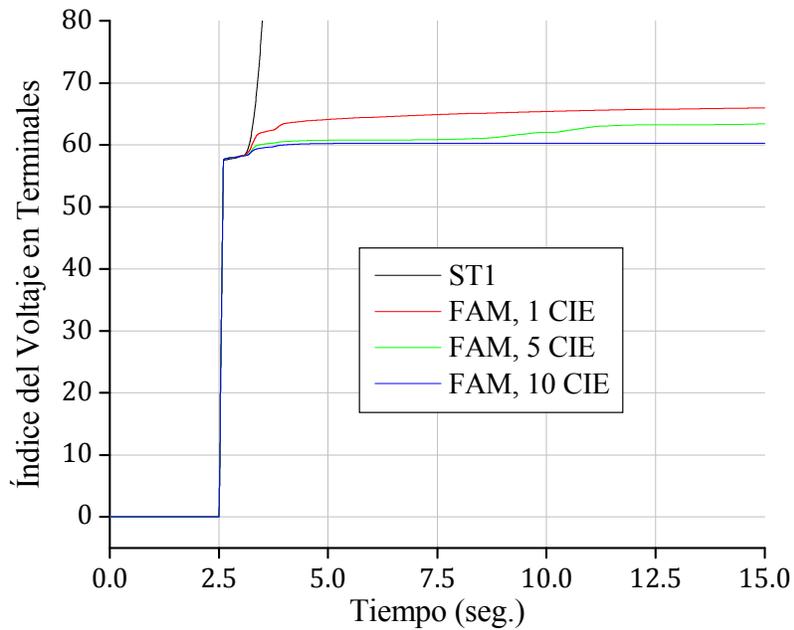


Figura 3.18. Índice de comportamiento del voltaje en terminales para la condición inicial No. 28 con 1, 5 y 10 CIE.

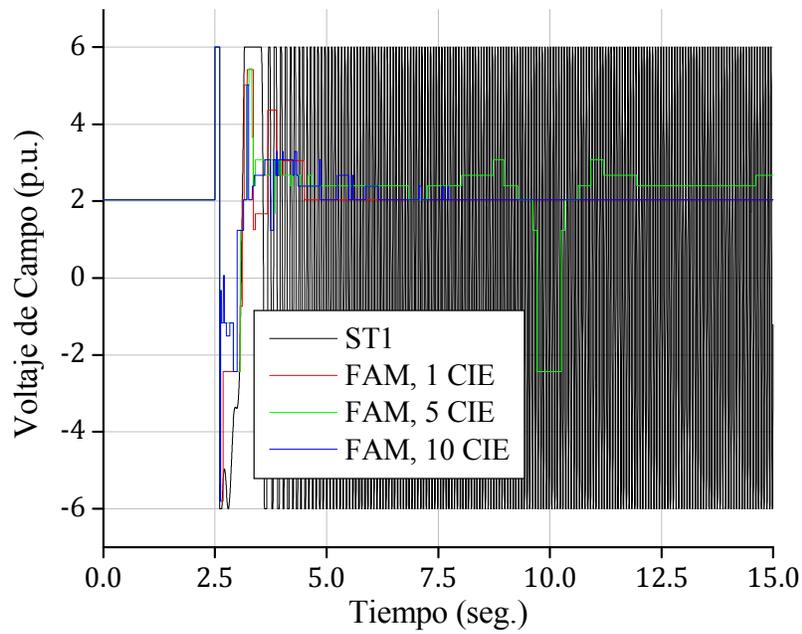


Figura 3.19. Voltaje de campo para la condición inicial No. 28 con 1, 5 y 10 CIE.

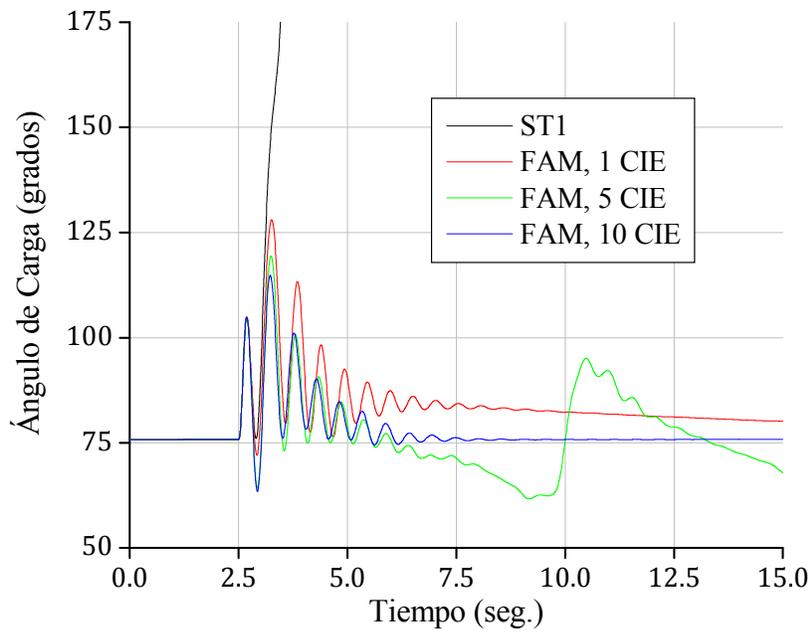


Figura 3.20. Ángulo de carga para la condición inicial No. 28 con 1, 5 y 10 CIE.

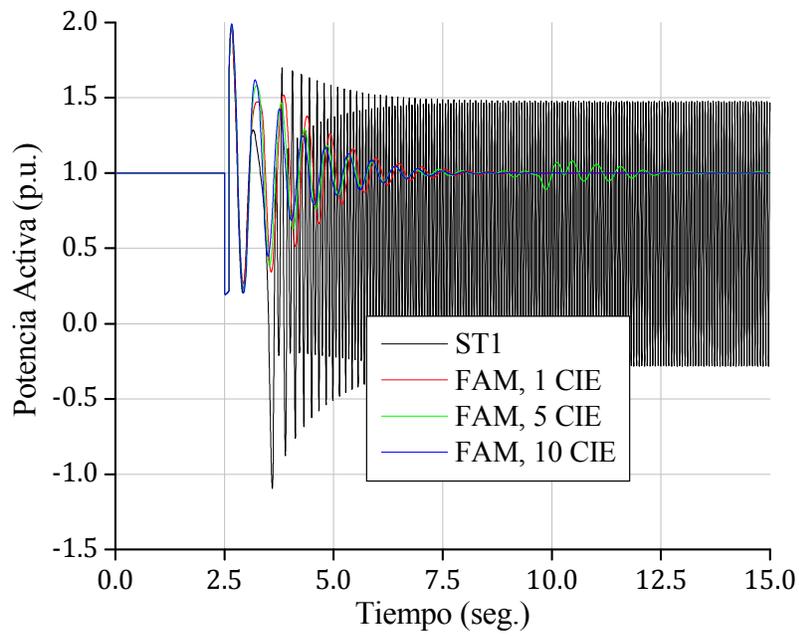


Figura 3.21. Potencia activa para la condición inicial No. 28 con 1, 5 y 10 CIE.

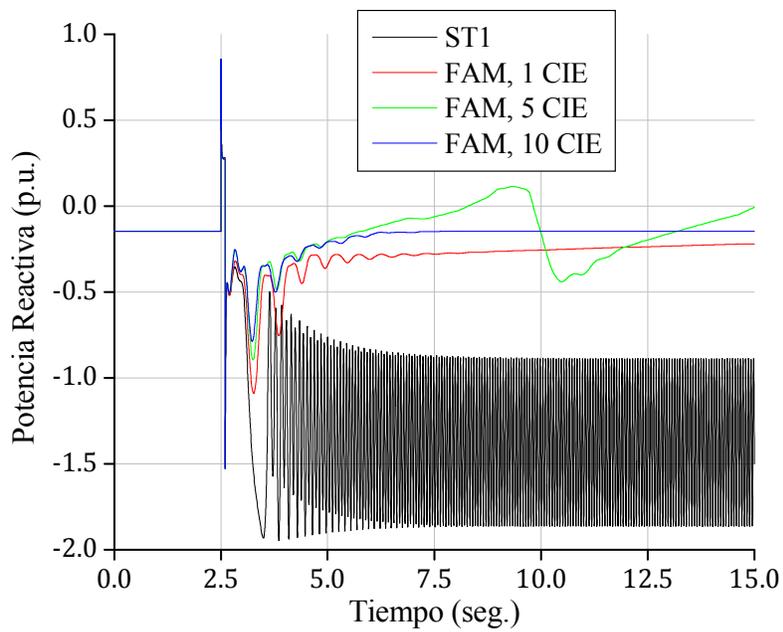


Figura 3.22. Potencia reactiva para la condición inicial No. 28 con 1, 5 y 10 CIE.

3.4.4 Análisis de los Resultados

En los resultados anteriores se puede apreciar lo siguiente:

- El funcionamiento de la arquitectura FAM para las condiciones iniciales 1 y 8 mejora a medida que se aumenta el número de CIE; sin embargo, esto no se cumple para la condición 28, puesto que los resultados que ofrece esta arquitectura con una CIE son mejores que con cinco CIE para esta condición inicial en particular. Por lo tanto, no es posible generalizar que con mayor número de CIE se obtienen mejores resultados para cualquier condición inicial.
- El voltaje de campo que proporciona la arquitectura FAM presenta comportamientos discontinuos (figuras 3.7, 3.13 y 3.19) debido a que, de acuerdo con las ecuaciones (3.18) y (3.19), depende de las categorías definidas en la red $FART_b$, las cuales tienen valores fijos. En la figura 3.23 se muestra un ejemplo de esta situación, donde se aprecia que el voltaje de campo en ningún instante adopta valores intermedios entre las categorías definidas. Para que este voltaje tenga un comportamiento más adecuado, es necesario que el número de categorías definidas en la red $FART_b$ sea lo más grande posible, lo que se puede lograr incrementando el parámetro de vigilancia correspondiente; sin embargo, el tiempo de cómputo también se incrementa en demasía, con lo que el sistema se vuelve impráctico.
- El voltaje de campo en estado estacionario es constante, lo que provoca que la respuesta del sistema con la arquitectura FAM sea demasiado lenta en la mayoría de los casos.

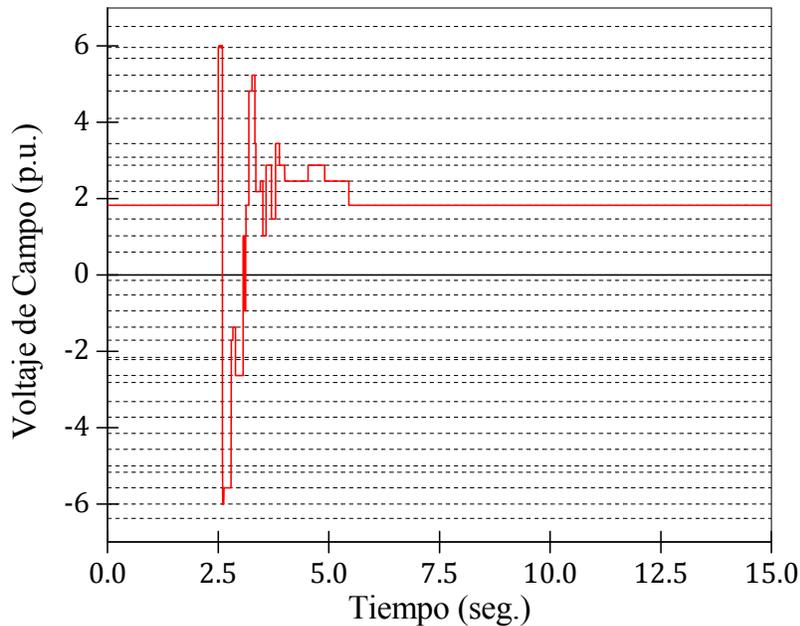


Figura 3.23. Voltaje de campo para la condición inicial No. 1 proporcionado por la arquitectura FAM con 10 CIE sobre las 33 categorías definidas en la red $FART_b$.

- A pesar de que la arquitectura FAM muestra mayor robustez que el regulador convencional ST1 ante fallas de corto circuito, también presenta inconsistencia en su funcionamiento, principalmente bajo condiciones iniciales críticas (figuras 3.17-3.22).

El análisis contenido en los puntos anteriores indica que la arquitectura FAM no es un modelo de red neuronal adecuado para la regulación de voltaje de un generador síncrono, al menos no directamente como en la figura 3.3b. No obstante, la capacidad de realizar mapeos multidimensionales que caracteriza a esta arquitectura puede ser aprovechada de otra manera; por ejemplo, como herramienta para inferir los valores adecuados que sintonicen un determinado controlador.

3.5 USO DE LA ARQUITECTURA ARTMAP DIFUSA PARA SINTONIZAR UN CONTROLADOR DIFUSO

En esta sección se explora la forma en que se puede usar la arquitectura FAM para sintonizar un controlador difuso, de tal manera que la regulación de voltaje del sistema bajo prueba sea lo más satisfactoria posible.

El objetivo es utilizar la arquitectura FAM para identificar si el sistema bajo prueba se encuentra en estado transitorio o en estado estacionario e inferir, en base a ello, un valor que pueda ser utilizado por un controlador difuso en la etapa de dedifusificación para proporcionar la salida más adecuada hacia el generador síncrono. En las figuras 3.7 y 3.13 se puede apreciar claramente que la arquitectura FAM cumple con tal objetivo, puesto que su salida experimenta un cambio drástico durante la falla, pero después regresa gradualmente al valor inicial. Sin embargo, en la figura 3.19 se observa que esto no es general para cualquier condición inicial, debido a la presencia de la sumatoria del error en las variables de entrada. Por consiguiente, para medir el grado de transitoriedad del sistema, basta con considerar únicamente el error y el incremento del error como variables de entrada a la red $FART_a$, mientras que la variable de entrada a la red $FART_b$ conviene que sea también la diferencia entre el voltaje de campo vigente y el inicial.

El esquema de entrenamiento, el entrenamiento y el ajuste de parámetros que se adoptan en este caso son los mismos que los descritos en las secciones anteriores, con la diferencia que la ecuación (3.12) para las entradas de la arquitectura FAM cambia a

$$\begin{aligned} \mathbf{A}(k) &= (e_{acM}(k), \Delta e_{acM}(k), 1 - e_{acM}(k), 1 - \Delta e_{acM}(k)) \\ \mathbf{B}(k) &= (dE_{FD\ acM}(k), 1 - dE_{FD\ acM}(k)) \end{aligned} \quad (3.21)$$

Nótese que ambas entradas están en función de k y no de τ , evidentemente porque el tiempo de muestreo definido en la tabla 3.1 es igual al paso de integración.

En la figura 3.24 se muestra el procedimiento propuesto para que la arquitectura FAM sintonice un controlador difuso, donde los pesos X_1 y X_2 son utilizados por este controlador en la etapa de dedifusificación.

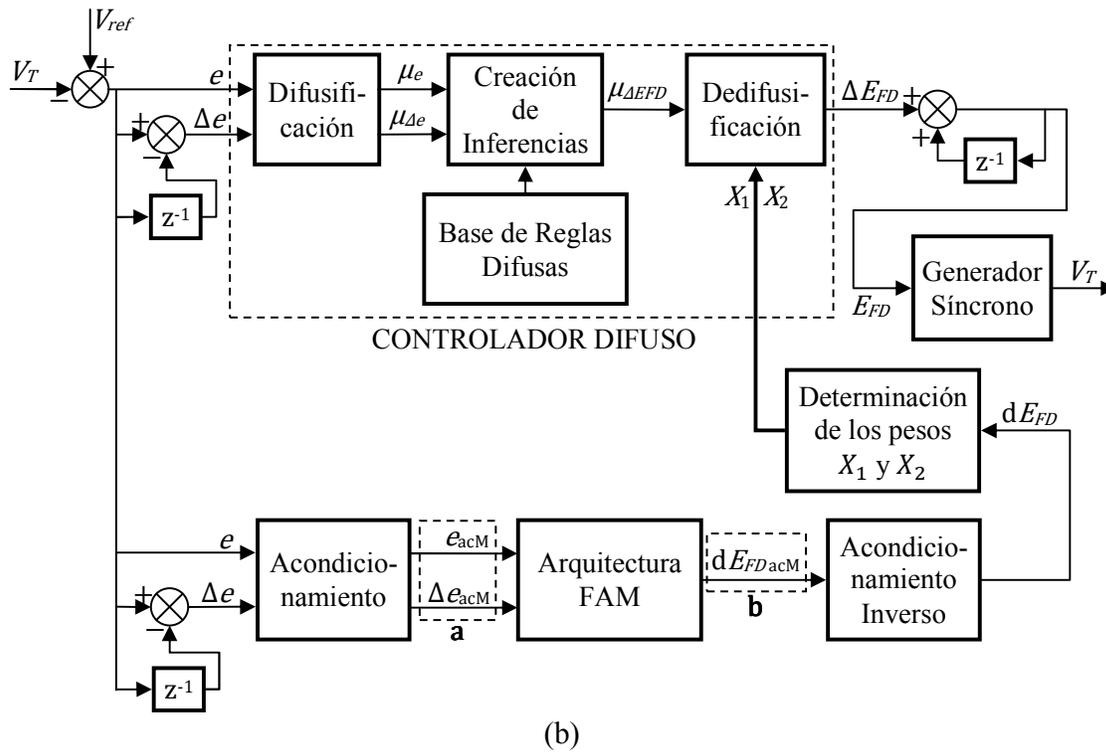
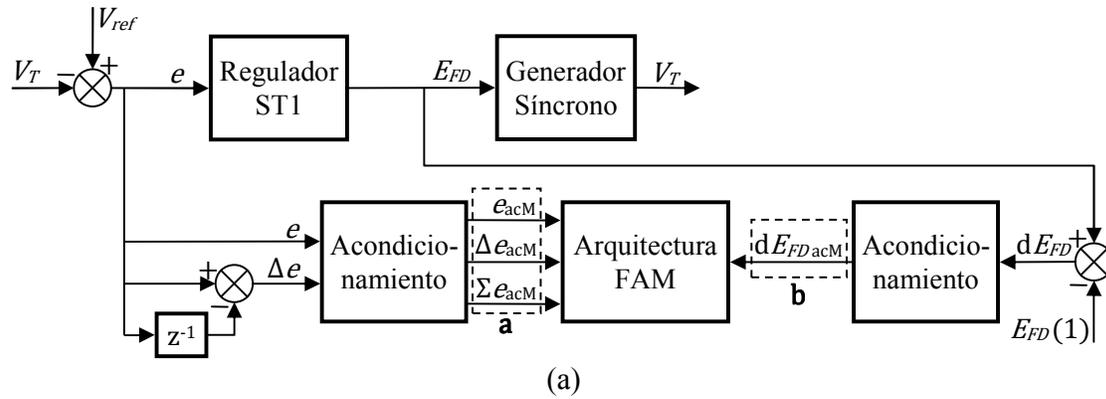


Figura 3.24. Procedimiento de aplicación de la arquitectura FAM para sintonizar un controlador difuso. (a) Entrenamiento. (b) Producción.

3.5.1 Selección y Funcionamiento del Controlador Difuso

De acuerdo con las ecuaciones (3.11) y (3.16), los prototipos de las categorías de la red $FART_b$ almacenan una medida de la diferencia entre el valor del voltaje de campo vigente y el valor inicial de éste. Por esta razón, el controlador difuso propuesto en este trabajo es un proporcional integral (PI) tipo Mamdani, con la finalidad de que la arquitectura FAM proporcione los valores adecuados de los pesos que se necesitan para la dedifusificación, puesto que conviene que estos pesos sean distintos para diferentes situaciones. En la figura 3.25 se muestra un bosquejo de la forma en que opera el controlador difuso seleccionado,

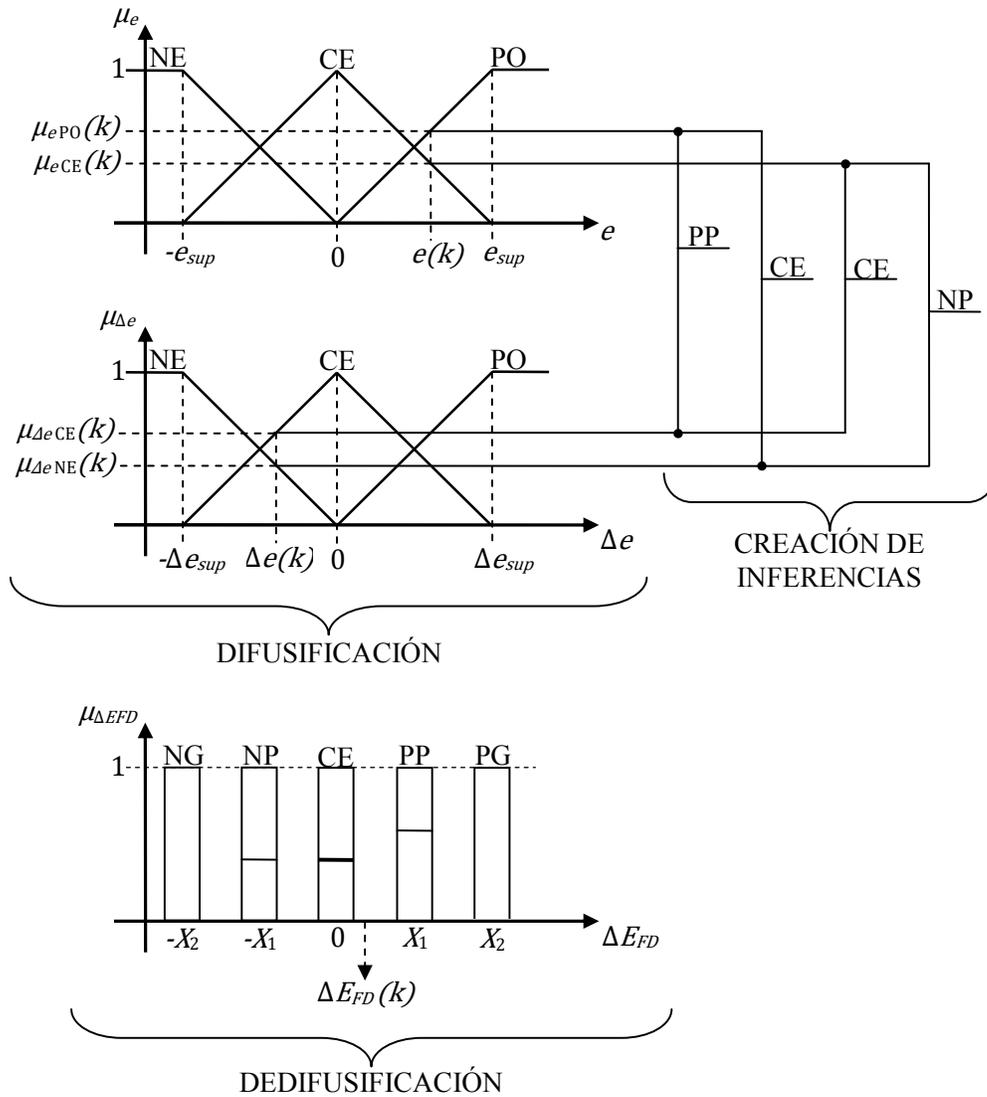


Figura 3.25. Representación gráfica de la operación del controlador difuso PI tipo Mamdani propuesto para regular el voltaje del sistema máquina bus infinito bajo prueba.

donde se utilizan tres funciones de membresía triangulares tanto para el error como para el incremento del error, mientras que para la dedifusificación se emplean cinco funciones de membresía tipo barra o singleton; esto por razones de simplicidad.

En el Apéndice D se describe a detalle el funcionamiento de un control difuso, por lo que en esta sección sólo se presentan los elementos necesarios para la aplicación del controlador difuso de la figura 3.25.

Dada la forma y simetría que presentan las tres funciones de membresía triangulares para la difusificación, los grados de membresía con que pertenece el error a los conjuntos difusos negativo, cero y positivo (representados por las funciones de membresía

triangulares NE, CE y PO de la figura 3.25, respectivamente) se calculan por medio del siguiente procedimiento:

Si $e(k) \geq 0$, entonces

$$\mu_{e_{PO}}(k) = \frac{e(k)}{e_{sup}}, \quad (3.22)$$

si $\mu_{e_{PO}}(k) > 1$, entonces $\mu_{e_{PO}}(k) = 1$,

$$\mu_{e_{CE}}(k) = 1 - \mu_{e_{PO}}(k) \quad (3.23)$$

y

$$\mu_{e_{NE}}(k) = 0; \quad (3.24)$$

de otro modo,

$$\mu_{e_{NE}}(k) = \frac{e(k)}{-e_{sup}}, \quad (3.25)$$

si $\mu_{e_{NE}}(k) > 1$, entonces $\mu_{e_{NE}}(k) = 1$,

$$\mu_{e_{CE}}(k) = 1 - \mu_{e_{NE}}(k) \quad (3.26)$$

y

$$\mu_{e_{PO}}(k) = 0. \quad (3.27)$$

Análogamente, este procedimiento se aplica para el incremento del error.

La base de reglas difusas que se utiliza para el controlador difuso propuesto se muestra en la tabla 3.5, donde las acciones de control son:

- NG = Acción Negativa Grande.
- NP = Acción Negativa Pequeña.
- CE = Acción Cero.
- PP = Acción Positiva Pequeña.
- PG = Acción Positiva Grande.

Tabla 3.5. Base de reglas para el controlador difuso.

| $\Delta e \backslash e$ | Negativo (NE) | Cero (CE) | Positivo (PO) |
|-------------------------|---------------|-----------|---------------|
| Negativo (NE) | NG | NP | CE |
| Cero (CE) | NP | CE | PP |
| Positivo (PO) | CE | PP | PG |

El valor de cada acción se infiere por medio de los operadores difusos mín y máx [14] de acuerdo a las siguientes ecuaciones:

$$NG(k) = \min(\mu_{e_{NE}}(k), \mu_{\Delta e_{NE}}(k)), \quad (3.28)$$

$$NP(k) = \max(\min(\mu_{e_{CE}}(k), \mu_{\Delta e_{NE}}(k)), \min(\mu_{e_{NE}}(k), \mu_{\Delta e_{CE}}(k))), \quad (3.29)$$

$$CE(k) = \max\left(\begin{array}{l} \min(\mu_{e_{PO}}(k), \mu_{\Delta e_{NE}}(k)), \min(\mu_{e_{CE}}(k), \mu_{\Delta e_{CE}}(k)), \\ \min(\mu_{e_{NE}}(k), \mu_{\Delta e_{PO}}(k)) \end{array}\right), \quad (3.30)$$

$$PP(k) = \max(\min(\mu_{e_{PO}}(k), \mu_{\Delta e_{CE}}(k)), \min(\mu_{e_{CE}}(k), \mu_{\Delta e_{PO}}(k))) \quad (3.31)$$

y

$$PG(k) = \min(\mu_{e_{PO}}(k), \mu_{\Delta e_{PO}}(k)). \quad (3.32)$$

Finalmente, en la etapa de dedifusificación se determina cuantitativamente la variable de salida del control difuso en cuestión. El método de dedifusificación empleado es el centroide o centro de gravedad para funciones de membresía tipo barra, por lo que la salida de este controlador está dada por

$$\Delta E_{FD}(k) = \frac{X_2 PG(k) + X_1 PP(k) - X_1 NP(k) - X_2 NG(k)}{PG(k) + PP(k) + CE(k) + NP(k) + NG(k)}, \quad (3.33)$$

donde X_1 y X_2 son los pesos de las funciones de membresía tipo barra.

3.5.2 Sintonización del Controlador Difuso Seleccionado a través de la Arquitectura ARTMAP Difusa

En general, un controlador difuso tipo Mamdani se sintoniza por medio de la manipulación de los parámetros que definen las funciones de membresía utilizadas en la etapa de dedifusificación y, de ser necesario, en la etapa de difusificación. En el caso del controlador difuso propuesto de la figura 3.25, los únicos parámetros por definir en la etapa de difusificación son e_{sup} y Δe_{sup} . Los valores asignados para estos parámetros son $e_{sup} = e_M$ (el error máximo global) y $\Delta e_{sup} = 0.001$. De este modo, en el procedimiento para calcular los grados de membresía del error de la Sección 3.5.1 se tiene que

$$\mu_{e_{PO}}(k) = \frac{e(k)}{e_M} \quad (3.34)$$

para $e \geq 0$, y

$$\mu_{e_{NE}}(k) = \frac{e(k)}{-e_M} \quad (3.35)$$

para $e < 0$. Para el cálculo de los grados de membresía del incremento del error se tiene que

$$\mu_{\Delta e_{PO}}(k) = 1000 \Delta e(k) \quad (3.36)$$

para $\Delta e(k) \geq 0$, y

$$\mu_{\Delta e_{NE}}(k) = -1000\Delta e(k) \quad (3.37)$$

para $\Delta e(k) < 0$.

Consecuentemente, la sintonización del controlador difuso propuesto se realiza mediante la manipulación de los valores de los pesos X_1 y X_2 de las funciones de membresía utilizadas para la dedifusificación. La arquitectura FAM proporciona el valor de dE_{FD} a través del bloque de acondicionamiento inverso (figura 3.24b), el cual realiza la misma función de la ecuación (3.18), para después calcular el valor de los pesos X_1 y X_2 en el bloque de determinación de estos pesos por medio de

$$X_1(k) = \frac{0.025}{dE_{FDM}} |dE_{FD}(k)| + 0.075 \quad (3.38)$$

$$X_2(k) = 2X_1(k), \quad (3.39)$$

donde $|\cdot|$ es el valor absoluto. El peso X_2 es simplemente el doble de X_1 por razones de simetría con respecto a la acción CE. Las ecuaciones (3.38) y (3.39) se deducen después de numerosas pruebas, donde la constante 0.075 es el valor adecuado para el peso X_1 que sintoniza el controlador difuso en cuestión operando sin la arquitectura FAM; además, esta constante tiene la finalidad de evitar que el valor de los pesos sea nulo en un determinado instante. De esta manera, el procedimiento de dedifusificación de la ecuación (3.33) se convierte en

$$\Delta E_{FD}(k) = \frac{X_2(k)PG(k) + X_1(k)PP(k) - X_1(k)NP(k) - X_2(k)NG(k)}{PG(k) + PP(k) + CE(k) + NP(k) + NG(k)}, \quad (3.40)$$

donde los pesos varían en diferentes instantes de integración.

De acuerdo con las ecuaciones (3.18) y (3.38), el peso X_1 aumenta conforme el primer elemento de los prototipos almacenados en la red $FART_b$ se aleja de 0.5; es decir, a medida que aumenta la diferencia entre el voltaje de campo vigente y el inicial (de aquí la conveniencia de utilizar el acondicionamiento de las ecuaciones (3.2) y (3.3) para las variables de entrada). Con esto, los pesos X_1 y X_2 pueden adoptar tantos valores como categorías definidas en la red $FART_b$ dentro de los intervalos $[0.075, 0.1]$ y $[0.15, 0.2]$, respectivamente, lo que permite que el controlador difuso se adapte a cambios repentinos en la dinámica de todo el sistema.

3.5.3 Configuración Resultante de la Arquitectura ARTMAP Difusa

Durante la fase de producción, se requiere que los pesos puedan adoptar la mayor cantidad de valores dentro de sus respectivos intervalos para disminuir lo mayormente posible las oscilaciones en todo el sistema una vez que se presenta un evento como la falla de corto circuito en el bus descrita en la Sección 3.3.1; pero también es conveniente que el tiempo de cómputo no se incremente demasiado. Por lo tanto, la arquitectura FAM se entrena con las mismas diez condiciones iniciales de la tabla 3.2, puesto que con esto se consigue un balance entre ambos aspectos.

La configuración resultante de la arquitectura FAM para su aplicación en la sintonización del controlador difuso seleccionado es la que se muestra en la figura 3.26,

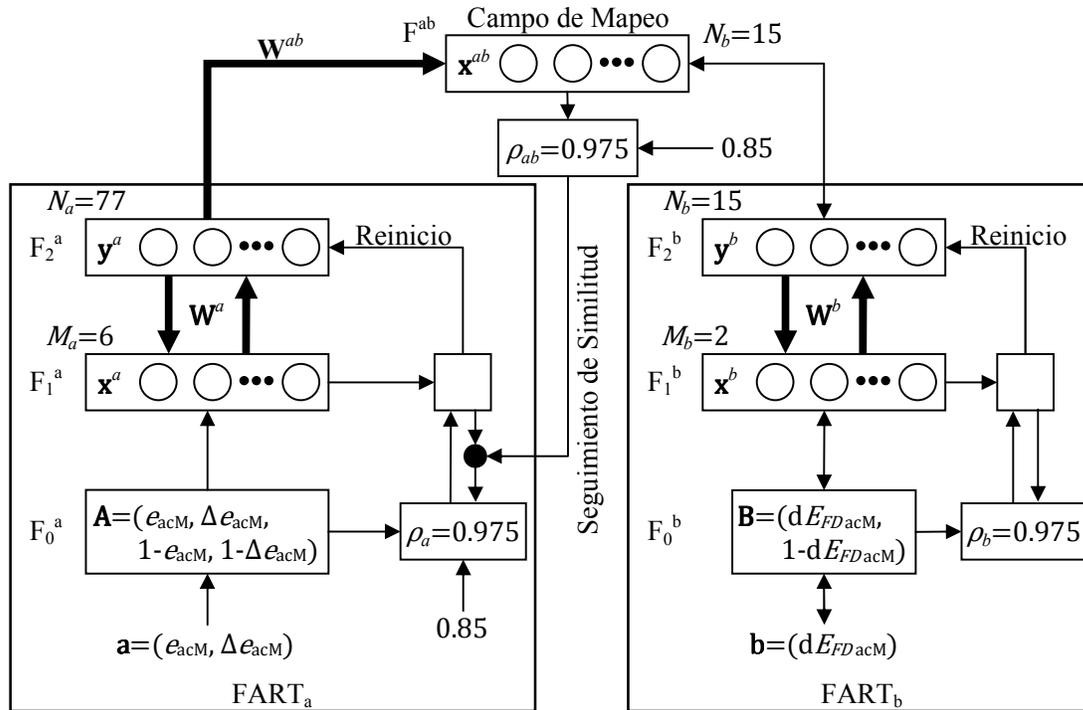


Figura 3.26. Configuración resultante de la arquitectura FAM para su aplicación en la sintonización del controlador difuso seleccionado.

donde se muestran las variables de entrada seleccionadas, el ajuste de los parámetros de vigilancia tanto para la fase de entrenamiento como para la fase de producción (estos últimos fuera de los bloques), y el número de categorías generadas en ambas redes FART. En el Apéndice C se presentan las matrices \mathbf{W}^a , \mathbf{W}^b y \mathbf{W}^{ab} de la arquitectura FAM resultantes del entrenamiento de la figura 3.24a.

En resumen, la arquitectura FAM identifica el estado en que se encuentra el sistema en base al entrenamiento realizado, después infiere el valor adecuado de los pesos X_1 y X_2 por medio de los bloques de acondicionamiento inverso y determinación de los pesos X_1 y X_2 (figura 3.24b) y, finalmente, el controlador difuso regula el voltaje del sistema máquina bus infinito bajo prueba.

3.6 DESARROLLO DEL PROGRAMA DE SIMULACIÓN

El núcleo del presente trabajo de tesis es el desarrollo de un programa computacional para simular el sistema máquina bus infinito bajo prueba con regulación de voltaje a través de la arquitectura FAM, ya sea como actuando como regulador o como sintonizador del controlador difuso de la Sección 3.5.1. A continuación se presenta el programa realizado en forma de diagrama de flujo; el código completo del programa se presenta en el Apéndice C.

En la figura 3.27 se muestra el diagrama de flujo correspondiente al programa principal.

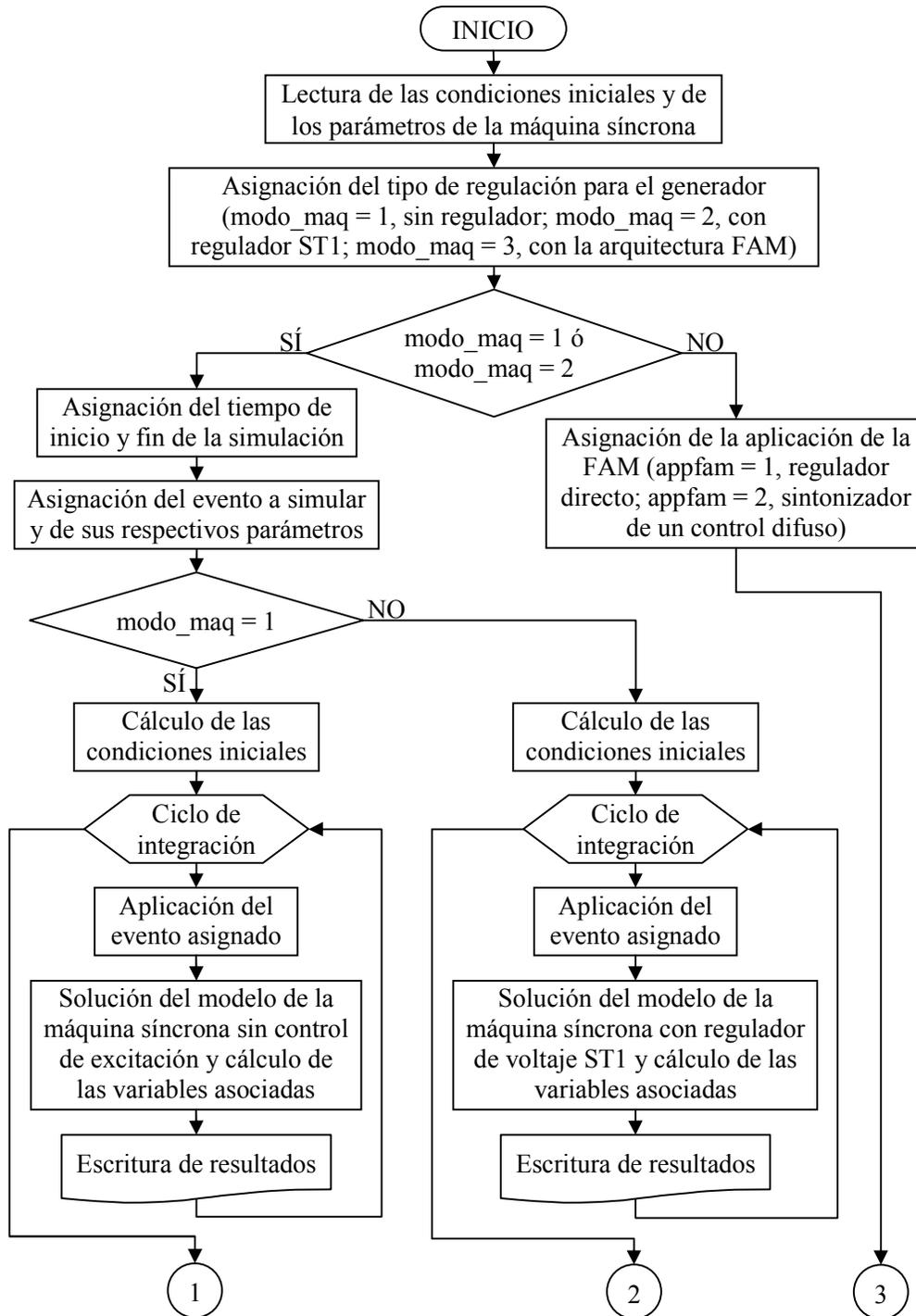


Figura 3.27. Diagrama de flujo del programa principal desarrollado (1ª parte).

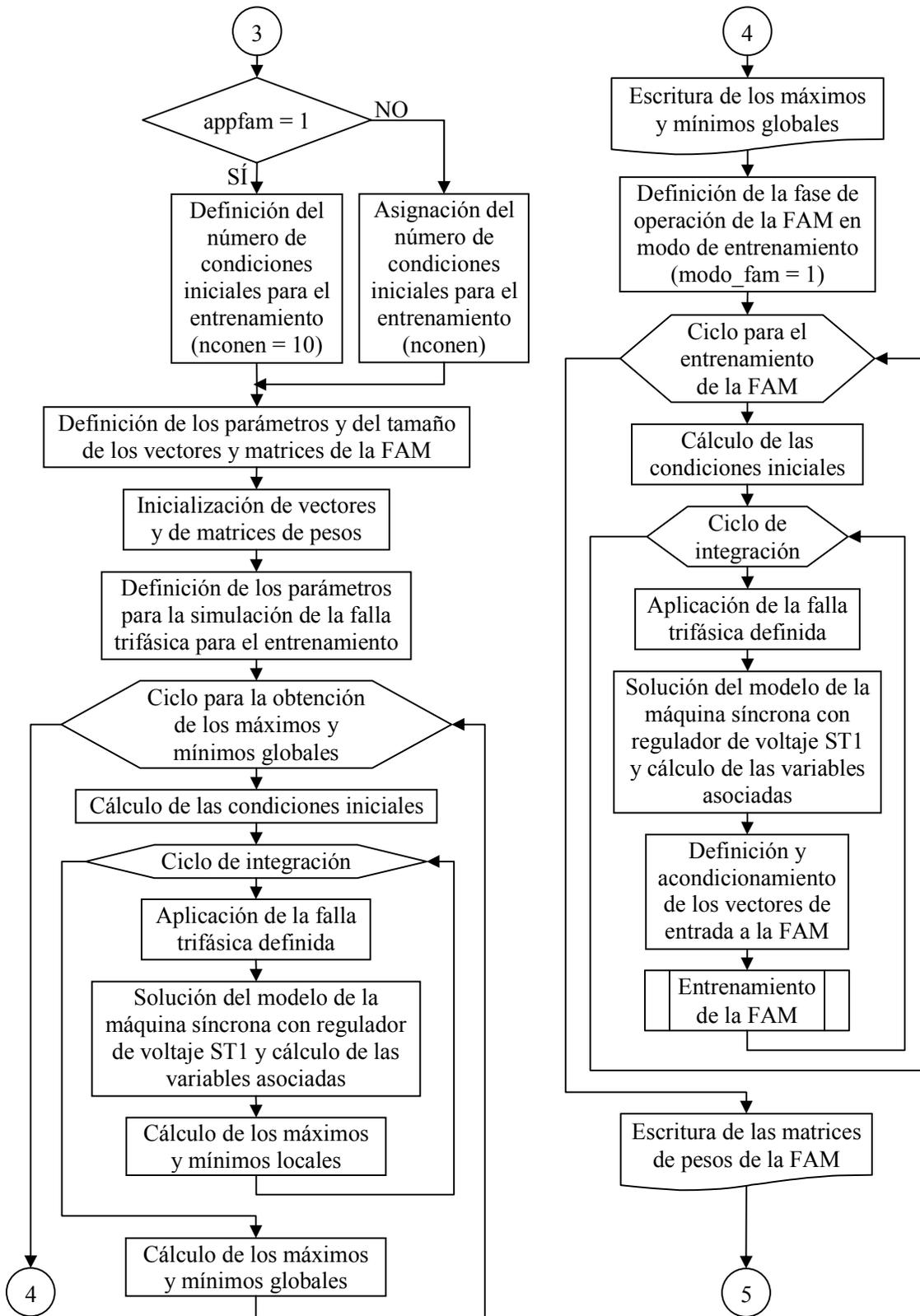


Figura 3.27. Diagrama de flujo del programa principal desarrollado (2ª parte).

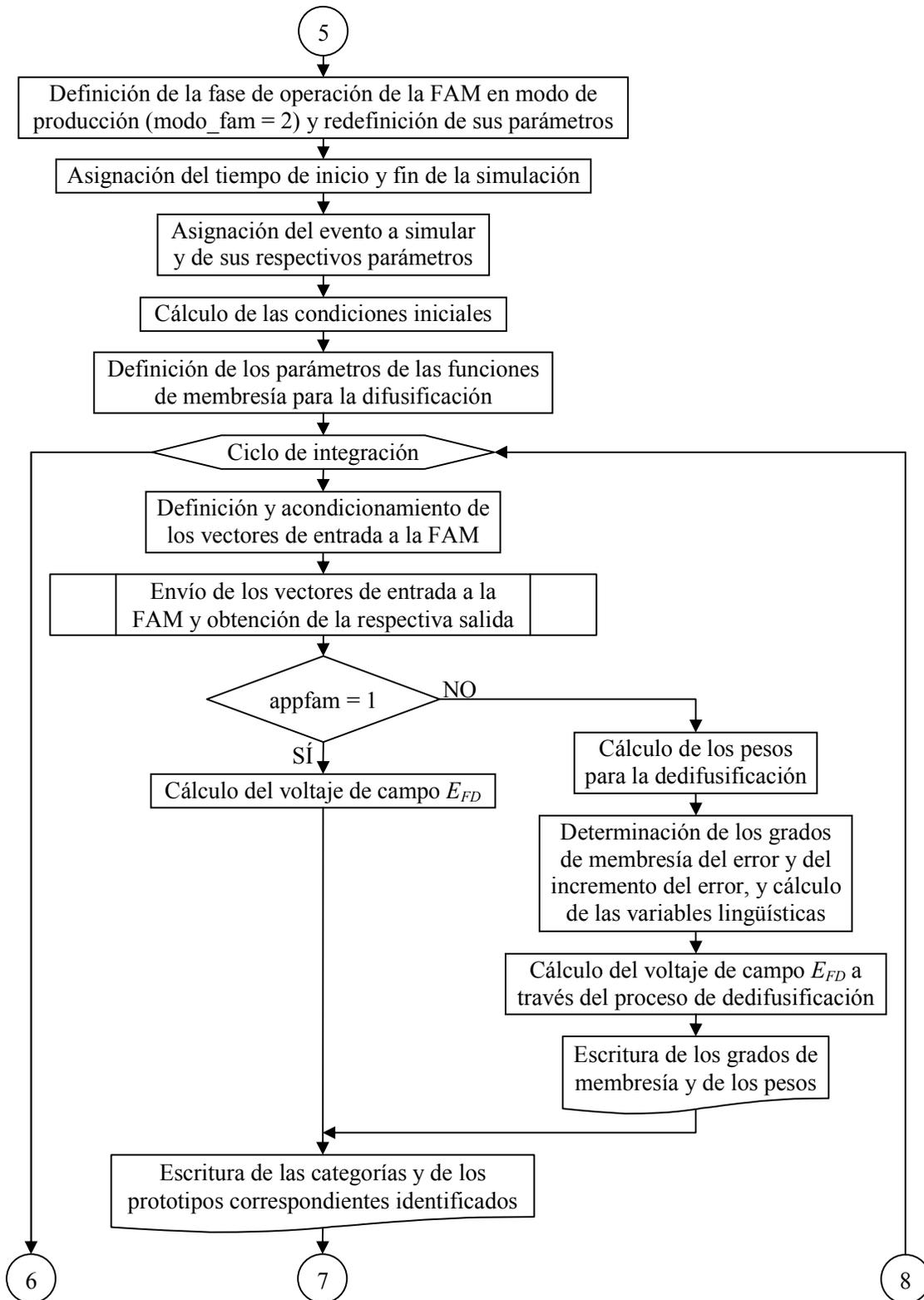


Figura 3.27. Diagrama de flujo del programa principal desarrollado (3ª parte).

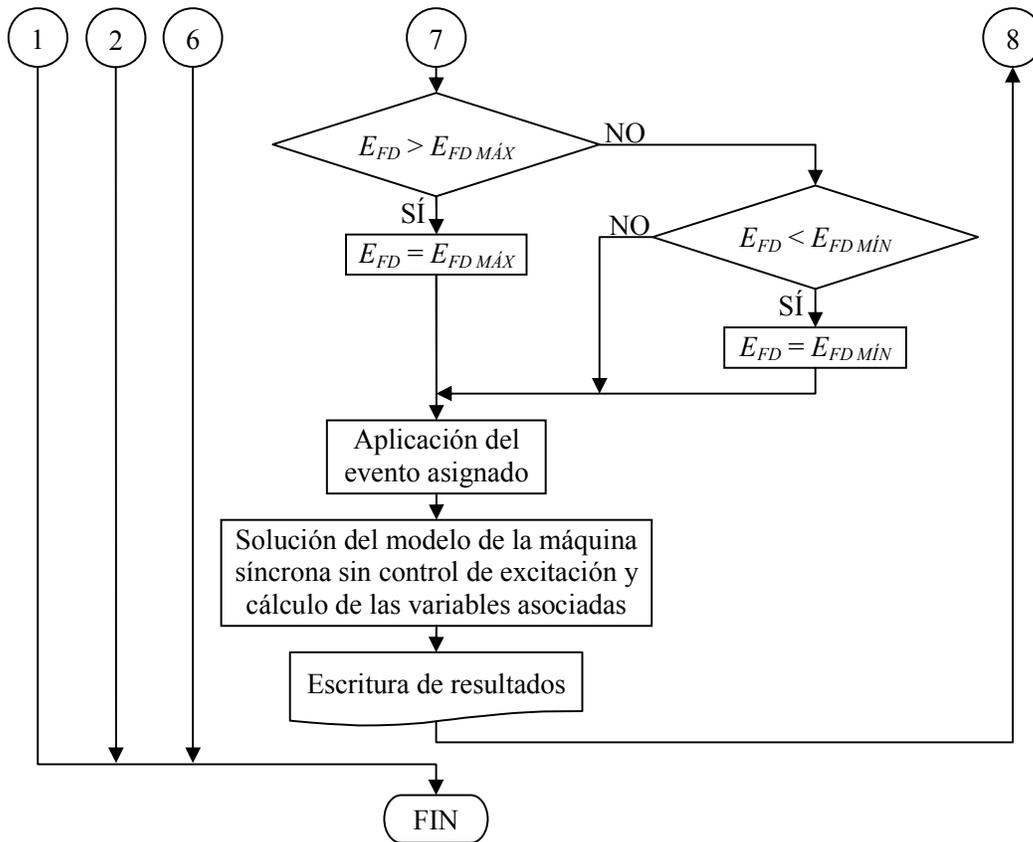


Figura 3.27. Diagrama de flujo del programa principal desarrollado (4ª parte).

En la figura 3.27 (1ª parte) se aprecia que el programa principal ofrece tres opciones principales, las cuales ofrecen la posibilidad de simular el sistema sin regulador, con regulador ST1 o con la arquitectura FAM. Lo sucesivo a las dos primeras opciones es muy similar, únicamente cambia el modelo del generador por el aumento de ecuaciones que conlleva un regulador de voltaje convencional.

La utilización de la arquitectura FAM la representa la tercera opción, la cual se subdivide a su vez en dos opciones: regulador directo y sintonizador de un controlador difuso. Con la primera de estas opciones se tiene la posibilidad de ingresar el número de condiciones iniciales para el entrenamiento de la arquitectura FAM en tiempo de ejecución, y el voltaje de campo se calcula directamente por medio de las ecuaciones (3.18) y (3.19). En la opción de sintonizador de un controlador difuso no es conveniente tener la posibilidad de asignar (ingresar dinámicamente) el número de condiciones iniciales para el entrenamiento, por lo que este valor se fija en 10; en cuanto al voltaje de campo, éste se determina por medio del proceso de dedifusificación del controlador difuso de la figura 3.25 previamente sintonizado por la arquitectura FAM.

Los bloques con franja lateral de la figura 3.27 (2ª y 3ª parte) representan el llamado al algoritmo de la arquitectura FAM en las fases de entrenamiento y producción, respectivamente. En la figura 3.28 se muestra el diagrama de flujo correspondiente a la arquitectura FAM como regulador y como sintonizador del control difuso seleccionado.

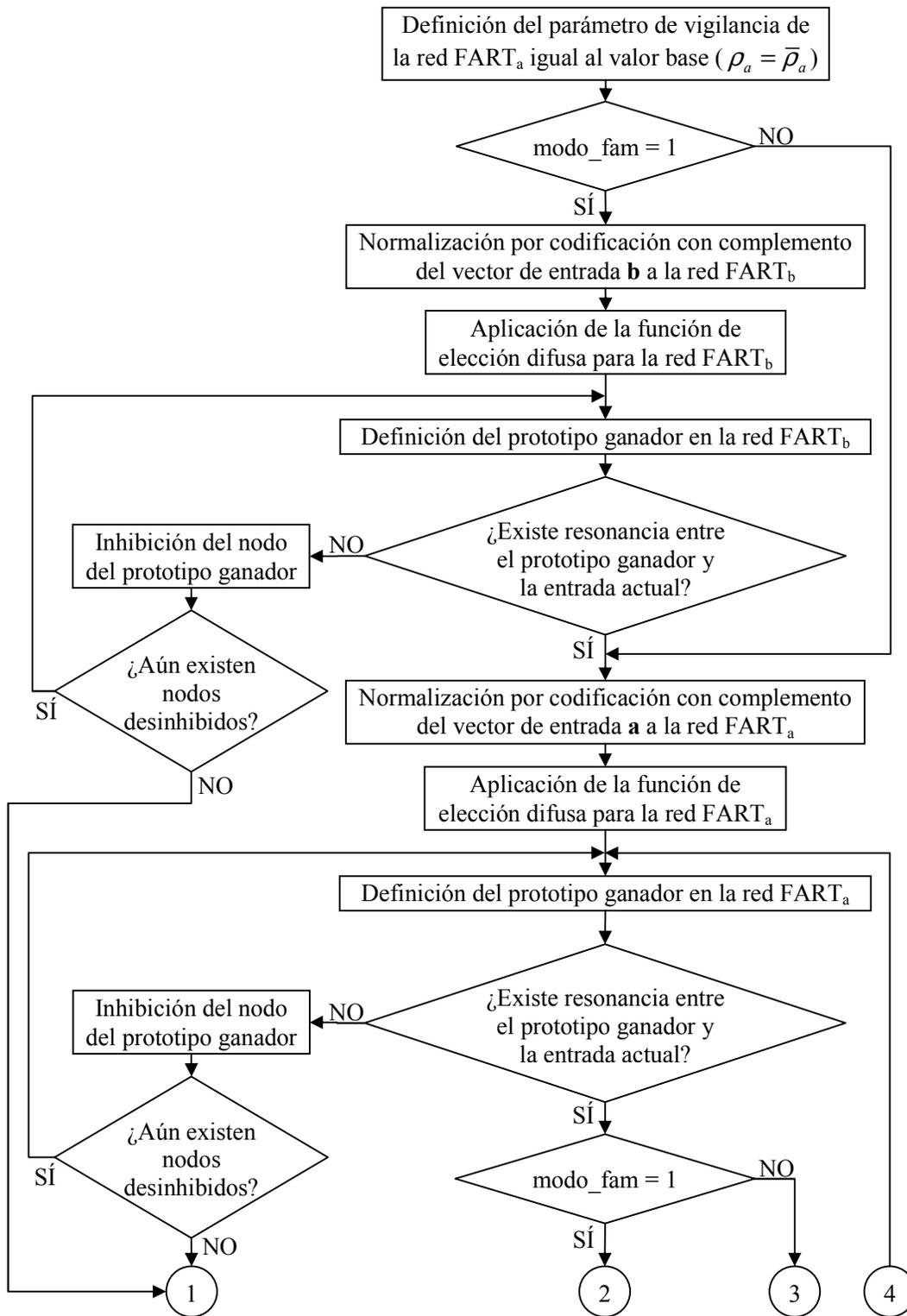


Figura 3.28. Diagrama de flujo de la aplicación de la arquitectura FAM en la regulación de voltaje (1ª parte).

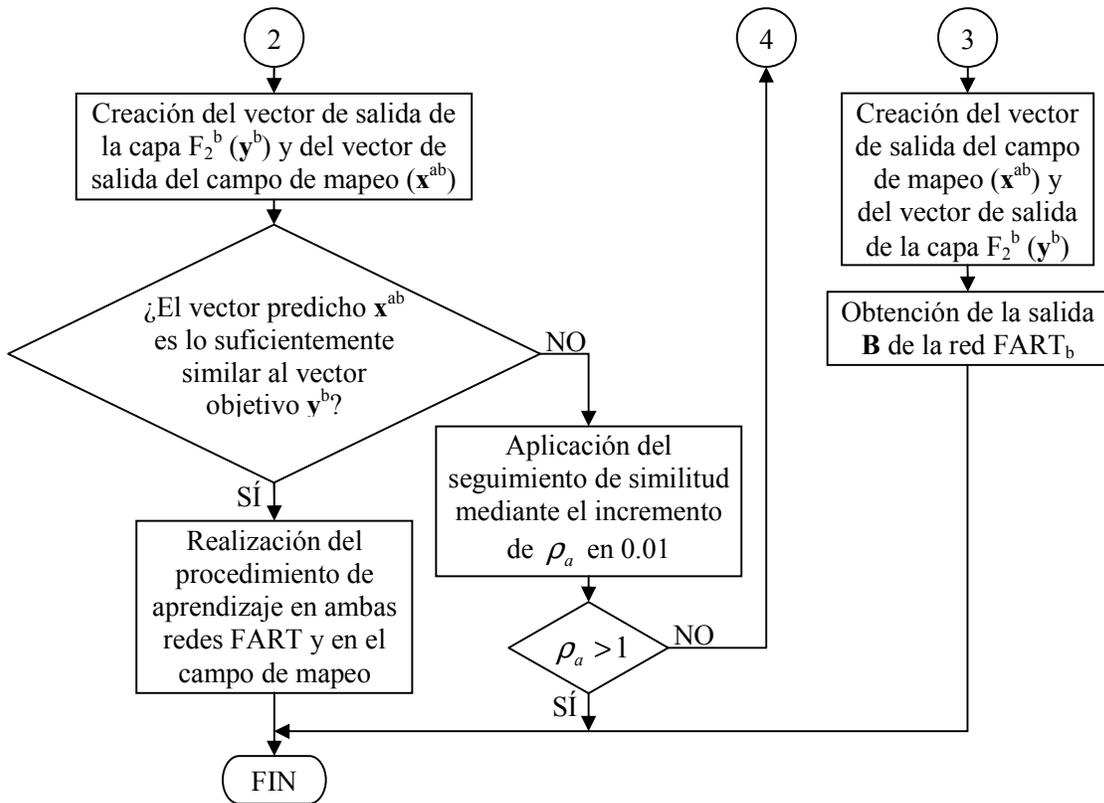


Figura 3.28. Diagrama de flujo de la aplicación de la arquitectura FAM en la regulación de voltaje (2ª parte).

Los análisis de los comportamientos de ambos reguladores se pueden realizar de forma más íntegra si se realizan pruebas con diferentes eventos o disturbios. Las pruebas contempladas en el programa principal se pueden aplicar con los tres tipos de regulación incluidos, lo que refleja una gran versatilidad del programa desarrollado; ya que además, la mayoría de los parámetros para la simulación de los eventos o disturbios se ingresan en tiempo de ejecución.

CAPÍTULO 4

PRUEBAS Y RESULTADOS

4.1 INTRODUCCIÓN

En este capítulo se presentan los resultados que se obtienen aplicando el procedimiento descrito en la Sección 3.5 para regular el voltaje en terminales del sistema máquina bus infinito bajo prueba, y se comparan con los respectivos que ofrece el regulador ST1. Estos resultados constan de las respuestas para las mismas variables contempladas en la Sección 3.4; es decir, el voltaje en terminales, el índice de comportamiento del voltaje en terminales, el voltaje de campo, el ángulo de carga y las potencias activa y reactiva. Además de estas variables, también es conveniente visualizar la variación de los pesos X_1 y X_2 que dependen de la salida de la arquitectura FAM hacia el control difuso, por lo que estas respuestas también se presentan en todos los casos.

Con la finalidad de verificar el funcionamiento del control propuesto de una forma más general, durante la fase de producción se aplican diversas pruebas al sistema resultante de la figura 3.24b. Estas pruebas incluyen la simulación de los siguientes eventos:

- Sin evento alguno (sólo seguir la referencia inicial).
- Fallas de corto circuito trifásicas sin apertura de línea.
- Fallas de corto circuito trifásicas con apertura de línea.
- Cambios de carga.
- Cambios de referencia.

A excepción de los cambios de carga, estos eventos se simulan partiendo de dos diferentes condiciones iniciales con distintas características. En la tabla 4.1 se muestran las condiciones iniciales seleccionadas para realización de las pruebas indicadas, cuya numeración, al igual que en las tablas 3.2 y 3.3, corresponde al orden presentado en el Apéndice B. Las condiciones iniciales elegidas son la No. 25 y la No. 30, debido a que la primera es de las menos críticas y la que mayor cantidad de potencia reactiva genera de todo el conjunto de condiciones iniciales del Apéndice B; por su parte, la No.30 es la más crítica de ese mismo conjunto.

Tabla 4.1. Condiciones iniciales utilizadas para las distintas pruebas.

| No. de Condición Inicial | Voltaje en Terminales (p.u.) | Ángulo de Carga (grados) | Potencia Activa (p.u.) | Potencia Reactiva (p.u.) |
|--------------------------|------------------------------|--------------------------|------------------------|--------------------------|
| 25 | 1.0594970 | 39.0086232 | 0.7149498 | 0.4203133 |
| 30 | 0.9904293 | 79.4581517 | 0.9592349 | -0.2236688 |

4.2 RESPUESTAS SIN EVENTO

En esta sección se presentan las respuestas obtenidas del sistema bajo prueba sin la presencia de evento alguno para las dos condiciones iniciales de la tabla 4.1; es decir, estas respuestas muestran únicamente la forma en que el control propuesto (CP) sigue la referencia inicial en comparación con el regulador ST1. En ambos casos, no se muestra la respuesta del índice de comportamiento del voltaje en terminales, debido a que su rango de valores es demasiado pequeño.

4.2.1 Resultados para una Condición Inicial No Crítica

En las figuras 4.1-4.6 se muestran los resultados obtenidos para la condición inicial no crítica correspondiente a la No. 25 de la tabla 4.1.

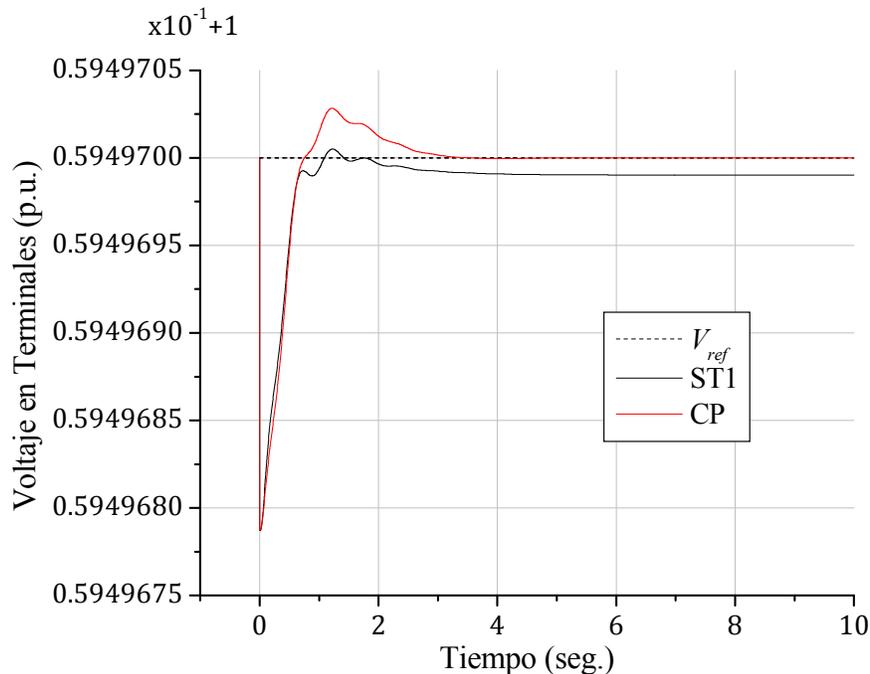


Figura 4.1. Voltaje en terminales para la condición inicial No. 25 sin evento.

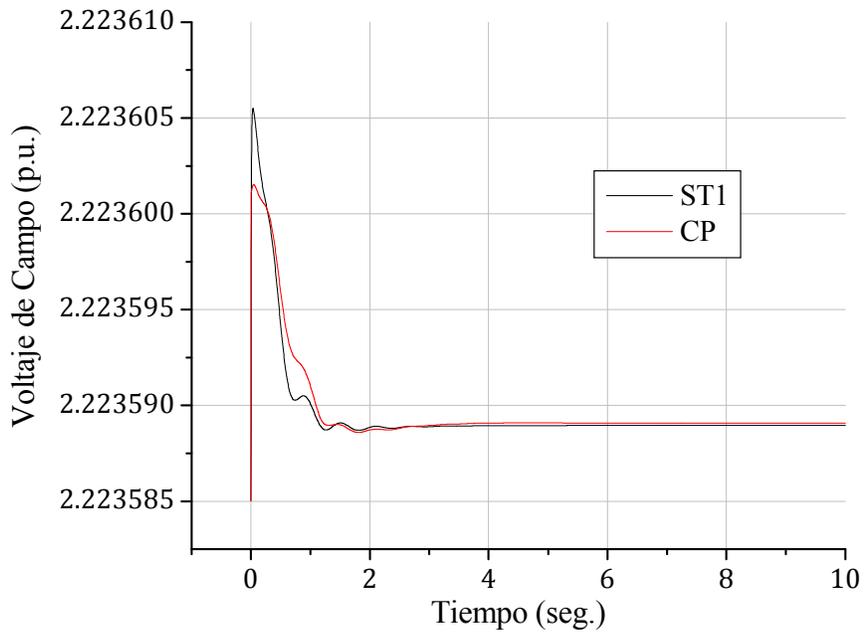


Figura 4.2. Voltaje de campo para la condición inicial No. 25 sin evento.

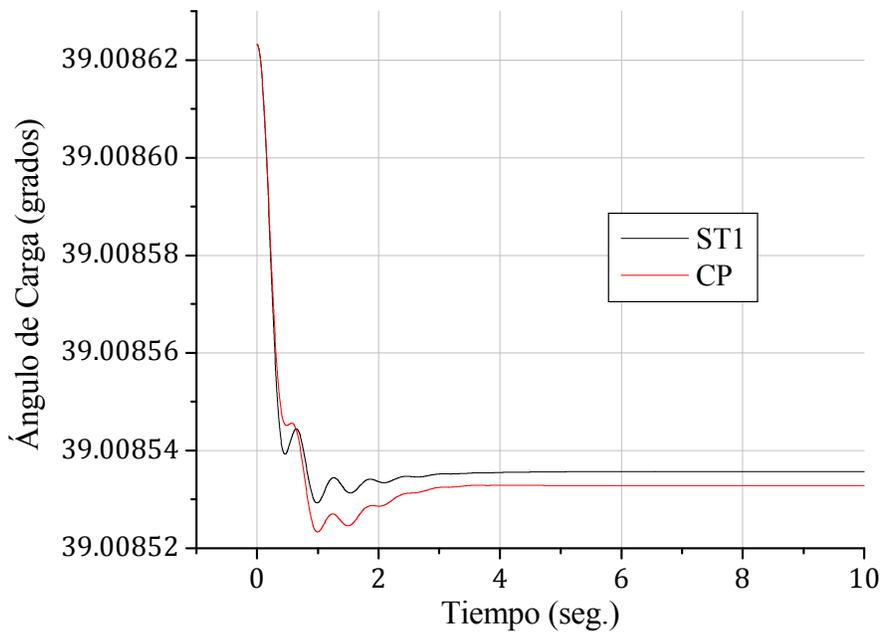


Figura 4.3. Ángulo de carga para la condición inicial No. 25 sin evento.

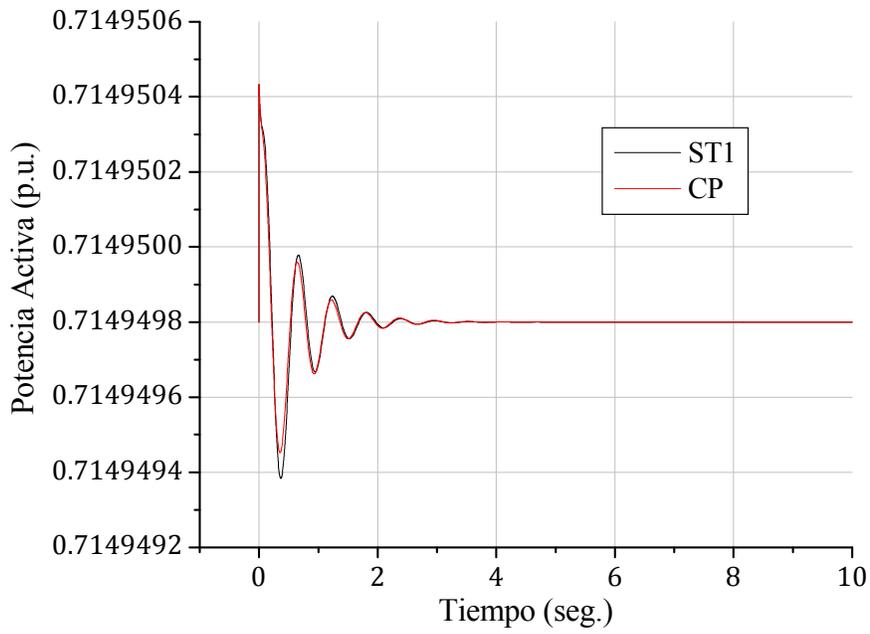


Figura 4.4. Potencia activa para la condición inicial No. 25 sin evento.

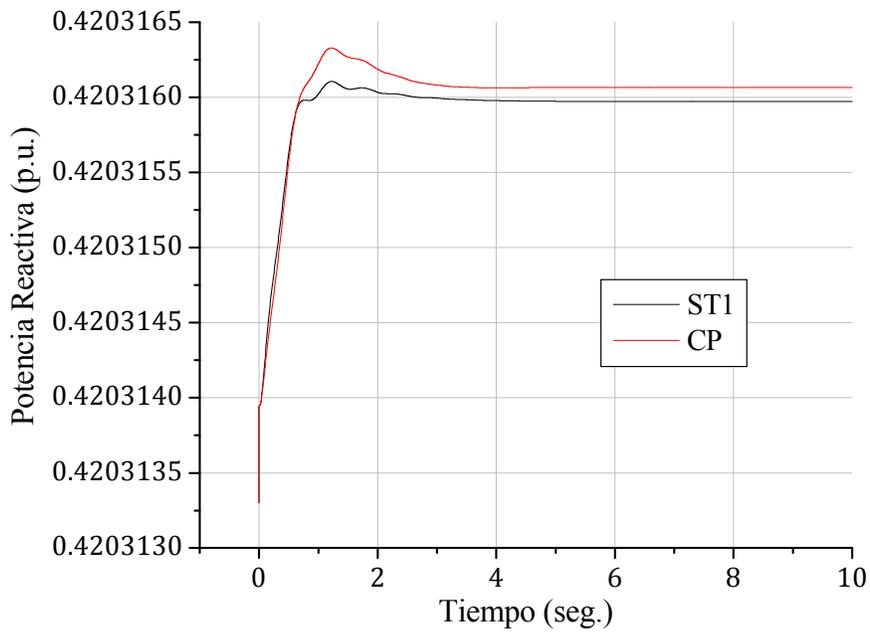


Figura 4.5. Potencia reactiva para la condición inicial No. 25 sin evento.

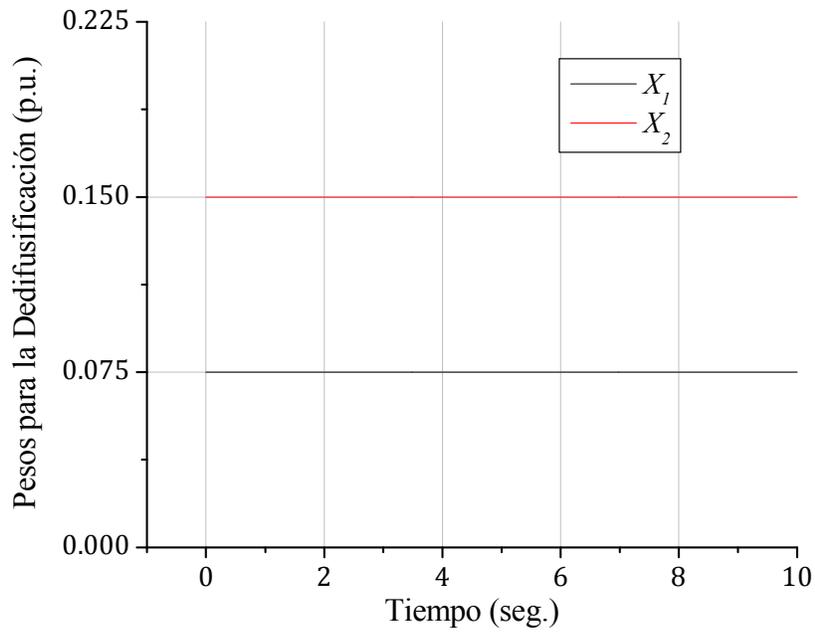


Figura 4.6. Pesos X_1 y X_2 para la condición inicial No. 25 sin evento.

4.2.2 Resultados para una Condición Inicial Crítica

En las figuras 4.7-4.12 se muestran los resultados obtenidos para la condición inicial crítica (No. 30) de la tabla 4.1.

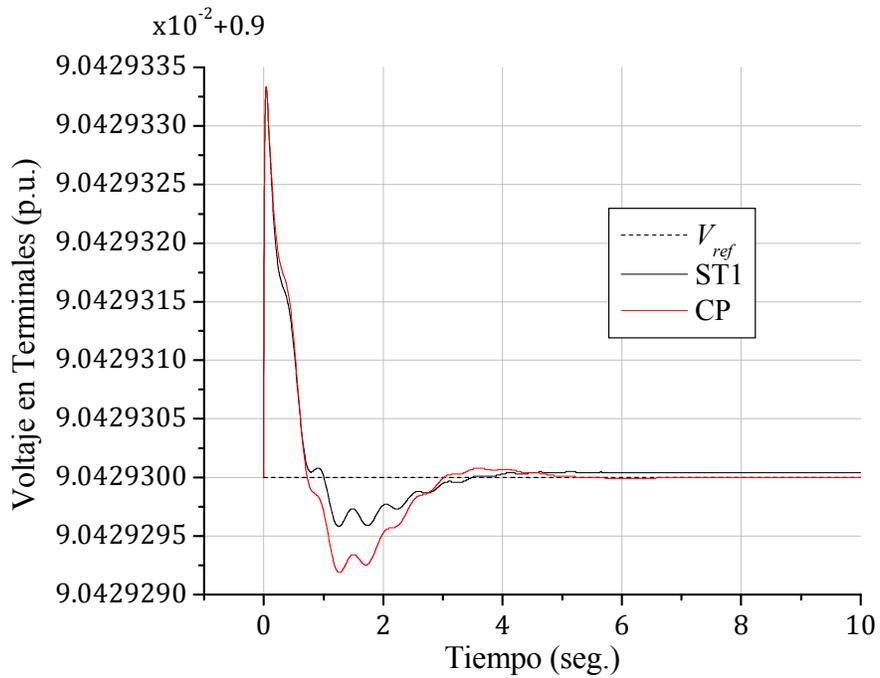


Figura 4.7. Voltaje en terminales para la condición inicial No. 30 sin evento.

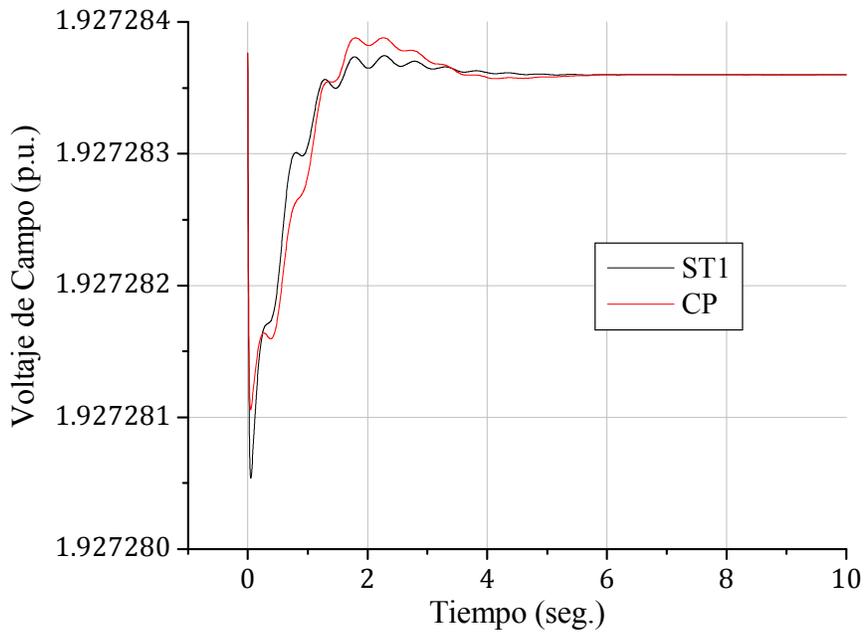


Figura 4.8. Voltaje de campo para la condición inicial No. 30 sin evento.

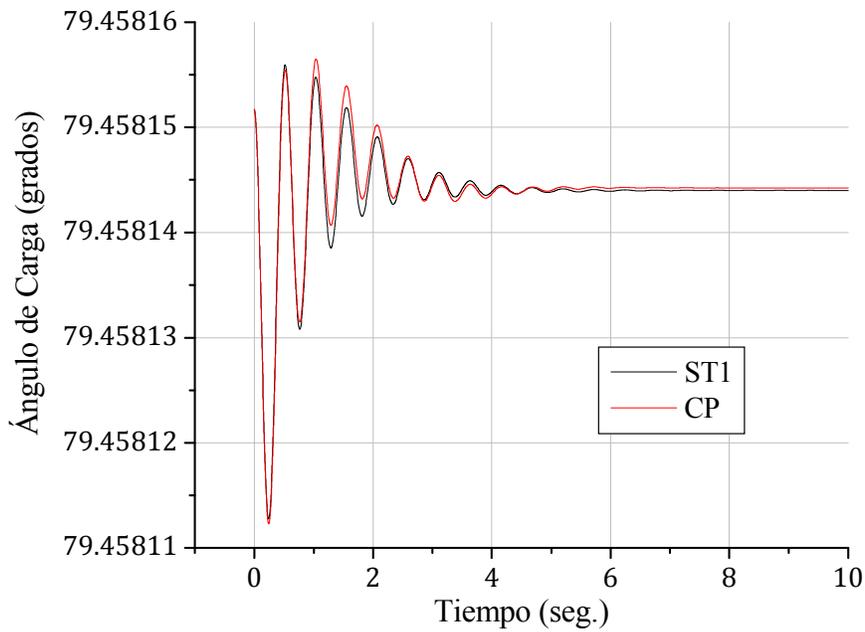


Figura 4.9. Ángulo de carga para la condición inicial No. 30 sin evento.

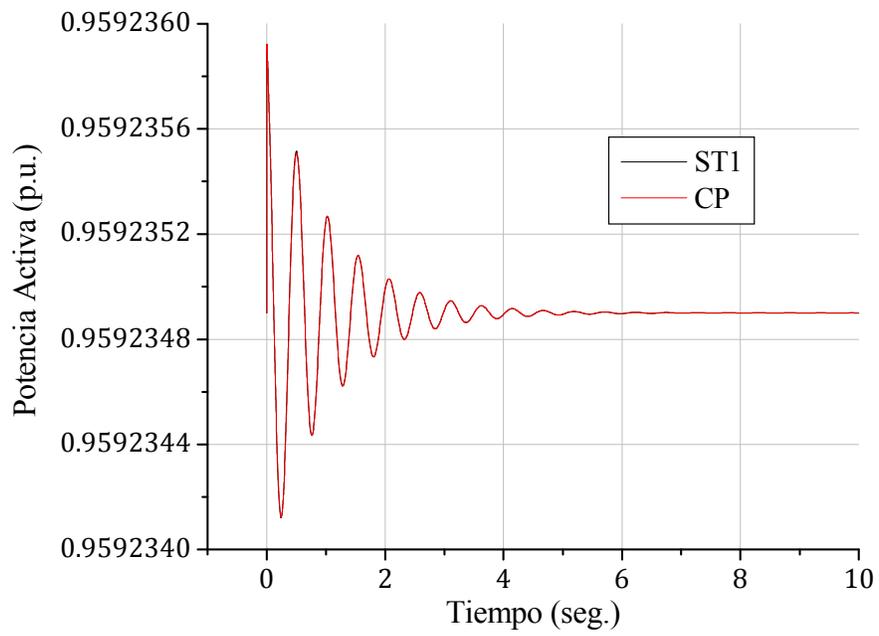


Figura 4.10. Potencia activa para la condición inicial No. 30 sin evento.

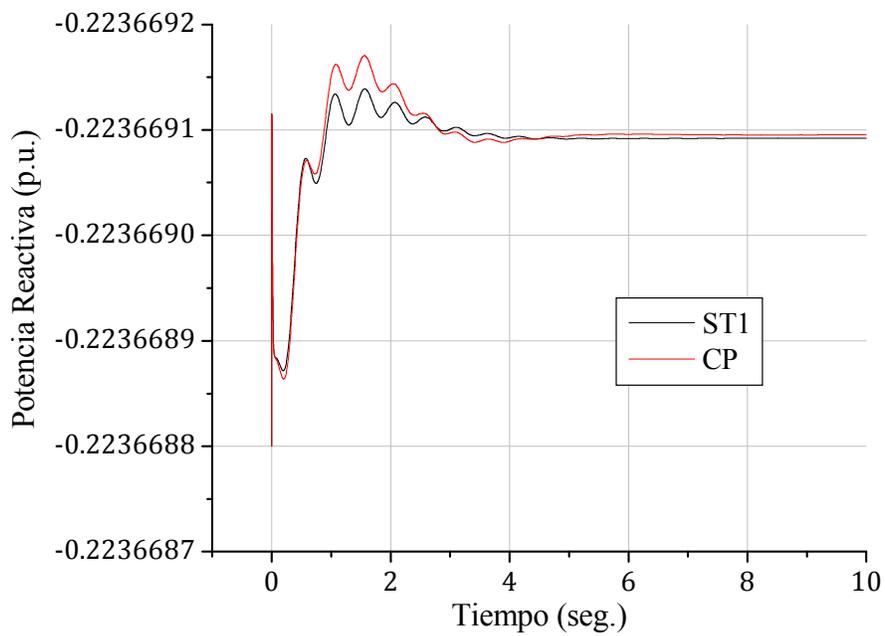


Figura 4.11. Potencia reactiva para la condición inicial No. 30 sin evento.

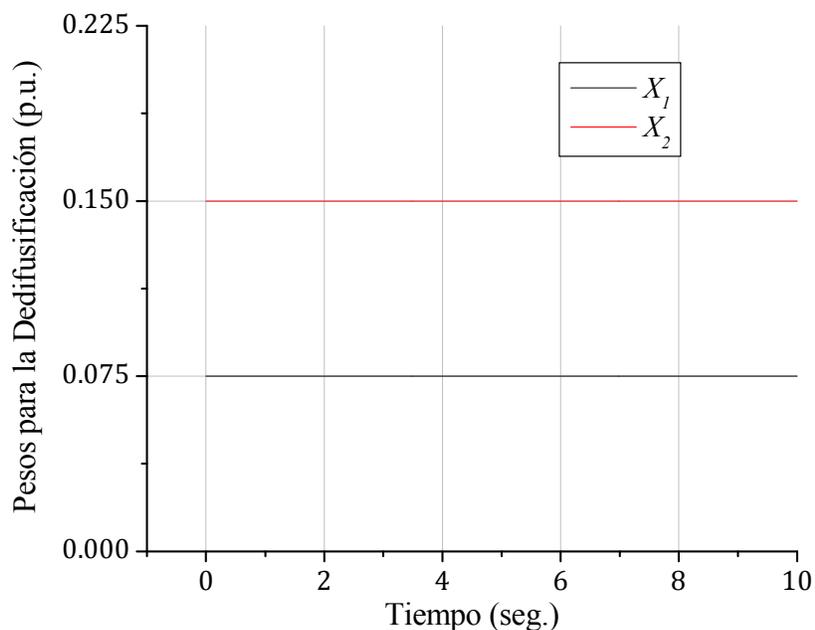


Figura 4.12. Pesos X_1 y X_2 para la condición inicial No. 30 sin evento.

4.2.3 Análisis de los Resultados

En las figuras 4.1 y 4.7 se puede apreciar que la respuesta que proporciona el regulador ST1 presenta un error en estado estacionario, mientras que la respuesta obtenida con el controlador propuesto elimina este error. En la tabla 4.2 se muestran estas características para ambas condiciones iniciales simuladas, donde se puede observar que el regulador ST1 tarda más que el controlador propuesto en alcanzar el valor definitivo en estado estacionario para la condición inicial No. 25; sin embargo, para la condición inicial No. 30, el controlador propuesto tarda más que el ST1 en alcanzar el respectivo valor definitivo.

Tabla 4.2. Errores en estado estacionario con ambos controladores para las dos condiciones iniciales simuladas.

| No. de Condición Inicial | Error en Estado Estacionario | | | |
|--------------------------------|------------------------------|------------|-----------------------|------------|
| | Regulador ST1 | | Controlador Propuesto | |
| | Error | Tiempo | Error | Tiempo |
| 25 | 0.98×10^{-8} | 5.861 seg. | 0.0 | 5.328 seg. |
| 30 | -0.4×10^{-9} | 5.664 seg. | 0.0 | 6.573 seg. |

En las figuras 4.2 y 4.8 se observa que el voltaje de campo proporcionado por ambos controladores es muy similar, salvo que este voltaje presenta oscilaciones más pronunciadas cuando proviene del regulador ST1, sobre todo para la condición inicial no crítica.

Las respuestas correspondientes al resto de las variables (ángulo de carga y potencias activa y reactiva) muestran un comportamiento del sistema bajo prueba muy similar con ambos controladores cuando no se suscita ningún tipo de disturbio.

Las figuras 4.6 y 4.12 muestran que la arquitectura FAM, dada la naturaleza de su entrenamiento, no considera en instante alguno necesaria la modificación de los pesos X_1 y X_2 para el proceso de dedifusificación, lo que significa que el sistema se encuentra en un punto de operación relativamente muy cercano al inicial durante todo el tiempo de simulación.

4.3 RESPUESTAS ANTE FALLAS DE CORTO CIRCUITO TRIFÁSICAS SIN APERTURA DE LÍNEA

En esta sección se presentan las respuestas obtenidas del sistema bajo prueba ante fallas trifásicas para las dos condiciones iniciales de la tabla 4.1.

Puesto que uno de los objetivos es determinar el tiempo crítico de liberación de falla que ofrece el controlador propuesto en comparación con los propios del regulador ST1 para las distintas condiciones iniciales, la liberación de la falla se realiza simplemente restableciendo el voltaje en el bus; es decir, de la misma forma que en la Sección 3.3.1. Por lo tanto, no se contempla el aislamiento de la falla por medio de la apertura de la línea donde se suscita, con lo que la impedancia de la red permanece constante durante todo el tiempo de simulación.

En los resultados de la sección anterior se aprecia que los transitorios son mínimos para ambas condiciones iniciales cuando $t = 2$ seg., por lo que se toma este tiempo como inicio de falla en todos los casos. El tiempo de duración de cada falla es diferente para cada condición inicial, puesto que presentan diferentes características. Estos tiempos se especifican en cada caso.

4.3.1 Resultados para una Condición Inicial No Crítica

En las figuras 4.13-4.19 se muestran los resultados obtenidos para la condición inicial no crítica (No. 25) de la tabla 4.1 ante una falla de 12 ciclos (200 mseg.) de duración.

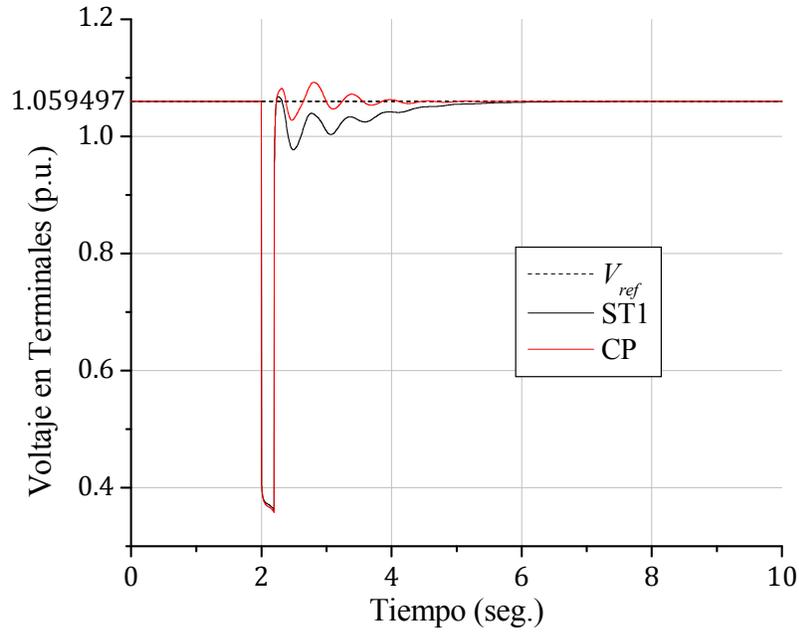


Figura 4.13. Voltaje en terminales para la condición inicial No. 25 ante una falla de 12 ciclos de duración.

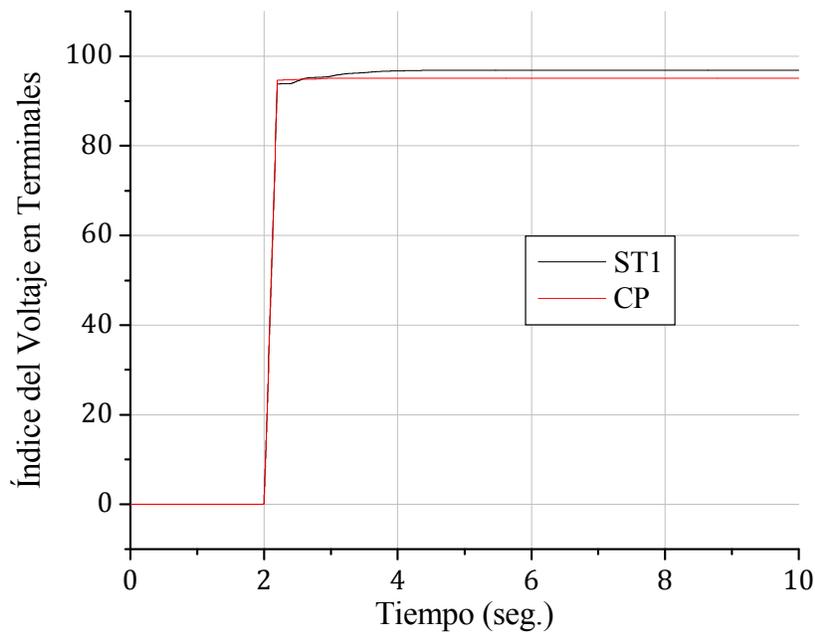


Figura 4.14. Índice de comportamiento del voltaje en terminales para la condición inicial No. 25 ante una falla de 12 ciclos de duración.

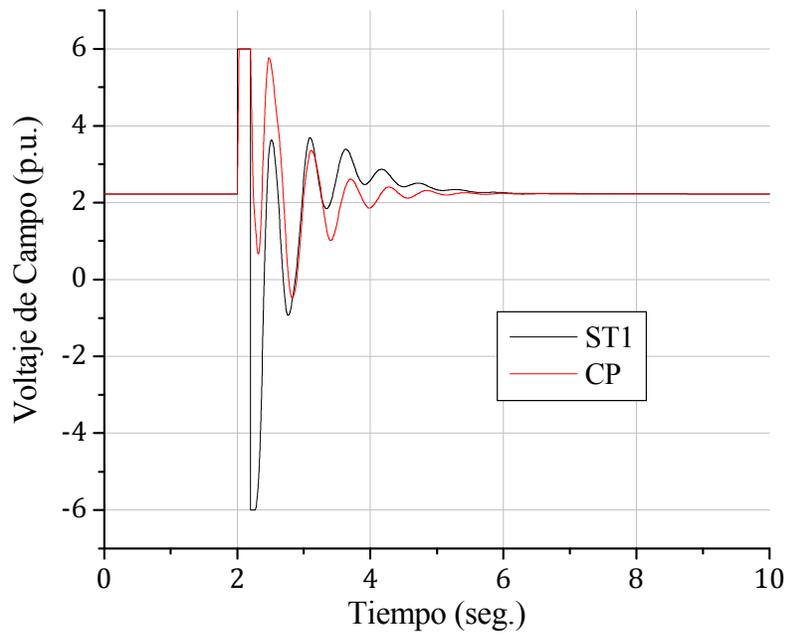


Figura 4.15. Voltaje de campo para la condición inicial No. 25 ante una falla de 12 ciclos de duración.

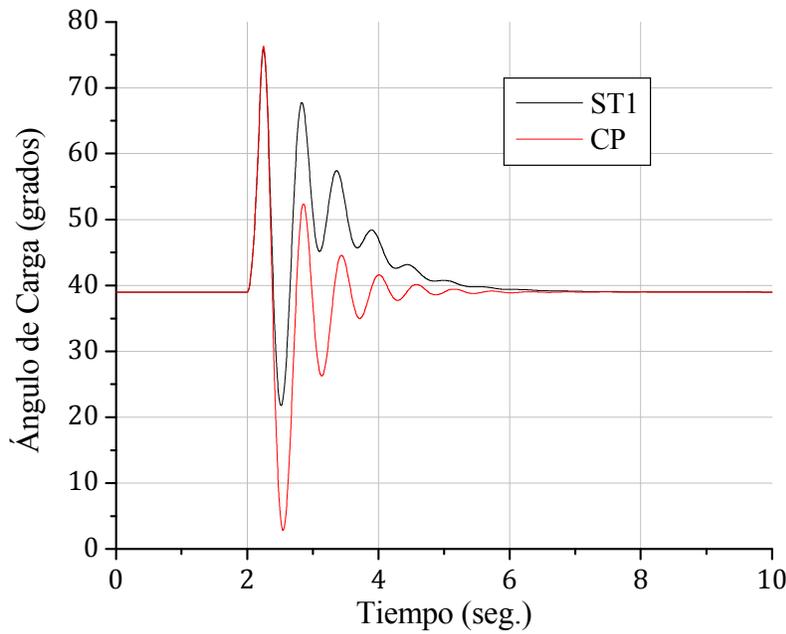


Figura 4.16. Ángulo de carga para la condición inicial No. 25 ante una falla de 12 ciclos de duración.

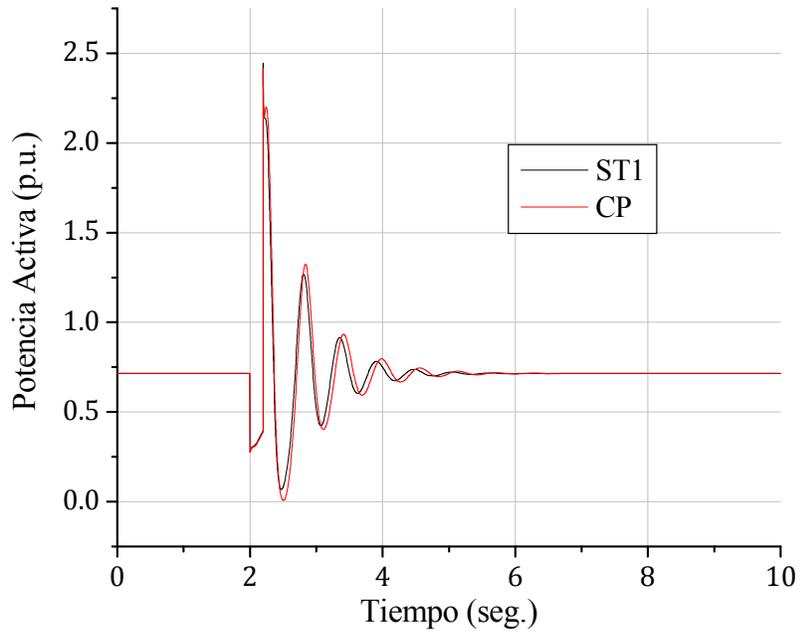


Figura 4.17. Potencia activa para la condición inicial No. 25 ante una falla de 12 ciclos de duración.

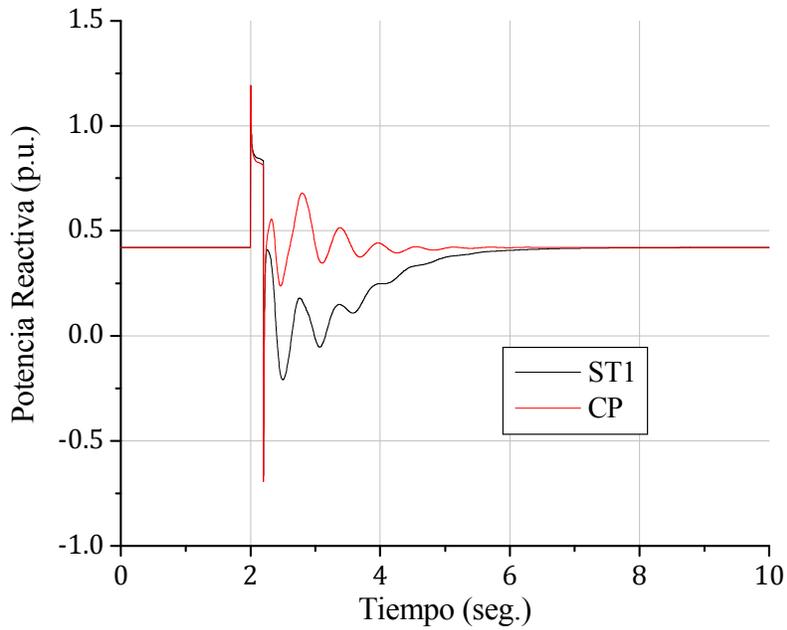


Figura 4.18. Potencia reactiva para la condición inicial No. 25 ante una falla de 12 ciclos de duración.

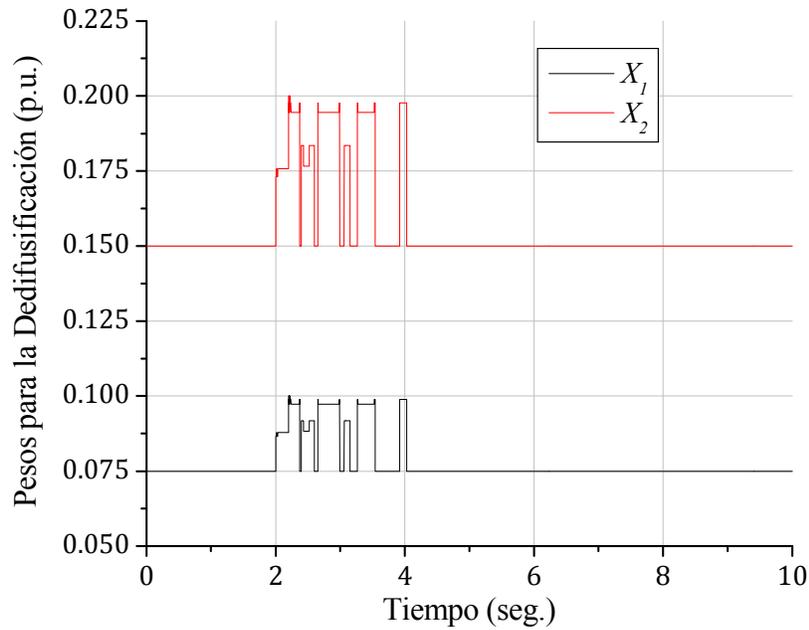


Figura 4.19. Pesos X_1 y X_2 para la condición inicial No. 25 ante una falla de 12 ciclos de duración.

En las figuras 4.20-4.26 se muestran los resultados obtenidos para la condición inicial no crítica (No. 25) de la tabla 4.1 ante una falla con una duración igual al tiempo crítico de liberación de cada controlador, siendo de 24.48 ciclos (408 mseg.) para el regulador ST1 y de 24.30 ciclos (405 mseg.) para el control propuesto.

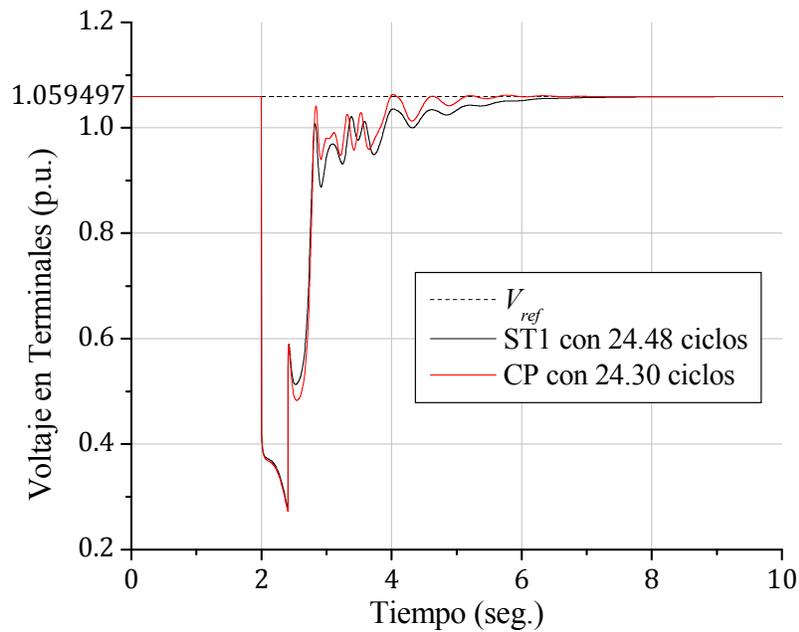


Figura 4.20. Voltaje en terminales para la condición inicial No. 25 ante una falla con una duración igual al tiempo crítico de liberación de cada controlador.

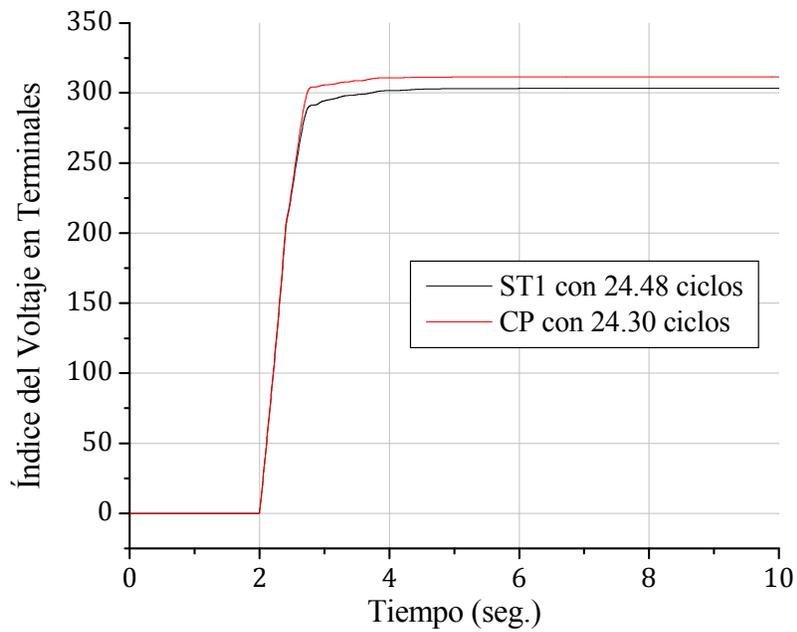


Figura 4.21. Índice de comportamiento del voltaje en terminales para la condición inicial No. 25 ante una falla con una duración igual al tiempo crítico de liberación de cada controlador.

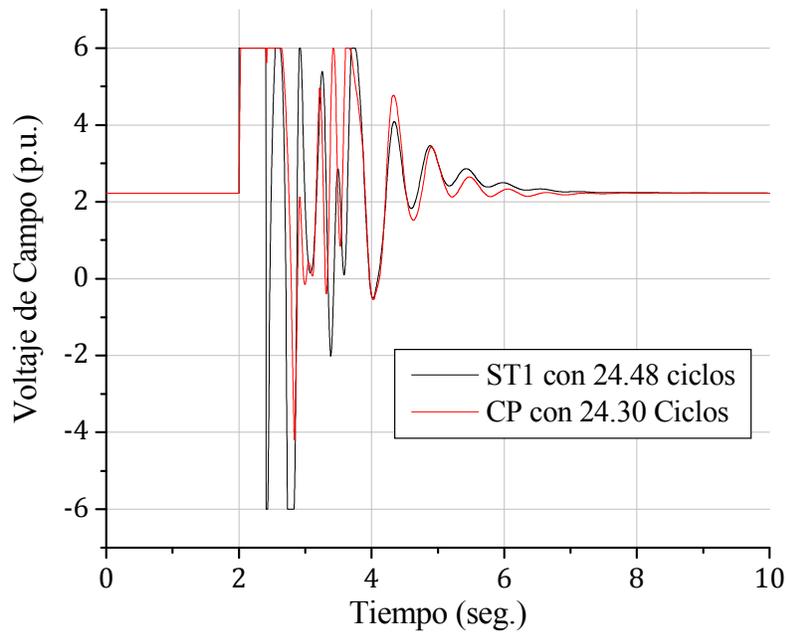


Figura 4.22. Voltaje de campo para la condición inicial No. 25 ante una falla con una duración igual al tiempo crítico de liberación de cada controlador.

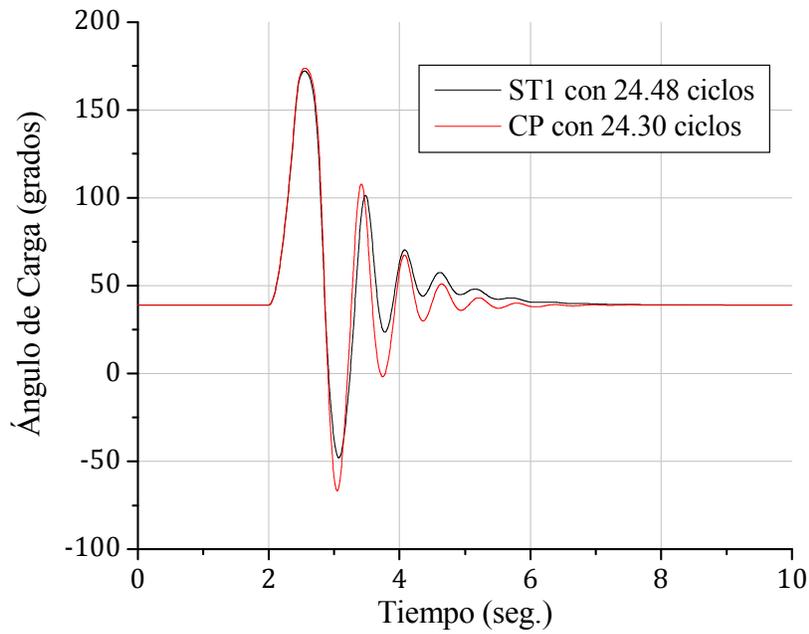


Figura 4.23. Ángulo de carga para la condición inicial No. 25 ante una falla con una duración igual al tiempo crítico de liberación de cada controlador.

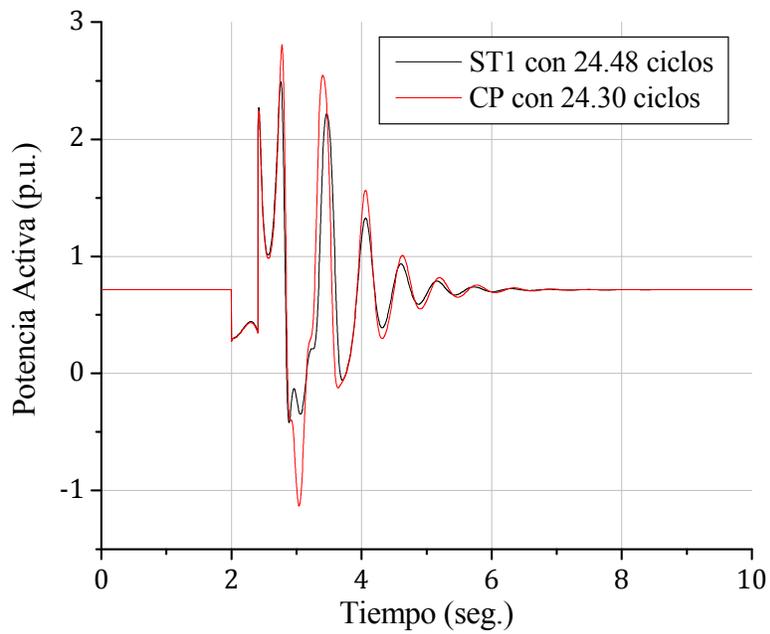


Figura 4.24. Potencia activa para la condición inicial No. 25 ante una falla con una duración igual al tiempo crítico de liberación de cada controlador.

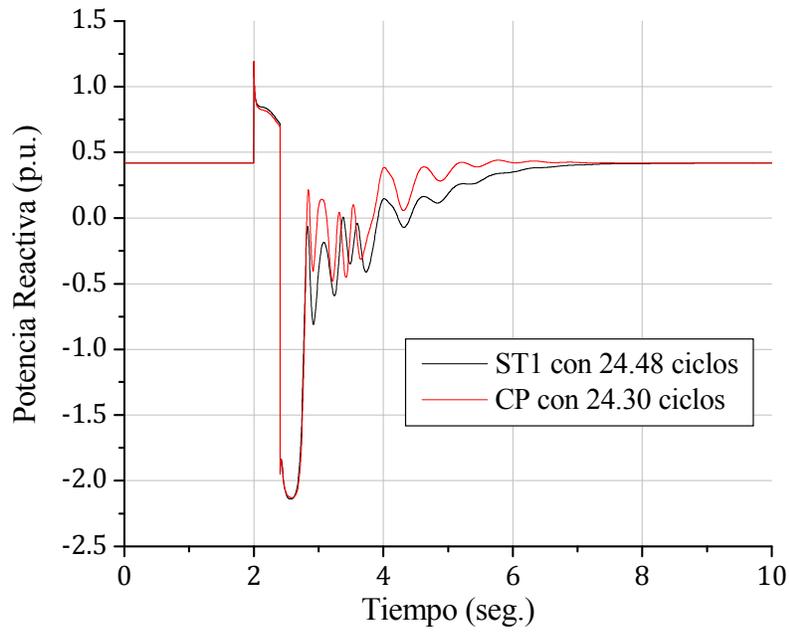


Figura 4.25. Potencia reactiva para la condición inicial No. 25 ante una falla con una duración igual al tiempo crítico de liberación de cada controlador.

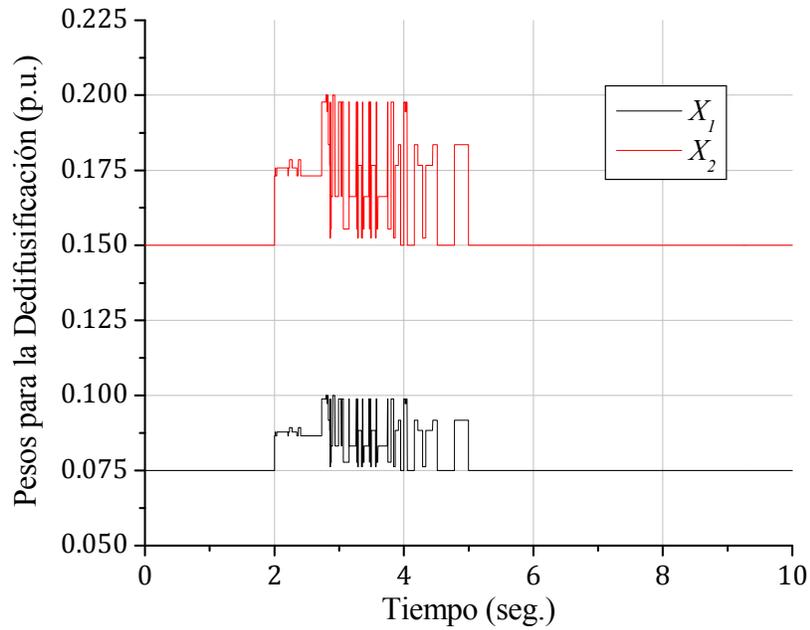


Figura 4.26. Pesos X_1 y X_2 para la condición inicial No. 25 ante una falla de 24.30 ciclos de duración.

4.3.2 Resultados para una Condición Inicial Crítica

En las figuras 4.27-4.33 se muestran los resultados obtenidos para la condición inicial crítica (No. 30) de la tabla 4.1 ante una falla de 1.8 ciclos (30 mseg.) de duración.

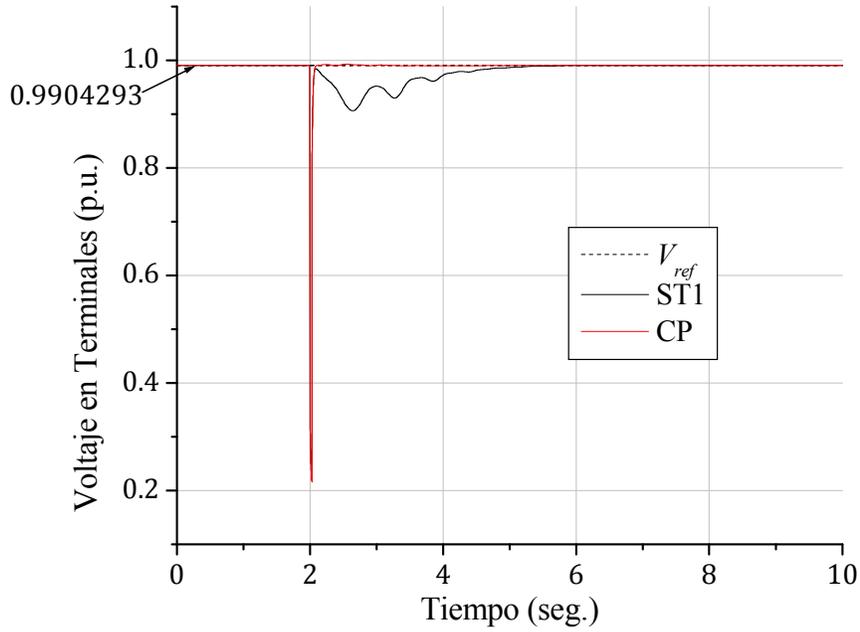


Figura 4.27. Voltaje en terminales para la condición inicial No. 30 ante una falla de 1.8 ciclos de duración.

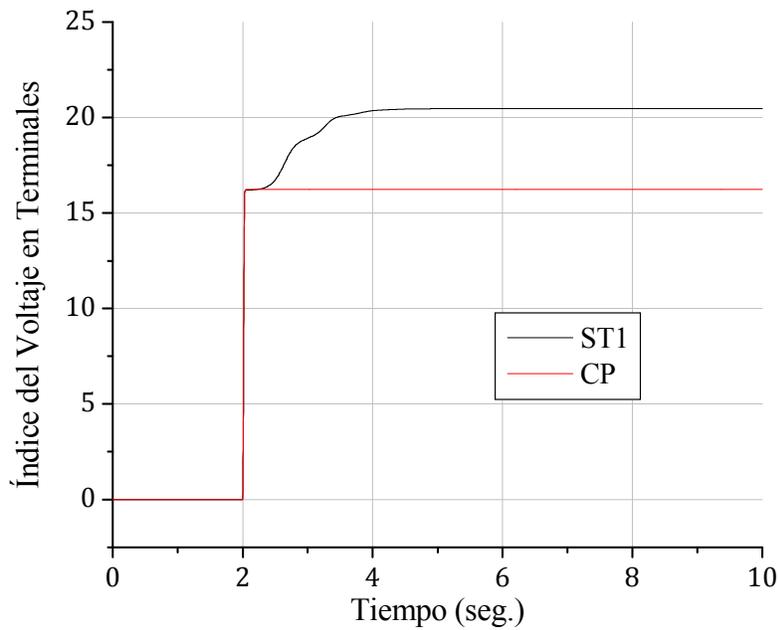


Figura 4.28. Índice de comportamiento del voltaje en terminales para la condición Inicial No. 30 ante una falla de 1.8 ciclos de duración.

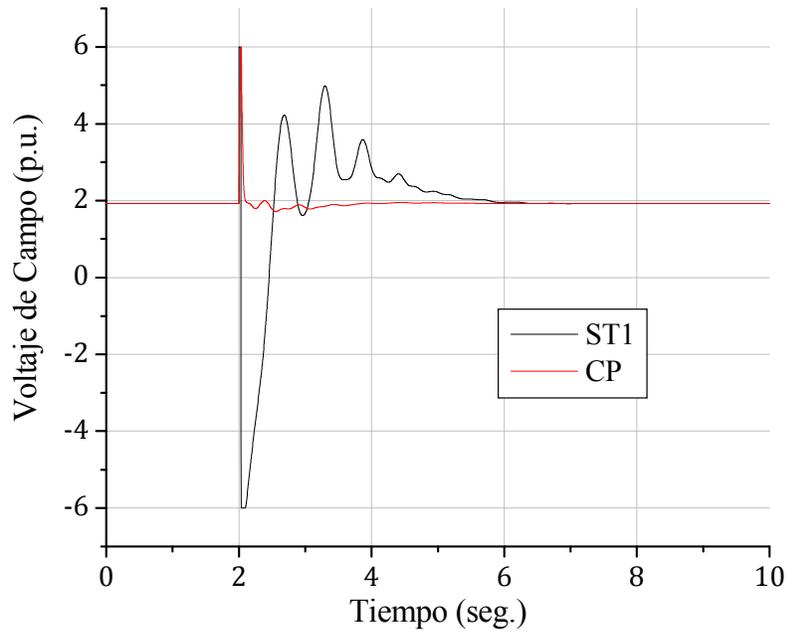


Figura 4.29. Voltaje de campo para la condición inicial No. 30 ante una falla de 1.8 ciclos de duración.

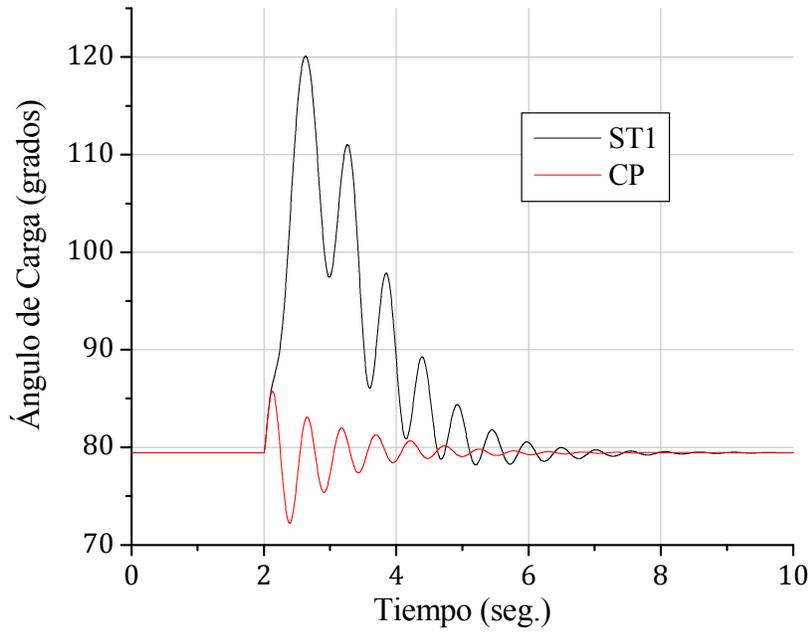


Figura 4.30. Ángulo de carga para la condición inicial No. 30 ante una falla de 1.8 ciclos de duración.

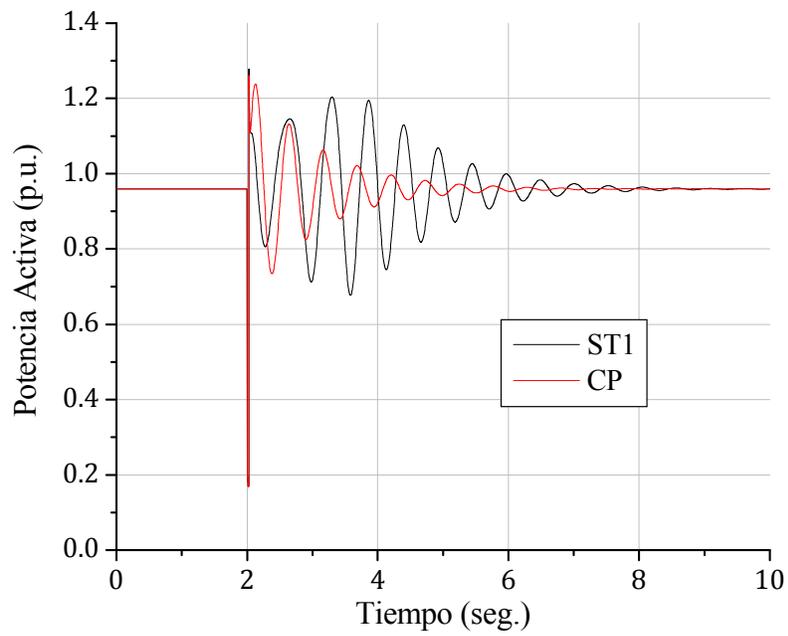


Figura 4.31. Potencia activa para la condición inicial No. 30 ante una falla de 1.8 ciclos de duración.

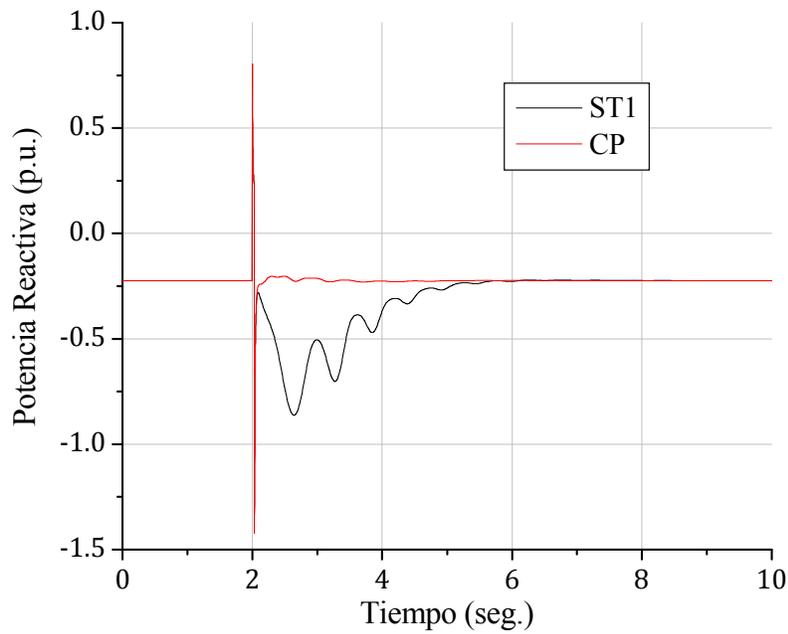


Figura 4.32. Potencia reactiva para la condición inicial No. 30 ante una falla de 1.8 ciclos de duración.

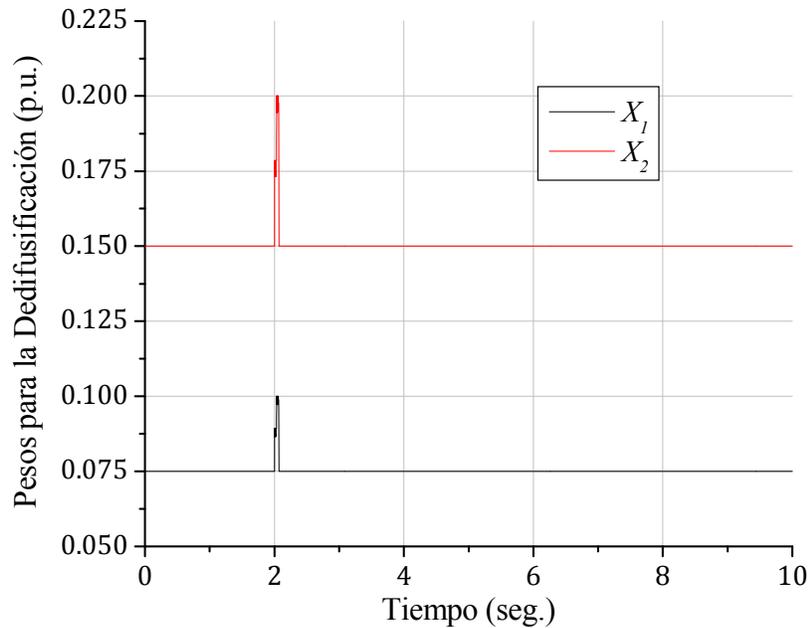


Figura 4.33. Pesos X_1 y X_2 para la condición inicial No. 30 ante una falla de 1.8 ciclos de duración.

En las figuras 4.34-4.40 se muestran los resultados obtenidos para la condición inicial crítica (No. 30) de la tabla 4.1 ante una falla con una duración igual al tiempo crítico de liberación de cada controlador, siendo de 3.60 ciclos (60 msec.) para el regulador ST1 y de 10.74 ciclos (179 msec.) para el control propuesto.

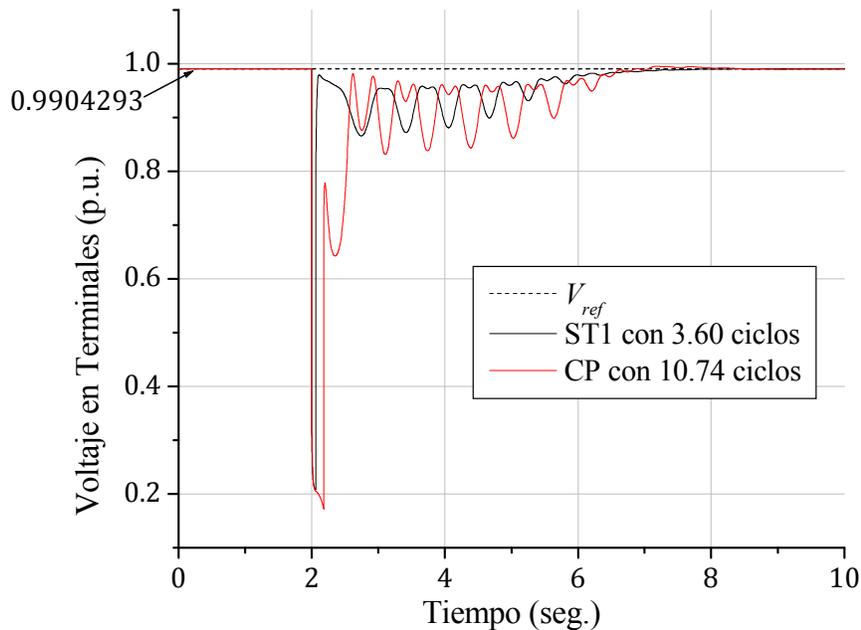


Figura 4.34. Voltaje en terminales para la condición inicial No. 30 ante una falla con una duración igual al tiempo crítico de liberación de cada controlador.

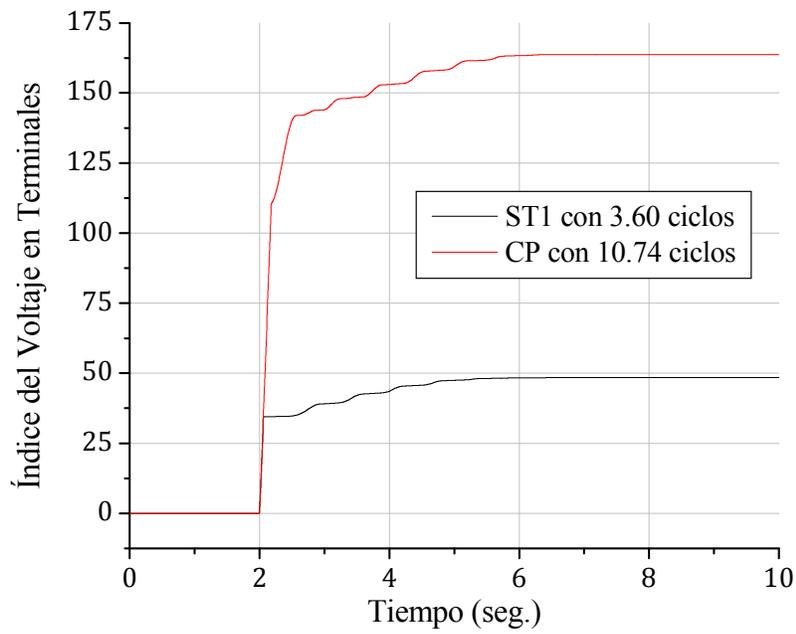


Figura 4.35. Índice de comportamiento del voltaje en terminales para la condición inicial No. 30 ante una falla con una duración igual al tiempo crítico de liberación de cada controlador.

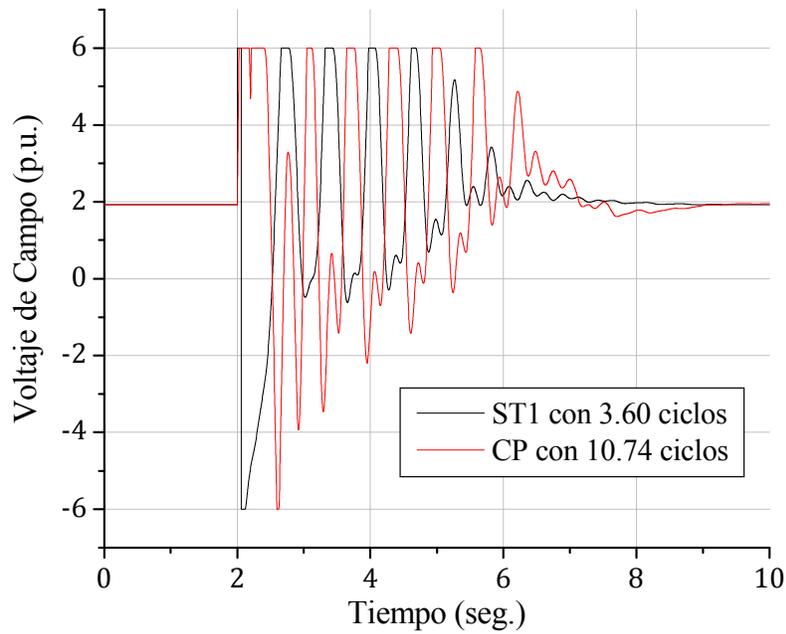


Figura 4.36. Voltaje de campo para la condición inicial No. 30 ante una falla con una duración igual al tiempo crítico de liberación de cada controlador.

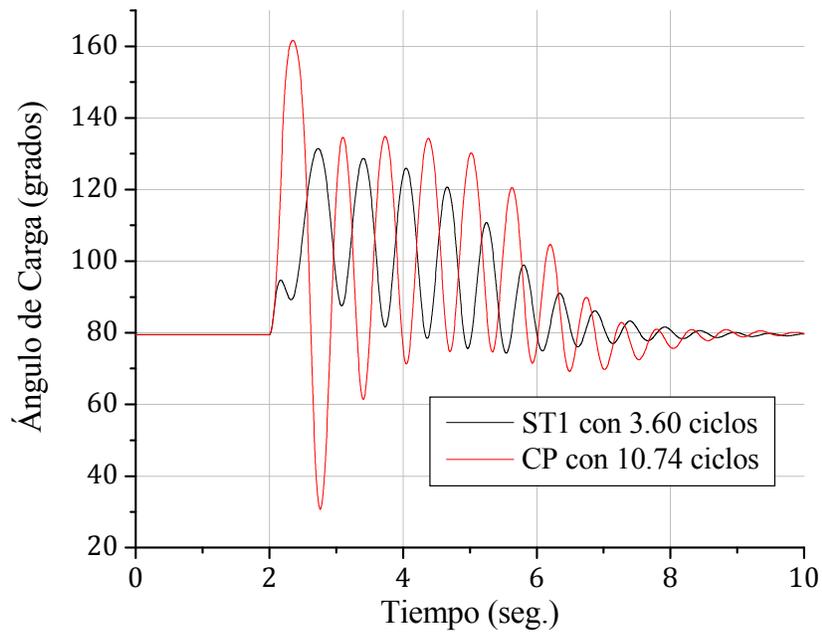


Figura 4.37. Ángulo de carga para la condición inicial No. 30 ante una falla con una duración igual al tiempo crítico de liberación de cada controlador.

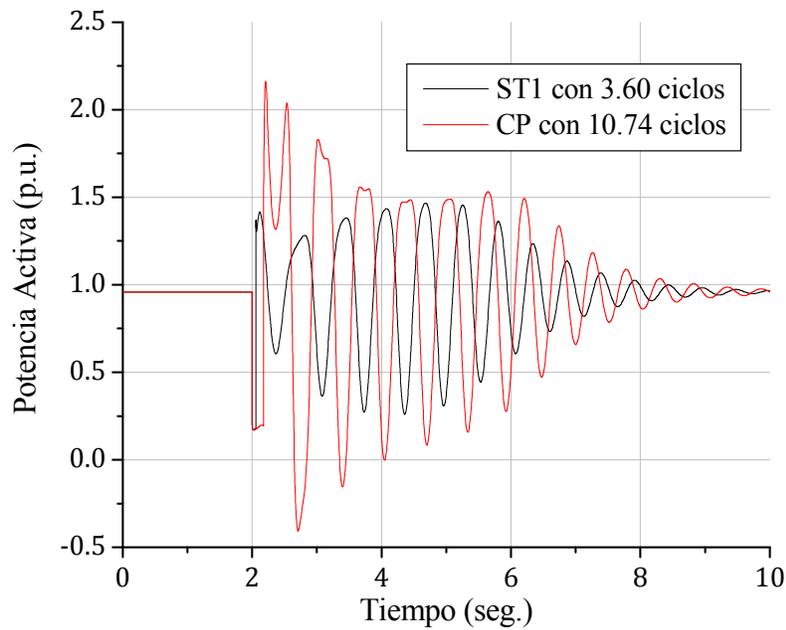


Figura 4.38. Potencia activa para la condición inicial No. 30 ante una falla con una duración igual al tiempo crítico de liberación de cada controlador.

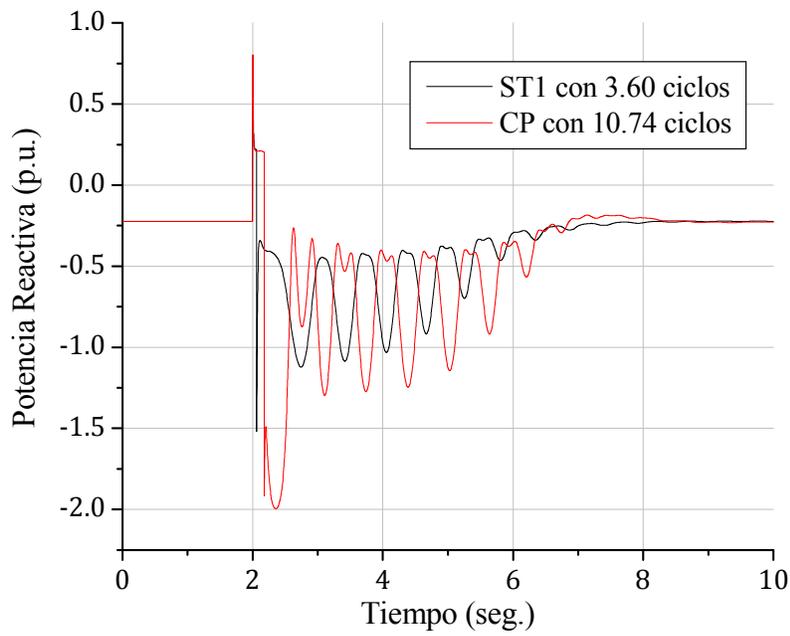


Figura 4.39. Potencia reactiva para la condición inicial No. 30 ante una falla con una duración igual al tiempo crítico de liberación de cada controlador.

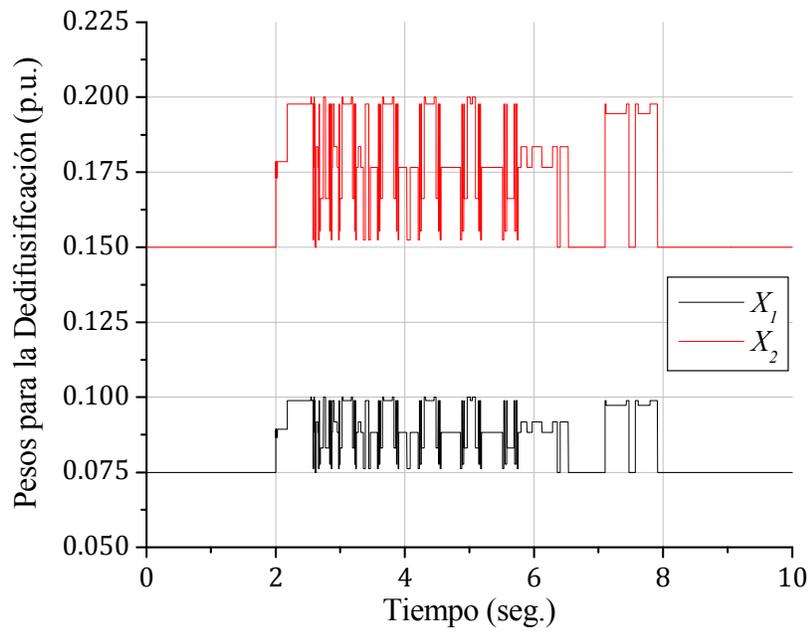


Figura 4.40. Pesos X_1 y X_2 para la condición inicial No. 30 ante una falla de 10.74 ciclos de duración.

4.3.3 Análisis de los Resultados

En general, en las figuras 4.13-4.18, 4.20-4.25, 4.27-4.32 y 4.34-4.39 se observa que el sistema bajo prueba regresa al punto inicial de operación más rápido con el controlador propuesto que con el regulador ST1 después de que se suscita una falla de corto circuito trifásica, sobre todo si dicha falla se libera en un tiempo no muy cercano al tiempo crítico de liberación de falla.

En las figuras 4.19, 4.26, 4.33 y 4.40 se aprecia que, una vez que se suscita una falla trifásica, la arquitectura FAM modifica continuamente el valor de los pesos X_1 y X_2 hasta que la respuesta del voltaje en terminales regresa al estado estacionario inicial. Por lo tanto, a medida que se incrementa la duración de la falla y que la condición inicial de la que parte el sistema es más crítica, las variaciones de los pesos son más significativas y duraderas, debido a que la arquitectura FAM nota al sistema en estado transitorio por periodos de tiempo cada vez mayores.

En ambas condiciones iniciales, con fallas de 12 y 1.8 ciclos, respectivamente, las respuestas del voltaje en terminales y del índice de comportamiento de este voltaje son claramente mejores con el controlador propuesto que con el regulador ST1 (figuras 4.13, 4.14, 4.27 y 4.28). Las respuestas del voltaje de campo, del ángulo de carga y de la potencia reactiva obtenidas con el controlador neurodifuso son más rápidas y presentan oscilaciones menos pronunciadas que las respuestas que ofrece respectivamente el regulador ST1 (figuras 4.15, 4.16, 4.18, 4.29, 4.30 y 4.32). Este comportamiento también se manifiesta en la respuesta de la potencia activa para la condición inicial crítica (figura 4.31); sin embargo, para la condición inicial no crítica, la potencia activa es muy similar en cuanto a rapidez y nivel de oscilación con ambos controladores (figura 4.17).

El tiempo crítico de liberación de falla para el sistema bajo prueba partiendo de la condición inicial no crítica es mayor con el regulador ST1 que con el controlador neurodifuso; a pesar de esta diferencia, las respuestas que se obtienen con este último controlador son más rápidas (figuras 4.20-4.25). Por otro lado, el tiempo crítico de liberación que se obtiene con el controlador propuesto es mayor para la condición inicial crítica, ofreciendo respuestas similares que el regulador ST1 con un tiempo de liberación de falla significativamente menor (figuras 4.34, 4.36-4.39).

En la tabla 4.3 se muestra una comparación entre los tiempos críticos de liberación de falla obtenidos con el regulador ST1 y con el controlador propuesto para las 30 condiciones iniciales del Apéndice B, donde las filas sombreadas corresponden a las condiciones iniciales simuladas. En esta tabla se puede apreciar que los tiempos críticos de liberación de falla son mayores con el regulador ST1 para condiciones iniciales no críticas, teniéndose una diferencia máxima de 0.54 ciclos (9 mseg.), que corresponde a la condición inicial No. 21. Sin embargo, para condiciones iniciales más críticas, el controlador neurodifuso propuesto es el que ofrece tiempos de liberación de falla mayores, teniéndose una diferencia máxima de 7.14 ciclos (119 mseg.) con respecto al regulador ST1, que corresponde a la condición inicial No. 30.

Tabla 4.3. Comparación de tiempos críticos entre el regulador ST1 y el controlador propuesto.

| No. de Condición Inicial | Tiempo Crítico de Liberación de Falla | | | |
|--------------------------|---------------------------------------|--------------|-------------------|--------------|
| | Regulador ST1 | | Control Propuesto | |
| | No. de Ciclos | Milisegundos | No. de Ciclos | Milisegundos |
| 1 | 7.38 | 123 | 13.02 | 217 |
| 2 | 16.62 | 277 | 16.62 | 277 |
| 3 | 23.88 | 398 | 23.70 | 395 |
| 4 | 16.44 | 274 | 16.44 | 274 |
| 5 | 21.48 | 358 | 21.36 | 356 |
| 6 | 23.46 | 391 | 23.28 | 388 |
| 7 | 19.32 | 322 | 19.20 | 320 |
| 8 | 22.26 | 371 | 22.08 | 368 |
| 9 | 25.92 | 432 | 25.68 | 428 |
| 10 | 26.76 | 446 | 26.40 | 440 |
| 11 | 21.78 | 363 | 21.66 | 361 |
| 12 | 28.14 | 469 | 27.84 | 464 |
| 13 | 21.54 | 359 | 21.36 | 356 |
| 14 | 14.58 | 243 | 14.58 | 243 |
| 15 | 15.30 | 255 | 15.30 | 255 |
| 16 | 6.66 | 111 | 13.14 | 219 |
| 17 | 16.26 | 271 | 16.20 | 270 |
| 18 | 22.32 | 372 | 22.14 | 369 |
| 19 | 16.08 | 268 | 16.08 | 268 |
| 20 | 15.84 | 264 | 15.78 | 263 |
| 21 | 33.60 | 560 | 33.06 | 551 |
| 22 | 25.98 | 433 | 25.74 | 429 |
| 23 | 15.48 | 258 | 15.48 | 258 |
| 24 | 16.32 | 272 | 16.32 | 272 |
| 25 | 24.48 | 408 | 24.30 | 405 |
| 26 | 22.26 | 371 | 22.08 | 368 |
| 27 | 13.38 | 223 | 13.44 | 224 |
| 28 | 4.56 | 76 | 11.04 | 184 |
| 29 | 3.96 | 66 | 11.16 | 186 |
| 30 | 3.60 | 60 | 10.74 | 179 |

4.4 RESPUESTAS ANTE FALLAS DE CORTO CIRCUITO TRIFÁSICAS CON APERTURA DE LÍNEA

En esta sección se presentan las respuestas obtenidas del sistema bajo prueba ante fallas trifásicas para las dos condiciones iniciales de la tabla 4.1 contemplando el aislamiento de la falla mediante la apertura en ambos extremos de la línea donde se suscita el evento, por lo que la impedancia de la red cambia en el momento de la liberación de la falla.

La simulación de este evento requiere de un modelo de red desglosado; es decir, se necesitan los valores de las impedancias de ambas líneas así como la del transformador. Desafortunadamente, el modelo de la red del Apéndice A consta sólo de una impedancia equivalente, tal y como se muestra en la figura 4.41a; por lo tanto, en este trabajo se propone la red de la figura 4.41b, cuya impedancia equivalente corresponde a la de la figura 4.41a. De este modo, la impedancia equivalente de la red después de liberada la falla cambia a $z_{ex} = 0.072 + j0.305$.

Las simulaciones se realizan considerando que la falla inicia en $t = 2$ seg. con una duración cercana al tiempo crítico de liberación de falla de cada condición inicial. La comparación entre el controlador propuesto y el regulador ST1 sólo se realiza para la condición inicial No. 25, ya que, en el caso de la condición inicial No. 30, el tiempo crítico de liberación de falla que proporciona el regulador ST1 es de tan sólo 3.60 ciclos (tabla 4.3), con lo que el tiempo en el que este regulador opera de manera satisfactoria incluyendo el aislamiento de la línea en falla es aun más breve, de hecho no es mayor a 0.06 ciclos (1 mseg.).

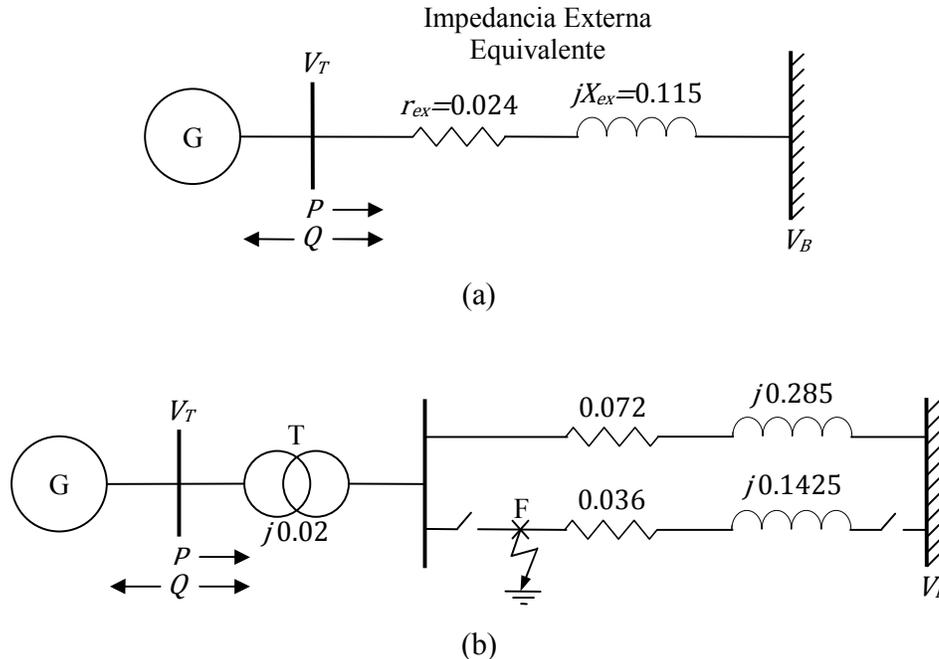


Figura 4.41. (a) Sistema bajo prueba original. (b) Sistema propuesto para simular fallas trifásicas con apertura de línea.

4.4.1 Resultados para una Condición Inicial No Crítica

En las figuras 4.42-4.48 se muestran los resultados obtenidos para la condición inicial no crítica (No. 25) de la tabla 4.1 ante una falla de 21 ciclos (350 mseg.) de duración con apertura de línea.

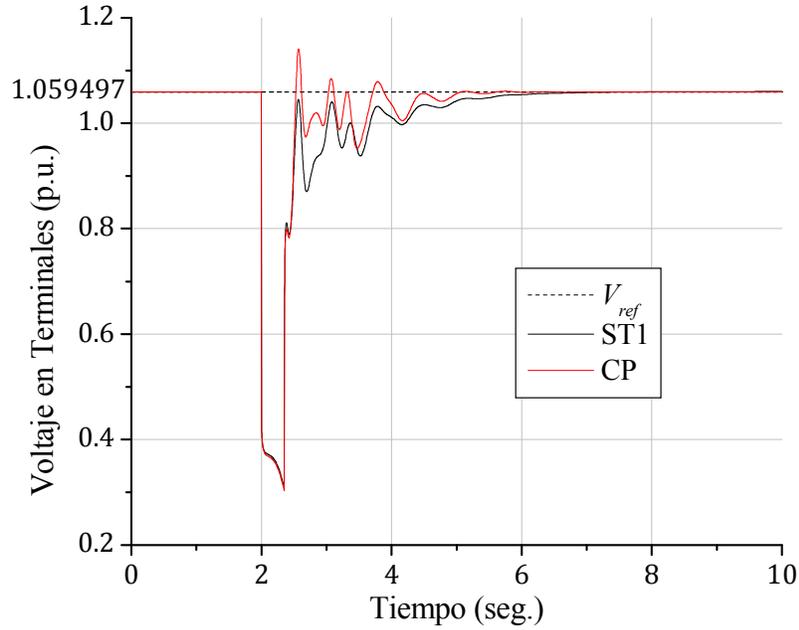


Figura 4.42. Voltaje en terminales para la condición inicial No. 25 ante una falla de 21 ciclos de duración con apertura de línea.

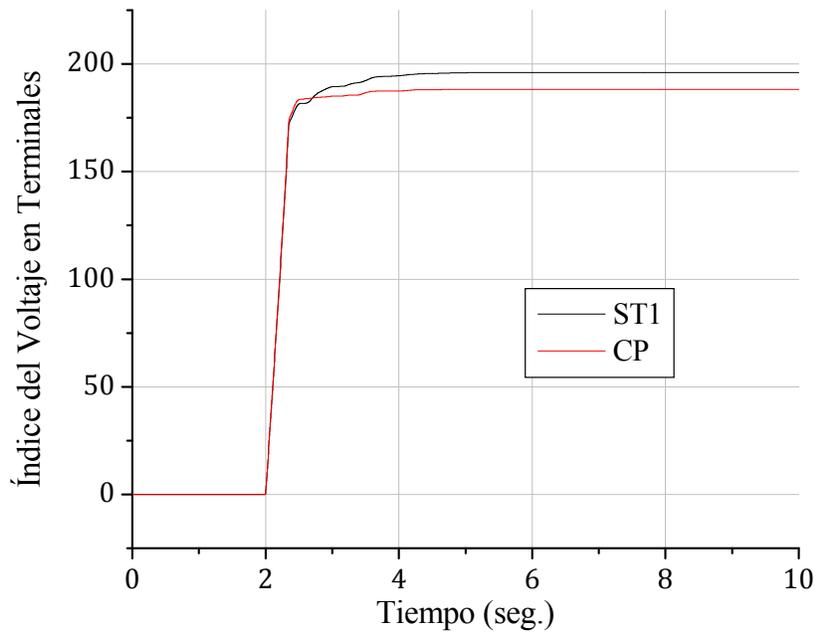


Figura 4.43. Índice de comportamiento del voltaje en terminales para la condición inicial No. 25 ante una falla de 21 ciclos de duración con apertura de línea.

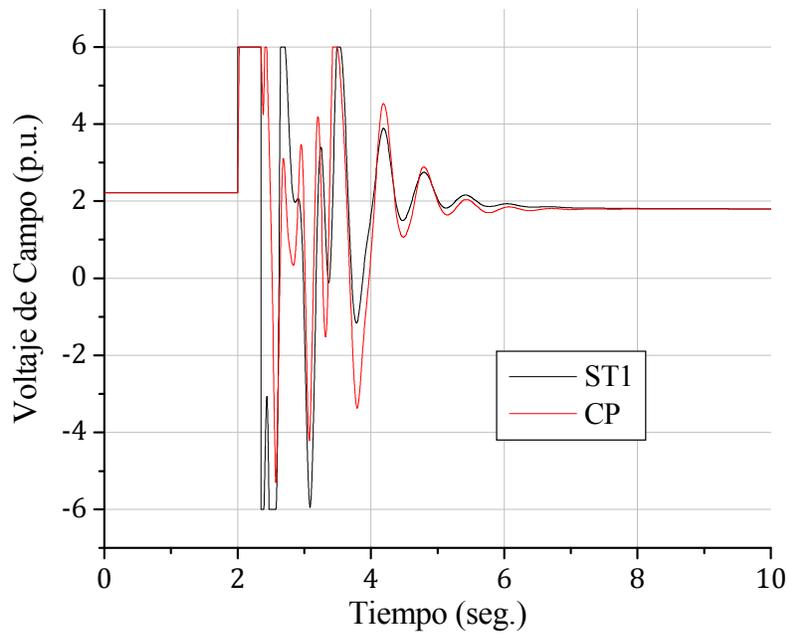


Figura 4.44. Voltaje de campo para la condición inicial No. 25 ante una falla de 21 ciclos de duración con apertura de línea.

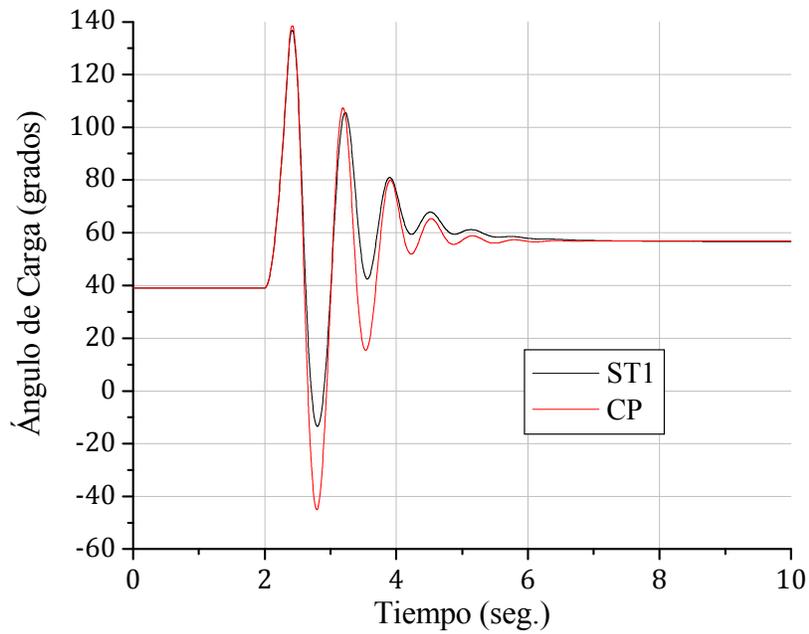


Figura 4.45. Ángulo de carga para la condición inicial No. 25 ante una falla de 21 ciclos de duración con apertura de línea.

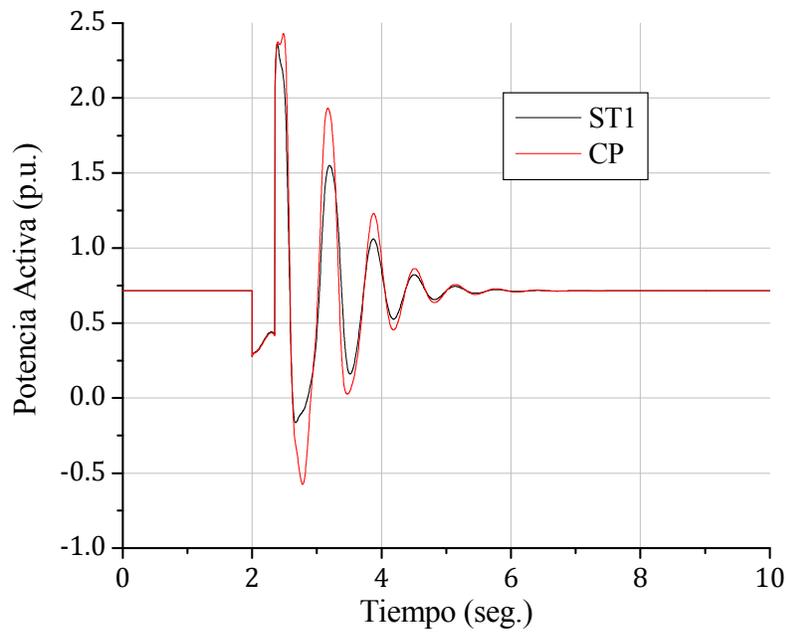


Figura 4.46. Potencia activa para la condición inicial No. 25 ante una falla de 21 ciclos de duración con apertura de línea.

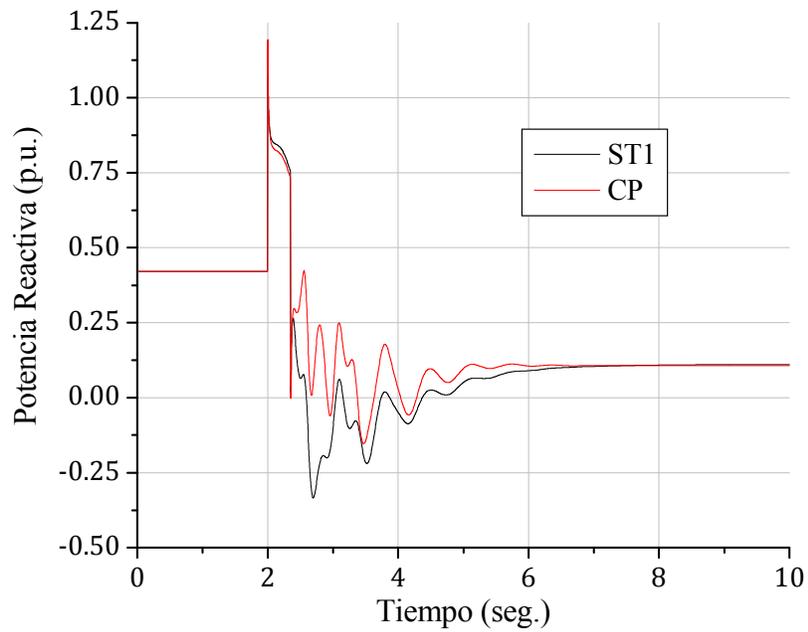


Figura 4.47. Potencia reactiva para la condición inicial No. 25 ante una falla de 21 ciclos de duración con apertura de línea.

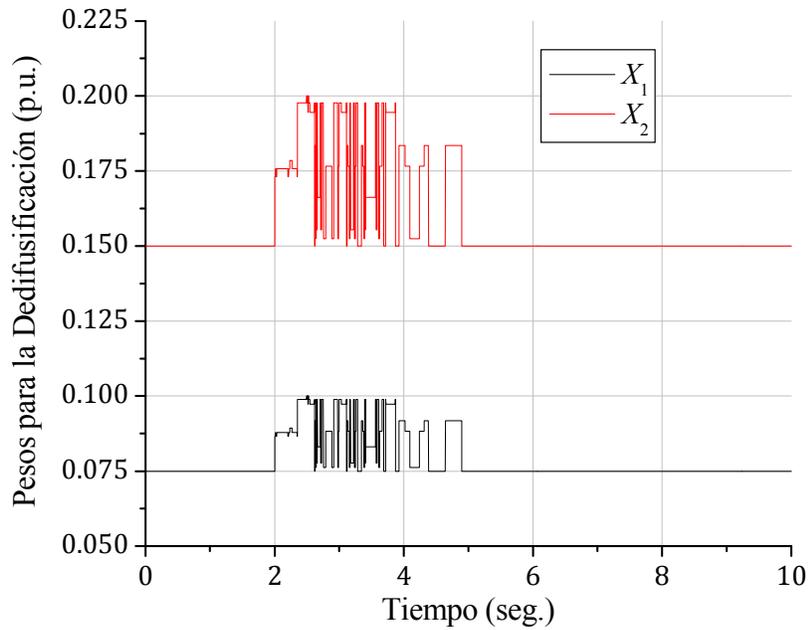


Figura 4.48. Pesos X_1 y X_2 para la condición inicial No. 25 ante una falla de 21 ciclos de duración con apertura de línea.

4.4.2 Resultados para una Condición Inicial Crítica

En las figuras 4.49-4.55 se muestran los resultados obtenidos con el control propuesto para la condición inicial crítica (No. 30) de la tabla 4.1 ante una falla de 5.4 ciclos (90 mseg.) de duración con apertura de línea.

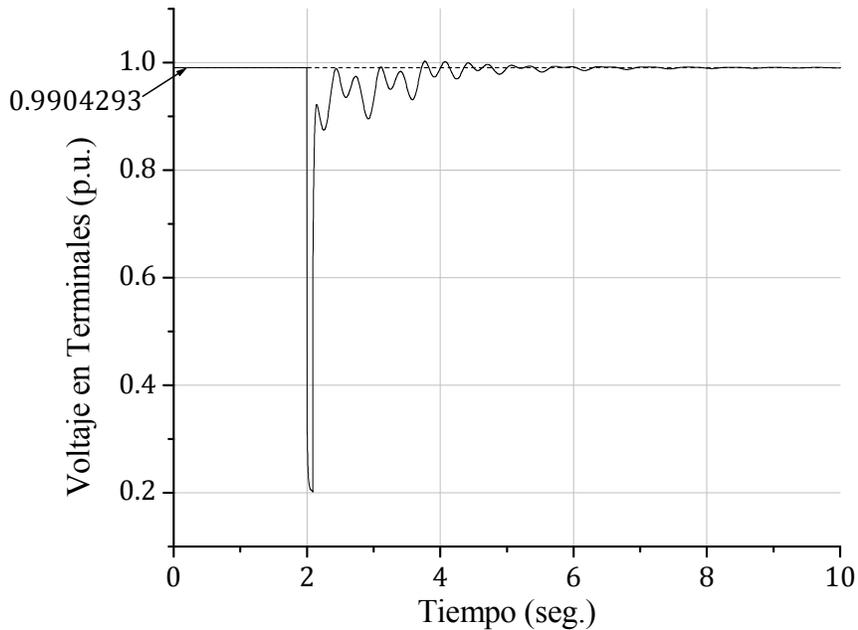


Figura 4.49. Voltaje en terminales para la condición inicial No. 30 ante una falla de 5.4 ciclos de duración con apertura de línea.

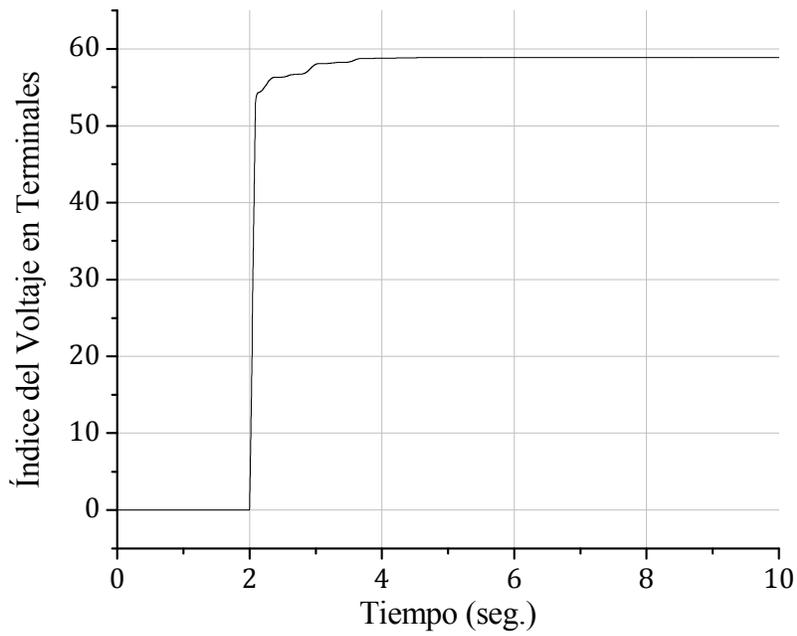


Figura 4.50. Índice de comportamiento del voltaje en terminales para la condición inicial No. 30 ante una falla de 5.4 ciclos de duración con apertura de línea.

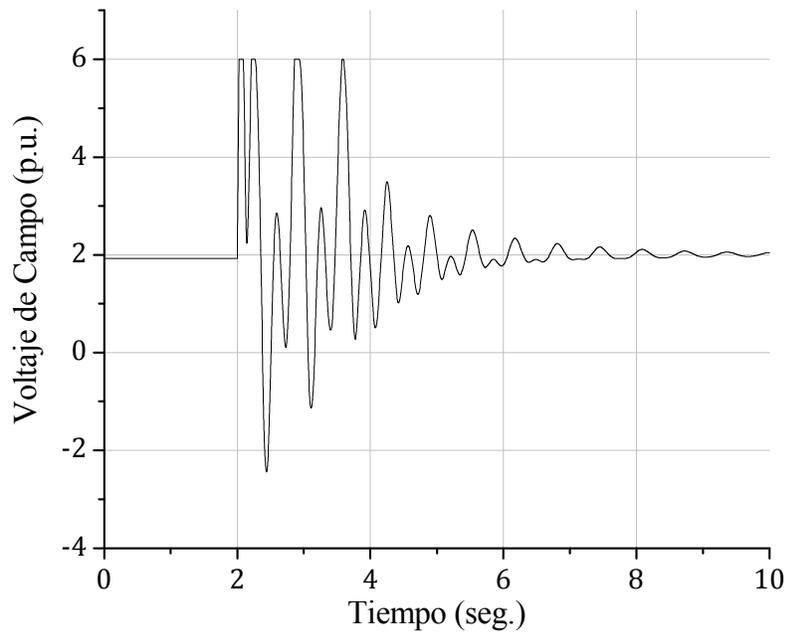


Figura 4.51. Voltaje de campo para la condición inicial No. 30 ante una falla de 5.4 ciclos de duración con apertura de línea.

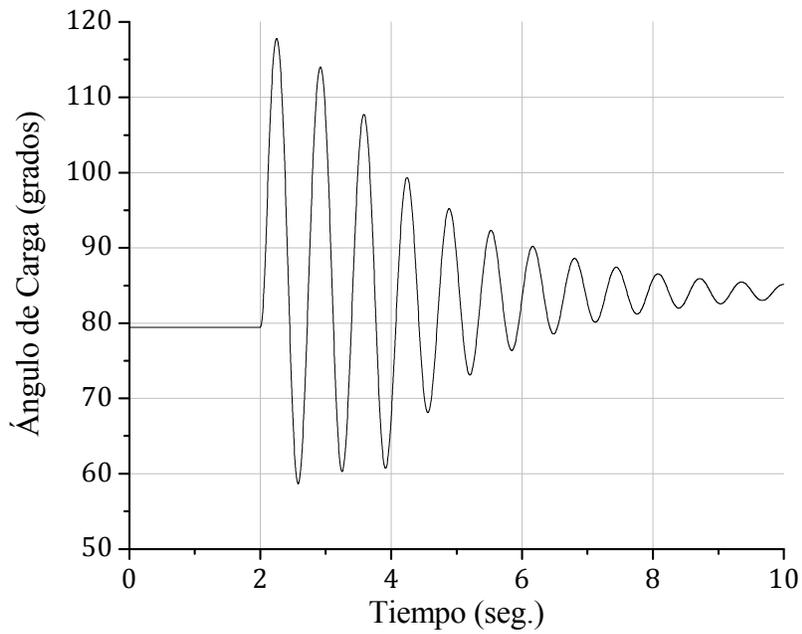


Figura 4.52. Ángulo de carga para la condición inicial No. 30 ante una falla de 5.4 ciclos de duración con apertura de línea.

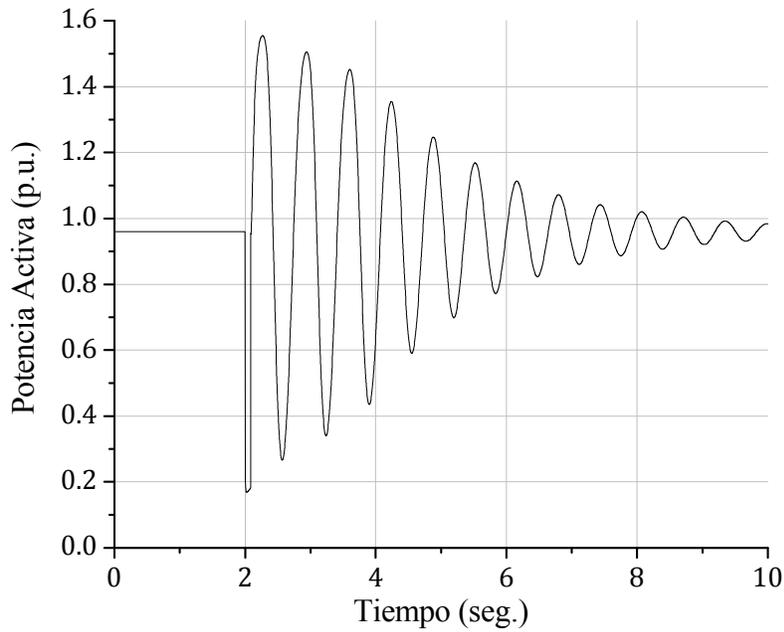


Figura 4.53. Potencia activa para la condición inicial No. 30 ante una falla de 5.4 ciclos de duración con apertura de línea.

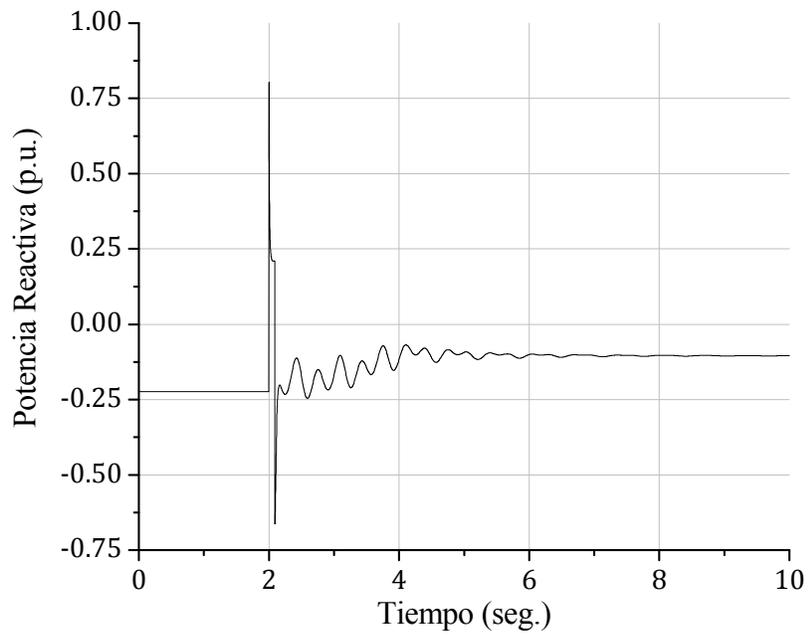


Figura 4.54. Potencia reactiva para la condición inicial No. 30 ante una falla de 5.4 ciclos de duración con apertura de línea.

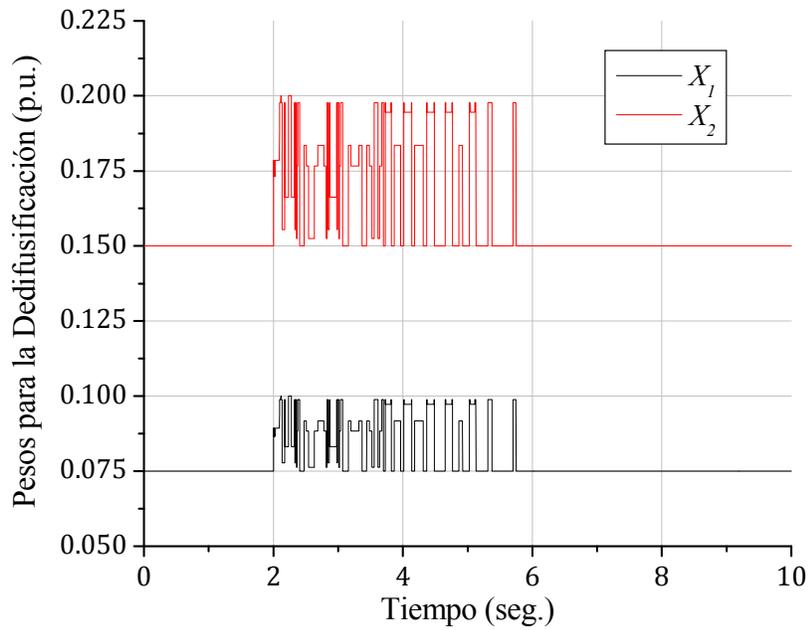


Figura 4.55. Pesos X_1 y X_2 para la condición inicial No. 30 ante una falla de 5.4 ciclos de duración con apertura de línea.

4.4.3 Análisis de los Resultados

En los resultados obtenidos de las simulaciones del generador con falla trifásica incluyendo el aislamiento del circuito en falla se tiene que, después de transcurrida la falla, el sistema regresa a un punto de operación diferente al inicial. Este efecto se puede apreciar en las figuras 4.44, 4.45 y 4.47 para la condición inicial no crítica, y en las figuras 4.51, 4.52 y 4.54 para la condición inicial crítica.

La figura 4.42 muestra una regulación de voltaje más eficiente por parte del controlador propuesto que del regulador ST1 para una condición inicial no crítica, característica que es confirmada por el índice de comportamiento del voltaje en terminales que se presenta en la figura 4.43. Las respuestas del voltaje de campo, ángulo de carga y potencia reactiva (figuras 4.44, 4.45 y 4.47) demuestran una habilidad superior del controlador propuesto para definir el nuevo punto de operación del sistema; sin embargo, la figura 4.46 muestra que la potencia activa del sistema, inmediatamente después de suscitarse la falla, es menos oscilatoria cuando se utiliza el regulador ST1, aunque las oscilaciones que se tienen con el controlador propuesto disminuyen más velozmente, con lo que la potencia activa vuelve al estado estacionario prácticamente en el mismo instante para ambos controladores.

En las figuras 4.49 y 4.50 se observa una regulación de voltaje eficiente por parte del controlador propuesto para el generador partiendo de una condición inicial crítica. Estas respuestas, que constan del voltaje en terminales así como de su índice de comportamiento, presentan un mínimo de oscilaciones en estado estacionario. Aunque las figuras 4.51-4.53 muestran oscilaciones pronunciadas para las respuestas del voltaje de campo, ángulo de carga y potencia activa, respectivamente, cada una de estas respuestas converge a un nuevo punto de operación.

Similarmente a la Sección 4.3, la salida de la arquitectura FAM hacia el controlador difuso comienza a variar en el instante en que se suscita una falla (figuras 4.48, 4.55). Las variaciones continúan hasta que esta arquitectura considera, de acuerdo a su entrenamiento, que el generador se encuentra en estado estacionario. A pesar de que el tiempo de liberación de falla considerado para la condición inicial crítica es significativamente menor que el considerado para la condición inicial no crítica, la participación activa de la arquitectura FAM es más prolongada para el caso de la condición inicial crítica (figura 4.55), puesto que el voltaje en terminales presenta oscilaciones consideradas por esta arquitectura como trascendentes durante más tiempo después de aislada la falla.

4.5 RESPUESTAS ANTE CAMBIOS DE CARGA

La simulación de los cambios de carga se lleva a cabo mediante el incremento de la potencia activa a través de la manipulación de la entrada de control del gobernador de velocidad u_g . En la tabla 4.4 se resume la manera de efectuar estos cambios para las simulaciones realizadas.

Tabla 4.4. Cambios de potencia activa para la simulación de los cambios de carga.

| Tiempo (seg.) | Potencia Activa (p.u.) | Razón de Incremento |
|---------------|------------------------------------|---------------------|
| 0 | Se conecta el generador al sistema | |
| 10 | Cambio de 0 a 0.3 | 0.1/seg. |
| 30 | Cambio de 0.3 a 0.6 | 0.1/seg. |
| 50 | Cambio de 0.6 a 0.8 | 0.05/seg. |
| 70 | Cambio de 0.8 a 0.995 | 0.05/seg. |

En las figuras 4.56-4.62 se muestran las respuestas obtenidas con el controlador propuesto (CP) en comparación con los que ofrece el regulador ST1 ante los cambios de carga señalados.

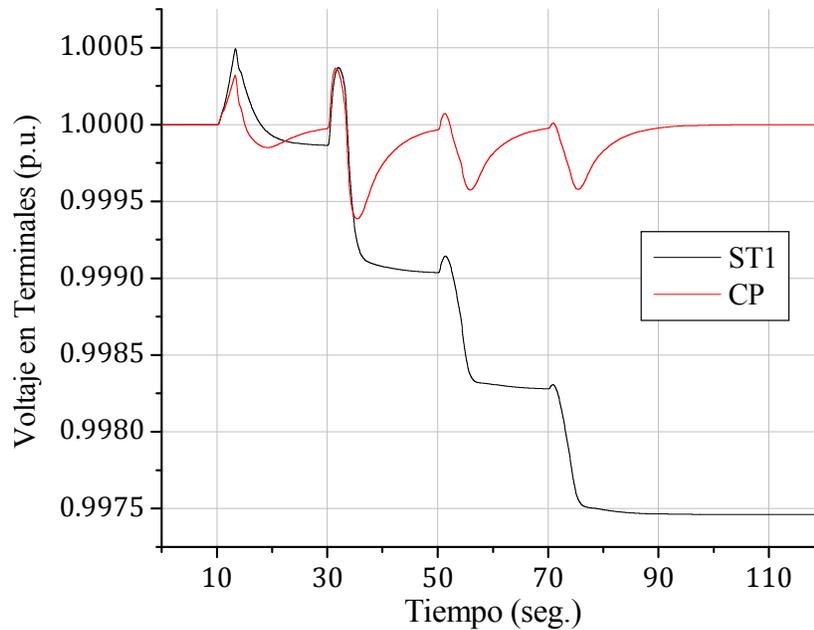


Figura 4.56. Voltaje en terminales ante cambios de carga.

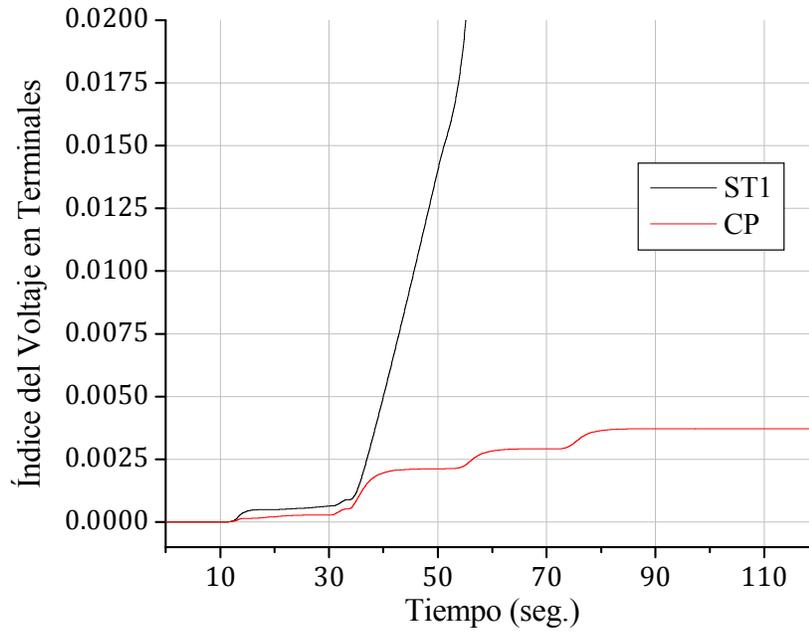


Figura 4.57. Índice de comportamiento del voltaje en terminales ante cambios de carga.

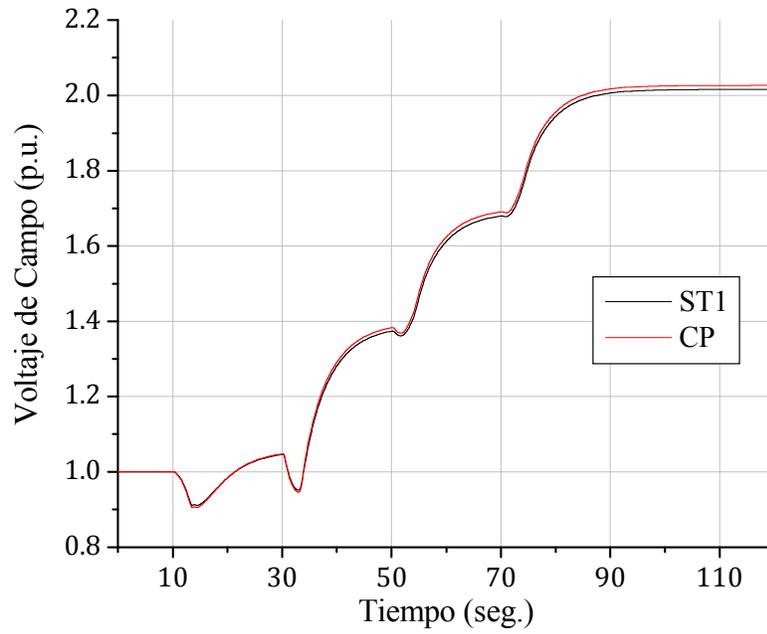


Figura 4.58. Voltaje de campo ante cambios de carga.

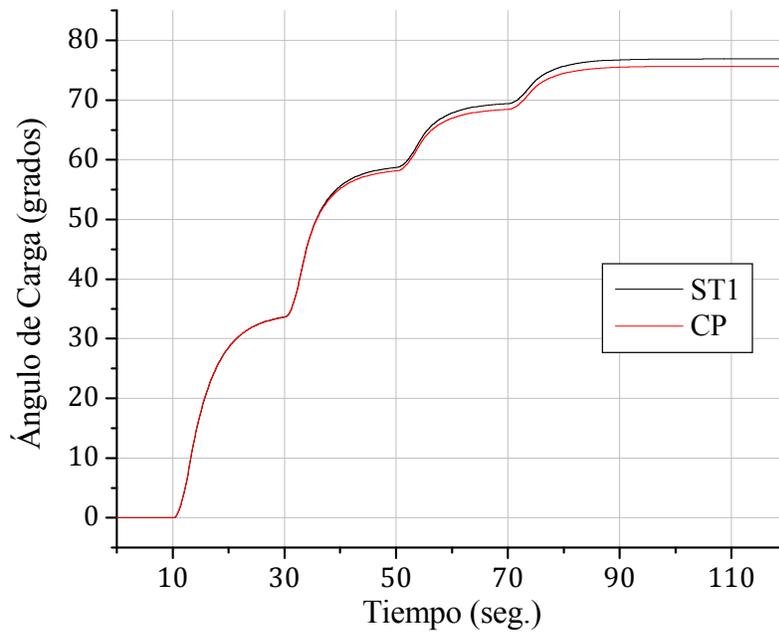


Figura 4.59. Ángulo de carga ante cambios de carga.

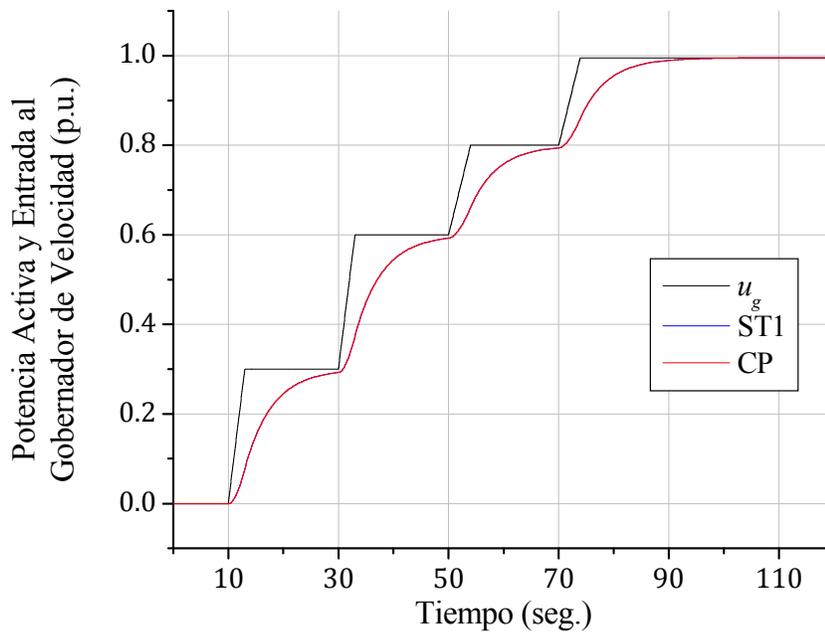


Figura 4.60. Potencia activa ante cambios de carga sobre la manipulación de la entrada del gobernador de velocidad u_g .

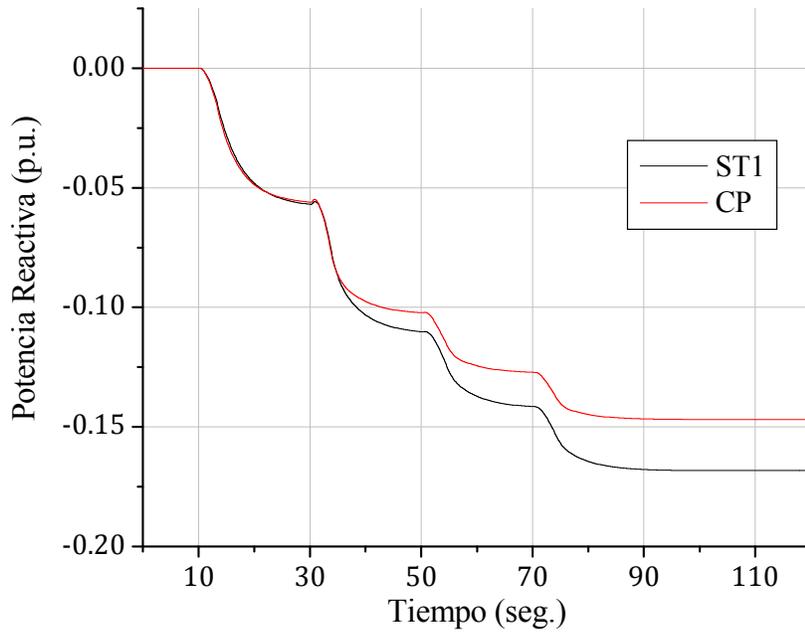


Figura 4.61. Potencia reactiva ante cambios de carga.

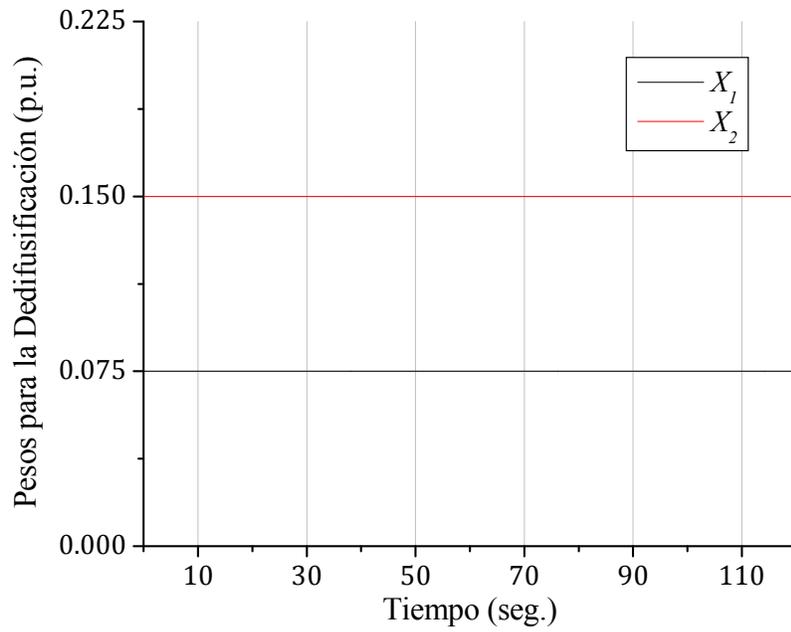


Figura 4.62. Pesos X_1 y X_2 ante cambios de carga.

4.5.1 Análisis de los Resultados

La figura 4.56 muestra que el voltaje en terminales recupera su estado inicial después de un incremento de carga cuando se utiliza el controlador propuesto, característica que no se presenta con el regulador ST1, ya que en esta misma figura se aprecia que la respuesta obtenida con este controlador decae cada vez que se presenta un cambio de carga, con lo que se genera un error en estado estacionario cada vez mayor, con un valor aproximado de 2.54×10^{-3} al final de la simulación. Esta situación se ve reflejada también en la figura 4.57.

La forma de la respuesta proporcionada por ambos controladores es muy similar, como se puede apreciar en la figura 4.58; la única diferencia es que el voltaje de campo proporcionado por el controlador propuesto se expande un poco más, adoptando valores más lejanos del valor inicial.

La regulación de voltaje más eficiente que brinda el controlador propuesto permite que los incrementos del ángulo de carga (figura 4.59) y la absorción de potencia reactiva (4.61) sean menos pronunciados que cuando se utiliza el regulador ST1. Las respuestas de estas variables junto con la propia de la potencia activa (que es igual para ambos controladores, razón por la que no se aprecia la curva correspondiente al regulador ST1 en la figura 4.60) conforman un punto de operación final menos crítico para el generador.

Similarmente que en la sección 4.2, la arquitectura FAM no detecta perturbaciones significativas que generen la necesidad de aumentar el valor de los pesos X_1 y X_2 , razón por la que estos pesos permanecen constantes durante todo el tiempo de simulación, tal y como lo muestra la figura 4.62.

4.6 RESPUESTAS ANTE CAMBIOS DE REFERENCIA

En esta sección se muestra el comportamiento del sistema bajo prueba al modificarse el voltaje de referencia para las dos condiciones iniciales de la tabla 4.1. En la tabla 4.5 se resumen los cambios de referencia simulados. De nueva cuenta, los resultados obtenidos con el controlador propuesto se comparan con los ofrecidos por el regulador ST1.

Tabla 4.5. Cambios del voltaje de referencia.

| Tiempo (seg.) | Voltaje de Referencia (p.u.) | Razón de Incremento |
|---------------|---|---------------------|
| 0 | Inicio de la simulación con condiciones iniciales específicas | |
| 10 | Cambio de 100 a 95% del valor inicial | 0.05 p. u./seg. |
| 40 | Cambio de 95 a 105% del valor inicial | 0.05 p. u./seg. |

4.6.1 Resultados para una Condición Inicial No Crítica

En las figuras 4.63-4.69 se muestran los resultados obtenidos para la condición inicial no crítica (No. 25) de la tabla 4.1 ante los cambios de referencia indicados en la tabla 4.5.

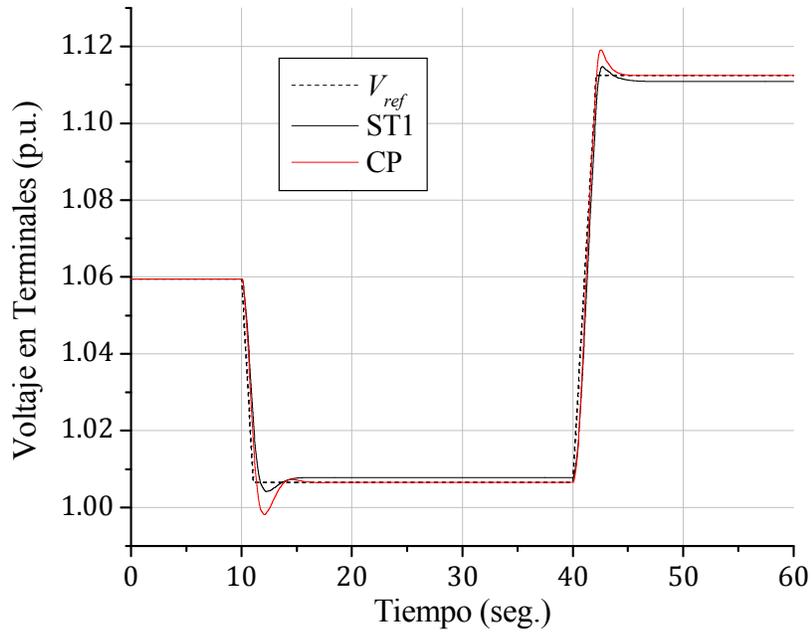


Figura 4.63. Voltaje en terminales para la condición inicial No. 25 ante cambios de referencia.

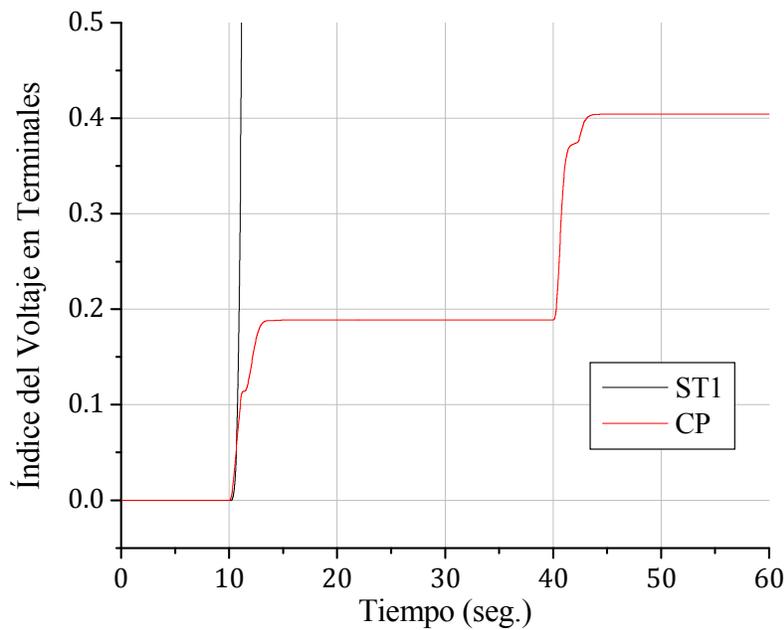


Figura 4.64. Índice de comportamiento del voltaje en terminales para la condición inicial No. 25 ante cambios de referencia.

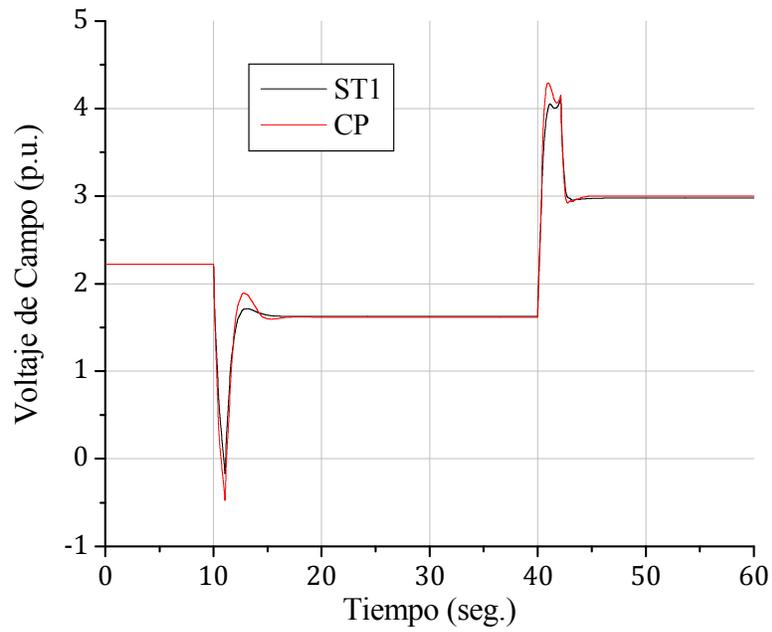


Figura 4.65. Voltaje de campo para la condición inicial No. 25 ante cambios de referencia.

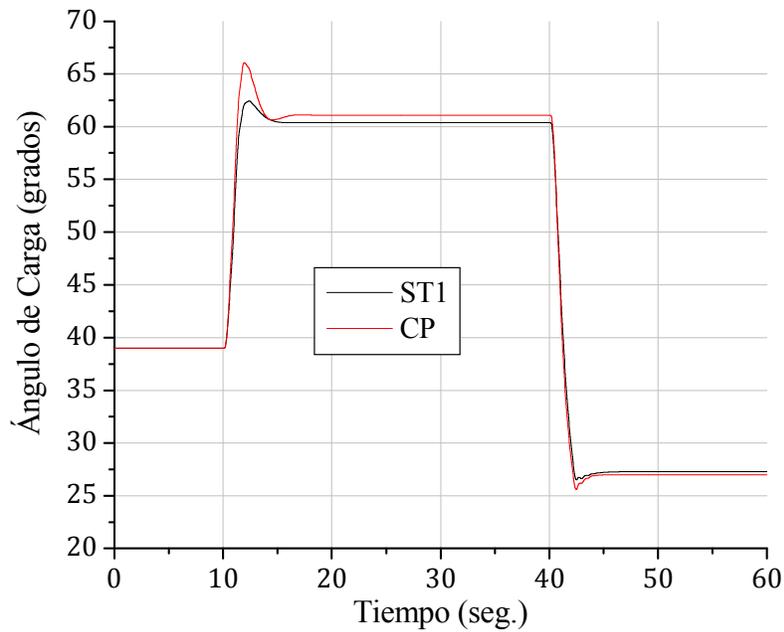


Figura 4.66. Ángulo de carga para la condición inicial No. 25 ante cambios de referencia.

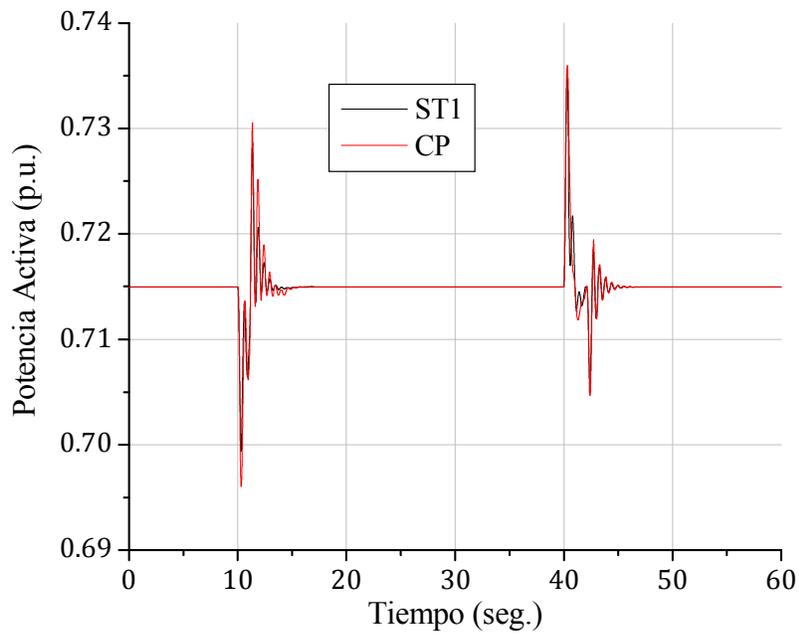


Figura 4.67. Potencia activa para la condición inicial No. 25 ante cambios de referencia.

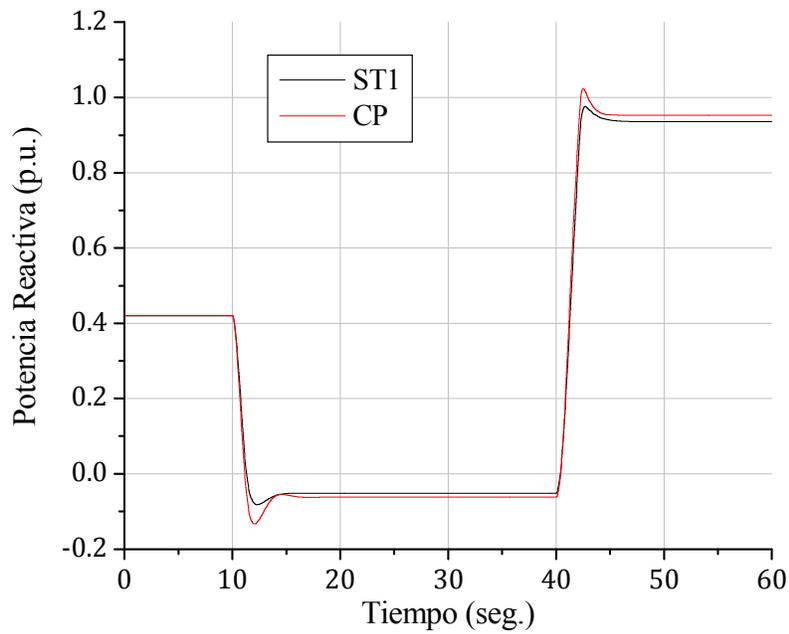


Figura 4.68. Potencia reactiva para la condición inicial No. 25 ante cambios de referencia.

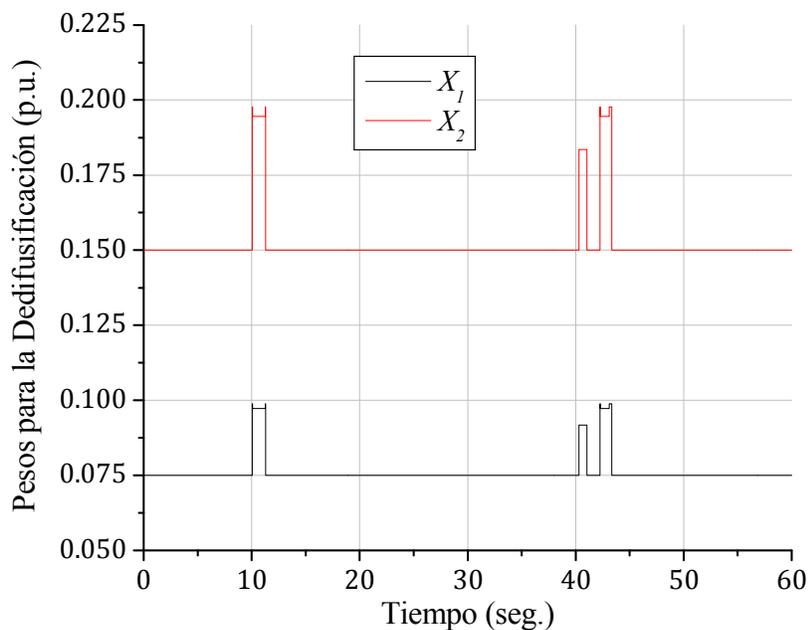


Figura 4.69. Pesos X_1 y X_2 para la condición inicial No. 25 ante cambios de referencia.

4.6.2 Resultados para una Condición Inicial Crítica

En las figuras 4.70-4.76 se muestran los resultados obtenidos para la condición inicial crítica (No. 30) de la tabla 4.1 ante los cambios de referencia indicados en la tabla 4.5.

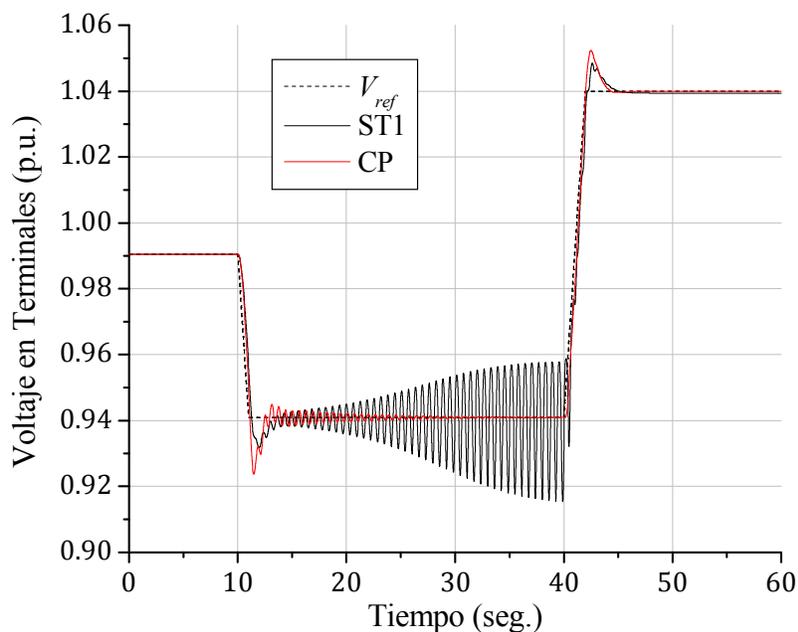


Figura 4.70. Voltaje en terminales para la condición inicial No. 30 ante cambios de referencia.

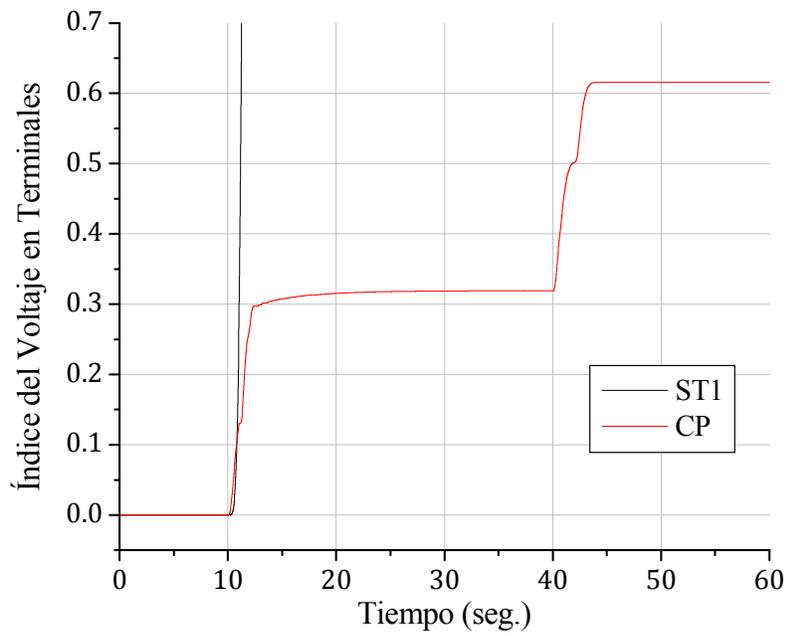


Figura 4.71. Índice de comportamiento del voltaje en terminales para la condición inicial No. 30 ante cambios de referencia.

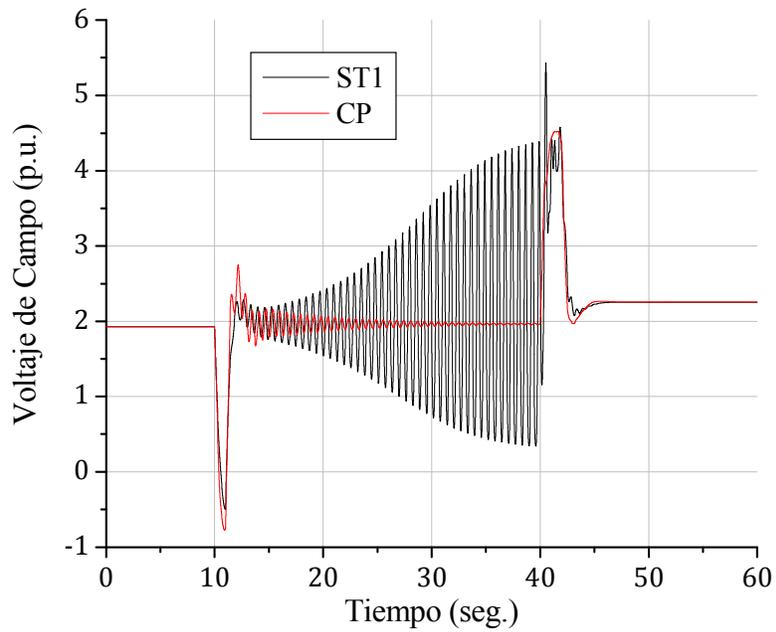


Figura 4.72. Voltaje de campo para la condición inicial No. 30 ante cambios de referencia.

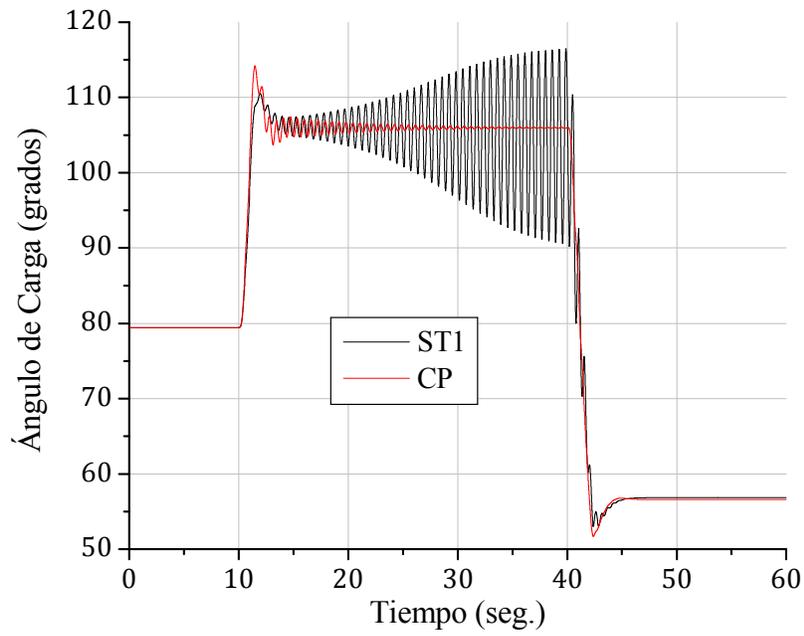


Figura 4.73. Ángulo de carga para la condición inicial No. 30 ante cambios de referencia.

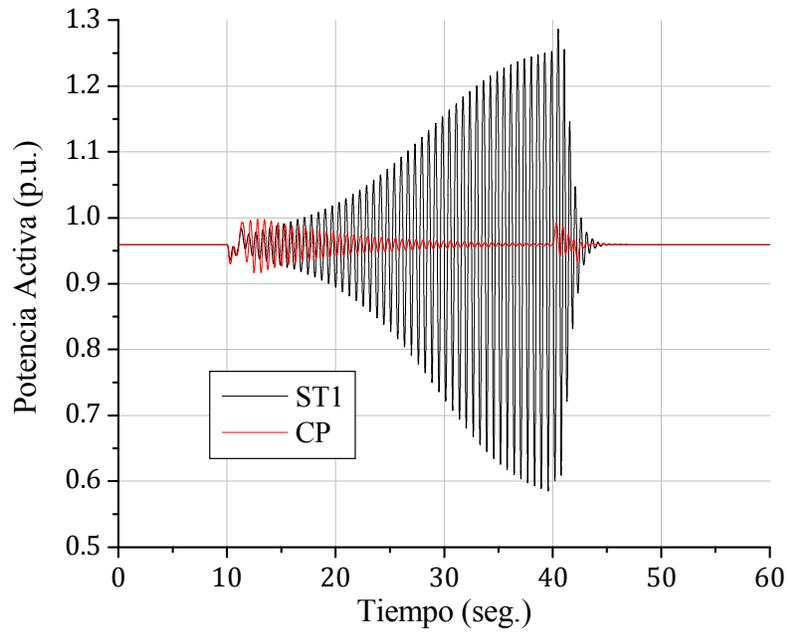


Figura 4.74. Potencia activa para la condición inicial No. 30 ante cambios de referencia.

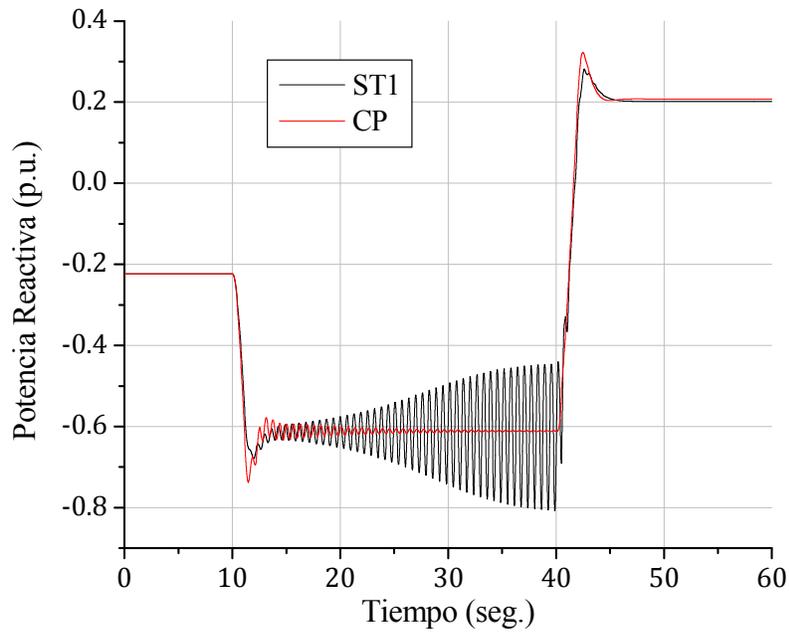


Figura 4.75. Potencia reactiva para la condición inicial No. 30 ante cambios de referencia.

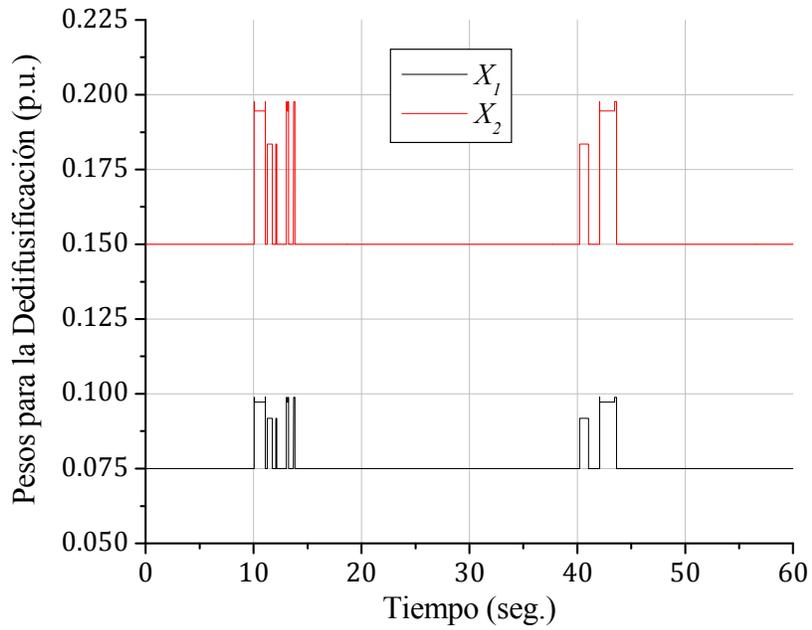


Figura 4.76. Pesos X_1 y X_2 para la condición inicial No. 30 ante cambios de referencia.

4.6.3 Análisis de los Resultados

En la figura 4.63 se observa que, en el caso de la condición inicial no crítica, el voltaje en terminales obtenido con el regulador ST1 presenta errores en estado estacionario cuando la referencia es constante, mientras que con el controlador propuesto se elimina tal error; este efecto se puede confirmar observando la respuesta del índice de comportamiento del voltaje en terminales para ambos controladores (figura 4.64). Asimismo, el controlador neurodifuso es el que sigue más fielmente a las rampas descendente y ascendente, motivo por el cual el sobretiro que se tiene con este controlador es un poco mayor que con el convencional.

En las figuras 4.65, 4.66 y 4.68 se observa que las respuestas para el voltaje de campo, ángulo de carga y potencia reactiva son similares con ambos controladores, con la diferencia que las curvas obtenidas con el controlador propuesto presentan mayores sobretiros, consiguiéndose así una mayor rapidez en la ubicación de nuevos puntos de operación para el generador. Por su parte, la respuesta de la potencia activa (figura 4.67) presenta el mismo tipo de oscilaciones con ambos controladores cuando se disminuye y se aumenta la referencia, para después regresar a su mismo valor inicial.

En las figuras 4.70, 4.72-4.75 resulta evidente la presencia de oscilaciones incrementales una vez que la referencia se fija en el 95% de su valor inicial cuando se utiliza el regulador ST1 para regular el voltaje de un generador que parte de condiciones iniciales críticas; aunque, en el momento en que inician los incrementos de la referencia, estas oscilaciones desaparecen. En el caso del controlador propuesto, las oscilaciones son mínimas cuando la referencia alcanza el 95% de su valor inicial; además, estas oscilaciones disminuyen con el tiempo hasta volverse prácticamente imperceptibles.

La arquitectura FAM realiza una tarea importante al cambiar la referencia del sistema bajo prueba. En las figuras 4.69 y 4.76 se observa que esta arquitectura detecta la presencia de perturbaciones en el instante en que se comienza a variar la referencia de entrada del sistema bajo prueba y decide incrementar el valor de los pesos X_1 y X_2 para una regulación más eficiente por parte del controlador difuso utilizado.

CAPÍTULO 5

CONCLUSIONES, APORTACIONES Y RECOMENDACIONES PARA TRABAJOS FUTUROS

5.1 CONCLUSIONES

A pesar de que la arquitectura FAM ha sido aplicada principalmente en tareas de reconocimiento de patrones y procesamiento de datos en diversas áreas del conocimiento, en el presente trabajo se demuestra que esta arquitectura bien puede utilizarse en labores de control.

Si bien es cierto que la arquitectura FAM no posee las características necesarias para ser considerada como un regulador de voltaje confiable aun cuando es capaz de copiar y extrapolar la dinámica de un controlador convencional a otros puntos de operación, tal y como lo demuestran los resultados mostrados en la Sección 3.4, la habilidad que posee esta arquitectura de aprender supervisada e incrementalmente categorías de reconocimiento estables para el mapeo multidimensional entre patrones de entrada analógicos se puede aprovechar como herramienta en la definición de las ganancias y/o los parámetros necesarios para sintonizar un determinado controlador; en este caso, un controlador difuso PI tipo Mamdani.

En los resultados presentados en el Capítulo 4 se puede apreciar que el control difuso propuesto operando en conjunción con la arquitectura FAM ofrece, en general, mejores resultados que el regulador convencional ST1 ante las pruebas realizadas.

En las respuestas de las Secciones 4.2 y 4.5 para las pruebas sin evento y con cambios de carga, respectivamente, la salida de la arquitectura FAM es constante durante el tiempo de simulación, lo que indica que esta arquitectura no encuentra al sistema lo suficientemente perturbado como para aumentar el valor de los pesos utilizados en el proceso de dedifusificación. Consecuentemente, el control propuesto se simplifica a únicamente el control difuso de la Sección 3.5.1 para estas simulaciones.

En los resultados obtenidos ante fallas de corto circuito trifásicas sin salida de línea de la Sección 4.3, los tiempos críticos de liberación de falla que se obtienen con el regulador ST1 son ligeramente mayores (no más de 0.54 ciclos) que los ofrecidos por el controlador propuesto para condiciones iniciales menos críticas (tabla 4.3); pero, para condiciones iniciales más críticas, el sistema bajo prueba presenta tiempos de liberación de falla significativamente mayores (hasta 7.14 ciclos) operando con el controlador propuesto que con el regulador ST1. De este modo, se aprecia que el controlador propuesto es más eficiente y confiable que el regulador ST1 ante este tipo de disturbios para cualquier condición inicial en general.

En las respuestas obtenidas ante fallas de corto circuito trifásicas con salida de línea de la Sección 4.4 se puede apreciar que, después de activadas las protecciones para dejar fuera a la línea en falla, el sistema bajo prueba se estabiliza en un nuevo punto de operación más rápidamente y con menos oscilaciones cuando se utiliza el controlador propuesto. Más aún, debido a que los tiempos críticos de liberación de falla obtenidos con el regulador ST1 para las condiciones iniciales críticas contempladas son muy bajos, no mayores a 5 ciclos, en la práctica resulta muy complicado activar protecciones dentro de este tiempo, máxime si se considera que el tiempo crítico de liberación de falla disminuye cuando se aísla físicamente el circuito en falla. Por el contrario, la activación oportuna de protecciones se facilita con el controlador propuesto, puesto que los tiempos críticos de liberación de falla para las condiciones iniciales críticas consideradas son mayores a 10 ciclos.

Las respuestas obtenidas ante cambios de referencia de la Sección 4.6 muestran también un comportamiento del sistema bajo prueba más favorable con el controlador propuesto que con el regulador ST1. Esta prueba en particular resulta relevante porque se puede visualizar la respuesta del sistema ante rampas con pendiente tanto negativa como positiva. Bajo este entorno, se observa que las respuestas obtenidas con el controlador propuesto, además de que siguen de mejor manera a las rampas definidas, presentan mucho menos oscilaciones que las respuestas brindadas por el regulador ST1.

Todos los resultados contenidos en el Capítulo 4 muestran un comportamiento típico de un control PI por parte del controlador propuesto en determinados segmentos de tiempo, lo que sugiere que el sistema presente oscilaciones excesivas en el instante de presentarse algún evento o disturbio. Sin embargo, este efecto se ve disminuido por la presencia de la arquitectura FAM, la cual indica en cada instante si se debe aumentar o no el valor de los pesos en el proceso de dedifusificación. Este efecto se puede visualizar claramente en las figuras 4.19, 4.26, 4.33, 4.40, 4.48, 4.55, 4.69 y 4.76, donde los valores de los pesos X_1 y X_2 experimentan cambios repentinos durante la presencia de los eventos simulados, para después regresar a sus respectivos valores iniciales, lo que indica que el sistema vuelve a encontrarse en estado estacionario.

En suma a los resultados presentados en el Capítulo 4, en el Apéndice F se muestran las respuestas obtenidas con el controlador propuesto ante una falla trifásica en el bus, sólo que en ese caso, tales respuestas se comparan con las que se obtienen con el controlador difuso de la Sección 3.5.1 con pesos fijos. En tales respuestas se puede apreciar que, en el momento en que se libera la falla, el sistema regresa más rápido y con oscilaciones menos pronunciadas cuando se regula el voltaje en terminales con el controlador propuesto que con el control difuso con parámetros fijos. Esto se debe a la oportuna intervención de la arquitectura FAM para incrementar los valores de los pesos en el momento necesario; es decir en el instante en que se suscita la falla.

En resumen, la arquitectura FAM puede ser utilizada para inferir valores que sintonicen un determinado controlador a través de mapeos multidimensionales entre vectores de entrada. Esta aplicación es de gran ayuda si se considera el hecho de que esta arquitectura posee una estructura bien definida, además de que resulta más sencilla de programar y de aplicar que los modelos de redes neurodifusas diseñadas específicamente para control, como las redes FALCONART y ANFIS; esto sin mencionar el tiempo relativamente corto que le toma categorizar y aprender a relacionar vectores de entrada analógicos.

5.2 APORTACIONES

Los resultados presentados en este trabajo se pueden comprobar a través del programa desarrollado, el cual se encuentra detallado en el Apéndice C. Este programa es capaz de simular el sistema máquina bus infinito del Apéndice A con tres opciones iniciales diferentes: sin regulación (voltaje de campo constante), con el sistema de regulación ST1 del Apéndice A o con un controlador neurodifuso creado a partir de la arquitectura FAM. Dentro de esta última opción se tiene la posibilidad de aplicar la arquitectura FAM ya sea directamente como regulador o como sintonizador del controlador difuso PI tipo Mamdani presentado en la Sección 3.5.1.

Dentro de cualquier opción de regulación, el programa desarrollado ofrece la posibilidad de simular fallas trifásicas sin y con salida de la línea en falla, cambios de carga, cambios de referencia y sin la presencia de evento o disturbio alguno para diferentes condiciones iniciales. Además, este programa presenta pocas limitantes en cuanto al tiempo de estudio y al tiempo de inicio y término de los diferentes disturbios que se deseen simular, puesto que la mayoría de estos parámetros se definen en tiempo de ejecución. Estas características reflejan una gran versatilidad por parte del programa para visualizar el comportamiento del generador en cuestión bajo distintos escenarios de operación, lo que resulta ser de gran ayuda en el análisis de este tipo de máquinas eléctricas.

5.3 RECOMENDACIONES PARA TRABAJOS FUTUROS

Como principal recomendación para trabajos futuros, se propone la implementación de un control difuso sintonizado por una red neurodifusa para regular el voltaje de un sistema máquina bus infinito. Para ello no es completamente necesaria la aplicación de la arquitectura FAM como tal, sino que puede usarse una versión simplificada de la misma, todo depende de la creatividad del o los diseñadores del control. En cualquier caso, se recomienda la programación de los algoritmos involucrados en lenguaje C, dado que este lenguaje es el más ampliamente utilizado por dispositivos electrónicos de control, como los microcontroladores y los procesadores de señales digitales (DSP por sus siglas en inglés); estos últimos destacan por la habilidad que poseen de procesar información de forma paralela, característica que facilita la implementación de redes neurodifusas.

En la aplicación de redes neuronales o neurodifusas para el copiado de la dinámica de un controlador existente, es recomendable comenzar el entrenamiento con las entradas que

utiliza dicho controlador, y después, en base a pruebas y análisis de resultados, definir las variables adecuadas para formar los vectores de entrada con los que se obtenga un mejor comportamiento del sistema en general. Esto se debe realizar de forma óptima; es decir, buscar los mejores resultados con la menor cantidad de entradas posible, debido a que un gran número de entradas vuelve a la red neuronal o neurodifusa impráctica por la gran cantidad de datos que tiene que procesar tanto en fase de entrenamiento como en fase de producción; esto implica inherentemente la necesidad de localizar un balance entre funcionalidad y rapidez.

Se recomienda también buscar alternativas o elementos auxiliares que complementen la configuración de la arquitectura FAM con la finalidad de que pueda ser utilizada directamente como regulador de voltaje. Estas alternativas pueden consistir de la adición de una o varias capas extra de neuronas con funciones de activación definidas en base a las categorías de salida de la arquitectura FAM, o de la asignación de relevancias a las entradas.

Finalmente, en muchos trabajos de investigación, incluyendo el presente, se ha demostrado la superioridad en funcionamiento y confiabilidad de los controles inteligentes en comparación con los convencionales; es por ello que se recomienda la aplicación de los controles inteligentes en los diferentes niveles de los sistemas eléctricos de potencia. Evidentemente, la sustitución de un gran número de controles es una labor enteramente complicada; esto se puede lograr implementando este tipo de controles a nivel laboratorio, después en sistemas piloto, para que en un futuro se puedan implementar en un sistema real. Una alternativa viable la constituyen los sistemas de generación distribuida, dado que estos sistemas en muchos países no cuentan con una reglamentación y una normatividad bien definidas. En cualquier caso, el primer paso para la utilización de los controles inteligentes en SEP's es la disposición por parte de las diferentes áreas del conocimiento involucradas para cooperar en este tipo de proyectos; puesto que en los últimos años, las distancias entre las ingenierías se han reducido enormemente, generando la ineludible necesidad de desarrollar proyectos de investigación multidisciplinarios.

REFERENCIAS

- [1] Prabha Kundur, "Power system stability and control", McGraw-Hill, 1994.
- [2] IEEE Committee Report, "Dynamic models for steam and hydro turbines in power system studies", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-92, pp. 1904-1915, Oct. 1972.
- [3] IEEE Committee Report "Proposed excitation system definitions for synchronous machines", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-88, No. 8, Aug. 1969.
- [4] IEEE Committee Report, "Excitation system models for power system stability studies", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-100, No. 2, pp. 494-509, Feb. 1981.
- [5] Olle I. Elgerd, "Control of electric power systems", IEEE, pp. 4-16, Mar. 1981.
- [6] Nasser Hosseinzadeh and Akhtar Kalam, "A rule-based fuzzy power system stabilizer tuned by a neural network", IEEE Transactions on Energy Conversion, Vol. 14, No. 3, pp. 773-779 Sep. 1999.
- [7] E. A. Feilat, A. M. Jaroshi and S. M. Radaideh, "Adaptive neuro-fuzzy technique for tuning power system stabilizer", Proceedings of the 41st International Universities Power Engineering Conference, Vol. 1, pp. 170-174, Sep. 2006.
- [8] Swasti R. Kunthia and Sidhartha Panda, "Comparative study of different controllers for automatic generation control of an interconnected hydro-thermal system with generation rate constraints", International Conference on Industrial Electronics, Control and Robotics, pp. 243-246, Dec. 2010.
- [9] Gail A. Carpenter, Stephen Grossberg, Natalya Markuzon, John H. Reynolds and David B. Rosen, "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps", IEEE Transactions on Neural Networks, Vol. 3, No. 5, pp. 698-713, Sep. 1992.
- [10] Răzvan Andonie and Lucian Sasu, "Fuzzy ARTMAP with input relevances", IEEE Transactions on Neural Networks, Vol. 17, No. 4, pp. 929-941, Jul. 2006.

- [11] Răzvan Andonie, Levente Fabry-Asztalos, Christopher Badi' Abdul-Wahid, Sarah Abdul-Wahid, Grant I. Barker and Lukas C. Magill, "Fuzzy ARTMAP prediction of biological activities for potential HIV-1 protease inhibitors using a small molecular data set", *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 8, No. 1, pp. 80-93, Jan./Feb. 2011.
- [12] Riyadh Kenaya and Rakan Chabaan, "Neural controllers for electrical power steering systems", *International Conference on Electro/Information Technology*, pp. 1-6, Oct. 2010.
- [13] Jacob E. Díaz Labariega y David Romero Romero, "Desarrollo de un regulador de voltaje para un generador síncrono usando la teoría de resonancia adaptable", Tesis de Maestría, SEPI-ESIME IPN, México, Jun. 2001.
- [14] Adrián Rubio Solís y David Romero Romero, "Control difuso autosintonizable de voltaje para un generador síncrono por medio de la búsqueda por tabú y algoritmos genéticos", Tesis de Maestría, SEPI-ESIME IPN, México, May. 2006.
- [15] Anil K. Jain and Jianchang Mao, "Artificial neural networks: A tutorial", *IEEE Computer*, Vol. 29, No. 3, pp. 31-44, Mar. 1996.
- [16] Lotfi A. Zadeh, "Fuzzy sets", *Information and Control*, 8, pp. 338-353, 1965.
- [17] Chin-Teng Lin and C. S. George Lee, "Neural fuzzy systems: A neuro-fuzzy synergism to intelligent systems", Prentice-Hall, 1996.
- [18] Gail A. Carpenter and Stephen Grossberg, "The ART of adaptive pattern recognition by a self-organizing neural network", *Computer*, 21, pp. 77-88, 1988.
- [19] Jyh-Shing R. Jang, Chuen-Tsai Sun and Eiji Mizutani, "Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence", Prentice-Hall, 1997.
- [20] Gail A. Carpenter and Stephen Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine", *Computer Vision, Graphics and Image Processing*, 37, pp. 54-115, 1987.
- [21] Gail A. Carpenter and Stephen Grossberg, "ART2: Self-organization of stable category recognition codes for analog input patterns", *Applied Optics*, Vol. 26, No. 23, pp. 4918-4930, Dec. 1987.
- [22] Gail A. Carpenter, Stephen Grossberg and David B. Rosen, "ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition", *Neural Networks*, Vol. 4, pp. 493-504, 1991.
- [23] Gail A. Carpenter and Stephen Grossberg, "ART 3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures", *Neural Networks*, Vol. 3, pp. 129-152, 1990.
- [24] Gail A. Carpenter, Stephen Grossberg and David B. Rosen, "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system", *Neural Networks*, Vol. 4, pp. 759-771, 1991.
- [25] Gail A. Carpenter and Bart Kosko, "Fuzzy ART", Technical Report, Nov. 1993.

-
- [26] Gail A. Carpenter, Stephen Grossberg and John H. Reynolds, "ARTMAP: A self-organizing neural network architecture for fast supervised learning and pattern recognition", International Joint Conference on Neural Networks, Seattle I, pp. 863-868, IEEE Service Center, 1991.
- [27] Gail A. Carpenter, Stephen Grossberg and John H. Reynolds, "ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network", Neural Networks, Vol. 4, pp. 565-588, 1991.
- [28] Daozhi Xia and G. T. Heydt, "Self-tuning controller for generator excitation control", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-102, No. 6, Jun. 1983.
- [29] J. Y. Jackson, "Interpretation and use of generator reactive capability diagrams", IEEE Transactions on Industry and General Applications, Vol. IGA-7, No. 6, Nov./Dec. 1971.
- [30] Dimiter Driankov, Hans Hellendoorn and Michael Reinfrank, "An introduction to fuzzy control", Springer, 2nd Edition, 1996.

APÉNDICE A

SISTEMA MÁQUINA BUS INFINITO BAJO PRUEBA

En este apartado se presentan el modelo matemático y los datos del sistema máquina bus infinito utilizados en el presente trabajo, los cuales fueron tomados de [13] y [28]. En la figura A.1 se muestra el diagrama de este sistema.

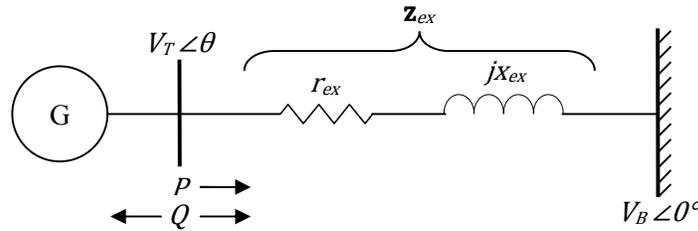


Figura A.1. Diagrama del sistema máquina bus infinito utilizado en el presente trabajo [13].

A.1 MODELO DEL GENERADOR SÍNCRONO

El modelo del generador está conformado por cinco ecuaciones diferenciales (modelo de 5° orden) y tres ecuaciones algebraicas. Estas ecuaciones son las siguientes:

$$p\delta = \omega_o s \tag{A.1}$$

$$Mps = -k_d s + T_m - T_e \tag{A.2}$$

$$T'_{do} p e'_q = e_{fd} - (x_d - x'_d) i_d - e'_q \tag{A.3}$$

$$T''_{do} p e''_q = e'_q - (x'_d - x''_d) i_d - e''_q \tag{A.4}$$

$$T_{qo}'' p e_d'' = (x_q - x_q'') i_q - e_d'' \quad (\text{A.5})$$

$$e_d'' = v_d + r_a i_d - x_q'' i_q \quad (\text{A.6})$$

$$e_q'' = v_q + r_a i_q + x_d'' i_d \quad (\text{A.7})$$

$$T_e = e_d'' i_d + e_q'' i_q - (x_d'' - x_q'') i_d i_q \quad (\text{A.8})$$

Donde:

- δ es el ángulo del rotor (ángulo de carga),
- ω_o es la frecuencia síncrona,
- s es el deslizamiento,
- M es la constante de inercia,
- k_d es el coeficiente de amortiguamiento,
- T_e y T_m son el par eléctrico y mecánico,
- e_q' es el voltaje transitorio en el eje cuadratura,
- e_d'' y e_q'' son los voltajes subtransitorios en los ejes directo y cuadratura,
- e_{fd} es el voltaje de campo,
- x_d y x_q son las reactancias en los ejes directo y cuadratura,
- x_d' y x_q' son las reactancias transitorias en los ejes directo y cuadratura,
- x_d'' y x_q'' son las reactancias subtransitorias en los ejes directo y cuadratura,
- i_d e i_q son las corrientes en los ejes directo y cuadratura,
- v_d y v_q son los voltajes en los ejes directo y cuadratura,
- r_a es la resistencia de armadura, y
- $p = \frac{d}{dt}$.

A.2 MODELO DE LA RED DE TRANSMISIÓN

El modelo de la red de transmisión por medio de la cual se conecta el generador al bus infinito es una impedancia equivalente $\mathbf{z}_{ex} = r_e + jx_e$, tal y como lo muestra la figura A.1. Por lo tanto, el voltaje en terminales y sus componentes en los ejes directo y cuadratura quedan determinados por

$$v_d = v_b \cos \delta + r_e i_d - x_e i_q, \quad (\text{A.9})$$

$$v_q = v_b \sin \delta + r_e i_q + x_e i_d \quad (\text{A.10})$$

y

$$v_t^2 = v_d^2 + v_q^2; \quad (\text{A.11})$$

donde v_b es el voltaje en el bus y v_t es el voltaje en terminales de la máquina.

A.3 MODELO DEL GOBERNADOR DE VELOCIDAD Y DE LA TURBINA DE VAPOR

El sistema bajo prueba incluye un gobernador de velocidad del tipo mecánico-hidráulico y una turbina térmica de simple recalentamiento modificada. El modelo del gobernador es idealizado; es decir, no se considera el limitador de posición de la válvula ni la función de transferencia que describe al relevador de velocidad. Por lo tanto, el modelo gobernador-turbina utilizado sólo considera los elementos necesarios para controlar la potencia activa a través de la manipulación de la entrada del gobernador de velocidad u_g sin ejercer control alguno sobre el lazo primario. En la figura A.2 se muestra el diagrama de bloques del modelo gobernador-turbina utilizado, de donde se define que las ecuaciones que describen este sistema son las siguientes:

$$T_{SM} pP_{GV} = u_g - P_{GV} \tag{A.12}$$

$$T_{CH} pP_{Hp} = P_{GV} - P_{Hp} \tag{A.13}$$

$$T_{RH} pT_m = F_{Hp} pP_{Hp} + P_{Hp} - T_m \tag{A.14}$$

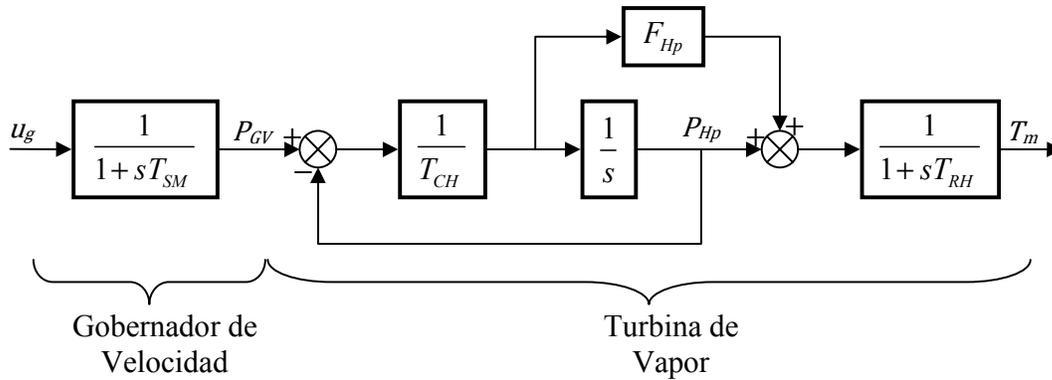


Figura A.2. Diagrama de bloques del sistema gobernador-turbina.

A.4 MODELO DEL SISTEMA DE EXCITACIÓN Y DEL REGULADOR DE VOLTAJE

El sistema de regulación utilizado es el ST1 (excitador con fuente de potencial regulada por rectificadores controlados), que es uno de los más empleados en sistemas de potencia de grandes dimensiones, dado que posee una alta capacidad de esfuerzos post-falla, un relativo bajo costo y un fácil mantenimiento. Para la obtención del modelo del este regulador, se desprecia el efecto de los transformadores de potencial que se usan para el acondicionamiento de la señal del voltaje en terminales, debido a que las constantes de tiempo de estos dispositivos son muy bajas (del orden de 10^{-4} seg.); asimismo, no se considera el efecto de la saturación de la excitatriz, puesto que los límites

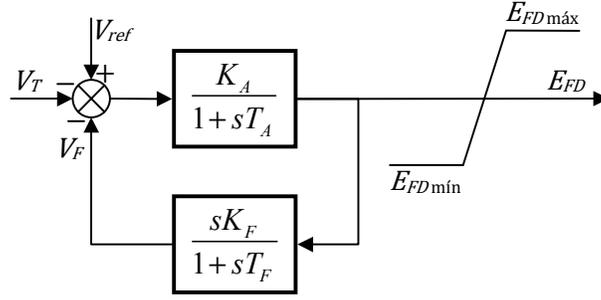


Figura A.3. Diagrama de bloques del regulador ST1 [13].

del voltaje de campo son proporcionales al voltaje generado. En base a estas consideraciones, el modelo del regulador ST1 en diagrama de bloques es el mostrado por la figura A.3, donde se puede apreciar que se omite el bloque correspondiente al excitador; esto se debe a que las constantes de tiempo inherentes al excitador en este tipo de sistemas de regulación son muy pequeñas [2]. De acuerdo con la figura A.3, las ecuaciones para el regulador ST1 son las siguientes:

$$T_A p e_{fd} = K_A (v_{ref} - v_t - v_f) - e_{fd} \quad (A.15)$$

$$T_F p v_f = K_F p e_{fd} - v_f \quad (A.16)$$

A.5 ECUACIONES DINÁMICAS

En cada iteración de la resolución numérica del sistema, es necesario determinar las corrientes en los ejes d y q para calcular el voltaje en terminales, las potencias activa y reactiva y el par eléctrico. Al sustituir la ecuación (A.9) en la (A.6) resulta

$$e_d'' = v_b \cos \delta + (r_e + r_a) i_d - (x_e + x_q'') i_q, \quad (A.17)$$

y sustituyendo la ecuación (A.10) en la (A.7) se tiene

$$e_q'' = v_b \sin \delta + (r_e + r_a) i_q + (x_e + x_d'') i_d. \quad (A.18)$$

Resolviendo simultáneamente las ecuaciones (A.17) y (A.18), para ambas corrientes se obtiene

$$i_d = \frac{(r_e + r_a)(e_d'' - v_b \cos \delta) + (x_e + x_q'')(e_q'' - v_b \sin \delta)}{(r_e + r_a)^2 + (x_e + x_d'')(x_e + x_q'')} \quad (A.19)$$

e

$$i_q = \frac{(r_e + r_a)(e_q'' - v_b \sin \delta) - (x_e + x_d'')(e_d'' - v_b \cos \delta)}{(r_e + r_a)^2 + (x_e + x_d'')(x_e + x_q'')} \quad (A.20)$$

Con lo que se puede determinar la corriente de armadura, la cual está dada en forma compleja por

$$\mathbf{i}_a = i_q + j i_d. \quad (\text{A.21})$$

Con las corrientes calculadas, los voltajes en los ejes d y q se pueden obtener directamente por las ecuaciones (A.9) y (A.10), respectivamente; o bien, despejándose de las ecuaciones (A.6) y (A.7). De cualquier modo, el voltaje en terminales se obtiene por medio de la ecuación (A.11), que en su forma compleja es

$$\mathbf{v}_t = v_q + j v_d. \quad (\text{A.22})$$

Esta expresión del voltaje en terminales, junto con la ecuación (A.21) para la corriente de armadura, resulta conveniente, puesto que para las potencias activa y reactiva se tiene que

$$\mathbf{S} = \mathbf{v}_t \mathbf{i}_a^*, \quad (\text{A.23})$$

$$P = \text{Re}(\mathbf{S}) \quad (\text{A.24})$$

y

$$Q = -\text{Im}(\mathbf{S}). \quad (\text{A.25})$$

Por último, el par eléctrico también se puede calcular una vez que se tienen las corrientes en los ejes d y q , esto se realiza de acuerdo a la ecuación (A.8).

A.6 DETERMINACIÓN DE LAS CONDICIONES INICIALES

Las condiciones iniciales del sistema se calculan en un inicio a partir de cuatro datos conocidos: el voltaje en terminales v_t , el voltaje en el bus v_b y el ángulo de éstos, θ y 0° , respectivamente. De estos datos se desarrolla lo siguiente:

$$\mathbf{v}_t = v_t \cos \theta + j v_t \sin \theta \quad (\text{A.26})$$

$$\mathbf{v}_b = v_b + j0.0 \quad (\text{A.27})$$

$$\mathbf{S} = \mathbf{v}_t \left(\frac{\mathbf{v}_t - \mathbf{v}_b}{\mathbf{z}_{ex}} \right)^* \quad (\text{A.28})$$

Las potencias activa y reactiva se determinan por las ecuaciones (A.24) y (A.25); mientras que, comparando las ecuaciones (A.23) y (A.28), para la magnitud de la corriente de armadura se tiene

$$i_a = \frac{v_t - v_b}{z_{ex}} = \frac{S}{v_t} = \frac{\sqrt{P^2 + Q^2}}{v_t}, \quad (\text{A.29})$$

y el ángulo de la misma con respecto al voltaje en terminales queda definido por

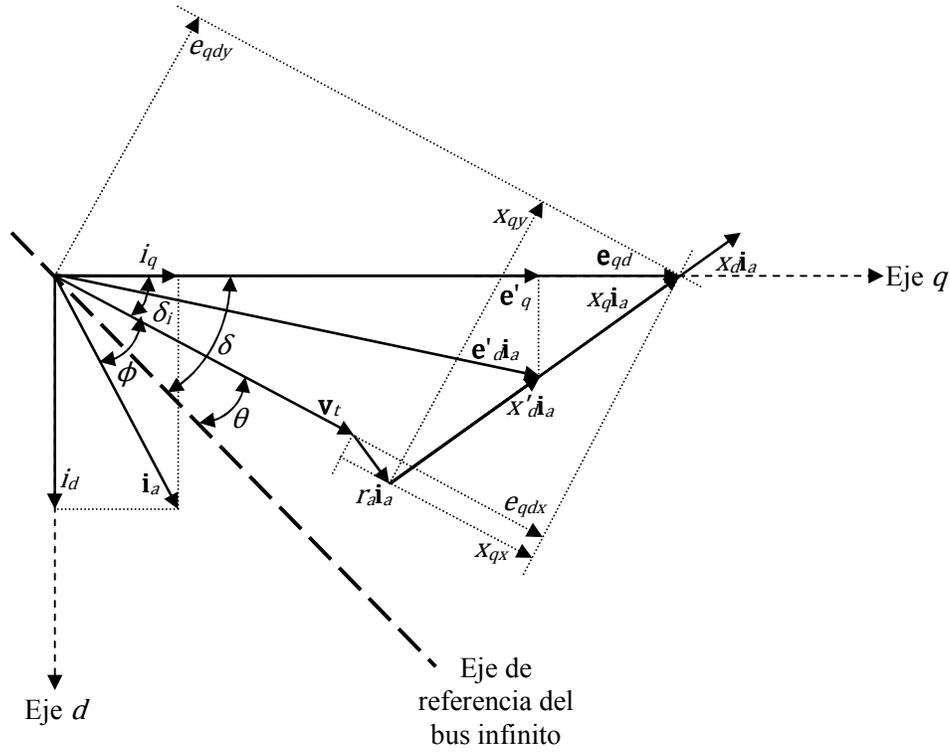


Figura A.4. Diagrama vectorial del sistema máquina bus infinito en estado subtransitorio [13].

$$\phi = \arctan \frac{Q}{P}. \quad (\text{A.30})$$

Por otra parte, del diagrama vectorial de la figura A.4, se tiene que el ángulo de carga está dado por

$$\delta = \delta_i + \theta, \quad (\text{A.31})$$

donde

$$\delta_i = \arctan \frac{e_{qdy}}{e_{qdx}}. \quad (\text{A.32})$$

Los voltajes de la ecuación (A.32) se pueden obtener también del diagrama de la figura A.4; éstos son

$$e_{qdx} = v_t + r_a i_a \cos \phi + x_q i_a \sin \phi \quad (\text{A.33})$$

y

$$e_{qdy} = -r_a i_a \sin \phi + x_q i_a \cos \phi. \quad (\text{A.34})$$

Respecto a los valores iniciales de las variables del generador, es conveniente primero determinar las corrientes en ambos ejes; éstas, según la figura A.4, están dadas por

$$i_d = i_a \sin(\delta_i + \phi) \quad (\text{A.35})$$

e

$$i_q = i_a \cos(\delta_i + \phi). \quad (\text{A.36})$$

Ahora bien, igualando a cero las derivadas de las ecuaciones (A.4) y (A.5) y despejando los voltajes subtransitorios en los ejes d y q , respectivamente, se tiene que

$$e_d'' = x_q i_q - x_q'' i_q \quad (\text{A.37})$$

y

$$e_q'' = e_q' - x_d' i_d + x_d'' i_d; \quad (\text{A.38})$$

pero, si se sustituyen las ecuaciones (A.35) y (A.36) en el segundo término del segundo miembro de la ecuación (A.38) y en el primer término del segundo miembro de la ecuación (A.37), respectivamente, entonces se tiene que

$$e_d'' = x_q i_a \cos(\delta_i + \phi) - x_q'' i_q \quad (\text{A.39})$$

y

$$e_q'' = e_q' - x_d' i_a \sin(\delta_i + \phi) + x_d'' i_d. \quad (\text{A.40})$$

En el diagrama de la figura A.4 se puede observar que $x_q i_a \cos(\delta_i + \phi) = v_t \sin \delta_i + r_a i_d$, y que $e_q' - x_d' i_a \sin(\delta_i + \phi) = v_t \cos \delta_i + r_a i_d$. Por lo tanto, los voltajes subtransitorios en ambos ejes quedan determinados finalmente por

$$e_d'' = v_t \sin \delta_i + r_a i_d - x_q'' i_q \quad (\text{A.41})$$

y

$$e_q'' = v_t \cos \delta_i + r_a i_d + x_d'' i_d. \quad (\text{A.42})$$

De la misma ecuación (A.4) se despeja el voltaje transitorio en el eje q después de igualar la derivada a cero, quedando

$$e_q' = e_q'' + (x_d' - x_d'') i_d. \quad (\text{A.43})$$

El deslizamiento, el voltaje de campo y el par mecánico del generador se obtienen de forma similar; es decir, igualando a cero las derivadas de las ecuaciones (A.1)-(A.3) y despejando la variable en cuestión. Así, se tiene que

$$s = 0, \quad (\text{A.44})$$

$$e_{fd} = e_q' + (x_d - x_d') i_d \quad (\text{A.45})$$

y

$$T_m = T_e = e_d'' i_d + e_q'' i_q - (x_d'' - x_q'') i_d i_q. \quad (\text{A.46})$$

En cuanto a las condiciones iniciales de las variables del modelo gobernador-turbina, al igualar a cero las derivadas de las ecuaciones (A.12)-(A.14) y despejando resulta lo siguiente:

$$P_{GV} = u_g \quad (\text{A.47})$$

$$P_{Hp} = P_{GV} \quad (\text{A.48})$$

$$T_m = P_{Hp} \quad (\text{A.49})$$

$$\therefore u_g = T_m = T_e \quad (\text{A.50})$$

Por último, al igualar a cero las derivadas de las ecuaciones (A.15) y (A.16) y al despejar las variables correspondientes, resulta que las condiciones iniciales para el regulador ST1 son las siguientes:

$$v_f = 0 \quad (\text{A.51})$$

$$v_{ref} = \frac{e_{fd}}{K_A} + v_t \quad (\text{A.52})$$

A.7 DATOS DEL SISTEMA BAJO PRUEBA

Los datos utilizados corresponden a un sistema de generación de 645 MVA conectado a un bus infinito a través de una línea corta representada por una impedancia equivalente $\mathbf{z}_e = r_e + jx_e$, tal y como lo muestra la figura A.1. A continuación se muestran estos datos cuyos valores se encuentran en p.u., a menos que se indiquen las unidades.

- Generador

$$\begin{aligned} x_d &= 1.904 & x_q'' &= 0.266 & T_{do}'' &= 0.041 \text{ seg.} \\ x_d' &= 0.312 & r_a &= 0.0 & T_{qo}'' &= 0.065 \text{ seg.} \\ x_d'' &= 0.266 & k_d &= 3.0 & M &= 5.5294 \\ x_q &= 1.881 & T_{do}' &= 5.66 \text{ seg.} & \omega_o &= 377 \text{ rad./seg.} \end{aligned}$$

- Red de Transmisión

$$r_e = 0.024 \quad x_e = 0.115$$

- Gobernador y Turbina

$$\begin{aligned} T_{SM} &= 0.1 \text{ seg.} & T_{RH} &= 5.0 \text{ seg.} \\ T_{CH} &= 0.15 \text{ seg.} & F_{Hp} &= 0.33 \end{aligned}$$

- Regulador de Voltaje

$$\begin{aligned} K_A &= 400 & K_F &= 0.008 \\ T_A &= 0.02 \text{ seg.} & T_F &= 1.0 \text{ seg.} \end{aligned}$$

APÉNDICE B

CONDICIONES INICIALES

En la tabla B.1 se muestra el conjunto de condiciones iniciales utilizado en este trabajo.

Tabla B.1. Condiciones iniciales consideradas [13].

| Número de Condición Inicial | Voltaje en Terminales (p.u.) | Ángulo de Carga (grados) | Potencia Activa (p.u.) | Potencia Reactiva (p.u.) |
|------------------------------------|-------------------------------------|---------------------------------|-------------------------------|---------------------------------|
| 1 | 1.0000000 | 71.8097608 | 0.8805000 | -0.1361000 |
| 2 | 1.0426910 | 53.0227670 | 0.8971906 | 0.2394987 |
| 3 | 1.0291030 | 46.5707683 | 0.6517569 | 0.1460104 |
| 4 | 0.9998438 | 65.8952537 | 0.7357140 | -0.1215235 |
| 5 | 1.0152080 | 53.4362629 | 0.6588190 | 0.0210546 |
| 6 | 1.0453240 | 43.0389266 | 0.7031847 | 0.2880037 |
| 7 | 0.9813626 | 69.8150523 | 0.5678285 | -0.2549196 |
| 8 | 1.0332250 | 47.3260930 | 0.6972846 | 0.1772876 |
| 9 | 1.0563940 | 38.2865221 | 0.6808600 | 0.3954524 |
| 10 | 1.0016400 | 52.8155086 | 0.5273286 | -0.0787794 |
| 11 | 1.0270150 | 49.5169576 | 0.6890222 | 0.1221480 |
| 12 | 1.0389280 | 40.2191230 | 0.6077517 | 0.2420443 |
| 13 | 0.9919246 | 62.1324364 | 0.5712479 | -0.1675409 |
| 14 | 1.0180660 | 63.1788147 | 0.8913938 | 0.0185088 |
| 15 | 1.0493000 | 53.9286729 | 0.9793495 | 0.2917894 |
| 16 | 0.9919636 | 74.4397104 | 0.8302205 | -0.1984313 |
| 17 | 1.0485680 | 52.3647172 | 0.9334847 | 0.2899110 |
| 18 | 1.0161500 | 52.1678756 | 0.6448828 | 0.0312059 |
| 19 | 1.0123020 | 62.2518490 | 0.8040097 | -0.0222813 |
| 20 | 1.0179620 | 60.9635777 | 0.8376989 | 0.0234341 |

Tabla B.1. Condiciones iniciales consideradas (continuación).

| Número de Condición Inicial | Voltaje en Terminales (p.u.) | Ángulo de Carga (grados) | Potencia Activa (p.u.) | Potencia Reactiva (p.u.) |
|------------------------------------|-------------------------------------|---------------------------------|-------------------------------|---------------------------------|
| 21 | 1.0427960 | 35.8111620 | 0.5543289 | 0.2858777 |
| 22 | 1.0470590 | 40.1914882 | 0.6595582 | 0.3102633 |
| 23 | 0.9967496 | 68.6110830 | 0.7552863 | -0.1500301 |
| 24 | 1.0095190 | 62.6945098 | 0.7821750 | -0.0439321 |
| 25 | 1.0594970 | 39.0086232 | 0.7149498 | 0.4203133 |
| 26 | 0.9826534 | 65.7404267 | 0.5175695 | -0.2373971 |
| 27 | 1.0168930 | 65.8037201 | 0.9445974 | 0.0027810 |
| 28 | 1.0000000 | 75.7887306 | 1.0000000 | -0.1473878 |
| 29 | 0.9896917 | 79.0575924 | 0.9319504 | -0.2272967 |
| 30 | 0.9904293 | 79.4581517 | 0.9592349 | -0.2236688 |

APÉNDICE C

PROGRAMA DESARROLLADO

En este segmento se presenta el programa desarrollado para la obtención de los resultados presentados en los Capítulos 3 y 4. Este programa es el resultado de numerosas pruebas realizadas con la finalidad de obtener los mejores resultados posibles en diferentes escenarios; es decir, simulando el sistema máquina bus infinito bajo prueba desde diferentes condiciones iniciales y ante diversos eventos o disturbios.

El lenguaje computacional utilizado es FORTRAN 90 bajo el software Compaq Visual Fortran versión 6.6.

C.1 EJECUCIÓN DEL PROGRAMA

En el momento de ejecutar el programa, en pantalla aparecen tres modos principales para simular el generador bajo prueba, los cuales son: sin regulador, con regulador ST1 y con la arquitectura FAM. Después de esto, se desglosa una serie de opciones y parámetros a definir durante la ejecución. En la figura C.1 se muestra un diagrama de bloques de las opciones que ofrece el programa en pantalla, donde los números indican el valor que se debe ingresar para elegir la opción deseada. Cabe señalar que, en caso de seleccionar el modo de operación del generador sin regulador, el programa no ofrece la posibilidad de simular cambios de referencia, evidentemente porque el voltaje de campo es constante y no existe medio alguno por el que se pueda modificar la referencia de entrada.

En las figuras C.2-C.4 se muestran tres ejemplos de ejecución del programa. La figura C.2 muestra la pantalla de ejecución resultante de simular el generador sin regulador, con un inicio y un fin de simulación en $t = 0.0$ seg. y $t = 12.0$ seg., respectivamente. En este ejemplo se simula una falla en el bus que inicia en $t = 2.0$ seg. con 8 ciclos de duración, la cual incluye aislamiento de falla. La condición inicial seleccionada es la No. 1 de la tabla B.1.

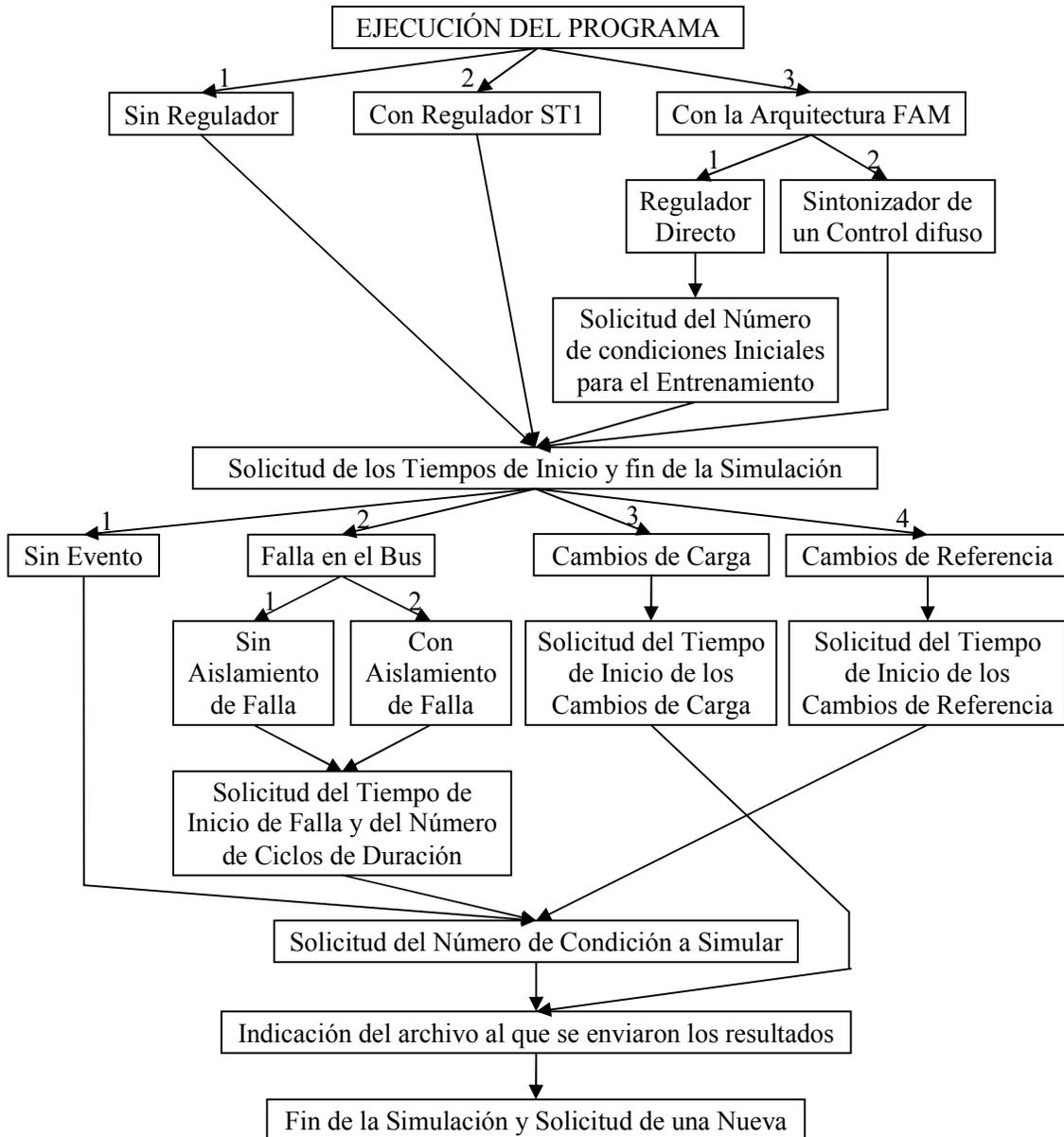


Figura C.1. Diagrama de bloques de las opciones que ofrece el programa desarrollado en tiempo de ejecución.

En la figura C.3 se muestra la pantalla resultante de simular el generador con el regulador de voltaje ST1. El inicio de la simulación es en $t = 0.0$ seg. y el fin de la misma es en $t = 120.0$ seg. El evento elegido corresponde a los cambios de carga con un inicio en $t = 10.0$ seg., de aquí que el tiempo de simulación sea muy largo, puesto que este evento así lo requiere. En este caso, no se necesita especificar el número de condición inicial a simular, dado que se considera que el generador no genera potencia activa ni absorbe o provee reactivos desde o hacia el bus infinito; por lo tanto, las condiciones iniciales, a excepción del voltaje en terminales, son cero.

```

"C:\Control_FAM\Debug\Control_FAM.exe"
          INSTITUTO POLITECNICO NACIONAL
          ESCUELA SUPERIOR DE INGENIERIA MECANICA Y ELECTRICA
          SECCION DE ESTUDIOS DE POSGRADO E INVESTIGACION

PROGRAMA PARA SIMULAR UN GENERADOR SINCRONO SIN REGULADOR DE VOLTAJE, CON UN
REGULADOR ST1 O CON UN REGULADOR NEURODIFUSO UTILIZANDO LA ARQUITECTURA
ARTMAP DIFUSA <FAM>

Indique el tipo de regulador para el Generador
[1] Sin Regulador [2] Con Regulador ST1 [3] Con la Arquitectura FAM: 1

    Tiempo inicial de simulacion = 0.0
    Tiempo final de simulacion = 12.0

Indique el tipo de evento que se presentara durante la simulacion
[1] Sin evento [2] Falla en el bus [3] Cambios de carga: 2

Indique el tipo de falla
[1] Sin aislamiento de falla [2] Con aislamiento de falla: 2

    Tiempo de inicio de falla = 2.0
    Numero de ciclos de duracion de la falla = 8.0

Indique el numero de condicion inicial a simular <1 - 30>: 1
Los resultados se enviaron al archivo res_sc_fb.dat

Desea realizar otra simulacion? [s] [n]: _

```

Figura C.2. Pantalla resultante de simular el generador sin regulador, ante una falla en el bus incluyendo aislamiento de falla, para la condición inicial No. 1.

```

"C:\Control_FAM\Debug\Control_FAM.exe"
          INSTITUTO POLITECNICO NACIONAL
          ESCUELA SUPERIOR DE INGENIERIA MECANICA Y ELECTRICA
          SECCION DE ESTUDIOS DE POSGRADO E INVESTIGACION

PROGRAMA PARA SIMULAR UN GENERADOR SINCRONO SIN REGULADOR DE VOLTAJE, CON UN
REGULADOR ST1 O CON UN REGULADOR NEURODIFUSO UTILIZANDO LA ARQUITECTURA
ARTMAP DIFUSA <FAM>

Indique el tipo de regulador para el Generador
[1] Sin Regulador [2] Con Regulador ST1 [3] Con la Arquitectura FAM: 2

    Tiempo inicial de simulacion = 0.0
    Tiempo final de simulacion = 120.0

Indique el tipo de evento que se presentara durante la simulacion
[1] Sin evento [2] Falla en el bus [3] Cambios de carga
[4] Cambios de referencia: 3

    Tiempo de inicio de los cambios de carga = 10.0
Los resultados se enviaron al archivo res_st1_cc.dat

Desea realizar otra simulacion? [s] [n]: _

```

Figura C.3. Pantalla resultante de simular el generador con regulador ST1 ante cambios de carga.

```

"C:\Control_FAM\Debug\Control_FAM.exe"
INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE INGENIERIA MECANICA Y ELECTRICA
SECCION DE ESTUDIOS DE POSGRADO E INVESTIGACION

PROGRAMA PARA SIMULAR UN GENERADOR SINCRONO SIN REGULADOR DE VOLTAJE. CON UN
REGULADOR ST1 O CON UN REGULADOR NEURODIFUSO UTILIZANDO LA ARQUITECTURA
ARTMAP DIFUSA (FAM)

Indique el tipo de regulador para el Generador
[1] Sin Regulador [2]Con Regulador ST1 [3] Con la Arquitectura FAM: 3

Indique la operacion de la FAM
[1] Regulador directo [2] Sintonizador de un control difuso: 2

El entrenamiento se esta realizando, espere...

Se ha concluido el entrenamiento de la FAM.

Tiempo inicial de simulacion = 0.0
Tiempo final de simulacion = 60.0

Indique el tipo de evento que se presentara durante la simulacion
[1] Sin evento [2] Falla en el bus [3] Cambios de carga
[4] Cambios de referencia: 4

Tiempo de inicio de los cambios de referencia = 10.0

Indique el numero de condicion inicial a simular (1 - 30): 25

Los resultados se enviaron al archivo res_fam_cr.dat

Desea realizar otra simulacion? [s] [n]: _

```

Figura C.4. Pantalla resultante de simular el generador con la arquitectura FAM actuando como sintonizador de un controlador difuso, ante cambios de referencia para la condición inicial No. 25.

En la figura C.4 se muestra la pantalla que resulta de simular el generador con la arquitectura FAM actuando como sintonizador de un control difuso. La simulación inicia en $t = 0.0$ seg. y finaliza en $t = 60.0$ seg. El evento a suscitarse es el correspondiente a los cambios de referencia con un inicio en $t = 10.0$ seg. La condición inicial a simular es la No. 25 de la tabla B.1.

C.2 CÓDIGO DEL PROGRAMA

El programa desarrollado consta de las siguientes partes principales:

- Programa Principal.
- Módulos de Declaración de Variables.
- Subrutinas referentes a la Máquina Síncrona.
- Subrutinas referentes a la Arquitectura FAM.
- Subrutinas referentes a los Eventos a Simular.

C.2.1 Programa Principal

El programa principal contiene la lógica de ejecución del programa, las estructuras, los llamados a las diferentes subrutinas, los ciclos de solución de la máquina síncrona y las indicaciones para la escritura de los resultados. El código se presenta a continuación.

```

|*****
|***** PROGRAMA PRINCIPAL *****
PROGRAM CONTROL_FAM

|*****
|*****
|***                                     ***
|*** PROGRAMA PARA SIMULAR UN GENERADOR SÍNCRONO SIN REGULADOR DE VOLTAJE, ***
|*** CON REGULADOR ST1 Y CON LA ARQUITECTURA ARTMAP DIFUSA, YA SEA ***
|*** DIRECTAMENTE COMO REGULADOR O COMO SINTONIZADOR DE UN CONTROLADOR ***
|*** DIFUSO PI TIPO MAMDANI ***
|***                                     ***
|***                                     ***
|***                                     ***
|*** AUTORIZADO POR: MARTÍN EMMANUEL CÁRDENAS RIVAS ***
|*** FECHA DE ELABORACIÓN: MAYO DE 2010 - ENERO DE 2011 ***
|*** LUGAR DE ELABORACIÓN: SEPI-ESIME ZACATENCO, IPN ***
|*** BAJO LA DIRECCIÓN DE: DR. DAVID ROMERO ROMERO ***
|***                                     ***
|*****
|*****

USE Parametros
USE Var_FAM
IMPLICIT NONE
Real(8) h, tnic, tfin, tif, ciclos, tff, tc1, tcr1
Real(8) e, de, inte, dvf, vainic, vfinic, inva
Real(8) emax, emin, demax, demin, intemax, intemin, dvfmax, dvfmin
Real(8) emaxg, eming, demaxg, deming, intemaxg, inteming, dvfmaxg, dvfming
Real(8),Allocatable :: t(:)

|***** VARIABLES AUXILIARES *****
Integer i, j, l, modo_maq, n, ncon, evento, nconen, appfam, con, modo_falla
Character(1) ejec
Real(8) PARAM(30), CITOT(4,31), CIACT(4), men, cataux, eaux, vrefaux
|*****
|***** VARIABLES DE LA MÁQUINA *****
Real(8) vbus, va, thet, p, q, ug, id, iq, te, vf, vref
Real(8),Dimension(nec) :: ai, ai1, ai2, ai3, ai4, ai5
|*****
|***** VARIABLES PARA EL CONTROL DIFUSO *****
Real(8) esup, desup, mue1, mue2, mue3, mude1, mude2, mude3
Real(8) NG, NP, CE, PP, PG, X1, X2
|*****

Write(6,*) ' INSTITUTO POLITECNICO NACIONAL '
Write(6,*) ' ESCUELA SUPERIOR DE INGENIERIA MECANICA Y ELECTRICA '
Write(6,*) ' SECCION DE ESTUDIOS DE POSGRADO E INVESTIGACION '
Write(*,*)
Write(*,*)
Write(6,*) 'PROGRAMA PARA SIMULAR UN GENERADOR SINCRONO SIN REGULADOR DE VOLTAJE, CON UN '
Write(6,*) 'REGULADOR ST1 O CON UN REGULADOR NEURODIFUSO UTILIZANDO LA ARQUITECTURA '
Write(6,*) 'ARTMAP DIFUSA (FAM)'

|***** LECTURA DE CONDICIONES INICIALES *****
Open(Unit=1, File="cond_inic.dat")
Read(1,101) ((CITOT(j,i), j=1, 4), i=1, 31)
101 Format(4f11.7)
Close(Unit=1)
|*****
|***** LECTURA Y ASIGNACIÓN DE LOS PARÁMETROS DE LA MÁQUINA *****
Open (Unit=2, File="parametros.dat")

```

```

Read(2,102) (PARAM(i), i=1, 30)
102 Format(f8.4)
VMAX = PARAM(1)
VMIN = PARAM(2)
raa = PARAM(3)
re = PARAM(4)
xe = PARAM(5)
kd = PARAM(6)
w0 = PARAM(7)
M = PARAM(8)
tsm = PARAM(9)
tch = PARAM(10)
trh = PARAM(11)
fhp = PARAM(12)
tx = PARAM(13)
tq02p = PARAM(14)
td0p = PARAM(15)
td02p = PARAM(16)
xd = PARAM(17)
xdp = PARAM(18)
xd2p = PARAM(19)
xq = PARAM(20)
xq2p = PARAM(21)
KAA = PARAM(22)
ta = PARAM(23)
kf = PARAM(24)
tf = PARAM(25)
raz1 = PARAM(26)
raz2 = PARAM(27)
raz3 = PARAM(28)
raz4 = PARAM(29)
raz5 = PARAM(30)
Close (Unit=2)
!*****
h = 0.0010
vbus = 1.0
ejec = 's'
!***** LETREROS DURANTE LA EJECUCIÓN *****
10 Format(//,1X,'Indique el tipo de regulador para el Generador')
11 Format(1X,'[1] Sin Regulador [2]Con Regulador ST1 [3] Con la Arquitectura FAM: ',\ )
12 Format(/,4X,'Tiempo inicial de simulacion = ',\ )
13 Format(4X,'Tiempo final de simulacion = ',\ )
16 Format(/,1X,'Indique el tipo de evento que se presentara durante la simulacion')
17 Format(1X,'[1] Sin evento [2] Falla en el bus [3] Cambios de carga: '\ )
18 Format(1X,'[1] Sin evento [2] Falla en el bus [3] Cambios de carga'&
& ,/ ,1X,'[4] Cambios de referencia: ',\ )
19 Format(/,1X,'Indique el tipo de falla')
20 Format(1X,'[1] Sin aislamiento de falla [2] Con aislamiento de falla: '\ )
21 Format(/,4X,'Tiempo de inicio de falla = ',\ )
22 Format(4X,'Numero de ciclos de duracion de la falla = ',\ )
23 Format(/,4X,'Tiempo de inicio de los cambios de carga = ',\ )
24 Format(/,4X,'Tiempo de inicio de los cambios de referencia = ',\ )
25 Format(/,1X,'Indique el numero de condicion inicial a simular (1 - 30): ',\ )
26 Format(/,1X,'Los resultados se enviaron al archivo res_sc_se.dat',/)
27 Format(/,1X,'Los resultados se enviaron al archivo res_sc_fb.dat',/)
28 Format(/,1X,'Los resultados se enviaron al archivo res_sc_cc.dat',/)
29 Format(/,1X,'Los resultados se enviaron al archivo res_st1_se.dat',/)
30 Format(/,1X,'Los resultados se enviaron al archivo res_st1_fb.dat',/)
31 Format(/,1X,'Los resultados se enviaron al archivo res_st1_cc.dat',/)
32 Format(/,1X,'Los resultados se enviaron al archivo res_st1_cr.dat',/)
33 Format(/,1X,'Los resultados se enviaron al archivo res_fam_se.dat',/)
34 Format(/,1X,'Los resultados se enviaron al archivo res_fam_fb.dat',/)
35 Format(/,1X,'Los resultados se enviaron al archivo res_fam_cc.dat',/)
36 Format(/,1X,'Los resultados se enviaron al archivo res_fam_cr.dat',/)
37 Format(/,1X,'Indique la operacion de la FAM')
38 Format(1X,'[1] Regulador directo [2] Sintonizador de un control difuso: ',\ )
39 Format(/,1X,'Numero de condiciones iniciales para el entrenamiento: ',\ )
40 Format(/,1X,'El entrenamiento se esta realizando, espere...'\ )
41 Format(//,1X,'Se ha concluido el entrenamiento de la FAM.')

```

```

42 Format(/,1X,'Desea realizar otra simulacion? [s] [n]: ',\ )
|*****
Write(6,10)
Write(6,11)
Read(5,*) modo_maq
|***** ESPECIFICACIÓN DE FORMATOS PARA LOS RESULTADOS *****
201 Format(f8.4,7f16.10)
202 Format(f8.4,8f14.10)
301 Format(/,1X,'eming = 'f15.10,10X,'emaxg = 'f15.10)
302 Format(1X,'deming = 'f15.10,9X,'demaxg = 'f15.10)
303 Format(1X,'inteming = 'f15.10,7X,'intemaxg = 'f15.10)
304 Format(/,1X,'dvfming = 'f15.10,10X,'dvfmaxg = 'f15.10,2/)
305 Format(<kamax>i7)
306 Format(<kamax>f7.4)
307 Format(<kbmax>f7.4)
308 Format(<kbmax>f14.10)
309 Format(f8.4,<2*ma>f14.10,<2*mb>f14.10,2i4)
|*****
!** SE PRESENTAN LOS TRES CASOS: SIN REGULADOR, CON REGULADOR ST1 Y CON LA FAM **
IF ( (modo_maq.EQ.1).OR.(modo_maq.EQ.2) ) THEN
    DO WHILE (ejec.EQ.'s')
        re = PARAM(4)
        xe = PARAM(5)
        Write(6,12)
        Read(5,*) tinic
        Write(6,13)
        Read(5,*) tfin
        Write(6,16)
        If (modo_maq.EQ.1) then
            Write(6,17)
            Read(5,*) evento
        Else
            Write(6,18)
            Read(5,*) evento
        End if
        If (evento.EQ.2) then
            Write(6,19)
            Write(6,20)
            Read(5,*) modo_falla
            Write(6,21)
            Read(5,*) tif
            Write(6,22)
            Read(5,*) ciclos
            tff = tif + ciclos/60.0
        Else if (evento.EQ.3) then
            Write(6,23)
            Read(5,*) tc1
            tc2 = tc1 + 20.0
            tc3 = tc2 + 20.0
            tc4 = tc3 + 20.0
        Else if (evento.EQ.4) then
            Write(6,24)
            Read(5,*) tcr1
            tcr2 = tcr1 + 30.0
        End if
        n = (tfin - tinic)/h + 2
        Allocate(t(n))
        Do i=1, n
            t(i) = 0.0
        End do
        t(1) = tinic
        If (evento.EQ.3) then
            Do i=1, 4
                CIACT(i) = CITOT(i,31)
            End do
        Else
            Write(6,25)
            Read(5,*) ncon
            Do i=1, 4

```

```

                CIACT(i) = CITOT(i,ncon)
            End do
        End if
!***** SIN REGULADOR *****
If (modo_maq.EQ.1) then
    Open(Unit=11, File="res_sc_se.dat")
    Open(Unit=12, File="res_sc_fb.dat")
    Open(Unit=13, File="res_sc_cc.dat")
    Call condicion_inicial1(va,thet,te,id,iq,vf,p,q,ug,ai,ciact)
    vainic = va
    e = vainic - va
    inva = e**2
    If (evento.EQ.1) then
        Write(11,201) t(1), va, inva, vf, ai(1), p, q, vainic
    else if (evento.EQ.2) then
        Write(12,201) t(1), va, inva, vf, ai(1), p, q, vainic
    else if (evento.EQ.3) then
        Write(13,201) t(1), va, inva, vf, ai(1), p, q, ug
    End if
    Do i=2, n
        t(i) = t(i-1) + h
        tinic = t(i)
        If (evento.EQ.2) then
            Call falla(tinic,tif,tff,vbus,modo_falla)
        else if (evento.EQ.3) then
            Call cambio_carga(tinic,tc1,ug,h)
        End if
        Call kutta1(ai,id,iq,te,va,ug,vf,h)
        Call dinamica(ai,id,iq,te,va,p,q,vbus)
        e = vainic - va
        inva = inva + e**2
        If (evento.EQ.1) then
            Write(11,201) t(i), va, inva, vf, ai(1), p, q, vainic
        else if (evento.EQ.2) then
            Write(12,201) t(i), va, inva, vf, ai(1), p, q, vainic
        else if (evento.EQ.3) then
            Write(13,201) t(i), va, inva, vf, ai(1), p, q, ug
        End if
    End do
    If (evento.EQ.1) then
        Write(6,26)
    else if (evento.EQ.2) then
        Write(6,27)
    else if (evento.EQ.3) then
        Write(6,28)
    End if
    Close(Unit=11)
    Close(Unit=12)
    Close(Unit=13)
!***** CON REGULADOR ST1 *****
Else
    Open(Unit=21, File="res_st1_se.dat")
    Open(Unit=22, File="res_st1_fb.dat")
    Open(Unit=23, File="res_st1_cc.dat")
    Open(Unit=24, File="res_st1_cr.dat")
    Call condicion_inicial2(va,thet,te,id,iq,vref,p,q,ug,ai,ciact)
    vainic = va
    e = vainic - va
    vrefaux = vref
    inva = e**2
    If (evento.EQ.1) then
        Write(21,201) t(1), va, inva, ai(6), ai(1), p, q, vref
    else if (evento.EQ.2) then
        Write(22,201) t(1), va, inva, ai(6), ai(1), p, q, vref
    else if (evento.EQ.3) then
        Write(23,201) t(1), va, inva, ai(6), ai(1), p, q, ug
    else if (evento.EQ.4) then
        Write(24,201) t(1), va, inva, ai(6), ai(1), p, q, vref
    End if

```

```

End if
Do i=2, n
  t(i) = t(i-1) + h
  tinc = t(i)
  If (evento.EQ.2) then
    Call falla(tinc,tif,tff,vbus,modo_falla)
  else if (evento.EQ.3) then
    Call cambio_carga(tinc,tc1,ug,h)
  else if (evento.EQ.4) then
    Call cambio_ref(tinc,tcr1,vref,vrefaux,h)
  End if
  Call kutta2(ai,id,iq,te,va,ug,vref,h)
  Call dinamica(ai,id,iq,te,va,p,q,vbus)
  e = vainic - va
  inva = inva + e**2
  If (evento.EQ.1) then
    Write(21,201) t(i), va, inva, ai(6), ai(1), p, q, vref
  else if (evento.EQ.2) then
    Write(22,201) t(i), va, inva, ai(6), ai(1), p, q, vref
  else if (evento.EQ.3) then
    Write(23,201) t(i), va, inva, ai(6), ai(1), p, q, ug
  else if (evento.EQ.4) then
    Write(24,201) t(i), va, inva, ai(6), ai(1), p, q, vref
  End if
End do
End If
If (evento.EQ.1) then
  Write(6,29)
else if (evento.EQ.2) then
  Write(6,30)
else if (evento.EQ.3) then
  Write(6,31)
else if (evento.EQ.4) then
  Write(6,32)
End if
Close(Unit=21)
Close(Unit=22)
Close(Unit=23)
Close(Unit=24)
End If
!*****
Deallocate(t)
Write(6,42)
Read(5,*) ejec
END DO
!***** CON LA ARQUITECTURA FAM *****
ELSE
  !** DEFINICIÓN DE LOS PARÁMETROS Y DEL TAMAÑO **
  !** DE LOS VECTORES Y DE LAS MATRICES DE LA FAM **
  Write(6,37)
  Write(6,38)
  Read(5,*) appfam
  If (appfam.EQ.1) then
    Write(6,39)
    Read(5,*) nconen
    ma = 3
  else
    nconen = 10
    ma = 2
  end if
  mb = 1
  na = 292
  nb = 50
  eps = 0.01
  beta = 1.0
  rabs = 0.9750
  ra = 0.0
  rb = 0.9750
  rab = 0.9750
  ka = 0

```

```

kb = 0
resa = 1
resb = 1
resab = 1
delta = 0.01
kamax = 1
Allocate(IA(ma), IB(mb), A(2*ma), B(2*mb), BAUX(2*mb), WA(na,2*ma), WB(nb,2*mb), WAB(na,nb))
Allocate(WJ(2*ma), WK(2*mb), YB(nb), XAB(nb), TJ(na), TJAUX(na), TK(nb), TKAUX(nb))
Allocate(CB(nb))
!*****
!***** INICIALIZACIÓN DE PESOS Y VECTORES *****
Do i=1, ma
    IA(i) = 0.0
End do
Do i=1, mb
    IB(i) = 0.0
End do
Do i=1, 2*ma
    A(i) = 0.0
    WJ(i) = 0.0
End do
Do i=1, 2*mb
    B(i) = 0.0
    BAUX(i) = 0.0
    WK(i) = 0.0
End do
Do i=1, 2*ma
    Do j=1, na
        WA(j,i) = 1.0
    End do
End do
Do i=1, 2*mb
    Do j=1, nb
        WB(j,i)=1.0
    End do
End do
Do i=1, nb
    Do j=1, na
        WAB(j,i)=1.0
    End do
    CB(i) = 0.0
End do
!*****
Write(6,40)
tfin = 12.0
tif = 5.0
ciclos = 6.0
tff = tif + ciclos/60.0
n = tfin/h + 1
modo_fam = 1
modo_falla = 1
Allocate(t(n))
!*** OBTENCIÓN DE LOS MÁXIMOS Y MÍNIMOS GLOBALES ***
DO j=1, nconen
    Do i=1, n
        t(i) = 0.0
    End do
    Do i=1, 4
        CIACT(i) = CITOT(i,j)
    End do
    Call condicion_inicial2(va,thet,te,id,iq,vref,p,q,ug,ai,ciact)
    vainic = va
    e = vainic - va
    eaux = e
    de = e - eaux
    inte = ai(6)
    vfinic = ai(6)
    dvf = ai(6) - vfinic
    If (j.EQ.1) then

```

```

        emaxg = e
        eming = e
        demaxg = de
        deming = de
        intemaxg = inte
        inteming = inte
        dvfmaxg = dvf
        dvfming = dvf
End if
emax = e
emin = e
demax = de
demin = de
intemax = inte
intemin = inte
dvfmax = dvf
dvfmin = dvf
Do i=2, n
    t(i) = t(i-1) + h
    tinic = t(i)
    Call falla(tinic,tif,tff,vbus,modo_falla)
    Call kutta2(ai,id,iq,te,va,ug,vref,h)
    Call dinamica(ai,id,iq,te,va,p,q,vbus)
    eaux = e
    e = vainic - va
    inte = inte + e
    de = e - eaux
    dvf = ai(6) - vfinic
    If (e.GT.emax) then
        emax = e
    else if (e.LT.emin) then
        emin = e
    End if
    If (de.GT.demax) then
        demax = de
    else if (de.LT.demin) then
        demin = de
    End if
    If (inte.GT.intemax) then
        intemax = inte
    else if (inte.LT.intemin) then
        intemin = inte
    End if
    If (dvf.GT.dvfmax) then
        dvfmax = dvf
    else if (dvf.LT.dvfmin) then
        dvfmin = dvf
    End if
End do
If (emax.GT.emaxg) then
    emaxg = emax
End if
If (emin.LT.eming) then
    eming = emin
End if
If (demax.GT.demaxg) then
    demaxg = demax
End if
If (demin.LT.deming) then
    deming = demin
End if
If (intemax.GT.intemaxg) then
    intemaxg = intemax
End if
If (intemin.LT.inteming) then
    inteming = intemin
End if
If (dvfmax.GT.dvfmaxg) then
    dvfmaxg = dvfmax

```

```

        End if
        If (dvfmin.LT.dvfmimg) then
            dvfmimg = dvfmin
        End if
    END DO
    Open(Unit=50, File="pesos_fam.dat")
    Write(50,*)
    Write(50,*) '===== MÍNIMOS Y MÁXIMOS GLOBALES DE LAS VARIABLES DE ENTRADA ====='
    Write(50,301) eming, emaxg
    Write(50,302) deming, demaxg
    If (appfam.EQ.1) then
        Write(50,303) inteming, intemaxg
    End if
    Write(50,304) dvfmimg, dvfmaxg
    Write(50,*)
    emaxg = DMAX1( DABS(emaxg),DABS(eming) )
    demaxg = DMAX1( DABS(demaxg),DABS(deming) )
    intemaxg = DMAX1( DABS(intemaxg),DABS(inteming) )
    dvfmaxg = DMAX1( DABS(dvfmaxg),DABS(dvfmimg) )
    !*****
    !***** SE REALIZA EL ENTRENAMIENTO DE LA FAM *****
    modo_fam = 1
    DO j=1, nconen
        Do i=1, n
            t(i) = 0.0
        End do
        Do i=1, 4
            CIACT(i) = CITOT(i,j)
        End do
        Call condicion_inicial2(va,thet,te,id,iq,vref,p,q,ug,ai,ciact)
        vainic = va
        e = vainic - va
        eaux = e
        de = e - eaux
        inte = ai(6)
        vfinic = ai(6)
        dvf = ai(6) - vfinic
        Do i=2, n
            t(i) = t(i-1) + h
            tinic = t(i)
            Call falla(tinic,tif,tff,vbus,modo_falla)
            Call kutta2(ai,id,iq,te,va,ug,vref,h)
            Call dinamica(ai,id,iq,te,va,p,q,vbus)

            dvf = ai(6) - vfinic
            If (appfam.EQ.1) then
                IA(1) = (e + emaxg)/(2.0*emaxg)
                IA(2) = (de + demaxg)/(2.0*demaxg)
                IA(3) = (inte + intemaxg)/(2.0*intemaxg)
            Else
                IA(1) = (e + emaxg)/(2.0*emaxg)
                IA(2) = (de + demaxg)/(2.0*demaxg)
            End if
            IB(1) = (dvf + dvfmaxg)/(2.0*dvfmaxg)
            Call fam()
            eaux = e
            e = vainic - va
            inte = inte + e
            de = e - eaux
        End do
    END DO
    !*****
    tinic = 0.0
    tfin = 0.0
    tif = 0.0
    ciclos = 0.0
    tff = 0.0
    modo_fam = 2
    rabs = 0.850

```

```

ra = 0.0
rb = 0.850
rab = 0.850
eps = 0.05
Deallocate(t)
Write(50,*) '==== MATRICES RESULTANTES EN LA ARQUITECTURA FAM ====='
Write(50,*)
Write(50,305) (i, i=1, kamax)
Write(50,*)
Write(50,*) 'Matriz WA:'
Write(50,306) ((WA(j,i), j=1, kamax), i=1, 2*ma)
Write(50,*)
Write(50,*) 'Matriz WB:'
Write(50,307) ((WB(j,i), j=1, kbmax), i=1, 2*mb)
Write(50,*)
Write(50,*) 'Matriz WAB:'
Write(50,306) ((WAB(j,i), j=1, kamax), i=1, kbmax)
Write(50,*)
Close(Unit=50)
Write(6,41)
!*** SE EMPLEA LA FAM PARA REGULAR EL VOLTAJE *****
DO WHILE (ejec.EQ.'s')
  re = PARAM(4)
  xe = PARAM(5)
  Write(6,12)
  Read(5,*) tinic
  Write(6,13)
  Read(5,*) tfin
  Write(6,16)
  Write(6,18)
  Read(5,*) evento
  If (evento.EQ.2) then
    Write(6,19)
    Write(6,20)
    Read(5,*) modo_falla
    Write(6,21)
    Read(5,*) tif
    Write(6,22)
    Read(5,*) ciclos
    tff = tif + ciclos/60.0
  Else if (evento.EQ.3) then
    Write(6,23)
    Read(5,*) tc1
    tc2 = tc1 + 20.0
    tc3 = tc2 + 20.0
    tc4 = tc3 + 20.0
  Else if (evento.EQ.4) then
    Write(6,24)
    Read(5,*) tcr1
    tcr2 = tcr1 + 30.0
  End if
  n = (tfin - tinic)/h + 2
  Allocate(t(n))
  If (evento.EQ.3) then
    Do i=1, 4
      CIACT(i) = CITOT(i,31)
    End do
  Else
    Write(6,25)
    Read(5,*) ncon
    Do i=1, 4
      CIACT(i) = CITOT(i,ncon)
    End do
  End if
  Do i=1, n
    t(i) = 0.0
  End do
  t(1) = tinic
  Call condicion_inicial1(va,thet,te,id,iq,vf,p,q,ug,ai,ciact)

```

```

vainic = va
vref = vainic
e = vainic - va
eaux = e
de = e - eaux
inte = vf
vfinic = vf
dvf = vf - vfinic
inva = e**2
Do i=1, 2*mb
  B(i) = 0.0
End do
Open(Unit=31, File="res_fam_se.dat")
Open(Unit=32, File="res_fam_fb.dat")
Open(Unit=33, File="res_fam_cc.dat")
Open(Unit=34, File="res_fam_cr.dat")
Open(Unit=61, File="mapeo_sal.dat")
Open(Unit=62, File="func_memb.dat")
If (evento.EQ.1) then
  Write(31,201) t(1), va, inva, vf, ai(1), p, q, vref
else if (evento.EQ.2) then
  Write(32,201) t(1), va, inva, vf, ai(1), p, q, vref
else if (evento.EQ.3) then
  Write(33,201) t(1), va, inva, vf, ai(1), p, q, ug
else if (evento.EQ.4) then
  Write(34,201) t(1), va, inva, vf, ai(1), p, q, vref
End if
esup = emaxg
desup = 0.001
Do l=2, n
  t(l) = t(l-1) + h
  tinc = t(l)
  If (appfam.EQ.1) then
    IA(1) = (e + emaxg)/(2.0*emaxg)
    IA(2) = (de + demaxg)/(2.0*demaxg)
    IA(3) = (inte + intemaxg)/(2.0*intemaxg)
  Else
    IA(1) = (e + emaxg)/(2.0*emaxg)
    IA(2) = (de + demaxg)/(2.0*demaxg)
  End if
  Call fam()

  dvf = 2.0*B(1)*dvfmaxg - dvfmaxg
  Do i=1, nb
    If (B(1).EQ.CB(i)) then
      j = i
    end if
  End do
  !===== FAM COMO REGULADOR DE VOLTAJE =====!
  If (appfam.EQ.1) then
    vf = dvf + vfinic
  !=====!
  !==== APLICACIÓN DE LA FAM PARA SINTONIZAR UN CONTROL PI ====!
  !===== DIFUSO TIPO MAMDANI =====!
  Else
    X1 = 0.075 + DABS(dvf)*0.025/dvfmaxg
    X2 = 2.0*X1
    If (e.GE.0.0) then
      mue3 = e/esup
      If (mue3.GT.1.0) then
        mue3 = 1.0
      end if
      mue2 = 1.0 - mue3
      mue1 = 0.0
    else
      mue1 = DABS(e/esup)
      If (mue1.GT.1.0) then
        mue1 = 1.0
      end if
    end if
  end if
End do

```

```

        mue2 = 1.0 - mue1
        mue3 = 0.0
    End if
    If (de.GE.0.0) then
        mude3 = de/desup
        If (mude3.GT.1.0) then
            mude3 = 1.0
        end if
        mude2 = 1.0 - mude3
        mude1 = 0.0
    else
        mude1 = DABS(de/desup)
        If (mude1.GT.1.0) then
            mude1 = 1.0
        end if
        mude2 = 1.0 - mude1
        mude3 = 0.0
    End if
    PG = DMIN1(mue3,mude3)
    PP = DMAX1( DMIN1(mue3,mude2),DMIN1(mue2,mude3) )
    CE = DMAX1( DMIN1(mue3,mude1),DMIN1(mue2,mude2),DMIN1(mue1,mude3) )
    NP = DMAX1( DMIN1(mue2,mude1),DMIN1(mue1,mude2) )
    NG = DMIN1(mue1,mude1)
    vf = vf + (X2*PG + X1*PP - X1*NP - X2*NG)/(PG + PP + CE + NP + NG)
    Write(62,202) t(1),mue1,mue2,mue3,mude1,mude2,mude3,X1,X2
End if
!=====
Write(61,309) t(1),(WA(ka,i),i=1,2*ma),(B(i),i=1,2*mb),ka,kb
If (vf.GT.VMAX) then
    vf = VMAX
Else if (vf.LT.VMIN) then
    vf = VMIN
End if
If (evento.EQ.2) then
    Call falla(tinic,tif,tff,vbus,modo_falla)
else if (evento.EQ.3) then
    Call cambio_carga(tinic,tc1,ug,h)
else if (evento.EQ.4) then
    Call cambio_ref(tinic,tcr1,vref,vainic,h)
End if
Call kuttal(ai,id,iq,te,va,ug,vf,h)
Call dinamica(ai,id,iq,te,va,p,q,vbus)
eaux = e
e = vref - va
inte = inte + e
de = e - eaux
inva = inva + e**2
If (evento.EQ.1) then
    Write(31,201) t(1), va, inva, vf, ai(1), p, q, vref
else if (evento.EQ.2) then
    Write(32,201) t(1), va, inva, vf, ai(1), p, q, vref
else if (evento.EQ.3) then
    Write(33,201) t(1), va, inva, vf, ai(1), p, q, ug
else if (evento.EQ.4) then
    Write(34,201) t(1), va, inva, vf, ai(1), p, q, vref
End if
End do
Deallocate(t)
If (evento.EQ.1) then
    Write(6,33)
else if (evento.EQ.2) then
    Write(6,34)
else if (evento.EQ.3) then
    Write(6,35)
else if (evento.EQ.4) then
    Write(6,36)
End if
Close(Unit=31)
Close(Unit=32)

```

```

        Close(Unit=33)
        Close(Unit=34)
        Close(Unit=61)
        Close(Unit=62)
        Write(6,42)
        Read(5,*) ejec
    END DO
    !*****
END IF
!*****

END PROGRAM CONTROL_FAM
!*****
!*****

```

C.2.2 Módulos de Declaración de Variables

El programa incluye dos módulos designados para la declaración de variables. En el primero de éstos se tienen las variables para la definición de los parámetros de la máquina, del regulador ST1 y de los eventos considerados a simular. El segundo módulo contempla la declaración de las variables para los vectores, las matrices y los parámetros que involucra la arquitectura FAM. El código de estos módulos se presenta a continuación.

```

!*** PARÁMETROS DE LA MÁQUINA, DEL REGULADOR ST1 Y DE LOS EVENTOS A SIMULAR ***
MODULE Parametros
IMPLICIT NONE
Integer,Parameter      :: nec=10
Real(8) VMAX, VMIN, raa, re, xe, kd, w0, M, tsm, tch, trh, fhp
Real(8) tx, tq02p, td0p, td02p, xd, xdp, xd2p, xq, xq2p
Real(8) KAA, ta, kf, tf
Real(8) tc2, tc3, tc4, raz1, raz2, raz3, raz4, tcr2, raz5
END MODULE Parametros
!*****

!***** VARIABLES Y PARÁMETROS DE LA ARQUITECTURA FAM *****
MODULE Var_FAM
IMPLICIT NONE
Integer ma, mb, na, nb, ka, kamax, kb, kbm, resa, resb, resab, modo_fam
Real(8) eps, beta, rabs, ra, rb, rab, delta
Real(8),Allocatable :: IA(:), IB(:), A(:), B(:), BAUX(:), WA(:, :)
Real(8),Allocatable :: WB(:, :), WAB(:, :), WJ(:), WK(:), YB(:), XAB(:)
Real(8),Allocatable :: TJ(:), TJAUX(:), TK(:), TKAUX(:), CB(:)
END MODULE Var_FAM
!*****

```

C.2.3 Subrutinas Referentes a la Máquina Síncrona

Dentro de este bloque se encuentran las subrutinas para la simulación de la máquina síncrona, como la obtención de las condiciones iniciales, el desarrollo de la solución numérica por el método de Runge-Kutta de 4º orden y la obtención del resto de las variables de interés mediante la aplicación de las ecuaciones dinámicas. El código de estas subrutinas se presenta a continuación.

```

!***** SUBROUTINAS PARA LA SIMULACIÓN DE LA MÁQUINA SÍNCRONA *****
Subroutine condicion_inicial1(va,thet,te,id,iq,vf,p,q,ug,ai,ciact)
Use Parametros
Real(8),Intent(inout)      :: thet, va
Real(8),Intent(out)        :: p, q, te, id, iq, vf, ug
Real(8),Dimension(nec),Intent(out) :: ai

```

```

Complex(8) ea, eb, pow, zpq
Real(8) phi, del, delr, eqx, eqy, edppr, eqppr, eqpr, ear, eai, ia
Real(8) ciact(4)
va = ciact(1)
thet = ciact(2)
p = ciact(3)
q = ciact(4)
thet = thet*3.141592653589793/180.0
ia = DSQRT(p**2+q**2)/va
If (ia.GT.1.0e-06) then
    phi = DATAN2(q,p)
else
    phi = 0.0
End if
eqx = va + raa*ia*DCOS(phi) + xq*ia*DSIN(phi)
eqy = -raa*ia*DSIN(phi) + xq*ia*DCOS(phi)
del = DATAN2(eqy,eqx)
id = ia*DSIN(del+phi)
iq = ia*DCOS(del+phi)
edppr = va*DSIN(del) + raa*id - xq2p*iq
eqppr = va*DCOS(del) + raa*iq + xd2p*id
eqpr = eqppr + (xdp-xd2p)*id
vf = eqpr + (xd-xdp)*id
te = id*edppr + iq*eqppr
delr = thet + del
ug = te
ai(1) = delr
ai(2) = 0.0
ai(3) = eqpr
ai(4) = eqppr
ai(5) = edppr
ai(6) = te
ai(7) = te
ai(8) = te
End Subroutine condicion_inicial1

Subroutine condicion_inicial2(va,thet,te,id,iq,vref,p,q,ug,ai,ciact)
Use Parametros
Real(8),Intent(inout)          :: thet, va
Real(8),Intent(out)           :: p, q, te, id, iq, vref, ug
Real(8),Dimension(nec),Intent(out) :: ai
Complex(8) ea, eb, pow, zpq
Real(8) phi, del, delr, eqx, eqy, edppr, eqppr, eqpr, ear, eai, ia, vf
Real(8) ciact(4)
va = ciact(1)
thet = ciact(2)
p = ciact(3)
q = ciact(4)
thet = thet*3.141592653589793/180.0
ia = DSQRT(p**2+q**2)/va
If (ia.GT.1.0e-06) then
    phi = DATAN2(q,p)
else
    phi = 0.0
End if
eqx = va + raa*ia*DCOS(phi) + xq*ia*DSIN(phi)
eqy = -raa*ia*DSIN(phi) + xq*ia*DCOS(phi)
del = DATAN2(eqy,eqx)
id = ia*DSIN(del+phi)
iq = ia*DCOS(del+phi)
edppr = va*DSIN(del) + raa*id - xq2p*iq
eqppr = va*DCOS(del) + raa*iq + xd2p*id
eqpr = eqppr + (xdp-xd2p)*id
vf = eqpr + (xd-xdp)*id
te = id*edppr + iq*eqppr
delr = thet + del
vref = (vf/KAA) + va
ug = te
ai(1) = delr

```

```

ai(2) = 0.0
ai(3) = eqpr
ai(4) = eqppr
ai(5) = edppr
ai(6) = KAA*(vref-va)
ai(7) = te
ai(8) = te
ai(9) = te
ai(10) = 0.0
End Subroutine condicion_inicial2

Subroutine kutta1(ainic,id,iq,te,va,ug,vf,h)
Use Parametros
Real(8),Intent(in)                :: id, iq, te, va, ug, vf, h
Real(8),Dimension(:),Allocatable  :: temp, dk1, dk2, dk3, dk4
Real(8),Dimension(nec),Intent(inout) :: ainic
Allocate(temp(nec), dk1(nec), dk2(nec), dk3(nec), dk4(nec))
Call ecdif1(ainic,dk1,iq,id,te,va,ug,vf)
Do i=1, nec
    temp(i) = ainic(i) + dk1(i)*h*0.5
End do
Call ecdif1(temp,dk2,iq,id,te,va,ug,vf)
Do i=1, nec
    temp(i) = ainic(i) + dk2(i)*h*0.5
End do
Call ecdif1(temp,dk3,iq,id,te,va,ug,vf)
Do i=1, nec
    temp(i) = ainic(i) + dk3(i)*h
End do
Call ecdif1(temp,dk4,iq,id,te,va,ug,vf)
Do i=1, nec
    ainic(i) = ainic(i) + (h/6.0)*(dk1(i) + 2.0*dk2(i) + 2.0*dk3(i) + dk4(i))
End do

Deallocate(temp, dk1, dk2, dk3, dk4)
End Subroutine kutta1

Subroutine kutta2(ainic,id,iq,te,va,ug,vref,h)
Use Parametros
Real(8),Intent(in)                :: id, iq, te, va, ug, vref
Real(8),Intent(in)                :: h
Real(8),Dimension(:),Allocatable  :: temp, dk1, dk2, dk3, dk4
Real(8),Dimension(nec),Intent(inout) :: ainic
Allocate(temp(nec), dk1(nec), dk2(nec), dk3(nec), dk4(nec))
Call ecdif2(ainic,dk1,iq,id,te,va,ug,vref)
Do i=1, nec
    temp(i) = ainic(i) + dk1(i)*h*0.5
End do
Call ecdif2(temp,dk2,iq,id,te,va,ug,vref)
Do i=1, nec
    temp(i) = ainic(i) + dk2(i)*h*0.5
End do
Call ecdif2(temp,dk3,iq,id,te,va,ug,vref)
Do i=1, nec
    temp(i) = ainic(i) + dk3(i)*h
End do
Call ecdif2(temp,dk4,iq,id,te,va,ug,vref)
Do i=1, nec
    ainic(i) = ainic(i) + (h/6.0)*(dk1(i) + 2.0*dk2(i) + 2.0*dk3(i) + dk4(i))
End do
If (ainic(6).GT.VMAX) then
    ainic(6) = VMAX
End if
If (ainic(6).LT.VMIN) then
    ainic(6) = VMIN
End if
deallocate(temp, dk1, dk2, dk3, dk4)
End Subroutine kutta2

```

```

Subroutine ecdif1(a,d,iq,id,te,va,ug,vf)
Use Parametros
Real(8),Intent(in)                ::id, iq, te, va, ug, vf
Real(8),Dimension(nec),Intent(in) :: a
Real(8),Dimension(nec),Intent(out) :: d
d(1) = w0*a(2)
d(2) = (-kd*a(2) + a(8) - te)/M
d(3) = (-a(3) + vf - (xd-xdp)*id)/td0p
d(4) = (a(3) - a(4) - (xdp-xd2p)*id)/td02p
d(5) = (-a(5) + (xq-xq2p)*iq)/tq02p
d(6) = (-a(6) + ug)/tsm
d(7) = (a(6) - a(7))/tch
d(8) = (fhp*d(7) + a(7) - a(8))/trh
End Subroutine ecdif1

Subroutine ecdif2(a,d,iq,id,te,va,ug,vref)
Use Parametros
Real(8),Intent(in)                :: id, iq, te, va, ug, vref
Real(8),Dimension(nec),Intent(in) :: a
Real(8),Dimension(nec),Intent(out) :: d
d(1) = w0*a(2)
d(2) = (-kd*a(2) + a(9) - te)/M
d(3) = (-a(3) + a(6) - (xd-xdp)*id)/td0p
d(4) = (a(3) - a(4) - (xdp-xd2p)*id)/td02p
d(5) = (-a(5) + (xq-xq2p)*iq)/tq02p
d(6) = (KAA*(vref-va-a(10)) - a(6))/ta
d(7) = (-a(7) + ug)/tsm
d(8) = (a(7) - a(8))/tch
d(9) = (fhp*d(8) + a(8) - a(9))/trh
d(10) = (kf*d(6) - a(10))/tf
End Subroutine ecdif2

Subroutine dinamica(ai,id,iq,te,va,p,q,vbus)
Use Parametros
Real(8),Dimension(nec),Intent(in) :: ai
Real(8),Intent(in)                :: vbus
Real(8),Intent(out)                :: id, iq, te, va, p, q
Real(8) b1, b2, b3, b4, b5, vd1, vq1
Complex(8) s, vet, tti
b1 = re + raa
b2 = ai(5) - vbus*DSIN(ai(1))
b3 = ai(4) - vbus*DCOS(ai(1))
b4 = xe + xq2p
b5 = xe + xd2p
id = (b1*b2 + b4*b3)/(b1**2 + b5*b4)
iq = (b1*b3 - b5*b2)/(b1**2 + b5*b4)
te = ai(5)*id + ai(4)*iq - (xd2p - xq2p)*id*iq
vd1 = ai(5) - raa*id + xq2p*iq
vq1 = ai(4) - raa*iq - xd2p*id
va = DSQRT(vd1**2 + vq1**2)
vet = DCMLPX(vq1,vd1)
tti = DCMLPX(iq,-id)
s = vet*tti
p = DREAL(s)
q = -DIMAG(s)
End Subroutine dinamica
|*****

```

C.2.4 Subrutinas Referentes a la Arquitectura ARTMAP Difusa

En este bloque se encuentran las subrutinas creadas para la programación de la arquitectura FAM en base al algoritmo descrito en la Sección 2.3.3. En seguida se presenta el código correspondiente.

```

!***** ALGORITMO DE LA ARQUITECTURA ARTMAP DIFUSA (FAM) *****
Subroutine fam()
Use Var_FAM
ra=rabs
Do i=1, 2*mb
  BAUX(i) = B(i)
  B(i) = 0.0
End do
Do i=1, na
  TJ(i) = 0.0
  TJAUX(i) = 0.0
End do
Do i=1, nb
  TK(i) = 0.0
  TKAUX(i) = 0.0
  YB(i) = 0.0
  XAB(i) = 0.0
End do
Do i=1, ma
  If (IA(i).GT.1.0) then
    IA(i) = 1.0
  else if (IA(i).LT.0.0) then
    IA(i) = 0.0
  End if
End do
If (modo_fam.EQ.1) then
  Do i=1, mb
    If (IB(i).GT.1.0) then
      IB(i) = 1.0
    else if (IB(i).LT.0.0) then
      IB(i) = 0.0
    End if
  End do
  Call comp_cod(B, IB, mb)
  Call choice_function(TK, B, WB, eps, 2*mb, nb)
  10 Call cat_choice(TK, TKAUX, nb, kb)
  Do i=1, 2*mb
    WK(i) = WB(kb,i)
  End do
  Call res_res(B, WK, 2*mb, rb, resb)
  If (resb.EQ.1) then
    Goto 20
  Else
    TKAUX(kb)=-1.0
    Do i=1, nb
      If (TKAUX(i).EQ.0.0) then
        Goto 10
      End if
    End do
    Goto 70
  End if
End If
20 Call comp_cod(A, IA, ma)
Call choice_function(TJ, A, WA, eps, 2*ma, na)
30 Call cat_choice(TJ, TJAUX, na, ka)
Do i=1, 2*ma
  WJ(i)=WA(ka,i)
End do
Call res_res(A, WJ, 2*ma, ra, resa)
If (resa.EQ.1) then
  Goto 40
else
  TJAUX(ka)=-1.0
  Do i=1, na
    If (TJAUX(i).EQ.0.0) then
      Goto 30
    End if
  End do
  End do
  Goto 70

```

```

End if
40 Do i=1, na
    TJAUX(i)=0.0
End do
If (modo_fam.EQ.1) then
    Do i=1, nb
        YB(i)=0.0
        If (i.EQ.kb) then
            YB(i)=1.0
        End if
    End do
    Do i=1, nb
        XAB(i)=DMIN1(YB(i),WAB(ka,i))
    End do
    Call map_test(XAB, YB, nb, rab, resab)
    If (resab.EQ.1) then
        Goto 60
    else
        Goto 50
    End if
    50 Call match_tracking(A, WJ, 2*ma, ra, delta)
    If (ra.GT.1.0) then
        Goto 70
    else
        Goto 30
    End if
    60 Call learning_fart(A, WJ, 2*ma, beta)
    Do i=1, 2*ma
        WA(ka,i)=WJ(i)
    End do
    Call learning_fart(B, WK, 2*mb, beta)
    Do i=1, 2*mb
        WB(kb,i)=WK(i)
    End do
    Call learning_map(WAB, na, nb, ka, kb)
    If (ka.GT.kamax) then
        kamax = ka
    End if
    If (kb.GT.kbmax) then
        kbmax = kb
    End if
Else
    If (ka.LE.kamax) then
        Do i=1, kbmax
            XAB(i)=WAB(ka,i)
            YB(i)=XAB(i)
            If (YB(i).GT.0.9) then
                kb = i
            End if
        End do
        Do i=1, 2*mb
            Do j=1, nb
                B(i)=WB(j,i)*YB(j)+B(i)
            End do
        End do
    Else
        Do i=1, 2*mb
            B(i) = BAUX(i)
        End do
    End if
End If
70 End Subroutine fam
!*****
!***** SUBROUTINAS PARA EL ALGORITMO DE LA FAM *****
Subroutine comp_cod(vec_cc, vec_in, tam)
Integer i, tam
Real(8) vec_cc(2*tam), vec_in(tam)
Do i=1, 2*tam

```

```
        vec_cc(i)=0.0
End do
Do i=1, tam
    vec_cc(i)=vec_in(i)
    vec_cc(i+tam)=1-vec_in(i)
End do
End Subroutine comp_cod

Subroutine choice_function(t, vec_in, wmat, eps, tam, nodos)
Integer i, j, tam, nodos
Real(8) t(nodos), vec_in(tam), wmat(nodos,tam), eps, num, den
num=0.0
den=0.0
Do j=1, nodos
    Do i=1, tam
        num=DMIN1(vec_in(i),wmat(j,i))+num
        den=wmat(j,i)+den
    End do
    t(j)=num/(eps+den)
    num=0.0
    den=0.0
End do
End Subroutine choice_function

Subroutine cat_choice(t, taux, nodos, cat)
Integer j, nodos, cat
Real(8) t(nodos), taux(nodos), mayor
mayor=-0.1
Do j=1, nodos
    If (taux(j).GE.0.0) then
        If (t(j).GT.mayor) then
            mayor=t(j)
            cat=j
        End if
    End if
End do
mayor=-0.1
End Subroutine cat_choice

Subroutine res_res(vec, ww, tam, ro, res)
Integer i, tam, res
Real(8) vec(tam), ww(tam), ro, num, den
num=0.0
den=0.0
Do i=1, tam
    num=DMIN1(vec(i),ww(i))+num
    den=vec(i)+den
End do
If ((num/den).GE.ro) then
    res=1
Else
    res=2
End if
End Subroutine res_res

Subroutine map_test(x, y, tam, ro, res)
Integer i, tam, res
Real(8) x(tam), y(tam), ro, num, den
num=0.0
den=0.0
Do i=1, tam
    num=x(i)+num
    den=y(i)+den
End do
If ((num/den).GE.ro) then
    res=1
else
    res=2
End if
```

```

End Subroutine map_test

Subroutine match_tracking(vec, ww, tam, ro, d)
Integer j, tam
Real(8) vec(tam), ww(tam), ro, d, num, den
num=0.0
den=0.0
Do j=1, tam
    num=DMIN1(vec(j),ww(j))+num
    den=vec(j)+den
End do
ro=num/den+d
End Subroutine match_tracking

Subroutine learning_fart(vec, ww, tam, b)
Integer i, tam, comp
Real(8) vec(tam), ww(tam), b
comp=0
Do i=1, tam
    If (ww(i).NE.1.0) then
        comp=1
    End if
End do
If (comp.EQ.0) then
    b=1.0
Else
    b=1.0
End if
Do i=1, tam
    ww(i)=b*DMIN1(vec(i),ww(i))+(1.0-b)*ww(i)
End do
End Subroutine learning_fart

Subroutine learning_map(wmat, col, fil, cata, catb)
Integer k, col, fil, cata, catb
Real(8) wmat(col,fil)
Do k=1, fil
    wmat(cata,k)=0.0
    If (k.EQ.catb) then
        wmat(cata,k)=1.0
    End if
End do
End Subroutine learning_map
|*****

```

C.2.5 Subrutinas referentes a los Eventos a Simular

Las subrutinas creadas para la simulación de los diferentes eventos se encuentran en este bloque, cuyo código es el que se presenta a continuación.

```

|***** SUBRUTINAS PARA LOS EVENTOS A SIMULAR *****
Subroutine falla(tinic,tif,tff,vbus,modo)
Use Parametros
Integer,Intent(in)    :: modo
Real(8),Intent(in)   :: tinic, tif, tff
Real(8),Intent(inout) :: vbus
If ( (tinic.GE.tif) .AND. (tinic.LE.tff) ) then
    vbus = 0.1
Else
    vbus = 1.0
End if
If ( (modo.EQ.2).AND.(tinic.GT.tff) ) then
    re = 0.072
    xe = 0.305
End if

```

```

End Subroutine falla

Subroutine cambio_carga(tinic,tc1,ug,h)
Use Parametros
Real(8),Intent(in)      :: tinic, tc1, h
Real(8),Intent(inout)  :: ug
If (tinic.LT.tc1) then
    ug = 0.0
Else if ( (tinic.GE.tc1) .AND. (tinic.LE.tc2) ) then
    ug = ug + h*raz1
    If (ug.GE.0.3) then
        ug = 0.3
    End if
Else if ( (tinic.GT.tc2) .AND. (tinic.LE.tc3) ) then
    ug = ug + h*raz2
    If (ug.GE.0.6) then
        ug = 0.6
    End if
Else if ( (tinic.GT.tc3) .AND. (tinic.LE.tc4) ) then
    ug = ug + h*raz3
    If (ug.GE.0.8) then
        ug = 0.8
    End if
Else
    ug = ug + h*raz4
    If (ug.GE.0.995) then
        ug = 0.995
    End if
End If
End Subroutine cambio_carga
Subroutine cambio_ref(tinic,tcr1,vref,vrefaux,h)
Use Parametros
Real(8),Intent(in)      :: tinic, tcr1, vrefaux, h
Real(8),Intent(inout)  :: vref
If (tinic.LT.tcr1) then
    vref = vrefaux
Else if ( (tinic.GE.tcr1).AND.(tinic.LT.tcr2) ) then
    vref = vref - h*raz5
    If (vref.LE.0.95*vrefaux) then
        vref = 0.95*vrefaux
    End if
Else
    vref = vref + h*raz5
    If (vref.GE.1.05*vrefaux) then
        vref = 1.05*vrefaux
    End if
End If
End Subroutine cambio_ref
|*****

```

C.3 ARCHIVOS DE ENTRADA Y SALIDA

El programa desarrollado incluye una serie de archivos de entrada y salida necesarios para una correcta ejecución y una mejor visualización de los resultados. Todos estos archivos tienen extensión .dat para que su manipulación entre software de diferentes fabricantes sea más versátil.

C.3.1 Archivos de entrada

En un inicio, el programa necesita leer los datos correspondientes al conjunto de condiciones iniciales y a los parámetros tanto del sistema máquina bus infinito del Apéndice A como de los eventos considerados a simular. Las condiciones iniciales se encuentran en un archivo llamado “cond_inic.dat” y los parámetros del sistema bajo prueba y de los eventos se encuentran en otro archivo llamado “parametros.dat”. Ambos archivos se encuentran en una carpeta llamada “Datos_Entrada”.

En el archivo “cond_inic.dat” se encuentran las 30 condiciones iniciales de la tabla B.1 con una condición inicial extra al final, la cual es utilizada para la simulación de los cambios de carga, razón por la que las variables, a excepción del voltaje en terminales, son cero. Este archivo aparece de la siguiente forma:

```

1.0000000,6.0000000,0.8805000,-0.1361000
1.0426910,5.3615530,0.8971906,0.2394987
1.0291030,3.9811040,0.6517569,0.1460104
0.9998438,5.0219660,0.7357140,-0.1215235
1.0152080,4.2513340,0.6588190,0.0210546
1.0453240,4.0569430,0.7031847,0.2880037
0.9813626,4.1733860,0.5678285,-0.2549196
1.0332250,4.2145420,0.6972846,0.1772876
1.0563940,3.7346060,0.6808600,0.3954524
1.0016400,3.5793800,0.5273286,-0.0787794
1.0270150,4.2609490,0.6890222,0.1221480
1.0389280,3.5363330,0.6077517,0.2420443
0.9919246,4.0302020,0.5712479,-0.1675409
1.0180660,5.7538650,0.8913938,0.0185088
1.0493000,5.7771790,0.9793495,0.2917894
0.9919636,5.7996450,0.8302205,-0.1984313
1.0485680,5.4940960,0.9334847,0.2899110
1.0161500,4.1430040,0.6448828,0.0312059
1.0123020,5.2709690,0.8040097,-0.0222813
1.0179620,5.3985650,0.8376989,0.0234341
1.0427960,3.1271720,0.5543289,0.2858777
1.0470590,3.7457330,0.6595582,0.3102633
0.9967496,5.2069820,0.7552863,-0.1500301
1.0095190,5.1720450,0.7821750,-0.0439321
1.0594970,3.9037920,0.7149498,0.4203133
0.9826534,3.8054920,0.5175695,-0.2373971
1.0168930,6.1285100,0.9445974,0.0027810
1.0000000,6.8077180,1.0000000,-0.1473878
0.9896917,6.5345850,0.9319504,-0.2272967
0.9904293,6.7073600,0.9592349,-0.2236688
1.0000000,0.0000000,0.0000000,0.0000000

```

Por otra parte, el archivo “parametros.dat” aparece de la forma mostrada en la columna de la derecha de la siguiente lista; la columna de la izquierda es sólo para mostrar el parámetro al que se refiere cada valor de la derecha y no aparece en el archivo de salida.

| | |
|--------|----------------------|
| 6.0 | $E_{FD} \text{ máx}$ |
| -6.0 | $E_{FD} \text{ mín}$ |
| 0.0 | r_a |
| 0.024 | r_e |
| 0.115 | x_e |
| 3.0 | k_d |
| 377.0 | ω_o |
| 5.5294 | M |
| 0.1 | T_{SM} |
| 0.15 | T_{CH} |
| 5.0 | T_{RH} |
| 0.33 | F_{Hp} |

| | |
|-------|---|
| 0.065 | T''_{qo} |
| 5.66 | T'_{do} |
| 0.041 | T''_{do} |
| 1.904 | x_d |
| 0.312 | x'_d |
| 0.266 | x''_d |
| 1.881 | x_q |
| 0.266 | x''_q |
| 400.0 | K_A |
| 0.02 | T_A |
| 0.008 | K_F |
| 1.0 | T_F |
| 0.10 | Primera razón de incremento de potencia activa. |
| 0.10 | Segunda razón de incremento de potencia activa. |
| 0.05 | Tercera razón de incremento de potencia activa. |
| 0.05 | Cuarta razón de incremento de potencia activa. |
| 0.05 | Razón de incremento del voltaje de referencia. |

Los últimos cinco parámetros del archivo “parametros.dat” son usados por el programa para incrementar la potencia activa y el voltaje de referencia (tablas 4.4 y 4.5), respectivamente.

C.3.2 Archivos de salida

Los archivos de salida se encuentran en dos carpetas llamadas “Resultados_Gráficas” y “Config_FAM”. La primera de estas carpetas contiene once archivos, de los cuales, dependiendo del tipo de regulación y evento simulados, se elige el archivo predeterminado para escribir los resultados y así graficar posteriormente de forma más conveniente. En la tabla C.1 se presentan los nombres de estos archivos y los resultados que se escriben en cada uno de éstos después de realizada la simulación correspondiente.

Tabla C.1. Nombres de los archivos de salida para la escritura de resultados en función de la simulación realizada.

| Nombre del Archivo (.dat) | Simulación Realizada |
|---------------------------|---|
| res_fam_cc | Regulación de voltaje con la arquitectura FAM ante cambios de carga. |
| res_fam_cr | Regulación de voltaje con la arquitectura FAM ante cambios de referencia. |
| res_fam_fb | Regulación de voltaje con la arquitectura FAM ante una falla en el bus. |
| res_fam_se | Regulación de voltaje con la arquitectura FAM sin evento alguno. |
| res_sc_cc | Sin regulación de voltaje ante cambios de carga. |
| res_sc_fb | Sin regulación de voltaje ante una falla en el bus. |
| res_sc_se | Sin regulación de voltaje sin evento alguno. |
| res_st1_cc | Regulación de voltaje con ST1 ante cambios de carga. |
| res_st1_cr | Regulación de voltaje con ST1 ante cambios de referencia. |
| res_st1_fb | Regulación de voltaje con ST1 ante una falla en el bus. |
| res_st1_se | Regulación de voltaje con ST1 sin evento alguno. |

Los resultados contenidos en los archivos de la tabla C.1 se escriben en ocho columnas, las cuales corresponden a: el tiempo, el voltaje en terminales, el índice de comportamiento del voltaje en terminales, el voltaje de campo, el ángulo de carga, la potencia activa, la potencia reactiva y el voltaje de referencia; esta última variable cambia a la entrada al gobernador de velocidad u_g cuando se simulan cambios de carga. Como ejemplo, en el siguiente listado se muestran los resultados de simular el sistema máquina bus infinito bajo prueba ante una falla de 12 ciclos (200 mseg.) de duración que inicia en $t = 2$ seg. para la condición inicial No. 25 de la tabla B.1, regulando el voltaje en terminales por medio del controlador difuso de la Sección 3.5.1, el cual es sintonizado a través de la arquitectura FAM; evidentemente, en este caso el archivo seleccionado es “res_fam_fb.dat”. Por razones de espacio, sólo se muestran los resultados para el intervalo de tiempo comprendido entre 1.995 seg. y 2.015 seg.

| | | | | | | | |
|--------|--------------|--------------|--------------|---------------|--------------|--------------|--------------|
| 1.9950 | 1.0594963721 | 0.0000013328 | 2.2248673311 | 39.0770302670 | 0.7163819842 | 0.4201118381 | 1.0594970000 |
| 1.9960 | 1.0594963845 | 0.0000013328 | 2.2248664409 | 39.0770328297 | 0.7163820406 | 0.4201119484 | 1.0594970000 |
| 1.9970 | 1.0594963965 | 0.0000013328 | 2.2248655740 | 39.0770352064 | 0.7163820909 | 0.4201120565 | 1.0594970000 |
| 1.9980 | 1.0594964082 | 0.0000013328 | 2.2248647305 | 39.0770373971 | 0.7163821351 | 0.4201121623 | 1.0594970000 |
| 1.9990 | 1.0594964196 | 0.0000013328 | 2.2248639101 | 39.0770394017 | 0.7163821732 | 0.4201122658 | 1.0594970000 |
| 2.0000 | 0.4335676750 | 0.3917888527 | 2.2248631130 | 39.0770412201 | 0.2772513083 | 1.2006850060 | 1.0594970000 |
| 2.0010 | 0.4282277937 | 0.7902896635 | 2.3818377279 | 39.0779004261 | 0.2777191176 | 1.1658627581 | 1.0594970000 |
| 2.0020 | 0.4234675805 | 1.1948230860 | 2.5394131393 | 39.0804730596 | 0.2784542546 | 1.1353485032 | 1.0594970000 |
| 2.0030 | 0.4192170473 | 1.6047815039 | 2.6975241282 | 39.0847558421 | 0.2793819651 | 1.1085357406 | 1.0594970000 |
| 2.0040 | 0.4154145063 | 2.0196237625 | 2.8561133499 | 39.0907445990 | 0.2804420690 | 1.0849078183 | 1.0594970000 |
| 2.0050 | 0.4120057191 | 2.4388687214 | 3.0151304001 | 39.0984345233 | 0.2815864569 | 1.0640243258 | 1.0594970000 |
| 2.0060 | 0.4089430958 | 2.8620891037 | 3.1745309772 | 39.1078203875 | 0.2827769948 | 1.0455095734 | 1.0594970000 |
| 2.0070 | 0.4061849523 | 3.2889057354 | 3.3342761339 | 39.1188967123 | 0.2839837720 | 1.0290428332 | 1.0594970000 |
| 2.0080 | 0.4036948284 | 3.7189822237 | 3.4943316126 | 39.1316578995 | 0.2851836372 | 1.0143500735 | 1.0594970000 |
| 2.0090 | 0.4014408673 | 4.1520200974 | 3.6546672583 | 39.1460983358 | 0.2863589771 | 1.0011969543 | 1.0594970000 |
| 2.0100 | 0.3993952566 | 4.5877544091 | 3.8152564999 | 39.1622124721 | 0.2874966985 | 0.9893828908 | 1.0594970000 |
| 2.0110 | 0.3975337269 | 5.0259497840 | 3.9760758957 | 39.1799948840 | 0.2885873808 | 0.9787360206 | 1.0594970000 |
| 2.0120 | 0.3958351066 | 5.4663968927 | 4.1371047345 | 39.1994403151 | 0.2896245724 | 0.9691089350 | 1.0594970000 |
| 2.0130 | 0.3942809267 | 5.9089093169 | 4.2983246872 | 39.2205437083 | 0.2906042067 | 0.9603750587 | 1.0594970000 |
| 2.0140 | 0.3928550730 | 6.3533207757 | 4.4597195026 | 39.2433002260 | 0.2915241188 | 0.9524255764 | 1.0594970000 |
| 2.0150 | 0.3915434809 | 6.7994826795 | 4.6212747425 | 39.2677052624 | 0.2923836464 | 0.9451668240 | 1.0594970000 |

Por otra parte, la carpeta “Config_FAM” contiene tres archivos en los que se escribe información acerca de la arquitectura FAM y de la forma en que mapea durante las distintas simulaciones. El primero de estos archivos se llama “mapeo_sal.dat” y muestra los vectores prototipo y los números de las categorías elegidas en ambas redes FART de la arquitectura FAM para cada instante de integración. El siguiente listado es resultado del mismo ejemplo de arriba, donde la primer columna representa el tiempo, las siguientes cuatro columnas son los vectores prototipo de las categorías elegidas en la red FART_a, las siguientes dos columnas son los vectores prototipo elegidos en la red FART_b, la penúltima columna indica los números de las categorías a las que corresponden los vectores prototipo de la red FART_a y la última columna indica los números de las categorías de salida (en la red FART_b) definidas por las categorías elegidas en la red FART_a a través del campo de mapeo.

| | | | | | | | | |
|--------|--------------|--------------|--------------|--------------|--------------|--------------|----|---|
| 1.9950 | 0.4988437709 | 0.4999680297 | 0.4932036297 | 0.4999902730 | 0.4999787361 | 0.4750213060 | 1 | 1 |
| 1.9960 | 0.4988437709 | 0.4999680297 | 0.4932036297 | 0.4999902730 | 0.4999787361 | 0.4750213060 | 1 | 1 |
| 1.9970 | 0.4988437709 | 0.4999680297 | 0.4932036297 | 0.4999902730 | 0.4999787361 | 0.4750213060 | 1 | 1 |
| 1.9980 | 0.4988437709 | 0.4999680297 | 0.4932036297 | 0.4999902730 | 0.4999787361 | 0.4750213060 | 1 | 1 |
| 1.9990 | 0.4988437709 | 0.4999680297 | 0.4932036297 | 0.4999902730 | 0.4999787361 | 0.4750213060 | 1 | 1 |
| 2.0000 | 0.4988437709 | 0.4999680297 | 0.4932036297 | 0.4999902730 | 0.4999787361 | 0.4750213060 | 1 | 1 |
| 2.0010 | 0.9067565685 | 0.9988308227 | 0.0922925168 | 0.0000030477 | 0.7310106110 | 0.2461470401 | 2 | 2 |
| 2.0020 | 0.9109524299 | 0.5039794793 | 0.0876231012 | 0.4942046735 | 0.7310106110 | 0.2461470401 | 27 | 2 |
| 2.0030 | 0.9138473548 | 0.5031703685 | 0.0835674635 | 0.4964497742 | 0.7310106110 | 0.2461470401 | 64 | 2 |

| | | | | | | | | |
|--------|--------------|--------------|--------------|--------------|--------------|--------------|----|----|
| 2.0040 | 0.9138473548 | 0.5031703685 | 0.0835674635 | 0.4964497742 | 0.7310106110 | 0.2461470401 | 64 | 2 |
| 2.0050 | 0.9187448309 | 0.5028357100 | 0.0812551691 | 0.4971642900 | 0.7310106110 | 0.2461470401 | 65 | 2 |
| 2.0060 | 0.9208167194 | 0.5020520861 | 0.0756499149 | 0.4965308706 | 0.7310106110 | 0.2461470401 | 30 | 2 |
| 2.0070 | 0.9208167194 | 0.5020520861 | 0.0756499149 | 0.4965308706 | 0.7310106110 | 0.2461470401 | 30 | 2 |
| 2.0080 | 0.9208167194 | 0.5020520861 | 0.0756499149 | 0.4965308706 | 0.7310106110 | 0.2461470401 | 30 | 2 |
| 2.0090 | 0.9258586659 | 0.5015140546 | 0.0715436048 | 0.4972226982 | 0.7310106110 | 0.2461470401 | 31 | 2 |
| 2.0100 | 0.9258586659 | 0.5015140546 | 0.0715436048 | 0.4972226982 | 0.7310106110 | 0.2461470401 | 31 | 2 |
| 2.0110 | 0.9290829984 | 0.5015476318 | 0.0552191978 | 0.4928636019 | 0.7575240374 | 0.2318335856 | 16 | 10 |
| 2.0120 | 0.9290829984 | 0.5015476318 | 0.0552191978 | 0.4928636019 | 0.7575240374 | 0.2318335856 | 16 | 10 |
| 2.0130 | 0.9290829984 | 0.5015476318 | 0.0552191978 | 0.4928636019 | 0.7575240374 | 0.2318335856 | 16 | 10 |
| 2.0140 | 0.9290829984 | 0.5015476318 | 0.0552191978 | 0.4928636019 | 0.7575240374 | 0.2318335856 | 16 | 10 |
| 2.0150 | 0.9293313274 | 0.5001768036 | 0.0399765111 | 0.4957334974 | 0.7310106110 | 0.2461470401 | 12 | 2 |

El segundo archivo de la carpeta “Config_FAM” se llama “pesos_dedif.dat” y contiene los valores de los pesos utilizados para la dedifusificación en cada instante de integración. El siguiente listado es resultado del mismo ejemplo de arriba, donde la primer columna corresponde al tiempo, y las otras dos columnas indican los valores de los pesos X_1 y X_2 , respectivamente.

| | | |
|--------|--------------|--------------|
| 1.9950 | 0.0750010662 | 0.1500021323 |
| 1.9960 | 0.0750010662 | 0.1500021323 |
| 1.9970 | 0.0750010662 | 0.1500021323 |
| 1.9980 | 0.0750010662 | 0.1500021323 |
| 1.9990 | 0.0750010662 | 0.1500021323 |
| 2.0000 | 0.0750010662 | 0.1500021323 |
| 2.0010 | 0.0865505337 | 0.1731010674 |
| 2.0020 | 0.0865505337 | 0.1731010674 |
| 2.0030 | 0.0865505337 | 0.1731010674 |
| 2.0040 | 0.0865505337 | 0.1731010674 |
| 2.0050 | 0.0865505337 | 0.1731010674 |
| 2.0060 | 0.0865505337 | 0.1731010674 |
| 2.0070 | 0.0865505337 | 0.1731010674 |
| 2.0080 | 0.0865505337 | 0.1731010674 |
| 2.0090 | 0.0865505337 | 0.1731010674 |
| 2.0100 | 0.0865505337 | 0.1731010674 |
| 2.0110 | 0.0878762050 | 0.1757524101 |
| 2.0120 | 0.0878762050 | 0.1757524101 |
| 2.0130 | 0.0878762050 | 0.1757524101 |
| 2.0140 | 0.0878762050 | 0.1757524101 |
| 2.0150 | 0.0865505337 | 0.1731010674 |

Por último, el tercer archivo de la carpeta “Config_FAM” se llama “pesos_fam.dat” y contiene las matrices de pesos \mathbf{W}^a , \mathbf{W}^b y \mathbf{W}^{ab} resultantes en las redes FART_a , FART_b y el campo de mapeo, respectivamente. Después de realizado el entrenamiento de la arquitectura FAM, estas matrices aparecen de la siguiente forma:

==== MATRICES RESULTANTES EN LA ARQUITECTURA FAM =====

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|-------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Matriz \mathbf{W}^a : | 0.4988 | 0.9068 | 0.9127 | 0.9603 | 0.6117 | 0.5736 | 0.5230 | 0.4941 | 0.5174 | 0.5453 | 0.5548 | 0.9293 | 0.5602 | 0.5543 |
| | 0.5000 | 0.9988 | 0.5048 | 0.5001 | 0.0362 | 0.4881 | 0.4931 | 0.4969 | 0.4999 | 0.5004 | 0.4996 | 0.5002 | 0.0194 | 0.4920 |
| | 0.4932 | 0.0923 | 0.0433 | 0.0010 | 0.3718 | 0.3816 | 0.4320 | 0.4774 | 0.4712 | 0.4525 | 0.4330 | 0.0400 | 0.4129 | 0.4194 |
| | 0.5000 | 0.0000 | 0.4893 | 0.4954 | 0.9547 | 0.5074 | 0.5020 | 0.4998 | 0.4997 | 0.4996 | 0.4998 | 0.4957 | 0.9651 | 0.5055 |
| | 0.9123 | 0.9291 | 0.9425 | 0.5615 | 0.5072 | 0.9593 | 0.9801 | 0.5736 | 0.5676 | 0.9580 | 0.5776 | 0.5586 | 0.9110 | 0.9152 |
| | 0.5036 | 0.5015 | 0.5000 | 0.4931 | 0.5002 | 0.5000 | 0.5000 | 0.4920 | 0.4926 | 0.5007 | 0.0213 | 0.4936 | 0.5040 | 0.5044 |
| | 0.0710 | 0.0552 | 0.0427 | 0.4361 | 0.4828 | 0.0208 | 0.0101 | 0.4245 | 0.4305 | 0.0400 | 0.4182 | 0.4414 | 0.0876 | 0.0834 |
| | 0.4902 | 0.4929 | 0.4948 | 0.5062 | 0.4997 | 0.4962 | 0.4990 | 0.5078 | 0.5073 | 0.4990 | 0.9739 | 0.5064 | 0.4942 | 0.4948 |

elementos cada uno). Los vectores de esta última matriz contienen un solo elemento diferente de cero (igual a la unidad), que sirve para definir a la categoría apropiada en la red $FART_b$ y, por consiguiente, el vector prototipo de salida de la arquitectura FAM en la fase de producción. Los mapeos en esta fase se efectúan por medio de la matriz \mathbf{W}^{ab} de la siguiente forma: la categoría definida en la red $FART_a$ (vector columna de la matriz \mathbf{W}^a) activa el vector columna que se debe elegir de la matriz \mathbf{W}^{ab} (el mismo número de columna), el cual, de acuerdo con la posición vertical del elemento igual a 1, elige el número de categoría en la red $FART_b$ (número de columna de la matriz \mathbf{W}^b), cuyo prototipo es tomado como salida en determinado instante de integración.

APÉNDICE D

FUNDAMENTOS DE CONTROL DIFUSO

En este segmento se presentan los fundamentos de los sistemas de control difuso, para lo cual es necesario mencionar en primera instancia algunas de las características de los conjuntos difusos y de las operaciones entre éstos. La información de este apartado fue tomada de [17, 19, 30].

D.1 CONJUNTOS DIFUSOS

En general, un conjunto es una colección de objetos que comparten determinadas características. De este modo, si se define un conjunto nítido A dentro de un universo de discurso U y si se define también una determinada característica que deban cumplir los elementos del universo para pertenecer al conjunto A , a los elementos que cumplan tal característica se les asigna el *grado de membresía o de pertenencia* de 1 y se dice que son miembros del conjunto A ; mientras que a aquellos elementos del universo que no cumplan con el requisito, se les asigna el grado de membresía de 0 y se dice que no pertenecen al conjunto A ; esto quiere decir que la frontera del conjunto A es totalmente rígida. En base a esto, se puede definir que

$$\mu_A(x) = \begin{cases} 1 & \text{si y sólo si } x \in A \\ 0 & \text{si y sólo si } x \notin A \end{cases} \quad (\text{D.1})$$

donde $\mu_A(x)$ es el grado de membresía del elemento x al conjunto A . Sin embargo, si se considera que el conjunto \tilde{A} es difuso, entonces el grado de membresía de todos los elementos del universo caerá en el intervalo real $[0, 1]$, entonces se tiene que

$$\mu_{\tilde{A}}(x) = [0, 1] \quad \forall x \in U. \quad (\text{D.2})$$

Por consiguiente, un conjunto difuso \tilde{A} en el universo de discurso U puede ser definido como un conjunto de pares ordenados

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) \mid x \in U\}, \tag{D.3}$$

donde $\mu_{\tilde{A}}(\cdot)$ es la llamada función de membresía (o función característica) de \tilde{A} y $\mu_{\tilde{A}}(x)$ es el grado de membresía de x en \tilde{A} , lo cual indica el grado con el que x pertenece a \tilde{A} . La función de membresía $\mu_{\tilde{A}}(\cdot)$ mapea del universo U al espacio de membresía M ; es decir,

$$\mu_{\tilde{A}} : U \rightarrow M. \tag{D.4}$$

Evidentemente, si $M = \{0, 1\}$, entonces el conjunto en cuestión es nítido; pero si $M = [0, 1]$, entonces el conjunto es difuso. En la tabla D.1 se muestran las funciones de membresía más comunes así como su caracterización matemática.

Tabla D.1. Funciones de membresía más comunes y su caracterización matemática [19].

| | FORMA | CARACTERIZACIÓN MATEMÁTICA |
|-------------|-------|--|
| TRIANGULAR | | $\mu(x; a, b, c) = \begin{cases} 0, & x \leq a. \\ \frac{x-a}{b-a}, & a \leq x \leq b. \\ \frac{c-x}{c-b}, & b \leq x \leq c. \\ 0, & c \leq x. \end{cases}$ <p>o bien</p> $\mu(x; a, b, c) = \text{máx} \left[\text{mín} \left(\frac{x-a}{b-a}, \frac{c-x}{c-b} \right), 0 \right]$ |
| TRAPEZOIDAL | | $\mu(x; a, b, c, d) = \begin{cases} 0, & x \leq a. \\ \frac{x-a}{b-a}, & a \leq x \leq b. \\ 1, & b \leq x \leq c. \\ \frac{d-x}{d-c}, & c \leq x \leq d. \\ 0, & d \leq x. \end{cases}$ <p>o bien</p> $\mu(x; a, b, c, d) = \text{máx} \left[\text{mín} \left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c} \right), 0 \right]$ |
| GAUSSIANA | | $\mu(x; c, \sigma) = e^{-\frac{1}{2} \left(\frac{x-c}{\sigma} \right)^2}$ |

Tabla D.1. Funciones de membresía más comunes y su caracterización matemática (continuación).

| | FORMA | CARACTERIZACIÓN MATEMÁTICA |
|-----------|-------|---|
| CAMPANA | | $\mu(x; a, b, c) = \frac{1}{1 + \left \frac{x-c}{a} \right ^{2b}}$ |
| SIGMOIDAL | | $\mu(x; a, c) = \frac{1}{1 + e^{-a(x-c)}}$ |

D.2 OPERACIONES CON CONJUNTOS DIFUSOS

En la tabla D.2 se resumen las operaciones más comunes entre dos conjuntos nítidos A y B dados dentro de un universo U así como su generalización difusa.

Tabla D.2. Operaciones básicas entre conjuntos nítidos y difusos [19].

| CONJUNTOS NÍTIDOS | CONJUNTOS DIFUSOS | |
|--|--|---|
| Unión | | |
| $A \cup B = \{x \in U \mid x \in A \text{ ó } x \in B\}$ | Máximo | $\tilde{A} \vee \tilde{B} = \text{máx} [\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)]$ |
| | Suma Algebraica | $\tilde{A} \vee \tilde{B} = \mu_{\tilde{A}}(x) + \mu_{\tilde{B}}(x) - \mu_{\tilde{A}}(x)\mu_{\tilde{B}}(x)$ |
| Intersección | | |
| $A \cap B = \{x \in U \mid x \in A \ \& \ x \in B\}$ | Mínimo | $\tilde{A} \wedge \tilde{B} = \text{mín} [\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)]$ |
| | Producto Algebraico | $\tilde{A} \wedge \tilde{B} = \mu_{\tilde{A}}(x)\mu_{\tilde{B}}(x)$ |
| Complemento | | |
| $\bar{A} = \{x \in U \mid x \notin A\}$ | $\bar{\tilde{A}} = 1 - \mu_{\tilde{A}}(x)$ | |

La teoría de conjuntos difusos incluye una gran cantidad de operaciones, definiciones y teoremas que pueden ser muy útiles en el diseño de controles difusos; sin embargo, para aplicaciones no muy complicadas, es suficiente con lo aquí mencionado.

D.3 SISTEMA DE CONTROL DIFUSO

Básicamente, un controlador difuso es un creador de decisiones que opera en un sistema en lazo cerrado en tiempo real. Este controlador obtiene información de la salida de la planta, la compara con la entrada de referencia y después decide cuál es la entrada adecuada hacia la planta para asegurar que la salida del sistema cumpla con los objetivos especificados. Este tipo de controladores no necesitan de un modelo matemático de la planta (o al menos no muy preciso), dado que son capaces de tomar decisiones en base al conocimiento en forma lingüística de un experto en el comportamiento de la planta; es decir, emulan las acciones de un operador humano.

En la figura D.1 se muestra el diagrama de bloques característico de un sistema de control difuso, donde se puede apreciar que este tipo de controladores consta de cuatro bloques fundamentales.

1. Difusificación. En esta etapa se difusifica la entrada del controlador; es decir, se evalúa el grado de membresía de esta señal (entre 0 y 1) a una serie de conjuntos difusos que pueden ser de diferentes formas (como los mostrados en la tabla D.1).
2. Base de Reglas Difusas. Son un conjunto de reglas que se definen con la finalidad de cuantificar el conocimiento lingüístico del experto sobre la forma de lograr un control eficiente. La forma de estas reglas es: *si la entrada es <propiedad>, entonces la salida es <propiedad>*; es decir, dada una premisa, existe un consecuente.

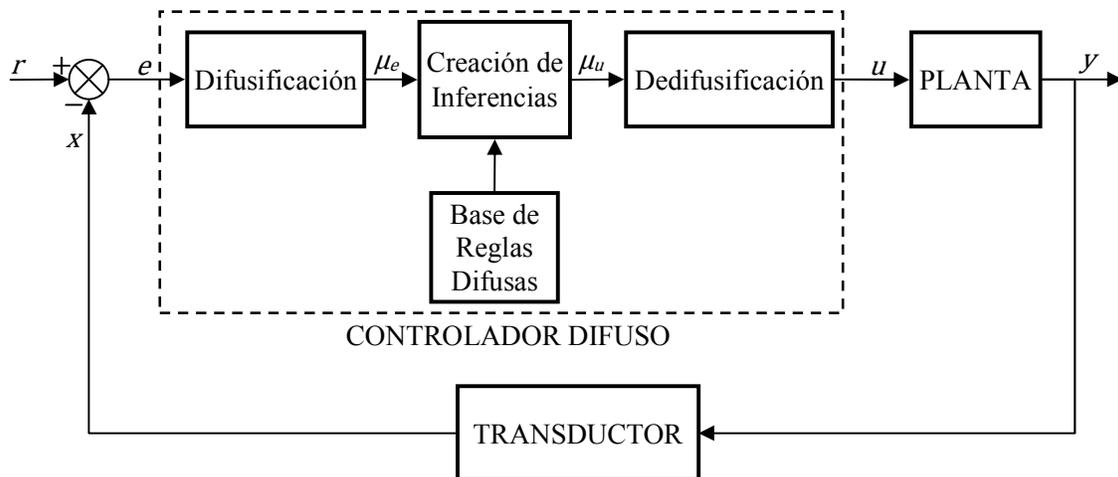


Figura D.1. Sistema de Control Difuso [30].

3. Creación de Inferencias. En esta etapa se emula la toma de decisiones del experto a través de la manipulación de los grados de membresía de la señal de entrada aplicando los operadores de la lógica difusa, y se infieren los valores correspondientes a los grados de membresía de la salida de acuerdo a la base de reglas predefinida.
4. Defusificación. En esta etapa se convierten las conclusiones provenientes de la etapa de creación de inferencias en valores numéricos hacia la planta. Para ello existen diferentes técnicas con distintas características.

D.4 CONTROLES DIFUSOS

La forma más ampliamente utilizada en el diseño y desarrollo de un controlador difuso es basándose en el mismo principio que para los controladores convencionales P, I, PD, PI y PID. Al igual que en los controles convencionales, en los controles difusos proporcionales se utiliza sólo la señal de error, en los integrales se utiliza la integral del error y en los derivativos se hace uso de la derivada del error; siendo las combinaciones de estas señales las que forman los controles difusos PD, PI y PID. Por consiguiente, la entrada de un controlador difuso puede estar constituida, además del error como en la figura D.1, por la integral y/o la derivada del error.

En general, los controladores difusos se clasifican en base al tipo de sistema de inferencias difusas que utilizan, teniéndose los controles tipo Mamdani, Sugeno (también conocido como TSK por Takagi, Sugeno y Kang) y Tsukamoto, (en honor a sus autores). En la figura D.2 se resume la operación del primero de esos controles para dos entradas y una salida; o bien, dos condiciones (premisas) por cada acción (consecuente). Evidentemente, el control de la figura D.2 es el equivalente difuso de un convencional PI, dado que las entradas son el error y su incremento, y la salida es el incremento de la variable manipulada (u en la figura D.1).

Un control difuso tipo Mamdani define las acciones de control también por medio de funciones de membresía, por lo que las reglas difusas para este sistema se expresan como:

SI e es Positivo (P) & Δe es Positivo (P), ENTONCES Δu es Positivo (P).

SI e es Positivo (P) & Δe es Negativo (N), ENTONCES Δu es Cero (C).

SI e es Negativo (N) & Δe es Positivo (P), ENTONCES Δu es Cero (C).

SI e es Negativo (N) & Δe es Negativo (N), ENTONCES Δu es Negativo (N).

En la tabla D.3 se muestra esta misma base de reglas con otra representación muy usual, sobre todo en controladores difusos tipo Mamdani.

Tabla D.3. Base de reglas para el control difuso de la figura D.2.

| $\Delta e \backslash e$ | Negativo (N) | Positivo (P) |
|-------------------------|--------------|--------------|
| Negativo (N) | N | C |
| Positivo (P) | C | P |

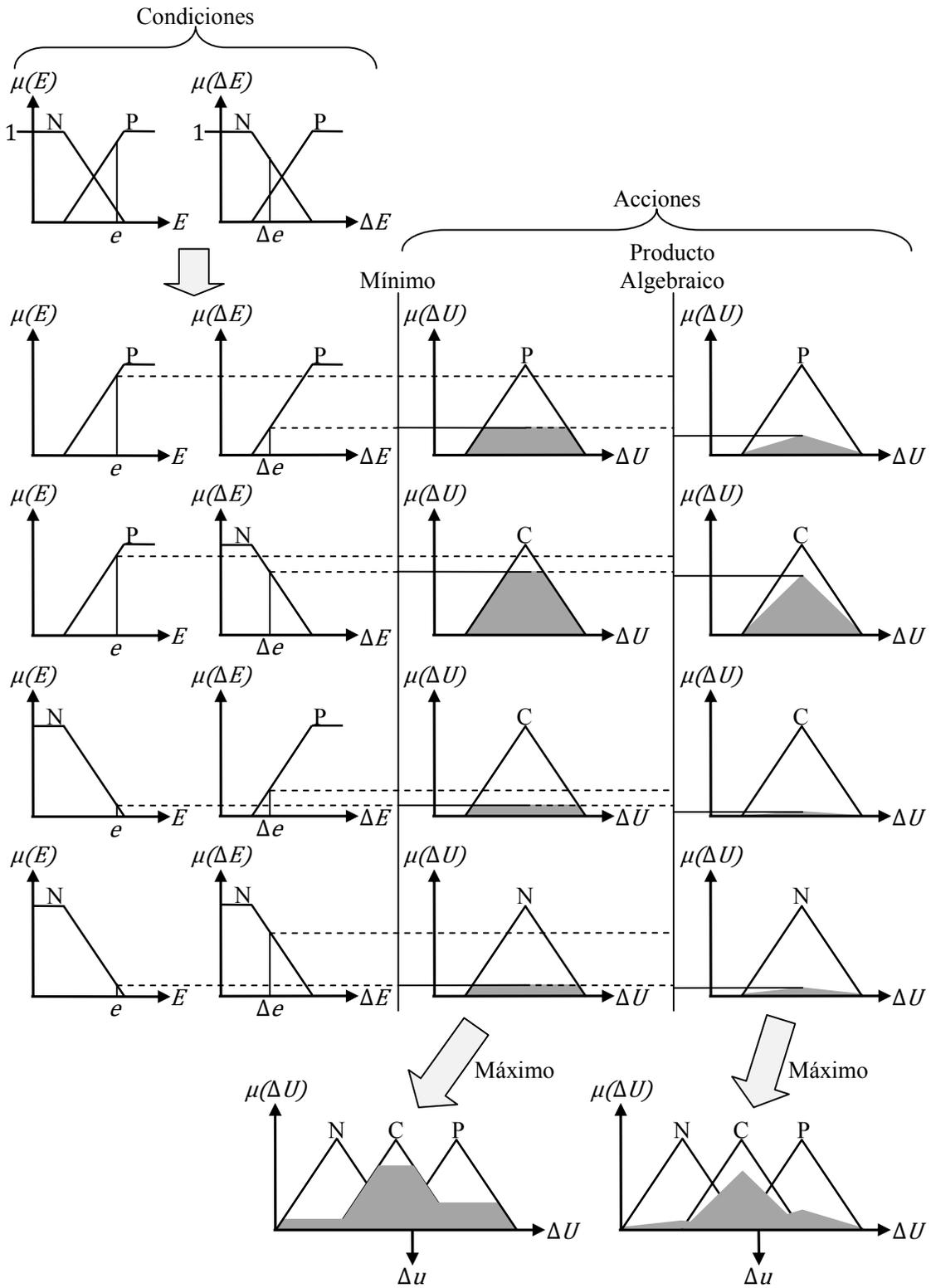


Figura D.2. Control difuso tipo Mamdani para dos entradas y una salida [19].

En la misma figura D.2 se puede ver que, cuando se utiliza el operador mínimo para evaluar la intersección de los conjuntos difusos de las condiciones y para inferir la salida de cada regla, la acción adopta la forma de un conjunto difuso segmentado drásticamente (nítidamente); mientras que, cuando se utiliza el operador producto algebraico, la forma adoptada por la acción es de un conjunto difuso encogido hacia abajo. Las funciones de membresía utilizadas en la figura D.2 son triangulares, pero obviamente pueden ser de cualquiera de las formas mostradas en la tabla D.1; inclusive es posible, aunque no recomendable, usar funciones de membresía de diferentes tipos en las acciones que en las condiciones.

La determinación cuantitativa de la salida Δu del control de la figura D.2 se realiza por alguno de los métodos de dedifusificación existentes. En la tabla D.4 se muestran algunos de estos métodos de forma gráfica junto con su expresión matemática. En esta tabla se puede observar que el método de las alturas resulta ser el mismo que el del centroide con funciones de membresía tipo barra o singleton.

Tabla D.4. Métodos de dedifusificación más comunes [19, 30].

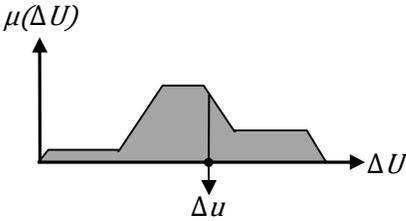
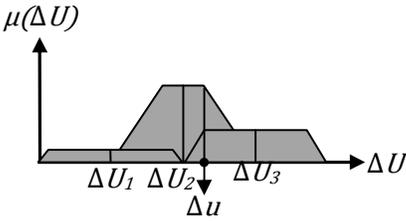
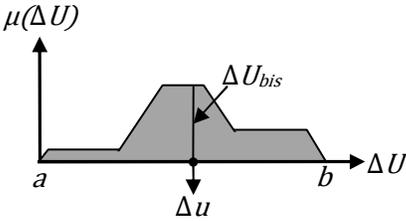
| REPRESENTACIÓN GRÁFICA | EXPRESIÓN MATEMÁTICA | |
|---|---|--|
| Centroide, Centro del Área o Centro de Gravedad | | |
|  | Caso Continuo | Caso Discreto |
| | $\Delta u = \frac{\int \Delta U \mu(\Delta U) d\Delta U}{\int \mu(\Delta U) d\Delta U}$ | $\Delta u = \frac{\sum_{i=1}^n (U_i) (\mu_{U_i})}{\sum_{i=1}^n \mu_{U_i}}$ |
| Alturas | | |
|  | $\Delta u = \frac{\sum_{i=1}^n (U_i) (\mu_{U_i})}{\sum_{i=1}^n \mu_{U_i}}$ | |
| Bisectriz del Área | | |
|  | $\Delta u = \left\{ \Delta U_{bis} \mid \int_a^{\Delta U_{bis}} \mu(\Delta U) d\Delta U = \int_{\Delta U_{bis}}^b \mu(\Delta U) d\Delta U \right\}$ | |

Tabla D.4. Métodos de dedifusificación más comunes (continuación).

| Primera, Última y Media Máxima | |
|--------------------------------|---|
| | <p>Primera Máxima</p> $\Delta u = \text{men}_{\Delta U} \left\{ \Delta U \mid \mu(\Delta U) = \mu(\Delta U)_{sup} \right\}$ <p>men = menor</p> |
| | <p>Última Máxima</p> $\Delta u = \text{may}_{\Delta U} \left\{ \Delta U \mid \mu(\Delta U) = \mu(\Delta U)_{sup} \right\}$ <p>may = mayor</p> |
| | <p>Media Máxima</p> $\Delta u = \left[\text{men}_{\Delta U} \left\{ \Delta U \mid \mu(\Delta U) = \mu(\Delta U)_{sup} \right\} + \text{may}_{\Delta U} \left\{ \Delta U \mid \mu(\Delta U) = \mu(\Delta U)_{sup} \right\} \right] / 2$ |

Por otra parte, en la figura D.3 se muestran los controles difusos tipo Sugeno y Tsukamoto para el mismo caso considerado en la figura D.2. Las condiciones para estos controles son iguales que para el tipo Mamdani (por medio de funciones de membresía), lo que cambia es la acción. El control difuso tipo Sugeno define una expresión matemática en función de las entradas para inferir la salida de cada regla, por lo que la base de reglas en este control se expresa de la siguiente manera:

SI e es Positivo (P) & Δe es Positivo (P), ENTONCES $\Delta u = f_1(e, \Delta e)$.

SI e es Negativo (N) & Δe es Negativo (N), ENTONCES $\Delta u = f_2(e, \Delta e)$.

Las funciones en estas reglas son nítidas y pueden ser tener cualquier forma, aunque usualmente se emplean funciones polinomiales en las variables de entrada; por ejemplo, $f_1(e, \Delta e) = a_1 e + b_1 \Delta e + c_1$ y $f_2(e, \Delta e) = a_2 e + b_2 \Delta e + c_2$.

En el control difuso tipo Tsukamoto, la acción es definida por un conjunto difuso con una función de membresía monótonica, tal y como lo muestra la figura D.3. Como resultado, la salida inferida de cada regla es definida como un valor nítido inducido por el *peso de disparo* de la regla w . La base de reglas de este tipo de control se define como sigue:

SI e es Positivo (P) & Δe es Positivo (P), ENTONCES Δu es C_1 .

SI e es Negativo (N) & Δe es Negativo (N), ENTONCES Δu es C_2 .

Los controles Sugeno y Tsukamoto infieren valores nítidos en cada regla, por lo que la salida de todo el control en ambos casos se determina por medio de la media ponderada, con lo que se evita el consumo de tiempo que involucra un proceso de dedifusificación. Además, en estos controles generalmente no es necesario definir una regla para cada combinación posible de las entradas; muchas veces basta con definir una regla por cada acción, lo que contribuye también a un menor consumo de tiempo.

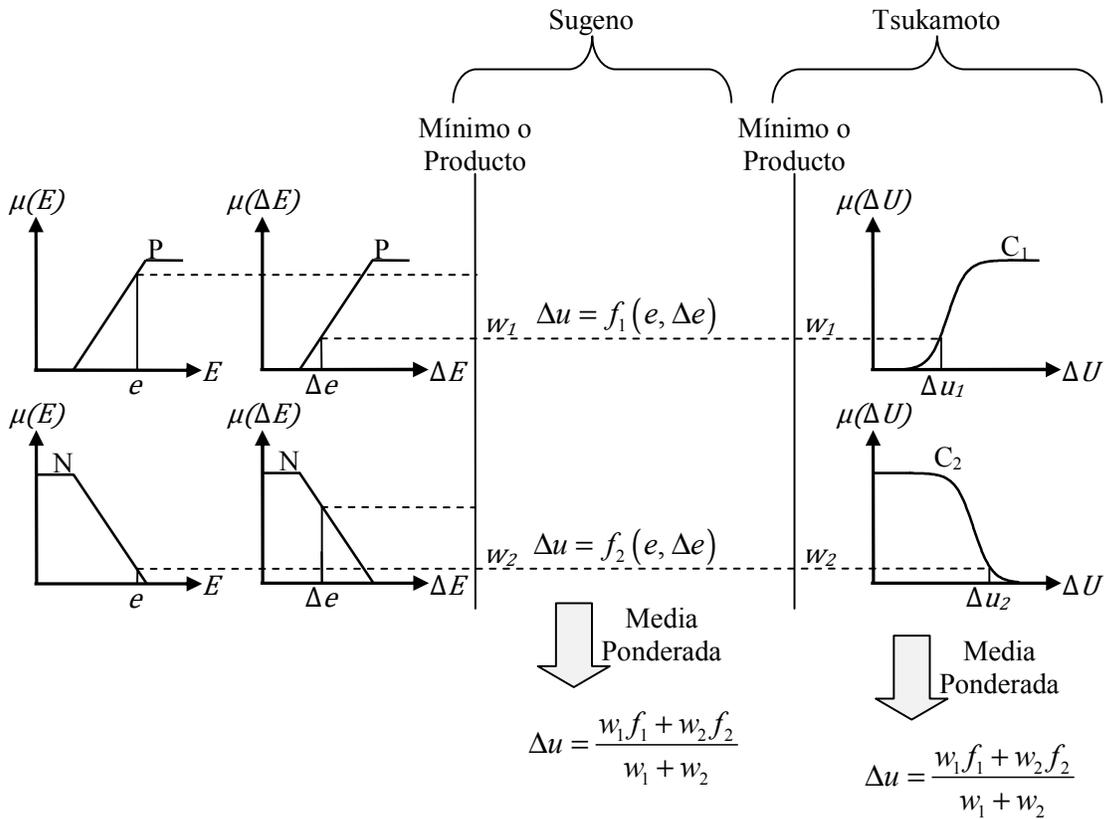


Figura D.3. Controles difusos tipo Sugeno y Tsukamoto para dos entradas y una salida [19].

La sintonización de los controles difusos se realiza por medio de la manipulación de los parámetros involucrados en la definición del valor de salida de cada regla. Por consiguiente, un control difuso tipo Mamdani se sintoniza ajustando los parámetros de las funciones de membresía de las acciones y de las condiciones; un control difuso tipo Sugeno se sintoniza por medio de la selección de las funciones para las acciones y de los ajustes de los parámetros de éstas; y un control difuso tipo Tsukamoto se sintoniza manipulando los parámetros que definen las curvas de las acciones.

CONTROL CON REDES NEURONALES

La habilidad de aprender, de aproximar funciones complejas y de clasificar patrones que caracterizan a las redes neuronales son las razones por las cuales son una gran alternativa en los sistemas de control, puesto que estos sistemas son cada vez más complejos. Además, las redes neuronales son capaces de implementar diversas funciones para controlar sistemas con un alto grado de autonomía [17].

El tipo de red neuronal más comúnmente usado con fines de control es la red multicapa progresiva con aprendizaje supervisado. La principal cualidad que esta red aporta a los sistemas de control es su habilidad de generar mapas de entrada-salida que pueden aproximar cualquier función con un alto grado de precisión. Las redes neuronales han sido utilizadas en los sistemas de control principalmente para *identificación de sistemas* y como *control* en sí. En la identificación de sistemas, la tarea consiste en modelar el comportamiento de entrada-salida de un sistema dinámico; la red es entrenada usando información de entrada-salida y los pesos son ajustados empleando el algoritmo de retropropagación. Por otro lado, cuando una red neuronal multicapa es entrenada como controlador, ya sea en lazo abierto o cerrado, la mayoría de las características son similares a las de una red utilizada en la identificación de sistemas, la diferencia es que la salida deseada de la red no se encuentra disponible, sino que tiene que ser inducida a partir de la salida que se pretende tenga la planta.

Las redes neuronales pueden ser usadas también para detectar fallas en los sistemas y como ayuda en el almacenamiento de información para la toma de decisiones proveyendo la habilidad de decidir cuándo cambiar a un controlador diferente de entre un número determinado de éstos.

La información que se describe en las secciones siguientes fue tomada de [17].

E.1 MODELADO DE UN CONTROLADOR EXISTENTE

En el caso de que exista un controlador capaz de controlar una determinada planta, entonces la información necesaria para entrenar la red neuronal puede obtenerse a partir de este controlador, tal y como se muestra en la figura E.1a. La salida objetivo de la red para una entrada dada es la salida del controlador existente correspondiente a tal entrada. De este modo, la red aprende a modelar (copiar) el controlador existente. Una de las razones para considerar este procedimiento de modelado, es porque el controlador existente puede consistir en un dispositivo poco práctico de utilizar, como un individuo (experto). Además, en algunos casos sólo se conoce un número finito de pares entrada-acción de un controlador deseado, por lo que una red neuronal puede ser entrenada para emular este controlador y así conseguir una interpolación eficiente de estos pares.

E.2 IDENTIFICACIÓN DE SISTEMAS

En el uso de redes neuronales para identificación de sistemas, los datos de entrenamiento pueden ser obtenidos observando el comportamiento de entrada-salida de la planta a identificar, tal y como se muestra en la figura E.1b, donde la red recibe la misma entrada que la planta y la salida de la planta es la salida objetivo de la red. En casos más complejos, la entrada hacia el modelo puede consistir en varios valores retardados de las entradas de la planta y el modelo de la red puede ser un modelo recursivo. Una vez que la dinámica de la planta ha sido identificada por una red neuronal, es posible diseñar un controlador apropiado basándose en esta red.

E.3 IDENTIFICACIÓN DEL SISTEMA INVERSO

La figura E.1c(i) muestra la manera en que una red neuronal puede ser utilizada para identificar la dinámica inversa de una planta, donde la entrada a la red es la salida de la planta y la salida objetivo de la red es la entrada de la planta. Después de este procedimiento, se puede usar la dinámica inversa determinada con propósitos de control como se muestra en la figura E.1c(ii). La salida deseada de la planta es tomada como entrada de la red, y la salida resultante de la red es entonces usada como entrada a la planta. Puesto que la red es una planta inversa, la entrada de la planta genera la salida deseada de la misma. Así, un modelo inverso de una planta no lineal tiene una inmediata utilidad para controlar, a diferencia de un modelo progresivo.

El mayor problema que se tiene con este tipo de aplicaciones es que la dinámica inversa de la planta no siempre está bien definida; esto quiere decir que diferentes entradas de la planta pueden producir la misma salida. En este caso, la red pretende mapear la misma entrada hacia diferentes respuestas objetivo, dando como resultado que la red tienda a promediar sobre varios objetivos y, en consecuencia, produzca mapeos que no necesariamente representen la dinámica inversa requerida.

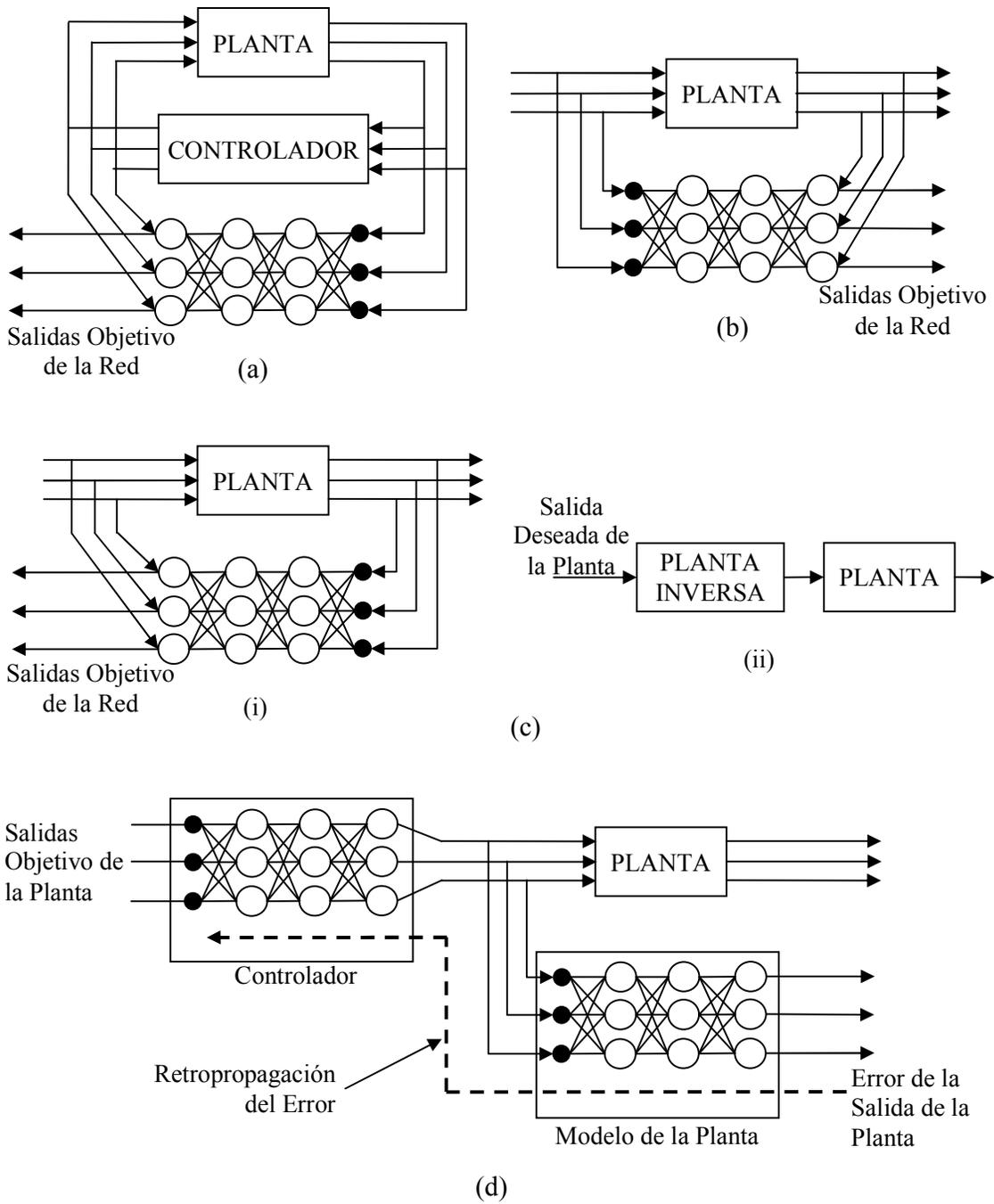


Figura E.1. Configuraciones básicas para el entrenamiento de redes neuronales como identificación de sistemas de control [17]. (a) Modelado de un controlador existente. (b) Identificación de sistemas. (c) Identificación de un sistema inverso. (d) Retropropagación a través de un modelo progresivo de la planta para determinar los errores del controlador.

E.4 DIFERENCIACIÓN DE UN MODELO

En este esquema, una red multicapa es en primera instancia entrenada para identificar la dinámica de la planta, como se muestra en la figura E.1b, después esta red en conjunción con otra red multicapa usada como controlador son configuradas como lo muestra la figura E.1d. La ventaja de un modelo progresivo con esta forma es que se puede estimar eficientemente la derivada de la salida del modelo con respecto a su entrada por medio del proceso de retropropagación, el cual evalúa la transpuesta del Jacobiano de la red en el vector de entrada actual. Como resultado, la retropropagación de los errores entre las salidas actual y deseada de la planta a través del modelo progresivo produce error en la señal de control, la cual puede ser utilizada en el entrenamiento de otra red para ser un controlador. Esta ruta de retropropagación del error se ilustra en la figura E.1d por medio de una línea discontinua.

APÉNDICE F

CONTROLADOR PROPUESTO COMPARADO CON UN CONTROLADOR DIFUSO CON PARÁMETROS FIJOS ANTE UNA FALLA TRIFÁSICA

En este apartado se presenta un conjunto de respuestas obtenidas con el controlador propuesto en comparación con las que se obtienen con el controlador difuso de la Sección 3.5.1 con parámetros fijos; es decir, los pesos en la etapa de dedifusificación no varían durante todo el tiempo de simulación. Los valores de estos pesos son: $X_1 = 0.075$ y $X_2 = 2X_1 = 0.15$. Esta comparación se efectúa con la finalidad de observar la influencia que ejerce la arquitectura FAM sobre el sistema resultante (figura 3.24) al ser utilizada para modificar los pesos del controlador difuso en cuestión.

El evento que se simula para realizar esta comparación es el mismo de la Sección 4.3.2; esto es, una falla de 1.8 ciclos (30 mseg.) de duración para la condición inicial crítica (No. 30) de la tabla 4.1.

Las figuras F.1-F.6 muestran los resultados obtenidos para ambos casos ante el evento simulado. En éstas se puede apreciar que cuando se regula el voltaje en terminales del sistema bajo prueba con el controlador difuso con parámetros fijos, las respuestas tardan más en llegar al estado estacionario que cuando se utiliza la arquitectura FAM para modificar los pesos para la dedifusificación; esto es, evidentemente, debido a que las acciones correctivas ejercidas por el controlador difuso aumentan en estado transitorio y disminuyen en estado estacionario cuando se utiliza la arquitectura FAM de la manera mostrada en la figura 3.24.

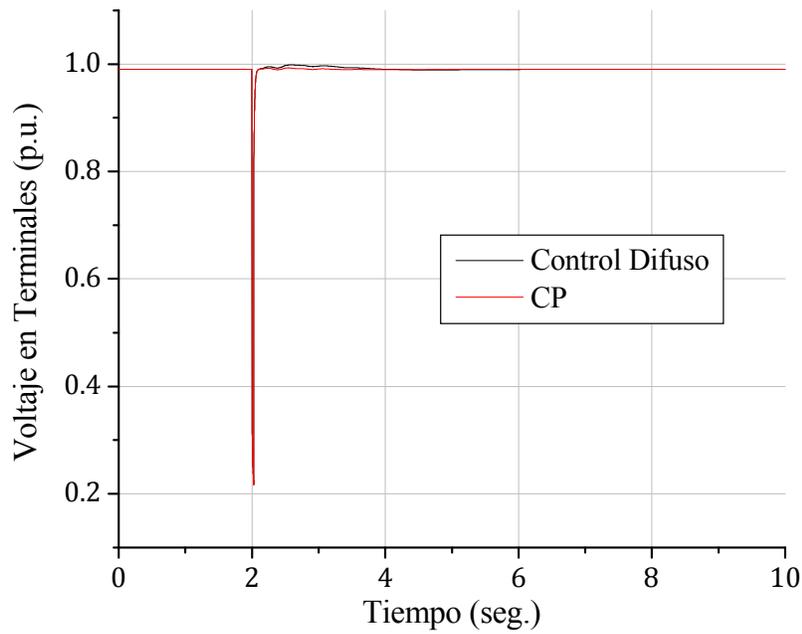


Figura F.1. Voltaje en terminales para la condición inicial No. 30 ante una falla de 1.8 ciclos de duración.

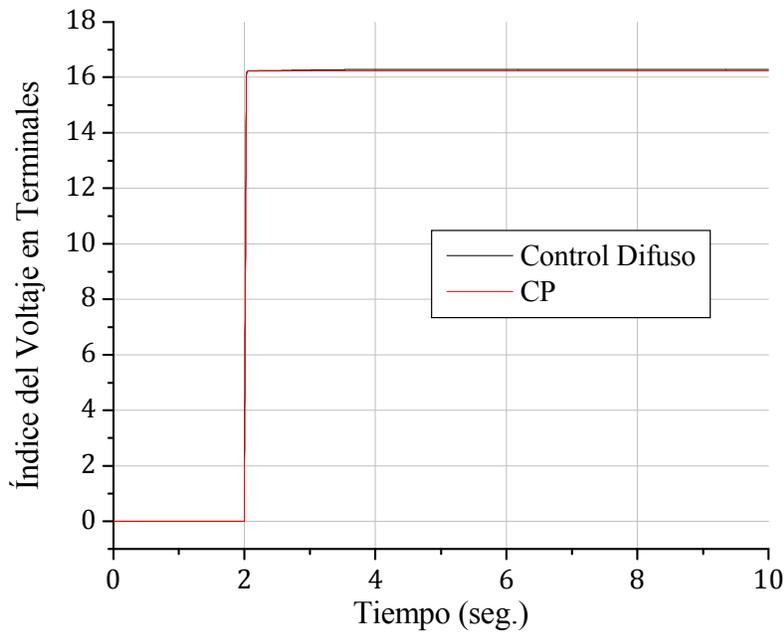


Figura F.2. Índice de comportamiento del voltaje en terminales para la condición Inicial No. 30 ante una falla de 1.8 ciclos de duración.

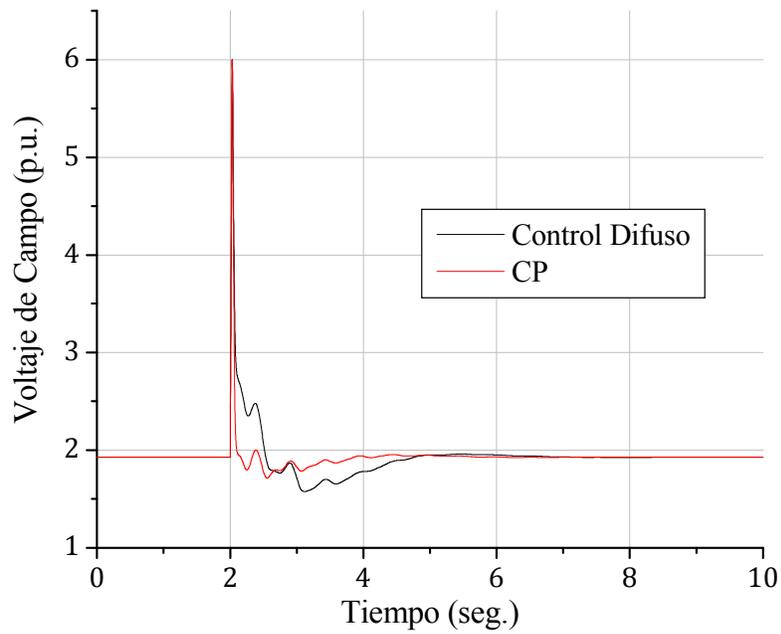


Figura F.3. Voltaje de campo para la condición inicial No. 30 ante una falla de 1.8 ciclos de duración.

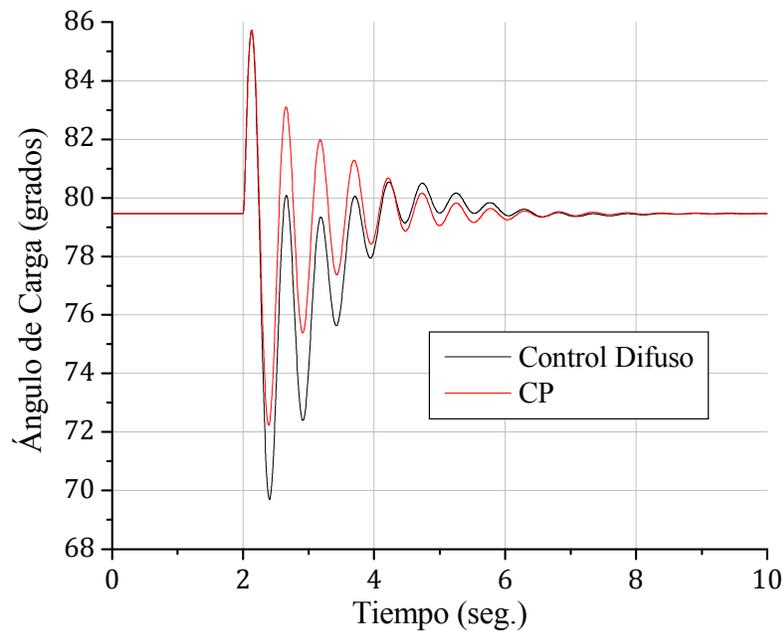


Figura F.4. Ángulo de carga para la condición inicial No. 30 ante una falla de 1.8 ciclos de duración.

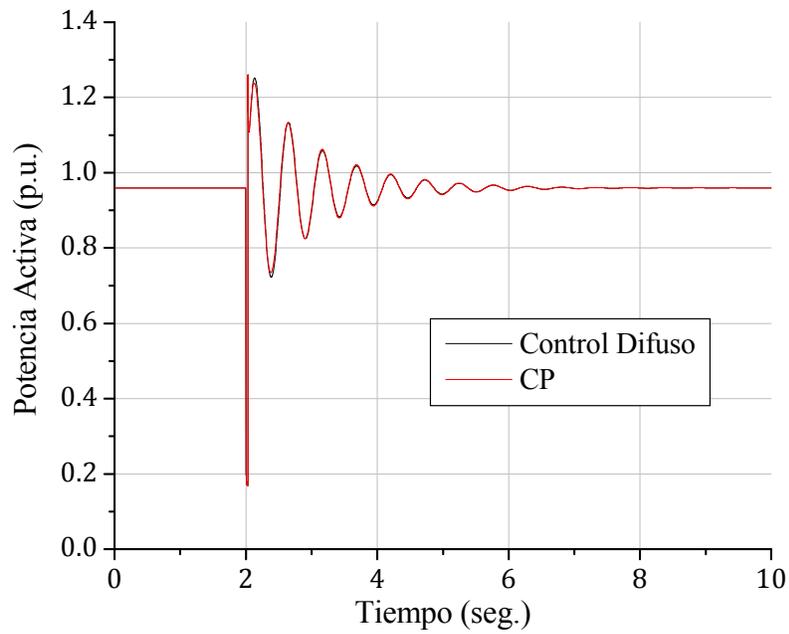


Figura F.5. Potencia activa para la condición inicial No. 30 ante una falla de 1.8 ciclos de duración.

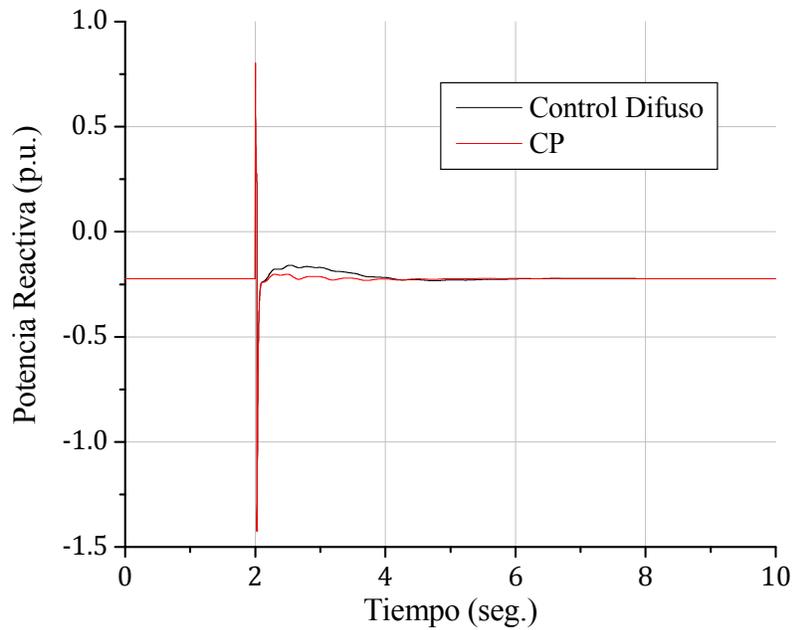


Figura F.6. Potencia reactiva para la condición inicial No. 30 ante una falla de 1.8 ciclos de duración.